

Desenvolvimento de uma Plataforma para Manipulação e Análise de Dados de Programas Sociais do Governo Federal

Savio de Paiva Macedo¹, Tassio Ferenzini Martins Sirqueira¹

¹ Centro de Ensino Superior de Juiz de Fora (CES/JF) – Juiz de Fora – MG – Brasil

saviom.cedo@outlook.com.br, tassio@tassio.eti.br

Resumo. *O trabalho desenvolvido tem como objetivo auxiliar os profissionais da área de economia na coleta e análise dos dados liberados pelo Governo Federal, no âmbito do Programa de Dados Abertos (PDA), focando mais especificamente nas bases dos programas sociais vinculados ao Cadastro Único (CadÚnico). A plataforma desenvolvida, foi denominada de “Social Program Analysis Platform” ou simplesmente “PSPA”, e busca ser uma solução que facilite a coleta e descoberta das informações das bases do PDA por economistas, fornecendo filtros passíveis de customização e adaptados aos dados a serem manipulados dentro da plataforma. Com isso, busca-se com essa solução agilizar e dar mais confiabilidade na extração de amostras para estudos, além de otimizar o processo que em muitos casos é feita de forma manual. Além disso, como a base de dados é bastante extensa e fornecida na maioria dos casos em formato de texto simples ou CSV, seu uso pela maioria das ferramentas de estatística demanda um trabalho extra, dado que esses softwares estatísticos são genéricos e não oferecem os recursos de coleta e armazenamento dos dados em sistemas gerenciadores de banco de dados.*

Abstract. *The objective of this work is to assist economics professionals in the collection and analysis of data released by the Federal Government, within the scope of the Open Data Program (PDA), focusing more specifically on the bases of social programs linked to the “Cadastro Único” (CadÚnico). The platform developed, was called “Social Program Analysis Platform” or simply “PSPA”, and seeks to be a solution that facilitates the collection and discovery of PDA database information by economists, providing customizable and data-tailored filters, manipulated within the platform. This solution aims to speed up and give greater reliability in the extraction of samples for studies, and optimize the process that in many cases is done manually. Also, because the database is quite large and provided in most cases in plain text or CSV format, its use by most statistical tools requires extra work, as these statistical software are generic and do not offer the features data collection and storage in database management systems.*

1. Introdução

Na área de ciências econômicas está surgindo o campo denominado economia computacional, que explora da interseção da economia com a computação, conforme Sirqueira *et al.* (2018).

O objetivo deste trabalho é o de apresentar uma plataforma que pode ser acessada por navegadores de internet e que tenha como finalidade facilitar a importação, filtro, análise e organização dos dados disponibilizados pelo Governo Federal através do Programa de Dados Abertos (PDA), apoiando a economia computacional.

Em especial, a plataforma desenvolvida irá trabalhar com os dados liberados de programas sociais baseados no Cadastro Único (CadÚnico). Esta plataforma nasceu a partir da análise das dificuldades encontradas pelos pesquisadores de economia da Universidade Federal de Juiz de Fora (UFJF), em coletar e analisar os dados liberados pelo Governo Federal, visto que normalmente são bases de dados em formato texto ou CSV, extensas e que demandam uma enorme quantidade de recursos e esforço para análise dos dados.

Segundo Sosa Eскурdero (2017), a revolução do Big Data está mudando radicalmente a forma como os dados são produzidos, gerenciados, armazenados e analisados pelos economistas, tornando mais acessíveis abordagens estatísticas e computacionais comumente utilizadas em outros campos, mas pouco exploradas na economia. Além disso, Sirqueira & Dalpra (2018) abordam que os desafios do Big Data vão além, devido o volume de dados a ser analisado, a velocidade necessária para o processamento dos mesmos e em muitos casos a falta de estruturas fixas dos dados, dificultando a extração de informações úteis.

Neste contexto, esta plataforma, cujo nome dado foi “*Social Program Analysis Platform*” (PSPA), tem como finalidade o de otimizar o tempo na realização das tarefas, auxiliando no tratamento dos arquivos até a análise dos dados, com base nos arquivos CSV já prontos, para a utilização dos pesquisadores.

Para o desenvolvimento desta plataforma, se fez necessário o estudo de ambientes multiplataforma e que tivessem que ser já preparados para um alto escalonamento de informação. Com base nisso, foi escolhida como plataforma de programação o *.NET Core*, pois ele possui suporte para uso e desenvolvimento nos ambientes *Windows, Linux e MacOS*.

Para armazenar informações da aplicação, foi empregado o sistema gerenciador de bando de dados *MySQL*, por ser multiplataforma e gratuito. Além do *MySQL*, para a manipulação das informações, foi utilizado o banco de dados não relacional *ElasticSearch*, pois este tem como premissa o de fornecer uma performance de leitura e busca otimizadas, mesmo quando escalonamos com informações de *Big Data*.

Além desta introdução, na seção 2 apresentaremos algumas informações dos programas sociais do Governo Federal e das tecnologias utilizadas no desenvolvimento desta plataforma. A seção 3 apresentamos a arquitetura e os detalhes da plataforma *PSPA*. Já na seção 4 abordaremos algumas limitações, nossas considerações finais e os trabalhos futuros.

2. Referencial Teórico

Neste capítulo iremos abordar de forma geral temas que estão diretamente relacionados ao desenvolvimento da aplicação em si. Assuntos estes que necessitam de um entendimento básico para entender os motivadores para a construção do software.

2.1 Cadastro Único para Programas Sociais

O Cadastro Único (CadÚnico) para Programas Sociais, é uma plataforma voltada para identificação e caracterização socioeconômica das famílias brasileiras consideradas de baixa renda, onde os programas sociais do Governo Federal voltados ao atendimento deste público devem, obrigatoriamente, utilizá-lo para a seleção e integração de beneficiários em tais programas.

Desde a sua criação em 2001, o CadÚnico tinha como objeto o de integrar os programas sociais do Governo Federal para concentrar esforços nas famílias que possuem um perfil com características e necessidades comuns. Mas por falta de definições sobre o público-alvo, regras, além da dificuldade de manter a atualização e manutenção dos dados, esta integração não aconteceu de forma imediata (WWP, 2016).

