

UNIVERSIDADE FEDERAL DE SANTA CATARINA

MARCELO HENRIQUE HOFFMANN ECKER

DEMOCRACIA PARTICIPATIVA E DADOS ABERTOS: um MVP para o cidadão
acompanhar a compra da merenda escolar

Florianópolis - SC
2020

MARCELO HENRIQUE HOFFMANN ECKER

DEMOCRACIA PARTICIPATIVA E DADOS ABERTOS: um MVP para o cidadão
acompanhar a compra da merenda escolar

Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação, da UNIVERSIDADE FEDERAL DE SANTA CATARINA, como requisito parcial para a Obtenção do grau de Bacharel em Sistemas de Informação

Florianópolis - SC
2020

MARCELO HENRIQUE HOFFMANN ECKER

DEMOCRACIA PARTICIPATIVA E DADOS ABERTOS: um MVP para o cidadão
acompanhar a compra da merenda escolar

Trabalho de Conclusão de Curso apresentado ao
curso de Sistemas de Informação, da
UNIVERSIDADE FEDERAL DE SANTA
CATARINA, como requisito parcial para a
Obtenção do grau de Bacharel em Sistemas de
Informação

Florianópolis - Santa Catarina, 30 de junho de 2020

BANCA EXAMINADORA

Prof. José Eduardo de Lucca
Orientador

Prof. José Francisco Danilo de Guadalupe Correa Fletes
Membro da banca

Amilton Justino
Membro da banca

Dedico este trabalho a meu avô Ernani, que está nos
olhando lá de cima de algum lugar do céu.

E a todos que não perderam a esperança.

AGRADECIMENTOS

Cursar uma universidade federal não é simplesmente fazer uma faculdade, principalmente para quem veio de uma cidade do interior. No momento que começamos o curso não estamos só frequentando os bancos da sala de aula de um ensino superior público de qualidade, estrato educacional infelizmente muito restrito em nosso país, mas estamos encarando um desafio que vem para transformar nossas vidas de forma muito profunda. Estamos deixando o conforto da casa de nossos pais para morar sozinhos em lugares às vezes inóspitos, enfrentando tanto os desafios do dia a dia como lavar a roupa de forma correta, economizar para comprar um micro-ondas, quanto aquelas aulas de Análise e Projetos de Sistemas que te desafiam a desenvolver diagramas de sequência, a quantidade de boletos que insistem em não parar de chegar e resolver aquele exercício que não quer compilar nunca. A vida de quem vive a UFSC não é nada simples.

Este trabalho não é só um trabalho de conclusão de curso comum. É a finalização de um ciclo. De um ciclo de desenvolvimento pessoal. Como gosto de falar a meus amigos: “chegamos na universidade meninos, e aqui estamos virando homens”. Seja lá o que isso signifique. A verdade é que somos filhos da UFSC. E nunca vamos nos esquecer desse lugar. Nem que vivêssemos 1000 anos, os corredores da universidade, o espaço de convivência, o Restaurante Universitário, as praças e os bares ao redor ainda ficarão gravados em nossas memórias com carinho. Como o local em que aprendemos muito mais do que lições técnicas de nossas futuras profissões. Ficarão marcadas como o lugar que aprendemos mais sobre a própria vida e sobre o sentido de viver. Por isso, o meu primeiro agradecimento vai a sociedade que paga os impostos que propiciam que ambientes como a Universidade Federal de Santa Catarina existam. Fazendo ciência, produzindo conhecimento e produzindo seres humanos muito melhores para o mercado de trabalho e para o desenvolvimento da sociedade.

Gostaria de estender esse agradecimento à Universidade Federal de Santa Catarina no geral. Nela tive a oportunidade de conseguir os primeiros estágios e a partir deles, tive a chance de iniciar oficialmente a minha carreira profissional com o meu primeiro emprego de carteira assinada. Hoje já estou em outros desafios profissionais, que também foram possíveis graças aos conhecimentos que acumulei durante todo esse período, sejam eles técnicos ou de relacionamentos interpessoais.

Gostaria de agradecer a todos os professores pelas aulas ministradas, em especial ao meu orientador José Eduardo de Lucca, que esteve ao meu lado durante esse longo processo de gestação do trabalho final e foi figura importantíssima nessa

reta final de desenvolvimento com reuniões de videoconferência que extrapolavam o horário comercial, em meio a esse período de pandemia muito séria que estamos vivendo. Agradeço também a banca pela avaliação do meu trabalho.

Aos meus colegas que entraram comigo na universidade, e que hoje viraram mais do que colegas, viraram amigos queridos; deixo a minha mais profunda gratidão pelos momentos vividos, na certeza de que muitos mais virão. Gostaria de mencionar alguns nomes. Para começar, gostaria de agradecer ao meu amigo Vitor Schweitzer por todas as conversas que tivemos durante todos esses anos. É espantosa a quantidade de sabedoria que você pode carregar. Aprendi muito contigo e tenho orgulho de ser seu amigo. Ao Francisco Sacco, que me ajudou em alguns momentos chaves da minha vida, foi companheiro de copo em outros tantos, sempre haverá uma cerveja gelada para você em minha geladeira. Ou uma coca zero, se você estiver sem beber. Me faltam palavras bonitas para te dizer, afinal, nessa hora, ao contrário de seu blog, não programamos pensamentos. Ao meu amigo Péricles da Costa, gostaria que todas as pessoas do mundo tivessem o mesmo senso de justiça que você. Ele me inspira e me faz acreditar que viveremos dias melhores. Ao meu amigo Raphael da Silveira, parceiro de tantos momentos de alegria e de festas durante o período do curso, fica o meu mais fraternal agradecimento e desejo que a vida seja um eterno hot dog depois do open-bar.

Aos meus colegas de empresa, destacando assim o período que trabalhei na InCuca e na empresa que estou trabalhando hoje, a BRy Tecnologia. Vocês foram fundamentais durante todo esse processo de formação profissional e foi um prazer trabalhar ao lado de pessoas tão qualificadas quanto vocês. Gostaria de destacar meus amigos Vitor Cardoso, Nicolas Busatto e Leonardo Nones por todas as conversas durante o período de realização deste trabalho. Vocês me inspiram a me tornar uma pessoa melhor e me deixam com a certeza de que o mundo seria um lugar muito melhor se houvesse mais pessoas como vocês.

A todas os outros que convivi durante o período desse curso fiquem com a consciência de que cada um foi importante e me deixou uma lição. Se hoje eu sou a pessoa que sou é porque conheci pessoas maravilhosas como vocês durante todo esse período. Cabe destacar os meus amigos Cauê, Thiago, Samuel e Sid. Vocês são os melhores amigos mais antigos que qualquer um poderia ter.

Por fim, não poderia deixar de agradecer aos meus pais e minha família por todo o apoio durante esse período do curso. Gostaria de agradecer em especial duas mulheres: minha mãe e minha vó, que foram essenciais para que eu tivesse a estabilidade de dar os primeiros passos de minha vida adulta, sabendo que sempre teria onde correr caso a estrada do caminho me fizesse tropeçar. Vocês são uma inspiração e merecem ser louvadas nesse momento. Junto a vocês, queria destacar

a presença eterna de meu avô, que hoje descansa no céu, mas está eternamente presente em nossa memória e também foi parte fundamental desse processo, já que provavelmente ninguém se preocupou tanto comigo quanto ele. À minha irmã, que está crescendo como uma pessoa maravilhosa que muito me dá orgulho. E à minha tia e meu primo Erick Roscete que são presenças constantes e que deixam a família ainda melhor. Ao meu pai e meus irmãos Artur Ecker e Vitor Ecker fica o carinho e a lembrança. Meu amor eterno a todos vocês.

Mesmo representando o final de um ciclo, esse trabalho também representa a certeza de que muitos outros virão. Uma frase que me marcou muito durante a infância foi uma de Miguel de Cervantes em sua obra-prima Dom Quixote: “Os grandes feitos estão reservados aos grandes homens”. Deus me permita que, um dia, eu seja um deles.

“Porque a revolução é uma pátria e uma família”
(Jorge Amado)

RESUMO

Atualmente, os meios de comunicação e a tecnologia são praticamente onipresentes em nossa vida. Desde a origem dos computadores, notebooks, dispositivos portáteis e assistentes pessoais, a disseminação da informação chega ao seu ápice na figura do smartphone. De outro lado possuímos órgãos governamentais que estão iniciando a disponibilização de informações de interesse ao cidadão em seus portais da transparência. Em meio a este contexto, o cidadão tem na democracia participativa a oportunidade de contribuir de forma ativa no governo, fiscalizando as ações e propondo melhorias para a sociedade. Entretanto, nem sempre as informações de interesse público estão disponibilizadas de forma compreensível ao cidadão ou mesmo para a execução de algoritmos, portanto necessita-se de ferramentas que o auxiliem na busca e compreensão dos dados, bem como facilitem a sua atuação diante dos resultados encontrados. O presente trabalho desenvolve uma extensa revisão bibliográfica sobre o assunto, e a partir desta constrói um ecossistema tecnológico para atender estas necessidades. Como estudo de caso optou-se pelo monitoramento de compras governamentais de merenda escolar. A aplicação desenvolvida implementa uma API REST para consultas das compras realizadas. Essa informação é disponibilizada através de um aplicativo mobile onde o cidadão pode avaliar indícios de fraude. Para auxiliar os cidadãos nessa tarefa também são informadas as compras dos municípios vizinhos e uma estimativa gerada através de um processo de machine learning com os dados cadastrados. Também foi desenvolvido a integração das avaliações dos usuários com a rede social Twitter, trazendo as compras ainda mais próximas das pessoas que não queiram utilizar o aplicativo.

Palavras-chave: Sociedade da Informação. Portal da Transparência. Dados abertos. Machine learning. Desenvolvimento web

ABSTRACT

Today, the media and technology are practically ubiquitous in our lives. Since the origin of computers, notebooks, portable devices and personal assistants, the dissemination of information reaches its peak in the figure of the smartphone. On the other hand, we have government agencies that are starting to make information of interest to citizens available on their transparency portals. In the midst of this context, the citizen has in the participative democracy the opportunity to contribute actively in the government, inspecting the actions and proposing improvements for the society. However, information of public interest is not always available in a comprehensible way to the citizen or even for the execution of algorithms, so tools are needed to assist in the search and understanding of the data, as well as to facilitate their performance as citizens before results found. The present work develops an extensive bibliographic review on the subject, and from this it builds a technological ecosystem to meet these needs. As a case study, we chose to monitor government purchases of school meals. The developed application implements a REST API to consult the purchases made. This information is made available through a mobile application where the citizen can assess signs of fraud. To assist citizens in this task, purchases from neighboring municipalities are also informed and an estimate generated through a machine learning process with the registered data. The integration of user reviews with the social network Twitter was also developed, bringing purchases even closer to people who do not want to use the application.

Keywords: Information Society. Open data. Machine Learning. Web Development

LISTA DE ILUSTRAÇÕES

Figura 1 — Ranking sobre a percepção da corrupção no mundo (na escala IPC)	24
Figura 2 — Características do Big Data - 3V's	35
Figura 3 — Características do Big Data - 5V's	36
Figura 4 — Como funciona o MapReduce	41
Figura 5 — Como funciona o Hadoop	42
Figura 6 — Resumo do ecossistema do MVP	52
Figura 7 — Modelo Conceitual - Avaliação	58
Figura 8 — Modelo Conceitual - Preço	60
Figura 9 — Modelo Lógico - Avaliação	62
Figura 10 — Modelo Lógico - Arquivo	63
Figura 11 — Camadas da aplicação backend	64
Figura 12 — Diretórios da API REST	67
Figura 13 — Classe Compra	68
Figura 14 — Declaração da classe de compra	70
Figura 15 — Método "index"	71
Figura 16 — Método "indexByMunicipio"	72
Figura 17 — Método "random"	72
Figura 18 — Método "indexMunicipiosProximos"	73
Figura 19 — Roteamento REST - Declaração	75
Figura 20 — Roteamento - Compra	76
Figura 21 — Roteamento - Exportação	76
Figura 22 — Configurações prévias de roteamento	77
Figura 23 — Uso do Insomnia	77
Figura 24 — Arquitetura do frontend	82
Figura 25 — Visão dos diretórios do frontend	84
Figura 26 — Visão detalhada da pasta "pages"	85
Figura 27 — Camada de domínio "Compra"	86
Figura 28 — Definição da classe de serviço	87
Figura 29 — Cabeçalho da página "Avaliar compra"	88
Figura 30 — Definição da estrutura de exibição da compra	89
Figura 31 — Definição da estrutura de exibição da estimativa	90
Figura 32 — Definição da exibição de compras nos municípios vizinhos	90
Figura 33 — Definição da exibição de comentários	91
Figura 34 — Definição da classe de controle	92
Figura 35 — Inicialização dos dados da classe	93
	95

Figura 36 — Definição da visualização	95
Figura 37 — Método de carregamento dos dados	97
Figura 38 — Filtragem dos outliers	98
Figura 39 — Criação do modelo	99
Figura 40 — Convertendo dados para tensores	100
Figura 41 — Treinamento do modelo	101
Figura 42 — Teste do modelo	102
Figura 43 — Execução do modelo	103
Figura 44 — Criação do objeto Twit	105
Figura 45 — Criação do objeto para postagem	105
Figura 46 — Condicional na avaliação	106
Figura 47 — Resultado de integração com o Twitter	106

LISTA DE TABELAS

Tabela 1 — Lista dos endpoints desenvolvidos	78
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface (Interface de Programação de Aplicações)
CORS	Cross-origin Resource Sharing
CRUD	Create - Read - Update - Delete (criar, ler, atualizar, deletar)
CSS	Cascading Style Sheets
FNDE	Fundo Nacional de Desenvolvimento da Educação
HTML	Linguagem de Marcação de HiperTexto
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
JSON	JavaScript Object Notation
LAI	Lei de Acesso à Informação
MVC	Model-View-Controller
MVP	Minimum Viable Product (Produto mínimo viável)
OGP	Open Government Partnership
ORM	Object-relational mapping (Mapeamento objeto-relacional)
PNAE	Programa Nacional de Alimentação Escolar
REST	Representational State Transfer
SASS	Syntactically Awesome Stylesheets
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
TCC	Trabalho de Conclusão de Curso
UFSC	Universidade Federal de Santa Catarina
WWW	World Wide Web

SUMÁRIO

1	INTRODUÇÃO	17
1.1	OBJETIVO GERAL	19
1.2	OBJETIVOS ESPECÍFICOS	19
1.3	METODOLOGIA DO TRABALHO	19
2	FUNDAMENTAÇÃO TÉORICA	21
2.1	SOCIEDADE DA INFORMAÇÃO	21
2.2	CORRUPÇÃO	22
2.3	TRANSPARÊNCIA COMO ATENUANTE DA CORRUPÇÃO	24
2.4	LEI DE ACESSO À INFORMAÇÃO	25
2.5	DADOS ABERTOS	26
2.6	GOVERNO ABERTO	28
2.7	DADOS ABERTOS GOVERNAMENTAIS	29
2.8	CIBERATIVISMO E HACKTIVISMO	30
2.9	DEMOCRACIA PARTICIPATIVA	31
2.10	PORTAL DA TRANSPARÊNCIA	32
2.11	BIG DATA	34
2.11.1	Volume	36
2.11.2	Variedade	37
2.11.3	Velocidade	37
2.11.4	Variabilidade	37
2.11.5	Veracidade	38
2.12	DATA ANALYTICS	38
2.13	PROGRAMAÇÃO UTILIZANDO MAPREDUCE	40
2.14	HADOOP	41
2.15	TENSORFLOW	43
2.16	TENSORFLOW.JS	43
2.17	MODELOS E CAMADAS	44
2.18	FERRAMENTAS AUXILIARES	44
2.18.1	Ionic Framework	45
2.18.2	MySQL	45
2.18.3	Express	46
2.18.4	API de Localidades	46
3	TRABALHOS CORRELATOS	47
4	PROPOSTA DO TRABALHO	51
5	DESENVOLVIMENTO DO TRABALHO	54
5.1	BACK-END DA APLICAÇÃO	54
		55

5.1.1	Ferramentas utilizadas	55
5.1.2	Tecnologias utilizadas	56
5.1.3	Implementação	57
5.1.4	Modelagem de banco de dados	57
5.1.4.1	Construção do modelo lógico	61
5.1.5	API REST Merenda Democrática	63
5.1.5.1	Visão Geral	66
5.1.5.2	Camada de domínio	67
5.1.5.3	Camada de controle e camada de acesso a dados	69
5.1.5.4	Camada de roteamento REST	74
5.1.6	Endpoints	77
5.1.7	Disponibilização do código-fonte do backend	79
5.2	FRONTEND DA APLICAÇÃO	79
5.2.1	Ferramentas utilizadas	79
5.2.2	Tecnologias utilizadas	80
5.2.3	Implementação	81
5.2.3.1	Visão do diretório da aplicação	83
5.2.3.2	Camada de domínio (modelos e serviços)	86
5.2.3.3	Camada de visualização	87
5.2.3.4	Camada de controle	91
5.2.4	Disponibilização do código-fonte do frontend	93
5.3	INDICADORES DE PREÇO DE COMPRA	94
5.3.1	Implementação do algoritmo	94
5.3.1.1	Definição da camada de visualização	95
5.3.1.2	Carregamento e tratamento dos dados	96
5.3.1.3	Arquitetura do modelo	98
5.3.1.4	Preparando os dados para treinamento	99
5.3.1.5	Treinamento do modelo	100
5.3.1.6	Realizando as previsões	101
5.3.1.7	Iniciando a execução	102
5.3.2	Disponibilização do código-fonte	103
5.4	COMUNICAÇÃO COM REDES SOCIAIS	103
6	CONCLUSÃO	107
6.1	TRABALHOS FUTUROS	108
	REFERÊNCIAS	110
	APÊNDICE A — TELAS DO APLICATIVO FRONTEND	116
	APÊNDICE B — OBTENÇÃO DE COMPRAS DO PAINEL DE PREÇOS	
	DO FNDE	124
		129

APÊNDICE C — EXECUÇÃO DA APLICAÇÃO UTILIZANDO TENSORFLOW.JS	129
--	------------

1 INTRODUÇÃO

Nos dias de hoje, os meios de comunicação e a tecnologia são praticamente onipresentes em nossa vida. Desde a origem dos computadores, notebooks, dispositivos portáteis e assistentes pessoais, a disseminação da informação chega ao seu ápice na figura do smartphone. Nesse cenário estamos conectados com um manancial de tamanho quase infinito de informações a poucos cliques da ponta de nossos dedos.

Com a quantidade de elementos tecnológicos disponíveis, possuímos uma série de oportunidades para resolver os nossos problemas cotidianos, sejam eles individuais e de cunho prático ou sociais e de cunho filosófico. (TEIXEIRA, 2006) A disseminação irreversível e progressiva, dia a dia, da tecnologia da informação e a utilização incondicional dos sistemas de informação automatizados em todos os níveis da administração pública direta ou indiretamente atua como motivador para que problemas que envolvam a máquina pública tenham suas soluções estabelecidas por meios tecnológicos (MARTINS, 2014).

É notório que um dos maiores problemas que o país possui atualmente é a má administração e gestão da máquina pública. Caminha (2014) ressalta que o estado possui um orçamento bastante amplo e acaba, até mesmo por falta de ferramentas concisas que possam estabelecer um suporte ou devido a má-fé de entes públicos envolvidos, gastando um valor muito mais alto do que os preços de mercado em pregões públicos, conhecido também como licitações. Fatos dessa natureza ocorrem em praticamente todos os segmentos que vão desde obras de infra-estrutura (pontes, estradas, postos de saúde, etc), material de custeio, remédios e a compra dos itens da merenda escolar. A má-gestão da máquina pública abre espaço para que outro mal se alastre, a corrupção. Tendo uma estrutura que não é eficiente e transparente, gestores mal-intencionados possuem a oportunidade de se beneficiar de forma pessoal da sua atuação profissional. Pela legislação existe corrupção quando há o desvio por parte de um funcionário público dos deveres formais do cargo devido à busca de uma recompensa para si ou para os outros; a definição de mercado considera corrupção a utilização do cargo público pelo seu ocupante como uma forma de maximizar a sua renda pessoal. Definições à parte, fato é que a corrupção é um dos principais motivos que colaboram para que o estado se torne ineficiente e que o país não consiga se tornar um país desenvolvido, com distribuição de renda e qualidade de vida para todos os seus habitantes (CAMINHA, 2004).

Entretanto já vigoram medidas que visam tornar a administração pública mais moderna e transparente. De acordo com a Constituição Federal (1988) em seu

art.5º, XXXIII, todos os cidadãos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidades ressalvadas informações que são confidenciais e remetem à questões que são imprescindíveis à segurança da sociedade e do Estado. A principal ferramenta que atua na divulgação das informações de gestão pública é o Portal da Transparência (RIBEIRO, 2016)

O Portal da Transparência é uma iniciativa da Controladoria Geral da União (CGU), lançado em novembro de 2004, para assegurar a boa e correta aplicação dos recursos públicos. O objetivo do desenvolvimento do portal é aumentar a transparência da gestão pública, permitindo que cidadãos acompanhem e ajudem a fiscalizar como o dinheiro público está sendo utilizado (MARTINS, 2004).

A educação básica é uma das maiores responsabilidades que o estado possui. Está previsto na constituição em seu artigo 6º que a educação é um direito garantido de todos os brasileiros e pode e deve ser exigida de seus órgãos competentes quando esse direito for violado ou desrespeitado (CONSTITUIÇÃO, 1989). Um dos aspectos básicos que garantem o pleno desenvolvimento das atividades de ensino-aprendizagem é a oferta de merenda escolar para todos os estudantes matriculados na rede pública. De acordo com Silva (2010), estudantes que se alimentam mostram melhor disposição, melhora no humor, no crescimento físico e no desenvolvimento sócio-psicomotor. O fornecimento de merenda escolar é responsabilidade do estado e não é raro encontrar denúncias de fraudes e irregularidades, noticiadas pelos mais diversos setores da imprensa (G1, 2019). Para tornar tudo ainda mais perverso, muitas crianças têm no momento da merenda escolar a única refeição do seu dia (GAÚCHAZH, 2014).

Pretende-se com esse trabalho realizar um estudo sobre medidas tecnológicas que auxiliam no combate a corrupção e na otimização e gestão das finanças públicas. Tal estudo será utilizado como base para desenvolvimento de uma aplicação que permite ao cidadão através da democracia participativa, analisar os preços dos itens de compra da merenda escolar obtidos através de bases de dados públicas, possibilitando assim a criação de indicadores de fraude. A aplicação também fornecerá alguns dados que possibilitam melhor avaliação do valor da merenda por parte do cidadão: compras similares realizadas por municípios vizinhos e valor previsto através de um processo de aprendizado de máquina (*machine learning*).

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver uma ferramenta para que os cidadãos tenham acesso aos dados de compras da merenda escolar disponibilizadas pelo Fundo Nacional de Desenvolvimento da Educação (FNDE), no âmbito do Plano Nacional de Alimentação Escolar (PNAE), permitindo que os mesmos possam contribuir fiscalizando a compra da merenda escolar, buscando assim que o processo de compra possa se tornar mais barato, transparente e eficiente.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar uma revisão bibliográfica sobre Sociedade da Informação, corrupção e medidas de combate, como a transparência.
- Realizar uma revisão bibliográfica sobre o Portal da Transparência, Dados Abertos e conceitos relacionados à Lei de Acesso à Informação.
- Realizar uma revisão bibliográfica sobre ciberativismo e hacktivismo.
- Realizar uma revisão bibliográfica e apresentar possibilidades de tecnologias que permitam o desenvolvimento de aplicações para analisar o preço unitário de produtos presentes no Portal da Transparência de instituições públicas.
- Propor e desenvolver uma API REST que permite que o cidadão acessar de forma simplificada as compras da merenda escolar realizadas pelos municípios.
- Propor e desenvolver uma solução mobile aberta para obter avaliações dos cidadãos sobre as compras efetuadas pelas entidades públicas.
- Propor e desenvolver um modelo que prevê o preço do valor unitário esperado para determinadas compras de produtos na merenda escolar.
- Propor e desenvolver formas de disseminar essas informações, como integração com redes sociais para que o cidadão que não deseja instalar o aplicativo tenha acesso aos indícios de fraude.

1.3 METODOLOGIA DO TRABALHO

Neste trabalho será realizada uma revisão bibliográfica sobre conceitos como dados abertos e transparência da informação, sociedade da informação, corrupção,

Big Data e análise de dados, bem como será retratado as características, funcionalidades, vantagens e desvantagens que esses modelos possuem. Em seguida serão expostos trabalhos correlatos, que possuem objetivos semelhantes a este e abordam temas como combate à corrupção, transparência de dados, análise das contas públicas e democracia participativa.

O trabalho será composto também por uma parte prática, com o desenvolvimento de uma aplicação para permitir a avaliação de empenhos de compras da merenda escolar oriundas de Portais da Transparência, bem como o desenvolvimento de *insights* que possam auxiliar o cidadão no ato de avaliar, como preço previsto da compra comparando com as outras compras que estão disponíveis na plataforma e preço do mesmo produto praticado nos municípios vizinhos. A aplicação também irá permitir que o usuário alimente a base de dados enviando imagens e o preço dos produtos, de forma a construir uma base de preços nacional que auxilie em futuras análises, reforçando também o intuito de democracia participativa estabelecido como chave no trabalho.

2 FUNDAMENTAÇÃO TÉORICA

Encontra-se neste capítulo conceitos relacionados com sociedade da informação, corrupção, Lei de Acesso à Informação, Portal da Transparência, Big Data, Data Analytics e programação utilizando MapReduce e Hadoop, TensorFlow e sua biblioteca para Javascript TensorFlow.JS, bem como ferramentas auxiliares no desenvolvimento deste trabalho. Será realizado uma revisão ampla que permitirá contextualizar o cenário do problema e relacionar aspectos sociais com os aspectos tecnológicos.

2.1 SOCIEDADE DA INFORMAÇÃO

Com o desenvolvimento da tecnologia e a popularização das ferramentas tecnológicas, um novo modo de desenvolvimento social e econômico surge, tendo a informação como meio de criação de conhecimento e desempenhando um papel fundamental na produção de riqueza e na contribuição para o bem-estar e qualidade de vida dos cidadãos (AHLERT, 2016).

De acordo com Castells (2000), o termo que melhor se adequa a sociedade da informação é o termo “sociedade em rede”, porque a informação e o conhecimento sempre foram e sempre serão essenciais em qualquer sociedade. Castells define que o conceito se trata de um termo que reflete *“uma estrutura social baseada em redes operadas por tecnologias de comunicação e informação fundamentadas na microeletrônica e em redes digitais de computadores que geram, processam e distribuem informação a partir do conhecimento acumulado nos nós dessa rede”*. Castells também destaca o potencial que as tecnologias possuem para alterar a dinâmica das sociedades.

Ahlert (2016) pondera que há uma condição fundamental para que aquilo que conhecemos como “Sociedade da Informação” se desenvolva, a possibilidade de todos terem acesso às Tecnologias da Informação, que são instrumentos indispensáveis às comunicações pessoais, profissionais e de lazer. Infelizmente isso ainda não se trata de uma realidade, causando injustiças sociais e desigualdade, em um processo que pode ser denominado de “desinclusão digital”.

Castells (2000) também define que algumas características são típicas da “Sociedade da Informação”, como: a informação como matéria-prima, os efeitos da alta tecnologia possuindo alta penetrabilidade (ou seja, toda atividade humana, individual ou coletiva e, portanto todas essas atividades tendem a serem afetadas diretamente pela nova tecnologia), predomínio da lógica de redes, flexibilidade e

crescente convergência de tecnologias de diversos campos a microeletrônica, telecomunicações, optoeletrônica, computadores e crescentemente, a biologia.

Bugarin (2003) ressalta que o advento do ciberespaço transformou a forma com que as pessoas recebem informação. Se antes as informações estavam restritas por meio de comunicações estatais e privadas, com destaque a televisão, hoje o cidadão possui um vasto contingente de possibilidades de comunicação e informação, notadamente com o advento da World Wide Web (WWW) que propicia com que pessoas comuns possam demonstrar livremente o seu pensamento.

Essas novas tecnologias também estão sendo adotadas pelo poder público, especialmente pelas administrações públicas, de forma a manter um controle maior sobre a gestão e aplicação das verbas, utilizando sistemas informatizados que possibilitam a confrontação de dados lançados em rede (AHLERT, 2016). Essas tecnologias poderiam ser utilizadas para encarar o controle das práticas corruptivas, através de uma política de transparências dos dados que divulgue as informações com qualidade permitindo que os cidadãos tenham acesso não só a dados que anteriormente não eram divulgados, mas que tenham acesso a esses dados de uma forma que possam ser facilmente compreendidos e possam ter real utilidade (AHLERT, 2016).

Hoje é notório que o enfrentamento à corrupção no Brasil ainda faz pouco uso das potencialidades das redes sociais nem de outros elementos relacionados à participação social (AHLERT, 2016). Há algumas medidas interessantes sendo desenvolvidas, principalmente por órgãos e grupos não-oficiais, como a Serenata do Amor, que utilizando uma base de dados de gastos do cartão corporativo dos deputados e tecnologias que traçam padrões nesses gastos, consegue indicar contas que apresentam um comportamento estranho. A participação coletiva entra nesse modelo quando o robzinho emite um alerta e as pessoas têm acesso ao empenho e podem classificar a conta como suspeita ou não (OPERAÇÃO SERENATA DO AMOR, 2018). Para compreender melhor o cenário de medidas de combate a corrupção é necessário definir o termo por si só. Este ponto será realizado na próxima seção.

2.2 CORRUPÇÃO

Pereira (2014) define que corrupção ou a prática corruptiva se trata de um desvio de comportamento de uma pessoa que exerce uma função pública, porém possui interesses pessoais ou de grupos, violando determinadas regras que orientam o tipo de comportamento esperado para agentes públicos. No Brasil,

Pereira, diz que a corrupção essencialmente se caracteriza pela troca de favores e pelo tratamento privilegiado de investidores nas decisões que envolvem fundos e políticas públicas, originada pelos crescentes custos e pelo financiamento privado de campanhas eleitorais, por grupos econômicos que visam obter vantagens e possuem interesse direto no resultado das eleições.

Lemos (2016), reforça que a corrupção perpassa a história da sociedade humana, independente da época ou região, sendo possível afirmar que é um fenômeno universal que tem se intensificado com o aumento do fluxo de pessoas, capital e informação. Ahlert (2016) relata que a corrupção no Brasil se trata de um fenômeno cultural, oriundo desde os tempos da colonização portuguesa, que gera uma ética perversa, com base em conceitos como esperteza, ganância, hipocrisia, exploração e fraude em detrimento das ações de cidadãos que são considerados “corretos e honestos”.

Pereira (2014) demonstra duas vertentes para a corrupção: a primeira é considerada de menor monta, e acontece através de uso de propina, onde um funcionário público recebe alguma vantagem em troca de um favor, no qual é utilizado a influência de seu cargo. Esse tipo de corrupção é menos danoso em termos financeiros, ou seja, envolvem quantias menores do que o segundo tipo, mas cria uma espécie de consciência pública em que as pessoas não confiam no poder e no papel das instituições. O segundo tipo de corrupção é considerado o de grande monta, em que há um maior prejuízo financeiro para a sociedade, e Pereira (2014) destaca que esse tipo de corrupção acontece principalmente em processos de licitações e contratações entre o poder público e a iniciativa privada. Nessas licitações são realizadas práticas como a formação de cartel de forma a inflar os preços e dividir as obras entre os grupos empresariais interessados, bem como superfaturamento e falta de priorização de requisitos técnicos.

Segundo um estudo divulgado pela Transparência Internacional, o Brasil ocupa o 76º lugar em ranking sobre a percepção da corrupção no mundo, em um levantamento em que foram analisados 168 países (TRANSPARENCY INTERNACIONAL, 2015). A transparência Internacional é referência mundial na análise da corrupção e o relatório é desenvolvido anualmente desde 1995, com base em diversos estudos e pesquisas sobre os níveis de percepção da corrupção no setor público de diferentes países. A figura 1 apresenta a colocação dos países da América do Sul no Ranking sobre a percepção da corrupção no mundo. A escala do IPC vai de 0 a 100, na qual 0 significa que o país é percebido como altamente corrupto e 100 significa que o país é percebido como muito íntegro.

Figura 1 - Ranking sobre a percepção da corrupção no mundo (na escala IPC)

Rank	Country/territory	2015	2014	2013	2012
21	Uruguai	74	73	73	72
23	Chile	70	73	71	72
76	Brasil	38	43	42	43
83	Colômbia	37	37	36	36
88	Peru	36	38	38	38
99	Bolívia	34	35	34	34
107	Argentina	32	34	34	35
107	Equador	32	33	35	32
130	Paraguai	27	24	24	25
158	Venezuela	17	19	20	19

Fonte: Transparency Internacional (2015)

Ahlert (2016) ressalta que a corrupção causa um ambiente de enormes prejuízos, sejam eles através da crise de confiança nas instituições, ou mesmo financeiros, como em obras em que acontece superfaturamento ou obras que nunca terminam, sendo necessários novos aditivos para garantir a conclusão do projeto. Desse modo, Ahlert (2016) frisa que é essencial a promoção de uma cultura de transparência de controle social dos gastos públicos, para que se possa garantir um ambiente de pleno exercício da cidadania através de condutas éticas e sociais, transformando assim os próprios cidadãos em autores da construção de um país melhor e mais justo para todos.

2.3 TRANSPARÊNCIA COMO ATENUANTE DA CORRUPÇÃO

Ahlert (2016) defende que o acompanhamento eletrônico das ações da Administração Pública, pode ser uma maneira que a sociedade dispõe para fiscalizar as ações do Estado. Nesse cenário, conceitos como o combate à corrupção, controle social e transparência estão estritamente ligados, já que com o acesso às informações disponíveis de forma clara, há a possibilidade de o cidadão fiscalizar os atos da Administração Pública e estar municiado na hora de exercer o seu direito de participação na gestão dos recursos públicos.

Pereira (2014) destaca que ainda há grandes empecilhos para que ocorra a implantação da participação do cidadão e esclarecimento maior sobre temas voltados à administração pública. Alguns desses empecilhos seriam a falta da cultura participativa no Brasil, bem como a dificuldade que um cidadão comum sente ao buscar entender o formato em que os dados estão disponibilizados para poder

assim relacionar com outras informações de fontes variadas. Pereira (2014) ressalta que a burocracia é outro empecilho para que a sociedade possa contribuir de forma proativa na gestão e nas ações da Administração Pública, já que a mesma engessa o estado e caracteriza uma falta de agilidade e fluidez em um cenário em que as relações sociais são mediadas pela tecnologia e existem mecanismos como as redes sociais, dispositivos de comunicação instantânea e demais avanços caracterizados pela Sociedade da Informação.

2.4 LEI DE ACESSO À INFORMAÇÃO

A Lei de Acesso à Informação (LAI), de número 12.527 de 18 de Novembro de 2011, foi implementada no ano de 2012 na Administração Pública Federal a partir de decreto assinado pela então presidente Dilma Rousseff no dia 16 de maio de 2012 (BRASIL, Ministério da Transparência, Fiscalização e Controladoria-Geral da União, 2016).

Essa legislação versa sobre o dever do Estado de garantir o direito ao acesso à informação que deverá ser disponibilizada “mediante procedimentos objetivos e ágeis, de forma transparente, clara e em linguagem de fácil compreensão” (BRASIL, 2011).

O Ministério da Transparência, Fiscalização e Controle é o órgão responsável por acompanhar e garantir a aplicação da LAI no âmbito da administração pública, bem como pelo treinamento e capacitação de agentes públicos no desenvolvimento de boas práticas de transparência. Outras atribuições do Ministério da Transparência quanto ao escopo contemplado pela Lei de Acesso à Informação seriam o fomento a uma cultura de transparência, bem como a disseminação de uma consciência coletiva do direito fundamental de acesso à informação. O Ministério da Transparência também teria o poder de avaliar negativas de acesso à informação, conforme prescrito dentro dos autos da lei (BRASIL, Ministério da Transparência, Fiscalização e Controladoria-Geral da União, 2016).

Os procedimentos previstos na LAI são destinados a assegurar o direito fundamental de acesso à informação, devendo ser executados em conformidade com alguns princípios básicos da administração pública e com as seguintes diretrizes:

- I - observância da publicidade como preceito geral e do sigilo como exceção;
- II - divulgação de informações de interesse público, independentemente de solicitações;
- III - utilização de meios de comunicação viabilizados pela tecnologia da informação;

IV - fomento ao desenvolvimento da cultura de transparência na administração pública;
V - desenvolvimento do controle social da administração pública. (BRASIL, 2011)

De acordo com o artigo 4º da Lei de Acesso à Informação, são apresentados algumas definições importantes para contextualizar os aspectos envolvidos no tema. Por exemplo informação são *"dados, processados ou não, que podem ser utilizados para produção e transmissão de conhecimento, contidos em qualquer meio, suporte ou formato"* (BRASIL, 2011). Já por exemplo disponibilidade significa *"a qualidade da informação que pode ser conhecida e utilizada por indivíduos, equipamentos ou sistemas automatizados"* (BRASIL, 2011). Quanto a tratamento da informação, a LAI define que é o *"conjunto de ações referentes à produção, recepção, classificação, utilização, acesso, reprodução, transporte, transmissão, distribuição, arquivamento, armazenamento, eliminação, avaliação, destinação ou controle da informação"* (BRASIL, 2011).

