



## **Sejam bem-vindos ao Hackathon 2025!**

Estamos empolgados em iniciar esta jornada de inovação e colaboração junto com vocês.

Logo abaixo estão descritos os DESAFIOS, alinhados às temáticas escolhidas por cada um no momento da inscrição (vocês deverão realizar APENAS o desafio para a temática escolhida na inscrição).

Este é o start para que apresentem suas ideias e seu potencial!

 Vocês terão de 18 a 22/08 para concluir e entregar o desafio.

Preparem-se para mergulhar no problema proposto, explorar soluções e apresentar um resultado que reflita seu comprometimento e visão.

Boa sorte!!!

Estamos ansiosos para ver o que vocês irão criar!

### **SOBRE A ETAPA**

1. Esta etapa tem como objetivo avaliar os conhecimentos técnicos e estratégicos dos participantes, bem como sua capacidade de propor soluções inovadoras e aplicáveis aos desafios reais da CAIXA.
2. Podem participar apenas os inscritos previamente no Hackathon. Será considerado apenas o desafio encaminhado pelo candidato para o tema escolhido durante a inscrição.
3. O Desafio será realizado à distância, no prazo de 18 a 22/08.
4. Essa etapa tem caráter classificatório e eliminatório.
5. Serão avaliados durante a etapa a aderência com o tema, inovação e aplicabilidade da solução proposta.
6. Regras de pontuação da etapa:
  - a. Pontuação máxima – 100 pontos.
  - b. Pontuação mínima – 50 pontos, sendo eliminado o candidato que não a atinja.
7. O avanço para próxima etapa se dará conforme ranking de classificação. Os melhores colocados serão convocados para a próxima etapa do Hackathon de acordo com o número de vagas – a ser definido pelo grupo diretivo.



**8. Orientações sobre o Entregável:**

- a. **Formato do nome do arquivo:** NOME DO CANDIDATO\_TEMA DA INSCRIÇÃO em formato PDF. Para o desafio Back-end obrigatoriamente o arquivo deverá ser NOME DO CANDIDATO\_TEMA DA INSCRIÇÃO.ZIP.
  - b. **Tamanho máximo:** 2MB
  - c. **Prazo de envio:** até 22/08 às 23h59
  - d. **Canal de envio:** Plataforma SIPSI (<https://sipsi.caixa>) ou para empregados afastados (exceto LTS e LAT) [cepes05@caixa.gov.br](mailto:cepes05@caixa.gov.br).
9. Ao enviar o desafio, o participante declara que o conteúdo é original e de sua autoria, não violando direitos autorais ou confidencialidade de terceiros."
10. **Canal de Dúvidas** - Dúvidas podem ser encaminhadas via [atendimentopessoas.caixa/](https://atendimentopessoas.caixa/).



## **Cibersegurança**

### **Desafio Técnico – Identificação de vulnerabilidades**

O objetivo deste desafio é fortalecer a cibersegurança da CAIXA identificando vulnerabilidades antes que elas possam ser exploradas por agentes maliciosos. Assim, agora você atua como um hacker ético da CAIXA e atuará em prol dos três pilares da segurança: disponibilidade, integridade e confidencialidade. Esses pilares devem ser considerados para a identificação das vulnerabilidades.

---

#### Objetivo do Desafio

Escolher um sistema web ou aplicativo da CAIXA e identificar vulnerabilidades de cibersegurança.

1. Sistema escolhido (informar SIGLA e URL)
2. Para cada vulnerabilidade identificada, detalhar os passos executados para se chegar na falha de segurança;
3. Descrição da solução para resolução da vulnerabilidade.

---

#### Requisitos Técnicos

- As vulnerabilidades encontradas devem ser categorizadas conforme OWASP Top 10.

---

#### Critérios de Avaliação

- Confirmação da vulnerabilidade informada a partir dos passos informados pelo candidato
  - Proposta de solução viável
-



## **Back-end**

### **Desafio Técnico – Simulação de Crédito**

Para realizar o desafio, você terá acesso a alguns recursos, e precisará de algumas referências, todos serão apresentados na sequência. O trabalho de pesquisa, leitura e compreensão do desafio, além da elaboração de proposta de solução fazem parte da avaliação.

