



# HortaShop

[hortashop.com](http://hortashop.com)



# HortaShop

**Conectando Produtores Locais aos Consumidores**

Problema identificado:

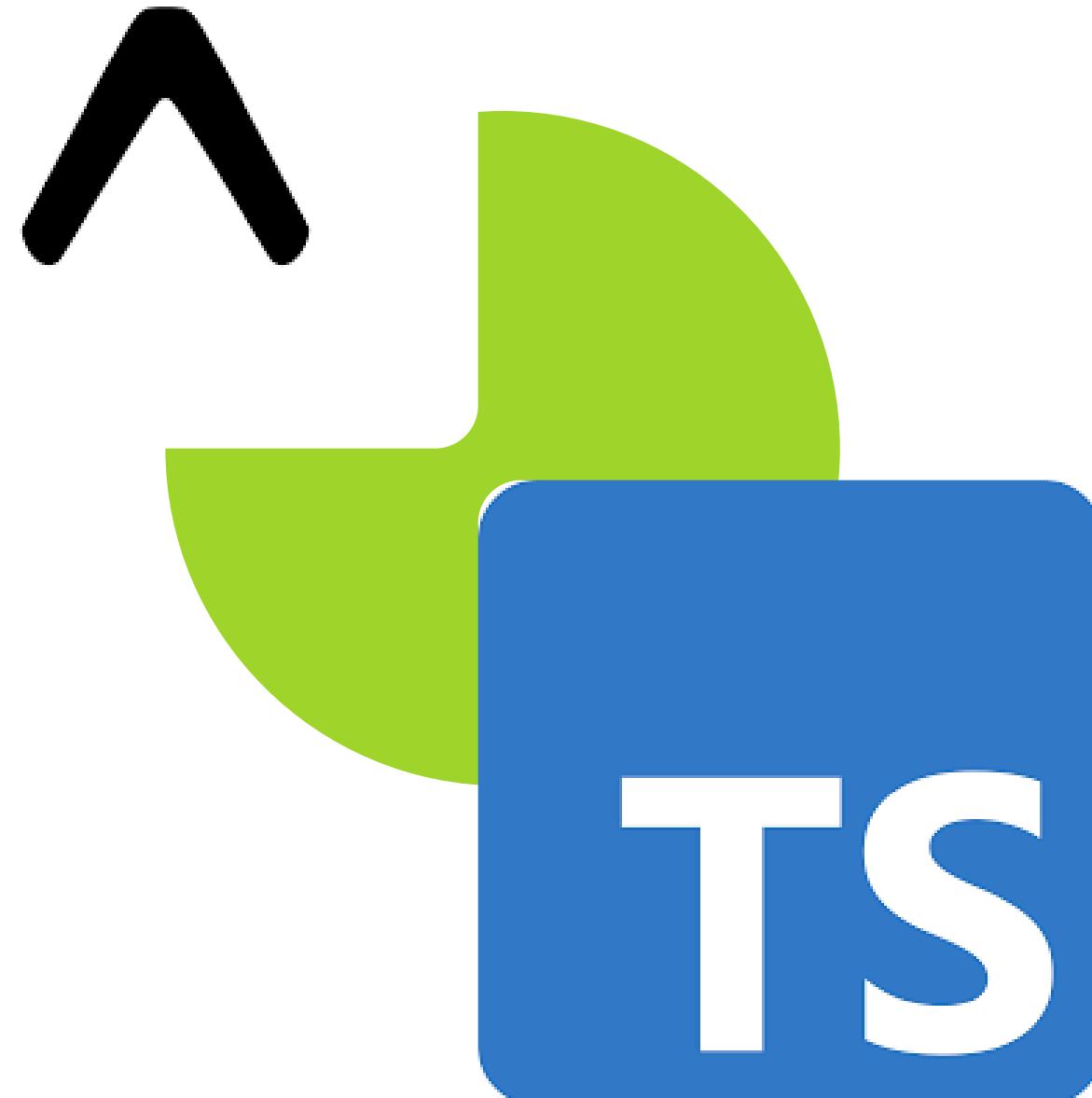
- Dificuldade de acesso a produtos frescos locais em Palmas
- Produtores familiares com limitações para alcançar consumidores
- Falta de transparência na cadeia de distribuição

Solução proposta:

- Marketplace digital para produtos hortifrutílicos locais
- Conexão direta entre produtores e consumidores
- Sistema de entregas otimizado com entregadores parceiros

Proposta de valor:

- Para consumidores: Produtos frescos, locais e com rastreabilidade
- Para produtores: Ampliação do mercado e redução de intermediários
- Para entregadores: Oportunidade de renda com gestão eficiente



# Frontend:

- 01 **React Native com Expo**  
Desenvolvimento multiplataforma eficiente

- 02 **TypeScript**  
Adiciona tipagem estática ao JavaScript, permitindo a detecção de erros durante o desenvolvimento, não na execução.