Sobre a disponibilização dos dados públicos, a Lei de Acesso à Informação (LAI) estabelece que as ferramentas de disponibilização tenham entre outros aspectos, ferramentas de pesquisa que permitam o acesso à informação de forma objetiva, possibilidade de gravar os dados em formatos eletrônicos abertos como planilhas e textos, possibilitar acesso automatizado por sistemas externos em formatos abertos e estruturados, divulgar os formatos utilizados para estruturar a informação, bem como garantir preceitos básicos como autenticidade, integridade e temporalidade e acessibilidade para pessoas com deficiência. A LAI também prega que as informações disponibilizadas devem ser acompanhadas de meios para que o cidadão entre em contato com o órgão ou entidade detentora do portal de acesso à informação, através de indicações de local e instruções *"que permitam ao interessado comunicar-se por via eletrônica ou telefônica"*, de forma a se obter esclarecimentos sobre os dados disponibilizados e/ou formatos adotados, bem como solicitar informações e esclarecimentos caso o interessado julgar necessário (BRASIL, 2011).

2.5 DADOS ABERTOS

De acordo com o Ministério da Transparência e Controladoria Geral da União (2013), os dados abertos são *"dados que são livremente disponíveis para todos utilizarem e redistribuir como desejar, sem restrição de licenças, patentes ou mecanismos de controle"*. A Open Knowledge Foundation (2018) determina que os dados abertos podem estar restritos *"no máximo, à exigência de atribuição de fonte e compartilhamento pelas mesmas regras"*.

O Portal Brasileiro de Dados Abertos (2018) defende que os dados abertos são pautados por três leis e oito princípios básicos, sendo as leis definidas especificamente para os Dados Abertos Governamentais, mas podem ser aplicadas aos Dados Abertos de uma forma geral:

1. Se o dado não pode ser encontrado e indexado na Web, ele não existe;
2. Se não estiver aberto e disponível em formato compreensível por máquina, ele não pode ser reaproveitado; e
3. Se algum dispositivo legal não permitir sua replicação, ele não é útil. (PORTAL BRASILEIRO DE DADOS ABERTOS, 2018)

Entretanto, esses princípios foram desenvolvidos por um grupo de trabalho em 2007 na Califórnia, nos Estados Unidos da América. Quando a lei brasileira de dados abertos foi elaborada, estes princípios definidos anteriormente pela lei americana foram aproveitados. Todos os princípios originalmente foram conceitualizados para dados abertos governamentais, mas podem ser aplicados a dados abertos de forma geral, com exceção do primeiro (PORTAL BRASILEIRO DE DADOS ABERTOS, 2018).

1. **Completo.** Todos os dados públicos são disponibilizados. Dados são informações eletronicamente gravadas, incluindo, mas não se limitando a, documentos, bancos de dados, transcrições e gravações audiovisuais. Dados abertos são dados que não estão sujeitos a limitações válidas de privacidade, segurança ou controle de acesso, reguladas por estatutos.
2. **Primários.** Os dados são publicados da forma com que são coletados na fonte, com a mais fina granularidade possível, e não de forma agregada ou transformada.
3. **Atuais.** Os dados são disponibilizados o quanto rapidamente seja necessário para preservar o seu valor.
4. **Acessíveis.** Os dados são disponibilizados para o público mais amplo possível e para os propósitos mais variados possíveis.
5. **Processáveis por máquina.** Os dados são razoavelmente estruturados para possibilitar o seu processamento automatizado.
6. **Acesso não discriminatório.** Os dados estão disponíveis a todos, sem que seja necessária identificação ou registro.
7. **Formatos não proprietários.** Os dados estão disponíveis em um formato sobre o qual nenhum ente tenha controle exclusivo.
8. **Livres de licenças.** Os dados não estão sujeitos a regulações de direitos autorais, marcas, patentes ou segredo industrial. Restrições razoáveis de privacidade, segurança e controle de acesso podem ser permitidas na forma regulada por estatutos. (PORTAL BRASILEIRO DE DADOS ABERTOS, 2018)

O grupo que desenvolveu os princípios dos dados abertos também definiu que a conformidade com esses princípios precisa ser verificável e uma pessoa deve ser designada como contato responsável pelos dados (PORTAL BRASILEIRO DE DADOS ABERTOS, 2018).

O Tribunal de Contas da União apresenta 5 razões para que as organizações públicas invistam em iniciativas de abertura de dados públicos, sendo elas: a

transparência na gestão pública, a contribuição da sociedade com serviços inovadores ao cidadão, o aprimoramento na qualidade nos dados governamentais, a viabilização do desenvolvimento de novos negócios e a obrigatoriedade por lei (TRIBUNAL DE CONTAS DA UNIÃO, 2016).

Weinsten e Goldstein (2012) afirmam que dados abertos são os precursores do conceito de governo aberto, firmado na pressão cívica por transparência e controle social, oriundos da possibilidade fornecida pela Lei de Acesso à Informação norte-americana e do conceito de democracia participativa.

2.6 GOVERNO ABERTO

De acordo com a OGP - acrônimo de Open Government Partnership (2011) a expressão governo aberto (do inglês, *open government*) refere-se a projetos e ações que visam a promoção da transparência, a luta contra a corrupção, ao incremento da participação social na tomada de decisões por parte dos agentes públicos e ao desenvolvimento de novas tecnologias, de forma a tornar os governos mais abertos, efetivos e responsáveis.

Há quatro princípios estabelecidos que definem o conceito de governo aberto sendo eles: a transparência, a prestação de contas e responsabilização (Accountability), a participação cidadã e a tecnologia e inovação.

Transparência

As informações sobre as atividades de governo são abertas, compreensíveis, tempestivas, livremente acessíveis e atendem ao padrão básico de dados abertos.

Prestação de Contas e Responsabilização (Accountability)

Existem regras e mecanismos que estabelecem como os atores justificam suas ações, atuam sobre críticas e exigências e aceitam as responsabilidades que lhes são incumbidas.

Participação Cidadã

O governo procura mobilizar a sociedade para debater, colaborar e propor contribuições que levam a um governo mais efetivo e responsivo.

Tecnologia e Inovação

O governo reconhece a importância das novas tecnologias no fomento à inovação provendo acesso à tecnologia e ampliando a capacidade da sociedade de utilizá-la.

(OGP - acrônimo de Open Government Partnership (2011))

A Declaração de Governo Aberto da OGP (2011) ressalta que para um governo ser considerado aberto, ele deve buscar alcançar quatro objetivos, sendo eles: “- *umentar a disponibilidade de informações sobre atividades governamentais; apoiar a participação social; implementar os padrões mais altos de integridade*

profissional da Administração e ampliar o acesso a novas tecnologias para fins de abertura e prestação de contas”.

De forma a se adequar aos preceitos de governo aberto, o Brasil desenvolveu a INDA (Infraestrutura Nacional de Dados Abertos) como política para promover o acesso às informações oriundas das ações do Poder Executivo Federal para os cidadãos interessados através do Portal Brasileiro de Dados Abertos (ARAÚJO, 2017). O desenvolvimento de uma base nacional de informações levou estados e municípios a desenvolver ações para disponibilização de dados a níveis estaduais e municipais (CUNHA, 2015). Cunha estima que cerca de 70% das Unidades Federativas possuem dados abertos em algum dos três principais portais existentes (Portal de Dados Abertos, Portal da Transparência e Portal do Executivo), enquanto nas capitais dos estados o percentual é de 55,5% e nas grandes cidades é de 44,1%. A esfera da maior parte dos dados governamentais apresentados é de operações financeiras. (CUNHA, 2015)

2.7 DADOS ABERTOS GOVERNAMENTAIS

Os Dados Abertos Governamentais são *“uma metodologia para a publicação de dados do governo em formatos reutilizáveis, visando o aumento da transparência e maior participação política por parte do cidadão, além de gerar diversas aplicações desenvolvidas colaborativamente pela sociedade”* (Ministério do Planejamento, Desenvolvimento e Gestão, 2018). No Brasil diversos órgãos publicaram os dados na web para que os cidadãos possam acompanhar os resultados das ações do governo (Ministério do Planejamento, Desenvolvimento e Gestão, 2018).

Os Dados Abertos Governamentais possuem as mesmas leis e os mesmos princípios dos dados abertos, com a diferença que o princípio I somente se aplica aos dados governamentais (PORTAL BRASILEIRO DE DADOS ABERTOS, 2018).

Princípio I - Completos. Todos os dados públicos são disponibilizados. Dados são informações eletronicamente gravadas, incluindo, mas não se limitando a, documentos, bancos de dados, transcrições e gravações audiovisuais. Dados públicos são dados que não estão sujeitos a limitações válidas de privacidade, segurança ou controle de acesso, reguladas por estatutos.

(PORTAL BRASILEIRO DE DADOS ABERTOS, 2018)

Os dados abertos movimentam comunidades no mundo inteiro para cobrar informações públicas de governos, propiciando assim a geração de cenários e soluções que permitam aos cidadãos colaborar com os processos de governo e com o controle social das políticas. Essas comunidades praticam o que é conhecido como *ciberativismo* e *hacktivismo* (OPEN DATA, 2018).

2.8 CIBERATIVISMO E HACKTIVISMO

Ciberativismo pode ser considerado como um tipo de ativismo realizado por grupos politicamente motivados, que utilizam como bases redes como a Internet - com o objetivo de realizar, mobilizar e divulgar causas políticas, culturais, sociais ou ambientais (GRUPO CIBERATIVISMO, 2016). Para Vegh (2003, p.71), o ciberativismo pode ser definido como *“um movimento politicamente motivado que ocorre na internet”*.

Moraes (2015) defende que o ciberativismo é uma *“arena complementar de mobilização e politização, somando-se a assembleias, passeatas, atos públicos e panfletos que acontecem no que conhecemos como vida real”*. De acordo com Moraes (2015), o ciberativismo rompe o paradigma da distribuição de informação e reivindicação, já que não depende dos meios tradicionais de comunicação - a mass media. O uso da internet configura-se como uma alternativa acessível de propagação de ideias e reivindicações, já que pode ser utilizado por qualquer um que possua acesso às redes, fornecendo a chance de todo mundo *“se tornar intérprete de uma causa”*.

Lemos (2014) ressalta que o ciberativismo também impacta em mudanças de paradigma de distribuição de conteúdo e de informação, em que existe uma passagem dos meios mais populares hoje - o mass media (TV, rádio, imprensa e cinema) para formas individualizadas de produção, difusão e estoque de informações, subvertendo o conceito de difusão por hierarquia de árvore (um-todos) e possibilitando a difusão pela multiplicidade do rizoma (todos-todos), ou seja muda-se de uma passividade na recepção de informação, para uma situação em que o usuário consome a informação de forma ativa. Lemos (2014) também destaca que isso pode ocasionar problemas como *“a difusão de informações que não correspondem à realidade ou são enviesadas de acordos com conceitos anteriormente estabelecidos em seu meio”*.

O ciberativismo se desenvolveu principalmente depois da eclosão da web 2.0, que possibilita mais interação e participação entre os usuários na criação e *“organização de um conteúdo mais dinâmico e democrático”* (GRUPO CIBERATIVISMO, 2016).

Lemos (2014) separa o ciberativismo em três grandes grupos, sendo eles: conscientização e informação; organização e mobilização; e *“hacktivism”*. Silveira (2010) define que hackers *“são programadores de computador que solucionam problemas de informática utilizando meios não convencionais”*. Já Raymond (2011)

define hacker como sendo *“um indivíduo que se dedica, com intensidade incomum, a conhecer e modificar aspectos mais internos de dispositivos, programas e redes de computadores”*. Raymond também ressalta que graças a conhecimentos fora dos padrões sobre tecnologia, um hacker *“frequentemente consegue encontrar soluções e efeitos extraordinários, que extrapolam o limite do funcionamento ‘normal’ dos sistemas desenvolvidos por seus criadores, incluindo ainda contornar barreiras que deveriam impedir o acesso e controle de certos dados”*.

Machado (2013) diz que as origens do Hacktivismismo remontam a meados da década de 90, devido ao engajamento na grande rede de colaboração com o apoio ao movimento zapatista no México, considerado o *“primeiro movimento de guerrilha informacional”* da História. Ao longo dos anos o Hacktivismismo ganhou diversos grupos que executaram ações como *“interceptação de dados, desenvolvimento de aplicativos que permitissem furar bloqueios de censura na internet, ataques de negação de serviço (DDoS), paródias virtuais e ainda desfiguramento de sites, entre outros”* (MACHADO, 2013).

Segundo Castells (2009), os grupos de hacktivismismo, ativistas digitais e pessoas que acreditam no poder da internet começaram a *“dar mais atenção para temas como transparência pública e democracia no acesso aos dados públicos”*. As redes sociais propiciaram novas formas de comunicação, como a comunicação em massa. E essas mesmas redes sociais podem se tornar aliados importantes na construção, execução e difusão de um modelo de democracia participativa (CASTELLS, 2009).

2.9 DEMOCRACIA PARTICIPATIVA

Segundo Cordi (2000), democracia participativa pode ser entendida como a possibilidade dos cidadãos interferir diretamente nos procedimentos de tomada de decisão e controlar o exercício do Poder por meio dos parlamentares eleitos nesse contexto. É um regime onde existem mecanismos que possibilitam o controle da sociedade civil sobre a administração pública, não limitado apenas a questão do voto, mas também criando soluções que permitam que o controle e a opinião dos cidadãos entrem na pauta política das casas de representação (ALESP, 2016).

Lüchmann (2002) estabelece que a democracia participativa pode ser considerada um *“modelo ou ideal de justificação de exercício de poder político pautado no debate público entre cidadãos livres e em condições de igual participação”*. Também advoga que *“legitimidade das decisões políticas advém de processos de discussão orientados pelos princípios da inclusão, do pluralismo, da igualdade participativa, da autonomia e da justiça social, conferindo um aspecto de reordenamento na lógica de poder político tradicional”*.

Uma das maiores ações de democracia participativa que temos no Brasil hoje é o Orçamento Participativo. Genro (2001) destaca que o orçamento participativo tem como intuito destinar parte dos recursos do orçamento a demandas comunitárias abertas aos cidadãos, onde primeiro são coletadas as propostas, para que depois sejam definidas prioridades e no final essas sejam encaminhadas ao governo para que ele possa atender a solicitação com a verba do Orçamento Participativo. Um estudo do IPEADATA, revelou que nos municípios em que são adotados as práticas do Orçamento Participativo, indicadores sociais apresentam avanços, exceto no Índice de Theil, que mede a concentração de renda (VARGAS, 2007).

Para Oliveira (2017), *“face a passagem das tecnologias analógicas às tecnologias digitais, o fluxo do processo informacional e a direção dos fluxos comunicativos é alterado, trazendo novas formas e meios de participação da sociedade na gestão e participação democrática”*. Neste cenário, destaca-se a contribuição que as Tecnologias da Informação e Comunicação (TICs) possui, de forma a maximizar as oportunidades dos cidadãos no processo político, ampliando o conceito de esfera pública e reduzindo barreiras entre o cidadão e a Administração Pública. Hoje um setor que utiliza a tecnologia de forma intensiva para poder prestar contas à sociedade é o setor financeiro do Governo Federal, responsável pelo controle de gastos. Para tornar pública as informações financeiras do governo foi desenvolvido o Portal da Transparência, que é hoje o maior case de sucesso utilizando o conceito de governo aberto e democracia participativa (OLIVEIRA, 2017).

2.10 PORTAL DA TRANSPARÊNCIA

A divulgação e disponibilização dos dados públicos permite ao cidadão acompanhar os gastos e receitas executadas pela Administração Pública. O Portal da Transparência é um canal em que qualquer cidadão pode acompanhar consultas relativas aos gastos realizados pelo poder público - sejam eles provenientes de ações realizadas pela administração direta, por autarquias, fundações, casas legislativas, etc (TORRES, 2018).

O Portal da Transparência é *“uma página do Portal Oficial do Órgão Público, Federal, Estadual e Municipal, destinado a divulgar através da Internet, os dados e informações referentes aos atos administrativos dos órgãos da Administração Pública. O Portal apresenta dados, informações e demonstrativos sobre a execução orçamentária de cada exercício fiscal, nos termos da Lei de Responsabilidade*

Fiscal, bem como informações sobre desembolso com fornecedores em suas diferentes modalidades” (SÁ, 2013).

O Portal da Transparência foi lançado pela Controladoria Geral da União em novembro de 2004 para que o cidadão acompanhe a aplicação dos recursos públicos em âmbito federal. Através do Portal, o cidadão também se informa sobre os repasses que o seu município recebeu da Administração Federal. Desde o início, o portal ganhou recursos tecnológicos, aumentou a oferta de dados ano após ano e consolidou-se como a principal ferramenta de transparência de dados dentro do país, sendo reconhecido dentro e fora do Brasil, até mesmo com premiações internacionais (PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL, 2018).

Para atender de forma mais eficiente a população quanto a crescente demanda e obrigações de Transparência, no ano de 2018 foi desenvolvido um projeto para reestruturar o Portal. Como resultado desse projeto foi lançado no final de 2018 o novo Portal da Transparência, onde são disponibilizadas formas diversas de apresentação de dados, um mecanismo de busca integrada e intuitivo, melhorias de usabilidade e de acessibilidade, mais recursos gráficos para facilitar na compreensão das informações pelo cidadão, bem como integração com redes sociais, adequação e responsividade para plataformas móveis e outras ferramentas que permitem uma maior interatividade (PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL, 2018).

As informações do Portal são provenientes de diversas fontes, entre as quais os grandes sistemas estruturados do Governo Federal, como o Siafi (Sistema Integrado de Administração Financeira do Governo Federal) e o Siape (Sistema Integrado de Administração de Recursos Humanos), bases de benefícios sociais, faturas de Cartões de Pagamento do Governo Federal, base de imóveis funcionais, entre outras. Os órgãos repassam essa informação para a CGI que reúne, trata e as disponibiliza na ferramenta. A periodicidade com que as informações chegam ao Portal é variável de acordo com o domínio tratado, e os formatos disponibilizados englobam desde painéis, consultas detalhadas, gráficos e dados abertos (PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL, 2018).

Na reformulação do Portal da Transparência um dos recursos adicionados foi a realização de consultas através de API (*Application Programming Interface*) permitindo assim que o acesso dos dados aos desenvolvedores ocorra através de uma interface direta sem precisar navegar pelo site ou utilizar robôs para a obtenção das informações de forma automatizada. A API foi implementada utilizando REST (*Representational State Transfer*) por ser um formato mais simples de ser utilizado, bem como por ser o padrão de mercado e escolha da grande maioria dos desenvolvedores. O acesso à API REST facilita a execução de consultas e a

obtenção de dados que possibilitem aos desenvolvedores extrair informações relevantes através de processos envolvendo análise de uma grande quantidade de dados (Big Data) disponíveis sobre os mais variados domínios da administração pública (PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL, 2018).

2.11 BIG DATA

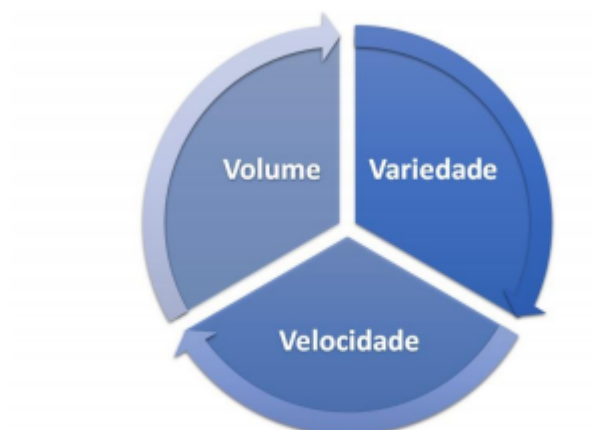
As origens do Big Data remontam ao ano de 1997, onde dois cientistas da NASA (*National Aeronautics and Space Administration*) Michael Cox e David Ellsworth abordaram um problema crônico na área da computação, onde a quantidade de recursos disponíveis no computador eram insuficientes para processar tamanha quantidade de informação, ou seja, o volume de dados era tão grande que os recursos tecnológicos existentes não conseguiam processar rapidamente inúmeras informações porque não tinham espaço para expandir. Esse artigo trata também sobre como a demanda de informações das pessoas aumentava e como essas mesmas pessoas estavam produzindo e divulgando muito mais conhecimento (COX; ELLSWORTH, 1997).

Esse aumento da quantidade de informação, levou a cientistas e pesquisadores da área de tecnologia a se aprofundar e realizar novas pesquisas que formariam a base do que conhecemos como Big Data. Antes disso, o conceito de Big Data estava atrelado a tudo aquilo que não conseguiam explicar e definir dentro das novas tecnologias (COX; ELLSWORTH, 1997).

Davenport (2014) define Big Data como sendo “*uma massa de dados brutos fluindo pelos sistemas*”, enquanto que para Schönberger e Cukier(2013), Big Data se trata de um grupo de tecnologias lidando com várias informações das organizações em fluxo contínuo. Com o decorrer do tempo e com o desenvolvimento da variedade dos dados, que não representavam apenas textos e formatos arcaicos planilhados, o volume passou a não ser a única característica fixa e único do Big Data (MAYER - SCHÖNBERGER; CUKIER, 2013).

Laney (2001) ressalta que mais dois fatores além do volume formam as características que definem o Big Data, compondo o conjunto de três segmentos que ficou conhecido como 3V's: volume, variedade e velocidade, conforme exemplificado na figura 2.

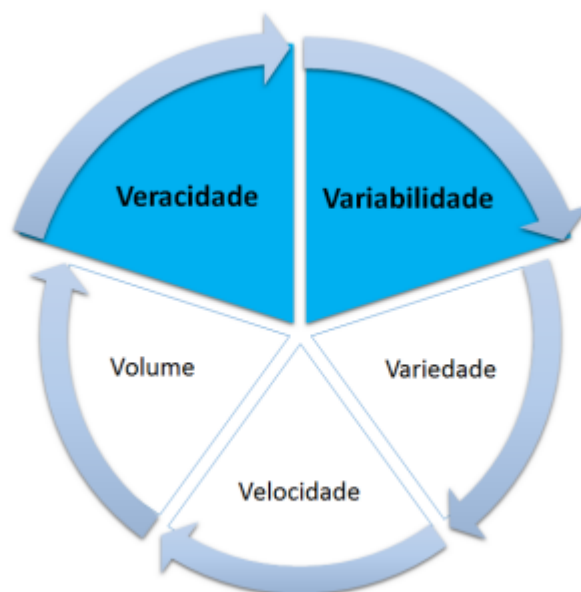
Figura 2 - Características do Big Data - 3V's



Fonte: (IVANOV; KORFIATIS; ZICARI, 2013)

Com o desenvolvimento de novos estudos na área foram propostas novas outras características fundamentais para compreender a estrutura do Big Data, mais 2 V's que vão além das definições apresentadas anteriormente por Laney (2001). Essas novas características possuem um enfoque mais qualitativo, enquanto que as dimensões definidas por Laney possuem uma perspectiva basicamente quantitativa. As novas características estão ligadas aos dados não sendo apenas variáveis contidas dentro do sistema, mas também precisam ser levado em conta aspectos como a confiabilidade da informação (**Veracidade**) e em aspectos que não foram contextualizados naquele momento (**Variabilidade**) (IVANOV; KORFIATIS; ZICARI, 2013). A figura 3 apresenta as características do Big Data atualizadas.

Figura 3 - Características do Big Data - 5V's



Fonte: (IVANOV; KORFIATIS; ZICARI, 2013)

2.11.1 Volume

A característica **Volume** foi o primeiro atributo definido para o Big Data. Ele tem origem na quantidade progressivamente maior do número de dados dentro dos sistemas, obrigando ao desenvolvimento de novos métodos para auxiliar na administração desse grande fluxo de dados. Entre os métodos que foram utilizados para lidar com essa grande quantidade de informação, está o Big Data. O volume é caracterizado por uma intensa troca de informações, com uma alta quantidade acessada e utilizada de informações dentro das redes e/ou sistemas. (MAYER - SCHÖNBERGER; CUKIER, 2013).

No ano de 2015 estima-se que eram produzidos milhares de petabytes de informação com as interações diárias de dados acumulados. Para se ter uma ideia da dimensão de tudo isso, 1 petabyte é o equivalente a 1000 terabytes e 1 terrabyte é o equivalente a 1000 gigabytes. Como os dados estimados são anteriores a 2015, então a conclusão que podemos chegar é que podem ser bem maiores hoje em dia, devido a expansão e consolidação de serviços como o streaming de filmes como a Netflix, Amazon Prime, Youtube, etc e das redes sociais, entre elas o Facebook e o Instagram. As redes sociais por exemplo, eram a maior responsável pelo volume de dados produzidos na época do estudo (SCUDERE, 2015).

2.11.2 Variedade

É a dimensão que representa diferentes topologias de dados e de âmbitos possíveis. Na formação dessa dimensão são considerados elementos de todas as origens, formatos e extensões variadas, estruturados ou não. É a característica que atribui diversidade aos dados dentro dos sistemas (IVANOV; KORFIATIS; ZICARI, 2013).

Nos primórdios dos sistemas estruturados, o que era conhecido como informação vinha apenas como textos produzidos de origem documental, impressos ou distribuídos através do meio digital. Com a variabilidade da informação, agora o desafio é muito maior, já que existem mídias de diferentes estilos e formatos e extensões que trazem novas variáveis na análise de dados. “Quando consideramos esses novos formatos, podemos levar em conta que cada email trocado, cada vídeo enviado em plataformas de vídeo como o Youtube ou cada tweet realizado pode ser considerado informação e é agregado para compor a análise no sistema” (IVANOV; KORFIATIS; ZICARI, 2013).

2.11.3 Velocidade

É a característica que descreve o quão rápido esses dados podem ser recuperados, armazenados e processados. A característica velocidade hoje é considerada um dos atributos mais significativos, já que com o grande volume de dados, oriundos de diferentes tipos de dados e diversas fontes, é necessário que tenhamos velocidade para processar essa informação de forma a se obter conclusões que tornem esses dados relevantes ao âmbito que está sendo tratado (IVANOV; KORFIATIS; ZICARI, 2013).

2.11.4 Variabilidade

É a característica que determina as distintas interpretações que os dados possam ter dependendo de parâmetro, ângulo ou ainda quais são as variáveis que serão trabalhadas, dentro de um contexto específico a ser analisado. A variabilidade pode retornar diferentes resultados ou até mesmo uma nova concepção que não foi vista em realidades anteriores, já que esse aspecto possui a ideia de que todo o dado pode ser utilizado para o objetivo da pesquisa, se não agora, mas em algum dado momento no futuro (; KORFIATIS; ZICARI, 2013).

2.11.5 Veracidade

É a característica que determina a confiabilidade da informação e está relacionada não só a integridade das fontes de informação, mas a precisão que os dados estão disponíveis, evitando assim que sejam realizadas análises que destoam da realidade dos dados analisados (IVANOV; KORFIATIS; ZICARI, 2013).

2.12 DATA ANALYTICS

Data Analytics é o “*processo de analisar informações (dados) com um propósito específico*”. Ou seja, pesquisar e responder perguntas com base em dados e com uma metodologia clara para todos os participantes. Um projeto de Analytics pode ser executado de maneira individual, mas dependendo da dinâmica e complexidade das questões e dos dados, é muito comum que um projeto de análise de dados seja colaborativo, com indivíduos atuando em papéis diferentes ao longo de cada etapa (AQUARELA, 2018).

Alguns conceitos são muito importantes para entender o que é o Data Analytics. São eles: os indicadores, a cultura de Analytics e a comunicação entre as pessoas que realizam a análise. De acordo com Aquarela (2018), os processos de Analytics seguem um padrão bastante consistente e possuem basicamente os seguintes passos:

- Origem dos dados.
- Limpeza e Transformação.
- Análise de negócio.
- Validação do Analytics.

A origem dos dados está no começo de todo o processo, ou seja, é a fonte do qual os dados que vamos analisar estão disponíveis. É muito comum que essa fonte esteja disponível através de arquivos .csv ou em algum formato típico do Excel, mas também podem estar presentes em formatos mais técnicos ou diretamente do banco de dados (AQUARELA, 2018).

De acordo com Aquarela (2018), uma vez que o dado está extraído, é muito comum que ele não esteja pronto para a análise. Dados que estão escritos de maneiras diferentes, mas que se referem as mesmas coisas; informações que estão duplicadas na fonte e informações que estão incompletas são alguns exemplos de casos que podem aparecer nesse cenário, impossibilitando uma análise assertiva da

massa de dados. Por isso é necessário a execução de uma série de processos que visam realizar uma limpeza e organização dos dados antes de qualquer análise, seja ela feita de forma manual ou através de algoritmos de inteligência artificial (AQUARELA, 2018).

Entretanto, outro aspecto importante do exercício de Data Analytics de acordo com Aquarela (2018), é definir de forma clara as perguntas que precisam ser respondidas, de forma a manter o foco e a objetividade do trabalho. Por fim, o último grande aspecto a se considerar para entender as nuances do processo de análise de dados é a validação do resultado, ou seja verificar se os ajustes e os indícios sugeridos na etapa de processamento estão surtindo efeitos reais e são suficientes para que valha a pena continuar seguindo a mesma linha de pesquisa exploratória (AQUERELA, 2018).

De acordo com Davenport (2014) a análise de dados pode ser dividida em três eras principais, sendo elas a era Analytics 1.0, a era Analytics 2.0 e a era Analytics 3.0. Essas eras não segue propriamente um fluxo onde uma começa quando termina a outra, mas se entrelaçam em algum momento da história.

O Analytics 1.0 começou no ano de 1954 e durou até 2005, sendo caracterizado pelas fontes de dados serem pequenas e internas, ou seja, há muito poucos dados disponíveis e coleta deles dentro do sistema/empresa. Outros aspectos bastante característicos dessa época é que os dados são armazenados antes mesmo de serem analisados e a atividade analítica consistia basicamente na produção de relatórios (DAVENPORT, 2014).

O Analytics 2.0 começa nos anos 2000, onde com o início do Big Data as empresas já estão voltadas para as novas explorações de dados online. Nessa era um aspecto bastante característico é que as empresas já tinham dados sobre os clientes, mas não utilizavam como comparação de análise com outros dados (DAVENPORT, 2014).

Hoje se encontramos no processo de transição entre o Analytics 2.0 e o Analytics 3.0, onde ocorrerá a junção de pontos das duas eras anteriores mas possuirão uma característica qualitativamente bem marcante: será uma mescla entre Big Data e Analytics tradicional com velocidade e impacto, onde serão obtidos grandes ofertas e insights para qualquer empresa e seus serviços. Podemos ver esse modelo em prática em ferramentas como a recomendação de compras da Amazon e a recomendação do que assistir da Netflix, que buscam aproximar o usuário de seus catálogos presentes no serviço, de acordo com buscas realizadas ou com o material que o usuário já consumiu anteriormente (DAVENPORT, 2014).

Há algumas ferramentas que auxiliam no processamento de grandes quantidades de dados, sendo elas: computação em nuvem, YARN, HDFS,

MapReduce, Hadoop, MPP, Hbase e o Spark (GALDINO, 2016), TensorFlow e TensorFlow.js. Nesse trabalho exploraremos duas possibilidades de realizar a análise dos dados: (1) adotar a ferramenta MapReduce e o Hadoop já que elas possuem uma extensa documentação disponível, além de uma comunidade ativa; (2) adotar o TensorFlow.js por se tratar de uma biblioteca Javascript que facilita o desenvolvimento de aplicações que realizam processos de Machine Learning.

2.13 PROGRAMAÇÃO UTILIZANDO MAPREDUCE

Com a escalada do volume e da variedade de dados oriundos do Big Data, há um aumento de dificuldades para o processamento das informações. A única abordagem viável para tratar esse volume tão grande de dados e de diferentes formatos é dividir e conquistar. Nessa estratégia a abordagem básica é dividir um problema grande em problemas menores, de forma que possamos tratar os subproblemas independentemente em diferentes threads, núcleos de processadores ou máquinas em clusters (LYN E DYER, 2010).

No entanto, em ambientes tradicionais de programação paralela ou distribuída, deve ser tratadas questões complexas como: forma de paralelização, tolerância a falhas, distribuição de dados entre os nós de processamento e balanceamento de cargas e acesso à memória compartilhada. Essas questões devem ser tratadas explicitamente pelo programador e podem dificultar o processo de análise de dados, além de desvirtuar o programador do principal a ser explorado: o domínio do problema (PAIVA E REVOREDO, 2016).

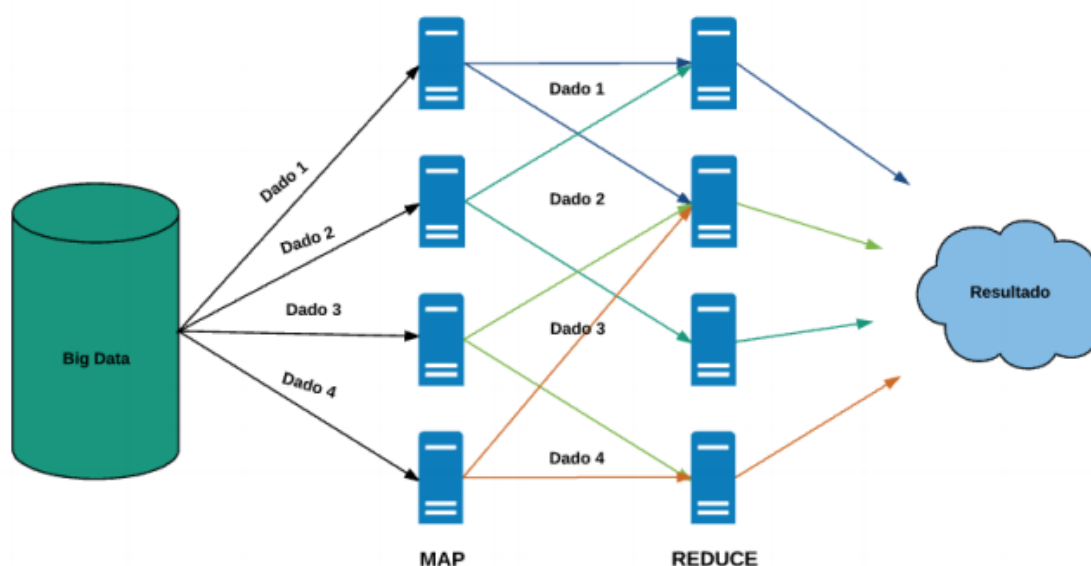
Para resolver questões como essa, foi desenvolvido o modelo de programação MapReduce. Ele é um modelo de programação paralela definido para tratar especificamente grandes volumes de dados, utilizando a inspiração da estratégia dividir para conquistar, mas sem onerar o programador com os problemas típicos do gerenciamento de aplicações distribuídas, permitindo que o desenvolvedor possa focar na solução do problema a ser tratado, deixando para a aplicação executar funções como o paralelismo e a distribuição do processamento entre os nós (DEAN E GHEMAWAT, 2004).

O modelo MapReduce roda em clusters de computadores, onde é implementada uma arquitetura mestre-escravo, em que o processamento é distribuído entre os nós de forma que um nó mestre fique responsável pelo gerenciamento e nós escravos fiquem responsáveis pelo processamento propriamente dito. Nesse modelo de programação, é necessário desenvolver apenas duas funções principais, a função Map e a função Reduce (PAIVA E REVOREDO, 2016).

As funções Map e Reduce trabalham basicamente com um conjunto de pares (chave-valor) intermediários passados para a função Reduce, que é responsável por processar e juntar todos esses pares (chave-valor) intermediários, associando os pares com a mesma chave em uma espécie de sumarização dos resultados. Os pares chave-valor gerados pelos nós de map são distribuídos de forma aleatória para os nós responsáveis pelo reduce. Entre as operações de Map e reduce, esses valores passam por uma operação que é conhecida de “Shuffle and Sort”, que é responsável por ordenar os pares de chave-valor pela chave, de forma que a função Reduce já receba os dados ordenados (PAIVA E REVOREDO, 2016).

Na figura 4 é demonstrado o funcionamento do algoritmo de MapReduce.

Figura 4 - Como funciona o MapReduce



Fonte: (DEAN; GHEMAWAT, 2004)

Existem diversas ferramentas que implementam o modelo de programação MapReduce, mas a ferramenta mais conhecida e utilizada pelo mercado é o Hadoop (PAIVA E REVOREDO, 2016).

