

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Fábio Davanzo Franco de Oliveira

Renato Oliveira e Castro

DESENVOLVIMENTO DE UM PROGRAMA QUE AVALIA A
CAPACIDADE FÍSICA E FUNCIONAL DO IDOSO

Florianópolis, março de 2005

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Fábio Davanzo Franco de Oliveira

Renato Oliveira e Castro

DESENVOLVIMENTO DE UM PROGRAMA QUE AVALIA A
CAPACIDADE FÍSICA E FUNCIONAL DO IDOSO

Relatório final da disciplina de Projetos II do curso
de Bacharelado em Ciências da Computação da
Universidade Federal de Santa Catarina
Orientadora: Lúcia Helena Martins Pacheco

Florianópolis, março de 2005

Resumo

Esse trabalho contribuiu para um programa de capacitação física do idoso. Para isto foi desenvolvido um *software* que armazena dados colhidos em campo, e possibilita o posterior tratamento destas informações. O *software* facilita o estudo desses dados, gerando um relatório simples para o usuário de como ele está com relação a sua saúde e seu desempenho físico. Outros dados relevantes também podem ser armazenados, para que se possa fazer um levantamento de como o usuário é nos dias atuais, no passado e uma possível previsão para o futuro. O programa desenvolvido foi implementado em Java (*Net Beans 4.0*) que contém um banco de dados (*MySQL 4.1, Dbdesigner 4.0*), onde todos os dados do usuário são armazenados e manuseados de forma a garantir uma busca sobre os principais aspectos de estudo, como: idade, renda, sexo, doenças, dentre outros.

Palavras-chave: idoso, física, base de dados, Java

Abstract

This work shows software that was developed to help a program of qualification of elder people for physical fitness. Such software allows storing of field-collected data, to arrange them in a database in order to information management. Also, it makes possible to generate a report about any subject that is being evaluated and keeps individual records updated. So, a personal profile of physical fitness is recorded during some time what allows several academic researches about this matter and also prognostics. Collected data includes, amongst others, age, income, sex and diseases. But other data can be easily added with few changes in software. The implementation used Java language (Net Beans 4,0) and the data-bank MySQL (Mysql 4,1, to Dbdesigner 4,0) because they are freeware and provide all functionalities that this kind of software demands.

Keywords: elder people, physical fitness, database, Java, MYSQL

Lista de Figuras

FIGURA 1: Ilustração gráfica do teste de flexibilidade (adaptada de OSNESS et al., 1990). Pág: 26.

FIGURA 2: Dinâmico e agilidade AAHPERD. Pág: 29.

FIGURA 3: Ilustração gráfica do teste de coordenação (adaptada de OSNESS et al., 1990). Pág: 31.

FIGURA 4: Cadastro de Turmas. Pág: 49.

FIGURA 5: Cadastro de Pessoa. Pág: 50.

FIGURA 6: Testes. Pág: 51.

FIGURA 7: Relatório. Pág: 52.

FIGURA 8: Buscas. Pág: 53.

FIGURA 9: Tabelas. Pág: 55.

Lista de Siglas e Abreviaturas

AAHPERD (American Alliance for Health, Physical Education, Recreation)

MySQL (My Structured Query Language)

SQL (Structured Query Language)

ACSM (American College of Sports Medicine)

XP (Extreme Programming)

DBA (Administradores de Banco de Dados)

BD (Banco de Dados)

RAG (Resistência aeróbica geral e habilidade de andar)

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 Justificativa.....	12
1.2 Apresentação do Trabalho.....	15
2 OBJETIVOS.....	17
2.1 Objetivo Geral.....	17
2.2 Objetivo Específico.....	17
3 MEDIAÇÃO DA APTIDÃO FÍSICA.....	18
3.1 Importância da Atividade Física na Terceira Idade.....	18
3.2 Métodos e Testes.....	22
3.2.1 Bateria de Testes AAHPERD.....	23
3.3 Necessidade de Aquisição e Mediação de Dados.....	31
3.4 Criação de Base de Dados.....	31
4 METODOLOGIAS PARA MODELAGEM COMPUTACIONAL.....	33
4.1 Programação Orientada a Objeto.....	33
4.2 Linguagem Java.....	37
4.3 SQL, MySQL e Banco de Dados.....	39
4.4 XP (<i>Extreme Programming</i>).....	45

5	MODELAGEM PROPOSTA / IMPLEMENTAÇÃO.....	49
5.1	Use Cases.....	49
5.2	Modelagem do Banco.....	53
6	CONCLUSÃO E SUGESTÕES PARA FUTUROS TRABALHOS.....	56
7	REFERÊNCIAS.....	57
8	APÊNDICE.....	59
8.1	Código Fonte.....	59
8.2	Artigo.....	151

1 INTRODUÇÃO

Em nosso País, no início da década de 80, ainda não era comum encontrarmos estudos normativos preocupados em relacionar a aptidão física à saúde. Temos nesse contexto que a literatura é rica em avaliações voltadas à performance motora, não incluindo, na maioria das vezes, todos os componentes da aptidão física relacionada à saúde, proposta pela *AAHPERD (American Alliance for Health, Physical Education, Recreation)* 1984.

O emprego de normas de aptidão física relacionada à saúde começou então, timidamente, a ser utilizado por alguns profissionais em suas instituições de ensino. Alguns estudos, nesse campo, foram realizados por Barbanti (1982, *apud* Revista Eletrônica da Terceira Idade), Nahas (1985, *ibidem*) e Madureira (1987, *ibidem*). O primeiro e o terceiro abrangiam população de escolares, enquanto o segundo, universitários. Após isto, outros estudos foram realizados contribuindo para a ampliação da bibliografia na área. ¹

Há uma diferenciação entre a aptidão física relacionada à saúde e a aptidão motora. A diferença na bateria de testes de aptidão física relacionada à saúde e aptidão motora está no fato de que a primeira solicita apenas os componentes estritamente relacionados à saúde do indivíduo, ou seja, a função

¹ Idade Ativa: revista eletrônica da Terceira Idade. Disponível em:

<http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>

cardiorrespiratória e função músculo-esquelética enquanto que a segunda bateria, além desses componentes, pode envolver testes de agilidade, força, equilíbrio e velocidade (AAHPERD, 1984).

Para Pate (1983 *apud* Mazo et al., 2003), os componentes de aptidão motora e aptidão física relacionada à saúde nem sempre têm sido claramente distinguidos uns dos outros. A aptidão motora está relacionada diretamente à performance, pois seus testes exigem, em grande parte, agilidade, velocidade, força, potência, resistência muscular, equilíbrio etc. (*apud* Mazo et al., 2003).

Conforme explica Pate (1983, *ibidem*), comparando aptidão motora com aptidão física relacionada à saúde, o segundo apresenta um conceito restrito, que geralmente inclui somente os componentes associados à prevenção e/ou promoção da saúde.

O estabelecimento do conceito de aptidão física relacionada à saúde derivou de estudos clínicos que evidenciaram a incidência maior de problemas de saúde entre adultos e indivíduos mais idosos de vida sedentária. Sua definição diz que:

“é a habilidade de executar atividades físicas com energia e vigor sem excesso de fadiga, e a demonstração de qualidades e capacidades físicas que conduzam ao menor risco de desenvolvimento de doenças hipocinéticas” (Pate, 1983, p. 19 *apud* Mazo et al., 2003).

É necessário, neste momento, frisar que não se é contra as baterias de testes de aptidão motora, mas que a preocupação reside, sobretudo, na dificuldade de aplicação dos testes, pelo tempo que necessita ser dispensado e pelo próprio acompanhamento de cada usuário.

Hoje em dia, pela dificuldade em se adotar uma bateria de testes mais simples que exija menos tempo na coleta dos dados, grande parte dos profissionais acabam por deixar de realizar qualquer tipo de teste, conseqüentemente não ficam sabendo quase nada sobre o desenvolvimento do seu aluno.

O objetivo de trabalhos assim é apresentar mais um recurso na avaliação do usuário, procurando levar em consideração a realidade sócioeconômica, as dificuldades existentes, o tempo dispendido na coleta dos dados, a validade, a objetividade e a fidedignidade dos testes.

Admite-se que a bateria de testes da AAHPERD (1984) apresenta o mínimo necessário para que se possa prevenir, manter e/ou desenvolver um trabalho básico voltado à saúde do usuário e não só a avaliações motoras do mesmo.

Cabe perguntar: como viabilizar este trabalho? Primeiramente, necessita-se definir quais as medidas e os testes a serem aplicados. Alguns deles como Força de membros inferiores – pernas, Força de membros superiores – braço direito, Flexibilidade (FLEX), Flexibilidade de ombros: Alcançar atrás costas D – direito e E – esquerdo, Dinâmico e agilidade, Coordenação (COO), entre outros. Após coletados os dados, passa-se à elaboração dos *percentis/decis* e, assim, ter-se-á

uma representação numa escala (norma) que é capaz de informar como determinada pessoa está em relação ao grupo ao qual pertence. A estatística auxiliará na elaboração das tabelas para cada variável, as quais fornecerão as normas antropométricas e de aptidão física relacionada à saúde.

Com base nisso este trabalho propõe um software para gerenciar e administrar todos esses dados gerando um banco de dados que armazenará todos os testes, dados pessoais, resultados, enfim o que for relevante para se fazer um bom levantamento de como a pessoa está com relação aos demais. Assim, pode-se mostrar a evolução do usuário no decorrer do ano e imprimir um relatório qualitativo e de fácil entendimento. Isto auxiliaria os avaliadores; que não mais ficaram horas na frente do computador analisando planilhas de dados. Também o usuário pode ter um relatório simplificado, com seus dados, seu desempenho, e tudo isso de forma bastante rápida.

O projeto/software foi desenvolvido em Java 1.5, sendo utilizado o *Net Beans 4.0* para a implementação do programa, e usaremos um banco de dados, em *Mysql 4.1* e *dbdesigner 4.0*, para armazenar as informações. A interface do mesmo, o tratamento dos dados recebidos, a geração do relatório, e algumas consultas ao banco de dados foram projetados com relação ao AAHPERD.

1.1 Justificativa

“Na sociedade atual o sedentarismo é um fato predominante que atinge todas as camadas sociais e faixas etárias. A falta de movimento deve-se, muitas

vezes, aos recursos tecnológicos (informática, televisão, vídeo game, dentre outros). As atividades da vida diária na sociedade tecnicamente desenvolvida não fornece trabalho físico suficiente para estimular o sistema cardiorrespiratório e muscular, havendo a necessidade de uma prática regular de atividade física para manter a capacidade funcional” (Mazo, et al., 2003, p. 109)

Existem alguns meios de se julgar se a pessoa é ou não sedentária, normalmente isso é calculado pelo gasto de calorias. O sedentarismo acontece em todas as idades, mas no idoso pode acarretar danos maiores.

O ser humano precisa de movimento, um mínimo esforço físico pode dar resultados grandiosos, manter uma certa atividade física pode significar mais saúde e maior independência.

Buchner e Wagner (1992, *apud* Mazo, et al., 2003), Elward e Larson (1992, *ibidem*), ACSM (2000, *ibidem*) apontam que com a atividade física, especialmente para idosos, alguns benefícios são atingidos com bastante sucesso: maior longevidade, redução das taxas de morbidade e mortalidade, redução do número de medicamentos prescritos, melhoria da capacidade fisiológica em portadores de doenças crônicas, prevenção do declínio cognitivo, manutenção de status funcional elevado, redução da frequência de quedas e fraturas, manutenção da independência e autonomia.

Outro aspecto importante são os benefícios psicológicos. Com as atividades físicas a pessoa fica mais disposta, com isso sua auto-imagem estará mais saudável, com os músculos mais tenses e com uma aparência estética bem

melhor. Melhorando a auto-imagem tem-se uma melhora na auto-estima a pessoa se sentirá bem, mais bonita, mais sociável o que gerará uma melhor aproximação para com outras pessoas, um contato social mais fácil, e de certa forma a pessoa tenderá a ter mais prazer pela vida. Analisando de um modo geral, notamos que a preocupação com as atividades físicas na velhice acarreta benefícios em três pontos fundamentais: nos aspectos biológicos, nos aspectos psicológicos e nos aspectos sociais.

Exemplificando: Os idosos tendem a enfrentar alguns problemas, como a solidão, ausência de objetivo de vida, de atividades ocupacionais, sociais, de lazer, artístico-culturais e físicas. Com a atividade física aumenta a possibilidade de uma reintegração na sociedade minimizando esses problemas.

Devido a importância da atividade física, programas destinados a idosos começaram a se espalhar pelo mundo. Sendo assim, a necessidade de testes para idosos começava a ficar imprescindível, e com isso vários dos mesmos foram criados:

Essas avaliações para idosos têm alguns objetivos bastante interessantes:

- diagnosticar e verificar a eficiência dos programas quanto à aptidão física, capacidade funcional, aspectos psicossociais, ingestão alimentar e nível de atividade física.
- Verificar a evolução da aptidão física e da capacidade funcional, em relação ao processo de envelhecimento.

- Verificar as associações existentes nos aspectos antropométricos, neuromotores e metabólicos da aptidão física, capacidade funcional, aspectos psicossociais e nutricionais dos idosos.
- Determinar as variáveis que devem ser priorizadas na elaboração dos programas de intervenção.

O projeto das professoras Giovana Z. Mazo, Marize A. Lopes e Tânia B. Benedetti elabora testes para idosos e, para maior eficiência, precisa de um programa computacional capaz de gerenciar esses testes, armazenar os dados, tratar os dados e gerar um resultado. Como a questão da velhice, provavelmente afetará a todos, uma ajuda nessa área seria de grande interesse, pelos benefícios que poderia trazer para melhoria da qualidade de vida na terceira idade.

1.2 Apresentação do Trabalho

Será mostrado no capítulo próximo os objetivos do trabalho: gerais e específicos.

No terceiro capítulo será mostrada as mediações da aptidão física, primeiramente mostrando a importância da atividade física na vida das pessoas, depois alguns métodos e testes para a avaliação da aptidão física, e nesse contexto a importância das mediações e aquisições de dados para a base de dados que será implementada.

No Capítulo 4 será mostrada as metodologias para modelagem computacional com as relações entre linguagem orientada a objeto, a linguagem Java e a programação XP relacionados ao projeto e a base de dados do mesmo.

Enfim será apresentada a conclusão, algumas sugestões para projetos futuros, as referencias bibliográficas e em apêndice o código fonte do sistema.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um sistema computacional capaz de gerenciar os aspectos os testes físicos realizados para idosos. Num primeiro momento o trabalho se focará nos testes AAHPERD. Entretanto, pretende-se futuramente continuar o trabalho para que se possa dispor de outros tipos de testes similares. Além disto, buscas mais aprimoradas=relacionando dados a especificações mais detalhadas ou mais abrangentes poderão ser providas pelo sistema.

2.2 Objetivos Específicos

- Implementar um programa capaz de gerenciar toda uma bateria de testes específicos do AAHPERD;
- Fazer para esse programa uma interface amigável para facilitar o trabalho de pessoas leigas em computação;
- Criar um banco de dados suficientemente simples e de fácil acesso, para que se possa fazer boas buscas nele de acordo com a necessidade do usuário;
- Tratar de forma adequada os dados coletados para geração de um relatório bem estruturado e de fácil entendimento;

3 MEDIAÇÃO DA APTIDÃO FÍSICA

3.1 Importância da Atividade Física na Terceira Idade

O contato com atividades físicas na terceira idade começa geralmente por indicação médica. O fato é que essas pessoas acabam encontrando nos exercícios muito mais que alívio para suas “dores”, fazem novas amizades e têm momentos de descontração.

Além dos aspectos ligados à saúde, as atividades físicas trazem inúmeros benefícios psicológicos, de auto-estima e de melhoria do relacionamento social, aspectos muito importantes para pessoas da terceira idade, devido às inúmeras mudanças advindas desta fase da vida. O horário de exercício físicos é um momento em que a pessoa reserva o tempo para si própria, para cuidar de sua beleza, de sua saúde e mesmo para refletir sobre a própria vida. “Infelizmente, ainda há muita resistência das pessoas em dedicar um espaço para a realização dessas atividades” analisa Mara Patrícia Traina Chacon-Mikahil, professora e

pesquisadora do Departamento de Ciências do Esporte da Faculdade de Educação Física da Unicamp.²

A maioria das pessoas busca praticar atividades físicas por problemas de saúde – mais acentuados na terceira idade – e não como lazer. Ainda é pouco desenvolvido no Brasil o hábito de realizar atividade física na juventude. O que freqüentemente ocorre é que as pessoas chegam à terceira idade com diversas degenerescências e patologias resultantes do sedentarismo, o que as leva a procurar exercícios físicos³

“Durante o processo do envelhecimento temos alterações fisiológicas e em nossas capacidades físicas, tais como (Mazo, et al., 2003):

- Redução da força.
- Redução do volume muscular.
- Aumento do tecido não contrátil (gordura e tecido conectivo) no músculo.

² Idade Ativa: revista eletrônica da Terceira Idade. Disponível em: <http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>

³ Idade Ativa: revista eletrônica da Terceira Idade. Disponível em: <http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>

- Redução na área de secção transversa do músculo esquelético **que** se torna mais pronunciada a partir da 5 década de vida.
- As fibras tipo II, com o envelhecimento, reduzem em tamanho enquanto que as fibras do tipo I permanecem praticamente inalteradas.
- A redução da área de secção transversa do músculo também se dá às custas da redução do número de fibras ao longo do processo de envelhecimento.
- O envelhecimento parece provocar redução no número tanto de fibras do tipo I como do tipo II.
- Com o envelhecimento ocorre também uma redução no número de unidades motoras. Esse fenômeno parece ser resultante da perda de neurônios motores alfa da medula espinhal. **Em contrapartida** as unidades motoras remanescentes aumentam de tamanho.
- Capacidade reduzida no idoso em gerar força em alta velocidade (potência).
- Vários estudos têm relacionado a redução da força muscular a uma maior suscetibilidade a quedas, fraturas e dependência do idoso.
- Parte da redução da capacidade aeróbia (50%) no idoso tem sido atribuída a sua perda de massa muscular.”

Os efeitos da diminuição natural do desempenho físico podem ser atenuados se forem desenvolvidos com os idosos, programas de atividades físicas que visem a melhoria das capacidades motoras que apoiam a realização de sua vida cotidiana, dando ênfase na manutenção das aptidões físicas de principal importância no seu bem estar como: a força muscular, a flexibilidade, a mobilidade articular e a resistência.

A força dos membros inferiores é muito importante para a prevenção de quedas. A força dos músculos abdominais e lombares para a manutenção da estática corporal e no andar, evitando dores crônicas e muita incapacidade.

A mobilidade articular, está diretamente relacionada com a amplitude do movimento, a flexibilidade com sua qualidade. A resistência está ligada a frequência com que o idoso se dedica a ocupações ativas. A redução desses fatores prejudica a performance de muitas atividades do dia a dia.

A promoção de saúde e a qualidade de vida são os objetivos mais importantes numa atividade física com idosos. É fundamental que o idoso aprenda a lidar com as transformações de seu corpo e tire proveito de sua condição, prevenindo e mantendo em bom nível sua plena autonomia. Para isso é necessário que se procure estilos de vida ativos, integrando atividades físicas a sua vida cotidiana.

Fazer atividades moderadas de forma sistemática, dão melhor resultado do que atividades intensas durante um curto espaço de tempo. O aumento da carga de exercícios deve ocorrer de forma gradual, evitando o cansaço intenso e a dor,

objetivando desenvolver a resistência e manter níveis aceitáveis de capacitação física.

Um controle médico regular é fundamental para se saber as atividades mais aconselháveis e estabelecer eventuais restrições sobre o que se faz.

Os exercícios devem ser: atraentes, diversificados, com intensidade moderada, de baixo impacto - nas estruturas musculares, esqueléticas e articulares - realizados de forma gradual, e promover a aproximação social, sendo desenvolvidos de preferência coletivamente, respeitando as individualidades de cada um, sem estimular atividades competitivas, pois tanto a ansiedade como o esforço aumentam os fatores de risco.

Com isso é possível se alcançar níveis bastante satisfatórios de desempenho físico, gerando autoconfiança, satisfação, bem-estar psicológico e interação social. Deve-se levar em conta que o equilíbrio entre as limitações e as potencialidades da pessoa idosa ajudam a lidar com as inevitáveis perdas decorrentes do envelhecimento.

3.2 Métodos e Testes

Existem vários métodos para avaliar as aptidões físicas das pessoas, normalmente encontramos dois seguimentos de avaliações e testes: um para aptidões físicas relacionadas a saúde, e outro para aptidões físicas relacionadas a aptidão motora.

“A diferença na bateria de testes de aptidão física relacionada à saúde e aptidão motora está no fato de que a primeira solicita apenas os componentes estritamente relacionados à saúde do indivíduo, ou seja, a função cardiorrespiratória e função músculo-esquelética enquanto que a segunda bateria, além desses componentes, pode envolver testes de agilidade, força, equilíbrio e velocidade” (AAHPERD, 1984 *apud* Mazo, et al., 2003).

Dentre elas foi escolhido a bateria de testes AAHPERD. Essa bateria de testes agrupa os dois tipos de avaliação e é bastante completo:

3.2.1 Bateria de Testes AAHPERD

A bateria AAHPERD (Mazo, et al., 2003) mostra alguns componente de aptidão física relacionados à saúde, se preocupando, além do desempenho físico, com o bem estar e a saúde do indivíduo.

Alguns dos testes dessa bateria são:

Força de membros inferiores – pernas AAHPERD.

Instrumentos: cadeira sem braços, cronômetro.

Posição do avaliado: sentado na cadeira e com os pés apoiados ao solo.

Posição do avaliador: ao lado com cronômetro nas mãos para ser acionado.

Procedimento: ao sinal o indivíduo irá levantar-se totalmente da cadeira e sentar-se, durante 30 segundos. O avaliador controlará o número de repetições, que será executado pelo avaliado.

Observação: ter o cuidado de que o idoso sente totalmente na cadeira e contar o número de repetições cada vez que sentar. Pois esse número será armazenado

Força de membros superiores – braço direito AAHPERD.

Instrumentos: halteres, cronômetro, cadeira sem braços. Mulheres (halteres de 1,84 Kg) e homens (halteres com 3,63 Kg).

O teste será realizado com os pesos acima e com o peso proposto abaixo: mulheres 2 Kg e homens 3 Kg (Matsudo, 2001 *apud* Mazo, et al., 2003).

Posição do avaliado: sentado encostado na cadeira sem braços, tronco ereto, pés apoiados ao solo. O braço dominante deve permanecer relaxado e estendido ao longo do corpo enquanto a mão não dominante deverá estar sobre a coxa. O peso estará na mão dominante estendida com a mão voltada para trás. O peso deverá estar posicionado paralelamente ao solo na mão dominante, com a palma da mão voltada para frente.

Posição do avaliador: ajoelhado atrás do avaliado tocando no músculo tríceps para que durante o teste este não se mova para trás.

Procedimento: ao sinal “vai”, o avaliado deverá virar a palma da mão para frente e flexionar o antebraço sobre o braço “contração do bíceps”, e estender novamente. Quando esta prática de tentativa for realizada, o halter será colocado no chão e o avaliado terá 1 minuto de descanso. Após este tempo, o teste é iniciado, repetindo-se o mesmo procedimento, mas desta vez o avaliado realizou o maior número de repetições no tempo de 30 segundos, que será anotado como resultado final do teste.

Observação: O avaliador deverá cuidar para que o braço não seja hiperestendido. O número a ser armazenada é o número de repetições no período de 30 segundos

Flexibilidade (FLEX) - AAHPERD

O objetivo é verificar a flexibilidade coxo-femural

Instrumentos: 50,8 cm de fita adesiva, fita métrica de metal (63,5 cm).

Organização dos instrumentos: fita adesiva de 50,8 cm afixada no solo e uma fita métrica de metal também afixada no solo perpendicularmente, com a marca de 63,5 cm diretamente colocada sobre a fita adesiva. São realizadas duas marcas equidistantes 15,2 cm do centro da fita métrica (Figura 4).

Posição do avaliado: sentado no solo, pés descalços e afastados 30,4 cm entre si, os artelhos apontando para cima e os calcanhares centrados nas marcas feitas na fita adesiva. O zero da fita métrica aponta para o participante.

Posição do avaliador: segura os joelhos do participante para não permitir que o mesmo flexione.

Procedimento: com as mãos, uma sobre a outra, o participante vagarosamente desliza as mãos sobre a fita métrica tão distante quanto pode, permanecendo na posição final no mínimo por 2 segundos. São realizadas duas tentativas de prática, seguidas de duas tentativas de teste. O resultado final será dado pela melhor das duas tentativas anotadas.

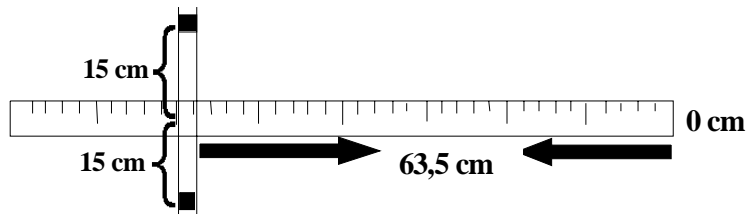


FIGURA 1: Ilustração gráfica do teste de flexibilidade (adaptada de OSNESS et al., 1990 apud Mazo, et al., 2003).

Será armazenado o valor que for obtido com a melhor medição.

Flexibilidade de ombros: Alcançar atrás costas D – direito e E - esquerdo
AAHPERD.

Instrumento: fita métrica milimétrica.

Posição do avaliado: em pé, elevar o braço direito (mensurar o lado direito), flexionar o cotovelo. Ao mesmo tempo estender o braço esquerdo para baixo, dobrar o cotovelo.

Procedimento: o cotovelo que está flexionado deverá tocar a posição mais baixa possível e ao mesmo tempo o braço esquerdo deverá subir o máximo possível tentando sobrepor os dedos da mão direita. Para o braço esquerdo repete-se o procedimento com o braço esquerdo em cima e o direito embaixo.

Observação: Para mesurar deve ser assim considerado:

Se os dedos das mãos apenas tocarem as pontas = zero

Se os dedos das mãos não se tocarem deve ser medida a distância que falta com o sinal negativo (exemplo: - 4).

Se os dedos das mãos se sobrepuserem , deverá ser mensurado o quanto uma mão sobrepôs a outra, considerando o dedo médio com sinal positivo (exemplo: +5).

Será armazenado o valor da medição em considerando o ponto indicado como 0 e com valores positivos e negativos.

Dinâmico e agilidade AAHPERD.

Instrumentos: cadeira com braços (altura 43 cm), 2 cones, cronômetro e fita métrica.

Organização dos instrumentos: medido os pontos no chão onde os pés serão colocados (X), os cones são colocados com a extremidade mais longe a 1,80 para o lado e 1,50 para atrás do ponto inicial (X). Permanecendo estes na diagonal posterior direita e esquerda da cadeira. A área deve ser bem iluminada, o chão plano e não escorregadio.

Posição do avaliado: os sujeitos iniciam sentados, com os pés no chão.

Posição do avaliador: perto da área, mas que o avaliado tenha boa visibilidade, com cronômetro para ser acionado.