- 03 **Expo Router**  
Expo Router para navegação

- 04 **Expo Maps**  
Expo Maps para geolocalização



# Backend:

01

## NestJS

O NestJS é um framework Node.js que utiliza TypeScript para criar aplicações de back-end eficientes, escaláveis e confiáveis. Sua arquitetura é modular e fortemente inspirada no Angular, organizando o código em módulos, controladores e serviços. Isso promove uma estrutura robusta e de fácil manutenção, ideal para a construção de APIs e microsserviços complexos.

02

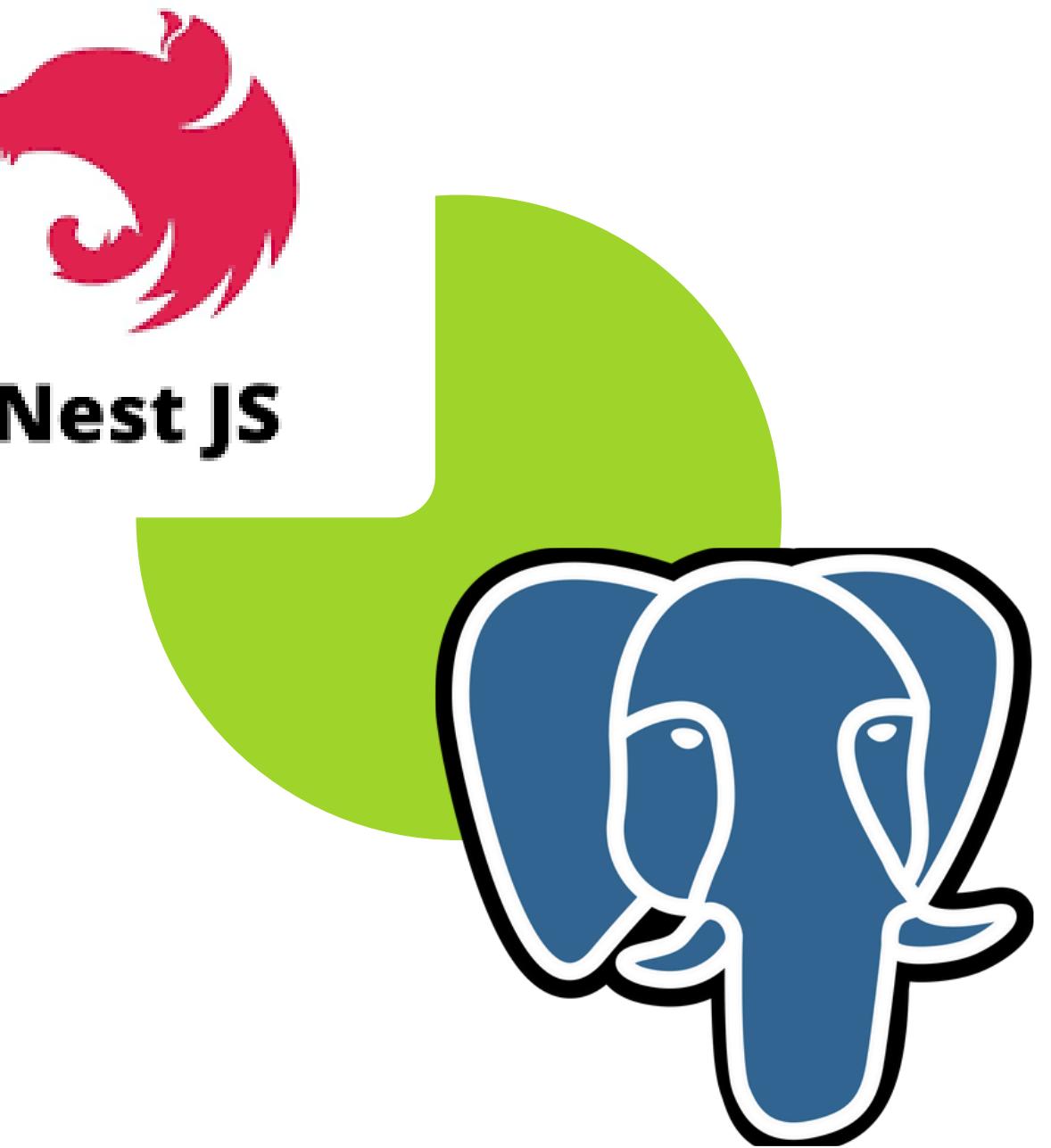
## TypeORM

Ele permite que você trabalhe com seus dados usando classes e objetos em vez de escrever SQL manualmente, agilizando o desenvolvimento. Suportando padrões como Active Record e Data Mapper, ele aumenta a produtividade e a portabilidade do código entre diferentes bancos de dados.

02

## PostgreSQL da AWS

Forte conformidade com o padrão SQL e por seu rico conjunto de funcionalidades, como tipos de dados avançados, indexação complexa e alta extensibilidade. Essas características o tornam ideal para aplicações complexas e que exigem alta integridade e escalabilidade dos dados. Além da robustez e estabilidade



# Outras tecnologias

**01**

## Git + Git Flow

O Gitflow organiza o trabalho no Git com ramos (branches) específicos para funcionalidades, lançamentos e correções, o que padroniza o fluxo de desenvolvimento. Isso justifica seu uso em projetos que necessitam de um ciclo de vida de software bem definido, garantindo um histórico de versões mais limpo e facilitando o trabalho paralelo da equipe.