2.14 HADOOP

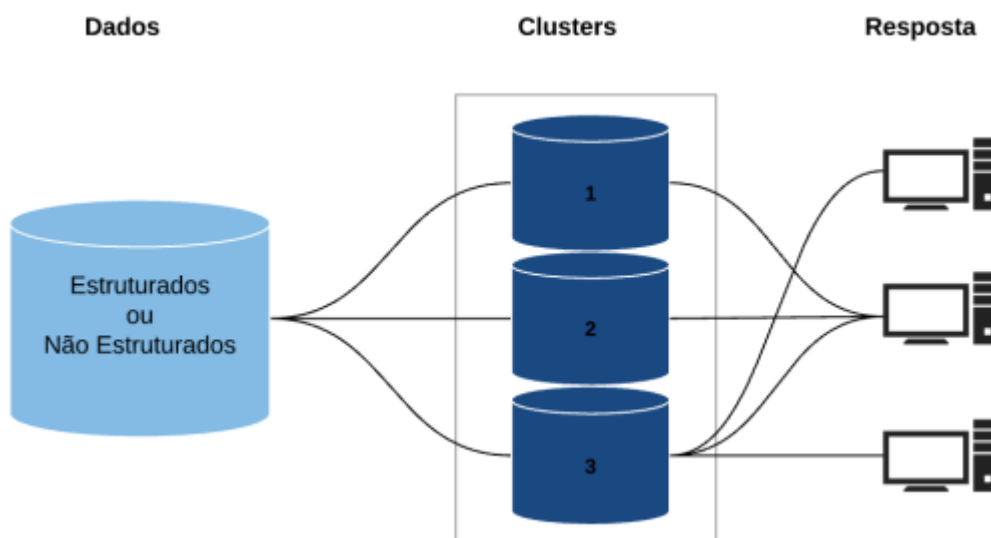
O Hadoop foi desenvolvido pela Apache desde 2005, tendo sido oficialmente lançado em 2011. A sua origem foi inspirado no conceito do que o Google desenvolveu através do MapReduce, tendo a sua estrutura baseado no mesmo estilo de processamento de dados (HADOOP, 2018; TAURION, 2015).

Desenvolvido como um projeto de código livre, o Hadoop processa uma grande quantidade de dados estruturados ou não, a partir de um único servidor,

distribuídos em clusters para armazenamento, com a resposta dependendo da análise realizada. Por ser uma ferramenta de código livre o Hadoop pode ser expandido e modificado para a criação de sistemas específicos sob medida que atendam as necessidades de cada empresa (HADOOP, 2018).

Um esquema do funcionamento do Hadoop é demonstrado na figura 5.

Figura 5 - Como funciona o Hadoop



Fonte: (HADOOP, 2018)

O Hadoop se divide em 4 módulos, sendo eles:

- Hadoop Common.
- Hadoop Distributed File System (HDFS).
- Hadoop YARN.
- Hadoop MapReduce (HADOOP, 2018).

Cada um desses módulos possui uma finalidade específica. Enquanto o módulo Hadoop Common encapsula as bibliotecas e utilitários que suportam os módulos do Hadoop; o módulo HDFS se trata de um sistema de arquivos distribuídos que fornece acesso de grande produtividade aos dados; o módulo Hadoop YARN realiza o escalonamento de tarefas e gerencia os recursos em clusters e o módulo Hadoop MapReduce, utilizando da YARN, tem a função de processar grandes conjuntos de dados (HADOOP, 2018).

2.15 TENSORFLOW

TensorFlow é um ecossistema que auxilia na resolução de desafios e problemas do mundo real utilizando Machine Learning. Ele oferece múltiplos níveis de abstração que permitem que o desenvolvedor possa configurá-lo para as suas necessidades, como a definição e treinamentos de modelo com a Keras API, que permitem que o TensorFlow e o próprio machine learning tenham uma curva de aprendizado muito mais simples (TENSORFLOW, 2020).

O TensorFlow fornece uma poderosa plataforma de experimentação e pesquisa, onde é possível utilizar o estado de arte no que se refere a definição e treinamento de modelos de dados (TENSORFLOW, 2020).

2.16 TENSORFLOW.JS

TensorFlow.js é um framework para desenvolver aplicações que utilizam tensores em Javascript. Um tensor é uma generalização de vetores e matrizes de grandes dimensões, sendo a unidade central de toda a arquitetura do framework (TENSORFLOW, 2020). Tensores são muito próximos a arrays multidimensionais e podem conter algumas propriedades:

- **rank**: define quantas dimensões o tensor contém.
- **shape**: define o tamanho de cada dimensão de dados.
- **dtype**: define o tipo de dados do tensor.

O TensorFlow.js funciona tanto no navegador quanto em servidores web rodando com Node, e há muitas configurações disponíveis para cada uma das plataformas, com recursos próprios e compartilhados entre elas. No navegador por exemplo, TensorFlow.js suporta a execução em dispositivos móveis, enquanto na versão Node, há suporte direto para comunicação direta com a TensorFlow API e também a possibilidade de rodar as ferramentas disponíveis em máquinas com menor quantidade de recursos, como CPU por exemplo (TENSORFLOW, 2020).

O TensorFlow utiliza conceitos muito interessantes de *machine learning*, como a possibilidade definir um modelo e suas camadas (sejam elas camadas de entrada, camadas intermediárias que guiarão o processo de aprendizagem e camadas de saída), de forma que este modelo possa ser treinado e retorne previsões (através de métodos como regressão e classificação, por exemplo) de acordo com os dados informados para esse nosso processo de treinamento. Esses conceitos poderão ser úteis para o desenvolvimento do trabalho.

2.17 MODELOS E CAMADAS

Em Inteligência Artificial (IA), um modelo é uma função com parâmetros que podem ser aprendidos, mapeando uma entrada para uma saída. Os considerados parâmetros ótimos (melhores parâmetros que poderíamos ter para realizar a análise) são obtidos treinando o modelo com os dados. Um modelo bem treinado consegue através dos dados de entrada, retornar saídas corretas e de acordo com aquilo que o mapeamento está esperando (TENSORFLOW, 2020).

No TensorFlow.js há duas formas de criar um modelo, sendo elas: a utilização da Layer API onde é construído um modelo utilizando camadas (conhecido como *Sequential Model* - ou modelo sequencial, em tradução livre), e pela Core API (outra API disponível) onde há a invocação de funções globais (conhecido como *Functional Model* - ou modelo funcional, em tradução livre). Para o nosso projeto iremos adotaremos o primeiro modelo, por ser o modelo mais comumente utilizado por outras aplicações do gênero (TENSORFLOW, 2020).

De acordo com a documentação do TensorFlow.js, o treinamento de um modelo consiste de alguns passos como definição dos dados de entrada, realização de previsões utilizando o modelo treinado e após isso comparando os resultados previstos com os dados verdadeiros, para que possamos assim alterar os parâmetros definidos para que o modelo possa realizar previsões melhores no futuro para esse mesmo conjunto de dados.

2.18 FERRAMENTAS AUXILIARES

No desenvolvimento deste trabalho serão utilizados algumas ferramentas auxiliares. Para o desenvolvimento de uma aplicação de cidadania participativa que permita ao cidadão acessar e fiscalizar as compras da merenda escolar será utilizado o *framework* para desenvolvimento híbrido Ionic. Esse *framework* permite que seja construído aplicações para os dois principais ecossistemas mobile (iOS e Android).

A API responsável pelo backend da aplicação será desenvolvida utilizando Node.js, devido ao fato de ser um *framework* web que permite utilizar o interpretador V8 Javascript Engine, possibilitando assim trabalhar com a mesma linguagem de desenvolvimento (Javascript) tanto no lado do cliente quanto no lado do servidor. Essa tecnologia não só traz rapidez ao desenvolvimento, como permite também que os próximos programadores que venham a realizar projetos futuros relacionados ao

presente Trabalho de Conclusão de Curso (TCC) precisam ter conhecimento de apenas uma linguagem de programação.

Para produzir indicadores que possam auxiliar os usuários na avaliação das compras serão utilizadas tecnologias como a API de Localidades do IBGE e o Tensor Flow, descritas em seções anteriores.

Os dados oriundos do resultado da obtenção de indicadores da compra da merenda escolar, bem como das avaliações e interação realizadas pelo aplicativo serão armazenadas em um banco de dados MySQL, e disponibilizados através de uma REST API desenvolvida com o Express.

2.18.1 Ionic Framework

Ionic Framework é um projeto open source (MIT license) que permite aos desenvolvedores construir aplicativos mobile performáticos e de alta qualidade, utilizando tecnologias web em que a maior parte dos desenvolvedores estão habituados - como HTML, CSS e JavaScript (IONIC, 2018).

O Ionic é focado na experiência e no *“look and feel”* de um app, utilizando o Cordova para acessar os recursos disponíveis nos smartphones, enquanto fornece ferramentas que simplificam o desenvolvimento do frontend das aplicações mobile (IONIC, 2018).

2.18.2 MySQL

O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL. Hoje é de propriedade da Oracle e possui mais de 10 milhões de instalações no mundo, sendo utilizado por empresas como o Google, Adobe, Zappos, Cisco e NASA, entre outras (MYSQL, 2020).

Uma das principais características do MySQL é a portabilidade (suportando praticamente qualquer plataforma atual), facilidade no gerenciamento dos dados e manuseamento, bem como compatibilidade com diversos drivers de praticamente todas as linguagens de programação estabelecidas no mercado. Outras características importantes desse SGBD são excelente desempenho e estabilidade, bem como a pouca exigência em relação a hardware para a sua implementação. Por ser um software open source com base em GPL, possui uma comunidade bastante ativa (MYSQL, 2020)

A popularização do MySQL ocorreu em grande medida devido à fácil integração com o PHP, onde o driver e a conexão já estavam presentes na versão nativa da linguagem. Outra característica bastante importante do MySQL é que

praticamente todos os pacotes de hospedagem oferecem instalações do banco como parte do pacote essencial (MYSQL, 2020). O WordPress (CMS) que detém cerca de 30% de todos os websites desenvolvidos no mundo tem o MySQL como seu banco de dados padrão (WORDPRESS, 2020).

2.18.3 Express

O Express é um framework que permite o desenvolvimento de aplicações web utilizando Node.js, lançado como open source sob a licença MIT. É destinado para desenvolver aplicações web e principalmente APIs. Logo após o seu lançamento ganhou espaço como o mais utilizado framework Node do mercado, se tornando referência no desenvolvimento de APIs (EXPRESS, 2020).

A estrutura do Express leva em um conceito bastante interessante: minimalismo. A partir deste, temos um framework que possui uma estrutura bastante enxuta e muitas outras features através de plugins. É comum a utilização do Express junto com outros conjuntos de tecnologias formando uma stack, como MEAN (Mongo - Express - Angular - Node) ou React-Express-Node-Postgres por exemplo (EXPRESS, 2020).

Algumas empresas que utilizam de forma massiva o Express em suas aplicações são o PayPal, a IBM, o Uber e o Fox Sports (EXPRESS, 2020).

2.18.4 API de Localidades

No desenvolvimento do presente trabalho utilizaremos a API de serviço de dados disponibilizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) onde será possível detectar todos os municípios vizinhos daqueles em que a compra foi realizada. Para que tal objetivo seja atendido, utilizaremos a API de dados de Localidades do IBGE.

A API de dados de Localidade do IBGE possui as principais divisões administrativas do nosso território, como por exemplo, distritos, mesorregiões, microrregiões, municípios e unidades federativas, sendo atualizadas anualmente conforme mudanças de organização do espaço são realizadas. Hoje a API do IBGE é referência em dados de organização administrativa no Brasil (IBGE, 2020).

3 TRABALHOS CORRELATOS

Durante o processo de pesquisa foi realizado o cruzamento de temas relacionados à corrupção, análise de dados, dados abertos e Big Data, retornando alguns artigos, monografias e projetos com informações relevantes para a fundamentação e desenvolvimento deste trabalho. Neles foram apresentados conceituações, possibilidades, modelos de desenvolvimento, tecnologias possíveis para tratamento de grandes volumes de informações, fontes de obtenção de dados públicos, bem como estudos de caso sobre análise de dados governamentais.

Paiva e Revoredo (2016) apresentam um artigo que estabelece um cenário sobre a transparência e as obrigações governamentais. Nele, é destacado que os entes governamentais devem manter portais da transparência que evidenciam as despesas executadas diariamente pela máquina pública. Paiva e Revoredo destacam que infelizmente, apenas a disponibilização das informações não assegura um aumento efetivo da transparência das entidades, pois o grande volume de dados disponíveis aliado com a falta de padrões torna inviável qualquer tipo de acompanhamento sistemático dessas informações. O artigo sugere então a aplicação de técnicas de programação paralela baseadas no paradigma de MapReduce, para fazer a identificação de um conjunto de produtos comprados pela Administração Pública, além de propor uma forma que facilite a visualização das disparidades encontradas nesse conjunto de dados. A solução proposta pelo artigo foi testada em um estudo de caso executado com os dados disponíveis no Portal da Transparência e os resultados obtidos apontaram que as técnicas utilizadas podem ser promissoras para questões ligadas à transparência, que normalmente tratam grande volume de dados, mas que nem sempre apresentam informações de qualidade.

Ahlert (2016) apresenta em seu artigo uma abordagem de enfrentamento da corrupção através do uso de tecnologias disponíveis no que conhecemos atualmente como sociedade da informação. O estudo realizado pelo artigo desenvolvido por Ahlert, está embasado na pesquisa doutrinária sobre a corrupção na administração pública e a busca por uma solução para tentar pelo menos reduzir a incidência desse fenômeno. O artigo foca primeiro nas transformações que as novas tecnologias causam na vida das pessoas, depois aborda o tema corrupção como um fenômeno multidisciplinar, para posteriormente aprofundar a pesquisa com o intuito de analisar se as tendências globais no enfrentamento da corrupção incluem a viabilidade do uso de aplicativos e programas informatizados para o

controle social das contas públicas, bem como, se a transparência pode ser uma ferramenta eficaz no combate da corrupção que assola o nosso país.

Souza (2017) analisa em sua monografia os conceitos e definições do modelo Big Data, contextualizando a evolução histórica que levou à criação e desenvolvimento desse modelo, bem como descrevendo os fundamentos, metodologias e tecnologias que são utilizadas pelo Big Data. A monografia apresenta o Big Data como uma grande quantidade de volume informacional existente e uma real necessidade de tratar esses dados para extrair e transformar essa massa em informação útil. O artigo conclui com potencialidades em diversas áreas, sejam elas desde recomendação de filmes para o usuário assistir no final de semana, bem como o uso do modelo em bibliotecas como apoio às tarefas cotidianas e operacionais. É abordado também questões como custos de gerenciamento e a questão da privacidade, tema que ganha relevância quando levamos em conta que estamos lidando com um grande volume de dados e possivelmente dados sensíveis para o usuário.

Paiva e Revoredo (2016) destacam em seu artigo que o governo brasileiro tem se empenhado em disponibilizar seus dados através do Portal da Transparência. No, entanto nem sempre o formato que os dados estão disponíveis é o mais apropriado para realizar análises sistemáticas. O artigo apresenta uma proposta de dissertação de mestrado que visa melhorar a qualidade dessas informações, com o desenvolvimento de uma técnica que permite a identificação automática dos produtos que são descritos de forma textual nas especificações das compras apresentadas nos portais de transparência.

A **Operação Serenata de Amor** é um *“projeto aberto que usa ciência de dados - as mesmas tecnologias utilizadas por gigantes como Google, Facebook e Netflix - com a finalidade de fiscalizar gastos públicos e compartilhar as informações de forma acessível para qualquer pessoa”* (OPERAÇÃO SERENATA DE AMOR, 2018).

Uma das ações mais significativas da Operação Serenata de Amor foi a criação da Rosie e do Jarbas. A Rosie é uma inteligência artificial com a capacidade de analisar os gastos reembolsados pela Cota para Exercício da Atividade Parlamentar (CEAP), de deputados federais e senadores, feitos no exercício da função, identificando comportamentos e gastos que podem ser considerados suspeitos e incentivando a população a fiscalizá-los e questioná-los. O Jarbas é um site onde é possível visualizar os gastos e descobrir mais sobre cada suspeita, de

uma maneira simples e descomplicada, levando a população a entender as informações geradas pela Rosie. É no Jarbas onde começa o trabalho de validação de uma suspeita levantada pela Rosie (OPERAÇÃO SERENATA DE AMOR, 2018).

Um dos objetivos do projeto é a expansão, onde o grupo mantenedor da Serenata visa expandir o alcance a área de atuação do projeto, seja ensinando novas coisas que possam auxiliar na identificação de fraudes pela Rosie, bem como adaptando toda essa tecnologia para outras esferas, como as câmaras municipais. A Serenata de Amor tem como missão levar mais informação para a população, *“transformando os dados públicos em conteúdo acessível por meio das redes sociais e por meio da produção independente de conteúdo”* (OPERAÇÃO SERENATA DE AMOR, 2018).

Alguns dos feitos mais significativos realizados pela operação são o reembolso de uma nota com o valor de R\$ 6.205,00 bem como a devolução do dinheiro de dois deputados que pediram reembolso de 13 refeições feitas no mesmo dia (G1, 2016).

O **Diagnóstico Público** (2020) é uma ferramenta de análise de dados públicos que utiliza os dados de portais de transparência governamentais para gerar relatórios instantâneos e intuitivos para o usuário. Venceu em 2017 o prêmio HackBrazil 2017, na Brazil Conference at Harvard & MIT, e tem como principal objetivo auxiliar gestores na administração e tomada de decisões, com foco em prefeituras, câmaras, institutos de pesquisa, órgãos de controle, etc. Nessa plataforma é possível acompanhar as contas de 2 mil municípios do Brasil, bem como há a possibilidade de consultas ao perfil das despesas e receitas dos municípios, finanças, licitações e credores - incluindo uma ferramenta que realiza a análise de compras e de riscos financeiros ao se fechar um contrato governamental.

Fiquem Sabendo (2020) é uma agência de notícias independente especializada na Lei de Acesso à Informação que possui o objetivo de divulgar informações que possam ser de interesse do cidadão, principalmente aquelas que o governo não divulga. Possui uma newsletter bastante assinada, que fornece notícias de bastidores, tutoriais, entrevistas e reportagens utilizando as informações disponíveis na Lei de Acesso à Informação (LAI). O nome da Newsletter é *“Don’t LAI to me”*, fazendo uma espécie de trocadilho com o acrônimo da lei. O link para acessar o website desta agência de dados independente é: <https://fiquemsabendo.com.br>

Brasil.io é uma iniciativa que visa tornar os dados disponíveis através da Lei

de Acesso à Informação (LAI) acessíveis para que possam ser utilizados por toda a população. A iniciativa surgiu devido a uma série de problemas com os dados disponibilizados pelos órgãos públicos, como por exemplo, disponibilização de arquivos em formatos que não estão estruturados, ou dados que estão em formatos que desencorajam ou impossibilitam a realização da análise de forma mais ampla. Alguns dos problemas apontados no manifesto da iniciativa são sites que até fornecem a informação, mas não possuem a opção de download dos dados, para que o usuário consiga manuseá-los fora do ambiente original; informações disponibilizadas em PDF (formato não-estruturado) em vez de planilhas eletrônicas (formato mais prático para realizar algum processo de análise e computação); planilhas em formatos fechados e proprietários (como XLS e XLXS, em detrimento de formatos abertos como o CSV) e informação fracionada (uma quantidade grande de arquivos, geralmente dividido por estados e municípios no lugar de um único grande arquivo - que facilitaria o processo de análise automatizada). Não só aplicações de controle social podem se beneficiar desta iniciativa, mas também o jornalismo de dados e a própria gestão pública (checando e comparando informações de outras organizações ao redor do Brasil) podem ganhar com modelos mais acessíveis de distribuição das informações da LAI. O link para acessar o website da iniciativa é: <https://brasil.io/home/>

O **Colaborados** é uma iniciativa que surgiu para reunir, demonstrar, investigar e monitorar o acesso à informação de dados públicos no Brasil. Possui como principal objetivo agregar bases de bancos de organizações públicas, com o objetivo de que o cidadão possa encontrar em um único lugar uma lista de portais da transparência disponíveis no Brasil. Um dos projetos mais interessantes realizados pela iniciativa é o Colaborobot, que se trata de um robô que fica monitorando todos os portais da transparência presentes no banco de dados do Colaborados, para verificar se eles estão funcionando corretamente. O intuito desse bot é analisar se o portal se encontra operacional para que os cidadãos possam acessar as informações presentes em suas bases de dados, permitindo que a população cobre para que a LAI esteja sendo cumprida, bem como possa questionar os órgãos acerca de eventuais problemas. Ao encontrar algum portal que esteja fora do ar, o bot realiza um tweet informando o órgão responsável e a população sobre a indisponibilidade. O link para acessar o website da iniciativa é: <https://colaborados.com.br/>

4 PROPOSTA DO TRABALHO

Tendo em vista as possibilidades que a sociedade da informação nos oferece, bem como o movimento de transparência nos dados governamentais (principalmente a Operação Serenata do Amor e o movimento Brasil.io), uma das propostas desse trabalho é aproximar o cidadão comum das informações que são disponibilizadas nas bases de dados de portais da transparência, tornando-as mais compreensíveis e mais intuitivas para o usuário. Dessa forma será desenvolvido um aplicativo MVP que permite que o cidadão tenha acesso as compras efetuadas pelas verbas do Fundo Nacional de Desenvolvimento da Educação (FNDE), no âmbito do Programa Nacional de Alimentação Escolar (PNAE), e possa realizar uma avaliação indicando se a compra possui ou não indícios de fraude. Esse recurso vai permitir que o cidadão exerça a democracia participativa, onde haverá um espaço para realizar comentários sobre as avaliações efetuadas, bem como interagir de forma anônima com outros usuários que realizaram a mesma avaliação. Essas avaliações serão de extrema importância para projetos futuros, onde poderão ser aplicados algoritmos de classificação para prever novos indícios de forma automatizada.

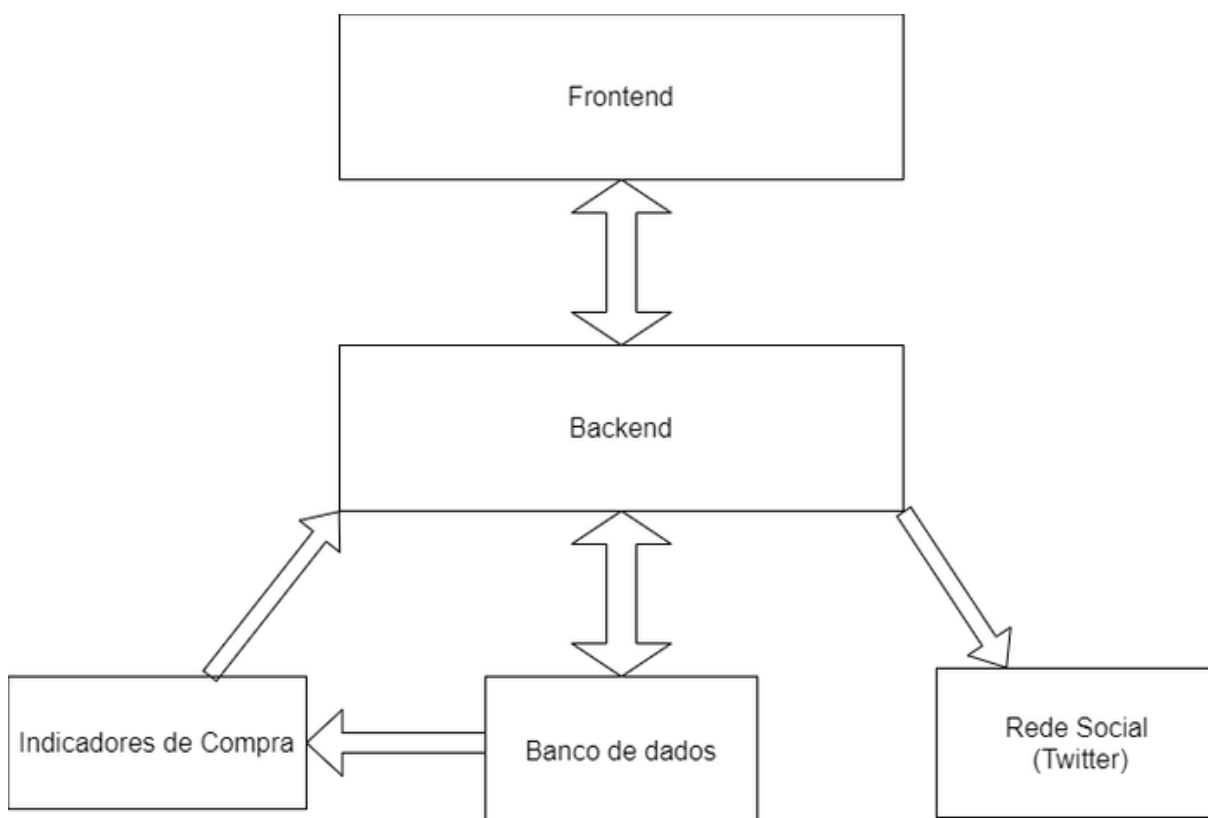
De forma a facilitar a realização da avaliação, o usuário terá acesso aos dados de compras realizadas por municípios vizinhos da compra em análise, bem como uma estimativa do preço esperado utilizando regressão linear e aprendizado de máquina (obtidos em nosso ambiente de *machine learning* construído utilizando o TensorFlow). Essas informações colaborarão na medida em que serão dados concretos de comparação, propiciando algo para que o usuário fuja somente do uso do "*feeling*" na hora de avaliar uma compra.

Em outro cenário de uso de nosso aplicativo, o usuário pode preencher os preços atuais dos produtos alimentícios nos supermercados de sua cidade. Essa feature auxiliará na construção de uma base nacional de preços e será muito importante na realização de futuras análises de dados, podendo ser utilizada até mesmo como informação para auxiliar os usuários na tomada de decisão. Esse cenário de uso mitigará o problema de que hoje não há uma base nacional pública de preços de produtos, que possa ser facilmente acessada e utilizada como comparação para obter um preço esperado por unidade.

Buscando tornar os dados ainda mais transparentes para a população, será desenvolvido também uma integração com o Twitter que informa os indícios de fraude classificados por usuários da plataforma.

Para dar suporte as operações realizadas no aplicativo será desenvolvido uma API REST (também chamada no presente trabalho de aplicação backend) que será central na conexão do aplicativo com as demais partes do ecossistema, conforme está demonstrado na figura 6.

Figura 6 - Resumo do ecossistema do MVP



Fonte: O autor (2020)

O aplicativo poderá ser utilizado tanto em dispositivos Android e iOS após a realização do processo de *build*, mas para esse trabalho será utilizado a interface de desenvolvimento do Ionic, que permite a visualização do estado atual da aplicação através do navegador. O backend da aplicação estará disponível a partir de um servidor web, utilizando Node.js e Express, fornecendo acesso aos endpoints da API através da porta 3333. O banco de dados da aplicação será o MySQL que rodará na porta 3306 da máquina, se comunicando com o backend de forma a prover os dados solicitados pela aplicação.

Para a obtenção de indicadores foram analisados dois cenários propostos na fundamentação teórica do presente trabalho, sendo optado ao final pelo uso do TensorFlow.js. Alguns motivos levaram a essa adoção, como os destacados abaixo:

- Experiência do autor do presente trabalho na linguagem Javascript, facilitando assim a produtividade e o desenvolvimento do projeto.
- Não há necessidade de realizar a instalação de ambientes customizados para executar os testes (ao contrário do Hadoop), o TensorFlow roda simplesmente importando um arquivo Javascript e executando dentro do

navegador.

- Características dos dados. Hadoop iria funcionar melhor se tivesse uma base nacional de preços para realizar a comparação.
- Modelos de execução de machine learning prontos. O TensorFlow.js possui APIs que abstraem a definição de modelos e de tensores, facilitando assim o processo de execução de aprendizagem de máquina.

5 DESENVOLVIMENTO DO TRABALHO

De acordo com a arquitetura e proposta apresentadas no capítulo 4, foi possível começar o desenvolvimento do MVP do Trabalho de Conclusão de Curso, batizado como “Merenda Democrática”.

O desenvolvimento do MVP foi realizado em quatro etapas, sendo elas: aplicação backend (responsável pela API com os *endpoints* dando suporte às *features* do aplicativo, tanto para servir como conexão com os dados obtidos através do Painel de Prestação de Contas do FNDE e com as previsões realizadas pelo TensorFlow), aplicação frontend (o aplicativo para o usuário realizar a avaliação), ambiente TensorFlow.js para gerar os indicadores de preços e o bot para comunicar eventuais fraudes no Twitter.

Conforme explicitado em capítulos anteriores deste trabalho, as escolhas sobre ferramentas de implementação levaram em conta alguns aspectos técnicos e principalmente as experiências acadêmicas e profissionais do autor deste trabalho, aliadas com a disponibilidade de materiais na internet necessários para entender a arquitetura e as peculiaridades das tecnologias, bem como a disponibilidade de uma comunidade ativa em cada uma dessas tecnologias para que possa servir de suporte na resolução de determinados problemas técnicos durante o desenvolvimento do MVP.

As tecnologias utilizadas durante o desenvolvimento deste trabalho englobam vários tópicos estudados durante o período do curso de Sistemas de Informação, como desenvolvimento orientado a objetos, desenvolvimento web, banco de dados relacionais, versionamento de código, boas práticas de engenharia de software, inteligência artificial e estatística sendo um excelente modelo para sintetizar os conhecimentos obtidos através da grade curricular do curso, bem como das experiências profissionais ocorridas dentro do período da universidade.

5.1 BACK-END DA APLICAÇÃO

A aplicação backend tem um papel fundamental no projeto do MVP, já que serve de suporte para as demais partes do sistema e é recurso essencial para o funcionamento das mesmas, já que possui *endpoints* que são utilizados por outros componentes do ecossistema.

A aplicação backend possui *endpoints* que são utilizados tanto para acessar as informações presentes no banco de dados da aplicação, sejam elas avaliações realizadas pelo usuário, ou dados da compra da merenda escolar efetuadas pelos

municípios e obtidos através do Painel de Preços do FNDE (Fundo Nacional de Desenvolvimento da Educação), bem como as previsões realizadas pelo TensorFlow.js. Há também métodos que controlam as postagens realizadas no perfil do Twitter deste trabalho, de forma a servir como uma espécie de “bot” alertando aos seguidores sobre possíveis indícios de fraude na compra da merenda escolar relatadas por usuários que utilizam a aplicação.

Os *endpoints* estão expostos utilizando a estrutura de API REST, padrão de comunicação entre aplicações mais utilizado do mercado, bem como toda a arquitetura de código está desenvolvida com base no Express, *framework* Node.js que facilita a construção de APIs, padronizando assim o escopo da aplicação para que ela seja facilmente estendida em possíveis trabalhos futuros.

5.1.1 Ferramentas utilizadas

Para o desenvolvimento da aplicação, levando em conta aspectos como produtividade e qualidade de código, foram utilizadas algumas ferramentas listadas abaixo. Essas ferramentas foram essenciais para o desenvolvimento do backend do MVP, pois oferecem recursos mínimos que possibilitaram atender aspectos de qualidade e produtividade durante o desenvolvimento do projeto.

- Node: software utilizado como base no desenvolvimento da aplicação. Serve tanto como espécie de kit de desenvolvimento (SDK), como também suporte para rodar as aplicações desenvolvidas.
- Yarn: software utilizado para realizar o gerenciamento dos pacotes das dependências necessárias para a execução e para o desenvolvimento da aplicação.
- Code: IDE utilizada para implementação do código da API.
- brModelo: software utilizado para realizar a modelagem do banco de dados conceitual e lógico.
- Insomnia: software que possibilita a execução e testes dos endpoints da API.
- Git: ferramenta de versionamento de código.

Todas as ferramentas são Open Source e podem ser baixadas de maneira gratuita dentro dos seus respectivos sites oficiais. Por acreditar na filosofia de aplicações de código aberto e gratuitas, e na distribuição de tecnologia como forma de contribuir de forma efetiva no desenvolvimento das sociedades, o autor deste trabalho deu preferência para o uso de ferramentas e bibliotecas que acreditam nessa mesma filosofia.

5.1.2 Tecnologias utilizadas

A API REST do MVP Merenda Democrática foi desenvolvida utilizando as seguintes tecnologias: a linguagem de programação Javascript, o seu ambiente de execução de código server-side Node.js e seu *framework* para desenvolvimento de aplicações backend Express, além do banco de dados MySQL.

Alguns fatores levaram a escolha dessas tecnologias para o desenvolvimento desse trabalho, além do domínio por parte do autor. Algumas das razões estão explicitadas abaixo.

- O Node.js possui um gerenciador de pacotes chamado NPM (com equivalência no YARN) que é o maior repositório de softwares no mundo. Isso traz uma flexibilidade gigantesca, com um potencial para ser utilizar essa tecnologia em praticamente qualquer situação de uma maneira simples e efetiva.
- O Node.js é operacionalmente bastante flexível e não precisa de muitos recursos computacionais para poder executar. Em conjunto com aplicações como Docker o cenário é ainda melhor, sendo possível realizar deploy e escalonamento de máquinas de uma forma muito rápida e simples.
- O Node.js possui o *framework* Express, uma das melhores tecnologias quanto a arquitetura de desenvolvimento de APIs, simplificando o design e a codificação das aplicações.
- O Node.js oferece uma alta produtividade por além de possuir o maior repositório de pacotes do mundo, rodar a partir da linguagem de programação Javascript, permitindo assim que seja utilizado uma mesma linguagem tanto no frontend quanto no backend da aplicação.
- O banco de dados MySQL é um dos mais utilizados do mercado, é bastante leve e possui algumas IDEs excelentes que facilitam o seu gerenciamento.

A aplicação back-end consiste na disponibilização de uma API para que o frontend tenha acesso aos recursos necessários ao seu funcionamento. Entre esses recursos estão o acesso ao banco de dados com as informações consolidadas durante o trabalho, bem como aos demais features desenvolvidas para facilitar o acesso das informações por parte dos cidadãos.

5.1.3 Implementação

O desenvolvimento da API foi dividida nas seguintes etapas:

1. Configuração do ambiente.
2. Modelagem do banco de dados.
3. Desenvolvimento da API.
4. Criação dos *endpoints* de teste.
5. Extração dos dados do Painel de Preços praticados pelo PNAE.
6. Tratamento e transformação dos dados.
7. Importação dos dados no banco de dados.

A etapa de configuração do ambiente consiste na instalação das ferramentas necessárias para o desenvolvimento das aplicações listadas anteriormente neste projeto na seção 5.1.1. Após a instalação é recomendado adicionar alguns plugins no Visual Studio Code para poder manter o padrão de código utilizado: sendo eles o ESLint e o Prettier. Os dois estão presentes no gerenciador de pacotes da IDE de forma gratuita e não necessitam nenhuma configuração adicional, já que estão previamente configurados dentro do código da aplicação. É necessário também realizar as instalações dos pacotes descritos no arquivo "package.json". Para realizar a instalação basta rodar o comando "*yarn add*" ou "*npm install*" dependendo do gerenciador de pacotes que está sendo utilizado.

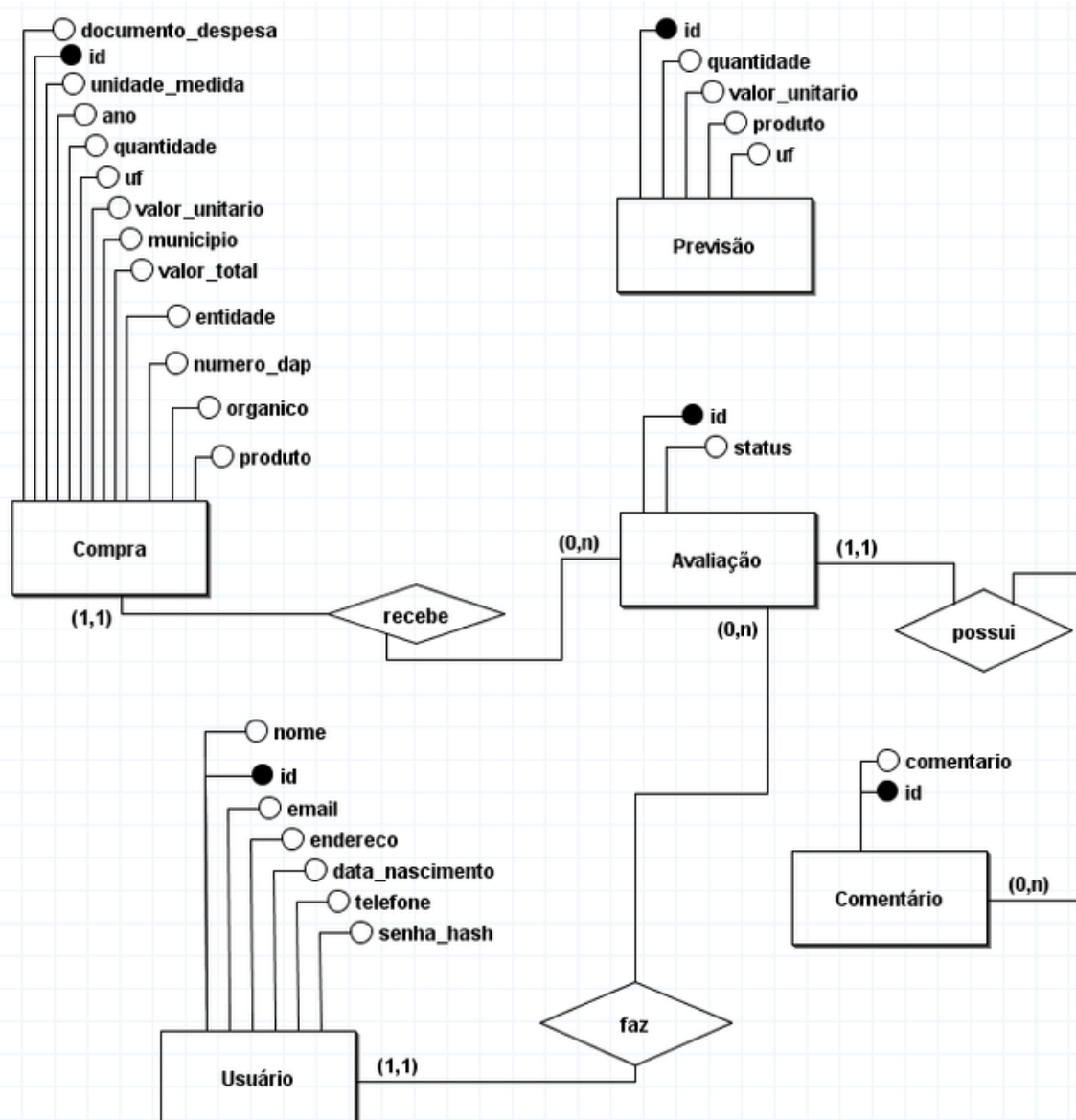
Tendo em vista que o foco do trabalho está na implementação, configurações adicionais serão mencionadas durante a explicação do mesmo, caso seja necessário. Abaixo será especificado de forma simplificada a construção do modelo conceitual-lógico, bem como foram estruturados e desenvolvidos os *endpoints* da API.

