



**UNIVERSIDADE DO SUL DE SANTA CATARINA**  
**MATHEUS BERKENBROCK NEDEL**

**SISTEMA INTEGRADO PARA ATENDIMENTO EM RESTAURANTES**

Florianópolis - SC

2020

**MATHEUS BERKENBROCK NEDEL**

**SISTEMA INTEGRADO PARA ATENDIMENTO DE RESTAURANTES**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de informação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Maria Inés Castiñeira, Dra.

Florianópolis - SC

2020

**MATHEUS BERKENBROCK NEDEL**

**SISTEMA INTEGRADO PARA ATENDIMENTO DE RESTAURANTES**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina.

Florianópolis, 15 de julho de 2020.

---

Professor e orientador Maria Inés Castiñeira, Dra.  
Universidade do Sul de Santa Catarina

---

Prof. Aran Bey Tcholakian Morales, Dr.  
Universidade do Sul de Santa Catarina

---

Prof. Edson Orivaldo Lessa Junior, Esp.  
Universidade do Sul de Santa Catarina

## **AGRADECIMENTOS**

Primeiramente, à minha mãe, Andréa Berkenbrock, que nunca mediu esforços para providenciar o que fosse necessário aos seus filhos. Nada disso teria sido possível em sua ausência.

Ao meu irmão Lucas, sempre presente e preocupado em me ajudar em momentos difíceis.

Ao meu pai, Flávio, e ao restante de minha família, por toda a empatia e afeto.

Ao namorado da minha mãe, Hélio.

À minha orientadora, Maria Inés Castiñeira, pela paciência, conhecimento e ajuda durante todo o processo deste trabalho.

À todos os professores que ajudaram a formar quem sou hoje.

E, por último, mas não menos importante, à meus amigos: Bruno, Thiago, Facundo, e todos outros.

“A suprema arte da guerra é derrotar o inimigo sem lutar.” (Sun Tzu)

## RESUMO

Dentro do contexto atual de forte uso de tecnologia, existem várias oportunidades de modernizar e melhorar processos cotidianos. Um forma comum de se fazer é criar um meio em que o fornecedor de serviço possa contatar diretamente seus clientes através do uso da tecnologia, e a partir disso, obtém-se duas melhorias fundamentais: a eficiência e qualidade do processo, assim como a coleta de dados referentes ao serviço em questão. Portanto, este trabalho propõe uma solução que forneça estas melhorias, em particular dentro do ramo de serviços alimentícios, criando um sistema integrado de atendimento em restaurantes, composto por três frentes: uma aplicação para o cliente, uma para o gestor e outra para a cozinha. A aplicação do cliente é uma interface onde se pode fazer pedidos e executar pagamento, entre outras funções; a do gestor, onde se gere os recursos do restaurante referentes ao sistema em questão e por fim, a da cozinha, onde se recebe e controla os pedidos sendo feitos pelos clientes. Inicialmente foi feita uma pesquisa por abordagens de desenvolvimento mobile e uma comparação entre as características e funcionalidades entre tecnologias de desenvolvimento mobile nativo, híbrido e web. Ao concluir a pesquisa optou-se por desenvolver um PWA para a aplicação do cliente, que consiste numa abordagem de tecnologia web. Em específico as tecnologias escolhidas foram: React, Firebase e Material UI, entre outras. Foi seguido uma abordagem metodológica de reunir informações, conduzir estudos de temas e ferramentas a serem utilizados, modelar o sistema a partir do levantamento de requisitos e criação de protótipos e por fim, conduzir uma avaliação da interface do cliente. Sendo assim, o resultado final foi a criação de um sistema capaz de satisfazer os objetivos propostos e receber uma avaliação bastante positiva pelos usuários testados.

Palavras-chave: React. Firebase. Material UI. Aplicativo para restaurantes. PWA. Abordagens desenvolvimento mobile.

## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>10</b>
1.1 PROBLEMÁTICA	11
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral	12
1.2.2 Objetivos Específicos	12
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO	13
<b>2 REVISÃO BIBLIOGRÁFICA</b>	<b>15</b>
2.1 SISTEMAS SIMILARES	15
2.1.1 Goomer	15
2.1.2 Menu Digital	16
2.1.3 MENU	17
2.1.4 Comparações finais	18
2.2 DESENVOLVIMENTO MOBILE	19
2.2.1 Abordagens de Desenvolvimento	21
2.2.2 Ferramentas para Desenvolvimento Mobile	22
2.2.2.1 React Native	23
2.2.2.2 Ionic	24
2.2.2.3 Progressive Web Apps	24
2.4 CONSIDERAÇÕES FINAIS	25
<b>3 MÉTODO</b>	<b>27</b>
3.1 TIPO DE PESQUISA	27
3.2 ETAPAS	28
3.4 AVALIAÇÃO DO SISTEMA	30
3.5 DELIMITAÇÕES	30
<b>4 MODELAGEM</b>	<b>31</b>
4.1 ATORES	31
4.2 REQUISITOS	33
4.2.1 Requisitos Funcionais	33
4.2.2 Requisitos Não Funcionais	38
4.3 CASOS DE USO	39
4.3.1 Casos de uso da aplicação do cliente	40

4.3.1.1	Descrição dos casos de uso	41
4.3.1.1.1	Caso de uso de abrir conta no restaurante	41
4.3.1.1.2	Caso de uso de consultar cardápio	41
4.3.1.1.3	Caso de uso de fazer pedido	42
4.3.1.1.4	Caso de uso de realizar pagamento	43
4.3.2	Casos de uso da aplicação do gestor	45
4.3.2.1	Descrição dos casos de uso	46
4.3.2.1.1	Caso de uso de consultar cardápio	46
4.3.2.1.2	Caso de uso de adicionar ou editar item	46
4.3.2.1.3	Caso de uso de remover item	47
4.3.2.1.3	Caso de uso de editar detalhes do restaurante	48
4.3.2.1.4	Caso de uso de consultar estatísticas	49
4.3.3	Casos de uso da aplicação do cozinheiro	50
4.3.3.1	Descrição dos casos de uso	50
4.3.3.1.1	Caso de uso de consultar pedidos abertos	50
4.3.3.1.2	Caso de uso de editar estado de preparação do pedido	51
4.4	PROTOTIPAÇÃO DAS TELAS	53
4.4.1	Telas da aplicação do cliente	53
4.4.2	Telas da aplicação do gestor	60
4.4.3	Telas da aplicação da cozinha	64
<b>5</b>	<b>DESENVOLVIMENTO</b>	<b>66</b>
5.1	TECNOLOGÍAS E FERRAMENTAS	66
5.1.1	TypeScript	67
5.1.2	Firebase	67
5.1.3	Preact	68
5.1.4	Webpack	68
5.1.5	Material UI	69
5.1.6	Workbox	70
5.2	HISTÓRICO DE DESENVOLVIMENTO	70
5.2.1	Estudos e configuração do projeto	70
5.2.2	Implementação do Firebase	71
5.2.3	Roteamento	71
5.2.4	Complexidade de dados	72
5.3	SISTEMA DESENVOLVIDO	73
5.3.1	Telas iniciais	73
5.3.2	Tela Home	74
5.3.3	Telas do cardápio	75



5.3.4 Tela de fechar conta	77
5.2 AVALIAÇÃO DO SISTEMA	79
<b>6 CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>84</b>
6.1 CONCLUSÕES	84
6.2 TRABALHOS FUTUROS	85
<b>REFERÊNCIAS</b>	<b>87</b>
<b>APÊNDICE A - CRONOGRAMA</b>	<b>91</b>

## INTRODUÇÃO

Segundo uma pesquisa da FGV (2019), o Brasil possui 230 milhões de smartphones, mais celulares que pessoas. A adoção intensa de celulares para diversas tarefas, graças a evolução da capacidade computacional dos aparelhos, substitui o uso dos computadores convencionais. Esta característica de ter um computador presente a todo momento do dia, e com um vasto ecossistema de aplicativos, trouxe e continua trazendo grandes mudanças nas dinâmicas da sociedade, tanto na forma como nos comunicamos, obtemos informações, realizamos tarefas cotidianas como operações financeiras, serviços de delivery e até transporte. Mudanças estas que dificilmente teriam sido previstas, como exemplo do astrônomo Clifford Stoll que em 1995, no início da revolução digital, escreveu (STOLL, 1995, tradução nossa):

Besteira. Nossos especialistas em computadores não possuem nenhum bom senso? A verdade é que nenhum database online substituirá o seu jornal diário, nenhum CD-ROM pode tomar o lugar de um professor competente e nenhuma rede de computadores irá mudar a forma como o governo funciona.

Uma demonstração relevante destas mudanças é o aplicativo Uber, que criou um mecanismo digital que conecta motoristas individuais com passageiros, efetivamente digitalizando a forma como os táxis funcionam. Dada as proporções que a empresa Uber tomou, este fenômeno de adaptar serviços existentes ao meio digital tem sido constantemente chamado de *uberização*, e é um movimento natural de uma sociedade cada vez mais moderna. De acordo com Minetto (2015) "a tendência é que cada vez mais serviços sejam *uberizados*, e quem não entrar nessa onda deve ficar para trás".

A ideia de uberizar consiste na substituição de um intermediário por um mecanismo digital, onde se tem o contato direto do fornecedor do serviço com o consumidor. Resultando no que pode ser chamado de uma economia de compartilhamento, ou *People-to-People Economy*, conforme descreve Nurvala (2015, p. 233, tradução nossa):

A *Uberização* do setor de serviços se tornará o padrão no futuro. O fator impulsionante por trás deste desenvolvimento é um colapso dramático em custos de operação feitos possíveis por novos e mais desenvolvidos sistemas de conexão baseadas em internet. O que está emergindo pode ser chamado de *People-to-People Economy*, um termo que descreve a natureza difusa do novo modelo.

Este novo modelo apresenta vários benefícios que giram em torno do aumento de eficiência, reduzindo custos para quem fornece o serviço e, em contrapartida, também para quem o contrata. Este modelo muitas vezes também simplifica a contratação e até mesmo aumenta a qualidade dos serviços com mecanismos de qualificação.

## 1.1 PROBLEMÁTICA

A ausência de um sistema digital e automatizado para o atendimento em restaurantes pode gerar vários aspectos de ineficiência e custos para o estabelecimento.

Para o empreendedor, se tem a necessidade de uma quantidade de garçons, que pode ser excessiva, refletindo num aumento na dificuldade de gestão e na responsabilidade com a segurança do trabalhador. De acordo com NIENKÖTTER (2018, p. 10) "A preocupação com a saúde e a segurança do trabalhador é antiga e se estende até os dias atuais". Além disso, há também um custo elevado na operação e por consequência no produto final. Conforme Souza et al. (2012, p. 4) "o custo de um trabalhador pode ser de 2,83 vezes o salário de carteira dele (aumento de 183%), no caso de um vínculo com doze meses de duração".

Existe também a necessidade de impressão de novos cardápios a cada vez que algum item é atualizado e uma certa ineficiência de recebimento de pedidos para cozinha, como atrasos e esquecimentos de adicionais de pedidos.

Já para o cliente, além de preços mais acessíveis, um sistema digital pode contribuir com mais informações e imagens sobre itens do cardápio, mais controle para

fazer pedidos à cozinha, e fornecer uma alternativa de pagamento mais objetiva e transparente.

Em particular, este trabalho aborda um sistema genérico (por ser um sistema único que pode ser utilizado por diversos restaurantes) de cardápio digital pelo qual o usuário pode realizar a solicitação do seu pedido, entre outros serviços de atendimento como pagamento e avaliação de pratos e serviço.

Por fim, este trabalho propõe o desenvolvimento de um PWA (*Progressive Web App*), aplicações web que podem ser baixadas para o dispositivo do usuário e funcionar de forma similar à aplicativos nativos, que providencie uma experiência melhor tanto para o gestor e empreendedor de um estabelecimento como para o cliente final.

## 1.2 OBJETIVOS

A seguir o objetivo geral e específicos deste trabalho.

### 1.2.1 Objetivo Geral

Este trabalho tem como objetivo desenvolver um sistema de informação, chamado aqui de cardápio digital, através de uma solução web e mobile que agilize a forma de atendimento em restaurantes.

### 1.2.2 Objetivos Específicos

São objetivos específicos deste trabalho os seguintes:

- Desenvolver uma solução que permita simplificar o atendimento de restaurantes.
- Adquirir dados que auxiliem na gestão do estabelecimento.
- Avaliar o funcionamento do sistema.

### 1.3 JUSTIFICATIVA

É eminente que restaurantes e diversos tipos de negócios devem se atentar as tendências de digitalização de serviços. Conforme Chiavenato (2008, p.2) "O mundo está mudando com uma rapidez incrível. E com intensidade cada vez maior. A mudança sempre existiu na história da humanidade, mas não com o volume, rapidez e impacto com que ocorre hoje." A concorrência de mercado cria a necessidade de constante aprimoramento ao risco de ser excluído do mesmo, portanto, o estabelecimento deve-se manter à frente das tendências e tecnologias.

O atendimento de restaurantes é um tipo de serviço que pode ser aprimorado em grande quantidade com a implementação de um sistema digital integrado, de forma que só se necessitaria atendentes dedicados às interações de fato necessárias com o cliente, sem desperdícios de tempo com anotação de pedidos ou processamento de pagamentos; a seleção de pedidos poderia ser feita de forma mais simples, rápida e clara; a cozinha teria um sistema próprio exibindo instantaneamente novos pedidos, de forma objetiva e ordenada; e a gestão do estabelecimento como um todo se tornaria mais simples e efetiva.

Diante destas oportunidades de melhoria, este trabalho tem como justificativa o desenvolvimento de um sistema integrado que proporcione um meio de atendimento e gestão efetivo e mais adequado às tecnologias atuais.

### 1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em 6 capítulos, descritos a seguir:

- Introdução: define o trabalho, explorando o problema e os objetivos a serem solucionados;

- Revisão bibliográfica: aborda os temas e conhecimentos que fundamentam o trabalho e que serão utilizados para desenvolver a solução proposta;
- Método: apresenta a forma como será discorrido o trabalho, com classificação de pesquisa, atividades metodológicas e delimitações;
- Modelagem: demonstra o esboço inicial do projeto, bem como requisitos, casos de uso e prototipação;
- Desenvolvimento: etapa onde o sistema será desenvolvido de fato, apresentando a estrutura, conceitos técnicos e tecnologias utilizadas, também ressaltando pontos relevantes e dificuldades ao longo do desenvolvimento e, por fim, demonstrando o resultado final do sistema;
- Considerações finais: impressões finais do resultado do trabalho e levantamento de possíveis melhorias.

## 2 REVISÃO BIBLIOGRÁFICA

A seguir serão apresentados os assuntos teóricos abordados por este trabalho, tais como estado da arte, tecnologias relevantes a sistemas móveis como *Firebase*, *React Native*, técnicas de desenvolvimento como design patterns, entre outros.

### 2.1 SISTEMAS SIMILARES

Nesta seção é analisado os principais serviços de propósito semelhante ao deste trabalho, tanto de origem nacional como internacional, com o objetivo de coletar informações a fim de elaborar uma solução mais refinada para o problema.

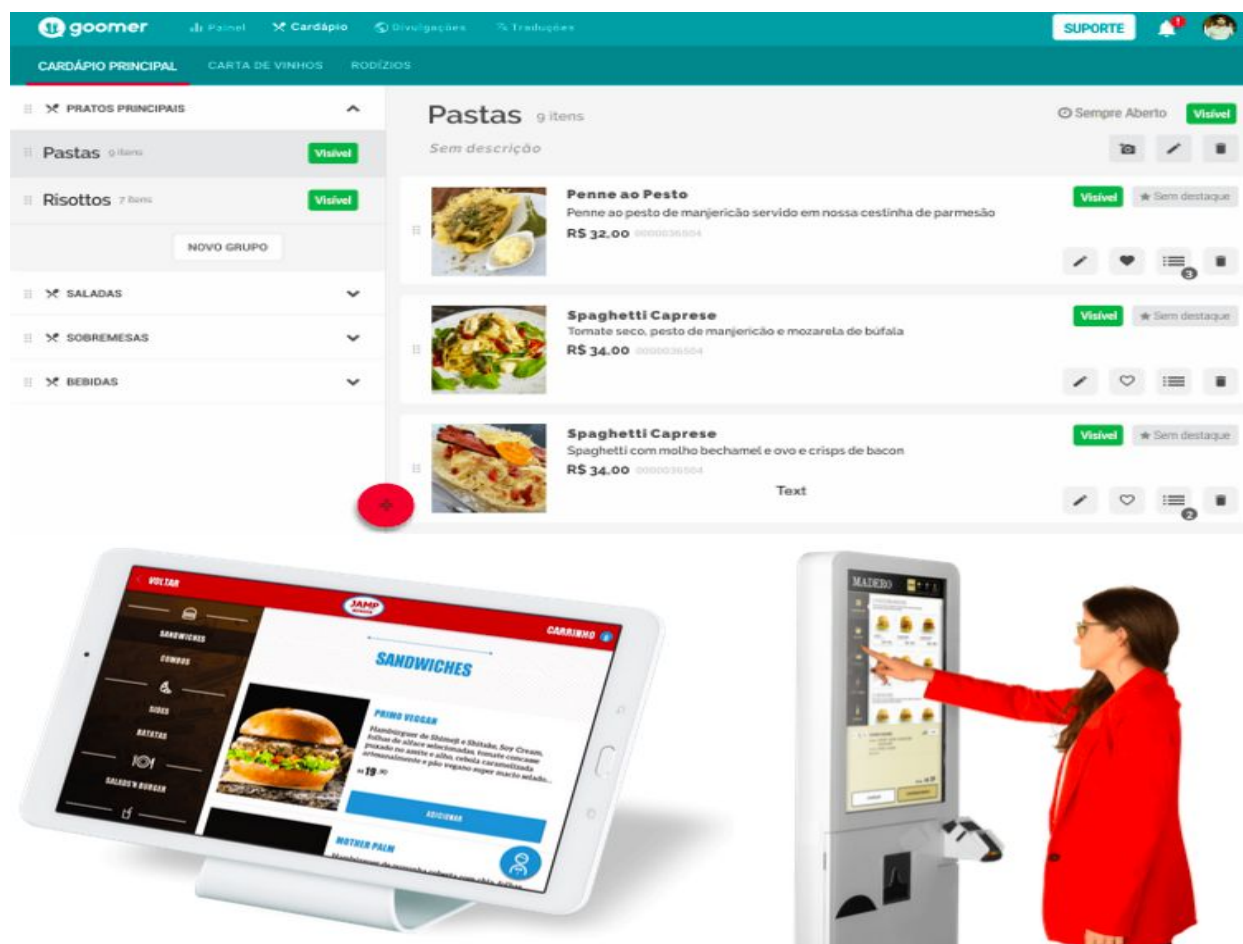
#### 2.1.1 Goomer

Goomer (2019) é uma empresa brasileira que fornece serviços de cardápio digital e totens de autoatendimento. É utilizada em restaurantes como Madero, Jeronimo, Guacamole, entre outros.

