UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL DE SÁ BESEN

DAVID PEREIRA CABRAL

LUCAS MOROSKI

RHONNER MATHEUS NUNES AFANACI

**FIDELITI : SISTEMA DE FIDELIDADE PARA EMPREENDIMENTOS**

.

CURITIBA

2020

UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL DE SÁ BESEN

DAVID PEREIRA CABRAL

LUCAS MOROSKI

RHONNER MATHEUS NUNES AFANACI

**FIDELITI : SISTEMA DE FIDELIDADE PARA EMPREENDIMENTOS**

Trabalho de conclusão apresentado ao Curso Superior

de Tecnologia em Análise e Desenvolvimento de

Sistemas, Setor de Educação Profissional e

Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas

Montaño

CURITIBA

2020

AGRADECIMENTOS

(ALGUMA FRASE ESCOLHIDA)

**RESUMO**

Partindo da premissa de que os clientes fazem o empreendimento e levando ela como inspiração, é notável que nos dias atuais não há grandes obstáculos para empreendedores aplicarem suas ideias. Porém, após algumas pesquisas de campos realizadas pela nossa equipe em diversos estabelecimentos como: barbearias, hamburguerias, lanchonetes, salões de beleza, etc, foi observado que estes estabelecimentos ou seus parceiros utilizam de fidelidades esquematizadas por cupons físicos, onde muitas vezes ocorre o extravio deles, gerando divergências ao estado real da fidelidade do cliente. Tendo isto em vista, temos como objetivo facilitar a aplicação de cupons de fidelidade, e auxiliar a reabertura de muitos pequenos comércios na pós-pandemia, proporcionando uma novidade favorável tanto  para os clientes consumidores quanto para os empreendimentos,  utilizando da era tecnológica em que estamos vivendo, onde a grande maioria das pessoas possuem um celular.

Oferecemos um sistema de aplicação e gerenciamento de fidelidades para os empreendimentos e seus clientes consumidores, contando com relatórios relativos às fidelidades utilizadas e a maneira que elas agem como holofotes para a empresa. Contamos com um sistema web, possuindo o back-end da API em .NET framework, front-end no framework Angular10, com portabilidade PWA para aplicação mobile e armazenamento dos dados no SQL Server. Visando a simplicidade, o nome deste sistema será FideliTI, unificando fidelidade e Tecnologia da Informação.

**ABSTRACT**

**LISTA DE ILUSTRAÇÕES**

**LISTA DE TABELAS**

**LISTA DE ABREVIATURAS E SIGLAS**

**SUMÁRIO**

**NÍVEL 1** 1

NIVEL 2 2

Nível 3 3

**NÍVEL 1** 4

NÍVEL 2 5

Nível 3 6

**1 INTRODUÇÃO**

Segundo Jeff Bezos (presidente e CEO da Amazon), empresas que focam no cliente e o escutam estão sempre inovando e criando estratégias de longo prazo, que garantem uma vida longa e próspera ao negócio. Partindo desta premissa, normalmente tanto os empreendimentos recém formados como os mais antigos partilham do mesmo problema: concorrência. Utilizando este motivo como meta, partimos com a ideia de fazer o básico que  difere os poucos empreendimentos de sucesso dos muitos existentes que almejam o topo.

A fidelidade com empreendimentos, como o próprio nome já diz, cria certa ideia de exclusividade com o cliente, utilizando de possíveis promoções, queimas de estoque, compra contínua, etc para aumentar exponencialmente a popularidade do estabelecimento com relação à seus concorrentes que, na maioria dos casos não disponibilizam promoções com alternativas aplicadas com prazos regulares e irregulares; tampouco com aproveitamento da tecnologia para seu marketing.

Um dos muitos problemas que os empreendedores têm como obstáculo, segundo a Vhsys (blog), é não conhecer seus clientes e não se aproveitar de seu público-alvo. Conforme SEBRAE(2015) :

“É importante reconhecer que, para alguns, preço é mais importante do que a qualidade. Para outros, a qualidade é o que conta no momento de adquirir um produto. Existem ainda aquelas pessoas que consideram o atendimento muito mais relevante do que preço ou qualidade. Tudo vai depender do contexto da compra e da melhor opção oferecida.”(SEBRAE, Programa de Fidelidade, Uma Chance de manter as vendas na crise).

Também é importante reconhecer que, para alguns, preço é mais importante do que a qualidade. Para outros, a qualidade é o que conta no momento de adquirir um produto. Tudo vai depender do contexto da compra e da melhor opção oferecida, ou seja, além das fidelidades é possível trabalhar com estes dados, apresentando-os de forma que sejam fatores chave para tomadas de decisões.