5.1.4 Modelagem de banco de dados

A modelagem do banco de dados foi construída a partir dos estudos realizados durante a etapa de fundamentação teórica, do conhecimento da estrutura dos dados das compras disponíveis no Painel de Preços do PNAE e da definição das features na aplicação. O mapeamento das entidades aconteceu após o processo de definição dos objetivos do trabalho.

As figuras 7 e 8 apresentam o modelo conceitual elaborado utilizando o software "brModelo". Para poder melhor visualizar os detalhes e aproveitando a estrutura do modelo, as entidades foram divididas em duas imagens.

Figura 7 - Modelo Conceitual - Avaliação



Fonte: O autor (2020)

Ao construir o modelo conceitual com a ferramenta "brModelo", foram mapeadas sete entidades que compõem a nossa aplicação, sendo elas: Usuário, Comentário, Avaliação, Compra, Previsão, Preço e Arquivo. Também foi possível mapear seus atributos e seus relacionamentos.

Compra é a entidade responsável por representar as compras que são obtidas através dos dados disponíveis no Painel de Preços do FNDE. Possui relação com a entidade Avaliação, na medida que uma compra recebe 0 ou mais avaliações e uma avaliação possui uma compra obrigatoriamente e pode possuir no máximo

uma compra. De acordo com a estrutura dos dados disponíveis no Painel de Preços, foi possível realizar o mapeamento dos seguintes atributos:

- Ano: valor que representa o ano em que a compra foi realizada.
- UF: representa a unidade federativa (estado) da compra.
- Município: representa o município que a compra foi realizada.
- Entidade: representa a entidade que efetuou o empenho de compra. Os valores comuns para esse campo são prefeituras municipais, secretarias estaduais de educação, secretarias municipais de educação, etc.
- Número DAP: representa o número da compra realizada através dos produtores locais, através da agricultura familiar.
- Orgânico: indica se o produto é orgânico ou não.
- Produto: indica o nome do produto comprado.
- Documento da Despesa: possui o número do documento da compra na base de dados das prefeituras municipais (nem todas as compras possuem esse documento cadastrado e não há informações específicas para obter essas notas fiscais para conferência).
- Unidade de medida: unidade de medida do produto utilizada na descrição da compra, como quilo, grama, litro, etc.
- Quantidade: quantidade de produtos adquiridas na compra.
- Valor unitário: valor de 1 unidade de produto da compra.
- Valor total: valor de todas as unidades de produto da compra.

A entidade Avaliação representa as avaliações realizadas pelos usuários no aplicativo frontend, e possui relacionamento tanto com a entidade Usuário quanto com a entidade Compra. Uma compra pode possuir 0 ou N avaliações enquanto uma avaliação precisa ter necessariamente uma compra e só pode ter no máximo uma compra. Já um usuário também pode ter 0 ou N avaliações, enquanto uma avaliação necessita de um usuário, e pode ter no máximo um usuário também. Outro atributo adicional da avaliação é um campo de status, de valor booleano (verdadeiro ou falso), indicando se a compra avaliada possui indícios de fraude ou não.

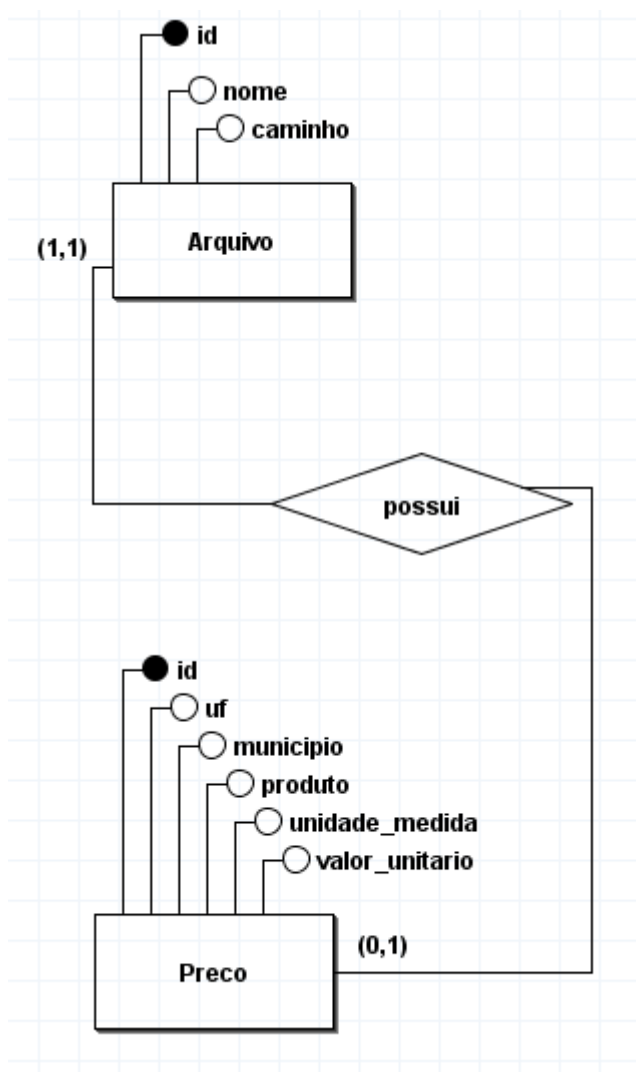
A entidade Usuário representa os usuários que estão cadastrados no sistema e podem assim realizar avaliações. A entidade Usuário se relaciona com a entidade Avaliação, conforme explicitado no parágrafo acima. Possui outros atributos adicionais como nome, email, endereço, data de nascimento, telefone e um campo "senha_hash", que armazena a senha do usuário criptografada.

A entidade Comentário representa os comentários que os usuários podem deixar nas avaliações que realizaram. Comentários se relacionam com avaliações,

na medida que uma avaliação pode ter 0 ou N comentários, enquanto um comentário possui obrigatoriamente uma avaliação e pode ter no máximo 1 avaliação também.

A entidade Previsão representa os resultados que obtivemos durante o nosso processo de obtenção dos indicadores de preço através da utilização da rede neural construída pelo TensorFlow.js. Essa entidade não possui relacionamento com nenhuma outra do sistema. Para possibilitar que sejam realizadas buscas nela, ela possui atributos produto e UF, identificando o produto e a unidade federativa (estado) à que a previsão se refere. Os campos quantidade e valor unitário indicam os valores numéricos das previsões realizadas pela ferramenta.

Figura 8 - Modelo Conceitual - Preço



Fonte: O autor (2020)

A entidade Preço representa os preços cadastrados pelos usuários. Ela

possui relacionamento com outra entidade chamada Arquivo, na medida em que os usuários poderão enviar uma imagem da nota fiscal ou do preço na gôndola comprovando as informações do preenchimento. Essa separação em duas entidades é utilizada de forma que seja possível aproveitar a entidade Arquivo em outras partes do sistema, como por exemplo, arquivos relativos às compras já efetuadas ou imagem de avatar de perfil do usuário na aplicação por exemplo. Alguns outros atributos formam a entidade preço, como:

- UF: representa a unidade federativa (estado) que o usuário se encontra.
- Município: representa o município do usuário.
- Produto: representa o produto do qual o usuário está cadastrando o preço.
- Unidade de Medida: representa a unidade de medida do produto comprado (quilo, litro, etc).
- Valor Unitário: representa o valor unitário do produto.

Por fim, a entidade Arquivo possui além do relacionamento com a entidade Preço, dois atributos próprios: nome da imagem (utilizado para identificar o arquivo) e caminho do arquivo (utilizado para recuperar o seu conteúdo em consultas futuras).

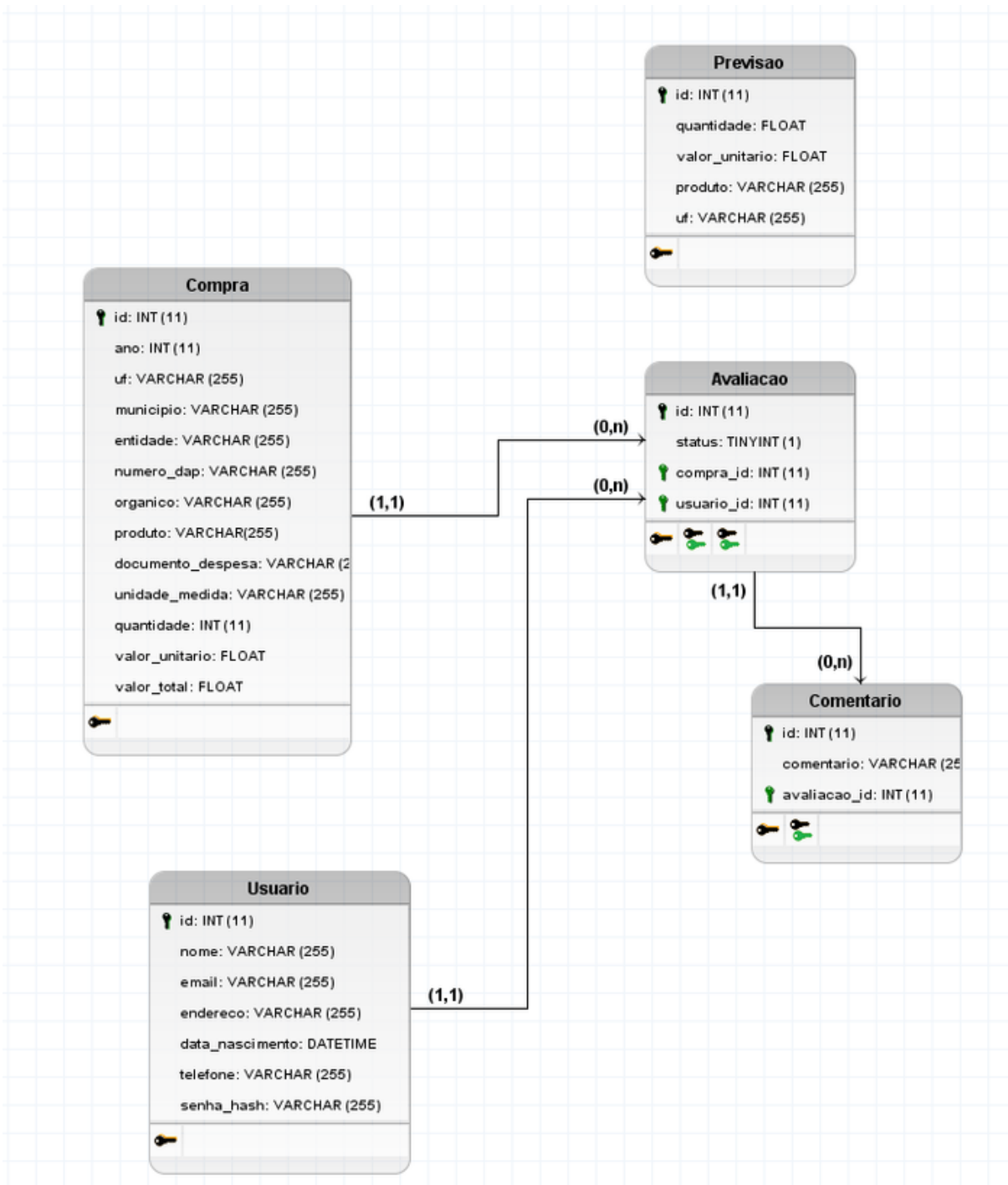
5.1.4.1 Construção do modelo lógico

Para a construção do modelo lógico foi utilizado como base o modelo conceitual construído na seção anterior. O software utilizado para o desenvolvimento da modelagem do banco de dados, chamado "brModelo", possui uma ferramenta que facilmente permite essa transformação. Nessa próxima etapa, algumas mudanças aconteceram como por exemplo: as entidades deixam de ser apenas conceituais e se tornam tabelas, os atributos tornam-se campos tipados, os identificadores tornam-se chaves privadas e são adicionados também chaves estrangeiras nas ligações com a cardinalidade com maior dependência.

Nessa etapa tivemos também que definir os tipos dos campos das tabelas. A partir dessa definição e consequente finalização do modelo lógico, foi possível realizar a conversão para os modelos do Sequelize (ORM utilizado para realizar o mapeamento do banco de dados da aplicação e abstrair alguns recursos fundamentais, como por exemplo, a escrita de consultas para manipulações básicas de dados - conhecidas como CRUD) e criar as *migrations* de criação das tabelas, com um simples mapeamento das informações do modelo lógico em tipagem e

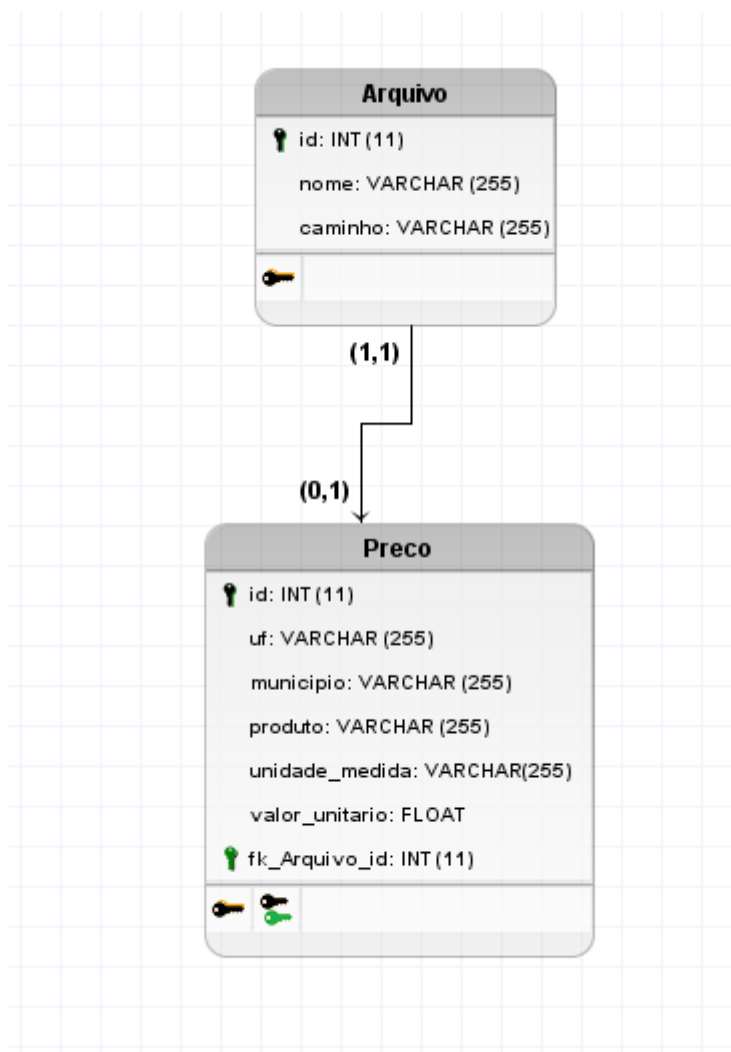
definição SQL respectivamente. As figuras 9 e 10 apresentam o resultado da definição de nosso modelo lógico.

Figura 9 - Modelo Lógico - Avaliação



Fonte: O autor (2020)

Figura 10 - Modelo Lógico - Arquivo



Fonte: O autor (2020)

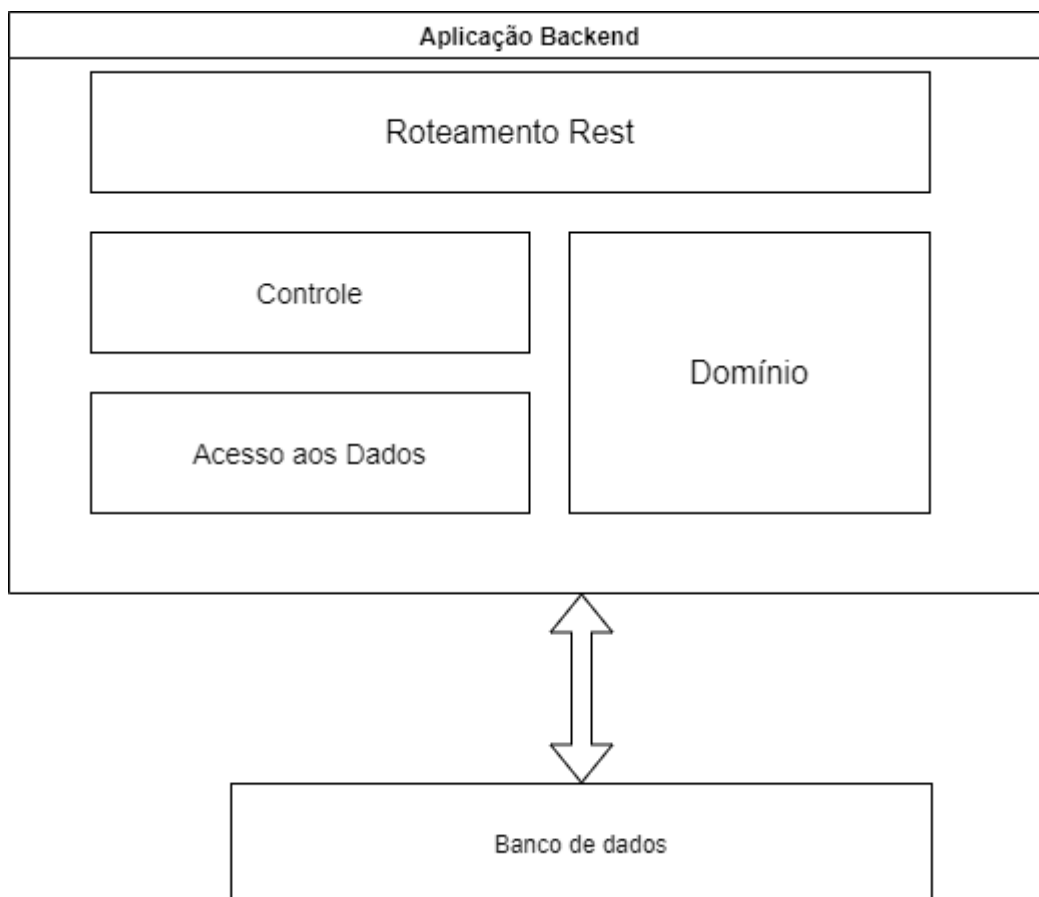
5.1.5 API REST Merenda Democrática

A API Merenda Democrática foi construída com a IDE Visual Studio Code, *plugins* para padronização de código-fonte ESLint e Prettier, bem como com a linguagem de programação Javascript. Futuros trabalhos relacionados a esta API demandarão o conhecimento prévio nas tecnologias citadas na seção 5.1.2 ou tecnologias que possuem alguma equivalência. Por se tratarem de tecnologias conceituadas no mercado, há uma extensa documentação, cursos, tutoriais e dúvidas resolvidas, facilitando assim o processo de conhecimento dos requisitos mais técnicos do trabalho.

Para estruturar a aplicação e facilitar aspectos como visibilidade do código e manutenibilidade, aspectos importantes de Engenharia de Software, optou-se por

um desenvolvimento realizado em camadas de forma a isolar e separar as responsabilidades de cada uma. A figura 11 demonstra um esquema mais completo com as camadas inerentes a API do MVP.

Figura 11 - Camadas da aplicação backend



Fonte: O autor (2020)

A API do MVP Merenda Democrática está dividida em quatro camadas, sendo elas: a camada que chamaremos de “roteamento REST”, a camada de controle, a camada de acesso aos dados e a camada de domínio. Essas camadas desempenham funções específicas e possuem responsabilidades inerentes a essas mesmas funções. Abaixo há uma explicação mais detalhada sobre as responsabilidades e sobre as características de cada uma das camadas da aplicação.

- Roteamento REST é a camada de comunicação que permite aplicações clientes requisitarem o acesso às funcionalidades da API e receberem as respostas desejadas. Nessa camada são estabelecidas definições técnicas

como as rotas que a aplicação possui, as mensagens de comunicação que são trocadas entre as aplicações, bem como tipo de requisição HTTP que cada rota possuirá (por exemplo GET, POST, PUT e DELETE).

- A camada de controle está logo abaixo da camada de roteamento da aplicação. São os controladores (em tradução direta de *controllers*), que irão ser executados após a camada de roteamento realizar a sua função. Essa camada possui como objetivo intermediar a comunicação entre a camada de serviços e a camada de roteamento REST, recebendo as requisições e chamando os respectivos métodos necessários para que seja enviado uma resposta para o usuário de acordo com aquilo que foi solicitado. Essa camada pode possuir alguns métodos auxiliares que realizarão a transformação de dados.
- A camada de acesso aos dados comunica-se com o banco de dados da aplicação, sendo a única camada do software que possui essa prerrogativa. Ela comumente é construída através das classes que implementam o service, que são responsáveis por invocar as classes da camada de domínio que por sua vez estendem o pacote Sequelize. O pacote Sequelize possui alguns utilitários de acesso como o *findAll*, *findOne*, *find* e *count*, bem como a estrutura para realizar consultas personalizadas com o uso de *where*, *order*, *limit* e outros recursos comuns às consultas em banco de dados. Para simplificar o desenvolvimento da aplicação, essa camada será implementada junto aos controles da aplicação, já que todos os serviços que desenvolvemos possuem uma lógica relativamente simples.
- A camada de domínio reflete o modelo lógico do banco de dados da aplicação, ou seja, nessa camada há equivalências para as tabelas do banco de dados. Essa camada possui uma característica peculiar de ser conhecida por todas as outras camadas, já que os seus objetos de domínio e suas propriedades são utilizadas em todas as partes da aplicação.

Foram utilizados ainda dois conceitos bastante importantes para produzir códigos de qualidade. O primeiro conceito é o de criação de *migrations*. Esse recurso funciona como uma espécie de versionamento do banco de dados, onde permite manter acesso a todas as mudanças na estrutura do banco, sejam elas criação de novas tabelas, edição de campos e tabelas já existentes, ou até mesmo manipulação de dados. O recurso de *migrations* ainda oferece alguns comandos utilitários para realizar as operações básicas com todas as migrações desenvolvidas, como rodar um lote de mudanças, rodar algumas migrações específicas e desfazer alguma migração. Esse recurso é importante na medida que as aplicações mudam o

seu modelo lógico no decorrer do desenvolvimento (inclusive a própria aplicação do presente trabalho) e conforme essas mudanças ocorrem é muito complicado manter a compatibilidade entre os bancos de dados dos desenvolvedores ou até mesmo de múltiplas instâncias de banco de dados que possam em cenário hipotético, servir a aplicação. Se fossemos contextualizar as migrações dentro das camadas da nossa aplicação, elas estariam na camada de acesso aos dados. Essas migrações, no entanto, não englobam o povoamento da tabela de compras com os dados oriundos do Painel de Compras do FNDE, processo descrito no Apêndice B do presente trabalho.

O segundo conceito bastante importante para elevar o padrão de qualidade da aplicação são os *middlewares*. Os *middlewares* são códigos que interceptam as requisições na camada de roteamento REST e executam algumas ações específicas, notadamente verificar se o usuário que está solicitando possui acesso ao recurso desejado, ou verificar se o recurso desejado existe. No nosso trabalho estamos utilizando o conceito de *middleware* nesses dois cenários: para bloquear o acesso de alguns *endpoints* da nossa aplicação para usuários que não estão autenticados, bem como verificar se alguns recursos solicitados existem antes de chegar na camada de controle.

Com as responsabilidades e as particularidades compreendidas, as próximas seções apresentarão mais detalhes sobre a implementação de cada uma das camadas para um dos objetos de domínio da nossa aplicação.

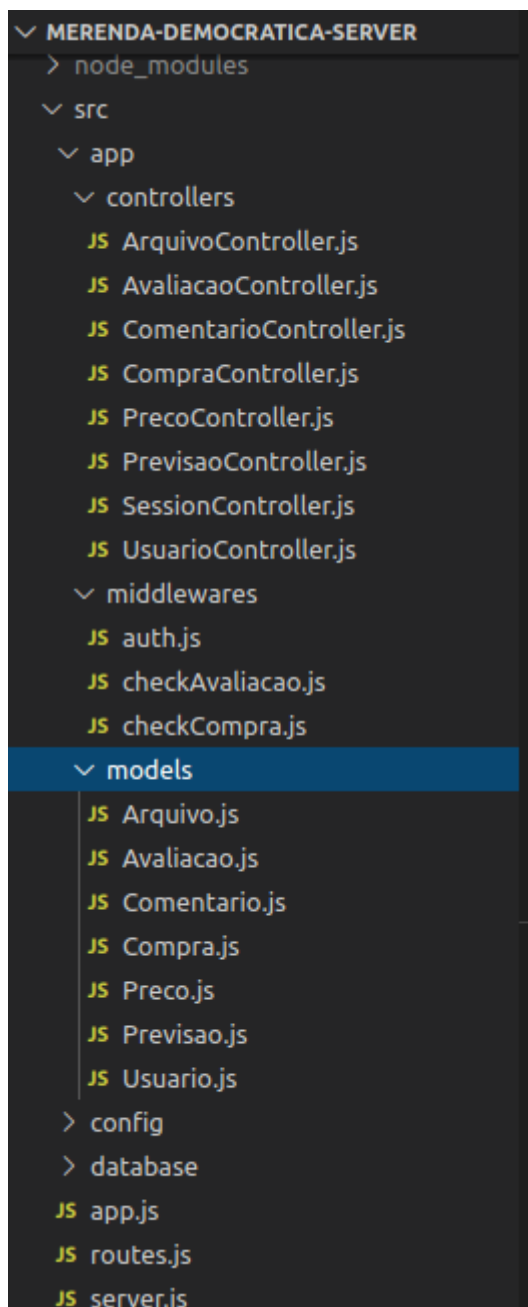
5.1.5.1 Visão Geral

A figura 12 ilustra os arquivos da API dentro das suas respectivas pastas. Nessa imagem estão ilustrados todas as camadas da aplicação explicadas anteriormente, como “*controllers*” que se referem a camada de controle, “*services*” que se referem a camada de acesso aos dados, “*models*” à camada de domínio, bem como o arquivo *routes.js*. Esse arquivo implementa toda a primeira camada de roteamento da aplicação. Na imagem também vemos destacados os “*middlewares*” e a pasta “*database*” com as migrações mantendo o histórico do banco de dados da aplicação.

Nessa figura há denominadores comuns nos nomes dos arquivos. Cada denominador comum é relativo a uma das entidades previstas no modelo lógico, ou seja, “Arquivo”, “Avaliação”, “Preço”, “Compra”, “Previsão”, “Comentário” e “Usuário”. Entretanto todas elas possuem estruturas de programação bastante similares, de forma que para evitar explicações muito parecidas será relatado todas as classes

que são relativas ao domínio “Compra”, por serem eventualmente mais elaboradas e relacionadas com os dados presentes no Painel de Preços praticados pelo FNDE (Fundo Nacional de Desenvolvimento da Educação).

Figura 12 - Diretórios da API REST



Fonte: O autor (2020)

5.1.5.2 Camada de domínio

Conforme explicitado em outras seções do texto, a camada de domínio reflete

as tabelas do banco de dados da API, dessa forma as classes presentes na camada de domínio também estão presentes no banco de dados físico.

Para demonstração foi escolhido um trecho de código da classe “Compra”. Nessa classe são definidos os campos presentes no domínio, que refletem cada um, uma coluna do banco de dados. Javascript implementa também automaticamente uma série de métodos que facilitam o acesso e manipulação dos dados (conhecidos como métodos *getters* e *setters*). A figura 13 apresenta o código de implementação da classe Compra.

Figura 13 - Classe Compra

```
src > app > models > JS Compras > Compra > init
1
2 import Sequelize, { Model } from 'sequelize';
3
4 class Compra extends Model {
5
6   static init(sequelize) {
7     super.init(
8       {
9         ano: Sequelize.NUMBER,
10        uf: Sequelize.STRING(2),
11        municipio: Sequelize.STRING,
12        entidade: Sequelize.STRING,
13        numero_dap: Sequelize.STRING,
14        organico: Sequelize.STRING(1),
15        produto: Sequelize.STRING,
16        documento_despesa: Sequelize.STRING,
17        unidade_medida: Sequelize.STRING,
18        quantidade: Sequelize.FLOAT,
19        valor_unitario: Sequelize.FLOAT,
20        valor_total: Sequelize.FLOAT,
21      },
22      {
23        sequelize,
24      }
25    );
26
27    return this;
28  }
29 }
30
31 export default Compra;
32
```

Fonte: O autor (2020)

Para ganhar produtividade no desenvolvimento é utilizado o pacote Sequelize, que é responsável por gerenciar conexões com múltiplos tipos de banco de dados (MySQL no nosso caso), implementando uma interface única de codificação. O

Sequelize utiliza o conceito de "*promises*" ao realizar as operações do banco de dados. Possui suporte a algumas características fundamentais e clássicas de um banco de dados, como definição de relacionamentos entre as tabelas, suporte a transações, carregamento com "*lazy loading*" e replicações, entre outras. Para o nosso trabalho utilizaremos as características básicas do Sequelize, como definição de modelos para refletir as entidades da camada de domínio, definição de relacionamentos, suporte à migrações e manipulação dos dados utilizando os métodos auxiliares, tanto para escrita na base de dados quanto para leitura através de funções utilitárias disponibilizadas pelo Sequelize.

As demais classes da camada de domínio estão concentradas dentro da pasta "*models*" mostrada na seção acima e possuem o mesmo padrão de estrutura da classe *Compra*.

5.1.5.3 Camada de controle e camada de acesso a dados

Após explanarmos mais detalhes da camada de domínio é importante explicarmos como funciona a camada de controle e a camada de acesso aos dados. De forma geral resolvemos implementar as duas no mesmo arquivo devido à complexidade dos métodos no geral, que não justificam uma separação na arquitetura basicamente por preciosismo.

A primeira das duas camadas que comentaremos se trata da camada de acesso aos dados. Essa camada é responsável pelas operações de manipulação (inserção, atualização, seleção e remoção) dos dados. Modificações de estrutura na base de dados, como criação e remoção de tabelas, bem como atualização das duas colunas estão sendo definidas em um arquivo especial conhecido como "*migration*".

A utilização do pacote Sequelize permite que as operações sejam simplificadas, já que não precisamos realizar a criação de todas as instruções SQL e nem tratar o retorno de uma conexão nativa com o banco de dados. Delegando ainda essa responsabilidade ao Sequelize podemos abstrair qual sistema gerenciador de banco de dados (SGBD) utilizaremos, já que o Sequelize cria uma camada que abstrai alguns aspectos-chaves como a implementação da linguagem SQL que os bancos realizam, bem como especificidades em sua construção. Isso permite entre outras coisas, que possamos realizar rapidamente a troca de um banco de dados MySQL para um banco de dados PostgreSQL por exemplo. Nessa camada de acesso aos dados os recursos que iremos mais utilizar são os utilitários que permitem realizar as consultas utilizando as classes dos objetos de domínio.

A segunda camada que trataremos nesta seção é a camada de controle da

aplicação. Essa camada é responsável por tratar das requisições que vieram da camada de roteamento REST, e retornar uma resposta para o usuário de maneira adequada. Normalmente a camada de controle engloba diversas chamadas a serviços e funções utilitárias que realizam o tratamento dos dados. É uma boa prática manter nessa camada somente essas chamadas, como uma espécie de registro do fluxo de execução do método, mas devido a relativa simplicidade dos métodos da classe e do MVP iremos na própria camada de controle realizar todas as operações necessárias para dar uma resposta adequada ao usuário.

Na figura 14 está representado a definição da classe "CompraController", responsável por implementar tanto a camada de controle quanto a camada de acesso aos dados (serviço) da nossa aplicação:

Figura 14 - Declaração da classe de compra

```
import Sequelize, { Op } from 'sequelize';
import * as Yup from 'yup';
import axios from 'axios';

import Compra from '../models/Compra';

class CompraController {
```

Fonte: O autor (2020)

Nesse trecho de código estamos importando os pacotes e as classes que utilizaremos durante a execução dos métodos da aplicação, como Sequelize e Op (relativo aos operadores de ordenação e comparação do Sequelize), bem como o Yup (responsável por verificar o conteúdo das nossas requisições) e emitir um erro quando a requisição está com algum conteúdo problemático, como faltando parâmetros necessários ou com um parâmetro com tipo diferente do que era esperado. Também estamos importando o axios, pacote responsável por fazer requisições para outras APIs (nesse cenário utilizaremos para se comunicar com a API de Localidades do IBGE - descrita na fundamentação teórica) e importando a classe Compra, responsável por definir o domínio de classe. Por fim estamos realizando a declaração da classe.

Figura 15 - Método "index"

```
async index(req, res) {  
  const limit = req.query.limit ? Number(req.query.limit) : 100;  
  const page = req.query.page ? Number(req.query.page) : 1;  
  
  const orders = await Compra.findAll({  
    limit: limit,  
    offset: (page - 1) * limit,  
    attributes: ['id', 'ano', 'uf', 'municipio', 'entidade', 'numero_dap', 'organico',  
                | 'produto', 'documento_despesa', 'unidade_medida', 'quantidade', 'valor_unitario', 'valor_total'],  
  });  
  
  return res.json(orders);  
}
```

Fonte: O autor (2020)

O método `index` (presente na figura 15) é responsável por fazer a listagem de todas as compras, realizando uma paginação. Vale citar alguns detalhes importantes neste código, como por exemplo, o uso do utilitário do Sequelize `findAll`, que é responsável por procurar todos os registros da base de dados. Estamos passando também alguns parâmetros para realizar essa consulta como o parâmetro `limit` que define o número de registros que a consulta retornará e o parâmetro `offset` que definirá a partir de que número de registro começam os nossos resultados. É realizado um tratamento para definir o valor desses dois parâmetros e caso a requisição não os define, é estabelecido 100 para a quantidade de registros retornados e 1 para que sejam retornados os primeiros resultados da consulta. O campo `attribute` define quais atributos do registro irão ser retornados em nossa consulta. Por fim todos os registros são retornados para a aplicação cliente em formato JSON através da chamada do método `res.json`.

Figura 16 - Método "indexByMunicipio"

```

async indexByMunicipio(req, res) {
  const limit = req.query.limit ? Number(req.query.limit) : 100;
  const page = req.query.page ? Number(req.query.page) : 1;

  const schema = Yup.object().shape({
    municipio: Yup.string().required(),
  });

  if (!(await schema.isValid(req.body))) {
    return res.status(400).json({ error: 'Erro na validação.' });
  }

  const municipio = String(req.body.municipio).toUpperCase();
  const Op = Sequelize.Op;

  const orders = await Compra.findAll({
    limit: limit,
    where: {
      municipio: {
        [Op.like]: `%${municipio}%`
      }
    },
    offset: (page - 1) * limit,
    attributes: ['id', 'ano', 'uf', 'municipio', 'entidade', 'numero_dap', 'organico', 'produto',
      'documento_despesa', 'unidade_medida', 'quantidade', 'valor_unitario', 'valor_total'],
  });

  return res.json(orders);
}

```

Fonte: O autor (2020)

O método "indexByMunicipio" (presente na figura 16) é responsável por listar todas as compras de um determinado município passado para a aplicação como parâmetro no corpo da requisição. Além das características listadas no método "index" acima, temos algumas particularidades muito interessantes. Nesse método estamos utilizando o Yup para poder validar as características do campo da requisição "município". Caso a requisição não atenda todos os critérios é lançado uma exceção e disparado uma mensagem de erro. Também é utilizado o utilitário *where*, que permite retornar só os registros que atendam determinada condição, como nesse caso que tenham o nome do município conforme o conteúdo do corpo da requisição. É retornado uma resposta para a aplicação cliente em formato JSON.

Figura 17 - Método "random"

```

async random(req, res) {
  const order = await Compra.findOne({
    order: [
      Sequelize.fn('RAND'),
    ],
    attributes: ['id', 'ano', 'uf', 'municipio', 'entidade', 'numero_dap', 'organico', 'produto',
      'documento_despesa', 'unidade_medida', 'quantidade', 'valor_unitario', 'valor_total'],
  });

  return res.json(order);
}

```

Fonte: O autor (2020)

O método *random* (presente na figura 17) é responsável por retornar uma única compra (valor definido no uso do utilitário *findOne*) de forma aleatória (definido no parâmetro *order*). No mais, o método segue o mesmo padrão apresentado pelos métodos anteriores. No final da execução do código ele retorna os dados da compra em formato JSON.

