Documentação de Projeto

para o sistema

GoTicket

Versão 1.0

Projeto de sistema elaborado pelo aluno Guilherme de Almeida Rocha Vieira
como parte da disciplina **Projeto de Software**.

27/10/2025

Tabela de Conteúdo

1. Introdução **1**
2. Modelos de Usuário e Requisitos **1**
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
3. Modelos de Projeto **1**
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
4. Modelos de Dados **2**

Histórico de Revisões

Nome	**Data**	**Razões para Mudança**	**Versão**
Guilherme de Almeida Rocha Vieira	08/11/2025	Criação da versão inicial do documento. Definição de atores, casos de uso e geração de todos os diagramas de modelagem (UML, ERD).	1.0

1. Introdução

Este documento apresenta a documentação de projeto de software para o sistema GoTicket. O objetivo deste projeto é modelar e detalhar a concepção de uma plataforma online para a venda e gerenciamento de ingressos para eventos. O GoTicket atua como um marketplace, conectando **Organizadores de Evento**, que desejam publicar e vender ingressos, e **Clientes**, que buscam uma maneira segura e centralizada de descobrir e adquirir ingressos. O sistema também inclui um perfil de **Administrador** para garantir a moderação e o bom funcionamento da plataforma. Ao longo das seções a seguir, serão apresentados os modelos que definem o sistema, incluindo a análise de requisitos e usuários (Modelos de Usuário), o design da arquitetura de microsserviços (Modelos de Projeto), e o esquema de banco de dados (Modelos de Dados). Este documento serve como o guia central para o desenvolvimento e a manutenção do sistema.

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

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

- Cliente:

  Descrição: O Cliente é o usuário final que utiliza a plataforma para interagir com os eventos. Seu objetivo principal é descobrir eventos de seu interesse, adquirir ingressos de forma segura e gerenciar os ingressos comprados (como visualizar ou transferir).

- Organizador do Evento:

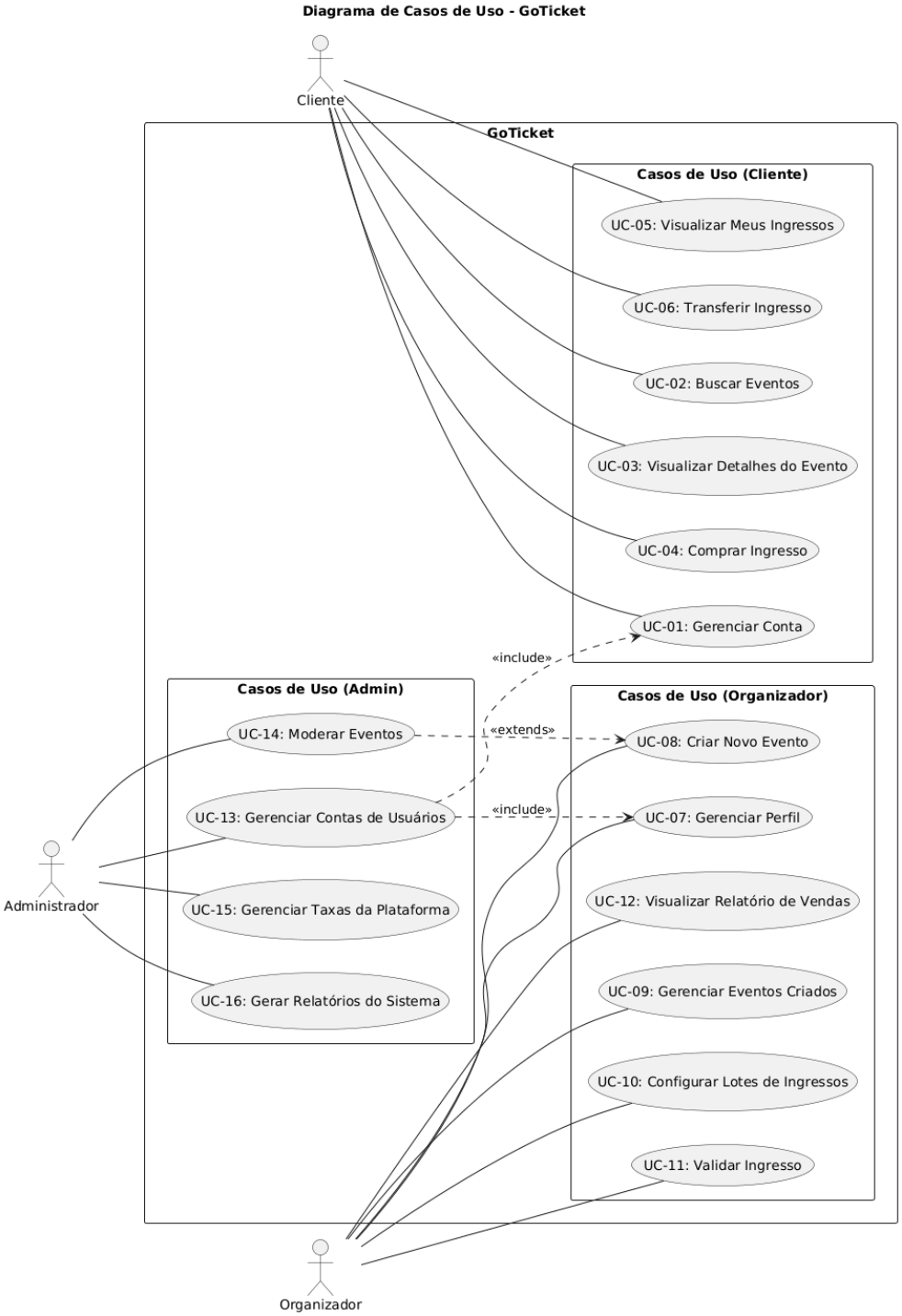
  Descrição: O Organizador é o usuário ou entidade responsável por criar, publicar e gerenciar os eventos na plataforma GoTicket. Ele define os detalhes do evento (data, local, descrição), configura os tipos de ingressos (preços, lotes, quantidade) e acompanha as vendas. Este ator também utiliza o sistema para validar os ingressos na entrada do evento.

- Administrador da Plataforma:

  Descrição: O Administrador é um funcionário da GoTicket com privilégios elevados. Ele é responsável pela manutenção geral do sistema, gerenciamento de contas de usuários (Clientes e Organizadores), moderação de eventos publicados, configuração de taxas de serviço e resolução de disputas ou problemas técnicos.

