

Faculdade

**XPe**



# RELATÓRIO

---

PROJETO  
APLICADO

**XP Educação**  
**Relatório do Projeto Aplicado**

# **Modernização da Arquitetura do Sistema de Simulados Colaborativos**

**Michele Cristina da Paixão Silva**

**Orientador(a): Vinicius Fernandes De Almeida Terço**

**Novembro de 2024**



MICHELE CRISTINA DA PAIXÃO SILVA

XP EDUCAÇÃO

RELATÓRIO DO PROJETO APLICADO

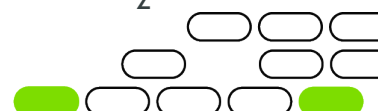
# Modernização da Arquitetura do Sistema de Simulados Colaborativos

Relatório de Projeto Aplicado desenvolvido para fins de conclusão do curso de Pós-graduação em Arquitetura de Software e Soluções.

Orientador (a): Vinicius Fernandes De Almeida Terço

São Paulo

Novembro de 2024



# Sumário

1. CANVAS do Projeto Aplicado.....	4
1.1 Desafio.....	6
1.1.1 Análise de Contexto.....	6
1.1.2 Personas.....	9
1.1.3 Benefícios e Justificativas.....	14
1.1.4 Hipóteses.....	16
1.2 Solução.....	18
1.2.1 Objetivo SMART.....	18
1.2.2 Premissas e Restrições.....	19
1.2.3 Backlog de Produto.....	21
2. Área de Experimentação.....	22
2.1 Sprint 1.....	22
2.1.1 Solução.....	22
Evidência do planejamento:.....	22
Evidência da execução de cada requisito:.....	23
Evidência dos resultados:.....	27
2.1.2 Lições Aprendidas.....	37
2.2 Sprint 2.....	38
2.2.1 Solução.....	38
Evidência do planejamento:.....	38
Evidência da execução de cada requisito:.....	38
Evidência dos resultados:.....	38
2.2.2 Lições Aprendidas.....	38
2.3 Sprint 3.....	39
2.3.1 Solução.....	39
Evidência do planejamento:.....	39
Evidência da execução de cada requisito:.....	39
Evidência dos resultados:.....	39
2.3.2 Lições Aprendidas.....	39
3. Considerações Finais.....	40
3.1 Resultados.....	40
3.2 Contribuições.....	40
3.3 Próximos passos.....	40



# 1. CANVAS do Projeto Aplicado

Figura conceitual, que representa todas as etapas do Projeto Aplicado.



Figura 1: Canvas do Projeto Aplicado

Abaixo estão preenchidas as mesmas informações do CANVAS, com o objetivo de facilitar a leitura:

- **Título/Versão:** Modernização da Arquitetura do Sistema de Simulados Colaborativos
- **Análise de Contexto:** Criar arquitetura escalável, disponibilizar sistema em outras plataformas
- **Personas:** CIO da empresa, arquiteto de software, desenvolvedor de software, usuário (estudante)
- **Benefícios e Justificativas:** Colaboração entre estudantes, disponibilidade do sistema, usabilidade do sistema
- **Hipóteses:** Falta de perguntas, custos com recursos sistêmicos ociosos, problemas de usabilidade

- **Objetivo SMART:** Criar uma arquitetura de referência que se adéque à demanda dos usuários, utilizando as melhores práticas, em 2 meses
- **Premissas e Restrições:** A arquitetura deve ser desenvolvida durante o Projeto Aplicado, deve ser garantida alta disponibilidade
- **Backlog:**
  - Modelo C4 da arquitetura atual;
  - Modelo C4 até nível 3 da nova arquitetura;
  - Definir requisitos arquiteturais para backend, frontend e mobile;
  - Definir tecnologia para IA;
  - Definir tecnologia para salas virtuais;
- **Área de Experimentação:** esta parte do CANVAS será preenchida no decorrer das sprints e finalização do Projeto Aplicado
- **Resultado:** esta parte do CANVAS será preenchida no decorrer das sprints e finalização do Projeto Aplicado
- **Solução:** esta parte do CANVAS será preenchida no decorrer das sprints e finalização do Projeto Aplicado
- **Lições Aprendidas:** esta parte do CANVAS será preenchida no decorrer das sprints e finalização do Projeto Aplicado

Nas seções deste documento, as informações preenchidas no CANVAS são apresentadas com mais detalhes.



## 1.1 Desafio

### 1.1.1 Análise de Contexto

Para obter uma certificação, por exemplo “AWS Cloud Practitioner”, “Microsoft Certified Solutions Associate (MCSA)”, ou até mesmo para realizar uma prova para um vestibular, além do estudo teórico (e às vezes prático), muitos estudantes optam por realizar simulados para entender a estrutura da prova que realizará e aumentar seu índice de aprovação.

Pensando em melhorar a experiência dos estudantes e torná-la mais interativa, a empresa “Junes From Inaba Corp” criou um sistema de simulados chamado “Sistema de Simulados Colaborativos”. Suas principais funcionalidades são:

- Cadastro de perguntas de um determinado tipo de prova pelos próprios usuários;
- Registro de feedbacks das perguntas cadastradas;
- Gerar simulados de um determinado tema;
- Registrar as respostas do usuário para o simulado criado;
- Gerar gráficos com o desempenho do usuário de acordo com simulados realizados;

A empresa construiu o sistema utilizando uma arquitetura monolítica on-premises, utilizando React no frontend, sem considerar a responsividade do sistema e ou uma versão mobile. Já o backend foi construído utilizando NodeJS.

Devido à arquitetura atual do sistema, foi identificado, através de ferramentas de observabilidade que os recursos computacionais designados para o “Sistema de Simulados Colaborativos” ficam ociosos 80% do tempo, pois o sistema não possui a elasticidade necessária para aumentar ou diminuir recursos de acordo com a demanda dos usuários.

Alguns relatórios apontam que muitos usuários abandonam o sistema quando um tema não conta com a quantidade suficiente de perguntas cadastradas para geração de simulados. Além disso, os próprios usuários relataram que gostariam de uma funcionalidade que possibilitasse a interação em tempo real entre estudantes que desejam praticar um tema em conjunto.



Os problemas e necessidades foram identificados e mapeados através de reuniões de alinhamento com a área de negócio responsável pelo “Sistema de Simulados Colaborativos” da empresa “Junes From Inaba Corp” e por feedbacks dos usuários (em geral, estudantes que efetuam simulados).

O primeiro passo para analisar o problema que este projeto resolverá foi a utilização da matriz CSD. Essa ferramenta permite explorar três questões essenciais do projeto:

1. Certezas, ou seja, o que já se sabe sobre o problema/desafio;
2. Suposições, ou seja, as hipóteses relacionadas ao problema/desafio; e
3. Dúvidas, de forma que ainda existem perguntas a serem respondidas.

Nessa matriz, também é possível analisar o problema sob diferentes óticas, sendo as mais comuns:

1. Atores, que são as pessoas envolvidas no problema;
2. Cenários, onde ocorrem os problemas; e
3. Regras, que representam a relação entre os atores e os cenários.

A figura a seguir ilustra a matriz CSD preenchida para a análise do problema no contexto apresentado para o “Sistema de Simulados Colaborativos”:





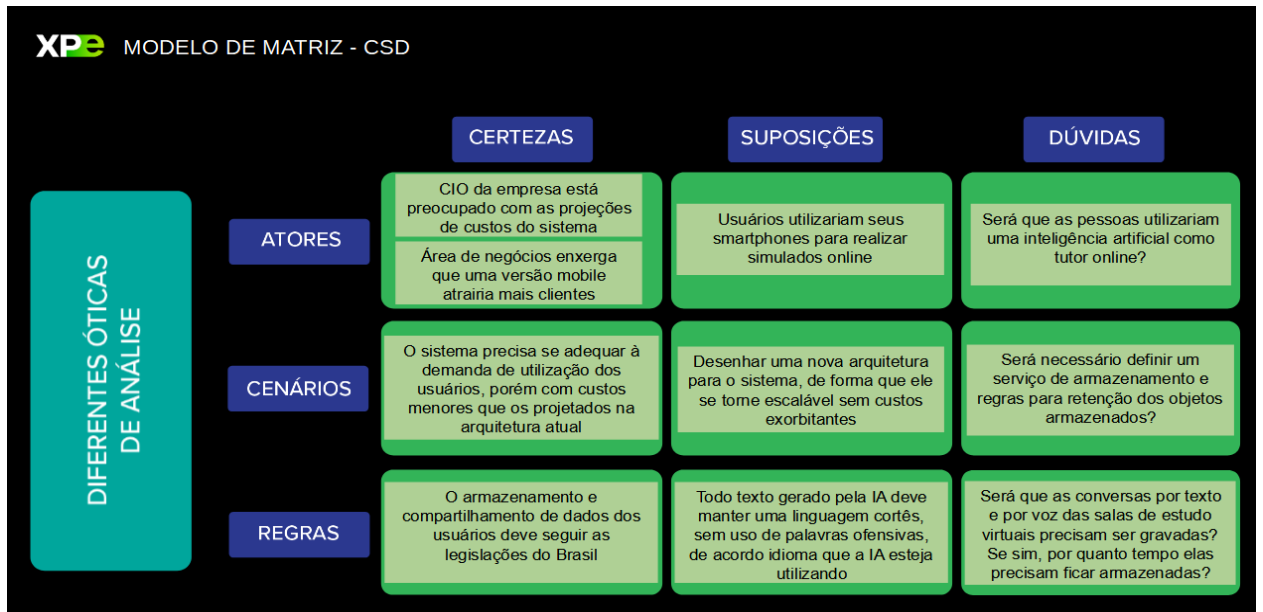


Figura 2: Matriz CSD

Para realizar uma imersão profunda no problema com o objetivo de ter uma visão sistêmica do ambiente no qual o desafio está situado, será utilizada a ferramenta de observação POEMS, que é um acrônimo para:

- **People (Pessoas):** quem são as pessoas que estão neste local e por que elas estão lá?
- **Objects (Objetos):** quais objetos (máquinas, equipamentos, etc.) estão presentes? Existe relação entre eles?
- **Environment (Ambiente):** em quais locais as atividades acontecem? Existem diferentes ambientes no contexto em questão?
- **Messages (Mensagens):** quais mensagens estão sendo comunicadas e como são transmitidas?
- **Services (Serviços):** quais são os serviços oferecidos nos diferentes ambientes?

A figura a seguir ilustra a observação POEMS preenchida para a análise do problema no contexto apresentado para o “Sistema de Simulados Colaborativos”:

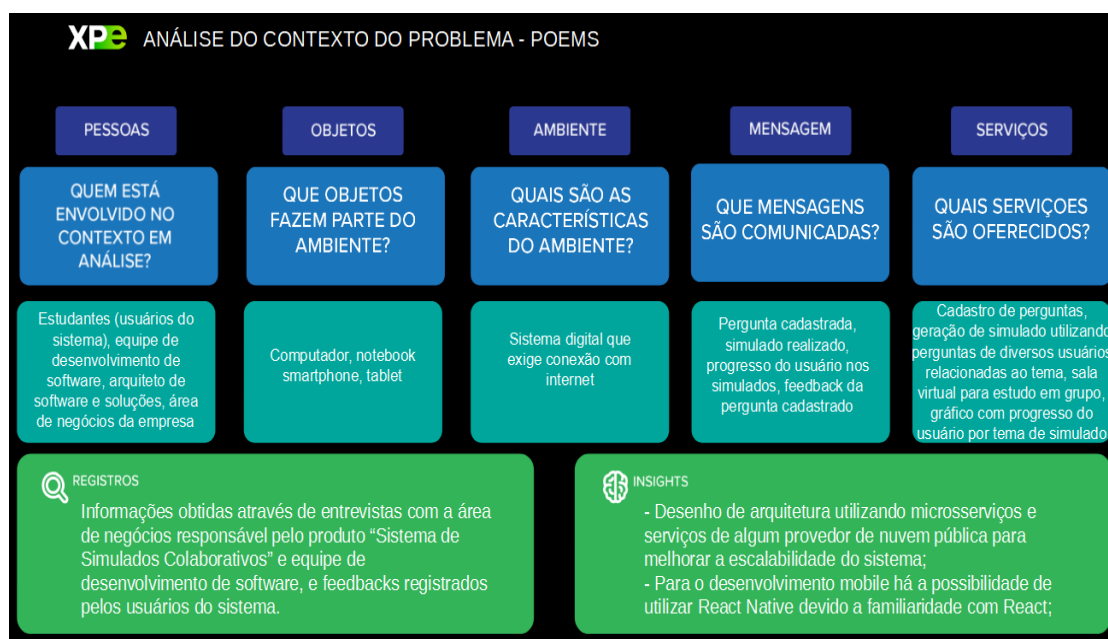


Figura 3: Observação POEMS

### 1.1.2 Personas

Para compreender os sentimentos e desejos do público-alvo, foi utilizada a ferramenta Mapa de Empatia. Ele permite que entremos na mente do nosso público-alvo, compreendendo seus desejos de maneira a direcionar os produtos e serviços que sejam mais adequados ao contexto.

