U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Gestão de Campeonatos Polidesportivos

(Relatório Final)

06/11/2015

Turma 1, Grupo A:

João Manuel Estrada Pereira Gouveia up201303988@fe.up.pt
Rui Daniel Cruz e Silva da Costa Gonçalves ei12185@fe.up.pt
João Pedro Bernardes Mendonça up201304605@fe.up.pt

Índice

Descrição do tema do trabalho

No âmbito da unidade curricular de Algoritmos e Estruturas de Dados, do 2ºano do Mestrado Integrado de Engenharia Informática e Computação, foi estabelecido como tema para o projeto a desenvolver ao longo do semestre o tema 7 (Campeonatos Polidesportivos). Este tema consiste na criação de um programa, utilizando a linguagem de programação C++, que permita a gestão de um campeonato com vários desportos e várias modalidades, bem como todas as equipas que estejam inscritas nesse campeonato e os seu atletas.

Este programa deve possibilitar a gestão de tudo que um campeonato real tem de gerir como infraestruturas, funcionários do campeonato, as provas que se têm de realizar, os atletas que realizaram essas provas e em que posição ficaram. Para isso, o utilizador terá disponível um conjunto de operações, como criar, apagar, atualizar e ler a informação já existente, guardada em ficheiros de texto apropriados.

Descrição da solução implementada

Implementámos as classes exigidas e, como esperado, relacionam-se entre si, estando todas, direta ou indiretamente, ligadas. A classe principal é a classe Campeonato, que contém vetores contendo toda a informação relativa ao campeonato. A informação previamente existente é carregada, através de funções load, para esses vetores e são esses vetores que vão ser alterados sempre é feita uma operação CRUD por parte do utilizador. Os elementos destas estruturas de dados que utilizámos são apontadores para outras classes do programa, fazendo com que, a partir da classe Campeonato, se consiga aceder a toda a informação necessária.

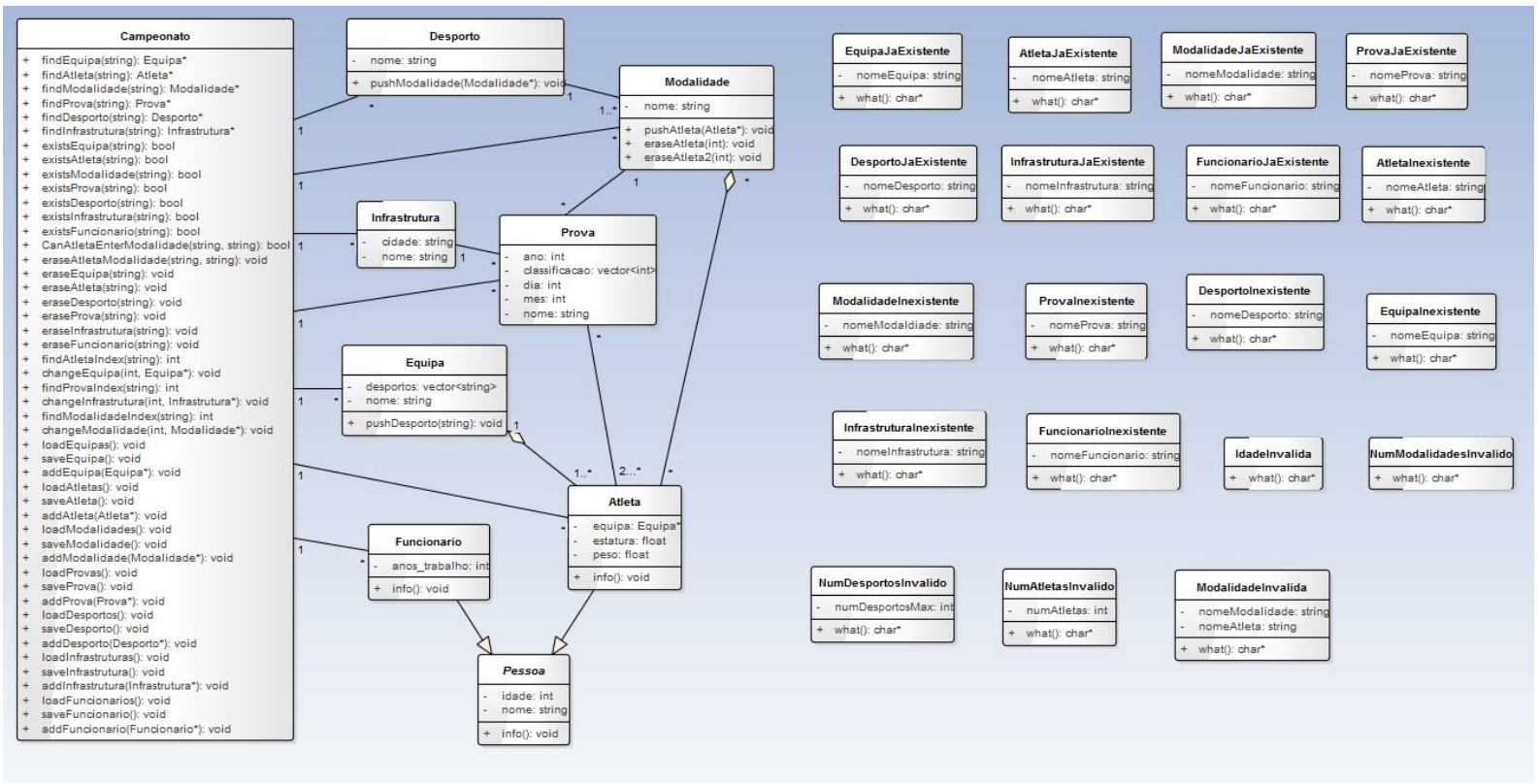
Nas restantes classes pedidas, mais pequenas, implementámos as estruturas de dados que achamos convenientes, nomeadamente vetores e um map, para fazer uma associação entre os atletas e as respetivas classificações. A classe Desporto tem um vetor de objetos da classe Modalidade, que por sua vez tem um vetor de vetor de objetos da classe Atleta. Esta classe Atleta, descendente da classe Pessoa, tem um membro-dado do tipo Equipa*, uma vez que cada atleta apena pode ter uma equipa. A classe Equipa, por sua vez, tem um vetor do tipo string contendo o nome dos desportos em que esta se inscreveu. Para além destas classes, temos também implementadas as classes Funcionário, também esta descendente da classe Pessoa, a classe Infrastrutura e a classe Prova, que tem como membros-dado, entre outros, um map, como já foi referido acima, um apontador para um objeto da classe Modalidade e também um apontador para um objeto da classe Infrastrutura. As provas têm uma data e isso é depois utilizado para fazer uma divisão das provas que já ocorreram e que ainda hão de ocorrer.

