

Associação Propagadora Esdeva  
Centro Universitário Academia –  
UniAcademia  
Curso de Sistemas de Informação  
Trabalho de Conclusão de Curso – Artigo

## UMA FERRAMENTA PARA MINERAÇÃO DE DADOS DE PROGRAMAS SOCIAIS DO GOVERNO FEDERAL

*Lucas Gomes da Silva<sup>1</sup>*  
*Centro Universitário Academia, Juiz de Fora, MG*  
*Tassio Ferezini Martins Sirqueira<sup>2</sup>*  
*Centro Universitário Academia, Juiz de Fora, MG*

Linha de Pesquisa: Banco de Dados

### RESUMO

O Governo Federal possui vários programas sociais para reduzir a pobreza no país. Contudo, mesmo com vários recursos injetados na distribuição de renda, o Brasil ainda apresenta um número elevado de pessoas em situação de vulnerabilidade social. O objetivo deste trabalho é fornecer uma ferramenta Web para análise de dados disponibilizados pelo Governo Federal através da mineração de dados. A ferramenta denominada de AIA-Social utiliza a API do *weka* para gerar regras de associação, através do algoritmo *Apriori*. Após a análise dos dados, a ferramenta apresenta o conjunto de regras descobertas para serem verificadas por um especialista, auxiliando na formulação de políticas públicas para programas sociais. Embora a ferramenta utilize até o momento somente o algoritmo *Apriori*, possibilitou identificar na amostra de dados estudada algumas paridades entre as características dos indivíduos.

**Palavras-chave:** Mineração. Apriori. CadÚnico. Weka.

### ABSTRACT

*The Federal Government has several social programs to reduce poverty in the country. However, even with several resources injected into the income distribution, Brazil still has a high number of socially vulnerable people. The objective of this work is to provide a Web tool for data analysis made available by the Federal Government through data mining. The tool called AIA-Social uses the weka API to generate association rules through the Apriori algorithm. After the data analysis, the tool presents the set of rules discovered, to be verified by an expert, helping in the formulation of public policies for social programs. Although the tool uses only the Apriori algorithm so far, it has made it possible to identify in the data sample studied some parities between the characteristics of individuals.*

---

<sup>1</sup> Discente do Curso de Sistemas de Informação do Centro Universitário Academia – UniAcademia. E-mail: lucas.sist.inf@gmail.com

<sup>2</sup> Docente do Curso de Sistemas de Informação do Centro Universitário Academia. Orientador.

## 1 INTRODUÇÃO

Atualmente o Governo Federal possui diversos programas sociais para redução da pobreza no país, sendo o principal deles o Programa Bolsa Família (PBF). Conforme Facioli & Sirqueira (2017), a seleção das famílias para os programas sociais são realizadas através do Ministério de Desenvolvimento Social (MDS), com base nos dados do Cadastro Único (CadÚnico), sendo a seleção feita a partir de cotas de benefícios municipais, previamente estabelecido com base no número de famílias que vivem em situação de pobreza em cada cidade.

Mesmo com diversos recursos injetados na distribuição de renda, o Brasil ainda sofre com o nível de pobreza, conforme Barros *et al.* (2000). De acordo Carmen Nery (2019), a partir de 2015 o Brasil passou a enfrentando um crescimento no número de miseráveis, contudo, em 2018 a velocidade de crescimento foi similar ao final do ano de 2014.

De acordo com esse pressuposto, cabe avaliar se as regras de elegibilidade estão de acordo com as necessidades dos indivíduos para uma futura diminuição da pobreza e desigualdades sociais no país, dado que tais regras devem representar as condições dos indivíduos na faixa de pobreza e extrema pobreza. Para isso pode-se descobrir similaridades entre o perfil dos indivíduos a fim de tentar entender se as regras de condicionalidade e elegibilidade do programa atendem as pessoas em situação de maior vulnerabilidade, a partir dessas regras pode-se gerar políticas públicas as quais embasadas nesse tipo de conhecimento podem trazer maior assertividade aos programas sociais. Para avaliar as condições dos indivíduos podemos utilizar diversas bases de dados governamentais disponíveis no Portal da Transparência do Governo Federal<sup>3</sup>.

Diante do exposto, a mineração de dados pode ser uma ferramenta muito poderosa para auxiliar no entendimento do motivo pelo qual o país ainda segue com altos índices de pobreza. Com o uso de algoritmos matemáticos, podemos gerar regras de associações nas quais é possível minerar dados de beneficiários e não beneficiários do PBF e associar quais suas condições de vida e quais são suas similaridades, porém, as ferramentas disponíveis no mercado ainda são de difícil uso e dependem muito esforço de quem as está utilizando, o que pode diminuir a busca por essas soluções e atrasar resultados de políticas sociais.

Para o presente artigo, foram utilizadas as amostras de 2018<sup>4</sup>, das bases de dados do CadÚnico, disponibilizadas por amostragem anualmente. Para atuar de maneira mais assertiva, as análises foram feitas utilizando o algoritmo Apriori (BORGELT & KRUSE, 2002), a fim de localizar regras de associações que geram algum conhecimento acerca da amostra.

Além desta introdução, na seção 2 será abordado o referencial teórico sobre o CadÚnico, do Programa Bolsa Família (PBF) e do algoritmo Apriori. Na seção 3 será apresentada a ferramenta desenvolvida, denominada de “AIA-Social”, com seus detalhes técnicos. Já na seção 4 serão abordadas as diretrizes de utilização da ferramenta. Por fim, na seção 5, serão apresentadas algumas considerações finais, as limitações da ferramenta e as

<sup>3</sup> Portal da Transparência. Disponível em: <<http://www.portaltransparencia.gov.br/>>. Acessado em 12 de junho de 2020.

<sup>4</sup> Dados do CadÚnico. Disponível em: <<http://mds.gov.br/area-de-imprensa/noticias/2018/setembro/aplicativo-facilita-acesso-aos-dados-do-cadastro-unico>>. Acessado em 12 de junho de 2020.

propostas de trabalhos futuros.

## **2 REFERENCIAL TEÓRICO**

Essa seção tem por objetivo vislumbrar os pontos motivadores que levaram ao desenvolvimento deste trabalho, abordando os temas que podem passar despercebidos durante o dia-a-dia, mas que são base de conhecimento necessário para o entendimento final da ferramenta e seu objetivo social.

### **2.1 Cadastro Único**

O Cadastro Único (CadÚnico) tem por objetivo saber como as pessoas de baixa renda vivem no Brasil (BRASIL, 2020). De acordo com Brasil (2020) o programa foi criado pelo Governo Federal e é mantido pelas prefeituras, que fazem o cadastro das pessoas de baixa renda, para que possam pleitear acesso à diversos programas sociais do Governo Federal, desde que atendam suas respectivas regras de condicionalidade e elegibilidade. Possuir cadastrado no CadÚnico não significa que a pessoa ou família seja atendida por um programa social, mas é o primeiro passo para que possa ser contemplada com algum auxílio governamental.

De acordo com Brasil (2017), o CadÚnico em 2017, possuía dados de aproximadamente 26 milhões de famílias, onde esses dados são utilizados por diversos programas sociais. Os dados são disponibilizados periodicamente para possibilitar ao público acesso às informações, porém, garantindo a confidencialidade dos indivíduos. O Ministério da Cidadania também disponibiliza bases amostrais reduzidas para que pesquisadores possam trabalhar, porém, passíveis de mineração e análise dos dados.

### **2.2 Bolsa Família**

O Governo Federal possui diversos programas sociais, sendo um dos principais o Programa Bolsa Família (PBF). Segundo Brasil (2020), o programa tem por objetivo contribuir para o combate à pobreza e oferece acompanhamento às famílias nas áreas da saúde e educação. As famílias que são cadastradas e contempladas têm acompanhamento de saúde e frequência escolar, para crianças e adolescentes entre 6 a 17 anos de idade. Podem participar do PBF famílias extremamente pobres, ou seja, que vivem com renda per capita de até 89,00 reais mensais ou famílias pobres que possuam gestantes, crianças ou adolescentes e que a renda mensal per capita esteja entre 89,01 e 178,00 reais (BRASIL, 2020).

O PBF não trabalha somente com o acompanhamento das famílias, mas também com a transferência de renda mensal de acordo com o enquadramento da família a partir das informações do CadÚnico. Segundo dados do Portal da Transparência, o programa pagou, em 2019, o montante de aproximadamente 33,01 bilhões de reais, o que correspondeu a 1,27% dos gastos públicos para o ano. Com as regras de elegibilidade do programa as regiões sudeste e nordeste foram as mais beneficiadas pela transferência de renda (Brasil 2020).

