UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

PEDRO SALGADO PERRONE

Uma ferramenta web para a automatização de relatórios da Sociedade Brasileira de Computação sobre dados referentes ao ensino nacional de tecnologia

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof[a]. Dr[a]. Renata Galante

Porto Alegre
2023

AGRADECIMENTOS

Agradeço à todos os que de alguma forma me apoiaram ao longo da graduação. Aos meus colegas de trabalho, que tanto me ensinaram, aos meus amigos, que tantas vezes compreenderam minha ausência, e, especialmente, à minha família, que sempre me acompanhou tão de perto e com tanto empenho.

Muito obrigado.

RESUMO

Anualmente a Sociedade Brasileira de Computação confecciona um relatório com dados sobre o Ensino Superior de tecnologia no Brasil. Os relatórios, contruídos utilizando dados coletados pelo Censo de Educação Superior do INEP e disponibilizados de maneira pública, são gerados de maneira manual e são publicados em formato de um arquivo PDF. Visando a redução do custo operacional deste processo e uma maior flexibilidade no seu resultado, o objetivo deste trabalho é o desenvolvimento de uma ferramenta que automatize a geração destes relatórios, com o processamento automatizado de novos dados e filtros que podem ser modificados a qualquer momento. O resultado foi uma plataforma *web* construída com Elixir, JavaScript e PostgreSQL que recebe novas planilhas como entrada, alimenta o banco de dados e exibe os dados inseridos conforme os filtros definidos pelo usuário.

Palavras-chave: Visualização de dados. Ensino de tecnologia. Educação superior.

A web tool for automating reports of the Brazilian Computing Society on data related to national technology education

ABSTRACT

Annually, the Brazilian Computing Society produces a report with data on Higher Education in technology in Brazil. The reports, constructed using data collected by the INEP's Higher Education Census and made publicly available, are generated manually and published in the form of a PDF file. In order to reduce the operational cost of this process and provide greater flexibility in its output, the objective of this work is to develop a tool that automates the generation of these reports, with automated processing of new data and filters that can be modified at any time. The result was a *web* platform built with Elixir, JavaScript and PostgreSQL that receives new sheets as inputs, seeds the database and displays the inserted data according to the filters defined by the user.

Keywords: Data visualization, Higher education, Technology education.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

MEC	Ministério da Educação
INEP	Institutio Nacional de Estudos e Pesquisas Educacionais
SQL	*Structured Query Language* - Linguagem de Consulta Estruturada
HTTP	*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto
CSV	*Comma-separated values* - Valores Separados por Vírgula
EAD	Educação a Distância
JSON	*JavaScript Object Notation* - Notação de Objecto de JavaScript

SUMÁRIO

1 INTRODUÇÃO

As áreas de *Business Intelligence* (BI) e análise de dados são frequentemente utilizadas para lidar com grandes quantidades de dados coletados em censos e pesquisas. BI é um conjunto de técnicas e ferramentas que têm como objetivo o refinamento de dados e a sua transformação em informação (Inmon e Nesavich 2008). O BI pode ser utilizado para descobrir tendências que podem ser úteis para a tomada de decisões. Enquanto isso, a análise de dados é a área mais ampla que envolve a coleta e o tratamento de dados aliados a técnicas de modelagem de dados, análise estatística, mineiração de dados, aprendizado de máquina, entre outros. Ela é frequentemente utilizada em pesquisas para se obter um esclarecimento sobre algum fenômeno.

A área de visualização de dados se dedica a criar representações visuais de dados e informações. O objetivo da visualização de dados é fornecer uma maneira de comunicar informações para que o leitor tenha maior facilidade na sua interpretação, auxiliando na percepção de tendências e correlações que não são facilmente identificáveis somente através dos dados crus. Outra contribuição da área de visualização de dados é tornar a leitura e a interpretação de dados mais acessíveis para todos os grupos.

No Brasil, a entidade responsável pelo levantamento de dados sobre o ensino superior é o INEP. Estes dados são disponibilizados para o público e todos os anos a Diretoria de Educação da Sociedade Brasileira de Computação, uma sociedade científica sem fins lucrativos com o objetivo fomentar o acesso à informação e cultura por meio da informática, monta um relatório com a visualização dos dados fornecidos pelo INEP. Este relatório é confeccionado de maneira manual, gerando um custo operacional para a Sociedade Brasileira de Computação e resultando numa ferramenta estática de visualização, na qual filtros não podem ser dinamicamente aplicados.

Com o cenário descrito, este trabalho visa o desenvolvimento de uma plataforma que automatiza as consultas de visualização de dados para a geração dos relatórios da Sociedade Brasileira de Computação. O objetivo é, além de reduzir o custo operacional, aumentar a quantidade de diferentes recursos de visualização de dados através do uso de filtros customizáveis.

O restante deste trabalho está organizado da seguinte maneira: o Capítulo 2 aborda os conceitos, as ferramentas e as tecnologias que permitem a criação da ferramenta abordada neste trabalho; o Capítulo 3 explora outros trabalhos relacionados ao tópico estudado e monta uma comparação entre eles e este trabalho; o Capítulo 4 descreve em mais de-

talhes o processo de construção do relatório e a motivação para a construção desta nova ferramenta, junto do levantamento de requisitos; o Capítulo 5 descreve o processo de desenvolvimento e o funcionamento da versão final da ferramenta; o Capitulo 6 analisa os dados que foram fornecidos e os resultado dos dados tratados e propõe uma série de questionamentos cujas respostas podem ser encontradas com o auxílio da ferramenta; e, por fim, o Capítulo 7 revisa as conclusões e contribuições deste trabalho e apresenta sugestões de possíveis trabalhos futuros.

2 CONCEITOS E TECNOLOGIAS UTILIZADAS

Neste capítulo, são apresentados os conceitos e as tecnologias que foram usadas para realizar a normalização dos dados providos pelo MEC e a construção de uma ferramenta para a visualização dos dados.

2.1 Conceitos

Nesta seção, são explorados os principais conceitos importantes para a compreensão deste trabalho.

2.1.1 Sistema de Gerenciamento de Bancos de Dados

Um Sistema de Gerenciamento de Bancos de Dados (SGBD) (Connolly e Begg 2014) é um *software* que permite que o usuário defina, crie, mantenha e controle o acesso à bancos de dados. Por mais que o conjunto de funcionalidades oferecidas por um SGBD possa variar dependendo das especificidades de cada implementação, (Codd 1970) sugere uma série de recursos com os quais um sistema de propósito geral deve contar, como armazenamento, leitura e atulização de dados, apresentação de um catálogo ou dicionário de dados descrevendo os metadados, suporte para transações e concorrência, suporte para autorização de acesso e atualização de dados e imposição de restrições para garantir que os dados do banco de dados seguem certas regras definidas pelo usuário.

2.1.2 Banco de Dados Relacional

Bancos de dados relacionais são bancos de dados baseados no modelo de dados relacional proposto por (Codd 1970). O modelo relacional organiza dados em uma ou mais tabelas (ou "relações") com linhas e colunas com uma chave única identificando cada linha. Linhas também são referidas como tuplas, enquanto colunas são também chamadas de atributos. De maneira geral, linhas representam uma instância de dados sendo armazenados e as colunas os atributos dessa instância específica.

A tabela 2.1 exemplifica o conceito descrito. Nela, cada linha representa uma

Tabela 2.1 – Exemplo de tabela "alunos"de um banco de dados relacional

ID	**Nome**	**Número de matrícula**
1	John Doe	00274697
2	Jane Doe	00274698
3	John Smith	00274699

instância de um aluno, com cada coluna representando um atributo. Se tomarmos como exemplo a linha 1, vemos os atributos de um aluno chamado John Doe a quem foram atributos o identificador único 1 e o número de matrícula 00274697.

2.1.3 Sistema de Gerenciamento de Bancos de Dados Relacionais

Um Sistema de Gerenciamento de Bancos de Dados Relacionais (SGBDR) é um SGBD que atua sobre um banco de dados relacional. Um SGBDR provê uma interface entre aplicações e usuários e o banco de dados. É também esperado que ele possa definir tabelas, atualizar suas definições e performar operações sobre as tabelas, como consular, inserir e atualizar dados.

2.1.4 Normalização de dados

Os dados de um banco de dados são ditos normalizados quando se segue um conjunto de regras que visam reduzir a redundância e aumentar a consistência de dados, evitando anomalias que podem ser introduzidas na inserção, atualização ou deleção de elementos em bases não normalizadas (Codd 1970).

Foram propostos diversos modelos de formas normais de bancos de dados, sendo elas descritas nas seguintes seções. As formas normais foram propostas na ordem na qual foram apresentadas e são incrementais. Em outras palavras, se um banco de dados está em uma forma normal, sabemos que ele também está nas formais normais concebidas anteriormente.

2.1.4.1 Primeira forma normal

Diz-se que um banco de dados está na primeira forma normal (Codd 1970) quando as suas tabelas respeitam as seguintes duas regras:

1. Toda coluna da tabela contém somente valores atômicos, com um valor atômico

sendo um valor que não pode ser dividido;

2. As tabelas não têm grupos de repetição. Em outras palavras, não há diferentes colunas numa mesma tabela representando o mesmo tipo de dados.

As tabelas 2.2 e 2.3 violam as regras 1 e 2, respectivamente. No primeiro caso, a coluna "Cidade"é multivalorada, com a segunda linha tenho um valor não atômico. O segundo caso conta com duas colunas representando a mesma informação: as colunas Cidade 1 e Cidade 2.

Tabela 2.2 – Exemplo de tabela "clientes"com valores não atômicos

ID	**Nome**	**Cidade**
1	John Doe	Porto Alegre
2	Jane Doe	Porto Alegre, São Leopoldo
3	John Smith	Novo Hamburgo

Tabela 2.3 – Exemplo de tabela "clientes"com grupos de repetição

ID	**Nome**	**Cidade 1**	**Cidade 2**
1	John Doe	Porto Alegre	
2	Jane Doe	Porto Alegre	São Leopoldo
3	John Smith	Novo Hamburgo	

Para normalizar este banco de dados, devemos separar as tabelas de clientes e de endereços, conforma demonstrado nas tabelas 2.4 e 2.5. Desta forma, não há a necessidade de termos atributos multivalorados, pois podemos utilizar a adição de novas linhas na tabela de endereços para armazenarmos mais valores e também não temos nenhum conjunto de colunas representando o mesmo tipo de informação.

Tabela 2.4 – Exemplo de tabela "clientes"na primeira forma normal

ID	**Nome**
1	John Doe
2	Jane Doe
3	John Smith

Tabela 2.5 – Exemplo de tabela "endereços"na primeira forma normal

ID	**ID do Cliente**	**Cidade**
1	1	Porto Alegre
2	2	Porto Alegre
2	2	São Leopoldo
3	3	Novo Hamburgo

2.1.4.2 Segunda forma normal

Um banco de dados está na segunda formal (Codd 1971) normal quando está na primeira forma normal e, em todas as suas tabelas, se pode observar que não existe uma dependência parcial entre atributos primários e não primários. Nesta definição, atributos primários são atributos que compõe uma chave candidata, ou seja, um conjunto de atributos mínimo que compõe uma chave que identifica cada linha unicamente. Os atributos que não se encaixam nesta definição são chamados não-primários.

Tabela 2.6 – Exemplo de tabela "compras"com dependências parciais

Número da compra	**ID do produto**	**Nome do produto**	**Valor**
1	55	Impressora	R$150,00
2	97	Teclado	R$50,00
3	33	Monitor	R$1500,00

A tabela 2.6 tem como chaves candidatas a composição entre o número da compra e o ID do produto e o número de compra e o nome do produto. Desta forma, o único atributo não primário é o valor. Existe uma dependência parcial entre o valor da compra e as duas chaves candidatas dado que ele depende ou do ID do produto ou do nome do produto, mas nunca do número da compra. Sendo assim, este valor deve ser extraído para uma nova tabela, como demonstrado pelas tabelas 2.7 e 2.8.

Tabela 2.7 – Exemplo de tabela "produtos"na segunda forma normal

ID	**Nome do produto**	**Valor**
55	Impressora	R$150,00
97	Teclado	R$50,00
33	Monitor	R$1500,00

Tabela 2.8 – Exemplo de tabela "compras"na segunda forma normal

Número da compra	**ID do produto**
1	55
2	97
3	33

Na tabela de produtos temos duas chaves candidatas e as duas são compostas por um único atributo: ID do produto e nome do produto. Assim, o valor depende sempre da chave candidata inteira, sem dependências parciais.

2.1.4.3 Terceira forma normal

Para um banco de dados estar na terceira forma normal (Codd 1971) ele deve estar na segunda e verificar que nenhuma coluna não-primária depende de nenhuma outra coluna não-primária. Para ilustrar a situação, estendamos a tabela normalizada do exemplo anterior para incluir a quantidade de unidades e valor final de cada compra na tabela de compras.