**02**

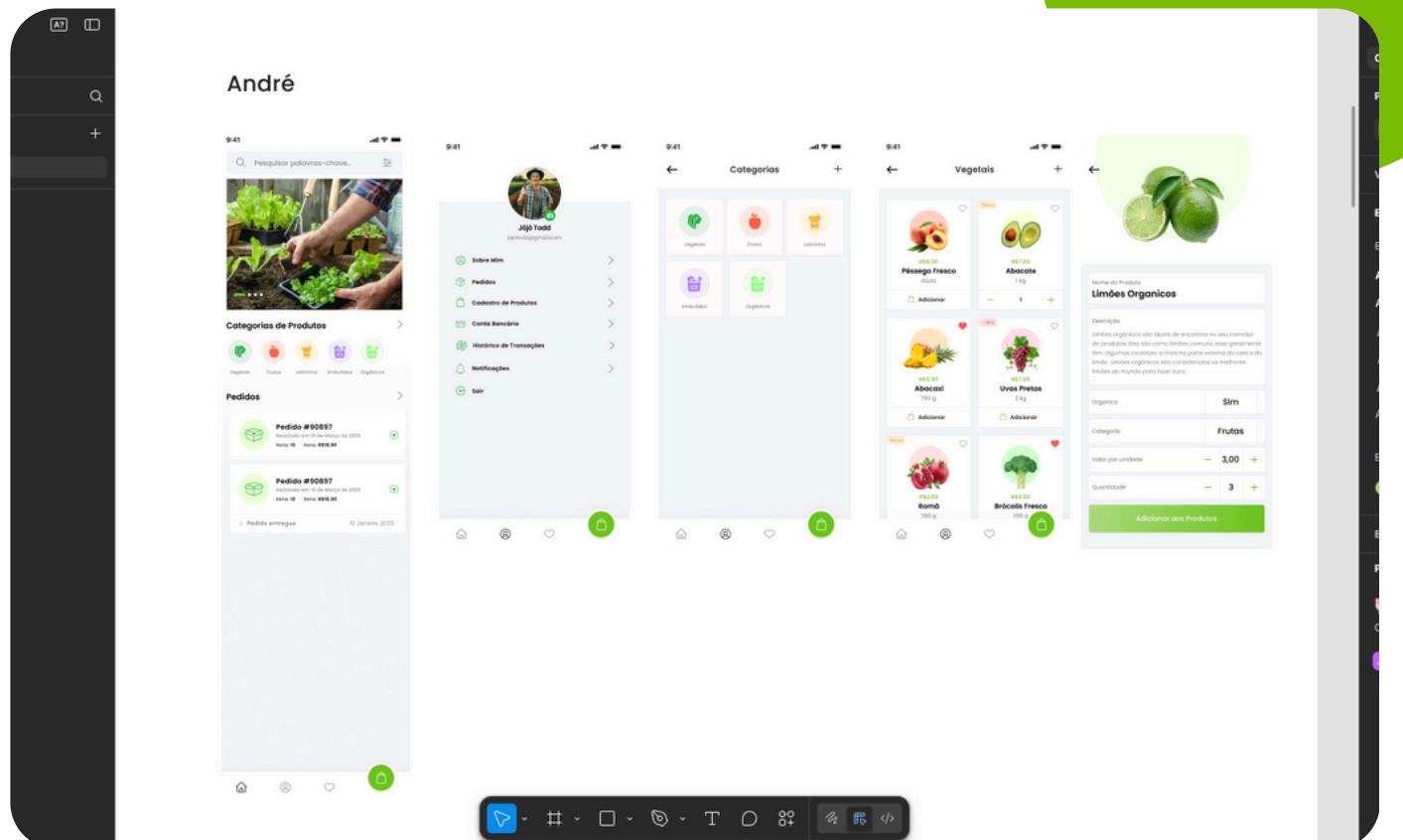
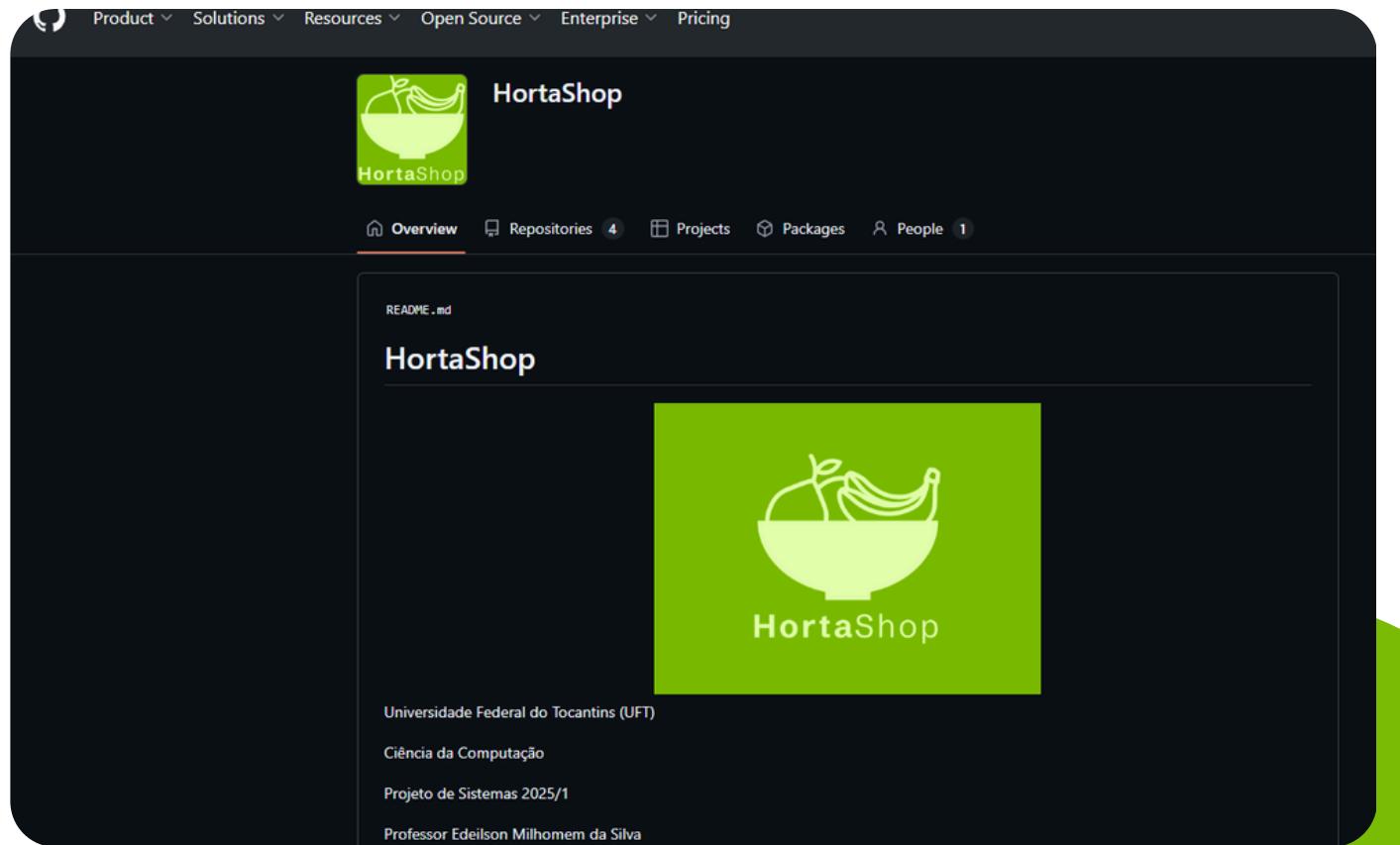
## GitHub