Mesmo com as regras de condicionalidades e elegibilidade dos programas sociais do Governo Federal, é visível que ainda não atendem todos os indivíduos que necessitam do subsídio governamental. Segundo Carmen Nery (2019), em 2018 o Brasil ainda possuía 13,5 milhões de pessoas vivendo na linha da extrema pobreza e o estado o Maranhão era o estado

mais pobre do país, além disso pessoas pretas e pardas, que possuíam formação fundamental incompleta ou sequer possuíam formação, correspondiam a cerca de 73% dos mais pobres.

### 2.3 Apriori

O Apriori é um algoritmo de associação que gera padrões para a ocorrência de determinados itens. O Algoritmo realiza diversas passagens na base de dados, considerando que um conjunto de  $n$  itens existem  $(2^n - 1)$  regras possíveis, a fim de encontrar os melhores conjuntos de itens frequentes que atendam o suporte e a confiança mínima informada. O Apriori inicia o processamento para conjuntos de apenas 1 item, logo em seguida passa para conjuntos de 2 itens até conjuntos de  $k$  itens, utilizando o princípio de que cada subconjunto de um determinado conjunto deve ser frequente, sendo assim, reduz o número de candidatos para serem comparados caso algum subconjunto não seja frequente (BRUSSO et al., 2000).

### 2.4 Dificuldade de Mineração

Segundo Sirqueira *et al.* (2018) ferramentas estatísticas são independentes e dependem muito tempo e esforço dos pesquisadores para realizarem tarefas simples, além de muitas vezes economistas necessitarem de análise de dados complexos e essas ferramentas não fornecerem recursos suficientes para coletar e extrair amostras dos dados. Diante disso, as ferramentas disponíveis atualmente no mercado podem acabar se tornando complexas quando o uso deve ser feito por pessoas com baixo conhecimento sobre hardware, tecnologias e softwares em geral.

Atualmente existem diversas ferramentas no mercado que podem ajudar o pesquisador a analisar e minerar dados, como por exemplo o WEKA<sup>5</sup>. Ferramenta desenvolvida pela Universidade de *Waikato* que possui diversos algoritmos matemáticos já implementados. Essa ferramenta possui duas formas para o pesquisador trabalhar, a primeira é uma API que pode ser integrada com diversas linguagens de programação e a segunda é uma interface na qual o usuário pode realizar a análise com diversos algoritmos.

O Weka torna o processo mais simples, pois já possui os algoritmos de mineração implementados, porém, mesmo com essas qualidades, assim como outras aplicações na mesma categoria e que são instaladas na máquina do usuário, ainda exige que o pesquisador que a utiliza tenha em sua máquina todos os recursos operacionais necessários para mineração de sua base de dados, isso demanda muito tempo e pode ser muito caro quando se trata de bases de dados governamentais, às quais ultrapassam milhões de registros e exigem muita capacidade operacional para serem analisadas, o que pode inviabilizar o processo de mineração em busca de conhecimento.

## 3 A FERRAMENTA AIA-SOCIAL

Diante de todos os pontos citados até agora, nesta seção é abordada a ferramenta denominada AIA-Social (*Artificial Intelligence Analyzer-Social* ou AIA), aplicação desenvolvida durante a construção do presente artigo e que visa tornar o processo de mineração de dados em busca de conhecimento simples e objetivo, não obrigando que o usuário gaste esforço

---

<sup>5</sup> WEKA. Disponível em: < <https://www.cs.waikato.ac.nz/ml/weka/>>. Acessado em 12 de junho de 2020.

desnecessário para atingir suas necessidades. Essa ferramenta foi desenvolvida como uma aplicação Web e isso possibilitou que todo o uso computacional ficasse a cargo dos servidores da ferramenta, não necessitando uso de recursos operacionais disponíveis na máquina do usuário.

Nesta seção, serão apresentados os detalhes arquiteturais e de implementação, além de como o uso da ferramenta tornará o processo de mineração de informações mais rápido e simples.

### 3.1 Tecnologias

A AIA foi desenvolvida visando simplicidade e fácil utilização, voltada para usuário com um nível de conhecimento baixo sobre algoritmos de mineração e de programação. Para atingir isso, como requisitos de usabilidade e performance, foram utilizadas tecnologias modernas e que oferecem recursos extremamente competentes para à finalidade.

Cabe ressaltar que, conforme demonstrado na Figura 1, a aplicação foi construída com *Front-End*<sup>6</sup> e *Back-End*<sup>7</sup> desvinculados, ou seja, foi utilizado o padrão *RestFul* para construir 2 API's e o padrão SPA (Single Page Application) para construir nossa estrutura de Layout.

A SPA foi construída utilizando *JavaScript* com o *framework ReactJS*<sup>8</sup>, que possibilita a componentização de todo o Layout. Para a API primária foi utilizado o *NodeJS* com o *framework Express*<sup>9</sup> e por fim, para a API de mineração de dados<sup>10</sup>, o Java com *SpringBoot*<sup>11</sup>.

A API em *NodeJS* é responsável pelo controle de todo o fluxo do usuário, bem como pela importação e exportação dos dados para o sistema gerenciador de banco de dados (SGBD) *MongoDB*<sup>12</sup>, através das funcionalidades *MongoImport* e *MongoExport*, respectivamente, ambas disponibilizadas pelo próprio SGBD, conforme a Figura 1.

Já a API em *SpringBoot* é responsável pela mineração dos dados e pela busca dos atributos de um documento da coleção. Na Figura 1, ilustramos como funciona o fluxo de requisições da aplicação e dos dados.

Por conta da necessidade de trabalhar com dados que não possuem um esquema definido e também tratar dados que se relacionam, a AIA utiliza dois mecanismos de banco de dados, conforme Figura 1, o *MongoDB* (SGBD não relacional) para lidar com os registros do CSV (Comma-separated values) e o *PostgreSQL*<sup>13</sup> (SGBD relacional) para lidar com as informações que se relacionam, ou seja: dados do usuário, categorias e registros de

<sup>6</sup> *Front-End* AIA-Social. Disponível em: <<https://github.com/ces-uf/AIA-Social-Front>>. Acessado em 14 de junho de 2020.

<sup>7</sup> *Back-End* AIA-Social. Disponível em: <<https://github.com/ces-uf/AIA-Social>>. Acessado em 14 de junho de 2020.

<sup>8</sup> ReactJS. Disponível em: <<https://pt-br.reactjs.org/>>. Acessado em 12 de junho de 2020.

<sup>9</sup> Express. Disponível em: <<https://expressjs.com/pt-br/>>. Acessado em 12 de junho de 2020

<sup>10</sup> AIA-Social-Miner. Disponível em: <<https://github.com/ces-uf/AIA-Social-Miner>>. Acessado em 25 de junho de 2020

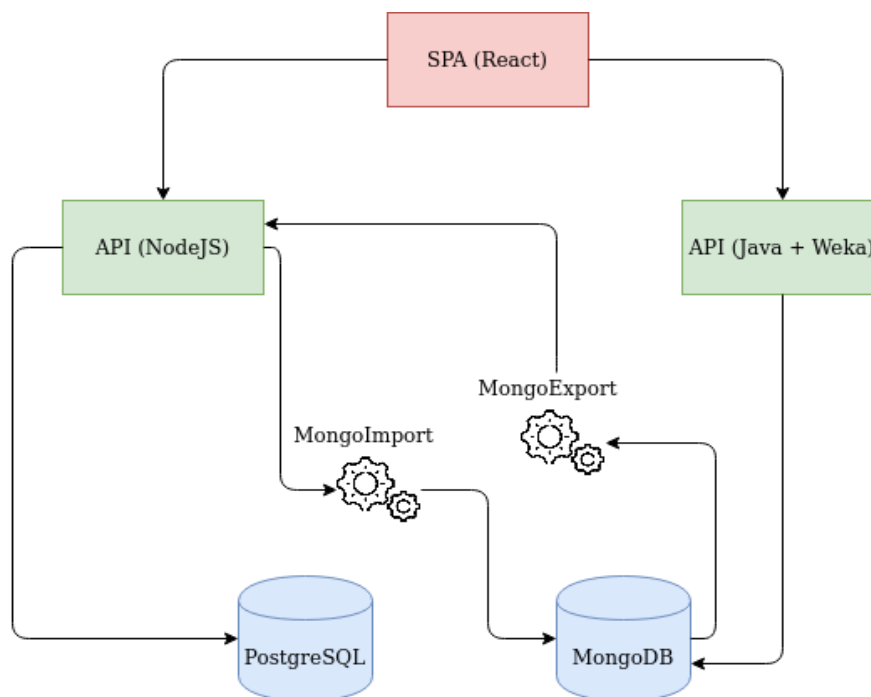
<sup>11</sup> SpringBoot. Disponível em: <<https://spring.io/projects/spring-boot>>. Acessado em 12 de junho de 2020.