Procedimento: ao sinal, levanta-se da cadeira, segue para a direita, passando pela parte interna do cone, rodeando o mesmo e retornando à cadeira, sentando-se e levantando os pés do chão. Sem hesitar, o sujeito levanta-se novamente e realiza o mesmo percurso para a esquerda, completando o primeiro

circuito. O teste consiste em dois circuitos completos. Deve-se realizar duas séries com intervalo de descanso de 30s. Anotar o tempo de cada série, sendo considerada a melhor série.

Observação: quando o indivíduo sentar deve fazê-lo de maneira completa e total, podendo utilizar as mãos para auxiliar no sentar e levantar da cadeira

Os resultados variam de 15 a 35 segundos.

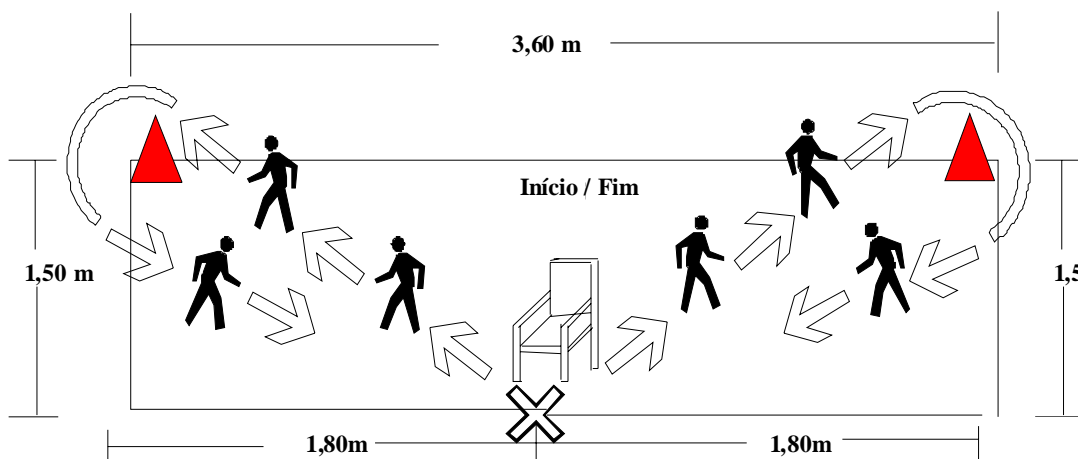


FIGURA 2: *Dinâmico e agilidade AAHPERD (Mazo et al., 2003).*

Coordenação (COO): AAHPERD

Verificar a coordenação dos idosos (A aplicação deverá ser realizada em local isolado/reservado).

Instrumentos: fita adesiva, cronômetro, três latas de refrigerante fechadas com areia.

Organização dos instrumentos: um pedaço de fita adesiva com 76,2 cm de comprimento foi fixado sobre uma mesa. Sobre a fita foram feitas 6 marcas com 12,7 cm equidistantes entre si, com a primeira e última marca a 6,35 cm de distância das extremidades da fita. Sobre cada uma das 6 marcas foi afixado, perpendicularmente à fita, um outro pedaço de fita adesiva com 7,6 cm de comprimento (Figura 3).

Posição do avaliado: o participante senta de frente para a mesa e usa sua mão dominante para realizar o teste. Se a mão dominante for a direita, uma lata de refrigerante era colocada na posição 1, a lata dois na posição 3 e, a lata três na posição 5. A mão direita foi colocada na lata 1, com o polegar para cima, estando o cotovelo flexionado num ângulo de 100 a 120 graus.

Posição do avaliador: de frente para o avaliado com cronômetro na mão.

Procedimento: quando o avaliador sinalizou, um cronômetro foi acionado e, o participante, virando a lata inverteu sua base de apoio, de forma que a lata 1 fosse colocada na posição 2; a lata 2 na posição 4 e; a lata 3 na posição 6. Sem perder tempo, o avaliado, estando agora com o polegar apontado para baixo, apanhou a lata 1 e inverteu novamente sua base, recolocando-a na posição 1 e, da mesma forma procedeu colocando a lata 2 na posição 3 e a lata 3 na posição 5, completando assim um circuito

Observação: uma tentativa equivale a realização do circuito duas vezes, sem interrupções. No caso do participante ser canhoto, o mesmo procedimento foi adotado, exceto que as latas foram colocadas a partir da esquerda, invertendo-se as posições. A cada participante foi concedidas duas tentativas de prática, seguidas por outras duas válidas para avaliação, sendo estas últimas duas anotadas até décimos de segundo, e considerado como resultado final o menor dos tempos obtidos.

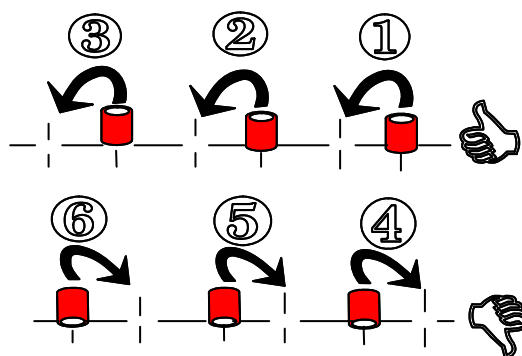


FIGURA 3: Ilustração gráfica do teste de coordenação (adaptada de OSNESS et al., 1990 apud Mazo et al., 2003).

Resistência aeróbica geral e habilidade de andar (RAG)

O teste consiste em caminhar 804,67 metros (1/2 milha) no menor tempo possível.

Instrumentos: Pista de atletismo, preferencialmente de 400 metros.

Posição do avaliado: Marca inicial da pista

Posição do avaliador: Na marca inicial da pista, com cronômetro na mão.

Procedimento: O participante deverá caminhar (sem correr) 804,67 metros, numa pista de atletismo de 400 m, o mais rápido possível.

Observação: Será registrado o tempo gasto para realizar tal tarefa foi anotado em minutos e segundos, e transformados para segundos.

Para o teste da milha o participante deverá continuar o percurso que fez acima.

Número de quedas a cada três meses. Os meses em que deverá ser anotado nas fichas é MARÇO – JUNHO – SETEMBRO – DEZEMBRO, sempre retroativo aos últimos três meses. O professor deverá perguntar para a turma a cada três meses se alguém caiu, o porque da queda e qual foi a consequência da queda.

É importante realizar o registro na ficha de avaliação sobre o número de quedas, mas não se deve esquecer de anotar atrás da ficha os detalhes da mesma.

3.3 Necessidade de Aquisição e Mediação de Dados

Como os testes das baterias AAHPERD apresentam vários pontos, com vários dados a serem armazenados é muito importante que a aquisição e a mediação de dados sejam bem elaboradas.

Nesse meio é visto que é trabalhado com muitos dados de várias pessoas e e esses dados servem para fazer avaliações de como o idoso se encontra em determinada situação. Por exemplo, como o idoso está com a força dos membros superiores numa determinada faixa etária, e não é só isso, com esses dados é possível fazer vários estudos sobre as condições físicas e de saúde de uma amostra real, com levantamentos que favoreceram a melhoria da bateria de exercícios, dos testes e em consequência da saúde e auto-estima do idoso.

3.4 Criação de Base de Dados

Sabe-se que são inúmeros dados a serem armazenados manualmente em uma folha de papel, e com vários nomes com esses vários dados, vejamos:

Um determinado idoso possui dados como seu nome, endereço, telefone, idade... ele fará alguns testes por ano, esses testes terão que ser anotados com várias entradas como velocidade, distância, tempo...

Cada processo, possui uma série de dados a serem guardados e até o momento da criação deste projeto não tínhamos nenhum auxílio as pessoas que faziam estes testes com os idosos no sentido de facilitar o trabalho das mesmas.

Então precisa-se de um banco de dados que possa armazenar todos esses pontos, além de um sistema que possa gerenciar esse banco e esses dados, daí é fácil perceber várias vantagens no tratamento desse problema através do computador, que terá ferramentas para tratar esses dados, e um sistema que é o alvo desse projeto, que fará buscas, gerenciamentos, relatórios etc.

4 METODOLOGIAS PARA MODELAGEM COMPUTACIONAL

4.1 Programação Orientada a Objeto

A programação Orientada a objetos (POO) é uma forma especial de programar, mais próxima de como expressaríamos as coisas na vida real do que outros tipos de programação.

Com a POO pensa-se as abstrações computacionais de uma maneira distinta. Escreve-se os programas em termos de objetos, propriedades e métodos. A seguir comenta-se as principais características da POO.⁴.

Existe alguma motivação para programarmos assim?

Durante anos os programadores se dedicaram a construir aplicações muito parecidas que resolviam, uma vez ou outra, os mesmos problemas. Para conseguir que os esforços dos programadores possam ser utilizados por outras pessoas foi criado a POO. Esta é uma série de normas de realizar as coisas de maneira com que outras pessoas possam utilizá-las e adiantar seu trabalho, de maneira que se consiga que o código possa ser reutilizado.

A POO é uma forma especial de pensar, às vezes subjetiva de quem a programa. Embora possa-se fazer os programas de formas distintas, nem todas

⁴ Criar Web, O que é a POO Disponível em:

<<http://www.criarweb.com/artigos/215.php?manual=11>>

elas são corretas. O difícil não é programar orientado a objetos e sim, programar bem, para se aproveitar todas as vantagens desta metodologia.

Como se pensa em objetos?

Pensar em termos de objetos é muito parecido a como faríamos na vida real. Por exemplo, considerando um carro para dar um modelo em um esquema de POO. O carro é o elemento principal que tem uma série de características, como poderiam ser a cor, o modelo ou a marca. Ademais tem uma série de funcionalidades associadas, como podem ser andar, parar ou estacionar.

Então em um esquema POO o carro seria o objeto, as propriedades seriam as características como a cor ou o modelo e os métodos seriam as funcionalidades associadas como andar ou parar.

Classes em POO:

As classes são declarações de objetos, também se pode definir como abstrações de objetos. Isto quer dizer que a definição de um objeto é a classe. Quando programa-se um objeto e define-se suas características e funcionalidades, na verdade, o que se está fazendo é programar uma classe.

Propriedades em classes:

As propriedades ou atributos são as características dos objetos. Quando se define uma propriedade, normalmente, se especifica seu nome e seu tipo.

Pode-se ter a idéia de que as propriedades são algo assim como as variáveis onde se armazena os dados relacionados com os objetos.

Métodos nas classes:

São as funcionalidades associadas aos objetos. Quando se esta programando as classes estas são chamadas de métodos. Os métodos são como funções que estão associadas a um objeto.

Objetos em POO:

Os objetos são exemplares de uma classe qualquer. Quando se cria um exemplar tem-se que especificar a classe a partir da qual se criará. Esta ação de criar um objeto a partir de uma classe chama-se *instance* (que significa em inglês exemplificar). Por exemplo, um objeto da classe carro é um Golf (Volkswagen). O conceito ou definição de carro seria a classe, mas um modelo de carro em particular chama-se de objeto.

Para criar um objeto tem-se que escrever uma instrução especial que possa ser distinta dependendo da linguagem de programação que se empregue, mas será algo parecido a isto.

```
meuCarro = new Carro()
```

Com a palavra *new* especifica-se uma *instance* da classe que continua a seguir. Dentro dos parênteses poderia-se colocar parâmetros com os quais se inicia o objeto da classe carro.

Estados em objetos:

Quando se tem um objeto suas propriedades tomam valores. Por exemplo, quando um carro tem a propriedade cor, esta tomará um valor em concreto, como vermelho, cinza etc. O valor concreto de uma propriedade de um objeto chama-se estado.

Para acessar a um estado de um objeto, para ver seu valor ou mudá-lo utiliza-se o operador ponto.

```
meuCarro.cor = vermelho
```

O objeto é “meuCarro”, logo coloca-se o operador ponto e por último o nome da propriedade a qual se deseja acessar. Neste exemplo, se esta mudando o valor do estado da propriedade do objeto a vermelho com uma simples atribuição.

Mensagens em objetos:

Uma mensagem em um objeto é a ação de efetuar uma chamada a um método. Por exemplo, quando diz-se a um objeto carro para andar, se lhe está passando a mensagem "ande".

Para mandar mensagens aos objetos utiliza-se o operador ponto, seguido do método que deseja-se utilizar.

```
meuCarro.andar()
```

Neste exemplo, passa-se a mensagem andar(). Deve-se colocar parênteses, assim como com qualquer chamada a uma função. Dentro dos parênteses iriam os parâmetros.

Herança:

Também na POO existem mecanismos como a herança, que é uma das possibilidades mais potentes desta metodologia.

A herança serve para criar objetos que incorporem propriedades e métodos de outros objetos. Assim, pode-se construir objetos a partir de outros sem ter que reescrevê-los integralmente.

Neste trabalho:

Neste trabalho utiliza-se uma linguagem orientada a objeto, o Java 1.5. Isto foi feito para facilitar a continuidade do mesmo em futuras atualizações.

4.2 Linguagem Java

O Java é ao mesmo tempo um ambiente e uma linguagem de programação, produzido pela *Sun Microsystems, Inc.* Trata-se de mais um

representante da nova geração de linguagens orientadas a objetos e foi projetado para resolver os problemas da área de programação cliente/servidor.⁵

O Java foi criado como parte de um grande projeto da *Sun*, cuja missão era desenvolver aplicativos complexos e avançados para aplicação em pequenos dispositivos eletrônicos. Esses dispositivos são sistemas portáteis, distribuídos, confiáveis e incorporados em tempo real, dirigidos aos consumidores em geral. As primeiras tentativas de criação de tais dispositivos receberam nomes como Digital Assistance e Portable Data Assistents(PDAs).

A primeira tentativa da *Sun* com esse aplicativos deveria ter sido implementada em C++. Mas em consequência de problemas de compiladores uma lista cada vez maior de problemas com o C++ - principalmente vazamentos de memória e vários problemas de herança - a *Sun* e implementou uma nova linguagem, o Java. Desde o seu início em 1992, surgiram muitas outras aplicações para o Java. O Java tem a capacidade de criar programas auxiliares, que podem ser usados para montar documentos HTML que são particularmente interessantes para a programação para *web*.

A arquitetura do Java são compilados em um código de bytes independente de arquitetura. Esse código de bytes pode então ser executado em qualquer plataforma que suporte um interpretador Java. O Java requer somente

⁵ A História da Linguagem JAVA, Disponível em:

<<http://campus.fortunecity.com/psychology/196/javatudo.html>>

uma fonte e um binário e, mesmo assim, é capaz de funcionar em diversas plataformas, o que faz dele um sonho de todos os que realizam manutenção em programas.

O Java apresenta algumas características marcantes: ele é orientado a objeto, é distribuído, é ao mesmo tempo compilado e interpretado, é independente de arquitetura, é portátil, é multi-tarefa, é dinâmico, é robusto, é seguro, etc. Ou seja possui uma série de características marcantes e úteis.

O Java provavelmente é um dos melhores ambientes de programação do mundo e além de tudo ele é *freeware*, ou seja não precisamos pagar nada para poder utiliza-lo.

Outro motivo bastante interessante é sua portabilidade, como não sabemos em que ambientes o programa será instalado, nada melhor do que um compilador/interpretador que aceita ser instalado em praticamente qualquer sistema operacional.

Acima de tudo, como o Java é feito para uso seguro na Internet, em um futuro próximo é desejável os bancos de dados estivessem interligados pela *web*, e nada melhor que um ambiente que nos acarretará uma possível migração para esse ambiente interligado é o Java e por isto foi escolhido para a implementação.

Utilizar-se-á o Java 1.5, que é a versão mais recente do Java. Para a sua implementação usar-se-á o *Netbeans 4.0* que é uma IDE *Open-Source* desenvolvida em conjunto com a *Sun Microsystems*.

4.3 SQL, MySQL e Banco de Dados

O MySQL é um sistema de banco de dados baseado em SQL que vem sendo largamente utilizado em aplicações para web, pois possui várias características favoráveis a este tipo de aplicação, tais como⁶:

O MySQL é desenvolvido com a constante preocupação com a velocidade do sistema, pois a preocupação com a velocidade em aplicações *web* é uma constante “dor de cabeça” para os desenvolvedores. A Internet hoje possui uma arquitetura muito heterogênea, o que gera lentidão no processamento. Por isso, um dos principais objetivos do desenvolvedor é tentar compensar o atraso na resposta do sistema procurando otimizar o tempo de processamento do mesmo.

Assim como o Java, o MySQL também é gratuito e possui código aberto, gerando um custo zero para implementação.

Apesar ser um sistema relativamente simples, o MySQL possui suporte a maioria dos comandos da linguagem SQL padrão, facilitando assim para os desenvolvedores.

As aplicações em rede são cada dia mais numerosas e versáteis. Em muitos casos, o esquema básico de operação é uma série de *scripts* que dirigem o comportamento de uma base de dados.

⁶ Tecnologia - Inovativa Internet Solutions. Disponível em:

<<http://www.enovativa.com.br/tecnologia.html>>

Devido à diversidade de linguagens e de base de dados existentes, a maneira de comunicação entre elas torna-se complicada. Porém, a existência de padrões permite realizar as operações básicas de uma forma universal.

O *Structured Query Language* – SQL é uma linguagem padrão de comunicação com base de dados. É, portanto, uma linguagem normalizada que permite trabalhar com qualquer tipo de linguagem (como, por exemplo, ASP ou PHP) em combinação com qualquer tipo de base de dados (*MS Access, SQL Server, MySQL* etc.).

O fato de ser padrão não quer dizer que seja idêntico para cada base de dados. Na prática, determinadas bases de dados implementam funções específicas que não têm necessariamente que funcionar em outras.

Banco de Dados ⁷:

Atualmente há gigantescas bases de dados gerenciando nossas vidas. Por exemplo, nossa conta bancária faz parte de uma coleção imensa de contas bancárias de nosso banco. Nosso Título Eleitoral ou nosso Cadastro de Pessoa Física, certamente estão armazenados em Bancos de Dados colossais. Quando sacamos dinheiro no Caixa Eletrônico de nosso banco, nosso saldo e as movimentações existentes em nossa conta bancária já estão à nossa disposição.

⁷ Banco de Dados. Disponível em:

<http://www.webinformation.hpg.ig.com.br/computer_bancodedados.htm>

Nestas situações existe uma necessidade em se realizar o armazenamento de uma série de informações que não se encontram efetivamente isoladas umas das outras, ou seja, existe uma ampla gama de dados que se referem a relacionamentos existentes entre as informações a serem manipuladas.

Estes Bancos de Dados, além de manterem todo este volume de dados organizado, também devem permitir atualizações, inclusões e exclusões do volume de dados, sem nunca perder a consistência. E na maioria das vezes se está lidando com acessos concorrentes a várias tabelas de banco de dados, e algumas vezes, com mais de um acesso ao mesmo registro de uma mesma tabela.

Um Banco de Dados contém os dados dispostos numa ordem pré-determinada em função de um projeto de sistema, sempre para um propósito muito bem definido. Este representará sempre aspectos do Mundo Real.

Assim sendo, uma Base de Dados (ou Banco de Dados, ou ainda BD) é uma fonte de onde pode-se extrair uma vasta gama de informações derivadas, que possui um nível de interação com eventos como o Mundo Real que representa. A forma mais comum de interação Usuário e Banco de Dados, dá-se através de sistemas específicos que por sua vez acessam o volume de informações geralmente através da linguagem SQL.

Os Administradores de Banco de Dados (DBA) são responsáveis pelo controle ao acesso aos dados e pela coordenação da utilização do BD. Já os projetistas de Banco de Dados (DBP) são analistas que identificam os dados a

serem armazenados em um Banco de Dados e pela forma como estes serão representados.

Os Analistas e Programadores de Desenvolvimento, criam sistemas que acessam os dados da forma necessária ao Usuário Final, que é aquele que interage diretamente com o Banco de Dados.

No sistema aqui proposto tem-se um banco de dados relacional que guarda as informações importantes e facilita a entrada e saída de dados. Este banco de dados é “enxuto” e simples, porém com o melhor custo-benefício possível, armazenando de forma eficiente todos os dados do usuário.

Dbdesigner 4.0 ⁸:

O *Dbdesigner 4.0* é um editor visual para criação de banco de dados *mySQL*. Este integra criação, modelagem, desenvolvimento e manutenção dos bancos em um ambiente simples e agradável. Comparável com produtos como Oracle's Designer, IBM's Relational Rose, CA Erwin. O *DBDesigner* é *OpenSource* distribuído sobre a licença GPL.

4.4 XP (*Extreme Programming*)

Extreme Programming (XP)⁹ - é a mais conhecida das metodologias de engenharia de software ditas "ágeis". As raízes de XP está na comunidade

⁸ Webmasters Online – o Portal dos Webmasters Disponível em: <<http://www.wmonline.com.br/>>

⁹ Disponível em: <<http://xp.edugraf.ufsc.br/>>

Smalltalk, e na estreita colaboração desenvolvida por Kent Beck e Ward Cunningham (o criador do *Wiki*) no final dos anos 80. Ambos refinaram suas práticas nos anos seguintes, estendendo a abordagem de um processo de desenvolvimento que fosse tanto adaptativo como orientado a pessoas.

Extreme Programming (XP) é um processo de desenvolvimento que possibilita a criação de software de alta qualidade, de maneira ágil, econômica e flexível. Vem sendo adotado com enorme sucesso na Europa, nos Estados Unidos e, mais recentemente, no Brasil.

Cada vez mais empresas convivem com ambientes de negócios que requerem mudanças freqüentes em seus processos, as quais afetam os projetos de software. Os processos de desenvolvimento tradicionais são caracterizados por uma grande quantidade de atividades e artefatos que buscam proteger o software contra mudanças, o que faz pouco ou nenhum sentido, visto que os projetos devem se adaptar a tais mudanças ao invés de evitá-las.

A passagem de prática informal para uma metodologia tem como marco inicial o projeto C3, desenvolvido por Kent Beck em 1996, usando novos conceitos em desenvolvimento de *software*. O princípio é que você pode aprimorar o processo de desenvolvimento de software em quatro dimensões principais:

- melhorar a comunicação
- buscar a simplicidade
- obter *feedback* constante sobre o andamento
- proceder com coragem

A Comunicação, a Simplicidade, o *Feedback* e a Coragem são então os "valores" apregoados pela "filosofia XP".

Os programadores XP se comunicam com o cliente e com seus colegas programadores. Eles mantêm o projeto simples e claro. Eles obtêm *feedback* testando o software desde seu primeiro dia. Eles distribuem o sistema aos usuários tão cedo quanto possível, implementando as modificações que forem solicitadas. Com esses fundamentos, eles estão aptos a responder corajosamente a requisitos e tecnologias instáveis.

Os testes são um dos principais fundamentos de XP. Embora todas as metodologias mencionem os testes, a do XP se destaca ao colocar os testes na base do desenvolvimento, com cada programador escrevendo código de testes ao mesmo tempo em que escreve código de produção. Os testes XP são ligados a um processo de integração que produz uma plataforma muito estável para o desenvolvimento futuro.

Sobre essa plataforma, XP constrói um processo evolucionário de projeto, baseado em refazer um sistema base a cada iteração. O resultado é um disciplinado processo de projeto. Surpreendentemente, a metodologia XP combina essa disciplina com alto grau de adaptatividade, o que a torna a mais bem desenvolvida das metodologias adaptativas.

Kent Beck escreveu *Extreme Programming Explained*, o manifesto-chave de XP, aonde explica a racionalidade por trás da metodologia. Existem muitos livros sobre XP, e um grande número de recursos *web*.

O XP concentra os esforços da equipe de desenvolvimento em atividades que geram resultados rapidamente na forma de software intensamente testado e alinhado às necessidades de seus usuários. Além disso, simplifica e organiza o trabalho combinando técnicas comprovadamente eficazes e eliminando atividades redundantes. Por fim, reduz o risco dos projetos desenvolvendo software de forma iterativa e reavaliando permanentemente as prioridades dos usuários.

A programação XP foi escolhida pois funciona extremamente bem para as modalidades deste trabalho. Devido a dinamicidade deste é necessário um tipo de programação que permita os testes mais imediatos e que trabalhe de uma forma rápida.

5 MODELAGEM PROPOSTA / IMPLEMENTAÇÃO

A expectativa de vida na sociedade atual está aumentando, e com isso há um crescimento da população um pouco mais velha. Assim, para se promover uma melhor qualidade de vida deste grupo de pessoas é importante incentivá-las à prática de atividades físicas. Para isso é necessário dispor-se de testes sobre esses exercícios e seus utilizadores para um efetivo controle dos mesmos sobre a saúde desta população. Desta forma pode-se saber o quão bem o mesmo está fazendo para a pessoa e quais seriam as melhores baterias de testes para um indivíduo em particular.

A proposta deste trabalho é desenvolver um sistema capaz de gerenciar esses testes

Em um primeiro plano a bateria é a AAHPERD (explicada anteriormente).

O trabalho consiste em armazenar os dados obtidos nas baterias de testes em um banco de dados para que esses dados possam ser utilizados no futuro para comparações e pesquisas.

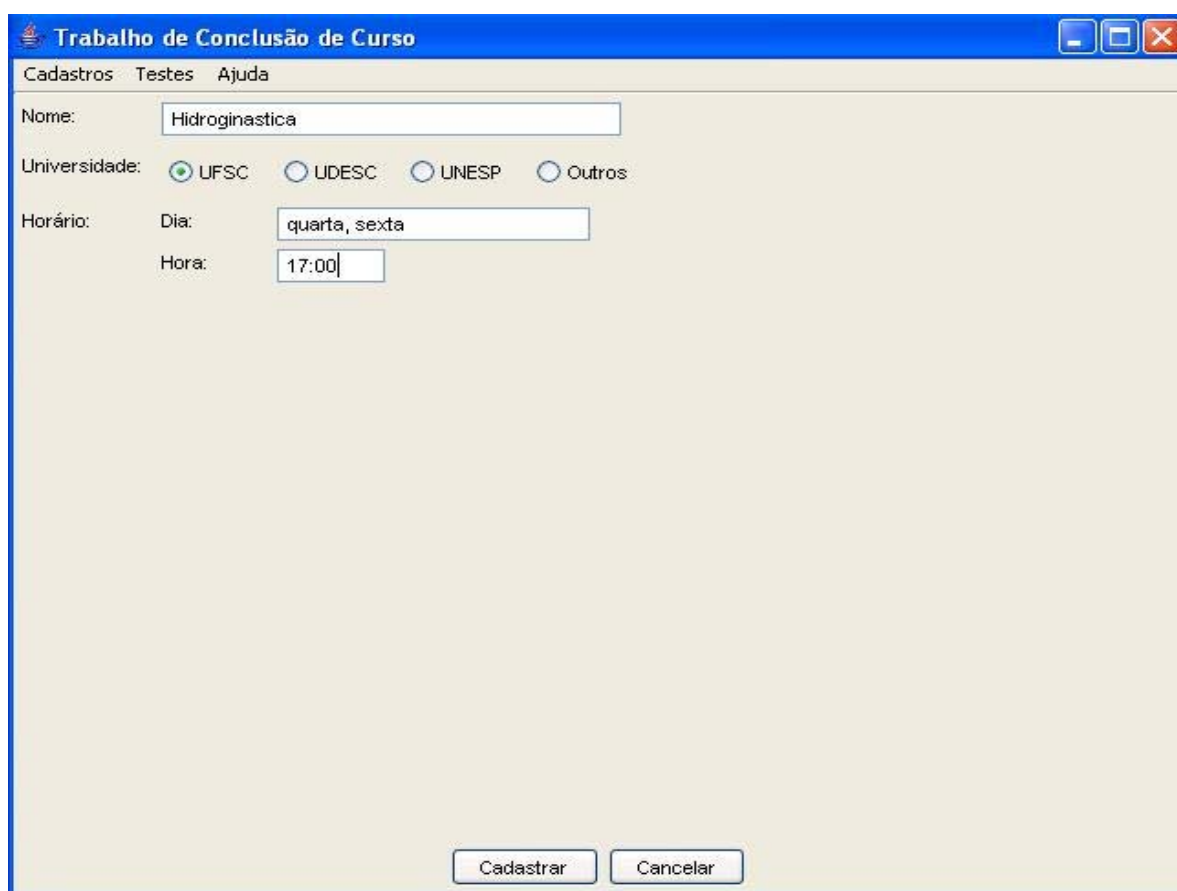
São efetuados alguns tratamentos nesses dados com o objetivo de gerar um relatório que demonstrela a condição física do idoso. Esta é dividida em cinco categorias: muito fraco, fraco, regular, bom e muito bom. Também será possível observar a evolução do idoso.

