

Programação Orientada aos Objetos

Gabriel Antunes a 101101, Guilherme Pinho a105533

12 de maio de 2024

Grupo 33



Gabriel Antunes, Guilherme Pinho

1 Introdução

O presente relatório tem como base relatar o processo de desenvolvimento de uma aplicação que gera planos de treino para um determinado utilizador. Esta aplicação visa não apenas registrar as atividades realizadas, mas também proporcionar uma análise detalhada do desempenho de cada utilizador, oferecendo indicadores relevantes para o acompanhamento de seu progresso. Além disso, a funcionalidade avançada de geração de planos de treino personalizados ou automáticos, com base nos objetivos individuais e na estimativa de calorias gastas, representa um diferencial significativo.

Este projeto foi realizado no âmbito da unidade curricular Programação orientada aos Objetos, sendo portanto desenvolvido na linguagem JAVA, de forma que nos foi possível tirar proveito da abstração, da hierarquia de classes, do encapsulamento e do polimorfismo.

Neste relatório, serão detalhadas as características e funcionalidades essenciais desta aplicação, bem como os desafios e considerações inerentes ao seu desenvolvimento e implementação.

2 Classes

2.1 Main

Esta classe tem como sua responsabilidade apenas inicializar o programa.

2.2 Menu

```
private String userCode;  
private Utilizadores utilizadores;  
private ListaAtividades lAtividades;  
private PlanosTreino planosTreino;
```

Esta é uma classe abstrata responsável por definir métodos abstratos utilizados nas subClasses MenuUser e MenuAdmin, que vão ser especificadas mais a frente.

2.2.1 MenuAdmin

Esta é uma subClasse da classe Menu, tendo os métodos abstratos definidos na classe Menu aqui desenvolvidos, todos os menus e submenus que são vistos pelo utilizador, tal como o método login que retorna o utilizador que está a utilizar a app no momento e o método registrarAtividade que é responsável por registrar atividades novas para o utilizador mais tarde usar nos seus planos de treino.

2.2.2 MenuAdmin

Esta é uma subClasse da classe Menu, tendo os métodos abstratos definidos na classe Menu aqui desenvolvidos, todos os menus e submenus que são vistos pelo utilizador, tal como o método login que retorna o utilizador que está a utilizar a app no momento e o método registrarAtividade que é responsável por registrar atividades novas que o utilizador decidiu praticar fora do seu plano de treino.

2.3 Utilizador

```
private String codigoUtilizador;  
private String nome;  
private String morada;  
private String email;  
private int freqCardiacaMedia;  
private int altura; //altura centimetros  
private double peso; //peso em kg  
private Dificuldades nivel;  
private String sexo;
```

Esta classe assume o papel de criação de novos utilizadores.

2.3.1 Utilizadores

```
private ArrayList<Utilizador> utilizadores;
```

Esta classe é responsável pelo armazenamento de utilizadores

2.4 Atividades

```
private String nomeAtividade;  
private LocalDate data;  
private double duracao;  
private TipoAtividade tipo;
```

Esta é uma classe abstrata responsável por definir o que é uma atividade básica e quais os seus requisitos. Tendo 3 métodos abstratos que serão desenvolvidos nas suas SubClasses.

2.4.1 Altimetria

```
private int alturaMax;  
private int alturaMedia;  
private double distancia;
```

Esta é uma subclasse da classe Atividades, tendo os métodos abstratos definidos na classe Atividades aqui desenvolvidos, o método `calculaDeCalorias`, que calcula as calorias perdidas num exercício do tipo Altimetria, o método `geraVariante` que tem como auxiliar `calculaVariante`, em que a sua função é gerar valores random para as variáveis de instância.

2.4.2 Distancia

```
private double distancia;
```

Esta é uma subclasse da classe Atividades, tendo os métodos abstratos definidos na classe Atividades aqui desenvolvidos, o método `calculaDeCalorias`, que calcula as calorias perdidas num exercício do tipo Distancia, o método `geraVariante` que tem como auxiliar `calculaVariante`, em que a sua função é gerar valores random para as variáveis de instância.

