
Organização de Castings

MIEIC - Algoritmo e Estrutura de Dados - 2017/2018

André Lázaro / Marisa Oliveira - 24 de Novembro de 2017



Índice

Índice	2
Introdução	3
Descrição do Tema	4
Descrição da Solução Implementada	5
Diagrama UML do modelo de dados concebido	9
Listas de casos de utilização identificados para a aplicação	10
Principais dificuldades encontradas	10
Esforço dedicado por cada elemento do grupo	10

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 primeira 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.

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.

A solução encontrada foi projetada com base na utilização de classes adequadas para a representação das entidades envolvidas e na seleção de atributos que caracterizam de forma eficaz essas entidades. Sempre que foi considerado pertinente, foram consideradas relações entre classes, utilizando o conceito de herança e polimorfismo, assim como estruturas lineares, tais como vetores, para facilitar a gestão de conjuntos de elementos de uma mesma entidade ou classe. O tratamento conveniente de possíveis exceções, assim como a atualização de informação em ficheiros de texto para uso futuro foi também um dos nossos objetivos. Esta metodologia permite ao utilizador da aplicação registar e gerir entidades, conjuntos de entidades e relações entre elas, ou seja, permite a realização de operações básicas CRUD (Create, Read, Update, Delete). Em seguida, faz-se uma abordagem particular de algumas das classes criadas.

Na classe **Pessoa**, classe base das classes **Candidato** e **Jurado**, foram considerados os seguintes **membros-dado**:

- ID da pessoa;
- Nome da pessoa;
- Morada da pessoa;
- Género da pessoa.

Os membros-função prendem-se apenas com as funções que permitem obter ou alterar o id, nome, morada e género. Nesta situação utilizámos o conceito de herança e polimorfismo.

Na classe **Candidato**, derivada de Pessoa, considerámos os seguintes **membros-dado**:

- Data de nascimento.

Os **membros-função** definidos são:

- Obter a data de nascimento do candidato;
- Alterar a data de nascimento do candidato.

Na classe **Jurado**, derivada de Pessoa, considerámos os seguintes membros-dado:

- Telemóvel;

Os **membros-função** definidos são:

- Obter o telemóvel do jurado;
- Alterar o telemóvel do jurado.

Na classe **Casting**, foram considerados os seguintes **membros-dado**:

- O nome dos ficheiro de textos criados;
- O vetor geral dos candidatos;
- O vetor geral dos jurados;
- O vetor geral das sessões;
- O vetor geral das classificações;
- O máximo ID a ser gerado aquando da criação de um novo candidato ou jurado.

Nesta classe foram considerados diversos **membros-função**, que permitem ao utilizador da aplicação realizar diversas operações, sendo possível:

- Obter o numero de candidatos e jurados;
- Procurar candidatos ou jurados pelo seu ID;
- Adicionar/Editar/Remover um candidato;
- Adicionar/Editar/Remover um jurado;

-
- Adicionar / Editar uma nova sessão;
 - Classificar candidatos;
 - Listar todos os candidatos por nome ou por ID;
 - Listar todos os jurados por nome ou por ID;
 - Listar as classificações dos candidatos;
 - Guardar todas as alterações feitas nos respetivos ficheiros de texto.

Na classe **Classificação**, foram considerados os seguintes **membros-dado**:

- O ID do candidato a ser avaliado;
- Um array para as 3 notas atribuídas ao candidato;
- O ID da sessão onde o candidato participou e obteve as classificações;
- A média das classificações da primeira fase;
- A média das classificações da segunda fase.

Nesta classe foram considerados diversos **membros-função** que servem para:

- Calcular as classificações da primeira fase;
- Calcular as classificações da segunda fase.

Na classe **Sessão**, foram considerados os seguintes **membros-dado**:

- O ID da sessão;
- O género de arte da sessão;
- A data em que ocorre a sessão.