* 1. PROBLEMA

A cada ano mais recorrentes, as recompensas beneficentes são aquelas que visam ajudar determinada instituição ou empresa com o seu crescimento. Ou seja, não necessariamente o consumidor Irá sempre comparar na mesma loja, com isso o cliente ao fazer uma compra, está buscando algum benefício de alguma forma.

A fidelidade é um compromisso a longo prazo. Que no mercado atual, devido à grande quantidade de produtos e serviços que são ofertados, é difícil fazer o consumidor comprar ou consumir sempre da mesma marca ou empresa. Já não precisa somente ter um bom produto e um atendimento ao cliente excelente: precisa gerar engajamento para fazer o cliente consumir cada vez mais o seu produto ou serviço.

Fidelizar o cliente, ser capaz de tornar a sua marca numa referência para o cliente lembrar de você primeiro sempre que tem uma necessidade específica, só será possível se trabalhar muitos aspectos de seu negócio e se relacionar da maneira mais adequada durante o período de tempo suficiente para que isso vire uma realidade.

O cliente de hoje quer sentir que está sendo tratado de maneira personalizada. Quer se sentir especial. O processo relacionado à importância da personalização é complexo, porque é preciso considerar várias necessidades do consciente e subconsciente. As pessoas querem se sentir especiais, por isso, enviar campanhas massivas oferecendo produtos que não vão suprir suas necessidades ou interesses e não “alimentar” o relacionamento podem significar uma perda de confiança quase irreparável em sua marca ou em seu negócio.

Um erro muito comum dos programas de fidelidade é oferecer aquilo que as empresas querem e não aquilo que os consumidores precisam. Isso, por vezes, acontece quando as promoções são realizadas por haver um excedente de estoque, quando o que na verdade precisa fazer é entender quais os produtos ou serviços que são mais relevantes para seus clientes, mesmo que que isso implique a compra de produtos fora de seu nicho.

O ser humano, de uma forma geral, tem aquilo a se pode chamar de sistema de gratificação imediata, ou seja, uma preferência por aquilo que pode conseguir agora em prejuízo daquilo que poderia ganhar mais tarde mesmo que o benefício fosse maior. Por isso, se atrasar demasiado a recompensa, pode fazer o cliente pensar que não vale a pena considerando o tempo que precisa esperar para conseguir ela.

Além do mais, um intervalo de tempo demasiado prolongado entre o ato que o cliente precisou realizar para ganhar a sua recompensa e o recebimento dela pode esfriar a relação do consumidor com a marca. Ou seja, o cérebro não vai estabelecer uma relação de causa-efeito, por isso, as probabilidades de adesão são muito menores.

* 1. JUSTIFICATIVA

Um programa de fidelidade significa ir muito além de juntar pontos e trocar por prêmios. Na verdade, ele te fornece dezenas de recursos para criar vínculos com os clientes, tornando as ações de marketing de uma empresa mais eficiente e, no final das contas, aumentar as vendas e serviços prestados. Um programa de fidelidade se ajusta para qualquer negócio com vendas e serviços recorrentes. E cabe perfeitamente nas pequenas e médias empresas. Criar uma fidelidade e abrir um canal de relacionamento com o cliente, fazendo com que o relacionamento entre consumidor e empresa seja uma receita infalível para aumentar as vendas. Normalmente as pessoas desejam criar vínculos com as marcas por meio de consumo que seja coerente com aquilo que acreditam. Este é o “algo a mais” que vai destacar o seu negócio no meio de todos os concorrentes.

Um programa de fidelidade é ideal para construir um relacionamento duradouro com o seu cliente. É o elo por meio do qual o empresário pode se comunicar com seu consumidor, e saber dos seus desejos e visualizar em quais clientes ele deve dar mais atenção, fazendo que com isso fique mais centralizado o foco de divulgação do marketing para o seu publico alvo.

Não é à toa que grandes companhias investem milhões em programas de fidelidade para manter a empresa próxima dos clientes. Mas as empresas de pequeno e médio porte, não precisa gastar muito para investir no relacionamento com o seu consumidor.

E de grande importância Defender o preço dos seus produtos para efetuar grande descontos com o seu sistema de vendas. Com isso a empresa fica dependente dos descontos e os seus clientes também. Quando percebesse que a um grande salto nas vendas e o seu consumidor gosta de um bom negócio. O cliente se acostuma com os descontos ele não vai comprar quando o preço voltar ao normal. Descontos reduzem a margem de lucro da empresa, e isso pode provocar uma queda de rendimento no faturamento mensal ou anual da empresa. No caso um programa de fidelidade te ajuda a evitar uma guerra de preços com seus concorrentes porque adiciona valor aos seus produtos e serviços e à experiência de compra do cliente.