<sup>12</sup> MongoDB. Disponível em: <<https://www.mongodb.com/>>. Acessado em 12 de junho de 2020.

<sup>13</sup> PostgreSQL. Disponível em: <<https://www.postgresql.org/>>. Acessado em 12 de junho de 2020.

importação.

**Figura 1. Fluxo de Requisições e Acesso a Dados na Aplicação.**



**Fonte: Elaboração Própria.**

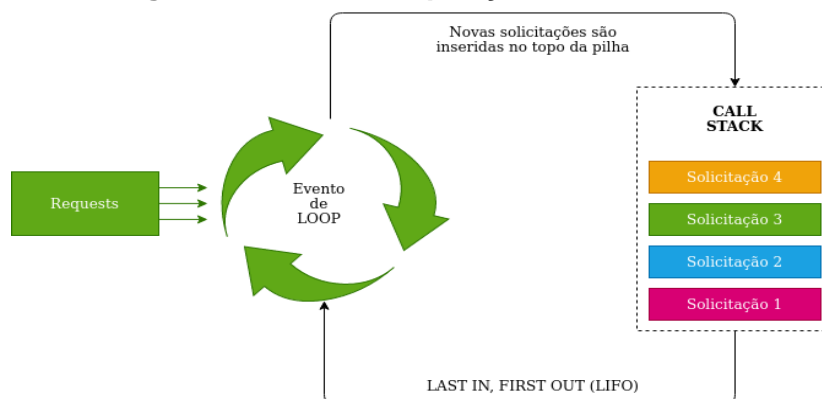
### 3.1.1 NodeJS

O Node basicamente é a forma que temos atualmente para rodar códigos construídos em JavaScript do lado do servidor. O Node funciona diferentemente das demais linguagens utilizando somente uma única thread para processar as requisições em uma pilha.

O Node também consegue processar as requisições de forma assíncrona, sendo assim, a requisição poderá ser respondida ao usuário e continuar rodando algum processo em plano de fundo, não obrigando o usuário esperar até que seja finalizado um processo que demanda muitos recursos do servidor, diferentemente de linguagens mais tradicionais esse fluxo já vem por padrão, não necessitando instanciar *threads*, e é denominado *Nonblocking I/O*. Esse recurso foi útil para a ferramenta porque trabalha com bases de dados formadas por uma grande massa de registros, o sistema foi programado de forma que o upload dos arquivos CSV (Comma-separated values) por parte do usuário, não trave o fluxo de uso nas demais partes da aplicação.

O fluxo padrão de uma *request* em Node está apresentado na Figura 2, conforme pode-se perceber na imagem, a requisição recém chegada passa pelo Evento de *LOOP* e é inserida no topo da pilha de solicitações, a cada solicitação finalizada uma nova é invocada no padrão *LIFO*, ou seja, a última que entrou será a primeira a ser processada.

Figura 2. Fluxo de Requisições em NodeJS.



Fonte: Elaboração Própria.

### 3.1.2 PostgreSQL

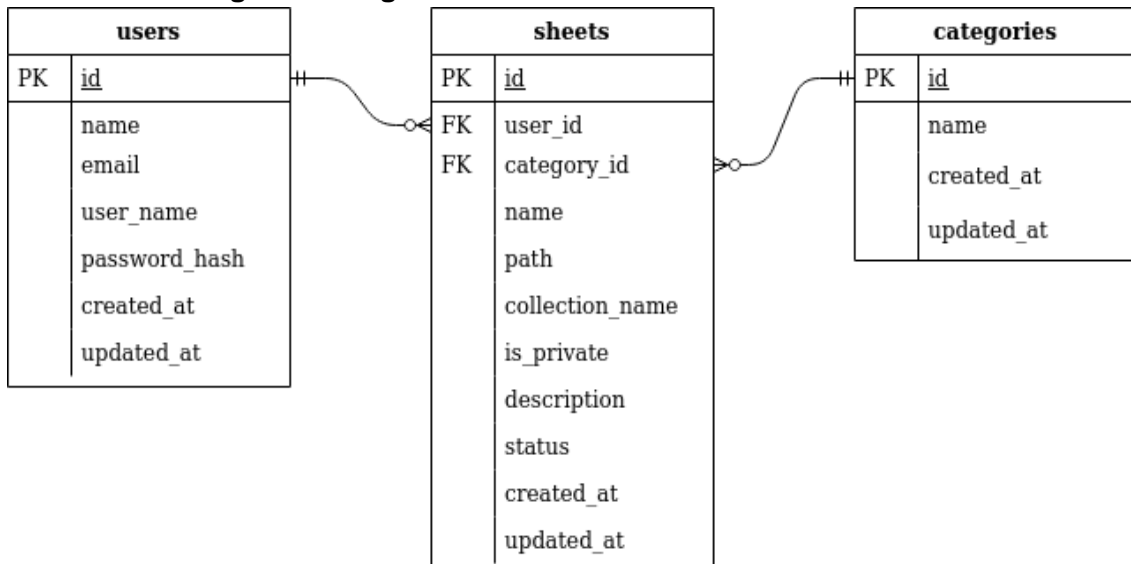
A escolha do PostgreSQL para essa aplicação se deu pela necessidade de um banco de dados relacional, seu uso fez-se simples para as necessidades da ferramenta e sua integração com *JavaScript* foi simplificada utilizando o ORM (*Object-relational mapping* - Mapeamento objeto-relacional) *Sequelize*<sup>14</sup>, além de se tratar de um banco de dados de código aberto.

Na Figura 3, pode-se verificar o Diagrama de Tabelas do banco, onde a ferramenta possui 3 tabelas que servem como controladoras das informações da aplicação, conforme detalhado a seguir:

- *users*: essa tabela será responsável por controlar os dados do usuário, a partir dela a ferramenta irá permitir o gerenciamento da conta e das funcionalidades disponíveis na aplicação.
- *sheets*: essa tabela é o principal ponto da aplicação, com ela é possível controlar todas as planilhas já importadas e como está a situação da transformação dos dados para documentos dentro de uma coleção do MongoDB. Nessa mesma tabela o NodeJS gerencia o status da importação dos registros, podendo ocorrer os seguintes casos:
  - *Enviado*: a planilha foi enviada para a aplicação, agora está sendo convertida para um formato que o MongoDB pode importar.
  - *Erro*: a base de dados foi processada, porém, no processo de importação houveram erros e o processo teve de ser abortado, a base não poderá ser utilizada para análises.
  - *Sucesso*: a base de dados foi processada e a importação ocorreu com sucesso, esse é o único status que a base estará disponível para ser analisada pelo usuário.
- *categories*: Essa tabela serve somente para auxiliar na categorização das coleções, a informação contida será exibida ao usuário para poder categorizar suas importações.

<sup>14</sup> Sequelize. Disponível em: < <https://sequelize.org/> >. Acessado em 25 de junho de 2020

**Figura 3. Diagrama de Entidades e Relacionamentos.**



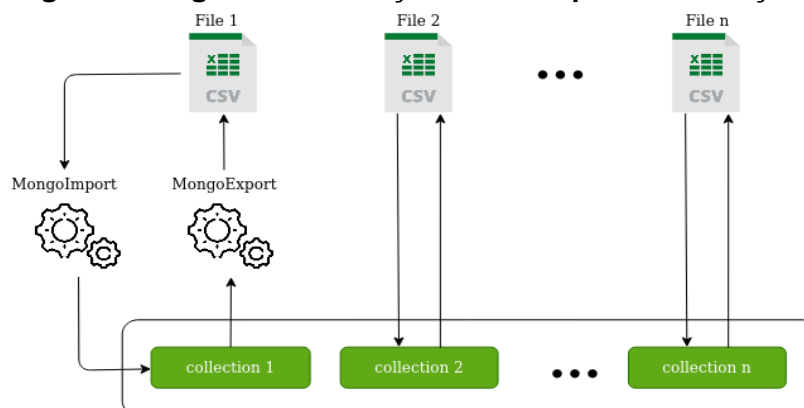
**Fonte: Elaboração Própria.**

### 3.1.3 MongoDB

Os dados que o usuário importam para aplicação não tem um esquema definido, haja visto que o mesmo poderá enviar arquivos de qualquer natureza para análise. O MongoDB é um banco de dados não relacional baseado em documento no qual é possível trabalhar com esquema livres, ou seja, não precisamos definir o esquema dos dados para inserir novos registros.