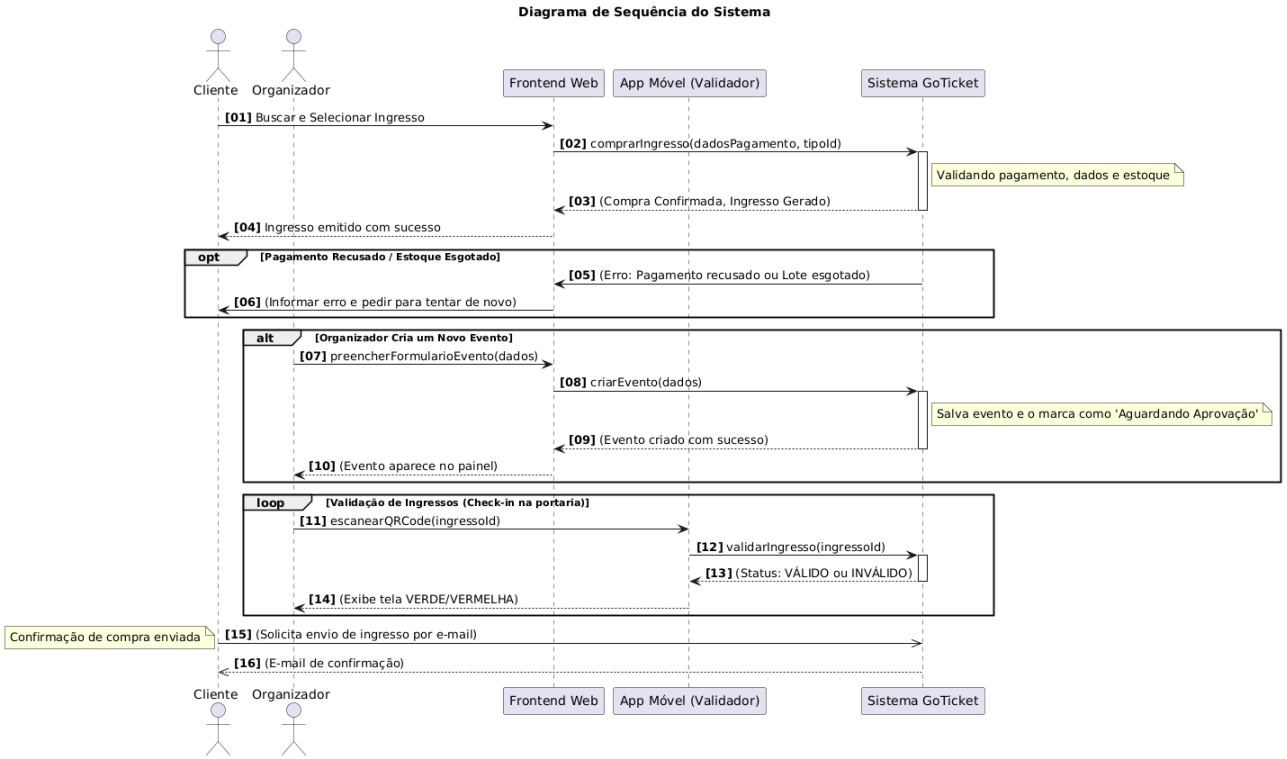
2.2 Modelo de Casos de Uso

Nesta subseção é apresentado o diagrama de casos de uso do sistema. Para cada um deles, utilize um ID que possa servir de referência no restante do documento. Por exemplo UC-01 para o Caso de Uso 01.



2.3 Diagrama de Sequência do Sistema

Nesta subseção é apresentado o diagrama de sequência do sistema de pelo menos, 3 Casos de Uso ou Histórias de Usuário descritos na Seção 2.3.



Contrato – Comprar Ingresso (UC-04)

Contrato	Comprar Ingresso
Operação	comprarIngresso(tipoIngressoID, quantidade, dadosPagamento)
Referências cruzadas	UC-04: Comprar Ingresso
Pré-condições	1. O cliente deve estar autenticado no sistema 2. O "tipoIngressoID" deve existir e pertencer a um Evento que esteja com evento **PUBLICADO** 3. A quantidade solicitada deve ser menor ou igual à "quantidadeDisponivel" do "TipoIngresso" (Lote)
Pós-condições	1. Uma nova instância de Pedido é criada com o status **PAGO** 2. Uma ou mais instâncias de Ingresso (na quantidade solicitada) são criadas e associadas a este Pedido 3. A "quantidadeDisponivel" do "TipoIngresso" (Lote) é decrementada pela quantidade comprada. 4. A transação de pagamento é registrada (via Gateway de Pagamento externo)

	5. Uma notificação de confirmação (com os ingressos) é enviada ao Cliente.

Contrato – Criar Novo Evento (UC-08)

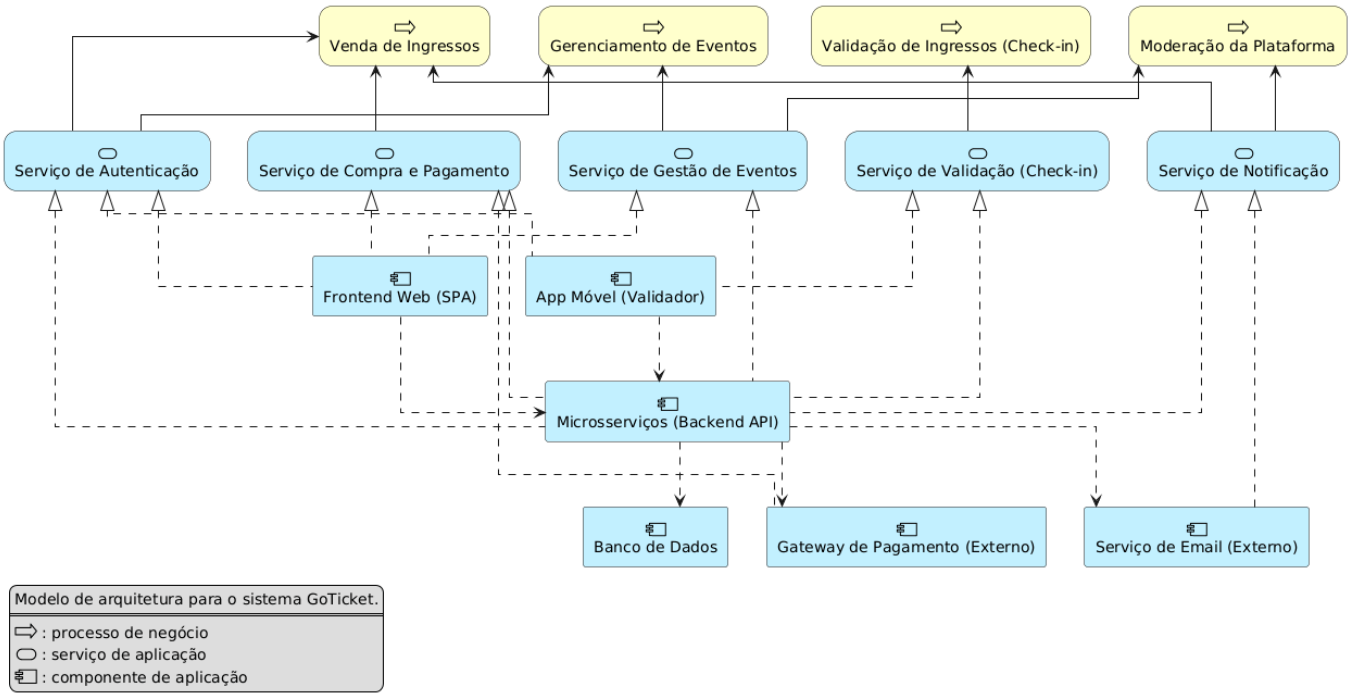
Contrato	Criar Novo Evento
Operação	criarNovoEvento(dadosEvento)
Referências cruzadas	UC-08: Criar Novo Evento
Pré-condições	1. O Usuário deve estar autenticado e possuir a role de Organizador 2. O Usuário (Organizador) deve ter seu perfil de organizador aprovado pelo Admin
Pós-condições	1. Uma nova instância de Evento é criada e persistida no banco de dados 2. O status do novo Evento é definido como "**Aguardando_Aprovacao**" 3. O "organizador_id" do Evento é associado ao id do Usuário (Organizador) que o criou. 4. Uma notificação é enviada ao Administrador da Plataforma para moderação (relacionada ao UC-14).

