

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**DESENVOLVIMENTO DO SISTEMA DE INFORMAÇÃO PARA APOIO
À ANÁLISE DA QUALIDADE DE PLANOS DE GOVERNO COM BASE
NA METODOLOGIA DE BAER (1997)**

Dorival Silva dos Anjos

Florianópolis

2004/2

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**DESENVOLVIMENTO DO SISTEMA DE INFORMAÇÃO PARA APOIO
À ANÁLISE DA QUALIDADE DE PLANOS DE GOVERNO COM BASE
NA METODOLOGIA DE BAER (1997)**

Dorival Silva dos Anjos

*Trabalho de conclusão de curso
apresentado como parte dos requisitos
para obtenção do grau de Bacharel em
Sistemas de Informação*

Florianópolis

2004/2

Dorival Silva dos Anjos

**DESENVOLVIMENTO DO SISTEMA DE INFORMAÇÃO PARA APOIO
À ANÁLISE DA QUALIDADE DE PLANOS DE GOVERNO COM BASE
NA METODOLOGIA DE BAER (1997)**

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Roberto Carlos dos Santos Pacheco, Dr.
Orientador

Banca Examinadora:

Prof^a Aline França de Abreu, Ph D.

Prof. Fernando A. Ostuni Gauthier, Dr.

À memória do meu pai, Lourival,

À minha mãe Marinalva.

À Maria da Luz.

À Roselei Santin.

À Camila S. dos Anjos.

SUMÁRIO

LISTA DE FIGURAS	V
LISTA DE TABELAS	VI
RESUMO	VII
CAPÍTULO 1 - INTRODUÇÃO	1
1 CONTEXTO.....	1
2 QUESTÕES DE PESQUISA	8
3 OBJETIVOS.....	10
4 JUSTIFICATIVA E MOTIVAÇÕES	11
5 METODOLOGIA	12
6 ESTRUTURA DO TRABALHO.....	18
7 DELIMITAÇÃO DO TRABALHO.....	19
CAPÍTULO 2 - AVALIAÇÃO DE PLANOS	21
1 INTRODUÇÃO.....	21
2 DEFINIÇÃO DE PLANO DE GOVERNO	22
3 PRINCIPAIS ABORDAGENS DE DESENVOLVIMENTO DE PLANOS	24
4 ESTÁGIOS PARA AVALIAÇÃO DE UM PLANO DE GOVERNO	27
5 METODOLOGIA DE BAER.....	30
6 CONSIDERAÇÕES FINAIS	36
CAPÍTULO 3 - SISTEMA PROPOSTO	37
1 INTRODUÇÃO.....	37
2 PLANEJAMENTO E ELABORAÇÃO	39
3 DESENVOLVIMENTO	52
4 APRESENTAÇÃO DA INTERFACE COM DIFERENTES TELAS E FORMAS DE USO.....	92
5 CONSIDERAÇÕES FINAIS	140
CAPÍTULO 4 - TESTE DO SISTEMA	141
1 INTRODUÇÃO.....	141
2 TESTES.....	142
CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS.....	147
BIBLIOGRAFIA.....	150
APÊNDICE A - ARTIGO	152
APÊNDICE B - CÓDIGO FONTE.....	161
APÊNDICE C - BANCO DE DADOS	254

LISTA DE FIGURAS

Figura 1 - Mapa conceitual do problema	44
Figura 2 - Diagrama de casos de uso do sistema em UML.....	51
Figura 3 - Diagrama de seqüência do sistema cadastrar usuário.....	54
Figura 4 - Diagrama de seqüência do sistema conexão de usuário.....	55
Figura 5 - Diagrama de seqüência do sistema alterar senha	56
Figura 6 - Diagrama de seqüência do sistema desconexão de usuário	57
Figura 7 - Diagrama de seqüência do sistema cadastro de plano.....	58
Figura 8 - Diagrama de seqüência do sistema avaliação de plano de governo	59
Figura 9 - Diagrama de seqüência do sistema cadastro de critério.....	60
Figura 10 - Diagrama de seqüência do sistema definição de pesos	61
Figura 11 - Diagrama de seqüência do sistema exibir relatório.....	62
Figura 12 - Diagrama de estados do sistema.....	63
Figura 13 - Diagrama de estados quando o sistema está conectado.....	64
Figura 14 - Diagrama de colaboração iniciar o sistema.....	78
Figura 15 - Diagrama de colaboração cadastro de usuário.....	79
Figura 16 - Diagrama de colaboração conexão.....	80
Figura 17 - Diagrama de colaboração alterar senha	81
Figura 18 - Diagrama de colaboração desconexão	82
Figura 19 - Diagrama de colaboração cadastro de plano de governo.....	83
Figura 20 - Diagrama de colaboração avaliação de plano de governo.....	84
Figura 21 - Diagrama de colaboração cadastro de critério.....	85
Figura 22 - Diagrama de colaboração definição de pesos	86
Figura 23- Diagrama de colaboração exibição de relatório	87
Figura 24 - esquema de banco de dados	91
Figura 25 - Janela inicial do sistema	93
Figura 26 - Janela de cadastro de usuário	95
Figura 27 - Janela de conexão de usuário	97
Figura 28 - Janela de alteração de senha do usuário.....	99
Figura 29 - Opção desconexão de usuário na janela inicial	101
Figura 30 - Janela de cadastro de plano de governo	103
Figura 31 - Janela de escolha do plano de governo que será avaliado.....	108
Figura 32 - Janela do questionário de avaliação do plano de governo.....	112
Figura 33 - Janela de cadastro de critério	122
Figura 34 - Janela de definição de pesos dos critérios.....	125
Figura 35 - Janela de definição dos pesos dos tópicos	129
Figura 36 - Janela de definição dos pesos das questões.....	133
Figura 37 - Janela de parâmetros do relatório de avaliação.....	135
Figura 38 - Janela do relatório dos resultados	139

LISTA DE TABELAS

Tabela 1 - Caso de uso iniciar uso	47
Tabela 2 - Caso de uso cadastrar usuário.....	47
Tabela 3 - Caso de uso conectar usuário	47
Tabela 4 - Caso de uso alterar senha de usuário.....	48
Tabela 5 - Caso de uso desconectar usuário	48
Tabela 6 - Caso de uso cadastrar plano de governo.....	48
Tabela 7 - Caso de uso avaliar plano de governo	49
Tabela 8 - Caso de uso cadastrar critério de avaliação de plano de governo	49
Tabela 9 - Caso de uso definir pesos dos tópicos em um critério de avaliação	49
Tabela 10 - Caso de uso definir pesos das questões em um critério de avaliação	50
Tabela 11 - Caso de uso exibir resultados da avaliação de planos de governo.....	50
Tabela 12 - Caso de uso finalizar uso	50
Tabela 13 - Contrato do comportamento do sistema iniciar	66
Tabela 14 - Contrato do comportamento do sistema cadusuario	67
Tabela 15 - Contrato do comportamento do sistema conexao	68
Tabela 16 - Contrato do comportamento do sistema altsenha	69
Tabela 17 - Contrato do comportamento do sistema desconectausuario.....	70
Tabela 18 - Contrato do comportamento do sistema cadplano	71
Tabela 19 - Contrato do comportamento do sistema avaliar	72
Tabela 20 - Contrato do comportamento do sistema cadcritério	73
Tabela 21 - Contrato do comportamento do sistema defpesostopicos.....	74
Tabela 22 - Contrato do comportamento do sistema defpesosquestoes.....	75
Tabela 23 - Contrato do comportamento do sistema relatorios.....	76
Tabela 24 - Contrato do comportamento do sistema sair	77
Tabela 25 - respostas das avaliações para o teste do sistema	144

RESUMO

Um dos principais fatores de sucesso no planejamento público está na qualidade de seu instrumento formal (i.e., em seu plano). Ainda assim, a literatura sobre planejamento público tem-se concentrado muito mais em métodos e processos de elaboração de planos. Recentemente surgiu metodologia que propõe que a avaliação de planos públicos ocorra durante a formulação do mesmo. Trata-se da Metodologia de William Baer, proposta em 1997. Baer inclui os critérios de avaliação do plano como parte de sua concepção, permitindo que o gestor público possa tanto verificar o atendimento a variedade de itens esperados para um plano de governo como o estabelecimento de critérios pelos quais deseja que seu plano venha a ser avaliado. Embora referenciado em diversos projetos públicos, não se verificou disponibilidade de um sistema de informação que venha a sistematizar o apoio aos gestores na atividade de análise de seu planejamento. No presente trabalho, estabelece-se a concepção de um sistema de informação com essa finalidade. O sistema inclui a aplicação da metodologia de Baer. Para verificar a consistência de resultados produzidos pelo sistema, avaliou-se um plano público nas situações limites e intermediária de atendimento aos critérios da metodologia. Os resultados seguem o que é preconizado por Baer.

Palavras chave: Sistemas de Informação, avaliação de planos de governo.

CAPÍTULO 1 - INTRODUÇÃO

[...] a ordem política ainda é antiga e está ainda bloqueada, não há instituições que processem politicamente aquilo que já ocorre na vida econômica, que são transformações que afetam globalmente. A política não é assim ainda, a política ainda é uma política muito mais tradicional, se eu posso dizer assim [...]
(CARDOSO, 2004. p. 8).

1 Contexto

A velocidade lenta com que os assuntos políticos avançam se contrapõe ao ritmo intenso das transformações econômicas, particularmente nos últimos 50 anos, com o acentuado e crescente ritmo da evolução tecnológica. Em ambos os setores, no entanto, planejamento é procedimento fundamental no estabelecimento de metas, diretrizes e ações para alcançá-las. No plano tecnológico, planejamento permite estabelecer cenários, ciclos de vida e tendências para novas tecnologias e seus benefícios. No âmbito da política pública, planos mais eficazes e eficientes, avaliáveis em termos de forma, conteúdo e alcance de resultados, podem diminuir o tempo até que os benefícios da execução exitosa sejam percebidos pela sociedade. Uma questão que surge, então, é se as evoluções técnico-científicas podem contribuir para uma maior efetividade do planejamento público. A tecnologia não pode contribuir de forma isolada. Pode, no entanto, contribuir para a sistematização de metodologias desenvolvidas no âmbito da administração pública que visem melhorar a atividade de planejamento público.

Esse trabalho tem como contexto e motivação essa aproximação entre a tecnologia da informação e o planejamento público. Para tal, adotou-se metodologia internacionalmente conhecida para avaliação de planos de governo, que foi sistematizada em um sistema de informação específico, que permite a simulação de

avaliação de planos governamentais, apontando deficiências que podem, segundo a metodologia adotada, indicar fragilidades futuras na execução do plano.

Trata-se, portanto, de iniciativa que visa contribuir no sentido da mudança da tendência de distanciamento dos tempos da evolução do plano político e das tecnologias, por meio justamente do aporte de tecnologia da informação à metodologia de avaliação dos planos de governo.

I Planos de governo

Quando o governo determina a implementação de um projeto, ele deve ter um planejamento geral. A execução de uma ação do governo, seja ela imediata ou de longo prazo, demanda, em geral, grande quantidade de recursos financeiros dentre outros recursos. A administração pública deve se cercar de cuidados para que as dificuldades que venham a surgir sejam antecipadas no plano e, portanto, contornadas ou, na pior hipótese, que os prejuízos advindos dessas dificuldades sejam minimizados. Além disso, os recursos devem ser utilizados de forma mais racional possível, e os objetivos determinados não podem ser esquecidos com o passar do tempo (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Os planos de governo são muito importantes para o desenvolvimento da sociedade, segundo Dalton, "... o planejamento permanece como uma das principais ferramentas do planejador para influenciar o crescimento futuro e o desenvolvimento" (1989 apud BAER, 1997, p.329). Essa afirmação deixa clara a importância que os planos de governo têm para o desenvolvimento social e o crescimento, que são indispensáveis (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Os planos de governo sugerem o uso racional dos recursos e o alcance dos objetivos para os quais foi desenvolvido. O êxito no cumprimento dos objetivos depende da capacidade do plano abranger e propor soluções para as diversas situações e problemas que possam surgir após a sua implantação (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001). O sucesso de um plano depende da qualidade desse plano. (BAER, 1997)

Nesse contexto, uma questão que naturalmente se coloca é como definir a qualidade de um plano de governo? Não existe uma receita simples para o desenvolvimento de um plano de governo com qualidade (BAER, 1997). Além disso, segundo Berke e French (1994 apud BAER, 1997, p.329), "é difícil definir a qualidade de um plano de governo". Como os planos de governo são desenvolvidos para atender demandas e áreas diversas e casos específicos, não podem, dessa forma, ser padronizados. A padronização poderia ser uma maneira de garantir qualidade, mas é inviável (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

A necessidade de planos de qualidade tem feito com que pessoas e instituições busquem a melhoria dos planos de governo:

Planejadores tem conseguido algo em torno de um sucesso por ano nos últimos 50 anos no seu empenho para convencer legisladores estaduais em requerer planejamentos municipais compreensivos. (COBB, 1994, apud PENDALL, 2001, p 154).

Os governos estaduais norte-americanos estão, cada vez mais, ordenando, através de requerimentos metódicos e até exaustivos, preparação para planejamento local. (GALÉ 1992; INNES 1992; BUSHBAUM e SMITH, 1993, apud BAER, 1997, p.329)

Como a elaboração de planos de governos é essencial para a condução das atividades governamentais e uma ferramenta para influenciar o crescimento e o desenvolvimento (DALTON 1989 apud BAER, 1997, p.329), pesquisadores vem desenvolvendo métodos de avaliações do plano em suas diversas fases, visando a melhoria desses planos, através de avaliações (QUEIRÓS, 2001), como será visto adiante.

II Avaliação de planos de governo

A Práticas

Alguns profissionais estudam formas de avaliar planos de governo e propõem métodos que, se aplicados aos planos, podem dar uma aproximação de sua qualidade. Um desses pesquisadores, William Baer, professor de Planejamento Urbano e Desenvolvimento na Escola de Planejamento e Desenvolvimento Urbano da Universidade Southern Califórnia, desenvolveu uma teoria de avaliação de planos de governo. Como evidenciou-se durante a presente pesquisa, o trabalho de Baer é referência para trabalhos de avaliação de planos. Pesquisas sobre planejamento urbano, abordando, por exemplo, os problemas ambientais advindos de planejamentos urbanos, planejamentos urbanos com finalidades discriminatórias ou econômicas, dentre outros. Notou-se, então, que a teoria de Baer pode ser considerada como referência para os profissionais do planejamento.

Baer aborda o tema da avaliação de planos de forma abrangente e detalhada e aponta uma lista de critérios atuais para avaliar a documentação que expressa um plano de governo, favorecendo a sua sistematização. Por esse motivo será tomado como base para o desenvolvimento teórico do presente trabalho, não

descartando, no entanto, outros artigos que possam contribuir para o desenvolvimento das idéias do mesmo.

B Uso de Sistemas de informação na avaliação de planos de governo

Durante a pesquisa bibliográfica para este trabalho, procuraram-se publicações que tratassem da aplicação de sistemas de informação no apoio à avaliação de planos de governo. No entanto, essas publicações não foram encontradas. Não foi encontrada, também, nenhuma referência indicando simplesmente a existência de um sistema que fora utilizado ou que pudesse apoiar tal tarefa. Parece haver uma lacuna a ser preenchida nesta área. Os avaliadores de planos de governo podem até mesmo utilizar computadores no momento da avaliação de planos, mas é provável que o apoio que têm obtido da Tecnologia da Informação está aquém do que se pode realizar, especialmente quanto à avaliação sistematizada ou reutilizável em novas avaliações.

Sem sistemas especificamente projetados para a tarefa de avaliação de planos, o avaliador pode, por exemplo, criar um texto ou planilha eletrônica, digitar todas as questões que se referem à avaliação e respondê-las. Posteriormente, analisa o texto ou planilha resultante para descobrir as condições necessárias que não foram atingidas para chegar as conclusões da avaliação. Provavelmente, nessa sistemática de trabalho, o avaliador necessita iniciar o processo a cada avaliação, criando um novo documento apontando as ações a tomar para corrigir as condições não satisfeitas. Procedimentos como esse configuram-se em mau utilização dos recursos atualmente oferecidos pela Tecnologia da Informação. Os sistemas de informação podem ir muito além do que a tarefa de servirem como editores de textos ou de planilhas eletrônicas. Quando corretamente projetados para uma tarefa, os

sistemas de informação possuem a capacidade de processar e gerar novas informações tais como relatórios, advertências para erros cometidos, orientações sobre ações a serem tomadas, dirigindo dessa forma o usuário por um caminho correto. A reutilização das informações armazenadas produz economia de tempo e de trabalho do usuário e a administração dessa informação deixa o usuário livre para se preocupar exclusivamente com a atividade fim, que é, avaliar o plano.

Esse é justamente o objetivo desse trabalho. Por meio da concepção, análise e desenvolvimento de um sistema de informação especificamente projetado, pretende-se apoiar a tarefa de avaliação de planos de governo.

III Sistemas de Informação como apoio à avaliação de planos

Este trabalho visa à construção de um Sistema de Informação como apoio à avaliação de planos de governo. De acordo com Abreu & Abreu, um Sistema de Informação (SI), pode ser definido como: "[...] um conjunto de procedimentos que coletam (ou recuperam), processam, armazenam e disseminam informações para o suporte na tomada de decisões [...]" (ABREU & ABREU, 2000, p. 14).

Esta definição deixa claro quais são os elementos componentes e a finalidade de um SI:

- conjunto de procedimentos: no caso, procedimentos escritos na forma de instruções que podem ser interpretadas e executadas por um computador. Deseja-se que a avaliação de planos de governo, no todo ou em parte, possa ser sistematizada na forma de procedimentos (também chamados algoritmos);

- coleta (ou recuperação) de informações: os procedimentos de um SI objetivam coletar, de alguma forma, informações do mundo externo ao sistema, ou então recuperar informações que foram coletadas no passado e armazenadas, como se verá mais adiante;
- processamento de informações: além de coletar e recuperar informações, SI possuem meios de processá-las e fornecer saídas inteligíveis aos usuários no domínio de seu interesse. O tratamento da informação é a tarefa repetitiva, que poderá ser sistematizada e transformada em algoritmos. O processamento da informação pode ser feito inúmeras vezes pelo computador, em uma velocidade superior a humana, economizando o tempo das pessoas envolvidas;
- armazenamento de informações: se um sistema coleta informação, é importante que o mesmo apresente alguma forma de armazená-lo para em outra oportunidade, recuperá-la, evitando a necessidade de coletá-la de sua origem novamente, o que seria um trabalho desnecessário;
- disseminação de informações: após processar informações, o sistema pode gerar novas informações que serão o resultado do processamento, essas novas informações precisam ser disseminadas, tanto para estarem disponíveis a usuários como acessíveis a outros sistemas de informação.

Sistemas de Informação, portanto, são configurados em domínios de aplicação específicos. Um domínio é caracterizado pela área, pelo setor ou pelo campo de utilização do sistema. A avaliação de planos de governo constitui um domínio de aplicação. Como tal, é necessário evidenciar em que medida as tarefas

realizadas nesse domínio são passíveis de formulação como procedimentos modeláveis e programáveis em instruções computacionais.

Uma análise da Metodologia de Baer evidencia que esse é o caso, justamente quando a avaliação de planos segue seus procedimentos. Isso ocorre, por exemplo, durante a fase de sua formulação, quando o profissional está desenvolvendo os critérios com que avaliará o plano, há um grau de repetitividade. O planejador não acessa o plano de uma única vez, e sim por etapas, fazendo avaliações parciais. Após ser avaliado, um plano pode precisar de correções e ser novamente avaliado. Esse ciclo pode se repetir inúmeras vezes. Visto que um Sistema de Informação tem a capacidade de armazenar e recuperar informações, o planejador pode se valer desse recurso para armazenar as informações de uma avaliação parcial, e, após alterar o plano, em qualquer tempo, recuperar as informações da avaliação anterior, reaplicando a avaliação apenas nos itens que não estão satisfatórios. Um segundo benefício é o processamento das informações armazenadas, gerando novas informações, e, por último, estas informações podem ser disseminadas pelo sistema atendendo as necessidades do planejador ou outra pessoa interessada (BAER, 1997; ABREU & ABREU, 2000).

2 Questões de Pesquisa

As idéias que nortearam o desenvolvimento deste trabalho se resumem nas duas questões apresentadas e discutidas a seguir:

I É possível aplicar Sistemas de Informação na avaliação de planos de governo?

Apesar de terem sido encontradas diversas publicações na área de avaliação de planos de governo, não se observou nenhuma publicação ou referência que tratasse da aplicação de sistemas de informação no apoio à avaliação de planos de governo como será tratado no presente trabalho. Dessa forma, esta primeira questão, e sua resposta, podem trazer uma contribuição ao estudo da avaliação de planos de governo, trazendo uma nova possibilidade na forma com que se verifica a efetividade de planos.

Para tal, é necessário analisar a natureza das técnicas disponíveis para a avaliação de planos de governo, procurando-se por componentes (tarefas) que possam ser sistematizados. Ainda que avaliar planos de governo seja uma tarefa intrinsecamente subjetiva, pois dependente do parecer humano, não se pode descartar a possibilidade de sistematização das tarefas de natureza procedural. Elas surgem justamente em metodologias que prescrevem passos à avaliação de planos, como ocorre com a Metodologia de Baer.

Sistematizar a avaliação de planos não é uma tarefa dos profissionais da Tecnologia da Informação e sim de pesquisadores ligados à área de planejamento. A sistematização, no entanto, evidencia tarefas processuais que podem ser modeladas na forma de procedimentos computacionais, que dão base a um sistema de informação. O estudo da Metodologia de Baer mostrou que a avaliação de planos pode ser realizada como tarefa sistêmica e, mais especificamente, que esse modelo pode ser base para a concepção e desenvolvimento de um sistema de informação, conforme proposto no presente trabalho.

II É possível apoiar o avaliador com Sistemas de Informação?

Esta segunda questão complementa a primeira, adicionando àquela um caráter mais prático. E, como resposta a esta questão, neste trabalho é proposto o desenvolvimento de um protótipo de Sistema de Informação que poderá apoiar o avaliador, bem como outras pessoas interessadas, em algum momento da avaliação de algum tipo de plano de governo.

O sistema proposto visa mostrar que é possível apoiar o avaliador na avaliação de planos, naquilo em que couber e, como, um sistema precursor, servir de estímulo, base e/ou modelo para outros sistemas que possam ser desenvolvidos no futuro visando objetivos semelhantes. Não se pretende ser exaustivo na proposição de uma solução informatizada ao apoio à avaliação de planos e sim abrir caminho para o desenvolvimento de novos sistemas que apresentem um aprofundamento maior, tanto no que se refere à metodologia de avaliação de planos como na utilização de outras abordagens sistêmicas como base de concepção de SI.¹

3 Objetivos

I Objetivo geral

Desenvolver um Sistema de Informação para apoio à avaliação de planos de Governo, com aplicação em planos de governo.

¹ Nesse sentido, o eventual aprimoramento do sistema proposto por quem desejar dar continuidade a este trabalho será facilitado, já que o código fonte do sistema estará disponível a interessados.

II Objetivos específicos

Esse trabalho deverá, adicionalmente, buscar os seguintes objetivos:

- Estudar a área de planos de governo, quanto à sua definição, relevância, componentes e, principalmente, metodologias de avaliação;
- Identificar os tipos de avaliação de planos de governo que têm natureza sistêmica e, como tal, podem ser sistematizadas e automatizadas em um sistema de informação;
- Desenvolver um sistema de domínio público, o qual implementa o tipo de avaliação escolhido, e disponibilizar esse sistema;
- Aplicar o sistema em um plano de governo disponível ao público, com o objetivo de verificar os resultados que o sistema apresenta nas extremidades das análises (i.e., quando os critérios são integralmente insatisfatórios ou satisfatórios) e em situações intermediárias.

4 Justificativa e Motivações

Apesar da existência de metodologias para avaliação de planos governamentais datarem de mais de uma década, conforme descreve Baer (1997), não se encontrou disponível na literatura pesquisada o registro de sistema de informação que apóie a aplicação de nenhuma destas metodologias. Com isso, não só a execução de avaliações é dificultada como a própria divulgação da prática de análise de planos de governo se mostra mais distante de gestores públicos. Assim, acredita-se que a concepção, desenvolvimento e divulgação de um sistema com essa natureza venham tanto apoiar aqueles gestores que praticam a tarefa da

análise de planos governamentais como permitir que essa prática seja mais facilmente conhecida por gestores públicos.

O trabalho proposto mostra-se relevante também porque tem potencial para servir de ponto de partida para aprimoramentos em sistemas de informação aplicados à avaliação de planos de governo. Nesse sentido, uma das contribuições está no fato de a pesquisa ter verificado a insipiência da área de avaliação qualitativas de planos no País.

5 Metodologia

Nesta seção é apresentada a metodologia de pesquisa empregada, incluindo a descrição e justificativa da escolha do tipo de pesquisa utilizado, os locais de pesquisa, uma breve descrição do motivo da escolha da plataforma onde foi desenvolvido o sistema, descrição das ferramentas e tecnologias empregadas e, por fim, uma sucinta descrição dos resultados da pesquisa.

Este trabalho, como outros que objetivam o desenvolvimento de um sistema, foi inicialmente subdividido em duas fases: a primeira fase se constituiu na pesquisa sobre o domínio do sistema de informação a ser desenvolvido, e, ainda nesta fase, pesquisa e escolha das tecnologias que poderiam ser utilizadas no desenvolvimento do sistema. Na segunda fase, desenvolveu-se o sistema propriamente dito e, posteriormente, esse sistema foi aplicado em um plano de governo, com o objetivo de se verificar o comportamento do sistema em diferentes extremos e situações intermediárias de caracterização do plano.

I Pesquisa teórica e escolha de tecnologias

A Pesquisa

Durante a primeira fase, na pesquisa do tema abordado, a metodologia utilizada foi a da pesquisa bibliográfica, que, segundo Gil (1991), é feita a partir de material já elaborado, como livros, e artigos científicos. Essa pesquisa foi feita na biblioteca central da Universidade Federal de Santa Catarina (UFSC), em biblioteca digital de teses e dissertações, na internet, no portal da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e em livros. O portal da CAPES é um convênio no qual, através de uma conexão com uma universidade, alunos e professores podem consultar periódicos publicados em todo o mundo por inúmeras instituições e autores nas mais diversas áreas do conhecimento. No portal da Capes, foram encontrados alguns dos artigos que auxiliaram a pesquisa.

Foram encontrados diversos artigos e, na biblioteca, um livro cujo tema trata sobre planejamento. Os trabalhos encontrados, em sua maioria, são específicos, abordando, em geral a avaliação de um determinado plano de governo e apresentando comentários a respeito do plano em estudo. Estes trabalhos serviram de base teórica e enriqueceram a pesquisa, mas foi um artigo de Baer que estabeleceu a metodologia de avaliação de planos de governo mais adequada à aplicação de um sistema de informação.

Baer desenvolveu uma metodologia que analisa planos com base em sessenta critérios, que, segundo ele, podem avaliar a qualidade de planos ainda em sua fase de sua formulação (BAER 1997, p. 329). Esses sessenta critérios são agrupados em oito classificações básicas (BAER,1997, p. 338). O sistema proposto deverá apoiar o avaliador no momento da aplicação do questionário, guardando as

respostas resultantes da aplicação deste questionário, permitindo ao avaliador alterar os pesos de cada questão de acordo com a importância desta questão para o plano específico que está sendo avaliado.

B Escolha de tecnologias

Na primeira fase, também foram escolhidas as tecnologias utilizadas para o desenvolvimento do sistema. A escolha limitou-se às possibilidades existentes e à diretriz de se procurar o software livre. Tecnologias mais práticas ao usuário final poderiam ser utilizadas, mas não havia disponibilidade ou suporte para as mesmas.

Na etapa de desenvolvimento, um dos pressupostos do sistema foi de que os dados, tanto os fornecidos pelos usuários, quanto os nativos do sistema, não estivessem armazenados no computador do usuário. Dessa forma, não se exige que o usuário precise instalar um banco de dados em seu computador e o desenvolvedor do sistema obtém informações sobre sua utilização e, com essas, pode incluir novos dados na base, que imediatamente se tornam disponíveis ao usuário².

A idéia inicial para o desenvolvimento do sistema de apoio à avaliação de planos de governo envolvia um servidor JSP (*Java Server Pages* - Servidor de Páginas Java). O servidor de páginas JSP é um programa que fica instalado em um servidor de internet e analisa requisições vindas de um navegador da internet (browser) utilizado pelo usuário, e devolve páginas HTML (*HyperText Markup Language* - linguagem de marcação de hipertexto). O usuário não toma

² Em uma aplicação final, no entanto, há de se considerar o fato de que avaliadores podem não se sentir confortáveis com o registro de informações sobre seus planos em servidores remotos. Uma solução é o termo de compromisso do detentor do servidor como fiel depositário ou, no campo tecnológico, o desenvolvimento de uma versão *stand alone* para o sistema de avaliação de planos.

conhecimento deste processo, e tudo se passa como se ele estivesse simplesmente acessando páginas HTML. Esta alternativa apresenta a vantagem da extrema facilidade para o usuário: ele acessa a página de avaliação de planos, preenche os dados solicitados, faz as opções desejadas e recebe respostas, tudo através de um browser, sem a necessidade de instalar programas.

Diante da indisponibilidade de um servidor JSP, partiu-se para uma segunda alternativa: desenvolver um programa em Java. Para que essa solução não seja muito trabalhosa para o usuário, deve-se prever um sistema de instalação que providencie todos os processos de implantação de um sistema *Java stand alone*. Uma terceira alternativa é utilizar Ferramentas IDE como Delphi, porém essa opção foi descartada já que boa parte da codificação do programa em Java poderá facilmente ser reutilizada caso se deseje, em oportunidade futura utilizar JSP. Outro motivo da escolha do Java foi o fato de que o programa Java pode ser utilizado em qualquer sistema operacional, tais como Linux, Windows, OS/2, Solaris. A plataforma Delphi para desenvolvimento de programas na linguagem Pascal, na verdade Object Pascal, apresenta a vantagem de ser mais rápido, mas esta característica não sobrepujou as apresentadas pelo Java.

Definida a plataforma de desenvolvimento como sendo a Linguagem Java, partiu-se para a definição de um banco de dados. A escolha recaiu sobre o MySQL. Os três motivos que determinaram essa decisão foram os seguintes: havia um servidor na UFSC que disponibilizava o Banco de dados MySQL; O MySQL é gratuito para fins não comerciais; e, é um banco de dados portátil não consumindo muitos recursos do servidor de dados (computador onde o banco de dados é instalado). Esse terceiro motivo era importante já que na fase de desenvolvimento e testes esse banco de dados seria instalado em um computador pessoal, que

possuía uma capacidade limitada de recursos, esse mesmo computador seria o servidor de dados e a plataforma de desenvolvimento do sistema. O banco de dados MySQL apresenta algumas limitações técnicas, devido a sua simplicidade, as quais dificulta de certa forma o desenvolvimento do sistema. Essas limitações podem ser contornadas, e não serão discutidas neste trabalho.

II Desenvolvimento do sistema

Desenvolver um programa em Java, que irá apresentar uma interface gráfica, envolve o pré-projeto e a programação desta interface. Além disso, é necessário digitar o código fonte, compilar e executar várias vezes em busca de erros. Existem algumas ferramentas (programas) que permitem ao usuário executar duas ou mais destas tarefas concomitantemente. Essas ferramentas são conhecidas como IDE (*Integrated Development Environment* - Ambiente Integrado de Desenvolvimento). Como exemplo de ferramentas IDE, encontramos o Borland JBuilder, NetBeans, Oracle, Jdeveloper, e, o Forte for Java. Como havia um CD, fornecido por uma revista, com uma distribuição para avaliação do Forte for Java, este foi o escolhido. Um Ambiente Integrado de Desenvolvimento é importante, pois aumenta sobremaneira a produtividade do programador. Entretanto, estes ambientes para desenvolvimento de programas Java apresentam a necessidade de grandes quantidades de memória física. Valores abaixo dos solicitados fazem com que a interface se torne lenta. Este fato ocorreu durante o desenvolvimento, pois o computador utilizado tinha uma quantidade de memória abaixo da recomendada. O ambiente de desenvolvimento estava em um computador operando acima de sua capacidade de trabalho, tornando o trabalho de codificação lento.

Existem diferentes abordagens para projetos de sistemas de informação. A programação estruturada e a programação orientada a objetos (POO) são as duas mais utilizadas. Enquanto o número de programas escritos em linguagem estruturada vem decrescendo, a programação orientada a objetos apresenta uma quantidade cada vez maior de adeptos. Como a linguagem de programação escolhida foi o Java, a abordagem adotada para o desenvolvimento do sistema é a programação orientada a objetos.

Segundo Silva (2001), é necessário uma abordagem metódica de desenvolvimento, manutenção e gerenciamento ao longo da existência de um software. Algumas metodologias foram criadas com o objetivo de darem suporte para a análise metódica das etapas envolvidas no desenvolvimento de um sistema usando programação orientada a objetos, podendo-se citar, como exemplos, a metodologia de Coad e Yourdon, a técnica de modelagem de objetos (*Object Modeling Technique* - OMT), a metodologia de Booch, a engenharia de software orientada a objetos (*Object-Oriented Software Engineering* - OOSE), e, a metodologia fusion.

A UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada) é uma notação resultante, até certo ponto, da unificação das metodologias de Booch, OMT e OOSE, e tende a ser adotada como padrão da especificação, visualização, e documentação de sistemas orientados a objetos (SILVA, 2001), e, segundo Mellor (1999), provê uma grande quantidade de informação sobre a aplicação e de seu projeto. Segundo Larman (2000, p. 42), a linguagem UML padroniza artefatos e notação, mas ela não define um processo padrão de desenvolvimento, a UML é apenas uma linguagem de notação. A modelagem UML foi escolhida como notação para projeto do presente sistema.

Silva (2001), propõe que o desenvolvimento de um sistema deve ocorrer dentro de um ciclo de vida, o qual deve ser previamente planejado. Os principais modelos de um ciclo de vida de um sistema são: modelo cascata (*waterfall*), modelo em V, modelo de desenvolvimento incremental, modelo espiral e o modelo de transformações. Para o desenvolvimento do sistema de avaliação de planos de governo será adotado o modelo de desenvolvimento incremental, que apresenta as seguintes etapas que são cíclicas: análise, projeto implementação e testes. Estas fases só terminam quando o trabalho se apresenta satisfatório.

Dessa forma ficam determinadas a linguagem de programação, a metodologia, as técnicas e as ferramentas para o desenvolvimento do sistema: linguagem java, orientação a objetos, banco de dados MySQL, IDE Forte for Java, notação UML e o modelo de desenvolvimento incremental.

6 Estrutura do Trabalho

Nesta seção será apresentada a estrutura do trabalho e o que cada uma das seções apresenta e qual a sua finalidade.

O capítulo 1, apresenta uma introdução ao assunto que se deseja desenvolver com a finalidade de familiarizar o leitor com o contexto da avaliação de planos de governo, mostrar quais são as questões de pesquisa, os objetivos que se deseja atingir, as justificativas e motivações que fundamentaram o esforço para a busca dos objetivos, a metodologia utilizada, uma breve descrição da estrutura do trabalho, e as delimitações do trabalho.

O capítulo 2 apresenta os resultados obtidos durante a pesquisa bibliográfica efetuada para o desenvolvimento do sistema. Inicialmente é apresentada uma

introdução à avaliação de planos de governo, definindo o significado de plano de governo, são estudadas as abordagens de avaliação de planos de governo, e os estágios em que os planos podem ser analisados. A metodologia de Baer é então apresentada, sendo o capítulo encerrado.

O capítulo 3 dá ênfase aos aspectos técnicos do desenvolvimento do sistema proposto. Uma introdução apresenta a filosofia do desenvolvimento, depois, no capítulo são apresentados, resumidamente, alguns aspectos sobre o planejamento e o desenvolvimento do sistema, as interfaces do sistema e como utiliza-lo.

O capítulo 4 apresenta a aplicação do sistema em quatro situações como forma de avaliar o funcionamento do sistema.

O capítulo 5 apresenta as conclusões do trabalho e as sugestões para trabalhos futuros.

O capítulo 7 apresenta a bibliografia e os programas utilizados para o desenvolvimento deste trabalho e do sistema proposto.

7 Delimitação do Trabalho

O trabalho pretende estudar o questionário de avaliação de planos de governo desenvolvido por Baer. Em segundo lugar propor um sistema que armazene este questionário e permita ao usuário responder às questões vinculando as respostas a um plano de governo específico. O sistema permitirá ao usuário cadastrar pesos para cada uma das questões e, com base nas respostas do questionário e dos pesos, emitir um relatório, em forma de planilha mostrando o

porcentual de respostas positivas. Em terceiro lugar, pretende-se aplicar o sistema em um plano de governo real e apresentar os resultados.

Portanto, o trabalho não pretende avaliar planos de governo e sim estabelecer um sistema de informação que, com base em configurações realizadas por especialistas em planos na área específica de governo a que o plano atende, permite a aplicação da Metodologia de Baer para avaliação de plano governamental. Nesse sentido, os exemplos discutidos no Capítulo 4 servem como ilustração da aplicabilidade prática do sistema desenvolvido e, assim, não dizem respeito à avaliação efetiva de qualidade real dos planos tomados como exemplo.

Outra delimitação significativa do trabalho está no fato de se ter optado por tomar como base para o sistema proposto uma única metodologia de avaliação de planos. Embora seja um dos trabalhos mais referenciados na área, é importante ressaltar que não há uma única forma de se avaliar planos de governo. Caso os avaliadores não estejam familiarizados com a metodologia de Baer, poderão utilizar o sistema para conhecer melhor seus pressupostos.

CAPÍTULO 2 - AVALIAÇÃO DE PLANOS

1 Introdução

O objetivo deste capítulo é mostrar uma síntese dos resultados obtidos durante a pesquisa bibliográfica efetuada para a execução deste trabalho.

Avaliar plano de governo é uma tarefa não muito clara. Baer mostra que muitos profissionais autores de projetos o fazem de maneira subjetiva e pessoal:

Como você saberia se um plano é bom caso você visse um? A resposta dos autores de projetos está perigosamente próxima a uma resposta de veracidade duvidosa a uma pergunta similar sobre a arte boa: 'eu não sei muito sobre a arte (planos) mas eu sei o que eu gosto'. Como uma profissão, nós desenvolvemos poucos guias. (BAER, 1997, p. 329).

Percebe-se nas palavras de Baer que as pessoas que avaliam planos de governos podem dizer se um plano é bom ou ruim, mas nem sempre conseguem esclarecer quais os critérios utilizados na avaliação.

A qualidade de um plano é difícil de definir. Os autores de planos podem frequentemente diferenciar planos da qualidade elevada dos de qualidade baixa, mas são pressionados duramente para definir explicitamente as características chaves da qualidade do plano. A literatura do planejamento é surpreendentemente limitada quando vem definir em que se constitui um bom plano. A profissão do planejamento geralmente evitou esta pergunta normativa e focalizou-a preferivelmente nos métodos e nos processos de fazer do plano (BERKE e FRENCH, 1994, apud BAER, 1997, p. 329).

Segundo Alexander e Faludi (1989, apud BAER 1997, p. 329), "se o planejamento precisa ter alguma credibilidade como uma disciplina ou uma

profissão, os critérios da avaliação devem permitir um julgamento real da eficácia do planejamento: o planejamento bom deve ser distinguível do ruim."

A avaliação de planos de governo faz-se necessária. Mas, se é difícil definir se um plano é ou não bom, muito mais difícil ainda torna-se criar critérios genéricos que possam avaliar qualquer plano. Como se verá adiante, é necessário restringir quais tipos e em quais momentos, um plano de governo pode ser avaliado de uma forma objetiva. A palavra objetiva aqui tem o significado oposto ao da palavra subjetiva, significando que o plano deva ser avaliado com critérios genéricos e previamente estabelecidos. A tentativa de Baer de contornar, ou solucionar esse problema que, no seu todo, e da forma como se apresenta, aparentemente não pode ser solucionado, é descrita nas seções que se seguem.

2 Definição de Plano de Governo

É necessário, antes de se iniciar este estudo, definir o que seja um plano de governo. De acordo com Bruton³ (1974, p. 7), sempre e em qualquer lugar que planejamento é discutido, é criada uma confusão para definir o significado preciso dos termos utilizados. Bruton tenta se servir do *Concise Oxford Dictionary* para explicar o que é plano, mas, segundo ele, sem muito sucesso. A definição encontrada sobre planejamento não contém a essência que explica em torno de quê o planejamento se situa. Segundo ele, a confusão está em não se diferenciar planejamento geral e planejamento físico. Bruton então define planejamento, diferenciando planejamento geral de planejamento físico:

³ Bruton foi chefe do Departamento de Planejamento Urbano do Instituto de Ciência e Tecnologia da Universidade de Wales.

[...] planejamento geral é visto como um procedimento por meio do qual alguém faz planos ou preparativos antecipadamente, e como uma atividade é um compromisso para uma maior ou menor extensão por indivíduos (planejando viagens, férias, gastos) e corporações (planejando orçamentos, políticas, etc.). Planejamento físico, por outro lado, é visto como uma referência a um desenho físico de alguma coisa que já existe ou pode vir a existir no futuro e esta forma (este tipo) de plano é uma representação, no sentido geográfico ou espacial, de estruturas físicas ou elementos atuais. Agora, torna-se aparente que a maioria dos planos gerais possui componentes físicos, caso ilustrativo um plano geral individual para uma excursão de férias pelo continente deve envolver o mapeamento das rotas que serão seguidas se as férias devem ser organizadas eficientemente [...]. (BRUTON, 1974, p. 7)

Considerando-se as afirmações de Bruton (1974), e uma definição da palavra plano no dicionário essencial da língua portuguesa (SACCONI, 2001), define-se plano de governo, neste trabalho, como sendo um planejamento geral que determina uma linha de conduta governamental, formal, precisa, traçada por etapas para atingir determinado objetivo. Mais especificamente, um plano de governo pode ser feito para se concretizar uma construção ou obra, um plano econômico ou ainda definir ações sociais, diretrizes e ações para a safra agrícola, educação, moradia, crescimento urbano dentre outros. Os planos de governo são desenvolvidos com a finalidade de atender demandas do poder público. Dentre os trabalhos sobre planos de governos encontrados, a maioria trata de planejamento urbano, no Brasil chamado de plano diretor da cidade. Os planos diretores visam estabelecer critérios e limites para o crescimento dos centros urbanos, nos Estados Unidos, alguns Estados, através de leis, determinam que seus municípios possuam cada um o seu planejamento urbano. Os critérios para avaliação de planos de governo, organizados por Baer, parecem ter sido desenvolvidos para avaliar estes planos urbanos, mas podem ser generalizados para outros tipos de planos (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

3 Principais Abordagens de Desenvolvimento de Planos

Baer (1997, p. 335) afirma que os critérios apropriados para avaliar um planejamento devem estar implícitos na própria concepção do planejamento, ou seja, os critérios dependem do próprio plano, ou mais especificamente, de sua categoria. Ainda segundo ele, cada vez que um plano é preparado, quem o prepara deve especificar os critérios pelos quais deseja que ele seja avaliado. Para criar um critério de avaliação genérica, Baer faz uma revisão sobre os tipos de planejamento para enfatizar quais critérios, de acordo com os tipos de planejamento, são apropriados para a avaliação. Os seja, os critérios de avaliação mudam de acordo com o estilo ou método utilizado para o desenvolvimento do planejamento. A seguir uma descrição resumida dos tipos de planejamentos, e, para alguns destes, uma descrição geral do que os critérios devem esperar:

- O planejamento enquanto visão (projeção)

Esta forma de planejamento, de acordo com Baer (1997), tenta conduzir o leitor a um pensamento dedutivo, sendo o leitor induzido, pelas afirmações e sugestões feitas no plano, maravilhar-se e concordar com o panorama projetado. Esse tipo de planejamento apela para os sentimentos. Baer não descreve o que devem esperar os critérios de avaliação de planos enquanto visão;

- O planejamento enquanto fotografia (*blueprint*)

Mostra elementos como num mapa municipal. Inicialmente creditava-se que um plano do tipo visão poderia se converter num plano fotográfico. Para este tipo de plano, os critérios de avaliação devem esperar que "cada aspecto do planejamento, bem como suas interconexões, sejam descritas, ou mapeadas, ou diagramadas detalhadamente" (BAER, 1997, p. 335);

- O planejamento como guia de uso territorial

Apareceu na década de 50, é uma espécie de visão do futuro, "uma proposição política, mas não um programa de ações; uma formulação de metas, mas não programas, prioridades ou estimativas de custo" (BAER, 1997, p. 336). Dividiu-se posteriormente em quatro ramificações principais (KAISER e GODSCHALK, 1995 apud in BAER, 1997, p. 335): planejamentos de utilização territorial; planejamento de classificação territorial; planejamento de política verbal; e planejamento de desenvolvimento e gerência;

- O planejamento como remédio

É uma forma de remediar o que não foi planejado. É uma correção de um erro que já aconteceu, como exemplo, política de planejamento para favelas (BAER, 1997, p. 336);

- O planejamento enquanto requerimento administrativo de fundos federais

Tratava-se de planejamento com a finalidade de atender critérios para obter recursos federais. "O planejamento social administrativo Cidades Modelos foi o cume da abrangência federal consolidada" (BAER, 1997, p. 336). Listas muito grandes de critérios foram criados para este tipo de planejamento;

- Processo, não planejamento

Esta abordagem colocava o planejamento em segundo plano. O importante era o estabelecimento de um processo contínuo do qual o planejamento era apenas um elemento. Altshuler (1965, apud in BAER, 1997, p. 337) sustentava que: "planejar é mais importante do que qualquer plano em si". Neste modelo, o

planejamento era apenas uma das partes do processo que era mais amplo. Algumas críticas, no entanto, contra este modelo, foram apresentadas:

Quais critérios podem ser estabelecidos para os processos de planejamento? Como estão preparados os planejamentos para sustentar as responsabilidades de suas recomendações e, como estão preparados os políticos para sustentar as responsabilidades de suas decisões, se todo planejamento é um processo contínuo? (BAER, 1997, p. 337)

- O planejamento enquanto ação pragmática

O tipo de planejamento, iniciado nos anos 70, que valorizava os aspectos práticos aos teóricos. Era baseada em critérios processuais e tomava menos cuidado com a preocupação do planejamento enquanto bússola do desenvolvimento (HAGMAN e JUERGERSMEYER, 1986, apud in BAER, 1997, p. 337);

- O planejamento como resposta a mandatos de planejamentos federais e estaduais

Surgiu com a observação de que efeitos ambientais eram fortes critérios de avaliação.

No nível federal, outras ordens, comandos, mandatos administrativos foram introduzidos e deram nova forma ao plano. Brooks (1979) descreveu o resultado como “modelo regulatório administrativo” refletindo requerimentos federais para planejamentos por água e ar limpos (Planejamentos de Implementação Estaduais [SIPs]). Os critérios chaves para estes planejamentos eram aqueles que mostravam como ele seria implementado e quem seria responsável pela implementação. (BAER, 1997, p. 338)

Observa-se que diversas são as abordagens que podem ser utilizadas para o desenvolvimento de planos de governo. Cada abordagem de desenvolvimento possui os seus próprios critérios de avaliação, como havia sido comentado anteriormente. Algumas abordagens podem apresentar critérios claros para a sua

avaliação. Esses critérios, ou a parte objetiva deles, poderiam ser sistematizados. Baer, no artigo referenciado, no entanto, não se aprofunda em nenhuma destas teorias. Entretanto, ele apresenta uma forma de avaliar planos que pode ser aplicada a maioria das abordagens de desenvolvimento de planos. Para apresentar este estudo, na seção seguinte, será mostrado que os planos de governo podem ser avaliados em diversos momentos de sua existência, e também, durante a sua concepção.

4 Estágios para Avaliação de um Plano de Governo

Segundo Baer, a avaliação de planos de governo pode ser feita em diferentes estágios do planejamento, como se pode observar na transcrição seguinte:

[...] os critérios apropriados para a avaliação de planejamento dependem de distinguirmos os diferentes estágios de planejamento quando a avaliação pode ser realizada. A rubrica de avaliação inclui: (1) avaliação de planejamento, (2) teste e avaliação de planejamento, (3) crítica de planejamento, (4) pesquisa comparada e avaliação profissional e avaliação post hoc⁴ dos resultados do planejamento. As vezes a mesma palavra tem sido usada para todas estas diferentes fases. Distingui-las é importante para identificar quem está encarregado da avaliação e sua relação com os autores do plano, para identificar quando a avaliação é empreendida (ex. em qual estágio, durante a preparação do planejamento ou após sua finalização) e, finalmente o “o quê” da avaliação. O “o quê” toma diversas formas:

(1) a essência das alternativas de planejamento, e/ou

(2) o planejamento como um pacote, incluindo o documento que o comunica:

[I] metas e objetivos

[II] necessidades e problemas

[III] hipóteses e métodos de raciocínio

[IV] propostas específicas e (talvez)

[V] estratégias de implementação (ordenação,

orçamento, etc)

(3) o resultado seguido da implementação do plano

Diversos tipos de avaliação de planejamento podem ser realizados e em diferentes estágios do processo de planejamento [...]

(BAER, 1997. p. 330).

⁴ Após a implantação do plano.

Uma descrição mais aprofundada dos tipos de avaliação de planejamento, de acordo com as definições de Baer, é apresentada a seguir:

- Crítica do planejamento

Foi o primeiro tipo de avaliação de planejamento. Era semelhante a uma resenha de um livro ou filme, a crítica era feita, por pessoas treinadas, após a publicação do planejamento mas, em geral, antes de ser posto em prática. Os critérios eram pessoais e empíricos enfatizando o planejamento como documentação. Elas são essencialmente informativas, raramente sistemáticas, muito menos analíticas mas possui a flexibilidade de formular uma crítica própria para a ocasião e introduzir novas preocupações.

- Teste e avaliação de planejamento

Tornou-se proeminente no início dos anos 70 como um acordo para o processo de avaliar caminhos alternativos e chegar ao plano desejado (ROBINSON 1972; LICHFIELD, KETTLE e WHITEBREAD 1975, apud in BAER, 1997, p. 332). Os planejadores adotavam as melhores alternativas após analisa-las e testa-las. A segunda parte do modelo de planejamento de Lichfield, Kettle e Whitebread sugerem o seguinte vocabulário e os seguintes passos para avaliação do processo de planejamento: testando planejamentos alternativos (fase 07); avaliação destes testes (fase 08); escolhendo o melhor (fase 09); implementando o plano (fase 10) e, na fase 11, comparando os resultados obtidos com os previstos e avaliando a importância de qualquer consequência não antecipada.

Observa-se que vários aspectos foram introduzidos. A avaliação é realizada pelo grupo que prepara o planejamento, não por críticos de fora. Os métodos são explícitos e reproduzíveis por outros e não idiossincráticos. Projetos analíticos, tais como análise de custo-benefício ou análise de realização de metas (HILL, 1968, apud in BAER, 1997, p. 332), são usados e não critérios implícitos dependentes das virtudes individuais dos críticos.

- Pesquisa comparada e avaliação profissional

Este tipo ocorre após o planejamento ter sido aprovado, porém antes que os resultados possam ser avaliados, ou quando estes resultados estão fora dos objetivos da avaliação. O avaliador é um pesquisador dentro da organização que prepara o planejamento. Em geral, vários planejamentos são comparados de forma sistêmica. Gruft e Gutstein (1972, apud in BAER, 1997, p. 333) para estabelecer seus critérios, determinaram que: (1) o planejamento deve ser um processo racional, (2) deve ser um processo democrático de decisão e (3) planejamentos (e relatórios sobre planejamentos) devem acrescentar algo em relação aos projetos urbanos.

- Avaliação post hoc

Os planos podem ser avaliados após serem aprovados e implementados, de forma empírica, em relação aos seus resultados. A razão é, geralmente, descobrir se o planejamento foi implementado e, se foi, como foi seu desempenho (ALTERMAN e HILL, 1978, apud in BAER, 1997, p. 333) ou qual foi a sua competência (FALUDI, 1987, apud in BAER, 1997, p. 333).

A descrição dos estágios de avaliação apresentados não tem a finalidade de ser um estudo aprofundado do assunto, mas de familiarizar o leitor com o tema,

mostrando as idéias e o contexto no qual está inserido. Se houver interesse em um maior aprofundamento, é sugerida a leitura do artigo de Baer.

A seção seguinte apresenta os critérios de avaliação de planejamento propostos por Baer e as objeções existentes ao uso de critérios.

5 Metodologia de Baer

A apresentação dos principais estágios para avaliação de um plano de governo, mostra que, com exceção do critério de teste e avaliação de planejamento, todos os demais ocorrem após o planejamento ter sido formulado. Baer apresenta uma outra proposta de avaliação de planos de governo que é feita enquanto o plano está sendo formulado. Para diferencia-lo do critério de teste e avaliação de planejamento, que avalia planos alternativos, Baer resolveu chamar esta fase da avaliação de análise do planejamento em processo (BAER, 1997, p. 338). A análise do planejamento em processo tem a finalidade de verificar "a habilidade profissional: o planejamento enquanto documentação de comunicação metodológica, raciocínio e conteúdo: ele opera no momento de por o planejamento em ação" (BAER, 1997, p. 338).

Um planejamento, enquanto documento, deve possuir atributos essenciais que garantam a sua qualidade. Os critérios tem a finalidade de verificar a existência desses atributos. Um exemplo de atributo desejável de um plano é que o mesmo não possua erros por omissão (BAER, 1997, p. 338-9). Se um critério exige que se explique o porquê das decisões de um plano, pode reduzir essa possibilidade. Um a forma de se tentar impedir a omissão seria definir os critérios para um mal plano e reprimi-los. Dentro dessa linha, segundo Baer, Hansen (1968, apud in BAER, 1997

p. 339) argumenta que a inviabilidade, estilo vagaroso, obsolência, pouca abrangência, falta de realidade ou negligência indicaria um plano ao qual se deve ter pouco respeito. Mas acredita que para estabelecer critérios deve-se tomar uma atitude mais positiva em vez de se fazer apenas advertências. Os critérios positivos, segundo ele, devem especificar o que os planejamentos devem ter. As afirmações expressas em um plano devem deixar evidências de que de fato elas são verdades (BAER, 1997).

Uma lista dos critérios, segundo Baer, é enfadonha, porém se torna importante no momento de sua utilidade, assim como uma lista de tarefas domésticas, compara. Os critérios são muito importantes para quem está preparando ou avaliando um planejamento. Os critérios, segundo ele não são definitivos, são possíveis considerações que podem ser utilizadas durante a preparação do planejamento. Os autores do planejamento podem projetar variantes para os critérios, e, também podem fazer acréscimos (BAER, 1997, p. 339).

Baer apresenta algumas das possíveis objeções ao uso dos critérios, que classificou em modernistas e pós-modernistas.

- Objeções modernistas

A primeira objeção modernista diz que os critérios que serão apresentados devem ser utilizados por profissionais do planejamento, dificultando a sua utilização por outras pessoas ou grupos. Como solução, Baer sugere a criação de outras listas de critérios, porém formuladas por pessoas ou grupos que apresentem interesse e que não sejam profissionais do planejamento.

A segunda objeção diz que "clareza e um planejamento bem articulado e documentado de acordo com os critérios pode, ainda assim, ter outro efeito no mundo real (DROR 1968, apud in BAER, 1997, p. 340)". Segundo Baer (1997), "esta crítica reflete a visão de que 'as coisas não são como parecem' e diz 'o que realmente está acontecendo é...'"'. Segundo ele, esta objeção "evoca tramas e intrigas, ou deplora a mistificação empregada pelo planejamento para disfarçar o que realmente transpirava" (BAER, 1997, p. 340). Ou seja, um planejador pode fazer um plano com intenções diferentes da apresentada pelo documento, e ainda assim este plano estar de acordo com os critérios para a confecção de um planejamento. De acordo com Baer, essa prática cai por terra em pouco tempo, e o que estava escondido acaba por se revelar e mostra o "quão insidioso e malicioso o plano é".

A terceira objeção diz respeito à utilização dos critérios, quando se tornam instrumentos legais, com finalidades de grupos interessados para intervenções. Outro aspecto é que o plano pode conter omissões planejadas ou falhas técnicas com finalidades políticas. Segundo Baer (1997), um observador de fora não consegue fazer esse discernimento.

- Objecções pós-modernistas

Uma das objeções modernistas diz que os planejamentos não devem ter uma aparência científica, já que são criações humanas (KRIEGER, 1981 apud in BAER, 1997, p. 341). Outra objeção, mais contundente que a primeira, diz que a perícia tecnocrática deve ser desmistificada e que os planejamentos devem ser vistos como parte de um discurso, um argumento, e que uma orientação por habilidades retóricas (persuadindo com o discurso) deve ser, pelo menos, tão importante quanto a técnica.

A despeito das objeções, modernistas e pós-modernistas, descritas, Baer desenvolveu uma lista de critérios expressos em forma de questionário com sessenta questões. As questões são divididas em oito classificações. As classificações agrupam as questões em torno de um tópico ou assunto.

A seguir são transcritos os tópicos, numerados de 1 a 8, e, dentro destes, as questões propostas por Baer:

1. **Adequação ao Contexto. (Explica o contexto e a colocação: o para quê e o porque da documentação. Não é evidente por si só ao público).**
 - A. O contexto político/legal do planejamento é explicado (exemplo: encontros, ordens estaduais, discussões públicas, questões de alta prioridade)?
 - B. A autoridade administrativa para preparação foi indicada (Conselho de estado ou Comissão do Planejamento, lei de resolução estadual, requerimentos federais, etc)?
 - C. O papel da firma ou agência indicada para preparação foi adequadamente explicado (uma carta de transmissão)?
 - D. São apresentadas informações de suporte (razões para apresentação do planejamento)?
 - E. Está claro para quem o planejamento se destina (cidadãos, Conselho Municipal, junta)?
 - F. Os propósitos do planejamento são explicados (estudo, informação, decisão, transporte de informação)?
 - G. O tipo de planejamento e sua extensão são relatados no início para alertar o leitor sobre o que esperar (se é altamente quantitativo ou analítico, se é de longo alcance ou específico, ou técnico)?
 - H. São fornecidos uma visão panorâmica e um sumário (por exemplo, um “sumário executivo”)?
 - I. A origem dos fundos para planejamento é mostrada (federal, estadual, local, privada)?
 - J. É mostrado o total de tempo para preparação?

2. **Considerações do “Modelo Racional”. (Mostra considerações básicas sobre o planejamento baseada nas teorias destacadas e seu critérios. Há várias teorias e tipos de planejamento, e muito mais poderia ser anexado aqui. Os autores do planejamento devem ter claro o que estão fazendo, para transmitir clareza ao leitor).**
 - A. Dado o tipo de planejamento a ser preparado, as formulações do planejamento são claras sobre os critérios que irão usar para avaliar seu progresso enquanto estão sendo formulados?
 - B. Estes critérios estão explícitos no planejamento?
 - C. Os problemas são especificamente identificados (ou apenas supostos)?
 - D. Metas e objetivos são explicitamente identificados?
 - E. O tom do planejamento é adaptado a abordagem de planejamento recomendada (por exemplo, abrangência, desenvolvimento, justificação)?

- 1) Se a intenção do planejamento é ser abrangente, ele a relata substancialmente (relações horizontais em outras agências e corporações governamentais adjacentes)?
 - 2) O planejamento leva em conta o governo regional ou o governo superior (relações verticais)?
 - 3) Existe planejamento para coordenação continuada com outro planejamento e agências?
 - F. É descrita a capacidade ou adequação de infraestrutura existente e sistemas organizacionais?
 - G. São listadas alternativas, ou pelo menos consideradas?
 - H. As alternativas são identificadas como variações de um tema ou como sendo radicalmente diferentes?
 - I. São permitidos acordos livres?
3. Validade de Conduta (Regulamento). (Explica o “quem” e “como” do planejamento; informa o leitor como foi o planejamento e o que vai acontecer com sua divulgação)
- A. Quem está envolvido na formulação do planejamento (equipes de diferentes agências ou departamentos, grupos de cidadãos, políticos)?
 - B. Como foram selecionados (com base na perícia, interesse, voluntariado ou auto-seleção)?
 - C. Como foram envolvidos (discussões em grupo, memorandos e documentos para equipes internas, encontros, reuniões públicas)?
 - D. Como foram usados dados, modelos e metas e outras informações pertinentes na orientação de políticas ou ações?
 - E. Como questões técnicas foram transformadas em recomendações políticas (através do conhecimento comum, experiência, treinamento científico, treinamento em projetos)?
 - F. Foi utilizado algum grupo consultivo?
 - G. Preliminarmente, foi posto em circulação algum esboço para que o público fosse ouvido?
4. Adequação de Extensão. (Mostra como o planejamento conecta-se com o mundo, num sentido amplo)
- A. Todas as questões pertinentes possíveis foram consideradas (físicas, sociais, econômicas, políticas, psicológicas, culturais)?
 - B. Foram consideradas questões como eficiência, retidão e capacidade de prognósticos?
 - C. Foi considerada distribuição de custos e benefícios entre diferentes grupos e partes?
 - D. Foram consideradas implicações oriundas de mudanças/relocações?
 - E. Foram consideradas implicações fiscais/financeiras?
 - F. Foram consideradas implicações legais?
 - G. Foram consideradas as possibilidades no amplo contexto político?
5. Orientação para Implementação. (A maioria dos planejamentos tem a intenção de realizar algo. Consideremos as ferramentas [leis, regulamentações, orçamentos, programas, etc] e as agências e pessoas responsáveis pelo funcionamento do planejamento. Eles devem ser incluídos? [um plano de projeção não teria um aspecto de implementação, ao invés, teria uma seção lidando com os próximos passos].)
- A. Os recursos para implementação estão de acordo com o planejamento?
 - B. Existem prioridades para implementação?

