

# Programação e Desenvolvimento de Software II

## Trabalho Prático I

Valor: 15 pontos

Data de entrega: 27/10 às 23h55

O código deve ser enviado através do Laboratório de Programação Virtual do Moodle.

Cada grupo deverá ter 3 integrantes

### 1. Introdução

O objetivo deste trabalho prático é desenvolver um sistema que simula a venda de bilhetes para eventos. As principais entidades de tal sistema a serem modeladas são os eventos e os usuários, que podem comprar ou vender bilhetes para eventos. O trabalho está dividido em duas partes. Na primeira entrega, você deverá ler os dados de entrada e inicializar as estruturas adequadas. Na segunda parte, a tarefa será implementar a lógica de venda de bilhetes.

### 2. Primeira parte

O objetivo da primeira parte do trabalho é implementar a leitura dos dados e inicializar as estruturas que compõem o sistema de venda de bilhetes. Os dados estão armazenados em arquivos de texto no formato CSV<sup>1</sup> (*comma-separated values*), sem espaços entre os campos. Você deve implementar uma função `le_entrada`, que recebe o nome de dois arquivos contendo os dados de usuários e de eventos e inicializa as estruturas do sistema.

### 3. Arquivos de entrada

#### 3.1. Dados de usuários

Cada linha do arquivo de usuários está organizada da seguinte maneira:

“Id,Categ,Nome,Idade,Saldo,Responsável”.

- Id: identificador do usuário.
- Categ: categoria do usuário:
  - criança
  - adulto
  - idoso
- Nome: nome do usuário.
- Idade: idade, em anos, do usuário.
- Saldo: valor em reais que o usuário possui em sua conta para comprar bilhetes.
- Responsável: ID do responsável (apenas se o usuário for da categoria Criança).

Note que as linhas do arquivo de usuários podem ter tamanhos diferentes.

#### 3.2. Dados de eventos

O arquivo de eventos, por sua vez, contém os seguintes campos:

“Id,Categ,Subcateg,Nome,Dono,NumTipos,Capac 1,Valor 1,...,Capac k,Valor k”.

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Comma-separated\\_values](https://en.wikipedia.org/wiki/Comma-separated_values)

- Id: identificador do evento.
- Categ: categoria do evento:
  - adulto
  - infantil
- Subcateg: subcategoria do evento:
  - cinema
  - show
  - boate
  - fantoche
- Nome: nome do evento.
- Dono: ID do usuário que é o responsável pelo evento.
- NumTipos: número de tipos de ingressos.
- Capac\_i: número de ingressos de valor igual a Valor\_i,  $i = 1; 2; \dots; NumTipos$ .

Os diferentes tipos de evento também possuem campos específicos após os campos de **capacidade** e **valor**.

- Evento de categoria adulto:
  - Quota\_idoso: número de ingressos que precisam ser reservados para idosos.
- Cinema:
  - Horário\_i: horários de início das sessões do filme.
  - Duração: duração, em horas, do filme.
- Fantoche:
  - Horário\_i: horários de início das sessões de teatro de fantoches.
- Show:
  - Abertura: horário de abertura dos portões.
  - Artista\_i: artistas que vão performar no show.
- Boate:
  - Início: horário de início.
  - Fim: horário de término.

Neste trabalho, os valores de **Valor\_i** estão limitados a 50, 100, 150, 200, 250 e 300 reais. Os valores para horário de eventos estão limitados a horas inteiras, entre 0 e 23. Assim como no caso do arquivo de usuários, as linhas do arquivo de eventos podem ter tamanhos diferentes. Note que eventos livres, como **cinema**, não possuem o campo categoria no arquivo de entrada.