Um bom programa de fidelização dificulta o avanço da concorrência. Um cliente fiel ficará contigo independentemente dos atrativos da concorrência, que terá de gerar várias ações para chamar sua atenção e captar os seus próprios. Por conta desses fatores, é muito mais produtivo para uma empresa investir na perpetuação de um cliente já existente do que focar em conquistar novos. Acho que não preciso dar mais motivos para você entender o quão importante é [investir na fidelização](http://52.205.157.7/fidelizacao/fidelizacao-de-clientes/" \t "https://blog.izio.com.br/fidelizacao/programa-de-fidelidade/_blank).

* 1. OBJETIVOS
     1. OBJETIVO-GERAL

Desenvolver um sistema web, permitindo a existência de um laço duradouro entre empresas e seus consumidores finais.

* + 1. Objetivos específicos

• Oferecer diferentes planos de assinatura para empresas;

• Facilitar o uso de fidelidades com o aumento da praticidade do sistema;

• Exibir gráficos no sistema com informações relevantes para a empresa;

• Exibir para uma empresa uma lista de produtos e suas respectivas fidelidades;

• Flexibilizar o cadastro de fidelidades, deixando a empresa escolher a melhor estratégia;

• Permitir que uma empresa consiga inativar uma determinada fidelidade;

• Exibir para uma empresa e seus funcionários o progresso de fidelidades ativas de um determinado cliente;

• Analisar informações para melhorar a experiência do usuário;

* 1. ESTRUTURA DO DOCUMENTO

Este documento apresenta as informações do desenvolvimento do projeto e está estruturado conforme a seguir.

O Capítulo 2 tem como objetivo trazer a fundamentação teórica, que complementa o assunto tratado no Capítulo 1.

O Capítulo 3 possui a descrição da metodologia aplicada pela equipe, as

tecnologias utilizadas para desenvolver este projeto, o cronograma de atividades desenvolvidas e a descrição das atividades realizadas.

No Capítulo 4 é possível visualizar a arquitetura de software desenvolvida

para o projeto e informações sobre o funcionamento do aplicativo móvel e o módulo web, descrição de funcionalidades acompanhadas das respectivas telas que foram implementadas.

1. **FUNDAMENTAÇÃO TEÓRICA**

Neste capítulo serão apresentados temas, conceitos e as tecnologias utilizadas de forma aprofundada, as quais foram essenciais para o desenvolvimento do projeto.

* 1. FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO
     1. Os Empreendimentos na Pandemia

Segundo uma matéria do O Globo, mais de 700 mil empresas fecharam suas portas devido à crise causada pela pandemia e não têm previsão de reabertura; Causa esta que deixa os empreendimentos que se mantém ativos com um alto nível de concorrência tendo em vista que, a maioria dos empreendimentos na ativa estão inovando em todos os aspectos para tentar aumentar ou até mesmo apenas manter seus clientes.

* + 1. Fidelidades em empreendimentos

Já não é novidade comprar algum produto ou serviço e receber propostas de tentadores brindes ou descontos para continuar as compras. Isso estava se tornando mais comum devido ao aumento quantitativo das concorrências e atualmente está se tornando comum devido ao aumento qualitativo das mesmas (resultado da pandemia). Estas mudanças exigem que o empreendimento se adapte ao mercado à sua volta, sempre oferecendo algo inovador ou tentador para manter ou até mesmo aumentar o número de consumidores.

* + 1. Lucratividade de Empreendimentos com Fidelidade

Qualquer que seja o negócio, possuir clientes leais é vital para a sobrevivência da empresa. Segundo o site Brasilturis, manter clientes existentes é de cinco a 25 vezes mais barato que adquirir novos.

Todo empreendimento possui alguém que avalia ou faz análises dos clientes para criar alguma proposta ou se antecipar à compra de recursos para aumentar o investimento no público-alvo. Estes dados quando somados à programas que promovem a fidelização dos clientes através das premiações e disponibilizam uma quantidade extraordinária de dados, que servem de incentivo para a compra contínua. Naturalmente, com estas estratégias atingindo o público alvo, é comum o aumento de clientes, seguido pelo aumento de lucros.

* 1. FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS

Os temas relacionados as tecnologias utilizadas serão apresentados a seguir.

* + 1. Linguagens de Programação

O site da Rockcontent relata que são várias as linguagens de programação existentes, porém com funções distintas tais quais: criação de jogos, sites, softwares, etc.

Os trabalhadores desta área são conhecidos como programadores ou desenvolvedores. Responsáveis por escrever todas as linhas de código que irão compor o sistema final. Existem também os analistas que por sua vez, fazem o trabalho de levantar os requisitos (informações), regras, etc, além de criar uma documentação sobre o projeto que será desenvolvido.