- C. O custo da implementação é considerado em relação a sua não implementação?
- D. Há um momento reservado para a implantação do planejamento?
- E. Existe provisão de programação e coordenação das propostas de implementação?
- F. As propostas podem realizar suas intenções se implementadas?
- G. Existe uma proposta ou programa para análise de impacto?
- H. A agência ou pessoa responsável pela implantação está identificada?
- I. Pode se esperar que a agência responsável realisticamente implemente o planejamento?
6. **Abordagem, Dados e Metodologia.** (Esclarece as bases técnicas, se há alguma, do planejamento; de onde vem os dados e como são usados, para que outros possam acessar a linha de pensamento do planejamento, acessando as mesmas fontes)
- A. O plano foi baseado em um amplo espectro de dados, onde possível?
- B. O planejamento é suficientemente flexível para incluir novos dados e achados?
- C. As fontes de dados são citadas?
- D. As fontes em relação a metodologia são citadas?
- E. Os níveis de agregação de dados são relevantes ou significativos ao estudo?
7. **Qualidade de Comunicação.** (Uma clara comunicação é necessária para ouvirmos todos com justiça)
- A. O cliente ou público é identificado (público em geral, outros profissionais)?
- B. As idéias são apresentadas de forma convincente, considerando-se a natureza da platéia?
- C. As razões por trás das decisões são efetivamente apresentadas?
- D. As propostas/recomendações/conclusões são compatíveis com os objetivos?
- E. O tom da documentação é compatível com a mensagem transmitida (por exemplo, não apresentado no tempo passado como um fato consumado quando o planejamento é para estudo e revisão)?
- F. Estão indicados os critérios pelos quais se espera que o planejamento seja julgado?
8. **Formatos de Planejamento.** (Outras formas de comunicação são encontradas no formato do planejamento em si, bem como evidências de quem assume a responsabilidade profissional pela formulação do planejamento, quando foi adotado e outras preocupações incidentais semelhantes, que, no entanto, comunicam sobre competência profissional).
- A. O formato e tamanho contribuem para o uso pretendido (exemplo, um planejamento muito grande é difícil de arquivar e copiar, desde que não seja usado como referência constante e uso diário)?
- B. A data de publicação é mostrada?
- C. O nome dos autores é mostrado para indicar responsabilidade profissional (nomes dos funcionários que trabalharam no planejamento, bem como agências ou nomes de firmas)?
- D. Existe um índice?
- E. Existem páginas numeradas?
- F. Os gráficos são bem utilizados?
- G. O planejamento está exposto de forma atraente?
(BAER, 1997, p. 343-6).

6 Considerações Finais

Desenvolver um plano de governo não é simples. Esta tarefa apresenta uma grande quantidade de elementos subjetivos, interesses de diversos setores da sociedade, algumas vezes divergentes, e depende de fatores externos ao planejamento em si, que nem sempre são controláveis. O trabalho de avaliar um plano de governo, portanto, não é uma tarefa fácil. Existem muitas abordagens de desenvolvimento de planos como foram descritas: o planejamento enquanto visão (projeção); o planejamento enquanto fotografia (*blueprint*); o planejamento como guia de uso territorial; o planejamento como remédio; o planejamento enquanto requerimento administrativo de fundos federais; o planejamento enquanto ação pragmática, e, também a abordagem que assume o planejamento como sendo apenas parte de um processo. Para cada abordagem de desenvolvimento cabem critérios diferenciados para avaliar o plano (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Além da existência das diferentes abordagens para o seu desenvolvimento, os planos podem ser avaliados em diferentes estágios de sua existência, conforme resumo na seção 4 deste capítulo. Baer sugere a avaliação do documento que apresenta o plano de governo. A avaliação seria a resposta de um questionário com sessenta questões sobre o documento e seu conteúdo (BAER, 1997). Esta avaliação pode ser automatizada através de um Sistema de Informação cujo desenvolvimento é apresentado no capítulo a seguir.

CAPÍTULO 3 - SISTEMA PROPOSTO

1 Introdução

De acordo com Larman (2000), para se desenvolver um Sistema de informação, faz-se necessário, primeiro, que o problema seja descrito, deixando claro o que é este problema, e, em segundo lugar, definir os requisitos, ou seja, o que o sistema deve fazer, para, posteriormente, desenvolver o sistema, através da codificação, em uma ou mais linguagens de programação. Estas atividades podem ser divididas em três passos: planejar e elaborar, que, juntos, constituem o primeiro passo, construir, que é o segundo passo e instalar, o terceiro passo.

A fase planejar e elaborar se constitui em um passo preliminar que pode incluir a concepção inicial, a investigação das alternativas possíveis, o planejamento e a especificação de requisitos, dentre outras atividades. Vários artefatos podem ser utilizados, dentre eles a especificação de requisitos, os casos de uso, os diagramas de caso de uso, o glossário e o rascunho do modelo conceitual. (LARMAN, 2000, p. 45).

A fase desenvolver pode ser feita através de vários ciclos de desenvolvimento, nos quais o sistema é estendido. (LARMAN, 2000, p. 46). Cada ciclo de desenvolvimento possui várias etapas que se repetem. Na fase desenvolver, são refinados os artefatos criados na fase anterior, é feita a análise, é feito o projeto, inicia-se a construção do sistema e é feito o seu teste. Depois de terminado um ciclo, ele reinicia até que o sistema esteja completamente desenvolvido. Durante a etapa de análise podem ser utilizados os vários artefatos, dentre eles os diagramas de seqüência do sistema, os contratos de operação, os diagramas de estado, bem

como os refinamentos dos artefatos existentes, como o modelo conceitual. Na etapa de projeto, outros artefatos podem também ser utilizados, dentre eles, os diagramas de interação, os diagramas de classes de projeto e o esquema de Banco de Dados. A etapa construir é a codificação do projeto, e, na etapa de testes o cumprimento dos requisitos é verificado (LARMAN, 2000).

A instalação é o momento em que o sistema é posto a funcionar (LARMAN, 2000).

Não existe uma fronteira clara separando as etapas de análise e projeto dentro da fase de desenvolvimento do sistema. Inicialmente, quando é dada ênfase na investigação do problema buscando a definição de uma solução está em andamento a etapa chamada análise. A segunda etapa é chamada de projeto, quando se procura uma solução lógica de como o sistema atenderá aos requisitos. A outra etapa, dentro da fase de desenvolvimento, é a construção propriamente dita, quando os componentes, que foram projetados, são implementados em uma linguagem de programação (LARMAN, 2000, p. 29-30). Como foi comentado, não há uma fronteira clara entre as etapas analisar e projetar da fase de desenvolvimento, apenas uma ênfase maior é dada a um dos aspectos, sendo possível a presença de elementos de uma fase em outra. Reforçando esta afirmação, Larman (2000, p. 38) propõe que "o trabalho de análise e/ou projeto existe num contínuo [...]. Portanto, não é de grande valia ser rígido ao que se constitui um passo de análise por oposição a um passo de projeto". Entretanto, segundo ele, é vantajoso separar o que é a pesquisa do problema das formas de criar uma solução para o problema. Além disso, Larman (2000 p.43) considera que o ciclo de vida do desenvolvimento deve ser iterativo, baseando-se no aumento e no refinamento sucessivo de um sistema através de múltiplos ciclos de análise, projeto,

implementação e teste. Entretanto, essa seqüência cíclica é demasiadamente longa de descrever em um documento e deixa de ser interessante ao leitor. Dessa forma, o planejamento e elaboração, e o desenvolvimento, contendo a análise, o projeto, e a construção propriamente dita do sistema, serão estudados em duas seções distintas, para o seu entendimento, respeitando as restrições impostas pela dificuldade de separar análise de projeto. A seqüência de ciclos iterativos será abstraída, de forma que será mostrado ao leitor o resultado final de cada artefato.

2 Planejamento e Elaboração

No o planejamento e elaboração do problema é necessário definir os conceitos do problema, os requisitos do sistema e descrever os seus casos de uso. Estes tópicos são apresentados de forma linear, mas isto não significa que o desenvolvimento dos mesmos siga uma ordem determinada, eles podem ser desenvolvidos em qualquer ordem, de forma alternada, ou mesmo paralelamente.

I Conceitos do problema

Uma quantidade significativa de informação é necessária para explicar completamente a análise e o projeto orientados a objetos. A essência da análise é a ênfase na descoberta e na descrição dos conceitos do problema (LARMAN, 2000). Com a finalidade de descobrir e descrever os conceitos do problema, uma breve revisão da metodologia de Baer será feita neste momento.

A metodologia de Baer visa avaliar planos de governo, enquanto documento, na fase em que este plano está sendo formulado, o que ele chamou de análise do planejamento em processo. Um planejamento, enquanto documento, deve possuir

atributos essenciais que garantam a sua qualidade (BAER, 1997, p. 338). A análise tem a finalidade de verificar a existência destes atributos, tais como "a habilidade profissional: o planejamento enquanto documentação de comunicação metodológica, raciocínio e conteúdo..." (BAER, 1997, p. 338), dentre outros. Para isto, Baer apresentou uma lista de sessenta critérios que devem especificar o que os planejamentos devem ter. Os autores do planejamento podem projetar variantes para os critérios, e, também podem fazer acréscimos (BAER, 1997, p. 339). Os critérios de Baer são agrupados em oito classificações básicas (BAER, 1997, p. 338).

Baer apresentou no seu artigo, uma crítica a ambigüidade do uso das palavras por diferentes profissionais. Esta ambigüidade traria dificuldades para a análise do problema, para definição dos conceitos, com vistas a sua automação. Para o desenvolvimento do sistema, é necessário definir os conceitos para cada elemento ou ação proposta pela teoria de Baer. Larman (2000, p. 139), acredita que a consistência do significado e a compreensão compartilhada dos termos são muito importantes no desenvolvimento de uma aplicação. Segundo ele, os significados dos termos devem estar em um glossário. Este glossário define todos os termos que requerem algum tipo de esclarecimento. Dessa forma, foi criada uma relação para definir os conceitos que serão utilizados no projeto do sistema. Não há a certeza de que os conceitos utilizados aqui sejam os mais precisos ou indicados. Pode-se dizer que eles são provisórios e podem ser modificados com o amadurecimento das idéias propostas por Baer. Os conceitos do domínio do problema, para este sistema, foram assim definidos:

- Usuário: pessoa que utiliza o sistema. Pode ser um desenvolvedor de planos de governo ou outra pessoa que tenha a intenção de utilizar o

sistema para o suporte à avaliação de planos de governo ou outra atividade que o sistema disponibilize;

- Teoria: são as idéias como as propostas no trabalho de Baer. Não haverá referências físicas a elas, apenas referências abstratas, as idéias de um autor receberá um nome e esse nome será a referência abstrata para as suas idéias. Quando houver necessidade de identificar quais idéias se está referenciando, o nome da teoria será utilizado. A teoria em si, como um documento, não será armazenada no sistema;
- Método: a expressão da teoria expressa de forma objetiva, é na verdade um questionário. O método será concreto e de fato existirá no sistema através de um questionário que sintetiza as idéias da teoria. Um autor poderá propor, por exemplo uma teoria para avaliar dois tipos de planos de governo, havendo, nesta teoria, uma pequena diferença nas questões apresentadas de acordo com a área para qual é especificado o plano. No sistema, então, serão cadastrados uma teoria e dois métodos desta teoria. Cada método, possuirá o seu conjunto de questões próprios, que serão os questionários (cada método, um questionário);
- Questionário: o conjunto de todas as questões que pertencem a um método. Um questionário é um outro nome para o método. Se analisarmos; de cima para baixo, evidentemente que o questionário é um método da teoria, mas se analisarmos de baixo para cima, veremos que a o método agrupa questões classificadas em tópicos, ou seja, um questionário. Dessa forma, os dois termos serão utilizados de acordo com o aspecto que se deseja evidenciar;

- Tópico: um grupo de questões. Como as questões podem ser separadas em classes, a cada uma destas classes deu-se o nome de tópico. Quando o questionário não possui suas questões classificadas em tópicos, ele terá um tópico único, ao qual estarão associadas todas as questões. Um tópico sempre pertence ou está associado a um questionário (método);
- Questão: é de fato uma pergunta de um determinado método que pertence a um tópico. As questões estão sempre associadas a um tópico e, através deste, a um questionário;
- Critério: é uma tabela de pesos que são associados a um método. Vários critérios podem ser associados a um mesmo método. Os critérios associam a cada questão e a cada tópico um peso que faz com que o avaliador, de acordo com a finalidade e o tipo de seu plano, possa estabelecer quais questões e quais tópicos são mais importantes na avaliação. Os critérios são criados pelos usuários do sistema, mas o sistema apresenta critérios com todos os pesos unitários para facilitar o trabalho do usuário que não precisa cadastrar pesos quando estes forem todos unitários;
- Plano de governo: assim como a teoria, o plano de governo no sistema servirá apenas para identificar um plano. Os planos de governo que serão avaliados não serão armazenados no sistema. A finalidade de se cadastrar um plano de governo no sistema é para que o usuário possa identificar a qual plano de governo ele está se referindo. Os planos de governo são cadastrados pelos usuários e não estão associados a

nenhuma teoria, método, tópico, questão ou critério, mas, se for avaliado, será nesse momento associado a essa avaliação. Um plano pode ser avaliado por mais de um método se houver mais de um método disponível no sistema e caso o usuário deseje isto;

- Avaliação: ocorre quando é respondido um questionário de avaliação para avaliar um plano de governo. Após ter-se respondido, total ou parcialmente, ao questionário, é criada uma avaliação que serve de vínculo entre o plano de governo e o método;
- Respostas: são as respostas às questões de um método que estarão associadas, através da avaliação, ao método e ao plano de governo. As respostas são dadas pelo usuário e armazenadas pelo sistema;
- Relatório: documento, em forma de texto, gerado pelo sistema. Um relatório é feito para um plano de governo, avaliado através de um método por um usuário. Um mesmo relatório pode ser fornecido levando em conta vários critérios, nesse caso, para cada critério, haverá colunas com as informações para esse critério. Na confecção de um relatório, o sistema cruza a avaliação de um plano específico com cada um dos critérios escolhidos, fornecendo os resultados de acordo com os pesos definidos no critério.

Para que os significados e relacionamentos dos conceitos que serão utilizados fiquem ainda mais claros, é apresentado na figura 1 o mapa cognitivo dos conceitos do domínio do problema.

Uma outra atividade importante na fase de análise do problema é a especificação dos requisitos do sistema.

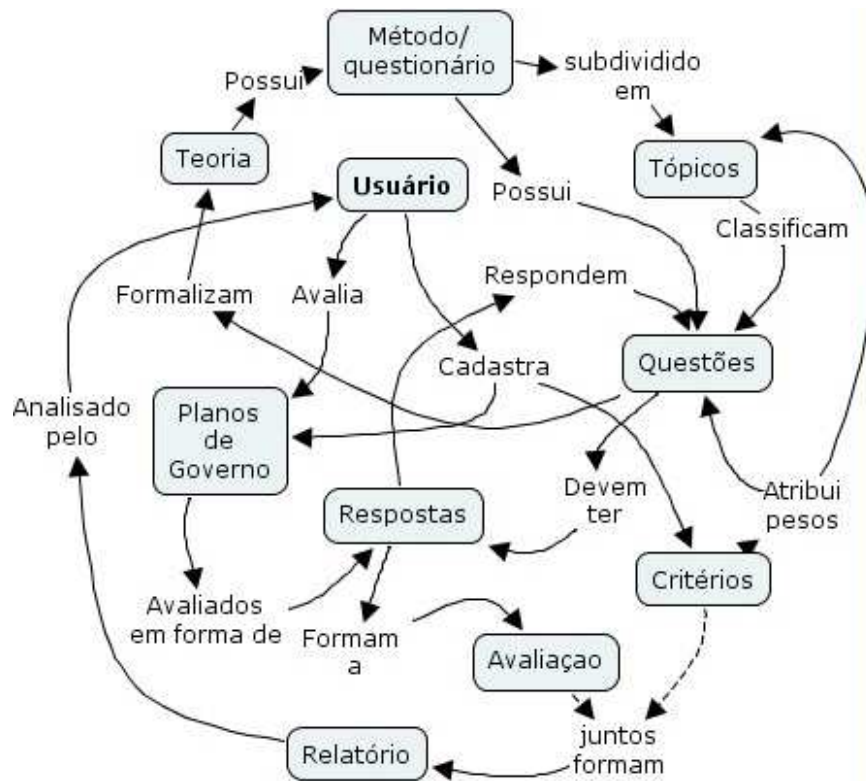


Figura 1 - Mapa conceitual do problema

II Especificação de requisitos

Segundo Larman, a especificação de requisitos correta e abrangentes são a essência de um projeto de sucesso. Os requisitos descrevem as necessidades ou os desejos para um produto. Larman recomenda cinco artefatos na fase de requisitos: um texto com a visão geral, clientes, objetivos, funções do sistema e atributos do sistema (LARMAN, 2000, p. 59).

- Visão geral do sistema: o sistema de apoio à avaliação de planos de governo deve automatizar, no que couber, a avaliação de planos de governo. Dessa forma, o sistema deve dar suporte para que o usuário

possa responder questões de avaliação, atribuir pesos para essas questões e receber, como resultado da interação do questionário e dos pesos, um relatório preliminar de avaliação. Como a metodologia desenvolvida por Baer classifica as questões, essas devem ser apresentadas ao usuário dentro de sua classificação e juntamente a esta.

- Clientes: desenvolvedores de planos de governo ou outra pessoa que deseje avaliar um plano de governo. São os usuários.
- Objetivos: automatizar, no que couber, a aplicação do questionário de avaliação de planos de governo da metodologia de Baer, ou outras equivalentes, para dar apoio à avaliação de planos de governo;
- Funções do sistema: o sistema deve: armazenar nomes de teorias de avaliação de planos de governo; armazenar metodologias de avaliação de planos de governo vinculados a uma teoria; armazenar tópicos destas metodologias de avaliação; armazenar questões destas metodologias de avaliação classificadas em tópicos; controlar qual usuário está utilizando o sistema; permitir a qualquer pessoa cadastrar a si como usuário; permitir ao usuário conectar-se ao sistema para utilizar as funções; permitir ao usuário cadastrar planos de governo; permitir ao usuário cadastrar critérios com pesos para os tópicos e as questões de uma dada metodologia; permitir ao usuário avaliar planos de governo respondendo às questões apresentadas pelo sistema (avaliação); permitir ao usuário armazenar a avaliação; elaborar para o usuário um

relatório onde os critérios são aplicados aos planos de governo; efetuar outras funções acessórias que tornem as funções descritas possíveis;

- Atributos do sistema: facilidade de uso; interface gráfica; independente de plataforma de software/hardware; utilização gratuita.

III Casos de uso essenciais e de alto nível

Segundo Larman, para uma melhor compreensão dos requisitos, a descrição narrativa dos processos do domínio, chamada de casos de uso, se mostra uma técnica excelente (2000, p. 66). Um caso de uso é um documento que descreve a seqüência ações de um ator, que é um agente externo utilizando um sistema para a execução de um processo (JACOBSON, 1992, apud in LARMAN 2000, p. 68). Como uma descrição de todos os casos de uso pode se tornar muito longa, somente os casos de uso de alto nível e essenciais à compreensão do domínio serão abordados aqui. Os casos de uso consideram o sistema como sendo a fronteira dos casos.

Tabela 1 - Caso de uso iniciar uso

Caso de uso:	Iniciar uso
Ator:	Usuário
Descrição:	Um usuário, inicia o uso do sistema. Após iniciar uso o sistema estará apto a operar.

Tabela 2 - Caso de uso cadastrar usuário

Caso de uso:	Cadastrar usuário
Ator:	Usuário
Descrição:	Um usuário, que não é cadastrado deseja se cadastrar. Durante o cadastro o usuário pode solicitar que ao término deste ele seja conectado. Após utilizar a função cadastrar usuário este usuário estará cadastrado no sistema e, caso solicite, estará também conectado ao sistema.

Tabela 3 - Caso de uso conectar usuário

Caso de uso:	Conectar usuário
Ator:	Usuário
Descrição:	Um usuário, que é cadastrado deseja se conectar. A conexão é necessária para utilizar funções do sistema que não seja cadastrar usuário e alterar senha de usuário. O usuário pode se conectar ao sistema no momento em que se cadastra ou altera a sua senha. Após utilizar a função conectar usuário este usuário estará conectado ao sistema.

Tabela 4 - Caso de uso alterar senha de usuário

Caso de uso:	Alterar senha de usuário
Ator:	Usuário
Descrição:	Um usuário, que é cadastrado, estando ou não conectado deseja alterar sua senha. Durante a alteração sua senha o usuário que não está conectado pode solicitar que ao término desta ele seja conectado. Após utilizar a função alterar senha de usuário este usuário terá sua senha alterada no sistema e, caso solicite, estará também conectado ao sistema. Se o usuário estiver previamente conectado ao sistema, a conexão permanecerá.

Tabela 5 - Caso de uso desconectar usuário

Caso de uso:	Desconectar usuário
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja se desconectar. Após utilizar a função desconectar usuário este usuário estará desconectado do sistema.

Tabela 6 - Caso de uso cadastrar plano de governo

Caso de uso:	Cadastrar plano de governo
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja cadastrar um plano de governo. Após utilizar a função cadastrar plano de governo o usuário terá cadastrado um novo plano de governo no sistema.

Tabela 7 - Caso de uso avaliar plano de governo

Caso de uso:	Avaliar plano de governo
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja avaliar um plano de governo cadastrado por ele, e, que já foi escolhido. Após utilizar a função avaliar plano de governo o usuário terá avaliado um plano de governo que ele cadastrou no sistema.

Tabela 8 - Caso de uso cadastrar critério de avaliação de plano de governo

Caso de uso:	Cadastrar critério de avaliação de plano de governo
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja cadastrar um critério de avaliação de plano de governo. Após utilizar a função cadastrar critério de avaliação de plano de governo, o usuário terá cadastrado um critério de avaliação de plano de governo no sistema.

Tabela 9 - Caso de uso definir pesos dos tópicos em um critério de avaliação

Caso de uso:	Definir pesos dos tópicos em um critério de avaliação
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja definir pesos dos tópicos em um critério de avaliação de plano de governo cadastrado por ele, e, que foi escolhido. Após utilizar a função definir pesos dos tópicos em um critério de avaliação de plano de governo o usuário terá definido os pesos dos tópicos em um critério de avaliação de plano de governo que ele cadastrou no sistema.

Tabela 10 - Caso de uso definir pesos das questões em um critério de avaliação

Caso de uso:	Definir pesos das questões em um critério de avaliação
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja definir pesos das questões em um critério de avaliação de plano de governo cadastrado por ele, e, que foi escolhido. Após utilizar a função definir pesos das questões em um critério de avaliação de plano de governo o usuário terá definido os pesos das questões em um critério de avaliação de plano de governo que ele cadastrou no sistema.

Tabela 11 - Caso de uso exibir resultados da avaliação de planos de governo

Caso de uso:	Exibir resultados da avaliação de planos de governo
Ator:	Usuário
Descrição:	Um usuário, que está conectado deseja exibir resultados da avaliação de planos de governo que foi configurado. Após utilizar a função exibir resultados da avaliação de planos de governo, o sistema exibirá os resultados da avaliação de planos de governo.

Tabela 12 - Caso de uso finalizar uso

Caso de uso:	Finalizar uso
Ator:	Usuário
Descrição:	Um usuário, finaliza o uso do sistema. Após finalizar uso o sistema será encerrado.

A notação UML padroniza a representação dos casos de uso de maneira gráfica. A figura 2 apresenta o diagrama de casos de uso do sistema de acordo com a notação UML.

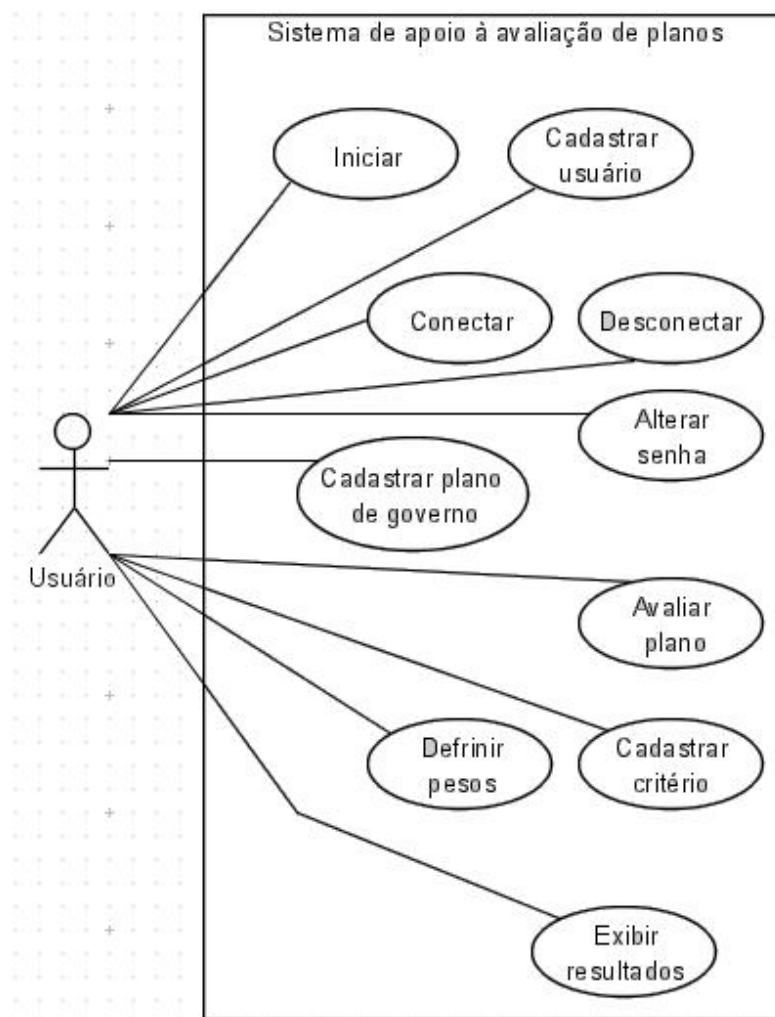


Figura 2 - Diagrama de casos de uso do sistema em UML

Tendo sido apresentados os conceitos do problema, a especificação de requisitos e os casos de uso essenciais e de alto nível, a próxima atividade será o desenvolvimento do sistema.

3 Desenvolvimento

O desenvolvimento do sistema será feito em duas fases, conforme descrito na introdução do presente capítulo: análise e o projeto. É necessário que durante o desenvolvimento do sistema seja definido qual será a sua solução. Para comunicar a solução proposta, a linguagem UML propõe alguns artefatos. Os artefatos propostos pela UML que serão utilizados são os diagramas de seqüência do sistema, os contratos de comportamento do sistema, os diagramas de colaboração (LARMAN, 2000 p. 33), o diagrama de classes de projeto (LARMAN, 2000 p. 35) e o esquema de banco de dados. Estes tópicos são apresentados de forma linear, mas isto não significa que o desenvolvimento dos mesmos siga uma ordem determinada, eles podem ser desenvolvidos em qualquer ordem, de forma alternada, ou mesmo paralelamente.

I Análise

Para Larman (2000), antes de se proceder a um projeto lógico do funcionamento de um sistema, é necessário investigar e definir o seu comportamento como uma "caixa-preta". "O comportamento do sistema é uma descrição do que o sistema faz, sem explicar como ele o faz" (LARMAN, 2000, p. 143). Os artefatos, definidos na Linguagem de Modelagem Unificada (UML) que podem ser úteis para definição deste comportamento e que serão apresentados nesta seção são os diagramas de seqüência do sistema, os diagramas de estados e os contratos do comportamento do sistema. A apresentação dos três artefatos será resumida, já que uma apresentação completa fugiria ao escopo do trabalho. Após a apresentação destes três artefatos será analisado o projeto do sistema.

A Diagramas de seqüência do sistema

Como propõe Larman (2000, p. 143), os casos de uso sugerem como os atores devem interagir com o sistema que está sendo criado, para obter os efeitos esperados. O ator, segundo ele, deve gerar os eventos reconhecidos pelo sistema esperando alguma operação do sistema como resposta. Ainda segundo ele, é importante isolar e ilustrar cada uma das operações que um ator solicita ao sistema para se compreender o comportamento do sistema. Larman (2000) explica que o UML inclui diagramas de seqüência para ilustrar as operações que são iniciadas por atores. O diagrama consiste de uma figura que mostra um cenário específico de um caso de uso. O diagrama mostra os eventos gerados pelos atores e sua ordem. O diagrama tem a finalidade de mostrar os eventos que rompem a fronteira entre o sistema, os atores e outros sistemas. A seguir são mostrados os diagramas de seqüência mais importantes do sistema, que correspondem aos casos de mais alto nível. Os eventos gerados pelo usuário serão simplificados, um conjunto de eventos pode ser apresentado como um evento apenas. Também não serão abordadas situações complexas, já que o objetivo é apenas entender o que o sistema deve oferecer ao usuário.

a) Cadastro de usuário: como mostra a figura 3, o usuário deve informar ao sistema que deseja cadastrar um novo usuário. Em seguida informa os dados deste usuário e pode, se não estiver conectado e desejar se conectar como o usuário cadastrado, solicitar a conexão automática ao sistema como este usuário quando terminar o cadastramento . Após esta operação o novo usuário estará cadastrado no sistema.

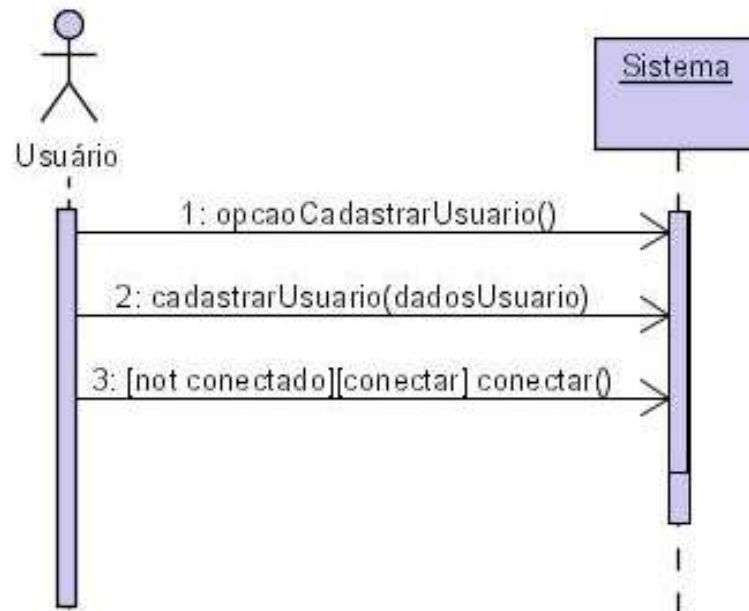


Figura 3 - Diagrama de seqüência do sistema cadastrar usuário

Conexão: como mostra a figura 4, o usuário deve informar ao sistema que deseja se conectar. Informa, em seguida, os dados do usuário. Ao término da operação o usuário estará conectado e identificado junto ao sistema.

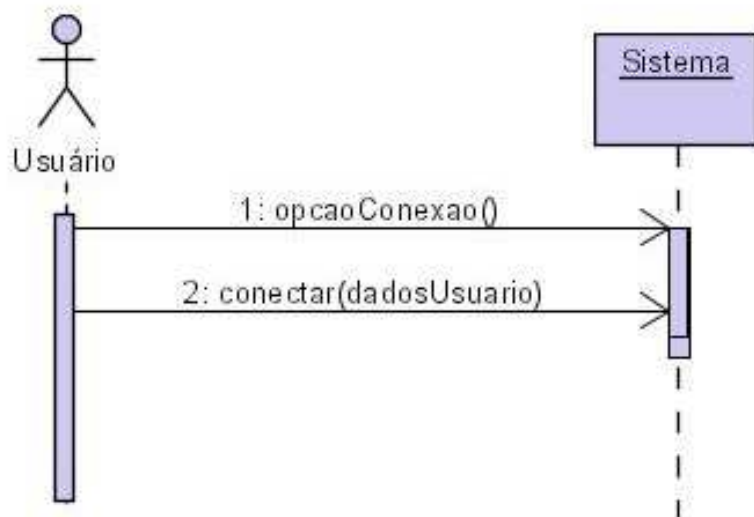


Figura 4 - Diagrama de seqüência do sistema conexão de usuário

b) Alteração de senha: na figura 5 é mostrado o diagrama de seqüência do sistema alterar senha. O usuário deve informar ao sistema que deseja alterar a senha, informa, em seguida, os dados deste usuário e pode, se não estiver conectado e desejar se conectar como o usuário que está alterando a senha, solicitar a conexão automática ao sistema. Após o termino da operação o usuário terá alterado sua senha no sistema.

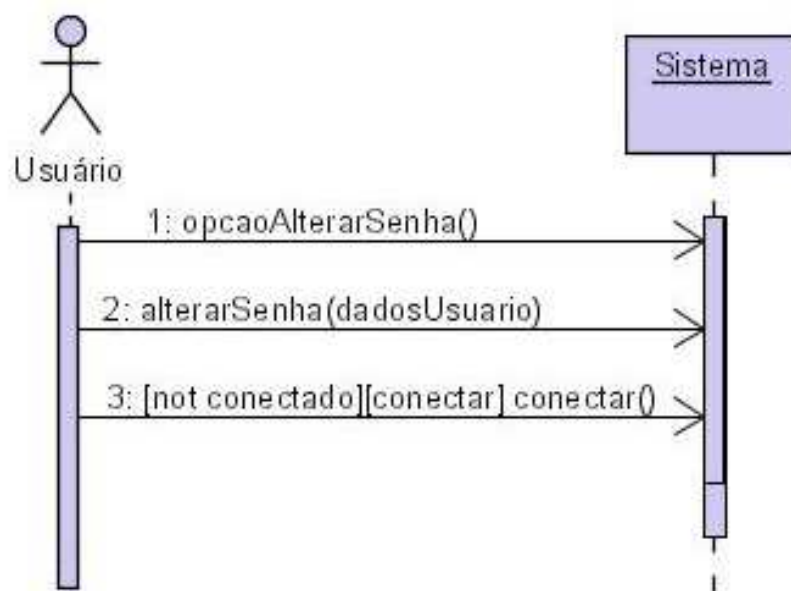


Figura 5 - Diagrama de seqüência do sistema alterar senha

c) Desconexão: na figura 6 é mostrado o diagrama de seqüência do sistema desconexão, o usuário informa ao sistema que deseja desconectar e a operação é efetuada. Após esta operação o usuário estará desconectado do sistema.

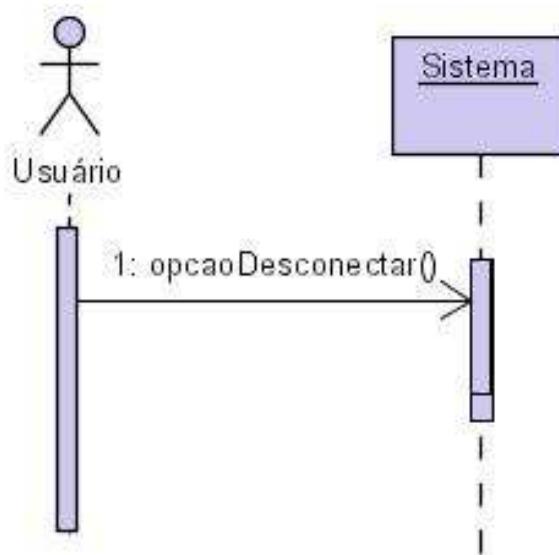


Figura 6 - Diagrama de seqüência do sistema desconexão de usuário

d) Cadastro de plano: como mostra a figura 7, o usuário deve informar ao sistema que deseja cadastrar um novo plano de governo e passa em seguida os dados do novo plano. Após esta operação o novo plano de governo fica cadastrado no sistema.

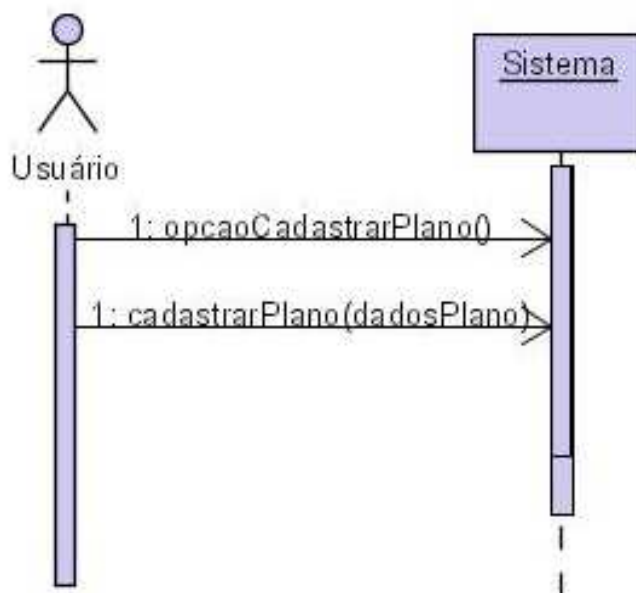


Figura 7 - Diagrama de seqüência do sistema cadastro de plano

e) Avaliação de plano de governo: na figura 8 é mostrado o diagrama de seqüência do sistema avaliar plano de governo. O diagrama mostra que o usuário deve fazer a opção de avaliar um plano de governo. O segundo passo é informar a teoria e o método que será utilizado, e informar o plano que será avaliado. Posteriormente, o usuário passa a responder o questionário e, por último, salvar as respostas, caso deseje. Normalmente, após esta operação o plano de governo estará avaliado e as respostas salvas.

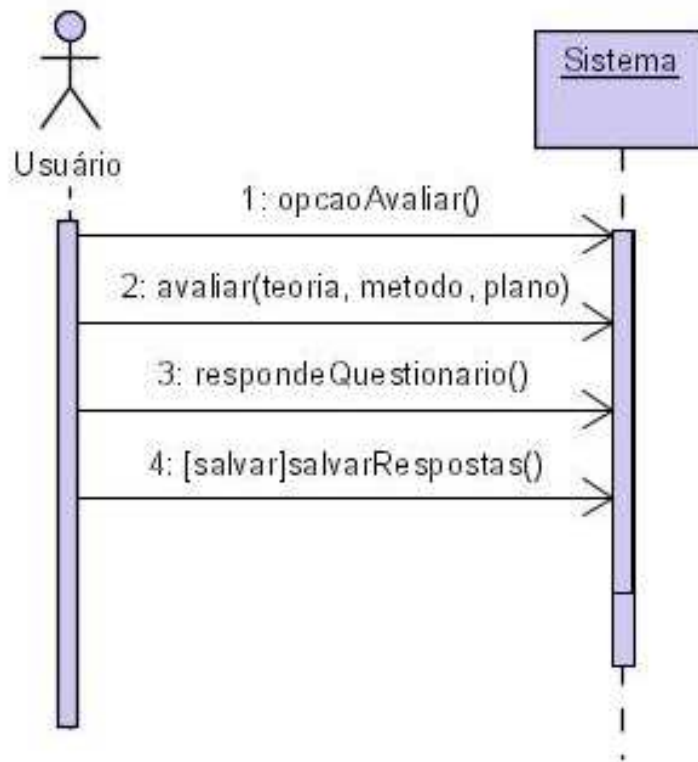


Figura 8 - Diagrama de seqüência do sistema avaliação de plano de governo

f) Cadastro de critério: como mostra a figura 9, o usuário primeiramente informa ao sistema que deseja cadastrar um critério de avaliação de plano de governo, informa, em seguida, os dados do critério a ser cadastrado, bem como a qual teoria e método o critério pertencerá. Após a operação o critério de avaliação estará cadastrado no sistema.

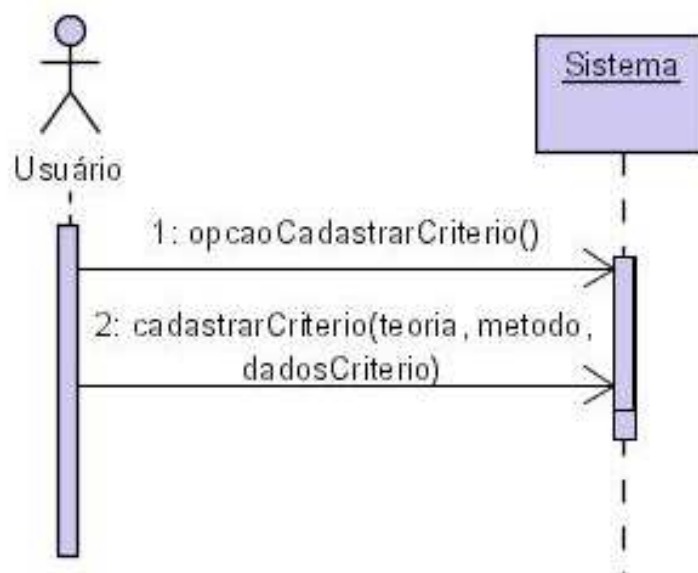


Figura 9 - Diagrama de seqüência do sistema cadastro de critério

g) Definição de pesos: como mostra a figura 10, o usuário inicialmente informa ao sistema que deseja definir pesos de critério de avaliação. O segundo passo é informar a teoria e o método que será utilizado, e informar o critério que terá seus pesos definidos. Posteriormente, o usuário passa a definir os pesos dos tópicos e das questões e, por último, salvar estes pesos, caso deseje. Normalmente, após esta operação o critério terá os seus pesos definidos.

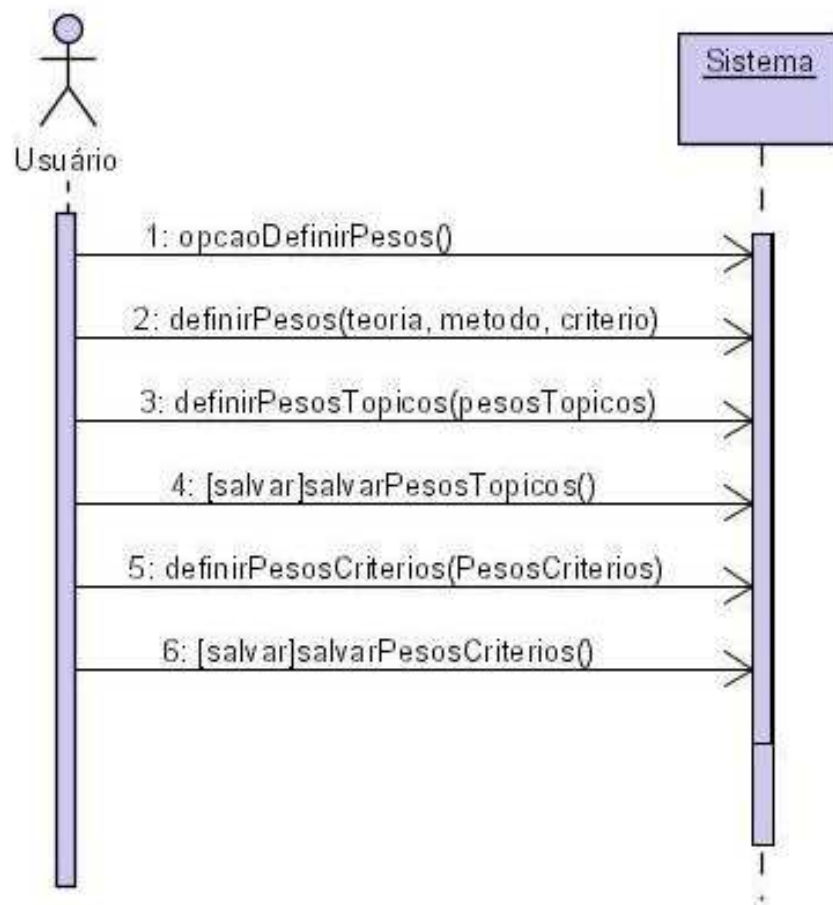


Figura 10 - Diagrama de seqüência do sistema definição de pesos

h) Exibição de relatório: como mostra a figura 11, o usuário inicialmente informa ao sistema que deseja exibir o relatório da avaliação, posteriormente informa a teoria, método, plano e os critérios que devem fazer parte do relatório. Após esta operação o usuário poderá visualizar o relatório da avaliação do plano de governo.

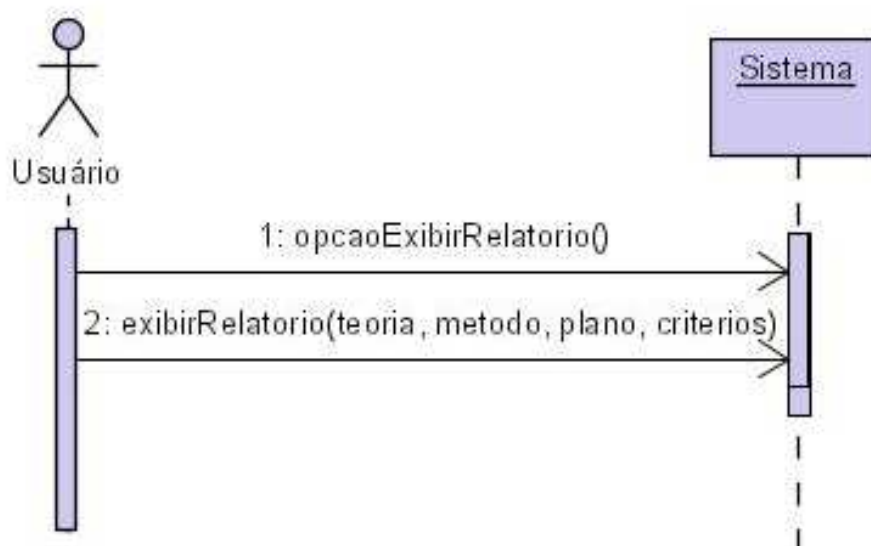


Figura 11 - Diagrama de seqüência do sistema exibir relatório

B Diagramas de estados

A figura 12 mostra o diagrama de estados do sistema. O estado conectado é composto de vários sub estados conforme mostra a figura 13. O sistema possui outros estados intermediários cuja análise está fora dos objetivos do trabalho.

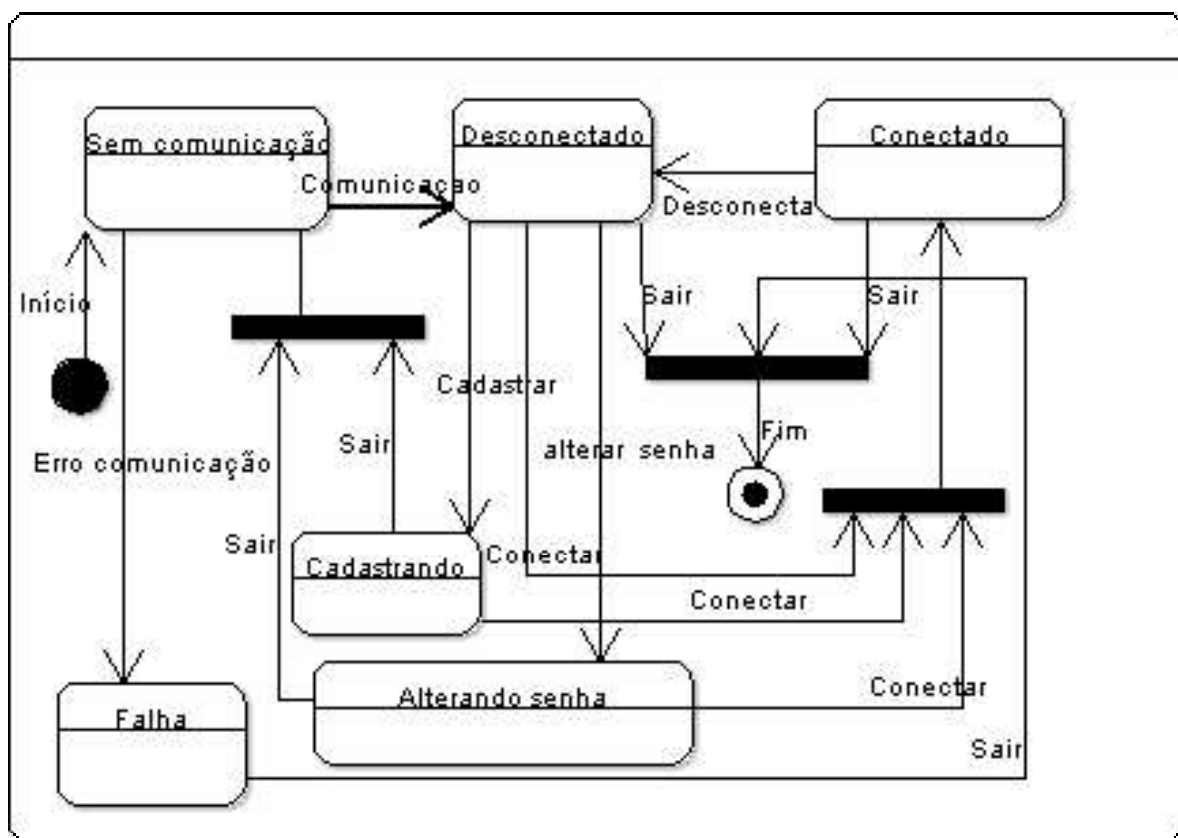


Figura 12 - Diagrama de estados do sistema

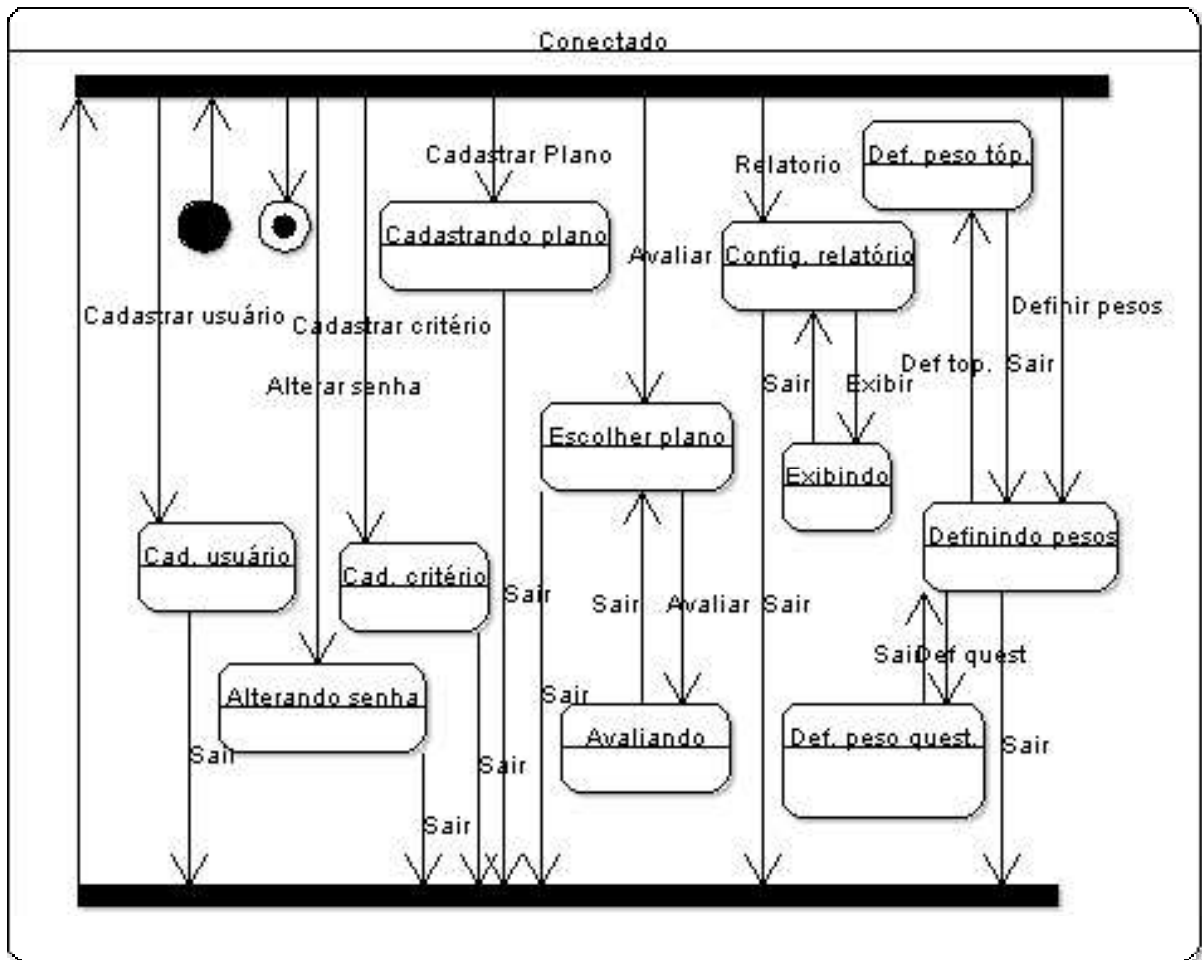


Figura 13 - Diagrama de estados quando o sistema está conectado

C Contratos do comportamento do sistema

Segundo Larman (2000), antes de se prosseguir para o projeto lógico, é necessário investigar e definir o comportamento do sistema como uma "caixa-preta". O comportamento, segundo ele, descreve o que o sistema faz sem explicar como ele faz, definindo as mudanças de estado do sistema que ocorrem a partir da invocação de uma operação. Os principais contratos do sistema de avaliação de planos de governo são os descritos a seguir.

Tabela 13 - Contrato do comportamento do sistema iniciar

Nome::	iniciar()
Responsabilidades:	Inicializa o sistema deixando-o pronto para ser utilizado pelo usuário.
Tipo:	Sistema
Referências cruzadas:	cadusuario(), conexao(), altsenha(), cadplano(), avaliar(), cadcriterio(), defpesoscrit(), defpesostopicos(), defpesosquestoes(), relatorios(), exiberelatorio().
Notas:	Acessar banco de dados.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro.
Saída:	
Pré-condições:	
Pós-condições:	O sistema estabeleceu comunicação com a base de dados; Todas as demais janelas foram criadas; Um menu de opções é disponibilizado ao usuário.

Tabela 14 - Contrato do comportamento do sistema cadusuario

Nome::	cadusuario()
Responsabilidades:	Cadastra um novo usuário no sistema.
Tipo:	Classe/interface
Referências cruzadas:	Conexão()
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário já estiver cadastrado, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário não pode estar cadastrado no sistema.
Pós-condições:	O novo usuário estará cadastrado no sistema. Se o novo usuário desejar ele já estará conectado ao sistema.

Tabela 15 - Contrato do comportamento do sistema conexao

Nome::	conexao()
Responsabilidades:	Conecta o usuário ao sistema, permitindo que utilize as funções do sistema que necessitam da conexão do usuário.
Tipo:	Classe/interface.
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver cadastrado, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar cadastrado no sistema.
Pós-condições:	O usuário estará conectado ao sistema.

Tabela 16 - Contrato do comportamento do sistema altsenha

Nome::	Altsenha()
Responsabilidades:	Altera a senha de um usuário.
Tipo:	Classe/interface.
Referências cruzadas:	Conexão()
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver cadastrado, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar cadastrado no sistema.
Pós-condições:	O usuário terá uma nova senha no sistema. Se o novo usuário desejar ele já estará conectado ao sistema.

Tabela 17 - Contrato do comportamento do sistema desconectausuario

Nome::	desconectausuario()
Responsabilidades:	Desconecta o usuário atual do sistema.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema.
Pós-condições:	O usuário estará desconectado do sistema.

Tabela 18 - Contrato do comportamento do sistema cadplano

Nome::	cadplano()
Responsabilidades:	Cadastra os dados de um plano de governo.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se já existir um plano com a mesma descrição para o mesmo usuário, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema; O mesmo plano não deve estar cadastrado no sistema..
Pós-condições:	O novo plano de governo estará cadastrado no sistema; O novo plano de governo estará vinculado ao usuário que está conectado.

Tabela 19 - Contrato do comportamento do sistema avaliar

Nome::	avaliar()
Responsabilidades:	Permite ao usuário escolher um plano de governo por ele cadastrado e responder um questionário de avaliação de um método disponível no sistema, cujas respostas serão vinculadas ao plano escolhido.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se não existir um plano para o usuário avaliar, exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema; Um plano deve estar cadastrado no sistema para ser avaliado.
Pós-condições:	Uma avaliação de plano será criada ou alterada caso já exista. A avaliação será vinculada ao usuário que está conectado, ao questionário e ao plano de governo respectivo.

Tabela 20 - Contrato do comportamento do sistema cadcritério

Nome::	cadcritério()
Responsabilidades:	Cadastra os dados de um critério de avaliação.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se já existir um critério com a mesma descrição para o mesmo usuário, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema; O mesmo critério não deve estar cadastrado no sistema..
Pós-condições:	O novo critério de avaliação estará cadastrado no sistema; O novo critério de avaliação estará vinculado ao usuário que está conectado e ao questionário respectivo.

Tabela 21 - Contrato do comportamento do sistema defpesostopicos

Nome::	defpesostopicos()
Responsabilidades:	Permite ao usuário definir os pesos dos tópicos de um critério de avaliação.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se não existir um critério para o usuário e questionário escolhido, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema; Um critério deve estar cadastrado no sistema.
Pós-condições:	Os pesos dos tópicos de um critério de avaliação estarão cadastrados no sistema.

Tabela 22 - Contrato do comportamento do sistema defpesosquestoes

Nome::	defpesosquestoes()
Responsabilidades:	Permite ao usuário definir os pesos das questões de um critério de avaliação.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se não existir um critério para o usuário e questionário escolhido, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	
Pré-condições:	O usuário deve estar conectado no sistema; Um critério deve estar cadastrado no sistema.
Pós-condições:	Os pesos das questões de um critério de avaliação estarão cadastrados no sistema.

Tabela 23 - Contrato do comportamento do sistema relatorios

Nome::	relatorios()
Responsabilidades:	Elabora o relatório e permite a sua consulta.
Tipo:	Classe/interface
Referências cruzadas:	
Notas:	Acessar banco de dados; disponibilizar uma interface amigável para escolha das opções.
Exceções:	Se não for possível estabelecer comunicação com a base de dados, exibir erro; Se o usuário não estiver conectado, exibir erro; Se não existir um plano de governo para o usuário exibir erro; Se não existir um critério para o usuário e questionário escolhido, exibir erro; Se o usuário não fornecer os dados solicitados exibir erro.
Saída:	Relatório da avaliação
Pré-condições:	O usuário deve estar conectado no sistema; Um plano deve estar cadastrado no sistema; Um critério deve estar cadastrado no sistema.
Pós-condições:	

Tabela 24 - Contrato do comportamento do sistema sair

Nome::	sair()
Responsabilidades:	Encerra o aplicativo
Tipo:	Sistema
Referências cruzadas:	
Notas:	
Exceções:	
Saída:	
Pré-condições:	
Pós-condições:	O sistema será finalizado e todas as janelas fechadas.

II Projeto

A Diagramas de interação

Os diagramas de interação mostram as interações com mensagens entre instâncias de classes do sistema. O objetivo das interações é a satisfação das pós-condições dos contratos de operações. Dois tipos de diagramas, os diagramas de colaboração e os diagramas de seqüência, são definidos pelo UML como diagramas de interação, e qualquer um deles pode ser utilizado para expressar interações entre mensagens (LARMAN, 2000). Os diagramas que serão utilizados são os diagramas de colaboração, apresentados a seguir.

a) Iniciar o sistema: como mostra a figura 14, o objeto que faz a função de controlar o sistema é o objeto 'avplgov'. Ao iniciar o sistema, o objeto 'avplgov' inicia os seus componentes e estabelece a comunicação com o banco de dados. Após esta operação, o objeto 'avplgov' cria as instâncias dos demais objetos que fazem parte do sistema. Estes objetos, quando criados, fazem a inicialização de seus componentes. O objeto 'avplgov' instancia o objeto criado enviando mensagem para que este estabeleça a comunicação com a base de dados. O objeto 'avplgov' é quem fará o controle do usuário que está utilizando o sistema, repassando esta informação aos demais objetos. Após estas operações o sistema estará inicializado e pronto para operar.

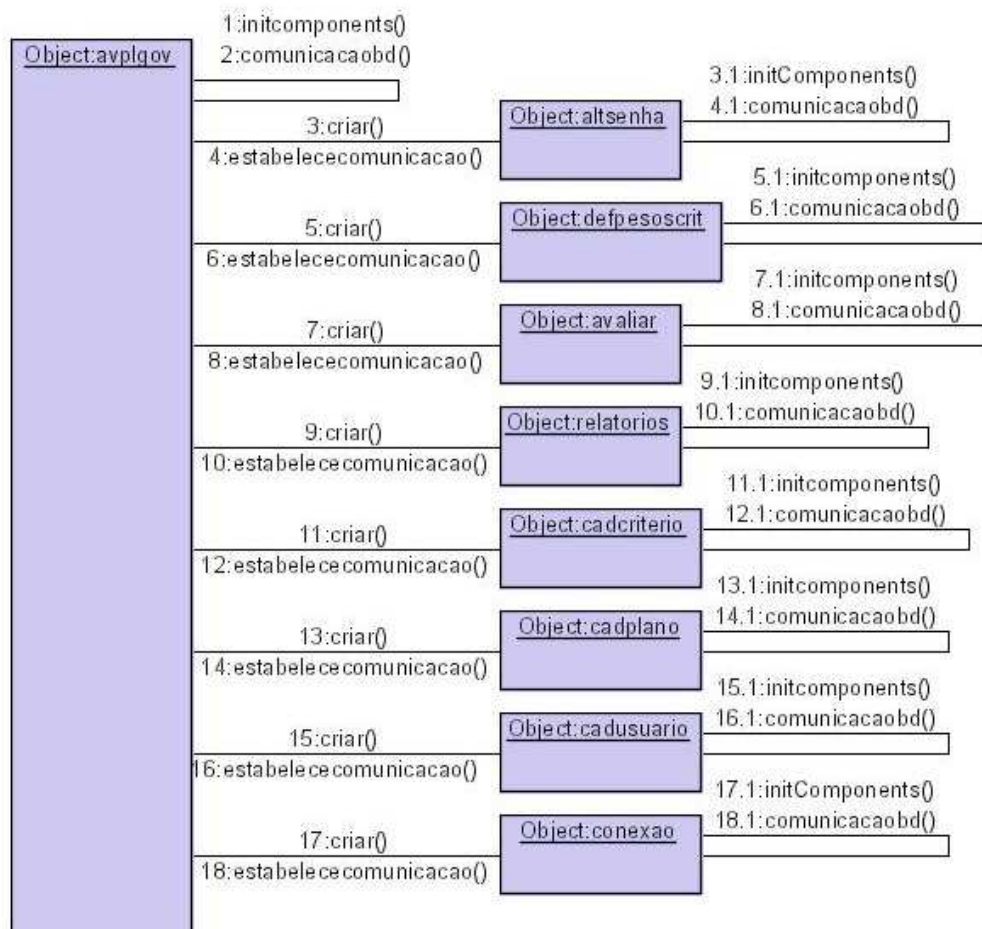


Figura 14 - Diagrama de colaboração iniciar o sistema

b) Cadastro de usuário: como mostra a figura 15, o diagrama de colaboração do sistema no cadastro de usuário se dá entre os objetos 'avplgov' (objeto controlador do sistema) e o objeto 'cadusuario' que oferece a função de cadastro através de interface com o usuário. O objeto 'avplgov' envia a mensagem 'cadastrarUsuario', passando como parâmetro o usuário que está conectado, caso haja um. O objeto 'cadusuario' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto 'cadusuario' atualiza a base de dados e retorna o controle ao objeto 'avplgov'. O objeto 'avplgov' envia mensagem ao objeto 'cadusuario' solicitando se houve uma conexão durante o cadastro e, em caso positivo, obtém o usuário. Uma terceira mensagem é enviada ao objeto 'cadusuario' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de cadastro de um novo usuário. Por último o objeto 'avplgov' faz alterações em seu próprio menu, caso seja necessário adapta-lo ao novo contexto.

