

## Trabalho

### Avaliação Física

Segunda-feira, 21 de maio de 2018.

Um profissional de educação física que trabalha como *personal trainer* necessita de um aplicativo que o permita acompanhar a evolução dos treinamentos de seus clientes, e baseado nesse acompanhamento realizar a avaliação física do indivíduo para fazer ajustes nos exercícios prescritos. O programa que o *personal trainer* necessita deve importar os dados de um aplicativo (*app*) instalado no *smartphone* de seus clientes. Esse *app* permite exportar um arquivo texto com os seguintes dados das atividades de corrida ou caminhada realizadas por um indivíduo.

**Exercício:** Caminhada

Arquivo 1

----- Usuário -----

Nome: Bruno Silva

Sexo: masculino

Altura: 1,70 m

Peso: 76,0 Kg

Data de nascimento: 22/04/1980

E-mail: bruno.silva@gmail.com

----- Detalhes do exercício -----

Data: 20/05/2018

Tempo: 10:30 - 11:32

Duração: 01:01:39

Distância: 6,75 Km

Calorias perdidas: 375 Kcal

Passos: 8.298

Velocidade média: 6,5 Km/h

Velocidade máxima: 9,0 Km/h

Ritmo médio: 09'08" /Km

Ritmo máximo: 06'40" /Km

Menor elevação: 1.044 m

Maior elevação: 1.154 m

----- Ritmo -----

1 Km: 09'15"

2 Km: 09'00"

3 Km: 08'54"


4 Km: 08'56"


5 Km: 09'06"


6 Km: 09'09"


6,75 Km: 09'45"


DETALHES DO EXERCÍCIO



**01:01:39**  
Duração exercício



**6,75 km**  
Distância


**375 kcal**  
Calorias perdidas


**8298 passos**  
Passos


**6,5 km/h**  
Velocidade média


**09'08" /km**  
Ritmo médio


**1044 m**  
Menor elevação

**9,0 km/h**  
Velocidade máx.

**06'40" /km**  
Ritmo máximo

**1154 m**  
Maior elevação

DIVISÕES

km	Ritmo(/km)
1	09'15"
2	09'00"
3	08'54"
4	08'56"
5	09'06"
6	09'09"
6,75	09'45"

Média 09'08"

Para compartilhar esses dados coletados pelo *app* durante a realização do exercício, o usuário deve gerar um arquivo texto com os dados supracitados usando a função exportar do *app*. O arquivo texto possui o formato exibido acima.

É possível que o *app* monitore vários tipos diferentes de atividade física. Por exemplo, se o indivíduo jogar basquete e usar o *app* para monitorar o seu desempenho, ao exportar o arquivo os dados que não forem aferidos não farão parte do arquivo texto, como Velocidade média, Velocidade máxima, Ritmo médio, Ritmo máximo, Menor elevação e Maior elevação. Sendo assim, o arquivo teria somente os dados abaixo:

Exercício: Basquete	Arquivo 2
----- Usuário -----	
Nome: Bruno Silva	
Sexo: masculino	
Altura: 1,70 m	
Peso: 76,0 Kg	
Data de nascimento: 22/04/1980	
E-mail: bruno.silva@gmail.com	
----- Detalhes do exercício -----	
Data: 20/05/2018	
Tempo: 15:00 - 15:30	
Duração: 00:30:00	
Distância: 4,54 Km	
Calorias perdidas: 580 Kcal	
Passos: 4.152	

Pra cada tipo de atividade física que o usuário do *app* deseja compartilhar com o *personal trainer* ele deve gerar um arquivo texto. É um arquivo por atividade física realizada. Arquivos gerados para uma determinada atividade física que possuem mais dados detalhando a avaliação do exercício, por exemplo corrida, possuem a estrutura idêntica a do Arquivo 1. E atividades físicas que possuem menos dados detalhando a avaliação do exercício, por exemplo dança, possuem a estrutura idêntica a do Arquivo 2. Em fim, independente ao tipo de esporte praticado pelo usuário do *app* o arquivo texto gerado será sempre igual ao formato do Arquivo 1 ou do Arquivo 2.

