

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
PUC Minas Virtual
Pós-graduação *Lato Sensu* em Engenharia de *Software*

Projeto Integrado

Relatório Técnico

TasksMe

Felipe Alves da Silva
Lucas Nilton Santos Garuti

Belo Horizonte
Março 2024

Projeto Integrado

Sumário

Projeto Integrado	2
1. Cronograma de Trabalho	3
2. Introdução	4
3. Definição Conceitual da Solução	4
3.1 Diagrama de Casos de Uso	6
3.2 Requisitos Funcionais	7
3.3 Requisitos Não-funcionais	7
4. Protótipo Navegável do Sistema	7
5. Diagrama de Classes de Domínio	7
6. Arquitetura da Solução	8
6.1 Padrão Arquitetural	8
6.2 C4 model - Diagrama de Contexto	10
7. Frameworks de Trabalho	11
8. Estrutura Base do Front End	113
9. Modelo Relacional ou Projeto de Banco de Dados NoSQL	15
10. Plano de Testes	166
11. Relatório de Execução de Testes de Software	166
12. Apropriação de Horas no Projeto	166
13. Código da Aplicação	17
14. Avaliação Retrospectiva	17
14.1 Objetivos Estimados	18
14.2 Objetivos Alcançados	18
14.3 Lições aprendidas	19
15. Referências	19

1. Cronograma de Trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
02 / 01 / 24	04 / 01 / 24	1. Construção dos objetivos do projeto, apresentação do problema e descrição geral do Software.	Relatório técnico e descritivo
05 / 01 / 24	07 / 01 / 24	2. Levantamento dos casos de uso e desenvolvimento do diagrama de casos de uso	Diagrama de casos de uso
08 / 01 / 24	10 / 01 / 24	3. Levantamento dos requisitos funcionais e não funcionais do projeto	Apresentação dos requisitos funcionais e não funcionais do sistema
11 / 01 / 24	28 / 01 / 24	4. Desenvolvimento do protótipo navegável e interativo da aplicação	Protótipo de interface navegável
29 / 01 / 24	30 / 01 / 24	5. Desenvolvimento do diagrama de classes de domínio	Diagrama de classes de domínio
31 / 01 / 24	31 / 01 / 24	6. Definição do padrão arquitetural	Padrão arquitetural da aplicação
01 / 02 / 24	02 / 02 / 24	7. Desenvolvimento do diagrama de contexto	Diagrama de contexto C4 model
03 / 02 / 24	03 / 02 / 24	8. Definição das frameworks de trabalho	Apresentação dos frameworks definidos
04 / 02 / 24	06 / 02 / 24	9. Estrutura do front-end base	Apresentação do layout das principais telas da aplicação
07 / 02 / 24	09 / 02 / 24	10. Desenvolvimento do modelo relacional do projeto	Modelo relacional do projeto
10 / 02 / 24	10 / 02 / 24	11. Elaboração do plano de testes	Plano de testes da aplicação
11 / 02 / 24	25 / 02 / 24	12. Desenvolvimento do código back-end da aplicação	Código fonte do back-end da aplicação
26 / 03 / 24	14 / 03 / 24	13. Desenvolvimento do código front-end da aplicação	Código fonte do front-end da aplicação
15 / 03 / 24	16 / 03 / 24	14. Gravação do vídeo	Vídeo de apresentação do protótipo navegável e casos de uso
17 / 03 / 24	17 / 03 / 24	15. Avaliação retrospectiva	Levantamento do aprendizado adquirido durante o projeto
18 / 03 / 24	18 / 03 / 24	16. Objetivos estimados	Descrição dos objetivos estimados
18 / 03 / 24	18 / 03 / 24	17. Objetivos alcançados	Descrição dos objetivos alcançados
19 / 03 / 24	19 / 03 / 24	18. Retrospectiva e lições aprendidas	Descrição das lições aprendidas

2. Introdução

Com o avanço da tecnologia nos últimos anos, passamos a ter acesso ao mundo em nossas mãos com enorme facilidade, fator que nos exige a capacidade de fazer mais coisas do que antes fazíamos durante nosso dia. Seja em casa e/ou no trabalho, nosso dia a dia é vivido à base de execução de atividades e tarefas, sendo suas conclusões importantes para o bom funcionamento das nossas rotinas.

Em contrapartida, se o nosso número de atividades diárias aumentar, o nosso tempo diminui, o que torna ainda mais precioso. Qualquer esquecimento e minuto desperdiçado pode ocasionar numa tarefa planejada não concluída no fim do dia nos fazendo perder mais tempo e deixando de cumprir com nossas obrigações. Isso nos remete a boa e velha “listinha” de tarefas, com seu propósito cada vez mais funcional para auxiliar no gerenciamento e cumprimento de nossas atividades. Porém, um dos problemas nos dias atuais, é que nossas atividades não cabem mais em um post it ou folha de papel a ser pregada na geladeira de casa ou na mesa do trabalho. A quantidade de papel e espaço gasto seria imenso e principalmente sua visibilidade nos faria voltar ao problema precursor para a solução: falta de gerenciamento e controle das atividades a serem executadas.

Se por um lado a tecnologia trouxe alguns “problemas” nesse sentido, ela também possibilita os solucionar. Baseado neste contexto, o presente trabalho visa por meio de uma lista de tarefas virtual suprir essa necessidade, oferecendo de maneira simples e prática o gerenciamento da execução de atividades de forma eficaz. Através da aplicação dos conceitos da Engenharia de Software é esperado uma aplicação acessível, confiável e de fácil manutenção, onde os usuários serão beneficiados pela simples usabilidade da aplicação.

O **objetivo geral** deste trabalho é apresentar o desenvolvimento do projeto de uma aplicação para gerenciamento de tarefas através de uma “To do List” voltada para usuários inseridos em pequenas empresas e organizações ou até mesmo famílias que queiram organizar e gerenciar tarefas domiciliares. Serão aplicadas práticas e conceitos aprendidos durante as disciplinas do curso.