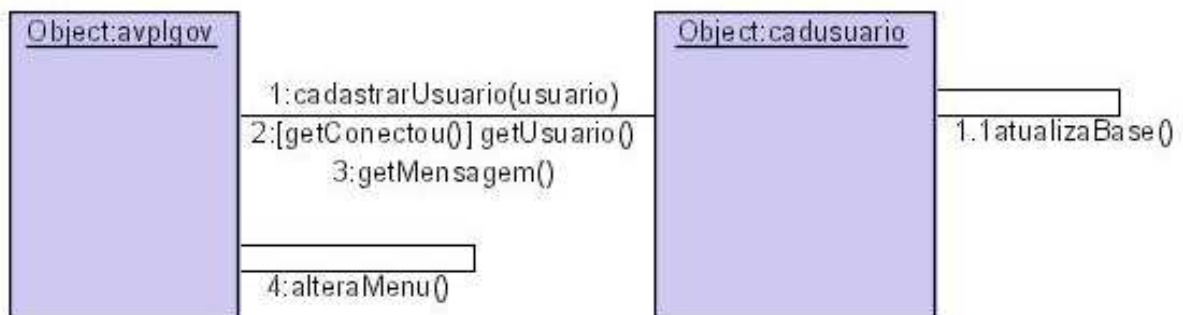


Figura 15 - Diagrama de colaboração cadastro de usuário

c) Conexão: como mostra a figura 16, o diagrama de colaboração do sistema na conexão se dá entre os objetos 'avplgov' (objeto controlador do sistema) e o objeto 'conexao' que oferece a função de conexão através de interface com o usuário. O objeto 'avplgov' envia a mensagem 'conectar'. O objeto 'conexao' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto 'conexao' atualiza a base de dados e retorna o controle ao objeto 'avplgov'. O objeto 'avplgov' envia mensagem ao objeto 'conexao' solicitando se houve uma conexão e, em caso positivo, obtém o usuário. Uma terceira mensagem é enviada ao objeto 'cadusuario' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de conexão. Por último o objeto 'avplgov' faz alterações em seu próprio menu, caso seja necessário adapta-lo ao novo contexto.

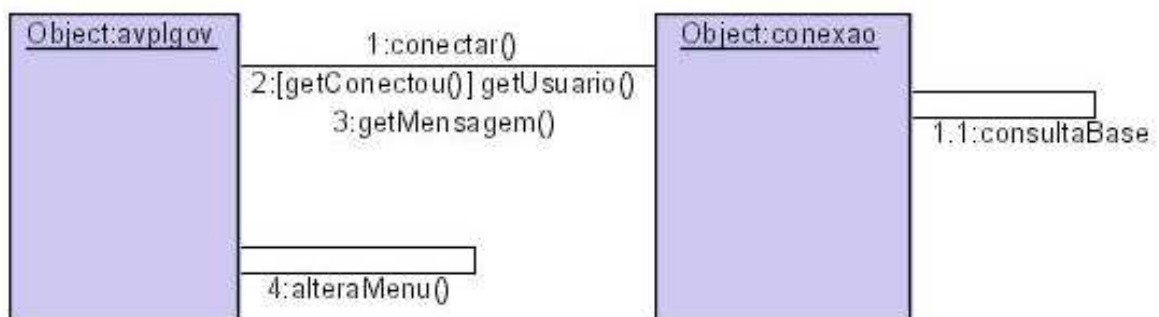


Figura 16 - Diagrama de colaboração conexão

d) Alteração de senha: como mostra a figura 17, o diagrama de colaboração do sistema na alteração de senha se dá entre os objetos 'avplgov' (objeto controlador do sistema) e o objeto 'altsenha' que oferece a função de alteração de senha através de interface com o usuário. O objeto 'avplgov' envia a mensagem 'alterarSenha'. O objeto 'altsenha' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto 'altsenha' atualiza a base de dados e retorna o controle ao objeto 'avplgov'. O objeto 'avplgov' envia mensagem ao objeto 'altsenha' solicitando se houve uma conexão e, em caso positivo, obtém o usuário. Uma terceira mensagem é enviada ao objeto 'altsenha' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de conexão. Por último o objeto 'avplgov' faz alterações em seu próprio menu, caso seja necessário adaptá-lo ao novo contexto.

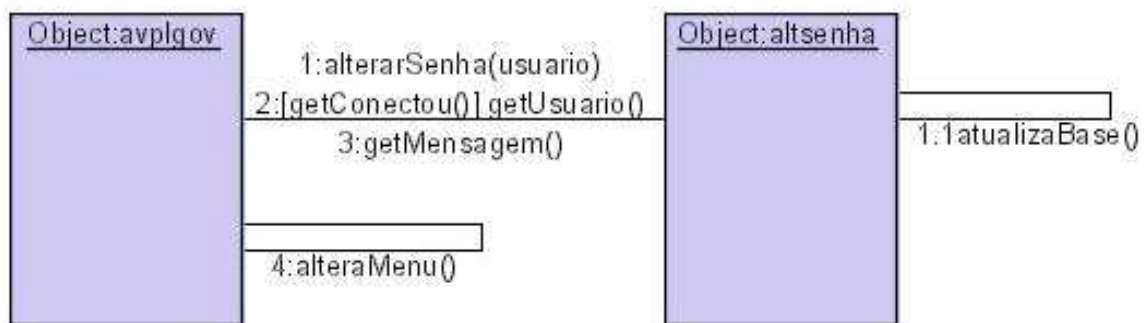


Figura 17 - Diagrama de colaboração alterar senha

e) Desconexão: na figura 18, o diagrama de colaboração do sistema na desconexão. A desconexão se é feita pelo próprio objeto '*avplgov*' (objeto controlador do sistema). Duas mensagens são enviadas, o comando de '*desconectar*' e o comando '*alteraMenu*' onde o objeto '*avplgov*' faz alterações em seu próprio menu, para adapta-lo ao novo contexto.



Figura 18 - Diagrama de colaboração desconexão

f) Cadastro de plano: como mostra a figura 19, o diagrama de colaboração do sistema no cadastramento de plano de governo se dá entre os objetos 'avplgov' (objeto controlador do sistema) e o objeto 'cadplano' que oferece a função de cadastramento de planos através de interface com o usuário. O objeto 'avplgov' envia a mensagem 'cadastrarPlano'. O objeto 'cadplano' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto 'cadplano' atualiza a base de dados e retorna o controle ao objeto 'avplgov'. Uma segunda mensagem é enviada ao objeto 'cadplano' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de cadastro.

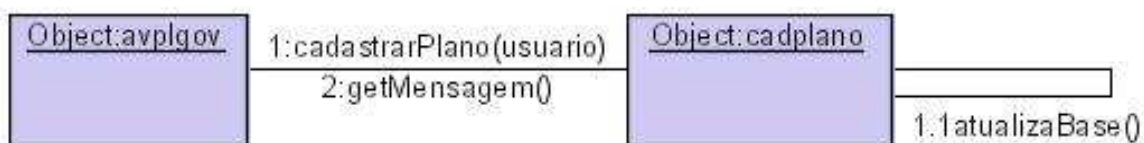


Figura 19 - Diagrama de colaboração cadastro de plano de governo

g) Avaliação de plano de governo: como mostra a figura 20, o diagrama de colaboração do sistema na avaliação de plano de governo se dá entre os objetos '*avplgov*' (objeto controlador do sistema) e o objeto '*avaliar*' que oferece a função de avaliação de planos através de interface com o usuário. O objeto '*avplgov*' envia a mensagem '*avaliar*'. O objeto '*avaliar*' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto '*avaliar*' atualiza a base de dados e retorna o controle ao objeto '*avplgov*'. Uma segunda mensagem é enviada ao objeto '*avaliar*' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de avaliação.

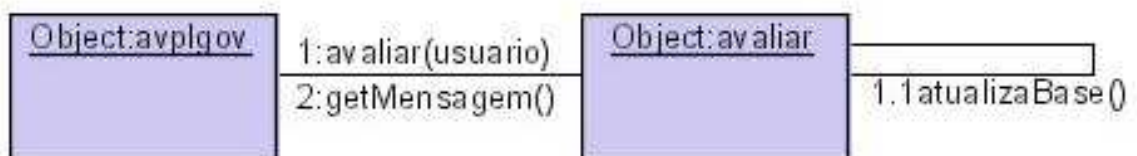


Figura 20 - Diagrama de colaboração avaliação de plano de governo

h) Cadastro de critério: como mostra a figura 21, o diagrama de colaboração do sistema no cadastramento de critério de cadastramento de critério de avaliação de plano de governo se dá entre os objetos '*avplgov*' (objeto controlador do sistema) e o objeto '*cadcritério*' que oferece a função de cadastramento de critério de avaliação de planos através de interface com o usuário. O objeto '*avplgov*' envia a mensagem '*cadcritério*'. O objeto '*cadcritério*' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto '*cadcritério*' atualiza a base de dados e retorna o controle ao objeto '*avplgov*'. Uma segunda mensagem é enviada ao objeto '*cadcritério*' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de cadastramento de critério de avaliação.

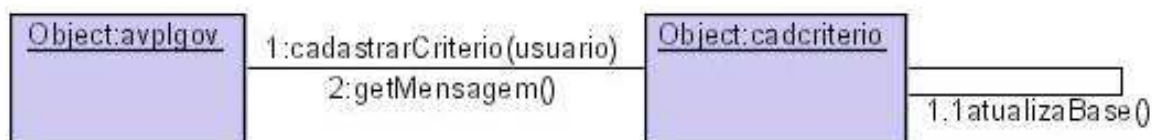


Figura 21 - Diagrama de colaboração cadastro de critério

i) Definição de pesos: como mostra a figura 22, o diagrama de colaboração do sistema na definição de pesos de plano de governo se dá entre os objetos '*avplgov*' (objeto controlador do sistema) e o objeto '*defpesoscrit*' que oferece a função de definição de pesos de planos através de interface com o usuário. O objeto '*avplgov*' envia a mensagem '*definirPesos*'. O objeto '*defpesoscrit*' exibe então uma interface ao usuário e passa a interagir com este. Após operar com o usuário, o objeto '*defpesoscrit*' atualiza a base de dados e retorna o controle ao objeto '*avplgov*'. Uma segunda mensagem é enviada ao objeto '*defpesoscrit*' solicitando uma mensagem que informará ao usuário como se deu a finalização do processo de definição de pesos.

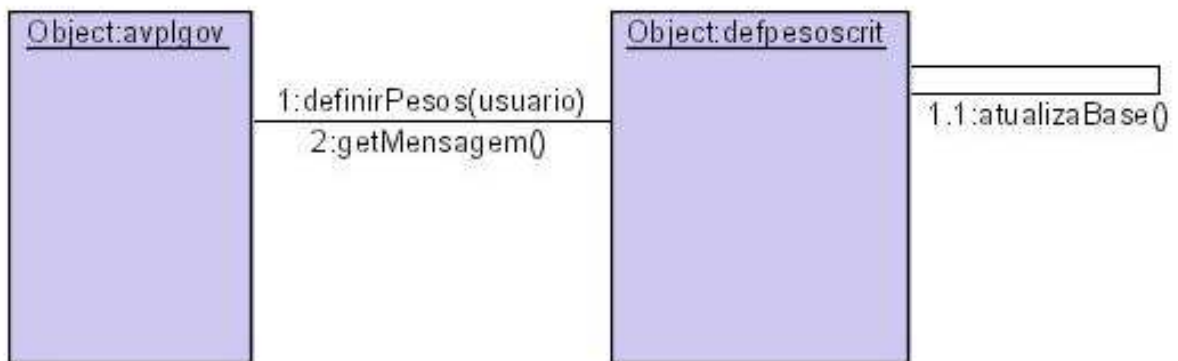


Figura 22 - Diagrama de colaboração definição de pesos

j) Exibição de relatório: como mostra a figura 23, o diagrama de colaboração do sistema na exibição de relatório de avaliação de plano se dá entre os objetos 'avplgov' (objeto controlador do sistema) e o objeto 'relatorios' que oferece a função de exibição de relatórios de avaliação de planos através de interface com o usuário. O objeto 'avplgov' envia a mensagem 'exibirRelatorio'. O objeto 'relatorios' exibe então uma interface ao usuário e passa a interagir com este. Durante a operação com o usuário, o objeto 'relatorios' consulta a base de dados e monta o relatório. Após operar com o usuário, o objeto 'relatorios' retorna o controle ao objeto 'avplgov'.

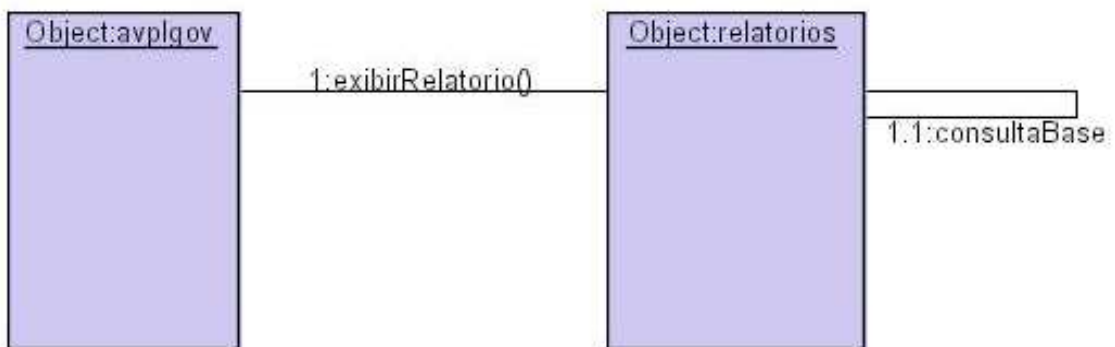


Figura 23- Diagrama de colaboração exibição de relatório

B Classes de projeto

De acordo com Larman (2000, p. 252), após ter feito o os diagramas de interação é possível identificar as classes de que participarão do software. As principais classes (finais) que colaboram entre si para efetuarem as funções definidas neste trabalho são as seguintes:

- a) Classe AVPLGOV: tem a finalidade de gerenciar as demais classes do sistema;
- b) Classe CADUSUARIO: tem a finalidade de permitir que sejam cadastrados usuários para o sistema;
- c) Classe ALTSENHA: tem a finalidade de permitir que o usuário altere sua senha;
- d) Classe CONEXAO: tem a finalidade de permitir que o usuário efetue a sua identificação no sistema;
- e) Classe CADPLANO: tem a finalidade de permitir que o usuário cadastre um novo plano de governo;
- f) Classe AVALIAR: tem a finalidade de permitir que o usuário responda o questionário, avaliando um plano de governo;
- g) Classe CADCRITERIO: tem a finalidade de permitir que o usuário cadastre um novo critério de avaliação de planos de governo;
- h) Classe DEFPEOSCRIT: tem a finalidade de permitir que o usuário defina os pesos das questões e dos tópicos das questões;
- i) Classe RELATORIOS: tem a finalidade de permitir que o usuário escolha os parâmetros que servirão de base para a montagem de um relatório da avaliação. Após esta escolha, esta classe então monta o relatório e exibe ao usuário;

C Esquema de Banco de Dados

Para o armazenamento das informações que são utilizadas pelo sistema, utilizou-se uma estrutura de banco de dados. Para se manter repositórios compartilhados de dados são usados sistemas de gerência de banco de dados (SGBD). O desenvolvimento de sistemas de informação ocorre quase que exclusivamente sobre banco de dados, com o uso de sistemas de gerenciamento de banco de dados relacional. A decisão pela utilização de um banco de dados se justifica pela simplicidade que esta solução apresenta, facilitando a inserção, alteração e exclusão de dados. Estas operações são efetuadas através de comandos relativamente simples que são executados com velocidade satisfatória. Para a criação do banco de dados, é necessário identificar quais informações precisam ser armazenadas e de que forma (HEUSER, 2001).

Para que sejam descritas as necessidades de armazenamento de um sistema utiliza-se um modelo de dados. Um modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. Para se construir um modelo de dados utiliza-se uma linguagem de modelagem de dados. Dois níveis de abstração podem ser considerados no desenvolvimento de um modelo de dados: o modelo conceitual e o modelo lógico. O modelo conceitual registra que dados aparecem no banco de dados, mas não registra como estes dados estão armazenados no SGBD. A técnica mais difundida na modelagem conceitual é a abordagem entidade-relacionamento (ER). O modelo lógico representa o banco de dados no nível de abstração do SGBD, ou seja, como ele será implementado em um caso particular de SGBD (HEUSER, 2001). O modelo lógico é representado na forma de tabelas. Os detalhes da construção das tabelas podem ser encontrados no apêndice A.

Na figura 24 é apresentado o modelo conceitual do banco de dados utilizado no sistema de avaliação de planos de governo. Este modelo apresenta as principais entidades do modelo e os relacionamentos mais importantes.

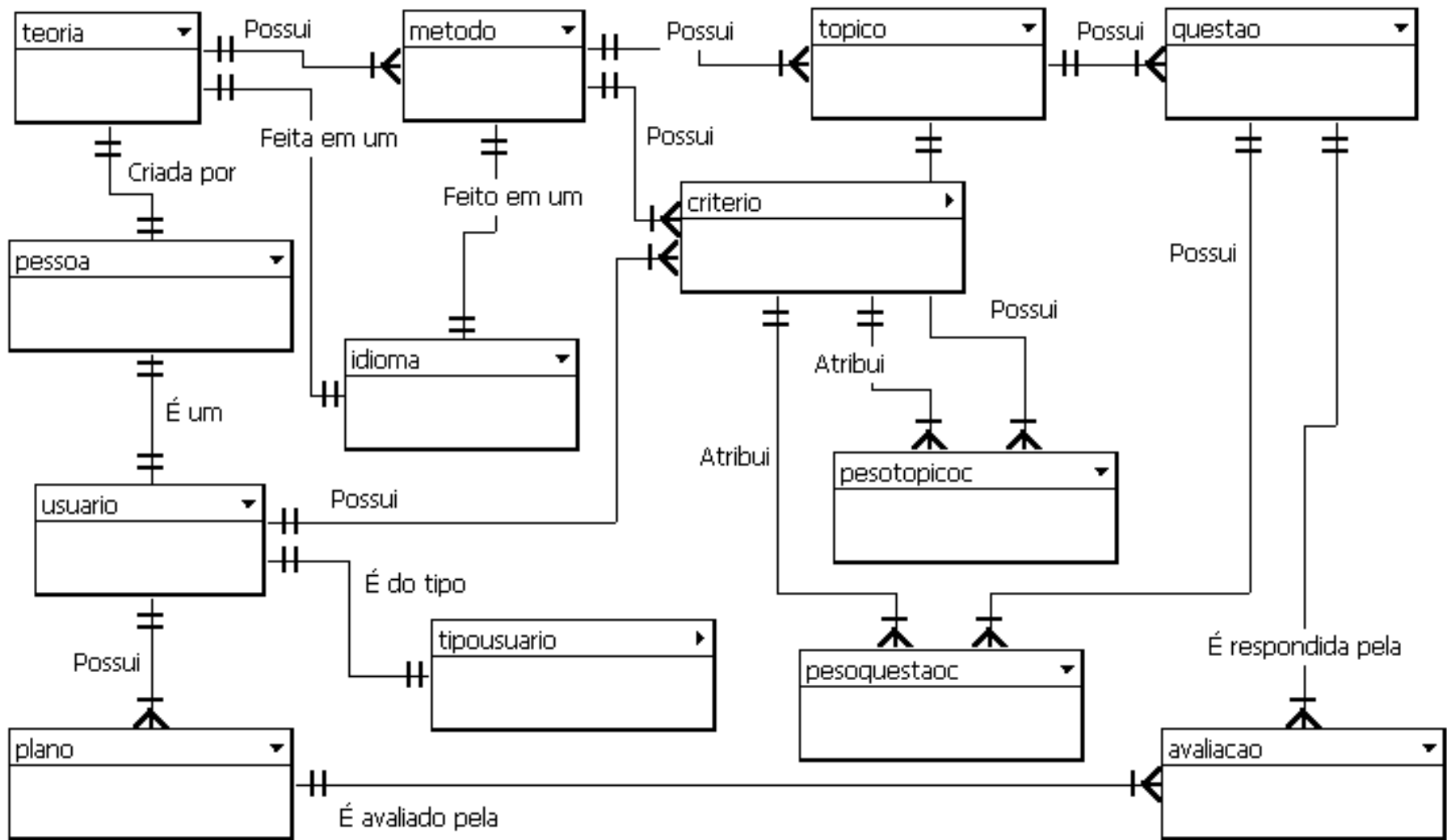


Figura 24 - esquema de banco de dados

4 Apresentação da Interface com Diferentes Telas e Formas de Uso

I Janela inicial do Sistema de Avaliação de Planos de Governo:

A janela inicial, mostrada na figura 25, que é a janela principal do sistema, é composta por quatro conjuntos de botões: cadastro e conexão; avaliações de planos; ferramentas para administradores; e, diversos. Na figura 25 podemos ver a janela inicial (na figura 25 não há usuário conectado). Algumas informações sobre cada um dos conjuntos de botões são dadas a seguir:

a) Cadastro e Conexão: sob este menu, o usuário encontra as funções de cadastro de usuários; conexão de usuário ou administrador; alteração de senhas; e, desconexão. O botão desconexão somente será habilitado caso o usuário esteja conectado, já o botão conexão somente estará habilitado caso o usuário esteja desconectado. Na parte inferior há um local apropriado para exibição de mensagens ao usuário. Como não há mensagem, significa que o sistema foi iniciado corretamente e houve comunicação com o servidor de dados.

b) Avaliações de Planos: sob este menu, o usuário encontra as funções de cadastro de plano de governo; avaliação de plano de governo; cadastro de critério de avaliação de plano; definição de pesos do(s) critério(s) de avaliação; e, exibição dos relatórios;

c) Ferramentas para Administradores: sob este menu, o(s) administrador(es) pode(m) cadastrar usuários e outros administradores; cadastrar novas teorias de avaliação de planos de governo; pesquisar problemas encontrados na base de dados. Nenhuma destas funções estão disponíveis, já que não fizeram parte do projeto do sistema, mas devem estar disponíveis no futuro; e,

d) Diversos: sob este menu o usuário pode obter informações sobre o sistema, ajuda e encerrar a execução do sistema. As funções de ajuda não estão disponíveis no momento, já que não fizeram parte do projeto do sistema, mas devem estar disponíveis no futuro.



Figura 25 - Janela inicial do sistema

II Cadastro de usuários:

Opção cadastro do menu cadastro e conexão. Nesta janela, mostrada na figura 26, o usuário que deseja utilizar o sistema e ainda não esteja cadastrado deve informar os seus dados. Deve ser utilizada apenas uma vez por cada novo usuário na primeira utilização conforme instruções a seguir:

- a) Nome completo: o usuário deve digitar o seu nome completo;
- b) E-Mail: o usuário deve digitar o seu e-mail;
- c) Identificador: também conhecido por login, é uma espécie de pseudônimo com o qual o usuário se identificará ao sistema. É de livre escolha do usuário, porém o sistema não aceita dois usuários com o mesmo identificador. Caso já haja um usuário com o identificador escolhido o sistema emitirá um aviso e solicitará a mudança;
- d) Sexo: marcar a opção correspondente ao sexo do usuário;
- e) Senha: o usuário deve escolher uma senha de segurança;
- f) Confirmação de senha: o usuário deve digitar a mesma senha que foi digitada no campo senha;
- g) Opção conectar: se selecionada esta opção e caso haja sucesso no cadastramento, o usuário já estará automaticamente conectado após o cadastramento;
- h) Botão ok: o usuário confirma todos os dados e solicita ao sistema que finalize o cadastramento. Caso haja sucesso esta janela será fechada e o sistema retornará à

janela inicial. Caso haja algum erro uma mensagem será exibida no rodapé da janela;

i) Botão cancelar: o usuário solicita ao sistema para ignorar os dados, abandonar esta janela e voltar para a janela inicial.



A imagem mostra uma janela de software intitulada "Cadastro de Usuários". O título da janela é "Cadastre os seus dados". A interface contém os seguintes elementos:

- Um campo de texto para "Nome Completo:".
- Um campo de texto para "E-Mail:".
- Um campo de texto para "Identificador:".
- Um campo de texto para "Senha:".
- Um campo de texto para "Confirme a Senha:".
- Um grupo de opções para "Sexo:" com dois botões de opção: "Masculino" e "Feminino".
- Um botão de opção desativado rotulado "Conectar".
- Dois botões de ação: "Ok" e "Cancelar".
- Um espaço reservado vazio na base da janela.

Figura 26 - Janela de cadastro de usuário

III Conexão de usuário:

Opção conexão do menu cadastro e conexão. Nesta janela, mostrada na figura 27, o usuário que deseja utilizar o sistema e já esteja cadastrado deve confirmar os seus dados de acordo com as instruções a seguir:

- a) Identificador: também conhecido por login, é uma espécie de pseudônimo com o qual o usuário se identificará ao sistema. O usuário deve informar o identificador com o qual se cadastrou;
- b) Senha: o usuário deve informar a sua senha de segurança;
- c) Botão ok: o usuário confirma todos os dados e solicita ao sistema que tente conectar com os dados fornecidos. Caso haja sucesso esta janela será fechada e o sistema retornará à janela inicial. Caso haja algum erro uma mensagem será exibida no rodapé da janela;
- d) Botão cancelar: o usuário solicita ao sistema para ignorar os dados, abandonar esta janela e voltar para a janela inicial.

Conexão de Usuário

Conexão de usuário

Identificador:

Senha:

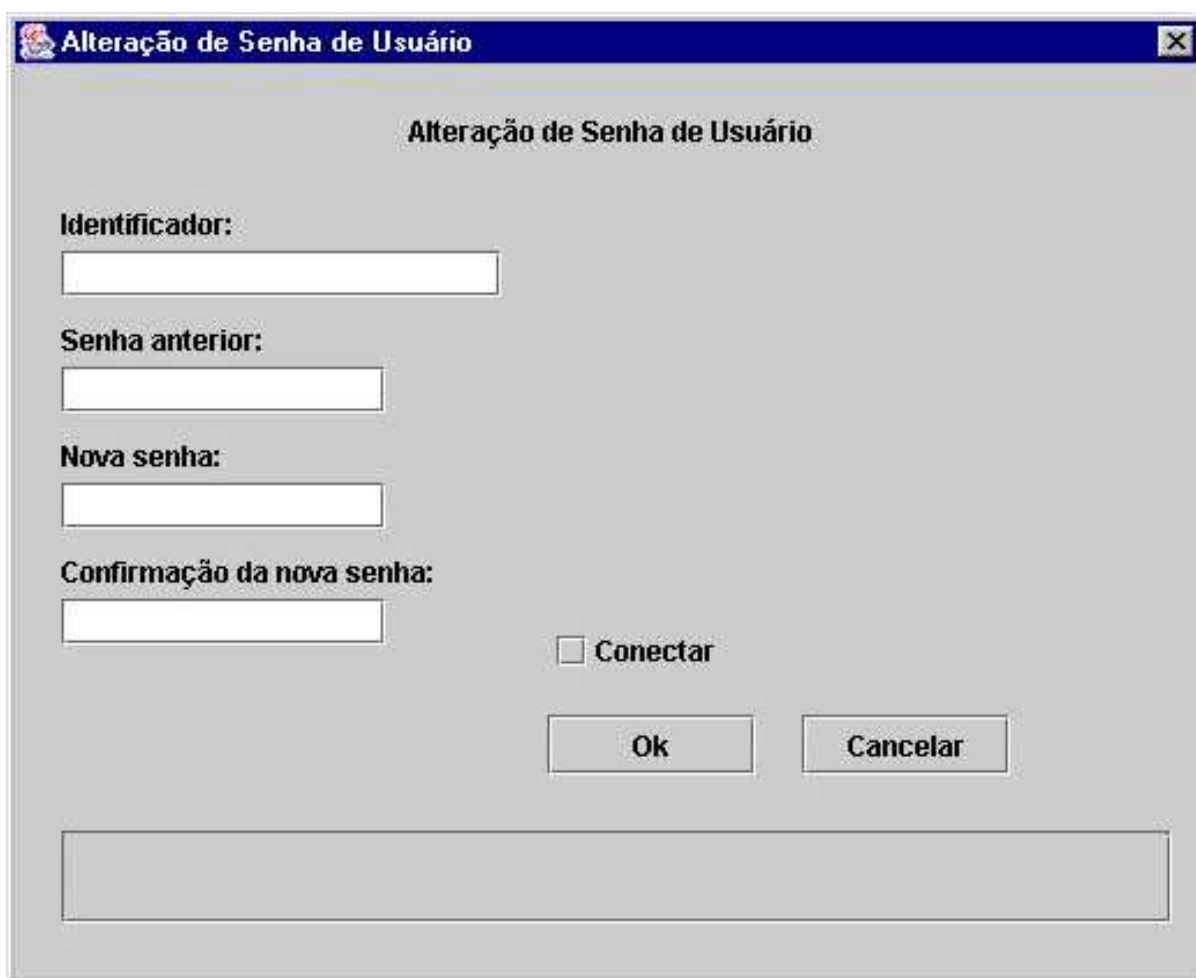
Ok **Cancelar**

Figura 27 - Janela de conexão de usuário

IV Alteração de senha:

Opção alteração de senha do menu cadastro e conexão. Nesta janela, mostrada na figura 28, o usuário que deseja alterar a sua senha e já esteja cadastrado deve confirmar os seus dados de acordo com as instruções a seguir:

- a) Identificador: O usuário deve informar o identificador com o qual se cadastrou;
- b) Senha anterior: o usuário deve informar a sua senha de segurança que deseja alterar;
- c) Nova senha: o usuário deve escolher uma nova senha de segurança;
- d) Confirmação da nova senha: o usuário deve digitar a mesma senha que foi digitada no campo nova senha;
- e) Opção conectar: se selecionada esta opção e caso haja sucesso na alteração da senha, o usuário já estará automaticamente conectado após esta alteração;
- f) Botão ok: o usuário confirma todos os dados e solicita ao sistema que tente alterar a senha com os dados fornecidos. Caso haja sucesso esta janela será fechada e o sistema retornará à janela inicial. Caso haja algum erro uma mensagem será exibida no rodapé da janela;
- g) Botão cancelar: o usuário solicita ao sistema para ignorar os dados, abandonar esta janela e voltar para a janela inicial.



A janela de diálogo intitulada "Alteração de Senha de Usuário" apresenta os seguintes campos e elementos:

- Identificador:** Campo de texto para o nome de usuário.
- Senha anterior:** Campo de texto para a senha atual.
- Nova senha:** Campo de texto para a nova senha.
- Confirmação da nova senha:** Campo de texto para confirmar a nova senha.
- Conectar
- Botões "Ok" e "Cancelar".
- Uma barra decorativa horizontal na base da janela.

Figura 28 - Janela de alteração de senha do usuário

V Desconexão de usuário:

Opção desconexão do menu cadastro e conexão. A figura 29 mostra em destaque o botão desconexão. Basta clicar no botão desconexão e o usuário será desconectado. Nenhuma janela será aberta. Nenhuma confirmação será solicitada. O botão desconexão será desabilitado e o botão conexão será habilitado permitindo uma nova conexão.

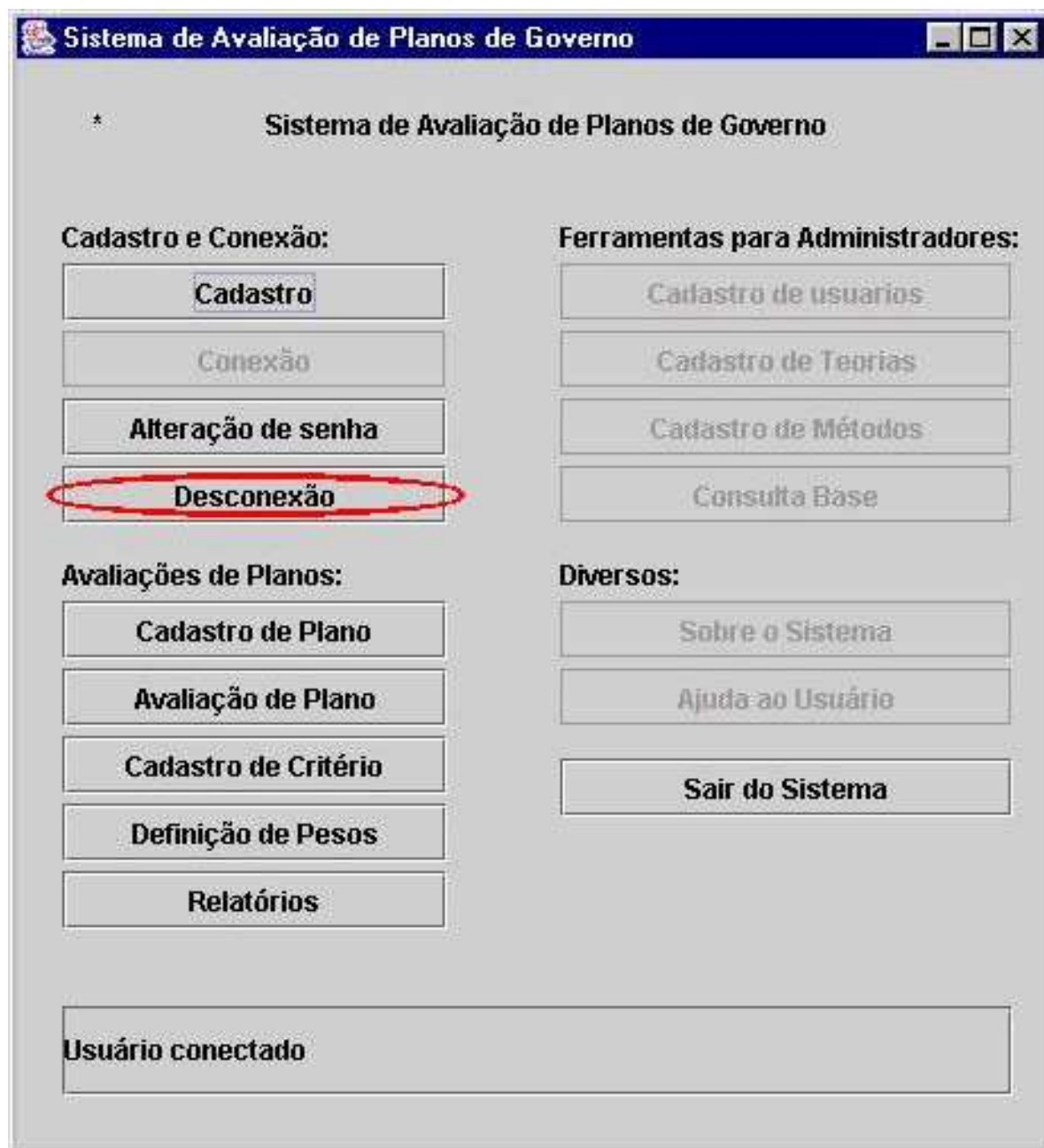


Figura 29 - Opção desconexão de usuário na janela inicial

VI Cadastro de plano de governo:

Opção cadastro de plano da janela principal. Ao contrário do que o título acima talvez possa indicar, o usuário não cadastra o conteúdo de um plano de governo no sistema. Na opção de cadastro de plano, o usuário irá cadastrar apenas um nome ao qual ele vincule o plano de governo que deseja posteriormente avaliar. A forma como o plano será nomeado é livre, mas é importante que haja alguma informação esclarecedora no nome ao próprio usuário no futuro. Essa orientação leva em conta que inúmeros planos podem ser cadastrados pelo mesmo usuário. Dessa forma, nomes sugestivos podem evitar confusão futura. Orientamos que no nome haja, por exemplo, indicação do nome do órgão governamental, data de elaboração e local ou área onde o plano irá ser executado ou aplicado.

A figura 30, mostra a janela de cadastro de plano de governo. A utilização da janela deve ser feita de acordo com as instruções a seguir:

- a) Nome do plano de governo: informar o nome do plano de governo que será avaliado. O nome é de livre escolha do usuário mas deve possuir algum vínculo com o plano que será avaliado. A indicação do nome do órgão governamental, data de elaboração e local ou área onde o plano irá ser executado ou aplicado facilita posteriormente a identificação do plano;
- b) Botão cadastrar: o usuário confirma o nome do plano e solicita ao sistema que tente cadastrá-lo. No caso de sucesso: o plano será incluído na base de dados do sistema e automaticamente vinculado ao usuário que o cadastrou; a janela será fechada e o sistema retornará à janela inicial; e, uma mensagem de sucesso será

exibida na janela inicial. Caso haja algum erro uma mensagem será exibida no rodapé desta janela que não será fechada;

c) Botão cancelar: o usuário solicita ao sistema para ignorar o nome do plano de governo sem cadastrá-lo, abandonar esta janela e voltar para a janela inicial.



Cadastro de Plano de Governo

Cadastro de Plano de Governo para avaliação

Nome do Plano de Governo:

Plano diretor da cidade de Planalto - RS

Cadastrar **Cancelar**

Figura 30 - Janela de cadastro de plano de governo

VII Avaliação de plano de governo:

Opção avaliação de plano do menu avaliações de planos. Avaliar um plano é, fundamentalmente, responder a um questionário de avaliação. É uma tarefa que gera como resultado uma resposta do usuário para cada questão do questionário de avaliação.

As questões são elaboradas por especialistas em avaliações de planos de governo, com base em sua experiência profissional. A um conjunto fechado de questões de avaliação compiladas por um especialista chamamos de método de avaliação. Um método é baseado em uma teoria de avaliação formulada também por um especialista. Uma mesma teoria pode gerar vários métodos de avaliação que dependem da interpretação da teoria e do contexto onde será aplicada. Exemplo: o sistema inicialmente possui uma teoria de avaliação cadastrada. Essa teoria foi desenvolvida por William Baer. Na sua teoria, William Baer já propõe, explicitamente, questões de avaliação. Mas isto pode ser modificado. Baer propõe que os itens podem ser alterados e, um determinado intérprete, também especialista, propor um outro modelo de questões (método). Esse método, poderia ser válido em determinado país com suas leis, usos e costumes, mas não em outro com leis, usos e costumes diferentes. Necessário seria criar dois métodos da mesma teoria, mas com suas questões adaptadas ao contexto específico.

As questões, por sua vez podem ser agrupadas em temas semelhantes. Ao grupo de temas semelhantes chamou-se de tópico. O tópico é uma forma de agrupar as questões. O grupamento de questões em tópicos é importante como se verá adiante, mas não é obrigatório.

Pode-se ver na figura 31 a janela que é exibida, quando é pressionado o botão avaliação de plano do menu de avaliações de planos, esta é a janela de escolha do plano que será avaliado.

Na janela de escolha de plano, figura 31, devem ser escolhidos: a teoria; seu método de avaliação; e, o plano que será avaliado. Ao se escolher uma teoria e um método de avaliação, entende-se que o usuário deseja avaliar o plano através da teoria e método escolhidos. O sistema verificará se já existe uma avaliação do plano através daquela teoria e método. Se a avaliação existir as respostas gravadas serão recuperadas e disponibilizadas ao usuário para alteração. Caso contrário, uma avaliação do plano será criada. A criação de uma avaliação deixa sempre as respostas em branco, ou seja, as questões não recebem uma resposta 'sim' ou 'não'. Essa decisão de projeto permite ao usuário responder parcialmente ao questionário, ou responder ao questionário em uma seqüência diversa da apresentada pelo sistema. As questões não respondidas permanecem em branco. Quando o usuário retornar a mesma avaliação identificará as questões não respondidas, facilitando a continuação do trabalho.

A utilização da janela deve ser feita de acordo com as instruções a seguir:

a) Escolha uma teoria de avaliação: nesta caixa de escolha (será chamado de 'combobox') estarão listadas as teorias de avaliação que foram cadastradas pelo administrador do sistema. Somente o administrador do sistema pode incluir teorias de avaliação. Inicialmente somente haverá uma teoria de avaliação disponível. Além da descrição da teoria, informações adicionais são apresentadas, tais como autor, data da publicação e idioma original em que foi escrita. A figura 31 mostra a descrição de uma teoria intitulada "teoria para avaliação geral de planos: uma forma

de se fazer planos melhores ...", esta teoria foi desenvolvida por Baer. Se não for possível apresentar de forma completa a descrição das teorias existentes no 'combobox', por falta de espaço horizontal, uma barra de rolagem será apresentada na parte inferior desse 'combobox' (ver as setas para esquerda e direita). O item que aparece inicialmente no 'combobox' é o primeiro, ele está automaticamente selecionado. Para selecionar outro item, o usuário deve clicar na descrição do item inicial e o 'combobox' se abrirá para que o usuário possa selecionar o item desejado. Ao mudar o item selecionado, o outro 'combobox', o que contém os métodos, será reinicializado e o seu primeiro item selecionado, esse fato merece atenção caso o usuário escolha um método de avaliação e depois altere a teoria;

b) Escolha um método: neste 'combobox' estarão listados os métodos de avaliação que foram cadastradas pelo administrador do sistema e que estão vinculados a teoria escolhida no 'combobox' de escolha de teorias. Somente o administrador do sistema pode incluir métodos de avaliação. Inicialmente somente haverá dois métodos de avaliação disponíveis para a teoria existente de William Baer. Na verdade os dois métodos são compostos das mesmas questões em idiomas diferentes. A primeira forma de expressar as questões propostas é em português e a outra em inglês. A inclusão do método em inglês tem caráter informativo. Além da descrição do método, informações adicionais são apresentadas, tais como data da publicação e idioma original em que foi desenvolvido o método. A figura 31 mostra a descrição de um método intitulado "método com questões em português da teoria para avaliação geral de planos...", este método foi desenvolvido por Baer. Se não for possível apresentar de forma completa a descrição dos métodos existentes no 'combobox', por falta de espaço horizontal, uma barra de rolagem será apresentada na parte inferior desse 'combobox' (ver as setas para esquerda e direita). O item que

aparece inicialmente no '*combobox*' é o primeiro, e ele está automaticamente selecionado. Para selecionar outro item, o usuário deve clicar na descrição do item inicial e o '*combobox*' se abrirá para que o usuário selecione o item desejado. Os métodos são vinculados às teorias, portanto, cada teoria possui um conjunto próprio de métodos. Os itens do '*combobox*' métodos são alterados quando o usuário altera a teoria que está selecionada no '*combobox*' teoria;

c) Escolha o plano de governo que deve ser avaliado: no terceiro '*combobox*' o usuário irá escolher o plano que deseja avaliar. Os planos de governo listados são apenas os cadastrados pelo usuário que está conectado. Os planos cadastrados por outros usuários não podem ser visualizados por enquanto. Futuramente, em outra fase do projeto, será criado um meio para que um usuário possa avaliar um plano cadastrado por outro através de permissão concedida pelo usuário que cadastrou o plano. Tal ferramenta agregará maior transparência e precisão na avaliação do plano já que diversas pessoas poderão avaliar o mesmo plano e os resultados serão consolidados em apenas um relatório;

d) Botão avaliar: o usuário confirma todas as escolhas e solicita ao sistema que abra uma segunda janela de avaliação de planos que é mostrada na figura 32. Se existir uma avaliação do plano através da teoria e método selecionados, os seus dados são recuperados e exibidos, senão será criada uma avaliação vinculando o plano à teoria e método escolhidos. Caso haja algum erro uma mensagem será exibida no rodapé desta janela;

e) Botão cancelar: o usuário solicita ao sistema para abandonar esta janela e voltar para a janela inicial.

Avaliação de Planos de Governo

Avaliação de Planos de Governo - Escolha a teoria o método e o plano que será avaliado:

Escolha uma teoria de avaliação

1 - Teoria para avaliação geral de planos: uma forma de se fazer planos melhores - William C. Baer - english (U.S.A)

Escolha um método:

1 - Método com questões em português da teoria para avaliação geral de planos - português (Brasil) - 0000-00-00

Escolha o plano de governo que será avaliado:

Plano diretor da cidade de Planalto - RS

Avaliar Cancelar

Figura 31 - Janela de escolha do plano de governo que será avaliado

VIII Questionário de avaliação de plano de governo:

Na janela de questionário, apresentada na figura 32, e obtida após pressionado o botão avaliar do menu avaliações de planos, o usuário obtém uma série de informações sobre a questão de avaliação que é apresentada. Na mesma janela o usuário acessa o texto da questão, devendo decidir a sua resposta.

Informações apresentadas na janela e funcionamento dos seus botões e opções:

- a) Teoria: nome da teoria de avaliação de planos que foi escolhida;
- b) Autor: nome do autor da teoria;
- c) Idioma da teoria: idioma em que a teoria foi escrita;
- d) Ano da teoria: ano em que a teoria foi desenvolvida;
- e) Quantidade de tópicos: quantidade de tópicos do método;
- f) Quantidade total de questões: quantidade total de questões do método;
- g) Método usado: descrição do método que foi escolhido;
- h) Idioma do método: idioma em que o método foi escrito;
- i) Tópico atual/total: informa o número do tópico ao qual a questão apresentada pertence e o número total de tópicos separados por uma barra;
- j) Questão atual do tópico/total no tópico: informa o número da questão apresentada e o número total de questões no tópico atual separados por uma barra;
- k) Questão atual total/total geral: informa o número seqüencial da questão apresentada e o número total de questões no tópico atual separados por uma barra;

- l) Tópico: exibe o texto do tópico ao qual a questão pertence;
- m) Questão: exibe a questão que deve ser analisada e respondida pelo usuário;
- n) Resposta sim/não: o usuário, após analisar a questão, pode escolher uma das respostas possíveis - sim ou não - ou deixá-la sem resposta caso não haja condições de respondê-la no momento. Ao ser cadastrado um plano de governo todas as respostas para os questionários estarão automaticamente em branco. Uma resposta em branco significa que não foi definida uma resposta sim ou não para a questão. Após respondida uma questão com o sim ou o não, e desde que a resposta tenha sido salva - ver item *opção salvar respostas ao sair sem concluir*, a mesma não mais poderá voltar para a situação "em branco". O usuário poderá livremente alterar a resposta de uma pergunta de sim para não e de não para sim;
- o) Botões voltar e avançar: permitem a navegação seqüencialmente entre as questões. O botão voltar estará desabilitado ao usuário quando estiver na primeira questão. O botão avançar estará desabilitado ao usuário quando estiver na última questão;
- p) Opção salvar respostas ao sair sem concluir: esta opção permite ao usuário salvar ou não as respostas do questionário não respondidas completamente. Esta opção somente estará habilitada ao usuário caso haja pelo menos uma questão sem resposta. Se esta opção estiver habilitada e selecionada, ao clicar no botão sair o sistema atualizará as respostas efetuadas pelo usuário;
- q) Botão concluir: este botão somente estará habilitado caso o questionário tenha sido completamente respondido. Ao clicar no botão concluir o sistema atualizará incondicionalmente as respostas efetuadas pelo usuário. Caso o usuário não deseje atualizar as respostas deve utilizar o botão sair (quando o botão concluir estiver

habilitado, o item *opção salvar respostas ao sair sem concluir* estará desabilitada fazendo com que as respostas não sejam atualizadas ao clicar no botão sair);

r) Botão sair: este botão estará sempre habilitado e seu funcionamento é dependente da *opção salvar respostas ao sair sem concluir*. Dependendo desta opção, as respostas serão ou não atualizadas. Quando todo o questionário for respondido, o botão sair não atualizará as respostas. Para atualizar as respostas nessa situação é necessário utilizar o botão concluir, que apenas nessa situação estará habilitado;

Questionário

Teoria: Teoria para avaliação geral de planos: uma forma de se fazer planos melhores
Autor: William C. Baer
Idioma da teoria: english (U.S.A)
Ano da teoria: 2004 **Quantidade de tópicos:** 8 **Quantidade total de questões:** 60
Método usado: Método com questões em português da teoria para avaliação geral de planos
Idioma do método: português (Brasil)

Tópico Atual/Total: 1/8 **Questão Atual do Tópico/Total no Tópico:** 1/10 **Questão Atual Total/Total Geral:** 1/60

Tópico:
Explica o contexto e a colocação: o quê e o porquê da documentação. Não é evidente por si só ao público..

Questão:
O contexto político/legal do planejamento é explicado (exemplo: encontros, ordens estaduais, discussões públicas, que

Resposta:
 Sim
 Não

Salvar respostas ao sair sem concluir

Há questões não respondidas.

Figura 32 - Janela do questionário de avaliação do plano de governo

IX Cadastro de critério:

Na janela de cadastro de critério, apresentada na figura 33, e obtida após pressionado o botão cadastro de critério do menu avaliações de planos, o usuário deve escolher uma teoria e um método que será vinculado ao novo critério de avaliação. Um critério de avaliação é um conjunto de pesos que são atribuídos às questões e aos tópicos. Quando é cadastrado um critério de avaliação para um método de uma teoria, deseja-se que as questões tenham um peso diferente no resultado da avaliação. Para exemplificar: suponha que em um determinado questionário com três questões, para um determinado contexto, a primeira questão seja um fator muito importante na avaliação do plano. A segunda questão é de importância intermediária, enquanto a terceira possui menos importância.

É fácil perceber que a primeira questão deverá possuir, na avaliação dos resultados, um peso maior que o peso da segunda, e, este, por sua vez, um peso maior que o da terceira. Pode-se, então, ainda como exemplo, arbitrar um peso três para a primeira questão, um peso dois para a segunda e peso unitário para a terceira. Esta análise porém é subjetiva.

Como as questões estão separadas em tópicos, pode acontecer que um determinado tópico seja mais importante que outro na avaliação do plano. Nesse caso são atribuídos pesos diferenciados para estes tópicos. Os pesos das questões e dos tópicos permanecem separados, ou seja, dentro de um mesmo tópico as questões podem ter pesos diferentes. No momento do cálculo dos resultados, o sistema irá multiplicar o peso da questão pelo peso de seu respectivo tópico para obter o peso final da questão.

Para esclarecer o assunto serão mostrados três exemplos de critérios que serão comentados.

Para os exemplos, supõe-se um método que possui três tópicos, tópico A, tópico B, tópico C, com duas, três e quatro questões respectivamente, formando um total de nove questões. As questões 1 e 2 pertencem ao tópico A, as questões 3, 4 e 5 pertencem ao tópico B, e, as questões 6, 7, 8 e 9 pertencem ao tópico C.

Exemplo do critério nº 1:

Pesos dos tópicos:

Tópico	Peso
A	1
B	2
C	1

Pesos das questões:

Questão	Peso
1	1
2	2
3	2
4	1
5	2
6	1
7	1
8	2
9	2

Analisando-se os pesos dos tópicos, pode-se observar, no exemplo do critério nº 1, que o tópico B possui um peso maior que cada um dos outros dois tópicos individualmente. Isto significa que cada questão pertencente ao tópico B

vale, inicialmente, o dobro, para fins de avaliação, em relação a cada questão dos tópicos A e C.

Analisando o peso das questões, pode-se observar que algumas questões têm o peso unitário e outras o peso dois. Chega-se a uma conclusão idêntica à dos tópicos.

Analisando as duas tabelas em conjunto, chega-se então a composição final:

Questão	Tópico pertencente	Peso da questão	Peso do tópico	Peso Final	% peso da questão
1	A	1	1	1	5,26%
2	A	2	1	2	10,53%
3	B	2	2	4	21,05%
4	B	1	2	2	10,53%
5	B	2	2	4	21,05%
6	C	1	1	1	5,26%
7	C	1	1	1	5,26%
8	C	2	1	2	10,53%
9	C	2	1	2	10,53%
Total	-	-	-	19	100,00%

No exemplo acima, as questões 3 e 5, sozinhas, representam mais que 42% do peso total do questionário.

Exemplo do critério nº 2:

Pesos dos tópicos:

Tópico	Peso
A	2
B	3
C	2

Pesos das questões:

Questão	Peso
1	2
2	3
3	3
4	2
5	3
6	2
7	2
8	3
9	3

Os pesos dos tópicos no critério nº 2 foram estabelecidos da seguinte maneira:

- Os tópicos do critério nº 1 que possuíam peso 1 passaram a ter peso 2;
- O tópico do critério nº 1 que possuía peso 2 passou a ter peso 3;
- As questões do critério nº 1 que possuíam peso 1 passaram a ter peso 2;
- As questões do critério nº 1 que possuíam peso 2 passaram a ter peso 3;

Em resumo, aumentou-se em uma unidade o valor dos pesos dos tópicos e das questões do critério nº 1.

Analisando-se o peso dos tópicos, pode-se observar, no exemplo do critério nº 2, que o tópico B possui um peso maior que cada um dos outros dois tópicos individualmente. Isto significa que cada questão pertencente ao tópico B vale, inicialmente, uma vez e meia, para fins de avaliação, em relação a cada questão dos tópicos A e C.

Analisando-se o peso das questões, pode-se observar que algumas questões têm o peso dois e outras o peso três.

Analisando as duas tabelas em conjunto, chega-se então a composição final:

Questão	Tópico pertencente	Peso da questão	Peso do tópico	Peso Final	% peso da questão
1	A	2	2	4	7,41%
2	A	3	2	6	11,11%
3	B	3	3	9	16,67%
4	B	2	3	6	11,11%
5	B	3	3	9	16,67%
6	C	2	2	4	7,41%
7	C	2	2	4	7,41%
8	C	3	2	6	11,11%
9	C	3	2	6	11,11%
Total	-	-	-	54	100,00%

No exemplo acima, as questões 3 e 5, sozinhas representam agora pouco mais que 33% do peso total do questionário, uma redução em comparação ao critério nº 1. Este exemplo mostra que:

- não há obrigatoriedade de pesos unitários do sistema;
- os pesos das questões são relativos entre si: se todas as questões de um critério, por exemplo, possuem peso 4, então os resultados finais serão iguais ao resultado apresentado se todas as questões possuísem peso 1;
- o que vale é a proporcionalidade entre os pesos e não o valor dos números em si.

Exemplo do critério nº 3:

Se for repetido o procedimento acima, substituindo, em relação ao critério nº 1, os pesos de valor 1 para 10 e os pesos de valor 2 para 11, é obtida a tabela abaixo:

Questão	Tópico pertencente	Peso da questão	Peso do tópico	Peso Final	% peso da questão
1	A	10	10	100	10,18%
2	A	11	10	110	11,20%
3	B	11	11	121	12,32%
4	B	10	11	110	11,20%
5	B	11	11	121	12,32%
6	C	10	10	100	10,18%
7	C	10	10	100	10,18%
8	C	11	10	110	11,20%
9	C	11	10	110	11,20%
Total	-	-	-	982	100,00%

No exemplo acima, as questões 3 e 5, sozinhas representam agora pouco mais que 24% do peso total do questionário. Se todos os tópicos e questões apresentassem pesos unitários, essa soma seria pouco mais que 22%.

Esta estratégia, de valores de pesos altos e próximos, pode ser utilizada quando um tópico ou questão representar pequenas diferenças de peso em relação ao outro tópico ou questão.

O sistema aceita também números fracionários e o usuário pode utilizá-los. Neste caso, aconselhamos para que o menor peso seja unitário e os demais correspondam a valores que representem a proporcionalidade pretendida.

Um critério está sempre vinculado a um método, que por sua vez está vinculado a uma teoria. Um critério não está vinculado a um plano de governo nem a

avaliação de um plano de governo. O critério, entretanto, pode ser utilizado para exibir os resultados de uma avaliação. Ou seja, o usuário cadastra um plano e responde ao questionário (avaliação de plano) de um método específico. Cadastra pelo menos um critério de avaliação para este método específico e define os pesos para os tópicos e questões do método. Agora é possível ver os resultados da avaliação (das respostas do questionário) utilizando como parâmetro o critério. O mesmo critério pode ser reutilizado como parâmetro para exibição de resultados de outros planos, desde que haja uma avaliação destes planos através do mesmo método para o qual o critério foi criado. Vários critérios podem ser criados para o mesmo método. É possível exibir os resultados de uma avaliação utilizando vários critérios ao mesmo tempo, como será visto adiante.

Para facilitar a utilização do sistema, foi cadastrado um critério que pode ser utilizado por todos os usuários e possui pesos unitários. Este critério não pode ter seus pesos alterados.

Para cadastrar um critério, utilizar as instruções a seguir:

- a) Escolha uma teoria de avaliação: observar as instruções do item VII a) na página 105;
- b) Escolha um método: instruções do item VII b) na página 106;
- c) Descrição: informar a descrição do critério que será cadastrado. A descrição é de livre escolha do usuário mas deve possuir algum vínculo com os pesos do critério. Pode-se, por exemplo, citar o tipo de ênfase que o critério dá aos tópicos. A indicação do tipo de planos que utilizarão o critério facilita posteriormente a sua identificação. O texto digitado neste campo será o apresentado no momento de se

escolher o critério. Exemplo de preenchimento: 'Critério que estabelece peso unitário para todas as questões. 80% para aprovação. Método de Baer';

d) Observações: O texto digitado neste campo facilitará posteriormente a identificação do critério. Na versão inicial do sistema o texto digitado não será utilizado, mas planeja-se, para versões posteriores, apresentar as observações no momento da escolha do critério para a apresentação dos resultados. O usuário deve preencher o campo pois o texto é armazenado para uso futuro. Exemplo de preenchimento: 'Este critério estabelece uma porcentagem mínima de 80% para a aprovação do plano. Todos os tópicos possuem peso unitário. Todas as questões possuem peso unitário. Não estabelece porcentagem mínima para cada um dos tópicos';

e) Porcentagem mínima para aprovação do plano: este campo deve ser preenchido com o valor do percentual necessário para que o plano seja aprovado dentro desse critério. Um critério pode possuir pesos diferenciados para as suas questões. O usuário pode informar ao sistema qual a menor porcentagem de questões respondidas afirmativamente, juntamente com seus pesos, satisfaz ao critério. No momento da apresentação dos resultados o sistema informará o percentual de questões respondidas afirmativamente e se este percentual atinge o valor mínimo para a aprovação que foi informado. A análise de um plano de governo é subjetiva. Portanto esta análise percentual tem apenas a finalidade de auxiliar ao usuário. Como o sistema calcula o percentual no momento da exibição dos resultados? Primeiro, somam-se os pesos de todas as questões do questionário do método ao qual o critério se refere e armazena-se o resultado como valor 'a'; em seguida, somam-se os pesos de todas as questões do mesmo questionário que foram respondidas afirmativamente na avaliação do plano e armazena-se como valor 'b';

por último, multiplica-se o valor 'b' por cem e o resultado é dividido pelo valor 'a', obtendo-se o percentual de questões que foram respondidas afirmativamente. Este percentual é comparado com o valor que foi informado pelo usuário e se for igual ou maior a este último, o plano será considerado aprovado pelo sistema, caso contrário o plano será considerado reprovado;

f) Botão cadastrar: o usuário solicita ao sistema que cadastre um novo critério vinculado a teoria e ao método escolhidos, e, também, ao usuário conectado. No caso de sucesso: o critério será incluído na base de dados do sistema e automaticamente vinculado ao usuário que o cadastrou a teoria e método escolhidos; a janela não será fechada, uma mensagem de sucesso será exibida no rodapé da janela, o botão cancelar se tornará botão sair e o sistema aguardará o cadastro de um novo critério ou o clique no botão sair. Caso haja algum erro uma mensagem será exibida no rodapé desta janela;

g) Botão cancelar: o usuário solicita ao sistema para ignorar os dados, abandonar esta janela e voltar para a janela inicial. O texto deste botão será alterado para sair após o usuário cadastrar pelo menos um critério, neste caso, clicando o botão sair o usuário solicita ao sistema para ignorar apenas os dados que não foram cadastrados, abandonar a janela e voltar para a janela inicial.

Cadastro de Critério do Usuário

Cadastro de Critério do Usuário - Escolha a teoria e o método para cadastrar o critério desejado:

Escolha uma teoria de avaliação

1 - Teoria para avaliação geral de planos: uma forma de se fazer planos melhores - William C. Baer - english (U.S)

Escolha um método:

1 - Método com questões em português da teoria para avaliação geral de planos - português (Brasil) - 0000-00-0

Dados do novo critério

Descrição:
Critério com ênfase para questões subjetivas. 80% para aprovação. Método de Baer

Observações:
Os planos que serão avaliados sob este critério precisam satisfazer com maior ênfase os itens subjetivos do r

Porcentagem mínima para aprovação do plano:
80

Cadastrar **Cancelar**

Figura 33 - Janela de cadastro de critério

X Definição de pesos dos critérios:

Na janela de definição de pesos dos critérios, apresentada na figura 34, e obtida após pressionado o botão definição de pesos do menu avaliações de planos, o usuário deve escolher uma teoria, um método e o critério que terá os seus pesos definidos ou alterados. Ao escolher uma teoria e um método, somente os critérios que foram cadastrados pelo usuário para a teoria e método escolhidos serão listados. Um critério de avaliação é um conjunto de pesos que são atribuídos às questões e aos tópicos. Quando se cadastra um critério de avaliação para um método de uma teoria, deseja-se que as questões tenham um peso diferente no resultado da avaliação. Mais informações sobre critérios de avaliação podem ser obtidas no item IX cadastro de critério na página 113. O usuário deverá definir os pesos dos tópicos e os pesos das questões em dois momentos.