Tabela 2.9 – Exemplo de tabela "compras"com dependência entre colunas não-primárias

Número da compra	**ID do produto**	**Unidades**	**Valor Total**
1	55	2	R$300,00
2	97	3	R$150,00
3	33	1	R$1500,00

Com o banco de dados sendo composto pelas tabelas 2.7 e 2.9, verificamos que a tabela de compras tem somente uma chave candidata composta pelo número da compra e ID do produto. Há uma dependência entre a colunda Unidades e Valor Total dado que o valor total é calculado com base no valor do produto com o ID daquela linha e na quantidade de unidades. Uma vez que Unidades e Valor Total são colunas não-primárias, podemos afirmar que o banco de dados não está na terceira forma normal. Para normalizar o banco de dados devemos remover a coluna Valor Total, conforme 2.10.

Tabela 2.10 – Exemplo de tabela "compras"na terceia forma normal

Número da compra	**ID do produto**	**Unidades**
1	55	2
2	97	3
3	33	1

2.1.4.4 Forma normal de Boyce-Codd

A forma normal de Boyce-Codd (Codd 1974) diz que cada coluna deve ser dependente inteiramente de cada chave candidata, além de requirir que o banco de dados esteja na terceira forma normal. Embora o conceito se pareça com os da terceira e segunda forma normal, ele cobre alguns casos específicos que não são cobertos pela terceira forma normal.

A tabela 2.11 mostra os filmes mais bem classificados pela plataforma *Rotten Tomatoes* por ano. Assumindo que o título de um filme é único, podemos definir as

Tabela 2.11 – Exemplo de tabela "filmes"

Ano de lançamento	**Ranking**	**Título**	**Mês e ano de lançamento**
2015	1	Mad Max: Fury Road	05/2015
2015	2	Inside Out	09/2015
2015	3	Star Wars: The Force Awakens	12/2015
2014	1	Boyhood	07/2014
2014	2	The LEGO Movie	02/2014
2014	3	Nightcrawler	10/2014

seguintes chaves candidatas para esta tabela:

- Título;
- Ano de lançamento, Ranking;
- Mês e ano de lançamento, Ranking.

Todos os quatro atributos aparecem em alguma chave candidata, o que significa que todos eles são atributos primários e, portanto, temos a terceira forma normal. Contudo, vemos que não são todos os atributos que dependem de todas as chaves candidatas. O atributo Mês e ano de lançamento não depende de toda a chave candidata Ano de lançamento, Ranking pois não há dependência no Ano de lançamento. Isso abre espaço para anomalias como uma linha tendo o valor 2015 como ano de lançamento e 10-2014 como Mês e ano de lançamento, com dois valores diferentes para o ano.

Tabela 2.12 – Exemplo de tabela "filmes"na forma normal de Boyce-Codd

Ano de lançamento	**Ranking**	**Título**	**Mês de lançamento**
2015	1	Mad Max: Fury Road	05
2015	2	Inside Out	09
2015	3	Star Wars: The Force Awakens	12
2014	1	Boyhood	07
2014	2	The LEGO Movie	02
2014	3	Nightcrawler	10

A tabela 2.12 coloca a tabela na forma normal de Boyce-Codd ao transformar a coluna Mês e ano de lançamento em Mês de lançamento. Desta forma a última coluna deixa de ser um atributo primário e podemos observar que ele depende inteiramente das outras chaves candidatas.

2.2 Tecnologias

Nesta seção são apresentadas as tecnologias utilizadas na execução do projeto.

2.2.1 PostgreSQL

PostgreSQL (Group 2023) é um SGBDR *open source* com suporte a SQL. A sua distribuição e o seu uso são completamente gratuitos. PostgreSQL é utilizado neste trabalho para armazenar de maneira estruturadas os dados providos pelo MEC e para servir como fonte de dados nas etapas de visualização.

Esta ferramenta foi escolhida por ser gratuita e robusta graças a anos de uso e manutenção de uma comunidade forte. Entre os Sistemas de Gerenciamento de Bancos de Dados Relacionais com características semelhantes, este é o com o qual o autor tem a maior similaridade.

2.2.2 Elixir

Elixir (Elixir 2023) é uma linguagem funcional de propósito geral baseada no modelo de atores (Hewitt 2010) e com ênfase em computação concorrente e distribuída. O código é compilado para o *byte code* da BEAM, a máquina virtual do Erlang (Erlang 2023).

A linguagem foi escolhida principalmente pelo fato de que, entre as linguagens de propósito geral, esta é a liguagem com a qual o autor tem a maior fluência, aumentando a produtividade no desenvolvimento da ferramenta. Além disso, como mencionado, a ênfase em computação paralela e distribuída acaba gerando um conjunto de recursos que tornam confortável o desenvolvimento de sistemas *web* robustos.

2.2.3 Phoenix

Phoenix (Framework 2023) é um *framework* escrito em Elixir para a implementação de aplicações *web*. Ele provê recursos para manejar requisições HTTP, fazer uso de *websockets* e para fazer o uso de bancos de dados.

Este *framework* foi escolhido por ser o mais completo e amplamente adotado deste

tipo, por ser *open source*, ser mais utilizado significa ser também mais robusto.

2.2.4 LiveView

O LiveView (Framework 2023) é uma extensão do Phoenix para a implementação de páginas com atualizações em tempo real. Com ele, toda vez que uma página é renderizada, é mantido um *websocket* aberto e um processo no servidor mantendo o estado deste cliente. A ferramenta contém mecanismos eficientes de cálculos de diferença e é capaz de mandar modificações do conteúdo renderizado para o cliente sem ter que reenviar a página inteira. Além disso, mantém aberto um canal birecional de fluxo de dados.

Esta biblioteca foi utilizada no projeto porque ela facilita a atualização dos dados na tela de maneira assíncrona. Tendo em vista a atualização dinâmica dos filtros que entraram nos requisitos do trabalho, foi compreendido que o LiveView seria uma boa adição.

2.2.5 ExUnit

O ExUnit (ExUnit 2023) é uma biblioteca para a criação de testes automatizados para programas escritos em Elixir. Além das asserções ajudarem a verificar o correto funcionamento dos módulos escritos, as descrições por escrito dos casos de teste ajudam a documentar o comportamento esperado. Contando com mensagens significativas em caso de falhas nas asserções, a biblioteca já vem no *core* da linguagem Elixir, o que significa que nenhuma ação extra foi necessária para contar com ela.

2.2.6 Chart.js

O *Chart.js* (ChartJS 2023) é uma biblioteca escrita em Javascript que tem como finalidade a construção de ferramentas de visualização de dados. A ferramenta conta com recursos para a exibição de diversos tipos de gráficos e fornece diferentes formas de interagir com tais gráficos.

A escolha desta biblioteca se deu por, depois de pesquisas sobre bibliotecas semelhantes, se concluir que ela atendia a todos os requisitos necessários e, dado o conjunto de conhecimentos do autor, a sua configuração seria a mais fácil.

2.3 Considerações Finais

Neste capítulo foram apresentados os conceitos e as ferramentas que foram usados no desenvolvimento deste trabalho para a implementação da aplicação. São usadas as ferramentas PostgreSQL, para armazenamento e leitura de dados; *Phoenix*, para a implementação da parte *web* da aplicação; e *Chart.js*, para a exibição dos dados.

3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados alguns trabalhos relacionados e discutidos semelhanças e diferenças com este projeto.

3.1 Descrição dos trabalhos

O primeiro trabalho escolhido, intitulado *Ferramenta de Visualização de Dados do Censo da Educação Superior do INEP* (Borchardt et al. 2022) iniciou a construção de uma ferramenta para a visualização dos dados fornecidos pelo INEP. Assim como o presente trabalho, o resultado final é uma ferramenta *web*. Outras semelhanças são o uso de gráficos para a visualização de dados e a definição dinâmica dos parâmetros. Contudo, o projeto descrito no artigo analisa dados difentes, como o gênero dos alunos e o tamanho do corpo docente. Também pode-se evidenciar que não existe uma idea de automatização do processamento de novos dados, com o conjunto de dados disponível sendo somente o censo de 2019. Por fim, a ferramenta não conta com instrumentos para segmentar as consultas por região do país. A ferramenta aqui desenvolvida, por outro lado, tem consultas no entorno do ensino remoto e regiões do país, além de contar com todos os recursos necessários para a adição de novos dados para que se considere os censos que estão por vir.

O texto com o título *Mineração em bases de dados do INEP: uma análise exploratória para nortear melhorias no sistema educacional brasileiro* (Fonseca e Namen 2016) propõe o uso da base de dados disponibilizada pelo INEP para analisar quais são os fatores do campo docente que intereferem no desempenho do corpo discente no aprendizado de matemática no Rio de Janeiro. Nele, o algoritmo *Naive Bayes* é usado sobre diferentes atributos depois de determinar quais professores tiveram influências positivas ou negativas sobre os alunos. A partir destas determinações, o artigo se propõe a discutir possíveis melhorias no sistema de ensino. Da mesma forma que o artigo descrito, o projeto aqui descrito também visa a análise dos dados fornecidos pelo INEP. Contudo, lá se busca fazer a análise enquanto aqui se busca facilitar a visualização de dados para que o leitor faça as análises. Também vale destacar que o escopo dos dados são diferentes, com um focando no ensino de uma matéria da educação básica em um único estado enquanto o outro foca no ensino superior em todo o país.

A dissertação de mestrado com o título *A construção de um data warehouse uti-*

lizando os indicadores educacionais do INEP (Ilha 2021) descreve o uso das planilhas fornecidas pelo INEP e da ferramenta *Microsoft Power BI* para a visualização de dados referentes ao ensino fundamental no município de Santa Maria. Houve também conversas com as administrações das escolas para melhor entender como funciona o uso de dados no meio educacional e como é a comunicação entre o INEP e essas instituições para o levantamento de dados. Mais uma vez a semelhança entre a dissertação e o presente texto é o uso dos dados dos censos do INEP. Contudo, enquanto o primeiro foca no ensino fundamental de um município e engloba também a compreensão do contexto de uso e levantamento de dados, o presente tem outro foco de dados e tem como público alvo a Sociedade Brasileira de Computação.

O artigo intitulado *Dados Abertos Educacionais Brasileiros: Um Mapeamento Sistemático da Literatura* (12) faz uma análise da importância e do uso de dados relativos a educação no cenário de pesquisa brasileiro. O projeto define uma série de critérios que qualificam trabalhos para ser parte da análise e estudam quais são os principais usos que eles fazem das bases de dados, como quais são as ferramentas mais utilizadas, os algoritmos mais utilizados para *data mining* e a evolução da produção de artigos deste tipo ao longo dos anos. Vê-se que os algoritmos mais usados são o J48, normalmente acompanhado do uso da plataforma Weka, e do Naive Bayes. O artigo também destaca que, representando quase dois terços do total de artigos analisados, a maioria das pesquisas trabalha com dados no âmbito nacional e não a nível local.

O trabalho nomeado *Modelagem e Visualização Científica de Dados Educacionais: Estudo de Caso sobre o Desempenho em Componentes Curriculares* (1) propõe o uso de técnicas de *data mining* junto das de visualização de dados. Ao longo do texto, diferentes técnicas são empregadas em conjunto numa sequência determinada pelas conclusões tiradas do passo anterior. Ao contrário da ferramenta aqui desenvolvida, o artigo busca trabalhar os dados em vez de simplesmente auxiliar na visualização. Além disso, ele não trabalha com dados a nível nacional: os dados utilizados são referentes ao ensino de Licenciatura em Espanhol e Segurança do Trabalho no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

O artigo chamado *Determinantes da repetência escolar no Brasil: uma análise de painel dos censos escolares entre 2007 e 2010* (Oliveira e Soares 2012) busca entender um dos fenômenos que, de acordo com estudos mencionados no próprio artigo, é central no tema de evasão escolar no Brasil: a repetência. Assim como a ferramenta aqui desenvolvida, eles fazem uso dos dados gerados pelos censos do INEP. Contudo, os auto-

res tinham outro objecto de estudo: a educação básica ao contrátio da educação superior. Eles também buscam analisar os dados em vez de criar formas de visualização que serão posteriormente usadas em análises.

3.2 Análise Comparativa

Neste capítulo foram trazidos trabalhos que, de alguma forma, fazem uso dos dados dos censos do INEP. Eles variam no uso dos dados e no objectivo final, indo de extração de dados a visualização. Nenhum deles, contudo, visa a automatização do processamento e exibição de dados como o presente trabalho. A tabela 3.1 ilustra a comparação entre os trabalhos relacionados e a ferramenta desenvolvida para a SBC.