Centraliza os repositórios, no nosso caso a Organization, de um projeto na nuvem, permitindo o trabalho em equipe, o gerenciamento de tarefas e a integração contínua (CI/CD).

**03**

## Figma

Uma ferramenta de design de interfaces (UI) e prototipagem baseada na web que permite a colaboração em tempo real. Centraliza o trabalho de design, facilitando a criação, o teste e o compartilhamento de layouts de aplicativos e sites entre designers e desenvolvedores.

**05**



# MVP Definido vs Implementado

100%

## Funcionalidades 100% Implementadas:

- Sistema completo de autenticação (consumidores, produtores, entregadores)
- Catálogo de produtos com busca e filtros avançados
- Carrinho de compras e checkout completo
- Sistema de pagamentos mockados (PIX e cartão)
- Gestão completa de pedidos
- Sistema de avaliações
- Histórico e rastreamento de pedidos
- Sistema de notificações push



# MVP Definido vs Implementado

85%

## Funcionalidades Parcialmente Implementadas (85%)

- Integração com Google Maps (estrutura pronta, API pendente)
- Sistema de cupons (backend completo, frontend 90%)

0%

## Funcionalidades Não Implementadas

- Chat em tempo real entre usuários
- Sistema de análise de dados avançado
- Sistema de cupons





# Demonstração do Produto

The image displays three screenshots of the HortaShop mobile application, illustrating its user interface and functionality.

- Screenshot 1: Welcome Screen**

Shows the welcome message "Bem-Vindo ao HortaShop" and the tagline "Encontre hortifrutti fresquinho e de qualidade direto do produtor para a sua mesa." The time is 9:41. A hand is shown holding a green reusable produce bag filled with various vegetables.
- Screenshot 2: Home Screen**

Shows a promotional banner for a 20% discount on the first purchase. Below it are sections for "Categorias" (Vegetais, Frutas, Bebidas, Mercado) and "Produtos em destaque". Two products are highlighted: "Pêssego Fresco" (R\$8.00 per dúzia) and "Abacate" (R\$7.00 per 1kg). An "Adicionar" button is at the bottom.
- Screenshot 3: Account Creation Screen**

Shows a registration form with fields for "Número" (phone number) and a password. There are links for existing users ("Entrar") and for sellers ("Criar conta de vendedor").