A utilização da janela deve ser feita de acordo com as instruções a seguir:

- a) Escolha uma teoria de avaliação: observar as instruções do item VII a) na página 105;
- b) Escolha um método: observar as instruções do item VII b) na página 106;
- c) Escolha o critério de avaliação para atribuir pesos: no terceiro '*combobox*' o usuário irá escolher o critério cujos pesos deseja definir ou alterar. Os critérios do próprio sistema não podem ser alterados e portanto não serão apresentados neste '*combobox*'. Os critérios cadastrados por outros usuários não podem ser utilizados e também não são apresentados neste '*combobox*';

d) Botão definir pesos dos tópicos: o usuário confirma todas as escolhas e solicita ao sistema que abra uma janela de definição de pesos dos tópicos que é mostrada na figura 35. O sistema automaticamente identificará se o critério já tem os pesos dos tópicos definidos ou não. Caso os pesos já tenham sido definidos eles serão apresentados e poderão ser alterados senão os pesos terão seus valores iniciais definidos pelo usuário. Caso haja algum erro uma mensagem será exibida no rodapé desta janela;

e) Botão definir pesos das questões: o usuário confirma todas as escolhas e solicita ao sistema que abra uma janela de definição de pesos das questões que é mostrada na figura 36. O sistema automaticamente identificará se o critério já tem os pesos das questões definidos ou não. Caso os pesos já tenham sido definidos eles serão apresentados e poderão ser alterados senão os pesos terão seus valores iniciais definidos pelo usuário. Caso haja algum erro uma mensagem será exibida no rodapé desta janela;

f) Botão cancelar: o usuário solicita ao sistema abandonar esta janela e voltar para a janela inicial.

Definição de Pesos dos Critérios

Escolha a teoria, o método e o critério para definir os pesos do critério escolhido:

Escolha uma teoria de avaliação

1 - Teoria para avaliação geral de planos: uma forma de se fazer planos melhores - William C. Baer - english (U.S.A)

Escolha um método:

1 - Método com questões em português da teoria para avaliação geral de planos - português (Brasil) - 0000-00-00

Escolha o critério de avaliação para atribuir pesos:

1 - Critério com ênfase para questões subjetivas. 80% para aprovação. Método de Baer

Definir Pesos dos Tópicos Definir Pesos das Questões Cancelar

Figura 34 - Janela de definição de pesos dos critérios

XI Pesos dos tópicos:

Na janela de pesos dos tópicos, apresentada na figura 35, e obtida após pressionado o botão definir pesos dos tópicos da janela definição de pesos dos critérios, o usuário obtém uma série de informações sobre o tópico que terá seu peso definido ou alterado. Na mesma janela o usuário acessa também informações sobre as questões que pertencem a este tópico para facilitar a sua decisão.

Informações apresentadas na janela e funcionamento dos seus botões e opções:

- a) Teoria: nome da teoria de avaliação de planos que foi escolhida;
- b) Autor: nome do autor da teoria;
- c) Idioma da teoria: idioma em que a teoria foi escrita;
- d) Ano da teoria: ano em que a teoria foi desenvolvida;
- e) Quantidade de tópicos: quantidade de tópicos do método;
- f) Quantidade total de questões: quantidade total de questões do método;
- g) Método usado: descrição do método que foi escolhido;
- h) Idioma do método: idioma em que o método foi escrito;
- i) Tópico atual/total: informa o número do tópico ao qual a questão apresentada pertence e o número total de tópicos separados por uma barra;
- j) Critério: informa a descrição do critério cujos tópicos terão seus pesos definidos ou alterados, conforme o caso;
- k) Tópico: exibe o texto do tópico cujo peso será definido ou alterado;

l) Questões do tópico: exibe as questões do tópico e os pesos definidos para cada uma delas caso esses pesos tenham sido cadastrados. Este campo tem como finalidade fornecer informações adicionais para que o usuário possa definir o valor mais apropriado para o peso do tópico. O usuário pode navegar dentro das questões utilizando a barra de rolagem vertical;

m) Peso do tópico: neste campo, o usuário, após analisar as informações apresentadas, deve definir um valor numérico, correspondente ao peso do tópico, ou deixá-lo sem valor, caso não haja condições de defini-lo no momento. Ao ser cadastrado um critério de avaliação, todos os pesos estarão automaticamente em branco. Um peso em branco significa que não foi definido um peso para o tópico específico. Após definido o peso do tópico, e desde que o peso tenha sido salvo -ver item o) *opção salvar respostas ao sair sem concluir*, o mesmo não mais poderá voltar para a situação "em branco". O usuário poderá livremente alterar o peso de um tópico. Se o tópico já possuía um peso definido, esse valor será apresentado, cabendo ao usuário decidir se irá alterá-lo. Se o tópico não possuía um peso definido então o campo estará em branco;

n) Botões voltar e avançar: permitem a navegação seqüencialmente entre os tópicos. O botão voltar estará desabilitado ao usuário quando estiver no primeiro tópico. O botão avançar estará desabilitado ao usuário quando estiver no último tópico;

o) Opção salvar respostas ao sair sem concluir: esta opção permite ao usuário salvar os pesos dos tópicos que foram definidos. Esta opção somente estará habilitada ao usuário caso haja pelo menos um tópico sem peso. Se esta opção estiver habilitada e selecionada, ao clicar no botão sair o sistema atualizará os pesos definidos pelo usuário;

p) Botão concluir: este botão somente estará habilitado caso os pesos tenham sido completamente definidos. Ao clicar no botão concluir o sistema atualizará incondicionalmente os pesos definidos pelo usuário. Caso o usuário não deseje atualizar os pesos deve utilizar o botão sair (quando o botão concluir estiver habilitado, o item *opção salvar respostas ao sair sem concluir* estará desabilitado fazendo com que as respostas não sejam atualizadas ao clicar no botão sair);

q) Botão sair: este botão estará sempre habilitado e seu funcionamento é dependente da *opção salvar respostas ao sair sem concluir*. Dependendo desta opção, os pesos serão ou não atualizados. Quando todos os pesos forem definidos pelo usuário, o botão sair não atualizará as respostas. Para atualizar as respostas nessa situação é necessário utilizar o botão concluir, que apenas nessa situação estará habilitado;

Pesos dos Tópicos

Teoria: Teoria para avaliação geral de planos: uma forma de se fazer planos melhores
Autor: William C. Baer
Idioma da teoria: english (U.S.A)
Ano da teoria: 2004 **Quantidade de tópicos:** 8 **Quantidade total de questões:** 60
Método usado: Método com questões em português da teoria para avaliação geral de planos
Idioma do método: português (Brasil)
Tópico Atual/Total: 1/8

Critério: Critério com ênfase para questões subjetivas. 80% para aprovação. Método de Baer

Tópico:
Explica o contexto e a colocação: o quê e o porquê da documentação. Não é evidente por si só ao público.

Questões do Tópico:
O contexto político/legal do planejamento é explicado (exemplo: encontros, ordens estaduais, discussões públicas, questões de alta prioridade)?
===== > Peso desta questão: 1.80

Peso do tópico: Salvar pesos ao sair sem concluir

Há pesos não definidos 9.

Figura 35 - Janela de definição dos pesos dos tópicos

XII Pesos das questões:

Na janela de pesos das questões, apresentada na figura 36, e obtida após pressionado o botão definir pesos das questões da janela definição de pesos dos critérios, o usuário obtém uma série de informações sobre o questão que terá seu peso definido ou alterado. Na mesma janela o usuário acessa também informações sobre o tópico ao qual a questão pertence para facilitar sua decisão.

Informações apresentadas na janela e funcionamento dos seus botões e opções:

- a) Teoria: nome da teoria de avaliação de planos que foi escolhida;
- b) Autor: nome do autor da teoria;
- c) Idioma da teoria: idioma em que a teoria foi escrita;
- d) Ano da teoria: ano em que a teoria foi desenvolvida;
- e) Quantidade de questões: quantidade de questões do método;
- f) Quantidade total de questões: quantidade total de questões do método;
- g) Método usado: descrição do método que foi escolhido;
- h) Idioma do método: idioma em que o método foi escrito;
- i) Questão atual/total: informa o número da questão ao qual a questão apresentada pertence e o número total de questões separadas por uma barra;
- j) Critério: informa a descrição do critério cujas questões terão seus pesos definidos ou alterados, conforme o caso;

k) Tópico: exibe o texto do tópico ao qual a questão cujo peso será definido pertence. Este campo tem como finalidade fornecer informações adicionais para que o usuário possa definir o valor mais apropriado para o peso da questão;

l) Peso do tópico: informa o peso do tópico caso este já tenha sido definido. Este campo tem como finalidade fornecer informações adicionais para que o usuário possa definir o valor mais apropriado para o peso da questão;

m) Questão: exibe o texto da questão que terá o seu peso definido. O usuário pode utilizar a barra de rolagem vertical;

n) Peso da questão: neste campo, o usuário, após analisar as informações apresentadas, deve definir um valor numérico, correspondente ao peso da questão, ou deixá-lo sem valor, caso não haja condições de defini-lo no momento. Ao ser cadastrado um critério de avaliação, todas os pesos estarão automaticamente em branco. Um peso em branco significa que não foi definido um peso para a questão específica. Após definido o peso da questão, e desde que o peso tenha sido salvo - ver item p) *opção salvar respostas ao sair sem concluir*, o mesmo não mais poderá voltar para a situação "em branco". O usuário poderá livremente alterar o peso de uma questão. Se a questão já possuía um peso definido, esse valor será apresentado cabendo ao usuário decidir se irá alterá-lo. Se a questão não possuía um peso definido então o campo estará em branco;

o) Botões voltar e avançar: permitem a navegação seqüencialmente entre as questões. O botão voltar estará desabilitado ao usuário quando estiver na primeira questão. O botão avançar estará desabilitado ao usuário quando estiver na última questão;

p) Opção salvar respostas ao sair sem concluir: esta opção permite ao usuário salvar os pesos das questões não respondidas completamente. Esta opção somente estará habilitada ao usuário caso haja pelo menos uma questão sem peso. Se esta opção estiver habilitada e selecionada, ao clicar no botão sair o sistema atualizará os pesos definidos pelo usuário;

q) Botão concluir: este botão somente estará habilitado caso os pesos tenham sido completamente definidos. Ao clicar no botão concluir o sistema atualizará incondicionalmente os pesos definidos pelo usuário. Caso o usuário não deseje atualizar os pesos deve utilizar o botão sair (quando o botão concluir estiver habilitado, o item *opção salvar respostas ao sair sem concluir* estará desabilitado fazendo com que as respostas não sejam atualizadas ao clicar no botão sair);

r) Botão sair: este botão estará sempre habilitado e seu funcionamento é dependente da *opção salvar respostas ao sair sem concluir*. Dependendo desta opção, os pesos serão ou não atualizados. Quando todos os pesos forem definidos pelo usuário, o botão sair não atualizará as respostas. Para atualizar as respostas nessa situação é necessário utilizar o botão concluir, que apenas nessa situação estará habilitado;

Pesos das Questões

Teoria: Teoria para avaliação geral de planos: uma forma de se fazer planos melhores
Autor: William C. Baer
Idioma da teoria: english (U.S.A)
Ano da teoria: 2004 **Quantidade de tópicos:** 8 **Quantidade total de questões:** 60
Método usado: Método com questões em português da teoria para avaliação geral de planos
Idioma do método: português (Brasil)
Tópico Atual/Total: 1/8 **Questão Atual do Tópico/Total no Tópico:** 1/10 **Questão Atual Total/Total Geral:** 1/60

Critério: Critério com ênfase para questões subjetivas. 80% para aprovação. Método de Baer

Tópico: **Peso do tópico:** 1.60

Explica o contexto e a colocação: o quê e o porquê da documentação. Não é evidente por si só ao público.

Questão:

O contexto político/legal do planejamento é explicado (exemplo: encontros, ordens estaduais, discussões públicas, questões de alta prioridade)?

Peso da questão: 1.80 **Salvar pesos ao sair sem concluir**

Há pesos não definidos 3.

Figura 36 - Janela de definição dos pesos das questões

XIII Parâmetros do relatório de avaliação:

Na janela de parâmetros do relatório de avaliação, apresentada na figura 37, e obtida após pressionado o botão relatórios do menu avaliações de planos, o usuário deve escolher uma teoria, um método, um plano de governo e um ou mais critérios para que o sistema apresente um relatório com os resultados da avaliação. Ao escolher uma teoria e um método, somente os critérios que foram cadastrados pelo usuário para a teoria e método escolhidos serão listados.

A utilização da janela deve ser feita de acordo com as instruções a seguir:

- a) Escolha uma teoria de avaliação: observar as instruções do item VII a) na página 105;
- b) Escolha um método: observar as instruções do item VII b) na página 106;
- c) Escolha o plano de governo que foi avaliado: no terceiro 'combobox' o usuário irá escolher o plano que foi avaliado para exibir o relatório da avaliação. Neste 'combobox' serão apresentados todos os planos de governo cadastrados pelo usuário, independentemente se a avaliação deste plano foi ou não concluída. Os planos de governo listados são apenas os cadastrados pelo usuário que está conectado. Os planos cadastrados por outros usuários não podem ser visualizados;
- d) Escolha o(s) critério(s) cuja(s) análise(s) deseja exibir: no quarto 'combobox' o usuário irá escolher um ou mais critérios que farão parte do relatório com os resultados. Os critérios do próprio sistema também serão apresentados e podem ser escolhidos. Os critérios cadastrados por outros usuários não podem ser utilizados e não são apresentados neste 'combobox'. Para selecionar mais de um critério deve-

se utilizar as teclas de suspensão '*shift*' e '*ctrl*' em associação com o dispositivo apontador ('*mouse*');

e) Botão ok: o usuário confirma todos os dados e solicita ao sistema que exiba o relatório com os resultados da avaliação, mostrado na figura 38. Caso haja sucesso, uma nova janela será aberta. Caso haja algum erro uma mensagem será exibida no rodapé desta janela;

f) Botão cancelar: o usuário solicita ao sistema para abandonar esta janela e voltar para a janela inicial.

Parâmetros do Relatório de Avaliação

Parâmetros do Relatório de Avaliação

Escolha uma teoria de avaliação

1 - Teoria para avaliação geral de planos: uma forma de se fazer planos melhores - William C. Baer - english (U.S.A)

Escolha um método:

1 - Método com questões em português da teoria para avaliação geral de planos - português (Brasil) - 0000-00-00

Escolha o plano de governo que foi avaliado:

Plano diretor da cidade de Planalto - RS

Escolha o(s) critério(s) cuja(s) análise(s) deseja exibir:

1 - Critério que estabelece peso unitário para todas as questões. 80% para aprovação
2 - Critério com ênfase para questões subjetivas. 80% para aprovação. Método de Baer

Ok Cancelar

Todos os critérios existentes foram selecionados. Se desejar ver o relatório com todos os critérios basta pressiona

Figura 37 - Janela de parâmetros do relatório de avaliação

XIV Relatório dos resultados:

Na janela de relatório dos resultados, apresentada na figura 38, e obtida após pressionado o botão ok da janela parâmetros do relatório de avaliação, o usuário obtém o relatório dos resultados da avaliação do plano de governo.

Informações apresentadas na janela e funcionamento dos seus botões e opções:

- a) Teoria: nome da teoria de avaliação de planos que foi escolhida;
- b) Autor: nome do autor da teoria;
- c) Idioma da teoria: idioma em que a teoria foi escrita;
- d) Ano da teoria: ano em que a teoria foi desenvolvida;
- e) Quantidade de tópicos: quantidade de tópicos do método;
- f) Quantidade total de questões: quantidade total de questões do método;
- g) Método usado: descrição do método que foi escolhido;
- h) Idioma do método: idioma em que o método foi escrito;
- i) Resultados: são mostradas, em uma janela de edição de texto, as seguintes informações:

Teoria que foi utilizada;

Método da teoria que foi utilizado para avaliar o plano;

Plano de governo avaliado;

Tabela: apresenta as questões agrupadas por tópicos e, nas colunas encontram-se: a resposta do usuário para aquela questão, e, para cada critério selecionado na janela da figura 37; o peso do tópico; o peso da questão; e, os pontos acumulados. Após apresentar todas as questões do tópico, um subtotal do tópico é apresentado e, após o último tópico um resultado final. Um exemplo de relatório, possuindo dois critérios de avaliação, é mostrado abaixo:

Questão nº	Tópico nº	Resposta	Critério 1			Critério 2		
			Peso Top	Peso Quest	Pontos	Peso Top	Peso Quest	Pontos
1-(1/3)	1/2	Sim	1,1	1,1	1,2	1,1	2,1	2,3
2-(2/3)	1/2	???	1,1	1,1	0,0	1,1	2,1	0,0
3-(3/3)	1/2	Sim	1,1	1,1	1,2	1,1	2,1	2,3
Subtotal do tópico 1:			1,1	--	2,5 (66,7%)	1,1	--	4,7 (66,7%)
4 - (1/2)	2/2	Não	1,2	1,2	0,0	1,2	2,2	0,0
5 - (2/2)	2/2	Sim	1,2	1,2	1,5	1,2	2,2	2,7
Subtotal do tópico 2:			1,2	--	1,5 (50,2%)	1,2	--	2,7 (50,1%)
Resultado final:			--	--	3,9 (59,4%)	--	--	7,3 (59,5%)

Conclusões: para cada critério selecionado na janela da figura 37, será apresentada uma sentença informando se o plano foi aprovado ou não aprovado.

Exemplo:

Critério 1: 3,9 pontos de 6,6 possíveis. 59,4%. Porcentagem mínima do critério: 80,0%. -> Plano reprovado sob este critério. Há questões não respondidas que podem alterar o resultado!

Critério 2: 7,3 pontos de 12,3 possíveis. 59,5%. Porcentagem mínima do critério: 80,0%. -> Plano reprovado sob este critério. Há questões não respondidas que podem alterar o resultado!

Legenda: apresenta uma legenda com os símbolos apresentados neste relatório.

Exemplo:

*** => valor inválido;

??? => o valor não foi fornecido pelo usuário.

Crítérios: identifica os critérios, que na tabela são apresentados com números. A identificação consiste na descrição do critério que foi cadastrada pelo usuário. Veja o exemplo:

Descrição do critério 1: Critério de TESTES número 1 (para o método de testes 1). 80% para aprovação.

Descrição do critério 2: Critério de TESTES número 2 (para o método de testes 1). 80% para aprovação.

Relatório dos resultados

Teoria: Teoria para avaliação geral de planos: uma forma de se fazer planos melhores
Autor: William C. Baer
Idioma da teoria: english (U.S.A)
Ano da teoria: 2004 **Quantidade de tópicos:** 8 **Quantidade total de questões:** 60
Método usado: Método com questões em português da teoria para avaliação geral de planos
Idioma do método: português (Brasil)

Resultados:

Teoria utilizada: 1 - Teoria para avaliação geral de planos: uma forma de se fazer planos melhores
Método utilizado: 1 - Método com questões em português da teoria para avaliação geral de planos
Plano de Governo Avaliado: 2 - Plano diretor da cidade de Planalto - RS

Questão n°	Tópico n°	Resposta	Peso Top	Critério 1			Critério 2		
				Peso Quest	Pontos	Peso Top	Peso Quest	Pontos	
1 - (1/10)	1/8	Sim	1,0	1,0	1,0	1,6	1,8	2,9	
2 - (2/10)	1/8	Não	1,0	1,0	0,0	1,6	1,2	0,0	
3 - (3/10)	1/8	Não	1,0	1,0	0,0	1,6	1,5	0,0	
4 - (4/10)	1/8	Sim	1,0	1,0	1,0	1,6	1,7	2,7	
5 - (5/10)	1/8	Sim	1,0	1,0	1,0	1,6	1,6	2,6	
6 - (6/10)	1/8	Não	1,0	1,0	0,0	1,6	???	0,0	
7 - (7/10)	1/8	???	1,0	1,0	0,0	1,6	1,8	0,0	
8 - (8/10)	1/8	Sim	1,0	1,0	1,0	1,6	1,6	2,6	

Sair

Figura 38 - Janela do relatório dos resultados

5 Considerações Finais

No projeto do sistema de avaliação de planos do governo foi utilizada a modelagem UML como forma de representar uma possível solução para o problema de automatizar a sistematização de avaliação de planos proposta por Baer. O modelo de desenvolvimento incremental definiu a forma de desenvolvimento do sistema projetado. Foram definidos os requisitos iniciais para o sistema e criados algoritmos computacionais que, pretende-se, atendam a esses requisitos. A próxima etapa é avaliar o sistema, verificando se as suas funcionalidades atendem aos requisitos propostos (LARMAN, 2001; SILVA, 2001).

CAPÍTULO 4 - TESTE DO SISTEMA

1 Introdução

A atividade de avaliar planos de governo, enquanto documento, foi sistematizada por William Baer em um conjunto de sessenta questões que, segundo ele, podem avaliar a qualidade de planos ainda em sua fase de sua formulação (BAER, 1997, p. 329). Estes critérios também podem ser utilizados para avaliar planos de governo que já foram colocados em prática. É o que fez Maria de Lourdes Queirós, quando avaliou os planos plurianuais (PPAs) analisados segundo a formação de egressos, no âmbito do Conselho Nacional do Desenvolvimento Científico e Tecnológico (CNPq) (QUEIRÓS, 2001).

Queirós (2001) aplicou a metodologia de Baer em dois planos plurianuais de Ciência e Tecnologia (C&T) e comparou os resultados aos efetivamente obtidos nos instrumentos do CNPq. Para aplicar a metodologia de Baer, Queirós montou oito planilhas eletrônicas, cada qual representando uma das oito classificações dos critérios de Baer. Nas linhas da primeira coluna de cada planilha elencou as questões, propostas por Baer, que são pertencentes à classificação correspondente ao tópico da planilha em questão. Nas linhas da segunda coluna de cada planilha as respostas respectivas àquelas questões correspondentes à avaliação do primeiro PPA e, nas linhas da terceira coluna as respostas correspondentes à avaliação do segundo PPA (QUEIRÓS, 2001, p. 77-83).

As respostas fornecidas por Queirós podem servir de base para os testes do sistema, e se espera obter os mesmos resultados obtidos pela pesquisadora em seu trabalho.

Além das duas avaliações de planos feitas por Queirós, que irá testar o sistema em situações reais, o sistema também será testado em duas situações extremas para verificar se o seu comportamento corresponde ao esperado. As situações extremas em que o sistema será testado corresponde às avaliações que recebem apenas um tipo de resposta, ou seja, uma avaliação recebe apenas respostas positivas, enquanto a outra recebe apenas respostas negativas.

2 Testes

I Avaliação dos PPA's

Segundo Queirós (2001, p. 19-21) as ações governamentais durante as últimas cinco décadas nem sempre se mostraram articuladas no que diz respeito a integração das políticas de diferentes áreas. Para integrar as ações governamentais nas diferentes áreas de governo a Constituição Federal de 1988 cria os Planos Plurianuais. Os Planos Plurianuais procuram associar o Orçamento da União aos compromissos e estratégias do governo para possibilitar a compreensão dos rumos propostos para cada área do governo. O primeiro Plano Plurianual, elaborado pelo governo vencedor das eleições de 1989, vigorou de 1991 a 1995 e o segundo PPA vigorou de 1996 a 1999.

Assim como as demais áreas governamentais a política de Ciência e Tecnologia (C&T) integra os Planos Plurianuais. (QUEIRÓS, 2001, p. 19). No âmbito da C&T, "o texto do primeiro PPA ressalta a necessidade de consolidação da base técnico-científica nacional, a fim de que o País pudesse atingir padrões de qualidade internacional nas atividades de pesquisa" (QUEIRÓS, 2001, p. 21). Ainda no âmbito da C&T, "o objetivo geral do segundo PPA foi a capacitação científica e tecnológica,

também considerada fator essencial para viabilizar o projeto de desenvolvimento sócio-econômico sustentável do País" (QUEIRÓS, 2001, p. 25). Queirós (2001, p. 28) ressalta que uma das finalidades de seu trabalho "[...] consiste em verificar que procedimentos são recomendados na literatura internacional, de forma a instrumentalizar a avaliação e, principalmente, sistematizar a construção de planos governamentais [...]".

Após a aplicação da metodologia de Baer aos dois primeiros Planos Plurianuais, Queirós (2001, p. 99) observou que estes PPAs obtiveram um desempenho fraco, portanto, o sistema de avaliação de planos de governo deve apontar um rendimento percentual baixo para esses PPAs, caso se forneçam as mesmas respostas dadas por Queirós em sua avaliação.

II Registro no sistema

Foram registradas quatro avaliações com a finalidade de testar o sistema. Duas avaliações testam o sistema em situações extremas, nesse caso, as respostas foram sempre positivas no primeiro teste e sempre negativas no segundo. Além destas avaliações, foram registradas no sistema as avaliações dos dois primeiros PPAs efetuadas por Queirós (2001, p. 77-83). A tabela a seguir apresenta as respostas ao questionário de Baer (1997, p. 343-6) para cada uma das quatro avaliações propostas. Na tabela, a resposta '???' , no caso das avaliações dos PPAs,

significa que não existe informação suficiente para responder a questão⁵ (QUEIRÓS,2001, p. 77-83).

Tabela 25 - respostas das avaliações para o teste do sistema

Questão nº	Tópico nº	Respostas para o teste nº 1	Respostas para o teste nº 2	Respostas para o PPA nº 1	Respostas para o PPA nº 2
1	1	Sim	Não	Sim	Sim
2	1	Sim	Não	Sim	Sim
3	1	Sim	Não	Sim	Sim
4	1	Sim	Não	Sim	Sim
5	1	Sim	Não	Sim	Sim
6	1	Sim	Não	Sim	Sim
7	1	Sim	Não	Não	Não
8	1	Sim	Não	Não	Não
9	1	Sim	Não	???	???
10	1	Sim	Não	Não	Não
11	2	Sim	Não	Não	Não
12	2	Sim	Não	Não	Não
13	2	Sim	Não	???	???
14	2	Sim	Não	Sim	Sim
15	2	Sim	Não	???	???
16	2	Sim	Não	???	Sim
17	2	Sim	Não	Não	Sim
18	2	Sim	Não	Não	???
19	2	Sim	Não	???	???
20	3	Sim	Não	Sim	Sim
21	3	Sim	Não	Não	Não
22	3	Sim	Não	Não	Não
23	3	Sim	Não	Não	Não
24	3	Sim	Não	Não	Não
25	3	Sim	Não	???	???
26	3	Sim	Não	???	???
27	4	Sim	Não	???	???
28	4	Sim	Não	Não	Não
29	4	Sim	Não	???	???
30	4	Sim	Não	???	???
31	4	Sim	Não	???	???
32	4	Sim	Não	???	???
33	4	Sim	Não	???	???
34	5	Sim	Não	???	???
35	5	Sim	Não	???	???
36	5	Sim	Não	???	???
37	5	Sim	Não	Sim	Sim
38	5	Sim	Não	Não	Não

⁵ No seu trabalho, Queirós (2001, p. 77-83), convencionou utilizar o número 3 para indicar que não existia informação suficiente no texto do plano para responder a questão correspondente, o número 1 para respostas afirmativas e o número 2 para respostas negativas.

Questão nº	Tópico nº	Respostas para o teste nº 1	Respostas para o teste nº 2	Respostas para o PPA nº 1	Respostas para o PPA nº 2
39	5	Sim	Não	???	???
40	5	Sim	Não	Não	Não
41	5	Sim	Não	Sim	Sim
42	5	Sim	Não	???	???
43	6	Sim	Não	???	???
44	6	Sim	Não	???	???
45	6	Sim	Não	Não	Não
46	6	Sim	Não	Não	Não
47	6	Sim	Não	Não	Sim
48	7	Sim	Não	Sim	Sim
49	7	Sim	Não	???	???
50	7	Sim	Não	???	???
51	7	Sim	Não	Sim	Sim
52	7	Sim	Não	Sim	Sim
53	7	Sim	Não	Não	Não
54	8	Sim	Não	Sim	Sim
55	8	Sim	Não	Sim	Sim
56	8	Sim	Não	Não	Sim
57	8	Sim	Não	Sim	Sim
58	8	Sim	Não	Sim	Sim
59	8	Sim	Não	Não	Não
60	8	Sim	Não	???	???

III Critérios de análise

Tendo em vista que, na avaliação de Queirós (2001, p. 77-83), algumas questões não puderam ser respondidas, foram estabelecidos três conjuntos de pesos (critérios) para as questões aplicadas. O primeiro critério, denominado "critério com pesos unitários", possui todos os pesos valendo um (peso unitário) para todas as questões. O segundo critério, denominado "critério para avaliação do PPA n. 1", foi estabelecido da seguinte forma: pesos unitários, na avaliação do primeiro PPA, para questões cujas respostas foram "sim" ou "não"; peso zero, na avaliação do primeiro PPA, para questões que não puderam ser respondidas. O terceiro critério, denominado "critério para avaliação do PPA n. 2", foi estabelecido da seguinte forma: pesos unitários, na avaliação do segundo PPA, para questões cujas respostas foram "sim" ou "não"; peso zero, na avaliação do segundo PPA, para questões que não puderam ser respondidas.

IV Resultados obtidos

Após aplicar o sistema os resultados obtidos foram os seguintes:

Teste número 1 (respostas "sim"): o relatório para este teste foi configurado com o critério de pesos unitários. A conclusão apresentada pelo sistema é de que o plano apresentou 60 pontos dos 60 possíveis, com um rendimento de 100%.

Teste número 2 (respostas "não"): o relatório para este teste também foi configurado com o critério de pesos unitários. A conclusão apresentada pelo sistema é de que o plano apresentou 0 ponto de 60 possíveis, com um rendimento de 0%.

Teste número 3 (PPA nº 1): o relatório para este teste foi configurado com o critério para avaliação do PPA n. 1. A conclusão apresentada pelo sistema é de que o plano apresentou 17 pontos dos 37 possíveis, com um rendimento de 45,9%. Tal análise está de acordo com a avaliação efetuada por Queirós (2001, p. 77-83), que considerou o desempenho do primeiro PPA baixo (QUEIRÓS, 2001, p. 99)

Teste número 4 (PPA nº 2): o relatório para este teste foi configurado com o critério para avaliação do PPA n. 2. A conclusão apresentada pelo sistema é de que o plano apresentou 21 pontos dos 37 possíveis, com um rendimento de 56,8%. Tal análise também está de acordo com a avaliação efetuada por Queirós (2001, p. 77-83), que considerou o desempenho do segundo PPA baixo (QUEIRÓS, 2001, p. 99)

CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS

Mesmo tendo sido descrito o domínio do problema da sistematização da avaliação de planos de governo através de uma pesquisa bibliográfica do tema, não há certeza de que os requisitos projetados para o sistema atendam as necessidades reais de um usuário final. Essas idéias vão ao encontro das idéias propostas por Sommerville (2003), que mostra, nesse caso, a possibilidade do desenvolvimento inicial de um protótipo. Este protótipo seria uma versão inicial do sistema, com a finalidade de avaliar se as decisões de projetos satisfazem aos interesses dos usuários. O que se deseja evitar é que, mesmo sendo bem projetado, um sistema possa vir a fracassar caso identifique-se, posteriormente ao desenvolvimento, necessidades não contempladas durante a fase de projeto. Sommerville (2003) defende que o desenvolvimento de um protótipo demanda menos tempo, diminui os custos e oferece a possibilidade de se conhecer novos conceitos que não ficaram claros. O uso de um sistema traz reais necessidades ao usuário que não podem ser vislumbradas durante o projeto. Ainda, de acordo com Sommerville (2003), o protótipo apóia duas atividades: o levantamento de requisitos e a validação de requisitos.

O Sistema de Informação para apoio à avaliação de planos de governo proposto neste trabalho visa ser apenas um protótipo, um modesto início, se não houver outro, na tarefa de auxiliar, com sistemas de informação, avaliadores de planos de governo. Como tal, não foi dada a ênfase necessária ao seu projeto de desenvolvimento, tendo em vista a necessidade de tê-lo funcionando para verificar a sua viabilidade e quais são realmente os requisitos do usuário que emergem com o seu uso (SOMMERVILLE, 2003). As possibilidades de agregar-lhe novas

funcionalidades, associá-lo ou incorporá-lo a outros sistemas correlatos, ou mesmo redefini-lo totalmente, são incontáveis. A continuidade do desenvolvimento deste sistema, ou a criação de um novo, mais completo, seguro, funcional e de acordo com padrões de engenharia de software dependem do interesse e necessidade de utilização por parte dos profissionais que se dedicam ao desenvolvimento e avaliação de planos de governo.

Baer propõe que novos critérios de avaliação sejam desenvolvidos, inclusive por aqueles que estão desenvolvendo planos, como se vê na transcrição:

**[...]. Eu discuto que os planejadores devem criar critérios para este estágio, de modo que entendam completamente o que estão fazendo enquanto preparam seus planos. É apresentado um quadro composto de critérios apropriados para ilustrar o conceito de avaliação. O que não se configura, contudo, como uma receita aprovada, que se seguida passo a passo irá garantir a competência profissional. Ao contrário, pretendem ilustrar como e porque grupos individuais de planejadores devem desenvolver tais critérios eles mesmos a cada novo planejamento.
(Baer, 1997, p. 330)**

**Cada vez que um plano é preparado, quem o prepara deve especificar os critérios pelos quais espera que ele seja avaliado: o esforço inevitavelmente aprimora seu entendimento em relação ao que está fazendo.
(Baer, 1997, p. 336)**

**A intenção de ambos é ser uma lista de possíveis considerações durante a preparação do planejamento e para estimular os autores do planejamento a projetarem variantes e acréscimos que achem pertinentes aos seus planos. A meta é desenvolver uma reflexão, formal e consciente, do quê se esta fazendo no planejamento *enquanto ele esta sendo feito*.
(Baer, 1997, p. 336)**

De acordo com estas propostas de Baer, ficam como sugestões para trabalhos futuros: a pesquisa bibliográfica de novos critérios para a avaliação de planos de governo; disponibilizar a funcionalidade que permite ao usuário incluir no

sistema outros questionários de avaliação de planos de governo; e, implementar no sistema suporte para avaliação do mesmo plano por várias pessoas, oferecendo uma análise estatística das diferentes avaliações.

BIBLIOGRAFIA

1 Livros, Periódicos, Apostilas, Artigos e Monografias:

- ABREU, Pedro Felipe de, ABREU, Aline França de. Sistemas de Informações Gerenciais - uma abordagem orientada a negócios. [Florianópolis] Ed. IGTI - UFSC, 2000. 158p. (Apostila).
- BAER⁶, William C. **General Plan Evaluation Criteria**: An approach to making better plans. **Journal of the American Planning Association**. Chicago, Vol. 63, Issue 3, p. 329-344, Summer 1997. Disponível na Internet. <http://www.scielo.br>. 08 out. 2004.
- BRUTON, Michael J. (Ed.) The Spirit and Purpose of Planning. [London]: Hutchinson, p. 7, Apr 1974.
- CARDOSO, Fernando Henrique. O homem e o novo cenário geopolítico internacional. In: CICLO UNIVERSO DO CONHECIMENTO, 2004, [São Marcos]. **O homem no mundo contemporâneo — impasses e perspectivas**, ago 2004., p 8. Disponível na internet: <http://www.universodoconhecimento.com>. 06 out. 2004.
- GIL, Antônio Carlos. Como elaborar projetos de pesquisa. 3.ed. São Paulo. Ed. Atlas, 1991.
- HEUSER, Carlos Alberto. Projeto de banco de dados. 4. ed. Porto Alegre. Ed. Sagra Luzzatto, 2001.
- LARMAN, Craig. **Utilizando UML e padrões**: uma introdução à análise e ao projeto orientados a objetos. Ed. Bookman, 2001.
- MELLOR, S. Automatic code generation from UML models. 1999. Disponível na internet: <http://www.projtech.com/pdfs/autocg-1.pdf>. 08 out. 2001.
- PENDALL, Rolf. Municipal Plans, State Mandates, and Property Rights: Lessons from Maine. **Journal of Planning Education and Research**. Vol. 21, No. 2. p.154-165, 2001.
- QUEIRÓS, Maria de Lourdes. **Avaliação de Planos de Governo**: os planos plurianuais analisados segundo a formação de egressos, no âmbito do CNPq. Florianópolis, 2001. 104 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001. Disponível na Internet. <http://www.stela.ufsc.br>. 21 nov. 2003.

⁶ Apesar do artigo publicado no Journal of the American Planning Association possuir 16 páginas numeradas de 329-344, o artigo disponibilizado possui 20 páginas. Tal diferença deve ser dar em função do tamanho da folha do jornal ser diferente do tamanho da folha A4 na qual o artigo foi disponibilizado.

SACCONI, Luiz Antonio. Dicionário essencial da língua portuguesa. São Paulo. Ed. Atual, 2001, p. 721.

SILVA, Ricardo Pereira. **Engenharia de software**. Florianópolis : Universidade Federal de Santa Catarina, 2001. 50f. Apostila utilizada durante aulas de Engenharia de Software, realizada em 2001, em Florianópolis.

SOMMERVILLE, Ian. **Engenharia de Software**. 6.ed. São Paulo: Ed. Addison-Wesley, 2003. 592 p.

2 Aplicativos:

Institute for Human and Machine Cognition. **CmapTools**. Versão 3.5.
<http://cmap.ihmc.us>

MySQL AB. **MySQL**. Versão 4.1. <http://dev.mysql.com/downloads/mysql/4.1.html>

Sun microsystems. **Forte for Java**. Versão 4. CD.

Sun microsystems. **J2SE 1.4 SDK**. Versão 1.4. CD.

The Regents of the University of Califórnia. **ArgoUML**. Versão v0.16.1.
<http://argouml.tigris.org>.

Visual Paradigm. **Visual Paradigm for UML Professional Edition**. Versão 3.2.
<http://www.visual-paradigm.com/>

APÊNDICE A - ARTIGO

**Desenvolvimento do Sistema de Informação para Apoio à Análise da
Qualidade de Planos de Governo com Base na Metodologia de Baer
(1997)**

Dorival Silva dos Anjos

**Departamento de Informática e de Estatística
Universidade Federal de Santa Catarina**

Resumo

Um dos principais fatores de sucesso no planejamento público está na qualidade dos documentos que expressam este planejamento (i.e., em seu plano). Recentemente surgiu metodologia que propõe que a avaliação de planos públicos ocorra durante a formulação do mesmo. Trata-se da Metodologia de William Baer, proposta em 1997. Embora referenciado em diversos projetos públicos, não se verificou disponibilidade de um sistema de informação que venha a sistematizar o apoio aos gestores na atividade de análise de seu planejamento. No presente artigo propõe-se a concepção de um sistema de informação com essa finalidade. Para verificar a consistência de resultados produzidos pelo sistema, avaliou-se um plano público.

Palavras chave: Sistemas de Informação, avaliação de planos de governo.

Introdução

Os planos de governo são muito importantes para o desenvolvimento da sociedade, segundo Dalton, "... o planejamento permanece como uma das principais ferramentas do planejador para influenciar o crescimento futuro e o desenvolvimento" (1989 apud BAER, 1997, p.329). Essa afirmação deixa clara a importância que os planos de governo têm para o desenvolvimento social e o

crescimento, que são indispensáveis (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Os planos de governo sugerem o uso racional dos recursos e o alcance dos objetivos para os quais foi desenvolvido. O êxito no cumprimento dos objetivos depende da capacidade do plano abranger e propor soluções para as diversas situações e problemas que possam surgir após a sua implantação (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001). O sucesso de um plano depende da qualidade desse plano (BAER, 1997). Nesse contexto, uma questão que naturalmente se coloca é como definir a qualidade de um plano de governo? Como forma de propor uma resposta para esta questão, pesquisadores vem desenvolvendo métodos de avaliações de planos em suas diversas fases, visando a melhoria desses planos, através destas avaliações (QUEIRÓS, 2001). Um desses pesquisadores, William Baer, professor de Planejamento Urbano e Desenvolvimento na Escola de Planejamento e Desenvolvimento Urbano da Universidade Southern Califórnia, desenvolveu uma teoria de avaliação de planos de governo. Apesar da aplicação da teoria desenvolvida por Baer poder ser apoiada por um Sistema de Informação, este não foi encontrado. Parece haver uma lacuna a ser preenchida nesta área. Dessa forma,

não só a execução de avaliações é dificultada, como a própria divulgação da prática de análise de planos de governo se mostra mais distante de gestores públicos.

Os objetivo deste artigo é propor o desenvolvimento de um Sistema de Informação para apoio à análise da qualidade de planos de governo com base na metodologia de Baer (1997), devendo para isso:

Estudar a metodologia de avaliação de planos de governo desenvolvida por Baer (1997);

Desenvolver um sistema de domínio público, o qual implementa a metodologia estudada, e disponibilizar esse sistema;

Aplicar o sistema, com a finalidade de testa-lo, apresentando os resultados.

O trabalho proposto mostra-se relevante porque a concepção, o desenvolvimento, e a divulgação de um sistema com essa natureza venham tanto apoiar gestores que praticam a tarefa da análise de planos governamentais como permitir que a prática de avaliação de planos seja mais facilmente conhecida por estes gestores públicos. Além disso, o sistema tem potencial para servir de ponto de partida para desenvolvimento de outros sistemas de informação aplicados à avaliação de planos de governo, observando-se os requisitos que não foram atendidos. Além destas, outra contribuição está no fato de a pesquisa ter verificado a insipiência da área de avaliação qualitativas de planos no País.

Planos de governo

Define-se plano de governo, neste artigo, combinando-se as definições de Bruton (1974, p. 7) com as do dicionário essencial da língua portuguesa (SACCONI, 2001), como sendo um procedimento por meio do qual uma instituição governamental faz planos ou preparativos antecipadamente, os quais determinam uma linha de conduta governamental, formal, precisa, traçada por etapas para atingir determinado

objetivo. Mais especificamente, um plano de governo pode ser feito para se concretizar uma construção ou obra, um plano econômico ou ainda definir ações sociais, diretrizes e ações para a safra agrícola, educação, moradia, crescimento urbano dentre outros. Os planos de governo são desenvolvidos com a finalidade de atender demandas do poder público (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Desenvolver um plano de governo não é simples. Esta tarefa apresenta uma grande quantidade de elementos subjetivos, interesses de diversos setores da sociedade, algumas vezes divergentes, e depende de fatores externos ao planejamento em si, que nem sempre são controláveis. O trabalho de avaliar um plano de governo, portanto, não é uma tarefa fácil (BAER, 1997; BRUTON, 1974; PENDALL, 2001; QUEIRÓS, 2001).

Segundo Alexander e Faludi (1989, apud BAER 1997, p. 329), "se o planejamento precisa ter alguma credibilidade como uma disciplina ou uma profissão, os critérios da avaliação devem permitir um julgamento real da eficácia do planejamento: o planejamento bom deve ser distinguível do ruim."

A avaliação de planos de governo faz-se necessária para se definir, antes de se colocar em prática, se um plano é bom ou não. É grande a complexidade que envolve o desenvolvimento de um plano de governo, como visto anteriormente, também é considerada difícil a tarefa de se definir se um plano é ou não bom. Muito mais difícil ainda torna-se criar critérios genéricos que possam avaliar qualquer plano (BAER, 1997). A tentativa de Baer de solucionar esse problema é resumida na seção a seguir.

Metodologia de Baer

Baer (1997, p. 335) afirma que os critérios apropriados para avaliar um planejamento devem estar implícitos na própria concepção do planejamento, ou seja, os critérios dependem do próprio plano, ou mais especificamente, de sua categoria. Ainda segundo ele, cada vez

que um plano é preparado, quem o prepara deve especificar os critérios pelos quais deseja que ele seja avaliado. Para criar um critério de avaliação genérica, Baer faz uma revisão das abordagens de desenvolvimento de planos de governo para enfatizar quais critérios, de acordo com os tipos de planejamento, são apropriados para a avaliação. Os seja, os critérios de avaliação mudam de acordo com o estilo ou método utilizado para o desenvolvimento do planejamento. Oito abordagens de desenvolvimento de planejamentos foram estudadas por Baer para o desenvolvimento de critérios corretivos (BAER, 1997).

Segundo Baer, a avaliação de planos de governo pode ser feita em diferentes estágios do planejamento (BAER, 1997). Baer apresenta uma proposta de avaliação de planos de governo que é feita enquanto o plano está sendo formulado e resolveu chamar esta fase da avaliação de análise do planejamento em processo (BAER, 1997, p. 338). A análise do planejamento em processo tem a finalidade de verificar "a habilidade profissional: o planejamento enquanto documentação de comunicação metodológica, raciocínio e conteúdo: ele opera no momento de por o planejamento em ação" (BAER, 1997, p. 338).

Um planejamento, enquanto documento, deve possuir atributos essenciais que garantam a sua qualidade. Os critérios tem a finalidade de verificar a existência desses atributos. Um exemplo de atributo desejável de um plano é que o mesmo não possua erros por omissão (BAER, 1997, p. 338-9). Se um critério exige que se explique o porquê das decisões de um plano, pode reduzir essa possibilidade. Um a forma de se tentar impedir a omissão seria definir os critérios para um mal plano e reprimi-los. Dentro dessa linha, segundo Baer, Hansen (1968, apud in BAER, 1997 p. 339) argumenta que a inviabilidade, estilo vagaroso, obsolência, pouca abrangência, falta de realidade ou negligência indicaria um plano ao qual se deve ter pouco respeito. Mas acredita que para estabelecer critérios deve-se tomar uma atitude mais positiva em vez de se fazer apenas

advertências. Os critérios positivos, segundo ele, devem especificar o que os planejamentos devem ter. As afirmações expressas em um plano devem deixar evidências de que de fato elas são verdades (BAER, 1997).

Uma lista dos critérios, segundo Baer, é enfadonha, porém se torna importante no momento de sua utilidade, assim como uma lista de tarefas domésticas, compara. Os critérios são muito importantes para quem está preparando ou avaliando um planejamento. Os critérios, segundo ele não são definitivos, são possíveis considerações que podem ser utilizadas durante a preparação do planejamento. Os autores do planejamento podem projetar variantes para os critérios, e, também podem fazer acréscimos (BAER, 1997, p. 339).

Baer desenvolveu uma lista de critérios expressos em forma de questionário com sessenta questões. As questões são divididas em oito classificações. As classificações agrupam as questões em torno de um tópico ou assunto. Baer sugere a avaliação do documento que apresenta o plano de governo. A avaliação seria a resposta do questionário sobre o documento e seu conteúdo (BAER, 1997). Esta avaliação pode ser automatizada através de um Sistema de Informação cujo desenvolvimento é apresentado na seção a seguir.

Sistema Proposto

Os sistemas de informação, quando corretamente projetados para uma tarefa, possuem a capacidade de processar e gerar novas informações, dirigindo o usuário por um caminho correto. A administração da informação deixa o usuário livre para se preocupar exclusivamente com a atividade fim, que é a avaliação do plano.

De acordo com Abreu & Abreu, um Sistema de Informação (SI), pode ser definido como: "[...] um conjunto de procedimentos que coletam (ou recuperam), processam, armazenam e disseminam informações para o suporte

na tomada de decisões [...]” (ABREU & ABREU, 2000, p. 14)

Sistemas de Informação, portanto, são configurados em domínios de aplicação específicos. Um domínio é caracterizado pela área, pelo setor ou pelo campo de utilização do sistema. A avaliação de planos de governo constitui um domínio de aplicação. Como tal, é necessário evidenciar em que medida as tarefas realizadas nesse domínio são passíveis de formulação como procedimentos modeláveis e programáveis em instruções computacionais.

Uma análise da Metodologia de Baer evidencia que esse é o caso. O planejador não acessa o plano de uma única vez, e sim por etapas, fazendo avaliações parciais. Após ser avaliado, um plano pode precisar de correções e ser novamente avaliado. Esse ciclo pode se repetir inúmeras vezes. Visto que um Sistema de Informação tem a capacidade de armazenar e recuperar informações, o planejador pode se valer desse recurso para armazenar as informações de uma avaliação parcial, e, após alterar o plano, em qualquer tempo, recuperar as informações da avaliação anterior, reaplicando a avaliação apenas nos itens que não estão satisfatórios. Um segundo benefício é o processamento das informações armazenadas, gerando novas informações, e, por último, estas informações podem ser disseminadas pelo sistema atendendo as necessidades do planejador ou outra pessoa interessada (BAER, 1997; ABREU & ABREU, 2000).

Ainda que avaliar planos de governo seja uma tarefa intrinsecamente subjetiva, pois dependente do parecer humano, não se pode descartar a possibilidade de sistematização das tarefas de natureza procedural. Elas surgem justamente em metodologias que prescrevem passos à avaliação de planos, como ocorre com a Metodologia de Baer.

Sistematizar a avaliação de planos não é uma tarefa dos profissionais da Tecnologia da Informação e sim de pesquisadores ligados à área de planejamento. A sistematização, no

entanto, evidencia tarefas processuais que podem ser modeladas na forma de procedimentos computacionais, que dão base a um sistema de informação. O estudo da Metodologia de Baer mostrou que a avaliação de planos pode ser realizada como tarefa sistêmica e, mais especificamente, que esse modelo pode ser base para a concepção e desenvolvimento de um Sistema de Informação, conforme proposto no presente trabalho.

O sistema proposto visa mostrar que é possível apoiar o avaliador na avaliação de planos, naquilo em que couber e, como, um sistema precursor, servir de estímulo, base e/ou modelo para outros sistemas que possam ser desenvolvidos no futuro visando objetivos semelhantes. Não se pretende ser exaustivo na proposição de uma solução informatizada ao apoio à avaliação de planos e sim abrir caminho para o desenvolvimento de novos sistemas que apresentem um aprofundamento maior, tanto no que se refere à metodologia de avaliação de planos como na utilização de outras abordagens sistêmicas como base de concepção de SI.

Segundo Silva (2001), é necessário uma abordagem metódica de desenvolvimento, manutenção e gerenciamento ao longo da existência de um software. Algumas metodologias foram criadas com o objetivo de darem suporte para a análise metódica das etapas envolvidas no desenvolvimento de um sistema usando programação orientada a objetos.

De acordo com Larman (2000), para se desenvolver um Sistema de informação, faz-se necessário, primeiro, que o problema seja descrito, deixando claro o que é este problema, e, em segundo lugar, definir os requisitos, ou seja, o que o sistema deve fazer, para, posteriormente, desenvolver o sistema, através da codificação, em uma ou mais linguagens de programação.

No planejamento e elaboração do problema é necessário definir os conceitos

do problema, os requisitos do sistema e descrever os seus casos de uso.

Larmann (2000, p. 139), acredita que a consistência do significado e a compreensão compartilhada dos termos são muito importantes no desenvolvimento de uma aplicação.

Dessa forma, foi criado um mapa conceitual que mostra o relacionamento entre os conceitos do problema. Pode-se dizer que estes conceitos são provisórios e podem ser modificados com o



Figura 1 - Mapa conceitual do problema

amadurecimento das idéias propostas por Baer. O mapa conceitual é apresentado na figura 1.

Uma outra atividade importante na fase de análise do problema é a especificação dos requisitos do sistema. Segundo Larman, a especificação de requisitos correta e abrangentes são a essência de um projeto de sucesso. Os requisitos descrevem as necessidades ou os desejos para um produto. Larman recomenda cinco artefatos na fase de requisitos: um texto com a visão geral, clientes, objetivos, funções do sistema e atributos do sistema (LARMAN, 2000, p. 59).

Visão geral do sistema: o sistema de apoio à avaliação de planos de governo deve automatizar, no que couber, a avaliação de planos de governo. Dessa forma, o sistema

deve dar suporte para que o usuário possa responder questões de avaliação, atribuir pesos para essas questões e receber, como resultado da interação do questionário e dos pesos, um relatório preliminar de avaliação. Como a metodologia desenvolvida por Baer classifica as questões, essas devem ser apresentadas ao usuário dentro de sua classificação e juntamente a esta.

Clientes: desenvolvedores de planos de governo ou outra pessoa que deseje avaliar um plano de governo. São os usuários.

Objetivos: automatizar, no que couber, a aplicação do questionário de avaliação de planos de governo da metodologia de Baer, ou outras equivalentes, para dar apoio à avaliação de planos de governo;

Funções do sistema: o sistema deve: armazenar nomes de teorias de avaliação de planos de governo; armazenar metodologias de avaliação de planos de governo vinculados a uma teoria; armazenar tópicos destas metodologias de avaliação; armazenar questões destas metodologias de avaliação classificadas em tópicos; controlar qual usuário está utilizando o sistema; permitir a qualquer pessoa cadastrar a si como usuário; permitir ao usuário conectar-se ao sistema para utilizar as funções; permitir ao usuário cadastrar planos de governo; permitir ao usuário cadastrar critérios com pesos para os tópicos e as questões de uma dada metodologia; permitir ao usuário avaliar planos de governo respondendo às questões apresentadas pelo sistema (avaliação); permitir ao usuário armazenar a avaliação; elaborar para o usuário um relatório onde os critérios são aplicados aos planos de governo; efetuar outras funções

acessórias que tornem as funções descritas possíveis;

Atributos do sistema: facilidade de uso; interface gráfica; independente de plataforma de software/hardware; utilização gratuita.

Segundo Larman, para uma melhor compreensão dos requisitos, a descrição narrativa dos processos do domínio, chamada de casos de uso, se mostra uma técnica excelente (2000, p. 66). Um caso de uso é um documento que descreve a seqüência ações de um ator, que é um agente externo utilizando um sistema para a execução de um processo (JACOBSON, 1992, apud in LARMAN 2000, p. 68). Como uma descrição de todos os casos de uso pode se tornar muito longa, somente os casos de uso de alto nível e essenciais à compreensão do domínio serão abordados aqui. Os casos de uso consideram o sistema como sendo a fronteira dos casos.

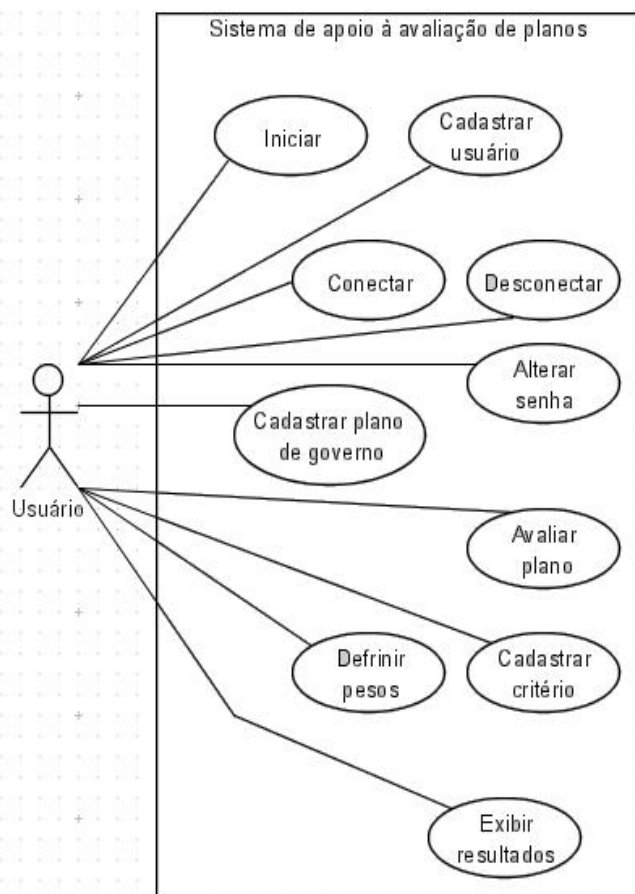


Figura 2 - Diagrama de casos de uso

A notação UML padroniza a representação dos casos de uso de maneira gráfica. A figura 2 apresenta o diagrama de casos de uso do sistema de acordo com a notação UML.

No projeto do sistema de avaliação de planos do governo foi utilizada a modelagem UML como forma de representar uma possível solução para o problema de automatizar a sistematização de avaliação de planos proposta por Baer. O modelo de desenvolvimento incremental definiu a forma de desenvolvimento do sistema projetado. Foram definidos os requisitos iniciais para o sistema e criados algoritmos computacionais que, pretende-se, atendam a esses requisitos. A próxima etapa é avaliar o sistema, verificando se as suas funcionalidades atendem aos requisitos propostos (LARMAN, 2001; SILVA, 2001).

Aplicação

A atividade de avaliar planos de governo, enquanto documento, foi sistematizada por William Baer em um conjunto de sessenta questões que, segundo ele, podem avaliar a qualidade de planos ainda em sua fase de sua formulação (BAER, 1997, p. 329). Estes critérios também podem ser utilizados para avaliar planos de governo que já foram colocados em prática. É o que fez Maria de Lourdes Queirós, quando avaliou os planos plurianuais (PPAs) analisados segundo a formação de egressos, no âmbito do Conselho Nacional do Desenvolvimento Científico e Tecnológico (CNPq) (QUEIRÓS, 2001).

Queirós (2001) aplicou a metodologia de Baer em dois planos plurianuais de Ciência e Tecnologia (C&T) e comparou os resultados aos efetivamente obtidos nos instrumentos do CNPq. Para aplicar a metodologia de Baer, Queirós montou oito planilhas eletrônicas, cada qual representando uma das oito classificações dos critérios de Baer. As respostas fornecidas por Queirós (2001) podem servir de base para os testes do sistema, e se espera obter os mesmos

resultados obtidos pela pesquisadora em seu trabalho.

Além das duas avaliações de planos feitas por Queirós, que irá testar o sistema em situações reais, o sistema também será testado em duas situações extremas para verificar se o seu comportamento corresponde ao esperado. As situações extremas em que o sistema será testado corresponde às avaliações que recebem apenas um tipo de resposta, ou seja, uma avaliação recebe apenas respostas positivas, enquanto a outra recebe apenas respostas negativas.

Após aplicar o sistema os resultados obtidos foram os seguintes:

Teste número 1 (respostas "sim"): a conclusão apresentada pelo sistema é de que o plano apresentou 60 pontos dos 60 possíveis, com um rendimento de 100%, portanto obteve-se o resultado esperado.

Teste número 2 (respostas "não"): a conclusão apresentada pelo sistema é de que o plano apresentou 0 ponto dos 60 possíveis, com um rendimento de 0%, portanto obteve-se o resultado esperado.

Teste número 3 (PPA nº 1): a conclusão apresentada pelo sistema é de que o plano apresentou 17 pontos dos 37 possíveis, com um rendimento de 45,9%. Tal análise está de acordo com a avaliação efetuada por Queirós (2001, p. 77-83), que considerou o desempenho do primeiro PPA baixo (QUEIRÓS, 2001, p. 99)

Teste número 4 (PPA nº 2): a conclusão apresentada pelo sistema é de que o plano apresentou 21 pontos dos 37 possíveis, com um rendimento de 56,8%. Tal análise também está de acordo com a avaliação efetuada por Queirós (2001, p. 77-83), que considerou o desempenho do segundo PPA baixo (QUEIRÓS, 2001, p. 99)

Conclusões

O Sistema de Informação para apoio à avaliação de planos de governo

proposto neste artigo visa ser apenas um protótipo, um modesto início, se não houver outro, na tarefa de auxiliar, com Sistemas de Informação, avaliadores de planos de governo. Como tal, não foi dada a ênfase necessária ao seu projeto de desenvolvimento, tendo em vista a necessidade de tê-lo funcionando para verificar a sua viabilidade e quais são realmente os requisitos do usuário que emergem com o seu uso (SOMMERVILLE, 2003). As possibilidades de agregar-lhe novas funcionalidades, associá-lo ou incorporá-lo a outros sistemas correlatos, ou mesmo redefini-lo totalmente, são incontáveis. A continuidade do desenvolvimento deste sistema, ou a criação de um novo, mais completo, seguro, funcional e de acordo com padrões de engenharia de software dependem do interesse e necessidade de utilização por parte dos profissionais que se dedicam ao desenvolvimento e avaliação de planos de governo.

Baer propõe que novos critérios de avaliação sejam desenvolvidos, inclusive por aqueles que estão desenvolvendo planos (BAER, 1997, p. 330 e 336). De acordo com estas propostas de Baer, ficam como sugestões para trabalhos futuros: a pesquisa bibliográfica de novos critérios para a avaliação de planos de governo; disponibilizar a funcionalidade que permite ao usuário incluir no sistema outros questionários de avaliação de planos de governo; e, implementar no sistema suporte para avaliação do mesmo plano por várias pessoas, oferecendo uma análise estatística das diferentes avaliações.

Bibliografia

ABREU, Pedro Felipe de, ABREU, Aline França de. Sistemas de Informações Gerenciais - uma abordagem orientada a negócios. [Florianópolis] Ed. IGTI - UFSC, 2000. 158p. (Apostila).

- BAER⁷, William C. **General Plan Evaluation Criteria**: An approach to making better plans. **Journal of the American Planning Association**. Chicago, Vol. 63, Issue 3, p. 329-344, Summer 1997. Disponível na Internet. <http://www.scielo.br>. 08 out. 2004.
- BRUTON, Michael J. (Ed.) **The Spirit and Purpose of Planning**. [London]: Hutchinson, p. 7, Apr 1974.
- Institute for Human and Machine Cognition. **CmapTools**. Versão 3.5. <http://cmap.ihmc.us>
- LARMAN, Craig. **Utilizando UML e padrões**: uma introdução à análise e ao projeto orientados a objetos. Ed. Bookman, 2001.
- PENDALL, Rolf. Municipal Plans, State Mandates, and Property Rights: Lessons from Maine. **Journal of Planning Education and Research**. Vol. 21, No. 2. p.154-165, 2001.
- QUEIRÓS, Maria de Lourdes. **Avaliação de Planos de Governo**: os planos plurianuais analisados segundo a formação de egressos, no âmbito do CNPq. Florianópolis, 2001. 104 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001. Disponível na Internet. <http://www.stela.ufsc.br>. 21 nov. 2003.
- SACCONI, Luiz Antonio. **Dicionário essencial da língua portuguesa**. São Paulo. Ed. Atual, 2001, p. 721.
- SILVA, Ricardo Pereira. **Engenharia de software**. Florianópolis : Universidade Federal de Santa Catarina, 2001. 50f. Apostila utilizada durante aulas de Engenharia de Software, realizada em 2001, em Florianópolis.

⁷ Apesar do artigo publicado no Journal of the American Planning Association possuir 16 páginas numeradas de 329-344, o artigo disponibilizado possui 20 páginas. Tal diferença deve ser dar em função do tamanho da folha do jornal ser diferente do tamanho da folha A4 na qual o artigo foi disponibilizado.

