CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Ronicley Silva de Sá

APLICAÇÃO DE BOAS PRÁTICAS NA VISUALIZAÇÃO DE DADOS MINERADOS

Palmas – TO

2020

Ronicley Silva de Sá

APLICAÇÃO DE BOAS PRÁTICAS NA VISUALIZAÇÃO DE DADOS MINERADOS

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Fabiano Fagundes
Coorientadora: Prof.a M.e Heloise Acco Tives Leão.

Palmas – TO

2020

Ronicley Silva de Sá

APLICAÇÃO DE BOAS PRÁTICAS NA VISUALIZAÇÃO DE DADOS MINERADOS

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Fabiano Fagundes
Coorientadora: Prof.a M.e Heloise Acco Tives Leão.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Prof. M.e Fabiano Fagundes

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof.a M.e. Heloise Acco Tives

Coorientador

Instituto Federal do Paraná - IFPR

_____

Prof.a M.e. Madianita Bogo Marioti

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2020

Dedico este trabalho primeiramente a mim, por acreditar em mim mesmo e nunca desistir. Dedico também à minha Mãe, minha namorada e meus amigos, por sempre me ajudar e me incentivar a continuar em meio a todas as adversidades.

AGRADECIMENTOS

Primeiramente agradeço ao universo por ter conspirado para que esse fatídico dia pudesse chegar.

À minha mãe, minha namorada e meus amigos que sempre estiveram me apoiando e me dando forças para continuar em meio a todas as adversidades que a vida me colocou.

A todas as pessoas que olharam para mim e perguntaram ao meu pai se ele estava me ensinando a profissão de marceneiro. Essas mesmas pessoas que acham que por alguém ser pobre, só lhes resta o trabalho braçal.

Ao Corpo Docente do CEULP/ULBRA por sempre dar a direção e auxiliar a partir de conselhos e conversas.

Agradeço meu orientador e Prof. M.e. Fabiano Fagundes por ter me adotado como seu orientando e por sempre responder, quase que imediatamente, minhas dúvidas no *WhatsApp*.

À minha ex-professora e coorientadora Profa. M.e. Heloíse Acco Tives, por me ajudar, tanto na escrita quanto na escolha do tema deste projeto e por ser sempre prestativa e solícita.

A todos que de alguma forma puderam contribuir para que eu pudesse chegar ao patamar de bacharel em Ciência da Computação, os meu sinceros obrigados.

“No mundo da programação nunca podes esquecer um ponto e vírgula (;)..... pois a nossa vida é bastante complicada para usares ponto final (.);”.

Alirio Wilson

RESUMO

SÁ, Ronicley Silva. **APLICAÇÃO DE BOAS PRÁTICAS PARA MELHORAR A VISUALIZAÇÃO DE DADOS MINERADOS DO ENADE**. 2019. 69f. Trabalho de Conclusão de Curso (Graduação) – Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas/TO, 2019.

Com o crescente emprego da mineração de dados e com as políticas governamentais sobre transparência de informações, diversos pesquisadores, estudantes e entusiastas vêm trabalhando com bases de dados abertas que os governos disponibilizam para a obtenção de informações. A mineração e a visualização de dados são duas áreas que andam em conjunto, sem a visualização tarefas como a análise seriam difíceis de serem praticadas devido a quantidade de informações extraídas das bases de dados. O tipo dos dados influencia muito na interpretação do contexto, o ser humano tem uma maior facilidade para interpretar uma figura do que entender uma tabela com vários números. Nesse contexto, o objetivo do presente trabalho foi melhorar a disponibilização dos microdados do ENADE, fornecidos pelo INEP, sobre os acadêmicos da área de Computação nos anos de 2008, 2011 e 2014 que já foram minerados e têm sua visualização disponível atualmente na plataforma ENADE-DM. Portanto, fazendo o uso de figuras, gráficos, dentre outras técnicas, a visualização dos dados está disponível em uma plataforma online de fácil usabilidade, simples e intuitiva.

Palavras-chave: ENADE. Visualização de informações. Inep. Computação.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

3D - *Three Dimension* (Três dimensões)

API - *Application Programming Interface* (Interface de programação de aplicativos)

CSS3 - *Cascading Style Sheets* (Folha de estilos em cascata)

DNPM - Nacional de Produção Mineral

ENADE - Exame Nacional de Desempenho dos Estudantes

HTML - *Hyper Text Markup* (Linguagem de marcação de hipertexto)

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

PC's - *Personal Computer* (computador pessoal)

SBC - Sociedade Brasileira de Computação

JSON - *JavaScript Object Notation* (Notação de Objeto JavaScript)

CDN - *Content Delivery Network* (Rede de fornecimento de conteúdo)

NPM - Node package manager (Gerenciador de pacotes node)

UIs – *User Interface* (Interface de usuário)

SUMÁRIO

1 INTRODUÇÃO

Dados governamentais abertos vêm cada vez mais sendo disponibilizados pelos diversos órgãos públicos brasileiros. Segundo o DNPM (Departamento Nacional de Produção Mineral) (2017), os dados abertos são disponibilizados em formato legível por máquina e não possuem restrições de uso, ou seja, não possuem direitos autorais. A ideia por trás da disponibilização desses dados para o público é uma maior utilização por toda a sociedade.

Um dos órgãos governamentais que disponibilizam seus dados para o público é o INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira), que é uma autarquia federal vinculada ao Ministério da Educação (MEC). Um dos conjuntos de dados disponibilizados pelo INEP se refere ao Exame Nacional de Desempenho dos Estudantes (Enade), aplicado pelo INEP desde 2004, que avalia o rendimento dos formandos dos cursos de ensino superior em relação aos conteúdos previstos na matriz curricular dos cursos, o desenvolvimento de competências e habilidades cruciais para o aprofundamento da formação geral e profissional e o nível de atualização dos estudantes com relação à realidade brasileira e mundial (INEP, 2019).

Um exemplo de sistema que faz uso dos dados do ENADE é o ENADE-DM (2018), que "é uma plataforma de visualização de dados minerados do ENADE, com o atual foco nos cursos da área da Computação: Ciência da Computação, Sistemas de Informação e Engenharia da Computação". A plataforma foi inicialmente pensada como uma maneira de visualização dos resultados da aplicação de algoritmos de mineração de dados. A mineração extraiu dados sobre os gabaritos das provas dos alunos que fizeram o exame. Os gabaritos assim como os demais dados foram disponibilizados pelo INEP sobre o ENADE (ALVES, 2018). A plataforma cumpre o papel de entregar a visualização dos dados, porém não faz isso de forma clara podendo deixar o usuário com dúvidas. A visualização que ela disponibiliza é confusa e falta elementos descritivos, além de os que tem não dizem muito e acabam se tornando irrelevantes.

Para que dados possam ser entendidos e informações relevantes possam ser extraídas, tem-se a visualização, que é o método de transformar informações ou dados para forma visual permitindo ao público sua observação. Através da visualização o usuário consegue perceber características que estão escondidas nos dados, mas que são de grande importância para a atividade de exploração e análise dos dados (BOTELHO, 2002). Para que a mineração de dados seja eficiente, é importante incluir o ser humano no processo de estudo dos dados e mesclar flexibilidade, criatividade e o conhecimento geral do ser humano com a grande

capacidade computacional dos PC's atuais. Com isso, a exploração visual de dados visa integrar o ser humano no processo de exploração de dados, aplicando suas habilidades perceptivas aos grandes conjuntos de dados disponíveis nos sistemas de computadores atuais (IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS (IEEE T VIS COMPUT GR), 2002).

O trabalho se norteia por meio do seguinte questionamento: como os dados apresentados pela plataforma ENADE-DM podem ser melhor visualizados? Dado esta indagação, para melhor visualização das informações mineradas e já disponibilizadas na plataforma ENADE-DM, uma nova interface de apresentação pode ser criada.

Uma das etapas mais importantes da análise de dados é a visualização, e que essa visualização só é eficiente se conseguir fazer com que o visualizador compreenda e interprete as informações contidas. A plataforma ENADE-DM foi pensada como uma forma de disponibilizar a visualização dos dados minerados, porém, o objetivo central do trabalho foi a mineração dos dados, com isso o presente trabalho se faz necessário pois busca aplicar técnicas de visualização de dados, como o próprio uso de gráficos, tabelas dentre outros meios para ajudar o visualizador em sua análise.

Neste âmbito, o presente trabalho tem por objetivo desenvolver uma plataforma online (ENADE-VD), que teve por objetivo, remodelar a visualização de dados e que buscou uma nova abordagem mais aprofundada, no que tange a visualização de dados, para isso, boas práticas referentes a visualização de dados foram estudadas e aplicadas. É importante ressaltar que este trabalho não tem o objetivo de minerar, tão quanto analisar os dados, mas sim de apresentar de forma melhor esses dados minerados e previamente analisados.

O trabalho está estruturado da seguinte forma: a seção 2 apresenta o referencial teórico que está dividido entre Mineração de Dados, que apresenta uma breve contextualização no que tange DM, Visualização de Dados onde são apresentados conceitos técnicas e boas práticas de VD e Trabalhos Relacionados, que apresenta dois projetos ("Visualização de dados como ferramenta em sistemas de Bases de dados para Data mining." (BOTELHO, 2002) e "A Visualização de Informações e a Transparência de Dados Públicos" (Paula et al., 2011)) que utilizaram técnicas de VD em seu desenvolvimento. A seção 3 são apresentados a metodologia e os materiais utilizados no desenvolvimento deste trabalho. A seção 4 apresenta os resultados obtidos e como foi o processo de desenvolvimento da plataforma ENADE-VD. No capítulo 5 são relacionadas as conclusões deste trabalho e, por fim, as referências.

2 REFERENCIAL TEÓRICO

Para formar a base teórica deste trabalho foram abordados alguns conceitos de desenvolvimento web, suas técnicas, bem como algumas das ferramentas mais utilizadas na área de desenvolvimento de *software*. Também foram abordados métodos para apresentação de dados minerados.

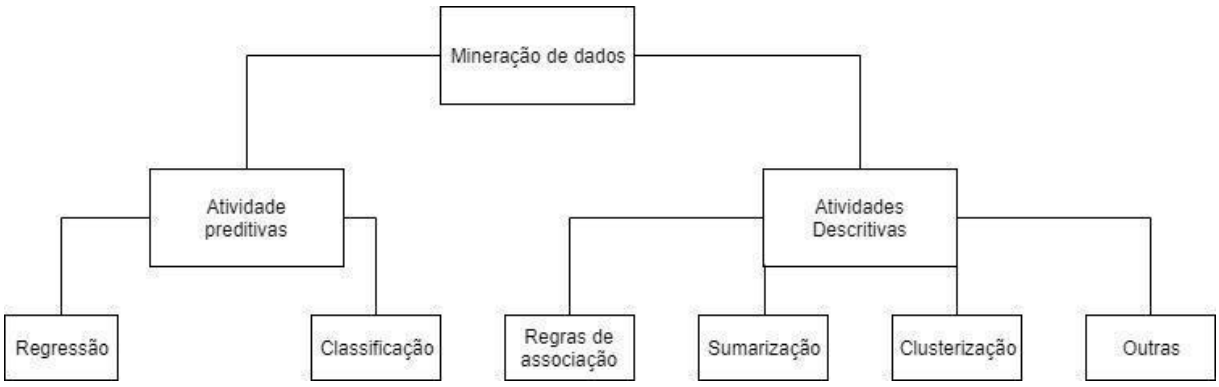
2.1 MINERAÇÃO DE DADOS

A definição sobre o que é mineração de dados varia de autor para autor, para Hand et al (2001) "Mineração de Dados é a análise de grandes conjuntos de dados a fim de encontrar relacionamentos inesperados e de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados". Para Cabena et al. (1998) “é um campo interdisciplinar que junta técnicas de máquinas de conhecimentos, reconhecimento de padrões, estatísticas, banco de dados e visualização, para conseguir extrair informações de grandes bases de dados".

Desta maneira, *Data Mining* pode ser entendida como um processo automático - inteligência artificial, *machine learning*, etc - ou semiautomático - humano + computador - capaz de efetuar uma exploração analítica de grandes *databases*, com o objetivo de descobrir padrões, informações úteis que são cruciais para assimilação de informações importantes e que oferecem suporte para geração de conhecimento útil (SILVA; PERES; BOSCARIOLI, 2016).

A Figura 1 demonstra as tarefas que a mineração de dados realiza assim como suas principais atividades.

Figura 1 - Tarefas de mineração de dados.



Fonte: Tronchoni, 2010.

A mineração de dados tem grande importância no cenário educacional pois, juntamente com regras de associação, pode ser usada para identificar padrões no desenvolvimento do aprendizado e melhorar o desempenho acadêmico dos estudantes.

Através de seu uso e de regras de associação, tomadas de decisões podem ser melhoradas o que implica diretamente na qualidade do ensino e no desempenho dos estudantes, podendo aumentar as chances destes alunos de obterem sucesso no meio acadêmico (LEÃO, 2018).

2.2 VISUALIZAÇÃO DE DADOS

O objetivo principal de qualquer sistema de visualização de dados é a partir de um grande volume de dados extrair o máximo de informação de maneira rápida, clara e precisa (CARVALHO; MARCOS, 2009). Segundo Gershon e Eick (1997) a visualização une duas ferramentas poderosas na análise de dados, o poder de cognição e raciocínio do ser humano e a capacidade de processamento e armazenamento dos computadores.

Outros autores seguem uma mesma linha, segundo Ward (2013) a visualização é a apresentação gráfica da informação, com o objetivo de fornecer ao espectador uma compreensão qualitativa de seu conteúdo. Essa definição enfatiza a necessidade de se obter informações relevantes a partir dos dados visualizados. Isso é importante porque dados que não tem significado se tornam inúteis.

Desta maneira, ao desenvolver sistemas de visualização de dados, os projetistas devem considerar tanto a melhor forma de mapear as informações para uma forma gráfica que seja de fácil interpretação por parte do usuário, como também disponibilizar meios que organizem melhor a forma que os usuários recebem estas informações. (FREITAS et al., 2001).

Embora a visualização de dados seja uma ferramenta importante e uma maneira extremamente eficiente de comunicar uma mensagem, ela pode causar confusão nos usuários que não possuem habilidade ou acesso para avaliar e analisar os dados subjacentes (BURN-MURDOCH, 2013). Outro cuidado necessário é a atenção com a veracidade dos dados fornecidos para que o usuário não seja exposto a dados errados e para que sua interpretação não seja afetada. Portanto, é importante que visualizações de dados criadas sejam estatisticamente válidas.

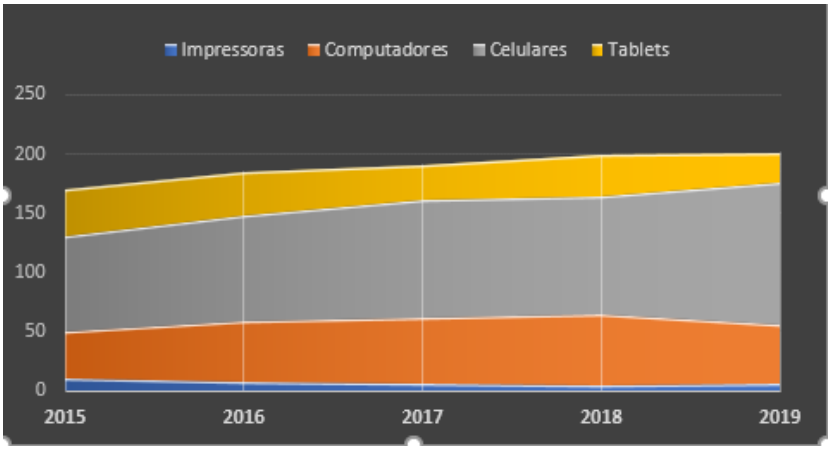
2.3 TÉCNICAS DE VISUALIZAÇÃO DE DADOS E INFORMAÇÕES

Dados minerados em estado bruto são de difícil avaliação e interpretação, uma forma eficaz de encontrar informações relevantes em grandes conjuntos de dados e entender o que esses dados representam é com o uso de figuras ou gráficos que auxiliam nesta interpretação. A seguir são apresentados alguns exemplos de formas de representação da informação para melhor visualização.

Gráfico de Área

Os gráficos de área são úteis para enfatizar a magnitude das mudanças ao longo do período, eles também são utilizados para mostrar a relação das partes com o todo. Os gráficos de área são como gráficos de linhas, mas as áreas abaixo das linhas são preenchidas com cores (IBM, 2019). A Figura 2 apresenta um exemplo de gráfico de área.

Figura 2 - Gráfico de Área.

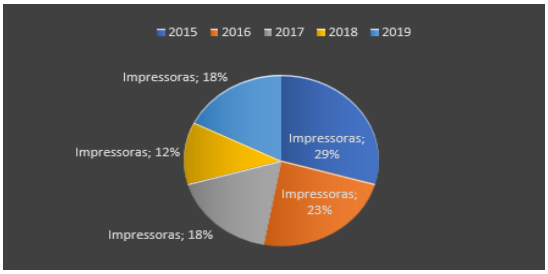


Fonte: Do autor.

Gráficos de Pizza

Gráficos de pizza são gráficos divididos em setores, cada setor da pizza exibe o tamanho de uma parte da informação relacionada. Normalmente, são utilizados para exibir os tamanhos relativos das partes de um todo (TIBCO, 2019). Alguns cuidados devem ser levados em consideração quanto ao uso de gráficos desse tipo, por exemplo o não uso de valores negativos, não ter mais de sete categorias por série de dados e não ter valores zerados (OFFICE, 2013).

Figura 3 - Exemplo de gráfico de pizza.

