



Título do trabalho

ATIVIDADE-DUPLA-ENTREGA#02-IMPLEMENTAÇÃO DAS
CLASSES

Nome do professor: Carlos Veríssimo

Nome do Aluno: Pedro Paulo Da Silveira Chaves

Nome da disciplina: PROGRAMAÇÃO ORIENTADA A
OBJETOS

Diagrama de Casos de Uso:

Caso de Uso "Criar Evento": Este caso de uso permite que um usuário (ou organizador) crie um novo evento. Os atores envolvidos são “Usuário” e “Organizador”.

Caso de Uso "Editar Evento": Este caso de uso permite que um usuário (ou organizador) edite as informações de um evento existente.

Caso de Uso "Excluir Evento": Este caso de uso permite que um usuário (ou organizador) exclua um evento existente.

Caso de Uso "Lista de Eventos": Este caso de uso permite que um usuário veja uma lista de eventos disponíveis.

Caso de Uso "Registrador para Evento": Este caso de uso permite que um usuário se registre para participar de um evento.

Diagrama de Classes:

Classe "Evento": Uma classe que representa um evento com atributos como nome, data, local, descrição, etc.

Classe "Usuário": Uma classe que representa um usuário com atributos como nome, email, senha, etc.

Classe "Organizador": Uma subclasse de "Usuário" que representa um organizador de eventos. Podem ter atributos adicionais, como histórico de eventos organizados.

Classe "Inscrição": Representa a inscrição de um usuário em um evento.

Codigo:

```
import  
  
java.text.ParseException;
```

```
import
java.text.SimpleDateFormat;
t;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Scanner;

public class
SistemaGerenciamientoEventos {
    public static void
main(String[] args) {
        Scanner scanner =
new Scanner(System.in);
        List<Evento> eventos
= new ArrayList<>();
        SimpleDateFormat
dateFormat = new
```

```
SimpleDateFormat("dd/MM/yyyy");
```

```
while (true) {
```

```
System.out.println("Escolha uma opção:");
```

```
System.out.println("1. Criar evento");
```

```
System.out.println("2. Listar eventos");
```

```
System.out.println("3. Editar evento");
```

```
System.out.println("4. Remover evento");
```

```
System.out.println("5.  
Sair");
```

```
        int escolha =  
scanner.nextInt();  
        scanner.nextLine();  
// Limpar o buffer de  
entrada
```

```
        switch (escolha) {  
            case 1:  
  
criarEvento(scanner,  
eventos, dateFormat);  
            break;  
            case 2:  
  
listarEventos(eventos);  
            break;
```

case 3:

```
editarEvento(scanner,  
eventos, dateFormat);
```

```
break;
```

case 4:

```
removerEvento(scanner,  
eventos);
```

```
break;
```

case 5:

```
System.out.println("Saindo  
do programa.");
```

```
scanner.close();
```

```
return;
```

default:

```
System.out.println("Opção
```

```
inválida. Tente  
novamente.");  
        break;  
    }  
}  
}
```

```
private static void  
criarEvento(Scanner  
scanner, List<Evento>  
eventos,  
SimpleDateFormat  
dateFormat) {
```

```
    System.out.println("Digite  
o nome do evento:");  
    String nome =  
    scanner.nextLine();
```



```
System.out.println("Digite  
a data do evento  
(DD/MM/AAAA):");  
    String dataStr =  
scanner.nextLine();  
    Date data = null;  
  
    try {  
        data =  
dateFormat.parse(dataStr)  
;  
    } catch  
(ParseException e) {  
  
System.out.println("Forma  
to de data inválido. O  
evento não será criado.");  
        return;  
    }
```

```
System.out.println("Digite  
o nome do organizador:");
```

```
    String  
    nomeOrganizador =  
    scanner.nextLine();
```

```
        Evento evento = new  
Evento(nome, data,  
nomeOrganizador);  
        eventos.add(evento);
```

```
System.out.println("Evento  
criado com sucesso!");  
    }
```

```
    private static void  
listarEventos(List<Evento>  
eventos) {
```

```
System.out.println("Lista  
de Eventos:");  
    for (int i = 0; i <  
eventos.size(); i++) {  
        Evento evento =  
eventos.get(i);  
        System.out.println((i  
+ 1) + ". " +  
evento.getNome() + " - " +  
evento.getDataFormatada(  
) + " - Organizador: " +  
evento.getOrganizador());  
    }  
}
```

```
private static void  
editarEvento(Scanner  
scanner, List<Evento>  
eventos,
```

SimpleDateFormat

dateFormat) {

listarEventos(eventos);

System.out.println("Digite
o número do evento a ser
editado:");

int numeroEvento =
scanner.nextInt();

scanner.nextLine(); //
Limpar o buffer de entrada

if (numeroEvento >= 1
&& numeroEvento <=
eventos.size()) {
Evento
eventoSelecionado =
eventos.get(numeroEvento
- 1);

```
System.out.println("Digite  
o novo nome do evento:");  
  
    String novoNome =  
scanner.nextLine();
```

```
System.out.println("Digite  
a nova data do evento  
(DD/MM/AAAA):");  
  
    String novaDataStr =  
scanner.nextLine();  
  
    Date novaData =  
null;  
  
    try {  
  
        novaData =  
dateFormat.parse(novaDat  
aStr);
```

```
        } catch  
(ParseException e) {  
  
    System.out.println("Forma  
to de data inválido. A  
edição não será  
realizada.");  
        return;  
    }  
}
```

```
eventoSelecionado.setNo  
me(novoNome);
```

```
eventoSelecionado.setDat  
a(novaData);
```

```
System.out.println("Evento  
editado com sucesso!");
```

```
    } else {
```

```
        System.out.println("Número  
de evento inválido.");
```

```
    }
```

```
}
```

```
    private static void  
    removerEvento(Scanner  
    scanner, List<Evento>  
    eventos) {  
        listarEventos(eventos);
```

```
        System.out.println("Digite  
o número do evento a ser  
removido:");
```

```
        int numeroEvento =  
        scanner.nextInt();
```

```
        if (numeroEvento >= 1
&& numeroEvento <=
eventos.size()) {
            Evento
eventoRemovido =
eventos.remove(numeroEv
ento - 1);

System.out.println("Evento
'" +
eventoRemovido.getNome
() + "' removido com
sucesso!");
        } else {

System.out.println("Número
o de evento inválido.");
        }
    }
}
```



```
class Evento {  
    private String nome;  
    private Date data;  
    private String  
organizador;  
    private  
SimpleDateFormat  
dateFormat = new  
SimpleDateFormat("dd/M  
M/yyyy");  
  
    public Evento(String  
nome, Date data, String  
organizador) {  
        this.nome = nome;  
        this.data = data;  
        this.organizador =  
organizador;  
    }  
}
```

```
public String getNome()  
{  
    return nome;  
}
```

```
public void  
setNome(String nome) {  
    this.nome = nome;  
}
```

```
public Date getData() {  
    return data;  
}
```

```
public void setData(Date  
data) {  
    this.data = data;  
}
```

```
    public String  
    getOrganizador() {  
        return organizador;  
    }
```

```
    public String  
    getDataFormatada() {  
        return  
        dateFormat.format(data);  
    }  
}
```