UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

BEATRIZ DE ALMEIDA FERNANDES DE LIMA

PILAS: UM SISTEMA WEB PARA CONTROLE DO CONSUMO DE GULOSEIMAS EM UMA EMPRESA DE TECNOLOGIA

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2022

BEATRIZ DE ALMEIDA FERNANDES DE LIMA

PILAS: UM SISTEMA WEB PARA CONTROLE DO CONSUMO DE GULOSEIMAS EM UMA EMPRESA DE TECNOLOGIA

Proposta de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso de Tecnologia em Sistemas para Internet - TSI - da Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Guarapuava, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Andres Jessé Porfirio

Coorientadores: Henrique Fernando de Oliveira Rodrigues e Maurício Barfknecht

GUARAPUAVA
2022

4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1 PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

1.1 TÍTULO

Pilas: um sistema web para controle do consumo de guloseimas em uma empresa de tecnologia

1.2 MODALIDADE DO TRABALHO

Desenvolvimento de Sistemas

1.3 ÁREA DO TRABALHO

Desenvolvimento de um sistema web.

1.4 RESUMO

Tendo em vista a importância da presença de uma forte cultura organizacional nas empresas, uma empresa de tecnologia implantou um projeto que proporciona aos seus colaboradores acesso a diversos itens alimentícios em troca de uma moeda fictícia, denominada Pila. Com o intuito de auxiliar esse projeto, o presente trabalho busca desenvolver um sistema web com versão em PWA[1] que gerencia e automatiza os processos envolvidos, como o cadastro de produtos para compra e as movimentações de aquisição dos mesmos.

[1] Progressive Web App: aplicação híbrida entre as páginas da web regulares e um aplicativo móvel.

2 DESCRIÇÃO DA PROPOSTA

2.1 INTRODUÇÃO

Cultura empresarial ou organizacional é o conjunto de valores, costumes e ideais inseridos no dia a dia de uma empresa. Sua importância está no fato de ser uma excelente forma de engajar os colaboradores e torná-los mais unidos e motivados em prol do sucesso da organização, além de proporcionar bem estar e qualidade de vida à equipe. Alguns exemplos de atividades, atitudes ou rituais que podem compor uma cultura organizacional são: um ambiente descontraído com música e liberdade para conversar, *happy hours*[1], comemorações de aniversários e conquistas da equipe, premiações e refeições gratuitas (MOTTA; VASCONCELOS, 2002). O bem estar de um colaborador em uma empresa é fundamental, primeiro porque incentiva a produtividade, e depois, porque ajuda a amenizar o estresse que muitas vezes algumas atividades do trabalho podem trazer (MASLACH; SCHAUFELI; LEITER, 2001).

Pensando no que foi dito, uma empresa de tecnologia, que preza a construção de uma forte cultura organizacional, deu início a um controle interno de consumo de produtos, batizado de *Pilas*. São exemplos desses produtos: balinhas, paçocas, biscoitos, chocolates, frutas, iogurtes, entre outros. Para consumi-los, cada colaborador recebe, por mês, um determinado saldo de uma moeda fictícia chamada *Pila*.

O abastecimento do estoque e a distribuição da moeda, é responsabilidade de um membro da equipe, eleito mês a mês de forma democrática como "prefeito". Atualmente, o controle de itens no estoque, de saldo dos consumidores e movimentações de compra, são controlados através de uma planilha compartilhada no Google Sheets[2]. O projeto *Pilas*, conta também com um agregado chamado de "Fome Zero", através do qual, a equipe tem livre acesso a alimentos básicos para lanches, como café, leite, pão, manteiga, doce de leite, bolacha, açúcar, dentre outras opções.

Dessa forma, o objetivo com o *Pilas* é disponibilizar todo mês **guloseimas**[3] **adicionais**, e que normalmente uma empresa não provê gratuitamente. Contudo, se esse tipo de produto fosse de livre consumo, haveria um grande risco de todo o estoque ser consumido compulsivamente em poucos dias e de forma desproporcional, mais por alguns do que por outros. Adicionalmente, existe a preocupação com a saúde dos colaboradores, pois a ingestão exagerada desses alimentos ultraprocessados cotidianamente, pode ser prejudicial (BIELEMANN et al., 2015).

