A influência do patrimônio pessoal do candidato no pleito eleitoral

**Cláudia Paratela Gama, Elias Nogueira Salgado, Gabriel De Deus Souza Pereira, Gustavo Fonseca Alvares, Luis Fernando Dornas Cancela Lino, Thales de Mattos Oliveira**

**Orientador: Nesley Jesus Daher**

PUC Minas

Curso de Tecnologia em Banco de Dados

1431286@sga.pucminas.br, 1434561@sga.pucminas.br, 1431704@sga.pucminas.br, 1437828@sga.pucminas.br, 1402045@sga.pucminas.br, 1055750@sga.pucminas.br, 69596@sga.pucminas.br

***Resumo:*** *Com o avanço das inovações tecnológicas, facilitando assim o acesso e a disseminação da informação, novos mecanismos de comunicação e influência estão à disposição de partidos políticos e candidatos. Meios próprios e riqueza patrimonial pessoal podem ser propulsores na adesão, financiamento e assertividade no uso destes novos mecanismos. Através da análise de dados públicos cedidos pelo TSE, buscaremos compreender a relação entre estas vantagens financeiras e a vitória nas urnas.*

# Introdução

A realidade socioeconômica do Brasil e do mundo vem mudando de maneira acelerada desde a virada do século. As inovações tecnológicas permitiram que qualquer pessoa seja capaz de produzir ou acessar informações de qualquer parte do mundo, sobre qualquer assunto a qualquer momento, desde que consiga arcar com o custo de sua urgência.

No contexto eleitoral, a população votante do Brasil aumenta a cada eleição, e novas formas de influenciar o eleitor estão à disposição de todo candidato ou partido político que deseje utilizá-las.

O poder econômico pode conferir vantagem significativa na aquisição de meios para propagar a mensagem que o candidato pretende, porém não é possível afirmar sem estudos que o patrimônio pessoal do candidato, sua riqueza pessoal, refletirá a vantagem que o poder econômico pode trazer.

Neste contexto, faz-se relevante analisar a influência do patrimônio pessoal do candidato no processo eleitoral.

Desta maneira, o estudo proporá compreender, pela análise de dados divulgados pelo Tribunal Superior Eleitoral (TSE), qual o impacto do patrimônio declarado pelos candidatos em relação à vitória nas urnas. Serão analisados, também pelo prisma do patrimônio, o grau de instrução, gênero, faixa etária, cargo pretendido e região do País, para buscar entender a relação entre estes fatores, em conjunção com o valor dos bens declarados, e o sucesso eleitoral.

Portanto, este trabalho se justifica por permitir verificar se, apesar de não haver critérios de renda para candidatura e votação no pleito eleitoral desde a Proclamação da República, em realidade, comprovada pela análise de dados públicos desde a eleição de 2014, o valor da riqueza pessoal ainda se impõe no resultado das urnas, mesmo em uma sociedade em cuja informação é tão facilmente produzida e distribuída.

# Descrição das Bases de Dados Utilizadas

Foram examinadas bases de dados dos anos 2014, 2018 e 2022, disponíveis no Tribunal Superior Eleitoral.

Os dados são disponibilizados em arquivos “.CSV” separados por assunto (Candidatos, Bens de candidatos, Coligações, Vagas e Motivo de Cassação) e unidade federativa, com um arquivo que consolida os dados de todo o País. Serão utilizados os dados dos arquivos “Candidatos” e “Bens de candidatos” consolidados de todo o País.

Para relacionar os dados, foi utilizada uma das colunas que está presente em ambos os arquivos e que mostra um número sequencial do candidato e poderá ser usada como identificador único.

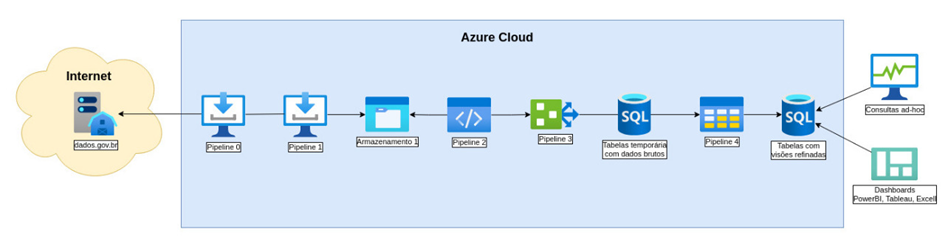
Do arquivo “Bens de candidatos”, serão utilizadas as colunas abaixo:

| **Nome da coluna/campo** | **Descrição** | **Tipo de dado** |
| --- | --- | --- |
| SQ\_CANDIDATO | Número sequencial do candidato gerado internamente pelos sistemas eleitorais. | Inteiro |
| CD\_TIPO\_BEM\_CANDIDATO | Código de registro para o tipo de bem | String |
| DS\_TIPO\_BEM\_CANDIDATO | Descrição de registro para o tipo de bem | String |
| VR\_BEM\_CANDIDATO | Valor declarado em reais do bem do candidato | Inteiro |