Tabela 3.1 – Tabela comparativa entre trabalhos relacionados

Título	**Escopo dos dados**	**Produz visualização de dados**	**Produz mineração de dados**	**Possibilidade de adição de dados**
Ferramenta de Visualização de Dados do Censo da Educação Superior do INEP	Nacional, Ensino Superior	Sim	Não	Não
Mineração em bases de dados do INEP: uma análise exploratória para nortear melhorias no sistema educacional brasileiro	Regional, Ensino Básico	Não	Sim	Não
A construção de um data warehouse utilizando os indicadores educacionais do INEP	Regional, Ensino Básico	Sim	Não	Não
Dados Abertos Educacionais Brasileiros: Um Mapeamento Sistemático da Literatura	Nacional, Ensino Básico e Superior	Não	Sim	Não
Modelagem e Visualização Científica de Dados Educacionais: Estudo de Caso sobre o Desempenho em Componentes Curriculares	Regional, Ensino Superior	Sim	Sim	Não
Determinantes da repetência escolar no Brasil: uma análise de painel dos censos escolares entre 2007 e 2010	Nacional, Ensino Básico	Não	Sim	Não
Uma ferramenta web para a automatização de relatórios da Sociedade Brasileira de Computação sobre dados referentes ao ensino nacional de tecnologia	Nacional, Ensino Superior	Sim	Não	Sim

4 CONTEXTUALIZAÇÃO E LEVANTAMENTO DE REQUISITOS

Este capítulo apresenta o contexto e a motivação para a implementação do projeto, bem como as entidades envolvidas e os processos existentes.

4.1 Entidades envolvidas

A primeira entidade envolvida na viabilização desta ferramenta é o Ministério da Educação (MEC). Como um ministério da República Federativa do Brasil, suas competências abrangem políticas nacionais de todos os níveis (GovBR 2023). Um dos braços do MEC é o Instituto Nacional de Estudos e Pesquisas Educacionais (INEP), que visa planejar, desenvolver, implementar e organizar sistemas de avaliação, estatísticas, testes de desempenho, pesquisas quantitativas e qualitativas ou qualquer outra metodologia necessária à produção e à disseminação de informações sobre os sistemas educacionais (GovBR 2023). Dentro dessa descrição, é particularmente interessante destacar a realização do Censo Escolar da Educação Básica e o Censo da Educação Superior, que serve como fonte de dados para o funcionamento desta ferramenta. Os dados resultantes dos censos são disponibilizados de maneira aberta para todos os públicos no formato de arquivos CSV e podem ser encontrados on site do INEP (INEP 2023).

Outra entidade envolvida na criação desta ferramenta é a Sociedade Brasileira de Computação (SBC), uma sociedade científica sem fins lucrativos composta por estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o país. O seu objetivo fomentar o acesso à informação e cultura por meio da informática, promover a inclusão digital, incentivar a pesquisa e o ensino em computação no Brasil e contribuir para a formação do profissional da computação com responsabilidade social (SBC 2023). A SBC conta com uma diretoria específica para os assuntos relacionados à educação. À diretoria de educação compete presidir a Comissão de Educação, bem como supervisionar a realização de eventos relativos à discussão de assuntos ligados ao ensino de computação e ao exercício da profissão (SBC 2023). A diretoria é, no momento, dirigida pela Profa. Itana Maria de Souza Gimenes.

A Diretoria de Educação da SBC elabora anualmente um Relatório da Educação Superior em Computação a partir da base de dados fornecida pelo INEP. Estes relatórios são gerados manualmente e disponibilizados em formato PDF (SBC 2023). Seu conteúdo faz comparações com os dados de anos anteriores. O fato de que estes relatórios são

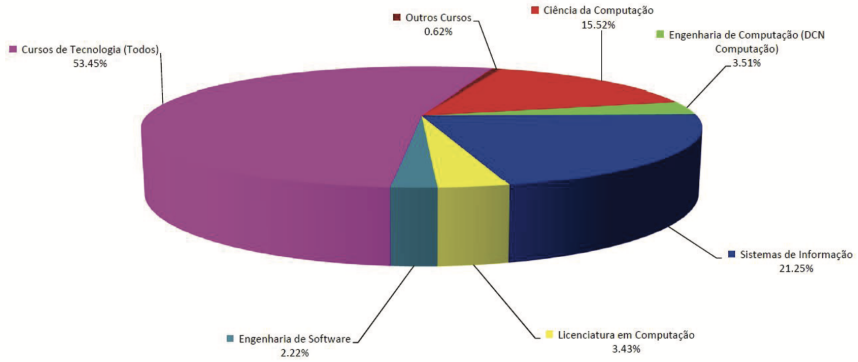
desenvolvidos de forma manual gera um alto custo operacional na sua produção. As figuras 4.1 e 4.2 trazem exemplos de formas de visualização de dados no relatório do ano de 2019.

Figura 4.1 – Gráfico e tabela de evolução dos cursos de tecnologia do relatório do ano de 2019

Distribuição dos Cursos

Ano

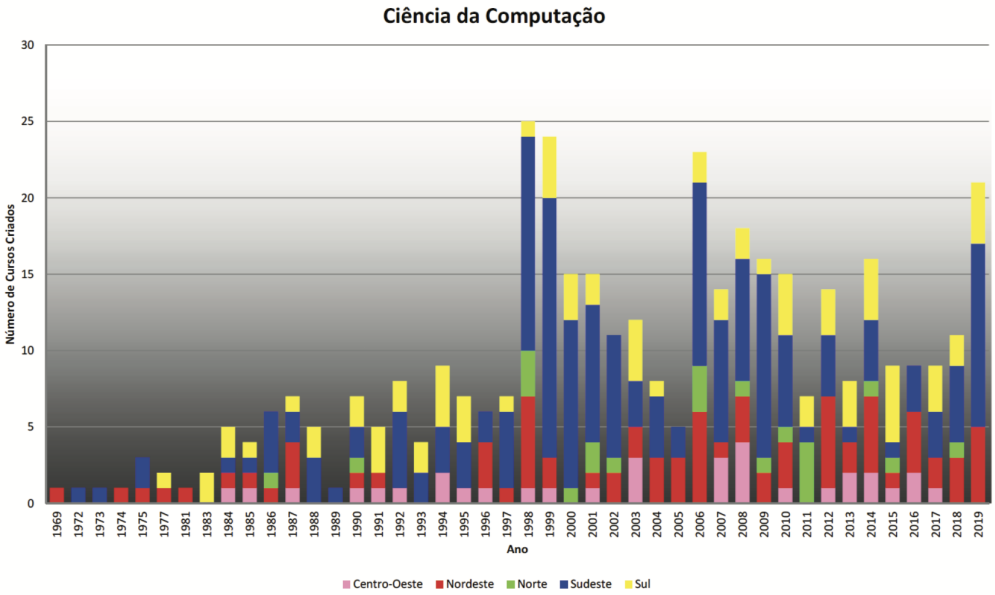
Modalidade de Cursos	2018	2019	Evolução (%)	Panorama 2019 (%)
Ciência da Computação	379	398	5.01	15.52
Engenharia de Computação (DCN Computação)	85	90	5.88	3.51
Engenharia de Software	39	57	46.15	2.22
Sistemas de Informação	562	545	-3.02	21.25
Licenciatura em Computação	85	88	3.53	3.43
Cursos de Tecnologia (Todos)	1282	1371	6.94	53.45
Outros Cursos	16	16	0.00	0.62
Total	2448	2565	4.78	100.00



4.2 Projeto e levantamento de requisitos

Conforme mencionado, o processo descrito tem um alto custo operacional, pois requer uma agregação manual de dados todos os anos. A solução atual também conta com a falta de granularidade nas análises que ficam disponíveis. Os dados levam em consideração uma lista pré-determinada de cursos em todas as instituições de ensino do país e, portanto, não é possível analisar a evolução nos números de cada curso separadamente e nem considerar outros aspectos políticos e sociais das suas aplicações por não podermos segmentar as regiões geográficas do país.

Figura 4.2 – Gráfico da evolução do curso de Ciência da Computação em diferentes regiões do país do relatório do ano de 2019



Tendo em mente o processo existente e os problemas citados, surgiu como proposta a elaboração de uma ferramenta que automatizasse tal processo e trouxesse dinamicidade para as consultas. Para a criação da ferramenta, foram levantados os seguintes requisitos funcionais:

- Visualização da quantidade de alunos matriculados em cursos de tecnologia no país ao longo dos anos. A informação deve ser apresentada em forma de tabela, com os valores escritos por extenso, e em forma de gráfico;
- Visualização da quantidade de alunos matriculados na modalidade EAD em cursos de tecnologia no país ao longo dos anos. A informação deve ser apresentada da mesma forma que a descrita no requisito anterior;
- Para cada uma das consultas acima descritas, deve-se poder selecionar quais são os cursos que devem fazer parte das consultas, assim como quais regiões do país e qual a faixa de anos que as consultas devem considerar;
- Deve ser possível fazer o upload de novos dados visando a realização de novos censos no futuro que devem ser incluídos na base de dados atual.

Os requisitos não funcionais são expressos na seguinte lista:

- Tempo de resposta abaixo de um segundo para novas visualizações;
- Novos usuários devem ser capazes de ver a ferramenta funcionando com pouco esforço e em um pequeno período de tempo;

- Felixibilidade para que cada usuário possa escolher os seus filtros individualmente.

5 IMPLEMENTAÇÃO

Este capítulo apresenta os detalhes e o processo de implementação da ferramenta, como a estruturação do banco de dados, o tratamento dos dados fornecidos pelo MEC, a construção das consultas e a validação dos requisitos.

5.1 Planejamento e desenvolvimento

O planejamento começou com conversas com a Diretoria de Educação da SBC. Nessas conversas buscamos compreender as demandas mais urgentes para podermos definir um escopo que fosse realista em termos de quantidade e tempo de trabalho. O resultado foram os requisitos funcionais e não funcionais descritos na seção 4.2.

Para a implementação, a ordem foi a seguinte:

1. Modelagem de dados;
2. Processamento das planilhas para a inserção de dados;
3. Implementação da visualização da quantidade total de alunos sem filtros;
4. Adição dos filtros;
5. Implementação da visualização da quantidade total de alunos na modalidade EAD, já com filtros;
6. Implementação da página para o *upload* de planilhas.

A partir da terceira etapa houve comunicação regular com a Diretoria de Educação para a validação dos requisitos. Quando necessário, ajustes eram feitos. Visando o rastreamento das mudanças feitas e armazenamento do código, o conteúdo produzido foi sempre versionado com *git* e enviado para a plataforma *GitHub*.

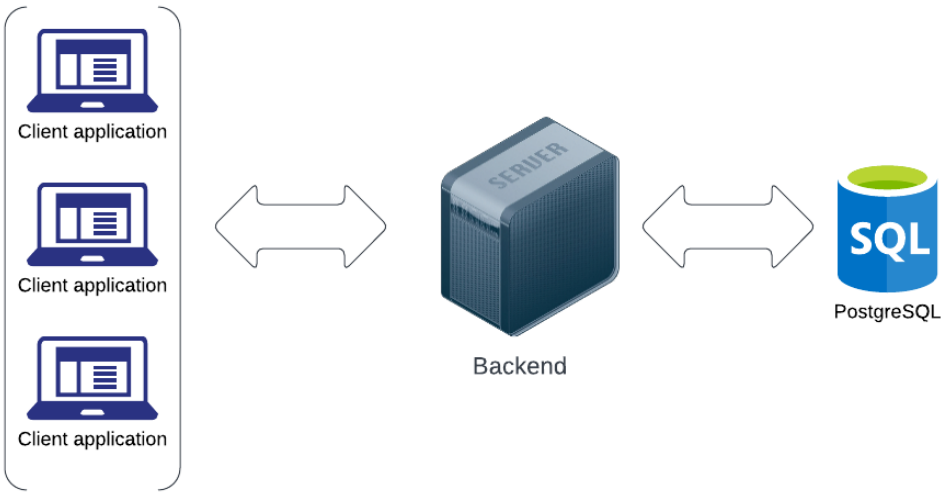
5.2 Arquitetura

Para a codificação, foram utilizadas as linguagens de programação de propósito geral Elixir e JavaScript. Visto que a proposta é construir uma ferramenta *web*, foi utilizado o *framework* Phoenix, que serve para construir aplicações *web* com Elixir. Foi adicionada também a biblioteca LiveView para melhorar experiência e facilitar o desenvolvimento de operações assíncronas, como a aplicação de filtros depois que a página já

foi carregada. Para construir a parte que roda nos clientes, JavaScript foi utilizado com a biblioteca ChartJS, que tem a função de auxiliar na exibição de gráficos. O banco de dados utilizado foi o PostgreSQL.

Sendo uma aplicação *web*, a ferramenta se baseia em uma arquitetura cliente servidor, como expresso na figura 5.1. É desta mesma maneira que o *backend* e o banco de dados se comunicam, mas dessa vez o *backend* tem o papel de cliente em vez de servidor.

