

Revista

Interação

ISSN 1981-2183



CONGRESSO ACADÊMICO DE TECNOLOGIA DA INFORMAÇÃO

V CONGRESSO ACADÊMICO DE TECNOLOGIA DA INFORMAÇÃO DA FAM — CATI



Revista Interação | v. 17, n. 1, 2024 | ISSN 1981-2183



Revista Interação | v. 17, n. 1, 2024 | ISSN 1981-2183

**CENTRO UNIVERSITÁRIO DAS AMÉRICAS – FAM
CURSOS DE TECNOLOGIA DA INFORMAÇÃO**

**V CONGRESSO ACADÊMICO DE TECNOLOGIA DA INFORMAÇÃO
DA FAM – CATI**

CURSOS DE TECNOLOGIA DA INFORMAÇÃO

Data do evento: 21 a 24 de novembro de 2023.

REITORA

Dr^a. Leila Mejdalani Pereira

PRÓ-REITOR

Prof. Dr. Luís Antônio Baffile Leoni

COORDENAÇÃO GERAL DOS CURSOS PRESENCIAIS

Profa. Ms. Camila Lopes Vaiano

COORDENAÇÃO GERAL DOS CURSOS EAD

Prof. Ms. Osorio Moreira Couto Junior

COMISSÃO ORGANIZADORA E CURADORIA

Prof. Ms. Claudiney Sanches Júnior

Prof^a Ms. Viviane de Oliveira Souza Gerardi

Prof^o Dr. Wagner Varalda

Prof. Ms. Leonardo Reitano

COMISSÃO CIENTÍFICA DO EVENTO

Prof. Ms. Adriano Arrivabene

Prof. Ms. Claudiney Sanches Júnior

Profa. Ms. Camila de Oliveira Ghendev

Prof. Ms. Eliane Cristina Amaral

Prof. Dr. Eugênio Akihiro Nassu

Prof. Dr. Fernando Trevisan Saez Parra

Prof. Dr. Fábio Vieira do Amaral

Prof. Ms. Gevanildo Batista dos Santos

Prof. Dr. Giocondo Marino Gallotti

Prof. Ms. Gregório Perez Peiro

Prof. Ms. João Carlos Silva de Souza

Prof. Ms. José Picovsky

Prof. Dr. Juliano Schimiguel

Profa. Lucia Contente Mós

Profa. Ms Luciana Akemi Nakabayachi

Prof. Ms. Leonardo Reitano

Prof. Ms. Marcelo Falco

Prof. Ms. Marcus Grilo

Prof. MS. Marcos Paulo de Souza Silva

Prof. Ms. Ranieri Marinho de Souza

Profa. Ms. Viviane de Oliveira Souza Gerardi

Prof. Dr. Wagner Varalda

EDITOR CHEFE

Prof. Dr. André Rinaldi Fukushima

EDIÇÃO DOS ANAIS

Prof. Dr. André Rinaldi Fukushima

Prof. Ms. Claudiney de Sanches Júnior

DIVULGAÇÃO

Agência Panda

LOCAL DO EVENTO E REALIZAÇÃO

Centro Universitário da Américas – FAM

Rua Borges de Figueiredo, 510, Mooca, São Paulo - SP

OBSERVAÇÃO – TODOS OS CONTEÚDOS DOS TRABALHOS

DESENVOLVIDOS E APRESENTADOS SÃO DE RESPONSABILIDADE DOS AUTORES

SISTEMA GERENCIADOR DE LANCHONETE: LANCHE ON NET

Edmilson Barcauscas Júnior (1), Felipe Victor da Silva (2), Marcus Vinícius da Silva Souza (3), Matheus Marano de Vasconcelos (4), Murilo Manoel Mello (5), Orientadora: Prof. Me. Eliane Cristina Amaral. (1) 7-SI-215708, (2) 7-SI-227648, (3) 7-SI-223230, (4) 7-SI-219013, (5) 7-SI-167325.

RESUMO

Com o avanço tecnológico, a utilização de sistemas para gerenciamento de negócios se tornou cada vez mais comum e necessário para o sucesso no mercado competitivo. Porém, muitos pequenos e médios empresários enfrentam desafios para investir em sistemas devido à falta de recursos financeiros ou conhecimento técnico. Diante desse cenário, o sistema desenvolvido apresenta uma solução web para os empreendedores que buscam um sistema gerenciador de pedidos, destinado a lanchonetes ou restaurantes, visando contribuir para o sucesso das pequenas empresas. O sistema foi desenvolvido com as tecnologias mais recentes do mercado como *NodeJS* e *MongoDB*, visando a flexibilidade e praticidade do sistema. Uma das análises realizadas apontou que pequenos estabelecimentos enfrentam dificuldades para obter resultados significativos, já que eles não têm tecnologia inserida em seu ambiente de trabalho, portanto a utilização de um sistema de gerenciamento pode fazer a diferença. Desta forma, a criação deste sistema visa atender às necessidades dos pequenos empreendedores, facilitando o processo de gerenciamento de pedidos e contribuindo para a expansão dos seus negócios. Este trabalho pretende apresentar o processo de desenvolvimento do sistema e a viabilidade de sua aplicação no mercado, permitindo aos pequenos empreendedores melhorarem suas atividades no mercado competitivo. O resultado foi benéfico, onde as pesquisas apontam uma boa probabilidade de efetivar o sistema no mercado.

Palavras-Chave: Sistema gerenciador de lanchonetes; *NoSQL*; App lanche on net

1. INTRODUÇÃO

Com o avanço tecnológico e a acessibilidade das plataformas para pequenos negócios, foi identificado uma oportunidade para desenvolver um sistema que apoia pequenas lanchonetes, restaurantes e bares de menor porte de modo a impulsionar suas operações. Ao incorporar a tecnologia em seu ambiente, esses estabelecimentos podem aprimorar a experiência de seus clientes durante o consumo de seus produtos e serviços. Uma Pesquisa do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) indica que empresas que investem em tecnologia têm maior probabilidade de obter resultados sólidos e um retorno mais expressivo sobre o capital investido [1]. Porém, é importante enfatizar que, geralmente, são as empresas maiores e consequentemente mais estabelecidas no mercado com maior capital para investir em tecnologia. O público-alvo deste sistema são empresas de pequeno porte, porém, sólidas em seus serviços, mas que enfrentam dificuldades para aumentar seus lucros. Outro estudo realizado pelo SEBRAE aponta que pequenas empresas não conseguem crescer devido à falta de conhecimento em tecnologia e a falta de capital para investir em *softwares*, isso porque os sistemas são oferecidos a um público-alvo diferente, com maior patrimônio, portanto, a maioria dos pequenos empreendedores são esquecidos [2].

Além disso, os pequenos empreendedores têm grandes participações no mercado de trabalho, onde aproximadamente 6,4 milhões de estabelecimentos no Brasil, são pequenos empreendedores [3], sendo que nesse número, 33% são do setor comercial, portanto estão inclusos aproximadamente 1 milhão de pequenos empresários [4], essas informações são relevantes onde apresentam que o sistema desenvolvido

tem uma grande margem em público-alvo. Portanto, é possível compreender que um *software* simples e acessível financeiramente pode ser uma solução viável para essas empresas.

Diante dessas informações, foi esclarecida uma boa oportunidade para inserir esse sistema no mercado para auxiliar esses pequenos empreendedores. No entanto, foi essencial determinar o método de desenvolvimento mais viável para implementar o sistema nos estabelecimentos, já que muitos deles podem não ter os recursos tecnológicos necessários para a efetivação do *software*.

Após uma análise cuidadosa, foi identificado que a criação de um sistema, a aplicação web é a abordagem mais pertinente, isso porque a praticidade de usar *Application Programming Interface* (API) para integrar recursos online podem auxiliar de maneira significativa, e também é possível inserir a implementação da responsividade, ou seja, capacidade do *software* de se adaptar a diferentes tamanhos de telas, como tablets e *smartphones* em geral, portanto, o ambiente web utilizando tecnologias atuais como *JavaScript*, *NodeJs*, *MongoDB*, e *VueJs*, pode ser uma ótima alternativa para os empreendedores que buscam uma solução confiável, prática e atual no mercado, com essas tecnologias foram usadas estratégias para aumentar a praticidade da codificação com metodologia ágil. Todas essas informações relacionadas as tecnologias e metodologias serão apresentadas nos tópicos a seguir.

