



Universidade do Minho
Escola de Engenharia

JosefinFitnessApp

POO, Programação Orientada a Objetos (2024)

Grupo 01

Nome Completo	Número de Estudante
Alex Sá	a104257
Paulo Ferreira	a96268
Rafael Fernandes	a104271



Braga, Portugal , 10/05/2024

Conteúdo

1	Abstract	2
2	Introdução	1
2.1	Contexto e Descrição do Trabalho	1
2.2	Objetivos	1
3	Arquitetura e Classes	2
3.1	Users	2
3.2	JosefinFitnessApp	2
3.3	Atividades	2
3.4	Planos de Treino	2
3.5	IDManager	3
3.6	Estatísticas	3
3.7	Subclasses	3
3.8	Cli	3
4	Funcionamento do Programa	4
4.1	Inicialização da Aplicação	4
4.2	Modos de Acesso	4
4.3	Users	4
4.4	Atividades	5
4.5	Planos de Treino	5
4.6	Estatísticas	5
5	Objetivos Alcançados	7
5.1	Arquitetura de Classes	7
5.2	Funcionalidades Básicas	7
5.3	Sistema de Estatísticas	7
5.4	Integração de Atividades Hard	7
5.5	Geração de Planos de Treino Personalizados	7
5.6	Passagem do Tempo	7
5.7	Produção de Relatório Detalhado	7
6	Conclusão	8
	Referências	9
A	Appendix A title	10

1 | Abstract

Este relatório apresenta o processo de desenvolvimento e as principais características da JosefinFitnessApp, uma aplicação de software projetada para auxiliar os utilizadores na gestão das suas atividades físicas e planos de treino. A aplicação implementa uma arquitetura robusta orientada a objetos e fornece funcionalidades como gestão de utilizadores, acompanhamento de atividades, planeamento de treinos e análise estatística. Através de uma descrição detalhada da arquitetura, funcionalidades e decisões de implementação, este relatório tem como objetivo fornecer uma visão geral do projeto JosefinFitnessApp.

2 | Introdução

A aplicação desenvolvida, denominada *JosefinFitnessApp*, surge no âmbito da Unidade Curricular de Programação Orientada a Objetos (POO). Este trabalho visa implementar um sistema de gestão de atividades e planos de treino para praticantes de atividades físicas, abrangendo diferentes tipos de utilizadores e as suas necessidades específicas.

2.1 | Contexto e Descrição do Trabalho

Considerando a crescente preocupação com a saúde e o bem-estar, a prática regular de atividades físicas tornou-se uma prioridade para muitas pessoas. Nesse sentido, a *JosefinFitnessApp* propõe-se a oferecer uma solução eficaz para a gestão e acompanhamento do treino, permitindo aos utilizadores registar as suas atividades, acompanhar o seu progresso e alcançar os seus objetivos de forma personalizada.

2.2 | Objetivos

O principal objetivo da *JosefinFitnessApp* é fornecer uma plataforma robusta e intuitiva para a gestão das atividades e planos de treino dos utilizadores. Além disso, os objetivos específicos incluem:

- Criar uma arquitetura de classes bem definida que suporte todas as funcionalidades da aplicação.
- Implementar funcionalidades básicas, como a criação de utilizadores, atividades e planos de treino, e o registo da execução de atividades.
- Desenvolver um sistema de estatísticas que permita aos utilizadores acompanhar o seu progresso e revelar aqueles com os mais altos recordes dentro da aplicação.
- Integrar a noção de atividades "Hard" para oferecer aos utilizadores opções de treino mais desafiadoras.
- Gerar planos de treino personalizados com base nos objetivos e preferências de cada utilizador.
- Produzir um relatório detalhado descrevendo a arquitetura de classes utilizada, funcionalidades implementadas e decisões de design tomadas durante o processo de desenvolvimento.

Através da concretização destes objetivos, espera-se que a *JosefinFitnessApp* possa proporcionar uma experiência completa e satisfatória aos utilizadores, auxiliando-os no alcance dos seus objetivos de saúde e fitness.

3 | Arquitetura e Classes

Para o desenvolvimento desta aplicação, foi adotada uma arquitetura baseada em cinco classes principais, a classe `User`, `Activity`, `TraninigPlan`, `Cli` e `JosefinFitnessApp` sendo a classe `JosefinFitnessApp` a principal responsável por coordenar todas as outras classes, atuando como uma fachada (*facade*). Essa escolha de arquitetura foi feita com base na necessidade de modularidade, encapsulamento e separação de preocupações, visando um código mais organizado e de fácil manutenção.

