



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Geração Semi Automática de Gráficos Para Jornalismo de Dados Usando Dados Abertos: Um Estudo de Caso do Censo da Educação Superior.

Felipe C. P. Magalhães

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador

Prof. Dr. Edison Ishikawa

Coorientadora

Prof.a Dr.a Suzana Guedes Cardoso

Brasília
2023

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

M189g Magalhães, Felipe
Geração Semi Automática de Gráficos Para Jornalismo de
Dados Usando Dados Abertos: Um Estudo de Caso do Censo da
Educação Superior / Felipe Magalhães; orientador Edison
Ishikawa; co-orientador Suzana Guedes Cardoso. -- Brasília,
2023.
93 p.

Monografia (Graduação - Computação) -- Universidade de
Brasília, 2023.

1. Jornalismo de Dados. 2. Dados Abertos. 3. Censo da
Educação Superior. 4. INEP. 5. Gráficos. I. Ishikawa,
Edison, orient. II. Guedes Cardoso, Suzana, co-orient. III.
Título.

Dedicatória

Dedico este trabalho aos meus pais, pilares fundamentais em minha vida, que sempre acreditaram na educação como o grande agente de transformação social que desejamos.

Agradecimentos

Agradeço primeiramente a Deus que jamais me desamparou mesmo nos momentos mais difíceis, me dando forças para continuar e vencer cada etapa dessa graduação.

Agradeço a toda minha família especialmente aos meus pais que pavimentaram meu caminho auxiliando no ingresso nesta universidade.

Agradeço a todos educadores que ao longo da jornada acadêmica compartilharam seus dons e conhecimentos, possibilitando meu crescimento científico e profissional. Especialmente ao meu orientador, Prof. Dr. Edison Ishikawa que com muita paciência e resiliência, me conduziu até o resultado desse projeto desafiador. Além de ter estado próximo em toda graduação, me possibilitando ser monitor e tutor de disciplinas por ele ministradas.

Agradeço também a todos amigos e colegas que pude fazer durante essa incrível e transformadora jornada, que ao longo de anos dividimos horas de estudos e vivências.

Agradeço também a mim mesmo por ter acreditado no meu potencial, quebrando padrões limitantes e sociais, crendo sempre que esse dia chegaria.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Jornalismo Computacional se refere às ferramentas e algoritmos que os jornalistas usam para contar histórias. Neste contexto os dados abertos podem ser a fonte da notícia. Diferentes técnicas de manipulação de dados como *Data Warehouse*, *Data Mining* e *Business Intelligence*, têm sido aplicadas para responder perguntas sobre dados, porém, não se fazem por meio interativo, tornando suas respostas estáticas às perguntas definidas previamente. Essas tecnologias demandam uma alta complexidade e custo operacional elevado para extrair tais informações. Esse projeto visou criar uma aplicação capaz de gerar gráficos dinâmicos a partir da seleção instantânea de filtros sobre os dados do Censo da Educação Superior do Brasil, tendo como objetivo facilitar a interpretação desses dados por jornalistas. O protótipo implementado foi avaliado usando a escala *System Usability Scale* obtendo resultados satisfatórios em relação à usabilidade do sistema. Desta forma, tornou-se possível a extração de novas informações que poderão se concretizar em matérias jornalísticas.

Palavras-chave: Jornalismo de Dados, Dados Abertos, Censo da Educação Superior, INEP, Gráficos

Abstract

Computational Journalism is related to the tools and algorithms that journalists use to tell stories. In this context, open data can be the source of news. Different data manipulation techniques as Data Warehouses, Business Intelligence and Data Mining, have been applied to answer questions about data, but they are not done interactively, making their answers static and previously defined. These technologies demand a high complexity and expensive operational cost to extract this information. This project aimed to create a web application capable of generating dynamic graphics by on time selection about filters on data about the Higher Education College Census of Brazil, with the objective of facilitating the interpretation of these data by journalists community. The prototype implemented was evaluated using the System Usability Scale, obtaining satisfactory results regarding the usability of the system. In this way, it becomes possible to extract new information that can become journalistic articles.

Keywords: Data Journalism, Open Data, Censo da Educação Superior do Brasil, INEP, Graphics

Sumário

1	Introdução	1
1.1	Identificação do Problema	2
1.2	Justificativa	3
1.3	Objetivo Geral	3
1.4	Objetivos Específico	3
1.5	Metodologia	3
1.6	Descrição dos Capítulos	4
2	Referencial Teórico	5
2.1	Dados Abertos	5
2.2	Dados Abertos Governamentais	6
2.3	Jornalismo de Dados	6
2.4	Trabalhos Relacionados	7
2.5	Desafio do Projeto	9
3	Modelagem da Solução	10
3.1	Tipo de Arquitetura	10
3.2	Desenho do Dicionário de Dados Gráfico	11
4	Implementação do Protótipo	16
4.1	Tecnologias Empregadas	16
4.1.1	JavaScript	16
4.1.2	React.js	17
4.1.3	Next.js	18
4.1.4	Axios	19
4.1.5	Node.js	19
4.1.6	Express.js	20
4.1.7	Knex.js	20
4.1.8	PostgreSQL	21
4.1.9	Heroku	21

4.2	Manipulação dos Dados	21
4.3	Camada de Aplicação - Backend	24
4.4	Camada de Apresentação - Frontend	25
4.5	Implantação da Aplicação	27
4.6	Apresentação do Protótipo	27
4.6.1	Página Principal	27
4.6.2	Seleção de Anos de Pesquisa	27
4.6.3	Seleção de Filtros de Instituição de Ensino	28
4.6.4	Seleção de Filtros de Curso	29
4.6.5	Seleção de Filtros de Aluno	29
4.6.6	Seleção de Modelo Gráfico	30
4.6.7	Resultado Final	31
4.6.8	Página do Dicionário de Dados Gráfico	31
5	Resultados	34
5.1	Estatísticas Censo da Educação Superior	34
5.2	System Usability Scale	34
5.3	Estatísticas de Usabilidade	35
6	Conclusões e Trabalhos Futuros	44
6.1	Conclusões	44
6.2	Limitações	44
6.3	Trabalhos Futuros	45
	Referências	46
	Anexo	48
I	Modelos lógicos para criação do banco de dados	49
II	Queries SQL executadas para criação do banco de dados	54

Lista de Figuras

3.1	Padrão arquitetural para aplicação <i>three-tier</i> . Fonte: Three-tier architecture overview[1].	13
3.2	Mapa conceitual dos microdados do CES. Fonte: própria (2021).	14
3.3	Dicionário de dados gráfico gerada a partir dos módulos Aluno, Cursos, IES. Fonte: própria (2021).	15
4.1	Arquitetura three-tier e tecnologias utilizadas no protótipo. Fonte: própria (2023).	17
4.2	Linguagens de programação mais conhecidas 2018 - 2020. Fonte: RackerRank [2].	18
4.3	Frameworks mais conhecidas 2018 - 2020. Fonte: RackerRank [2].	18
4.4	Modelo Lógico contemplando entidades Alunos, Cursos, IES e tabelas auxiliares. Fonte: própria (2022).	22
4.5	Página principal da aplicação. Fonte: própria (2023).	28
4.6	Seleção de anos a serem pesquisados. Fonte: própria (2023).	28
4.7	Seleção de filtros de instituição de ensino. Fonte: própria (2023).	29
4.8	Seleção de filtros de curso. Fonte: própria (2023).	30
4.9	Seleção de filtros de alunos. Fonte: própria (2023).	30
4.10	Seleção de tipo de gráfico. Fonte: própria (2023).	31
4.11	Apresentação do gráfico com os resultados. Fonte: própria (2023).	32
4.12	Apresentação do esquema do dicionário de dados gráfico com todos os campos possíveis de serem utilizados na consulta. Fonte: própria (2023).	33
5.1	Participantes. Fonte: própria (2023).	36
5.2	Porcentagem de jornalistas e não jornalistas Fonte: própria (2023).	36
5.3	Anos de profissão caso seja jornalista. Fonte: própria (2023).	37
5.4	Resultados pergunta 1. Fonte: própria (2023).	38
5.5	Resultados pergunta 2. Fonte: própria (2023).	38
5.6	Resultados pergunta 3. Fonte: própria (2023).	39
5.7	Resultados pergunta 4. Fonte: própria (2023).	39

5.8	Resultados pergunta 5. Fonte: própria (2023).	40
5.9	Resultados pergunta 6. Fonte: própria (2023).	40
5.10	Resultados pergunta 7. Fonte: própria (2023).	41
5.11	Resultados pergunta 8. Fonte: própria (2023).	41
5.12	Resultados pergunta 9. Fonte: própria (2023).	42
5.13	Resultados pergunta 10. Fonte: própria (2023).	42
5.14	Resultados Final Teste Usabilidade. Fonte: própria (2023).	43
I.1	Modelo Lógico Entidade Aluno. Fonte: própria (2022).	50
I.2	Modelo Lógico Entidade Curso. Fonte: própria (2022).	51
I.3	Modelo Lógico Entidade IES. Fonte: própria (2022).	52
I.4	Modelo Lógico Final. Fonte: própria (2022).	53

Lista de Tabelas

4.1	Dimensões de tabelas eletrônicas dos microdados de 2019.	23
-----	--	----

Lista de Abreviaturas e Siglas

API Application Programming Interface.

BI Business Intelligence.

CDN Content Delivery Network.

CES Censo da Educação Superior.

CMS Sistema de Gestão de Conteúdo.

CRUD Create, Read, Update, Delete.

DOM Document Object Model.

DSR Design Science Research.

HTML Hyper Text Markup Language.

HTTP Hypertext Transfer Protocol.

IDC International Data Corporation.

IES Instituições de Educação Superior.

INEP Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira.

JSON JavaScript Object Notation.

MEC Ministério da Educação.

MIT Massachusetts Institute of Technology.

NOSQL No Structured Query Language.

RDF Resource Description Framework.

REST Representational State Transfer.

SPA Single Page Application.

SQL Structured Query Language.

SUS System Usability Scale.

UI User Interface.

URL Uniform Resource Locator.

Capítulo 1

Introdução

As novas formas de armazenamento de dados na web, tem permitido a descoberta e publicação de informações que possibilitam a geração de novos conhecimentos com acesso a toda população, o que em muitos casos era antes restrito por se tratar de informações limitadas a determinados grupos de pessoas [3]. Essas informações quando disponíveis de forma livre, são classificados como conhecimentos abertos e segundo a Open Definition [4], o conhecimento é aberto se qualquer pessoa está livre para acessá-lo, utilizá-lo, modificá-lo, e compartilhá-lo — restrito, no máximo, a medidas que preservam a proveniência e abertura. Esse tipo de conhecimento aberto, tem crescido exponencialmente ao passo que a humanidade está cada vez mais conectada às redes de intercomunicação como a internet, produzindo e consumindo informações numa velocidade nunca antes vista. A International Data Corporation - IDC [5] prevê que até o ano de 2025 a esfera de dados global aumentará para 163 *zettabytes*, isso corresponde mais de um trilhão de *gigabytes*. Seguindo a mesma ótica de conhecimento aberto, dados abertos para serem assim classificados necessitam ser livremente usados, reutilizados e redistribuídos, sendo necessário apenas a citação de sua fonte [6].

Com base nesse contexto, este trabalho visou explorar os dados abertos do Censo da Educação Superior do Brasil. Estas informações são fornecidas de forma anual pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira - INEP, onde são disponibilizados dados sobre as instituições de educação superior - IES do país, além de seus cursos ofertados, docentes e alunos. Estes indicadores têm por objetivo, segundo o próprio INEP, oferecer à comunidade acadêmica e à sociedade em geral, informações detalhadas sobre a situação e as grandes tendências do setor. A análise destes dados viabiliza a geração de novos conhecimentos por meio de consultas voltadas ao ambiente de jornalismo computacional. Partindo do cruzamento de dados selecionados pelo usuário, são gerados gráficos a fim de facilitar a compreensão desses números, exibindo informações visuais através de modelos gráficos. Objetivou-se desta forma, ampliar o entendimento

estatístico obtido por meio do censo.

A grande massa de informação gerada hoje pelos meios computacionais, diferentemente da produção humana onde quem coleta os dados tem seu contato associado diretamente aos fatos e sua origem, pode em muitos casos se resumir a informações numéricas, em sequências de 0 e 1, processadas de forma automáticas e muitas vezes incompreensíveis ao entendimento comum. As informações qualitativas, permitem ao pesquisador, um alto grau de aprofundamento e especificação, visto que existe o contato direto com o observado, entrevistando-o, analisando seu comportamento, e por meio de suas opiniões e visões de mundo, ocorre a coleta desses dados. Diferentemente desta abordagem, os dados que são objetos deste estudo são quantitativos. Isto significa dizer que esses dados sendo de ordem estatística, ofertam um maior potencial de extração de inferências, identificação de padrões e tendências. Ao mesmo tempo que são mais difíceis de serem compreendidos tanto por seu pesquisador quanto pelo público-alvo, são também criticados por reduzirem seu conteúdo a modelos extremamente simplistas [3].

A fim de estabelecer ligações para efetuar inferências sobre os dados, a utilização de um dicionário de dados pode ser necessária quando se trata de situações complexas. Um dicionário de dados é uma compilação de definições de dados para vários elementos que existam em um armazenamento de dados. Para sistemas que fornecem dados, os dicionários de dados devem ser mantidos para orientar os usuários de dados no significado e seu uso de forma adequada. Com intuito de promover a comunicação e trazer este significado, os dicionários de dados devem ser acessíveis a todos usuários que inserem ou extraem informações de um armazenamento [7]. Existem várias patentes relacionadas ao uso de dicionários para organizar metadados [8], no entanto, os artigos publicados que mencionam dicionários de dados tendem a abster-se de incluir o formalismo associado.

Desta forma, fez-se necessário a criação de uma interface amigável ao jornalista, de forma a auxiliar na tradução desses números em informação visual de fácil compreensão, não exigindo deste usuário, conhecimentos aprofundados no ambiente computacional.

1.1 Identificação do Problema

Os dados do Censo da Educação Superior (CES) são disponibilizados pelo INEP no formato de planilhas eletrônicas com extensão “.csv” em uma linha histórica que decorre do ano de 1995 a 2021, até a data de publicação deste trabalho. A falta de um dicionário de dados de fácil entendimento que defina as relações entre esses dados de forma analítica, e uma aplicação que dê poder de decisão ao usuário, dificulta realizar inferências capazes de gerar novas informações relevantes para a geração de textos jornalísticos.

1.2 Justificativa

Este trabalho permite aos jornalistas realizarem o cruzamento dos dados do CES disponibilizados pelo INEP, gerando assim novas informações a partir das quais pode-se fazer inferências que auxiliem na redação das matérias jornalísticas não se atendo somente aos dados puros coletados.

1.3 Objetivo Geral

Este trabalho teve por objetivo geral, implementar um protótipo capaz de relacionar os microdados disponibilizados pelo INEP, para serem apresentados de forma gráfica de acordo com o critério de seleção e cruzamento dos dados realizados pelo usuário.

1.4 Objetivos Específico

Para atingir o objetivo geral, os seguintes objetivos específicos foram definidos:

- Criar um mapa conceitual entre todas as entidades e tabelas utilizando o programa CMapTools.
- Criar um dicionário de dados de forma gráfica com informações dos alunos sendo objeto principal de estudo utilizando a plataforma Miro.
- Realizar a análise, filtragem, organização e população dos dados na base de dados PostgreSQL.
- Realizar a implementação de uma API que possa ser consumida pela interface gráfica utilizando do *framework* web Express.js.
- Realizar a implementação da interface gráfica que será utilizada pelo jornalista fazendo uso da biblioteca React.js.
- Apresentar os gráficos resultantes do cruzamento de dados selecionados previamente pelo usuário.

1.5 Metodologia

Este trabalho utilizou-se da metodologia de desenvolvimento Design Science Research (DSR) [9] adaptada às necessidades deste projeto, onde foram seguidas as seguintes etapas:

1. **Identificação do problema:** Questão de pesquisa formalizada.

2. **Conscientização do problema:** Formalização das faces do problema, compreensão do ambiente externo, requisitos do artefato.
3. **Identificação das classes de problemas e artefatos existentes:** Classes de Problemas e artefatos identificados, soluções satisfatórias explicitadas.
4. **Proposição do artefato para resolver o problema específico:** Proposta de artefatos formalizada.
5. **Projeto do artefato selecionado:** Projeto explicitando técnicas e ferramentas para o desenvolvimento e a avaliação do artefato, e detalhamento dos requisitos do artefato.
6. **Desenvolvimento do artefato:** Heurística de construção, artefato em seu estado funcional.
7. **Avaliação do artefato:** Heurísticas contingenciais, artefato avaliado.
8. **Conclusão:** Resultados da pesquisa, principais decisões tomadas e limitações da pesquisa.
9. **Comunicação dos resultados:** Publicação em jornais, revistas setoriais, seminários, congressos, etc.

1.6 Descrição dos Capítulos

Para atingir o objetivo proposto, este trabalho foi estruturado da seguinte maneira:

1. **Capítulo 1:** Contextualização do problema.
2. **Capítulo 2:** Fundamentação teórica dos principais conceitos empregados no projeto e revisão de trabalhos de outros autores.
3. **Capítulo 3:** Descrição da arquitetura da solução proposta e o dicionário dos dados.
4. **Capítulo 4:** Apresentação das tecnologias empregadas, etapas de implementação e funcionamento do protótipo.
5. **Capítulo 5:** Exposição dos resultados dos testes com usuários.
6. **Capítulo 6:** Apresentação das conclusões e sugestões futuras.

Capítulo 2

Referencial Teórico

Este capítulo apresenta o referencial teórico necessário para o estudo de caso do projeto além de trabalhos anteriores relacionados com Dados Abertos Conectados e geração de gráficos. Este capítulo está organizado da seguinte forma: a Sessão 2.1 descreve o conceito de dados abertos, a Sessão 2.2 detalha o que são dados abertos governamentais, a Sessão 2.3 aborda os principais conceitos sobre Jornalismo de Dado e a Sessão 2.4 faz uma revisão sobre os trabalhos relacionados anteriormente com temáticas próximas a este projeto, a Sessão 2.5 faz uma análise resumida dos trabalhos relacionados e explicita os desafios que foram enfrentados em seguida.

2.1 Dados Abertos

Pela definição da Open Knowledge Foundation [4] “*Dados abertos são dados que podem ser livremente usados, reutilizados e redistribuídos por qualquer pessoa - sujeitos, no máximo, à exigência de atribuição da fonte e compartilhamento pelas mesmas regras*”. Ainda segundo esta definição, dados abertos carregam três normas ditas fundamentais que são descritas a seguir:

- **Disponibilidade e acesso:** de forma preferencial, os dados devem estar armazenados na internet de forma conveniente e modificável, sendo aberto e acessível a qualquer pessoa.
- **Reuso e redistribuição:** termos que permitam a reutilização e redistribuição, sujeitos no máximo à citação da fonte original e compartilhamento das mesmas licenças que as informações foram compartilhadas.
- **Participação universal:** qualquer pessoa deve ser capaz de ter acesso aos dados, excluindo de restrições de uso não comercial e restrições de utilização dos dados para fins específicos.

