

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CIÊNCIAS DA COMPUTAÇÃO

**UM APLICATIVO VOLTADO PARA O AUXÍLIO E
FORMALIZAÇÃO DO PROCESSO
DE SELEÇÃO DE PESSOAS**

Shalmanny Lee Barduzzi
Wanderson Rigo

Florianópolis, 2005

Shalmanny Lee Barduzzi
Wanderson Rigo

**UM APLICATIVO VOLTADO PARA O AUXÍLIO E
FORMALIZAÇÃO DO PROCESSO
DE SELEÇÃO DE PESSOAS**

Prof. Dr. Mauro Roisenberg
Orientador

Alessandro Vieira dos Reis
Co-orientador

Florianópolis, 2005

“Cada vez que eu indico alguém para um cargo crio dez inimigos e um ingrato.”

MOLIÈRE

SUMÁRIO

01-INTRODUÇÃO	7
02-MOTIVAÇÃO	10
03-CONTEXTUALIZAÇÃO	13
3.1-COMPETÊNCIAS	13
3.2-MOVIMENTO DA COMPETÊNCIA	16
3.3-ORIGEM DO MOVIMENTO DE VALORIZAÇÃO DA COMPETÊNCIA	18
3.4-MUDANÇA DE VALORES NAS AVALIAÇÕES DE PESSOAS E PROFISSIONAIS	19
3.5- ANÁLISE DE COMPETÊNCIAS	19
04-O SOFTWARE	22
4.1-CONCEPÇÃO	22
4.2-OBJETIVOS DO SOFTWARE	23
4.3-FUNIONAMENTO	23
05-EMBASAMENTOS TEÓRICOS PARA A ELABORAÇÃO DO SOFTWARE	30
5.1-AVALIAÇÃO PSICOLÓGICA	30
5.2-PSICOMETRIA	31
5.3- ALIANDO A TEORIA PSICOLÓGICA AO SOFTWARE	32
5.4-INTERPRETANDO AS SAÍDAS DO SOFTWARE	34
06-INTELIGÊNCIA ARTIFICIAL	38
6.1-RETROSPECTO	38
6.2-OBJETIVOS DA IA	39
6.3-DESAFIOS DA IA	39
6.4-ABORDAGENS DA INTELIGÊNCIA ARTIFICIAL	40
6.5-REDES NEURAS	41
6.5.1-BREVE HISTÓRICO	42
6.5.2-MORFOLOGIA	42
6.5.3-APRENDIZADO	45
6.5.4-CARACTERÍSTICAS	46
6.5.5-AVALIAÇÃO	48
6.6-DEFINIÇÃO DO MODELO DE RNA USADA	49
6.6.1-TREINAMENTO DA SOM	50
07-METODOLOGIA ADOTADA PARA O USO DA RNA	52
7.1-COLETA DE DADOS 7.1.1-PERFIL DE CARACTERÍSTICAS	52
7.1.2-QUESTIONÁRIO	55
7.2-FORMATAÇÃO DOS DADOS	61
7.3-CONFIGURAÇÃO DA REDE	62
7.4-TREINAMENTO/TESTE DA REDE	63
08-CONFRONTANDO OS DADOS SOFTWARE X REDE NEURAL AUTO-ORGANIZÁVEL	64
8.1- CLASSIFICAÇÃO EFETUADA PELO SOFTWARE	64
8.2- CLASSIFICAÇÃO EFETUADA PELA RNA	65

8.3-DISCREPÂNCIAS ENTRE AS CLASSIFICAÇÕES	69
09-CONCLUSÃO	75
10-APÊNDICES.....	78
A- CÓDIGO FONTE SOFTWARE.....	79
B-CÓDIGO FONTE RNA	222
C-SUGESTÕES	224
11- REFERÊNCIAS	225
12-ARTIGO.....	228

LISTA DE FIGURAS

Figura 1: Fluxograma dos dados.	8
Figura 2: Processo de construção de competências [CRUZ; PEREIRA; SOUZA, 2004]. ..	15
Figura 3: Exemplo de Matriz de Diferencial Semântico.	33
Figura 4: Fórmula para cálculo do Desvio Padrão.	34
Figura 5: Exemplo de Pirâmide de Osgood bem formatada.....	35
Figura 6: Exemplo de Pirâmide de Osgood mal formatada.....	35
Figura 7: Fluxograma de parte do funcionamento do <i>software</i> Perfil.....	37
Figura 8: Neurônio Biológico.....	42
Figura 9: Exemplo de RNA com as conexões destacadas (adaptado de [BARRETO, 2001]).	44
Figura 10: Neurônio Artificial (adaptado de [BARRETO, 2001])......	45
Figura 11: Topografia de uma Rede Auto-Organizável (Mapas Auto-Organizáveis).	50
Figura 12: Matriz Semântica oriunda da Seleção Hipotética.	54
Figura 13: Dados gerados durante a elaboração do Perfil de características para a Seleção Hipotética.	54
Figura 14: Pirâmide de Osgood da Seleção Hipotética.	55
Figura 15: Pontuação dos candidatos.	65
Figura 16: Saída da Rede Neural Auto-Organizável (5000 épocas).	67
Figura 17: Saída da Rede Neural Auto-Organizável (5000 épocas).	68
Figura 18: Saída da Rede Neural Auto-Organizável (3500 épocas).	69
Figura 19: Saída da Rede Neural Auto-Organizável (exemplo).....	72
Figura 20: Saída da Rede Neural Auto-Organizável (exemplo).....	74
Figura 21: Pontuação dos candidatos (exemplo).....	74

01 INTRODUÇÃO

A competição por melhores resultados e desempenho nas organizações já atinge os níveis tecnológicos há décadas e agora, mais do que nunca, procura no fator humano buscar o diferencial que irá projetar a expoentes certas instituições do cenário contemporâneo. Embora a oferta de mão-de-obra seja crescente, sabe-se que somente a escolha de profissionais capacitados pode oferecer o respaldo imediato que é esperado pelas organizações, pois, como disse o diretor de recrutamento da Microsof, David Pritchard em [POUNDSTONE, 2005], “A melhor coisa que podemos fazer por nossos concorrentes é contratar mal”.

Dado este panorama, indaga-se: como selecionar os mais aptos, os que possuem determinadas características e aptidões, bem como demonstram as competências exigidas, enfim, os candidatos que atendam ao perfil desenhado pelo detentor da vaga e com certeza terão capacidades para elevar o potencial da instituição? [GRAMIGNA,2003]

Visando dar subsídios aos profissionais que diariamente convivem com esta questão e tomando a ótica das organizações frente aos processos seletivos, o presente trabalho apresenta o *software Perfil*, o qual foi desenvolvido com a incumbência de ser **uma ferramenta para amparar** de forma segura e significativa os processos de seleção nas instituições. A fim de atingir esses objetivos, a metodologia usada em sua implementação contempla o uso de mecanismos transparentes de avaliação e confronto de informações.

O *software*, que é concebido para o ambiente Web, além de prover suporte ao trabalho dos profissionais de RH e às demais instituições envolvidas com processos seletivos, visa agilizar o processo de elaboração do perfil de características pessoais e profissionais a ser buscado em um candidato pelo detentor da vaga. Ele favorece a aplicação da metodologia de seleção baseada em competências, a qual será esmiuçada nas seções seguintes, porém, por hora, pode ser entendida como “conjunto de habilidades, conhecimentos e atitudes que contribuem para uma atuação de destaque, de excelência em determinados contextos” [GRAMIGNA, 2003].

Quanto aos aspectos técnicos de implementação, o *software Perfil* foi desenvolvido com as tecnologias e ferramentas que mais se destacam na atualidade e que, apesar de serem expoentes, são todas de distribuição gratuita e possuem uma ampla documentação.

A partir da concepção do protótipo do *software*, este foi disponibilizado para a

execução de testes de usabilidade e aplicabilidade via a elaboração de uma seleção hipotética para o Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de Santa Catarina (PET-CCO). Os testes foram realizados junto ao grupo de bolsistas desse programa e ainda contaram com a participação de outros voluntários, os quais foram convidados via *e-mail*. Como produto destes testes, foram obtidas críticas e sugestões, as quais encontram-se relatadas no Apêndice C. Algumas não foram atendidas pois fugiam do escopo deste trabalho, já que o objetivo principal dos testes foi a **coleta de um conjunto de dados**, os quais correspondem às respostas que os usuários do *software* Perfil deram às questões elaboradas para a seleção hipotética do PET-CCO.

A partir desses dados, o programa classificou decrescentemente os usuários mediante a soma da pontuação atribuída a cada alternativa das respostas dadas por eles no questionário. Esses dados também foram fornecidos a uma Rede Neural Artificial Auto-Organizável, a qual, de acordo com as características das entradas (as respostas do questionário aplicado pelo *software*) que recebeu, tentou agrupar os candidatos em nichos, sendo que estes representam grupos de candidatos com características semelhantes.

Por hora, é necessário esclarecer que Redes Neurais são modelos computacionais não lineares inspirados na estrutura e operação do cérebro humano que procuram reproduzir características humanas, tais como: aprendizado, associação, generalização e abstração. Devido à sua estrutura, as Redes Neurais são bastante efetivas no aprendizado de padrões a partir de dados não-lineares, incompletos, com ruído e até compostos de exemplos contraditórios. Assim sendo, esperava-se como saída da Rede uma possível triagem e agrupamento dos candidatos que responderam ao questionário implementado pelo *software*.

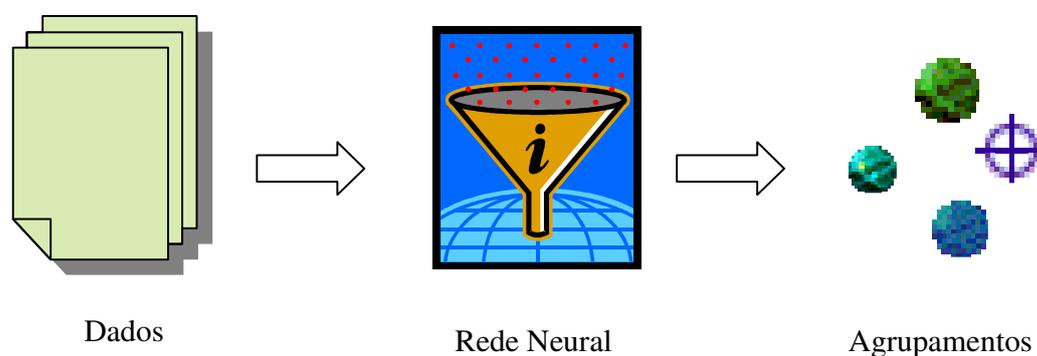


Figura 1: Fluxograma dos dados.

Esta classificação serviu como ponto de partida para o confronto das informações obtidas pelo *software* Perfil e pela Rede Neural, o que propiciou uma avaliação de ambos os instrumentos.

Logo, definiu-se como objetivo principal deste trabalho **o desenvolvimento de um software para auxiliar e formalizar os processos seletivos, bem com a comparação entre a classificação realizada por este software e a classificação realizada pela Rede Neural Artificial.**

02 MOTIVAÇÃO

A busca pela excelência operacional e funcional das instituições vai muito além da admissão de pessoas para preencher os cargos e as vagas oferecidas. O que se quer é a busca de perfis específicos para o sucesso em atribuições específicas. Para se conseguir isso é preciso ter técnica e ferramentas que permitam um trabalho de qualidade na avaliação/confronto das características dos candidatos em relação ao perfil delineado como ideal [BISPO, 2003].

De certa forma, deve-se traduzir as expectativas da gerência em um perfil passível de ser avaliado, mensurado, descrito em competências e que sinalize em direção à assertividade nas escolhas [GRAMIGNA, 2003]. Além disso, sabe-se que os processos de avaliação **concebidos intuitivamente, sem método**, pecam em vários quesitos, pois um avaliador que esteja mal humorado, cansado, ou que não simpatize com um candidato, com certeza fará uma avaliação equivocada deste.

A fim de ilustrar essa falta de objetividade científica nas seleções, é pertinente aqui citar um trecho curioso é arrebatador retirado de [POUNDSTONE, 2005]. Eis ele: “Nalini Ambady e Robert Rosenthal, dois psicólogos de Harvard, fizeram uma experiência particularmente arrasadora. No início, Ambady desejava pesquisar o que determinava a eficiência dos professores. Suspeitava de que sinais não verbais, como linguagem do corpo, fossem importantes. Para testar sua hipótese, usou alguns vídeos em que professores de Harvard foram filmados. Pensou em mostrar para um grupo de pessoas clipes silenciosos do vídeo e pedir que **avaliassem a eficiência dos professores**. Ambady pretendia usar clipes com duração de um minuto de cada professor. Infelizmente, os vídeos não foram filmados com este objetivo e mostravam os professores interagindo com os alunos. Isso era um problema porque a presença de estudantes nos vídeos poderia afetar a opinião dos avaliadores, ainda que inconscientemente. Ambady avisou o seu orientador que isso não iria funcionar.

Assistindo aos vídeos pela segunda vez, ela verificou que poderia passar filmes de dez segundos em que os alunos não apareceriam. Fez o estudo com esses clipes de dez segundos. Baseados apenas neles, os avaliadores julgaram as qualidades dos professores a partir de uma lista de quinze itens.

Está certo, se você tem de julgar alguém a partir de um filme de dez segundos, você

pode fazer isso, mas, provavelmente, não espera que esse julgamento tenha algum valor.

Ambady repetiu a experiência editando um filme de cinco segundos dos mesmos professores. Outro grupo de avaliadores os julgou. Suas avaliações foram, descontando erros estatísticos, idênticas às avaliações feitas pelo grupo que viu o filme de dez segundos.

Ambady formou então um outro grupo de avaliadores que dessa vez assistiu a filmes de dois segundos com os mesmos professores. Novamente, as notas foram essencialmente as mesmas.

O mais interessante foi que Ambady comparou as notas dadas pelos avaliadores às notas que os alunos haviam dado aos mesmos professores no final do semestre. Os alunos conheciam bem mais que os avaliadores. Ainda assim, suas avaliações ficaram muito próximas das avaliações feitas pelas pessoas que conheciam os professores apenas pelos clipes. **As opiniões de um estranho que assistiu a um vídeo de dois segundos eram quase as mesmas das opiniões dos alunos que tiveram contato com os professores durante um semestre inteiro.**

Parece que as pessoas conseguem julgar instantaneamente alguém em dois segundos sem se basear em algo que essa pessoa tenha dito. Muito raramente, alguma coisa que aconteça após esses dois segundos afeta de forma significativa a primeira impressão do avaliador.”

Pelo exposto acima é possível se supor que nas entrevistas das seleções os fatores subjetivos imperam sobre a racionalidade, delegando à entrevista um papel de “acerto de contas”, onde as perguntas e respostas que se seguem são um pretexto para justificar e amparar a primeira impressão, uma maneira de convencer a ambos de que existe uma base racional para a decisão de contratar ou não. Na verdade, a decisão já foi tomada em bases que não poderiam ser mais superficiais [POUNDSTONE, 2005]. Corroborando com essa opinião, o co-orientador deste trabalho, Alessandro Vieira do Reis, o qual é gestor de RH, em certa ocasião mencionou: “prefiro testes escritos a entrevistas”.

Ainda que a avaliação seja amparada por instrumentos psicológicos, como psicométricos ou projetivos, existe toda uma tarefa de processamento dos dados obtidos junto aos candidatos. Essa tarefa, na maioria das vezes, é realizada por pessoas, fato que torna o processo passível de erros e impropriedades devido ao manuseio de muitas informações, à distração, à atividades repetitivas de cálculo etc.

Valendo-se desses argumentos propõe-se uma ferramenta que venha auxiliar psicólogos, profissionais de RH e até leigos. Ela, além de contornar os entraves relatados anteriormente, fará a parte mecânica da seleção, livrando o profissional do trabalho não-criativo, repetitivo e que pode ser automatizado. Visa-se, dessa forma, isentar o processo de incorretudes, além de, é claro, indicar/sugerir os candidatos que mais se aproximam do perfil desejado pela organização.

Ressalta-se que esta metodologia irá estimular as organizações a definirem um perfil do profissional que estão buscando. Isso, de certa forma, já é positivo pelo fato de questionar que características pessoais e profissionais são mais valorizadas dentro dessas organizações. Dito de outra forma: a metodologia determina um “norte de características” a ser buscado nos candidatos quando o assunto é a contratação de novos profissionais.

Cabe aqui ressaltar que toda a saída do *software* será avaliada por um profissional conhecedor das atribuições que são buscadas nos candidatos. Assim, a ferramenta **não procura desprestigiar e muito menos substituir a avaliação humana que vem sendo feita nos processos seletivos**. O que se anseia é que o uso dessa ferramenta instigue a construção de um perfil claro, bem como vise dotar ao processo de seleção facilidades para avaliar os candidatos com imparcialidade e ética. Também pretende-se dar consistência, foco e objetividade às seleções, tendo como consequência resultados mais eficazes, os quais estarão refletidos na indicação de profissionais mais capacitados para o cargo alvo.

Sendo assim, é evidente que o *software* Perfil necessita de um profissional qualificado para elaborar as questões que farão emergir dos candidatos as suas características e, na medida do possível, as suas competências pessoais. Ainda este mesmo profissional irá emitir um laudo devolutivo a partir das saídas do *software* visando dar um *feedback* aos candidatos participantes do processo seletivo. Nesta ocasião, a qual pode constituir-se uma entrevista de devolução, ele apresentará os resultados e esclarecerá as dúvidas do candidato examinado, fundamentando os porquês da escolha ou não de sua pessoa [ALCHIERI; CRUZ, 2003]. Não finda aqui o trabalho do profissional, já que o mesmo poderá ainda preparar as entrevistas abertas baseado-se no laudo previamente emitido.

Portanto deixa-se explicitamente claro que o objetivo do *software* Perfil é a **instrumentalização** do processo seletivo e **não a sua automatização**.

03 CONTEXTUALIZAÇÃO

O mundo evolui por fases cíclicas e dentro delas acontecem movimentos evolutivos de diversas naturezas. Tais movimentos determinam o surgimento de novas demandas e necessidades para as sociedades, para as organizações e para as pessoas. São essas necessidades que levam a sociedade a se interessar ou a se dedicar mais a determinados assuntos e a mudar paradigmas conceituais, estruturais, de valores e comportamentais, bem como a redirecionar os esforços nas pesquisas e estudos.

Quando o assunto se refere a seleção de pessoal, o foco atual das instituições tem se voltado para as seleções baseadas em competências, a última palavra para tentar resolver o princípio básico da seleção e promoção: “colocar a pessoa certa no lugar certo” [RESENDE, 2003]. Em virtude disso, é necessário esclarecer alguns pontos inerentes a competências e isso será feito nessas próximas seções.

Porém não passou despercebida e é útil citar aqui a crítica que Wanderley Codo [LANE; CODO, 1986] faz a este princípio, o qual é o alvo predileto de crítica à Psicologia do Trabalho. Segundo Codo, buscar a escolha do homem certo para o lugar certo (*right man to the right place*) é “uma tentativa de eliminação do indivíduo que trabalha, desvalorizando-o e eliminando a sua dignidade, pelo menos do ponto de vista psicológico. Trata-se de transformar o trabalho do operário em força de trabalho e utilizá-lo como qualquer outra força (elétrica, mecânica) no processo produtivo”. Esta filosofia, ainda segundo Codo, só teria sentido em “um capitalismo em expansão, com taxas de crescimento superiores ao crescimento vegetativo da oferta de mão de obra”.

3.1 COMPETÊNCIAS

O projeto de uma pessoa ser alguém numa sociedade está intimamente relacionado a organização e planejamento da vida profissional dela. Tudo isso pode ser traduzidos por um processo contínuo de **construção de competências**, o qual não se restringe apenas ao ambiente escolar e profissional, mas engloba todas as esferas da vida de uma pessoa. A partir de um processo dessa magnitude é que os indivíduos podem tornar-se vendedores daquilo que são capazes de fazer, além de constituírem a força de trabalho que os torna atraentes aos olhos do mercado de trabalho [CRUZ; PEREIRA; SOUZA, 2004].

Dada essa importância, torna-se necessário esclarecer o que são e como caracterizar

competências, já que ainda não abordou-se formalmente esta definição. Sem maiores delongas, parte-se para o trabalho então...

Segundo afirma Cruz, “as pessoas não possuem competências, mas demonstram.” Seu ponto de vista está embasado em [CRUZ; PEREIRA; SOUZA, 2004], artigo em que ele, juntamente com outros autores, discorre sobre “Competências, perfis profissionais e mercado de trabalho em psicologia.” Para o escopo desse trabalho julgou-se útil extrair daquele a definição do que são “competências”, sendo que aqui são elencados alguns trechos que bem retratam o tema.

Cruz defende que “um aspecto importante na caracterização de um comportamento competente diz respeito a uma ênfase nos resultados dos atos realizados (desempenho)” e compartilha essa perspectiva com Foster e Ritchey [FOSTER, RITCHEY, apud CRUZ, PEREIRA, SOUZA, 2004] e com Hubbard e Coie [HUBBARD, COIE, apud CRUZ, PEREIRA, SOUZA, 2004], já que eles definem competência como **“a capacidade de ser eficaz na realização de objetivos a partir de metas estabelecidas previamente...”**

Ainda nesse artigo, Rey [REY, apud CRUZ, PEREIRA, SOUZA, 2004] explana que **“competência pode ser definida por comportamentos observáveis, exteriores e impessoais em uma determinada situação.”** De certa forma isso converge para a forma como Fleury e Fleury [FLEURY, FLEURY, apud CRUZ, PEREIRA, SOUZA, 2004] abordam competência, pois para eles as pessoas só serão competentes a partir dos resultados obtidos em suas atuações, já que **“os conhecimentos e o *know how* não adquirem status de competência a não ser que sejam comunicados e utilizados.”**

A fim de se ampliar o discurso, procurou-se em outras fontes mais algumas perspectivas a respeito de competências. Essas contribuições são mostradas logo adiante, sendo que inicialmente explicita-se a opinião de Enio Rezende.

Ele defende que o conceito de competência se aplica a uma característica ou um conjunto de características ou requisitos. **“Conhecimento ou uma só habilidade ou aptidão, por exemplo, indicados como uma condição capaz de produzir efeitos de resultados, de soluções de problemas, pode ser chamado de competência.”** Entende-se, de forma pragmática, como ações voltadas para se obter resultados, ou seja, é a **“transformação de conhecimentos, aptidões, habilidades, interesse, vontade etc. em resultados práticos”**. Também **“são as idéias claras ou implícitas de saber fazer, saber**

aplicar, saber agir, saber resolver, já que saber operacionalizado tem como produto os resultados” [RESENDE, 2003].

Ainda segundo Resende, competência “é resultante da combinação de **conhecimentos** com **comportamentos**. Conhecimentos que incluem formação, treinamento, experiência, autodesenvolvimento, prática, teoria. Comportamentos que englobam habilidades, interesse, atitudes, vontade”.

Julgou-se mais completa a forma como em [CRUZ; PEREIRA; SOUZA, 2004] tratou-se do processo de construção de competências. Aborda-se que “**de acordo com as situações, as experiências vividas e as aprendizagens são transformadas em conhecimentos e habilidades que, por sua vez, são a base para a formação de atitudes, a qual constrói a capacidade de atuar**”.

Baseado nisso e compactuando com a ilustração que descreve esse processo, adaptou-se de [CRUZ; PEREIRA; SOUZA, 2004] o seguinte esquema:

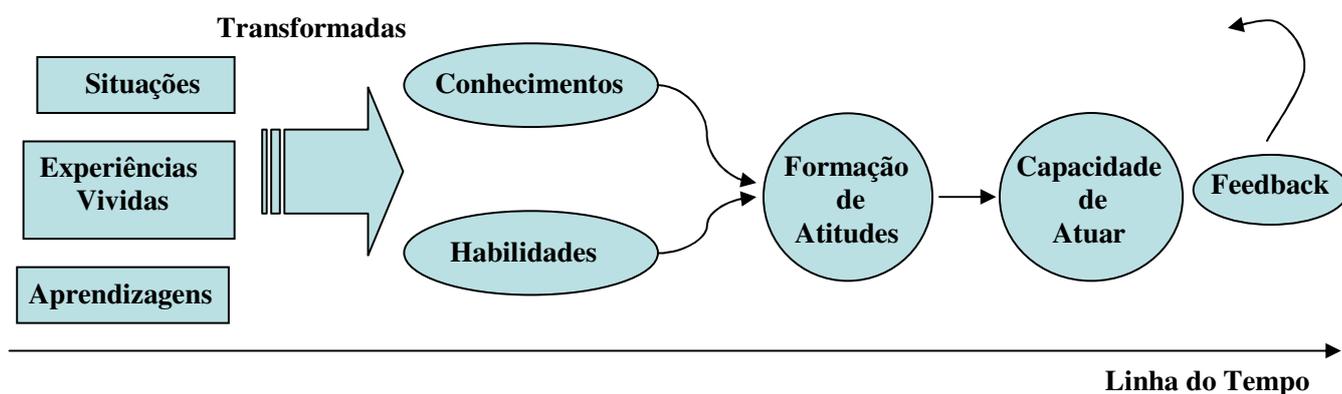


Figura 2: Processo de construção de competências [CRUZ; PEREIRA; SOUZA, 2004].

“Esse processo ocorre de forma contínua no decorrer da vida das pessoas. Novos conhecimentos e habilidades surgem na própria atuação, assim como novas demandas ou necessidades sociais no trabalho, fazendo com que os profissionais tenham que novamente buscar conhecimentos para o aprimoramento de sua prática profissional. Dessa forma, as competências são construídas fundamentalmente por meio de *feedback* positivo, por um processo de melhoria contínua.”

Baseando-se nos parágrafos anteriores e fazendo-se uma síntese a respeito do que os diversos autores versam sobre o tema competência, os autores deste trabalho concluem que Competência pode ser entendida como **“capacidade ou habilidade de atuar eficientemente na realização de alguma atividade, sendo que a motivação pessoal somada a conhecimentos previamente assimilados produz resultados práticos”**.

3.2 MOVIMENTO DA COMPETÊNCIA

A valorização da competência constitui uma importante mudança de paradigma no que tange a conceitos e valores e terá grande influência nos destinos das organizações, nas carreiras das pessoas e em evoluções na sociedade. Antes valorizava-se as pessoas pelo perfil intelectual, formação acadêmica e prestígio da escola freqüentada. Hoje essa importância é relativa e os quesitos preponderantes que guiam a busca de profissionais agora são definidos pelas **características das pessoas que fazem bem o trabalho** [RESENDE, 2003].

Existe uma causa comum da valorização recente dos conceitos de qualidade e competência: o aumento da competitividade no mundo das empresas e dos negócios, o qual se deve à globalização da economia, à rápida evolução das tecnologias e os efeitos das crises econômicas. Resende explicita isso da seguinte forma:

Competitividade na Economia: É o fator determinante mais forte, pois impacta com tudo aquilo que está envolvido com o mundo dos negócios, no qual as mudanças e evoluções acontecem com mais agilidade e efetividade. A cobrança por melhores padrões de qualidade e menores preços obrigou as organizações que produzem e prestam serviços a encontrarem maneiras e meios de maximizar o desempenho dos profissionais, das equipes, das lideranças. Dentre as soluções buscadas, duas se destacam como fundamentais:

- 1) melhorar a utilização do potencial humano (identificar e desenvolver competências e habilidades);
- 2) melhorar a eficácia (competência) organizacional e de gestão dos recursos utilizados.

Evolução da Democracia: Mudanças positivas no comportamento das pessoas

na sociedade, favorecidas pelo fortalecimento da democracia que, por sua vez, contribui para o surgimento de entidades que estimulam ações de cidadania em direção a cobranças por melhor qualidade de produtos e serviços. Então, para oferecer melhor qualidade de produtos e serviços, as empresas precisam desenvolver suas competências técnicas, operacionais, gerenciais e profissionais.

Todos esses fatores mencionados obrigam as empresas a adquirirem as competências negligenciadas no passado de “vacas gordas”, quando qualidade e produtividade ficavam em segundo plano e a preocupação maior era controlar a conduta dos empregados em vez de cuidar da qualificação e desempenho dos mesmos. Esse aumento da competitividade faz crescer a preocupação com a melhoria de resultados que, por sua vez, conduz à busca de novas formas ou recursos para se conseguir isso. **A busca mais atual de novas soluções para melhorar desempenho e resultados de organizações e pessoas leva ao fator competência** [RESENDE, 2003].

Um ponto a ser considerado quanto ao surgimento seqüencial das ondas que evidenciam o valor da qualidade, primeiro, e da competência, agora, e que merece destaque como fator importante para o sucesso das pessoas e das organizações, é que a busca de melhoria da qualidade de sistemas e processos organizacionais e de produtos e serviços não foi suficiente para assegurar o sucesso e a sobrevivência das empresas [RESENDE, 2003].

Ao contrário de outros assuntos relacionados com o desempenho das organizações – eficiência e qualidade, por exemplo - **competência é o primeiro que leva em conta as pessoas**, até porque competência se manifesta principalmente por meio de pessoas, enquanto qualidade tem sido mostrada por meio de produtos ou operações, embora acabe ficando demonstrado, posteriormente, que o **efeito** qualidade de produtos ou serviços é resultante da **causa** competência das pessoas, as quais atuam na condição de profissionais ou gerentes [RESENDE, 2003].

Tudo leva a crer que o movimento da competência surgiu como exigência da necessidade de aumentar a capacidade de resolver problemas e melhorar desempenhos, sendo que esses processos estão centrados nos recursos humanos, ou melhor, no tratamento das pessoas de forma singular.

3.3 ORIGEM DO MOVIMENTO DE VALORIZAÇÃO DA COMPETÊNCIA

Tendo por base [RESENDE, 2003], aqui insere-se um trecho do seu livro que descreve a origem do movimento da competência. Esta seção tem o propósito de sanar mais a curiosidade do que ser uma referência científica. Eis ele:

“David McClelland, no início da década de 70, realizou estudos enquanto participava de processo de seleção de pessoal para o Departamento de Estado americano. Ai se iniciava a aplicação do conceito de competência em psicologia. McClelland publicou, em 1973, um artigo intitulado “*Testing for Competence Rather Than Intelligence*”, no qual afirma que os testes tradicionais de conhecimento e inteligência não eram capazes de prever sucesso no trabalho e na vida e favoreciam preconceitos contra minorias, mulheres e pessoas de nível socioeconômico inferior.

Essa constatação o levou a encontrar métodos de avaliação que lhe permitissem identificar variáveis de competência que pudessem prever êxito nas atuações no trabalho e na vida e selecionar pessoas sem discriminação.

McClelland desenvolveu e aplicou uma técnica especial para destacar as variáveis de comportamento que explicassem porque determinados diplomatas do Departamento de Estado tinham sucesso, e outros não, em suas difíceis missões em países onde havia forte rejeição à presença dos EUA. Os resultados dessa pesquisa indicaram claramente diferenças de habilidades, aptidões e atitudes entre os melhores embaixadores e os medíocres. A identificação dessas características pessoais que faziam diferença no resultado das ações desses dois grupos permitiu a McClelland e sua equipe:

- propor e enfatizar um novo conceito para competência;
- confirmar as suposições a respeito das limitações dos testes tradicionais para avaliar competências;
- propor novos métodos e critérios de seleção e avaliação de pessoal.”

Segundo Rezende, uma seleção por competências pode evitar a frustrante situação em que um profissional é escalado para trabalhar em um posto de trabalho para o qual não possui os requisitos de qualificação, aptidões e habilidades. Uma boa seleção pode-se evitar a frustração, o desajuste, o estresse, a perda de autoconfiança, a baixa auto-estima e o

desgaste. São nítidas as conseqüências negativas que podem ser geradas ao não se atender ao princípio básico da seleção e da promoção: “colocar a pessoa certa no lugar certo” [RESENDE, 2003].

3.4 MUDANÇA DE VALORES NAS AVALIAÇÕES DE PESSOAS E PROFISSIONAIS

A partir dos estudos de McClelland começava uma importante e significativa mudança de paradigma na valorização de requisitos para o trabalho, profissão e carreira. Tradicionalmente se tem atribuído grande importância ao perfil intelectual e à formação acadêmica, bem como ao prestígio da escola freqüentada. Dito de uma forma mais direta: *curriculum vitae* com indicação de escolas renomadas e quantidade de cursos, associada a resultados de testes psicométricos com altos percentis de inteligência (Q.I.), eram o suficiente para garantir o emprego. Se a pessoa fracassasse, depois, no trabalho, seriam procuradas desculpas na empresa [RESENDE, 2003].

A valorização e popularização do novo conceito de competência tendem a determinar uma importância relativa ao fator escolaridade, pelo menos para fins de seleção e promoção de pessoal, além de que agora em vez de se dar mais importância às características do cargo ou do trabalho, passou-se a dar mais importância às **características das pessoas que fazem bem o trabalho**. Tal fato já está provocando mudanças de foco na seleção, no treinamento, na renumeração, nas avaliações de desempenho, agora destacando mais as competências requeridas das pessoas, já que, segundo [REIS, 2005], “os reflexos das competências se mostram mais em pessoas com **atitude**, isto é, no modo pessoal de ver as coisas, baseado-se em crenças, valores.”

3.5 ANÁLISE DE COMPETÊNCIAS

O *software* desenvolvido, atento às mudanças e integrando-se ao contexto da sociedade, agrega mecanismos que permitem que pessoas com histórico de sucesso assinalem as características que julgam essenciais para um futuro colega de profissão,

sendo que esses dados serão posteriormente usados para montar o perfil ideal relativo a aquele grupo de pessoas. Em outras palavras, identifica-se as características e competências das pessoas que fazem bem o trabalho, sendo que essas características serão usadas como balizadores na seleção: a partir das respostas de um questionário personalizado elaborando em vista dessas qualidades (focado-se nelas), será possível analisar quais candidatos se aproximam mais desses predicados. Estes, possivelmente, obterão mais êxito em suas atividades e irão gerar resultados mais nobres para suas organizações.

Então, descrita essa pequena panacéia dos processos seletivos, pergunta-se: como será possível fazer emergir em um questionário as competências de um candidato?

De acordo com [REIS, 2005], “através de **auto-relato**, já que um questionário preenchido por uma pessoa é sempre um auto-relato, sendo que neste o candidato **irá projetar a imagem ou representação que tem de suas competências**. Ainda não se demonstra resultados de nada, apenas **alega-se** que aquele preenchimento que ela fez no questionário tem congruência com algum fato. Por exemplo, se a pessoa marca 8, numa escala de 0 a 10, para seu estado de alegria momentâneo, ela está **alegando** que encontra-se alegre em nível 8, fato que, não necessariamente pode ser verdade.”

Esta afirmação não preenche completamente todos os preceitos pregados pela definição de competências que foram abordados previamente, já que, reiterando Fleury e Fleury, “os conhecimentos e o *know how* não adquirem status de competência a não ser que sejam comunicados e utilizados”. Nota-se a peculiaridade de que um questionário abre brechas para a falta de compromisso com a verdade, além de que é difícil captar e utilizar esses “conhecimentos que devem ser comunicados e utilizados”.

Porém, segundo Reis, “ainda assim o processo se caracteriza como uma seleção baseada em competências, só que competências alegadas. Logo, é possível, no momento da elaboração do questionário, focá-lo de modo que seja possível explorar e captar o que o candidato alega ter em termos de:

1. atitudes (mistura de crenças, valores);
2. habilidades técnicas (línguas, linguagens de programação, etc.);
3. conhecimentos teóricos (currículo acadêmico);

Se for levado em conta que 1+2+3 formam as competências do candidato, conclui-se que o *software* **capta as competências que o candidato alegar ter** e, quanto mais eficiente for o *software* em medir estas competências, mais congruência entre a alegação e as competências reais haverá”.

