

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

WEIDYNER ALEXSANDRO SANTOS SILVA

**SISTEMA DE CONTROLE DE COMPRAS ATRAVÉS DA NOTA FISCAL DO
CONSUMIDOR (NFC-e)**

PONTA GROSSA

2021

WEIDYNER ALEXSANDRO SANTOS SILVA

**SISTEMA DE CONTROLE DE COMPRAS ATRAVÉS DA NOTA FISCAL DO
CONSUMIDOR (NFC-e)**

PURCHASE CONTROL SYSTEM THROUGH THE CONSUMER INVOICE (NFC-e)

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do DEPARTAMENTO ACADÊMICO DE INFORMÁTICA, da UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR Campus Ponta Grossa.
Orientador(a): Prof. Msc. Rogério Ranthum

PONTA GROSSA

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

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

WEIDYNER ALEXSANDRO SANTOS SILVA

**SISTEMA DE CONTROLE DE COMPRAS ATRAVÉS DA NOTA FISCAL DO
CONSUMIDOR (NFC-e)**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do DEPARTAMENTO ACADÊMICO DE INFORMÁTICA, da UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR Campus Ponta Grossa.

Data de aprovação: 02/Dezembro/2021

Rogério Ranthum
Mestrado
Universidade Tecnológica Federal do Paraná

Geraldo Ranthum
Mestrado
Universidade Tecnológica Federal do Paraná

Lourival Aparecido de Góis
Doutorado
Universidade Tecnológica Federal do Paraná

**PONTA GROSSA
2021**

Dedico este trabalho a minha mãe Rosicler Santos Silva e a meu pai Edinir Silva que sempre me incentivaram a estudar e me proporcionaram, através de muito sacrifício pessoal, a oportunidade de chegar até aqui. A minha esposa que sempre me apoiou nessa caminhada.

AGRADECIMENTOS

Agradeço a Deus acima de tudo que, de forma gratuita, me deu a oportunidade de conquistar mais essa etapa em minha vida. Agradeço aos meus pais pelo amor, conselhos, paciência e recursos que disponibilizaram para o meu crescimento. A minha amada esposa Cristiane Silva pelo seu companheirismo e amor em todo esse tempo. As minhas filhas Isabela e Helena, que abriram mão de momentos juntos para que eu pudesse continuar os estudos. Agradeço também ao Professor Rogério Ranthum pela disponibilidade e apoio neste trabalho. Aos meus colegas de turma, dos quais destaco, Pedro Henrique pela amizade e compartilhamento de conhecimento em momentos oportunos. Aos meus amigos que torceram e me incentivaram sempre nesse percurso. Por fim, a instituição de ensino UTFPR de Ponta Grossa e a todos os professores que de alguma forma cooperaram para o meu crescimento intelectual e pessoal.

*E tudo quanto fizerdes, fazei-o de todo o coração, como ao
Senhor; e não aos homens.
Colossenses 3:23*

RESUMO

SILVA, Weidyner. SISTEMA DE CONTROLE DE COMPRAS ATRAVÉS DA NOTA FISCAL DO CONSUMIDOR (NFC-e). 2021. 44 f. Trabalho de Conclusão de Curso – TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. Ponta Grossa, 2021.

A alteração no preço dos produtos, no Brasil, torna-se cada dia mais frequente e traz um grande impacto na vida de todos os brasileiros. Essas variações podem ocorrer por diversos fatores, seja por influência de oferta e demanda, aumento de custos na produção, ou até mesmo fatores naturais porém, independentemente da causa, essas variações precisam ser monitoradas pelos consumidores para que, estes, tenham conhecimento e possam estudar, planejar e decidir da melhor forma possível seus gastos e, com isso, manter uma vida financeira saudável e equilibrada. As ferramentas para esse tipo de monitoramento são diversas e estão potencializadas com a difusão da tecnologia atual. Dentre elas destacam-se os aplicativos mobile que estão presentes no dia-a-dia de boa parte da população. Esses aplicativos, somados a informações concretas, confiáveis e atualizadas, podem resultar em um auxílio significativo para vida financeira de toda a sociedade. Mediante isto, este trabalho procura auxiliar o indivíduo na tarefa de monitoramento de gastos proporcionando o acesso facilitado às informações de valores por meio de um sistema de gerenciamento moderno, simples e confiável. Este sistema de gerenciamento será desenvolvido sob a metodologia PWA, uma metodologia que visa utilizar as linguagens de programação web e os recursos disponíveis nos navegadores modernos para criar aplicativos para celulares com características de aplicativos nativo. Através do aplicativo desenvolvido será possível acessar as informações da nota de consumidor eletrônica e guardá-las em um banco de dados para que possam ser acessadas de forma rápida e organizada. Assim o consumidor poderá ter uma base de informações para auxiliá-lo em seu controle financeiro.

Palavras-chave: PWA. Nota Fiscal do Consumidor. QR Code.

ABSTRACT

SILVA, Weidyner. PURCHASE CONTROL SYSTEM THROUGH THE CONSUMER INVOICE (NFC-e). 2021. 44 f. Work of Conclusion Course (Graduation in Systems Analysis and Development Technology) – Federal Technological University of Paraná. Ponta Grossa, 2021.

The change in the price of products in Brazil is becoming more and more frequent and has a great impact on the lives of all Brazilians. These variations can occur due to several factors, whether due to the influence of supply and demand, increased costs in production, or even natural factors, but regardless of the cause, these variations need to be monitored by consumers so that they are aware of and can study, plan and decide your expenses in the best way possible and, with that, maintain a healthy and balanced financial life. The tools for this type of monitoring are diverse and are enhanced with the spread of current technology. Among them, mobile applications stand out, which are present in the daily lives of a large part of the population. These applications, added to concrete, reliable and up-to-date information, can result in significant financial support for the entire society. Through this, this work seeks to assist the individual in the task of monitoring expenses, providing easy access to valuable information through a modern, simple and reliable management system. This management system will be developed under the PWA methodology, a methodology that aims to use web programming languages and the resources available in modern browsers to create mobile applications with native application characteristics. Through the developed application it will be possible to access the information of the electronic consumer note and store them in a database so that it can be accessed quickly and organized. Thus, the consumer will be able to have an information base to help him in his financial control.

Keywords: PWA. CONSUMER INVOICE. QR Code.

LISTA DE FIGURAS

Figura 1 – Categorias de despesas financeiras mensais	2
Figura 2 – Variação do salário mínimo	3
Figura 3 – Compatibilidade entre API Indexeddb com navegadores desktops	13
Figura 4 – Compatibilidade entre API Indexeddb com navegadores mobile	13
Figura 5 – Diagrama de caso de uso	14
Figura 6 – Diagrama de sequencia do cadastro de compras	15
Figura 7 – Sequência de cadastro para compras offline.	16
Figura 8 – Diagrama de sequencia do processo de pesquisa.	16
Figura 9 – Diagrama de sequência do processo de pesquisa offline.	17
Figura 10 – Código de registro do Service Worker.	17
Figura 11 – Service Worker registrado.	18
Figura 12 – Código para a criação de cache no navegador.	19
Figura 13 – Cache do navegador Chrome	19
Figura 14 – Códigos de redirecionamento para possíveis falhas de requisição.	20
Figura 15 – Trecho do arquivo Manifest.	21
Figura 16 – Página inicial do sistema de controle de compras.	22
Figura 17 – Simulação do status offline da página index.html.	22
Figura 18 – Página com a câmera inicializada.	23
Figura 19 – Métodos da biblioteca Instascam.	24
Figura 20 – Código para processo de leitura do QR Code pela câmera.	24
Figura 21 – Página de cadastro de compra offline	25
Figura 22 – Redirecionamento em situação offline da pagina Cadastrar Compra.	25
Figura 23 – Página da NFC-e na SEFAZ-PR.	26
Figura 24 – Código para obter estrutura do site.	27
Figura 25 – Informações obtidas pelo método de raspagem de dados.	27
Figura 26 – Mensagem enviada quando o cadastro é realizado com sucesso.	28
Figura 27 – Exclusão e criação de database na API IndexedDB	28
Figura 28 – Trecho do código para a criação da Tabela na API IndexedDB.	29
Figura 29 – Banco de dados indexedDB criado com as respectivas tabelas.	29
Figura 30 – Pagina de pesquisa online	30
Figura 31 – Trecho de código da promise inicil da pesquisa.	31
Figura 32 – Código css para alteração de largura da tela.	32
Figura 33 – Resultados gerais.	32
Figura 34 – Resultado detalhado de Performace.	33
Figura 35 – Resultado detalhado de Acesisibilidade.	33
Figura 36 – Resultado detalhado Boas Práticas.	34
Figura 37 – Resultado detalhado de Otimização nos mecanismos de buscas.	34
Figura 38 – Resultado sobre Aspectos PWA.	35
Figura 39 – Opção de criação de atalho no navegador Chrome	36
Figura 40 – Atalho criado para acesso ao aplicativo no desktop.	36

Figura 41 – Layout do aplicativo web para site	37
Figura 42 – Criação de atalho para aplicativo web pelo navegador Chrome no smartphone	38
Figura 43 – Atalho criado para acesso ao aplicativo no smatphone.	39
Figura 44 – Layout do aplicativo web para o site no smartphone	40

LISTA DE TABELAS

Tabela 1 – Classificação de despesas financeiras.	2
---	---

LISTA DE ABREVIATURAS E SIGLAS

App	Do inglês Application
IPVA	Imposto sobre a Propriedade de Veículos Automotores
IPTU	Imposto Predial Territorial Urbano
IPCA	Índice Nacional de Preços ao Consumidor
SIDRA	Sistema IBGE de Recuperação Automática
IBGE	Instituto Brasileiro de Geografia e Estatística
INPC	Índice Nacional de Preço ao consumidor
SINIEF	Sistema Nacional Integrado de Informações Econômico-Fiscais
NFE	Nota Fiscal Eletrônica
NFC-e	Nota Fiscal do Consumidor Eletrônica
SEFAZ	Secretaria do Estado da Fazenda
AIM	Procedimentos de autorização de introdução no mercado
EDI	Do inglês Electronic Data Interchange
SPC	Serviço de Proteção ao Crédito
RPA	Revista de Pesquisa em Administração
IOS	Do inglês iPhone Operating System
HTML	Do inglês Hyper Text Markup Language
CSS	Do inglês Cascading Style Sheets
URL	Do inglês Uniform Resource Locator
PWA	Progressive Web App
PHP	Acrônimo recursivo para Hypertext Preprocessor
ASP	Do inglês Active Server Page
W3C	Do inglês World Wide Web Consortium
DOM	Do inglês Document Object Model
HTTPS	Do inglês Hyper Text Transfer Protocol Secure
JSON	Do inglês JavaScript Object Notation
NCSA	Norma do Sistema do Comando da Aeronáutica

WWW	Do inglês World Wide Web
XML	Do inglês eXtensible Markup Language
JPEG	Do inglês Joint Photographics Experts Group
GIF	Do inglês Graphics Interchange Format
PNG	Do inglês Portable Network Graphics
ECMA	Do inglês European Computer Manufacture's Association

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Justificativa	1
1.2	Objetivo geral	4
1.3	Objetivos específicos	4
1.4	Organização do trabalho	4
2	REFERENCIAL TEÓRICO	5
2.1	Nota fiscal do consumidor eletrônica	5
2.2	<i>QR Code</i>	5
2.3	Aplicativos para controle financeiro	6
2.4	Aplicativos para controle de compras.	7
2.5	Aplicativos baseados na NFC-e	7
2.6	Tipos de aplicativos	7
2.6.1	Aplicações nativas	7
2.6.2	Aplicações híbridas	7
2.6.3	Aplicações Web Apps	8
2.7	<i>Progressive Web App</i>	8
2.7.1	Service worker	9
2.7.2	<i>HTTPS</i>	9
2.7.3	<i>Manifest web</i>	9
2.7.4	<i>Promise</i>	10
2.8	Evolução dos navegadores	10
2.9	Javascript	11
2.9.1	Bibliotecas Javascript	11
2.10	CSS	12
2.11	Raspagem de dados	12
2.12	Banco de dados indexeddb	12
3	DESENVOLVIMENTO	14
3.1	Descrição do projeto	14
3.2	Caso de uso	14
3.3	Diagrama de sequência	14
3.4	Estrutura do sistema	17
3.5	Service worker	17
3.6	Manifest	21
3.7	Index	21
3.8	Cadastrar compra	22
3.9	Página para cadastrar compra quando há falhas na requisição	25
3.10	Página para mostrar compras	26
3.11	Página para gravar a compra	27
3.12	Página de pesquisa	28
3.13	Página de pesquisa quando há falha na requisição	30

3.14	Página CSS	31
3.15	Teste de metodologia PWA	32
3.16	Configuração do site como aplicativo web no desktop	35
3.17	Configuração do site como aplicativo web no smartphone	37
4	CONCLUSÃO	41
4.1	Considerações finais	41
4.2	Trabalhos futuros	41
	REFERÊNCIAS	42

1 INTRODUÇÃO

Na vida cotidiana a palavra consumo é bem conhecida por todos, este termo está relacionado às necessidades humanas e baseia-se na utilização de bens e serviços para a satisfação destas necessidades (GALVES, 2003). Esta palavra também está de certa forma ligada ao termo receita, que é o recebimento de valores de forma regular (MICHAELIS, 2021). Quando este consumo é maior do que a receita na vida financeira de um indivíduo, tem-se um desequilíbrio extremamente prejudicial para a saúde financeira do mesmo. Esse desequilíbrio acontece por diversos fatores, alguns relacionados ao consumo como: o consumo pelo prazer e satisfação pessoal em ter, possuir ou adquirir algo; o consumo relacionado a necessidade de lazer e divertimento (BLACKWELL, 2001). Há também outros relacionados à receita, como algum incidente que diminua os valores recebidos, por exemplo, a perda de emprego ou de benefícios. Por fim, este desequilíbrio pode acarretar sérios problemas financeiros, familiares e pessoais.

Neste cenário as ferramentas para controle de gastos financeiros, como os aplicativos de celulares, podem ser muito úteis para a população em geral. Muitos aplicativos são voltados ao auxílio no orçamento familiar e pessoal, alguns aplicativos coletam os dados diretamente da conta bancária, e em outros casos os dados são inseridos manualmente pelo usuário (GAVIOLI, 2021).

No geral, os aplicativos para controle de gastos deixam de maneira genérica as informações, separando-as em despesas como luz, água, gás, transporte, internet, cartões em geral, alimentação, entre outros.

No que diz respeito às compras de supermercado, os aplicativos focam em lista de itens, com o objetivo de auxiliar o usuário, incluindo os itens necessários para a compra de supermercado, porém deixam de lado os valores respectivos. Contudo, com a frequente variação no valor dos preços dos alimentos, essas informações podem ser muito úteis para um melhor controle financeiro.

O uso da nota fiscal do consumidor eletrônica(NFC-e) pode ser de grande valia para os aplicativos de controle de gastos. Neste sentido, o uso das informações contidas nas NFC-e podem abastecer um aplicativo com os dados da compra de maneira fácil e rápida. Assim, o usuário pode controlar o gasto em cada item, qual o impacto do mesmo no valor da renda destinada a compra, quais os itens que se tornaram mais caros com o passar do tempo e quais itens têm sido os vilões na compra de supermercado.

1.1 Justificativa

Segundo o dicionário Michaelis a receita é definida como todos os valores recebidos em determinado período de tempo, sejam eles rendas, proventos, etc (MICHAELIS, 2021). As despesas, por sua vez, são definidas como gastos feitos com o intuito de adquirir algo que satisfaça uma necessidade. As despesas encontram-se presentes na vida financeira comum, seja na obtenção de algo de necessidade básica ou até mesmo de supérfluos.

As despesas compreendem basicamente 3 tipos: fixas, variáveis e eventuais (BULGARIM, 2012), como mostra a Tabela 1. Despesas fixas são os gastos financeiros constantes ou habituais que podem ocorrer uma ou várias vezes ao mês. Ex. Aluguel, água, luz, telefone, supermercado, etc. O uso da palavra fixa para esse tipo de despesa se dá pela frequência e não pelo valor, pois o valor de uma despesa fixa pode ser variado. Mesmo assim, é possível prever com certa exatidão o valor da despesa pela sua constância. Um exemplo da variância do valor das despesas fixas são as compras de supermercado. Elas são constantes porém seu valor varia de acordo com a quantidade e valores dos produtos.