2.2 Dados Abertos Governamentais

Os dados disponibilizados pelo INEP que são objetos deste estudo, se classificam como dados abertos governamentais. Segundo a World Wide Web Consortium - W3C [10] "*Dados Abertos Governamentais são a publicação e a disseminação das informações do setor público na web, compartilhadas em formato bruto aberto, compreensíveis logicamente, de modo a permitir sua reutilização em aplicações digitais desenvolvidas pela sociedade*". Inspirados pela filosofia do código aberto fundamentadas pelos princípios de abertura, participação e colaboração, entendendo que dados governamentais são propriedades comuns assim como informações científicas.

O termo Dados Conectados se refere a um conjunto de boas práticas para publicação e conexão de dados estruturados na web, usando padrões internacionais recomendados pelo W3C. No entanto Dados Conectados não necessariamente precisam ser abertos. O sistema cinco estrelas é utilizado para fazer a classificação dos graus de abertura dos dados. As 5 estrelas para Dados Abertos são [11]:

1. Disponível na Internet (em qualquer formato; por exemplo, PDF), desde que com licença aberta, para que seja considerado Dado Aberto.
2. Disponível na Internet de maneira estruturada (em um arquivo Excel com extensão XLS).
3. Disponível na Internet, de maneira estruturada e em formato não proprietário (CSV em vez de Excel).
4. Seguindo todas as regras anteriores, mas dentro dos padrões estabelecidos pelo W3C (RDF e SPARQL): usar URL para identificar coisas e propriedades, de forma que as pessoas possam direcionar para suas publicações.
5. Todas as regras anteriores, mais: conectar seus dados a outros dados, de forma a fornecer um contexto.

Os dados do Censo da Educação superior fornecidos pelo INEP podem ser classificados com três estrelas, pois são disponibilizados na internet de forma aberta em formato não proprietário CSV *Comma-Separated Values*.

2.3 Jornalismo de Dados

Vivendo hoje em um mundo digital, dados não se restringem somente a planilhas com dados numéricos. Fotos, vídeos e áudios, são descritos com os mesmos valores binários que quaisquer outros dados disponíveis na web. O que faz o jornalismo de dados diferente de

outros jornalismos é a possibilidade de combinar o faro da notícia com a enorme variedade de informações digitais disponíveis. Jornalismo de dados pode ajudar jornalistas a contar histórias complexas por meio de infográficos que prendam a atenção. Os dados podem ser utilizados como a fonte do jornalismo de dados ou podem ser ferramentas com a qual se conta histórias [12].

O jornalismo de dados é a área do jornalismo que pode ser definida como conjunto de práticas para coletar, analisar, visualizar e publicar dados para fins jornalísticos, assim define Berret e Philips (2016) [13]. Outra definição de Howard (2014) coloca o jornalismo ao lado de dados científicos e define o jornalismo de dados como uma *“aplicação da ciência de dados ao jornalismo, onde a ciência de dados é definida como o estudo da extração de conhecimento a partir de dados”* [14]. Howard considera o jornalismo de dados como um processo que engloba reunir, limpar, organizar, analisar, visualizar e publicar dados para apoiar a criação de atos de jornalismo. Ainda segundo Heravi (2019) jornalismo de dados é *“encontrar histórias em dados – histórias que são de interesse do público – e apresentá-las da maneira mais apropriada para uso público e reutilização”* [15].

Portanto o jornalismo de dados promove uma abordagem científica e baseada em fatos. Essa abordagem exige que o jornalismo seja tratado e praticado da mesma forma que as investigações científicas, convidando a métodos científicos, objetividade científica, transparência científica e reprodutibilidade científica. Esses métodos científicos incluem análise de dados quantitativos e qualitativos para investigação, produção de conteúdo jornalístico e comunicação desses conteúdos ao público.

2.4 Trabalhos Relacionados

Desde o advento da Lei 12.527, de 18 de novembro de 2011, Lei de Acesso à Informação que legisla sobre dados abertos, muitos trabalhos veem sendo realizados a fim de melhor organizar esses indicadores ou fazer novas medições com propósito de ampliar os horizontes sobre esses dados públicos.

O trabalho de Halley e Cardoso (2017) traz um exemplo prático desse tipo de extração [16]. Nele os dados do Censo da Educação Superior são utilizados para realizar consultas analíticas, resultantes de um processo de extração e transformação dos dados dos micro-dados, para posterior carga em um Data Warehouse. Utilizaram se dos dados do censo 1996 a 2014 para responder algumas perguntas já previamente definidas como:

1. Quantidade de alunos matriculados no Ensino Superior por curso.
2. Sexo dos alunos do ensino superior por grupo de faixa etária.
3. Raça/cor dos alunos do Ensino Superior por Ano de Ingresso.

Nesse estudo os gráficos foram gerados pela ferramenta geradora de *dashboard Pentaho BI-Server*.

Em [17], Silva F. e Netto B. (2016) realizam análise sobre os dados Programa Universidade Para Todos - PROUNI, utilizando a ferramenta de visualização de dados *Tableau Public* para gerar gráficos interativos possibilitando analisar dados referentes ao período de 2005 a 2016, verificando o desempenho do programa desde o início de sua criação, traçando o perfil dos beneficiários desse tipo de programa e realizando um comparativo com outras informações sobre a população brasileira. Os seguintes objetivos de estudo foram realizados nesse experimento:

1. Comparar o número de bolsas concedidas por raça/cor, sexo e deficiência em relação ao total da população brasileira.
2. Distinguir as regiões com mais beneficiários.
3. Apresentar o tipo de bolsa mais procurada, assim como a modalidade de ensino e o turno.
4. Identificar as instituições e cursos que possuem um maior número de bolsistas.
5. Verificar se houve crescimento do programa.

Segundo os autores, a utilização de ferramenta, possibilita criar e compartilhar rapidamente gráficos interativos, mapas e painéis em tempo real de forma fácil, tornando possível a compreensão e interpretação destes por agentes humanos, que posteriormente podem auxiliar na tomada de decisões importantes. Inicialmente os dados passam por um processo de limpeza assim como trabalho realizado com Data Warehouse, a fim de garantir a consistência dos dados. Após isso todos os dados de cada ano são agrupados com objetivo de gerar uma grande base de dados que será utilizada para montagem dos *dashboards*.

Como último caso, podemos citar o trabalho de Araújo e Souza (2011) que utilizou a prática de dados ligados para criar um *data set* de um conjunto de dados abertos de políticos brasileiros com informações coletadas em diferentes fontes [18]. Para isso, este projeto se utilizou do contexto de web semântica criado por Tim Berners-Lee que prevê a organização das informações de maneira legível para computadores através de padrões de formatação de dados como o RDF. Para isso as informações coletadas de diferentes fontes foram agrupadas e transformadas em dados ligados. A captura desses dados foi realizada por meio de um *Web Crawler*, desenvolvido para extrair os dados de forma automatizada de diferentes páginas HTML. Após a obtenção de todos dados estruturados, a informação foi representada utilizando o modelo RDF. Em seguida foram escolhidos vocabulários para representar as propriedades dos recursos, propriedades essas que definem o tipo de relacionamento entre eles. Após todo um processamento mais técnico dessas informações

o *data set* foi adicionado em um repositório para que pudesse ser encontrado por outras máquinas e humanos. A representação HTML para consulta e visualização dos dados foi a última camada a ser desenvolvida, permitindo realizar buscas pelos políticos por meio de diferentes critérios como nome, situação, cargo, estado, partido e sexo. Com o *data set* de dados de políticos brasileiros online, foi possível extrair os dados publicados via RDF, realizar consultas em cima desses dados, ligar esses dados com dados de outros *data set* e desenvolver diferentes aplicações web.

2.5 Desafio do Projeto

Como observado por trabalhos anteriores, o desafio de organizar informações públicas e disponibilizá-las de forma adequada para serem consultadas é um grande desafio. Existem várias técnicas que podem ser aplicadas desde a preparação desses dados até sua exibição forma concisa. Porém, a grande maioria dessas técnicas preveem uma pré-seleção de perguntas que o modelo irá responder. Nesse aspecto, este trabalho se organiza de maneira diferente, permitindo o usuário realizar a seleção dos seus filtros com menos restrições, ampliando assim, o número de possibilidades de informações que podem ser geradas com o cruzamento dos dados.

Capítulo 3

Modelagem da Solução

Este capítulo apresenta a metodologia empregada no desenvolvimento do projeto, desde a modelagem da solução proposta à elaboração do dicionário de dados gráfico para entendimento do relacionamento dos microdados, que foram fundamentais para posterior geração do banco de dados. Este capítulo está organizado da seguinte forma: A Sessão 3.1 detalha o tipo da arquitetura escolhida para o desenvolvimento do protótipo e a Sessão 3.2 refere-se às etapas de análise dos dados e desenho do dicionário de dados gráfico.

3.1 Tipo de Arquitetura

Em engenharia de software, camadas é um modo de separar responsabilidades e gerenciar dependências, assim, cada camada tem uma responsabilidade específica. Uma camada superior pode usar serviços em uma camada inferior, mas não o contrário. Essas camadas podem ser separadas fisicamente e executarem em máquinas distintas, sendo que uma camada pode se conectar à outra diretamente ou utilizando serviços de mensagens assíncronas. Fisicamente, separar camadas possibilita escalabilidade, porém adiciona latência advinda da comunicação de rede adicional.

A arquitetura em múltiplas camadas ou n-camadas, possibilita que a lógica de negócios e o gerenciamento de dados sejam fisicamente separadas. Aplicações n-camadas fornecem um modelo em que desenvolvedores podem criar aplicações flexíveis e reutilizáveis. Uma aplicação n-camadas pode ter uma arquitetura de camada fechada ou uma arquitetura de camada aberta. Em uma arquitetura fechada, uma camada pode se comunicar unicamente com a camada imediatamente próxima. Separando uma aplicação em camadas, os desenvolvedores podem assegurar a opção de modificar uma camada específica ou adicionar uma nova camada ao invés de trabalhar na aplicação inteira. Como benefícios desse tipo de arquitetura podemos citar [19]:

- Portabilidade entre nuvem e local, e entre plataformas de nuvem.

- Menos curva de aprendizado para a maioria dos desenvolvedores.
- Evolução natural do modelo de aplicação tradicional.
- Aberto para ambiente heterogêneo (Windows/Linux)

A estrutura escolhida para implementação do protótipo desse trabalho é também a mais difundida na arquitetura em múltiplas camadas e é denominada *three-tier* ou arquitetura em três camadas. Na arquitetura *three-tier* há subsistemas separados para as seguintes camadas [20]:

Apresentação: Camada responsável pela exibição da interface do usuário. Por meio dela o usuário é capaz de interagir com a aplicação (*presentation tier*).

Aplicação: Camada responsável pelo controle de toda lógica de negócios, bem como comunicação com o banco de dados (*logic tier*).

Dados: Camada responsável pelo armazenamento de todos os dados necessários ao funcionamento da aplicação (*data tier*).

É importante destacar que a arquitetura *n-tier* é por vezes confundida de forma equivocada com *n-layer*, usando termos como *presentation layer* ou *application layer*. Enquanto *layer* refere-se a uma divisão funcional do software, ou seja, os *layers* podem ser subdivisões de um mesmo sistema agrupadas em pastas diferentes e executadas em um mesmo *tier*. Enquanto uma estrutura *n-tier* representa uma divisão física onde cada camada é executada em máquinas diferentes, podendo ser sistemas operacionais distintos, com servidores dedicados ou virtuais, permitindo que os serviços de cada camada sejam personalizados e otimizados sem afetar uns aos outros.

3.2 Desenho do Dicionário de Dados Gráfico

Os microdados do Censo da Educação Superior reúnem um conjunto de informações detalhadas sobre o censo, que são disponibilizados de forma anual no portal do INEP e separados em cinco módulos distintos: Aluno, Curso, IES, Local de Oferta e Docente. Cada módulo constitui o conjunto de dados que foram coletados em cada questionário relacionado à sua entidade equivalente e métricas armazenadas em variáveis derivadas previamente processadas a partir da coleta inicial dos dados.

Como primeira etapa para o desenho do dicionário de dados gráfico, foi realizada a análise campo a campo de cada módulo, a fim de gerar correlação entre eles. O documento *Dicionário de Variáveis.xls* que é fornecido pelo órgão nos microdados de cada ano, é o

descritivo que detalha o significado de cada variável e seus possíveis valores de entrada. O mapa conceitual elaborado na Figura 3.2, foi gerado a fim de apresentar uma visão horizontal do problema e trazer clareza quanto à complexidade dos dados contidos nos microdados.

Após realizada a análise de cada módulo e suas respectivas ligações, constatou-se que de 2015 a 2019 havia em média por ano, cerca de 105 atributos relacionados à entidade Aluno, 112 atributos à entidade Curso, 48 atributos à entidade IES, 50 à entidade Local de Oferta e 41 à entidade Docente. Como pode ser notado, a complexidade para se estruturar um trabalho dessa magnitude levaria um tempo consideravelmente grande com uma massiva possibilidade de cruzamentos de informações, o que acarretaria uma profundidade maior da pesquisa fugindo do escopo de um trabalho de monografia. Sendo assim, entendendo as métricas essenciais desejadas para serem extraídas da criação desse protótipo, identificou-se a entidade Aluno ser possuidora do maior número de atributos de interesse a serem consultados e extraídas novas informações. Com isso, as entidades Local de Oferta e Docente foram desprezadas para esse estudo pois não possuíam ligação direta com o objeto central de pesquisa. Desta forma, foi possível realizar a elaboração de um dicionário de dados gráfico capaz de exemplificar as possíveis correlações dentre os dados dessas três entidades e seus respectivos cruzamentos como pode ser observado na Figura 3.3.

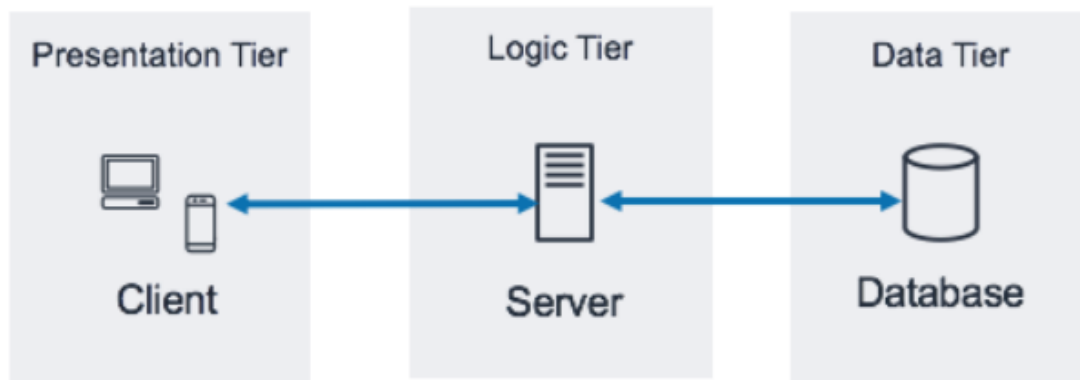


Figura 3.1: Padrão arquitetural para aplicação *three-tier*. Fonte: Three-tier architecture overview[1].

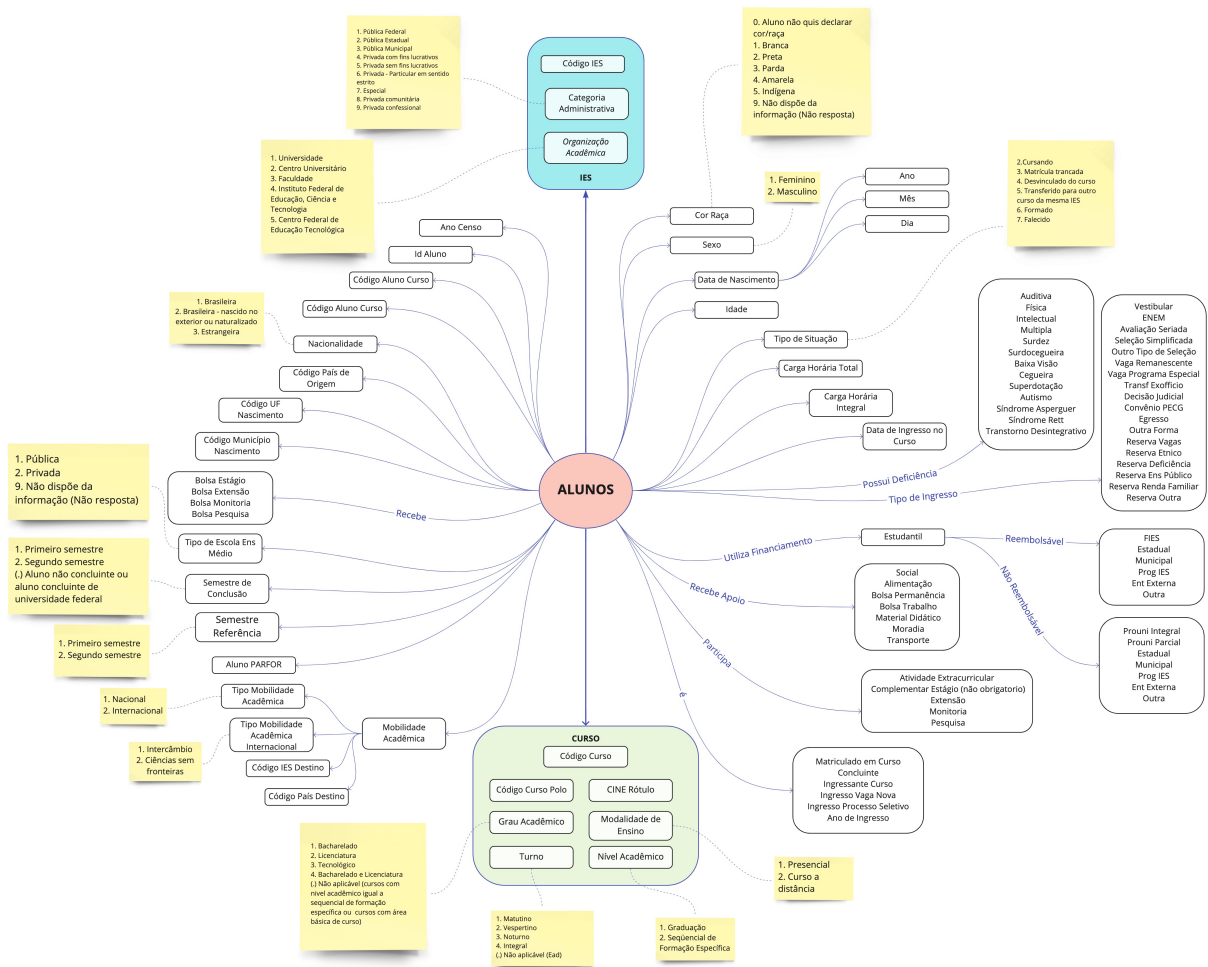


Figura 3.3: Dicionário de dados gráfico gerada a partir dos módulos Aluno, Cursos, IES. Fonte: própria (2021).