3.1 | Users

A lógica relacionada aos utilizadores é gerida principalmente pelas classes `User`, `Users` e `UserController`. A classe `User` é responsável por armazenar informações específicas de cada utilizador, como atividades praticadas e planos de treino. Ela fornece métodos para adicionar, remover e atualizar essas informações, garantindo a integridade dos dados e a consistência do sistema.

A classe `Users` atua como um repositório para todos os utilizadores, mantendo referências para eles em uma lista e em mapas, utilizando o identificador único e o nome de utilizador. Isso facilita a busca e acesso rápido às informações dos utilizadores, contribuindo para a eficiência operacional da aplicação.

O `UserController` serve como uma interface de controle entre as classes mencionadas anteriormente e a fachada principal (`JosefinFitnessApp`), garantindo a comunicação e interação adequadas entre os componentes do sistema.

3.2 | JosefinFitnessApp

A classe `JosefinFitnessApp`, funciona como uma fachada, separando o backend do frontend da aplicação. Ela recebe entradas dos utilizadores, sejam administradores ou não, e executa diversas operações, como adicionar atividades, planos de treino e utilizadores, bem como remover e atualizar informações. Além disso, é responsável por fornecer informações solicitadas pelos utilizadores, como estatísticas do programa e listas de atividades. A sua implementação foi projetada para ser flexível e extensível, permitindo a inclusão de novas funcionalidades e adaptações futuras.

3.3 | Atividades

A lógica relacionada às atividades segue uma abordagem semelhante à dos utilizadores, utilizando as classes `Activity`, `Activities` e `ActivityController`. A classe `Activity` armazena informações sobre cada atividade e fornece métodos para manipulá-las, como adicionar, remover e atualizar atividades. Isso permite os utilizadores gerirem as suas atividades de forma eficiente, adaptando-as às suas necessidades individuais.

A classe `Activities` mantém todas as atividades em um repositório, permitindo o acesso eficiente a elas por meio de listas e mapas. Isso garante a organização e estruturação adequadas das informações relacionadas às atividades, facilitando a sua manipulação e utilização pelos utilizadores.

O `ActivityController` atua como um intermediário entre as classes de manipulação de atividades, garantindo a integração harmoniosa desses componentes e a consistência das operações realizadas.

3.4 | Planos de Treino

A classe relacionada aos planos de treino (`TrainingPlan`) difere um pouco das classes de atividades e utilizadores. Ela verifica a viabilidade de um plano de treino antes de adicioná-lo ao utilizador, garantindo que as restrições sejam atendidas. As atividades presentes no plano de treino são as mesmas que estão presentes no `User`, ou seja, estão por agregação. Neste contexto, essa escolha parece-nos fazer sentido devido ao facto de ambas existirem no contexto do utilizador. Tendo em conta que ele é "dono" das duas, o facto de a atividade criada ser referenciada tanto nas lista de atividades do mesmo como no plano de treino não quebra o encapsulamento e torna todo o processo de atualização e controlo mais eficaz e simples.

3.5 | IDManager

Esta classe desempenha um papel fundamental na nossa aplicação, sendo responsável por gerar identificadores únicos para cada uma das entidades. Ela assegura a unicidade e a sequência dos IDs, garantindo que não haja entidades repetidas, mesmo quando a aplicação carrega informações de sessões anteriores.

3.6 | Estatísticas

Para o cálculo de estatísticas e recordes, foi desenvolvido um módulo separado (**Statistics**) visando modularidade e escalabilidade. Esse módulo, quando necessário, calcula estatísticas com base nas informações fornecidas pela **JosefinFitnessApp**, facilitando a geração de estatísticas e recordes. Esse módulo pode ser expandido no futuro para incluir estatísticas de longo prazo, tal como *All Time Records*, utilizando por exemplo o *desing pattern Observer* para atualização contínua dos dados.

3.7 | Subclasses

Para atender aos objetivos propostos e facilitar a escalabilidade da aplicação, foram criadas subclasses de utilizadores e atividades. As subclasses de utilizadores incluem **ProfessionalUser**, **AmateurUser** e **CasualUser**, cada uma com cálculos de calorias específicos. Já as subclasses de atividades foram organizadas em dois níveis de hierarquia, permitindo uma melhor categorização das atividades e suas características individuais.

