

Programação Orientada aos Objetos

Projeto Prático

LCC

Gonçalo Gonçalves Barroso

A102931



José Pedro Mendes Castro

A100070



Rafaela Antunes Pereira

A102527



Introdução

Este projeto consistiu na realização de uma aplicação, na linguagem de programação Java, destinada à gestão de atividades e de planos de treino, de praticantes de atividades físicas, tendo esta como principal objetivo proporcionar ao utilizador a possibilidade de registar e monitorizar as suas atividades físicas.

Classes

Acesso

```
private String senha;  
private String UserName;  
private String Nome;  
private String Morada;  
private String Mail;  
private double peso;  
private int frequenciaCardiaca;  
private TiposUtilizador tipodeutilizador;
```

A classe Acesso é responsável pela gestão do acesso dos utilizadores ao programa, esta armazena e altera as informações.

TiposUtilizador

```
Amador(1.1),  
Profissional(1.4),  
Ocasional(1),  
Outro(0);  
private double fator;
```

TiposUtilizador é uma enumeração que define os diferentes tipos de utilizadores e os respetivos fatores multiplicativos para o cálculo de calorias gastas.

Atividade

```
private String nomeAtividade;  
  
private double calorias;  
  
private int duracao;  
  
private boolean realizada;
```

Atividade é uma classe abstrata que é utilizada como uma estrutura que abrange todos os diferentes tipos de atividades que se encontram no programa, esta classe permite que seja realizado um acréscimo de componentes que sejam específicas de tipos de atividades.

Series

```
private List<Integer> repeticoes;  
  
private int series;  
  
private int tempoDescanso;
```

Series é uma das classes que estende a classe abstrata *Atividade*, na qual adiciona atributos específicos para a atividade que esta classe representa.

SeriesPesos

```
private double peso;
```

SeriesPesos é igualmente uma das classes que estende a classe abstrata *Atividade*, que adiciona os atributos específicos para as atividades que são do tipo *SeriesPesos*.

Trilha

```
private double distancia;  
  
private int altura;
```

A classe *Trilha*, estende a classe abstrata *Atividade*, na qual adiciona os atributos específicos para as atividades que são do tipo *Trilha*.

Resistencia

```
private double distancia;  
  
private double velocidade;
```

A classe *Resistencia*, estende a classe abstrata *Atividade*, que tem como função adicionar os atributos específicos para as atividades que são do tipo *Resistencia*.

ExerciciosMet

```
CorridaModerada(9.5),  
    CorridaRápida(11),  
_ CorridaIntensa(12.8),  
_ TrilhaPlana(2.5),  
_ TrilhaModerada(4),  
_ TrilhaÍngreme(6);  
  
private double met;
```

O *ExerciciosMet*, corresponde a uma enumeração que define os valores de MET para os diferentes tipos de exercícios, sendo assim mais fácil aceder aos valores de MET e consequentemente facilitar no cálculo de calorias queimadas.

PlanoTreino

```
private String nome;  
  
private String diaSemana;  
  
private List<Atividade> atividades;
```

A classe *PlanoTreino*, corresponde a representação de um plano de treino, que consiste numa lista de atividades que devem ser executadas num determinado dia da semana.

PlanosDeTreino

```
private HashMap<String, List<PlanoTreino>> planosDeTreino;
```

PlanosDeTreino é uma classe responsável por gerenciar os planos de treino dos utilizadores, a mesma utiliza uma HashMap para armazenar os planos de treinos, respetivamente com os utilizadores associados a estes.

PlanoRealizado

```
private String nomePlano;  
private double calorias;  
private LocalDate data;  
private int freqCardiacaMedia;
```

PlanoRealizado, é uma classe que representa um plano de treino que foi realizado por um utilizador, contendo o nome do plano, as calorias que foram queimadas, a data na qual este plano foi realizado e a frequência cardíaca do utilizador enquanto realizava o plano de treino.

PlanosRealizados

```
private HashMap<String, List<PlanoRealizado>> planosRealizados;
```

A classe *PlanosRealizados*, utiliza uma HashMap para armazenar uma lista de planos de treinos que foram executados, juntamente com o nome do utilizador que os realizou.

Fitness

```
private HashMap<String, Acesso> utilizadores;  
  
private Acesso utilizador;  
  
private PlanosDeTreino planosDeTreino;  
  
private Atividades atividades;  
  
private LocalDate data;  
  
private PlanosRealizados planosRealizados;
```

A classe *Fitness*, responsável por representar o estado atual do programa, tem como função de interface entre o utilizador e o programa, fazendo com que sejam fáceis a interação e o gerenciamento de dados.

TextoInterface

```
private Fitness model;  
  
private Scanner scanner;
```

TextoInterface é responsável por fornecer uma interface de linha de comando para ser possível a interação com o fitness, fornecendo assim várias funcionalidades.

AppNewGen

Corresponde a uma classe que cria uma instância da interface de texto do programa e a inicia.

Menu

A classe *Menu* é usada para criar planos interativos no programa

Descrição/Ilustração

```
---- App Fitness ----  
1 - Login  
2 - Registrar  
3 - Sair  
Opção: |
```

```
---- Menu Inicial ----  
1 - Atividades  
2 - Planos de Treino  
3 - Defenições  
4 - Tabelas  
5 - Sair  
Opção: |
```

```
---- Atividades ----  
1 - Adicionar Atividade  
2 - Remover Atividade  
3 - Listar Atividades  
4 - Voltar ao Menu Inicial  
Opção: |
```

```
---- Adicionar Atividade ----  
1 - Resistencia  
2 - Trilha  
3 - Series  
4 - Series com pesos  
5 - Voltar ao Menu Atividades  
Opção: |
```

```
---- Planos de Treino ----  
1 - Adicionar Plano de Treino  
2 - Remover Plano de Treino  
3 - Listar Planos de Treino e Atividades  
4 - Marcar Plano de Treino como realizado  
5 - Listar Planos de Treino Realizados  
6 - Voltar ao Menu Inicial  
Opção: |
```

---- Definições ----

1 - Ver Data

2 - Mudar Data

3 - Voltar ao Menu Inicial

Opção: |

---- Tabelas ----

1 - Tabela Calorias

2 - Tabela Utilizador Com Mais Atividades Realizadas

3 - Tabela Atividade Mais Realizada

4 - Alterar Intervalo de Tempo

5 - Voltar ao Menu Inicial

Opção: |