

**Grupo:**

Marco Guyard Miranda / 2006110208

Pedro Martins Feliciano / 2017195105

## **POAO**

### **---- Relatório ---**

1. Introdução
2. Estrutura do programa e código
3. Diagrama de classes

#### **1. Introdução**

A vida social e académica de Coimbra gerou a necessidade de se criar um programa para gerir convívios. Este programa foi implementado utilizando a linguagem JAVA. Teve como objectivos a utilização das principais características da Programação Orientada a Objectos: polimorfismo, herança, classes, etc...

#### **2. Estrutura do programa e código**

O programa foi organizado em várias classes. Duas classes centrais são abstratas: “Pessoa” e “Local”. Estas constituem superclasses das quais derivam as respectivas sub-classes. É importante salientar a importância da classe “Inscricao” que relaciona “Pessoa” e “Local”. As inscrições (geradas na classe “Inscricao”) são geridas na classe “Convivio”. Por sua vez, a classe “GestorDeConvivios” gere os convívios gerados na classe “Convivio”.

##### **2.1 Passagens não triviais**

###### **2.1.1**

Na classe “Bar”, o processo de adicionar uma inscrição (método addInscrito - linhas 47 a 59) funciona da seguinte forma: primeiro é verificada se a lotação não foi atingida. Se não foi, a inscrição é imediatamente adicionada. Caso contrário, é necessário verificar se existe alguém com um perfil que não seja “Boêmio”. Caso não encontre, a função vai retornar FALSO e a inscrição não será efectuada. Caso encontre, o nome do inscrito é substituído e consequentemente o número de locais que pode visitar é incrementado, uma vez que foi removido do bar.

Para realizar a procura da pessoa não boémia (procuraNaoBoemio - linhas 79 a 95), são percorridas todas as inscrições realizadas nesse bar e quando é encontrada a última inscrição de um não boémio, dentro do limite do tamanho da guest list, esta última inscrição é devolvida para o método addInscrito onde a sua substituição pela pessoa boémia é feita.

###### **2.1.2**

Na classe “GestorDeConvivios”, ao gerar os ficheiros de texto para locais e pessoas, os campos foram separados por vírgulas. Foi adicionado um campo extra, na primeira posição dos ficheiros que funcionou como prefixo para o sub-tipo do elemento que se está a tratar. Por exemplo, “al” para aluno, “func” para funcionário,

“jar” para jardim, “exp” para exposição, etc... Para as pessoas das linhas 85 a 112. Para locais das linhas 283 a 300.

O objectivo deste prefixo foi facilitar a leitura do ficheiro de modo a separar os campos num array, usando o método split e como argumento a vírgula, sendo posteriormente descartado. Cada posição do array corresponde a um determinado campo, sendo que a partir dessas posições o construtor da classe é chamado.

### 3. Diagrama de Classes

As duas principais alterações do diagrama de classes inicial (ficheiro em anexo: DiagramaClasses\_Inicial”) para o diagrama de classes final (ficheiro em anexo: DiagramaClasses\_Final.violet) foram a introdução de duas novas classes: “Inscricao” e “Convivio”.

A classe convívio surgiu da necessidade de gerir vários convívios.

A classe “Inscricao” surgiu da necessidade de organizar melhor a relação entre inscrições, pessoas e os respectivos locais.