Funciona providenciando uma versão específica do sistema para a marca contratante, com uma página de gestão do cardápio (imagem superior na figura 1) sendo que a marca contratante é encarregada dessa gestão de cardápio. Tem opções de tablets distribuídos mesa em mesa ou a utilização de totens de autoatendimento.

A seguir algumas imagens do sistema:

Figura 1 - Imagens do sistema da empresa Goomer.



Fonte: GOOMER, 2019.

## 2.1.2 Menu Digital

Menu Digital (2019) é outra empresa brasileira. Tem o funcionamento parecido ao que este trabalho propõe, porém disponibilizando o cardápio ao cliente em formato de um site específico para o restaurante. Percebe-se nesta empresa como pontos negativos a falta de um aplicativo como opção e uma interface desatualizada e de pouca usabilidade.



Figura 2 - Logo da empresa Menu Digital.



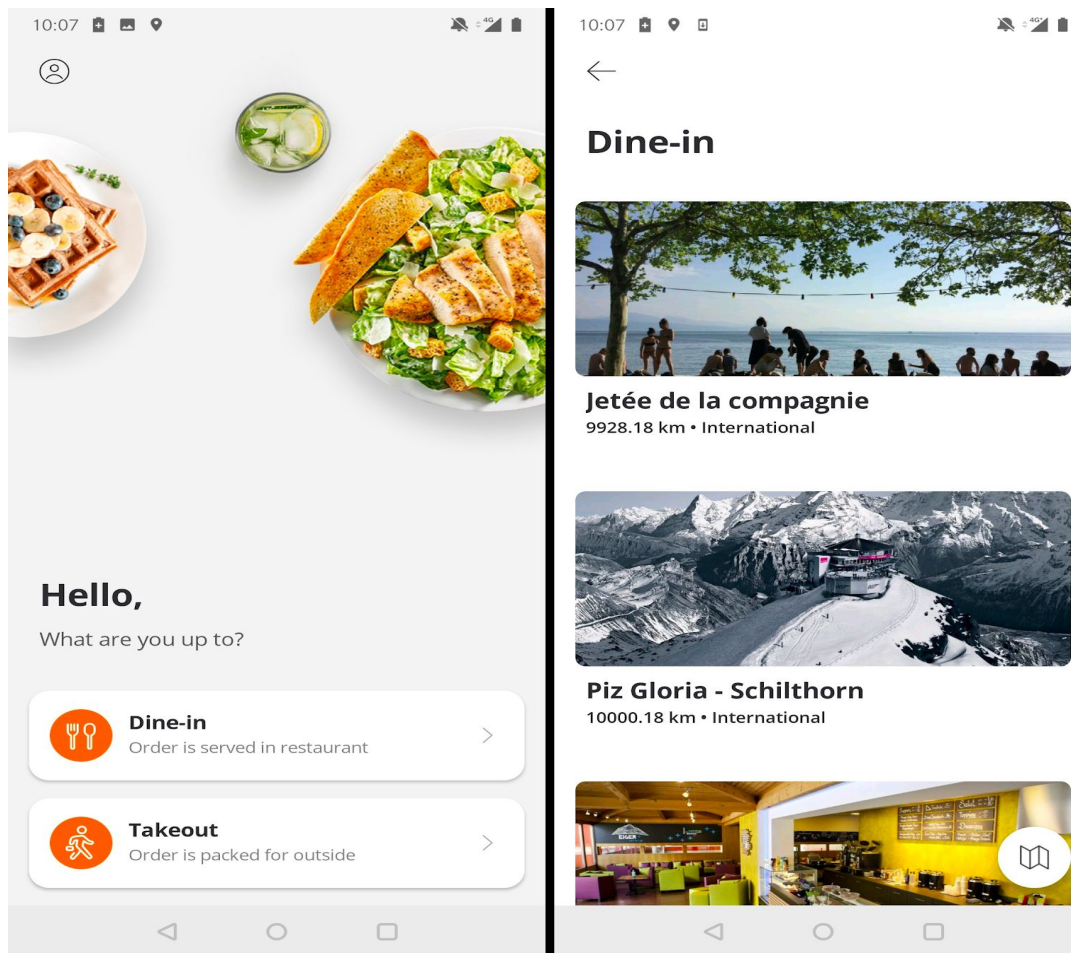
Fonte: Menu Digital, 2019.

Tem recursos como painel de controle para alteração de preços, serviços, ocultar itens indisponíveis do cardápio, entre outros.

### **2.1.3 MENU**

Esta é uma empresa suíça que funciona da mesma forma aqui proposta, um aplicativo único que pode ser utilizado para realizar pedidos à diversos restaurantes. Possui uma interface agradável e diferenciais como busca de restaurantes na região e disponibilidade de serviços para hotéis (MENU, 2019).

Figura 3 - Imagens do sistema da empresa MENU.



Fonte: MENU, 2019.

São pontos interessantes do sistema desta empresa recursos como gestão de filas, pagamento automatizado através do aplicativo e opção de take away.

#### 2.1.4 Comparações finais

O quadro a seguir compara as funcionalidades de cada sistema analisado com a solução proposta neste trabalho.

Quadro 2 - Comparação dos diferentes sistemas.

	Goomer	Menu Digital	MENU	Solução proposta
Gestao de cardapio	✓	✓	✓	✓
Utilizado por tablets em cada mesa	✗	✗	✗	✗
Totens de auto atendimento	✓	✗	✗	✗
Site para auto atendimento de clientes	✗	✓	✗	✓
Aplicativo para auto atendimento de clientes	✗	✗	✓	✓
Busca de restaurantes	✗	✗	✓	✗

Fonte: Elaboração do Autor, 2020.

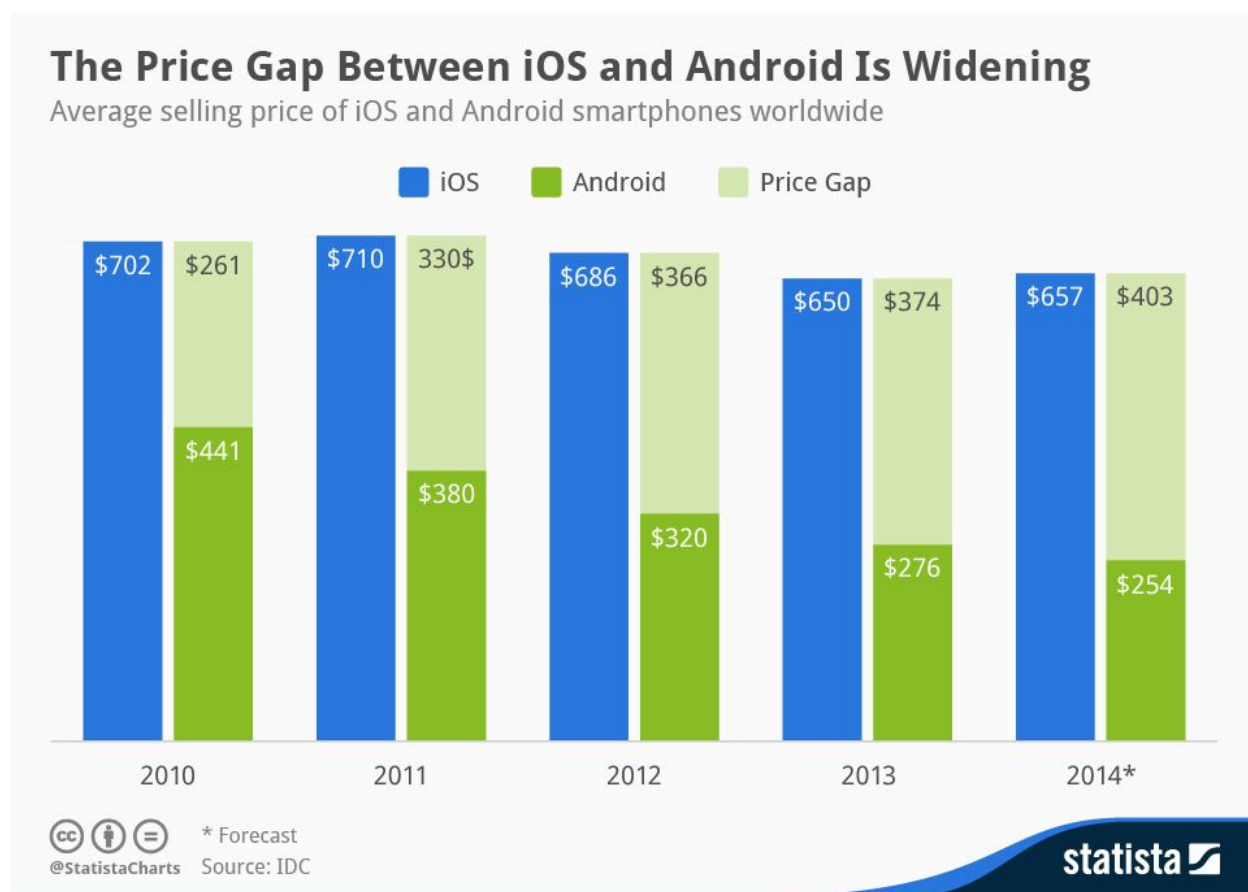
## 2.2 DESENVOLVIMENTO MOBILE

O desenvolvimento para plataformas móveis é focado principalmente nas plataformas Android e iOS por conta de sua maior adoção de mercado. Existem vários outros sistemas operacionais móveis, como mencionados por Palmieri, Singh e Cicchetti (2012) Windows, Blackberry OS, Ubuntu Touch, KaiOS, entre outros, porém, somente o Android e o iOS se sobressaíram na disputa pelos consumidores.

O sistema iOS é desenvolvido pela empresa Apple (2019), foi apresentado inicialmente em 2007 na apresentação do primeiro iPhone e causou agitação no público por ser um dos primeiros smartphones com tela sensível ao toque. Apesar de ser um sistema de ótima qualidade, ainda possui uma adoção de mercado relativamente baixa se comparada ao Android. Isso se dá por conta do sistema não ser distribuído pela Apple para ser utilizado em outros aparelhos, além dos seus próprios, que tendem a ter um preço consideravelmente acima da média.

Em comparação, o sistema Android é distribuído gratuitamente, possui código aberto (GOOGLE SOURCE, 2019) e, por conta destas características, fabricantes de todos os tipos podem utilizá-lo, resultando num número maior de aparelhos por um preço geralmente mais acessível. No gráfico à seguir é possível ver uma comparação de preço de sistemas com iOS e Android, demonstrando que em 2014, por exemplo, existia uma diferença de 403 dólares entre cada sistema.

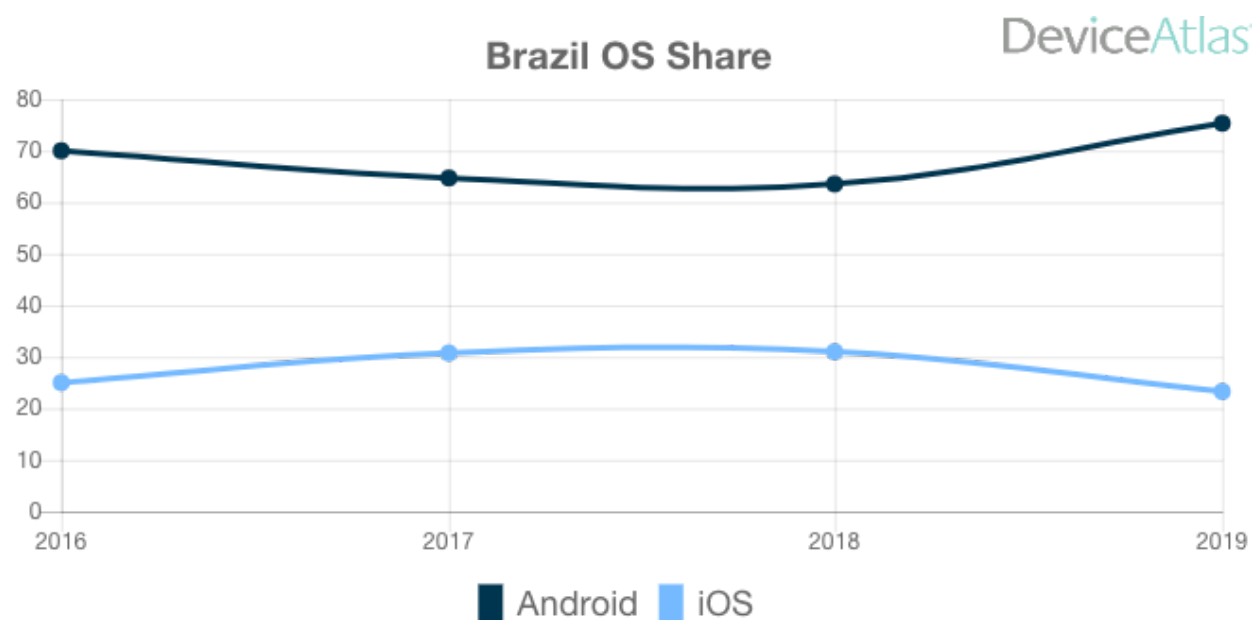
Gráfico 1 - Comparação de preços de aparelhos com iOS e Android.



Fonte: Statista, 2019.

Esta diferença acentuada de preços naturalmente causa que em países cuja população tenha um poder de compra limitado, os sistemas de custo menor tendem a ter uma adoção de mercado maior, como é o caso do Brasil, demonstrado no gráfico a seguir.

Gráfico 2 - Distribuição de sistemas operacionais móveis Brasil.



Fonte: DeviceAtlas, 2019.

Apesar do sistema Android apresentar uma adoção de mercado muito maior, o iOS, com aproximadamente 25% de adoção, ainda é muito relevante, e isso constituiu um problema pelo ponto de vista de desenvolvimento para dispositivos móveis, pois um programa desenvolvido para uma plataforma dificilmente era compatível com a outra.

### 2.2.1 Abordagens de Desenvolvimento

A principal dificuldade no desenvolvimento de aplicativos móveis era a necessidade de utilizar a linguagem de programação e estrutura específica de cada plataforma, Java para Android e Objective-C para iOS, resultando no desenvolvimento

simultâneo de dois aplicativos idênticos. Ao decorrer do tempo surgiram linguagens mais modernas para cada plataforma como Kotlin e Swift (MARQUES, 2017), mas o problema original persistiu.

Com o avanço das tecnologias tanto móveis quanto para web, começou a surgir alternativas de desenvolvimento que suportam múltiplas plataformas, chamadas de *cross-platform development approaches*, ou abordagens de desenvolvimento multiplataforma. De acordo com Kremer (2019, tradução nossa), estas abordagens podem ser divididas em híbrido-nativo e híbrido-web.

A abordagem híbrido-nativo consiste em desenvolver com uma só linguagem que através do framework utilizado faz chamadas à componentes nativos da plataforma móvel específica. Por exemplo, a utilização de um mesmo componente de texto de um framework pode resultar em dois componentes diferentes: *TextView* para Android e *UIView* para iOS (Kremer, 2019, tradução nossa). O resultado é que o código é mantido o mesmo, mas os componentes de interface são diferentes.

Por outro lado, a abordagem híbrido-web funciona rodando em um container web chamado *webview*, que é um browser embutido na plataforma (KIRUPA, 2019, tradução nossa), ou seja, é capaz de interpretar tecnologias web como HTML, CSS e JavaScript. O resultado disso é que além do código, os componentes também são compartilhados entre as diferentes plataformas.