Do arquivo “Candidatos”, foram utilizadas as seguintes colunas:

| **Nome da coluna/campo** | **Descrição** | **Tipo** |
| --- | --- | --- |
| SQ\_CANDIDATO | Número sequencial do candidato gerado internamente pelos sistemas eleitorais. | String |
| SG\_UE | Sigla da Unidade Eleitoral em que o candidato concorre na eleição. Em caso de abrangência estadual a sigla é a UF da candidatura. Em caso de abrangência federal utiliza-se a sigla BR. | String |
| DS\_CARGO | Cargo ao qual o candidato concorre na eleição | String |
| NM\_UE | Nome da Unidade Eleitoral | String |
| CD\_SIT\_TOT\_TURNO | Código da situação de totalização do candidato, naquele turno da eleição, após totalização dos votos (“eleito”, “não eleito”...) | Integer |
| DS\_SIT\_TOT\_TURNO | Descrição da situação de totalização do candidato | String |
| NR\_TURNO | Número do turno | Integer |
| DT\_ELEICAO | Data da Eleição | Date |
| NM\_CANDIDATO | Nome do candidato | String |
| NR\_CANDIDATO | Número do candidato | String |
| CD\_CARGO | Código do cargo ao qual o candidato concorre | Integer |
| DS\_CARGO | Descrição do cargo ao qual o candidato concorre | String |
| NM\_URNA\_CANDIDATO | Nome do candidato na urna eletrônica | String |
| NM\_SOCIAL\_CANDIDATO | Nome social do candidato | String |
| NR\_PROTOCOLO\_CANDIDATURA | Número do protocolo de registro de candidatura para o pleito | Integer |
| NR\_PROCESSO | Número do processo | Integer |
| NR\_PARTIDO | Número do partido pelo qual o candidato concorre | Integer |
| SG\_PARTIDO | Sigla do partido pelo qual o candidato concorre | String |
| NM\_PARTIDO | Nome do partido pelo qual o candidato concorre | String |
| SQ\_COLIGACAO | Identificador único da coligação | String |
| NM\_COLIGACAO | Nome da coligação | String |
| DS\_COMPOSICAO\_COLIGACAO | Descrição dos partidos que compõem a coligação | String |
| DS\_NACIONALIDADE | Descrição da nacionalidade do candidato | String |
| NR\_IDADE\_DATA\_POSSE | Número da idade que o candidato tomará posse | Integer |
| DS\_GENERO | Descrição do gênero do candidato | String |
| DS\_GRAU\_INSTRUCAO | Descrição do grau de instrução do candidato | String |
| DS\_ESTADO\_CIVIL | Descrição do estado civil do candidato | String |
| DS\_COR\_RACA | Descrição a qual raça/cor o candidato se identifica | String |
| DS\_OCUPACAO | Descrição da atividade ocupacional/profissional do candidato | String |
| VR\_DESPESA\_MAX\_CAMPANHA | Valor máximo da despesa da campanha | Double |
| SG\_UF\_NASCIMENTO | Sigla da Unidade Federal em que o candidato nasceu | String |
| NM\_MUNICIPIO\_NASCIMENTO | Nome do município em que o candidato nasceu | String |
| DT\_NASCIMENTO | Data de Nascimento do candidato | Date |
| CD\_SITUACAO\_CANDIDATO\_PLEITO | Código da situação do candidato para participar do pleito | Integer |
| DS\_SITUACAO\_CANDIDATO\_PLEITO | Descrição da situação do candidato para participar do pleito | String |
| NM\_URNA\_CANDIDATO | Nome do candidato na urna eletrônica | String |
| NM\_SOCIAL\_CANDIDATO | Nome social do candidato | String |
| NR\_PROTOCOLO\_CANDIDATURA | Número do protocolo da candidatura para o pleito | Integer |
| NR\_PROCESSO | Número do processo | Integer |

# Arquitetura



O modelo arquitetural proposto para a execução do projeto consiste em uma estrutura de ELT *(extract-load-transform)* onde cada *pipeline* tem a responsabilidade única pela sua tarefa e atua de forma conjunta e sequencial executando todas as etapas de extração, carga e transformação dos dados no seu armazenamento de destino.

O modelo de processamento ELT foi escolhido pela flexibilidade de trabalhar com os dados brutos e visões em um projeto onde o escopo pode ser modificado ou onde haja a necessidade de criar visões para responder às perguntas desejadas. Dessa forma, foram carregados os dados para dentro do banco de dados em uma zona temporária e em seguida foram aplicados os tratamentos, limpezas e criação das estruturas e tabelas finais.

