



MOOVIN

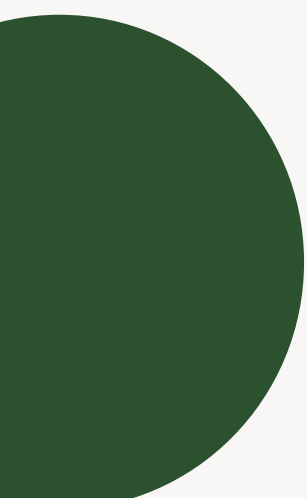




A IDEIA DO PRODUTO

A Moovin é um app e plataforma de gestão que facilita a busca e o aluguel de imóveis, oferecendo recursos como filtros inteligentes, verificação de perfis e gerenciamento de contratos, com foco em segurança e eficiência.





O foco do aplicativo é prover uma forma de conectar inquilinos e proprietários de modo simplificado e seguro. Nosso público-alvo são os habitantes de cidades médias e grandes na faixa etária de 18 a 30 anos



Redes sociais:

- Grupos de WhatsApp, Instagram e Facebook.

Sites e aplicativos:

- OLX, ZAP e QuintoAndar

Anunciantes próprios:

- Imobiliárias, cartaz em postes, outdoors, entre outros.



Desenvolvimento do Escopo e do MVP.

Após a coleta de requisitos foi levantado todos os requisitos funcionais e não funcionais para o projeto na definição do Escopo. Em seguida, foi feita a delimitação daquilo que seria implementado durante a fase de MVP.

Diferenciais levantados:

Facilidade, transparência e segurança

No MVP foi definido que o produto seria um aplicativo **mobile**, porém com algumas funcionalidades de gerenciamento em **web**





Nossa Proposta de valor

Para Inquilinos

- Filtros intuitivos para encontrar o imóvel ideal.
- **Transparência** com informações confiáveis.
- Decisão segura com poucos cliques.

Para Proprietários

- Praticidade para **gerenciar** aluguéis
- Chat integrado para comunicação direta.
- Sistema de avaliações mútuas para confiança.



Funcionalidades planejadas no MVP

[Home](#)[About Us](#)[Contact](#)

- **RF 01:** Cadastro de usuários (locatários e inquilinos).
- **RF 02:** Cadastro, edição e exclusão de imóveis (CRUD).
- **RF 03:** Pesquisa de imóveis com filtros avançados (localização, preço, tipo de imóvel etc.).
- **RF 04:** Exibição de fotos dos imóveis.
- **RF 06:** Exibição de perfil de locatários e inquilinos.
- **RF 07:** Sistema de avaliação para inquilinos, locatários e imóveis.
- **RF 08:** Notificações automáticas para atualizações e interações no app.
- **RF 09:** Perfil detalhado dos imóveis cadastrados.
- **RF 11:** Plano premium para inquilinos e proprietários.
- **RF 16:** Guardar contratos dos aluguéis (recurso premium).
- **RF 17:** Gerenciamento avançado de imóveis (recursos premium).
- **RF 18:** Geração de relatórios sobre imóveis alugados e anunciados (premium).
- **RF 19:** Agendamento de visitas a imóveis (recurso premium).
- **RF 20:** Permitir que o inquilino acompanhe os prazos relacionados ao aluguel (premium).



AS TECNOLOGIAS:

Para a parte Web foi escolhido o uso do Framework Django e na parte mobile o framework Flutter.

Essas tecnologias foram escolhidas por causa do conhecimento prévio de parte da equipe com essas tecnologias, além de suas diversas funcionalidades de otimização do desenvolvimento de software.

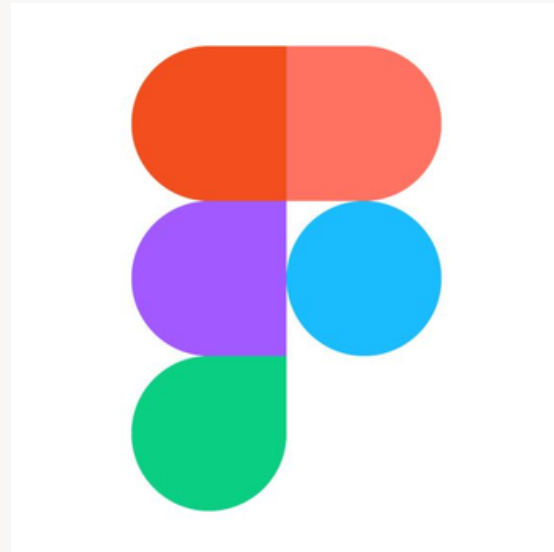


AS TECNOLOGIAS:





Outras ferramentas utilizadas:





A EQUIPE:



Mayconn



Ana Júlia



Victhor



João Vitor



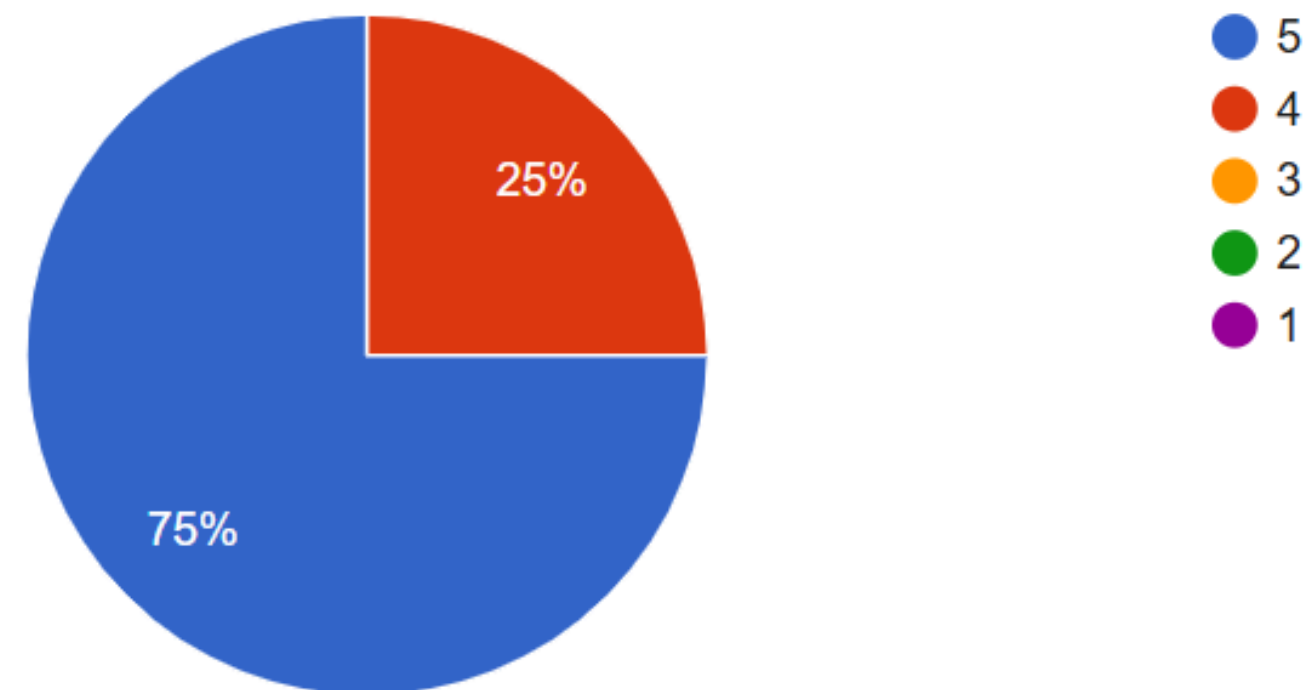
Pedro Lucas



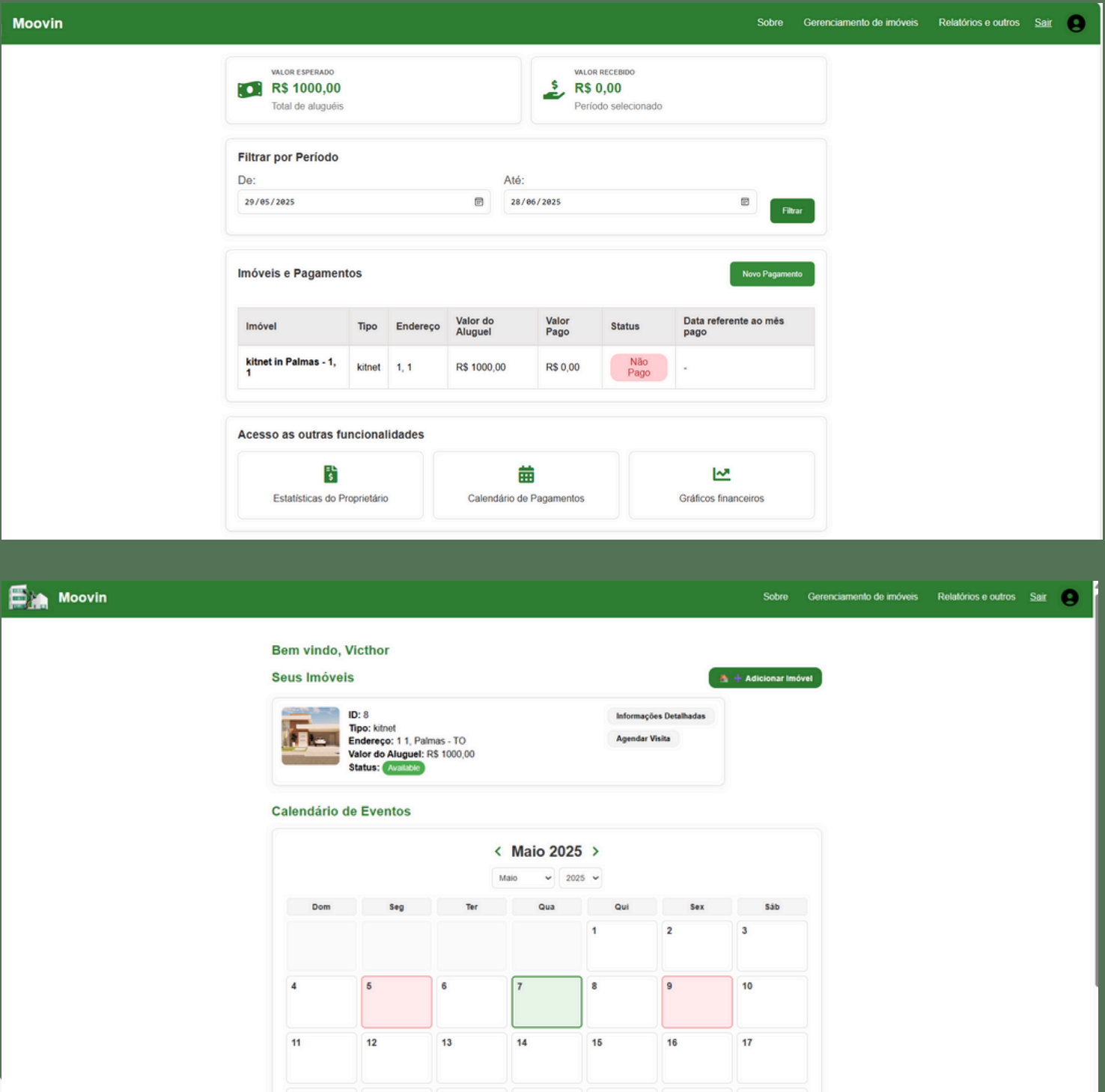
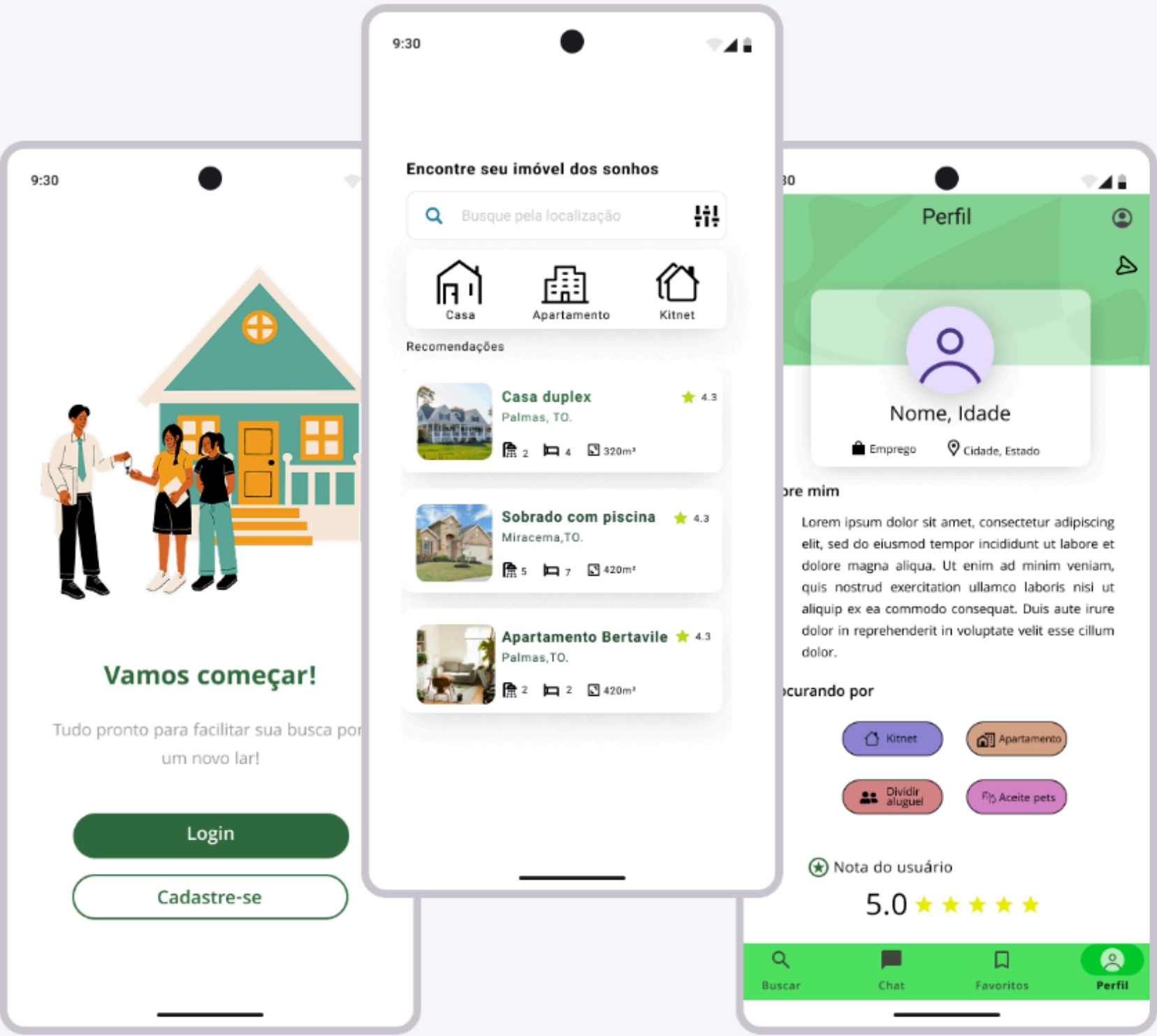
Após a definição da ideia do projeto, foi realizado a confecção do modelo de negócio Canvas para estruturar a proposta. Além de pesquisas com Inquilinos e Locatários reais para coletar os requisitos