Apesar da abordagem híbrido-web parecer apresentar vantagens sobre a híbrido-nativo, ela tende a ter uma performance um pouco mais baixa pela necessidade de rodar dentro de um container web (SORAL, 2019, tradução nossa).

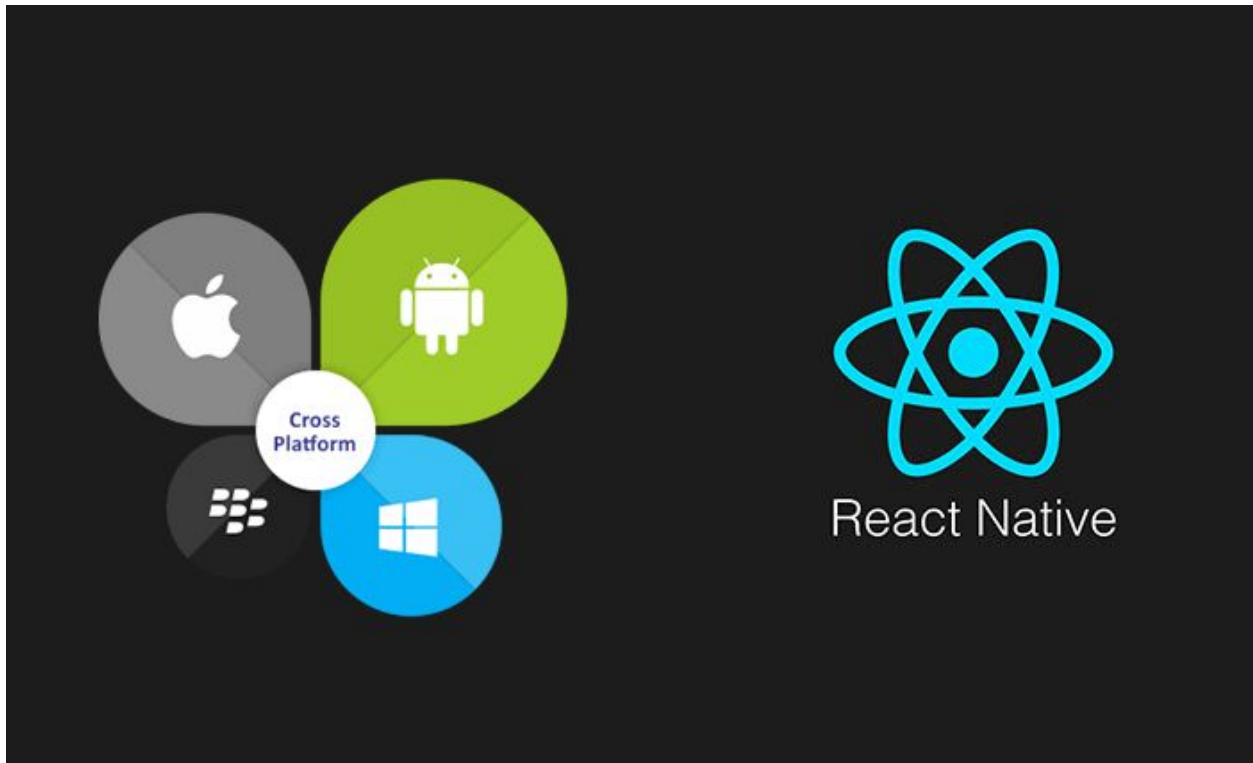
## **2.2.2 Ferramentas para Desenvolvimento Mobile**

Dentro destas abordagens de desenvolvimento existem diversas ferramentas, cada uma com suas características e particularidades. A seguir será discutido um pouco sobre dois notáveis frameworks de cada abordagem.

### 2.2.2.1 React Native

React Native é a versão para o desenvolvimento de aplicativos móveis do framework ReactJS.

Figura 4 - Ferramenta de desenvolvimento React Native



Fonte: Thinkwik, 2018.

Como visto na figura 4, com esta ferramenta é possível desenvolver para todas as principais plataformas mobile. Esta abordagem se encaixa no modelo híbrido-nativo pois funciona mapeando chamadas de interface para os componentes nativos de cada plataforma.

É atualmente uma das ferramentas mais utilizadas para o desenvolvimento mobile.

#### 2.2.2.2 Ionic

Ionic é uma ferramenta de desenvolvimento híbrida-web. Começou originalmente utilizando como base o framework Angular, da Google, para criar um aplicativo com tecnologias web emulado dentro de um webview, mas atualmente suporta outros frameworks de desenvolvimento como React e VueJS.

Figura 5 - Ferramenta de desenvolvimento Ionic.



Fonte: Ionic, 2019.

Na figura 5 consta o logotipo da empresa por trás da ferramenta. É importante notar que além de mobile, o Ionic hoje suporta desenvolvimento para quase todos outros tipos de plataforma, incluindo desktop.

#### 2.2.2.3 Progressive Web Apps

De acordo com DeJong (2018) os PWAs oferecem diversos benefícios em oposição a aplicativos tradicionais, como não precisarem serem instalados através da loja de apps específica da plataforma, podem ser indexados por mecanismos de buscas, entre outros benefícios, listados no quadro a seguir.



Quadro 2 - Características de diversos tipos de aplicativos.

	Aplicativo Nativo	Website Responsivo	PWA
Funcionamento offline	✓	✗	✓
Notificações push	✓	✗	✓
Instalável	✓	✗	✓
Funcionamento em full screen	✓	✗	✓
Indexável por mecanismos de busca	✗	✓	✓
Funciona em plataformas diferentes	✗	✓	✓
Download não necessário	✗	✓	✓
Updates não necessários	✗	✓	✓

Fonte: One North, 2018.

Por conta destas características, diversas empresas reconstruíram seus sites móveis para se tornarem PWAs, e obtiveram grandes resultados. Um exemplo é o Twitter, que teve um aumento de 75% de tweets enviados e 65% de páginas por sessão, e uma diminuição de 20% na taxa de rejeição - usuários que saíram do site após visitar somente uma página (GOOGLE, 2017).

## 2.4 CONSIDERAÇÕES FINAIS

Este capítulo abordou diversos assuntos que serão aplicados e que são necessário para a fundamentação teórica do trabalho. Foram abordados assuntos como o estado da arte, ideias similares a que este trabalho propõe; os diferentes tipos de desenvolvimento mobile, diferentes abordagens e algumas ferramentas utilizadas

em cada uma destas abordagens. Em seguida se dará continuidade com o método do trabalho.

### 3 MÉTODO

O método aqui utilizado é de imprescindível dada a natureza científica desta pesquisa. Portanto, a seguir será abordado e detalhado os diferentes métodos utilizados na realização deste trabalho.

#### 3.1 TIPO DE PESQUISA

Para começar o trabalho de definir e detalhar os métodos, surge a necessidade do entendimento das classificações e diferentes conceitos de pesquisa. Pesquisa de forma geral surge da necessidade de coletar informações, conhecimentos e formular hipóteses para encontrar uma solução de um problema definido. De acordo com Hansen (2009, online, tradução nossa) “O critério principal de sucesso em pesquisa aplicada é a contribuição para a solução de problemas práticos específicos”. Assim, este trabalho pode ser caracterizado como uma pesquisa aplicada.

No caso específico deste trabalho a pesquisa consiste no desenvolvimento de um protótipo que tem como objetivo solucionar o problema proposto. Para tal é necessário a busca de informações sobre desenvolvimento de software, especificamente na área mobile, diferentes abordagens e suas vantagens e desvantagens, assim como as ferramentas disponíveis no mercado e qual delas melhor se aplica a este caso, assim desenvolvendo previamente uma pesquisa bibliográfica.

Dentro deste contexto, é cabível afirmar que a pesquisa qualitativa é a que melhor se encaixa para os fins aqui definidos, buscando aprofundar os conhecimentos em livros e trabalhos relacionados a fim de encontrar uma proposta ideal para o problema proposto. Günther (2006) afirma que a pesquisa qualitativa pode ser caracterizada por cinco grupos de atributos: características gerais, coleta de dados, objeto de estudo, interpretação dos resultados e generalização.

De acordo com Souza, Santos e Dias (2013, p. 60):

A pesquisa sempre parte de um problema, de uma interrogação, uma situação para a qual o repertório de conhecimento disponível não gera resposta adequada. Para solucionar esse problema, são levantadas hipóteses que podem ser confirmadas ou refutadas pela pesquisa. Portanto, toda pesquisa se baseia em uma teoria que serve como ponto de partida para a investigação. Assim, pesquisar, num sentido amplo, é procurar uma informação que não sabemos e que precisamos saber. Consultar livros e revistas, verificar documentos, conversar com pessoas, fazendo perguntas para obter respostas, são formas de pesquisa, considerada como sinônimo de busca, de investigação e indagação.

### 3.2 ETAPAS

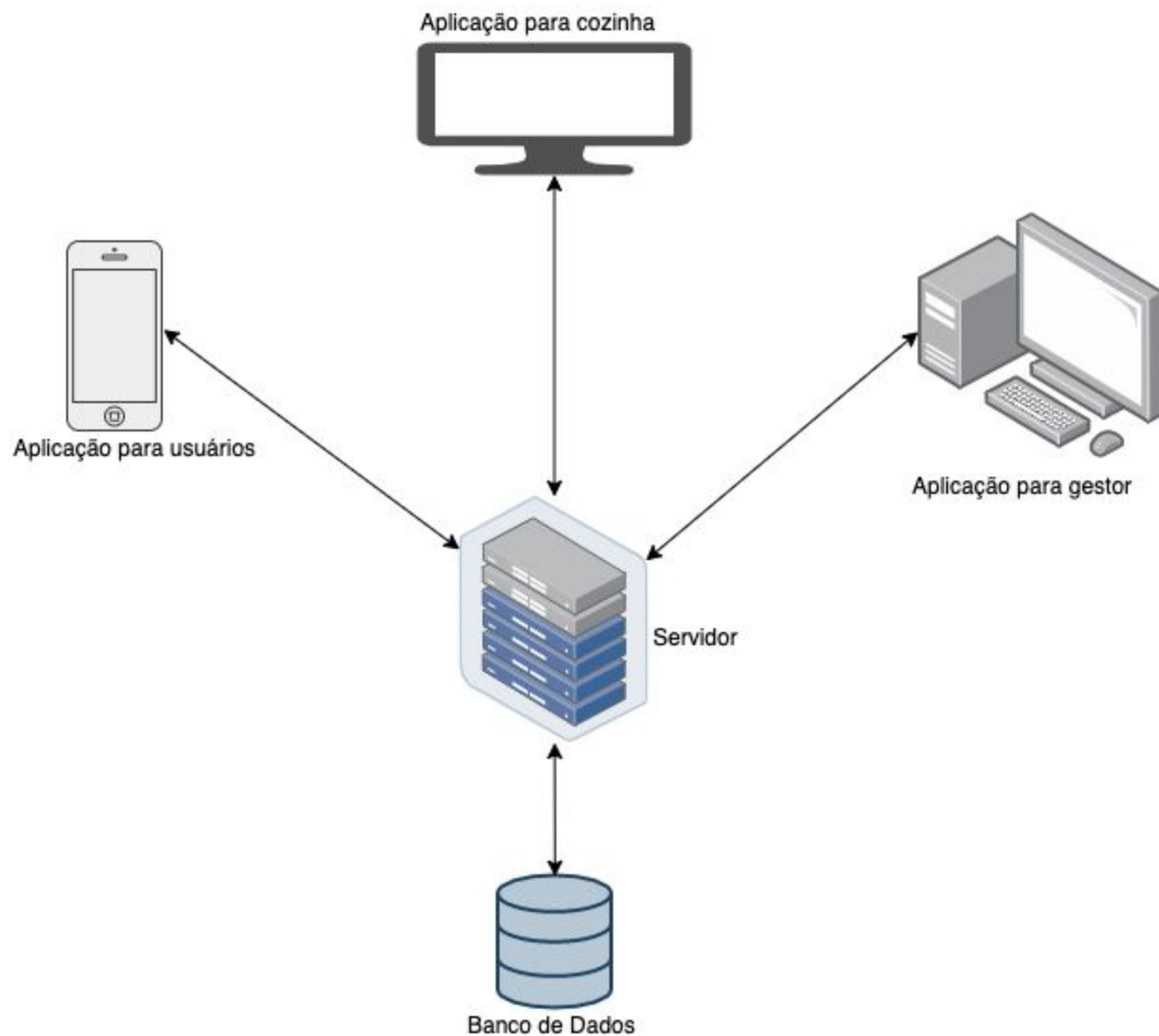
As etapas para desenvolver uma solução para o problema proposto são:

1. Levantamento de informações: Buscar conhecimento necessário para o desenvolvimento do sistema, técnicas, ferramentas, necessidades e ambientes em que deve funcionar.
2. Estudar tecnologias de desenvolvimento: Aprofundar os conhecimentos nas diferentes tecnologias que serão utilizadas para a construção do sistema.
3. Levantamento de requisitos: Definir os requisitos funcionais e não funcionais do sistema.
4. Modelagem do sistema: Modelar cada entidade e suas interações no sistema.
5. Prototipação: Criação de protótipos com os requisitos já definidos.
6. Testes: Testes manuais para verificar erros e possíveis melhorias.
7. Avaliação: Avaliação do produto final e de desenvolvimento futuro.

### 3.3 DESENHO DA SOLUÇÃO PROPOSTA

A figura a seguir demonstra uma visão macro da arquitetura do sistema.

Figura 6 - Arquitetura do sistema.



Fonte: Elaboração do Autor, 2019.

O sistema terá três frentes: a aplicação para o usuário que consistirá num PWA, a aplicação para cozinha que será uma simples listagem dos pedidos abertos, e o site da aplicação de gerenciamento onde o gestor poderá analisar compras efetuadas através do sistema, alterar itens do cardápio, etc. Todas as frentes do sistema utilizarão o mesmo servidor, assim tendo uma base de dados única e integrada.

### 3.4 AVALIAÇÃO DO SISTEMA

Este trabalho trata do desenvolvimento de um protótipo funcional, portanto, inicialmente só conterá as funcionalidades essenciais. Para avaliar estas funcionalidades, o software será entregue para pelos menos duas pessoas para avaliar a interface do cliente.

### 3.5 DELIMITAÇÕES

O desenvolvimento do sistema aqui descrito é limitado às funcionalidades essenciais para funcionamento e avaliação do sistema, sendo assim, a versão inicial não contará com os seguintes itens:

- Sistema de controle de estoque.
- Emissão de nota fiscal.
- Busca por restaurantes similares.
- Sistema de pagamento.
- Recomendação de pratos a partir de histórico de pedidos do usuário.
- Avaliação de pratos.

## 4 MODELAGEM

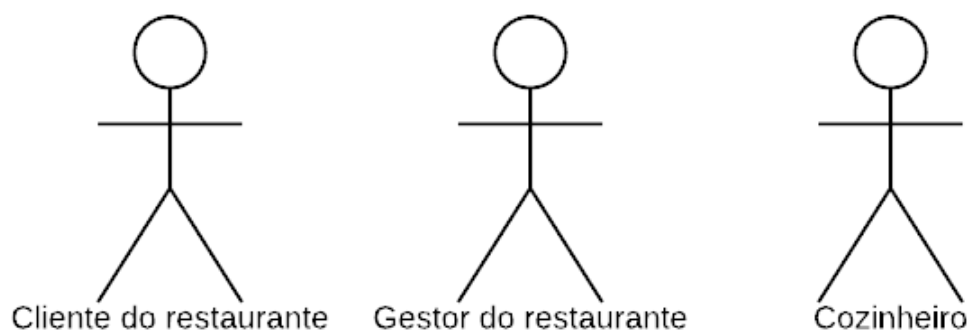
Esta seção do trabalho definirá a modelagem do sistema, que consistirá das imagens de prototipação das telas, assim como técnicas de engenharia de software como definição de atores, requisitos funcionais e não funcionais, e diagramas de casos de uso, da linguagem UML, que, conforme Booch, Rumbaugh, e Jacobson (2006, p. 13), é:

A UML (Unified Modeling Language) é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software.

### 4.1 ATORES

Os atores são os devidos papéis que cada usuário terá no sistema. A solução proposta será composta por três sistemas, cada um com um tipo de usuário específico.

Figura 7 - Atores do sistema.



Fonte: Elaboração do autor, 2020.

Nos quadros a seguir será detalhada a definição de cada um destes atores.

Quadro 3 - Definição do ator cliente do restaurante.

<b>Ator</b>	Cliente do restaurante.
<b>Definição</b>	O cliente é o usuário final, aquele que será servido pelo restaurante. Utilizará o aplicativo para realizar pedidos e pagamentos.
<b>Frequência de uso</b>	Variável.
<b>Permissão de acesso</b>	Acesso ao cardápio do restaurante em que se encontra, assim como à realização de pedidos e pagamentos.

Fonte: Elaboração do autor, 2020.

Quadro 4 - Definição do do ator gestor do restaurante.