Contrato – Validar Ingresso (UC-11)

Contrato	Validar Ingresso (Check-in)
Operação	validarIngresso(ingressoID)
Referências cruzadas	UC-11: Validar Ingresso (Check-in)
Pré-condições	1. O Usuário deve estar autenticado no App Móvel (Validador) com permissões de Organizador 2. O "ingressoID" (QR Code) foi lido e fornecido à operação
Pós-condições	1. A instância de Ingresso correspondente ao "ingressoID" é recuperada 2. Se o Ingresso existir e seu status for **Pago** (ou **Disponível** para uso): a. O status do Ingresso é alterada para **Utilizado** b. Uma resposta do sucesso (Check-in OK) é retornada 3. Se o Ingresso existir e seu status já for **Utilizado** ou **Cancelado**: a. Uma resposta de falha ("Ingresso já utilizado") é retornada. 4. Se o Ingresso não for encontrado no banco de dados: a. Uma resposta de falha é retornada

3. Modelos de Projeto

3.1 Arquitetura



A arquitetura do sistema **GoTicket** foi projetada seguindo uma abordagem moderna de **Microsserviços baseada em Nuvem** (Cloud-Native). Esta escolha visa garantir alta escalabilidade, manutenibilidade e resiliência, permitindo que diferentes partes do sistema evoluam de forma independente.

O sistema é logicamente dividido nas seguintes camadas principais:

1. **Camada de Cliente (Frontend):** Composta por aplicações cliente que consomem os dados, incluindo uma **SPA (Single Page Application)** para a web (usada tanto por Clientes quanto por Organizadores) e um **Aplicativo Móvel** focado na validação de ingressos (check-in).

2. **Camada de Borda (Edge Layer):** Atua como o ponto de entrada único para todas as requisições. Utiliza um **Load Balancer** para distribuir o tráfego e um **Servidor Web (CDN)** para entregar o conteúdo estático (o Frontend) com baixa latência.

3. **Camada de Aplicação (Backend):** Onde reside a lógica de negócio. É composta por um **API Gateway**, que roteia as requisições para os microsserviços adequados. Os serviços principais incluem:

    a. Serviço de Usuários: Gerencia autenticação e perfis.

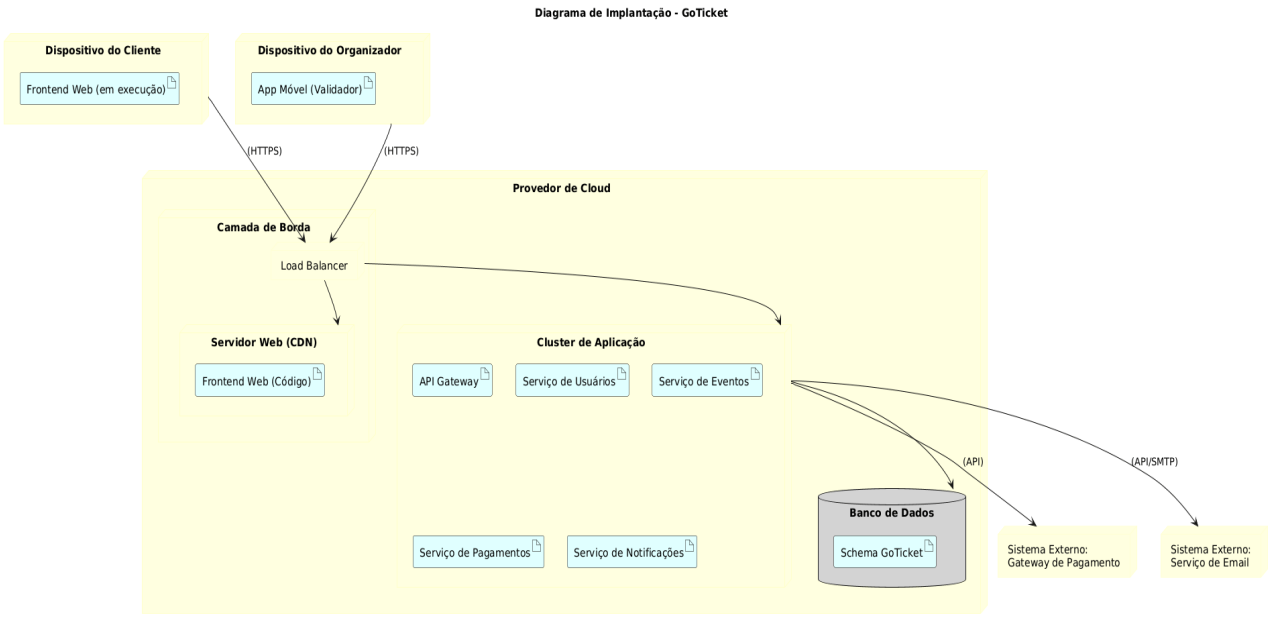
b. Serviço de Eventos: Gerencia o ciclo de vida dos eventos, lotes (Tipos de Ingresso) e o processo de check-in (validação).
c. Serviço de Pagamentos: Orquestra a compra, cria Pedidos e se comunica com o sistema externo.
d. Serviço de Notificações: Envia e-mails (ex: confirmação de compra, evento novo para moderação).