Capítulo 4

Implementação do Protótipo

Este capítulo apresenta as etapas de implementação do projeto desde a modelagem do banco de dados até modelo de arquitetura Cliente/Servidor utilizado na codificação do protótipo e todas tecnologias empregadas até a etapa de publicação da solução em um servidor web, possibilitando o acesso e utilização da plataforma. Este capítulo está organizado da seguinte forma: a Sessão 4.1 aborda as tecnologias empregadas no desenvolvimento do protótipo, a sessão 4.2 descreve as etapas de processamento dos dados e seu armazenamento, a Sessão 4.3 descreve a arquitetura da API *backend*, Sessão 4.4 descreve a arquitetura do Cliente ou *frontend* da aplicação e a Sessão 4.6 descreve o funcionamento do protótipo exibindo seu fluxo de funcionamento.

4.1 Tecnologias Empregadas

Segundo Spinellis [21] os critérios para escolha de tecnologias de um projeto são: produtividade do programador, facilidade de manutenção, eficiência, portabilidade, suporte a ferramentas e interfaces de software e hardware. Levando estes princípios como fundamentos, as tecnologias empregadas na construção do artefato, foram selecionadas a fim de possibilitar o acesso dessa aplicação por meio de uma interface web. A Figura 4.1 exhibe as camadas de arquitetura com as respectivas tecnologias escolhidas para seu desenvolvimento. A seguir, cada tecnologia utilizada é descrita para um melhor entendimento de sua aplicabilidade.

4.1.1 JavaScript

Segundo o RackerRank DeveloperSkills Report [2] publicado em 2020, o JavaScript é a linguagem de programação mais conhecida no mundo hoje, o que leva a ter uma grande comunidade que fortalece sua documentação, e criação de novas bibliotecas. É uma lin-

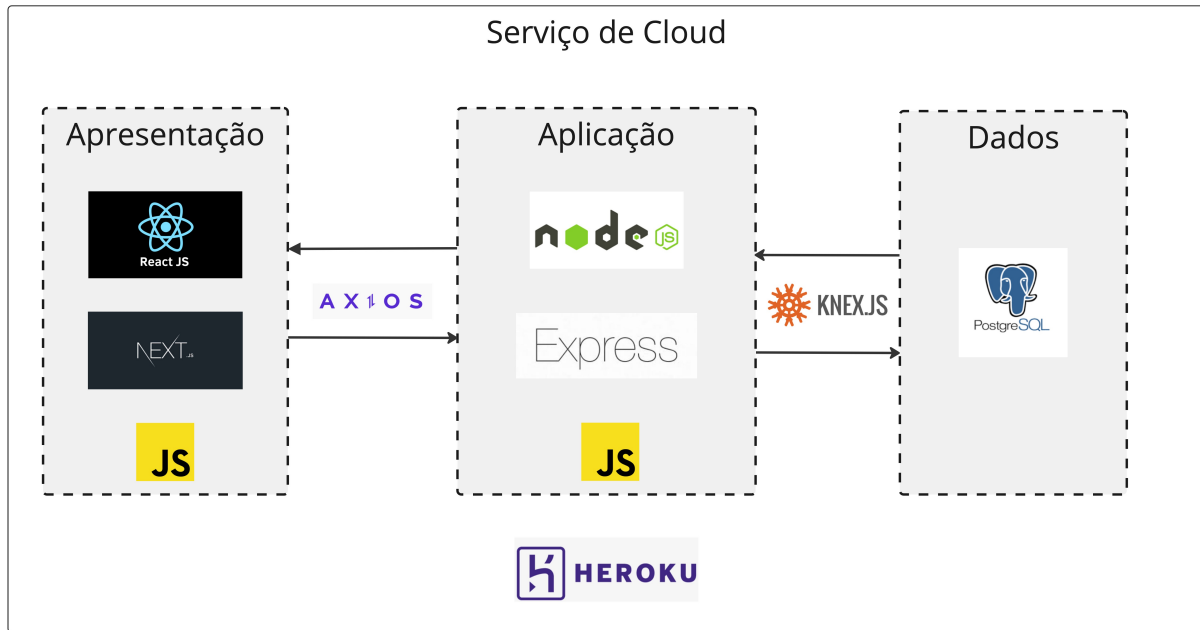


Figura 4.1: Arquitetura three-tier e tecnologias utilizadas no protótipo. Fonte: própria (2023).

guagem dinâmica baseada em protótipo, multiparadigma, de *thread* único, que suporta estilos orientados a objetos, imperativos e declarativos como programação funcional [22]. Devido ao crescimento da linguagem ao longo dos anos o que antes era apenas para o desenvolvimento *frontend*, hoje é utilizado para o desenvolvimento completo de diversos tipos de softwares, este sendo um dos principais motivos para adotá-la na implementação desse projeto.

4.1.2 React.js

React é uma biblioteca JavaScript declarativa, eficiente e flexível utilizada para a criação de interfaces de usuário. Ela permite compor interfaces de usuários complexas a partir de pequenos trechos de código isolados chamados componentes. Essa forma de trabalhar permite aos desenvolvedores reutilizar componentes de uma interface em outras, evitando duplicações e otimizando a reutilização de código. A atualização da página é realizada através de uma Virtual DOM que controla o estado de cada componente, não sendo necessário o recarregamento de toda página quando uma alteração é observada na árvore de elementos, tornando a experiência do usuário mais fluida. Para isso, essa biblioteca utiliza o conceito de *Single Page Applications* - SPA, onde uma aplicação carrega apenas um documento web, em seguida, atualiza o conteúdo do corpo desse documento por meio

	2020	2019	2018
JavaScript	1	1	2
Java	2	2	1
C	3	3	3
Python	4	4	5
C++	5	5	4
C#	▲ 6	7	6
PHP	▼ 7	6	7
TypeScript	8	8	8
Pascal	9	9	9
R	10	10	10

Figura 4.2: Linguagens de programação mais conhecidas 2018 - 2020. Fonte: Racker-Rank [2].

de APIs JavaScript. É uma das bibliotecas JavaScript mais utilizadas atualmente (vide Figura 4.3) devido a facilidade de criação de UIs, além de ser *open source* e declarativa.

	2020	2019	2018
AngularJS	1	1	1
React	2	2	3
Spring	3	3	2
Django	▲ 4	6	6
ExpressJS	▼ 5	4	4
ASP	▼ 6	5	5
.NETCore	7	7	7
Vue.js	▲ 8	9	10
Ruby on Rails	▼ 9	8	8
JSF	10	10	9

Figura 4.3: Frameworks mais conhecidas 2018 - 2020. Fonte: RackerRank [2].

4.1.3 Next.js

Next.js é uma estrutura flexível do React que fornece blocos de construção para criar aplicações web de forma rápida. Durante a criação de aplicações modernas é importante

considerar os seguintes pontos [23]:

Interface do usuário: como os usuários consumir e interagir com a aplicação.

Roteamento: como os usuários navegarão entre diferentes partes da aplicação.

Busca de dados: onde os dados estão armazenados e como recuperá-los.

Renderização: quando e onde renderizar conteúdo estático ou dinâmico.

Integrações: quais serviços de terceiros utilizar (*CMS, autenticação, pagamentos etc.*) e como se conectar a eles.

Infraestrutura: onde publicar a aplicação, armazena e executa o código da aplicação (*Serverless, CDN, Edge, etc.*).

Performance: como otimizar a aplicação para os usuários finais.

Escalabilidade: como a aplicação se adapta ao passo que a equipe, os dados e o tráfego aumentam.

Experiência do desenvolvedor: a experiência da equipe na criação e manutenção da aplicação.

Para solucionar essas questões, o *framework* Next.js lida com as ferramentas e configurações necessárias para o React e fornece tanto a estrutura quanto os recursos e otimizações adicionais para a aplicação. Sendo assim, o React pode ser usado para criar a UI e adotar gradualmente os recursos do Next.js para resolver requisitos comuns da aplicação, como roteamento, busca de dados e integrações.

4.1.4 Axios

Axios.js é uma biblioteca baseada em *Promises HTTP* que permite consumir os serviços de uma API. Possibilita ainda fazer requisições assíncronas HTTP de diferentes formas como *Get, Post, Put e Delete*. Axios pode ser utilizada com qualquer *framework* JavaScript, e uma vez instalada, permite a aplicação de interagir com uma API (REST) performando operações CRUD de forma mais fácil [24].

4.1.5 Node.js

Node.js é um ambiente em tempo de execução multiplataforma que possui código aberto e permite que o JavaScript seja executado no lado do servidor, ou seja, autoriza que o JavaScript seja executado fora do browser. Possui vários módulos sendo amplamente

utilizado no desenvolvimento web. Teve seu lançamento realizado em 2009 sobre a licença MIT e grande parte do seu sucesso se dá por sua I/O assíncrona, arquitetura orientada a evento, leveza, velocidade, escalabilidade e o fato de utilizar da linguagem de programação web largamente difundida, o JavaScript. Se baseia na arquitetura single-threaded tendo sido criado sobre a V8 JavaScript engine. Com aplicabilidade em diversos meios, dentre eles podemos citar [25]:

- Criação de aplicações de chats e mensagens instantâneas.
- Criação de API's escaláveis.
- Aplicações web que funcionam em real-time.
- Aplicações CLI (Client Line Interface), entre outros.

4.1.6 Express.js

Algumas tarefas comuns no desenvolvimento web são suportadas diretamente pelo Node.js. O Express é um framework que oferece soluções para gerenciar requisições de diferentes verbos HTTP como *Get*, *Post*, *Put* e *Delete* em diferentes URLs, integra templates para mostrar as respostas (*responses*) de maneira dinâmica, define configurações comuns da aplicação como porta a ser usada para conexão e localização dos modelos usados para renderizar respostas. Também é possível adicionar novos processos de requisição por meio de *middleware* em qualquer ponto da fila de requisições [26]. Também conta com diferentes bibliotecas para realizar integrações com cookies, sessões, logins, cabeçalhos de segurança, etc. Dentre suas principais características, podemos citar:

- Possui um sistema de rotas completo.
- Possibilita o tratamento de exceções dentro da aplicação.
- Permite a integração de vários sistemas de templates que facilitam a criação de páginas web para suas aplicações.
- Gerencia diferentes requisições HTTP com seus mais diversos verbos.
- Propicia a criação rápida de aplicações utilizando um conjunto pequeno de arquivos e pastas.

4.1.7 Knex.js

Compor consultas SQL no navegador para execução no servidor é altamente desencorajado, pois isso pode ser a causa de sérias vulnerabilidades de segurança [27]. Para

isso, o Knex.js é uma biblioteca de acesso a banco de dados para Node.js que suporta os seguintes bancos: PostgreSQL, CockroachDB, MSSQL, MySQL, MariaDB, SQLite3, Better-SQLite3, Oracle e Amazon Redshift . Ele é definido como um *query builder*, ou seja, um construtor de consultas para o banco de dados que ele se conectar. É pensado para ser flexível e simples de usar, fornecendo um conjunto de funcionalidades básicas para realizar qualquer tipo de operação no banco de dados.

4.1.8 PostgreSQL

PostgreSQL é um poderoso sistema de gerenciamento de banco de dados objeto-relacional *open source*, que utiliza e estende a linguagem SQL combinada com várias funcionalidades que de forma segura armazena e dimensiona as cargas de trabalho mais complexas. É um banco de dados multiplataforma, estando presente na maioria de variantes do Linux e Windows, além de serviços de nuvem como Amazon, Google e Microsoft. Tem suporte à diversas linguagens de programação e a dados geométricos, além de JSON e outras características NOSQL Possui ainda, uma grande comunidade que auxilia no desenvolvimento de novas *features*.

4.1.9 Heroku

De acordo com a própria documentação, Heroku é uma plataforma em nuvem que auxilia companhias a realizar a hospedagem, configuração, testagem e publicação de projetos diretamente na nuvem. É a forma mais rápida de ir de uma ideia à uma URL, contornando todas da complexidade gerada pela infraestrutura. De forma resumida, é uma plataforma capaz de hospedar aplicações, criar bancos de dados em nuvem, e fazer de forma automatizada as atualizações na aplicação logo que estas são empurradas pelo desenvolvedor para o repositório onde o código está armazenado [28].

4.2 Manipulação dos Dados

Após o entendimento de como as informações são organizadas dentro dos microdados disponibilizados pelo INEP e os requisitos básicos já bem definidos, foi possível realizar a elaboração do modelo lógico do banco de dados (vide Anexo I) que detalha as tabelas e seus relacionamentos, campos e seus tipos e tamanhos associados. Com o modelo lógico pronto que pode ser visto na Figura 4.4, foram escritos então os *scripts SQL* para criação do modelo físico do banco de dados. Todos estes *scripts* estão documentados no Anexo II.

O grande volume de dados coletados em todo território nacional de forma detalhada já explicitado anteriormente sobre cada entidade, tornou o tamanho final de cada uma

Tabela 4.1: Dimensões de tabelas eletrônicas dos microdados de 2019.

Nome do arquivo	Tamanho
SUP_ALUNO_2019.CSV	3.09 GB
SUP_CURSO_2019.CSV	9.8 MB
SUP_DOCENTE_2019.CSV	49.5 MB
SUP_IES_2019.CSV	712 KB
SUP_LOCAL_OFERTA_2019.CSV	161.9 MB

Tabela 4.1 lista os arquivos “.csv” dos microdados e seus tamanhos para o ano censo de 2019.

Tabelas com dimensões de volumes muito altos como estas, também são inviáveis de serem abertas para realizar sua manipulação em computadores de uso pessoal como os utilizados pelos orientadores e orientando deste trabalho. Sendo assim, a estratégia adotada para trabalhar com esses arquivos foi realizar desfragmentação desses arquivos originais em arquivos menores de no máximo 10mb para somente então serem carregados na base de dados física. Para esse objetivo foi utilizado os seguintes *scripts* em um interpretador *Shell*.

```
# Como dividir arquivos CSV

##Por quantidade de linhas
'''bash
split -l 10 path_arquivo.csv
'''

##Por tamanho em Mb
'''bash
split -b 10m path_arquivo.csv
'''

##Renomear todos arquivos para ###'.csv'
'''bash
for i in *; do mv "$i" "$i.csv"; done
'''
```

Após a divisão das tabelas em documentos menores, foi então possível realizar a carga no banco de dados PostgreSQL, verificando de forma mais eficaz, erros que poderiam ocorrer durante o carregamento da massa de dados nas tabelas, e adequação por exemplo, do tamanho de campos específicos que necessitaram atualizar seu tamanho de armazenamento.

A tabela “*SUP_CURSO_2019.CSV*” demonstra bem o número expressivo de informações contidas nesses arquivos. Somente ela, conta com um número de 40.709 registros referentes ao número de Cursos vinculados a Instituições de Ensino Superior cadastradas junto ao MEC no ano de 2019. Já a tabela “*SUP_ALUNO_2019.CSV*”, a mais volumosa em número de registros por comportar todos os alunos vinculados a uma IES no ano de 2019, ultrapassa a unidade de dezenas de milhões. Tendo como exemplo apenas os registros referente a um dos anos das pesquisas, e sendo um dos objetivos desse trabalho realizar a exibição de uma linha histórica entre os anos da consulta escolhida pelo usuário, tornou-se inviável, neste momento, o armazenamento de toda massa de dados em um servidor dado seu alto custo operacional e tempo de latência das consultas SQL realizando a varredura em toda essa massa de dados e só então chegar ao objetivo final que é a exibição dos infográficos. Como medida para minimizar esses efeitos colaterais, foi realizada uma adequação a nível de protótipo, tomando como medida a redução do escopo dos dados consultados no projeto. De dados de abrangência nacional passou-se a ser trabalhado com informações referente somente ao Distrito Federal. Para isso, foram realizadas filtragens em todas as tabelas das entidades carregadas a fim de armazenar registros somente de Alunos vinculados a Instituições de Ensino Superior sediadas no Distrito Federal, e Cursos vinculados à essas mesmas IESs. Para tal tarefa, foram criadas novas tabelas para armazenamento desses dados previamente filtrados e executados com *scripts* detalhados também no Anexo II.

4.3 Camada de Aplicação - Backend

A Camada de Aplicação (*Controller*), foi escrita usando Node.js e Express.js. Essa camada representa o Servidor da Aplicação que atua como a ponte de comunicação entre a Camada de Apresentação e a Camada de Dados. Desta forma, ela serve as páginas HTML no dispositivo do usuário e aceita solicitações REST vindas da Camada de Apresentação e devolvendo as respostas apropriadas [26]. Ela orquestra toda dinâmica de quais filtros podem ser aplicados em cada etapa de consulta, realiza o cruzamento das informações e retorna os valores finais que são utilizados para criação de gráficos na última etapa da funcionalidade da aplicação. A biblioteca Knex.js foi utilizada para fazer a comunicação com a Camada de Dados.

A estrutura do projeto de *backend* se desenha da seguinte forma:

```

BACK-PL
├── src
│   ├── controllers          - armazena APIs (GET, POST, PUT, DELETE)
│   └── CollegeController.js - realiza todo controle de dados referentes

```

```

    aos cursos
  |— IesController.js      - realiza todo controle de dados referentes
                           às instituições de ensino
  |— ResearchController.js - realiza todo controle de dados referentes
                           cruzamento de informações da pesquisa
  |— StudentController.js  - realiza todo controle de dados referentes
                           aos alunos
  |— database
    |— connection.js      - realiza configuração com o SQL query builder
  |— utils
    |— studentSelectorFilters.json - armazena os filtros de multipla
                                   escolha referentes aos alunos
    |— studentStaticFilters.json  - armazena os filtros de binários
                                   referentes aos alunos
  |— app.js               - configuração base da aplicação com importação das principais
                           bibliotecas
    |— routes.js          - realiza o gerenciamento de rotas
    |— server.js          - realiza configuração de porta e starta aplicação
  |— knexfile.js         - exporta configurações de conexão com banco de dados
  |— package.json        - configura todos pacotes necessários na aplicação
  |— procfile            - arquivo de configuração do heroku

```

O código fonte do *backend* pode ser encontrado em:

<https://github.com/fcpmagalhaes/back-pl>

4.4 Camada de Apresentação - Frontend

A Camada de Apresentação (*View*), foi escrita utilizando-se da biblioteca JavaScript React.js, tendo Next.js como framework. Essa camada representa o Cliente, ou seja, ela consome e exibe todas informações advindas da camada de Aplicação para o usuário final. Nela é possível realizar toda interação com as páginas web e verificar todas funcionalidades do projeto. Como uma aplicação que roda no *browser*, ela é a responsável por gerar todos componentes gráficos, e o *design* da aplicação, usabilidades e funcionalidades que auxiliam o usuário a chegar no resultado final esperado.