A última ressalva defendida por Reis é a de que “parte-se do pressuposto de que o candidato está sendo honesto e, caso ele não seja, a psicometria envolvida irá captar discrepâncias em suas respostas. Ainda se, na pior das hipóteses, o candidato que faltou com a verdade prosseguir no processo seletivo, ele será pego em uma fase posterior da seleção (uma entrevista ou dinâmica de grupo, por exemplo)”.

Frisa-se que o *software* não pretende ser a palavra final nos processos seletivos, mas sim o direcionador inicial dos passos de todo o processo.

04 O SOFTWARE

Nesta seção serão abordados aspectos da implementação e das funcionalidades do *software* Perfil. Por meio dele é que foram coletados os dados usados como entradas para a Rede Neural Auto-Organizável. Não se discorre a respeito da estruturação interna do *software*, suas classes, lógica de funcionamento etc., já que esses detalhes foram abstraídos em prol de se aprofundar na descrição dos comportamentos e funcionalidades que o *software* provê aos usuários finais (candidatos, funcionários, administrador).

4.1 CONCEPÇÃO

O *software* Perfil foi desenvolvido para o ambiente Web em virtude da portabilidade propiciada pelos navegadores. Esse ambiente também facilita o acesso por parte dos usuários, os quais podem estar fisicamente em qualquer lugar do planeta.

Ele foi implementado seguindo a arquitetura MVC (*Model-View-Controller*). A MVC é um padrão que separa de maneira independente o Modelo, que representa os objetos de negócio (*Model*), da camada de apresentação (*View*), que compreende a interface com o usuário ou outro sistema; e o Controle de fluxo da aplicação (*Controller*).

Para a implementação da camada *Model* e *Controller*, foi usada a Linguagem Java, indiscutivelmente uma das mais robustas e poderosas existentes no mercado, sem falar, é claro, na independência de plataforma de sistema operacional provida por ela. Para camada *View*, usou-se a linguagem HTML como implementadora do conteúdo estático. Já para suprir a parte de conteúdo dinâmico, utilizou-se a Linguagem JSP (*Java Server Pages*), uma extensão da API (*Application Program Interface*) de Servlets de Java que usa lógica e dados no lado do servidor.

As informações utilizadas no *software* são armazenadas em um banco de dados MySQL, o qual se comunica com a aplicação através da API da tecnologia JDBC (*Java Database Connectivity*).

Toda a aplicação Web precisa ser hospedada em um *container* que suporte conteúdo dinâmico, o qual funciona como servidor de páginas JSP e Servlets. O container Tomcat foi o escolhido para fornecer estes serviços ao *software* desenvolvido neste trabalho.

O ambiente de programação usado para o desenvolvimento do *software* Perfil foi a

ferramenta Eclipse, bem conhecida pelos proponentes desse trabalho. Também optou-se por essa IDE (*Interface Development Enviroment*) porque a mesma propicia aos programadores inúmeras facilidades.

A aplicação está disponível em <http://pet.inf.ufsc.br:8080/tcc3> rodando todos os serviços providos pelas ferramentas acima descritas em um servidor configurado com a distribuição Fedora da plataforma Linux.

Todas as tecnologias usadas são consideradas *softwares* livres, já que podem ser baixadas gratuitamente na Internet. Os seguintes sítios podem ser procurados a fim de se consultar a documentação bem como para se fazer o *download* delas :

- Tomcat: <http://jakarta.apache.org/tomcat>
- Java: <http://java.sun.com>
- JSP: <http://java.sun.com/products/jsp/>.
- MySQL: <http://www.mysql.com/>
- HTML: <http://www.w3.org/MarkUp>
- Eclipse: <http://www.eclipse.org>

4.2 OBJETIVOS DO SOFTWARE

Este *software* dá suporte a seleção de pessoal baseado-se no conceito de competências. Pode-se elaborar um perfil de qualidades a ser buscado nos candidatos do recrutamento e o *software* irá sugerir métricas para a criação de um questionário, o qual servirá para investigar nos candidatos essas características. Isso será feito numa fase posterior, ou seja, no momento em que eles acessarem o sistema para responderem as questões. A fim de cumprir esse objetivo, o *software* prove a criação de um banco de adjetivos, banco de usuários cadastrados, banco de questionários e banco de respostas dos questionários (inerente a cada candidato). Como o módulo de criação do perfil é desvinculado do módulo de criação do questionário, é possível ao administrador, a qualquer momento, criar apenas o questionário, o qual pode ser usado em uma seleção tradicional.

4.3 FUNCIONAMENTO

A fim de esclarecer o uso da ferramenta, descreve-se aqui um resumo das

funcionalidades e casos de uso. Esta descrição também pode ser encontrada na página de “Informações” do programa, assim busca-se esclarecer aos usuários as características de funcionamento do *software* Perfil.

Convencionou-se chamar de **agentes** a todas as pessoas que tem algum contato com o *software*. Assim sendo, algumas definições são úteis para uma melhor compreensão das funcionalidades do programa:

Agentes

Administrador: O principal operador do sistema. Tem total autonomia sobre as funcionalidades e livre navegação entre todas as páginas do *software*. Segundo os idealizadores dessa ferramenta, tem-se em mente que o Administrador seja um Psicólogo(a).

Usuários: Possivelmente serão membros de uma equipe que tenha a necessidade de definir que características esperam de um novo membro. Terão autonomia para colaborar na definição do perfil do candidato a ser escolhido.

Candidatos: São as pessoas interessadas em participar da seleção. Tem acesso somente aos questionários.

Agora são explicitados os casos de uso inerentes a cada tipo de agente:

Administrador

1 Autenticação:

O administrador do sistema acessa a página principal do programa, se identifica mediante o fornecimento de um *login* e de uma senha (o acesso é restrito, pois o módulo resguarda informações de uso exclusivo do administrador do sistema).

2 Controle de Projetos:

Feita a autenticação, ele visualizará a página de “Controle de Projetos”, onde poderá cadastrar e descadastrar projetos de seleção de pessoal mediante o fornecimento do nome do projeto. Também pode selecionar um projeto já cadastrado e gerenciar as configurações que serão descritas na próxima seção.

3 Módulo de Gerenciamento do Projeto:

Várias opções relativas ao projeto previamente selecionado são mostradas ao administrador:

- Controle de Usuários
- Controle de Atributos Desejados
- Computar Atributos / Gerar Matriz Semântica
- Definir Perfil
- Elaborar Questionário
- Visualizar Questionário
- Visualizar Resultados do Teste

3.1 Módulo de Controle de Usuários

O administrador pode cadastrar e descadastrar usuários. Eles terão autonomia para escolher, dentre todos os atributos cadastrados, aqueles indispensáveis ao candidato ideal para ocupar a vaga. Os atributos que mais se destacarem irão compor o perfil de competências. Cada usuário terá um nome, um *login*, uma senha, um endereço de *e-mail*, um CPF e um número de identidade a ser preenchido pelo administrador.

3.2 Módulo de Controle de Atributos Desejados

Responsável pela definição do conjunto universo dos atributos que poderão ser selecionados pelos usuários para montar o perfil de competências. Fornece as funcionalidades de inserção e remoção de adjetivos no espaço amostral. O administrador também define um número *x* de atributos que os usuários terão que escolher dentre todos os fornecidos.

3.3 Módulo Computar Atributos / Gerar Pirâmide Semântica

Após todos os usuários cadastrados escolherem os atributos de sua preferência (Módulo Selecionar Atributos Desejados), o administrador terá disponível a opção de

computar tais atributos. Esse módulo lista em ordem decrescente os adjetivos que foram citados como essenciais para a seleção atual, juntamente com o número de vezes que foram assinalados. Então, o administrador analisando esses dados, pode selecionar os adjetivos que mais foram citados e estes irão compor a Matriz Semântica.

3.4 Módulo Definir Perfil

Logo após todos os usuários cadastrados terem valorado a Matriz Semântica (Módulo Valorar a Matriz Semântica), o administrador terá disponível a média, o desvio padrão e o resultado da diferença entre a média e o desvio padrão de cada atributo. Com base nisso, ele tem uma indicação segura do quão cada atributo julgou-se importante para o grupo de usuários. Isso irá embasar a elaboração do questionário, o qual pode explorar de maneira subliminar em cada questão essas informações computadas na definição do perfil.

3.5 Módulo Elaborar Questionário

Fornece ao administrador uma interface para elaboração de questionários. De acordo com as características que o administrador queira explorar nos candidatos, são necessários determinados formatos de questões. Para esta tarefa, o *software* prove suporte às seguintes modalidades (tipos) de questões:

•Múltipla Escolha

Questões do tipo “checkbox”. Elas possibilitam ao candidato marcar diversas alternativas.

Ex: Minhas cores preferidas são:

Azul;

Vermelho;

Verde;

Amarelo;

•Única Escolha

Questões que instigam o candidato a marcar apenas uma alternativa.

Ex: Como estava o clima ontem:

Quente

Frio

•Multivalorada

Questões que exigem que o candidato quantifique alguma característica.

Ex: seu nível de conhecimento na língua inglesa, numa escala de 0-10, em cada item é:

Compreende

Fala

Lê

Escreve

•Descritiva

Questões abertas, onde o candidato terá a oportunidade de descrever algo.

Ex: Comente sobre a sua experiência em trabalhar em grupo.

3.5 Módulo Visualizar Questionário

Nesse módulo é possível analisar e excluir, a qualquer tempo, as questões que já compõe o questionário.

3.6 Módulo Visualizar Resultados do Teste

Após os candidatos responderem o questionário, é possível visualizar a pontuação e as respostas de cada um deles por meio desse módulo. Uma tabela contendo o nome, RG, CPF, *e-mail* e a pontuação de cada candidato é mostrada. Pode-se averiguar o desempenho individual que cada um alcançou no questionário. Basta selecionar algum candidato e depois clicar no botão “Mostrar Resposta”.

Feito isso, será possível avaliar as respostas que o candidato deu a cada pergunta, bem como os valores associados a cada resposta. Isso dá transparência ao processo seletivo, já que é possível demonstrar através da soma das respostas como foi construído o escore de cada candidato. Com base nisso, o psicólogo poderá gerar um laudo de cada candidato, o qual explicará o desempenho que o aspirante teve no questionário, o que se esperava dele, apontar alguma contradição nas respostas, porque foi aprovado ou não etc.

Usuários

1 Módulo Selecionar Atributos Desejados:

Os usuários, após identificarem-se no sistema via *login* e senha (definidos pelo administrador), terão autonomia para escolher o número *x* (também definido pelo administrador) de atributos que julguem indispensáveis ao candidato ideal para ocupar a vaga da seleção em questão. A seleção é baseada em um espaço amostral, definido pelo Administrador do sistema. Cada atributo selecionado é incluído em uma lista de atributos “Selecionados”, a qual pode ser alterada pelo usuário via inclusão e exclusão de atributos. Depois de computada as respostas de todos os usuários, os atributos que mais se destacaram irão compor o perfil de competências.

2 Módulo Valorar a Matriz Semântica

Somente depois de todos os usuários cadastrados no projeto terem selecionado os *x* atributos requisitados, o administrador poderá gerar a Matriz Semântica, a qual será valorada pelos usuários. Ela será formada pela escolha do administrador dos *x* adjetivos que mais foram citados dentre todos os usuários cadastrados no projeto.

O processo de valorar a matriz consiste em atribuir uma nota entre 0 e 5 para o grau de similaridade entre dois atributos, sendo que a matriz é montada de modo que todos os *x* atributos sejam combinados entre si.

Candidatos

1 Módulo Selecionar Projeto

Os candidatos poderão escolher dentre os projetos seletivos vigentes qual (quais) é (são) de seu interesse participar. Em seguida os candidatos deverão fornecer os seguintes dados:

- Nome;
- RG;
- CPF
- e-mail;

Feito isso, o candidato estará inscrito no projeto que previamente escolheu e logo será apresentado ao questionário definido pelo Administrador para este projeto. Então o candidato irá responder às questões e as respostas serão armazenadas no Banco de Dados para posterior consulta.

05 EMBASAMENTOS TEÓRICOS PARA A ELABORAÇÃO DO SOFTWARE

Este trabalho exigiu que seus proponentes adentrassem em um campo do conhecimento totalmente desconhecido a eles, visto que a área de Psicologia foge do escopo do curso de Ciências da Computação. Esse fato, além de se constituir um grande desafio, causou a nítida sensação de que se estava “pisando em território desconhecido.” Porém, com a ajuda dos orientadores e da literatura, tentou-se superar as dificuldades e enriquecer esta monografia com o conteúdo teórico do saber psicológico que é necessário ao trabalho.

O breve apanhado em torno do embasamento teórico usado neste trabalho se baseia essencialmente em [ALCHIERI; CRUZ, 2003], sendo que várias passagens destacadas na seção seguinte foram reproduzidas *Ipsis literis* do livro desses dois autores.

5.1 AVALIAÇÃO PSICOLÓGICA

De [ALCHIERI; CRUZ, 2003], sabe-se que, ao longo dos tempos, as diversas disciplinas criadas em torno do saber psicológico têm almejado erigir explicações sobre as finalidades e tendências das condutas humanas. Baseando-se nessa orientação geral, uma sub-área do conhecimento psicológico chamada “Avaliação Psicológica” se firmou historicamente como uma disciplina que tem por objetivo medir fenômenos ou processos psicológicos. A meta primordial da Avaliação Psicológica é “responder às demandas sociais e às possibilidades de avaliação e previsão da conduta dos indivíduos, a partir da leitura de suas condições de existência e de sua ação individual ou coletiva”.

De uma forma geral pode-se afirmar que avaliação psicológica se refere ao “modo de conhecer fenômenos e processos psicológicos por meio de procedimentos de diagnóstico e prognóstico”. Entende-se como “um corpo de conhecimentos sobre as medidas das características e das diferenças individuais”. Avaliação psicológica ainda refere-se aos procedimentos de exame propriamente ditos e pode ser entendida como “um processo teórico e tecnicamente organizado em torno de instrumentos de testagem para responder a

uma demanda específica de uma pessoa ou grupo, durante um tempo específico” [ALCHIERI; CRUZ, 2003].

A fim de atingir seus objetivos, a avaliação de fenômenos e processos psicológicos e a investigação de características dos padrões comportamentais humanos são respaldadas pelos pressupostos da medida. Segundo [ALCHIERI; CRUZ, 2003], medir “apresenta-se como um correlato de identificação e caracterização de um determinado fenômeno psicológico”. As aferições são realizadas via instrumentos, os quais visam “avaliar e quantificar comportamentos observáveis por meio de técnicas e metodologias específicas, embasadas cientificamente em construtos teóricos que norteiam a análise de seus resultados”.

No caso da investigação psicológica, estes instrumentos são denominados de testes psicológicos, os quais podem representar pela medida uma determinada ação que equivale a um comportamento. Dessa forma, indiretamente, os testes estão mensurando este aspecto comportamental.

Porém não basta apenas medir, já que se a medida for interpretada em si mesma, “os resultados de um procedimento qualquer poderão valer mais que os sentidos teóricos, subjacentes ao entendimento do fenômeno”. É necessária uma base teórica que “permita dar sentido aos fatos relacionando-os como os dados obtidos por meio da medida do fenômeno ou processo psicológico contextualizado”. Tal sentido possibilita “a interpretação dos resultados obtidos mediante uma sintaxe própria.” Logo, conclui-se que são as teorias explicativas do comportamento que “justificam e conseqüentemente revestem de sentido a necessidade da medida sobre este ou em outro fenômeno”.

5.2 PSICOMETRIA

O aperfeiçoamento da qualidade dos testes psicológicos é constantemente buscado pela Psicometria, que é uma especialidade da área da Psicologia que concentra as teorias e técnicas de mensuração do comportamento. Ela foi elaborada por psicólogos de preocupações predominantemente estatísticas e sua origem está relacionada com diversas disciplinas: estatística, Psicologia experimental, metodológica e computacional [ALCHIERI; CRUZ, 2003].

A evolução da Psicometria está vinculada aos trabalhos de Binet na França e de

Spearman na Inglaterra, bem como aos recentes esforços de Galton e seu laboratório antropométrico. Sua história remete à invenção dos testes psicológicos e à invenção dos primeiros questionários mais elaborados na Europa do século XIX, destacando-se o Stanford-Binet e os antigos testes de inteligência. O que a Psicometria se propõe a estudar, contudo, não são apenas aptidões, mas todo tipo de atributo cognitivo, como memória, aprendizagem, percepção etc. [FORDHAM UNIVERSITY, 2005].

Sabe-se ainda que a mensuração de comportamentos observáveis e registráveis apresenta-se como uma modalidade instrumental do profissional de Psicologia que pode ser usada para analisar estados ou características psicológicas dos objetos de estudo. Tais estados ou características podem ser inferidos a partir da observação, registro e mensuração de comportamentos.

5.3 ALIANDO A TEORIA PSICOLÓGICA AO SOFTWARE

Parte das avaliações psicométricas envolvem a participação do avaliando, seja por meio de atribuição de valores numéricos a enunciados ou por auto-relato. Ele precisa apresentar um certo comportamento na presença de um ou mais estímulos para que seja possível inferir a quantidade de uma propriedade psicológica [PASQUALI, 1956]. No caso do *software* Perfil, as perguntas com as quais o candidato irá se defrontar servirão de estímulo para que dele possa emergir o comportamento a ser mensurado.

A técnica psicométrica aplicada pelo software para mensurar o grau de discernimento dos selecionadores em relação às competências exigidas para um cargo denomina-se "Matriz de Diferencial Semântico". Ela foi desenvolvida por Charles Osgood, um psicólogo norte-americano que em 1957 inaugurou esse conceito o qual, por meio de quantificações, gera representações de conceitos em forma de gráficos. Busca-se gerar um julgamento conotativo (ao invés de denotativo) dos significados de palavras, usando escalas como bom-ruim, grande-pequeno, forte-fraco, ativo-passivo etc. [DERWING, B. L.; ALMEIDA, 2004].

Essa técnica é evocada quando da elaboração do perfil de competências exigido para a(s) vaga(s) da seleção em questão. Os usuários, os quais possivelmente serão membros de uma equipe que tenha a necessidade de definir que características esperam

do(s) novo(s) membro(s), irão se deparar com uma matriz de diferencial semântico a ser preenchida/valorada. Suas linhas e colunas são compostas pelos atributos que mais foram citados como essenciais ao perfil de competências que o cargo exige, os quais são extraídos do espaço amostral definido pelo administrador do sistema.

Bom c/P	2	4	4	2	2	
Iniciativa	3	2	5	3		2
Inteligente	5	3	4		3	2
Líder	4	4		4	5	4
Disciplinado	2		4	3	2	4
Crítico		2	4	5	3	2
	Crítico	Disciplinado	Líder	Inteligente	Iniciativa	Bom c/P

Média	15,3	13,3	19,9	15,2	15,6	14,7
Des.Padrão	4,9	6,7	3,7	5,2	5,0	4,9
Diferença	10,4	6,6	16,2	10	10,6	9,8

Figura 3: Exemplo de Matriz de Diferencial Semântico.

Tal matriz permite que, dois a dois, os atributos definidos como prioritários pelo grupo de usuários sejam confrontados (pela intersecção linha X coluna). Desse processo emergirá um número pertinente ao intervalo de 0 a 5, o qual é uma estimativa numérica que representa o grau de proximidade/similaridade entre os predicados, segundo o julgamento do usuário. Maior valor significa maior conexão entre eles, menor valor significa o oposto. Apesar de ser um número, esse valor na verdade é subjetivo, pois captura a estimativa do respondente sobre aquele assunto. Pode-se dizer então que essa técnica capta cientificamente o senso comum sobre um assunto, depura-o e representa-o como um conceito formal [HAYS, 1970].

Após o usuário preencher a metade inferior da matriz, ela é “espelhada”, ou seja, os valores abaixo da diagonal principal são replicados para as células acima da diagonal. Esse processo serve para realizar as atribuições faltantes. É algo como a propriedade comutativa da multiplicação, ou seja, CRITICO x DISCIPLINADO resulta em um valor, o qual é o mesmo para DISCIPLINADO x CRITICO. Após o “espelhamento” têm-se todas as

combinações. Em seguida é calculada a soma de cada coluna, a qual representa o valor total atribuído a um predicado. Então, após todos os usuários terem preenchido a matriz de diferencial semântico, é calculado a média e o desvio padrão de cada coluna (predicado).

A média é obtida pela soma do valor total de cada uma das colunas de mesmo nome da matriz de cada um dos usuários dividida pelo número de usuários (média simples). Já o desvio padrão (s) é calculado pela raiz quadrada da variância, a qual é obtida pela média das distâncias entre cada valor e a média.

Onde:

s = desvio padrão

n = número de dados;

X_i = dado i;

X = média;

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{(n - 1)}}$$

Figura 4: Fórmula para cálculo do Desvio Padrão.

O desvio padrão representa a dispersão estatística entre os dados, de forma que será tanto maior quanto mais variabilidade houver entre eles. Se ele for 0 (zero), indica que os dados são todos iguais.

Em seguida é calculada a diferença entre a média e o desvio padrão de cada coluna, obtendo-se assim um número quantificador, o qual serve para ordenar a lista dos atributos. Essa lista ordenada decrescentemente caracteriza uma pirâmide (Pirâmide de Osgood).

5.4 INTERPRETANDO AS SAÍDAS DO SOFTWARE

Quanto mais pontos recebe um atributo, mais conexões ele tem com os outros, portanto mais central ele é para a compreensão do todo. Por exemplo, para o conceito “chá”, supõe-se que o elemento “erva” seja o mais votado. Isso significa que o respondente acha que o fundamental em termos de chá é a erva, e que tudo mais (água quente, xícara, açúcar etc.) não é tão essencial quando alguém se refere ao conceito “chá” [REIS, 2005].

Os atributos, após passarem pelo processo de triagem e receberem a manipulação matemática, representam um mapa semântico das qualidades visadas para o conceito de um profissional. Denomina-se isso de mapa mental (*Mind Map*) de tal conceito [REIS, 2005].

A fim de facilitar a interpretação das Pirâmides de Osgood, mostra-se na Figura 5 o conceito bem formatado de um profissional. Vê-se na pirâmide que os atributos estão bem delimitados e numa escala hierárquica. Isto é, os componentes estão claros, cada um tem seu lugar:



Figura 5: Exemplo de Pirâmide de Osgood bem formatada.

Já no caso mostrado na Figura 6, conclui-se que o conceito está nebuloso, pois os atributos estão dispersos, sem hierarquia. Isso reflete certo grau de discordância em relação ao conceito do profissional. Se quer um líder ou um crítico?

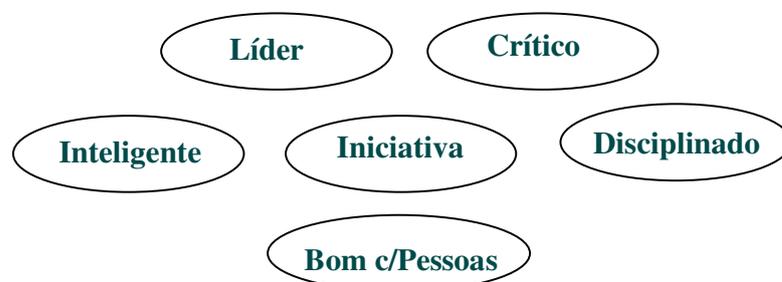


Figura 6: Exemplo de Pirâmide de Osgood mal formatada.

Para a interpretação e avaliação da Pirâmide de Osgood é **imprescindível a avaliação de um profissional competente**, o qual irá se basear nos ensinamentos que recebeu na Academia para melhor compreender a semântica que estes conceitos interligados estão refletindo. Só assim, após definida e compreendida a Pirâmide, pode-se dizer que o perfil de competências delineado pelos usuários está formado.

Assim sendo, o profissional que opera o *software* Perfil já têm bases sólidas para elaborar o questionário voltado para extrair dos candidatos essas competências elencadas como principais. De [ALCHIERI; CRUZ, 2003], sabe-se que “na preparação de um instrumento ou teste psicológico, quatro condições são necessárias para garantir a sua qualidade e possibilidade de uso seguro: a elaboração e análise de itens, estudos da validade, da precisão e de padronização.”

Porém, não é da competência dos proponentes desse trabalho abarcar e descrever a metodologia usada para a elaboração de instrumentos de testes. Tem-se plena consciência de que essa tarefa cabe aos profissionais da área da Psicologia, pois, ainda segundo [ALCHIERI; CRUZ, 2003], “do ponto de vista do conhecimento e da intervenção, podemos afirmar que a ciência psicológica e, especialmente as disciplinas associadas ao campo da avaliação psicológica, possuem um acúmulo de informações sobre métodos e técnicas de observação e medida, disponíveis e caracterizados na produção do conhecimento sistematizado, suficiente para dotar o psicólogo, de uma capacidade teórica e técnica para mensurar fenômenos e processos psicológicos”.

O fluxograma a seguir ampara todo o processo que foi previamente descrito e também descreve sucintamente parte do funcionamento do *software* Perfil. Chama-se a atenção para as etapas do processo que requerem a intervenção de um profissional da Psicologia: elas estão salientadas com o a letra grega psi (Ψ), a qual identifica essa ciência.

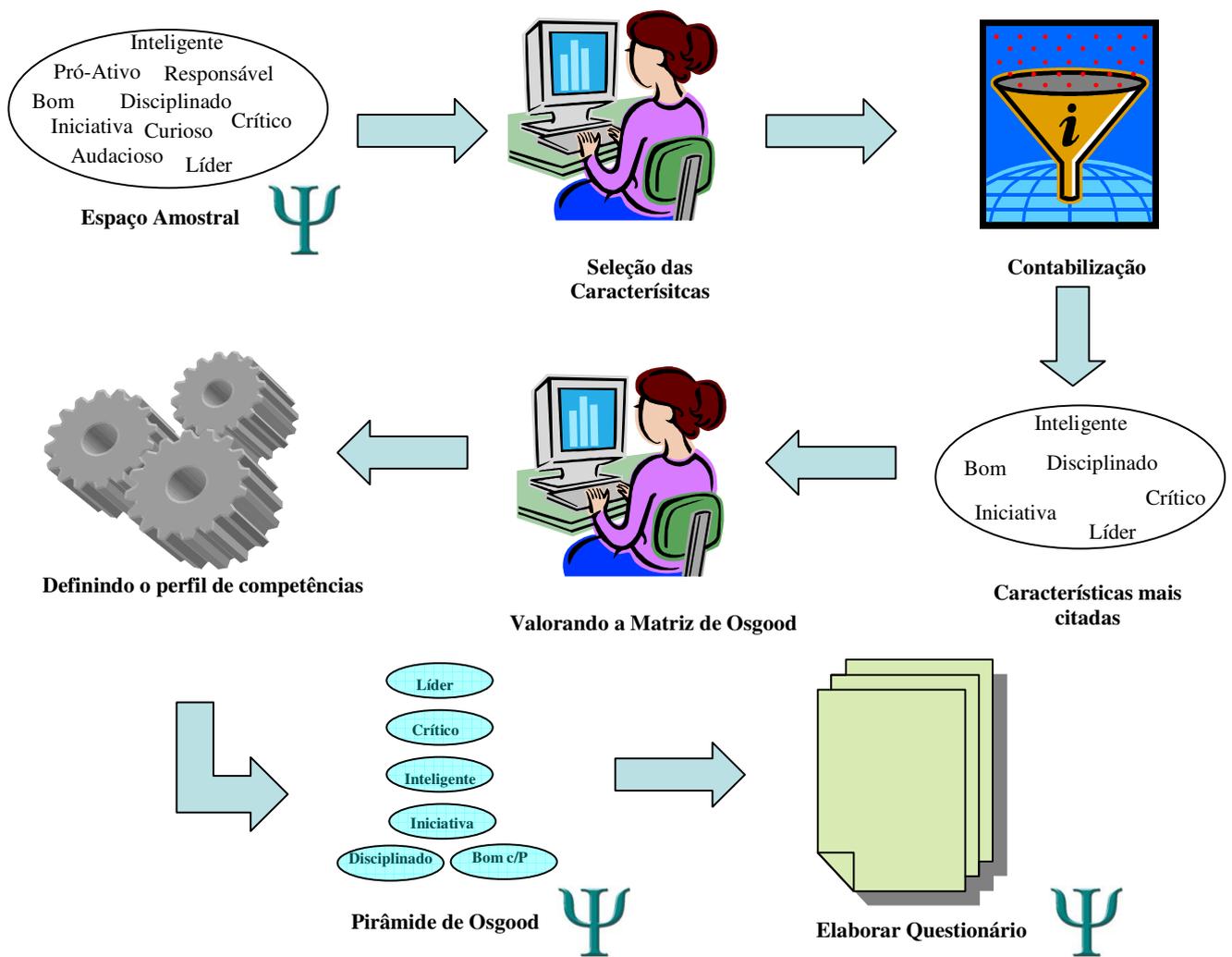


Figura 7: Fluxograma de parte do funcionamento do *software* Perfil.

06 INTELIGÊNCIA ARTIFICIAL

Este capítulo apresenta o ferramental científico relacionado as técnicas de Inteligência Artificial usadas neste trabalho. Não procurou-se relatar historicamente esse campo de estudo e nem definir formalmente todos os conceitos relacionados. Sabe-se que, para o escopo deste trabalho, não é interessante se alongar com discussões em torno de algumas definições que não facilitam a interpretação e o entendimento dos procedimentos realizados aqui, além de que, como está salientado em [BARRETO, 2001], “gostaria de poder apresentar uma definição convincente (sobre Inteligência Artificial). Infelizmente, tal não é o caso”, é difícil encontrar uma definição que agrade a todos quando se busca definir o que é Inteligência Artificial.

6.1 RETROSPECTO

A fascinante busca do ser humano visando entender a si mesmo anda em paralelo com a trajetória da vida nesse planeta. Há mais de dois milênios que o homem questiona-se e reflete sobre si mesmo, procurando encontrar as respostas sobre seu corpo, sua razão, seus sentimentos [BARRETO, 2001]. Dentre todos os aspectos que podem ser explorados da essência humana, com certeza um dos fatores mais complexos e curiosos é a inteligência.

Sabe-se que no decorrer dos tempos vários filósofos e cientistas se dedicaram a análise dos vários aspectos que constituem a inteligência humana e, embora o estudo sobre a inteligência tenha se iniciado dentro do campo de estudo da filosofia, o mesmo extrapolou os limites dessa área e a inteligência passou a ser estudada de forma científica por outros campos do saber humano, tais como engenharia, psicologia, pedagogia, ciência cognitiva, neurologia, lingüística, computação, entre outros. Hoje a inteligência também é alvo de estudos da ciência conhecida como Inteligência Artificial (IA) [ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, ?].

6.2 OBJETIVOS DA I.A.

Pode-se dizer que, de forma pragmática, a IA busca entender a cognição humana e tenta imitar/reproduzir/emular os seus processos em máquinas e computadores por meio da construção de programas, os quais seriam capazes de resolver problemas difíceis [RICH, 1988]. Para isso, essa ciência levanta questões do tipo: Como ocorre o pensar? Como o homem extrai conhecimentos do mundo? Como a memória, os sentidos e a linguagem ajudam no desenvolvimento da inteligência? Como surgem as idéias? Como a mente processa informações e tira conclusões decidindo por uma coisa ao invés de outra? Essas são algumas perguntas que a I.A. procura responder para simular o raciocínio humano e implementar aspectos da inteligência em máquinas, as quais podem adquirir a flexibilidade e a capacidade de resolver esses problemas difíceis, o que, segundo [BARRETO, 2001], significa que elas possuem características de um comportamento inteligente.

E porque é que se busca-se inspiração no modelo humano de desempenhar tarefas para se construir programas “inteligentes”? Ora, parafraseando [RICH, 1988], “as pessoas são os mais conhecidos realizadores da maioria das tarefas com que a I.A. lida, portanto será de bom senso procurar nelas pistas sobre como agir”. Em última análise, o que se busca é um bom modelo dos processos envolvidos no raciocínio inteligente. Com esse objetivo é que, juntos, psicólogos, lingüistas e cientistas da computação trabalham na área do conhecimento denominada de ciência cognitiva [RICH, 1988].

6.3 DESAFIOS DA I.A.

Como foi descrito anteriormente, as pesquisas na área de I.A. são voltadas para o estudo de métodos e técnicas úteis ao desenvolvimento de programas que simulam nas máquinas comportamentos inteligentes, isto é, tornam os computadores capazes de “pensar” e tomar decisões. Tal qual acontece com os processos cognitivos humanos, as técnicas de I.A. necessitam de uma grande quantidade de conhecimentos e de mecanismos de manipulação de símbolos, os quais exigem formas de serem representados, modificados e ampliados [RICH, 1988]. Além disso, sabe-se que a inteligência é algo extremamente complexo e entendê-la não é uma tarefa trivial. Ainda que existam muitas conclusões

relevantes, ainda há muito a ser desvendado, uma vez que não existe uma teoria completa sobre a mente humana e os processos de raciocínio.

Vários têm sido os esforços feitos no sentido de se simular os tipos de raciocínios utilizados pelo ser humano e implementá-los no computador por meio da I.A. Desses esforços é que originaram-se as diversas técnicas usadas pela I.A., as quais destacam-se os Sistemas Especialistas, a Lógica Fuzzy e, em especial para o foco deste trabalho, as Redes Neurais. Essas técnicas são amplamente utilizadas como meios auxiliares para expandir a capacidade de inteligência do homem e até mesmo como forma de substituí-lo em diversas funções [ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, ?].

6.4 ABORDAGENS DA INTELIGÊNCIA ARTIFICIAL

Baseado nos diversos campos de estudo que ajudaram na fundamentação dos princípios teóricos da inteligência artificial, sabe-se de [BARRETO, 2001] que a I.A. se dividiu em duas correntes:

Inteligência Artificial Simbólica - Também denominada de descendente ou simbolista, dá ênfase aos processos cognitivos, ou seja, a forma como o ser humano raciocina. Objetiva encontrar uma explicação para comportamentos inteligentes baseado-se em aspectos psicológicos e processos algorítmicos, sendo que o comportamento global é simulado, sem considerar os mecanismos responsáveis por este comportamento.

Inteligência Artificial Conexionista - Também denominada de biológica ou ascendente, dá ênfase no modelo de funcionamento do cérebro, dos neurônios e das conexões neurais. Apropria que construindo máquinas que imitem a estrutura do cérebro humano, estas apresentarão inteligência. Tal abordagem se sustenta no **Princípio da Réplica**, o qual é descrito em [BARRETO, 2001] da seguinte forma: “Se for construído um modelo suficientemente preciso do cérebro, este modelo apresentará um comportamento inteligente. Se apenas uma pequena parte do cérebro for reproduzida, a função exercida por esta parte emergirá do modelo”.

A abordagem conexionista engloba a técnica que foi usada neste trabalho. Mais a

frente discute-se os detalhes do porquê dessa escolha e a metodologia adotada para se usufruir dos benefícios da Inteligência Artificial Conexionista.

Não se vai entrar no mérito de discutir aqui qual abordagem é melhor, já que cada uma tem suas vantagens e desvantagens para determinados contextos. Apenas salienta-se que a Inteligência Artificial Conexionista é mais indicada para problemas mal definidos, onde falta conhecimento explícito para se realizar uma tarefa [BARRETO, 2001]. Alguns dos campos de aplicação dessa técnica são:

- reconhecimento de padrões (assinaturas, impressões digitais etc.);
- robótica;
- em sistemas especialistas;

6.5 REDES NEURAIS

O cérebro humano é composto por cerca de 100 bilhões de neurônios, os quais são conectados entre si por meio de sinapses. Durante uma sinapse, os neurônios trocam informações via neurotransmissores, os quais são descarregados pelo axônio de um neurônio e recebidos pelos dendritos do neurônio seguinte. As sinapses têm um papel fundamental na memorização das informações, principalmente as do córtex cerebral e as das partes mais profundas do cérebro. Pode-se imaginar que em cada sinapse, a quantidade de neurotransmissores que podem ser liberados para uma mesma frequência de pulsos do axônio representa a informação armazenada nesta sinapse [BARRETO, 2001].