Nosso “to do list”, chamado de TasksMe tem o propósito de elencar, organizar, acompanhar e controlar as tarefas listadas, auxiliando o usuário e facilitando assim a execução das mesmas.

No presente trabalho serão abordadas várias etapas processuais da engenharia de software, como: análises de requisitos, testes do sistema e sua implementação. Além disso, aspectos de usabilidade, escalabilidade e segurança de dados serão levadas em consideração para se atingir uma solução simples e robusta.

A aplicação será desenvolvida para web, onde cada usuário terá sua própria interface de acordo com suas tarefas criadas. Uma vez criada, o usuário poderá fazer edições nas tarefas, comentários e deletá-las. Ele também terá a opção de compartilhamento das tarefas (via link) com outros usuários, que neste primeiro momento só poderão acompanhá-las, adicionar comentários e deletá-los. Essas funcionalidades poderão ser estendidas para app mobile, além de possibilitar a criação de novas funcionalidades.

Os objetivos específicos são:

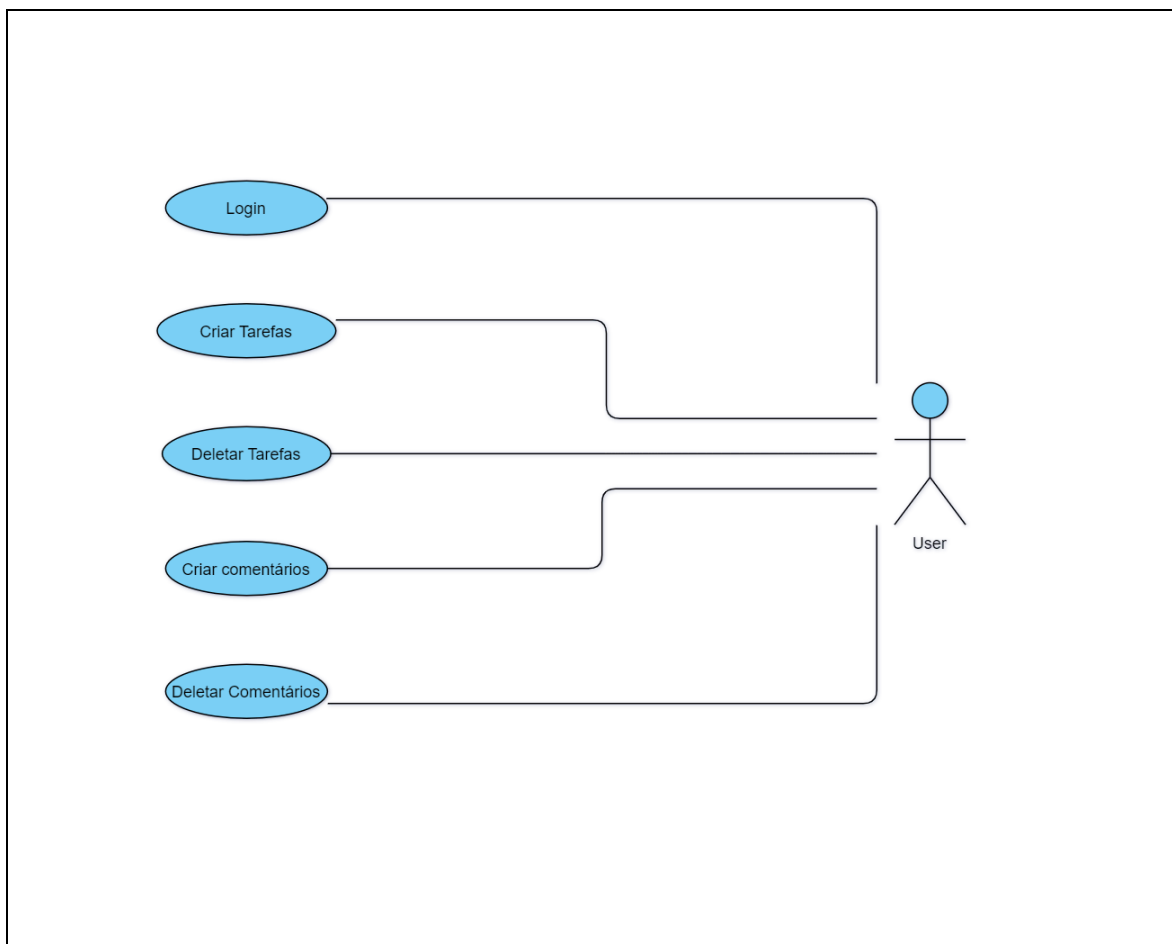
- Definição da Arquitetura: serão realizadas pesquisas e estudos detalhados das arquiteturas disponíveis para melhor atendimento das demandas da aplicação, visando identificar a que melhor entrega nos quesitos de escala, segurança e desempenho;
- Desenvolvimento de Software: definida a arquitetura, seguiremos com o desenvolvimento do software onde criaremos uma simples e eficiente aplicação, banco de dados e interface intuitiva com usuário. Para isso, serão utilizadas ferramentas tecnológicas das mais usuais e atuais disponíveis;
- Disponibilização em Cloud: a aplicação desenvolvida será disponibilizada em uma plataforma de armazenamento cloud (na nuvem) permitindo o acesso de forma remota e flexível dos usuários via internet em variados dispositivos com acesso e de qualquer lugar.

Os objetivos específicos apresentados guiarão o desenvolvimento do projeto de forma eficaz ao propósito final de implementar nossa aplicação na nuvem para acesso dos usuários que desejam ter uma lista de tarefas virtual.