Qual o seu nível de interesse em um aplicativo que permite alugar imóveis sem a intermediação de uma imobiliária? (1 - Nenhum interesse / 5 - Muito interesse)

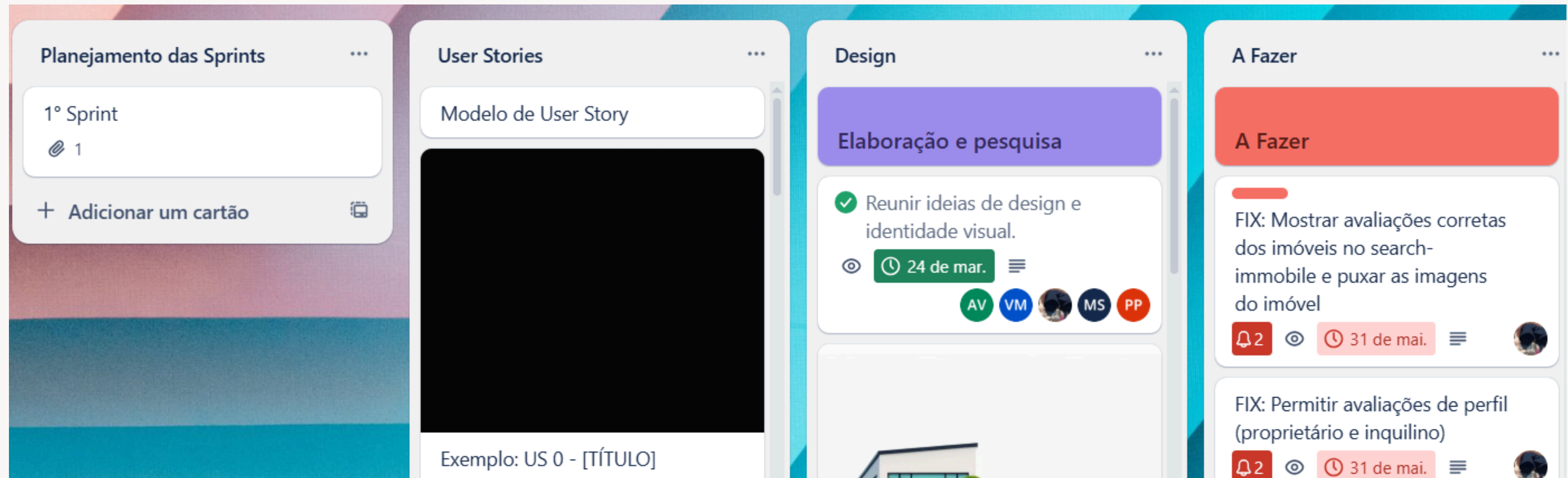
12 respostas



Prototipação



Planejamento de Sprints

[Home](#)[About Us](#)[Contact](#)



Usamos a metodologia Ágil SCRUM no projeto, fazendo iterações de duração de 2 semanas para entregar uma nova versão estável e melhorada do produto.