Figura 5.1 – Arquitetura da derramenta



O *backend* implementa uma variação do padrão *Model View Controller* (Burbeck 1987). No começo, a implementação é bem padrão: as requisições HTTP chegam, são resolvidas pela *controller* com o uso da *model* e o resultado, além dos possíveis efeitos colaterais como alterações no banco de dados, é uma *view*. Como mencionado no capítulo 2.2.4, o LiveView mantém um *websocket* aberto o tempo todo para que haja uma dinamicidade maior na página. Sendo assim, a *controller* neste caso tem a função extra de tratar eventos gerados pela interação entre o usuário e a *view* e enviados para o servidor.

Quando, por exemplo, a página que exibe a quantidade total de alunos é aberta, a *controller* interage com a *model* e gera uma *view* que é enviada para o cliente. O cliente exibe os dados e permite que o usuário faça modificações nos filtros. Quando os filtros são definidos e o cliente manda os filtros serem aplicados, a ordem é enviada pelo *websocket* e recebida pela *controller*. A *controller*, então, faz uma nova consulta ao banco de dados e envia os dados atualizados para a *view* pelo mesmo *websocket*.

5.3 Dados fornecidos

Para cada ano, o INEP disponiliza um conjunto de arquivos com o resultado do censo. Dentro de cada conjunto, se encontra, além dos dados em si, dicionários de dados explicando o que cada uma das colunas das planilhas significa.

Entre os dados, são dois os CSVs com dados sobre a educação no ensino superior. Uma das planilhas contém informações a respeito das instituições de ensino que participaram do censo daquele ano ennquanto a outra contém informações sobre os cursos ministrados.

5.3.1 Instituições de ensino

A planilha de instituições de ensino conta com 81 colunas com atributos referentes à localização geográfica, corpo docente e instalações das instituições. Destas colunas, as utilizadas pela aplicação são:

- **CO_IES**: número inteiro único por instituição de ensino. O campo não exige tratamento.
- **NO_IES**: nome da instituição de ensino. O campo não exige tratamento.
- **SG_UF_IES**: sigla do estado no qual a instituição de ensino está baseada. O atributo passa por conversões para ser adaptado de texto para tipos internos que representam os estados da República Federativa do Brasil.
- **NO_REGIAO_IES**: nome da região administrativa na qual a instituição de ensino se encontra. O atributo passa por conversões para ser adaptado de texto para tipos internos que representam as regiões do país.

5.3.2 Cursos

A planilha de cursos registra a ocorrência de um curso sendo ministrado em uma instituição de ensino num determinado ano. Com 201 colunas com diferentes atributos sobre os cursos, são disponibilizados dados sobre os alunos e localizações geográficas dos cursos. Destas colunas, as utilizadas pela aplicação são:

- **NO_CINE_ROTULO**: Nome do curso, conforme adaptação da Classificação In-

ternacional Normalizada da Educação Cine/Unesco.

- **CO_CINE_ROTULO**: Código de identificação do curso. Além de permitir que seja identificada a ocorrência de um mesmo curso sendo ministrado em diferentes anos e por múltiplas instutuições de ensino, o código de identificação nos dá informações sobre a área do curso. Isso é especialmente interessante para esse projeto pois sabemos que cursos de tecnologia têm um código com o prefixo *06*.
- **CO_IES**: código que representa uma instituição de ensino, também presente na planilha descrita na seção 5.3.1.
- **NU_ANO_CENSO**: ano no qual o senso foi aplicado.
- **QT_ING**: quantidade de ingressantes em determinado curso para uma instituição de ensino no ano do senso.
- **QT_CONC**: quantidade de concluintes do curso em uma instituição de ensino no ano do senso.
- **QT_INSCRITO_TOTAL_EAD**: quantidade de alunos matriculados no curso em uma instituição de ensino no ano do senso na modalidade de ensino a distância.
- **QT_ING_FEM**: quantidade de ingressantes do sexo feminino em determinado curso para uma instituição de ensino no ano do senso.
- **QT_CONC_FEM**: quantidade de concluintes do sexo feminino do curso em uma instituição de ensino no ano do senso.
- **QT_VG_TOTAL**: Quantidade total de vagas oferecidas.

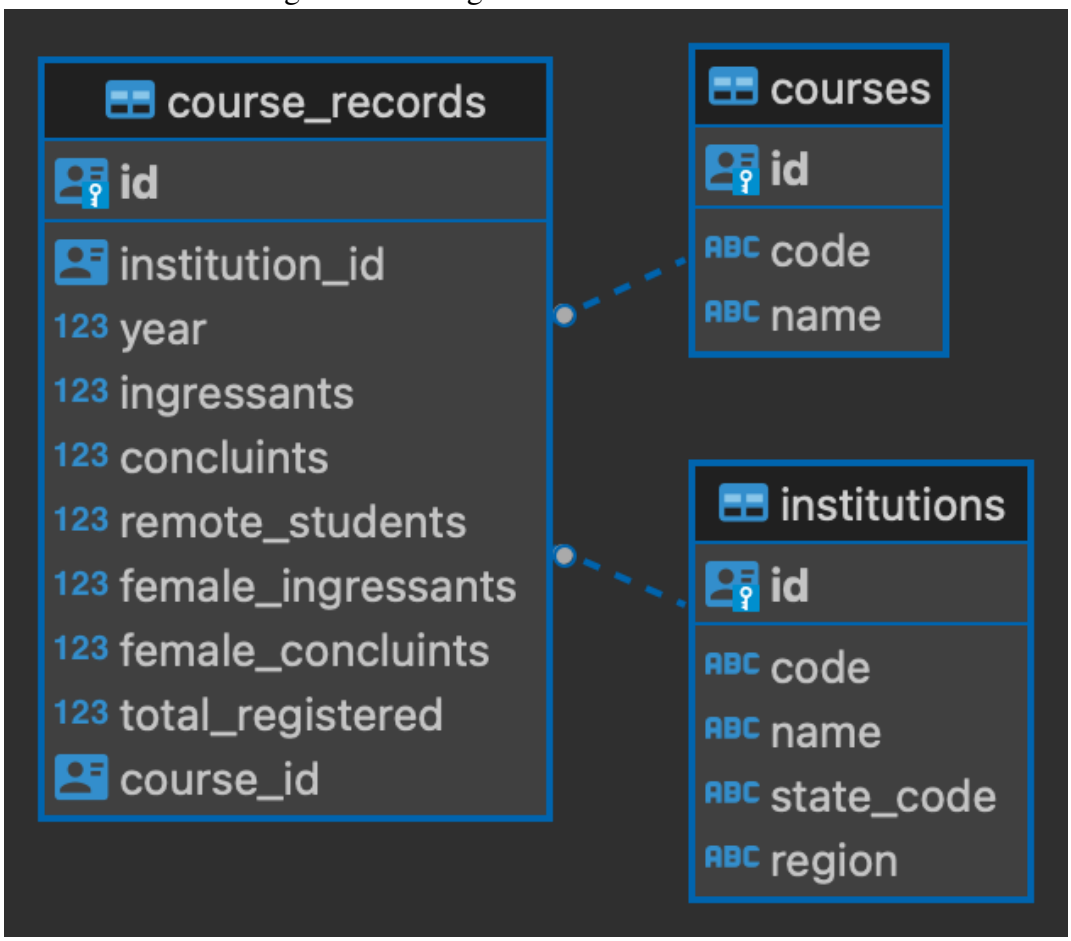
5.4 Modelagem de dados

Para armazenar os dados descritos na seção 5.3, criou-se um *schema* com três tabelas, conforme descrito pela figura 5.2 através de um diagrama Entidade-Relacionamento.

As tabelas têm os seguintes objetivos:

- *institutions*: conta com informações sobre as instituições de ensino que fizeram parte dos censos. Para esta aplicação os valores relevantes são o nome, o código e a região administrativa da instituição.
- *courses*: contém informações a respeito dos cursos ministrados pelo país. Seus valores relevantes são o nome e o código do curso.

Figura 5.2 – Diagrama Entidade-Relacionamento



- *course_records*: registra a ocorrência de um curso sendo ministrado em uma instituição de ensino em um determinado ano. Além de relacionar um curso a uma instituição, contém dados sobre discentes do curso, como quantidades de ingressantes e de concluintes.

Visando a normalização de dados, optou-se por não se fazer um mapeamento direto das planilhas fornecidas para as tabelas. Ao invés disso, a planilha de registro de cursos foi separada em uma tabela para cursos e uma para ocorrências de cursos, evitando a duplicação do código e nome do curso.

Para demonstrar que as tabelas se encontram na forma normal de Boyce-Codd devemos destacar as chaves candidatas:

- *courses*: $\{id\}$, $\{code\}$, $\{name\}$. Cada atributo em separado é uma chave candidata;
- *institutions*: $\{id\}$, $\{code\}$, $\{name\}$;
- *course_records*: $\{id\}$. Não existe outra forma de identificar um registro de curso unicamente. Nas planilhas disponibilizadas pelo INEP encontramos múltiplas linhas com as mesmas combinações de $\{intitution\_id,\ year,\ course\_id\}$.

A tabela *courses* está na primeira forma normal porque não tem atributos multivalorados ou grupos de repetição. A segunda e a terceira formas normais são trivialmente constatadas porque todos os atributos são primários. A forma de Boyce-Codd é visível no fato de que todos os atributos são integralmente dependentes das chaves candidatas.

A tabela *institutions* também não conta com atributos multivalorados ou grupos de repetição. Os atributos não primários, ou seja, *state_code* e *region*, não têm dependências parciais sobre nenhuma das chaves candidatas. Isso é trivial de ser observado visto que nenhuma chave candidata tem valores compostos. Vê-se também que os atributos não primários não têm dependências entre si, colocando a tabela na segunda forma normal. Por fim, todos os atributos dependem inteiramente de todas as chaves candidatas, colocando a tabela na forma normal de Boyce-Codd.

A tabela *course_records*, assim como as duas anteriores, não conta com as características que a tirariam da primeira forma normal. Os seus atributos não primários são todos menos o *id*. A única chave candidata não é composta, o que discarta a possibilidade de existência de dependência parcial, os atributos não primários não dependem uns dos outros e todos depêndem da chave candidata. Sendo assim, as segunda e terceira formas normais podem ser constatadas, assim como a de Boyce-Codd.

5.5 Inserção de dados

A inserção de dados para um ano de censo ocorre em três etapas:

1. Preenchimento da tabela *institutions*. Consiste em fazer um mapeamento direto entre os atributos da planilha de instituições de ensino para a tabela. A única forma de tratamento de dados é fazer a conversão de texto plano para um tipo interno criado para representar as regiões do país. Há um índice de unicidade (https://www.postgresql.org/docs/current/indexes-unique.html) na coluna do código da instituição de forma que duplicações podem ser identificadas. Em caso de conflito na inserção da instituição de ensino a operação é descartada e somente o registro que já estava no banco de dados é mantido. Este mecanismo é útil para que não haja duplicações processando instituições presentes em censos de anos diferentes, ou seja, presentes em mais de um arquivo, ou para podermos proteger o banco de dados contra o processamento repetido do mesmo arquivo.
2. Preenchimento da tabela *courses*. É a primeira de duas vezes que se varre o arquivo de cursos. Neste caso há um mapeamento direto das colunas relevantes no CSV para os atributos equivalentes na tabela. Assim como para instituições de ensino, há um índice de unicidade sobre o código para a identificação de duplicações e registros repetidos são ignorados.
3. Preenchimento da tabela *course_records*. Na segunda vez que se varre a planilha de instituições de ensino há um pouco mais de trabalho a ser feito. Dado que os registros contam com códigos do curso e das instituições de ensino e a tabela conta com chaves estrangeiras, é preciso fazer leituras no banco para mapear códigos para ids. Além disso, é preciso fazer a conversão texto para números inteiros nos atributos numéricos.

Conforme demonstrado na seção 5.7, há uma página para o upload dos arquivos para possibilitar os seus processamentos.

5.6 Consultas de dados

Na definição do escopo do projeto foram escolhidas duas consultas para serem implementadas: a progressão da quantidade total de alunos e da quantidade de alunos na modalidade EAD em cada curso ao longo dos anos. Filtros dinâmicos devem poder ser

definidos pelo usuário e os dados devem ser apresentados adequadamente.

5.6.1 Filtros

Para a implementação dos filtros, foi definido que as regiões administrativas, a faixa de anos e os cursos incluídos nas consultas devem ser customizáveis.

As regiões administrativas são inerentementes fixas e têm as suas definições fora do escopo deste projeto. Portanto, as regiões elegíveis para uso são todas as que fazem parte do tipo definido na aplicação que espelha as regiões existentes fora dela.

A faixa de anos disponíveis depende dos dados com os quais a aplicação conta. Desta forma, são procurados no banco de dados os valores mínimo e máximo para ocorrências de cursos e os valores escolhidos pelo usuário devem estar nesta faixa. Tratar estes valores de forma dinâmica é importante para permitir que mais dados sejam inseridos nos anos por vir depois que mais censos sejam realizados.