3. Definição Conceitual da Solução

Como dito a aplicação será simples e funcional, onde o usuário logado, proprietário de uma tarefa, terá acesso para gerenciar e compartilhar a mesma com outros usuários; usuários não proprietários da tarefa logados, terão acesso a essa tarefa via link compartilhado, que possibilitará o acompanhamento, realização de comentários e exclusão dos seus comentários; usuários não proprietários da tarefa e não logados, só poderão acompanhar a tarefa compartilhada.

3.1 Diagrama de Casos de Uso



3.2 *Requisitos Funcionais*

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O usuário deve se auto cadastrar no sistema.	B	A
RF02	O usuário deve ser capaz de criar uma tarefa	B	A
RF03	O usuário deve ser capaz de excluir uma tarefa criada	M	A
RF04	O usuário deve ser capaz de visualizar tarefas de outras pessoas se ela estiver pública	B	M
RF05	O usuário deve ser capaz de compartilhar o link de sua tarefa pública	B	M
RF06	O usuário deve ser capaz de comentar em uma tarefa pública, desde que esteja logado.	B	M
RF07	O usuário deve ser capaz de criar tarefas	M	A

* B = Baixa, M = Média, A = Alta.

3.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve apresentar tempo de resposta abaixo de 200 ms no processamento de 95% das operações de consulta.	A
RNF02	O sistema web deve ser responsivo de forma a proporcionar a utilização de qualquer uma de suas funcionalidades em qualquer resolução	A
RNF03	O sistema deve estar disponível em qualquer período em regime 24/7	A
RNF04	O sistema deve ser hospedado em cloud provider para melhor disponibilidade	B
RNF05	O sistema deve cumprir as exigências de LGPD, dando possibilidade de exclusão dos dados do usuário que solicitar	A

4. *Protótipo Navegável do Sistema*

O protótipo navegável de nosso sistema nada mais é que o front-end da aplicação em wireframes, o que a deixa num formato mais real.

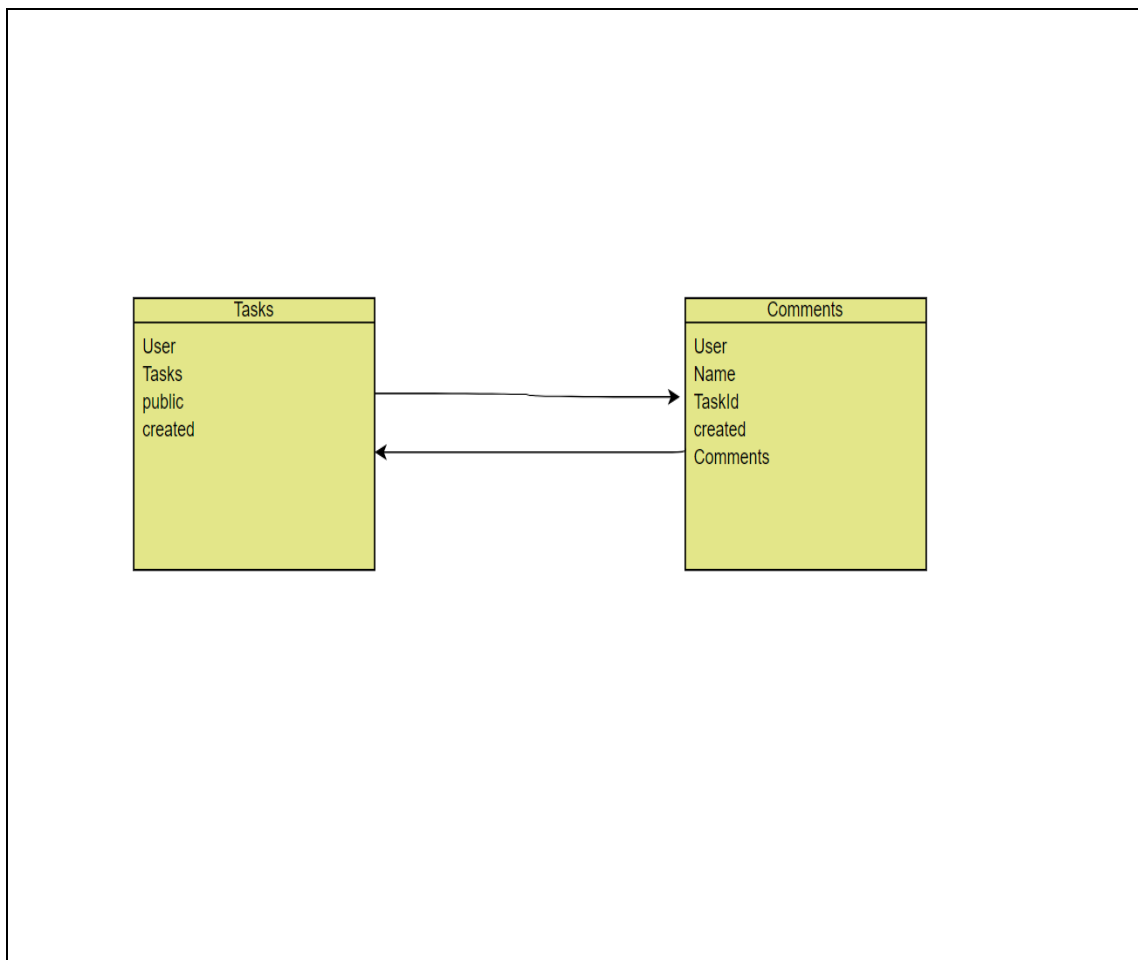
Para demonstrar o funcionamento de nosso projeto, disponibilizamos dois links para visualização das telas interativas do sistema e os casos de uso:

Link do protótipo navegável: <https://tasks-me-zras.vercel.app/>

Link do vídeo de apresentação dos casos de uso:

<https://www.loom.com/share/08b008ecc27845beba5ffdc81ff7edd3?sid=cde2c426-1e66-4138-ab8b-0d7703624994>

5. Diagrama de Classes de Domínio



6. Arquitetura da Solução

6.1 Padrão Arquitetural

O padrão de arquitetura adotado no projeto foi o cliente-servidor (SSR). Esse padrão arquitetural foi escolhido visando garantir eficiência, escalabilidade e uma experiência de usuário consistente. A utilização do Next.js, um framework baseado em React, oferece uma abordagem robusta para implementar essa arquitetura, permitindo o desenvolvimento de aplicações web modernas e responsivas.

