

Desafio em Equipe

Loja virtual API

Objetivo

Praticar uso de APIs usando a api (dummyjson.com), utilizando Git e GitHub, simulando o trabalho colaborativo em uma empresa de desenvolvimento de software.

Crie as seguintes telas para o APP

1. Tela principal com botão para navegação (montar juntos)
2. Tela com listagem de produtos e detalhe
3. Tela que mostre os carrinhos de compras e detalhe
4. Tela que mostre os usuários e detalhe
5. Tela que mostre postagens e detalhe

3 - 5 pessoas por equipe

Entendendo o App

1 Tela principal

Produtos

Carrinho

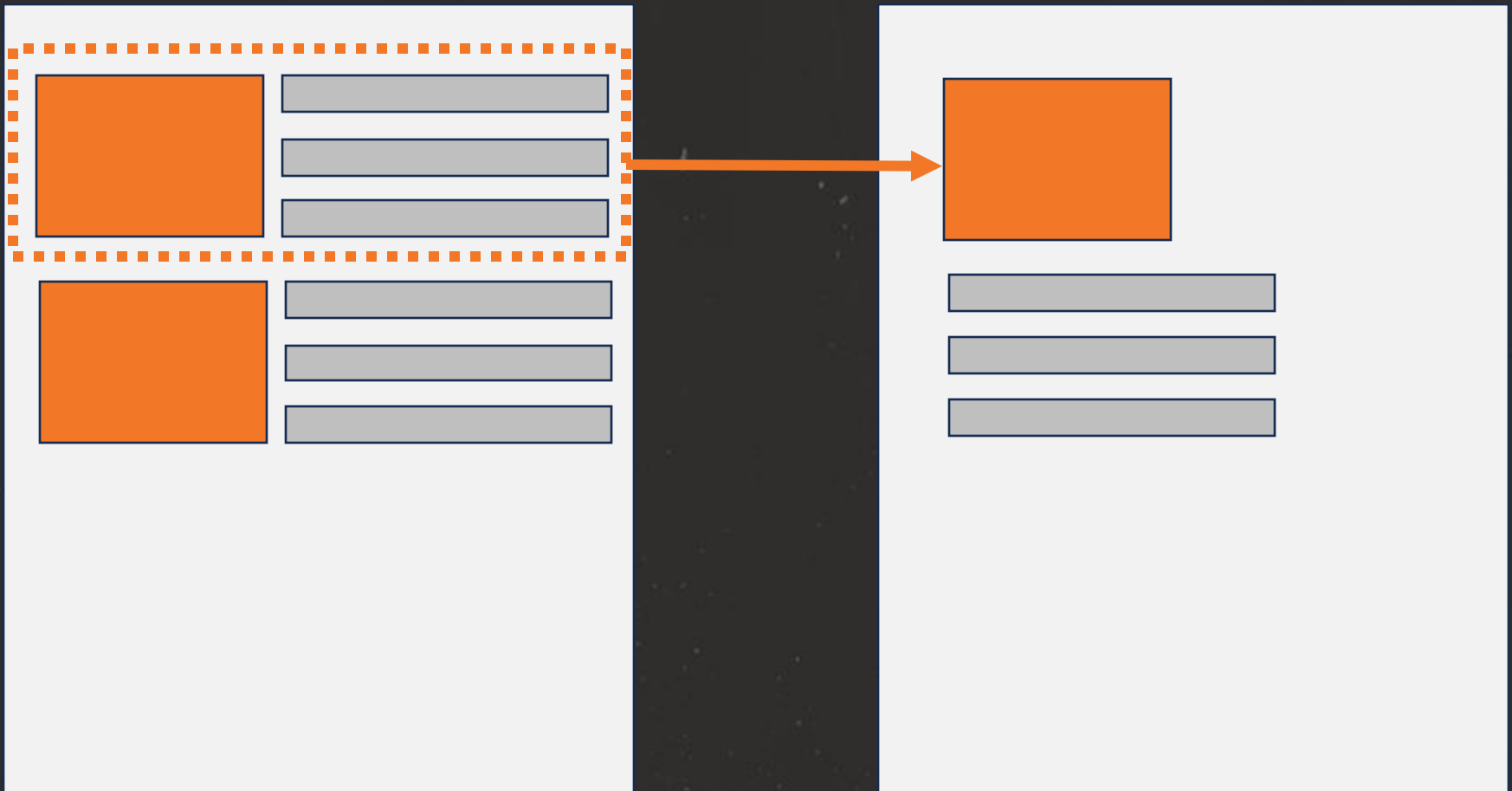
Usuários

Postagens

Entendendo o App

2 Tela com listagem de produtos e detalhes

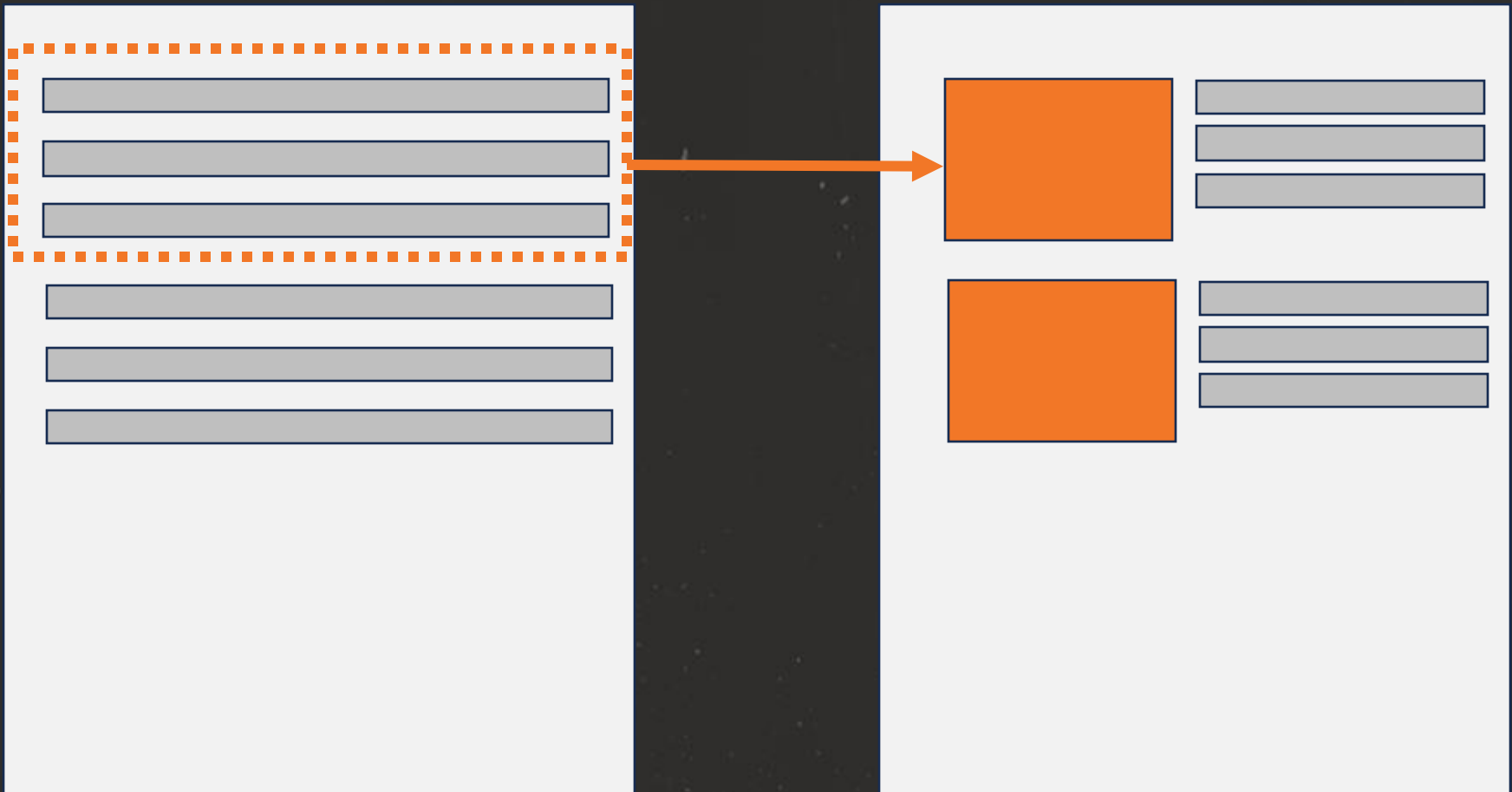
Rota: /products



Entendendo o App

3 Tela que mostre os carrinhos de compras e detalhe

