

---

# Documentação de Projeto

para o sistema

## EvenTaste

**Versão 1.0**

Projeto de sistema elaborado pelo(s) aluno(s) Eduarda Silva Santos Nunes  
Giovanna Ferreira dos Santos de Almeida  
Miguel Amaral Lessa Xavier  
como parte da disciplina **Projeto de Software**.

**Maio de 2025**

## Tabela de Conteúdo

|  |          |
|--|----------|
| <b>1. Introdução</b>   | <b>1</b> |
| <b>2. Modelos de Usuário e Requisitos</b>                    | <b>1</b> |
| 2.1 Descrição de Atores                                      | 1        |
| 2.2 Modelo de Casos de Uso e Histórias de Usuários           | 1        |
| 2.3 Diagrama de Sequência do Sistema e Contrato de Operações | 1        |
| <b>3. Modelos de Projeto</b>                                 | <b>1</b> |
| 3.1 Arquitetura  | 1        |
| 3.2 Diagrama de Componentes e Implantação.                   | 2        |
| 3.3 Diagrama de Classes                                      | 2        |
| 3.4 Diagramas de Sequência                                   | 2        |
| 3.5 Diagramas de Comunicação                                 | 2        |
| 3.6 Diagramas de Estados                                     | 2        |
| <b>4. Modelos de Dados</b>                                   | <b>2</b> |

## Histórico de Revisões

| Nome | Data | Razões para Mudança | Versão |
|------|------|---------------------|--------|
|      |      |                     |        |
|      |      |                     |        |

## 1. Introdução

O EvenTaste é um sistema de gestão especializado em eventos culturais gastronômicos, projetado para facilitar a organização, promoção e participação em festivais gastronômicos, workshops culinários, degustações e outros eventos relacionados à cultura alimentar. O sistema visa conectar organizadores, participantes, chefs e estabelecimentos gastronômicos em uma plataforma integrada.

## 2. Modelos de Usuário e Requisitos

### 2.1 Descrição de Atores

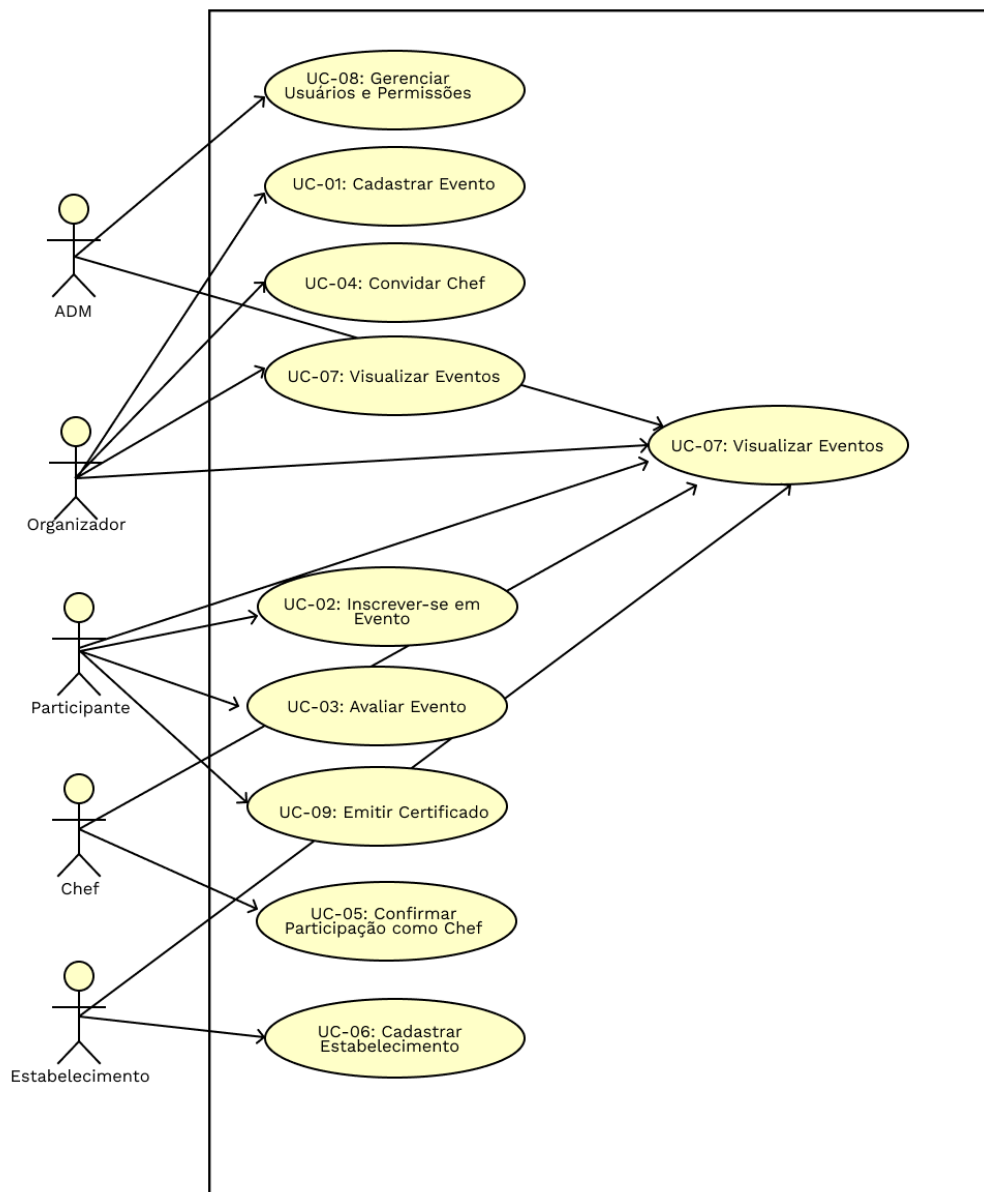
Nesta subseção é apresentada descrição de cada um dos atores que interagem com o sistema.