Para cada mapa de empatia, vamos criar uma persona, que é uma representação fictícia das características de um usuário ideal para este projeto aplicado. Todas as personas neste documento são fictícias e foram criadas através do site "Gerador de Personas" (<https://www.geradordepersonas.com.br/>).

## Mapa de Empatia: CIO da Empresa



Figura 4: Mapa de Empatia - CIO da Empresa

## Persona: CIO da Empresa

**JEFFREY PRESTON JORGENSEN**  
CIO (CHIEF INFORMATION OFFICER)  
ADULTO DE MEIA-IDADE (41-59)

**Mini-bio** | Jeffrey gosta de passar seu tempo livre criando projetos paralelos. Ele tem o sonho de viajar para o espaço pelo menos uma vez com sua família.

**Detalhes Pessoais**

<b>Localização</b> Brasil	<b>Renda Familiar</b> Acima de R\$10.000,00	<b>Nível Educacional</b> Graduação em engenharia elétrica e informática	<b>Status de Relacionamento</b> Divorciado/Separado
------------------------------	--	--	--

**Carreira**

<b>Empresa</b> Junes From Inaba Corp	<b>Tamanho da Empresa</b> Grande	<b>Responsabilidades Profissionais</b> Definir a estratégia de tecnologia e supervisionar as operações de infraestrutura, sistemas de computação e outras necessárias
<b>Objetivos</b> Garantir que a TI e seus recursos sejam os mais eficientes possíveis	<b>Desafios</b> Incentivar inovação na área de TI, tentando manter os níveis de desperdícios menores possíveis	

**Canais de Comunicação**

Figura 5: Persona - CIO da Empresa

## Mapa de Empatia: Arquiteto de software

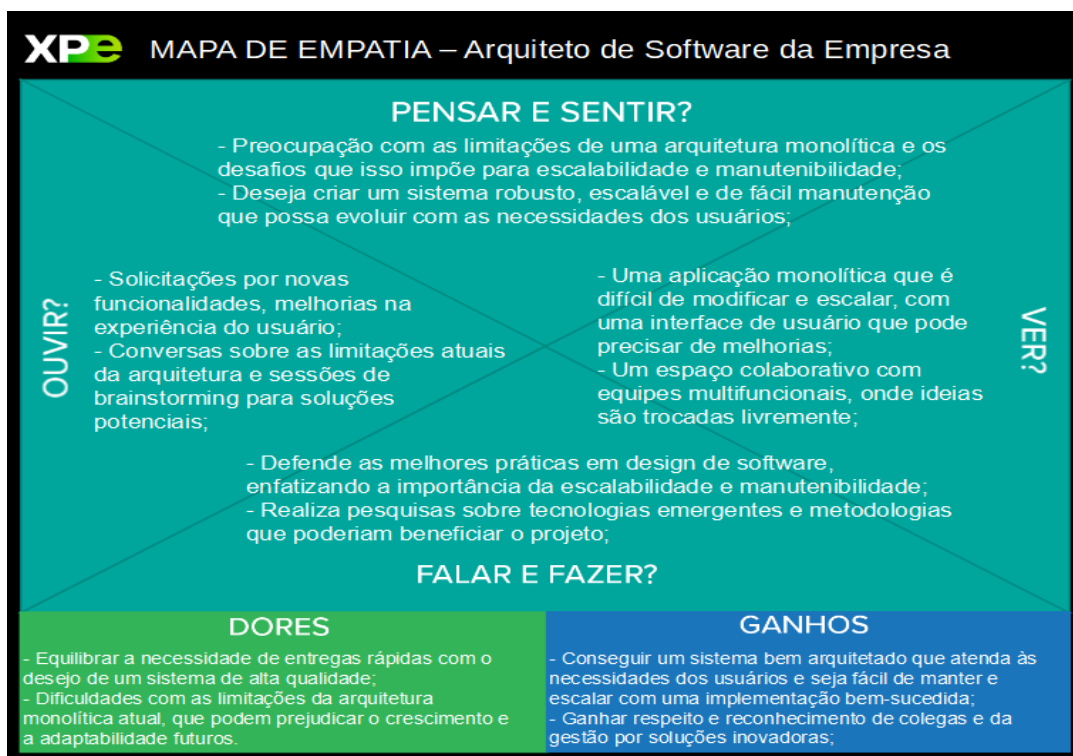


Figura 6: Mapa de Empatia - Arquiteto de software

## Persona: Arquiteto de software

### JACK FROST

ARQUITETO DE SOFTWARE

ADULTO (26-40)

**Mini-bio**

Jack Frost, 35 anos, arquiteto de software visionário. Apaixonado por inovação, ele cria soluções escaláveis e eficientes, sempre em busca de tecnologias que transformem a experiência do usuário.

**Detalhes Pessoais**

<b>Localização</b>	<b>Renda Familiar</b>	<b>Nível Educacional</b>	<b>Status de Relacionamento</b>
Brasil	Acima de R\$10.000,00	Pós graduado em arquitetura de software e soluções	É complicado

**Carreira**

<b>Empresa</b>	<b>Tamanho da Empresa</b>	<b>Responsabilidades Profissionais</b>
Junes From Inaba Corp	Grande	Defender as melhores práticas em design de software, enfatizando a importância da escalabilidade e manutenibilidade
<b>Objetivos</b>	<b>Desafios</b>	
Garantir que a arquitetura suporte as funcionalidades do sistema e aumentam o engajamento dos usuários	Equilibrar a necessidade de entregas rápidas com o desejo de um sistema de alta qualidade	

**Canais de Comunicação**

Figura 7: Persona - Arquiteto de software

## Mapa de Empatia: Desenvolvedor de software

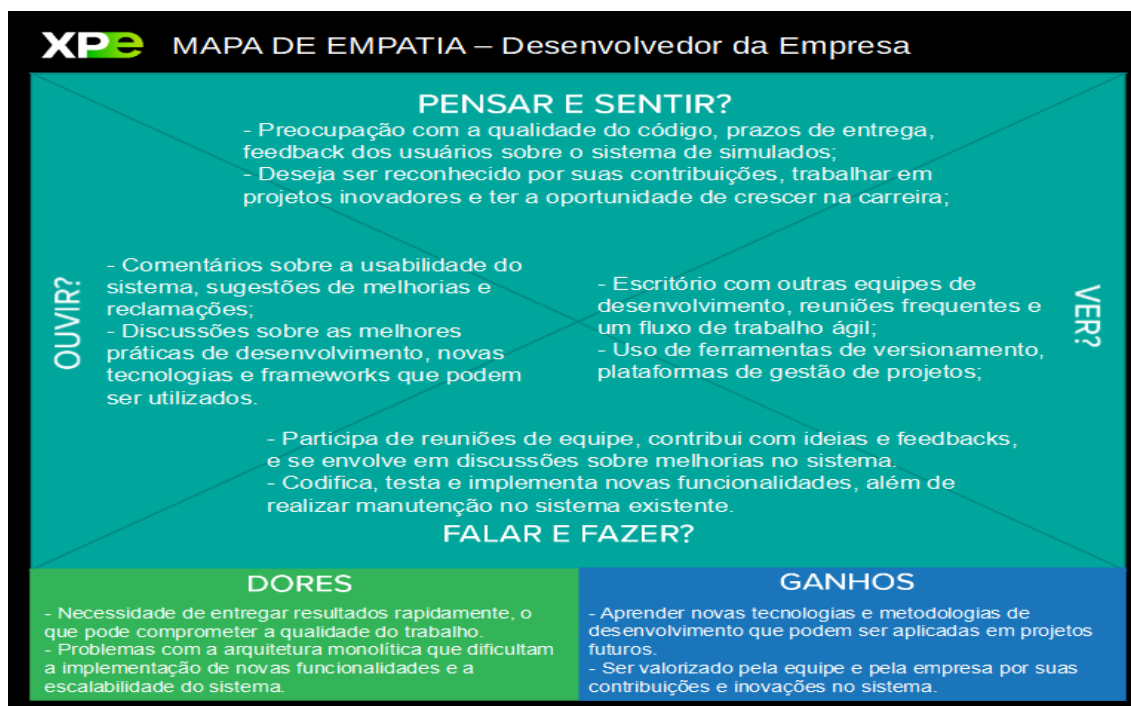


Figura 8: Mapa de Empatia - Desenvolvedor de software

## Persona: Desenvolvedor de software

**KRYSTAL JUNG**  
DESENVOLVEDORA DE SOFTWARE  
ADULTO (26-40)

**Mini-bio** | Krystal, 30 anos, desenvolvedora de software apaixonada por tecnologia. Adora resolver problemas, colaborar em equipe e criar soluções que transformam a experiência de aprendizado dos estudantes.

**Detalhes Pessoais**

<b>Localização</b> Brasil	<b>Renda Familiar</b> De R\$6.500 a R\$10.000,00	<b>Nível Educacional</b> Graduação em engenharia de software	<b>Status de Relacionamento</b> Em um relacionamento
------------------------------	---	---	---

**Carreira**

<b>Empresa</b> Junes From Inaba Corp	<b>Tamanho da Empresa</b> Grande	<b>Responsabilidades Profissionais</b> Codificar, testar e implementar novas funcionalidades, além de realizar manutenção no sistema existente.
<b>Objetivos</b> Ser reconhecida por suas contribuições, trabalhar em projetos inovadores e ter a oportunidade de crescer na carreira	<b>Desafios</b> Entregar resultados rapidamente, o que pode comprometer a qualidade do trabalho	

**Canais de Comunicação**

✉️ in 📷 📺 📺 📺 📺 📺

Figura 9: Persona - Desenvolvedor de software

Mapa de Empatia: Usuário do sistema (estudante)

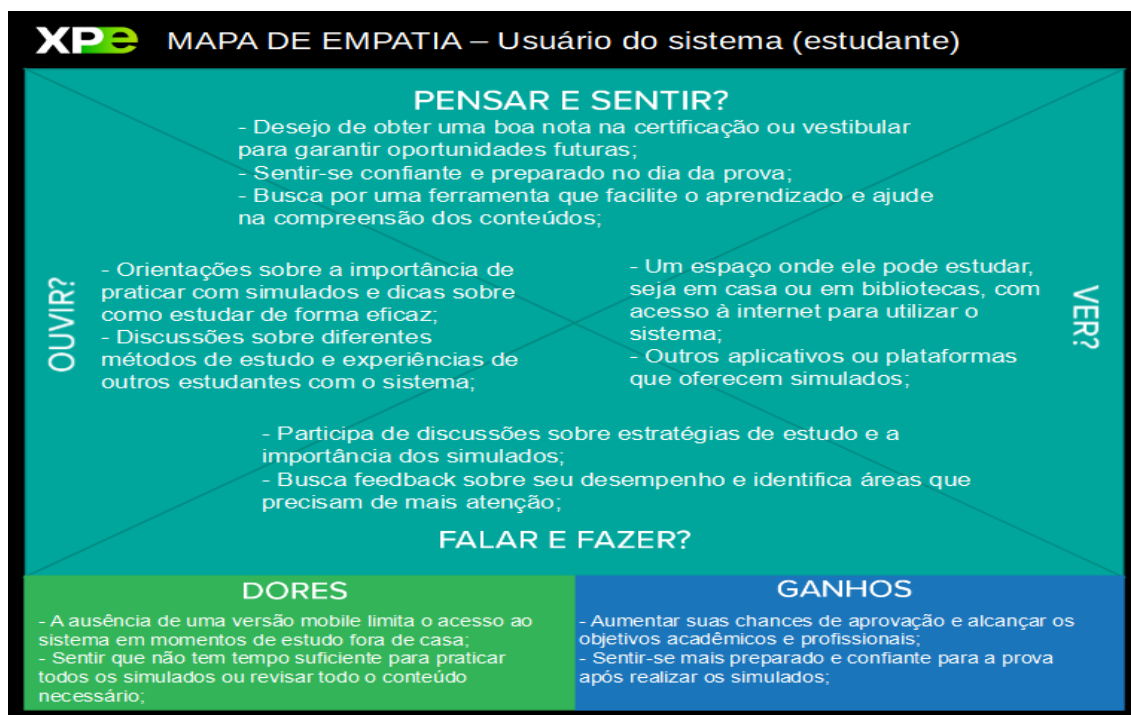


Figura 10: Mapa de Empatia - Usuário do sistema (estudante)

Persona: Usuário do sistema (estudante)

**TOBIRAMA SENJU**  
ESTUDANTE / DESENVOLVEDOR DE SOFTWARE  
JOVEM ADULTO (19-25)

**Mini-bio** | Tobirama, 24 anos, estudante dedicado em busca da certificação AWS. Apaixonado por tecnologia, ele utiliza simulados para se preparar e se destacar no mercado de trabalho em nuvem.