Na arquitetura Server Side Rendering, o sistema é dividido em duas partes principais: o cliente, que é a interface de usuário executada no navegador do usuário, e o servidor, que gerencia a lógica de negócios, o processamento de dados e a interação com o banco de dados.

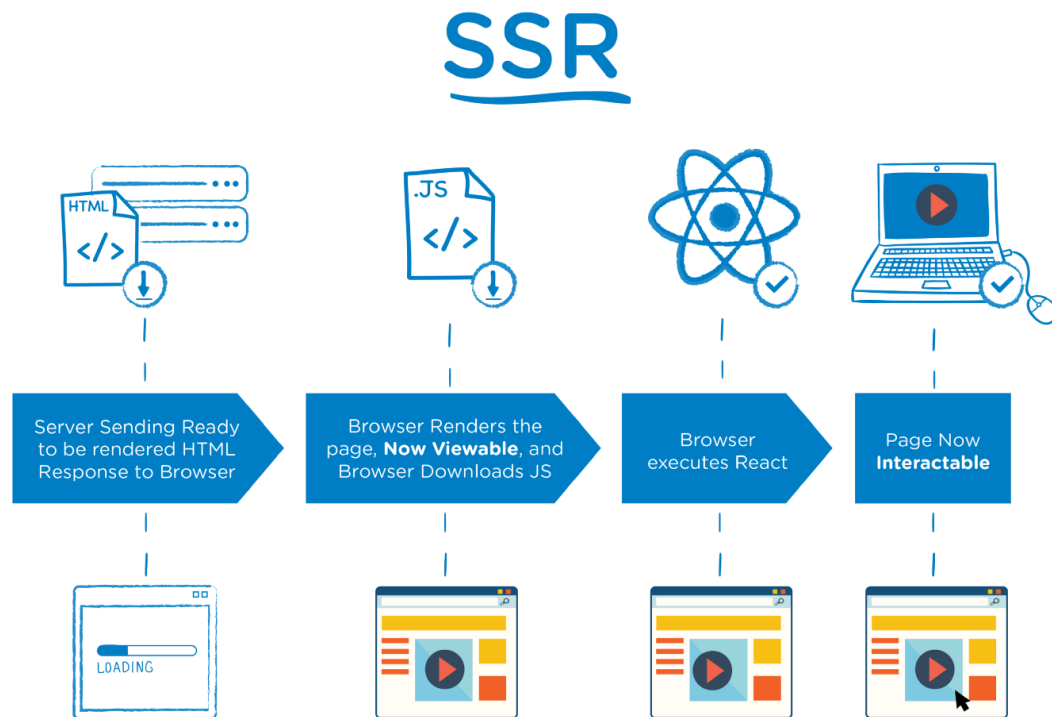
Com o Next.js, a construção do lado do servidor (SSR) é um dos recursos principais. Ele permite que as páginas sejam renderizadas no servidor antes de serem enviadas para o cliente, resultando em tempos de carregamento mais rápidos e melhorando a otimização para motores de busca. Isso significa que parte da lógica de apresentação pode ser realizada no servidor, proporcionando uma experiência mais eficiente para o usuário.

Além do SSR, o Next.js também suporta a pré-renderização estática, gerando páginas HTML no momento da construção. Essa abordagem é particularmente útil para conteúdos estáticos, proporcionando uma entrega mais rápida de páginas e reduzindo a carga no servidor.

A arquitetura Server Side Rendering em uma aplicação Next.js também facilita a implementação de APIs. O framework oferece o conceito de "API Routes", permitindo que os desenvolvedores criem endpoints específicos para manipulação de dados, facilitando a comunicação entre o cliente e o servidor.

Outro benefício importante é o suporte à CSS-in-JS, permitindo que estilos sejam encapsulados dentro dos componentes, melhorando a modularidade e a manutenção do código. O Next.js também suporta code-splitting, dividindo o código da aplicação em partes menores que são carregadas sob demanda, melhorando o desempenho geral.

A utilização de TypeScript com Next.js proporciona uma camada adicional de segurança e clareza no desenvolvimento, facilitando a identificação de erros e melhorando a manutenção do código. Em resumo, a aplicação da arquitetura Server Side Rendering em uma aplicação web utilizando Next.js oferece vantagens significativas em termos de desempenho, escalabilidade e facilidade de desenvolvimento, permitindo a construção de aplicações web modernas e eficientes.