Todo sistema apresentável a um usuário possui duas partes: *back-end* e *front-end. O* parte responsável pelo funcionamento do sistema é o *back-end;* lá se encontra todas as regras, aplicação de lógica para manuseio de dados, etc. Já a parte visual do sistema, área onde o usuário terá interação, se chama *front-end.*

A linguagem de [programação](https://rockcontent.com/br/blog/marketing-e-programacao/" \t "https://rockcontent.com/br/blog/linguagem-de-programacao/_blank) é a base para a construção de todos os aplicativos digitais usados no dia a dia e são classificados em dois tipos principais: linguagem de baixo e alto nível.

A linguagem de alto nível visam facilitar o trabalho do programador, pois usam instruções e comandos que são mais fáceis de serem entendidos. Estas linguagens utilizam idiomas conhecidos (português, espanhol, inglês,etc) para serem usados como comandos aplicáveis.

Já as linguagens de baixo nível são totalmente orientadas à máquina, utilizando isto como uma interface para integrar hardware e software.

* + - 1. Linguagem de programação Java e Spring Framework

Lançado em 1995, o Java é uma linguagem de programação orientada a objetos que atualmente faz parte do núcleo da Plataforma Java.

A Orientação a Objetos, ou Programação Orientada a Objetos, do inglês *Object-Oriented Programming*, é um tipo de paradigma de análise, para a programação de sistemas no qual todos os elementos inseridos são objetos.

Nascendo diretamente do Java, O Spring Framework é um framework de aplicação de código-fonte aberto popular que pode facilitar o desenvolvimento do Java EE. Ele consiste em um contêiner, um framework para gerenciar componentes, e um conjunto de serviços de snap-in para interfaces de usuário, transações e persistência da Web. Uma parte do Spring Framework é o Spring Web MVC, um framework MVC extensível para criação de aplicações Web.

O IDE oferece suporte incorporado para Spring Framework 4.x e 3.x. As bibliotecas de framework são fornecidas com o IDE e adicionadas automaticamente ao classpath do projeto quando o framework é selecionado. As definições de configuração são fornecidas, tais como nomeação e mapeamento do DispatcherServlet do Spring Web MVC. A biblioteca JSTL pode, como opção, ser registrada na criação do projeto. O suporte para arquivos de configuração de bean Spring XML também é fornecido.

* + 1. Linguagem de programação JavaScript

O JavaScript é uma linguagem interpretada utilizada para programação de páginas web. Segundo a MDN (sigla para MozDevNet), uma definição que podemos utilizar para entendimento do que é o JavaScript:

“[...] é uma linguagem de scripting baseada em protótipos, multi-paradigma e dinâmica, suportando os estilos orientado a objetos, imperativo e funcional. JavaScript roda no client side da web, o que pode ser usado para projetar / programar o comportamento de uma página web a partir da ocorrência de um evento. JavaScript pode funcionar tanto como uma linguagem procedural como uma linguagem orientada a objetos. Objetos são criados programaticamente em JavaScript, onde métodos e propriedades são anexados a objetos vazios em tempo de execução [...].”

Uma variante dessa linguagem, é o Typescript, que apesar de construída em cima do JavaScript, fornece o recurso de tipagem de artigos da linguagem, seja esse artigo, uma classe, um objeto, um parâmetro ou um valor de retorno de um método, tornando o desenvolvimento de aplicações JavaScript mais robusto e escalável.

Para execução do JavaScript em navegadores como dito anteriormente, utiliza-se o NodeJS, um ambiente de execução assíncrono projetado para desenvolvimento de aplicações JavaScript e estruturado no Chrome’s V8, um interpretador de alta performance escrito em C++.

* + - 1. Angular

O Angular é um framework de desenvolvimento de aplicações JavaScript e uma plataforma de criação dessas aplicações. Suporta também o desenvolvimento de aplicações móveis e para área de trabalho, por meio de suas vertentes, como por exemplo o Ionic. Utiliza a linguagem TypeScript para desenvolvimento e é executado pelo ambiente NodeJS, assim como outros frameworks baseados em JavaScript.

* + - 1. Angular Material

É uma biblioteca de componentes estilizados para aplicações móveis e para área de trabalho desenvolvidas em Angular, estruturada no sistema de design Material criada pela Google.

* + 1. Banco de Dados

O termo “banco de dados” pode-se ser pensado em seu sentido literal moderno, onde é a organização e armazenagem de informações sobre um domínio específico. Basicamente o local onde serão armazenados os dados do sistema especificado.

Para tal ação é utilizado um Sistema de Gerenciamento de Banco de Dados Relacionais (SGBD), que possui uma Linguagem Estruturada de Consultas (SQL) para realizar todos os comandos de manipulação necessários para manusear os dados (criação, exclusão e alteração de dados).

* + - 1. PostgreSQL

O PostgreSQL Database Server é um sistema de banco de dados relacional que pertence à classe de software de código aberto que se destaca por sua robustez. Existem versões disponíveis para uma grande variedade de sistemas operacionais. Ele possui mais de 15 anos de desenvolvimento ativo e uma arquitetura que ganhou uma forte reputação devido a sua estabilidade e integridade de dados.