O principal objetivo do sistema desenvolvido é agilizar os processos, como a solicitação e organização dos pedidos, com isso é possível minimizar a perda de tempo. Além disso, o sistema também visa mitigar o uso de papel, onde o cliente pode ser o responsável por realizar seus pedidos, se o mesmo não ter dificuldades com a manipulação de tecnologia, se o cliente não tiver facilidade com o sistema o garçom pode realizar os pedidos para ele.

2. MATERIAL E MÉTODO (OU METODOLOGIA)

De acordo com Mascarenha (2018) [5] pesquisa descritiva, tem o objetivo principal foi avaliar as características em uma população definida, e por meio dos resultados identificar se há relações entre elas, com isso em mente, foi utilizado a base desse método, onde envolve técnicas de entrevistas e avaliações por meio de questionários, esse tipo de análise é relacionada ao sistema, já que é necessário observar a padronização de resultados nas pequenas empresas, a fim de obter uma vantagem competitiva no mercado. Junto a isso, também foi desenvolvida a análise de precificação, onde em global os custos para os clientes e os lucros da equipe.

Foi necessário elaborarmos a Gestão da Análise de Riscos, onde será registrado toda a base do que pode influenciar negativamente a evolução e o desenvolvimento do *software*, a análise de risco foi desenvolvida em uma planilha, onde consta algumas possibilidades de ameaças, e suas respectivas mitigações e contingências. A planilha desenvolvida mostra que aproximadamente 33% dos riscos são de gravidade “média” enquanto os 67% são de gravidade “baixa”, portanto, o sistema está conforme os riscos esperados, resultando em um sistema viável [6].

Sendo análise geral, foi inserida a técnica de gerenciamento da equipe, ou seja, qual metodologia será aplicada para o desenvolvimento do *software*, a técnica utilizada foi a “*SCRUM*” onde, o time se gerencia de maneira eficaz com metas e “*sprints*” definidos. Os *Sprints* são reuniões regulares, onde cada integrante da equipe apresenta o que fez ou o que irá fazer, tornando-a equipe consciente da evolução do sistema e melhorando a qualidade do sistema [7]. Junto à metodologia foi pensado na técnica de codificação do sistema, ou seja, como o sistema será desenvolvido e melhorado, foi entendido que para esse sistema o modelo “Evolutivo” é o mais apropriado. Isso porque o modelo evolutivo se adequa a flexibilidade e mu-

danças que podem ocorrer, visto que, cada cliente pode solicitar algumas mudanças de interface no *software*. Segundo Pressman (2016) os *softwares*, ao longo do desenvolvimento, vão ficando cada vez mais complexos, com isso os requisitos do sistema vão evoluindo, logo, o prazo se torna mais extenso, portanto, se torna inadequado utilizar um planejamento em linha reta [8].

Realizando as definições, foi aplicado as ferramentas para a codificação do sistema, onde pensando em um sistema *web* foi definido as ferramentas, Linguagem de Marcação de Hipertexto (*HTML*), Folhas de Estilo em Cascata (*CSS*), linguagem de programação *JavaScript*, *framework Node js Express*, *Framework Vue.js*, *MongoDB*, *mongoose* *GitHub*. Essas ferramentas foram definidas porque o ambiente *web* é quase totalmente utilizado as linguagens-base *web*, em vez de outras linguagens como *Java* ou *Python*. Basicamente o *HTML* e *CSS* são o conteúdo e estilo das páginas, onde apresenta a “essência” dos sites e plataformas, o *JavaScript*, *Nodejs* e *Vue* fazem as interações e funções do site, ou seja, tudo que os usuários processam, o *MongoDB* e o *Mongoose*, são o *back-end* da aplicação, portanto são todos os dados salvos da aplicação para o banco de dados, onde podem ser manipulados pelo *front-end*. *Front-end* é tudo que o cliente final visualiza, nesse caso são os clientes e funcionários dos estabelecimentos, enquanto o *back-end* são os dados salvos, e todas as interações não visíveis aos usuários.

Essas definições, foi verificado com os elementos principais, em competição de mercado com o auxílio da Matriz *Swot* (*Strenghts*, *Weaknesses*, *Opportunities*, *Threats*), onde foi possível observar os aspectos gerais do sistema, como força, fraqueza, oportunidade e ameaças com isso é possível implementar melhorias continuamente, buscando a satisfação do cliente. Com os dados expostos é possível observar os pontos positivos e negativos, onde cada ponto deve ser visto como uma melhoria, onde será possível evoluir o sistema e expandir ideias para implementação do sistema, sendo assim possível obter um sistema cada vez melhor [9]. Para descobrir o que os proprietários precisam e entender as diversas necessidades, a equipe está realizando uma pesquisa de campo, onde é possível entender melhor o público-alvo e ajustar suas demandas com o sistema. A pesquisa consiste em um questionário do *Google* onde os pequenos e grandes empresas respondam de acordo com sua visão, com isso é possível comparar as respostas, tanto de um cliente que não tem o sistema, quanto o cliente com uma plataforma semelhante, e com essa comparação deve ser possível de encontrar uma solução para determinada situação do pequeno empreendedor.

Também foi necessário desenvolver as tabelas de requisitos, onde descrevem os requisitos funcionais e não funcionais do sistema, as tabelas oferecem ênfase à documentação do sistema, descrevendo de forma objetiva as relações detalhadas do sistema em geral, com isso é possível ter embasamento antes da codificação do sistema [10]. Outra parte importante é entender os processos do sistema desenvolvido, para isso foi desenvolvido alguns diagramas como o diagrama de fluxo de processos, o diagrama *Business Process Model and Notation* (*BPMN*) e o diagrama de navegação. O diagrama de fluxo de processos, que é uma representação gráfica que mostra de forma geral o passo a passo das execuções dos processos do sistema, portanto, mostra como o sistema deve agir, desse modo fica mais simples os usuários entenderem o funcionamento do sistema. Esse diagrama utiliza figuras geométricas sendo que cada figura pode determinar uma ação ou um processo e todas as figuras são conectadas por setas, onde indica a direção que os processos ocorrem O diagrama *BPMN* mostra de forma detalhada as relações do sistema, é utilizado para modelar processos, portanto é mais detalhado do que o fluxo de processos, onde apresenta não só os processos como também eventos e possibilidades que podem ocorrer ao utilizar o sistema. Os processos estarem efetuando, foi desenvolvido o diagrama de navegação, esse diagrama apresenta as páginas do site, onde fornecem informações semelhantes a um navegador *web*, com isso é possível entender as bases importantes do sistema, além disso, é possível identificar possíveis erros na lógica do site, e com isso corrigir

a estruturação [11]. O diagrama de navegação, diferente dos diagramas de processos, não apresentam as funções em detalhes, mas sim detalhes das páginas dos sites de forma geral.

Para estruturar o sistema de forma visual, está sendo desenvolvidas algumas telas *Wireframe*, essas telas representam a base do layout, portanto pode ser considerado o esboço do sistema, onde apresentam os componentes da página, ou seja, o *design* com as localizações dos campos e botões, com essas telas é possível entender a estruturação do sistema, essas telas são usadas para ajudar o desenvolvedor programar os *softwares* e ajudam os usuários, como os clientes e os funcionários dos estabelecimentos a interpretar o *design* da plataforma com mais facilidade [12].

3. DESENVOLVIMENTO

Mediante a esse cenário, onde pequenos empresários podem utilizar um sistema *web* para gerenciar os pedidos dos clientes, pode ser analisado que diversos problemas podem ser solucionado com o sistema desenvolvido, problemas como gestão e praticidade, onde os processos como anotação de pedidos no papel podem ser substituídos com o uso tecnológico, deixando o processo geral mais prático e agradável, tanto para o cliente quanto para os funcionários. Com o embasamento apresentado na metodologia, foi possível seguir os passos para a criação do sistema.