As despesas variáveis são despesas que ocorrem habitualmente mas não necessariamente mês a mês como é o caso de compra de roupas, calçados, lazer, etc (BULGARIM, 2012). Outra classificação de despesas são as despesas eventuais, elas são os gastos feitos a partir de demandas pontuais e/ou por despesas inesperadas. Dentre estes estão

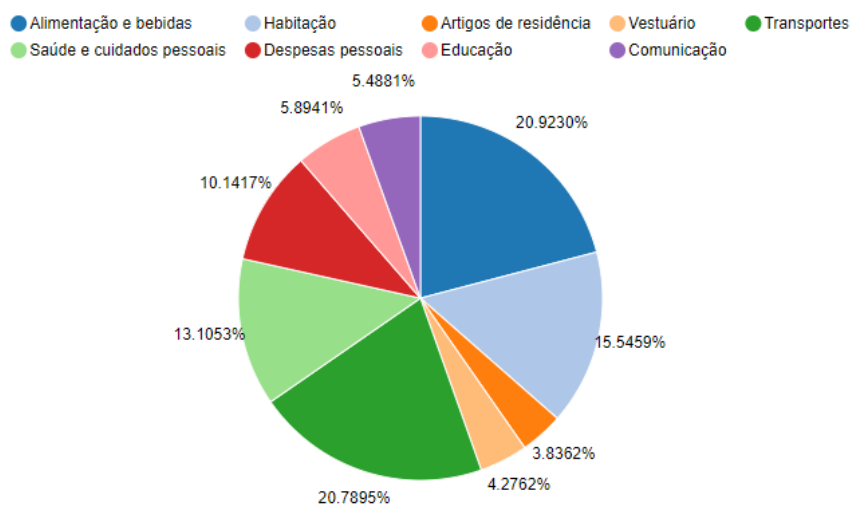
gastos com impostos como IPVA, IPTU, Imposto de Renda, concertos em geral, consultas médicas, remédios, etc.

Tabela 1 – Classificação de despesas financeiras.

Tipo de despesa	Definição	Exemplo
Fixa	Gastos financeiros constantes	Alimentação, água, luz, telefone, aluguel, etc.
Variável	Gastos financeiros habituais	Lazer, vestuário, etc.
Eventual	Demandas pontuais	Concertos, etc...

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), as despesas dos brasileiros estão divididas em 9 categorias. São elas: Alimentação e bebidas; habitação; artigos de residência; vestuário; transportes; saúde e cuidados pessoais; despesas pessoais; educação e comunicação, como ilustra a Figura 1 (Sidra - IPCA, 2021).

Figura 1 – Categorias de despesas financeiras mensais dos brasileiros Junho de 2021



Fonte: SIDRA- IPCA, 2021

A parcela da receita dos brasileiro utilizada para cada categoria está relacionada no gráfico da Figura 1. Dentro das despesas dos brasileiros, a despesa com alimentos e bebidas consome 20,9% da receita mensal, e dentre as categorias listadas é o maior valor das despesa (IBGE, 2021), sendo maior que as despesas com habitação e gastos com saúde. Esta categoria é classificada nas despesas fixas e que apresentam seu valor variável. Uma das causas dessa variação é o aumento no valor dos produtos. Este aumento é influenciado por alguns fatores tais como inflação; oferta e demanda; custo de produção e clima.

A inflação é o aumento generalizado dos preços. A inflação equilibrada é benéfica para a economia, quando os preços aumentam, teoricamente, o salário também aumenta fazendo com que o povo continue a consumir, criando um ciclo econômico (IBGE, 2021). O problema é quando o salário não acompanha a inflação e o consumidor acaba tendo perda no seu poder de compra.

O IBGE produz dois índices para observar a inflação: o Índice Nacional de Preços ao Consumidor Amplo (IPCA) e o Índice Nacional de Preço ao Consumidor (INPC). O IPCA usa como base os brasileiros com renda de 1 a 40 salários mínimos enquanto que o INPC pesquisa os brasileiros com ganhos de 1 a 5 salários mínimos. Esses índices são utilizados pelo governo para o reajuste do salário mínimo nacional. O reajuste do salário mínimo nacional tem seu valor nominal e real. O valor nominal é a diferença entre o valor anterior e o valor reajustado e o valor real é a porcentagem

do reajuste comparado ao INPC, ou seja, a inflação do ano anterior. Assim pode-se analisar o impacto da inflação nas finanças do brasileiros.

Ao longo dos anos de 2016, 2017, 2018 e 2019 a variação do salário mínimo teve um aumento real menor do que 1,2% sendo que nos anos de 2017 e 2018 o valor foi negativo como é possível verificar na Figura 2.

Figura 2 – Variação do salário mínimo no período de 2004 à 2019

PERÍODO	SALÁRIO MÍNIMO (R\$)	REAJUSTE NOMINAL (%)	INPC (%)	AUMENTO REAL (%)
mai/04	260	-	-	-
mai/05	300	15,38	6,61	8,23
abr/06	350	16,67	3,21	13,04
abr/07	380	8,57	3,30	5,10
mar/08	415	9,21	4,98	4,03
fev/09	465	12,05	5,92	5,79
jan/10	510	9,68	3,45	6,02
jan/11	545	6,86	6,47	0,37
jan/12	622	14,13	6,08	7,59
jan/13	678	9	6,20	2,64
jan/14	724	6,78	5,56	1,16
jan/15	788	8,84	6,23	2,46
jan/16	880	11,68	11,28	0,36
jan/17	937	6,48	6,58	-0,10
jan/18	954	1,81	2,07	-0,25
jan/19	998	4,61	3,43	1,14

Aumentos reais negativos significam perda no poder de compra, ou seja, será necessário mais dinheiro para comprar o mesmo item, porém com a mesma renda relativa. A oferta e demanda também afeta o preço dos itens de consumo. O consumo de alimentos tem crescido no mundo, porém a produção de alimentos não acompanha o crescimento da demanda. Temos países emergentes como a China e próprio Brasil que tem apresentado uma melhora na condição de vida e por isso acabam podendo consumir mais (G1, 2021). Com um maior consumo tem-se menos produtos no mercado fazendo com que os produtos sejam mais disputados e aumentando assim seu valor. Situações como a pandemia por conta do Covid-19, também podem aumentar a demanda, como ocorreu em 2020 fazendo com que o preço do arroz e óleo de soja aumentassem consideravelmente nos supermercados (JANKAVSKI, 2021).

Os custos na produção de alimentos podem variar influenciados por outros fatores como custo do combustível tanto para utilização de equipamentos como para o transporte da produção, fertilizantes e demais insumos (G1, 2021).

O clima, por sua vez, pode afetar diretamente o custo dos produtos. Um estudo desenvolvido pelo banco central relata a ação de fenômenos naturais como a Lã Niña e o El Niño, na inflação dos alimentos. No Brasil, os efeitos do fenômeno El Niño se caracterizam pelo aumento nas temperaturas, excesso de chuva no Sul e Sudeste, e estiagem no Norte e Nordeste (KEMPSON, 2021).

Para auxiliar no controle de gastos, existem algumas ferramentas que podem ajudar a equilibrar as finanças, uma delas é o orçamento. O orçamento é uma ótima ferramenta para um controle financeiro adequado, ela consiste na amostragem antecipada de receita e despesas que deverão ocorrer em certo período (BULGARIM, 2012). Para criar um orçamento ou fazer qualquer análise financeira é necessário informações, dados que possam ser utilizados para tal

análise.

As despesas com alimentos em supermercados nos possibilitam uma grande quantidade de informações. Quando se efetua uma compra em qualquer supermercado é obrigatório a geração de documento fiscal com os produtos e valores da transação. Em 2013 foi instituída, através do ajuste SINIEF 01/13, a Nota Fiscal de Consumidor Eletrônica podendo ser aderida por todas as unidades da federação (CONFAZ, 2021). Com este documento é possível obter os dados de produtos e valores adquiridos de forma fácil e em tempo real (SPED-PR, 2021). Estes dados podem, de maneira rápida, serem empregados em ferramentas como o orçamento financeiro.

Diante dos fatores apresentados e no fato de que gastos com a alimentação comprometem a maior parte dos gastos e despesas financeiras dos brasileiros, é prudente analisá-los com maior intensidade, com o intuito de contribuir para uma vida financeira mais saudável e equilibrada. Suas variâncias promovidas por diversos fatores devem ser mostradas com maior transparência e clareza para que o consumidor possa estar a par das mudanças reais em suas despesas.

1.2 Objetivo geral

Desenvolver um aplicativo de controle de compras de supermercado para auxiliar no monitoramento das variações do preços dos produtos e dos gastos em alimentos, possibilitando ao usuário gravar suas compras e fazer pesquisas com filtros específicos para analisá-las.

1.3 Objetivos específicos

- Desenvolver um aplicativo baseado em metodologia *PWA* para cadastro e pesquisa de compras.
- Utilizar o *QR Code* da Nota Fiscal do Consumidor Eletrônica (NFC-e) para acessar as informações da nota
- Extrair os dados da NFC-e diretamente para o aplicativo.
- Gravar os dados para uso *on-line* e *off-line*.

1.4 Organização do trabalho

Este trabalho foi estruturado da seguinte forma: o capítulo 2 apresenta a fundamentação teórica sobre os temas abordados no trabalho, apresentando uma breve descrição das ferramentas empregadas em aplicativos de controle de compras e orçamento. O capítulo 3 apresenta o desenvolvimento do projeto mostrando as principais páginas do site de forma separada, apresentando os resultados da auditoria da ferramenta *Lighthouse* e o site em formato de aplicativo web para *desktop* e *mobile*. Por fim o capítulo 4 conclui o trabalho, apresentando uma análise sobre o que foi feito e indicando possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Nota fiscal do consumidor eletrônica

O Projeto da Nota Fiscal eletrônica foi criado com o objetivo da aquisição de um documento com validade jurídica garantida por meio de assinatura digital do emissor e que contribuísse para diminuição da impressão de documentos fiscais em papel. De igual modo, o projeto da Nota Fiscal do Consumidor Eletrônica(NFC-e), nasceu com o mesmo objetivo porém, além de diminuir os gastos com impressão de papel, o objetivo era também diminuir os gastos com obrigações acessórias como o caso das impressoras fiscais conhecidas como Emissor de Cupom Fiscal(ECF) (SEFAZ, 2021). A utilização da NFC-e traz vários benefícios tanto para empresa emissora, dispensando o uso de impressora fiscal, de mapa de caixa, aposição de lacres, registro de intervenções técnicas, possibilitando integração de plataformas de vendas físicas e virtuais, para o consumidor que pode fazer a consulta em tempo real do documento fiscal e ao Estado que pode monitorar as transações, cruzar informações e fazer auditoria eletrônica.

No Paraná a NFC-e foi adotada através da resolução SEFA nº 145/2015 Publicada no DOE 9428 de 09.04.2015 e tornou-se obrigatória a partir de janeiro de 2017 (AEN, 2021). Desde então, todos os mercados devem utilizar o modelo em suas transações no varejo. Para acessar as informações da NFC-e há duas formas, recebendo a nota impressa no momento da compra ou acessando o site da Secretaria da Fazenda do Estado em que foi emitido. O acesso pelo site pode ser feito com o número da chave da nota ou através do QR Code impresso na nota fiscal. Este QR Code é item obrigatório nas NFC-e de acordo com o item 2.1.2 da Norma de Procedimento Fiscal 100/2014 (SPED-PR, 2021).

2.2 QR Code

A história do *QR Code* inicia em meados dos anos 60, o Japão havia entrado no período de crescimento da sua economia, período no qual os supermercados vendiam uma quantidade enorme de produtos e iam se expandindo rapidamente. Nesta época, os funcionários dos caixas ainda utilizavam o método de digitar manualmente os preços de cada produto na caixa registradora. Isto ocasionava dormência nos membros superiores e causavam problemas como síndromes.

A invenção dos códigos de barras veio para solucionar este problema. Desenvolveram-se sistemas onde o preço de cada mercadoria era apresentado na caixa registradora e, automaticamente, lido o código de barras do produto por um sensor óptico. Com a propagação do uso dos códigos de barras, ficaram aparentes também suas limitações. Uma das maiores era o fato de que poderia ser usado apenas 20 caracteres alfanuméricos para as informações.

Usuários contactaram a *DENSO WAVE INCORPORATED*, uma divisão da *DENSO CORPORATION*, que estavam desenvolvendo leitores de código de barras naquele tempo, para esta, verificar a possibilidade de desenvolver um código de barras com mais informações. Encorajados, um time de desenvolvimento da *DENSO WAVE*, começaram o desenvolvimento de um novo código bidimensional com o propósito de acomodar todas as necessidades dos usuários.

O encarregado pelo desenvolvimento era Masahiro Hara que procurava um código que suportasse um número grande de informações e ao mesmo tempo fosse de fácil leitura, o grande desafio encontrava-se no fato da necessidade de um código que permitisse uma leitura simples e rápida. Um dia, eles tiveram a ideia de adicionar posições indicativas referenciando o código a ser lido. Assim surgiu o padrão de detecção de posição composto de marcas quadradas tornando possível a leitura de alta velocidade.

As marcas de forma quadrada foram utilizadas porque, de acordo com Hara, era o padrão menos provável de aparecer em várias formas de negócios visto que se há uma marca de aparência semelhante é possível que o leitor se confunda na leitura. Houve também uma exaustiva pesquisa em imagens como panfletos, fotos, símbolos, revistas, etc

para verificar a proporção de áreas brancas em áreas pretas para ser criado um padrão da proporção em distância que acabou sendo de 1:1:3:1:1. Assim determinou-se as distâncias e a largura das áreas pretas. Dessa forma o código cuja orientação podia ser determinada, independentemente, do ângulo da digitalização. Um ano e a meio depois do início do desenvolvimento do projeto, depois de várias tentativas e erros, foi criado um código que conseguia salvar até 7.000 numerais e em torno de 20 vezes mais rápido que outros códigos.

Por ser um código aberto em que qualquer pessoa pode usar o código QR Code, por meio da padronização, seu uso se difundiu ainda mais. Em 1997, foi aprovado como um padrão AIM, e foi empregado na indústria de identificação automática. Em 1999, foi aprovado como um código 2D padrão pela Japan Industrial Standards e fez um símbolo 2D padrão nos formulários de transação padrão *EDI* da *Japan Automobile Manufacturers Association*. Em 2000, a ISO aprovou como um de seus padrões internacionais. Atualmente, o uso do *QR Code* é tão difundido que é empregado em todo o mundo (CODE, 2019).

2.3 Aplicativos para controle financeiro

Pesquisas realizadas em 2017 pelo Serviço de Proteção ao Crédito (SPC) e a Confederação Nacional de Dirigentes Lojistas indicaram que 6 em cada 10 brasileiros fazem o controle de seu orçamento, e 58,7% destes, sentem dificuldade em monitorar seus gastos e despesas. Dos entrevistados 77,3% consideraram que enfrentaram ao menos uma situação em que a renda familiar não foi suficiente para quitar ou suprir as contas e despesas assumidas no mês em 2017. O que indica que existem sérios problemas nos orçamentos das famílias brasileiras. Dos entrevistados, a maioria relatou o uso de caderno de anotações, agenda, papel para controle financeiro, torno de 28,2%, 18,3% relataram o uso de planilha de computador e apenas 8,6% o uso de aplicativos de celular (SPCBRASIL, 2018).

O uso de aplicativos de celulares para controle financeiro não está “alinhado” com o gigante crescimento do seu uso para outras finalidades pelos brasileiros.

Em 2018, um estudo divulgado pela *App Annie*, empresa que faz pesquisas mundiais sobre o mercado de aplicativos, mostrou que o brasileiro usa em média 12 *apps* a cada 24 horas em seu dispositivo móvel. Esta mesma pesquisa aponta que o brasileiro possui 83 *apps* em média instalados em seu *smartphone* dos quais 41 são utilizados frequentemente com 3 horas de utilização.

A Google colocou o mercado brasileiro na lista chamada “*Next Billion Users*”, o próximo bilhão de usuários da internet. Esta lista mostra mercados em expansão e que tem muito potencial para melhorar (G1, 2021). Entre 2015 e 2018 o Brasil teve um crescimento de 20% em downloads de aplicativos, sendo um dos mercados emergentes com mais rápido crescimento no mundo quando se fala em aplicativos. A *App Annie* prevê que em 2022 o Brasil ultrapasse a marca de 8 Bilhões de downloads ficando na quarta posição no mundo, atrás apenas de Estados Unidos, Índia e China (RIBEIRO, 2021).