4. **Camada de Persistência (Dados):** Utiliza um **Banco de Dados** relacional centralizado (ex: PostgreSQL), que armazena os dados de todos os microsserviços.
5. **Serviços Externos:** O sistema se integra a serviços de terceiros essenciais para sua operação: um **Gateway de Pagamento** (para processar as transações financeiras) e um **Serviço de Email** (para disparar notificações).

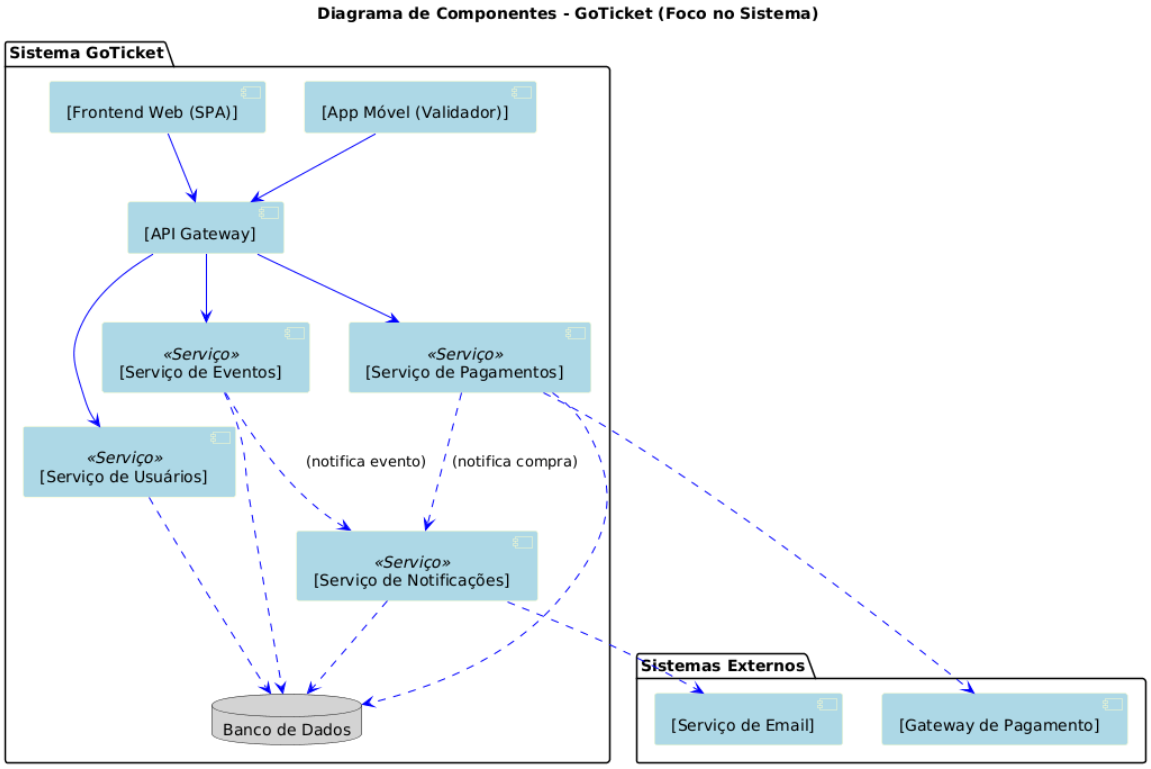
3.2 Diagrama de Componentes e Implantação.

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.