Além disso, realizando algumas análises, foi possível observar que um Sistema Gerenciador *web* pode ser uma solução interessante para os donos de lanchonetes e restaurantes, já que a grande demanda pode se beneficiar com uma solução simples, amigável, com preço competitivo. O embasamento teórico e ferramentas para codificação definidas, o sistema continuou a ser desenvolvido, é importante ressaltar que esse sistema é uma evolução dos sistemas anteriores, portanto o sistema atual se mostra mais completo e eficaz, abrangendo novas melhorias. A continuação do sistema começou na reestruturação dos requisitos funcionais e não funcionais, onde é primordial ter as funcionalidades e a demanda do cliente. Observe a seguir as tabelas desenvolvidas

Tabela 1. Requisitos funcionais

RF Tela de Pedidos		Oculto ()
Descrição:	Tela inicial do sistema, onde os clientes ou garçons dos estabelecimentos irão solicitar os pedidos	
Requisitos funcionais		
Nome	Descrição	Categoria
RF1 Lanches disponíveis para à compra	Página inicial com todos os lanches disponíveis em cada estabelecimento	interface do usuário
RF2 Descrição dos lanches	“Modal” que apresenta a descrição do lanche selecionado, ou seja, os ingredientes do lanche	interface do usuário
RF3 Botão adicionar ao carrinho e input para inserir o nome do cliente	Após a verificação da descrição do lanche haverá um botão para o cliente adicionar o pedido no carrinho. Possibilitando o cliente adicionar acompanhamentos e a quantidade de lanches	interface do usuário/Processamento
RF4 Página de apagamento	Após a seleção e envio do pedido, o cliente poderá escolher o meio de pagamento	interface do usuário

RF5 Movimentação do pedido	Após o pedido ser realizado com sucesso, ele será transportado para a tela do funcionário, gerando um ID e o nome do cliente, onde poderá ser manipulado de acordo com a situação atual	Processamento
----------------------------	---	---------------

Fonte: Autores

Tabela 2. Requisitos funcionais

RF Tela de gerenciamento		Oculto (X)
Descrição:	Tela para os funcionários da empresa gerenciarem os lanches solicitados	
Requisitos funcionais		
Nome	Descrição	
RF6 Aba de login	A página inicial para os funcionários será uma tela onde os funcionários terão credenciais predefinidas e seguras para logar no sistema	interface do usuário/Segurança
RF7 Aba de gerenciamento	Após o login ser realizado os funcionários poderão visualizar os pedidos solicitados	interface do usuário
RF8 Manipulação do status dos pedidos	Ao visualizarem cada pedido com seu respectivo nome de cliente, será possível manipular o “status” de cada pedido, ou seja qual situação o pedido selecionado está. A manipulação será dividida em um botão com 3 possíveis opções, são elas: Solicitado, Em produção, Entregue	Processamento
RF9 Botão de logout	Após o trabalho dos funcionários forem finalizados, será possível fazer o logout no sistema, para que não comprometa a segurança	interface do usuário/Segurança

Fonte: Autores

Tabela 3. Requisitos não funcionais

RN Análise geral do sistema	Oculto ()	
Descrição:	Detalhamento do sistema	
Requisitos não funcionais		
Nome	Descrição	
RN1 Sistema web	O sistema será web, localizado em uma hospedagem, quando o cliente contratar o serviço, será disponibilizado o link contendo a página da plataforma, ou se o cliente preferir o sistema pode ser instalado localmente na máquina	Usabilidade
RN2 Utilizará JavaScript, Node js, Vue.js, MongoDB , mongoose	Tecnologias atuais e muito utilizadas no ramo web	implementação
RN3 Executável em qualquer plataforma	O site deve funcionar em ambiente Windows, Linux e Mac	Portabilidade

Fonte: Autores

Tabela 4. Requisitos não funcionais

RN4 Tela de pedido	Oculto ()	
Descrição:	Tela inicial do sistema, onde os clientes ou garçons dos estabelecimentos irão solicitar os pedidos	
Requisitos não funcionais		
Nome	Descrição	Categoria
RN5 Design simples	Design de fácil visualização para os usuários não terem dificuldade de utilizar a plataforma	Usabilidade
RN6 Responsividade	Site deve ser adaptável para vários tamanhos de dispositivos, seja tela de notebooks ou celulares	Usabilidade
RN7 Praticidade	O site deve responder rapidamente ao cadastro de novos pedidos para eles serem preparados no momento certo	Desempenho

Fonte: Autores

Tabela 5. Requisitos não funcionais

RN Tela de processamento		Oculto (X)
Descrição:	Após a inserção dos dados do cliente, as informações serão transferidas em um banco de dados, com isso os pedidos serão processados, finalizando o pedido, o garçom irá levar à mesa ou será solicitado a retirada.	
Requisitos não funcionais		
Nome	Descrição	Categoria
RN8 Tela simples	A tela onde o funcionário poderá acompanhar o pedido dos clientes	Usabilidade
RN9 Responsividade	Adaptável em dispositivos seja notebooks, tablets ou smartphones	Usabilidade

Fonte: Autores

O sistema apresenta dois fluxos de utilização, um dos clientes da hamburgueria e outro para os funcionários/administradores, cada fluxo possui uma interface diferente com funções diferentes para a utilização. No fluxo dos clientes eles podem acessar o cardápio e selecionar os produtos de duas maneiras, por meio de um *QRcode* exibido na mesa ou em algum local da hamburgueria, esse *QRcode* redireciona o cliente para a página do cardápio onde ele pode navegar e visualizar as informações dos produtos ou navegar no site de cardápio onde também apresentará as informações como nome, descrição, preço e uma imagem do hambúrguer e selecionar os produtos desejados, esses produtos selecionados vão para um carrinho de compras onde o cliente pode validar se selecionou corretamente o que queria e alterar a quantidade dos produtos selecionados e visualizar o valor total do pedido, assim que finalizado a etapa de seleção de produtos o cliente entra no processo de *checkout* informando o nome para a retirada do pedido e escolhem a forma de pagamento feito mediante uma *API* externa, após a forma de pagamento ter sido devidamente

escolhida e o pagamento ter sido efetuado, o pedido é confirmado e o cliente pode aguardar o pedido ficar pronto para a retirada que, dependendo da empresa pode ter tanto a opção para retirada quanto a refeição no local.

O fluxo do funcionário/administração consiste no gerenciamento dos pedidos, podendo alterar o *status* de cada pedido feito pelo fluxo dos clientes com o intuito de mostrar que o pedido foi aceito, esta sendo preparado ou está pronto para a retirada, os funcionários também, tem uma interface para visualizar todos os pedidos e seus detalhes, como a quantidade de cada produto que o cliente solicitou e a descrição para poder preparar o pedido, além do nome do cliente para poder chamá-lo quando estiver pronto, no fluxo de funcionário/administração também é possível adicionar, excluir ou alterar os produtos da loja também, impactando também na visualização dos clientes.

Esses dois fluxos foram desenvolvidos com base em um banco de dados *MongoDB*, uma *API* feita em *Nodejs* e o *back-end* para cada fluxo também em *Nodejs*, o *front-end* foi desenvolvido utilizando *HTML*, *CSS* e *JavaScript* com o *framework* *VueJs*, os produtos são armazenados em tabelas diferentes, os hambúrgueres, bebidas e porções, onde a *API* verifica os dados contidos no BD e envia por meio de rotas *Hypertext Transfer Protocol (HTTP)* para o *back-end* dos dois fluxos, o *back-end* do fluxo de funcionarios feito em *NodeJs* processa os dados recebidos da *API* criando, lendo, alterando e deletando os produtos (*CRUD*), disponibilizando rotas *HTTP* para cada operação onde o *front-end* consome essas funções e exibe para os funcionários usarem através da interface, pelo fluxo do cliente, outro *back-end* utiliza os dados tratados pelo fluxo dos funcionários para exibir cada produto e seus atributos corretamente, fazendo a chamada das funções com base nos métodos do *back-end* da aplicação.