**Detalhes Pessoais**

<b>Localização</b> Brasil	<b>Renda Familiar</b> De R\$3.501,00 a R\$6.500,00	<b>Nível Educacional</b> Graduação em Ciência da Computação	<b>Status de Relacionamento</b> Solteiro(a)
------------------------------	---	--	--

**Carreira**

<b>Empresa</b> Anbu Konoha	<b>Tamanho da Empresa</b> Pequeno	<b>Responsabilidades Profissionais</b> Buscar feedback sobre seu desempenho e identificar áreas que precisam de mais atenção
<b>Objetivos</b> Passar na prova desejada e alcançar uma certificação que abra portas para uma carreira melhor	<b>Desafios</b> Conciliar estudos, trabalho e vida pessoal	

**Canais de Comunicação**

✉️ 🌐 📷 📺 📠

Figura 11: Persona - Usuário do sistema (estudante)



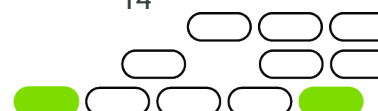
### 1.1.3 Benefícios e Justificativas

Para uma análise mais assertiva sobre os benefícios e justificativas deste projeto aplicado, serão utilizadas as seguintes ferramentas:

- Business Design Blueprint (também conhecido apenas como Blueprint): metodologia que nos permite explorar o dia a dia do cliente com o objetivo de planejar a solução em detalhes.
- CANVAS de Proposta de Valor: responsável por explicar por que um cliente escolhe uma solução em vez de outras disponíveis no mercado. Representa os benefícios oferecidos a um segmento de clientes, a um preço que eles estão dispostos a pagar.

As imagens a seguir representam o Blueprint deste Projeto Aplicado:

Modernização da Arquitetura do Sistema de Simulados Colaborativos						
Ações do estudante	Cadastro de Perguntas	Realização de Simulados	Fornecimento de Feedback sobre Perguntas	Análise de Desempenho	Colaboração com Colegas	Acesso ao Sistema em Dispositivos Móveis
<b>Objetivos</b>	Contribuir para o banco de perguntas para simulações futuras. Garantir uma variedade de perguntas em diversos tópicos.	Avaliar o conhecimento e a preparação para exames reais. Familiarizar-se com o formato do exame e tipos de perguntas.	Melhorar a qualidade das perguntas no sistema. Ajudar outros usuários destacando os pontos fortes e fracos das perguntas.	Identificar pontos fortes e fracos nas áreas de conhecimento. Rastrear o progresso ao longo do tempo para aprimorar estratégias de estudo.	Aprimorar o aprendizado por meio de estudo em grupo e discussão. Fomentar um senso de comunidade entre os usuários.	Aumentar a acessibilidade e a conveniência de uso do sistema. Permitir que os usuários estudem em qualquer lugar.
<b>Atividades</b>	Acessar a interface de cadastro de perguntas. Inserir o conteúdo da pergunta, opções de resposta e respostas corretas. Enviar a pergunta para aprovação ou uso direto.	Selecionar um tópico ou tema específico para o simulado. Completar o teste dentro de um limite de tempo definido. Enviar as respostas para avaliação.	Revisar perguntas após realizar um simulado. Avaliar perguntas e fornecer comentários ou sugestões.	Acessar gráficos e relatórios de desempenho. Revisar resultados de simulações passadas e identificar tendências.	Criar ou ingressar em grupos de estudo virtuais. Participar de discussões ao vivo e testes colaborativos.	Baixar e instalar o aplicativo móvel (uma vez disponível). Acessar todas as funcionalidades disponíveis na versão web.
<b>Questões</b>	Como posso garantir que minhas perguntas atendam aos padrões de qualidade? Quais tópicos estão mais em demanda para o cadastro de perguntas?	Como interpreto meus resultados de desempenho? Há perguntas suficientes disponíveis para os tópicos que quero estudar?	Como meu feedback é utilizado para melhorar o sistema? Receberei reconhecimento por minhas contribuições?	Como posso usar meus dados de desempenho para melhorar meus hábitos de estudo? Quais áreas específicas preciso focar com base nos meus resultados?	Como encontro colegas com objetivos de estudo semelhantes? Quais ferramentas estão disponíveis para colaboração eficaz (áudio, chat)?	A versão móvel terá os mesmos recursos que a versão web? Como posso garantir que meus dados sejam sincronizados entre dispositivos?
<b>Barreiras</b>	Falta de orientação sobre como criar perguntas de qualidade. Possível relutância em contribuir devido a mecanismos de feedback pouco claros.	Número insuficiente de perguntas para certos tópicos. Problemas técnicos ou falta de uma interface amigável.	Processo de feedback pouco claro pode desestimular os usuários a fornecerem input. Preocupações sobre o impacto do feedback nas perguntas futuras.	Complexidade da apresentação dos dados pode dificultar a compreensão. Falta de insights acionáveis a partir da análise de desempenho.	Funcionalidade limitada para interação em tempo real (por exemplo, áudio e chat). Dificuldade potencial em coordenar horários com colegas.	Falta de uma versão móvel pode limitar o engajamento do usuário. Preocupações sobre o desempenho do aplicativo móvel.



Modernização da Arquitetura do Sistema de Simulados Colaborativos						
Ações do estudante	Cadastro de Perguntas	Realização de Simulados	Fornecimento de Feedback sobre Perguntas	Análise de Desempenho	Colaboração com Colegas	Acesso ao Sistema em Dispositivos Móveis
<b>Funcionalidades</b>	Interface do usuário para inserir os detalhes da pergunta (texto, opções, resposta correta). Verificações de validação para garantir que as perguntas atendam aos padrões de qualidade. Opção de categorizar as perguntas por tópico ou nível de dificuldade.	Seleção de tópicos e temas para o simulado. Funcionalidade de cronômetro para simular condições reais de exame. Pontuação automática e feedback após a conclusão.	Opção de avaliar perguntas (por exemplo, polegar para cima/baixo, classificações em estrelas). Caixa de texto para comentários e sugestões detalhadas. Visualização de feedbacks anteriores sobre as perguntas.	Geração de gráficos e relatórios de desempenho com base nos resultados dos simulados. Comparação do desempenho atual com desempenhos passados. Sugestões de melhoria com base na análise.	Criação de grupos de estudo ou salas virtuais. Funcionalidade de chat e áudio em tempo real para discussões em grupo. Acesso compartilhado a simulações dentro do grupo.	Interface amigável para dispositivos móveis para todas as funcionalidades existentes.
<b>Interação</b>	Os usuários podem preencher um formulário para enviar perguntas. Os usuários recebem feedback imediato sobre a qualidade de sua submissão (por exemplo, erros, sugestões).	Os usuários podem iniciar um simulado com um clique. Os usuários recebem um resumo de suas respostas e resultados após completar o teste.	Os usuários podem fornecer feedback imediatamente após realizar um teste. Os usuários recebem confirmação de que seu feedback foi registrado.	Os usuários podem navegar até um painel de desempenho. Os usuários podem clicar em métricas específicas para obter insights detalhados.	Os usuários podem convidar outros para participar do seu grupo de estudo. Os usuários podem se comunicar via texto ou voz durante a sessão.	Os usuários podem fazer login e navegar no aplicativo móvel da mesma forma que na versão web. Os usuários podem receber atualizações e notificações diretamente em seus dispositivos móveis.
<b>Mensagem</b>	"Sua pergunta foi enviada com sucesso para revisão. Obrigado por contribuir!"	"Você completou o simulado! Aqui está o seu desempenho: [Resumo dos resultados]."	"Obrigado pelo seu feedback! Ele nos ajuda a melhorar a qualidade das nossas perguntas."	"Aqui está sua análise de desempenho. Concentre-se nessas áreas para melhorar: [Áreas específicas]."	"Você criou um grupo de estudo com sucesso! Convide seus amigos para participar e comece a colaborar."	"Bem-vindo à versão móvel! Agora você pode estudar a qualquer hora, em qualquer lugar."

Modernização da Arquitetura do Sistema de Simulados Colaborativos						
Ações do estudante	Cadastro de Perguntas	Realização de Simulados	Fornecimento de Feedback sobre Perguntas	Análise de Desempenho	Colaboração com Colegas	Acesso ao Sistema em Dispositivos Móveis
<b>Onde Ocorre</b>	Interface web do sistema	Interface web do sistema	Interface web do sistema, após a realização do simulado.	Interface web do sistema, na seção de desempenho.	Interface web do sistema e salas virtuais.	Aplicativo móvel (futuro) ou versão responsiva do site.
<b>Tarefas Aparentes</b>	Acessar a interface de cadastro. Preencher o formulário com a pergunta, opções e resposta correta. Enviar a pergunta para revisão.	Selecionar o tema do simulado. Iniciar o simulado e responder às perguntas. Submeter o simulado após a conclusão.	Avaliar as perguntas respondidas. Fornecer feedback (comentários e classificações).	Acessar a seção de análise de desempenho. Visualizar gráficos e relatórios de desempenho.	Criar ou ingressar em um grupo de estudo. Participar de discussões e responder simulados em grupo.	Baixar o aplicativo ou acessar a versão móvel do site. Navegar pelas funcionalidades disponíveis no dispositivo móvel.
<b>Tarefas Escondidas</b>	Pesquisar sobre a qualidade e relevância da pergunta. Revisar e editar a pergunta antes do envio.	Preparar-se mentalmente para o simulado. Gerenciar o tempo durante a realização do simulado.	Refletir sobre a clareza e relevância das perguntas. Considerar como o feedback pode impactar a qualidade do sistema.	Interpretar os dados apresentados. Identificar áreas de melhoria com base nos resultados.	Gerenciar dinâmicas de grupo e interações. Estabelecer regras de colaboração e comunicação.	Adaptar-se à interface móvel e suas limitações. Sincronizar dados entre dispositivos (se aplicável).
<b>Processos de Suporte</b>	Sistema de validação de perguntas para garantir a qualidade. Banco de dados para armazenar perguntas cadastradas.	Sistema de cronometragem para simular condições reais de exame. Algoritmo para gerar perguntas com base no tema escolhido.	Sistema de coleta e análise de feedback. Banco de dados para armazenar feedbacks fornecidos.	Algoritmos para gerar gráficos de desempenho. Sistema de comparação de resultados com simulações anteriores.	Sistema de chat e áudio para comunicação em tempo real. Algoritmo para gerenciar as salas virtuais.	Desenvolvimento de uma interface responsiva ou aplicativo móvel. Sistema de notificações push para atualizações e lembretes.
<b>Saída Desejada</b>	Perguntas cadastradas e aprovadas que possam ser utilizadas em simulados.	Resultados do simulado, incluindo pontuação e feedback.	Feedback registrado que pode ser utilizado para melhorar as perguntas.	Relatórios de desempenho que ajudam na identificação de pontos fortes e fracos.	Experiência de estudo colaborativo que aumenta a motivação e o aprendizado.	Acesso completo ao sistema de simulados em qualquer lugar e a qualquer momento.