Figura 18 - Método "indexMunicipiosProximos"

```

async indexMunicipiosProximos(req, res) {

  const compra_id = req.params.compraId;

  const order = await Compra.findByPk(compra_id);

  const municipiosUF = await axios.get(
    `https://servicodados.ibge.gov.br/api/v1/localidades/estados/${order.uf}/municipios`);

  const municipioCompra = municipiosUF.data.filter(item => String(item.nome)
    .toLowerCase() === String(order.municipio)
    .toLowerCase())[0];

  const microrregiaoId = municipioCompra ? municipioCompra.microrregiao.id : -1;

  const municipiosMicrorregiao = await axios.get(
    `https://servicodados.ibge.gov.br/api/v1/localidades/microrregioes/${microrregiaoId}/municipios`);

  const municipiosArray =
    municipiosMicrorregiao.data.map(item => String(item.nome)
      .toUpperCase())
      .filter(item => item !== order.municipio);

  const compras = await Compra.findAll({
    where: {
      municipio: {
        [Op.in]: municipiosArray
      },
      produto: order.produto,
      unidade_medida: order.unidade_medida,
    },
    limit: 15,
    order: [
      Sequelize.fn('RAND'),
    ],
  });

  return res.json(compras);
}

```

Fonte: O autor (2020)

Por fim, o método mais elaborado da classe, "*indexMunicipiosProximos*", apresentado na figura 18. Esse método é responsável por retornar compras dos municípios vizinhos do mesmo produto e da mesma unidade de medida. Ele recebe por parâmetro na URL da requisição o identificador da compra desejada, realiza na base de dados a busca pela compra informada, e na API do IBGE a busca por todos os municípios do estado da compra. Após realizar essa busca é feito um "*match*" entre o município da compra e o objeto de município vindo da API do IBGE. Com o

objeto identificado é filtrado a microrregião a que esse município pertence e realizado outra consulta a API do IBGE para retornar todos os municípios daquela microrregião. Após isso é tratado o objeto de municípios da microrregião de forma a se tornar uma string com os nomes de todos os municípios, excluindo o município da compra. Por fim é realizado uma consulta passando na cláusula *where* uma condição de que a compra dos municípios deveriam ter o mesmo produto da compra original, a mesma unidade de medida da compra original e o município definido no array passado para a cláusula *in*. O resultado da consulta é retornado via JSON para a aplicação cliente.

Dessa forma cobrimos as peculiaridades da classe Compra nas camadas de controle e de acesso aos dados. Por fim, a última camada do backend da API Merenda Democrática a ser tratada na próxima seção é a camada de roteamento REST.

5.1.5.4 Camada de roteamento REST

A camada de roteamento REST é responsável por implementar uma interface de entrada aos recursos (*endpoints*) da nossa aplicação. Os *endpoints* acessam controladores da camada de controle de forma a tratar as requisições do sistema e acessar recursos como os dados por exemplo.

Para implementar o código da camada de roteamento de forma mais eficiente foi utilizado as features previstas pelo *framework* Express. Na figura 19 está representado o código base da camada de roteamento presente no arquivo `routes.js`:

Figura 19 - Roteamento REST - Declaração

```
import { Router } from 'express';
import multer from 'multer';
import multerConfig from './config/multer';

import UsuarioController from './app/controllers/UsuarioController';
import ArquivoController from './app/controllers/ArquivoController';
import SessionController from './app/controllers/SessionController';
import AvaliacaoController from './app/controllers/AvaliacaoController';
import ComentarioController from './app/controllers/ComentarioController';
import CompraController from './app/controllers/CompraController';
import CompraController from './app/controllers/CompraController';
import PrecoController from './app/controllers/PrecoController';
import PrevisaoController from './app/controllers/PrevisaoController';

import authMiddleware from './app/middlewares/auth';
import checkAvaliacao from './app/middlewares/checkAvaliacao';
import checkCompra from './app/middlewares/checkCompra';

const routes = new Router();
const upload = multer(multerConfig);

routes.post('/usuarios', UsuarioController.store);

routes.post('/sessions', SessionController.store);

routes.get('/', (req, res) => res.send('ok'));

routes.use(authMiddleware);
routes.put('/usuarios', UsuarioController.update);
```

Fonte: O autor (2020)

No código acima é realizado a importação da classe Router do *framework* Express, que é responsável por criar uma entidade de roteamento dentro dos sistemas, de forma a encaminhar cada *endpoint* para o seu determinado método na camada de controle. É importado também o pacote multer que é responsável por gerenciar o envio de arquivos, *feature* que nossa aplicação frontend utiliza para realizar o cadastro de novos preços de produtos. Após a importação dos pacotes essenciais é realizado a importação dos controles para realizar o mapeamento dos *endpoints* para os seus respectivos métodos.

É também importado os *middlewares*, que conforme descrito em seções anteriores atuarão como uma espécie de interceptadores para realizar algumas verificações como, por exemplo, se o usuário tem acesso aquele determinado recurso ou se a compra e a avaliação desejada existem na base de dados. Com a criação da variável routes é possível começar a realizar a criação dos *endpoints* e o

seu mapeamento. No código acima estão sendo mapeados métodos POST e PUT para os métodos do controlador de usuários, bem como post para o método do controlador de sessão. Também é adicionado um *endpoint* “raiz” para testar se a aplicação está em funcionamento.

Figura 20 - Roteamento - Compra

```
routes.get('/compras', CompraController.index);  
routes.get('/compras-municipio', CompraController.indexByMunicipio);  
routes.get('/compras/random', CompraController.random);  
routes.get('/compras/:compraId/municipios-proximos',  
checkCompra,  
CompraController.indexMunicipiosProximos);
```

Fonte: O autor (2020)

O código presente na figura 20 define os *endpoints* para o controlador de compras, mesmos métodos listados na seção 5.1.5.4. Por se tratar de uma aplicação onde é realizado basicamente a avaliação das compras das merendas escolares, que possuem como procedência os dados cadastradas no Painel de Preços do FNDE não é necessário expor nenhuma rota de cadastro de compra, já que a aplicação frontend não possuirá essa funcionalidade. Por isso todos os *endpoints* implementam o método GET do HTTP, já que são na prática métodos de consulta a base de dados já existente. Outras características importantes de mencionar é o uso da anotação `:compraId` na URL do *endpoint* (essa anotação indica um parâmetro - nesse caso o identificador da compra) e o uso do *middleware* `checkCompra` com o objetivo de verificar se a compra passada por parâmetro na URL existe no sistema, emitindo um erro caso ele não encontre a compra especificada.

Figura 21 - Roteamento - Exportação

```
export default routes;
```

Fonte: O autor (2020)

Conforme demonstrado na figura 21, no final é exportado o objeto de rotas para que o arquivo de entrada da aplicação "app.js" tenha acesso à camada de roteamento. O arquivo "app.js", presente na figura 22, implementa a definição das rotas e dos *middlewares* globais, como o uso de JSON como parâmetro de comunicação para o corpo da requisição e o uso de CORS para poder acessar a aplicação de qualquer domínio.

Figura 22 - Configurações prévias de roteamento

```

middlewares() {
  this.server.use(express.json());
  this.server.use(cors());
  this.server.use(
    '/files',
    express.static(path.resolve(__dirname, '..', 'tmp', 'uploads'))
  );
}

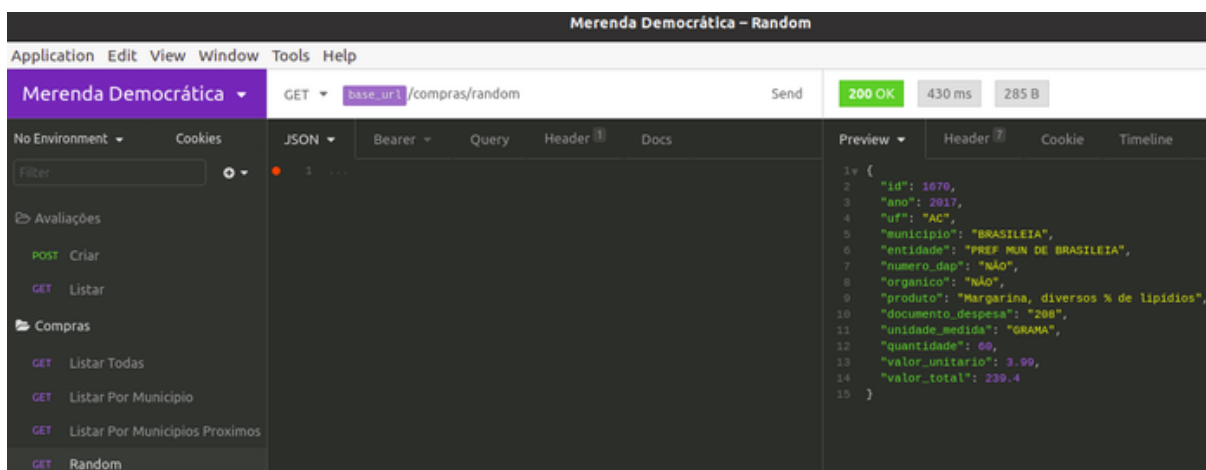
routes() {
  this.server.use(routes);
}

```

Fonte: O autor (2020)

No processo de desenvolvimento da API foi utilizado o software Insomnia para testar o funcionamento dos *endpoints*. Com o Insomnia é possível realizar requisições para o backend a partir dos *endpoints* definidos pela aplicação. Na figura 23 há uma chamada para uma requisição do tipo GET para o *endpoint* que retorna uma compra aleatória.

Figura 23 - Uso do Insomnia



Fonte: O autor (2020)

5.1.6 Endpoints

Para servir de suporte às funcionalidades presentes no aplicativo do MVP Merenda Democrática foram implementados os seguintes endpoints disponíveis na tabela 1.

Tabela 1 - Lista dos endpoints desenvolvidos (continua)

Endpoint	Método HTTP	Descrição
Usuários		
/usuarios	POST	Cria um novo usuário na base de dados do aplicativo
/usuarios	PUT	Atualiza o perfil do usuário logado
Sessão		
/sessions	POST	Cria uma sessão para autenticação
Avaliações		
/avaliacoes	GET	Retorna todas as avaliações cadastradas
/avaliacoes/indicio-fraude	GET	Retorna todos os indícios de fraude
/avaliacoes/:avaliacaoid	GET	Retorna uma avaliação específica
/avaliacoes/:usuarioid/usuario	GET	Retorna todas as avaliações realizadas por um determinado usuário
/avaliacoes	POST	Cria uma nova avaliação
Comentários		
/comentarios/:avaliacaoid	GET	Retorna todos os comentários de uma determinada avaliação
/comentarios	POST	Adiciona um novo comentário
Compras		
/compras	GET	Retorna todas as compras (com paginação)
/compras-municipio	GET	Retorna todas as compras de um município passado no corpo da requisição (com paginação)
/compras/random	GET	Retorna uma compra aleatória
/compras/:comprald/municipios-proximos	GET	Retorna as compras dos municípios vizinhos para o mesmo produto e mesma unidade de medida (retorna 10 resultados por consulta)
Preços		
/precos	POST	Adiciona um novo preço
Previsões		
/previsoes	GET	Retorna todas as previsões realizadas pelo TensorFlow
/previsoes/compra	GET	Retorna a previsão de preço para uma compra específica enviada por parâmetro no corpo da requisição
/previsoes/executar	GET	Executa previsões sobre um produto passado por

Tabela 1 - Lista dos endpoints desenvolvidos (conclusão)

Endpoint	Método HTTP	Descrição
		parâmetro no corpo das requisições

Fonte: O autor (2020)

Mais detalhes sobre as implementações dos endpoints podem ser obtidos a partir da análise do código-fonte.

5.1.7 Disponibilização do código-fonte do backend

O código-fonte está disponibilizado de forma pública no perfil pessoal do autor no GitHub, no seguinte link <https://github.com/MarceloHEcker/merenda-democratica-server>, bem como no repositório de códigos da UFSC.

5.2 FRONTEND DA APLICAÇÃO

O desenvolvimento da aplicação frontend ocorreu após a definição clara da arquitetura e o desenvolvimento da API REST, de forma que o frontend (aplicação desenvolvida com Ionic) possa utilizar todas as potencialidades do backend. De forma geral, a aplicação frontend comunica-se com o backend para apresentar os dados necessários, estruturar e captar as interações do usuário na avaliação de compra de merenda escolar. A aplicação frontend possui também controle de acesso e permite aos usuários visualizar as avaliações de compra que foram realizadas na plataforma por ele e por outros usuários.

Nos capítulos a seguir serão apresentadas as tecnologias e padrões de projeto utilizados no desenvolvimento do aplicativo, com detalhamento de sua implementação e telas (presentes no Apêndice A). Na seção “Disponibilização do código fonte” há mais detalhes de como o código da aplicação frontend pode ser acessado possibilitando assim sanar dúvidas de eventuais interessados, bem como servir de suporte para possíveis trabalhos futuros.

5.2.1 Ferramentas utilizadas

Para o desenvolvimento da aplicação frontend, levando em conta aspectos como produtividade e qualidade de código, foram utilizadas algumas ferramentas listadas abaixo. Essas ferramentas foram essenciais para o desenvolvimento do backend do MVP, pois oferecem recursos mínimos que possibilitaram atender

aspectos de qualidade e produtividade durante o desenvolvimento do projeto.

- NodeJS: software utilizado para construir aplicações Javascript. Serve como plataforma de execução e de desenvolvimento para o Ionic (tecnologia utilizada para construção de uma aplicação híbrida).
- Yarn: software utilizado para gerenciamento de pacotes e dependências. Apresenta algumas vantagens em relação aos seus competidores no mercado, como rapidez na instalação dos pacotes, estabilidade e fácil configuração.
- VS Code: IDE utilizada para desenvolver o código da aplicação. Possui alguns recursos como terminal integrado, que facilitaram a execução e testes do mesmo.
- Git: utilizado para versionamento do código.
- Google Chrome: navegador web que permite visualizar uma prévia de desenvolvimento da aplicação.

5.2.2 Tecnologias utilizadas

Para o desenvolvimento do frontend (aplicativo) foram utilizadas as seguintes tecnologias:

- Angular: *framework* Javascript para o desenvolvimento de aplicações web. Desenvolvido pelo Google se encontra hoje na versão 9.0, se tornando um dos *frameworks* mais utilizados pelo mercado.
- Ionic: *framework* Javascript para o desenvolvimento de aplicativos híbridos que rodam nas duas plataformas mais utilizadas no momento: iOS e Android. Permite que seja desenvolvido aplicações utilizando diversos *frameworks* como o Angular, React e Vue. Para esse trabalho iremos adotar o Angular, por ser o *framework* de maior conhecimento do autor.
- Cordova: API que acessa recursos dos dispositivos móveis, permitindo que o Ionic utilize recursos nativos dos smartphones, como câmera e o armazenamento.
- Typescript: *superset* que estende as funcionalidades do Javascript, adicionando alguns recursos como tipagem de variáveis e funções auxiliares. Desenvolvido pela Microsoft, é utilizado no Angular e no Ionic, tendo um excelente suporte na IDE Visual Studio Code.
- HTML: linguagem de marcação base para definir o conteúdo e a estrutura

da página. Por estarmos utilizando um framework Javascript, também temos acesso dentro do código HTML às anotações e *binding* de propriedades, tornando a definição da estrutura das páginas ainda mais poderosa.

- SASS: linguagem implementada com base no CSS, abreviação de “*Syntactically Awesome Style Sheets*”, permite descrever os estilos que as páginas web terão. Por se tratar de uma extensão do CSS, SASS possui alguns outros recursos adicionais que auxiliam no desenvolvimento e na manutenção do código do projeto, como a presença de variáveis e possibilidade de criar regras de estilo aninhada, facilitando assim a leitura das definições dos elementos.
- RXJS: biblioteca padrão do Angular para realizar requisições ao backend da aplicação, RxJS é utilizada para desenvolver chamadas assíncronas e baseadas em eventos usando sequências de dados conhecidas como Observables.

Todas as tecnologias adotadas são Open Source e podem ser utilizadas de maneira gratuita.

5.2.3 Implementação

Por se tratar de uma aplicação tendo como base o Ionic, que por sua vez usa os recursos do Angular, estamos implementando um paradigma muito próximo ao paradigma MVC (Model - View - Controller), mesmo com uma abstração de sua estrutura na forma de componentes. O paradigma MVC (que seria em uma tradução direta algo próximo à Modelo - Visão - Controle) é um dos paradigmas mais clássicos no desenvolvimento de aplicações e um dos primeiros paradigmas abordados durante o curso de Sistemas de Informação.

Utilizando esse paradigma podemos definir a estrutura da aplicação frontend. Cabe ressaltar algumas características dos componentes que formam o MVC antes de demonstrar como as responsabilidades e as camadas estão definidas dentro da arquitetura do projeto. Seguiremos a ordem das letras que dá origem a sigla.

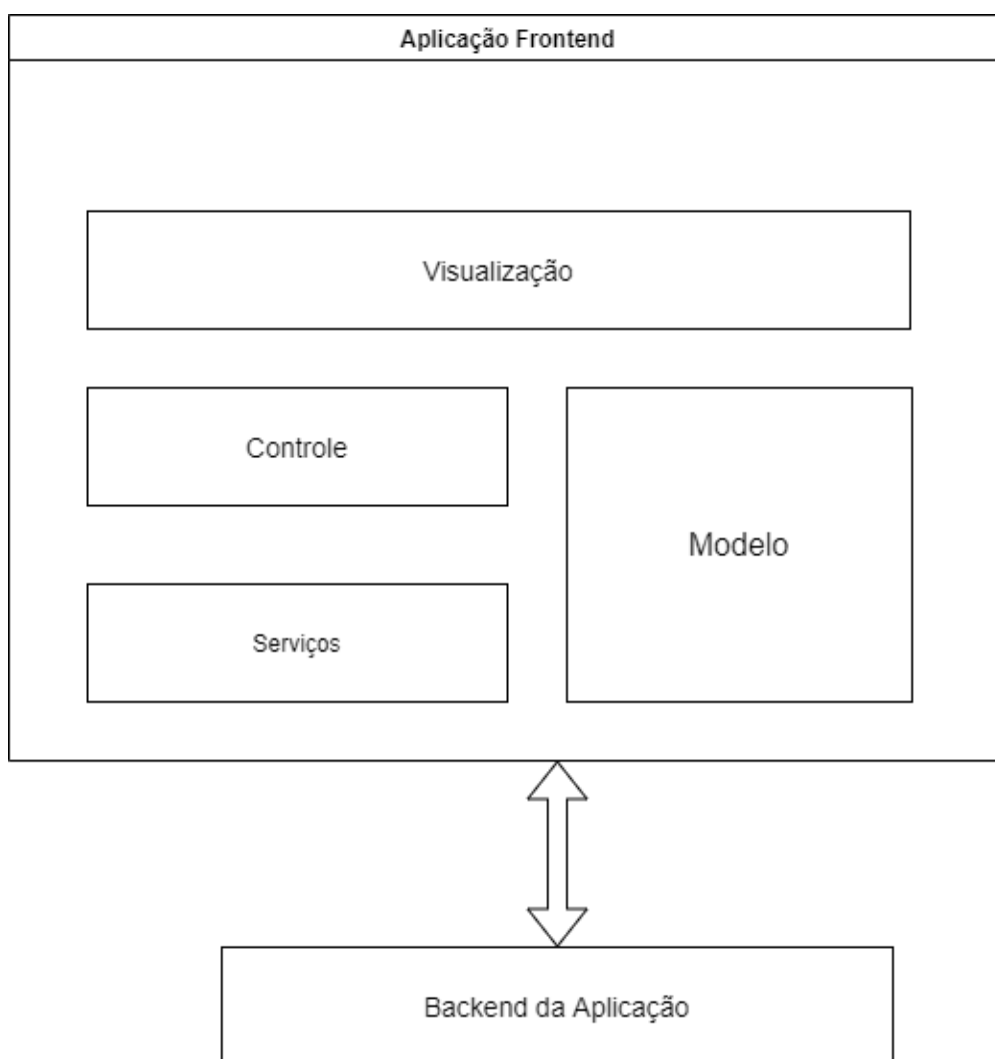
- Modelo: representado pela camada de domínio, reflete as entidades presentes no banco de dados, através da construção de interfaces e definição de seus atributos. Uma característica importante da camada de modelo dentro da nossa arquitetura proposta é que além das classes que declaram as entidades presentes na aplicação, faz parte dela os *providers* que implementam serviços com a função de interagir com os *endpoints* da API

REST e também definir regras de negócio específicas da aplicação.

- Visão: representada por “aquilo que pode ser visto”, a camada de visão tem como objetivo estruturar as páginas que serão exibidos para os usuários da aplicação. Recebe as informações da camada de controle.
- Controle: realiza uma espécie de intermediação entre os modelos e as views, sendo responsável por interpretar as ações dos usuários e retornar informações adequadas a essas ações.

Após termos como base os conceitos desse *framework* podemos iniciar o processo de definir a estrutura do frontend do nosso MVP Merenda Democrática. Abaixo, a figura 24 ilustra as nossas camadas de forma visual. Trataremos detalhes de implementação de cada uma dessas camadas durante o decorrer das próximas sessões.

Figura 24 - Arquitetura do frontend



Fonte: O autor (2020)

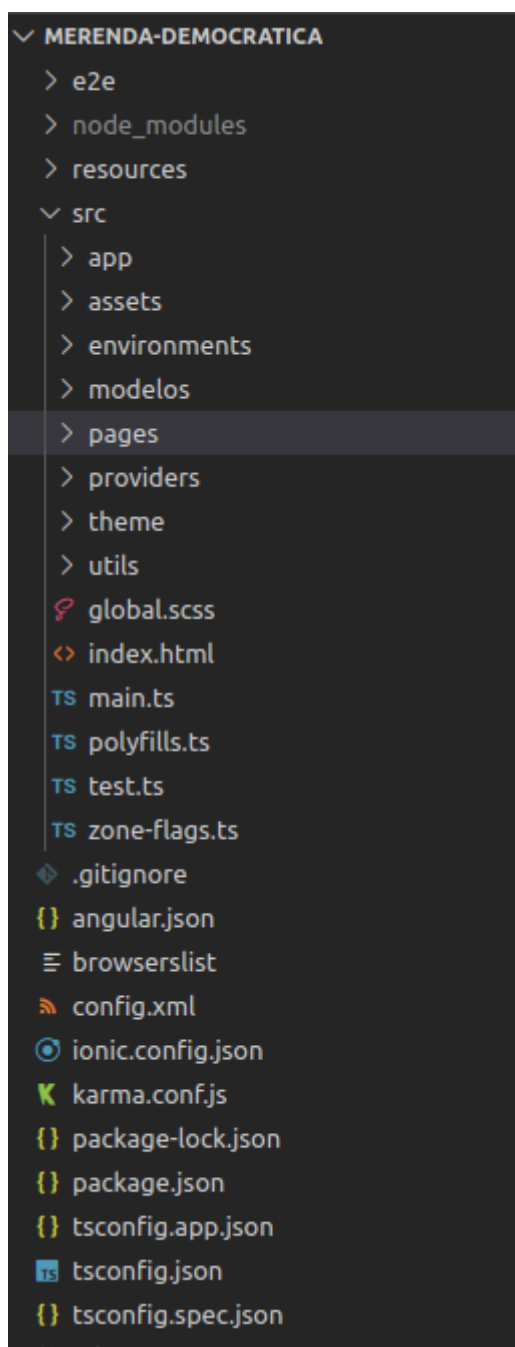
5.2.3.1 Visão do diretório da aplicação

Esta seção tem como objetivo demonstrar uma visão geral dos diretórios do frontend do nosso MVP Merenda Democrática. Basicamente todo o código-fonte customizado da nossa aplicação está disponível dentro da pasta “src” que centraliza também alguns recursos como funções utilitárias, *assets* da aplicação (notadamente imagens e ícones que são utilizadas pelo nosso app) e *environments* (com configurações específicas de cada um dos ambientes de desenvolvimento e de produção).

As pastas mais importantes da nossa aplicação são a pasta “app” que representa o componente de entrada da aplicação, a pasta “modelos” responsável pela camada de domínio, com a definição das entidades dos *endpoints* de nossa API REST, a pasta “*providers*” com os serviços de comunicação com a API REST e a pasta “pages” onde estão reunidos todas as páginas do sistema, sejam elas de login até a página de avaliação de compra.

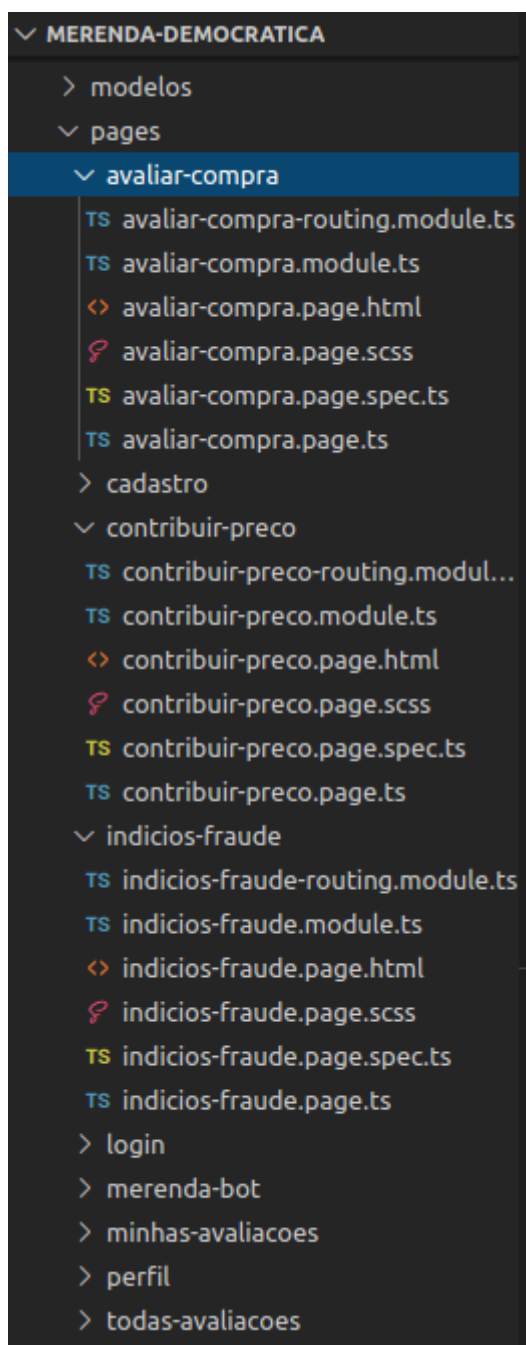
A figura 25 retrata uma visão global do conteúdo dos diretórios da aplicação, enquanto a figura 26 especifica o conteúdo da pasta “pages” de forma detalhada.

Figura 25 - Visão dos diretórios do frontend



Fonte: O autor (2020)

Figura 26 - Visão detalhada da pasta "pages"



Fonte: O autor (2020)

Cada uma das páginas funciona como uma espécie de componente, onde internamente é aplicado o padrão MVC. Cabe destacar os arquivos “.ts” geralmente se refere a camada de controle, enquanto “.scss” e “.html” à visualização das páginas. Nas seções abaixo explicaremos os detalhes de implementação da página “avaliar-compra” por ser a página mais importante do nosso aplicativo. As outras páginas também seguem um padrão de organização bastante similar.

5.2.3.2 Camada de domínio (modelos e serviços)

Esta camada da aplicação frontend é composta por classes que representam as entidades do modelo do banco de dados e as classes que implementam serviços que realizam a conexão com a API REST.

A figura 27 demonstra a implementação na classe "compra.ts". Todos os demais modelos seguem a mesma estrutura.

Figura 27 - Camada de domínio "Compra"

```
export interface Compra {
  id: number;
  ano: number;
  uf: string & { length: 2 };
  municipio: string;
  entidade: string;
  numero_dap: string;
  organico: string & { length: 1 };
  produto: string;
  documento_despesa: string;
  unidade_medida: string;
  quantidade: number;
  valor_unitario: number;
  valor_total: number;
  flag: string;
}
```

Fonte: O autor (2020)

Já a implementação da classe de serviços relativa a entidade Compra define de que forma a aplicação frontend interagirá com os *endpoints* de compras desenvolvidos em nosso backend. Para poder atingir tais objetivos são utilizados algumas classes utilitárias do Angular, como a classe "Injectable" que com uma anotação adicionado acima da definição da classe, indica que ela implementa um padrão de serviço e pode ser utilizada em outras classes através da injeção de dependências. A classe "HttpClient" implementa comunicação entre aplicações através do protocolo HTTP. Um objeto da classe "HttpClient" é injetado (através do conceito de injeção de dependências) dentro da classe de serviço.

No código da classe "ComprasServiceProvider" ocorre quatro comunicações com o backend, todas utilizando o protocolo GET, sendo elas: um método para listar todas as compras, um método para listar uma compra em específico, um método para uma compra aleatória e por fim, um método que solicita todas as compras dos municípios vizinhos de determinada compra. Todas as implementações são muito

parecidas, retornando para classe de controle para que realize a inscrição para receber os valores da requisição (utilizando assim o padrão de comunicação *Observable*, do RxJS).

A figura 28 demonstra a implementação da classe "ComprasServiceProvider". Demais classes seguem o mesmo padrão:

Figura 28 - Definição da classe de serviço

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

import { Compra } from '../../../modelos/compra';

@Injectable()
export class ComprasServiceProvider {

  private _url = 'http://localhost:3333';

  constructor( private _http: HttpClient ) {
  }

  index() {
    return this._http.get<Compra[]>( `${ this._url }/compras` );
  }

  get( compraId: number ) {
    return this._http.get<Compra>( `${ this._url }/compras/${ compraId }` );
  }

  getRandom() {
    return this._http.get<Compra>( `${ this._url }/compras/random` );
  }

  getMunicipiosProximos(orderId) {
    return this._http.get<any[]>(
      `${ this._url }/compras/${orderId}/municipios-proximos` );
  }
}
```

Fonte: O autor (2020)

5.2.3.3 Camada de visualização

Como relatado anteriormente, a camada de visão é a camada responsável por definir a estrutura das páginas do nosso aplicativo e pode ser identificada através dos arquivos que possuem extensão “.html” e “.scss”.

Uma característica bastante importante do Ionic é que ele já oferece alguns

componentes prontos que implementam algumas estruturas comuns de aplicações web, como listas, cards, campos para poder preencher dados e botões. Todas esses componentes estão presentes na documentação do Ionic que pode ser acessada a partir do site oficial do *framework*. Outra característica bastante importante é o sistema de *grids*, que é responsável por realizar a responsividade entre diferentes dispositivos, como *smartphones*, *tablets* e navegador de forma automática, possibilitando assim que seja desenvolvido somente um código, com instruções mais diretas e enxutas. Os componentes do Ionic possuem geralmente o sufixo “ion” e podem claramente ser identificados no código da tela de avaliação de compra. Essa tela é a tela mais importante da aplicação, já que é a partir dela que um usuário poderá realizar a avaliação da compra, informando para a nossa base de dados se a compra aparenta ser um indício de fraude ou não.

No topo de cada página vai o cabeçalho da aplicação responsável por identificar ao usuário em qual página ele se encontra. A figura 29 demonstra o código de estrutura do cabeçalho.

Figura 29 - Cabeçalho da página "Avaliar compra"

```
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <ion-title>Avaliar Compra</ion-title>
  </ion-toolbar>
</ion-header>
```

Fonte: O autor (2020)

O primeiro card da tela representa as informações da compra e foi estruturado como demonstrado na figura 30.

Figura 30 - Definição da estrutura de exibição da compra

```

<ion-card class="compra-card">
  <ion-card-content>
    <ion-list lines="none" *ngIf="compra">
      <ion-item>
        <ion-label>
          <h1>{{ compra.produto}}</h1>
          <p style="font-weight: bold;">{{compra.municipio}}/{{compra.uf}}</p>
        </ion-label>
      </ion-item>
      <ion-item>
        <ion-icon name="calendar-outline" slot="start"></ion-icon>
        <ion-label>
          <h2>{{compra.ano}}</h2>
          <p>Ano</p>
        </ion-label>
      </ion-item>
      <ion-item>
        <ion-icon name="calculator-outline" slot="start"></ion-icon>
        <ion-label>
          <h2>{{compra.unidade_medida}}</h2>
          <p>Medida</p>
        </ion-label>
      </ion-item>
      <ion-item>
        <ion-icon name="pricetags-outline" slot="start"></ion-icon>
        <ion-label>
          <h2>{{ compra.valor_total }}</h2>
          <p>Valor Total</p>
        </ion-label>
      </ion-item>
      <ion-item>
        <ion-icon name="reorder-four-outline" slot="start"></ion-icon>
        <ion-label>
          <h2>{{ compra.quantidade }}</h2>
          <p>Quantidade</p>
        </ion-label>
      </ion-item>
      <ion-item>
        <ion-icon name="pricetag-outline" slot="start"></ion-icon>
        <ion-label>
          <h2>{{ compra.valor_unitario }}</h2>
          <p>Valor Unitário</p>
        </ion-label>
      </ion-item>
    </ion-list>

    <div class="buttons-container">
      <ion-button color="success" shape="round">Aprovar</ion-button>
      <ion-button color="danger" shape="round">Rejeitar</ion-button>
    </div>
  </ion-card-content>
</ion-card>

```

Fonte: O autor (2020)

Cabe destacar a presença dos componentes do Ionic que possuem o sufixo “ion”, como ion-card e ion-list, além das marcações do angular como o *binding* de propriedades e utilitários do *framework* como **ngIf*.

O segundo card da tela mostra o preço estimado do produto de acordo com o processo de estimativa que será descrito em próximas seções do presente trabalho.

A definição da estrutura deste card encontra-se na figura 31.

Figura 31 - Definição da estrutura de exibição da estimativa

```

<ion-card *ngIf="estimativaPreco">
  <ion-card-header>
    <ion-card-subtitle>Gerada pelo aplicativo</ion-card-subtitle>
    <ion-card-title>Estimativa de Preço</ion-card-title>
  </ion-card-header>

  <ion-card-content>
    <ion-list text-wrap>
      <p>De acordo com todas as compras presentes no Merenda Democrática, a estimativa de preço para esse produto é de:</p>
      <h2 style="margin-top: 10px;font-weight: bold;">{{ estimativaPreco.valor_unitario | currency:'BRL':true }}</h2>
    </ion-list>
  </ion-card-content>
</ion-card>

```

Fonte: O autor (2020)

O terceiro card possui referências à compras similares (que possuem mesmo produto e mesma unidade de medida) realizada por municípios vizinhos ao município da compra. A definição da estrutura deste card encontra-se na figura 32.

Figura 32 - Definição da exibição de compras nos municípios vizinhos

```

<ion-card>
  <ion-card-header>
    <ion-card-subtitle>Compras - Valor Unitário</ion-card-subtitle>
    <ion-card-title>Municípios Próximos</ion-card-title>
  </ion-card-header>

  <ion-card-content>
    <ion-list>
      <ion-item *ngFor="let compra of comprasMunicipios" text-wrap>
        <ion-label>
          <h2>{{ compra.valor_unitario | currency:'BRL':true }}</h2>
          <h3 style="font-weight: bold;">{{ compra.municipio }}</h3>
          <p>{{ compra.produto }}</p>
        </ion-label>
      </ion-item>

      <ion-item *ngIf="!comprasMunicipios.length" text-wrap>
        <p>Nenhuma compra encontrada nos municípios próximos.</p>
      </ion-item>
    </ion-list>
  </ion-card-content>
</ion-card>

```

Fonte: O autor (2020)

O quarto e último card (presente na figura 33) da tela remete a comentários realizados por outros usuários do Merenda Democrático que já efetuaram a avaliação.

Figura 33 - Definição da exibição de comentários

```
<ion-card>
  <ion-card-header>
    <ion-card-subtitle>0 que outros usuários acham</ion-card-subtitle>
    <ion-card-title>Comentários</ion-card-title>
  </ion-card-header>

  <ion-card-content>
    <ion-list text-wrap>
      <ion-item *ngFor="let comentario of comentarios" text-wrap>
        <ion-avatar slot="start">
          
        </ion-avatar>
        <ion-label>
          <p>{{ comentario.comentario }}</p>
        </ion-label>
      </ion-item>

      <ion-item *ngIf="!comentarios.length" text-wrap>
        <p>Nenhum comentário encontrado para essa compra</p>
      </ion-item>
    </ion-list>
  </ion-card-content>
</ion-card>
```

Fonte: O autor (2020)

O resultado dos códigos da camada de visualização pode ser observado no Apêndice A.

5.2.3.4 Camada de controle

Por fim, temos a camada de controle, que é a responsável por realizar as chamadas tanto para os modelos e serviços quanto para as views. A classe de controle possui uma codificação via de regra mais simples do que as outras camadas. Para começar devemos importar a classe Component do Angular responsável por juntar as peças entre views e arquivos de estilo, bem como definir um seletor para a nova página. Esse seletor servirá tanto para personalizar o CSS, como para incluir em novos componentes construindo assim aplicações multicomponentes. A figura 34 exemplifica essas operações.

Figura 34 - Definição da classe de controle