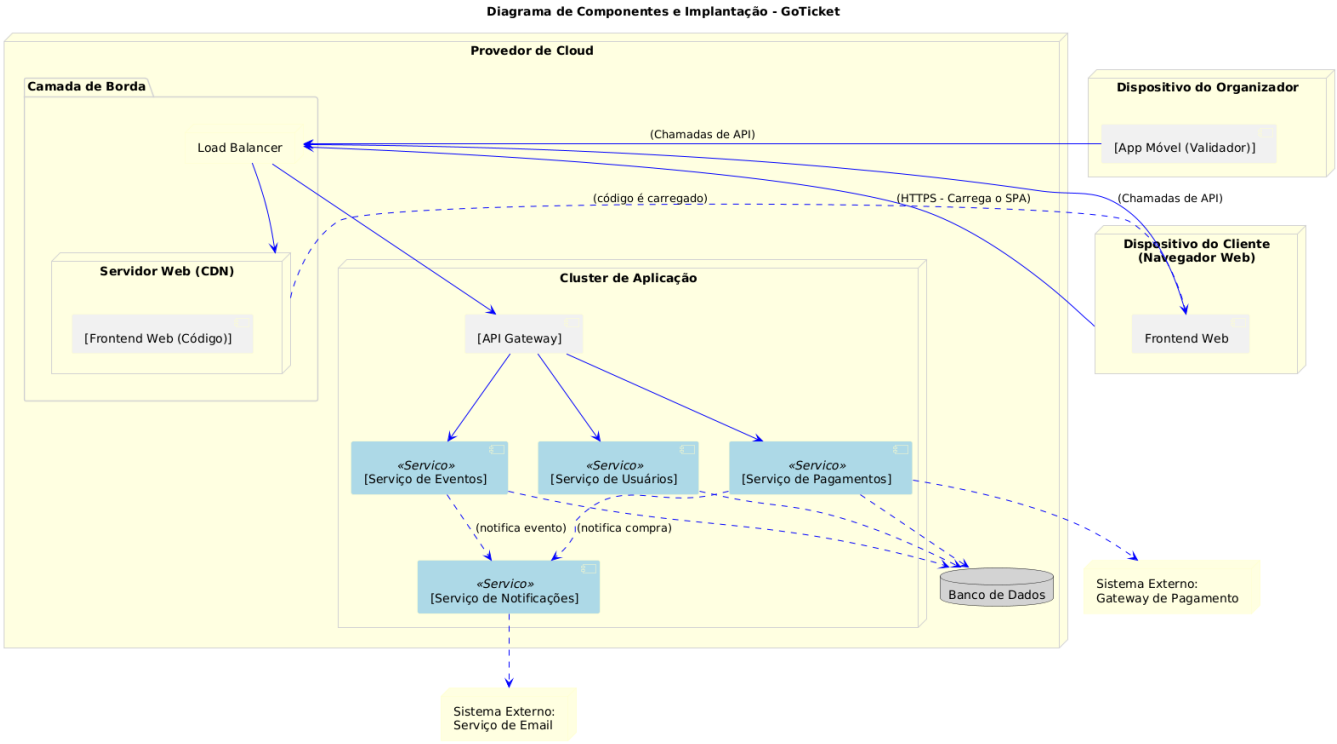
- Diagrama de Implantação



- Diagrama de Componente

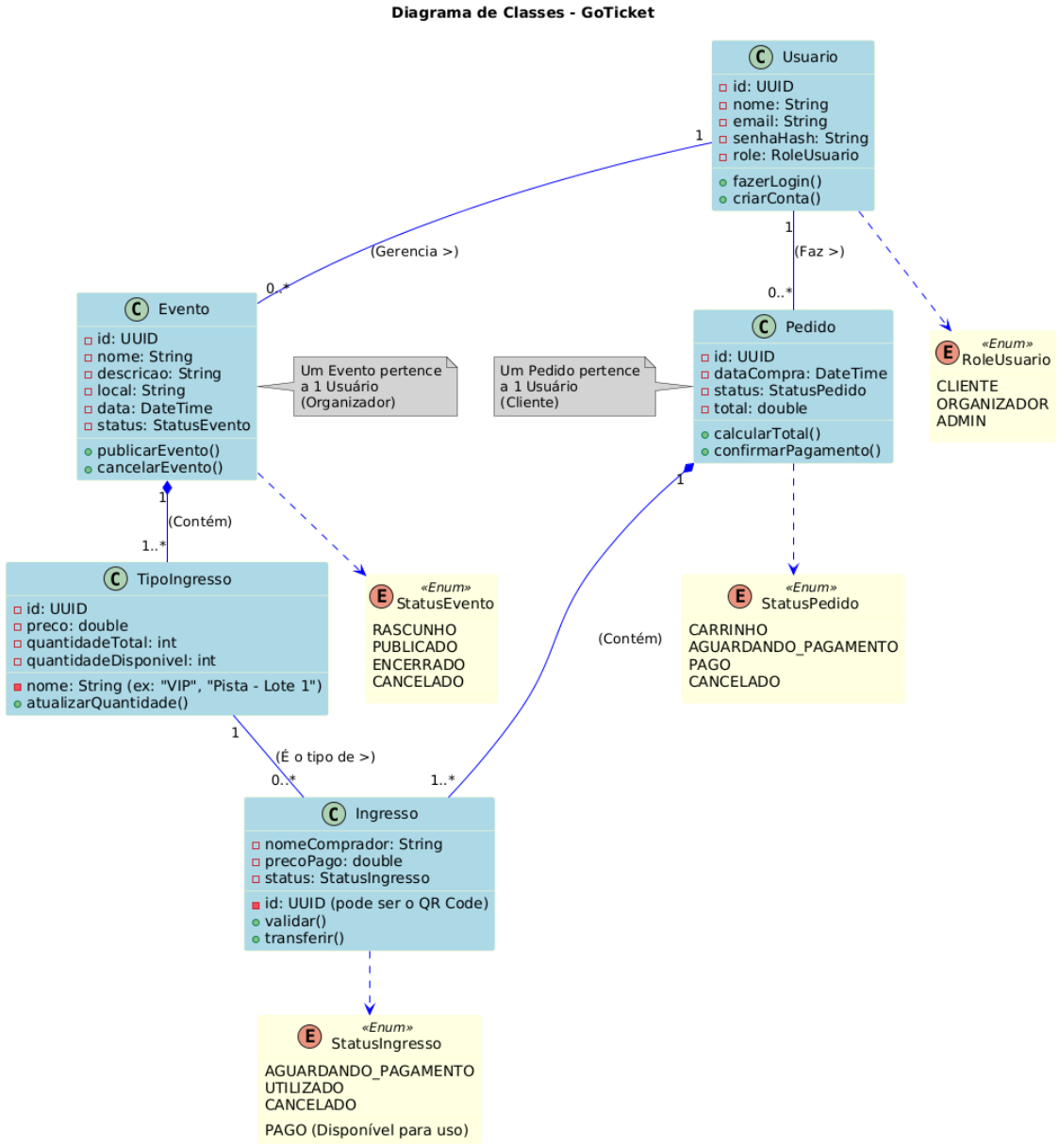


- Diagrama de Componentes e Implantação



3.3 Diagrama de Classes

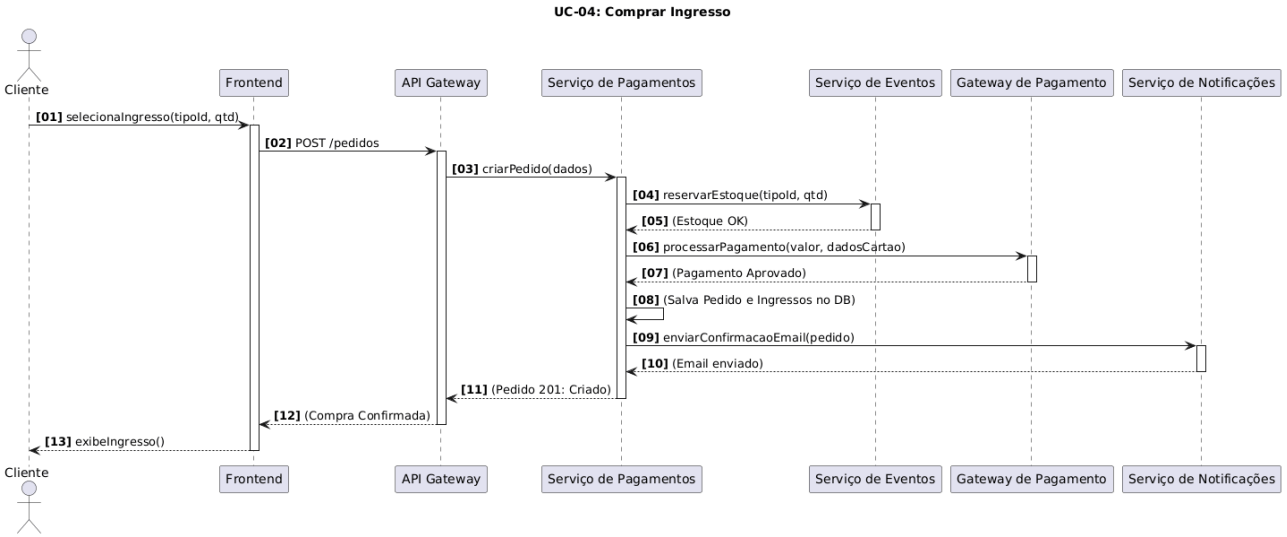
Diagrama de classes do sistema.