### 3.3. Exemplos

As Figuras 1 e 2 ilustram exemplos de arquivos de entrada de dados de usuários e eventos, respectivamente.

```

1 0,adulto,Arthur Weasley,41,100
2 1,idoso,Albus Dumbledore,109,600
3 2,adulto,Petunia Dursley,38,300
4 3,adulto,Sr Granger,39,300
5 4,adulto,Lucius Malfoy,38,500
6 5,criança,Harry Potter,11,200,2
7 6,criança,Dudley Dursley,11,100,2
8 7,criança,Ron Weasley,11,50,0
9 8,criança,Hermione Granger,11,200,3

```

Figura 1. Exemplo de arquivo de dados de usuários.

```

1 0,infantil,fantoches,Show de Bonecas Voodoo,1,2,50,50,50,100,8,12,20
2 1,cinema,História do Quadribol,1,2,500,50,400,100,12,14,18,2
3 2,infantil,fantoches,Hogwarts: Uma História,0,1,10,50,10,14
4 3,infantil,fantoches,Apresentação de espantalhos falantes,1,1,250,50,16
5 4,adulto,boate,Baladinha Mística,1,2,50,100,200,250,20,23,6
6 5,infantil,fantoches,História da Magia,0,1,200,50,14,16,18
7 6,cinema,Importantes Descobertas Modernas da Magia,1,2,50,50,13,15,19,21,3
8 7,adulto,show,Noite musical no Três Vassouras,1,1,550,300,80,17,Orquestra de Hogwarts,As Esquisitonas
9 8,adulto,boate,Festa no Cabeça de Javali,4,3,30,50,30,100,30,150,20,23,5
10 9,infantil,fantoches,Os Contos do Chapéu de Sapo,1,2,350,50,60,100,9,11

```

Figura 2. Exemplo de arquivo de dados de eventos.

## 4. Modelagem

Você deve modelar as entidades (usuários e eventos) do sistema como classes, de acordo com o diagrama de classes<sup>2</sup> da Figura 3. Os triângulos representam relações de herança, sendo posicionados próximos à superclasse.

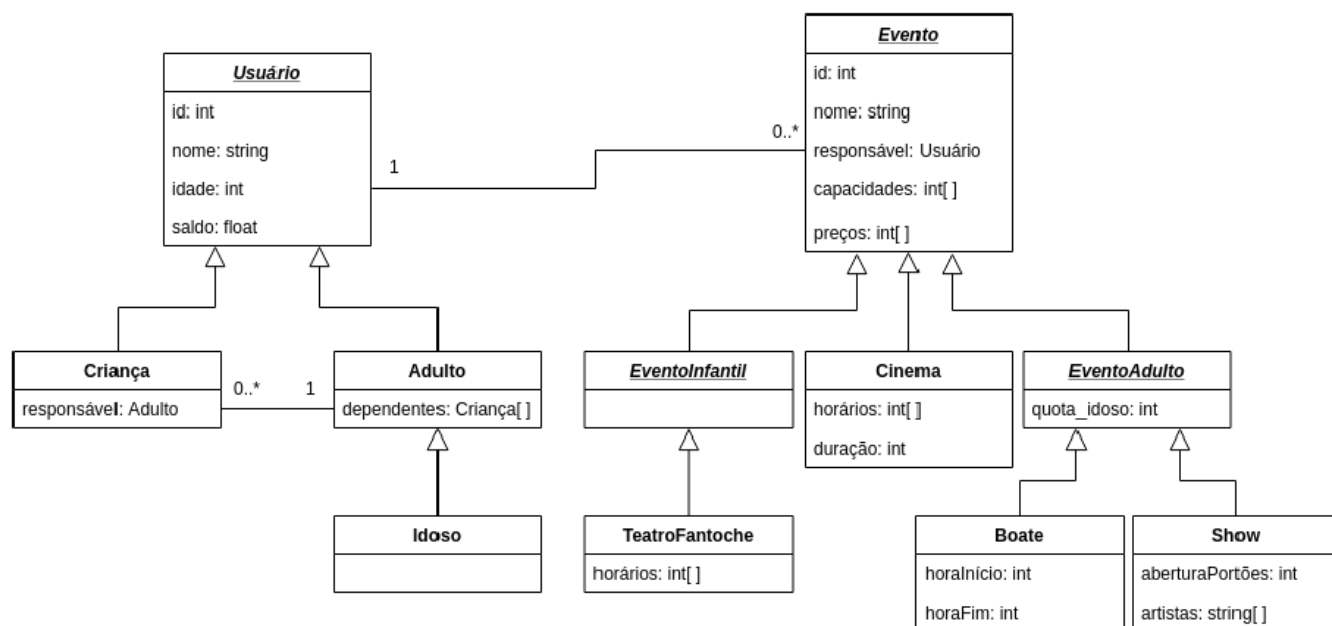


Figura 3. Diagrama de classes do sistema de venda de bilhetes.