5.1 Use Cases:

Use Case Cadastro de Turmas.

O primeiro passo ao se iniciar a utilização do programa é cadastrar as turmas existentes, toda pessoa deve ser relacionado com uma turma, as turmas ficam armazenadas no banco de dados e podem ser adicionadas novas turmas a qualquer momento.

As turmas são compostas de: nome, universidade (UFSC, UDESC, UNESP e Outras), dias e horários.



The image shows a screenshot of a software application window titled "Trabalho de Conclusão de Curso". The window has a blue title bar and standard Windows window controls (minimize, maximize, close) in the top right corner. Below the title bar, there are three menu items: "Cadastros", "Testes", and "Ajuda". The main area of the window is a form for registering a class. It contains the following fields and options:

- Nome:** A text input field containing the text "Hidroginastica".
- Universidade:** A group of radio buttons with the following options: UFSC, UDESC, UNESP, and Outros.
- Horário:** A sub-section with two input fields:
 - Dia:** A text input field containing the text "quarta, sexta".
 - Hora:** A text input field containing the text "17:00".

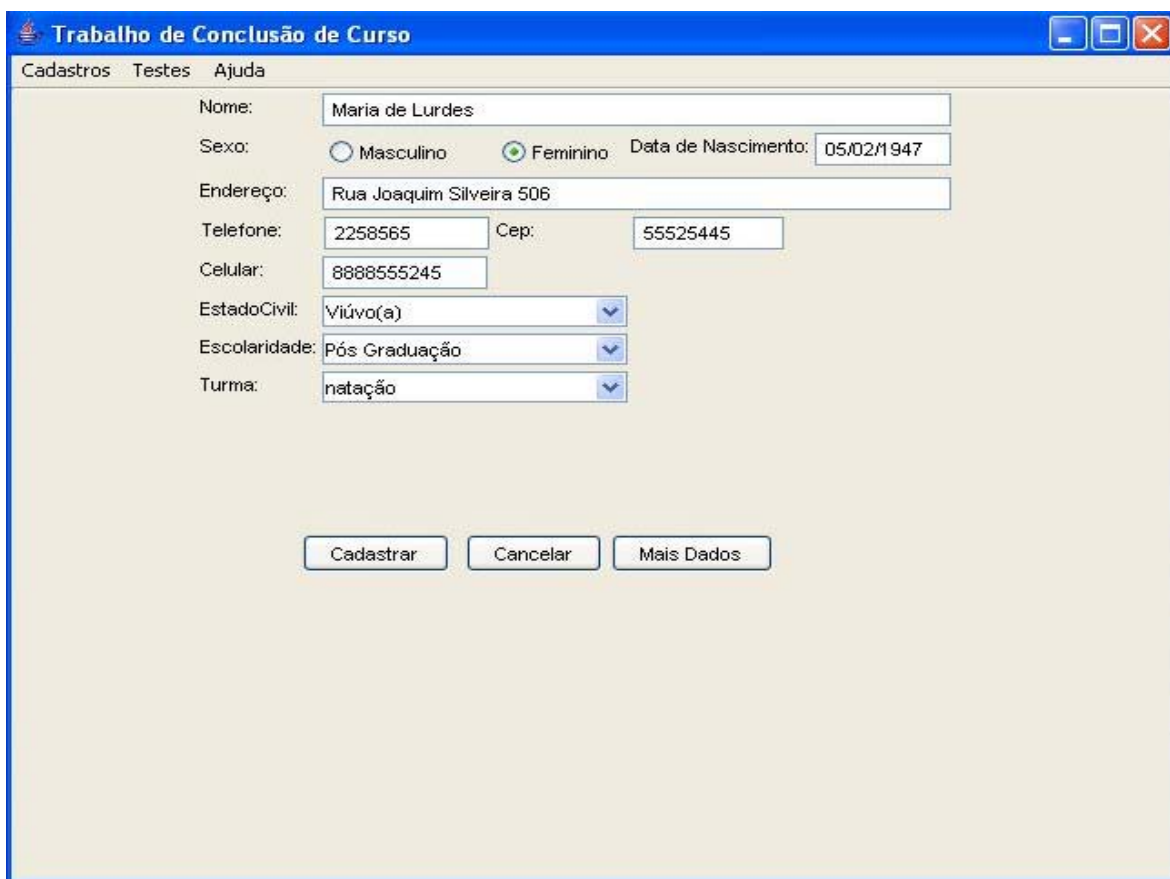
At the bottom of the form, there are two buttons: "Cadastrar" and "Cancelar".

FIGURA 4: Cadastro de Turmas

Use Case Cadastro de Pessoas

O segundo passo é cadastrar as pessoas, todos os dados das pessoas ficarão armazenados no banco de dados e depois poderão ser acessados e modificados.

O cadastro das pessoas é composto de: nome, sexo, data de nascimento, endereço, cep, telefone, celular, escolaridade (primeiro grau, segundo grau, superior), estado civil (casado(a), solteiro(a) e viúvo(a)) e turma (o programa disponibiliza para escolha todas as turmas cadastradas).



The image shows a screenshot of a software application window titled "Trabalho de Conclusão de Curso". The window has a menu bar with "Cadastros", "Testes", and "Ajuda". The main area contains a registration form for a person. The form fields are as follows:

Nome:	Maria de Lurdes		
Sexo:	<input type="radio"/> Masculino	<input checked="" type="radio"/> Feminino	Data de Nascimento: 05/02/1947
Endereço:	Rua Joaquim Silveira 506		
Telefone:	2258565	Cep:	55525445
Celular:	8888555245		
Estado Civil:	Viúvo(a)		
Escolaridade:	Pós Graduação		
Turma:	natação		

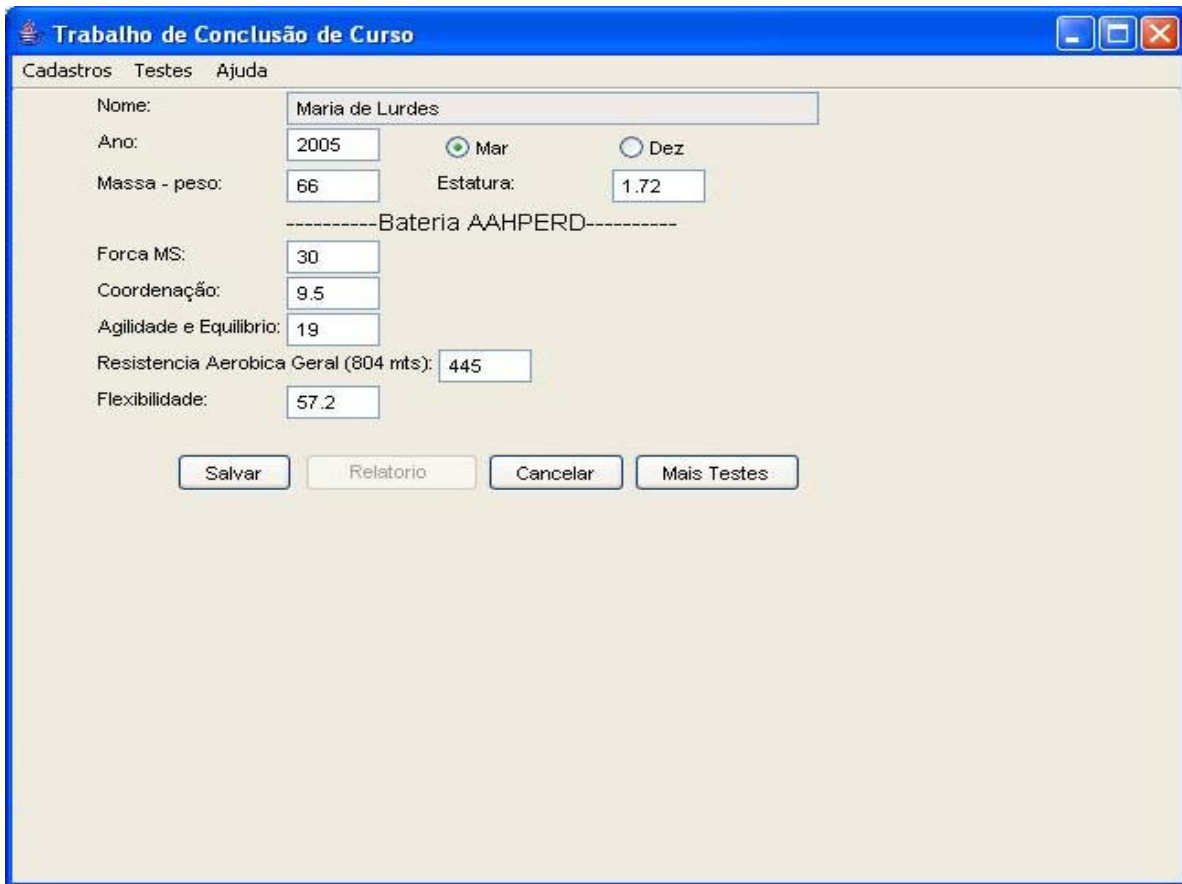
At the bottom of the form, there are three buttons: "Cadastrar", "Cancelar", and "Mais Dados".

FIGURA 5: Cadastro de Pessoa.

Use Case Testes

No terceiro passo com as pessoas já cadastradas pode se entrar com os dados dos testes AAHPERD, esses dados ficarão armazenados no banco de dados.

Os testes AAHPERD são: Força MS, Coordenação, Agilidade e Equilíbrio, Resistência Aeróbica Geral (804mts) e Equilíbrio.



The screenshot shows a software window titled "Trabalho de Conclusão de Curso" with a menu bar containing "Cadastros", "Testes", and "Ajuda". The main area is a form for entering test results. The form fields are as follows:

Nome:	Maria de Lurdes		
Ano:	2005	<input checked="" type="radio"/> Mar	<input type="radio"/> Dez
Massa - peso:	66	Estatura:	1.72
-----Bateria AAHPERD-----			
Força MS:	30		
Coordenação:	9.5		
Agilidade e Equilíbrio:	19		
Resistencia Aerobica Geral (804 mts):	445		
Flexibilidade:	57.2		

At the bottom of the form, there are four buttons: "Salvar", "Relatorio", "Cancelar", and "Mais Testes".

FIGURA 6: Testes.

Use Case Gerar Relatório

No quarto passo logo após os dados terem sido inseridos no programa pode se gerar um relatório que mostra os resultados de todos os testes e a condição física do idoso.

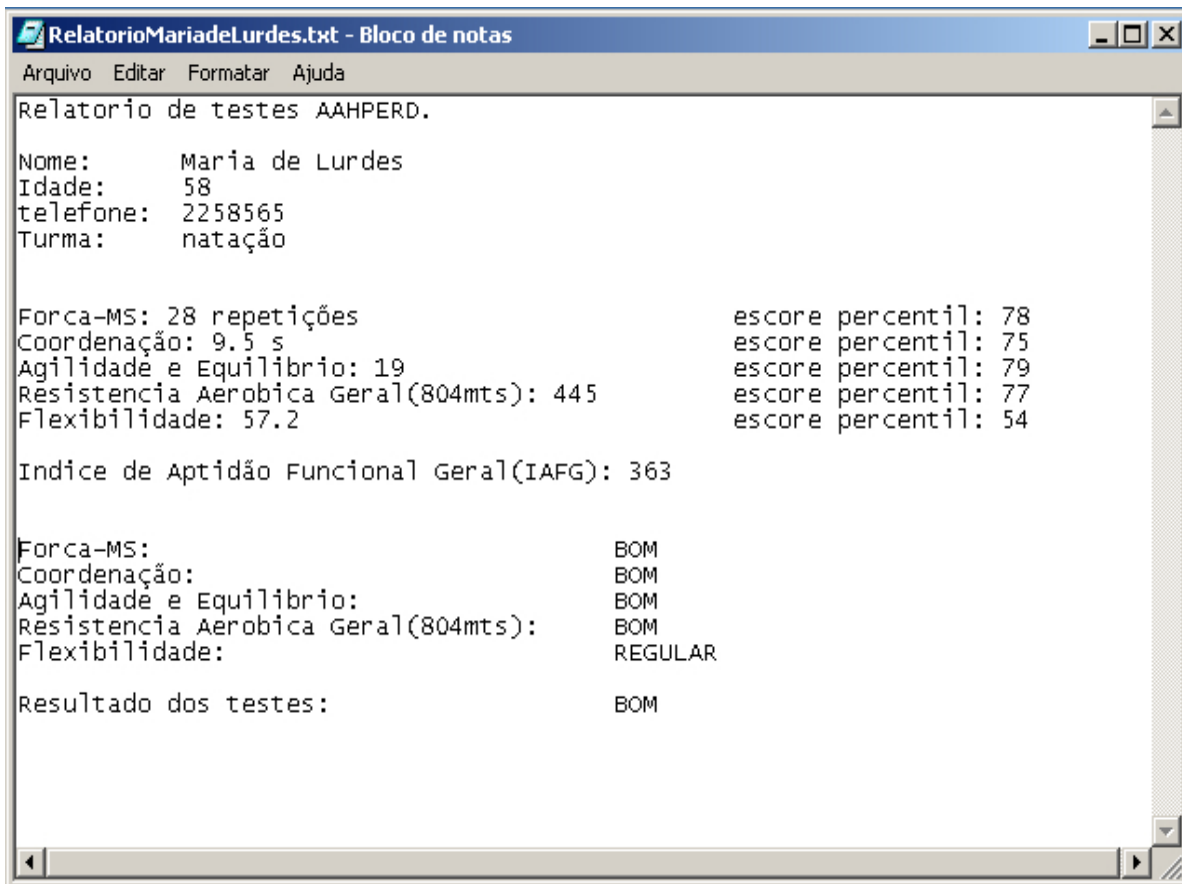


FIGURA 7: *Relatório*

Use Case Buscas

No quinto passo depois que alguns testes já foram inseridos pode se realizar buscas desses dados.

As buscas podem levar em consideração: idade, turma e sexo.

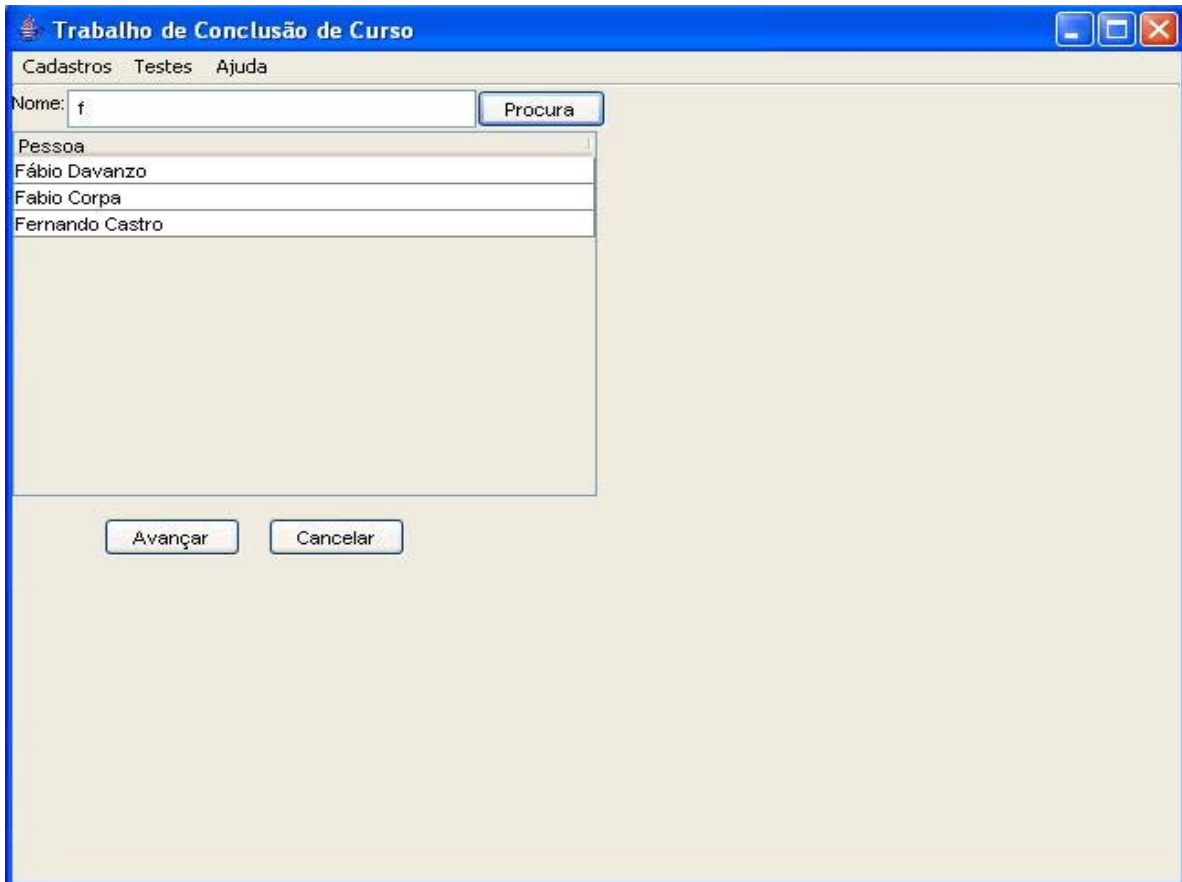


FIGURA 8: *Buscas.*

5.2 Modelagem do Banco de Dados

O Banco de Dados foi criado com três tabelas:

Pessoa: que possui como identificador o idPessoa que é do tipo *INTEGER*, e possui a chave estrangeira Turma_idTurma que também é do tipo *INTEGER*. Os atributos da tabela pessoas são: nome que é do tipo *VARCHAR*, sexo que é do tipo *CHAR*, endereço que é do tipo *VARCHAR*, telefone que é do tipo *VARCHAR*, data_nascimento que é do tipo *DATE*, cep que é do tipo *VARCHAR*, celular que é do tipo *VARCHAR*, estado_civil que é do tipo *INTEGER*, escolaridade que é do tipo *INTEGER*.

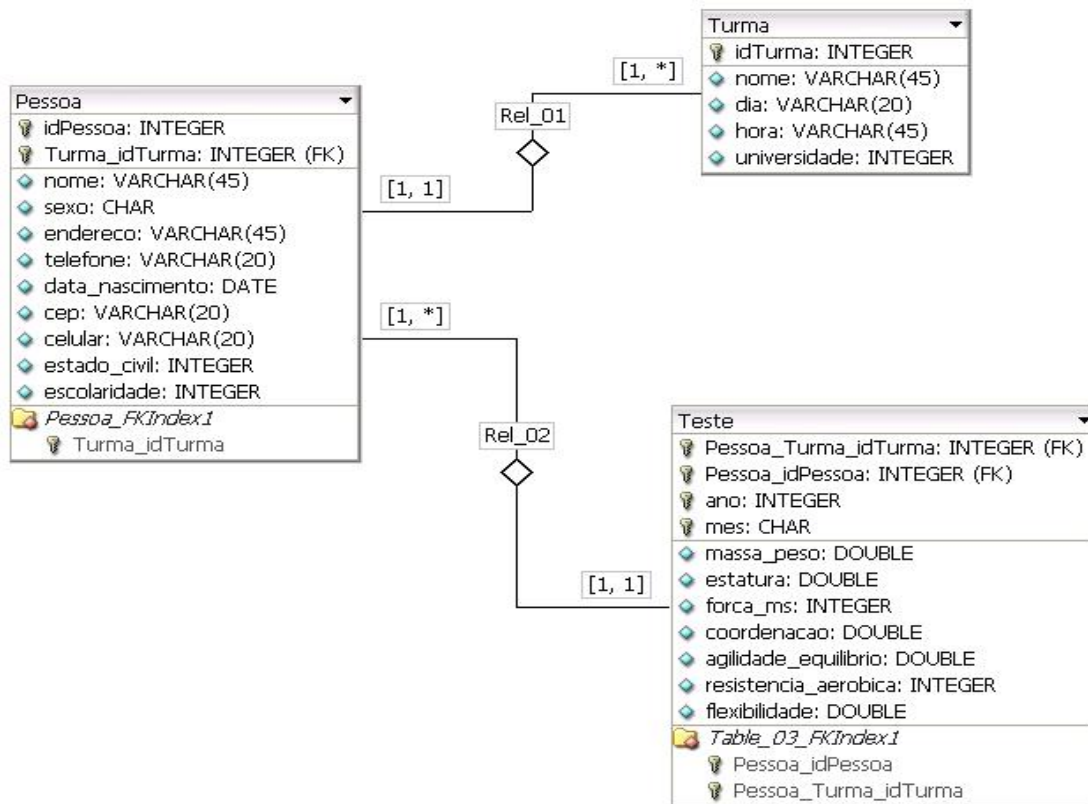
Turma: que possui como identificador o idTurma que é do tipo *INTEGER*. A tabela Turma possui os seguintes atributos: nome que é do tipo *VARCHAR*, dia que é do tipo *VARCHAR*, hora que é do tipo *VARCHAR*, universidade que é do tipo *INTEGER*.

Teste : que possui as chaves estrangeiras; Pessoa_Turma_idTurma que é do tipo *INTEGER* e Pessoa_idPessoa que é do tipo *INTEGER*. Possui o ano que é do tipo *INTEGER*, mes que é do tipo *CHAR*. A Tabela Teste possui os seguintes atributos: massa_peso que é do tipo *DOUBLE*, estatura que é do tipo *DOUBLE*, forca_ms que é do tipo *INTEGER*, coordenacao que é do tipo *DOUBLE*, agilidade_equilibrio que é do tipo *DOUBLE*, resistencia_aerobica que é do tipo *INTEGER*, flexibilidade que é do tipo *DOUBLE*.

As Relações entre as tabelas seguem o seguinte contexto: Um elemento da tabela Pessoa poderá estar ligado a um ou mais elementos da tabela Turma, e um elemento da tabela Turma poderá estar ligado a um ou mais elementos da tabela Pessoa. Um elemento da tabela Pessoa poderá estar ligado a um ou mais

elementos da tabela Teste, e um elemento da tabela Teste só poderá estar ligado a um elemento da tabela Pessoa.

Segue abaixo as Tabelas e suas ligações:



FIGURA[9]: Tabelas.

6 CONCLUSÃO E SUGESTÕES PARA FUTUROS TRABALHOS

O projeto permite a realização de diversos projetos futuros.

Uma idéia que seria muito útil e interessante seria criar um banco de dados via web, que poderia interligar todas as universidades ou instituições onde os testes são realizados sendo assim se teria um banco de dados muito mais completo e poderia se realizar pesquisas muito mais detalhadas e precisas, pois haveria um número muito maior de idosos de diversos lugares e regiões.

O que pode ser bom até mesmo para o próprio *software*, adicionando novos tipos de testes, realizando assim cadastros de pessoas mais detalhadas e precisos, o que acarreta realizar buscas mais detalhadas que auxiliam nas pesquisas. O *software* já tem um campo em aberto para a inserção de novos testes e novos dados cadastrais

O *software* não se encontra completamente pronto já que o projeto se estenderá por mais 6 meses.

Foi um trabalho robusto, com um grande tempo tomado para detalhes, mas que depois chegou a um ponto em comum interessante, tanto para os desenvolvedores como para a equipe que fará a coleta de dados dos idosos.

Os resultados até agora estão bons, e o trabalho terá continuidade para incorporar outras facilidades computacionais que permitam torná-lo um instrumento mais adequado à pesquisa a que se destina na área de esportes.

O tempo não esteve do nosso lado, mas a ajuda de pessoas como as das professoras Tânia, Giovana e Lúcia foi imprescindível para que alcançássemos o sucesso.

7 REFERÊNCIAS:

[1] Idade Ativa: revista eletrônica da Terceira Idade. Disponível em:

<http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>.

Acesso em: abril 2005.

[2] Mazo, Giovana Z., Lopes, Marize A. e Benedetti, Tânia B. A Atividade Física e o Idoso: concepção gerontológica 2003)

[3] Criar Web, O que é a POO?. Disponível em:

<<http://www.criarweb.com/artigos/215.php?manual=11>>.

Acesso em: Abril 2005.

[4] Tecnologia - Enovativa *Internet Solutions*. Disponível em:

<<http://www.enovativa.com.br/tecnologia.html>>.

Acesso em: Abril 2005.

[5] Banco de Dados. Disponível em:

<http://www.webinformation.hpg.ig.com.br/computer_bancodedados.htm>.

Acesso em: Abril 2005.

[6] Java Technology. Disponível em:

<<http://java.sun.com/>>.

Acesso em: Abril 2005

[7] MySQL AB : *The World's Most Popular Open Source Database*. Disponível em:
<<http://www.mysql.com/>>.

Acesso em: abril 2005.

[8] DB *Design Group*. Disponível em:

<<http://www.dbdesign.com/>>

Acesso em: abril 2005.

[9] AAHPERD (*American Alliance for Health, Physical Education, Recreation*)

1984. Disponível em:

<<http://www.aahperd.org/>>

Acesso em: abril 2005.

8 APÊNDICE

8.1 Código Fonte

Obs: Classes listadas em ordem alfabética.