Desse complexo emaranhado de conexões que interligam os neurônios é que se atribui a responsabilidade pela manifestação de características inteligentes nos seres vivos. Com o desenvolvimento da Inteligência Artificial surgiu a idéia de representar por meio de determinados programas o funcionamento do processo de aprendizagem do cérebro humano. A tentativa de simular a rede neural do cérebro deu origem à chamada Rede Neural Artificial (RNA) [BARRETO, 2001].

Diagrama de um neurônio

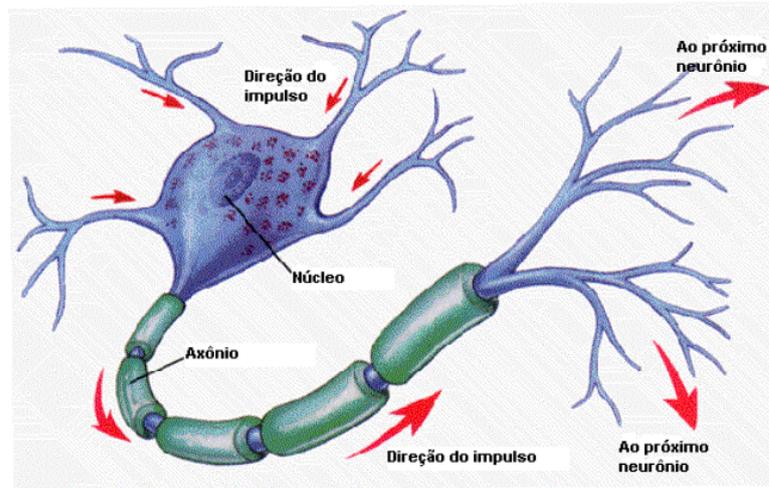


Figura 8: Neurônio Biológico.

6.5.1 BREVE HISTÓRICO

Os trabalhos sobre redes neurais se iniciaram na década de 40, na Universidade de Illinois, com o neurofisiologista McCulloch e o matemático Walter Pitts. Suas idéias foram publicadas no artigo *A Logical Calculus of the Ideas Immanent in Nervous Activity* (1943). McCulloch e Pitts estabeleceram uma analogia entre o processo de comunicação das células nervosas vivas e o processo de comunicação por transmissão elétrica e propuseram a criação de neurônios formais (artificiais), os quais podem ser entendidos como modelos matemáticos. Em 1947 eles conseguiram demonstrar que era possível conectar os neurônios formais e formar uma rede capaz de executar funções complexas [ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, ?].

6.5.2 MORFOLOGIA

Assim como o cérebro humano é composto de células biológicas, as redes neurais são compostas por neurônios artificiais. Eles são modelados de acordo com algumas características dos neurônios biológicos, pois ainda não é possível reproduzir totalmente o

sistema de processamento paralelo existente nas células biológicas. É válido ressaltar que não existe por parte da neurocomputação a pretensão de imitar o cérebro humano: o seu objetivo é o aprendizado da máquina, sendo que este é baseado em modelos que possam ser implementados para desempenhar funções inerentes ao cérebro humano. Em suma, o que se busca é uma inspiração biológica na natureza, porém as RNAs estão muito distantes das redes neurais naturais e, frequentemente, as semelhanças são mínimas [BARRETO, 2001]. Para aprofundamento, sugere-se a leitura de Barreto, já que na página 99 do seu livro aqui citado, ele discorre muito bem a respeito da relação entre os pesquisadores de Inteligência Artificial Conexionista e plausibilidade biológica de seus trabalhos.

As pesquisas em Redes Neurais Artificiais tentam simular o cérebro humano, focando principalmente na sua capacidade de aprender e se adaptar a eventuais mudanças, sendo que o principal objetivo é simular a capacidade de aprendizado e a capacidade de generalização do cérebro humano. Munidas dessas características, as RNAs podem executar tarefas que os programas convencionais não conseguiam realizar, pois não tinham essa capacidade de aprendizagem e adaptabilidade [ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, ?].

O psicólogo Donald Hebb (1949) deu uma importante contribuição ao estudo das redes neurais ao elaborar uma teoria baseada no processo de aprendizagem que ocorre no cérebro humano. Pode-se imaginar que, seguindo um princípio válido da biologia, o uso de um órgão favorece o seu desenvolvimento. Logo, cada vez que uma sinapse é ativada ou consegue ativar outro neurônio, o número de neurotransmissores liberados aumenta na próxima vez que o neurônio for ativado. Isto representa um aumento da conexão entre os dois neurônios e recebe o nome de **facilitação**. O mecanismo de facilitação inspirou a conhecida lei de Hebb: “A intensidade de uma conexão sináptica entre dois neurônios aumenta quando os dois neurônios estão excitados simultaneamente.” [BARRETO, 2001]. Esta teoria serviu de base para a aprendizagem das redes neurais.

Agora, transpondo o modelo neural biológico para a óptica da I.A., é correto afirmar que as redes neurais artificiais podem ser entendidas como conjuntos bem estruturados de unidades de processamentos (neurônios artificiais), interligadas por canais de comunicação (sinapses), cada qual tendo um determinado peso (neurotransmissores) correspondente a um valor numérico. Tudo isso organizado estruturalmente em forma de camadas (entrada,

intermediárias e saída), as quais são interconectadas por meio de conexões entre as camadas.

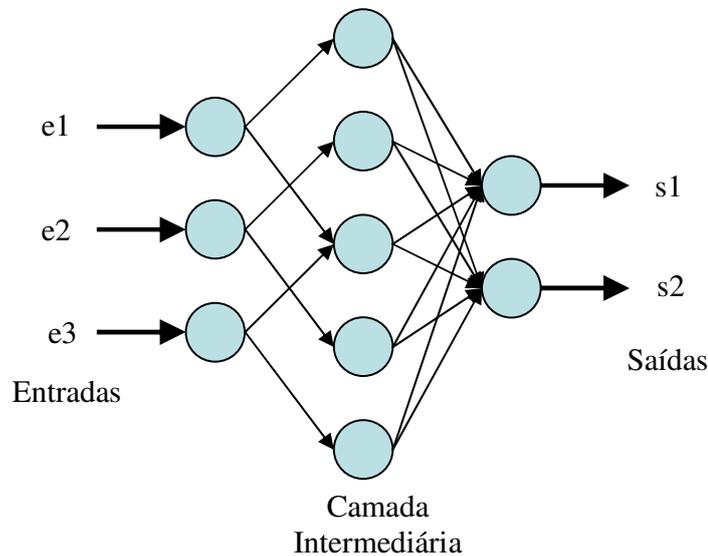
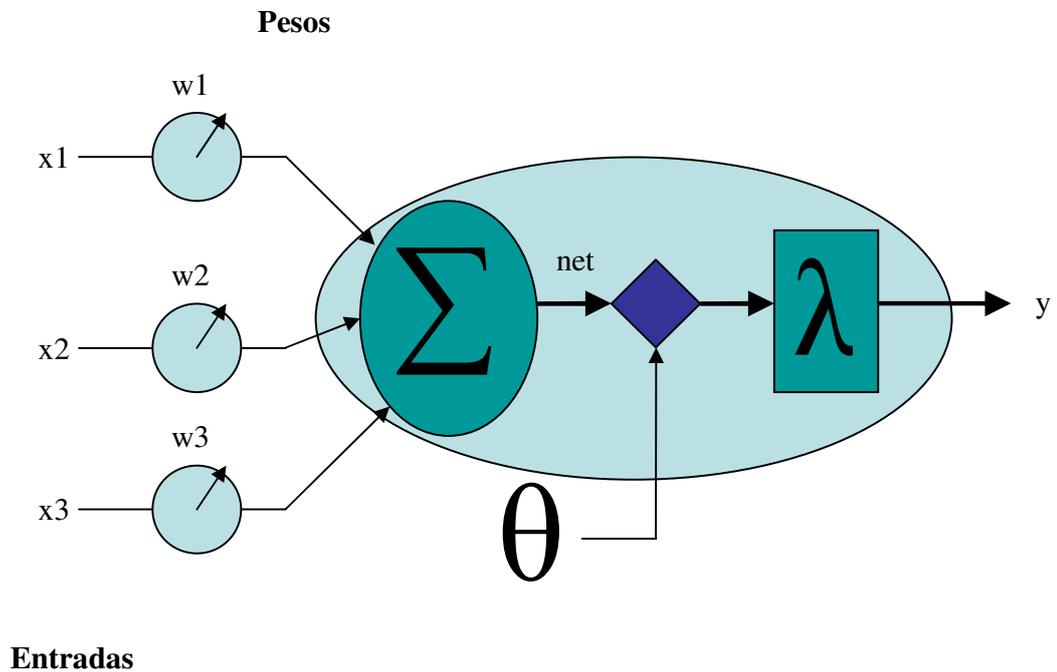


Figura 9: Exemplo de RNA com as conexões destacadas (adaptado de [BARRETO, 2001]).

Cada neurônio artificial recebe uma ou mais conexões, as quais podem ser oriundas de entradas externas (correspondendo aos neurônios dos sentidos) ou de saídas de outros neurônios. Tais conexões são moderadas por valores numéricos chamados de pesos. Os pesos podem ser considerados como um valor que representa a força de conectividade sináptica entre os neurônios. O somatório de todas estas entradas multiplicadas pelos pesos respectivos de cada conexão dá origem ao chamado “net” de um neurônio. Após a determinação do “net”, o valor de saída do neurônio é produzido por meio de uma função de saída λ , a qual comumente é uma função do tipo Linear, Sigmoidal ou Logística ou ainda Tangente Hiperbólica. Este valor deverá ser maior que um limiar definido (θ), o chamado “threshold”, para que o neurônio dispare uma sinapse e venha a excitar a conexão que sai dele.



$$NET = \sum_{i=1}^n (x_i * w_i)$$

n = número de entradas da rede;

x_i = é a i-ésima entrada do padrão apresentado à rede

w_i = peso do neurônio (referente à entrada).

Figura 10: Neurônio Artificial (adaptado de [BARRETO, 2001]).

6.5.3 APRENDIZADO

Essa contextualização previamente descrita, a qual foi baseada em [BARRETO, 2001] e [ROISEMBERG, 2005], fornece bases para se entender o aprendizado das RNAs, o qual pode ser visto como um processo iterativo de adaptação aplicado aos parâmetros da rede (pesos e *thresholds*), onde os conhecimentos são armazenados na intensidade de suas conexões sinápticas após cada interação. Mudando-se os valores desses parâmetros de conexões muda-se o comportamento da rede. Estas mudanças de comportamento

representam o aprendizado da rede [BARRETO, 2001].

Pode-se dizer que uma RNA aprende por meio de analogias e exemplos, visto que durante o processo de aprendizado, em cada iteração é apresentado a ela um conjunto de dados. Então ela vai ajustando os pesos das conexões entre os neurônios a fim de captar as características do conjunto de dados apresentados. Após a finalização do processo de aprendizagem, se este foi bem sucedido, a RNA possui a capacidade de generalização, ou seja, responderá satisfatoriamente mesmo a estímulos de entrada não vistos durante o processo de treinamento.

6.5.4 CARACTERÍSTICAS

Este tópico originou-se da necessidade de explicar alguns conceitos em torno das redes neurais. Sabe-se que este não é o propósito deste trabalho, porém um certo embasamento aqui será muito benéfico para o entendimento dos tópicos que se seguem.

A literatura especializada mostra que o desempenho de uma Rede Neural é afetado diretamente por diversos fatores, dentre eles destacam-se:

- **A Arquitetura:** é definida pelo tipo de rede, número de camadas, número de neurônios por camada etc. Existem diversos tipos de redes e cada uma implementa uma forma diferente de realizar a interconexão dos neurônios;
- **A Capacidade de aprendizado:** é afetada pelo tipo de aprendizado, pelo algoritmo usado no aprendizado, pelo número de interações ao qual a rede é submetida durante esse processo, o número de exemplos que é apresentado em cada interação etc..

Pode-se entender como algoritmo de aprendizado um conjunto de regras bem definidas que norteiam o processo de aprendizado. Os algoritmos de aprendizado podem ser agrupados em quatro categorias:

- Aprendizado por correção de erro;

- Hebbiano;
- Competitivo;
- De Boltzmann;

A título de curiosidade, é útil ressaltar que no escopo deste trabalho será explorado mais a frente o aprendizado competitivo.

Quanto ao tipo de treinamento para aprendizado as redes apresentam:

- **Aprendizado Supervisionado:** Exige a presença de um agente externo (tutor) na fase de aprendizado. O tutor adquire conhecimento da rede, na forma de mapeamentos (padrões) de entrada-saída, sempre monitorando as diferenças entre o valor esperado (entrada) e o valor efetivo (saída). Ele avalia e informa à rede sobre a sua performance, indicando explicitamente um comportamento bom ou ruim. A partir disso, as devidas modificações nos pesos são efetivadas.
- **Aprendizado não Supervisionado:** Não possui um tutor (crítico). A rede é autônoma, trabalha com os dados que lhes são apresentados e apreende a refletir sobre as suas propriedades a partir de suas saídas. Os valores das conexões sinápticas não são modificados baseando-se se a resposta da rede foi correta ou não. Usa-se um mecanismo que, simplificarmente pode ser entendido assim: para exemplos semelhantes, a rede responde de modo semelhante. Esse tipo de aprendizado pode ser utilizado com um algoritmo competitivo ou hebbiano.
- **Aprendizado por Reforço:** Possui um crítico externo que avalia as respostas fornecidas pela rede e direciona o ajuste dos pesos. O aprendizado é *online*, feito por um processo de tentativas e erros, que visa maximizar um dado índice de desempenho, denominado de sinal de reforço.

6.5.5 AVALIAÇÃO

Pode-se destacar como pontos positivos a favor do uso de Redes Neurais na resolução de problemas o fato delas propiciarem:

- Inferência de múltiplas variáveis;
- Grande tolerância a falhas;
- Modelagem direta do problema;
- Paralelismo inerente;

Porém, o fator limitante dessa técnica da Inteligência Artificial é que a Redes Neurais Artificiais trabalham com um número reduzido (centenas) de neurônios artificiais, enquanto as redes neurais biológicas trabalham com milhões de neurônios. Esse precipício quantitativo ainda não pode ser contornado, pois as ferramentas computacionais não possuem capacidade de processamento e armazenamento suficientes para modelarem uma rede semelhante a cérebro e obterem respostas em tempo hábil.

A fim de provar a valia desses instrumentos da Inteligência Artificial, exemplifica-se abaixo uma lista de aplicações de Redes Neurais [ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, ?]:

- Análise de assinaturas
- Análise de características demográficas para marketing
- Análise do grau de satisfação de um cliente
- Controle de processos industriais
- Detecção de cartões de crédito falsos
- Monitoramento para manutenção de motores
- Previsão da bolsa de valores e cotação de moedas
- Previsão do mercado financeiro

- Reconhecimento de caracteres e impressões digitais
- Reconhecimento ótico de caracteres

6.6-DEFINIÇÃO DO MODELO DE RNA USADA

Atualmente existem dezenas de modelos de redes neurais estruturados para as mais diversas aplicações. No escopo deste trabalho, julgou-se mais adequado ao problema o modelo de Rede Neural denominado **Mapas Auto-Organizáveis** ou *Self Organizing Maps* (SOM). A escolha deve-se a capacidade da SOM se auto-organizar (caracterizando um aprendizado do tipo não supervisionado) e manter uma ordenação topológica dos padrões treinados [CARVALHO;MARTINS, ?]. Essas características permitem a rede detectar similaridades entre vários sinais, agrupando-os em classes de similaridades, sendo, desta forma, um passo para a determinação de padrões de uma coleção de sinais [BARRETO, 2001]. No contexto deste trabalho espera-se que, a partir das respostas dadas pelos candidatos ao questionário elaborado, a rede venha a classificar os respondentes baseado-se nas características das informações coletadas.

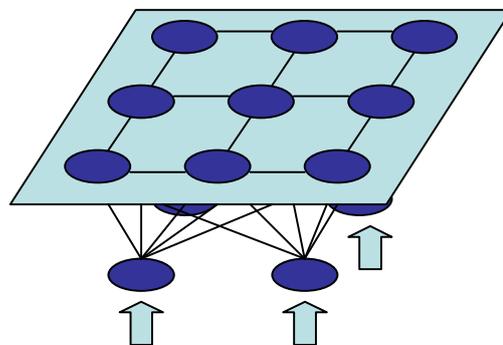
É pertinente citar aqui que este modelo de Rede Neural foi desenvolvida por Teuvo Kohonen na década de oitenta e é biologicamente sustentado por mapas topológicos presentes no córtex cerebral, o qual possui centros especializados em diversas áreas, tais como fala, visão, audição, sensorial, motora etc. Estes centros estão localizados em regiões bem definidas e próximas umas das outras [ROISEMBERG, 2005].

Na rede SOM, neurônios próximos (na disposição do mapa) devem responder por classes de vetores de entrada que também estejam próximos uns dos outros (sejam parecidos). Dito de outra forma: para estímulos semelhantes, as mesmas áreas topológicas do mapa devem responder, já que as áreas individuais apresentam um ordenamento lógico consistente com a sua função [ROISEMBERG, 2005]. Essa característica assimila-se ao funcionamento do cérebro dos animais evoluídos, pois determinadas áreas respondem por estímulos de mesma natureza: fala, audição, visão etc. Para exemplificar, cita-se o mapa Tonotópico da Audição: para sons de mesma frequência, neurônios “vizinhos” respondem

de maneira similar em uma seqüência ordenada das frequências mais altas para as mais baixas [ROISEMBERG, 2005].

Os Mapas Auto-Organizáveis são constituídos basicamente por duas camadas: Camada de Entrada e Camada Competitiva. Na camada de entrada cada sinal do mundo externo é percebido por um neurônio sensorial determinado e este liga-se a todos os demais neurônios da Camada Competitiva. Essa última camada normalmente está organizada como uma grade bidimensional, sendo que cada neurônio está conectado a todos os neurônios de sua vizinhança [ROISEMBERG, 2005].

Camada Competitiva



Camada de Entrada

Figura 11: Topografia de uma Rede Auto-Organizável (Mapas Auto-Organizáveis).

6.6.1 TREINAMENTO DA SOM

O treinamento de uma rede SOM é baseado no algoritmo de aprendizagem por competição. Os neurônios da camada de saída competem entre si para definir o mais semelhante (neurônio vencedor) ao padrão de entrada apresentado. Os neurônios são localmente interconectados por uma relação de vizinhança, determinando a topologia do mapa (linear em anel, hexagonal e retangular) [CARVALHO;MARTINS, ?].

Durante o processo de treinamento, a realimentação positiva (ajuste dos pesos do neurônio vencedor) atinge não só o próprio neurônio, como também a uma vizinhança finita em torno dele. Todos os neurônios que recebem sinais excitatórios do neurônio vencedor também têm seus pesos modificados, participando assim do processo de “aprendizado”. Estes sinais excitatórios são propagados através das conexões laterais.

Generalizando-se a visão sobre o processo de aprendizado pode-se dizer que para cada tipo de entrada se estabelecerá uma configuração de neurônios excitados. Entradas semelhantes irão gerar configurações semelhantes, as quais permitirão a classificação das entradas [BARRETO, 2001].

A determinação do neurônio vencedor em uma rede SOM pode ser feita utilizando diversas métricas, porém, o procedimento mais comum é verificar qual neurônio possui a menor distância Manhattan (ou distância de Hamming) em relação ao padrão apresentado (vetor de entrada) [CARVALHO; MARTINS, ?].

Essa distância pode ser calculada da seguinte forma:

$$\text{Distância} = \sum_{i=1}^n \| X_i - W_{ji} \|$$

n = número de entradas da rede;

X_i = é a i-ésima entrada do padrão apresentado à rede
(elemento i do vetor);

W_{ji} = peso do neurônio (referente à entrada).

A grande valia destas redes é tentar projetar em uma superfície bidimensional características de vetores multidimensionais, já que é mais fácil visualizar neurônios que estão próximos uns dos outros em uma superfície bidimensional do que determinar que classes de vetores estão próximas entre si em um espaço multidimensional. Esta redução da magnitude dimensional preservando a ordem natural dos vetores de entrada permite que se visualizem relações importantes entre os dados, que, de outro modo, poderiam passar despercebidas [ROISEMBERG, 2005].

Diante do exposto, fica evidente que este modelo de rede é adequado ao problema de classificação que este trabalho se propõe a elucidar. Os autores das literaturas aqui citadas dão bases consistentes para reiterar a escolha por esse modelo de rede neural e, sendo assim, segue-se o trabalho, porém, agora deixando claro que já se tem um ferramental de trabalho definido.

07 METODOLOGIA ADOTADA PARA O USO DA RNA

Atualmente existem diversas ferramentas computacionais que dão suporte a todo o processo inerente a criação, treinamento e testes de redes neurais. Em prol da facilidade de uso e por ser conhecido pelos acadêmicos proponentes deste trabalho, bem como pelo professor orientador, o *software* MatLab foi usado para dar suporte a todas as atividades ligadas à redes neurais.

Depois de definida a ferramenta, seguiu-se a seguinte metodologia até se fazer o uso das Redes Neurais Auto-Organizáveis neste trabalho:

- Coleta de dados
- Formatação dos dados
- Configuração da rede
- Treinamento/Teste

7.1-COLETA DE DADOS

A fim de concretizar e por em prática os conhecimentos adquiridos no decorrer da elaboração desse trabalho, nada melhor que expor o *software* Perfil ao mundo real. Esse contato foi útil para testar as suas funcionalidades e corrigir alguns *bugs*, bem como serviu para a coleta dos dados que foram usados pela rede neural.

7.1.1- PERFIL DE CARACTERÍSTICAS

Primeiramente, usando-se o *software* Perfil, foram coletados dados relativos a uma seleção hipotética para o Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de Santa Catarina (PET-CCO). Esta etapa visou a elaboração do perfil de características a ser buscado em um “petiano” (bolsista do Programa PET).

Para isso, juntamente com auxílio do co-orientador Alessandro Vieira dos Reis elaborou-se um espaço amostral de adjetivos, sendo que os 10 membros do grupo PET-CCO, os quais foram cadastrados no sistema, tiveram a oportunidade de, individualmente,

selecionar dentre esse espaço amostral 6 características que julgaram indispensáveis a um futuro membro do grupo. O conjunto universo foi composto pelos seguintes predicados:

- independente
- líder
- auto-gerenciável
- obediente
- cooperador
- técnico
- metódico
- falante
- ouvinte
- espontâneo
- inteligente
- responsável
- honesto
- dedicado
- pró-ativo
- altruísta
- amigo
- bom trato com pessoas
- participativo

Em seguida, valendo-se das facilidades implementadas pelo *software* Perfil, contabilizou-se os 6 adjetivos que mais foram citados como essenciais para o perfil de um petiano. Em ordem decrescente, obteve-se a seguinte listagem:

- participativo
- cooperador
- responsável
- inteligente
- dedicado
- bom trato com pessoas

O próximo passo foi a geração da Matriz Semântica, a qual foi composta pelos adjetivos da listagem mostrada previamente. Ela ficou assim definida:

bom c/p						#
dedicado					#	
inteligente				#		
responsável			#			
cooperador		#				
participativo	#					
	participativo	cooperador	responsável	inteligente	dedicado	bom c/p

Figura 12: Matriz Semântica oriunda da Seleção Hipotética.

A partir do processo de valoração da Matriz Semântica, foi depurado o perfil ideal a ser buscado em um candidato que queira fazer parte do grupo PET-CCO. Mostra-se abaixo as saídas obtidas pelo *software* Perfil nesta fase do processo da seleção hipotética:

Adjetivo	Média	Desvio Padrão	Diferença
dedicado	17.0	4.843	12.157
responsável	16.2727	4.3712	11.901499
cooperador	17.2727	5.412	11.860699
participativo	15.2727	4.2015	11.0712
bom trato com pessoas	12.0	5.3936	6.6064
inteligente	10.4545	7.0112	3.4433002

Figura 13: Dados gerados durante a elaboração do Perfil de características para a Seleção Hipotética.

Formatando esses dados, obtemos a seguinte Pirâmide de Osgood, a qual representa graficamente o conceito do petiano ideal:



Figura 14: Pirâmide de Osgood da Seleção Hipotética.

Interpretando-se a pirâmide gerada, pode-se concluir que os membros do PET-CCO definiram como competência essencial para um futuro membro do grupo a **dedicação**. Este também deverá ser **responsável** e **cooperador**, bem como **participar** das atividades, sempre **tratando bem as pessoas** envolvidas. Tudo isso usando sua inteligência. Percebe-se claramente que o grupo sabe quais são as características que deseja encontrar nos candidatos de uma seleção real.

7.1.2-QUESTIONÁRIO

Novamente com auxílio do co-orientador Alessandro Vieira dos Reis, foi elaborado um questionário composto por 17 perguntas baseadas nas características que foram depuradas quando da definição do perfil de competências elencadas como essências para os membros do grupo PET-CCO.

Enfatiza-se aqui a ressalva de que este instrumento de teste leva em conta somente a percepção do PET-CCO. O mesmo **não foi elaborado sob os preceitos e critérios que prega a Avaliação Psicológica**, já que nessa seleção hipotética não há interesse em medir de fato as aptidões e características dos respondentes, tampouco ver quem é o melhor. Essa seleção fictícia é somente um pretexto para se colher os dados e se realizar o confronto entre a rede neural artificial e o programa, além de provar a utilidade do *software* Perfil no

processo de elaboração de questionários funcionais. Dito de forma pragmática: “não se buscou criar questionários, mas a ferramenta de fazer questionários”.

Esclarecido tudo isso, mostra-se logo a frente o questionário, juntamente com a pontuação estimada a cada resposta. Em negrito estão as respostas esperadas de um candidato considerado “ideal” ao perfil delineado pelo PET-CCO.

1) Você cursou Ensino Médio Técnico?

(**10**)**Sim** (0)**Não**

2) Você desempenha outras atividades além de estudar?

(**10**)**Sim** (0)**Não**

3) Você pratica alguma atividade física?

(**10**)**Sim** (0)**Não**

4) Tem por hábito se dedicar à leitura de fontes que lhe tragam conhecimentos diversificados (literatura, política, ciência, etc.)?

(**10**)**Sim** (0)**Não**

5) Você tem experiência profissional?

(**10**)**Sim** (0)**Não**

6) Você deixou de realizar alguma atividade depois que entrou para a Universidade?

(0)**Sim** (**10**)**Não**

7) Dentre as qualidades abaixo, assinale as 6 que você julga que mais se aproximam das suas:¹

(**8**) **participativo**

(**9**) **cooperador**

¹ Esta questão foi desconsiderada para o uso na Rede Neural.

(9) responsável	(1) amigo
(6) inteligente	(3) pró-ativo
(10) dedicado	(3) técnico
(7) bom trato c/ pessoas.	(1) metódico
(2) honesto	(0) falante
(0) obediente	(0) ouvinte
(7) sociável	(1) espontâneo
(5) auto-gerenciável	(3) independente
	(4) líder
(0) altruísta	

8) Dentre os defeitos abaixo, assinale os 3 que você julga que mais se aproximam dos seus:²

(-7) prepotente	(-9) competidor ao extremo
(-10) individualista	(-2) inseguro
(-7) egoísta	(-10) temperamento explosivo
(-4) perfeccionista ao extremo	(0) nenhum desses
(-7) chato	(-5) mimado
(-10) preguiçoso	(-3) tímido ao extremo
(-9) desorganizado	(-7) revoltado
(-9) irresponsável	(-5) desobediente
(-7) anti-social	(-3) instável
(-7) teimoso	(-4) confuso
(-8) nervoso	(-2) desarticulado
(-10) invejoso	(-3) desastrado

9) O seu nível de conhecimento em Língua Inglesa, numa escala de 0-10, em cada item é:

Compreende (**10**)

² Esta questão foi desconsiderada para o uso na Rede Neural.

Fala (10)

Lê (10)

Escreve (10)

10) Dimensione numa escala de 0-10 o seu conhecimento acerca de:

Linguagens de Programação (Java, C++, Pascal, Smalltalk etc.): (10)

Ferramenta para WEB (Editores, Linguagens etc.): (10)

Hardware (montar, comprar, manutenção, funcionamento etc.): (10)

Sistemas Operacionais (instalar, usar, configurar, funcionamento etc.):(10)

11) O PET realiza inúmeras atividades que podem envolver várias pessoas. A respeito disso penso que:

(10) São boas oportunidades para cooperar com os demais.

(-5) Não me sinto muito a vontade trabalhando com outras pessoas.

(0) Meu rendimento é maior quando trabalho sozinho.

(10) Prefiro trabalhar em equipe, pois podemos compartilhar idéias e ações.

(-10) Atividades em grupo não funcionam, perde-se muito tempo só para organizar as pessoas.

(0) Sou flexível e me adapto a qualquer situação

(-3) Nunca tive a oportunidade de trabalhar em grupo

12) Caracterizo o PET-CCO como (uma alternativa):

(5) Um grupo de bolsistas que desempenha atividades semelhantes às desenvolvidas num laboratório.

(-3) Um grupo de bolsistas que produzem *software* para empresas.

(-10) Alunos que se destacam na graduação e foram contemplados com uma bolsa de auxílio, além de uma invejável infra-estrutura para poderem desenvolver suas atividades.

(-5) Um grupo de bolsistas que realiza atividades de diversos tipos, cujo objetivo é a formação de um profissional com ampla visão do mercado de trabalho.

(10) Alunos que buscam uma formação ampla, e nesse processo desenvolvem pesquisas, promovem eventos e cursos. Também visam a melhoria nas atividades relacionadas ao curso (Ensino) .

(0) Não tive a oportunidade de conhecer o trabalho do PET-CCO.

13) O que me motivou a escolher o curso que frequentei foi (uma alternativa):

(10) A aptidão natural que tenho a assuntos ligados à área.

(5) Influência de familiares, parentes, amigos.

(5) A possibilidade de no futuro me tornar um empresário e ganhar muito dinheiro.

(10) A possibilidade de desenvolver pesquisas e contribuir para a ciência.

(-10) O baixo número de candidatos por vaga.

(-10) Ainda não sei o motivo da minha escolha.

(3) Adoro mexer com computadores e me dou muito bem com eles.

14) Quanto às atividades que desenvolvo em minha vida, eu (uma alternativa):

(10) Planejo as atividades que faço.

(-10) Vivo um dia de cada vez.

(0) Algumas coisas eu planejo, outras não.

(8) Preciso agendar rigorosamente tudo o que tenho que fazer.

(-10) Tenho noção dos meus compromissos, porém não gosto de ficar planejando o que vou fazer.

15) Quanto à realização de tarefas (uma alternativa):

(-5) Prefiro receber ordens sobre o que tenho que fazer

(5) Consigo perceber as atividades que me cabem e as delego

(10) Me julgo capacitado a organizar e a delegar aos outros as atividades.

(-10) Prefiro me isentar de responsabilidades que não me cabem.

(10) Minha colaboração parte tanto da atividade de realizar as tarefas bem como da de coordenar e organizá-las.

16) Encaro as atividades cotidianas de uma forma:

(10) Estrategista, pois sempre busco metas e objetivos.

(-5) Nunca pensei a respeito disso.

(10) Pontual, pois horários são feitos para serem cumpridos.

(-10) Informal, cumpro o que eu conseguir.

(10) Metódica, pois (quase) tudo pode ser avaliado e pensado.

(10) Disciplinada, assim torna-se mais fácil gerencia-las.

(-10) Desleixada, pois sempre dou um jeito mesmo.

(-5) Descompromissada, deixo que elas guiem minha rotina.

17) Avaliando o meu desempenho frente ao meu curso, considero que (uma alternativa):

(10) Estou se saindo muito bem e já estou familiarizado com os temas.

(5) Estou enfrentando algumas dificuldades, pois acho que o curso não é bem “minha praia”.

(2) Posso melhorar muito, pois no momento estou só curtindo a vida acadêmica.

(0) Prefiro não opinar sobre isso.

(10) Estou me dedicando ao máximo.

(10) Não me limito às deficiências do curso, pois tenho total liberdade para ir muito além desses limites.

(-10) É maravilhoso, não tenho nenhuma queixa.

Após todas essas etapas, finalmente, é que os supostos candidatos a petianos, os quais foram contatados via e-mail, puderam responder ao questionário. O conjunto dos “candidatos” foi composto principalmente por alunos do curso de Ciências da Computação da Universidade Federal de Santa Catarina. Além desses respondentes, outras pessoas

também ajudaram a formar esse conjunto, já que o *software* Perfil está hospedado em domínio público e qualquer pessoa na Internet tem acesso a ele.

7.2-FORMATAÇÃO DOS DADOS

De todos os “candidatos” contatados via e-mail para responderem ao questionário, 30 o responderam, os quais, de agora em diante, serão chamados de candidatos respondentes.

Tão logo que findaram-se os testes com *software* Perfil e o questionário implementado por ele, julgou-se necessário **desconsiderar** duas das questões do questionário de teste: a de número 7 e a de número 8. Isso se deve ao fato de que cada alternativa dessas questões iria exigir um neurônio a mais na entrada da rede neural, logo seriam necessários 20 neurônios só para a questão 7. Já a questão 8 exigiria 24 neurônios, sendo que no total 78 neurônios deveria ter a entrada da rede para mapear o questionário inteiro. Esse super-dimensionamento da entrada iria impingir a rede um tempo de treinamento muito grande, logo os testes seriam demorados. Deu-se ciência ao professor orientador a respeito desse procedimento, bem como ressalta-se que as devidas atualizações e modificações na classificação dos candidatos também foram feitas, de modo que não houve prejuízo nenhum ao trabalho.

Voltando-se às tarefas realizadas, é lícito registrar que aos dados dos respondentes foram adicionadas as respostas de um candidato fictício dito “ideal”, o qual respondeu às perguntas da melhor forma que podia ser esperado pelo selecionador. Este, a partir do somatório de todas as alternativas respondidas, atingiu a marca de **250** pontos. Da mesma forma, porém no sentido oposto, criou-se um candidato o mais distante possível do “ideal”, o qual obteve **-90** pontos. Assim delineou-se dois candidatos de perfis extremos, os quais, juntamente com os respondentes, foram analisados pela rede.

Para a Rede Neural, o que importa do questionário são os valores associados a cada resposta e não a semântica envolvida nas questões. Dessa forma, as respostas do questionário de teste devem ser formatadas para se adequarem a entrada da rede. Isso exige que elas sejam vetorizadas, ou seja, o valor correspondente a cada alternativa será um elemento de um vetor, o qual corresponde a todas as respostas de um candidato. Para

ilustrar esse processo, mostra-se abaixo a forma que as respostas do candidato “ideal” e do “não ideal” tomaram após a vetorização, bem como o escore alcançado por cada um deles no questionário, segundo os critérios fornecido ao *software*:

Ideal = [10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 0 0 10 0 0 0 10 10 10 10 10 0 10 0
10 10 0 0 10]
 $\Sigma = 250$ pontos.

NãoIdeal = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -5 0 0 -10 0 0 -10 -10 -10 -10 0 -5 0 -10 0 0 -10 0 -
10]
 $\Sigma = -90$ pontos.

Depois de determinados todos os vetores do conjunto de dados, eles foram pré-processados por meio de normalizações. Isso os tornou mais apropriados à utilização pela rede, já que os valores reais utilizados nas respostas do questionário ocasionariam uma grande demora durante o processo de treinamento. Foi utilizado o seguinte critério para se efetuar a normalização:

- Todos os dados de entrada da rede foram ajustados e redimensionados para o serem compreendidos no intervalo de -1 a 1.