Já o modelo relacional escolhido, para organizar as informações e criar as tabelas necessárias para entregar as respostas desejadas, foi o multidimensional de fato e dimensão, em uma estrutura semelhante ao modelo de dados dimensional (*star schema)*. Dessa forma é possível manter uma imutabilidade dos dados fato e um histórico de evolução e atualização.

Todos os serviços e *pipelines* rodam de forma virtualizada através de containeres no padrão *Oracle Cloud Infrastructure (OCI*), sendo assim escaláveis conforme a necessidade e demanda de processamento e entrega de dados. Os containeres são compatíveis com quaisquer orquestradores do mercado como o Docker, o Podman e o Kubernetes.

Para a construção da aplicação, foram utilizadas as tecnologias:

- Container OCI: para criação das imagens dos pipelines responsáveis pelos processos de extração, carga e transformação dos dados; além dos serviços de armazenamento e banco de dados.

- Docker: para a orquestração dos containeres de serviços e pipelines.

- MySQL: para o armazenamento persistente dos dados pré e pós processados e para serviços de entrega de dados para visualizadores e consultas ad-hoc.

- MinIO: para armazenamento bruto dos arquivos ZIP e CSV extraídos das fontes originais dos dados.

- PowerBI: para a visualização dos dados, construção de visões compartilhadas, consultas ad-hoc e dashboards gráficos.

- Shellscript: para a linguagem de programação para os pipelines de extração.

- SQL: para a linguagem de programação para a limpeza, tratamento, processamento e carga dos dados brutos para as bases oficiais.

A extração dos dados ocorre por download através de requisições de dados usando o protocolo HTTP. Os arquivos baixados são descomprimidos com a ferramenta UnZip e guardados temporariamente em um armazenamento efêmero em tempo de descompressão. Assim que são descomprimidos, os arquivos são movidos para o armazenamento compartilhado para dados brutos servido pela tecnologia MinIO.

A carga bruta e inicial dos dados se dá através do arquivo CSV original obtido na fonte de dados. Esse arquivo é copiado no armazenamento compartilhado para dados brutos e carregado para uma área de *staging* no banco de dados MySQL. Essa área de *staging* serve para receber os dados brutos para que sejam limpos e tratados antes de serem copiados para as bases e tabelas finais.

As transformações dos dados ocorrem em tempo de criação das tabelas finais onde os dados são filtrados, limpos, transformados e selecionados conforme a necessidade da tabela a ser criada.

Após o processo de ELT, os dados estarão disponíveis para consumo através de quaisquer ferramentas de BI que aceitem conexão com MySQL. Para fins de demonstração dos dados em dashboards dinâmicos, foi definido o uso do PowerBI.

# Metodologia

## Metodologia do trabalho

A metodologia utilizada neste trabalho foi o Método Ágil Scrum.

Características:

* O que precisa ser feito não é totalmente entendido ou definido no início do trabalho;
* O Banco de Dados vai sendo criado aos poucos;
* Os requisitos podem mudar com o passar do tempo;
* Cada nova versão vai sendo criada e disponibilizada a cada período curto: uma, duas ou três semanas;
* O foco concentra em responder de forma ágil cada etapa que é solicitada;
* O feedback é rápido, podendo efetuar alterações ao longo do processo;
* É utilizada muita informação informal;
* A equipe é pequena;
* Os incrementos são pequenos;
* Iterações curtas;
* A liderança é diluída: Todos da equipe são responsáveis pelo trabalho;
* Há comunicação e cooperação entre os membros do grupo;
* O Facilitador é o SCRUM MASTER: Professor Nesley;
* O trabalho é organizado a partir do backlog do produto;