A estrutura do projeto de *frontend* se desenha da seguinte forma:

```

FRONT-PL
  |— src

```

- └─ components - realiza estruturação de cada componente reutilizável
 - └─ charts
 - └─ InputValue
 - └─ Layout
 - └─ Modal
 - └─ MultipleSelect
 - └─ Ontology
 - └─ ResearchDescription
- └─ pages - realiza estruturação física de cada página da aplicação
 - └─ em-construcao
 - └─ infografico
 - └─ dicionario
 - └─ _app.js - estrutura de mãe páginas, menus, comunicação com store e estilo da aplicação
 - └─ _document.js - estruturação base da página html que renderizará o restante da aplicação
 - └─ index.js - página home da aplicação
- └─ services
 - └─ axios
 - └─ index.js - declara configuração inicial do Axios
- └─ store
 - └─ infographic
 - └─ actions.js - realiza o controle das chamadas de cada alteração no state
 - └─ reducer.js - cria o estado inicial do state e gerencia suas atualizações
 - └─ sagas.js - realiza comunicação com a API
 - └─ index.js - realiza configuração middleware e comunicação com reducer
 - └─ sagas.js - realiza configuração do redux sagas
- └─ theme
 - └─ index.js - Armazena guia de estilos geral da aplicação
- └─ package.json - configura todos os pacotes necessários na aplicação

O código fonte do *frontend* pode ser encontrado em:

<https://github.com/fcpmagalhaes/front-pl>

4.5 Implantação da Aplicação

Para realizar a disponibilização da aplicação na web, todo código fonte foi hospedado no Github que é um serviço de gerenciamento de projetos em nuvem. Com o projeto finalizado, foi realizada a integração com o serviço de hospedagem gratuita Heroku. Essa camada de nuvem, abstrai boa parte das configurações que necessitam ser realizadas de forma manual em um servidor. Com processos automatizados, rapidamente é possível realizar a implantação da aplicação num servidor Web e disponibilizar uma URL para acesso. Também dentro desta plataforma, foi criada a instância do banco de dados PostgreSQL do ambiente de produção. Tendo toda aplicação integrada dentro da mesma plataforma, é mais fácil realizar o controle de integração contínua, que garante que o portal é atualizado de forma automática a cada vez que uma atualização é empurrada para o repositório principal.

A aplicação na Web pode ser acessada em:

<https://pl2-frontend.herokuapp.com/>

4.6 Apresentação do Protótipo

Toda aplicação foi desenvolvida para ser simples e intuitiva, direcionando o usuário a executar um fluxo de operações até chegar no resultado esperado que é a geração do infográfico. Para isso, foi realizado o desenvolvimento de um fluxo de seis etapas, dentre as quais existem estágios de seleções obrigatórias e outras opcionais. Ao passo que o usuário clica em "avançar" para próxima fase, as informações da etapa anterior são utilizadas para filtrar dinamicamente informações do estágio seguinte além de serem armazenadas para gerar a consulta final. Em seguida é dado detalhamento de cada página da aplicação.

4.6.1 Página Principal

A página principal possui uma breve descrição sobre a funcionalidade da aplicação junto a um botão "Iniciar" para onde o usuário é direcionado para o primeiro passo do fluxo de pesquisa. Há ainda um menu lateral esquerdo com opções de navegação para outras páginas informativas sobre a pesquisa, Figura 4.5.

4.6.2 Seleção de Anos de Pesquisa

Na página de ano de pesquisa, o usuário necessita realizar a seleção de ao menos um ano para que no final seja criado o infográfico trazendo uma linha histórica sobre o cruzamento



Figura 4.5: Página principal da aplicação. Fonte: própria (2023).

de dados realizados durante anos. Nessa etapa os anos exibidos correspondem às cargas de dados dos microdados que foram realizados dentro do banco de dados, Figura 4.6.

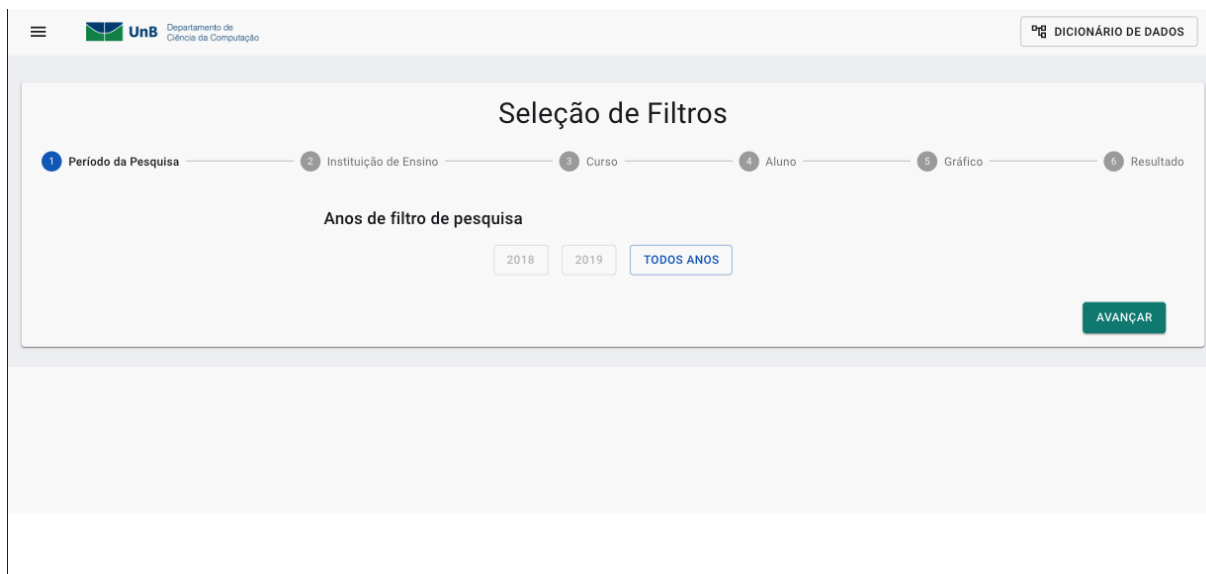


Figura 4.6: Seleção de anos a serem pesquisados. Fonte: própria (2023).

4.6.3 Seleção de Filtros de Instituição de Ensino

Em filtros de instituição de ensino, o usuário poderá fazer dois tipos de seleção. Caso opte por pesquisar pelos nomes das instituições, os campos organização acadêmica e categoria

administrativa se desativam, pois a pesquisa ocorrerá pela instituição selecionada. Caso opte por organização acadêmica e/ou categoria administrativa, o contrário também vale, desativando a opção de consulta por nome da instituição. Caso nenhuma opção seja selecionada, nenhum filtro sobre instituições de ensino será aplicado, Figura 4.7.

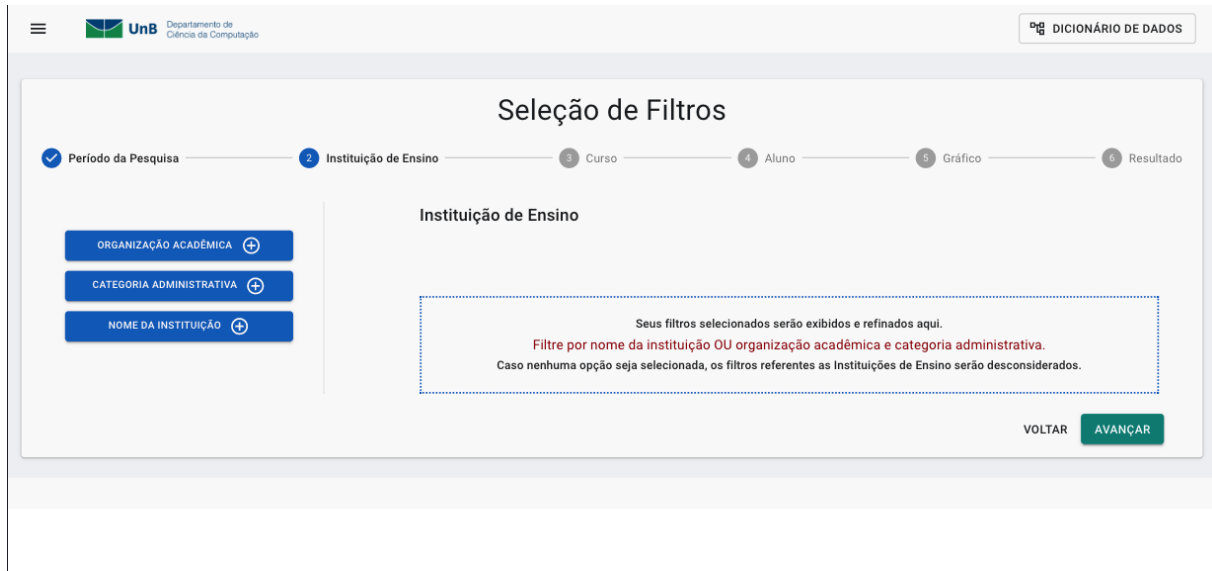


Figura 4.7: Seleção de filtros de instituição de ensino. Fonte: própria (2023).

4.6.4 Seleção de Filtros de Curso

Durante a seleção de filtros de cursos, todas opções são selecionáveis, sendo inclusive possível realizar pesquisa por mais de um item do mesmo campo, por exemplo, "Turno" pode ser pesquisado por apenas um período "Matutino" ou quantos períodos desejar "Matutino" "Vespertino", "Noturno" etc. A mesma regra se aplica aos demais campos, se fazendo valer também de que caso nenhuma opção seja selecionada, nenhum filtro sobre cursos será aplicado, Figura 4.8.

4.6.5 Seleção de Filtros de Aluno

Na seleção de aluno, valem todas as premissas apresentadas na sessão Filtros de Curso, explicitada anteriormente. Esse passo configura a última etapa de seleção de filtros, Figura 4.9.

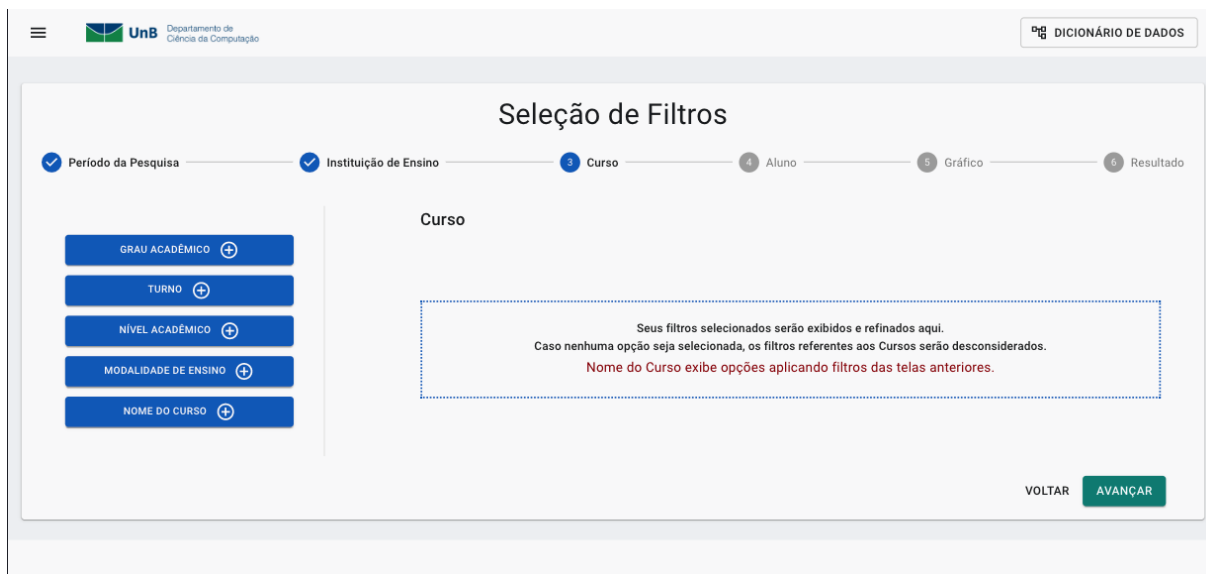


Figura 4.8: Seleção de filtros de curso. Fonte: própria (2023).

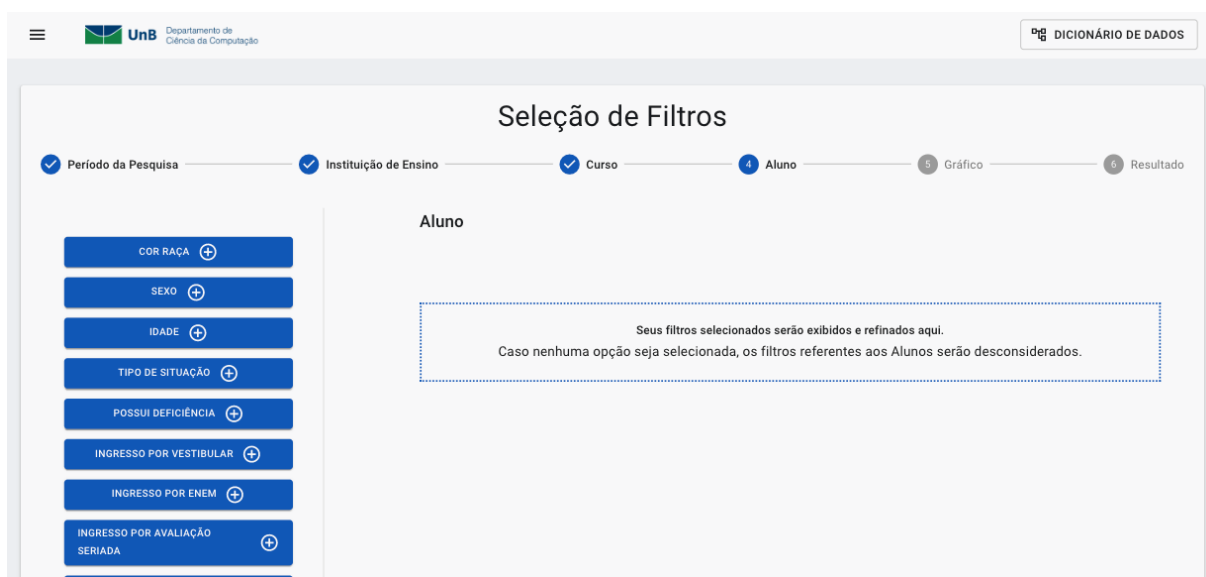


Figura 4.9: Seleção de filtros de alunos. Fonte: própria (2023).

4.6.6 Seleção de Modelo Gráfico

Na última etapa antes da exibição do infográfico com o resultado dos filtros aplicados anteriormente, o usuário necessita escolher de forma obrigatória qual tipo de gráfico deseja exibir. Conta com quatro opções, sendo elas, gráfico de área, gráfico de barra, gráfico de linha e gráfico de pizza, Figura 4.10.

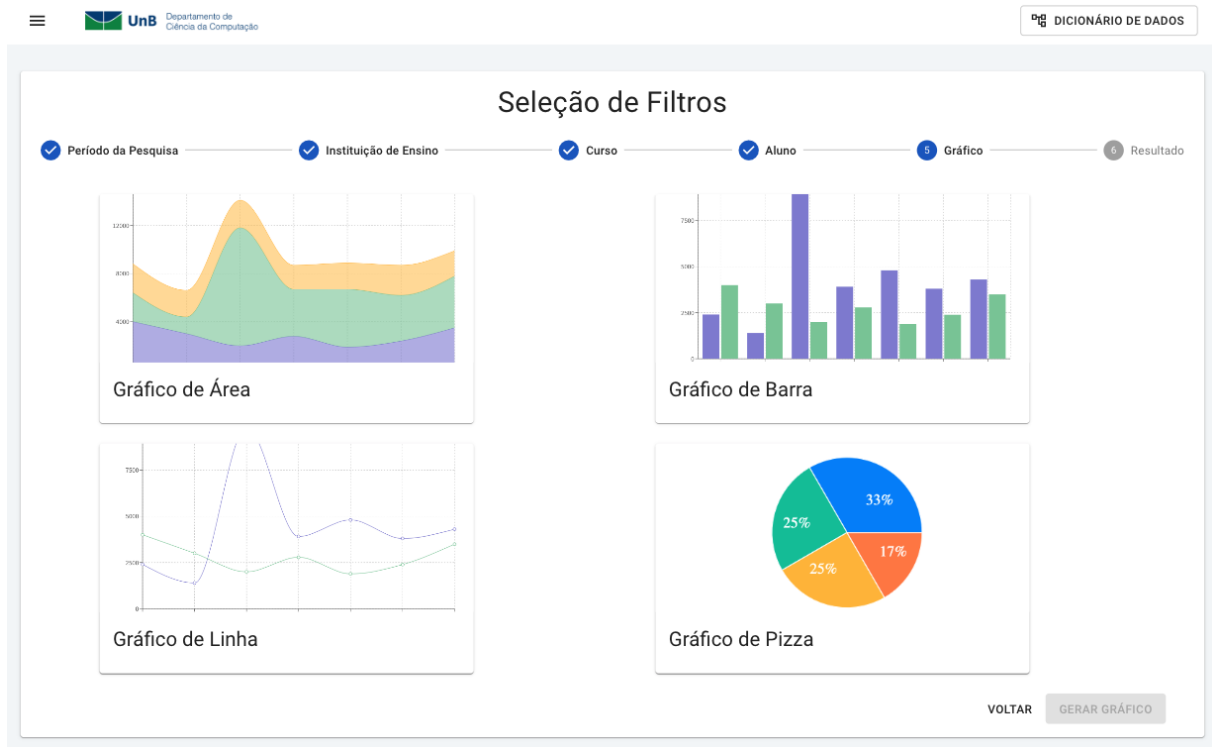


Figura 4.10: Seleção de tipo de gráfico. Fonte: própria (2023).

4.6.7 Resultado Final

Na etapa de exibição de infográfico, é apresentado ao usuário, um gráfico com os dados obtidos na pesquisa de filtros selecionados anteriormente, bem como o descritivo de que filtros foram aplicados nesta seleção. Aqui é o fim do fluxo e o objetivo final do protótipo foi concluído, Figura 4.11.

4.6.8 Página do Dicionário de Dados Gráfico

No menu superior consta um botão com título "Dicionário de Dados" que abre sobre qualquer outra página, uma modal com o dicionário de dados gráfico, permitindo ao usuário consultar o mapa de atributos de cada entidade e seus respectivos relacionamentos. O botão lateral direito com mesmo título direciona a uma página com mesma funcionalidade, Figura 4.12.



Figura 4.11: Apresentação do gráfico com os resultados. Fonte: própria (2023).