A falta de utilização de *apps* para controle financeiro pelos brasileiros pode ser explicada no estudo feito pela revista RPA - Revista de Pesquisa em Administração de Caruaru - PE. Esta pesquisa revelou que 39% dos usuários que disseram não utilizar nenhum *app* para controle financeiro, não o utilizam por falta de conhecimento, falta de interesse ou por não achar necessário. Dentre as respostas relatadas, algumas expressam um cenário que pode ser melhorado com novas soluções. São elas: Não ter encontrado um *app* bom, falta de paciência e falta de tempo.

Com essas informações, é importante o estudo de soluções que contornem a falta de tempo e paciência dos usuários, com recursos automatizados e confiáveis para auxílio do controle financeiro (SILVA, 2020).

2.4 Aplicativos para controle de compras.

A *Play Store* é a loja virtual de aplicativos oficial para download de *apps* para celulares da plataforma Android. Através dela, listou-se os 5 aplicativos de controle de compras mais baixados (LATEX, 2021). Segue:

- 1-*Out Of Milk*
- 2-*Listonic*
- 3-Lista de Compras - *TK solution*
- 4-*Bring!* Lista de Compras.
- 5-*By Me And Pie*

Analisando os aplicativos, é possível encontrar muitas funcionalidades que trazem facilidades para os usuários, algumas delas são: Lista de compras, Sincronização e compartilhamento de listas em tempo real, Armazenamento de histórico de compras, Dispensa, Antecipação de valores totais e parciais, Comparação histórica de preços. Todas essas funcionalidades funcionam baseando-se em informações cadastradas manualmente pelos usuários.

2.5 Aplicativos baseados na NFC-e

O software Menor Preço foi disponibilizado no dia 29 de novembro de 2016 para dispositivos Android e IOS. Ele é mantido pelo Governo do Paraná e traz algumas funcionalidades baseadas nas NFC-es geradas nas vendas realizadas nos estabelecimentos do estado. Os preços dos produtos são atualizados em tempo real e o usuário pode conferir os estabelecimentos que estão com os melhores preços. O software pesquisa os produtos através da descrição, marca e código de barras. Sua pesquisa mostra dentro de um raio específico, apresentando os resultados para os estabelecimentos mais próximos. Também é possível compartilhar os preços com outras pessoas através de vários canais de comunicação. O sistema tem a opção de criar listas de produtos onde a lista pesquisa todos os produtos listados de uma vez (CELEPAR, 2021).

2.6 Tipos de aplicativos

2.6.1 Aplicações nativas

Aplicações nativas são softwares desenvolvidos para plataformas específicas utilizando linguagem de programação própria para a plataforma. Essas aplicações podem utilizar todos os recursos dos dispositivos tais como: Câmera, GPS, acelerômetro, *push notifications*, lista de contatos, bússola entre outros. *Apps* nativos podem ainda funcionar sem acesso a internet caso tenha conteúdo embarcado. Estes *apps* tem como características rapidez, integração ao hardware do dispositivo, podem manipular o funcionamento do aparelho, têm acesso às funções nativas do aparelho podendo até substituí-las e tem possibilidade de integração com outros *apps*. Aplicativos nativos precisam ser disponibilizados em lojas específicas para sua plataforma, eles são instalados consumindo espaço de armazenamento interno. Sua forma de publicação pode ser lenta e burocrática, além de ter custo financeiro para quem disponibiliza-lo (TRINDADE P. E.; AFFINI, 2018).

2.6.2 Aplicações híbridas

As aplicações híbridas foram criadas com programação baseada na web (HTML, CSS, JavaScript) porém são executadas em ambiente de processo nativo. Eles utilizam uma funcionalidade do sistema chamado *WebView*. Esta funcionalidade apresenta o código web como uma aplicação responsiva, sem barra de endereço, com tela cheia e sem os demais controles do navegador. Esse modelo de aplicação se torna mais eficaz em questão de custo pois reutiliza o código para várias plataformas visto que se trata de tecnologias web. Estas aplicações são distribuídas de forma similar

às aplicações nativas, elas utilizam as lojas específicas de cada plataforma para serem disponibilizadas (TRINDADE, 2020).

2.6.3 Aplicações *Web Apps*

Web Apps é o nome dado às aplicações que são executadas nos *browsers* dos aparelhos celulares. Eles utilizam as linguagens padrões da internet como HTML, CSS e JavaScript e tem o objetivo de criar aplicações com aparência de aplicações nativas. Há pouco acesso aos recursos de hardwares nessas aplicações pois seu código é apenas interpretado pelo navegador. *Web Apps* podem ser executadas em qualquer plataforma pois não há controle do Sistema Operacional diretamente em suas funcionalidades e conteúdos. Este tipo de aplicação não necessita de instalação e também não precisa ser disponibilizada por uma loja de aplicativos, sendo que seu acesso é via URL. Este tipo de aplicativo se popularizou quando a internet se tornou mais acessível, tornando comum o acesso à internet através dos dispositivos móveis. Aplicações Webs podem criar um ícone na tela inicial do aparelho para serem acessados com mais facilidade, suas atualizações tem efeito imediato sem a necessidade de reinstalação (TRINDADE, 2020).

2.7 *Progressive Web App*

Progressive Web App(PWA) pode ser traduzido por Aplicativo Web Progressivo, este termo é utilizado para metodologia de criação de aplicativos *webs* que são progressivamente aprimorados (MICROSOFT, 2021). Esta metodologia utiliza várias tecnologias e recursos web para criar aplicativos que executem em navegadores, mas que possibilitem ao usuário a experiência de estar utilizando um aplicativo nativo. Em outras palavras, o usuário está acessando uma página que interage com ele com os mesmos recursos de um aplicativo nativo. Os PWAs, não são um conceito novo, e em navegadores modernos, já tem sido possível usufruir de muitas de suas técnicas. Atualmente é possível utilizar vários recursos que permitem uma melhor usabilidade ao usuário (MOZILLA, 2021).

Os PWAs são possíveis graças a evolução da web, mudando de HTML estático para documentos dinâmicos como PHP, ASP.Net, Ajax entre outro rodando do lado do servidor e funcionalidades introduzidas no HTML5, CSS3 e JavaScript. Esse desenvolvimento nessas linguagens proporcionaram interações personalizadas com elementos da página chegando até renderizações de vídeos, sem o uso do Adobe Flash Player, e jogos (WIKIPEDIA, 2021).

Podemos destacar algumas características do PWA como:

- Progressivos, funcionando em qualquer navegador e são aprimorados progressivamente como princípio fundamental.
- Responsivos, se adequam a qualquer formato seja celular, *tablet*, *notebook*, *desktop*, etc.
- Independente de conectividade, podem utilizar o *Service Workers* para trabalhar offline ou em redes de baixa qualidade.
- Semelhantes a aplicativos, pois são compilados no modelo de shell de aplicativos, sua navegação é idêntica à de um aplicativo nativo, interações e navegação trazem a mesma experiência de um aplicativo nativo.
- Atual, podem estar sempre atualizados graças a atualização proporcionada pelo *service worker*.
- Seguros, pois são fornecidos apenas via *https* para evitar invasões e garantir a integridade dos dados.
- Descobrível, podem ser identificados como aplicativo pelo uso dos manifestos W3C e o escopo de registro do *service worker* que permitem que os mecanismos de pesquisa os encontrem.
- Instalável, pois permite que seja criado um acesso rápido ao aplicativo como ícone na tela inicial do dispositivo. Para prover esta interação com o usuário e atender os conceitos da metodologia PWA são necessário o uso de algumas tecnologias que podem ser encontrados nos navegadores mais modernos (MOZILLA, 2021).

2.7.1 *Service worker*

Recursos que anteriormente eram fornecidos apenas por aplicativos nativos como: experiências offline, sincronizações em segundo plano, *notificações push*, hoje são possíveis na web através do *Service Worker*. Um *Service Worker*, nada mais é do que um script que executa em segundo plano separado da página. Como script, rodando em segundo plano, ele trabalha como um *proxy* de rede programável permitindo controlar as solicitações de rede da página. Ele não possui acesso ao DOM diretamente mas manipulam respondendo as mensagens enviadas pela interface *postMessage* das paginas que ele gerencia.

O ciclo de vida de um *Service Worker* é separado da página. Ele começa com o registro no navegador através de códigos JavaScript na página, após o registro o navegador iniciará a etapa de instalação em segundo plano. Nesta etapa da instalação alguns recursos estáticos podem ser armazenados em cache. Se todos os arquivos forem armazenados corretamente o *Service Worker* é instalado, caso ocorra algum problema nessa etapa e algum dos arquivos não sejam baixados, ele não será instalado e haverá uma nova tentativa na próxima vez em que a página for acessada. Após ter sucesso na etapa de instalação iniciará a etapa de ativação, nesta etapa é possível fazer o gerenciamento de caches antigos.

A etapa seguinte é a etapa de ativação, onde o *Service Worker* poderá controlar as páginas dentro do escopo. Porém, a página que o registrou só será controlada ao ser carregada novamente.

As opções de navegadores que suportam essa tecnologia estão crescendo. Já há compatibilidade com os navegadores mais usados como Chrome, Firefox, Opera e Edge (GAUNT, 2021).

2.7.2 *HTTPS*

Quando utiliza-se o *service worker* para desenvolvimento, por meio do *localhost*, ainda é possível executar o *service worker* sem *HTTPS*, porém para implantação é necessário ter o *HTTPS* configurado no servidor. O *service worker* pode fazer inúmeros processos como: sequestrar conexões, fabricar e filtrar respostas, assim, ele pode ser uma ferramenta ótima para ser utilizada em um bom motivo mas também pode ser usada, por um intermediário, para um motivo bem diferente. O uso do *HTTPS* assegura que o arquivo não foi adulterado antes de chegar até o cliente.

2.7.3 *Manifest web*

O arquivo *Manifest* é um script escrito em javascript baseado no formato *JavaScript Object Notation (JSON)* que proporciona um lugar centralizado para guardar os metadados associados com a aplicação web. Este arquivo não é limitado apenas a informações como: nome do arquivo, link dos ícones, urls preferenciais para abrir a aplicação mas também permite declarar a orientação que se quer e o modo do *display* que a aplicação irá executar. Com o uso deste arquivo o desenvolvedor pode prover ao usuário uma experiência comparável a uma aplicação nativa.

O *Manifest* é comumente usado quando se quer instalar um aplicativo web porém com um novo modo de instalação visto que a instalação não será como aplicativos nativo. Isto distingue um aplicativo web utilizando o *Manifest* de um simples site salvo nos favoritos de um navegador. Por exemplo, uma página utilizando um *Manifest* web criará um ícone na tela inicial dando acesso ao aplicativo web, este, pode estar configurado para abrir como *display fullscreen* sem a caixa de pesquisa comum dos navegadores. Assim este aplicativo web se torna indistinguível de um programa nativo em sua página inicial para o usuário.

Quando um *Manifest* é lincado a um documento, isso indica que ele é autoritário, ou seja, deve se usar os metadados contidos neste arquivo ao invés do que estão no documento. Porém se por algum motivo os metadados forem perdidos será usado as informações contidas no documento (W3C, 2021).

2.7.4 Promise

O JavaScript usa um *thread* apenas, assim ele não pode executar duas partes do script ao mesmo tempo. É necessário seguir uma seqüência e ainda compartilha esta *thread* com muitos outros itens que variam para cada navegador. A *promise* permite contornar esta situação pois se trata de um objeto utilizado em processo assíncrono para proporcionar um resultado síncrono no script.

Sintaxe: `new promise(/* executor */ function(resolve, reject) ...)`. Das funções pode-se citar: *Executor*: É uma função que recebe dois argumentos *resolve* e *reject*. Ela é executada pela implementação do *promise*, passando a função *resolve* ou *reject*. A função *executor* é chamada antes que o construtor da *Promise* retorne o objeto. *Resolve*: Quando há sucesso, a função *resolve* é chamada para resolver a *promise*. *Reject*: Quando há erro, é chamada a função *reject* que rejeitará a *promise*.

O trabalho dentro do *executor* é assíncrono, quando ele é concluído, chama uma das funções, *resolve* ou *reject*, para definir o estado da *promise*.

O grande trunfo da *promise* é fazer com que métodos assíncronos retornem valores como métodos síncronos. O método assíncrono nesse caso não retorna um valor final, mas uma promessa ao valor em algum momento do futuro.

Agindo assim, a *promise* é praticamente um *proxy* para um valor que não é ainda conhecido quando a *promise* é criada. Então são associados métodos de tratamento para eventos da ação assíncronas num caso de sucesso ou falha.

As *promises* possuem 4 estados:

- *pending* (pendente): Estado inicial, que não foi realizada nem rejeitada.
- *fulfilled* (realizada): sucesso na operação.
- *rejected* (rejeitado): falha na operação.
- *settled* (estabelecida): Que foi realizada ou rejeitada.

A *promise* no status *pending* pode se tornar realizada, recebendo um valor ou rejeitada por algum motivo, como um erro. Quando algum desses estados ocorrem a *promise* chama o método *then()*. Esse método, por sua vez, chamará o método de tratamento associado ao estado(*reject* ou *resolved*) (MOZILLA, 2021).

2.8 Evolução dos navegadores

Todos os sistemas evoluem, sofrem, de alguma maneira, alterações e melhoramento ao longo do tempo. As mutações genéticas que ocorrem durante a evolução dos seres vivos permitem o surgimento de novas espécies. Diferentemente, a evolução nos sistemas computacionais a evolução ocorre de tal forma que novas funcionalidades são implementadas ao sistema, sendo assim, as versões anteriores podem ser extintas ou modificadas. Nota-se que há uma somatória de novas funcionalidades que são adicionadas ao longo do tempo, ou seja, os sistemas não são “criados do zero” mas sim são transformações, adaptações e melhoramento de sistemas pre existentes (CALVÃO, 2011).

Os navegadores de internet passaram por evoluções ao decorrer do tempo. Um navegador pode ser definido como um programa que permite a interação de seus usuários com documentos HTML hospedados em um servidor de rede. As características comuns aos navegadores incluem interfaces que voltam à página anterior ou ir à página posterior, recarregar a atual, espaço para o endereço URL, favoritos, e o conceito das abas, além de manterem um histórico dos sites navegados. A principal finalidade desses softwares é a busca por determinado conteúdo na Web e a exibição dos resultados para esta busca.

O uso dos navegadores se iniciou com Tim Bernes-Lee com o *World Wide Web* (www), que por meio de hipertexto, criou o primeiro navegador para compartilhar informações. A grande popularidade da web ocorreu com a NCSA, que era um navegador gráfico.

Em 1994 o Navigator foi lançado pela Netscape, tornando-se popular no ano seguinte, sendo esta empresa

responsável pela popularização da Web. A Guerra dos navegadores web, que ocorreu entre 1995 e 1999, foi o período em que a Netscape, até então líder absoluta no mercado de softwares, perde a liderança para a Microsoft, detentora do internet explorer. Este período foi responsável por colocar a internet a disposição de milhões de usuários, mas também mostrou como esta disputa poderia arruinar os esforços de padronização dos navegadores. O ano de 1998 marcou o fim desta disputa, pois o domínio do mercado foi conquistado pela Microsoft, em parte pelas ações da empresa em integrar seu navegador ao sistema operacional. Em consequência a Netscape liberou seu produto como código aberto, criando o Mozilla, acelerando o declínio da companhia. Este período foi importante na área da ciência da computação pois marcou a transição do uso de um software para outro, e também foi marcado pelo surgimento dos projetos como o Mozilla e Opera. A introdução das linguagens HTML5 e CSS3, adicionou extensos scripts e permitiu o uso de outros dispositivos para a navegação da internet, como o uso dos *smartphones*, por exemplo.

O HTML5 foi apreciado pelos desenvolvedores Web, e foi foco de discussões na mídia em abril de 2010 e seu desenvolvimento implicaria na desuso do Adobe Flash. Em meados de 2011 a Adobe declarou interromper o desenvolvimento do Flash voltando seu foco no desenvolvimento de ferramentas empregando o HTML5 (WIKIPEDIA, 2019). Atualmente o Explorer tem sido um dos navegadores mais empregados, embora o futuro parece pertencer ao Safari, navegador próprio da Apple, baseado em KHTML.

Os navegadores comunicam-se geralmente com vários tipos de servidores de rede, por meio de protocolos de transferência de hipertexto como o HTTP, além de serem capazes de ler vários tipos diferentes de arquivos, sendo os mais comuns HTML, XML, JPEG, GIF, PNG, Flash, Java, etc.