Com os detalhes do sistema predefinidos, foi iniciado a codificação, onde está sendo transferido do planejamento para a realidade. O ambiente de desenvolvimento integrado ou *integrated development environment(IDE)* escolhido foi o *Visual Studio Code*, pois é uma das *IDE'S* mais utilizadas nesse momento, essa *IDE* suporta diversas linguagens de programação e formatos de arquivos, entre elas, *HTML*, *CSS*, *JavaScript*, e todas as ferramentas necessárias para a codificação do sistema, além disso, essa *IDE* conta com vários recursos chamado "extensões", onde é possível adicionar recursos de fora da aplicação desenvolvida para auxiliar os programadores, portanto essa *IDE* é a melhor opção para o desenvolvimento desse sistema [13].

A primeira parte do desenvolvimento do sistema foi o *front-end* e como mencionado anteriormente, o *front-end* é responsável pela apresentação do sistema para o usuário, portanto é tudo que o usuário visualiza e interage. No caso desse sistema toda a apresentação do sistema é injetado de um *framework* para a página *HTML*, nesse caso as páginas são injetadas do *framework* *Vue*. *Frameworks* são conjuntos de bibliotecas, que englobam funcionalidades e estruturas com o intuito de fornecer soluções para vários tipos de problemas, além de ser mais prático a codificação do sistema, é possível reutilizar o código, facilitando ainda mais e deixando o desenvolvimento padronizado [14]. Após a interação do *framework* com *HTML*, uma parte do conteúdo é apresentado para os usuários, é importante entender que o *framework* está conectado com uma *API* criada, que é uma aplicação mediadora entre o banco de dados (*back-end*) e apresentação do sistema na tela(*front-end*). Em geral, o sistema se comunica do banco de dados, para o *framework* *Node js*, depois para o *framework* *mongoose* e após traduzido para o *framework* *Vue.JS*, onde os componentes são injetados para a tela do usuário.

O sistema consiste em dois fluxos, requeridos pela *API*, essa *API* contém os dados de cardápio, onde apresenta os nomes, ID's, imagens e descrições dos hambúrgueres, além disso, apresenta as porções e

bebidas disponíveis. Essa *API* poderá ser modificada conforme a necessidade do cliente, ou seja, cada cliente terá um lanche específico com uma imagem e descrição, portanto, com a modificação dos dados da *API* será possível apresentar os lanches de cada estabelecimento com facilidade, sem a necessidade de alterar o código estrutural.

O *back-end* é composto pelo banco de dados *MongoDB* que é um banco de dados não relacional (*NoSQL*), portanto, diferente dos bancos de dados baseados em tabelas como *Oracle* e *MySQL*. Composto pelo *framework Node js Express*, que é um *framework* de *JavaScript*, e pelo *Mongoose*, que é uma biblioteca de Modelagem de Dados de Objeto (ODM) para *MongoDB* e *Node js*. O sistema está com banco de dados não relacional pela flexibilidade e velocidade do sistema, onde é possível manipular arquivos *JavaScript Object Notation (JSON)* e *Binary JSON (BSON)* que são *scripts* mais escaláveis. Esse é um ponto extremamente importante, pois, se o cliente (empresário/funcionário) precisar gerenciar grandes camadas de dados, com essa arquitetura o sistema mitigará travamentos e lentidões. Além disso, banco de dados não relacional está sendo implementados de maneira significativa no mercado, onde com a evolução tecnológica, é necessário utilizar outros meios para resolver determinados problemas, e isso é mais um ponto positivo para esse sistema, onde a “durabilidade” das tecnologias serão amplas [15]. Basicamente, o *Node js express* foi utilizado a criação da *API* e para facilitar as requisições *HTTP's* onde é necessário esse procedimento para o consumo da *API* criada, portanto ele é o *framework* que compila a *API* para o *mongoose*. O *Mongoose*, gerencia o relacionamento entre os dados da aplicação, portanto, ele fornece as validações e traduz os objetos no código e a representação dos objetos no *MongoDB*. Refletindo para o sistema, o *mongoose* traduz os tipos de arquivos requeridos da *API* pelo *Node js express*, arquivos salvos no *MongoDB*, e após a compilação são enviados ao *front-end* por meio protocolo *HTTP*, sendo assim, apresentados à tela dos usuários.

A comunicação dos fluxos funciona da seguinte forma, depois do *back-end* processado os componentes são injetados ao *front-end*, através das “rotas”, que são bibliotecas utilizado em *frameworks front-end*, as rotas, são os componentes que “explica” aos elementos o que fazerem. Nesse caso, quando os usuários fizerem alguma alteração, seja adicionar, modificar ou remover informações no pedido, será feita uma alteração no banco de dados e apresentará um respectivo componente para aquela ação. A diferença entre a tela do cliente e a tela do funcionário são as permissões, onde o cliente só poderá adicionar ou remover os próprios dados, enquanto os funcionários poderão visualizar e editar todas as informações de qualquer cliente, além disso, eles podem alterar o cardápio, realizando mudanças nos hambúrgueres, porções e bebidas, portanto ênfase na segurança e organização do sistema.

Paralelamente à codificação, foi realizada a análise de mercado que está englobada à pesquisa descritiva, portanto se baseiam em entrevistas com funcionários dos estabelecimentos, a fim de encontrar resultados para a implantação do sistema no mercado, as entrevistas são realizadas por meio de um questionário do *Google*, com a base das respostas é possível encontrar padronizações e com isso ter conhecimento sobre as informações obtidas. As pesquisas são direcionadas ao público de pequenos empreendedores, isso porque, com as informações dos pequenos empreendedores é possível compreender se o sistema será bem aceito no mercado, além de descobrir se o sistema realmente fará uma diferença significativa no negócio e se pode melhorar a produtividade dos processos. O questionário se compõe por 4 questões objetivas, onde o objetivo é ser preciso e obter as respostas sem perder muito tempo. O estudo foi realizado em pontos estratégicos e com as respostas obtidas será possível gerar gráficos que auxiliam as decisões e aplicabilidade do sistema, além de obter uma base argumentativa para justificar a implementação do sistema no mercado. Além disso, foi realizada a análise de precificação, ou seja, o estudo relacionado a quanto os

clientes do sistema vão investir e a quanto a equipe irá lucrar, essa pesquisa engloba o valor médio do mercado, portanto, foi verificado em *sítes* os valores gerais no mercado para ter noção dos gastos e lucros dos clientes e da equipe.

Um ponto importante a ser discutido é o modo de codificação, onde muitos programadores podem não utilizar técnica de codificação ou boas normas, para o desenvolvimento desse projeto está sendo aplicado a filosofia “*Clean Code*”, onde é imprescindível a utilização de código limpo. Essa filosofia visa as melhores práticas para a codificação em geral, não importando a linguagem de programação e qual sistema será criado, o importante é que o sistema tenha um código visualmente harmônico e seja eficiente [16]. O sistema desenvolvido, envolve pequenas funções, o mínimo possível de complexidade no código e uma fácil interpretação do sistema, todas essas bases são fundamentais para a resolução de futuros problemas, além disso, o *framework* utilizado, trabalha com a melhoria de semântica, onde todo o conteúdo é injetado após a compilação do código. A utilização de código limpo não é necessário para o sistema funcionar, mas sim para os desenvolvedores ter um entendimento melhor e poderem programar sem muitos problemas. A filosofia *clean code* pode levar a outro assunto importante, a qualidade de *software*, na metodologia desse artigo, foi apresentada as técnicas de codificação como modelo evolutivo e a base de requisitos funcionais e não funcionais, porém, não foi abordado que esses elementos são uma das bases da qualidade de *software*, aplicada nesse sistema.

