STRA DA

Uma Alternativa Segura e Sustentável para a Mobilidade Estudantil





Recapitulando...

O Strada é um aplicativo desenvolvido para conectar pais e alunos que oferecem ou solicitam caronas, garantindo segurança, praticidade e redução do trânsito.

- 01 -

Arquitetura Integrada e Colaborativa do Sistema

O Strada foi desenvolvido com uma arquitetura modular orientada a serviços, integrando funcionalidades distribuídas como autenticação, rotas, caronas, avaliações e notificações.

A comunicação entre os serviços se dá via gRPC e Apache Kafka, garantindo robustez e escalabilidade.

A integração entre o app mobile, o banco de dados e os serviços backend permite que todos os processos funcionem de forma contínua, desde o cadastro de usuários (RF-001) até o monitoramento de viagens e histórico (RF-006, RF-011).

· 02 ·

Requisitos Funcionais Implementados

O desenvolvimento do sistema Strada contemplou os requisitos funcionais planejados, priorizando inicialmente os de alta prioridade para garantir a base crítica de operação da aplicação.

Requisitos de Alta Prioridade:

RF-001: Cadastro unificado de usuários com associação de papéis (Responsável e Motorista)

RF-002: Validação de segurança do motorista por meio do upload de CNH

RF-003: Cadastro de alunos e vinculação ao perfil do responsável

RF-004: Criação e gerenciamento de rotas pelos motoristas

RF-005: Busca inteligente de rotas compatíveis com base em localização e horários

RF-006: Agendamento de caronas por parte dos responsáveis

RF-007: Edição de perfil por parte dos usuários

Requisitos de Média e Baixa Prioridade:

RF-008: Avaliações e feedbacks entre motoristas e responsáveis após cada carona

RF-009: Visualização de histórico de viagens realizadas pelo usuário

- 03 -

Modelagem de Dados com Ênfase em Mobilidade e Segurança O banco de dados foi projetado para atender às necessidades específicas do transporte geolocalizado e seguro.

A base relacional utiliza PostgreSQL com extensão PostGIS, viabilizando:

- Análise espacial e filtros por raio para busca de rotas (RF-005)
- Armazenamento eficiente de coordenadas e trajetos
- Além disso, serviços que demandam dados não relacionais, utilizam MongoDB,

- 04 Interfaces Completas com Experiência Funcional

O aplicativo Strada oferece uma experiência amigável e fluida ao usuário, com todas as telas fundamentais operacionais:

- Onboarding e autenticação segura
- Cadastro de rotas e alunos (RF-001 a RF-004)
- Solicitação e agendamento de caronas com filtros e critérios inteligentes (RF-005, RF-006)
- Tela de histórico de caronas e sistema de avaliações (RF-011, RF-009)

Todos os CRUDs implementam validações de campos, feedback visual e navegação fluida. A camada de frontend em React Native interage diretamente com os endpoints.

- 05 - Implantação em Nuvem

O sistema Strada está implantado na Google Cloud Platform (GCP) com uso de Kubernetes para orquestração dos serviços. Todos os serviços estão containerizados com Docker e gerenciados no cluster Kubernetes

Utilizamos Nginx como proxy reverso para roteamento das requisições

Deploy automatizado com GitHub Actions garante agilidade nas entregas

Acesso aos serviços:

Autenticação: https://auth.strada.appbr.store

Gerenciamento de caronas: https://ride.strada.appbr.store

- 06 -

Segurança no Armazenamento de Senhas

Visando a proteção de senhas, implementamos:

- Criptografia de senhas com bcrypt, um dos algoritmos mais seguros e amplamente adotados no mercado.
- As senhas nunca são armazenadas em texto claro.
- A autenticação utiliza tokens JWT, com tempo de expiração curto e renovação via refresh tokens, criando uma camada extra de segurança.

Essa abordagem garante a confidencialidade e a integridade dos dados sensíveis.

• 07 • Telas e Operações CRUD

Atualmente todas as telas e operações CRUD estão implementadas e operacionais, permitindo o fluxo completo do sistema conforme o escopo inicial.

Naturalmente, como em qualquer software em evolução, o monitoramento e pequenos ajustes contínuos são parte do ciclo de melhorias, mas as funcionalidades principais estão disponíveis e funcionando.

• 08 • Prevenção Contra SQL Injection

Nosso sistema adota práticas rigorosas contra vulnerabilidades conhecidas, como o SQL Injection. Utilizamos a ORM Prisma, que gera prepared statements automaticamente.

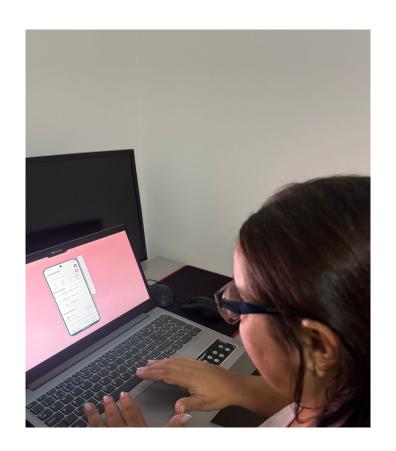
As consultas ao banco são feitas por meio de bindings seguros e sanitização de dados, prevenindo qualquer injeção maliciosa.

Toda a camada de dados segue as boas práticas de segurança da OWASP.

Essa proteção está integrada de forma nativa no código, garantindo robustez e confiabilidade.

- 09 -

Apresentação do projeto a empresa parceira



Todo o código-fonte do sistema Strada, bem como a documentação técnica, está disponível publicamente em nosso repositório no GitHub:



https://github.com/ICEI-PUC-Minas-PBE-ADS-SI/2025-1-p5-tias-caron a-escolar

- OBRIGADO(A)!