O CadÚnico passou a ser consolidado como uma plataforma de integração a partir do Programa Bolsa Família (PBF), em 2003. O PBF é um programa de transferência de renda do Governo Federal que tem como objetivo o de unificar e ampliar diversos programas de transferência de renda, tais como: Bolsa Escola; Bolsa Alimentação; Auxílio-Gás e Fome Zero (FACIROLI & SIRQUEIRA 2017).

Atualmente o CadÚnico possui informações socioeconômicas de cerca de 26 milhões de famílias, e está base vem sendo utilizada por mais de 20 programas sociais. A partir da imensa massa de dados que é gerada, e com a finalidade de permitir fomentar estudos de cunho social, o Governo Federal vem liberando, desde 2017, informações sobre os programas que compõem o CadÚnico no Portal Brasileiro de Dados Abertos (PDA). (BRASIL, 2017).

O PDA é uma plataforma construída pelo Governo para centralizar a busca e o acesso dos dados e informações públicas. A mesma foi construída para atender a Lei de Acesso à Informação Pública (Lei 12.527/2011), que teve como objetivo o de regular o acesso a dados e informações detidas pelo Governo, um marco para a democratização da informação pública e que tem como um dos requisitos o de que a informação solicitada pelo cidadão deve seguir critérios tecnológicos alinhados com as “3 Leis de Dados Abertos”.

As “3 Leis de Dados Abertos” que regem o PDA, são leis propostas pelo especialista em TI David Eaves, onde a organização que padroniza a criação e interpretação de conteúdos para Web (W3C) à adotou.

As três leis dizem que:

- Se o dado não pode ser encontrado e indexado na Web, ele não existe;
- Se não estiver aberto e disponível no formato compreensível por máquina, ele não pode ser reaproveitado; e
- Se algum dispositivo legal não permitir sua replicação, ele não é útil.

Os dados divulgados pelo PDA são os principais motivadores para a criação deste projeto, uma vez que os mesmos são divulgados apenas para atenderem a legalidade da transparência, mas não existe um cuidado para que eles possam ser analisados por qualquer pessoa não técnica.

2.2 Tecnologias Utilizadas

A plataforma desenvolvida foi criada sobre o *.NET core* e para que possamos entendê-la, é necessário que seja apresentado o *.Net Standard*, que nada mais é do que “uma especificação formal de *APIs* do *.NET*”, cujo objetivo é o de “estabelecer maior uniformidade” no ecossistema de software.

As diversas implementações do *.NET* se destinam a versões específicas do *.Net Standard*. Cada *Framework* pode conversar entre si desde que suportados pelas versões da especificação aos quais foram construídos, tendo suporte a retro compatibilidade.

Já o *.NET Core* “é uma plataforma de desenvolvimento de código aberto de uso geral mantida pela *Microsoft* e pela comunidade *.NET* no *GitHub*.” (MICROSOFT, 2019). A *Microsoft* ainda define como sendo uma “plataforma cruzada”, compatível com *Windows*; *Linux* e *MacOS*, e que pode ser usada no desenvolvimento de software na nuvem e em aplicativos de IoT, conceito que trata qualquer objeto físico como “coisa” ao qual estão conectados e se comunicando entre si e com o usuário (PROOF, 2016).

O *.NET Core* lançada em 2014 foi desenvolvido com o apoio da comunidade de forma a transformar esta nova plataforma em *Open Source*, ou seja, o código fonte era aberto para que pessoas possam olhar sua estrutura, estudá-la e melhorá-la (MICROSOFT, 2019).

Com o objetivo de ser multiplataforma, o *.Net Core* foi completamente construído e fiel ao padrão *.Net Standard* dando suporte as seguintes plataformas e recursos:

- *Desktop (Somente Windows)*
- *Web – Asp.Net (Rodando em cima do Windows, Linux ou MacOS)*
- *Dispositivos Móveis – Xamarin*
- *Cloud – Integração com a Azure*
- *Desenvolvimento de Jogos – Unity*
- *IoT*
- *Inteligência Artificial – ML.NET*

Além de oferecer um novo leque de ferramentas de desenvolvimento:

- *Visual Studio (Windows)*
- *Visual Studio for MAC (MacOS)*
- *Visual Studio Code (Windows, MacOS e Linux)*
- *Command Line Interface (Windows, MacOS e Linux)*

Outra novidade com o *.NET Core*, foi que junto a plataforma, surgiu suporte nativo para que as aplicações possam ser executadas em *containers Docker*.

O *Docker* nada mais é do que um software *contêiner* construído pela empresa *Docker Inc*. Esta ferramenta faz uso da tecnologia de contêineres, cujo objetivo é o de oferecer um tempo de *deploy* de uma aplicação ou infraestrutura, mais otimização com relação aos métodos tradicionais. Os containers compartilham o *kernel* do SO hospedeiro e também algumas bibliotecas e binários.

Para cada aplicação ou serviço, os alocamos em diferentes contêineres e utilizamos o *docker-compose* para gerenciá-los. O *Docker-Compose* é uma ferramenta para definição e execução de múltiplos containers *Docker*. Com ela nós definimos os

parâmetros necessários para executar cada contêiner a partir de um arquivo de configuração (*docker-compose.yml*) os instanciamos. Dentro deste arquivo de definição, apontamos cada container como sendo um serviço e especificamos quais volumes e rede serão criados para serem utilizados.

Nesse projeto utilizamos a tecnologia do *Docker-Compose* para podermos gerenciar os serviços da aplicação *PSPA* e dos bancos de dados *MySQL* e *ElasticSearch*.

O *ElasticSearch* é um sistema gerenciador de banco de dados que foi construído para trabalhar com grandes volumes de dados, e vem sendo uma poderosa ferramenta para realização de buscas e análise dos dados, onde sua principal particularidade é a de indexar documentos e realizar buscas em tempo real.

O *ElasticSearch* é multiplataforma, baseado no *Apache Lucene* e feito pela empresa *Elastic*. Ele habita na categoria de bancos de dados não relacionais, compatível com qualquer ambiente multiplataforma e facilmente integrável, a ferramenta trabalha como um mecanismo de busca e análise, sendo interagida através de chamadas *RESTful*.