* + 1. Versionamento de Software

O versionamento de software é um processo de controle de versões estabelecido por meio de numerações diferentes. Essa preocupação existe, pois geralmente mais de uma pessoa faz parte da composição do código. A alternativa que muitos utilizam é a de armazenamento na nuvem. É funcional, porém não é uma tática ágil. Pensando nisso, sistemas de controle de versão foram criados, onde é possível acompanhar todas as mudanças realizadas em ordem cronológica e detectar mesclas de códigos e conflitos, tudo de maneira automática.

* + - 1. GitHub

O GitHub é uma plataforma de gerenciamento de projetos e versões de código, permitindo a criação de um repositório remoto de um projeto poderá ser alterado, revisado, testado e gerenciado por equipes e pessoas externas.

* + 1. Scrum

O Scrum é uma metodologia de gerenciamento de projetos que surgiu no Manifesto Ágil e assim como outros frameworks da área, tem como objetivo desenvolver equipes para o execução de projetos de forma mais ágil. Essa metodologia ficou mais relevante no desenvolvimento de software sendo mais eficiente do que os antiquados métodos de cascata, por meio de ciclos curtos de entregas e alinhamentos diários entre os líderes a equipe de desenvolvimento.

Como apresentado por Jeff Sutherland, Cocriador da metologia, no livro “SCRUM A arte de fazer o dobro do trabalho na metado do tempo”:

“Tradicionalmente, a gerência quer duas coisas em qualquer projeto: controle e previsibilidade. O resultado disso é uma quantidade imensa de relatórios, gráficos e diagramas [...]. O Scrum questiona o porque tanto tempo e esforço são gastos na realização de uma tarefa, e porque somos tão ruins em prever o tempo e esforço que as atividades vão exigir. [...] o Scrum se baseia em uma ideia simples: quando começamos um projeto, porque não verificar a intervalos regulares se ele está indo no caminho certo e se isso é de fato o que as pessoas querem?.”

Além do controle de tempo e previsibilidade, o foco aqui são os intervalos regulares no caminho do projeto, denominados Sprints. A essência do Scrum é a realização de cerimonias que agilizam a realização de alinhamentos e tarefas e a atribuição de alguns papéis da equipe dentro do ciclo da Sprint.

Os papéis são: Product Owner, responsável por estabelecer a interação com os stakeholders, estando por dentro de todos os detalhes e priorizando as tarefas que serão realizadas na Sprint; Scrum Master, responsável por coordenar as cerimonias, resolver todos os impedimentos que surgirem da equipe de desenvolvimento e extrair métricas da Sprint; Dev Team, todo o restante da equipe que realiza a tarefas da Sprint.

As cerimonias são: Sprint Planning, reuniões realizadas no início do ciclo, onde são definidas todas as tarefas a serem realizadas no ciclo atual, levando cerca de 2 à 4 horas (lembrando que o correto gerenciamento de tempo significa disciplina). Daily Meeting, reuniões de 15 minutos no período da manhã entre toda a equipe para alinhamento das tarefas diárias e resolução de impedimentos, geralmente seguem o modelo de 3 perguntas entre cada integrante, são elas, “O que foi realizado no dia anterior?”, “Quais são os impedimentos enfrentados?” e “Quais serão as tarefas de hoje?”; Backlog Refinement, reuniões realizadas entre a equipe e o Product Owner para revisar as tarefas presentes no backlog para as próximas Sprints e mensurar a complexidade técnica de cada tarefa; Sprint Review, reuniões realizadas no fechamento do ciclo, também com o Product Owner que avaliará os resultados da Sprint, aprovando ou não a entrega efetiva da Sprint; Retrospective, realizadas após o fechamento do Sprint Review, onde a Dev Team analisa o desempenho da equipe na Sprint e abre uma rodada de Feedbacks com foco na melhoria contínua para os próximos ciclos.

Por fim, o Scrum não é um “elixir mágico” que tornará a equipe mais ágil em um curto período de tempo, mas sim, aumentará a previsibilidade e controle das entregas a cada ciclo, tornando equipes autogerenciáveis e aumentando o valor de cada entrega para os clientes, consequentemente, cada equipe alcançará um comportamento mais ágil à sua maneira.

1. **MATERIAIS E MÉTODOS**

Este capítulo irá apresentar sobre as metodologias, padrões de projetos, cronograma de atividades, tecnologias utilizadas no desenvolvimento do sistema.

* 1. METODOLOGIA SCRUM

A metodologia escolhida pela equipe para ser aplicada neste projeto foi a SCRUM, pelos fato dos membros da equipe conhecerem um pouco melhor, além de se sentirem mais confortáveis com a maneira como são divididas as tarefas e a frequência das reuniões.