O modelo atual em execução é falho, pois existem muitas brechas para inconsistências, visto que cada um anota seus gastos, e por engano ou mesmo má fé, pode acabar fazendo

[1] Reunião ou comemoração informal, normalmente envolvendo colegas de trabalho ou estudo, realizada após expediente de trabalho com petiscos e bebidas.

[2] Serviço de planilhas online, website oficial: https://sheets.google.com/

[3] Gíria utilizada para se referir a gulas, doces.

anotações incorretas, não havendo uma forma de fiscalizar isso. Por conseguinte, destaca-se a necessidade de implementação de uma solução computacional com maior poder de gerência em relação às atuais planilhas, promovendo controle e segurança nas transações de consumo de guloseimas da empresa.

Dito isso, o presente documento propõe o desenvolvimento de um sistema web com versão também em PWA, que fará toda essa gestão, hoje realizada através de planilhas. Por meio deste sistema, será possível o cadastro de usuários, a verificação de saldo e extrato de *Pilas* e a realização de compra dos itens cadastrados. Além de possuir uma área exclusiva para o prefeito, que fará a gestão dos produtos, saldos, usuários e permissões.

2.2 OBJETIVOS

2.2.1 Objetivo Geral

Desenvolvimento de um sistema para controle do consumo de guloseimas pelos colaboradores de uma empresa de tecnologia.

2.2.2 Objetivos Específicos

- Desenvolver um sistema web que possibilite o cadastro de usuários com diferentes níveis de permissão;
- Permitir que o prefeito gerencie produtos;
- Permitir que o prefeito controle os saldos dos usuários, podendo conceder bônus ou confiscar *Pilas*;
- Permitir ao usuário a consulta de saldo e extrato de consumo de *Pila*;
- Possibilitar aos usuários realizarem movimentações de compra de itens disponíveis;
- Gerar relatórios de consumo para aprimorar o controle de estoque;
- Possuir compatibilidade com diversos tipos de sistemas operacionais, incluindo os de dispositivos móveis;
- Publicar a aplicação em PWA;
- Promover facilidade e agilidade na localização e compra de itens através de interface minimalista e intuitiva;
- Permitir que o sistema esteja disponível exclusivamente de forma online, evitando que o usuário precise realizar instalações e configurações.

2.3 TRABALHOS CORRELATOS

Dada a necessidade de um sistema para o controle supracitado, foram identificadas e analisadas algumas soluções existentes que poderiam automatizar o processo e auxiliar neste

objetivo, como o Nex[4], o Contestoque[5] e o Estoque Simples[6].

O **Nex**, é um sistema para gestão de pequenos comércios, que permite o cadastro de produtos, fornecedores, histórico de movimentação, conferência de estoque, entre outros recursos relacionados. Embora possua funções semelhantes as buscadas, o método utilizado é pensado exclusivamente para comércios, não se encaixando com o *Pilas*, onde cada colaborador poderá agir autonomamente, comprando seus próprios itens dentro do sistema. O sistema Nex como um todo é complexo para o propósito desejado e contém vários processos desnecessários para o *Pilas*, além de não possuir uma gestão de usuários com saldo. A Figura 1 traz um exemplo de tela do sistema Nex.