Dicionário de Dados do Projeto

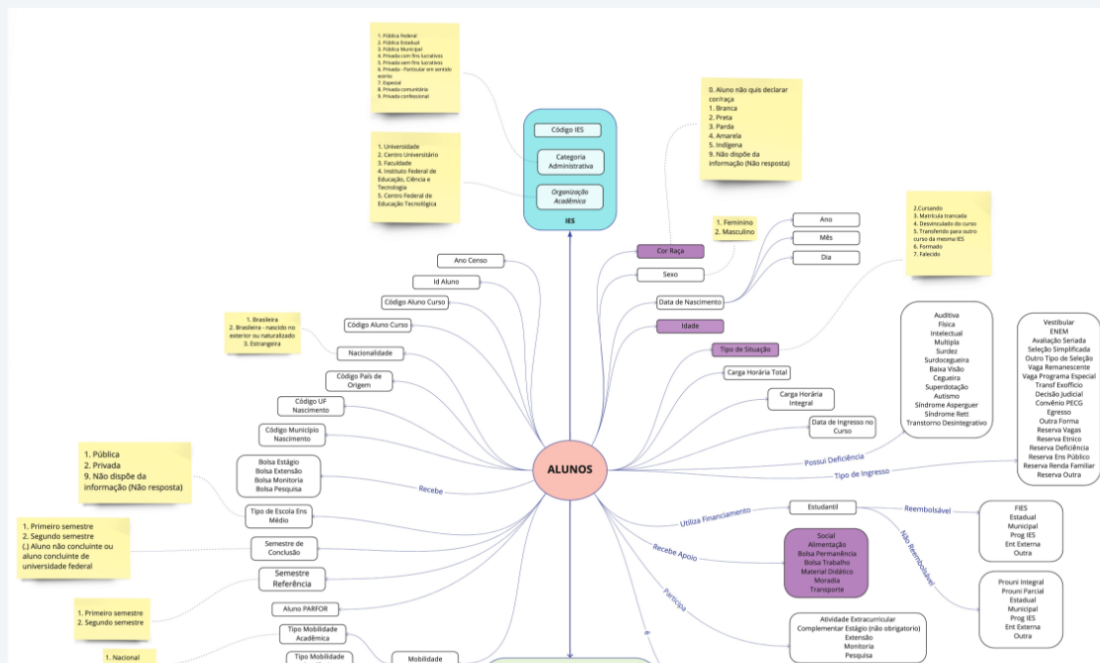


Figura 4.12: Apresentação do esquema do dicionário de dados gráfico com todos os campos possíveis de serem utilizados na consulta. Fonte: própria (2023).

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos a partir da coleta de dados de usuários sobre a usabilidade do protótipo e está organizado da seguinte forma: a Sessão 5.1 apresenta o que motivou a realização dos testes nos moldes como foram realizados, a Sessão 5.2 descreve o que é a escala utilizada na pesquisa e o questionário de avaliação aplicado, a Sessão 5.3 exibe as estatísticas coletadas sobre o formulário.

5.1 Estatísticas Censo da Educação Superior

Nos primeiros ensaios sobre este trabalho que se originou em 2019, os microdados do CES eram disponibilizados somente por planilhas eletrônicas em formato (".csv"), cabendo aos jornalistas manipulá-las para extrair informações ou restringir-se às informações divulgadas nas coletivas de imprensa anuais para noticiar as últimas estatísticas. Nos anos que seguiram, o Inep fez o lançamento de dois portais para divulgação dos dados, sendo o mais recente em dezembro de 2022. O Painel de BI do Censo da Educação Superior, apresenta as principais estatísticas e seus indicadores resultantes, objetivando facilitar a consulta dos usuários às essas informações. Com propósitos parecidos, mas aplicabilidade diferentes, foi elaborado então um modo para dimensionar a percepção de usuários ao fazerem uso do Painel de BI lançado pelo INEP e o portal protótipo desenvolvido neste trabalho.

5.2 System Usability Scale

O SUS é uma escala criada em 1986 por John Brooke que tem por objetivo avaliar efetividade (o usuário consegue completar o objetivo), eficiência (quanto esforço e recurso é necessário para isso) e satisfação (a experiência foi satisfatória). Para isso utiliza uma escala de 1 a 5 sendo 1, discordo totalmente, e 5, concordo totalmente, que posteriormente

é calculada para se chegar a um resultado final. As dez perguntas realizadas no teste foram:

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Para realizar o cálculo final da pontuação são utilizadas as seguintes regras: Para respostas ímpares (1, 3, 5), subtraia 1 da pontuação que o usuário respondeu. Para respostas pares (2 e 4), subtraia a resposta de 5. Some todos os valores das dez perguntas, e multiplique por 2,5. Essa é sua pontuação final pode ir de 0 a 100 [29]. De acordo com Sauro [30], a pontuação SUS média dos 500 estudos que realizou foi de 68 pontos.

5.3 Estatísticas de Usabilidade

Para o teste de usabilidade, foi aplicado o questionário descrito anteriormente, solicitando ao usuário realizar uma consulta em cada aplicação, a fim de testar a facilidade de navegação dentro de cada plataforma. Quarenta por cento dos avaliadores foram jornalistas e sessenta por cento foram usuários comuns que poderiam se interessar sobre o tema.

Os seguintes cenários foram descritos:

Caso de Uso 1 - Desejo realizar uma pesquisa contabilizando quantos alunos pretos pardos e indígenas do sexo masculino se formaram no curso de Computação Licenciatura na Universidade de Brasília entre os anos de 2018 e 2019, disponibilizando essas informações por meio de um gráfico de barra.

Caso de Uso 2 - Desejo realizar uma pesquisa por alunos brancos e pretos/pardos que se formaram no curso de Computação Licenciatura na Universidade de Brasília entre os

anos de 2018 e 2019, disponibilizando essas informações por meio de um gráfico de pizza com informação em porcentagem.

O primeiro caso de uso foi aplicado ao protótipo desenvolvido neste projeto que pode ser acessado em (Aplicação Protótipo), enquanto o segundo caso de uso foi executado dentro do Painel de BI do INEP que pode ser acessado em (Novo Painel de Estatísticas do Censo da Educação Superior). O resultado da avaliação dos usuários nas duas plataformas

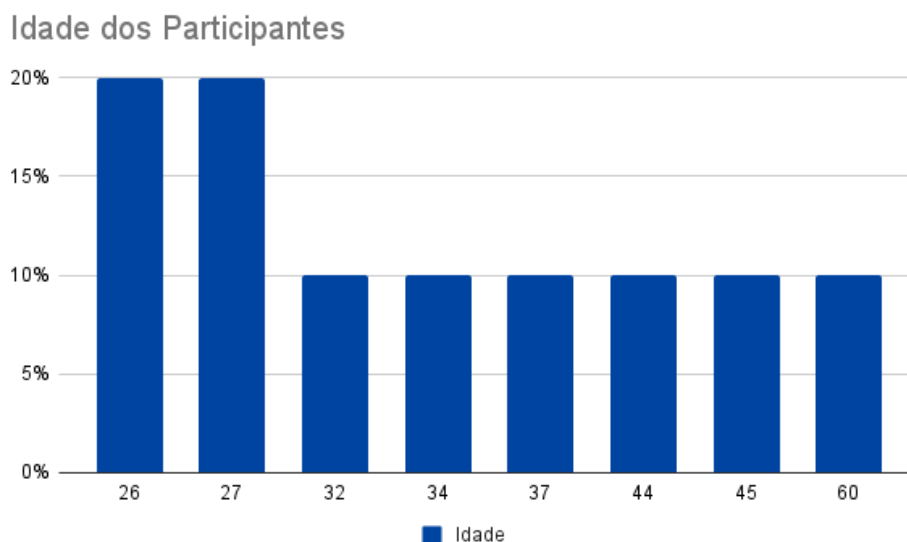


Figura 5.1: Participantes. Fonte: própria (2023).

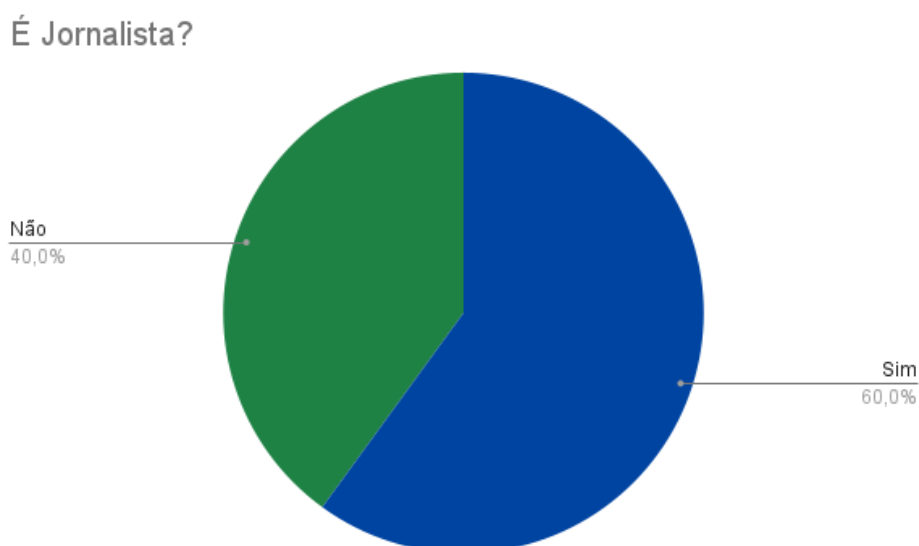


Figura 5.2: Porcentagem de jornalistas e não jornalistas Fonte: própria (2023).

é informado nas figuras seguintes.

Como resultado do cálculo realizado pela escala *System Usability Scale* podemos observar que a média adquirida pela aplicação do protótipo foi de 84 e a média adquirida pelo Painel BI foi de 45. Esses valores não são expressos em porcentagem, e sim, em uma escala de 0 a 100 onde a média de uma classificação positiva é 68 pontos como pode ser vista na Figura 5.14.

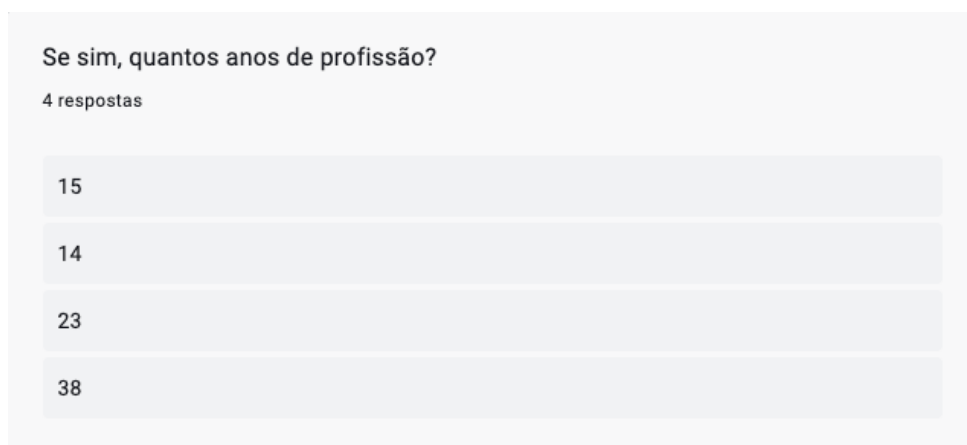


Figura 5.3: Anos de profissão caso seja jornalista. Fonte: própria (2023).

1. Eu acho que gostaria de usar esse sistema com frequência.

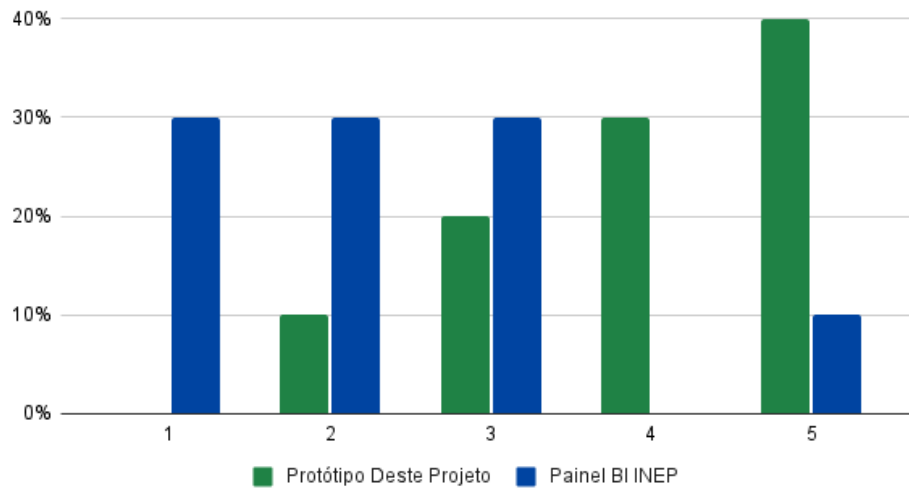


Figura 5.4: Resultados pergunta 1. Fonte: própria (2023).

2. Eu acho o sistema desnecessariamente complexo.

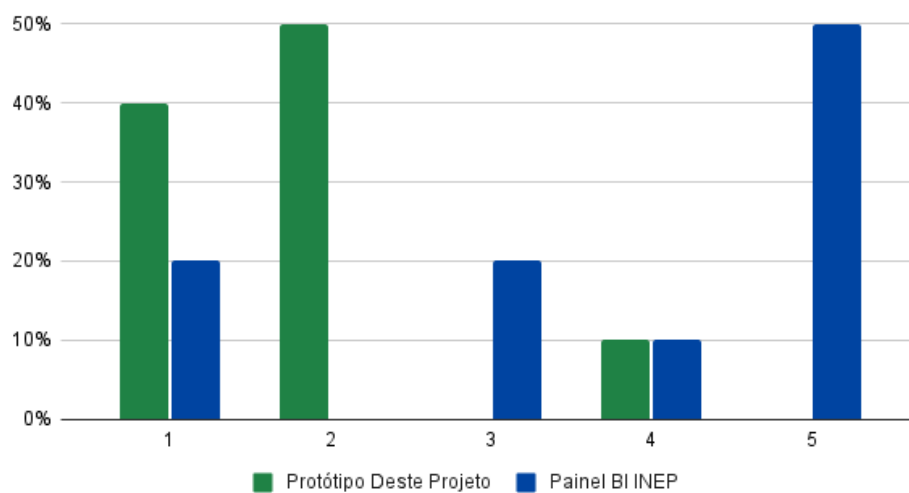


Figura 5.5: Resultados pergunta 2. Fonte: própria (2023).

3. Eu achei o sistema fácil de usar.

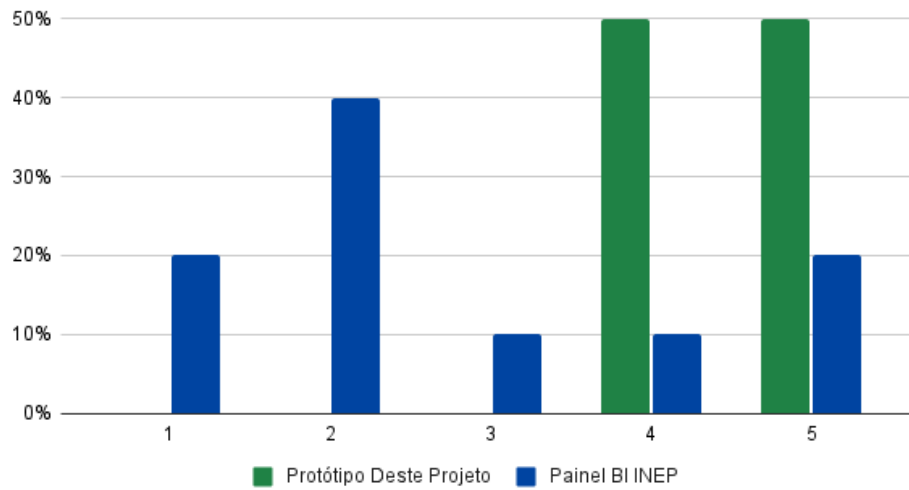


Figura 5.6: Resultados pergunta 3. Fonte: própria (2023).

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

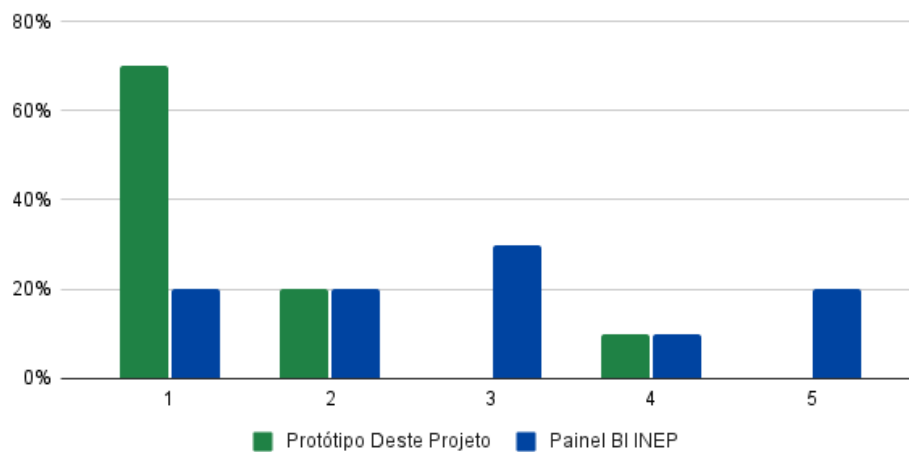


Figura 5.7: Resultados pergunta 4. Fonte: própria (2023).

5. Eu acho que as várias funções do sistema estão muito bem integradas.

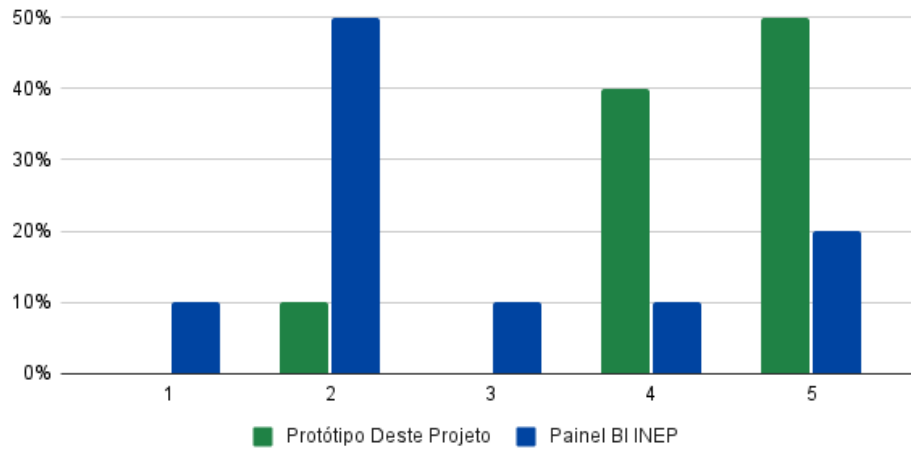


Figura 5.8: Resultados pergunta 5. Fonte: própria (2023).