A imagem a seguir representa o CANVAS de Proposta de Valor deste Projeto Aplicado:



Figura 12: CANVAS de Proposta de Valor do Projeto Aplicado

#### 1.1.4 Hipóteses

A tabela a seguir possui as informações coletadas e estruturadas em uma matriz de observações e hipóteses, que é uma ferramenta eficaz para sintetizar o material existente até o momento:

Observação	Hipóteses
Não ter perguntas suficientes cadastradas para realizar o simulado de um tema	Implementar uma funcionalidade de geração automática de perguntas utilizando inteligência artificial. Isso pode aumentar a quantidade de perguntas disponíveis e garantir uma variedade maior de temas.
Falta de interação/colaboração entre os usuários	Criar uma funcionalidade de salas virtuais onde os usuários possam realizar simulados em grupo, com suporte para chat de texto e áudio. Isso pode aumentar a interação e o engajamento entre os usuários.
Ociosidade dos recursos de infraestrutura, gerando custos desnecessários	Migrar o sistema para uma arquitetura de microsserviços na nuvem, que permita escalar os recursos de forma mais eficiente, reduzindo custos durante períodos de baixa demanda.

Problemas de usabilidade da plataforma	Tornar a versão web responsiva. Disponibilizar uma versão mobile para usuários.
--	---

A partir da matriz acima, as hipóteses serão priorizadas por meio da Matriz BASICO. BASICO é um acrônimo para: Benefícios (B), Abrangência (A), Satisfação (S), Investimentos (I), Cliente (C) e Operacionalidade (O). Cada critério é avaliado em uma escala de 1 a 5, na qual 1 representa a menor avaliação e 5, a maior.

A tabela a seguir representa a priorização das hipóteses utilizando a Matriz BASICO:

Ideias	B	A	S	I	C	O	Soma	Priorização
Disponibilizar geração automática de perguntas com inteligência artificial	4	5	4	3	3	2	21	4
Disponibilizar salas virtuais (chat de texto e áudio) para realizar simulados em grupo	4	4	4	2	3	2	19	5
Migrar arquitetura do sistema para microserviços na nuvem	5	5	5	2	5	2	24	1
Tornar versão web responsiva	4	5	4	4	3	4	24	2
Criar versão mobile	4	4	4	4	4	2	22	3

Para estes cinco critérios de análise, os balizadores indicados para as notas são apresentados na tabela a seguir:

Escala	B - Benefícios	A - Abrangência	S - Satisfação	I - Investimentos	C - Cliente	O - Operacionalidade
5	De vital importância	Total (de 70 a 100%)	Muito grande	Pouquíssimo investimento	Nenhum impacto	Muito fácil
4	Significativo	Muito grande (de 40 a 70%)	Grande	Alguns investimentos	Impacto pequeno	Fácil
3	Razoável	Razoável (de 20 a 40%)	Média	Médio investimento	Médio impacto	Média facilidade
2	Poucos benefícios	Pequena (de 5 a 20%)	Pequena	Alto investimento	Impacto grande	Difícil
1	Algum benefício	Muito pequena	Quase não é notada	Altíssimo investimento	Impacto muito grande no cliente	Muito difícil

Fonte: <https://engenhariaexercicios.com.br/gestao-de-qualidade/matriz-gut-basico-conceito-aplicacao-das-matrizes-priorizacao/>.



## 1.2 Solução

### 1.2.1 Objetivo SMART

Para alinhar as expectativas, de maneira clara e objetiva, visando maximizar as chances de alcançar os resultados esperados, o objetivo será escrito de maneira SMART, ou seja, Específico (Specific), Mensurável (Measurable), Atingível (Attainable), Relevante (Relevant) e Temporal (Time-based).

- S (Specific - Específico): elaborar a documentação de arquitetura de software que considere a migração do “Sistema de Simulados Colaborativos” para uma arquitetura de microsserviços na nuvem com uma versão mobile
- M (Measurable - Mensurável): reduzir o tempo em que os recursos ficam ociosos de 80% para 20%
- A (Attainable - Atingível): orientar os desenvolvedores a construírem uma solução que utilize as melhores práticas e otimize os recursos
- R (Relevant - Relevante): adequar-se à demanda dos usuários por uma experiência interativa e acessível a qualquer momento
- T (Time based - Temporal): em até três sprints, ou seja, 2 meses.

*Criar uma arquitetura de referência que considere a migração do “Sistema de Simulados Colaborativos” (S) em 2 meses (T), de forma que os recursos que ficam ociosos diminuam de 80% para 20% do tempo (M), orientando os desenvolvedores a construírem uma solução que utilize as melhores práticas (A) e otimize os recursos para se adequar à demanda dos usuários por uma experiência interativa e acessível a qualquer momento (R).*



### 1.2.2 Premissas e Restrições

Para este projeto, foram identificadas as seguintes **premissas**:

- Serão utilizadas cerca de 10 horas semanais para o desenvolvimento deste projeto aplicado.
- A arquitetura deve ser projetada para garantir alta disponibilidade, utilizando mecanismos de redundância e balanceamento de carga para minimizar o tempo de inatividade.
- O sistema precisa disponibilizar as mesmas informações, independentemente da plataforma utilizada pelo usuário.

Para este projeto, foram identificadas as seguintes **restrições**:

- A arquitetura deve ser desenvolvida durante o Projeto Aplicado, ou seja, cerca de 2 meses.
- A arquitetura deve cumprir todas as regulações e leis aplicáveis, como LGPD.
- Para implementação de serviços na nuvem, a arquitetura deve considerar serviços disponíveis na AWS devido ao conhecimento da equipe.
- Se houver a necessidade de refatoração ou criação de novos serviços, a arquitetura deve considerar tecnologias como .NET, NodeJS, React e SQL devido ao conhecimento da equipe.

A partir das premissas e restrições, foram identificados os seguintes **riscos**:

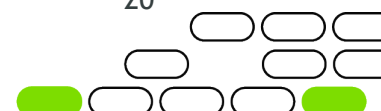
- R001: Migração de uma arquitetura monolítica para microsserviços pode causar interrupções no serviço, afetando a experiência do usuário e a confiança no sistema.
- R002: Inteligência artificial pode não gerar perguntas de qualidade suficiente, levando a uma insatisfação dos usuários e à falta de engajamento com o sistema.
- R003: Problemas técnicos nas salas virtuais podem impedir a interação entre os usuários, levando à frustração e desistência do uso do sistema.



- R004: Migração para a nuvem pode resultar em custos operacionais mais altos do que o previsto, especialmente se a utilização dos recursos não for otimizada.

Risco	Ações Preventivas	Ações Corretivas
R001	Realizar um planejamento detalhado da migração, incluindo a criação de um ambiente de teste; adotar uma abordagem incremental para a migração, movendo um serviço de cada vez.	Se houver problemas durante a migração, reverter para a versão monolítica temporariamente e analisar os erros para ajustar o plano de migração.
R002	Realizar testes rigorosos e validações da IA com um conjunto de dados representativo antes de sua implementação; envolver especialistas no desenvolvimento da IA.	Se a IA não atender às expectativas, ajustar os algoritmos e treinar novamente o modelo com mais dados ou feedbacks dos usuários.
R003	Realizar testes de carga e usabilidade antes do lançamento; garantir que a infraestrutura de backend suporte o tráfego esperado.	Implementar soluções de fallback e suporte técnico em tempo real durante o uso das salas virtuais para resolver problemas rapidamente.
R004	Realizar uma análise detalhada de custos antes da migração; implementar monitoramento e alertas para utilização de recursos em tempo real.	Se os custos aumentarem, revisar e otimizar a utilização de recursos, como escalonamento automático e desligamento de instâncias ociosas.

Risco	Probabilidade	Impacto	Prioridade
R001	Média	Alto	Alta
R002	Média	Alto	Alta
R003	Alta	Médio	Alta
R004	Média	Alto	Alta



### 1.2.3 Backlog de Produto

A imagem a seguir ilustra o backlog do produto, elaborado na ferramenta Trello, com base nas análises realizadas durante as etapas anteriores descritas neste documento. Nesta imagem, as atividades do backlog já estão divididas nas sprints:

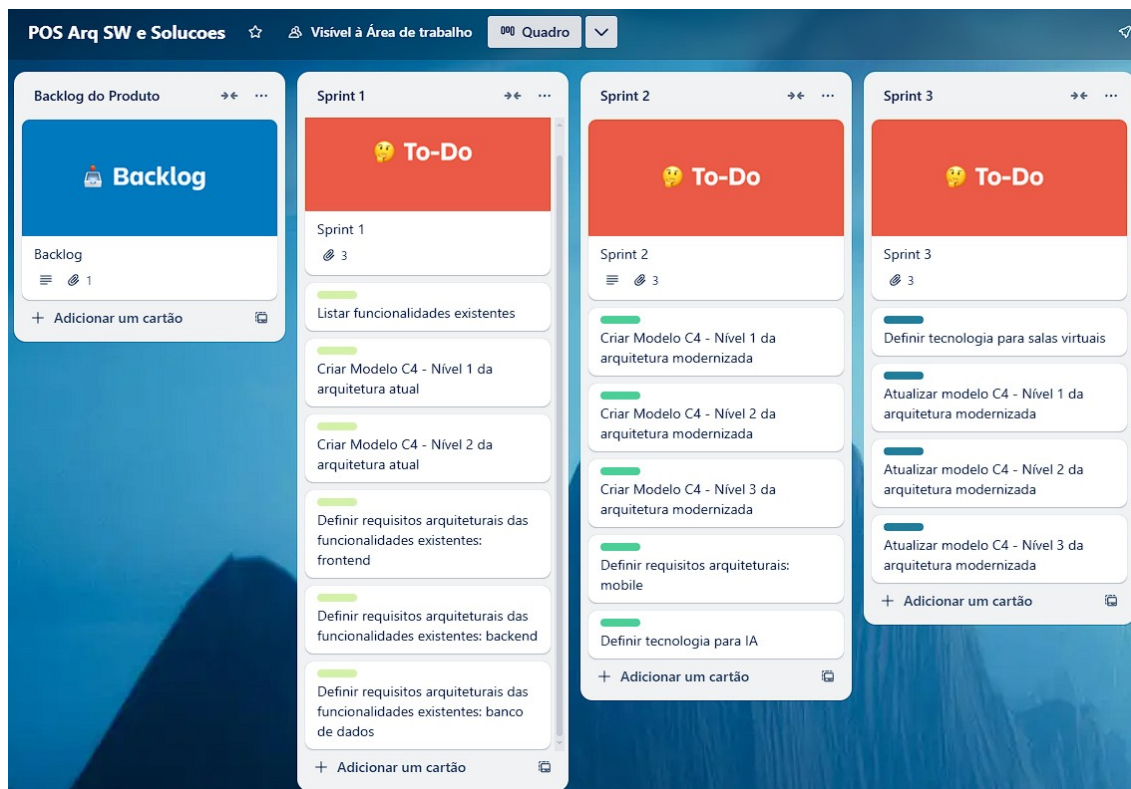


Figura 13: Backlog do Produto do Projeto Aplicado - via Trello

Link para acesso do backlog do produto na ferramenta Trello

<https://trello.com/invite/b/672ac7efb295a3de703e81f5/ATTI6515368af72c24ef384e7ce27da7c6845BFC600A/pos-arq-sw-e-solucoes>

## 2. Área de Experimentação

Nesta seção, serão apresentadas evidências do planejamento dos requisitos selecionados do Backlog de Produto a cada sprint, além de demonstrar como eles foram desenvolvidos e registrar os resultados alcançados.