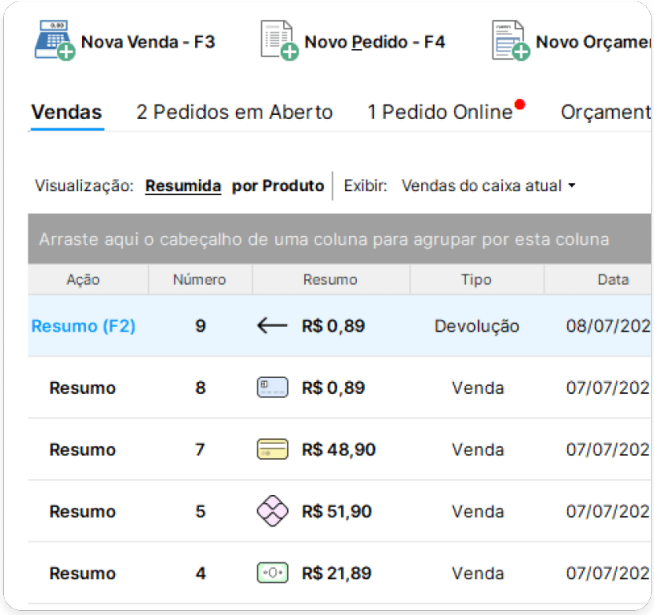


Figura 1 – Tela de resumo de vendas
Fonte: Nex (2022)

Já o **Contestoque** foi desenvolvido de forma exclusiva para controlar estoque. Ele permite o cadastro de produtos com data de vencimento e código de barras, contando com um leitor que utiliza a câmera do celular e permitindo também o controle de quantidade dos itens (Figura 2. Esse aplicativo atende principalmente aqueles que possuem estoque de alimentos, dos quais precisam ter completo controle, como comércios, restaurantes, ou cozinhas de instituições em geral. Pode também atender a necessidade de organizações que possuem outros tipos de estoque, como de ferramentas e equipamentos. Apesar de englobar algumas das funções buscadas para o *Pilas* na parte de estoque, o Contestoque não possui gestão de usuários ou de qualquer tipo de moeda, tampouco funções para compra de produtos, o que vem a descartar seu uso.

[4]https://www.nextar.com.br
[5]http://www.contestoque.com.br/
[6]https://play.google.com/store/apps/details?id=br.com.gameloop.estoquesimples

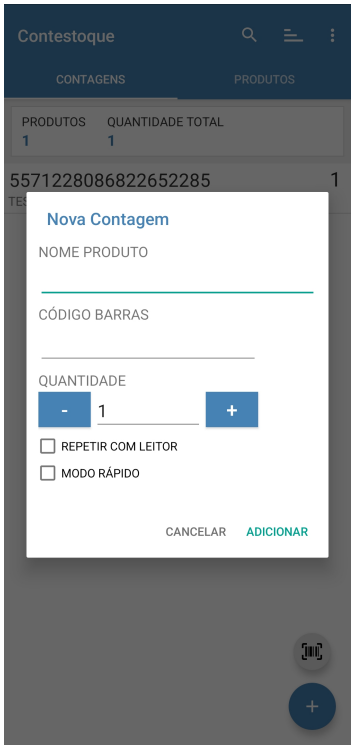


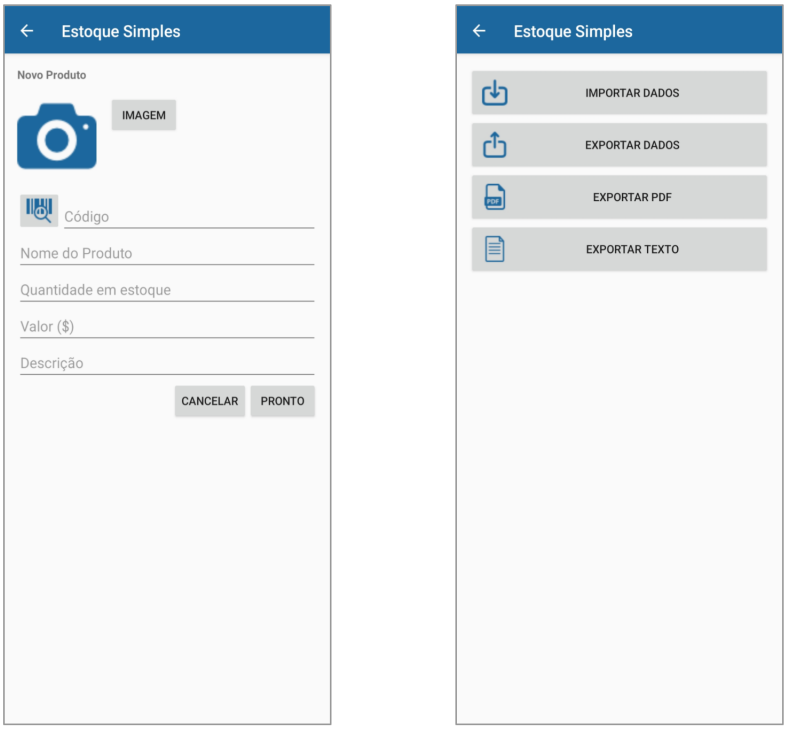
Figura 2 – Tela de registro de um produto no aplicativo Contestoque
Fonte: Contestoque (2022)