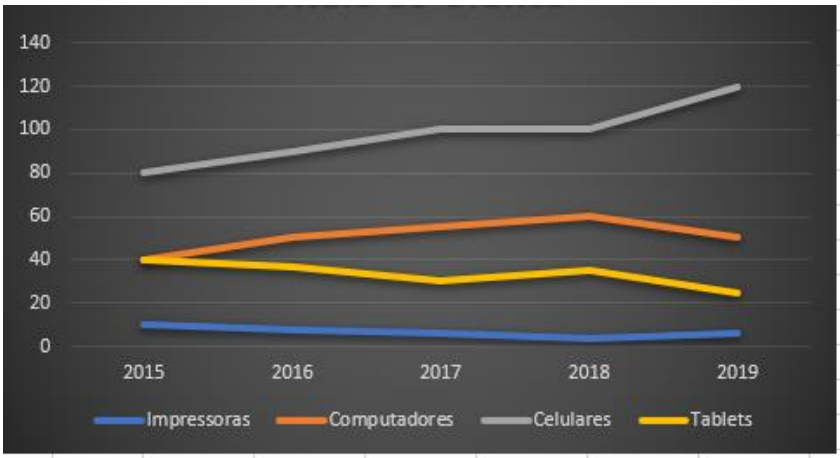


Fonte: Do autor.

Gráficos de Linhas

Os gráficos de linhas mostram as informações como uma série de pontos de dados que estão conectadas por segmentos de linha reta. As categorias são mostradas ao longo do eixo x e as estatísticas são mostradas ao longo do eixo y (ARCGIS, 2019). Este gráfico pode ou não apresentar os pontos ou ícones que representam as ligações entre os segmentos de reta. Ele é comumente usado para visualização de tendência nos dados ao longo de um intervalo de tempo, significa que ele é usado para mostrar tendência no conjunto de dados e ilustrar o comportamento dos dados com a passagem do tempo ou durante um intervalo de tempo específico (INCORPORATED, 2008; SALKIND, 2016). A Figura 4 demonstra um exemplo de gráfico de linha.

Figura 4 - Exemplo de gráfico de linhas.

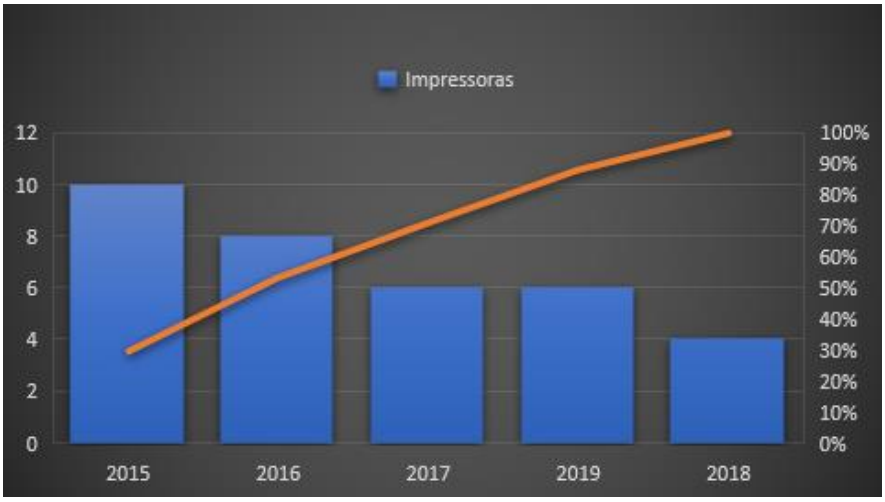


Fonte: Do autor.

Histograma

De acordo com Muzammil KHAN e Sarwar KHAN (2011) as noções sobre o que é um histograma foram apresentadas pela primeira vez por Karl Pearson. O histograma é uma representação precisa da distribuição de dados numéricos, ele é um gráfico de frequência que visa mostrar como determinada amostra ou população de dados está distribuída. Ele, assim como o *dot-plot*, mede quantas vezes determinado valor aparece dentro da distribuição de um conjunto dados (FM2S, 2019). A Figura 5 apresenta um exemplo de histograma.

Figura 5 - Exemplo de histograma.



Fonte: Do autor.

Gráfico de Bolhas

Um gráfico de bolhas é um tipo de gráfico de dispersão, onde os pontos passam a ser bolhas e o tamanho das bolhas passam a ser uma dimensão adicional dos dados. Esse tipo de gráfico, assim como os de dispersão, não tem um eixo que representa a categoria dos valores, ou seja, os eixos horizontais e verticais representam os valores em si. Além dos eixos **X** e **Y**, o gráfico de bolhas tem um terceiro eixo, o **Z,** que representa a dimensão dos valores (no gráfico ele se apresenta como o tamanho da bolha). O gráfico de bolhas é recomendado para conjuntos de dados com 3 séries e o eixo **Z** é determinado pela terceira série desse conjunto (MICROSOFT, 2020).

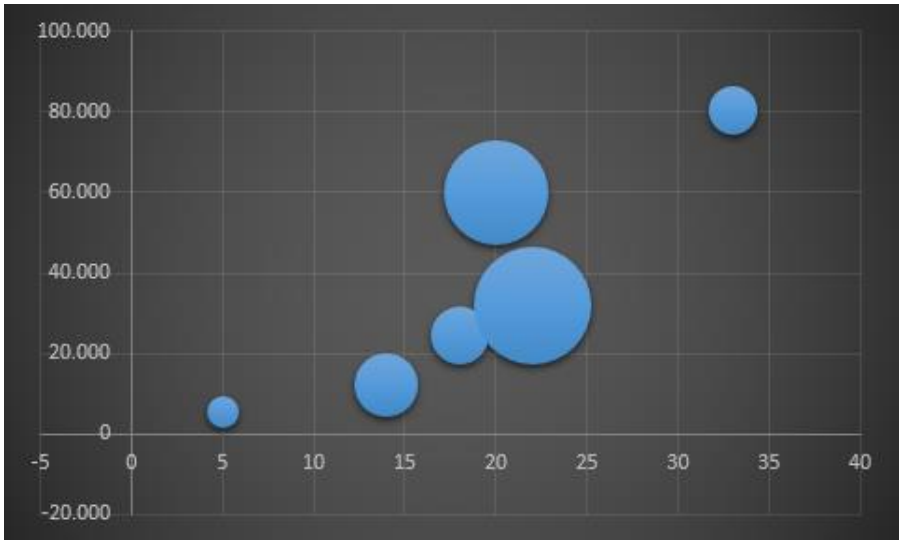
Figura 6 - Tabela de dados

	A	B	C
1	Numero dos produtos	Preços	Percentual de participação de mercado
2	5	5.500	3%
3	14	12.200	12%
4	20	60.000	33%
5	18	24.400	10%
6	22	32.000	42%
7	33	80.258	7%

Fonte: Do autor.

A Figura 6 apresenta a tabela com os dados e a Figura 7 a representação gráfica dos dados.

Figura 7 - Exemplo de gráfico de bolhas.



Fonte: Do autor.

Gráfico Dot-plot

A Figura 8 apresenta um exemplo de gráfico *Dot-plot* (gráfico de pontos), que representa cada observação obtida em uma escala horizontal, permitindo visualizar a distribuição dos dados ao longo deste eixo. No eixo horizontal, divide-se a escala dos valores em pequenos intervalos, sendo marcado um ponto por observação. O *Dot-plot* é muito útil para visualizar estratificações. A estratificação é uma técnica que agrupa dados em subgrupos, de acordo com determinados critérios, aumentando o poder da análise (PORTALACTION, 2019).

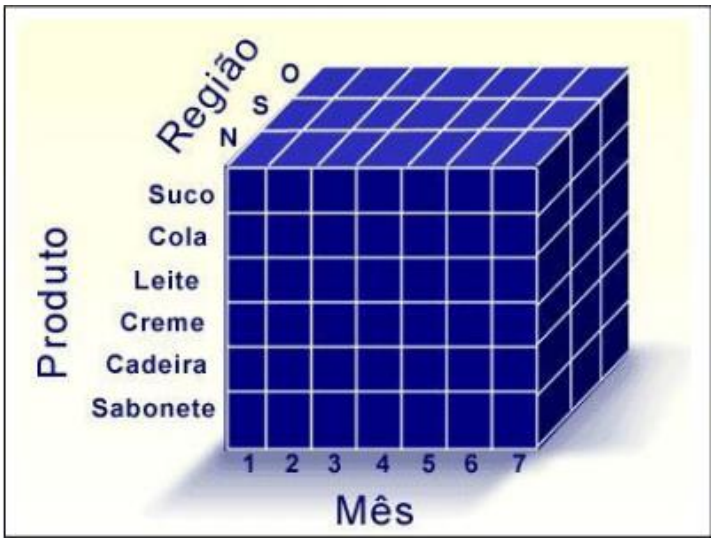
Figura 8 - Exemplo de gráfico de pontos

Fonte: Do autor.

Gráfico de Coordenadas paralelas

Coordenadas paralelas é uma técnica de visualização de informação usada para plotar elementos individuais em várias dimensões. As dimensões são relacionadas com um dos eixos verticais e os dados são exibidos como uma sequência de pontos conectados ao longo dos eixos (SHNEIDERMAN, 2009). Esse tipo de gráfico é bastante útil para grandes quantidades de dados e para os dados que possuam mais de uma dimensão. A Figura 9 possibilita a compreensão do conceito de dado multidimensional.

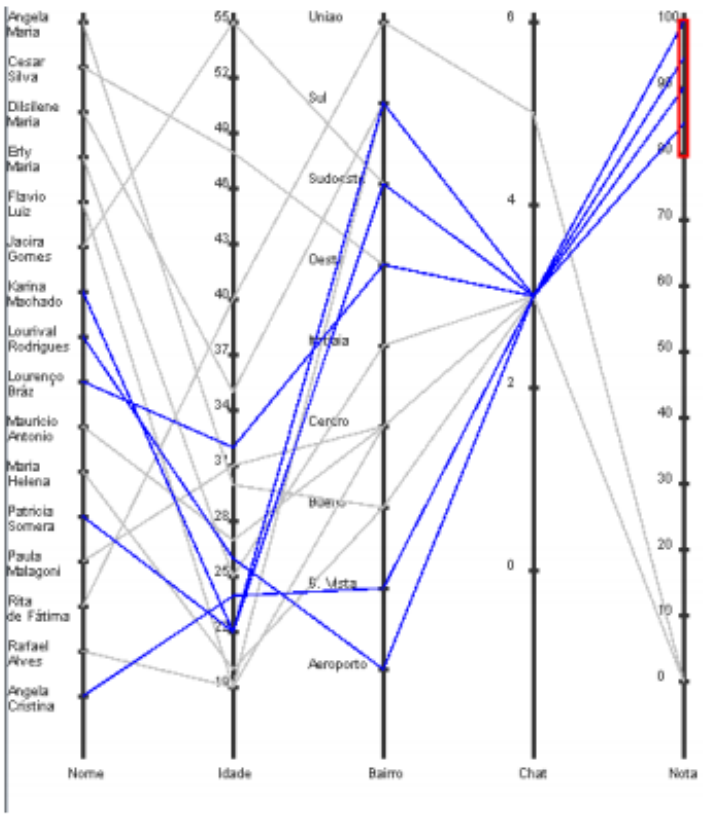
Figura 9 - Dado multidimensional



Fonte: Nardi (2007)

Segundo Stephen Few (2006), a maioria dos meios de visualização padrão não conseguem representar de forma concisa dados multidimensionais. Coordenadas paralelas conseguem fazer essa representação de forma adequada, com ela é possível a plotagem de dados com milhares de dimensões.

Figura 10 - Gráfico de coordenadas paralelas



Fonte: Hugo (2005)

Tabelas de dados

Grande parte dos meios de geração de gráficos partem de tabelas, estas que são uma das formas mais simples de visualização de dados, ela possui um formato estruturado que facilita seu entendimento. A estrutura de uma tabela é bem simples, ela possui linhas e colunas, onde as linhas podem ser entendidas como o conjunto de variáveis e as colunas podem ser entendidas como o conjunto de valores. Isso não é uma regra, portanto, em alguns casos, linhas podem ser os conjuntos de valores e colunas as variáveis (KHAN, Muzammil; KHAN, Sarwar, 2011). Tabelas possuem diversos tipos, estilos e finalidades, sua aplicação é simples e consegue deixar o visualizador a par dos dados com facilidade.

Figura 11 - Exemplo de Tabela

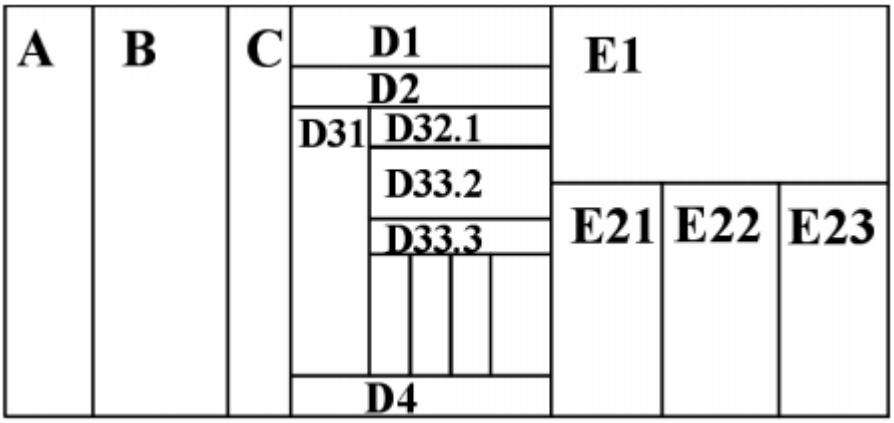
Título	Título	Título	Título
item	item	item	item
item	item	item	item

Fonte: Do autor.

Mapas de Árvore

Visualização de informação por meio de mapas de árvore são um meio eficaz de se visualizar estruturas hierárquicas. Esse tipo de gráfico utiliza retângulos aninhados para representação das informações, isso facilita a comparação por parte dos usuários, dos nós e subnós da árvore.

Figura 12 - Mapa de árvore (Tree map)



Fonte: Freitas (2011)

Visualização de dados visa formar conhecimento a partir de dados gerados de meios específicos e é preciso que informações relevantes sejam retiradas destes dados e estejam acessíveis de forma mais simples e intuitiva possível, nesse contexto, o apelo visual agrega muito. Gráficos, curvas de tendência, dashboards coloridos, sistemas online e alertas simbólicos são ótimas ferramentas nesse quesito (CARVALHO; ARAÚJO, 2008). Nesse cenário, visualização de dados e visualização de informações se fazem necessárias, na maioria dos casos essa ferramenta é um meio de descobrir informações que levariam tempo para serem conquistadas.

2.4 TRABALHOS RELACIONADOS

Visualização de dados como ferramenta em sistemas de Bases de dados para Data mining.

O algoritmo *Fastmap* se baseia na projeção dos *n* objetos de um conjunto de dados em um espaço dimensional ***E***, formado por um conjunto de eixos onde cada eixo é definido por um par de objetos chamados pivôs, obtidos do conjunto de dados. O algoritmo exige que as distâncias entre os objetos no espaço original sejam computadas usando uma função de distância definida pelo usuário (Faloutsos e Lin, 1995). Seu uso visa mapear dados para um espaço de menor dimensionalidade com tempo computacional linear. Além disso, tenta preservar as distâncias entre os objetos minimizando as distorções causadas pela redução, essa redução faz com que os algoritmos de mineração trabalhem mais rapidamente.

Redução de dimensionalidade segundo Lima (2017) consiste em Reconhecimento de Padrões, Estatística e Teoria da Informação e objetivam reduzir o número de dimensões agrupando ou excluindo atributos irrelevantes ou redundantes da descrição das instâncias de dados.

Com o auxílio da ferramenta *FastmapDB*, Botelho (2002), em seu trabalho, construiu um módulo de visualização de dados. Esse módulo foi desenvolvido para integrar a ferramenta, dispensando o uso de qualquer outra ferramenta de visualização externa. O módulo permite uma interação maior do usuário com a visualização dos dados, onde os dados não são mostrados de uma forma estática, mas são possíveis diversas operações interativas, tais como rotação, translação, escala dentre outras.

No módulo visualizador de dados, os processos de Seleção Visual dos dados e mapeamento inverso foram implementados. Nele o usuário tem a possibilidade de selecionar, com o *mouse*, um objeto ou uma região de objetos que lhes interesse. Já no mapeamento inverso, o usuário pode conhecer quais são esses objetos selecionados na base de dados.

A Visualização de Informações e a Transparência de Dados Públicos

O objetivo do trabalho desenvolvido por Paula et al. (2011) foi o de apresentar um estudo realizado para apurar os métodos de InfoVis e sua utilização no contexto governamental. O foco principal da InfoVis é o de transformar um conjunto de dados abstratos em uma representação gráfica e interativa de forma a facilitar o seu entendimento e ajudar no descobrimento de novas informações. Neste trabalho, a disseminação de dados abertos possui duas fases, neste resumo foi apresentado apenas o primeiro. Um experimento

inicial foi realizado e este foi denominado de Ensaio Piloto, seu objetivo foi especificar quais técnicas de visualização seriam consideradas em uma fase posterior do trabalho.

Os estudos demonstraram que a utilização do gráfico de colunas é mais indicada para a análise de discrepância de dados, porém os estudos concluíram também que é necessário investigar cuidadosamente qual o tipo de conhecimento que se pretende criar para assim escolher a melhor forma de apresentar as informações.

3 METODOLOGIA

3.1 OBJETO DE ESTUDO

O objeto de estudo deste trabalho é a aplicação de técnicas de visualização de dados para melhorar o entendimento dos dados por parte do público. Os dados utilizados foram extraídos do ENADE e minerados pelo egresso do curso de Sistema de Informação do CEULP, Renato Marinho Alves, com a plataforma ENADE-DM (2018).

3.2 MATERIAIS

Com intuito de melhorar a apresentação dos resultados obtidos no ENADE-DM, uma aplicação WEB foi desenvolvida com o uso das seguintes tecnologias de desenvolvimento *frontend*:

- ***Java Script* ES6:** é uma linguagem de programação que possibilita criar conteúdo que se atualiza dinamicamente, controlar multimídias, imagens animadas, dentre outros (CONTRIBUTORS, 2019). ES6, *ECMAScript* 6 ou ES2015, é a nova versão do *JavaScript* que foi utilizada em conjunto com o *framework React js* para o desenvolvimento da plataforma de visualização de dados;
- **CSS3:** é a sigla para o termo em inglês *Cascading Style Sheets* (Folha de estilos em Cascatas), utilizada para estilizar elementos escritos em HTML (*Hyper Text Markup Language)* (GONÇALVES, 2019). As *tags* HTML não têm estilização, ou seja, necessitam de algo que faça isso, por esse motivo o CSS3 é necessário. Foi usado em conjunto com o *framework Material UI*. A Figura 13 exemplifica uma mesma página com e sem a aplicação da folha de estilos.