A lista de cursos que podem ser incluídos nas consultas é construído de forma dinâmica. Conforme mecionado na seção 5.3.2, cursos de tecnologia têm códigos com o prefixo *06*. Com a crescente relevância de diversas áreas da computação e o consequênte surgimento de cursos com diferentes ênfases, é importante que a aplicação permita que novos cursos sejam lidos das planilhas e tratados na visualização de dados.

5.6.2 Consulta SQL

O algoritmo 5.6.2 demonstra a consulta SQL com os filtros dinâmicos descritos na seção 5.6.1.

```
SELECT c1."name",
       s0."registered"
FROM
  (SELECT ss0."course_id" AS "course_id",
          JSON_OBJECT_AGG(ss0."year", ss0."summed_field"
             ) AS "registered"
   FROM
     (SELECT ssc2."id" AS "course_id",
             ssc0."year" AS "year",
             sum(ssc0."total_registered") AS "
                summed_field"
      FROM "course_records" AS ssc0
      INNER JOIN "institutions" AS ssi1 ON ssi1."id" =
         ssc0."institution_id"
      INNER JOIN "courses" AS ssc2 ON ssc2."id" = ssc0."
         course_id"
      WHERE ((ssc2."code" = ANY($1)
              AND ssc0."year" = ANY($2))
             AND ssi1."region" = ANY($3))
      GROUP BY ssc2."id",
               ssc0."year") AS ss0
   GROUP BY ss0."course_id") AS s0
INNER JOIN "courses" AS c1 ON c1."id" = s0."course_id"
```

A *query* consiste em um anihamento de duas *subqueries* dentro da *query* principal.

A primeira *subquery* opera uma junção entre todas as três tabelas, filtra pelos parâmetros fornecidos, agrupa por ano e id do curso e soma o atributo desejado. No exemplo, é a quantidade total de alunos registrados. Sendo assim, o resultado desta *query* é um conjunto de triplas contendo o id de um curso, o ano do censo e a quantidade total de alunos registrados naquele curso respeitando os filtros.

A segunda *subquery* usa a primeira como fonte de dados. Ela agrupa as linhas retornadas pelo id do curso e executa a fução *JSON_OBJECT_AGG* para construir, para cada curso, um JSON no qual as chaves são os anos e os valores são a quantidade de alunos matriculados neste ano. Os valores selecionados são este JSON e os ids dos cursos.

Por fim, a *query* principal faz uma junção entre o valor retornado pela segunda

subquery e a tabela de cursos. A seleção final é o nome do curso e o JSON que relaciona os aos de censo à quantidade de alunos matriculados.

5.6.3 Exibição de dados

As páginas que exibem os dados são construídas em duas etapas: a montagem da tabela e o preenchimento do gráfico. As páginas foram implementadas usando a biblioteca descrita na seção 2.2.4. Sendo assim, é mantido um *websocket* aberto o tempo todo.

Para a montagem da tabela tudo é feito por esta ferramenta, o que significa que uma primeira versão da página é renderizada com a tabela já pronta e, quando os filtros mudam, são enviadas para o cliente as partes da página que mudaram.

A montagem do gráfico ocorre não através da manipulação de HTML puro, mas sim através da biblioteca *Chart.js*. Ela faz o uso de um elemento HTML do tipo *canva* e usa JavaScript para transformar o *canva* nos gráficos. Como isso não se encaixa no uso da biblioteca que envia as diferenças para o cliente, é preciso dar um passo a mais para gerar um novo gráfico quando os filtros mudam. Para solucionar este problema, se fazem envios de eventos do servidor para o cliente mandando como parâmetros os dados atualizados. Então, um *event listener* captura os dados e os repassa para o *Chart.js* para atualizar os dados.

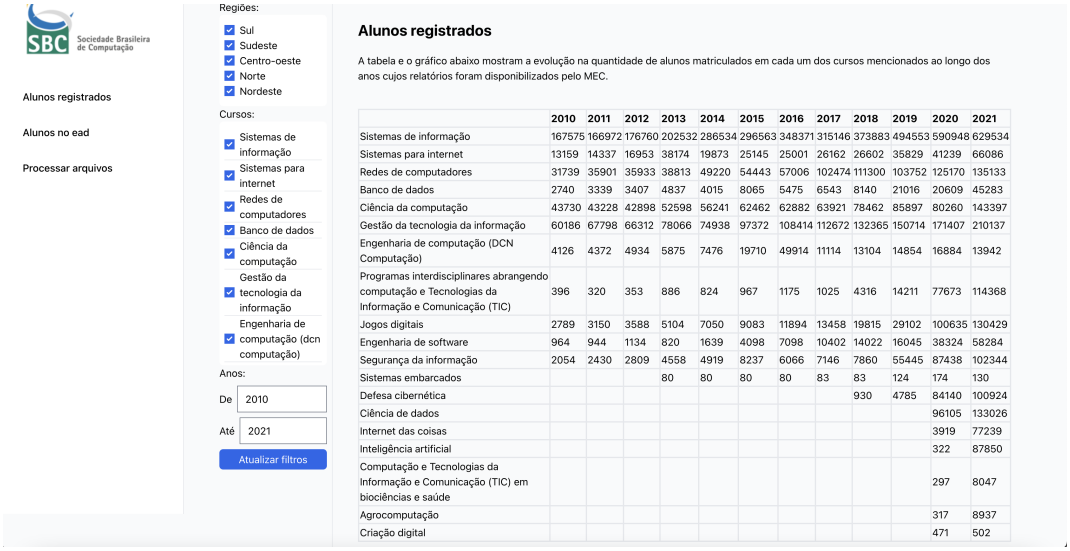
5.7 Páginas da aplicação

A aplicação conta com 3 páginas: uma para a visualização do total de alunos registrados, uma para o total de alunos registrados na modalidade EAD e uma para o *upload* de novos arquivos.

A página para a visualização de dados sobre o total de alunos registrados apresenta uma tabela e um gráfico com as devidas informações. As figuras 5.3 e 5.4 mostram a página. Pode-se também observar, na esquerda da página, os filtros mencionados na seção 5.6.1.

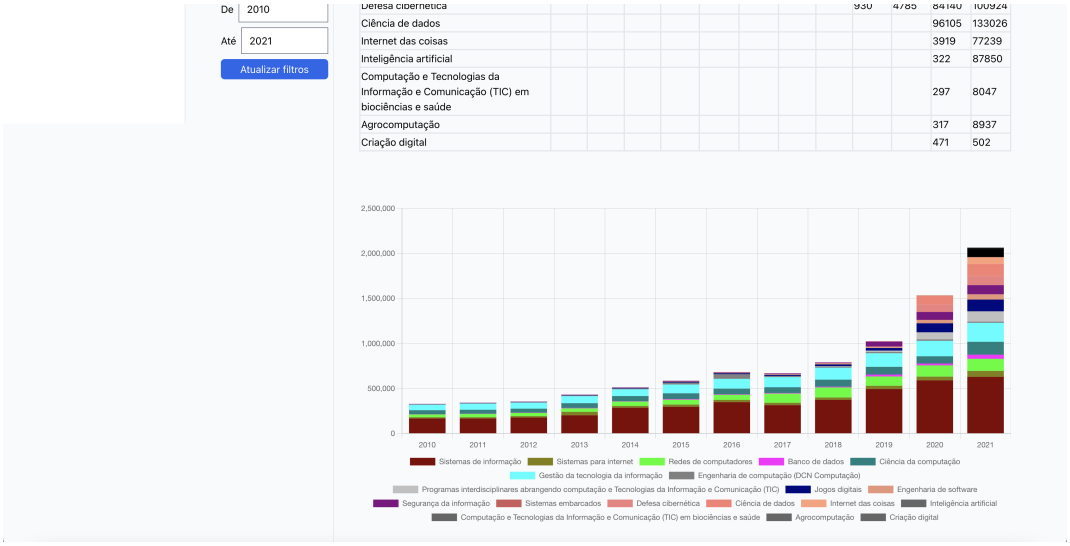
Esta página conta com os mesmos recursos que a página sobre o total de alunos

Figura 5.3 – Screenshot da tabela com dados sobre o total de alunos registrados em cursos de tecnologia



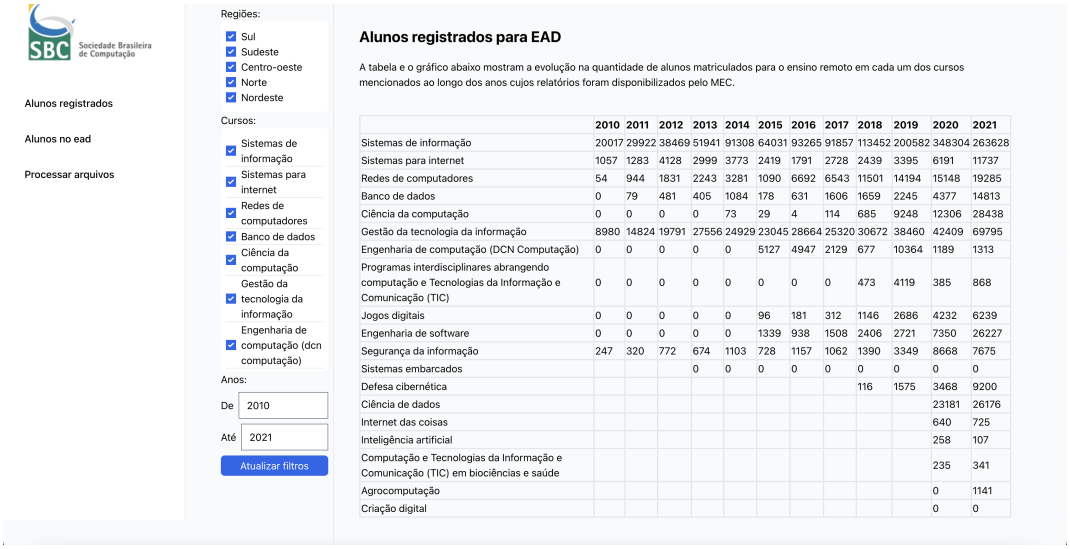
	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Sistemas de informação	167575	166972	176760	202532	286534	296563	348371	315146	373883	494553	590948	629534
Sistemas para internet	13159	14337	16953	38174	19873	25145	25001	26162	26602	35829	41239	66086
Redes de computadores	31739	35901	35933	38813	49220	54443	57006	102474	111300	103752	125170	135133
Banco de dados	2740	3339	3407	4837	4015	8065	5475	6543	8140	21016	20609	45283
Ciência da computação	43730	43228	42898	52598	56241	62462	62882	63921	78462	85897	80260	143397
Gestão da tecnologia da informação	60186	67798	66312	78066	74938	97372	108414	112672	132365	150714	171407	210137
Engenharia de computação (DCN Computação)	4126	4372	4934	5875	7476	19710	49914	11114	13104	14854	16884	13942
Programas interdisciplinares abrangendo computação e Tecnologias da Informação e Comunicação (TIC)	396	320	353	886	824	967	1175	1025	4316	14211	77673	114368
Jogos digitais	2789	3150	3588	5104	7050	9083	11894	13458	19815	29102	100635	130429
Engenharia de software	964	944	1134	820	1639	4098	7098	10402	14022	16045	38324	58284
Segurança da informação	2054	2430	2809	4558	4919	8237	6066	7146	7860	55445	87438	102344
Sistemas embarcados				80	80	80	80	83	83	124	174	130
Defesa cibernética									930	4785	84140	100924
Ciência de dados											96105	133026
Internet das coisas											3919	77239
Inteligência artificial											322	87850
Computação e Tecnologias da Informação e Comunicação (TIC) em biociências e saúde											297	8047
Agrocomputação											317	8937
Criação digital											471	502

Figura 5.4 – Screenshot do gráfico com dados sobre o total de alunos registrados em cursos de tecnologia



registrados. As figuras 5.5 e 5.6 fazem a demonstração.