<b>Ator</b>	Gestor do restaurante.
<b>Definição</b>	É responsável por administrar os itens do cardápio através da interface de gestor do sistema. Pode também fazer uso dos gráficos no painel do restaurante para tomar decisões de negócio.
<b>Frequência de uso</b>	Diária.
<b>Permissão de acesso</b>	Acesso de escrita e leitura à informações do restaurante, e de leitura à pedidos e pagamentos.

Fonte: Elaboração do autor, 2020.



Quadro 5 - Definição do ator cozinheiro.

<b>Ator</b>	Cozinheiro.
<b>Definição</b>	Responsável por executar os pedidos feitos por clientes através do aplicativo. Faz isso através da interface de cozinha do sistema.
<b>Frequência de uso</b>	Diária.
<b>Permissão de acesso</b>	Acesso de escrita ao estado do pedido e a leitura a pedidos.

Fonte: Elaboração do autor, 2020.

## 4.2 REQUISITOS

Segundo Pressman (2008, p. 132) "A engenharia de requisitos estabelece uma base sólida para o projeto e para a construção. Sem ela, o software resultante tem grande probabilidade de não atender às necessidades do cliente.". Portanto, a seguir será definido os requisitos funcionais específicos para cada um dos três sistemas, e após, os requisitos não funcionais, que serão válidos para todos os sistemas.

### 4.2.1 Requisitos Funcionais

A seguir no quadro 6 são apresentados os requisitos funcionais da aplicação do cliente do restaurante.

Quadro 6 - Requisitos funcionais da interface do cliente.

Identificação	Nome	Descrição
RF01	Logar por telefone	A autenticação da

		aplicação para o cliente deve ser feita utilizando o número de celular como identificador, e como substituição de senha, autenticação por SMS.
RF02	Deslogar	A aplicação deve oferecer a opção de deslogar.
RF03	Listar itens do cardápio	A aplicação deve listar o cardápio do restaurante atual.
RF04	Listar detalhes de item específico	A aplicação deve listar os detalhes específicos de cada item, como descrição completa e opcionais.
RF05	Adicionar item à pedido atual	A aplicação deve possibilitar a adição de itens ao pedido atual.
RF06	Remover item do pedido atual	A aplicação deve possibilitar a remoção de itens do pedido atual.
RF07	Mostrar detalhes do pedido atual	A aplicação deve possibilitar mostrar todos os itens do pedido atual,

		e detalhes como preço, etc.
RF08	Pedir pedido atual	A aplicação deve possibilitar a concretização do pedido.
RF09	Mostrar detalhes da conta atual (todos pedidos já feitos)	A aplicação deve possibilitar mostrar a conta atual, que contém todos os pedidos já feitos.
RF10	Realizar pagamento	A aplicação deve possibilitar ao usuário realizar o pagamento de sua conta.
RF11	Abrir conta em mesa por QR Code	A aplicação deve possibilitar que o usuário abra uma conta escaneando um QRCode.
RF12	Abrir conta por código da mesa	A aplicação deve possibilitar que o usuário abra uma conta inserindo o código da mesa.

Fonte: Elaboração do autor, 2020.

A seguir no quadro 7 são apresentados os requisitos funcionais da aplicação do gestor do restaurante.

Quadro 7 - Requisitos funcionais da interface do gestor do restaurante.

Identificação	Nome	Descrição
RF13	Logar por email	Para a autenticação da interface do gestor, a aplicação deve possibilitar o login por email.
RF14	Deslogar	A aplicação deve oferecer a opção de deslogar.
RF15	Listar itens do cardápio	A aplicação deve listar o cardápio do restaurante atual.
RF16	Mostrar detalhes de item específico	A aplicação deve listar os detalhes específicos de cada item, como descrição completa e opcionais.
RF17	Adicionar itens ao cardápio	A aplicação deve possibilitar a adição de itens ao cardápio do restaurante.
RF18	Editar um item do cardápio	A aplicação deve possibilitar a edição de itens existentes.
RF19	Remover um item do cardápio	A aplicação deve possibilitar a remoção de itens do cardápio do restaurante.
RF20	Listar histórico de contas	A aplicação deve listar o histórico de contas de aquisições no restaurante.

RF21	Editar detalhes do restaurante	A aplicação deve possibilitar a edição de detalhes do restaurante, como nome ou tipo de culinária.
RF22	Mostrar horários de pico	A aplicação deve mostrar um gráfico contendo os horários com a frequência de clientes ativos.
RF23	Mostrar pratos mais pedidos	A aplicação deve mostrar um gráfico contendo o número de pedidos feitos em cada item do cardápio.
RF24	Mostrar faturamento por período	A aplicação deve mostrar a soma do pagamento de todas as contas em um período selecionado pelo usuário.

Fonte: Elaboração do autor, 2020.

A seguir no quadro 8 são apresentados os requisitos funcionais da aplicação da cozinha.

Quadro 8 - Requisitos funcionais da aplicação da cozinha.

Identificação	Nome	Descrição
RF25	Logar por email	Para a autenticação da interface do cozinheiro, a aplicação deve possibilitar o login por email.

RF26	Deslogar	A aplicação deve oferecer a opção de deslogar.
RF27	Listar pedidos abertos	A aplicação deve listar os pedidos em aberto.
RF28	Alterar estado de um pedido	A aplicação deve permitir ao usuário alterar o estado de um pedido (aberto, em preparação, ou concluído).

Fonte: Elaboração do autor, 2020.

#### 4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais não representam funcionalidades específicas do sistema, mas sim, definições de como o sistema deve ser desenvolvido ou se comportar em determinada situação.

Quadro 9 - Requisitos não funcionais.

Identificação	Nome
RNF01	A aplicação deve ser no formato de um PWA.
RNF02	A aplicação deve ser totalmente utilizável em dispositivos móveis.
RNF03	As interfaces da aplicação devem utilizar o protocolo HTTPS.
RNF04	A aplicação deve permitir pagamentos com cartão de crédito.

Fonte: Elaboração do autor, 2020.

### 4.3 CASOS DE USO

Segundo Booch, Rumbaugh, e Jacobson (2006, p. 217):

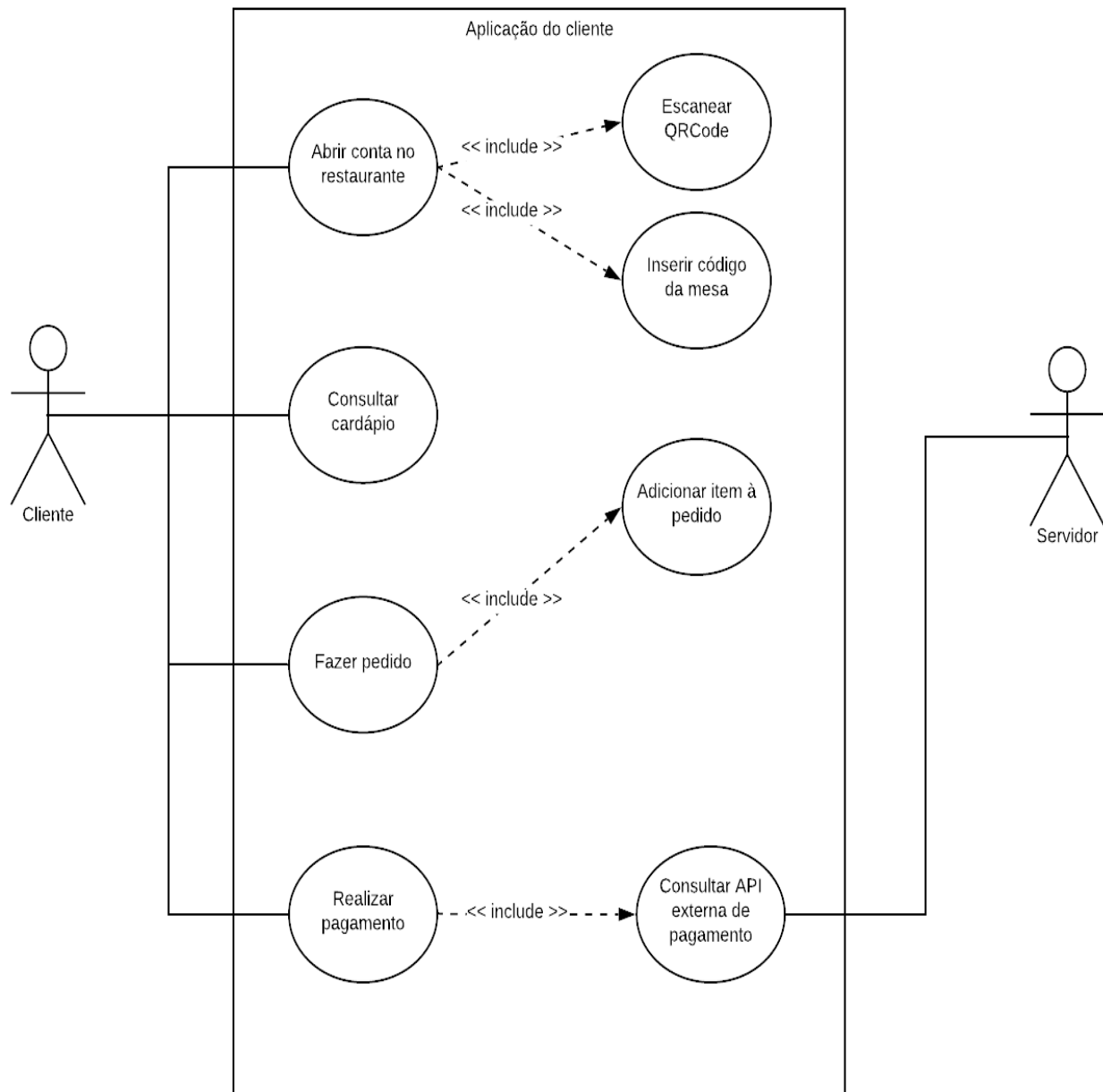
Um caso de uso especifica o comportamento de um sistema ou de uma parte de um sistema e é uma descrição de um conjunto de sequências de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.

Aqui, novamente, os casos de uso serão divididos para cada uma das aplicações específicas, do cliente, do gestor e do cozinheiro.

### 4.3.1 Casos de uso da aplicação do cliente

A seguir é apresentado os casos de uso da aplicação do cliente.

Figura 8 - Casos de uso da aplicação do cliente.



Fonte: Elaboração do autor, 2020.



#### 4.3.1.1 Descrição dos casos de uso

##### 4.3.1.1.1 Caso de uso de abrir conta no restaurante

<b>Nome</b> Abrir conta no restaurante
<b>Descrição</b> A abertura de conta será feita através da leitura de um QRCode do código presente na mesa. As contas serão individuais por usuário.
<b>Ator</b> Cliente
<b>Requisitos associados</b> <ul style="list-style-type: none"><li>• RF11</li><li>• RF12</li></ul>
<b>Pré-condições</b> <ul style="list-style-type: none"><li>• O dispositivo deve ter uma camera para escanear o QR Code</li><li>• O usuário deve estar autenticado</li></ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"><li>1. O usuário se autentica utilizando o número de seu celular</li><li>2. O usuário escaneia o QR Code da mesa ou insere o código da mesa</li><li>3. O usuário é direcionado ao cardápio do restaurante e pode fazer pedidos.</li></ol>

##### 4.3.1.1.2 Caso de uso de consultar cardápio

<b>Nome</b> Consultar cardápio
<b>Descrição</b> O usuário navega pela listagem de pratos do restaurante, podendo expandir itens para ver mais detalhes.
<b>Ator</b> Cliente
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF03</li> <li>• RF04</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> <li>• O usuário deve ter uma conta em aberto no restaurante</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>1. O usuário segue o fluxo de abertura de conta</li> <li>2. O usuário navega pelo cardápio</li> </ol>

#### 4.3.1.1.3 Caso de uso de fazer pedido

<b>Nome</b> Fazer pedido
<b>Descrição</b> O usuário adiciona ou remove itens do pedido, e depois concretiza o pedido.
<b>Ator</b>

Cliente
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF05</li> <li>• RF06</li> <li>• RF07</li> <li>• RF08</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> <li>• O usuário deve ter uma conta em aberto no restaurante</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>1. O usuário segue o fluxo de abertura de conta</li> <li>2. O usuário adiciona um item ao pedido</li> <li>3. O usuario faz o pedido</li> </ol>

#### 4.3.1.1.4 Caso de uso de realizar pagamento

<b>Nome</b> Realizar Pagamento
<b>Descrição</b> Após fazer os pedidos, o usuário deve realizar o pagamento utilizando um cartão bancário.
<b>Ator</b> Cliente
<b>Requisitos associados</b>

- RF09
- RF10

**Pré-condições**

- O usuário deve estar autenticado
- O usuário deve ter uma conta em aberto no restaurante

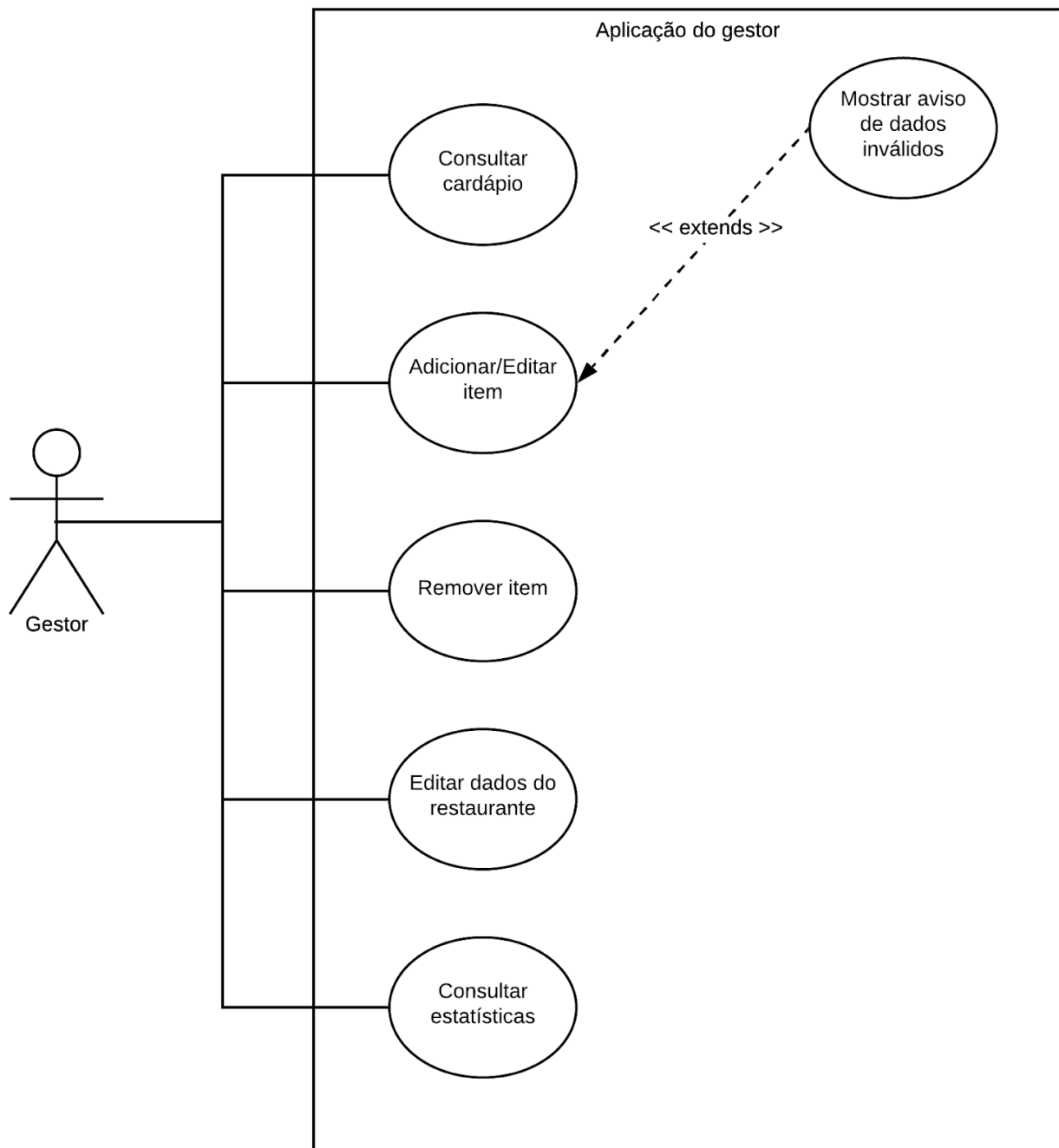
**Fluxo normal**

1. O usuário segue o fluxo de abertura de conta
2. O usuário expande a fatura atual, contendo todos pedidos já feitos
3. O usuário realiza o pagamento

### 4.3.2 Casos de uso da aplicação do gestor

A seguir é apresentado os casos de uso da aplicação do gestor.

Figura 9 - Casos de uso da aplicação do gestor.



Fonte: Elaboração do autor, 2020.