| **Como um(a)** | **Eu quero ser capaz de** | **Para que** | **Prioridade** |
| --- | --- | --- | --- |
| Usuário | Obter a relação geral de candidatos que efetivamente participam do pleito eleitoral, por UF e por cargo. | Eu tenha um panorama geral dos candidatos participantes do pleito. | Alta |
| Usuário | Obter intervalos de valores de patrimônio declarado pelos candidatos. Ex.: Quantos candidatos declararam patrimônio até 100 mil, 101 a 250 mil, 250 a 500 mil ..., qual faixa possui mais candidatos | Eu possa estabelecer eventuais relações entre o patrimônio dos candidatos e o sucesso no pleito eleitoral. Ex.: qual percentual de candidatos na faixa de patrimônio até 100 mil reais foi eleito, e na faixa 100 a 250 mil, 250 a 500 mil... qual faixa patrimonial possui mais vencedores | Alta |
| Usuário | Visualizar através de tabela ou gráfico os percentuais patrimoniais dos candidatos e o percentual de eleitos para cada faixa/intervalo de patrimônio | Eu possa estabelecer eventuais equivalências entre patrimônio e eleição de candidatos. Ex.: 60% dos candidatos tem patrimônio até 500 mil, contudo somente 30% dos candidatos nessa faixa patrimonial foram eleitos. | Alta |
| Usuário | Verificar o percentual de mulheres e homens participantes dos pleitos eleitorais, separando o patrimônio declarado por sexo. | Eu possa verificar, por exemplo, em que faixa patrimonial se encontra a maior parte das mulheres candidatas e a maior parte das eleitas | Alta |
| Usuário | Verificar o percentual de participantes dos pleitos por cor e raça. | Eu possa verificar, por exemplo, em que faixa patrimonial se encontra a maior parte dos candidatos que se declararam pardos. | Alta |
| Usuário | Verificar o percentual de participantes dos pleitos por grau de instrução. | Eu possa estabelecer a relação entre a faixa patrimonial e o grau de instrução dos candidatos. | Alta |
| Usuário | Verificar o percentual de participantes dos pleitos por localidade. | Eu possa verificar em quais localidades estão os candidatos que declaram possuir mais bens. | Alta |

Papéis:

* Product Owner: Dono do Produto: Coordenação do Curso;
* Scrum Master: Facilitador: Professor Nesley:

1. Toma iniciativas para melhorar o trabalho;
2. Resolve os impedimentos;
3. Não se envolve com o trabalho técnico.

* Equipe de Desenvolvimento: Membros do Grupo 10:

1. Auto-organizada;
2. Responsável por entregar valor por meio do desenvolvimento do trabalho.

No SCRUM temos as fases:

* Planejamento Geral e Projeto de Arquitetura
* Ciclo Sprint: Cada etapa do trabalho é considerado um Ciclo Sprint.

1. Seleciona o que precisa ser feito: Definido previamente pelos professores do curso;
2. Desenvolve o que deve ser feito: Membros do Grupo 10;
3. Revisa o que deve ser feito: Membros do Grupo 10 e o Professor Nesley;
4. Avalia o que deve ser feito: Professor Nesley.

Observação: O ciclo Sprint é repetido até o final do trabalho.

* Cerimônias:

1. Planejamento: O grupo reúne e define quem desenvolve cada parte da sprint;
2. Reuniões: São feitas reuniões entre o professor, SCRUM MASTER e a equipe de desenvolvimento, o Grupo 10;
3. Revisão da Sprint: SCRUM MASTER e a equipe de desenvolvimento: Professor e Grupo 10;
4. Retrospectiva: Equipe de Desenvolvimento.

* Encerramento do Projeto: Entrega o projeto para o professor e finaliza o trabalho.

## Metodologia do tratamento de dados

* + 1. **Pipeline 0**

Criação do usuário, do banco de dados e dos privilégios para o usuário acessar o banco de dados.

* + 1. **Pipeline 1 e Armazenamento 1**

Como explicado no item 2, foram utilizadas as bases de dados disponibilizadas pelo TSE, que podem ser encontradas no site dados.gov.br. Os arquivos “.ZIP” foram baixados, descomprimidos e salvos em um repositório de dados. Este repositório de dados foi acessado e apenas os arquivos com os dados consolidados de todo o país foram separados.

* + 1. **Pipeline 2**

Criação das estruturas no banco de dados para receber os dados dos arquivos “.CSV”.

* + 1. **Pipeline 3**

Analisando as fontes originais foi encontrada grande quantidade de dados que não foram utilizados para responder às questões enfrentadas. Dessa forma, primeiramente, cabe destacar que para cada eleição um conjunto de dois arquivos foi utilizado, como explicado no item 3, e destes arquivos, as tabelas foram construídas e relacionadas pelo número sequencial de cada candidato.

Os dados foram inseridos nas respectivas tabelas no banco de dados. A coluna SQ\_CANDIDATO, presente em todas as tabelas, serviu como a chave para relacioná-las.

* + 1. **Pipeline 4**

Nos arquivos “Bens de Candidatos” foram construídas tabelas de dimensão contendo todos os bens declarados por todos os candidatos, e tabelas de fatos registrando a declaração de cada bem por cada um dos candidatos. Através do fato de declaração dos bens do candidato foi possível criar uma visão para calcular o montante total declarado pelo candidato à Justiça Eleitoral. Desta forma conseguimos saber o valor total de cada candidato eleito nos pleitos analisados.

Nos arquivos “Candidatos” foram criadas tabelas de dimensão para candidatos (número, registro, situação, etc.), eleições (turnos, Estados, data, etc.), variáveis dos candidatos (gênero, escolaridade, etnia, etc.), cargos a serem disputados e partidos. Desta forma, foram utilizadas todas as colunas necessárias para a criação das dimensões mencionadas.