Figura 13 - Site de criação de referências com e sem CSS3

Fonte: Do autor.

- ***Bootstrap* 4**: é um *framework frontend* gratuito que auxilia no desenvolvimento de aplicações web. Ele inclui modelos de design baseados em HTML e CSS3 para tipografia, formulários, botões, tabelas, navegação, modais, carrosséis de imagens e muitos outros recursos, além de plugins *JavaScript* opcionais, também oferece a capacidade de criar facilmente designs responsivos (W3SCHOOLS, 2019). Foi usado neste trabalho para estilização das páginas criadas com HTML;
- **HTML5:** expressão da língua inglesa que significa "Linguagem de Marcação de Hipertexto". Consiste em uma linguagem de marcação utilizada para confecção de páginas web, que permite a criação de documentos que podem ser interpretados em praticamente qualquer tipo de computador e transmitidos pela internet (SIGNIFICADOS, 2018). É composto por *tags*, que possibilitam a escrita em blocos. Esse recurso foi importante para o desenvolvimento do trabalho, pois com ele foi possível a construção das páginas da aplicação;
- ***React js*:** faz com que a criação de UIs interativas seja uma tarefa fácil, pois permite a criação de *views* simples para cada estado da aplicação. Ele irá atualizar e renderizar de forma eficiente apenas os componentes necessários na medida em que os dados mudam. (REACTJS, 2019). O *React js* engloba tudo, tratando toda as regras de negócio para que a aplicação funcione perfeitamente. Foi a principal ferramenta para construção da aplicação, pois se trata de um *framework* para criação de *frontend;*
- ***Node* js:** é uma plataforma altamente escalável e de baixo nível que executa código *JavaScript*. Nele é possível programar diretamente com diversos protocolos de rede e internet, ou utilizar bibliotecas que acessam diversos recursos do sistema

operacional (PEREIRA, 2016). Foi usado neste trabalho por se tratar de uma ferramenta que utiliza linguagem de baixo nível, diversas bibliotecas e *frameworks* foram construídos utilizando *Node js* como base, de modo a disponibilizar diversos serviços, plugins e uma estrutura completa para criação de ambientes tanto web quanto *desktop;*

- ***Visual Studio Code*:** é um editor de código-fonte que roda localmente. Ele vem com suporte interno para *JavaScript*, *TypeScript* e Node.js, possui um rico ecossistema de extensões para outras linguagens (VISUALSTUDIO, 2019). *Visual Studio Code* é um dos editores de código mais usados nos dias atuais, pois disponibiliza diversos plugins e abrange muitas linguagens, além de ser totalmente gratuito, devido a esses atributos auxiliará no desenvolvimento deste trabalho;
- ***React Google Chart*:** é uma extensão do Google *chart* para *React js*. Ele é uma biblioteca que auxilia na apresentação de dados no formato gráfico, além de possuir diversos modelos de gráficos seu uso é totalmente gratuito. (DEVELOPERS, 2019). Devido suas características, foi escolhido como ferramenta de apresentação dos gráficos;
- ***Django Rest Framework*:** ele é uma biblioteca para o *Framework Django* que disponibiliza funcionalidades para implementar APIs *Rest* de forma extremamente rápida (RABAIOLI, 2017). Foi utilizada para fornecer os dados para o *frontend* apresentar;
- ***Firebase hosting*:** é uma plataforma de hospedagem de conteúdo da *Web* de nível de produção para desenvolvedores. Com um único comando, é possível implantar apps da Web rapidamente e exibir conteúdo estático e dinâmico a uma rede de distribuição de conteúdo (CDN) global (GOOGLE, 2020). O *Firebase hosting* foi usando para hospedar o *frontend* da plataforma.
- ***Heroku*:** é uma plataforma como um serviço baseado em um sistema de contêiner gerenciado, com serviços de dados integrados e um poderoso ecossistema para implantar e executar aplicativos modernos. A experiência do desenvolvedor *Heroku* é uma abordagem centrada no aplicativo para entrega de software, integrada às ferramentas e fluxos de trabalho mais populares da atualidade (HEROKU, 2020). Terá grande importância, pois foi usado para a hospedagem do *backend*.
- **Npm:** é um gerenciador de pacotes para `node.js`. Ele foi criado em 2009 como um projeto *open source* para facilitar o compartilhamento de pacotes feitos por

desenvolvedores *JavaScript* (NPMJS, 2020). Ele foi usado como gerenciador das bibliotecas necessárias para o desenvolvimento da plataforma.

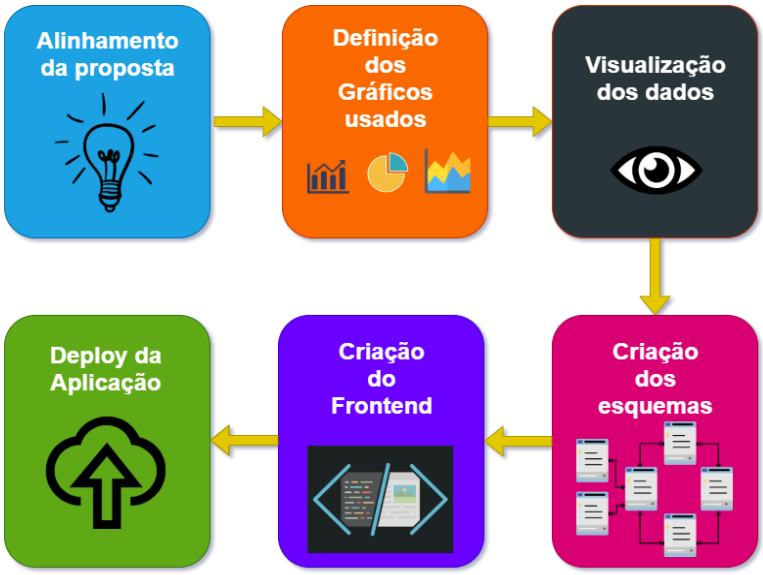
- ***SQLite*:** é uma biblioteca que implementa um mecanismo de banco de dados transacional, independente, sem servidor e com configuração zero. O código para *SQLite* é de domínio público e, portanto, é gratuito para uso para qualquer finalidade, comercial ou privada (SQLITE, 2020). O *SQLite* foi mantido como sistema de armazenamento de dados.
- ***Material ui*:** é um *framwork* para *React js* que auxilia os desenvolvedores a construir interfaces de usuário de forma prática e rápida (MATERIAL-UI, 2020). Foi usado para construir os *layouts* e interfaces.

As tecnologias mencionadas foram utilizadas para desenvolver o ambiente que permitirá a apresentação dos dados minerados do ENADE de forma clara para o público. A interface da ferramenta foi desenvolvida utilizando o *framework ReactJs* juntamente com os *frameworks Material UI,* para estilização do layout e a biblioteca, *React Google Chart,* para apresentação dos dados no formato gráfico.

3.3 MÉTODOS

Os passos para o desenvolvimento deste trabalho estão descritos e organizados em alinhamento da proposta, definição dos gráficos a serem usados na plataforma, criação dos esquemas, criação do *frontend* e *deploy* da aplicação, como esquematizado na Figura 14.

Figura 14 - Passos para o desenvolvimento da proposta.



Fonte: Do autor.

No **alinhamento da proposta**, juntamente com a professora coorientadora foi percebida a necessidade de melhorar e criar novas visualizações dos dados minerados do ENADE e disponibilizados na plataforma ENADE-DM, para isso uma nova plataforma de apresentação destes dados foi projetada.

A **definição dos gráficos** seguiu alguns dos padrões de boas práticas para apresentação visual de dados que estão descritos na seção 2.3 deste trabalho. Uma delas diz respeito ao uso de cores para construção dos gráficos, o que é importante por facilitar as relações que os usuários podem fazer entre os dados.

Os gráficos utilizados para apresentação visual dos dados foram os disponíveis na biblioteca `react-google-charts` (ver seção 3.2), que possui cerca de 25 tipos de gráficos que variam dentro de cada subtipo, por exemplo o gráfico de pizza pode ser o 2D, em 3D ou em formato de *donut,* contudo somente alguns deles foram utilizados.

A visualização dos dados é a parte mais importante desse processo, sendo que os dados terão sua apresentação disponibilizada utilizando algumas das técnicas de visualização de dados apresentadas no referencial teórico deste trabalho e através da plataforma que foi disponibilizada online. A plataforma foi desenvolvida através do uso das ferramentas computacionais citadas na seção 3.2, servindo para a apresentação intuitiva dos resultados obtidos através da mineração de dados. A visualização poderá ter diferentes pontos de vista, que só foi possível devido a elaboração de filtros dinâmicos. Com isso, a ideia principal é que o usuário ao fazer uso da plataforma, consiga perceber padrões, tomar decisões, ter novas ideias e que reflexões surjam a partir disso.

A plataforma ENADE-DM, assim como outros sistemas web, possui um *frontend* que no caso é a própria plataforma de visualização de dados. Possui também uma API (*backend*) que fornece os dados para o consumo do *frontend.* Com isso, **na criação dos esquemas**, possivelmente os dados terão ajustes para que possam ser compatíveis com os gráficos selecionados, visto que essa API foi modelada para outra aplicação. Os dados retornados através do *backend* que se encontra no formato JSON (*JavaScript Object Notation* - Notação de Objeto *JavaScript*) que, por ser um formato de leitura simples entre as diferentes aplicações, facilita a comunicação de dados com o *frontend.*

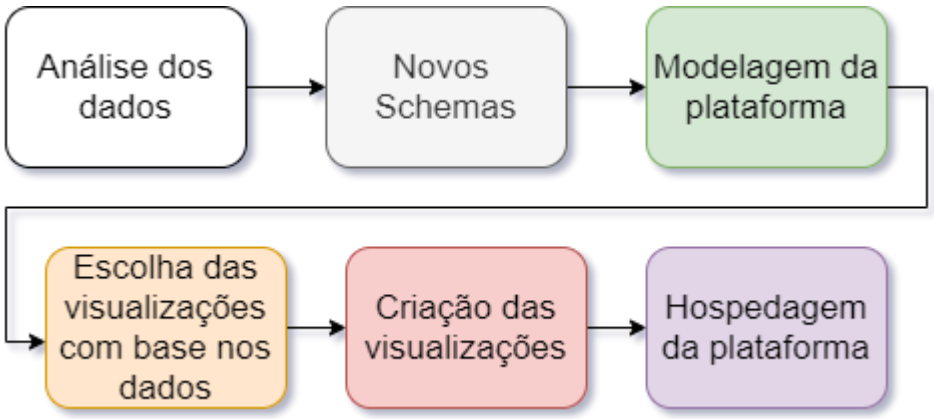
O passo final é a disponibilização online da plataforma ou *Deploy* **da Aplicação**, nesta fase foi escolhida uma ferramenta para fazer a hospedagem da plataforma. Atualmente, existem diversos serviços de hospedagem, como por exemplo o *GitHub,* que é uma plataforma para hospedagem e versionamento de código e que possui um serviço de disponibilização de sites o *GitHub Pages* (ABOUT, 2020). Existe também o *Heroku*, que "é

uma plataforma em nuvem que permite que as empresas criem, entreguem, monitorem e dimensionem aplicativos - somos o caminho mais rápido para ir da ideia ao URL, contornando todas essas dores de cabeça na infraestrutura.” (HEROKU, 2020). O *Firebase* é uma outra opção que é semelhante às ferramentas citadas. Todas possuem serviços gratuitos com algumas limitações, mas que, para o presente trabalho, suprem perfeitamente suas necessidades. Com base nas ferramentas citadas, para o presente trabalho foram escolhidas como serviço de hospedagem do *backend* o *Heroku* e para *frontend* o *Firebase.*

4 RESULTADOS E DISCUSSÕES

Esta seção tem por objetivo apresentar os passos que foram necessários para o desenvolvimento das visualizações propostas. A Figura 15 apresenta o fluxo de desenvolvimento da plataforma, onde em cada uma das etapas os métodos utilizados e os passos que foram desenvolvidos são descritos nas subseções a seguir. Cada etapa possui seu devido grau de importância e deve ser realizada sequencialmente.

Figura 15 - Fluxo de desenvolvimento do trabalho.



4.1 ANÁLISE DOS DADOS

Esta etapa busca analisar os dados a fim de encontrar novas informações para que novas visualizações possam ser construídas. A Figura 16, para exemplificar, apresenta uma tabela contendo dados referentes ao ano de 2008 no curso de Ciência da Computação na área de Engenharia de Software.

Figura 16 - Tabela de dados

Ano	Curso	Area	Região	Vol. Incidências	Qtd. Questões	Qtd. Certas	Qtd. Erradas	Qtd. Branco	Certas (%)	Erradas (%)	Branco (%)
2008	Ciência da Computação	Engenharia de Software	Nordeste	78	1	0	0	1	0%	0%	100%
2008	Ciência da Computação	Engenharia de Software	Centro-Oeste	81	1	0	0	1	0%	0%	100%
2008	Ciência da Computação	Engenharia de Software	Norte	400	1	0	1	0	0%	100%	0%
2008	Ciência da Computação	Engenharia de Software	Nordeste	1422	1	0	1	0	0%	100%	0%
2008	Ciência da Computação	Engenharia de Software	Centro-Oeste	997	1	0	1	0	0%	100%	0%

Fonte: ENADE DM, 2018.

A tabela possui os seguintes atributos: ano, curso, área, região, volume de incidências, quantidade de questões, quantidade de questões certas, quantidade de questões erradas, porcentagens de certas, porcentagem de erradas e porcentagem de questões em branco. Cada linha da tabela faz referência a um objeto no banco de dados.

Foram minerados dados do ENADE nos **anos** de 2008, 2011 e 2014 relativos às questões das provas e seus gabaritos, para cada aluno que realizou o exame. A mineração abrangeu os **cursos** de Ciência da Computação, Sistemas de Informação e Engenharia de Software.

Fazendo referência a coluna **áreas** da tabela, apresentada na Figura 16, a Figura 17 apresenta as demais **áreas e disciplinas** que foram mineradas. Na Figura 17 os dados são apresentados no formato JSON, sendo que todas as figuras que estão no mesmo padrão seguem esse formato.

Figura 17 - Disciplinas


```
{
    "id": 1,
    "area": "Engenharia de Software"
},
{
    "id": 2,
    "area": "Bancos de Dados"
},
{
    "id": 3,
    "area": "Lógica"
},
{
    "id": 4,
    "area": "Tópicos Avançados de Engenharia"
},
{
    "id": 5,
    "area": "Arquitetura de Computadores"
},
{
    "id": 6,
    "area": "Estruturas de Dados"
},
{
    "id": 7,
    "area": "Sistemas Operacionais"
},
{
    "id": 8,
    "area": "Teorias da Computação"
},
{
    "id": 9,
    "area": "Sistemas de Informação"
},
{
    "id": 10,
    "area": "Redes de Computadores"
},
{
    "id": 11,
    "area": "Inteligência Artificial"
},
{
    "id": 12,
    "area": "Computação Gráfica"
},
```


Fonte: Do autor.

Dados referentes às provas de todos os alunos do Brasil que fizeram a exame do ENADE foram coletados. A Figura 18 traz o *schema* que representa as regiões.

Figura 18 - Regiões


```
{
    "id": 1,
    "regiao": "Norte"
},
{
    "id": 2,
    "regiao": "Nordeste"
},
{
    "id": 3,
    "regiao": "Centro-Oeste"
},
{
    "id": 4,
    "regiao": "Sudeste"
},
{
    "id": 5,
    "regiao": "Sul"
}
```


Fonte: Do autor.

A coluna **volume de incidências,** apresentada na tabela da Figura 16, representa a quantidade de alunos que responderam à questão, esse atributo pode ser renomeado para quantidade de respostas, tendo em vista que isso ajudaria usuários mais leigos. As colunas seguintes apresentam se a resposta foi **certa**, **errada** ou **nula,** onde é marcado com 1 para certo e 0 para errado.

Os dados apresentados foram minerados utilizando a tarefa de agrupamento, em que uma incidência que se enquadra em determinadas características, formam um grupo. Uma incidência pode fazer parte de 1 a n grupos de agrupamento.

4.2 NOVOS *SCHEMAS*

Esta seção busca definir novos *schemas* com base no modelo atual. Este modelo que foi obtido a partir do *backend* que fornece os dados para a plataforma ENADE-DM. A Figura 19 apresenta os *endpoints* atuais que a API (*Application Programming Interface* - Interface de Programação de Aplicativos*)* modelada para a plataforma ENADE-DM disponibiliza. A partir desses esquemas ou *endpoints*, e com base nas visualizações desenvolvidas, novos esquemas foram modelados para que, assim, possam atender as requisições de maneira rápida

eficaz e, ainda, disponibilizar apenas os dados necessários para o *frontend*, deixando toda regra de negócio encapsulada.

Figura 19 - *Schemas* atuais.

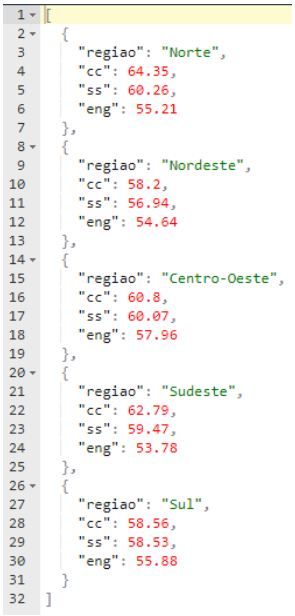
```
 1. resultados
 2. areas
 3. cursos
 4. regioes
 5. anos
 6. resultados/<int:id_curso>
 7. resultados-por-anos
 8. resultados/<int:ano>/<int:id_curso>
 9. resultados/<int:ano>/<int:curso>/<int:area>
10. resultados-associacao/<int:ano>/<int:id_curso>
11. errosbyregion/<int:ano>
12. hitsbyregion/<int:ano>
13. nullsbyregion/<int:ano>
14. number-of-questions-by-area
15. students-by-region/<int:ano>/<int:province>
16. result-by-areas-for-coursers
17. admin
18. api-auth/
```

Fonte: Do autor.