O MongoDB armazena os registros dos arquivos CSVs dos usuários em forma de documento, independentemente do tipo de registro que contém. Cada arquivo CSV importado para a AIA criará uma nova coleção de dados, conforme ilustrado na Figura 4. Sendo assim, caso o usuário tenha N arquivos importados, também terá N coleções no MongoDB, cada uma dessas coleções será devidamente controlada dentro da base de dados do PostgreSQL.

**Figura 4. Diagrama de Relação entre Arquivos e Coleções.**



**Fonte: Elaboração Própria.**



A importação de arquivos dentro de bases de dados pode ser muito demorada, porém, o MongoDB oferece recursos para esse processo, sendo:

- **MongoImport:** esse complemento do banco de dados permite importar dados para coleções, em formato CSV e JSON (*JavaScript Object Notation*), com alta quantidades de registros de modo otimizado.
- **MongoExport:** esse complemento é utilizado para gerar exportações de dados que já estão presentes no MongoDB. Essa exportação foi utilizada na AIA para que o *WEKA* consiga analisar os dados diretamente de arquivos CSV.

A importação dos arquivos será realizada utilizando o **MongoImport** diretamente pelo servidor Node. Ao solicitar a importação, o CSV será convertido para um novo arquivo seguindo um formato que atenda ao **MongoImport** e, após a conversão dos dados, o **MongoImport** será invocado passando o arquivo convertido como parâmetro. Ao final será gerada uma nova coleção no MongoDB com os registros em formato BSON.

A exportação dos dados para o **WEKA** utilizará o **MongoExport**, o qual será invocado dentro do servidor Node e será processado com os dados que estão na coleção solicitada do MongoDB. Os registros persistidos na coleção serão exportados para um arquivo CSV formatado com os parâmetros para o **WEKA**, que realizará a análise e mineração.

### 3.1.4 Java e WEKA

Para mineração dos dados importados pelos usuários da AIA, utiliza a API do **WEKA** para processar o algoritmo **Apriori** e gerar as informações que serão disponibilizadas para os usuários. Buscando uma melhor integração com o **WEKA**, foi construída uma API que é responsável pela integração com os dados disponibilizados pela coleção do MongoDB. Para trabalhar com essa API utilizou-se o *framework* **SpringBoot**, por meio de requisições HTTP.

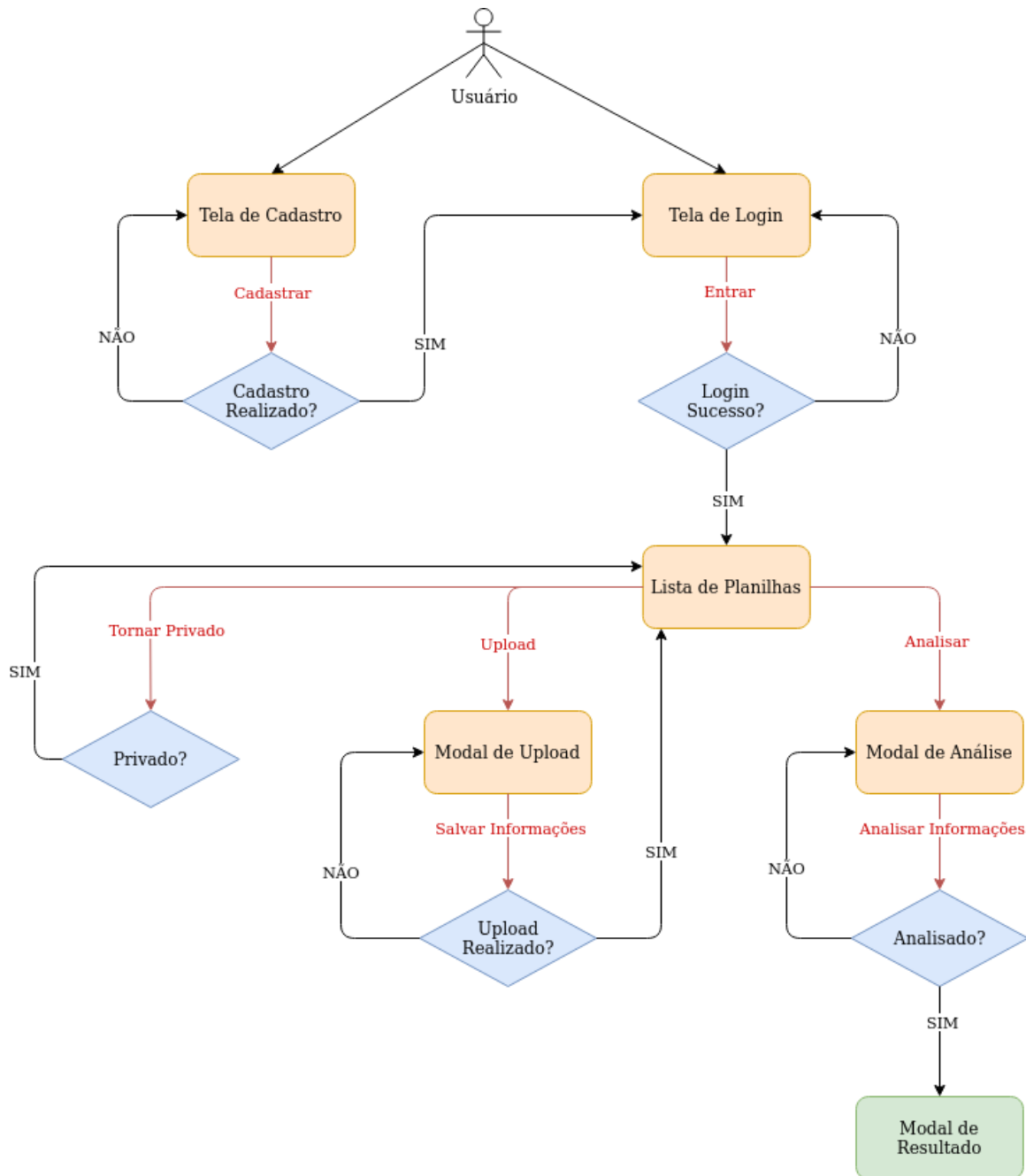
## 3.2 Funcionamento

A ferramenta visa realizar a mineração de dados em bases do Governo Federal, limitando-se a estas por suas peculiaridades que são tratadas durante a importação para o MongoDB, demandando um pré-processamento dos dados a fim de possibilitar o uso da *API* do *Weka* para analisar os registros. Até o momento a AIA já possui configurado o algoritmo **Apriori**, e a partir da importação da base de dados e solicitação de análise, os usuários podem visualizar o resultado do algoritmo na própria ferramenta.

Por se tratar de uma aplicação que visa facilitar a usabilidade do usuário, a AIA possui apenas 6 telas com fluxos simples e que podem ser acessadas conforme o usuário do sistema. Na Figura 5 está diagramado o fluxo de uso de toda a aplicação. Qualquer usuário poderá realizar seu cadastro e a partir do momento que se autenticar na aplicação terá o controle de todas as planilhas que possuem os dados importados do Portal da Transparência.

O ponto inicial da aplicação é a tela de cadastro ou, caso o usuário já tenha credenciais de login, a tela de login, o ponto final de todo o fluxo é o modal de resultado, o qual terá a análise do processamento dos dados pelo **Apriori**.

