

# Reserva de Restaurantes

Gerenciamento de Restaurantes renomados e reservas

Erick Bernat  
Samuel Victor  
Stella Hada

# Objetivo

Apresentar o sistema de reserva de restaurante, destacando suas funcionalidades e a aplicação dos princípios de orientação a objeto.

# Funcionalidades do sistema

# Funcionalidades do Administrador

- **Adicionar Restaurante:** Permite ao administrador adicionar novos restaurantes com detalhes como nome, localização, tipo de cozinha, etc.
- **Editar Restaurante:** Permite ao administrador atualizar informações de restaurantes existentes.
- **Excluir Restaurante:** Permite ao administrador remover restaurantes do sistema.
- **Gerenciar Reservas:** Visualizar e gerenciar todas as reservas feitas pelos clientes.

# Funcionalidades do Cliente

- **Visualizar Restaurantes:** Clientes podem ver a lista de restaurantes disponíveis.
- **Fazer Reserva:** Clientes podem selecionar um restaurante e fazer uma reserva especificando quantidade de lugares na mesa, data, tipo de experiência etc.

Design 00

Tela Login



Cadastrar

Entrar

Entrar como administrador

Cadastro Cliente

Nome

Email

Senha

Confirmar senha

CPF

Endereço

Entrar

Login Cliente

Nome

Senha

Entrar

## Restaurantes

Digite o nome do restaurante que deseja

### Mani



### Paris 6



### Mani



Endereço: rua mani

Chef: erick

Descrição: restaurante brasileiro

Experiencia

☐ Basica

☒ Completa

☐ Vip

Lugares no menu

Realizar Pagamento



Relatorio de Reservas

Digite uma data para pesquisar por reservas



Nome: Erick Data: 31/05/2024 Restaurante: Mani Valor: R\$2000.0

Nome: Erick Data: 01/06/2024 Restaurante: Mani Valor: R\$2000.0

Nome: Erick Data: 02/06/2024 Restaurante: Mani Valor: R\$700.0

Nome: Erick Data: 02/06/2024 Restaurante: Paris 6 Valor: R\$800.0

Nome: Erick Data: 05/06/2024 Restaurante: Paris 6 Valor: R\$2000.0

Nome: Erick Data: 06/06/2024 Restaurante: Paris 6 Valor: R\$800.0

Nome: Erick Data: 06/06/2024 Restaurante: Mani Valor: R\$700.0

Nome: Erick Data: 08/06/2024 Restaurante: Paris 6 Valor: R\$500.0

Nome: stella Data: 09/06/2024 Restaurante: Paris 6 Valor: R\$400.0

Nome: Heloysa Data: 08/06/2024 Restaurante: Mani Valor: R\$1400.0

Excluir Restaurante

-

paris 6



Paris 6



Cadastrar Restaurante



URL(imagem)