Ressalta-se que a normalização mantém a coerência escalar das respostas.

7.3-CONFIGURAÇÃO DA REDE

O terceiro passo foi a configuração da rede. Ele pode ser melhor explicado se dividido em três etapas:

Determinação da topologia da rede a ser utilizada:

- Rede Auto-Organizável;
- 34 neurônios na primeira camada;

- $10 \times 10 = 100$ neurônios na segunda camada;
- Topologia: Função Camada Hexagonal (hextop);

Determinação de parâmetros do algoritmo de treinamento e funções de ativação.

- Número máximo de épocas permitido: 5000;
- Frequência de atualização do gráfico: 100 épocas;
- Função de aprendizagem (atualização dos pesos): learnsom;
- Função de transferência: compet;

Inicialização dos pesos das ligações entre neurônios

- Valor médio compreendido entre o valor máximo e o valor mínimo encontrado em cada vetor de entrada (*midpoint*).

7.4-TREINAMENTO/TESTE DA REDE

O quarto passo foi o treinamento e teste da rede. Nesta fase, seguindo a filosofia do algoritmo de treinamento competitivo, foram apresentados aleatoriamente a rede os vetores correspondentes às respostas dos candidatos. Devido a aleatoriedade desse processo, é inviável reproduzir o treinamento de uma Rede Neural Artificial, ou seja, a cada novo treinamento tem-se outra rede. Isso inviabiliza a reprodução e obtenção das mesmas respostas, mesmo que seja para o mesmo conjunto de dados. Durante o treinamento, a rede periodicamente ajustou os pesos das conexões dos seus neurônios em direção aos valores dos vetores de entrada.

Quanto ao tempo de treinamento, o critério de parada foi definido pelo número máximo de ciclos (épocas). Após o término do treinamento da rede, esta foi testada via simulação. Esse processo compreende a apresentação de cada vetor de entrada à rede, e esta responde indicando qual o neurônio que apresenta os pesos das conexões mais próximos daquele dos valores do vetor apresentado (diz-se o “neurônio vencedor”).

Então, tendo-se todos os dados de saída da rede (todos os neurônios vencedores), pôde-se determinar e analisar a performance da mesma. Discorre-se sobre essa etapa no próximo capítulo.

08 CONFRONTANDO OS DADOS: SOFTWARE X REDE NEURAL AUTO-ORGANIZÁVEL

Todos os trabalhos realizados nos capítulos anteriores serviram para fornecer os insumos utilizados neste capítulo. A fim de se atingir esse grau de proficiência de informações, seguiu-se metodicamente todos os passos definidos junto aos orientadores. Além de um *software* funcional, tem-se até aqui uma lista de candidatos ordenados pela pontuação que alcançaram no questionário da seleção hipotética, um conjunto de dados obtidos a partir dela, bem como uma Rede Neural configurada para trabalhar com esses dados. Adiante se discorre sobre os resultados obtidos a partir dos dados coletados.

8.1 CLASSIFICAÇÃO EFETUADA PELO SOFTWARE.

O que se fez neste trabalho e que se supõe é que: ao elaborar o questionário a ser aplicado aos candidatos, o profissional competente siga a lógica de que as respostas mais valorizadas (mais adequadas ao perfil profissional/pessoal que se busca) para a seleção de competências em questão recebam as maiores pontuações. Do mesmo modo, as respostas dissonantes do perfil buscado devem receber pontuações menores e até negativas. Sendo assim, como a classificação feita pelo *software* Perfil é um somatório de todas as alternativas do questionário assinaladas pelo candidato, sejam elas positivas ou negativas, é fácil deduzir que o candidato melhor classificado é aquele que fez mais pontos em relação aos demais. Sendo assim, ele ocupará a posição imediatamente posterior ao candidato “ideal”. Os outros virão em seguida, ordenados sequencialmente em ordem decrescente de pontuação, até que se chegue ao último candidato, o chamado “não ideal”.

Nessa classificação é fácil perceber a diferença entre cada candidato e o candidato dito “ideal”, já que os pontos servem de parâmetro para se efetuar a comparação. Além disso, detalhadamente, pode-se averiguar cada resposta dos candidatos por meio do Módulo Visualizar Resultados do Teste.

Mostra-se logo à frente a pontuação dos 30 candidatos, de acordo com as respostas que eles assinalaram no questionário implementando pelo *software* Perfil:

Nome do Candidato	Pontuação	Nome do Candidato	Pontuação
(ideal) 1	250	16	149
2	214	17	148
3	195	18	144
4	189	19	128
5	186	20	119
6	174	21	109
7	173	22	107
8	173	23	106
9	172	24	103
10	166	25	97
11	165	26	78
12	161	27	76
13	158	28	48
14	155	29	14
15	151	(não ideal) 30	-90

Figura 15: Pontuação dos candidatos.

Nota-se que os nomes dos candidatos foram substituídos por números para se preservar as suas identidades. O número também corresponde à colocação em que cada candidato foi posicionado devido ao número de pontos obtidos. Logo, é fácil fazer uma leitura a partir da lista e concluir quais são os candidatos que estão mais próximos do perfil delineado como ideal. Isso de acordo com as competências que eles alegaram no momento de responderem às questões.

8.2 CLASSIFICAÇÃO EFETUADA PELA RNA.

Já para a rede neural, sugerir uma classificação ordenada não é tão simples assim. A fim de compreender-se essa dificuldade, é útil recapitular o que abordou-se em relação às Redes Auto-Organizáveis em capítulos anteriores: segundo a filosofia e o propósito dessas redes, elas servem para classificar sinais, sendo que mantém na classificação destes sinais uma ordenação topológica dos mesmos, de acordo com as suas características. Explicando melhor: ela agrupa cada um dos sinais semelhantes em regiões topológicas próximas uma das outras, refletindo a

similaridade de características encontradas entre esses sinais. Logo, um padrão de sinal A, o qual é muito semelhante ao sinal B, será representado em uma região muito próxima a da região de B.

Essa definição das regiões a serem ocupadas é determinada pelo neurônio vencedor. Ele responde maximamente caso seu vetor de pesos assemelha-se aos dados do vetor de sinais que é amostrado à rede. Sabe-se de outrora que esses pesos são ajustados durante o processo de treinamento da rede, ocasião em que os vetores de entrada são apresentados aleatoriamente à rede. Sabe-se ainda que quanto mais semelhantes aos dados do vetor de entrada forem os pesos do neurônio, menor será a distância (somatório das diferenças entre os valores dos vetores).

Neurônios próximos ao neurônio vencedor também participam do processo de aprendizado, ou seja, seus pesos são alterados em direção ao vetor de entrada. Dessa forma, quando um vetor B excitar as entradas da rede, quanto mais semelhante este for ao vetor A, o qual determinou o neurônio vencedor N1, mais próximo da região do neurônio vencedor N1 é que emergirá uma resposta da rede para o vetor B.

Com base nisso é que se esperava que a rede agrupasse os candidatos cujos dados do questionário mais se assemelhassem aos do candidato ideal. O agrupamento deveria ocorrer em uma região próxima ao neurônio vencedor determinado pelo vetor de dados do candidato “ideal”.

Porém, depois de inúmeros testes, pode-se perceber a falta de precisão oriunda desse agrupamento. Alguns vetores até que mantiveram-se fiéis em relação a classificação efetuada pelo *software* Perfil, ou seja, os candidatos que mais pontuaram estavam melhor posicionados (mais próximos) do candidato “ideal” no mapa gerado pela rede. Logo a seguir, isso pode ser verificado no gráfico, para os candidatos de número 2, 3. E numa distância mais acentuada, para os candidatos de número 4 e 5, bem como para o 7 e 8.

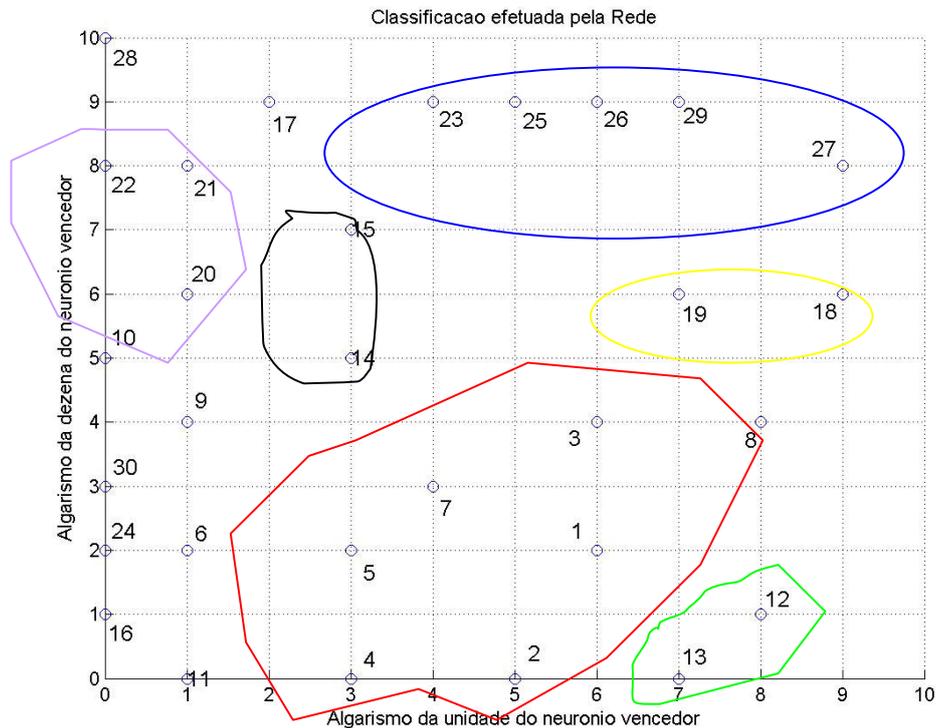


Figura 16: Saída da Rede Neural Auto-Organizável (5000 épocas).

Ainda é possível se observar alguns agrupamentos de candidatos isolados, ou seja, a rede entendeu que eles apresentavam características individuais singulares a ponto de merecerem uma categorização diferente (um novo *cluster*). Essas disparidades percebidas pela rede distanciaram o posicionamento dos candidatos em relação à área de circunscrição do candidato ideal. Logo, o confronto imediato da classificação feita pelo *software* e pela rede não pôde ser efetivamente realizado.

Ressalta-se que foram exaustivas as tentativas de se ajustar a rede ao problema, sendo que vários parâmetros foram alternadamente experimentados a fim de se gerar um resultado mais apreciável:

- número de épocas usadas no treinamento (de 0 a 40000 épocas);
- função de aprendizagem;
- número de neurônios da camada competitiva (8x8, 10x10, 20x20);
- topologia da rede (hextop, gridtop, randtop);
- diminuição do número de vetores usados no treinamento (30, 20, 10, 5);

- uso de vetores parametrizados para o treinamento (cada um deles composto por um mesmo valor. Ex: [10 10 10 ...] , [9 9 9...], [8 8 8...], etc.);
- replicação dos dados de forma uniforme etc. (repetir o mesmo vetor na entrada).

Como não se conseguiu obter as saídas claramente organizadas, mostra-se aqui o melhor que foi possível se alcançar com diversas redes, sendo que o que diferencia uma da outra é apenas o número de épocas usadas no treinamento:

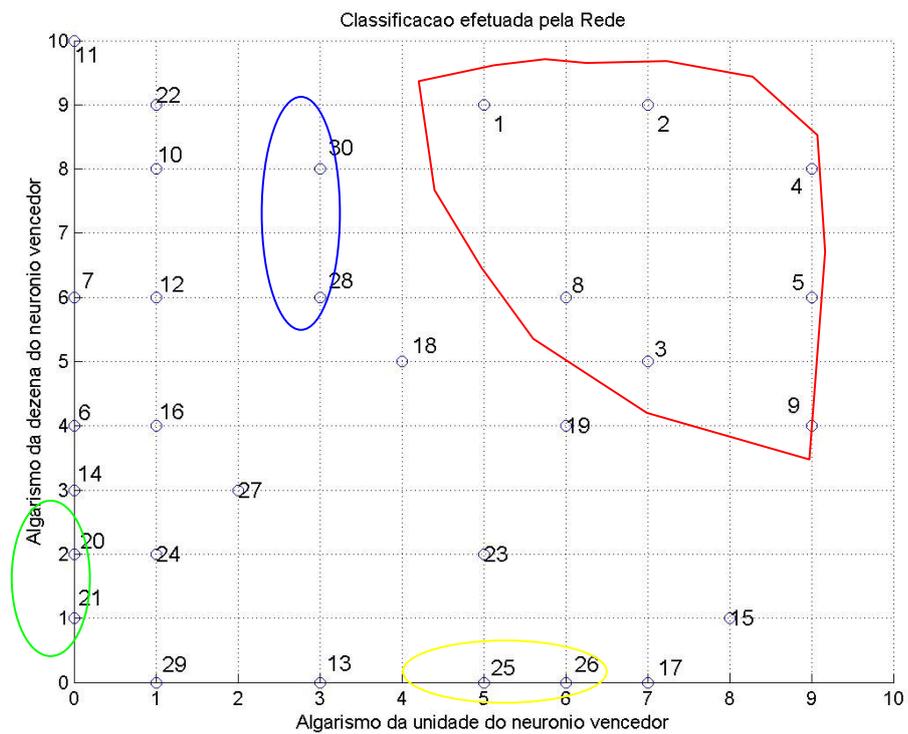


Figura 17: Saída da Rede Neural Auto-Organizável (5000 épocas).

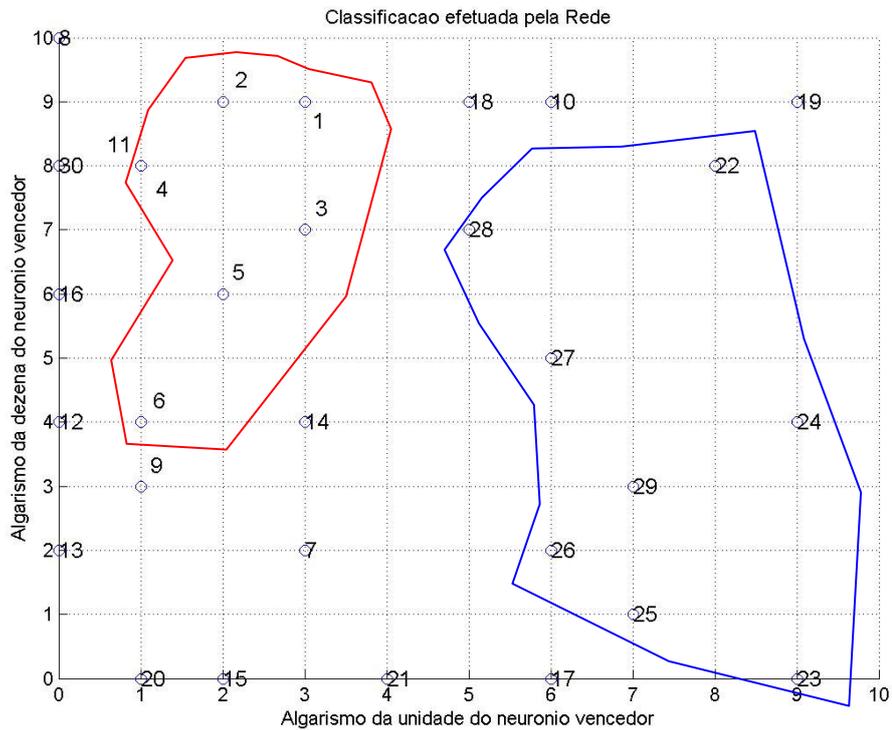


Figura 18: Saída da Rede Neural Auto-Organizável (3500 épocas).

8.3 DISCREPÂNCIAS ENTRE AS CLASSIFICAÇÕES.

Levanta-se agora alguns dos fatores que colaboraram para aumentar a discrepância entre as duas classificações. Eles mais se devem à semântica e às características dos dados do que à configuração da rede neural artificial.

1. Semântica dos Dados

Para o software, a pontuação e consequentemente a classificação dos candidatos é dada pelo somatório de todas as alternativas do questionário assinaladas. Assim sendo, se um candidato José obtivesse o seguinte vetor de respostas:

Candidato José = [10 0 10 0 10 0 10 0 10 0]

$\Sigma = 50$ pontos.

E um candidato Pedro:

Candidato Pedro = [0 10 0 10 0 10 0 10 0 10]

$\Sigma = 50$ pontos.

Ambos apresentam a mesma pontuação, logo apareceriam juntos na lista da classificação por pontos. Pode-se então concluir que eles obtiveram o mesmo desempenho.

Já para a rede, eles são casos completamente distintos, pois ela utiliza simultaneamente todas as variáveis do vetor para a interpretação do conjunto de dados. Além disso, a rede analisa individualmente os dados do vetor, já que cada um deles representa uma entrada para um neurônio. Assim sendo, a ordem dos dados importa e [10 0 10 0 10 0 10 0 10 0] é muito diferente de [0 10 0 10 0 10 0 10 0 10].

Diante do exposto, pode-se concluir que ela não faz uma classificação quantitativa e ordenada, mas sim uma classificação qualitativa dos dados de entrada de acordo com suas características. Poderia-se dizer que ela encontra “padrões de perfis pessoais” mediante a análise das respostas dos candidatos. Bastaria saber que perfis são esses, porém isso já foge da alçada desse trabalho. Logo é útil se salientar que este assunto poderia ser explorado em trabalhos futuros.

2. Métricas usadas pela Rede

Aos olhos dos proponentes deste trabalho, a distância Manhattan (ou distância de Hamming) não é um bom fator de medida para se tentar classificar candidatos de acordo com as características de suas respostas. Por exemplo, se a resposta de uma questão possuísse, de acordo com a Psicometria, uma capacidade muito grande de diferenciar entre um tipo de perfil de pessoa e outro, esse grau de ênfase deveria ser transposto para a rede. Mostra-se agora um exemplo ilustrativo de um pequeno questionário, onde a pergunta 1 foi definida como “divisor de águas” para os candidatos. Ela serve para definir o gênero e foi valorada com 1 para Masculino e 0 para Feminino:

1) Qual o seu sexo?

(1) Masculino

(0) Feminino

2) Você tem filhos?

(2) Sim

(0) Não

3) Você trabalha?

(3) Sim

(4) Não

Supondo que o sexo definisse características e comportamentos totalmente diferentes entre as pessoas, então um candidato que assinalasse “Masculino”, independente de todas as outras respostas serem iguais às de uma candidata, a qual, evidentemente assinalou para essa mesma pergunta “Feminino”, deveria ser classificado em um agrupamento totalmente diferente do dela. Para isso, o neurônio de entrada que trata dessa característica deveria ser especializado para trabalhar com o valor de entrada de forma diferenciada, ou seja, o cálculo da distância entre o padrão de entrada e os pesos do neurônio poderia ser modificado a fim de enfatizar ou amenizar o julgamento do quão esses vetores são parecidos.

A fim de se ilustrar esse discurso e baseando-se no mini-questionário anteriormente visto, mostra-se agora o exemplo de uma classificação efetuada por uma rede com dois neurônios na camada competitiva, onde tem-se:

Vetores dos Candidatos³:

José = [**1** 2 3] (candidato 1)

Maria = [**0** 2 3] (candidato 2)

Filomena = [**0** 2 4] (candidato 3)

³ Em negrito estão as entradas referentes ao sexo dos candidatos.

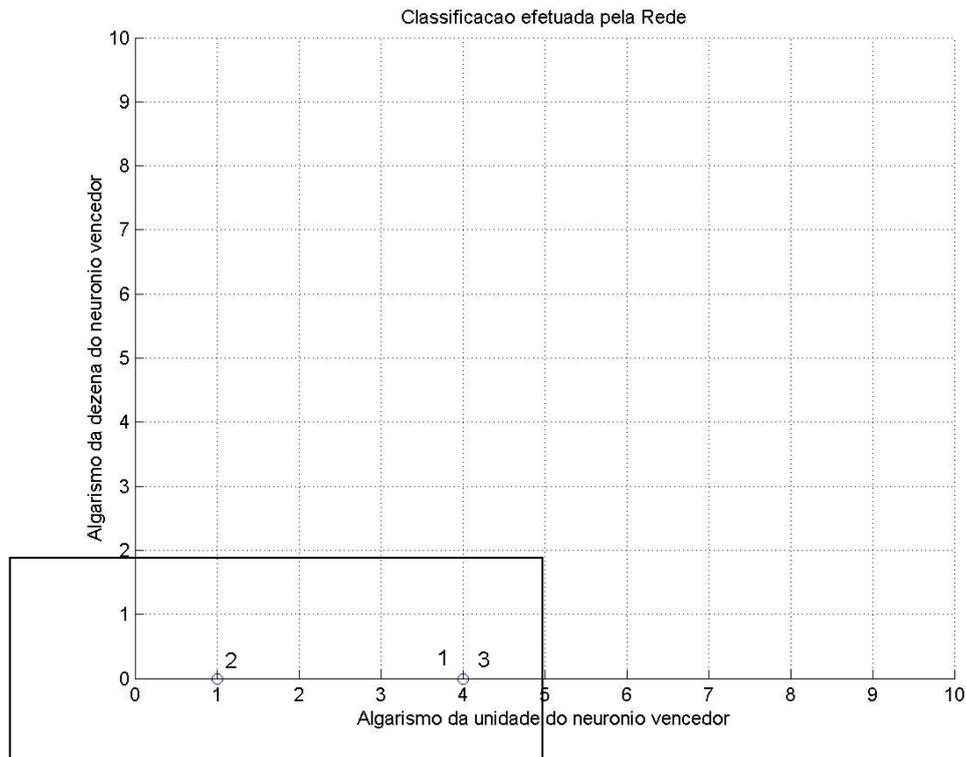


Figura 19: Saída da Rede Neural Auto-Organizável (exemplo).

Observa-se que os candidatos José (1) e Filomena (3) foram classificados da mesma forma, independente de serem de sexos diferentes. Já Maria (2), que deveria ser classificada junto com Filomena, ficou isolada e também não obteve uma boa classificação. Pelo visto, em virtude dos neurônios vencedores, candidatos diferentes foram classificados no mesmo *cluster*, ao passo que candidatos semelhantes ficaram separados, dada a falta de preciosismo⁴ da Rede em tratar a característica “gênero”.

Visando a obtenção de uma classificação melhor, também pensou-se em aumentar a pontuação para determinadas características. Tomando-se o exemplo anterior, só que agora redefinindo os valores da questão 1, onde “Masculino” foi valorado com 100, mostra-se os novos resultados:

1) Qual o seu sexo?

(100) Masculino

⁴ Neste caso, há uma carência de critérios mais nobres para a Rede tratar do gênero.

(0) Feminino

2) Você tem filhos?

(2) Sim

(0) Não

3) Você trabalha?

(3) Sim

(4) Não

Vetores dos Candidatos:

José = [100 2 3] (candidato 1)

Maria = [0 2 3] (candidato 2)

Filomena = [0 2 4] (candidato 3)

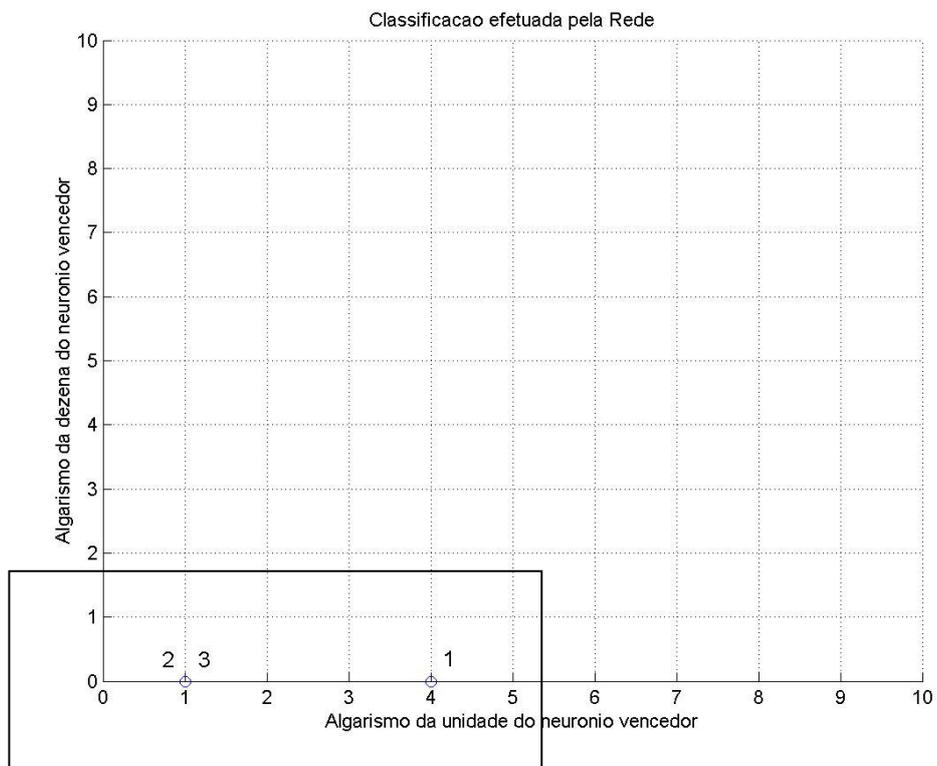


Figura 20: Saída da Rede Neural Auto-Organizável (exemplo).

Observa-se que a classificação foi precisa e todos os candidatos agora estão representados por neurônios que captaram bem as suas características, inclusive a de gênero. Se essa métrica fosse estendida ao *software* Perfil, uma classificação baseada no somatório dos pontos geraria “degraus” na pontuação, já que algumas alternativas seriam super-valoradas em detrimento a outras. Para o exemplo mostrado, uma possível classificação seria a seguinte:

Nome do Candidato	Pontuação
José	105
Filomena	6
Maria	5

Figura 21: Pontuação dos candidatos (exemplo).

A fim de se sanar esses problemas relativos à semântica dos dados, uma possível solução seria a relativização do impacto de cada questão no algoritmo que calcula a distância entre os vetores e os pesos dos neurônios. Assim, questões com caráter mais decisório na definição de um perfil ou outro impactariam mais nas decisões da rede, ao passo que outras mais insignificantes, não afetariam tanto.

09 CONCLUSÃO

“Se nossa meta ao compor um programa for simular o desempenho humano em uma tarefa, a medida de sucesso será o grau em que o comportamento do programa corresponda ao desempenho humano, conforme medição por vários tipos de experimentos e análises de protocolo. Não queremos, com isso, um programa que faça tudo tão bem quanto possível, mas também um programa que falhe quando as pessoas falharem” [RICH, 1988].

A partir desse trecho retirado de Rich, pode-se ilustrar e de certa forma justificar o desempenho aquém do esperado que obteve-se com a Rede Neural na tentativa de perceber similaridades entre os dados dos candidatos respondentes e os dados do candidato delineado como ideal. Não se conseguiu plenamente o confronto de informações que era esperado, porém abre-se um caminho para que outros tomem gosto pelo desafio e colaborem com idéias e estudos para alimentar as pesquisas que visam fornecer respostas à indagação que este trabalho parcialmente pôde responder .

Sabe-se que é preciso experiência e técnica para se efetuar uma boa seleção de pessoal, bem como para se tratar problemas complexos ligados a I.A. Assim sendo, durante esse breve tempo de contato que se teve com as teorias das áreas de I.A. e de Psicologia, os proponentes dessa monografia se julgaram aptos a apontar para algumas direções a serem seguidas.

A fim de se melhorar a classificação dos candidatos, o que se sugere é um trabalho mais detalhado junto a psicólogos especialistas em Psicometria. Esses esforços poderiam levar a construção de questionários mais precisos, os quais serviriam para se extrair e medir mais precisamente as características dos candidatos.

Então, tendo-se um questionário devidamente “calibrado” e preciso, os pormenores deste poderiam ser mapeados para a rede. Nesse ponto junta-se uma nova ferramenta de trabalho, a qual foi visualizada somente no final da elaboração dessa monografia. Trata-se da **Análise de Componentes Principais**, técnica do campo da Estatística que, com os devidos estudos e adequações, poderia ajudar na transposição da ênfase oriunda das perguntas do questionário para as entradas da rede. Resumidamente, a Análise de Componentes Principais trata da geração de novas variáveis por meio de uma transformação matemática especial realizada sobre as variáveis originais. Cada componente principal é uma combinação linear de todas as variáveis originais. Por exemplo, um sistema com oito variáveis, após a transformação, teria oito componentes

principais. Cada uma destas componentes principais, por sua vez, seria escrita como uma combinação linear das oito variáveis originais. Nestas combinações, cada variável terá uma importância ou peso diferente [J. M. MOITA NETO, 2004].

Assim sendo, esse seria o caminho a ser buscado para refletir a semântica e a importância de cada resposta do questionário para a rede. Talvez o desenvolvimento de uma nova ferramenta, a qual pudesse mapear os valores atribuídos a cada alternativa das questões para a rede, enfatizando ou suavizando os valores quando do cálculo da distância entre os vetores.

Além da importância dada aos estudos desenvolvidos nas áreas teóricas da Psicologia e I.A., dá-se crédito também a área ferramental da Ciência da Computação, a qual forneceu meios para que os autores dessa monografia, na qualidade de desenvolvedores de *software*, pudessem traduzir seus anseios em mais uma ferramenta a ser usada por usuários de áreas diversas. Na visão dos autores, esse é o maior objetivo da Ciência da Computação, ou seja, não findar-se em si mesma, mas abarcar outras áreas e oferecer todo seu potencial a fim de amenizar ou resolver problemas, principalmente aqueles cuja solução não seja de cunho criativo, mas sim repetitivo e maçante.

Apesar deste trabalho findar-se sem respostas muito precisas, dado o tamanho do desafio, sabe-se, sem sombra de dúvida, que os esforços não foram em vão, porque além da experiência adquirida ao se buscar associar conhecimentos de áreas tão distintas, como a Inteligência Artificial e a Psicologia, nota-se que o saber ainda concentra-se no fator humano. Baseando-se nos processos envolvidos com o raciocínio humano é que são extraídos os modelos cognitivos inspiradores das técnicas de I.A.. É a partir do estudo do homem é que a Psicologia desenvolve suas pesquisas, bem como a Psicométrica elabora e explora as ferramentas de aferição dos estados internos do homem. Também é a partir das mãos dos desenvolvedores de *software* que a Ciência da Computação e áreas afins edificam os programas, os atuais instrumentos de trabalho presentes até nas mais simples atividades humanas.

De modo poético e fazendo-se uma analogia com as artes, pode-se dizer que o *software* que foi implementado nada mais é que o pincel de um artista, ou seja, a partir do uso dele é que a inspiração e as idéias do pintor são refletidas na tela nua. O melhor quadro que se gostaria que ele ajudasse a pintar seria um em que as pessoas estivessem felizes desempenhando suas atividades laborais, que suas inteligências não fossem subutilizadas, que suas profissões não fossem impostas pela lei das poucas oportunidades, bem como pela lei do que é que se sobra

para fazer e, principalmente, seus esforços não fossem só motivados por uma simples recompensa financeira.

Se algum dia essa paisagem for composta, os autores dessa monografia iriam se sentir muito orgulhosos de ter contribuído para que o artista principal da existência assinasse nessa obra de arte: “O Homem”.