A qualidade de *software* é basicamente a conformidade e fidelidade do *software* ao que foi proposto, portanto, está intimamente ligada aos requisitos funcionais e não funcionais [17]. Além disso, a qualidade de *software* está ligada aos testes de *software*, visando testar todas as funções do sistema, a fim de encontrar algum erro ou anomalia. Durante a codificação do sistema, foi necessário realizar diversos testes para encontrar possíveis erros e corrigi-los, como o sistema conta com novos recursos como a tela de login e mudança no fluxo de processos foram necessárias rever e modificar as funções, a fim de aplicar as melhorias com segurança e conformidade.

Com todas as alterações realizadas foi possível obter um sistema funcional, conforme a proposta apresentada, onde o sistema segue todos os aspectos dos requisitos funcionais e não funcionais, visando se adequar para as necessidades dos clientes. Também é possível analisar que todas as tecnologias utilizadas foram previamente pensadas a fim de obter um sistema moderno e coerente com o mercado.

Com base nas entrevistas, foi possível obter resultados positivos em relação à implementação do sistema no mercado, as entrevistas foram realizadas de forma presencial, onde foi solicitado os funcionários a responderem algumas perguntas em relação à implementação da tecnologia no estabelecimento, com os resultados foi possível criar gráficos que contribui para uma análise no mercado, todas as imagens desenvolvidas estão anexadas à bibliografia, isso para ter uma melhor visualização e compreensão das imagens, observe o gráfico anexado [18].

Essa observação está relacionada ao sistema e foi possível considerar se os clientes realmente gostariam ou precisariam ter um sistema para gerenciar seus pedidos, um sistema *web* com as sessões e podendo ser implementado em totem, isso porque não faria sentido criar um sistema sem utilização. Como é visível, 71% dos entrevistados gostariam de implementar essa tecnologia em seus estabelecimentos, portanto é um número positivo, vale ressaltar a relatividade entre a pesquisa, já que o lugar pesquisado pode sofrer resultados diferentes, como mencionado anteriormente, essa pesquisa foi estruturada estrategicamente, em locais com grande movimentação de empresas, portanto o cenário estudado é tecnológico e abrangente.

A segunda análise é referente se o cliente já implantou alguma tecnologia semelhante ao sistema no estabelecimento, a análise foi realizada em apenas 10 estabelecimentos, essa análise foi somente para ter um simples embasamento, portanto, o resultado pode variar conforme o local. Essa pesquisa resultou positivamente com 55% onde os clientes não têm tecnologias em seus estabelecimentos, portanto, é encontrado uma boa margem para incluir o sistema nesses locais, observe o gráfico anexado [19].

A pesquisa de precificação resultou que o gasto médio para o cliente com *software* é de R\$325,00, onde de acordo com SEBRAE [20] existem sistemas muito simples de baixo valor, a partir de R\$50,00 e sistemas mais complexos podendo chegar aos R\$600,00 o valor mensal para ativação do sistema, além disso, os sistemas podem não ter o auxílio técnico incluso, ou seja, pode ser necessário os clientes pagarem outro valor para ter suporte aos sistemas. Conhecendo essas informações é possível considerar a inclusão do preço médio para a efetivação do sistema, como a ideia do sistema desenvolvido é incluir um sistema bom com preço competitivo, os clientes que contratarem esse sistema terão o suporte técnico incluso com o sistema, portanto, além de minimizar a possibilidade de não contratação do sistema, deve ser possível obter a fidelidade do cliente e isso pode gerar vantagem competitiva em relação aos outros sistemas semelhantes.

Paralelamente foi realizado o estudo para verificar a implantação de *tablets* nos estabelecimentos, isso porque os clientes poderiam realizar os pedidos nos equipamentos dos estabelecimentos, facilitando as solicitações de pedidos, a análise foi realizada comparando 3 marcas, com valores e potências diferentes, resultando o valor mínimo de aproximadamente R\$400,00 para *tablets* simples, podendo chegar à R\$3.500,00 para *tablets* potentes [21], esse valor dependerá das funções dos *tablets*, entretanto, como o sistema é *web*, os empresários poderão implementar *tablets* de baixo valor, com funções básicas, e mesmo assim, o sistema será bem executado, isso porque o sistema não utiliza grande processamento ou grande utilização de memória. A possibilidade da implantação de *tablets* pode ser uma boa opção, porque certamente os empresários não terão recursos financeiros para a compra de totens para os clientes solicitarem os pedidos, já que o valor pode variar de R\$1.000,00 à R\$300.000,00 [22]. Portanto, é visível que a implantação de *tablets* pode ser uma boa alternativa, agregando valor à empresa, já que grandes empresários utilizam totens para a facilitação de seus processos, desse modo, mitigando os problemas com as demandas, além disso, os clientes podem eventualmente não conseguirem realizar seus pedidos, seja por problema de conexão ou algo relacionado ao equipamento pessoal, desse modo, será possível utilizar o equipamento da organização.

Em resultado, com a pesquisa descritiva, é possível notar que a chance de sucesso na implantação do *software* é relativamente alta, já que a maioria dos pequenos empresários não tem solução tecnológica, além de desejar ou necessitar implantar novas tecnologias, portanto com as técnicas de *marketing* corretas, será possível publicar a solução desenvolvida e tentar obter resultados positivos.

Iniciando a parte técnica do sistema, foi estruturado o desenvolvido de todos os elementos abordados na sessão metodologia onde foi possível obter resultados positivos em relação ao desenvolvimento e base estratégica do sistema, o primeiro ponto como mencionado foi a definição da tabela de matriz “SWOT”, onde apresenta as colocações, tanto positivos quanto negativos do sistema. Observe o resultado da definição da mesma

Tabela 6. Matriz SWOT

Forças	Fraquezas	Oportunidades	Ameaças
Sistema de baixa complexidade	Novo no mercado	Atualizações	Concorrência

Web	Sem engajamento	Personalizável	Má satisfação do cliente
Preço competitivo		Público-alvo extenso	

Fonte: Autores

Como é possível observar, que o sistema desenvolvido, tem capacidade de se expandir no mercado, isso porque o sistema é web e pode ser implementado em várias máquinas, além de ter um público-alvo abrangente, porém há pontos negativos, onde é importante a observação das mudanças no mercado, mesmo assim, fica claro a vantagem competitiva desse sistema no mercado.

A tabela matriz *SWOT*, apresenta somente informações gerais sobre o planejamento de negócio, porém é importante saber como o sistema funciona de fato e para entender os processos internos, foi desenvolvido o diagrama de fluxo de processos, como anunciado na metodologia, observe o anexo [23] contendo o diagrama de fluxos de processos

Como é possível visualizar, o diagrama consiste em formas geométricas, onde os quadrados significam as ações efetuadas, seja manuais ou ações do sistema, os losangos significam as verificações para a tomada de decisão, portanto quando for verificada uma certa ação, o fluxo de processo poderá mudar, e por fim o quadrilátero irregular representa as ações de alterações, ou seja, quando o cliente ou funcionário fizer alguma alteração no sistema. O processo do sistema é simples, onde o cliente ou o funcionário registra as informações no sistema, e o pedido é adicionado ao carrinho onde será possível manipular a quantidade de hambúrgueres, bebidas e porções, e após a confirmação, o sistema processará o pedido, gerando um *ID*, após isso, será apresentado uma tela de pagamento, onde o cliente deve inserir suas informações para o pedido ser processado para à tela do funcionário, onde poderá manipular as informações de cada pedido.

Também é importante entender as funcionalidades do sistema de outra perspectiva, onde o sistema poderá sofrer alterações, eventuais erros ou atualizações, pensando nisso foi desenvolvido o diagrama *BPMN*, onde apresenta uma visão gerencial dos processos, envolvendo a equipe desenvolvedora e os usuários do sistema, observe o anexo [24] com diagrama desenvolvido.