De modo a fornecer para o *frontend* dados para a visualização, os seguintes esquemas foram criados:

`errosbyregion` retorna uma lista de objetos (Figura 20), esses objetos contém as propriedades, `regiao`, `cc`, `ss` e `eng` em que o primeiro representa o nome da região em que as porcentagens foram calculadas, a segunda, terceira e quarta propriedades representam os valores calculados para as porcentagens de erros dos cursos de Ciência da computação, Sistemas de informação e Engenharia de Software.

Figura 20 - Retorno da api para o *endpoint* de *errosbyregion*.




```
[
  {
    "regiao": "Norte",
    "cc": 64.35,
    "ss": 60.26,
    "eng": 55.21
  },
  {
    "regiao": "Nordeste",
    "cc": 58.2,
    "ss": 56.94,
    "eng": 54.64
  },
  {
    "regiao": "Centro-Oeste",
    "cc": 60.8,
    "ss": 60.07,
    "eng": 57.96
  },
  {
    "regiao": "Sudeste",
    "cc": 62.79,
    "ss": 59.47,
    "eng": 53.78
  },
  {
    "regiao": "Sul",
    "cc": 58.56,
    "ss": 58.53,
    "eng": 55.88
  }
]
```


Fonte: Do autor.

O código da Figura 21 é usado para retornar os objetos vistos na Figura 20. Através dos *models* e *serializers* do *django*, uma consulta no banco de dados é realizada de modo a filtrar e calcular a porcentagem de erros para cada curso com base no identificador da região e do curso.

Figura 21 - Código para o *endpoint* de *errosbyregion*.

```
class ResultForCourseThatHadMoreErrorsByRegion(generics.ListAPIView):
    serializer_class = ErrosRigaoByCurso

    def get(self, request, *args, **kwargs):
        regiao_id = list(Dim_regiao.objects.all().values_list('id', flat=True))
        regiao_name = list(Dim_regiao.objects.all().values_list('regiao', flat=True))
        curso_id = list(Dim_curso.objects.all().values_list('id', flat=True))
        total = []

        for i in regiao_id:
            porcentagemcc = 0;
            porcentagemss = 0;
            porcentagemeng = 0;
            for j in curso_id:

                result = Ft_resultado.objects.filter(Q(id_curso=j) & Q(id_regiao=i)).aggregate(
                    qtd_erradas=Sum('qtd_erradas'), qtd_questoes=Sum('qtd_questoes'))

                if j == 1:
                    qtd_erradas = int(result.get('qtd_erradas'))
                    qtd_questoes = int(result.get('qtd_questoes'))
                    porcentagemcc = (qtd_erradas / qtd_questoes) * 100
                elif j == 2:
                    qtd_erradas = int(result.get('qtd_erradas'))
                    qtd_questoes = int(result.get('qtd_questoes'))
                    porcentagemss = (qtd_erradas / qtd_questoes) * 100
                elif j == 3:
                    qtd_erradas = int(result.get('qtd_erradas'))
                    qtd_questoes = int(result.get('qtd_questoes'))
                    porcentagemeng = (qtd_erradas / qtd_questoes) * 100

            total.append(json.loads(
                '{"regiao": "' + str(regiao_name[i - 1]) + '", "cc": ' + str(
                    round(porcentagemcc, 2)) + ', "ss": ' + str(
                    round(porcentagemss, 2)) + ', "eng": ' + str(round(porcentagemeng, 2)) + ' }'))

        return JsonResponse(total, safe=False)
```

Fonte: Do autor.

hitsbyregion tem o comportamento parecido com ***errorsbyregion***, através do qual os mesmos passos são executados, contudo faz isso para a porcentagem de acertos de cada curso por região. A Figura 22 demonstra o retorno do *endpoint* ***hitsbyregion***.

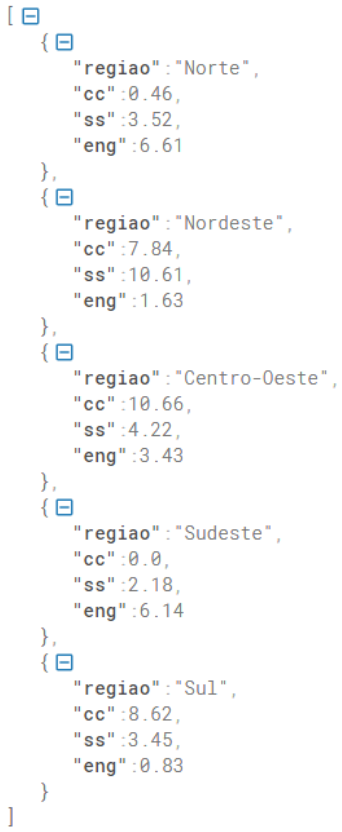
Figura 22 - Saída para o *endpoint* de *hitsbyregion*.


```
[
  {
    "regiao": "Norte",
    "cc": 33.45,
    "ss": 38.23,
    "eng": 42.05
  },
  {
    "regiao": "Nordeste",
    "cc": 37.96,
    "ss": 38.43,
    "eng": 44.64
  },
  {
    "regiao": "Centro-Oeste",
    "cc": 34.3,
    "ss": 38.06,
    "eng": 40.67
  },
  {
    "regiao": "Sudeste",
    "cc": 35.64,
    "ss": 39.59,
    "eng": 40.65
  },
  {
    "regiao": "Sul",
    "cc": 37.36,
    "ss": 40,
    "eng": 43.75
  }
]
```


Fonte: Do autor.

Assim como ***hitsbyregion*** e ***errorsbyregion,*** o *endpoint* ***nullsbyregion*** apresenta a saída esperada ao acesso do *schema.* Ele retorna a porcentagem de questões nulas ou em branco. Os 3 *schemas* possibilitam filtragens escolhendo um ano em particular ou escolhendo “Todos” que representam todos os anos somados. O resultado para `nullsbyregion` pode ser visto na Figura 23.

Figura 23 - Saída para o *endpoint nullsbyregion*.

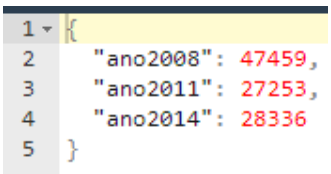


```
[
  {
    "regiao":"Norte",
    "cc":0.46,
    "ss":3.52,
    "eng":6.61
  },
  {
    "regiao":"Nordeste",
    "cc":7.84,
    "ss":10.61,
    "eng":1.63
  },
  {
    "regiao":"Centro-Oeste",
    "cc":10.66,
    "ss":4.22,
    "eng":3.43
  },
  {
    "regiao":"Sudeste",
    "cc":0.0,
    "ss":2.18,
    "eng":6.14
  },
  {
    "regiao":"Sul",
    "cc":8.62,
    "ss":3.45,
    "eng":0.83
  }
]
```


Fonte: Do autor.

`resultados-por-anos` retorna o volume de incidências, ou seja, a quantidade de alunos que participaram do ENADE nos anos de 2008, 2011 e 2014. A Figura 24 apresenta o resultado para a requisição, ele é um objeto contendo as propriedades `ano2008`, `ano2011` e `ano2014` que é a chave para o valor de cada propriedade.

Figura 24 - Saída para o *endpoint* de resultados-por-anos



```
{
  "ano2008": 47459,
  "ano2011": 27253,
  "ano2014": 28336
}
```


Fonte: Do autor.

A Figura 25 apresenta o código da função `ResultByAno` que busca no banco de dados e faz a soma das incidências encontradas retornando assim o objeto JSON apresentado na Figura 24.

Figura 25 - Código para o *endpoint ResultByAno*.

```
class ResultByAno(generics.ListCreateAPIView):
    serializer_class = ResultadoSerializerAno

    def get(self, request, *args, **kwargs):
        anos = list(Dim_ano.objects.all().values_list('ano', flat=True))
        results2008=''
        results2011=''
        results2014=''

        for i in anos:
            if i == '2008':
                results2008 = Ft_resultado.objects.filter(ano=i).aggregate(volume_incidencias=Sum('volume_incidencias')
            elif i =='2011':
                results2011 = Ft_resultado.objects.filter(ano=i).aggregate(volume_incidencias=Sum('volume_incidencias')
            elif i == '2014':
                results2014 = Ft_resultado.objects.filter(ano=i).aggregate(volume_incidencias=Sum('volume_incidencias')

        result = json.loads(
            '{"ano2008" : '+ str(results2008.get('volume_incidencias')) +
            ',"ano2011" : '+ str(results2008.get('volume_incidencias')) +
            ',"ano2014" : '+ str(results2008.get('volume_incidencias')) +
            '}'
        )

        return JsonResponse(result)
```

Fonte: Do autor.

A Figura 26 apresenta o código para o endpoint que retorna a quantidade de estudantes por regiões. A função consegue, através do parâmetro passado na rota, decidir se é necessário retornar todos os dados somados para cada região ou se é por um determinado ano.

Figura 26 - Código para o *endpoint* de *ResultStudentsByAno*.

```
class ResultStudentsByAno(generics.ListAPIView):
    serializer_class = ResultadoSerializer

    def get(self, request, *args, **kwargs):
        ano = self.kwargs['ano']
        regions = list(Dim_regiao.objects.all().values_list('id', flat=True))
        regions_name = list(Dim_regiao.objects.all().values_list('regiao', flat=True))
        result = []
        if ano == 1:
            for i in regions:
                results = Ft_resultado.objects.filter(Q(id_regiao=i)).aggregate(volume_incidencias=Sum('volume_incidencias')
                )
                result.append(json.loads('{"' + str(regions_name[i - 1]).replace('-', '').lower() + '" : ' + str(results.get('volume_incidencias')) + '}'))
        elif ano == 2008:
            for i in regions:
                results = Ft_resultado.objects.filter(Q(id_regiao=i) & Q(ano=ano)).aggregate(volume_incidencias=Sum('volume_incidencias'))
                result.append(json.loads('{"' + str(regions_name[i - 1]).replace('-', '').lower() + '" : ' + str( results.get('volume_incidencias')) + '}'))
        elif ano == 2011:
            for i in regions:
                results = Ft_resultado.objects.filter(Q(id_regiao=i) & Q(ano=ano)).aggregate(volume_incidencias=Sum('volume_incidencias'))
                result.append(json.loads('{"' + str(regions_name[i - 1]).replace('-', '').lower() + '" : ' + str(results.get('volume_incidencias')) + '}'))
        elif ano == 2014:
            for i in regions:
                results = Ft_resultado.objects.filter(Q(id_regiao=i) & Q(ano=ano)).aggregate(volume_incidencias=Sum('volume_incidencias'))
                result.append(json.loads('{"' + str(regions_name[i - 1]).replace('-', '').lower() + '" : ' + str(results.get('volume_incidencias')) + '}' ))
        return JsonResponse(result, safe=False)
```

Fonte: Do autor.

Após a consulta ao *endpoint,* uma lista com a região e a quantidade de estudantes que fizeram a prova naquela região é retornada. Esse resultado pode ser visto na Figura 27.

Figura 27 - Resultado para a consulta ao *endpoint* de *ResultStudentsByAno*.


```
[
  {
    "norte": 5811
  },
  {
    "nordeste": 16303
  },
  {
    "centrooeste": 9968
  },
  {
    "sudeste": 53096
  },
  {
    "sul": 17881
  }
]
```


Fonte: Do autor.

`result-by-areas-for-coursers` retorna as áreas de avaliação que cada curso possui. Na Figura 28 pode-se ver que o objeto tem as propriedades CC, SS e ENG que representam os cursos de Ciência da Computação, Sistemas de Informação e Engenharia de Software, respectivamente. Assim como as outras saídas apresentadas esta também está no formato JSON.

Figura 28 - Resultado para a consulta ao *endpoint* de *ResultByAreasForCoursers*


```
[
  {
    "CC": [
      "Engenharia_de_Software",
      "Bancos_de_Dados",
      "Lógica",
      "Arquitetura_de_Computadores",
      "Estruturas_de_Dados",
      "Sistemas_Operacionais",
      "Teorias_da_Computação",
      "Sistemas_de_Informação",
      "Redes_de_Computadores",
      "Inteligência_Artificial",
      "Computação_Gráfica"
    ]
  },
  {
    "SS": [
      "Engenharia_de_Software",
      "Bancos_de_Dados",
      "Lógica",
      "Arquitetura_de_Computadores",
      "Estruturas_de_Dados",
      "Sistemas_Operacionais",
      "Teorias_da_Computação",
      "Sistemas_de_Informação",
      "Redes_de_Computadores"
    ]
  },
  {
    "ENG": [
      "Engenharia_de_Software",
      "Lógica",
      "Tópicos_Avançados_de_Engenharia",
      "Arquitetura_de_Computadores",
      "Estruturas_de_Dados",
      "Sistemas_Operacionais",
      "Teorias_da_Computação",
      "Redes_de_Computadores",
      "Inteligência_Artificial",
      "Computação_Gráfica"
    ]
  }
]
```


Fonte: Do autor.

A função que pode ser vista na Figura 29, através do identificador da área e curso, faz uma varredura no banco de dados, procurando pelas áreas que cada curso possui.

Figura 29 - Código para o *endpoint* de *ResultByAreasForCoursers*.

```
class ResultByAreasForCoursers(generics.ListAPIView):
    serializer_class = AreasByCoursers
    def get(self, request, *args, **kwargs):
        idArea = list(Ft_resultado.objects.values_list('id_area', flat=True).filter(id_area_id__lte=12).filter(
            Q(id_regiao_id=1) & Q(ano_id=2008)))

        idCurso = list(Ft_resultado.objects.values_list('id_curso', flat=True).filter(id_area_id__lte=12).filter(
            Q(id_regiao_id=1) & Q(ano_id=2008)))

        nameArea = list(Dim_area_enquadramento.objects.all().values_list('area', flat=True))[:12]
        nameCurso = list(Dim_curso.objects.all().values_list('curso', flat=True))[:3]
        cienciaAreas = []
        sistemasAreas = []
        engeAreas = []
        lista = []
        for i in range(len(idCurso)):
            if idCurso[i] == 1:
                cienciaAreas.append(idArea[i])
            elif idCurso[i] == 2:
                sistemasAreas.append(idArea[i])
            else:
                engeAreas.append(idArea[i])
        cienciaAreas = list(dict.fromkeys(cienciaAreas))
        sistemasAreas = list(dict.fromkeys(sistemasAreas))
        engeAreas = list(dict.fromkeys(engeAreas))
        area = ''

        for i in cienciaAreas:
            area += '"' + nameArea[i - 1] + '",'
        lista.append(json.loads('{"CC": [' + area[:-1].replace(' ', '_') + ']}'))
        area = ''

        for i in sistemasAreas:
            area += '"' + nameArea[i - 1] + '",'

        lista.append(json.loads('{"SS": [' + area[:-1].replace(' ', '_') + ']}'))

        area = ''
        for i in engeAreas:
            area += '"' + nameArea[i - 1] + '",'

        lista.append(json.loads('{"ENG": [' + area[:-1].replace(' ', '_') + ']}'))
        return JsonResponse(lista, safe=False)
```

Fonte: Do autor.

As funções e saídas apresentadas para cada *schema* criado demonstram que apenas os resultados necessários sejam retornados para o *frontend.*

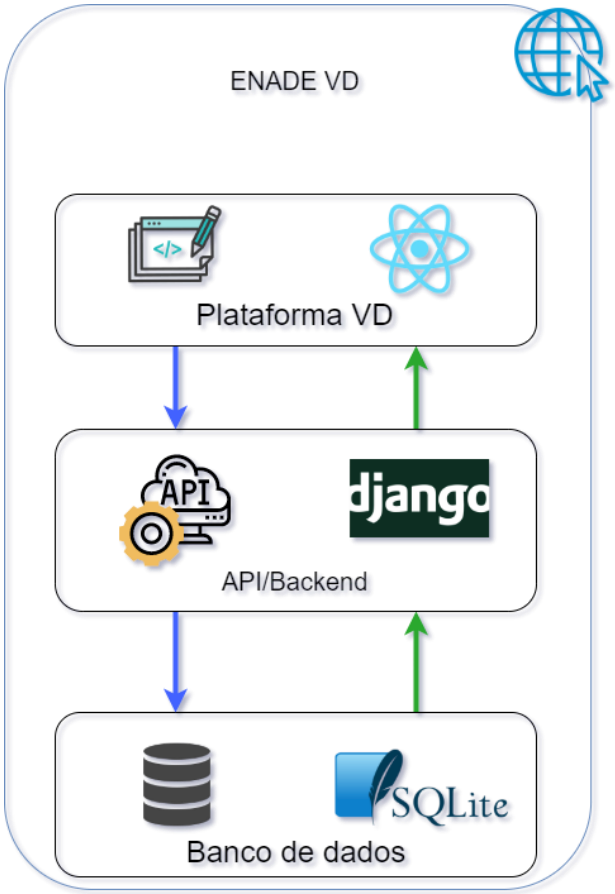
4.3 MODELAGEM DA PLATAFORMA

De modo a facilitar o entendimento com relação ao funcionamento e aos componentes a Figura 30 apresentam desenho da plataforma, que é uma representação visual da plataforma como um todo e demonstra os módulos e como eles interagem entre si. Na figura, a seta azul representa uma requisição de dados e a seta verde representa as respostas com os dados. O primeiro bloco da figura pode ser entendido como o *frontend* e é nele que está localizada toda a parte visual da plataforma. O segundo bloco é a API ou *backend,* onde fica toda a regra de negócio.

É nessa parte também que as consultas no banco de dados são realizadas. O banco de dados é representado pelo bloco três, nele ficam armazenados todos os dados minerados. O uso do SQLite foi mantido pelo fato dos dados já estarem armazenados nele e por sua compatibilidade nativa com *Django*.

Para modelagem da plataforma, um template *admin* foi usado como ponto de partida, para que todo trabalho seja focado apenas na modelagem das visualizações. Ainda, é interessante esclarecer que não foi utilizado qualquer esquema de autenticação ou restrição de uso, ou qualquer técnica que pudessem limitar a experiência do usuário ao utilizar a plataforma. Isso poderia ser uma barreira de entrada e a intenção é que todos os usuários consigam acessar a plataforma sem nenhuma dificuldade.