```
import { Component, OnInit } from '@angular/core';

import { Compra } from 'src/modelos/compra';
import { ComprasServiceProvider } from 'src/providers/compras-service/compras-service';
import { ComentariosServiceProvider } from 'src/providers/comentarios-service/comentarios-service';
import { Comentario } from 'src/modelos/comentario';
import { PrevisoesServiceProvider } from 'src/providers/previsoes-service/previsoes-service';
import { Previsao } from 'src/modelos/previsao';

@Component( {
  selector: 'app-avaliar-compra',
  templateUrl: './avaliar-compra.page.html',
  styleUrls: [ './avaliar-compra.page.scss' ],
} )
export class AvaliarCompraPage implements OnInit {
```

Fonte: O autor (2020)

Uma classe de controle normalmente implementa os métodos do ciclo de vida de um componente no Angular. Nesse caso estamos implementando o método *OnInit* que executará durante a inicialização do componente. Também precisamos definir de forma global os valores que o controle poderá ter e definir também um construtor para esse componente, onde realizaremos a injeção de dependências dos serviços que ele utilizará. Por fim é exemplificado na figura 35, o código do método *OnInit*, que é executado ao inicializar o componente e é responsável por chamar os serviços de compra para obter uma compra aleatória e seus respectivos comentários, bem como compras de municípios vizinhos e estimativa de preço de acordo com o processo de aprendizagem descrito em seções abaixo.

Figura 35 - Inicialização dos dados da classe

```

compra: Compra;
comprasMunicipios: Compra[] = [];
comentarios: Comentario[] = [];
estimativaPreco: Previsao[];

constructor(
  private compraSvc: ComprasServiceProvider,
  private comentarioSvc: ComentariosServiceProvider,
  private previsaoSvc: PrevisoesServiceProvider,
) { }

ngOnInit() {

  this.compraSvc.getRandom().subscribe( res => {
    this.compra = res;

    this.compraSvc.getMunicipiosProximos( this.compra.id ).subscribe( compraData => {
      this.comprasMunicipios = compraData;
    } );

    this.comentarioSvc.getByOrder( this.compra.id ).subscribe( comentarioData => {
      this.comentarios = comentarioData;
    } );

    let payload = {
      'produto': this.compra.produto,
      'uf': this.compra.uf
    };

    this.previsaoSvc.indexCompra( payload ).subscribe( res => {
      this.estimativaPreco = this.getNearest( { quantidade: this.compra.quantidade }, res );
    } );
  } );
}

```

Fonte: O autor (2020)

Os demais métodos da classe poderão ser acessados no código-fonte da aplicação, mas não serão objetos de maiores explicações aqui por se tratarem de métodos de transformação e ordenação de dados, como o método *getNearest* e não demonstram os conceitos mais importantes da arquitetura.

5.2.4 Disponibilização do código-fonte do frontend

O código-fonte está disponibilizado de forma pública no perfil pessoal do autor no GitHub, no seguinte link <https://github.com/MarceloHEcker/merenda-democratica>, bem como no repositório de códigos da UFSC.

5.3 INDICADORES DE PREÇO DE COMPRA

Para oferecer melhores indicadores para os usuários poderem avaliar as compras oriundas do Painel de Preços do PNAE, foi proposto realizar alguns processos de *machine learning*. Neste trabalho preveremos o preço unitário de determinado produto, de acordo com a quantidade que foi comprada. Escolhemos essas duas variáveis com base nos atributos da entidade Compra, já que eram os dois campos com uma relação mais promissora dentre todos os atributos presentes na entidade.

Para realizar a previsão desses dados foi construído uma aplicação tendo como base o TensorFlow.js, biblioteca Javascript que implementa os métodos do TensorFlow, que possui uma série de componentes e estruturas prontas para poder realizar processamentos de aprendizagem de máquina. O ambiente escolhido para executar a previsão desses dados foi o navegador Google Chrome, já que permite visualizar detalhes dos procedimentos em forma de gráficos.

Na seção abaixo serão exemplificados mais detalhes técnicos do processo de implementação desse algoritmo. Para explicar de melhor forma as instruções realizadas dentro do código, além dos comentários presentes no fluxo normal do presente trabalho serão adicionados comentários no código-fonte da aplicação.

5.3.1 Implementação do algoritmo

Devido ao fato de estarmos treinando um modelo para realizar previsões de uma série de números contínuos, essa tarefa é comumente definido como uma tarefa de regressão. No presente trabalho realizaremos uma espécie de aprendizado supervisionado, onde executaremos a entrada de uma série de dados de quantidade de compras, com o resultado correto equivalente ao valor unitário, deixando que o modelo aprenda com essas informações e traça então padrões para prever novos dados.

A implementação do modelo de previsão dos dados foi baseado no artigo descrito pelo Google na sua plataforma de CodeLabs (2020).

- Carregar os dados e analisar *outliers*.
- Definir a arquitetura do modelo a ser utilizada.
- Preparação dos dados para a realização do treinamento.
- Treinar o modelo e monitorar a performance enquanto ele é treinado.
- Avaliar o modelo treinado realizando algumas previsões.

O presente trabalho também realizará algumas boas práticas de *machine learning*, notadamente regressão, recomendadas pelos desenvolvedores do TensorFlow, como ordenar os registros de forma aleatória para evitar que a distribuição dos dados dentro de nosso arquivo de dados interfira nos valores previstos, bem como normalizar os dados em uma faixa de 0-1 para reduzir o efeito de disparidades muito grande nos valores da distribuição (será aplicado um algoritmo que remove também *outliers* muito grandes dos registros). No presente trabalho também será utilizado a biblioteca “tfjs-vis” para construir alguns gráficos que tornarão o processo de previsão dos dados ainda mais transparente e acessível.

5.3.1.1 Definição da camada de visualização

O primeiro passo da implementação é definir o "index.html" responsável por permitir o acesso aos recursos do TensorFlow.js no nosso ambiente de execução, o navegador. O código do arquivo "index.html" está representado na figura 36.

Figura 36 - Definição da visualização

```
<!DOCTYPE html>
<html>

<head>
  <title>Obtenção de Indicadores da Merenda</title>

  <!-- HTML 4 -->
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <!-- HTML5 -->
  <meta charset="utf-8" />

  <!-- Import TensorFlow.js -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0/dist/tf.min.js"></script>
  <!-- Import tfjs-vis -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs-vis@1.0.2/dist/tfjs-vis.umd.min.js"></script>

  <!-- Import the main script file -->
  <script src="script.js"></script>
</head>

<body>
</body>

</html>
```

Fonte: O autor (2020)

Basicamente é definido a estrutura HTML e importado dois arquivos Javascript externos, o arquivo base do TensorFlow.js responsável pela realização dos processos de aprendizado de máquina, bem como o arquivo tfjs-vis,

responsável pelos gráficos de visualização da execução do processo e do resultado final da previsão. É importado também o arquivo `script.js` onde constará a implementação dos passos de treinamento do nosso modelo.

5.3.1.2 Carregamento e tratamento dos dados

O segundo passo foi realizar o carregamento dos dados a serem adicionados ao modelo, bem como filtrar os outliers para que o modelo possa compreender os dados da melhor forma e retornar previsões mais adequadas ao preço do valor unitário das compras. Buscando produzir previsões válidas, o treinamento foi realizado de acordo com cada um dos produtos presentes na nossa base de compras, bem como adotando a mesma unidade de medida para todas compras presentes no treinamento. Para testar a viabilidade desta solução foi realizado o treinamento somente dos 5 produtos mais presentes na nossa base de compras, sendo eles “Açúcar”, “Banana”, “Biscoito salgado”, “Carne Bovina” e “Polpa de frutas”. O processo de realização das previsões está descrito no Apêndice C do presente trabalho.

A figura 37 demonstra a implementação do carregamento e da preparação dos dados adicionados ao modelo. A função basicamente realiza a leitura de uma série de valores dentro de um arquivo JSON, transforma todos os dados em um array de JSON, bem como realiza o mapeamento das quantidades e dos respectivos valores unitários. Após isso é realizada uma filtragem dos outliers pelas duas dimensões dos dados da amostra (quantidade e valor unitário).

Figura 37 - Método de carregamento dos dados

```
/**
 * Pegando os dados brutos e realizando uma limpeza
 */
async function getData() {

  const productsDataReq = await fetch('http://localhost/tensorflow-preview/produtos/acucar.json');
  const productsData = await productsDataReq.json();
  const cleaned = productsData.map(product => ({
    valor_unitario: parseFloat(product.valor_unitario),
    quantidade: parseFloat(product.quantidade),
  })))
  .filter(product => (product.valor_unitario != null && product.quantidade != null));

  //filtrando registros com outliers no valor unitário
  const filteredXValues = filterOutliers(cleaned.map(item => item.valor_unitario));
  const filteredXData = cleaned.filter(item => filteredXValues.includes(item.valor_unitario));

  //filtrando registros com outliers na quantidade
  const filteredValues = filterOutliers(filteredXData.map(item => item.quantidade));
  const filteredData = filteredXData.filter(item => filteredValues.includes(item.quantidade));

  return filteredData;
}
```

Fonte: O autor (2020)

A função que realiza filtra os outliers está exemplificada na figura 38. Os outliers são dados que diferem drasticamente de todos os outros, são conhecidos também como pontos fora da curva. O grande problema da existência de outliers muito grandes em nossa amostra de dados é que eles podem enviesar de forma negativa os resultados obtidos pelo nosso modelo.

Figura 38 - Filtragem dos outliers

```
/**
 * Método que filtra outliers de elementos presentes em um array
 *
 */
function filterOutliers(someArray) {
  var values = someArray.concat();

  values.sort(function (a, b) {
    return a - b;
  });

  var q1 = values[Math.floor((values.length / 4))];

  var q3 = values[Math.ceil((values.length * (3 / 4)))];
  var iqr = q3 - q1;

  var maxValue = q3 + iqr * 1.5;
  var minValue = q1 - iqr * 1.5;

  var filteredValues = values.filter(function (x) {
    return (x <= maxValue) && (x >= minValue);
  });

  return filteredValues;
}
```

Fonte: O autor (2020)

5.3.1.3 Arquitetura do modelo

O terceiro passo é definir a arquitetura do modelo. A arquitetura do modelo nada mais é qual algoritmo utilizaremos para realizar as nossas previsões, pegando uma entrada e produzindo uma saída. Quando utilizamos redes neurais como no caso deste trabalho a arquitetura é composta por uma série de camadas de “neurônios” com um peso, que direcionarão os dados de saída. O processo de treinamento aprende o valor ideal para esses pesos.

A figura 39 retrata a função utilizada para definir a arquitetura de nosso modelo, bem como apresenta comentários demonstrando o que é definido em cada linha:

Figura 39 - Criação do modelo

```
/**
 * Método responsável por criar o modelo para realizar
 */
function createModel() {

  //criando um modelo utilizando a API sequencial (1 entrada - 1 saída)
  const model = tf.sequential();

  //definindo uma camada de entrada dos dados
  model.add(tf.layers.dense({ inputShape: [1], units: 1, useBias: true }));

  //definindo camadas intermeditárias ocultas que utilizam a função de ativação sigmóide
  model.add(tf.layers.dense({ units: 50, activation: 'sigmoid' }));

  //definindo uma camada de saída dos dados
  model.add(tf.layers.dense({ units: 1, useBias: true }));

  return model;
}
```

Fonte: O autor (2020)

5.3.1.4 Preparando os dados para treinamento

O quarto passo é a preparação dos dados para realização do treinamento. Buscando obter os benefícios de performance que o TensorFlow nos oferece, nós precisamos converter os nossos dados em tensores. Tensores já foram definidos na contextualização teórica do presente trabalho. Para produzir melhores previsões nós executaremos algumas transformações em nossos dados, como a distribuição de forma randômica dos dados dentro da nossa coleção e normalização seguindo a escala 0 - 1.

A figura 40 ilustra a função utilizada para preparar os dados para o treinamento, bem como apresenta comentários que exemplificam pontualmente os detalhes do nosso código:

Figura 40 - Convertendo dados para tensores

```

/*
 * Converte os dados de entrada para tensores que possamos realizar aprendizado
 * de forma mais eficiente
 */
function convertToTensor(data) {

  return tf.tidy(() => {

    //randomizando a ordem dos dados
    tf.util.shuffle(data);

    //convertendo os dados para tensores. 1 tensor de entrada e outro tensor com as saídas
    const inputs = data.map(d => d.quantidade)
    const labels = data.map(d => d.valor_unitario);

    const inputTensor = tf.tensor2d(inputs, [inputs.length, 1]);
    const labelTensor = tf.tensor2d(labels, [labels.length, 1]);

    // normalizando os dados na escala 0-1 utilizando as funções utilitárias min e max do TensorFlow
    const inputMax = inputTensor.max();
    const inputMin = inputTensor.min();
    const labelMax = labelTensor.max();
    const labelMin = labelTensor.min();

    // funções auxiliares do tensor flow evitam que seja necessário laços para determinar
    // valores mínimos e máximos
    const normalizedInputs = inputTensor.sub(inputMin).div(inputMax.sub(inputMin));
    const normalizedLabels = labelTensor.sub(labelMin).div(labelMax.sub(labelMin));

    // retornando os tensores e as informações para poder recuperar essas valores novamente
    return [
      inputs: normalizedInputs,
      labels: normalizedLabels,
      inputMax,
      inputMin,
      labelMax,
      labelMin,
    ];
  });
}

```

Fonte: O autor (2020)

5.3.1.5 Treinamento do modelo

O próximo passo no processo de machine learning é executar o treinamento do modelo, já que temos a instância do modelo criada e os dados representados em formato de tensores. Nesse processo é importante alguns conceitos como a escolha do otimizador e da medida de erro. O otimizador é o algoritmo que vai atuar como controlador das atualizações do modelo, governando os parâmetros enviados para o modelo durante a execução das interações dentro do conjunto de dados (*dataset*), com o objetivo de construir um modelo cada vez melhor (que tenha melhores medidas de erro). Há muitos otimizadores disponíveis no TensorFlow.js, mas neste trabalho utilizaremos um otimizador conhecido como adam. Ele é um dos

otimizadores mais utilizados, é bastante efetivo e não necessita de nenhuma configuração adicional. Já a medida de erro, é a função que diz como o modelo está desempenhando seu papel em cada um dos blocos de dados que estão sendo analisados. Para esse trabalho nós utilizaremos uma medida comum chamada "*Mean Squared Error - mse*", medida de erro ao quadrado em tradução direta, obtida dividindo o quadrado das diferenças de valores previstos pelo modelo aos valores reais dividido pelo número de elementos do bloco analisado. Essa função é amplamente utilizada em exemplos de regressão linear.

A figura 41 abaixo ilustra a função utilizada para realizar o treinamento dos dados, bem como apresenta comentários que exemplificam pontualmente os detalhes do nosso código:

Figura 41 - Treinamento do modelo

```
async function trainModel(model, inputs, labels) {  
  
  //preparando o modelo para treinamento  
  model.compile({  
    //escolhendo o otimizador  
    optimizer: tf.train.adam(),  
    //definindo a medida de erro  
    loss: tf.losses.meanSquaredError,  
    metrics: ['mse'],  
  });  
  
  //definindo o tamanho dos blocos de dados utilizados no treinamento  
  const batchSize = 32;  
  //definindo o numero de interações inteiras realizadas no dataset  
  const epochs = 50;  
  
  //iniciando o treinamento e exibindo valores do índice de erro em um gráfico  
  return await model.fit(inputs, labels, {  
    batchSize,  
    epochs,  
    shuffle: true,  
    callbacks: tfvis.show.fitCallbacks(  
      { name: 'Training Performance' },  
      ['loss', 'mse'],  
      { height: 200, callbacks: ['onEpochEnd'] }  
    )  
  });  
}
```

Fonte: O autor (2020)

5.3.1.6 Realizando as previsões

Nesse passo definiremos a função que executa as previsões dos valores unitários de compra utilizando o nosso modelo treinado nos passos anteriores. Cabe destacar nesse processo a criação de 100 valores que estão entre o range 0 - 1 atentando assim o critério de que nossos valores previstos estejam entre os valores

possíveis da nossa faixa de dados.

A figura 42 abaixo representa a implementação das previsões de preço de valor unitário:

Figura 42 - Teste do modelo

```
function testModel(model, inputData, normalizationData) {
  const { inputMax, inputMin, labelMin, labelMax } = normalizationData;

  const [xs, preds] = tf.tidy(() => {

    // gerando previsões para uma faixa uniforme de valores entre 0 e 1
    const xs = tf.linspace(0, 1, 100);
    const preds = model.predict(xs.reshape([100, 1]));

    //desnormalizando dos dados
    const unNormXs = xs
      .mul(inputMax.sub(inputMin))
      .add(inputMin);

    const unNormPreds = preds
      .mul(labelMax.sub(labelMin))
      .add(labelMin);

    return [unNormXs.dataSync(), unNormPreds.dataSync()];
  });

  //dados previstos
  const predictedPoints = Array.from(xs).map((val, i) => {
    return { x: val, y: preds[i], produto: 'Açúcar', uf: 'AC' }
  });

  //dados originais
  const originalPoints = inputData.map(d => ({
    x: d.quantidade, y: d.valor_unitario,
  }));

  // plotando o resultado de nossas previsões em um gráfico
  tfvis.render.scatterplot(
    { name: 'Previsões do Modelo vs Valores Originais' },
    { values: [originalPoints, predictedPoints], series: ['original', 'predicted'] },
    {
      xLabel: 'Quantidade',
      yLabel: 'Valor Unitário',
      height: 300
    }
  );
}
```

Fonte: O autor (2020)

5.3.1.7 Iniciando a execução

Após todas as funções que compõem o processo de machine learning estarem definidas, é hora de colocar todos os pedaços juntos e invocar a função que

começa a executar o processo quando carregamos a página web. Após o treinamento e execução do modelo de previsão dos valores unitários, os resultados foram armazenados em nossa base de dados de acordo com procedimento realizado no Apêndice C. Imagens do processo de machine learning em execução também estarão nesse mesmo apêndice.

A figura 43 demonstra a implementação da função responsável por agregar todos os passos descritos anteriormente e a execução da mesma quando a página (DOM) for carregado.

Figura 43 - Execução do modelo

```
async function run() {  
  
  // carregando os dados  
  const data = await getData();  
  
  // construindo modelo  
  const model = createModel();  
  tfvis.show.modelSummary({ name: 'Definição do Modelo' }, model);  
  
  // convertendo os dados para tensores  
  const tensorData = convertToTensor(data);  
  const { inputs, labels } = tensorData;  
  
  // realizando o treinamento do modelo  
  await trainModel(model, inputs, labels);  
  
  // realizando previsões com o modelo treinado  
  testModel(model, data, tensorData);  
}  
  
document.addEventListener('DOMContentLoaded', run);
```

Fonte: O autor (2020)

A descrição do processo de execução pode ser encontrada no Apêndice C do presente trabalho.

5.3.2 Disponibilização do código-fonte

O código-fonte está disponibilizado de forma pública no perfil pessoal do autor no GitHub, no seguinte link <https://github.com/MarceloHEcker/merenda-democratica-server>, bem como no repositório de códigos da UFSC.

5.4 COMUNICAÇÃO COM REDES SOCIAIS

Para divulgar possíveis indícios de fraude apontados por usuários da

aplicação nas redes sociais - no caso deste trabalho, o Twitter - foi criada uma integração que dispara um evento para realizar um post na rede social. Essa postagem tem como objetivo alertar a população e os órgãos competentes sobre possíveis compras suspeitas. Todas as postagens são realizadas dentro do perfil do projeto @MerendaBot e não identificam o usuário que realizou a avaliação.

A integração com as redes sociais é um dos objetivos específicos mais importantes do presente trabalho, já que busca de forma trazer as compras realizadas pelas entidades de forma mais acessível possível para o cidadão, que entre outras razões, se vê assim desobrigado de baixar o aplicativo para ter acesso a possíveis indícios de fraude. A escolha da rede social foi realizada seguindo essa mesma premissa, já que o Twitter é hoje a rede social onde a comunicação acontece de forma mais instantânea, devido entre outras razões - a pequena quantidade de texto característico de um *tweet*, além de ter uma comunidade bastante ativa, incluindo entre os membros dessa comunidade perfis de pessoas públicas e de entidades organizacionais.

Para realizar a integração com o Twitter foi necessário realizar o cadastro dentro da plataforma Twitter Developers. Esse cadastro é bastante complexo em relação às outras plataformas, mas após o preenchimento de um formulário com uma série de perguntas acerca do uso previsto para a conta, o Twitter avisa que a aplicação está sob análise. Depois de alguns dias o Twitter pode retornar concedendo o acesso, negando o cadastro como desenvolvedor ou pedindo mais detalhes sobre o uso da aplicação. Esse último cenário aconteceu com o presente trabalho. Após o retorno para a equipe responsável pelo cadastro das aplicações, foi concedido o acesso de desenvolvedor ao perfil do trabalho. Com isso foi necessário gerar uma *"consumer_key"*, *"consumer secret"*, *"access token"* e *"access token secret"*, sendo que essas informações serão vitais na hora de definirmos o código de integração.

A integração irá apenas realizar uma postagem no Twitter. Para isso foi utilizado um pacote chamado *Twit*, que é responsável por abstrair as operações realizadas junto à API do Twitter. O código abaixo, apresentado na figura 44, é responsável por criar e inicializar o objeto do pacote *Twit*, informando assim os dados do aplicativo que criamos na nossa conta de desenvolvedor.

Figura 44 - Criação do objeto Twit

```
const T = new Twit({
  consumer_key: '...',
  consumer_secret: '...',
  access_token: '...',
  access_token_secret: '...',
  timeout_ms: 60 * 1000, // optional HTTP request timeout to apply to all requests.
  strictSSL: true,      // optional - requires SSL certificates to be valid.
});
```

Fonte: O autor (2020)

Após a criação do objeto Twit temos acesso a todas as suas funcionalidades. Nesse caso realizaremos uma postagem com a informação de um possível indício de fraude. O código presente na figura 45 demonstra a criação da frase que será postada no perfil, bem como a chamada ao objeto para realizar a postagem.

Figura 45 - Criação do objeto para postagem

```
const tweetContent = `Indício de fraude apontado por usuário na compra realizada! De
  ${order.produto} - ${order.valor_unitario} por ${order.unidade_medida} em
  ${order.municipio}/${order.uf} efetuada em ${order.year}`;

T.post('statuses/update', { status: tweetContent }, function (err, data, response) {
  console.log('data from twitter', data);
});
```

Fonte: O autor (2020)

Só será realizado a postagem caso as avaliações indiquem indícios de fraude. Para realizar todo esse processo é criado uma condicional após a realização da avaliação. O código presente na figura 46 demonstra todo esse processo, e une os códigos presentes tanto na figura 44 quanto na figura 45.

Figura 46 - Condicional na avaliação

```

if (!avaliacao.status) {
  const order = await Compra.findByPk(compra_id);

  const T = new Twit({
    consumer_key: '...',
    consumer_secret: '...',
    access_token: '...',
    access_token_secret: '...',
    timeout_ms: 60 * 1000, // optional HTTP request timeout to apply to all requests.
    strictSSL: true, // optional - requires SSL certificates to be valid.
  });

  const tweetContent = `Indício de fraude apontado por usuário na compra realizada! De
  ${order.produto} - ${order.valor_unitario} por ${order.unidade_medida} em
  |${order.municipio}/${order.uf} efetuada em ${order.year}`;