Portanto, o diagrama *BPMN* é composto por retângulos, quadrados e losangos com diferentes símbolos, onde os retângulos representam as raias, que podem ser grupos ou elementos que realizam as determinadas ações, nesse caso, são representados como a equipe desenvolvedora, os usuários do sistema e os clientes dos estabelecimentos. Os quadrados são as tarefas realizadas pelos grupos, é possível visualizar que algumas dessas tarefas/ações contem pequenos símbolos na parte superior, esses símbolos representam os modelos das atividades, os símbolos que contem a imagem de uma mão representa que as atividades são manuais, portanto o sistema não processa, as atividades com o símbolo de perfil são atividades que os usuários realizam. Os losangos com símbolos de cartas representam envio e recebimento de e-mails, e o losango com um pentágono representam os eventos, nesse caso são os possíveis e eventuais erros que podem ocorrer ao longo da utilização do sistema. De forma geral os clientes e funcionários utilizam o sistema desenvolvido e entram em contato com a equipe de desenvolvimento para solucionar possíveis erros ou até realizar mudanças no sistema, enquanto os clientes dos estabelecimentos utilizam o sistema somente para realizarem os pedidos.

Diante desse cenário, é importante embasar como será a estruturação do sistema, onde é importante documentar as páginas disponíveis, pensando nisso foi desenvolvido o diagrama de navegação, onde apre-

senta todo o contexto do site, como as páginas, conteúdos e as relações entre si, esse diagrama é utilizado para enfatizar o entendimento do sistema e a documentação desenvolvida, observe o anexo [25].

Nota-se que no sistema há três principais páginas, onde apresenta os produtos disponíveis como, os hambúrgueres, bebidas e porções. Nessas páginas todo o conteúdo é apresentado por esses elementos, onde mostra como “Hambúrguer 1”, todos os conteúdos podem ser modificados conforme a necessidade do cliente. No sistema esses conteúdos serão apresentados em “cards” que são estruturas em “divs” do HTML, essas estruturas foram definidas, pois podem ser dinâmicas e modificadas com facilidade, garantindo assim a produtividade em alterações no sistema.

Após os pedidos serem selecionados eles irão ser adicionados na página de carrinho, onde poderá ser visualizado o pedido completo e poderá ser modificado a quantidade ou excluído algum item, após essa etapa o usuário será levado à tela de pagamento onde será processado a forma de pagamento e após o pagamento ser confirmado o pedido será processado para a tela do funcionário, onde poderá ser modificado a situação de cada pedido. Também há a tela de login do funcionário, que será conectada à tela dos pedidos realizados.

Finalizando as noções de processos, foi desenvolvido diagramas de *wireframe*, onde apresenta o protótipo do projeto finalizado, apresenta o layout desenvolvido com a respectiva modelagem das localizações de botões e a estruturação completa, observe o anexo [26] onde apresenta o diagrama *wireframe* da tela inicial do sistema.

Como é possível observar, o sistema terá duas barras laterais, onde a barra esquerda será a localização do menu apresentado os produtos disponíveis, portanto, os hambúrgueres, as bebidas e as porções, e quando o botão for clicado será mostrado todos os respectivos produtos, portanto quando o usuário clicar no botão “bebidas” a pagina rolará automaticamente aos elementos disponíveis. Essas caixas com “descrição e preço” serão todos produtos disponíveis, contendo uma imagem superior, a descrição e o valor, e quando clicado, será adicionado uma unidade ao carrinho automaticamente. A barra inferior será a aba “carrinho”, onde todos os pedidos adicionados serão adicionados a esta aba, quando for clicado no espaço reservado do carrinho a aba será aberta sem mudar de página, ou seja, será atribuída uma animação onde os itens na tela serão substituídos pela aba carrinho. Nessa aba mostrará todos os pedidos selecionados. Veja o anexo [27], onde apresenta o *wireframe* da aba “carrinho”.

Como é possível visualizar, a aba carrinho apresenta todos os pedidos adicionados, na parte superior haverá um botão de retorno ao cardápio, para ser adicionado outros produtos. O espaço central são os produtos, separados por *cards*, e contendo imagens, nome e preço de cada item, os clientes poderão selecionar a quantidade dos produtos adicionando ou removendo um por um. Na parte inferior à direita, o sistema processará o valor total do pedido, e no canto inferior esquerdo haverá dois botões e um *select*, um para realizar o pagamento de forma virtual, utilizando cartões ou pagamento no caixa. Se o cliente clicar no botão para pagar o pedido virtual, o sistema processará uma página de pagamento onde será necessário inserir as informações para processar o pedido. Se o cliente preferir pagar o pedido no caixa, o pedido será adicionado à tela de gerenciamento do caixa, onde ele poderá manipular a situação de “Aguardando pagamento” para “Processando pedido”, sendo realizada essa mudança de *status* o pedido será processado para os funcionários responsáveis para a preparação dos pedidos. Portanto, o fluxo será segundo as permissões estabelecidas, o caixa poderá somente processar o pagamento para ser passado à equipe de cozinha para a preparação dos pedidos e com isso a equipe iniciará a preparação do pedido. O botão de seleção será

informado aos funcionários se o cliente quer o pedido para comer no local, ou para levar, essa informação irá de forma automática à tela de gerenciamento dos funcionários.

Após o cliente clicar no botão “Pagar pedido”, ele será levado à tela de pagamento, foi desenvolvido um protótipo com *wireframe*, observe o anexo [28].

A página de pagamento terá as opções para processar cartões e pix, e após a inserção dos dados terá o botão para a confirmação, essa tela é baseada na *API* de pagamento, onde o layout pode variar de acordo com o gateway escolhido, portanto essa tela é apenas uma amostra da tela de pagamento. Após o processo ser concluído o pedido solicitado será enviado à tela dos funcionários onde poderá ser manipulada as situações de cada pedido. Porém, o fluxo dos funcionários consiste em uma tela de *login*, onde os funcionários terão acessos a algumas ferramentas, dependendo do funcionário poderá haver alguma restrição na manipulação dos pedidos, restrições para aumentar a segurança do sistema. Observe o anexo [29] o *wireframe* desenvolvido contendo a tela de *login*.

Como é possível reparar, a tela de login em muito simples, onde haverá somente uma imagem lateral, e dois campos para informar o usuário e a senha, e após a inserção dos dados, o sistema verificará se estão cadastrados no banco de dados e com isso será retornado o resultado, entrando na aplicação ou retornando o erro de senha/usuário inválido.

Após os funcionários logarem no sistema será apresentado todos os pedidos realizados com os respectivos *ID's* e sendo assim, os funcionários poderão manipular os *status* dos pedidos. Observe o anexo [30] *wireframe* da tela de gerenciamento de pedidos.

A tela de gerenciamento se estrutura em cards, contendo as informações dos pedidos, como número do pedido gerado automaticamente, a quantidade de cada produto solicitado, ou seja, os hambúrgueres, porções ou bebidas e o campo de status contendo 3 etapas, “Solicitado”, “Em preparação” e “Finalizado”. Após a finalização do lanche o funcionário solicitará a retirada do pedido pelo ID gerado.

Com todo o desenvolvimento dos *wireframe* foi possível prototipar o ambiente a ser desenvolvido, onde a ideia inicial é criar um sistema com layout simples e agradável, tanto para os clientes quanto para os funcionários, lembrando que o modelo desenvolvido é apenas uma ideia de layout o sistema poderá sofrer alterações consoante a necessidade dos negociantes, podendo ser modificado tanto em layout, quanto em funções, o objetivo do sistema desenvolvido é a venda da ideia de facilitação de processos e não a venda de layouts em si.

Com a tela de gerenciamento também foi desenvolvido o *wireframe* para o adição dos lanches, onde os funcionários poderão alterar o cardápio, caso aja necessidade ou mudanças no estabelecimento, observe o anexo [31]. Como é possível observar, a tela consistirá em alguns campos para os funcionários preencherem com os dados necessários, onde poderá ser incluso o nome, o valor, a descrição e uma imagem do item a ser adicionado e com isso o item é lançado automaticamente à tela de home, onde os clientes poderão visualizar e selecionar os produtos disponíveis.