10 APÊNDICES

A-CÓDIGO FONTE SOFTWARE

```
package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class CadastrarPerguntaCMD implements Command{

private String next;

public CadastrarPerguntaCMD(String next){

    this.next = next;

}

public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

    return next;

}

}

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/** Interface que deve ser implementada por todas as classes que irão executar um commando específico a partir de um dado formulário. Especifica um único método, o execute(). */

public interface Command {

    public String execute(HttpServletRequest req, HttpServletResponse response) throws CommandException;

}

package br.com.perfil.action;

/**

*Classe que exibe textos para excessões que ocorrem na classe Command.

**/

public class CommandException extends Exception {
```

```

/**Construtor desta classe. */
public CommandException()
{
    super();
}
/**
 * @param cause
 */
public CommandException(Throwable cause)
{
    super(cause);
}
/**
 * @param message
 * @param cause
 */
public CommandException(String message, Throwable cause)
{
    super(message, cause);
}
/**Construtor desta classe com parametro do texto da mensagem a ser exibida. */
public CommandException(String msg)
{
    super(msg);
}
}
package br.com.perfil.action;
import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;

public class MostrarCadastroProjeto implements Command {

    private String next;

    public MostrarCadastroProjeto(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        return next;

    }

}

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class MostrarFormGeraPergunta implements Command {

    private String next;

    public MostrarFormGeraPergunta(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        if ("descreverpergunta".equals(req.getParameter("cmd"))) {

            req.setAttribute("numOpcoes", req.getParameter("numOpcoes"));

            req.setAttribute("descreverpergunta", "descreverpergunta");

            req.setAttribute("tipoPergunta", req.getParameter("tipoPergunta"));

        } else if ("cmdtipopergunta".equals(req.getParameter("cmd"))) {

            req.setAttribute("tipoPergunta", req.getParameter("tipoPergunta"));

        }

    }

}

```

```

        req.setAttribute("mostrargerapergunta", "mostrargerapergunta");
        return next;
    }
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class MostrarProjetoFuncionario implements Command {

    private String next ;

    public MostrarProjetoFuncionario(String next){

        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        return next;
    }
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ProcesseProjetos implements Command {

    private String next;

    public ProcesseProjetos(String next) {

        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

```

```

        throws CommandException {
    return next;
}
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class ProcesseSair implements Command {
    private String next;

    public ProcesseSair(String next) {
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        HttpSession session = req.getSession();
        session.invalidate();
        return next;
    }
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ProcesseUsuario implements Command {
    private String next;

    public ProcesseUsuario(String next) {

```

```

    this.next = next;
}
public String execute(HttpServletRequest req, HttpServletResponse response)
    throws CommandException {
    return next;
}
}

```

```
package br.com.perfil.action;
```

```

public class Parser {
    public static String formate(String descrPergunta) {
        descrPergunta = descrPergunta.replaceAll("á", "&acute;");
        descrPergunta = descrPergunta.replaceAll("â", "&acirc;");
        descrPergunta = descrPergunta.replaceAll("à", "&agrave;");
        descrPergunta = descrPergunta.replaceAll("ã", "&atilde;");
        descrPergunta = descrPergunta.replaceAll("ç", "&ccedil;");
        descrPergunta = descrPergunta.replaceAll("é", "&eacute;");
        descrPergunta = descrPergunta.replaceAll("ê", "&ecirc;");
        descrPergunta = descrPergunta.replaceAll("í", "&iacute;");
        descrPergunta = descrPergunta.replaceAll("ó", "&oacute;");
        descrPergunta = descrPergunta.replaceAll("ô", "&ocirc;");
        descrPergunta = descrPergunta.replaceAll("õ", "&otilde;");
        descrPergunta = descrPergunta.replaceAll("ú", "&uacute;");
        descrPergunta = descrPergunta.replaceAll("ü", "&uuml;");
        //maiusculas
        descrPergunta = descrPergunta.replaceAll("Á", "&Aacute;");
        descrPergunta = descrPergunta.replaceAll("Â", "&Aacirc;");
        descrPergunta = descrPergunta.replaceAll("Ã", "&Agrave;");
    }
}

```

```

    descrPergunta = descrPergunta.replaceAll("Â", "&Atilde;");
    descrPergunta = descrPergunta.replaceAll("Ç", "&Ccedil;");
    descrPergunta = descrPergunta.replaceAll("É", "&Eacute;");
    descrPergunta = descrPergunta.replaceAll("Ê", "&Ecirc;");
    descrPergunta = descrPergunta.replaceAll("Í", "&Iacute;");
    descrPergunta = descrPergunta.replaceAll("Ó", "&Oacute;");
    descrPergunta = descrPergunta.replaceAll("Ô", "&Ocirc;");
    descrPergunta = descrPergunta.replaceAll("Õ", "&Otilde;");
    descrPergunta = descrPergunta.replaceAll("Ú", "&Uacute;");
    descrPergunta = descrPergunta.replaceAll("Ü", "&Uuml");
    return descrPergunta;
}
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.bean.PerguntaBean;
import br.com.perfil.dao.PerfilDAO;

public class CadastrarPergunta implements Command {
    private String next;

    public CadastrarPergunta(String next){
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {

```

```

HttpSession session = req.getSession();

int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());

    PerguntaBean pergunta = new PerguntaBean();

    String descrPergunta = req.getParameter("descrPergunta");

    System.out.println("descrPergunta antes :" + descrPergunta);

    descrPergunta = Parser.formate(descrPergunta);

    System.out.println("descrPergunta depois :" + descrPergunta);

//pergunta.setValorPergunta(Integer.parseInt(req.getParameter("valorPerg")));

    int tpPergunta = Integer.parseInt(req.getParameter("tipoPergunta"));

    PerfilDAO dao = new PerfilDAO();

    int id_pergunta = dao.inserirPergunta(descrPergunta,tpPergunta,id_projeto);

    if(!(tpPergunta == 3)){//diferente de pergunta descritiva

        String[] strDescOpc = req.getParameterValues("descrOpcao");

        String[] strValorOpc = req.getParameterValues("valorOpcao");

        for (int i = 0; i < strValorOpc.length; i++) {

dao.inserirOpcoes(id_pergunta,Parser.formate(strDescOpc[i]),Integer.parseInt(strValorOpc[i]));

        }

    }

    return next;

}

}

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import br.com.perfil.bean.ProjetoBean;

import br.com.perfil.dao.PerfilDAO;

```

```

public class CadastrarQtdAdjetivos implements Command {

    private String next;

    public CadastrarQtdAdjetivos(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        int qtd = Integer.parseInt(req.getParameter("qtd_adjetivos"));

        ProjetoBean projetoBean = (ProjetoBean)req.getSession().getAttribute("projetoBean");

        new PerfilDAO().inserirQtdAdjProjeto(projetoBean.getId_projeto(),qtd);

        projetoBean.setQtd_adjetivos(new Integer(qtd));

        return next;

    }

}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import br.com.perfil.dao.PerfilDAO;

public class MostrarAdjetivos implements Command {

    private String next;

    public MostrarAdjetivos(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        int id_projeto = Integer.parseInt(req.getSession().getAttribute("id_projeto").toString());

```

```

        PerfilDAO bd = new PerfilDAO();

        req.getSession().setAttribute("qtd_adj",bd.getProjeto(id_projeto).getQtd_adjetivos());

        return next;
    }
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import br.com.perfil.dao.PerfilDAO;

public class MostrarCadastroCandidato implements Command {

    private String next;

    public MostrarCadastroCandidato(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

    throws CommandException {

        System.out.println("selectProjetos:"+ req.getParameter("selectProjetos"));

        req.getSession().setAttribute("id_projeto",req.getParameter("selectProjetos"));

        req.getSession().setAttribute("projetoBean",new

PerfilDAO().getProjeto(Integer.parseInt(req.getParameter("selectProjetos"))));

        return next;

    }

}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```

```

import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class MostrarCandidatos implements Command {

    private String next ;

    public MostrarCandidatos(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

    throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt((String)session.getAttribute("id_projeto"));

        PerfilDAO bd = new PerfilDAO();

        req.setAttribute("arrCandidatos",bd.getAllCandidatos(id_projeto));

        System.out.println(bd.getAllCandidatos(id_projeto).get(0));

        return next;

    }

}

```

```

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import br.com.perfil.dao.PerfilDAO;

public class MostrarProjetos implements Command {

    private String next ;

    public MostrarProjetos(String next){

        this.next = next;

    }

}

```

```

public String execute(HttpServletRequest req, HttpServletResponse response)
throws CommandException {
    try {
        PerfilDAO bd = new PerfilDAO();
        ArrayList arrProjetos = bd.getAllProjetos();
        req.setAttribute("arrProjetos", arrProjetos);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return next;
}
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseCadastrarAdjetivo implements Command {
    private String next;

    public ProcesseCadastrarAdjetivo(String next) {
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
    throws CommandException {
        HttpSession session = req.getSession();
        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());
        String adj = req.getParameter("adjetivo");

```

```

PerfilDAO bd = new PerfilDAO();

//adjetivos são inseridos com status não selecionado

try {

    bd.inserirAdjetivo(id_projeto, Parser.formate(adj),0, "n");

    next = "controleDeAdjetivos.jsp";

} catch (Exception e) {

    e.printStackTrace();

}

return next;

}

}

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import br.com.perfil.dao.PerfilDAO;

public class ProcesseDefinirPerfil implements Command {

    private String next;

    public ProcesseDefinirPerfil(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException

        return next;

    }

}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import br.com.perfil.dao.PerfilDAO;

public class ProcessoDescadastrarProjeto implements Command {

    private String next;

    public ProcessoDescadastrarProjeto(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        int nome = Integer.parseInt(req.getParameter("selectProjetos"));

        PerfilDAO bd = new PerfilDAO();

        try {

            bd.excluaProjeto(nome);

        } catch (Exception e1) {

            // TODO Auto-generated catch block

            e1.printStackTrace();

        }

        return next;

    }

}

```

```

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseGerarMatriz implements Command {

    private String next;

    public ProcesseGerarMatriz(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        PerfilDAO bd = new PerfilDAO();

        String adj;

        ArrayList adjetivos = new ArrayList();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());

        String checkMatriz[] = req.getParameterValues("checkMatriz");

        for (int i = 0; i < checkMatriz.length; i++) {

            bd.setStatusSelecParaMatrizAdjetivo(id_projeto, checkMatriz[i]);

        }

        return next;

    }

}

```

```

package br.com.perfil.action;

import java.util.Enumeration;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import br.com.perfil.dao.PerfilDAO;

```

```

public class ProcesseInserirAdjetivo implements Command {
    private String next;

    public ProcesseInserirAdjetivo(String next) {
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        String adjetivoSelecEx;
        int id_projeto = Integer.parseInt(req.getSession().getAttribute("id_projeto").toString());
        int id_funcionario = Integer.parseInt(req.getSession().getAttribute("id_funcionario").toString());
        PerfilDAO bd = new PerfilDAO();
        String[] opcoes = req.getParameterValues("opcoes");
        for (int i = 0; i < opcoes.length; i++) {
            bd.inserirAdjetivoSelecFuncionario(id_projeto, id_funcionario, opcoes[i], "s");
        }
        return next;
    }
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.bean.UsuarioBean;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseLogin implements Command {
    private String next;

    public ProcesseLogin(String next) {

```

```

    this.next = next;
}

public String execute(HttpServletRequest req, HttpServletResponse response)
    throws CommandException {
    HttpSession session = req.getSession();
    String usuario = req.getParameter("usuario");
    String senha = req.getParameter("senha");
    HttpSession sessao = req.getSession();
    sessao.setAttribute("login", "invalido");
    System.out.println("usuario" + usuario);
    if ((usuario != null) && (senha != null)) {
        try {
            PerfilDAO bd = new PerfilDAO();
            UsuarioBean usuarioBean = bd.getUsuario(usuario, senha);
            System.out.println("usuarioBean" + usuarioBean);
            if (usuarioBean != null) {
                System.out.println("usuario cadastrado");
                if (usuarioBean.getIdTipoUsuario() == 0) { //admin
                    next = "servlet/perfil?cmd=mostrar_projetos";
                } else if (usuarioBean.getIdTipoUsuario() == 1) { // funcionario para preencher a matriz
                    int id_projeto = bd.getProjetoFuncionario(usuarioBean
                        .getId_usuario());
                    session.setAttribute("id_projeto", "" + id_projeto);
                    session.setAttribute("projetoBean", ""
                        + bd.getProjeto(id_projeto));
                    session.setAttribute("id_funcionario", ""
                        + usuarioBean.getId_usuario());
                    next = "menuFuncionario.jsp";
                }
            }
        }
    }
}

```

```

    } else {
        next = "usuario.jsp";
    }
    session.setAttribute("id_usuario", ""
        + usuarioBean.getId_usuario());
    session.setAttribute("usuarioBean", usuarioBean);
    req.setAttribute("autorizado", "sim");
    req.removeAttribute("login");
} else {
    next = "index.jsp";
}
} catch (Exception e) {
    e.printStackTrace();
}
}
return next;
}
}

```

```

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.com.perfil.bean.MatrizBean;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseMatriz implements Command {
    private String next;

```

```

public ProcesoMatriz(String next) {
    this.next = next;
}

public String execute(HttpServletRequest req, HttpServletResponse response)
    throws CommandException {
    HttpSession session = req.getSession();
    int id_projeto = Integer.parseInt(session.getAttribute("id_projeto")
        .toString());
    System.out.println("cheguei1");
    int id_usuario = Integer.parseInt(session.getAttribute("id_usuario")
        .toString());
    System.out.println("cheguei2");
    PerfilDAO bd = new PerfilDAO();
    ArrayList arrAdjetivosMatriz = bd.getAllAdjetivosSelecionadosParaMatriz(id_projeto);
    MatrizBean matriz = new MatrizBean();
    matriz.setArray(arrAdjetivosMatriz.size());
    System.out.println("valor01 = " +req.getParameter("valor01"));
    for(int i = 0; i < (matriz.getTamanho()-1); i++){
        for(int k = i; k < (matriz.getTamanho()); k++){
            if(i != k){
                matriz.setElementAt(i, k, Integer.parseInt(req.getParameter("valor"+i+k)));
                System.out.println("matriz" + i + k + " = " +
                    Integer.parseInt(req.getParameter("valor"+i+k)));
            }
        }
    }
    matriz.espelharMatriz();
    matriz.someColunas();
}

```

```

        for (int i = 0; i < matriz.getTamanho(); i++) {
            bd.inserirSomaColuna(id_projeto,id_usuario,i,matriz.getSomatorio(i));
        }
        System.out.println(matriz.getTodosElementos());
        System.out.println(matriz.getTodoSomatorio());

        return next;
    }
}

package br.com.perfil.action;

import java.util.ArrayList;
import java.util.Enumeration;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseRejeitarAdjetivo implements Command {
    private String next;

    public ProcesseRejeitarAdjetivo(String next) {
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        String adjetivoSelecEx;
        HttpSession session = req.getSession();
        int id_projeto = Integer.parseInt(req.getSession().getAttribute("id_projeto").toString());
        int id_funcionario = Integer.parseInt(req.getSession().getAttribute("id_funcionario").toString());
        PerfilDAO bd = new PerfilDAO();

```

```

String[] opcoes =req.getParameterValues("seleccionados");

    try {

        for (int i = 0; i < opcoes.length; i++) {

            bd.excluirAdjetivoSelecFuncionario(id_projeto, id_funcionario, opcoes[i]);

        }

    } catch (Exception e1) {

        e1.printStackTrace();

    }

    return next;

}

}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseSuperUsuario implements Command {

    private String next;

    public ProcesseSuperUsuario(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute(("id_projeto"))

            .toString());

        PerfilDAO bd = new PerfilDAO();

```

```

    req.setAttribute("arrUsuarios", bd.getAllFuncionarios(id_projeto));

    return next;
}
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class CadastrarCandidato implements Command {

    private String next;

    public CadastrarCandidato(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

    throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt((String)session.getAttribute("id_projeto"));

        String nome = Parser.formate(req.getParameter("nomeInsercao"));

        String e_mail = req.getParameter("e_mail");

        String cpf = req.getParameter("cpf");

        String rg = req.getParameter("rg");

        String usuario = nome;

        PerfilDAO bd = new PerfilDAO();

        try { //ve se já tem alguém cadastrado com o mesmo nome

            int id_usuario =bd.inserirCandidato(id_projeto,nome,e_mail,rg,cpf);

            session.setAttribute("id_usuario", ""+id_usuario);

```

```

        session = req.getSession();

        req.getSession().setAttribute("arrayPerguntas",bd.getAllPerguntas(id_projeto));

        next = "responderquestionario.jsp";

    } catch (Exception e) {

        e.printStackTrace();

    }

    return next;

}

}

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import br.com.perfil.dao.PerfilDAO;

public class MenuSuperUsuario implements Command {

    private String next;

    public MenuSuperUsuario(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        int id_projeto = Integer.parseInt(req.getParameter("selectProjetos"));

        req.getSession().setAttribute("id_projeto",req.getParameter("selectProjetos"));

        req.getSession().setAttribute("projetoBean",new PerfilDAO().getProjeto(id_projeto));

        System.out.println("projetoBean " + new PerfilDAO().getProjeto(id_projeto));

        return next;

    }

}

```

```

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.com.perfil.bean.PerguntaBean;
import br.com.perfil.dao.PerfilDAO;

public class MostrarFormQuestionario implements Command {

    private String next;

    public MostrarFormQuestionario(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

    throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());

        ArrayList perguntas = new PerfilDAO().getAllPerguntas(id_projeto);

        req.getSession().setAttribute("arrayPerguntas",perguntas);

        for (int i = 0; i < perguntas.size(); i++) {

            System.out.println("Pergunta " + i + ":" + ((PerguntaBean)perguntas.get(i)).getStrPergunta());

        }

        return next;

    }

}

```

```

package br.com.perfil.action;

import java.sql.Date;

import javax.servlet.http.HttpServletRequest;

```

```

import javax.servlet.http.HttpServletResponse;
import br.com.perfil.dao.PerfilDAO;
public class ProcesseCadastrarProjeto implements Command {
    private String next;
    public ProcesseCadastrarProjeto(String next) {
        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        String nome = req.getParameter("nome_projeto");
        //Date data = Date.valueOf(req.getParameter("data_projeto"));
        PerfilDAO bd = new PerfilDAO();
        try {
            System.out.println("cadastrando projetos1212: " +next );
            bd.inserirProjeto(Parser.formate(nome), new Date(System.currentTimeMillis()));
        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        return next;
    }
}

```

```

package br.com.perfil.action;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

```

```

import org.apache.log4j.Logger;

import br.com.perfil.dao.PerfilDAO;

public class ProcesseCadastrarUsuario implements Command {

    private String next;

    private static Logger log;

    public ProcesseCadastrarUsuario(String next) {

        this.next = next;

        log = Logger.getLogger(ProcesseCadastrarUsuario.class);
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt((String)session.getAttribute("id_projeto"));

        String nome = req.getParameter("nomeInsercao");

        String usuario = req.getParameter("usuario");

        String senha = req.getParameter("senha");

        String e_mail = req.getParameter("e_mail");

        String cpf = req.getParameter("cpf");

        String rg = req.getParameter("rg");

        PerfilDAO bd = new PerfilDAO();

        if ((usuario != null) && (senha != null)) {

            try { //ve se ja tem alguem cadastrado com o mesmo nome

                if(bd.procurarUsuario(id_projeto,usuario,senha) == false){

                    bd.inserirFuncionario(id_projeto, Parser.formate(usuario), Parser.formate(senha), Parser.formate(nome),

                    Parser.formate(e_mail),cpf, rg);

                    log.info("inseriu nome: "+nome+" usuario: "+usuario+" senha: "+senha+" e-mail: "+e_mail+" cpf: "+cpf+"

                    rg:"+rg);

                    next = "servlet/perfil?cmd=todososcadastrados";

```

```

        }else{
            req.setAttribute("msg","Erro: ja existe um usuario cadastrado com esse login e esta senha!");
            log.info("Setou atributo msg...");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
return next;
}
}

```

```

package br.com.perfil.action;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import br.com.perfil.dao.PerfilDAO;

public class ProcesseDefinirMatriz implements Command {

    private String next;

    public ProcesseDefinirMatriz(String next) {

        this.next = next;
    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute(("id_projeto")).toString());

        PerfilDAO bd = new PerfilDAO();

```

```

        ArrayList arrAdjetivos = bd.getAllAdjetivosMaisCitados(id_projeto);
        req.setAttribute("adjetivos",arrAdjetivos );
        System.out.println("ProcesseDefinirMatriz finalizado");
    return next;
}
}

```

```

package br.com.perfil.action;
import java.util.Enumeration;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;
public class ProcesseDescadastrarAdjetivo implements Command {
    private String next;
    public ProcesseDescadastrarAdjetivo(String next) {
        this.next = next;
    }
    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        PerfilDAO bd = new PerfilDAO();
        String adjetivoEspacoAmostralselecEx;
        HttpSession session = req.getSession();
        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto")
            .toString());
        String[] adjetivosExclusao = req.getParameterValues("adjetivos");
        for (int i = 0; i < adjetivosExclusao.length; i++) {
            adjetivoEspacoAmostralselecEx = Parser.formate(adjetivosExclusao[i]);

```

```

System.out.println(adjetivoEspacoAmostralselecEx);

try {

    bd.excluaAdjetivo(id_projeto, objetivoEspacoAmostralselecEx);

} catch (Exception e1) {

    e1.printStackTrace();

}

}

next = "controleDeAdjativos.jsp";

return next;

}

}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;

public class ProcesseDescadastrarUsuario implements Command {

    private String next;

    public ProcesseDescadastrarUsuario(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());

        String adj = req.getParameter("adjetivo");

        int id_funcionario = Integer.parseInt(req.getParameter("id_funcionario"));

```

```

PerfilDAO bd = new PerfilDAO();

    try {

        bd.excluaFuncionario(id_projeto, id_funcionario);

        bd.excluaUsuario(id_funcionario);

        next = "servlet/perfil?cmd=todososcadastrados";

    } catch (Exception e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

    return next;

}

}

```

```

package br.com.perfil.action;

import java.util.Enumeration;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import br.com.perfil.dao.PerfilDAO;

public class ProcesseIncluirNaLista implements Command {

    private String next;

    public ProcesseIncluirNaLista(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto"))

```

```

        .toString());
PerfilDAO oBD = new PerfilDAO();
for (Enumeration e = req.getParameterNames(); e.hasMoreElements();) {
    String adjetivoSelec = e.nextElement().toString();
    if (!adjetivoSelec.equals("opcao")) {
        oBD.incrementeContagemAdjetivo(id_projeto, adjetivoSelec);
    }
}
return next;
}
}

package br.com.perfil.action;
import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.bean.PerguntaBean;
import br.com.perfil.dao.PerfilDAO;
public class CadastrarQuestionario implements Command {
    private String next;
    public CadastrarQuestionario(String next) {
        this.next = next;
    }
    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        ArrayList arrPerguntas = (ArrayList) req.getSession().getAttribute("array");
        HttpSession session = req.getSession();

```

```

int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());
int id_usuario = Integer.parseInt(session.getAttribute("id_usuario").toString());
int pontuacao = 0;
PerfilDAO dao = new PerfilDAO();
PerguntaBean perguntaBean = null;
for (int i = 0; i < arrPerguntas.size(); i++) {
    perguntaBean = (PerguntaBean) arrPerguntas.get(i);
    OpcaoBean[] arrOpcaoBean = perguntaBean.getOpcoes();
    if (arrOpcaoBean != null) {
        switch (perguntaBean.getTpPergunta()) {
            case 0://multipla escolha
                String[] strEscolhas = req.getParameterValues("pergunta" + i);
                System.out.println("strEscolhas: " + strEscolhas);
                int indOpc = -1;
                for (int j = 0; j < strEscolhas.length; j++) {
                    indOpc = Integer.parseInt(strEscolhas[j]);
                    dao.inserirRespostaUsuario(id_projeto, id_usuario,
                                                perguntaBean.getIdPergunta(),
                                                arrOpcaoBean[indOpc].getIdOpcao(),
                                                arrOpcaoBean[indOpc].getValorOpcao() + "");
                    pontuacao += arrOpcaoBean[indOpc].getValorOpcao();
                }
                break;
            case 1://opcao escolhida
                int ind = Integer
                    .parseInt(req.getParameter("pergunta" + i));
                dao.inserirRespostaUsuario(id_projeto, id_usuario,
                                           perguntaBean.getIdPergunta(), arrOpcaoBean[ind]

```

```

        .getIdOpcao(), arrOpcaoBean[ind].getValorOpcao()+ "");
        pontuacao += arrOpcaoBean[ind].getValorOpcao();
        break;
    case 2://multipla valorada
        String[] strValorada = req.getParameterValues("pergunta"+ i);
        System.out.println("strValorada: " + strValorada);
        for (int j = 0; j < strValorada.length; j++) {
            dao.inserirRespostaUsuario(id_projeto, id_usuario,
                perguntaBean.getIdPergunta(), arrOpcaoBean[j].getIdOpcao(), strValorada[j]);
            pontuacao += Integer.parseInt(strValorada[j]);
        }
        break;
    }
}

// descritiva nao tem valor
    } else {
        String strDescricao = req.getParameter("pergunta" + i);
        dao.inserirRespostaUsuario(id_projeto, id_usuario, perguntaBean
            .getIdPergunta(), -1, Parser.formate(strDescricao));
    }
    System.out.println("pontuacao = " + pontuacao);
    dao.insererPontuacao(id_projeto, id_usuario, pontuacao);
}
return next;
}
}

```

```

package br.com.perfil.action;
import java.util.ArrayList;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.bean.PerguntaBean;
import br.com.perfil.dao.PerfilDAO;

public class ExcluirPergunta implements Command {

    private String next;

    public ExcluirPergunta(String next){

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)
throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());

        int id_pergunta = Integer.parseInt(req.getParameter("id_pergunta"));

        System.out.println("id pergunta do excluir: " + id_pergunta);

        int tpPergunta = Integer.parseInt(req.getParameter("tipoPergunta"));

        PerfilDAO dao = new PerfilDAO();

        try {

            dao.excluaPergunta(id_projeto,id_pergunta);

        } catch (Exception e) {

            e.printStackTrace();

        }

        if(!(tpPergunta == 3 || tpPergunta == 1)){//diferente de pergunta descritiva e unica escolha

            try {

                dao.excluaOpcoes(id_pergunta);

            } catch (Exception e1) {

                e1.printStackTrace();

            }

        }

    }
}

```

```

    }
    }
    ArrayList perguntas = dao.getAllPerguntas(id_projeto);
    req.getSession().setAttribute("arrayPerguntas",perguntas);
    return next;
}
}
package br.com.perfil.action;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.dao.PerfilDAO;
public class ProcesseQuestaoDescritiva implements Command {
    private String next;
    public ProcesseQuestaoDescritiva(String next) {
        this.next = next;
    }
    public String execute(HttpServletRequest req, HttpServletResponse response)
        throws CommandException {
        HttpSession session = req.getSession();
        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto")
            .toString());
        int id_usuario = Integer.parseInt(req.getParameter("id_candidato"));
        System.out.println("id_usuario = " + id_usuario);
        int id_pergunta = Integer.parseInt(req.getParameter("id_pergunta"));
        System.out.println("id_pergunta = " + id_pergunta);
        int valor = Integer.parseInt(req.getParameter("valor_nota"+id_pergunta));
        System.out.println("valor da nota = " + valor);
    }
}

```

```

PerfilDAO dao = new PerfilDAO();

dao.inserePontuacaoDescritiva(id_projeto, id_usuario, id_pergunta, valor);

dao.incrementaNumAvaliacoes(id_projeto, id_usuario, id_pergunta);

req.getSession().setAttribute("arrayPerguntas", dao.getAllPerguntas(id_projeto));

    req.setAttribute("arrResposta", dao.getRespostasUsuario(id_projeto, id_usuario));

    req.setAttribute("id_candidato", new Integer(id_usuario));

return next;

}

}

```

```

package br.com.perfil.action;

import java.util.Enumeration;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import br.com.perfil.dao.PerfilDAO;

public class ProcesseAtualizaPontuacaoGeral implements Command {

    private String next;

    public ProcesseAtualizaPontuacaoGeral(String next) {

        this.next = next;

    }

    public String execute(HttpServletRequest req, HttpServletResponse response)

        throws CommandException {

        HttpSession session = req.getSession();

        int id_projeto = Integer.parseInt(session.getAttribute("id_projeto")

            .toString());

        int id_usuario = Integer.parseInt(req.getParameter("id_candidato"));

```

```

PerfilDAO bd = new PerfilDAO();
for (Enumeration e = req.getParameterNames(); e.hasMoreElements();) {
    String element = e.nextElement().toString();
    if(element.startsWith("media")){
        double mediaQuestao = Double.parseDouble(req.getParameter(element));
        System.out.println("ELEMENTO = " + element + "MEDIA = " + mediaQuestao);
        bd.addPontuacaoDescritiva(id_projeto,id_usuario,mediaQuestao);
    }
}
req.setAttribute("arrCandidatos",bd.getAllCandidatos(id_projeto));
return next;
}
}

```

```

package br.com.perfil.action;
import java.util.ArrayList;
import java.util.Iterator;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.bean.RespostaBean;
import br.com.perfil.bean.RespostaOpcaoBean;
import br.com.perfil.dao.PerfilDAO;
public class MostrarFormResposta implements Command {
    private String next;
    public MostrarFormResposta(String next) {
        this.next = next;
    }
}

```

```

public String execute(HttpServletRequest req, HttpServletResponse response)
throws CommandException {
    HttpSession session = req.getSession();
    int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());
    int id_usuario = Integer.parseInt(req.getParameter("candidato"));
    System.out.println("id_usuario : " + id_usuario);

    PerfilDAO dao = new PerfilDAO();
    req.getSession().setAttribute("arrayPerguntas", dao.getAllPerguntas(id_projeto));
    req.setAttribute("id_candidato", new Integer(id_usuario));
    req.setAttribute("arrResposta", dao.getRespostasUsuario(id_projeto, id_usuario));
    ArrayList respostas = dao.getRespostasUsuario(id_projeto, id_usuario);
    for (Iterator iter = respostas.iterator(); iter.hasNext(); ) {
        RespostaBean element = (RespostaBean) iter.next();
        System.out.println("id_resposta no for : " + element.getArrOpcaoBean());
        RespostaOpcaoBean[] opcao = element.getArrOpcaoBean();
        for (int i = 0; i < opcao.length; i++) {
            System.out.println("opcao[" + i + "] no for : " + opcao[i].getValor());
        }
    }
    return next;
}
}

```

```

package br.com.perfil.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class MostrarCadastroProjeto implements Command {
    private String next;

```

```

public MostrarCadastroProjeto(String next){
    this.next = next;
}
public String execute(HttpServletRequest req, HttpServletResponse response)
throws CommandException {
    return next;
}

```

```

package br.com.perfil.bean;

public class AdjetivoBean {
    private String[] adjetivos;
    private int numAdjetivos;
    private String nome;
    private String status;
    private int quantidade;
    public AdjetivoBean(){ }
    public AdjetivoBean(int tamanho){
        adjetivos = new String[tamanho];
    }
    public int getNumAdjetivos() {
        return numAdjetivos;
    }
    public void setNumAdjetivos(int numAdjetivos) {
        this.numAdjetivos = numAdjetivos;
    }
    public String getAdjetivo(int indice){
        return adjetivos[indice];
    }
}

```

```

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public int getQuantidade() {
    return quantidade;
}

public void setQuantidade(int quantidade) {
    this.quantidade = quantidade;
}

public String getStatus() {
    return status;
}

public void setStatus(String status) {
    this.status = status;
}
}

package br.com.perfil.bean;

public class MatrizBean {
    private int[][] matriz;
    private int[] somatorio;
    private String[] nomesAdjetivos;
    private int tamanho;
}

```

```

public MatrizBean() {}

public void setElementAt(int linha, int coluna, int valor) {
    matriz[linha][coluna] = valor;
}

public String getTodosElementos() {
    String texto = "";
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz.length; j++) {
            texto = texto + matriz[i][j] + ",";
        }
    }
    System.out.println(texto);
    return texto;
}

/*
 * Dá o somatório de uma coluna
 */

public int getSomatorio(int coluna) {
    return somatorio[coluna];
}

public String getAdjColuna(int coluna) {
    return nomesAdjetivos[coluna];
}

public void setAdjColuna(String nome, int coluna) {
    nomesAdjetivos[coluna] = nome;
}

public String getTodoSomatorio() {

```

```

String todo = "";
for (int i = 0; i < somatorio.length; i++) {
    todo = todo + somatorio[i] + ",";
}
return todo;
}

public void someColunas() {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz.length; j++) {
            somatorio[i] = somatorio[i] + matriz[j][i];
        }
    }
}

public void espelharMatriz() {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz.length; j++) {
            if (i != j) {
                matriz[j][i] = matriz[i][j];
            }
        }
    }
}

public int[][] getMatriz() {
    return matriz;
}

public void setMatriz(int[][] matriz) {
    this.matriz = matriz;
}

```

```

    }

    public void setSomatorio(int[] somatorio) {
        this.somatorio = somatorio;
    }

    public int getTamanho() {
        return tamanho;
    }

    public int setTamanho(int tamanho) {
        return this.tamanho = tamanho;
    }

    /**
     * define o tamanho da Matriz e a instancia
     */

    public void setArray(int tamanho) {
        matriz = new int[tamanho][tamanho];
        somatorio = new int[tamanho];
        nomesAdjetivos = new String[tamanho];
        setTamanho(tamanho);
    }
}

```

```

package br.com.perfil.bean;

public class OpcaoBean {

    private int idOpcao;

    private String strOpcao;

    private int valorOpcao;

    private String respostaDescritiva;

    public int getIdOpcao() {

```

```

        return idOpcao;
    }
    public void setIdOpcao(int idOpcao) {
        this.idOpcao = idOpcao;
    }
    public String getStrOpcao() {
        return strOpcao;
    }
    public void setStrOpcao(String strOpcao) {
        this.strOpcao = strOpcao;
    }
    public int getValorOpcao() {
        return valorOpcao;
    }
    public void setValorOpcao(int valorOpcao) {
        this.valorOpcao = valorOpcao;
    }
    public String getRespostaDescritiva() {
        return respostaDescritiva;
    }
    public void setRespostaDescritiva(String respostaDescritiva) {
        this.respostaDescritiva = respostaDescritiva;
    }
}

```

```

package br.com.perfil.bean;

public class RespostaBean {

```

```

private int id_pergunta;

private RespostaOpcaoBean[] arrOpcaoBean;

public RespostaOpcaoBean[] getArrOpcaoBean() {

    return arrOpcaoBean;

}

public void setArrOpcaoBean(RespostaOpcaoBean[] arrOpcaoBean) {

    this.arrOpcaoBean = arrOpcaoBean;

}

public int getId_pergunta() {

    return id_pergunta;

}

public void setId_pergunta(int id_pergunta) {

    this.id_pergunta = id_pergunta;

}

}

```

```

package br.com.perfil.bean;

public class RespostaOpcaoBean {

    int id_opcao;

    String valor;

public int getId_opcao() {

    return id_opcao;

}

public void setId_opcao(int id_opcao) {

    this.id_opcao = id_opcao;

}

public String getValor() {

```

```
    return valor;
}
public void setValor(String valor) {
    this.valor = valor;
}
}
```

```
import java.sql.Date;
public class ProjetoBean {
    private int id_projeto;
    private String nome;
    private Date data;
    private Integer qtd_adjetivos;
    public ProjetoBean() {
    }
    public Date getData() {
        return data;
    }
    public void setData(Date data) {
        this.data = data;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

```

public int getId_projeto() {
    return id_projeto;
}
public void setId_projeto(int id_projeto) {
    this.id_projeto = id_projeto;
}
public Integer getQtd_adjetivos() {
    return qtd_adjetivos;
}
public void setQtd_adjetivos(Integer qtd_adjetivos) {
    this.qtd_adjetivos = qtd_adjetivos;
}
}

```

```

package br.com.perfil.bean;
public class PerguntaBean {
    private int idPergunta;
    private String strPergunta;
    private int tpPergunta;
    private OpcaoBean[] opcoes;
    public int getIdPergunta() {
        return idPergunta;
    }
    public void setIdPergunta(int idPergunta) {
        this.idPergunta = idPergunta;
    }
    public int getTpPergunta() {
        return tpPergunta;
    }
}

```

```

    }

    public void setTpPergunta(int tpPergunta) {
        this.tpPergunta = tpPergunta;
    }

    public OpcaoBean[] getOpcoes() {
        return opcoes;
    }

    public void setOpcoes(OpcaoBean[] opcoes) {
        this.opcoes = opcoes;
    }

    public String getStrPergunta() {
        return strPergunta;
    }

    public void setStrPergunta(String strPergunta) {
        this.strPergunta = strPergunta;
    }
}

package br.com.perfil.bean;

public class UsuarioBean {
    private String cpf;
    private String e_mail;
    private int id_projeto;
    private int id_usuario;
    private int idTipoUsuario;
    private String nome;
    int pontuacao;
    private String rg;
    private String senha;
}

```

```
private String usuario;

public UsuarioBean() {
}

public String getCpf() {
    return cpf;
}

public String getE_mail() {
    return e_mail;
}

public int getId_projeto() {
    return id_projeto;
}

public int getId_usuario() {
    return id_usuario;
}

public int getIdTipoUsuario() {
    return idTipoUsuario;
}

public String getNome() {
    return nome;
}

public int getPontuacao() {
    return pontuacao;
}

public String getRg() {
    return rg;
}

public String getSenha() {
```

```
        return senha;
    }

    public String getUsuario() {
        return usuario;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    public void setE_mail(String e_mail) {
        this.e_mail = e_mail;
    }

    public void setId_projeto(int id_projeto) {
        this.id_projeto = id_projeto;
    }

    public void setId_usuario(int id_usuario) {
        this.id_usuario = id_usuario;
    }

    public void setIdTipoUsuario(int idTipoUsuario) {
        this.idTipoUsuario = idTipoUsuario;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public void setPontuacao(int pontuacao) {
        this.pontuacao = pontuacao;
    }
}
```

```

public void setRg(String rg) {
    this.rg = rg;
}
public void setSenha(String senha) {
    this.senha = senha;
}
public void setUsuario(String usuario) {
    this.usuario = usuario;
}
public String toString(){
    return "id_projeto: " + id_projeto + " , id_usuario: " + id_usuario +
    " , nome: " + nome + " , senha: " + senha + " , e-mail: " + e_mail +
    " , idTipoUsuario: " + idTipoUsuario + " , rg: " + rg + " , cpf: " + cpf +
    " , pontuacao: " + pontuacao;
}
}

```

```

package br.com.perfil.servlet;
import java.io.IOException;
import java.util.HashMap;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import br.com.perfil.action.CadastrarCandidato;