| Ator            | Descrição  |
|-----------------|--|
| Organizador     | Responsável por criar, editar e gerenciar eventos gastronômicos. Pode visualizar participantes, aprovar inscrições e gerar relatórios do evento.   |
| Participante    | Usuário interessado em participar dos eventos. Pode se inscrever, avaliar eventos, visualizar agenda e emitir certificados de participação.        |
| Chef            | Profissional convidado para conduzir oficinas, palestras e degustações. Pode gerenciar suas atividades no evento e interagir com os participantes. |
| Estabelecimento | Restaurantes, cafés ou espaços gastronômicos parceiros que podem sediar eventos ou ofertar experiências.   |
| Administrador   | Responsável pela manutenção do sistema, aprovação de cadastros de organizadores e estabelecimentos, e controle geral de usuários e permissões.     |

### 2.2 Modelo de Casos de Uso

| ID    | Nome do caso de uso | Ator         |
|-------|---------------------|--------------|
| UC-01 | Cadastrar evento    | Organizador  |
| UC-02 | Inscrever em evento | Participante |
| UC-03 | Avaliar evento      | Participante |

|       |                                 |                 |
|-------|---------------------------------|-----------------|
| UC-04 | Convidar chefe                  | Organizador     |
| UC-05 | Confirmar participação          | Chefe           |
| UC-06 | Cadastrar estabelecimento       | Estabelecimento |
| UC-07 | Visualizar eventos              | Todos           |
| UC-08 | Gerenciar usuários e permissões | Administrador   |
| UC-09 | Emitir certificado              | Participante    |



## 2.3 Diagrama de Sequência do Sistema

### UC-01 Cadastrar Evento

Organizador -> Sistema: solicitarCadastroEvento()  
 Sistema -> Organizador: exibirFormulario()  
 Organizador -> Sistema: enviarFormulario(dadosEvento)  
 Sistema -> Validador: validarDados(dadosEvento)