APÊNDICE B - CÓDIGO FONTE

```
package pavplgov;
public interface classeinterface { public void estabelececomunicacao(Comunicacao comunicacao) ;
public boolean getsucesso() ;
public Usuario getUsuario() ;
public void setUsuario(Usuario usuario) ;
public boolean getconectar() ;
public String getmensagem() ;}
```

```
package pavplgov;
import javax.swing.*.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class classeabstrata implements classeinterface { protected java.sql.Connection con;
protected java.sql.Statement globalstmt;
protected Comunicacao comunicacao;
protected Usuario atualusuario;
protected String mensagem;
protected boolean sucesso, conectar, estavaconectado;
protected javax.swing.JLabel labeldamensagem;
protected String codteoria, teoria, codmetodo, metodo, idiomat, ano_t, idiomam, pessoa;
public classeabstrata() { }
protected void comunicacaobd() throws Exception { java.sql.Statement stmt;
Class.forName("com.mysql.jdbc.Driver").newInstance();
con = java.sql.DriverManager.getConnection(comunicacao.getUrl(), comunicacao.getUsuario(),
comunicacao.getSenha());
stmt = con.createStatement();
stmt.close();}
protected boolean confirmasenha(String id, String senha) throws Exception { java.sql.Statement
stmtpessoa;
java.sql.Statement stmtusuario;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rsusuario;
boolean senhaok;
novamensagem("");
apagadados();
stmtusuario=con.createStatement();
rsusuario=stmtusuario.executeQuery("select * from usuario where id="+ id +" and senha=MD5(""+
senha +""");
senha="";
rsusuario.last();
senhaok=(rsusuario.getRow() > 0);
if (senhaok) { stmtpessoa=con.createStatement();
rspessoa=stmtpessoa.executeQuery("select * from pessoa where cod="+rsusuario.getString(2)+"");
rspessoa.last();
if (rspessoa.getRow() > 0) { atualusuario.nomeusuario=rspessoa.getString(2).trim();
atualusuario.emailusuario=rspessoa.getString(3).trim();
atualusuario.sexo=rspessoa.getString(4).trim();
atualusuario.codusuario=rsusuario.getString(1).trim();
atualusuario.codpessoa=rsusuario.getString(2).trim();
atualusuario.codtipousuario=rsusuario.getString(3).trim();
atualusuario.idusuario=rsusuario.getString(4).trim();}
else throw new SQLException();
rspessoa.close();}
```

```

stmtpessoa.close();}
rsusuario.close();
stmtusuario.close();
return senhaok;}
protected void desfazcomunicacaobd() throws Exception { con.close();}
protected boolean existeid(String id) throws Exception { return existeid(id, "", false);}
protected boolean existeid(String id, String codpessoa, boolean preencherdados) throws Exception {
java.sql.Statement stmtusuario;
java.sql.ResultSet rsusuario;
java.sql.Statement stmtpessoa;
java.sql.ResultSet rspessoa;
boolean existeoid;
novamensagem(" ");
stmtusuario= con.createStatement();
rsusuario = stmtusuario.executeQuery("select * from usuario where id="+ id +";");
rsusuario.last();
existeoid = (rsusuario.getRow() > 0);
if (preencherdados) { if (existeoid) { atualusuario.setCodusuario(rsusuario.getString(1).trim());
atualusuario.setCodpessoa(rsusuario.getString(2).trim());
atualusuario.setCodtipousuario(rsusuario.getString(3).trim());
atualusuario.setldusuario(rsusuario.getString(4).trim());}
else { apagadados();}
rsusuario.close();
stmtusuario.close();
stmtpessoa= con.createStatement();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where cod = "+ codpessoa +");");
rspessoa.last();
if (rspessoa.getRow() > 0) { atualusuario.setNomeusuario(rspessoa.getString(2).trim());
atualusuario.setEmailusuario(rspessoa.getString(3).trim());
atualusuario.setSexo(rspessoa.getString(4).trim());}
else { apagadados();}}
return existeoid;}
protected void novamensagem(String msg) { mensagem = msg;
labeldamensagem.setText(mensagem);}
protected void apagadados() { atualusuario.setCodusuario("");
atualusuario.setCodpessoa("");
atualusuario.setCodtipousuario("");
atualusuario.setldusuario("");
atualusuario.setNomeusuario("");
atualusuario.setEmailusuario("");
atualusuario.setSexo("");}
public void estabelececomunicacao(Comunicacao comunicacao) { this.comunicacao = comunicacao;
try { comunicacaobd();
novamensagem("");}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Não foi possível estabelecer comunicação com o servidor");}}
public Usuario getUsuario() { return atualusuario;}
public boolean getsucesso() { return sucesso;}
public boolean getconectar() { return conectar;}
public String getmensagem() { return mensagem;}
public void setUsuario(Usuario usuario) { }}

package pavplgov;
import javax.swing.*.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;

```

```

public class avplgov extends javax.swing.JFrame { protected java.sql.Connection con;
protected java.sql.Statement globalstmt;
protected Comunicacao comunicacao;
protected Usuario atualusuario;
protected String mensagem;
protected String codcriteriosdousuario[], codplanosdousuario[], codavaliacoesdousuario[];
protected int quantcriterios, quantplanos, quantavaliacoes;
protected boolean estaconectado, administrador;
protected altsenha dialogaltsenha;
protected avaliar dialogavaliar;
protected cadplano dialogcadplano;
protected cadcriterio dialogcadcriterio;
protected cadusuario dialogcadusuario;
protected conexao dialogconexao;
protected defpesoscrit dialogdefpesoscrit;
protected relatorios dialogrelatorios;
public avplgov(String usuario, String senha, String url) { comunicacao = new Comunicacao();
atualusuario = new Usuario();
comunicacao.setUsuario(usuario);
comunicacao.setSenha(senha);
comunicacao.setUrl(url);
initComponents();
try { novamensagem("");
apagadados();
templatesinicial();
estaconectado = false;
administrador = false;
comunicacaobd();
try { dialogaltsenha = new altsenha(this, true);
dialogavaliar = new avaliar(this, true);
dialogcadplano = new cadplano(this,true);
dialogcadusuario = new cadusuario(this, true);
dialogconexao = new conexao(this,true);
dialogcadcriterio = new cadcriterio(this,true);
dialogdefpesoscrit = new defpesoscrit(this,true);
dialogrelatorios = new relatorios(this, true);
try { dialogrelatorios.estabelececomunicacao(comunicacao);
dialogdefpesoscrit.estabelececomunicacao(comunicacao);
dialogcadcriterio.estabelececomunicacao(comunicacao);
dialogconexao.estabelececomunicacao(comunicacao);
dialogcadusuario.estabelececomunicacao(comunicacao);
dialogcadplano.estabelececomunicacao(comunicacao);
dialogavaliar.estabelececomunicacao(comunicacao);
dialogaltsenha.estabelececomunicacao(comunicacao);}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("A comunicação com o servidor foi estabelecido mas houve erro durante a
comunicação. Pode ser que haja problemas com o servidor de dados ou com a internet.");//
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Erro durante a inicialização. Pode ser que não haja memória suficiente.");
templatesemcomunicacao();}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Não foi possível estabelecer comunicação com o servidor. Pode ser que haja
problemas com o servidor de dados ou com a internet. Saia do programa e tente mais tarde.");
templatesemcomunicacao();}}
private void initComponents() { jButtoncadastro = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();

```

```

jLabel3 = new javax.swing.JLabel();
jButtonavaliar = new javax.swing.JButton();
jButtonrelatorios = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jButtoncadastrousuadm = new javax.swing.JButton();
jButtonaltsenha = new javax.swing.JButton();
jButtonteorias = new javax.swing.JButton();
jButtonmetodos = new javax.swing.JButton();
jButtonsair = new javax.swing.JButton();
jButtonconexao = new javax.swing.JButton();
jButtondesconexao = new javax.swing.JButton();
jLabelconexao = new javax.swing.JLabel();
jButtoncadplanogov = new javax.swing.JButton();
jButtoncadastrocriterio = new javax.swing.JButton();
jButtondefinirpesos = new javax.swing.JButton();
jButtonconsultabase = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButtonsobre = new javax.swing.JButton();
jButtonajuda = new javax.swing.JButton();
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Sistema de Avalia\u00e7\u00e3o de Planos de Governo");
setName("frameavplgov");
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { exitForm(evt);}});
jButtoncadastro.setText("Cadastro");
jButtoncadastro.setMaximumSize(new java.awt.Dimension(156, 26));
jButtoncadastro.setMinimumSize(new java.awt.Dimension(156, 26));
jButtoncadastro.setPreferredSize(new java.awt.Dimension(156, 26));
jButtoncadastro.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncadastroActionPerformed(evt);}});
getContentPane().add(jButtoncadastro, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 90,
170, -1));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("Sistema de Avalia\u00e7\u00e3o de Planos de Governo");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 20, -1, -1));
jLabel2.setText("Cadastro e Conex\u00e3o:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, -1, -1));
jLabel3.setText("Avalia\u00e7\u00f5es de Planos:");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, -1, -1));
jButtonavaliar.setText("Avalia\u00e7\u00e3o de Plano");
jButtonavaliar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonavaliarActionPerformed(evt);}});
getContentPane().add(jButtonavaliar, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 270,
170, -1));
jButtonrelatorios.setText("Relat\u00f3rios");
jButtonrelatorios.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonrelatoriosActionPerformed(evt);}});
getContentPane().add(jButtonrelatorios, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 360,
170, -1));
jLabel4.setText("Ferramentas para Administradores:");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 70, -1, -1));
jButtoncadastrousuadm.setText("Cadastro de usuarios");
getContentPane().add(jButtoncadastrousuadm, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 90, 200, -1));
jButtonaltsenha.setText("Altera\u00e7\u00e3o de senha");
jButtonaltsenha.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonaltsenhaActionPerformed(evt);}});

```

```

getContentPane().add(jButtonaltsenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
150, 170, -1));
jButtonteorias.setText("Cadastro de Teorias");
getContentPane().add(jButtonteorias, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 120,
200, -1));
jButtonmetodos.setText("Cadastro de M\u00e9todos");
getContentPane().add(jButtonmetodos, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 150,
200, -1));
jButtonsair.setText("Sair do Sistema");
jButtonsair.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonActionPerformed(evt);}});
getContentPane().add(jButtonsair, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 310, 200, -
1));
jButtonconexao.setText("Conex\u00e3o");
jButtonconexao.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonconexaoActionPerformed(evt);}});
getContentPane().add(jButtonconexao, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120,
170, -1));
jButtondesconexao.setText("Desconex\u00e3o");
jButtondesconexao.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtondesconexaoActionPerformed(evt);}});
getContentPane().add(jButtondesconexao, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
180, 170, -1));
jLabelconexao.setText("");
getContentPane().add(jLabelconexao, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 20, 10,
-1));
jButtoncadplanogov.setText("Cadastro de Plano");
jButtoncadplanogov.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncadplanogovActionPerformed(evt);}});
getContentPane().add(jButtoncadplanogov, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
240, 170, -1));
jButtoncadastrocriterio.setText("Cadastro de Crit\u00e9rio");
jButtoncadastrocriterio.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncadastrocriterioActionPerformed(evt);}});
getContentPane().add(jButtoncadastrocriterio, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
300, 170, -1));
jButtondefinirpesos.setText("Defini\u00e7\u00e3o de Pesos");
jButtondefinirpesos.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtondefinirpesosActionPerformed(evt);}});
getContentPane().add(jButtondefinirpesos, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
330, 170, -1));
jButtonconsultabase.setText("Consulta Base");
jButtonconsultabase.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonconsultabaseActionPerformed(evt);}});
getContentPane().add(jButtonconsultabase, new org.netbeans.lib.awtextra.AbsoluteConstraints(240,
180, 200, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 420, 420,
40));
jLabel5.setText("Diversos:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 220, -1, -1));
jButtonsobre.setText("Sobre o Sistema");
jButtonsobre.setEnabled(false);
getContentPane().add(jButtonsobre, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 240,
200, -1));
jButtonajuda.setText("Ajuda ao Usu\u00e1rio");
jButtonajuda.setEnabled(false);
getContentPane().add(jButtonajuda, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 270,
200, -1));
pack();

```

```

java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(462, 510));
setLocation((screenSize.width-462)/2,(screenSize.height-510)/2);}
private void formWindowClosed(java.awt.event.WindowEvent evt) { }
private void jButtonrelatoriosActionPerformed(java.awt.event.ActionEvent evt) { if (estaconectado) {
dialogrelatorios.setUsuario(atualusuario);
dialogrelatorios.show();
novamensagem(dialogrelatorios.getMensagem());} else { novamensagem("É necessário conectar
primeiro");}}
private void jButtondefinirpesosActionPerformed(java.awt.event.ActionEvent evt) { if (estaconectado) {
dialogdefpesoscrit.setUsuario(atualusuario);
dialogdefpesoscrit.show();
novamensagem(dialogdefpesoscrit.getMensagem());} else { novamensagem("É necessário conectar
primeiro");}}
private void jButtonconsultabaseActionPerformed(java.awt.event.ActionEvent evt) { }
private void jButtoncadastrocriterioActionPerformed(java.awt.event.ActionEvent evt) { if
(estaconectado) { dialogcadcriterio.setUsuario(atualusuario);
dialogcadcriterio.show();
novamensagem(dialogcadcriterio.getMensagem()); }
else { novamensagem("É necessário conectar primeiro");}}
private void jButtoncadplanogovActionPerformed(java.awt.event.ActionEvent evt) { if (estaconectado) {
dialogcadplano.setUsuario(atualusuario);
dialogcadplano.show();
novamensagem(dialogcadplano.getMensagem());} else { novamensagem("É necessário conectar
primeiro"); } }
private void jButtondesconexaoActionPerformed(java.awt.event.ActionEvent evt) {
desconectausuario();}
private void jButtonconexaoActionPerformed(java.awt.event.ActionEvent evt) { if (!estaconectado) {
dialogconexao.setUsuario(atualusuario);
dialogconexao.show();
if (dialogconexao.getsucesso()) { estaconectado = true;
atualusuario = dialogconexao.getUsuario();
if (atualusuario.getCodtipousuario().equalsIgnoreCase("1")) { administrador = true;
templatesconectadoadm();}
else { administrador = false;
templatesconectadousuario();}}
else { novamensagem(dialogconexao.getMensagem());} else { novamensagem("É necessário
desconectar primeiro");}}
private void jButtonaltsenhaActionPerformed(java.awt.event.ActionEvent evt) {
dialogaltsenha.setconectado(estaconectado);
dialogaltsenha.setUsuario(atualusuario);
dialogaltsenha.show();
if (dialogaltsenha.getconectar()) { if (dialogaltsenha.getsucesso()) { estaconectado = true;
atualusuario = dialogaltsenha.getUsuario();
if (atualusuario.getCodtipousuario().equalsIgnoreCase("1")) { administrador = true;
templatesconectadoadm();} else { administrador = false;
templatesconectadousuario(); } } else { novamensagem(dialogaltsenha.getMensagem());}
else { novamensagem(dialogaltsenha.getMensagem());}}
private void jButtonavaliarActionPerformed(java.awt.event.ActionEvent evt) { if (estaconectado) {
dialogavaliar.setUsuario(atualusuario);
dialogavaliar.show();
if (dialogavaliar.gethouvealgumsucesso()) { } else { novamensagem(dialogavaliar.getMensagem());}
else { novamensagem("É necessário conectar primeiro");}}
private void jButtononsairActionPerformed(java.awt.event.ActionEvent evt) { System.exit(0);}
private void jButtoncadastroActionPerformed(java.awt.event.ActionEvent evt) {
dialogcadusuario.setconectado(estaconectado);
dialogcadusuario.setUsuario(atualusuario);
dialogcadusuario.show();
if (dialogcadusuario.getconectar()) { if (dialogcadusuario.getsucesso()) { estaconectado = true;
atualusuario = dialogcadusuario.getUsuario();

```

```

if (atualusuario.getCodtipousuario().equalsIgnoreCase("1")) { administrador = true;
templatesconectadoadm();} else { administrador = false;
templatesconectadousuario();}}
else { novamensagem(dialogcadusuario.getMensagem());}
else { novamensagem(dialogcadusuario.getMensagem());}
private void exitForm(java.awt.event.WindowEvent evt) { System.exit(0);}
public static void main(String args[]) { String _usuario = "sistema";
String _senha ="sis2502";
String _url = "jdbc:mysql://ava.egr.ufsc.br/PLGOV"; //ava.egr.ufsc.br
if ((args.length > 0) && (args[0].equalsIgnoreCase("local"))) { //local
_usuario = "SISPLGOV";
_senha ="sis2502";
_url = "jdbc:mysql://localhost/PLGOV";};
new avplgov(_usuario,_senha,_url).show();}
public void novamensagem(String msg) { mensagem = msg;
jLabelmensagem.setText(mensagem);}
protected void comunicacaobd() throws Exception { java.sql.Statement stmt;
Class.forName("com.mysql.jdbc.Driver").newInstance();
con = java.sql.DriverManager.getConnection(comunicacao.getUrl(), comunicacao.getUsuario(),
comunicacao.getSenha());
stmt = con.createStatement();
stmt.close();}
public void desfazcomunicacaobd() throws Exception { con.close();}
boolean existecriterioparausuario() throws Exception { java.sql.Statement stmtcriterio;
java.sql.ResultSet rscriterio;
boolean existeocriterio;
int indicelocal;
stmtcriterio= con.createStatement();
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod_usuario="+
atualusuario.getCodusuario() +"");
rscriterio.last();
quantcriterios = rscriterio.getRow();
codcriteriosdousuario = new String[quantcriterios];
if (quantcriterios > 0 ) { existeocriterio = true;
rscriterio.beforeFirst();
indicelocal = 0;
while (indicelocal < quantcriterios) { rscriterio.next();
codcriteriosdousuario[indicelocal] = rscriterio.getString(1);
indicelocal++;}}
else { existeocriterio = false;}
rscriterio.close();
stmtcriterio.close();
return existeocriterio;}
boolean existeplanoparausuario() throws Exception { java.sql.Statement stmtplano;
java.sql.ResultSet rsplano;
boolean existeoplano;
int indicelocal;
stmtplano= con.createStatement();
rsplano = stmtplano.executeQuery("select * from plano where cod_usuario="+
atualusuario.getCodusuario() +"");
rsplano.last();
quantplanos = rsplano.getRow();
codplanosdousuario = new String[quantplanos];
if (quantplanos > 0 ) { existeoplano = true;
rsplano.beforeFirst();
indicelocal = 0;
while (indicelocal < quantplanos) { rsplano.next();
codplanosdousuario[indicelocal] = rsplano.getString(1);
indicelocal++;}}
else { existeoplano = false;}

```

```

rsplano.close();
stmtplano.close();
return existeoplano;}
boolean existeavaliacaoparausuario() throws Exception { java.sql.Statement stmtavaliacao;
java.sql.ResultSet rsavaliacao;
boolean existeavaliacao;
int indicelocal;
String clausulawhere;
if (existeplanoparausuario()) { clausulawhere = new String("cod_plano = " + codplanosdousuario[0]);
indicelocal = 1;
while (indicelocal < quantplanos) { clausulawhere = clausulawhere + " or cod_plano = " +
codplanosdousuario[indicelocal] ;
indicelocal++;}
stmtavaliacao= con.createStatement();
rsavaliacao = stmtavaliacao.executeQuery("select cod from avaliacao where " + clausulawhere +
" group by cod_plano;");
rsavaliacao.last();
quantavaliacoes = rsavaliacao.getRow();
codavaliacoedousuario = new String[quantavaliacoes];
if (quantavaliacoes > 0 ) { existeavaliacao = true;
rsavaliacao.beforeFirst();
indicelocal = 0;
while (indicelocal < quantavaliacoes) { rsavaliacao.next();
codavaliacoedousuario[indicelocal] = rsavaliacao.getString(1);
indicelocal++;}}
else { existeavaliacao = false;}
rsavaliacao.close();
stmtavaliacao.close(); }
else { quantavaliacoes = 0;
codavaliacoedousuario = new String[quantavaliacoes];
existeavaliacao = false;}
return existeavaliacao;}
public Usuario getUsuario() { return atualusuario;}
public boolean getestaconectado() { return estaconectado;}
public String getmensagem() { return mensagem;}
protected void desconectausuario() { templatesinicial();
apagadados();
estaconectado = false;
administrador = false;}
protected void apagadados() { atualusuario.setCodusuario("");
atualusuario.setCodpessoa("");
atualusuario.setCodtipousuario("");
atualusuario.setldusuario("");
atualusuario.setNomeusuario("");
atualusuario.setEmailusuario("");
atualusuario.setSexo("");
codcriteriosdousuario = new String[0];
quantcriterios = 0;
codplanosdousuario = new String[0];
quantplanos = 0;
codavaliacoedousuario = new String[0];
quantavaliacoes = 0;}
protected void templatesinicial() { templatesnaoconectado();}
protected void templatesnaoconectado() { if (estaconectado) { if (administrador) {
novamensagem("Administrador desconectado");} else { novamensagem("Usuário desconectado");}};
//cadastro e conexão:
jButtoncadastro.setEnabled(true);
jButtonconexao.setEnabled(true);
jButtonaltsenha.setEnabled(true);
jButtondesconexao.setEnabled(false);

```



```

//avaliação de planos:
jButtoncadplanogov.setEnabled(false);
jButtonavaliar.setEnabled(false);
jButtoncadastrocriterio.setEnabled(false);
jButtondefinirpesos.setEnabled(false);
jButtonrelatorios.setEnabled(false);
//ferramentas para administradores:
jButtoncadastrousuadm.setEnabled(false);
jButtonteorias.setEnabled(false);
jButtonmetodos.setEnabled(false);
jButtonconsultabase.setEnabled(false);
jLabelconexao.setText("");}
protected void templatesconectadousuario() { novamensagem("Usuário conectado");
//cadastro e conexão:
jButtoncadastro.setEnabled(true);
jButtonconexao.setEnabled(false);
jButtonaltsenha.setEnabled(true);
jButtondesconexao.setEnabled(true);
//avaliação de planos:
jButtoncadplanogov.setEnabled(true);
jButtonavaliar.setEnabled(true);
jButtoncadastrocriterio.setEnabled(true);
jButtondefinirpesos.setEnabled(true);
jButtonrelatorios.setEnabled(true);
//ferramentas para administradores:
jButtoncadastrousuadm.setEnabled(false);
jButtonteorias.setEnabled(false);
jButtonmetodos.setEnabled(false);
jButtonconsultabase.setEnabled(false);
jLabelconexao.setText(" *");}
protected void templatesconectadoadm() { novamensagem("Administrador conectado");
//cadastro e conexão:
jButtoncadastro.setEnabled(true);
jButtonconexao.setEnabled(false);
jButtonaltsenha.setEnabled(true);
jButtondesconexao.setEnabled(true);
//avaliação de planos:
jButtoncadplanogov.setEnabled(true);
jButtonavaliar.setEnabled(true);
jButtoncadastrocriterio.setEnabled(true);
jButtondefinirpesos.setEnabled(true);
jButtonrelatorios.setEnabled(true);
//ferramentas para administradores:
jButtoncadastrousuadm.setEnabled(true);
jButtonteorias.setEnabled(true);
jButtonmetodos.setEnabled(true);
jButtonconsultabase.setEnabled(true);
jLabelconexao.setText(" #");}
protected void templatessemcomunicacao() { //cadastro e conexão:
jButtoncadastro.setEnabled(false);
jButtonconexao.setEnabled(false);
jButtonaltsenha.setEnabled(false);
jButtondesconexao.setEnabled(false);
//avaliação de planos:
jButtoncadplanogov.setEnabled(false);
jButtonavaliar.setEnabled(false);
jButtoncadastrocriterio.setEnabled(false);
jButtondefinirpesos.setEnabled(false);
jButtonrelatorios.setEnabled(false);
//ferramentas para administradores:

```

```

jButtoncadastrousuadm.setEnabled(false);
jButtonteorias.setEnabled(false);
jButtonmetodos.setEnabled(false);
jButtonconsultabase.setEnabled(false);}
private javax.swing.JLabel jLabel4;
private javax.swing.JButton jButtonaltsenha;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JButton jButtonavaliar;
private javax.swing.JButton jButtonajuda;
private javax.swing.JLabel jLabel2;
private javax.swing.JButton jButtondefinirpesos;
private javax.swing.JButton jButtonteorias;
private javax.swing.JButton jButtoncadastrousuadm;
private javax.swing.JButton jButtonconsultabase;
private javax.swing.JButton jButtonssobre;
private javax.swing.JButton jButtoncadplanogov;
private javax.swing.JButton jButtonrelatorios;
private javax.swing.JButton jButtonconexao;
private javax.swing.JButton jButtonssair;
private javax.swing.JLabel jLabelconexao;
private javax.swing.JButton jButtoncadastro;
private javax.swing.JButton jButtoncadastrocriterio;
private javax.swing.JButton jButtondesconexao;
private javax.swing.JButton jButtonmetodos;
private javax.swing.JLabel jLabel5;}

```

```

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class cadusuario extends dialogoabstrato { protected String topicos, questoestop, questoestot,
topicoatual, questaoatualtop, questaoatualtot;
protected int quanttopicos, quantquestoestop, quantquestoestot, numtopicoatual,
numquestaoatual, numquestaoatualtopatual;
public cadusuario(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;}
private void initComponents() { buttonGroupsexo = new javax.swing.ButtonGroup();
jTextFieldnome = new javax.swing.JTextField();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jButtonok = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
jTextFieldemail = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jTextFieldid = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jPasswordFieldsenha2 = new javax.swing.JPasswordField();
jPasswordFieldsenha1 = new javax.swing.JPasswordField();
jLabel51 = new javax.swing.JLabel();
jRadioButtonmasculino = new javax.swing.JRadioButton();
jRadioButtonfeminino = new javax.swing.JRadioButton();
jLabel21 = new javax.swing.JLabel();

```

```

jCheckBoxconectar = new javax.swing.JCheckBox();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Cadastro de Usu\u00e1rios");
setModal(true);
setName("dialogcadusuario");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
getContentPane().add(jTextFieldnome, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70,
330, -1));
jLabel1.setText("Cadastre os seus dados");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 20, -1, -1));
jLabel2.setText("Nome Completo: ");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 50, 140, -1));
jButtonok.setText("Ok");
jButtonok.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonokActionPerformed(evt);}});
getContentPane().add(jButtonok, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, 90, -
1));
jLabel3.setText("E-Mail:");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 90, -1));
getContentPane().add(jTextFieldemail, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120,
330, -1));
jLabel4.setText("Identificador:");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 150, -1, -1));
getContentPane().add(jTextFieldid, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, 190, -
1));
jLabel5.setText("Confirme a Senha:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 350,
90, -1));
getContentPane().add(jPasswordFieldsenha2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
270, 140, -1));
getContentPane().add(jPasswordFieldsenha1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
220, 140, -1));
jLabel51.setText("Senha:");
getContentPane().add(jLabel51, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 200, -1, -1));
jRadioButtonmasculino.setText("Masculino");
buttonGroupsexo.add(jRadioButtonmasculino);
getContentPane().add(jRadioButtonmasculino, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 170, -1, -1));
jRadioButtonfeminino.setText("Feminino");
buttonGroupsexo.add(jRadioButtonfeminino);
getContentPane().add(jRadioButtonfeminino, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
190, -1, -1));
jLabel21.setText("Sexo:");
getContentPane().add(jLabel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 150, 40, -1));
jCheckBoxconectar.setText("Conectar");
getContentPane().add(jCheckBoxconectar, new org.netbeans.lib.awtextra.AbsoluteConstraints(140,
310, 100, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 400, 330,
40));
pack();

```

```

java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(372, 492));
setLocation((screenSize.width-372)/2,(screenSize.height-492)/2);}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
if (!houvealgumsucesso) { sucesso = false;};
conectar = false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void formComponentShown(java.awt.event.ComponentEvent evt) { apagatemplates();
sucesso = false;
houvealgumsucesso = false;
jButtoncancelar.setText("Cancelar");
novamensagem("");}
private void jButtonokActionPerformed(java.awt.event.ActionEvent evt) { boolean jaexiste, deucerto;
String nome, mail, id, senha1, senha2 ;
jaexiste = false;
deucerto = true;
sucesso = false;
novamensagem("");
conectar = ((jCheckBoxconectar.isSelected()) && (jCheckBoxconectar.isEnabled()));
try { nome = jTextFieldnome.getText().trim();
jTextFieldnome.setText(nome);
if (!(jRadioButtonmasculino.isSelected() | jRadioButonfeminino.isSelected())) { novamensagem("O
sexo deve ser selecionado");
atualusuario.setSexo("");
deucerto = false;} else { if (jRadioButtonmasculino.isSelected()) { atualusuario.setSexo("m"); }
else { atualusuario.setSexo("f"); }
if (nome.length() < 4) { novamensagem("O nome deve possuir pelo menos quatro caracteres");
deucerto = false; }
else { mail = jTextFieldemail.getText().trim();
jTextFieldemail.setText(mail);
if (mail.length() < 4) { novamensagem("O e-mail deve possuir pelo menos quatro caracteres");
deucerto = false;} else { id = jTextFieldid.getText().trim();
jTextFieldid.setText(id);
if (id.length() < 4) { novamensagem("O identificador deve possuir pelo menos quatro caracteres");
deucerto = false;} else { if (id.length() > 16) { novamensagem("O identificador deve possuir no máximo
dezesseis caracteres");
deucerto = false;} else { senha1 = new String(jPasswordFieldsenha1.getPassword());
senha2 = new String(jPasswordFieldsenha2.getPassword());
if (!senha2.equals(senha1)) { novamensagem("A senha deve ser igual a confirmação da senha");
deucerto = false;} else { if (senha1.length() < 4) { novamensagem("A senha deve possuir no mínimo
quatro caracteres");
deucerto = false;} else { if (senha1.length() > 16) { novamensagem("A senha deve possuir no máximo
dezesseis caracteres");
deucerto = false; }
else { if (existeid(jTextFieldid.getText())) { novamensagem("Já existe o usuário com identificador ""+
jTextFieldid.getText()+"" tente outro");
jaexiste = true;}
else { jaexiste = false;}}}}
senha1 = "";
senha2 = "";}}}}}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
deucerto = false;
if (conectar) { novamensagem("Não foi possível inserir e conectar, tente mais tarde...");} else {
novamensagem("Não foi possível inserir, tente mais tarde...");}}
if (!jaexiste && deucerto) { try { if (insereusuario()) { jButtoncancelar.setText("Sair");

```

```

houvealgumsucesso = true;
if (conectar) { sucesso = true;
novamensagem("Usuário "+ jTextFieldid.getText()+" inserido e conectado com sucesso");
apagatemplates();}
else { novamensagem("Usuário "+ jTextFieldid.getText()+" inserido com sucesso");
apagatemplates();}
else { if (conectar) { novamensagem("Não foi possível inserir e conectar, tente novamente...");}
else { novamensagem("Não foi possível inserir, tente novamente...");}}
apagassenhas();}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
apagassenhas();
if (conectar) { novamensagem("Não foi possível inserir nem conectar, tente mais tarde...");}
else { novamensagem("Não foi possível inserir, tente mais tarde...");}}
else { apagassenhas();}
if (sucesso) { setVisible(false);
dispose();}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Operação 'Cadastro' concluída");} else { sucesso = false;
novamensagem("Operação 'Cadastro' cancelada pelo usuário"); }
conectar = false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
public static void main(String args[]) { new cadusuario(new javax.swing.JFrame(), true).show();}
boolean insereusuario() throws Exception { java.sql.Statement stmtusuario;
java.sql.Statement stmtpessoa;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rsusuario;
boolean inseriuid;
String codigopessoa, senha;
novamensagem("");
apagadados();
stmtpessoa= con.createStatement();
rspessoa = stmtpessoa.executeQuery("select cod from pessoa where nome =
"+jTextFieldnome.getText()+" and email = "+jTextFieldemail.getText()+" and sexo = "+
atualusuario.getSexo() +";");
rspessoa.last();
inseriuid = (rspessoa.getRow() > 0);
rspessoa.close();
stmtpessoa.close();
if (!(inseriuid)) { stmtpessoa= con.createStatement();
inseriuid = stmtpessoa.execute("INSERT INTO pessoa (cod, nome, email, sexo) VALUES (NULL,
"+jTextFieldnome.getText()+" , "+jTextFieldemail.getText()+" , "+ atualusuario.getSexo() +");");
stmtpessoa.close();}
stmtpessoa= con.createStatement();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where nome =
"+jTextFieldnome.getText()+" and email = "+jTextFieldemail.getText()+" and sexo = "+
atualusuario.getSexo() +";");
rspessoa.last();
inseriuid = (rspessoa.getRow() > 0);
if (inseriuid) { atualusuario.setCodpessoa(rspessoa.getString(1).trim());
atualusuario.setNomeusuario(rspessoa.getString(2).trim());
atualusuario.setEmailusuario(rspessoa.getString(3).trim());
atualusuario.setSexo(rspessoa.getString(4).trim());
stmtusuario= con.createStatement();
senha = new String(jPasswordFieldsenha1.getPassword());

```

```

inseriuoid = stmtusuario.execute("INSERT INTO usuario (cod, cod_pessoa, cod_tipousuario,
id, senha) VALUES (NULL, "+ atualusuario.getCodpessoa() +", 5, '"+ jTextFieldid.getText() + "",
MD5(""+ senha +"));");
senha = "";
stmtusuario.close();
stmtusuario = con.createStatement();
rsusuario = stmtusuario.executeQuery("select * from usuario where cod_pessoa = "+
atualusuario.getCodpessoa() + " and id = '"+ jTextFieldid.getText() + "");
rsusuario.last();
inseriuoid = (rsusuario.getRow() > 0);
if (inseriuoid) { atualusuario.setCodusuario(rsusuario.getString(1).trim());
atualusuario.setCodpessoa(rsusuario.getString(2).trim());
atualusuario.setCodtipousuario(rsusuario.getString(3).trim());
atualusuario.setldusuario(rsusuario.getString(4).trim());} else { apagadados();}
rsusuario.close();
stmtusuario.close();}
else { apagadados();}
rspessoa.close();
stmtpessoa.close();
return inseriuoid;}

public void preenchetipousuario() throws Exception { java.sql.Statement stmttipousuario;
java.sql.ResultSet rstipousuario;
stmttipousuario= con.createStatement();
rstipousuario = stmttipousuario.executeQuery("select * from tipousuario;");
rstipousuario.last();
rstipousuario.close();
stmttipousuario.close();}

public void setconectado(boolean conectado) { estavaconectado = conectado;
if (estavaconectado) { jCheckBoxconectar.setSelected(false);
jCheckBoxconectar.setEnabled(false);}
else { jCheckBoxconectar.setSelected(false);
jCheckBoxconectar.setEnabled(true);}}

protected void apagatemplates() { buttonGroupsexo.remove(jRadioButtonmasculino);
buttonGroupsexo.remove(jRadioButonfeminino);
jRadioButtonmasculino.setSelected(false);
jRadioButonfeminino.setSelected(false);
buttonGroupsexo.add(jRadioButtonmasculino);
buttonGroupsexo.add(jRadioButonfeminino);
jTextFieldeemail.setText("");
jPasswordFieldsenha1.setText("");
jTextFieldid.setText("");
jTextFielddnome.setText("");
jPasswordFieldsenha2.setText("");
jCheckBoxconectar.setSelected(false);}

protected void apagassenhas() { jPasswordFieldsenha1.setText("");
jPasswordFieldsenha2.setText("");}

private javax.swing.JCheckBox jCheckBoxconectar;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel21;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField jTextFieldemail;
private javax.swing.JPasswordField jPasswordFieldsenha1;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JTextField jTextFieldid;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel51;
private javax.swing.JTextField jTextFieldnome;
private javax.swing.JPasswordField jPasswordFieldsenha2;

```

```

private javax.swing.JButton jButtonok;
private javax.swing.ButtonGroup buttonGroupsexo;
private javax.swing.JRadioButton jRadioButtonfeminino;
private javax.swing.JRadioButton jRadioButtonmasculino;
private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabel5;}

package pavplgov;
public class Comunicacao { protected String usuario;
protected String senha;
protected String url;
public Comunicacao() { }
public String getUsuario() { return usuario;}
public String getSenha() { return senha;}
public String getUrl() { return url;}
public void setUsuario(String usuario) { this.usuario = usuario;}
public void setSenha(String senha) { this.senha = senha;}
public void setUrl(String url) { this.url = url;}}

package pavplgov;
public class Usuario { String codusuario;
String codpessoa;
String codtipousuario;
String idusuario;
String nomeusuario;
String emailusuario;
String sexo;
public Usuario() { }
public String getCodusuario() { return codusuario;}
public String getCodpessoa() { return codpessoa;}
public String getCodtipousuario() { return codtipousuario;}
public String getIdusuario() { return idusuario;}
public String getNomeusuario() { return nomeusuario;}
public String getEmailusuario() { return emailusuario;}
public String getSexo() { return sexo;}
public void setCodusuario(String codusuario) { this.codusuario = codusuario;}
public void setCodpessoa(String codpessoa) { this.codpessoa = codpessoa;}
public void setCodtipousuario(String codtipousuario) { this.codtipousuario = codtipousuario;}
public void setIdusuario(String idusuario) { this.idusuario = idusuario;}
public void setNomeusuario(String nomeusuario) { this.nomeusuario = nomeusuario;}
public void setEmailusuario(String emailusuario) { this.emailusuario = emailusuario;}
public void setSexo(String sexo) { this.sexo = sexo;}}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class dialogoabstrato extends javax.swing.JDialog implements classeinterface { protected
java.sql.Connection con;
protected java.sql.Statement globalstmt;
protected Comunicacao comunicacao;
protected Usuario atualusuario;
protected String mensagem, mensagem1;
protected boolean sucesso, houvealgumsucesso ;
protected boolean conectar, estavaconectado;
protected javax.swing.JLabel labeldamensagem;
protected String codteoria, teoria, codmetodo, metodo, idiomat, ano_t, idiomam, pessoa;

```

```

protected int quantidadeorias, quantidademetodos, quantidadeplanogovs,
quantidadecriterios;
protected int codComboBoxteoria[], codComboBoxmetodo[], codComboBoxplanogov[],
codComboBoxcriterio[], codlistcriterio[];
protected String codplanogov, planogov, codavaliacao, avaliacao, codcriterio, criterio,
porcentmincriterio;
public dialogoabstrato(java.awt.Frame parent, boolean modal) { super(parent, modal);
atualusuario = null;
sucesso = false;
houvealgumsucesso = false;}
private void initComponents() { }
public static void main(String args[]) { new dialogoabstrato(new javax.swing.JFrame(), true).show();}
protected void comunicacaobd() throws Exception { java.sql.Statement stmt;
Class.forName("com.mysql.jdbc.Driver").newInstance();
con = java.sql.DriverManager.getConnection(comunicacao.getUrl(), comunicacao.getUsuario(),
comunicacao.getSenha());
stmt = con.createStatement();
stmt.close();}
protected boolean confirmasenha(String id, String senha) throws Exception { java.sql.Statement
stmtpessoa;
java.sql.Statement stmtusuario;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rsusuario;
boolean senhaok;
novamensagem("");
apagadados();
stmtusuario=con.createStatement();
rsusuario=stmtusuario.executeQuery("select * from usuario where id='"+ id +" and senha=MD5("'+
senha +"'");");
senha="";
rsusuario.last();
senhaok=(rsusuario.getRow() > 0);
if (senhaok) { stmtpessoa=con.createStatement();
rspessoa=stmtpessoa.executeQuery("select * from pessoa where cod="+rsusuario.getString(2)+"");");
rspessoa.last();
if (rspessoa.getRow() > 0) { atualusuario.setNomeusuario(rspessoa.getString(2).trim());
atualusuario.setEmailusuario(rspessoa.getString(3).trim());
atualusuario.setSexo(rspessoa.getString(4).trim());
atualusuario.setCodusuario(rsusuario.getString(1).trim());
atualusuario.setCodpessoa(rsusuario.getString(2).trim());
atualusuario.setCodtipousuario(rsusuario.getString(3).trim());
atualusuario.setldusuario(rsusuario.getString(4).trim());}
else throw new SQLException();
rspessoa.close();
stmtpessoa.close();}
rsusuario.close();
stmtusuario.close();
return senhaok;}
protected void desfazcomunicacaobd() throws Exception { con.close();}
protected boolean existeid(String id) throws Exception { return existeid(id, "", false);}
protected boolean existeid(String id, String codpessoa, boolean preencherdados) throws Exception {
java.sql.Statement stmtusuario;
java.sql.ResultSet rsusuario;
java.sql.Statement stmtpessoa;
java.sql.ResultSet rspessoa;
boolean existeoid;
novamensagem(" ");
stmtusuario= con.createStatement();
rsusuario = stmtusuario.executeQuery("select * from usuario where id='"+ id +"'");
rsusuario.last();

```



```

existeoid = (rsusuario.getRow() > 0);
if (preencherdados) { if (existeoid) { atualusuario.setCodusuario(rsusuario.getString(1).trim());
atualusuario.setCodpessoa(rsusuario.getString(2).trim());
atualusuario.setCodtipousuario(rsusuario.getString(3).trim());
atualusuario.setldusuario(rsusuario.getString(4).trim());}
else { apagadados();}
rsusuario.close();
stmtusuario.close();
stmtpessoa= con.createStatement();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where cod = "+ codpessoa +";");
rspessoa.last();
if (rspessoa.getRow() > 0) { atualusuario.setNomeusuario(rspessoa.getString(2).trim());
atualusuario.setEmailusuario(rspessoa.getString(3).trim());
atualusuario.setSexo(rspessoa.getString(4).trim());}
else { apagadados();}}
return existeoid;}
protected void novamensagem(String msg) { mensagem = msg;
labeldamensagem.setText(mensagem);}
protected void apagadados() { if (atualusuario != null) { atualusuario.setCodusuario("");
atualusuario.setCodpessoa("");
atualusuario.setCodtipousuario("");
atualusuario.setldusuario("");
atualusuario.setNomeusuario("");
atualusuario.setEmailusuario("");
atualusuario.setSexo("");}}
public void estabelececomunicacao(Comunicacao comunicacao) { this.comunicacao = comunicacao;
try { comunicacaobd();
novamensagem("");}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Não foi possível estabelecer comunicação com o servidor");}}
protected boolean existecriterio(String descricao, String codmetodo) throws Exception {
java.sql.Statement stmtcriterio;
java.sql.ResultSet rscriterio;
boolean existeocriterio;
labeldamensagem.setText("");
stmtcriterio= con.createStatement();
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod_usuario=" +
atualusuario.getCodusuario() + " and cod_metodo=" + codmetodo + " and descr=" + descricao + ";"");
rscriterio.last();
existeocriterio = (rscriterio.getRow() > 0);
rscriterio.close();
stmtcriterio.close();
return existeocriterio;}
public void setUsuario(Usuario usuario) { atualusuario = usuario;}
public Usuario getUsuario() { return atualusuario;}
public boolean getsucesso() { return sucesso;}
public boolean gethouvealgumsucesso() { return houvealgumsucesso;}
public boolean getconectar() { return conectar;}
public String getmensagem() { return mensagem;}
protected boolean preencheteoria(javax.swing.JComboBox boxteoria,
javax.swing.JComboBox boxmetodo, javax.swing.JComboBox boxplanogov,
javax.swing.JComboBox boxcriterio, javax.swing.JList listcriterio,
javax.swing.JButton botaoparadesabilitar1,
javax.swing.JButton botaoparadesabilitar2)
throws Exception { java.sql.Statement stmtteoria;
java.sql.Statement stmtidioma;
java.sql.Statement stmtpessoa;
java.sql.ResultSet rsteoria;
java.sql.ResultSet rsidioma;

```

```

java.sql.ResultSet rspessoa;
String lteoria;
int i;
boolean teoriaok;
novamensagem("");
boxteoria.removeAllItems();
boxmetodo.removeAllItems();
if (boxplanogov != null) { boxplanogov.removeAllItems();}
if (boxcriterio != null) { boxcriterio.removeAllItems();}
if (listcriterio != null) { listcriterio.removeAll();}
stmtteoria = con.createStatement();
stmtidioma= con.createStatement();
stmtpessoa= con.createStatement();
rsteoria = stmtteoria.executeQuery("select * from teoria;");
rsteoria.last();
quantidadeteorias = rsteoria.getRow();
codComboBoxteoria = new int[quantidadeteorias];
if (quantidadeteorias < 1) { boxteoria.addItem("Não há teoria de avaliação cadastrada...");
boxteoria.setToolTipText("Não há teoria de avaliação cadastrada...");
boxteoria.setEnabled(false);
boxmetodo.setEnabled(false);
if (boxplanogov != null) { boxplanogov.setEnabled(false);}
if (boxcriterio != null) { boxcriterio.setEnabled(false);}
if (listcriterio != null) { listcriterio.setEnabled(false);}
novamensagem("É necessário cadastrar uma teoria de avaliação");
botaoparadesabilita1.setEnabled(false);
if (botaoparadesabilita2 != null) { botaoparadesabilita2.setEnabled(false);}
codteoria = "null";
teoria = "";
teoriaok = false;}
else { boxteoria.setToolTipText("");
boxteoria.setEnabled(true);
rsteoria.beforeFirst();
i = 1;
while (i <= quantidadeteorias) { rsteoria.next();
rsidioma = stmtidioma.executeQuery("select descr from idioma where cod="+rsteoria.getString(3)+");");
rsidioma.first();
rspessoa = stmtpessoa.executeQuery("select nome from pessoa where
cod="+rsteoria.getString(2)+");");
rspessoa.first();
lteoria = String.valueOf(i) + " - " + rsteoria.getString(4)+ " - " + rspessoa.getString(1) + " -
"+rsidioma.getString(1)+" - "+rsteoria.getString(6);
boxteoria.addItem(lteoria);
rspessoa.close();
rsidioma.close();
codComboBoxteoria[i-1] = rsteoria.getInt(1);
i++;}
codteoria = String.valueOf(codComboBoxteoria[boxteoria.getSelectedIndex()]);
teoria = "";
teoriaok = true;}
rsteoria.close();
stmtpessoa.close();
stmtidioma.close();
stmtteoria.close();
return teoriaok;}
protected boolean preenchemetodo(javax.swing.JComboBox boxteoria,
javax.swing.JComboBox boxmetodo, javax.swing.JComboBox boxplanogov,
javax.swing.JComboBox boxcriterio, javax.swing.JList listcriterio,
javax.swing.JButton botaoparadesabilita1,
javax.swing.JButton botaoparadesabilita2)

```

```

throws Exception { java.sql.Statement stmtmetodo;
java.sql.Statement stmtidioma;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidioma;
String lmetodo;
int i;
boolean metodook;
novamensagem("");
boxmetodo.removeAllItems();
if (boxplanogov != null) { boxplanogov.removeAllItems();}
if (boxcriterio != null) { boxcriterio.removeAllItems();}
if (listcriterio != null) { listcriterio.removeAll();}
stmtmetodo = con.createStatement();
stmtidioma= con.createStatement();
rsmetodo = stmtmetodo.executeQuery("select * from metodo where
cod_teorias="+String.valueOf(boxteoria.getSelectedIndex()+1)+";");
rsmetodo.last();
quantidademetodos = rsmetodo.getRow();
codComboBoxmetodo = new int[quantidademetodos];
if (quantidademetodos < 1) { boxmetodo.addItem("Não há método cadastrado...");
boxmetodo.setToolTipText("Não há método cadastrado...");
boxmetodo.setEnabled(false);
if (boxplanogov != null) { boxplanogov.setEnabled(false);}
if (boxcriterio != null) { boxcriterio.setEnabled(false);}
if (listcriterio != null) { listcriterio.setEnabled(false);}
novamensagem("É necessário cadastrar um método de avaliação");
botaoparadesabilitar1.setEnabled(false);
if (botaoparadesabilitar2 != null) { botaoparadesabilitar2.setEnabled(false);}
codmetodo = "null";
metodo = "";
metodook = false;}
else { boxmetodo.setToolTipText("");
boxmetodo.setEnabled(true);
rsmetodo.beforeFirst();
i = 1;
while (i <= quantidademetodos) { rsmetodo.next();
rsidioma = stmtidioma.executeQuery("select descr from idioma where
cod="+rsmetodo.getString(3)+";");
rsidioma.first();
lmetodo = String.valueOf(i) + " - " + rsmetodo.getString(4) + " - " + rsidioma.getString(1) + " - " +
rsmetodo.getString(6);
boxmetodo.addItem(lmetodo);
rsidioma.close();
codComboBoxmetodo[i-1] = rsmetodo.getInt(1);
i++;}
codmetodo = String.valueOf(codComboBoxmetodo[boxmetodo.getSelectedIndex()]);
metodo = "";
metodook = true;}
rsmetodo.close();
stmtidioma.close();
stmtmetodo.close();
return metodook;}
protected boolean preencheplanogov(javax.swing.JComboBox boxplanogov,
javax.swing.JList listcriterio, javax.swing.JButton botaoparadesabilitar) throws Exception {
java.sql.Statement stmtplanogov;
java.sql.ResultSet rsplanogov;
String lplanogov;
int i;
boolean planogovok;
novamensagem("");

```

```

boxplanogov.removeAllItems();
if (listcriterio != null) { listcriterio.removeAll();}
stmtplanogov = con.createStatement();
rsplanogov = stmtplanogov.executeQuery("select * from plano where cod_usuario=" +
atualusuario.getCodusuario() + ";");
rsplanogov.last();
quantidadeplanogovs = rsplanogov.getRow();
codComboBoxplanogov = new int[quantidadeplanogovs];
if (quantidadeplanogovs < 1) { boxplanogov.addItem("Não há plano de governo cadastrado...");
boxplanogov.setToolTipText("Não há plano de governo cadastrado...");
boxplanogov.setEnabled(false);
if (listcriterio != null) { listcriterio.setEnabled(false);}
novamensagem("É necessário cadastrar um plano de governo para avaliar");
botaoparadesabilitar.setEnabled(false);
codplanogov = "null";
planogov = "";
planogovok = false;}
else { boxplanogov.setToolTipText("");
boxplanogov.setEnabled(true);
botaoparadesabilitar.setEnabled(true);
rsplanogov.beforeFirst();
i = 1;
while (i <= quantidadeplanogovs) { rsplanogov.next();
lplanogov = rsplanogov.getString(3);
boxplanogov.addItem(lplanogov);
codComboBoxplanogov[i-1] = rsplanogov.getInt(1);
i++;}
codplanogov = String.valueOf(codComboBoxplanogov[boxplanogov.getSelectedIndex()]);
planogov = "";
planogovok = true;}
rsplanogov.close();
stmtplanogov.close();
return planogovok;}}

```

```

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class conexao extends dialogoabstrato { public conexao(java.awt.Frame parent, boolean
modal) { super(parent, modal);
initComponents();
labeldamensagem = JLabelmensagem;}
private void initComponents() { jDialog1 = new javax.swing.JDialog();
jLabel1 = new javax.swing.JLabel();
jButtonok = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jTextFieldid = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jPasswordFieldsenha = new javax.swing.JPasswordField();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jDialog1.getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Conex\u00e3o de Usu\u00e1rio");
setModal(true);
setName("dialogconexao");

```

```

addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Conex\u00e3o de usu\u00e1rio ");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 20, -1, -1));
jButtonok.setText("Ok");
jButtonok.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonokActionPerformed(evt);}});
getContentPane().add(jButtonok, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 190, 90, -
1));
jLabel4.setText("Identificador:");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, -1, -1));
getContentPane().add(jTextFieldid, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 190, -
1));
jLabel5.setText("Senha:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 110, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 190,
90, -1));
getContentPane().add(jPasswordFieldsenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
130, 140, -1));
jScrollPane1.setMinimumSize(new java.awt.Dimension(330, 40));
jScrollPane1.setPreferredSize(new java.awt.Dimension(330, 40));
jScrollPane1.setAutoscrolls(true);
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 240, 330,
40));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(372, 330));
setLocation((screenSize.width-372)/2,(screenSize.height-330)/2);}
private void formComponentShown(java.awt.event.ComponentEvent evt) { sucesso=false;
apagadados();
apagatemplates();
novamensagem("");}
private void jButtonokActionPerformed(java.awt.event.ActionEvent evt) { boolean jaexiste, deucerto,
senhaconfere;
String nome, mail, id, senha;
jaexiste=false;
deucerto=true;
senhaconfere=false;
sucesso=false;
novamensagem("");
try { id=jTextFieldid.getText().trim();
jTextFieldid.setText(id);
if (id.length() < 4) { novamensagem("O identificador deve possuir pelo menos quatro caracteres");
deucerto=false;}
else { if (id.length() > 16) { novamensagem("O identificador deve possuir no máximo dezesseis
caracteres");
deucerto=false;}
else { senha=new String(jPasswordFieldsenha.getPassword());
if (senha.length() < 4) { novamensagem("A senha deve possuir no mínimo quatro caracteres");
deucerto=false;}
else { if (senha.length() > 16) { novamensagem("A senha deve possuir no máximo dezesseis
caracteres");
deucerto=false;}

```

```

else { if (!(existeid(jTextFieldid.getText())) { novamensagem("Nao existe o usuário com
identificador ""+ jTextFieldid.getText()+"" tente outro!");
jaexiste=false;}
else { jaexiste=true;
if (!(confirmasenha(jTextFieldid.getText(), new String(jPasswordFieldsenha.getPassword())))) {
novamensagem("Senha não confere...");
senhaconfere=false;}
else { senhaconfere=true;}}}}
senha="";}}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
deucerto=false;
novamensagem("Nao foi possível conectar, tente mais tarde...");
jPasswordFieldsenha.setText("");}
if (jaexiste && deucerto && senhaconfere) { novamensagem("Conexão com sucesso");
sucesso=true;
jTextFieldid.setText("");
jPasswordFieldsenha.setText("");
setVisible(false);
dispose();}
else { sucesso=false;
apagadados();
jPasswordFieldsenha.setText("");}}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) {
novamensagem("Operação 'Conexão' cancelada pelo usuário");
sucesso=false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
sucesso=false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
public static void main(String args[]) { new conexao(new javax.swing.JFrame(), true).show();}
protected void apagatemplates() { jTextFieldid.setText("");
jPasswordFieldsenha.setText("");}
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
public javax.swing.JTextField jTextFieldid;
public javax.swing.JButton jButtonok;
private javax.swing.JDialog jDialog1;
public javax.swing.JPasswordField jPasswordFieldsenha;
private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabel5;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;

```

```

public class cadcriterio extends dialogoabstrato { protected String squanttopicos,
squantquestoestopatual, squantquestoestot, snumtopicoatual, snumquestaoatopatual,
snumquestaoatualtotal, snummaxqueststop;
protected int quanttopicos, quantquestoestopatual, quantquestoestot, numtopicoatual,
numquestaoatopatual, numquestaoatualtotal, nummaxqueststop;
protected String codcriterio, criterio;
public cadcriterio(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;
apagatemplates();}
private void initComponents() { jLabel1 = new javax.swing.JLabel();
jButtoncadastrar = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jTextFielddescricao = new javax.swing.JTextField();
jLabel21 = new javax.swing.JLabel();
jTextFielddescricoes = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jTextFielddporcentagmin = new javax.swing.JTextField();
jPanel2 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
jComboBoxteoria = new javax.swing.JComboBox();
jScrollPane3 = new javax.swing.JScrollPane();
jComboBoxmetodo = new javax.swing.JComboBox();
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Cadastro de Crit\u00e9rio do Usu\u00e1rio");
setModal(true);
setName("dialogcadcriterio");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Cadastro de Crit\u00e9rio do Usu\u00e1rio - Escolha a teoria e o m\u00e9todo para
cadastrar o crit\u00e9rio desejado:");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 20, -1, -1));
jButtoncadastrar.setText("Cadastrar");
jButtoncadastrar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncadastrarActionPerformed(evt);}});
getContentPane().add(jButtoncadastrar, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 440,
130, -1));
jLabel4.setText("Escolha uma teoria de avalia\u00e7\u00e3o");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, -1, -1));
jLabel5.setText("Escolha um m\u00e9todo:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(570, 440,
90, -1));
jLabel2.setText("Descri\u00e7\u00e3o:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 270, -1, -1));
jTextFielddescricao.setBorder(new javax.swing.border.EtchedBorder());
getContentPane().add(jTextFielddescricao, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
290, 610, -1));
jLabel21.setText("Observa\u00e7\u00f5es:");
getContentPane().add(jLabel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 320, -1, -1));

```

```

getContentPane().add(jTextFieldobservacoes, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 340, 610, -1));
jLabel6.setText("Porcentagem m\u00eddnima para aprova\u00e7\u00e3o do plano:");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 370, -1, -1));
jTextFieldporcentagmin.setMinimumSize(new java.awt.Dimension(46, 20));
jTextFieldporcentagmin.setPreferredSize(new java.awt.Dimension(46, 20));
getContentPane().add(jTextFieldporcentagmin, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
390, 46, 20));
jPanel2.setBorder(new javax.swing.border.TitledBorder("Dados do novo crit\u00e9rio"));
getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, 640,
170));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 490, 640,
40));
jComboBoxteoria.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxteoriaItemStateChanged(evt);}});
jScrollPane2.setViewportView(jComboBoxteoria);
getContentPane().add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 90, 640,
50));
jComboBoxmetodo.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxmetodoItemStateChanged(evt);}});
jScrollPane3.setViewportView(jComboBoxmetodo);
getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 180, 640,
50));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(682, 582));
setLocation((screenSize.width-682)/2,(screenSize.height-582)/2);}
private void jComboBoxmetodoItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidademetodos > 0) { codmetodo =
String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]);}
else { codmetodo = "null";}
metodo = "";}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();}}
private void jComboBoxteoriaItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadeteorias > 0) { codteoria =
String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()]);}
preenchemetodo();}
else { codteoria = "null";}
teoria = "";}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();}}
private void formComponentShown(java.awt.event.ComponentEvent evt) { novamensagem("");
apagatemplates();
sucesso = false;
houvealgumsucesso = false;
try { if (preencheteoria()) { preenchemetodo();}}
catch (Exception e) { novamensagem(e.toString());
novamensagem("Não foi possível acessar o banco de dados.");
e.printStackTrace();}}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
if (!houvealgumsucesso) { sucesso = false;};
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Operação 'Cadastro de Critério' concluída.");}

```



```

else { sucesso = false;
novamensagem("Operação 'Cadastro de Critério' cancelada pelo usuário.");}
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void jButtoncadastrarActionPerformed(java.awt.event.ActionEvent evt) { boolean deucerto,
jaexiste;
String descricao, observacoes, porcentagmin;
int valporcentagmin;
deucerto = true;
sucesso = false;
jaexiste = false;
novamensagem("");
try { descricao = jTextFielddescricao.getText().trim();
jTextFielddescricao.setText(descricao);
if (descricao.length() < 4) { novamensagem("A descrição deve possuir pelo menos quatro
caracteres");
deucerto = false;}
else { observacoes = jTextFieldobservacoes.getText().trim();
jTextFieldobservacoes.setText(observacoes);
if (observacoes.length() < 4) { novamensagem("As observações devem possuir pelo menos quatro
caracteres");
deucerto = false;}
else { porcentagmin = jTextFieldporcentagmin.getText().trim();
jTextFieldporcentagmin.setText(porcentagmin);
if (porcentagmin.length() < 1) { novamensagem("A porcentagem mínima deve ser preenchida com um
número inteiro entre 0 e 100");
deucerto = false;}
else { try { valporcentagmin = Integer.parseInt(porcentagmin);
if ((valporcentagmin < 0) | (valporcentagmin > 100 )) { novamensagem("A porcentagem mínima deve
ser um número inteiro entre 0 e 100");
deucerto = false;}
else { try { if (existecriterio(jTextFielddescricao.getText(), codmetodo)) { novamensagem("Já existe um
critério com a mesma descricao para o usuario atual e o método/teoria escolhidos.");
jaexiste = true;}
else { jaexiste = false;}}}
catch (Exception e) { novamensagem(e.toString());
deucerto = false;}}}
catch (Exception e) { novamensagem("A porcentagem mínima deve ser um número inteiro entre 0 e
100");
deucerto = false;}}}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
deucerto = false;
novamensagem("Não foi possível inserir, tente mais tarde...");}
if ((deucerto) & (!jaexiste)) { try { if (inserircritério()) { houvealgumsucesso = true;
jButtoncancelar.setText("Sair");
sucesso = true;
novamensagem("Critério de avaliação inserido com sucesso");}
else { novamensagem("Não foi possível inserir, tente novamente...");}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Não foi possível inserir, tente mais tarde...");}};
if (sucesso) { }}
public static void main(String args[]) { new cadcritério(new javax.swing.JFrame(), true).show();}
protected boolean preencheteoria() throws Exception { return super.preencheteoria(jComboBoxteoria,
jComboBoxmetodo,
null, null, null, jButtoncadastrar,
null);}

```

```

protected boolean preenchemetodo() throws Exception { return
super.preenchemetodo(jComboBoxteoria, jComboBoxmetodo,
null, null, null, jButtoncadastrar,
null);}
protected boolean inseriucriterio() throws Exception { java.sql.Statement stmtcriterio;
java.sql.ResultSet rscriterio;
boolean inseriucriterio;
novamensagem("");
stmtcriterio= con.createStatement();
inseriucriterio = stmtcriterio.execute("INSERT INTO criterio (cod, cod_usuario, cod_metodo,
porcentmin, geral, descr, observacoes) VALUES (NULL, " + atualusuario.getCodusuario() + ", " +
String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]) + ", " +
jTextFieldporcentagmin.getText() + ", 0, " + jTextFieldddescricao.getText() + ", " +
jTextFieldobservacoes.getText() + "");");
stmtcriterio.close();
stmtcriterio = con.createStatement();
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod_usuario = " +
atualusuario.getCodusuario() + " and cod_metodo = "+
String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]) + " and geral =0 and
descr = " + jTextFieldddescricao.getText()+ " and observacoes=" + jTextFieldobservacoes.getText()
+ "");");
rscriterio.last();
inseriucriterio = (rscriterio.getRow() > 0);
if (inseriucriterio) { codcriterio= rscriterio.getString(1);}
else { codcriterio="null";}
rscriterio.close();
stmtcriterio.close();
return inseriucriterio;}
protected void apagadados() { super.apagadados();
quantidadeteorias = 0;
quantidademetodos = 0;}
protected void apagatemplates() { jLabelmensagem.setText("");
jButtuncancelar.setText("Cancelar");
jComboBoxteoria.removeAllItems();
jComboBoxmetodo.removeAllItems();
jTextFieldddescricao.setText("");
jTextFieldobservacoes.setText("");
jTextFieldporcentagmin.setText("");}
protected void desabilitaedicaocriterio() { jTextFieldddescricao.setEnabled(false);
jTextFieldobservacoes.setEnabled(false);
jTextFieldporcentagmin.setEnabled(false);}
protected void habilitaedicaocriterio() { jTextFieldddescricao.setEnabled(true);
jTextFieldobservacoes.setEnabled(true);
jTextFieldporcentagmin.setEnabled(true);}
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel21;
private javax.swing.JTextField jTextFieldobservacoes;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JComboBox jComboBoxmetodo;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JComboBox jComboBoxteoria;
private javax.swing.JButton jButtoncadastrar;
private javax.swing.JLabel jLabel6;
private javax.swing.JTextField jTextFieldddescricao;
private javax.swing.JTextField jTextFieldporcentagmin;
private javax.swing.JScrollPane jScrollPane3;

