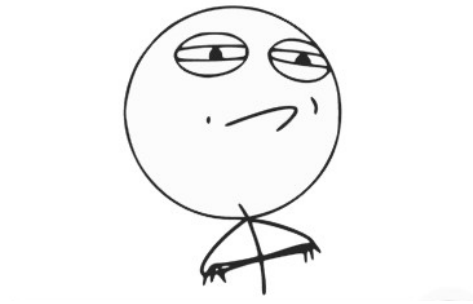


CHALLENGE ACCEPTED

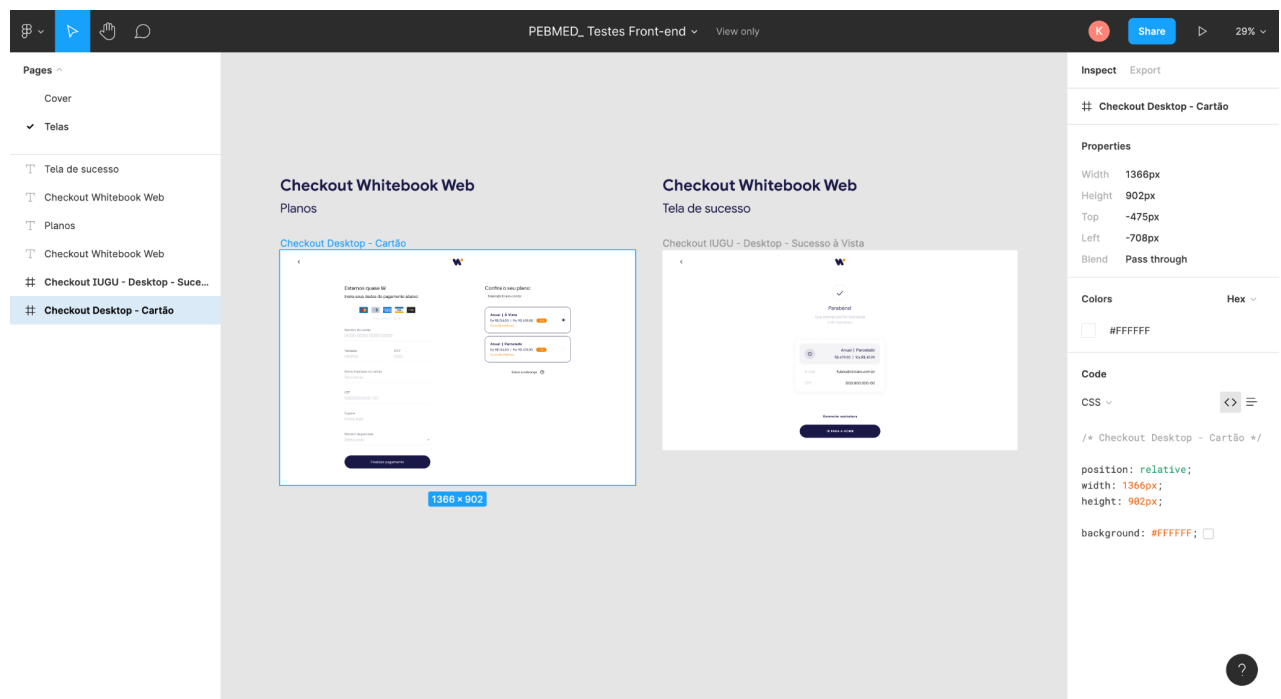


Desafio frontend

Desenvolver o frontend de um sistema de checkout onde um usuário poderá fazer uma assinatura anual ou mensal apresentando sucesso ou erro ao submeter o formulário, enviando as informações de pagamento para processamento pelo backend.

1. O que devo fazer?

- Criar uma aplicação web simulando um fluxo de compra de produto digital, contendo as seguintes telas:
 - Tela de planos
 - Tela de pagamento
 - Tela de confirmação
- As telas com todas as medidas, cores e textos necessários estão disponíveis no Figma:
https://www.figma.com/file/YqvAioQ7Txa5H3HUb4CYzf/PEBMED_-_Testes-Front-end?node-id=1%3A444



2. Requisitos não funcionais e orientações gerais

- Pode ser feito em react.js, next.js, vue.js ou angular (javascript ou typescript)
- Os dados devem ser validados (existência e formatos) na inserção/atualização para garantir consistência da base;
- Implementar testes unitários e/ou de integração e/ou documentação de testes (casos de teste / script de teste);
- User a fonte: https://fonts.google.com/specimen/DM+Sans?preview.text_type=custom
- Consumir endpoint de ofertas na tela de planos:
GET <https://private-0ced4-pebmeddesafiofrontend.apiary-mock.com/offer>
response-body-example:

```
[
  {
    "id":32,
    "storeId":"anual_parcelado_iugu",
    "title":"Premium Anual",
    "description":"Parcelado",
    "caption":"7 Dias Grátis",
    "fullPrice":600,
    "discountAmmount":60,
    "discountPercentage":0.1,
    "periodLabel":"mês",
    "period":"annually",
    "discountCouponCode":null,
    "order":2,
    "priority":2,
    "gateway":"iugu",
    "splittable":true,
    "installments":12,
    "acceptsCoupon":true
  },
]
```

- Consumir o endpoint de pagamento na tela de checkout:
POST <https://private-0ced4-pebmeddesafiofrontend.apiary-mock.com/subscription>
request-body:

```
{
  "couponCode": null,
  "creditCardCPF": "98765432100",
  "creditCardCVV": "123",
  "creditCardExpirationDate": "10/21",
  "creditCardHolder": "Cássio Scofield",
  "creditCardNumber": "5555444433332222",
  "gateway": "iugu",
  "installments": 1,
  "offerId": 18,
  "userId": 1
}
```

response-body:

```
{
  "couponCode": null,
  "creditCardCPF": "98765432100",
  "creditCardCVV": "123",
  "creditCardExpirationDate": "10/21",
  "creditCardHolder": "Cássio Scofield",
  "creditCardNumber": "5555444433332222",
  "gateway": "iugu",
  "installments": 1,
  "offerId": 18,
  "userId": 1,
  "id": 1
}
```

3. O que deverá ser entregue?

- As instruções de como rodar o projeto (use o readme.md);
- Qualquer artefato que seja relevante para a execução do projeto como arquivos, scripts de banco, informações de conexão com o banco e etc (também no readme);
- O projeto deve ser hospedado no git (bitbucket, github, gitlab, etc) para ser avaliado;

4. O que será avaliado?

- Funcional (alcançou os requisitos propostos?);
- Não funcional (alcançou os requisitos propostos?);
- Utilização de boas práticas (princípios SOLID, code-smells, padrões w3c, etc);
- Estrutura e organização (componentização, uso de camadas, etc);
- Legibilidade do código (nomenclatura de classes, métodos e variáveis, lint, etc);
- Implementação de testes que garantam que o código está atendendo os requisitos;
- Documentação (histórico de commits no git, readme, diagramas, padrões);

5. O que será considerado como diferencial?

- Layout responsivo (mobile-friendly)
- Fidelidade com o layout proposto e atenção a detalhes (pixel perfect)
- Testes automatizados (unitários, integrados ou E2E)
- Tratamento de erros (logs/instrumentação e mensagens)
- Hospedar em um ambiente cloud a sua escolha (Heroku, AWS EBS, IBM Cloud, etc)
- Implementar alguma ferramenta de lint ou qualidade (sonar, code-quality, eslint, etc)

Boa sorte!