2.9 Javascript

O JavaScript é uma linguagem de programação de alto nível, interpretada e não tipada, ou seja, não importa o tipo de dado que está contido na variável. Ela é uma linguagem muito utilizada no desenvolvimento Web e possui interpretador na maioria dos navegadores modernos. O termo JavaScript na verdade é uma marca registrada, licenciada pela *Sun Microsystems(Oracle)*. Esta linguagem foi enviada para a ECMA - *European Computer Manufacture's Association* para padronização pela Netscape, criadora da linguagem. Na versão padronizada recebeu o nome de ECMAScript e por motivos de ligados a marca registrada, recebeu o nome de JScript na versão da Microsoft (FLANAGAN, 2013). Através da linguagem JavaScript é possível desenvolver trechos de programação em uma página Web com o objetivo de validar dados, modificar dinamicamente valores, estilos, criar operações e manipulações junto ao navegador possibilitando controlar o comportamento da página. Dentro das linguagens webs que trabalham no lado do servidor ou do lado do cliente, o javascript está no grupo das linguagens que trabalham no lado do cliente (MILETTO E. M.; BERTAGNOLLI, 2014).

2.9.1 Bibliotecas Javascript

Esta linguagem de programação teve várias alterações no decorrer dos anos, houveram muitas melhorias porém houveram muitas dificuldades de compatibilidade também. Para resolver esses problemas foram criadas várias bibliotecas que além de tratar as incompatibilidades ainda proporcionaram utilizar um potencial maior da linguagem (DUARTE, 2015). O Javascript possui várias bibliotecas que podem ser criadas para determinados tipos de necessidades como por exemplo a biblioteca Instascan e Lovefield.

A utilização da biblioteca Instascan proporciona o acesso a câmera do dispositivo e sua manipulação. Com ela é possível monitorar as imagens e decodificar *QR Code*. Ela está sob licença livre tanto para software livre quanto para proprietário (SCHMICH, 2020).

A biblioteca Lovefield por sua vez, provê a possibilidade de criação de um banco de dados relacional no navegador cuja sintaxe é semelhante a linguagem SQL. Sua licença é permissiva e possibilita a utilização em softwares

livres e de uso comercial (GITHUB, 2020).

2.10 CSS

CSS é uma sigla, do termo em inglês *Cascading Style Sheets*, que significa Folha de Estilo em Casacatas. Em outubro de 1994 surgiu a primeira proposta de CSS, por Hankin Li, com o objetivo de facilitar a programação de sites, pois era necessário o emprego de muitos códigos para se obter resultados simples.

Seu desenvolvimento foi realizado em 1995 pela W3C, desenvolvendo primeiramente o CSS1, motivados pela ausência de tags simples no HTML para formatar as páginas. Em 1997 e 1999 a linguagem recebeu destaque e ficou conhecida entre os programadores (PACIEVICH, 2021). Há uma forte relação entre o HTML e o CSS e pode-se dizer que o HTML é o “alicerce do site” e o CSS seu “estilo”. O CSS é aplicado para estilizar aspectos de imagem dos sites de linguagem HTML, separando os conteúdos da representação visual do site. Pode-se dizer de maneira simples que o CSS é utilizado para a decoração da página web, é uma linguagem que determina a “aparência” do site, alterando fontes e cores de fundo e texto, inserindo e ajustando imagens, controlando elementos como margem, linhas, cores, alturas, posicionamento, etc. Pode-se dizer que a utilização da linguagem CSS não é uma obrigatoriedade, porém, utilizando apenas o HTML, o site ou página irá parecer abandonado.

No HTML toda a parte gráfica e estética da página deveria ser incluída na marcação, ou seja, era necessário escrever separadamente todos os detalhes estéticos. Porém com a utilização do CSS, toda a parte “estética” da página é escrita em arquivo separado do conteúdo, proporcionando a marcação HTML limpa e fácil de manter, além de permitir que se tenha mais de um estilo em página HTML (PACIEVICH, 2021).

É possível criar diversas funcionalidades utilizando CSS no lugar de linguagens mais pesadas como JavaScript por exemplo. Pode-se criar por exemplo animações complexas, inserir profundidade nas imagens adicionadas, criar sites interativos e jogos com HTML5 e CSS3 (GONÇALVES, 2021). Das vantagens que surgem do emprego do CSS pode-se citar: o controle de vários documentos HTML; a facilidade na criação de layouts com códigos simples; possui linguagem simples e de fácil aprendizado (PACIEVICH, 2021).

2.11 Raspagem de dados

Raspagem de dados ou *Data Scraping* é o nome dado a técnica de aquisição de dados não estruturados de forma automatizada. Quando essa técnica é utilizada para páginas na internet recebe o nome de *Web Scraping* (BORGES THIAGO DA CUNHA; GANIMI, 2018). Esta técnica é amplamente utilizada para análise de informações que não estão acessíveis para abstração de forma intencional. Muitas dessas informações estão inseridas nas camadas HTML, CSS, Javascript das páginas e podem ser obtidas utilizando programação específica (SILVA, 2019).

2.12 Banco de dados indexeddb

IndexedDB trata-se de uma API de armazenamento que suporta uma quantidade significativa de informações possuindo uma alta performance em buscas utilizando index. Esta API utiliza o sistema chave-valor para armazenamento dos dados e executa qualquer ação através de transações. As informações salvas ficam do lado do cliente mais especificamente no navegador, dessa forma é possível a criação de aplicativos webs que podem funcionar online e offline.

O IndexedDB não possui limite de tamanho porém em navegadores como Firefox pode ser necessário permissão ao ultrapassar a marca de 50mb (MOZILLA, 2017).

A Figura 3 mostra a compatibilidade da API indexedDB com os principais navegadores em desktops.

Figura 3 – Compatibilidade entre API Indexeddb com navegadores desktops

Característica	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari (WebKit)
API Assíncrona	11.0 webkit (en-US) 24	4.0 (2.0) moz (en-US) 16.0 (16.0)	10	17	11.1

Fonte: Mozilla2021

A API também possui compatibilidade com os navegadores desenvolvidos para dispositivos mobile, conforme a Figura 4.

Figura 4 – Compatibilidade entre API Indexeddb com navegadores mobile

Feature	Android	Firefox Mobile (Gecko)	IE Phone	Opera Mobile	Safari Mobile
API Assíncrona	4.4	6.0 (6.0) moz (en-US) 16.0 (16.0)	10	17	10.3

Fonte: Mozilla2021

3 DESENVOLVIMENTO

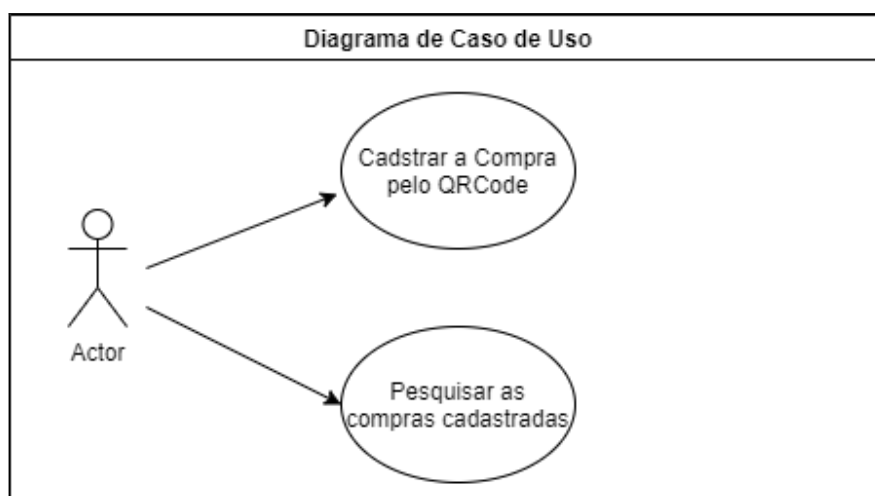
3.1 Descrição do projeto

O trabalho desenvolvido consiste em um site para cadastrar e pesquisar as compras que possuem nota fiscal do consumidor do Paraná, através do QR Code impresso na nota. Este site foi desenvolvido utilizando a metodologia PWA possibilitando a transformação do mesmo em um aplicativo web com funcionamento offline. Com estas funcionalidades acredita-se ser possível ao usuário salvar suas informações de compra, controlar seus gastos e monitorar a variação de preço dos produtos cadastrados.

3.2 Caso de uso

A Figura 5 representa o diagrama de caso de uso para o site, onde o Ator terá as opções de cadastrar uma nova compra através do QR Code e pesquisar as compras cadastradas anteriormente.

Figura 5 – Diagrama de caso de uso

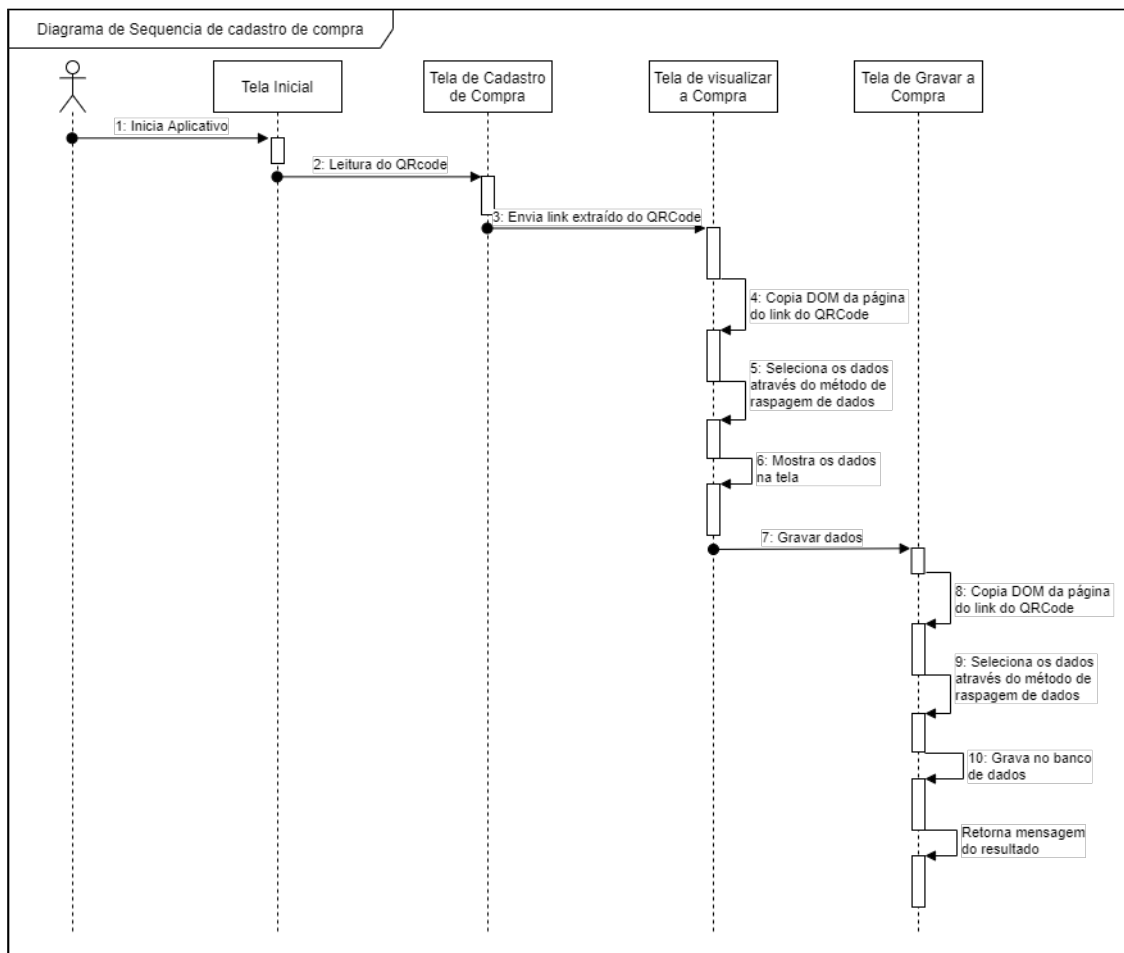


Fonte: Autoria própria (2021)

3.3 Diagrama de sequência

Os diagramas de sequência mostram o fluxo do sistema de maneira gráfica proporcionando o entendimento do resultado das ações executadas. A Figura 6 ilustra o diagrama de sequência do cadastro de compras que é iniciada ao selecionar a opção "Cadastrar Compra".

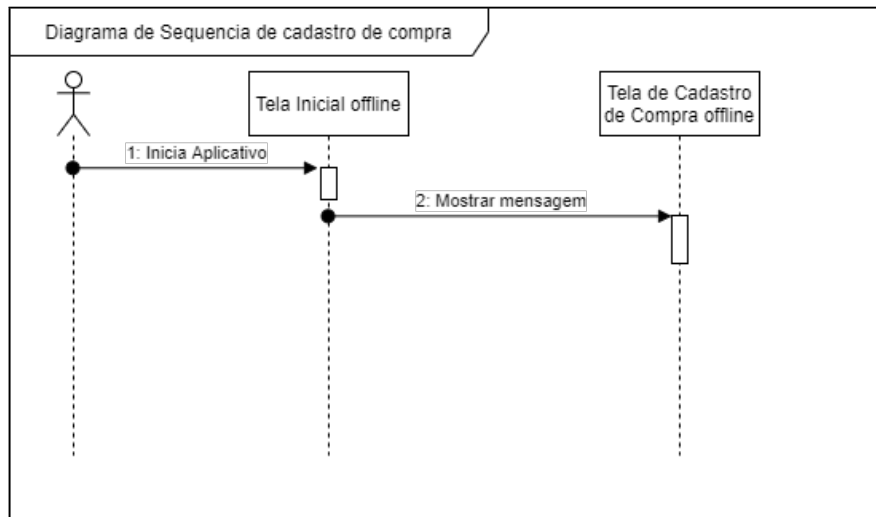
Figura 6 – Diagrama de sequencia do cadastro de compras



Fonte: Autoria própria (2021)

A Figura 7 representa a sequencia do cadastro de compras em uma situação offline.

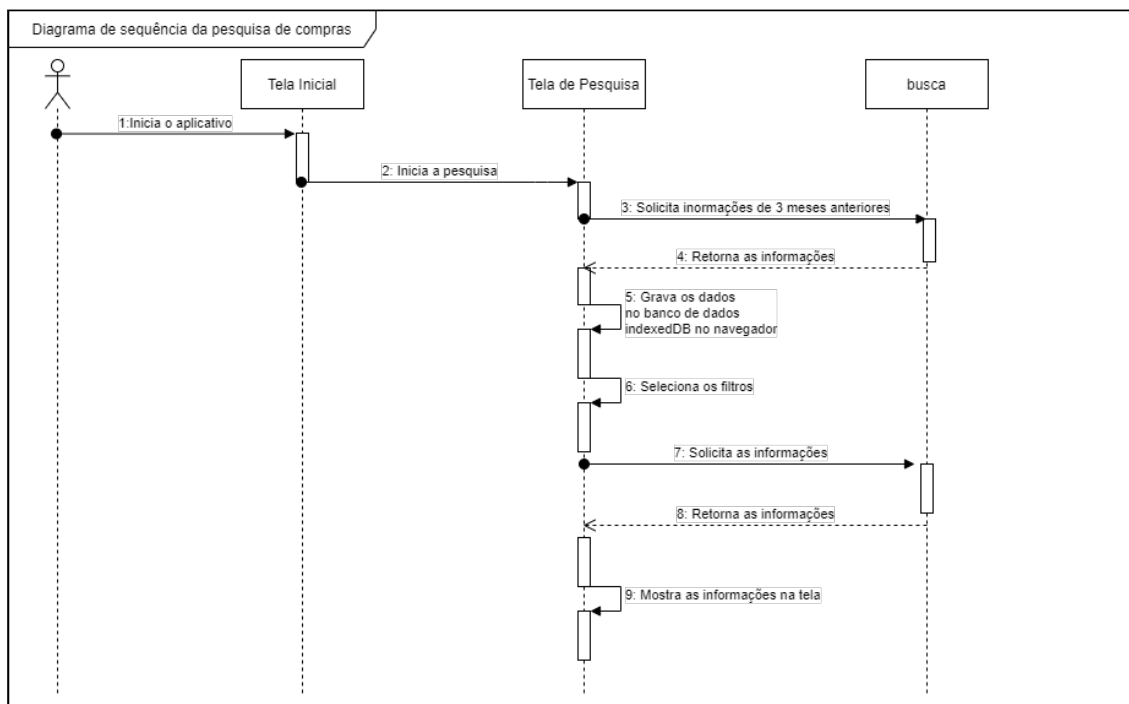
Figura 7 – Sequência de cadastro para compras offline.



