

**Universidade da Beira Interior**

**Engenharia Informática**

**Informática Web**

**Programação Orientada a Objetos**

2º Ano; 1º semestre

**Relatório do Projeto**

Tema 07- Tourada

**Docente: Paula Prata**

Grupo 15:

**Diogo Simões** **-** 41266

**Luís Espírito Santo -** 41400

**Pedro Lopes** **-** 41237

**Rui Teixeira** **-** 34633

**Índice**

1. **Introdução**
2. **Classes e Herança**
3. **Ficheiros**
4. **Main**
5. **Conclusão**

**Introdução**

O seguinte relatório é sobre a realização de um projeto para a Unidade Curricular de Programação Orientada a Objetos onde foi pedido que se criasse uma aplicação para a gestão de Touradas. Esta aplicação permite a gestão de eventos relacionados com touradas e seus envolventes. Neste relatório iremos explicar o funcionamento da nossa aplicação e as classes e subclasses que foram criadas

**Classes e Herança**

Para a realização deste projeto foram criadas as seguintes Classes:

* Capeias
* Cavaleiro
* Corridas
* Entidade
* EspetaculoDeRecortes
* GFA
* Ganadaria
* Largadas
* Local
* Toiros
* Tourada
* TouradasACorda
* Main

Sendo Tourada superclasse de Capeia, Corridas, EspetaculoDeRecortes, Largadas, TouradasACorda e Entidade superclasse de Cavaleiro e GFA. Assim podemos aproveitar as variáveis e métodos das superclasses para as subclasses. As classes Local, Ganadaria e Toiros não necessitam de herança, pois não existe relação entre si. A classe Tourada recebe no seu construtor um inteiro que é o número de assistências, uma data e o nome do evento, e ainda faz a contagem das touradas criadas. O construtor das subclasses de Tourada são muito idênticos pois recebem as mesmas variáveis da superclasse e todas elas precisam de ganadarias e toiros para a sua realização. Existe alguma diferença entre alguns eventos que é a utilização de cavaleiros e grupo de forcados amador(Gfa). Alguns desses eventos também tem bilhetes e total de lugares disponíveis e nessas classes temos funções para venda e devolução de bilhetes e ainda conseguimos verificar a percentagem de bilhetes vendidos. A classe Entidade é onde é feita a criação de uma entidade com nome e idade que depois são utilizadas nas subclasses GFA e Cavaleiro. Em que na classe GFA temos para além do nome e idade temos a terra de onde são e o nome do forcado que dá a cara. Na classe Cavaleiro recebe o nome e a idade e a quantidade de espetáculos que fez. A classe Local é onde são criados locais que é uma localidade e o seu país. A classe Ganadaria é para a criação de ganadarias com um nome uma origem e um conjunto de toiros. A classe Toiros é a criação de toiros com um nome peso e idade.

**Ficheiros**

Para o armazenamento dos dados em ficheiros foram utilizados 6 ficheiros distintos de modo a manter a informação separada. Os ficheiros são:

* fichCapeias.dat
* fichCorridas.dat
* fichEspecRec.dat
* fichLargadas.dat
* fichLocais.dat
* fichTouradasACorda.dat

Estes dados são lidos ao iniciar o programa, de modo a que a informação esteja disponível para o utilizador. Os dados são guardados automaticamente quando o utilizador fecha o programa. Para a gravar a informação em ficheiros foi utilizada a interface Serializable para depois podermos escrever os objetos nos ficheiros respetivos. Para a leitura dos ficheiros foi utilizado o ObjectInputStream em conjunto com o FileInputStream para depois ser possível a leitura dos objetos através do método readObject(). Usando o mesmo processo para a escrita dos ArrayList foi usado o ObjectOutputStream em conjunto com o FileOutputStream para depois ser feita a escrita através do método writeObject().

**Main**

**Conclusão**

Este trabalho foi-nos pedido no âmbito da Unidade Curricular Programação Orientada a Objetos, na vertente prática da mesma. Este projeto permitiu-nos, acima de tudo, consolidar conhecimentos adquiridos durante este primeiro semestre e demonstrar capacidades a partir destes conhecimentos, quer para resolver qualquer problema que fosse surgindo durante a realização deste projeto, quer para a introdução de tudo o que fosse considerado pertinente para que daqui resultasse o trabalho mais completo possível.