Action Demo:

```
/*
 * This relies on having the Java Look and Feel Graphics Repository
 * (jlfgr-1_0.jar) in the class path. You can download this file
 * from http://developer.java.sun.com/developer/techDocs/hi/repository/.
 * Put it in the class path using one of the following commands
 * (assuming jlfgr-1_0.jar is in a subdirectory named jars):
 *
 *   java -cp .:jars/jlfgr-1_0.jar ActionDemo [Microsoft Windows]
 *   java -cp .:jars/jlfgr-1_0.jar ActionDemo [UNIX]
 *
 * I needed to put quotation marks around the path, since I use a
 * UNIX-emulating shell on Win32:
 *
 *   java -cp ".:jars/jlfgr-1_0.jar" ActionDemo [UNIX shell on Win32]
 */

import javax.swing.AbstractAction;
import javax.swing.Action;

import javax.swing.JToolBar;
import javax.swing.JButton;
import javax.swing.ImageIcon;

import javax.swing.JMenuItem;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JMenu;
import javax.swing.JMenuBar;

import javax.swing.JFrame;
import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import javax.swing.JPanel;

import java.awt.*;
import java.awt.event.*;

public class ActionDemo extends JPanel
    implements ItemListener {
    protected JTextArea textArea;
    protected String newline = "\n";
    protected Action leftAction, middleAction, rightAction;
    protected JCheckBoxMenuItem[] cbmi;

    public ActionDemo() {
        super(new BorderLayout());

        //Create a scrolled text area.
        textArea = new JTextArea(5, 30);
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);

        //Lay out the content pane.
```

```

setPreferredSize(new Dimension(450, 150));
add(scrollPane, BorderLayout.CENTER);

//Create the actions shared by the toolbar and menu.
leftAction = new LeftAction( "Go left",
    createNavigationIcon("Back24"),
    "This is the left button.",
    new Integer(KeyEvent.VK_L));
middleAction = new MiddleAction("Do something",
    createNavigationIcon("Up24"),
    "This is the middle button.",
    new Integer(KeyEvent.VK_M));
rightAction = new RightAction( "Go right",
    createNavigationIcon("Forward24"),
    "This is the right button.",
    new Integer(KeyEvent.VK_R));
}

/** Returns an ImageIcon, or null if the path was invalid. */
protected static ImageIcon createNavigationIcon(String imageName) {
    String imgLocation = "toolbarButtonGraphics/navigation/"
        + imageName
        + ".gif";
    java.net.URL imageURL = ActionDemo.class.getResource(imgLocation);

    if (imageURL == null) {
        System.err.println("Resource not found: "
            + imgLocation);
        return null;
    } else {
        return new ImageIcon(imageURL);
    }
}

public JMenuBar createMenuBar() {
    JMenuItem menuItem = null;
    JMenuBar menuBar;

    //Create the menu bar.
    menuBar = new JMenuBar();

    //Create the first menu.
    JMenu mainMenu = new JMenu("Menu");

    Action[] actions = {leftAction, middleAction, rightAction};
    for (int i = 0; i < actions.length; i++) {
        menuItem = new JMenuItem(actions[i]);
        menuItem.setIcon(null); //arbitrarily chose not to use icon
        mainMenu.add(menuItem);
    }

    //Set up the menu bar.
    menuBar.add(mainMenu);
    menuBar.add(createAbleMenu());
    return menuBar;
}

public void createToolBar() {
    JButton button = null;

    //Create the toolbar.
    JToolBar toolBar = new JToolBar();
    add(toolBar, BorderLayout.PAGE_START);
}

```

```

        //first button
        button = new JButton(leftAction);
        if (button.getIcon() != null) {
            button.setText(""); //an icon-only button
        }
        toolBar.add(button);

        //second button
        button = new JButton(middleAction);
        if (button.getIcon() != null) {
            button.setText(""); //an icon-only button
        }
        toolBar.add(button);

        //third button
        button = new JButton(rightAction);
        if (button.getIcon() != null) {
            button.setText(""); //an icon-only button
        }
        toolBar.add(button);
    }

    protected JMenu createAbleMenu() {
        JMenu ableMenu = new JMenu("Action State");
        cbmi = new JCheckBoxMenuItem[3];

        cbmi[0] = new JCheckBoxMenuItem("First action enabled");
        cbmi[1] = new JCheckBoxMenuItem("Second action enabled");
        cbmi[2] = new JCheckBoxMenuItem("Third action enabled");

        for (int i = 0; i < cbmi.length; i++) {
            cbmi[i].setSelected(true);
            cbmi[i].addItemListener(this);
            ableMenu.add(cbmi[i]);
        }

        return ableMenu;
    }

    public void itemStateChanged(ItemEvent e) {
        JCheckBoxMenuItem mi = (JCheckBoxMenuItem)(e.getSource());
        boolean selected =
            (e.getStateChange() == ItemEvent.SELECTED);

        //Set the enabled state of the appropriate Action.
        if (mi == cbmi[0]) {
            leftAction.setEnabled(selected);
        } else if (mi == cbmi[1]) {
            middleAction.setEnabled(selected);
        } else if (mi == cbmi[2]) {
            rightAction.setEnabled(selected);
        }
    }

    public void displayResult(String actionDescription,
                             ActionEvent e) {
        String s = ("Action event detected: "
                   + actionDescription
                   + newline
                   + "    Event source: " + e.getSource()
                   + newline);
        textArea.append(s);
    }

```

```

}

public class LeftAction extends AbstractAction {
    public LeftAction(String text, ImageIcon icon,
                      String desc, Integer mnemonic) {
        super(text, icon);
        putValue(SHORT_DESCRIPTION, desc);
        putValue(MNEMONIC_KEY, mnemonic);
    }
    public void actionPerformed(ActionEvent e) {
        displayResult("Action for first button/menu item", e);
    }
}

public class MiddleAction extends AbstractAction {
    public MiddleAction(String text, ImageIcon icon,
                       String desc, Integer mnemonic) {
        super(text, icon);
        putValue(SHORT_DESCRIPTION, desc);
        putValue(MNEMONIC_KEY, mnemonic);
    }
    public void actionPerformed(ActionEvent e) {
        displayResult("Action for second button/menu item", e);
    }
}

public class RightAction extends AbstractAction {
    public RightAction(String text, ImageIcon icon,
                      String desc, Integer mnemonic) {
        super(text, icon);
        putValue(SHORT_DESCRIPTION, desc);
        putValue(MNEMONIC_KEY, mnemonic);
    }
    public void actionPerformed(ActionEvent e) {
        displayResult("Action for third button/menu item", e);
    }
}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Make sure we have nice window decorations.
    JFrame.setDefaultLookAndFeelDecorated(true);

    //Create and set up the window.
    JFrame frame = new JFrame("ActionDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Create/set menu bar and content pane.
    ActionDemo demo = new ActionDemo();
    frame.setJMenuBar(demo.createMenuBar());
    demo.createToolBar();
    demo.setOpaque(true); //content panes must be opaque
    frame.setContentPane(demo);

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}

```

```

public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```

Agilidade e Equilibrio:

```

package testes;

/**
 *
 * @author fabio
 */
public abstract class AgilidadeEquilibrio {

    /**
     * Retorna o score do teste de AgilidadeEquilibrio, de acordo
     * com a idade da pessoa, o sexo e o resultado no teste.
     */
    public static int getAgilidadeEquilibrio(double resultado, int idade, char
sexo) {
        int retorno = 0;

        if(sexo == 'M') {
            retorno = getAgilidadeEquilibrioMasculino(resultado, idade);
        } else {
            retorno = getAgilidadeEquilibrioFeminino(resultado, idade);
        }

        return retorno;
    }

    private static int getAgilidadeEquilibrioMasculino(double resultado, int
idade) {
        //TODO: tabela masculina ainda nao foi gerada
        return 0;
    }

    private static int getAgilidadeEquilibrioFeminino(double resultado, int
idade) {

        if(idade >= 50 && idade < 60) {

            if(resultado > 30.7) {
                return 0;
            }

            if(resultado <= 30.7 && resultado > 28.4) {
                return 1;
            }

            if(resultado <= 28.4 && resultado > 28) {
                return 2;
            }

            if(resultado <= 28 && resultado > 27.8) {

```



```

    return 4;
}

if(resultado <= 27.8 && resultado > 27.5) {
    return 5;
}

if(resultado <= 27.5 && resultado > 27.2) {
    return 6;
}

if(resultado <= 27.2 && resultado > 26.9) {
    return 7;
}

if(resultado <= 26.9 && resultado > 26.4) {
    return 8;
}

if(resultado <= 26.4 && resultado > 25.9) {
    return 9;
}

if(resultado <= 25.9 && resultado > 25.6) {
    return 10;
}

if(resultado <= 25.6 && resultado > 25.1) {
    return 11;
}

if(resultado <= 25.1 && resultado > 24.8) {
    return 12;
}

if(resultado <= 24.8 && resultado > 24.7) {
    return 13;
}

if(resultado <= 24.7 && resultado > 24.5) {
    return 14;
}

if(resultado <= 24.5 && resultado > 24.4) {
    return 15;
}

if(resultado <= 24.4 && resultado > 24.2) {
    return 16;
}

if(resultado <= 24.2 && resultado > 24.1) {
    return 19;
}

if(resultado <= 24.1 && resultado > 23.7) {
    return 21;
}

if(resultado <= 23.7 && resultado > 23.6) {
    return 22;
}

```

```
if(resultado <= 23.6 && resultado > 23.4) {
    return 23;
}

if(resultado <= 23.4 && resultado > 23.3) {
    return 24;
}

if(resultado <= 23.3 && resultado > 23.2) {
    return 25;
}

if(resultado <= 23.2 && resultado > 23.1) {
    return 26;
}

if(resultado <= 23.1 && resultado > 22.9) {
    return 27;
}

if(resultado <= 22.9 && resultado > 22.8) {
    return 28;
}

if(resultado <= 22.8 && resultado > 22.7) {
    return 30;
}

if(resultado <= 22.7 && resultado > 22.6) {
    return 31;
}

if(resultado <= 22.6 && resultado > 22.5) {
    return 32;
}

if(resultado <= 22.5 && resultado > 22.4) {
    return 34;
}

if(resultado <= 22.4 && resultado > 22.3) {
    return 36;
}

if(resultado <= 22.3 && resultado > 22.2) {
    return 37;
}

if(resultado <= 22.2 && resultado > 21.9) {
    return 40;
}

if(resultado <= 21.9 && resultado > 21.8) {
    return 43;
}

if(resultado <= 21.8 && resultado > 21.7) {
    return 44;
}

if(resultado <= 21.7 && resultado > 21.6) {
    return 48;
}
```

```
    if(resultado <= 21.6 && resultado > 21.5) {
        return 51;
    }

    if(resultado <= 21.5 && resultado > 21.3) {
        return 53;
    }

    if(resultado <= 21.3 && resultado > 21.2) {
        return 54;
    }

    if(resultado <= 21.2 && resultado > 21.1) {
        return 55;
    }

    if(resultado <= 21.1 && resultado > 21) {
        return 57;
    }

    if(resultado <= 21 && resultado > 20.9) {
        return 60;
    }

    if(resultado <= 20.9 && resultado > 20.8) {
        return 61;
    }

    if(resultado <= 20.8 && resultado > 20.7) {
        return 62;
    }

    if(resultado <= 20.7 && resultado > 20.6) {
        return 63;
    }

    if(resultado <= 20.6 && resultado > 20.5) {
        return 64;
    }

    if(resultado <= 20.5 && resultado > 20.4) {
        return 66;
    }

    if(resultado <= 20.4 && resultado > 20.2) {
        return 66;
    }

    if(resultado <= 20.2 && resultado > 19.9) {
        return 69;
    }

    if(resultado <= 19.9 && resultado > 19.6) {
        return 70;
    }

    if(resultado <= 19.6 && resultado > 19.5) {
        return 71;
    }

    if(resultado <= 19.5 && resultado > 19.4) {
        return 72;
    }
```

```
}

if(resultado <= 19.4 && resultado > 19.3) {
    return 73;
}

if(resultado <= 19.3 && resultado > 19.2) {
    return 74;
}

if(resultado <= 19.2 && resultado > 19.1) {
    return 75;
}

if(resultado <= 19.1 && resultado > 19) {
    return 76;
}

if(resultado <= 19 && resultado > 18.9) {
    return 79;
}

if(resultado <= 18.9 && resultado > 18.8) {
    return 80;
}

if(resultado <= 18.8 && resultado > 18.7) {
    return 81;
}

if(resultado <= 18.7 && resultado > 18.6) {
    return 82;
}

if(resultado <= 18.6 && resultado > 18.5) {
    return 83;
}

if(resultado <= 18.5 && resultado > 18.3) {
    return 84;
}

if(resultado <= 18.3 && resultado > 18.2) {
    return 85;
}

if(resultado <= 18.2 && resultado > 18) {
    return 88;
}

if(resultado <= 18 && resultado > 17.8) {
    return 89;
}

if(resultado <= 17.8 && resultado > 17.6) {
    return 90;
}

if(resultado <= 17.6 && resultado > 17.3) {
    return 91;
}

if(resultado <= 17.3 && resultado > 17) {
```

```

        return 92;
    }

    if(resultado <= 17 && resultado > 16.9) {
        return 93;
    }

    if(resultado <= 16.9 && resultado > 16.6) {
        return 94;
    }

    if(resultado <= 16.6 && resultado > 16.5) {
        return 95;
    }

    if(resultado <= 16.5 && resultado > 16.3) {
        return 96;
    }

    if(resultado <= 16.3 && resultado > 16.1) {
        return 97;
    }

    if(resultado <= 16.1 && resultado > 15.6) {
        return 98;
    }

    if(resultado == 15.6 ) {
        return 99;
    }

    if(resultado < 15.6) {
        return 100;
    }
} else if(idade > 60) {
    if(resultado > 44.4) {
        return 0;
    }

    if(resultado <= 44.4 && resultado > 38.2) {
        return 1;
    }

    if(resultado <= 38.2 && resultado > 32.8) {
        return 2;
    }

    if(resultado <= 32.8 && resultado > 31.8) {
        return 3;
    }

    if(resultado <= 31.8 && resultado > 31.3) {
        return 4;
    }

    if(resultado <= 31.3 && resultado > 30.9) {
        return 5;
    }

    if(resultado <= 30.9 && resultado > 30.2) {
        return 6;
    }
}

```

```
}

if(resultado <= 30.2 && resultado > 29.4) {
    return 7;
}

if(resultado <= 29.4 && resultado > 28.7) {
    return 8;
}

if(resultado <= 28.7 && resultado > 24.4) {
    return 9;
}

if(resultado <= 28.4 && resultado > 28.2) {
    return 11;
}

if(resultado <= 28.2 && resultado > 27.7) {
    return 12;
}

if(resultado <= 27.7 && resultado > 27) {
    return 13;
}

if(resultado <= 27 && resultado > 26.9) {
    return 14;
}

if(resultado <= 26.9 && resultado > 26.8) {
    return 15;
}

if(resultado <= 26.8 && resultado > 26.7) {
    return 16;
}

if(resultado <= 26.7 && resultado > 26.5) {
    return 17;
}

if(resultado <= 26.5 && resultado > 26.4) {
    return 18;
}

if(resultado <= 26.4 && resultado > 26.1) {
    return 20;
}

if(resultado <= 26.1 && resultado > 26) {
    return 21;
}

if(resultado <= 26 && resultado > 25.8) {
    return 22;
}

if(resultado <= 25.8 && resultado > 25.3) {
    return 23;
}

if(resultado <= 25.3 && resultado > 25.2) {
```

```

    return 24;
}

if(resultado <= 25.2 && resultado > 25.1) {
    return 26;
}

if(resultado <= 25.1 && resultado > 25) {
    return 27;
}

if(resultado <= 25 && resultado > 24.9) {
    return 29;
}

if(resultado <= 24.9 && resultado > 24.8) {
    return 30;
}

if(resultado <= 24.8 && resultado > 24.7) {
    return 31;
}

if(resultado <= 24.7 && resultado > 24.6) {
    return 32;
}

if(resultado <= 24.6 && resultado > 24.5) {
    return 33;
}

if(resultado <= 24.5 && resultado > 24.4) {
    return 34;
}

if(resultado <= 24.4 && resultado > 23.9) {
    return 35;
}

if(resultado <= 23.9 && resultado > 23.8) {
    return 37;
}

if(resultado <= 23.8 && resultado > 23.6) {
    return 38;
}

if(resultado <= 23.6 && resultado > 23.4) {
    return 40;
}

if(resultado <= 23.4 && resultado > 23.1) {
    return 41;
}

if(resultado <= 23.1 && resultado > 23) {
    return 42;
}

if(resultado <= 23 && resultado > 22.8) {
    return 43;
}

```

```
if(resultado <= 22.8 && resultado > 22.2) {
    return 44;
}

if(resultado <= 22.2 && resultado > 22.1) {
    return 45;
}

if(resultado <= 22.1 && resultado > 22) {
    return 46;
}

if(resultado <= 22 && resultado > 21.9) {
    return 48;
}

if(resultado <= 21.9 && resultado > 21.8) {
    return 53;
}

if(resultado <= 21.8 && resultado > 21.7) {
    return 55;
}

if(resultado <= 21.7 && resultado > 21.6) {
    return 56;
}

if(resultado <= 21.6 && resultado > 21.5) {
    return 57;
}

if(resultado <= 21.5 && resultado > 21.4) {
    return 58;
}

if(resultado <= 21.4 && resultado > 21.3) {
    return 61;
}

if(resultado <= 21.3 && resultado > 21.2) {
    return 62;
}

if(resultado <= 21.2 && resultado > 21.1) {
    return 63;
}

if(resultado <= 21.1 && resultado > 21) {
    return 64;
}

if(resultado <= 21 && resultado > 20.8) {
    return 65;
}

if(resultado <= 20.8 && resultado > 20.7) {
    return 66;
}

if(resultado <= 20.7 && resultado > 20.6) {
    return 67;
}
```



```
    if(resultado <= 20.6 && resultado > 20.5) {
        return 70;
    }

    if(resultado <= 20.5 && resultado > 20.4) {
        return 71;
    }

    if(resultado <= 20.4 && resultado > 20.2) {
        return 72;
    }

    if(resultado <= 20.2 && resultado > 20) {
        return 74;
    }

    if(resultado <= 20 && resultado > 19.8) {
        return 75;
    }

    if(resultado <= 19.8 && resultado > 19.7) {
        return 76;
    }

    if(resultado <= 19.7 && resultado > 19.6) {
        return 78;
    }

    if(resultado <= 19.6 && resultado > 19.5) {
        return 79;
    }

    if(resultado <= 19.5 && resultado > 19.3) {
        return 82;
    }

    if(resultado <= 19.3 && resultado > 19.1) {
        return 83;
    }

    if(resultado <= 19.1 && resultado > 18.9) {
        return 85;
    }

    if(resultado <= 18.9 && resultado > 18.7) {
        return 86;
    }

    if(resultado <= 18.7 && resultado > 18.4) {
        return 87;
    }

    if(resultado <= 18.4 && resultado > 18.2) {
        return 88;
    }

    if(resultado <= 18.2 && resultado > 18) {
        return 89;
    }

    if(resultado <= 18 && resultado > 17.9) {
        return 90;
    }
}
```

```

    }

    if(resultado <= 17.9 && resultado > 17.4) {
        return 93;
    }

    if(resultado <= 17.44 && resultado > 16.5) {
        return 94;
    }

    if(resultado <= 16.5 && resultado > 15.9) {
        return 95;
    }

    if(resultado <= 15.9 && resultado > 15.2) {
        return 96;
    }

    if(resultado <= 15.2 && resultado > 14.2) {
        return 97;
    }

    if(resultado <= 14.2 && resultado > 10.3) {
        return 98;
    }

    if(resultado == 10.3 ) {
        return 99;
    }

    if(resultado < 10.3) {
        return 100;
    }

}

return 0;

}
}

```

Cadastro de Pessoa

```

package tcc.InterfaceGrafica;

/*
 * CadastroPessoa.java
 *
 * Created on 18 de Abril de 2005, 19:50
 */
import beans.Pessoa;
import beans.Turma;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Date;

```

```

import java.sql.Statement;
import java.text.ParseException;
import java.util.Iterator;
import javax.swing.JFormattedTextField;
import javax.swing.JOptionPane;
import java.util.List;
import javax.swing.text.MaskFormatter;
import tcc.bd.ConexaoBD;
/**
 *
 * @author Fábio
 */
public class CadastroPessoa extends javax.swing.JPanel {

    /** Creates new form CadastroPessoa */
    public CadastroPessoa() {
        initComponents();

        carregaDados();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN: initComponents
        buttonGroupSexo = new javax.swing.ButtonGroup();
        buttonGroupDoenca = new javax.swing.ButtonGroup();
        buttonGroupMedicamentos = new javax.swing.ButtonGroup();
        buttonGroupEstadoSaude = new javax.swing.ButtonGroup();
        jLabelNome = new javax.swing.JLabel();
        jTextFieldNome = new javax.swing.JTextField();
        jLabelNascimento = new javax.swing.JLabel();
        jLabelSexo = new javax.swing.JLabel();
        jRadioButtonMasculino = new javax.swing.JRadioButton();
        jRadioButtonFeminino = new javax.swing.JRadioButton();
        jLabelEndereco = new javax.swing.JLabel();
        jTextFieldEndereco = new javax.swing.JTextField();
        jLabelCep = new javax.swing.JLabel();
        jTextFieldCep = new javax.swing.JTextField();
        jLabelTelefone = new javax.swing.JLabel();
        jTextFieldTelefone = new javax.swing.JTextField();
        jLabelTurma = new javax.swing.JLabel();
        jComboBoxTurma = new javax.swing.JComboBox();
        jButtonCadastrar = new javax.swing.JButton();
        jButtonCancelar = new javax.swing.JButton();
        jLabelCelular = new javax.swing.JLabel();
        jTextFieldCelular = new javax.swing.JTextField();
        jLabelEstadoCivil = new javax.swing.JLabel();
        jComboBoxEstadoCivil = new javax.swing.JComboBox();
        jLabelEscolaridade = new javax.swing.JLabel();
        jComboBoxEscolaridade = new javax.swing.JComboBox();
        jButtonMaisDados = new javax.swing.JButton();
        jFormattedTextFieldNascimento = new javax.swing.JFormattedTextField();

        setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabelNome.setText("Nome:");
        add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(104, 2,
-1, -1));

        jTextFieldNome.setColumns(30);

```

```

        add(jTextFieldNome,
org.netbeans.lib.awtextra.AbsoluteConstraints(171, 2, 346, -1));

        jLabelNascimento.setText("Data de Nascimento:");
        add(jLabelNascimento,
org.netbeans.lib.awtextra.AbsoluteConstraints(341, 26, -1, -1));

        jLabelSexo.setText("Sexo:");
        add(jLabelSexo, new org.netbeans.lib.awtextra.AbsoluteConstraints(104,
26, -1, -1));

        buttonGroupSexo.add(jRadioButtonMasculino);
        jRadioButtonMasculino.setSelected(true);
        jRadioButtonMasculino.setText("Masculino");
        add(jRadioButtonMasculino,
org.netbeans.lib.awtextra.AbsoluteConstraints(171, 26, -1, -1));

        buttonGroupSexo.add(jRadioButtonFeminino);
        jRadioButtonFeminino.setText("Feminino");
        add(jRadioButtonFeminino,
org.netbeans.lib.awtextra.AbsoluteConstraints(266, 26, -1, -1));

        jLabelEndereco.setText("Endere\u00e7o:");
        add(jLabelEndereco,
org.netbeans.lib.awtextra.AbsoluteConstraints(104, 53, -1, -1));

        jTextFieldEndereco.setColumns(40);
        add(jTextFieldEndereco,
org.netbeans.lib.awtextra.AbsoluteConstraints(171, 53, 346, -1));

        jLabelCep.setText("Cep:");
        add(jLabelCep, new org.netbeans.lib.awtextra.AbsoluteConstraints(267, 77,
-1, -1));

        jTextFieldCep.setColumns(9);
        add(jTextFieldCep, new org.netbeans.lib.awtextra.AbsoluteConstraints(342,
77, -1, -1));

        jLabelTelefone.setText("Telefone:");
        add(jLabelTelefone,
org.netbeans.lib.awtextra.AbsoluteConstraints(105, 77, -1, -1));

        jTextFieldTelefone.setColumns(10);
        add(jTextFieldTelefone,
org.netbeans.lib.awtextra.AbsoluteConstraints(172, 77, -1, -1));

        jLabelTurma.setText("Turma:");
        add(jLabelTurma, new org.netbeans.lib.awtextra.AbsoluteConstraints(104,
171, -1, -1));

        add(jComboBoxTurma,
org.netbeans.lib.awtextra.AbsoluteConstraints(171, 171, 168, -1));

        jButtonCadastrar.setText("Cadastrar");
        jButtonCadastrar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonCadastrarActionPerformed(evt);
            }
        });

        add(jButtonCadastrar,
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 270, -1, -1));

```

```

        jButtonCancelar.setText("Cancelar");
        jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonCancelarActionPerformed(evt);
            }
        });

        add(jButtonCancelar, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 270, -1, -1));

        jLabelCelular.setText("Celular:");
        add(jLabelCelular, new org.netbeans.lib.awtextra.AbsoluteConstraints(104,
101, -1, -1));

        jTextFieldCelular.setColumns(10);
        add(jTextFieldCelular, new org.netbeans.lib.awtextra.AbsoluteConstraints(171, 101, -1, -1));

        jLabelEstadoCivil.setText("EstadoCivil:");
        add(jLabelEstadoCivil, new org.netbeans.lib.awtextra.AbsoluteConstraints(104, 125, -1, -1));

        add(jComboBoxEstadoCivil, new org.netbeans.lib.awtextra.AbsoluteConstraints(171, 125, 168, -1));

        jLabelEscolaridade.setText("Escolaridade:");
        add(jLabelEscolaridade, new org.netbeans.lib.awtextra.AbsoluteConstraints(104, 148, -1, -1));

        add(jComboBoxEscolaridade, new org.netbeans.lib.awtextra.AbsoluteConstraints(171, 148, 168, -1));

        jButtonMaisDados.setText("Mais Dados");
        add(jButtonMaisDados, new org.netbeans.lib.awtextra.AbsoluteConstraints(330, 270, -1, -1));

        add(jFormattedTextFieldNascimento, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 30, 80, -1));

    } //GEN-END: initComponents

    private void jButtonCadastrarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonCadastrarActionPerformed

        if(ehValido()) {
            int result = 0;

            Pessoa pessoa = new Pessoa();

            pessoa.nome = jTextFieldNome.getText();

            pessoa.sexo = jRadioButtonMasculino.isSelected() ? 'M' : 'F';

            pessoa.dataNascimento = Date.valueOf(getFormatoData(jTextFieldNascimento.getText()));

            pessoa.endereco = jTextFieldEndereco.getText();

            pessoa.telefone = jTextFieldTelefone.getText();

            pessoa.cep = jTextFieldCep.getText();

            pessoa.celular = jTextFieldCelular.getText();

```

```

        pessoa.estadoCivil = jComboBoxEstadoCivil.getSelectedIndex();
        pessoa.escolaridade = jComboBoxEscolaridade.getSelectedIndex();
        pessoa.idTurma = ((Turma)jComboBoxTurma.getSelectedItem()).idTurma;

        if(pessoa.nome != null && pessoa.nome.trim().length() >0) {

            result = Pessoa.insererPessoa(pessoa);

            if(result > 0) {
                JOptionPane.showMessageDialog(this, "Pessoa incluída com
sucesso.", "Inclusão de Pessoa", JOptionPane.INFORMATION_MESSAGE);

                clear();
            } else {
                JOptionPane.showMessageDialog(this, "Erro ao incluir a
pessoa", "Erro", JOptionPane.ERROR_MESSAGE);
            }

        }
    }
} //GEN-LAST:event_jButtonCadastrarActionPerformed

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButtonCancelarActionPerformed
    this.getParent().remove(this);
} //GEN-LAST:event_jButtonCancelarActionPerformed

private void carregaDados() {
    try {
        //instala a mascara para a data
        MaskFormatter mask = new MaskFormatter( "##/##/####" );

        mask.setValidCharacters( "1234567890" );
        mask.install( (JFormattedTextField) jFormattedTextFieldNascimento);
    } catch(ParseException e) {
        e.printStackTrace();
    }