Com o cronograma individual dos membros da equipe muito disperço, e a pandemia impossibilitando reuniões presenciais, as reuniões ocorreram 1(uma) vez ao dia, após o horário de trabalho, ocorrendo normalmente entre 19:00 e 00:00, utilizando a plataforma de reuniões Discord. Todo e qualquer assunto referente ao Trabalho de Conclusão de Curso foi resolvido diretamente pelas plataformas de comunicação online.

Nas reuniões de Sprint, foram aplicadas as perguntas chave (“O que voce fez ?”, “Há algum obstáculo na atividade que você irá realizar hoje ?”), ocasionando na solução dos problemas no dia ou antes da proxima reunião ser realizada.

* 1. VISUAL STUDIO COMMUNITY 2017

O Visual Studio foi a ferramenta utilizada para desenvolvimento do código que é uma IDE que auxilia no desenvolvimento de várias linguagens de programação, sendo a escolhida para o sistema C# utilizando .NET framework.

* 1. GITHUB

O GitHub foi utilizado para o versionamento do projeto, recebendo o código do sistema após o mesmo receber o aval de pelo menos 3 membros. Caso o código esteja funcional e aprovado, ele é nomeado como versão oficial. Se algum membro necessite resgatar alguma funcionalidade excluída anteriormente, basta acessar o projeto pelo site do GitHub e rastrear a versão desejada.

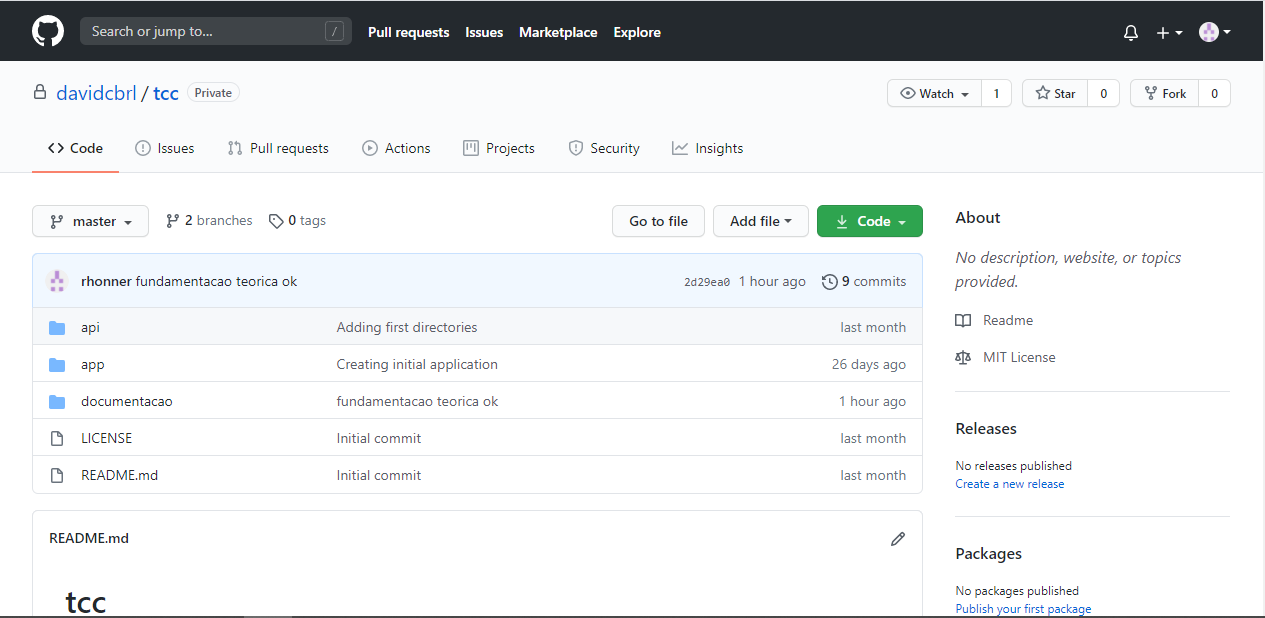


FIGURA X - UTILIZAÇÃO DA FERRAMENTA GITHUB

* 1. INFRAESTRUTURA DE DESENVOLVIMENTO

Para realizar o desenvolvimento deste projeto foram utilizados quatro

notebooks com as seguintes especificações:

* Notebook 1
* Proprietário: Rhonner Matheus Nunes Afanaci
* Fabricante: Acer
* Sistema Operacional: Windows 10
* Memória RAM: 8GB
* Processador: Intel Core i7 4ªGeração CPU 2.00GHz 2.60GHz
* Espaço de Armazenamento: SSD 256GB
* Mouse: Spider 2 Om-705
* Headphone: Sony MDR-ZX110
* Teclado: Naja Capricorn TCM-01N Switch Vermelho
* Monitor: Philco 24 Polegadas
* Internet: OneTech 200MB
* Notebook 2
* Proprietário: Daniel de Sá Besen
* Fabricante: Dell
* Sistema Operacional: Windows 10
* Memória RAM: 8GB
* Processador: G3 i9 8ªGeração
* Placa de Vídeo: GTX 1650
* Espaço de Armazenamento: SSD 512GB
* Mouse: Razer Deathadder
* Headphone: HeadSet HyperX Cloud
* Teclado: Fallen Ace
* Internet: Copel 300MB
* Notebook 3
* Proprietário: David Cabral Pereira
* Fabricante: Lenovo
* Sistema Operacional: Windows 10
* Memória RAM: 8GB
* Processador: i5 8ªGeração
* Espaço de Armazenamento: HDD 1TB
* Mouse: Logitech G305
* Headphone: HeadSet HyperX Cloud Stinger
* Teclado: Microsoft Keyboard 600
* Monitor: LG 24M38H-B
* Internet: Vivo 15MB
* Notebook 4
* Proprietário: Lucas Moroski
* Fabricante: Acer
* Sistema Operacional: Windows 10
* Memória RAM: 8GB
* Processador: i5 7ªGeração
* Placa de Vídeo: Geforce 940mx
* Espaço de Armazenamento: HDD 1TB
* Mouse: knup g14
* Headphone: Headset Husky 7.1 storm
* Teclado: Notebook
* Internet: NET/Claro 120MB
  1. RESPONSABILIDADES

|  |  |  |
| --- | --- | --- |
| **Número** | **Integrante** | **Responsabilidade** |
| 1 | Daniel de Sá Besen | Documentação do Projeto  Desenvolvimento Back-end  Modelagem Banco de Dados |
| 2 | David Cabral Pereira | Desenvolvimento Front-end  Prototipação das Telas  Desenvolvimento Mobile |
| 3 | Lucas Moroski | Desenvolvimento Front-end  Prototipação das Telas  Diagramação UML |
| 4 | Rhonner Matheus Nunes Afanaci | Documentação do Projeto  Desenvolvimento Back-end  Modelagem Banco de Dados |

QUADRO X - DEFINIÇÃO DE RESPONSABILIDADES

* 1. PLANO DE RISCOS

O plano de riscos tem como função considerar os riscos, possibilidades, impactos e soluções, visando sempre evitar e contornar possíveis imprevistos.

O QUADRO X mostra o plano de riscos deste projeto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Descrição** | **Solução para o risco** | **Probabilidade do risco** | **Impacto** |
| 1 | Falta de Tempo para desenvolvimento do projeto | Reorganização e divisão das atividades, aproveitando do cronograma pessoal e profissional dos membros | Alta | Grave |
| 2 | Mudança nos requisitos do projeto | Reunião para decisão se é possível adaptação e reaproveitamento de algo já criado.  Reorganização e divisão das atividades | Baixa | Moderado |
| 3 | Falta de conhecimento da tecnologia utilizada | Reunião com todos os membros para decisão de mudança ou reorganização das atividades | Moderada | Grave |
| 4 | Problemas na comunicação entre os membros | Supervisionamento comunitário entre os membros diante das atividades divididas, em reuniões semanais. | Baixa | Baixa |
| 5 | Tempo estimado muito baixo para cada atividade | Divisão da equipe em 2, aumentando a carga de esforço de cada membro, priorizando as atividades com prazo extrapolado | Moderado | Grave |
| 6 | Desistência de algum dos membros | Reorganização da equipe e atividades com todas as possibilidades em aberto, deixando em aberto a tecnologia utilizada | Baixa | Grave |

QUADRO X - PLANO DE RISCOS

* 1. AMBIENTE DE DESENVOLVIMENTO ANGULAR

Como dito anteriormente o Angular é um framework baseado em JavaScript, portanto devemos preparar o ambiente para suporte à essa linguagem e instalar as dependências para executar a aplicação.

* NodeJS:

Utilizando o sistema operacional windows, basta acessar o endereço de instalação do NodeJS: <https://nodejs.org/pt-br/>, realizar o download do arquivo com extensão .msi e seguir as orientações do instalador. Com isso, serão instalados o NodeJS juntamente com o gerenciador de pacotes npm (Node Package Manager) que utilizaremos para instalação do Angular.

* Angular:

Após a instalação do npm, utilizaremos o seguinte comando para instalação do Angular:

npm install -g @angular/cli

Considerando o projeto existente na máquina e presente no diretório, a aplicação possui o script customizado do npm para executar a aplicação, portanto, basta executar o seguinte comando:

npm run dev

* + 1. Visual Studio Code

Para desenvolvimento da aplicação web em Angular, foi utilizado o editor de código Visual Studio Code, juntamente com as extensões: Angular Language Service, GitLens, HTML Snippets, Editor Config for VSCode, Debugger for Chrome, ESLint, Material Icon Theme e TabNine.