**Figura 5. Diagrama de Fluxo de Uso da Aplicação.**



**Fonte: Elaboração Própria.**

#### **4 Diretrizes de Utilização**

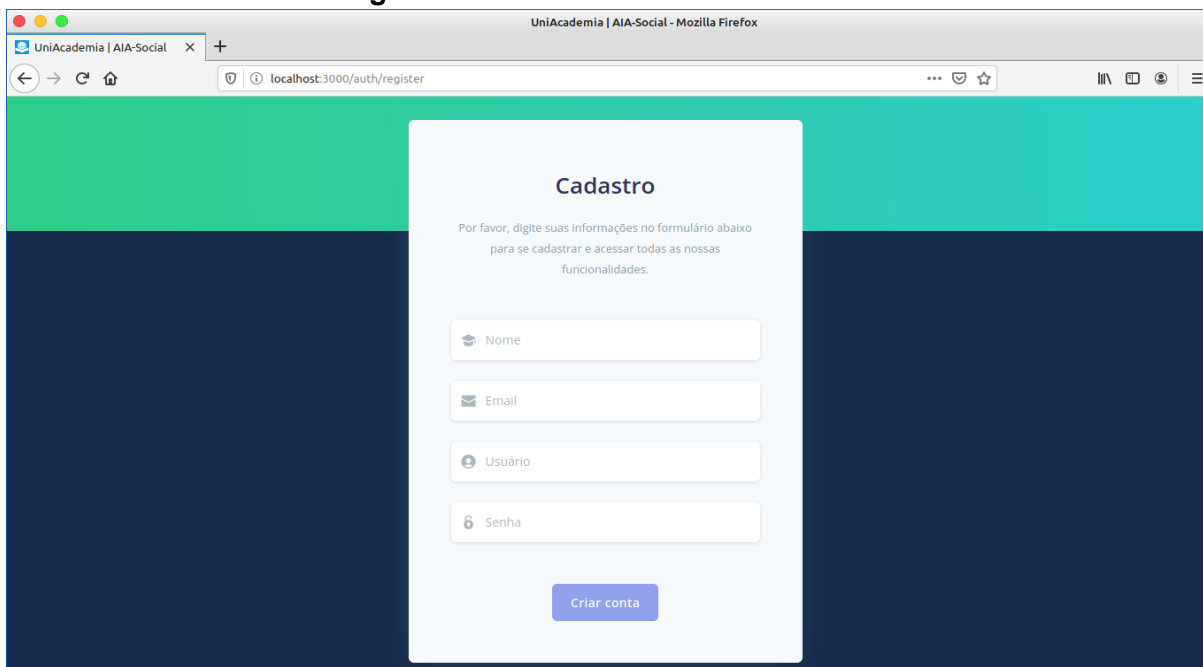
Conforme explicado nas seções anteriores, a AIA é uma ferramenta web, simples e de fácil uso. Por conta disso, nessa seção será abordado o uso da aplicação passando por suas telas e explicando o fluxo que o usuário deve seguir até atingir os resultados apresentados pela mineração dos dados.

## 4.1 Cadastro

Conforme apresentado na Figura 5, o ponto de entrada para a ferramenta é a tela de cadastro, Figura 6, nesse ponto o usuário deverá informar alguns dados para se registrar na AIA, todos esses dados serão persistidos na tabela users dentro da base de dados do PostgreSQL. Cada usuário deve ser único dentro da aplicação, sendo assim, os campos têm as seguintes regras:

- *Nome*: nome da pessoa que está criando o usuário, o campo é obrigatório e não possui regras de unicidade.
- *E-mail*: e-mail pessoal do usuário, deve ser no formato de um e-mail válido, é obrigatório e não pode estar cadastrado dentro do sistema; esse campo também será utilizado para garantir a unicidade do usuário dentro da AIA.
- *Usuário*: nome fictício para o usuário, esse campo é único e é obrigatório, será utilizado como uma informação para autenticá-lo na ferramenta.
- *Senha*: campo obrigatório que deve ter no mínimo 6 caracteres, é criptografado para ser persistido no banco de dados e somente o usuário conhece seu valor, administradores da AIA não tem conhecimento dessa informação de forma descriptografada.

**Figura 6. Tela de Cadastro de Usuário.**



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/auth/register'. The page content is a registration form titled 'Cadastro'. Below the title, there is a message: 'Por favor, digite suas informações no formulário abaixo para se cadastrar e acessar todas as nossas funcionalidades.' The form consists of four input fields with icons: 'Nome' (person icon), 'Email' (envelope icon), 'Usuário' (person icon), and 'Senha' (lock icon). At the bottom of the form is a blue button labeled 'Criar conta'.

**Fonte: Elaboração Própria.**

## 4.2 Login

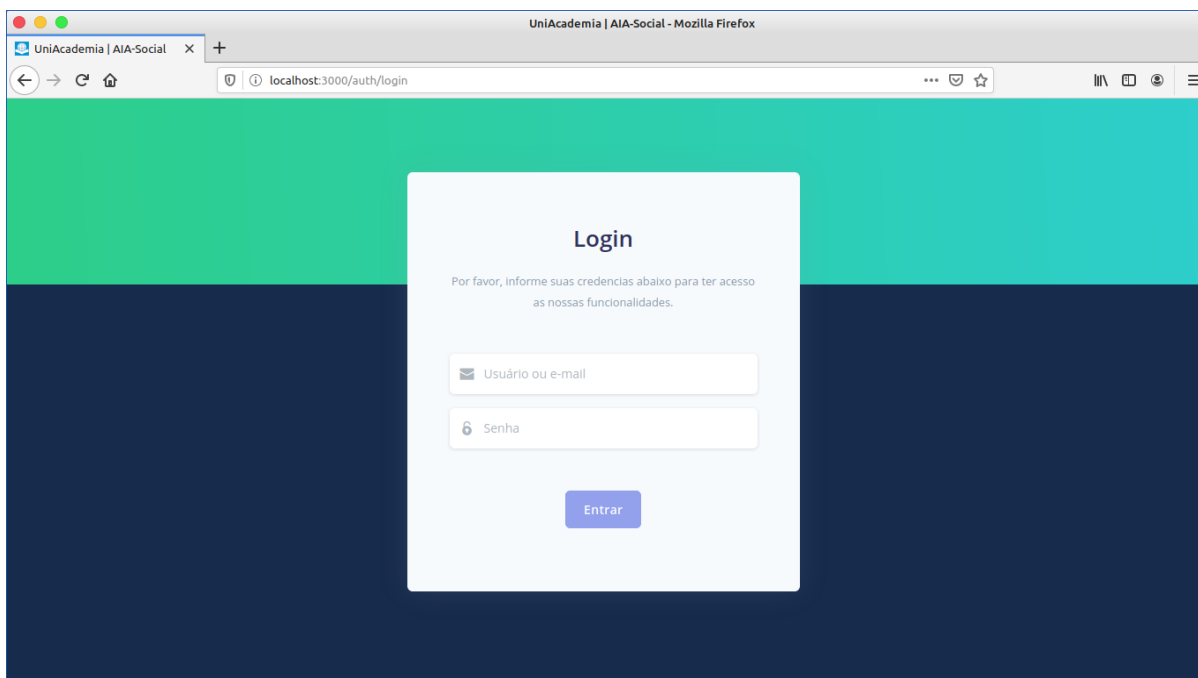
A segunda tela que o usuário terá acesso é a de Login, Figura 7, para realizar qualquer ação dentro da AIA deve-se estar autenticado. As informações solicitadas na tela de Cadastro (Figura 6) serão utilizadas aqui para verificar se o usuário existe e se suas credenciais estão de acordo com o registrado no banco de dados. Os campos solicitados para essa tela são:

- *Usuário* ou e-mail: esse campo é obrigatório e é uma junção das informações que

geram a unicidade do nosso usuário, tanto usuário quanto e-mail podem ser utilizados aqui para autenticá-lo na ferramenta.

- **Senha:** esse campo é obrigatório e será encriptado para ser comparado com a informação que está registrada no banco de dados.