```

```

private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabel5;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class altsenha extends dialogoabstrato { public altsenha(java.awt.Frame parent, boolean
modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;}
private void initComponents() { jLabel1 = new javax.swing.JLabel();
jButtonok = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jTextFieldid = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jPasswordFieldsenhaant = new javax.swing.JPasswordField();
jPasswordFieldsenha1 = new javax.swing.JPasswordField();
jPasswordFieldsenha2 = new javax.swing.JPasswordField();
jCheckBoxconectar = new javax.swing.JCheckBox();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Alterar\u00e7\u00e3o de Senha de Usu\u00e1rio");
setModal(true);
setName("dialogaltsenha");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Alterar\u00e7\u00e3o de Senha de Usu\u00e1rio");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 20, -1, -1));
jButtonok.setText("Ok");
jButtonok.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonokActionPerformed(evt);}});
getContentPane().add(jButtonok, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 280, 90, -
1));
jLabel4.setText("Identificador:");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, -1, -1));
getContentPane().add(jTextFieldid, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 190, -
1));
jLabel5.setText("Senha anterior:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 110, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(340, 280,
90, -1));
jLabel2.setText("Nova senha:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, -1, -1));
jLabel3.setText("Confirma\u00e7\u00e3o da nova senha:");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 210, -1, -1));
getContentPane().add(jPasswordFieldsenhaant, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 130, 140, -1));

```

```

getContentPane().add(jPasswordFieldsenha1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 180, 140, -1));
getContentPane().add(jPasswordFieldsenha2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
230, 140, -1));
jCheckBoxconectar.setText("Conectar");
getContentPane().add(jCheckBoxconectar, new org.netbeans.lib.awtextra.AbsoluteConstraints(230,
240, 200, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 330, 480,
40));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(520, 420));
setLocation((screenSize.width-520)/2,(screenSize.height-420)/2);}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
sucesso = false;
conectar = false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void formComponentShown(java.awt.event.ComponentEvent evt) { apagatemplates();
jButtoncancelar.setText("Cancelar");
novamensagem("");}
private void jButtonokActionPerformed(java.awt.event.ActionEvent evt) { boolean jaexiste, deucerto,
senhaconfere;
String nome, mail, id, senha1, senha2;
jaexiste = false; deucerto = true; senhaconfere = false; sucesso = false;
novamensagem("");
conectar = ((jCheckBoxconectar.isSelected()) && (jCheckBoxconectar.isEnabled()));
jButtoncancelar.setText("Sair");
try { id = jTextFieldid.getText().trim();
jTextFieldid.setText(id);
if (id.length() < 4) { novamensagem("O identificador deve possuir pelo menos quatro caracteres");
deucerto = false;} else { if (id.length() > 16) { novamensagem("O identificador deve possuir no máximo
dezesseis caracteres");
deucerto = false;} else { senha1 = new String(jPasswordFieldsenha1.getPassword());
senha2 = new String(jPasswordFieldsenha2.getPassword());
if (!senha2.equals(senha1)) { novamensagem("A senha deve ser igual a confirmação da senha");
deucerto = false;} else { if (senha1.length() < 4) { novamensagem("As senhas devem possuir no
mínimo quatro caracteres");
deucerto = false;} else { if (senha1.length() > 16) { novamensagem("As senhas devem possuir no
máximo dezesseis caracteres");
deucerto = false;} else { if (!(existeid(jTextFieldid.getText(), atualusuario.getCodpessoa(), true))) {
novamensagem("Nao existe o usuário com identificador "+ jTextFieldid.getText()+" tente outro!");
jaexiste = false;} else { jaexiste = true;
if (!(confirmaSenha(jTextFieldid.getText(), new String(jPasswordFieldsenhaant.getPassword())))) {
novamensagem("Senha anterior não confere...");
senhaconfere = false;
deucerto = false;} else { senhaconfere = true;}}}}
senha1="";
senha2="";}}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
deucerto = false;
if (conectar) { novamensagem("Não foi possível alterar e conectar, tente mais tarde...");} else {
novamensagem("Não foi possível alterar, tente mais tarde...");}}
if (jaexiste && deucerto && senhaconfere) { try { if (alterasenha()) { jButtoncancelar.setText("Sair");
if (conectar) { sucesso = true;

```

```

novamensagem("Senha alterada e usuario conectado com sucesso");
apagatemplates();} else { novamensagem("Senha alterada com sucesso");
apagatemplates();}} else { if (conectar) { novamensagem("Não foi possível alterar e conectar, tente
novamente...");} else { novamensagem("Não foi possível alterar, tente novamente...");}}
apagasenhas();
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
apagasenhas();
if (conectar) { novamensagem("Não foi possível alterar e conectar, tente mais tarde...");} else {
novamensagem("Não foi possível alterar, tente mais tarde...");}} else { apagasenhas();}
if (sucesso) { setVisible(false);
dispose();}}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(jButtoncancelar.getText().equalsIgnoreCase("Sair")) { novamensagem("Operação 'Alteração de
senha' concluída");}
else { novamensagem("Operação 'Alteração de senha' cancelada pelo usuário");}
sucesso = false;
conectar = false;
apagadados();
apagatemplates();
setVisible(false);
dispose();}
public static void main(String args[]) { new altsenha(new javax.swing.JFrame(), true).show();}
protected boolean alterasenha() throws Exception { java.sql.Statement stmtusuario;
java.sql.ResultSet rsusuario;
boolean alterousenha;
String codigopessoa;
String senha;
novamensagem(" ");
stmtusuario= con.createStatement();
senha = new String(jPasswordFieldsenha1.getPassword());
alterousenha = stmtusuario.execute("UPDATE usuario SET senha = MD5('"+ senha +"') WHERE id
='"+ jTextFieldid.getText() +"'");
stmtusuario.close();
stmtusuario = con.createStatement();
rsusuario = stmtusuario.executeQuery("select cod from usuario where id ='"+ jTextFieldid.getText() + "'
and senha = MD5('"+ senha +"')");
senha = "";
rsusuario.last();
alterousenha = (rsusuario.getRow() > 0);
rsusuario.close();
stmtusuario.close();
return alterousenha;}
protected void apagatemplates() { jTextFieldid.setText("");
apagasenhas();}
protected void apagasenhas() { jPasswordFieldsenhaant.setText("");
jPasswordFieldsenha1.setText("");
jPasswordFieldsenha2.setText("");}
public void setconectado(boolean conectado) { estavaconectado = conectado;
if (estavaconectado) { jCheckBoxconectar.setSelected(false);
jCheckBoxconectar.setEnabled(false);}
else { jCheckBoxconectar.setSelected(false);
jCheckBoxconectar.setEnabled(true);}}
private javax.swing.JCheckBox jCheckBoxconectar;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JPasswordField jPasswordFieldsenha1;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;

```

```

private javax.swing.JTextField jTextFieldid;
private javax.swing.JLabel jLabel2;
private javax.swing.JPasswordField jPasswordFieldsenhaant;
private javax.swing.JPasswordField jPasswordFieldsenha2;
private javax.swing.JButton jButtonok;
private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabel5;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class avaliar extends dialogoabstrato { protected String squanttopicos, squantquestoestopatual,
squantquestoestot, snumtopicoatual, snumquestaoatopatual, snumquestaoatualtotal,
snummaxquesttop;
protected int quanttopicos, quantquestoestopatual, quantquestoestot, numtopicoatual,
numquestaoatopatual, numquestaoatualtotal, nummaxquesttop;
protected int numquestoesportopico[];
protected String acao, savancar, svoltar, scancelar, sconcluir;
protected int deltatopico, deltaquestaoatot, deltaquestaoatop;
boolean voltoutopico, voltouquestao;
boolean atualizmsgbtsimnaoalterado;
int respostasquestionario[][];
public avaliar(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;
dimensionaestajanela();
dimensionadialogo();
apagatemplates();
apagatemplatesdialogo();
svoltar = "voltar";
savancar = "avancar";
sconcluir = "concluir";
scancelar = "cancelar";
acao = "";
houvealgumsucesso = false;
atualizmsgbtsimnaoalterado = true;}
private void initComponents() { jDialogrespondequestionario = new javax.swing.JDialog();
jLabelteoria = new javax.swing.JLabel();
jLabelautor = new javax.swing.JLabel();
jLabelidiomateoria = new javax.swing.JLabel();
jLabelano = new javax.swing.JLabel();
jLabelmetodo = new javax.swing.JLabel();
jLabeltitquestionario = new javax.swing.JLabel();
jLabelidiomametodo = new javax.swing.JLabel();
jLabeltopicos = new javax.swing.JLabel();
jLabelquestoes = new javax.swing.JLabel();
jLabeltopicoatual = new javax.swing.JLabel();
jLabelquestaoatual = new javax.swing.JLabel();
jRadioButtonsim = new javax.swing.JRadioButton();
jRadioButtonsnao = new javax.swing.JRadioButton();
jLabelresposta = new javax.swing.JLabel();
jButtonavancardialogo = new javax.swing.JButton();
jButtonairdialogo = new javax.swing.JButton();
jButtonvoltardialogo = new javax.swing.JButton();
jButtonconcluidialogo = new javax.swing.JButton();
jLabeltittopico = new javax.swing.JLabel();

```

```

jCheckBoxsalvaraosair = new javax.swing.JCheckBox();
jScrollPane2 = new javax.swing.JScrollPane();
jLabelmensagem1 = new javax.swing.JLabel();
jScrollPane6 = new javax.swing.JScrollPane();
jTextAreatopico = new javax.swing.JTextArea();
jScrollPane7 = new javax.swing.JScrollPane();
jTextAreaquestao = new javax.swing.JTextArea();
buttonGroupresposta = new javax.swing.ButtonGroup();
jLabel1 = new javax.swing.JLabel();
jButtonavaliar = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
jComboBoxteoria = new javax.swing.JComboBox();
jScrollPane4 = new javax.swing.JScrollPane();
jComboBoxmetodo = new javax.swing.JComboBox();
jScrollPane5 = new javax.swing.JScrollPane();
jComboBoxplanogov = new javax.swing.JComboBox();
jDialogrespondequestionario.getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
jDialogrespondequestionario.setTitle("Question\u00e1rio");
jDialogrespondequestionario.setModal(true);
jDialogrespondequestionario.setName("dialogquestionario");
jDialogrespondequestionario.addComponentListener(new java.awt.event.ComponentAdapter() { public
void componentShown(java.awt.event.ComponentEvent evt) {
jDialogrespondequestionarioComponentShown(evt);}});
jDialogrespondequestionario.addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) {
jDialogrespondequestionarioWindowClosing(evt);}});
jLabelteoria.setText("Teoria:");
jDialogrespondequestionario.getContentPane().add(jLabelteoria, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 600, -1));
jLabelautor.setText("Autor:");
jDialogrespondequestionario.getContentPane().add(jLabelautor, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 460, -1));
jLabelidiomateoria.setText("Idioma Orginal:");
jDialogrespondequestionario.getContentPane().add(jLabelidiomateoria, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 470, -1));
jLabelano.setText("Ano:");
jDialogrespondequestionario.getContentPane().add(jLabelano, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 150, -1));
jLabelmetodo.setText("M\u00e9todo:");
jDialogrespondequestionario.getContentPane().add(jLabelmetodo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 600, -1));
jLabeltitquestionario.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltitquestionario.setText("Quest\u00e3o:");
jDialogrespondequestionario.getContentPane().add(jLabeltitquestionario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 280, 130, -1));
jLabelidiomametodo.setText("Idioma do M\u00e9todo:");
jDialogrespondequestionario.getContentPane().add(jLabelidiomametodo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, 470, -1));
jLabeltopicos.setText("Quantidade de t\u00f3picos:");
jDialogrespondequestionario.getContentPane().add(jLabeltopicos, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 200, -1));
jLabelquestoes.setText("Quantidade total de quest\u00f5es:");

```

```

jDialogrespondequestionario.getContentPane().add(jLabelquestoes, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 80, 230, -1));
jLabeltopicoatual.setText("\u00f3pico Atual/Total: 0/0");
jDialogrespondequestionario.getContentPane().add(jLabeltopicoatual, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, 140, -1));
jLabelquestaoatual.setText("Quest\u00e3o Atual do \u00f3pico/Total no \u00f3pico: 0/0
Quest\u00e3o Atual Total/Total Geral: 0/0");
jDialogrespondequestionario.getContentPane().add(jLabelquestaoatual, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 160, 500, -1));
jRadioButtonsim.setMnemonic('S');
jRadioButtonsim.setText("Sim");
jRadioButtonsim.setToolTipText("Resposta: sim");
buttonGroupresposta.add(jRadioButtonsim);
jRadioButtonsim.addChangeListener(new javax.swing.event.ChangeListener() { public void
stateChanged(javax.swing.event.ChangeEvent evt) { jRadioButtonsimStateChanged(evt);}});
jDialogrespondequestionario.getContentPane().add(jRadioButtonsim, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 470, -1, -1));
jRadioButtonnao.setMnemonic('N');
jRadioButtonnao.setText("N\u00e3o");
jRadioButtonnao.setToolTipText("Resposta: n\u00e3o");
buttonGroupresposta.add(jRadioButtonnao);
jRadioButtonnao.addChangeListener(new javax.swing.event.ChangeListener() { public void
stateChanged(javax.swing.event.ChangeEvent evt) { jRadioButtonnaoStateChanged(evt);}});
jDialogrespondequestionario.getContentPane().add(jRadioButtonnao, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 490, -1, -1));
jLabelresposta.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelresposta.setText("Resposta:");
jDialogrespondequestionario.getContentPane().add(jLabelresposta, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 450, -1, -1));
jButtonavancardialogo.setMnemonic('A');
jButtonavancardialogo.setText("Avan\u00e7ar");
jButtonavancardialogo.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonavancardialogoActionPerformed(evt);}});
jDialogrespondequestionario.getContentPane().add(jButtonavancardialogo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 490, -1, -1));
jButtonsairdialogo.setText("Sair");
jButtonsairdialogo.setDefaultCapable(false);
jButtonsairdialogo.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonairsairdialogoActionPerformed(evt);}});
jDialogrespondequestionario.getContentPane().add(jButtonsairdialogo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 490, -1, -1));
jButtonvoltardialogo.setMnemonic('V');
jButtonvoltardialogo.setText("Voltar");
jButtonvoltardialogo.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonvoltardialogoActionPerformed(evt);}});
jDialogrespondequestionario.getContentPane().add(jButtonvoltardialogo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 490, -1, -1));
jButtonconcluirdialogo.setText("Concluir");
jButtonconcluirdialogo.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonconcluirdialogoActionPerformed(evt);}});
jDialogrespondequestionario.getContentPane().add(jButtonconcluirdialogo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 490, -1, -1));
jLabeltittopico.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltittopico.setText("\u00f3pico:");
jDialogrespondequestionario.getContentPane().add(jLabeltittopico, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 190, 130, -1));
jCheckBoxsalvaraosair.setSelected(true);
jCheckBoxsalvaraosair.setText("Salvar respostas ao sair sem concluir");
jDialogrespondequestionario.getContentPane().add(jCheckBoxsalvaraosair, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 460, -1, -1));

```



```

jScrollPane2.setViewportViewView(jLabelmensagem1);
jDialogrespondequestionario.getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 530, 660, 40));
jTextAreatopico.setEditable(false);
jScrollPane6.setViewportViewView(jTextAreatopico);
jDialogrespondequestionario.getContentPane().add(jScrollPane6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 210, 660, 60));
jScrollPane7.setViewportViewView(jTextAreaquestao);
jDialogrespondequestionario.getContentPane().add(jScrollPane7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 300, 660, 140));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Avalia\u00e7\u00e3o de Planos de Governo");
setModal(true);
setName("dialogavaliar");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Avalia\u00e7\u00e3o de Planos de Governo - Escolha a teoria o m\u00e9todo e o
plano que ser\u00e1 avaliado.");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 20, -1, -1));
jButtonavaliar.setText("Avaliar");
jButtonavaliar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonavaliarActionPerformed(evt);}});
getContentPane().add(jButtonavaliar, new org.netbeans.lib.awtextra.AbsoluteConstraints(470, 330,
90, -1));
jLabel4.setText("Escolha uma teoria de avalia\u00e7\u00e3o");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 50, -1, -1));
jLabel5.setText("Escolha um m\u00e9todo");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 330,
90, -1));
jLabel2.setText("Escolha o plano de governo que ser\u00e1 avaliado.");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 230, 270, -1));
jScrollPane1.setViewportViewView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 380, 660,
40));
jComboBoxteoria.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxteoriaItemStateChanged(evt);}});
jScrollPane3.setViewportViewView(jComboBoxteoria);
getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, 660,
50));
jComboBoxmetodo.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxmetodoItemStateChanged(evt);}});
jScrollPane4.setViewportViewView(jComboBoxmetodo);
getContentPane().add(jScrollPane4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, 660,
50));
jComboBoxplanogov.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxplanogovItemStateChanged(evt);}});
jScrollPane5.setViewportViewView(jComboBoxplanogov);
getContentPane().add(jScrollPane5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, 660,
50));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, 590));
setLocation((screenSize.width-700)/2,(screenSize.height-590)/2);}

```

```

private void jComboBoxteoriaItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadeteorias > 0) { codteoria =
String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()]);
if (preenchemetodo()) { preencheplanogov();}}else { codteoria = "null";}
teoria = "";}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();}}
private void jComboBoxplanogovItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadeplanogovs > 0) { codplanogov =
String.valueOf(codComboBoxplanogov[jComboBoxplanogov.getSelectedIndex()]);}
else { codplanogov = "null";}
planogov = "";}
catch (Exception e) { novamensagem1(e.toString());
e.printStackTrace();}}
private void jComboBoxmetodoItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidademetodos > 0) { codmetodo =
String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]);
preencheplanogov();}
else { codmetodo = "null";}
metodo = "";}
catch (Exception e) { novamensagem1(e.toString());
e.printStackTrace();}}
private void jRadioBottonaoStateChanged(javax.swing.event.ChangeEvent evt) { if
(atualizmsgbtsimnaoalterado) { atualizarestadobotoes();}}
private void jRadioButtonsimStateChanged(javax.swing.event.ChangeEvent evt) { if
(atualizmsgbtsimnaoalterado) { atualizarestadobotoes();}}
private void jButtonvoltardialogoActionPerformed(java.awt.event.ActionEvent evt) { acao = svoltar;
atualizatemplatesdialogo();
jDialogrespondequestionario.setVisible(false);
jDialogrespondequestionario.dispose();}
private void jButtonavancardialogoActionPerformed(java.awt.event.ActionEvent evt) { acao =
savancar;
atualizatemplatesdialogo();
jDialogrespondequestionario.setVisible(false);
jDialogrespondequestionario.dispose();}
private void jDialogrespondequestionarioComponentShown(java.awt.event.ComponentEvent evt) {
atualizatemplatesdialogo();}
private void jDialogrespondequestionarioWindowClosing(java.awt.event.WindowEvent evt) { if
((jCheckBoxsalvaraosair.isSelected())&(jCheckBoxsalvaraosair.isEnabled())) { try {
atualizabasepelasrespostasquestoes();
novamensagem1("Usuário fechou a janela. Respostas salvas.");}
catch (Exception e) { novamensagem1("Usuário fechou a janela. Houve erro salvando as respostas!");
e.printStackTrace();}} else { novamensagem1("Usuário fechou a janela. Respostas não foram
alteradas.");}
apagatemplatesdialogo();
acao = scancelar;
sucesso = false;
jDialogrespondequestionario.setVisible(false);
jDialogrespondequestionario.dispose();}
private void jButtonsaialogoActionPerformed(java.awt.event.ActionEvent evt) { if
((jCheckBoxsalvaraosair.isSelected())&(jCheckBoxsalvaraosair.isEnabled())) { try {
atualizabasepelasrespostasquestoes();
novamensagem1("Usuário abandonou a operação. Respostas parciais salvas.");} catch (Exception e)
{ novamensagem1("Usuário abandonou a operação. Houve erro salvando as respostas!");
e.printStackTrace();}}
else { novamensagem1("Usuário abandonou a operação. Respostas não foram alteradas.");}
apagatemplatesdialogo();
acao = scancelar;
sucesso = false;
jDialogrespondequestionario.setVisible(false);

```

```

jDialogrespondequestionario.dispose();}
private void jButtonconcluidialogoActionPerformed(java.awt.event.ActionEvent evt) { acao =
sconcluir;
novamensagem1("Avaliação salva.");
try { atualizabasepelasrespostasquestoes();
jButtoncancelar.setText("Sair");
sucesso = true;
houvealgumsucesso = true;}
catch (Exception e) { novamensagem1("Não foi possível salvar a avaliação!");
e.printStackTrace();
sucesso = false;}
apagatemplatesdialogo();
jDialogrespondequestionario.setVisible(false);
jDialogrespondequestionario.dispose();}
private void formComponentShown(java.awt.event.ComponentEvent evt) { novamensagem("");
apagatemplates();
apagatemplatesdialogo();
sucesso = false;
houvealgumsucesso = false;
atualizmsgbtsimnaoalterado = true;
try { if (preencheoria()) { if (preenchemetodo()) { preencheplanogov();}}}
catch (Exception e) { novamensagem(e.toString());
novamensagem("Não foi possível acessar o banco de dados.");
e.printStackTrace();}}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
if (!houvealgumsucesso) { sucesso = false;};
apagadados();
apagatemplates();
apagatemplatesdialogo();
setVisible(false);
dispose();}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Operação 'Avaliação de plano' concluída.");}
else { sucesso = false;
novamensagem("Operação 'Avaliação de plano' cancelada pelo usuário.");}
apagadados();
apagatemplates();
apagatemplatesdialogo();
setVisible(false);
dispose();}
private void jButtonavaliarActionPerformed(java.awt.event.ActionEvent evt) { java.sql.Statement
stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rsavaliacao;
try { preenchevariaveis();
novamensagem1("");
novamensagem("");
if (atualizarespostasquestoespelabase()) { stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
acao = savancar;
sucesso = false;
numtopicoatual =0;

```

```

numquestaoatualtotal = 0;
deltatopico = 0;
deltaquestaoatot = 0;
deltaquestaoatop = 0;
while (continuartopico(rstopico)) { numtopicoatual = numtopicoatual + deltatopico;
snumtopicoatual = String.valueOf(numtopicoatual);
jLabeltopicoatual.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
rsquestao = stmtquestao.executeQuery("select * from questao where cod_topico=" +
rstopico.getString(1) + " order by ordem;");
rsquestao.last();
quantquestoestopatual = rsquestao.getRow();
squantquestoestopatual = String.valueOf(quantquestoestopatual);
rsquestao.beforeFirst();
numquestaoatopatual = 0;
while (continuarquestao(rsquestao)) { numquestaoatopatual = numquestaoatopatual + deltaquestaoatop;
snumquestaoatopatual = String.valueOf(numquestaoatopatual);
numquestaoatualtotal = numquestaoatualtotal + deltaquestaoatot;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelquestaoatual.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatopatual + "/"
+ squantquestoestopatual + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestot);
jTextAreatopico.setText(rstopico.getString(6));
jTextAreakquestao.setText(rsquestao.getString(5));
jTextAreatopico.setCaretPosition(1);
jTextAreakquestao.setCaretPosition(1);
jDialogrespondequestionario.show();
jTextAreatopico.setText("");
jTextAreakquestao.setText("");}
rsquestao.close();}
novamensagem(mensagem1);
rstopico.close();
stmtquestao.close();
stmttopico.close();}
else { novamensagem("Não foram encontradas questões do método e teoria escolhidos para serem
respondidas. É necessário que sejam cadastradas questões para que se possa avaliar um plano.");}
catch (Exception e) { novamensagem(e.toString());
novamensagem1(e.toString());
e.printStackTrace();}
protected boolean faltaresponderquestao() { int locali, localj;
boolean resultado;
locali = 0;
resultado = false;
while (!(resultado) & (locali < quanttopicos)) { localj = 0;
while (!(resultado) & (localj < numquestoesportopico[locali])) { if ((respostasquestionario[locali][localj]
!= 1) & (respostasquestionario[locali][localj] != -1)) { resultado = true;};
localj++;}
locali++;}
return resultado;}
protected boolean continuartopico(java.sql.ResultSet rstopico) throws Exception { boolean resultado;
voltoutopico = false;
if (acao == savancar) { resultado = rstopico.next();
deltatopico = 1;} else
if ((acao == svoltar) & (numtopicoatual > 1)) { resultado = rstopico.previous();
voltoutopico = true;
deltatopico = -1;}
else { resultado = false;}
return resultado;}
protected boolean continuarquestao(java.sql.ResultSet rsquestao) throws Exception { boolean
resultado;
voltouquestao = false;

```

```

if (acao == savancar) { resultado = rsquestao.next();
deltaquestaoTot = 1;
deltaquestaoTop = 1;}
else if (acao == svoltar) { if (voltoutopico) { resultado = rsquestao.last();
voltouquestao = true;
deltaquestaoTop = rsquestao.getRow() - numquestaoAtual;
deltaquestaoTot = -1;}
else { if (numquestaoAtual > 1) { resultado = rsquestao.previous();
voltouquestao = true;
deltaquestaoTop = -1;
deltaquestaoTot = -1;} else { resultado = false;}}}
else { resultado = false;}
voltoutopico = false;
return resultado;}
public static void main(String args[]) { new avaliar(new javax.swing.JFrame(), true).show();}
public void novamensagem1(String msg) { mensagem1 = msg;
jLabelmensagem1.setText(msg);}
protected boolean preencheTeoria() throws Exception { return super.preencheTeoria(jComboBoxteoria,
jComboBoxmetodo,
jComboBoxplanogov, null, null, jButtonavaliar,
null);}
protected boolean preencheMetodo() throws Exception { return
super.preencheMetodo(jComboBoxteoria, jComboBoxmetodo,
jComboBoxplanogov, null, null, jButtonavaliar,
null);}
protected boolean preenchePlanogov() throws Exception { return
super.preenchePlanogov(jComboBoxplanogov, null, jButtonavaliar);}
void preencheVariaveis() throws Exception { java.sql.Statement stmtTeoria;
java.sql.Statement stmtMetodo;
java.sql.Statement stmtIdiomat;
java.sql.Statement stmtIdiomam;
java.sql.Statement stmtPessoa;
java.sql.Statement stmtTopico;
java.sql.Statement stmtQuestao;
java.sql.ResultSet rsteoria;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidiomat;
java.sql.ResultSet rsidiomam;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
novamensagem("");
stmtTeoria = con.createStatement();
stmtMetodo = con.createStatement();
stmtIdiomat = con.createStatement();
stmtIdiomam = con.createStatement();
stmtPessoa = con.createStatement();
stmtTopico = con.createStatement();
stmtQuestao = con.createStatement();
rsteoria = stmtTeoria.executeQuery("select * from teoria where
cod="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsteoria.first();
rspessoa = stmtPessoa.executeQuery("select * from pessoa where cod="+rsteoria.getString(2)+";");
rspessoa.first();
rsidiomat = stmtIdiomat.executeQuery("select * from idioma where cod="+rsteoria.getString(3)+";");
rsidiomat.first();
rsmetodo = stmtMetodo.executeQuery("select * from metodo where
cod="+String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()])+" and
cod_teorias="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsmetodo.first();

```

```

rsidiomam = stmtidiomam.executeQuery("select * from idioma where
cod="+rsmetodo.getString(3)+");
rsidiomam.first();
rstopico = stmttopico.executeQuery("select * from topico where
cod_metodo="+rsmetodo.getString(1)+" order by ordem;");
rstopico.last();
quanttopicos = rstopico.getRow();
numquestoesportopico = new int[quanttopicos];
squanttopicos = String.valueOf(quanttopicos);
if (quanttopicos > 0) { numtopicoatual = 1;
rstopico.first();
rsquestao = stmtquestao.executeQuery("select * from questao where cod_topico="+
rstopico.getString(1) + " order by ordem;");
rsquestao.last();
quantquestoestopatual = rsquestao.getRow();
rsquestao.close();
quantquestoestot = 0;
nummaxquesttop = 0;
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select * from questao where
cod_topico="+rstopico.getString(1)+" order by ordem;");
rsquestao.last();
numquestoesportopico[rstopico.getRow()-1] = rsquestao.getRow();
if (rsquestao.getRow() > nummaxquesttop) { nummaxquesttop = rsquestao.getRow();}
quantquestoestot = quantquestoestot + rsquestao.getRow();
rsquestao.close();}
rstopico.close();}
else { numtopicoatual = 0;
quantquestoestopatual = 0;
quantquestoestot = 0;
nummaxquesttop = 0;}
snumtopicoatual = String.valueOf(numtopicoatual);
squantquestoestopatual = String.valueOf(quantquestoestopatual);
snummaxquesttop = String.valueOf(nummaxquesttop);
squantquestoestot = String.valueOf(quantquestoestot);
codteoria = rteoria.getString(1);
teoria = rteoria.getString(4);
codmetodo = rsmetodo.getString(1);
metodo = rsmetodo.getString(4);
idiomat = rsidiomat.getString(3);
ano_t = rteoria.getString(6);
idiomam = rsidiomam.getString(3);
pessoa = rspessoa.getString(2);
codplanogov = String.valueOf(codComboBoxplanogov[jComboBoxplanogov.getSelectedIndex()]);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
numquestaoatopatual = 1;
snumquestaoatopatual = String.valueOf(numquestaoatopatual);
jLabelteoria.setText("Teoria: " + teoria);
jLabelautor.setText("Autor: " + pessoa);
jLabelidiomateoria.setText("Idioma da teoria: " + idiomat);
jLabelano.setText("Ano da teoria: " + ano_t);
jLabelmetodo.setText("Método usado: " + metodo);
jLabelidiomametodo.setText("Idioma do método: " + idiomam);
jLabeltopicos.setText("Quantidade de tópicos: " + squanttopicos);
jLabelquestoes.setText("Quantidade total de questões: " + squantquestoestot);
jLabeltopicoatual.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
jLabelquestaoatual.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatopatual + "/"
+ squantquestoestopatual + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestot);

```

```

rpessoa.close();
rsidiomam.close();
rsidiomat.close();
rsmetodo.close();
rsteoria.close();
stmtquestao.close();
stmpessoa.close();
stmtidiomam.close();
stmtidiomat.close();
stmtmetodo.close();
stmtteoria.close();
protected void dimensionaestajanela() { java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, (jScrollPane1.getY() + jScrollPane1.getHeight() + 50)));
setLocation((screenSize.width-700)/2,(screenSize.height-(jScrollPane1.getY() +
jScrollPane1.getHeight() + 50))/2);}
protected void dimensionadialogo() { java.awt.Dimension tamanhotela =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
jDialogrespondequestionario.setSize(new java.awt.Dimension(700, 620));
jDialogrespondequestionario.setLocation((tamanhotela.width-700)/2,(tamanhotela.height-620)/2);
jDialogrespondequestionario.setTitle("Questionário");}
protected boolean atualizarespostasquestoespelabase() throws Exception { java.sql.Statement
stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.ResultSet rstopico;
java.sql.ResultSet rquestao;
java.sql.ResultSet rsavaliacao;
int locali, localj;
int respostaavaliacao, numerotopico, numeroquestao;
boolean hatopico, haquestao;
haquestao = false;
hatopico = false;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
respostasquestionario = new int[quanttopicos][nummaxquesttop];
locali = 0;
while (locali < quanttopicos) { localj = 0;
while (localj < nummaxquesttop) { respostasquestionario[locali][localj] = 0;
localj++;}
locali++;}
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.last();
if (rstopico.getRow() > 0) { hatopico = true;
rstopico.beforeFirst();
while (rstopico.next()) { rquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rquestao.last();
if (rquestao.getRow() > 0) { haquestao = true;
rquestao.beforeFirst();
while (rquestao.next()) { rsavaliacao = stmtavaliacao.executeQuery("select * from avaliacao where
cod_questao=" + rquestao.getString(1) + " and cod_plano=" + codplanogov + ";");
rsavaliacao.last();
numerotopico = rstopico.getRow();
numeroquestao = rquestao.getRow();
if (rsavaliacao.getRow() > 0) { respostaavaliacao = rsavaliacao.getInt(4);
if ((respostaavaliacao >= -1) & (respostaavaliacao <= 1)) { respostasquestionario[numerotopico-
1][numeroquestao-1] = respostaavaliacao;}}};

```

```

rsavaliacao.close();});
rsquestao.close();});
rstopico.close();
return (hatopico & haquestao);}
protected void atualizarespostaquestao pelos botoes sim nao() { if (jRadioButtonsim.isSelected()) {
respostasquestionario[numtopicoatual-1][numquestaoatual-1] = 1;}
else { if (jRadioButtonsnao.isSelected()) { respostasquestionario[numtopicoatual-
1][numquestaoatual-1] = -1;}
else { respostasquestionario[numtopicoatual-1][numquestaoatual-1] = 0;}}}
protected void atualizabotoessimnao pelarespostaquestao() { if
(respostasquestionario[numtopicoatual-1][numquestaoatual-1] == 1) {
jRadioButtonsim.setSelected(true);
jRadioButtonsnao.setSelected(false);}
else { jRadioButtonsim.setSelected(false);
jRadioButtonsnao.setSelected(true);}
protected void atualizabase pelas respostas questoes() throws Exception { java.sql.Statement
stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rsavaliacao;
int rsavaliacaoint;
boolean rsavaliacaobool;
int locali, localj;
int respostausuario, numerotopico, numeroquestao;
String respostausuarios;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rsquestao.beforeFirst();
while (rsquestao.next()) { rsavaliacao = stmtavaliacao.executeQuery("select * from avaliacao where
cod_questao=" + rsquestao.getString(1) + " and cod_plano=" + codplanogov + ";");
rsavaliacao.last();
numerotopico = rstopico.getRow();
numeroquestao = rsquestao.getRow();
respostausuario = respostasquestionario[numerotopico-1][numeroquestao-1];
respostausuarios = String.valueOf(respostausuario);
if (rsavaliacao.getRow() > 0) { rsavaliacaoint = stmtavaliacao.executeUpdate("update avaliacao set
resposta = " + respostausuarios + " where cod_questao = " + rsquestao.getString(1) + " and
cod_plano=" + codplanogov + ";");}
else { rsavaliacaobool = stmtavaliacao.executeUpdate("INSERT INTO avaliacao (cod, cod_questao,
cod_plano, resposta ) VALUES (NULL, " + rsquestao.getString(1) + ", " + codplanogov + ", " +
respostausuarios + ");");}
rsavaliacao.close();}
rsquestao.close();}
protected void atualizatemplatesdialogo() { boolean aindafalta;
novamensagem1("");
aindafalta = faltaresponderquestao();
if (respostasquestionario[numtopicoatual-1][numquestaoatual-1] == 0) { apagaopcao(false);}
else { atualizabotoessimnao pelarespostaquestao();}
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagem1("Há questões não respondidas.");}
jButtonvoltardialogo.setEnabled(false);}
else { jButtonvoltardialogo.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogo.setEnabled(false);}

```



```

if (aindafalta) { novamensagem1("Há questões não respondidas.");};
else { jButtonavancardialogo.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogo.setEnabled(false);}
jCheckBoxsalvaraosair.setEnabled(true);}
else { jButtonconcluidialogo.setEnabled(true);}
jCheckBoxsalvaraosair.setEnabled(false);}
novamensagem1("Todas questões respondidas.");};
protected void atualizarestadobotoes() { boolean aindafalta;
novamensagem1("");
atualizarespostaquestaope losbotoessimnao();
aindafalta = faltaresponderquestao();
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagem1("Há questões não respondidas.");};
jButtonvoltardialogo.setEnabled(false);}
else { jButtonvoltardialogo.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogo.setEnabled(false);}
if (aindafalta) { novamensagem1("Há questões não respondidas.");};
else { jButtonavancardialogo.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogo.setEnabled(false);}
jCheckBoxsalvaraosair.setEnabled(true);}
else { jButtonconcluidialogo.setEnabled(true);}
jCheckBoxsalvaraosair.setEnabled(false);}
novamensagem1("Todas questões respondidas.");};
public boolean gethouvealgumsucesso() { return houvealgumsucesso;}
public String getmensagem1() { return mensagem1;}
protected void apagadados() { super.apagadados();
quantidadeteorias = 0;
quantidademetodos = 0;
quantidadeplanogovs = 0;}
protected void apagatemplatesdialogo() { jLabelmensagem1.setText("");
jLabelteoria.setText("Teoria: ");
jLabelautor.setText("Autor: ");
jLabelidiomateoria.setText("Idioma da teoria: ");
jLabelano.setText("Ano da teoria: ");
jLabelmetodo.setText("Método usado: ");
jLabelidiomametodo.setText("Idioma do método: ");
jLabeltopicos.setText("Quantidade de tópicos: ");
jLabelquestoes.setText("Quantidade total de questões: ");
jLabeltopicoatual.setText("Tópico Atual/Total: 0/0");
jLabelquestaoatual.setText("Questão Atual do Tópico/Total no Tópico: 0/0 Questão Atual Total/Total
Geral: 0/0");
jTextAreatopico.setText("");
jTextAreaquestao.setText("");
apagaopcao(false);}
protected void apagatemplates() { jLabelmensagem.setText("");
jLabelmensagem1.setText("");
jButtoncancelar.setText("Cancelar");
jComboBoxplanogov.removeAllItems();
jComboBoxteoria.removeAllItems();
jComboBoxmetodo.removeAllItems();}
protected void apagaopcao(boolean atualizarmensagem) { boolean opcaoaterarmsganterior;
opcaoaterarmsganterior = atualizmsgbtsimnaoalterado;
atualizmsgbtsimnaoalterado = atualizarmensagem;
buttonGroupresposta.remove(jRadioButtonsim);
buttonGroupresposta.remove(jRadioButtonnao);
jRadioButtonsim.setSelected(false);
jRadioButtonnao.setSelected(false);
buttonGroupresposta.add(jRadioButtonsim);
buttonGroupresposta.add(jRadioButtonnao);
atualizmsgbtsimnaoalterado = opcaoaterarmsganterior;}
private javax.swing.JLabel jLabel4;

```

```

private javax.swing.JTextArea jTextAreaquestao;
private javax.swing.JLabel jLabelidiomametodo;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JComboBox jComboBoxmetodo;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabelautor;
private javax.swing.JLabel jLabeltittopico;
private javax.swing.JLabel jLabelmensagem1;
private javax.swing.JButton jButtonavaliar;
private javax.swing.JComboBox jComboBoxplanogov;
private javax.swing.ButtonGroup buttonGroupresposta;
private javax.swing.JButton jButtonvoltardialogo;
private javax.swing.JLabel jLabel2;
private javax.swing.JRadioButton jRadioBottonao;
private javax.swing.JLabel jLabelmetodo;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JLabel jLabelteoria;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JLabel jLabeltopicos;
private javax.swing.JLabel jLabelquestaoatual;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JRadioButton jRadioButtonsim;
private javax.swing.JComboBox jComboBoxteoria;
private javax.swing.JLabel jLabeltitquestionario;
private javax.swing.JButton jButtonavancardialogo;
private javax.swing.JTextArea jTextAreatopico;
private javax.swing.JLabel jLabelquestoes;
private javax.swing.JScrollPane jScrollPane5;
private javax.swing.JLabel jLabeltopicoatual;
private javax.swing.JButton jButtononsairdialogo;
private javax.swing.JLabel jLabelano;
private javax.swing.JLabel jLabelidiomateoria;
private javax.swing.JLabel jLabelresposta;
private javax.swing.JDialog jDialogrespondequestionario;
private javax.swing.JCheckBox jCheckBoxsalvaraosair;
private javax.swing.JButton jButtonconcluidialogo;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane6;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class cadplano extends dialogoabstrato { protected String descrplano;
protected String squanttopicos, squantquestoestopatual, squantquestoestot, snumtopicoatual,
snumquestaoatualtop, snumquestaoatualtot;
protected int quanttopicos, quantquestoestopatual, quantquestoestot, numtopicoatual,
numquestaoatual, numquestaoatualtopatual;
protected boolean cancelar;
public cadplano(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;}
private void initComponents() { jLabel1 = new javax.swing.JLabel();
jButtoncadastrar = new javax.swing.JButton();

```

```

jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jTextFielddescriplano = new javax.swing.JTextField();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Cadastro de Plano de Governo");
setModal(true);
setName("dialogcadplano");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Cadastro de Plano de Governo para avalia\u00e7\u00e3o");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, -1, -1));
jButtoncadastrar.setText("Cadastrar");
jButtoncadastrar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncadastrarActionPerformed(evt);}});
getContentPane().add(jButtoncadastrar, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 130,
100, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(380, 130,
100, -1));
jLabel2.setText("Nome do Plano de Governo:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 160, -1));
getContentPane().add(jTextFielddescriplano, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
80, 460, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 180, 460,
40));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(504, 272));
setLocation((screenSize.width-504)/2,(screenSize.height-272)/2);
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usu\u00e1rio");
if (!houvealgumsucesso) { sucesso = false;};
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void formComponentShown(java.awt.event.ComponentEvent evt) { apagatemplates();
jButtoncancelar.setText("Cancelar");
sucesso = false;
houvealgumsucesso = false;
novamensagem("");}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Opera\u00e7\u00e3o 'Cadastro de Plano' concl\u00fada");}
else { sucesso = false;
novamensagem("Opera\u00e7\u00e3o 'Cadastro de Plano' cancelada pelo usu\u00e1rio");}
apagadados();
apagatemplates();
setVisible(false);
dispose();}
private void jButtoncadastrarActionPerformed(java.awt.event.ActionEvent evt) { boolean jaexiste,
deucerto;
String descricao;
jaexiste = false;

```

```

deucerto = true;
sucesso = false;
novamensagem("");
try { descricao = jTextFielddescrplano.getText().trim();
jTextFielddescrplano.setText(descricao);
if (descricao.length() < 4) { novamensagem("O nome do plano deve possuir pelo menos quatro
caracteres");
deucerto = false;}
else { if (existedescr()) { novamensagem("Este plano já está cadastrado");
jaexiste = true;}
else { jaexiste = false;}}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
deucerto = false;
novamensagem("Não foi possível inserir, tente mais tarde...");
jTextFielddescrplano.setText(e.getMessage());}
if (!jaexiste && deucerto) { try { if (inseredescr()) { houvealgumsucesso = true;
jButtoncancelar.setText("Sair");
sucesso = true;
novamensagem("Plano de Governo inserido com sucesso");
apagatemplates();
else { novamensagem("Não foi possível inserir, tente novamente...");}}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();
novamensagem("Não foi possível inserir, tente mais tarde...");}}
else { }
if (sucesso) { setVisible(false);
dispose();}}
public static void main(String args[]) { new avaliar(new javax.swing.JFrame(), true).show();}
boolean existedescr() throws Exception { java.sql.Statement stmtplano;
java.sql.ResultSet rsplano;
boolean existeoplano;
String descricao;
novamensagem("");
descricao = jTextFielddescrplano.getText().trim();
stmtplano= con.createStatement();
rsplano = stmtplano.executeQuery("select * from plano where cod_usuario="+
atualusuario.getCodusuario() +" and descr = " + descricao + " ");
rsplano.last();
existeoplano = (rsplano.getRow() > 0);
rsplano.close();
stmtplano.close();
return existeoplano;}
boolean inseredescr() throws Exception { java.sql.Statement stmtplano;
java.sql.ResultSet rsplano;
String descricao;
boolean inseriuadescr;
novamensagem("");
descricao = jTextFielddescrplano.getText().trim();
stmtplano= con.createStatement();
inseriuadescr = stmtplano.executeQuery("INSERT INTO plano (cod, cod_usuario, descr) VALUES (NULL,
"+ atualusuario.getCodusuario() +", "+ descricao + " ");
stmtplano.close();
stmtplano= con.createStatement();
rsplano = stmtplano.executeQuery("select * from plano where cod_usuario="+
atualusuario.getCodusuario() +" and descr = " + descricao + " ");
rsplano.last();
inseriuadescr = (rsplano.getRow() > 0);
rsplano.close();
stmtplano.close();

```

```

return inseriuadescr;}
protected void apagatemplates() { jTextFieldddescrplano.setText("");}
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JLabel jLabel1;
private javax.swing.JTextField jTextFieldddescrplano;
private javax.swing.JLabel jLabel2;
private javax.swing.JButton jButtoncadastrar;
private javax.swing.JButton jButtoncancelar;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class defpesoscrit extends dialogoabstrato { protected String mensagem1;
protected String squanttopicos, squantquestoestopatual, squantquestoestot, snumtopicoatual,
snumquestaotopatual, snumquestaoatualtotal, snummaxquesttop;
protected int quanttopicos, quantquestoestopatual, quantquestoestot, numtopicoatual,
numquestaotopatual, numquestaoatualtotal, nummaxquesttop;
protected int numquestoesportopico[];
protected String codcriterio, criterio, codavaliacao, avaliacao;
protected String acao, savancar, svoltar, scancelar, sconcluir;
protected int deltatopico, deltaquestaotot, deltaquestaotop;
boolean voltoutopico, voltouquestao;
boolean atualizmsgfieldpesoalterado;
String pesosquest[], pesostopic[];
public defpesoscrit(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;
dimensionaestajanela();
dimensionadialogoquest();
dimensionadialogotopic();
apagatemplates();
apagatemplatesdialogoquest();
apagatemplatesdialogotopic();
apagadadosdialogoquest();
apagadadosdialogotopic();
svoltar = "voltar";
savancar = "avancar";
sconcluir = "concluir";
scancelar = "cancelar";
acao = "";
atualizmsgfieldpesoalterado = true;}
private void initComponents() { jDialogdefpesoquest = new javax.swing.JDialog();
jLabelteoriaquest = new javax.swing.JLabel();
jLabelautorquest = new javax.swing.JLabel();
jLabelidiomateoriaquest = new javax.swing.JLabel();
jLabelanoquest = new javax.swing.JLabel();
jLabelmetodoquest = new javax.swing.JLabel();
jLabeltitquestionarioquest = new javax.swing.JLabel();
jLabelidiomametodoquest = new javax.swing.JLabel();
jLabeltopicosquest = new javax.swing.JLabel();
jLabelquestoesquest = new javax.swing.JLabel();
jLabeltopicoatualquest = new javax.swing.JLabel();
jLabelquestaoatualquest = new javax.swing.JLabel();
jLabelpesoquest = new javax.swing.JLabel();
jButtonavancardialogoquest = new javax.swing.JButton();

```

```

jButtonsairdialogoquest = new javax.swing.JButton();
jButtonvoltardialogoquest = new javax.swing.JButton();
jButtonconcluirdialogoquest = new javax.swing.JButton();
jLabeltitpesotopicoquest = new javax.swing.JLabel();
jCheckBoxsalvaraosairquest = new javax.swing.JCheckBox();
jScrollPane2 = new javax.swing.JScrollPane();
jLabelmensagem1quest = new javax.swing.JLabel();
jTextFieldpesoquest = new javax.swing.JTextField();
jScrollPane6 = new javax.swing.JScrollPane();
jTextAreaquestaoquest = new javax.swing.JTextArea();
jScrollPane7 = new javax.swing.JScrollPane();
jTextAreatopicoquest = new javax.swing.JTextArea();
jLabel3 = new javax.swing.JLabel();
jLabelcriterioquest = new javax.swing.JLabel();
jLabeltittopicoquest2 = new javax.swing.JLabel();
jDialogdefpesotopic = new javax.swing.JDialog();
jLabelteoriatopic = new javax.swing.JLabel();
jLabelautortopic = new javax.swing.JLabel();
jLabelidiomateoriatopic = new javax.swing.JLabel();
jLabelanotopic = new javax.swing.JLabel();
jLabelmetodotopic = new javax.swing.JLabel();
jLabeltitquestionariotopic = new javax.swing.JLabel();
jLabelidiomametodotopic = new javax.swing.JLabel();
jLabeltopicoستopic = new javax.swing.JLabel();
jLabelquestoestopic = new javax.swing.JLabel();
jLabeltopicoatualtopic = new javax.swing.JLabel();
jLabelquestaoatualtopic = new javax.swing.JLabel();
jLabelpesotopic = new javax.swing.JLabel();
jButtonavancardialogotopic = new javax.swing.JButton();
jButtonsairdialogotopic = new javax.swing.JButton();
jButtonvoltardialogotopic = new javax.swing.JButton();
jButtonconcluirdialogotopic = new javax.swing.JButton();
jLabeltittopicotopic = new javax.swing.JLabel();
jCheckBoxsalvaraosairtopic = new javax.swing.JCheckBox();
jScrollPane21 = new javax.swing.JScrollPane();
jLabelmensagem1topic = new javax.swing.JLabel();
jTextFieldpesotopic = new javax.swing.JTextField();
jScrollPane61 = new javax.swing.JScrollPane();
jTextAreaquestaoatopic = new javax.swing.JTextArea();
jScrollPane71 = new javax.swing.JScrollPane();
jTextAreatopicotopic = new javax.swing.JTextArea();
jLabel31 = new javax.swing.JLabel();
jLabelcriteriotopic = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jButtondefinirpesoquestoes = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
jComboBoxteoria = new javax.swing.JComboBox();
jScrollPane4 = new javax.swing.JScrollPane();
jComboBoxmetodo = new javax.swing.JComboBox();
jScrollPane5 = new javax.swing.JScrollPane();
jComboBoxcriterio = new javax.swing.JComboBox();
jButtondefinirpesotopicos = new javax.swing.JButton();
jDialogdefpesoquest.getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jDialogdefpesoquest.setTitle("Pesos das Quest\u00f5es");

```

```

jDialogdefpesoquest.setModal(true);
jDialogdefpesoquest.setName("dialogquestionario");
jDialogdefpesoquest.addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) {
jDialogdefpesoquestComponentShown(evt);}});
jDialogdefpesoquest.addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { jDialogdefpesoquestWindowClosing(evt);}});
jLabelteoriaquest.setText("Teoria:");
jDialogdefpesoquest.getContentPane().add(jLabelteoriaquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 600, -1));
jLabelautorquest.setText("Autor:");
jDialogdefpesoquest.getContentPane().add(jLabelautorquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 460, -1));
jLabelidiomateoriaquest.setText("Idioma Original:");
jDialogdefpesoquest.getContentPane().add(jLabelidiomateoriaquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 470, -1));
jLabelanoquest.setText("Ano:");
jDialogdefpesoquest.getContentPane().add(jLabelanoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 150, -1));
jLabelmetodoquest.setText("M\u00e9todo:");
jDialogdefpesoquest.getContentPane().add(jLabelmetodoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 600, -1));
jLabeltitquestionarioquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltitquestionarioquest.setText("Quest\u00e3o: ");
jDialogdefpesoquest.getContentPane().add(jLabeltitquestionarioquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 290, 200, -1));
jLabelidiomametodoquest.setText("Idioma do M\u00e9todo:");
jDialogdefpesoquest.getContentPane().add(jLabelidiomametodoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, 470, -1));
jLabeltopicosquest.setText("Quantidade de \u00f3picos:");
jDialogdefpesoquest.getContentPane().add(jLabeltopicosquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 200, -1));
jLabelquestoesquest.setText("Quantidade total de quest\u00f5es:");
jDialogdefpesoquest.getContentPane().add(jLabelquestoesquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 80, 230, -1));
jLabeltopicoatualquest.setText("T\u00f3pico Atual/Total: 0/0");
jDialogdefpesoquest.getContentPane().add(jLabeltopicoatualquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, 140, -1));
jLabelquestaoatualquest.setText("Quest\u00e3o Atual do T\u00f3pico/Total no T\u00f3pico: 0/0
Quest\u00e3o Atual Total/Total Geral: 0/0");
jDialogdefpesoquest.getContentPane().add(jLabelquestaoatualquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 140, 500, -1));
jLabelpesoquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelpesoquest.setText("Peso da quest\u00e3o: ");
jDialogdefpesoquest.getContentPane().add(jLabelpesoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 440, -1, -1));
jButtonavancardialogoquest.setMnemonic('A');
jButtonavancardialogoquest.setText("Avan\u00e7ar");
jButtonavancardialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonavancardialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonavancardialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 470, -1, -1));
jButtonsairdialogoquest.setText("Sair");
jButtonsairdialogoquest.setDefaultCapable(false);
jButtonsairdialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonsairdialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonsairdialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 470, -1, -1));
jButtonvoltardialogoquest.setMnemonic('V');

```

```

jButtonvoltardialogoquest.setText("Voltar");
jButtonvoltardialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonvoltardialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonvoltardialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 470, -1, -1));
jButtonconcluirdialogoquest.setText("Concluir");
jButtonconcluirdialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonconcluirdialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonconcluirdialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 470, -1, -1));
jLabelitpesotopicoquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelitpesotopicoquest.setForeground(new java.awt.Color(0, 0, 0));
jLabelitpesotopicoquest.setText("Peso do t\u00f3pico:");
jDialogdefpesoquest.getContentPane().add(jLabelitpesotopicoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 200, 520, -1));
jCheckBoxsalvaraosairquest.setSelected(true);
jCheckBoxsalvaraosairquest.setText("Salvar pesos ao sair sem concluir");
jDialogdefpesoquest.getContentPane().add(jCheckBoxsalvaraosairquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 440, -1, -1));
jScrollPane2.setViewportView(jLabelmensagem1quest);
jDialogdefpesoquest.getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 510, 660, 40));
jTextFieldpesoquest.addCaretListener(new javax.swing.event.CaretListener() { public void
caretUpdate(javax.swing.event.CaretEvent evt) { jTextFieldpesoquestCaretUpdate(evt);}});
jDialogdefpesoquest.getContentPane().add(jTextFieldpesoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(230, 440, 80, -1));
jTextAreaquestaoquest.setEditable(false);
jTextAreaquestaoquest.setLineWrap(true);
jScrollPane6.setViewportView(jTextAreaquestaoquest);
jDialogdefpesoquest.getContentPane().add(jScrollPane6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 660, 110));
jTextAreatopicoquest.setEditable(false);
jTextAreatopicoquest.setLineWrap(true);
jScrollPane7.setViewportView(jTextAreatopicoquest);
jDialogdefpesoquest.getContentPane().add(jScrollPane7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, 660, 60));
jLabel3.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel3.setText("Crit\u00e9rio:");
jDialogdefpesoquest.getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, 20));
jLabelcriterioquest.setFont(new java.awt.Font("Dialog", 1, 14));
jDialogdefpesoquest.getContentPane().add(jLabelcriterioquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, 600, 20));
jLabeltittopicoquest2.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltittopicoquest2.setText("T\u00f3pico:");
jDialogdefpesoquest.getContentPane().add(jLabeltittopicoquest2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 200, 130, -1));
jDialogdefpesotopic.getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jDialogdefpesotopic.setTitle("Pesos dos T\u00f3picos");
jDialogdefpesotopic.setModal(true);
jDialogdefpesotopic.setName("dialogquestionario");
jDialogdefpesotopic.addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) {
jDialogdefpesotopicComponentShown(evt);}});
jDialogdefpesotopic.addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { jDialogdefpesotopicWindowClosing(evt);}});
jLabelteoriatopic.setText("Teoria:");
jDialogdefpesotopic.getContentPane().add(jLabelteoriatopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 600, -1));

```



```

jLabelautortopic.setText("Autor:");
jDialogdefpesotopic.getContentPane().add(jLabelautortopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 460, -1));
jLabelidiomateoriaticopic.setText("Idioma Original:");
jDialogdefpesotopic.getContentPane().add(jLabelidiomateoriaticopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 470, -1));
jLabelanotopic.setText("Ano:");
jDialogdefpesotopic.getContentPane().add(jLabelanotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 150, -1));
jLabelmetodotopic.setText("M\u00e9todo:");
jDialogdefpesotopic.getContentPane().add(jLabelmetodotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 600, -1));
jLabeltitquestionariotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltitquestionariotopic.setText("Quest\u00f5es do T\u00fapico: ");
jDialogdefpesotopic.getContentPane().add(jLabeltitquestionariotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 290, 200, -1));
jLabelidiomametodotopic.setText("Idioma do M\u00e9todo:");
jDialogdefpesotopic.getContentPane().add(jLabelidiomametodotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, 470, -1));
jLabeltopicostopic.setText("Quantidade de t\u00fapicos:");
jDialogdefpesotopic.getContentPane().add(jLabeltopicostopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 200, -1));
jLabelquestoestopic.setText("Quantidade total de quest\u00f5es:");
jDialogdefpesotopic.getContentPane().add(jLabelquestoestopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 80, 230, -1));
jLabeltopicoatualtopic.setText("T\u00fapico Atual/Total: 0/0");
jDialogdefpesotopic.getContentPane().add(jLabeltopicoatualtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, 140, -1));
jLabelquestaoatualtopic.setText("Quest\u00e3o Atual do T\u00fapico/Total no T\u00fapico: 0/0
Quest\u00e3o Atual Total/Total Geral: 0/0");
jDialogdefpesotopic.getContentPane().add(jLabelquestaoatualtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 140, 500, -1));
jLabelpesotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelpesotopic.setText("Peso do t\u00fapico:");
jDialogdefpesotopic.getContentPane().add(jLabelpesotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 440, -1, -1));
jButtonavancardialogotopic.setMnemonic('A');
jButtonavancardialogotopic.setText("Avan\u00e7ar");
jButtonavancardialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonavancardialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonavancardialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 470, -1, -1));
jButtonsairdialogotopic.setText("Sair");
jButtonsairdialogotopic.setDefaultCapable(false);
jButtonsairdialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonairsairdialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonsairdialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 470, -1, -1));
jButtonvoltardialogotopic.setMnemonic('V');
jButtonvoltardialogotopic.setText("Voltar");
jButtonvoltardialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonvoltardialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonvoltardialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 470, -1, -1));
jButtonconcluidialogotopic.setText("Concluir");
jButtonconcluidialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonconcluidialogotopicActionPerformed(evt);}});

```

```

jDialogdefpesotopic.getContentPane().add(jButtonconcluidialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 470, -1, -1));
jLabeltittopicotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltittopicotopic.setText("\u00f3pico:");
jDialogdefpesotopic.getContentPane().add(jLabeltittopicotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 200, 130, -1));
jCheckBoxsalvaraosairtopic.setSelected(true);
jCheckBoxsalvaraosairtopic.setText("Salvar pesos ao sair sem concluir");
jDialogdefpesotopic.getContentPane().add(jCheckBoxsalvaraosairtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 440, -1, -1));
jScrollPane21.setViewportViewView(jLabelmensagem1topic);
jDialogdefpesotopic.getContentPane().add(jScrollPane21, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 510, 660, 40));
jTextFieldpesotopic.addCaretListener(new javax.swing.event.CaretListener() { public void
caretUpdate(javax.swing.event.CaretEvent evt) { jTextFieldpesotopicCaretUpdate(evt);}});
jDialogdefpesotopic.getContentPane().add(jTextFieldpesotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 440, 80, -1));
jTextAreaquestaotopic.setEditable(false);
jTextAreaquestaotopic.setLineWrap(true);
jScrollPane61.setViewportViewView(jTextAreaquestaotopic);
jDialogdefpesotopic.getContentPane().add(jScrollPane61, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 660, 110));
jTextAreaticotopic.setEditable(false);
jTextAreaticotopic.setLineWrap(true);
jScrollPane71.setViewportViewView(jTextAreaticotopic);
jDialogdefpesotopic.getContentPane().add(jScrollPane71, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, 660, 60));
jLabel31.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel31.setText("Crit\u00e9rio:");
jDialogdefpesotopic.getContentPane().add(jLabel31, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, 20));
jLabelcriteriotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jDialogdefpesotopic.getContentPane().add(jLabelcriteriotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, 600, 20));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Defini\u00e7\u00e3o de Pesos dos Crit\u00e9rios");
setModal(true);
setName("dialogdefpesoscrit");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Escolha a teoria, o m\u00e9todo e o crit\u00e9rio para definir os pesos do
crit\u00e9rio escolhido.");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, -1, -1));
jButtondefinirpesoquestoes.setText("Definir Pesos das Quest\u00f5es");
jButtondefinirpesoquestoes.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtondefinirpesoquestoesActionPerformed(evt);}});
getContentPane().add(jButtondefinirpesoquestoes, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 330, 200, -1));
jLabel4.setText("Escolha uma teoria de avalia\u00e7\u00e3o");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 50, -1, -1));
jLabel5.setText("Escolha um m\u00e9todo");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, -1, -1));
jButtoncancelar.setText("Cancelar");
jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 330,
90, -1));

```

```

jLabel2.setText("Escolha o crit\u00e9rio de avalia\u00e7\u00e3o para atribuir pesos:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 230,
330, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 390, 660,
40));
jComboBoxteoria.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxteoriaItemStateChanged(evt);}});
jScrollPane3.setViewportView(jComboBoxteoria);
getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, 660,
50));
jComboBoxmetodo.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxmetodoItemStateChanged(evt);}});
jScrollPane4.setViewportView(jComboBoxmetodo);
getContentPane().add(jScrollPane4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, 660,
50));
jComboBoxcrit\u00e9rio.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxcrit\u00e9rioItemStateChanged(evt);}});
jScrollPane5.setViewportView(jComboBoxcrit\u00e9rio);
getContentPane().add(jScrollPane5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, 660,
50));
jButtondefinirpesot\u00f3picos.setText("Definir Pesos dos T\u00f3picos");
jButtondefinirpesot\u00f3picos.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtondefinirpesot\u00f3picosActionPerformed(evt);}});
getContentPane().add(jButtondefinirpesot\u00f3picos, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 330, 200, -1));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, 610));
setLocation((screenSize.width-700)/2,(screenSize.height-610)/2);}
private void jButtonconcluirodialogotopicActionPerformed(java.awt.event.ActionEvent evt) { acao =
sconcluir;
novamensagemtopic("Pesos dos t\u00f3picos do crit\u00e9rio salvos.");
try { atualizabasepelospesostopic();
jButtoncancelar.setText("Sair");
sucesso = true;
houvealgumsucesso = true;}
catch (Exception e) { novamensagemtopic("N\u00e3o foi poss\u00edvel salvar os pesos dos t\u00f3picos do crit\u00e9rio!");
e.printStackTrace();
sucesso = false;}
apagatemplatesdialogotopic();
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jButtonsairedialogotopicActionPerformed(java.awt.event.ActionEvent evt) { if
((jCheckBoxsalvaraosairtopic.isSelected())&(jCheckBoxsalvaraosairtopic.isEnabled())) { try {
atualizabasepelospesostopic();
novamensagemtopic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Pesos parciais salvos.");}
catch (Exception e) { novamensagemtopic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Houve erro salvando os
pesos!");
e.printStackTrace();}}
else { novamensagemtopic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Pesos n\u00e3o foram alterados.");}
apagatemplatesdialogotopic();
acao = scancelar;
sucesso = false;
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jDialogdefpesotopicWindowClosing(java.awt.event.WindowEvent evt) { if
((jCheckBoxsalvaraosairtopic.isSelected())&(jCheckBoxsalvaraosairtopic.isEnabled())) { try {
atualizabasepelospesostopic();
novamensagemtopic("Usu\u00e1rio fechou a janela. Pesos salvos.");}

