



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE/PB
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM COMPUTAÇÃO**

SAMUEL RUFINO DA SILVA JUNIOR

**RELATÓRIO DE PROJETO EXPERIMENTAL: UMA PROPOSTA DE CATÁLOGO
DIGITAL PARA PEQUENOS COMÉRCIOS**

**CAMPINA GRANDE
2022**

SAMUEL RUFINO SILVA JUNIOR

**RELATÓRIO DE PROJETO EXPERIMENTAL: UMA PROPOSTA DE CATÁLOGO
DIGITAL PARA PEQUENOS COMÉRCIOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciências da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Computação.

Orientador: Prof. Me. Edson Holanda Cavalcante Júnior

**CAMPINA GRANDE
2022**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S586r Silva Junior, Samuel Rufino da.
Relatório de projeto experimental [manuscrito] : uma proposta de catálogo digital para pequenos comércios / Samuel Rufino da Silva Junior. - 2022.
26 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia , 2022.
"Orientação : Prof. Me. Edson Holanda Cavalcante Júnior , Departamento de Computação - CCT."

1. Pequenos negócios. 2. WhatsApp. 3. Comércio eletrônico . I. Título


21. ed. CDD 005.3


SAMUEL RUFINO DA SILVA JUNIOR


**RELATÓRIO DE PROJETO EXPERIMENTAL: UMA
PROPOSTA DE CATÁLOGO DIGITAL PARA PEQUENOS
COMÉRCIOS**

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 26 de Abril de 2022.


Prof. Me. Edson Holanda Cavalcante Júnior (DC - UEPB)
Orientador(a)


Prof. Dr. Wellington Candeia de Araújo (DC - UEPB)
Examinador(a)


Prof. Me. Francisco Anderson Mariano da Silva (CCEA - UEPB)
Examinador(a)

RESUMO

A ampla aceitação do comércio eletrônico brasileiro, resultou no aumento do número de pedidos nas pequenas empresas, muitas vezes não conseguindo atender seu público em tempo hábil. O presente estudo objetivou identificar a principal dificuldade no atendimento de pequenos negócios, de modo que se fizesse possível a sua automatização e melhora no processo utilizado através do canal de recebimento de pedidos, gerando assim, o desenvolvimento de um modelo capaz de automatizar tal processo de recebimento por meio do aplicativo de mensagens WhatsApp. Servindo como porta de entrada para pequenos comércios e até mesmo de prestadores de serviços de tecnologia, fontes viáveis de implementação de baixo custo e eficientes. Utilizando tecnologias e ferramentas gratuitas e de baixa complexidade a fim de promover uma ideia capaz atender necessidades básicas de pequenos comércios que acabam limitados ou excluídos do mundo tecnológico. Por fim, a abordagem e modelo proposto no estudo foram aplicadas em um projeto experimental, que mostrou-se adequado ao contexto da problemática citada, permitindo sua modificação, aplicação e utilização com quaisquer tecnologias que se assemelha ao descrito nesta pesquisa, contribuindo positivamente para a aplicação e divulgação de novas soluções.

Palavras-Chave: Pequenos negócios. Automatizar. Desenvolvimento. WhatsApp.

ABSTRACT

The wide acceptance of Brazilian electronic commerce, resulted in an increase in the number of orders in small businesses, often not being able to serve their audience in a timely manner. The present study aims to identify the main difficulty in serving small businesses, in order to make it possible to automate and improve the process used through the order receiving channel, thus developing a model capable of automating the order receipt through the WhatsApp messaging application. Serving as a gateway for small businesses and even technology service providers, viable sources of low-cost and efficient implementation. Using free and low-complexity technologies and tools in order to promote an idea capable of meeting the basic needs of small businesses that end up limited or excluded from the technological world. Finally, the approach and model proposed in the study were applied in an experimental project, which proved to be adequate to the context of the aforementioned problem, allowing its modification, application, and use with any technologies that are similar to those described in this research, contributing positively to the application and dissemination of new solutions.

Keywords: Small business. Automate. Development. Whatsapp.

LISTA DE ILUSTRAÇÕES

Figura 1 - Parte inicial do documento em HTML.....	11
Figura 2 - Primeira seção no <i>Body</i> do documento em HTML.....	11
Figura 3 - Segunda seção no <i>Body</i> do documento em HTML.....	12
Figura 4 - Terceira seção no <i>Body</i> do documento em HTML.....	12
Figura 5 - Finalização do documento em HTML.....	13
Figura 6 - Documento CSS.....	13
Figura 7 - Função dinâmica do elemento de Entrega no JavaScript.....	14
Figura 8 - Função JavaScript de Realizar Pedido.....	15
Figura 9 - Declarações de variáveis e estruturas de aceitação com dados para escrita da mensagem.....	15
Figura 10 - Declarações da variável ‘numero’ e estruturas de entrega da mensagem na API do WhatsApp.....	16
Figura 11 - Página inicial do Firebase.....	16
Figura 12 - Página (Passo 1) de criação de projeto do Firebase.....	17
Figura 13 - Página (Passo 2) de criação de projeto do Firebase.....	17
Figura 14 - Página (Passo 3) de criação de projeto do Firebase.....	18
Figura 15 - Página do projeto no Firebase.....	18
Figura 16 - Página (Passo 1) de adicionar o Firebase a aplicação.....	19
Figura 17 - Página (Passo 2) de adicionar o Firebase a aplicação.....	19
Figura 18 - Instalando o Firebase na pasta do projeto.....	20
Figura 19 - Página (Passo 3) de adicionar o Firebase a aplicação.....	20
Figura 20 - Instalando o CLI do Firebase na pasta do projeto.....	21
Figura 21 - Página (Passo 4) de adicionar o Firebase a aplicação.....	21
Figura 22 - Selecionando o <i>Hosting</i> , serviço do Firebase para hospedagem.....	22
Figura 23 - Carregando os arquivos e pastas do projeto para o Firebase.....	22
Figura 24 - Hospedagem no Firebase completa.....	22
Figura 25 - Modelo da aplicação hospedada.....	23

SUMÁRIO

1 INTRODUÇÃO.....	7
2 METODOLOGIA.....	9
3 TECNOLOGIAS UTILIZADAS.....	10
3.1 Modelo.....	10
3.1.1 HTML.....	10
3.1.2 CSS.....	13
3.2.3 JavaScript.....	14
3.2.4 Firebase.....	16
3.2.5 Apresentação do modelo.....	23
4 CONSIDERAÇÕES FINAIS.....	25
REFERÊNCIAS.....	26

1 INTRODUÇÃO

No contexto do mercado eletrônico, segundo uma pesquisa com 1.766 brasileiros entrevistados realizada pela Infobip à consultoria IDC (*International Data Corporation*), 88% dos brasileiros afirmam que já compraram online, ocupando a primeira posição no ranking regional com mais adesão, totalizando um grande crescimento em comparação aos anos anteriores, entre os países da América Latina (CANALTECH, 2021).