```

```
import br.com.perfil.action.CadastrarPergunta;
import br.com.perfil.action.CadastrarQtdAdjetivos;
import br.com.perfil.action.CadastrarQuestionario;
import br.com.perfil.action.Command;
import br.com.perfil.action.CommandException;
import br.com.perfil.action.ExcluirPergunta;
import br.com.perfil.action.MenuSuperUsuario;
import br.com.perfil.action.MostrarCadastroCandidato;
import br.com.perfil.action.MostrarCadastroProjeto;
import br.com.perfil.action.MostrarCandidatos;
import br.com.perfil.action.MostrarFormGeraPergunta;
import br.com.perfil.action.MostrarFormQuestionario;
import br.com.perfil.action.MostrarFormResposta;
import br.com.perfil.action.MostrarProjetoFuncionario;
import br.com.perfil.action.MostrarProjetos;
import br.com.perfil.action.ProcessoAtualizaPontuacaoGeral;
import br.com.perfil.action.ProcessoCadastrarAdjetivo;
import br.com.perfil.action.ProcessoCadastrarProjeto;
import br.com.perfil.action.ProcessoCadastrarUsuario;
import br.com.perfil.action.ProcessoDefinirMatriz;
import br.com.perfil.action.ProcessoDefinirPerfil;
import br.com.perfil.action.ProcessoDescadastrarAdjetivo;
import br.com.perfil.action.ProcessoDescadastrarProjeto;
import br.com.perfil.action.ProcessoDescadastrarUsuario;
import br.com.perfil.action.ProcessoIncluirNaLista;
import br.com.perfil.action.ProcessoInserirAdjetivo;
import br.com.perfil.action.ProcessoLogin;
import br.com.perfil.action.ProcessoMatriz;
```

```

import br.com.perfil.action.ProcesseQuestaoDescritiva;
import br.com.perfil.action.ProcesseRejeitarAdjetivo;
import br.com.perfil.action.ProcesseSair;
import br.com.perfil.action.ProcesseSuperUsuario;
import br.com.perfil.action.ProcesseUsuario;

public class PerfilServlet extends HttpServlet {

    private HashMap comandos;

    private String error = "error.jsp";

    private static Logger log;

    public void init(ServletConfig config) throws ServletException {

        initCommands();

        log = Logger.getLogger(PerfilServlet.class);

        log.info("PerfilServlet inicializado...");

    }

    private void initCommands() {

        comandos.put("login", new ProcessoLogin("index.jsp"));

        comandos.put("projetos", new ProcessoProjetos("projetos.jsp"));

        comandos.put("cadastrarprojeto", new ProcessoCadastrarProjeto("servlet/perfil?cmd=mostrar_projetos"));

        comandos.put("descadastrarprojeto", new ProcessoDescadastrarProjeto("servlet/perfil?cmd=mostrar_projetos"));

        comandos.put("usuarios", new ProcessoSuperUsuario("controleDeUsuarios.jsp"));

        comandos.put("cadastrarusuario", new ProcessoCadastrarUsuario("controleDeUsuarios.jsp"));

        comandos.put("todososcadastrados", new ProcessoSuperUsuario("controleDeUsuarios.jsp"));

        comandos.put("descadastrarusuario", new ProcessoDescadastrarUsuario("controleDeUsuarios.jsp"));

        comandos.put("voltar", new ProcessoSuperUsuario("superUsuario.jsp"));

        comandos.put("cadastraradjetivo", new ProcessoCadastrarAdjetivo("controleDeAdjetivos.jsp"));

        comandos.put("descadastraradjetivo", new ProcessoDescadastrarAdjetivo("controleDeAdjetivos.jsp"));

        comandos.put("inseriradjetivo", new ProcessoInserirAdjetivo("selecionarAdjetivos.jsp"));

    }
}

```

```
comandos.put("rejeitaradjetivo", new ProcessseRejeitarAdjetivo("selecionarAdjetivos.jsp"));
comandos.put("matriz", new ProcessseDefinirMatriz("gerarMatriz.jsp"));
comandos.put("perfilpiramide", new ProcessseDefinirPerfil("definirPerfil.jsp"));
comandos.put("sair", new ProcessseSair("index.jsp"));

    //Coletar Preferencias Pessoais de um Perfil
comandos.put("preferencias", new ProcessseUsuario("preferencias.jsp"));
comandos.put("incluiralista", new ProcessseIncluirNaLista("usuario.jsp"));

    //forçar que o usuario responda
comandos.put("responder", new MostrarFormQuestionario(
"responderquestionario.jsp"));
comandos.put("cmdtipopergunta", new MostrarFormGeraPergunta(
    "geraPergunta.jsp"));
comandos.put("gerapergunta", new MostrarFormGeraPergunta(
    "geraPergunta.jsp"));
comandos.put("descreverpergunta", new MostrarFormGeraPergunta(
    "geraPergunta.jsp"));
comandos.put("mostrarquestionario", new MostrarFormQuestionario(
"questionario.jsp"));
comandos.put("mostrarresposta", new MostrarFormResposta(
"respostaUsuario.jsp"));
comandos.put("cadastrarpergunta", new CadastrarPergunta("geraPergunta.jsp"));
comandos.put("excluirpergunta", new ExcluirPergunta("questionario.jsp"));
comandos.put("processardescritiva", new ProcessseQuestaoDescritiva("respostaUsuario.jsp"));
comandos.put("atualizarpontuacao", new ProcessseAtualizaPontuacaoGeral("mostrarCandidatos.jsp"));
comandos.put("cadastrarquestionario", new CadastrarQuestionario(
    "sucesso.jsp"));
comandos.put("mostrar_projetos", new MostrarProjetos(
"mostrarProjetos.jsp"));
```

```

comandos.put("mostrar_cadastro_projeto", new MostrarCadastroProjeto(
"cadastrarProjeto.jsp"));
comandos.put("menu_super_usuario", new MenuSuperUsuario("superUsuario.jsp"));
comandos.put("mostrar_projeto_funcionario", new MostrarProjetoFuncionario("superUsuario.jsp"));
comandos.put("mostrar_projetos_candidatos", new MostrarProjetos("mostrarProjetosCandidato.jsp"));
comandos.put("mostrar_cadastro_candidato", new MostrarCadastroCandidato("cadastrarCandidato.jsp"));
comandos.put("cadastrar_candidato", new CadastrarCandidato("responderquestionario.jsp"));
comandos.put("mostrar_candidatos", new MostrarCandidatos("mostrarCandidatos.jsp"));
comandos.put("selecionar_adjetivos", new MostrarAdjetivos("selecionarAdjetivos.jsp"));
comandos.put("cadastrar_qtd_adjetivo", new CadastrarQtdAdjetivos("controleDeAdjetivos.jsp"));
}

protected void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    java.io.IOException {
response.setContentType("text/html");
java.io.PrintWriter out = response.getWriter();
out.close();
}

private Command lookupCommand(String cmd) throws CommandException {
System.out.println("COMANDO =" + cmd);
if (cmd == null)
    cmd = "main-menu";
if (comandos.containsKey(cmd)
{
// retorna o Command relativo a chave passada como argumento
} else
    throw new CommandException("Comando nao identificado! " + cmd);
}

```

```

public void service(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
    String next;
    Command cmd = null;
    HttpSession session = req.getSession();
    if(session == null){
        RequestDispatcher rd = getServletContext().getRequestDispatcher("error.jsp");
    }
    String strCmdParameter = req.getParameter("cmd");
    String strCmdAExecutar = strCmdParameter;
    try {
        if (strCmdParameter.equals("logar")) {
            String strLogarCorreto = (String) session
                .getAttribute("logar_correto");
            if (strLogarCorreto == null) {
                strCmdAExecutar = "f5_verificasessao";
            }
        }
        cmd = lookupCommand(strCmdAExecutar);
        next = cmd.execute(req, res);
    } catch (CommandException e) {
        System.out.println(e);
        req.setAttribute("javax.servlet.jsp.jspException", e);
        next = error;
    }
    RequestDispatcher rd = getRequestDispatcher(
        jspdir + next);
    System.out.println("next:"+next);
}

```

```

        if (next != "" && next != null) {
            rd.forward(req, res);
        } else {
        }
    }
}

```

```

package br.com.perfil.dao;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Types;
import java.util.ArrayList;
import java.util.Enumeration;
import org.apache.log4j.Logger;
import br.com.perfil.bean.AdjetivoBean;
import br.com.perfil.bean.OpcaoBean;
import br.com.perfil.bean.ProjetoBean;
import br.com.perfil.bean.RespostaOpcaoBean;
import br.com.perfil.bean.UsuarioBean;
import br.com.perfil.servlet.PerfilServlet;
public class PerfilDAO {
    private static Logger log;
    public static void main(String[] args) {
        log = Logger.getLogger(PerfilDAO.class);
    }
}

```

```

log.info("PerfilDAO inicializado...");
PerfilDAO bd = new PerfilDAO();
try {
    System.out.println("abriu "
        + bd.getUsuario("admin", "admin").toString());
} catch (Exception e) {
    e.printStackTrace();
}
}
Connection con;
String driver = "org.gjt.mm.mysql.Driver";
PreparedStatement pt;
Statement st;
String url = "jdbc:mysql://localhost:
public void abreConexao() {
    try {
        if (con == null) {
            System.out.println("abrindo conexao...");
            Class.forName(driver);

            System.out.println("carregando driver...");
            con = DriverManager.getConnection(url, usr, senha);
            System.out.println("pegando conexao1111..");
        }
    } catch (Exception e) {
        System.out.println("erro ao abrir conexao...usando usuario= " + usr
            + " senha=" + senha);
        System.out.println("erro ao abrir conexao..." + e.getMessage());
    }
}

```

```

        e.printStackTrace();
    }
}

public void decumenteContagemAdjetivo(int id_projeto, String nome) {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("update adjetivo set quantidade = quantidade - '1' where id_projeto = ? and nome = ? ");
        pt.setInt(1, id_projeto);
        pt.setString(2, nome);
        pt.execute();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }
}

public void excluaAdjetivo(int id_projeto, String nome) throws Exception {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("delete from adjetivo where id_projeto = ? and nome = ? ");
        pt.setInt(1, id_projeto);
    }
}

```

```

    pt.setString(2, nome);
    pt.execute();
    System.out.println("executou..." + pt);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
}

public void excluaFuncionario(int id_projeto, int id_funcionario)
    throws Exception {
try {
    abreConexao();
    System.out.println("abriu...");
    System.out.println(" createStatement..");
    PreparedStatement pt = con
        .prepareStatement("delete from funcionario_projeto where id_funcionario = ? and id_projeto = ?");
    pt.setInt(1, id_funcionario);
    pt.setInt(2, id_projeto);
    pt.execute();

    System.out.println("executou...");
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
}

```

```

    }
}
public void excluaProjeto(int id_projeto) throws Exception {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("delete from projeto where id_projeto = ? ");
        pt.setInt(1, id_projeto);
        pt.execute();

        PreparedStatement pt9= con("delete usuario,candidato_projeto FROM usuario,candidato_projeto WHERE
usuario.id_usuario = candidato_projeto.id_usuario AND candidato_projeto.id_projeto = ?");

        pt9.setInt(1, id_projeto);
        pt9.execute();

        PreparedStatement pt1 = con.prepareStatement("delete from funcionario_projeto where id_projeto = ? ");
        pt1.setInt(1, id_projeto);
        pt1.execute();

        PreparedStatement pt2 = con.prepareStatement("delete from candidato_projeto where id_projeto = ? ");
        pt2.setInt(1, id_projeto);
        pt2.execute();

        PreparedStatement pt3 = con
            .prepareStatement("delete from resposta_usuario where id_projeto = ? ");
        pt3.setInt(1, id_projeto);
        pt3.execute();

        PreparedStatement pt4 = con.prepareStatement("delete from adjetivo where id_projeto = ? ");
        pt4.setInt(1, id_projeto);
        pt4.execute();
    }
}

```

```

PreparedStatement pt5 = con .prepareStatement("delete from matriz where id_projeto = ? ");
pt5.setInt(1, id_projeto);
pt5.execute();

PreparedStatement pt6 = con .prepareStatement("delete from adjetivo_funcionario where id_projeto = ? ");
pt6.setInt(1, id_projeto);
pt6.execute();

PreparedStatement pt11= con
    .prepareStatement("delete projeto_pergunta, opcao, pergunta FROM projeto_pergunta, opcao, pergunta WHERE
opcao.id_pergunta = pergunta.id_pergunta AND projeto_pergunta.id_projeto = ?");

pt11.setInt(1, id_projeto);
pt11.execute();

PreparedStatement pt8 = con .prepareStatement("delete from projeto_pergunta where id_projeto = ? ");
pt8.setInt(1, id_projeto);
pt8.execute();

System.out.println("executou...");
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
}

public void excluaUsuario(int id_usuario) throws Exception {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
    }
}

```

```

        PreparedStatement pt = con
            .prepareStatement("delete from usuario where id_usuario = ? ");
        pt.setInt(1, id_usuario);
        System.out.println("executou...");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public void excluaPergunta(int id_projeto, int id_pergunta)
    Exception {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("delete from pergunta where id_pergunta = ? ");
        pt.setInt(1, id_pergunta);
        pt.execute();
        System.out.println("executou..." + pt);

        pt = con
            .prepareStatement("DELETE FROM projeto_pergunta where id_projeto = ? and id_pergunta = ? ");
        pt.setInt(1, id_projeto);
        pt.setInt(2, id_pergunta);
        pt.execute();
        System.out.println("executou..." + pt);
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public void excluaOpcoes(int id_pergunta) throws Exception {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("DELETE FROM opcao where id_pergunta = ? ");

        pt.setInt(1, id_pergunta);

        pt.execute();

        System.out.println("executou..." + pt);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }
}

public void excluirAdjetivoSelecFuncionario(int id_projeto,
    int id_funcionario, String adjetivo) {
    try {
        abreConexao();

```

```

System.out.println("abriu...");

System.out.println(" createStatement..");

PreparedStatement pt = con
    .prepareStatement("delete from adjetivo_funcionario where id_projeto = ? and id_funcionario = ? and nome = ? ");
pt.setInt(1, id_projeto);
pt.setInt(2, id_funcionario);
pt.setString(3, adjetivo);
pt.execute();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
this.decrementaContagemAdjetivo(id_projeto, adjetivo);
}

public void fechaConexao() {
    con.close();
    try {
        if (st != null) {
            st.close();
        }
        if (pt != null) {
            pt.close();
        }
        con = null;
        System.out.println("fechou conexao...");
    } catch (Exception e) {

```

```

        e.printStackTrace();
    }
}

public ArrayList getAllAdjetivos(int id_projeto) {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("select * adjetivo where id_projeto = ? ");

        pt.setInt(1, id_projeto);

        ResultSet rs = pt.executeQuery();

        System.out.println(" Executou...");

        ArrayList adjetivos = new ArrayList();

        while (rs.next()) {
            adjetivos.add(makeAdjetivoBean(rs));
        }

        System.out.println(" Pegou respostas..");

        return adjetivos;

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }

    return null;
}

```

```

public ArrayList getAllAdjetivosMaisCitados(int id_projeto) {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("select * from adjetivo where id_projeto = ? order by quantidade desc");

        pt.setInt(1, id_projeto);

        ResultSet rs = pt.executeQuery();

        System.out.println(" Executou..." + pt);

        ArrayList adjetivos = new ArrayList();

        while (rs.next()) {
            adjetivos.add(makeAdjetivoBean(rs));
        }

        System.out.println(" Pegou respostas..");

        return adjetivos;

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }

    return null;
}

public ArrayList getAllAdjetivosSelecionados(int id_projeto,
    int id_funcionario) {
    try {

```

```

abreConexao();

System.out.println("abriu...");

System.out.println(" createStatement..");

PreparedStatement pt = con
    .prepareStatement("select * from adjetivo_funcionario where id_projeto = ? and id_funcionario = ? ");

pt.setInt(1, id_projeto);

pt.setInt(2, id_funcionario);

ResultSet rs = pt.executeQuery();

System.out.println(" Executou...");

ArrayList adjetivosSelec = new ArrayList();

while (rs.next()) {

    AdjetivoBean adjetivo = new AdjetivoBean();

    String nome = rs.getString("nome");

    adjetivo.setNome(nome);

    adjetivosSelec.add(adjetivo);

}

System.out.println(" Pegou respostas..");

return adjetivosSelec;

} catch (Exception e) {

    e.printStackTrace();

} finally {

    fechaConexao();

    System.out.println("fechando conexao..");

}

return null;

}

public ArrayList getAllAdjetivosSelecionadosParaMatriz(int id_projeto) {

    try {

```

```

    abreConexao();

    System.out.println("abriu...");

    System.out.println(" createStatement..");

    PreparedStatement pt = con
        .prepareStatement("select X from adjetivo where id_projeto = ? and status = 'm' ");

    pt.setInt(1, id_projeto);

    ResultSet rs = pt.executeQuery();

    System.out.println(" Executou..." + pt);

    ArrayList adjetivosSelec = new ArrayList();

    while (rs.next()) {

        adjetivosSelec.add(makeAdjetivoBean(rs));

    }

    System.out.println(" Pegou respostas..");

    return adjetivosSelec;

} catch (Exception e) {

    e.printStackTrace();

} finally {

    fechaConexao();

    System.out.println("fechando conexao..");

}

return null;

}

public ArrayList getAllCandidatos(int id_projeto) {

    try {

        abreConexao();

        System.out.println(" abriu..");

        System.out.println(" createStatement..");

        PreparedStatement pt = con

```

```

        .prepareStatement("SELECT * FROM candidato_projeto cand LEFT JOIN usuario usr on (cand.id_usuario =
usr.id_usuario) WHERE cand.id_projeto = "
            + id_projeto + " order by pontuacao desc");

System.out.println(" pegando pergunta..");
ResultSet rs = pt.executeQuery();

ArrayList arrPerguntas = new ArrayList();
while (rs.next()) {
    arrPerguntas.add(makeUsuarioBean(rs));
}
System.out.println("pegou pergunta...");
pt.close();
return arrPerguntas;
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return null;
}

public ArrayList getAllFuncionarios(int id_projeto) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con

```

```

        .prepareStatement("select * from funcionario_projeto fun_p LEFT JOIN usuario usr on(fun_p.id_funcionario =
usr.id_usuario ) where id_projeto = ? order by id_usuario ");

        pt.setInt(1, id_projeto);

        ResultSet rs = pt.executeQuery();

        System.out.println(" Executou...");

        ArrayList usuarios = new ArrayList();

        while (rs.next()) {

            usuarios.add(makeUsuarioBeanLogin(rs));

        }

        System.out.println(" Pegou respostas..");

        return usuarios;

    } catch (Exception e) {

        e.printStackTrace();

    } finally {

        fechaConexao();

        System.out.println("fechando conexao..");

    }

    return null;

}

public OpcaoBean[] getAllOpcoes(int idPergunta) {

    abreConexao();

    System.out.println(" abriu..");

    String query = "SELECT * FROM opcao WHERE id_pergunta = "

        + idPergunta;

    System.out.println(" createStatement.." + pt);

    PreparedStatement pt = con.prepareStatement(query);

```

```

ResultSet rs = pt.executeQuery();

ArrayList arrOpcoes = new ArrayList();
while (rs.next()) {
    arrOpcoes.add(makeOpcaoBean(rs));
}
pt.close();

return (OpcaoBean[]) arrOpcoes.toArray(new OpcaoBean[arrOpcoes
    .size()]);
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

public ArrayList getAllPerfis(int idProjeto) {
    return null;
}

public ArrayList getAllPerguntas(int idProjeto) { //PEGA PELO PROJETO
    try {
        abreConexao();

        System.out.println(" abriu..");

        //String query = "SELECT * FROM pergunta ";

        System.out.println(" createStatement..");

        PreparedStatement pt = con.prepareStatement("SELECT * FROM pergunta perg LEFT JOIN projeto_pergunta proj on
(perg.id_pergunta = proj.id_pergunta) WHERE proj.id_projeto = "+ idProjeto);

        System.out.println(" pegando pergunta..");

        ResultSet rs = pt.executeQuery();

        ArrayList arrPerguntas = new ArrayList();

```

```

while (rs.next()) {
    arrPerguntas.add(rs);
    for (int i = 0; i < rs.getBlob("descricao").length(); i++) {
        System.out.print((rs.getString("descricao").charAt(i)));
    }
}
System.out.println("pegou pergunta...");
pt.close();
return arrPerguntas;
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return null;
}

```

```

public ArrayList getAllProjetos() {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from projeto");
        System.out.println(" Pegando projetos.");
        ResultSet rs = pt.executeQuery();
        System.out.println(" Executou...");
    }
}

```

```

ArrayList projetos = new ArrayList();
while (rs.next()) {
    projetos.add(makeProjetoBean(rs));
}
System.out.println(" Pegou respostas..");
return projetos;

} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return null;
}

public ArrayList getAllUsuarios(int id_projeto) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from usuario where id_projeto = ? order by id_usuario ");
        pt.setInt(1, id_projeto);
        ResultSet rs = pt.executeQuery();
        System.out.println(" Executou...");
        while (rs.next()) {
            usuarios.add(makeUsuarioBean(rs));
        }
    }
}

```

```

        System.out.println(" Pegou respostas..");
        return usuarios;

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return null;
}

public float getDesvioPadraoColMatriz(int id_projeto, int coluna) {
    float dp = 0.f;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select STDDEV(valor) as desvioPadrao from matriz where id_projeto = ? and coluna = ? ");
        pt.setInt(1, id_projeto);
        pt.setInt(2, coluna);
        ResultSet rs = pt.executeQuery();
        while (rs.next()) {
            dp = rs.getFloat("desvioPadrao");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {

```

```

        fechaConexao();

        System.out.println("fechando conexao..");
    }
    return dp;
}

public int getIDdoUsuario(int id_projeto, String nome, String senha) {
    int id_usuario = -1;

    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("select * from Usuario where id_projeto = ? and nome = ? and senha = ? ");

        pt.setInt(1, id_projeto);

        pt.setString(2, nome);

        pt.setString(3, senha);

        ResultSet rs = pt.executeQuery();

        while (rs.next()) {
            id_usuario = rs.getInt("id_usuario");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }

    return id_usuario;
}

```

```

public float getMediaColMatriz(int id_projeto, int coluna) throws Exception {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("select (valor) as media from matriz where id_projeto = ? and coluna = ? ");

        pt.setInt(1, id_projeto);

        pt.setInt(2, coluna);

        ResultSet rs = pt.executeQuery();

        while (rs.next()) {
            media = rs.getFloat("media");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }

    return media;
}

public String getNomeProjeto(int id_projeto) {
    String nome = null;

    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con

```

```

        .prepareStatement("select * from Projeto where id_projeto = ? ");
pt.setInt(1, id_projeto);
ResultSet rs = pt.executeQuery();
while (rs.next()) {
    nome = rs.getString("nome_projeto");
}
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return nome;
}

public ProjetoBean getProjeto(int id_projeto) {
    ProjetoBean projeto = null;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from projeto where id_projeto = ? ");
        pt.setInt(1, id_projeto);
        ResultSet rs = pt.executeQuery();
        while (rs) {
            projeto = makeProjetoBean(rs);
        }
    } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return projeto;
}

public int getProjetoFuncionario(int id_usuario) {

    try {
        abreConexao();
        System.out.println(" abriu..");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("SELECT * FROM funcionario_projeto cand where id_funcionario = "
                + id_usuario);
        System.out.println(" pegando pergunta..");
        ResultSet rs = pt.executeQuery();
        rs.next();
        return rs.getInt("id_projeto");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return -1;
}

```

```

private RespostaOpcaoBean[] getRespostaOpcao(int id_pergunta,
    int id_projeto, int id_usuario) {
    try {
        PreparedStatement pt = con
            .prepareStatement("Select id_opcao,valor from resposta_usuario WHERE id_pergunta="
                + id_pergunta
                + " AND id_projeto ="
                + id_projeto
                + " AND id_usuario=" + id_usuario);

        ResultSet rs = pt.executeQuery();
        System.out.println(" Pegou resposta..." + pt);
        ArrayList arrRespostaOpcao = new ArrayList();
        while (rs.next()) {
            arrRespostaOpcao.add(makeRespostaOpcaoBean(rs));
        }
        pt.close();
        return (RespostaOpcaoBean[]) arrRespostaOpcao
            .toArray(new RespostaOpcaoBean[arrRespostaOpcao.size()]);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

public ArrayList getRespostasUsuario(int id_projeto, int id_usuario) {
    try {
        abreConexao();
        System.out.println(" abriu..");
        //String query = "INSERT INTO pergunta values(null,'teste2',0)";
    }
}

```

```

System.out.println(" createStatement..");

String query = "Select DISTINCT id_pergunta from resposta_usuario WHERE id_projeto = "
    + id_projeto + " AND id_usuario = " + id_usuario + " ORDER BY id_pergunta";

PreparedStatement pt = con.prepareStatement(query);

System.out.println(" antes Pegou resposta..");

ResultSet rs = pt.executeQuery();//67, 68

System.out.println(" Pegou resposta.." + pt);

ArrayList respostas = new ArrayList();

while (rs.next()) {

    respostas.add(makeRespostaBean(rs, id_projeto, id_usuario));

}

System.out.println(" Pegou respostasss..");

pt.close();

System.out.println(" Respostas size: " + respostas.size());

return respostas;

} catch (Exception e) {

    e.printStackTrace();

} finally {

    fechaConexao();

    System.out.println("fechando conexao..");

}

return null;

}

public UsuarioBean getUsuario(String usuario, String senha) {

    try {

        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

```

```

PreparedStatement pt = con
    .prepareStatement("select * from usuario where usuario = ? and senha = ?");
pt.setString(1, usuario);
pt.setString(2, senha);
ResultSet rs = pt.executeQuery();
UsuarioBean usuarioBean = null;
if (rs.next()) {
    usuarioBean = makeUsuarioBeanLogin(rs);
}
return usuarioBean;
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return null;
}

public void incrementeContagemAdjetivo(int id_projeto, String nome) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("update adjetivo set quantidade = quantidade + '1' where id_projeto = ? and nome = ? ");
        pt.setInt(1, id_projeto);
        pt.setString(2, nome);
        pt.execute();
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public void inserePontuacao(int id_projeto, int id_usuario, int pontuacao) {
    try {
        abreConexao();
        PreparedStatement pt = con
            .prepareStatement("UPDATE candidato_projeto SET pontuacao="
                + pontuacao + " WHERE id_projeto = " + id_projeto
                + " AND id_usuario=" + id_usuario);

        pt.execute();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public void addPontuacaoDescritiva(int id_projeto, int id_usuario, double mediaDaPontuacao) {
    try {
        abreConexao();
        PreparedStatement pt = con
            .prepareStatement("candidato_projeto SET pontuacao = pontuacao + "
                + mediaDaPontuacao + " WHERE id_projeto = " + id_projeto

```

```

        + " AND id_usuario=" + id_usuario);

    pt.execute();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();

    System.out.println("fechando conexao..");
}
}

public void inserePontuacaoDescritiva(int id_projeto, int id_usuario, int id_pergunta, int valor) {
    int num = this.getNumeroDeAvaliacoes(id_projeto, id_usuario, id_pergunta);
    String consulta = "UPDATE valor_descritiva SET valor = valor + "
        + valor + " WHERE id_projeto = " + id_projeto
        + " AND id_usuario=" + id_usuario
        + " AND id_pergunta=" + id_pergunta;
    if(num == 0){
        consulta = "INSERT INTO valor_descritiva (id_projeto, id_usuario, id_pergunta, valor) VALUES("+
id_projeto+", "+id_usuario+", "+id_pergunta+", "+valor+")";
    }
    try {
        abreConexao();

        PreparedStatement pt = con
            .prepareStatement(consulta);

        System.out.println(pt);

        pt.execute();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {

```

```

        fechaConexao();

        System.out.println("fechando conexao..");
    }
}

public void incrementaNumAvaliacoes(int id_projeto, int id_usuario, int id_pergunta) {

    try {
        abreConexao();

        PreparedStatement pt = con
            .prepareStatement("UPDATE valor_descritiva SET num_avaliadores = num_avaliadores + 1 "
                + " WHERE id_projeto = " + id_projeto
                + " AND id_usuario=" + id_usuario
                + " AND id_pergunta=" + id_pergunta);

        pt.execute();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }
}

public int getPontuacaoDescritiva(int id_projeto, int id_usuario, int id_pergunta) {

    int pontuacao = 0;

    try {
        abreConexao();

        PreparedStatement pt = con
            .prepareStatement("SELECT valor FROM valor_descritiva"
                + " WHERE id_projeto = " + id_projeto

```

```

        + " AND id_usuario=" + id_usuario
        + " AND id_pergunta=" + id_pergunta);
ResultSet rs = pt.executeQuery();
while (rs.next()) {
    pontuacao = rs.getInt("valor");
}
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return pontuacao;
}

public int getNumeroDeAvaliacoes(int id_projeto, int id_usuario, int id_pergunta) {
    int avaliacoes = 0;
    try {
        abreConexao();
        PreparedStatement pt = con
            .prepareStatement("SELECT num_avaliadores FROM valor_descritiva"
                + " WHERE id_projeto = " + id_projeto
                + " AND id_usuario=" + id_usuario
                + " AND id_pergunta=" + id_pergunta);
        ResultSet rs = pt.executeQuery();
        while (rs.next()) {
            avaliacoes = rs.getInt("num_avaliadores");
        }
    } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return avaliacoes ;
}

public double getMediaQuestaoDescritiva(int id_projeto, int id_usuario, int id_pergunta) {
    double media = 0.0;
    double somatorio = this.getPontuacaoDescritiva(id_projeto, id_usuario, id_pergunta);
    double avaliacoes = this.getNumeroDeAvaliacoes(id_projeto, id_usuario, id_pergunta);
    if((avaliacoes != 0) && (somatorio != 0)){
        media = somatorio / avaliacoes;
    }
    System.out.println("media no BD= " + media);
    return Math.ceil(media);
}

public void inserirAdjetivo(int id_projeto, String nome, int count,
    String status) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("insert into adjetivo values (?, ?, ?, ?)");
        pt.setInt(1, id_projeto);
        pt.setString(2, nome);
        pt.setInt(3, count);
    }
}

```

```

    pt.setString(4, status);

    pt.execute();

    System.out.println("executou...");

} catch (Exception e) {

    e.printStackTrace();

} finally {

    fechaConexao();

    System.out.println("fechando conexao..");

}

}

public void inserirAdjetivoSelecFuncionario(int id_projeto,

    int id_funcionario, String nome, String status) {

try {

    abreConexao();

    System.out.println("abriu...");

    System.out.println(" createStatement..");

    PreparedStatement pt = con

        .prepareStatement("insert into adjetivo_funcionario values (?, ?, ?)");

    pt.setInt(1, id_projeto);

    pt.setInt(2, id_funcionario);

    pt.setString(3, nome);

    pt.setString(4, status);

    pt.execute();

    System.out.println("setStatusSelecAdjetivo: " + pt);

    log.info("setStatusSelecAdjetivo: " + pt);

} catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    this.incrementeContagemAdjetivo(id_projeto, nome);
}

public void inserirCandidato(int id_projeto, int idUsuario, int pontuacao) {
    try {
        PreparedStatement pt = con
            .prepareStatement("update candidato_projeto set pontuacao = ? where id_projeto = ? and id_usuario = ? ");
        pt.setInt(1, pontuacao);
        pt.setInt(2, id_projeto);
        pt.setInt(3, idUsuario);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public int inserirCandidato(int id_projeto, String nome, String e_mail,
    String rg, String cpf) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con

```

```

        .prepareStatement("insert      into      usuario(usuario,e_mail,rg,cpf,id_tipo_usuario)      values
('candidato','candidato',?,?,?,2)");
        pt.setString(1, nome);
        pt.setString(2, e_mail);
        pt.setString(3, rg);
        pt.setString(4, cpf);
        pt.execute();

        System.out.println("executou...");

        ResultSet rs = pt.executeQuery("select  max(id_usuario) as id_usuario from usuario");
        rs.next();

        int id_usuario = rs.getInt("id_usuario");

        pt = con.prepareStatement("insert into candidato_projeto(id_projeto,id_usuario) values (?,?)");
        pt.setInt(1, id_projeto);
        pt.setInt(2, id_usuario);
        pt.execute();

        System.out.println("executou...");

        return id_usuario;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }

    return -1;
}

public void inserirFuncionario(int id_projeto, String usuario,
        String senha, String nome, String e_mail, String cpf, String rg) {    try {
        abreConexao();

```

```

System.out.println("abriu...");
System.out.println(" createStatement..");
PreparedStatement pt = con
    .prepareStatement("insert into usuario(id_tipo_usuario,usuario,senha,nome,e_mail,cpf, rg) values (1,?,?,?,?);");
pt.setString(1, usuario);
pt.setString(2, senha);
pt.setString(3, nome);
pt.setString(4, e_mail);
pt.setString(5, cpf);
pt.setString(6, rg);
pt.execute();
System.out.println("executou..." + pt);
int id_usuario = getUsuario(usuario, senha).getId_usuario();
System.out.println("inserir no projeto.");
abreConexao();
pt = con
    .prepareStatement("insert into funcionario_projeto values (?,?)");

pt.setInt(1, id_projeto);
pt.setInt(2, id_usuario);
pt.execute();
System.out.println("executou...");

} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}

```

```

    }
}

public void inserirOpcoes(int id_pergunta, String descr, int valor) {
    try {
        abreConexao();

        System.out.println(" abriu..");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("INSERT INTO opcao values(?,null,?,?)");

        pt.setObject(1, "" + id_pergunta);
        pt.setObject(2, descr);
        pt.setObject(3, "" + valor);

        System.out.println(" inseriu opcao..");

        pt.execute();

        pt.close();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();

        System.out.println("fechando conexao..");
    }
}

public int inserirPergunta(String descrPergunta, int tipoPergunta,
    int idProjeto) {
    try {
        abreConexao();

        System.out.println(" abriu..");

        System.out.println(" createStatement..");

```

```

    pt = con.prepareStatement("INSERT INTO pergunta values(null,?,?)");
    pt.setObject(1, descrPergunta);
    pt.setObject(2, "" + tipoPergunta);
    pt.execute();
    System.out.println("executou..");
    ResultSet rs = pt.executeQuery("SELECT      max(id_pergunta) from pergunta");
    rs.next();
    System.out.println("max(id_pergunta) "+ rs.getInt("max(id_pergunta)"));
    int id_pergunta = rs.getInt("max(id_pergunta)");
    pt = con.prepareStatement("INSERT INTO projeto_pergunta values(?,?)");
    pt.setInt(1, idProjeto);
    pt.setInt(2, id_pergunta);
    pt.execute();
    return id_pergunta;
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();
    System.out.println("fechando conexao..");
}
return -1;
}

public void inserirProjeto(String nome, Date data) {
    if (!this.procurarNomeDeProjeto(nome)) {
        try {
            abreConexao();
            System.out.println("abriu...");
            System.out.println(" createStatement..");

```

```

        PreparedStatement pt = con.prepareStatement("insert into projeto values (null,?,?,null
pt.setString(1, nome);
pt.setDate(2, data);
pt.execute();
System.out.println("executou...");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
} else {
    System.out.println("Projeto j□ cadastrado!");
}
}

public void inserirQtdAdjProjeto(int id_projeto, int qtd) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("Update projeto set min_adjetivos = ? where id_projeto = "
                + id_projeto
pt.setInt(1, qtd);

        pt.execute();
        System.out.println("executou...");
    } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

```

//somente para pergunta multivalorada

```

public void inserirRespostaUsuario(int idProjeto, int idUsuario,
    int idPergunta, int idOpcao, String strResposta) {
    try {
        abreConexao();
        System.out.println(" abriu..");
        //String query = "INSERT INTO pergunta values(null,'teste2',0)";
        System.out.println(" createStatement..");
        PreparedStatement pt = con.prepareStatement("INSERT INTO resposta_usuario values(?,?,?,?,?)");
        pt.setInt(1, idProjeto);
        pt.setInt(2, idUsuario);
        pt.setInt(3, idPergunta);
        if (idOpcao != -1) {
            pt.setInt(4, idOpcao);
        } else {
            pt.setNull(4, Types.INTEGER);
        }
        pt.setObject(5, strResposta);
        System.out.println(" inseriu resposta..");
        pt.execute();
        pt.close();
    } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
}

public void inserirResultPerfil(int idUsuario, int idProjeto,int resultFinal) {

}

public void inserirResultUsuario(int idUsuario, int idPergunta, String valor) {
}

public void inserirSomaColuna(int id_projeto, int id_usuario, int coluna,
    int valor) {
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con.prepareStatement("insert into matriz values (?,?,?,?)");
        pt.setInt(1, id_projeto);
        pt.setInt(2, id_usuario);
        pt.setInt(3, coluna);
        pt.setInt(4, valor);
        pt.execute();
        System.out.println("inseriu no BD a soma da coluna..");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
    }
}