# Metodologia Ágil Aplicada

01

## Estrutura das Sprints

- 1 Entrega Primária de 3 semanas
- 4 Sprints de 2 semanas cada
- Planejamento detalhado com distribuição de tarefas
- Participação cruzada - cada membro trabalhou em frontend e backend

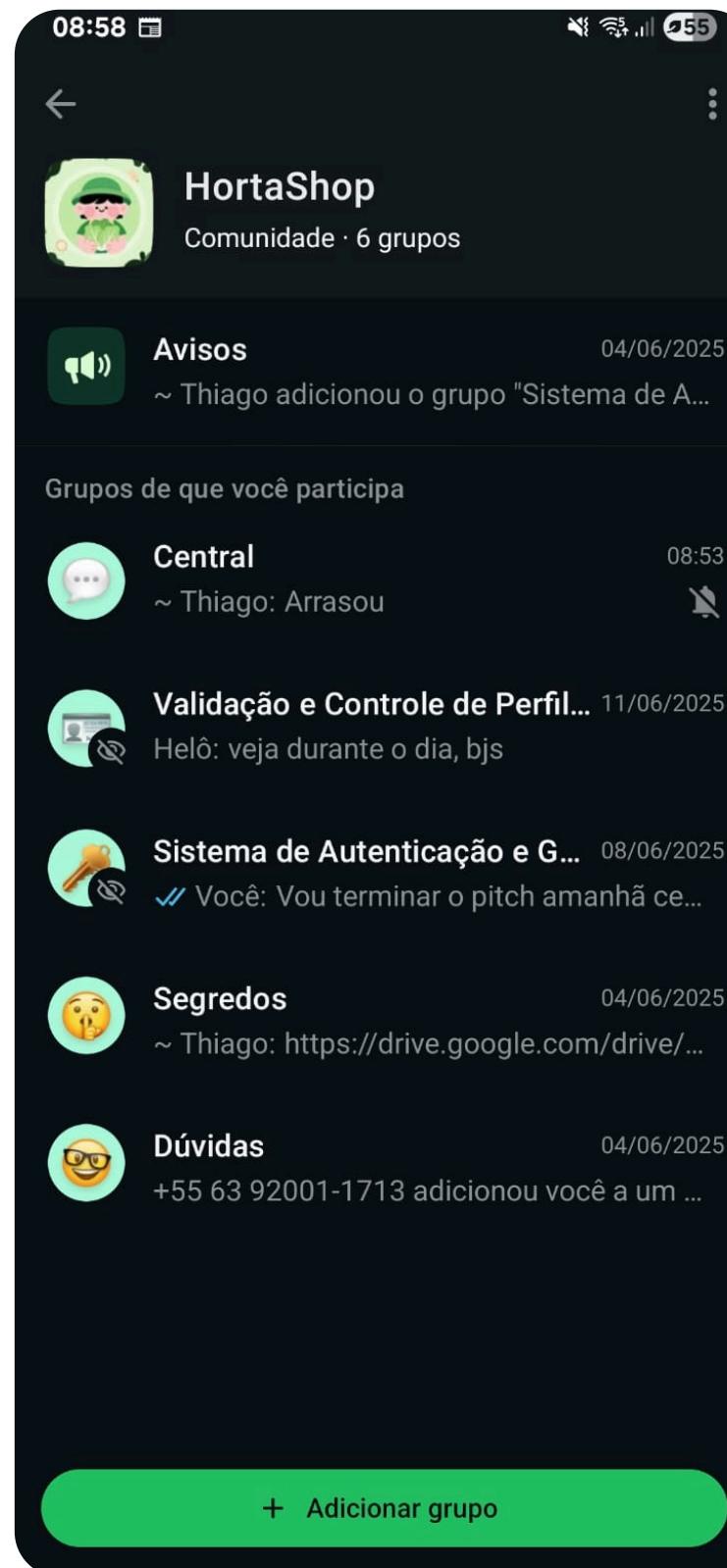
Sprint 1: Autenticação e perfis de cliente

Sprint 2: Catálogo, busca e perfil de produtor

Sprint 3: Sistema de pedidos completo

Sprint 4: App entregador e integrações avançadas





# Metodologia Ágil Aplicada

02

## Ferramentas de Gestão

- Jira para kanban
- Git Flow para versionamento
- Pull Requests com revisão obrigatória
- Comunidade no Whatsapp

03

## Colaboração do Time

- Divisão equilibrada entre frontend/backend
- Revisão cruzada de código
- Pair programming para funcionalidades complexas
- Formulários semanais de feedback
- Documentação colaborativa



# The twelve-factor app

01

## Codebase

Repositórios rastreados no Git com múltiplos deploys:

- Organization no GitHub que reúne diferentes repositórios com diferentes funções

02

## Dependencies

Dependências explicitamente declaradas

- package.json com versões específicas
- npm install para reproduzir ambiente

03

## Config

Configuração armazenada em variáveis de ambiente

```
DB_HOST=hortashop.ct60skegorsk.sa-east-1.amazonaws.com
DB_PORT=5432
DB_DATABASE=hortashop
DB_USERNAME=postgres
DB_PASSWORD=G6ySkp25CYKywUs6ecHf

JWT_SECRET=0HDzJ0/PqeWmRBfalu5cg24VXkbGQ

GOOGLE_CLIENT_ID=SEU_CLIENT_ID_AQUI
GOOGLE_CLIENT_SECRET=SEU_CLIENT_SECRET_AQUI
GOOGLE_CALLBACK_URL
=http://localhost:3000/auth/google/callback
```