Figura 30 - Desenho da plataforma

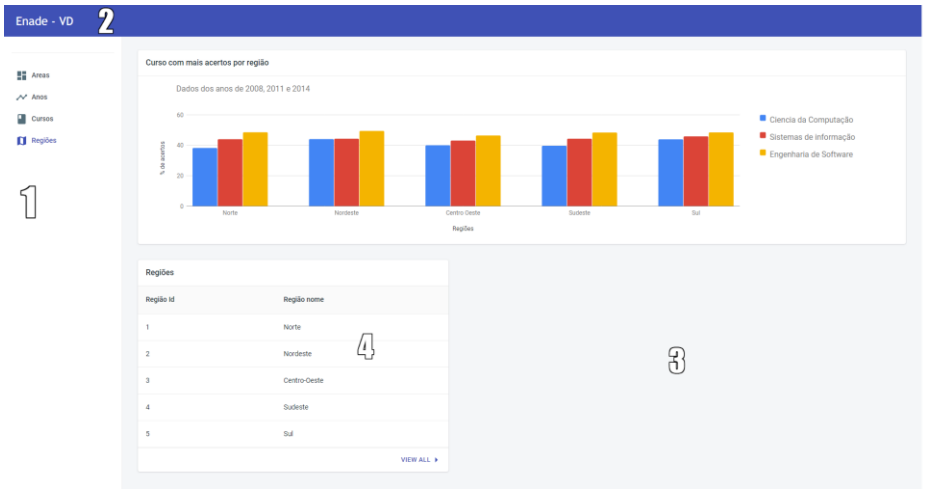


Fonte: Do autor.

Como pode ser visto na Figura 31, em sua parte esquerda é disponibilizado o menu lateral (1), que contém os *links* para navegação entre as páginas da plataforma. Logo mais ao centro (3) tem o componente de *Routes*, onde todos os outros componentes que são

selecionados via menu lateral são apresentados. O componente de *Grid* (4) possibilita construir as interfaces gráficas e tabulares no caso do presente trabalho.

Figura 31 - Plataforma Enade – VD.



Fonte: Do autor.

Para a apresentação dos dados retornados da API e construção do *frontend*, foi utilizado o *framework React Js*, que permite a estruturação de um projeto o dividindo em componentes. Neste cenário, foram utilizados pacotes do *framework* que possibilitam a apresentação destes resultados de forma lúdica. A plataforma foi desenvolvida utilizando dois módulos que incorporam componentes da aplicação, um módulo geral padrão do framework e um módulo *views* responsável pela centralização dos componentes gerais da plataforma.

Desenhar a estrutura da plataforma foi fundamental para o entendimento geral do que está sendo produzido, através dele é possível ter um panorama geral de como funcionam *frontend*, *backend* e banco de dados, é possível perceber também como esses componentes interagem entre si.

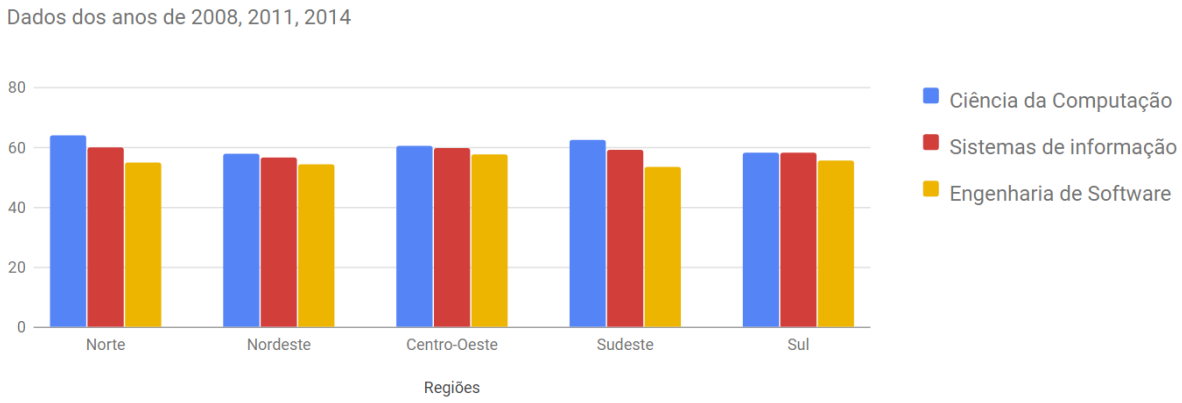
4.4 ESCOLHA DAS VISUALIZAÇÕES

Toda teoria estudada foi fundamental nesta parte do trabalho, pois através dela todo embasamento para determinar qual escolha seria melhor para qual montante de dados foi obtida. A maior parte dos dados são de caráter numéricos, esse foi um fator bastante considerado na escolha das visualizações. Nessas escolhas alguns cuidados foram tomados, pois, a quantidade de propriedades de um conjunto de dados ou o seu tipo tem que ser levado em consideração na hora da escolha.

Como dito antes e para exemplificar, dados com grandes quantidades de atributos e propriedades não são úteis de se apresentar usando um gráfico de pizza, pois a visualização se torna confusa e de difícil compreensão. Portanto, as visualizações escolhidas tentam representar da melhor forma possível um conjunto de dados.

Um dos gráficos escolhidos para representação visual dos dados foi o gráfico de barras, disponível na biblioteca `react-google-charts`, como pode ser visto na Figura 32.

Figura 32 - Gráfico de barras da biblioteca *react-google-charts*.



Fonte: Usage (2020)

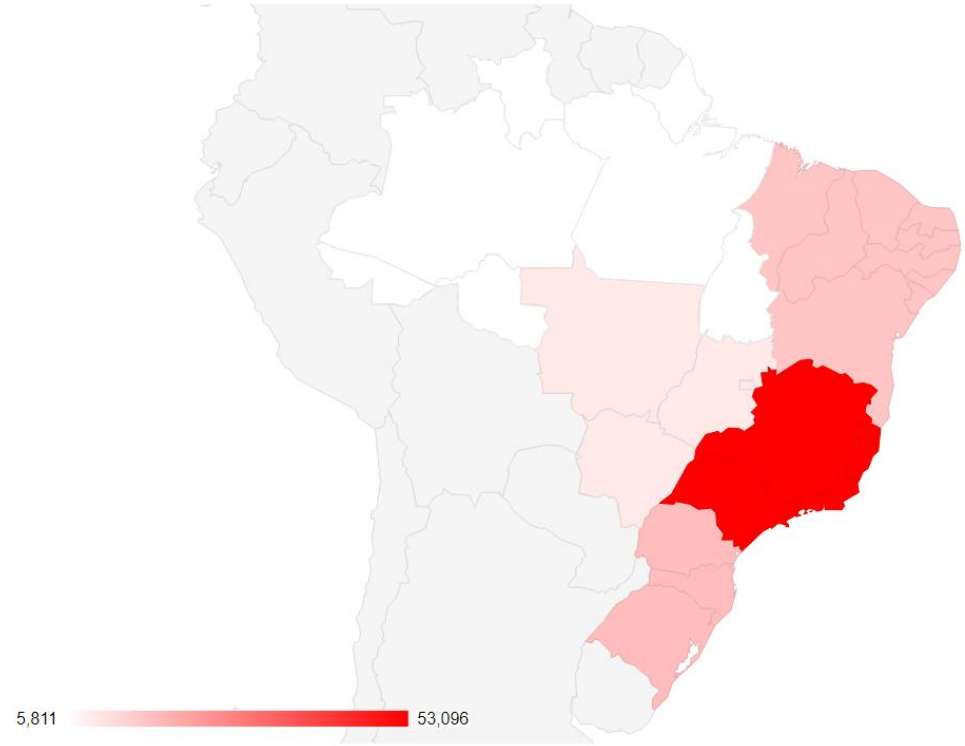
No canto direito tem da Figura 32, uma legenda que descreve o que a cor de cada barra representa. O gráfico ainda possui um título e subtítulo, que podem ser vistos no canto superior esquerdo. Como no exemplo, o gráfico está dividido em 3 variáveis para cada curso, estes cursos estão dispostos no eixo x e os valores de grandeza estão disponíveis no eixo y. Quando passa o *mouse* em cima de qualquer uma das barras, uma legenda é mostrada, nesta legenda é possível ver a região, o nome do curso e a quantidade em valor numérico que a barra selecionada representa, como pode ser visto na Figura 33.

Figura 33 - Legenda dinâmica.

Fonte: Usage (2020)

Outro gráfico usado é o *GeoChart* (Figura 34), que plota em um mapa dados numéricos relativos à determinada região, esta que pode ser um país, estado ou cidade. No canto inferior esquerdo é possível visualizar uma barra que demonstra o menor e o maior valor encontrado nos dados. Duas cores são escolhidas para traçar um gradiente, onde a primeira representa o número menor e a segunda representa o número maior, assim os valores intermediários são representados pelas demais cores do gradiente.

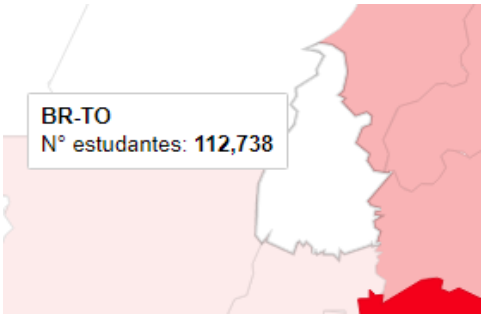
Figura 34 - Gráfico *Region GeoCharts*.



Fonte: Usage (2020)

O gráfico conta com alguns recursos interativos. A Figura 35 demonstra o que acontece ao passar do *mouse* acima de algum estado: um *tooltip* é mostrado e este traz algumas informações sobre o estado, país ou continente em questão.

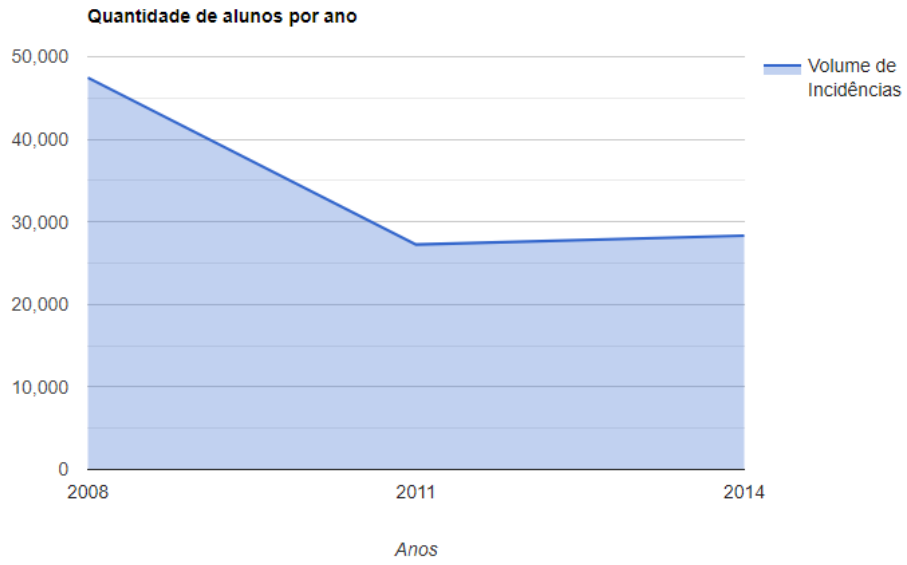
Figura 35 - *Tooltip* informativo para o gráfico *GeoChart*.



Fonte: Do autor.

A figura 36 representa o gráfico de área disponível na biblioteca `react-google-charts`. No canto superior esquerdo é possível ver o título, no canto superior direito tem uma legenda que indica o que cada área representa. No eixo X é possível ver uma legenda e os anos, no Y as quantidades totais. Alguns gráficos disponíveis nesta biblioteca não contam com uma legenda no eixo Y e, para solucionar esse problema, serão adicionadas legendas a todos que não possuírem este recurso.

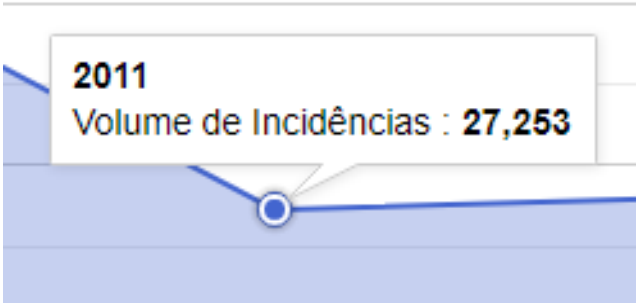
Figura 36 - Gráfico de área.



Fonte: Do autor.

Assim como os outros gráficos, este conta com um *tooltip* que é mostrado ao passar do mouse acima das intersecções. O *tooltip* mostra algumas informações sobre o ponto. Na Figura 37, por exemplo, é mostrado o ano e o volume de incidências para aquele ano.

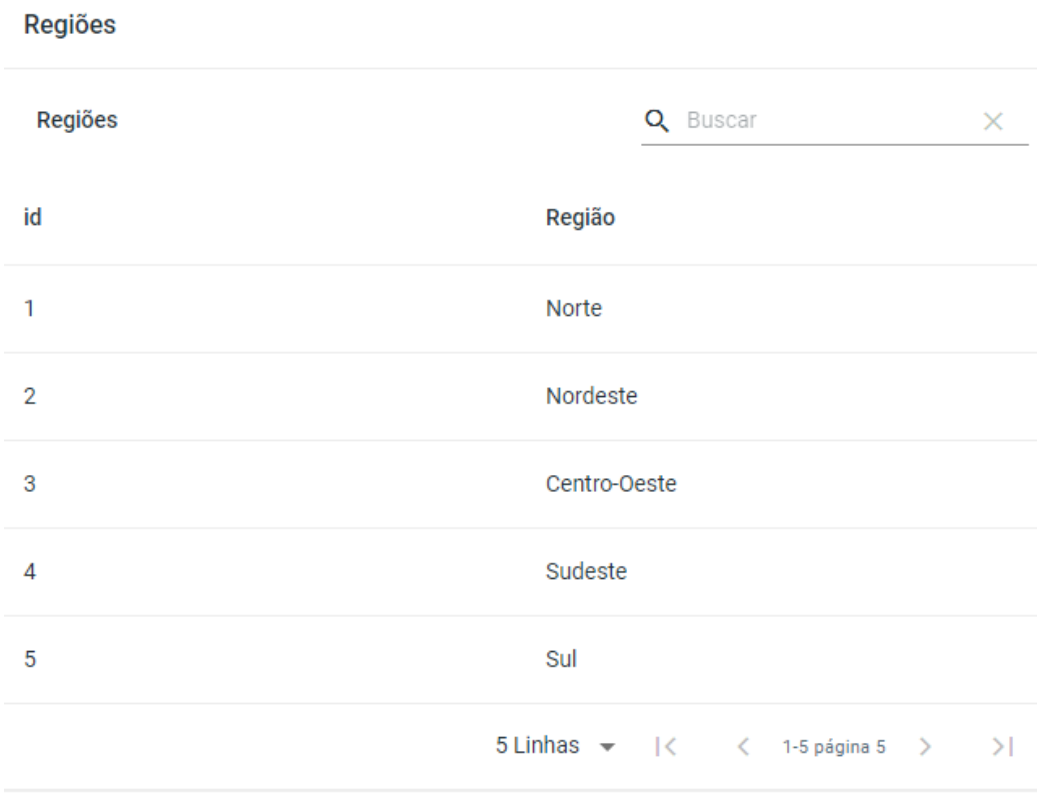
Figura 37 - *Tooltip* informativo do gráfico de área.



Fonte: Do autor.

Para apresentação dos dados em formato tabular foi utilizado o componente de tabela que conta com diversos recursos disponíveis na biblioteca *Material-UI*. Um exemplo de visualização que a utiliza pode ser visto na Figura 38

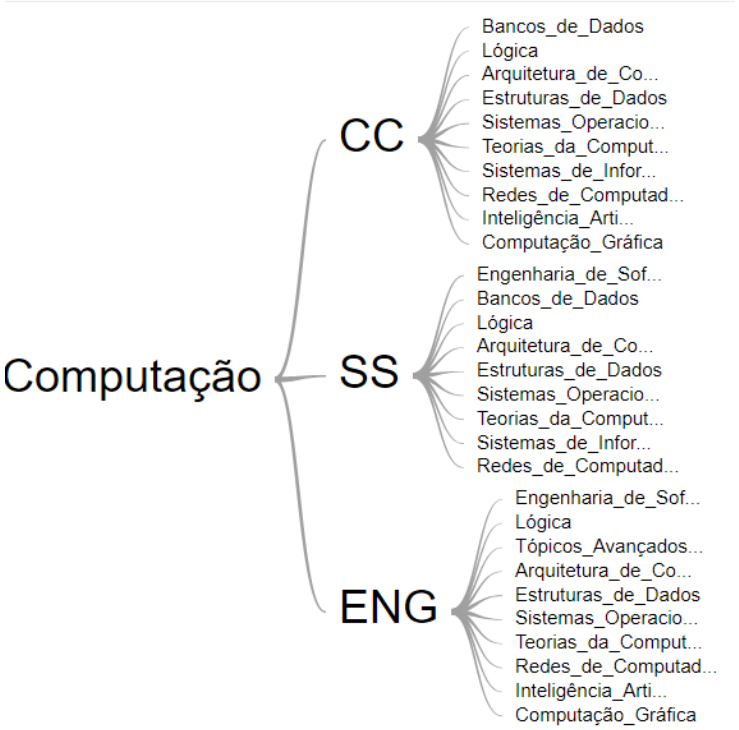
Figura 38 - Tabela com filtragens



Fonte: Do autor.

A biblioteca ainda conta com um gráfico interativo de palavras. Cada palavra do gráfico *Word Tree* (árvore de palavras) é um nó que possui interatividade. Como exemplo, na figura 39, se o usuário efetuar um clique na palavra *eat* (comer), todos os nós que esse nó possuir ficarão visíveis e os demais serão escondidos.

Figura 39- Árvore de palavras.



Fonte: Do autor.

A escolha dos gráficos usados foi uma das etapas mais importantes deste trabalho. Nesta seção os gráficos foram escolhidos, suas particularidades foram descritas e os ajustes necessários, para que estes pudessem ser usados sem que a experiência do usuário fosse prejudicada, foram feitos.