Em um contexto cada vez mais tecnológico, todas as relações – inclusive as bancárias – tem se tornado mais frequentes por meio dos canais digitais. Diversas tecnologias, serviços e processos estão evoluindo para não só acompanhar, mas também para tracionar este movimento, como nos casos do PIX, das contas digitais, do pagamento de benefícios sociais e da contratação de serviços bancários por meio de canais que estão “na mão do cliente”.

É este o cenário deste desafio, **você precisa disponibilizar para todos os brasileiros a possibilidade de simulação de empréstimo**. Por meio dessa solução, qualquer pessoa ou sistema pode descobrir quais são as condições oferecidas para uma negociação.

### **Desafio perfil Back-end: API Simulador**

Uma das ferramentas mais potentes neste aspecto, e fundamental para disponibilização de serviços nos canais tradicionais ou digitais, é a API, sigla em inglês para Interface de Programação de Aplicação, que estabelece um protocolo para troca de informações entre sistemas ou entre processos e camadas de um mesmo sistema.

Vamos desenvolver uma API em linguagem de programação Java 17+ ou C# (Dotnet) 8+ que terá como requisitos:

- Receber um envelope JSON, via chamada à API, contendo uma solicitação de simulação de empréstimo.
- Consultar um conjunto de informações parametrizadas em uma tabela de banco de dados SQL Server.
- Validar os dados de entrada da API com base nos parâmetros de produtos retornados no banco de dados.
- Filtrar qual produto se adequa aos parâmetros de entrada.

# Hackathon



- Realizar os cálculos para os sistemas de amortização SAC e PRICE de acordo com dados validados.
- Retornar um envelope JSON contendo o nome do produto validado, e o resultado da simulação utilizando dois sistemas de amortização (SAC e Price), gravando este mesmo envelope JSON no Eventhub. A gravação no Eventhub visa simular uma possibilidade de integração com a área de relacionamento com o cliente da empresa, que receberia em poucos segundos este evento de simulação, e estaria apta à execução de estratégia negocial com base na interação do cliente.
- Persistir em banco local a simulação realizada.
- Criar um endpoint para retornar todas as simulações realizadas.
- Criar um endpoint para retornar os valores simulados para cada produto em cada dia.
- Criar um endpoint para retornar dados de telemetria com volumes e tempos de resposta para cada serviço.
- Disponibilizar o código fonte, com todas as evidências no formato zip.
- Incluir no projeto todos os arquivos para execução via container (dockerfile / Docker compose)

## Links e Referências

1. O que é API: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>
2. Calculadora SAC e Price: <https://calculojuridico.com.br/calculadora-price-sac/>
3. O que é EventHub: <https://learn.microsoft.com/pt-br/azure/event-hubs/event-hubs-about>
4. SQL Server: <https://learn.microsoft.com/pt-br/sql/sql-server/?view=sql-server-ver16>
5. Dados para conexão com banco de dados:
  - a. URL: dbhackathon.database.windows.net
  - b. Porta: 1433
  - c. DB: hack



# Hackathon



- d. Login: hack
- e. Senha: Password23
- f. Tabela: dbo.Produto

## PRODUTO (dbo)

	Column Name	Data Type	Allow Nulls
💡	CO_PRODUTO	int	<input type="checkbox"/>
	NO_PRODUTO	varchar(200)	<input type="checkbox"/>
	PC_TAXA_JUROS	numeric(10, 9)	<input type="checkbox"/>
	NU_MINIMO_MESES	smallint	<input type="checkbox"/>
	NU_MAXIMO_MESES	smallint	<input checked="" type="checkbox"/>
	VR_MINIMO	numeric(18, 2)	<input type="checkbox"/>
	VR_MAXIMO	numeric(18, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

```
CREATE TABLE dbo.PRODUTO (  
    CO_PRODUTO int NOT NULL primary key,  
    NO_PRODUTO varchar(200) NOT NULL,  
    PC_TAXA_JUROS numeric(10, 9) NOT NULL,  
    NU_MINIMO_MESES smallint NOT NULL,  
    NU_MAXIMO_MESES smallint NULL,  
    VR_MINIMO numeric(18, 2) NOT NULL,  
    VR_MAXIMO numeric(18, 2) NULL  
);
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (1, 'Produto 1', 0.017900000, 0, 24, 200.00, 10000.00)
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (2, 'Produto 2', 0.017500000, 25, 48, 10001.00, 100000.00)
```