Por fim, o aplicativo **Estoque Simples**, muito semelhante ao anterior, foi pensado para controle de estoque (Figura 3 (a)). Algumas diferenças com relação ao Contestoque são a possibilidade de cadastro de imagem dos produtos, a não possibilidade para cadastro de validade dos mesmos e a função para exportar os itens cadastrados para PDF (Figura 3 (b)). Ele atende a cenários onde há necessidade de controle de estoque de itens em geral, como equipamentos e ferramentas, por exemplo. Da mesma forma que o aplicativo citado anteriormente, este possui carência de funções quando confrontado com o *Pilas*, que será desenvolvido, por não permitir gestão de usuários, saldos, funções de compra, entre outras mais específicas.

2.4 PILAS

No primeiro semestre de 2022, foi iniciado na disciplina de Programação para Dispositivos Móveis, na UTFPR Campus Guarapuava, o desenvolvimento de um aplicativo com a ideia de suprir as necessidades apontadas no presente trabalho. Para desenvolvimento inicial desse aplicativo, foram utilizadas tecnologias como React Native, Node.js, Expo e Firebase. Proporcionando a autenticação de usuários com diferentes permissões, cadastros de produtos e opção para compra dos mesmos utilizando o saldo disponível.

Após o término da disciplina, foi considerada a possibilidade de dar continuidade ao aplicativo, tendo em vista uma empresa de tecnologia com interesse em automatizar o seu processo, mas foram observadas algumas limitações. Este teria que ser implantado para utilização dos colaboradores da empresa, sendo disponibilizado nas lojas de aplicativos, tanto



(a) Registro de um produto (b) Exportação e importação de dados

Figura 3 – Telas do aplicativo Estoque Simples
Fonte: EstoqueSimples (2022)

de sistemas Android, quanto iOS. Analisando-se os termos dessas lojas, observou-se que ambas as plataformas exigem o pagamento de licenças e o custo para disponibilizá-lo para iOS[7] seria inviável em um primeiro momento.

Além disso, a disponibilização do sistema como um aplicativo móvel nativo nas lojas de Android e iOS não permite o teste instantâneo de novas funcionalidades e/ou correção de bugs, visto que cada plataforma possui um fluxo de *deploy*[8] próprio, que muitas vezes requer um período de espera até a aprovação de novas versões. Diante disso, cada atualização no sistema pode levar desde algumas horas, até dias para que todos os usuários recebam a notificação da nova versão e atualizem em seus dispositivos. Em contrapartida, uma aplicação web não requer essa espera, dado que feito o *deploy*, todos os usuários ficam imediatamente aptos a utilizarem o sistema atualizado.

Tendo em vista essas limitações, descartou-se a continuidade do aplicativo, dando início à ideia de desenvolvimento de um sistema web que inclui todos os tipos de usuários no que diz respeito aos seus dispositivos e fornece atualizações de maneira transparente. Destaca-se que a construção do sistema utilizando a abordagem mobile-first (ROTH, 2019) visando publicação como um PWA é uma possibilidade a ser investigada, pois permitiria um uso amigável do sistema em dispositivos móveis.

[7]Apple Developer Program: https://developer.apple.com/support/compare-memberships/
[8]Do inglês "implantar", é um termo utilizado em programação em seu sentido literal, para uma aplicação que é implantada.