Após a criação das dimensões, foram criadas tabelas fatos para registrar todas as informações referentes às eleições, à candidatura, ao pleito e as variáveis vinculadas ao candidato.

# Captura de Dados

## Infraestrutura

A arquitetura do projeto foi pensada para ser construída em cima de containers e rodar através do Docker. Dessa forma se torna fácil fazer o *deployment* do sistema em qualquer *cloud* pública ou privada da nossa escolha. Os serviços escolhidos para armazenar os dados brutos e os dados relacionais foram:

1. Banco de dados: MySQL
2. Repositório de dados: MinIO Object Storage

Os serviços do sistema foram configurados para serem executados através do *docker-compose*.

Os serviços de *storage*, banco de dados e gerenciador gráfico são os principais do sistema. Eles precisam rodar de forma permanente para que as etapas dos pipelines sejam concluídas com sucesso.

Os serviços de pipelines rodam de forma síncrona e encadeada para realizar todas as etapas descritas na metodologia. São serviços efêmeros que são executados por demanda conforme a necessidade de carregar (ou recarregar) dados de um ano eleitoral.

## Serviços Principais

Os serviços principais estão configurados da seguinte maneira:

* + 1. **Repositório de dados MinIO**

| storage:  container\_name: storage  image: minio/minio:latest  networks:  - net\_eixo01\_grupo10  restart: always  ports:  - "9000:9000"  - "9001:9001"  command: server /data --console-address ":9001"  environment:  - MINIO\_ROOT\_USER=minioadmin  - MINIO\_ROOT\_PASSWORD=minioadmin  volumes:  - minio-data:/data |
| --- |

* + 1. **Banco de dados MySQL**

| db:  container\_name: db  image: mysql:8  networks:  - net\_eixo01\_grupo10  restart: always  ports:  - "3306:3306"  environment:  MYSQL\_ROOT\_PASSWORD: grupo10  volumes:  - mysql-data:/var/lib/mysql  command: |  --default-authentication-plugin=mysql\_native\_password  --secure-file-priv=/tmp  --local-infile=1 |
| --- |

* + 1. **Gerenciador gráfico para o banco de dados**

| superset:  container\_name: superset  build:  context: ./image  dockerfile: Dockerfile.superset  networks:  - net\_eixo01\_grupo10  restart: always  ports:  - 8088:8088 |
| --- |

* + 1. **Serviços de Pipeline**

A ideia inicial para a criação dos *pipelines* foi a de ingerir o dado bruto o mais rápido possível em uma zona de *staging* dentro do banco de dados para que os dados pudessem ser analisados, tratados, limpos e transformados com a linguagem SQL.

O método de processamento escolhido foi algo semelhante ao método ELT onde será carregado para o banco de dados o dado bruto e a partir dele carregado será feito as transformações e criações de visões e *data-marts*.

* + - 1. *Pipeline 0 - Criação de usuários e database*

Esse *pipeline* tem como objetivo a criação das configurações básicas no banco de dados. Nessa etapa são criados o usuário de acesso, o banco de dados que armazenará as informações e os privilégios do usuário para administrar esse banco de dados. A configuração do serviço de pipeline 0 é:

| pipeline\_0\_install\_user\_and\_database:  container\_name: pipeline\_0\_install\_user\_and\_database  build: ./image  networks:  - net\_eixo01\_grupo10  entrypoint: /app/pipeline.sh  volumes:  - ./pipeline\_0\_install\_user\_and\_database/pipeline.sh:/app/pipeline.sh:ro  environment:  - MINIO\_ROOT\_USER=minioadmin  - MINIO\_ROOT\_PASSWORD=minioadmin  - MINIO\_NAME=local  - MINIO\_URL=http://storage:9000  - MINIO\_BUCKET\_NAME=eixo01-grupo10  - MYSQL\_ROOT\_USERNAME=root  - MYSQL\_ROOT\_PASSWORD=grupo10  - MYSQL\_USERNAME=grupo10  - MYSQL\_PASSWORD=grupo10  - MYSQL\_DATABASE=eixo01 |
| --- |

As instruções utilizadas nesse passo estão descritas no arquivo pipeline.sh. Variáveis com instruções de criação de usuário, banco de dados e privilégios:

| # NOTA: Instruções para criar o usuário e senha no banco de dados CREATE\_USER="CREATE USER IF NOT EXISTS '${MYSQL\_USERNAME}'@'%' IDENTIFIED BY '${MYSQL\_PASSWORD}';"  # NOTA: Instruções para criar o banco de dados CREATE\_DATABASE="CREATE DATABASE IF NOT EXISTS ${MYSQL\_DATABASE};"  # NOTA: Instrução para garantir privilégios do usuário no banco de dados criado CREATE\_PERMISSIONS="GRANT ALL ON ${MYSQL\_DATABASE}.\* TO '${MYSQL\_USERNAME}'@'%' WITH GRANT OPTION;"  # NOTA: Valida se existe o usuário no banco de dados VALIDATE\_USERNAME="SELECT User FROM mysql.user WHERE User = '${MYSQL\_USERNAME}';" |
| --- |