Fonte: Autoria própria (2021)

A Figura 8 ilustra o diagrama de sequência da pesquisa.

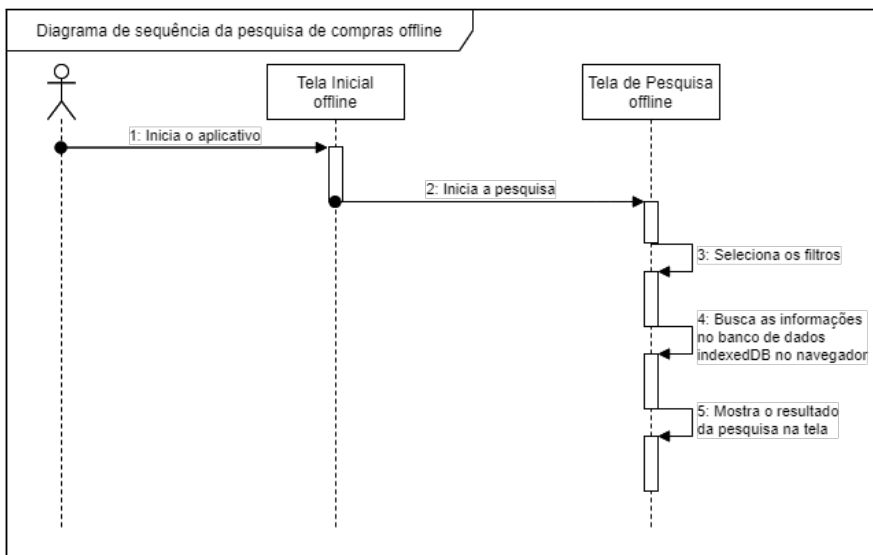
Figura 8 – Diagrama de sequencia do processo de pesquisa.



Fonte: Autoria própria (2021)

O diagrama de sequencia da pesquisa offline é ilustrado na Figura 9.

Figura 9 – Diagrama de seqüência do processo de pesquisa offline.



Fonte: Autoria própria (2021)

3.4 Estrutura do sistema

O site foi desenvolvido utilizando a metodologia PWA, desta forma criou-se os arquivos com programação Service Worker e Manifest para garantir que o site seja instalável, estável e assim possibilitar a transformação do site em um aplicativo web. Cada página possui aspectos que colaboram para que o site seja inserido na categoria de aplicativo web.

3.5 Service worker

O arquivo sw.js é um arquivo com códigos em javascript que funciona como o service worker do site . Através dele é possível programar o comportamento do site, em nosso projeto ele monitora as requisições das páginas para redirecioná-las caso haja alguma falha. Para a utilização do arquivo service worker é necessário registrá-lo no navegador pelo método navigator.serviceWorker.register(). Na Figura 10 pode-se verificar como este código é construído.

Figura 10 – Código de registro do Service Worker.

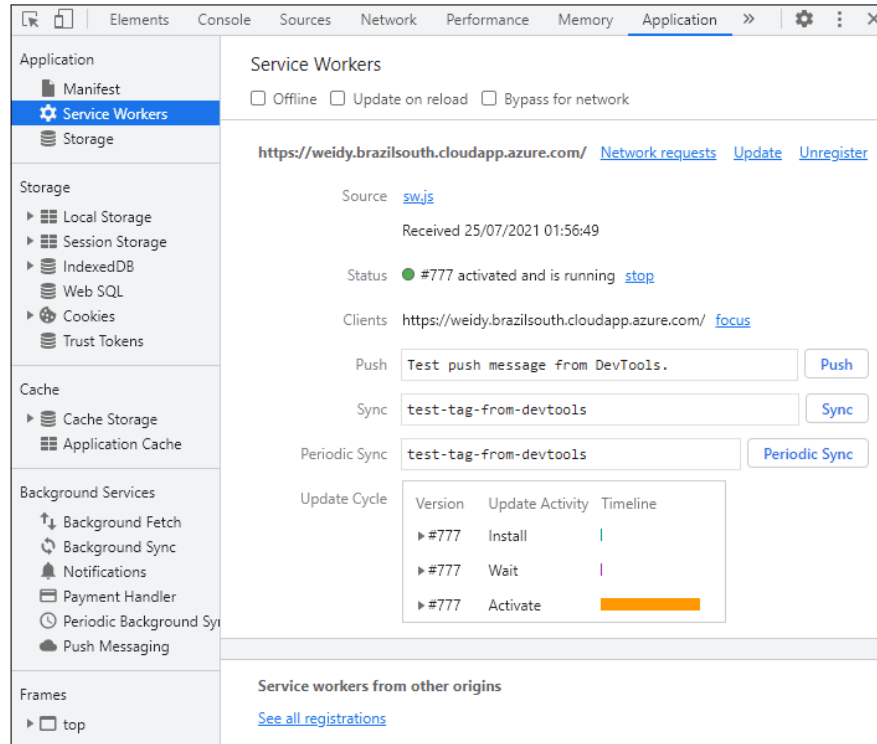
```

if (navigator.serviceWorker) {
  navigator.serviceWorker.register('./sw.js')
  .then(function (registration) {
    console.log("SW funcionando, registrado com o escopo:
    ", registration.scope);
  })
  .catch(function () {
    console.warn("service worker failed");
  });
}
    
```

Fonte: Autoria própria (2021)

No navegador Chrome, no setor de inspeção, é possível verificar na aba “Application”, opção “Service Workers” o resultado do registro, conforme a Figura 11.

Figura 11 – Service Worker registrado.



Fonte: Autoria própria (2021)

Após registrado o código inicia a execução criando um cache no navegador com o método `caches.open()` e na sequência é adicionado as páginas que serão utilizadas posteriormente no caso de falha na requisição das páginas principais, com o método `cache.addAll()`. O código para esse procedimento pode ser observado na Figura 12.

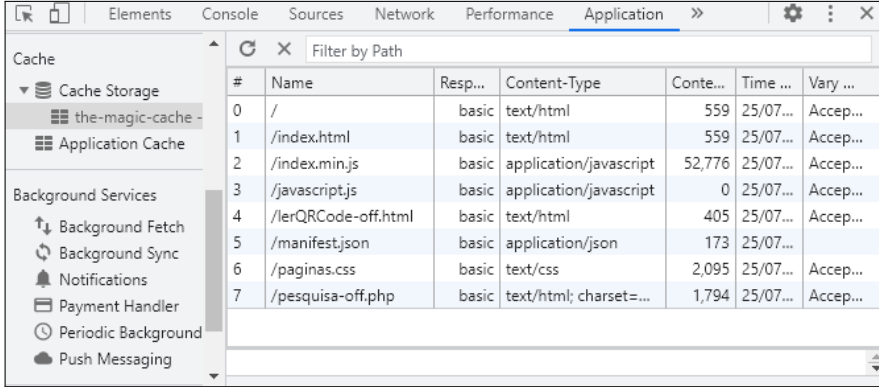
Figura 12 – Código para a criação de cache no navegador.

```
self.addEventListener('install', function(e) {
  e.waitUntil(
    caches.open('the-magic-cache').then(function(cache) {
      return cache.addAll([
        '/',
        '/index.html',
        '/manifest.json',
        '/lerQRCode-off.html',
        '/pesquisa-off.php',
        '/index.min.js',
        '/javascript.js',
        '/paginas.css',
      ]);
    });
});
```

Fonte: Autoria própria (2021)

A Figura 13 apresenta o resultado do código de criação do cache assim como a adição das páginas no navegador.

Figura 13 – Cache do navegador Chrome



The screenshot shows the Chrome DevTools Application tab with the Cache Storage section expanded. The 'the-magic-cache' is selected, and a table lists the cached resources. The table has columns for #, Name, Resp..., Content-Type, Conte..., Time..., and Vary... The resources listed are: / (559 bytes, text/html), /index.html (559 bytes, text/html), /index.min.js (52,776 bytes, application/javascript), /javascript.js (0 bytes, application/javascript), /lerQRCode-off.html (405 bytes, text/html), /manifest.json (173 bytes, application/json), /paginas.css (2,095 bytes, text/css), and /pesquisa-off.php (1,794 bytes, text/html; charset=...).

#	Name	Resp...	Content-Type	Conte...	Time ...	Vary ...
0	/	basic	text/html	559	25/07...	Accep...
1	/index.html	basic	text/html	559	25/07...	Accep...
2	/index.min.js	basic	application/javascript	52,776	25/07...	Accep...
3	/javascript.js	basic	application/javascript	0	25/07...	Accep...
4	/lerQRCode-off.html	basic	text/html	405	25/07...	Accep...
5	/manifest.json	basic	application/json	173	25/07...	Accep...
6	/paginas.css	basic	text/css	2,095	25/07...	Accep...
7	/pesquisa-off.php	basic	text/html; charset=...	1,794	25/07...	Accep...

Fonte: Autoria própria (2021)

Com as páginas salvas em cache, o Service Worker, através do método fetch, monitora todas as requisições definindo as páginas em cache que, respectivamente, seriam usadas em caso de falha na requisição. Como mostra o trecho de código na Figura 14, caso haja qualquer tipo de falha na requisição da página de pesquisa, o service worker redirecionará para a página pesquisa-off.php. Caso a falha seja na requisição da página “cadastrar compra”, o código redirecionará para cadastrar_compra-off.html, porém se a falha ocorrer na requisição da página index.html ela redirecionará para uma cópia sua salva em cache.

Figura 14 – Códigos de redirecionamento para possíveis falhas de requisição.

```
self.addEventListener("fetch", event => {  
  
  if(event.request.url == 'https://www.controlecompras.com/pesquisa.php'){  
    event.respondWith(  
      fetch(event.request).then(response => {  
        return fetch(event.request)  
      })  
      .catch(() => {  
        return caches.match('pesquisa-off.php');  
      })  
    )  
  }else if(event.request.url == 'https://www.controlecompras.com/lerQRCode.html'){  
    event.respondWith(  
      fetch(event.request).then(response => {  
        return fetch(event.request)  
      })  
      .catch(() => {  
        return caches.match("lerQRCode-off.html");  
      })  
    )  
  }else{  
    event.respondWith(  
      caches.match(event.request)  
      .then(response => {  
        return response || fetch(event.request);  
      })  
      .catch(() => {  
        return caches.match("/index.html");  
      })  
    )  
  }  
});
```

Fonte: Aatoria própria (2021)

3.6 Manifest

O arquivo manifest.json foi utilizado para criar o layout de aplicativo para o site. Através dele foi possível indicar como deve ser o formato do site quando este for adicionado ao smartphone ou computador como um aplicativo web. O código define que o formato deve ser sem barra de navegação através da opção “display = standalone”, o nome do aplicativo é configurado pela tag "name": "Controle de Compras". Outras opções também são definidas neste arquivo como ícone e cor de fundo. A Figura 15 exibe o conteúdo necessário para essas configurações.

Figura 15 – Trecho do arquivo Manifest.

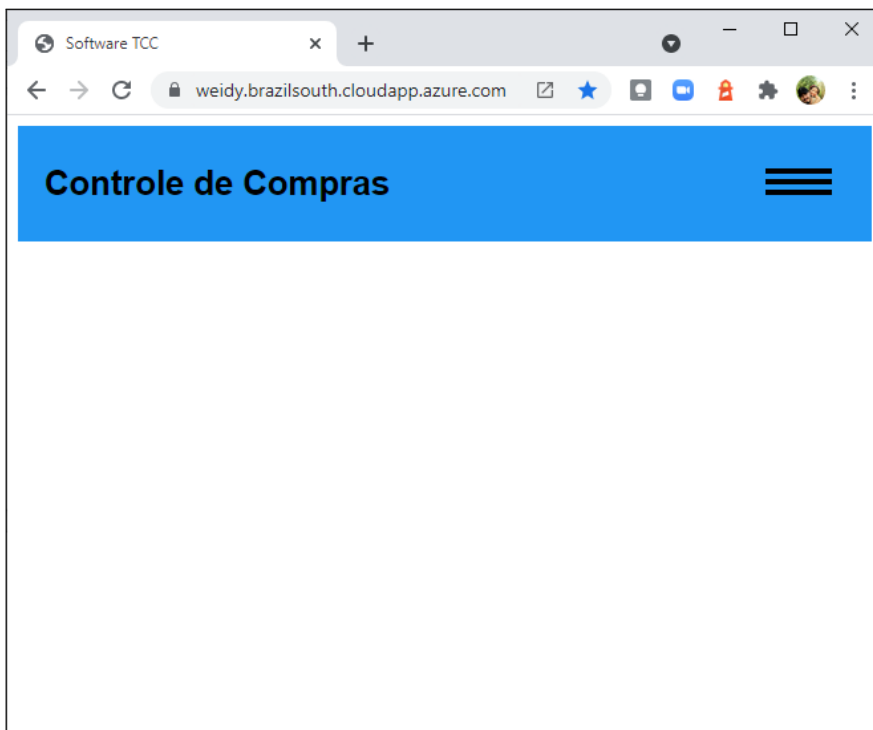
```
{
  "name": "Controle de Compras",
  "theme_color": "#2196f3",
  "background_color": "#2196f3",
  "display": "standalone",
  "Scope": "/",
  "start_url": "/",
  "splash_pages": null,
  "icons": [{
    "src": "img/icon.png",
    "sizes": "192x192",
    "type": "img/png"
  },
  {
    "src": "img/icon-grande.png",
    "sizes": "592x592",
    "type": "img/png"
  }
]
```

Fonte: Autoria própria (2021)

3.7 Index

A página index.html é a página inicial do sistema, nesta página são carregados os arquivos de service worker e manifest.json. A Figura 16 mostra a página inicial do sistema.

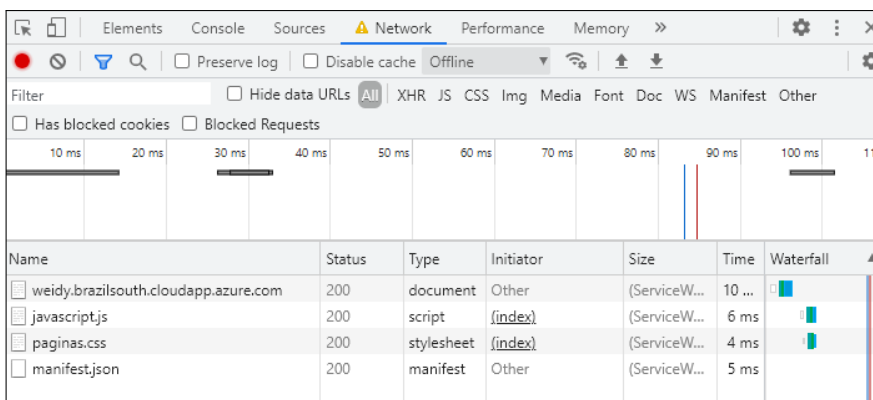
Figura 16 – Página inicial do sistema de controle de compras.



Fonte: Autoria própria (2021)

Esta mesma página é copiada para o cache do navegador e utilizada em caso de falhas na requisição. Fez-se a simulação do status offline da página e o redirecionamento foi executado com sucesso, conforme apresentado na Figura 17.

Figura 17 – Simulação do status offline da página index.html.