Com todos os protótipos construídos, o sistema foi desenvolvido com uma base sólida, seguindo todas as estratégias definidas, e com isso, foi possível obter um resultado significativo tanto na parte visual do sistema quanto na parte funcional. Observe o anexo [32] contendo a imagem da tela home desenvolvida.

Observe que a tela principal contém os lanches disponíveis e um botão para adicionar ao carrinho, na parte lateral esquerda há um menu com 3 links, onde são os itens disponíveis, nesse caso, os hambúrgueres, as

Bebidas e as porções e quando clicado, o sistema localiza automaticamente os itens relacionado ao clica, ou seja, se o cliente clicar nas bebidas, a tela será movida para baixo apresentado todas as bebidas disponíveis. No canto inferior a pré-visualização dos itens adicionados, além do botão para abrir o carrinho.

Após os clientes selecionarem os itens ele poderá clicar para abrir a aba do carrinho, observe o anexo [33] contendo a imagem do carrinho do sistema. Como é possível observar, o carinho apresenta os todos os pedidos com dois botões para a manipulação da quantidade de cada item adicionado, além disso, mostrando um botão na parte superior caso o cliente desejar adicionar outros itens, e também na parte inferior há 2 botões, um para pagar o pedido virtualmente ou pagar no caixa, ao lado também há o botão para selecionar se o pedido é para a viagem ou para o cliente comer no local, de acordo com essa decisão, o fluxo pode mudar, apresentando o modo que o funcionário deve montar o lanche, seja para comer no local ou levar.

Quando o cliente continuar com o pagamento do pedido ele será redirecionado ao *gateway* onde precisará incluir as informações, veja o anexo [34] da tela de pagamento de cartão ou pix. Como é possível observar, a tela apresenta os campos para a inserção dos dados para a continuação do processamento, e após os dados serem validados o pedido será processado e incluído à tela de gerenciamento de pedidos. Enquanto a aba de pix, apresenta o QRCode, com o valor total do pedido, a descrição e um botão para fechar a tela

Para os funcionários gerenciarem os pedidos é necessário realizar o login, observe o anexo [35] com a tela desenvolvida. Como é possível visualizar, a tela contém somente dois campos, com o botão para entrar no sistema e outro para criar uma conta nova. Caso o login estiver incorreto, será apresentado um erro na tela, para o funcionário corrigir a informação e tentar novamente.

Após o funcionário conectar no sistema ele será levado à tela de gerenciamento, onde será apresentado as informações referente aos pedidos realizados, contendo os campos anteriormente descritos, para a manipulação da situação de cada pedido, observe o anexo [36] contendo a tela de gerenciamento desenvolvida.

Finalizando o fluxo de processos, o sistema terá as telas para adicionar, alterar e remover os pedidos do sistema, onde caso o estabelecimento desejar alterar algum produto como bebida ou algum lanche, ele poderá realizar as manipulações sem dificuldades, observe o anexo [37], com as telas desenvolvidas. Como é possível observar todas as telas, como a tela de gerenciamento e as telas para adicionar, alterar e remover produtos serão apresentadas após os funcionários realizarem o login no sistema, essa estratégia facilita a manipulação das informações, onde não será necessário a separação das telas, com isso, é possível ganhar produtividade geral nas manipulações das informações.

Portanto, esse artigo apresentou, o desenvolvido completo de um sistema para o gerenciamento de lanchonetes, onde os processos poderão ser facilitados, tanto para os clientes quanto para os funcionários de uma determinada empresa, todas as técnicas e análises realizadas foram úteis para o desenvolvimento desse artigo, onde com bases sólidas foi possível encontrar “caminhos” para efetivar um sistema produtivo e eficiente.

4. CONSIDERAÇÕES FINAIS

Mediante aos problemas apresentados como, dificuldades em gestão, tempo ou praticidade, o sistema desenvolvido pode apresentar soluções viáveis para a melhoria geral dos estabelecimentos, portanto, foi analisado vários pontos gerenciais e técnicos para o desenvolvimento do sistema, sendo assim o resultado das pesquisas tanto de campo quanto de precificação são positivas onde o sistema pode ser implantado

pela falta de tecnologia e necessidade, além dos valores, serem bons para os pequenos empresários, podendo resultar em alavancagem de negócio, além disso, é possível observar que o sistema foi desenvolvido com sucesso, contendo todas as funções e recursos necessários para uma boa gestão de pedidos, também, é notória a boa oportunidade de atuação no mercado, onde as pesquisas realizadas pelo SEBRAE [2], [3], [4] mostram que o público-alvo de pequenos empreendedores, é um público abrangente, além de pouca tecnologia utilizada em seus estabelecimentos, com isso é possível ter uma alta chance de sucesso no mercado. Com todas as tecnologias e ferramentas utilizadas, o desenvolvimento do sistema se torna eficaz e atual no mercado, e isso se torna uma vantagem competitiva, onde os sistemas e softwares atuais estão sendo implantando tecnologias novas como MongoDB e Node Js, portanto, é importante está conforme as tendências do mercado.

Além disso, sendo desenvolvido um sistema web será possível alugar para vários comerciantes do país, não se limitando somente a uma cidade ou estado, outra vantagem em relação é a implantação em máquinas com pouca capacidade de processamento, onde os comerciantes podem não ter computadores potentes, e mesmo assim o sistema irá funcionar, pois o sistema será hospedado na internet, outra vantagem é a utilização do sistema em dispositivos móveis, já que o sistema é responsivo e pode ser adaptado para celulares e tablets. Também há a possibilidade de hospedar o sistema na máquina do cliente, caso ele não tiver acesso à internet ou criar um programa para instalar e executar o sistema sem a necessidade de hospedá-lo na máquina.

Com as análises de campo foi possível obter dados para a implementação do sistema, mesmo com relatividade da pesquisa, já que futuramente a probabilidade do avanço tecnológico é muito alta, onde a cada dia é construído novas tecnologias e sistemas que auxiliam os usuários, portanto, ainda há chance de melhoria no setor tecnológico e conseqüentemente a demanda por sistemas, e sendo assim, pode ser realizado ajustes nesse sistema para a implementação.

Em geral, o sistema é uma solução eficiente e personalizada para a Gestão de Pedidos em estabelecimentos do ramo alimentício. Com possíveis evoluções e melhorias, o sistema pode se tornar ainda mais completo e avançado para os clientes e usuários, além disso, é possível realizar outras pesquisas para encontrar “padrões” de propensões, para aplicar melhorias gerais no sistema, pois sendo assim, o sistema ficará com maior tendência para aplicações em lanchonetes e restaurantes, ficando mais prático para manutenções e alterações futuras.

Futuramente, será possível evoluir o sistema para obter uma nova versão, onde poderá conter algumas reestruturações e adicionamento de funções como, filtros de pesquisas para auxiliar os estabelecimentos a entender quais são os combos mais atrativos para os clientes e sendo assim, criar alguma promoção para ter um retorno significativo em suas vendas, ou realizar mudanças no cardápio visando atender melhor as demandas dos clientes. Criar uma tela de manual, onde explica detalhadamente como o sistema funciona e solucionar possíveis dúvidas com o manual. Também seria interessante criar uma tela de gerenciamento com gráficos de gestão para descobrir quais são os melhores horários e quais respectivos lanches são solicitados, com isso a equipe do estabelecimento poderiam planejar os horários de vendas e criar técnicas para alavancar seus negócios. Além disso, seria interessante criar a função para entrega Delivery, onde por meio da internet os clientes poderiam solicitar os pedidos sem a necessidade de ir ao estabelecimento, e criar uma aba com abertura de chamados para quando ocorrer algum erro no sistema em vez do cliente ter que ligar ou enviar um e-mail para os desenvolvedores, eles poderão preencher um formulário e escrever

seu problema ou dúvida e ter uma melhor organização de retorno, e por fim, criar um aplicativo executável e mobile, para os clientes dos estabelecimentos terem uma melhor experiência.

Portanto, com todas as observações realizadas, será possível obter mais uma versão de um sistema completo, atual e com funções atraentes para o público-alvo, além de atingir vários objetivos dos pequenos empreendedores, sempre buscando entender seus problemas e evoluir cada vez mais o sistema desenvolvido.