```

```

        System.out.println("fechando conexao..");
    }
}

public void inserirUsuario(int id_projeto, String nome, String senha,
    String e_mail) {
    if (!this.procurarNomeDeUsuario(id_projeto, nome)) {
        try {
            abreConexao();
            System.out.println("abriu...");
            System.out.println(" createStatement..");
            PreparedStatement pt = con.prepareStatement("insert into usuario values (?,null,?,?)");
            pt.setInt(1, id_projeto);
            pt.setString(2, nome);
            pt.setString(3, senha);
            pt.setString(4, e_mail);
            pt.execute();
            System.out.println("executou...");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            fechaConexao();
            System.out.println("fechando conexao..");
        }
    } else {
        System.out.println("Usuário já cadastrado nesse projeto!");
    }
}

private AdjetivoBean makeAdjetivoBean(ResultSet results) {

```

```

try {
    AdjetivoBean adjetivo = new AdjetivoBean();
    String nome = results.getString("nome");
    int quantidade = results.getInt("quantidade");
    adjetivo.setNome(nome);
    adjetivo.setQuantidade(quantidade);
    return adjetivo;
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

private OpcaoBean makeOpcaoBean(ResultSet results) {
    try {
        System.out.println("pegando opcao");
        OpcaoBean opcao = new OpcaoBean();
        System.out.println("opcao1 " + results.getInt("id_opcao"));
        opcao.setIdOpcao(results.getInt("id_opcao"));
        System.out.println("opcao2 " + results.getString("descricao"));
        opcao.setStrOpcao(results.getString("descricao"));
        System.out.println("opcao3 " + results.getInt("valor"));
        opcao.setValorOpcao(results.getInt("valor"));
        return opcao;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

```

```

private PerguntaBean makePerguntaBean(ResultSet results) {
    try {
        PerguntaBean pergunta = new PerguntaBean();
        int id_pergunta = results.getInt("id_pergunta");
        pergunta.setIdPergunta(id_pergunta);
        pergunta.setStrPergunta(results.getString("descricao"));
        System.out.println("String da pergunta BD: "+ results.getString("descricao"));
        int tipo = results.getInt("tipo");
        if (!(tipo == 3)) {
            pergunta.setOpcoes(getAllOpcoes(id_pergunta));
        }
        pergunta.setTpPergunta(tipo);
        return pergunta;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private ProjetoBean makeProjetoBean(ResultSet results) {
    try {
        ProjetoBean projeto = new ProjetoBean();
        projeto.setNome(results.getString("nome_projeto"));
        projeto.setData(results.getDate("data_projeto"));
        projeto.setId_projeto(results.getInt("id_projeto"));
        if (results.getInt("min_adjetivos") != 0) {
            projeto.setQtd_adjetivos(new Integer(results.getInt("min_adjetivos")));
        }
        return projeto;
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private RespostaBean makeRespostaBean(ResultSet results, int id_projeto,
    int id_usuario) {
    try {
        RespostaBean pergunta = new RespostaBean();
        int id_pergunta = results.getInt("id_pergunta");
        pergunta.setId_pergunta(id_pergunta);
        pergunta.setArrOpcaoBean(getRespostaOpcao(id_pergunta, id_projeto,
            id_usuario));

        return pergunta;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private RespostaOpcaoBean makeRespostaOpcaoBean(ResultSet results) {
    try {
        System.out.println("pegando opcao");
        RespostaOpcaoBean opcao = new RespostaOpcaoBean();
        int id_opcao = results.getInt("id_opcao");
        opcao.setId_opcao(results.getInt("id_opcao"));
        opcao.setValor(results.getString("valor"));
        return opcao;
    }
}

```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private UsuarioBean makeUsuarioBean(ResultSet results) {
    try {
        UsuarioBean usuario = new UsuarioBean();
        usuario.setId_usuario(results.getInt("id_usuario"));
        usuario.setIdTipoUsuario(results.getInt("id_tipo_usuario"));
        usuario.setUsuario(results.getString("usuario"));
        usuario.setSenha(results.getString("senha"));
        usuario.setNome(results.getString("nome"));
        usuario.setE_mail(results.getString("e_mail"));
        usuario.setCpf(results.getString("cpf"));
        usuario.setRg(results.getString("rg"));
        usuario.setPontuacao(results.getInt("pontuacao"));
        return usuario;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

private UsuarioBean makeUsuarioBeanLogin(ResultSet results) {
    try {
        UsuarioBean usuario = new UsuarioBean();
        usuario.setId_usuario(results.getInt("id_usuario"));
        usuario.setIdTipoUsuario(results.getInt("id_tipo_usuario"));

```

```

    usuario.setUsuario(results.getString("usuario"));
    usuario.setSenha(results.getString("senha"));
    usuario.setNome(results.getString("nome"));
    usuario.setE_mail(results.getString("e_mail"));
    usuario.setCpf(results.getString("cpf"));
    usuario.setRg(results.getString("rg"));

    return usuario;
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

```

```

public boolean procureFuncionario(int id_projeto, String nome) {
    boolean consulta = false;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from usuario where id_projeto = ? and nome = ? ");
        pt.setInt(1, id_projeto);
        pt.setString(2, nome);
        ResultSet rs = pt.executeQuery();
        if (rs.next()) {
            consulta = true;
        }
    } catch (Exception e) {

```

```

        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return consulta;
}

public boolean procureNomeDeProjeto(String nome) {
    boolean consulta = false;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from projeto where nome_projeto = ? ");
        pt.setString(1, nome);
        ResultSet rs = pt.executeQuery();
        if (rs.next()) {
            consulta = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return consulta;
}

```

```

public boolean procureNomeDeUsuario(int id_projeto, String nome) {
    boolean consulta = false;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
        PreparedStatement pt = con
            .prepareStatement("select * from usuario where id_projeto = ? and nome = ? ");
        pt.setInt(1, id_projeto);
        pt.setString(2, nome);
        ResultSet rs = pt.executeQuery();
        if (rs.next()) {
            consulta = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        fechaConexao();
        System.out.println("fechando conexao..");
    }
    return consulta;
}

```

```

public boolean procureUsuario(int id_projeto, String usuario, String senha) {
    boolean consulta = false;
    try {
        abreConexao();
        System.out.println("abriu...");
        System.out.println(" createStatement..");
    }
}

```

```

//procura se existe um usuario com o mesmo login e senha no mesmo projeto
PreparedStatement pt = con
    .prepareStatement("select * from usuario join funcionario_projeto where id_funcionario = id_usuario and
id_projeto = ? and usuario = ? and senha = ?");

pt.setInt(1, id_projeto);
pt.setString(2, usuario);
pt.setString(3, senha);

System.out.println(pt);

ResultSet rs = pt.executeQuery();

if (rs.next()) {
    consulta = true;
}

} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();

    System.out.println("fechando conexao..");
}

return consulta;
}

public void setStatusSelecParaMatrizAdjetivo(int id_projeto, String nome) {
    try {
        abreConexao();

        System.out.println("abriu...");

        System.out.println(" createStatement..");

        PreparedStatement pt = con
            .prepareStatement("update adjetivo set status = 'm' where id_projeto = ? and nome = ? ");

        pt.setInt(1, id_projeto);

```

```

    pt.setString(2, nome);

    pt.execute();

    System.out.println("Executou..." + pt);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    fechaConexao();

    System.out.println("fechando conexao..");
}
}
}

```

CadastrarCandidato.jsp

```

<%@ page language="java" %>

<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">

<html>

<head>

<title>Cadastro de Candidato</title>

<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>

<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>

<script >

function fnValideCampos(){

    if(fnEhVazio(document.forms[0].nomeInsercao)){

        return false;

    }

    if(fnEhVazio(document.forms[0].cpf)){

        return false;

    }
}

```

```

if(fnEhVazio(document.forms[0].rg)){
    return false;
}
if(fnEhVazio(document.forms[0].e_mail)){
    return false;
}

return fnAcessaServlet("cadastrar_candidato","<%= request.getContextPath() %>/servlet/perfil");
}
</script>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Candidato, forne&ccedil;a os dados solicitados:</h2>

<form name="formPerfil" method="post">
<table border="1">
<tr bgcolor="yellow"><th>Projeto</th><th>Nome</th><th>CPF</th><th>RG</th><th>E-mail</th></tr>
<tr><td><%=br.com.perfil.bean.ProjetoBean
                                projetoBean
                                =
(br.com.perfil.bean.ProjetoBean)session.getAttribute("projetoBean");%>
<td><input readonly type="text" name="id_projeto" value="<%=projetoBean.getNome()%>"></td>
<td><input type="text" name="nomeInsercao" value="" size="20"></td>
<td><input type="text" name="cpf" value="0" size="20"></td>
<td><input type="text" name="rg" value="0" size="20"></td>
<td><input type="text" name="e_mail" value="" size="20"></td>
</tr>
</table>
<table>
<tr>

```

```

        <td>&nbsp;</td>
    </tr>
</table>
<input type="hidden" name="cmd" size="20" value="" >
    <input type="button" name="ok" size="20" value=" Cadastrar " onclick='fnValideCampos();>
</form>
</body>
</html>

```

CadastrarProjeto.jsp

```

<%@ page import="java.util.ArrayList, br.com.perfil.dao.PerfilDAO,br.com.perfil.bean.ProjetoBean"%>
<html>
<head>
<title>Cadastrar Novo Projeto</title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
<script >
function fnValideCampos(){
    //alert(fnEhVazio(document.forms[0].nome_projeto));
    if(fnEhVazio(document.forms[0].nome_projeto)){
        return false;
    }
    return fnAcessaServlet("cadastrarprojeto","<%= request.getContextPath() %>/servlet/perfil");
}
</script>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Cadastrar Novo Projeto</h2>

```

```

<form name="formPerfil" method="post">
<table border="1">
  <tr bgcolor="yellow"><th>Nome do Projeto</th><th>Data</th></tr>
  <tr>
    <td><input type="text" name="nome_projeto" value="" size="40"></td>
    <%java.text.SimpleDateFormat formatter = new java.text.SimpleDateFormat("dd/MM/yyyy");
    %>
    <td><input type="text" name="data_projeto" value="<%=formatter.format(
new
java.util.GregorianCalendar().getTime())%>" size="10" readonly></td>
  </tr>
</table>
<table>
<tr>&nbsp;</tr>
</table>
<input type="hidden" name="cmd" value="" size="">
<input type="button" name="ok" size="20" value="Cadastrar Novo Projeto" onclick='fnValideCampos();'>
<input type="button" name="voltar" size="10" value="Voltar" onclick='fnAcessaServlet("mostrar_projetos","<%=
request.getContextPath() %>/servlet/perfil");'>
<input type="button" name="sair" size="10" value="Sair" onclick='fnAcessaServlet("sair","<%= request.getContextPath()
%>/servlet/perfil");'>
</center>
</form>
</body>
</html>

```

controleDeAdjetivos.jsp

```

<%@page import = "java.util.Enumeration"%>
<%@ page import="java.util.ArrayList,br.com.perfil.bean.AdjetivoBean,br.com.perfil.dao.PerfilDAO"%>

```

```

<%PerfilDAO oBD= new PerfilDAO();%>

<html>

<script src="<%= request.getContextPath()%>/javascript/js-geral/geral.js"></script>

<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>

<head>

<title> Controle de Atributos Desejados </title>

<script >

function fnValideCampos(){

    if(fnEhVazio(document.forms[0].qtd_adjetivos)){

        return false;

    }

    return fnAcessaServlet("cadastrar_qtd_adjetivo","<%= request.getContextPath() %>/servlet/perfil");

}

</script>

</head>

<body BGCOLOR="=#AAAAFF">

<center>

<h2>Controle de Atributos Desejados</h2>

<form name="formPsico" method="post">

<% br.com.perfil.bean.ProjetoBean projetoBean = (br.com.perfil.bean.ProjetoBean)session.getAttribute("projetoBean");

    String aux = null;

    if(projetoBean.getQtd_adjetivos() == null){

        aux = "";

    }else{

        aux = projetoBean.getQtd_adjetivos().toString();

    }

%>

```

```

<input type="hidden" name="cmd" value="">
<table border="1">
  <tr bgcolor="yellow"><th>Atributo</th><th>Minimo de adjetivos a serem selecionados:</th></tr>
  <tr align="center">
    <td><input type="text" name="adjetivo" value="" size="35"></td>
    <td><input type="text" name="qtd_adjetivos" align="center" value="<%=aux%>" size="45"></td>
  </tr>
  <tr align="center">
    <td><input type="button" name="ok" value="Incluir no Espaco Amostral"
onclick='fnAcessaServlet("cadastraradjetivo","<%= request.getContextPath() %>/servlet/perfil");'></td>
    <td><input type="button" name="ok" value=" OK " onclick='fnValideCampos();'></td>
  </tr>
</table>
<br>
<input name="voltar" type='button' size='10' value=" Voltar " onClick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");'></td>
</center>
<h2>Espaco Amostral:</h2><p>
<%
int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());
ArrayList arrAdjetivos = oBD.getAllAdjetivos(id_projeto);%>
<%
AdjetivoBean adjetivoBean = null;
for (int i = 0; i < arrAdjetivos.size(); i++) {
  adjetivoBean = (AdjetivoBean)arrAdjetivos.get(i);
  String adjetivo = adjetivoBean.getNome();%>
  <INPUT TYPE="CHECKBOX" NAME="adjetivos" VALUE="<%= adjetivo %>"> <%= adjetivo %><P>
<% }%>

```

```

<input type="button" name="ok" size="20" value="Descadastrar Adjetivo"
onclick='fnAcessaServlet("descadastraradjetivo","<%= request.getContextPath() %>/servlet/perfil");'>
</form>
</body>
</html>

```

controleDeUsuarios.jsp

```

<%@ page import="java.util.ArrayList,br.com.perfil.bean.UsuarioBean,
br.com.perfil.dao.PerfilDAO,br.com.perfil.bean.ProjetoBean"%>
<html>
<head>
<title>Controle de Usu&aacute;rios</title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
<script >
function fnValideCampos(){
//alert(fnEhVazio(document.forms[0].nome_projeto));
if(fnEhVazio(document.forms[0].nomeInsercao)){
return false;
}
if(fnEhVazio(document.forms[0].usuario)){
return false;
}
if(fnEhVazio(document.forms[0].senha)){
return false;
}
if(fnEhVazio(document.forms[0].e_mail)){
return false;
}
}

```

```

}
if(fnEhVazio(document.forms[0].cpf)){
    return false;
}
if(fnEhVazio(document.forms[0].rg)){
    return false;
}
return fnAcessaServlet("cadastrarusuario","<%= request.getContextPath() %>/servlet/perfil");
}
function fnValideDescadastrar(){
if(document.formPerfil.id_funcionario.checked){
    var size = 1;
    return fnAcessaServlet("descadastrarusuario","<%= request.getContextPath() %>/servlet/perfil");
}else{
    var size = document.formPerfil.id_funcionario.length;
    for(var i =0;i < size;i++){
        if(document.formPerfil.id_funcionario[i].checked){
            return fnAcessaServlet("descadastrarusuario","<%= request.getContextPath() %>/servlet/perfil");
        }
    }
}
    alert("Você deve selecionar pelo menos um Funcionário!");
}
</script>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Controle de Usu&acuteros</h2>

```

```

<form name="formPerfil" method="post">
<input type="hidden" name="cmd" size="20" value="" >
<table border="1">
  <tr
        bgcolor="yellow"><th>Projeto</th><th>Nome</th><th>Login</th><th>Senha</th><th>E-
mail</th><th>CPF</th><th>RG</th></tr>
  <tr>
        <td><input
                readonly
                type="text"
                name="id_projeto"
value="<%=((ProjetoBean)session.getAttribute("projetoBean")).getNome()%>" ></td>
        <td><input type="text" name="nomeInsercao" value="" size="25"></td>
        <td><input type="text" name="usuario" value="" size="10"></td>
        <td><input type="text" name="senha" size="10"></td>
        <td><input type="text" name="e_mail" value="" size="15"></td>
        <td><input type="text" name="cpf" value="" size="15"></td>
        <td><input type="text" name="rg" value="" size="15"></td>
  </tr>
</table>
<table>
  <tr>
        <td>&nbsp;</td>
  </tr>
</table>
<input type="button" name="cadastrarusuario" size="20" value="Cadastrar Usu&acut;rio" onclick='fnValideCampos();>

<!--<input
        type="button"
        name="cmd"
        size="20"
        value="todososcadastrados"
onclick='fnAcessaServlet("todososcadastrados","<%= request.getContextPath() %>/servlet/perfil");>-->
</table>
<table>
  <tr>

```

```

        <td>&nbsp;</td>
    </tr>
</table>
<% if(request.getAttribute("msg") != null ){%>
<font color="red">
<h2><%= request.getAttribute("msg")%></h2>
<META HTTP-EQUIV="REFRESH" CONTENT="2; URL=<%= request.getContextPath()
%/>/servlet/perfil?cmd=todosocadastrados">
<%=}%>
<%
    ArrayList arrUsuarios = (ArrayList)request.getAttribute("arrUsuarios");
if((arrUsuarios != null) && !(arrUsuarios.isEmpty())){%>
<table border="1">
<tr bgcolor="yellow"><th>Op&ccedil;&atilde;o</th>
<th>Nome</th>
<th>Login</th>
<th>Senha</th>
<th>E-mail</th>
<th>CPF</th>
<th>RG</th><tr>
<%=
    UsuarioBean usuarioBean = null;
    for (int i = 0; i < arrUsuarios.size(); i++) {
    usuarioBean = (UsuarioBean)arrUsuarios.get(i);%>
        <tr><td align="center"><INPUT TYPE="radio" NAME="id_funcionario"
value="<%=usuarioBean.getId_usuario()%>"></td>
        <td><%=usuarioBean.getNome()%></td>
        <td><%=usuarioBean.getUsuario()%></td>
        <td><%=usuarioBean.getSenha()%></td>

```

```

                <td><%=usuarioBean.getE_mail()%></td>
                <td><%=usuarioBean.getCpf()%></td>
                <td><%=usuarioBean.getRg()%></td>
            </tr>
        <%=}%>
</table>

<table>
    <tr>
        <td>&nbsp;</td>
    </tr>
</table>

<input type="button" name="descadastrarusuario" size="20" value="Descadastrar usu&aacute;rio"
onclick='fnValideDescadastrar();'>
<%=}%>

<input type="button" name="voltar" size="20" value="Voltar" onclick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");'>
</center>
</form>
</body>
</html>

```

definirPerfil.jsp

```

<%=@ page import="br.com.perfil.dao.PerfilDAO,br.com.perfil.bean.AdjetivoBean, java.util.ArrayList"%>
<%=PerfilDAO oBD= new PerfilDAO();%>
<%=! String adjetivo; %>
<%=! int numeroDeAdjetivos; %>

```

```

<%! String n[]; %>
<%! float media, desvioP, resultado; %>
<html>
<head>
<title> Controle de Atributos Desejados </title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Definindo o Perfil</h2>
</center>
<form name="formPsico" method="post">
<center>
<table border="1">
<tr bgcolor="yellow"><th>Adjetivo</th><th>Média</th><th>Desvio Padrão</th><th>Diferen&ccedil;a</th></tr>
<% int id_projeto = Integer.parseInt(session.getAttribute("id_projeto").toString());
    ArrayList arrAdjetivosMatriz = oBD.getAllAdjetivosSelecionadosParaMatriz(id_projeto);
    for (int i = 0; i < arrAdjetivosMatriz.size(); i++) {
        AdjetivoBean adj = (AdjetivoBean)arrAdjetivosMatriz.get(i);
        media = oBD.getMediaColMatriz(id_projeto,i);
        desvioP= oBD.getDesvioPadraoColMatriz(id_projeto,i);
        resultado = media - desvioP; %>
        <tr><td><%=adj.getNome()%></td><td><%= media %></td><td align="center"><%= desvioP%></td><td><%=
resultado%></td>
<% } %>
</table>
<p>

```

```

<input type="hidden" name="cmd" value="" >
<input name="voltar" type='button' size='10' value="Voltar" onClick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");'></td>
</center>
</form>
</body>
</html>

```

geraPergunta.jsp

```

<%@ page language="java" %>
<%
String descreverpergunta = (String) request.getAttribute("descreverpergunta");
String cabecalho = "";
String cabecalhoMulti = "<th>N&uacute;mero de alternativas na Resposta</th>";
String titulo = "Elaborar Quest&atilde;o";

if(descreverpergunta == null || descreverpergunta.trim().equals("")){
    titulo = "Elaborar Question&aacute;rio";
}
%>

<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title><%=titulo%></title>

<script src="<%= request.getContextPath()%>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>

```



```

<td>
    <select name="tipoPergunta" onchange='fnAcessaServlet("cmdtipopergunta","<%= request.getContextPath()
%/>/servlet/perfil");'>
        <option value="0" <%=tp.equals("0")?"selected":""%>>Multipla Escolha</option>
        <option value="1" <%=tp.equals("1")?"selected":""%>>Única Escolha</option>
        <option value="2" <%=tp.equals("2")?"selected":""%>>Multivalorada</option>
        <option value="3" <%=tp.equals("3")?"selected":""%>>Descritiva</option>
    </select>
</td>

<%
if( !(tp.trim().equals("3")) ){%> <!-- quando diferente de Descritiva deve aparecer o numero de alternativas-->
    <td>
        <input name="numOpcoes" type='text' value="2" size='40' >
    </td>
</tr>
<table>
    <tr>
        <td>&nbsp;</td>
    <tr>
    <tr>
        <td><input name="gerar" type='button' size='10' value="Gerar"
onClick='fnValidaGeraOpcoes("descreverpergunta","<%= request.getContextPath() %>/servlet/perfil");'>
        <input name="voltar" type='button' size='10' value="Voltar"
onClick='fnAcessaServlet("voltar","<%= request.getContextPath() %>/servlet/perfil");'></td>
    </tr>
</table>
<%}else{%>

```

```

</tr>
<table>
  <tr>
    <td>&nbsp;</td>
  <tr>
  <tr>
    <td><input      name="gerar"      type='button'      size='10'      value="Gerar"
onClick='fnAcessaServlet("descreverpergunta","<%= request.getContextPath() %>/servlet/perfil");'>
      <input      name="voltar"      type='button'      size='10'      value="Voltar"
onClick='fnAcessaServlet("voltar","<%= request.getContextPath() %>/servlet/perfil");'></td>
    </tr>
  </table>
  <%=}%>

```

```

<%=>// fim if(strNumOpcoes == null
else{
%>
<tr>
  <input name ="tipoPergunta" type='hidden' value='<%= (String)request.getAttribute("tipoPergunta") %>'></input>
  <td>Pergunta:</td>
  <td colspan=4><input name="descrPergunta" type='text' size='80' ></td>
  <!-- <td>valor da Pergunta</td>-->
  <!-- <td><input name ="valorPerg" type='text' size='10' ></td>-->
</tr>
<%=
String strNumOpcoes =(String) request.getAttribute("numOpcoes");
if(strNumOpcoes != null){

```

```

int intNumOpcoes = Integer.parseInt(strNumOpcoes);

for(int i =0;i < intNumOpcoes;i++){
%>

<tr>

<td>Opção <%=i%>:</td>

<td><input name="descrOpcao" type='text' size='50' value=""></td>

<% String readonly = "";

String valor = "";

String type = "text";

if(tp.equals("2")){

    readonly = "readonly";

    valor = "0";

    type = "hidden";

}else{ %>

<td>Valor da Opção <%=i%></td>

<%} %>

<td><input <%=readonly %> type = '<%=type%>' name ="valorOpcao" size='10' value ='<%=valor%>'

></td>

</tr>

<%

}

%>

<table>

<tr>

<td>&nbsp;</td>

</tr>

```

```

<tr><!--fnValidaCadastroPergunta-->
    <td><input          name="cadastrar"          type='button'          size='10'          value="Cadastrar"
onClick='fnValidaCadastroPergunta("cadastrarpergunta","<%=
request.getContextPath()
%>/servlet/perfil","<%=strNumOpcoes%>");'></td>
</tr>
</table>
<%
} // fim if(strNumOpcoes != null){
else{
%>
    <table>
<tr>
    <td><input          name="cadastrar"          type='button'          size='10'          value="Cadastrar"
onClick='fnAcessaServlet("cadastrarpergunta","<%=
request.getContextPath()
%>/servlet/perfil","<%=strNumOpcoes%>");'></td>
</tr>
</table>
<% }
}%>
</table>
</form>
</center>

</body>
</html>

```

gerarMatriz.jsp

```

<%@ page import="java.util.ArrayList,br.com.perfil.bean.AdjetivoBean"%>

```

```

<html>
<head>
<title> Atributos mais citados para o projeto </title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>

<script>
function fnValideGerarMatriz(){

    var size = document.formPerfil.checkMatriz.length;

    for(var i =0;i < size;i++){
        if(document.formPerfil.checkMatriz[i].checked){
            return fnAcessaServlet("gerarmatriz", "<%= request.getContextPath() %>/servlet/perfil");
        }
    }
    alert('Você deve selecionar pelo menos um adjetivo!');
}
</script>

</head>
<body BGCOLOR="#AAAFFF">
<center>
<h2>Atributos mais citados para o projeto:</h2>
<form name="formPerfil" method="post">
    <% ArrayList arrAdjetivos = (ArrayList)request.getAttribute("adjetivos");%>
<table border="1">
    <tr bgcolor="yellow"><th>Opção</th><th>Adjetivo</th><th>Ocorrência</th><tr>

```

```

<% AdjetivoBean adjetivoBean = null;

for (int i = 0; i < arrAdjetivos.size(); i++) {

    adjetivoBean = (AdjetivoBean)arrAdjetivos.get(i);

    String adjetivo = adjetivoBean.getNome();

    int quantidade = adjetivoBean.getQuantidade();%>

    <tr
        align="center"><td><INPUT
            TYPE="CHECKBOX"
            NAME="checkMatriz"
value="<%=adjetivo%>"></td><br><td><%=adjetivo%></td><td><%=quantidade%></td></tr>

</table>

<input type="hidden" name="cmd" value="" >

<p><input type="button" name="gerarmatriz" size="20" value=" Gerar Matriz " onclick=fnValideGerarMatriz();>

<input name="Voltar" type='button' size='10' value="Voltar"
onClick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");></td>

</form>

</center>

</body>

</html>

```

index.jsp

```

<html>

<head>

<title> Bem Vindo! </title>

<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>

<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>

</head>

<center>

<body BGCOLOR="#01b7e6">

<h2> EMPRESA </h2>

```

```

<IMG SRC="<%= request.getContextPath()%>/logo.gif">
<h2>Autentica&ccedil;&atilde;o para Administrador e Colaboradores</h2>
<form name="formPerfil" method="post">
<input type="hidden" name="cmd" value="" >
<table border="1">
  <tr bgcolor="yellow"><th>Login</th><th>Senha</th></tr>
  <tr>
    <td><input type="text" name="usuario" size="20" value="admin"></td>
    <td><input type="password" name="senha" size="10" value="admin"></td>
  </tr>
</table>
<table>
  <tr>
    <td>&nbsp;</td>
  </tr>
</table>
<input type="button" name="entrar" size="10" value="Entrar" onclick='fnAcessaServlet("login","<%=
request.getContextPath() %>/servlet/perfil");'>
<p><h2><a href='<%= request.getContextPath() %>/servlet/perfil?cmd=mostrar_projetos_candidatos'>Candidato clique
aqui</a></h2>
<p><h2><a href='<%= request.getContextPath() %>/help/help.html'>Informa&ccedil;&otilde;es</a></h2>
</form>
<%
  if(session.getAttribute("login") != null){
    if(session.getAttribute("login") == "invalido"){>
      <%@include file="loginInvalido.html" %>
      <META HTTP-EQUIV="REFRESH" CONTENT="1; URL=<%= request.getContextPath()
%>/index.jsp">
      <%session.invalidate();

```

```

    }
}
</center>
</body>
</html>

```

mostrarCandidatos.jsp

```

<%@ page import="java.util.ArrayList,br.com.perfil.bean.UsuarioBean, br.com.perfil.dao.PerfilDAO"%>
<html>
<head>
<title>Resultados do Teste</title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Pontuação dos Candidatos</h2>
<form name="formPerfil" method="post">
<table border="1">
<tr bgcolor="yellow"><th>Op&ccedil;ao</th><th>Nome do Candidato</th><th>RG</th><th>CPF</th><th>E-
mail</th><th>Pontuacao</th></tr>
<%
        ArrayList arrUsuarios = (ArrayList)request.getAttribute("arrCandidatos");
if(arrUsuarios != null){
        UsuarioBean usuarioBean = null;
        for (int i = 0; i < arrUsuarios.size(); i++) {
        usuarioBean = (UsuarioBean)arrUsuarios.get(i);%>
                <tr><td align=center><INPUT TYPE="radio" NAME="candidato"
value="<%=usuarioBean.getId_usuario()%>" <%= (i==0?"checked":"")%>></td>

```

```

        <td><%=usuarioBean.getNome()%></td>
        <td><%=usuarioBean.getRg()%></td>
        <td><%=usuarioBean.getCpf()%></td>
        <td><%=usuarioBean.getE_mail()%></td>
        <td><%=usuarioBean.getPontuacao()%></td>
    <% }
} %>
</table>
<table>
    <tr>
        <td>&nbsp;</td>
    </tr>
</table>
<input type="button" name="ok" size="20" value="Mostrar Respostas" onclick='fnAcessaServlet("mostrarresposta","<%=
request.getContextPath() %>/servlet/perfil");'>
<input type="button" name="voltar" size="10" value="Voltar" onclick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");'>
<input type="button" name="sair" size="10" value="Sair" onclick='fnAcessaServlet("sair","<%= request.getContextPath()
%>/servlet/perfil");'>
</center>
</form>
</body>
</html>

```

mostrarProjetos.jsp

```

<%@ page language="java" %>
<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">

```

```

<html>
<head>
<title>Perfil - Controle de Projetos</title>
<script src="<%= request.getContextPath()%>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
<%@ page import="java.util.ArrayList, br.com.perfil.bean.UsuarioBean, br.com.perfil.bean.ProjetoBean"%>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2>Controle de Projetos</h2>
<form name="formPerfil" method="post">
<%
    ArrayList arrProjetos = (ArrayList) request.getAttribute("arrProjetos");
    //out.print(arrProjetos);
%>
<input type="hidden" name="cmd" value="" >
<table border=1>
    <%
        if(arrProjetos != null && arrProjetos.size() > 0){
    %>
        <tr bgcolor="yellow" align="center">
            <th>Projetos:</th>
            <th><select name="selectProjetos" >
                <%
                    int size = arrProjetos.size();
                    ProjetoBean projetoBean = null;
                    for(int i =0 ; i < size;i++){
                        projetoBean = (ProjetoBean) arrProjetos.get(i);

```

```

        %>
        <option value ='<%=projetoBean.getId_projeto()%>' ><%=projetoBean.getNome()%></option>
        <%>// fecha for %>
    </select>
</th>
</tr>
</table>
<table>
    <tr>
        <td>&nbsp;</td>
    </tr>
</table>
<table>
    <tr align="center">
        <tr>
            <td><input    name="novo"    type='button'    size='10'    value="Novo    Projeto"
onClick='fnAcessaServlet("mostrar_cadastro_projeto","<%= request.getContextPath() %>/servlet/perfil");'></td>
            <td><input    name="OK"    type='button'    size='10'    value="    OK    "
onClick='fnAcessaServlet("menu_super_usuario","<%= request.getContextPath() %>/servlet/perfil");'></td>
            <td><input    name="excluir"    type='button'    size='10'    value="Excluir"
onClick='fnAcessaServlet("descadastrarprojeto","<%= request.getContextPath() %>/servlet/perfil");'></td>
        </tr>
    </table>
<%}else{//fecha if %>
    <tr>
        <td><input    name="novo"    type='button'    size='10'    value="Novo    Projeto"
onClick='fnAcessaServlet("mostrar_cadastro_projeto","<%= request.getContextPath() %>/servlet/perfil");'></td>

```

```

        </tr>
    </table>
    <% } %>
</center>
</body>
</html>

```

preferencias.jsp

```

<%@ page import="java.util.ArrayList,br.com.perfil.bean.AdjetivoBean, br.com.perfil.dao.PerfilDAO"%>
<%PerfilDAO oBD= new PerfilDAO();
    oBD.abreConexao(); %>
<html>
<head>
<title> Atributos Desejados para <--%= session.getAttribute("nome_projeto")%-->XXXXXX </title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
</head>
<body>
<center>
<h2>Atributos Desejados para o cargo de XXXXXX</h2>
</center>
<form name="formPerfil" method="post">
<h2>Espaço Amostral:</h2><p>
<p>Escolha X atributos</p>
<% int id_projeto = Integer.parseInt(session.getAttribute(("id_projeto")).toString());%>

```

```

<%
    ArrayList arrAdjetivosSelec = oBD.getAllAdjetivosSelecionados(id_projeto);

    AdjetivoBean adjetivoBean = null;

    for (int i = 0; i < arrAdjetivosSelec.size(); i++) {
        adjetivoBean = (AdjetivoBean)arrAdjetivosSelec.get(i);

        String adjetivo = adjetivoBean.getNome();%>

        <INPUT TYPE="CHECKBOX" NAME="<%= adjetivo %>" VALUE="<%= adjetivo %>"> <%= adjetivo %><P>

    <%}%>

<input type="hidden" name="cmd" value="" >
<input type="button" name="incluir" size="10" value="incluirnaalista" onclick="fnAcessaServlet("incluirnaalista","<%=
request.getContextPath() %>/servlet/perfil");">
</form>
</body>
</html>

```

questionario.jsp

```

<%@ page language="java" %>
<%@ page import="java.util.ArrayList,br.com.perfil.bean.PerguntaBean, br.com.perfil.bean.OpcaoBean"%>
<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Question&aacute;rio</title>
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
</head>
<body BGCOLOR="#AAAAFF">
<center>

```

```

<h2>Question&aacute;rio:</h2>
</center>
<% ArrayList arrPerguntas =(ArrayList) session.getAttribute("arrayPerguntas");
<form name="formPsico" method="post">
<input type="hidden" name="cmd" value="" >
    <%
        PerguntaBean perguntaBean = null;%>
        <h4>Numero de perguntas: <%= arrPerguntas.size()%></h4>
        <%for (int i = 0; i < arrPerguntas.size(); i++) {
            perguntaBean = (PerguntaBean)arrPerguntas.get(i);
        %>
        <table >
            <tr>
                <td><%= i + 1 + " " + perguntaBean.getStrPergunta()%></td>
            </tr>
        </table>
        <table>
            <%
                OpcaoBean[] arrOpcaoBean = perguntaBean.getOpcoes();%>
            <%if(arrOpcaoBean != null){
                for (int j = 0; j < arrOpcaoBean.length; j++) {
                    %>
                <tr>
                    <%
                        switch (perguntaBean.getTpPergunta()) {
                            case 0://multipla escolha %>
                                <td><input type='checkbox' name='<%= "pergunta" + i%>' value="<%= "" + j%>"
                    /></td>

```

```

        <%break;

        case 1: // unica escolha %>

        <td><input type='radio' name="<%= "pergunta" + i%>" value="<%= ""+j%>"

/></td>

        <%break;

        case 2: //multivalorada %>

        <td ><input type='text' name='<%= "pergunta" + i%>' size="5" value="" /></td>

        <% }//fim switch%>

        <td><%= arrOpcaoBean[j].getStrOpcao()%></td>

    </tr>

    <% }//fecha for arrOpcaoBean %>

</table>

    <% }else{//pergunta para o usuario descrever %>

    <tr>

    <td><TEXTAREA COLS=40 ROWS=5 NAME='<%= "pergunta" + i%>'> Deixe seu comentário </TEXTAREA>

    </td>

    </tr>

    <% }%>

</table>

    <tr>

        <input type="hidden" name="tipoPergunta" value="<%= perguntaBean.getTpPergunta()%>" >

        <%-- <input type="hidden" name="id_pergunta" value="<%= perguntaBean.getIdPergunta()%>" > --%>