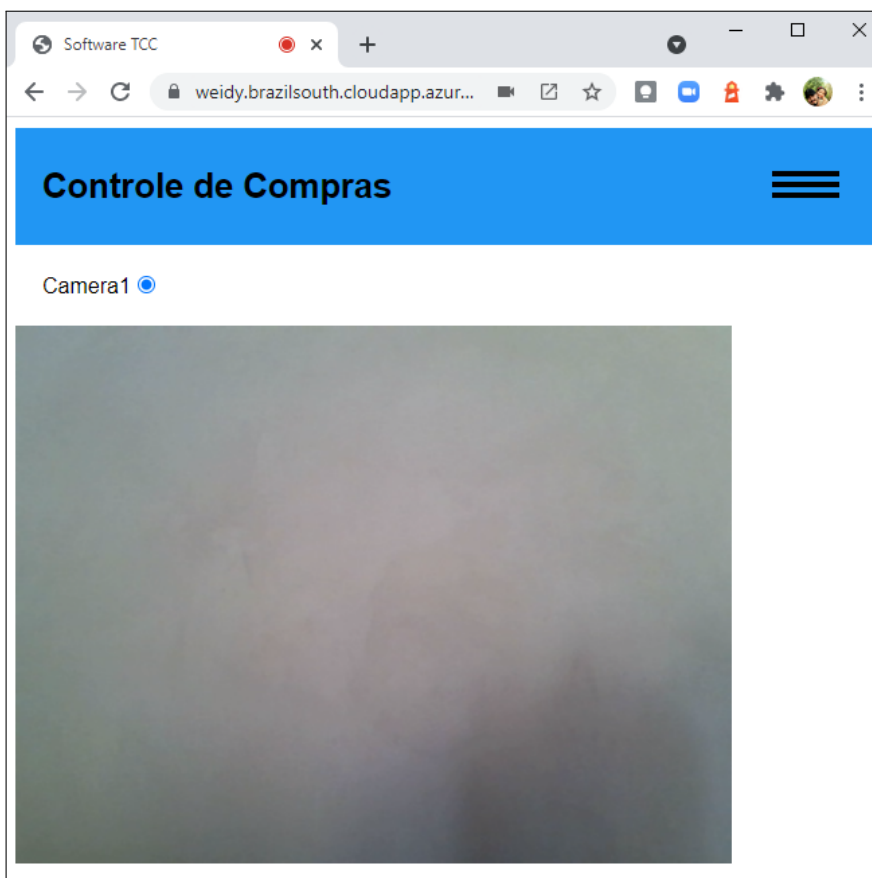


Fonte: Autoria própria (2021)

3.8 Cadastrar compra

Acessando a opção do menu cadastrar compra o link direciona para página cadastrar_compra.html. A Figura 18 mostra a página com a câmera inicializada.

Figura 18 – Página com a câmera inicializada.



Fonte: Autoria própria (2021)

Nesta página está presente a biblioteca Instascam na qual foi possível, através do método `getVideoInputDevices()`, verificar as câmeras do sistema e criar um elemento radio para a seleção da câmera a ser utilizada para leitura do QRCode. Na Figura 19 é possível observar onde ocorre a captura dos dispositivos.

Figura 19 – Métodos da biblioteca Instascam.

```
codeReader.getVideInputDevices()
    .then((videolnputDevices) => {
        selectedDeviceld = videolnputDevices[0].deviceld
        const sourceSelect = document.getElementById('checkboxVideo')
        if (videolnputDevices.length >= 1) {
            var i = 1;
            videolnputDevices.forEach((element) => {
                const sourceOption = document.createElement('input')
                const labelCam = document.createElement('span');
                sourceOption.value = element.deviceld
                sourceOption.type = "radio"
                if(i == 1){
                    sourceOption.checked = true
                }
                sourceOption.name = "check"
                labelCam.setAttribute("class", "badge text-light");
                labelCam.innerHTML = "Camera"+i
                sourceSelect.appendChild(labelCam)
                sourceSelect.appendChild(sourceOption)

                i += 1;
            })
        }
        return selectedDeviceld;
    })
```

Fonte: Aatoria própria (2021)

Após a verificação das câmeras do sistema, a página solicita a permissão de acesso a primeira câmera através do recurso `decodeFromInputVideoDevice()`. Este método monitora a imagem da câmera procurando por um elemento no formato QR Code. Quando a leitura é realizada, o link presente no QR Code é enviado como resposta. Esta resposta é incluída no elemento `input` e enviada para página `mostrar_compra.php`. O trecho de código na Figura 20 mostra como o processo ocorre.

Figura 20 – Código para processo de leitura do QR Code pela câmera.

```
codeReader.decodeFromInputVideoDevice(selectedDeviceld
, 'video').then((result) => {

    $("#url").val(result);

    formulario = document.getElementById('form1');

    document.form1.submit();

    })
```

Fonte: Aatoria própria (2021)

3.9 Página para cadastrar compra quando há falhas na requisição

Quando há falhas na requisição da página “cadastrar compra” o service worker redireciona para página cadastrar_compra_offline.html onde há uma mensagem explicativa do motivo pelo qual não esta sendo possível cadastrar a compra. A Figura 21 exibe a tela com a mensagem.

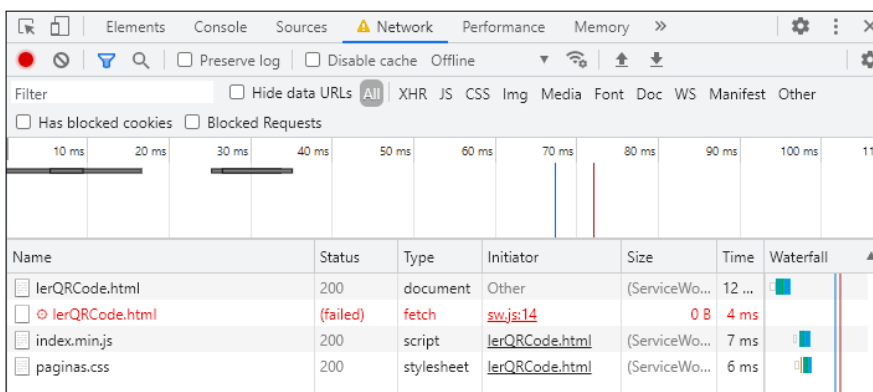
Figura 21 – Página de cadastro de compra offline



Fonte: Autoria própria (2021)

Simulou-se no navegador a situação offline e testou-se a requisição da página cadastrar_compra.html. O redirecionamento foi executado com sucesso como pode ser observado na Figura 22.

Figura 22 – Redirecionamento em situação offline da pagina Cadastrar Compra.



Fonte: Autoria própria (2021)

3.10 Página para mostrar compras

A página “mostrarCompra.php” recebe o link da página da Secretaria da Fazenda do Paraná onde encontra-se a página referente a compra efetuada. A Figura 23 mostra o layout da página quando é exibida no site da SEFAZ do Paraná.

Figura 23 – Página da NFC-e na SEFAZ-PR.

R Jacob Holzmann, 333, - Olarias, Ponta Grossa, PR	
<input type="text" value="Filtrar itens..."/>	
PLANTA NATURA UN (Código: 81572)	Vi. Total
Qtde.:1 UN: Un VI. Unit.: 2,98	2,98
PLANTA NATURA UN (Código: 81572)	Vi. Total
Qtde.:1 UN: Un VI. Unit.: 2,98	2,98
CHOC LACTA 45G (Código: 215771)	Vi. Total
Qtde.:1 UN: Un VI. Unit.: 1,99	1,99
qtd. total de itens: 3	
Valor a pagar R\$: 7,95	
Forma de pagamento: Outras Valor pago R\$: 10,00	
Informação dos Tributos Totais Incidentes (Lei Federal 12.741/2012) R\$ 0,88	
↑ Informações gerais da Nota	
EMISSÃO NORMAL	
Número: 101169 Série: 11 Emissão: 19/11/2020 17:51:13 - Via Consumidor	
Protocolo de Autorização: 141201355465135 19/11/2020 17:51:15	
Ambiente de Produção - Versão XML: 4.00 - Versão XSLT: 2.03	
↑ Chave de acesso	
Consulte pela Chave de Acesso em http://www.fazenda.pr.gov.br/	
Chave de acesso:	
4120 1176 4304 3800 3278 6501 1000 1011 6910 1133 8049	
↑ Consumidor	
Consumidor não Identificado	
Secretaria de Estado da Fazenda do Paraná - Data/Hora da Consulta: 26/07/2021 20:17:19	

Fonte: SEFAZ-PR

A programação desenvolvida copia toda a estrutura e as informações da página acessada pelo link recebido para uma variável utilizando a linguagem php. A Figura 24 exibe o código utilizado para receber as informações.

Figura 24 – Código para obter estrutura do site.

```
$url = $_POST["url"];
$site = file_get_contents($url);
$DOM = new DOMDocument();
libxml_use_internal_errors(true);
$DOM->loadHTML($site);
libxml_clear_errors();
$finder = new DomXPath($DOM);
```

Fonte: Autoria própria (2021)

De posse dessas informações, inicia-se o método de raspagem de dados onde a página é percorrida procurando as informações solicitadas com o método \$finder→query() onde é especificado o trecho do arquivo que se quer copiar. O código \$finderNota→query("//div[@id='infos']"), por exemplo, traz as informações sobre a nota fiscal como: emissão e data. Estas informações copiadas diretamente do site da SEFAZ do Paraná, são salvas em variáveis e informadas ao usuário na mesma tela, conforme apresentado na Figura 25.

Figura 25 – Informações obtidas pelo método de raspagem de dados.

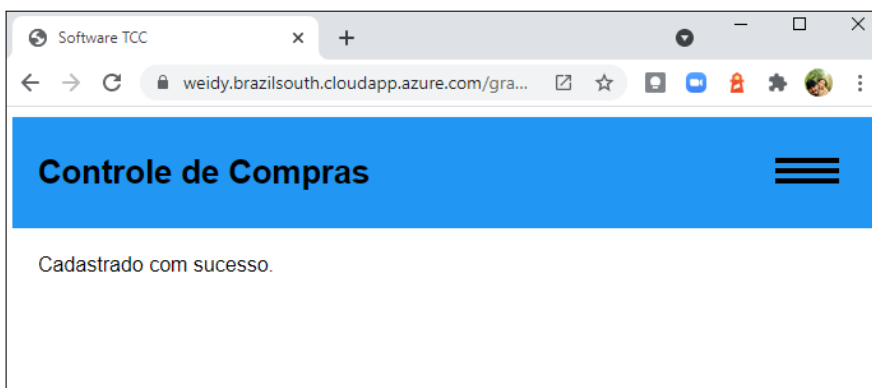


Fonte: Autoria própria (2021)

3.11 Página para gravar a compra

A página “gravarCompra.php” é solicitada ao clicar na opção gravar da página “mostrarCompra.php” onde o processo de raspagem é reiniciado e feito a gravação no banco de dados MYSQL. Caso a gravação ocorra corretamente uma mensagem de sucesso é enviada ao usuário como é mostrado na Figura 26.

Figura 26 – Mensagem enviada quando o cadastro é realizado com sucesso.



Fonte: Autoria própria (2021)

3.12 Página de pesquisa

A página pesquisa.php possui uma execução automática ao ser acessada. Essa execução faz uma pesquisa de informações gerais dos últimos 3 meses no banco de dados MySQL, cria um database no banco de dados IndexedDB do navegador e salva as informações no mesmo. O objetivo deste processo é manter uma base de dados para as pesquisas offline. Para esta gravação no banco de dados IndexedDB é utilizada a biblioteca Lovefield. O processo consiste em deletar o database antigo do IndexedDB com o método `indexedDB.deleteDatabase()` e criar um novo database com as informações atualizadas. A Figura 27 exibe o conteúdo utilizado para esse processo.

Figura 27 – Exclusão e criação de database na API IndexedDB

```
indexedDB.deleteDatabase('id16413066  
_compras');  
var schemaBuilder =  
lf.schema.create('id16413066_compras');
```

Fonte: Autoria própria (2021)

Esta criação é facilitada com o auxílio da biblioteca Lovefield onde utilizamos os métodos `lf.schema.create()` para criação do novo banco, o método `createTable()` é empregado para a criação das tabelas e `addColumn()` para adicionar os campos da tabela. A Figura 28 apresenta o trecho do código utilizado para criação da tabela nota, as demais criações de tabelas seguem o mesmo modelo.

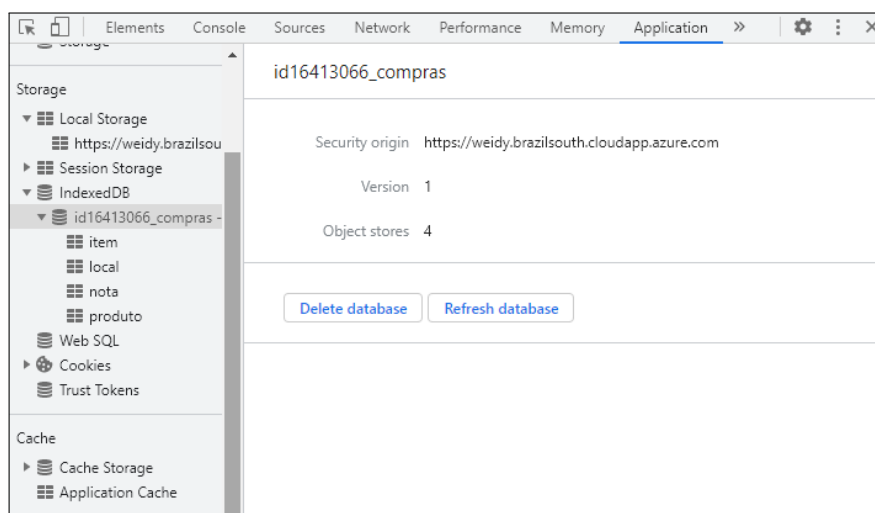
Figura 28 – Trecho do código para a criação da Tabela na API IndexedDB.

```
schemaBuilder.createTable('nota').  
  addColumn('chave', if.Type.STRING).  
  addColumn('cnpj', if.Type.STRING).  
  addColumn('data', if.Type.DATE_TIME).  
  addPrimaryKey(['chave']).  
  addIndex('by_cnpj', ['cnpj'], false, if.Order.DESC);
```

Fonte: Autoria própria (2021)

Após criado, conecta-se o banco com o método `connect()` e copia-se a estrutura do banco. Com esta estrutura utiliza-se o método `insertOrReplace()` para inserir as informações no banco. Na Figura 29 é possível verificar o banco criado com suas respectivas tabelas.

Figura 29 – Banco de dados indexedDB criado com as respectivas tabelas.



Fonte: Autoria própria (2021)

A página de pesquisa apresenta os seguintes filtros : -Local, Nome do Estabelecimento onde foi feita a compra.

-Produto, Nome do produto incluído nas compras.

-Data Inicial, Data mínima para pesquisar as notas fiscais)

-Data Final, Data máxima para pesquisar as notas fiscais)

Criou-se um botão de busca para iniciar a pesquisa. Esta é realizada diretamente no banco de dados MySQL fazendo uma requisição Ajax, com os dados dos filtros, ao arquivo `resultado.php`, o qual conecta no banco de dados MySQL e retorna com as informações da pesquisa. Esta resposta é mostrada na mesma página de pesquisa como mostra a Figura 30.

Figura 30 – Pagina de pesquisa online

Local	togo
Produto	ferrero
Data Inicial	01/03/2021
Data Final	31/03/2021

Busca

TOGO UVARANAS

Produto	Quantidade	Valor	Data
ART Ferrero 2L	1.000	15.90	20/03/2021

Fonte: Autoria própria (2021)

3.13 Página de pesquisa quando há falha na requisição

A página pesquisa-off.php em cache possui o mesmo filtro da pagina de pesquisa.php porém a pesquisa é feita no banco IndexedDB do navegador. Para não sobrecarregar o cache do navegador, optou-se por fazer a consulta no banco de dados com o javascript puro ao invés do uso da biblioteca Lovefield. Visto que o javascript é uma linguagem assíncrona, necessitou-se a utilização de promises para orientar o fluxo das informações necessárias na consulta. Na Figura 31 consta o trecho de código da promise inicial da pesquisa.

Figura 31 – Trecho de código da promise inicial da pesquisa.

```
var promisePesquisa = new Promise(function(resolve, reject){
var request = indexedDB.open('idl6413066_compras');
request.onsuccess = function(event){
    var db = event.target.result;
    var transaction = db.transaction(['local', 'nota', 'item', 'produto']);
    var itemStore = transaction.objectStore('item');
    var localStore = transaction.objectStore('local');
    var notaStore = transaction.objectStore('nota');
    var produtoStore = transaction.objectStore('produto');
    var localCursor;
    var notaCursor;
    var itemCursor;
    var ProdutoCursor;
    var resultado = [];
    var resultadoArray = {};

    notaStore.openCursor().onsuccess = function(event){
        notaCursor = event.target.result;
        if(notaCursor){
            if(new Date(notaCursor.value.value.data) >= data1 && new Date(notaCursor.value.value.data) <= data2){
                resultadoArray = {
                    chave: notaCursor.value.value.chave,
                    cnpj: notaCursor.value.value.cnpj,
                    data: notaCursor.value.value.data
                };
                resultado.push(resultadoArray);
            }
            notaCursor.continue();
        }else{
            resolve([localStore, itemStore, produtoStore, resultado]);
        }
    }
});
promisePesquisa.then(function([localStore, itemStore, produtoStore, resultado]){
```

Fonte: Autoria própria (2021)

3.14 Página CSS

Para o layout das páginas foi utilizado um arquivo externo de CSS. O arquivo `paginas.css` é responsável por determinar a aparência do site. O arquivo CSS tem a função de deixar o site responsivo visto que será utilizado em desktops e mobile. Para isso utilizou-se o recurso de `@media` proporcionando a alteração das características do site de acordo com o tamanho da tela. Na Figura 32 pode-se observar o trecho de código correspondente a mudança de largura da tela.