# The twelve-factor app

04

## Backing Services

Recursos como serviços anexados:

- PostgreSQL como serviço externo
- APIs de pagamento como recursos anexados
- Google Maps API como serviço backing

05

## Build, Release, Run

Separação estrita dos estágios:

- Build: Compilação TypeScript + bundle
- Release: Combinação build + config
- Run: Execução do release no ambiente

06

## Processes

Aplicação como processos stateless

- Backend NestJS como processo stateless
- Estado persistido apenas no banco de dados
- Sessões gerenciadas via JWT tokens

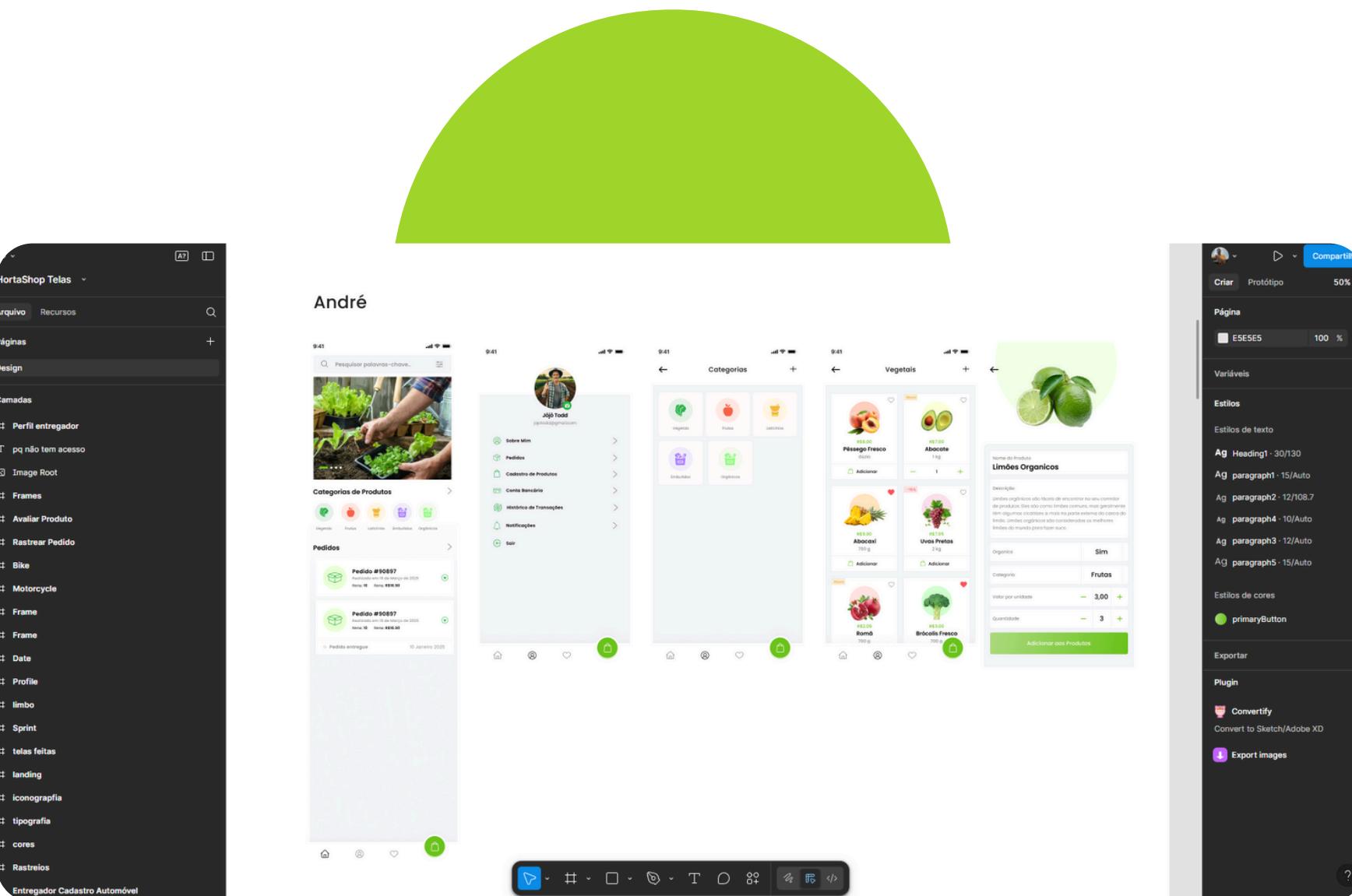
```
DB_HOST=hortashop.ct60skegorsk.sa-east-1.amazonaws.com
DB_PORT=5432
DB_DATABASE=hortashop
DB_USERNAME=postgres
DB_PASSWORD=G6ySkp25CYKywUs6ecHf
```

```
JWT_SECRET=0HDzJ0/PqeWmRBfalu5cg24VXkb
```

```
GOOGLE_CLIENT_ID=SEU_CLIENT_ID_AQUI
GOOGLE_CLIENT_SECRET=SEU_CLIENT_SECRET_AQUI
GOOGLE_CALLBACK_URL
=http://localhost:3000/auth/google/callback
```