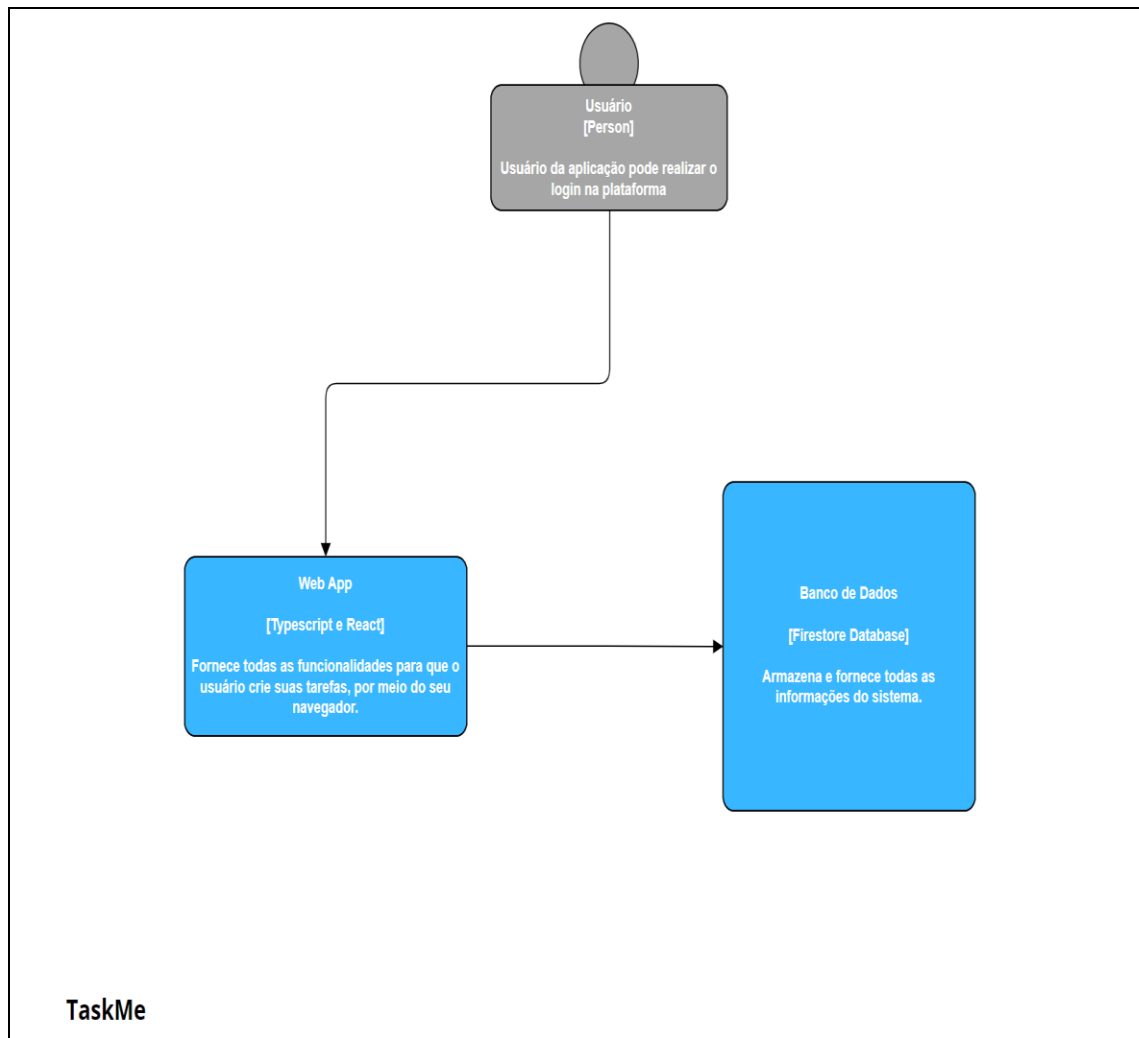
6.2 C4 model - Diagrama de Contexto

A seguir, apresentamos um diagrama de contexto simples que ilustra a estrutura geral da aplicação:

Temos o usuário que poderá criar suas tarefas, sendo elas públicas ou privadas, para as públicas poderá ser copiado o link e compartilhando com outras pessoas o usuário também, poderá comentar em outras tarefas que não sejam de sua autoria, mas que sejam públicas.

A interface serve como a parte de interação, proporcionando uma maneira amigável e intuitiva para que os usuários se engajem com a aplicação.

Uma parte crucial da aplicação Next.js é a comunicação que temos com o banco de dados do Firebase que nos ajuda no armazenamento das informações geradas pelo usuário, seja ela uma tarefa pública ou privada ou até mesmo comentários realizados em tarefas públicas, não criadas por pelo usuário.



7. *Frameworks de Trabalho*

Para o frontend do sistema, foi utilizado React com TypeScript e Next.js.

O React é uma biblioteca JavaScript de código aberto desenvolvida pelo Facebook. Ele é usado para criar interfaces de usuário (UI) interativas e reativas para aplicações web. O React permite que os desenvolvedores construam componentes reutilizáveis que representam partes específicas da interface do usuário, facilitando a criação de interfaces complexas e dinâmicas.

TypeScript é uma linguagem de programação desenvolvida pela Microsoft que estende o JavaScript adicionando recursos de tipagem estática. Ela é projetada para ser uma alternativa mais robusta e segura ao JavaScript padrão, especialmente para projetos de grande escala. O TypeScript compila para JavaScript, o que significa que o

código TypeScript é transformado em código JavaScript antes de ser executado nos navegadores ou ambientes Node.js.

O Next.js é um framework de código aberto para desenvolvimento web baseado em React. Ele foi criado pela equipe da Vercel (anteriormente conhecida como Zeit) e é projetado para facilitar a construção de aplicativos web modernos, oferecendo funcionalidades avançadas para aplicações React.

O Next.js fornece uma série de recursos poderosos que ajudam os desenvolvedores a criar aplicações web de maneira eficiente e escalável. Alguns dos principais recursos do Next.js incluem:

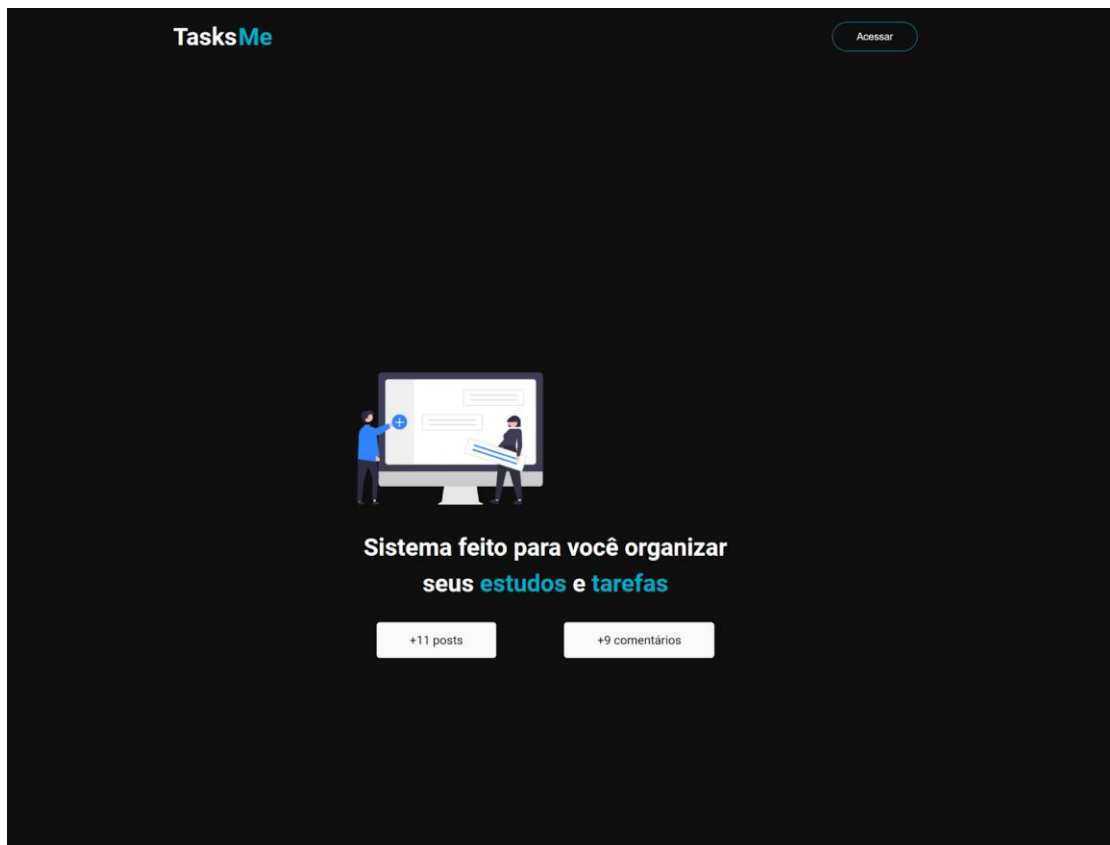
Renderização do lado do servidor e do lado do cliente: O Next.js oferece suporte à renderização tanto do lado do servidor (SSR) quanto do lado do cliente (CSR), permitindo que os desenvolvedores escolham a abordagem mais adequada para suas necessidades.

API Routes: Permite criar APIs facilmente, gerando rotas específicas para manipular solicitações de API.

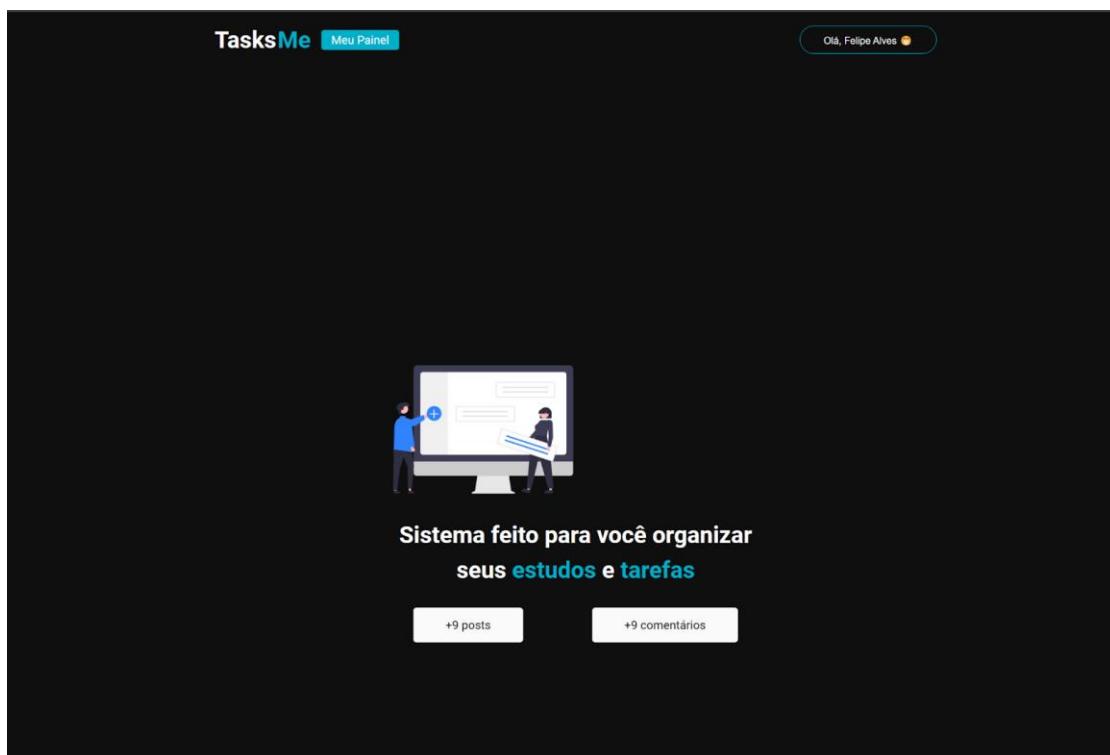
Suporte a TypeScript: Oferece suporte nativo ao TypeScript, o que possibilita o desenvolvimento de aplicativos com tipagem estática.

Para o backend e banco de dados, escolhemos o Firebase, uma plataforma desenvolvida pela Google que oferece uma variedade de serviços e ferramentas para o desenvolvimento de aplicativos móveis e web. O Firebase fornece funcionalidades prontas para uso, reduzindo a necessidade de gerenciar infraestrutura e permitindo que os desenvolvedores foquem mais na lógica do aplicativo. Isso inclui banco de dados em tempo real, autenticação de usuários e hospedagem de aplicativos estáticos.

