

# Briefing do Projeto CarrinGo

## Nome do Projeto

CarrinGo

## Resumo da Ideia

CarrinGo é uma aplicação web de e-commerce com foco em experiência de compra leve e funcional. A proposta é oferecer uma interface intuitiva para clientes visualizarem produtos, adicionarem ao carrinho e finalizarem a compra. O sistema também possui uma área administrativa para cadastro de novos produtos com autenticação protegida.

## Objetivos

- Desenvolver um e-commerce funcional com front-end e back-end integrados
- Implementar funcionalidades completas de compra: visualização de produtos, adição ao carrinho, checkout e confirmação
- Criar uma interface de cadastro de produtos autenticada por token
- Aplicar boas práticas de design, código e colaboração em equipe

## Público-alvo

- Estudantes de programação
- Pequenos empreendedores
- Professores e avaliadores acadêmicos

## Funcionalidades

Cliente:

- Listagem de produtos
- Adição e remoção do carrinho
- Feedback visual
- Filtro por categoria
- Tela de checkout e confirmação de pedido

Administrador:

- Tela protegida para cadastro de novos produtos
- Autenticação simples com token

- Comunicação segura com a API FastAPI

## **Tecnologias Utilizadas**

Front-end:

- HTML, CSS e JavaScript puro
- LocalStorage
- Fetch API

Back-end:

- FastAPI
- Validação de token via cabeçalho
- API RESTful para produtos

## **Identidade Visual (Branding)**

- Nome: CarrinGo
- Slogan: "O seu carrinho, a sua experiência."
- Paleta: Laranja vibrante, cinza escuro e tons neutros
- Tipografia: Inter ou Poppins
- Ícones: Lucide ou Font Awesome
- Estilo: Moderno, limpo, com foco em usabilidade

## **Equipe e Responsabilidades**

Front-end (4 alunos):

- Interface e responsividade
- Acessibilidade e animações
- Integração com API
- Tela de cadastro com autenticação

Back-end (4 alunos):

- Modelagem de dados
- Criação da API RESTful
- Autenticação por token
- Documentação via README.md

## **Infraestrutura**

- Código separado em módulos claros (SPA + API)
- Git e GitHub
- Uso de Kanban (Trello)
- Documentação contínua

## **Possibilidades Futuras**

- Login com JWT
- Painel de pedidos
- Pagamento com API externa
- Hospedagem com Docker + Render/Vercel