

1. Desenho Macro da Solução

a) Recursos de Infraestrutura

- **i. Base de Dados**
 - Tecnologia principal: Relacional Escalável - PostgreSQL
 - Detalhes:
 - **Ambiente:** Produção e Desenvolvimento
 - **Servidores necessários:** Utilize balanceadores de carga para distribuir o tráfego entre instâncias (AWS).
 - **ii. Recursos de Computação (Serverless/Containers)**
 - Tipos de serviços: Apache Spark, *AWS Lambda*, *Docker Containers*
 - Detalhes:
 - Processamento 8 vCPU's
 - RAM 16GB
 - Instâncias otimizadas para IO e memória.
 - SSDs NVMe para latência reduzida.
 - Escalabilidade Load Balancer
 - **iii. Recursos Adicionais de Terceiros**
 - Serviços como:
 - SQS (Simple Queue Service)
 - SNS (Simple Notification Service)
 - Detalhes sobre integrações e volumes esperados
ara lidar com um cenário de **50 mil acessos e 15 mil inserções** por segundo, com picos de **40 mil inserções** por segundo, o design da infraestrutura precisa ser robusto e escalável, especialmente considerando que o sistema gera processos de **ETL (Extract, Transform, Load)**.
 - Grande volumes de mensageria com necessidade de acompanhamento de logs.
 - Canal de Comunicação e resposta para gestão com automação.
-

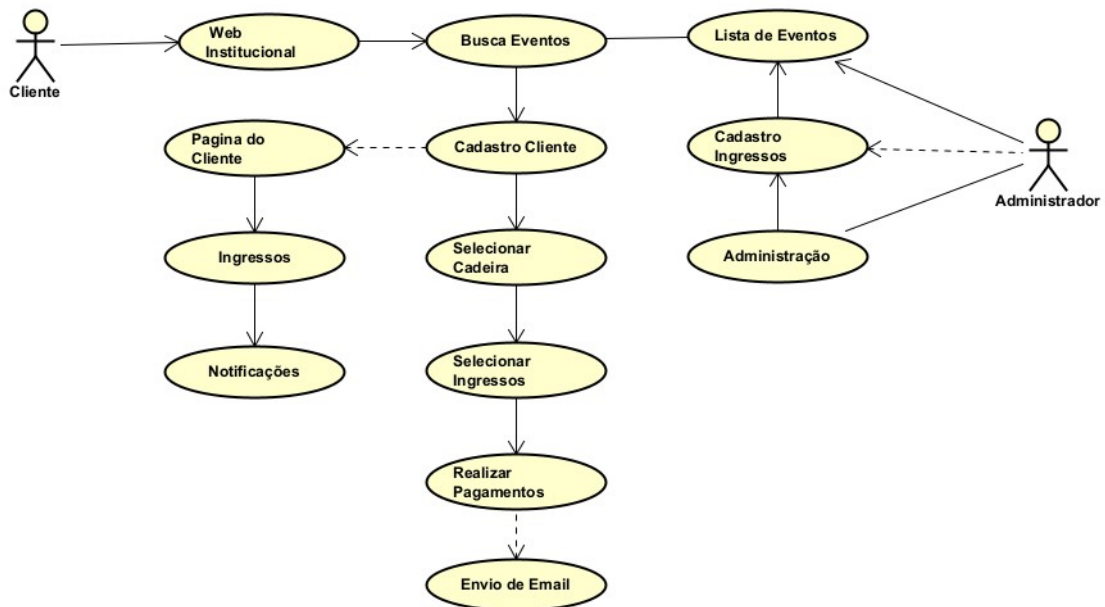
2. Desenho Micro da Solução

a) Recursos Técnicos

- **i. Frameworks**
 - *Spring Boot, Angular*
- **ii. Linguagens**
 - *Java 21, TypeScript*
- **iii. Ferramentas**
 - *Git, Pipelane, CD/CI*

Escopo e requisitos macro:

Uma plataforma web que permite comprar ingressos.



Todos os ingressos são cadastrados por um administrador.

Solução: Lista de Importação de dados de cadastro em lote para importação de arquivos.

O sistema deve garantir a idempotência das operações.

Solução: Tratamento de Sessões e fluxo de dados com painel do usuário para acompanhar e retirada ingressos. Autenticação por QR Codes para validações na plataforma.

Um usuário pode comprar ingressos via paypal/stripe.

1. Integração com PayPal/Stripe

1.1 Stripe

- API RESTful
- APIs RESTful do Stripe para operações como criação de pagamentos, gerenciamento de assinaturas e devoluções.
- Webhooks para capturar eventos, como confirmações de pagamento.
- Conexões Persistentes
- HTTP/2 para manter conexões persistentes e reduzir a latência em chamadas frequentes.
- Idempotência
- Envie cabeçalhos de idempotência (Idempotency-Key) para evitar pagamentos duplicados em caso de repetição de requisições.

