
Organização de Castings

Parte 2

MIEIC - Algoritmo e Estrutura de Dados - 2017/2018

André Lázaro / Marisa Oliveira - 5 de Janeiro de 2018



FEUP **FACULDADE DE ENGENHARIA**
UNIVERSIDADE DO PORTO

Introdução

No âmbito da unidade curricular de Algoritmos e Estrutura de Dados, constante no plano curricular do Mestrado Integrado de Engenharia Informática e Computação, foi desenvolvida a segunda parte do trabalho prático proposto pelos docentes, tendo sido realizado o projeto de “Organização de Castings”.

Neste trabalho foi modelado um problema recorrendo ao paradigma da orientação por objetos e utilizada a linguagem C++ para implementar a solução encontrada para resolver o problema.

Descrição do Tema

O desenvolvimento desta aplicação teve como objetivo a gestão/organização de castings, nas suas duas diferentes fases.

Cada sessão possui 3 jurados, sendo um deles o responsável pela sessão, tendo cada sessão um género de arte performativa associada e que será avaliada. Ainda em relação à sessão, nunca pode existir duas sessões de castings de uma arte performativa no mesmo dia.

Em relação aos candidatos, a aplicação deve ser capaz de registar a informação dos mesmos como, por exemplo, o seu nome e idade, entre outras. Cada candidato tem um id associado de forma a ser possível o jurado responsável avaliar a sua atuação.

Relativamente ao processo de seleção do candidato vencedor, este processo ocorre em duas fases. Numa primeira fase, a pontuação é calculada como a média das pontuações dadas por cada jurado. Numa segunda fase, participam os cinco candidatos com maior pontuação da primeira fase, sendo o que candidato com maior pontuação, o vencedor.

A aplicação, para facilitar a criação de sessões, é capaz de ordenar os candidatos pela arte a que estão inscritos. De forma a preparar as sessões, é possível realizar entrevistas de forma a saber se o candidato realizou outras sessões no passado e, ainda, é possível saber qual foi a sessão mais recente que o candidato realizou. Por fim, para conciliar as sessões com a agenda profissional dos candidatos, é pedido aos mesmos que indiquem os períodos que estão indisponíveis. Após o tempo de indisponibilidade ter terminado, os candidatos voltam normalmente aos castings.

Concluindo, o sistema é capaz de gerir todo o processo de organização dos castings, incluindo inscrições, seleção de jurados e registo de resultados.

Descrição da Solução Implementada

O desenvolvimento desta aplicação destina-se à organização de castings, estando as suas funcionalidades acessíveis a qualquer utilizador.

Nesta segunda parte do trabalho foram adicionadas árvores binárias de pesquisa, filas de prioridades e tabelas de dispersão, de modo a tornar o programa mais eficiente.

As árvores binárias de pesquisa implementadas, têm como objetivo armazenar os candidatos, ordenando-os por género de arte.

As filas de prioridade são utilizadas para saber quantas sessões já efetuou um candidato e qual foi a última sessão em que participou.

Por fim, as tabelas de dispersão, foram criadas de modo a ser possível obter a disponibilidade dos candidatos para as sessões, ou seja, se são compatíveis com uma dada sessão.

Candidato

A classe **Candidato**, herdeira da classe mãe **Pessoa**, é onde tem funções e atributos para estruturar um candidato.

Nesta segunda parte, a classe **Candidato** adquiriu novos atributos (período de indisponibilidade e razão) e métodos para os conseguir aceder e modificar os atributos adicionados.

Casting

A classe Casting é o núcleo da aplicação. Aqui estão implementadas todas as funções que permitem que a aplicação seja executada de forma eficiente, onde são carregadas todas as informações dos ficheiros e onde se guardam as novas informações nos ficheiros.

Nesta segunda parte do trabalho nesta classe foram acrescentadas as árvores binárias para ordenar candidatos pelo género de arte performativa, filas de prioridade para sabermos quais as sessões onde o candidato participou e qual a

mais recente e tabelas de dispersão para sabermos a disponibilidade dos candidatos.

Classificação

Nesta classe é onde se encontram os métodos que permitem avaliar a performance de um candidato.

Jurado

A classe **Jurado**, herdeira da classe mãe **Pessoa**, é onde tem funções e atributos para estruturar um jurado.

Pessoa

Na classe **Pessoa**, classe mãe, é onde tem métodos e atributos comuns entre os jurados e os candidatos.

Sessão

Na classe **Sessão** encontram-se as funções para criar uma sessão.

Listas de casos de utilização identificados para a aplicação

Esta segunda parte do trabalho recorre, tal como na primeira, a operações básicas CRUD (*Create, Read, Update, Delete*), desta vez relacionadas com as estruturas de dados não lineares supramencionadas.

Assim, destacam-se as seguintes funcionalidades da aplicação de acordo com o primeiro menu apresentado ao correr a aplicação:

Candidatos

- Um *utilizador* pode criar um novo candidato
- Um *utilizador* pode eliminar um candidato
- Um *utilizador* pode alterar um candidato
- Um *utilizador* pode pesquisar um candidato
- Um *utilizador* pode ver todos os candidatos
 - Todos
 - Por género de arte
- Um *utilizador* pode ver os candidatos Indisponíveis

Jurados

- Um *utilizador* pode criar um novo jurado
- Um *utilizador* pode eliminar um jurado
- Um *utilizador* pode alterar um jurado
- Um *utilizador* pode pesquisar um jurado
- Um *utilizador* pode ver todos os jurados
 - Todos
 - Por género de arte
- Um *jurado* pode classificar os candidatos

Sessões

- Um *utilizador* pode criar uma nova sessão
- Um *utilizador* pode listar todas as sessões

Informações

- Um *utilizador* pode ver todas as classificações da primeira e da segunda fase e a sua média
- Um *utilizador* pode ver todos os candidatos
- Um *utilizador* pode ver todos os jurados
- Um *utilizador* pode ver todas as sessões
- Um *utilizador* pode ver o conteúdo da fila de prioridade

Principais dificuldades encontradas

Ao longo do desenvolvimento do projecto o grupo encontrou algumas dificuldades em colocar as ideias em ordem de forma a fazer corretamente o que o enunciado pedia. Apesar do esforço da equipa, houve alguns obstáculos em responder na totalidade ao que era pretendido com o projecto, sendo que simplificamos o enunciado para conseguirmos obter o trabalho final. Além disso, a maior dificuldade encontrada, que nos atrasou um pouco, foi a junção do código. visto que cada elemento do grupo ficou responsável por uma dada parte do trabalho e, no momento da junção de ambas as partes, surgiram alguns erros que atrasaram a implementação de todas as funções pedidas.

Esforço dedicado por cada elemento do grupo

As tarefas foram igualmente repartidas e houve colaboração entre os elementos do grupo com o objectivo de se ultrapassarem as dificuldades que foram surgindo.