    String[] estadosCivis = {"Solteiro(a)", "Casado(a)", "Viúvo(a)"};
    for(int i=0; i<estadosCivis.length; i++) {
        jComboBoxEstadoCivil.addItem(estadosCivis[i]);
    }

    String[] escolaridade = {"Ensino Fundamental", "Ensino Médio", "Ensino
Superior", "Pós Graduação"};
    for(int i=0; i<escolaridade.length; i++) {
        jComboBoxEscolaridade.addItem(escolaridade[i]);
    }

    List lista = Turma.getTurmas();
    Iterator iter = lista.iterator();
    while(iter.hasNext()) {
        jComboBoxTurma.addItem((Turma)iter.next());
    }
}

private void clear() {
    jTextFieldNome.setText("");
    jRadioButtonMasculino.setSelected(true);
    jFormattedTextFieldNascimento.setText("");
}

```

```

        jTextFieldEndereco.setText("");
        jTextFieldTelefone.setText("");
        jTextFieldCep.setText("");
        jTextFieldCelular.setText("");
        jComboBoxEstadoCivil.setSelectedIndex(-1);
        jComboBoxEscolaridade.setSelectedIndex(-1);
        jComboBoxTurma.setSelectedIndex(-1);
    }

    private boolean ehValido() {
        boolean valido = true;
        String msg = "Os seguintes problemas foram encontrados: \n";

        if(jTextFieldNome.getText() == null ||
jTextFieldNome.getText().trim().length()==0) {
            valido = false;
            msg = msg+"O nome não está preenchido.\n";
        }

        if(jFormattedTextFieldNascimento.getText() == null ||
jFormattedTextFieldNascimento.getText().trim().length()==0) {
            valido = false;
            msg = msg+"A data de nascimento não está preenchida.\n";
        } else if(!dataValida(jFormattedTextFieldNascimento.getText())) {
            valido = false;
            msg = msg+"A data de nascimento é inválida.\n";
        }

        if(!valido) {
            JOptionPane.showMessageDialog(this, msg, "Validação",
JOptionPane.INFORMATION_MESSAGE);
        }

        return valido;
    }

    private boolean dataValida(String data) {
        boolean valido = true;

        int dia = 0;
        int mes = 0;
        int ano = 0;

        try {
            dia = Integer.parseInt(data.substring(0, 2));
            mes = Integer.parseInt(data.substring(3, 5));
            ano = Integer.parseInt(data.substring(6, 10));
        } catch(NumberFormatException e) {

            return false;
        }

        if(dia<=0 || dia>31) {
            valido = false;
        }
        if(mes<=0 || mes>12) {
            valido = false;
        }
        if(ano<=0) {
            valido = false;
        }

        return valido;
    }

```

```

    }

    /**
     * Retorna um string de data no formato AAAA-MM-DD
     */
    private String getFormatoData(String data) {
        int dia = 0;
        int mes = 0;
        int ano = 0;

        try {
            dia = Integer.parseInt(data.substring(0, 2));
            mes = Integer.parseInt(data.substring(3, 5));
            ano = Integer.parseInt(data.substring(6, 10));
        } catch(NumberFormatException e) {

            e.printStackTrace();
            return "";
        }

        return ano+"-"+mes+"-"+dia;
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.ButtonGroup buttonGroupDoenca;
    private javax.swing.ButtonGroup buttonGroupEstadoSaude;
    private javax.swing.ButtonGroup buttonGroupMedicamentos;
    private javax.swing.ButtonGroup buttonGroupSexo;
    private javax.swing.JButton jButtonCadastrar;
    private javax.swing.JButton jButtonCancelar;
    private javax.swing.JButton jButtonMaisDados;
    private javax.swing.JComboBox jComboBoxEscolaridade;
    private javax.swing.JComboBox jComboBoxEstadoCivil;
    private javax.swing.JComboBox jComboBoxTurma;
    private javax.swing.JFormattedTextField jFormattedTextFieldNascimento;
    private javax.swing.JLabel jLabelCelular;
    private javax.swing.JLabel jLabelCep;
    private javax.swing.JLabel jLabelEndereco;
    private javax.swing.JLabel jLabelEscolaridade;
    private javax.swing.JLabel jLabelEstadoCivil;
    private javax.swing.JLabel jLabelNascimento;
    private javax.swing.JLabel jLabelNome;
    private javax.swing.JLabel jLabelSexo;
    private javax.swing.JLabel jLabelTelefone;
    private javax.swing.JLabel jLabelTurma;
    private javax.swing.JRadioButton jRadioButtonFeminino;
    private javax.swing.JRadioButton jRadioButtonMasculino;
    private javax.swing.JTextField jTextFieldCelular;
    private javax.swing.JTextField jTextFieldCep;
    private javax.swing.JTextField jTextFieldEndereco;
    private javax.swing.JTextField jTextFieldNome;
    private javax.swing.JTextField jTextFieldTelefone;
    // End of variables declaration//GEN-END:variables
}

```

Cadastro de Pessoa Busca:

```
/*
```



```

* CadastroPessoaBusca.java
*
* Created on 18 de Abril de 2005, 00:48
*/

package tcc.InterfaceGrafica;

import java.awt.CardLayout;
import java.awt.Container;
/**
 *
 * @author Fábio
 */
public class CadastroPessoaBusca extends javax.swing.JPanel {

    /** Creates new form CadastroPessoaBusca */
    public CadastroPessoaBusca() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN:initComponents
        java.awt.GridBagConstraints gridBagConstraints;

        jLabelNome = new javax.swing.JLabel();
        jTextFieldNome = new javax.swing.JTextField();
        jButtonProcurar = new javax.swing.JButton();
        jButtonAvancar = new javax.swing.JButton();
        jButtonCancelar = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();

        setLayout(new java.awt.GridBagLayout());

        jLabelNome.setText("Nome:");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.fill = java.awt.GridBagConstraints.VERTICAL;
        gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
        gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
        add(jLabelNome, gridBagConstraints);

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.fill = java.awt.GridBagConstraints.VERTICAL;
        gridBagConstraints.ipadx = 200;
        gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
        gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
        add(jTextFieldNome, gridBagConstraints);

        jButtonProcurar.setText("Busca");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.fill = java.awt.GridBagConstraints.VERTICAL;
        gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);

```

```

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jButtonProcurar, gridBagConstraints);

jButtonAvancar.setText("Avan\u00e7ar");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 2;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jButtonAvancar, gridBagConstraints);

jButtonCancelar.setText("Cancelar");
jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCancelarActionPerformed(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 2;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jButtonCancelar, gridBagConstraints);

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null},
        {null},
        {null},
        {null},
        {null},
        {null},
        {null},
        {null},
        {null},
        {null}
    },
    new String [] {
        "Nome"
    }
));
jScrollPane1.setViewportViewView(jTable1);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
gridBagConstraints.gridwidth = 3;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.weighty = 0.1;
add(jScrollPane1, gridBagConstraints);

} //GEN-END: initComponents

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonCancelarActionPerformed
    this.getParent().remove(this);
} //GEN-LAST:event_jButtonCancelarActionPerformed

// Variables declaration - do not modify //GEN-BEGIN:variables
private javax.swing.JButton jButtonAvancar;
private javax.swing.JButton jButtonCancelar;

```

```

private javax.swing.JButton jButtonProcurar;
private javax.swing.JLabel jLabelNome;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextFieldNome;
// End of variables declaration//GEN-END:variables

}

```

Cadastro de Turma:

```

/*
 * CadastroTurma.java
 *
 * Created on 17 de Abril de 2005, 18:49
 */

package tcc.InterfaceGrafica;

import beans.Turma;
import java.sql.*;
import javax.swing.JOptionPane;
import tcc.bd.ConexaoBD;

/**
 *
 * @author fabio
 */
public class CadastroTurma extends javax.swing.JPanel {

    /** Creates new form CadastroTurma */
    public CadastroTurma() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN:initComponents
        java.awt.GridBagConstraints gridBagConstraints;

        buttonGroup1 = new javax.swing.ButtonGroup();
        jPanelDados = new javax.swing.JPanel();
        jTextFieldHora = new javax.swing.JTextField();
        jLabelNome = new javax.swing.JLabel();
        jTextFieldNome = new javax.swing.JTextField();
        jLabelUniversidade = new javax.swing.JLabel();
        jButtonUfsc = new javax.swing.JRadioButton();
        jButtonUdesc = new javax.swing.JRadioButton();
        jButtonUnesp = new javax.swing.JRadioButton();
        jButtonOutros = new javax.swing.JRadioButton();
        jLabelHorario = new javax.swing.JLabel();
        jLabelDia = new javax.swing.JLabel();
        jLabelHora = new javax.swing.JLabel();
        jTextFieldDia = new javax.swing.JTextField();
        jPanelBotoes = new javax.swing.JPanel();
        jButtonCadastrar = new javax.swing.JButton();
        jButtonCancelar = new javax.swing.JButton();
    } //GEN-END:initComponents
}

```

```

setLayout(new java.awt.BorderLayout());

jPanelDados.setLayout(new java.awt.GridBagLayout());

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 3;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(0, 5, 5, 5);
jPanelDados.add(jTextFieldHora, gridBagConstraints);

jLabelNome.setText("Nome:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jLabelNome, gridBagConstraints);

jTextFieldNome.setColumns(30);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridwidth = 4;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jTextFieldNome, gridBagConstraints);

jLabelUniversidade.setText("Universidade:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jLabelUniversidade, gridBagConstraints);

buttonGroup1.add(jRadioButtonUfsc);
jRadioButtonUfsc.setSelected(true);
jRadioButtonUfsc.setText("UFSC");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jRadioButtonUfsc, gridBagConstraints);

buttonGroup1.add(jRadioButtonUdesc);
jRadioButtonUdesc.setText("UDESC");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jRadioButtonUdesc, gridBagConstraints);

buttonGroup1.add(jRadioButtonUnesp);
jRadioButtonUnesp.setText("UNESP");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 3;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jRadioButtonUnesp, gridBagConstraints);

buttonGroup1.add(jRadioButtonOutros);

```

```

jRadioButtonOutros.setText("Outros");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.weightx = 0.1;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jRadioButtonOutros, gridBagConstraints);

jLabelHorario.setText("Hor\u00e1rio:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jLabelHorario, gridBagConstraints);

jLabelDia.setText("Dia:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jLabelDia, gridBagConstraints);

jLabelHora.setText("Hora:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 3;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(0, 5, 5, 5);
jPanelDados.add(jLabelHora, gridBagConstraints);

jTextFieldDia.setColumns(20);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 2;
gridBagConstraints.gridwidth = 3;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(5, 5, 5, 5);
jPanelDados.add(jTextFieldDia, gridBagConstraints);

add(jPanelDados, java.awt.BorderLayout.NORTH);

jButtonCadastrar.setText("Cadastrar");
jButtonCadastrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCadastrarActionPerformed(evt);
    }
});

jPanelBotoes.add(jButtonCadastrar);

jButtonCancelar.setText("Cancelar");
jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCancelarActionPerformed(evt);
    }
});

```

```

        jPanelBotoes.add(jButtonCancelar);

        add(jPanelBotoes, java.awt.BorderLayout.SOUTH);

    }//GEN-END:initComponents

    private void jButtonCadastrarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonCadastrarActionPerformed

        int result = 0;

        Turma turma = new Turma();

        turma.nome = jTextFieldNome.getText();

        if(jRadioButtonOutros.isSelected()) {
            turma.universidade = 0;
        } else if(jRadioButtonUfsc.isSelected()) {
            turma.universidade = 1;
        } else if(jRadioButtonUdesc.isSelected()) {
            turma.universidade = 2;
        } else if(jRadioButtonUnesp.isSelected()) {
            turma.universidade = 3;
        }

        turma.dia = jTextFieldDia.getText();

        turma.hora = jTextFieldHora.getText();

        if(turma.nome != null && turma.nome.trim().length() > 0) {

            result = Turma.insererTurma(turma);

            if(result > 0) {
                JOptionPane.showMessageDialog(this, "Turma incluída com
                sucesso.", "Inclusão de Turma", JOptionPane.INFORMATION_MESSAGE);

                clear();
            } else {
                JOptionPane.showMessageDialog(this, "Erro ao incluir a turma",
                "Erro", JOptionPane.ERROR_MESSAGE);
            }

        } else {

            JOptionPane.showMessageDialog(this, "Informe o nome da turma a ser
            criada.", "Atenção", JOptionPane.WARNING_MESSAGE);
        }

    } //GEN-LAST:event_jButtonCadastrarActionPerformed

    private void clear() {
        jTextFieldDia.setText("");
        jTextFieldHora.setText("");
        jTextFieldNome.setText("");
        jRadioButtonUfsc.setSelected(true);
    }

    private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonCancelarActionPerformed
        this.getParent().remove(this);
    } //GEN-LAST:event_jButtonCancelarActionPerformed

```

```

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JButton jButtonCadastrar;
private javax.swing.JButton jButtonCancelar;
private javax.swing.JLabel jLabelDia;
private javax.swing.JLabel jLabelHora;
private javax.swing.JLabel jLabelHorario;
private javax.swing.JLabel jLabelNome;
private javax.swing.JLabel jLabelUniversidade;
private javax.swing.JPanel jPanelBotoes;
private javax.swing.JPanel jPanelDados;
private javax.swing.JRadioButton jButtonOutros;
private javax.swing.JRadioButton jButtonUdesc;
private javax.swing.JRadioButton jButtonUfsc;
private javax.swing.JRadioButton jButtonUnesp;
private javax.swing.JTextField jTextFieldDia;
private javax.swing.JTextField jTextFieldHora;
private javax.swing.JTextField jTextFieldNome;
// End of variables declaration//GEN-END:variables
}

```

Conexão BD:

```

/*
 * ConexaoBD.java
 *
 * Created on 8 de Maio de 2005, 21:18
 */
package tcc.bd;

import java.sql.*;

/**
 *
 * @author fabio
 */
public class ConexaoBD {

    static Connection conn;
    static String url;
    static String user;
    static String password;

    public static void conecta(String urlJDBC, String userJDBC, String
passwordJDBC) throws ClassNotFoundException {
        url = urlJDBC;
        user = userJDBC;
        password = passwordJDBC;

        Class.forName("com.mysql.jdbc.Driver");
    }

    public static Connection getConnection() throws SQLException {
        if(conn == null) {
            conn = DriverManager.getConnection(url, user, password);
        }

        return conn;
    }
}

```

```
}  
}
```

Coordenação Java:

```
/*  
 * aahperd.java  
 *  
 * Created on 4 de Julho de 2005, 19:06  
 */  
  
package testes;  
  
/**  
 *  
 * @author fabio  
 */  
public abstract class Coordenacao {  
  
    /**  
     * Retorna o score do teste de coordenação, de acordo  
     * com a idade da pessoa, o sexo e o resultado no teste.  
     */  
    public static int getCoordenacao(double resultado, int idade, char sexo) {  
        int retorno = 0;  
  
        if(sexo == 'M') {  
            retorno = getCoordenacaoMasculino(resultado, idade);  
        } else {  
            retorno = getCoordenacaoFeminino(resultado, idade);  
        }  
  
        return retorno;  
    }  
  
    private static int getCoordenacaoMasculino(double resultado, int idade) {  
        //TODO: tabela masculina ainda nao foi gerada  
        return 0;  
    }  
  
    private static int getCoordenacaoFeminino(double resultado, int idade) {  
        if(idade >= 50 && idade < 60) {  
  
            if(resultado > 21.8) {  
                return 0;  
            }  
  
            if(resultado <= 21.8 && resultado > 21) {  
                return 1;  
            }  
  
            if(resultado <= 21 && resultado > 20.5) {  
                return 2;  
            }  
  
            if(resultado <= 20.5 && resultado > 20) {  
                return 3;  
            }  
  
            if(resultado <= 20 && resultado > 19.7) {  
                return 4;  
            }  
        }  
    }  
}
```



```

}

if(resultado <= 19.7 && resultado > 17.7) {
    return 5;
}

if(resultado <= 17.7 && resultado > 17) {
    return 6;
}

if(resultado <= 17 && resultado > 16.4) {
    return 7;
}

if(resultado <= 16.4 && resultado > 16.1) {
    return 8;
}

if(resultado <= 16.1 && resultado > 15.8) {
    return 9;
}

if(resultado <= 15.8 && resultado > 15.5) {
    return 10;
}

if(resultado <= 15.5 && resultado > 15.4) {
    return 11;
}

if(resultado <= 15.4 && resultado > 15.1) {
    return 12;
}

if(resultado <= 15.1 && resultado > 14.7) {
    return 13;
}

if(resultado <= 14.7 && resultado > 14.3) {
    return 14;
}

if(resultado <= 14.3 && resultado > 14.2) {
    return 15;
}

if(resultado <= 14.2 && resultado > 14) {
    return 16;
}

if(resultado <= 14 && resultado > 13.9) {
    return 17;
}

if(resultado <= 13.9 && resultado > 13.8) {
    return 18;
}

if(resultado <= 13.8 && resultado > 13.7) {
    return 19;
}

if(resultado <= 13.7 && resultado > 13.6) {

```

```

    return 20;
}

if(resultado <= 13.6 && resultado > 13.5) {
    return 22;
}

if(resultado <= 13.5 && resultado > 13.4) {
    return 23;
}

if(resultado <= 13.4 && resultado > 13.3) {
    return 25;
}

if(resultado <= 13.3 && resultado > 13.1) {
    return 26;
}

if(resultado <= 13.1 && resultado > 13) {
    return 27;
}

if(resultado <= 13 && resultado > 12.9) {
    return 28;
}

if(resultado <= 12.9 && resultado > 12.8) {
    return 29;
}

if(resultado <= 12.8 && resultado > 12.6) {
    return 30;
}

if(resultado <= 12.6 && resultado > 12.5) {
    return 31;
}

if(resultado <= 12.5 && resultado > 12.3) {
    return 32;
}

if(resultado <= 12.3 && resultado > 12.2) {
    return 33;
}

if(resultado <= 12.2 && resultado > 12.1) {
    return 35;
}

if(resultado <= 12.1 && resultado > 12) {
    return 36;
}

if(resultado <= 12 && resultado > 11.9) {
    return 37;
}

if(resultado <= 11.9 && resultado > 11.8) {
    return 41;
}

```

```
    if(resultado <= 11.8 && resultado > 11.7) {
        return 42;
    }

    if(resultado <= 11.7 && resultado > 11.6) {
        return 43;
    }

    if(resultado <= 11.6 && resultado > 11.5) {
        return 45;
    }

    if(resultado <= 11.5 && resultado > 11.3) {
        return 46;
    }

    if(resultado <= 11.3 && resultado > 11.2) {
        return 47;
    }

    if(resultado <= 11.2 && resultado > 11.1) {
        return 49;
    }

    if(resultado <= 11.1 && resultado > 11) {
        return 51;
    }

    if(resultado <= 11 && resultado > 10.9) {
        return 52;
    }

    if(resultado <= 10.9 && resultado > 10.8) {
        return 54;
    }

    if(resultado <= 10.8 && resultado > 10.7) {
        return 56;
    }

    if(resultado <= 10.7 && resultado > 10.6) {
        return 57;
    }

    if(resultado <= 10.6 && resultado > 10.5) {
        return 58;
    }

    if(resultado <= 10.5 && resultado > 10.4) {
        return 60;
    }

    if(resultado <= 10.4 && resultado > 10.3) {
        return 63;
    }

    if(resultado <= 10.3 && resultado > 10.2) {
        return 66;
    }

    if(resultado <= 10.2 && resultado > 10.1) {
        return 67;
    }
}
```

```
if(resultado <= 10.1 && resultado > 10) {
    return 68;
}

if(resultado <= 10 && resultado > 9.9) {
    return 69;
}

if(resultado <= 9.9 && resultado > 9.8) {
    return 70;
}

if(resultado <= 9.8 && resultado > 9.7) {
    return 71;
}

if(resultado <= 9.7 && resultado > 9.6) {
    return 72;
}

if(resultado <= 9.6 && resultado > 9.5) {
    return 74;
}

if(resultado <= 9.5 && resultado > 9.4) {
    return 75;
}

if(resultado <= 9.4 && resultado > 9.3) {
    return 79;
}

if(resultado <= 9.3 && resultado > 9.2) {
    return 82;
}

if(resultado <= 9.2 && resultado > 9.1) {
    return 83;
}

if(resultado <= 9.1 && resultado > 9) {
    return 87;
}

if(resultado <= 9 && resultado > 8.9) {
    return 88;
}

if(resultado <= 8.9 && resultado > 8.8) {
    return 89;
}

if(resultado <= 8.8 && resultado > 8.7) {
    return 90;
}

if(resultado <= 8.7 && resultado > 8.6) {
    return 91;
}

if(resultado <= 8.6 && resultado > 8.5) {
    return 92;
}
```

```

    }

    if(resultado <= 8.5 && resultado > 8.4) {
        return 93;
    }

    if(resultado <= 8.4 && resultado > 8.3) {
        return 94;
    }

    if(resultado <= 8.3 && resultado > 8.2) {
        return 95;
    }

    if(resultado <= 8.2 && resultado > 8) {
        return 96;
    }

    if(resultado <= 8 && resultado > 7.8) {
        return 97;
    }

    if(resultado <= 7.8 && resultado > 7.4) {
        return 98;
    }

    if(resultado == 7.4 ) {
        return 99;
    }

    if(resultado < 7.4) {
        return 100;
    }
} else if(idade > 60) {
    if(resultado > 25.3) {
        return 0;
    }

    if(resultado <= 25.3 && resultado > 23.4) {
        return 1;
    }

    if(resultado <= 23.4 && resultado > 21.9) {
        return 2;
    }

    if(resultado <= 21.9 && resultado > 19.7) {
        return 3;
    }

    if(resultado <= 19.7 && resultado > 19.1) {
        return 4;
    }

    if(resultado <= 19.1 && resultado > 18.9) {
        return 5;
    }

    if(resultado <= 18.9 && resultado > 18.6) {
        return 7;
    }
}

```

```

if(resultado <= 18.6 && resultado > 17.6) {
    return 8;
}

if(resultado <= 17.6 && resultado > 16.8) {
    return 9;
}

if(resultado <= 16.8 && resultado > 16.4) {
    return 10;
}

if(resultado <= 16.4 && resultado > 15.8) {
    return 11;
}

if(resultado <= 15.8 && resultado > 15.7) {
    return 12;
}

if(resultado <= 15.7 && resultado > 15.5) {
    return 13;
}

if(resultado <= 15.5 && resultado > 15.1) {
    return 14;
}

if(resultado <= 15.1 && resultado > 15) {
    return 15;
}

if(resultado <= 15 && resultado > 14.9) {
    return 16;
}

if(resultado <= 14.9 && resultado > 14.7) {
    return 17;
}

if(resultado <= 14.7 && resultado > 14.5) {
    return 18;
}

if(resultado <= 14.5 && resultado > 14.4) {
    return 20;
}

if(resultado <= 14.4 && resultado > 14.3) {
    return 21;
}

if(resultado <= 14.3 && resultado > 14.1) {
    return 22;
}

if(resultado <= 14.1 && resultado > 14) {
    return 23;
}

if(resultado <= 14 && resultado > 13.8) {
    return 24;
}

```

```

}

if(resultado <= 13.8 && resultado > 13.6) {
    return 25;
}

if(resultado <= 13.6 && resultado > 13.4) {
    return 26;
}

if(resultado <= 13.4 && resultado > 13.3) {
    return 27;
}

if(resultado <= 13.3 && resultado > 13.2) {
    return 28;
}

if(resultado <= 13.2 && resultado > 13.1) {
    return 32;
}

if(resultado <= 13.1 && resultado > 13) {
    return 34;
}

if(resultado <= 13 && resultado > 12.9) {
    return 35;
}

if(resultado <= 12.9 && resultado > 12.8) {
    return 38;
}

if(resultado <= 12.8 && resultado > 12.7) {
    return 39;
}

if(resultado <= 12.7 && resultado > 12.6) {
    return 40;
}

if(resultado <= 12.6 && resultado > 12.5) {
    return 42;
}

if(resultado <= 12.5 && resultado > 12.4) {
    return 43;
}

if(resultado <= 12.4 && resultado > 12.3) {
    return 45;
}

if(resultado <= 12.3 && resultado > 12.1) {
    return 46;
}

if(resultado <= 12.1 && resultado > 12) {
    return 47;
}

if(resultado <= 12 && resultado > 11.9) {

```

```

    return 49;
}

if(resultado <= 11.9 && resultado > 11.8) {
    return 51;
}

if(resultado <= 11.8 && resultado > 11.7) {
    return 54;
}

if(resultado <= 11.7 && resultado > 11.6) {
    return 57;
}

if(resultado <= 11.6 && resultado > 11.5) {
    return 61;
}

if(resultado <= 11.5 && resultado > 11.4) {
    return 61;
}

if(resultado <= 11.4 && resultado > 11.3) {
    return 63;
}

if(resultado <= 11.3 && resultado > 11.1) {
    return 64;
}

if(resultado <= 11.1 && resultado > 11) {
    return 65;
}

if(resultado <= 11 && resultado > 10.9) {
    return 66;
}

if(resultado <= 10.9 && resultado > 10.8) {
    return 67;
}

if(resultado <= 10.8 && resultado > 10.7) {
    return 70;
}

if(resultado <= 10.7 && resultado > 10.6) {
    return 72;
}

if(resultado <= 10.6 && resultado > 10.5) {
    return 73;
}

if(resultado <= 10.5 && resultado > 10.3) {
    return 74;
}

if(resultado <= 10.3 && resultado > 10.2) {
    return 76;
}

```



```
if(resultado <= 10.2 && resultado > 10.1) {
    return 77;
}

if(resultado <= 10.1 && resultado > 9.9) {
    return 79;
}

if(resultado <= 9.9 && resultado > 9.7) {
    return 82;
}

if(resultado <= 9.7 && resultado > 9.5) {
    return 84;
}

if(resultado <= 9.5 && resultado > 9.4) {
    return 85;
}

if(resultado <= 9.4 && resultado > 9.3) {
    return 86;
}

if(resultado <= 9.3 && resultado > 9.2) {
    return 87;
}

if(resultado <= 9.2 && resultado > 9) {
    return 88;
}

if(resultado <= 9 && resultado > 8.9) {
    return 89;
}

if(resultado <= 8.9 && resultado > 8.8) {
    return 90;
}

if(resultado <= 8.8 && resultado > 8.7) {
    return 92;
}

if(resultado <= 8.7 && resultado > 8.6) {
    return 93;
}

if(resultado <= 8.6 && resultado > 8.3) {
    return 94;
}

if(resultado <= 8.3 && resultado > 8.1) {
    return 95;
}

if(resultado <= 8.1 && resultado > 8) {
    return 96;
}

if(resultado <= 8 && resultado > 7.8) {
    return 94;
}
```

```

        if(resultado <= 7.8 && resultado > 7.7) {
            return 98;
        }

        if(resultado == 7.7 ) {
            return 99;
        }

        if(resultado < 7.7) {
            return 100;
        }

    }

    return 0;
}
}

```

Exclusão Pessoa:

```

/*
 * TestePessoaBusca.java
 *
 * Created on 26 de Abril de 2005, 15:53
 */

package tcc.InterfaceGrafica;

import beans.Pessoa;
import java.awt.CardLayout;
import java.awt.Container;
import java.awt.Cursor;
import java.sql.ResultSet;
import java.util.Iterator;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
/**
 *
 * @author Fábio
 */
public class ExclusaoPessoa extends javax.swing.JPanel {

    /** Creates new form TestePessoaBusca */
    public ExclusaoPessoa() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN:initComponents
        jLabelNome = new javax.swing.JLabel();

```

```

jTextFieldNome = new javax.swing.JTextField();
jButtonProcura = new javax.swing.JButton();
jButtonExcluir = new javax.swing.JButton();
jButtonCancelar = new javax.swing.JButton();
jScrollPane = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();

setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabelNome.setText("Nome:");
add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 2, -
1, -1));

jTextFieldNome.setColumns(30);
add(jTextFieldNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
2, 224, 23));

jButtonProcura.setText("Procura");
jButtonProcura.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonProcuraActionPerformed(evt);
    }
});

add(jButtonProcura,
org.netbeans.lib.awtextra.AbsoluteConstraints(254, 2, -1, -1));

jButtonExcluir.setText("Excluir");
jButtonExcluir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonExcluirActionPerformed(evt);
    }
});

add(jButtonExcluir, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
260, -1, -1));

jButtonCancelar.setText("Cancelar");
jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCancelarActionPerformed(evt);
    }
});

add(jButtonCancelar,
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 260, -1, -1));

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        },
    new String [] {
        "Pessoa"
    }
) {
    boolean[] canEdit = new boolean [] {
        false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

```

```

        jScrollPane.setViewportView(jTable1);

        add(jScrollPane, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
27, 320, 220));

    }//GEN-END: initComponents

    private void jButtonExcluirActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonExcluirActionPerformed
        Pessoa p = getPessoaSelecionada();

        if(p != null) {
            Pessoa.excluirPessoa(p);

            //recarrega as pessoas
            jButtonProcuraActionPerformed(null);

            JOptionPane.showMessageDialog(this, "Pessoa excluída com sucesso",
"Exclusão", JOptionPane.INFORMATION_MESSAGE);
        } else {

            JOptionPane.showMessageDialog(this, "Nenhuma pessoa selecionada",
"Atenção", JOptionPane.WARNING_MESSAGE);
        }
    } //GEN-LAST:event_jButtonExcluirActionPerformed

    private void jButtonProcuraActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonProcuraActionPerformed
        this.setCursor(new Cursor(Cursor.WAIT_CURSOR));

        DefaultTableModel model = ((DefaultTableModel)jTable1.getModel());

        //LIMPA A TABELA
        model.setRowCount(0);

        String nome = jTextFieldNome.getText() != null ?
jTextFieldNome.getText(): "";

        List lista = Pessoa.buscaPessoas(nome);

        Iterator iter = lista.iterator();
        while(iter.hasNext()) {
            Pessoa p = (Pessoa)iter.next();

            model.addRow(new Object[]{p});
        }

        this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
    } //GEN-LAST:event_jButtonProcuraActionPerformed

    private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonCancelarActionPerformed
        // Principal.getInstance().cancelar(this);
        this.getParent().remove(this);
    } //GEN-LAST:event_jButtonCancelarActionPerformed

    private Pessoa getPessoaSelecionada() {
        Pessoa pessoaRetorno = null;

        int linhaSelec = jTable1.getSelectedRow();

        if(linhaSelec >= 0) {
            pessoaRetorno = (Pessoa)jTable1.getValueAt(linhaSelec, 0);
        }
    }

```

```

    }

    return pessoaRetorno;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButtonCancelar;
private javax.swing.JButton jButtonExcluir;
private javax.swing.JButton jButtonProcura;
private javax.swing.JLabel jLabelNome;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextFieldNome;
// End of variables declaration//GEN-END:variables
}

```

Flexibilidade:

```

package testes;

/**
 *
 * @author fabio
 */
public abstract class Flexibilidade {

    /**
     * Retorna o score do teste de flexibilidade, de acordo
     * com a idade da pessoa, o sexo e o resultado no teste.
     */
    public static int getFlexibilidade(double resultado, int idade, char sexo) {
        int retorno = 0;

        if(sexo == 'M') {
            retorno = getFlexibilidadeMasculino(resultado, idade);
        } else {
            retorno = getFlexibilidadeFeminino(resultado, idade);
        }

        return retorno;
    }

    private static int getFlexibilidadeMasculino(double resultado, int idade) {
        //TODO: tabela masculina ainda nao foi gerada
        return 0;
    }

    private static int getFlexibilidadeFeminino(double resultado, int idade) {

        if(idade >= 50 && idade < 60) {

            if(resultado < 7.6) {
                return 0;
            }

            if(resultado == 7.6) {
                return 1;
            }

            if(resultado <= 15.3) {

```

```
    return 2;
}

if(resultado <= 18.5 ) {
    return 3;
}

if(resultado <= 20.8) {
    return 4;
}

if(resultado <= 22.8 ) {
    return 5;
}

if(resultado <= 24) {
    return 6;
}

if(resultado <= 24.7) {
    return 7;
}

if(resultado <= 26.2) {
    return 8;
}

if(resultado <= 31.4) {
    return 9;
}

if(resultado <= 36.4) {
    return 10;
}

if(resultado <= 36.8) {
    return 11;
}

if(resultado <= 36.9) {
    return 12;
}

if(resultado <= 38.5) {
    return 13;
}

if(resultado <= 40.7) {
    return 14;
}

if(resultado <= 42) {
    return 16;
}

if(resultado <= 43) {
    return 17;
}

if(resultado <= 43.2) {
    return 18;
}

if(resultado <= 43.5) {
    return 19;
}

if(resultado <= 44.4) {
    return 20;
}

if(resultado <= 45.2) {
    return 21;
}

if(resultado <= 46.6) {
    return 22;
}
```

```
    if(resultado <= 47) {
        return 23;
    }
    if(resultado <= 48.3) {
        return 24;
    }
    if(resultado <= 49) {
        return 26;
    }
    if(resultado <= 49.3) {
        return 27;
    }
    if(resultado <= 49.5) {
        return 29;
    }
    if(resultado <= 50) {
        return 30;
    }
    if(resultado <= 50.2) {
        return 31;
    }
    if(resultado <= 50.8) {
        return 34;
    }
    if(resultado <= 50.9) {
        return 35;
    }
    if(resultado <= 51) {
        return 36;
    }
    if(resultado <= 51.3) {
        return 37;
    }
    if(resultado <= 52) {
        return 38;
    }
    if(resultado <= 52.6) {
        return 39;
    }
    if(resultado <= 53) {
        return 40;
    }
    if(resultado <= 53.3) {
        return 42;
    }
    if(resultado <= 53.5) {
        return 43;
    }
    if(resultado <= 54.3) {
        return 44;
    }
    if(resultado <= 55.4) {
        return 45;
    }
    if(resultado <= 55.8) {
        return 52;
    }
    if(resultado <= 57) {
        return 53;
    }
    if(resultado <= 57.4) {
        return 54;
    }
}
```

```
    if(resultado <= 58.4) {
        return 58;
    }
    if(resultado <= 59) {
        return 59;
    }
    if(resultado <= 59.1) {
        return 60;
    }
    if(resultado <= 59.6) {
        return 61;
    }
    if(resultado <= 59.8) {
        return 62;
    }
    if(resultado <= 60) {
        return 65;
    }
    if(resultado <= 60.7) {
        return 66;
    }
    if(resultado <= 60.9) {
        return 70;
    }
    if(resultado <= 61) {
        return 71;
    }
    if(resultado <= 62.2) {
        return 72;
    }
    if(resultado <= 63.2) {
        return 73;
    }
    if(resultado <= 63.8) {
        return 80;
    }
    if(resultado <= 64.5) {
        return 81;
    }
    if(resultado <= 64.7) {
        return 82;
    }
    if(resultado <= 65) {
        return 83;
    }
    if(resultado <= 65.3) {
        return 84;
    }
    if(resultado <= 66) {
        return 88;
    }
    if(resultado <= 66.1) {
        return 89;
    }
    if(resultado <= 67.8) {
        return 90;
    }
    if(resultado <= 70.7) {
        return 92;
    }
    if(resultado <= 71.1) {
        return 93;
    }
}
```



```

        if(resultado <= 72.3) {
            return 94;
        }
        if(resultado <= 74.9) {
            return 95;
        }
        if(resultado <= 80.4) {
            return 96;
        }
        if(resultado <= 81.2) {
            return 97;
        }
        if(resultado <= 82.1) {
            return 98;
        }
        if(resultado <= 83.7) {
            return 99;
        }
        if(resultado > 83.7){
            return 100;
        }
    }else if (idade > 60){

        if(resultado < 11.5) {
            return 0;
        }

        if(resultado == 11.5) {
            return 1;
        }

        if(resultado <= 12.8) {
            return 2;
        }

        if(resultado <= 13.4) {
            return 3;
        }

        if(resultado <= 14.3) {
            return 4;
        }

        if(resultado <= 16 ) {
            return 5;
        }

        if(resultado <= 16.8) {
            return 6;
        }

        if(resultado <= 17.9) {
            return 7;
        }

        if(resultado <= 19.1) {
            return 8;
        }

        if(resultado <= 19.5) {
            return 9;
        }

        if(resultado <= 19.7) {
            return 10;
        }
    }

```

```
    if(resultado <= 20.4) {
        return 11;
    }
    if(resultado <= 21.2) {
        return 12;
    }
    if(resultado <= 21.8) {
        return 13;
    }
    if(resultado <= 22.6) {
        return 14;
    }
    if(resultado <= 23) {
        return 17;
    }
    if(resultado <= 23.1) {
        return 18;
    }
    if(resultado <= 24) {
        return 19;
    }
    if(resultado <= 24.5) {
        return 20;
    }
    if(resultado <= 24.9) {
        return 21;
    }
    if(resultado <= 25.4) {
        return 22;
    }
    if(resultado <= 25.9) {
        return 23;
    }
    if(resultado <= 26.4) {
        return 24;
    }
    if(resultado <= 26.8) {
        return 25;
    }
    if(resultado <= 27.3) {
        return 26;
    }
    if(resultado <= 29.4) {
        return 27;
    }
    if(resultado <= 34.4) {
        return 28;
    }
    if(resultado <= 37.8) {
        return 29;
    }
    if(resultado <= 38.9) {
        return 30;
    }
    if(resultado <= 39.9) {
        return 31;
    }
    if(resultado <= 40.6) {
        return 32;
    }
    if(resultado <= 41.1) {
        return 33;
    }
}
```

```
    if(resultado <= 42.6) {
        return 34;
    }
    if(resultado <= 44) {
        return 36;
    }
    if(resultado <= 44.5) {
        return 37;
    }
    if(resultado <= 45) {
        return 40;
    }
    if(resultado <= 45.6) {
        return 41;
    }
    if(resultado <= 46.8) {
        return 42;
    }
    if(resultado <= 46.9) {
        return 43;
    }
    if(resultado <= 47.8) {
        return 44;
    }
    if(resultado <= 48.1) {
        return 45;
    }
    if(resultado <= 48.2) {
        return 46;
    }
    if(resultado <= 48.7) {
        return 47;
    }
    if(resultado <= 49.3) {
        return 48;
    }
    if(resultado <= 49.7) {
        return 49;
    }
    if(resultado <= 50) {
        return 50;
    }
    if(resultado <= 50.3) {
        return 51;
    }
    if(resultado <= 51.3) {
        return 52;
    }
    if(resultado <= 52.2) {
        return 53;
    }
    if(resultado <= 52.6) {
        return 54;
    }
    if(resultado <= 53) {
        return 56;
    }
    if(resultado <= 53.3) {
        return 57;
    }
    if(resultado <= 53.4) {
        return 58;
    }
}
```

```
    if(resultado <= 54) {
        return 60;
    }
    if(resultado <= 55.7) {
        return 61;
    }
    if(resultado <= 55.8) {
        return 63;
    }
    if(resultado <= 55.9) {
        return 64;
    }
    if(resultado <= 56) {
        return 60;
    }
    if(resultado <= 56.6) {
        return 67;
    }
    if(resultado <= 57) {
        return 71;
    }
    if(resultado <= 57.8) {
        return 72;
    }
    if(resultado <= 59.3) {
        return 73;
    }
    if(resultado <= 60.2) {
        return 74;
    }
    if(resultado <= 60.9) {
        return 75;
    }
    if(resultado <= 61.1) {
        return 76;
    }
    if(resultado <= 61.5) {
        return 77;
    }
    if(resultado <= 62) {
        return 82;
    }
    if(resultado <= 62.8) {
        return 83;
    }
    if(resultado <= 63) {
        return 84;
    }
    if(resultado <= 63.7) {
        return 85;
    }
    if(resultado <= 64.7) {
        return 86;
    }
    if(resultado <= 65.6) {
        return 87;
    }
    if(resultado <= 66) {
        return 90;
    }
    if(resultado <= 66.9) {
        return 91;
    }
}
```

```

        if(resultado <= 68.4) {
            return 92;
        }
        if(resultado <= 69) {
            return 93;
        }
        if(resultado <= 69.6) {
            return 94;
        }
        if(resultado <= 71.2) {
            return 95;
        }
        if(resultado <= 72.3) {
            return 96;
        }
        if(resultado <= 74.7) {
            return 97;
        }
        if(resultado <= 82) {
            return 98;
        }
        if(resultado <= 82.5) {
            return 99;
        }
        if(resultado > 82.5){
            return 100;
        }
    }
    return 0;
}
}
}

```

Força:

```

package testes;

/**
 *
 * @author fabio
 */
public abstract class Forca {

    /**
     * Retorna o score do teste de Forca, de acordo
     * com a idade da pessoa, o sexo e o resultado no teste.
     */
    public static int getForca(double resultado, int idade, char sexo) {
        int retorno = 0;

        if(sexo == 'M') {
            retorno = getForcaMasculino(resultado, idade);
        } else {
            retorno = getForcaFeminino(resultado, idade);
        }

        return retorno;
    }
}

```

```

private static int getForcaMasculino(double resultado, int idade) {
    //TODO: tabela masculina ainda nao foi gerada
    return 0;
}
private static int getForcaFeminino(double resultado, int idade) {

    if(idade >= 50 && idade < 60) {

        if(resultado < 11) {
            return 0;
        }

        if(resultado == 11) {
            return 2;
        }

        if(resultado <= 12) {
            return 3;
        }

        if(resultado <= 13 ) {
            return 5;
        }

        if(resultado <= 14) {
            return 8;
        }

        if(resultado <= 15 ) {
            return 10;
        }

        if(resultado <= 16) {
            return 12;
        }
        if(resultado <= 17) {
            return 15;
        }
        if(resultado <= 18) {
            return 23;
        }
        if(resultado <= 19) {
            return 26;
        }
        if(resultado <= 20) {
            return 35;
        }
        if(resultado <= 21) {
            return 40;
        }
        if(resultado <= 22) {
            return 44;
        }
        if(resultado <= 23) {
            return 50;
        }
        if(resultado <= 24) {
            return 57;
        }
        if(resultado <= 25) {
            return 62;
        }
    }
}

```

```

        if(resultado <= 26) {
            return 68;
        }
        if(resultado <= 27) {
            return 74;
        }
        if(resultado <= 28) {
            return 78;
        }
        if(resultado <= 29) {
            return 86;
        }
        if(resultado <= 30) {
            return 90;
        }
        if(resultado <= 31) {
            return 93;
        }
        if(resultado <= 32) {
            return 94;
        }
        if(resultado <= 33) {
            return 97;
        }
        if(resultado <= 34) {
            return 98;
        }
        if(resultado <= 35) {
            return 99;
        }
        if(resultado > 35) {
            return 100;
        }
    }

}else if (idade > 60){

    if(resultado < 10) {
        return 0;
    }

    if(resultado == 10) {
        return 1;
    }

    if(resultado <= 11) {
        return 2;
    }

    if(resultado <= 12 ) {
        return 3;
    }

    if(resultado <= 13) {
        return 4;
    }

    if(resultado <= 14 ) {
        return 5;
    }

    if(resultado <= 15) {
        return 11;
    }
}

```

```
}
  if(resultado <= 16) {
    return 14;
  }
  if(resultado <= 17) {
    return 17;
  }
  if(resultado <= 18) {
    return 21;
  }
  if(resultado <= 19) {
    return 28;
  }
  if(resultado <= 20) {
    return 33;
  }
  if(resultado <= 21) {
    return 38;
  }
  if(resultado <= 22) {
    return 48;
  }
  if(resultado <= 23) {
    return 51;
  }
  if(resultado <= 24) {
    return 59;
  }
  if(resultado <= 25) {
    return 64;
  }
  if(resultado <= 26) {
    return 70;
  }
  if(resultado <= 27) {
    return 73;
  }
  if(resultado <= 28) {
    return 76;
  }
  if(resultado <= 29) {
    return 80;
  }
  if(resultado <= 30) {
    return 84;
  }
  if(resultado <= 31) {
    return 86;
  }
  if(resultado <= 32) {
    return 88;
  }
  if(resultado <= 33) {
    return 90;
  }
  if(resultado <= 34) {
    return 93;
  }
  if(resultado <= 35) {
    return 94;
  }
  if(resultado <= 36) {
    return 95;
  }
```



```

    }
    if(resultado <= 37) {
        return 96;
    }
    if(resultado <= 42) {
        return 97;
    }
    if(resultado <= 43) {
        return 99;
    }
    if(resultado > 43) {
        return 100;
    }
}
return 0;
}
}

```

Gerador Relatório:

```

/*
 * GeradorRelatorio.java
 *
 * Created on 4 de Julho de 2005, 18:52
 */

package testes;

import beans.Pessoa;
import beans.Teste;
import beans.Turma;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Calendar;
import java.util.Date;
import javax.swing.JFileChooser;
import utils.Utills;

/**
 *
 * @author fabio
 */
public class GeradorRelatorio {

    private static final String titulo = "==== Relatório de testes
- Bateria AAHPERD =====";

    private static final String fim =
"===== ";

    private File diretorio;

    public GeradorRelatorio(File diretorio) {
        this.diretorio = diretorio;
    }
}

```

```

/**
 * Gera um relatório em um arquivo texto.
 */
public void gerarRelatorio(Teste teste, Pessoa p) {
    if(teste != null && p != null) {

        String nmArquivo = diretorio+"/"+p+"_"+teste+".txt";
        int idade = Utils.getIdade(p);
        int coordenacao = Coordenacao.getCoordenacao(teste.coordenacao,
idade, p.sexo);
        int agilidade =
AgilidadeEquilibrio.getAgilidadeEquilibrio(teste.agilidadeEquilibrio,
idade,
p.sexo);
        int flexibilidade =
Flexibilidade.getFlexibilidade(teste.flexibilidade, idade, p.sexo);
        int forca = Forca.getForca(teste.forcaMS, idade, p.sexo);
        int resistencia =
Resistencia.getResistencia(teste.resistenciaAerobica, idade, p.sexo);
        int iafg = coordenacao+ agilidade +flexibilidade +forca +
resistencia;

        try {
            FileWriter escritor = new FileWriter(nmArquivo);
            escritor.write(titulo+"\n");
            escritor.write("\n");
            escritor.write("Nome: "+p.nome+"\n");
            escritor.write("Idade: "+idade+"\n");
            escritor.write("Sexo: "+Utils.getSexo(p.sexo)+"\n");
            escritor.write("Massa-peso: "+teste.massaPeso+"\n");
            escritor.write("Estatura: "+teste.estatura+"\n");
            escritor.write("Telefone: "+p.telefone+"\n");
            escritor.write("Endereço: "+p.telefone+"\n");
            escritor.write("Turma: "+Turma.getTurma(p.idTurma)+"\n");
            escritor.write("Ano do teste: "+teste.ano+"\n");
            escritor.write("Mês do teste: "+Utils.getMes(teste.mes)+"\n");
            escritor.write("\n");
            escritor.write("Resultado
Escore Percentil Condição\n");
            escritor.write("Coordenação = "+teste.coordenacao+
"+coordenacao+" "+getCondicao(coordenacao)+"\n");
            escritor.write("Agilidade/Equilibrio =
"+teste.agilidadeEquilibrio+" "+agilidade+
"+getCondicao(agilidade)+"\n");
            escritor.write("Flexibilidade = "+teste.flexibilidade+
"+flexibilidade+" "+getCondicao(flexibilidade)+"\n");
            escritor.write("Força = "+teste.forcaMS+
"+forca+" "+getCondicao(forca)+"\n");
            escritor.write("Resistência (804 mt) =
"+teste.resistenciaAerobica+" "+resistencia+
"+getCondicao(resistencia)+"\n");
            escritor.write("\n");
            escritor.write("\n");
            escritor.write("\n");
            escritor.write("Índice de Aptidão Funcional Geral (IAFG) =
"+iafg+" "+getCondicao(iafg/5)+"\n");
            escritor.write("\n");
            escritor.write(fim);

            escritor.close();
        } catch(IOException e) {
            e.printStackTrace();
        }
    }
}

```

```

    }

    private String getCondicao(int escore) {
        String result = "";

        if(escore > 0 && escore <= 19) {
            result = "Muito Fraco";
        } else if(escore > 19 && escore <= 39) {
            result = "Fraco";
        } else if(escore > 39 && escore <= 59) {
            result = "Regular";
        } else if(escore > 59 && escore <= 79) {
            result = "Bom";
        } else if(escore > 79 && escore <= 100) {
            result = "Muito Bom";
        }

        return result;
    }
}

```

Interface Teste:

```

package tcc.InterfaceGrafica;

import beans.Pessoa;
import beans.Teste;
import java.io.File;
import java.text.ParseException;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.text.MaskFormatter;
import testes.GeradorRelatorio;

/*
 * InterfaceTeste.java
 *
 * Created on 19 de Abril de 2005, 11:35
 */

/**
 *
 * @author Fábio
 */
public class InterfaceTeste extends javax.swing.JPanel {

    Pessoa p;
    Teste testeAtual;

    public InterfaceTeste(Pessoa p) {
        initComponents();

        this.p = p;

        carregaDados();
    }
}

```

```

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void initComponents() { //GEN-BEGIN:initComponents
    buttonGroupMes = new javax.swing.ButtonGroup();
    jLabelNome = new javax.swing.JLabel();
    jTextFieldNome = new javax.swing.JTextField();
    jLabelAno = new javax.swing.JLabel();
    jTextFieldAno = new javax.swing.JTextField();
    jLabelPeso = new javax.swing.JLabel();
    jTextFieldPeso = new javax.swing.JTextField();
    jLabelEstatura = new javax.swing.JLabel();
    jTextFieldEstatura = new javax.swing.JTextField();
    jButtonSalvar = new javax.swing.JButton();
    jButtonRelatorio = new javax.swing.JButton();
    jButtonCancelar = new javax.swing.JButton();
    jButtonRadioMar = new javax.swing.JRadioButton();
    jButtonRadioDez = new javax.swing.JRadioButton();
    jLabelBateria = new javax.swing.JLabel();
    jLabelForcaMS = new javax.swing.JLabel();
    jTextFieldForcaMS = new javax.swing.JTextField();
    jLabelFlexibilidade = new javax.swing.JLabel();
    jTextFieldFlexibilidade = new javax.swing.JTextField();
    jLabelCordenacao = new javax.swing.JLabel();
    jTextFieldCordenacao = new javax.swing.JTextField();
    jLabelAgilidadeEquilibrio = new javax.swing.JLabel();
    jTextFieldAgilidadeEquilibrio = new javax.swing.JTextField();
    jLabelResistencia = new javax.swing.JLabel();
    jTextFieldResistencia = new javax.swing.JTextField();
    jButtonMaisTestes = new javax.swing.JButton();

    setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabelNome.setText("Nome:");
    add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(47, 1,
-1, -1));

    jTextFieldNome.setColumns(35);
    jTextFieldNome.setEditable(false);
    add(jTextFieldNome,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 1, -1, -1));

    jLabelAno.setText("Ano:");
    add(jLabelAno, new org.netbeans.lib.awtextra.AbsoluteConstraints(47, 23,
-1, -1));

    jTextFieldAno.setColumns(5);
    add(jTextFieldAno, new org.netbeans.lib.awtextra.AbsoluteConstraints(150,
23, -1, -1));

    jLabelPeso.setText("Massa - peso:");
    add(jLabelPeso, new org.netbeans.lib.awtextra.AbsoluteConstraints(47, 48,
-1, -1));

    jTextFieldPeso.setColumns(5);
    add(jTextFieldPeso,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 48, -1, -1));

    jLabelEstatura.setText("Estatura:");
    add(jLabelEstatura,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(233, 48, -1, -1));

```

```

        jTextFieldEstatura.setColumns(5);
        add(jTextFieldEstatura,
org.netbeans.lib.awtextra.AbsoluteConstraints(328, 48, -1, -1));
new

        jButtonSalvar.setText("Salvar");
        jButtonSalvar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonSalvarActionPerformed(evt);
            }
        });

        add(jButtonSalvar, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
220, -1, -1));

        jButtonRelatorio.setText("Relatorio");
        jButtonRelatorio.setEnabled(false);
        jButtonRelatorio.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonRelatorioActionPerformed(evt);
            }
        });

        add(jButtonRelatorio,
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 220, 95, -1));
new

        jButtonCancelar.setText("Cancelar");
        jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonCancelarActionPerformed(evt);
            }
        });

        add(jButtonCancelar,
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 220, -1, -1));
new

        buttonGroupMes.add(jRadioButtonMar);
        jRadioButtonMar.setSelected(true);
        jRadioButtonMar.setText("Mar");
        add(jRadioButtonMar,
org.netbeans.lib.awtextra.AbsoluteConstraints(233, 23, -1, -1));
new

        buttonGroupMes.add(jRadioButtonDez);
        jRadioButtonDez.setText("Dez");
        add(jRadioButtonDez,
org.netbeans.lib.awtextra.AbsoluteConstraints(328, 23, -1, -1));
new

        jLabelBateria.setFont(new java.awt.Font("MS Sans Serif", 0, 14));
        jLabelBateria.setText("-----Bateria AAHPERD-----");
        add(jLabelBateria, new org.netbeans.lib.awtextra.AbsoluteConstraints(150,
70, -1, -1));

        jLabelForcaMS.setText("Forca MS:");
        add(jLabelForcaMS, new org.netbeans.lib.awtextra.AbsoluteConstraints(47,
91, -1, -1));

        jTextFieldForcaMS.setColumns(5);
        add(jTextFieldForcaMS,
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 91, -1, -1));
new

        jLabelFlexibilidade.setText("Flexibilidade:");
        add(jLabelFlexibilidade,
org.netbeans.lib.awtextra.AbsoluteConstraints(47, 179, -1, -1));
new

```