Essa separação em subclasses garante uma estrutura modular e flexível, permitindo a adição de novas funcionalidades e acomodando possíveis futuras expansões do sistema.

3.8 | Cli

Este módulo é responsável por toda a criação da interface do utilizador, fornecendo todas as opções e menus disponíveis durante o programa. Ele facilita a interação do utilizador com a aplicação, apresentando de forma clara e organizada as funcionalidades disponíveis e permitindo a navegação intuitiva entre as diferentes operações oferecidas pela **JosefinFitnessApp**.

4 | Funcionamento do Programa

Nesta secção, vamos explorar detalhadamente o funcionamento da nossa aplicação e as diferentes funcionalidades que ela oferece.

4.1 | Inicialização da Aplicação

Quando a aplicação é iniciada, um processo de inicialização é acionado. Durante esse processo, todas as informações da última sessão são carregadas para a memória. Isso é feito utilizando uma função que lê um arquivo binário contendo os dados do último *save* realizado pelo utilizador. Essa abordagem permite que os utilizadores possam continuar de onde pararam na última sessão, mantendo a consistência e a continuidade dos dados.

4.2 | Modos de Acesso

Após o carregamento das informações, a aplicação oferece três modos de acesso, cada um com as suas características específicas:

- **Sign Up:** Este modo permite que novos utilizadores se registem na aplicação. Ao escolher esta opção, o utilizador é direcionado para uma página onde pode preencher os seus dados pessoais e criar um novo perfil. Após a conclusão do registo, o utilizador é redirecionado para uma página onde pode adicionar atividades e planos de treino de acordo com o seu perfil.
- **Log In:** Utilizadores existentes podem fazer o login na aplicação selecionando esta opção. Após o login bem sucedido, o utilizador é direcionado para uma página onde pode visualizar e gerir as suas atividades e planos de treino.
- **Modo Admin:** Este modo é reservado para administradores da aplicação e oferece acesso total a todos os recursos e funcionalidades. Os administradores têm a capacidade de realizar operações em todos os utilizadores, atividades e planos de treino da aplicação. Isso inclui adicionar, remover e atualizar informações, garantindo uma gestão abrangente e eficiente do sistema.

Esses diferentes modos de acesso permitem que os utilizadores interajam com a aplicação de maneiras específicas, de acordo com suas necessidades e privilégios de acesso. Essa abordagem garante uma experiência flexível e adaptável para todos os utilizadores, independentemente do seu papel na aplicação.

4.3 | Users

A gestão de utilizadores é uma das funcionalidades essenciais da nossa aplicação. Nesta secção, discutiremos em detalhes as operações disponíveis para os utilizadores normais e para o administrador.

- **Operações para Utilizadores Normais:**
 - Os utilizadores têm a capacidade de adicionar, remover e atualizar as suas informações pessoais na aplicação.
 - Quando um novo utilizador é adicionado, é gerado um ID único para o mesmo. Este ID é utilizado para identificar de forma única cada utilizador na aplicação.
 - Durante o processo de criação de um novo utilizador, são solicitadas informações como nome, nome de utilizador, morada, data de nascimento, entre outras.
 - Além disso, os utilizadores têm a opção de eliminar a sua própria conta, o que os redireciona para a página de login.
- **Operações para o Administrador:**
 - O administrador possui privilégios adicionais em relação à gestão de utilizadores.
 - Pode adicionar e remover utilizadores diretamente utilizando o ID do utilizador.
 - Esta funcionalidade permite ao administrador manter o controlo sobre os utilizadores registados na aplicação, garantindo uma gestão eficiente dos perfis de utilizador.

4.4 | Atividades

A gestão de atividades é outra funcionalidade importante da nossa aplicação, que permite aos utilizadores criar e remover atividades de acordo com as suas preferências e necessidades de treino.

- **Operações para Utilizadores Normais:**

- Dentro da interface do utilizador, é possível criar e remover atividades.
- O utilizador pode selecionar entre uma variedade de atividades predefinidas, tais como remo, ciclismo de estrada, levantamento de pesos, entre outras.
- Após selecionar a atividade desejada, o utilizador é solicitado a fornecer as informações necessárias para criar a atividade, como a duração, a distância, a intensidade, etc.
- Uma vez completado este processo, a atividade é adicionada ao perfil do utilizador, ficando disponível para registo e acompanhamento.