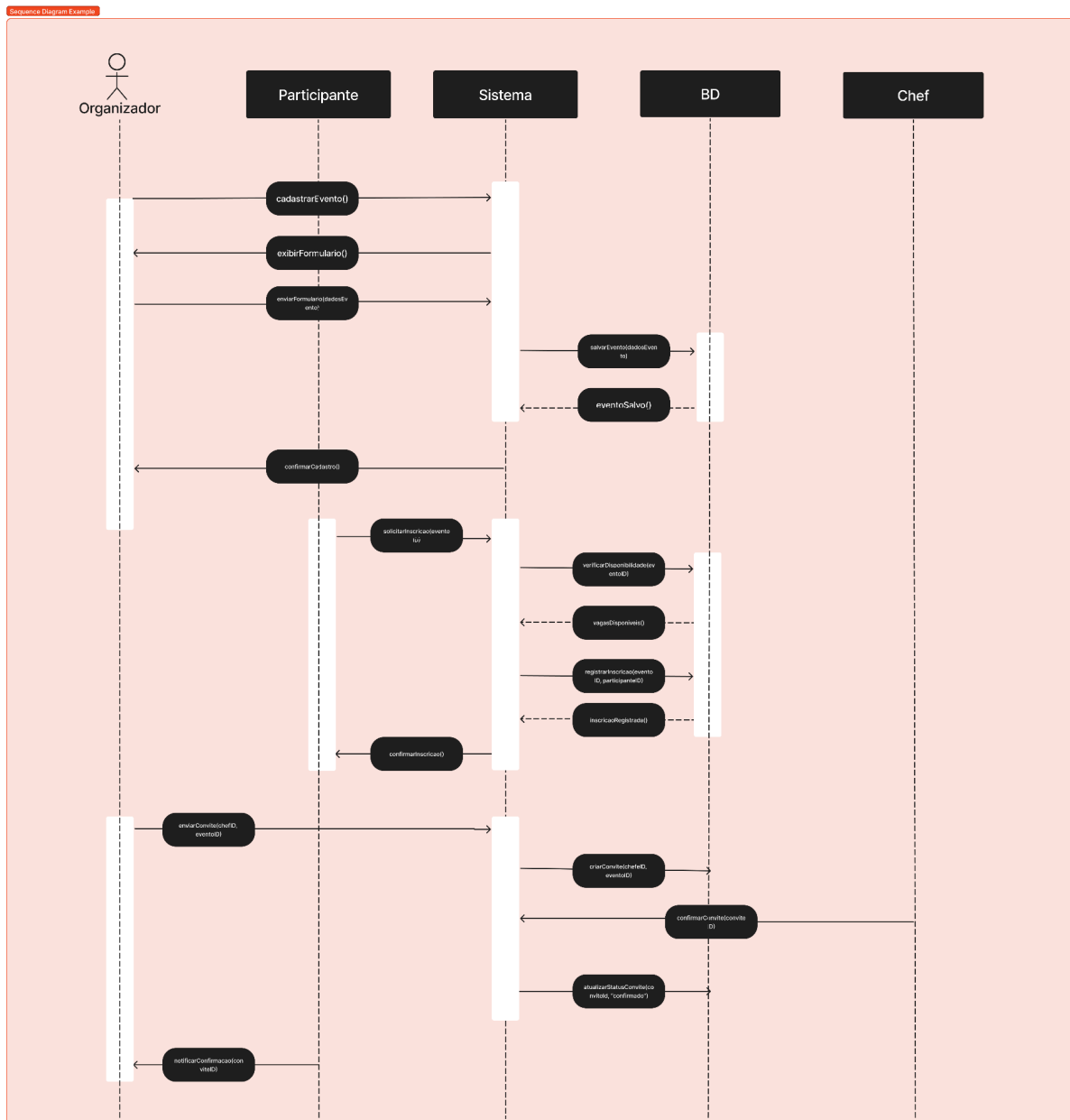
Validador --> Sistema: validaçãoOk  
Sistema -> BancoDeDados: salvarEvento(dadosEvento)  
BancoDeDados --> Sistema: eventoSalvo  
Sistema -> Organizador: confirmarCadastro()

#### **UC-02 Inscrever em evento**

Participante -> Sistema: solicitarInscricao(idEvento)  
Sistema -> BancoDeDados: verificarDisponibilidade(idEvento)  
BancoDeDados --> Sistema: vagasDisponiveis  
Sistema -> BancoDeDados: registrarInscricao(idEvento, idParticipante)  
BancoDeDados --> Sistema: inscricaoRegistrada  
Sistema -> Participante: confirmarInscricao()