2.5 FLUXO DE DESENVOLVIMENTO

O primeiro passo para a execução deste projeto é o levantamento de todos os seus requisitos, sendo esses funcionais e não funcionais, os quais serão classificados através do método MoSCoW (KUHN, 2009), conforme suas prioridades. O nome MoSCoW é um acrônimo das palavras em inglês **M**ust, **S**hould, **C**ould e **W**ouldn't, onde as letras "o", foram adicionadas apenas para tornar o acrônimo mais pronunciável. Este método traz 4 (quatro) tipos de classificações, sendo elas, requisitos que o projeto:

- Precisa ter (Must have): aqueles requisitos que são essenciais e obrigatórios para o funcionamento do projeto;
- Deveria ter (Should have): requisitos importantes, mas não essenciais;
- Poderia ter (Could have): são os requisitos desejáveis, mas não vitais, eles podem ou não ser atendidos, conforme disponibilidade de tempo e recursos;
- Não teria (Wouldn't have): requisitos que estão fora do atual escopo.

Feito o levantamento e a classificação dos requisitos, serão definidas as melhores tecnologias a serem utilizadas para desenvolvimento do projeto, analisando as que mais se adéquam ao objetivo proposto. Isso inclui tecnologias para *frontend*, *backend*, banco de dados e *frameworks* em geral que auxiliam e agilizam os processos. Em seguida, será construído um Design System (MACDONALD, 2019) com o auxílio da plataforma Figma[9], onde serão definidos padrões de layout e paletas de cores, por exemplo.

Uma vez que todas as definições inicias tenham sido feitas, os requisitos levantados serão passados para *features*[10], estabelecendo junto do orientador um prazo estimado para o desenvolvimento de cada uma delas, o que possibilita um planejamento mais preciso de tempo para o projeto como um todo. Menciona-se que tais prazos podem sofrer alterações na medida em que as funcionalidades forem desenvolvidas e suas reais complexidades se tornem mais evidentes.

As *features* serão gerenciadas pelo modelo **Feature Branch Workflow**, através da ferramenta Git. Este modelo consiste em um desenvolvimento guiado por *features* que seguem sempre um mesmo ciclo, o qual funcionará conforme ilustrado no fluxograma da Figura 4.

O ciclo tem início na escolha da *feature* a ser desenvolvida, a qual corresponde a um requisito que foi levantado. É neste momento que os prazos definidos poderão ser revisados pela aluna e pelo orientador caso haja necessidade. Em seguida é criada uma nova *branch* local para que se inicie a escrita dos códigos de programação, realizando novos *commits* a cada etapa até que a mesma seja concluída. Essa divisão em vários *commits*, auxilia para que haja um versionamento do código, facilitando que alterações sejam revertidas se necessário. Uma vez que a tarefa tenha sido desenvolvida, é feito o *push* dessas alterações para a *branch*

[9] https://www.figma.com/

[10] Termo utilizado para se referir a funcionalidades ou recursos a serem desenvolvidos para um sistema computacional, por exemplo.