Figura 5.5 – Screenshot da tabela com dados sobre o total de alunos registrados na modalidade EAD em cursos de tecnologia



	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Sistemas de informação	20017	29922	38469	51941	91308	64031	93265	91857	113452	200582	348304	263628
Sistemas para internet	1057	1283	4128	2999	3773	2419	1791	2728	2439	3395	6191	11737
Redes de computadores	54	944	1831	2243	3281	1090	6692	6543	11501	14194	15148	19285
Banco de dados	0	79	481	405	1084	178	631	1606	1659	2245	4377	14813
Ciência da computação	0	0	0	0	73	29	4	114	685	9248	12306	28438
Gestão da tecnologia da informação	8980	14824	19791	27556	24929	23045	28664	25320	30672	38460	42409	69795
Engenharia de computação (DCN Computação)	0	0	0	0	0	5127	4947	2129	677	10364	1189	1313
Programas interdisciplinares abrangendo computação e Tecnologias da Informação e Comunicação (TIC)	0	0	0	0	0	0	0	0	473	4119	385	868
Jogos digitais	0	0	0	0	0	96	181	312	1146	2686	4232	6239
Engenharia de software	0	0	0	0	0	1339	938	1508	2406	2721	7350	26227
Segurança da informação	247	320	772	674	1103	728	1157	1062	1390	3349	8668	7675
Sistemas embarcados				0	0	0	0	0	0	0	0	0
Defesa cibernética									116	1575	3468	9200
Ciência de dados											23181	26176
Internet das coisas											640	725
Inteligência artificial											258	107
Computação e Tecnologias da Informação e Comunicação (TIC) em biociências e saúde											235	341
Agrocomputação											0	1141
Criação digital											0	0

Figura 5.6 – Screenshot do gráfico com dados sobre o total de alunos registrados na modalidade EAD em cursos de tecnologia

A página responsável pela inserção de dados conta com instruções para o upload de arquivos, como pode ser verificado na figura 5.7. As instruções descrevem que existem dois espaços para fazer o upload de arquivos: um para o arquivo das instituições de ensino e um para o arquivo de cursos. Através desses componentes o usuário pode navegar pelo seu sistema de arquivos para selecionar as novas planilhas, conforme a figura 5.8. Por

fim, uma barra de progresso indica o progresso do carregamento do arquivo. Quando a barra se encontra totalmente verde, o usuário pode clicar on botão *upload* para que o processamento do arquivo inicie, como visto na figura 5.9. A partir disso, o feedback é dado a partir de uma barra no topo da página, como exibido na figura 5.10.

Figura 5.7 – Screenshot das instruções para a adição de novos dados

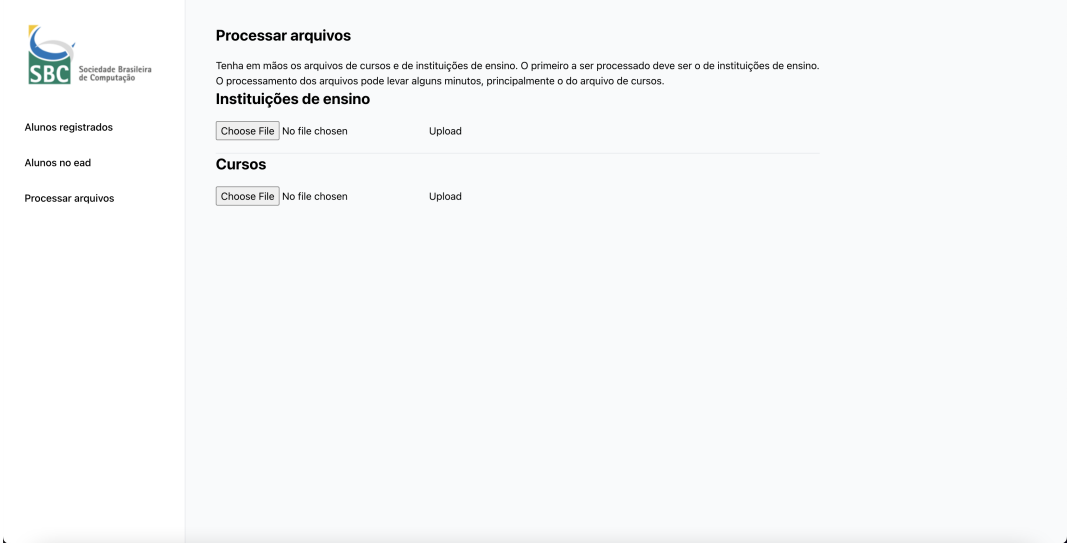


Figura 5.8 – Screenshot seleção do arquivo para a adição de novos dados

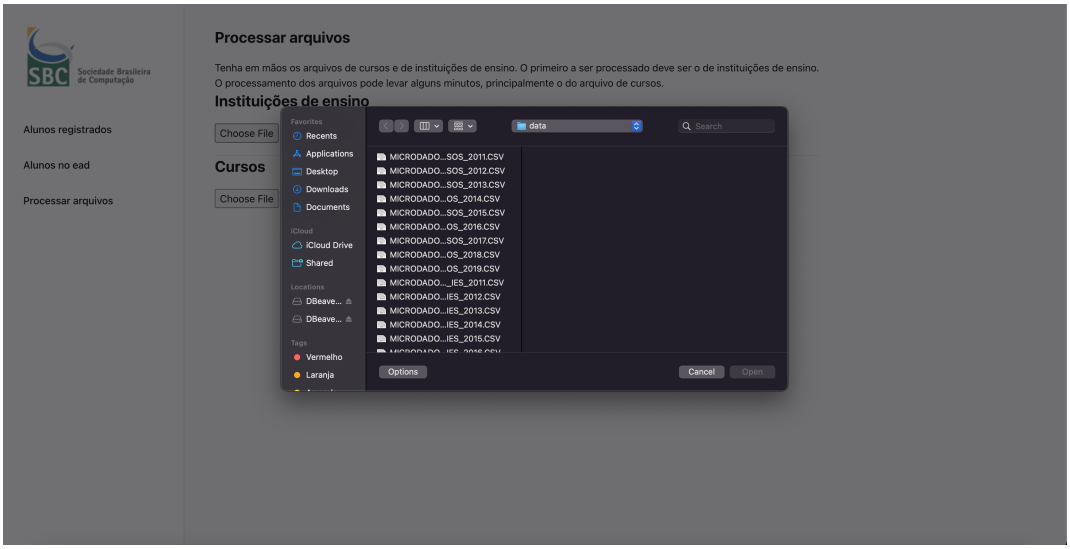


Figura 5.9 – Screenshot do progresso completo de carregamento de um arquivo

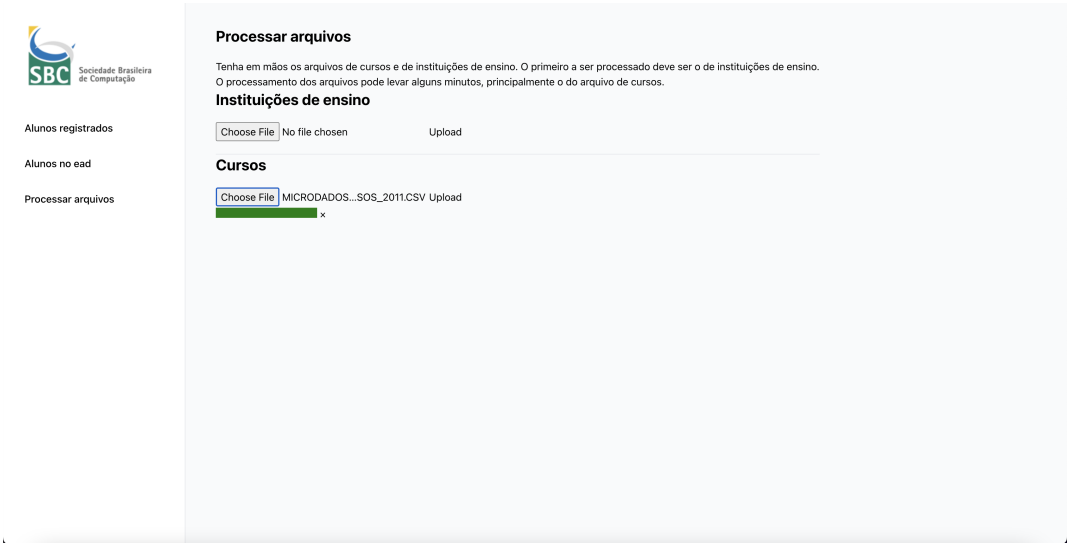
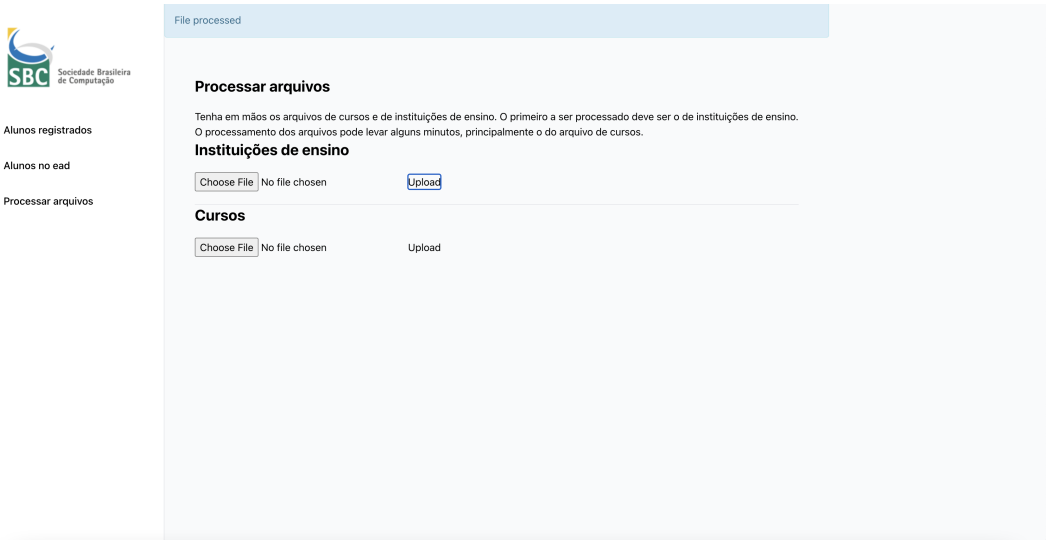


Figura 5.10 – Screenshot da indicação de progresso do processamento de um arquivo



5.8 Testes unitários

Durante todo o desenvolvimento do projeto foram desenvolvidos em paralelo testes unitários para as consultas e os módulos de inserção de dados. Foi utilizada a ferramenta ExUnit. Espera-se, com isso, aumentar a manutenibilidade do projeto, facilitando trabalhos futuros de extensão.

As funções de importação de dados via CSVs fazem o uso de arquivos de amostra com quantidades pequenas de dados para os testes. Os casos de teste usam o banco de dados no modo *sandbox*, o que significa que as inserções não são persistidas e são descartadas uma vez que o teste termina de ser executado.

As funções de consulta têm dados inseridos para criar diferentes cenários e então diferentes asserções são feitas sobre os resultados de diferentes consultas usando diferentes conjuntos de parâmetros.

5.9 Considerações Finais

Neste capítulo foram apresentados o formato dos dados disponibilizados pelo MEC e os diferentes componentes da aplicação, desde a fundação que toma forma antes da inserção de dados até as consultas e visualizações. Também é explicado como se espera que o usuário interaja com a aplicação para a inserção de novos dados, o que será útil conforme o MEC os disponibilize.

6 ANÁLISE DOS DADOS E DEMONSTRAÇÃO

Este capítulo descreve cenários hipotéticos para demonstrar o uso dos filtros descritos na seção 5.6.1. Para tais demonstrações, foram utilizados dados dos anos de 2010 até 2021.

6.1 Descrição dos dados

Conforme descrito na seção 5.3, cada ano conta com duas planilhas, com uma delas descrevendo as instituições de ensino que participaram do censo e a outra com registros dos cursos ministrados. A quantidade de dados é descrita nas tableas 6.1 e 6.2.

Tabela 6.1 – Descrição das planilhas de instituições de ensino

Ano	**Tamanho do arquivo**	**Quantidade de instituições de ensino participantes**
2010	890KB	2378
2011	879KB	2365
2012	897KB	2416
2013	890KB	2391
2014	882KB	2368
2015	887KB	2364
2016	903KB	2407
2017	914KB	2448
2018	951KB	2537
2019	979KB	2608
2020	930KB	2457
2021	1.1MB	2574

Nota-se que, embora não haja havido um aumento ano após ano na quantidade de instituições participantes, houve um crescimento na quantidade de cursos ministrados em todos os anos. É importante destacar que esses dados são sobre todas as áreas de ensino, não somente tecnologia.

Ao processarmos os dados, vemos 3399 instituições de ensino distintas, 365 cursos distintos e 1834809 ocorrências de cursos.

6.2 Experimentos

Esta seção propõe dois experimentos para que possamos usar os filtros.