6. Eu acho que o sistema apresenta muita inconsistência.

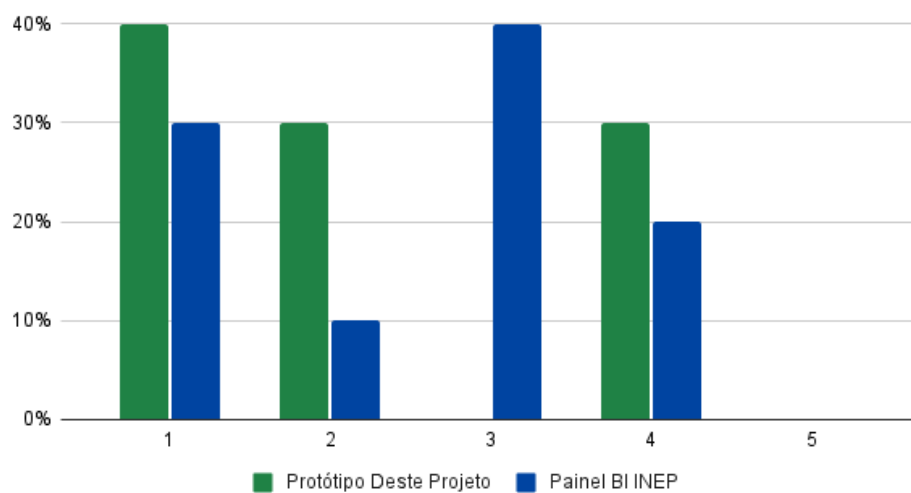


Figura 5.9: Resultados pergunta 6. Fonte: própria (2023).

7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

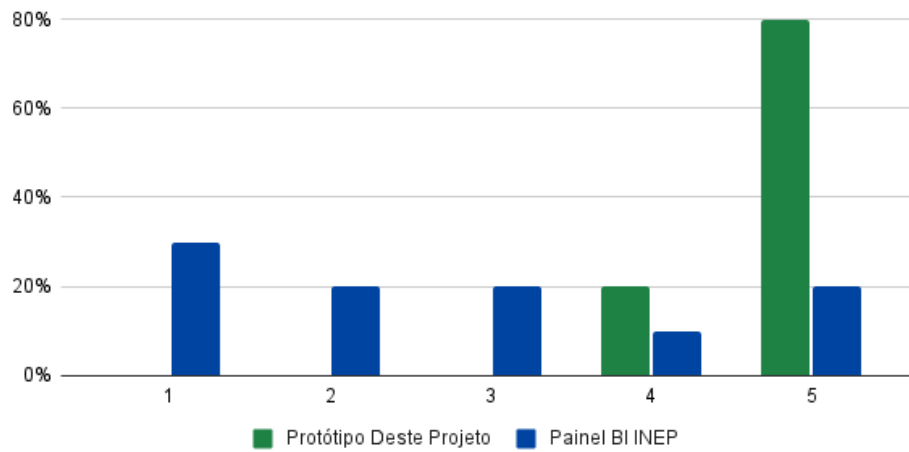


Figura 5.10: Resultados pergunta 7. Fonte: própria (2023).

8. Eu achei o sistema atrapalhado de usar.

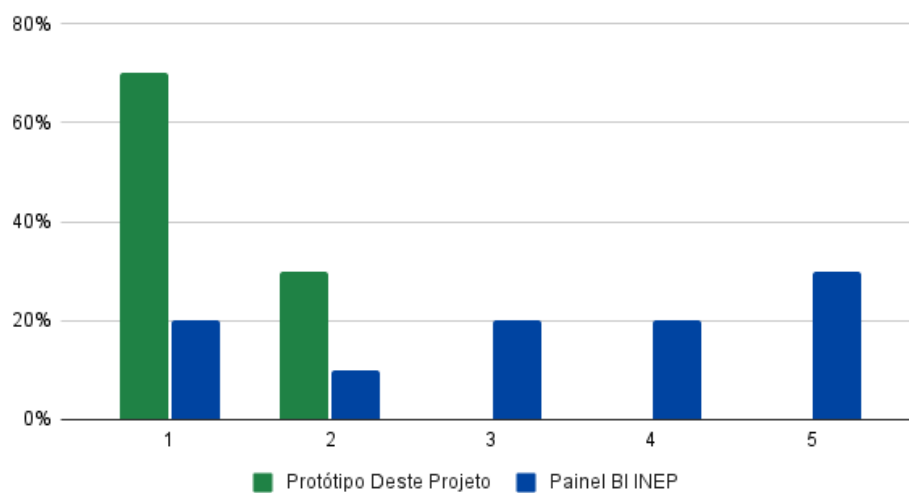


Figura 5.11: Resultados pergunta 8. Fonte: própria (2023).

9. Eu me senti confiante ao usar o sistema.

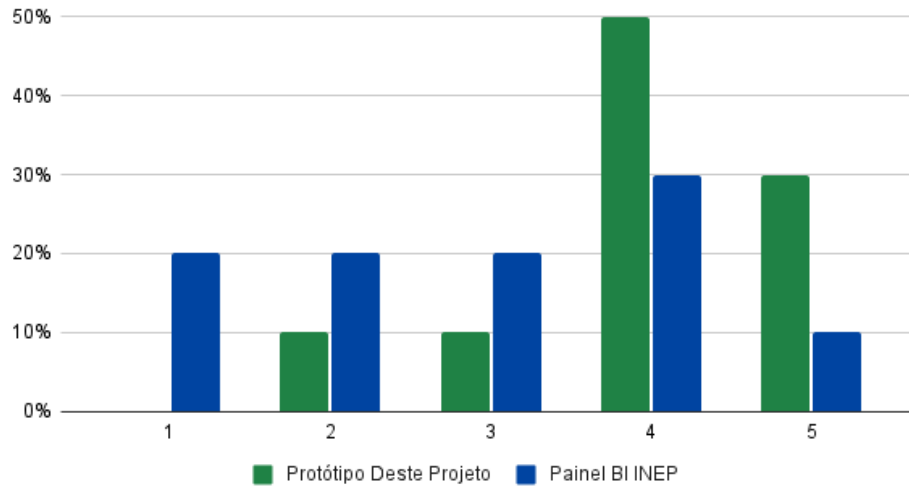


Figura 5.12: Resultados pergunta 9. Fonte: própria (2023).

10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

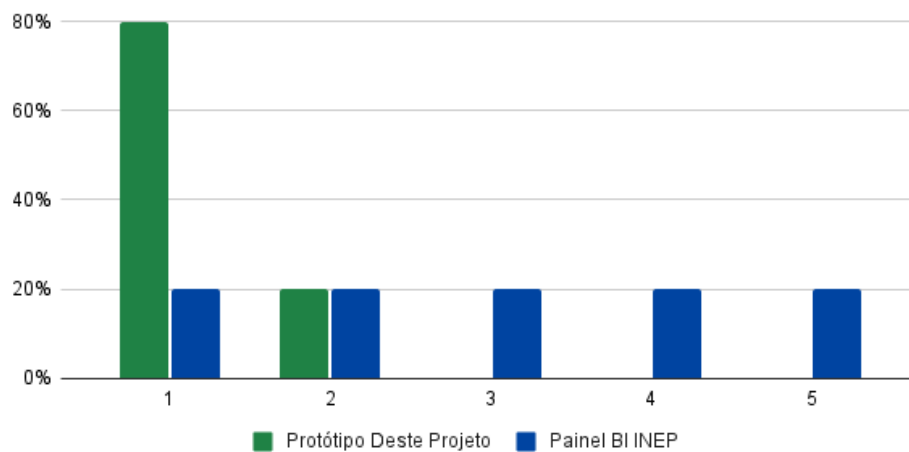


Figura 5.13: Resultados pergunta 10. Fonte: própria (2023).

Resultado System Usability Scale

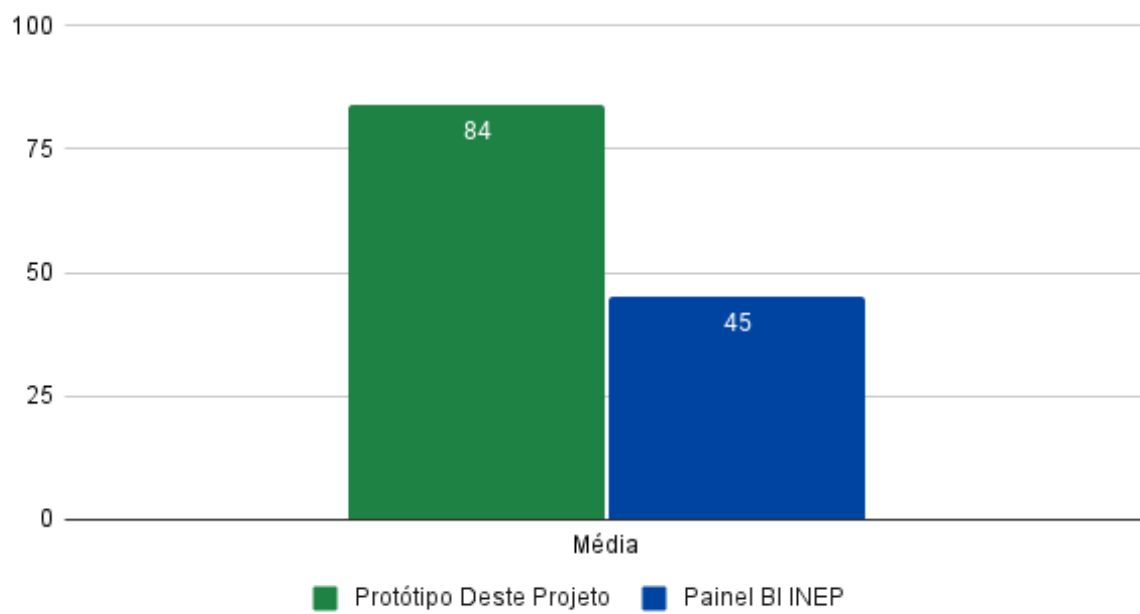


Figura 5.14: Resultados Final Teste Usabilidade. Fonte: própria (2023).

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

A grande quantidade de dados produzidos hoje, demanda cada vez mais seu processamento de maneira automatizada, afim de gerar valor por meio dessas informações. Técnicas como *Data Warehouses*, *Data Mining* e *Business Intelligence*, têm sido aplicadas para responder perguntas sobre dados, porém não se fazem por meio interativo, tornando suas respostas estáticas às perguntas definidas previamente, o que demanda uma alta complexidade e custo operacional para extrair informações específicas de forma dinâmica. Os dados do Censo da Educação Superior fornecidos pelo INEP, não diferente de muitos dados públicos disponibilizados na web, carecem de uma estrutura gráfica capaz de facilitar a consulta dos mesmos, gerando informação útil a jornalistas e à população em geral, que não dispõe de conhecimento aprofundado em computação. O objetivo deste trabalho foi criar um protótipo de aplicação web facilmente entendível para o usuário, possuindo um fluxo intuitivo que proporcionasse uma navegação fluida e autodidata. Os resultados obtidos por meio da aplicação do teste de usabilidade *System Usability Scale*, comprovaram que o modelo adotado para solução deste problema, a partir da construção de um projeto dedicado utilizando-se de arquitetura em três camadas, satisfaz o objetivo inicial deste trabalho.

6.2 Limitações

Durante a realização da pesquisa, alguns fatores foram limitantes para uma abrangência maior dos dados estudados. Em primeiro momento, a necessidade de máquinas robustas o suficiente para armazenar e processar a grande quantidade de dados, necessária para criar uma linha histórica com as informações de todos os anos de coleta do censo já registrados. Resultando em uma seleção mais restrita dos dados anuais, e posteriormente a redução

da abrangência nacional para tratar de informações somente do Distrito Federal. Ainda durante o andamento do projeto, houve adequações na disponibilização dos microdados por parte do INEP. Foram realizados estudos por parte do órgão e entidades associadas, que apontaram riscos de identificação de pessoas nos dados divulgados. Em atendimento à Lei de Acesso à Informação e à Lei Geral de Proteção de Dados Pessoais, os Microdados do Censo da Educação Superior passaram a partir do fim de 2022, a serem estruturados ao nível de Instituições de Ensino Superior e Cursos. Portanto, não são mais acessíveis de forma aberta informações referentes à entidade Alunos, que foi objeto central de estudo deste trabalho. Para tal, faz-se necessária solicitação mais detalhada junto ao órgão para liberação dos dados para fins de pesquisa.

6.3 Trabalhos Futuros

Como sugestão de trabalhos futuros, identifica-se a ampliação da base de dados para comportar informações de uma maior abrangência de anos, bem como implementações de melhorias na aplicação, como na tratativa de erros e emissão destas para o usuário, que segundo a avaliação heurística "erro", são essenciais para orientação do usuário em relação à execução de tarefas. A disponibilização da aplicação dentro de um servidor do Departamento de Ciência da Computação é ideal pois garante a continuidade de usabilidade do protótipo para futuros estudos e consultas. Quanto à readequação da disponibilização dos microdados por parte do INEP, é necessário fazer novos estudos de validação dos campos e suas respectivas entidades para readequação das consultas e armazenamento desses indicadores no banco de dados.

Referências

- [1] AWS, Amazon: *Three-tier architecture overview*. <https://docs.aws.amazon.com/pdfs/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/serverless-multi-tier-architectures-api-gateway-lambda.pdf#three-tier-architecture-overview>, acesso em 2023-02-11. x, 13
- [2] HackerRank: *Hackerrank developer skills report*. <https://info.hackerrank.com/rs/487-WAY-049/images/HackerRank-2020-Developer-Skills-Report.pdf>, acesso em 2023-02-09. x, 16, 18
- [3] Carneiro, Márcio: *Comunicação Digital e Jornalismo de Inserção*. Labcom Digital, 2016. 1, 2
- [4] Doctorow, Cory, Peter Suber, Tim Hubbard, Peter Murray-Rust, Jo Walsh, Prodromos Tsiavos e Erik Moeller.: *Open definition*. <https://opendefinition.org/od/2.0/pt-br/>, acesso em 2018-05-24. 1, 5
- [5] Reinsel, David, John Gantz e John Rydning: *Data age 2025: The evolution of data to life-critical*. IDC White Paper, 2017. <https://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf>, acesso em 2018-06-13. 1
- [6] Dietrich, Daniel, Jonathan Gray, Tim McNamara, Antti Poikola, P Pollock, Julian Tait e Ton Zijlstra: *Open data handbook*. http://opendatahandbook.org/guide/pt_BR/what-is-open-data/#o-que-é-aberto, acesso em 2018-05-25. 1
- [7] Council, Stony Brook Data Governance: *Data dictionary standards*. https://www.stonybrook.edu/commcms/irpe/about/data_governance/_files/DataDictionaryStandards.pdf, acesso em 2018-03-28. 2
- [8] Rashid, Sabbir, Jamie McCusker, Paulo Pinheiro, Marcello Bax, Henrique O. Santos, Jeanette Stingone, Amar Das e Deborah Mcguinness: *The semantic data dictionary – an approach for describing and annotating data*. Data Intelligence, 2:443–486, abril 2020. 2
- [9] Dresch, Lacerda e Antunes Jr: *Design Science Research: Método de Pesquisa para o avanço da Ciência e da Tecnologia*. Springer, 2015. 3
- [10] W3C, World Wide Web Consortium: *Dados abertos governamentais*, 1994. <http://www.w3c.br/divulgacao/pdf/dados-abertos-governamentais.pdf>, acesso em 2018-06-13. 6

- [11] Isotani, Seiji e Ig Ibert Bittencourt: *Dados Abertos Conectados*. Novatec Editora, 2015. 6
- [12] Jonathan Gray, Lucy Chambers, Liliana Bounegru: *The Data Journalism Handbook*. O'Reilly Media, 2012. 7
- [13] Berret, C. e C. Phillips: *Teaching data and computational journalism*. Columbia Journalism, 2016. <https://www.gitbook.com/book/columbiajournalism/teaching-data-computational-journalism/detail>. 7
- [14] Howard, A: *The art and science of data-driven journalism*. Columbia Journalism School, 2014. <http://towcenter.org/wp-content/uploads/2014/05/Tow-Center-Data-Driven-Journalism.pdf>. 7
- [15] Heravi, Bahareh R.: *3ws of data journalism education*. Journalism Practice, 2019. <https://www.tandfonline.com/doi/full/10.1080/17512786.2018.1463167>. 7
- [16] Halley, Magalhães, Luan Cardoso, Maristela Holanda, Edison Ishikawa e Marcio Victorino: *Análise de dados abertos sobre o ensino superior brasileiro*. CISTI, 2017. 7
- [17] Vanderlei Benedito da, Silva F. e Netto B. Leticia Souza: *Um estudo focado ao prouni através da análise de dados abertos: período de 2005 até 2016*. Prisma.com, (38):37–53, 2018. <https://www.proquest.com/scholarly-journals/um-estudo-focado-ao-prouni-atrav%C3%AAs-da-an%C3%A1lise-de/docview/2215854323/se-2>. 8
- [18] Araújo, Lucas, de Ramos e Jairo F. de Souza: *Aumentando a transparência do governo por meio da transformação de dados governamentais abertos em dados ligados/increasing government transparency by transforming open government data into linked data*. Revista Electronica de Sistemas de Informação, 10(1):1–15, 2011. <https://www.proquest.com/scholarly-journals/aumentando-transpar%C3%Aancia-do-governo-por-meio-da/docview/912389288/se-2>, Copyright - Copyright Faculdade Cenecista de Campo Largo - FACECLA 2011; Última atualização em - 2014-04-21. 8
- [19] Microsoft: *N-tier architecture style*. <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>, acesso em 2023-02-07. 10
- [20] Fowler, Martin: *Patterns of Enterprise Application Architecture*. Addison Wesley, 2002. 11
- [21] Spinellis, D.: *Choosing a programming language*. IEEE Software, 23(4):62–63, 2006. 16
- [22] ReactJS: *Tutorial: Intro to react*. <https://reactjs.org/tutorial/tutorial.html#developer-tools>, acesso em 2023-02-09. 17
- [23] Next.js: *Tutorial: Intro to react*. <https://nextjs.org/learn/basics/create-nextjs-app>, acesso em 2023-02-09. 19

- [24] Next.js: *What is axios*. <https://axios-http.com/docs/intro>, acesso em 2023-02-09. 19
- [25] Foundation, OpenJS: *Sobre node.js*. <https://nodejs.org/pt-br/about/>, acesso em 2023-02-09. 20
- [26] Mozilla: *Introdução express/node*. https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction, acesso em 2023-02-10. 20, 24
- [27] Knex: *Knex guide*. <https://knexjs.org/guide/>, acesso em 2023-02-10. 20
- [28] Heroku: *Heroku wat*. <https://www.heroku.com/what>, acesso em 2023-02-13. 21
- [29] Brooke, John: *Sus: A quick and dirty usability scale*. Usability Eval. Ind., 189, novembro 1995. 35
- [30] SAURO, Jeff.: *Measuring usability with the system usability scale (sus)*. 2009. <https://measuringu.com/sus/>, acesso em 2023-02-13. 35

