

**GRUPO 06**

**INTEGRANTES**

Guilherme Kaludin

João Oliveira

Juliana Monteiro

Lucas Daniel

Tainá Lara

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema: Aplicativo de caronas compartilhadas**



**GRUPO 06**

**INTEGRANTES**

Guilherme Kaludin

João Oliveira

Juliana Monteiro

Lucas Daniel

Tainá Lara

**PROJETO INTEGRADOR**

**ESCOPO DO PROJETO**

**Tema: Aplicativo de caronas compartilhadas**

Relatório solicitado pela Generation Brasil para compor o projeto final.O relatório refere-se ao escopo do projeto integrador.

1. **Título do Projeto e Modelo de Negócio escolhido**

Carona Legal.

Uma plataforma de gerenciamento de corridas, onde usuários podem solicitar corridas e motoristas podem realizar essas viagens.

1. **Descrição Geral do Projeto**

O aplicativo Carona Legal é uma solução fullstack desenvolvida para gerenciar corridas entre usuários e motoristas de forma eficiente e segura. No backend, a aplicação foi construída com Java e Spring Boot, garantindo robustez e escalabilidade, e utiliza MySQL como banco de dados para armazenar as informações de maneira estruturada e confiável. No frontend, foi utilizado React com TypeScript, proporcionando uma interface dinâmica e tipagem segura, além de Tailwind CSS para estilização, garantindo um design moderno, responsivo e agradável ao usuário. Essa arquitetura permite um sistema eficiente, intuitivo e de fácil manutenção, oferecendo uma experiência fluida tanto para motoristas quanto para passageiros.

1. **Descrição da Entidade/Model foi criada e seus atributos**

Criamos as Model’s Usuário, Corrida e Motorista.

(tb\_usuario) com os atributos id, nome, email e telefone.

(tb\_corrida) com os atributos id, distancia, velocidade, tempoCorrida, partida, destino e FK id usuario e id motorista.

(tb\_motorista) com os atributos id, nome e tipoVeiculo.

1. **Funcionalidades Principais (CRUD) implementadas**

Criamos as seguintes funcionalidades:

Método utilitário para formatar o tempo em hh:mm:ss

formatarTempo()

findAll()

findById()

findAllByNome()

post()

put()

delete()

1. **Tecnologias Utilizadas (banco de dados, backend e frontend)**

Java 17+

MySql Driver

Spring Boot DevTools

Spring Web

Validation

Spring Data JPA

React

TypeScript

JavaScript

Axios

Tailwind

HTML

CSS

React-dom-router

React-loader-spinner

Reactjs-popup

Phosphor Icon

HeroIcons