Com esse avanço nas vendas online, os empreendedores de pequenos negócios precisam desenvolver e elaborar estratégias sem que necessitem realizar um grande investimento para seus comércios.

Uma das soluções com ponto forte são os sites que podem ser acessados diretamente em qualquer navegador pré-instalado no *smartphone* ou em qualquer computador (CUNHA, 2021), o que significa que os pequenos negócios podem entrar no mundo da Web sem ter que arcar com os custos de um computador dedicado exclusivamente para aplicação, podendo ampliar a empregabilidade para qualquer tipo de negócio, caso não ocorra nenhuma grande restrição sobre os produtos oferecidos.

QRPedir, MeuMenuApp, Consumer e ShowKase Goomer são exemplos de algumas plataformas de pedidos para restaurantes que auxiliam nessa tarefa, porém, os custos elevados representam uma problemática por impossibilitar os pequenos comércios de arcarem com os custos desses sistemas.

Há também comércios e clientes que preferem utilizar a plataforma de mensagens *WhatsApp* para esse fim. Lançado em 2009, atualmente o *WhatsApp* conta com mais de 2 bilhões de usuários ativos por mês em todo mundo. Desses, mais 120 milhões somente no Brasil, com mais de 10 milhões de pequenos negócios utilizando seus serviços (G1, 2020).

Especificamente no caso dos pequenos restaurantes, com o aumento no número de pedidos via aplicativo de mensagem, as lojas não conseguem receber as informações necessárias e acabam congestionando o canal, elevando assim o tempo no atendimento, uma vez que representam uma estrutura bem menor do que as grandes empresas tradicionais têm e geralmente trabalham com um pequeno grupo de colaboradores.

Diante disso, empresas viram nessa problemática a oportunidade do crescimento e entraram com uma gama de serviços que não são prioridades para os pequenos negócios, uma vez que os mesmos desejam apenas receber de forma simples e organizada os pedidos, sem as amarras propostas no modelo das grandes empresas que dispõem de preços elevados de seus

planos, o que impossibilita que os pequenos negócios os adquiram e deixam de lado a usabilidade solicitada.

Há também empresas que oferecem sistemas mais simples com preço baixo ou até gratuito, mas que apresentam limitações no número de pedidos, o que pode causar constrangimento ao realizá-los ou fazer com que o proprietário seja surpreendido com cobranças indesejadas e pausa no atendimento. Além disso, são também pouco divulgadas, o que pode levar aos empreendedores dúvidas sobre as suas políticas, conduzindo-os ao questionamento acerca de futuras mudanças sem aviso prévio ou ao abandono em momentos difíceis.

Logo, o seguinte trabalho visa responder a seguinte questão norteadora: será que se faz possível a criação de uma aplicação que viabilize a introdução dos pequenos negócios no meio tecnológico e que seja capaz de satisfazer a necessidade dos usuários?

Diante dos desafios expostos, faz-se necessária a implementação modelo de sistema web que funcione como um cardápio digital ajudando na automatização dos pedidos em pequenos negócios, de forma gratuita, independente e simples. Desde sua estruturação e implantação até suas futuras manutenções, em que possam abrir novas janelas de negócios e serviços.

O objetivo da aplicação é minimizar e facilitar esse atendimento pelo canal *WhatsApp*, com o conceito de uma *Landing page*, ou seja, uma página de destino convertendo os visitantes do site em vendas, incluindo algum método para o visitante entrar em contato com a empresa e entregando todos os dados necessários para o pedido.

2 METODOLOGIA

Para desenvolver o presente estudo, se seguiu os seguintes passos:

Etapa 1: Realizou-se um levantamento sobre a problemática observando um caso real no qual uma empresa do ramo alimentício da cidade de Caicó - RN apresentava dificuldade em atender ao público utilizando-se apenas do aplicativo de mensagens *WhatsApp* para comunicação de pedidos, gerando congestionamento do canal;

Etapa 2: Foi necessário realizar um levantamento de ferramentas tecnológicas equivalentes às já existentes no mercado, identificando a real necessidade dos usuários;

Etapa 3: Estudo e escolha sobre as tecnologias que serão utilizadas para desenvolver o software:

- quais tecnologias de desenvolvimento serão escolhidas;
- onde será hospedado;

Etapa 4: Foi desenvolvido e hospedado o modelo, para demonstração para validação, baseado nas necessidades dos usuários.

3 TECNOLOGIAS UTILIZADAS

O modelo foi totalmente desenvolvido na *IDE Visual Studio Code* e com o conceito de *Landing page* utilizando apenas três arquivos e uma biblioteca, sendo eles:

- HTML (Linguagem de Marcação de HiperTexto), ele define o significado e a estrutura do conteúdo da página web;
- CSS (*Cascading Style Sheet*) é usado para estilizar elementos escritos em uma linguagem de marcação como HTML;
- *Framework Bootstrap* é uma biblioteca pré-preparada destinada a permitir um estilo mais fácil e mais compatível com os padrões de páginas da Web;
- *JavaScript* é utilizado para permitir que páginas da Web sejam interativas, definindo suas funcionalidades e comportamentos.

Além dessas tecnologias, é utilizada uma API (*Application Programming Interface*) desenvolvida pela empresa *WhatsApp*, a qual realiza a entrega da mensagem no aplicativo do usuário para a realização do pedido. Já para sua hospedagem e disponibilização na Web, foi utilizado o serviço *Firebase Hosting*, onde os arquivos foram hospedados de forma gratuita.