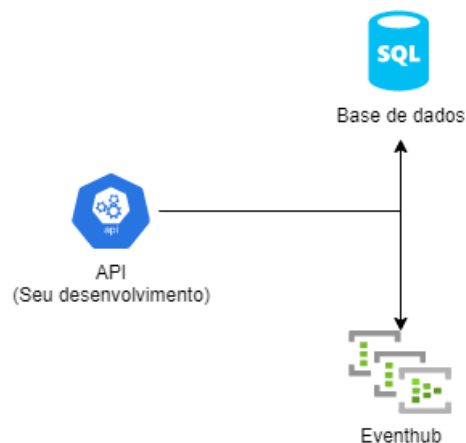
```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (3, 'Produto 3', 0.018200000, 49, 96, 100000.01, 1000000.00)
```

```
INSERT INTO dbo.PRODUTO (CO_PRODUTO, NO_PRODUTO, PC_TAXA_JUROS,  
    NU_MINIMO_MESES, NU_MAXIMO_MESES, VR_MINIMO, VR_MAXIMO)  
VALUES (4, 'Produto 4', 0.015100000, 96, null, 1000000.01, null)
```

# Hackathon



## 7. Arquitetura da Solução



2. Dados para conexão com EventHub:  
Endpoint=sb://eventhack.servicebus.windows.net/;SharedAccessKeyName=hack;SharedAccessKey=HeHeVaVqyVkntO2FnjQcs2Ilh/4MUD04y+AEhKp8z+g=;EntityPath=simulacoes

3. Modelo de Envelope para Simulação:

```
{
  "valorDesejado": 900.00,
  "prazo": 5
}
```

4. Modelo de Envelope de retorno para simulação:

```
{
  "idSimulacao": 20180702,
  "codigoProduto": 1,
  "descricaoProduto": "Produto 1",
  "taxaJuros": 0.0179,
  "resultadoSimulacao":
  [
    {
      "tipo": "SAC",
      "parcelas": [
        {
```

# Hackathon



```
        "numero": 1,
        "valorAmortizacao": 180.00,
        "valorJuros": 16.11,
        "valorPrestacao": 196.11
    },
    {
        "numero": 2,
        "valorAmortizacao": 180.00,
        "valorJuros": 12.89,
        "valorPrestacao": 192.89
    },
    {
        "numero": 3,
        "valorAmortizacao": 180.00,
        "valorJuros": 9.67,
        "valorPrestacao": 189.67
    },
    {
        "numero": 4,
        "valorAmortizacao": 180.00,
        "valorJuros": 6.44,
        "valorPrestacao": 186.44
    },
    {
        "numero": 5,
        "valorAmortizacao": 180.00,
        "valorJuros": 3.22,
        "valorPrestacao": 183.22
    }
]
},
{
    "tipo": "PRICE",
    "parcelas": [
        {
            "numero": 1,
            "valorAmortizacao": 173.67,
            "valorJuros": 16.11,
```



# Hackathon



```
        "valorPrestacao": 189.78
    },
    {
        "numero": 2,
        "valorAmortizacao": 176.78,
        "valorJuros": 13.00,
        "valorPrestacao": 189.78
    },
    {
        "numero": 3,
        "valorAmortizacao": 179.94,
        "valorJuros": 9.84,
        "valorPrestacao": 189.78
    },
    {
        "numero": 4,
        "valorAmortizacao": 183.16,
        "valorJuros": 6.62,
        "valorPrestacao": 189.78
    },
    {
        "numero": 5,
        "valorAmortizacao": 186.44,
        "valorJuros": 3.34,
        "valorPrestacao": 189.78
    }
]
}
```

## 5. Modelo de chamada para listar simulações

```
{
  "pagina":1,
  "qtdRegistros": 404,
  "qtdRegistrosPagina": 200,
  "registros": [
```