5. REFERÊNCIAS

- [1] BHS. Como o investimento em tecnologia contribui nos negócios. [publicação online]; 2021 [acesso em 02 set 2023]. Disponível em: <https://www.bhs.com.br/2023/01/18/investimento-em-tecnologia-crescimento-negocios/>
- [2] SEBRAE. Como vencer desafios tecnológicos para inovar? [publicação online]; 2013 [acesso em 02 set 2023]. Disponível em: <https://SEBRAE.com.br/sites/PortalSEBRAE/artigos/como-vencer-desafios-tecnologicos-para-inovar.b438d53342603410VgnVCM100000b272010aRCRD>
- [3] SEBRAE. Pequenos negócios em números [publicação online]; 2016 [acesso em 14 set 2023]. Disponível em: <https://www.SEBRAE.com.br/sites/PortalSEBRAE/ufs/sp/SEBRAEaz/pequenos-negocios-em-numeros.12e8794363447510VgnVCM1000004c00210aRCRD>
- [4] SEBRAE. Paronama dos pequenos negócios [publicação online]; 2018 [acesso em 14 set 2023]. Disponível em: [https://sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/SP/Pesquisas/Panorama dos Pequenos Negocios 2018 AF.pdf](https://sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/SP/Pesquisas/Panorama%20dos%20Pequenos%20Negocios%202018%20AF.pdf)
- [5] Mascarenhas SA. Metodologia Científica. (2th edição). São Paulo – SP [São Paulo - SP]: Casa de Ideias; 2018. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/183213/pdf/58>
- [6] Autores. Planilha_análise_Riscos [base de dados online]; 2023 [acesso em 10 set 2023]. Disponível em: <https://docs.google.com/spreadsheets/d/18hVccNchiOdck3H90RLuIvXlKEb96gaD/edit#gid=194331256>
- [7] Sbrocco JHTDC, Macedo PCD. Metodologias Ágeis - Engenharia de *Software* sob Medida. São Paulo - SP [São Paulo - SP]: Editora Saraiva; 2012. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788536519418/pageid/91>
- [8] Pressman RS, Maxim BR. Engenharia de *software*. (9th edição). Porto Alegre - RS [Porto Alegre - RS]: Grupo A; 2021. Disponível em: [https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/epubcfi/6/8%5B%3Bvnd.vst.idref%3DCreditos.xHTML%5D/4%5BPRESSMAN Completo-1%5D/2%5Bpage iv%5D/40/1:18%5B-70%2C00%5D](https://integrada.minhabiblioteca.com.br/reader/books/9786558040118/epubcfi/6/8%5B%3Bvnd.vst.idref%3DCreditos.xHTML%5D/4%5BPRESSMAN%20Completo-1%5D/2%5Bpage_iv%5D/40/1:18%5B-70%2C00%5D)
- [9] Consulting Eux, TUDO sobre MATRIZ SWOT: o que é, passo a passo e dicas para aplicar no planejamento estratégico, [publicação online]; 2020; [acesso em; 15 set 2023]. Disponível em: <https://blog.mettzer.com/formatacao-normas-vancouver/>
- [10] Reinehr S. Engenharia de requisitos. Porto Alegre - RS [Porto Alegre - RS]: Grupo A; 2020. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/pageid/218>
- [11] IBM Corporation. Diagramas de navegação [publicação online]. 2021 [acesso em 18 set 2023]. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=models-browse-diagrams>
- [12] Miro. *Wireframes* [publicação online]. 2023 [acesso em 18 set 2023]. Disponível em: <https://miro.com/pt/wireframe/o-que-e-wireframe/>
- [13] Kenzie [homepage na internet]. Melhor IDE para desenvolvimento web e suas características [acesso em 24 set 2023]. Disponível em: <https://kenzie.com.br/blog/ide-para-programacao/#:~:text=Segundo%20uma%20pesquisa%20feita%20pelo.editor%20de%20código%20Visual%20Studio>

- [14] [Balta.io](https://balta.io) [homepage na internet]. O que é um Framework. [acesso em 24 set 2023]. Disponível em: <https://balta.io/blog/o-que-e-um-framework>
- [15] DEVMEDIA. Introdução ao *MongoDB* [publicação online]. <https://www.ibm.com/br-pt>. 2021 [acesso em 28 Abr 2023]. Disponível em: <https://www.devmedia.com.br/introducao-ao-MongoDB/30792>
- [16] Martin RC. Código limpo: Habilidades práticas do Agile *Software*. Rio de Janeiro – SP. [Rio de Janeiro - RJ]: Editora Alta Books; 2009. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788550816043/pageid/40>
- [17] Zanin A, Júnior PAP, Rocha BC et al. Qualidade de *software*. Porto Alegre - RS [Porto Alegre - RS]: Grupo A; 2018. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595028401/pageid/2>
- [18] Autores. Gráfico: necessidades do cliente. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/16oGJp8rlp6PoVnQdTziYg6dGn2UiO3dm/view>
- [19] Autores. Gráfico: implementação do sistema. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/12YTA3L6dDRA4ZR5_W62kYXpYpJdJApVZ/view
- [20] SEBRAE Respostas. Quanto Custa um Sistema para Restaurante? [publicação online]; 2021 [acesso em 26 out 2023]. Disponível em: <https://respostas.sebrae.com.br/quanto-custa-um-sistema-para-restaurant/>
- [21] Autores. Análise de precificação de *tablets*. São Paulo; 2023. Disponível em: https://docs.google.com/document/d/12hbpRz_X-A4YOaOj5uYr_1ScNoGJYZre/edit?usp=sharing&oid=104163071683163909449&rtpof=true&sd=true
- [22] Aqua. Quanto custa um totem interativo? [publicação online]; 2017 [acesso em 26 out 2023] disponível em: <https://www.aqua.com.br/blog/preco-totem-interativo>
- [23] Autores. Figura: Fluxo de processos. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/13_6vDI_EQaa1NIPmDnMEGJaK4q6KuXLi/view
- [24] Autores. Figura: Diagrama *BPMN*. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/1Hu89dggTTxcDz_HQWrLe2gxB2RindjPV/view
- [25] Autores. Figura: Diagrama de navegação. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/1XBaaRY_Y_Zb95dj8hzBi0NL1v2w2HWSh/view
- [26] Autores. Figura: *Wireframe* tela home. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/1Edt4AoynwVLPaO9MZGXyMXnz1esH3Bvg/view?usp=sharing>
- [27] Autores. Figura: *Wireframe* aba carrinho. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/1C7-93jKKiN37yHVasVzEBekQ7X1RXPh8/view>
- [28] Autores. Figura: *Wireframe* tela de pagamento. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/1kUwtFXtfmWKgkLk8obuDLk8ZYzWGHZhtC/view>
- [29] Autores. Figura: *Wireframe* tela de *login* do funcionário. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/18JZKU0uKLRhsnZ6ICBXLqJFUmWN7_I6e/view
- [30] Autores. Figura: *Wireframe* tela de gerenciamento. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/1sxC4tAesc6Mht7eS-qBI2pahFlhVaGu-/view>
- [31] Autores. Figura: *Wireframe* tela para adicionar pedidos. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/17gWRe2f1o1wt6ZZH1tE3MP_dNyhfcRSx/view?usp=sharing
- [32] Autores. Figuras: Tela home. São Paulo; 2023. Disponível em: <https://drive.google.com/file/d/1jvy6YxfFiWnsqUhjhpPwRxc1QwfK5MI/view?usp=sharing>
- [33] Autores. Figuras: Aba Carrinho. São Paulo; 2023. Disponível em: https://drive.google.com/file/d/1A1V0NIL6F-JR4ULZVZMpabgX0Xtrg_h8/view?usp=sharing
- [34] Autores. Figuras: Tela de pagamento. São Paulo; 2023. Disponível em: https://drive.google.com/drive/folders/1O7buhURnStnVz4cdLDXE3_eHzoYsgkkK?usp=sharing