3.1 Modelo

Como mencionado, foram utilizados três arquivos, o primeiro sendo o HTML, no qual são escritos os produtos utilizando a estrutura segundo o padrão do HTML5, além de fazer a conexão com os outros dois arquivos e a tecnologia do *Framework Bootstrap*.

3.1.1 HTML

Algumas linhas iniciais, (1-7), são padrões do HTML5 para funcionamento correto do arquivo, dentro da TAG <head>. Podemos observar a linha 8 que será o Título do sistema na TAG <title>. Ademais, foi utilizado duas folhas de estilo, nas linhas 9 e 10, e a estrutura do *framework Bootstrap*, que é uma folha de estilo própria, na linha 11.

Figura 1 - Parte inicial do documento em HTML

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Lanchonete</title>
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet"
10  integrity="sha384-KyZXEAg3QhqLmP6G8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZ0J3BCsw2P0p/We" crossorigin="anonymous">
11  <link rel="stylesheet" href="estilo.css">
12
13 </head>

```

Fonte: Elaborado pelo autor, 2022¹.

Dentro da TAG <body>, em que se mostrará o conteúdo do site, temos seções, apresentadas pela TAG <section>, que dividem sua estrutura para melhor apresentação e por escolha prática foi dividido em apenas três segmentos.

A primeira seção apresenta dados de apresentação sobre a loja, sendo elas: foto da logomarca da empresa seguida de um título que será o nome da empresa.

Figura 2 - Primeira seção no *Body* do documento em HTML

```

15 <body>
16
17   <section>
18
19     <div>  </div> <br>
20     <h4 class="card-title text-center"> Lanchonete modelo </h4>
21
22   </section>

```

Fonte: Elaborado pelo autor, 2022.

A segunda seção apresenta todo o conteúdo de um produto da empresa realizado por divisões, sendo o primeiro o nome do produto e seu preço e em seguida tem-se a divisão que apresenta o sabor do produto, seus ingredientes e o nome “Quantidade”, pois abaixo há uma divisão interna com a TAG <input>, no qual será escrita a quantidade em que o futuro usuário digitará a quantidade que ele irá solicitar no pedido.

¹ Todas as imagens desse trabalho são *prints* de tela de computador produzidos pelo próprio autor.

Figura 3 - Segunda seção no Body do documento em HTML

```

24 <section>
25
26 <div>
27 <h3 class="card-header text-center" style="width: 17rem;"> ESPECIAL </h3>
28 <b>
29 <p class="card-header text-center" style="width: 17rem;"> R$ 2,00 </p>
30 </b>
31 </div>
32
33 <div class="card" style="width: 17rem;">
34 <div class="card-body">
35 <h5 class="card-title">Especial de Frango</h5>
36 <h6 class="card-title text-muted"> (Frango, Cheddar e Catupiry) </h6>
37 <p class="card-text card-subtitle mb-2 text-muted">Quantidade</p>
38 <div class="input-group mb-3">
39 <input id="especialFrango" type="number" class="form-control" name="especialFrango">
40 </div>
41 </div>
42 </div>

```

Fonte: Elaborado pelo autor, 2022.

A terceira seção é aquela em que se concentram as informações do cliente que irá solicitar o pedido contendo “Nome” e adicionará informações no campo “Observação” caso ele pretenda colocar algo a mais no pedido, selecionando também a modalidade de recebimento do pedido que poderá ser feita através do campo “Entrega” ou “Retirada na Loja”. Em caso de “Entrega”, também deverá informar o endereço seguido de: ‘Bairro’, o nome da “Rua”, “Nº”, e alguma “Referência” do endereço. Existe também um botão para clicar e realizar o pedido que será encaminhado através do canal via API para a *WhatsApp* do solicitante em conversa com a loja.

4 - Terceira seção no Body do documento em HTML

```

91 <section>
92
93 <h3>
94 <div class="card-header text-center" style="width: 17rem;"> <b>INFORMAÇÕES DO CLIENTE</b> </div>
95 </h3>
96
97 <div class="card" style="width: 17rem;">
98 <div class="card-body">
99 <p class="card-text card-subtitle mb-2 ">Nome: </p>
100 <div class="input-group mb-3">
101 <input id="nome" type="text" class="form-control" placeholder="Digite seu nome" aria-label="Digite o nome"
102 <input type="text" class="form-control" placeholder="Digite o nome"
103 </div>
104
105 <p class="card-text card-subtitle mb-2 ">Observações do Pedido: </p>
106 <div class="input-group mb-3">
107 <input id="observacoes" type="text" class="form-control" aria-describedby="basic-addon1">
108 </div>
109
110 <div class="btn-group btn-group-toggle" data-toggle="buttons" style="width: 14rem;"
111 <input type="checkbox" checked="" value="Entrega" /> Entrega R$8,00
112 <input type="checkbox" value="Retirada na loja" /> Retirada na loja
113 </div>
114
115 <div id="entrega">
116 <p class="card-text card-subtitle mb-2 ">Bairro: </p>
117 <div class="input-group mb-3">
118 <input id="bairro" type="text" class="form-control" placeholder="" aria-label=""
119 <input type="text" class="form-control" placeholder=""
120 </div>
121 <p class="card-text card-subtitle mb-2 ">Rua: </p>
122 <div class="input-group mb-3">
123 <input id="rua" type="text" class="form-control" placeholder="" aria-label=""
124 <input type="text" class="form-control" placeholder=""
125 </div>
126 <p class="card-text card-subtitle mb-2 ">NR: </p>
127 <div class="input-group mb-3">
128 <input id="num" type="text" class="form-control" placeholder="" aria-label=""
129 <input type="text" class="form-control" placeholder=""
130 </div>
131 <p class="card-text card-subtitle mb-2 ">Referência: </p>
132 <div class="input-group mb-3">
133 <input id="ref" type="text" class="form-control" placeholder="" aria-label=""
134 <input type="text" class="form-control" placeholder=""
135 </div>
136 </div>
137 <input class="btn btn-primary" type="button" value="Realizar Pedido" onclick="RealizarPedido()"
138 </div>
139 </div>
140 </div>
141 </div>
142
143 <div class="btn btn-primary" type="button" value="Realizar Pedido" onclick="RealizarPedido()"
144 </div>
145 </div>
146 </div>
147

```

Fonte: Elaborado pelo autor, 2022.

Por último, será feita a conexão com o arquivo em *JavaScript* para que as funções “RealizarPedido” e “checkBoxDeEntrega” sejam chamadas e executadas.

Figura 5 - Finalização do documento em HTML