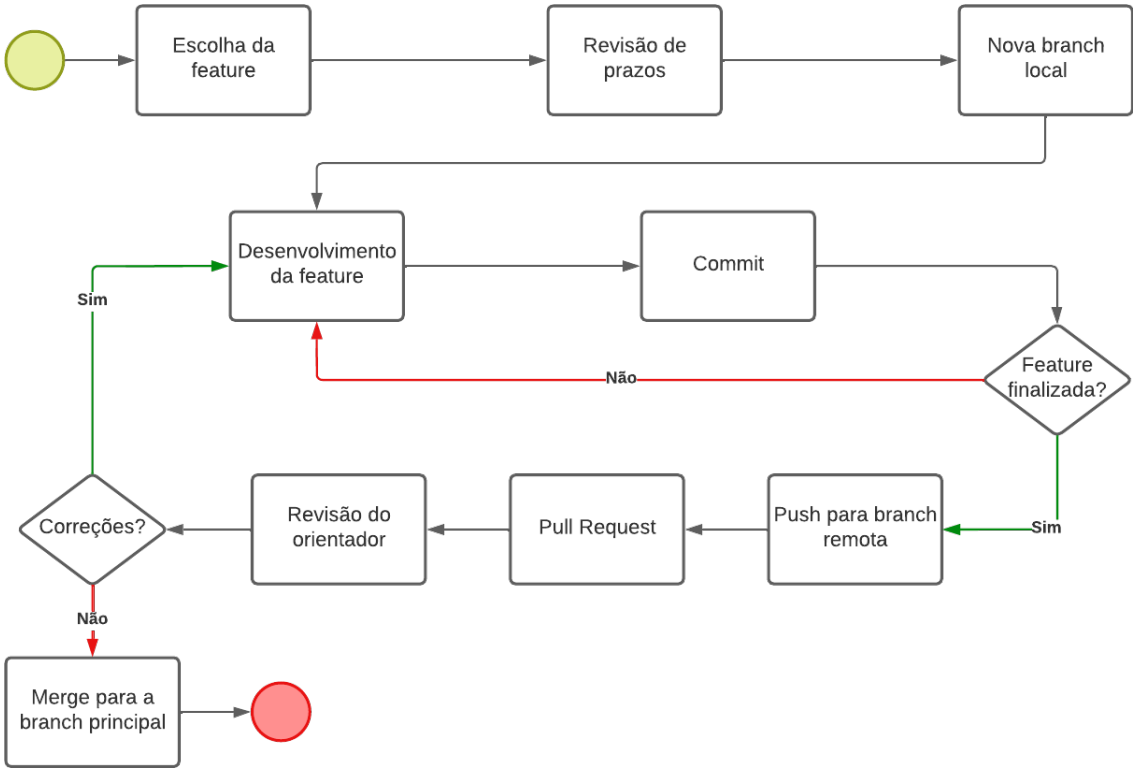


Figura 4 – Fluxograma do ciclo da feature
Fonte: A autora

remota e através do GitHub[11] é criada uma *Pull Request* (PR). Neste ponto, é solicitado que o professor orientador faça a revisão dos códigos e os apontamentos necessários e caso hajam correções ou alterações a serem realizadas, estas devem ser cumpridas e enviadas para a *branch* remota novamente para aguardarem nova revisão. Caso contrário, é feito o *merge* para a *branch* principal finalizando assim, o ciclo da *feature*.

A partir do momento em que as principais *features* tenham sido desenvolvidas e que obtenha-se um Produto Mínimo Viável – em inglês *Minimum Viable Product* ou MVP (LENARDUZZI; TAIBI, 2016) – , este será implantado na empresa em caráter de testes, para que os colaboradores possam dar suas opiniões e queixas a respeito da usabilidade e possíveis conflitos que possam surgir, os quais serão resolvidos a partir da criação e desenvolvimento de novas *features*.

2.6 CONSIDERAÇÕES FINAIS

A fim de alcançar o que é proposto será elaborado um conjunto preliminar de requisitos coerente ao problema com atribuições de prioridades, visando inicialmente construir um sistema funcional que atenda aos principais requisitos, ou seja, um MVP. Adicionalmente, pretende-se colocar esse sistema em teste para a equipe da empresa, coletando retornos e opiniões dos mesmos.

[11] https://github.com/

Entre algumas das dificuldades que podem ser enfrentadas, está a tarefa de atender a usuários de diferentes sistemas operacionais, incluindo dispositivos móveis, além da possibilidade de entregar um sistema que atenda a outras empresas que poderiam ter interesse em implantá-lo.

Destaca-se a importância desse projeto, principalmente para a empresa que levantou essas necessidades, já que automatiza um processo que até então é feito manualmente e é muito suscetível a falhas, além de unificar as funções do projeto *Pilas* em uma mesma plataforma, trazendo mais praticidade e agilidade aos usuários. Ademais, se outra empresa se interessar em adotar um projeto como esse para sua equipe, em prol de um clima organizacional melhor, esta poderá também utilizar o sistema que será desenvolvido.

2.7 PLANEJAMENTO DO TRABALHO

O planejamento do trabalho de TCC que será desenvolvido pela aluna, ao longo do período letivo, está descrito no cronograma do Quadro 1. Neste cronograma constam todas as atividades com seus respectivos prazos para o cumprimento.

Quadro 1 – Cronograma de Atividades.