q.restaurantguru.com/r511-Paris-6-Petit-Shopping-Iguatemi-logo.jpg

Nome

Descrição

Chef

Horário de Funcionamento

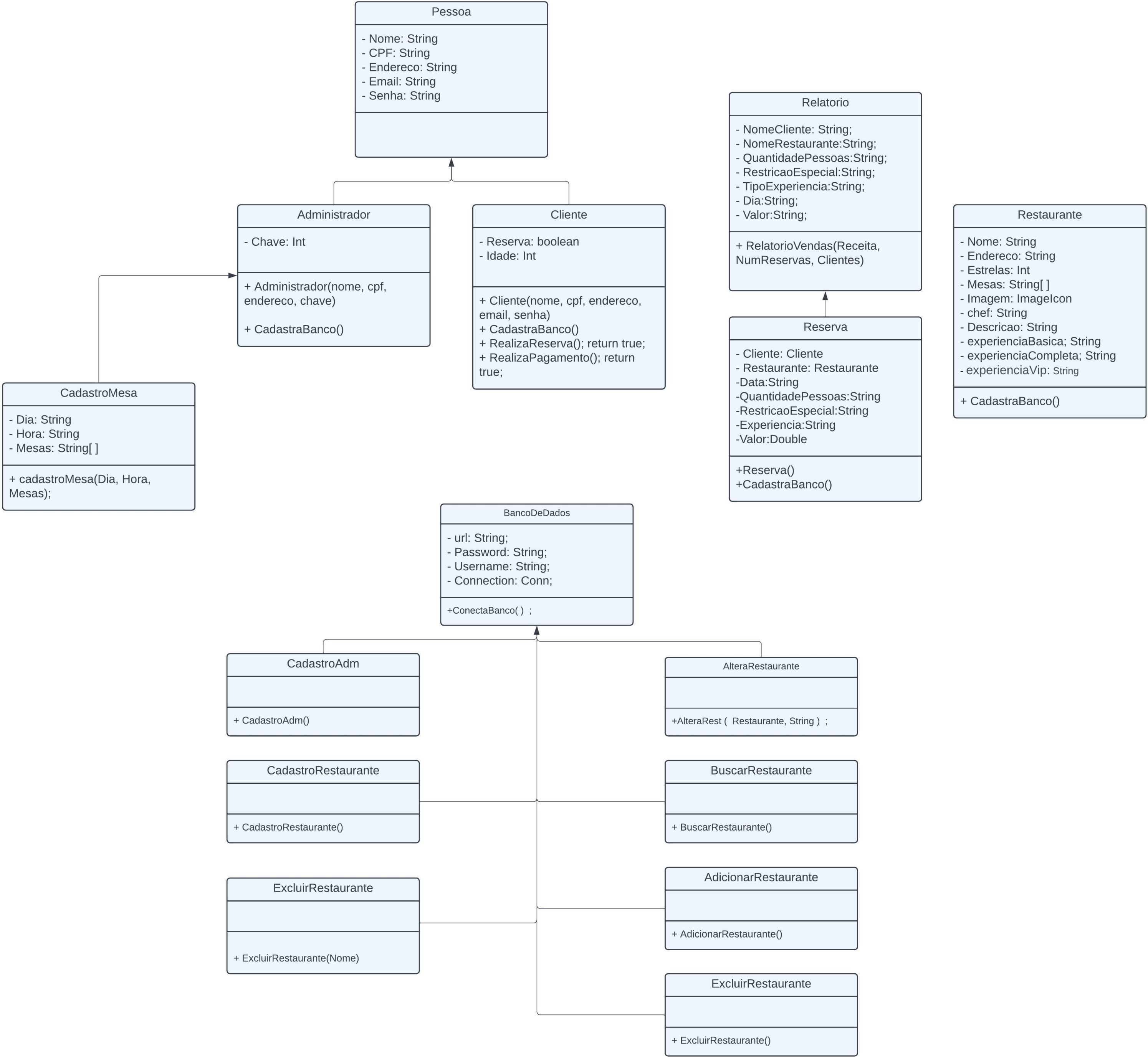
Endereço

- □ X

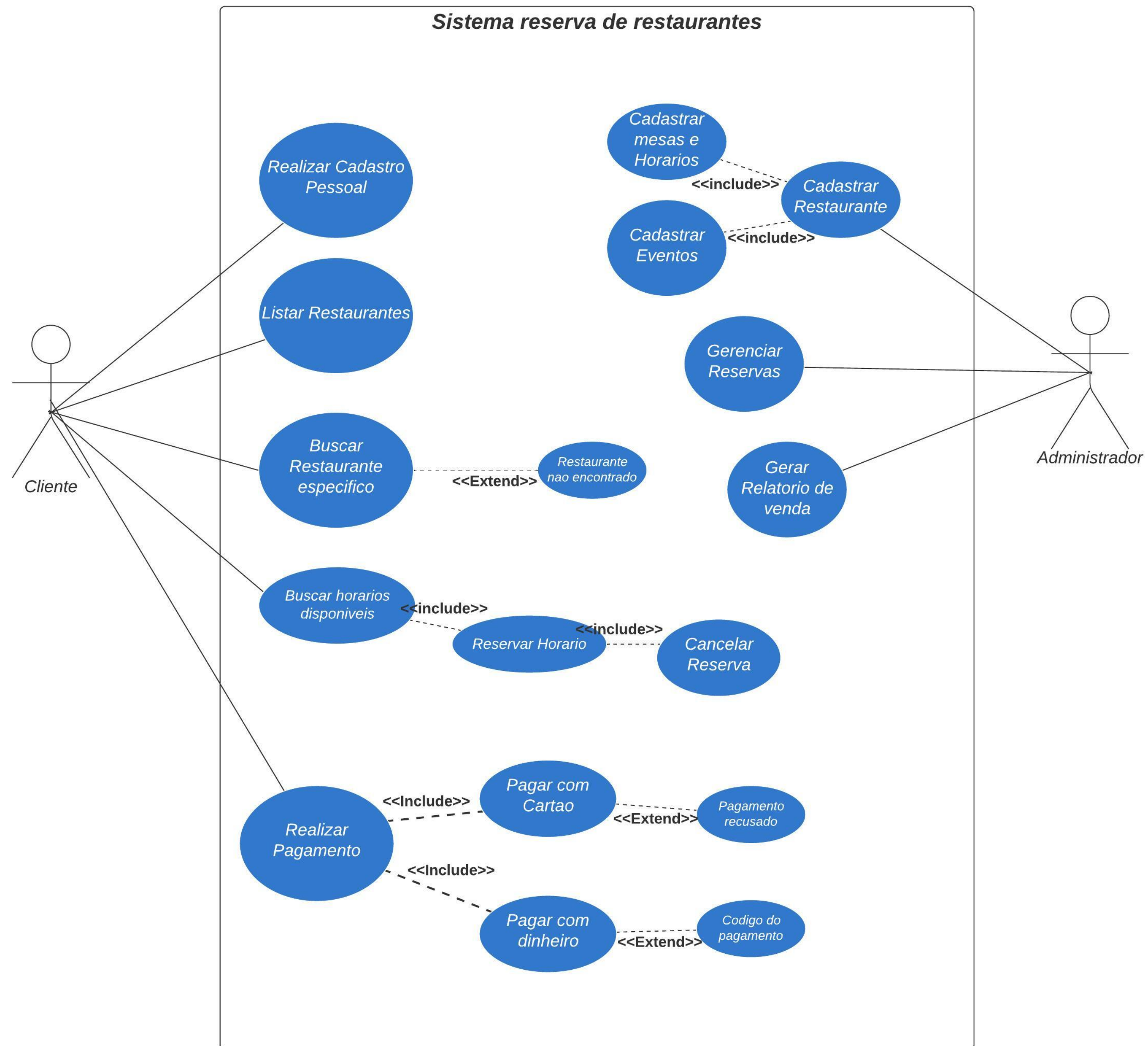
≡

▼

# Diagrama de Classes UML



# Diagrama de caso de uso UML





# Classes principaux

# Control

- Executavel.java
- BancoDeDados.java
- CrudRestaurante.java
- CrudCliente.java
- CrudAdministrador.java

# Model

- Experiencia.java
- Relatorio.java
- Reserva.java
- Pessoa.java
- Administrador.java
- Cliente.java
- Restaurante.java





- TelasAdm.java
- TelasCliente.java
- TelasBase.java

# Assets

- Icons

# Princípios de Orientação a Objetos

# Encapsulamento

Criar a proteção do código para que não haja troca de informações

```
public class Reserva {
    private Cliente cliente;
    private Restaurante restaurante;
    private String data;
    private String quantidadePessoas;
    private String restricaoEspecial;
    private String experiencia;
    private double valor;

    public Reserva() {

    }

    public Reserva(Cliente cliente, Restaurante restaurante, String data,
        String quantidadePessoas, String restricaoEspecial, String experiencia,
        double valor) {
        this.cliente=cliente;
        this.restaurante=restaurante;
        this.data=data;
        this.quantidadePessoas=quantidadePessoas;
        this.restricaoEspecial=restricaoEspecial;
        this.experiencia=experiencia;
        this.valor=valor;
    }

    public void cadastraBanco() {
        CadastraReserva cadastro = new CadastraReserva();
        cadastro.cadastraReserva(this);
    }
}
```

```
public void cadastraBanco() {
    CadastraReserva cadastro = new CadastraReserva();
    cadastro.cadastraReserva(this);
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Restaurante getRestaurante() {
    return restaurante;
}

public void setRestaurante(Restaurante restaurante) {
    this.restaurante = restaurante;
}

public String getData() {
    return data;
}

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

public String getQuantidadePessoas() {