O programa Java GUI a ser desenvolvido para o *personal trainer* deve fornecer os seguintes recursos:

1. Importação do arquivo texto gerado pelo *app* e armazenamento desses dados em uma base de dados mantendo os campos e-mail, data e tempo como campos de chave primária. Isso significa que um usuário muito ativo do *app* poderá realizar no mesmo dia mais de uma atividade física (corrida e/ou caminhada).
2. Pesquisa pelo nome do cliente do *personal trainer* para exibir os dados do usuário e os detalhes do exercício, incluindo o ritmo em cada quilômetro. Se no mesmo dia houve mais de um registro de atividade física, os dados a serem exibidos devem ser agrupados pelo campo tempo.
3. Exibe um relatório que mostre para cada cliente do *personal trainer* os valores abaixo e a data em que eles ocorreram.
  - a) a maior duração de um exercício;
  - b) a maior distância percorrida;
  - c) o maior número de calorias perdidas;
  - d) o maior número de passos dados;
  - e) a velocidade mais rápida (máxima).

4. Exibir um gráfico de colunas que permita ao *personal trainer* analisar o desempenho de cada cliente para os dados abaixo em um período (em dias) especificado por ele. Exibir um gráfico de colunas para cada tipo de atividade física realizada pelo indivíduo e para cada item abaixo:

- a) a duração de um exercício;
- b) a distância percorrida;
- c) o número de calorias perdidas;
- d) o número de passos dados;
- e) a velocidade média;
- f) o ritmo médio.

5. Exibir um gráfico de linhas que permita ao *personal trainer* analisar o desempenho de cada cliente para os dados abaixo em um período (em dias) especificado por ele. Exibir um gráfico de linhas para cada tipo de atividade física realizada pelo indivíduo e para cada item abaixo:

- a) a distância percorrida;
- b) o número de calorias perdidas;
- c) o número de passos dados.

6. Exibir um gráfico de colunas, a exemplo da figura ao lado, que mostre: 1. o número total de passos; 2. a distância média; 3. a distância total; 4. a média de calorias perdidas e; 5. o total de calorias perdidas; em um período (em dias) especificado pelo *personal trainer* para um determinado cliente. Considerar nesses cálculos os dados obtidos com todos os tipos de atividades físicas realizadas pelo cliente no período. Exibir um gráfico para cada item supracitado.

7. Geração de um arquivo PDF com os resultados obtidos dos itens 3, 4 e 5 explicados acima para cada cliente especificado pelo *personal trainer*.



## - Critérios de avaliação

1. O trabalho será avaliado considerando:

- 1.1 - validação dos dados fornecidos pelo usuário via expressões regulares;
- 1.2 - tratamento dos cálculos que possam abortar a execução do programa via tratamento de exceções;
- 1.3 - a lógica empregada na solução do problema;
- 1.4 - o funcionamento do programa;
- 1.5 - a usabilidade<sup>1</sup> do programa e da interface gráfica com o usuário;
- 1.6 - o conhecimento da linguagem de programação;
- 1.7 - o uso do paradigma de orientação a objetos e do princípio do privilégio mínimo;
- 1.8 - código fonte Java sem erros e sem advertências (*warnings*) do compilador;
- 1.9 - código fonte Java legível, indentado, organizado e comentado;

<sup>1</sup> 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.

1.10 - documentação do sistema em HTML gerada através da ferramenta *javadoc*.

2. O desenvolvimento do trabalho é individual.
3. A manutenção dos dados e seus relacionamentos deve ser feita usando os recursos de processamento de arquivo de acesso aleatório do Java I/O 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 construídos usando a biblioteca *JFreeChart*, disponível em <http://www.jfree.org/jfreechart>.
6. A geração dos relatórios (incluindo dados e gráficos) deve ser feita em formato PDF e usando a API *iText*, disponível em <https://sourceforge.net/projects/itext>.

### - 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*.

☞ **Atenção:** Se for usado banco de dados deve ser entregue o arquivo de *backup*, o script de criação, o usuário e a senha do banco de dados. É responsabilidade do aluno se certificar do correto funcionamento dos arquivos de *backup* e *script*.

Compacte todos os artefatos criando um arquivo 7z com o seu nome e sobrenome e envie para o e-mail [marlon.silva@ifsudestemg.edu.br](mailto:marlon.silva@ifsudestemg.edu.br).

### - Data e forma de apresentação

Terça-feira, 19 de junho de 2018.

Entrevista sobre o desenvolvimento e o funcionamento do sistema.

### - Valor do trabalho

25,0 (vinte e cinco) pontos.

Prof. Márlon Oliveira da Silva  
[marlon.silva@ifsudestemg.edu.br](mailto:marlon.silva@ifsudestemg.edu.br)