#### **UC-05 Confirmar participação como Chefe**

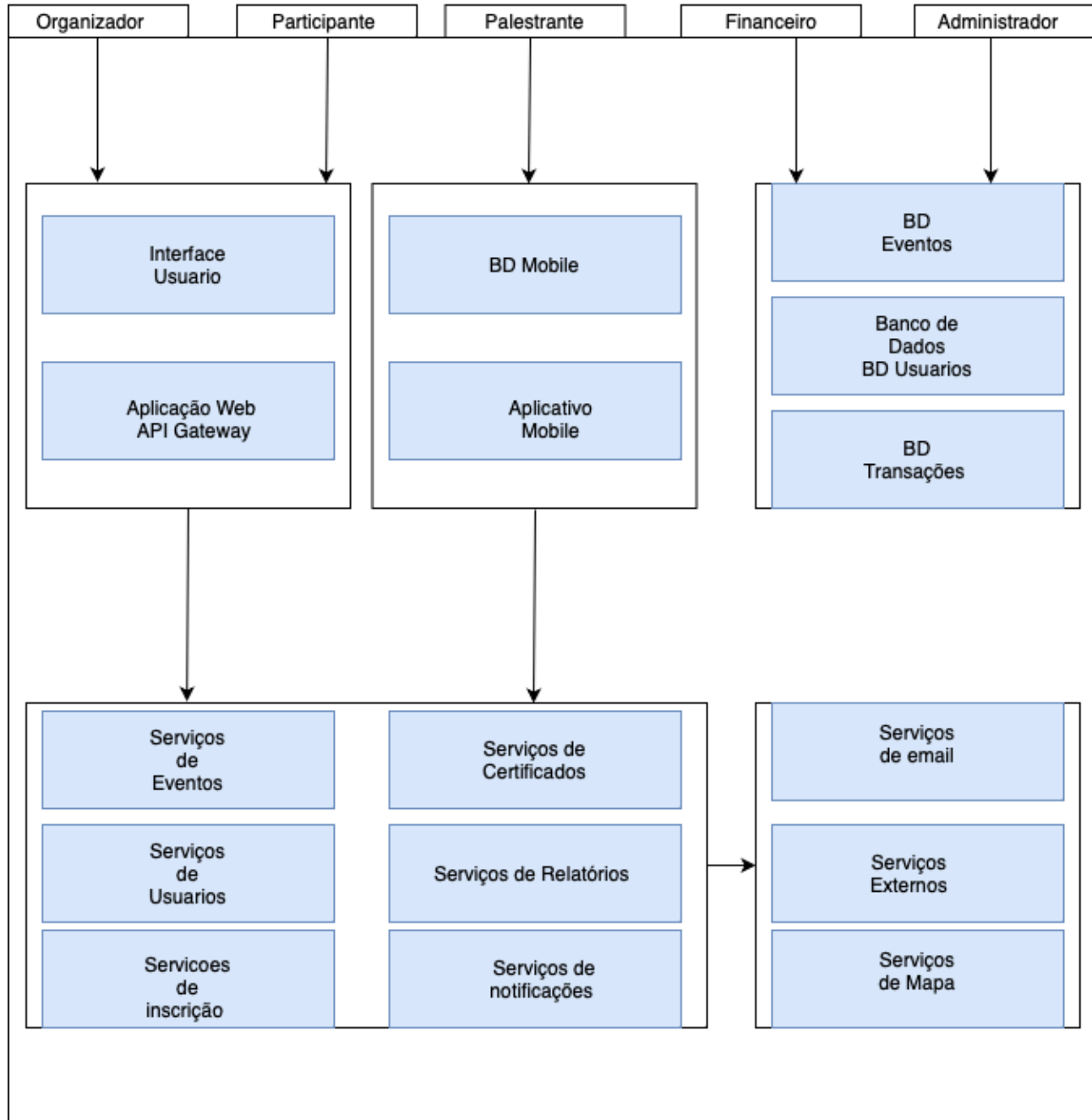
Organizador -> Sistema: enviarConvite(idChef, idEvento)  
Sistema -> BancoDeDados: criarConvite(idChef, idEvento)  
Chef -> Sistema: visualizarConvites()  
Sistema -> BancoDeDados: listarConvitesPendentes(idChef)  
BancoDeDados --> Sistema: listaDeConvites  
Chef -> Sistema: confirmarConvite(idConvite)  
Sistema -> BancoDeDados: atualizarStatusConvite(idConvite, "confirmado")  
Sistema -> Organizador: notificarConfirmacao(idConvite)