    return quantidadePessoas;
}
```

# Herança

```
public class Administrador extends Pessoa{
    private int chave;

    public Administrador() {

    }

    public Administrador(String nome,String cpf,String email,String senha,String endereco) {
        this.setSenha(senha);
        this.setCPF(cpf);
        this.setNome(nome);
        this.setEmail(email);
        this.setEndereco(endereco);
    }
}
```

A classe 'Administrador' herda os atributos da classe 'Pessoa' para reutilização de código.

# Abstração

```
public abstract class Pessoa {  
    private String nome;  
    private String CPF;  
    private String senha;  
    private String email;  
    private String endereco;
```

Classe especial que não pode ser instanciada, apenas herdada.

# Polimorfismo

```
JPanel novaBarra = new JPanel(new BorderLayout()) {  
    @Override  
    protected void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        g.setColor(Color.black);  
        g.fillRect(0, 0, getWidth(), getHeight());  
        g.setColor(Color.WHITE);  
        g.drawString("Restaurantes", getWidth() / 2 - 30, 25);  
    }  
};
```

Consiste em um mesmo método apresentar comportamentos diferentes, dependendo da classe em que seja chamado.

Subscreve o método `paintComponent` para criar uma nova barra de menu e remover a barra de menu padrão.

# Demonstração do sistema



# Conclusão

# Conclusão

## Resumo das Funcionalidades:

- O sistema permite que administradores adicionem, editem e excluam restaurantes, além de gerenciar reservas.
- Clientes podem visualizar restaurantes e fazer reservas de forma fácil e eficiente.

## Importância da Orientação a Objetos

- O uso de encapsulamento, herança, polimorfismo e abstração foi fundamental para criar um sistema modular e fácil de manter.
- Classes como Restaurante, Reserva, Cliente e Admin foram essenciais para organizar o código e as responsabilidades.

## Benefícios do Sistema:

- Facilita o gerenciamento de restaurantes e reservas.
- Melhora a experiência do cliente ao oferecer uma plataforma simples para fazer reservas.



Obrigado!