Não é objetivo desse trabalho entrar em detalhes quanto as diferenças de funcionalidades de um banco relacional a um banco não relacional, nem em relação a sua estrutura, mas como principal característica motivadora de se utilizar um banco não relacional, foi basicamente o potencial de escalonamento das informações e da forma com que estes bancos trabalham com tal questão sem que haja perda da performance.

Conforme podemos observar na Figura 1, um banco relacional ao lidar com poucos dados escalonados se dá melhor do que um não relacional, por tanto nem sempre este último vem a ser a melhor opção e para o *PSPA*, optamos por trabalhar com os dois tipos de bancos de dados para funções distintas, sendo o *MySQL* para armazenar informações da aplicação que são altamente tratadas e padronizadas e o *ElasticSearch* para os dados advindos do PDA, uma vez que este banco lida melhor com informações mais “voláteis”.



Figura 1: Relação Banco de Dados vs Escalabilidade, adaptado de Bauermeister (2019).

3. A Plataforma *PSPA*

Como o próprio nome diz o *Social Program Analysis Platform (PSPA)*, é um software construído para ser uma plataforma de manipulação e análise de dados de programas sociais do Governo Federal. Este sistema foi concebido para poder importar os dados, e a partir das bases importadas permitir o usuário escolher uma base e personalizar com filtros os dados a serem consultados, das informações previamente importadas. Sendo um intermediário entre um trabalho técnico a ser feito e o usuário final que não possui este conhecimento.

Dentre as entidades da plataforma nós temos, o usuário final; o sistema *PSPA* propriamente dito; um banco de dados *MySQL* e uma instância de base de dados não relacional do *ElasticSearch*. Um diagrama contendo a interligação entre estas entidades está representado pela Figura 2.

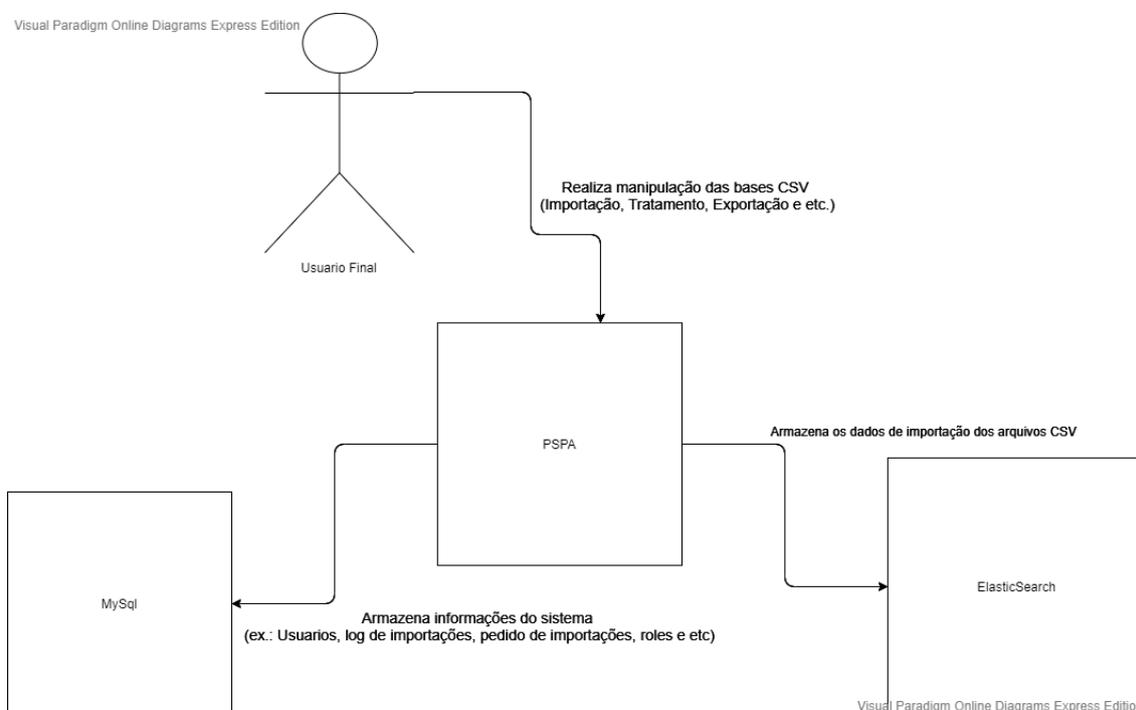


Figura 2: Diagrama do Sistema.

O *PSPA* foi construído em cima da plataforma do *.NET Core* sendo executado em cima de um container do *Docker*, sendo assim, segue a característica *Cross-Platform Software*, ou seja, o sistema consegue ser implantado em diferentes ecossistemas (*Windows, Linux ou MacOS*). O sistema *PSPA* pode ser executado através do arquivo *docker-compose.yml*.

No arquivo *docker-compose.yml*, conforme a Figura 3, será executado um container contendo o banco de dados *MySQL* e em seguida outro com a aplicação *PSPA*, a ordem de execução é importante pois a aplicação depende do banco para funcionar.

```

version: '3.4'

services:
  db:
    image: mysql:5.5.62
    restart: always
    volumes:
      - dbdata:/var/lib/mysql
      - ./_MySQL_Init_Script:/docker-entrypoint-initdb.d
    environment:
      MYSQL_ROOT_PASSWORD: root
    ports:
      - 3306:3306

  webapi:
    image: ${DOCKER_REGISTRY-}webmvcnet
    build:
      context: .
      dockerfile: WebMVCNET/Dockerfile
    depends_on:
      - db
    environment:
      - DBHOST=db

volumes:
  dbdata:

```

Figura 3: Instruções do arquivo *docker-compose.yml*.

Após o início da aplicação e do banco de dados *MySQL* pelo *docker-compose.yml*, é necessário que inicie também o serviço de banco de dados não relacional do *ElasticSearch*, ele também tem seu funcionamento através do *Docker*. Este banco de dados tem como objetivo o de armazenar as informações importadas a partir do CSV enviado pelo usuário para que seja mais rápida e fácil a leitura dos dados do arquivo.