```

        jTextFieldFlexibilidade.setColumns(5);
        add(jTextFieldFlexibilidade,
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 179, -1, -1));
new

        jLabelCordenacao.setText("Coordena\u00e7\u00e3o:");
        add(jLabelCordenacao,
org.netbeans.lib.awtextra.AbsoluteConstraints(47, 113, -1, -1));
new

        jTextFieldCordenacao.setColumns(5);
        add(jTextFieldCordenacao,
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 113, -1, -1));
new

        jLabelAgilidadeEquilibrio.setText("Agilidade e Equilibrio:");
        add(jLabelAgilidadeEquilibrio,
org.netbeans.lib.awtextra.AbsoluteConstraints(47, 135, -1, -1));
new

        jTextFieldAgilidadeEquilibrio.setColumns(5);
        add(jTextFieldAgilidadeEquilibrio,
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 135, -1, -1));
new

        jLabelResistencia.setText("Resistencia Aerobica Geral (804 mts):");
        add(jLabelResistencia,
org.netbeans.lib.awtextra.AbsoluteConstraints(47, 157, -1, -1));
new

        jTextFieldResistencia.setColumns(5);
        add(jTextFieldResistencia,
org.netbeans.lib.awtextra.AbsoluteConstraints(233, 157, -1, -1));
new

        jButtonMaisTestes.setText("Mais Testes");
        add(jButtonMaisTestes,
org.netbeans.lib.awtextra.AbsoluteConstraints(340, 220, -1, -1));
new

    } //GEN-END: initComponents

    private void jButtonRelatorioActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonRelatorioActionPerformed

        gerarRelatorio();

    } //GEN-LAST:event_jButtonRelatorioActionPerformed

    private void jButtonSalvarActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jButtonSalvarActionPerformed

        if(ehValido()) {
            Teste teste = new Teste();

            teste.idTurma = p.idTurma;
            teste.idPessoa = p.idPessoa;
            teste.ano = Integer.parseInt(jTextFieldAno.getText());
            teste.mes = jRadioButtonMar.isSelected() ? 'M' : 'D';
            teste.massaPeso = Double.parseDouble(jTextFieldPeso.getText());
            teste.estatura = Double.parseDouble(jTextFieldEstatura.getText());
            teste.forcaMS = Integer.parseInt(jTextFieldForcaMS.getText());
            teste.coordenacao
            Double.parseDouble(jTextFieldCordenacao.getText());
            teste.agilidadeEquilibrio
            Double.parseDouble(jTextFieldAgilidadeEquilibrio.getText());
            teste.flexibilidade
            Double.parseDouble(jTextFieldFlexibilidade.getText());
            teste.resistenciaAerobica
            Integer.parseInt(jTextFieldResistencia.getText());
            =
            =
            =
            =
            =
        }
    }

```

```

        int resultado = Teste.insereTeste(teste);
        if(resultado == 1){
            JOptionPane.showMessageDialog(this,"Teste      Incluido      com
Sucesso","Inclusão de Testes",
            JOptionPane.INFORMATION_MESSAGE);
            jButtonRelatorio.setEnabled(true);
            jButtonSalvar.setEnabled(false);

            testeAtual = teste;
        }else{
            JOptionPane.showMessageDialog(this,"Falha      na      Inclusão      dos
Testes","Inclusão de Testes",
            JOptionPane.ERROR_MESSAGE);
        }
    }
} //GEN-LAST:event_jButtonSalvarActionPerformed

/**
 * Gera o relatorio em um arquivo texto
 */
private void gerarRelatorio() {
    JFileChooser janela = new JFileChooser();
    janela.setSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    janela.setDialogTitle("Escolha o diretorio de destino para o arquivo");
    janela.showSaveDialog(this);

    File diretorio = janela.getSelectedFile();

    GeradorRelatorio gerador = new GeradorRelatorio(diretorio);
    gerador.gerarRelatorio(testeAtual, p);

    JOptionPane.showMessageDialog(this, "Relatorio criado com sucesso na
pasta \n"+diretorio, "Relatorio", JOptionPane.INFORMATION_MESSAGE);

    this.getParent().remove(this);
}

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_jButtonCancelarActionPerformed
    this.getParent().remove(this);
} //GEN-LAST:event_jButtonCancelarActionPerformed

private void carregaDados() {
    if(p != null) {
        jTextFieldNome.setText(p.nome);
    }
}

private boolean ehValido() {
    boolean valido = true;
    String msg = "Foram encontrado os seguintes problemas: \n";

    if(jTextFieldAno.getText() == null ||
jTextFieldAno.getText().trim().length() != 4) {
        valido = false;
        msg = msg+"Ano deve possuir 4 numeros.\n";
    } else {
        try {
            Integer.parseInt(jTextFieldAno.getText());
        } catch(NumberFormatException e) {
            msg = msg+"Ano deve possuir somente numeros.\n";
            valido = false;
        }
    }
}

```

```

    }
}

    if(jTextFieldPeso.getText() == null ||
jTextFieldPeso.getText().trim().length() == 0) {
    valido = false;
    msg = msg+"Peso não pode ser vazio.\n";
} else {
    try {
        Double.parseDouble(jTextFieldPeso.getText());
    } catch(NumberFormatException e) {
        msg = msg+"Peso deve possuir apenas numeros e \".\" .\n";
        valido = false;
    }
}

    if(jTextFieldEstatura.getText() == null ||
jTextFieldEstatura.getText().trim().length() == 0) {
    valido = false;
    msg = msg+"Estatura não pode ser vazia.\n";
} else {
    try {
        Double.parseDouble(jTextFieldEstatura.getText());
    } catch(NumberFormatException e) {
        msg = msg+"Estatura deve possuir apenas numeros e \".\" .\n";
        valido = false;
    }
}

    if(jTextFieldForcaMS.getText() == null ||
jTextFieldForcaMS.getText().trim().length() == 0) {
    valido = false;
    msg = msg+"Força não pode ser vazia.\n";
} else {
    try {
        Integer.parseInt(jTextFieldForcaMS.getText());
    } catch(NumberFormatException e) {
        msg = msg+"Força deve possuir apenas numeros.\n";
        valido = false;
    }
}

    if(jTextFieldCordenacao.getText() == null ||
jTextFieldCordenacao.getText().trim().length() == 0) {
    valido = false;
    msg = msg+"Coordenação não pode ser vazia.\n";
} else {
    try {
        Double.parseDouble(jTextFieldCordenacao.getText());
    } catch(NumberFormatException e) {
        msg = msg+"Coordenação deve possuir apenas numeros e \".\" .\n";
        valido = false;
    }
}

    if(jTextFieldAgilidadeEquilibrio.getText() == null ||
jTextFieldAgilidadeEquilibrio.getText().trim().length() == 0) {
    valido = false;
    msg = msg+"Agilidade/Equilibrio não pode ser vazia.\n";
} else {
    try {
        Double.parseDouble(jTextFieldAgilidadeEquilibrio.getText());
    } catch(NumberFormatException e) {

```



```

        msg = msg+"Agilidade/Equilibrio deve possuir apenas numeros e
\".\" .\n";
        valido = false;
    }
}

    if(jTextFieldFlexibilidade.getText() == null ||
jTextFieldFlexibilidade.getText().trim().length() == 0) {
        valido = false;
        msg = msg+"Flexibilidade não pode ser vazia.\n";
    } else {
        try {
            Double.parseDouble(jTextFieldFlexibilidade.getText());
        } catch(NumberFormatException e) {
            msg = msg+"Flexibilidade deve possuir apenas numeros e \".\"
.\n";
            valido = false;
        }
    }

    if(jTextFieldResistencia.getText() == null ||
jTextFieldResistencia.getText().trim().length() == 0) {
        valido = false;
        msg = msg+"Resistência não pode ser vazia.\n";
    } else {
        try {
            Integer.parseInt(jTextFieldResistencia.getText());
        } catch(NumberFormatException e) {
            msg = msg+"Resistência deve possuir apenas numeros.\n";
            valido = false;
        }
    }

    if(!valido) {
        JOptionPane.showMessageDialog(this, msg, "Validação",
JOptionPane.ERROR_MESSAGE);
    }

    return valido;
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.ButtonGroup buttonGroupMes;
private javax.swing.JButton jButtonCancelar;
private javax.swing.JButton jButtonMaisTestes;
private javax.swing.JButton jButtonRelatorio;
private javax.swing.JButton jButtonSalvar;
private javax.swing.JLabel jLabelAgilidadeEquilibrio;
private javax.swing.JLabel jLabelAno;
private javax.swing.JLabel jLabelBateria;
private javax.swing.JLabel jLabelCordenacao;
private javax.swing.JLabel jLabelEstatura;
private javax.swing.JLabel jLabelFlexibilidade;
private javax.swing.JLabel jLabelForcaMS;
private javax.swing.JLabel jLabelNome;
private javax.swing.JLabel jLabelPeso;
private javax.swing.JLabel jLabelResistencia;
private javax.swing.JRadioButton jButtononDez;
private javax.swing.JRadioButton jButtononMar;
private javax.swing.JTextField jTextFieldAgilidadeEquilibrio;
private javax.swing.JTextField jTextFieldAno;
private javax.swing.JTextField jTextFieldCordenacao;

```

```

private javax.swing.JTextField jTextFieldEstatura;
private javax.swing.JTextField jTextFieldFlexibilidade;
private javax.swing.JTextField jTextFieldForcaMS;
private javax.swing.JTextField jTextFieldNome;
private javax.swing.JTextField jTextFieldPeso;
private javax.swing.JTextField jTextFieldResistencia;
// End of variables declaration//GEN-END:variables
}

```

New class:

```

/*
 * NewClass.java
 *
 * Created on 24 de Abril de 2005, 22:01
 */

/**
 *
 * @author fabio
 */
public class NewClass {

    /** Creates a new instance of NewClass */
    public NewClass() {
    }

}

```

Pesquisa:

```

package tcc.InterfaceGrafica;

/*
 * Pesquisa.java
 *
 * Created on 19 de Abril de 2005, 18:55
 */

/**
 *
 * @author Fábio
 */
public class Pesquisa extends javax.swing.JPanel {

    /** Creates new form Pesquisa */
    public Pesquisa() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    private void initComponents() { //GEN-BEGIN:initComponents
        java.awt.GridBagConstraints gridBagConstraints;

```

```

buttonGroupMes = new javax.swing.ButtonGroup();
buttonGroupSexo = new javax.swing.ButtonGroup();
jLabelAno = new javax.swing.JLabel();
jTextFieldAno = new javax.swing.JTextField();
jRadioButtonMar = new javax.swing.JRadioButton();
jRadioButtonDez = new javax.swing.JRadioButton();
jLabelIdade = new javax.swing.JLabel();
jTextFieldIdade = new javax.swing.JTextField();
jLabelSexo = new javax.swing.JLabel();
jRadioButtonMasculino = new javax.swing.JRadioButton();
jRadioButtonFeminino = new javax.swing.JRadioButton();
jLabelTurma = new javax.swing.JLabel();
jComboBoxTurma = new javax.swing.JComboBox();
jLabelTeste = new javax.swing.JLabel();
jComboBoxTeste = new javax.swing.JComboBox();
jButtonPesquisar = new javax.swing.JButton();
jButtonCancelar = new javax.swing.JButton();

setLayout(new java.awt.GridBagLayout());

jLabelAno.setText("Ano:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jLabelAno, gridBagConstraints);

jTextFieldAno.setColumns(5);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jTextFieldAno, gridBagConstraints);

buttonGroupMes.add(jRadioButtonMar);
jRadioButtonMar.setText("Mar");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jRadioButtonMar, gridBagConstraints);

buttonGroupMes.add(jRadioButtonDez);
jRadioButtonDez.setText("Dez");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.weightx = 1.0;
add(jRadioButtonDez, gridBagConstraints);

jLabelIdade.setText("Idade:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jLabelIdade, gridBagConstraints);

jTextFieldIdade.setColumns(5);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 1;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jTextFieldIdade, gridBagConstraints);

jLabelSexo.setText("Sexo:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 2;
gridBagConstraints.ipady = 4;

```

```

gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jLabelSexo, gridBagConstraints);

buttonGroupSexo.add(jRadioButtonMasculino);
jRadioButtonMasculino.setText("Masculino");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 2;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jRadioButtonMasculino, gridBagConstraints);

buttonGroupSexo.add(jRadioButtonFeminino);
jRadioButtonFeminino.setText("Feminino");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 2;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jRadioButtonFeminino, gridBagConstraints);

jLabelTurma.setText("Turma:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 3;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jLabelTurma, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 3;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jComboBoxTurma, gridBagConstraints);

jLabelTeste.setText("Testes:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 4;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jLabelTeste, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 4;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
add(jComboBoxTeste, gridBagConstraints);

jButtonPesquisar.setText("Pesquisar");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 5;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.weighty = 1.0;
add(jButtonPesquisar, gridBagConstraints);

jButtonCancelar.setText("Cancelar");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 3;
gridBagConstraints.gridy = 5;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.weightx = 1.0;

```

```

        gridBagConstraints.weighty = 1.0;
        add(jButtonCancelar, gridBagConstraints);

    }//GEN-END:initComponents

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.ButtonGroup buttonGroupMes;
    private javax.swing.ButtonGroup buttonGroupSexo;
    private javax.swing.JButton jButtonCancelar;
    private javax.swing.JButton jButtonPesquisar;
    private javax.swing.JComboBox jComboBoxTeste;
    private javax.swing.JComboBox jComboBoxTurma;
    private javax.swing.JLabel jLabelAno;
    private javax.swing.JLabel jLabelIdade;
    private javax.swing.JLabel jLabelSexo;
    private javax.swing.JLabel jLabelTeste;
    private javax.swing.JLabel jLabelTurma;
    private javax.swing.JRadioButton jRadioButtonDez;
    private javax.swing.JRadioButton jRadioButtonFeminino;
    private javax.swing.JRadioButton jRadioButtonMar;
    private javax.swing.JRadioButton jRadioButtonMasculino;
    private javax.swing.JTextField jTextFieldAno;
    private javax.swing.JTextField jTextFieldIdade;
    // End of variables declaration//GEN-END:variables
}

```

Pessoa:

```

/*
 * Pessoa.java
 *
 * Created on 11 de Maio de 2005, 14:53
 */

package beans;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Date;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import tcc.bd.ConexaoBD;

/**
 *
 * @author fabio
 */
public class Pessoa {

    public int idPessoa;
    public int idTurma;
    public String nome;
    public char sexo;
    public String endereco;
    public String telefone;
    public Date dataNascimento;
}

```

```

public String cep;
public String celular;
public int estadoCivil;
public int escolaridade;

/** Creates a new instance of Pessoa */
public Pessoa() {
}

public String toString() {
    return this.nome;
}

public static int inserePessoa(Pessoa pessoa) {
    int result = 0;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        String sql = "INSERT INTO pessoa (nome, sexo, endereco, telefone,
data_nascimento, cep," +
            " celular, estado_civil, escolaridade, Turma_idTurma) VALUES
(?, ?, ?, ?, ?, ?, ?, ?, ?)";
        Connection conn = ConexaoBD.getConnection();
        stmt = conn.prepareStatement(sql);

        stmt.setString(1, pessoa.nome);
        stmt.setString(2, pessoa.sexo+"");
        stmt.setString(3, pessoa.endereco);
        stmt.setString(4, pessoa.telefone);
        stmt.setDate(5, pessoa.dataNascimento);
        stmt.setString(6, pessoa.cep);
        stmt.setString(7, pessoa.celular);
        stmt.setInt(8, pessoa.estadoCivil);
        stmt.setInt(9, pessoa.escolaridade);
        stmt.setInt(10, pessoa.idTurma);

        result = stmt.executeUpdate();

    } catch (SQLException ex) {
        ex.printStackTrace();

    } finally {
        try {
            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    return result;
}

public static boolean excluirPessoa(Pessoa pessoa) {
    PreparedStatement stmt = null;
    ResultSet rs = null;
    boolean result = false;

    try {
        String sql = "DELETE FROM pessoa WHERE idPessoa = "+pessoa.idPessoa;
        Connection conn = ConexaoBD.getConnection();
        stmt = conn.prepareStatement(sql);

```

```

        result = stmt.execute();
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try {
            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

return result;
}

public static List buscaPessoas(String nome) {
    Statement stmt = null;
    ResultSet rs = null;
    List retorno = null;

    try {
        String sql = "SELECT * FROM pessoa p WHERE p.nome LIKE '"+nome+"%";

        Connection conn = ConexaoBD.getConnection();
        stmt = conn.prepareStatement(sql);

        rs = stmt.executeQuery(sql);

        retorno = getLista(rs);
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try {
            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    return retorno;
}

private static List getLista(ResultSet rs) throws SQLException{
    List retorno = new ArrayList();

    if(rs != null) {

        while(rs.next()) {
            Pessoa pessoa = new Pessoa();

            pessoa.idPessoa = rs.getInt("idPessoa");
            pessoa.idTurma = rs.getInt("Turma_idTurma");
            pessoa.nome = rs.getString("nome");
            pessoa.sexo = rs.getString("sexo").charAt(0);
            pessoa.endereco = rs.getString("endereco");
            pessoa.telefone = rs.getString("telefone");
            pessoa.dataNascimento = rs.getDate("data_nascimento");
            pessoa.cep = rs.getString("cep");
            pessoa.celular = rs.getString("celular");
            pessoa.estadoCivil = rs.getInt("estado_civil");
        }
    }
}

```

```

        pessoa.escolaridade = rs.getInt("escolaridade");

        retorno.add(pessoa);
    }
}

return retorno;
}
}

```

Principal:

```

/*
 * Principal.java
 *
 * Created on 17 de Abril de 2005, 18:27
 */

package tcc.InterfaceGrafica;

import java.awt.CardLayout;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import tcc.bd.ConexaoBD;
import tcc.InterfaceGrafica.ExclusaoPessoa;

/**
 *
 * @author Fábio Davanzo e Renato Castro
 */
public class Principal extends javax.swing.JFrame {

    static Principal instance = null;

    /** Creates new form Principal */
    private Principal() {
        try {
            ConexaoBD.conecte("jdbc:mysql://localhost:3306/tcc", "root", "root");

            UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

            initComponents();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static Principal getInstance() {
        if(instance == null) {
            instance = new Principal();
        }

        return instance;
    }

    /** This method is called from within the constructor to
     * initialize the form.

```



```

* WARNING: Do NOT modify this code. The content of this method is
* always regenerated by the Form Editor.
*/
private void initComponents() { //GEN-BEGIN:initComponents
    jPanelJanela = new javax.swing.JPanel();
    jPanelImagem = new javax.swing.JPanel();
    jMenuItemBar1 = new javax.swing.JMenuItem();
    jMenuItemCadastros = new javax.swing.JMenuItem();
    jMenuItemTurma = new javax.swing.JMenuItem();
    jMenuItemPessoa = new javax.swing.JMenuItem();
    jMenuItemInclusao = new javax.swing.JMenuItem();
    jMenuItemExclusao = new javax.swing.JMenuItem();
    jMenuItemTestes = new javax.swing.JMenuItem();
    jMenuItemTeste = new javax.swing.JMenuItem();
    jMenuItemAjuda = new javax.swing.JMenuItem();
    jMenuItemAjuda = new javax.swing.JMenuItem();
    jMenuItemSobre = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Programa de Avalia\u00e7\u00e3o F\u00edsica e Funcional de
Idosos (AAHPERD)");
    getAccessibleContext().setAccessibleName("Programa de Avalia\u00e7\u00e3o
F\u00edsica e Funcional de Idosos (AAHPERD)");
    jPanelJanela.setLayout(new java.awt.CardLayout());

    jPanelJanela.setPreferredSize(new java.awt.Dimension(640, 480));
    jPanelJanela.add(jPanelImagem, "card2");

    getContentPane().add(jPanelJanela, java.awt.BorderLayout.CENTER);

    jMenuItemCadastros.setText("Cadastros");
    jMenuItemTurma.setText("Turma");
    jMenuItemTurma.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemTurmaActionPerformed(evt);
        }
    });

    jMenuItemCadastros.add(jMenuItemTurma);

    jMenuItemPessoa.setText("Pessoa");
    jMenuItemInclusao.setText("Inclus\u00e3o");
    jMenuItemInclusao.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemInclusaoActionPerformed(evt);
        }
    });

    jMenuItemPessoa.add(jMenuItemInclusao);

    jMenuItemExclusao.setText("Exclus\u00e3o");
    jMenuItemExclusao.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemExclusaoActionPerformed(evt);
        }
    });

    jMenuItemPessoa.add(jMenuItemExclusao);

    jMenuItemCadastros.add(jMenuItemPessoa);

    jMenuItemBar1.add(jMenuItemCadastros);

```

```

jMenuTestes.setText("Testes");
jMenuItemTeste.setText("Teste");
jMenuItemTeste.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemTesteActionPerformed(evt);
    }
});

jMenuTestes.add(jMenuItemTeste);

jMenuBar1.add(jMenuTestes);

jMenuAjuda.setText("Ajuda");
jMenuItemAjuda.setText("Ajuda");
jMenuItemAjuda.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemAjudaActionPerformed(evt);
    }
});

jMenuAjuda.add(jMenuItemAjuda);

jMenuItemSobre.setText("Sobre");
jMenuItemSobre.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemSobreActionPerformed(evt);
    }
});

jMenuAjuda.add(jMenuItemSobre);

jMenuBar1.add(jMenuAjuda);

setJMenuBar(jMenuBar1);

pack();
} //GEN-END: initComponents

private void jMenuItemSobreActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jMenuItemSobreActionPerformed
    String msg = "Este programa é um Trabalho de Conclusão de Curso (TCC) \n"
+
        "do curso de Ciências da Computação - UFSC. " +
        "Desenvolvido por: Fábio Davanzo e Renato Castro";

    JOptionPane.showMessageDialog(this, msg, "Sobre",
JOptionPane.INFORMATION_MESSAGE);
} //GEN-LAST:event_jMenuItemSobreActionPerformed

private void jMenuItemAjudaActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jMenuItemAjudaActionPerformed
    String msg = "Este trabalho ainda não está na sua versão final. \n" +
        "Qualquer dúvida mande um email para: rocastro@inf.ufsc.br ou
franco@inf.ufsc.br .";

    JOptionPane.showMessageDialog(this, msg, "Ajuda",
JOptionPane.INFORMATION_MESSAGE);
} //GEN-LAST:event_jMenuItemAjudaActionPerformed

private void jMenuItemExclusaoActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jMenuItemExclusaoActionPerformed
    setPanel(new ExclusaoPessoa());
} //GEN-LAST:event_jMenuItemExclusaoActionPerformed

```

```

    private void jMenuItemTesteActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jMenuItemTesteActionPerformed
        setPanel(new TestePessoaBusca());
    } //GEN-LAST:event_jMenuItemTesteActionPerformed

    private void jMenuItemInclusaoActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jMenuItemInclusaoActionPerformed
        setPanel(new CadastroPessoa());
    } //GEN-LAST:event_jMenuItemInclusaoActionPerformed

    private void jMenuItemTurmaActionPerformed(java.awt.event.ActionEvent evt)
    { //GEN-FIRST:event_jMenuItemTurmaActionPerformed
        setPanel(new CadastroTurma());
    } //GEN-LAST:event_jMenuItemTurmaActionPerformed

    public void setPanel(JPanel panel) {
        jPanelJanela.removeAll();

        jPanelJanela.add(panel, java.awt.BorderLayout.CENTER);
        ((CardLayout)jPanelJanela.getLayout()).next(jPanelJanela);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Principal().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify //GEN-BEGIN:variables
    private javax.swing.JMenu jMenuItemAjuda;
    private javax.swing.JMenuBar jMenuItemBar1;
    private javax.swing.JMenu jMenuItemCadastros;
    private javax.swing.JMenuItem jMenuItemAjuda;
    private javax.swing.JMenuItem jMenuItemExclusao;
    private javax.swing.JMenuItem jMenuItemInclusao;
    private javax.swing.JMenuItem jMenuItemSobre;
    private javax.swing.JMenuItem jMenuItemTeste;
    private javax.swing.JMenuItem jMenuItemTurma;
    private javax.swing.JMenu jMenuItemPessoa;
    private javax.swing.JMenu jMenuItemTestes;
    private javax.swing.JPanel jPanelImagem;
    private javax.swing.JPanel jPanelJanela;
    // End of variables declaration //GEN-END:variables
}

```

Resistência:

```

package testes;

/**
 *
 * @author fabio
 */

```

```

public abstract class Resistencia {

    /**
     * Retorna o score do teste de Resistencia, de acordo
     * com a idade da pessoa, o sexo e o resultado no teste.
     */
    public static int getResistencia(double resultado, int idade, char sexo) {
        int retorno = 0;

        if(sexo == 'M') {
            retorno = getResistenciaMasculino(resultado, idade);
        } else {
            retorno = getResistenciaFeminino(resultado, idade);
        }

        return retorno;
    }

    private static int getResistenciaMasculino(double resultado, int idade) {
        //TODO: tabela masculina ainda nao foi gerada
        return 0;
    }

    private static int getResistenciaFeminino(double resultado, int idade) {

        if(idade >= 50 && idade < 60) {

            if(resultado > 596) {
                return 0;
            }

            if(resultado <= 596 && resultado > 587) {
                return 1;
            }

            if(resultado <= 587 && resultado > 569) {
                return 2;
            }

            if(resultado <= 569 && resultado > 566) {
                return 4;
            }

            if(resultado <= 566 && resultado > 554) {
                return 5;
            }

            if(resultado <= 554 && resultado > 547) {
                return 6;
            }

            if(resultado <= 547 && resultado > 544) {
                return 8;
            }

            if(resultado <= 544 && resultado > 542) {
                return 9;
            }

            if(resultado <= 542 && resultado > 541) {
                return 10;
            }
        }
    }
}

```