**Figura 7. Tela de Login de Usuário.**

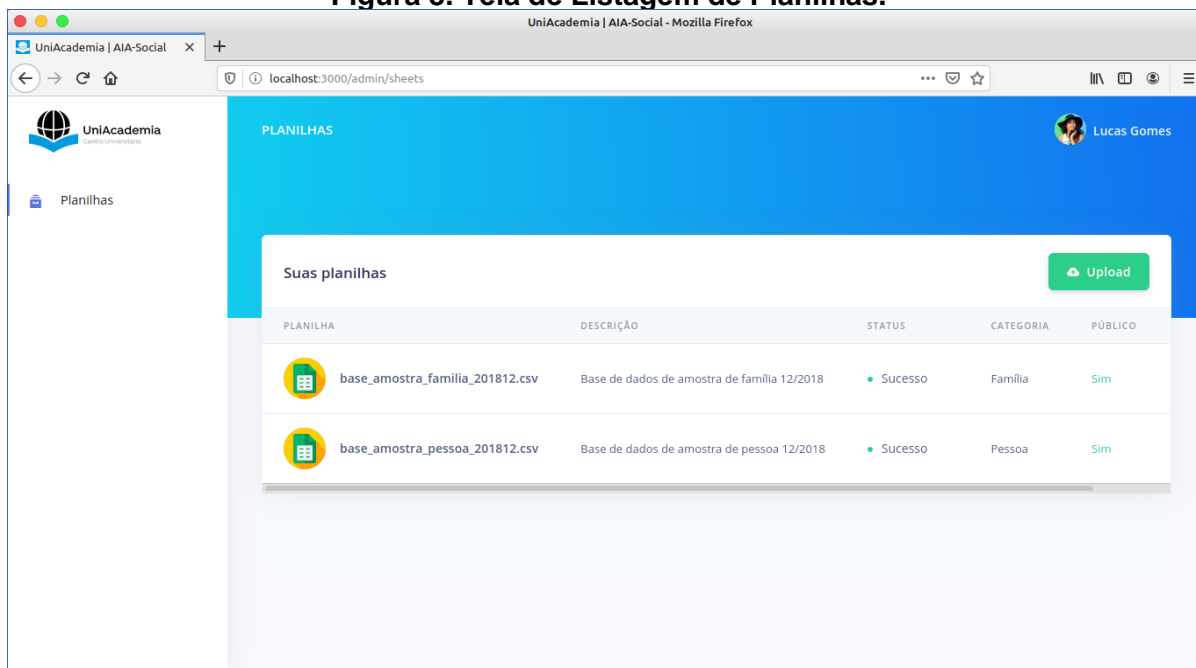


**Fonte: Elaboração Própria.**

### 4.3 Planilhas

A partir do momento que o usuário se autenticou na AIA, será redirecionado para a visualização de todas as planilhas que importou para a ferramenta ou que tenham sido importadas por outros usuários como planilhas públicas, conforme Figura 8.

**Figura 8. Tela de Listagem de Planilhas.**



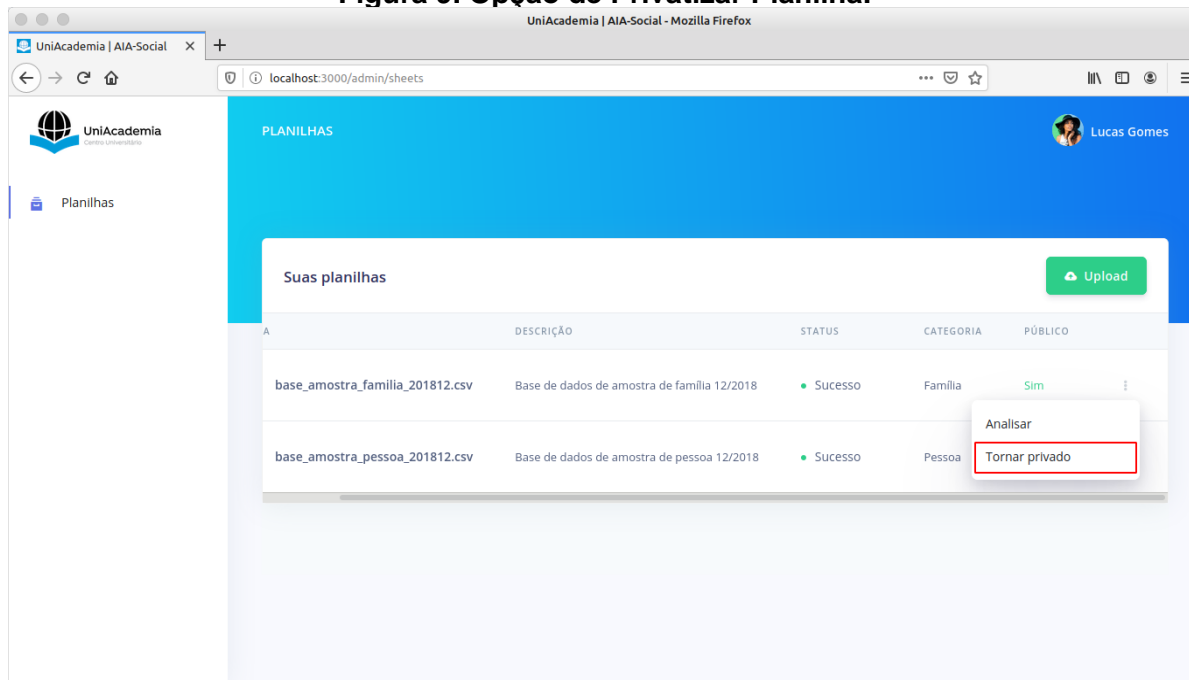
**Fonte: Elaboração Própria.**

A partir dessa tela (Figura 8) o usuário poderá fazer todas as demais ações na ferramenta, conforme os tópicos 4.3.1, 4.3.2, 4.3.3 e 4.3.4.

#### **4.3.1 Privatizar Planilhas**

A partir do momento que o usuário inicia o processo de importação de uma planilha, temos a opção de torná-la pública ou privada. Caso essa planilha tenha sido importada como pública, aparecerá na visualização para todos os outros usuários e, essa funcionalidade visa possibilitar o compartilhamento de informações entre os usuários. Porém, pode ser que em algum momento uma determinada base tenha de ser privatizada por motivos legais ou outras necessidades do usuário e por conta disso, pode privatizar uma dessas planilhas simplesmente clicando no ícone a frente do nome da planilha e em seguida clicar em “Tornar privado”, conforme a Figura 9.

**Figura 9. Opção de Privatizar Planilha.**



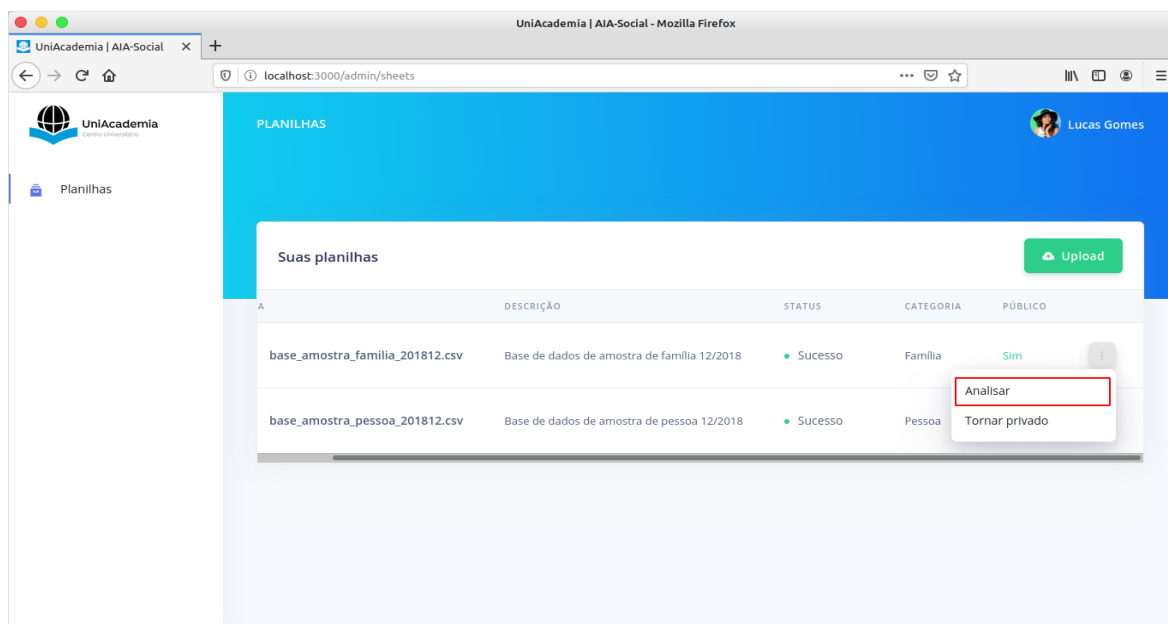
**Fonte: Elaboração Própria.**

Ao fazer isso essa opção somente o usuário importador terá acesso a planilha, e essa opção só está disponível para coleções importadas pelo usuário atualmente logado, planilhas públicas só podem ser privatizadas por seus proprietários.

#### **4.3.2 Analisar Coleções**

Na mesma opção de visualização das planilhas, o usuário também poderá solicitar a mineração dos dados através da opção “Analisar”, essa opção está disponível para todas as coleções que estejam com status igual a “Sucesso”, conforme Figura 10.