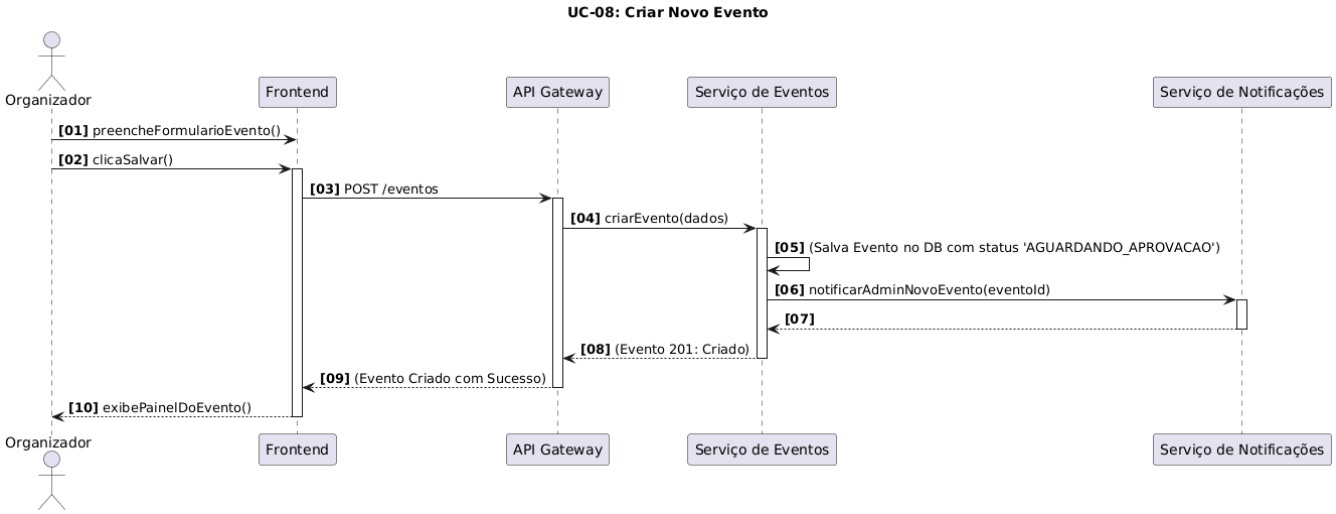
3.4 Diagramas de Sequência

Diagramas de sequência para realização de casos de uso.