### 2.1 Sprint 1

Os principais objetivos da Sprint 1 foram listar as funcionalidades atuais, criar diagramas C4 até o nível 2 do sistema atual para entender melhor seu funcionamento e definir os requisitos arquiteturais para essas funcionalidades já existentes.

#### 2.1.1 Solução

- Evidência do planejamento:

Na figura a seguir, encontra-se o backlog do projeto aplicado. Na coluna referente à sprint 1, estão os itens planejados, e a sprint está em andamento:

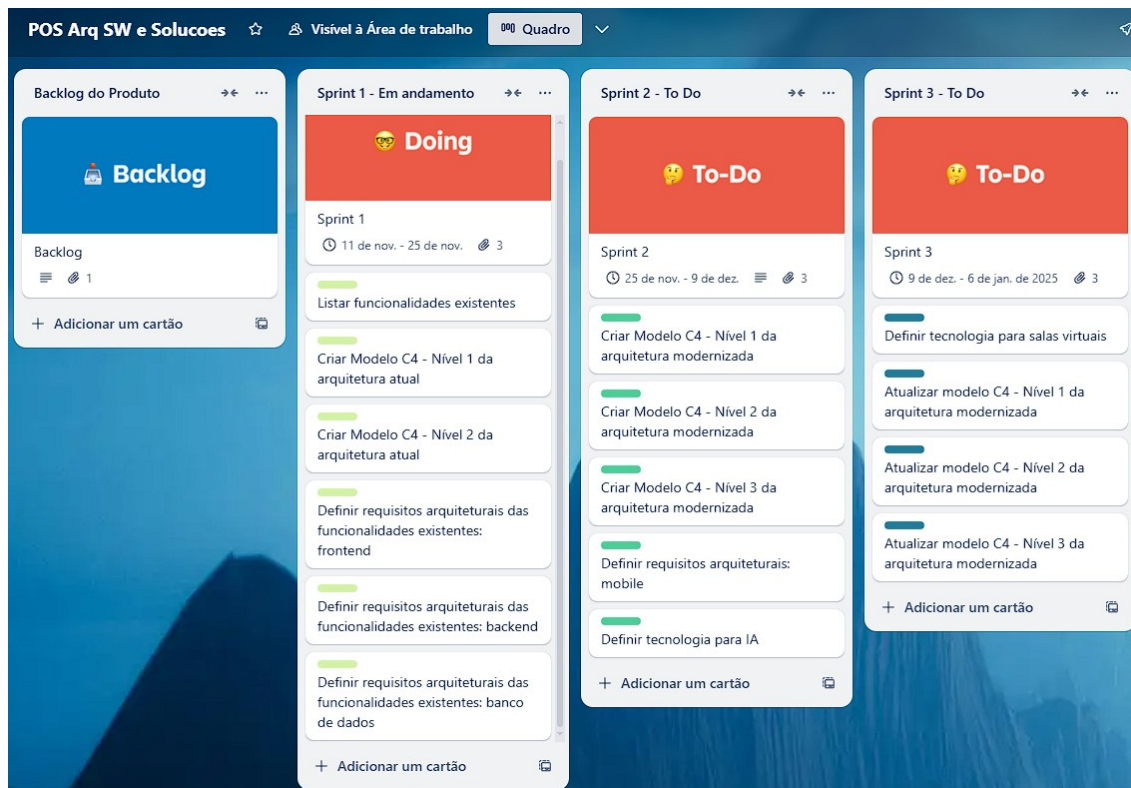


Figura 14: Evidência de planejamento da sprint 1 - via Trello

● Evidência da execução de cada requisito:

**Requisito 1: Listar funcionalidades existentes**

A imagem a seguir apresenta a evidência da reunião realizada para a listagem das funcionalidades existentes:

Funcionalidades existentes

segunda-feira, 11 de novembro de 2024 20:10

Reunião realizada em: 2024-11-11

**Participantes (Junes From Inaba Corp):**

- Área de negócio responsável pelo "Sistema de Simulados Colaborativos";
- Desenvolvedores;
- Arquiteto de soluções e software;

**Funcionalidades do "Sistema de Simulados Colaborativos"**

- Manutenção de usuário
  - Cadastrar
  - Visualizar
  - Alterar
  - Desativar
- Manutenção de questões
  - Cadastrar
  - Visualizar
    - Lista de questões
    - Detalhes de uma questão
  - Alterar
  - Desativar
- Manutenção de simulados
  - Gerar (cadastrar)
    - Vincular questões ao simulado
  - Visualizar
    - Lista de simulados (somente do próprio usuário)
    - Resultado de um simulado (somente do próprio usuário)
- Preenchimento de simulados
  - Responder questões vinculadas ao simulado
  - Encerrar simulado
  - Contar tempo decorrido
- Manutenção de comentários (feedback sobre as perguntas cadastradas)
  - Cadastrar
  - Visualizar
  - Editar
  - Excluir
- Análise de desempenho
  - Exibir gráfico de linha com o progresso nos simulados
  - Exibir gráfico de radar do estudante com base nos categorias das perguntas respondidas para cada tipo de simulado

*Figura 15: Lista de funcionalidades do sistema - obtidas via reunião*

**Requisito 2: Criar Modelo C4 - Nível 1 da arquitetura atual**

A imagem a seguir apresenta a evidência da criação do Modelo C4 nível 1 da arquitetura atual, criado através da ferramenta Draw.IO (<https://app.diagrams.net/>):





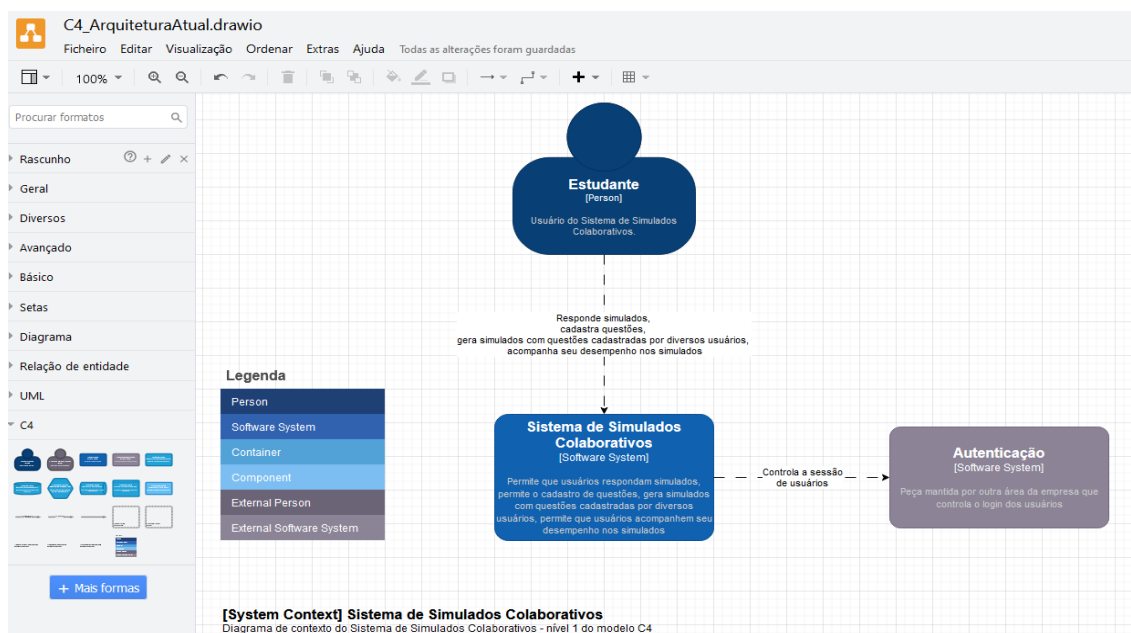


Figura 16: Diagrama nível 1 do modelo C4 - via Draw.IO

### Requisito 3: Criar Modelo C4 - Nível 2 da arquitetura atual

A imagem a seguir apresenta a evidência da criação do Modelo C4 nível 2 da arquitetura atual, criado através da ferramenta Draw.IO (<https://app.diagrams.net/>):

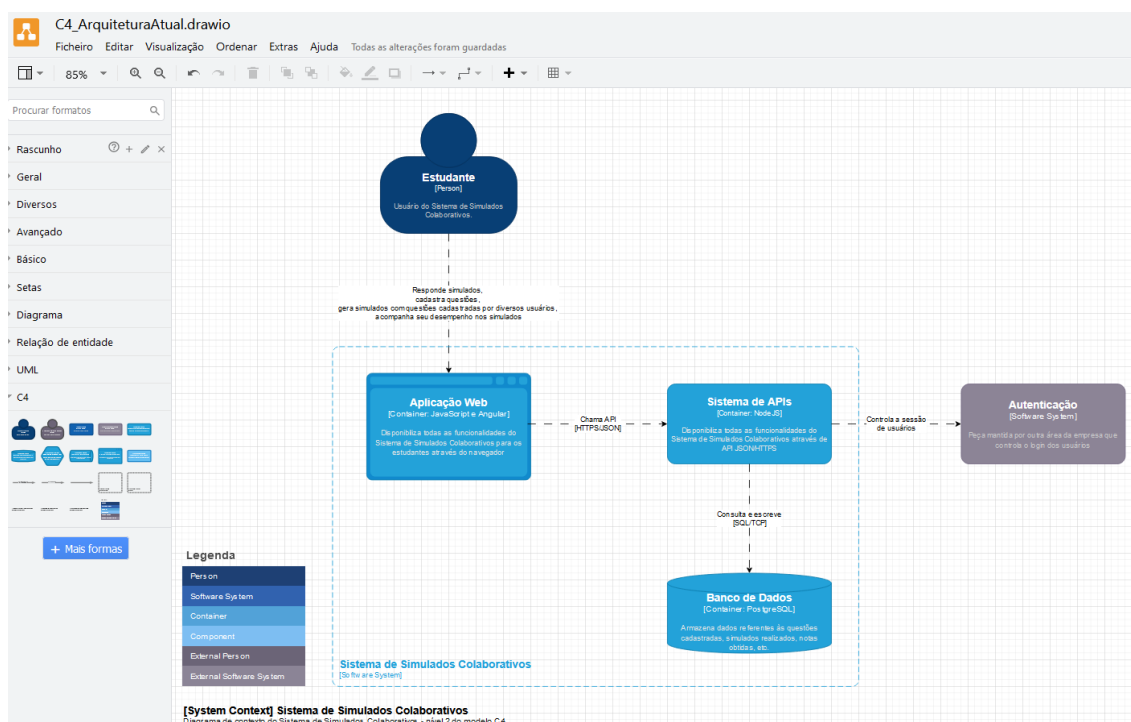


Figura 17: Diagrama nível 2 do modelo C4 - via Draw.IO

**Requisito 4: Definir requisitos arquiteturais das funcionalidades existentes: frontend**

Para definir os requisitos arquiteturais do frontend, serão analisados os serviços da AWS para processamento computacional, bem como frameworks destinados a melhorar a usabilidade e a responsividade do sistema. Para realizar essa análise, o quadro abaixo será preenchido na seção “Evidência dos resultados”:

Frontend	Tecnologias analisados	Serviço escolhido	Observações
Recurso computacional			
Usabilidade (Framework)			

**Requisito 5: Definir requisitos arquiteturais das funcionalidades existentes: backend**

Para definir os requisitos arquiteturais do backend para as funcionalidades existentes, serão listados os serviços analisados para cada funcionalidade, e será informado o serviço escolhido, além das observações que contribuirão para essa escolha. Para realizar essa análise, o quadro abaixo será preenchido na seção “Evidência dos resultados”

Funcionalidade	Serviços analisados	Serviço escolhido	Observações
Cadastrar usuário			
Visualizar usuário			
Alterar usuário			
Desativar usuário			
Cadastrar questão			
Visualizar lista de questões			
Visualizar detalhes de uma questão			
Alterar questão			
Desativar questão			



Cadastrar comentário			
Visualizar comentários de uma questão			
Editar comentário			
Excluir comentário			
Gerar simulado			
Visualizar lista de simulados (do usuário)			
Visualizar resultado de um simulado (do usuário)			
Responder questões vinculadas ao simulado			
Encerrar simulado			
Contar tempo decorrido de um simulado			
Gráfico de linha com o progresso (do usuário) nos simulados			
Gráfico de radar com nas categorias de perguntas respondidas pelo usuário) para cada tipo de simulado			

#### **Requisito 6: Definir requisitos arquiteturais das funcionalidades existentes: banco de dados**

Para definir os requisitos arquiteturais de armazenamento, foram analisados serviços da AWS para banco de dados e para armazenamento de arquivos. Para realizar essa análise, o quadro abaixo será preenchido na seção de “Evidência dos resultados”:

Armazenamento	Serviços analisados	Serviço escolhido	Observações
Banco de dados			
Armazenamento de arquivos			

● **Evidência dos resultados:**

**Requisito 1: Listar funcionalidades existentes**

Como evidência dos resultados, serão apresentadas imagens de algumas funcionalidades do sistema.

