

# Projecto prático de Programação Orientada aos Objectos

LEI + LCC

edição 23/24

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	As <i>Actividades</i> . . . . .	2
1.2	Os <i>Utilizadores</i> . . . . .	2
1.3	O <i>Plano de Treino</i> . . . . .	3
<b>2</b>	<b>O Funcionamento do programa</b>	<b>3</b>
<b>3</b>	<b>Requisitos</b>	<b>3</b>
3.1	Requisitos base de gestão das entidades . . . . .	3
3.2	Efectuar estatísticas sobre o estado do programa . . . . .	4
3.3	Criar a noção de actividade <i>Hard</i> . . . . .	4
3.4	Gerar um plano de treino de acordo com objectivos . . . . .	4
<b>4</b>	<b>Relatório</b>	<b>5</b>
<b>5</b>	<b>Salvaguarda do estado da aplicação</b>	<b>5</b>
<b>6</b>	<b>Cronograma</b>	<b>5</b>

## 1 Introdução

Considere que se pretende criar uma aplicação que faça a gestão das actividades e planos de treino, de praticantes de actividades físicas. A aplicação deverá considerar a existência de diversas actividades que podem ser praticadas (exemplo: corrida, bicicleta, pilates, levantamento de pesos, etc.), por diversos tipos de atletas (ex: profissionais, amadores, praticantes ocasionais, etc.). Deverá também ser possível registar um plano de treinos para determinado utilizador indicando quais são as actividades a realizar.

O objectivo da aplicação será o de permitir para cada utilizador registar a realização das actividades, permitindo depois obter uma série de indicadores relativos ao treino de cada um. Um segundo objectivo, mais elaborado e complexo, consistirá na geração de um plano de treino para um determinado utilizador de acordo com os objectivos deste em termos de tipo de exercício e de quantidade esperada de calorias gastas.

A aplicação deverá permitir registar que uma determinada actividade foi realizada por um determinado utilizador, associando a data e hora da realização, e será possível obter posteriormente informação relativa ao número de calorias consumidas num determinado período, ao número de kms percorridos num período, ao número de metros de altimetria acumulados num determinado período, entre outra informação que se julgue relevante.

A aplicação deverá ter também a noção de *plano de treino* de um utilizador, que não é mais do que uma colecção de exercícios que efectua numa sessão de treino e a indicação dos dias da semana em que faz esses mesmos exercícios e da recorrência com que os faz (ex: todos os dias, todas as segundas e quintas, etc.).

Um aspecto que é importante ter em linha de conta é que, para que o sistema possa ser simulado o mais possível num cenário de utilização real, será necessário efectuar *saltos no tempo*. Isto é, a aplicação deverá saber em que data é que se encontra e deverá estar explícita uma funcionalidade que permitirá avançar para uma data no futuro. O avançar do tempo (simulado por questões práticas e de redução de complexidade) deverá originar todas as acções pendentes à realização das actividades que estejam programadas.

## 1.1 As Actividades

A aplicação deverá possuir algumas actividades mais conhecidas e deverá também estar preparada para que seja fácil (do ponto de vista de escrita de código) acrescentar novas actividades sem grande alteração no código das classes existentes.

Numa primeira fase a aplicação deverá possuir quatro tipos de actividades: as que implicam distância e altimetria (ex: corrida na estrada, trail no monte, bicicleta de estrada e bicicleta de montanha), as que implicam apenas distância (ex: remo, corrida na pista de atletismo, patinagem), as que implicam fazer uma série de repetições (ex: abdominais, alongamentos, flexões) ou as que implicam uma série de repetições mas com pesos (ex: levantamento de pesos, extensão de pernas).

Além das características próprias de cada uma das actividades deverá ser registado para todas a informação sobre o tempo dispendido a efectua-la.

O cálculo do consumo de calorias dispendido deverá ser feita em função de cada actividade. Deverá ser codificada uma fórmula (à escolha de cada grupo) que determine as calorias gastas em cada actividade, sendo que essa fórmula deverá aceitar a informação do utilizador que a efectuou. Para o cálculo do consumo de calorias a frequência cardíaca média durante a actividade deverá ser registada e o cálculo deverá entrar em conta com o utilizador que efectuou a actividade. O tipo de utilizador (atleta profissional, amador, etc.) deverá ser um factor multiplicativo no cálculo das calorias (considera-se que um utilizador mais profissional gasta mais calorias num exercício do que um utilizador mais amador). A fórmula de cálculo deverá ser sugerida por cada grupo <sup>1</sup> e deverá ser parametrizada pelo utilizador que a efectuou (isto é, o utilizador deverá ser passado como parâmetro na fórmula de cálculo).