# Técnicas de UX/UI Aplicadas



01

## Design System

- Paleta de cores consistente (verde #6CC51D como primary)
- Tipografia padronizada (Poppins)
- Componentes reutilizáveis

02

## Prototipagem

- Figma para wireframes e protótipos de alta fidelidade
- Validação de fluxos antes da implementação
- Design responsivo para diferentes tamanhos de tela



# Técnicas de UX/UI Aplicadas

03

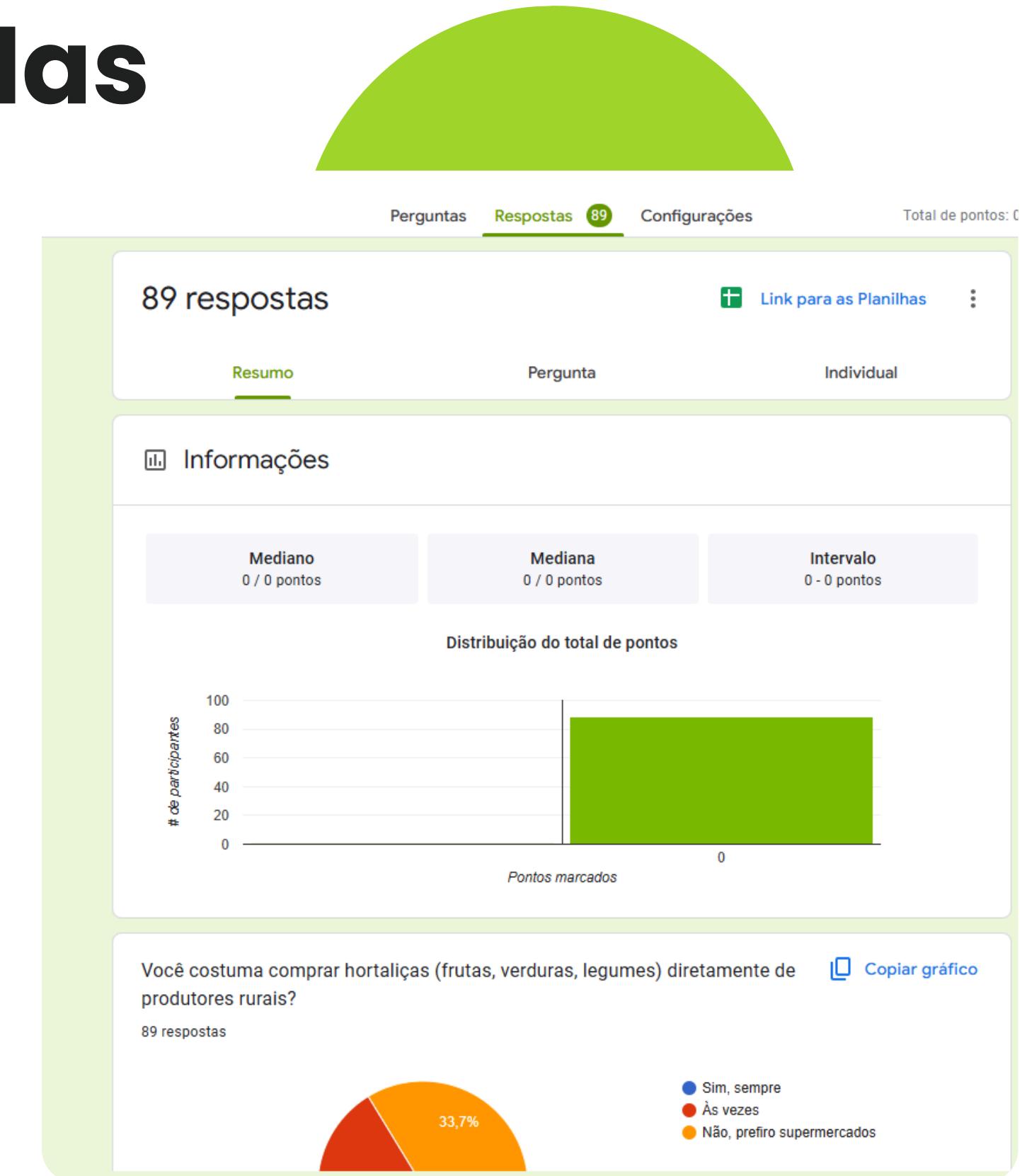
## Usabilidade

- Navegação intuitiva com bottom tabs
- Feedback visual para todas as ações do usuário
- Estados de loading e erro bem definidos
- Acessibilidade com contraste adequado

04

## Testes de Usabilidade

- Validação com usuários reais durante o desenvolvimento
- Iterações baseadas em feedback
- Simplificação de fluxos complexos





# Principais Desafios Enfrentados

01

## Técnicos

- Integração complexa entre múltiplos tipos de usuário
- Sincronização de estados entre apps
- Gestão de dependências entre módulos
- Performance com grandes volumes de dados