```

```

catch (Exception e) { novamensagemtopic("Usuário fechou a janela. Houve erro salvando os
pesos!");
e.printStackTrace();}
else { novamensagemtopic("Usuário fechou a janela. Pesos não foram alterados.");}
apagatemplatesdialogotopic();
acao = scancelar;
sucesso = false;
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jDialogdefpesotopicComponentShown(java.awt.event.ComponentEvent evt) {
atualizatemplatesdialogotopic();}
private void jButtonavancardialogotopicActionPerformed(java.awt.event.ActionEvent evt) { acao =
savancar;
atualizatemplatesdialogotopic();
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jButtonvoltardialogotopicActionPerformed(java.awt.event.ActionEvent evt) { acao =
svoltar;
atualizatemplatesdialogotopic();
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jTextFielddpesotopicCaretUpdate(javax.swing.event.CaretEvent evt) { if
(atualizmsgfieldpesoalterado) { atualizarestadofieldpesotopic();}}
private void jButtondefinirpesotopicosActionPerformed(java.awt.event.ActionEvent evt) {
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.Statement stmtpesoquestao;
java.sql.ResultSet rstopico;
java.sql.ResultSet rquestao;
java.sql.ResultSet rsavaliacao;
java.sql.ResultSet rspesoquestao;
try { preenchevariaveistopic();
novamensagemtopic("");
novamensagem("");
if (atualizapesospelabasetopic()) { stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
stmtpesoquestao= con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
acao = savancar;
sucesso = false;
numtopicoatual =0;
numquestaoatualtotal = 0;
deltatopico = 0;
deltaquestaoatot = 0;
deltaquestaoatop = 0;
while (continuartopicotopic(rstopico)) { numtopicoatual = numtopicoatual + deltatopico;
snumtopicoatual = String.valueOf(numtopicoatual);
jLabeltopicoatualtopic.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
rquestao = stmtquestao.executeQuery("select * from questao where cod_topico=" +
rstopico.getString(1) + " order by ordem;");
rquestao.last();
quantquestoestopatual = rquestao.getRow();
squantquestoestopatual = String.valueOf(quantquestoestopatual);
rquestao.beforeFirst();
numquestaoatopatual = 0;
snumquestaoatopatual = String.valueOf(numquestaoatopatual);

```

```

numquestaoatualtotal = 0;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelquestaoatualtopico.setText("");
jTextAreatopicotopic.setText(rstopico.getString(6));
if (quantquestoestopatual > 0) { jTextAreaquestaoatopic.setText("");
while (rsquestao.next()) { rspesoquestaooc = stmtpesoquestaooc.executeQuery("select * from
pesoquestaooc where cod_criterio="+ codcriterio + " and cod_questao =" + rsquestao.getString(1) +
";");
rspesoquestaooc.last();
jTextAreaquestaoatopic.append(rsquestao.getString(5));
if (rspesoquestaooc.getRow() > 0) { jTextAreaquestaoatopic.append("\n\n\n=====>
Peso desta questão: " + rspesoquestaooc.getString(4) + "\n\n *****
\n\n");}
else { jTextAreaquestaoatopic.append("\n\n\n=====> O PESO DESTA QUESTÃO
NÃO FOI DEFINIDO!\n\n ***** \n\n");}
rspesoquestaooc.last();}
else { jTextAreaquestaoatopic.setText("Não há questão cadastrada!");}
jTextAreatopicotopic.setCaretPosition(0);
jTextAreaquestaoatopic.setCaretPosition(0);
jDialogdefpesotopic.show();
jTextAreatopicotopic.setText("");
jTextAreaquestaoatopic.setText("");
rsquestao.close();
novamensagem(mensagem1);
rstopico.close();
stmtquestao.close();
stmttopico.close();
stmtpesoquestaooc.close();}
else { novamensagem("Não foi encontrado tópico ou questão do método/teoria escolhidos. É
necessário que sejam cadastrados tópicos/questões para que se possa definir pesos para o
critério.");}
catch (Exception e) { novamensagemtopic(e.toString());
e.printStackTrace();}
protected boolean continuartopicotopic(java.sql.ResultSet rstopico) throws Exception { boolean
resultado;
if (acao == savancar) { resultado = rstopico.next();
deltatopico = 1;}
else
if ((acao == svoltar) & (numtopicoatual > 1)) { resultado = rstopico.previous();
voltoutopico = true;
deltatopico = -1;}
else { resultado = false;}
return resultado;}
private void jTextFieidpesoquestCaretUpdate(javax.swing.event.CaretEvent evt) { if
(atualizmsgfieldpesoalterado) { atualizarestadofieldpesoquest();}
private void jComboBoxteoriaItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadeteorias > 0) { if (preenchemetodo()) { preenhecriterio();}
codteoria = String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()]);}
else { codteoria = "null";}
teoria = "";}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();}
private void jComboBoxmetodoItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidademetodos > 0) { preenhecriterio();}
codmetodo = String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]);}
else { codmetodo = "null";}
metodo = "";}
catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}

```

```

private void jComboBoxcritérioItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadecriterios > 0) { codcritério =
String.valueOf(codComboBoxcritério[jComboBoxcritério.getSelectedIndex()]);}
else { codcritério = "null";}
critério = "";}
catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}}
private void jButtonvoltardialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
svoltar;
atualizatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonavancardialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
savancar;
atualizatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jDialogdefpesoquestComponentShown(java.awt.event.ComponentEvent evt) {
atualizatemplatesdialogoquest();}
private void jDialogdefpesoquestWindowClosing(java.awt.event.WindowEvent evt) { if
((jCheckBoxsalvaraosairquest.isSelected())&(jCheckBoxsalvaraosairquest.isEnabled())) { try {
atualizabasepelospesosquest();
novamensagemquest("Usuário fechou a janela. Pesos salvos.");}
catch (Exception e) { novamensagemquest("Usuário fechou a janela. Houve erro salvando os
pesos!");
e.printStackTrace();}}
else { novamensagemquest("Usuário fechou a janela. Pesos não foram alterados.");}
apagatemplatesdialogoquest();
acao = scancelar;
sucesso = false;
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonairdialogoquestActionPerformed(java.awt.event.ActionEvent evt) { if
((jCheckBoxsalvaraosairquest.isSelected())&(jCheckBoxsalvaraosairquest.isEnabled())) { try {
atualizabasepelospesosquest();
novamensagemquest("Usuário abandonou a operação. Pesos parciais salvos.");}
catch (Exception e) { novamensagemquest("Usuário abandonou a operação. Houve erro salvando os
pesos!");
e.printStackTrace();}}
else { novamensagemquest("Usuário abandonou a operação. Pesos não foram alterados.");}
apagatemplatesdialogoquest();
acao = scancelar;
sucesso = false;
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonconcluidialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
sconcluir;
novamensagemquest("Pesos das questões do critério salvos.");
try { atualizabasepelospesosquest();
jButtoncancelar.setText("Sair");
sucesso = true;
houvealgumsucesso = true;}
catch (Exception e) { novamensagemquest("Não foi possível salvar os pesos das questões do
critério!");
e.printStackTrace();
sucesso = false;}
apagatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void formComponentShown(java.awt.event.ComponentEvent evt) { novamensagem("");}

```

```

apagatemplates();
apagatemplatesdialogoquest();
apagatemplatesdialogotopic();
sucesso = false;
houvealgumsucesso = false;
atualizmsgfieldpesoalterado = true;
try { if (preencheteoria()) { if (preenchemetodo()) { preenchecrierio();}}
catch (Exception e) { novamensagem(e.toString());
novamensagem("Não foi possível acessar o banco de dados.");
e.printStackTrace();}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
if (!houvealgumsucesso) { sucesso = false;};
apagadados();
apagadadosdialogoquest();
apagatemplates();
apagatemplatesdialogoquest();
setVisible(false);
dispose();}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Operação 'Definição de pesos' concluída.");}
else { sucesso = false;
novamensagem("Operação 'Definição de pesos' cancelada pelo usuário.");}
apagadados();
apagadadosdialogoquest();
apagatemplates();
apagatemplatesdialogoquest();
setVisible(false);
dispose();}
private void jButtondefinirpesoquestoesActionPerformed(java.awt.event.ActionEvent evt) {
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rsavaliacao;
java.sql.ResultSet rspesotopicoc;
try { preenchevariaveisquest();
novamensagemquest("");
novamensagem("");
if (atualizapesospelabasequest()) { stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
stmtpesotopicoc= con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
acao = savancar;
sucesso = false;
numtopicoatual =0;
numquestaoatualtotal = 0;
deltatopico = 0;
deltaquestaoatotal = 0;
deltaquestaoatop = 0;
while (continuartopicoquest(rstopico)) { numtopicoatual = numtopicoatual + deltatopico;
snumtopicoatual = String.valueOf(numtopicoatual);
jLabeltopicoatualquest.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
rsquestao = stmtquestao.executeQuery("select * from questao where cod_topico=" +
rstopico.getString(1) + " order by ordem;");

```

```

rsquestao.last();
quantquestoestopatual = rsquestao.getRow();
squantquestoestopatual = String.valueOf(quantquestoestopatual);
rsquestao.beforeFirst();
numquestaoatual = 0;
while (continuarquestaoquest(rsquestao)) { numquestaoatual = numquestaoatual +
deltaquestaoatual;
snumquestaoatual = String.valueOf(numquestaoatual);
numquestaoatualtotal = numquestaoatualtotal + deltaquestaoatual;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatual
+ "/" + squantquestoestopatual + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestopatual);
rspesotopicoc = stmtpesotopicoc.executeQuery("select * from pesotopicoc where cod_criterio=" +
codcriterio + " and cod_topico = " + rstopico.getString(1) + ";");
rspesotopicoc.last();
if (rspesotopicoc.getRow() > 0) { jLabelitpesotopicoquest.setForeground(new java.awt.Color(0, 0, 0));
jLabelitpesotopicoquest.setText("Peso do tópico: " + rspesotopicoc.getString(4));} else
{ jLabelitpesotopicoquest.setForeground(new java.awt.Color(200, 20, 20));
jLabelitpesotopicoquest.setText("Não há peso cadastrado para esse tópico!");}
rspesotopicoc.close();
jTextAreatopicoquest.setText(rstopico.getString(6));
jTextAreaquestaoquest.setText(rsquestao.getString(5));
jTextAreatopicoquest.setCaretPosition(1);
jTextAreaquestaoquest.setCaretPosition(1);
jDialogdefpesoquest.show();
jTextAreatopicoquest.setText("");
jTextAreaquestaoquest.setText("");}
rsquestao.close();}
novamensagem(mensagem1);
rstopico.close();
stmtquestao.close();
stmttopico.close();
stmtpesotopicoc.close();}
else { novamensagem("Não foi encontrada questão do método/teoria escolhidos. É necessário que
sejam cadastradas questões para que se possa definir pesos para o critério.");}
catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}
protected boolean continuarquestaoquest(java.sql.ResultSet rstopico) throws Exception { boolean
resultado;
voltoutopico = false;
if (acao == savancar) { resultado = rstopico.next();
deltatopico = 1;}
else
if ((acao == svoltar) & (numtopicoatual > 1)) { resultado = rstopico.previous();
voltoutopico = true;
deltatopico = -1;}
else { resultado = false;}
return resultado;}
protected boolean continuarquestaoquest(java.sql.ResultSet rsquestao) throws Exception { boolean
resultado;
voltouquestao = false;
if (acao == savancar) { resultado = rsquestao.next();
deltaquestaoatual = 1;
deltaquestaoatualtotal = 1;}
else
if (acao == svoltar) { if (voltoutopico) { resultado = rsquestao.last();
voltouquestao = true;
deltaquestaoatualtotal = rsquestao.getRow() - numquestaoatual;
deltaquestaoatual = -1;}

```



```

else { if (numquestaatopatal > 1) { resultado = rsquestao.previous();
voltouquestao = true;
deltaquestaatop = -1;
deltaquestaatot = -1;}
else { resultado = false;}}}
else { resultado = false;}
voltoutopico = false;
return resultado;}
public static void main(String args[]) { new avaliar(new javax.swing.JFrame(), true).show();}
public void novamensagem(String msg) { mensagem = msg;
jLabelmensagem.setText(msg);}
protected boolean preencheteoria() throws Exception { return super.preencheteoria(jComboBoxteoria,
jComboBoxmetodo,
null, jComboBoxcriterio, null, jButtondefinirpesoquestoes, jButtondefinirpesotopicos);}
protected boolean preenchemetodo() throws Exception { return
super.preenchemetodo(jComboBoxteoria, jComboBoxmetodo,
null, jComboBoxcriterio, null, jButtondefinirpesoquestoes, jButtondefinirpesotopicos);}
protected boolean preenhecriterio() throws Exception { java.sql.Statement stmtcriterio;
java.sql.ResultSet rscriterio;
String lcriterio;
int i;
boolean criteriobook;
novamensagem("");
jComboBoxcriterio.removeAllItems();
stmtcriterio = con.createStatement();
if (atualusuario.getCodtipousuario().trim().equalsIgnoreCase("1")) { rscriterio =
stmtcriterio.executeQuery("select * from criterio where cod_metodo = " + codmetodo + ";");}
else { rscriterio = stmtcriterio.executeQuery("select * from criterio where cod_usuario=" +
atualusuario.getCodusuario() + " and cod_metodo = " + codmetodo + " and geral = 0;");}
rscriterio.last();
quantidadecriterios = rscriterio.getRow();
codComboBoxcriterio = new int[quantidadecriterios];
if (quantidadecriterios < 1) { if (!atualusuario.getCodtipousuario().trim().equalsIgnoreCase("1")) {
rscriterio.close();
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod_metodo = " + codmetodo + ";");
rscriterio.last();
if (rscriterio.getRow() > 0) { jComboBoxcriterio.addItem("Não há critério de avaliação ...");
jComboBoxcriterio.setToolTipText("Não há critério de avaliação ...");
if (rscriterio.getRow() > 1) { novamensagem("Os critérios de avaliação existentes para o método
escolhido foram definidos pelo sistema, são de uso geral e não podem ser alterados por usuários.
Para definir pesos é necessário cadastrar pelo menos um critério de avaliação para uso próprio.");}
else { novamensagem("O critério de avaliação existente para o método escolhido foi definido pelo
sistema, é de uso geral e não pode ser alterado por usuários. Para definir pesos é necessário
cadastrar pelo menos um critério de avaliação para uso próprio.");}}
else { jComboBoxcriterio.addItem("Não há critério de avaliação cadastrado...");
jComboBoxcriterio.setToolTipText("Não há critério de avaliação cadastrado...");
novamensagem("É necessário cadastrar um critério de avaliação.");}}
else { jComboBoxcriterio.addItem("Não há critério de avaliação cadastrado...");
jComboBoxcriterio.setToolTipText("Não há critério de avaliação cadastrado...");
novamensagem("É necessário cadastrar um critério de avaliação.");}
jComboBoxcriterio.setEnabled(false);
jButtondefinirpesoquestoes.setEnabled(false);
jButtondefinirpesotopicos.setEnabled(false);
codcriterio = "null";
criterio = "";
criteriobook = false;}
else { jComboBoxcriterio.setToolTipText("");
jComboBoxcriterio.setEnabled(true);
jButtondefinirpesoquestoes.setEnabled(true);
jButtondefinirpesotopicos.setEnabled(true);

```

```

rscriterio.beforeFirst();
i = 1;
while (i <= quantidadecriterios) { rscriterio.next();
lcriterio = String.valueOf(i) + " - " + rscriterio.getString(6);
jComboBoxcritério.addItem(lcriterio);
codComboBoxcritério[i-1] = rscriterio.getInt(1);
i++;}
codcritério = String.valueOf(codComboBoxcritério[jComboBoxcritério.getSelectedIndex()]);
critério = "";
critérioook = true;}
rscriterio.close();
stmtcritério.close();
return critérioook;}

protected void dimensionaestajanela() { java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, (jScrollPane1.getY() + jScrollPane1.getHeight() + 50)));
setLocation((screenSize.width-700)/2,(screenSize.height-(jScrollPane1.getY() +
jScrollPane1.getHeight() + 50))/2);}
public boolean gethouvealgumsucesso() { return houvealgumsucesso;}
public String getmensagem1() { return mensagem1;}
protected void apagadados() { super.apagadados();
quantidadeteorias = 0;
quantidademetodos = 0;
quantidadecriterios = 0;}
protected void apagatemplates() { jLabelmensagem.setText("");
jLabelmensagem1quest.setText("");
jButtoncancelar.setText("Cancelar");
jComboBoxcritério.removeAllItems();
jComboBoxteoria.removeAllItems();
jComboBoxmetodo.removeAllItems();}
protected boolean faltaresponderpesoquest() { int locali, localj;
boolean resultado;
locali = 0;
resultado = false;
while (!(!resultado) & (locali < quanttopicos)) { localj = 0;
while (!(!resultado) & (localj < numquestoesportopico[locali])) { if
(pesosquest[locali][localj].trim().length()== 0) { resultado = true;};
localj++;}
locali++;}
return resultado;}
public void novamensagemquest(String msg) { mensagem1 = msg;
jLabelmensagem1quest.setText(msg);}
void preenchevariaveisquest() throws Exception { java.sql.Statement stmtteoria;
java.sql.Statement stmtmetodo;
java.sql.Statement stmtidiomat;
java.sql.Statement stmtidiomam;
java.sql.Statement stmtpessoa;
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtcritério;
java.sql.ResultSet rteoria;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidiomat;
java.sql.ResultSet rsidiomam;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rscriterio;
novamensagem("");
stmtteoria = con.createStatement();

```

```

stmtmetodo = con.createStatement();
stmtidiomat= con.createStatement();
stmtidiomam= con.createStatement();
stmtpessoa= con.createStatement();
stmttopico= con.createStatement();
stmtquestao= con.createStatement();
stmtcriterio= con.createStatement();
rsteoria = stmtteoria.executeQuery("select * from teoria where
cod="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsteoria.first();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where cod="+rsteoria.getString(2)+";");
rspessoa.first();
rsidiomat = stmtidiomat.executeQuery("select * from idioma where cod="+rsteoria.getString(3)+";");
rsidiomat.first();
rsmetodo = stmtmetodo.executeQuery("select * from metodo where
cod="+String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()])+" and
cod_teorias="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsmetodo.first();
rsidiomam = stmtidiomam.executeQuery("select * from idioma where
cod="+rsmetodo.getString(3)+";");
rsidiomam.first();
rstopico = stmttopico.executeQuery("select * from topico where
cod_metodo="+rsmetodo.getString(1)+" order by ordem;");
rstopico.last();
quanttopicos = rstopico.getRow();
numquestoesportopico = new int[quanttopicos];
squanttopicos = String.valueOf(quanttopicos);
if (quanttopicos > 0) { numtopicoatual = 1;
rstopico.first();
rquestao = stmtquestao.executeQuery("select * from questao where cod_topico="+
rstopico.getString(1) + " order by ordem;");
rquestao.last();
quantquestoestopatual = rquestao.getRow();
rquestao.close();
quantquestoestot = 0;
nummaxquesttop = 0;
rstopico.beforeFirst();
while (rstopico.next()) { rquestao = stmtquestao.executeQuery("select * from questao where
cod_topico="+rstopico.getString(1)+" order by ordem;");
rquestao.last();
numquestoesportopico[rstopico.getRow()-1] = rquestao.getRow();
if (rquestao.getRow() > nummaxquesttop) { nummaxquesttop = rquestao.getRow();}
quantquestoestot = quantquestoestot + rquestao.getRow();
rquestao.close();}
rstopico.close();}
else { numtopicoatual = 0;
quantquestoestopatual = 0;
quantquestoestot = 0;
nummaxquesttop = 0;}
snumtopicoatual = String.valueOf(numtopicoatual);
squantquestoestopatual = String.valueOf(quantquestoestopatual);
snummaxquesttop = String.valueOf(nummaxquesttop);
squantquestoestot = String.valueOf(quantquestoestot);
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod=" +
String.valueOf(codComboBoxcriterio[jComboBoxcriterio.getSelectedIndex()]) + ";");
rscriterio.first();
codteoria = rsteoria.getString(1);
teoria = rsteoria.getString(4);
codmetodo = rsmetodo.getString(1);
metodo = rsmetodo.getString(4);

```

```

idiomat = rsidiomat.getString(3);
ano_t = rteoria.getString(6);
idiomam = rsidiomam.getString(3);
pessoa = rspessoa.getString(2);
codcritério = rscritério.getString(1);
critério = rscritério.getString(6);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
numquestaotopatual = 1;
snumquestaotopatual = String.valueOf(numquestaotopatual);
jLabelteoriaquest.setText("Teoria: " + teoria);
jLabelautorquest.setText("Autor: " + pessoa);
jLabelidiomateoriaquest.setText("Idioma da teoria: " + idiomat);
jLabelanoquest.setText("Ano da teoria: " + ano_t);
jLabelmetodoquest.setText("Método usado: " + metodo);
jLabelidiomametodoquest.setText("Idioma do método: " + idiomam);
jLabeltopicosquest.setText("Quantidade de tópicos: " + squanttopicos);
jLabelquestoesquest.setText("Quantidade total de questões: " + squantquestoestot);
jLabeltopicoatualquest.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaotopatual
+ "/" + squantquestoestopatual + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestot);
jLabelcritérioquest.setText(critério);
rscritério.close();
rspessoa.close();
rsidiomam.close();
rsidiomat.close();
rsmetodo.close();
rteoria.close();
stmtcritério.close();
stmtquestao.close();
stmtpessoa.close();
stmtidiomam.close();
stmtidiomat.close();
stmtmetodo.close();
stmtteoria.close();}
protected void dimensionadialogoquest() { java.awt.Dimension tamanhotela =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
jDialogdefpesoquest.setSize(new java.awt.Dimension(700, 590));
jDialogdefpesoquest.setLocation((tamanhotela.width-700)/2,(tamanhotela.height-596)/2);}
protected boolean atualizapesospelabasequest() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtpesoquestaac;
java.sql.ResultSet rstopico;
java.sql.ResultSet rquestao;
java.sql.ResultSet rspesoquestaac;
int locali, localj;
int numerotopico, numeroquestao;
boolean hatopico, haquestao;
String respostapesoquestaac;
hatopico = false;
haquestao = false;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtpesoquestaac = con.createStatement();
pesosquest = new String[quanttopicos][nummaxquesttop];
locali = 0;
while (locali < quanttopicos) { localj = 0;
while (localj < nummaxquesttop) { pesosquest[locali][localj] = "";
localj++;}

```

```

locali++;
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + "
order by ordem;");
rstopico.last();
if (rstopico.getRow()> 0) { hatopico = true;
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rsquestao.last();
if (rsquestao.getRow()> 0) { haquestao = true;
rsquestao.beforeFirst();
while (rsquestao.next()) { rspesoquestaoc = stmtpesoquestaoc.executeQuery("select * from
pesoquestaoc where cod_questao=" + rsquestao.getString(1) + " and cod_criterio=" + codcriterio +
";");
rspesoquestaoc.last();
numerotopico = rstopico.getRow();
numeroquestao = rsquestao.getRow();
if (rspesoquestaoc.getRow() > 0 ) { respostapesoquestaoc = rspesoquestaoc.getString(4).trim();
try { Float.parseFloat(respostapesoquestaoc);
pesosquest[numerotopico-1][numeroquestao-1] = respostapesoquestaoc;}
catch (Exception e) { pesosquest[numerotopico-1][numeroquestao-1] = "";}
rspesoquestaoc.close();}
rsquestao.close();}};
rstopico.close();
return ( hatopico & haquestao );}
protected void atualizapesopelofieldpesoquest() { String spesolocal;
spesolocal = jTextFieldpesoquest.getText().trim();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
pesosquest[numtopicoatual-1][numquestaotopatual-1] = spesolocal;}
catch (Exception e) { }}
else { pesosquest[numtopicoatual-1][numquestaotopatual-1] = "";}
protected void atualizafieldpesopelopesoquest() {
jTextFieldpesoquest.setText(pesosquest[numtopicoatual-1][numquestaotopatual-1]);}
protected void atualizabasepelospesosquest() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtpesoquestaoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rspesoquestaoc;
int rspesoquestaocint;
boolean rspesoquestaocbool;
int locali, localj;
int numerotopico, numeroquestao;
String respostausuario;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtpesoquestaoc = con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rsquestao.beforeFirst();
while (rsquestao.next()) { rspesoquestaoc = stmtpesoquestaoc.executeQuery("select * from
pesoquestaoc where cod_questao=" + rsquestao.getString(1) + " and cod_criterio=" + codcriterio +
";");
rspesoquestaoc.last();
numerotopico = rstopico.getRow();
numeroquestao = rsquestao.getRow();
respostausuario = pesosquest[numerotopico-1][numeroquestao-1].trim();

```

```

if (rspesoquestaac.getRow() > 0 ) { if (respostausuario.length() > 0) { rspesoquestaacint =
stmtpesoquestaac.executeUpdate("update pesoquestaac set peso = " + respostausuario + "
where cod_questao = " + rsquestao.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { rspesoquestaacbool = stmtpesoquestaac.execute("delete from pesoquestaac where
cod_questao = " + rsquestao.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { if (respostausuario.length() > 0) { rspesoquestaacbool = stmtpesoquestaac.execute("INSERT
INTO pesoquestaac (cod, cod_questao, cod_criterio, peso) VALUES (NULL, " +
rsquestao.getString(1) + ", " + codcriterio + ", " + respostausuario + ");");}
rspesoquestaac.close();}
rsquestao.close();}
rstopico.close();}
protected void atualizatemplatesdialogoquest() { boolean aindafalta;
novamensagemquest("");
aindafalta = faltaresponderpesoquest();
if (pesosquest[numtopicoatual-1][numquestaoatual-1].trim().length() == 0) {
apagapesoquest(false);}
else { atualizafieldpesopelopesoquest();}
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não definidos.");}
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);
if (aindafalta) { novamensagemquest("Há pesos não definidos.");}
else { jButtonavancardialogoquest.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(true);}
else { jButtonconcluidialogoquest.setEnabled(true);
jCheckBoxsalvaraosairquest.setEnabled(false);
novamensagemquest("Todos os pesos definidos.");}
protected void atualizarestadofieldpesoquest() { boolean aindafalta;
String spesolocal;
novamensagemquest("");
spesolocal = jTextFieldpesoquest.getText().trim();
/* JTextAreaquestaoquest.setText("ANTES\n\nTopico Atual:" + String.valueOf(numtopicoatual-1) +
"\nQuestao atual:" + String.valueOf(numquestaoatual-1)+ "\n\n");
int locali = 0;
int localj = 0;
while (locali < quanttopicos) { localj = 0;
while (localj < nummaxquesttop) { JTextAreaquestaoquest.append("I: " + String.valueOf(locali) + " - J: "
+ String.valueOf(localj) + " - resp: " + String.valueOf(pesosquest[locali][localj]) + "\n");
localj++;}
locali++;}
*/
atualizapesopelofieldpesoquest();
aindafalta = faltaresponderpesoquest();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não definidos.");}
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);
if (aindafalta) { novamensagemquest("Há pesos não definidos.");}
else { jButtonavancardialogoquest.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(true);}
else { jButtonconcluidialogoquest.setEnabled(true);
jCheckBoxsalvaraosairquest.setEnabled(false);
novamensagemquest("Todos os pesos definidos.");}
catch (Exception e) { novamensagemquest("Erro: há valores que não são numéricos na definição do
peso. Corrigir para prosseguir.");
jButtonvoltardialogoquest.setEnabled(false);
jButtonavancardialogoquest.setEnabled(false);

```

```

jButtonconcluirdialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(false);}
else { if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não
definidos.");};
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);
if (aindafalta) { novamensagemquest("Há pesos não definidos.");};}
else { jButtonavancardialogoquest.setEnabled(true);}
jButtonconcluirdialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(true);}
protected void apagadadosdialogoquest() { }
protected void apagatemplatesdialogoquest() { jLabelmensagem1quest.setText("");
jLabelteoriaquest.setText("Teoria: ");
jLabelautorquest.setText("Autor: ");
jLabelidiomateoriaquest.setText("Idioma da teoria: ");
jLabelanoquest.setText("Ano da teoria: ");
jLabelmetodoquest.setText("Método usado: ");
jLabelidiomametodoquest.setText("Idioma do método: ");
jLabeltopicosquest.setText("Quantidade de tópicos: ");
jLabelquestoesquest.setText("Quantidade total de questões: ");
jLabeltopicoatualquest.setText("Tópico Atual/Total: 0/0");
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: 0/0 Questão Atual
Total/Total Geral: 0/0");
jTextAreatopicoquest.setText("");
jTextAreaquestaoquest.setText("");
jLabelcriterioquest.setText("");
apagapesoquest(false);}
protected void apagapesoquest(boolean atualizarmensagem) { boolean opcaoaterarmsganterior;
opcaoaterarmsganterior = atualizmsgfieldpesoalterado;
atualizmsgfieldpesoalterado = atualizarmensagem;
jTextFieldpesoquest.setText("");
atualizmsgfieldpesoalterado = opcaoaterarmsganterior;}
protected boolean faltaresponderpesotopic() { int locali;
boolean resultado;
locali = 0;
resultado = false;
while ((!resultado) & (locali < quanttopicos)) { if (pesostopic[locali].trim().length()== 0) { resultado =
true;}
locali++;}
return resultado;}
public void novamensagemtopic(String msg) { mensagem1 = msg;
jLabelmensagem1topic.setText(msg);}
void preenchevariaveistopic() throws Exception { java.sql.Statement stmtteoria;
java.sql.Statement stmtmetodo;
java.sql.Statement stmtidiomat;
java.sql.Statement stmtidiomam;
java.sql.Statement stmtpessoa;
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtcriterio;
java.sql.ResultSet rsteoria;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidiomat;
java.sql.ResultSet rsidiomam;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rscriterio;
novamensagem("");

```

```

stmtteoria = con.createStatement();
stmtmetodo = con.createStatement();
stmtidiomat= con.createStatement();
stmtidiomam= con.createStatement();
stmtpessoa= con.createStatement();
stmttopico= con.createStatement();
stmtquestao= con.createStatement();
stmtcriterio= con.createStatement();
rsteoria = stmtteoria.executeQuery("select * from teoria where
cod="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsteoria.first();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where cod="+rsteoria.getString(2)+";");
rspessoa.first();
rsidiomat = stmtidiomat.executeQuery("select * from idioma where cod="+rsteoria.getString(3)+";");
rsidiomat.first();
rsmetodo = stmtmetodo.executeQuery("select * from metodo where
cod="+String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()])+" and
cod_teoriam="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsmetodo.first();
rsidiomam = stmtidiomam.executeQuery("select * from idioma where
cod="+rsmetodo.getString(3)+";");
rsidiomam.first();
rstopico = stmttopico.executeQuery("select * from topico where
cod_metodo="+rsmetodo.getString(1)+" order by ordem;");
rstopico.last();
quanttopicos = rstopico.getRow();
numquestoesportopico = new int[quanttopicos];
squanttopicos = String.valueOf(quanttopicos);
if (quanttopicos > 0) { numtopicoatual = 1;
rstopico.first();
rquestao = stmtquestao.executeQuery("select * from questao where cod_topico="+
rstopico.getString(1) + " order by ordem;");
rquestao.last();
quantquestoestopatual = rquestao.getRow();
rquestao.close();
quantquestoestot = 0;
nummaxquesttop = 0;
rstopico.beforeFirst();
while (rstopico.next()) { rquestao = stmtquestao.executeQuery("select * from questao where
cod_topico="+rstopico.getString(1)+" order by ordem;");
rquestao.last();
numquestoesportopico[rstopico.getRow()-1] = rquestao.getRow();
if (rquestao.getRow() > nummaxquesttop) { nummaxquesttop = rquestao.getRow();}
quantquestoestot = quantquestoestot + rquestao.getRow();
rquestao.close();}
rstopico.close();}
else { numtopicoatual = 0;
quantquestoestopatual = 0;
quantquestoestot = 0;
nummaxquesttop = 0;}
snumtopicoatual = String.valueOf(numtopicoatual);
squantquestoestopatual = String.valueOf(quantquestoestopatual);
snummaxquesttop = String.valueOf(nummaxquesttop);
squantquestoestot = String.valueOf(quantquestoestot);
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod=" +
String.valueOf(codComboBoxcriterio[jComboBoxcriterio.getSelectedIndex()]) + ";");
rscriterio.first();
codteoria = rsteoria.getString(1);
teoria = rsteoria.getString(4);
codmetodo = rsmetodo.getString(1);

```



```

metodo = rsmetodo.getString(4);
idiomat = rsidiomat.getString(3);
ano_t = rsteoria.getString(6);
idiomam = rsidiomam.getString(3);
pessoa = rspessoa.getString(2);
codcritério = rscritério.getString(1);
critério = rscritério.getString(6);
numquestaoatualtotal = 1;
numquestaoatualtotal = String.valueOf(numquestaoatualtotal);
numquestaoatual = 1;
snumquestaoatual = String.valueOf(numquestaoatual);
jLabelteoriaticopic.setText("Teoria: " + teoria);
jLabelautortopic.setText("Autor: " + pessoa);
jLabelidiomateoriaticopic.setText("Idioma da teoria: " + idioma);
jLabelanotopic.setText("Ano da teoria: " + ano_t);
jLabelmetodopic.setText("Método usado: " + metodo);
jLabelidiomametodopic.setText("Idioma do método: " + idiomam);
jLabeltopicostopic.setText("Quantidade de tópicos: " + squanttopicos);
jLabelquestoestopic.setText("Quantidade total de questões: " + squantquestoestot);
jLabeltopicoatualtopic.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
jLabelquestaoatualtopic.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatual + "/" + squantquestoestot + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestot);
jLabelcritériopic.setText(critério);
rscritério.close();
rspessoa.close();
rsidiomam.close();
rsidiomat.close();
rsmetodo.close();
rsteoria.close();
stmtcritério.close();
stmtquestao.close();
stmtpessoa.close();
stmtidiomam.close();
stmtidiomat.close();
stmtmetodo.close();
stmtteoria.close();}
protected void dimensionadialogopic() { java.awt.Dimension tamanhotela =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
jDialogdefpesotopic.setSize(new java.awt.Dimension(700, 590));
jDialogdefpesotopic.setLocation((tamanhotela.width-700)/2,(tamanhotela.height-596)/2);}
protected boolean atualizapesospelabasetopic() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rspesotopicoc;
int locali;
int numerotopico, numeroquestao;
String respostapesotopicoc;
boolean hatopico;
hatopico = false;
stmttopico = con.createStatement();
stmtpesotopicoc = con.createStatement();
pesostopic = new String[quanttopicos];
locali = 0;
while (locali < quanttopicos) { pesostopic[locali] = "";
locali++;}
rstopico = stmttopico.executeQuery("select cod from topico where cod_metodo="+ codmetodo + "
order by ordem;");
rstopico.last();
if (rstopico.getRow()> 0) { hatopico = true;

```

```

rstopico.beforeFirst();
while (rstopico.next()) { rsesotopicoc = stmtpesotopicoc.executeQuery("select * from
pesotopicoc where cod_topico=" + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");
rsesotopicoc.last();
numerotopico = rstopico.getRow();
if (rsesotopicoc.getRow() > 0 ) { respostapesotopicoc = rsesotopicoc.getString(4).trim();
try { Float.parseFloat(respostapesotopicoc);
pesostopic[numerotopico-1] = respostapesotopicoc;}
catch (Exception e) { pesostopic[numerotopico-1] = "";}
rsesotopicoc.close();};
rstopico.close();
return hatopico;}

protected void atualizapesopelofieldpesotopic() { String spesolocal;
spesolocal = jTextFieldpesotopic.getText().trim();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
pesostopic[numtopicoatual-1] = spesolocal;}
catch (Exception e) { }}
else { pesostopic[numtopicoatual-1] = "";}

protected void atualizafieldpesopelopesotopic() {
jTextFieldpesotopic.setText(pesostopic[numtopicoatual-1]);}

protected void atualizabasepelopesostopic() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsesotopicoc;
int rsesotopicocint;
boolean rsesotopicocbool;
int locali;
int numerotopico;
String respostausuario;
stmttopico = con.createStatement();
stmtpesotopicoc = con.createStatement();
rstopico = stmttopico.executeQuery("select cod from topico where cod_metodo="+ codmetodo +
" order by ordem;");
rstopico.beforeFirst();
while (rstopico.next()) { rsesotopicoc = stmtpesotopicoc.executeQuery("select * from pesotopicoc
where cod_topico=" + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");
rsesotopicoc.last();
numerotopico = rstopico.getRow();
respostausuario = pesostopic[numerotopico-1].trim();
if (rsesotopicoc.getRow() > 0 ) { if (respostausuario.length() > 0) { rsesotopicocint =
stmtpesotopicoc.executeUpdate("update pesotopicoc set peso = " + respostausuario + " where
cod_topico = " + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { rsesotopicocbool = stmtpesotopicoc.execute("delete from pesotopicoc where cod_topico = " +
rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { if (respostausuario.length() > 0) { rsesotopicocbool = stmtpesotopicoc.execute("INSERT INTO
pesotopicoc (cod, cod_criterio, cod_topico, peso, porcentmin) VALUES (NULL, " + codcriterio + ", " +
rstopico.getString(1) + ", " + respostausuario + ", 0);");}
rsesotopicoc.close();}
rstopico.close();}

protected void atualizatemplatesdialogotopic() { boolean aindafalta;
novamensagemtopic("");
aindafalta = faltaresponderpesotopic();
if (pesostopic[numtopicoatual-1].trim().length() == 0) { apagapesotopic(false);}
else { atualizafieldpesopelopesotopic();}
if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);}
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);}
if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};}
else { jButtonavancardialogotopic.setEnabled(true);}

```

```

if (aindafalta) { jButtonconcluirdialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
else { jButtonconcluirdialogotopic.setEnabled(true);
jCheckBoxsalvaraosairtopic.setEnabled(false);
novamensagemtopic("Todos os pesos definidos.");}
protected void atualizarestadofieldpesotopic() { boolean aindafalta;
String spesolocal;
novamensagemtopic("");
spesolocal = jTextFieldpesotopic.getText().trim();
atualizapesopelofieldpesotopic();
aindafalta = faltaresponderpesotopic();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);}
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);
if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};}
else { jButtonavancardialogotopic.setEnabled(true);}
if (aindafalta) { jButtonconcluirdialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
else { jButtonconcluirdialogotopic.setEnabled(true);
jCheckBoxsalvaraosairtopic.setEnabled(false);
novamensagemtopic("Todos os pesos já estão definidos.");}
catch (Exception e) { novamensagemtopic("Erro: há valores que não são numéricos na definição do
peso. Corrigir para prosseguir.");
jButtonvoltardialogotopic.setEnabled(false);
jButtonavancardialogotopic.setEnabled(false);
jButtonconcluirdialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(false);}
else { if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);
if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};}
else { jButtonavancardialogotopic.setEnabled(true);}
jButtonconcluirdialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
protected void apagadadosdialogotopic() { }
protected void apagatemplatesdialogotopic() { jLabelmensagem1topic.setText("");
jLabelteoriatopic.setText("Teoria: ");
jLabelautortopic.setText("Autor: ");
jLabelidiomateoriatopic.setText("Idioma da teoria: ");
jLabelanotopic.setText("Ano da teoria: ");
jLabelmetodotopic.setText("Método usado: ");
jLabelidiomametodotopic.setText("Idioma do método: ");
jLabeltopicostopic.setText("Quantidade de tópicos: ");
jLabelquestoestopic.setText("Quantidade total de questões: ");
jLabeltopicoatualtopic.setText("Tópico Atual/Total: 0/0");
jLabelquestaoatualtopic.setText("Questão Atual do Tópico/Total no Tópico: 0/0 Questão Atual
Total/Total Geral: 0/0");
jTextAreatopicotopic.setText("");
jTextAreakuestaoatualtopic.setText("");
jLabelcriteriotopic.setText("");
apagapesotopic(false);}
protected void apagapesotopic(boolean atualizarmensagem) { boolean opcaoaterarmsganterior;
opcaoaterarmsganterior = atualizmsgfieldpesoalterado;
atualizmsgfieldpesoalterado = atualizarmensagem;
jTextFieldpesotopic.setText("");
atualizmsgfieldpesoalterado = opcaoaterarmsganterior;}
private javax.swing.JButton jButtonsaialogoquest;

```

```
private javax.swing.JComboBox jComboBoxmetodo;
private javax.swing.JTextArea jTextAreatopicoquest;
private javax.swing.JComboBox jComboBoxcriterio;
private javax.swing.JLabel jLabelmensagem;
private javax.swing.JDialog jDialogdefpesoquest;
private javax.swing.JScrollPane jScrollPane21;
private javax.swing.JLabel jLabelanoquest;
private javax.swing.JLabel jLabelquestoesquest;
private javax.swing.JButton jButtonconcluirdialogoquest;
private javax.swing.JCheckBox jCheckBoxsalvaraosairquest;
private javax.swing.JLabel jLabelanotopic;
private javax.swing.JButton jButtondefinirpesotopicos;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JLabel jLabelmensagem1topic;
private javax.swing.JLabel jLabelteoriaquest;
private javax.swing.JLabel jLabelidiomateoriaquest;
private javax.swing.JLabel jLabeltittopicotopic;
private javax.swing.JLabel jLabelmetodotopic;
private javax.swing.JLabel jLabeltopicosquest;
private javax.swing.JLabel jLabelpesotopic;
private javax.swing.JTextArea jTextAreaquestaoquest;
private javax.swing.JTextField jTextFieldpesotopic;
private javax.swing.JLabel jLabelidiomateoriatic;
private javax.swing.JLabel jLabelcriteriopic;
private javax.swing.JButton jButtonavancardialogopic;
private javax.swing.JLabel jLabeltitquestionarioquest;
private javax.swing.JScrollPane jScrollPane5;
private javax.swing.JButton jButtonsaairdialogopic;
private javax.swing.JLabel jLabelmensagem1quest;
private javax.swing.JLabel jLabelmetodoquest;
private javax.swing.JLabel jLabelquestoestopic;
private javax.swing.JLabel jLabelteoriatic;
private javax.swing.JButton jButtonvoltardialogopic;
private javax.swing.JButton jButtoncancelar;
private javax.swing.JLabel jLabelquestaoatualquest;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane6;
private javax.swing.JLabel jLabeltittopicoquest2;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabeltitpesotopicoquest;
private javax.swing.JButton jButtonconcluirdialogopic;
private javax.swing.JLabel jLabelidiomametodoquest;
private javax.swing.JTextArea jTextAreaquestaoopic;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JDialog jDialogdefpesopic;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane71;
private javax.swing.JLabel jLabeltopicoestopic;
private javax.swing.JButton jButtonavancardialogoquest;
private javax.swing.JLabel jLabel3;
private javax.swing.JCheckBox jCheckBoxsalvaraosairtopic;
private javax.swing.JLabel jLabeltopicoatualtopic;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabelidiomametodopic;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JTextArea jTextAreatopicotopic;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JLabel jLabelautorquest;
private javax.swing.JComboBox jComboBoxteoria;
private javax.swing.JButton jButtonvoltardialogoquest;
```

```

private javax.swing.JTextField jTextFieldpesoquest;
private javax.swing.JLabel jLabeltitquestionariotopic;
private javax.swing.JLabel jLabelcriterioquest;
private javax.swing.JLabel jLabelautortopic;
private javax.swing.JScrollPane jScrollPane61;
private javax.swing.JLabel jLabelpesoquest;
private javax.swing.JLabel jLabelquestaoatualtopic;
private javax.swing.JLabel jLabel31;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JButton jButtondefinirpesoquestoes;
private javax.swing.JLabel jLabeltopicoatualquest;}

package pavplgov;
import javax.swing.*;
import java.text.*;
import java.lang.*;
import java.sql.*;
import java.sql.Connection;
import java.sql.DriverManager;
public class defpesoscrit extends dialogoabstrato { protected String mensagem1;
protected String squanttopicos, squantquestoestopatual, squantquestoestot, snumtopicoatual,
snumquestaoatopatual, snumquestaoatualtotal, snummaxquesttop;
protected int quanttopicos, quantquestoestopatual, quantquestoestot, numtopicoatual,
numquestaoatopatual, numquestaoatualtotal, nummaxquesttop;
protected int numquestoesportopico[];
protected String codcriterio, criterio, codavaliacao, avaliacao;
protected String acao, savancar, svoltar, scancelar, sconcluir;
protected int deltatopico, deltaquestaoatot, deltaquestaoatop;
boolean voltoutopico, voltouquestao;
boolean atualizmsgfieldpesoalterado;
String pesosquest[ ][ ], pesostopic[ ];
public defpesoscrit(java.awt.Frame parent, boolean modal) { super(parent, modal);
initComponents();
labeldamensagem = jLabelmensagem;
dimensionaestajanela();
dimensionadialogoquest();
dimensionadialogotopic();
apagatemplates();
apagatemplatesdialogoquest();
apagatemplatesdialogotopic();
apagadadosdialogoquest();
apagadadosdialogotopic();
svoltar = "voltar";
savancar = "avancar";
sconcluir = "concluir";
scancelar = "cancelar";
acao = "";
atualizmsgfieldpesoalterado = true;}
private void initComponents() { jDialogdefpesoquest = new javax.swing.JDialog();
jLabelteoriaquest = new javax.swing.JLabel();
jLabelautorquest = new javax.swing.JLabel();
jLabelidiomateoriaquest = new javax.swing.JLabel();
jLabelanoquest = new javax.swing.JLabel();
jLabelmetodoquest = new javax.swing.JLabel();
jLabeltitquestionarioquest = new javax.swing.JLabel();
jLabelidiomametodoquest = new javax.swing.JLabel();
jLabeltopicosquest = new javax.swing.JLabel();
jLabelquestoesquest = new javax.swing.JLabel();
jLabeltopicoatualquest = new javax.swing.JLabel();
jLabelquestaoatualquest = new javax.swing.JLabel();

```

```

jLabelpesoquest = new javax.swing.JLabel();
jButtonavancardialogoquest = new javax.swing.JButton();
jButtonsairdialogoquest = new javax.swing.JButton();
jButtonvoltardialogoquest = new javax.swing.JButton();
jButtonconcluidialogoquest = new javax.swing.JButton();
jLabeltitpesotopicoquest = new javax.swing.JLabel();
jCheckBoxsalvaraosairquest = new javax.swing.JCheckBox();
jScrollPane2 = new javax.swing.JScrollPane();
jLabelmensagem1quest = new javax.swing.JLabel();
jTextFieldpesoquest = new javax.swing.JTextField();
jScrollPane6 = new javax.swing.JScrollPane();
jTextAreaquestaquest = new javax.swing.JTextArea();
jScrollPane7 = new javax.swing.JScrollPane();
jTextAreatopicoquest = new javax.swing.JTextArea();
jLabel3 = new javax.swing.JLabel();
jLabelcriterioquest = new javax.swing.JLabel();
jLabeltittopicoquest2 = new javax.swing.JLabel();
jDialogdefpesotopic = new javax.swing.JDialog();
jLabelteoriatopic = new javax.swing.JLabel();
jLabelautortopic = new javax.swing.JLabel();
jLabelidiomateoriatopic = new javax.swing.JLabel();
jLabelanotopic = new javax.swing.JLabel();
jLabelmetodotopic = new javax.swing.JLabel();
jLabeltitquestionariotopic = new javax.swing.JLabel();
jLabelidiomametodotopic = new javax.swing.JLabel();
jLabeltopicoستopic = new javax.swing.JLabel();
jLabelquestoestopic = new javax.swing.JLabel();
jLabeltopicoatualtopic = new javax.swing.JLabel();
jLabelquestaoatualtopic = new javax.swing.JLabel();
jLabelpesotopic = new javax.swing.JLabel();
jButtonavancardialogotopic = new javax.swing.JButton();
jButtonsairdialogotopic = new javax.swing.JButton();
jButtonvoltardialogotopic = new javax.swing.JButton();
jButtonconcluidialogotopic = new javax.swing.JButton();
jLabeltittopicotopic = new javax.swing.JLabel();
jCheckBoxsalvaraosairtopic = new javax.swing.JCheckBox();
jScrollPane21 = new javax.swing.JScrollPane();
jLabelmensagem1topic = new javax.swing.JLabel();
jTextFieldpesotopic = new javax.swing.JTextField();
jScrollPane61 = new javax.swing.JScrollPane();
jTextAreaquestaotopic = new javax.swing.JTextArea();
jScrollPane71 = new javax.swing.JScrollPane();
jTextAreatopicotopic = new javax.swing.JTextArea();
jLabel31 = new javax.swing.JLabel();
jLabelcriteriotopic = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jButtondefinirpesoquestoes = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jButtoncancelar = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jLabelmensagem = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
jComboBoxteoria = new javax.swing.JComboBox();
jScrollPane4 = new javax.swing.JScrollPane();
jComboBoxmetodo = new javax.swing.JComboBox();
jScrollPane5 = new javax.swing.JScrollPane();
jComboBoxcriterio = new javax.swing.JComboBox();
jButtondefinirpesotopicos = new javax.swing.JButton();

```

```

jDialogdefpesoquest.getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
jDialogdefpesoquest.setTitle("Pesos das Quest\u00f5es");
jDialogdefpesoquest.setModal(true);
jDialogdefpesoquest.setName("dialogquestionario");
jDialogdefpesoquest.addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) {
jDialogdefpesoquest.componentShown(evt);}});
jDialogdefpesoquest.addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { jDialogdefpesoquest.windowClosing(evt);}});
jLabelteoriaquest.setText("Teoria:");
jDialogdefpesoquest.getContentPane().add(jLabelteoriaquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 600, -1));
jLabelautorquest.setText("Autor:");
jDialogdefpesoquest.getContentPane().add(jLabelautorquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 460, -1));
jLabelidiomateoriaquest.setText("Idioma Original:");
jDialogdefpesoquest.getContentPane().add(jLabelidiomateoriaquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 470, -1));
jLabelanoquest.setText("Ano:");
jDialogdefpesoquest.getContentPane().add(jLabelanoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 150, -1));
jLabelmetodoquest.setText("M\u00e9todo:");
jDialogdefpesoquest.getContentPane().add(jLabelmetodoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 600, -1));
jLabelitquestionarioquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelitquestionarioquest.setText("Quest\u00e3o: ");
jDialogdefpesoquest.getContentPane().add(jLabelitquestionarioquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 290, 200, -1));
jLabelidiomametodoquest.setText("Idioma do M\u00e9todo:");
jDialogdefpesoquest.getContentPane().add(jLabelidiomametodoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, 470, -1));
jLabeltopicosquest.setText("Quantidade de \u00f3picos:");
jDialogdefpesoquest.getContentPane().add(jLabeltopicosquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 200, -1));
jLabelquestoesquest.setText("Quantidade total de quest\u00f5es:");
jDialogdefpesoquest.getContentPane().add(jLabelquestoesquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 80, 230, -1));
jLabeltopicoatualquest.setText("\u00f3pico Atual/Total: 0/0");
jDialogdefpesoquest.getContentPane().add(jLabeltopicoatualquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, 140, -1));
jLabelquestaoatualquest.setText("Quest\u00e3o Atual do \u00f3pico/Total no \u00f3pico: 0/0
Quest\u00e3o Atual Total/Total Geral: 0/0");
jDialogdefpesoquest.getContentPane().add(jLabelquestaoatualquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 140, 500, -1));
jLabelpesoquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelpesoquest.setText("Peso da quest\u00e3o: ");
jDialogdefpesoquest.getContentPane().add(jLabelpesoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 440, -1, -1));
jButtonavancardialogoquest.setMnemonic('A');
jButtonavancardialogoquest.setText("Avan\u00e7ar");
jButtonavancardialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonavancardialogoquest.actionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonavancardialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 470, -1, -1));
jButtonsairdialogoquest.setText("Sair");
jButtonsairdialogoquest.setDefaultCapable(false);
jButtonsairdialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonairdialogoquest.actionPerformed(evt);}});

```

```

jDialogdefpesoquest.getContentPane().add(jButtonsairdialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 470, -1, -1));
jButtonvoltardialogoquest.setMnemonic('V');
jButtonvoltardialogoquest.setText("Voltar");
jButtonvoltardialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonvoltardialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonvoltardialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 470, -1, -1));
jButtonconcluirdialogoquest.setText("Concluir");
jButtonconcluirdialogoquest.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonconcluirdialogoquestActionPerformed(evt);}});
jDialogdefpesoquest.getContentPane().add(jButtonconcluirdialogoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 470, -1, -1));
jLabelitpesotopicoquest.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelitpesotopicoquest.setForeground(new java.awt.Color(0, 0, 0));
jLabelitpesotopicoquest.setText("Peso do t\u00f3pico:");
jDialogdefpesoquest.getContentPane().add(jLabelitpesotopicoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 200, 520, -1));
jCheckBoxsalvaraosairquest.setSelected(true);
jCheckBoxsalvaraosairquest.setText("Salvar pesos ao sair sem concluir");
jDialogdefpesoquest.getContentPane().add(jCheckBoxsalvaraosairquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 440, -1, -1));
jScrollPane2.setViewportView(jLabelmensagem1quest);
jDialogdefpesoquest.getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 510, 660, 40));
jTextFieldpesoquest.addCaretListener(new javax.swing.event.CaretListener() { public void
caretUpdate(javax.swing.event.CaretEvent evt) { jTextFieldpesoquestCaretUpdate(evt);}});
jDialogdefpesoquest.getContentPane().add(jTextFieldpesoquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(230, 440, 80, -1));
jTextAreaquestaoquest.setEditable(false);
jTextAreaquestaoquest.setLineWrap(true);
jScrollPane6.setViewportView(jTextAreaquestaoquest);
jDialogdefpesoquest.getContentPane().add(jScrollPane6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 660, 110));
jTextAreatopicoquest.setEditable(false);
jTextAreatopicoquest.setLineWrap(true);
jScrollPane7.setViewportView(jTextAreatopicoquest);
jDialogdefpesoquest.getContentPane().add(jScrollPane7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, 660, 60));
jLabel3.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel3.setText("Crit\u00e9rio:");
jDialogdefpesoquest.getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, 20));
jLabelcriterioquest.setFont(new java.awt.Font("Dialog", 1, 14));
jDialogdefpesoquest.getContentPane().add(jLabelcriterioquest, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, 600, 20));
jLabeltittopicoquest2.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltittopicoquest2.setText("T\u00f3pico:");
jDialogdefpesoquest.getContentPane().add(jLabeltittopicoquest2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 200, 130, -1));
jDialogdefpesotopic.getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jDialogdefpesotopic.setTitle("Pesos dos T\u00f3picos");
jDialogdefpesotopic.setModal(true);
jDialogdefpesotopic.setName("dialogquestionario");
jDialogdefpesotopic.addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) {
jDialogdefpesotopicComponentShown(evt);}});
jDialogdefpesotopic.addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { jDialogdefpesotopicWindowClosing(evt);}});

```



```

jLabelteoriatopic.setText("Teoria:");
jDialogdefpesotopic.getContentPane().add(jLabelteoriatopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 600, -1));
jLabelautortopic.setText("Autor:");
jDialogdefpesotopic.getContentPane().add(jLabelautortopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40, 460, -1));
jLabelidiomateoriatopic.setText("Idioma Original:");
jDialogdefpesotopic.getContentPane().add(jLabelidiomateoriatopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 60, 470, -1));
jLabelanotopic.setText("Ano:");
jDialogdefpesotopic.getContentPane().add(jLabelanotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 80, 150, -1));
jLabelmetodotopic.setText("M\u00e9todo:");
jDialogdefpesotopic.getContentPane().add(jLabelmetodotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 600, -1));
jLabelitquestionariotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelitquestionariotopic.setText("Quest\u00f5es do T\u00fapico: ");
jDialogdefpesotopic.getContentPane().add(jLabelitquestionariotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 290, 200, -1));
jLabelidiomametodotopic.setText("Idioma do M\u00e9todo:");
jDialogdefpesotopic.getContentPane().add(jLabelidiomametodotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, 470, -1));
jLabeltopicostopic.setText("Quantidade de t\u00fapicos:");
jDialogdefpesotopic.getContentPane().add(jLabeltopicostopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 80, 200, -1));
jLabelquestoestopic.setText("Quantidade total de quest\u00f5es:");
jDialogdefpesotopic.getContentPane().add(jLabelquestoestopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 80, 230, -1));
jLabeltopicoatualtopic.setText("T\u00fapico Atual/Total: 0/0");
jDialogdefpesotopic.getContentPane().add(jLabeltopicoatualtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, 140, -1));
jLabelquestaoatualtopic.setText("Quest\u00e3o Atual do T\u00fapico/Total no T\u00fapico: 0/0
Quest\u00e3o Atual Total/Total Geral: 0/0");
jDialogdefpesotopic.getContentPane().add(jLabelquestaoatualtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 140, 500, -1));
jLabelpesotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabelpesotopic.setText("Peso do t\u00fapico:");
jDialogdefpesotopic.getContentPane().add(jLabelpesotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 440, -1, -1));
jButtonavancialogotopic.setMnemonic('A');
jButtonavancialogotopic.setText("Avan\u00e7ar");
jButtonavancialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonavancialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonavancialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 470, -1, -1));
jButtonsairdialogotopic.setText("Sair");
jButtonsairdialogotopic.setDefaultCapable(false);
jButtonsairdialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonsairdialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonsairdialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 470, -1, -1));
jButtonvoltardialogotopic.setMnemonic('V');
jButtonvoltardialogotopic.setText("Voltar");
jButtonvoltardialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtonvoltardialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonvoltardialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 470, -1, -1));
jButtonconcluidialogotopic.setText("Concluir");

```

```

jButtonconcluirdialogotopic.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtonconcluirdialogotopicActionPerformed(evt);}});
jDialogdefpesotopic.getContentPane().add(jButtonconcluirdialogotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 470, -1, -1));
jLabeltittopicotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jLabeltittopicotopic.setText("\u00f3pico:");
jDialogdefpesotopic.getContentPane().add(jLabeltittopicotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 200, 130, -1));
jCheckBoxsalvaraosairtopic.setSelected(true);
jCheckBoxsalvaraosairtopic.setText("Salvar pesos ao sair sem concluir");
jDialogdefpesotopic.getContentPane().add(jCheckBoxsalvaraosairtopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(330, 440, -1, -1));
jScrollPane21.setViewportViewView(jLabelmensagem1topic);
jDialogdefpesotopic.getContentPane().add(jScrollPane21, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 510, 660, 40));
jTextFieldpesotopic.addCaretListener(new javax.swing.event.CaretListener() { public void
caretUpdate(javax.swing.event.CaretEvent evt) { jTextFieldpesotopicCaretUpdate(evt);}});
jDialogdefpesotopic.getContentPane().add(jTextFieldpesotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 440, 80, -1));
jTextAreaquestaotopic.setEditable(false);
jTextAreaquestaotopic.setLineWrap(true);
jScrollPane61.setViewportViewView(jTextAreaquestaotopic);
jDialogdefpesotopic.getContentPane().add(jScrollPane61, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 660, 110));
jTextAreatopicotopic.setEditable(false);
jTextAreatopicotopic.setLineWrap(true);
jScrollPane71.setViewportViewView(jTextAreatopicotopic);
jDialogdefpesotopic.getContentPane().add(jScrollPane71, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, 660, 60));
jLabel31.setFont(new java.awt.Font("Dialog", 1, 14));
jLabel31.setText("Crit\u00e9rio:");
jDialogdefpesotopic.getContentPane().add(jLabel31, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 170, -1, 20));
jLabelcriteriotopic.setFont(new java.awt.Font("Dialog", 1, 14));
jDialogdefpesotopic.getContentPane().add(jLabelcriteriotopic, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, 600, 20));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
setTitle("Defini\u00e7\u00e3o de Pesos dos Crit\u00e9rios");
setModal(true);
setName("dialogdefpesoscrit");
addComponentListener(new java.awt.event.ComponentAdapter() { public void
componentShown(java.awt.event.ComponentEvent evt) { formComponentShown(evt);}});
addWindowListener(new java.awt.event.WindowAdapter() { public void
windowClosing(java.awt.event.WindowEvent evt) { formWindowClosing(evt);}});
jLabel1.setText("Escolha a teoria, o m\u00e9todo e o crit\u00e9rio para definir os pesos do
crit\u00e9rio escolhido.");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, -1, -1));
jButtondefinirpesoquestoes.setText("Definir Pesos das Quest\u00f5es");
jButtondefinirpesoquestoes.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
jButtondefinirpesoquestoesActionPerformed(evt);}});
getContentPane().add(jButtondefinirpesoquestoes, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 330, 200, -1));
jLabel4.setText("Escolha uma teoria de avalia\u00e7\u00e3o");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 50, -1, -1));
jLabel5.setText("Escolha um m\u00e9todo:");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 140, -1, -1));
jButtoncancelar.setText("Cancelar");