# Hackathon



```
{
  "idSimulacao": 20180702,
  "valorDesejado": 900.00,
  "prazo": 5,
  "valorTotalParcelas": 1243.28
}
```

6. Modelo de chamada para retornar o volume simulado por produto e por dia

```
{
  "dataReferencia": "2025-07-30",
  "simulacoes": [{
    "codigoProduto": 1,
    "descricaoProduto": "Produto 1",
    "taxaMediaJuro": 0.189,
    "valorMedioPrestacao": 300.00,
    "valorTotalDesejado": 12047.47,
    "valorTotalCredito": 16750.00
  }]
}
```

7. Modelo de chamada para retornar os dados de telemetria

```
{
  "dataReferencia": "2025-07-30",
  "listaEndpoints": [
    {
      "nomeApi": "Simulacao",
      "qtdRequisicoes": 135,
      "tempoMedio": 150, // em milisegundos,
      "tempoMinimo": 23,
      "tempoMaximo": 860,
    }
  ]
}
```



```
        "percentualSucesso": 0.98 // qtd de retorno 200 com  
relacao ao total  
    }  
]  
}
```

# Hackathon



## Front-end (React Native ou Angular 16+)

### **Desafio Técnico – App de Simulação de Empréstimos**

A CAIXA está evoluindo sua plataforma de crédito digital. O backend já está pronto e oferece endpoints para cadastrar produtos de empréstimo e simular financiamentos com base em taxa de juros anual e prazo.

Agora, o desafio é criar uma **experiência mobile completa** que permita:

- Cadastrar novos produtos de empréstimo
- Visualizar os produtos disponíveis
- Simular empréstimos com base nos dados cadastrados
- Exibir os resultados de forma clara, incluindo a memória de cálculo mês a mês

Imagine que Lucas, um microempreendedor, quer investir em novos equipamentos. Ele acessa o app, cadastra um novo produto de empréstimo com taxa personalizada, simula um financiamento e entende exatamente quanto pagará mês a mês. Seu app será a ponte entre a necessidade e a decisão financeira.

---

### Objetivo do Desafio

Criar um aplicativo React Native que permita:

#### 1. Cadastro de Produtos de Empréstimo

- Tela com formulário para:
- Nome do produto
- Taxa de juros anual (%)
- Prazo máximo (em meses)
- Enviar os dados para o backend via API

#### 2. Listagem de Produtos

- Buscar produtos cadastrados via API

# Hackathon



- Exibir nome, taxa de juros anual e prazo máximo

### 3. Simulação de Empréstimo

- Tela com formulário para:
- Selecionar um produto
- Informar valor do empréstimo
- Informar número de meses
- Enviar os dados para o endpoint de simulação
- Exibir:
- Dados do produto
- Taxa de juros efetiva mensal
- Valor total com juros
- Valor da parcela mensal
- Memória de cálculo mês a mês (juros, amortização, saldo)

---

#### Requisitos Técnicos

- Framework: React Native ou Angular 16+
- Estado: useState, useEffect, useReducer ou Context API
- Estilo: StyleSheet, Tailwind ou styled-components
- Testes: Jest e incluir para React - React Native Testing Library
- Cobertura mínima de 80%
- Integração com API: Fetch ou Axios
- Responsividade: Interface adaptável a diferentes tamanhos de tela

---

#### Endpoints esperados do backend



# Hackathon



- POST /produtos – Cadastrar produto
- GET /produtos – Listar produtos
- POST /simulações – Realizar simulação de empréstimo

---

## Critérios de Avaliação

- Funcionalidade completa e fluida
- Código limpo, modular e reutilizável
- Testes com cobertura mínima de 80%
- Clareza na exibição dos dados
- Boas práticas de componentes, hooks e integração com API
- Design de acordo com a marcas.caixa

---

## Exemplo de Resultado da Simulação

Produto: Empréstimo Pessoal

Valor solicitado: R\$ 10.000,00

Prazo: 12 meses

Taxa efetiva mensal: 1,39%

Parcela mensal: R\$ 931,50

Valor total com juros: R\$ 11.178,00

Memória de cálculo:

Mês 1: Juros R\$ 139,78 | Amortização R\$ 791,72 | Saldo: R\$ 9.208,28

Mês 2: Juros R\$ 128,74 | Amortização R\$ 802,76 | Saldo: R\$ 8.405,52



## **DevSecOps**

### **Desafio Técnico: Implantação Automatizada em Ambiente Orquestrado**

O desafio tem o objetivo de automatizar a implantação de uma aplicação Quarkus em um ambiente Kubernetes local, utilizando ferramentas de automação, simulando uma pipeline completa de entrega contínua. Esse desafio foca em práticas de DevOps, como entrega contínua, versionamento, e infraestrutura como código.