#### 4.3.2.1 Descrição dos casos de uso

##### 4.3.2.1.1 Caso de uso de consultar cardápio

<b>Nome</b> Consultar cardápio
<b>Descrição</b> O gestor navega pelo cardápio do seu restaurante
<b>Ator</b> Gestor
<b>Requisitos associados</b> <ul style="list-style-type: none"><li>• RF15</li><li>• RF16</li></ul>
<b>Pré-condições</b> <ul style="list-style-type: none"><li>• O usuário deve estar autenticado</li></ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"><li>3. O usuário se autentica</li><li>4. O usuário navega pelo cardápio</li></ol>

##### 4.3.2.1.2 Caso de uso de adicionar ou editar item

<b>Nome</b> Adicionar/Editar item
<b>Descrição</b>

O usuário adiciona ou edita itens do cardápio do seu restaurante.
<b>Ator</b> Gestor
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF16</li> <li>• RF17</li> <li>• RF18</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>4. O usuário se autentica</li> <li>5. O usuário adiciona ou edita itens</li> </ol>

#### 4.3.2.1.3 Caso de uso de remover item

<b>Nome</b> Remover item
<b>Descrição</b> O usuário remove itens do cardápio do seu restaurante.
<b>Ator</b> Gestor
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF19</li> </ul>

<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> <li>• O usuário deve ter itens adicionados ao cardápio</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>4. O usuário se autentica</li> <li>5. O usuario remove itens</li> </ol>

#### 4.3.2.1.3 Caso de uso de editar detalhes do restaurante

<b>Nome</b> Editar detalhes do restaurante
<b>Descrição</b> O gestor pode mudar informações do restaurante como nome, tipo de culinária, horário de funcionamento, entre outros.
<b>Ator</b> Gestor
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF21</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>6. O usuário se autentica</li> <li>7. O usuário edita informações do restaurante</li> </ol>



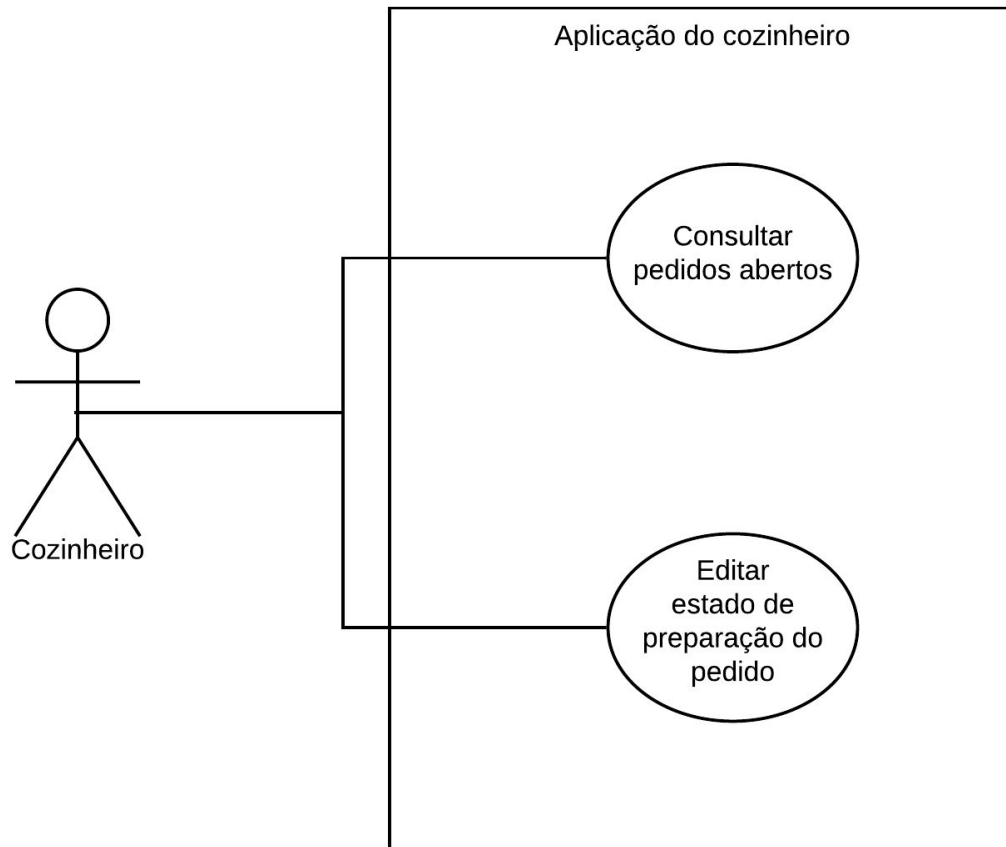
#### 4.3.2.1.4 Caso de uso de consultar estatísticas

<b>Nome</b> Consultar estatísticas
<b>Descrição</b> O gestor do restaurante pode consultar as estatísticas como gráfico de frequência de clientes, pratos mais pedidos, entre outros.
<b>Ator</b> Gestor
<b>Requisitos associados</b> <ul style="list-style-type: none"><li>• RF20</li><li>• RF22</li><li>• RF23</li></ul>
<b>Pré-condições</b> <ul style="list-style-type: none"><li>• O usuário deve estar autenticado</li></ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"><li>5. O usuário se autentica</li><li>6. O usuário consulta as estatísticas</li></ol>

### 4.3.3 Casos de uso da aplicação do cozinheiro

A seguir é apresentado os casos de uso da aplicação da cozinha.

Figura 10 - Casos de uso da aplicação do gestor.



Fonte: Elaboração do autor, 2020.

#### 4.3.3.1 Descrição dos casos de uso

##### 4.3.3.1.1 Caso de uso de consultar pedidos abertos

<b>Nome</b> Consultar pedidos abertos
<b>Descrição</b> A aplicação do cozinheiro consistirá basicamente de uma lista de pedidos em abertos cujo cozinheiro deve alterar o estado de preparação.
<b>Ator</b> Cozinheiro
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF27</li> <li>• RF28</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>7. O usuário se autentica</li> <li>8. O usuário consulta os pedidos</li> </ol>

#### 4.3.3.1.2 Caso de uso de editar estado de preparação do pedido

<b>Nome</b> Editar estado de preparação do pedido
<b>Descrição</b> O cozinheiro altera o estado para "em preparação" ou "concluído"
<b>Ator</b>

Cozinheiro
<b>Requisitos associados</b> <ul style="list-style-type: none"> <li>• RF28</li> </ul>
<b>Pré-condições</b> <ul style="list-style-type: none"> <li>• O usuário deve estar autenticado</li> </ul>
<b>Fluxo normal</b> <ol style="list-style-type: none"> <li>8. O usuário se autentica</li> <li>9. O usuário altera o estado do pedido</li> </ol>

## 4.4 PROTOTIPAÇÃO DAS TELAS

Nesta seção será exibida os protótipos de cada tela da aplicação do cliente, do gestor e do cozinheiro.

### 4.4.1 Telas da aplicação do cliente

Como dito, a aplicação do cliente consistirá de uma aplicação web focada em dispositivos móveis, a qual utilizando novas tecnologias, permitirá que o usuário instale o app em seu celular.

Na figura 11 temos a tela inicial, a de autenticação. Para diminuir a barreira de entrada da aplicação, foi utilizado o sistema de autenticação por SMS, em que o usuário digita o seu número de celular e recebe um código por SMS que deve ser inserido para confirmar sua identidade. Este processo tem o benefício de simplificar o cadastro e de não ser necessário lembrar de uma senha por parte do usuário.

Figura 11: Tela de autenticação




A tela de autenticação do aplicativo MENU apresenta o seguinte layout:

- Logo **MENU** em laranja no topo.
- Título: **Qual o seu número de celular?**
- Texto explicativo: "Você receberá um código de acesso. Seus dados ficarão seguros e não é necessário lembrar de uma senha :)"
- Forma de entrada de dados com campos separados por traços: **+55**, **48**, **991321616**.
- Botão laranja centralizado com o texto **CONFIRMAR**.

Fonte: Elaboração do autor, 2020.

A seguir, na figura 12, temos a tela de Home, inicial a partir do momento em que o usuário está autenticado. A tela é composta de duas opções, um botão para abrir a câmera e escanear o QR Code da mesa, ou um campo para o usuário inserir o código da mesa.

Figura 12: Tela Home



# Bem-vindo!

Para começar, insira o código de sua mesa  
ou escaneie o QR Code

ESCANEAR QR CODE

---

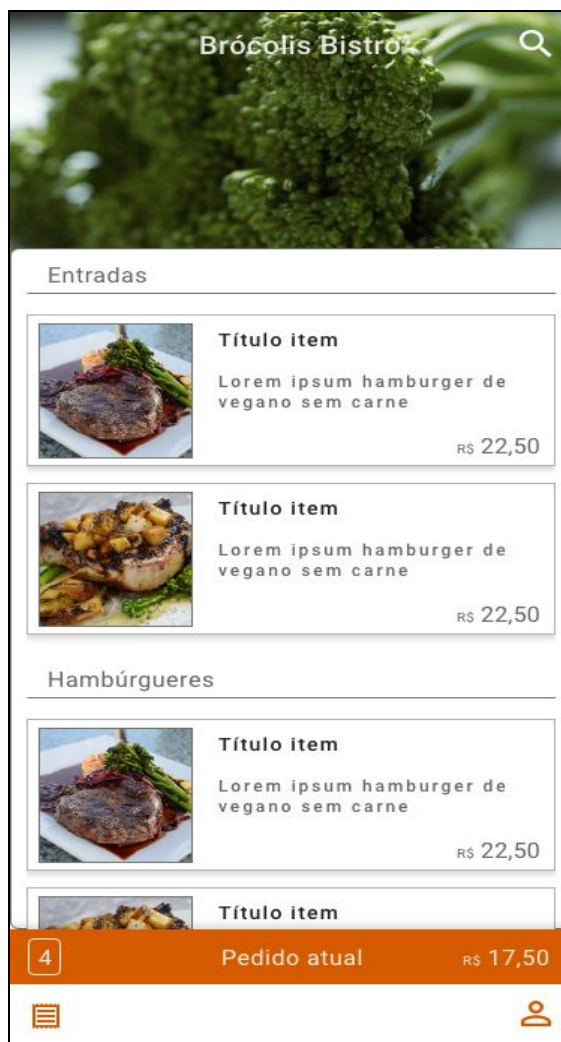
Código

QD4132

Fonte: Elaboração do autor

Ao inserir o código, é aberta uma sessão no restaurante, ou uma conta, o usuário é direcionado para o cardápio e pode fazer pedidos.

Figura 13: Tela do cardápio



Fonte: Elaboração do autor



Cada prato na listagem do cardápio é um item clicável que direciona para a tela de detalhes do item, como ilustrado na figura 14. Nesta tela o usuário pode ter uma visão melhor da imagem do prato, assim como selecionar os opcionais desejados.

Figura 14: Tela de detalhes do item



Fonte: Elaboração do autor, 2020.

A seguir, temos a tela do pedido atual, na figura 15. Nesta telas o usuário pode confirmar os pratos selecionados com seus respectivos opcionais, e então, fazer o pedido de fato.

Figura 15: Tela de pedido atual

← **PEDIDO ATUAL**

<b>1x Carne com brócolis</b>	<b>R\$ 54,50</b>	⋮
- 1x Ovo Observação: sem brócolis		
<hr/>		
<b>1x Carne com brócolis</b>	<b>R\$ 54,50</b>	⋮
- 1x Ovo Observação: sem brócolis		
<hr/>		
<b>1x Carne com brócolis</b>	<b>R\$ 54,50</b>	⋮
- 1x Ovo Observação: sem brócolis		
<hr/>		
<b>Total:</b>	<b>R\$ 150,00</b>	

**FAZER PEDIDO**

Fonte: Elaboração do autor, 2020.

Por fim, a tela de fechar conta, figura 16, é onde o usuário verifica todos os pedidos feitos, verifica o preço, e efetua o pagamento de sua conta utilizando um cartão bancário.

Figura 16: Tela de pedido atual

FECHAR CONTA

Pedido 01

1x Carne com brócolis

R\$ 54,50

- 1x Ovo

Observação: sem brócolis

1x Carne com brócolis

R\$ 54,50

- 1x Ovo

Observação: sem brócolis

Total pedido:

R\$ 100,00

Pedido 01

1x Carne com brócolis

R\$ 54,50

- 1x Ovo

Observação: sem brócolis

1x Carne com brócolis

R\$ 54,50

- 1x Ovo

Observação: sem brócolis

Total pedido:

R\$ 100,00

Total conta:

R\$ 150,00

Pagamento

Cartão

MASTERCARD \*\*\*\* 3451

Trocar

Débito

☒

Crédito

☐

CPF/CNPJ na nota

Adicionar

FAZER PEDIDO

Fonte: Elaboração do autor, 2020.

#### 4.4.2 Telas da aplicação do gestor

A aplicação do gestor, por sua vez, será focada em gerenciar o cardápio e demais informações do restaurante, como horário de funcionamento e tipo de culinária, assim como analisar estatísticas de funcionamento.

Pelo fato dos principais usuários dessa interface serem os gerentes e o dono do estabelecimento, supõe-se que eles farão mais uso da aplicação em um computador do tipo desktop. Portanto, a interface será desenvolvida com foco na resolução comum entre estes tipos de dispositivos.

A tela inicial, figura 17, é a de informações do restaurante, onde aparecerá todos detalhes do estabelecimento e um botão de edição, no canto superior direito.

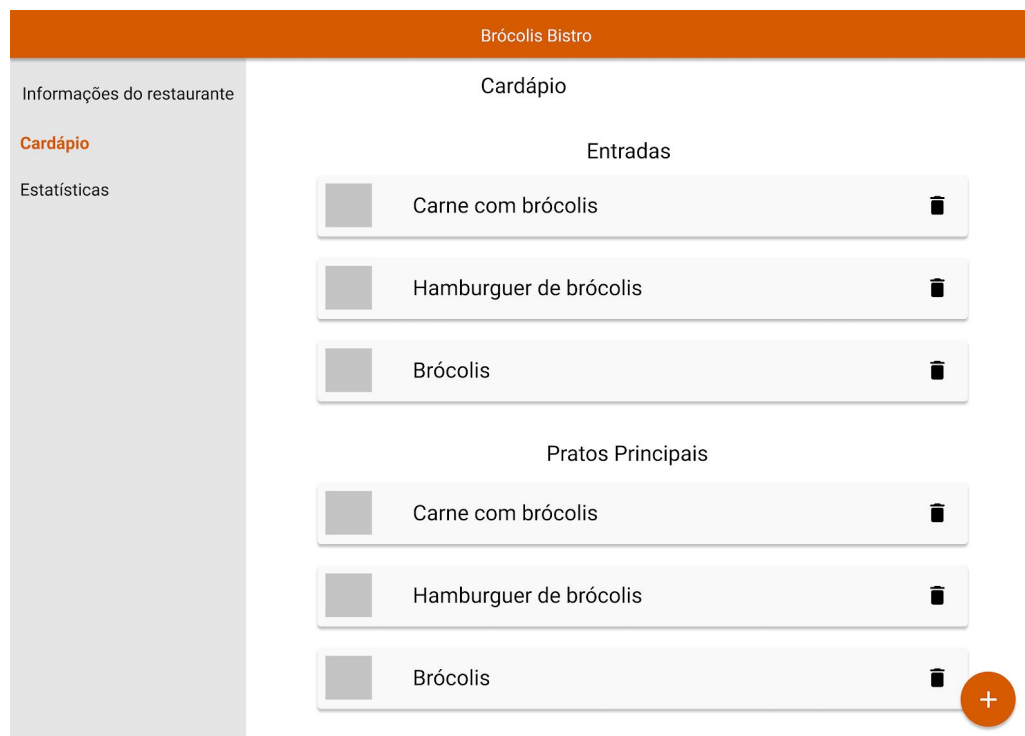
Figura 17: Tela inicial da aplicação do gestor

Brócolis Bistro	
Informações do restaurante	Informações do Restaurante
Cardápio	
Estatísticas	
Nome	Brócolis Bistro
Tipo de culinária	Brócolis
Endereço	Floripa
Lotação	15
Imagem de fundo	brócolis.png

Fonte: Elaboração do autor, 2020.

A seguir, na figura 18, temos a tela do cardápio, onde o usuário pode ver a lista de itens disponíveis no restaurante e adicionar ou remover pratos.

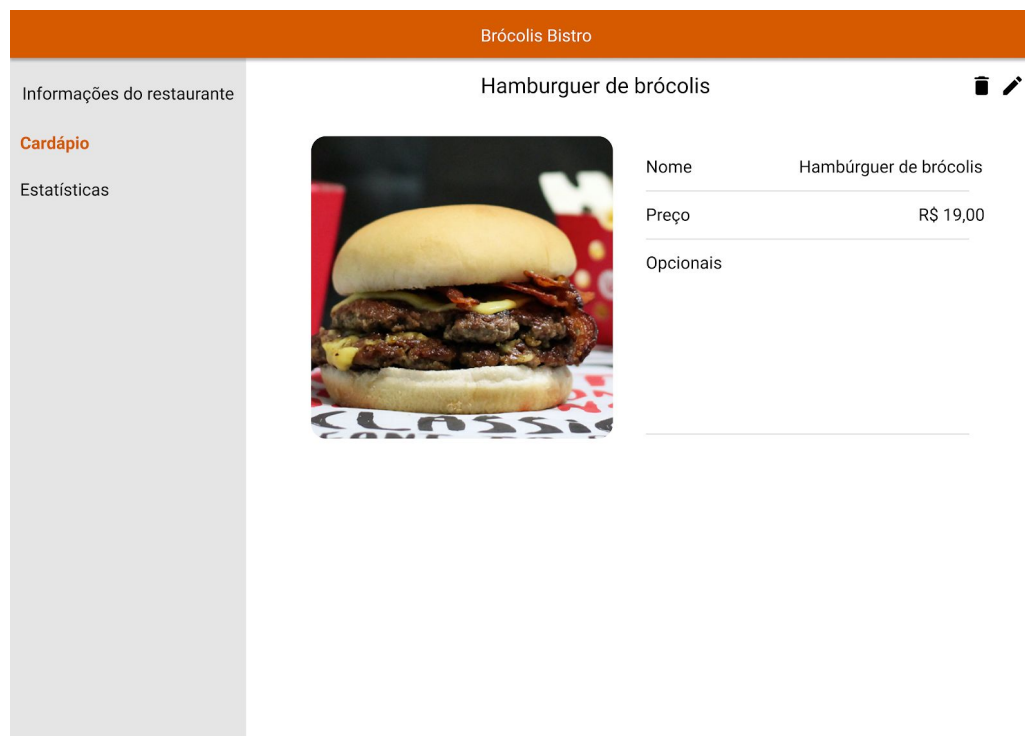
Figura 18: Tela de cardápio da aplicação do gestor.