```
156 <script src="script.js"> </script>
157
158 </body>
159
160 </html>
```

Fonte: Elaborado pelo autor, 2022.

3.1.2 CSS

Folhas de estilo são capazes de controlar a formatação dos elementos HTML de uma página. Nesse caso específico, além da folha de estilo apresentada abaixo, o HTML exposto anteriormente mescla a folha de estilo e o *framework Bootstrap* para dar estilo ao conteúdo.

Nessa nova folha de estilo, é escrito segundo um conjunto de regras com o nome do seletor e sua declaração. Na parte de declaração, deve-se inserir a propriedade e valor da propriedade separados por dois pontos (:). E dentro de cada conjunto de regras, deve-se usar um ponto e vírgula (;) para separar cada declaração da próxima.

Figura 6 - Documento CSS

```
1  body{
2    background: #000000;
3  }
4
5  head{
6    color: rgb(255, 255, 255);
7  }
8
9  img{
10   display: flex;
11   justify-content: left;
12   margin: auto;
13 }
14
15 .imgRedonda{
16   border-radius: 50%;
17 }
18
19
20 section{
21   background: rgb(255, 255, 255);
22   border-radius: 10px;
23   padding: 45px;
24   width: 400px;
25   max-width: 360px;
26   margin: auto;
27   box-shadow: 3px 3px 10px rgb(48, 45, 45);
28 }
29
30 div{
31   margin: 5px;
32 }
```

Fonte: Elaborado pelo autor, 2022.

3.2.3 JavaScript

JavaScript é uma linguagem de programação interpretada e estruturada de *script* em alto nível com tipagem dinâmica fraca e multiparadigma. Nesse modelo, o *JavaScript* tem o intuito de tornar o sistema dinâmico interpretando através das funções.

Uma das funções, chamada “checkboxDeEntrega”, será a de tornar dinâmico onde serão mostradas ou ocultadas as informações de entrega caso o usuário escolha entre ‘Entrega’ ou ‘Retirada em Loja’.

Figura 7 - Função dinâmica do elemento de Entrega no JavaScript

```
79 function checkboxDeEntrega() {
80
81     let checkbox1 = document.getElementById('option1');
82
83     if (checkbox1.checked) {
84         document.getElementById("entregaC").style.display = 'none';
85     } else {
86         document.getElementById("entregaC").style.display = 'block';
87     }
88
89 }
```

Fonte: Elaborado pelo autor, 2022.

Outra função será chamada “RealizarPedido”, capaz de interpretar e formular a mensagem que seguirá para a entrega na API do *WhatsApp*.

Seguindo o código, temos as declarações das variáveis:

- “numero”, será o número da loja para o qual a mensagem será encaminhada após a chamada da função;
- “whats” será a mensagem formatada que irá ser formulada para a entrega na API do WhatsApp;
- “nome”, “observacoes” e “especialFrango” serão informações que serão recebidas pelo usuário após a digitação no sistema;
- “preco”, será responsável por acumular o preço final do pedido.

Salientando que ‘especialFrango’ será a variável de quantidade digitada pelo usuário naquele item em específico.

Figura 8 - Função JavaScript de Realizar Pedido

```

1 function RealizarPedido() {
2
3     numero = 6826724231
4
5     whats = `https://api.whatsapp.com/send?phone=55${numero}&text=Olá%20quero%20fazer%20um%20pedido%20:%0A`
6     var nome = document.querySelector('input#nome').value
7     var observacoes = document.querySelector('input#observacoes').value
8     var preco = 0
9
10    /* Especial*/
11    var especialFrango = document.querySelector('input#especialFrango').value
12    if (especialFrango >= 1) {
13        whats += `%20${especialFrango}%20Especial de Frango%20%0A`
14        preco = preco + 2.00 * especialFrango
15    }

```

Fonte: Elaborado pelo autor, 2022.

Ao selecionar “Retirada em Loja”, a primeira função encaminhará o pedido diretamente a API do *WhatsApp*; caso esteja selecionado “Entrega”, irá reformular a mensagem com os dados de entrega e a encaminhará novamente ao API.

API (*Application Programming Interface*), que em Português traduz-se como Interface de Programação de Aplicações, representa um conjunto de definições e protocolos a fim de criar e integrar softwares de aplicações. No projeto em específico, foi utilizado um produto do WhatsApp que será nosso canal de comunicação entre os usuários.

Figura 9 - Declarações de variáveis e estruturas de aceitação com dados para escrita da mensagem

```

var bairro = document.querySelector('input#bairro').value
var rua = document.querySelector('input#rua').value
var num = document.querySelector('input#num').value
var ref = document.querySelector('input#ref').value
var endereco = `%0ABairro:%20${bairro}%0ARua:%20${rua}%0ANº:%20${num}%0AReferência:%20${ref}%0A`