#### **Objetivo do Desafio**

Ao final do desafio, você deverá apresentar os seguintes entregáveis:

- Cluster Kubernetes local provisionado
- Pipeline local de automação
- Compilação da Aplicação: A aplicação Quarkus deve ser compilada e empacotada em um executável JAR (ou outro formato otimizado para container, se preferir).
- Imagem de container com o pacote gerado na compilação
- Aplicação Quarkus implantada nos ambientes:
  - DES (Desenvolvimento): para testes e validações.
  - PRD (Produção): simulação de ambiente final.
- Documentação técnica contendo:
  - Passo a passo da implantação.
  - Estratégia de versionamento.
  - Os códigos de automação desenvolvidos, scripts e *Docker file*.

#### **Recomendações**

- Documentação: Gere uma documentação com o passo a passo da instalação, Comandos utilizados, Estratégia de versionamento e como validar se a aplicação está funcionando.
- Ferramentas: Escolha ferramentas locais para simular o cluster Kubernetes e Docker.
- Ambiente: Defina e instale previamente as ferramentas que vão suportar a implantação.

#### **Requisitos Técnicos**



- Imagem de Container Base: [https://hub.docker.com/\\_/eclipse-temurin](https://hub.docker.com/_/eclipse-temurin)
- Código da aplicação: <https://github.com/quarkusio/quarkus-quickstarts/tree/main/getting-started>

#### Critérios de Avaliação

- Clareza e organização do pipeline.
- Qualidade da automação e cobertura das etapas.
- Uso de boas práticas de segurança e versionamento.
- Documentação clara e objetiva.
- Criatividade na escolha e integração das ferramentas.



## **Desafio Técnico: Level Up Financeiro**

No Brasil, milhões de pessoas enfrentam dificuldades para gerenciar suas finanças, seja por falta de conhecimento sobre orçamento, investimentos ou controle de gastos. Bancos e fintechs têm investido em soluções digitais para ajudar seus clientes a melhorarem sua saúde financeira, mas o engajamento com esses recursos ainda é um desafio.

Seu objetivo é projetar uma experiência gamificada que incentive e eduque clientes de um banco a melhorar sua saúde financeira de forma intuitiva e envolvente. O desafio deve considerar aspectos como:

- **Motivação do usuário:** Como engajar diferentes perfis de clientes a adotarem hábitos financeiros mais saudáveis?
- **Mecânicas de gamificação:** Como o design pode tornar a experiência mais atrativa sem comprometer a credibilidade do banco?
- **Impacto real:** Como garantir que a solução contribua para mudanças positivas no comportamento financeiro dos usuários?

### Tarefas para os Participantes

As etapas do Processo de UX CAIXA e suas atividades podem ser consultadas no endereço: [ux.caixa/](https://ux.caixa/)

[Os textos em azul devem ser retirados da resposta final]

### Parte 1: Pesquisa e Fundamentação Teórica

- Identificação do problema e do público-alvo

[Nesta etapa, espera-se que a pessoa aprofunde a compreensão do desafio proposto, definindo com clareza qual problema específico deseja resolver e para quem. É importante



ir além da descrição genérica do cenário e identificar nuances, como comportamentos, necessidades, dores e barreiras enfrentadas pelo público-alvo.

Resultado esperado: texto que descreva de forma clara o problema central, perfis de usuários (personas ou arquétipos), seus objetivos e motivações, além de dados ou evidências que sustentem essa definição.]