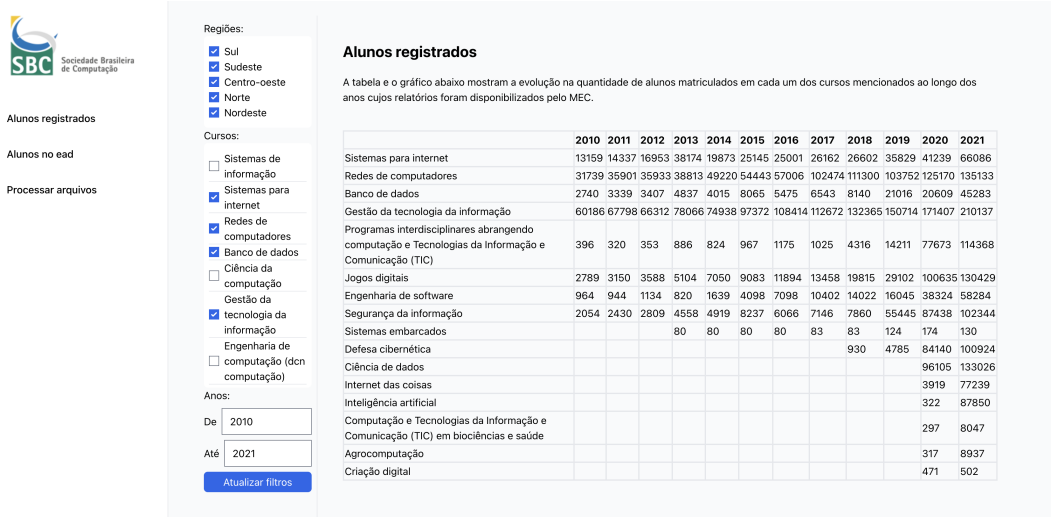
Tabela 6.2 – Descrição das planilhas de cursos ministrados

Ano	Tamanho do arquivo	Quantidade de cursos ministrados
2010	31MB	54194
2011	33MB	57737
2012	37MB	63855
2013	38MB	66126
2014	42MB	73569
2015	46MB	81156
2016	53MB	92866
2017	68MB	119798
2018	102MB	182892
2019	139MB	253139
2020	195MB	344691
2021	251MB	444786

6.2.1 Evolução dos cursos menos frequentes

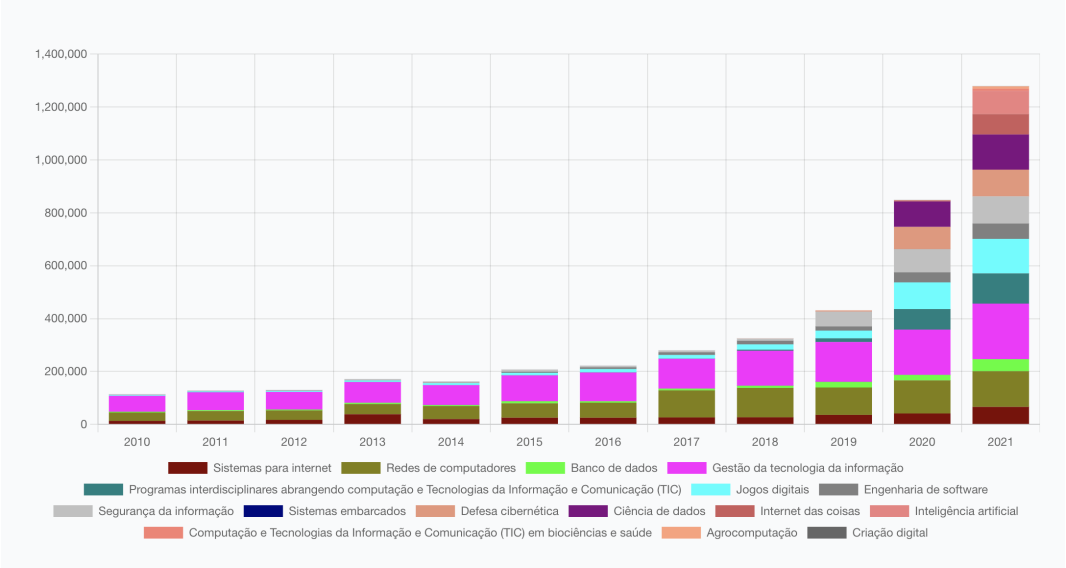
Historicamente, os cursos que apresentam a maior quantidade de alunos matriculados são Sistemas de Informação, Ciência da Computação e Engenharia de Computação. Tal afirmação é suportada pelas figuras 5.3 e 5.4, onde nenhum filtro é aplicado. Visto que as presenças destes cursos crescem bastante nos gráficos, cabe questionar se o aumento da quantidade de alunos nos diferentes cursos vem sendo diversificada ou se há um foco nestes três cursos. Para isso, podemos excluir os três da busca, obtendo a tabela e o gráfico nas figuras 6.1 e 6.2

Figura 6.1 – Consulta com exclusão de cursos - Tabela



	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Sistemas para internet	13159	14337	16953	38174	19873	25145	25001	26162	26602	35829	41239	66086
Redes de computadores	31739	35901	35933	38813	49220	54443	57006	102474	111300	103752	125170	135133
Banco de dados	2740	3339	3407	4837	4015	8065	5475	6543	8140	21016	20609	45283
Gestão da tecnologia da informação	60186	67798	66312	78066	74938	97372	108414	112672	132365	150714	171407	210137
Programas interdisciplinares abrangendo computação e Tecnologias da Informação e Comunicação (TIC)	396	320	353	886	824	967	1175	1025	4316	14211	77673	114368
Jogos digitais	2789	3150	3588	5104	7050	9083	11894	13458	19815	29102	100635	130429
Engenharia de software	964	944	1134	820	1639	4098	7098	10402	14022	16045	38324	58284
Segurança da informação	2054	2430	2809	4558	4919	8237	6066	7146	7860	55445	87438	102344
Sistemas embarcados				80	80	80	80	83	83	124	174	130
Defesa cibernética									930	4785	84140	100924
Ciência de dados											96105	133026
Internet das coisas											3919	77239
Inteligência artificial											322	87850
Computação e Tecnologias da Informação e Comunicação (TIC) em biociências e saúde											297	8047
Agrocomputação											317	8937
Criação digital											471	502

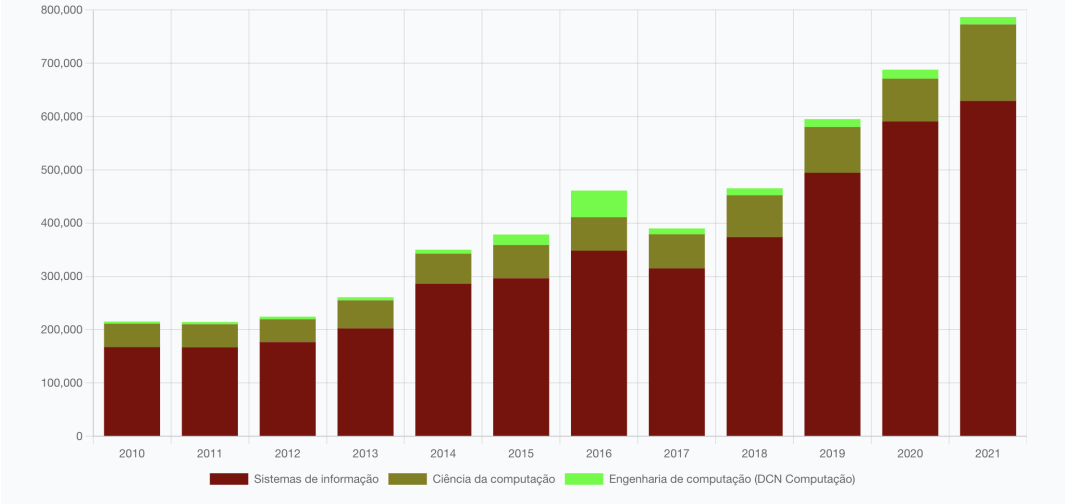
Figura 6.2 – Consulta com exclusão de cursos - Gráfico



Na sequência, podemos fazer uma consulta somente com os três, como visto na figura 6.3.

Figura 6.3 – Consulta somente de Sistemas de Informação, Ciência da Computação e Engenharia de Computação

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Sistemas de informação	167575	166972	176760	202532	286534	296563	348371	315146	373883	494553	590948	629534
Ciência da computação	43730	43228	42898	52598	56241	62462	62882	63921	78462	85897	80260	143397
Engenharia de computação (DCN Computação)	4126	4372	4934	5875	7476	19710	49914	11114	13104	14854	16884	13942



Observando os gráficos e as tabelas pode-se concluir que, por mais que o aumento absoluto da quantidade de alunos destes três cursos principais seja maior, a curva de aumento dos outros cursos é mais acentuada, significando, portanto, que os estudos de tecnologia estão se tornando mais diversos.

6.2.2 Evolução dos cursos voltados ao entretenimento

A região do país que tradicionalmente concentra a indústra do entretenimento é a região sudeste. Exemplos são as redes Record, Bandeirantes e SBT, baseadas em São Paulo, e a Rede Globo, no Rio de Janeiro. Sendo assim, cabe questionar se esta região do país também concentra os cursos de Criação Digital e de Jogos Digitais, que são os cursos de tecnologia com mais ênfase no entretenimento.

Para responder o questionamento, é valido realizar duas consultas: uma com somente os dois cursos e incluindo somente a região sudeste e outra com os mesmos dois cursos e excluindo somente a região sudeste. Os resultados podem ser observados nas figuras 6.4 e 6.5.

Figura 6.4 – Consulta de cursos voltados ao entretenimento na região sudeste

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Jogos digitais	1615	1780	1942	2797	3580	5153	6779	7469	11628	18617	17898	37805
Criação digital											255	279

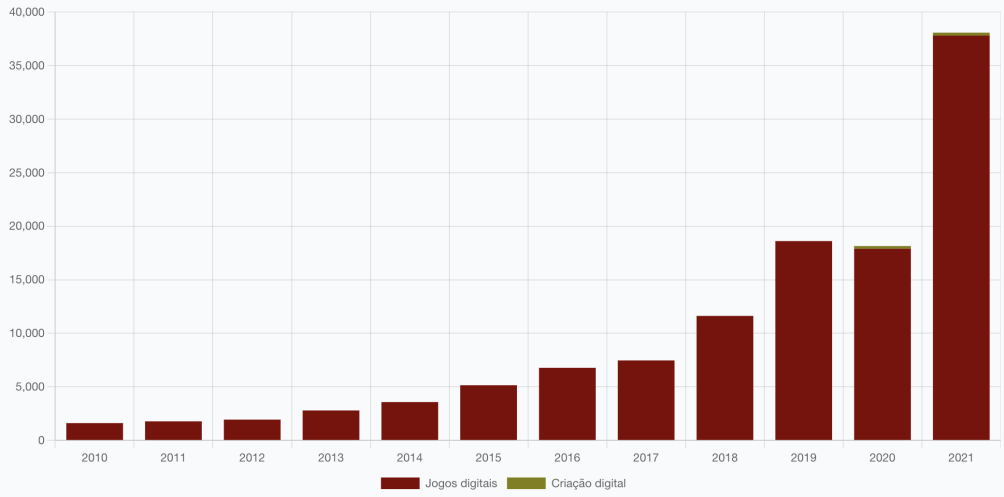
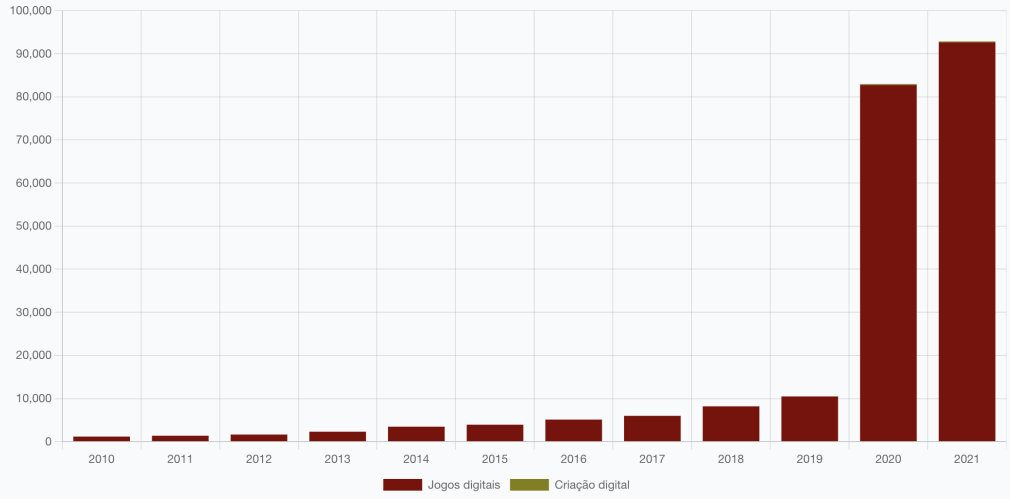


Figura 6.5 – Consulta de cursos voltados ao entretenimento nas demais regiões

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Jogos digitais	1174	1370	1646	2307	3470	3930	5115	5989	8187	10485	82737	92624
Criação digital											216	223



Pode-se observar que historicamente a região sudeste tem sim um domínio na ocorrência destes dois cursos. O curso de Criação Digital tem mais de metade de seus alunos matriculados na região sudeste em todos os anos com dados, mesmo que conte com somente dois anos de presença no senso. Até o ano de 2019 mais de metade dos alunos matriculados em Jogos Digitais no Brasil cursavam na região sudeste. Com um aumento repentino de alunos matriculados no ano de 2020, contudo, a região passou a contar com somente 17,8% dos aluno. No ano de 2021 este valor voltou a subir para 28,9%, mas ainda não chega nos termos pré 2020.