```
if(resultado <= 541 && resultado > 538) {
    return 11;
}

if(resultado <= 538 && resultado > 530) {
    return 12;
}

if(resultado <= 530 && resultado > 528) {
    return 13;
}

if(resultado <= 528 && resultado > 523) {
    return 14;
}

if(resultado <= 523 && resultado > 520) {
    return 15;
}

if(resultado <= 520 && resultado > 518) {
    return 16;
}

if(resultado <= 518 && resultado > 516) {
    return 17;
}

if(resultado <= 516 && resultado > 512) {
    return 18;
}

if(resultado <= 512 && resultado > 511) {
    return 19;
}

if(resultado <= 511 && resultado > 509) {
    return 20;
}

if(resultado <= 509 && resultado > 508) {
    return 21;
}

if(resultado <= 508 && resultado > 505) {
    return 22;
}

if(resultado <= 505 && resultado > 504) {
    return 25;
}

if(resultado <= 504 && resultado > 502) {
    return 26;
}

if(resultado <= 502 && resultado > 500) {
    return 27;
}

if(resultado <= 500 && resultado > 499) {
    return 29;
}
```

```
    if(resultado <= 499 && resultado > 497) {
        return 30;
    }

    if(resultado <= 497 && resultado > 496) {
        return 32;
    }

    if(resultado <= 496 && resultado > 495) {
        return 34;
    }

    if(resultado <= 495 && resultado > 494) {
        return 36;
    }

    if(resultado <= 494 && resultado > 491) {
        return 37;
    }

    if(resultado <= 491 && resultado > 489) {
        return 38;
    }

    if(resultado <= 489 && resultado > 488) {
        return 40;
    }

    if(resultado <= 488 && resultado > 487) {
        return 41;
    }

    if(resultado <= 487 && resultado > 485) {
        return 43;
    }

    if(resultado <= 485 && resultado > 484) {
        return 45;
    }

    if(resultado <= 484 && resultado > 483) {
        return 47;
    }

    if(resultado <= 483 && resultado > 482) {
        return 48;
    }

    if(resultado <= 482 && resultado > 481) {
        return 50;
    }

    if(resultado <= 481 && resultado > 480) {
        return 51;
    }

    if(resultado <= 480 && resultado > 478) {
        return 52;
    }

    if(resultado <= 478 && resultado > 477) {
        return 53;
    }
```

```

}

if(resultado <= 477 && resultado > 476) {
    return 54;
}

if(resultado <= 476 && resultado > 474) {
    return 55;
}

if(resultado <= 474 && resultado > 472) {
    return 57;
}

if(resultado <= 472 && resultado > 470) {
    return 59;
}

if(resultado <= 470 && resultado > 469) {
    return 61;
}

if(resultado <= 469 && resultado > 468) {
    return 62;
}

if(resultado <= 468 && resultado > 466) {
    return 62;
}

if(resultado <= 466 && resultado > 463) {
    return 63;
}

if(resultado <= 463 && resultado > 459) {
    return 65;
}

if(resultado <= 459 && resultado > 457) {
    return 68;
}

if(resultado <= 457 && resultado > 455) {
    return 69;
}

if(resultado <= 452 && resultado > 451) {
    return 70;
}

if(resultado <= 451 && resultado > 450) {
    return 73;
}

if(resultado <= 450 && resultado > 449) {
    return 74;
}

if(resultado <= 449 && resultado > 448) {
    return 75;
}

if(resultado <= 448 && resultado > 447) {

```

```
        return 76;
    }

    if(resultado <= 447 && resultado > 443) {
        return 77;
    }

    if(resultado <= 443 && resultado > 442) {
        return 78;
    }

    if(resultado <= 442 && resultado > 441) {
        return 79;
    }

    if(resultado <= 441 && resultado > 440) {
        return 81;
    }

    if(resultado <= 440 && resultado > 439) {
        return 83;
    }

    if(resultado <= 439 && resultado > 438) {
        return 84;
    }

    if(resultado <= 438 && resultado > 437) {
        return 86;
    }

    if(resultado <= 437 && resultado > 434) {
        return 87;
    }

    if(resultado <= 434 && resultado > 432) {
        return 88;
    }

    if(resultado <= 432 && resultado > 428) {
        return 89;
    }

    if(resultado <= 428 && resultado > 426) {
        return 90;
    }

    if(resultado <= 426 && resultado > 423) {
        return 91;
    }

    if(resultado <= 423 && resultado > 420) {
        return 92;
    }

    if(resultado <= 420 && resultado > 414) {
        return 93;
    }

    if(resultado <= 414 && resultado > 409) {
        return 95;
    }
}
```



```

    if(resultado <= 409 && resultado > 398) {
        return 96;
    }

    if(resultado <= 398 && resultado > 397) {
        return 97;
    }

    if(resultado <= 397 && resultado > 393) {
        return 98;
    }

    if(resultado == 393 ) {
        return 99;
    }

    if(resultado < 393) {
        return 100;
    }
} else if(idade > 60) {
if(resultado > 727) {
    return 0;
}

if(resultado <= 727 && resultado > 647) {
    return 1;
}

if(resultado <= 947 && resultado > 628) {
    return 2;
}

if(resultado <= 628 && resultado > 624) {
    return 3;
}

if(resultado <= 624 && resultado > 612) {
    return 4;
}

if(resultado <= 612 && resultado > 602) {
    return 5;
}

if(resultado <= 602 && resultado > 590) {
    return 6;
}

if(resultado <= 590 && resultado > 585) {
    return 7;
}

if(resultado <= 585 && resultado > 578) {
    return 8;
}

if(resultado <= 578 && resultado > 571) {
    return 9;
}

if(resultado <= 571 && resultado > 568) {

```

```
        return 10;
    }

    if(resultado <= 568 && resultado > 565) {
        return 11;
    }

    if(resultado <= 565 && resultado > 563) {
        return 12;
    }

    if(resultado <= 563 && resultado > 561) {
        return 13;
    }

    if(resultado <= 561 && resultado > 557) {
        return 14;
    }

    if(resultado <= 557 && resultado > 553) {
        return 15;
    }

    if(resultado <= 553 && resultado > 551) {
        return 16;
    }

    if(resultado <= 551 && resultado > 548) {
        return 17;
    }

    if(resultado <= 548 && resultado > 547) {
        return 18;
    }

    if(resultado <= 547 && resultado > 537) {
        return 19;
    }

    if(resultado <= 537 && resultado > 535) {
        return 20;
    }

    if(resultado <= 535 && resultado > 530) {
        return 21;
    }

    if(resultado <= 530 && resultado > 527) {
        return 22;
    }

    if(resultado <= 527 && resultado > 526) {
        return 23;
    }

    if(resultado <= 526 && resultado > 525) {
        return 24;
    }

    if(resultado <= 525 && resultado > 524) {
        return 26;
    }
}
```

```
if(resultado <= 524 && resultado > 523) {
    return 27;
}

if(resultado <= 523 && resultado > 521) {
    return 28;
}

if(resultado <= 521 && resultado > 519) {
    return 29;
}

if(resultado <= 519 && resultado > 518) {
    return 30;
}

if(resultado <= 518 && resultado > 517) {
    return 32;
}

if(resultado <= 517 && resultado > 516) {
    return 33;
}

if(resultado <= 516 && resultado > 515) {
    return 34;
}

if(resultado <= 515 && resultado > 514) {
    return 35;
}

if(resultado <= 514 && resultado > 511) {
    return 36;
}

if(resultado <= 511 && resultado > 510) {
    return 37;
}

if(resultado <= 510 && resultado > 509) {
    return 38;
}

if(resultado <= 509 && resultado > 508) {
    return 39;
}

if(resultado <= 508 && resultado > 506) {
    return 41;
}

if(resultado <= 506 && resultado > 505) {
    return 43;
}

if(resultado <= 505 && resultado > 505) {
    return 44;
}

if(resultado <= 504 && resultado > 503) {
    return 45;
}
```

```
    if(resultado <= 503 && resultado > 502) {
        return 46;
    }

    if(resultado <= 502 && resultado > 501) {
        return 47;
    }

    if(resultado <= 501 && resultado > 500) {
        return 48;
    }

    if(resultado <= 500 && resultado > 499) {
        return 49;
    }

    if(resultado <= 499 && resultado > 498) {
        return 50;
    }

    if(resultado <= 498 && resultado > 497) {
        return 51;
    }

    if(resultado <= 497 && resultado > 496) {
        return 52;
    }

    if(resultado <= 496 && resultado > 492) {
        return 56;
    }

    if(resultado <= 492 && resultado > 491) {
        return 57;
    }

    if(resultado <= 491 && resultado > 490) {
        return 59;
    }

    if(resultado <= 490 && resultado > 489) {
        return 60;
    }

    if(resultado <= 489 && resultado > 488) {
        return 61;
    }

    if(resultado <= 488 && resultado > 487) {
        return 62;
    }

    if(resultado <= 487 && resultado > 486) {
        return 63;
    }

    if(resultado <= 486 && resultado > 485) {
        return 64;
    }

    if(resultado <= 485 && resultado > 484) {
        return 65;
    }
```

```
}

if(resultado <= 484 && resultado > 483) {
    return 66;
}

if(resultado <= 483 && resultado > 482) {
    return 67;
}

if(resultado <= 482 && resultado > 478) {
    return 68;
}

if(resultado <= 478 && resultado > 473) {
    return 69;
}

if(resultado <= 473 && resultado > 471) {
    return 70;
}

if(resultado <= 471 && resultado > 470) {
    return 71;
}

if(resultado <= 470 && resultado > 469) {
    return 72;
}

if(resultado <= 469 && resultado > 467) {
    return 74;
}

if(resultado <= 467 && resultado > 464) {
    return 75;
}

if(resultado <= 464 && resultado > 463) {
    return 78;
}

if(resultado <= 463 && resultado > 461) {
    return 79;
}

if(resultado <= 461 && resultado > 456) {
    return 80;
}

if(resultado <= 456 && resultado > 455) {
    return 81;
}

if(resultado <= 455 && resultado > 452) {
    return 82;
}

if(resultado <= 452 && resultado > 450) {
    return 83;
}

if(resultado <= 450 && resultado > 449) {
```

```

        return 84;
    }

    if(resultado <= 449 && resultado > 448) {
        return 85;
    }

    if(resultado <= 448 && resultado > 445) {
        return 86;
    }

    if(resultado <= 445 && resultado > 444) {
        return 87;
    }

    if(resultado <= 444 && resultado > 443) {
        return 89;
    }

    if(resultado <= 443 && resultado > 445) {
        return 91;
    }

    if(resultado <= 445 && resultado > 436) {
        return 92;
    }

    if(resultado <= 436 && resultado > 432) {
        return 93;
    }

    if(resultado <= 432 && resultado > 427) {
        return 94;
    }

    if(resultado <= 427 && resultado > 422) {
        return 95;
    }

    if(resultado <= 422 && resultado > 417) {
        return 96;
    }

    if(resultado <= 417 && resultado > 398) {
        return 97;
    }

    if(resultado <= 398 && resultado > 393) {
        return 98;
    }
    if(resultado == 393 ) {
        return 99;
    }

    if(resultado < 393) {
        return 100;
    }
}
return 0;
}
}

```

Teste:

```
/*
 * Teste.java
 *
 * Created on 13 de Junho de 2005, 22:31
 */

package beans;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import tcc.bd.ConexaoBD;
import utils.Utills;

/**
 *
 * @author fabio
 */
public class Teste {

    public int idTurma;
    public int idPessoa;
    public int ano;
    public char mes;
    public double massaPeso;
    public double estatura;
    public int forcaMS;
    public double coordenacao;
    public double agilidadeEquilibrio;
    public int resistenciaAerobica;
    public double flexibilidade;

    /** Creates a new instance of Teste */
    public Teste() {
    }

    /** Retorna string no formato "mes_ano" */
    public String toString() {
        return Utills.getMes(mes)+"_"+ano;
    }

    public static int insereTeste(Teste teste) {
        int result = 0;

        PreparedStatement stmt = null;
        ResultSet rs = null;

        try {
            String sql = "INSERT INTO teste (Pessoa_Turma_idTurma,
Pessoa_idPessoa, ano, mes, massa_peso, estatura, forca_ms," +
                " coordenacao, agilidade_equilibrio, resistencia_aerobica,
flexibilidade) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

            Connection conn = ConexaoBD.getConnection();
            stmt = conn.prepareStatement(sql);

            stmt.setInt(1, teste.idTurma);
            stmt.setInt(2, teste.idPessoa);
```

```

        stmt.setInt(3, teste.ano);
        stmt.setString(4, teste.mes+"");
        stmt.setDouble(5, teste.massaPeso);
        stmt.setDouble(6, teste.estatura);
        stmt.setDouble(7, teste.forcaMS);
        stmt.setDouble(8, teste.coordenacao);
        stmt.setDouble(9, teste.agilidadeEquilibrio);
        stmt.setInt(10, teste.resistenciaAerobica);
        stmt.setDouble(11, teste.flexibilidade);

        result = stmt.executeUpdate();

    } catch (SQLException ex) {
        ex.printStackTrace();

    } finally {
        try {
            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    return result;
}
}

```

Teste Pessoa Busca:

```

/*
 * TestePessoaBusca.java
 *
 * Created on 26 de Abril de 2005, 15:53
 */

package tcc.InterfaceGrafica;

import beans.Pessoa;
import java.awt.CardLayout;
import java.awt.Container;
import java.awt.Cursor;
import java.sql.ResultSet;
import java.util.Iterator;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
/**
 *
 * @author Fábio
 */
public class TestePessoaBusca extends javax.swing.JPanel {

    /** Creates new form TestePessoaBusca */
    public TestePessoaBusca() {
        initComponents();
    }

    /** This method is called from within the constructor to

```



```

* initialize the form.
* WARNING: Do NOT modify this code. The content of this method is
* always regenerated by the Form Editor.
*/
private void initComponents() { //GEN-BEGIN:initComponents
    java.awt.GridBagConstraints gridBagConstraints;

    jLabelNome = new javax.swing.JLabel();
    jTextFieldNome = new javax.swing.JTextField();
    jButtonProcura = new javax.swing.JButton();
    jButtonAvancar = new javax.swing.JButton();
    jButtonCancelar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();

    setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabelNome.setText("Nome:");
    add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 2, -
1, -1));

    jTextFieldNome.setColumns(30);
    add(jTextFieldNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
2, 224, 23));

    jButtonProcura.setText("Procura");
    jButtonProcura.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonProcuraActionPerformed(evt);
        }
    });

    add(jButtonProcura,
org.netbeans.lib.awtextra.AbsoluteConstraints(254, 2, -1, -1));

    jButtonAvancar.setText("Avan\u00e7ar");
    jButtonAvancar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonAvancarActionPerformed(evt);
        }
    });

    add(jButtonAvancar, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
260, -1, -1));

    jButtonCancelar.setText("Cancelar");
    jButtonCancelar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonCancelarActionPerformed(evt);
        }
    });

    add(jButtonCancelar,
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 260, -1, -1));

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Pessoa"
        }
    ) {

```

```

        boolean[] canEdit = new boolean [] {
            false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    jScrollPane1.setViewportViewView(jTable1);

    add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
27, 320, 220));

    }//GEN-END:initComponents

    private void jButtonAvancarActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButtonAvancarActionPerformed
        Pessoa p = getPessoaSelecionada();

        if(p != null) {
            InterfaceTeste it = new InterfaceTeste(p);

            Container pai = this.getParent();
            pai.add(it, java.awt.BorderLayout.CENTER);
            ((CardLayout)this.getParent().getLayout()).next(pai);
        } else {

            JOptionPane.showMessageDialog(this, "Nenhuma pessoa selecionada",
"Atenção", JOptionPane.WARNING_MESSAGE);
        }
    }//GEN-LAST:event_jButtonAvancarActionPerformed

    private void jButtonProcuraActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButtonProcuraActionPerformed
        this.setCursor(new Cursor(Cursor.WAIT_CURSOR));

        DefaultTableModel model = ((DefaultTableModel)jTable1.getModel());

        //LIMPA A TABELA
        model.setRowCount(0);

        String nome = jTextFieldNome.getText() != null ?
jTextFieldNome.getText(): "";

        List lista = Pessoa.buscaPessoas(nome);

        Iterator iter = lista.iterator();
        while(iter.hasNext()) {
            Pessoa p = (Pessoa)iter.next();

            model.addRow(new Object[]{p});
        }

        this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
    }//GEN-LAST:event_jButtonProcuraActionPerformed

    private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt)
    {//GEN-FIRST:event_jButtonCancelarActionPerformed
        // Principal.getInstance().cancelar(this);
        this.getParent().remove(this);
    }//GEN-LAST:event_jButtonCancelarActionPerformed

    private Pessoa getPessoaSelecionada() {

```

```

        Pessoa pessoaRetorno = null;

        int linhaSelec = jTable1.getSelectedRow();

        if(linhaSelec >= 0) {
            pessoaRetorno = (Pessoa)jTable1.getValueAt(linhaSelec, 0);
        }

        return pessoaRetorno;
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButtonAvancar;
    private javax.swing.JButton jButtonCancelar;
    private javax.swing.JButton jButtonProcura;
    private javax.swing.JLabel jLabelNome;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField jTextFieldNome;
    // End of variables declaration//GEN-END:variables
}

```

Turma:

```

/*
 * Turma.java
 *
 * Created on 11 de Maio de 2005, 14:28
 */

package beans;

import java.util.List;
import java.util.LinkedList;
import java.sql.*;
import tcc.bd.ConexaoBD;

/**
 *
 * @author fabio
 */
public class Turma {

    public int idTurma;
    public String nome;
    public String dia;
    public String hora;
    public int universidade;

    /** Creates a new instance of Turma */
    public Turma() {
    }

    public String toString() {
        return this.nome;
    }

    public static List getTurmas() {
        List lista = new LinkedList();
    }
}

```

```

Statement stmt = null;
ResultSet rs = null;

try {
    String sql = "SELECT * FROM turma";
    Connection conn = ConexaoBD.getConnection();
    stmt = conn.createStatement();
    rs = stmt.executeQuery(sql);

    while (rs.next()) {
        Turma t = new Turma();
        t.idTurma = rs.getInt("idTurma");
        t.nome = rs.getString("nome");
        t.dia = rs.getString("dia");
        t.hora = rs.getString("hora");
        t.universidade = rs.getInt("universidade");

        lista.add(t);
    }

} catch (SQLException ex) {
    ex.printStackTrace();
} finally {
    try {
        stmt.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

return lista;
}

/**
 * Retorna a turma com chave "idTurma"
 */
public static Turma getTurma(int idTurma) {
    Turma turma = null;

    Statement stmt = null;
    ResultSet rs = null;

    try {
        String sql = "SELECT * FROM turma where idTurma = "+idTurma;
        Connection conn = ConexaoBD.getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(sql);

        if(rs.next()) {
            turma = new Turma();
            turma.idTurma = rs.getInt("idTurma");
            turma.nome = rs.getString("nome");
            turma.dia = rs.getString("dia");
            turma.hora = rs.getString("hora");
            turma.universidade = rs.getInt("universidade");
        }

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {

```

```

        try {
            stmt.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    return turma;
}

public static int insereTurma(Turma turma) {
    int result = 0;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        String sql = "INSERT INTO turma (nome, universidade, dia, hora)
VALUES (?, ?, ?, ?)";
        Connection conn = ConexaoBD.getConnection();
        stmt = conn.prepareStatement(sql);

        stmt.setString(1, turma.nome);
        stmt.setInt(2, turma.universidade);
        stmt.setString(3, turma.dia);
        stmt.setString(4, turma.hora);

        result = stmt.executeUpdate();

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try {
            stmt.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    return result;
}
}

```

Utils:

```

/*
 * Utils.java
 *
 * Created on 4 de Julho de 2005, 21:21
 */

package utils;

import beans.Pessoa;

/**
 *
 * @author fabio
 */

```

```

*/
public abstract class Utils {

    /** Decodifica o mes representado por 'c'. */
    public static String getMes(char mes) {

        return mes == 'M' ? "Março" : "Dezembro";
    }

    /** Decodifica o sexo representado por 'c'. */
    public static String getSexo(char sexo) {

        return sexo == 'M' ? "Masculino" : "Feminino";
    }

    /**
     * Calcula e retorna a idade da pessoa
     */
    public static int getIdade(Pessoa p) {
        long tempo = System.currentTimeMillis() - p.dataNascimento.getTime();

        long idade = (((tempo / (long)1000) / (long)60) / (long)60) / (long)24)
        / (long)365;

        return (int)idade;
    }
}

```

8.2 Artigo

DESENVOLVIMENTO DE UM PROGRAMA QUE AVALIAÇÃO CAPACIDADE FÍSICA E FUNCIONAL DO IDOSO

Fábio Davanzo Franco de Oliveira, Renato Oliveira e Castro(*)

RESUMO

Esse trabalho contribuiu para um programa de capacitação física do idoso. Para isto foi desenvolvido um software que armazena dados colhidos em campo, e possibilita o posterior tratamento destas informações. O software facilita o estudo desses dados, gerando um relatório simples para o usuário de como ele está com relação a sua saúde e seu desempenho físico.

ABSTRACT

This work shows software that was developed to help a program of qualification of elder people for physical fitness. Such software allows storing of field-collected data, to arrange them in a database in order to information management. Also, it makes possible to generate a report about any subject that is being evaluated and keeps individual records updated.

(*) Alunos do INE / CTC.

INTRODUÇÃO

Esse trabalho visa a implementação de um programa capaz de avaliar a capacidade física e funcional do idoso, auxiliando assim um projeto da Universidade Federal de Santa Catarina no seu desenvolvimento e tornando mais simples o estudos dos dados coletados. Já que o programa armazena e organiza esses dados, além de gerar um relatório de como o idoso está posteriormente a sua avaliação.

CONTEÚDO DO ARTIGO

Em nosso País, no início da década de 80, ainda não era comum encontrarmos estudos normativos preocupados em relacionar a aptidão física à saúde. Temos nesse contexto que a literatura é rica em avaliações voltadas à performance motora, não incluindo, na maioria das vezes, todos os componentes da aptidão física relacionada à saúde, proposta pela *AAHPERD (American Alliance for Health, Physical Education, Recreation)* 1984.

O emprego de normas de aptidão física relacionada à saúde começou então, timidamente, a ser utilizado por alguns profissionais em suas instituições de ensino. Alguns estudos, nesse campo, foram realizados por Barbanti (1982, *apud* Revista Eletrônica da Terceira Idade), Nahas (1985, *ibidem*) e Madureira (1987, *ibidem*). O primeiro e o terceiro abrangiam população de escolares, enquanto o segundo, universitários. Após isto, outros estudos foram realizados contribuindo para a ampliação da bibliografia na área.¹⁰

Há uma diferenciação entre a aptidão física relacionada à saúde e a aptidão motora. A diferença na bateria de testes de aptidão física relacionada à saúde e aptidão motora está no fato de que a primeira solicita apenas os componentes estritamente relacionados à saúde do indivíduo, ou seja, a função cardiorrespiratória e função músculo-esquelética enquanto que a segunda bateria, além desses componentes, pode envolver testes de agilidade, força, equilíbrio e velocidade (AAHPERD, 1984).

¹⁰ Idade Ativa: revista eletrônica da Terceira Idade. Disponível em:

<http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>

Para Pate (1983 *apud* Mazo et al., 2003), os componentes de aptidão motora e aptidão física relacionada à saúde nem sempre têm sido claramente distinguidos uns dos outros. A aptidão motora está relacionada diretamente à performance, pois seus testes exigem, em grande parte, agilidade, velocidade, força, potência, resistência muscular, equilíbrio etc. (*apud* Mazo et al., 2003).

Conforme explica Pate (1983, *ibidem*), comparando aptidão motora com aptidão física relacionada à saúde, o segundo apresenta um conceito restrito, que geralmente inclui somente os componentes associados à prevenção e/ou promoção da saúde.

O estabelecimento do conceito de aptidão física relacionada à saúde derivou de estudos clínicos que evidenciaram a incidência maior de problemas de saúde entre adultos e indivíduos mais idosos de vida sedentária. Sua definição diz que:

“é a habilidade de executar atividades físicas com energia e vigor sem excesso de fadiga, e a demonstração de qualidades e capacidades físicas que conduzam ao menor risco de desenvolvimento de doenças hipocinéticas” (Pate, 1983, p. 19 *apud* Mazo et al., 2003).

É necessário, neste momento, frisar que não se é contra as baterias de testes de aptidão motora, mas que a preocupação reside, sobretudo, na dificuldade de aplicação dos testes, pelo tempo que necessita ser dispensado e pelo próprio acompanhamento de cada usuário.

Hoje em dia, pela dificuldade em se adotar uma bateria de testes mais simples que exija menos tempo na coleta dos dados, grande parte dos profissionais acabam por deixar de realizar qualquer tipo de teste, conseqüentemente não ficam sabendo quase nada sobre o desenvolvimento do seu aluno.

O objetivo de trabalhos assim é apresentar mais um recurso na avaliação do usuário, procurando levar em consideração a realidade sócioeconômica, as dificuldades existentes, o tempo dispendido na coleta dos dados, a validade, a objetividade e a fidedignidade dos testes.

Admite-se que a bateria de testes da AAHPERD (1984) apresenta o mínimo necessário para que se possa prevenir, manter e/ou desenvolver um trabalho básico voltado à saúde do usuário e não só a avaliações motoras do mesmo.

Cabe perguntar: como viabilizar este trabalho? Primeiramente, necessita-se definir quais as medidas e os testes a serem aplicados. Alguns deles como Força de membros inferiores – pernas, Força de membros superiores – braço direito, Flexibilidade (FLEX), Flexibilidade de ombros: Alcançar atrás costas D – direito e E – esquerdo, Dinâmico e agilidade, Coordenação (COO), entre outros. Após

coletados os dados, passa-se à elaboração dos *percentis/decis* e, assim, ter-se-á uma representação numa escala (norma) que é capaz de informar como determinada pessoa está em relação ao grupo ao qual pertence. A estatística auxiliará na elaboração das tabelas para cada variável, as quais fornecerão as normas antropométricas e de aptidão física relacionada à saúde.

Com base nisso este trabalho propõe um software para gerenciar e administrar todos esses dados gerando um banco de dados que armazenará todos os testes, dados pessoais, resultados, enfim o que for relevante para se fazer um bom levantamento de como a pessoa está com relação aos demais. Assim, pode-se mostrar a evolução do usuário no decorrer do ano e imprimir um relatório qualitativo e de fácil entendimento. Isto auxiliaria os avaliadores; que não mais ficaram horas na frente do computador analisando planilhas de dados. Também o usuário pode ter um relatório simplificado, com seus dados, seu desempenho, e tudo isso de forma bastante rápida.

O projeto/software foi desenvolvido em Java 1.5, sendo utilizado o *Net Beans 4.0* para a implementação do programa, e usaremos um banco de dados, em *Mysql 4.1* e *dbdesigner 4.0*, para armazenar as informações. A interface do mesmo, o tratamento dos dados recebidos, a geração do relatório, e algumas consultas ao banco de dados foram projetados com relação ao AAHPERD.

COMENTÁRIOS FINAIS

O projeto permite a realização de diversos projetos futuros.

Uma idéia que seria muito útil e interessante seria criar um banco de dados via web, que poderia interligar todas as universidades ou instituições onde os testes são realizados sendo assim se teria um banco de dados muito mais completo e poderia se realizar pesquisas muito mais detalhadas e precisas, pois haveria um número muito maior de idosos de diversos lugares e regiões.

O que pode ser bom até mesmo para o próprio software, adicionando novos tipos de testes, realizando assim cadastros de pessoas mais detalhadas e precisos, o que acarreta realizar buscas mais detalhadas que auxiliam nas pesquisas. O software já tem um campo em aberto para a inserção de novos testes e novos dados cadastrais.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Idade Ativa: revista eletrônica da Terceira Idade. Disponível em:
<http://www.techway.com.br/techway/revista_idoso/lazer/lazer_audre.htm>.
Acesso em: abril 2005.

[2] Mazo, Giovana Z., Lopes, Marize A. e Benedetti, Tânia B. A Atividade Física e o Idoso: concepção gerontológica 2003)

[3] Criar Web, O que é a POO?. Disponível em:
<<http://www.criarweb.com/artigos/215.php?manual=11>>.
Acesso em: Abril 2005.

[4] Tecnologia - Enovativa Internet Solutions. Disponível em:
<<http://www.enovativa.com.br/tecnologia.html>>.
Acesso em: Abril 2005.

[5] Banco de Dados. Disponível em:
<http://www.webinformation.hpg.ig.com.br/computer_bancodedados.htm>.
Acesso em: Abril 2005.

[6] Java Technology. Disponível em:
<<http://java.sun.com/>>.
Acesso em: Abril 2005

[7] MySQL AB : The World's Most Popular Open Source Database. Disponível em:
<<http://www.mysql.com/>>.
Acesso em: abril 2005.

[8] DB Design Group. Disponível em:
<<http://www.dbdesign.com/>>
Acesso em: abril 2005.

[9] AAHPERD (American Alliance for Health, Physical Education, Recreation) 1984. Disponível em:
<<http://www.aahperd.org/>>
Acesso em: abril 2005.