  T.post('statuses/update', { status: tweetContent }, function (err, data, response) {
    console.log('data from twitter', data);
  });
}

```

Fonte: O autor (2020)

O resultado de execução da integração com a rede social Twitter pode ser visualizado na figura 47.

Figura 47 - Resultado de integração com o Twitter

The image shows a Twitter profile for 'Merenda Bot' (@MerendaBot). The bio reads 'Trabalho de conclusão de curso de Sistemas de Informação'. The profile was joined in May 2020 and has 0 following and 2 followers. The 'Media' tab is selected, showing a tweet from 41 seconds ago. The tweet text is: 'Indício de fraude apontado por usuário na compra realizada! De Banana - R\$ 17,50 por QUILO em MunicipioTeste efetuada em 2017'. The tweet includes icons for reply, retweet, like, share, and a menu.

Fonte: O autor (2020)

6 CONCLUSÃO

Neste trabalho foi realizado um extenso estudo envolvendo questões como dados abertos, transparência governamental e democracia participativa, bem como conceitos aprendidos durante as aulas do curso de Sistemas de Informação como Programação Orientada a Objetos, Engenharia de Software, Banco de Dados, Inteligência Artificial e Estatística. A junção desses conhecimentos caracteriza-se como um excelente apanhado dos conteúdos estudados durante o período da graduação.

As aplicações desenvolvidas durante esse projeto podem servir como modelo para tornar as compras da merenda escolar mais compreensíveis ao cidadão comum, que acredita na importância do uso dos recursos públicos de maneira consciente, evitando assim superfaturamento das compras ou ação de agentes corruptos durante o processo. Trazendo a tona as compras realizadas pelos órgãos públicos, é possível assim que sejam encontrados problemas que normalmente passariam despercebidos já que é incomum que os cidadãos acessem as bases de dados governamentais, ainda mais se elas apresentam problemas de performance, como é o caso do Painel de Preços do PNAE.

Podemos concluir que o trabalho atingiu o seu objetivo geral, já que foi possível realizar um estudo aprofundado, com revisão bibliográfica dos principais tópicos, de medidas tecnológicas que possam auxiliar a ampla execução da democracia participativa e fiscalização das compras efetuadas por órgãos governamentais e disponíveis através da Lei de Acesso à Informação (LAI). Também foi possível desenvolver o ambiente prático para que as pessoas possam contribuir avaliando as compras da merenda escolar deixando a sua impressão se as mesmas possuem indícios que possam ser caracterizados como fraude.

Os objetivos específicos também foram atingidos, na medida em que foi realizado uma revisão bibliográfica dos aspectos apontados na definição do trabalho: sociedade da informação, corrupção, Portal da Transparência, dados abertos, LAI, ciberativismo e *hacktivismo* e possíveis tecnologias que podem ser utilizadas nesse cenário como Programação MapReduce, Hadoop e TensorFlow.

Cabe destacar também que os objetivos específicos práticos foram atingidos, já que foi implementado tanto a API REST, a solução mobile e o modelo para prever o valor unitário das compras de nossa base de dados. De forma a buscar uma aproximação ainda maior dos cidadãos com as compra públicas, foi desenvolvido uma integração com a rede social *Twitter*. Para o desenvolvimento das aplicações propostas foi utilizado essencialmente tecnologias Open Source e tecnologias gratuitas, reforçando assim a distribuição do conhecimento e o papel da tecnologia

para transformar a vida das pessoas.

Há um potencial de utilização bastante interessante desse trabalho por parte da sociedade. Além da utilização dos cidadãos para conhecer mais detalhes sobre a compra da merenda escolar, instituições governamentais podem usar os resultados consolidados tanto para detectar fraudes, quanto para otimizar o seu processo de compra já que essa plataforma tem o potencial de ser expandida para um alcance nacional.

O código-fonte da parte prática do presente TCC se encontra disponível de forma pública, sendo possível o seu aperfeiçoamento em trabalhos futuros. Algumas decisões de arquitetura de projeto foram tomadas para facilitar esse desenvolvimento contínuo por parte de novos trabalhos. Por fim, cabe destacar que a informação é o produto de maior valor no nosso século. Somente o povo bem informado, poderá fazer frente a ações que não são de seu interesse e conduzir assim o Brasil no caminho do desenvolvimento para deixar de ser considerado o país do futuro, para se tornar finalmente, o país do presente.

6.1 TRABALHOS FUTUROS

Alguns trabalhos futuros podem ser desenvolvidos a partir da estrutura que construímos neste trabalho. A área com um campo maior de possibilidades é certamente a área de inteligência artificial que pode se beneficiar dos dados disponibilizados pela API REST, na medida em que novos modelos utilizando outras técnicas além regressão sejam construídos para analisar as compras. A partir das avaliações dos usuários também será possível construir um modelo de classificação, onde o algoritmo entende a estrutura dos dados que formam um possível indício de fraude e consegue comparar com novos registros, categorizando-os assim como fraude ou não. Também poderão ser utilizados algoritmos de aprendizagem não-supervisionados como o K-means, criando grupos (*clusters*) entre as compras para obter novas visões sobre o problema.

Outra possibilidade de trabalho futuro que temos é a integração deste com soluções já estabelecidas no mercado, como a Operação Serenata do Amor e os robôs Jarbas e Rosie. Refazer a nossa API e o nosso modelo de previsão de preços utilizando outras tecnologias como a linguagem de programação Python também apresenta um futuro bastante promissor, já que hoje Python é a linguagem de programação da moda na área de inteligência artificial.

É importante frisar que embora o objetivo desse trabalho seja voltado à democracia participativa e ao cidadão, nada impede que trabalhos futuros desenvolvam tecnologias que possam auxiliar os órgãos governamentais,

notadamente as prefeituras, na compra da merenda escolar. Toda tecnologia que possa tornar os órgãos governamentais mais eficientes é muito bem-vinda.

Por fim, projetos que envolvam mais áreas do que apenas a tecnologia podem ser realizados, como a condução de testes de experiência com o usuário, para poder redesenhar o aplicativo ou planejar novas features que sejam úteis no dia-a-dia. Para tornar a aplicação esteticamente mais agradável, um trabalho de desenvolvimento de um projeto focado no design do frontend é recomendado.

REFERÊNCIAS

AHLERT, Mara; AHLERT, Edson Moacir. A sociedade da informação e os avanços tecnológicos como instrumentos no combate à corrupção. **Revista Destaques Acadêmicos**, Lajeado, v. 8, n. 4, 2016.

ALESP - ASSEMBLÉIA LEGISLATIVA DO ESTADO DE SÃO PAULO. **Democracia Participativa**: uma nova forma de entendermos a democracia. 2016. Disponível em: https://www.al.sp.gov.br/repositorio/bibliotecaDigital/20293_arquivo.pdf. Acesso em: 15 Nov. 2018.

AQUARELA. **O que é Data Analytics?**. 2018. Disponível em: <https://www.aquare.la/o-que-e-data-analytics/>. Acesso em: 15 Nov. 2018.

ARAÚJO, Narallyne Maciel de. **Dados abertos do governo brasileiro**: entendendo as perspectivas de fornecedores de dados e desenvolvedores de aplicações ao cidadão. Rio Grande do Norte, f. 136, 2017. Dissertação (Mestrado em Sistemas e Computação) - Universidade Federal do Rio Grande do Norte.

ARAÚJO, Willian Fernandes. **Ciberativismo**: levantamento do estado da arte na pesquisa no Brasil. 2013. Disponível em: <https://document.onl/documents/ciberativismo-levantamento-do-estado-da-arte-na-pesquisa-710-dados.html>. Acesso em: 30 Jun. 2020.

BRASIL. Lei n. 8666, de 21 de junho de 1993. Regulamenta o art. 37, inciso Xxi da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências. Diário Oficial da União. Brasília, 21 de junho de 1993.

BRASIL. MINISTÉRIO DA TRANSPARÊNCIA E CONTROLADORIA GERAL DA UNIÃO. **Guia de implantação de Portal da Transparência**. **Imprensa Nacional**. Brasília, 2013. 32 p. Disponível em: http://www.cgu.gov.br/Publicacoes/transparencia-publica/brasil-transparente/arquivos/guia_portaltransparencia.pdf. Acesso em: 15 Nov. 2018.

BRASIL. MINISTÉRIO DA TRANSPARÊNCIA E CONTROLADORIA-GERAL DA UNIÃO. **Resultados**: 2017c. Disponível em: <http://www.cgu.gov.br/assuntos/transparencia-publica/escala-brasil-transparente>. Acesso em: 15 Nov. 2018.

BRASIL. MINISTÉRIO DA TRANSPARÊNCIA, FISCALIZAÇÃO E CONTROLADORIA-GERAL DA UNIÃO. **Aplicação da lei de acesso à informação na Administração Pública Federal**. **Imprensa Nacional**. Brasília, 2016. Disponível em: http://www.acessoainformacao.gov.br/central-de-conteudo/publicacoes/arquivos/aplicacao_lai_2edicao.pdf. Acesso em: 15 Nov. 2018.

BRASIL. MINISTÉRIO DO PLANEJAMENTO DESENVOLVIMENTO E GESTÃO. **Dados Abertos Governamentais**. 2018. Disponível em: <https://www.governodigital.gov.br/transformacao/cidadania/dados-abertos/dados-abertos-governamentais>. Acesso em: 15 Nov. 2018.

BRASIL. ROGÉRIO JORDÃO. CONTROLADORIA GERAL DA UNIÃO (ORG.). **Acesso à Informação Pública**: Uma introdução à Lei 12.527, de 18 de novembro de 2011. Imprensa Nacional, 2011. 25 p. Disponível em: <http://www.acessoinformacao.gov.br/central-de-conteudo/publicacoes/arquivos/cartilhaacessoinformacao.pdf>. Acesso em: 15 Nov. 2018.

BUGARIN, Maurício Soares. **Controle dos gastos públicos no Brasil: instituições oficiais, controle social e um mecanismo para ampliar o envolvimento da sociedade**. Rio de Janeiro: Adenauer-Stiftung, 2003.

CAMINHA, Marco Aurélio Lustosa. **A corrupção na Administração Pública do Brasil**. 2003. Disponível em: <https://jus.com.br/artigos/4657/a-corrupcao-na-administracao-publica-no-brasil>. Acesso em: 15 Nov. 2018.

CASTELLS, Manuel. **A era da informação: economia, sociedade e cultura**: In: A Sociedade em rede. São Paulo: Paz e Terra, v. 1, 2000.

CASTELLS, Manuel. **Redes de indignação e esperança**: Movimentos Sociais na era da internet. Rio de Janeiro: Zahar, 2013.

CODELABS. GOOGLE. 2020. Disponível em: <https://codelabs.developers.google.com/codelabs/tfjs-training-regression>. Acesso em: 17 Jun. 2020.

CORDI, Cassiano. **Para filosofar**. São Paulo: Scipione, 2000.

COX, Michael; ELLSWORTH, David. **Application-Controlled Demand Paging for Out-of-Core Visualization**. NASA Ames Research Center, 1997. Disponível em: <http://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf>. Acesso em: 15 Nov. 2018.

CUNHA, Maria Alexandra; GUISE, Mônica Steffen. **Dados abertos nos municípios, estados e governo federal brasileiros**. São Paulo: Fundação Getúlio Vargas, v. 3, 2015. (Projeto Democracia Digital).

DAVENPORT, Thomas. **Big data no trabalho: Derrubando mitos e descobrindo oportunidades**. Tradução Cristina Yamagami. Rio de Janeiro: Elsevier, 2014.

DEAN, Jeffrey; GHEMAWAT, Sanjay. **MapReduce: Simplified Data Processing on Large Clusters. San Francisco**. In: OSDI'04: SIXTH SYMPOSIUM ON OPERATING SYSTEM DESIGN AND IMPLEMENTATION, 2004. Disponível em: <https://ai.google/research/pubs/pub62>. Acesso em: 15 Nov. 2018.

DYER, Lin J C. **Data-intensive text processing with MapReduce**: Synthesis Lectures on Human Language Technologies. 2010, p. 1-177.

EXPRESS. **Express**. 2020. Disponível em: <https://expressjs.com/>. Acesso em: 2 Jun. 2020.

G1. **PF faz operação contra fraude na compra de merenda em 27 cidades de SP**. 2019. Disponível em:

<https://g1.globo.com/sp/ribeirao-preto-franca/noticia/2019/11/26/pf-faz-operacao-contra-fraude-na-compra-de-merenda-em-cidades-de-sp.ghtml>. Acesso em: 17 Jun. 2020.

GALDINO, Natanael. **Big Data: Ferramentas e Aplicabilidade**. 2016. Disponível em: <https://www.aedb.br/seget/arquivos/artigos16/472427.pdf>. Acesso em: 15 Nov. 2018.

GAÚCHAZH. **Mais de um milhão de crianças consomem merenda escolar no Estado**. 2014. Disponível em: <https://gauchazh.clicrbs.com.br/educacao-e-emprego/noticia/2014/03/mais-de-um-milhao-de-criancas-consomem-merenda-escolar-no-estado-cj5viv9zm0dfoxbj0mbnw4p4v.html>. Acesso em: 17 Jun. 2020.

GENRO, Tarso; SOUZA, Ubiratam. **Orçamento Participativo: A Experiência de Porto Alegre**. São Paulo: Fundação Perseu Abramo, 2001.

GRUPO CIBERATIVISMO. **Ciberativismo: o que é?**. 2016. Disponível em: <https://medium.com/@ciberativismo/ciberativismo-o-que-%C3%A9-b7e9b3c4f4ef>. Acesso em: 15 Nov. 2018.

HADOOP. **What Is Apache Hadoop?**. 2018. Disponível em: <http://hadoop.apache.org/>. Acesso em: 15 Nov. 2018.

IBGE. **API de serviço de dados**. 2020. Disponível em: <https://servicodados.ibge.gov.br/api/docs>. Acesso em: 2 Jun. 2020.

IONIC. **Core Concepts**. 2018. Disponível em: <https://ionicframework.com/docs/intro/concepts/>. Acesso em: 15 Nov. 2018.

IVANOV, Todor; KORFIATIS, Nikolaos; ZICARI, Roberto V. **On the inequality of the 3V's of Big Data Architectural Paradigms: A case for heterogeneity. Chair For Databases And Information Systems: Institute for Informatics and Mathematics**. Goethe University Frankfurt, 2013. Disponível em: <http://www.bigdata.uni-frankfurt.de/2013/11/inequality-3vs-big-data-architectural-paradigms-case-heterogeneity/>. Acesso em: 15 Nov. 2018.

LANEY, Doug. **3D data management: controlling data volume, velocity and variety. MetaGroup**. 2011. Disponível em: <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>. Acesso em: 15 Nov. 2018.

LEMONS, Ronaldo. **Ciberativismo**. 2014. Disponível em: <https://facom.ufba.br/ciberpesquisa/lemons/ciberativismo.pdf>. Acesso em: 25 Nov. 2018.

LEMONS, Ronaldo; STEIBEL, Fábio; SOUZA, Carlos Affonso. **Tecnologia, redes sociais e o combate à corrupção no Brasil**. Disponível em: https://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/caderno_lei_anticorrupcao_site.pdf. Acesso em: 15 Nov. 2018.

LÜCHMANN, Lígia Helena Hahn. **Possibilidades e limites da democracia deliberativa: a experiência do orçamento participativo de Porto Alegre**. Campinas,

2002. Tese (Doutorado) - Universidade Estadual de Campinas, Instituto de Filosofia e Ciências Humanas.

MACHADO, Murilo B. **Por dentro dos Anonymous Brasil: poder e resistência na sociedade de controle**. 2013. Dissertação (História) - Fundação Universidade Federal do Abc.

MARTINS, Maria das Graças. **A tecnologia da informação inovando a transparência nas universidades públicas**. Florianópolis: UFSC, 2014.

MAYER SCHÖNBERGER, Viktor; CUKIER, Kenneth. **Big data: como extrair volume, variedade, velocidade e valor da avalanche de informação cotidiana**. Tradução Paulo Polznoff Junior. Rio de Janeiro: Elsevier, 2013. 163 p.

MORAES, Dênis de (Org.). **Sociedade Mdiatizada**. Rio de Janeiro: Mauad, 2006.

MYSQL. **Estudos de caso**. 2020. Disponível em: <https://www.mysql.com/why-mysql/case-studies/>. Acesso em: 2 Jun. 2020.

OGP. OPEN GOVERNMENT PARTNERSHIP. **Declaração de Governo Aberto**. 2011. Disponível em: <http://www.governoaberto.cgu.gov.br/central-de-conteudo/documentos/arquivos/declaracao-governo-aberto.pdf>. Acesso em: 15 Nov. 2018.

OLIVEIRA, Janaína de; CATTANI, Michele Prass Scheffle. **O governo aberto como meio de efetivação da democracia participativa**. 2017. Disponível em: <http://coral.ufsm.br/congressodireito/anais/2017/3-5.pdf>. Acesso em: 15 Nov. 2018.

OPEN DATA. **A world where knowledge creates power for manu, not the few**. 2018. Disponível em: <https://okfn.org/>. Acesso em: 15 Nov. 2018.

OPEN KNOWLEDGE FOUNDATION. **O que são Dados Abertos?**. 2017. Disponível em: http://opendatahandbook.org/guide/pt_BR/what-is-open-data/. Acesso em: 15 Nov. 2018.

OPERAÇÃO SERENATA DE AMOR. **Inteligência artificial para controle social da administração pública**. 2016. Disponível em: <https://serenatadeamor.org/>. Acesso em: 15 Nov. 2018.

PAIVA, Eduardo de; REVOREDO, Kate. **Big Data e Transparência: Utilizando funções de Mapreduce para incrementar a transparência dos Gastos Públicos**. 2016. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2016/004.pdf>. Acesso em: 15 Nov. 2018.

PEREIRA, Ricardo A. de Castro; CAMPOS, Francisco de Assis Oliveira. **Corrupção e ineficiência no Brasil: Uma análise de equilíbrio geral**. **Estud. Econ.**. São Paulo, 2016. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-41612016000200373&lng=pt&nrm=iso. Acesso em: 15 Nov. 2018.

PORTAL BRASILEIRO DE DADOS ABERTOS. **O que são dados abertos?**. 2018. Disponível em: <http://dados.gov.br/pagina/dados-abertos>. Acesso em: 15 Nov. 2018.

PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL. **Sobre o Portal: O que você encontra no Portal**. 2018. Disponível em: <http://www.portaltransparencia.gov.br/sobre/o-que-voce-encontra-no-portal>. Acesso em: 15 Nov. 2018.

PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL. **API de dados**. 2018. Disponível em: <http://www.portaltransparencia.gov.br/api-de-dados>. Acesso em: 15 Nov. 2018.

PORTAL DA TRANSPARÊNCIA DO GOVERNO FEDERAL. **O que é e como funciona**. 2018. Disponível em: <http://www.portaltransparencia.gov.br/sobre/o-que-e-e-como-funciona>. Acesso em: 15 Nov. 2018.

PREFEITURA DE TORRES. **Portal da Transparência**. 2018. Disponível em: <https://torres.rs.gov.br/portal-da-transparencia/>. Acesso em: 15 Nov. 2018.

RAYMOND, Eric S. **The New Hacker's dictionary**. MIT Press, 2011.

RIBEIRO, Manuella Maia. **O uso das tecnologias de informação e comunicação como ferramentas de combate à corrupção**: comprasnet e o portal da transparência. São Paulo, 2016.

SCUDERE, Leonardo. **Risco digital na web 3.0**. Rio de Janeiro: Elsevier, 2015. 154 p.

SILVA, Cristiano de Assis; RIBEIRO, Gloria Teresa Duarte Bernardes. **Alimentação Escolar e o Processo Ensino Aprendizagem**. 2011. Disponível em: <https://www.lusocapixaba.edu.br/reieduc/Edicao-03/artigo-01-alimentacao-escolar-e-oprocesso-ensino-aprendizagem-cristiano-de-assis-&-gloria-teresa-p.-08-a-18.pdf>. Acesso em: 15 Nov. 2018.

SILVEIRA, Sergio Amadeu. Ciberativismo, cultura hacker e o individualismo colaborativo. **Revista da USP**, São Paulo, n. 86, 2010.

SOUZA, Glenda Rany Máximo de. **Big Data**: fundamentos, metodologias e tecnologias. Brasília, f. 63, 2017. Monografia (Bacharelado em Biblioteconomia) - Universidade de Brasília.

SÁ, Renata Yamamoto Giovani de. **A importância do Portal da Transparência na Administração Pública**. 2013. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/5784/1/PB_GP_III_2014_10.pdf. Acesso em: 15 Nov. 2018.

TAURION, Cezar. **Big data**. Rio de Janeiro: Brasfort, 2015. 170 p.

TEIXEIRA, Alberto. **Guia da cidadania para a transparência: prevenção para a corrupção**. Ceará: Fundação Konrad Adenauer, 2006.

TENSORFLOW. **TensorFlow.js guide**. 2020. Disponível em: <https://www.tensorflow.org/js/guide>. Acesso em: 2 Jun. 2020.

TRANSPARENCY INTERNACIONAL. **Corruption Perceptions Index 2015**. 2015.

Disponível em: <https://www.transparency.org/cpi2015>. Acesso em: 15 Nov. 2018.

TRIBUNAL DE CONTAS DA UNIÃO. **5 motivos para a abertura de dados na Administração Pública**. 2016. Disponível em: <http://portal3.tcu.gov.br/portal/pls/portal/docs/2689107.PDF>. Acesso em: 15 Nov. 2018.

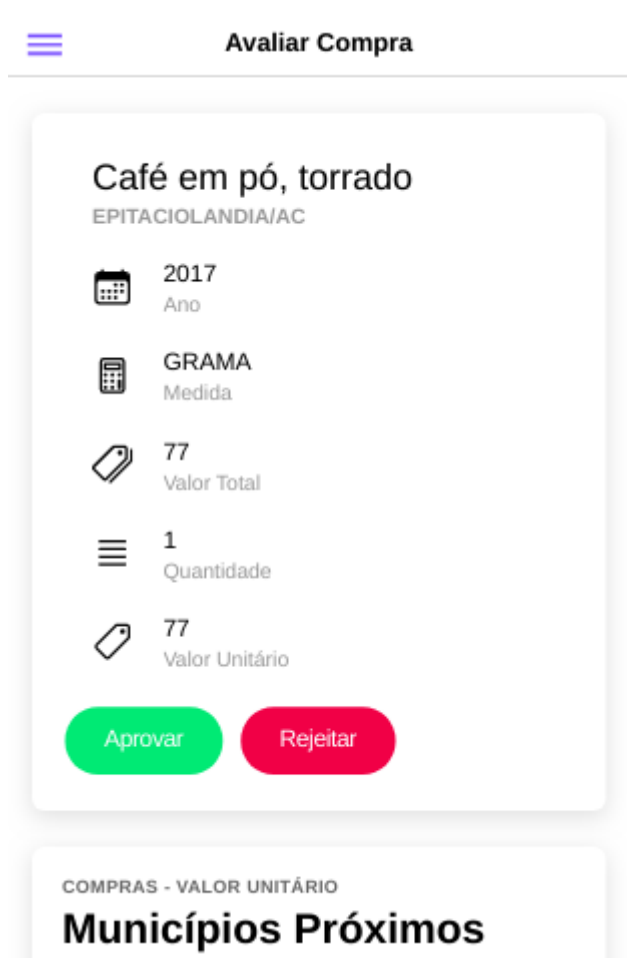
VARGAS, Cristina Ribas. **Democracia Participativa e Desenvolvimento Humano nos Municípios Brasileiros**: uma Análise a partir de Amartya Sen. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Administração, Contabilidade e Economia, 2007.




VEGH, S. Classifying forms of online activism: the case of cyberprotests against the World Bank. *In*: MCCAUGHEY, M; AYERS, M.D.. **Cyberactivism**: online activism in theory and practice. London: Routledge, 2003.

WEINSTEIN, Jeremy; GOLDSTEIN, Joshua. **The Benefits of a Big Tent**: Opening Up Government in Developing Countries - A Response to Yu & Robinson's The New Ambiguity of "Open Government". *UCLA Law Review Disclosure*, v. 60, 2012, p. 40-48.

WORDPRESS. **WordPress**. 2020. Disponível em: https://codex.wordpress.org/pt-br:P%C3%A1gina_Inicial. Acesso em: 2 Jun. 2020.

APÊNDICE A — TELAS DO APLICATIVO FRONTEND



	Minhas Avaliações
	<p>Pimenta, fresca, diversos tipos EPITACIOLANDIA - AC Compra realizada em 2017, de 0.5 QUILO por R\$ 4,00 cada um</p>
	<p>Fermento químico, em pó BUJARI - AC Compra realizada em 2017, de 0.5 QUILO por R\$ 420,00 cada um</p>

**Avaliar Compra**

GERADA PELO APLICATIVO

Estimativa de Preço

De acordo com todas as compras presentes no Merenda Democrática, a estimativa de preço para esse produto é de:

R\$ 2,92

COMPRAS - VALOR UNITÁRIO


Municípios Próximos

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 2,70
PORTO ACRE
Açúcar

R\$ 2,70

 **Avaliar Compra**

COMPRAS - VALOR UNITÁRIO

Municípios Próximos

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 2,70
PORTO ACRE
Açúcar

R\$ 2,70
PORTO ACRE
Açúcar

R\$ 4,00
ACRELANDIA
Açúcar

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 4,00
ACRELANDIA

R\$ 4,00
ACRELANDIA
Açúcar

R\$ 4,00
ACRELANDIA
Açúcar

R\$ 4,00
ACRELANDIA
Açúcar

R\$ 3,80
PLACIDO DE CASTRO
Açúcar

R\$ 4,00
ACRELANDIA
Açúcar

O QUE OUTROS USUÁRIOS ACHAM

Comentários

Nenhum comentário encontrado para essa compra



Todas Avaliações

Pimenta, fresca, diversos tipos
EPITACIOLANDIA - AC
 Compra realizada em 2017, de 0.5 QUILO por R\$ 4,00 cada um

Fermento químico, em pó
BUJARI - AC
 Compra realizada em 2017, de 0.5 QUILO por R\$ 420,00 cada um



Contribuir com Preço

DIGA QUANTO CUSTA EM SUA CIDADE

Contribuir para o MerendaBot

Estado* 


Cidade* 

Produto*


Unidade 

Valor Unitário*

[Enviar Preço](#)




Merenda Bot

Tweets por @MerendaBot 


Hmm... Uma timeline vazia. Isso é estranho.


[Verificar por Tweets](#)

[Incorporar](#) [Ver no Twitter](#)



Indícios de Fraude

 **Pimenta, fresca, diversos tipos**
EPITACIOLANDIA - AC
Compra realizada em 2017, de 0.5 QUILO por R\$ 4,00 cada um


**Perfil**

Atualizar Perfil

Nome*
Administrador

Login*
admin

Endereço*

Data de Nascimento*
dd/mm/aaaa 

Email*
merendademocratica@gmail.com

Telefone*

Atualizar Perfil

TCC - SISTEMAS DE INFORMAÇÃO

Merenda Democrática

Email*

merendademocratica@gmail.com

Senha*

•••••

ENTRAR

APÊNDICE B — OBTENÇÃO DE COMPRAS DO PAINEL DE PREÇOS DO FNDE

Para provar os conceitos propostos foi utilizado compras oriundas do Painel de Preços do PNAE. Essas compras estão disponíveis através da Lei de Acesso à Informação (LAI). O Painel de Preços do Programa Nacional de Alimentação Escolar (PNAE) é uma ferramenta de divulgação de dados desenvolvida pelo FNDE para permitir o controle social da política pública do PNAE, ampliando assim a transparência e o acesso às informações por parte de milhões de brasileiros. Com isso, é possível acompanhar através do Painel ações dos gestores municipais e estaduais no uso dos recursos do fundo e a execução do PNAE nos estados e municípios brasileiros.

Os dados disponíveis na base de dados do Painel são oriundos da aquisição de gêneros alimentícios em ano fiscal anterior, a partir do ano de 2017. As informações presentes no Painel foram extraídas junto ao Sistema Nacional de Gestão de Prestação de Contas (SIGPC Contas Online), cadastradas de forma manual pelo próprio gestor. O Painel possui apenas dados das notas fiscais dos alimentos adquiridos com recursos federais, repassados a estados e municípios pelo FNDE em caráter suplementar, então sendo assim os municípios e estados realizam compras de gêneros alimentícios para a merenda escolar utilizando outras fontes de recursos que não as verbas do PNAE. Os dados de prestação de contas enviados para o SIGPC em formato XML, que estão declarados em unidades diferentes de peso e volume, não estão contemplados no painel já não é possível realizar uma comparação correta com os preços praticados entre as compras que possuem o mesmo produto.

Como acessar?

Os dados do Painel de Preços do Programa Nacional de Alimentação Escolar (PNAE) estão disponíveis para qualquer cidadão, que deverá entrar no site do FNDE e buscar o PNAE na lista de programas, acessando os dados ao clicar no banner “Painel de Preços Praticados pelo PNAE”.

Em seguida há quatro opções de acesso, com as informações sendo expostas de maneira diferente: Estatística de Compras, Preços praticados por produto ou Compras por fornecedor. Ao selecionar a opção Preços praticados por produto é carregado um mapa com os estados brasileiros disponíveis para seleção. Escolhendo o estado o cidadão começa a visualizar os valores das compras

praticados pelo Portal.

VOCÊ ESTÁ AQUI: PÁGINA INICIAL > PROGRAMAS > PNAE > CONSULTAS > PAINEL DE PREÇOS



Últimas notícias

Perguntas frequentes

Legislação

Liberação de recursos

Contatos

SOBRE O PROGRAMA

ÁREA PARA GESTORES

Painel de preços praticados pelo PNAE

[Facebook](#) [Twitter](#) [Twitter](#) [Curtir](#)

O que é?

O Painel de Preços do Programa Nacional de Alimentação Escolar (PNAE) é uma ferramenta de divulgação desenvolvida pelo FNDE para o controle social da política pública do PNAE.

O objetivo essencial do Painel de Preços é ampliar a transparência e o acesso às informações das ações governamentais a milhões de brasileiros.

Com isto, a gestão pública transparente do FNDE permitirá à sociedade colaborar no controle das ações dos gestores municipais e estaduais e acompanhar o uso dos recursos e a execução do PNAE nos estados e municípios brasileiros.

O que tem no Painel?

Os dados constantes do Painel foram gerados pelos gestores municipais e estaduais a partir da aquisição, já finalizada, de gêneros alimentícios em gestões anteriores, a partir do exercício de 2017.

Figura: Página inicial do Painel de Preços do PNAE

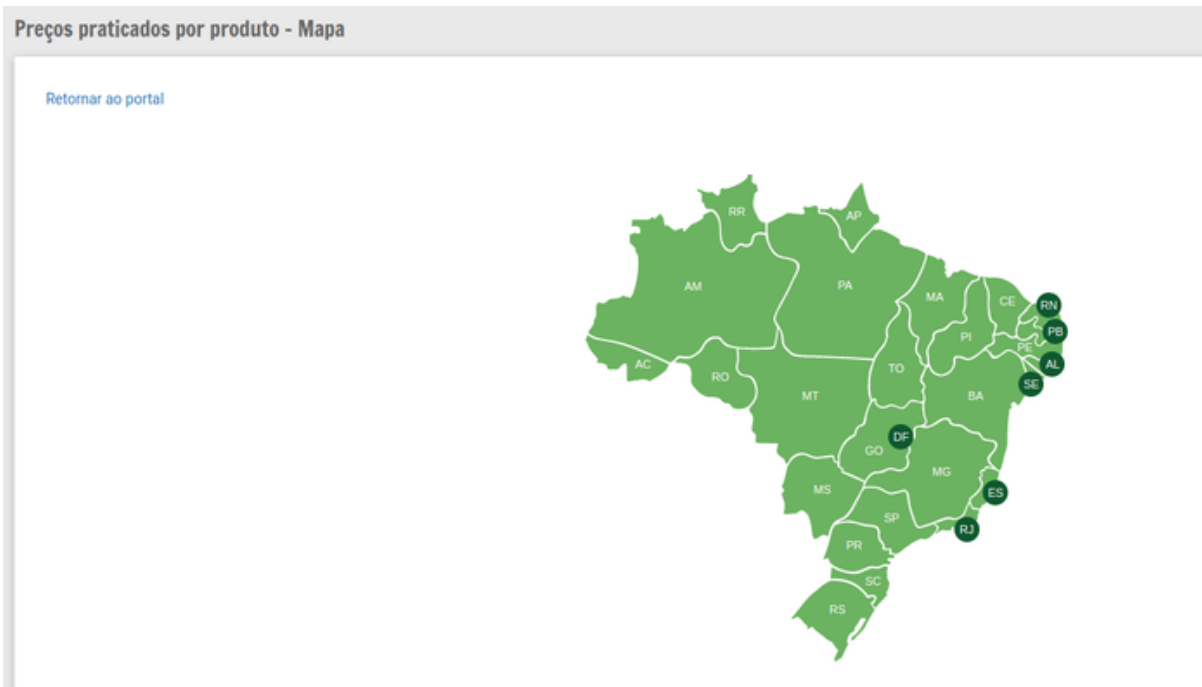


Figura: Mapa com os estados brasileiros no Painel de Prestação de Contas do PNAE

Preços praticados por produto - Município

[Voltar para o mapa](#)

2017 ▾ AC ▾ ACRELANDIA ▾

Mostrar 50 ▾ registros [Exportar XLS](#) Buscar:

Ano	UF	Município	Entidade Executora	Agricultura Familiar	Orgânico	Produto	Documento Despesa	Unidade Medida	Quantidade Comprada	Valor Unitário R\$	Valor Total R\$
2017	AC	ACRELANDIA	PREF MUN DE ACRELANDIA	SIM	NÃO	Abóbora	0411	QUILO	332,00	3,24	1.075,68
2017	AC	ACRELANDIA	PREF MUN DE ACRELANDIA	SIM	NÃO	Abóbora	0410	QUILO	182,00	3,24	589,68
2017	AC	ACRELANDIA	PREF MUN DE ACRELANDIA	NÃO	NÃO	Achocolatado em pó	000000303	GRAMA	72,00	3,00	216,00
2017	AC	ACRELANDIA	PREF MUN DE ACRELANDIA	NÃO	NÃO	Achocolatado em pó	3431	GRAMA	400,00	3,00	1.200,00

Figura: visão das contas de um município ao selecionar o estado no mapa

Os dados das compras estão disponíveis em um formato proprietário não aberto (XLS) e com a granularidade de município. Por isso é necessário realizar um processo de transformação para adicionar os dados na nossa base dados e permitir que processos de análise sejam automatizados.

Transformação dos dados

Para obter os dados do Painel de Preços do PNAE, foi desenvolvido um algoritmo Javascript que realiza a exportação do XLS e solicita o carregamento da nova cidade. Segue abaixo uma imagem demonstrando o código utilizado. Esse código encontra-se junto à aplicação da API REST.

```

var elements = document.getElementsByClassName('exportButton');
var x = document.getElementById('select_cidades');

async function wait() {
  return new Promise(function(resolve) {
    setTimeout(function() {
      elements[0].click();
      //console.log("vai resolver");
      resolve();
    }, 20000);
  });
}

(async function() {
  for (var i = 0; i < x.options.length; i++) {
    document.getElementById('select_cidades').selectedIndex = i;
    document.getElementById('select_cidades').dispatchEvent(new Event('change'));
    await wait();
    console.log('test')
  }
})();

```

Dessa forma é possível automatizar o processo de exportação dos dados das cidades. Entretanto a aplicação do FNDE apresenta bastante lentidão e travamento em algumas oportunidades, acarretando assim que o código proposto não funcione em 100% das vezes. O código proposto também pode ter problemas ao tratar com cidades com uma grande quantidade de dados, já que a aplicação do FNDE demanda o carregamento de todas as informações em tela para que assim seja possível exportar os dados da planilha com sucesso. Uma característica muito importante desse método de extração é que os dados ficam salvos em arquivos XLS. Para facilitar o acesso dessas informações por projetos futuros, o autor do presente trabalho disponibilizará as planilhas de forma pública a partir do repositório do código backend disponibilizando em seções anteriores do texto.

Entretanto, em alguns casos há informações que são claramente carregadas de forma errada na aplicação do Painel, notadamente no campo Entidade, onde no lugar de uma entidade local é comum encontrar entidades executoras da compra de outro estado ou até mesmo região do país, bem como valores diferentes ao executar o recarregamento da página. Esse erro aparentemente foi corrigido no período de finalização do trabalho, mas coloca em xeque a confiabilidade das informações disponibilizadas pelo Painel.

Para adicionar os arquivos XLS dentro do banco de dados foi realizado um tratamento nos arquivos, que consiste:

1. Conversão dos arquivos para formato .csv
2. União dos arquivos .csv resultantes do processo de conversão em um único arquivo
3. Importação deste arquivo único na tabela compras utilizando um SGBD.

Para realizar a conversão dos arquivos do formato xls para o formato .csv foi utilizado a solução gratuita “Online Converter Free”, disponível em: <https://onlineconvertfree.com/pt/convert-format/xls-to-csv/>, enquanto para realizar a união dos arquivos foi utilizada a ferramenta “Files Merge”, disponível em <https://www.filesmerge.com/pt/merge-csv-files>.

A importação do arquivo CSV resultante do processo de união foi realizada utilizando a ferramenta para gerenciamento phpMyAdmin, um aplicativo web de código aberto desenvolvido em PHP.

Prova de Conceito

Para o desenvolvimento da prova de conceito (POC) presente neste trabalho foram utilizados os municípios do estado do Acre. Os demais estados possuem os dados disponíveis no formato de arquivos XLS. Trabalhos futuros podem explorar algoritmos de tratamento e de conversão desses dados.

APÊNDICE C — EXECUÇÃO DA APLICAÇÃO UTILIZANDO TENSORFLOW.JS

Para treinar o modelo de rede neural e realizar as previsões de preços unitários foi utilizado TensorFlow.js, uma implementação do TensorFlow para a linguagem Javascript, que permite a execução de processos de machine learning dentro do navegador.

Produtos

Nossa base de dados possui uma gama bastante grande de produtos, por um dos pilares da merenda escolar ser a diversidade das refeições e também pelo fato de que os preenchimentos realizados pelos gestores não são padronizados. Para realizar a prova de conceito foi executado o treinamento do modelo para os cinco produtos com mais registros na base de dados.

Para encontrar esses cinco produtos foi utilizado uma consulta utilizando a seguinte instrução SQL:

```
SELECT count(id) as quantidade, c.produto FROM `compras` c GROUP BY produto ORDER BY quantidade DESC LIMIT 10
```

A execução da consulta retornou o seguinte resultado abaixo:

✓ A mostrar registros de 0 - 9 (10 total, A consulta demorou 0,0352 segundos.)

```
SELECT count(id) as quantidade, c.produto FROM `compras` c GROUP BY produto ORDER BY quantidade DESC LIMIT 10
```

+ Opções

	quantidade	produto
<input type="checkbox"/> Edita Copiar Apagar	1074	Polpa de frutas, diversos sabores, congelada
<input type="checkbox"/> Edita Copiar Apagar	884	Banana
<input type="checkbox"/> Edita Copiar Apagar	660	Biscoito salgado
<input type="checkbox"/> Edita Copiar Apagar	652	Açúcar
<input type="checkbox"/> Edita Copiar Apagar	647	Carne bovina de 1ª, resfriada/congelada
<input type="checkbox"/> Edita Copiar Apagar	633	Macarrão de trigo, com ou sem ovos, fresco/seco
<input type="checkbox"/> Edita Copiar Apagar	602	Leite de vaca integral, em pó
<input type="checkbox"/> Edita Copiar Apagar	577	Frango inteiro, resfriado/congelado
<input type="checkbox"/> Edita Copiar Apagar	544	Biscoito doce
<input type="checkbox"/> Edita Copiar Apagar	538	Café em pó, torrado

Marcar todos Com os seleccionados: Edita Copiar Apagar Exportar

Operações resultantes das consultas

Os cinco produtos mais presentes nas compras de nossa base de dados são:

- Polpa de frutas, diversos sabores, congelada
- Bana
- Biscoito salgado
- Açúcar
- Carne bovina de 1ª, resfriada/congelada

O processo agora é o mesmo para todos os cinco produtos, então realizaremos a demonstração com o produto “Açúcar”.

Treinamento do modelo com o produto Açúcar

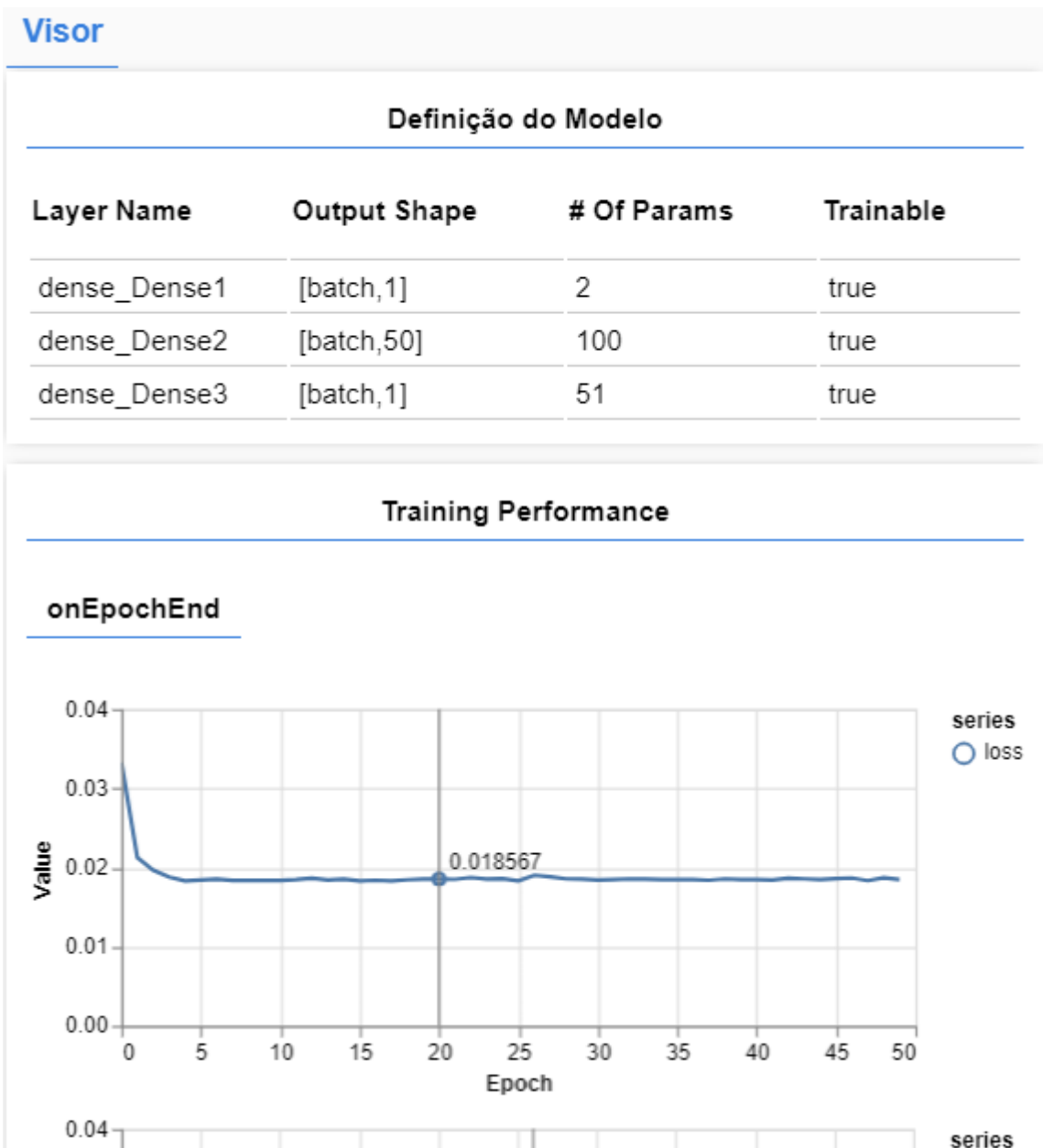
Para realizar o treinamento com o produto Açúcar, foi executado uma consulta na base de dados e exportado com as compras onde o produto é “Açúcar” e a unidade de medida está em quilos. Para extrair os dados de nossa base de dados foi realizado a seguinte consulta:

```
SELECT * FROM `compras` WHERE produto = 'Açúcar' AND
```

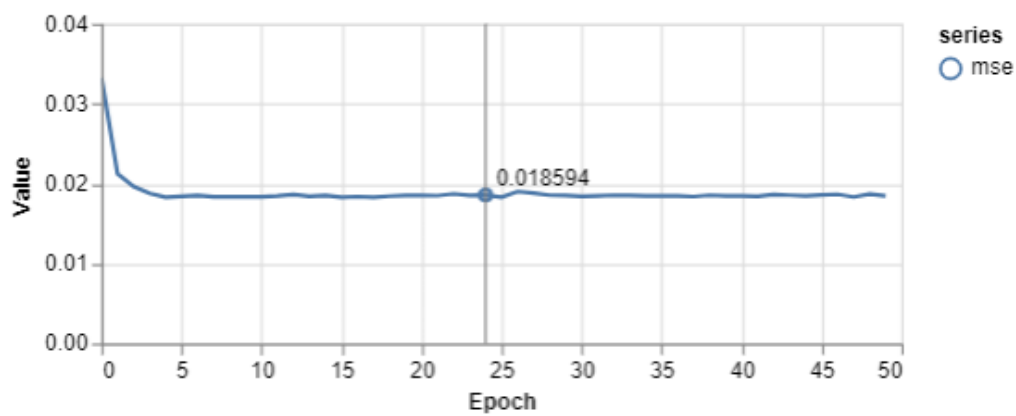
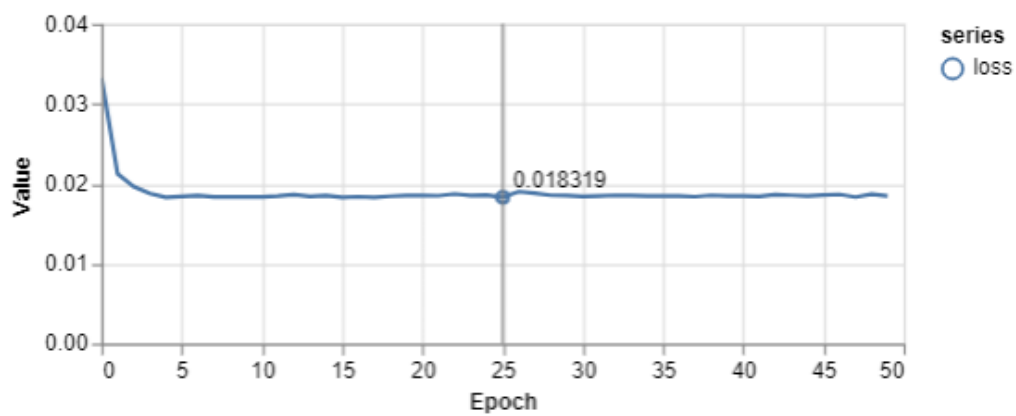
unidade_medida = 'QUILO';

Os resultados da consulta foram exportados para JSON e salvos em um arquivo que foi entrada para o treinamento do nosso modelo.

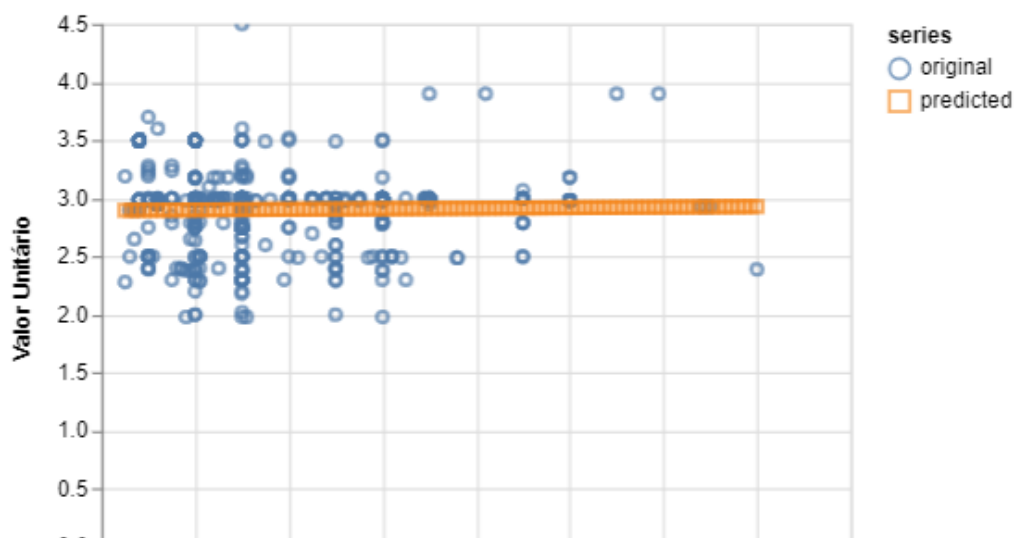
Segue imagens da execução do algoritmo. Detalhes técnicos foram especificados na seção 6.3 do trabalho.



onEpochEnd



Previsões do Modelo vs Valores Originais



A execução do modelo gerou as previsões em um Array de objetos que foram salvos em um arquivo .json.