Fonte: Elaboração do autor, 2020.

Ao clicar em algum dos itens na listagem do cardápio, o usuário é direcionado para a tela do perfil daquele item (figura 19), e pode então realizar alterações ou removê-lo.

Figura 19: Tela de perfil de item da aplicação do gestor.



Fonte: Elaboração do autor, 2020.

Por fim, temos a tela de estatísticas, na figura 20.

Figura 20: Tela de estatísticas da aplicação do gestor.



Fonte: Elaboração do autor, 2020.

#### **4.4.3 Telas da aplicação da cozinha**

A aplicação da cozinha é a mais simples das três pois consiste somente da listagem dos pedidos em abertos no restaurante e da ação de mudar o estado do pedido para em preparo ou concluído.

A fim de facilitar o processo da cozinha, foi dado um foco maior em clareza e distinção das informações nessa interface, fazendo uso de cores chamativas, fontes maiores e ações mais expressivas. Na figura 21, temos a tela da aplicação contendo a listagem dos pedidos. O primeiro pedido é o que está sendo preparado, indicado pela cor verde, e para mudar o estado do pedido, o usuário deve realizar a ação de puxar o pedido da esquerda para a direita.



Figura 21: Tela da aplicação da cozinha.

Brócolis Bistro - Cozinha

Mesa 12

Pedido #14

3x Hamburguer de brócolis

- Sem queijo

2x Coca-cola

2x Carne com brócolis

- Sem pimenta

Mesa 12

Pedido #14

3x Hamburguer de brócolis

- Sem queijo

2x Coca-cola

2x Carne com brócolis

- Sem pimenta

Mesa 12

Pedido #14

3x Hamburguer de brócolis

- Sem queijo

Fonte: Elaboração do autor, 2020.

## 5 DESENVOLVIMENTO

Esta seção descreve cada uma das tecnologias utilizadas para o desenvolvimento do projeto, porque foram escolhidas e como foram utilizadas. A seguir, a figura 22 ilustra cada uma dessas ferramentas.

### 5.1 TECNOLOGÍAS E FERRAMENTAS

Para este projeto foram escolhidas tecnologias modernas e que façam sentido para o produto sendo desenvolvido.

Figura 22: Tecnologías utilizadas



Fonte: Elaboração do autor, 2020.

### 5.1.1 TypeScript

A linguagem JavaScript surgiu como uma simples linguagem para manipular elementos HTML em websites, porém, ao decorrer do tempo foi ganhando popularidade e adaptada para rodar em diversos ambientes além do navegador. Hoje é uma das linguagens mais utilizadas, sendo encontrada em projetos de todos tamanhos e em diferentes dispositivos, incluindo até o foguete espacial Dragon, da SpaceX (HNAIDE, 2020).

Porém, provavelmente por conta de seu objetivo inicial, a linguagem JavaScript tem uma natureza dinâmica e alguns comportamentos surpreendentes para programadores inexperientes com a linguagem, e isto faz com que projetos de larga escala que a utilizem tendem a ser complexos e difíceis de ser administrados.

Tendo isso em vista, a empresa Microsoft criou a linguagem TypeScript (2020), que fornece a adição de tipagem estática para JavaScript, entre outras adições de sintaxe. Isto ajuda no processo de desenvolvimento pois é capaz de identificar diversos erros em tempo de compilação além demonstrar várias informações sobre métodos e bibliotecas diretamente na IDE do programador. Além disso, a linguagem Typescript é considerada um superset de JavaScript, ou seja, todo código JavaScript é válido para TypeScript, mas o oposto não é verdade. Para utilizar essa ferramenta, o código deve ser compilado para JavaScript para só depois ser interpretado de fato.

### 5.1.2 Firebase

Firebase (2020) é um serviço da Google que disponibiliza diversas ferramentas para serem utilizadas de forma automática sem que o desenvolvedor precise implementar um backend específico para isso. Possui diversos produtos como:

- Banco de dados NoSQL em tempo real.

- Autenticação.
- Computação em nuvem.
- Machine Learning.
- Armazenamento.
- Analytics.

Neste projeto serão utilizados o banco de dados em tempo real, especificamente o Firestore, e o serviço de autenticação.

### 5.1.3 Preact

Preact (2020) é uma alternativa à biblioteca React que disponibiliza a mesma API porém possui um tamanho de pacote menor, melhorando a performance do site em conexões mais lentas.

Em contrapartida, a biblioteca React é definida por (React, 2020, tradução nossa):

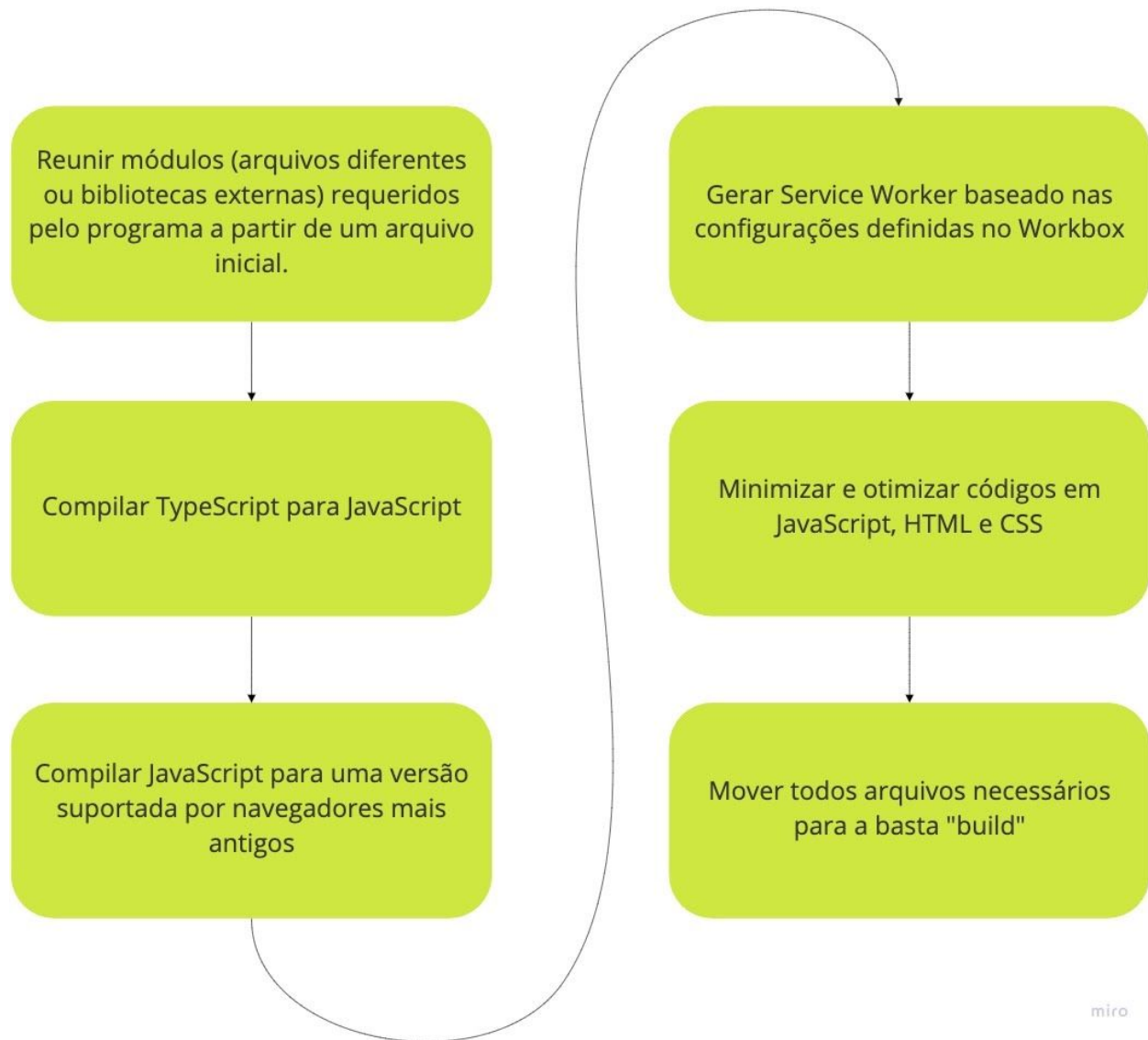
React é uma biblioteca JavaScript para construir interfaces de usuário de forma declarativa, eficiente e flexível. Ela permite que você constitua UIs complexas a partir de pequenos e isolados pedaços de código chamados de "componentes".

### 5.1.4 Webpack

O desenvolvimento web moderno é composto por variados recursos e ferramentas e deve disponibilizar o produto final para diversos navegadores que interpretam as tecnologias como JavaScript ou HTML de formas moderadamente diferentes, além de usuários com conexão de internet lenta. Para administrar isso, geralmente se faz uso de uma ferramenta chamada de *bundler*, que é o caso do Webpack (2020). Estas ferramentas criam um fluxo de execução de várias outras ferramentas utilizadas para otimizar o desenvolvimento assim como o produto final.

Utilizado o trabalho em questão como exemplo, o fluxo de execução é, de forma aproximada e simplificada, o seguinte:

Figura 23: Fluxo do Webpack.



Fonte: Elaboração do autor, 2020.

### 5.1.5 Material UI

Material UI (2020) é uma biblioteca de componentes React que seguem o padrão definidos pelo Material Design, que segundo o site (Material, 2020, tradução

própria, 2020) "é uma linguagem visual que sintetiza os princípios clássicos do bom design com a inovação da tecnologia e ciência".

### **5.1.6 Workbox**

Um dos requerimentos para que um site seja considerado um PWA, e por consequência possa ser instalado no dispositivo do usuário, é que o site disponibilize um Service Worker, que consistem em um arquivo JavaScript que, entre outras funcionalidades, define o comportamento offline do website.

O Workbox (2020) portanto é uma biblioteca criada pela Google que auxilia na criação destes arquivos, fornecendo estratégias de cache prontas para uso em produção.

## **5.2 HISTÓRICO DE DESENVOLVIMENTO**

O desenvolvimento deste projeto consistiu em um grande processo de aprendizado, dado que o autor não tinha conhecimento em algumas das tecnologias utilizadas.

### **5.2.1 Estudos e configuração do projeto**

Houve um tempo considerável de estudos para aprender no que consiste um PWA, quais estratégias e arquiteturas de sistema são utilizados e principalmente, como implementar um service worker, que, apesar da biblioteca Workbox fazer boa parte do trabalho, foi importante aprender o que acontece "por debaixo dos panos", para ter mais controle da aplicação.

Após aprender sobre PWA, foi iniciado o desenvolvimento do projeto. Apesar de existirem opções de criação de projetos para React com configurações prontas, como o

Create React App, também do Facebook, o autor escolheu implementar a configuração por conta própria, para obter maior aprendizado e controle destas configurações.

Uma das primeiras funcionalidades a ser implementada foi o escaneador de QR Code. Neste caso também foi necessário um pouco de estudos para aprender a lidar com a API de câmera do navegador, assim como encontrar uma biblioteca leve e robusta para interpretar as informações da imagem e extrair o código do QR Code.

### **5.2.2 Implementação do Firebase**

Com o projeto configurado, o autor teve que estudar sobre o Firebase que, apesar de ser um backend automatizado, ainda possui uma API própria e um banco de dados com sistema de modelagem e pesquisa de dados específicos.

Para a autenticação da aplicação do cliente, foi utilizado o serviço do Firebase de login por telefone. Este serviço contém uma grande infraestrutura para poder fazer o envio de mensagens SMS e a verificação dos respectivos códigos. Portanto, a fim de evitar tentativas de login automatizadas e sobrecarregar esta infraestrutura, é requerido o uso do reCAPTCHA, um serviço também da Google para detecção de robôs. A implementação do reCAPTCHA neste caso é feita especificamente pela API de autenticação do Firebase e não foi encontrada documentação o suficiente para o caso de uso deste projeto, e isto conferiu uma dificuldade para completar o sistema de autenticação da aplicação do cliente.

### **5.2.3 Roteamento**

Outro ponto relevante do processo de desenvolvimento foi a implementação do sistema de roteamento do site.

Em um site convencional o roteamento das páginas é feito no servidor, mas neste caso, por se tratar de um *single page application*, ou seja, um site cujo todo código é enviado ao cliente de uma só vez, o roteamento é feito por completo no

frontend, mudando o conteúdo do site de forma dinâmica a partir do caminho atual da URL.

Existem bibliotecas específicas para o React que gerenciam esta funcionalidade por conta própria, como o React Router (2020). Porém, por se tratar de um trabalho de aprendizado, o autor escolheu implementar também o roteamento do site por conta própria.

### 5.2.4 Complexidade de dados

No meio do desenvolvimento, a estrutura de dados dos pedidos, primariamente por causa dos opcionais de cada item, começou a se demonstrar relativamente complexa, como ilustrado na figura 23, onde *object* é a representação de um pedido, e uma sessão no restaurante, ou uma "conta", é composta por um ou vários pedidos.

Figura 24: Estrutura dos dados de um pedido.

```
▼ object {2}
  ▼ items [1]
    ▼ 0 {5}
      ▼ item {2}
        id : dGNsnMbnbn8mX0u0tmvS
        name : Classic Bacon
        amount : 1
      ▼ selectedOptionals {2}
        ▼ Extras [1]
          ▼ 0 {2}
            name : Queijo
            price : 1.5
          ▼ Tipo do Pão {1}
            name : Rosetta
          observation : value
          price : 22.5
        orderedAt : 2020-07-02T00:15:37.760Z
```

Fonte: Elaboração do autor, 2020.



Diante deste problema, foi decidido implementar a tecnologia TypeScript no projeto, a fim tornar mais nítida a estrutura de dados com que estava sendo trabalhada e auxiliar o desenvolvimento de forma geral. Este também foi um caso que exigiu aprendizado por parte do autor, visto que já havia trabalhado com a tecnologia porém há tempo demais para que se lembrasse de algo.

### 5.3 SISTEMA DESENVOLVIDO

No capítulo 4, foi apresentado a modelagem do sistema, incluindo a descrição de todas as telas das interfaces. Portanto, esta seção descreve as telas já implementadas que possuem as funcionalidades fundamentais para a operação do sistema.