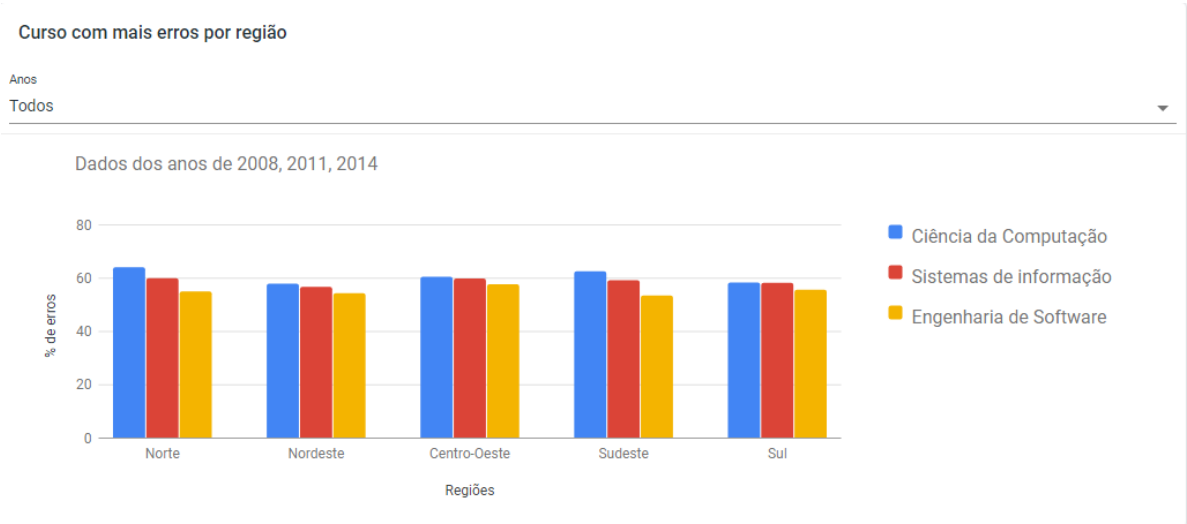
4.5 CRIAÇÃO DAS VISUALIZAÇÕES

Para representar os dados vindo do *backend* de forma gráfica, foi utilizada a biblioteca `react-google-charts`. Esta é uma biblioteca disponível para o *framework React js* e foi criada baseada na biblioteca *JavaScript* disponibilizada pelo *Google.*

A Figura 40 apresenta um *ranking* com os cursos que mais tiveram erros por região. No eixo x estão dispostas as regiões, as barras azuis para cada região representam o curso de

Ciência da Computação, as vermelhas o curso de Sistemas de Informação e as amarelas o curso de Engenharia de Software.

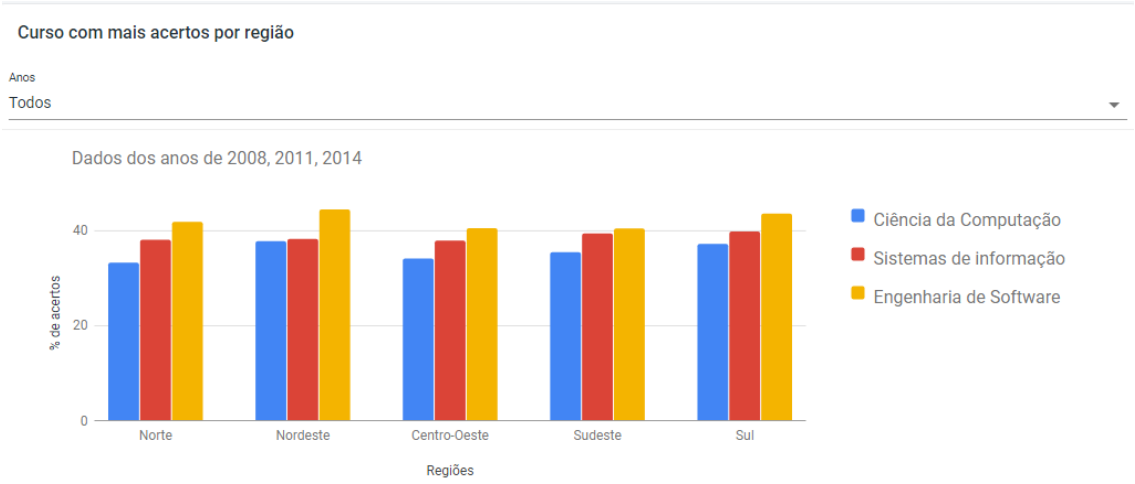
Figura 40 - Curso que teve mais erros por região.



Fonte: Do autor.

No gráfico apresentado na Figura 41 o mesmo padrão se repete, mas agora estão divididos pelo curso que teve mais acertos por região.

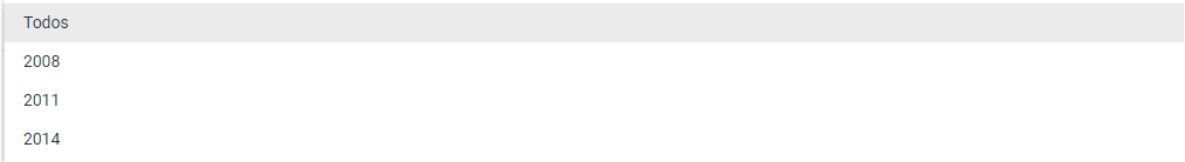
Figura 41 - Curso que teve mais acertos por região.



Fonte: Do autor.

Para as visualizações das Figuras 40 e 41, o usuário conta com a filtragem por ano. Ele pode selecionar ver os resultados para todos os anos, ou por um ano individualmente. O filtro pode ser visto na Figura 42.

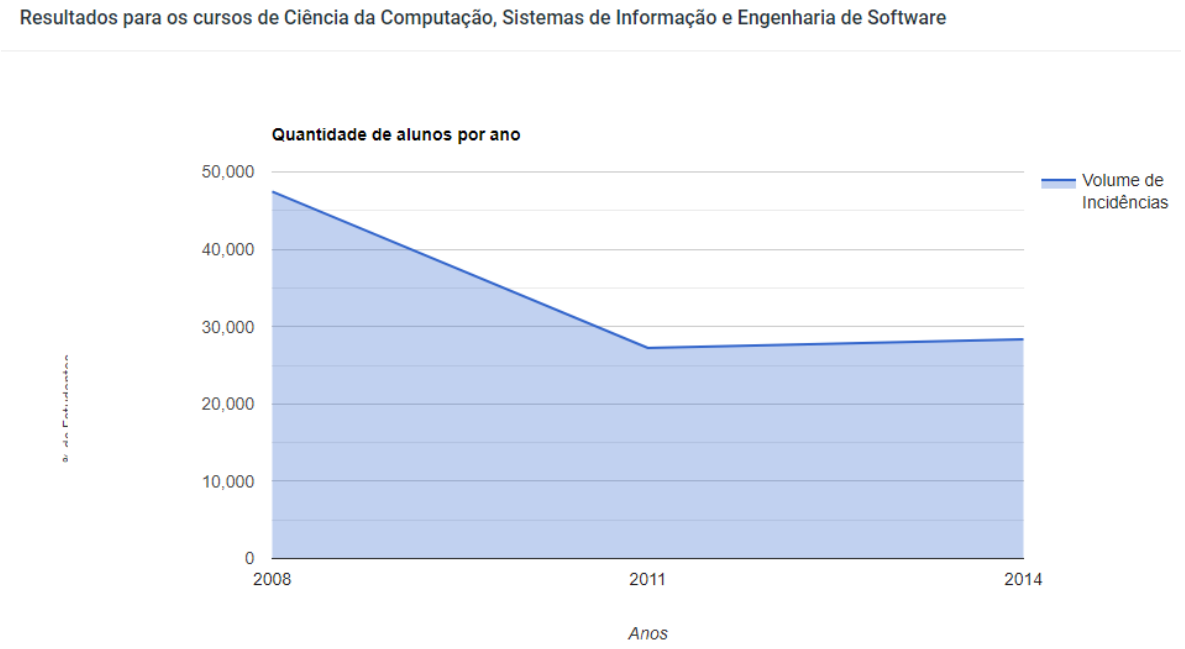
Figura 42 - Filtragem por ano, para erros e acertos por região.



Fonte: Do autor.

A visualização apresentada na Figura 43 fornece um panorama sobre a quantidade de alunos que participaram do ENADE nos anos minerados. É possível perceber que em 2008 o número de estudantes que participaram do exame foi superior aos anos de 2011 e 2014.

Figura 43 - Quantidade de alunos por anos



Fonte: Do autor.

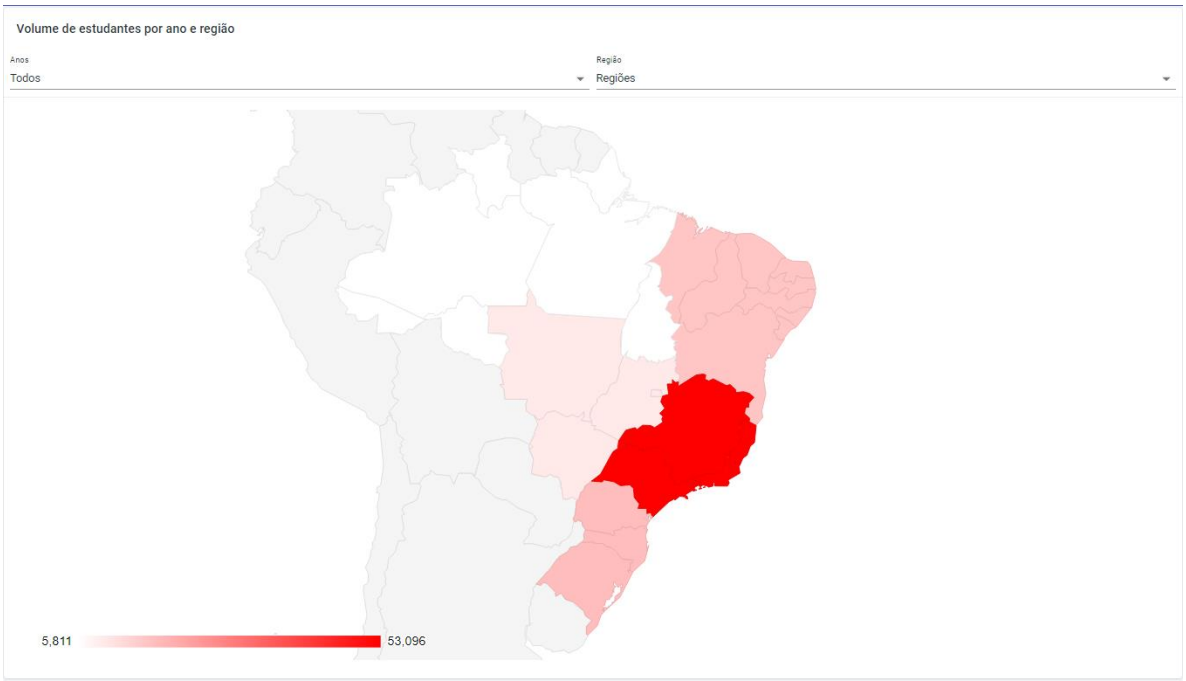
O gráfico *GeoChart,* usado para a visualização gráfica apresentada na Figura 44, não possui uma opção de seleção das regiões (norte, nordeste, centro-oeste, sudeste, sul) do país.

Para contornar esse problema e ainda conseguir apresentar os dados de acordo com a região, em cada estado foi atribuído o valor do número de estudantes pertencentes à região em que o estado está localizado. Por exemplo, a parte Sul do país conta com os estados de Santa Catarina, Rio Grande do Sul e Paraná. Portanto, para estes é atribuída a mesma

quantidade de estudantes que é a soma dos três estados. É possível observar que a região sudeste tem a maior parte concentração de estudantes que participaram do exame durante os períodos minerados.

A visualização conta com um filtro semelhante ao filtro exibido na Figura 42, onde é possível ver os resultados para todos os anos ou por cada um individualmente.

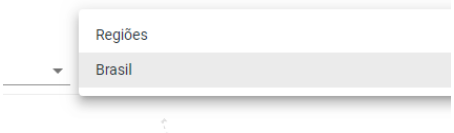
Figura 44 - Quantidade de alunos por região.



Fonte: Do autor.

A visualização ainda conta com outro filtro. Nele é possível selecionar os dados referentes ao país inteiro ou por suas regiões. O primeiro filtro, da esquerda para a direita, é possível ver informações referentes a todos os anos somados ou por cada ano individualmente. No filtro da direita, selecionando “regiões” os dados são divididos por região. Selecionando a opção “todos”, todas os dados de cada região são somados e o resultado é representado por todo mapa brasileiro demarcado, como pode ser visto na figura 46. A figura 45 demonstra o filtro e suas opções.

Figura 45 - Filtragem dos dados do país ou por regiões



Fonte: Do autor.

A Figura 46 demonstra o comportamento do gráfico após a aplicação do filtro. Como apenas um valor existe, o gráfico seleciona automaticamente a segunda cor. A legenda que pode ser vista na imagem apresenta o total de estudantes levando em consideração todas as regiões somadas.

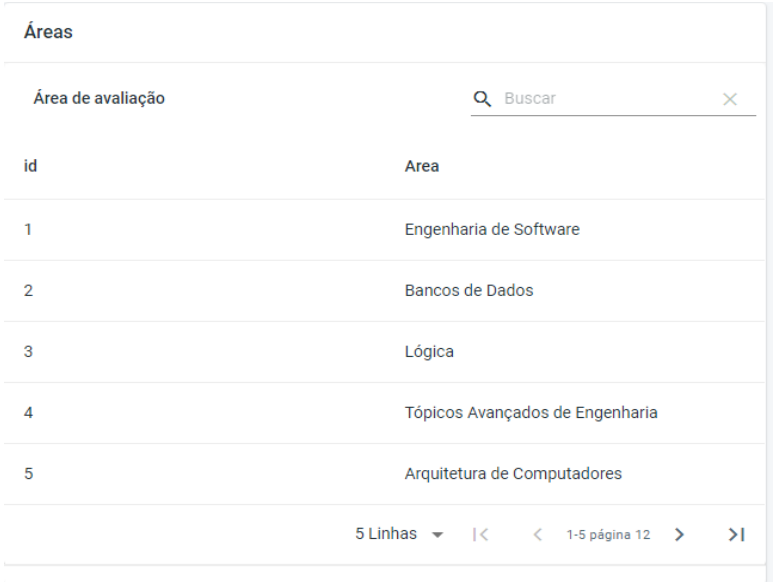
Figura 46 - Gráfico após a aplicação do filtro.

Fonte: Do autor.

A Figura 47 apresenta uma lista de áreas de avaliação. Estes dados estão disponíveis em uma tabela que conta com recursos de paginação, busca dinâmica e opção de seleção da quantidade de linhas mostradas. Essa visualização é simples, porém, consegue demonstrar de forma lúdica as áreas de avaliação disponíveis, através de seus recursos interativos

A visualização tabular foi utilizada também para listagem de Anos, Cursos e Regiões que funcionam da mesma maneira que é apresentada na Figura 47

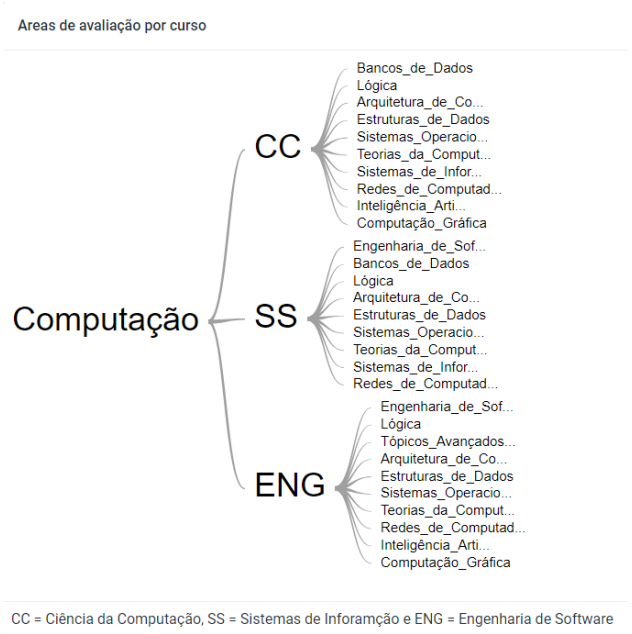
Figura 47 - Tabela com a listagem de áreas



Fonte: Do autor.

No gráfico de palavras apresentado na Figura 48 é possível ver as áreas de avaliação que cada curso possui. Através desta visualização é possível perceber as áreas em comum e as mais específicas de cada curso.

Figura 48 - Árvore de palavras.



Fonte: Do autor.

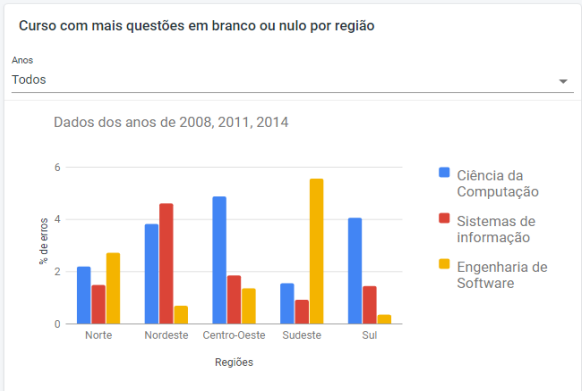
Ao selecionar o curso de Ciência da Computação as áreas de avaliação do curso são apresentadas, ao clicar novamente o gráfico retorna para o estado anterior ao clique. A Figura 49 demonstra este comportamento.

Figura 49 - Áreas para o curso de Ciência da Computação.

Fonte: Do autor.

A visualização apresentada na Figura 50 conta com os mesmos recursos utilizados nas demais visualizações que fazem uso do gráfico de barras. Esta visualização permite ao usuário ver o curso que mais teve questões deixadas em branco ou anuladas. O gráfico está dividido por regiões, assim como também há a possibilidade de se filtrar por acertos e erros. Ele conta, ainda, com um filtro que seleciona todos os anos somados ou cada um dos 3 individualmente.

Figura 50 - Quantidade de questões em branco e nulo por região.



Fonte: Do autor.