A criação do *bucket* do projeto no repositório de dados:

| # NOTA: Autenticação no Repositório de dados e cria um bucket mc alias set ${MINIO\_NAME} ${MINIO\_URL} ${MINIO\_ROOT\_USER} ${MINIO\_ROOT\_PASSWORD} && mc mb ${MINIO\_NAME}/${MINIO\_BUCKET\_NAME} |
| --- |

E por fim a execução das instruções no banco de dados:

| # NOTA: Executa a instrução de criar usuário mysql -h db -u ${MYSQL\_ROOT\_USERNAME} -p"${MYSQL\_ROOT\_PASSWORD}" <<< ${CREATE\_USER}  # NOTA: Executa a instrução de criar banco de dados mysql -h db -u ${MYSQL\_ROOT\_USERNAME} -p"${MYSQL\_ROOT\_PASSWORD}" <<< ${CREATE\_DATABASE}  # NOTA: Executa a instrução de criar os privilégios do usuário no banco de dados mysql -h db -u ${MYSQL\_ROOT\_USERNAME} -p"${MYSQL\_ROOT\_PASSWORD}" <<< ${CREATE\_PERMISSIONS} |
| --- |

* + - 1. *Pipeline 1 – Download dos Arquivos*

Esse pipeline tem como objetivo obter os arquivos oficiais que serão utilizados no projeto. Ele acessa o site do TSE e baixa os arquivos referentes ao ano eleitoral desejado.

Após executar o download do arquivo, ele o descomprime e copia para o repositório de dados somente o arquivo desejado. A configuração do serviço de pipeline 1 é:

| pipeline\_1\_download\_files:  container\_name: pipeline\_1\_download\_files  build: ./image  networks:  - net\_eixo01\_grupo10  entrypoint: /app/pipeline.sh  volumes:  - ./pipeline\_1\_download\_files/pipeline.sh:/app/pipeline.sh:ro  environment:  - MINIO\_ROOT\_USER=minioadmin  - MINIO\_ROOT\_PASSWORD=minioadmin  - MINIO\_NAME=local  - MINIO\_URL=http://storage:9000  - MINIO\_BUCKET\_NAME=eixo01-grupo10  command: -y 2022 |
| --- |

As instruções utilizadas nesse passo estão descritas no arquivo pipeline.sh. Criação de um conjunto contendo os anos eleitorais permitidos:

| YEARS=(  "2014"  "2018"  "2022" ) |
| --- |

Função responsável por baixar os arquivos do site do TSE:

| function download\_files() {  year=$1   files\_to\_download=(  "https://cdn.tse.jus.br/estatistica/sead/odsele/consulta\_cand/consulta\_cand\_\_\_YEAR\_\_.zip"  "https://cdn.tse.jus.br/estatistica/sead/odsele/bem\_candidato/bem\_candidato\_\_\_YEAR\_\_.zip"  )   for f in "${files\_to\_download[@]}"; do  url=$(echo ${f} | sed "s@\_\_YEAR\_\_@${year}@g")  wget \  -U "Mozilla/5.0 (compatible; MSIE 10.6; Windows NT 6.1; Trident/5.0; InfoPath.2; SLCC1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 2.0.50727) 3gpp-gba UNTRUSTED/1.0" \  ${url} \  -O /tmp/$(basename ${url})  done } |
| --- |

Função responsável por descomprimir os arquivos:

| function uncompress\_files() {  year=$1   files\_to\_uncompress=(  "/tmp/consulta\_cand\_\_\_YEAR\_\_.zip"  "/tmp/bem\_candidato\_\_\_YEAR\_\_.zip"  )   for f in "${files\_to\_uncompress[@]}"; do  file=$(echo ${f} | sed "s@\_\_YEAR\_\_@${year}@g")  (  export load="/tmp/load/$(basename ${file%.zip})"  mkdir -p ${load}  cd ${load}  unzip -x ${file}  )  done } |
| --- |

E a função responsável por salvar o arquivo CSV desejado no repositório de dados:

| function upload\_to\_repository() {  year=$1   FILES=(  "/tmp/load/bem\_candidato\_\_\_YEAR\_\_/bem\_candidato\_\_\_YEAR\_\_\_BRASIL.csv"  "/tmp/load/consulta\_cand\_\_\_YEAR\_\_/consulta\_cand\_\_\_YEAR\_\_\_BRASIL.csv"  )   for f in "${FILES[@]}"; do  fname=$(echo ${f} | sed "s@\_\_YEAR\_\_@${year}@g")  (iconv -f ISO-8859-1 -t UTF-8 ${fname} > ${fname}.iconv && mv ${fname}.iconv ${fname})  mc cp ${fname} ${MINIO\_NAME}/${MINIO\_BUCKET\_NAME}/$(basename ${fname})  done } |
| --- |

