Trabalho
Planejamento Financeiro

Sexta-feira, 8 de maio de 2015.

Desenvolva um aplicativo Java GUI para permitir o planejamento financeiro pessoal. Este programa deve fornecer os seguintes serviços:

1. Cadastro das receitas mensais incluindo data e valor.

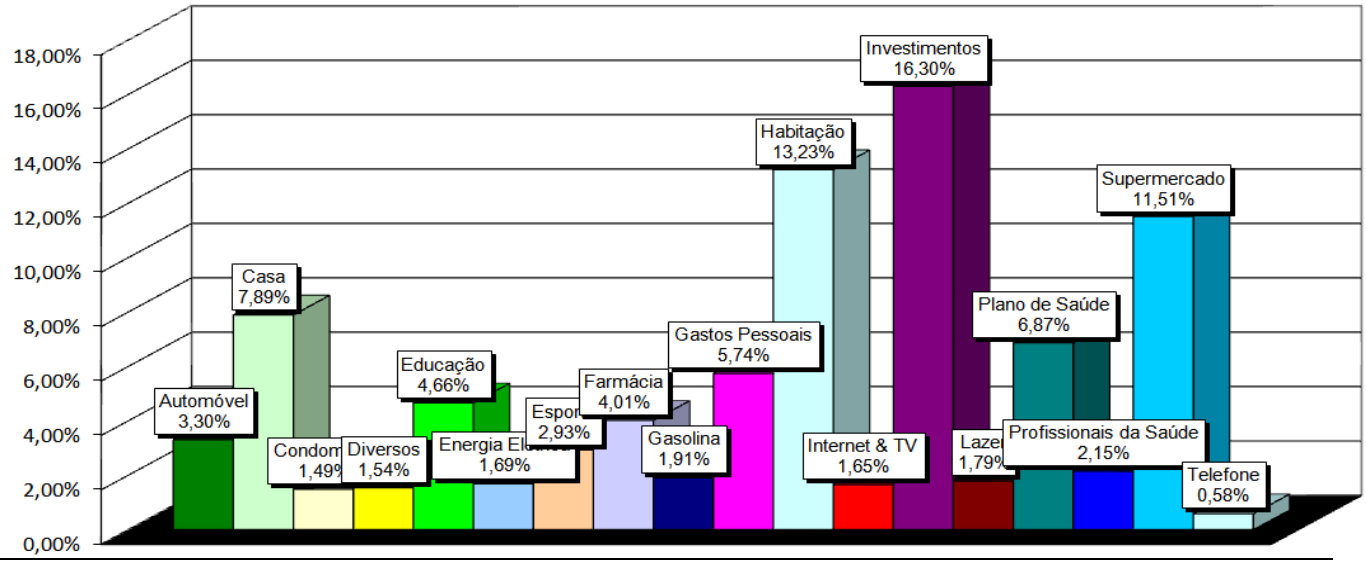
2. Cadastro de despesas mensais incluindo a data da despesa, o dia do pagamento, a forma de pagamento, o número do cheque (se forma de pagamento é cheque), número de parcelas (se pagamento não é à vista), descrição, categoria (p. ex., educação, lazer, alimentação) e valor da despesa.

   As despesas do planejamento financeiro devem incluir os gastos essenciais (educação, alimentação, habitação, etc.), gastos com estilo de vida pessoal (cinema, academia, viagens, etc.) e gastos com investimentos (poupança, fundo de renda fixa, ações, etc.). Esta organização possibilita a categorização das despesas do usuário.

3. Cadastro de metas por categoria de despesa. Esta função permite ao usuário definir quais valores em reais ele planeja gastar com cada categoria em determinado mês.

4. Balanço mensal com os valores da receita, dos investimentos (despesas com aplicações financeiras), gastos (despesas com gastos essenciais e do estilo de vida pessoal) e saldo atual. Usar um **gráfico de pizza** para exibir também uma visão gráfica deste balanço mensal.

5. Categorização das despesas mensais através de um **gráfico em barras** (ver exemplo abaixo). Deve ser gerado dois gráficos: um para exibir os valores em reais e outro em valores percentuais relativos a receita total do usuário.