- Benchmark de soluções existentes

[Aqui a pessoa deve pesquisar e analisar soluções já existentes no mercado – tanto nacionais quanto internacionais – que utilizem gamificação ou outras estratégias para melhorar a saúde financeira. A análise deve destacar os pontos fortes e fracos de cada exemplo, assim como oportunidades não exploradas.

Resultado esperado: link para painel colaborativo (Miro, Figjam, Whiteboard etc.) com os prints das soluções encontradas, os destaques de cada uma delas e uma análise final. O acesso deve ser público para a correção do material.]

- Referências em UX relacionadas a gamificação e comportamento do usuário

[O objetivo desta etapa é reunir conceitos, boas práticas e frameworks de UX que possam orientar o design da solução. Isso inclui teorias de motivação, elementos de gamificação, heurísticas de interação e estudos sobre comportamento do usuário no contexto de finanças.

Resultado esperado: uma lista de referências teóricas e práticas (artigos, estudos, livros, exemplos de cases), organizada de forma que possa servir como base para as decisões de design e para justificar escolhas ao longo do projeto.]

## Parte 2: Estratégia e Proposta de Solução

- Definição da proposta de valor

[Nesta etapa, a pessoa deve estabelecer com clareza o que torna a solução única e relevante para o público-alvo identificado na etapa anterior. A proposta de valor deve responder à pergunta: "Por que o usuário se engajaria com essa experiência, em vez de não fazer nada ou usar outra solução?". O texto deve refletir benefícios claros para o usuário e





diferenciais competitivos, considerando tanto a dimensão funcional quanto a dimensão emocional.

Resultado esperado: texto claro e objetivo da proposta de valor, seus benefícios e relação explícita com os problemas e motivações identificados na etapa de Pesquisa e Fundamentação Teórica.]

- Jornada do usuário e principais fluxos da experiência

[Aqui, a pessoa deve mapear o passo a passo da interação do usuário com a solução – desde o primeiro contato até a obtenção do resultado esperado. A jornada deve ser representada de forma visual e acompanhada de descrições que facilitem a compreensão.

Resultado esperado: link para mapa visual da jornada do usuário (Miro, Figjam, Whiteboard etc.), com destaque para os pontos críticos onde a gamificação será aplicada e fluxos definidos.]

- Escolha e justificativa das mecânicas de gamificação

[Nesta etapa, a pessoa deve selecionar quais elementos de gamificação serão utilizados e justificar a escolha com base no perfil do público-alvo, na proposta de valor e nas boas práticas levantadas na etapa de pesquisa. A justificativa deve considerar tipos de motivação, impacto esperado no comportamento do usuário e coerência com os objetivos da solução.

Resultado esperado: lista das mecânicas envolvidas, explicação do papel de cada mecânica na experiência do usuário, justificativa baseada nas referências listadas na etapa anterior e evidência do alinhamento com a proposta de valor e a jornada do usuário.]

### Parte 3: Protótipo Interativo

- Wireframes ou mockups de alta fidelidade

[Nesta etapa, a pessoa deve criar representações visuais de telas-chave da solução. Essas representações podem ser wireframes ou mockups. As telas escolhidas devem mostrar momentos essenciais da experiência.



Resultado esperado: link para o Figma contendo no mínimo 3 telas representando diferentes momentos da jornada do usuário. O acesso deve ser público para a correção do material.]

- Link para protótipo navegável

[O objetivo desta etapa é criar um protótipo interativo que permita ao avaliador simular a navegação pela solução. O protótipo deve ser suficientemente funcional para demonstrar o fluxo principal e permitir ao usuário experimentar as mecânicas de gamificação, mesmo que de forma simplificada.

Resultado esperado: link para o Figma com fluxo navegável. O acesso deve ser público para a correção do material.]

- Explicação das decisões de design e interação

[Nesta etapa, a pessoa deve documentar o raciocínio por trás das escolhas de design (cores, tipografia, layout, elementos visuais) e de interação (fluxos, micro interações, animações). A explicação deve demonstrar alinhamento com o público-alvo, proposta de valor e jornada definida anteriormente, além de considerar boas práticas de UX e acessibilidade.

Resultado esperado: texto explicando cada decisão relevante de design e interação, justificativas baseadas em princípios de UX e conexão explícita com as necessidades e motivações do público-alvo.