Figura 32 – Código css para alteração de largura da tela.

```

@media only screen and (min-width: 700px) {
  .container-button-responsive{
    display: none;
  }
  .container-links{
    display: block;
    min-width: 34%;
  }
  .container-links li{
    display: inline-block;
  }
  .col-25, .col-75, input[type=submit] {
    width: 100%;
    margin-top: 0;
  }
}
    
```

Fonte: Autoria própria (2021)

3.15 Teste de metodologia PWA

Para verificar se a metodologia PWA foi utilizada com sucesso na página, utilizou-se a ferramenta Lighthouse disponível como extensão do navegador Chrome. Com a extensão instalada, fez-se o teste da página, gerando o relatório de compatibilidade com o método PWA. O teste verifica 5 características do site, sendo elas: Performance, Acessibilidade, Boas praticas, Otimização em mecanismos de pesquisa e metodologia PWA. As métricas utilizadas pelo teste são:

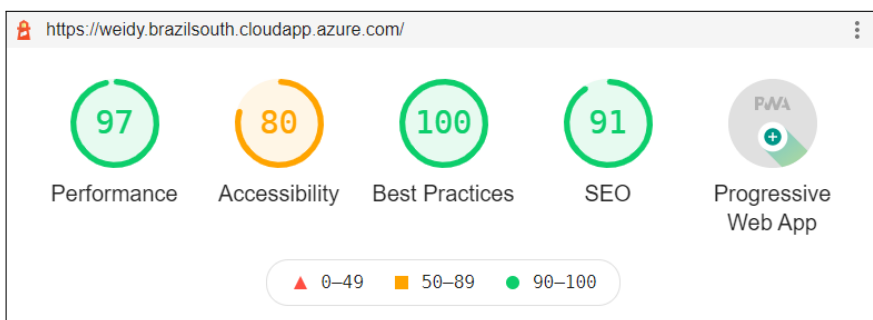
0 to 49 : Pobre

50 to 89: Precisa melhorar

90 to 100 : Bom

A Figura 33 mostra o resultado geral obtido pela ferramenta de auditoria Lighthouse.

Figura 33 – Resultados gerais.



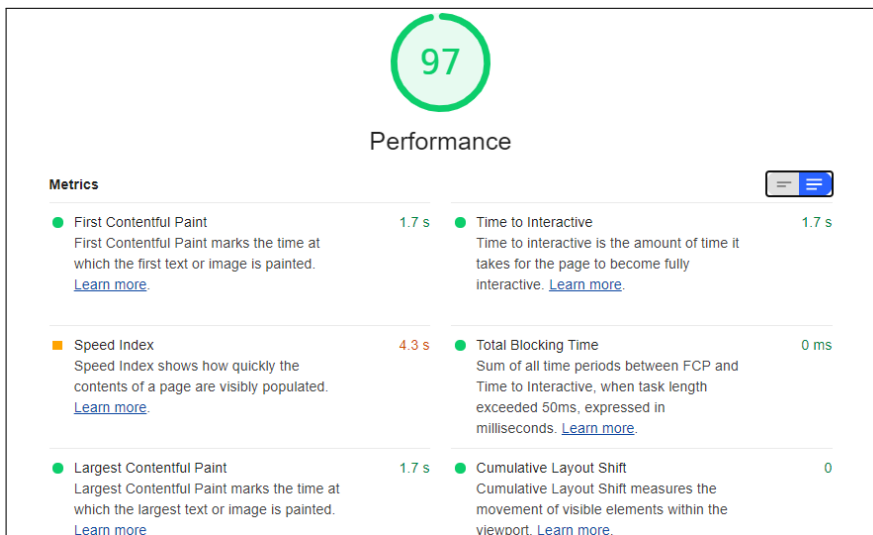
Fonte: Lighthouse

Segue os resultados individuais

Performance

O resultado obtido na Performance foi classificado como bom, porém perdeu-se pontos pela velocidade de carregamento na página index conforme pode se observar na Figura 34.

Figura 34 – Resultado detalhado de Performance.

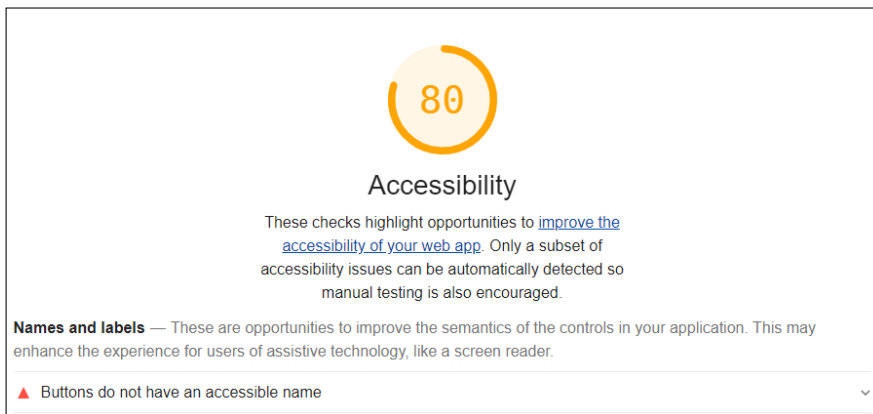


Fonte: Lighthouse

Acessibilidade

A acessibilidade foi o item com menor pontuação no teste , isso se deve a utilização do botão para criar o efeito no menu. Este botão necessita ser criado sem valores e adicionado o valor no caso em que a tela fica com a largura menor que 700px. A Figura 35 ilustra o resultado no quesito acessibilidade.

Figura 35 – Resultado detalhado de Acessibilidade.



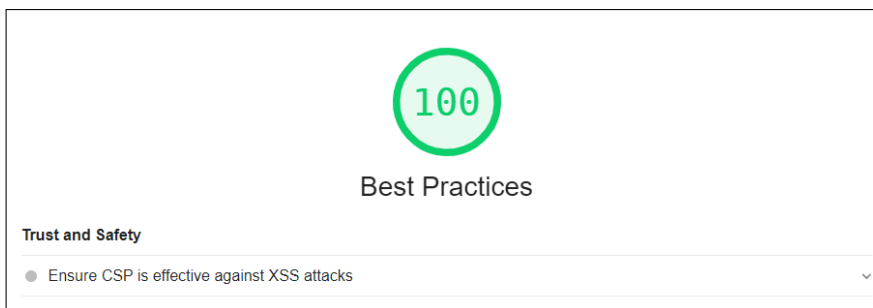
Fonte: Lighthouse

Boas práticas

Este item recebeu a pontuação máxima (Figura 32), respeitando todos os itens auditados para esse quesito dos quais podemos destacar a utilização do protocolo HTTPS, charset e doctype nas paginas HTML. A Figura 36 exibe

este resultado.

Figura 36 – Resultado detalhado Boas Práticas.

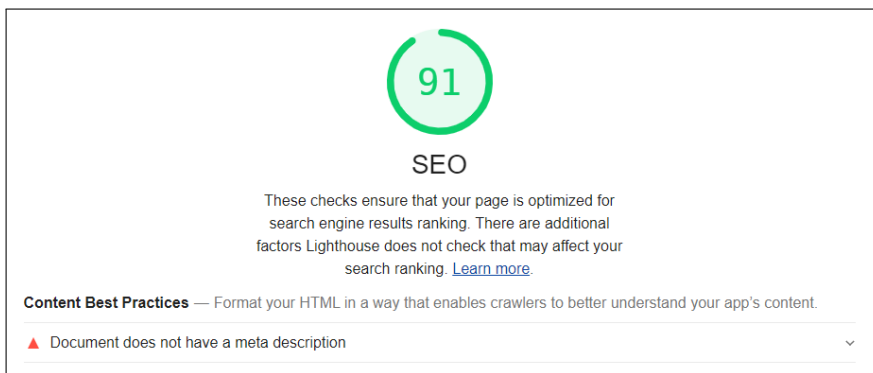


Fonte: Lighthouse

Otimização nos mecanismos de buscas

Nesse item o site também recebeu uma boa pontuação principalmente por atender alguns requisitos tais como: utilização da tag meta com os atributos de largura e escala definidos, fontes legíveis e linguagem conforme Figura 37.

Figura 37 – Resultado detalhado de Otimização nos mecanismos de buscas.

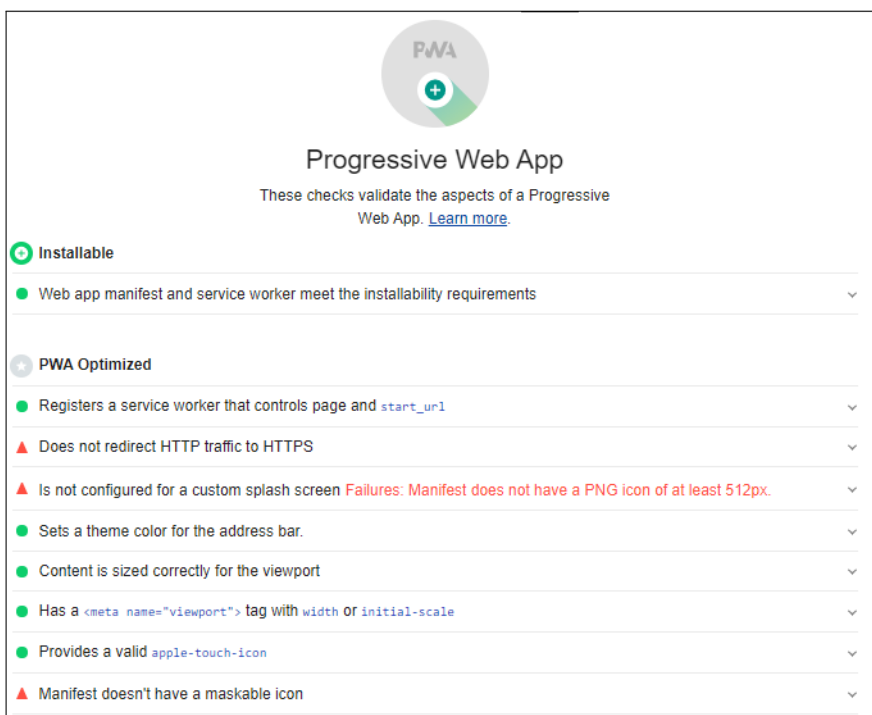


Fonte: Lighthouse

Aspectos de PWA

Este quesito não possui pontuação porém é possível verificar os pontos positivos do projeto e os que podem ser melhorados. O uso do Service Worker e Manifest garantem que o site seja instalável sendo esta uma característica forte do PWA. Por outro lado a falta de redirecionamento entre os protocolos HTTP para HTTPS é um dos pontos a ser melhorado de acordo com o relatório. Na Figura 38 pode-se verificar os pontos a serem melhorados e os que atendem aos requisitos para metodologia PWA.

Figura 38 – Resultado sobre Aspectos PWA.

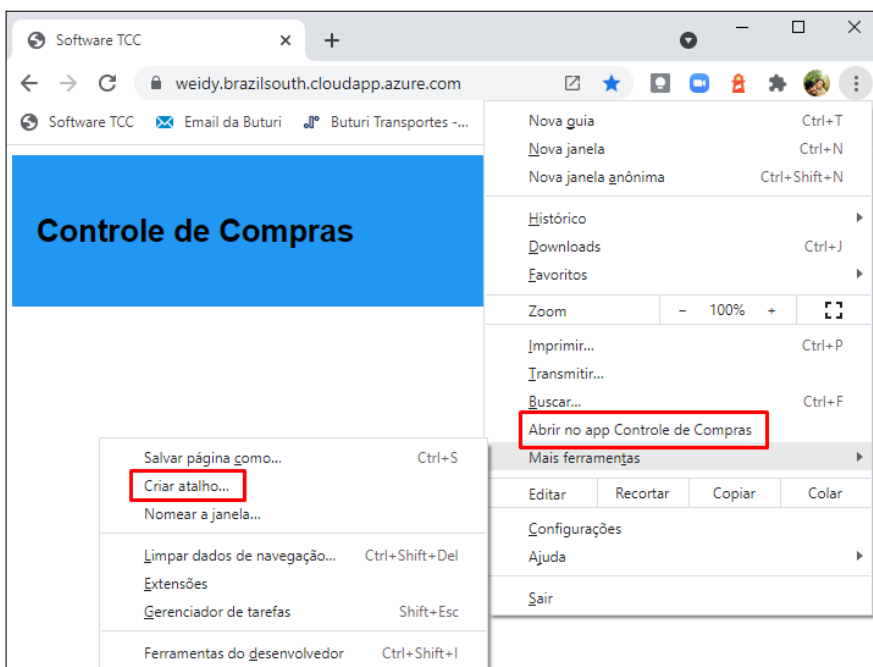


Fonte: Lighthouse

3.16 Configuração do site como aplicativo web no desktop

Por ter sido desenvolvido com a metodologia PWA, o site pode ser acessado no formato de aplicativo. No navegador Chrome é possível fazê-lo de duas maneiras, criando um atalho na Área de Trabalho do computador ou pela opção “Abrir no app Controle de Compras” conforme é mostrado na Figura 39.

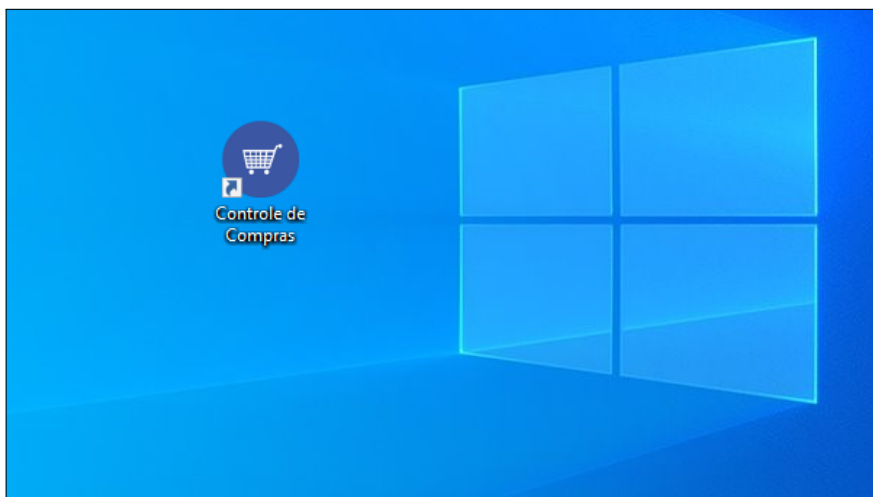
Figura 39 – Opção de criação de atalho no navegador Chrome



Fonte: Autoria própria (2021)

Utilizou-se a opção “Criar atalho”, e o resultado foi um ícone na Área de trabalho do desktop com o a imagem predefinida do arquivo manifest.json como pode-se observar na Figura 40.

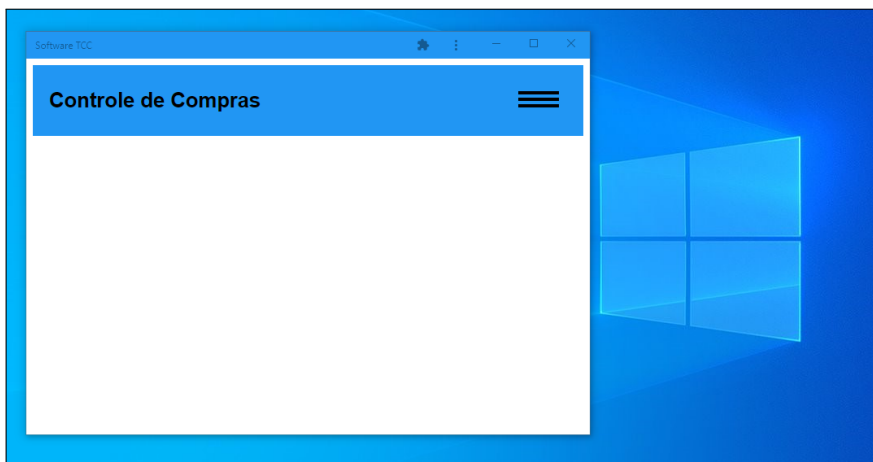
Figura 40 – Atalho criado para acesso ao aplicativo no desktop.