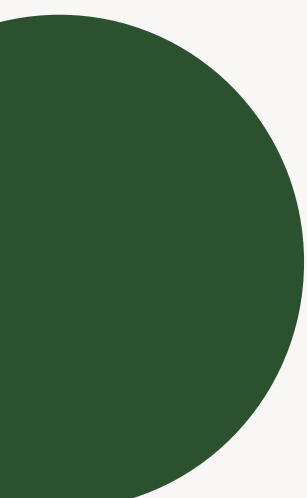
Planejamento de cada Sprint:

- No início da iteração é feita a definição das features a serem implementadas, assim como os responsáveis e os 2 revisores.
- Após a revisão de cada feature, é feita a correção de falhas e inconsistências.
- Ao final da Sprint é feita uma reunião para alinhamento do projeto e planejamento da próxima Sprint.

Foco na melhoria contínua do processo.



- Alinhamento da identidade visual.
- Inconsistências entre a parte Web e Mobile.
- Dificuldade em integrar as features de forma funcional ao final da Sprint.

A large green circle is positioned on the left side of the slide, partially cut off by the edge.

Ao longo do desenvolvimento das Sprints todas as funcionalidades propostas no MVP foram cumpridas, após uma reunião com a equipe foi discutido a viabilidade da adição de novas funcionalidades como o Chat para contato entre Inquilinos e Locatários.



Autoavaliação

Ao realizar uma autocrítica, a equipe percebeu que foi entregue um produto satisfatório e após a sua hospedagem e desenvolvimento da APK ele está pronto para uso com usuários testes, em busca de melhorias nas funcionalidades oferecidas.



Funcionalidades futuras

- Fornecer parcerias com prestadores de serviços. Oferecendo serviços diretamente no app.
- Fornecer a função de aluguel compartilhado



Usabilidade

Técnicas de usabilidade do projeto:

- User Stories
- User Flow
- Teste de usabilidade





Organização arquitetural

WEB

- Arquitetura MVT

(MODEL-VIEW-
TEMPLATE)

MOBILE

- Arquitetura MVVM

(Model-View-
ViewModel)





Manutenibilidade

Técnicas usadas para garantir a manutenibilidade do projeto:

- DRY - Don't Repeat Yourself
- Nomenclatura Clara de variáveis e métodos
- Integração Contínua (CI)
- Entrega Contínua (CD)



Testes com usuários reais

[Home](#)[About Us](#)[Contact](#)

- Foram conduzidos testes remotos com usuários reais para avaliar o aplicativo Moovin.
- Disponibilizamos o aplicativo, fornecemos tarefas específicas para execução e disponibilizamos um formulário para coletar feedback.