if (option2.checked) {
    if (rua != '' && num != '' && bairro != '') {
        precoFrete = preco + 8.00
        window.open(`${whats}-----
%0ACliente: ${nome}%0A-----
%0AEndereço ${endereco}-----
%0ATotal: R$ ${preco.toFixed(2)} Reais
%0ATotal Com Frete: R$ ${precoFrete.toFixed(2)} Reais
%0AObservações do Pedido: ${observacoes}`);
    } else {
        window.alert('Digite o Bairro, Nome da rua e o Número');
    }
} else {
    window.open(`${whats}-----
%0ACliente: ${nome}%0A-----
%0ATotal: R$ ${preco.toFixed(2)} Reais
%0AObservações do Pedido: ${observacoes}`);
}

```

Fonte: Elaborado pelo autor, 2022.

O número da loja é incluído na URL da API, o qual encaminhará o usuário à página de conversa com a loja junto com a mensagem.

Figura 10 - Declarações da variável ‘numero’ e estruturas de entrega da mensagem na API do WhatsApp

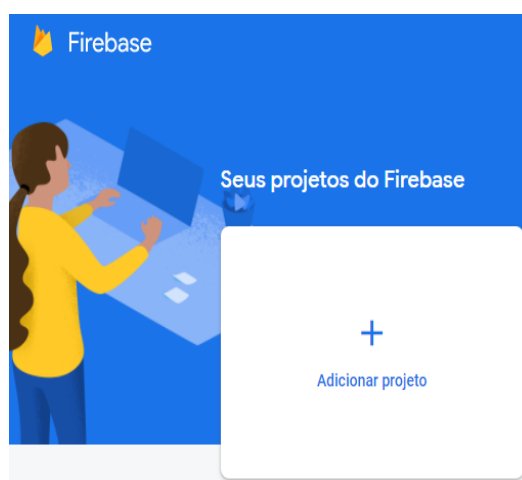
```
1 function RealizarPedido() {  
2  
3     numero = 6826724231  
4  
5     whats = `https://api.whatsapp.com/send?phone=55${numero}&text=Olá%20quero%20fazer%20um%20pedido%20:%0A`
```

Fonte: Elaborado pelo autor, 2022.

3.2.4 Firebase

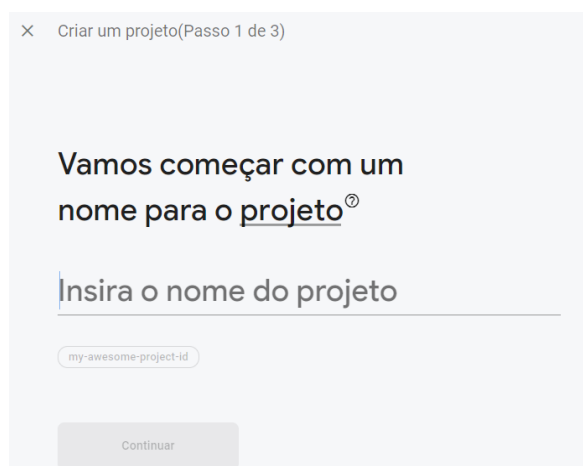
Para a hospedagem do projeto e criação do site no *FireBase*, é necessário possuir uma conta Google, criada gratuitamente pela plataforma. Após isso, entrar no site <https://firebase.google.com/> e entrar com *login* criado, sendo direcionado para a página <https://console.firebase.google.com/>, onde aparecerá “Seus projetos do Firebase”, então clicamos então em “Adicionar projeto” e em seguida surgirá um campo para inserir o nome do projeto e abaixo como ele será identificado e logo após a nomeação do projeto, deve-se clicar no botão “continuar”.

Figura 11 - Página inicial do Firebase



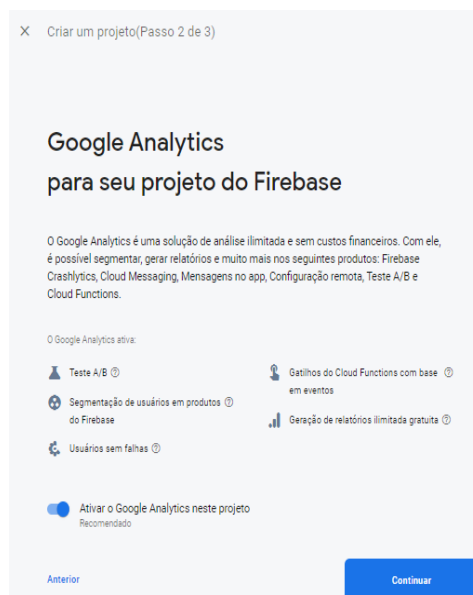
Fonte: Elaborado pelo autor, 2022².

² Print retirado do site <https://console.firebase.google.com/>. As figuras seguintes, até o número 17, também foram retiradas da mesma página, através de print em tela de computador.

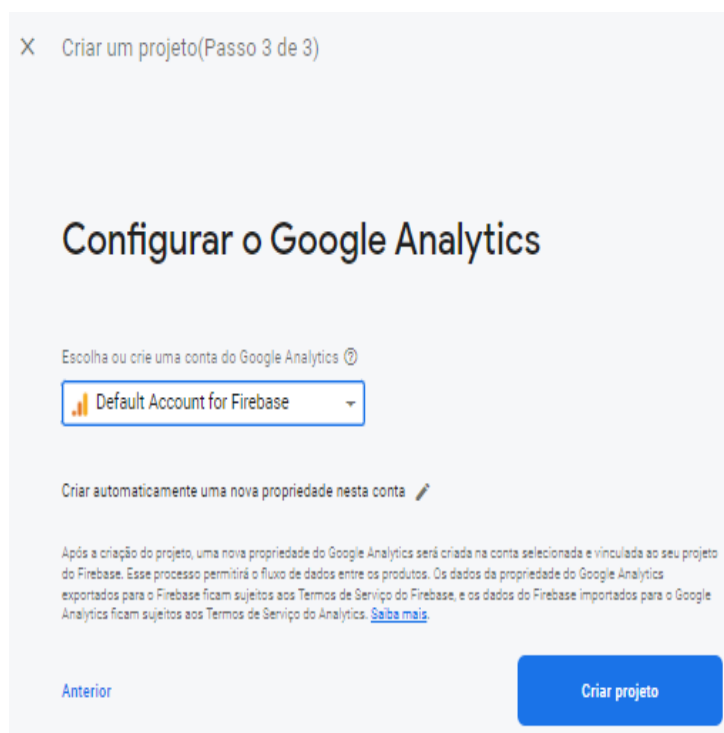
Figura 12 - Página (Passo 1) de criação de projeto do Firebase

Fonte: Elaborado pelo autor, 2022.

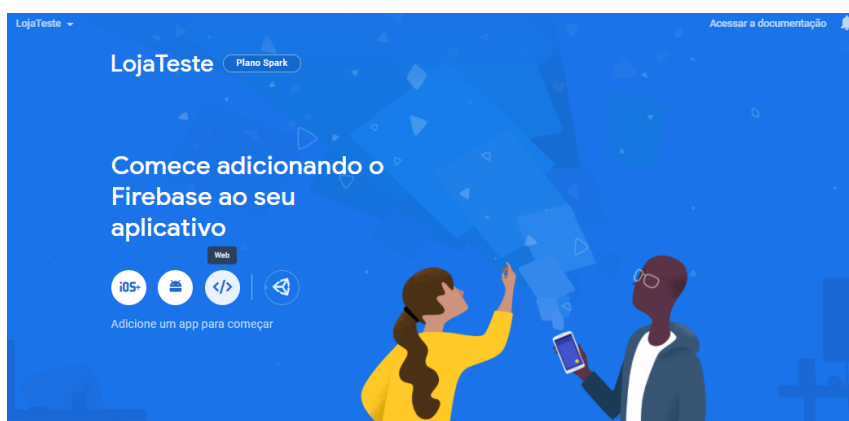
Nota-se que é oferecido um serviço adicional opcional “*Google Analytics*”, no qual é possível analisar algumas estatísticas do site após sua implantação e execução. Para o projeto teste, foi adicionado para termos uma demonstração final real. Seguimos então clicando em continuar, selecionando para ativação do *Google Analytics* e o botão continuar, para o qual será necessário selecionar a conta default do *Firebase* e em seguida clicar em “*Criar Projeto*”.

Figura 13 - Página (Passo 2) de criação de projeto do Firebase

Fonte: Elaborado pelo autor, 2022.

Figura 14 - Página (Passo 3) de criação de projeto do Firebase

Fonte: Elaborado pelo autor, 2022.

Figura 15 - Página do projeto no Firebase

Fonte: Elaborado pelo autor, 2022.

Será redirecionado para a página do projeto e aparecerão formas de implantar o serviço e, em seguida, ao clicar em Web, será redirecionado para página “Adicionar o Firebase ao App da Web”.

Na primeira opção “Registrar App”, criaremos um apelido do aplicativo e marcamos “Além disso, configure o Firebase Hosting para este app” para que faça toda a configuração do site com URL.

Figura 16 - Página (Passo 1) de adicionar o Firebase a aplicação

A captura de tela mostra uma janela de diálogo intitulada "Adicionar o Firebase ao seu app da Web". No topo, há um ícone de fechar (X) e o título. Abaixo, há um indicador de passo "1" e o texto "Registrar app".

Existem dois campos de entrada:

- Um campo rotulado "Apelido do app" com o ícone de ajuda (?), contendo o texto "Modelo do Produto".
- Um campo de seleção rotulado "modelodeteste10 (Nenhuma implantação air)", com um ícone de globo e uma seta para baixo.

Entre os campos, há uma seção com um ícone de checklist e o texto: "Além disso, configure o **Firebase Hosting** para este app. Saiba mais". Abaixo disso, um texto menor diz: "É possível configurar o recurso depois. Faça isso a qualquer momento sem custos financeiros."

No canto inferior direito, há um botão azul com o texto "Registrar app".

Fonte: Elaborado pelo autor, 2022.

Também é preciso adicionar o *SDK firebase*, kit de desenvolvimento ferramentas fornecidas pelo fabricante, também sendo necessária a criação de uma nova pasta vazia no computador, seguida pela abertura do *Prompt* de Comando e digitando o código, disposto na imagem em "Usar o npm" e clicando em "Enter".

Figura 17 - Página (Passo 2) de adicionar o Firebase a aplicação

A captura de tela mostra a segunda etapa da configuração, intitulada "2 Adicionar o SDK do Firebase".

Existem duas opções de seleção:

- Usar o npm
- Usar a tag <script>

Abaixo das opções, há um texto explicativo: "Se você já estiver usando o [npm](#) e um bundler de módulo, como [webpack](#) ou [Rollup](#), é possível executar o seguinte comando para instalar o SDK mais recente:"

Em uma caixa de texto cinza, o comando `$ npm install firebase` é exibido, com um ícone de copiar ao lado.

Fonte: Elaborado pelo autor, 2022.

Figura 18 - Instalando o Firebase na pasta do projeto