1.2 PayPal

- REST API ou SDKs
- A REST API do PayPal suporta pagamentos simples, assinaturas e operações de checkout.
- O SDK oficial do PayPal está disponível em várias linguagens e otimiza integrações.
- Smart Payment Buttons
- Para UX, utilize os Smart Payment Buttons, que suportam múltiplos métodos de pagamento.
- Webhooks
- Configure notificações para eventos, como status de pagamento, cancelamentos ou disputas.

2.1 Autenticação e Chaves

- Stripe: Utilize as chaves API (pública e privada) para todas as chamadas.
- PayPal: Use OAuth 2.0 para autenticação em vez de integrar diretamente com credenciais de conta.

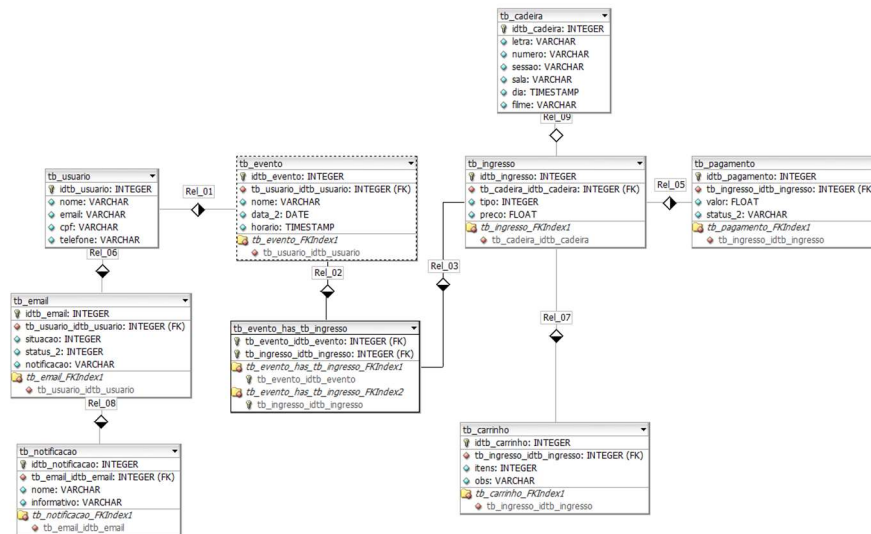
O sistema deve garantir que dois usuários não comprem o mesmo ingresso.

Solução: Garantir integridade por CPF

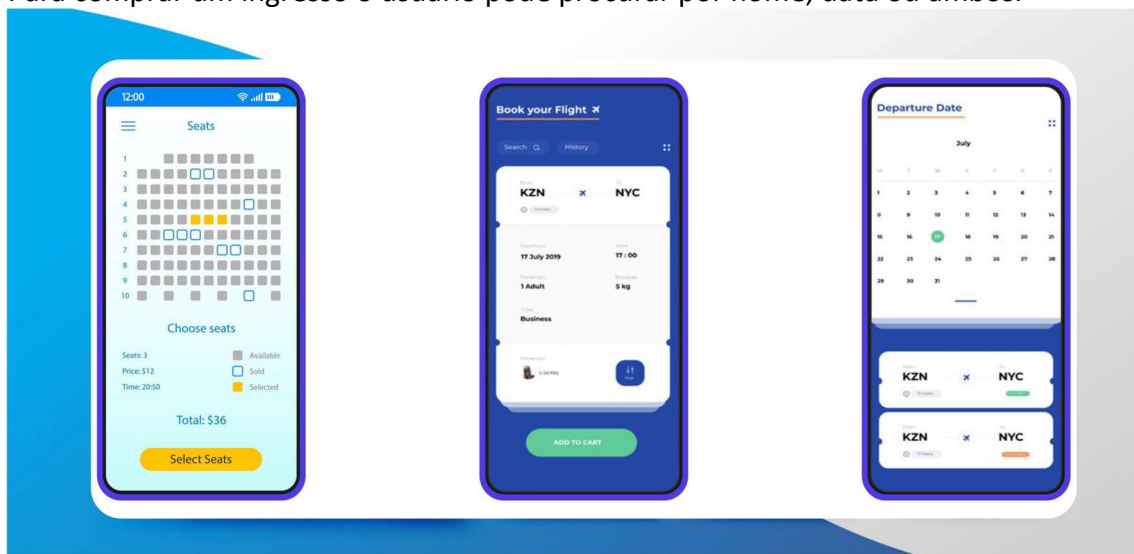
Cada usuário pode comprar até 10 ingressos da mesma sessão.

Solução: Validação por CPF

Um ingresso deve registrar: cadeira, sessão, sala, dia, horário e filme.



Para comprar um ingresso o usuário pode procurar por nome, data ou ambos.



O sistema precisa garantir uma atualização realtime dos ingressos disponíveis.

Solução: Criar Fila por sessão com data hora e *Precadastro e envio por email*

O usuário deve selecionar a cadeira que vai ocupar na sala.

Solução: Interface com atualização automatica e validação na seleção das cadeiras