6. Acompanhamento de metas por categoria através da verificação dos valores gastos em cada categoria com os valores previstos para cada uma. O programa deve sempre emitir um alerta quando o valor total das despesas em cada categoria atingir o valor definido como meta para cada categoria (sinal vermelho) e quando o valor total estiver a 70% (por exemplo) da meta (sinal amarelo). O valor do percentual do sinal amarelo deve ser definido pelo usuário. Usar um **gráfico em linhas** para exibir também uma visão gráfica do acompanhamento de metas mensal.

7. Balanço mensal com os valores totais das despesas segundo a forma de pagamento utilizada. Usar um **gráfico de pizza** para exibir também uma visão gráfica deste balanço mensal.

1. Modelo lógico do banco de dados

Notação: chave primária sublinhada; chave estrangeira precedida por cerquilha (#).

1. Renda(CódigoRenda, Descrição)
2. RendaMensal(#CódigoRenda, Data, Valor)
3. FormaPagamento(CódigoPagamento, Descrição)
4. Categoria(CódigoCategoria, Descrição)
5. Despesa(CódigoDespesa, Descrição, #CódigoCategoria, DataDespesa, DataPagamento, #CódigoPagamento, NúmeroCheque, Valor, NumeroDeParcelas)

   Nota: Se a forma de pagamento realizada é cheque, indicada através da chave estrangeira **CódigoPagamento**, o número do cheque deve ser armazenado no campo **NúmeroCheque**.

6. PlanejamentoMensal(MesAnoPlanejamento, #CódigoDespesa)
7. MetaMensal(MesAnoMeta, #CódigoCategoria, Valor)

2. Critérios de avaliação

1. O trabalho será avaliado considerando:

   1.1 - tratamento dos dados fornecidos pelo usuário e dos cálculos que possam abortar a execução do programa via **tratamento de exceções**;
   1.2 - a lógica empregada na solução do problema;
   1.3 - o funcionamento do programa;
   1.4 - a usabilidade[1] do programa e da interface gráfica com o usuário;
   1.5 - o conhecimento da linguagem de programação;
   1.6 - o uso do paradigma de orientação a objetos;
   1.7 - código fonte Java sem erros e sem advertências (*warnings*) do compilador;

[1] A usabilidade está diretamente ligada a interface e a capacidade do *software* em permitir que o usuário alcance suas metas de interação com o sistema. Ser de fácil aprendizagem, permitir uma utilização eficiente e apresentar poucos erros, são os aspectos fundamentais para a percepção da boa usabilidade por parte do usuário. Mas a usabilidade pode ainda estar relacionada com a facilidade de ser memorizada e ao nível de satisfação do usuário.

Fonte: http://pt.wikipedia.org/wiki/Usabilidade

1.8 - código fonte Java legível, indentado, organizado e comentado.
1.9 - documentação do sistema em HTML gerada através da ferramenta *javadoc*;
1.10 - uso da API *JFreeChart* para a geração dos gráficos.

2. O trabalho pode ser desenvolvido em equipe composta por no máximo 2 (dois) membros.

3. A implementação das tabelas e seus relacionamentos deve ser feita usando os recursos de **processamento de arquivo de acesso aleatório** do Java I/O (arquivos) ou banco de dados via JDBC.

4. Não é permitido o uso de *frameworks*, API, bibliotecas de classes ou qualquer ferramenta que permita, de maneira automática para o programador - sem escrever comandos SQL -, a persistência e a recuperação de objetos em banco de dados relacionais.

5. Os gráficos devem ser gerados pelo programa usando a API *JFreeChart*, disponível em http://www.jfree.org/jfreechart/.

3. Artefatos de software

Os seguintes artefatos de *software* devem ser entregues:

a) o projeto Java desenvolvido na IDE Eclipse;
b) o arquivo JAR executável da aplicação;
c) a documentação HTML do sistema Java gerado com o *javadoc*.

Compactar todos os artefatos criando um arquivo ZIP com o nome da equipe.

4. Data e forma de apresentação

- Terça-feira, 23 de junho de 2015.
- Entrevista sobre o desenvolvimento e o funcionamento do sistema.

5. Valor do trabalho

25,0 (vinte e cinco) pontos.

Prof**. M**árlon **O**liveira **d**a **S**ilva
marlon.silva@ifsudestemg.edu.br