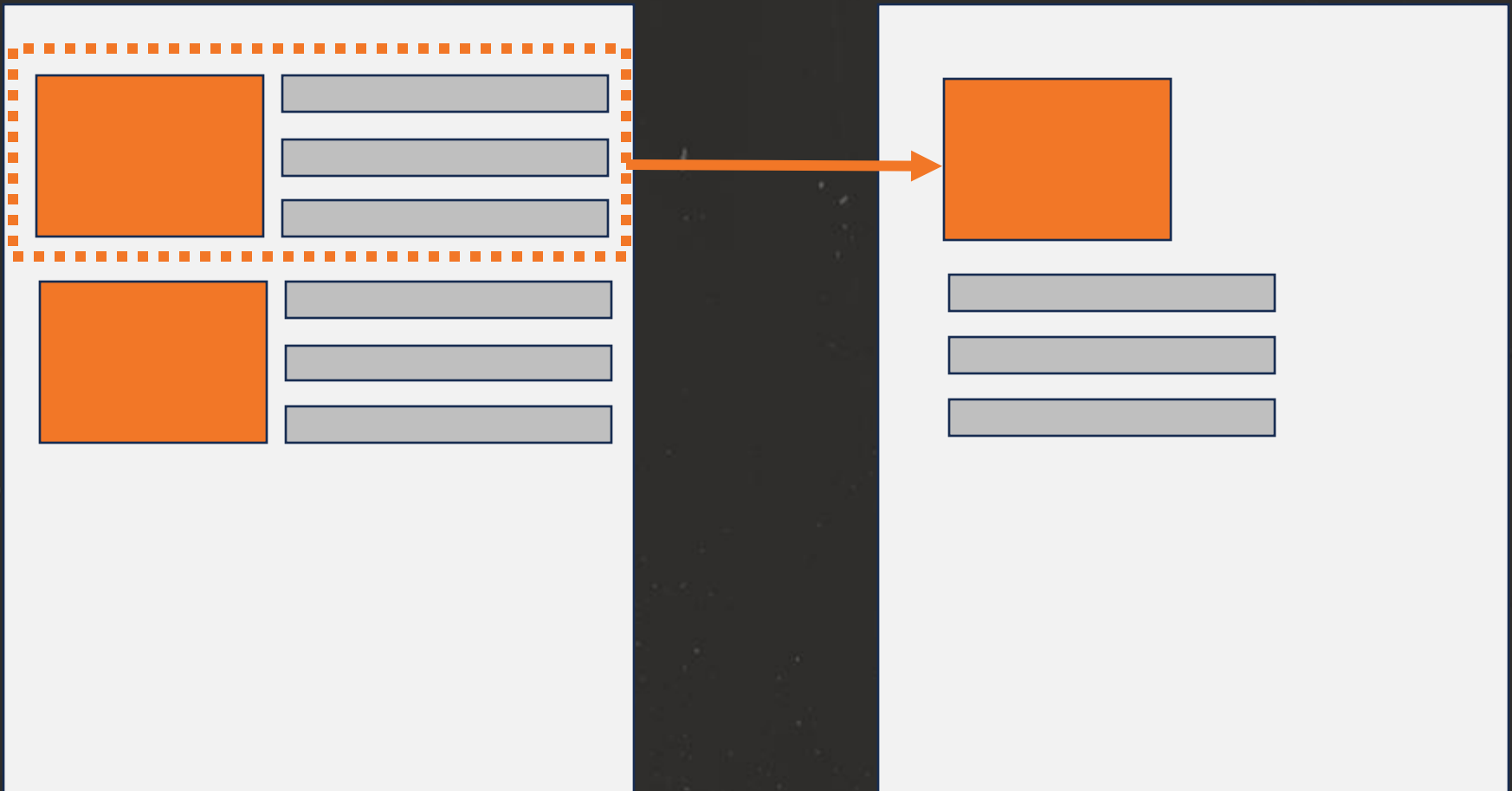
Rota: /carts



Entendendo o App

4 Tela que mostre os usuários e detalhe

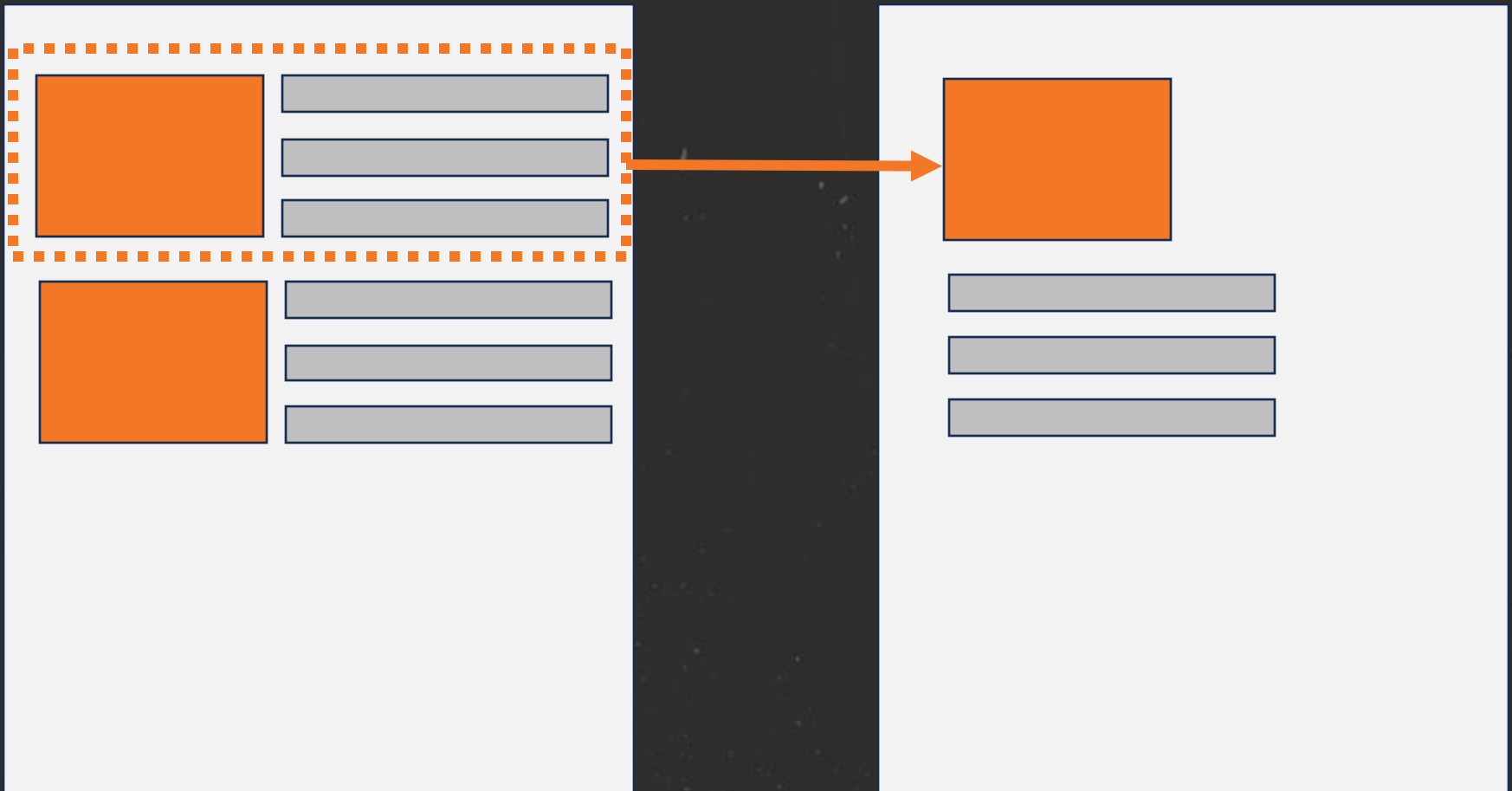
Rota: /users



Entendendo o App

5 Tela que mostre postagens e detalhe

Rota: /posts



Sorteio das equipes

Cada grupo terá entre 3 e 5 membros, e cada membro sorteado terá responsabilidade de criar um recurso do App, exemplo:

1 - João



1 Listagem de produtos...

2 - Maria



2 Listagem e carrinho...

...

O número sorteado para o seu nome, indicará qual recurso irá fazer, o João que tem o número 1 sorteado, ficará com o recurso "1 – Listagem de produtos..." e assim com todos os membros.

O membro de número 1 ficará responsável por criar grupos temporários no WhatsApp para organização das equipes, descendo para o número 2,3, 4...caso o membro 1 não possa participar.

Usar commit Semântico