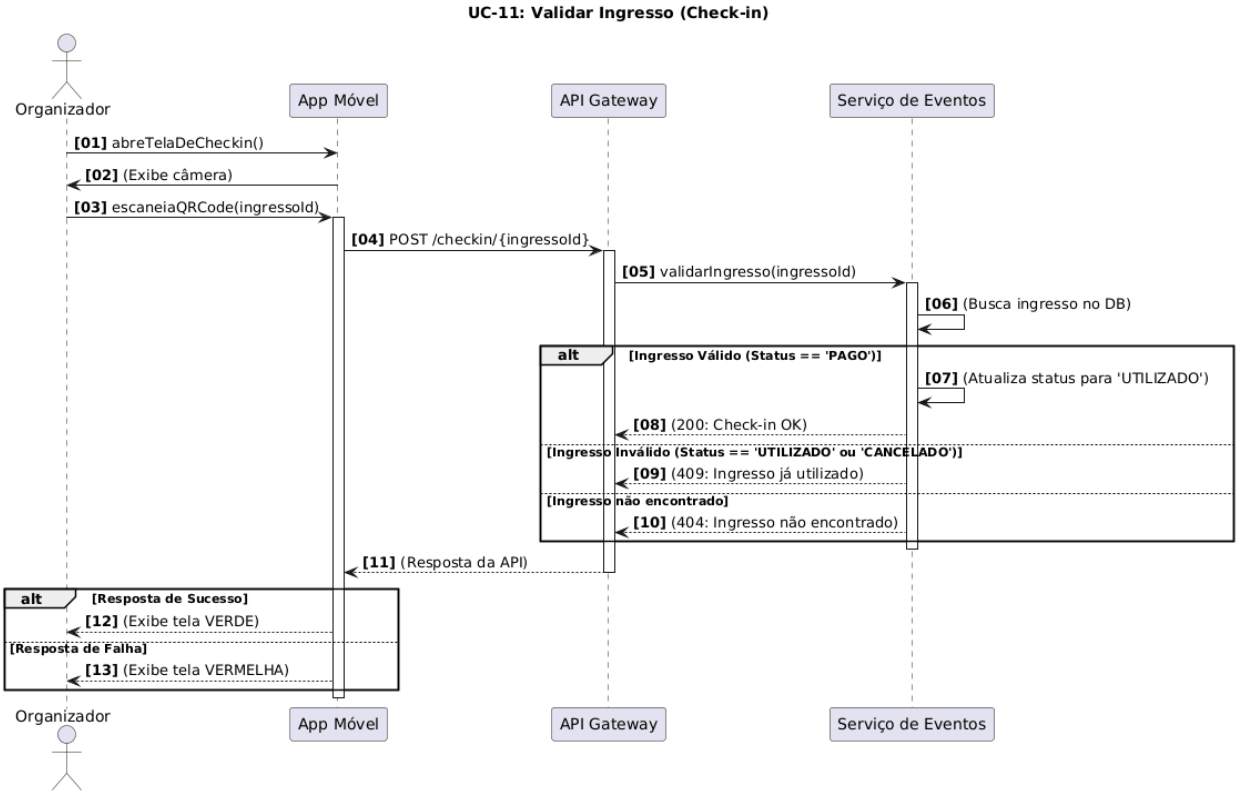
- UC-04



- UC-08



- UC-11

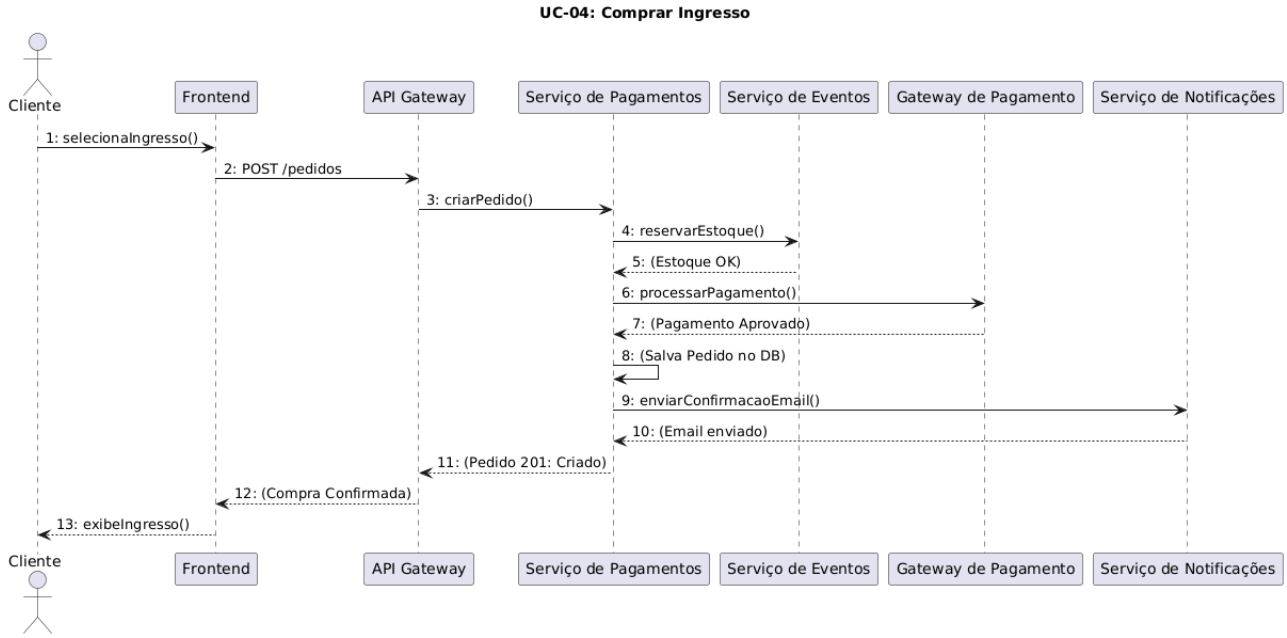

UC-11: Validar Ingresso (Check-in)
Organizador
App Móvel
API Gateway
Serviço de Eventos
[01] abreTelaDeCheckin()
[02] (Exibe câmera)
[03] escaneiaQRCode(ingressoId)
[04] POST /checkin/{ingressoId}
[05] validarIngresso(ingressoId)
[06] (Busca ingresso no DB)
alt
[Ingresso Válido (Status == 'PAGO')]
[07] (Atualiza status para 'UTILIZADO')
[08] (200: Check-in OK)
[Ingresso Inválido (Status == 'UTILIZADO' ou 'CANCELADO')]
[09] (409: Ingresso já utilizado)
[Ingresso não encontrado]
[10] (404: Ingresso não encontrado)
[11] (Resposta da API)
alt
[Resposta de Sucesso]
[12] (Exibe tela VERDE)
[Resposta de Falha]
[13] (Exibe tela VERMELHA)
Organizador
App Móvel
API Gateway
Serviço de Eventos

3.5 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.

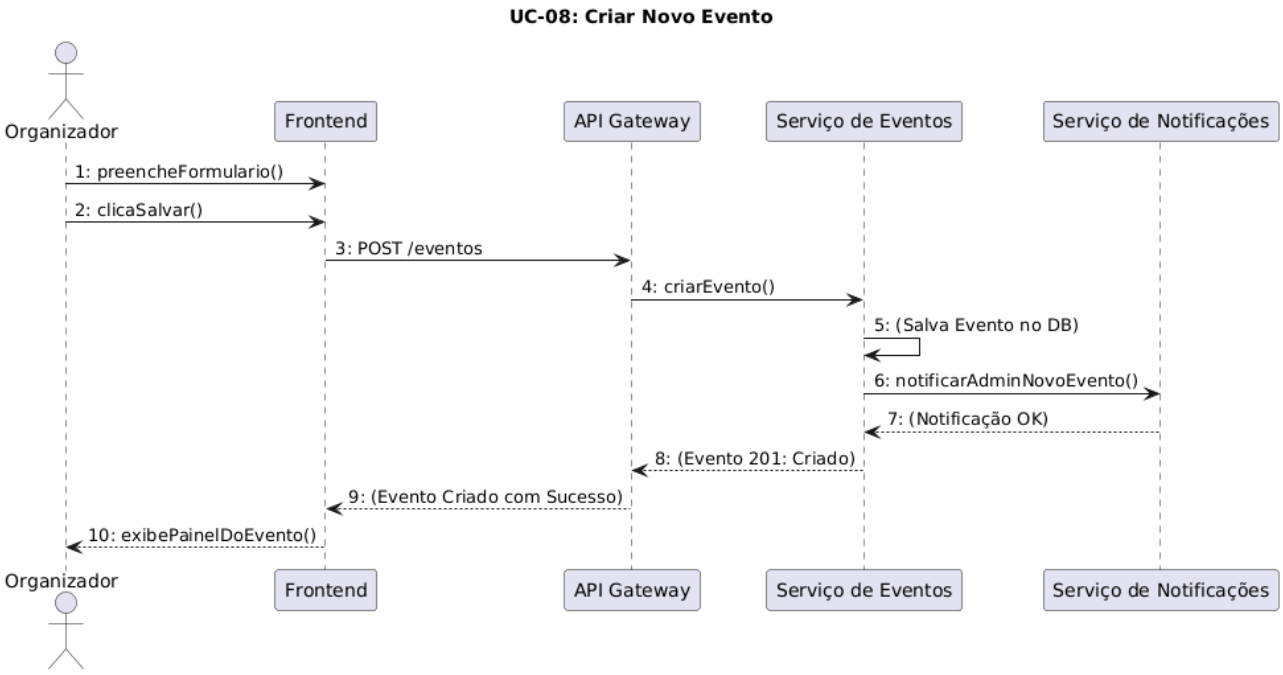
1. UC-04: Comprar Ingresso

   O diagrama de comunicação a seguir ilustra as interações de objetos necessárias para o caso de uso **"Comprar Ingresso"**. Ele demonstra como o Serviço de Pagamentos atua como o orquestrador central, comunicando-se tanto com o Serviço de Eventos para verificar o estoque quanto com o Gateway de Pagamento externo para processar a transação financeira.