## 3. Modelos de Projeto

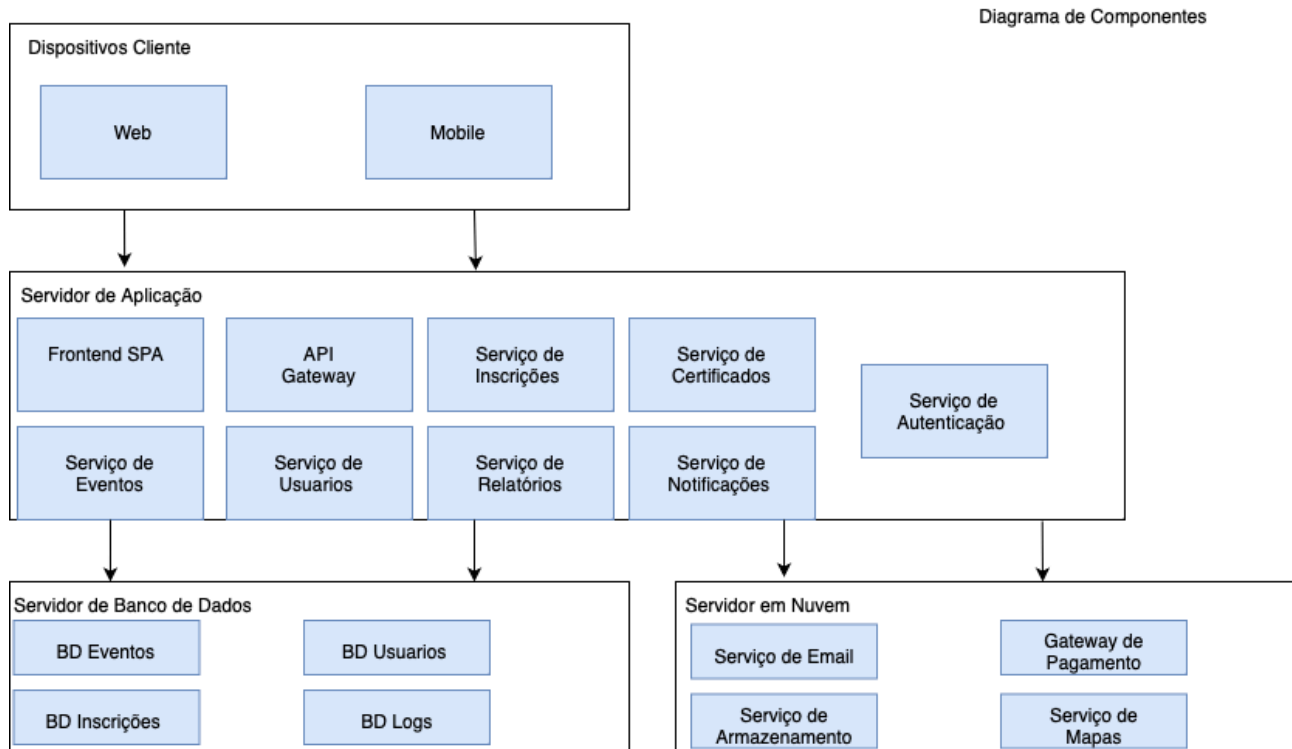
### 3.1 Arquitetura

Diagrama de Arquitetura