4.6 HOSPEDAGEM

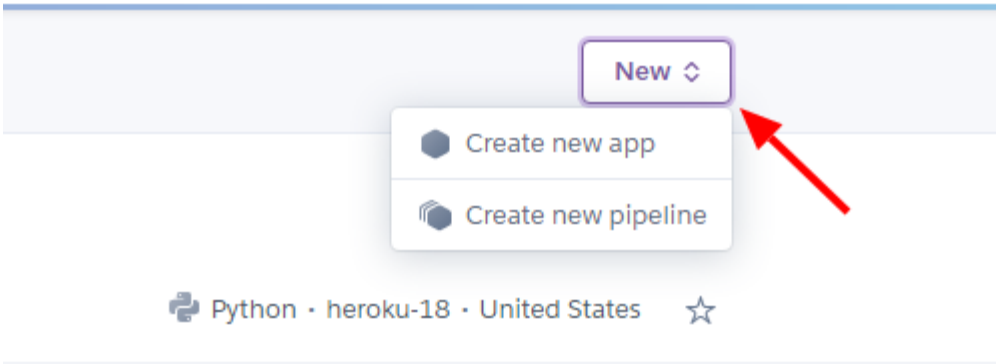
A hospedagem da plataforma é uma etapa relevante do trabalho, já que a partir da sua realização, a plataforma será disponibilizada para comunidade.

4.6.1 HOSPEDAGEM DO BACKEND

A hospedagem do *backend* foi um passo importante no desenvolvimento deste trabalho. É importante observar que ele não foi recriado, apenas novos *schemas* foram adicionados e estes não interferiram no processo de envio. Além disso, todas as configurações necessárias na plataforma para que seu *deployment* fosse realizado, já foram feitas. Apenas um novo projeto no *Heroku* foi criado para que a api pudesse ser hospedada.

Primeiramente, para que uma aplicação possa ser hospedada no *Heroku* é necessário criar uma aplicação no serviço. Para criar, basta clicar no botão *New* disponível na *dashboard* e apresentado na Figura 51. Ao selecionar serão mostradas duas opções e então selecionar a opção *"Create new app* - Criar novo aplicativo".

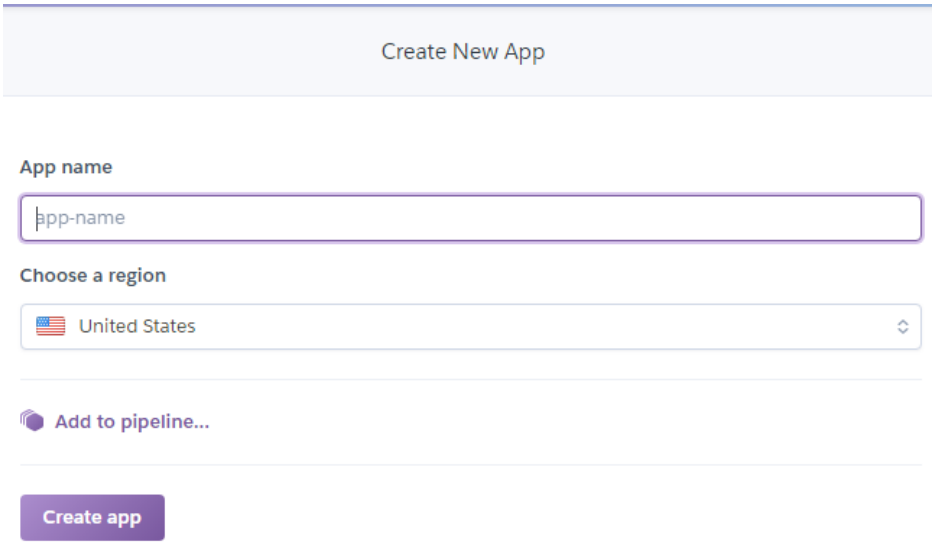
Figura 51 - *Dashboard* do *Heroku*.



Fonte: Do autor.

Ao selecionar a primeira opção um formulário será apresentado (Figura 52), o formulário deve ser preenchido com os dados requisitados, o botão "*Create app* - Criar aplicação" deve ser clicado para que a aplicação seja criada.

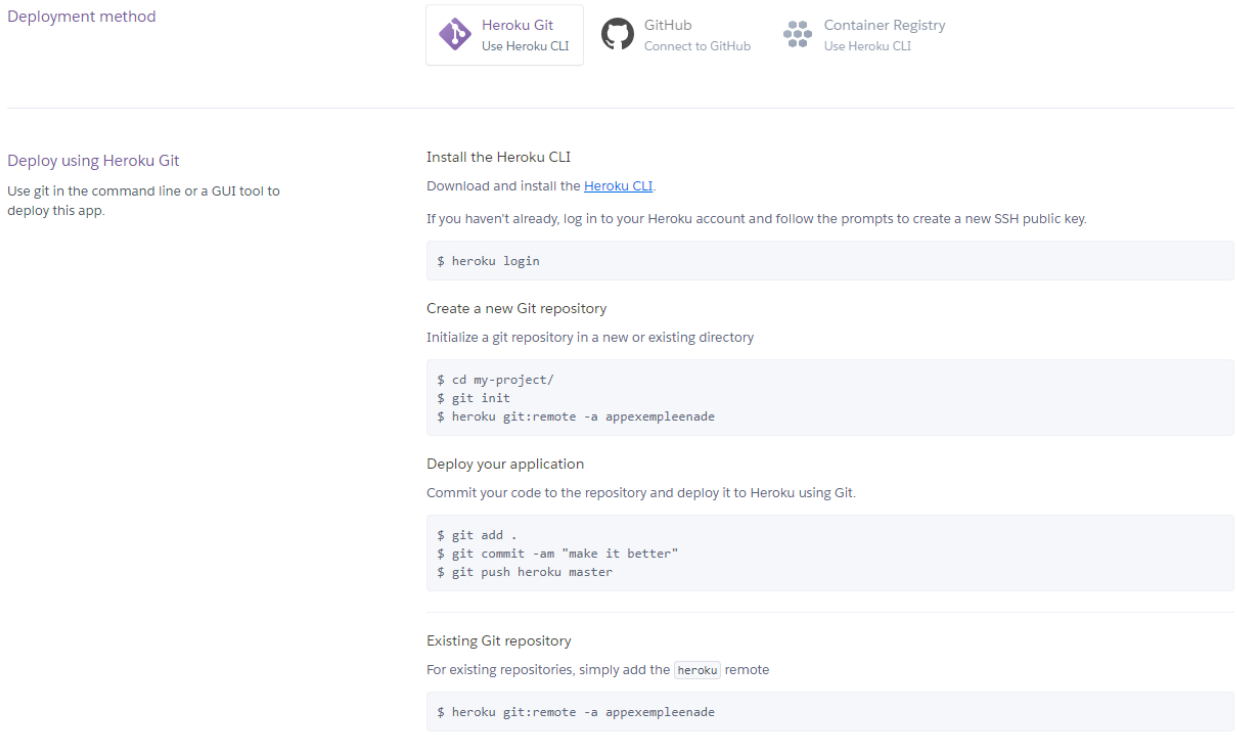
Figura 52 - Formulário de criação da aplicação.



Fonte: Do autor.

O *Heroku* fornece algumas opções de upload ou integração com outros serviços como *Heroku Git*, *GitHub* e *Container Registry* que podem ser vistos na Figura 53. Para o presente trabalho, como todo código do *backend* foi versionado na plataforma *Github*, apenas a configuração da integração com o repositório *Git* foi necessária.

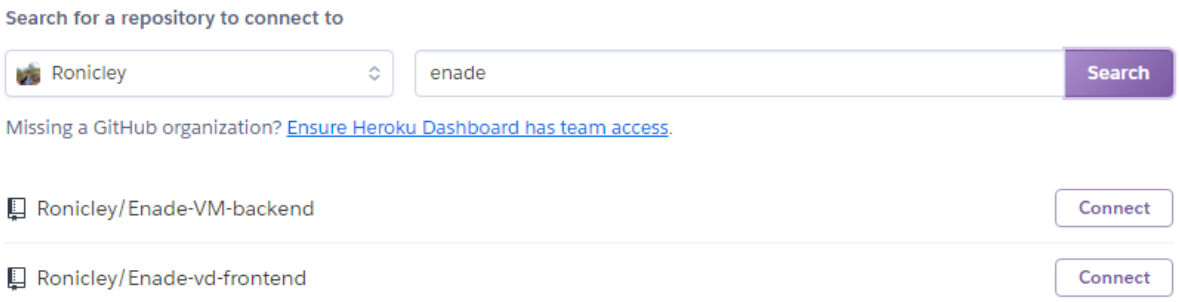
Figura 53 - Tela para *deploy* do *backend*



Fonte: Do autor.

Com o *Heroku* e *GitHub* conectados, o próximo passo foi buscar pelo repositório no GitHub, após isso selecionar repositório correto e clicar em *connect*. A Figura 54 apresenta a tela onde isso foi realizado.

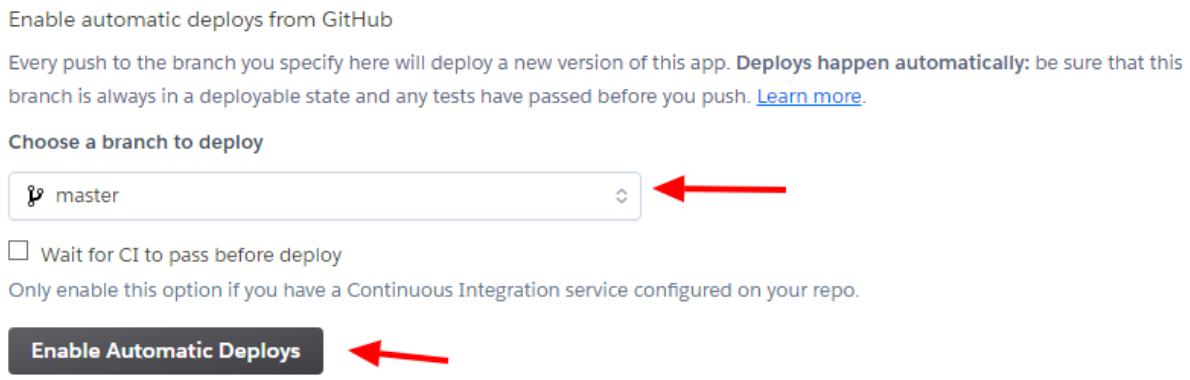
Figura 54 - Seleção do repositório Git.



Fonte: Do autor.

Com o repositório conectado, o próximo passo foi escolher a *branch* (Figura 55), por convenção a *branch master* foi selecionado, pois é nela em que a versão de produção ou final é disponibilizada. Na Figura 55 é possível ver a opção "*Enable Automatic Deploys* - Ativar Envio Automático", ela permite ao *Heroku* atualizar a aplicação sempre que uma nova versão for enviada para a *branch master* do *GitHub*. Com isso a configuração do *backend* foi concluída e assim ele passa a estar disponível para responder as requisições feitas pelo *frontend*

Figura 55 - Formulário de criação da aplicação.



Fonte: Do autor.

Seguindo os passos apresentados na seção 4.2 é possível ver o funcionamento do backend hospedado. Por utilizar o plano gratuito do *Heroku,* sempre que a aplicação deixa de ser usada a máquina virtual onde a hospedagem foi feita é “desligada”. Apesar disso, o heroku ainda é uma ótima opção devido aos seus benefícios do plano gratuito. Para a realizar os testes o endereço (https://enadevmapi.herokuapp.com) pode ser acessado e o retornos podem ser vistos. Com isso esta seção é finalizada e o próximo passo é a hospedagem do *frontend.*

4.6.2 HOSPEDAGEM DO FRONTEND

Para a hospedagem do *Frontend* foi necessária a utilização da versão de produção do projeto em *React js* que pode ser obtida a partir do comando `npm build`. Este comando comprime os arquivos de estilização e configuração do projeto em arquivos *JavaScripts* para sua utilização em servidores web. Feito o *build* do projeto, o próximo passo foi instalar a biblioteca *firebase-tools* através do comando `npm i firebase-tools` para fazer o envio da versão de produção para o *Firebase hosting.*

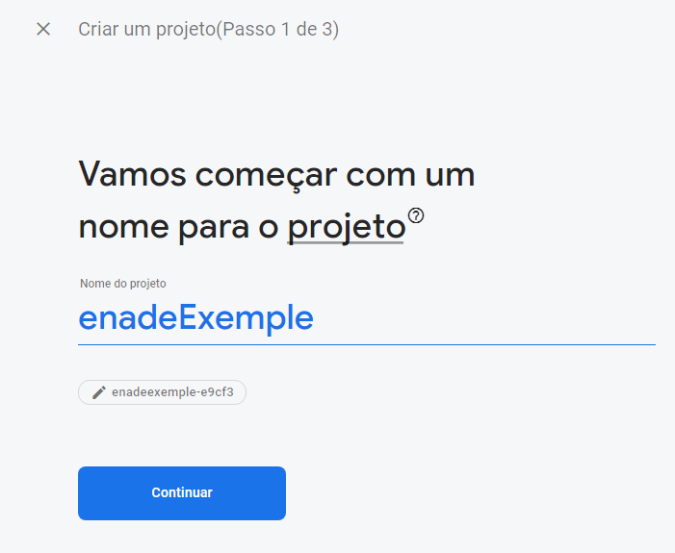
Para hospedagem de uma aplicação no *Firebase* foi necessário inicializar um projeto na *dashboard* do *Firebase Hosting* (Figura 56).

Figura 56 - Botão de adição de um projeto novo no *Firebase*

Fonte: Do autor.

Após clicar no botão de adicionar projeto, a tela da Figura 57 é apresentada. Nela, se coloca o nome do projeto e seleciona a opção continuar.

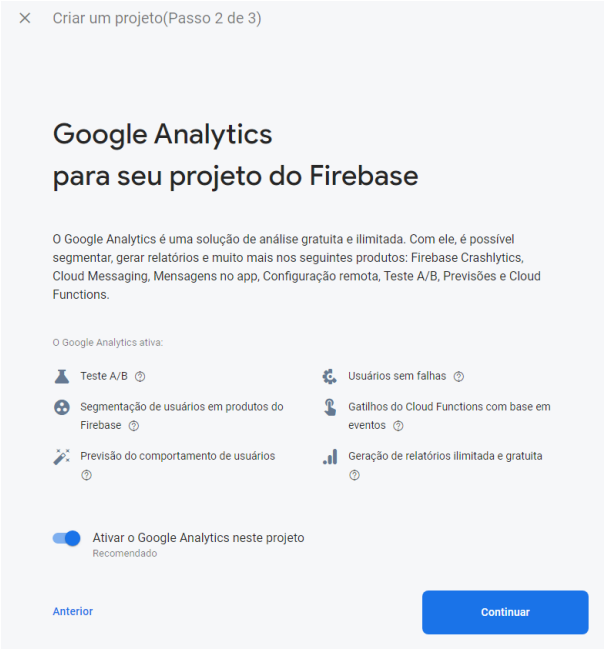
Figura 57 - Escolha do nome do projeto.



Fonte: Do autor.

Na Figura 58 é possível ver a opção “Ativar o *Google Analytics* neste projeto”. Ela ficou desativada pois não foi necessária para o projeto. Após desativar esta opção, o próximo passo é clicar no botão “Criar projeto”. Isso finaliza a criação do projeto no *Firebase Hosting*.

Figura 58 – Tela de *Analytics* do projeto

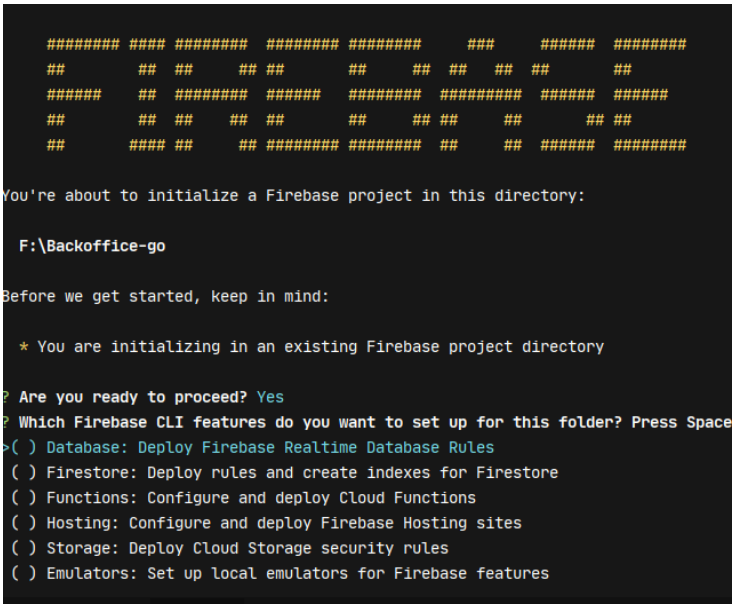


Fonte: Do autor.

Com isso, os próximos passos para fazer o envio do *Frontend* para a hospedagem do *Firebase* foram:

- `firebase init`: ao digitar este comando no terminal a saída demonstrada na Figura 59 foi apresentada, neste momento a CLI pergunta qual serviço do *Firebase* será usado e então foi selecionado a opção "*Hosting"*.

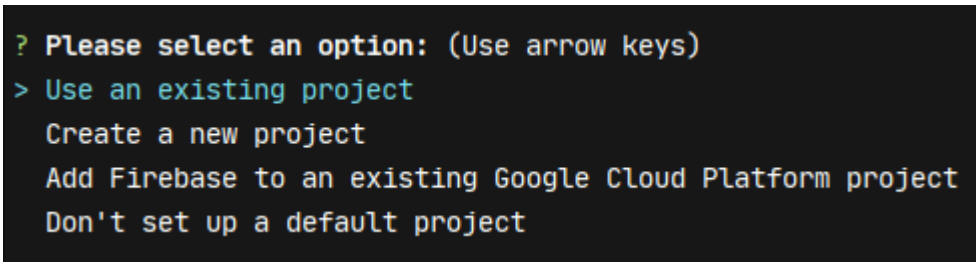
Figura 59 - Escolha do serviço.



Fonte: Do autor.

Neste ponto foi selecionada a opção "*Use an existing project* - Usar um projeto existente*"* que pode ser visto na Figura 60, pois o projeto já foi criado na *dashbord* do *Firebase.* Após isso uma lista com os projetos existentes é apresentada e então o projeto é selecionado.