* + - 1. *Pipeline 2 - Criação das estruturas para receber os dados*

O pipeline 2 tem como objetivo a criação da estrutura base para receber os dados brutos dos arquivos CSV dentro do banco de dados. A configuração do serviço de pipeline 2 é:

| pipeline\_2\_create\_structures:  container\_name: pipeline\_2\_create\_structures  build: ./image  networks:  - net\_eixo01\_grupo10  entrypoint: /app/pipeline.sh  volumes:  - ./pipeline\_2\_create\_structures/pipeline.sh:/app/pipeline.sh:ro  environment:  - MYSQL\_USERNAME=grupo10  - MYSQL\_PASSWORD=grupo10  - MYSQL\_DATABASE=eixo01 |
| --- |

As instruções utilizadas nesse passo estão descritas no arquivo “pipeline.sh”. Armazena na variável de nome “CREATE\_CONSULTA\_CAND” a instrução para a criação da tabela “raw\_consulta\_candidato” no banco de dados:

| CREATE\_CONSULTA\_CAND=$(cat <<EOF CREATE TABLE IF NOT EXISTS ${MYSQL\_DATABASE}.raw\_consulta\_candidato (  DT\_GERACAO VARCHAR(255),  HH\_GERACAO VARCHAR(255),  ANO\_ELEICAO VARCHAR(255),  CD\_TIPO\_ELEICAO VARCHAR(255),  NM\_TIPO\_ELEICAO VARCHAR(255),  NR\_TURNO VARCHAR(255),  CD\_ELEICAO VARCHAR(255),  DS\_ELEICAO TEXT,  DT\_ELEICAO VARCHAR(255),  TP\_ABRANGENCIA VARCHAR(255),  SG\_UF VARCHAR(255),  SG\_UE VARCHAR(255),  NM\_UE VARCHAR(255),  CD\_CARGO VARCHAR(255),  DS\_CARGO TEXT,  SQ\_CANDIDATO VARCHAR(255),  NR\_CANDIDATO VARCHAR(255),  NM\_CANDIDATO VARCHAR(255),  NM\_URNA\_CANDIDATO VARCHAR(255),  NM\_SOCIAL\_CANDIDATO VARCHAR(255),  NR\_CPF\_CANDIDATO VARCHAR(255),  NM\_EMAIL VARCHAR(255),  CD\_SITUACAO\_CANDIDATURA VARCHAR(255),  DS\_SITUACAO\_CANDIDATURA TEXT,  CD\_DETALHE\_SITUACAO\_CAND VARCHAR(255),  DS\_DETALHE\_SITUACAO\_CAND TEXT,  TP\_AGREMIACAO VARCHAR(255),  NR\_PARTIDO VARCHAR(255),  SG\_PARTIDO VARCHAR(255),  NM\_PARTIDO VARCHAR(255),  NR\_FEDERACAO VARCHAR(255),  NM\_FEDERACAO VARCHAR(255),  SG\_FEDERACAO VARCHAR(255),  DS\_COMPOSICAO\_FEDERACAO TEXT,  SQ\_COLIGACAO VARCHAR(255),  NM\_COLIGACAO VARCHAR(255),  DS\_COMPOSICAO\_COLIGACAO TEXT,  CD\_NACIONALIDADE VARCHAR(255),  DS\_NACIONALIDADE TEXT,  SG\_UF\_NASCIMENTO VARCHAR(255),  CD\_MUNICIPIO\_NASCIMENTO VARCHAR(255),  NM\_MUNICIPIO\_NASCIMENTO VARCHAR(255),  DT\_NASCIMENTO VARCHAR(255),  NR\_IDADE\_DATA\_POSSE VARCHAR(255),  NR\_TITULO\_ELEITORAL\_CANDIDATO VARCHAR(255),  CD\_GENERO VARCHAR(255),  DS\_GENERO TEXT,  CD\_GRAU\_INSTRUCAO VARCHAR(255),  DS\_GRAU\_INSTRUCAO TEXT,  CD\_ESTADO\_CIVIL VARCHAR(255),  DS\_ESTADO\_CIVIL VARCHAR(255),  CD\_COR\_RACA VARCHAR(255),  DS\_COR\_RACA TEXT,  CD\_OCUPACAO VARCHAR(255),  DS\_OCUPACAO TEXT,  VR\_DESPESA\_MAX\_CAMPANHA VARCHAR(255),  CD\_SIT\_TOT\_TURNO VARCHAR(255),  DS\_SIT\_TOT\_TURNO TEXT,  ST\_REELEICAO VARCHAR(255),  ST\_DECLARAR\_BENS VARCHAR(255),  NR\_PROTOCOLO\_CANDIDATURA VARCHAR(255),  NR\_PROCESSO VARCHAR(255),  CD\_SITUACAO\_CANDIDATO\_PLEITO VARCHAR(255),  DS\_SITUACAO\_CANDIDATO\_PLEITO TEXT,  CD\_SITUACAO\_CANDIDATO\_URNA VARCHAR(255),  DS\_SITUACAO\_CANDIDATO\_URNA TEXT,  ST\_CANDIDATO\_INSERIDO\_URNA VARCHAR(255),  NM\_TIPO\_DESTINACAO\_VOTOS VARCHAR(255),  CD\_SITUACAO\_CANDIDATO\_TOT VARCHAR(255),  DS\_SITUACAO\_CANDIDATO\_TOT TEXT,  ST\_PREST\_CONTAS VARCHAR(255)  )  CHARACTER SET utf8;  EOF ) |
| --- |