Dentro dos quatro tipos de actividades deverá ser considerado que existem algumas dessas actividades que são consideradas como sendo *Hard*. A título de exemplo nas actividades de distância e altimetria o trail no monte e as actividades de bicicleta de montanha são actividades mais difíceis (isto é, do tipo *Hard*).

## 1.2 Os Utilizadores

A aplicação deverá prever a existência de diferentes tipos de utilizador, uns mais profissionais e outros mais amadores. Sugere-se que tenham pelo menos três tipos de utilizadores, os *profissionais*, os *amadores* e os *praticantes ocasionais*. Tal como para as actividades deverá ser relativamente fácil acrescentar novos tipos de utilizadores escrevendo código apenas para as definições do que é o novo tipo de utilizador sem ter de modificar o restante programa.

Para cada utilizador deverá ser registado o seu código de utilizador, o seu nome, morada, email e a informação relativa à sua frequência cardíaca média. Consoante o tipo de utilizador e a informação do utilizador em causa deverá existir um método que determine um factor multiplicativo a utilizar no cálculo das calorias. Os utilizadores profissionais, amadores e praticantes ocasionais deverão ter fórmulas de cálculo diferentes para obter este factor, que depois será utilizado no método de cálculo das calorias das actividades (cf. explicado na secção anterior).

Para cada utilizador deverão ficar registadas as actividades que ele efectuou, quer sejam actividades isoladas ou inseridas dentro de um plano de treino.

---

<sup>1</sup>Sugere-se a pesquisa de informação sobre esta temática.

### 1.3 O Plano de Treino

Um *plano de treinos* é um conjunto de actividades que um utilizador realiza numa determinada altura. O plano de treino possui uma data em que se realiza e uma lista das actividades que devem ser cumpridas. O plano pode também especificar se as actividades são executadas uma única vez ou se são definidas  $n$  iterações das mesmas.

Os planos de treino estão associados aos utilizadores e podem ser construídos iterativamente no programa ou podem ser gerados pela aplicação em função de um determinado critério (ex: plano de treino de corrida ou plano de treino de força com pesos).

## 2 O Funcionamento do programa

Pretende-se que com este programa se possa criar uma espécie de simulação da execução das actividades por parte dos utilizadores e que se possa também obter informação estatística e sobre os recordes existentes para cada actividade, bem como quem os conseguiu).

Como foi atrás referido, terá de ser previsto que se possam efectuar mudanças de data (o que atrás se designou por *saltos no tempo*) e deverá também ser necessário entrar e sair do programa para simular acessos diferentes por parte de diferentes utilizadores. Mas isto ficará ao critério de cada grupo e da complexidade que pretender desenvolver (podem por exemplo considerar que a informação de todos os utilizadores é visível e que se pode efectuar registos para todos os utilizadores).

1. o programa deve ter uma fase em que é possível criar utilizadores, actividades e planos de treino. Uma hipótese é esta informação ser carregada de um ficheiro, dispensando quem utiliza o programa de fazer o setup da informação sempre que tal seja necessário.
2. a componente de simulação deve começar já com uma configuração em que existem utilizadores, actividades e planos de treino.
3. o programa deve pedir ao utilizador que determine o avançar do tempo. Este avançar do tempo faz-se em dias e o programa deve permitir **ir para uma determinada data**. Nessa altura executam-se as actividades previstas e são actualizadas as informações relativas às calorias consumidas bem como se regista os recordes de cada utilizador em cada tipo de actividade.
4. deverá ser possível pedir para criar um plano de treino para um determinado utilizador em função dos seus objectivos, ou seja, tipo de actividades que deseja fazer e recorrência das mesmas (diária,  $n$  vezes por semana, etc.)

## 3 Requisitos

O trabalho proposto tem vários níveis de requisitos, desde os mais básicos até aos mais complexos. Para que os vários níveis de requisitos identificados sejam considerados cumpridos o programa deve respeitar os princípios da programação por objectos, ser robusto no tratamento de erros, bem como gravar, e também carregar, o estado do programa para um ficheiro de objectos. Deverá possuir uma camada de interacção com o utilizador, que se aconselha ser feita por menus em modo texto (atenção que não é solicitada a existência de camada gráfica) e que permita aceder às funcionalidades do programa. Durante a sessão de apresentação será também avaliada a forma como o programa está construído, nomeadamente as classes existentes e como é que estão organizadas numa arquitectura que permita responder ao que é pedido e a utilização das estruturas de dados e a forma como são utilizadas e manipuladas.