```
C:\Windows\System32\cmd.exe
Microsoft Windows [versão 10.0.19044.1586]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\samue\Desktop\Modelo do Produto>npm install firebase

> core-js@3.6.5 postinstall C:\Users\samue\Desktop\Modelo do Produto\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> protobufjs@6.11.2 postinstall C:\Users\samue\Desktop\Modelo do Produto\node_modules\protobufjs
> node scripts/postinstall

npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\samue\Desktop\Modelo do Produto\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\samue\Desktop\Modelo do Produto\package.json'
npm WARN Modelo do Produto No description
npm WARN Modelo do Produto No repository field.
npm WARN Modelo do Produto No README data
npm WARN Modelo do Produto No license field.

+ firebase@9.6.9
added 113 packages from 94 contributors and audited 113 packages in 90.61s

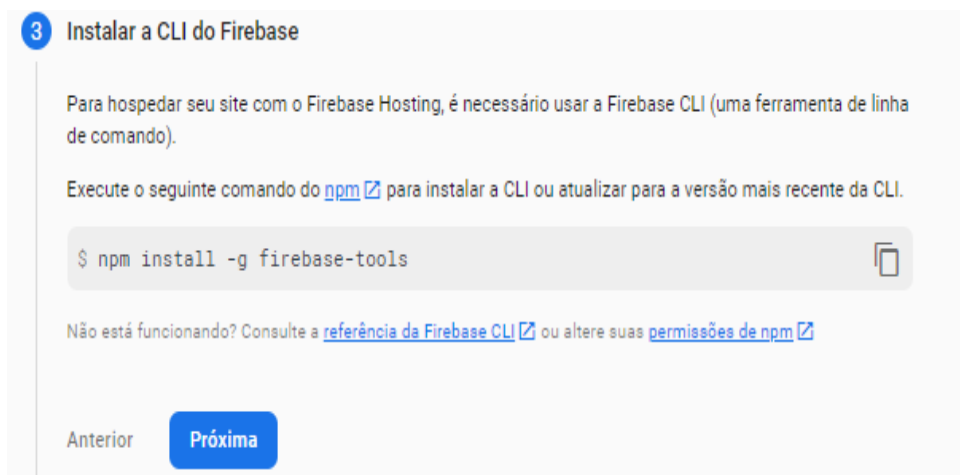
5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Fonte: Elaborado pelo autor, 2022.