A imagem abaixo mostra o formulário utilizado para a funcionalidade de manutenção de questões. Essa funcionalidade tem o objetivo de gravar uma questão, suas opções de resposta e o vínculo entre a questão e os tipos de provas no banco de dados, para que depois elas sejam utilizadas em simulados:

*Figura 18: Tela com formulário para cadastro de questão*

Na imagem abaixo, encontra-se o botão “Adicionar opção de resposta” e o componente que é exibido ao clicar nesse botão:



Opção de resposta

Digite uma opção de resposta:
 

0 / 1000

Opção de resposta correta:
 

---

Ordem em que a opção de resposta deve aparecer:

Justifique por que essa opção está correta / incorreta:
 

0 / 1000

Adicionar opção de resposta

Figura 19: Formulário para cadastro de questão - destaque para "Adicionar opção de resposta"

A imagem a seguir exibe a lista de questões cadastradas por um usuário:

Simulados Colaborativos

Simulados

Questões

Questões

Minhas questões

Todas as questões

Cadastro

**Questão:** Which AWS storage technology can be considered a "virtual hard disk in the cloud"

**Idioma:** Inglês

**Subtema:** Performance Efficiency

**Questão:** Which AWS service is used to enable multi-factor authentication?

**Idioma:** Inglês

**Subtema:** Security

Figura 20: Tela do sistema com a lista de questões cadastradas pelo usuário

A imagem abaixo mostra o formulário utilizado na funcionalidade de criação de simulados. O simulado é criado para um tipo de prova, de acordo com o idioma, a quantidade de questões e o tempo informado pelo usuário:

Simulados

Questões

Simulados

Meus simulados

Novo simulado

Criar novo simulado

Aplicador Prova:

AWS

Tipo de Prova:

AWS Certified Cloud Practitioner - Foundational

Idioma:

Inglês

Quantidade de perguntas:

5

Tempo para o simulado:

01:40

Limpar

Salvar

Figura 21: Tela com formulário para cadastro de simulado com exemplo de preenchimento

A imagem a seguir mostra a lista de simulados criados por um usuário, de acordo com os parâmetros informados na tela anterior, e apresenta o status de cada um deles:

Simulados Colaborativos			
Simulados			
<div>Meus simulados</div> <div>Novo simulado</div>			
Nome da prova: AWS Certified Cloud Practitioner - Foundational	Idioma: Inglês	Tempo selecionado: 01:50:00	Percentual necessário para aprovação: 75.000
Data de criação do simulado: 05/10/2022	Tempo utilizado: 0	Percentual de acerto: SIMULADO EM ANDAMENTO	
Quantidade questões: 6			
Nome da prova: AWS Certified Cloud Practitioner - Foundational	Idioma: Inglês	Tempo selecionado: 01:50:00	Percentual necessário para aprovação: 75.000
Data de criação do simulado: 05/10/2022	Tempo utilizado: 0	Percentual de acerto: SIMULADO EM ANDAMENTO	
Quantidade questões: 6			

Figura 22: Tela do sistema com a lista de simulados cadastrados pelo usuário

A imagem a seguir mostra os detalhes de um simulado em andamento, incluindo uma das questões que faz parte dele:

**Simulados Colaborativos**

Simulados
Questões

**Simulados**

Meus simulados
Novo simulado

**Realizando simulado 22**

Nome da Prova: AWS Certified Cloud Practitioner - Foundational		Idioma: Inglês	Data de criação: 05/10/2022
Percentual de Acerto Necessário: 75.000%	Quantidade de Questões do Simulado: 6	Tempo Seleccionado: 01:50:00	Tempo Utilizado: 00:00:00

1. Which AWS service is used to enable multi-factor authentication?

☐ AWS IAM

☐ Amazon STS

☐ Amazon EC2

☐ AWS KMS

*Figura 23: Tela do simulado com as questões a serem respondidas*

A imagem a seguir mostra o resultado de um simulado preenchido e finalizado:

**Simulados**

Meus simulados
Novo simulado

**Resultado do Simulado 11**

Nome da Prova: AWS Certified Cloud Practitioner - Foundational		Idioma: Inglês	Data de criação: 01/10/2022
Percentual de Acerto Necessário: 75.000%	Tempo Seleccionado: 01:00:00	Quantidade de Questões do Simulado: 2	
Percentual de Acerto Atingido: 50%	Tempo Disponível: 00:00:00	Quantidade de Questões Certas: 1	

1. When a customer deploys a database on Amazon RDS, what is the customer responsible for?

☒ Controlling network access through security groups

☐ Patching the underlying operating system

☐ Managing automatic backups of the database

☐ Configuring Auto Scaling for high availability of instances

**EXPLICAÇÃO (Resposta correta!)**

**Correta** "Controlling network access through security groups" The Amazon Relational Database Service (RDS) is a managed service where AWS manage administration tasks including hardware provisioning, database setup, patching and backups. The customer is responsible for configuring security groups to control access to the database.

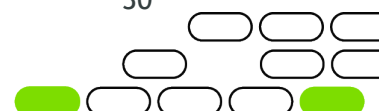
**Incorreta** "Patching the underlying operating system" This is taken care of by AWS.

**Incorreta** "Managing automatic backups of the database" This is taken care of by AWS.

**Incorreta** "Configuring Auto Scaling for high availability of instances" You cannot use Auto Scaling with RDS for availability of the database instances.

*Figura 24: Tela com o resultado de um simulado finalizado*

No menu “Progresso”, ao selecionar o tipo de prova, será exibido um gráfico com as notas obtidas em cada um dos simulados realizados pelo usuário ao longo do tempo, considerando apenas o tipo de prova selecionado. Além disso, será exibido um gráfico com o percentual médio de acertos nas categorias/subtemas das perguntas respondidas nos simulados desse tipo de prova, conforme a imagem a seguir:



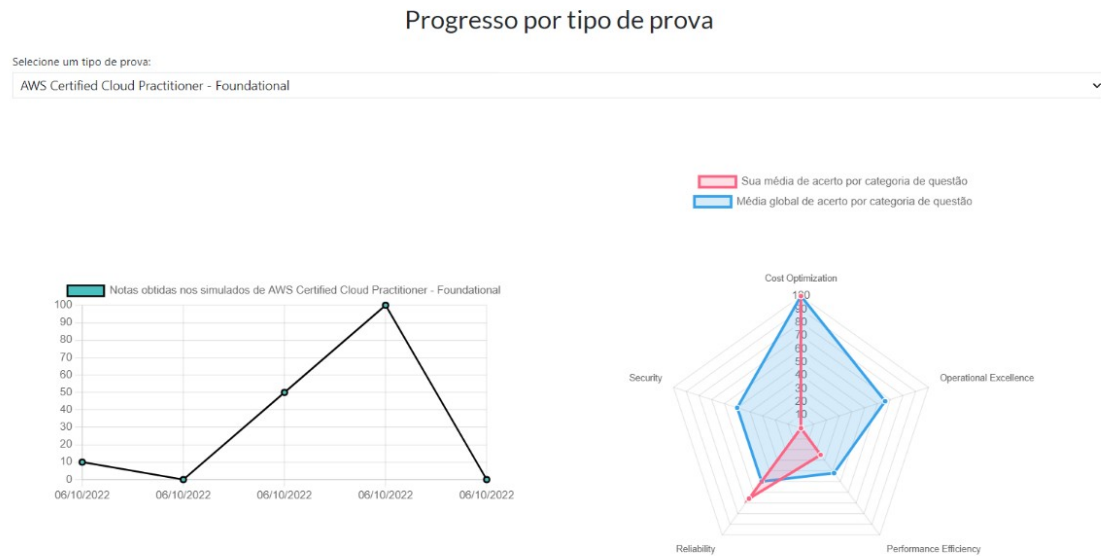


Figura 25: Tela com o progresso do usuário por tipo de prova

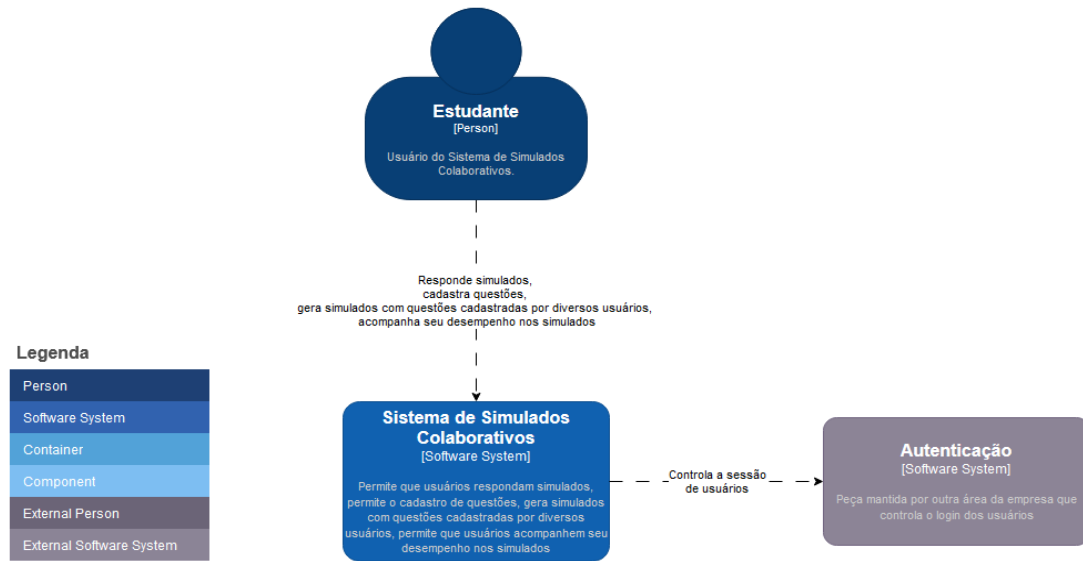
As imagens apresentadas durante a evidência deste item representam as principais funcionalidades do sistema atual.

## Requisito 2: Criar Modelo C4 - Nível 1 da arquitetura atual

No nível 1 do modelo C4, a ideia é mostrar as interações de forma macro, sem muitos detalhes, dando ênfase às comunicações e dependências entre sistemas e usuários que compõem e interagem com o software.

Para construir o nível 1 do modelo C4 para o sistema atual, foi utilizado um exemplo de arquitetura monolítica disponibilizado na documentação oficial do modelo C4: <https://c4model.com/abstractions/microservices#stage-1---monolithic-architectural-style>





**[System Context] Sistema de Simulados Colaborativos**  
Diagrama de contexto do Sistema de Simulados Colaborativos - nível 1 do modelo C4

Figura 26: Diagrama do sistema utilizando nível 1 do modelo C4

### Requisito 3: Criar Modelo C4 - Nível 2 da arquitetura atual

O modelo C4 no nível 2, também conhecido como "Diagrama de Contêineres", é uma representação que detalha a arquitetura de um sistema ao dividir suas funcionalidades em contêineres, que podem ser aplicações, serviços, bancos de dados ou qualquer outro tipo de unidade que execute uma parte do sistema.