1. **APRESENTAÇÃO DO SISTEMA FIDELITI**
   1. ARQUITETURA DO SISTEMA
   2. PRIMEIRO ACESSO
   3. USUÁRIO COMUM
   4. ADMINISTRADOR
2. **CONSIDERAÇÕES FINAIS**

**REFERÊNCIAS**

BRASILTURIS. **O segredo por trás dos programas de fidelidade**. Disponível em: https://brasilturis.com.br/o-segredo-por-tras-dos-programas-de-fidelidade/. Acesso em: 6 out. 2020.

CAELUM. **Um pouco sobre a história do C# e .Net**. Disponível em: https://www.caelum.com.br/apostila-csharp-orientacao-objetos/o-que-e-c-e-net#um-pouco-sobre-a-historia-do-c-e-net. Acesso em: 26 set. 2020.

GAEA. **Entenda-por-que-versionamento-de-software-e-tao-importante**. Disponível em: https://gaea.com.br/entenda-por-que-versionamento-de-software-e-tao-importante/. Acesso em: 30 set. 2020.

HOSTINGER. **O Que é GitHub e Para Que é Usado?**. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-github/. Acesso em: 30 set. 2020.

INFOESCOLA. **SQL Server**. Disponível em: https://www.infoescola.com/informatica/sql-server/. Acesso em: 26 set. 2020.

OGLOBO. **Mais de 700 mil empresas que fecharam as portas não vão reabrir após fim da pandemia**. Disponível em: https://oglobo.globo.com/economia/mais-de-700-mil-empresas-que-fecharam-as-portas-nao-vao-reabrir-apos-fim-da-pandemia-24535458. Acesso em: 30 set. 2020.

ROCKCONTENT. **Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa**. Disponível em: https://rockcontent.com/br/blog/banco-de-dados/. Acesso em: 26 set. 2020.

ROCKCONTENT. **O que é uma linguagem de programação e quais os tipos existem?**. Disponível em: https://rockcontent.com/br/blog/linguagem-de-programacao/. Acesso em: 26 set. 2020.

WOLIVEIRAS. **Introdução-a-versionamento-de-código-e-conhecendo-o-git**. Disponível em: https://woliveiras.com.br/posts/introdu%C3%A7%C3%A3o-a-versionamento-de-c%C3%B3digo-e-conhecendo-o-git/. Acesso em: 30 set. 2020.

# DRIBBLE. **UI Style Guidelines.** Disponível em: https://dribbble.com/shots/11627245-UI-Style-Guidelines

# DRIBBLE. **Setproduct.** Disponível em: https://dribbble.com/shots/10490036-E-Learning-UI-Exploration

MEDIUM. **12 Dicas de UX Para Criar Formulários Eficientes.** Disponível em:

https://medium.com/@allanroubertie/12-dicas-de-ux-para-criar-formul%C3%A1rios-eficientes-6fc621a3ea05

BLOG GAZIN ATACADO. **Tipos de programas de fidelidade e como escolher o melhor modelo.** Disponível em:

https://blog.gazinatacado.com.br/tipos-de-programas-de-fidelidade/

THATTO. **10 tipos de promoções para atrair clientes ao seu PDV.** Disponível em:

https://thatto.com.br/blog/pdv/10-tipos-de-promocoes-para-atrair-clientes-ao-seu-pdv/

MARKETING PARA INDÚSTRIA. **Promoção de vendas exemplos: conheça as melhores dicas do mercado.** Disponível em:

https://www.marketingparaindustria.com.br/gestao-vendas/estrategia-vendas/promocao-de-vendas-exemplos/

YOUTUBE. Tutorial Completo de FIGMA - **Ferramenta GRÁTIS para Design de Interfaces.** Disponível em:

https://www.youtube.com/watch?v=vg-INqhKD5c

JAVASCRIPT. **JavaScript.** Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript

NODE.JS. **NodeJS is a JavaScript runtime built on Chrome’s V8 JavaScript engine.** Disponível em: https://nodejs.org/en/

TYPESCRIPT. **Typed JavaScript at Any Scale.** Disponível em: https://www.typescriptlang.org/

ANGULAR. **Angular Docs.** Disponível em: https://angular.io/docs.

ANGULAR MATERIAL. **Components.** Disponível em: https://material.angular.io/components/categories

MATERIAL DESIGN. **Icons.** Disponível em: https://material.io/resources/icons

MEDIUM. **Introdução ao Angular HttpInterceptor.** Disponível em:

https://medium.com/matheus-bizutti/introdu%C3%A7%C3%A3o-ao-angular-httpinterceptor-1b06a95a0089