Atividades	**Nov**	**Dez**	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**	**Jul**
1. Revisão dos apontamentos da banca									
2. Revisão bibliográfica									
3. Redação do projeto de TCC									
4. Defesa do projeto de TCC									
5. Desenvolvimento do sistema proposto									
6. Escrita da Monografia de TCC									
7. Elaboração da apresentação final									
8. Defesa final do TCC									

2.7.1 DA PROPOSTA AO PROJETO

No período entre a aprovação da presente proposta e entrega do projeto, serão desenvolvidos e definidos os seguintes pontos:

- Desenvolvimento de um conjunto preliminar de requisitos;
- Definição e descrição das tecnologias a serem utilizadas; e
- Construção de um Design System.

Dessa forma, os itens descritos estarão apresentados no projeto de TCC.

2.8 RECURSOS NECESSÁRIOS

Para o desenvolvimento deste trabalho, será necessário um computador com acesso a Internet, bem como ferramentas de edição de código e acesso ao Git e GitHub. Na fase de implantação, será necessário um servidor para hospedagem e um domínio para proporcionar aos usuários o acesso à aplicação. Estes dois últimos serão fornecidos pela empresa que utilizará o sistema.

2.9 HORÁRIO DE TRABALHO

O horário destinado para realização das atividades do TCC, bem como o horário destinado para a reunião semanal/quinzenal com o orientador estão descritos no cronograma do Quadro 2. Este horário é definido com orientador levando em consideração a complexidade do trabalho a ser desenvolvido.

Quadro 2 – Horário de Trabalho.

Horário	**Seg**	**Ter**	**Qua**	**Qui**	**Sex**	**Sab**
07h30 - 08h20						
08h20 - 09h10						
09h10 - 10h00						
10h10 - 11h00						
11h00 - 11h50						
13h00 - 13h50						
13h50 - 14h40						
14h40 - 15h30						
15h40 - 16h30						
16h30 - 17h20						
17h20 - 18h10						
18h50 - 19h40				TCC		
19h40 - 20h30				TCC		
20h30 - 21h20		Orientação	TCC	TCC	TCC	
21h30 - 22h15		TCC	TCC		TCC	

Referências

BIELEMANN, R. M. et al. Consumo de alimentos ultraprocessados e impacto na dieta de adultos jovens. **Revista de Saúde Pública**, SciELO Public Health, v. 49, p. 28, 2015. Citado na página 2.

CONTESTOQUE. [S.l.], 2022. Disponível em: <http://www.contestoque.com.br/>. Acesso em: 12 de outubro de 2022. Citado na página 5.

ESTOQUESIMPLES. [S.l.], 2022. Disponível em: <https://play.google.com/store/apps/details?id=br.com.gameloop.estoquesimples/>. Acesso em: 12 de outubro de 2022. Citado na página 6.

KUHN, J. Decrypting the moscow analysis. **The workable, practical guide to Do IT Yourself**, v. 5, 2009. Citado na página 7.

LENARDUZZI, V.; TAIBI, D. Mvp explained: A systematic mapping study on the definitions of minimal viable product. In: IEEE. **2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.], 2016. p. 112–119. Citado na página 8.

MACDONALD, D. **Practical ui patterns for design systems: Fast-track interaction design for a seamless user experience**. [S.l.]: Apress, 2019. Citado na página 7.

MASLACH, C.; SCHAUFELI, W. B.; LEITER, M. P. Job burnout. **Annual review of psychology**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 52, n. 1, p. 397–422, 2001. Citado na página 2.

MOTTA, F. C. P.; VASCONCELOS, I. F. G. A cultura organizacional. **MOTTA, Fernando C. Prestes. Teoria geral da administração**, v. 3, 2002. Citado na página 2.

NEX. [S.l.], 2022. Disponível em: <http://www.nextar.com.br/>. Acesso em: 05 de outubro de 2022. Citado na página 4.

ROTH, R. What is mobile first cartographic design. In: INTERNATIONAL CARTOGRAPHIC ASSOCIATION BERN, SWITZERLAND. **ICA Joint Workshop on User Experience Design for Mobile Cartography**. [S.l.], 2019. Citado na página 6.