

**MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO**

**CASTINGS TV**

Antero Gandra, 201607926 up201607926@fe.up.pt

Bruno Vale Fernandes, 201 up201606214@fe.up.pt

Margarida Silva, 201606214 up201606214@fe.up.pt

**RESUMO**

Com este trabalho pretendemos demonstrar como é possível implementar um sistema de castings de uma empresa usando os diversos tipos de estruturas de dados.

De modo a abordar corretamente e organizadamente o jogo, separamo-lo em vários módulos diferentes para implementar: a empresa, as pessoas e as audições (cada um com os seus respetivos ﬁcheiros) .

A nossa maior diﬁculdade foi em saber como implementar xxxxxxxxxxxxxx sendo que o que nos deu mais problemas foi xxxxxxxxxxxxxxxxxxxxxxx. Acabamos por optar por xxxxxxxxxxxxxxxxxxxxxxxxxxxx.

Outro grande desaﬁo foi xxxxxxxxxxxxxxxxxxxxxxxxxxxxx para xxxxxxxxxxxxxxxxxxx. Perante este problema, a solução passou por xxxxxxxxxxxxxxxxxxxxxxxxxxx.

Deste modo conseguimos planificar o funcionamento correto da empresa e implementá-lo, de modo a que a experiência do utilizador com o nosso programa seja simples, organizada e o mais eficiente possível.

**ÍNDICE**

**DESCRIÇÃO DO PROBLEMA:**

Uma empresa do ramo televisivo pretende criar um sistema de planeamento de castings.

De modo a fazer um casting organizam-se sessões com 3 júris, sendo que cada sessão é baseada numa só arte performativa, sendo que o jurado responsável se considera especialista nessa área.

Por sessão são necessárias duas fases sendo que na primeira, a pontuação dos participantes é decidida com igualdade pelos 3 jurados. Já na segunda fase o peso da pontuação do júri responsável da sessão vai valer o dobro. Nesta fase só participam os 5 melhores concorrentes.

**REQUISITOS/RESTRIÇÕES:**

As pontuações variam entre um a dez.

O número mínimo de participantes por sessão é seis (de modo a, por questões logísticas, haverem duas fases).

O tempo por participante na primeira fase são quinze minutos e na segunda fase meia hora.

**DADOS:**

Por participante e por Júri -> id, nome, morada, número de telemóvel, morada, especialidade e participações anteriores.

Por sessão -> id, início, fim, especialidade, júri, líder.

**DIAGRAMA UML**

**GESTÃO DE PESSOAS**

A organização de um casting exige um controlo sobre quer quem é avaliado quer quem avalia. Deste modo criámos duas classes, a Judge e a Contestant (que são herdadas pela classe Person).

De modo a gerenciar os jurados usamos as funções blevleble, blebleble e blebleble, que basicamente o que fazem é vvvvvvvvvvvvvvvvvvvvvv.

Para gerenciar os candidatos usamos beferfeferfer, que faz bbbbbbbbbbbbbbbbbbbb.

Para além de befeferfeefe usamos as as funções xxx e xxxxxx, com as quais acedemos e alteramos brgergerfvgefrvre.

**CRIAÇÃO DE PESSOAS**

**GESTÃO DE TEMPO**

**PROCESSAMENTO DAS FASES**

**SISTEMA DE CLASSIFICAÇÃO DOS CANDIDATOS**

**OUTRAS FUNÇÕES UTEIS**

**INTERFACE**

**CONCLUSÃO**

Com este projeto conseguimos implementar de uma maneira interativa e interessante os conceitos aprendidos em aula. Aplicamos várias dos diversos tipos de algoritmos na resolução de problemas mais complexos e empregamos a maioria das estruturas de dados ensinados nas aulas. Deste modo adquirimos conhecimentos na área da eficiência algorítmica e na área de planeamento que é existente nos castings.

**AVALIAÇÃO INDIVIDUAL POR ELEMENTO DO GRUPO**

Antero Gandra:

Bruno Vale Fernandes:

Margarida Silva:

**RECURSOS**

* Slides das aulas teóricas
* http://www.cppreference.com/