Nesta classe foram considerados diversos **membros-função** que servem para:

- Obter o ID da sessão;
- Obter a data da sessão;
- Obter o género de arte da sessão;
- Calcular as classificações da primeira fase;
- Calcular as classificações da segunda fase;

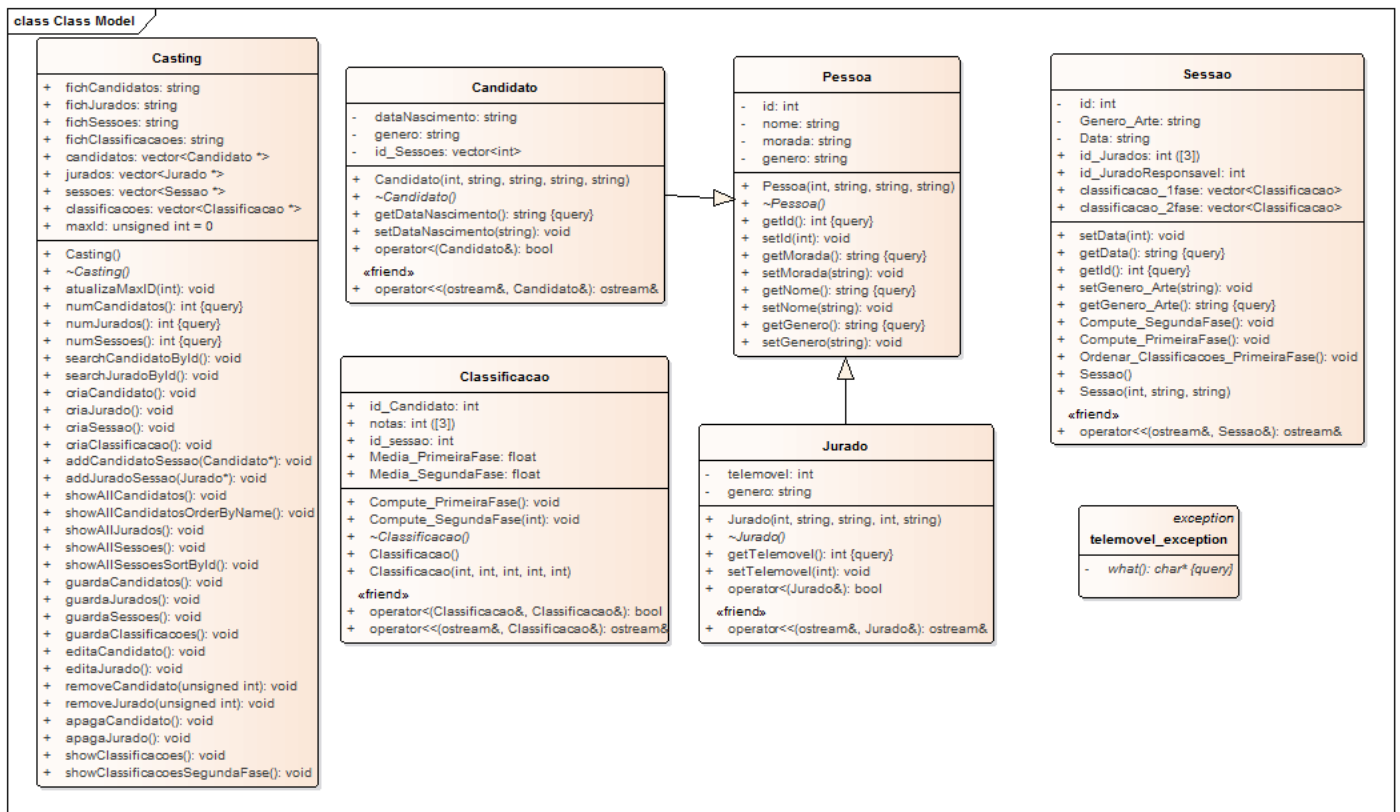
-
- Ordenar as classificações.

As exceções que considerámos mais relevantes implementar foram:

- Inserção de um numero de telemóvel inferior a 90000000 e superior a 99999999.
- Quando o utilizador escreve um nome com mais de 100 caracteres é lançada uma exceção.

Diagrama UML do modelo de dados concebido

Aqui apresenta-se o diagrama UML concebido:



Listas de casos de utilização identificados para a aplicação

A aplicação para a organização de Castings construída permite :

- Gerir candidatos, jurados, sessões e avaliar a performance dos candidatos nas sessões onde eles participam;
- Adicionar, editar e remover candidatos, jurados, sessões.

Principais dificuldades encontradas

Ao longo do desenvolvimento do projeto foram surgindo alguns contratemplos devido a incompatibilidades horários dos elementos do grupo, no entanto, o recurso às novas tecnologias permitiram ultrapassar estas dificuldades.

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.