O software permite que o usuário realize três grupos de operações com suas respectivas ações secundárias, sendo elas:

- *Administração do sistema:*
 - *Manutenção de roles;*
 - *Manutenção de usuários;*
- *Gerenciamento de importação de dados:*
 - *Gerenciar ordem de importações;*
 - *Solicitar nova ordem de importação.*
- *Consulta e análise de dados;*

Para controlar as funcionalidades do sistema, é utilizado o conceito de controle de acesso baseado em funções (*role-based security* ou somente *roles*). Este paradigma de segurança diz que as permissões dos usuários são concedidas com base na sua função

na empresa ou sistema, ou seja, não possuímos um único perfil em que possua todas as funcionalidades do sistema, mas sim um usuário que agrupa todas as funções sob ele, portanto para que um usuário possa usufruir de todas as funcionalidades do sistema o mesmo deve possuir todas as *roles*.

Para estas ações pré-definidas no sistema, ele possui três tipos de *roles* com suas respectivas permissões, conforme podemos verificar na Tabela 1.

Tabela 1: Relação de permissão funcionalidade x role.

Funcionalidade	<i>Queryable</i>	<i>Manager</i>	<i>Administrator</i>
Gerenciar ordem de importações		✓	
Solicitar nova ordem de importação		✓	
Manutenção de roles			✓
Manutenção de usuários			✓
Consulta de dados	✓		

A opção de solicitar uma nova ordem de importação, o usuário, tendo a role de *manager*, requisita para que o sistema inicie um novo processo de gravação de um arquivo CSV na base do *ElasticSearch*.

Os arquivos CSV a serem gravados no sistema devem atender aos seguintes requisitos para que possam ser gravados com sucesso na base de dados:

- *Os dados do CSV devem vir separados pelo caractere “;”;*
- *Os arquivos CSV, na sua primeira linha, deve vir com o cabeçalho dos dados informados;*
- *O arquivo deve vir salvo sob a codificação ISO-8859-1, padrão das bases emitidas pela base do governo federal*
- *Caso esta base de dados já exista no sistema, o cabeçalho do arquivo deve ser exatamente igual aos anteriores já importados, para continuidade de inserção na mesma base.*

Após realizar a solicitação de uma nova importação por meio do formulário que pode ser visto na Figura 4, deve-se informar um nome existente de uma base de dados, onde caso este último não exista será criada uma nova, e a url de onde o sistema irá baixar estes dados.

Request import order

Fill in the data below to request to import a new database into the system.

Database Name

File Address

Request Import Order

Figura 4: Formulário de requisição de ordem de serviço.

Submetendo o formulário da Figura 4, o usuário irá ter uma nova ordem de importação registrada, onde ele poderá acompanhar os *logs* das etapas na tela de gerenciamento de ordem de importações. Utilizando outros threads do sistema, o *PSPA* irá orquestrar as próximas atividades do sistema que são: i) baixar arquivo .zip informado na url, ii) extrair os arquivos, iii) ler e gravar estes dados na base de dados do *ElasticSearch* e iv) registrar as etapas nos *logs* do bando de dados *MySQL*.

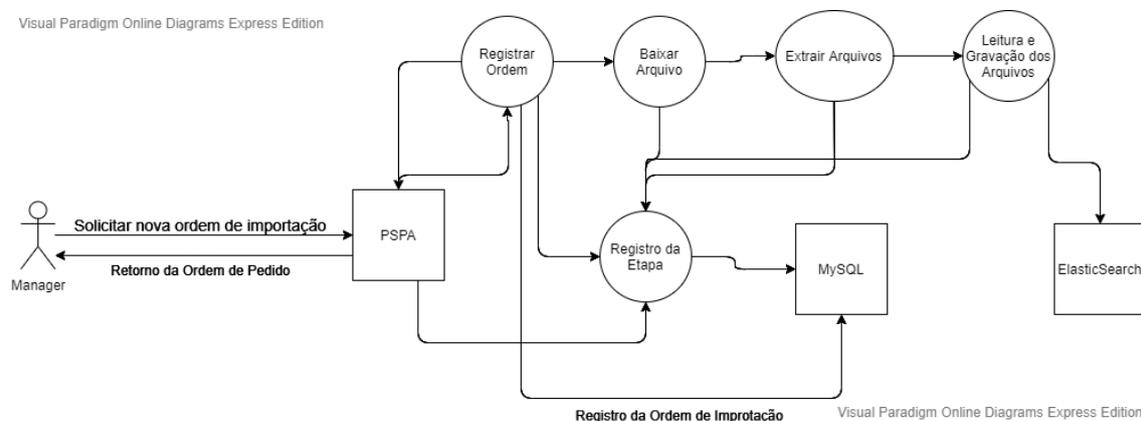


Figura 5: Diagrama de solicitação de nova ordem de importação.

Todo o processo de importação de uma ordem é executado em uma *thread* específica para aquela ordem, de forma que todas as etapas são feitas de forma síncrona, como podemos verificar na Figura 5. Durante o processo de leitura e gravação dos arquivos, foi construído sob a lógica de que o sistema lê somente duzentas (200) linhas por vez e as armazena no *ElasticSearch* de forma a manter a estabilidade no sistema e regular o consumo de memória RAM por parte da aplicação, uma vez que as bases de dados em CSV possuem muitos registros e consequentemente podem gerar um grande gargalo no servidor caso o mesmo armazene toda esta informação em sua memória de acesso temporário.

Após a realização de uma nova ordem e importação, desde que o usuário possua a role de *manager*, ele poderá acessar uma região do sistema onde ele irá conseguir verificar todas as ordens de importação feitas no sistema (Figura 6), podendo então

verificar o status final da mesma e sua atualização de dez em dez segundos, em que etapa do processo de importação a sua ordem se encontra (Figuras 7 e 8).

PSPA - Data Analysis Home Hello SavioMacedo! Query Data System Data Logout

Import Orders by SavioMacedo

Show entries Search:

Order	Base	State	End Date	
21	bolsafamiliaTemp	Concluido		
22	bolsafamiliaTemp	Aguardando		
23	bolsafamiliaTemp	Aguardando		
24	bolsafamiliaTemp	Aguardando		
25	bolsafamiliaTemp	Aguardando		

Showing 21 to 25 of 28 entries First Previous 1 2 3 4 5 6 Next Last

© 2019 - PSPA - Data Analysis - Privacy

Figura 6: Tela de verificação das ordens de importação.