Para construir o nível 2 do modelo C4 para o sistema atual, foi utilizado um exemplo de arquitetura monolítica disponibilizado na documentação oficial do modelo C4: <https://c4model.com/abstractions/microservices#stage-1---monolithic-architectural-style>

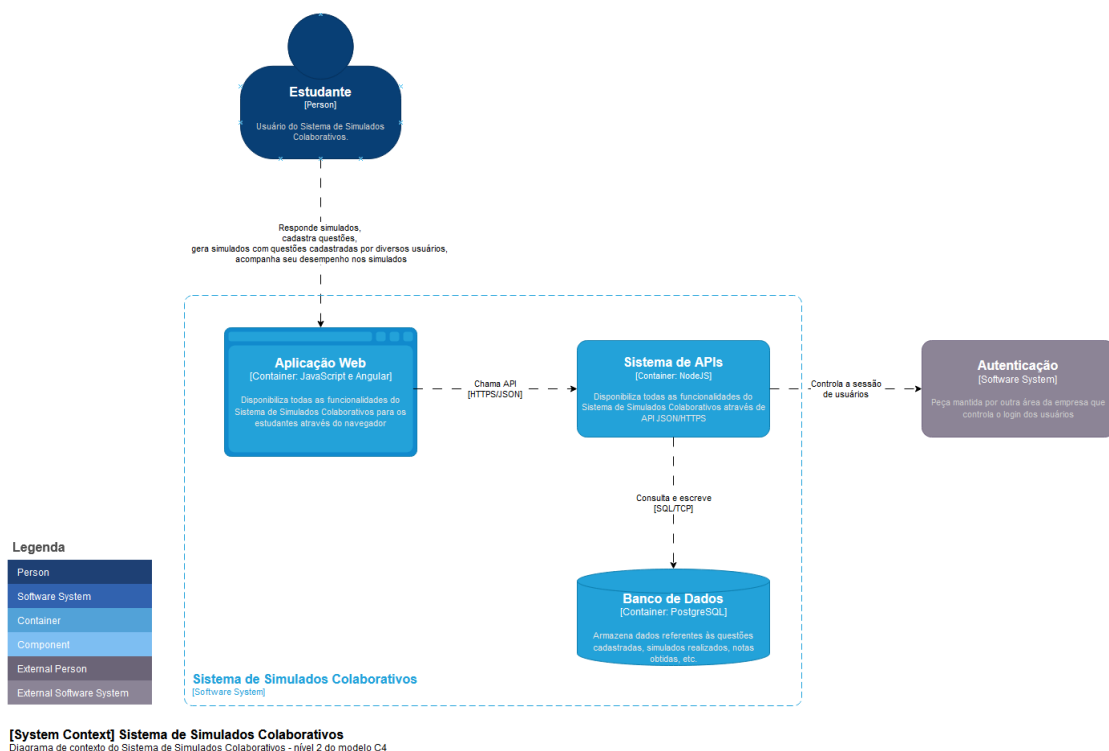


Figura 27: Diagrama do sistema utilizando nível 2 do modelo C4

#### Requisito 4: Definir requisitos arquiteturais das funcionalidades existentes: frontend

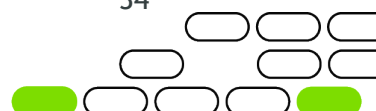
Segue o quadro apresentado na seção “Evidência da execução”, preenchido com os serviços analisados para processamento computacional e frameworks destinados a melhorar a usabilidade e a responsividade do sistema, com o objetivo de definir os requisitos arquiteturais do frontend:

Frontend	Tecnologias analisados	Serviço escolhido	Observações
Recurso computacional	ECS Fargate, EC2, EKS	ECS Fargate	Por enquanto, o frontend não será dividido em micro frontends. Este serviço foi escolhido devido à facilidade de gerenciar sua escalabilidade.
Usabilidade (Framework)	Bootstrap, Tailwind CSS, Foundation	Bootstrap	Este serviço foi escolhido devido ao conhecimento e à familiaridade da equipe.

### **Requisito 5: Definir requisitos arquiteturais das funcionalidades existentes: backend**

Segue o quadro apresentado na seção 'Evidência da execução', preenchido com os serviços analisados e o serviço escolhido para cada uma das funcionalidades existentes, além das observações que contribuíram para essa escolha, a fim de definir os requisitos arquiteturais do backend.

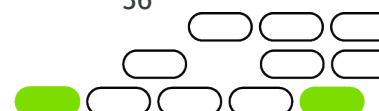
Funcionalidade	Serviços analisados	Serviço escolhido	Observações
<b>Cadastrar usuário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Visualizar usuário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Alterar usuário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Desativar usuário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Cadastrar questão</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Visualizar lista de questões</b>	ECS Fargate, Lambda, EC2, EKS	ECS Fargate	Essa funcionalidade é executada com muita frequência, seja pelos usuários ou para responder/visualizar simulados. Por isso, este serviço foi escolhido.



<b>Visualizar detalhes de uma questão</b>	ECS Fargate, Lambda, EC2, EKS	ECS Fargate	Essa funcionalidade é executada com muita frequência, seja pelos usuários ou para responder/visualizar simulados. Por isso, este serviço foi escolhido.
<b>Alterar questão</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Desativar questão</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Cadastrar comentário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Visualizar comentários de uma questão</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Editar comentário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Excluir comentário</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Gerar simulado</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi



			escolhido.
<b>Visualizar lista de simulados (do usuário)</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Visualizar resultado de um simulado (do usuário)</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Responder questões vinculadas ao simulado</b>	ECS Fargate, Lambda, EC2, EKS	ECS Fargate	Essa funcionalidade é executada com muita frequência e exige uma conexão constante entre o usuário e o servidor. Por isso, este serviço foi escolhido.
<b>Encerrar simulado</b>	ECS Fargate, Lambda, EC2, EKS	ECS Fargate	Essa funcionalidade exige uma conexão constante entre o usuário e o servidor. Por isso, este serviço foi escolhido.
<b>Contar tempo decorrido de um simulado</b>	ECS Fargate, Lambda, EC2, EKS	ECS Fargate	Essa funcionalidade é executada com muita frequência e exige uma conexão constante entre o usuário e o servidor. Por isso, este serviço foi escolhido.
<b>Gráfico de linha com o progresso (do usuário) nos simulados</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.
<b>Gráfico de radar com nas categorias de perguntas respondidas (pelo usuário) para cada tipo de simulado</b>	ECS Fargate, Lambda, EC2, EKS	Lambda	Essa funcionalidade não excede 1000 chamadas simultâneas e não é executada muitas vezes ao longo do dia; sua execução é rápida. Por isso, este serviço foi escolhido.



## **Requisito 6: Definir requisitos arquiteturais das funcionalidades existentes: banco de dados**

Segue o quadro apresentado na seção “Evidência da execução”, preenchido com os serviços analisados para banco de dados e para armazenamento de arquivos, com o objetivo de definir os requisitos arquiteturais de armazenamento:

Armazenamento	Serviços analisados	Serviço escolhido	Observações
Banco de dados	RDS PostgreSQL, Aurora PostgreSQL	Aurora PostgreSQL	Além da compatibilidade do banco de dados original com o serviço escolhido, ele oferece melhor performance, maior disponibilidade e menor custo entre os serviços analisados.
Armazenamento de arquivos	S3, EFS, EBS	S3	O serviço escolhido oferece tudo o que o sistema necessita, com o melhor preço em comparação aos demais.

### 2.1.2 Lições Aprendidas

Durante a Sprint 1, diversas experiências foram vivenciadas e muitas lições foram aprendidas. A seguir, serão destacadas as experiências mais marcantes:

- O levantamento das funcionalidades e o entendimento da arquitetura atual do sistema, por meio dos diagramas do modelo C4 de níveis 1 e 2, foram fundamentais para definir os requisitos arquiteturais da migração que será realizada para nuvem.
- No início, foi realizado um teste com a ferramenta Mermaid Chart para gerar os diagramas do modelo C4 através de scripts, mas seu resultado ficou confuso. Portanto a ferramenta Draw.IO foi utilizada, e os diagramas entregues ficaram mais claros. Provavelmente, essa será a ferramenta utilizada nas demais etapas do projeto.
- Foi necessário um aprofundamento no conhecimento do modelo C4 para criar os diagramas da arquitetura atual do sistema, e boa parte do conhecimento adquirido foi por meio do conteúdo em texto e vídeo disponibilizado no site oficial do modelo <https://c4model.com/>. Esse conhecimento também será muito utilizado nas demais etapas do projeto.



## 2.2 Sprint 2

Os principais objetivos da Sprint 2 foram criar diagramas C4 até o nível 3 das funcionalidades atuais, considerando a nova arquitetura na nuvem; definir requisitos arquiteturais para a versão mobile; e definir uma tecnologia de inteligência artificial.

### 2.2.1 Solução

- **Evidência do planejamento:**

Na figura a seguir, encontra-se o backlog do projeto aplicado. Na coluna referente à sprint 2, estão os itens planejados, e a sprint está em andamento:

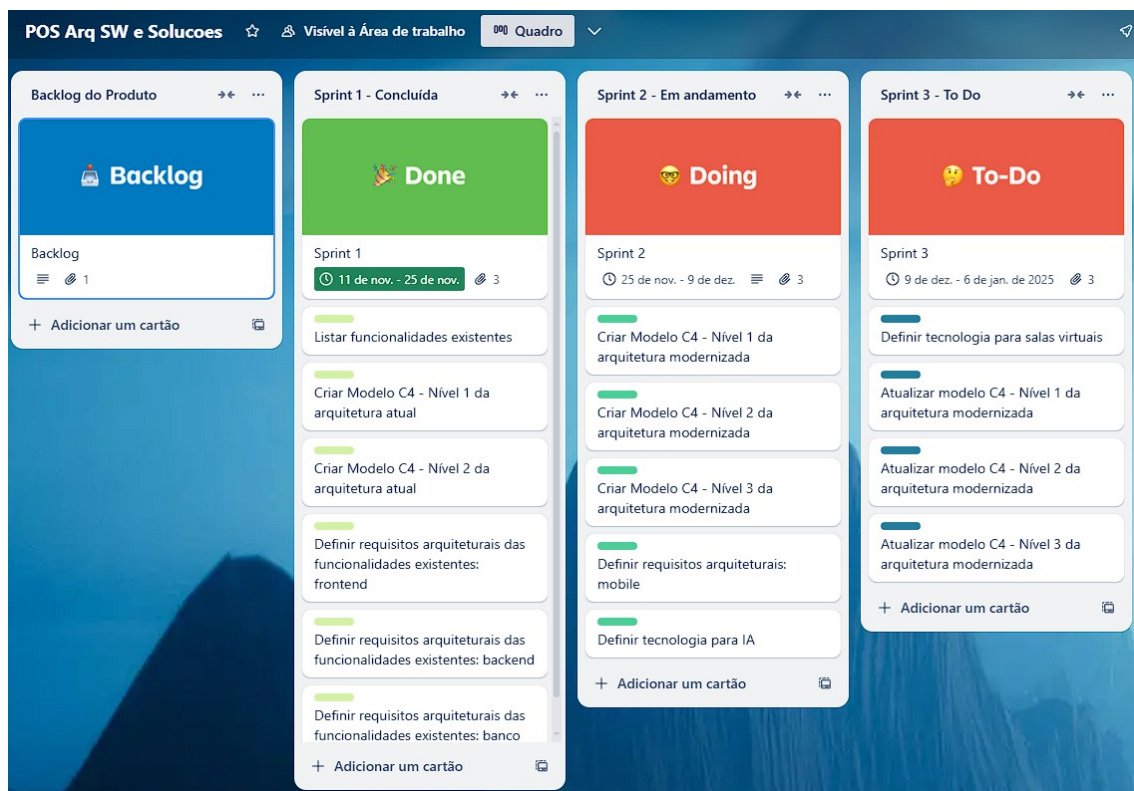


Figura 28: Evidência de planejamento da sprint 2 - via Trello

- **Evidência da execução de cada requisito:**

#### **Requisito 1: Criar Modelo C4 - Nível 1 da arquitetura modernizada**

A imagem a seguir apresenta a evidência da criação do Modelo C4 nível 1 da arquitetura modernizada, criado através da ferramenta Draw.IO (<https://app.diagrams.net/>):