- **Operações para o Administrador:**

- O administrador tem a capacidade de adicionar ou remover atividades em qualquer perfil de utilizador.
- Esta funcionalidade proporciona ao administrador um controlo total sobre as atividades disponíveis na aplicação, permitindo-lhe ajustar dinamicamente as opções de treino para todos os utilizadores.

4.5 | Planos de Treino

A implementação do Plano de Treino foi, sem dúvida, uma das partes mais desafiadoras do desenvolvimento da nossa aplicação. No entanto, uma vez concluída, proporcionou aos utilizadores a capacidade de criar planos de treino totalmente personalizados, adaptados às suas necessidades individuais.

Ao criar um plano de treino personalizado, os utilizadores podem definir várias opções, como a frequência do treino, os dias da semana em que desejam realizar as atividades e quais atividades desejam incluir. No entanto, é importante respeitar algumas restrições, tais como não poder ter mais de três atividades em um plano de treino, não poder ter mais de uma atividade difícil (*hard*) no mesmo dia e não poder ter atividades difíceis em dias consecutivos. Essas restrições foram implementadas para garantir que os planos de treino sejam equilibrados e seguros para os utilizadores.

Além da opção de criar um plano de treino personalizado, os utilizadores também têm a possibilidade de gerar planos de treino automaticamente. Nesse caso, o utilizador precisa apenas responder a algumas perguntas, tais como se deseja incluir atividades difíceis, o número de atividades por dia que quer realizar, em que dia deseja realizar a atividade, se deseja repeti-la em determinados dias da semana e qual é o número mínimo de calorias que deseja queimar com o plano de treino. Adicionalmente, o utilizador pode escolher se deseja um plano de treino equilibrado, focado em força ou em cardio.

Os administradores também têm permissão para adicionar ou remover planos de treino de qualquer utilizador, utilizando o mesmo processo de questionário para definição dos parâmetros do plano de treino. Esta funcionalidade proporciona aos administradores controle total sobre os planos de treino dos utilizadores, permitindo ajustes conforme necessário.

4.6 | Estatísticas

As estatísticas são uma parte fundamental da nossa aplicação, permitindo aos utilizadores e administradores acederem a registos dos utilizadores e a informações interessantes do programa.

- **Utilizador com Maior Quantidade de Calorias Queimadas:** Esta estatística identifica o utilizador que queimou o maior número de calorias em um período específico ou desde o início do registo. Fornece uma visão detalhada do desempenho dos utilizadores em termos de atividade física e gasto calórico.
- **Utilizador com Maior Número de Atividades Completadas:** Esta estatística destaca o utilizador que realizou o maior número de atividades num determinado período ou desde o início do registo. Ajuda a reconhecer os utilizadores mais ativos e dedicados à prática de exercícios.

- **Atividade Mais Comum:** Esta estatística revela qual é a atividade mais praticada pelos utilizadores. Oferece *insights* sobre as preferências e hábitos de exercício dos utilizadores.
- **Distância Percorrida pelo Utilizador:** Esta estatística indica a distância total percorrida por um utilizador num período específico ou desde o início do registo. É útil para avaliar o progresso e o desempenho do utilizador em atividades como corrida, ciclismo, entre outras.
- **Elevação (Altimetria) Realizada pelo Utilizador:** Esta estatística mostra a quantidade total de elevação ou altimetria acumulada por um utilizador num período específico ou desde o início do registo. É relevante para atividades como escalada, caminhadas em terrenos montanhosos, entre outras.
- **Lista de Atividades do Utilizador:** Esta estatística fornece uma lista detalhada de todas as atividades realizadas por um utilizador, permitindo uma visão abrangente do seu histórico de exercícios.
- **Recordes de Todos os Tempos:** Esta seção destaca os utilizadores que alcançaram os recordes de todos os tempos em termos de calorias queimadas e atividades completadas. Oferece uma perspetiva sobre os utilizadores mais bem-sucedidos e dedicados desde o início do registo na aplicação.

Essas estatísticas fornecem uma visão abrangente do desempenho e do comportamento dos utilizadores na aplicação, permitindo uma análise detalhada e a tomada de decisões informadas.

5 | Objetivos Alcançados

Nesta seção, vamos avaliar se alcançamos todos os objetivos descritos na introdução e explicar como cada um foi alcançado.

