



Universidade do Minho

PROGRAMAÇÃO ORIENTADA A OBJETOS

Trabalho Prático

Luís Carlos Fragoso Figueiredo, A100549

Afonso Santos Pimenta Silva, A100648

João Pedro Malheiro da Costa, A84701

Sumário

1 Introduction	5
2 Arquitetura	6
2.1 Utilizadores	6
2.2 Atividades	7
3 Funcionalidades implementadas	9
4 Conclusão	10

Lista de figuras

Figura 1: Diagrama de Classes dos Users	6
Figura 2: Diagrama de Classes das Atividades	8

Lista de tabelas

Capítulo 1

Introduction

Este relatório tem como intuito abordar o projeto da UC Programação Orientada a Objetos do ano letivo 2023/2024. O projeto proposto pela UC apresentou vários desafios que nos levaram a encontrar soluções criativas para os resolver.

Capítulo 2

Arquitetura

A arquitetura do nosso projeto baseia-se no conceito MVC (Model, View and Controller), esta arquitetura tem em conta o princípio de separação de tarefas, onde cada classe deve fazer apenas uma coisa.

No Model temos as bases para os nossos dados na aplicação. É aqui onde guardamos as informações sobre os nossos utilizadores, atividades e planos de treino.

2.1 Utilizadores

Existem três tipos de utilizador:

- Utilizadores casuais
- Utilizadores amadores
- Utilizadores profissionais

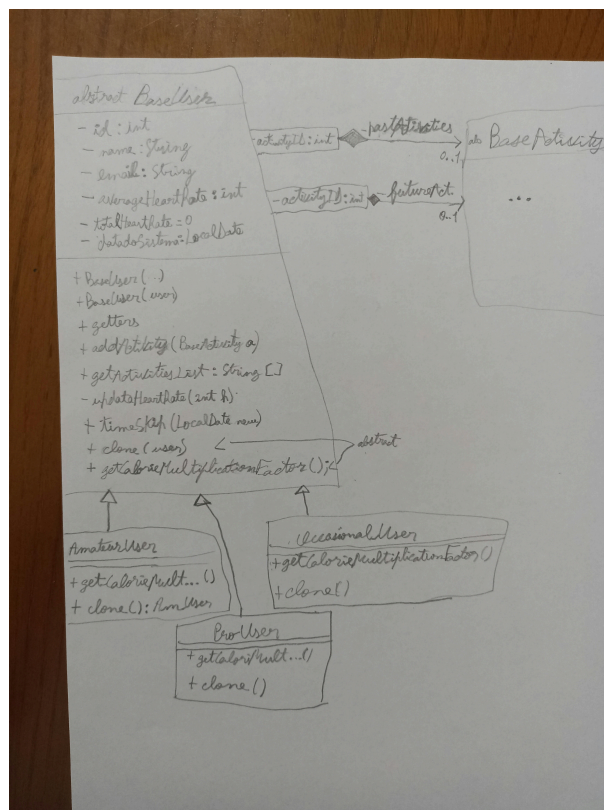


Figura 1: Diagrama de Classes dos Users

Todos eles derivam da classe abstrata `BaseUser` que guarda os dados comuns a todos os tipos de utilizador.

2.2 Atividades

Existem quatro tipos de atividades:

- Atividades de distância
- Atividades de altimetria
- Atividades com repetições
- Atividades com pesos

Estas são representadas por 4 interfaces: `IDistance`, `IAltimetria`, `IRepetitions` e `IWeight`, respetivamente. Escolhemos esta representação porque os requisitos de uma nova atividade desportiva a implementar podem exigir outro tipo de Atividade, que seria solucionado com a adição de outra interface, permitindo assim uma expansão controlada do projeto.

Para representar uma Atividade decidimos criar uma classe abstrata chamada `BaseActivity`, que por sua vez é estendida por atividades em concreto que vão implementar uma ou várias das interfaces anteriormente mencionadas. Optamos por esta estratégia de modo a permitir a adição posterior de mais classes Atividade de forma fácil e compreensível.

Neste momento temos apenas duas atividades implementadas, sendo elas a Corrida e o Supino.

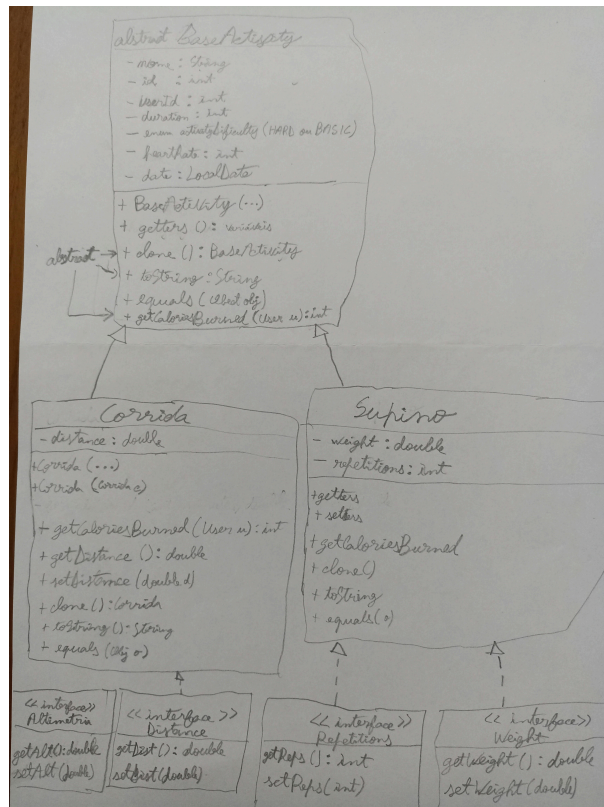


Figura 2: Diagrama de Classes das Atividades

Capítulo 3

Funcionalidades implementadas

- Guardar Estado do Programa a qualquer altura
- Carregar Estado do Programa no menu inicial
- Registar novo Utilizador
- Avançar no Tempo
- Iniciar Sessão (num Utilizador já existente)
- Consultar Atividades (realizadas e planeadas) de um Utilizador
- Registar Nova Atividade

Capítulo 4

Conclusão

Apesar de considerarmos ter feito o mais importante do projeto, reconhecemos que nos falta cumprir alguns requisitos.