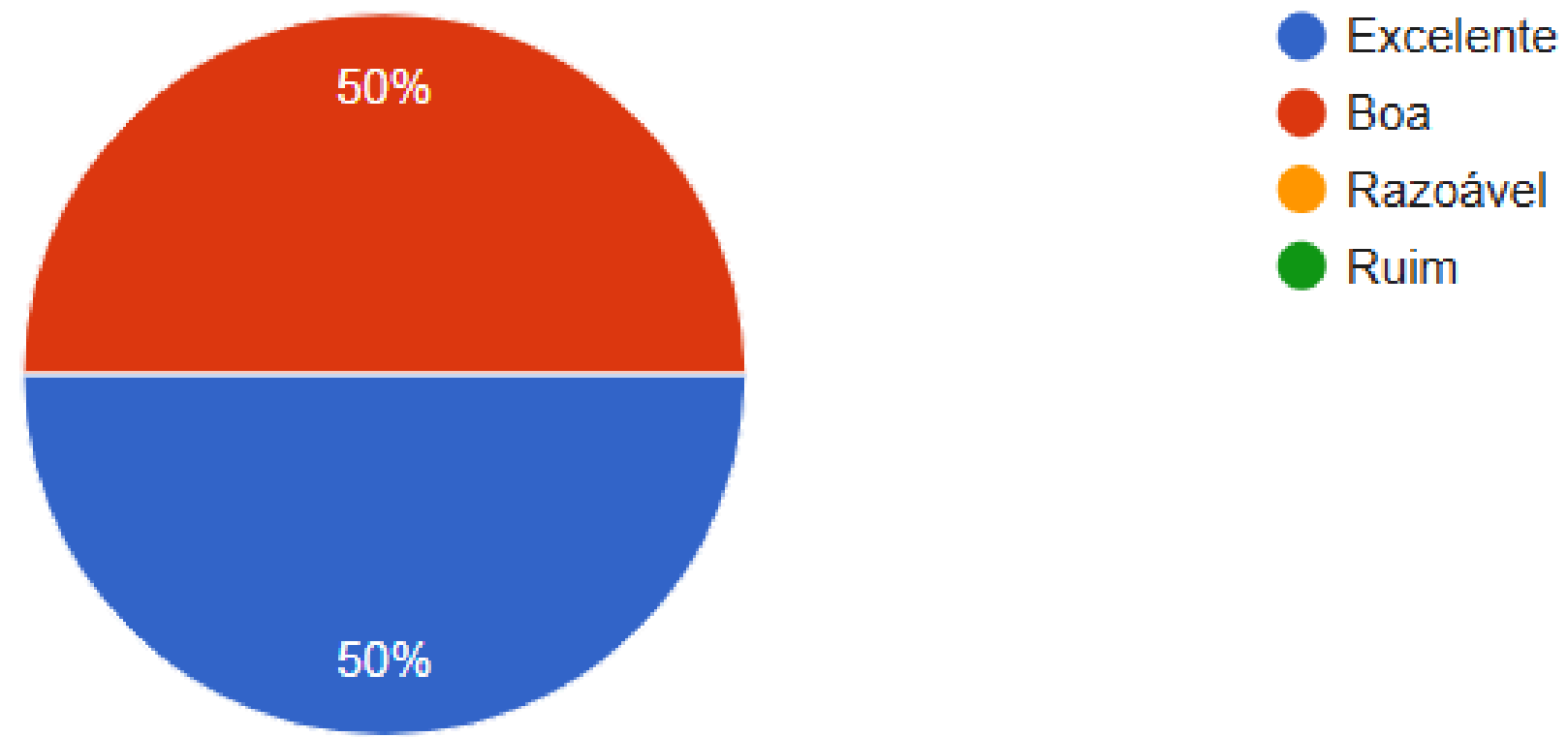
Testes com usuários reais

[Home](#)[About Us](#)[Contact](#)

Avaliação geral dos usuários após o teste com o aplicativo:

Como foi sua experiência geral usando o Moovin?

6 respostas



Testes com usuários reais

[Home](#)[About Us](#)[Contact](#)

Os usuários destacam o aplicativo como uma solução eficaz e promissora para o mercado imobiliário.

Você recomendaria o Moovin para outras pessoas? Por quê?

6 respostas

Sim, é promissor

Sim

Pq é uma vantagem ter um aplicativo desse para disponibilizar seus imóveis e procurar imóveis que se encaixam com oq vc deseja

Sim, app muito intuitivo

Sim, muito útil

Sim. Pela praticidade e funcionalidade

Testes com usuários reais

[Home](#)[About Us](#)[Contact](#)

Novas funcionalidades recomendadas pelos usuários:

Alguma sugestão de funcionalidade que não existe ainda?

4 respostas

Não

Inserir o contrato estipulado pelo proprietario para você poder ler antes de assinar