5.1 | Arquitetura de Classes

O objetivo de estabelecer uma arquitetura de classes bem definida foi alcançado por meio da criação das classes principais `User`, `Activity`, `TrainingPlan`, `Statistics` e `JosefinFitnessApp`. Essas classes foram cuidadosamente projetadas e organizadas para suportar todas as funcionalidades da aplicação de forma eficiente e escalável. Isto provou-se útil e correto ao longo do desenvolvimento, tendo sido feitas poucas alterações as mesmas, sendo que a maior parte das alterações vieram com o aumento da quantidade de classes.

5.2 | Funcionalidades Básicas

As funcionalidades básicas da aplicação, como a criação de utilizadores, atividades e planos de treino, bem como o registo da execução de atividades, foram implementadas com sucesso. Os utilizadores podem facilmente criar e gerir as suas contas, adicionar atividades ao seu perfil e acompanhar o seu progresso ao longo do tempo.

5.3 | Sistema de Estatísticas

O sistema de estatísticas foi desenvolvido para permitir que os utilizadores acompanhem o seu progresso e identifiquem os recordes da aplicação. Isso foi alcançado por meio da classe `Statistics`, que calcula e apresenta várias estatísticas relevantes, como utilizador com mais calorias queimadas, utilizador com mais atividades concluídas e atividade mais comum.

5.4 | Integração de Atividades Hard

A noção de atividades "Hard" foi integrada com sucesso na aplicação, proporcionando aos utilizadores opções de treino mais desafiadoras. Isso foi feito por meio da definição de atividades "Hard" e da implementação de restrições para garantir que essas atividades sejam usadas de forma equilibrada e segura.

5.5 | Geração de Planos de Treino Personalizados

Os planos de treino personalizados foram gerados com base nos objetivos e preferências de cada utilizador. Isso foi alcançado por meio da classe `TrainingPlan`, que permite aos utilizadores criar planos de treino sob medida, definindo atividades, frequência de treino e metas calóricas.

5.6 | Passagem do Tempo

A simulação da passagem do tempo foi adequadamente implementada na aplicação. Foi criada uma variável para controlar o tempo, e essa variável é utilizada para atualizar as estatísticas periodicamente. Essa funcionalidade foi testada e demonstrou ser eficaz na atualização das estatísticas ao longo do tempo, fornecendo aos utilizadores uma visão precisa e atualizada do seu progresso na aplicação.

5.7 | Produção de Relatório Detalhado

Por fim, um relatório detalhado foi produzido, descrevendo a arquitetura de classes utilizada, as funcionalidades implementadas e as decisões de design tomadas durante o desenvolvimento da aplicação. Isso proporciona uma visão abrangente do funcionamento da *JosefinFitnessApp* e ajuda a entender como cada objetivo foi alcançado.

Portanto, podemos concluir que todos os objetivos descritos na introdução foram alcançados com sucesso, resultando em uma aplicação completa e funcional que atende às necessidades dos utilizadores.

6 | Conclusão

Em suma, o desenvolvimento da *JosefinFitnessApp* foi um processo desafiador, mas gratificante. Ao longo deste projeto, conseguimos atingir os objetivos estabelecidos com sucesso, proporcionando uma plataforma funcional e robusta para a gestão de atividades e planos de treino.

A arquitetura de classes bem definida permitiu a implementação de todas as funcionalidades da aplicação de forma organizada e eficiente. A capacidade de criar utilizadores, atividades e planos de treino, bem como registar a execução de atividades, foi implementada de acordo com as especificações.

Além disso, o sistema de estatísticas foi desenvolvido para permitir aos utilizadores acompanhar o seu progresso e identificar os recordes alcançados. A integração da noção de atividades "Hard" proporcionou aos utilizadores opções de treino mais desafiadoras, promovendo assim uma experiência de utilização diversificada e estimulante.

A geração de planos de treino personalizados baseados nos objetivos e preferências de cada utilizador foi um dos pontos altos do projeto, destacando-se pela sua capacidade de fornecer orientação e apoio individualizado aos utilizadores.

Em conclusão, a *JosefinFitnessApp* representa não apenas o sucesso na concretização dos objetivos deste projeto, mas também o potencial para fornecer uma experiência completa e satisfatória aos utilizadores, ajudando-os a alcançar os seus objetivos de saúde e fitness de forma eficaz e personalizada.



Referências



A | Appendix A title