### 3.1 Requisitos base de gestão das entidades

Como requisitos base pretende-se ter a capacidade de criar utilizadores, actividades e planos de treino e registar a execução de uma actividade por parte de um utilizador.

Nota máxima: 12 valores

Tenham em atenção que, mesmo que se permitam criar utilizadores e actividades, mas não for possível registar a realização de uma actividade por parte de um utilizador, com o respectivo cálculo de calorias, a nota máxima será de 10 valores.

### 3.2 Efectuar estatísticas sobre o estado do programa

Além do anterior, o programa deverá ser capaz de avançar o tempo, actualizar a informação do sistema e actualizar os recordes para cada tipo de actividade. E ainda permitir saber:

1. qual é o utilizador que mais calorias dispendeu num período ou desde sempre
2. qual o utilizador que mais actividades realizou num período ou desde sempre
3. qual o tipo de actividade mais realizada
4. quantos kms é que um utilizdor realizou num período ou desde sempre
5. quantos metros de altimetria é que um utilizar totalizou num período ou desde sempre
6. qual o plano de treino mais exigente em função do dispêndio de calorias proposto
7. listar as actividades de um utilizador

Nota máxima: 16 valores

### 3.3 Criar a noção de actividade *Hard*

Além dos requisitos anteriores, o programa deverá permitir a existência de actividades que sejam *Hard* por forma a que seja possível acrescentar, escrevendo código, novas actividades deste tipo.

Nota máxima: 18 valores

### 3.4 Gerar um plano de treino de acordo com objectivos

Tendo em conta as características dos utilizadores o programa poderá gerar um plano de treino para um determinado utilizador permitindo que se especifique:

- o tipo de actividades que se pretende ter. Se o utilizador pretender ter actividades que sejam classificadas como *Hard* terá de ser respeitada a condição de não ter no mesmo dia mais nenhuma actividade *hard* e nunca ter dessas actividades em dias consecutivos.
- o número máximo de actividades por dia (que em qualquer situação nunca devem ser mais do que 3) e o número máximo de actividades distintas.
- a recorrência semanal das actividades (exemplo: 3 vezes por semana).
- o consumo calórico mínimo que se pretende ter.

O programa deverá gerar, se tal for possível, um plano de treino para o utilizador e deverá permitir executá-lo para se aquilatar do cumprimento dos objectivos.

Nota máxima: 20 valores

Obviamente que a nota a atribuir, em cada um dos escalões, irá reflectir a estruturação da solução que deverá respeitar as normas da programação orientada aos objectos, cf aulas teóricas, nomeadamente o encapsulamento, a abstracção de implementação e a capacidade de a aplicação evoluir de forma controlada.

Notem que **não serão considerados como merecedores de nota positiva trabalhos que não permitam a criação e visualização da informação relativa às entidades do programa.**

## 4 Relatório

O relatório deve descrever o trabalho realizado para desenvolver a aplicação solicitada. No mínimo, devem ser abordados os seguintes pontos:

- Capa com identificação da Unidade Curricular e do grupo (nome, número e fotografia).
- Descrição da arquitectura de classes utilizada (classes, atributos, etc.) e das decisões que foram tomadas na sua definição. Deverá ser entregue um **Diagrama de Classes** com a arquitectura de classes que suporta o programa desenvolvido.
- Descrição da aplicação desenvolvida (ilustração das funcionalidades).

**Todos estes itens são obrigatórios!**

A cotação do relatório é de 2 valores.

## 5 Salvaguarda do estado da aplicação

O programa deve permitir que em qualquer momento se possa guardar em ficheiro a informação existente em memória sobre a informação relevante das entidades. A gravação deve ser feita de forma a permitir que o estado que foi gravado seja recuperado novamente. Na altura da entrega do projecto deve ser também entregue um estado (guardado em ficheiro) que possa ser carregado durante a apresentação.

## 6 Cronograma

1. Entrega final de código, ficheiro com os dados a carregar durante a apresentação e relatório de projecto (feita por via electrónica no git)  
**Data Entrega:** *11 de maio de 2024*
2. Apresentação do projecto  
**Semana de:** *27 a 29 de maio de 2024 - data acordada com a direcção de curso*