6.2.3 Comparação de crescimento do ensino EAD entre os cursos mais e menos frequentes

Nos últimos anos a modalidade de ensino remoto vem ganhando força em várias áreas como trabalho, atendimento e ensino. Este crescimento se acentuou de forma geral no ano de 2020 com as restrições de convivência impostas pela pandemia de COVID 19. Antes disso, existia um estigma maior sobre esta modalidade sobre as áreas de ensino e trabalho mais tradicionais, o que podia levar áreas mais tradicionais a resistirem à sua adoção.

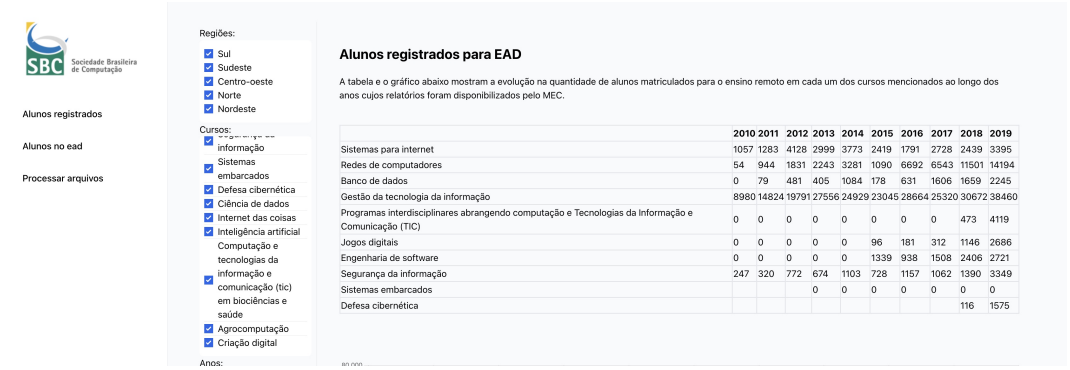
Conforme já discutido, existe um grupo de três cursos que, devido às suas con-

centrações históricas de alunos, podem ser chamados de mais tradicionais. Sendo assim, surge a dúvida de qual grupo de cursos (mais ou menos tradicionais) vinha adotando o ensino remoto mais rapidamente antes da pandemia.

Figura 6.6 – Consulta de cursos tradicionais na modalidade EAD até 2019

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Sistemas de informação	20017	29922	38469	51941	91308	64031	93265	91857	113452	200582
Ciência da computação	0	0	0	0	73	29	4	114	685	9248
Engenharia de computação (DCN Computação)	0	0	0	0	0	5127	4947	2129	677	10364

Figura 6.7 – Consulta de cursos menos tradicionais na modalidade EAD até 2019 - Tabela

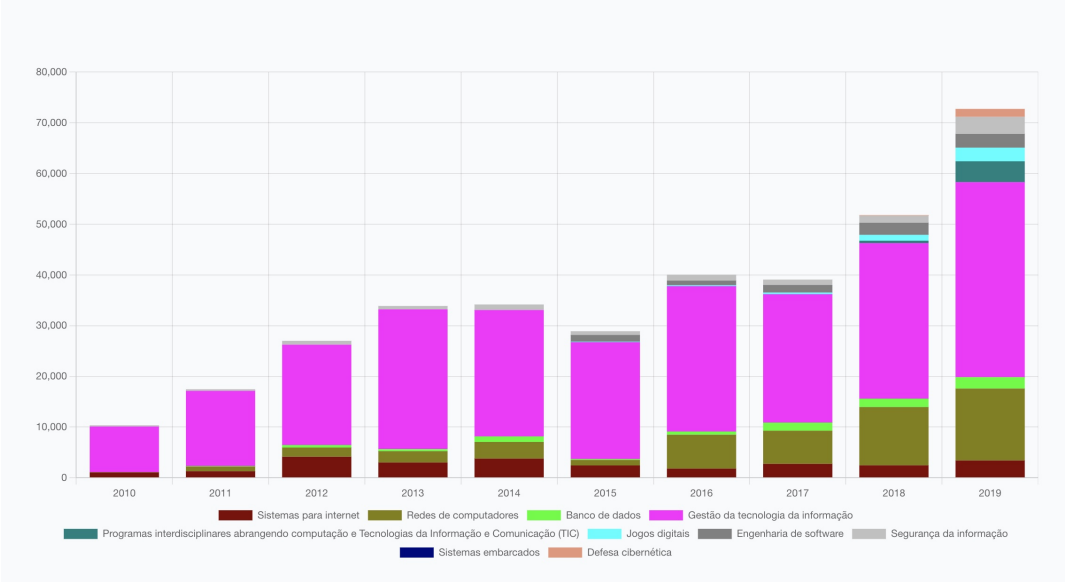


	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Sistemas para internet	1057	1283	4128	2999	3773	2419	1791	2728	2439	3395
Redes de computadores	54	944	1831	2243	3281	1090	6692	6543	11501	14194
Banco de dados	0	79	481	405	1084	178	631	1606	1659	2245
Gestão da tecnologia da informação	8980	14824	19791	27556	24929	23045	28664	25320	30672	38460
Programas interdisciplinares abrangendo computação e Tecnologias da Informação e Comunicação (TIC)	0	0	0	0	0	0	0	0	473	4119
Jogos digitais	0	0	0	0	0	96	181	312	1146	2686
Engenharia de software	0	0	0	0	0	1339	938	1508	2406	2721
Segurança da informação	247	320	772	674	1103	728	1157	1062	1390	3349
Sistemas embarcados				0	0	0	0	0	0	0
Defesa cibernética									116	1575

A figura 6.6 mostra a evolução na quantidade de alunos na modalidade EAD para os cursos ditos tradicionais. Já as figuras 6.7 e 6.8 mostram os resultados da consulta para os demais cursos. Como o questionamento é sobre o cenário pré pandemia, utilizou-se dados somente até 2019.

Cruzando estas novas figuras com os dados da figuras 6.3, 6.1 e 6.2, vemos que o percentual de alunos na modalidade EAD progride, nos anos de 2010, 2015 e 2019,

Figura 6.8 – Consulta de cursos menos tradicionais na modalidade EAD até 2019 - Gráfico



para aproximadamente 9%, 18% e 37% para os cursos mais populares. Para os demais cursos, nos mesmos anos, os valores aproximados são 9%, 14% e 17% nos mesmos anos. Assim, podemos observar que a proporção na quantidade de alunos na modalidade EAD já vinha aumentando mais rapidamente para estes três cursos que historicamente têm uma quantidade maior de alunos.

6.3 Limitações

A maior limitação apresentada pela base de dados é o fato de que um mesmo curso pode aparecer múltiplas vezes associado a uma mesma instituição de ensino em um dado ano. Sendo assim, não existe nenhum mecanismo que possa ser implementado para proteger a aplicação contra uma planilha sendo inserida mais de uma vez.

No âmbito das consultas, a maior limitação no momento é a falta de recursos visuais para indicar regiões do país. As regiões podem ser usadas em filtros e executando uma consulta com cada região podemos levantar dados que permitenos efetuar comparações, mas não existe nenhuma forma de ter comparações entre diferentes regiões do país na tela ao mesmo tempo.

6.4 Considerações Finais

Nestes capítulos foram apresentados exemplos de aplicabilidade dos filtros dinâmicos desenvolvidos. Eles nos dão a possibilidade de isolar fatores como os cursos e a região do país, permitindo uma análise mais granular dos dados que temos. Isso permite com que uma gama maior de hipóteses possam ser verificadas com o mesmo conjunto de dados quando comparando com os relatórios estáticos.

7 CONCLUSÃO

Neste trabalho foi explorado o contexto, o levantamento de requisitos e o desenvolvimento de uma ferramenta para a visualização de dados coletados e distribuídos pelo INEP sobre o ensino superior na área de tecnologia no Brasil. Foram feitos também alguns questionamentos referentes a tais dados para que pudessemos utilizar a ferramenta implementada para respondê-los.

As principais contribuições desta ferramenta são sobre a economia de recursos da Diretoria de Educação da Sociedade Brasileira de Computação e potencial aumento de conhecimento extraído das bases de dados disponibilizadas pelos censos. Com o uso desta ferramenta há uma considerável redução no custo operacional na geração de novos relatórios. Com a possibilidade de se fazer o upload de arquivos, esta ferramenta poderá ser utilizada por tempo indeterminado. Há também a vantagem do aumento da granularidade dos filtros, permitindo que consultas que baseiam hipóteses que surgem a qualquer momento podem ser feitas, não mais dependendo do conjunto de consultas que foram pensados na confecção do relatório.

Por fim, seguem potenciais pontos de melhoria ou de extensão da ferramenta em trabalhos futuros:

- Exibir uma linha com o total de cada ano nas tabelas;
- Implementar novas consultas que permitam a comparação entre diferentes regiões do país de maneira visual;
- Utilizar outros atributos da base dados para novas consultas, como a quantidade de alunos matriculados do gênero feminino ou a quantidade de discentes ingressos e egressos por ano;
- Implementar uma funcionalidade que permite a *download* de um *dump* do banco de dados para que usuários técnicos possam realizar suas próprias consultas sobre os dados já estruturados.

REFERÊNCIAS

BARROS, T.; SILVA, I.; GUEDES, L. A. Modelagem e visualização científica de dados educacionais: Estudo de caso sobre o desempenho em componentes curriculares. In: . [S.l.: s.n.], 2017. p. 654.

BORCHARDT, G. et al. Ferramenta de visualização de dados do censo da educação superior do inep. In: **Anais do X Workshop de Computação Aplicada em Governo Eletrônico**. Porto Alegre, RS, Brasil: SBC, 2022. p. 227–234. ISSN 2763-8723. Disponível em: <https://sol.sbc.org.br/index.php/wcge/article/view/20725>.

BURBECK, S. Applications programming in smalltalk-80: How to use model-view-controller (mvc). 1987.

CHARTJS. **ChartJS**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.chartjs.org>.

CODD, E. F. A relational model of data for large shared data banks. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 13, n. 6, p. 377–387, jun 1970. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/362384.362685>.

CODD, E. F. Further normalization of the data base relational model. **Research Report / RJ / IBM / San Jose, California**, RJ909, 1971.

CODD, E. F. Recent investigations into relational data base. 1974.

CONNOLLY, T.; BEGG, C. **Database Systems: A Practical Approach to Design, Implementation and Management**. [S.l.]: Pearson, 2014. (Always learning). ISBN 9781292061184.

ELIXIR. **Elixir**. 2023. Acessado em 18/03/2023. Disponível em: <https://elixir-lang.org>.

ERLANG. **Erlang**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.erlang.org>.

EXUNIT. **ExUnit**. 2023. Acessado em 18/03/2023. Disponível em: <https://hexdocs.pm/ex_unit/ExUnit.html>.

FERREIRA, L.; RODRIGUES, R.; SOUZA, R. Dados abertos educacionais brasileiros: Um mapeamento sistemático da literatura. In: **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 1186–1195. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/18141>.

FONSECA, S. O. d.; NAMEN, A. A. Mineração em bases de dados do inep: uma análise exploratória para nortear melhorias no sistema educacional brasileiro. **Educação em Revista**, Faculdade de Educação da Universidade Federal de Minas Gerais, v. 32, n. Educ. rev., 2016 32(1), p. 133–157, Jan 2016. ISSN 0102-4698. Disponível em: <https://doi.org/10.1590/0102-4698140742>.

FRAMEWORK, P. **Phoenix**. 2023. Acessado em 18/03/2023. Disponível em: <https://phoenixframework.org>.

FRAMEWORK, P. **Phoenix**. 2023. Acessado em 18/03/2023. Disponível em: <https://hexdocs.pm/phoenix_live_view/Phoenix.LiveView.html>.

GOVBR. **Competências**. 2023. Acessado em 18/03/2023. Disponível em: <http://portal.mec.gov.br/institucional/competencias>.

GOVBR. **Competências**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.gov.br/inep/pt-br/acesso-a-informacao/institucional/competencias>.

GROUP, P. G. D. **PostgreSQL**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.postgresql.org>.

HEWITT, C. Actor model of computation. 2010.

ILHA, L. B. A construção de um data warehouse utilizando os indicadores educacionais do inep. 2021.

INEP. **Censo de Educação Superior**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/censo-da-educacao-superior>.

INMON, W. H.; NESAVICH, T. F. **Business Intelligence: Roadmap to Success**. [S.l.]: Pearson Education, 2008.

OLIVEIRA, L. F. B. de; SOARES, S. S. D. **Determinantes da repetência escolar no Brasil: uma análise de painel dos censos escolares entre 2007 e 2010**. 2012.

SBC. **Diretoria de Educação**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.sbc.org.br/educacao/diretoria-de-educacao>.

SBC. **Relatórios Anuais**. 2023. Acessado em 08/04/2023. Disponível em: <https://www.sbc.org.br/documentos-da-sbc/category/196-relatorios-anuais>.

SBC. **Sobre**. 2023. Acessado em 18/03/2023. Disponível em: <https://www.sbc.org.br/institucional-3/sobre>.