Em seguida, será necessária também a instalação do comando *CLI(command-line interface)* do *Firebase*, ou seja, um programa onde permite que os usuários digitem comandos a um computador para fazer funções específicas. E então serão baixados no projeto os arquivos adicionais para a referência do *Firebase*, utilizando o código no *Prompt de Comando*.

Figura 19 - Página (Passo 3) de adicionar o Firebase a aplicação



Fonte: Elaborado pelo autor, 2022.

Figura 20 - Instalando o CLI do Firebase na pasta do projeto

```
C:\Users\samue\Desktop\Modelo do Produto>npm install -g firebase-tools
C:\Users\samue\AppData\Roaming\npm\firebase -> C:\Users\samue\AppData\Roaming\npm\node_modules\firebase-tools\lib\bin\firebase.js

> re2@1.17.3 install C:\Users\samue\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\re2
> install-from-cache --artifact build/Release/re2.node --host-var RE2_DOWNLOAD_MIRROR --skip-path-var RE2_DOWNLOAD_SKIP_PATH --skip-ver-var RE2_DOWNLOAD_SKIP_VER || npm
run rebuild

Trying https://github.com/uhop/node-re2/releases/download/1.17.3/win32-x64-83.br ...
Writing to build/Release/re2.node ...
Done.

> puppeteer@9.0.0 install C:\Users\samue\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\puppeteer
> node install.js

Downloading Chromium r869685 - 166.1 Mb [=====] 100% 0.0s
Chromium (869685) downloaded to C:\Users\samue\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\puppeteer\.local-chromium\win64-869685

> core-js@3.6.5 postinstall C:\Users\samue\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\core-js
> node -e "try(require('./postinstall'))catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job :-)
```

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\firebase-tools\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ firebase-tools@10.4.0
added 698 packages from 953 contributors, removed 90 packages and updated 146 packages in 108.277s
```

Fonte: Elaborado pelo autor, 2022.

Para transferir os arquivos do projeto, faz-se necessária a criação de uma pasta chamada *'public'* dentro da pasta criada anteriormente, além de anexar os arquivos HTML, CSS, JavaScript e a foto da loja.

Para o próximo passo, tem-se que voltar na pasta raiz e utilizar o *Prompt* de Comando. Agora será adicionando os comandos na imagem.

Figura 21 - Página (Passo 4) de adicionar o Firebase a aplicação

4 Implantar no Firebase Hosting

É possível implantar agora ou [depois](#). Para implantar agora, abra uma janela de terminal e navegue ou crie um diretório raiz a partir do seu app da Web.

Faça login no Google

```
$ firebase login
```

Iniciar seu projeto

Execute este comando no diretório raiz do seu app:

```
$ firebase init
```

Quando tudo estiver pronto, implante seu app da Web

Inclua seus arquivos estáticos (por exemplo, HTML, CSS, JS) no diretório de implantação do app (o padrão é "público"). Depois, execute este comando a partir do diretório raiz do app:

```
$ firebase deploy
```

Após a implantação, veja seu app em [modelodeteste10.web.app](#).

Precisa de ajuda? Confira a [documentação do Hosting](#)

Anterior Continuar no console

Fonte: Elaborado pelo autor, 2022.

Após digitar o primeiro comando irá direcionar ao login do Firebase, caso esteja logado irá aparecer o e-mail.

No segundo comando dá início a aplicação, utiliza-se as setas do teclado para mover entre os serviços e use a “Barra de espaço” para adicionar o serviço, nesse caso é selecionado o *Hosting*. A seguir, será necessário algumas configurações, é mantido o padrão, selecionando “Enter” em cada etapa.

Figura 22 - Selecionando o *Hosting*, serviço do Firebase para hospedagem

```
C:\Users\samue\Desktop\Modelo do Produto>firebase init

#####

You're about to initialize a Firebase project in this directory:

  C:\Users\samue\Desktop\Modelo do Produto

Before we get started, keep in mind:

  * You are initializing within an existing Firebase project directory

Are you ready to proceed? Yes

Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
  ( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
  ( ) Firestore: Configure security rules and indexes files for Firestore
  ( ) Functions: Configure a Cloud Functions directory and its files
  > (*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
  ( ) Hosting: Set up GitHub Action deploys
  ( ) Storage: Configure a security rules file for Cloud Storage
  ( ) Emulators: Set up local emulators for Firebase products
  (Move up and down to reveal more choices)
```

Fonte: Elaborado pelo autor, 2022

Figura 23 - Carregando os arquivos e pastas do projeto para o Firebase

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

i Using project modelodeteste10 (LojaTeste)

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
? Set up automatic builds and deploys with GitHub? No
? File public/404.html already exists. Overwrite? No
i Skipping write of public/404.html
? File public/index.html already exists. Overwrite? No
i Skipping write of public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...

+ Firebase initialization complete!
```

Fonte: Elaborado pelo autor, 2022.

Figura 24 - Hospedagem no Firebase completa

```
C:\Users\samue\Desktop\Modelo do Produto>firebase deploy

=== Deploying to 'modelodeteste10'...

i deploying hosting
i hosting[modelodeteste10]: beginning deploy...
i hosting[modelodeteste10]: found 5 files in public
+ hosting[modelodeteste10]: file upload complete
i hosting[modelodeteste10]: finalizing version...
+ hosting[modelodeteste10]: version finalized
i hosting[modelodeteste10]: releasing new version...
+ hosting[modelodeteste10]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/modelodeteste10/overview
Hosting URL: https://modelodeteste10.web.app
```

Fonte: Elaborado pelo autor, 2022.

Após todos os passos, o site será mostrado em “*Hosting URL*”. Ao realizar todos os passos, consegue-se acessar a URL e com isso obtém-se o resultado final da aplicação para uso, podendo ser utilizada seguindo a abordagem que o empreendedor e loja acharem melhor, disponibilizando o link na hora do atendimento ou até em outras redes sociais.