**Figura 10. Opção de Analisar Planilha.**

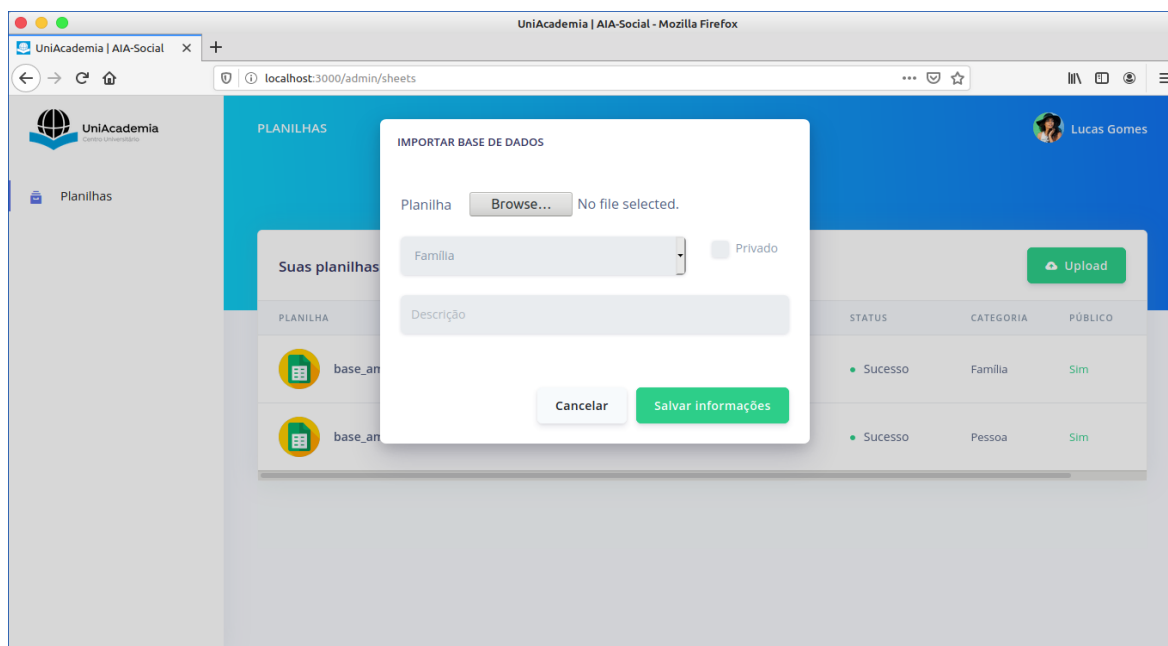


Fonte: Elaboração Própria.

### 4.3.3 Importar Arquivos

A tela de Upload (Figura 11) é uma das mais importantes de toda a aplicação, sendo o primeiro passo para análise de dados. Nessa tela o usuário poderá importar arquivos de extensão CSV, com os dados separados por ponto e vírgula (;).

**Figura 11. Modal de Upload de Arquivos.**



Fonte: Elaboração Própria.

O arquivo CSV será processado pela biblioteca ya-csv com a qual todos os registros serão percorridos e pré-processados a fim de torná-los legíveis para a função do MongolImport. O upload do arquivo será realizado assim que o usuário escolher o arquivo dentro da sua máquina. Logo que o upload for finalizado, os campos categoria, descrição e privado serão habilitados. O processo de importação será iniciado assim que o usuário clicar em “Salvar informações”.

A tabela sheets do PostgreSQL será atualizada e o usuário será redirecionado à visualização das coleções, assim que o processo de importação for finalizado será modificado o status do registro no PostgreSQL e o usuário poderá analisar seus dados.

#### 4.3.4 Filtrar Registros

Ao solicitar a análise de uma coleção, o usuário será redirecionado para o modal de filtragem de registros, essa visualização compõe todos os atributos do primeiro registro da coleção, sendo assim, todos os campos de dados irão aparecer e o usuário poderá selecionar quais deseja utilizar em sua análise.

Caso o usuário tenha interesse, também poderá informar uma condição, conforme Figura 12, até esse momento, a aplicação aceita somente que informe um valor para ser encontrado na base de dados, a fim de reduzir a quantidade de registros retornados. Outro detalhe é que o usuário também pode limitar a quantidade de registros que deseja analisar, em uma escala de quantidade, a qual por padrão foi limitada a 30.000 (trinta mil) registros.

**Figura 12. Modal de Filtro de Dados**



**Fonte: Elaboração Própria.**

#### 4.3.5 Modal de Resultado

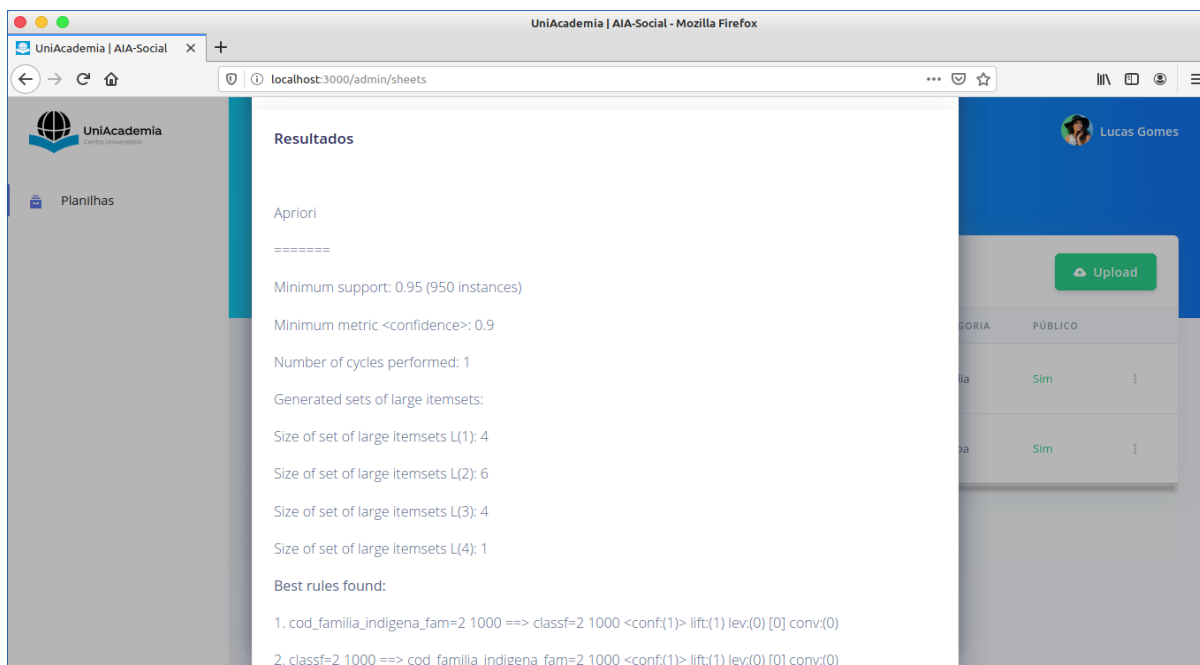
O ponto final do fluxo da aplicação é apresentado na Figura 13, com o resultado da mineração de dados a partir do algoritmo Apriori. Conforme a Figura 13, o usuário poderá visualizar as melhores regras encontradas pelo algoritmo, bem como informações como o



suporte e a confidencialidade do resultado gerado.

A análise dessa informação demanda um conhecimento mínimo sobre o algoritmo Apriori, necessitando de algum conhecimento técnico sobre conceitos de análise e mineração de dados, em casos mais complexos.

**Figura 13. Modal de Resultado.**



**Fonte: Elaboração Própria.**

#### 4.4 Resultados

Neste tópico vamos abordar um dos resultados encontrados no uso da ferramenta, para isso utilizamos uma amostra da base de dados de família disponibilizada pelo Portal da Transparência para o ano de 2018, onde foram aplicados os filtros definidos de acordo com o dicionário da base de dados:

- *cod\_escoa\_sanitario\_domic\_fam*: Forma de escoamento sanitário.
- *cod\_destino\_lixo\_domic\_fam*: Forma de coleta do lixo.
- *cod\_iluminacao\_domic\_fam*: Tipo de iluminação.
- *cod\_calçamento\_domic\_fam*: Tipo do calçamento.
- *marc\_pbf*: Pessoal beneficiário ou não do PBF.

Para essa análise foi selecionada a cidade de Turiaçu no Estado do Maranhão que, segundo Valéria Bretas (2016), era a primeira no ranking dos municípios com piores condições de vida do país no ano de 2016. Para realizarmos essa redução da amostra, informamos para o atributo *cod\_ibge* o valor '2112407' que é o código IBGE para o município, o qual está disponível em nossa base de dados amostral de família.

Para garantir que todos os registros do município sejam analisados, foi solicitada que a quantidade limite de dados seja de 30.000 registros, o qual não foi atingido por conta do número de habitantes do município e por nossa base de dados já ser uma amostra.

Após a realização do filtro e seleção dos atributos para análise, foi solicitada a mineração da base, a qual nos retornou algumas regras úteis, conforme Figura 14.