Ao clicar no botão com ícone de “lupa” conforme a Figura 6, o usuário consegue acessar detalhadamente os processos que estão sendo realizados com a sua ordem. Em todas as tabelas do sistema ele consegue ordenar, paginar de acordo com um número de registros e alternar entre elas.

PSPA - Data Analysis Home Hello SavioMacedo! Query Data System Data Logout

Logs for import order 28

Show entries Search:

ID	Message	State	Creation Date
6277	Aguardando processo de download iniciar...	Informativo	2019-08-31T01:45:10
6278	Criando diretorio de extração...	Informativo	2019-08-31T01:45:18
6279	Realizando download do arquivo inicial.zip...	Informativo	2019-08-31T01:45:18
6280	Progresso do Download: 10	Informativo	2019-08-31T01:46:46
6281	Progresso do Download: 20	Informativo	2019-08-31T01:48:13

Showing 1 to 5 of 22 entries First Previous 1 2 3 4 5 Next Last

© 2019 - PSPA - Data Analysis - Privacy

Figura 7: Tela de log de uma importação em específico.

Logs for import order 28

Show entries Search:

ID	Message	State	Creation Date
6292	Arquivos extraídos e sendo encaminhados para a fase de análise	Informativo	2019-08-31T02:01:46
6293	Listando arquivos baixados no disco...	Informativo	2019-08-31T02:02:35
6294	Registrando novos arquivos para leitura...	Informativo	2019-08-31T02:03:15
6295	Iniciando conexão com Elasticsearch...	Informativo	2019-08-31T02:03:41
6296	Iniciando leitura do arquivo 201901_BolsaFamilia_Pagamentos.csv...	Informativo	2019-08-31T02:04:12

Showing 16 to 20 of 22 entries First Previous 1 2 3 4 5 Next Last

Figura 8: Tela de log de uma importação em específico, ordenada pelo ID.

Na Figura 9 podemos verificar todo o processo diagramado do gerenciamento das ordens de importação.

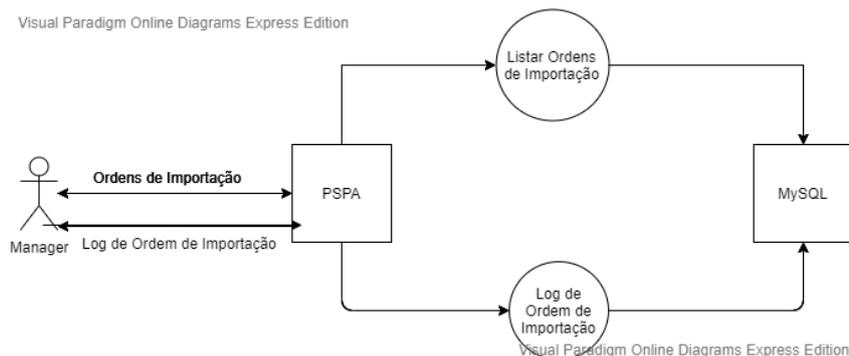


Figura 9: Diagrama do gerenciamento de ordens de importação.

Após terminar a importação do arquivo, o usuário, desde que de posse da role de *queryable*, poderá entrar na tela de realizar buscas para consultar as bases de dados importadas no sistema, ao entrar na tela o usuário, passa a ter a visão representada pela Figura 10, onde o mesmo pode identificar as bases de dados importadas no sistema, verificar as colunas que a mesma possui (Figura 11), aplicar filtros e outras configurações para aquela base, onde posteriormente poderá retornar os dados daquela consulta via CSV ou representada em gráficos.

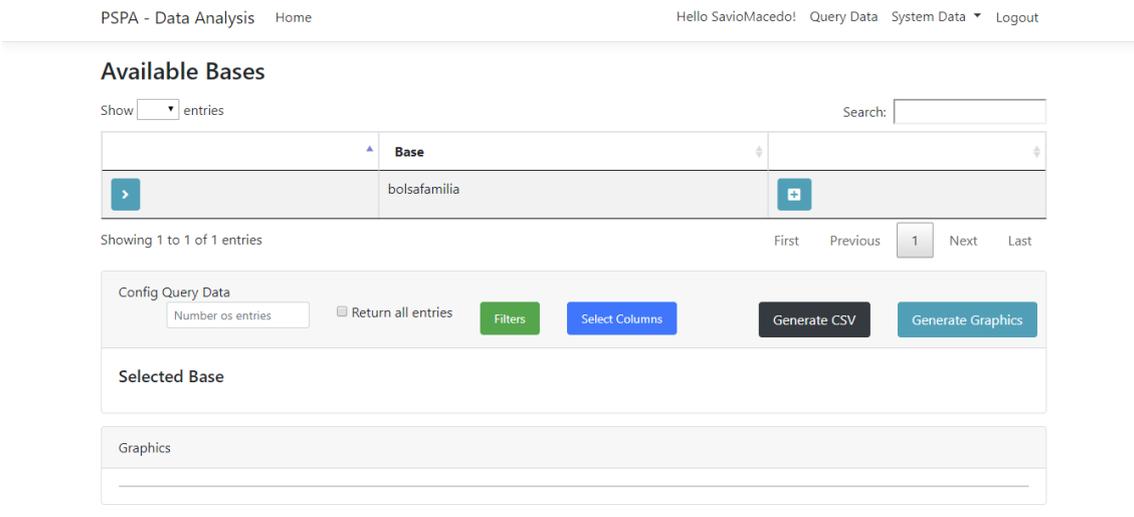


Figura 10: Tela de consulta de dados inicial, vista pelo usuário.