Filtrar por Pet friendly



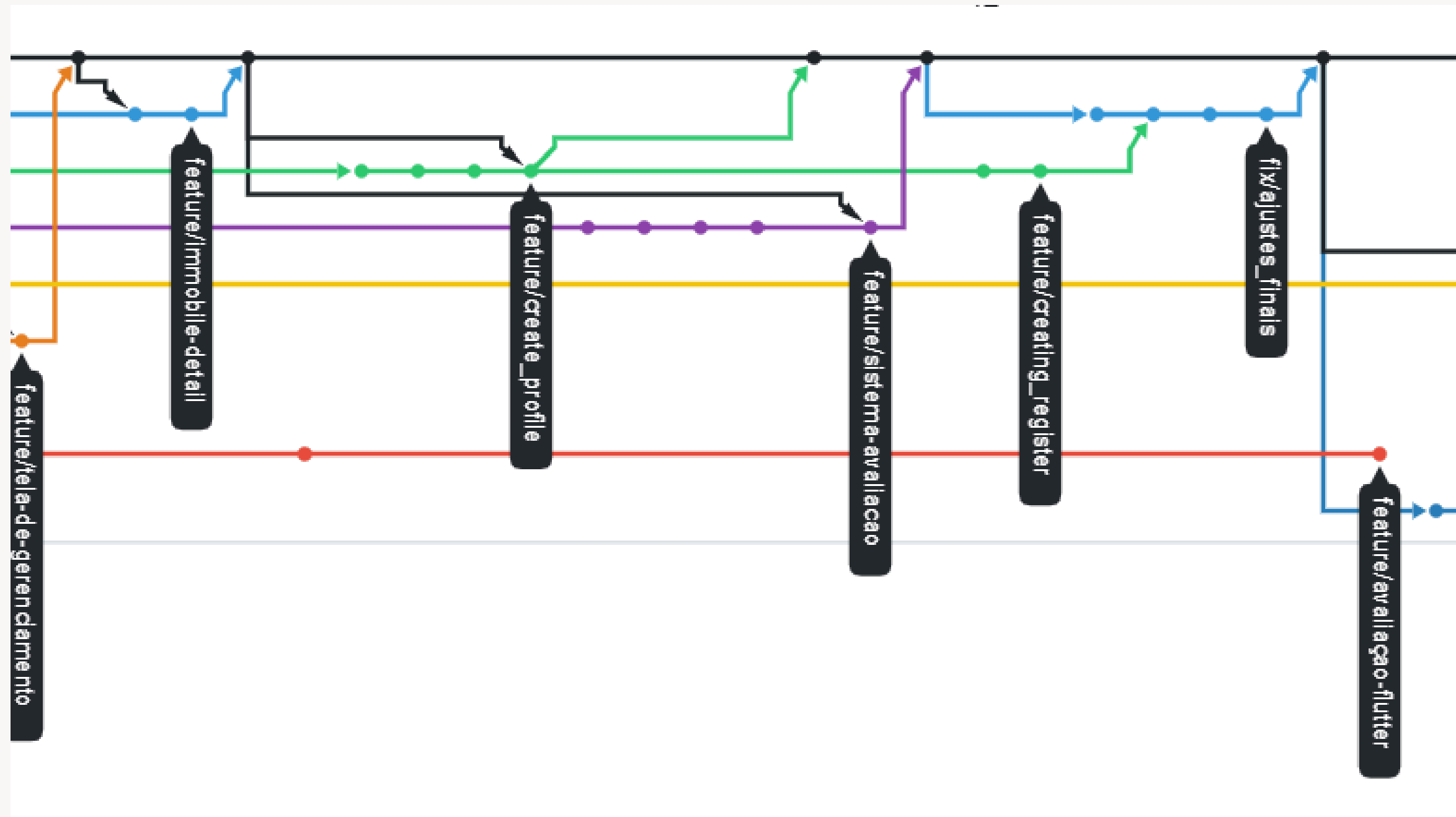
Avaliação do produto a partir do twelve factor-app



Nosso aplicativo **MOOVIN** foi desenvolvido a partir de boas práticas de desenvolvimento. Iremos demonstrar 6 fatores tirados da metodologia twelve factor app para evidenciar isso



Uso de controle de versão e revisão do código do sistema. No caso, a partir do git e gitflow



Dependencias


[Home](#)[About Us](#)[Contact](#)

Todas as dependencias devem ser declaradas explicitadamente.

```
dependencies:
  flutter:
    sdk: flutter
  http: ^1.2.2 # For making API requests
  provider: ^6.1.2 # For state management
  cupertino_icons: ^1.0.8
  intl: ^0.18.1
  flutter_secure_storage: ^9.0.0
  google_fonts: ^6.1.0
  shared_preferences: ^2.0.17
  url_launcher: ^6.1.12
  webview_flutter: ^4.4.2
  webview_flutter_android: ^3.13.0
  webview_flutter_wkwebview: ^3.10.0
  dio: ^5.0.0
  # webview_flutter: ^4.0.0
  image_picker: ^1.0.4
```