Note que a classe Usuário está sublinhada, indicando que ela é uma classe abstrata. Isso significa que ela não pode ser instanciada. Usuários devem pertencer a alguma das classes Criança, Adulto ou Idoso. Tal comportamento se repete para as classes Evento, EventoInfantil e EventoAdulto, que também não podem ser instanciadas.

## 5. Saída

Após ler os arquivos de entrada e criar as estruturas do sistema de venda de bilhetes, seu programa deverá imprimir na saída padrão as seguintes informações:

- Número de crianças, adultos<sup>3</sup> e idosos;
- Idade mínima, máxima e média para todos os usuários;
- Número mínimo, médio e máximo de dependentes por adulto;
- Lista de dependentes por adulto<sup>4</sup>;
- Número de eventos de cada tipo;

<sup>2</sup> [https://en.wikipedia.org/wiki/Class\\_diagram](https://en.wikipedia.org/wiki/Class_diagram)

<sup>3</sup> Note que idosos também são adultos.

<sup>4</sup> Se um adulto não tiver dependentes, seu nome não deve ser impresso.

- Número de eventos vendidos por cada usuário<sup>5</sup>;
- Nome e ID do evento com a maior cota para idosos;
- Número total de bilhetes de cada valor<sup>6</sup>.

A Figura 4 ilustra a saída esperada quando o programa recebe os arquivos de entrada das Figuras 1 e 2. Note que para dependentes e número de eventos por usuário, os usuários devem aparecer na saída ordenados por ID. O trabalho será corrigido de forma automática. Portanto, a formatação da saída deve ser **exatamente** igual à do exemplo fornecido.

```

1  Número de usuários:
2  Crianças: 4
3  Adultos: 5
4  Idosos: 1
5
6  Idade dos usuários:
7  Mínima: 11
8  Máxima: 109
9  Média: 34.33
10
11 Número de dependentes:
12 Mínimo: 0
13 Máximo: 2
14 Média: 0.80
15
16 Dependentes:
17 Arthur Weasley (ID: 0): Ron Weasley (ID: 7),
18 Petunia Dursley (ID: 2): Harry Potter (ID: 5), Dudley Dursley (ID: 6),
19 Sr Granger (ID: 3): Hermione Granger (ID: 8),
20
21 Número de eventos:
22 Adultos:
23     Boate: 2
24     Show: 1
25 Livres:
26     Cinema: 2
27 Infantis:
28     Teatro de Fantoches: 5
29
30 Número de eventos que o usuário possui:
31 Arthur Weasley (ID: 0): 2
32 Albus Dumbledore (ID: 1): 7
33 Lucius Malfoy (ID: 4): 1
34
35 Evento com maior cota para idoso:
36 Noite musical no Três Vassouras (ID: 7): 80
37
38 Número de ingressos por preço:
39 R$50.00: 1440
40 R$100.00: 590
41 R$150.00: 30
42 R$250.00: 200
43 R$300.00: 550
44

```

Figura 4. Saída esperada para os arquivos de entrada das Figuras 1 e 2.

## 6. Segunda parte

A segunda parte do trabalho será a implementação da lógica do sistema de vendas de bilhetes.

## 7. Referência

Adaptação do trabalho elaborado por Fernando Pereira, Helton Matos, Cristiano Pimenta e Tarsila Bessa

<sup>5</sup> Se um usuário não for dono de nenhum evento, seu nome não deve ser impresso.

<sup>6</sup> Em ordem crescente de valor. Se não houver ingressos de um determinado valor, não imprimir.