Após ele selecionar a base a ser consumida, o usuário pode informar a quantidade de registros que deseja receber ou marcar a opção de retornar todos os dados, Figura 12. Seguindo com o menu de filtros, no botão “*Filters*” o usuário poderá selecionar uma coluna para que sejam aplicados os seguintes filtros:

- *Maior que;*
- *Maior ou igual que;*
- *Menor ou igual que;*
- *Menor que; ou*
- *Igual a.*

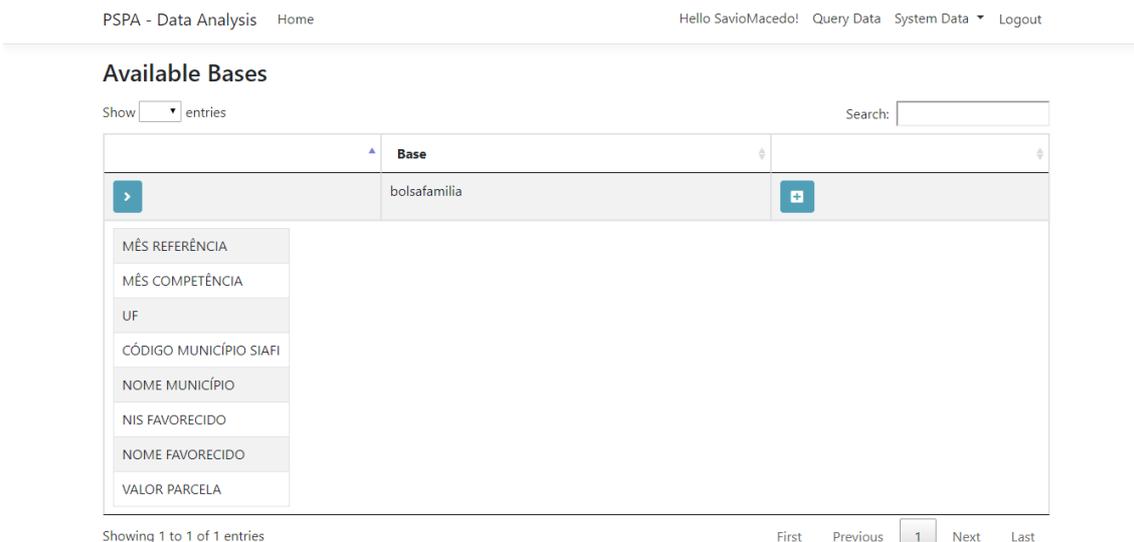


Figura 11: Listagem das colunas da base, acionada pelo usuário.

No filtro do botão “*Select Columns*” o usuário poderá selecionar quais coluna que deseja que sua busca tenha como retorno. Para o tipo de execução de geração de gráficos, deve-se necessariamente informar uma coluna no filtro de “*Select Columns*”, conforme Figura 13.

Nas visões seguintes, temos a possibilidade de geração de um arquivo CSV para exportar os dados para outras aplicações, tais como ferramentas estatísticas ou o botão para geração de gráficos, conforme a Figura 14, onde neste último temos as seguintes opções de representações gráficas:

- *Gráfico de Pizza;*
- *Gráfico de Barra;*
- *Gráfico de Colunas;*
- *Histograma;* e
- *Tabela*

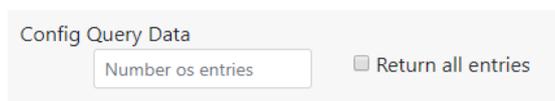


FIGURA 12: Filtro de quantidade de registros a serem retornados

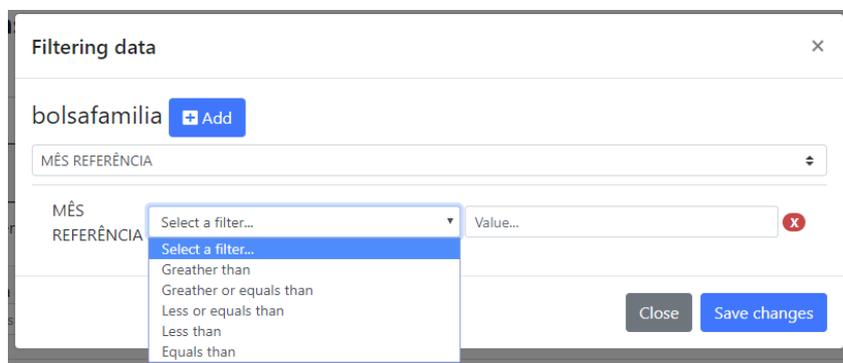


FIGURA 13 Tela para filtragem de valores de dados

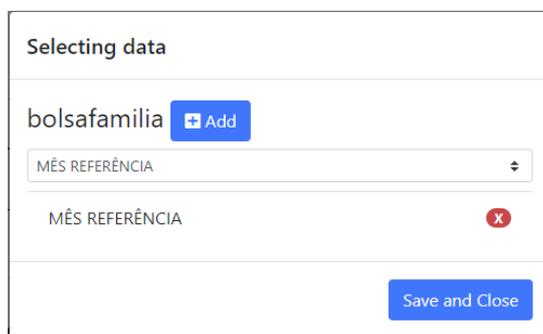


FIGURA 14: Tela para selecionar quais colunas devem retornar nos dados pesquisados

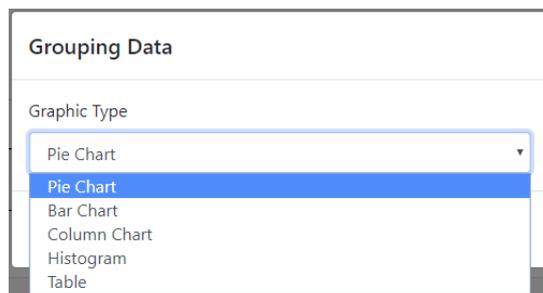


FIGURA 15: Tela após acionar execução do botão de consultar dados nos gráficos

Na Figura 16 podemos ver o exemplo de um relatório utilizando o gráfico de barras, gerado a partir da importação da base “bolsafamilia” (Figura 16), onde foram aplicados os filtros de “UF” igual a “CE” e selecionada para visualização a coluna “Nome Município”, onde temos o total de beneficiários por cada cidade do estado.