#### 5.3.1 Telas iniciais

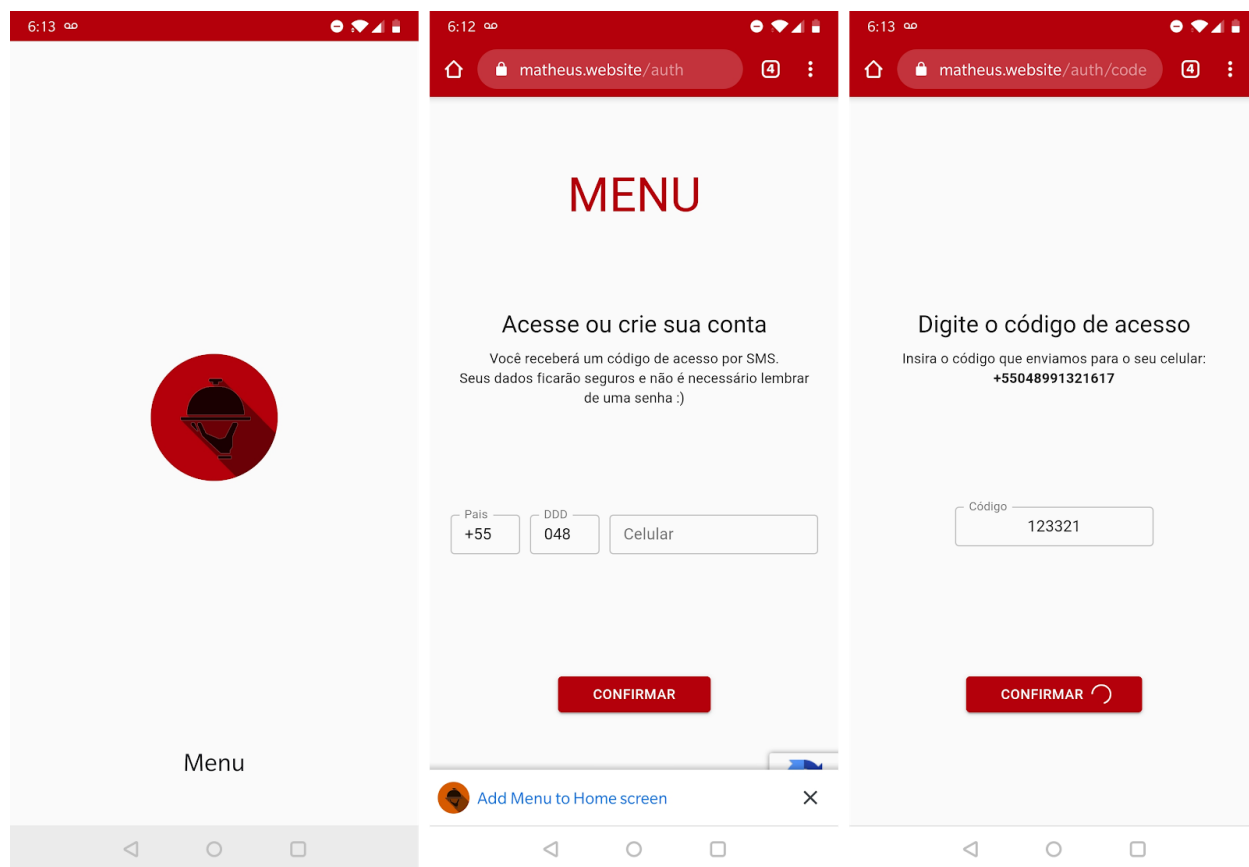
A figura 24 demonstra as telas iniciais da aplicação do cliente. A primeira tela que o usuário verá ao acessar a aplicação pela primeira vez é a tela de autenticação, a do meio, neste caso. Após inserir seu telefone, o usuário deve receber um código único que deve ser inserido na tela à direita. Assumindo que o código inserido foi o correto, o usuário é direcionado para a seção seguinte da aplicação, onde começará uma sessão no restaurante.

Pode-se observar na tela de autenticação, a do meio, que embaixo existe uma opção de "Adicionar Menu à tela inicial" (Add Menu to home screen). Esta é uma opção que é emitida automaticamente pelo navegador Chrome quando um site cumpre com os requerimentos de um PWA, permitindo assim que o site seja instalado no dispositivo do usuário.

Quando o usuário opta por instalar o site, as inicializações subsequentes da aplicação irão exibir uma tela de carregamento, chamada também de *splashscreen*,

como ilustrado na tela à esquerda da figura 24. Além disso, a aplicação rodará numa janela única do dispositivo e não será mais exibida a barra de endereço do navegador.

Figura 25: Telas iniciais da aplicação.



Fonte: Elaboração do autor, 2020.

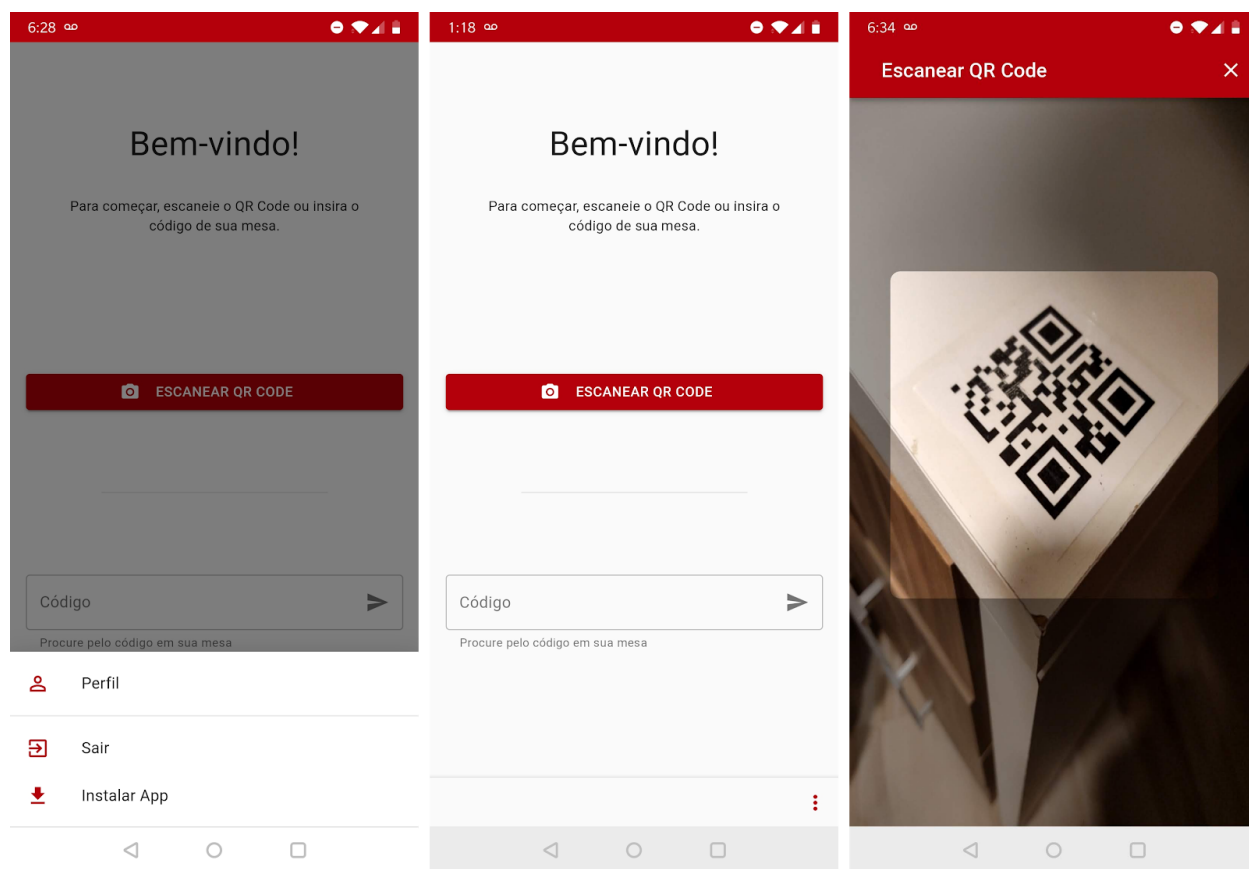
### 5.3.2 Tela Home

Caso o usuário esteja autenticado e não tenha nenhuma sessão aberta em nenhum restaurante, a tela home, da figura 25, é que será exibida. Os restaurantes que utilizarem este serviço deverão ter um código único por mesa, que deve também ser exibido através do formato de QR Code. Através deste código, os clientes poderão abrir uma sessão, utilizando as opções disponíveis na tela home.

É importante notar que, dado a unicidade dos códigos, o sistema é capaz de identificar o restaurante em que o usuário se encontra, removendo a necessidade do usuário ter que inserir em qual restaurante se encontra.

Pode-se observar também, na tela à esquerda, as opções no menu inferior, incluindo um opção para instalar a aplicação.

Figura 26: Tela home.



Fonte: Elaboração do autor, 2020.

### 5.3.3 Telas do cardápio

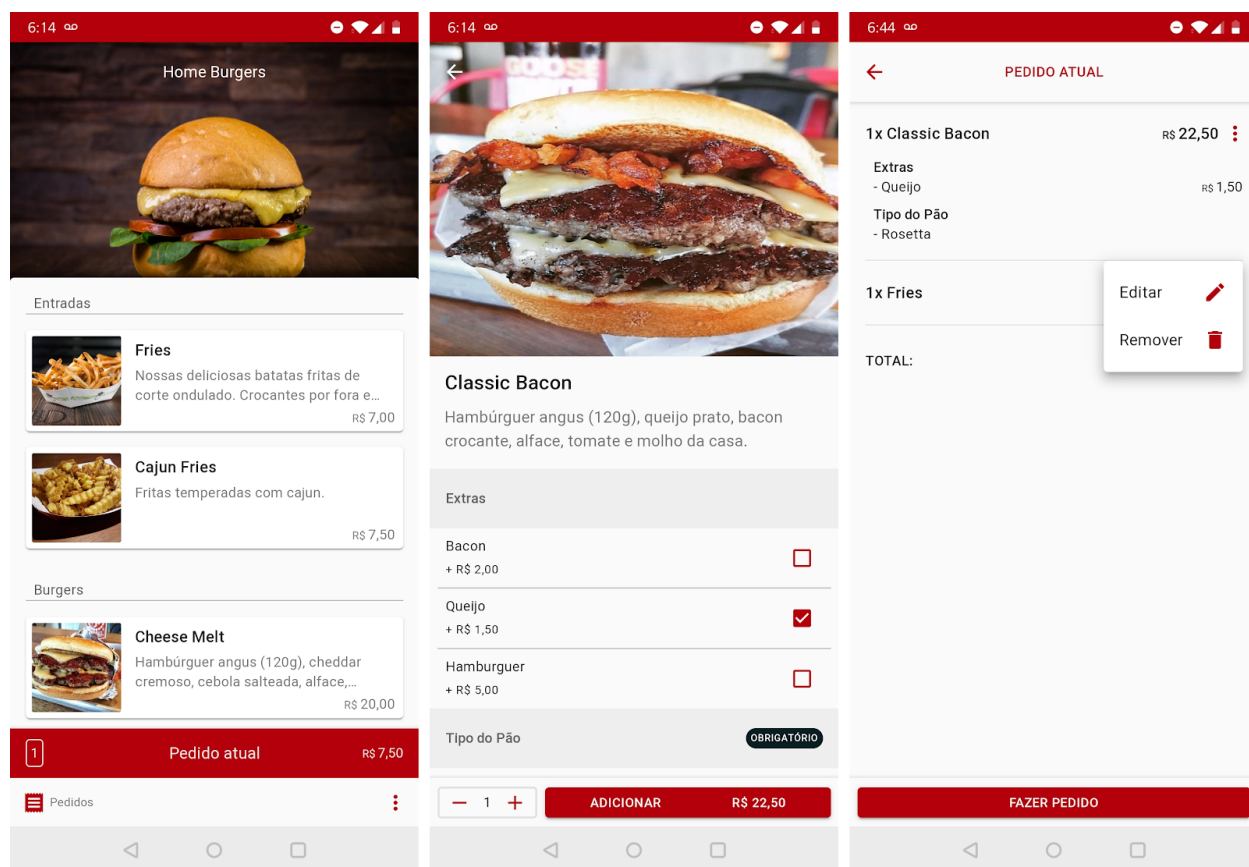
A seguir, na figura 26 à esquerda, a tela do cardápio. Esta tela consiste numa lista com todos os itens disponíveis do restaurante, além de duas barras de navegação na parte inferior da tela.

A primeira barra de navegação demonstra informações sobre o pedido atual, como quantos itens diferentes estão sendo pedidos e o preço total. A segunda barra, possui opções para ver todos pedidos já feitos, e um botão contendo mais opções, como informações de perfil de usuário, logoff, e instalar app.

No meio, a tela do perfil do item, o usuário pode ver melhor as imagens e descrição de cada item, e é apresentado com opções para customizar o seu pedido e selecionar a quantidade.

Já à direita, temos a tela do pedido atual, que lista todos os itens selecionados pelo usuário, bem como opcionais, preço de cada opcional, e preço total. Aqui o usuário tem também as opções de editar ou remover algum item.

Figura 27: Telas do cardápio.



Fonte: Elaboração do autor, 2020.

### 5.3.4 Tela de fechar conta

Por fim, a tela de fechar conta é onde o usuário confere todos os pedidos feitos em sua sessão no restaurante, incluindo cada um dos itens, quantidade e opcionais selecionados. Na figura 27 as duas imagens consistem na mesma tela, capturadas em posições verticalmente diferentes.

Abaixo da lista de pedidos há a seção de pagamento. O usuário deve selecionar um cartão de crédito já adicionado em sua conta, assim como selecionar se o pagamento será efetuado em crédito ou débito.

Figura 28: Tela de fechar conta.

6:14

← FECHAR CONTA

**Pedido 01**

1x Classic Bacon R\$ 22,50

Extras

- Queijo

Tipo do Pão

- Rosetta

---

TOTAL: R\$ 22,50

---

**TOTAL CONTA: R\$ 22,50**

---

**Pagamento**

---

**Cartão**

MASTERCARD \*\*\*\* 9028

 TROCAR

Débito ☒

Crédito ☐

**REALIZAR PAGAMENTO**

Fonte: Elaboração do autor, 2020.

## 5.2 AVALIAÇÃO DO SISTEMA

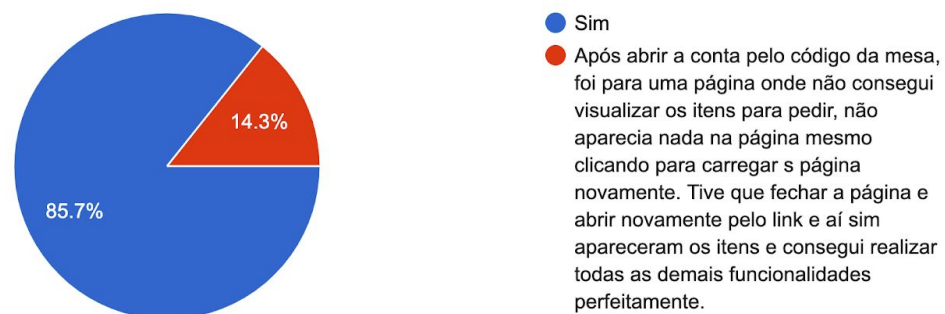
Após o término do desenvolvimento do software da interface do cliente, foi realizada uma avaliação, pedindo à algumas pessoas que seguissem o fluxo de um atendimento na aplicação e então, responder às perguntas elaboradas pelo autor.

Para um pesquisa mais confiável, foi possível obter a realizar a avaliação com 7 pessoas, cujas respostas seguem a seguir:

Gráfico 3 - Avaliação das funcionalidades da interface do cliente.

Foi possível executar todas funcionalidades listadas? Se não, qual delas não foi possível?

7 responses



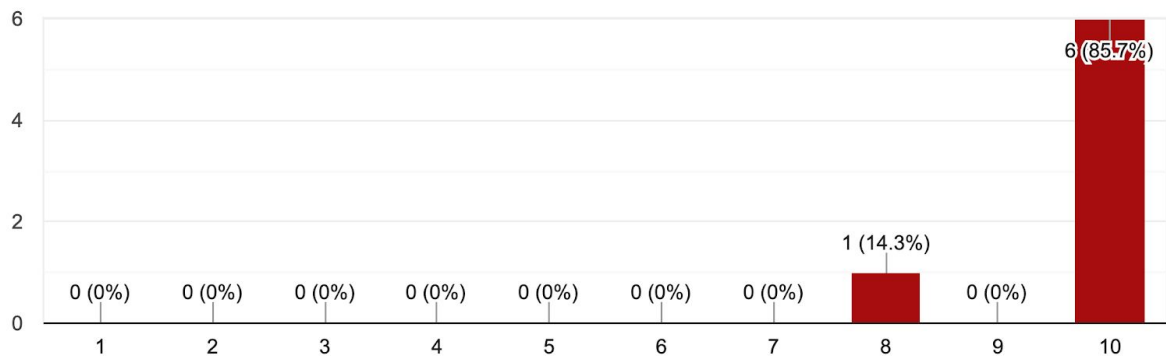
Fonte: Elaboração do Autor, 2020.

No gráfico 4, pode-se analisar que a interface agradou a todas pessoas que avaliaram o sistema, sendo que as respostas podiam ter valor entre 1 (interface nada fácil nem intuitiva) a 10 (interface totalmente fácil e intuitiva).

Gráfico 4 - Avaliação das usabilidade da interface.

O quão fácil e intuitivo é a interface?

7 responses



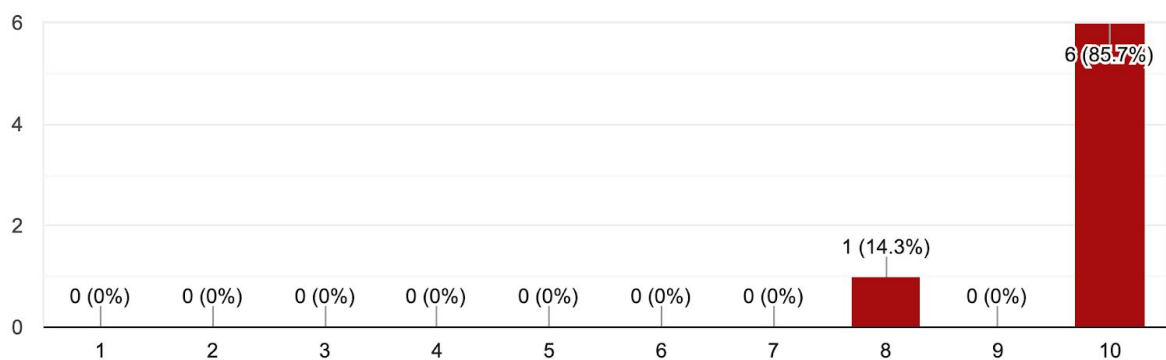
Fonte: Elaboração do Autor, 2020.