Fonte: Autoria própria (2021)

As configurações inseridas no arquivo manifest.json também formataram a abertura do site no formato de aplicativo. A Figura 41 mostra o layout do aplicativo web para este site.

Figura 41 – Layout do aplicativo web para site

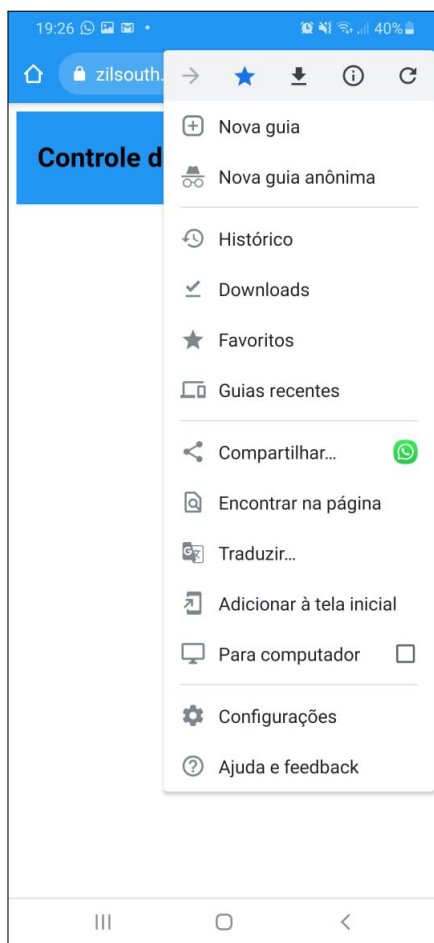


Fonte: Autoria própria (2021)

3.17 Configuração do site como aplicativo web no smartphone

A configuração do aplicativo para smartphone foi realizado de maneira semelhante ao que foi realizado para desktop. Utilizou-se a opção “Adicionar à tela inicial” disponível no navegador Chrome como mostra a Figura 42.

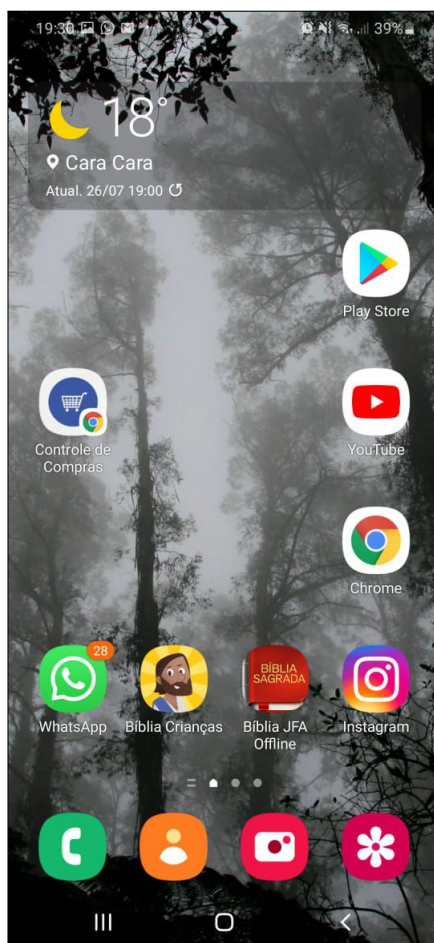
Figura 42 – Criação de atalho para aplicativo web pelo navegador Chrome no smartphone



Fonte: Autoria própria (2021)

Nesta opção cria-se um atalho na tela inicial do smartphone com o ícone do site. A Figura 43 mostra a tela de um smartphone com o ícone e o nome do site.

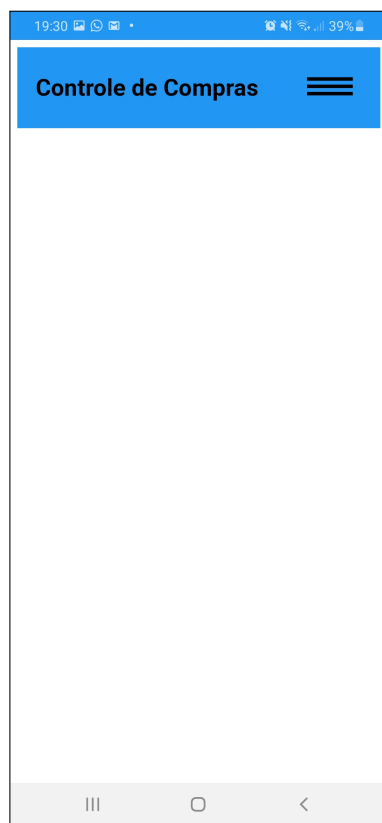
Figura 43 – Atalho criado para acesso ao aplicativo no smartphone.



Fonte: Autoria própria (2021)

Por fim a abertura do site foi configurada pelas informações predefinidas no arquivo manifest.json e o resultado pode ser observado na Figura 44.

Figura 44 – Layout do aplicativo web para o site no smartphone



Fonte: Autoria própria (2021)

4 CONCLUSÃO

Neste capítulo serão descritos as considerações finais e possíveis implementações futuras.

4.1 Considerações finais

A nota fiscal do consumidor eletrônica tem sido uma evolução no comércio brasileiro e pode ser observado facilmente no Estado do Paraná. Sua implementação tem sido bem aceita tanto por comerciantes quanto por consumidores e torna-se uma alternativa inteligente para o controle dos gastos. A utilização desse documento fiscal foi assertiva para o projeto pois trouxe confiabilidade de informações visto que com a decodificação do QR Code pôde-se obter um link único por meio do qual foi possível abstrair as informações necessárias. A abstração das informações deu-se pelo uso da técnica conhecida como raspagem de dados, essa técnica foi trabalhosa, como já era esperado, pois trata-se da obtenção de informações que não estão estruturadas para esse fim. Infelizmente, a utilização dessa forma de abstração traz o risco de com alguma alteração de layout ser necessário também a alteração no algoritmo desenvolvido para raspagem de dados. Quanto aos bancos de dados utilizados no projeto, pôde-se perceber a diferença entre a utilização de um banco antigo, relativo à programação WEB, como o MySQL e uma API nova, como é o caso da IndexedDB. O BD MySQL possui uma enorme quantidade de documentação, tutoriais, artigos e livros disponíveis que foram utilizados para uma configuração rápida e efetiva. Por outro lado a API indexedDB demandou mais tempo de estudo e testes para chegar ao fim desejado. A utilização de frameworks e bibliotecas facilitam o trabalho nessas situações por isso optou-se pela utilização da biblioteca Lovefield que economizou várias linhas de código para a programação pois tem vários procedimentos implementados que são executados em um método. O desenvolvimento do site com a metodologia PWA foi um sucesso o qual pode ser confirmada pela auditoria da ferramenta Lighthouse e com os testes que transformaram o site em aplicativo web mostrando um arquivo Manifest bem estruturado. Seu processo offline também demonstrou o bom funcionamento do Service Worker, um dos pilares da metodologia. Assim, todo o projeto alcançou os objetivos propostos proporcionando uma ferramenta leve, moderna, simples e funcional para uso.

4.2 Trabalhos futuros

Implementar doação da NFC-e para instituições filantrópicas por meio do aplicativo. Desenvolver gráficos e indicadores para um melhor entendimento do cenário financeiro.

Referências

- AEN. **Emissão de nota eletrônica será obrigatória a partir de janeiro de 2017**. 2021. Disponível em: <<https://www.aen.pr.gov.br/modules/noticias/article.php?storyid=92217&tit=Emissao-de-nota-eletronica-sera-obrigatoria-a-partir-de-janeiro-de-2017>>. Acesso em: 18 de julho 2021. Citado na página 5.
- BLACKWELL, R. D. **Consumer behavior**. Ohio: Fundação Brasileira de Contabilidade, 2001. Citado na página 1.
- BORGES THIAGO DA CUNHA; GANIMI, Z. O. **Extração de dados com web scraping para análise da variação de preço de veículos automotores**. 2018. Citado na página 12.
- BULGARIM, M. C. C. **Orçamento familiar e controle social : instrumento de organização da sociedade**. Brasília: Fundação Brasileira de Contabilidade, 2012. Citado 2 vezes nas páginas 1 e 3.
- CALVÃO, L. D. **Evolução dos Sistemas de Comunicação**. Paraty-RJ: Anais do VIII Simpósio Brasileiro de Sistemas Colaborativos, 2011. Citado na página 10.
- CELEPAR. **Menor Preço**. 2021. Disponível em: <<https://menorpreco.notaparana.pr.gov.br/index>>. Acesso em: 11 de junho de 2021. Citado na página 7.
- CODE, Q. **The LaTeX project**. 2019. Disponível em: <<https://www.qrcode.com/en/history/>>. Acesso em: 08 de julho 2019. Citado na página 6.
- CONFAZ. **Ajuste SINIEF 19/16, de dezembro de 2016**. 2021. Disponível em: <https://www.confaz.fazenda.gov.br/legislacao/ajustes/2016/AJ_019_16>. Acesso em: 12 de junho 2021. Citado na página 4.
- DUARTE, N. F. B. **Frameworks e Bibliotecas Javascript**. 2015. Citado na página 11.
- FLANAGAN, D. **Javascript o guia definitivo**. 6. ed. Porto Alegre: Editora Bookman, 2013. Citado na página 11.
- G1. **Entenda a alta dos preços dos alimentos**. 2021. Disponível em: <http://g1.globo.com/Noticias/Economia_Negocios/0,,MUL425943-9356,00-ENTENDA+A+ALTA+DOS+PRECOS+DOS+ALIMENTOS.html>. Acesso em: 14 de julho de 2021. Citado 2 vezes nas páginas 3 e 6.
- GALVES, C. **Manual de economia política atual**. 8. ed. Rio de Janeiro: Forense-Universitária, 2003. Citado na página 1.
- GAUNT, M. **Introdução aos service workers**. 2021. Disponível em: <<https://developers.google.com/web/fundamentals/primers/service-workers/?hl=pt-br>>. Acesso em: 11 de junho de 2021. Citado na página 9.
- GAVIOLI, A. **8 aplicativos e planilhas de controle financeiro para organizar gastos e despesas em 2021**. 2021. Disponível em: <<https://www.infomoney.com.br/minhas-financas/conheca-8-planilhas-e-aplicativos-para-organizar-gastos-e-controlar-despesas/>>. Acesso em: 27 de julho 2021. Citado na página 1.
- GITHUB. **Lovefield**. 2020. Disponível em: <<https://github.com/google/lovefield>>. Acesso em: 12 de julho de 2020. Citado na página 12.
- GONÇALVES, A. **O que é CSS? Guia Básico para Iniciantes**. 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css?ppc_campaign=google_performance_max&gclid=CjwKCAjwjdOIBhAEiwAHz8xm0pFf_5cqml3JNCQoh46OUig7cPI9gqBg4K-sVBgCUwq5V9fwXxVGBocFpWQAvD_BwE>. Acesso em: 20 de julho 2021. Citado na página 12.
- IBGE. **INFLAÇÃO**. 2021. Disponível em: <<https://www.ibge.gov.br/explica/inflacao.php#:~:text=Infla%C3%A7%C3%A3o%20C3%A9%20o%20nome%20dado,governo%20federal%2C%20e%20o%20INPC>>. Acesso em: 13 de julho 2021. Citado na página 2.

- JANKAVSKI, A. **Por que o arroz está tão caro? Saiba os motivos da alta do preço na cesta básica.** 2021. Disponível em: <<https://www.cnnbrasil.com.br/business/2020/09/09/por-que-o-arroz-esta-tao-carro-saiba-os-motivos-da-alta-do-preco-da-cesta-basica>>. Acesso em: 14 de julho 2021. Citado na página 3.
- KEMPSON, C. **Mudanças climáticas ameaçam produção de alimentos.** 2021. Disponível em: <https://cebds.org/mudancas-climaticas-ameacam-producao-alimentos/#.YE_s6p1KjIU>. Acesso em: 13 de julho 2021. Citado na página 3.
- LATEX. **Aplicativo de Lista de Compras.** 2021. Disponível em: <<https://www.mobills.com.br/blog/aplicativo-de-lista-de-compras/>>. Acesso em: 10 de julho 2021. Citado na página 7.
- MICHAELIS. **The LaTeX project.** 2021. Disponível em: <<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/receita/>>. Acesso em: 10 de julho de 2021. Citado na página 1.
- MICROSOFT. **Começar a trabalhar com Aplicativos Web Progressivos (Chromium).** 2021. Disponível em: <<https://docs.microsoft.com/pt-br/microsoft-edge/progressive-web-apps-chromium/get-started>>. Acesso em: 11 de junho de 2021. Citado na página 8.
- MILETTO E. M.; BERTAGNOLLI, S. C. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP.** São Paulo: Editora: Bookman, 2014. Citado na página 11.
- MOZILLA. **IndexedDB.** 2017. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/IndexedDB_API>. Acesso em: 12 de maio de 2020. Citado na página 12.
- MOZILLA. **Progressive web apps.** 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/Progressive_web_apps>. Acesso em: 16 de junho 2021. Citado 2 vezes nas páginas 8 e 10.
- PACIEVICH, Y. **Cascading Style Sheets (CSS).** 2021. Disponível em: <<https://www.infoescola.com/informatica/cascading-style-sheets-css/>>. Acesso em: 20 de julho de 2021. Citado na página 12.
- RIBEIRO, G. F. **Brasil já é campeão mundial em uma coisa: uso de aplicativo no celular.** 2021. Disponível em: <<https://www.uol.com.br/tilt/noticias/redacao/2018/06/15/brasil-ja-e-campeao-mundial-em-uma-coisa-uso-de-aplicativo-em-celular.htm>>. Acesso em: 21 de maio 2021. Citado na página 6.
- SCHMICH, C. **Instascan.** 2020. Disponível em: <<https://github.com/schmich/instascan>>. Acesso em: 12 de julho de 2020. Citado na página 11.
- SEFAZ. **Norma de Procedimento Fiscal N. 100/2014.** 2021. Disponível em: <http://www.sped.fazenda.pr.gov.br/arquivos/File/NFCE/NPF_100_2014_NFCe_Consolidada_com_NPF036_2015.pdf>. Acesso em: 08 de junho de 2021. Citado na página 5.
- SILVA, A. C. B. **Aplicativos de Gestão Financeira: um estudo exploratório.** 2020. Citado na página 6.
- SILVA, A. L. G. **USO DO PROCESSO DE EXTRAÇÃO, TRANSFORMAÇÃO E CARGA (ETL) VOLTADO A VISÃO DE ATRIBUTOS QUALIFICANTES, ASSOCIADOS AO AUXÍLIO À QUALIFICAÇÃO DO PROFISSIONAL DE TI.** 2019. Citado na página 12.
- SPC BRASIL. **EDUCAÇÃO FINANCEIRA E A GESTÃO DO ORÇAMENTO PESSOAL.** 2018. Disponível em: <https://www.spcbrasil.org.br/wpimprensa/wp-content/uploads/2018/01/Analise_Educacao_Financeira_2018.pdf>. Acesso em: 12 de junho 2021. Citado na página 6.
- SPED-PR. **SPED-PR. Nota Fiscal de Consumidor Eletrônica.** 2021. Disponível em: <<http://www.sped.fazenda.pr.gov.br/modules/conteudo/conteudo.php?conteudo=94>>. Acesso em: 08 de maio de 2021. Citado 2 vezes nas páginas 4 e 5.
- TRINDADE, P. E. **O PROGRESSIVE WEB APPS – PWA – COMO FERRAMENTA PARA A PRODUÇÃO AUDIOVISUAL.** São Paulo: Universidade Estadual Paulista, 2020. Citado na página 8.
- TRINDADE P. E.; AFFINI, L. P. **Apontamento a cerca do progressive Web Apps.** São Paulo: UFSCar, 2018. Citado na página 7.

W3C. **W3C, Brasil**. 2021. Disponível em: <<https://www.w3.org/community/webappstore/wiki/Manifest>>. Acesso em: 11 de junho de 2021. Citado na página 9.

WIKIPEDIA. **Wikipedia, HTML5**. 2019. Disponível em: <<https://pt.wikipedia.org/wiki/HTML5>>. Acesso em: 23 de novembro de 2019. Citado na página 11.

WIKIPEDIA. **Wikipedia, Progressive Web**. 2021. Disponível em: <https://pt.wikipedia.org/wiki/Progressive_web_app>. Acesso em: 18 de junho de 2021. Citado na página 8.