3.2.5 Apresentação do modelo

Por fim, no que tange à via do cliente, o usuário deverá digitar a quantidade do item desejado, selecionar o modo de entrega e, se necessário, outros parâmetros. Clicado no botão ‘Realizar Pedido’, será redirecionado ao canal de comunicação *WhatsApp* diretamente com a loja, com seu pedido digitado na barra de mensagem, na qual será necessário o mesmo confirmar o envio da mensagem à loja.

Figura 25 - Modelo da aplicação hospedada

The figure displays two screenshots of a mobile application interface. The left screenshot shows a menu item 'Especial de Frango' (Special Chicken) with a price of R\$ 2,00. The item description is '(Frango, Cheddar e Catupiry)'. The quantity is set to 2. Below the menu item is a form titled 'INFORMAÇÕES DO CLIENTE' (Customer Information) with fields for Name, Observações do Pedido (Order Observations), Bairro (Neighborhood), Rua (Street), Nº (Number), and Referência (Reference). The delivery options are 'Entrega R\$8,00' (Delivery R\$8,00) and 'Retirada na loja' (Pickup in-store). A blue 'Realizar Pedido' (Place Order) button is at the bottom.

The right screenshot shows the same 'INFORMAÇÕES DO CLIENTE' form, but with the 'Entrega R\$8,00' option selected. The 'Realizar Pedido' button is also visible at the bottom.

Fonte: Elaborado pelo autor, 2022.

Durante o desenvolvimento do trabalho, foi percebida a utilização dos diversos conhecimentos formados ao longo do curso, também proporcionando o contato com novos desafios que seguirão de aprendizado.

4 CONSIDERAÇÕES FINAIS

Com as informações obtidas, foi desenvolvido o sistema e logo depois de implementado, conclui-se que atingiu o objetivo proposto, tendo na sua viabilidade o intuito de suprir os requisitos necessários para que o estabelecimento atenda à demanda de forma automatizada e simples e liberando a retenção dos pedidos de forma que não se tenha custos altos utilizando o canal *WhatsApp* da loja.

Foi observada como vantagem da utilização das tecnologias propostas a implementação da ferramenta com baixo custo operacional. O software desenvolvido intenta, de forma simples e objetiva, a evolução dos pequenos negócios para atender às necessidades dos clientes de forma a não desperdiçar recursos obsoletos.

Possuindo ainda vantagens sobre os usuários maliciosos, uma vez que os mesmos não são mais capazes de acessar ou prejudicar o sistema, tendo em vista que o sistema apenas gera os dados necessários e organizados para o pedido entregue a API fornecida pelo *WhatsApp*. O cliente recebe essa mensagem, revisa e envia para a realização do pedido na loja.

No entanto, observam-se desvantagens no que tange à falta de um software *Back-end*, no qual poderia haver a possibilidade de salvar o pedido no sistema ou outros recursos em futuras modificações aplicadas no sistema, ficando assim restrito à plataforma de mensagens.

O modelo utilizado possibilita, em oportunidades futuras, promover a divulgação e implementação em pequenos comércios locais que também tenham os mesmos problemas. Além de viabilizar um serviço de manutenção e verificar mudanças necessárias para o bom funcionamento do software.

A limitação encontrada no decorrer do estudo está voltada para a escassez bibliográfica, a qual não apresenta focos voltados para as problemáticas encontradas e até mesmo às tecnologias com o objetivo principal, apenas encontrando notícias industriais, de cunho superficial, as quais não possuem embasamento científico suficiente para a reprodução ou aprendizagem.

No que se refere a estudos futuros, sugere-se que o modelo seja ampliado e difundido para o que esta discussão não se esgote aqui, sendo fonte para futuras pesquisas.

REFERÊNCIAS

- ANDRION, R. Estudo aponta que 88% dos brasileiros entrevistados compram online. **CanalTech**, [S.l.], 2021. Disponível em: <https://canaltech.com.br/negocios/estudo-aponta-que-88-dos-brasileiros-entrevistados-compra-online-195253/>. Acesso em: 05 mar. 2022.
- CNDL. **91% DOS INTERNAUTAS REALIZARAM COMPRAS PELA INTERNET NOS ÚLTIMOS 12 MESES, APONTA PESQUISA CNDL/SPC BRASIL**. [S.l.], 2021. Disponível: <https://site.cndl.org.br/91-dos-internautas-realizaram-compras-pela-internet-nos-ultimos-12-meses-aponta-pesquisa-cndlspc-brasil/> Acesso em: 05 mar. 2022.
- CUNHA, A. S. **PLANO DE NEGÓCIOS PARA UM RESTAURANTE COM DELIVERY: Uma Alternativa Durante a Pandemia COVID-19**. Trabalho de Conclusão de Curso. Santa Catarina, 2021. <https://repositorio.animaeducacao.com.br/handle/ANIMA/13312>. Acesso em: 05 mar. 2022.
- FIREBASE. **FIREBASE HOSTING**. [S.d.], 2022. Disponível em: <https://firebase.google.com/docs/hosting/>. Acesso em: 16 mar. 2022.
- G1. **5 PERGUNTAS SOBRE O MERCADO BILIONÁRIO POR TRÁS DA FUNÇÃO DE PAGAMENTOS DO WHATSAPP**. [S.l.], 2020. Disponível em: <https://g1.globo.com/economia/tecnologia/noticia/2020/06/17/5-perguntas-sobre-o-mercado-bilionario-por-tras-de-funcao-de-pagamentos-do-whatsapp.ghtml>. Acesso em: 07 mar. 2022.
- MDN WEB DOCS. **INTRODUÇÃO À WEB**. [S.d.], 2021. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web. Acesso em: 10 mar. 2022.
- SOMMERVILLE, I. **Engenharia de Software**: Tradução de Ivan Bosnic e Kalinka G. de O. Gonçalves. 9ª Ed. São Paulo: Pearson Prentice Hall, 2011.
- WHATSAPP. **COMO SE CONECTAR AO WHATSAPP USANDO OUTRO APP**. [S. n.]. Disponível em: <https://faq.whatsapp.com/iphone/how-to-link-to-whatsapp-from-a-different-app/>. Acesso em: 13 mar. 2022.