#### **Figura 14. Resultado da Análise da Amostra.**

1. cod\_escoa\_sanitario\_domic\_fam=4 cod\_destino\_lixo\_domic\_fam=3 marc\_pbf=1 448 ==>  
cod\_calcamento\_domic\_fam=3 428 <conf:(0.96)> lift:(1.1) lev:(0.02) [37] conv:(2.74)
2. cod\_escoa\_sanitario\_domic\_fam=4 cod\_destino\_lixo\_domic\_fam=3 511 ==>  
cod\_calcamento\_domic\_fam=3 487 <conf:(0.95)> lift:(1.09) lev:(0.02) [41] conv:(2.63)
3. cod\_escoa\_sanitario\_domic\_fam=4 590 ==> cod\_calcamento\_domic\_fam=3 555 <conf:(0.94)> lift:(1.08)  
lev:(0.02) [40] conv:(2.11)
4. cod\_escoa\_sanitario\_domic\_fam=4 marc\_pbf=1 514 ==> cod\_calcamento\_domic\_fam=3 483 <conf:  
(0.94)> lift:(1.08) lev:(0.02) [35] conv:(2.07)
5. cod\_destino\_lixo\_domic\_fam=3 marc\_pbf=1 1183 ==> cod\_calcamento\_domic\_fam=3 1110 <conf:  
(0.94)> lift:(1.08) lev:(0.04) [79] conv:(2.06)
6. cod\_destino\_lixo\_domic\_fam=3 1393 ==> cod\_calcamento\_domic\_fam=3 1302 <conf:(0.93)> lift:(1.07)  
lev:(0.05) [88] conv:(1.95)
7. cod\_destino\_lixo\_domic\_fam=3 cod\_iluminacao\_domic\_fam=1 marc\_pbf=1 754 ==>  
cod\_calcamento\_domic\_fam=3 704 <conf:(0.93)> lift:(1.07) lev:(0.03) [46] conv:(1.9)
8. cod\_destino\_lixo\_domic\_fam=3 cod\_iluminacao\_domic\_fam=1 909 ==> cod\_calcamento\_domic\_fam=3  
844 <conf:(0.93)> lift:(1.07) lev:(0.03) [51] conv:(1.77)
9. cod\_escoa\_sanitario\_domic\_fam=3 cod\_destino\_lixo\_domic\_fam=3 marc\_pbf=1 502 ==>  
cod\_calcamento\_domic\_fam=3 465 <conf:(0.93)> lift:(1.06) lev:(0.01) [27] conv:(1.7)
10. cod\_iluminacao\_domic\_fam=3 486 ==> cod\_calcamento\_domic\_fam=3 448 <conf:(0.92)> lift:(1.06)  
lev:(0.01) [24] conv:(1.6)

#### **Fonte: Elaboração Própria.**

Conforme a Figura 14, uma das regras geradas e que pode-se traduzir de forma simplificada é a seguinte: “Para o município informado, temos que, caso o escoamento sanitário seja realizado via vala a céu aberto, o destino do lixo seja queimado ou enterrado na propriedade e a pessoa seja beneficiário do PBF, então provavelmente não possui nenhum tipo de calçamento para chegar até sua residência”.

O objetivo deste tópico não foi de demonstrar uma regra de validade social, neste momento, todavia, foi elucidar que, conforme supracitado, com uma base de dados com informações de diversos nichos, pode-se gerar diversas regras de associações às quais podem ser utilizadas para tomar decisões sociais ou que lhe forem pertinentes.

## **5. CONSIDERAÇÕES FINAIS**

O uso dessa ferramenta para mineração de dados fez-se muito prático para as

necessidades do presente artigo, porém, o consumo de recursos operacionais é alto e o uso em servidores dedicados faz-se necessário. Também pode-se considerar como necessário à aplicação de novos algoritmos de mineração de dados para que usuários com diferentes necessidades sejam atendidas, sendo essas limitações consideradas trabalhos futuros.

Mesmo com os benefícios da ferramenta, foram encontradas algumas dificuldades que fizeram o desenvolvimento da aplicação ser limitado à algumas funcionalidades. O processo de análise e mineração de dados torna-se muito custoso frente a quantidade de registros e a complexidade das bases de dados a serem utilizadas, por conta disso, uma das principais dificuldades encontradas para a AIA foi o uso de recursos operacionais, por conta do consumo de memória da máquina, limitando a JVM (*Java Virtual Machine*) durante o processo de mineração.

Também cabe considerar, que todo o processo de análise, configuração e demonstração realizado no presente trabalho foi feito em uma única máquina, a qual rodou os containers *Docker* do *MongoDB* e *PostgreSQL*, bem como os servidores de aplicação *NodeJS*, Java e *ReactJS*. Distribuir esses servidores em máquinas distintas, pode garantir maior fluidez para a aplicação e reduzir o consumo de recursos da máquina.

Algumas limitações deste trabalho são: i) a quantidade de recursos necessários para rodar a ferramenta AIA-Social; ii) um único algoritmo de mineração de dados; iii) não existir a possibilidade de cruzar bases distintas; iv) a necessidade do pré-processamento do arquivo CSV importado, o qual está limitado somente a bases do portal da transparência; além dos conhecimentos sobre as condicionalidades dos programas sociais a serem analisados.

Como trabalhos futuros, pode-se aprimorar o uso dos recursos operacionais a fim de garantir maior fluidez na análise e mineração dos dados, bem como o incremento de novos algoritmos de mineração que podem tornar a ferramenta ainda mais completa e robusta para o processamento das bases do CadÚnico, que em conjunto são mais complexas, tornando a aplicação útil não somente para nossas necessidades. Ressalta-se que neste trabalho o foco é a ferramenta desenvolvida e espera-se como trabalho futuro a análise das regras de associação encontradas por um especialista.

## REFERÊNCIAS

BARROS, Ricardo Paes de; HENRIQUES, Ricardo; MENDONÇA, Rosane. **Desigualdade e pobreza no Brasil: retrato de uma estabilidade inaceitável**. Revista brasileira de ciências sociais, v. 15, n. 42, p. 123-142, 2000.

BORGELT, Christian; KRUSE, Rudolf. **Induction of association rules: Apriori implementation**. In: Compstat. Physica, Heidelberg, 2002. p. 395-400.

BRASIL. André Luiz Gomes. Secretaria Especial do Desenvolvimento Social (org.). **Dados do Cadastro Único estão disponíveis para pesquisadores**. 2017. Disponível em: <http://mds.gov.br/area-de-imprensa/noticias/2017/novembro/dados-do-cadastro-unico-estao-disponiveis-para-pesquisadores>. Acesso em: 11 jun. 2020.

BRASIL. GOVERNO DO BRASIL. . **Obter Benefícios do Programa Bolsa Família (PBF)**. Disponível em: <https://www.gov.br/pt-br/servicos/obter-beneficios-do-programa-bolsa-familia>. Acesso em: 11 jun. 2020.

---

BRASIL. GOVERNO DO BRASIL. (org.). **Inscriver-se no Cadastro Único**. Disponível em: <https://www.gov.br/pt-br/servicos/inscrever-se-no-cadastro-unico-para-programas-sociais-do-governo-federal>. Acesso em: 11 jun. 2020.

BRASIL. Portal Da Transparência. (Org.). **Inclusão Social Por Meio Do Bolsa Família, Do Cadastro Único**. Disponível em: <http://www.portaltransparencia.gov.br/programas-e-acoes/programa-orcamentario/2019?ano=2019>. Acesso em: 11 jun. 2020.

BRUSSO, Marcos José et al. **Access Miner: uma proposta para a extração de regras de associação aplicada à mineração do uso da Web**. 2000. 96 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000. Disponível em: <https://lume.ufrgs.br/handle/10183/2074>. Acesso em: 14 jun. 2020.

CARMEN NERY. Agência Ibge Notícias. **Extrema pobreza atinge 13,5 milhões de pessoas e chega ao maior nível em 7 anos**. 2019. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/25882-extrema-pobreza-atinge-13-5-milhoes-de-pessoas-e-chega-ao-maior-nivel-em-7-anos>. Acesso em: 11 jun. 2020.

SIRQUEIRA, Tassio Ferenzini Martins et al. **Uma Plataforma de Extração e Análise de Dados de Programas Sociais do Governo Brasileiro**. *Jornal Eletrônico Faculdade Vianna Júnior*, v. 10, n. 1, p. 20-20, 2018.

VALÉRIA BRETAS. *Revista Exame*. **As melhores e piores cidades do Brasil em qualidade de vida**. 2016. Disponível em: <https://exame.com/brasil/as-melhores-e-piores-cidades-do-brasil-em-qualidade-de-vida/>. Acesso em: 14 jun. 2020.