```

predictedPoints ▼ Array(100) ⓘ
▶0: {x: 5, y: 2.8976564407348633, produto: "Açucar", uf: "AC"}
▶1: {x: 6.363636493682861, y: 2.897982120513916, produto: "Açucar", uf: "AC"}
▶2: {x: 7.727272987365723, y: 2.8983073234558105, produto: "Açucar", uf: "AC"}
▶3: {x: 9.090909004211426, y: 2.898632526397705, produto: "Açucar", uf: "AC"}
▶4: {x: 10.454545974731445, y: 2.8989577293395996, produto: "Açucar", uf: "AC"}
▶5: {x: 11.818181991577148, y: 2.8992834091186523, produto: "Açucar", uf: "AC"}
▶6: {x: 13.181818008422852, y: 2.8996083736419678, produto: "Açucar", uf: "AC"}
▶7: {x: 14.545454025268555, y: 2.8999338150024414, produto: "Açucar", uf: "AC"}
▶8: {x: 15.909090995788574, y: 2.900259494781494, produto: "Açucar", uf: "AC"}
▶9: {x: 17.272727966308594, y: 2.9005842208862305, produto: "Açucar", uf: "AC"}
▶10: {x: 18.636363983154297, y: 2.900909662246704, produto: "Açucar", uf: "AC"}
▶11: {x: 20, y: 2.901235342025757, produto: "Açucar", uf: "AC"}
▶12: {x: 21.363637924194336, y: 2.9015605449676514, produto: "Açucar", uf: "AC"}
▶13: {x: 22.72727394104004, y: 2.901885986328125, produto: "Açucar", uf: "AC"}
▶14: {x: 24.090909957885742, y: 2.9022116661071777, produto: "Açucar", uf: "AC"}
▶15: {x: 25.454545974731445, y: 2.9025368690490723, produto: "Açucar", uf: "AC"}
▶16: {x: 26.81818199157715, y: 2.902862310409546, produto: "Açucar", uf: "AC"}
▶17: {x: 28.18181800842285, y: 2.9031877517700195, produto: "Açucar", uf: "AC"}
▶18: {x: 29.545454025268555, y: 2.903512954711914, produto: "Açucar", uf: "AC"}
▶19: {x: 30.909088134765625, y: 2.903838634490967, produto: "Açucar", uf: "AC"}
▶20: {x: 32.27272415161133, y: 2.9041643142700195, produto: "Açucar", uf: "AC"}
▶21: {x: 33.63636016845703, y: 2.904489517211914, produto: "Açucar", uf: "AC"}
▶22: {x: 34.999996185302734, y: 2.9048149585723877, produto: "Açucar", uf: "AC"}
▶23: {x: 36.36363220214844, y: 2.9051401615142822, produto: "Açucar", uf: "AC"}
▶24: {x: 37.72726821899414, y: 2.905465602874756, produto: "Açucar", uf: "AC"}
▶25: {x: 39.090904235839844, y: 2.9057912826538086, produto: "Açucar", uf: "AC"}
▶26: {x: 40.45454025268555, y: 2.9061167240142822, produto: "Açucar", uf: "AC"}
▶27: {x: 41.818180084228516, y: 2.906442165374756, produto: "Açucar", uf: "AC"}
▶28: {x: 43.18181610107422, y: 2.9067673683166504, produto: "Açucar", uf: "AC"}
▶29: {x: 44.54545593261719, y: 2.907092809677124, produto: "Açucar", uf: "AC"}
▶30: {x: 45.90909194946289, y: 2.9074182510375977, produto: "Açucar", uf: "AC"}
▶31: {x: 47.272727966308594, y: 2.9077439308166504, produto: "Açucar", uf: "AC"}
▶32: {x: 48.63636779785156, y: 2.908069133758545, produto: "Açucar", uf: "AC"}
▶33: {x: 50.000003814697266, y: 2.9083950519561768, produto: "Açucar", uf: "AC"}
▶34: {x: 51.363643646240234, y: 2.9087202548980713, produto: "Açucar", uf: "AC"}
▶35: {x: 52.72727966308594, y: 2.909045457839966, produto: "Açucar", uf: "AC"}
▶36: {x: 54.090919494628906, y: 2.9093711376190186, produto: "Açucar", uf: "AC"}
▶37: {x: 55.45455551147461, y: 2.909696578979492, produto: "Açucar", uf: "AC"}
▶38: {x: 56.81819534301758, y: 2.910022020339966, produto: "Açucar", uf: "AC"}
▶39: {x: 58.18183135986328, y: 2.9103474617004395, produto: "Açucar", uf: "AC"}
▶40: {x: 59.54547119140625, y: 2.910673141479492, produto: "Açucar", uf: "AC"}
▶41: {x: 60.90910720825195, y: 2.910998821258545, produto: "Açucar", uf: "AC"}
▶42: {x: 62.272743225097656, y: 2.9113240242004395, produto: "Açucar", uf: "AC"}
▶43: {x: 63.636383056640625, y: 2.911649465560913, produto: "Açucar", uf: "AC"}
▶44: {x: 65.00001525878906, y: 2.9119749069213867, produto: "Açucar", uf: "AC"}
▶45: {x: 66.36366271972656, y: 2.9123005867004395, produto: "Açucar", uf: "AC"}

```

Esse arquivo salvo foi convertido utilizando a ferramenta gratuita “JSON to CSV Online” disponível no link: <https://json-csv.com/>. Após a execução da conversão o arquivo CSV resultante foi importado no banco de dados na tabela “Previsões”.

Arquivos intermediários e código-fonte do processo estão disponível dentro do projeto do backend, na pasta “tensorflow-preview”.

Democracia Participativa e dados abertos: um MVP para o cidadão acompanhar a compra da merenda escolar

Marcelo Henrique Hoffmann Ecker

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
(UFSC) Florianópolis – SC – Brazil

ABSTRACT. *Today, the media and technology are practically ubiquitous in our lives. Since the origin of computers, notebooks, portable devices and personal assistants, the dissemination of information reaches its peak in the figure of the smartphone. On the other hand, we have government agencies that are starting to make information of interest to citizens available on their transparency portals. In the midst of this context, the citizen has in the participative democracy the opportunity to contribute actively in the government, inspecting the actions and proposing improvements for the society. However, information of public interest is not always available in a comprehensible way to the citizen or even for the execution of algorithms, so tools are needed to assist in the search and understanding of the data, as well as to facilitate their performance as citizens before results found. The present work develops an extensive bibliographic review on the subject, and from this it builds a technological ecosystem to meet these needs. As a case study, we chose to monitor government purchases of school meals. The developed application implements a REST API to consult the purchases made. This information is made available through a mobile application where the citizen can assess signs of fraud. To assist citizens in this task, purchases from neighboring municipalities are also informed and an estimate generated through a machine learning process with the registered data. The integration of user reviews with the social network Twitter was also developed, bringing purchases even closer to people who do not want to use the application.*

RESUMO. *Atualmente, os meios de comunicação e a tecnologia são praticamente onipresentes em nossa vida. Desde a origem dos computadores, notebooks, dispositivos portáteis e assistentes pessoais, a disseminação da informação chega ao seu ápice na figura do smartphone. De outro lado possuímos órgãos governamentais que estão iniciando a disponibilização de informações de interesse ao cidadão em seus portais da transparência. Em meio a este contexto, o cidadão tem na democracia participativa a oportunidade de contribuir de forma ativa no governo, fiscalizando as*

ações e propondo melhorias para a sociedade. Entretanto, nem sempre as informações de interesse público estão disponibilizadas de forma compreensível ao cidadão ou mesmo para a execução de algoritmos, portanto necessita-se de ferramentas que o auxiliem na busca e compreensão dos dados, bem como facilitem a sua atuação diante dos resultados encontrados. O presente trabalho desenvolve uma extensa revisão bibliográfica sobre o assunto, e a partir desta constrói um ecossistema tecnológico para atender estas necessidades. Como estudo de caso optou-se pelo monitoramento de compras governamentais de merenda escolar. A aplicação desenvolvida implementa uma API REST para consultas das compras realizadas. Essa informação é disponibilizada através de um aplicativo mobile onde o cidadão pode avaliar indícios de fraude. Para auxiliar os cidadãos nessa tarefa também são informadas as compras dos municípios vizinhos e uma estimativa gerada através de um processo de machine learning com os dados cadastrados. Também foi desenvolvido a integração das avaliações dos usuários com a rede social Twitter, trazendo as compras ainda mais próximas das pessoas que não queiram utilizar o aplicativo.

1. Introdução

Nos dias de hoje, os meios de comunicação e a tecnologia são praticamente onipresentes em nossa vida. Desde a origem dos computadores, notebooks, dispositivos portáteis e assistentes pessoais, a disseminação da informação chega ao seu ápice na figura do smartphone. Nesse cenário estamos conectados com um manancial de tamanho quase infinito de informações a poucos cliques da ponta de nossos dedos.

Com a quantidade de elementos tecnológicos disponíveis, possuímos uma série de oportunidades para resolver os nossos problemas cotidianos, sejam eles individuais e de cunho prático ou sociais e de cunho filosófico. (TEIXEIRA, 2006) A disseminação irreversível e progressiva, dia a dia, da tecnologia da informação e a utilização incondicional dos sistemas de informação automatizados em todos os níveis da administração pública direta ou indiretamente atua como motivador para que problemas que envolvam a máquina pública tenham suas soluções estabelecidas por meios tecnológicos (MARTINS, 2014).

É notório que um dos maiores problemas que o país possui atualmente é a má-administração e gestão da máquina pública. Caminha (2014) ressalta que o estado possui um orçamento bastante amplo e acaba, até mesmo por falta de ferramentas concisas que possam estabelecer um suporte ou devido a má-fé de entes públicos envolvidos, gastando um valor muito mais alto do que os preços de mercado em pregões públicos, conhecido também como licitações. Fatos dessa natureza ocorrem em praticamente todos os segmentos que vão desde obras de infra-estrutura (pontes, estradas, postos de saúde, etc), material de custeio, remédios e a compra dos itens da merenda escolar. A má-gestão da máquina pública abre espaço para que outro mal se alastre, a corrupção. Tendo uma estrutura que não é eficiente e

transparente, gestores mal-intencionados possuem a oportunidade de se beneficiar de forma pessoal da sua atuação profissional. Pela legislação existe corrupção quando há o desvio por parte de um funcionário público dos deveres formais do cargo devido à busca de uma recompensa para si ou para os outros; a definição de mercado considera corrupção a utilização do cargo público pelo seu ocupante como uma forma de maximizar a sua renda pessoal. Definições à parte, fato é que a corrupção é um dos principais motivos que colaboram para que o estado se torne ineficiente e que o país não consiga se tornar um país desenvolvido, com distribuição de renda e qualidade de vida para todos os seus habitantes (CAMINHA, 2004).

Entretanto já vigoram medidas que visam tornar a administração pública mais moderna e transparente. De acordo com a Constituição Federal (1993) em seu art.5º, XXXIII, todos os cidadãos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidade ressalvadas informações que são confidenciais e remetem à questões que são imprescindíveis à segurança da sociedade e do Estado. A principal ferramenta que atua na divulgação das informações de gestão pública é o Portal da Transparência (RIBEIRO, 2016) O Portal da Transparência é uma iniciativa da Controladoria Geral da União (CGU), lançado em novembro de 2004, para assegurar a boa e correta aplicação dos recursos públicos. O objetivo do desenvolvimento do portal é aumentar a transparência da gestão pública, permitindo que cidadãos acompanhem e ajudem a fiscalizar como o dinheiro público está sendo utilizado (MARTINS, 2004).

A educação básica é uma das maiores responsabilidades que o estado possui. Está previsto na constituição em seu artigo 6º que a educação é um direito garantido de todos os brasileiros e pode e deve ser exigida de seus órgãos competentes quando esse direito for violado ou desrespeitado (CONSTITUIÇÃO, 1989). Um dos aspectos básicos que garantem o pleno desenvolvimento das atividades de ensino-aprendizagem é a oferta de merenda escolar para todos os estudantes matriculados na rede pública. De acordo com Silva (2010), estudantes que se alimentam mostram melhor disposição, melhora no humor, no crescimento físico e no desenvolvimento sócio-psicomotor. O fornecimento de merenda escolar é responsabilidade do estado e não é raro encontrar denúncias de fraudes e irregularidades, noticiadas pelos mais diversos setores da imprensa (G1, 2019). Para tornar tudo ainda mais perverso, muitas crianças têm no momento da merenda escolar a única refeição do seu dia (GAÚCHAZH, 2014).

Pretende-se com esse trabalho realizar um estudo sobre medidas tecnológicas que auxiliam no combate a corrupção e na otimização e gestão das finanças públicas. Tal estudo será utilizado como base para desenvolvimento de uma aplicação que permite ao cidadão através da democracia participativa, analisar os preços dos itens de compra da merenda escolar obtidos através de bases de dados públicas, possibilitando assim a criação de indicadores de fraude. A aplicação também fornecerá alguns dados que possibilitam melhor avaliação do valor da merenda por parte do cidadão: compras similares realizadas por municípios vizinhos e valor previsto através de um processo de aprendizado de máquina (*machine learning*).

Dessa forma o objetivo geral deste trabalho é desenvolver uma ferramenta para que os cidadãos tenham acesso aos dados de compras da merenda escolar disponibilizados pelo Fundo Nacional de Desenvolvimento da Educação, no âmbito do PNAE (Plano Nacional de Alimentação Escolar), permitindo que os mesmos possam contribuir fiscalizando a compra da merenda escolar, buscando assim que o processo de compra possa se tornar mais barato, transparente e eficiente.

Os objetivos específicos são:

- Realizar uma revisão bibliográfica e apresentar possibilidades de tecnologias que permitam o desenvolvimento de aplicações para analisar o preço unitário de produtos presentes no Portal da Transparência de instituições públicas.
- Propor e desenvolver uma API REST que permite que o cidadão acesse de forma simplificada as compras da merenda escolar realizadas pelos municípios.
- Propor e desenvolver uma solução mobile aberta para obter avaliações dos cidadãos sobre as compras efetuadas pelas entidades públicas.
- Propor e desenvolver um modelo que prevê o preço do valor unitário esperado para determinadas compras de produtos na merenda escolar.
- Propor e desenvolver formas de disseminar essas informações, como integração com redes sociais para que o cidadão que não deseja instalar o aplicativo tenha acesso aos indícios de fraude.

2. Proposta do Trabalho

Tendo em vista as possibilidades que a sociedade da informação nos oferece, bem como o movimento de transparência nos dados governamentais (principalmente a Operação Serenata do Amor e o movimento Brasil.io), uma das propostas desse trabalho é aproximar o cidadão comum das informações que são disponibilizadas nas bases de dados de portais da transparência, tornando-as mais compreensíveis e mais intuitivas para o usuário. Dessa forma será desenvolvido um aplicativo MVP que permite que o cidadão tenha acesso as compras efetuadas pelas verbas do Fundo Nacional de Desenvolvimento da Educação (FNDE), no âmbito do Programa Nacional de Alimentação Escolar (PNAE), e possa realizar uma avaliação indicando se a compra possui ou não indícios de fraude. Esse recurso vai permitir que o cidadão exerça a democracia participativa, onde haverá um espaço para realizar comentários sobre as avaliações efetuadas, bem como interagir de forma anônima com outros usuários que realizaram a mesma avaliação. Essas avaliações serão de extrema importância para projetos futuros, onde poderão ser aplicados algoritmos de classificação para prever novos indícios de forma automatizada.

De forma a facilitar a realização da avaliação, o usuário terá acesso aos dados de compras realizadas por municípios vizinhos da compra em análise, bem como uma estimativa do preço esperado utilizando regressão linear e aprendizado de máquina (obtidos em nosso ambiente de machine learning construído utilizando o TensorFlow). Essas informações irão

colaborar na medida em serão dados concretos de comparação, propiciando algo para que o usuário fuja somente do uso do "feeling" na hora de avaliar uma compra.

Em outro cenário de uso de nosso aplicativo, o usuário pode preencher os preços atuais dos produtos alimentícios nos supermercados de sua cidade. Essa feature irá auxiliar na construção de uma base nacional de preços e será muito importante na realização de futuras análises de dados, podendo ser utilizada até mesmo como informação para auxiliar os usuários na tomada de decisão. Esse cenário de uso irá mitigar o problema de que hoje não há uma base nacional pública de preços de produtos, que possa ser facilmente acessada e utilizada como comparação para obter um preço esperado por unidade.

Buscando tornar os dados ainda mais transparentes para a população, será desenvolvido também uma integração com o Twitter que informa os indícios de fraude classificados por usuários da plataforma. Para dar suporte às operações realizadas no aplicativo será desenvolvido uma API REST (também chamada no presente trabalho de aplicação backend) que será central na conexão do aplicativo com as demais partes do ecossistema, conforme está demonstrado na figura 1

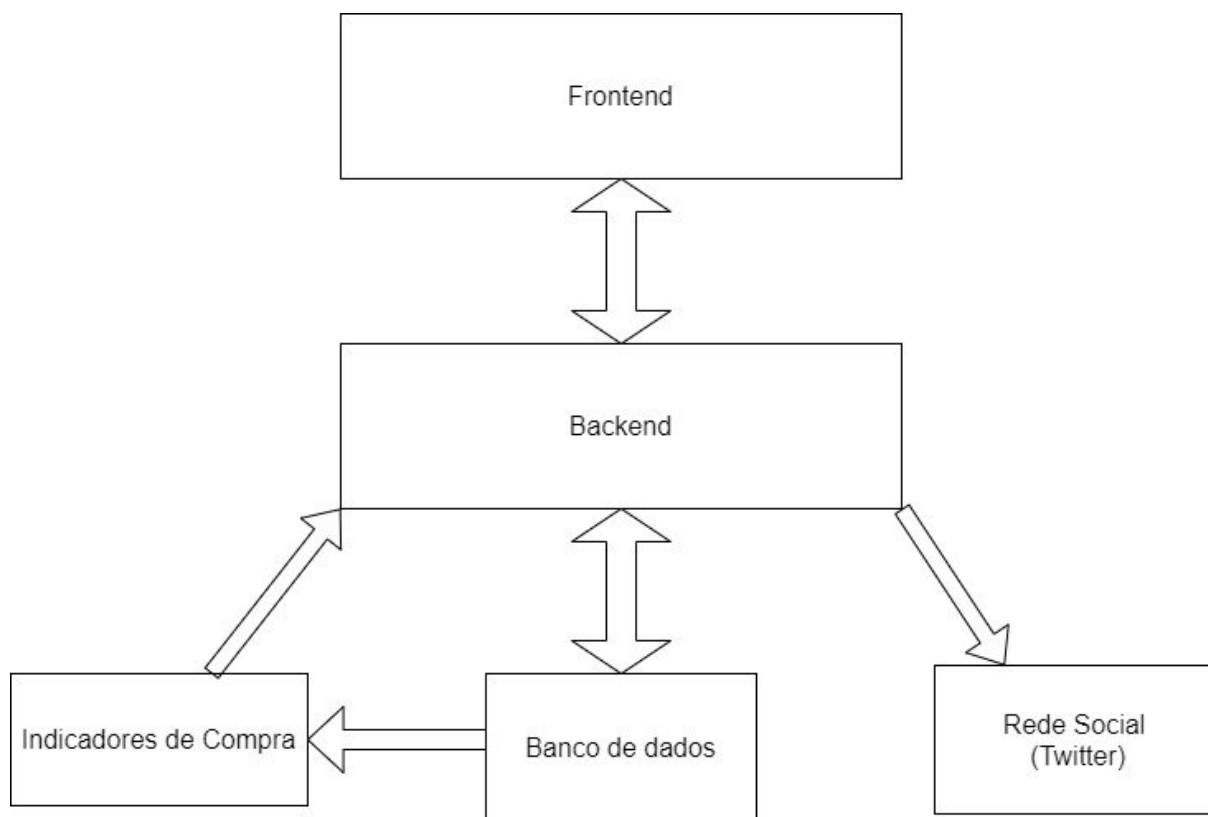


Figura 1. Arquitetura do MVP

O aplicativo poderá ser utilizado tanto em dispositivos Android e iOS após a realização do processo de build, mas para esse trabalho será utilizado a interface de desenvolvimento do Ionic, que permite a visualização do estado atual da aplicação através do navegador. O backend da aplicação estará disponível a partir de um servidor web, utilizando

Node.js e Express, fornecendo acesso aos endpoints da API através da porta 3333. O banco de dados da aplicação será o MySQL que estará rodando na porta 3306 da máquina, se comunicando com o backend de forma a prover os dados solicitados pela aplicação.

Para a obtenção de indicadores foram analisados dois cenários, sendo optado ao final pelo uso do TensorFlow.js. Alguns motivos levaram à essa adoção, como os destacados abaixo:

- Experiência do autor do presente trabalho na linguagem Javascript, facilitando assim a produtividade e o desenvolvimento do projeto.
- Não há necessidade de realizar a instalação de ambientes customizados para executar os testes (ao contrário do Hadoop), o TensorFlow roda simplesmente importando um arquivo Javascript e executando dentro do navegador.
- Características dos dados. Hadoop iria funcionar melhor se tivesse uma base nacional de preços para realizar a comparação.
- Modelos de execução de machine learning prontos. O TensorFlow.js possui APIs que abstraem a definição de modelos e de tensores, facilitando assim o processo de execução de aprendizagem de máquina.

3. Desenvolvimento

De acordo com a arquitetura e proposta apresentadas no capítulo 4, foi possível começar o desenvolvimento do MVP do Trabalho de Conclusão de Curso, batizado como “Merenda Democrática”. O desenvolvimento do MVP foi realizado em quatro etapas, sendo elas: aplicação backend (responsável pela API com os endpoints dando suporte às features do aplicativo, tanto para servir como conexão com os dados obtidos através do Painel de Prestação de Contas do FNDE e com as previsões realizadas pelo TensorFlow), aplicação frontend (o aplicativo para o usuário realizar a avaliação), ambiente TensorFlow.js para gerar os indicadores de preços e o bot para comunicar eventuais fraudes no Twitter.

A aplicação backend tem um papel fundamental no projeto do MVP, já que serve de suporte para as demais partes do sistema e é recurso essencial para o funcionamento das mesmas, já que possui endpoints que são utilizados por outros componentes do ecossistema. A aplicação backend possui endpoints que são utilizados tanto para acessar as informações presentes no banco de dados da aplicação, sejam elas avaliações realizadas pelo usuário, ou dados da compra da merenda escolar efetuados pelos 54 municípios e obtidos através do Painel de Preços do FNDE (Fundo Nacional de Desenvolvimento da Educação), bem como as previsões realizadas pelo TensorFlow.js. Há também métodos que controlam as postagens realizadas no perfil do Twitter deste trabalho, de forma a servir como uma espécie de “bot” alertando aos seguidores sobre possíveis indícios de fraude na compra da merenda escolar relatados por usuários que utilizam a aplicação. Os endpoints estão expostos utilizando a estrutura de API REST, padrão de comunicação entre aplicações mais utilizado do mercado, bem como toda a arquitetura de código está desenvolvida com base no Express, framework

Node.js que facilita a construção de APIs, padronizando assim o escopo da aplicação para que ela seja facilmente estendida em possíveis trabalhos futuros.

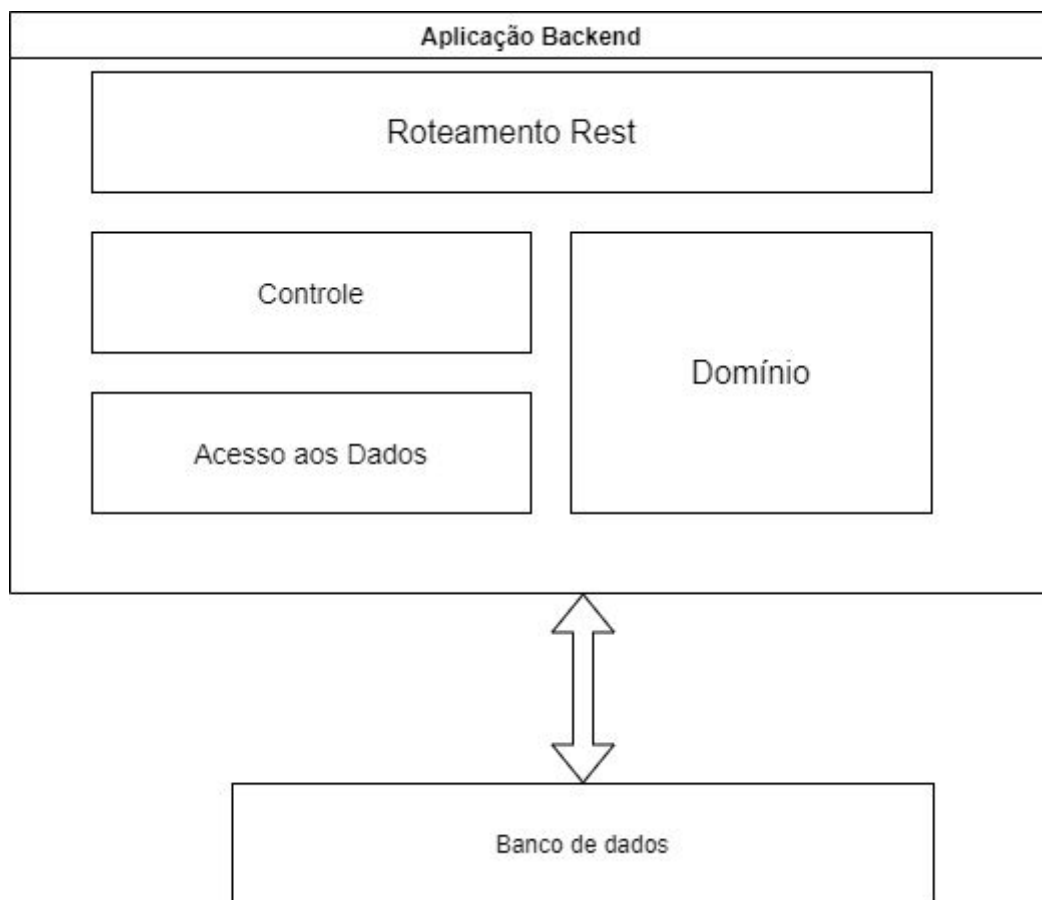


Figura 2. Arquitetura do backend

O desenvolvimento da aplicação frontend ocorreu após a definição clara da arquitetura e o desenvolvimento da API REST, de forma que o frontend (aplicação desenvolvida com Ionic) possa utilizar todas as potencialidades do backend. De forma geral, a aplicação frontend comunica-se com o backend para apresentar os dados necessários, estruturar e captar as interações do usuário na avaliação de compra de merenda escolar. A aplicação frontend possui também controle de acesso e permite aos usuários visualizar as avaliações de compra que foram realizadas na plataforma por ele e por outros usuários.

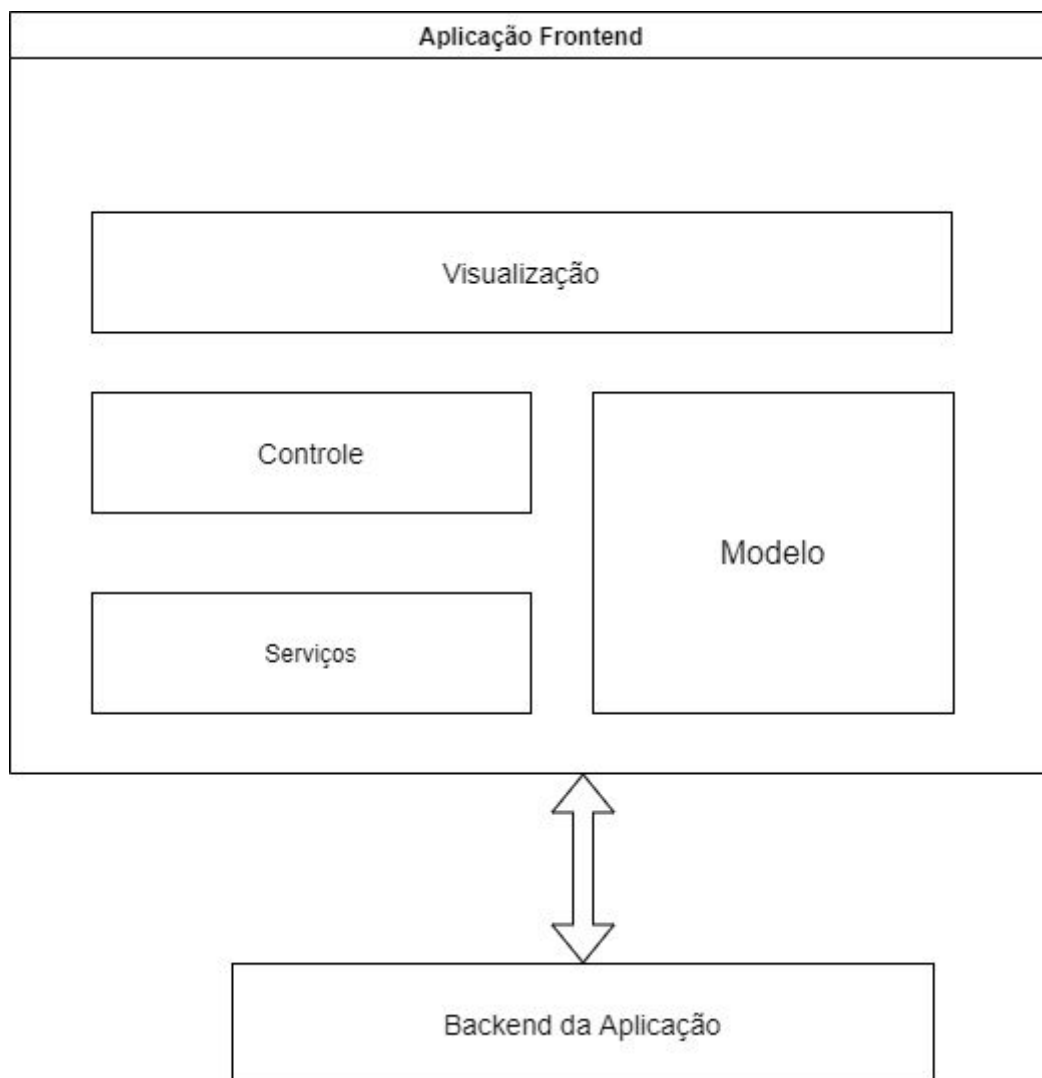


Figura 3 - Arquitetura do frontend

Para oferecer melhores indicadores para os usuários poderem avaliar as compras oriundas do Painel de Preços do PNAE, foi proposto realizar alguns processos de machine learning. Neste trabalho iremos prever o preço unitário de determinado produto, de acordo com a quantidade que foi comprada. Escolhemos essas duas variáveis com base nos atributos da entidade Compra, já que eram os dois campos com uma relação mais promissora dentre todos os atributos presentes na entidade. Para realizar a previsão desses dados foi construído uma aplicação tendo como base o TensorFlow.js, biblioteca Javascript que implementa os métodos do TensorFlow, que possui uma série de componentes e estruturas prontas para poder realizar processamentos de aprendizagem de máquina. O ambiente escolhido para executar a previsão desses dados foi o navegador Google Chrome, já que permite visualizar detalhes dos procedimentos em forma de gráficos.

Para divulgar possíveis indícios de fraude apontados por usuários da aplicação nas redes sociais - no caso deste trabalho, o Twitter - foi criada uma integração que dispara um evento para realizar um post na rede social. Essa postagem tem como objetivo alertar a população e os órgãos competentes sobre possíveis compras suspeitas. Todas as postagens

são realizadas dentro do perfil do projeto @MerendaBot e não identificam o usuário que realizou a avaliação. A integração com as redes sociais é um dos objetivos específicos mais importantes do presente trabalho, já que busca de forma trazer as compras realizadas pelas entidades de forma mais acessível possível para o cidadão, que entre outras razões, se vê assim desobrigado de baixar o aplicativo para ter acesso a possíveis indícios de fraude. A escolha da rede social foi realizada seguindo essa mesma premissa, já que o Twitter é hoje a rede social onde a comunicação acontece de forma mais instantânea, devido entre outras razões - a pequena quantidade de texto característico de um tweet, além de ter uma comunidade bastante ativa, incluindo entre os membros dessa comunidade perfis de pessoas públicas e de entidades organizacionais.

4. Conclusão

Neste trabalho foi realizado um extenso estudo envolvendo questões como dados abertos, transparência governamental e democracia participativa, bem como conceitos aprendidos durante as aulas do curso de Sistemas de Informação como Programação Orientada a Objetos, Engenharia de Software, Banco de Dados, Inteligência Artificial e Estatística. A junção desses conhecimentos caracteriza-se como um excelente apanhado dos conteúdos estudados durante o período da graduação.

As aplicações desenvolvidas durante esse projeto podem servir como modelo para tornar as compras da merenda escolar mais compreensíveis ao cidadão comum, que acredita na importância do uso dos recursos públicos de maneira consciente, evitando assim superfaturamento das compras ou ação de agentes corruptos durante o processo. Trazendo a tona as compras realizadas pelos órgãos públicos, é possível assim que sejam encontrados problemas que normalmente passariam despercebidos já que é incomum que os cidadãos acessem as bases de dados governamentais, ainda mais se elas apresentam problemas de performance, como é o caso do Painel de Preços do PNAE.

Podemos concluir que o trabalho atingiu o seu objetivo geral, já que foi possível realizar um estudo aprofundado, com revisão bibliográfica dos principais tópicos, de medidas tecnológicas que possam auxiliar a ampla execução da democracia participativa e fiscalização das compras efetuadas por órgãos governamentais e disponíveis através da Lei de Acesso à Informação (LAI). Também foi possível desenvolver o ambiente prático para que as pessoas possam contribuir avaliando as compras da merenda escolar deixando a sua impressão se as mesmas possuem indícios que possam ser caracterizados como fraude.

Os objetivos específicos também foram atingidos, na medida em que foi realizado uma revisão bibliográfica dos aspectos apontados na definição do trabalho: sociedade da informação, corrupção, Portal da Transparência, dados abertos, LAI, ciberativismo e hacktivismo e possíveis tecnologias que podem ser utilizadas nesse cenário como Programação MapReduce, Hadoop e TensorFlow.

Cabe destacar também que os objetivos específicos práticos foram atingidos, já que foi implementado tanto a API REST, a solução mobile e o modelo para prever o valor

unitário das compras de nossa base de dados. De forma a buscar uma aproximação ainda maior dos cidadãos com as compra públicas, foi desenvolvido uma integração com a rede social Twitter. Para o desenvolvimento das aplicações propostas foi utilizado essencialmente tecnologias Open Source e tecnologias gratuitas, reforçando assim a distribuição do conhecimento e o papel da tecnologia 107 para transformar a vida das pessoas.

Há um potencial de utilização bastante interessante desse trabalho por parte da sociedade. Além da utilização dos cidadãos para conhecer mais detalhes sobre a compra da merenda escolar, instituições governamentais podem usar os resultados consolidados tanto para detectar fraudes, quanto para otimizar o seu processo de compra já que essa plataforma tem o potencial de ser expandida para um alcance nacional.

O código-fonte da parte prática do presente TCC se encontra disponível de forma pública, sendo possível o seu aperfeiçoamento em trabalhos futuros. Algumas decisões de arquitetura de projeto foram tomadas para facilitar esse desenvolvimento contínuo por parte de novos trabalhos. Por fim, cabe destacar que a informação é o produto de maior valor no nosso século. Somente o povo bem informado, poderá fazer frente à ações que não são de seu interesse e conduzir assim o Brasil no caminho do desenvolvimento para deixar de ser considerado o país do futuro, para se tornar finalmente, o país do presente.

Alguns trabalhos futuros podem ser desenvolvidos a partir da estrutura que construímos neste trabalho. A área com um campo maior de possibilidades é certamente a área de inteligência artificial que pode se beneficiar dos dados disponibilizados pela API REST, na medida em que novos modelos utilizando outras técnicas além regressão sejam construídos para analisar as compras. A partir das avaliações dos usuários também será possível construir um modelo de classificação, onde o algoritmo entende a estrutura dos dados que formam um possível indício de fraude e consegue comparar com novos registros, categorizando-os assim como fraude ou não. Também poderão ser utilizados algoritmos de aprendizagem não supervisionados como o K-means, criando grupos (clusters) entre as compras para obter novas visões sobre o problema.

Outra possibilidade de trabalho futuro que temos é a integração deste com soluções já estabelecidas no mercado, como a Operação Serenata do Amor e os robôs Jarbas e Rosie. Refazer a nossa API e o nosso modelo de previsão de preços utilizando outras tecnologias como a linguagem de programação Python também apresenta um futuro bastante promissor, já que hoje Python é a linguagem de programação da moda na área de inteligência artificial.

É importante frisar que embora o objetivo desse trabalho seja voltado à democracia participativa e ao cidadão, nada impede que trabalhos futuros desenvolvam tecnologias que possam auxiliar os órgãos governamentais, notadamente as prefeituras, na compra da merenda escolar. Toda tecnologia que possa tornar os órgãos governamentais mais eficientes é muito bem-vinda.

Por fim, projetos que envolvam mais áreas do que apenas a tecnologia podem ser realizados, como a condução de testes de experiência com o usuário, para poder redesenhar o aplicativo ou planejar novas features que sejam úteis no dia-a-dia. Para tornar a aplicação esteticamente mais agradável, um trabalho de desenvolvimento de um projeto focado no design do frontend é recomendado.

Referências

BRASIL. Lei n. 8666, de 21 de junho de 1993. Regulamenta o art. 37, inciso Xxi da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências. Diário Oficial da União. Brasília, 21 de junho de 1993.

CAMINHA, Marco Aurélio Lustosa. **A corrupção na Administração Pública do Brasil**. 2003. Disponível em: <https://jus.com.br/artigos/4657/a-corrupcaonaadministracao-publica-no-brasil>. Acesso em: 15 Nov. 2018.

G1. **PF faz operação contra fraude na compra de merenda em 27 cidades de SP**. 2019. Disponível em: <https://g1.globo.com/sp/ribeirao-preto-franca/noticia/2019/11/26/pf-faz-operacaocontra-fraud-e-na-compra-de-merenda-em-cidades-de-sp.ghtml>. Acesso em: 17 Jun. 2020.

GAÚCHAZH. **Mais de um milhão de crianças consomem merenda escolar no Estado**. 2014. Disponível em: <https://gauchazh.clicrbs.com.br/educacao-eemprego/noticia/2014/03/mais-de-um-milhao-de-criancas-consomem-merendaescolar-no-estado-cj5viv9zm0dfoxbj0mbnw4p4v.html>. Acesso em: 17 Jun. 2020.

MARTINS, Maria das Graças. **A tecnologia da informação inovando a transparência nas universidades públicas**. Florianópolis: UFSC, 2014.

TEIXEIRA, Alberto. **Guia da cidadania para a transparência: prevenção para a corrupção**. Ceará: Fundação Konrad Adenauer, 2006