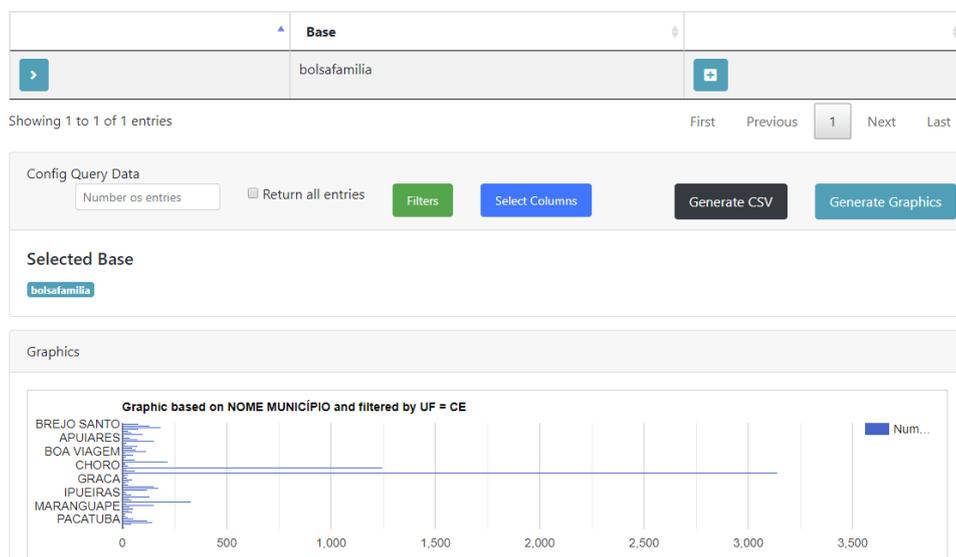


FIGURA 16: Tela de consulta de dados para os gráficos

Na Figura 17 podemos ter uma visão detalhada dos processos feitos nas requisições dos usuários para a realização de uma consulta através do sistema.

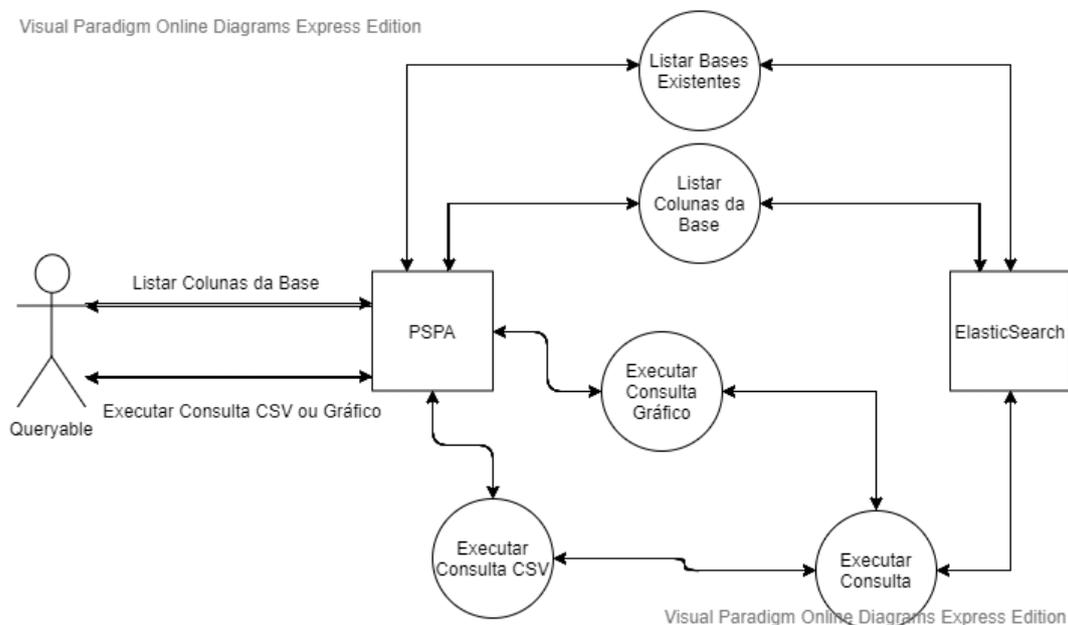


FIGURA 17: Diagrama de execução de consultas pelo usuário

Para os usuários de role “*administrator*”, eles podem ter a visão de todos os usuários cadastrados no sistema e a partir deles, deletar, visualizar ou editar suas informações (Figuras 18 e 19), tais como:

- *Username*;
- *First Name*;
- *Second Name*;

- *E-mail*;
- *Roles*

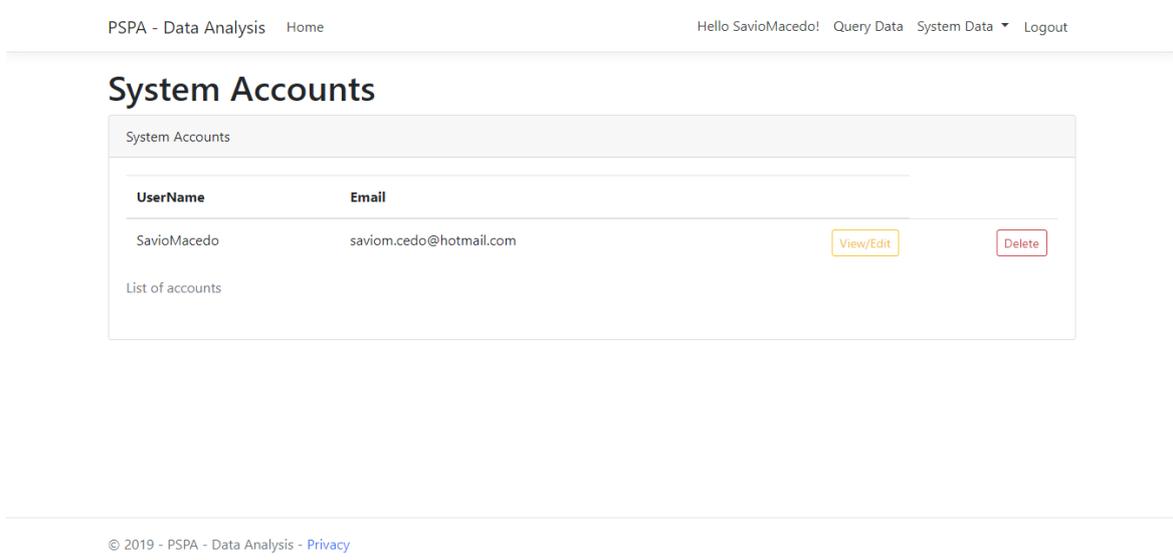


FIGURA 18: Listagem de usuários do sistema

The screenshot shows the 'Edit User' form. It is divided into several sections:

- First Name:** A text input field containing 'Savio'.
- Second Name:** A text input field containing 'de Paiva Macedo'.
- Authentication Information:** A section containing an 'Email' input field with 'saviom.cedo@hotmail.com' and a blue 'Update Account' button.
- Roles Information:** A section with a dropdown menu and a blue 'Add Role' button.
- Role List:** A table with two columns: 'Name' and 'Role'. It lists 'Administrator' and 'Queryable', each with a red 'Remove Role' button to its right.