Anexo I

Modelos lógicos para criação do banco de dados

MER Microdados INEP Censo da Educação Superior do Brasil

Felipe Cordero Pires Magalhães | October 7, 2021

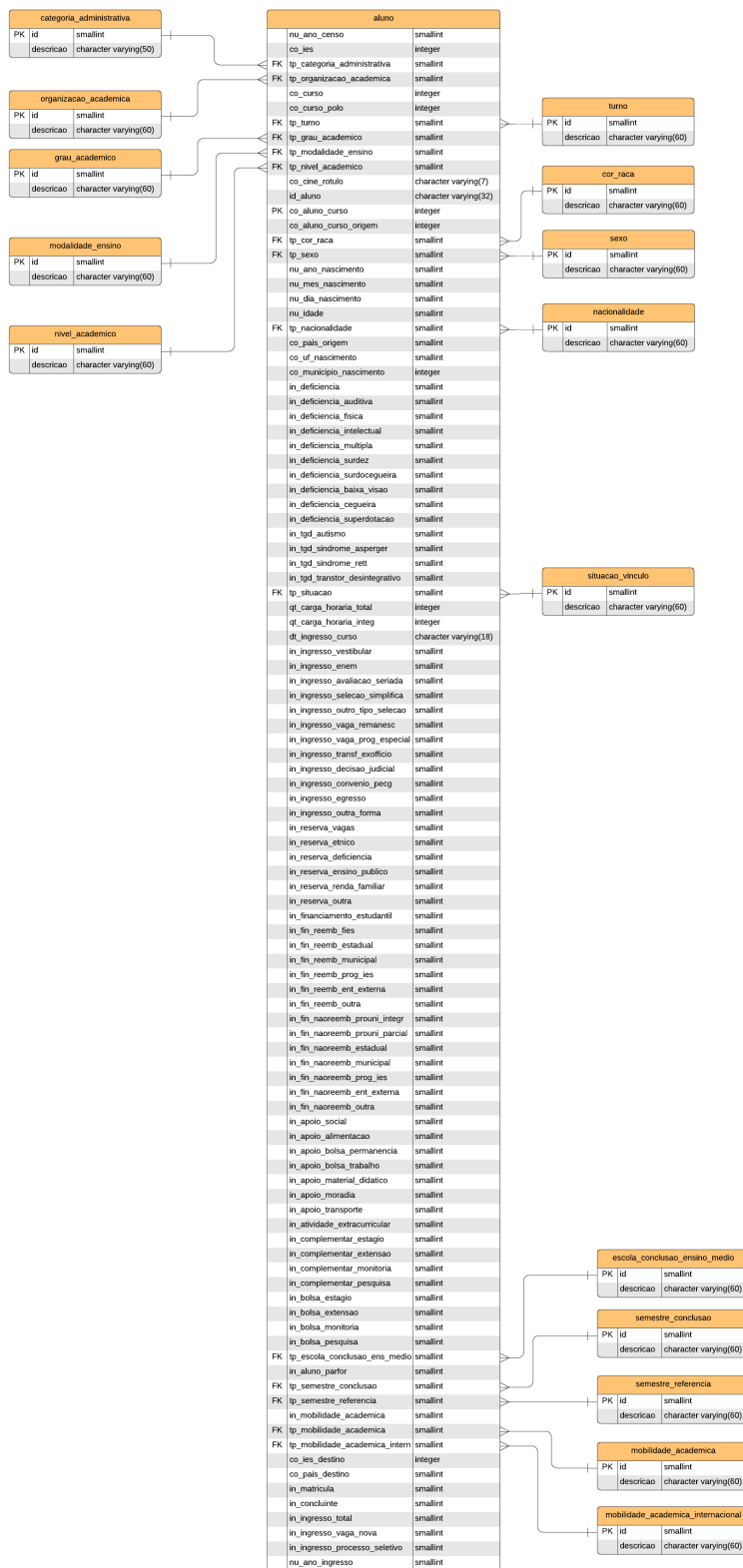


Figura I.1: Modelo Lógico Entidade Aluno. Fonte: própria (2022).

MER Microdados INEP Censo da Educação Superior do Brasil

Felipe Cardoso Pezza Magalhães | October 7, 2021



Figura I.2: Modelo Lógico Entidade Curso. Fonte: própria (2022).

MER Microdados INEP Censo da Educação Superior do Brasil

Felipe Cordeiro Pires Magalhães | October 7, 2021

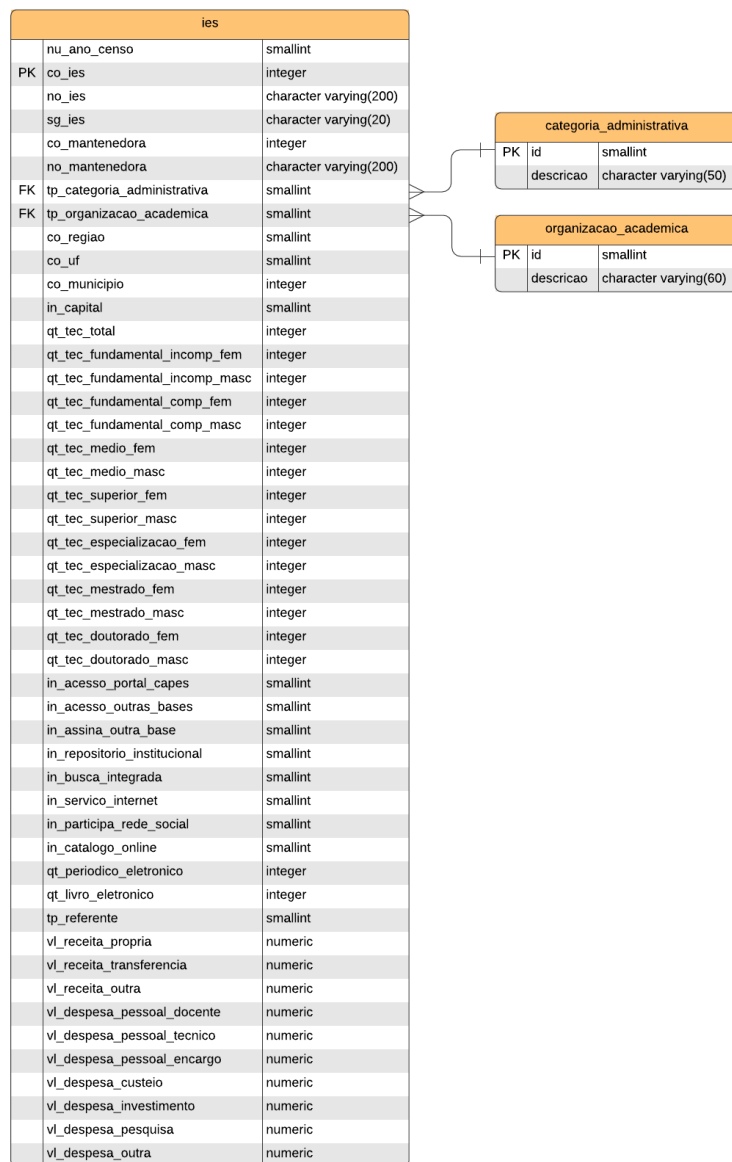


Figura I.3: Modelo Lógico Entidade IES. Fonte: própria (2022).

Anexo II

Queries SQL executadas para criação do banco de dados

Queries para criar e popular tabela ALUNO

```
CREATE TABLE alunos (  
    nu_ano_censo smallint,  
    co_ies integer,  
    tp_categoria_administrativa smallint,  
    tp_organizacao_academica smallint,  
    co_curso integer,  
    co_curso_polo integer,  
    tp_turno smallint,  
    tp_grau_academico smallint,  
    tp_modalidade_ensino smallint,  
    tp_nivel_academico smallint,  
    co_cine_rotulo varchar(7),  
    id_aluno varchar(32),  
    co_aluno_curso integer PRIMARY KEY,  
    co_aluno_curso_origem integer,  
    tp_cor_raca smallint,  
    tp_sexo smallint,  
    nu_ano_nascimento smallint,  
    nu_mes_nascimento smallint,  
    nu_dia_nascimento smallint,  
    nu_idade smallint,  
    tp_nacionalidade smallint,  
    co_pais_origem smallint,  
    co_uf_nascimento smallint,  
    co_municipio_nascimento integer,  
    in_deficiencia smallint,  
    in_deficiencia_auditiva smallint,  
    in_deficiencia_fisica smallint,  
    in_deficiencia_intelectual smallint,  
    in_deficiencia_multipla smallint,  
    in_deficiencia_surdez smallint,  
    in_deficiencia_surdocegueira smallint,  
    in_deficiencia_baixa_visao smallint,  
    in_deficiencia_egueira smallint,  
    in_deficiencia_superdotacao smallint,  
    in_tgd_autismo smallint,
```

in_tgd_sindrome_asperger smallint,
in_tgd_sindrome_rett smallint,
in_tgd_transtor_desintegrativo smallint,
tp_situacao smallint,
qt_carga_horaria_total integer,
qt_carga_horaria_integ integer,
dt_ingresso_curso character varying(18),
in_ingresso_vestibular smallint,
in_ingresso_enem smallint,
in_ingresso_avaliacao_seriada smallint,
in_ingresso_selecao_simplifica smallint,
in_ingresso_outro_tipo_selecao smallint,
in_ingresso_vaga_remanesc smallint,
in_ingresso_vaga_prog_especial smallint,
in_ingresso_transf_exofficio smallint,
in_ingresso_decisao_judicial smallint,
in_ingresso_convenio_pecg smallint,
in_ingresso_egresso smallint,
in_ingresso_outra_forma smallint,
in_reserva_vagas smallint,
in_reserva_etnico smallint,
in_reserva_deficiencia smallint,
in_reserva_ensino_publico smallint,
in_reserva_renda_familiar smallint,
in_reserva_outra smallint,
in_financiamento_estudantil smallint,
in_fin_reemb_fies smallint,
in_fin_reemb_estadual smallint,
in_fin_reemb_municipal smallint,
in_fin_reemb_prog_ies smallint,
in_fin_reemb_ent_externa smallint,
in_fin_reemb_outra smallint,
in_fin_naoreemb_prouni_integr smallint,
in_fin_naoreemb_prouni_parcial smallint,
in_fin_naoreemb_estadual smallint,
in_fin_naoreemb_municipal smallint,
in_fin_naoreemb_prog_ies smallint,

```

in_fin_naoreemb_ent_externa smallint,
in_fin_naoreemb_outra smallint,
in_apoio_social smallint,
in_apoio_alimentacao smallint,
in_apoio_bolsa_permanencia smallint,
in_apoio_bolsa_trabalho smallint,
in_apoio_material_didatico smallint,
in_apoio_moradia smallint,
in_apoio_transporte smallint,
in_atividade_extracurricular smallint,
in_complementar_estagio smallint,
in_complementar_extensao smallint,
in_complementar_monitoria smallint,
in_complementar_pesquisa smallint,
in_bolsa_estagio smallint,
in_bolsa_extensao smallint,
in_bolsa_monitoria smallint,
in_bolsa_pesquisa smallint,
tp_escola_conclusao_ens_medio smallint,
in_aluno_parfor smallint,
tp_semestre_conclusao smallint,
tp_semestre_referencia smallint,
in_mobilidade_academica smallint,
tp_mobilidade_academica smallint,
tp_mobilidade_academica_intern smallint,
co_ies_destino integer,
co_pais_destino smallint,
in_matricula smallint,
in_concluente smallint,
in_ingresso_total smallint,
in_ingresso_vaga_nova smallint,
in_ingresso_processo_seletivo smallint,
nu_ano_ingresso smallint,
CONSTRAINT fk_tp_categoria_administrativa
    FOREIGN KEY(tp_categoria_administrativa)
    REFERENCES categoria_administrativa(id),
CONSTRAINT fk_tp_organizacao_academica

```

```

FOREIGN KEY(tp_organizacao_academica)
REFERENCES organizacao_academica(id),
CONSTRAINT fk_tp_turno
FOREIGN KEY(tp_turno)
REFERENCES turno(id),
CONSTRAINT fk_tp_grau_academico
FOREIGN KEY(tp_grau_academico)
REFERENCES grau_academico(id),
CONSTRAINT fk_tp_modalidade_ensino
FOREIGN KEY(tp_modalidade_ensino)
REFERENCES modalidade_ensino(id),
CONSTRAINT fk_tp_nivel_academico
FOREIGN KEY(tp_nivel_academico)
REFERENCES nivel_academico(id),
CONSTRAINT fk_tp_cor_raca
FOREIGN KEY(tp_cor_raca)
REFERENCES cor_raca(id),
CONSTRAINT fk_tpsexo
FOREIGN KEY(tpsexo)
REFERENCES sexo(id),
CONSTRAINT fk_tp_nacionalidade
FOREIGN KEY(tp_nacionalidade)
REFERENCES nacionalidade(id),
CONSTRAINT fk_tp_situacao
FOREIGN KEY(tp_situacao)
REFERENCES situacao_vinculo(id),
CONSTRAINT fk_tp_escola_conclusao_ens_medio
FOREIGN KEY(tp_escola_conclusao_ens_medio)
REFERENCES escola_conclusao_ensino_medio(id),
CONSTRAINT fk_tp_semestre_conclusao
FOREIGN KEY(tp_semestre_conclusao)
REFERENCES semestre_conclusao(id),
CONSTRAINT fk_tp_semestre_referencia
FOREIGN KEY(tp_semestre_referencia)
REFERENCES semestre_referencia(id),
CONSTRAINT fk_tp_mobilidade_academica
FOREIGN KEY(tp_mobilidade_academica)

```



```
REFERENCES mobilidade_academica(id),
CONSTRAINT fk_tp_mobilidade_academica_intern
FOREIGN KEY(tp_mobilidade_academica_intern)
REFERENCES mobilidade_academica_internacional(id)
)
```

COPY alunos

```
From '~/microdados_ed_superior_2019/dados/SUP_ALUNO_2019.CSV'
DELIMITER '|'
ENCODING 'LATIN1'
CSV HEADER;
```

Filtro para popular apenas com dados de alunos vinculados a instituições do Distrito Federal

```
INSERT INTO alunos_df
SELECT *
FROM alunos
WHERE co_ies in (SELECT co_ies FROM ies_df);
```

Queries para criar e popular tabela CURSO

```
CREATE TABLE curso_2019(  
    nu_ano_censo smallint,  
    co_ies integer,  
    tp_categoria_administrativa smallint,  
    tp_organizacao_academica smallint,  
    co_local_oferta integer,  
    co_uf smallint,  
    co_municipio integer,  
    in_capital smallint,  
    co_curso integer PRIMARY KEY,  
    no_curso character varying(200),  
    tp_situacao smallint,  
    co_cine_rotulo character varying(7),  
    tp_grau_academico smallint,  
    tp_modalidade_ensino smallint,  
    tp_nivel_academico smallint,  
    in_gratuito smallint,  
    tp_atributo_ingresso smallint,  
    nu_carga_horaria integer,  
    dt_inicio_funcionamento character varying(18),  
    dt_autorizacao_curso character varying(18),  
    in_ajuda_deficiente smallint,  
    in_material_digital smallint,  
    in_material_ampliado smallint,  
    in_material_tatil smallint,  
    in_material_impreso smallint,  
    in_material_audio smallint,  
    in_material_braille smallint,  
    in_material_libras smallint,  
    in_disciplina_libras smallint,  
    in_tradutor_libras smallint,  
    in_guia_interprete smallint,  
    in_recursos_comunicacao smallint,  
    in_recursos_informatica smallint,  
    in_integral smallint,
```

in_matutino smallint,
in_vespertino smallint,
in_noturno smallint,
nu_integralizacao_integral numeric,
nu_integralizacao_matutino numeric,
nu_integralizacao_vespertino numeric,
nu_integralizacao_noturno numeric,
nu_integralizacao_ead numeric,
in_oferece_disc_semi_pres smallint,
nu_perc_carga_semi_pres numeric,
in_possui_laboratorio smallint,
qt_insc_vaga_nova_integral integer,
qt_insc_vaga_nova_matutino integer,
qt_insc_vaga_nova_vespertino integer,
qt_insc_vaga_nova_noturno integer,
qt_insc_vaga_nova_ead integer,
qt_insc_vaga_reman_integral integer,
qt_insc_vaga_reman_matutino integer,
qt_insc_vaga_reman_vespertino integer,
qt_insc_vaga_reman_noturno integer,
qt_insc_vaga_reman_ead integer,
qt_insc_prog_esp_integral integer,
qt_insc_prog_esp_matutino integer,
qt_insc_prog_esp_vespertino integer,
qt_insc_prog_esp_noturno integer,
qt_insc_prog_esp_ead integer,
qt_insc_principal_integral integer,
qt_insc_principal_matutino integer,
qt_insc_principal_vespertino integer,
qt_insc_principal_noturno integer,
qt_insc_principal_ead integer,
qt_insc_outra_vaga_integral integer,
qt_insc_outra_vaga_matutino integer,
qt_insc_outra_vaga_vespertino integer,
qt_insc_outra_vaga_noturno integer,
qt_insc_outra_vaga_ead integer,
qt_insc_anual_integral integer,

qt_insc_anual_matutino integer,
qt_insc_anual_vespertino integer,
qt_insc_anual_noturno integer,
qt_insc_anual_ead integer,
qt_vagas_novas_integral integer,
qt_vagas_novas_matutino integer,
qt_vagas_novas_vespertino integer,
qt_vagas_novas_noturno integer,
qt_vagas_novas_ead integer,
qt_vagas_reman_integral integer,
qt_vagas_reman_matutino integer,
qt_vagas_reman_vespertino integer,
qt_vagas_reman_noturno integer,
qt_vagas_reman_ead integer,
qt_vagas_prog_esp_integral integer,
qt_vagas_prog_esp_matutino integer,
qt_vagas_prog_esp_vespertino integer,
qt_vagas_prog_esp_noturno integer,
qt_vagas_prog_esp_ead integer,
qt_vagas_principal_integral integer,
qt_vagas_principal_matutino integer,
qt_vagas_principal_vespertino integer,
qt_vagas_principal_noturno integer,
qt_vagas_principal_ead integer,
qt_vagas_outras_integral integer,
qt_vagas_outras_matutino integer,
qt_vagas_outras_vespertino integer,
qt_vagas_outras_noturno integer,
qt_vagas_outras_ead integer,
qt_vagas_anual_integral integer,
qt_vagas_anual_matutino integer,
qt_vagas_anual_vespertino integer,
qt_vagas_anual_noturno integer,
qt_vagas_anual_ead integer,
qt_matricula_total integer,
qt_concluinte_total integer,
qt_ingresso_total integer,

```

qt_ingresso_vaga_nova integer,
qt_ingresso_processo_seletivo integer,
qt_vaga_total integer,
qt_inscrito_total integer,
CONSTRAINT fk_tp_categoria_administrativa
    FOREIGN KEY(tp_categoria_administrativa)
    REFERENCES categoria_administrativa(id),
CONSTRAINT fk_tp_organizacao_academica
    FOREIGN KEY(tp_organizacao_academica)
    REFERENCES organizacao_academica(id),
CONSTRAINT fk_tp_situacao
    FOREIGN KEY(tp_situacao)
    REFERENCES situacao_curso(id),
CONSTRAINT fk_tp_grau_academico
    FOREIGN KEY(tp_grau_academico)
    REFERENCES grau_academico(id),
CONSTRAINT fk_tp_modalidade_ensino
    FOREIGN KEY(tp_modalidade_ensino)
    REFERENCES modalidade_ensino(id),
CONSTRAINT fk_tp_nivel_academico
    FOREIGN KEY(tp_nivel_academico)
    REFERENCES nivel_academico(id),
CONSTRAINT fk_tp_atributo_ingresso
    FOREIGN KEY(tp_atributo_ingresso)
    REFERENCES atributo_ingresso(id)
)

COPY curso_2019
FROM '~/microdados_ed_superior_2019/dados/SUP_CURSO_2019.CSV'
DELIMITER '|'
ENCODING 'LATIN1'
CSV HEADER;