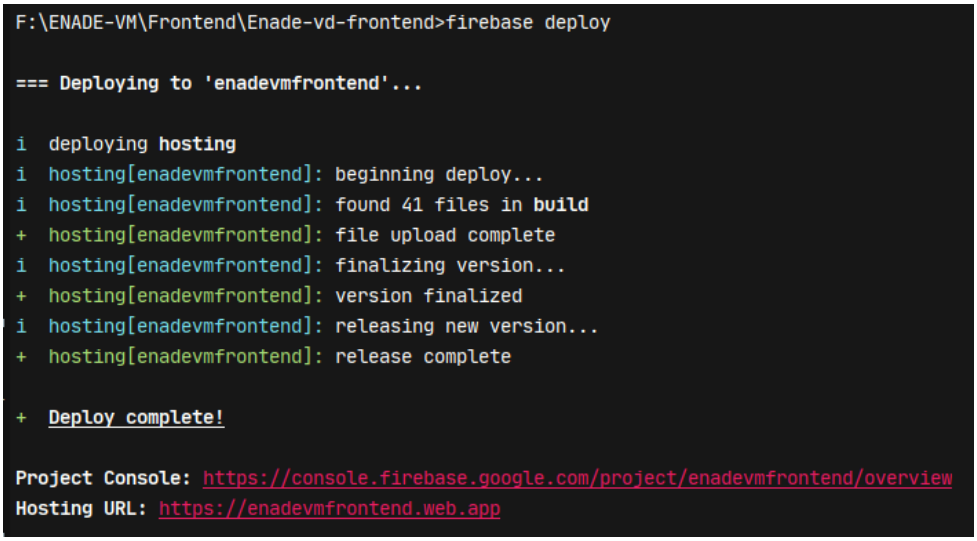
Figura 60 - Escolha de um projeto existente.



Fonte: Do autor.

- `firebase deploy`: com este comando o projeto é enviado para o servidor do *Firebase Hosting*. Na saída um link (https://enadevmfrontend.web.app) para acesso à plataforma é disponibilizado. A Figura 61 representa a saída para o comando.

Figura 61 - Saída para o comando *Firebase deploy*.



Fonte: Do autor.

Assim, após os comandos listados acima, a versão de produção gerada pelo comando `npm run build` é enviada para o servidor e com isso a plataforma fica disponibilizada para qualquer um que a queira acessar.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como resultado a construção de uma plataforma de visualização de dados, também elucidou sobre processos, técnicas e boas práticas referentes a visualização de dados. O desenvolvimento do trabalho teve como objetivo melhorar a apresentação dos dados disponibilizados atualmente na plataforma ENADE-DM, dados que são relativos aos gabaritos dos estudantes de ensino superior dos cursos de Computação que fizeram a prova do ENADE nos anos de 2008, 2011 e 2014. Dessa maneira, foi apresentado um referencial teórico sobre o tema abordado que apresenta fatos, conceitos e técnicas referentes a visualização de dados e também uma breve explanação sobre mineração de dados.

A partir das visualizações construídas, é possível analisar aspectos relacionados aos estudantes, às regiões de aplicação das provas e às diferenças entre os anos de aplicação do exame, assim como outros aspectos. Isso possibilita ao usuário a identificação, de acordo com seu interesse, de padrões e características obtidas a partir dos dados apresentados. Ainda, com auxílio de filtros o usuário pode escolher, por exemplo, um determinado ano em especifico ou o período todo que se tem minerado.

A partir do endereço https://enadevmfrontend.web.app/regioes é possível visualizar a plataforma ENADE-VM. A plataforma foi disponibilizada publicamente na internet de modo a ser acessada facilmente por qualquer pessoa interessada nos resultados. A princípio essa será utilizada como forma de visualização das informações, entretanto, seus resultados podem ser utilizados em conjunto com outras pesquisas com foco em regiões ou cursos específicos, de maneira a ampliar a descoberta de informações ou auxiliar em novas pesquisas.

Como sugestão de trabalhos futuros, novas visualizações podem ser criadas, com base em trabalhos de mineração de dados do mesmo âmbito. Assim, novas seções, módulos e melhorias podem ser adicionadas à plataforma ENADE-VM, de modo a abranger outros cursos, áreas de conhecimento e proporcionar cada vez mais uma melhor experiência para o usuário.

REFERÊNCIAS

ABDULLAH, Hanin M.; ZEKI, Ahmed M. Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example. **2014 3rd International Conference On Advanced Computer Science Applications And Technologies**, [s.l.], p.85-89, dez. 2014. IEEE. http://dx.doi.org/10.1109/acsat.2014.22.

ABOUT. **GitHub is how people build software**. Disponível em: <https://github.com/about>. Acesso em: 20 jan. 2020.

ALVES, Renato Marinho. **Plataforma de mineração e apresentação dos dados do ENADE da área de Computação dos anos de 2008 a 2014. 2018**. 75 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2018

ARCGIS. **Criar e utilizar um gráfico de linhas**. 2019. Disponível em: <https://doc.arcgis.com/pt-br/insights/create/line-graph.htm>. Acesso em: 11 set. 2019.

BARBOSA, Simone Diniz Junqueira. **Interação humano-computador.** Rio de Janeiro - Rj: Elsevier, 2010.

BOTELHO, Elisângela. **Visualização de dados como ferramenta de Classificação em sistemas de Bases de Dados para Data Mining**. 2002. 97 f. Dissertação (Mestrado) - Curso de Ciência da Computação e Matemática Computacional, Universidade de São Paulo, São Carlos - Sp, 2002.

BURN-MURDOCH, John. **Why you should never trust a data visualisation**. 2013. Disponível em: <https://www.theguardian.com/news/datablog/2013/jul/24/why-you-should-never-trust-a-data-visualisation>. Acesso em: 17 jan. 2020.

CABENA, Peter et al. **Discovering Data Mining**: From Concept to Implementation. New Jersey: Prentice Hall, 1998. 224 p

CAELUM. **Curso UX e Usabilidade Aplicados em Mobile e Web:** Padrões e Princípios do Design de Interação. Disponível em: <https://www.caelum.com.br/apostila-ux-usabilidade-mobile-web/principios/#as-dez-heursticas-de-nielsen>. Acesso em: 16 set. 2019.

CARD, Stuart K.; NEWELL, Allen; MORAN, Thomas P. **The Psychology of Human-Computer Interaction**. Michigan: L. Erlbaum Associates, 1983. 469 p.

CARVALHO, Edilson Alves de; ARAÚJO, Paulo César de. **Leituras cartográficas e interpretações estatísticas I: geografia**. Natal, Rn: Edufrn, 2008. 248 p

CARVALHO, Elizabeth Simão; MARCOS, Adérito Fernandes. **Visualização de informação**. Guimarães: Centro de Computação Gráfica, 2009.

CASAGRANDE, Thais. **Liberdade e controle do usuário**. 2013. Disponível em: <http://www.robsongalloppi.com.br/liberdade-e-controle-do-usuario/>. Acesso em: 16 set. 2019.

CONTRIBUTORS, Mdn. **O que é JavaScript?** 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript>. Acesso em: 18 out. 2019.

DEVELOPERS. **Google Charts: Quick Start**. 2019. Disponível em: <https://developers.google.com/chart/interactive/docs/quick_start>. Acesso em: 21 out. 2019.

DJANGOPROJECT. **Why Django?** Disponível em: <https://www.djangoproject.com/start/overview/>. Acesso em: 6 set. 2019.

DNPM. **Dados Abertos**. 2017. Disponível em: <http://www.anm.gov.br/acesso-a-informacao/dados-abertos>. Acesso em: 7 out. 2019.

ENADEDM. **Bem-vindo à EnadeDM!** 2018. Disponível em: <https://enadedm.herokuapp.com/inicio>. Acesso em: 7 out. 2019.

FALOUTSOS, Christos; LIN, King-ip. FastMap. **Proceedings Of The 1995 Acm Sigmod International Conference On Management Of Data - Sigmod '95**, [s.l.], p.1-25, 1995. ACM Press. http://dx.doi.org/10.1145/223784.223812.

FEW, Stephen. **Multivariate Analysis Using Parallel Coordinates**. 2006. Disponível em: <https://www.perceptualedge.com/articles/b-eye/parallel_coordinates.pdf>. Acesso em: 16 jan. 2020.

FEW, Stephen. **The Encyclopedia of Human-Computer Interaction, 2nd Ed.: 35. Data Visualization for Human Perception.** Disponível em: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception?refresh=true%3Frefresh%3Dtrue>. Acesso em: 24 out. 2019.

FM2S. O que é histograma? Quando utilizar? Como construir. Disponível em: <https://www.fm2s.com.br/histograma/>. Acesso em: 11 set. 2019.

FREITAS, Carla Maria dal Sasso et al. **Introdução à Visualização de Informações**. Rita, Pelotas, Rs, v. 8, n. 2, p.144-158, nov. 2001.

GERSHON, N.; EICK, S.g.. Information Visualization. **Ieee Computer Graphics And Applications**, [s.l.], v. 17, n. 4, p.29-31, jul. 1997. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/mcg.1997.595265.

GONÇALVES, Ariane. **O que é CSS? Guia Básico para iniciantes**. 2019. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>. Acesso em: 18 out. 2019.

GOOGLE. **Firebase Hosting**. Disponível em: https://firebase.google.com/docs/hosting?hl=pt-br. Acesso em: 09 abr. 2020.

HAND, David; MANNILA, Heikki; SMYTH, Padhraic. **Principles of Data Mining**. Cambridge: The Mit Press, 2001.

HEROKU. **What is Heroku?** Disponível em: <https://www.heroku.com/what>. Acesso em: 20 jan. 2020.

IBM. **Gráficos de área**. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSEP7J_11.0.0/com.ibm.swg.ba.cognos.ug_cr_pps.doc/c_as_ti_charts_area.html>. Acesso em: 11 set. 2019.

IEEE Transactions on visualization and computer graphics (IEEE VIS COMPUT GR). S.i.: Institute Of Electrical And Electronics Engineers, jan. 2002.

INCORPORATED, Adobe Systems. **Adobe Flex 3:** ADVANCED DATA VISUALIZATION DEVELOPER GUIDE. San Jose: S.i., 2008. 395 p.

INEP. **Sobre o Inep: Institucional.** Disponível em: <http://portal.inep.gov.br/web/guest/sobre-o-inep>. Acesso em: 7 out. 2019.

ISO 9241-210. **International Organization for Standardization**. 2010. Disponível em: <https://www.iso.org/standard/52075.html>. Acesso em: 5 set. 2019.

KHAN, Muzammil; KHAN, Sarwar Shah. Data and Information Visualization Methods, and Interactive Mechanisms:: A Survey. **International Journal Of Computer Applications**, S.i., v. 34, n. 1, p.1-14, nov. 2011. Disponível em: <https://research.ijcaonline.org/volume34/number1/pxc3875722.pdf>. Acesso em: 13 jan. 2020.

LEÃO, H. A. T. et al. Mining ENADE Data from the Ulbra Network Institution. **International Conference on Information Technology**: New Generations, Las Vegas, v. 15, n. 15, p. 1-20, out./2018. Disponível em: www.researchgate.net/publication/328451218_Mining_ENADE_Data_from_the_Ulbra_Network_Institution. Acesso em: 9 jun. 2020.

LIMA, Mateus Curcino de. **Combinando semi-supervisão e hubness para aprimorar o agrupamento de dados em alta dimensão**. 2017. 90 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Faculdade de Computação, Universidade Federal de Uberlândia, Uberlândia, 2017.

MATERIAL-UI. **About Us**. Disponível em: https://material-ui.com/company/about/. Acesso em: 2 maio 2020.

MICROSOFT. **Present your data in a bubble chart**. Disponível em: <https://support.office.com/en-US/article/Present-your-data-in-a-bubble-chart-424D7BDA-93E8-4983-9B51-C766F3E330D9>. Acesso em: 15 jan. 2020.

MOREIRA, Luiz Paulo. **O que é gráfico?** 2019. Disponível em: <https://brasilescola.uol.com.br/o-que-e/matematica/o-que-e-grafico.htm>. Acesso em: 3 set. 2019.

NARDI, Alexandre Ricardo. **Fundamentos e Modelagem de Bancos de Dados Multidimensionais**. 2007. Disponível em: <https://docs.microsoft.com/pt-br/previous-versions/technical-articles/cc518031(v=msdn.10)?redirectedfrom=MSDN#XSLTsection127121120120>. Acesso em: 16 jan. 2020.

NPMJS. **About npm**: npm is lots of things.. npm is lots of things.. Disponível em: https://www.npmjs.com/about. Acesso em: 10 abr. 2020.

OFFICE, Support. **Apresentar os dados em um gráfico de rosca**. 2013. Disponível em: <https://support.office.com/pt-br/article/apresentar-os-dados-em-um-gr%C3%A1fico-de-rosca-0ac0efde-34e2-4dc6-9b7f-ac93d1783353#top>. Acesso em: 14 jan. 2020.

OLIVEIRA, William. **O universo da programação: Um guia de carreira em desenvolvimento de software**. [s. L.]: Editora Casa do Código, 2018. 222 p.

PAULA, Melise Maria Veiga de et al. A Visualização de Informação e a Transparência de Dados Públicos. In: **Simpósio Brasileiro de sistemas de informação, 7., 2011**, São Paulo. Anais. S.i.: Sbc, 2011. p. 384 - 395.

PEREIRA, Caio Ribeiro. **Construindo APIs REST com Node.js** [s. L.]: Editora Casa do Código, 2016. 193 p.

PORTALACTION. **1.8 - Gráfico de pizza.** Disponível em: <http://www.portalaction.com.br/estatistica-basica/18-grafico-de-pizza>. Acesso em: 3 set. 2019.

RABAIOLI, Marcos. **Criando Uma API Rest Utilizando Django Rest Framework — Parte 1**. 2017. Disponível em: <https://medium.com/@marcosrabaioli/criando-uma-api-rest-utilizando-django-rest-framework-parte-1-55ac3e394fa>. Acesso em: 27 out. 2019.

ROBERTO, João. **O que é Laravel? Porque usá-lo?** 2017. Disponível em: <https://medium.com/joaorobertopb/o-que-é-laravel-porque-usá-lo-955c95d2453d>. Acesso em: 6 set. 2019.

ROSSINI, L. A. S.; SILVA, R. R. DE P.; SOTTO, E. C. S.; ARAÚJO, L. S. **Data mining: conceitos e consequências.** Revista Interface Tecnológica, v. 15, n. 2, p. 50-59, 30 dez. 2018.

ROUSE, Margaret; CHURCHVILLE, Fred; DANG, Mike. **What is user interface (UI)?** 2005. Disponível em: <https://searchapparchitecture.techtarget.com/definition/user-interface-UI>. Acesso em: 4 set. 2019.

RUBYONRAILS. **Imagine what you could build if you learned Ruby on Rails…** disponível em: <https://rubyonrails.org>. Acesso em: 9 set. 2019.

SALKIND, Neil J. **Statistics for People Who (Think They) Hate Statistics:** Using Microsoft Excel 2016. 4. ed. Nova York: Sage Publications, 2016. 544 p.

SANTIAGO, Vitor. **O que é Histograma?** 2018. Disponível em: <https://certificacaoiso.com.br/o-que-e-histograma/>. Acesso em: 3 set. 2019

SBC. **Interação Humano-Computador**. 2019. Disponível em: <https://www.sbc.org.br/14-comissoes/390-interacao-humano-computador>. Acesso em: 15 set. 2013.

SHNEIDERMAN, Ben. **Treemaps for space-constrained visualization of hierarchies.** 2009. Disponível em: <http://www.ifs.tuwien.ac.at/~silvia/wien/vu-infovis/articles/shneiderman_treemap-history_1998-2009.pdf>. Acesso em: 16 jan. 2020.

SICARIUS. **Blur Screenshots (Windows).** 2010. Disponível em: <https://www.mobygames.com/game/windows/blur/screenshots/gameShotId,445464/>. Acesso em: 16 set. 2019.

SIGNIFICADOS. **Significado de HTML: O que é HTML**. 2018. Disponível em: <significados.com.br/html/>. Acesso em: 19 out. 2019.

SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. **Introdução à mineração de dados: com aplicações em R**. Rio de Janeiro: Elsevier, 2016.

SQLITE. **About SQLite**. Disponível em: https://www.sqlite.org/about.html. Acesso em: 1 maio 2020.

STAUFFER, Matt. **Desenvolvendo com Laravel:** Um framework para a construção de aplicativos PHP modernos. [s. L.]: Novatec Editora, 2019.

TIBCO. **O que é um Gráfico de Pizza**. Disponível em: <https://docs.tibco.com/pub/spotfire_web_player/6.0.0-november-2013/pt-BR/WebHelp/GUID-8B1036A5-6BE9-4A84-B532-8E15060CABA9.html>. Acesso em: 11 set. 2019.

TRONCHONI, Alex B. et al. Descoberta de conhecimento em base de dados de eventos de desligamentos de empresas de distribuição. **Sba**: Controle & Automação Sociedade Brasileira de Automatica, [s.l.], v. 21, n. 2, p.185-200, abr. 2010. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/s0103-17592010000200007.

USAGE: **Material Design**. Disponível em: https://react-google-charts.com/bar-chart. Acesso em: 08 abr. 2020.

VISUALSTUDIO. **Getting Started**. 2019. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 18 out. 2019.

VUEJS. **Introduction: What is Vue.js?** 2019. Disponível em: <https://vuejs.org/v2/guide/>. Acesso em: 10 out. 2019.

W3SCHOOLS. **Bootstrap 4 Introdução:** O que é o Bootstrap? 2019. Disponível em: <https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp>. Acesso em: 18 out. 2019.

WARD, Matthew. **Overview of data visualization**. 2013. Disponível em: <http://web.cs.wpi.edu/~matt/courses/cs563/talks/datavis.html>. Acesso em: 17 jan. 2020.

WYDAY. **Windows 7 Taskbar Progress Bar with C# and .NET**. 2009. Disponível em: <https://wyday.com/blog/2009/windows-7-taskbar-progress-bar-with-csharp-and-dotnet/>. Acesso em: 16 set. 2019.