02

## De Processo

- Coordenação de 5 desenvolvedores trabalhando simultaneamente
- Balanceamento entre qualidade e prazo
- Gestão de conflitos de merge em repositório compartilhado

03

## De Negócio

- Validação de requisitos com stakeholders reais
- Balanceamento de funcionalidades vs simplicidade
- Definição de prioridades com recursos limitados



# Soluções Implementadas



01

## Técnicos

- Arquitetura modular bem definida
- Padronização de interfaces entre módulos

02

## Processo

- Comunicação 24/7
- Code review obrigatório
- Pair programming para funcionalidades críticas

03

## Negócio

- Prototipagem antes da implementação
- Feedback loops curtos com usuários
- Foco no MVP essencial

# Equipe de desenvolvimento

---

**Antonio**

Autenticação, pagamentos, app entregador

**Gustavo**

Pedidos, histórico, rastreamento tempo real

**Helorrayne**

Gestão de produtos, perfil de produtores

**Raphael**

Checkout, Google Maps, rotas de entrega

**Thiago**

Carrinho, avaliações, notificações push



# Pontos Positivos

---

01

## Colaboração Efetiva

- Todos trabalharam em frontend E backend
- Revisão cruzada garantiu qualidade
- Conhecimento compartilhado em todas as áreas

02

## Comunicação Clara

- Formulários de feedbacks nas releases
- Comunidade no Whatsapp sempre ativa e bem estruturada
- Documentação colaborativa atualizada

03

## Flexibilidade

- Adaptação rápida a mudanças de requisitos
- Apoio mútuo em deadlines apertados
- Distribuição dinâmica de tarefas conforme necessidade





# Pontos de Melhoria

---

01

## Conflitos

- Conflitos ocasionais em merges complexos
- Necessidade de melhor sincronização para releases

02

## Padronização

- Alguns padrões de código estabelecidos tardeamente
- Diferenças de estilo iniciais entre desenvolvedores
- Melhoria na gestão de tempo



# Metodologia Aplicada na Disciplina



## Processo Ágil Efetivo

- Sprints bem estruturadas com objetivos claros
- Adaptação contínua baseada em feedback
- Entregas incrementais com valor real

## Auto-gerenciamento

- Liberdade para tomar decisões técnicas
- Responsabilidade compartilhada pelo sucesso
- Gestão autônoma de prazos e qualidade

## Comunicação Entre Pares

- Colaboração natural entre equipe e as equipes
- Troca de experiências em desafios similares
- Suporte mútuo em tecnologias específicas

## Avaliação dos Pares

- Processo transparente e construtivo
- Feedback honesto sobre contribuições
- Reconhecimento de pontos fortes individuais



# Impacto na Aprendizagem

## Aprendizado Prático

- Aplicação real de conceitos teóricos
- Experiência completa de ciclo de desenvolvimento
- Contato direto com usuários finais

## Soft Skills

- Trabalho em equipe efetivo
- Gestão de tempo e prioridades
- Comunicação técnica clara

# Análise Crítica do Resultado



01

## Proposta de Valor Atingida

- Produto funcional resolvendo problema real
- Usuários conseguem identificar valor rapidamente
- Benefícios claros para todos os stakeholders

02

## Produto Pronto para Uso

- 95% das funcionalidades implementadas
- Testes extensivos realizados
- Interface polida e intuitiva
- Performance adequada para uso real

03

## MVP Validado

- Todas as funcionalidades essenciais implementadas
- Feedback positivo dos usuários testadores
- Viabilidade técnica e comercial comprovada



# Principais Lições Aprendidas

01

## Técnicas

- Importância da arquitetura bem planejada desde o início
- Valor da tipagem forte (TypeScript) em projetos colaborativos
- Necessidade de testes para confiança em mudanças



02

## Processo

- Code review é essencial para qualidade e conhecimento compartilhado
- Comunicação frequente previne problemas maiores
- Prototipagem economiza tempo de desenvolvimento



03

## Pessoais

- Trabalho em equipe potencializa resultados individuais
- Feedback de usuários reais é insubstituível
- Planejamento flexível é melhor que planejamento rígido





# Sugestões para a Disciplina

01

## Pontos Positivos

- Liberdade para escolha de tecnologias
- Problema real com usuários reais (mais real possível)
- Foco em MVP viável
- Avaliação baseada em resultado prático

02

## Sugestões de Melhoria

- Workshops técnicos opcionais (Docker, CI/CD, Cloud)
- Mentoria técnica para decisões arquiteturais
- Sessões de feedback previstas no calendário da disciplina entre equipes



A disciplina proporcionou experiência valiosa próxima ao mercado real

# Obrigade

---

## Agradecimentos especiais

**Professor Edeilson Milhomem** - Pela orientação e metodologia aplicada

**Colegas de turma** - Pela colaboração e troca de experiências

## Equipe HortaShop

**Antonio André • Gustavo • Helorrayne • Raphael • Thiago**

**"Conectando produtores locais aos consumidores através da tecnologia"**

**Frescos, Locais e Digitais**