        <td><input          name="excluir"          type='button'          size='10'          value="Excluir"

onClick='fnAcessaServlet("excluirpergunta", "<%= request.getContextPath() %>/servlet/perfil?id_pergunta=<%= perguntaBean.getIdPergunta()%>");'></td>

    </tr>

```

```

        </table>

        <hr size="2">

        <% } %>

<table>

        <tr>

                <td> <input name="voltar" type='button' size='10' value="Voltar" onClick='fnAcessaServlet("voltar","<%=
request.getContextPath() %>/servlet/perfil");></td>

                </tr>

</form>

</body>

</html>

```

respostaUsuario.jsp

```

<%@ page language="java" %>

<%@      page      import="java.util.ArrayList,br.com.perfil.dao.PerfilDAO,          br.com.perfil.bean.PerguntaBean,
br.com.perfil.bean.OpcaoBean,br.com.perfil.bean.RespostaBean,br.com.perfil.bean.RespostaOpcaoBean" %>

<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">

<html>

<head>

<title>Respostas do Candidato</title>

<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js" ></script>

<script src="<%= request.getContextPath() %>/funcoesPerfil.js" ></script>

<script >

function getValorNota(campo){

    var valor = document.getElementById("valor_nota" + campo).value;

    var id = campo.value;

    alert("o valor eh "+ valor);

```

```

fnAcessaServlet("processardescritiva", "<%=
                                                                    request.getContextPath()
%>/servlet/perfil?id_pergunta=id&valor_nota=valor");

return valor;

}

</script>

</head>

<body BGCOLOR="#AAAAFF">

<center>

<h2>Respostas do Candidato</h2>

</center>

<% ArrayList arrPerguntas =(ArrayList) session.getAttribute("arrayPerguntas");

Integer id_projeto = new Integer(session.getAttribute("id_projeto").toString());

Integer id_candidato =(Integer)request.getAttribute("id_candidato");

ArrayList arrResposta =(ArrayList)request.getAttribute("arrResposta");

out.println("Resposta size: " + arrResposta.size());

out.println("id_candidato: " + id_candidato.intValue());

out.println("Perguntas size: " + arrPerguntas.size());

String output = "";

PerfilDAO bd = new PerfilDAO();

//bd.abreConexao();

%>

<form name="formPsico" method="post">

<input type="hidden" name="cmd" value="" >

PerguntaBean perguntaBean = null;%>

<h4>Numero de perguntas: <%=arrPerguntas.size()%></h4>

<%RespostaBean respostaBean = null;

```

```

//sobre as perguntas e respostas
for (int i = 0; i < arrPerguntas.size(); i++) {
    perguntaBean = (PerguntaBean)arrPerguntas.get(i);
    respostaBean = (RespostaBean)arrResposta.get(i);
%>
<table >
<tr>
<td><%= i + 1 + " " + perguntaBean.getStrPergunta()%>:</td>
</tr>
</table>
<table>
<%
OpcaoBean[] arrOpcaoBean = perguntaBean.getOpcoes();
RespostaOpcaoBean[] arrRespostaOpcao = respostaBean.getArrOpcaoBean();
if(arrOpcaoBean != null){
    //sobre as opcoes da pergunta
    for (int j = 0; j < arrOpcaoBean.length; j++) {
%>
<tr>
<td></td>
<%
switch (perguntaBean.getTpPergunta()) {
    case 0://multipla escolha
        String strChek = "";
        String bgcolor = "";
        String valor = " 0 ";

```

```

for (int x = 0; x < arrRespostaOpcao.length; x++) {
    if(arrRespostaOpcao[x].getId_opcao()
arrOpcaoBean[j].getIdOpcao() ){
        strChek = "checked";
        bgcolor = "bgcolor = yellow";
        valor = arrOpcaoBean[j].getValorOpcao() + " ";

    }
}

output = output + valor;

%>
<td <%=bgcolor%> ><input type='checkbox' name='<%= "pergunta" + i%>'
value="<%= "" + j%>" <%=strChek%>/></td><td><%= "(" + arrOpcaoBean[j].getValorOpcao() + ")" %></td>
<%=break;
case 1: // unica escolha
String strSel = "";
String bg = "";

    if(arrRespostaOpcao[0].getId_opcao()
arrOpcaoBean[j].getIdOpcao() ){
        strSel = "checked";
        bg = "bgcolor = yellow";
        output = output + arrOpcaoBean[j].getValorOpcao() + " ";
    }
}

%>
<td <%=bg%>><input type='radio' name="<%= "pergunta" + i%>"
value="<%= "" + j%>" <%=strSel%> /></td><td><%= "(" + arrOpcaoBean[j].getValorOpcao() + ")" %></td>

```

```

        <%break;

        case 2: //multivalorada

        String strVal = "";

        String bgcor = "";

        for (int y = 0; y < arrRespostaOpcao.length; y++) {

                if(arrRespostaOpcao[y].getId_opcao()

                ==

arrOpcaoBean[j].getIdOpcao() ){

                strVal = arrRespostaOpcao[y].getValor();

                bgcor= "bgcolor = yellow";

                output = output + arrRespostaOpcao[y].getValor() + " ";

                }

                }

        %>

        <td <%=bgcor%> ><input type='text' name='<%= "pergunta" + i%>' size="5"

value="<%=strVal%>" /></td><td><%= "("+arrOpcaoBean[j].getValorOpcao()+")"%></td>

        <% } //fim switch%>

        <td><%= arrOpcaoBean[j].getStrOpcao()%></td>

</tr>

<% } //fecha for arrOpcaoBean %>

</table>

<% } else { //pergunta para o usuario descrever %>

<tr>

<td><%=out.println("Perguntas size: " + arrRespostaOpcao.length);%></td>

<td><TEXTAREA                COLS=80                ROWS=5                NAME='<%= "pergunta"                +

i%>'><%=arrRespostaOpcao[0].getValor()%></TEXTAREA>

```

```

        </td>
    </tr>
<table>
<tr>
        <td>Núm.Avaliações:<input
                readonly="true"
                type="text"
                size="5"
value="<%=bd.getNumeroDeAvaliacoes(id_projeto.intValue(),id_candidato.intValue(),perguntaBean.getIdPergunta())%>"></td>
        <td width=5>&nbsp;</td>
        <td>Nota(0-100):<input type="text" name="valor_nota<%=perguntaBean.getIdPergunta()%>"
size="5" value=""></td>
        <td width=5>&nbsp;</td>
        <td>Somatório:<input
                type="text"
                readonly="true"
                size="5"
value="<%=bd.getPontuacaoDescritiva(id_projeto.intValue(),id_candidato.intValue(),perguntaBean.getIdPergunta())%>"></td>
        <td width=5>&nbsp;</td>
        <td>Média:<input
                type="text"
                size="5"
                readonly="true"
name="media<%=perguntaBean.getIdPergunta()%>"
value="<%=bd.getMediaQuestaoDescritiva(id_projeto.intValue(),id_candidato.intValue(),perguntaBean.getIdPergunta())%>"
></td>
        <td width=5>&nbsp;</td>
        <input type="hidden" name="id_candidato" value="<%= id_candidato.intValue()%>" >
        <!-- <input type="text" name="id_pergunta" value="<%= perguntaBean.getIdPergunta()%>" >
-->
        <td><input
                name="processar"
                type='button'
                size='10'
                value="Processar"
onClick='fnAcessaServlet("processardescritiva","<%= request.getContextPath() %>/servlet/perfil?id_pergunta=<%= perguntaBean.getIdPergunta()%>");'></td>
    </tr>
</table>

```

```

<% } %>
</table>
<hr size="2">
<% } %>
<td> <input name="voltar" type='button' size='10' value="Voltar" onClick='fnAcessaServlet("mostrar_candidatos","<%=
request.getContextPath() %>/servlet/perfil");'></td>
<td> <input name="voltar" type='button' size='10' value="Atualizar Pontuação Geral"
onClick='fnAcessaServlet("atualizarpontuacao","<%= request.getContextPath() %>/servlet/perfil");'></td>
</form>
<table>
<tr>
<td><%=//out.println(output);%></td>
</tr>
</table>
</body>
</html>

```

sucesso.jsp

```

<html>
<head>
<title> Página de Sucesso </title>
</head>
<body BGCOLOR="#AAAAFF">
<center>
<h2> Operação Realizada com sucesso! </h2>
</center>
<%=session.invalidate();%>

```

```
<META HTTP-EQUIV="REFRESH" CONTENT="1; URL=<%= request.getContextPath() %>/index.jsp">
```

```
</body>
```

```
</html>
```

superUsuario.jsp

```
</jsp:include page="autenticador.jsp" flush="true" />
```

```
<html>
```

```
<head>
```

```
<title>Opções do Administrador</title>
```

```
<script src="<%= request.getContextPath() %>/javascript/js-geral/geral.js"></script>
```

```
<script src="<%= request.getContextPath() %>/funcoesPerfil.js"></script>
```

```
</head>
```

```
<body BGCOLOR="#AAAAFF">
```

```
<center>
```

```
<h2>Administrador, escolha uma das opções abaixo:</h2>
```

```
<input type="hidden" name="cmd" value="">
```

```
<table>
```

```
<tr><td><input type="radio" name="opcao" value="usuarios" checked> Controle de Usuários</td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="adjetivos"> Controle de Atributos Desejados</td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="matriz"> Computar Atributos / Gerar Matriz Semântica
```

```
</td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="perfilpiramide"> Definir Perfil </td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="gerapergunta"> Elaborar Questionário</td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="mostrarquestionario"> Visualizar Questionário</td></tr>
```

```
<tr><td><input type="radio" name="opcao" value="mostrar_candidatos"> Visualizar Resultados do Teste</td></tr>
```

```
<input type="button" name="ok" size="10" value="Processar" onclick='fnProcessarSelecao("<%= request.getContextPath() %>/servlet/perfil");'>
```

```
<input type="button" name="sair" size="10" value=" Sair " onclick='fnAcessaServlet("sair","<%= request.getContextPath()
```

```
%>/servlet/perfil");'>
```

```
</center>
```

```
</body>
```

```
</html>
```

B-CÓDIGO FONTE RNA

```
%normalizando a entrada entre -1 e +1
EntradaN = premnmx(Entrada);
%EntradaN = premnmx(Param);
%transpondo a entrada
EntradaN = EntradaN';
%Criando a rede (((NET)))...
% limites dos padrões de entrada [min(EntradaN)' max(EntradaN)' ]
% número de neurônios da camada competitiva [10 10]
% Topologia: Função Camada Hexagonal 'hextop'
% learnsom is the self-organizing map weight learning function.
% dist = Euclidean distance weight function.
% mandist = Manhattan distance weight function.
net = newsom( [min(EntradaN)' max(EntradaN)' ], [10 10], 'hextop');
net.trainParam.show = 200;
% numero de epocas para o treinamento
net.trainParam.epochs = 1000;
%treinando a rede (escolhe vetores aleatórios da entrada)
net = train(net,EntradaN);
%simulando a rede (simula de acordo com a ordem da EntradaN, ou seja, sequencialmente)
Saida = sim(net,EntradaN);
%neuronios vencedores (joga para SaidaC os numeros dos neuronios q tiveram saida 1)
SaidaC = vec2ind(Saida);
%atribuindo o numero de candidatos
numCandidatos = size(EntradaN,2);
%carregando os pesos
for i= 1:numCandidatos
    %arredonda para inteiro menor
```

```

Y(i) = floor(SaidaC(i)/10);

%resto da divisao de SaidaC(i)/10

X(i) = rem(SaidaC(i),10);

end

%habilita / desabilita exibição de gráficos na mesma janela.

hold on

%habilita / desabilita grid do gráfico

grid on

%nome do eixo x

xlabel('Algarismo da unidade do neuronio vencedor');

%nome do eixo y

ylabel('Algarismo da dezena do neuronio vencedor');

%titulo do gráfico

title('Classificacao efetuada pela Rede');

%altera limites do gráfico (escala)

axis([0 10 0 10]);

for i =1:numCandidatos

    plot(X(1,i),Y(1,i),'bo');

    text(X(1,i),Y(1,i),num2str(i),'FontSize',12);

end

```

C-SUGESTÕES

- Mudar a cor de fundo das páginas do software (azul);
- Tornar a interface mais amigável;
- Modificar as questões, pois penso que o candidato pode assinalar a que ele sabe que é a melhor resposta, e não as que correspondam com a maneira como ele age, principalmente a questão relacionada às qualidades e aos defeitos.
- Validar os dados fornecidos pelo usuário ao responder o questionário, de modo que ele não possa inserir valores errados (fora dos limites representáveis, de tipos diferentes do exigido, etc.);
- Implementar a geração do gráfico do conceito do contratado ideal no *software*.

11 REFERÊNCIAS

ALCHIERI, João Carlos; CRUZ, Roberto Moraes. **Avaliação psicológica: conceito, métodos e instrumentos**. São Paulo: Casa do Psicólogo, (Coleção temas em avaliação psicológica) 2003.

ALUNOS DO CURSO DE COMPUTAÇÃO DA UNIMONTES, **Inteligência Artificial**. Montes Claros, 200?. Disponível em <<http://www.biocomputer.vilabol.uol.com.br/inteligenciartificial.htm>>. Acesso em 20 de setembro de 2005.

BARRETO, Jorge Muniz. **Inteligência Artificial no limiar do século XXI**. 3ª ed. Florianópolis: O Autor, 2001.

BISPO, Patrícia. **Uma seleção baseada em competências**. RH.com.br Conteúdo para Profissionais de Recursos Humanos. Rio de Janeiro, 2003. Disponível em <<http://www.rh.com.br>>. Acesso em 22 de julho 2005.

CARVALHO, Sirlon Diniz de; MARTINS, Weber, **Mapas Auto-organizáveis aplicados a Sistemas Tutores Inteligentes**, UFMG, [entre 1990 e 2005]

CRUZ, Roberto Moraes; PEREIRA, Andréia Chagas; Souza, Juliana de. Competências, perfis profissionais e mercado de trabalho em psicologia. **Psicologia Brasil**, São Paulo, n.8, p.24-27, abril 2004.

DERWING, B. L.; ALMEIDA, Roberto G. de, Métodos Experimentais em Linguística, In: **Processamento da Linguagem**. Porto Alegre: Educart, 2004.

FLEURY, M.T. L.; FLEURY, A. Construindo o conceito de competência. **Revista de Administração Contemporânea**, Edição Especial. apud CRUZ, Roberto Moraes; PEREIRA, Andréia Chagas; Souza, Juliana de. Competências, perfis profissionais e mercado de trabalho em psicologia. **Psicologia Brasil**, São Paulo, n.8, p.25, abril 2004.

FORDHAM UNIVERSITY The Jesuit University of New York **Applied Psychometric Society: What is Psychometrics?** Disponível em: <http://www.fordham.edu/Academics/Colleges__Graduate_S/Graduate__Profession/Arts__Sciences/Academics/Psychology/Psychometrics/Applied_Psychometric/psychometrics_12499.html>. Acesso em 20 de setembro 2005.

FOSTER, S. L.; RITCHEY, W.L. Issues in the Assessment of Social Competence in Children. **Journal of Applied Behavior Analysis**. apud CRUZ, Roberto Moraes; PEREIRA, Andréia Chagas; Souza, Juliana de. Competências, perfis profissionais e mercado de trabalho em psicologia. **Psicologia Brasil**, São Paulo, n.8, p.25, abril 2004.

GRAMIGNA, Maria Rita. Seleção por Competências - Garimpendo talentos e potenciais. **RH.com.br Conteúdo para Profissionais de Recursos Humanos**. Rio de Janeiro, 2003. Disponível em <<http://www.rh.com.br>>. Acesso em 22 de julho 2005.

HAYS, W. L. **Quantificacao em Psicologia**. Sao Paulo: Editora Herder, 1970.

HUBBARD, J. A.; COIE, J.D. Emotional Correlates of Social Competence in Children's Peer relationships. (s.d.) apud CRUZ, Roberto Moraes; PEREIRA, Andréia Chagas; Souza, Juliana de. Competências, perfis profissionais e mercado de trabalho em psicologia. **Psicologia Brasil**, São Paulo, n.8, p.25, abril 2004.

J. M. MOITA NETO, Estatística multivariada - Uma visão didática-metodológica. **Crítica - Revista de filosofia e ensino**. ?, 2004. Disponível em <http://www.criticarede.com/cien_estadistica.html>. Acesso em 05 de novembro de 2005.

LANE, Silvia T.M.; CODO, Wanderley. **O papel do psicólogo na organização industrial**. In: _____. **Psicologia Social: o homem em movimento**. 4. ed. São Paulo: Editora Brasiliense, 1986. p.195-202

LAKATOS, Eva Maria , MARCONI, Marina de Andrade. **Metodologia do trabalho científico**. 3. ed. rev. ampl. São Paulo: Editora Atlas, 1991.

MARRAS, Jean Pierre. **Subsistema de Recrutamento & Seleção**. In: _____. **Administração de Recursos Humanos: do operacional ao estratégico**. 6. ed. São Paulo: Editora Futura, 2002. p. 79-85

PASQUALI, Luiz. **Teoria e Métodos de Medida em Ciências do Comportamento**, Brasília-DF, 1956.

POUNDSTONE , William. **Como mover o Monte Fuji**, São Paulo: Editora Ediouro, 2005.

REIS, Alessandro Vieira dos. **Re: Psicometria** [mensagem pessoal]. Mensagem recebida por

wander@inf.ufsc.br em 20 de setembro de 2005.

RESENDE, Enio. **O livro das competências: desenvolvimento das competências: a melhor auto-ajuda para pessoas, organizações e sociedade.** 2ª ed. Rio de Janeiro: Editora Qualitymark, 2003.

RICH, Elaine. **Inteligência Artificial.** São Paulo: Editora McGraw-Hill, 1988.

REY, B. **As competências transversais em questão.** Porto Alegre: Editora Artmed. apud CRUZ, Roberto Moraes; PEREIRA, Andréia Chagas; Souza, Juliana de. Competências, perfis profissionais e mercado de trabalho em psicologia. **Psicologia Brasil**, São Paulo, n.8, p.25, abril 2004.

ROISENBERG, Mauro. **Redes Neurais Artificiais: Redes Auto-Organizáveis.** Florianópolis: Mauro Roisenberg, [200?]. 20 slides, color.

12 ARTIGO

Um Aplicativo voltado para o Auxílio e Formalização do Processo de Seleção de Pessoas

Shalmanny Lee Barduzzi, Wanderson Rigo

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brasil

{barduzzi,wander}@inf.ufsc.br

***Abstract.** The present job discourse about the development of the one software to help and to formalize the selectives process based in competence, as well as approach the theories wich it is based on. Further to do a comparison between the classification of the candidates realized by this software and the classification realized by a Self-Organizing Neural Network.*

***Resumo.** O presente trabalho discorre sobre o desenvolvimento de um software para auxiliar e formalizar os processos seletivos baseados em competências, bem como aborda as teorias nas quais ele está baseado. Ainda faz uma comparação entre a classificação de candidatos realizada por este software e a classificação realizada por uma Rede Neural Artificial Auto-Organizável.*

1. Introdução

Embora a oferta de mão-de-obra seja crescente, sabe-se que somente a escolha de profissionais capacitados pode oferecer o respaldo imediato que é esperado pelas organizações.

Visando dar subsídios aos profissionais que diariamente convivem com esta demanda, o presente trabalho apresenta o *software Perfil*, o qual foi desenvolvido com a incumbência de ser uma ferramenta para amparar de forma segura e significativa os processos de seleção nas instituições. Ele visa contornar os entraves relativos ao processamento manual dos dados obtidos junto aos candidatos, bem como livra o profissional do trabalho não-criativo, repetitivo e que pode ser automatizado.

O *software*, que é concebido para o ambiente Web, além de prover suporte aos processos seletivos, visa agilizar o processo de elaboração do perfil de características pessoais e profissionais a ser buscado em um candidato, já que ele favorece a aplicação da metodologia de **seleção baseada em competências**.

2. Competências

Fazendo-se uma síntese a respeito do que os diversos autores versam sobre o tema competência, os proponentes deste trabalho proferem que Competência pode ser entendida como “capacidade ou habilidade de atuar eficientemente na realização de alguma atividade, sendo que a motivação pessoal somada a conhecimentos previamente assimilados produz resultados práticos”.

O *software Perfil*, por meio de técnicas **Psicométricas**, ajuda a identificar as características e competências das pessoas que fazem bem o trabalho, sendo que essas características serão usadas como balizadores na seleção: a partir das respostas dadas em um

questionário personalizado elaborando em vista dessas qualidades (focado-se nelas), será possível analisar quais candidatos se aproximam mais desses predicados.

3. Psicometria

O aperfeiçoamento da qualidade dos testes psicológicos é constantemente buscado pela Psicometria, que é uma especialidade da área da Psicologia que concentra as teorias e técnicas de mensuração do comportamento.

A técnica psicométrica aplicada pelo software para mensurar o grau de discernimento dos selecionadores em relação às competências exigidas para um cargo denomina-se **Matriz de Diferencial Semântico**, a qual, por meio de quantificações, gera representações de conceitos em forma de gráficos. Essa técnica é evocada quando da elaboração do perfil de competências exigido para a(s) vaga(s) da seleção em questão. Os usuários irão se deparar com uma matriz de diferencial semântico a ser preenchida/valorada. Suas linhas e colunas são compostas pelos atributos que mais foram citados como essenciais ao perfil de competências que o cargo exige, os quais são extraídos de um espaço amostral definido pelo administrador do sistema.

Bom c/P	2	4	4	2	2	
Iniciativa	3	2	5	3		2
Inteligente	5	3	4		3	2
Líder	4	4		4	5	4
Disciplinado	2		4	3	2	4
Crítico		2	4	5	3	2
	Crítico	Disciplinado	Líder	Inteligente	Iniciativa	Bom c/P
Média	15,3	13,3	19,9	15,2	15,6	14,7
Des. Padrão	4,9	6,7	3,7	5,2	5,0	4,9
Diferença	10,4	6,6	16,2	10	10,6	9,8

Figura 22: Exemplo de Matriz de Diferencial Semântico.

Tal matriz permite que, dois a dois, os atributos definidos como prioritários pelo grupo de usuários sejam confrontados (pela intersecção linha X coluna). Desse processo emergirá um número pertinente ao intervalo de 0 a 5, o qual é uma estimativa numérica que representa o grau de proximidade/similaridade entre os predicados, segundo o julgamento do usuário.

Após todos os usuários terem preenchido a matriz de diferencial semântico é realizado o tratamento estatístico dos dados. Tal processo ocasiona uma ordenação na lista dos predicados, a qual ordenada decrescentemente caracteriza uma pirâmide (Pirâmide de Osgood).

Quanto mais pontos recebe um atributo, mais conexões ele tem com os outros, portanto mais central ele é para a compreensão do todo. Por exemplo, para o conceito “chá”, supõe-se que o elemento “erva” seja o mais votado. Isso significa que o respondente acha que o fundamental em termos de chá é a erva, e que tudo mais (água quente, xícara, açúcar etc.) não é tão essencial quando alguém se refere ao conceito “chá” [REIS, 2005].

Só assim, após definida e compreendida a Pirâmide, pode-se dizer que o **perfil de competências** delineado pelos usuários está formado. Assim sendo, o profissional que opera o software Perfil já têm bases sólidas para elaborar o **questionário** voltado para averiguar nos candidatos essas competências elencadas como principais.

4. Fazendo-se uso do Software Perfil

A partir da concepção do protótipo do software, este foi disponibilizado na WEB para a execução de testes de usabilidade e aplicabilidade via a elaboração de uma seleção hipotética para o Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de Santa Catarina (PET-CCO).

Os testes foram realizados junto ao grupo de bolsistas desse programa e ainda contaram com a participação de outros voluntários, os quais foram convidados via e-mail. O objetivo principal dos testes foi a coleta de um conjunto de dados, os quais correspondem às respostas que os usuários do software Perfil deram às questões elaboradas para uma seleção hipotética do PET-CCO.

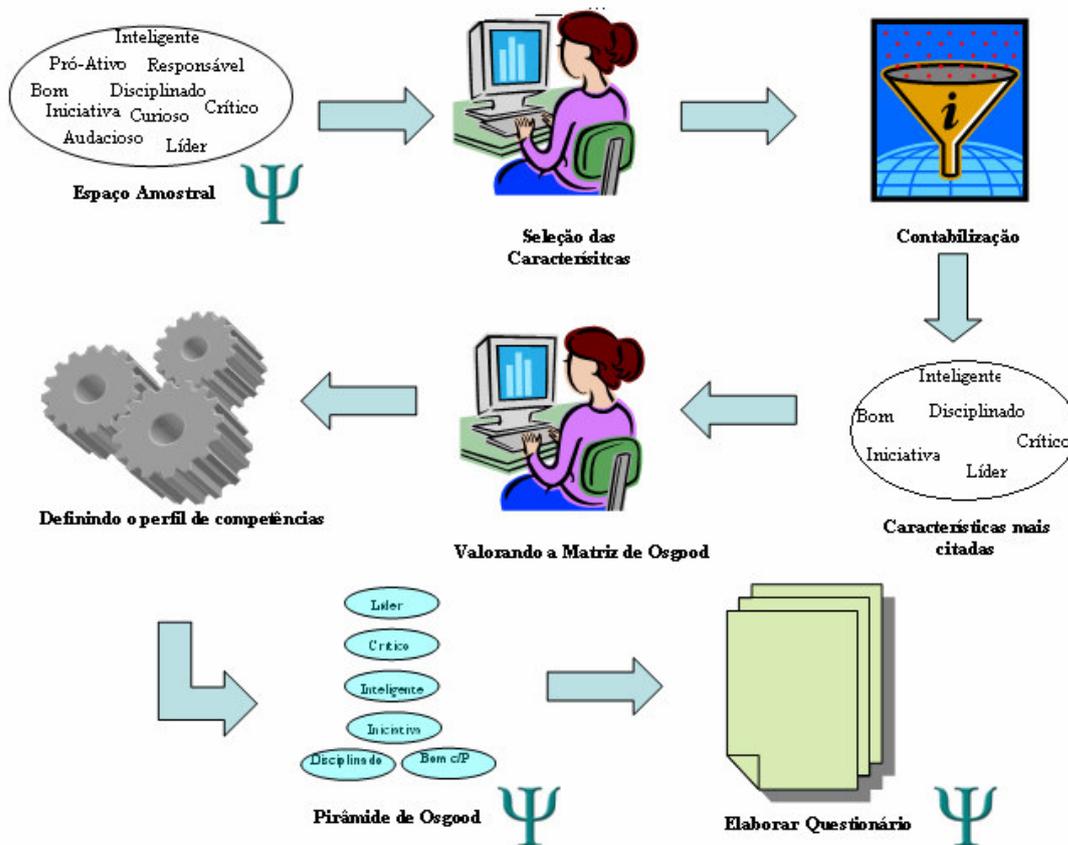


Figura 23: Fluxograma de parte do funcionamento do software Perfil

A partir desses dados, o programa classificou decrescentemente os usuários mediante a soma da pontuação atribuída a cada alternativa das respostas dadas por eles no questionário. Esses dados também foram fornecidos a uma Rede Neural Artificial Auto-Organizável, a qual tentou agrupar os candidatos em nichos.

5. Redes Neurais Auto-Organizáveis

Atualmente existem dezenas de modelos de redes neurais estruturados para as mais diversas aplicações. No escopo deste trabalho, julgou-se mais adequado ao problema o modelo de Rede

Neural denominado Mapas Auto-Organizáveis ou *Self Organizing Maps* (SOM). A escolha deve-se a capacidade da SOM se auto-organizar (caracterizando um aprendizado do tipo não supervisionado) e manter uma ordenação topológica dos padrões treinados [CARVALHO; MARTINS, ?].

Essas características permitem a rede detectar similaridades entre vários sinais, agrupando-os em classes de similaridades, sendo, desta forma, um passo para a **determinação de padrões** de uma coleção de sinais [BARRETO, 2001]. No contexto deste trabalho espera-se que, a partir das respostas dadas pelos candidatos ao questionário elaborado, a rede viesse a classificar os respondentes baseado-se nas características das informações coletadas.

6. Metodologia usada para o uso da Rede Neural Auto-Organizável

Para a Rede Neural, o que importa do questionário são os valores associados a cada resposta e não a semântica envolvida nas questões. Logo, mostra-se abaixo a forma que as respostas de um candidato fictício “ideal” tomaram após a formatação dos dados, bem como o escore alcançado por ele segundo os critérios fornecidos ao software:

Ideal = [10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 0 0 10 0 0 0 10 10 10 10 10 0 10 0 10 10 0 0 10]

$\Sigma = 250$ pontos.

Depois de configurada e treinada a rede, esta foi testada via simulação. Esse processo compreende a apresentação de cada vetor de entrada à rede, e esta responde indicando qual o neurônio que apresenta os pesos das conexões mais próximos daquele dos valores do vetor apresentado (diz-se o “neurônio vencedor”), ou seja, o neurônio que melhor responde àquelas entradas.

Então, tendo-se todos os dados de saída da rede (todos os neurônios vencedores), pôde-se determinar e analisar a performance da mesma.

7. Confrontando as classificações

Como a classificação feita pelo software Perfil é um somatório de todas as alternativas do questionário assinaladas pelo candidato, sejam elas positivas ou negativas, é fácil deduzir que o candidato melhor classificado é aquele que fez mais pontos em relação aos demais.

Já para a Rede Neural, esperava-se que, de acordo com as características das entradas (as respostas do questionário aplicado pelo software) que recebeu, ela agrupasse os candidatos em nichos, sendo que estes representariam grupos de candidatos com características semelhantes.

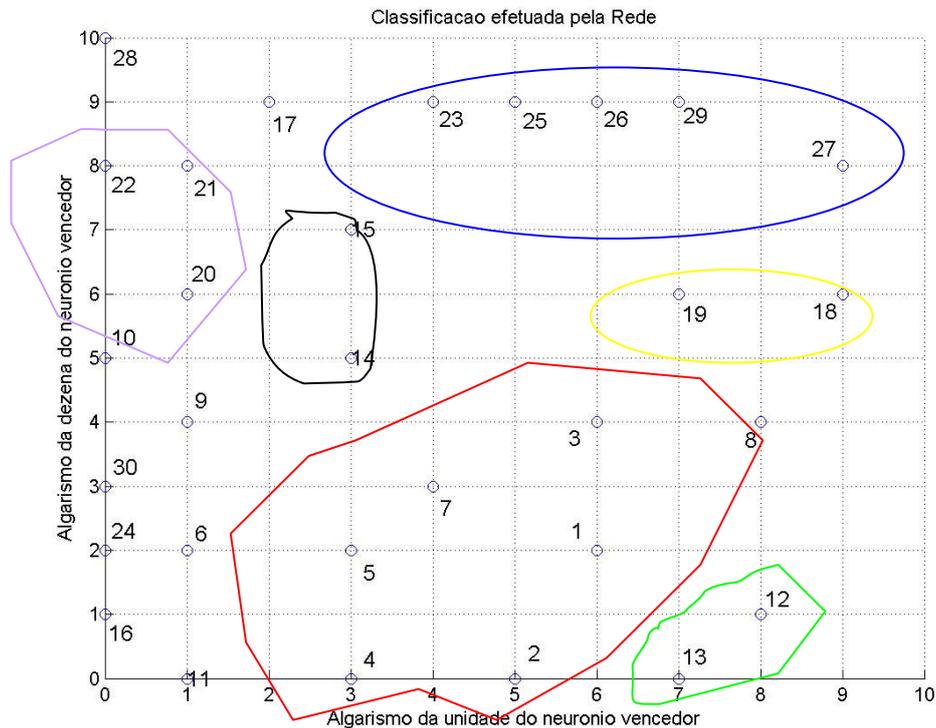


Figura 24: Saída da Rede Neural Auto-Organizável (5000 épocas).

Porém, como pode ser visto na Figura 24, isso não ocorreu plenamente, já que os números, os quais indicam a posição em que o candidato ficou na classificação efetuada pelo *software* Perfil, não estão adequadamente agrupados em mesmos nichos de acordo com a ordem de grandeza.

Levanta-se agora alguns fatores que colaboraram para aumentar a discrepância entre as duas classificações. Eles mais se devem à semântica e às características dos dados do que à configuração da rede neural artificial.

7.1. Semântica dos Dados

Para o software, a pontuação e conseqüentemente a classificação dos candidatos é dada pelo somatório de todas as alternativas do questionário assinaladas. Assim sendo, se um candidato José obtivesse o seguinte vetor de respostas:

Candidato José = [10 0 10 0 10 0 10 0 10 0]

$\Sigma = 50$ pontos.

E um candidato Pedro:

Candidato Pedro = [0 10 0 10 0 10 0 10 0 10]

$\Sigma = 50$ pontos.

Ambos apresentam a mesma pontuação, logo apareceriam juntos na lista da classificação por pontos. Já para a rede, eles são casos completamente distintos, pois ela utiliza simultaneamente todas as variáveis do vetor para a interpretação do conjunto de dados. Além disso, a rede analisa individualmente os dados do vetor, já que cada um deles representa uma

entrada para um neurônio. Diante do exposto, pode-se concluir que ela não faz uma classificação quantitativa e ordenada, mas sim uma classificação **qualitativa** dos dados de entrada de acordo com suas características.

7.2. Métricas usadas pela Rede

Aos olhos dos proponentes deste trabalho, o método homogêneo e generalista usado para calcular a distância entre dos dados de entrada da rede e os pesos dos neurônios não é um bom fator de medida para se tentar classificar candidatos de acordo com as características de suas respostas. Por exemplo, se a resposta de uma questão possuísse, de acordo com a Psicometria, uma capacidade muito grande de diferenciar entre um tipo de perfil de pessoa e outro, esse grau de ênfase deveria ser transposto para a rede. Isso não ocorre no momento, já que todos os neurônios são tratados da mesma forma.

8. Conclusão

Sabe-se que é preciso experiência e técnica para se efetuar uma boa seleção de pessoal, bem como para se tratar problemas complexos ligados a I.A. Assim sendo, durante esse breve tempo de contato que se teve com as teorias das áreas de I.A. e de Psicologia, os proponentes dessa monografia se julgam aptos a apontar para algumas direções a serem seguidas.

A fim de se melhorar a classificação dos candidatos, o que se sugere é um trabalho mais detalhado junto a psicólogos especialistas em Psicometria. Esses esforços poderiam levar a construção de questionários mais robustos, os quais serviriam para se extrair e medir mais precisamente as características dos candidatos.

Então, tendo-se um questionário devidamente “calibrado” e preciso, os pormenores deste poderiam ser mapeados para a rede. Nesse ponto junta-se uma nova ferramenta de trabalho, a qual foi visualizada somente no final da elaboração dessa monografia. Trata-se da Análise de Componentes Principais, técnica do campo da Estatística que, com os devidos estudos e adequações, poderia ajudar na transposição da ênfase oriunda das perguntas do questionário para as entradas da rede.

Referências

- Barreto, Jorge Muniz. (2001) “Inteligência Artificial no limiar do século XXI”. 3ª ed. Florianópolis: O Autor.
- Carvalho, Sirlon Diniz de; Martins, Weber. (entre 1990 e 2005) “Mapas Auto-organizáveis aplicados a Sistemas Tutores Inteligentes”. UFMG.
- Reis, Alessandro Vieira dos. Re: Psicometria [mensagem pessoal]. Mensagem recebida por wander@inf.ufsc.br em 20 de setembro de 2005.
- Rigo, Wanderson; Barduzzi, Shalmanny Lee. (2005) “Um Aplicativo voltado para o Auxílio e Formalização do Processo de Seleção de Pessoas”. Monografia (Graduação em Ciências da Computação) – Universidade Federal de Santa Catarina, Florianópolis-SC