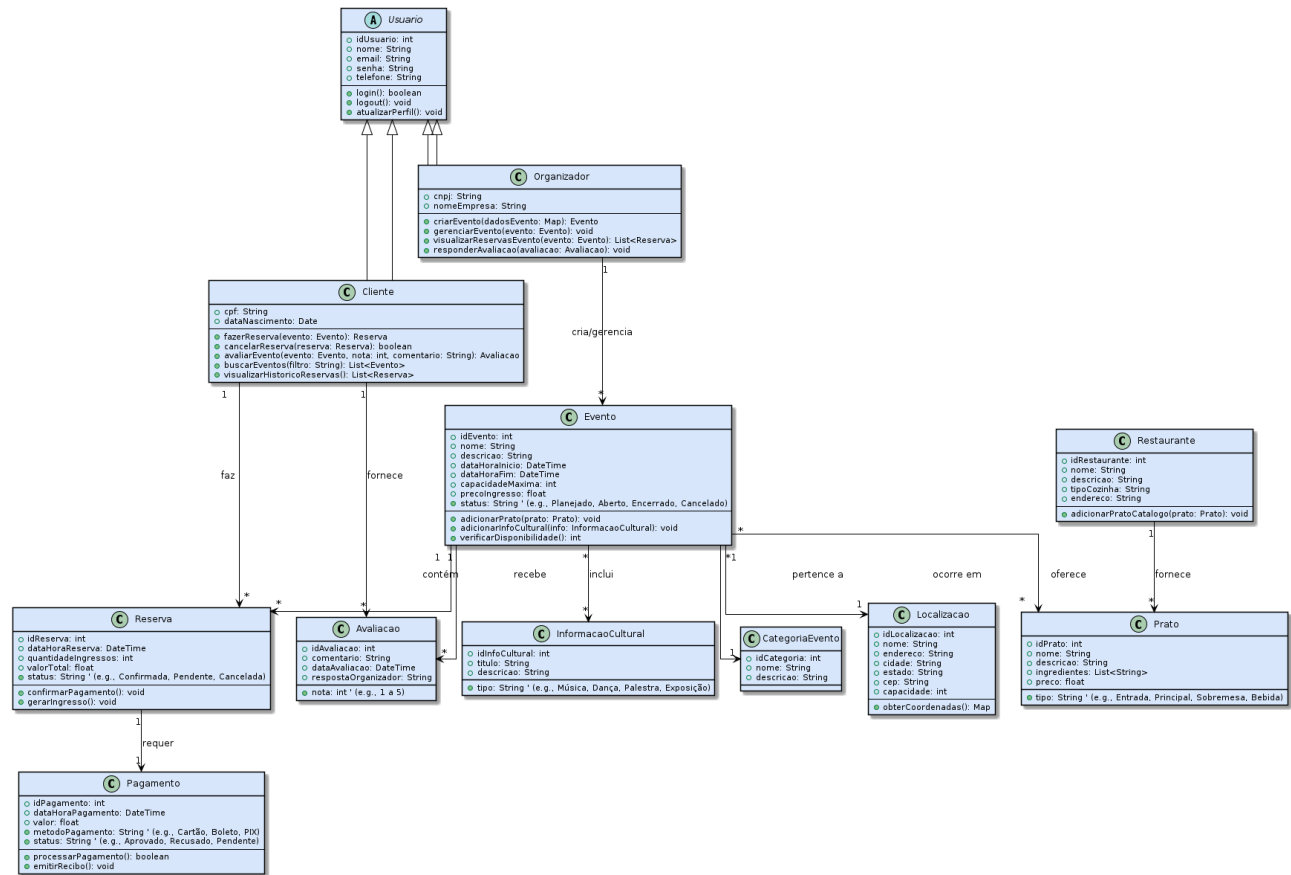




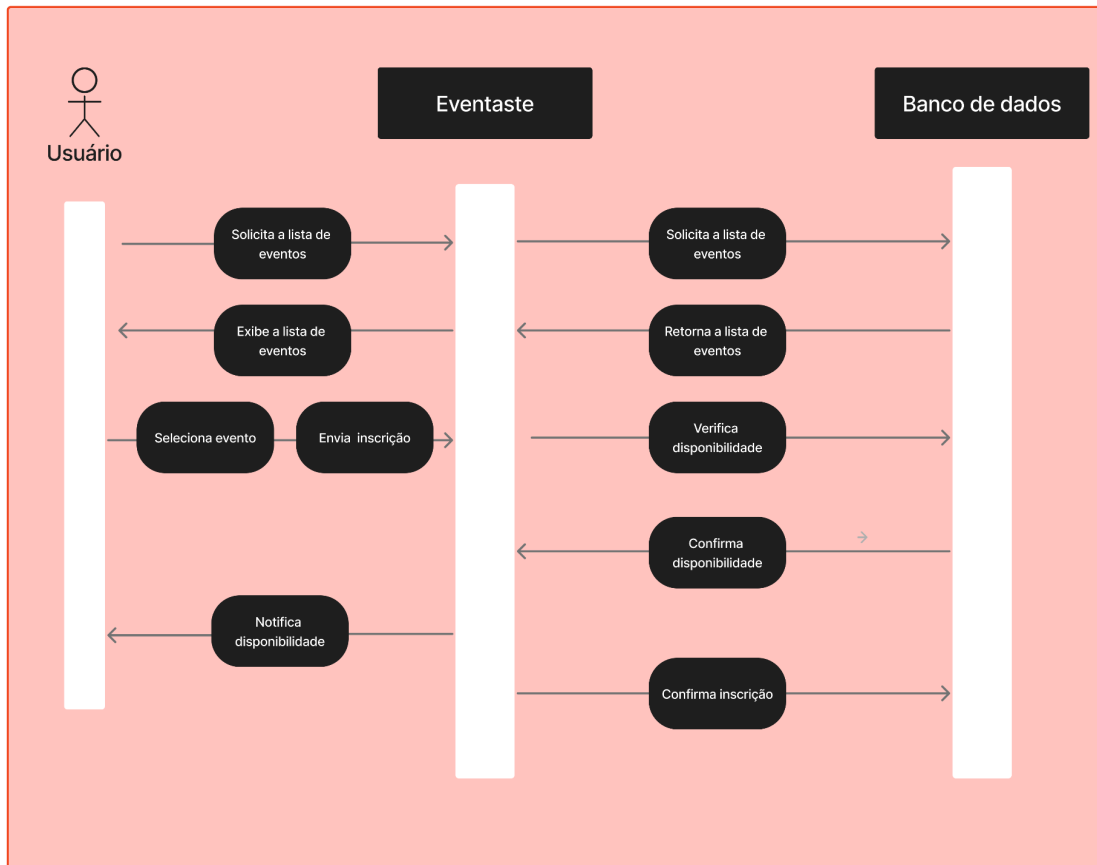
### 3.2 Diagrama de Componentes e Implantação.