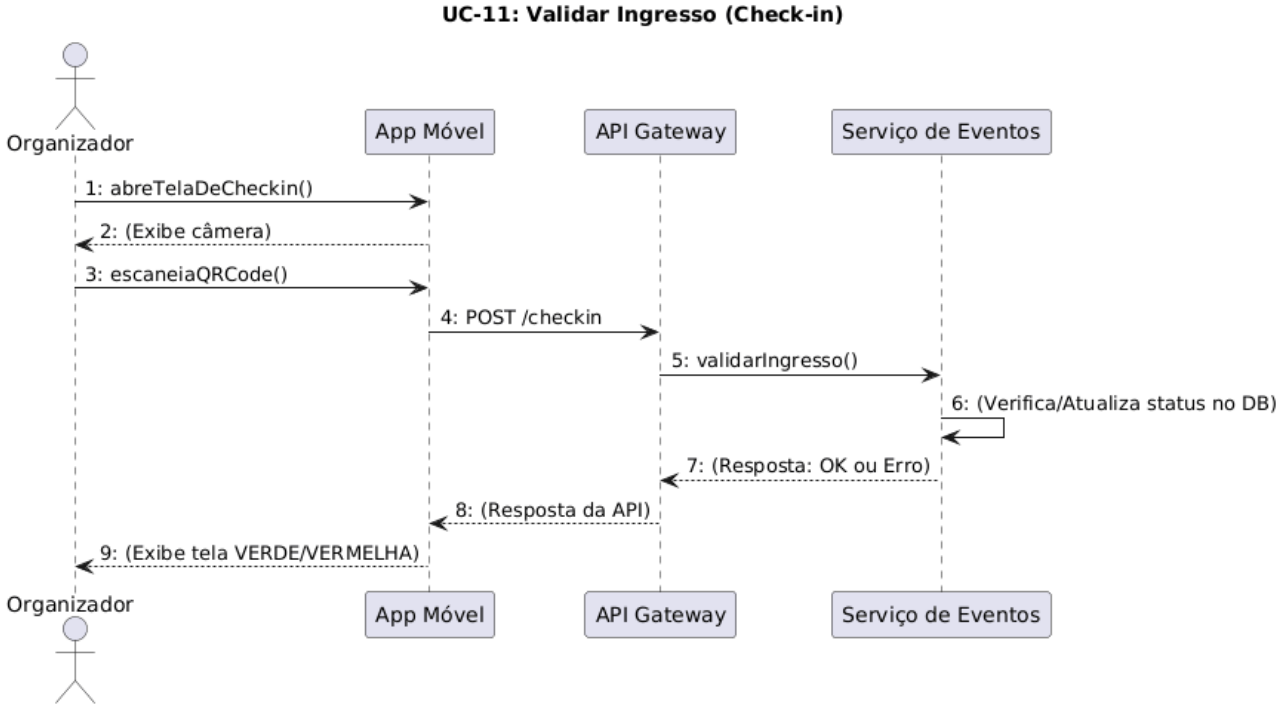
2. UC-08: Criar Novo Evento:

   Este diagrama detalha as mensagens trocadas quando um Organizador cadastra um novo evento. O fluxo é iniciado pelo Frontend e processado pelo Serviço de Eventos, que, após salvar o novo evento no banco de dados, estabelece comunicação com o Serviço de Notificações para alertar os administradores sobre a necessidade de moderação.



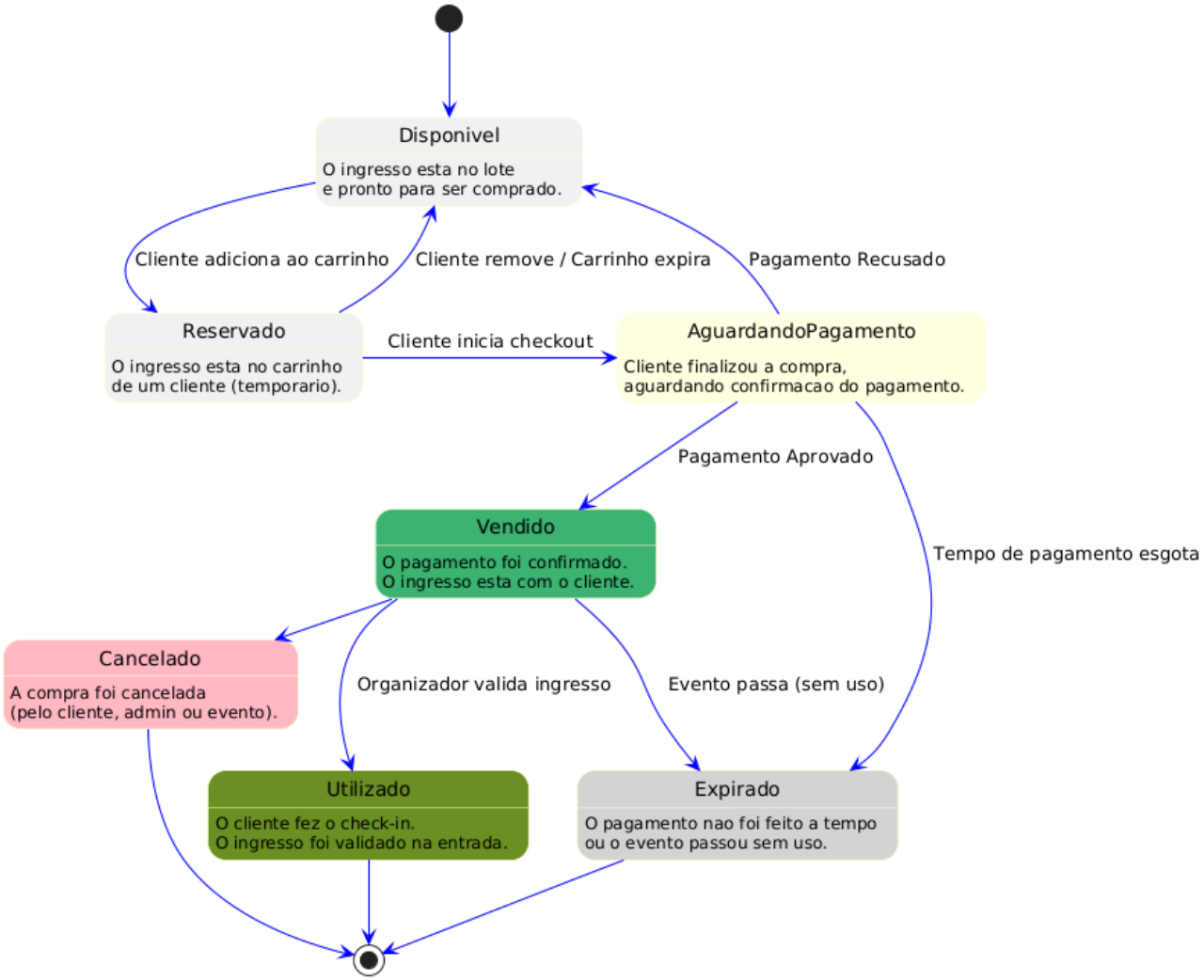
4. UC-11: Validar Ingresso (Check-in)

O diagrama abaixo modela a comunicação para o caso de uso crítico de **"Validar Ingresso"**. O foco é a interação em tempo real entre o App Móvel do organizador e o Serviço de Eventos. O diagrama mostra a sequência numerada de mensagens, desde o escaneamento do QR Code até a verificação e atualização do status do ingresso no sistema.



4.1 Diagramas de Estados

Diagramas de estados do sistema.



5. Modelos de Dados

Deve-se apresentar os esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.

Modelo de Dados (Esquema de Banco de Dados) - GoTicket