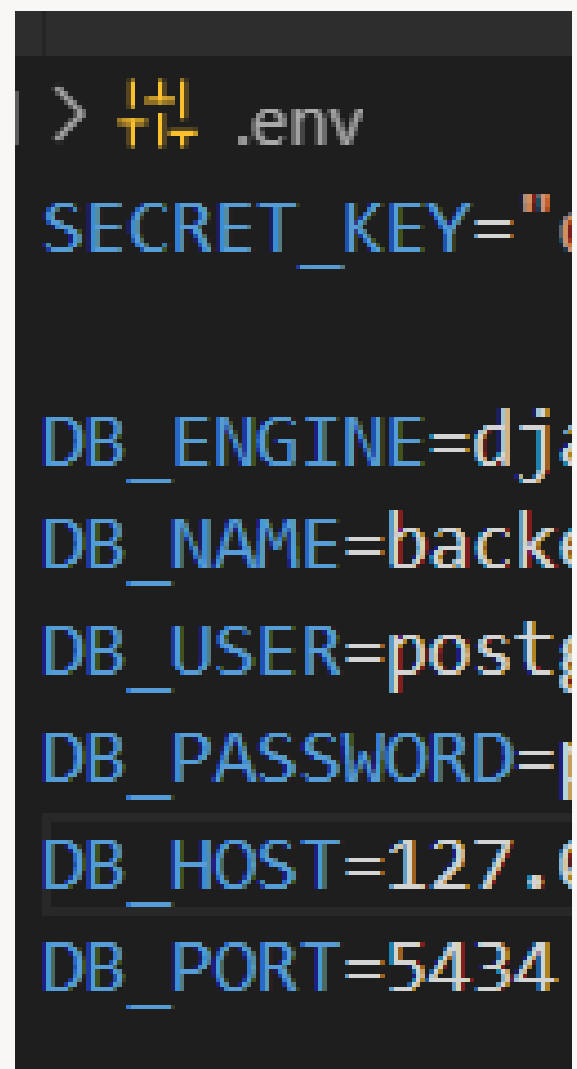



As configurações devem ser estabelecidas em um arquivo separado de variáveis de ambiente e não junto ao código. No caso em um arquivo .env

```
>  .env  
SECRET_KEY="o  
  
DB_ENGINE=dja  
DB_NAME=back  
DB_USER=postg  
DB_PASSWORD=p  
DB_HOST=127.0  
DB_PORT=5434
```



Serviços externos como Banco de dados como recursos anexados. Facilitando a escalabilidade do projeto.



```
>  .env
SECRET_KEY="o

DB_ENGINE=dja
DB_NAME=backe
DB_USER=postg
DB_PASSWORD=p
DB_HOST=127.0
DB_PORT=5434
```


Processos Administrativos



No projeto, utilizamos de processos administrativos. Executando comandos administrativos como:

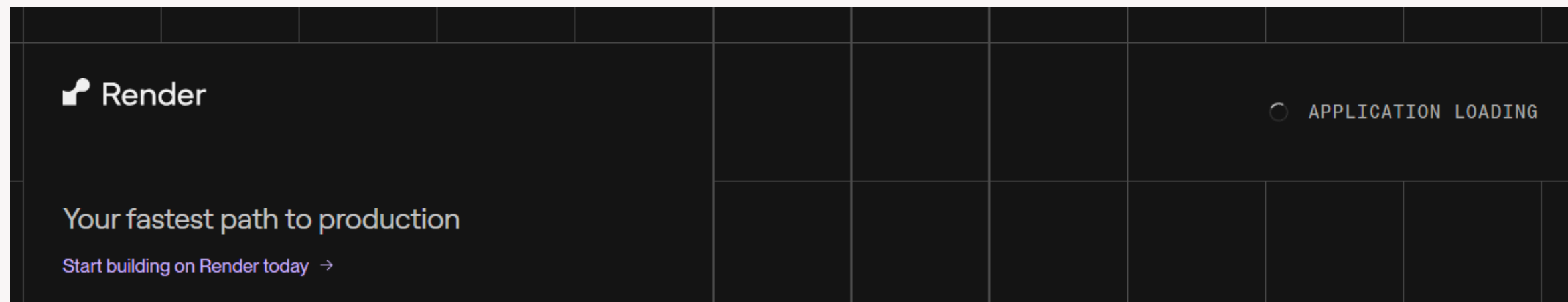
- migrate
- createsuperuser
- shell

por meio do manage.py, de forma pontual e isolada. Esses comandos rodam no mesmo ambiente da aplicação, sem acoplamento ao servidor, seguindo as boas práticas da metodologia Twelve-Factor App.

Descartabilidade

[Home](#)[About Us](#)[Contact](#)

Os processos do aplicativo devem ser “descartáveis”, podem ser iniciados e encerrados a qualquer momento. Isso é garantido pelas APIs REST que não armazenam informações da sessão do usuário no próprio servidor e pelo Servidor Render que é leve e confiável.





Sobre a disciplina





Obrigado pela atenção!