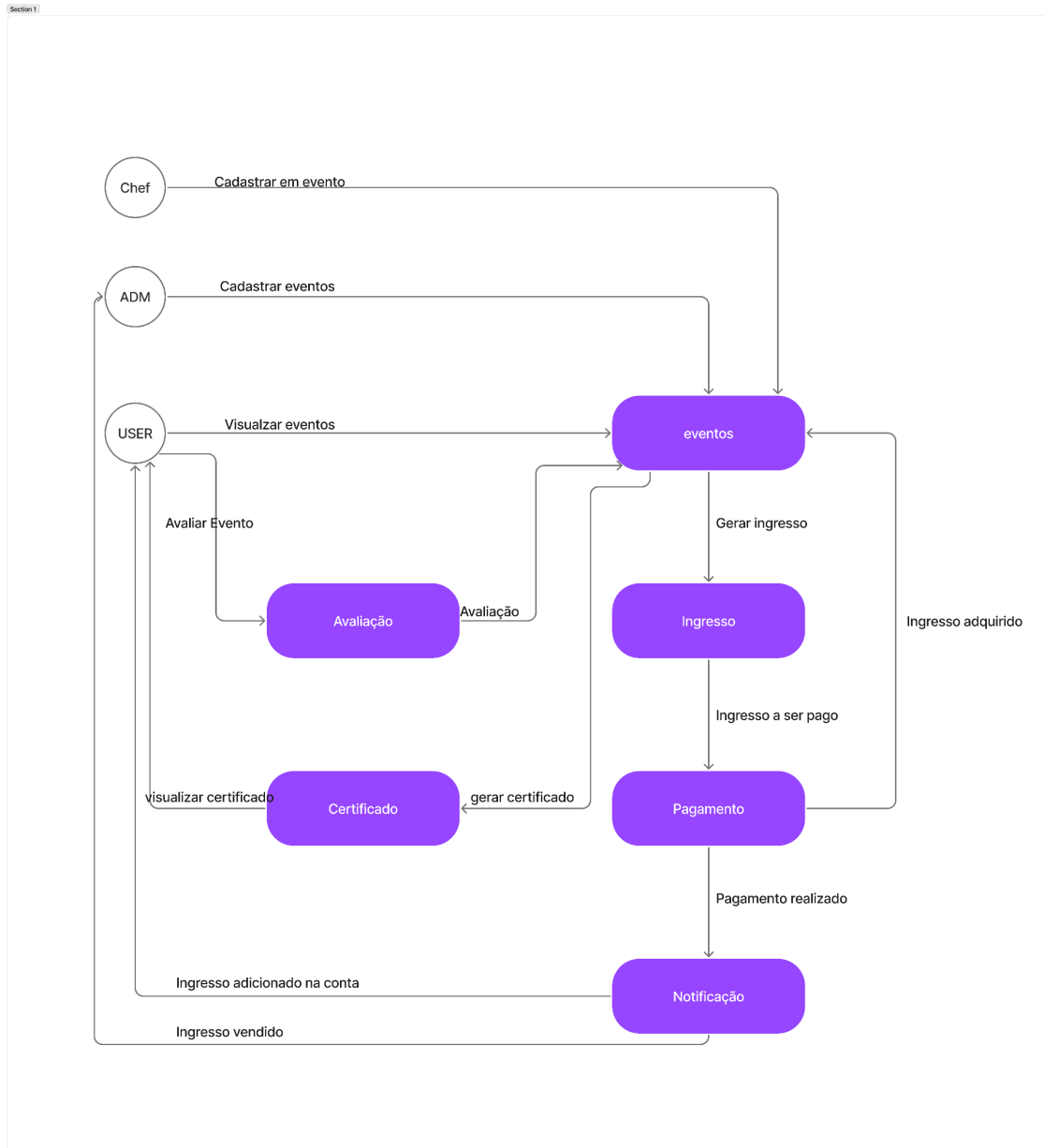
### 3.3 Diagrama de Classes



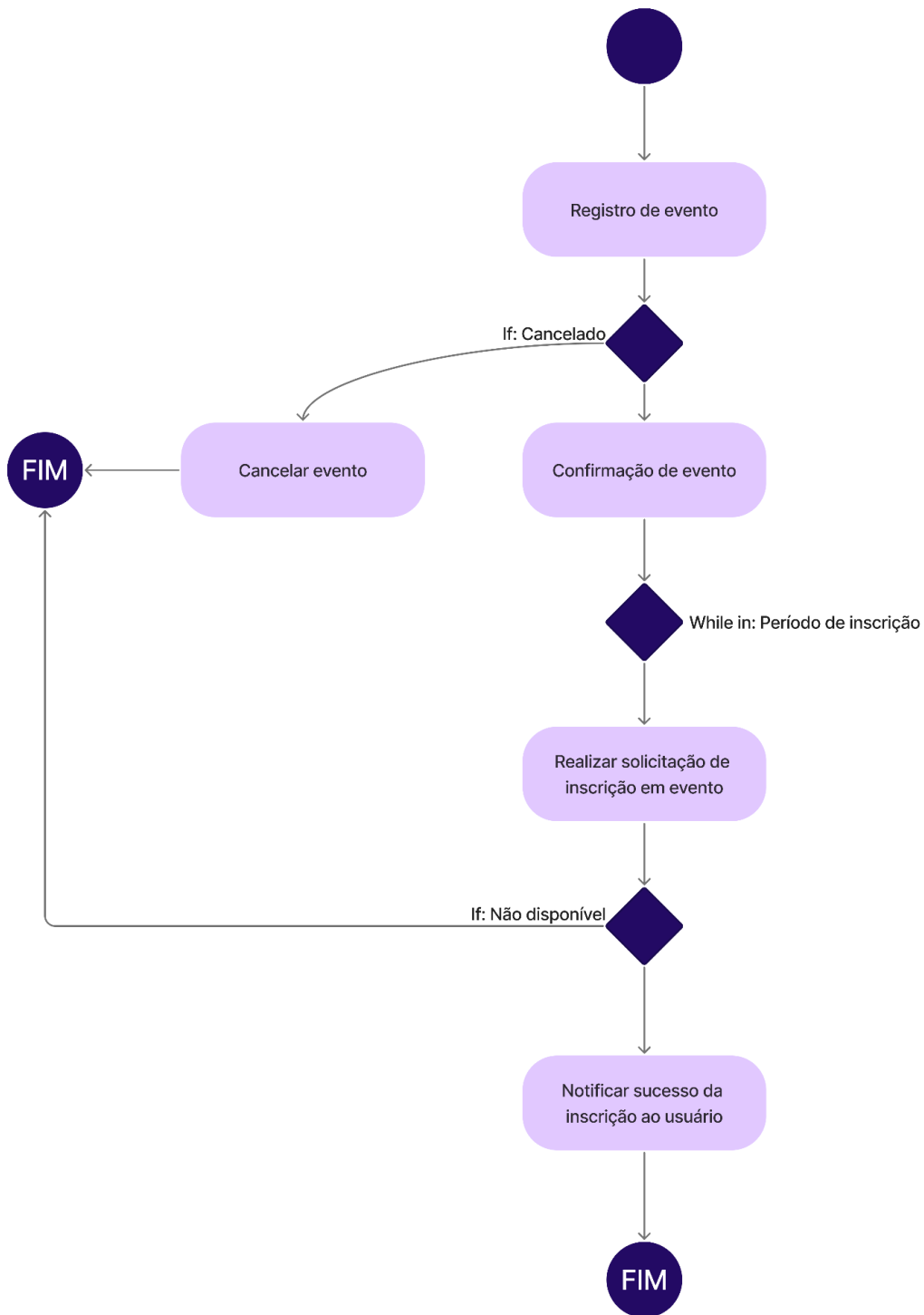
### 3.4 Diagramas de Sequência



### 3.5 Diagramas de Comunicação



### 3.6 Diagramas de Estados



## 4. Modelos de Dados