2.4.3 Repeticoes

```
private int repeticoes;
```

Esta é uma sucClasse da classe Atividades, tendo os metodos abastratos definidos na classe Atividades aqui desenvolvidos, o método calculoDeCalorias, que calcula as calorias perdidas num exercicio do tipo Repeticoes, o método geraVariante que tem como auxiliar calculoVariante, em que a sua função é gerar valores random para as variaveis de instância.

2.4.4 Altimetria

```
private int repeticoes;
```

```
private double pesoHaltere;
```

Esta é uma sucClasse da classe Atividades, tendo os metodos abastratos definidos na classe Atividades aqui desenvolvidos, o método calculoDeCalorias, que calcula as calorias perdidas num exercicio do tipo RepeticoesPeso, o método geraVariante que tem como auxiliar calculoVariante, em que a sua função é gerar valores random para as variaveis de instância.

2.5 Lista Atividades

```
private ArrayList<Atividades> lAtividades;
```

Esta classe tem como seu fundamento armazenar todas as atividades num ArraList, tal como outros métodos para o calculo para a realização de todos os rankings.

2.5.1 PlanosSemanal

```
private LinkedHashMap<String,ListaAtividades> planosSemanal;
```

Esta classe é responsavel por armazenar num dicionário a lista de atividades tendo como key os dias da semana, tem 8 espaços para o caso do utilizador optar por realizar uma atividade extra, seja possivel armazena-la no plano semanal.

2.5.2 PlanosTreino

```
private HashMap<String,PlanoSemanal> planosTreino;
```

Esta classe é responsável por armazenar num dicionário todos os planos semanais, que são definidos na classe PlanosSemanal, tendo como key o user Code do utilizador.

3 Diagrama de Classes

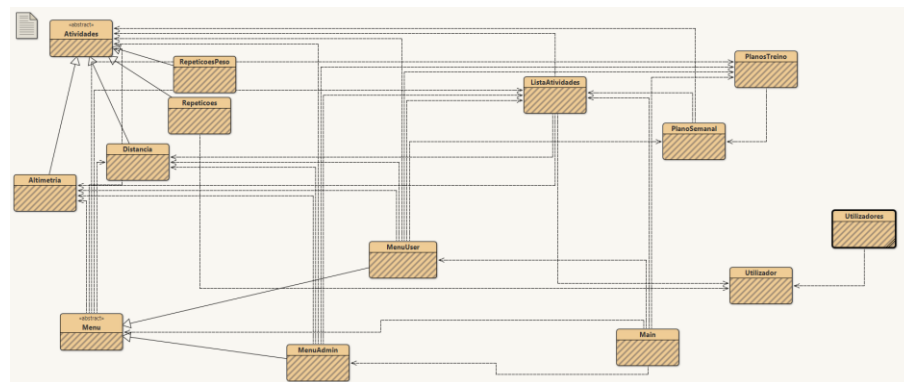


Diagrama de classes. Esta figura também se encontra em anexo para maior detalhe

4 Conclusão

Acreditamos que ao desenvolver este projeto respeitamos todas as regras relativas a Programação Orientada aos Objetos. Durante o processo de desenvolvimento, enfrentamos desafios que nos exigiram criatividade e dedicação para encontrar soluções eficientes. A análise cuidadosa dos requisitos dos usuários e a escolha criteriosa das tecnologias foram fundamentais para o sucesso do projeto. Além disso, a realização de testes rigorosos ao longo do desenvolvimento garantiram a qualidade e a usabilidade da aplicação final. O desenvolvimento deste projeto ajudou a solidificar os conhecimentos adquiridos nas aulas práticas e teóricas, tendo sido importante para ter uma noção das impleções do desenvolvimento em ambiente JAVA.