8. *Estrutura Base do Front End*



Página inicial do site



Página quando o usuário está logado

TESTE 1

PUBLICO

🔍

TESTE

PUBLICO

🔍

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.What is Lorem Ipsum?

TasksMe

Meu Painel

Olá, Felipe Alves

Tarefa

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. What is Lorem Ipsum?

Deixar comentário

Digite seu comentário...

Enviar comentário

Todos comentários

Felipe Alves

TESTE

Devs Do Futuro

TESTE

TasksMe
Acessar

Tarefa

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. What is Lorem Ipsum?

Deixar comentário

Digite seu comentário...

Enviar comentário

Todos comentários

Felipe Alves

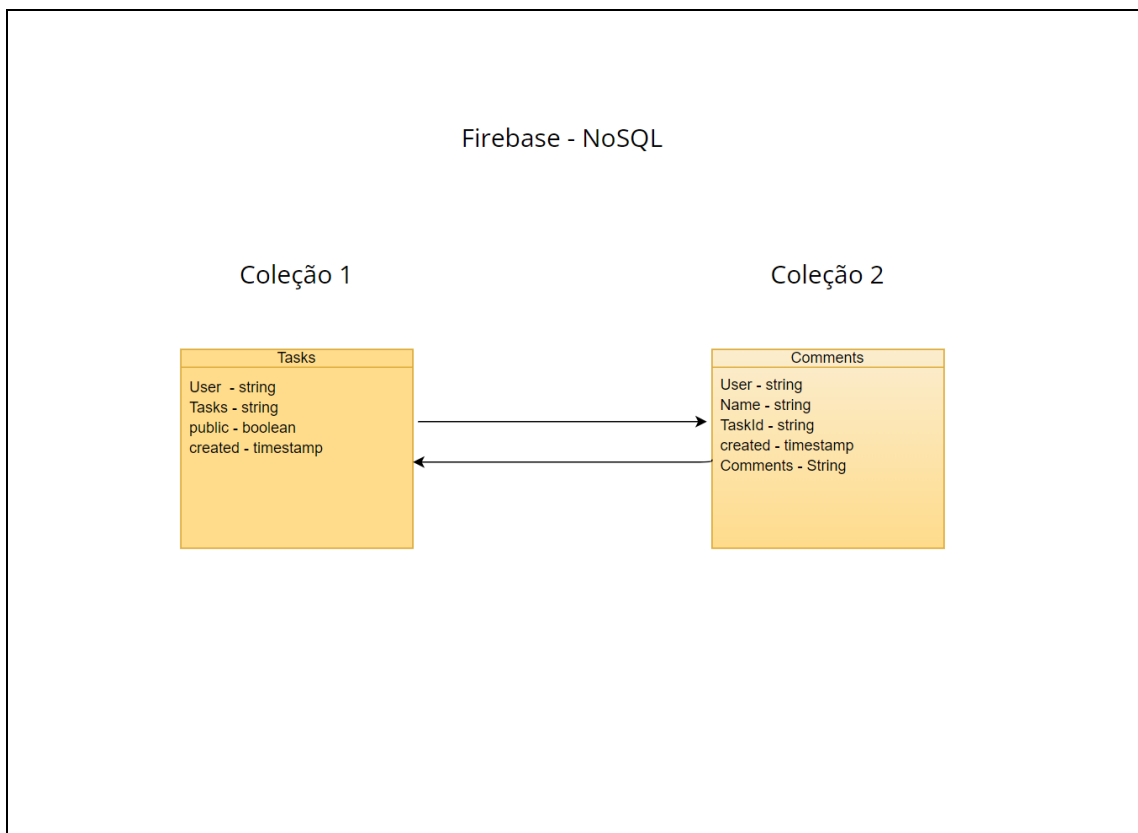
TESTE

Devs Do Futuro

TESTE

Página de quando o usuário acessa uma tarefa pública seja criada por ele ou por outra pessoa, mas não está logado

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL



10. Plano de Testes

Número	Caso de uso	Objetivo do caso de teste	Entradas	Resultados esperados
1	Criar Tarefas	Criar novas tarefas	Cria novas tarefas públicas ou privadas	Consegue criar tarefas
2	Criar comentários	Verificar se não é possível comentar não estando logado	Realizar o processo de comentar em uma tarefa pública sem está logado	Não conseguir comentar em uma tarefa pública, sem está logado.
3	Compartilhar tarefas	Conseguir compartilhar as tarefas públicas	Clicar no ícone de compartilhar e aparecer uma alerta dizendo que a tarefa foi copiada.	Mostra um alerta informando que o link da tarefa foi copiado.

11. Relatório de Execução de Testes de Software

Número	Caso de teste	Saída esperada	Resultados encontrados	Aprovado?
1	Usuário não conseguir comentar não estando logado	O botão de enviar comentário fica desativado	O botão de enviar comentários ficou azul claro e se a pessoa passar o mouse por cima de ele mostra um símbolo de bloqueado	SIM
2	Usuário conseguir logar com o login social (Google)	Ser enviado a tela do google para que possa escolher qual e-mail ele gostaria de logar na plataforma	E encaminhado para a tela de login do google.	SIM

12. Apropriação de Horas no Projeto

Histórico de apropriação de horas		
Data do registro	Atividade	Quantidade de horas
02 / 01 / 24	Contextualização e objetivos do projeto	5 horas
05 / 01 / 24	Definição conceitual – diagrama de casos de uso	4 horas
06 / 01 / 24	Definição conceitual – diagrama de casos de uso	4 horas
08 / 01 / 24	Definição conceitual – requisitos funcionais	3 horas
09 / 01 / 24	Definição conceitual – requisitos funcionais	4 horas
10 / 01 / 24	Definição conceitual – requisitos não funcionais	3 horas
11 / 01 / 24	Protótipo navegável do sistema	4 horas
12 / 01 / 24	Protótipo navegável do sistema	4 horas
13 / 01 / 24	Protótipo navegável do sistema	5 horas
15 / 01 / 24	Protótipo navegável do sistema	3 horas
17 / 01 / 24	Protótipo navegável do sistema	3 horas
20 / 01 / 24	Protótipo navegável do sistema	6 horas