O gráfico 5 demonstra que o tempo de resposta foi adequado.

Gráfico 5 - Avaliação do tempo de resposta das funcionalidades.

O quão aceitável é o tempo de resposta das funcionalidas?

7 responses



Fonte: Elaboração do Autor, 2020.

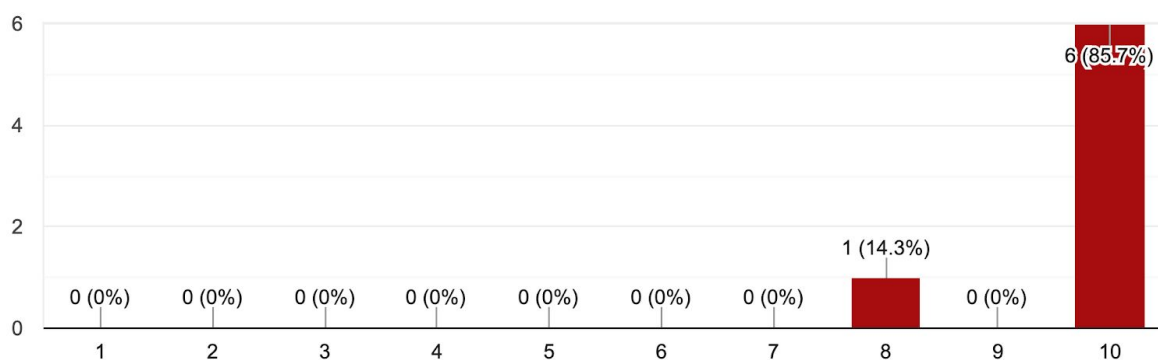


O gráfico 6 demonstra que a proposta elaborada neste trabalho tem utilidades para as pessoas que avaliaram o sistema.

Gráfico 6 - Avaliação da utilidade do serviço.

O quão útil você considera este app/serviço?

7 responses



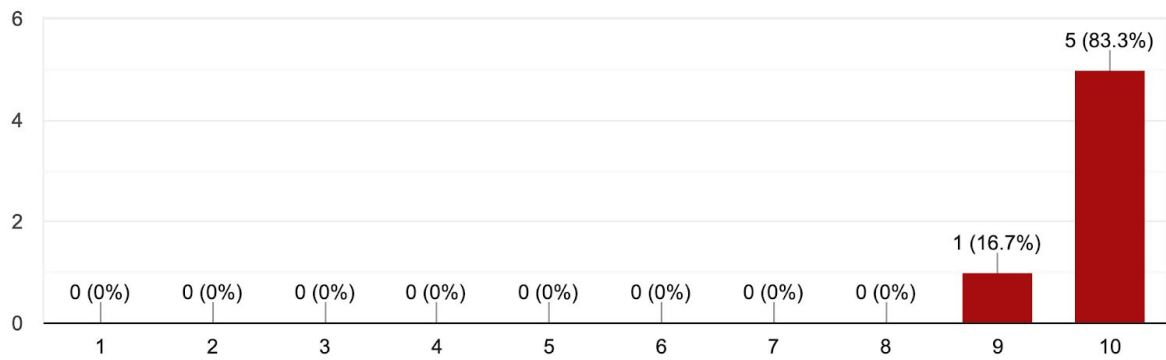
Fonte: Elaboração do Autor, 2020.

O gráfico 7 demonstra que a aplicação é eficaz em fornecer uma boa experiência de atendimento.

Gráfico 7 - Avaliação da proposta da aplicação.

Quão bem-sucedido é o app em relação ao que ele se propõe a fazer?

6 responses



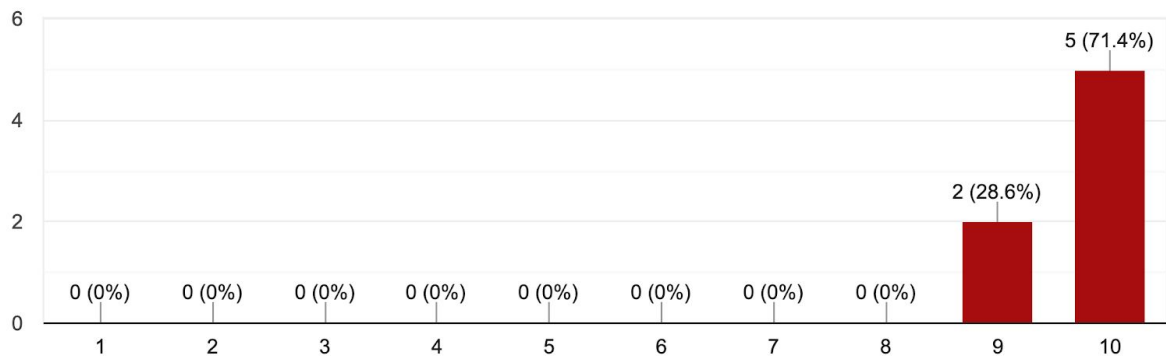
Fonte: Elaboração do Autor, 2020.

No gráfico 8, pode-se analisar que a interface do cliente é agradável.

Gráfico 8 - Avaliação da estética da interface do cliente.

O quão agradável é a interface?

7 responses



Fonte: Elaboração do Autor, 2020.

Por fim, na figura 8, é obtido sugestões gerais de algumas das pessoas que realizaram a avaliação.

Figura 28 - Sugestão de usuários que avaliaram a interface do cliente.

Tem alguma sugestão?

3 responses

- Está muito bom para o que se propõe.
- Aplicativo com interação intuitiva e clara de usar, muito promissor
- Selecionar o público alvo corretamente, escolher o seu nicho e deixar a interface própria para ele

Fonte: Elaboração do Autor, 2020.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo consiste nas considerações finais do trabalho como um todo, assim como os possíveis trabalhos futuros a serem desenvolvidos no projeto.

### 6.1 CONCLUSÕES

Dado o contexto com forte uso de tecnologia dos dias atuais, acredita-se que o projeto desenvolvido neste trabalho é capaz de apresentar uma alternativa efetiva e de baixo custo à estabelecimentos que procurem se alavancar no uso da tecnologia para aumentar sua eficiência de operação, assim como providenciar uma experiência mais agradável e simples aos clientes.

Durante a criação do sistema, a aplicação do cliente foi a que se demonstrou mais difícil de ser desenvolvida por conta de sua proposta de apresentar os benefícios de um aplicativo, porém no formato de site, sendo que as tecnologias disponibilizadas pelos navegadores para PWA não estão desenvolvidas por completo e que o ambiente de execução não é tão estável em relação à aplicativos nativos rodando diretamente no sistema.

Tendo dito isso, este trabalho se mostrou uma grande oportunidade de aperfeiçoamento das habilidades do autor, abordando diversas tecnologias e técnicas diferentes, tais como:

- Prototipação de aplicações com ferramentas como Adobe XD e Figma.
- Utilização de câmera do dispositivo através do navegador.
- Biblioteca de componentes Material UI.
- Gestão de estado de aplicações react utilizando Redux e Redux Thunk.
- Animações com React.
- Especificações de um PWA, bem como *service workers*.

- Firebase.
- Webpack.
- Typescript.
- Navegação em *single page applications*.

Além destas tecnologias, foi possível aplicar diversos conhecimentos desenvolvidos durante o curso de Sistemas de Informação.

Por fim, o sistema desenvolvido foi capaz de atender todos os objetivos e requisitos propostos, e a avaliação do sistema realizada por pessoas não relacionadas ao trabalho demonstraram um bom resultado do projeto.

## 6.2 TRABALHOS FUTUROS

O sistema desenvolvido já possui as funcionalidades básicas de um fluxo convencional de atendimento, porém, para realmente se tornar uma alternativa completa, pode-se ainda implementar diversas outras funcionalidades para auxiliar a operação de estabelecimentos do ramo alimentício, tais como:

- Avaliação de pedidos.
- Aplicação específica para recebimento de pedidos de bebidas.
- Histórico de compras do usuário.
- Seção de ajuda/informações.
- Pedido de ajuda à atendentes.
- Seção de detalhes do restaurante.
- Itens favoritos.
- Opção de entrega delivery ou retirar pedido no local.

Além de funcionalidades extras, existem diversas oportunidades de melhorias do lado técnico do sistema. Uma das mais relevantes seria adicionar testes

automatizados, que neste caso não foram implementados por conta do backend "terceirizado" e do escopo considerável de desenvolver três interfaces diferentes, mas ainda assim, algo necessário para um sistema que venha a ser utilizado de fato. Além de testes, é possível fazer as seguintes melhorias técnicas:

- Renderizar aplicação no servidor.
- Lazy load (carregamento tardio) de páginas.
- Otimizar performance.
- Otimizar ou diminuir tamanho da aplicação.

## REFERÊNCIAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. 3. Ed. São Paulo: Elsevier 2006. p. 13.

CHIAVENATO, Idalberto. **Gestão de Pessoas**. 3. ed. São Paulo: Elsevier, 2008.

DEVICEATLAS. **Android v iOS market share 2019**. Disponível em: <<https://deviceatlas.com/blog/android-v-ios-market-share>>. Acesso em 08 nov. 2019.

MENU DIGITAL. **Menu Digital**. São Paulo, [2019]. Disponível em: <http://www.digitalmenu.com.br>. Acesso em: 07 nov. 2019.

DINO. Expansão da automação para restaurantes é aposta para 2018; confira tendências. **Rev. EXAME**. São Paulo, 2017. Disponível em: <<https://exame.abril.com.br/negocios/dino/expansao-da-automacao-para-restaurantes-e-aposta-para-2018-confira-tendencias/>> Acesso em: 11 set. 2019.

DEJONG, Jessica. **What's next in Mobile: Progressive Web Apps**. Disponível em <<https://www.onenorth.com/blog/post/whats-next-in-mobile-progressive-web-apps>>. Acesso em: 18 nov. 2019.

FGV. **Resumo de Notícias: 30ª Pesquisa Anual do FGVcia da FGV/EAESP**. Disponível em: <[https://eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2019fgvcia\\_2019.pdf](https://eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2019fgvcia_2019.pdf)>. Acesso em: 12 set. 2019.

FIREBASE. **Firestore**. Estados Unidos, [2020]. Disponível em: <<https://firebase.google.com/>>. Acesso em: 28 jun. 2020.

GOOGLE. **Twitter Lite PWA Significantly Increases Engagement and Reduces Data Usage**. Disponível em: <<https://developers.google.com/web/showcase/2017/twitter>>. Acesso em 10 nov. 2019.

GOOGLE SOURCE. **Android Open Source Project**. California [2019]. Disponível em: <<https://android-review.googlesource.com/>>. Acesso em 18 nov. 2019.

GOOMER. **Cardápio Digital**. São Paulo, [2019]. Disponível em: <https://goomer.com.br>. Acesso em: 07 nov. 2019.

GÜNTHER, Hartmut. **Pesquisa Qualitativa Versus Pesquisa Quantitativa: Esta É a Questão?**. Disponível em <<http://www.scielo.br/pdf/%0D/ptp/v22n2/a10v22n2.pdf>>. Acesso em: 18. Nov 2019.

HANSEN, Nils Roll. **Why the distinction between basic (theoretical) and applied (practical) research is important in the politics of science**. London: CPNSS. 2009. Disponível em:  
<<https://pdfs.semanticscholar.org/62f0/dced123c24c7bc89b7d0d72bfcf885634a43.pdf>>. Acesso em: 18 nov. 2019.

HNAIDE, Sofian. **We are the SpaceX software team, ask us anything!**. Disponível em:  
<[https://www.reddit.com/r/spacex/comments/gxb7j1/we\\_are\\_the\\_spacex\\_software\\_team\\_ask\\_us\\_anything/](https://www.reddit.com/r/spacex/comments/gxb7j1/we_are_the_spacex_software_team_ask_us_anything/)>. Acesso em: 28 jun. 2020.

KIRUPA. **Understanding WebViews**. Disponível em:  
<<https://www.kirupa.com/apps/webview.htm>>. Acesso em 08 nov. 2019.

KREMER, Matt. **Ionic vs. React Native: A Comparison Guide**. Disponível em:  
<<https://ionicframework.com/resources/articles/ionic-vs-react-native-a-comparison-guide>>. Acesso em 08 nov. 2019.

MARQUES, Thiago. **Kotlin ou Java?**. Disponível em:  
<<https://medium.com/@thiago.marques/kotlin-ou-java-f47c00ba6303>>. Acesso em: 11 nov. 2019.

MATERIAL DESIGN. **Material Design**. Estados Unidos, [2020]. Disponível em:  
<<https://material.io/>>. Acesso em: 28 jun. 2020.

MATERIAL UI. **Material UI**. Estados Unidos, [2020]. Disponível em:  
<<https://material-ui.com/>>. Acesso em: 28 jun. 2020.

MENU. **Menu**. Suíça, [2019]. Disponível em: <https://menu.app>. Acesso em: 07 nov. 2019.

MINETTO, Dirceu. **"Uberização" é o presente e futuro dos negócios**. Disponível em:  
<<https://canaltech.com.br/negocios/uberizacao-e-o-presente-e-o-futuro-dos-negocios-51625/>>. Acesso em: 11 set. 2019.

NIENKÖTTER, Bruno Faraco. **(IM)POSSIBILIDADE DE CUMULAÇÃO DOS ADICIONAIS DE INSALUBRIDADE E PERICULOSIDADE NO CONTRATO DE TRABALHO, do curso de Direito da CESUSC**. 2018. Monografia (Graduação em Direito) - CESUSC, Florianópolis, 2018. p. 10.

NURVALA, Juha-Pekka. **'Uberisation' Is the Future of the Digitalised Labour Market**. Disponível em:  
<<https://journals.sagepub.com/doi/full/10.1007/s12290-015-0378-y>>. Acesso em: 12 set. 2019. p. 233.



PALMIERI, Manuel; SINGH, Inderjeet; CICCHETTI, Antonio. **Comparison of cross-platform mobile development tools**. Berlin. 2019. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/6376023/authors#authors>>. Acesso em 18 nov. 2019.

PREACT. **Preact**. Estados Unidos, [2020]. Disponível em: <<https://preactjs.com/>>. Acesso em: 28 jun. 2020.

REACT. **React**. Estados Unidos, [2020]. Disponível em: <<https://reactjs.org/>>. Acesso em: 28 jun. 2020.

REACT ROUTER. **React Router**. Estados Unidos, [2020]. Disponível em: <<https://reacttraining.com/react-router/>>. Acesso em: 28 jun. 2020.

RICHTER, Felix. **The Price Gap Between iOS and Android Is Widening**. Disponível em: <<https://www.statista.com/chart/1903/average-selling-price-of-android-and-ios-smartphones/>>. Acesso em 08 nov. 2019.

SIMFORM. **React Native vs Ionic: Comparing performance, User Experience and much more!**. Disponível em: <<https://www.simform.com/react-native-vs-ionic/>>. Acesso em 08 nov. 2019.

SOUZA, André Portela *et al.* **Custo do Trabalho no Brasil: Proposta de uma nova metodologia de mensuração**. rev. e atual. Brasília: Ed. FGV, 2012. Disponível em: <[https://cmicro.fgv.br/sites/cmicro.fgv.br/files/file/Custo%20do%20Trabalho%20no%20Brasil%20-%20Relatório%20Final\(1\).pdf](https://cmicro.fgv.br/sites/cmicro.fgv.br/files/file/Custo%20do%20Trabalho%20no%20Brasil%20-%20Relatório%20Final(1).pdf)>. Acesso em: 12 set. 2019. p. 4.

SOUZA, Gírlene Santos De; SANTOS, Anacleto Ranulfo dos; DIAS, Viviane Borges. **METODOLOGIA DA PESQUISA CIENTÍFICA: A construção do conhecimento e do pensamento científico no processo de aprendizado**. Porto Alegre: Animal, 2013. p. 60.

STOLL, Clifford. **Why The Web Won't Be Nirvana**. Disponível em: <<https://www.newsweek.com/clifford-stoll-why-web-wont-be-nirvana-185306>>. Acesso em: 11 set. 2019.

THINKWIK. **React Native: What is it? and, Why is it used?**. Disponível em: <<https://medium.com/@thinkwik/react-native-what-is-it-and-why-is-it-used-b132c3581df>>. Acesso em: 07 nov. 2019.

TYPESCRIPT. **Typescript**. Estados Unidos, [2020]. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 28 jun. 2020.

WEBPACK. **Webpack**. Estados Unidos, [2020]. Disponível em:  
<<https://webpack.js.org/>>. Acesso em: 28 jun. 2020.

WORKBOX. **Workbox**. Estados Unidos, [2020]. Disponível em:  
<<https://developers.google.com/web/tools/workbox/>>. Acesso em: 28 jun. 2020.

## APÊNDICE A - CRONOGRAMA

Atividade	Jan	Fev	Mar	Abr	Mai	Jun	Jul
Desenvolvimento de telas	X						
Levantamento de requisitos		X					
Modelagem de dados		X					
Desenvolvimento		X	X	X	X		
Entrega modelagem				X			
Testes manuais do produto					X		
Validação do produto					X		
Análise					X		
Entrega trabalho completo						X	
Defesa						X	
Correções						X	X
Entrega do trabalho final							X