```

Filtro para popular apenas com dados de cursos vinculados a instituições do Distrito Federal

```

INSERT INTO curso
select *

```

```

from curso_2019
WHERE EXISTS (
select * from ies_2019
WHERE ies_2019.co_ies = curso_2019.co_ies
)

```

Alterações em 2015 a 2017 para adequação e padronização de tabelas com anos posteriores:

- Remove: co_cine_rotulo e qt_inscrito_total integer.
- Adiciona: co_ocde_area_geral, co_ocde_area_especifica, co_ocde_area_detalhada, co_ocde.

```

CREATE TABLE curso_2017(
    nu_ano_censo smallint,
    co_ies integer,
    tp_categoria_administrativa smallint,
    tp_organizacao_academica smallint,
    co_local_oferta integer,
    co_uf smallint,
    co_municipio integer,
    in_capital smallint,
    co_curso integer PRIMARY KEY,
    no_curso character varying(200),
    tp_situacao smallint,
    co_ocde_area_geral character varying(1),
    co_ocde_area_especifica character varying(2),
    co_ocde_area_detalhada character varying(3),
    co_ocde character varying(15),
    tp_grau_academico smallint,
    tp_modalidade_ensino smallint,
    tp_nivel_academico smallint,
    in_gratuito smallint,
    tp_atributo_ingresso smallint,
    nu_carga_horaria integer,
    dt_inicio_funcionamento character varying(18),
    dt_autorizacao_curso character varying(18),

```

in_ajuda_deficiente smallint,
in_material_digital smallint,
in_material_ampliado smallint,
in_material_tatil smallint,
in_material_impreso smallint,
in_material_audio smallint,
in_material_braille smallint,
in_material_libras smallint,
in_disciplina_libras smallint,
in_tradutor_libras smallint,
in_guia_interprete smallint,
in_recursos_comunicacao smallint,
in_recursos_informatica smallint,
in_integral smallint,
in_matutino smallint,
in_vespertino smallint,
in_noturno smallint,
nu_integralizacao_integral numeric,
nu_integralizacao_matutino numeric,
nu_integralizacao_vespertino numeric,
nu_integralizacao_noturno numeric,
nu_integralizacao_ead numeric,
in_oferece_disc_semi_pres smallint,
nu_perc_carga_semi_pres numeric,
in_possui_laboratorio smallint,
qt_insc_vaga_nova_integral integer,
qt_insc_vaga_nova_matutino integer,
qt_insc_vaga_nova_vespertino integer,
qt_insc_vaga_nova_noturno integer,
qt_insc_vaga_nova_ead integer,
qt_insc_vaga_reman_integral integer,
qt_insc_vaga_reman_matutino integer,
qt_insc_vaga_reman_vespertino integer,
qt_insc_vaga_reman_noturno integer,
qt_insc_vaga_reman_ead integer,
qt_insc_prog_esp_integral integer,
qt_insc_prog_esp_matutino integer,

qt_insc_prog_esp_vespertino integer,
qt_insc_prog_esp_noturno integer,
qt_insc_prog_esp_ead integer,
qt_insc_principal_integral integer,
qt_insc_principal_matutino integer,
qt_insc_principal_vespertino integer,
qt_insc_principal_noturno integer,
qt_insc_principal_ead integer,
qt_insc_outra_vaga_integral integer,
qt_insc_outra_vaga_matutino integer,
qt_insc_outra_vaga_vespertino integer,
qt_insc_outra_vaga_noturno integer,
qt_insc_outra_vaga_ead integer,
qt_insc_anual_integral integer,
qt_insc_anual_matutino integer,
qt_insc_anual_vespertino integer,
qt_insc_anual_noturno integer,
qt_insc_anual_ead integer,
qt_vagas_novas_integral integer,
qt_vagas_novas_matutino integer,
qt_vagas_novas_vespertino integer,
qt_vagas_novas_noturno integer,
qt_vagas_novas_ead integer,
qt_vagas_reman_integral integer,
qt_vagas_reman_matutino integer,
qt_vagas_reman_vespertino integer,
qt_vagas_reman_noturno integer,
qt_vagas_reman_ead integer,
qt_vagas_prog_esp_integral integer,
qt_vagas_prog_esp_matutino integer,
qt_vagas_prog_esp_vespertino integer,
qt_vagas_prog_esp_noturno integer,
qt_vagas_prog_esp_ead integer,
qt_vagas_principal_integral integer,
qt_vagas_principal_matutino integer,
qt_vagas_principal_vespertino integer,
qt_vagas_principal_noturno integer,


```

qt_vagas_principal_ead integer,
qt_vagas_outras_integral integer,
qt_vagas_outras_matutino integer,
qt_vagas_outras_vespertino integer,
qt_vagas_outras_noturno integer,
qt_vagas_outras_ead integer,
qt_vagas_anual_integral integer,
qt_vagas_anual_matutino integer,
qt_vagas_anual_vespertino integer,
qt_vagas_anual_noturno integer,
qt_vagas_anual_ead integer,
qt_matricula_total integer,
qt_concluinte_total integer,
qt_ingresso_total integer,
qt_ingresso_vaga_nova integer,
qt_ingresso_processo_seletivo integer,
qt_vaga_total integer,
CONSTRAINT fk_tp_categoria_administrativa
    FOREIGN KEY(tp_categoria_administrativa)
    REFERENCES categoria_administrativa(id),
CONSTRAINT fk_tp_organizacao_academica
    FOREIGN KEY(tp_organizacao_academica)
    REFERENCES organizacao_academica(id),
CONSTRAINT fk_tp_situacao
    FOREIGN KEY(tp_situacao)
    REFERENCES situacao_curso(id),
CONSTRAINT fk_tp_grau_academico
    FOREIGN KEY(tp_grau_academico)
    REFERENCES grau_academico(id),
CONSTRAINT fk_tp_modalidade_ensino
    FOREIGN KEY(tp_modalidade_ensino)
    REFERENCES modalidade_ensino(id),
CONSTRAINT fk_tp_nivel_academico
    FOREIGN KEY(tp_nivel_academico)
    REFERENCES nivel_academico(id),
CONSTRAINT fk_tp_atributo_ingresso
    FOREIGN KEY(tp_atributo_ingresso)

```

```
) REFERENCES atributo_ingresso(id)
```

Queries para criar e popular tabela IES

```
CREATE TABLE ies(  
    nu_ano_censo smallint,  
    co_ies integer PRIMARY KEY,  
    no_ies character varying(200),  
    sg_ies character varying(200),  
    co_mantenedora integer,  
    no_mantenedora character varying(200),  
    tp_categoria_administrativa smallint,  
    tp_organizacao_academica smallint,  
    co_regiao smallint,  
    co_uf smallint,  
    co_municipio integer,  
    in_capital smallint,  
    qt_tec_total integer,  
    qt_tec_fundamental_incomp_fem integer,  
    qt_tec_fundamental_incomp_masc integer,  
    qt_tec_fundamental_comp_fem integer,  
    qt_tec_fundamental_comp_masc integer,  
    qt_tec_medio_fem integer,  
    qt_tec_medio_masc integer,  
    qt_tec_superior_fem integer,  
    qt_tec_superior_masc integer,  
    qt_tec_especializacao_fem integer,  
    qt_tec_especializacao_masc integer,  
    qt_tec_mestrado_fem integer,  
    qt_tec_mestrado_masc integer,  
    qt_tec_doutorado_fem integer,  
    qt_tec_doutorado_masc integer,  
    in_acesso_portal_capes smallint,  
    in_acesso_outras_bases smallint,  
    in_assina_outra_base smallint,  
    in_repositorio_institucional smallint,  
    in_busca_integrada smallint,  
    in_servico_internet smallint,  
    in_participa_rede_social smallint,
```

```

in_catalogo_online smallint,
qt_periodico_eletronico integer,
qt_livro_eletronico integer,
tp_referente smallint,
vl_receita_propria numeric,
vl_receita_transferencia numeric,
vl_receita_outra numeric,
vl_despesa_pessoal_docente numeric,
vl_despesa_pessoal_tecnico numeric,
vl_despesa_pessoal_encargo numeric,
vl_despesa_custeio numeric,
vl_despesa_investimento numeric,
vl_despesa_pesquisa numeric,
vl_despesa_outra numeric,
CONSTRAINT fk_tp_categoria_administrativa
    FOREIGN KEY(tp_categoria_administrativa)
    REFERENCES categoria_administrativa(id),
CONSTRAINT fk_tp_organizacao_academica
    FOREIGN KEY(tp_organizacao_academica)
    REFERENCES organizacao_academica(id)
)

```

```

COPY ies
FROM '~/microdados_ed_superior_2019/dados/SUP_IES_2019.CSV'
DELIMITER '|'
ENCODING 'LATIN1'
CSV HEADER;

```

Filtro para popular apenas com dados de cursos vinculados a instituições do Distrito Federal. Rodar novamente query de criação de tabela trocando nome para `ies_2019`

```

INSERT INTO ies_2019
SELECT * FROM ies
WHERE ies.co_uf = 53

```

53 é o código do IBGE que referencia o Distrito Federal. Alterações em 2015 a 2017 para adequação e padronização de tabelas com anos posteriores:

- Remove: o campo `in_assina_outra_base`

```

CREATE TABLE ies(
    nu_ano_censo smallint,
    co_ies integer PRIMARY KEY,
    no_ies character varying(200),
    sg_ies character varying(20),
    co_mantenedora integer,
    no_mantenedora character varying(200),
    tp_categoria_administrativa smallint,
    tp_organizacao_academica smallint,
    co_regiao smallint,
    co_uf smallint,
    co_municipio integer,
    in_capital smallint,
    qt_tec_total integer,
    qt_tec_fundamental_incomp_fem integer,
    qt_tec_fundamental_incomp_masc integer,
    qt_tec_fundamental_comp_fem integer,
    qt_tec_fundamental_comp_masc integer,
    qt_tec_medio_fem integer,
    qt_tec_medio_masc integer,
    qt_tec_superior_fem integer,
    qt_tec_superior_masc integer,
    qt_tec_especializacao_fem integer,
    qt_tec_especializacao_masc integer,
    qt_tec_mestrado_fem integer,
    qt_tec_mestrado_masc integer,
    qt_tec_doutorado_fem integer,
    qt_tec_doutorado_masc integer,
    in_acesso_portal_capes smallint,
    in_acesso_outras_bases smallint,
    in_repositorio_institucional smallint,
    in_busca_integrada smallint,
    in_servico_internet smallint,
    in_participa_rede_social smallint,
    in_catalogo_online smallint,
    qt_periodico_eletronico integer,
    qt_livro_eletronico integer,

```

```
tp_referente smallint,  
vl_receita_propria numeric,  
vl_receita_transferencia numeric,  
vl_receita_outra numeric,  
vl_despesa_pessoal_docente numeric,  
vl_despesa_pessoal_tecnico numeric,  
vl_despesa_pessoal_encargo numeric,  
vl_despesa_custeio numeric,  
vl_despesa_investimento numeric,  
vl_despesa_pesquisa numeric,  
vl_despesa_outra numeric,  
CONSTRAINT fk_tp_categoria_administrativa  
    FOREIGN KEY(tp_categoria_administrativa)  
    REFERENCES categoria_administrativa(id),  
CONSTRAINT fk_tp_organizacao_academica  
    FOREIGN KEY(tp_organizacao_academica)  
    REFERENCES organizacao_academica(id)  
)
```

Queries para criar e popular tabelas auxiliares para categoria ALUNOS

```
CREATE TABLE categoria_administrativa (  
    id smallint PRIMARY KEY,  
    descricao varchar(50) NOT NULL  
);  
  
INSERT INTO categoria_administrativa(id, descricao)  
VALUES  
    (1, 'Pública Federal'),  
    (2, 'Pública Estadual'),  
    (3, 'Pública Municipal'),  
    (4, 'Privada com fins lucrativos'),  
    (5, 'Privada sem fins lucrativos'),  
    (6, 'Privada - Particular em sentido estrito'),  
    (7, 'Especial'),  
    (8, 'Privada comunitária'),  
    (9, 'Privada confessional');
```

```
CREATE TABLE organizacao_academica (  
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);  
  
INSERT INTO organizacao_academica(id, descricao)  
VALUES  
    (1, 'Universidade'),  
    (2, 'Centro Universitário'),  
    (3, 'Faculdade'),  
    (4, 'Instituto Federal de Educação, Ciência e Tecnologia'),  
    (5, 'Centro Federal de Educação Tecnológica');
```

```
CREATE TABLE turno (  
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);
```

```
INSERT INTO turno(id, descricao)
VALUES
    (1, 'Matutino'),
    (2, 'Vespertino'),
    (3, 'Noturno'),
    (4, 'Integral');
```

```
CREATE TABLE grau_academico (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```

```
INSERT INTO grau_academico(id, descricao)
VALUES
    (1, 'Bacharelado'),
    (2, 'Licenciatura'),
    (3, 'Tecnológico'),
    (4, 'Bacharelado e Licenciatura');
```

```
CREATE TABLE modalidade_ensino (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```

```
INSERT INTO modalidade_ensino(id, descricao)
VALUES
    (1, 'Presencial'),
    (2, 'Curso a distância');
```

```
CREATE TABLE nivel_academico (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```



```
INSERT INTO nivel_academico(id, descricao)
VALUES
    (1, 'Graduação'),
    (2, 'Seqüencial de Formação Específica');
```

```
CREATE TABLE cor_raca (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```

```
INSERT INTO cor_raca(id, descricao)
VALUES
    (0, 'Aluno não quis declarar cor/raça '),
    (1, 'Branca'),
    (2, 'Preta'),
    (3, 'Parda'),
    (4, 'Amarela'),
    (5, 'Indígena'),
    (9, 'Não dispõe da informação (Não resposta)');
```

```
CREATE TABLE sexo (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```

```
INSERT INTO sexo(id, descricao)
VALUES
    (1, 'Feminino'),
    (2, 'Masculino');
```

```
CREATE TABLE nacionalidade (
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
```

```
);
```

```
INSERT INTO nacionalidade(id, descricao)
```

```
VALUES
```

```
(1, 'Brasileira'),  
(2, 'Brasileira - nascido no exterior ou naturalizado'),  
(3, 'Estrangeira');
```

```
CREATE TABLE situacao_vinculo (
```

```
id smallint PRIMARY KEY,  
descricao varchar(60) NOT NULL
```

```
);
```

```
INSERT INTO situacao_vinculo(id, descricao)
```

```
VALUES
```

```
(2, 'Cursando'),  
(3, 'Matrícula trancada'),  
(4, 'Desvinculado do curso'),  
(5, 'Transferido para outro curso da mesma IES'),  
(6, 'Formado'),  
(7, 'Falecido');
```

```
CREATE TABLE escola_conclusao_ensino_medio (
```

```
id smallint PRIMARY KEY,  
descricao varchar(60) NOT NULL
```

```
);
```

```
INSERT INTO escola_conclusao_ensino_medio(id, descricao)
```

```
VALUES
```

```
(1, 'Pública'),  
(2, 'Privada'),  
(9, 'Não dispõe da informação (Não resposta)');
```

```
CREATE TABLE semestre_conclusao (
```

```
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);
```

```
INSERT INTO semestre_conclusao(id, descricao)  
VALUES  
    (1, 'Primeiro semestre'),  
    (2, 'Segundo semestre');
```

```
CREATE TABLE semestre_referencia (  
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);
```

```
INSERT INTO semestre_referencia(id, descricao)  
VALUES  
    (1, 'Primeiro semestre'),  
    (2, 'Segundo semestre');
```

```
CREATE TABLE mobilidade_academica (  
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);
```

```
INSERT INTO mobilidade_academica(id, descricao)  
VALUES  
    (1, 'Nacional'),  
    (2, 'Internacional');
```

```
CREATE TABLE mobilidade_academica_internacional (  
    id smallint PRIMARY KEY,  
    descricao varchar(60) NOT NULL  
);
```

```
INSERT INTO mobilidade_academica_internacional(id, descricao)
VALUES
    (1, 'Intercâmbio'),
    (2, 'Ciências sem fronteiras');
```

```
CREATE TABLE possui_deficiencia(
    id smallint PRIMARY KEY,
    descricao varchar(60) NOT NULL
);
```

```
INSERT INTO possui_deficiencia(id, descricao)
VALUES
    (0, 'Não'),
    (1, 'Sim'),
    (9, 'Não dispõe de informação (Não resposta)');
```

Queries para criar e popular tabelas auxiliares para categoria CURSOS

```
CREATE TABLE situacao_curso (  
    id smallint PRIMARY KEY,  
    descricao varchar(15) NOT NULL  
);
```

```
INSERT INTO situacao_curso(id, descricao)  
VALUES  
    (1, 'Em atividade'),  
    (2, 'Extinto'),  
    (3, 'Em extinção');
```

```
CREATE TABLE atributo_ingresso (  
    id smallint PRIMARY KEY,  
    descricao varchar(45) NOT NULL  
);
```

```
INSERT INTO atributo_ingresso(id, descricao)  
VALUES  
    (0, 'Normal'),  
    (1, 'Área Básica de Ingresso'),  
    (2, 'Bacharelado ou Licenciatura Interdisciplinar');
```

```
CREATE TABLE co_uf_ibge (  
    id smallint PRIMARY KEY,  
    estado character varying(25),  
    sigla character varying(2)  
)
```

```
INSERT INTO co_uf_ibge(id, estado, sigla)  
VALUES  
    (11, 'Rondônia', 'RO'),  
    (12, 'Acre', 'AC'),
```

- (13, 'Amazonas', 'AM'),
- (14, 'Roraima', 'RR'),
- (15, 'Pará', 'PA'),
- (16, 'Amapá', 'AP'),
- (17, 'Tocantins', 'TO'),
- (21, 'Maranhão', 'MA'),
- (22, 'Piauí', 'PI'),
- (23, 'Ceará', 'CE'),
- (24, 'Rio Grande do Norte', 'RN'),
- (25, 'Paraíba', 'PB'),
- (26, 'Pernambuco', 'PE'),
- (27, 'Alagoas', 'AL'),
- (28, 'Sergipe', 'SE'),
- (29, 'Bahia', 'BA'),
- (31, 'Minas Gerais', 'MG'),
- (32, 'Espírito Santo', 'ES'),
- (33, 'Rio de Janeiro', 'RJ'),
- (35, 'São Paulo', 'SP'),
- (41, 'Paraná', 'PR'),
- (42, 'Santa Catarina', 'SC'),
- (43, 'Rio Grande do Sul', 'RS'),
- (50, 'Mato Grosso do Sul', 'MS'),
- (51, 'Mato Grosso', 'MT'),
- (52, 'Goiás', 'GO'),
- (53, 'Distrito Federal', 'DF');