### **Requisito 2: Criar Modelo C4 - Nível 2 da arquitetura modernizada**

A imagem a seguir apresenta a evidência da criação do Modelo C4 nível 2 da arquitetura modernizada, criado através da ferramenta Draw.IO (<https://app.diagrams.net/>):

### **Requisito 3: Criar Modelo C4 - Nível 3 da arquitetura modernizada**

A imagem a seguir apresenta a evidência da criação do Modelo C4 nível 3 da arquitetura modernizada, criado através da ferramenta Draw.IO (<https://app.diagrams.net/>):

### **Requisito 4: Definir requisitos arquiteturais mobile**

Para definir os requisitos arquiteturais da versão mobile, destinados a melhorar a usabilidade do sistema, serão analisados os modelos de arquitetura mobile e plataformas disponíveis. Para realizar essa análise, o quadro abaixo será preenchido na seção “Evidência dos resultados”:

Modelo Arquitetura	Descrição	Prós	Contras
Nativo			
Híbrido			
Progressive Web App (PWA)			
Cross-Platform			

### **Requisito 5: Definir tecnologia para IA**

Para definir a tecnologia utilizada em inteligência artificial, serão listados os serviços analisados, assim como os prós e contras que contribuiram para a escolha. Para realizar essa análise, o quadro abaixo será preenchido na seção “Evidência dos resultados”:

Modelo Arquitetura	Descrição	Prós	Contras
Nativo			
Híbrido			



- **Evidência dos resultados:**

**Requisito 1: Criar Modelo C4 - Nível 1 da arquitetura modernizada**

No nível 1 do modelo C4, a ideia é mostrar as interações de forma macro, sem muitos detalhes, dando ênfase às comunicações e dependências entre sistemas e usuários que compõem e interagem com o software.

A imagem a seguir detalha o modelo C4 no nível 1 para a versão modernizada do sistema:

**Requisito 2: Criar Modelo C4 - Nível 2 da arquitetura modernizada**

O modelo C4 no nível 2, também conhecido como "Diagrama de Contêineres", é uma representação que detalha a arquitetura de um sistema ao dividir suas funcionalidades em contêineres, que podem ser aplicações, serviços, bancos de dados ou qualquer outro tipo de unidade que execute uma parte do sistema.

A imagem a seguir detalha o modelo C4 no nível 2 para a versão modernizada do sistema:

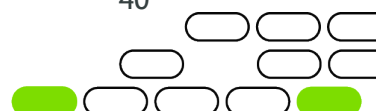
**Requisito 3: Criar Modelo C4 - Nível 3 da arquitetura modernizada**

O modelo C4 no nível 3, refere-se à representação de componentes e suas interações dentro de um sistema, oferecendo uma visão detalhada da estrutura interna. Neste nível o objetivo é proporcionar uma compreensão clara da arquitetura do sistema, facilitando a identificação de responsabilidades, fluxos de dados e a interação entre diferentes partes do software, o que é essencial para o desenvolvimento, manutenção e evolução do sistema.

A imagem a seguir detalha o modelo C4 no nível 3 para a versão modernizada do sistema:

**Requisito 4: Definir requisitos arquiteturais mobile**

Segue o quadro apresentado na seção "Evidência da execução", preenchido com os modelos de arquitetura mobile e plataformas disponíveis analisados, com o objetivo de definir os requisitos arquiteturais da versão mobile:



Modelo Arquitetura	Descrição	Prós	Contras
Nativo	Aplicativos desenvolvidos para plataformas específicas (iOS, Android).	<ul style="list-style-type: none"> <li>- Melhor desempenho</li> <li>- Acesso total a APIs nativas</li> <li>- Interface mais alinhada com guidelines da plataforma</li> </ul>	<ul style="list-style-type: none"> <li>- Desenvolvimento separado para cada plataforma</li> <li>- Maior custo e tempo de desenvolvimento</li> </ul>
Híbrido	Combina elementos de aplicativos nativos e web. Usualmente utiliza frameworks como Ionic ou Cordova.	<ul style="list-style-type: none"> <li>- Código compartilhado entre plataformas</li> <li>- Menor custo de desenvolvimento</li> <li>- Tempo de entrega mais rápido</li> </ul>	<ul style="list-style-type: none"> <li>- Desempenho inferior ao nativo</li> <li>- Dependência de plugins para recursos nativos</li> </ul>
Progressive Web App (PWA)	Aplicativos web que oferecem uma experiência similar à de um aplicativo nativo.	<ul style="list-style-type: none"> <li>- Não requer instalação</li> <li>- Funciona em qualquer dispositivo com um navegador</li> <li>- Atualizações automáticas</li> </ul>	<ul style="list-style-type: none"> <li>- Acesso limitado a APIs nativas</li> <li>- Desempenho pode variar dependendo do navegador</li> </ul>
Cross-Platform	Utiliza frameworks como React Native ou Flutter para desenvolver um único código que roda em várias plataformas.	<ul style="list-style-type: none"> <li>- Código compartilhado</li> <li>- Desempenho próximo ao nativo (especialmente com Flutter)</li> <li>- Tempo de desenvolvimento reduzido</li> </ul>	<ul style="list-style-type: none"> <li>- Curva de aprendizado para frameworks</li> <li>- Limitações em algumas APIs nativas</li> </ul>

Após realizar o levantamento acima, o modelo de arquitetura escolhido foi o Cross-Platform com a tecnologia React Native.

Essa escolha foi feita pois os desenvolvedores que já possuem experiência com ReactJS podem aproveitar seus conhecimentos ao trabalhar com React Native de diversas maneiras:

1. **Conceitos Básicos:** Ambos utilizam os mesmos conceitos fundamentais de componentes, estado (state) e propriedades (props), o que facilita a transição.
2. **JSX:** O uso de JSX é comum em ambos, permitindo que os desenvolvedores escrevam a estrutura de seus componentes de forma semelhante.
3. **Gerenciamento de Estado:** Ferramentas como Redux e Context API funcionam da mesma forma em ambos os ambientes, permitindo que os desenvolvedores reutilizem suas habilidades.
4. **Ciclo de Vida dos Componentes:** Os métodos de ciclo de vida dos componentes são semelhantes, o que facilita a adaptação.



5. **Lógica de Negócio:** A lógica de negócios pode ser compartilhada entre o frontend web e a versão mobile, já que ambos utilizam JavaScript.

Portanto, a transição de ReactJS para React Native pode ser bastante suave, e os desenvolvedores podem aproveitar sua experiência anterior para desenvolver aplicações móveis de forma mais eficiente.

### **Requisito 5: Definir tecnologia para IA**

Segue o quadro apresentado na seção “Evidência da execução”, preenchido com as tecnologias de inteligência artificial:

Tecnologia	Descrição	Prós	Contras
<b>OpenAI GPT-3 ou GPT-4</b>	Modelos de linguagem que geram texto coerente e contextualizado com base em prompts fornecidos.	<ul style="list-style-type: none"> <li>- Geração de texto de alta qualidade.</li> <li>- Flexibilidade em diversos temas.</li> </ul>	<ul style="list-style-type: none"> <li>- Custo pode ser elevado dependendo do uso.</li> <li>- Respostas podem ser inconsistentes.</li> </ul>
<b>Google Cloud Natural Language</b>	API que analisa texto e pode ser usada para gerar perguntas e respostas a partir de dados.	<ul style="list-style-type: none"> <li>- Integração fácil com outros serviços do Google.</li> <li>- Boa análise semântica.</li> </ul>	<ul style="list-style-type: none"> <li>- Pode não gerar perguntas diretamente.</li> <li>- Requer mais configuração para personalização.</li> </ul>
<b>IBM Watson Assistant</b>	Plataforma de IA que permite criar assistentes virtuais e chatbots, podendo gerar perguntas.	<ul style="list-style-type: none"> <li>- Boa capacidade de entendimento de linguagem natural.</li> <li>- Integração com outros serviços da IBM.</li> </ul>	<ul style="list-style-type: none"> <li>- Curva de aprendizado para configuração.</li> <li>- Pode ser complicado para iniciantes.</li> </ul>
<b>Microsoft Azure Cognitive Services</b>	Conjunto de APIs que inclui geração de texto e análise de linguagem.	<ul style="list-style-type: none"> <li>- Integração com outros serviços da Microsoft.</li> <li>- Boa documentação e suporte.</li> </ul>	<ul style="list-style-type: none"> <li>- Custos podem aumentar rapidamente com o uso.</li> <li>- Pode ser complexo para implementar.</li> </ul>
<b>Hugging Face Transformers</b>	Biblioteca de modelos de linguagem pré-treinados que podem ser ajustados para tarefas específicas.	<ul style="list-style-type: none"> <li>- Acesso a uma variedade de modelos.</li> <li>- Comunidade ativa e muitos recursos.</li> </ul>	<ul style="list-style-type: none"> <li>- Requer conhecimento técnico para implementação.</li> <li>- Necessidade de infraestrutura própria.</li> </ul>



Analisando as opções mencionadas acima, considerando a familiaridade dos desenvolvedores alinhada aos prós e contras, a tecnologia escolhida para implementação de inteligência artificial foi OpenAI GPT-3 ou GPT-4.

A imagem abaixo apresenta um trecho de código com a implementação de como é possível utilizar a API da OpenAI para gerar texto a partir de um prompt em Node.js:

```
1  const axios = require('axios');
2
3  const API_URL = 'https://api.openai.com/v1/chat/completions';
4  const API_KEY = 'YOUR_OPENAI_API_KEY'; // Substitua pelo token da OpenAI
5
6  async function gerarTexto(prompt) {
7    try {
8      const response = await axios.post(API_URL, {
9        model: "gpt-3.5-turbo", // ou "gpt-4" se tiver acesso
10       messages: [{ role: "user", content: prompt }],
11       max_tokens: 150 // Ajuste conforme necessário
12     }, {
13       headers: {
14         'Authorization': `Bearer ${API_KEY}`,
15         'Content-Type': 'application/json'
16       }
17     });
18
19     return response.data.choices[0].message.content;
20   } catch (error) {
21     console.error('Erro ao gerar texto:', error);
22     return null;
23   }
24 }
25
26 // Exemplo de uso
27 const prompt = "Você é um especialista em AWS e está criando simulados para a prova de Cloud Practitioner. Gere uma pergunta";
28 gerarTexto(prompt).then(resposta => {
29   console.log('Resposta GTP:', resposta);
30 });
31
```

### 2.2.2 Lições Aprendidas

Durante a Sprint 2, diversas experiências foram vivenciadas e muitas lições foram aprendidas. A seguir, serão destacadas as experiências mais marcantes:

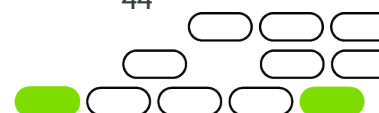


## 2.3 Sprint 3

### 2.3.1 Solução

- Evidência do planejamento:
- Evidência da execução de cada requisito:
- Evidência dos resultados:

### 2.3.2 Lições Aprendidas



## 3. Considerações Finais

### 3.1 Resultados

Por meio de um texto detalhado, apresente os principais resultados alcançados pelo seu Projeto Aplicado.

Cite os pontos positivos e negativos, as dificuldades enfrentadas e as experiências vivenciadas durante todo o processo.

### 3.2 Contribuições

Apresente quais foram as contribuições que o seu Projeto Aplicado trouxe para que o Desafio proposto fosse solucionado.

Cite, por exemplo, as inovações, as vantagens sobre os similares, as melhorias alcançadas, entre outros.

### 3.3 Próximos passos

Descreva quais são os próximos passos que poderão contribuir com o aprimoramento da solução apresentada pelo seu Projeto Aplicado.

Disponibilizar um tutor online baseado em inteligência artificial

Diagrama de arquitetura de aplicações