```

```

jButtoncancelar.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtoncancelarActionPerformed(evt);}});
getContentPane().add(jButtoncancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 330,
90, -1));
jLabel2.setText("Escolha o crit\u00e9rio de avalia\u00e7\u00e3o para atribuir pesos:");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 230, 330, -1));
jScrollPane1.setViewportView(jLabelmensagem);
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 390, 660,
40));
jComboBoxteoria.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxteoriaItemStateChanged(evt);}});
jScrollPane3.setViewportView(jComboBoxteoria);
getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70, 660,
50));
jComboBoxmetodo.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxmetodoItemStateChanged(evt);}});
jScrollPane4.setViewportView(jComboBoxmetodo);
getContentPane().add(jScrollPane4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 160, 660,
50));
jComboBoxcrit\u00e9rio.addItemListener(new java.awt.event.ItemListener() { public void
itemStateChanged(java.awt.event.ItemEvent evt) { jComboBoxcrit\u00e9rioItemStateChanged(evt);}});
jScrollPane5.setViewportView(jComboBoxcrit\u00e9rio);
getContentPane().add(jScrollPane5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 250, 660,
50));
jButtondefinirpesot\u00f3picos.setText("Definir Pesos dos T\u00f3picos");
jButtondefinirpesot\u00f3picos.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) { jButtondefinirpesot\u00f3picosActionPerformed(evt);}});
getContentPane().add(jButtondefinirpesot\u00f3picos, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 330, 200, -1));
pack();
java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, 610));
setLocation((screenSize.width-700)/2,(screenSize.height-610)/2);}
private void jButtonconcluiDialogoT\u00f3picosActionPerformed(java.awt.event.ActionEvent evt) { acao =
sconcluir;
novamensagemt\u00f3pic("Pesos dos t\u00f3picos do crit\u00e9rio salvos.");
try { atualizabasepelospesost\u00f3pic();
jButtoncancelar.setText("Sair");
sucesso = true;
houvealgumsucesso = true;}
catch (Exception e) { novamensagemt\u00f3pic("N\u00e3o foi poss\u00edvel salvar os pesos dos t\u00f3picos do crit\u00e9rio!");
e.printStackTrace();
sucesso = false;}
apagatemplatesdialogot\u00f3pic();
jDialogdefpesot\u00f3pic.setVisible(false);
jDialogdefpesot\u00f3pic.dispose();}
private void jButtonsaireDialogoT\u00f3picActionPerformed(java.awt.event.ActionEvent evt) { if
((jCheckBoxsalvaraosairt\u00f3pic.isSelected())&(jCheckBoxsalvaraosairt\u00f3pic.isEnabled())) { try {
atualizabasepelospesost\u00f3pic();
novamensagemt\u00f3pic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Pesos parciais salvos.");}
catch (Exception e) { novamensagemt\u00f3pic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Houve erro salvando os
pesos!");
e.printStackTrace();}
else { novamensagemt\u00f3pic("Usu\u00e1rio abandonou a opera\u00e7\u00e3o. Pesos n\u00e3o foram alterados.");}
apagatemplatesdialogot\u00f3pic();
acao = scancelar;
sucesso = false;
jDialogdefpesot\u00f3pic.setVisible(false);
jDialogdefpesot\u00f3pic.dispose();}

```

```

private void jDialogdefpesotopicWindowClosing(java.awt.event.WindowEvent evt) { if
((jCheckBoxsalvaraosairtopic.isSelected())&(jCheckBoxsalvaraosairtopic.isEnabled())) { try {
atualizabasepelospesostopic();
novamensagemtopic("Usuário fechou a janela. Pesos salvos.");}
catch (Exception e) { novamensagemtopic("Usuário fechou a janela. Houve erro salvando os pesos!");
e.printStackTrace();}}
else { novamensagemtopic("Usuário fechou a janela. Pesos não foram alterados.");}
apagatemplatesdialogotopic();
acao = scancelar;
sucesso = false;
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jDialogdefpesotopicComponentShown(java.awt.event.ComponentEvent evt) {
atualizatemplatesdialogotopic();}
private void jButtonavancardialogotopicActionPerformed(java.awt.event.ActionEvent evt) { acao =
savancar;
atualizatemplatesdialogotopic();
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jButtonvoltardialogotopicActionPerformed(java.awt.event.ActionEvent evt) { acao =
svoltar;
atualizatemplatesdialogotopic();
jDialogdefpesotopic.setVisible(false);
jDialogdefpesotopic.dispose();}
private void jTextFielddpesotopicCaretUpdate(javax.swing.event.CaretEvent evt) { if
(atualizmsgfieldpesoalterado) { atualizarestadofieldpesotopic();}}
private void jButtondefinirpesotopicosActionPerformed(java.awt.event.ActionEvent evt) {
java.sql.Statement stmttopic;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.Statement stmtpesoquestao;
java.sql.ResultSet rstopic;
java.sql.ResultSet rquestao;
java.sql.ResultSet rsavaliacao;
java.sql.ResultSet rspesoquestao;
try { preenchevariaveistopic();
novamensagemtopic("");
novamensagem("");
if (atualizapesospelabasetopic()) { stmttopic = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
stmtpesoquestao= con.createStatement();
rstopic = stmttopic.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopic.beforeFirst();
acao = savancar;
sucesso = false;
numtopicoatual =0;
numquestaoatualtotal = 0;
deltatopico = 0;
deltaquestaoatot = 0;
deltaquestaoatop = 0;
while (continuartopicotopic(rstopic)) { numtopicoatual = numtopicoatual + deltatopico;
numtopicoatual = String.valueOf(numtopicoatual);
jLabeltopicoatualtopic.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
rquestao = stmtquestao.executeQuery("select * from questao where cod_topico=" +
rstopic.getString(1) + " order by ordem;");
rquestao.last();
quantquestoestopatual = rquestao.getRow();
squantquestoestopatual = String.valueOf(quantquestoestopatual);

```

```

rsquestao.beforeFirst();
numquestaoatual = 0;
snumquestaoatual = String.valueOf(numquestaoatual);
numquestaoatualtotal = 0;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelquestaoatualtopic.setText("");
jTextAreatopicopic.setText(rstopic.getString(6));
if (quantquestoestopatual > 0) { jTextAreaquestaoopic.setText("");
while (rsquestao.next()) { rspesoquestaooc = stmtpesoquestaooc.executeQuery("select * from
pesoquestaooc where cod_criterio="+ codcriterio + " and cod_questao =" + rsquestao.getString(1) +
";");
rspesoquestaooc.last();
jTextAreaquestaoopic.append(rsquestao.getString(5));
if (rspesoquestaooc.getRow() > 0) { jTextAreaquestaoopic.append("\n\n\n=====>
Peso desta questão: " + rspesoquestaooc.getString(4) + "\n\n *****
\n\n");}
else { jTextAreaquestaoopic.append("\n\n\n=====> O PESO DESTA QUESTÃO
NÃO FOI DEFINIDO!\n\n ***** \n\n");}
rspesoquestaooc.last();}
else { jTextAreaquestaoopic.setText("Não há questão cadastrada!");}
jTextAreatopicopic.setCaretPosition(0);
jTextAreaquestaoopic.setCaretPosition(0);
jDialogdefpesotopic.show();
jTextAreatopicopic.setText("");
jTextAreaquestaoopic.setText("");
rsquestao.close();}
novamensagem(mensagem1);
rstopic.close();
stmtquestao.close();
stmttopico.close();
stmtpesoquestaooc.close();}
else { novamensagem("Não foi encontrado tópico ou questão do método/teoria escolhidos. É
necessário que sejam cadastrados tópicos/questões para que se possa definir pesos para o
critério.");}
catch (Exception e) { novamensagemtopic(e.toString());
e.printStackTrace();}
protected boolean continuartopicopic(java.sql.ResultSet rstopic) throws Exception { boolean
resultado;
if (acao == savancar) { resultado = rstopic.next();
deltatopico = 1;}
else
if ((acao == svoltar) & (numtopicoatual > 1)) { resultado = rstopic.previous();
voltoutopico = true;
deltatopico = -1;}
else { resultado = false;}
return resultado;}
private void jTextFieldpesoquestCaretUpdate(javax.swing.event.CaretEvent evt) { if
(atualizmsgfieldpesoalterado) { atualizarestadofieldpesoquest();}
private void jComboBoxteorialtemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadeteorias > 0) { if (preenchemetodo()) { preenhecriterio();}
codteoria = String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()]);}
else { codteoria = "null";}
teoria = "";}
catch (Exception e) { novamensagem(e.toString());
e.printStackTrace();}
private void jComboBoxmetodoItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidademetodos > 0) { preenhecriterio();}
codmetodo = String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()]);}
else { codmetodo = "null";}
metodo = "";}

```

```

catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}
private void jComboBoxcritérioItemStateChanged(java.awt.event.ItemEvent evt) { try { if
(quantidadecriterios > 0) { codcritério =
String.valueOf(codComboBoxcritério[jComboBoxcritério.getSelectedIndex()]);}
else { codcritério = "null";}
critério = "";}
catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}
private void jButtonvoltardialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
svoltar;
atualizatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonavancardialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
savancar;
atualizatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jDialogdefpesoquestComponentShown(java.awt.event.ComponentEvent evt) {
atualizatemplatesdialogoquest();}
private void jDialogdefpesoquestWindowClosing(java.awt.event.WindowEvent evt) { if
((jCheckBoxsalvaraosairquest.isSelected())&(jCheckBoxsalvaraosairquest.isEnabled())) { try {
atualizabasepelospesosquest();
novamensagemquest("Usuário fechou a janela. Pesos salvos.");}
catch (Exception e) { novamensagemquest("Usuário fechou a janela. Houve erro salvando os
pesos!");
e.printStackTrace();}
else { novamensagemquest("Usuário fechou a janela. Pesos não foram alterados.");}
apagatemplatesdialogoquest();
acao = scancelar;
sucesso = false;
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonairdialogoquestActionPerformed(java.awt.event.ActionEvent evt) { if
((jCheckBoxsalvaraosairquest.isSelected())&(jCheckBoxsalvaraosairquest.isEnabled())) { try {
atualizabasepelospesosquest();
novamensagemquest("Usuário abandonou a operação. Pesos parciais salvos.");}
catch (Exception e) { novamensagemquest("Usuário abandonou a operação. Houve erro salvando os
pesos!");
e.printStackTrace();}
else { novamensagemquest("Usuário abandonou a operação. Pesos não foram alterados.");}
apagatemplatesdialogoquest();
acao = scancelar;
sucesso = false;
jDialogdefpesoquest.setVisible(false);
jDialogdefpesoquest.dispose();}
private void jButtonconcluidialogoquestActionPerformed(java.awt.event.ActionEvent evt) { acao =
sconcluir;
novamensagemquest("Pesos das questões do critério salvos.");
try { atualizabasepelospesosquest();
jButtoncancelar.setText("Sair");
sucesso = true;
houvealgumsucesso = true;}
catch (Exception e) { novamensagemquest("Não foi possível salvar os pesos das questões do
critério!");
e.printStackTrace();
sucesso = false;}
apagatemplatesdialogoquest();
jDialogdefpesoquest.setVisible(false);

```

```

jDialogdefpesoquest.dispose();
private void formComponentShown(java.awt.event.ComponentEvent evt) { novamensagem("");
apagatemplates();
apagatemplatesdialogoquest();
apagatemplatesdialogotopic();
sucesso = false;
houvealgumsucesso = false;
atualizmsgfieldpesoalterado = true;
try { if (preencheteoria()) { if (preenchemetodo()) { preenchecrierio();}}
catch (Exception e) { novamensagem(e.toString());
novamensagem("Não foi possível acessar o banco de dados.");
e.printStackTrace();}}
private void formWindowClosing(java.awt.event.WindowEvent evt) { novamensagem("Janela fechada
pelo usuário");
if (!houvealgumsucesso) { sucesso = false;};
apagadados();
apagadadosdialogoquest();
apagatemplates();
apagatemplatesdialogoquest();
setVisible(false);
dispose();}
private void jButtoncancelarActionPerformed(java.awt.event.ActionEvent evt) { if
(houvealgumsucesso) { novamensagem("Operação 'Definição de pesos' concluída.");}
else { sucesso = false;
novamensagem("Operação 'Definição de pesos' cancelada pelo usuário.");}
apagadados();
apagadadosdialogoquest();
apagatemplates();
apagatemplatesdialogoquest();
setVisible(false);
dispose();}
private void jButtondefinirpesoquestoesActionPerformed(java.awt.event.ActionEvent evt) {
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtavaliacao;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rquestao;
java.sql.ResultSet rsavaliacao;
java.sql.ResultSet rspesotopicoc;
try { preenchevariaveisquest();
novamensagemquest("");
novamensagem("");
if (atualizapesospelabasequest()) { stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtavaliacao = con.createStatement();
stmtpesotopicoc= con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
acao = savancar;
sucesso = false;
numtopicoatual =0;
numquestaoatualtotal = 0;
deltatopico = 0;
deltaquestaoatot = 0;
deltaquestaoatop = 0;
while (continuartopicoquest(rstopico)) { numtopicoatual = numtopicoatual + deltatopico;
snumtopicoatual = String.valueOf(numtopicoatual);
jLabeltopicoatualquest.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);

```

```

rsquestao = stmtquestao.executeQuery("select * from questao where cod_topico=" +
rstopico.getString(1) + " order by ordem;");
rsquestao.last();
quantquestoestopatual = rsquestao.getRow();
squantquestoestopatual = String.valueOf(quantquestoestopatual);
rsquestao.beforeFirst();
numquestaoatual = 0;
while (continuarquestaoquest(rsquestao)) { numquestaoatual = numquestaoatual +
deltaquestaoatop;
snumquestaoatual = String.valueOf(numquestaoatual);
numquestaoatualtotal = numquestaoatualtotal + deltaquestaoatop;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatual
+ "/" + squantquestoestopatual + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" +
quantquestoestop);
rspesotopicoc = stmtpesotopicoc.executeQuery("select * from pesotopicoc where cod_criterio=" +
codcriterio + " and cod_topico = " + rstopico.getString(1) + ";");
rspesotopicoc.last();
if (rspesotopicoc.getRow() > 0) { jLabeItitpesotopicoquest.setForeground(new java.awt.Color(0, 0, 0));
jLabeItitpesotopicoquest.setText("Peso do tópico: " + rspesotopicoc.getString(4));} else
{ jLabeItitpesotopicoquest.setForeground(new java.awt.Color(200, 20, 20));
jLabeItitpesotopicoquest.setText("Não há peso cadastrado para esse tópico!");}
rspesotopicoc.close();
jTextAreatopicoquest.setText(rstopico.getString(6));
jTextAreakquestaoquest.setText(rsquestao.getString(5));
jTextAreatopicoquest.setCaretPosition(1);
jTextAreakquestaoquest.setCaretPosition(1);
jDialogdefpesoquest.show();
jTextAreatopicoquest.setText("");
jTextAreakquestaoquest.setText("");
rsquestao.close();}
novamensagem(mensagem1);
rstopico.close();
stmtquestao.close();
stmttopico.close();
stmtpesotopicoc.close();}
else { novamensagem("Não foi encontrada questão do método/teoria escolhidos. É necessário que
sejam cadastradas questões para que se possa definir pesos para o critério.");}
catch (Exception e) { novamensagemquest(e.toString());
e.printStackTrace();}
protected boolean continuartopicoquest(java.sql.ResultSet rstopico) throws Exception { boolean
resultado;
voltoutopico = false;
if (acao == savancar) { resultado = rstopico.next();
deltatopico = 1;}
else
if ((acao == svoltar) & (numtopicoatual > 1)) { resultado = rstopico.previous();
voltoutopico = true;
deltatopico = -1;}
else { resultado = false;}
return resultado;}
protected boolean continuarquestaoquest(java.sql.ResultSet rsquestao) throws Exception { boolean
resultado;
voltouquestao = false;
if (acao == savancar) { resultado = rsquestao.next();
deltaquestaoatop = 1;
deltaquestaoatop = 1;}
else
if (acao == svoltar) { if (voltoutopico) { resultado = rsquestao.last();
voltouquestao = true;

```



```

deltaquestao = rsquestao.getRow() - numquestaoatual;
deltaquestao = -1;
else { if (numquestaoatual > 1) { resultado = rsquestao.previous();
voltouquestao = true;
deltaquestao = -1;
deltaquestao = -1;
else { resultado = false;}}
else { resultado = false;}
voltoutopico = false;
return resultado;}
public static void main(String args[]) { new avaliar(new javax.swing.JFrame(), true).show();}
public void novamensagem(String msg) { mensagem = msg;
jLabelmensagem.setText(msg);}
protected boolean preencheTeoria() throws Exception { return super.preencheTeoria(jComboBoxteoria,
jComboBoxmetodo,
null, jComboBoxcritério, null, jButtondefinirpesoquestoes, jButtondefinirpesotopicos);}
protected boolean preencheMetodo() throws Exception { return
super.preencheMetodo(jComboBoxteoria, jComboBoxmetodo,
null, jComboBoxcritério, null, jButtondefinirpesoquestoes, jButtondefinirpesotopicos);}
protected boolean preencheCritério() throws Exception { java.sql.Statement stmtcritério;
java.sql.ResultSet rsCritério;
String lcritério;
int i;
boolean critérioOK;
novamensagem("");
jComboBoxcritério.removeAllItems();
stmtcritério = con.createStatement();
if (atualusuario.getCodTipousuario().trim().equalsIgnoreCase("1")) { rsCritério =
stmtcritério.executeQuery("select * from critério where cod_metodo = " + codmetodo + ";");
else { rsCritério = stmtcritério.executeQuery("select * from critério where cod_usuario=" +
atualusuario.getCodusuario() + " and cod_metodo = " + codmetodo + " and geral = 0;");}
rsCritério.last();
quantidadecritérios = rsCritério.getRow();
codComboBoxcritério = new int[quantidadecritérios];
if (quantidadecritérios < 1) { if (!atualusuario.getCodTipousuario().trim().equalsIgnoreCase("1")) {
rsCritério.close();
rsCritério = stmtcritério.executeQuery("select * from critério where cod_metodo = " + codmetodo + ";");
rsCritério.last();
if (rsCritério.getRow() > 0) { jComboBoxcritério.addItem("Não há critério de avaliação ...");
jComboBoxcritério.setToolTipText("Não há critério de avaliação ...");
if (rsCritério.getRow() > 1) { novamensagem("Os critérios de avaliação existentes para o método
escolhido foram definidos pelo sistema, são de uso geral e não podem ser alterados por usuários.
Para definir pesos é necessário cadastrar pelo menos um critério de avaliação para uso próprio.");}
else { novamensagem("O critério de avaliação existente para o método escolhido foi definido pelo
sistema, é de uso geral e não pode ser alterado por usuários. Para definir pesos é necessário
cadastrar pelo menos um critério de avaliação para uso próprio.");}}
else { jComboBoxcritério.addItem("Não há critério de avaliação cadastrado...");
jComboBoxcritério.setToolTipText("Não há critério de avaliação cadastrado...");
novamensagem("É necessário cadastrar um critério de avaliação.");}}
else { jComboBoxcritério.addItem("Não há critério de avaliação cadastrado...");
jComboBoxcritério.setToolTipText("Não há critério de avaliação cadastrado...");
novamensagem("É necessário cadastrar um critério de avaliação.");}
jComboBoxcritério.setEnabled(false);
jButtondefinirpesoquestoes.setEnabled(false);
jButtondefinirpesotopicos.setEnabled(false);
codcritério = "null";
critério = "";
critérioOK = false;}
else { jComboBoxcritério.setToolTipText("");
jComboBoxcritério.setEnabled(true);

```

```

jButtondefinirpesoquestoes.setEnabled(true);
jButtondefinirpesotopicos.setEnabled(true);
rscriterio.beforeFirst();
i = 1;
while (i <= quantidadecriterios) { rscriterio.next();
lccriterio = String.valueOf(i) + " - " + rscriterio.getString(6);
jComboBoxcriterio.addItem(lccriterio);
codComboBoxcriterio[i-1] = rscriterio.getInt(1);
i++;}
codcriterio = String.valueOf(codComboBoxcriterio[jComboBoxcriterio.getSelectedIndex()]);
criterio = "";
criteriook = true;}
rscriterio.close();
stmtcriterio.close();
return criteriook;}

protected void dimensionaestajanela() { java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setSize(new java.awt.Dimension(700, (jScrollPane1.getY() + jScrollPane1.getHeight() + 50)));
setLocation((screenSize.width-700)/2,(screenSize.height-(jScrollPane1.getY() +
jScrollPane1.getHeight() + 50))/2);}
public boolean gethouvealgumsucesso() { return houvealgumsucesso;}
public String getmensagem1() { return mensagem1;}
protected void apagadados() { super.apagadados();
quantidadeteorias = 0;
quantidademetodos = 0;
quantidadecriterios = 0;}
protected void apagatemplates() { jLabelmensagem.setText("");
jLabelmensagem1quest.setText("");
jButtoncancelar.setText("Cancelar");
jComboBoxcriterio.removeAllItems();
jComboBoxteoria.removeAllItems();
jComboBoxmetodo.removeAllItems();}
protected boolean faltaresponderpesoquest() { int locali, localj;
boolean resultado;
locali = 0;
resultado = false;
while ((!resultado) & (locali < quanttopicos)) { localj = 0;
while ((!resultado) & (localj < numquestoesportopico[locali])) { if
(pesosquest[locali][localj].trim().length()== 0) { resultado = true;};
localj++;}
locali++;}
return resultado;}
public void novamensagemquest(String msg) { mensagem1 = msg;
jLabelmensagem1quest.setText(msg);}
void preenchevariaveisquest() throws Exception { java.sql.Statement stmtteoria;
java.sql.Statement stmtmetodo;
java.sql.Statement stmtidiomat;
java.sql.Statement stmtidiomam;
java.sql.Statement stmtpessoa;
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtcriterio;
java.sql.ResultSet rsteoria;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidiomat;
java.sql.ResultSet rsidiomam;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rscriterio;

```

```

novamensagem("");
stmtteoria = con.createStatement();
stmtmetodo = con.createStatement();
stmtidiomat= con.createStatement();
stmtidiomam= con.createStatement();
stmtpessoa= con.createStatement();
stmttopico= con.createStatement();
stmtquestao= con.createStatement();
stmtcriterio= con.createStatement();
rsteoria = stmtteoria.executeQuery("select * from teoria where
cod="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsteoria.first();
rspessoa = stmtpessoa.executeQuery("select * from pessoa where cod="+rsteoria.getString(2)+";");
rspessoa.first();
rsidiomat = stmtidiomat.executeQuery("select * from idioma where cod="+rsteoria.getString(3)+";");
rsidiomat.first();
rsmetodo = stmtmetodo.executeQuery("select * from metodo where
cod="+String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()])+" and
cod_teoriam="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsmetodo.first();
rsidiomam = stmtidiomam.executeQuery("select * from idioma where
cod="+rsmetodo.getString(3)+";");
rsidiomam.first();
rstopico = stmttopico.executeQuery("select * from topico where
cod_metodo="+rsmetodo.getString(1)+" order by ordem;");
rstopico.last();
quanttopicos = rstopico.getRow();
numquestoesportopico = new int[quanttopicos];
squanttopicos = String.valueOf(quanttopicos);
if (quanttopicos > 0) { numtopicoatual = 1;
rstopico.first();
rquestao = stmtquestao.executeQuery("select * from questao where cod_topico="+
rstopico.getString(1) + " order by ordem;");
rquestao.last();
quantquestoestopatual = rquestao.getRow();
rquestao.close();
quantquestoestot = 0;
nummaxquesttop = 0;
rstopico.beforeFirst();
while (rstopico.next()) { rquestao = stmtquestao.executeQuery("select * from questao where
cod_topico="+rstopico.getString(1)+" order by ordem;");
rquestao.last();
numquestoesportopico[rstopico.getRow()-1] = rquestao.getRow();
if (rquestao.getRow() > nummaxquesttop) { nummaxquesttop = rquestao.getRow();}
quantquestoestot = quantquestoestot + rquestao.getRow();
rquestao.close();}
rstopico.close();}
else { numtopicoatual = 0;
quantquestoestopatual = 0;
quantquestoestot = 0;
nummaxquesttop = 0;}
snumtopicoatual = String.valueOf(numtopicoatual);
squantquestoestopatual = String.valueOf(quantquestoestopatual);
snummaxquesttop = String.valueOf(nummaxquesttop);
squantquestoestot = String.valueOf(quantquestoestot);
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod=" +
String.valueOf(codComboBoxcriterio[jComboBoxcriterio.getSelectedIndex()]) + ";");
rscriterio.first();
codteoria = rsteoria.getString(1);
teoria = rsteoria.getString(4);

```

```

codmetodo = rsmetodo.getString(1);
metodo = rsmetodo.getString(4);
idiomat = rsidiomat.getString(3);
ano_t = rsteoria.getString(6);
idiomam = rsidiomam.getString(3);
pessoa = rspessoa.getString(2);
codcritério = rscritério.getString(1);
critério = rscritério.getString(6);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelteoriaquest.setText("Teoria: " + teoria);
jLabelautorquest.setText("Autor: " + pessoa);
jLabelidiomateoriaquest.setText("Idioma da teoria: " + idioma);
jLabelanoquest.setText("Ano da teoria: " + ano_t);
jLabelmetodoquest.setText("Método usado: " + metodo);
jLabelidiomametodoquest.setText("Idioma do método: " + idiomam);
jLabeltopicosquest.setText("Quantidade de tópicos: " + squanttopicos);
jLabelquestoesquest.setText("Quantidade total de questões: " + squantquestoestot);
jLabeltopicoatualquest.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatualtotal + "/" + squantquestoestot + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" + squantquestoestot);
jLabelcritérioquest.setText(critério);
rscritério.close();
rspessoa.close();
rsidiomam.close();
rsidiomat.close();
rsmetodo.close();
rsteoria.close();
stmtcritério.close();
stmtquestao.close();
stmtpessoa.close();
stmtidiomam.close();
stmtidiomat.close();
stmtmetodo.close();
stmtteoria.close();}
protected void dimensionadialogoquest() { java.awt.Dimension tamanhotela =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
jDialogdefpesoquest.setSize(new java.awt.Dimension(700, 590));
jDialogdefpesoquest.setLocation((tamanhotela.width-700)/2,(tamanhotela.height-596)/2);}
protected boolean atualizapesospelabasequest() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtpesoquestao;
java.sql.ResultSet rstopico;
java.sql.ResultSet rquestao;
java.sql.ResultSet rspesoquestao;
int locali, localj;
int numerotopico, numeroquestao;
boolean hatopico, haquestao;
String respostapesoquestao;
hatopico = false;
haquestao = false;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtpesoquestao = con.createStatement();
pesosquest = new String[quanttopicos][nummaxquesttop];
locali = 0;
while (locali < quanttopicos) { localj = 0;

```

```

while (localj < nummaxquesttop) { pesosquest[locali][localj] = "";
localj++;}
locali++;}
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.last();
if (rstopico.getRow()> 0) { hatopico = true;
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rsquestao.last();
if (rsquestao.getRow()> 0) { haquestao = true;
rsquestao.beforeFirst();
while (rsquestao.next()) { rspesoquestaoc = stmtpesoquestaoc.executeQuery("select * from
pesoquestaoc where cod_questao=" + rsquestao.getString(1) + " and cod_criterio=" + codcriterio +
";");
rspesoquestaoc.last();
numerotopico = rstopico.getRow();
numeroquestao = rsquestao.getRow();
if (rspesoquestaoc.getRow() > 0 ) { respostapesoquestaoc = rspesoquestaoc.getString(4).trim();
try { Float.parseFloat(respostapesoquestaoc);
pesosquest[numerotopico-1][numeroquestao-1] = respostapesoquestaoc;}
catch (Exception e) { pesosquest[numerotopico-1][numeroquestao-1] = "";}
rspesoquestaoc.close();}}
rsquestao.close();}};
rstopico.close();
return ( hatopico & haquestao );}
protected void atualizapesopelofieldpesoquest() { String spesolocal;
spesolocal = jTextFieldpesoquest.getText().trim();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
pesosquest[numtopicoatual-1][numquestaotopatual-1] = spesolocal;}
catch (Exception e) { }}
else { pesosquest[numtopicoatual-1][numquestaotopatual-1] = "";}
protected void atualizafieldpesopelopesoquest() {
jTextFieldpesoquest.setText(pesosquest[numtopicoatual-1][numquestaotopatual-1]);}
protected void atualizabasepelospesosquest() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtpesoquestaoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;
java.sql.ResultSet rspesoquestaoc;
int rspesoquestaocint;
boolean rspesoquestaocbool;
int locali, localj;
int numerotopico, numeroquestao;
String respostausuario;
stmttopico = con.createStatement();
stmtquestao = con.createStatement();
stmtpesoquestaoc = con.createStatement();
rstopico = stmttopico.executeQuery("select * from topico where cod_metodo="+ codmetodo + " order
by ordem;");
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select cod from questao where
cod_topico=" + rstopico.getString(1) + " order by ordem;");
rsquestao.beforeFirst();
while (rsquestao.next()) { rspesoquestaoc = stmtpesoquestaoc.executeQuery("select * from
pesoquestaoc where cod_questao=" + rsquestao.getString(1) + " and cod_criterio=" + codcriterio +
";");
rspesoquestaoc.last();
numerotopico = rstopico.getRow();

```

```

numeroquestao = rsquestao.getRow();
respostausuario = pesosquest[numerotopico-1][numeroquestao-1].trim();
if (rspesoquestaooc.getRow() > 0) { if (respostausuario.length() > 0) { rspesoquestaoocint =
stmtpesoquestaooc.executeUpdate("update pesoquestaooc set peso = " + respostausuario + " where
cod_questao = " + rsquestao.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { rspesoquestaoocbool = stmtpesoquestaooc.executeUpdate("delete from pesoquestaooc where
cod_questao = " + rsquestao.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { if (respostausuario.length() > 0) { rspesoquestaoocbool = stmtpesoquestaooc.executeUpdate("INSERT
INTO pesoquestaooc (cod, cod_questao, cod_criterio, peso) VALUES (NULL, " +
rsquestao.getString(1) + ", " + codcriterio + ", " + respostausuario + ");");}
rspesoquestaooc.close();}
rsquestao.close();}
rstopico.close();}
protected void atualizatemplatesdialogoquest() { boolean aindafalta;
novamensagemquest("");
aindafalta = faltaresponderpesoquest();
if (pesosquest[numtopicoatual-1][numquestaoatopatual-1].trim().length() == 0) {
apagapesoquest(false);}
else { atualizafieldpesopelopesoquest();}
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não definidos.");};
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);}
if (aindafalta) { novamensagemquest("Há pesos não definidos.");};}
else { jButtonavancardialogoquest.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogoquest.setEnabled(false);}
jCheckBoxsalvaraosairquest.setEnabled(true);}
else { jButtonconcluidialogoquest.setEnabled(true);}
jCheckBoxsalvaraosairquest.setEnabled(false);}
novamensagemquest("Todos os pesos definidos.");}
protected void atualizarestadofieldpesoquest() { boolean aindafalta;
String spesolocal;
novamensagemquest("");
spesolocal = jTextFieldpesoquest.getText().trim();
/* JTextAreaquestaoquest.setText("ANTES\n\nTopico Atual:" + String.valueOf(numtopicoatual-1) +
"\nQuestao atual:" + String.valueOf(numquestaoatopatual-1)+ "\n\n");
int locali = 0;
int localj = 0;
while (locali < quanttopicos) { localj = 0;
while (localj < nummaxqueststtop) { JTextAreaquestaoquest.append("I: " + String.valueOf(locali) + " - J: "
+ String.valueOf(localj) + " - resp: " + String.valueOf(pesosquest[locali][localj]) + "\n");
localj++;}
locali++;}
*/
atualizapesopelofieldpesoquest();
aindafalta = faltaresponderpesoquest();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);}
if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não definidos.");};
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);}
if (aindafalta) { novamensagemquest("Há pesos não definidos.");};}
else { jButtonavancardialogoquest.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogoquest.setEnabled(false);}
jCheckBoxsalvaraosairquest.setEnabled(true);}
else { jButtonconcluidialogoquest.setEnabled(true);}
jCheckBoxsalvaraosairquest.setEnabled(false);}
novamensagemquest("Todos os pesos definidos.");}
catch (Exception e) { novamensagemquest("Erro: há valores que não são numéricos na definição do
peso. Corrigir para prosseguir.");}

```

```

jButtonvoltardialogoquest.setEnabled(false);
jButtonavancardialogoquest.setEnabled(false);
jButtonconcluidialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(false);}
else { if (numquestaoatualtotal == 1) { if (aindafalta) { novamensagemquest("Há pesos não
definidos.");};
jButtonvoltardialogoquest.setEnabled(false);}
else { jButtonvoltardialogoquest.setEnabled(true);}
if (numquestaoatualtotal == quantquestoestot) { jButtonavancardialogoquest.setEnabled(false);
if (aindafalta) { novamensagemquest("Há pesos não definidos.");};}
else { jButtonavancardialogoquest.setEnabled(true);}
jButtonconcluidialogoquest.setEnabled(false);
jCheckBoxsalvaraosairquest.setEnabled(true);}
protected void apagadadosdialogoquest() { }
protected void apagatemplatesdialogoquest() { jLabelmensagem1quest.setText("");
jLabelteoriaquest.setText("Teoria: ");
jLabelautorquest.setText("Autor: ");
jLabelidiomateoriaquest.setText("Idioma da teoria: ");
jLabelanoquest.setText("Ano da teoria: ");
jLabelmetodoquest.setText("Método usado: ");
jLabelidiomametodoquest.setText("Idioma do método: ");
jLabeltopicosquest.setText("Quantidade de tópicos: ");
jLabelquestoesquest.setText("Quantidade total de questões: ");
jLabeltopicoatualquest.setText("Tópico Atual/Total: 0/0");
jLabelquestaoatualquest.setText("Questão Atual do Tópico/Total no Tópico: 0/0 Questão Atual
Total/Total Geral: 0/0");
jTextAreatopicoquest.setText("");
jTextAreaquestaoquest.setText("");
jLabelcriterioquest.setText("");
apagapesoquest(false);}
protected void apagapesoquest(boolean atualizarmensagem) { boolean opcaoaterarmsganterior;
opcaoaterarmsganterior = atualizmsgfieldpesoalterado;
atualizmsgfieldpesoalterado = atualizarmensagem;
jTextFieldpesoquest.setText("");
atualizmsgfieldpesoalterado = opcaoaterarmsganterior;}
protected boolean faltaresponderpesotopic() { int locali;
boolean resultado;
locali = 0;
resultado = false;
while ((!resultado) & (locali < quanttopicos)) { if (pesostopic[locali].trim().length()== 0) { resultado =
true;}
locali++;}
return resultado;}
public void novamensagemtopic(String msg) { mensagem1 = msg;
jLabelmensagem1topic.setText(msg);}
void preenchevariaveistopic() throws Exception { java.sql.Statement stmtteoria;
java.sql.Statement stmtmetodo;
java.sql.Statement stmtidiomat;
java.sql.Statement stmtidiomam;
java.sql.Statement stmpessoa;
java.sql.Statement stmttopico;
java.sql.Statement stmtquestao;
java.sql.Statement stmtcriterio;
java.sql.ResultSet rsteoria;
java.sql.ResultSet rsmetodo;
java.sql.ResultSet rsidiomat;
java.sql.ResultSet rsidiomam;
java.sql.ResultSet rspessoa;
java.sql.ResultSet rstopico;
java.sql.ResultSet rsquestao;

```

```

java.sql.ResultSet rscriterio;
novamensagem("");
stmtteoria = con.createStatement();
stmtmetodo = con.createStatement();
stmtidiomat= con.createStatement();
stmtidiomam= con.createStatement();
stmtpeessoa= con.createStatement();
stmttopico= con.createStatement();
stmtquestao= con.createStatement();
stmtcriterio= con.createStatement();
rsteoria = stmtteoria.executeQuery("select * from teoria where
cod="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsteoria.first();
rspessoa = stmtpeessoa.executeQuery("select * from pessoa where cod="+rsteoria.getString(2)+";");
rspessoa.first();
rsidiomat = stmtidiomat.executeQuery("select * from idioma where cod="+rsteoria.getString(3)+";");
rsidiomat.first();
rsmetodo = stmtmetodo.executeQuery("select * from metodo where
cod="+String.valueOf(codComboBoxmetodo[jComboBoxmetodo.getSelectedIndex()])+" and
cod_teoriam="+String.valueOf(codComboBoxteoria[jComboBoxteoria.getSelectedIndex()])+";");
rsmetodo.first();
rsidiomam = stmtidiomam.executeQuery("select * from idioma where
cod="+rsmetodo.getString(3)+";");
rsidiomam.first();
rstopico = stmttopico.executeQuery("select * from topico where
cod_metodo="+rsmetodo.getString(1)+" order by ordem;");
rstopico.last();
quanttopicos = rstopico.getRow();
numquestoesportopico = new int[quanttopicos];
squanttopicos = String.valueOf(quanttopicos);
if (quanttopicos > 0) { numtopicoatual = 1;
rstopico.first();
rsquestao = stmtquestao.executeQuery("select * from questao where cod_topico="+
rstopico.getString(1) + " order by ordem;");
rsquestao.last();
quantquestoestopatual = rsquestao.getRow();
rsquestao.close();
quantquestoestot = 0;
nummaxquesttop = 0;
rstopico.beforeFirst();
while (rstopico.next()) { rsquestao = stmtquestao.executeQuery("select * from questao where
cod_topico="+rstopico.getString(1)+" order by ordem;");
rsquestao.last();
numquestoesportopico[rstopico.getRow()-1] = rsquestao.getRow();
if (rsquestao.getRow() > nummaxquesttop) { nummaxquesttop = rsquestao.getRow();}
quantquestoestot = quantquestoestot + rsquestao.getRow();
rsquestao.close();}
rstopico.close();}
else { numtopicoatual = 0;
quantquestoestopatual = 0;
quantquestoestot = 0;
nummaxquesttop = 0;}
snumtopicoatual = String.valueOf(numtopicoatual);
squantquestoestopatual = String.valueOf(quantquestoestopatual);
snummaxquesttop = String.valueOf(nummaxquesttop);
squantquestoestot = String.valueOf(quantquestoestot);
rscriterio = stmtcriterio.executeQuery("select * from criterio where cod=" +
String.valueOf(codComboBoxcriterio[jComboBoxcriterio.getSelectedIndex()]) + ";");
rscriterio.first();
codteoria = rsteoria.getString(1);

```



```

teoria = rteoria.getString(4);
codmetodo = rsmetodo.getString(1);
metodo = rsmetodo.getString(4);
idiomat = rsidiomat.getString(3);
ano_t = rteoria.getString(6);
idiomam = rsidiomam.getString(3);
pessoa = rspessoa.getString(2);
codcritério = rscritério.getString(1);
critério = rscritério.getString(6);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
numquestaoatualtotal = 1;
snumquestaoatualtotal = String.valueOf(numquestaoatualtotal);
jLabelteoriaticopic.setText("Teoria: " + teoria);
jLabelautortopic.setText("Autor: " + pessoa);
jLabelidiomateoriaticopic.setText("Idioma da teoria: " + idiomat);
jLabelanotopic.setText("Ano da teoria: " + ano_t);
jLabelmetodopic.setText("Método usado: " + metodo);
jLabelidiomametodopic.setText("Idioma do método: " + idiomam);
jLabeltopicostopic.setText("Quantidade de tópicos: " + squanttopicos);
jLabelquestoestopic.setText("Quantidade total de questões: " + squantquestoestot);
jLabeltopicoatualtopic.setText("Tópico Atual/Total: " + snumtopicoatual + "/" + squanttopicos);
jLabelquestaoatualtopic.setText("Questão Atual do Tópico/Total no Tópico: " + snumquestaoatualtotal + "/" + squantquestoestot + " Questão Atual Total/Total Geral: " + numquestaoatualtotal + "/" + squantquestoestot);
jLabelcritériopic.setText(critério);
rscritério.close();
rspessoa.close();
rsidiomam.close();
rsidiomat.close();
rsmetodo.close();
rteoria.close();
stmtcritério.close();
stmtquestao.close();
stmtpessoa.close();
stmtidiomam.close();
stmtidiomat.close();
stmtmetodo.close();
stmtteoria.close();}
protected void dimensionadialogopic() { java.awt.Dimension tamanhotela =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
jDialogdefpesotopic.setSize(new java.awt.Dimension(700, 590));
jDialogdefpesotopic.setLocation((tamanhotela.width-700)/2,(tamanhotela.height-596)/2);}
protected boolean atualizapesoselabasetopic() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rspesotopicoc;
int locali;
int numerotopico, numeroquestao;
String respostapesotopicoc;
boolean hatopico;
hatopico = false;
stmttopico = con.createStatement();
stmtpesotopicoc = con.createStatement();
pesostopic = new String[quanttopicos];
locali = 0;
while (locali < quanttopicos) { pesostopic[locali] = "";
locali++;}
rstopico = stmttopico.executeQuery("select cod from topico where cod_metodo="+ codmetodo + "
order by ordem;");

```

```

rstopico.last();
if (rstopico.getRow() > 0) { hatopico = true;
rstopico.beforeFirst();
while (rstopico.next()) { rspesotopicoc = stmtpesotopicoc.executeQuery("select * from pesotopicoc
where cod_topico=" + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");
rspesotopicoc.last();
numerotopico = rstopico.getRow();
if (rspesotopicoc.getRow() > 0) { respostapesotopicoc = rspesotopicoc.getString(4).trim();
try { Float.parseFloat(respostapesotopicoc);
pesostopic[numerotopico-1] = respostapesotopicoc;}
catch (Exception e) { pesostopic[numerotopico-1] = "";}
rspesotopicoc.close();}};
rstopico.close();
return hatopico;}
protected void atualizapesopelofieldpesotopic() { String spesolocal;
spesolocal = jTextFieldpesotopic.getText().trim();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
pesostopic[numtopicoatual-1] = spesolocal;}
catch (Exception e) { }}
else { pesostopic[numtopicoatual-1] = "";}
protected void atualizafieldpesopelopesotopic() {
jTextFieldpesotopic.setText(pesostopic[numtopicoatual-1]);}
protected void atualizabasepelospesostopic() throws Exception { java.sql.Statement stmttopico;
java.sql.Statement stmtpesotopicoc;
java.sql.ResultSet rstopico;
java.sql.ResultSet rspesotopicoc;
int rspesotopicocint;
boolean rspesotopicocbool;
int locali;
int numerotopico;
String respostausuario;
stmttopico = con.createStatement();
stmtpesotopicoc = con.createStatement();
rstopico = stmttopico.executeQuery("select cod from topico where cod_metodo="+ codmetodo + "
order by ordem;");
rstopico.beforeFirst();
while (rstopico.next()) { rspesotopicoc = stmtpesotopicoc.executeQuery("select * from pesotopicoc
where cod_topico=" + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");
rspesotopicoc.last();
numerotopico = rstopico.getRow();
respostausuario = pesostopic[numerotopico-1].trim();
if (rspesotopicoc.getRow() > 0) { if (respostausuario.length() > 0) { rspesotopicocint =
stmtpesotopicoc.executeUpdate("update pesotopicoc set peso = " + respostausuario + " where
cod_topico = " + rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { rspesotopicocbool = stmtpesotopicoc.execute("delete from pesotopicoc where cod_topico = " +
rstopico.getString(1) + " and cod_criterio=" + codcriterio + ";");}
else { if (respostausuario.length() > 0) { rspesotopicocbool = stmtpesotopicoc.execute("INSERT INTO
pesotopicoc (cod, cod_criterio, cod_topico, peso, porcentmin) VALUES (NULL, " + codcriterio + ", " +
rstopico.getString(1) + ", " + respostausuario + ", 0);");}
rspesotopicoc.close();}
rstopico.close();}
protected void atualizatemplatesdialogotopic() { boolean aindafalta;
novamensagemtopic("");
aindafalta = faltaresponderpesotopic();
if (pesostopic[numtopicoatual-1].trim().length() == 0) { apagapesotopic(false);}
else { atualizafieldpesopelopesotopic();}
if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);}
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);}

```

```

if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
else { jButtonavancardialogotopic.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
else { jButtonconcluidialogotopic.setEnabled(true);
jCheckBoxsalvaraosairtopic.setEnabled(false);}
novamensagemtopic("Todos os pesos definidos.");}
protected void atualizarestadofieldpesotopic() { boolean aindafalta;
String spesolocal;
novamensagemtopic("");
spesolocal = jTextFieldpesotopic.getText().trim();
atualizapesopelofieldpesotopic();
aindafalta = faltaresponderpesotopic();
if (spesolocal.length() > 0) { try { Float.parseFloat(spesolocal);
if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);}
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);
if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};}
else { jButtonavancardialogotopic.setEnabled(true);}
if (aindafalta) { jButtonconcluidialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
else { jButtonconcluidialogotopic.setEnabled(true);
jCheckBoxsalvaraosairtopic.setEnabled(false);}
novamensagemtopic("Todos os pesos já estão definidos.");}
catch (Exception e) { novamensagemtopic("Erro: há valores que não são numéricos na definição do
peso. Corrigir para prosseguir.");
jButtonvoltardialogotopic.setEnabled(false);
jButtonavancardialogotopic.setEnabled(false);
jButtonconcluidialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(false);}
else { if (numtopicoatual == 1) { if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};
jButtonvoltardialogotopic.setEnabled(false);}
else { jButtonvoltardialogotopic.setEnabled(true);}
if (numtopicoatual == quanttopicos) { jButtonavancardialogotopic.setEnabled(false);
if (aindafalta) { novamensagemtopic("Há pesos não definidos.");};}
else { jButtonavancardialogotopic.setEnabled(true);}
jButtonconcluidialogotopic.setEnabled(false);
jCheckBoxsalvaraosairtopic.setEnabled(true);}
protected void apagadadosdialogotopic() { }
protected void apagatemplatesdialogotopic() { jLabelmensagem1topic.setText("");
jLabelteoriaticopic.setText("Teoria: ");
jLabelautortopic.setText("Autor: ");
jLabelidiomateoriaticopic.setText("Idioma da teoria: ");
jLabelanotopic.setText("Ano da teoria: ");
jLabelmetodotopic.setText("Método usado: ");
jLabelidiomametodotopic.setText("Idioma do método: ");
jLabeltopicostopic.setText("Quantidade de tópicos: ");
jLabelquestoestopic.setText("Quantidade total de questões: ");
jLabeltopicoatualtopic.setText("Tópico Atual/Total: 0/0");
jLabelquestaoatualtopic.setText("Questão Atual do Tópico/Total no Tópico: 0/0 Questão Atual
Total/Total Geral: 0/0");
jTextAreatopicotopic.setText("");
jTextAreaquestaatopic.setText("");
jLabelcriteriotopic.setText("");
apagapesotopic(false);}
protected void apagapesotopic(boolean atualizarmensagem) { boolean opcaoaterarmsganterior;
opcaoaterarmsganterior = atualizmsgfieldpesoalterado;
atualizmsgfieldpesoalterado = atualizarmensagem;
jTextFieldpesotopic.setText("");

```

```
atualizmsgfieldpesoalterado = opcaoaterarmsganterior;}  
private javax.swing.JButton jButtonSairDialogoquest;  
private javax.swing.JComboBox jComboBoxmetodo;  
private javax.swing.JTextArea jTextAreatopicoquest;  
private javax.swing.JComboBox jComboBoxcritério;  
private javax.swing.JLabel jLabelmensagem;  
private javax.swing.JDialog jDialogdefpesoquest;  
private javax.swing.JScrollPane jScrollPane21;  
private javax.swing.JLabel jLabelanoquest;  
private javax.swing.JLabel jLabelquestoesquest;  
private javax.swing.JButton jButtonconcluirDialogoquest;  
private javax.swing.JCheckBox jCheckBoxsalvaraosairquest;  
private javax.swing.JLabel jLabelanotopic;  
private javax.swing.JButton jButtondefinirpesotopicos;  
private javax.swing.JScrollPane jScrollPane7;  
private javax.swing.JLabel jLabelmensagem1topic;  
private javax.swing.JLabel jLabelteoriaquest;  
private javax.swing.JLabel jLabelidiomateoriaquest;  
private javax.swing.JLabel jLabeltittopicotopic;  
private javax.swing.JLabel jLabelmetodotopic;  
private javax.swing.JLabel jLabeltopicosquest;  
private javax.swing.JLabel jLabelpesotopic;  
private javax.swing.JTextArea jTextArequestaoquest;  
private javax.swing.JTextField jTextFieldpesotopic;  
private javax.swing.JLabel jLabelidiomateoriaticopic;  
private javax.swing.JLabel jLabelcriteriotopic;  
private javax.swing.JButton jButtonavancardialogotopic;  
private javax.swing.JLabel jLabeltitquestionarioquest;  
private javax.swing.JScrollPane jScrollPane5;  
private javax.swing.JButton jButtonSairDialogotopic;  
private javax.swing.JLabel jLabelmensagem1quest;  
private javax.swing.JLabel jLabelmetodoquest;  
private javax.swing.JLabel jLabelquestoestopic;  
private javax.swing.JLabel jLabelteoriaticopic;  
private javax.swing.JButton jButtonvoltardialogotopic;  
private javax.swing.JButton jButtoncancelar;  
private javax.swing.JLabel jLabelquestaoatualquest;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JScrollPane jScrollPane6;  
private javax.swing.JLabel jLabeltittopicoquest2;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabeltitpesotopicoquest;  
private javax.swing.JButton jButtonconcluirDialogotopic;  
private javax.swing.JLabel jLabelidiomametodoquest;  
private javax.swing.JTextArea jTextArequestaoatopic;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JDialog jDialogdefpesotopic;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JScrollPane jScrollPane71;  
private javax.swing.JLabel jLabeltopicostopic;  
private javax.swing.JButton jButtonavancardialogoquest;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JCheckBox jCheckBoxsalvaraosairtopic;  
private javax.swing.JLabel jLabeltopicoatualtopic;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabelidiomametodotopic;  
private javax.swing.JScrollPane jScrollPane4;  
private javax.swing.JTextArea jTextAreatopicotopic;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JLabel jLabelautorquest;
```

```
private javax.swing.JComboBox jComboBoxteoria;
private javax.swing.JButton jButtonvoltardialogoquest;
private javax.swing.JTextField jTextFieldpesoquest;
private javax.swing.JLabel jLabeltitquestionariotopic;
private javax.swing.JLabel jLabelcriterioquest;
private javax.swing.JLabel jLabelautortopic;
private javax.swing.JScrollPane jScrollPane61;
private javax.swing.JLabel jLabelpesoquest;
private javax.swing.JLabel jLabelquestaoatualtopic;
private javax.swing.JLabel jLabel31;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JButton jButtondefinirpesoquestoes;
private javax.swing.JLabel jLabeltopicoatualquest;}
```

APÊNDICE C - BANCO DE DADOS

Criação da Base de Dados

```
CREATE DATABASE PLGOV;
```

Criação do usuário SISPLGOV:

```
GRANT ALL PRIVILEGES ON PLGOV.* TO sispgov IDENTIFIED BY 'sis2502';
FLUSH PRIVILEGES;
```

Informando a base de dados que será usada:

```
USE PLGOV;
```

Apagando as tabelas que possam existir:

```
DROP TABLE IF EXISTS avaliacao;
DROP TABLE IF EXISTS plano;
DROP TABLE IF EXISTS pesoquestao;
DROP TABLE IF EXISTS pesotopicoc;
DROP TABLE IF EXISTS criterio;
DROP TABLE IF EXISTS teodoc;
DROP TABLE IF EXISTS questao;
DROP TABLE IF EXISTS topico;
DROP TABLE IF EXISTS metodo;
DROP TABLE IF EXISTS teoria;
DROP TABLE IF EXISTS usuario;
DROP TABLE IF EXISTS tipousuario;
DROP TABLE IF EXISTS orientacao;
DROP TABLE IF EXISTS idioma;
DROP TABLE IF EXISTS pessoa;
DROP TABLE IF EXISTS tipoorientac;
DROP TABLE IF EXISTS titulo;
```

2) Criação das tabelas:

2.1) Criação da tabela titulo:

Esta tabela cadastra os tipos de títulos de graduação, pós-graduação, etc:

```
CREATE TABLE titulo (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  abbrev VARCHAR(5) NOT NULL,
  descr VARCHAR(10) NOT NULL,
  descrgrad VARCHAR(15) NOT NULL,
  PRIMARY KEY (cod))
TYPE=INNODB
COMMENT='títulos de graduação, pós-graduação...';
```

2.2) Criação da tabela tipoorientac:

Esta tabela cadastra os tipos de orientações de um trabalho de conclusão de curso:

```
CREATE TABLE tipoorientac (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  descr VARCHAR(20) NOT NULL,
  PRIMARY KEY (cod))
TYPE=INNODB
```

COMMENT='tipos de orientações de um tcc, orientador, banca...';

2.3) Criação da tabela pessoa:

Esta tabela cadastra as pessoas - usuários e desenvolvedores - que, de alguma forma utiliza o sistema:

```
CREATE TABLE pessoa (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  nome VARCHAR(45) NOT NULL,
  email VARCHAR(30) NULL,
  sexo CHAR NOT NULL,
  PRIMARY KEY (cod))
TYPE=INNODB
COMMENT='pessoas cadastradas no sistema';
```

2.4) Criação da tabela idioma:

Esta tabela cadastra os tipos de idiomas:

```
CREATE TABLE idioma (cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  abrev VARCHAR(10) NOT NULL,
  descr VARCHAR(20) NOT NULL,
  PRIMARY KEY (cod))
TYPE=INNODB
COMMENT='cadastra os tipos de idiomas';
```

2.5) Criação da tabela orientacao:

Esta tabela cadastra os orientadores do trabalho:

```
CREATE TABLE orientacao (
  cod_pessoa INT UNSIGNED NOT NULL,
  cod_titulo INT UNSIGNED NOT NULL,
  cod_tipoor INT UNSIGNED NOT NULL,
  PRIMARY KEY (cod_pessoa, cod_titulo, cod_tipoor),
  INDEX (cod_pessoa),
  FOREIGN KEY(cod_pessoa) REFERENCES pessoa(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT,
  INDEX (cod_titulo),
  FOREIGN KEY(cod_titulo) REFERENCES titulo(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT,
  INDEX (cod_tipoor),
  FOREIGN KEY(cod_tipoor) REFERENCES tipoorientac(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT)
TYPE=INNODB
COMMENT='cadastra os orientadores do trabalho';
```

2.6) Criação da tabela tipousuario:

Esta tabela cadastra os tipos de permissões que os usuários podem ter:

```
CREATE TABLE tipousuario (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  descr VARCHAR(30) NOT NULL,
  poderes VARCHAR(255) NULL,
  PRIMARY KEY (cod))
TYPE=INNODB
COMMENT='tipos de permissões que os usuários podem ter';
```

2.7) Criação da tabela usuario:

Esta tabela cadastra os tipos de permissões que os usuários possuem:

```
CREATE TABLE usuario (
```

```

cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_pessoa INT UNSIGNED NOT NULL,
cod_tipousuario INT UNSIGNED NOT NULL,
id VARCHAR(16) BINARY UNIQUE NOT NULL,
senha VARCHAR(40) BINARY NOT NULL,
PRIMARY KEY (cod, cod_pessoa, cod_tipousuario),
INDEX (cod_pessoa),
FOREIGN KEY(cod_pessoa) REFERENCES pessoa(cod) ON UPDATE CASCADE ON DELETE
RESTRICT,
INDEX (cod_tipousuario),
FOREIGN KEY(cod_tipousuario) REFERENCES tipousuario(cod) ON UPDATE CASCADE ON
DELETE RESTRICT)
TYPE=INNODB
COMMENT='tipos de permissoes que os usuarios podem ter';

```

2.8) Criação da tabela teoria:

Esta tabela cadastra as teorias para avaliação de planos de governo:

```

CREATE TABLE teoria (
cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_pessoa INT UNSIGNED NOT NULL,
cod_idioma INT UNSIGNED NOT NULL,
descr VARCHAR(255) NOT NULL,
topicos BOOL NOT NULL,
ano YEAR NOT NULL,
PRIMARY KEY (cod),
INDEX (cod_pessoa),
FOREIGN KEY(cod_pessoa) REFERENCES pessoa(cod) ON UPDATE CASCADE ON DELETE
RESTRICT,
INDEX (cod_idioma),
FOREIGN KEY(cod_idioma) REFERENCES idioma(cod) ON UPDATE CASCADE ON DELETE
RESTRICT)
TYPE=INNODB
COMMENT='teorias para avaliação de planos de governo';

```

2.9) Criação da tabela metodo:

Esta tabela cadastra os métodos de aplicação de teorias para avaliação de planos de governo:

```

CREATE TABLE metodo (
cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_teorja INT UNSIGNED NOT NULL,
cod_idioma INT UNSIGNED NOT NULL,
descr VARCHAR(255) NOT NULL,
percentmin INT UNSIGNED NOT NULL,
data DATE NOT NULL,
PRIMARY KEY (cod, cod_teorja, cod_idioma),
INDEX (cod_teorja),
FOREIGN KEY(cod_teorja) REFERENCES teoria(cod) ON UPDATE CASCADE ON DELETE
RESTRICT,
INDEX (cod_idioma),
FOREIGN KEY(cod_idioma) REFERENCES idioma(cod) ON UPDATE CASCADE ON DELETE
RESTRICT)
TYPE=INNODB
COMMENT='método de aplicação de teorias para avaliação';

```

2.10) Criação da tabela topico:

Esta tabela cadastra os tópicos do questionário de um método de avaliação de planos de governo:

```

CREATE TABLE topico (

```



```

cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_metodo INT UNSIGNED NOT NULL,
ordem INT UNSIGNED NOT NULL,
quantquest INT UNSIGNED NOT NULL,
peso DECIMAL(5, 2) UNSIGNED NOT NULL,
comentario TEXT NULL,
descr VARCHAR(255) NOT NULL,
correcoes TEXT NULL,
PRIMARY KEY (cod, cod_metodo, ordem),
INDEX (cod_metodo),
FOREIGN KEY(cod_metodo) REFERENCES metodo(cod) ON UPDATE CASCADE ON DELETE
RESTRICT)
TYPE=INNODB
COMMENT='tópicos de um questionário de um método';

```

2.11) Criação da tabela questao:

Esta tabela cadastra as questões dentro dos tópicos do questionário caso haja um tópico:

```

CREATE TABLE questao (
cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_topico INT UNSIGNED NULL,
ordem INT UNSIGNED NOT NULL,
peso DECIMAL(5, 2) UNSIGNED NOT NULL,
descr TEXT NOT NULL,
correcoes TEXT NULL,
PRIMARY KEY (cod),
INDEX (cod_topico),
FOREIGN KEY(cod_topico) REFERENCES topico(cod) ON UPDATE CASCADE ON DELETE
RESTRICT)
TYPE=INNODB
COMMENT='questões de um tópico de um método';

```

2.12) Criação da tabela teodoc:

Esta tabela cadastra as questões dentro dos tópicos do questionário caso haja um tópico:

```

CREATE TABLE teodoc (
cod_teorias INT UNSIGNED NOT NULL,
tipo_doc VARCHAR(255) NOT NULL,
INDEX (cod_teorias),
FOREIGN KEY(cod_teorias) REFERENCES teoria(cod) ON UPDATE CASCADE ON DELETE
RESTRICT)
TYPE=INNODB
COMMENT='documento que se encontra disponível';

```

2.13) Criação da tabela criterio:

Esta tabela cadastra os critérios para aplicação dos métodos de avaliação de planos:

```

CREATE TABLE criterio (
cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
cod_usuario INT UNSIGNED NOT NULL,
cod_metodo INT UNSIGNED NOT NULL,
porcentmin INT UNSIGNED NOT NULL,
geral BOOL NOT NULL,
descr VARCHAR(255) NOT NULL,
observacoes TEXT,
PRIMARY KEY (cod, cod_usuario, cod_metodo),
INDEX (cod_usuario),
FOREIGN KEY(cod_usuario) REFERENCES usuario(cod) ON UPDATE CASCADE ON DELETE
RESTRICT,
INDEX (cod_metodo),

```

```
FOREIGN KEY(cod_metodo) REFERENCES metodo(cod) ON UPDATE CASCADE ON
DELETE RESTRICT)
TYPE=INNODB
COMMENT='critério de avaliação de plano de governo';
```

2.14) Criação da tabela pesotopicoc:

Esta tabela cadastra os pesos dos tópicos de um método por critério:

```
CREATE TABLE pesotopicoc (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  cod_criterio INT UNSIGNED NOT NULL,
  cod_topico INT UNSIGNED NOT NULL,
  peso DECIMAL(5, 2) UNSIGNED NOT NULL,
  percentmin INT UNSIGNED NOT NULL,
  PRIMARY KEY (cod, cod_criterio, cod_topico),
  INDEX (cod_criterio),
  FOREIGN KEY(cod_criterio) REFERENCES criterio(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT,
  INDEX (cod_topico),
  FOREIGN KEY(cod_topico) REFERENCES topico(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT)
TYPE=INNODB
COMMENT='peso de um tópico de um método por critério';
```

2.15) Criação da tabela pesoquestaoc:

Esta tabela cadastra os pesos das questões de um método por critério:

```
CREATE TABLE pesoquestaoc (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  cod_criterio INT UNSIGNED NOT NULL,
  cod_questao INT UNSIGNED NOT NULL,
  peso DECIMAL(5, 2) UNSIGNED NOT NULL,
  PRIMARY KEY (cod, cod_criterio, cod_questao),
  INDEX (cod_criterio),
  FOREIGN KEY(cod_criterio) REFERENCES criterio(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT,
  INDEX (cod_questao),
  FOREIGN KEY(cod_questao) REFERENCES questao (cod) ON UPDATE CASCADE ON DELETE
  RESTRICT)
TYPE=INNODB
COMMENT='peso de uma questão de um método por critério';
```

2.16) Criação da tabela plano:

Esta tabela cadastra os planos de governo que serão avaliados:

```
CREATE TABLE plano (
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,
  cod_usuario INT UNSIGNED NOT NULL,
  descr VARCHAR(255) NOT NULL,
  PRIMARY KEY (cod, cod_usuario),
  INDEX (cod_usuario),
  FOREIGN KEY(cod_usuario) REFERENCES usuario(cod) ON UPDATE CASCADE ON DELETE
  RESTRICT)
TYPE=INNODB
COMMENT='planos de governo que serão avaliados';
```

2.17) Criação da tabela avaliacao:

Esta tabela cadastra as avaliações dos planos feitas pelos usuários:

```
CREATE TABLE avaliacao (  
  cod INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  cod_questao INT UNSIGNED NOT NULL,  
  cod_plano INT UNSIGNED NOT NULL,  
  resposta INT NOT NULL,  
  PRIMARY KEY (cod, cod_questao, cod_plano),  
  INDEX (cod_questao),  
  FOREIGN KEY(cod_questao) REFERENCES questao(cod) ON UPDATE CASCADE ON DELETE  
  RESTRICT,  
  INDEX (cod_plano),  
  FOREIGN KEY(cod_plano) REFERENCES plano (cod) ON UPDATE CASCADE ON DELETE  
  RESTRICT)  
TYPE=INNODB  
COMMENT='avaliações dos planos feitas pelos usuários';
```