Utilizámos conceitos de herança e polimorfismo nas classes Atleta e Funcionário, tendo como classe mãe a classe Pessoa. Uilizámos também tratamento de excepções em todos os casos em que achamos relevante ao longo do trabalho.

Utilizámos o algoritmo de ordenação por inserção para ordenar o vetor e apresentar os atletas com a classificação ordenada crescentemente nas classificações das provas, usamos algoritmo de bubble sort para ordenar os funcionários pelos anos de trabalho, usamos o algoritmo de pesquisa sequencial para encontrar as posições de determinados elementos, todas as funções com “find” no nome, usam este algoritmo.

Diagrama de classes UML

(anexo no .zip para permitir uma compreensão mais eficaz)



Lista de Casos de utilização

- Criar atletas
- Eliminar atletas
- Criar equipas
- Eliminar equipas
- Mudar a equipa de um Atleta
- Criar um desporto e respetiva(s) modalidade(s)
- Eliminar um desporto
- Adicionar atletas à modalidade
- Eliminar atletas duma modalidade
- Criar provas
- Eliminar provas
- Criar infraestruturas
- Eliminar infraestruturas
- Atribuir infraestrutura a uma prova
- Alterar a infraestrutura de uma prova
- Visualizar uma lista das provas que já aconteceram e as que ainda estão para acontecer com base na data atual introduzida
- Visualizar lista de atletas
- Visualizar lista de modalidades
- Visualizar as classificações de uma prova
- Visualizar as classificações de um atleta em todas as provas da modalidade

Dificuldades

O trabalho foi mais extenso que o esperado e isso em conjugação com mais dois trabalhos e relatórios para fazer complicou bastante o desenvolvimento deste projecto

A área de maior dificuldade foi a construção das funções de gravação e carregamento de ficheiros pois tínhamos muitas classes e muitos vetores.

Esforço dedicado pelos elementos do grupo

Dividimos funções e classes, implementámo-las, e depois ajudamo-nos mutuamente para construir as funções de gravação e leitura de ficheiros, bem como os menus. Foi um trabalho bastante extenso que exigiu a participação de todos.