22 / 01 / 24	Protótipo navegável do sistema	4 horas
23 / 01 / 24	Protótipo navegável do sistema	3 horas
26 / 01 / 24	Protótipo navegável do sistema	3 horas
27 / 01 / 24	Protótipo navegável do sistema	6 horas
28 / 01 / 24	Protótipo navegável do sistema	4 horas
29 / 01 / 24	Diagrama de classes de domínio	4 horas
30 / 01 / 24	Diagrama de classes de domínio	4 horas
31 / 01 / 24	Definição do padrão arquitetural	3 horas
01 / 02 / 24	Diagrama de contexto	5 horas
03 / 02 / 24	Escolha dos frameworks de trabalho	3 horas
04 / 02 / 24	Estrutura base do frontend	2 horas
07 / 02 / 24	Desenvolvimento do modelo relacional	3 horas
10 / 02 / 24	Elaboração do plano de testes	2 horas
11 / 02 / 24 a 25 / 03 / 24	Desenvolvimento do backend	50 horas
26 / 03 / 24 a 14 / 03 / 24	Desenvolvimento do frontend	64 horas
14 / 03 / 24	Código da aplicação	3 horas
15 / 03 / 24	Vídeo de apresentação do protótipo	3 horas
16 / 03 / 24	Vídeo de apresentação do protótipo	3 horas
17 / 02 / 24	Avaliação retrospectiva	4 horas

13.Código da Aplicação

Link do repositório: <https://github.com/felipealvessi/TasksMe>

Link da aplicação:

- TasksMe (Projeto Desenvolvido) - Link: • <https://tasks-me-zras.vercel.app/>
 - Credencial de acesso - Logar com a conta Google
- Tarefa Pública para Teste - Link: <https://tasks-me-zras.vercel.app/task/9VKxptCQtzbQIxEFsYWr>
- Link do vídeo: <https://www.loom.com/share/08b008ecc27845beba5ffdc81ff7edd3?sid=d6baf87d-01ce-4df6-9c1c-e964276d19df>

14.Avaliação Retrospectiva

O objetivo geral no início deste trabalho era apresentar o desenvolvimento do projeto de uma aplicação para gerenciamento de tarefas através de uma “To do List” voltada para usuários inseridos em pequenas empresas e organizações ou até mesmo famílias que queiram organizar e gerenciar tarefas domiciliares.

14.1 Objetivos Estimados

1. Definição da Arquitetura: realizar pesquisas e estudos detalhados das arquiteturas disponíveis que melhor atenderiam as demandas da aplicação, visando escala, segurança e desempenho;
2. Desenvolvimento de Software: desenvolver o software de uma simples e eficiente aplicação com banco de dados e interface intuitiva com usuário, utilizando as mais adequadas ferramentas e tecnologias.
3. Disponibilização em Cloud: disponibilizar a aplicação em uma plataforma de armazenamento cloud (na nuvem) permitindo o acesso, flexibilidade e escalabilidade.

14.2 Objetivos Alcançados

Objetivo 1 - Definição da Arquitetura:

- Objetivo Alcançado: Realizamos o estudo de arquitetura, definindo uma estrutura sólida para o melhor desenvolvimento da aplicação.

Objetivo 2 - Desenvolvimento de Software:

- Objetivo Alcançado: Desenvolvemos com êxito um software que no geral atendeu os requisitos estabelecidos, com um banco de dados robusto e interface com usuário simples.

Objetivo 3 - Disponibilização em Cloud:

- Objetivo Alcançado: Finalizamos a implantação da nossa aplicação disponibilizando a mesma em uma plataforma na nuvem, garantindo acesso, flexibilidade e escalabilidade.

14.3 Lições aprendidas

A avaliação retrospectiva nos mostra o alcance com sucesso dos objetivos estabelecidos, oferecendo uma aplicação simples e eficaz para o gerenciamento de tarefas. As muitas lições aprendidas no decorrer do projeto certamente agregaram experiências práticas e contribuirão para melhorar projetos de desenvolvimento de software futuramente. Este TCC foi uma oportunidade prática de grande valia para aplicarmos os conceitos e princípios da Engenharia de Software adquiridos durante o curso.

Retrospectiva (Lições Aprendidas)		
	Descrição da Lição	Classificação
1	Planejamento e Pesquisa: As etapas iniciais de planejamento e pesquisa foram essenciais para êxito do projeto. Realizar uma análise profunda da arquitetura permitiu tomadas de decisões assertivas.	Positiva
2	Escolha de Tecnologia: A seleção analítica das tecnologias utilizadas no desenvolvimento do projeto foram de suma importância. Escolher as tecnologias corretas, cooperou para a eficiência do sistema.	Positiva
3	Desafios de Desenvolvimento e Implantação: O desenvolvimento do Software e a implantação do sistema em uma plataforma na nuvem nos gerou desafios técnicos, porém nos rendeu resultados consideráveis. Os desafios enfrentados nos fizeram aprimorar conhecimentos oportunizando a aplicação dos mesmos em nosso sistema.	Positiva
4	Usabilidade Contínua: A usabilidade da aplicação é um ponto crítico. Durante desenvolvimento, notamos a importância de se realizar testes contínuos da interface do usuário.	Positiva
5	Adaptação Futura: A decisão de começar com uma solução simples, porém robusta, se mostrou estrategicamente eficaz. Possibilitando no futuro a criação de novas funcionalidades no sistema a partir da nossa aplicação desenvolvida.	Positiva

15. Referências

Não se aplica.