Para padronizar os commits, utilize o "Commits semânticos" (Convensão de Commits), que permitirão melhor organização, utilize os commit semânticos de acordo com o que você precisa criar, criando um padrão de nomenclatura, tais como:

feat: adicionar nova funcionalidade

fix: corrigir um bug ou comportamento

style: alterações visuais ou de layout sem impacto funcional

refactor: reestruturação de código sem alteração de funcionalidade

docs: atualizações no README ou documentação

test: adicionar ou corrigir testes

chore: tarefas administrativas, como ajustes de build, configs, etc.

Usar commit Semântico

Exemplos de utilização

feat: criar layout da tela de listagem de produtos

feat: implementar navegação da tela de produtos para detalhes de produto

style: ajustar alinhamento de botões na tela de produtos

fix: corrigir navegação incorreta ao clicar em produtos

docs: adicionar instruções de execução no README

chore: configurar arquivo .gitignore

Regras de colaboração

1. Crie uma branch para cada funcionalidade (ex: feature/produtos)
2. Cada membro inicia a tarefa na sua branch
3. Ao terminar, cria um Pull Request para a branch "main"
4. Outro membro do grupo deve revisar e aprovar
5. Resolver conflitos em caso de conflitos no merge em equipe
6. Commits devem ser claros e padronizados
7. Todos devem participar da revisão de ao menos 1 PR de colega

Observações importantes

1. Obrigatório criar uma listagem de itens e outra tela para exibição de detalhes de cada item (cada membro).
2. Obrigatório trabalhar na sua própria Branch, crie uma Branch para isso
3. Cada membro, tem que ter ao menos 4 commits durante a criação de cada tela.