FIGURA 19: Edição de usuário do sistema.

A plataforma PSPA encontra-se disponível para *download* em <https://github.com/ces-jf/PSPA>. Também podemos encontrar um vídeo demonstrando em tempo de execução as operações acima descritas em <https://www.youtube.com/watch?v=arzkpOjGqms>.

4. Considerações Finais

Observando os problemas motivadores para a criação da ferramenta, podemos concluir que a TI é cada vez mais uma área multidisciplinar e que, como exemplo deste projeto, podemos otimizar e melhorar as atividades fins de diversas outras áreas. Neste caso, os pesquisadores demandam muito tempo de suas pesquisas para baixar e extrair amostras para uso nas ferramentas estatísticas, até mesmo para responder a questões simples de pesquisa. Então podemos através da plataforma proposta diminuir o tempo

de execução destas tarefas, feitas pelos pesquisadores, além de ampliar o leque de detalhes em que os mesmos podem adentrar.

Com relação as tecnologias empregadas, plataforma *.Net Core*; *Docker*, *MySQL* e o *ElasticSearch*, são tecnologias atuais e que podem apresentar um mundo muito mais amplo ao aprofundar em seus usos. Destacamos que o *ElasticSearch* neste trabalho foi uma quebra de paradigma, pois havia-se um certo receio com relação ao uso de uma tecnologia de bancos de dados não relacional em memória, devido a demanda computacional.

Destacamos neste projeto que toda nova tecnologia lançada, tal como o *ElasticSearch*, tem como propósito resolver algum problema em específico, portanto, ela pode ser apenas um mero “modismo” no momento ao qual a conhece, mas a mesma nasceu com um objetivo em mente. Portanto nem sempre por ser uma tecnologia nova ela deve realmente ser utilizada impreterivelmente, assim como não é porque ela é a tecnologia “da moda” que ela é inútil e devemos descartá-la, neste caso, as tecnologias escolhidas atenderam as demandas e superaram as expectativas.

Durante o desenvolvimento da aplicação, foram encontrados diversos problemas que atrasaram e dificultaram a maneira de construir a mesma. Como a proposta da ferramenta é facilitar o trabalho com grandes bases de dados e na falta de recursos para construir a mesma, montar o cenário com os dois bancos de dados mais a aplicação executando na mesma máquina dificultou bastante o trabalho a ser realizado. Outro problema a ser encontrado foi que o mesmo foi construído em cima de um ambiente Windows e simulado no mesmo e nem todas as funções disponíveis no *.Net Core* são igualmente chamadas e funcionais nos ecossistemas Windows e Linux, a solução para o mesmo foi passar a simular a execução da aplicação sempre em um Docker Linux.

O software *PSPA* ainda pode ser melhorado com novas formas de análise de dados já disponíveis no mercado, tais como *machine learning*, *ontologias* e novas visualizações.

Como trabalhos futuros, pode-se estender as seguintes funcionalidades:

- *Junção entre base de dados diferentes, com isso podemos por exemplo trabalhar com a base de dados do Bolsa Família junto com a base de dados de Erradicação do Trabalho Infantil, fazendo cruzamento de dados;*
- *Otimizar o tempo de gravação dos dados do arquivo CSV na base do ElasticSearch;*
- *Abstrair os processos de armazenamento de dados para uma arquitetura de Micro Serviços, permitindo a integração direta com outras ferramentas ; e*
- *Agendar a busca e extração de arquivos CSV periodicamente disponibilizados no PDA, com notificação via e-mail.*

Nosso planejamento é que o desenvolvimento da ferramenta continue ativo e que a mesma seja melhorada e incrementada ainda por várias versões, queremos que ela seja de fato uma plataforma que auxilie significativamente os seus usuários.

Referências

- BAUERMEISTER, Rylan. **When to Drop SQL**. Disponível em: <https://medium.com/swlh/when-to-drop-sql-a7cc1b9ab2e5>. Acesso em: 09 de nov. 2019.
- BRASIL. MINISTÉRIO DO DESENVOLVIMENTO SOCIAL. **Dados do Cadastro Único estão disponíveis para pesquisadores**. Disponível em: <http://mds.gov.br/area-de-imprensa/noticias/2017/novembro/dados-do-cadastro-unico-estao-disponiveis-para-pesquisadores>. Acesso em: 10 de nov. 2019.
- FACIROLI, Jéssica; SIRQUEIRA, Tassio Ferenzini Martins. Análise espacial da taxa de cobertura do Programa Bolsa Família para os municípios brasileiros. **Desenvolvimento Regional em debate: DRd**, v. 7, n. 2, p. 118-137, 2017.
- MICROSOFT. **Sobre o .NET Core**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/core/about>. Acesso em: 25 de out. 2019.
- PROOF. **Internet das Coisas: Um Desenho do Futuro**. Disponível em: <https://www.proof.com.br/blog/internet-das-coisas/>. Acesso em: 10 de nov. 2019.
- 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.
- SIRQUEIRA, Tassio; DALPRA, Humberto. NoSQL e a importância da Engenharia de Software e da Engenharia de Dados para o Big Data. **37º Jornada de Atualização da Informática (JAI)**. Congresso da Sociedade Brasileira de Computação (CSBC). Cap. 2. 2018.
- SOSA ESCUDERO, Walter. Big data: desafios para la docencia en Ciencias Económicas. **Econo**, v. 8, 2017.
- WWP. **Breve Histórico do Cadastro Único**. Disponível em: https://wwp.org.br/wp-content/uploads/2016/12/breve_historico_do_cadastro_unico.pdf. Acesso em: 10 de nov. 2019.