Armazena na variável de nome “CREATE\_BEM\_CANDIDATO” a instrução para a criação da tabela “raw\_bem\_candidato” no banco de dados:

| CREATE\_BEM\_CANDIDATO=$(cat <<EOF CREATE TABLE IF NOT EXISTS ${MYSQL\_DATABASE}.raw\_bem\_candidato (  DT\_GERACAO VARCHAR(255),  HH\_GERACAO VARCHAR(255),  ANO\_ELEICAO VARCHAR(255),  CD\_TIPO\_ELEICAO VARCHAR(255),  NM\_TIPO\_ELEICAO VARCHAR(255),  CD\_ELEICAO VARCHAR(255),  DS\_ELEICAO TEXT,  DT\_ELEICAO VARCHAR(255),  SG\_UF VARCHAR(255),  SG\_UE VARCHAR(255),  NM\_UE VARCHAR(255),  SQ\_CANDIDATO VARCHAR(255),  NR\_ORDEM\_CANDIDATO VARCHAR(255),  CD\_TIPO\_BEM\_CANDIDATO VARCHAR(255),  DS\_TIPO\_BEM\_CANDIDATO TEXT,  DS\_BEM\_CANDIDATO TEXT,  VR\_BEM\_CANDIDATO VARCHAR(255),  DT\_ULTIMA\_ATUALIZACAO VARCHAR(255),  HH\_ULTIMA\_ATUALIZACAO VARCHAR(255)  )  CHARACTER SET utf8;  EOF ) |
| --- |

E por fim a execução das instruções no banco de dados:

| mysql -h db -u ${MYSQL\_USERNAME} -p”${MYSQL\_PASSWORD}” –database=${MYSQL\_DATABASE} <<< ${CREATE\_CONSULTA\_CAND} mysql -h db -u ${MYSQL\_USERNAME} -p”${MYSQL\_PASSWORD}” –database=${MYSQL\_DATABASE} <<< ${CREATE\_BEM\_CANDIDATO} |
| --- |

* + - 1. *Pipeline 3 - Ingestão dos dados brutos para área temporária*

O pipeline 3 tem o objetivo de copiar do repositório de dados para o disco local do pipeline o arquivo CSV do ano eleitoral desejado. Após copiado, o pipeline realiza a ingestão dos dados do arquivo nas tabelas “raw\_bem\_candidato” e “raw\_consulta\_candidato” do banco de dados. A configuração do serviço de pipeline 3 é:

| pipeline\_3\_ingest\_into\_staging\_area:  container\_name: pipeline\_3\_ingest\_into\_staging\_area  build: ./image  networks:  - net\_eixo01\_grupo10  entrypoint: /app/pipeline.sh  volumes:  - ./pipeline\_3\_ingest\_into\_staging\_area/pipeline.sh:/app/pipeline.sh:ro  environment:  - MINIO\_ROOT\_USER=minioadmin  - MINIO\_ROOT\_PASSWORD=minioadmin  - MINIO\_NAME=local  - MINIO\_URL=http://storage:9000  - MINIO\_BUCKET\_NAME=eixo01-grupo10  - MYSQL\_USERNAME=grupo10  - MYSQL\_PASSWORD=grupo10  - MYSQL\_DATABASE=eixo01  command: -y 2022 |
| --- |

As instruções utilizadas nesse passo estão descritas no arquivo pipeline.sh. Criação de um conjunto contendo os anos eleitorais permitidos:

| YEARS=(  "2014"  "2018"  "2022" ) |
| --- |

Função que imprime a instrução necessária para carregar os dados do arquivo “consulta\_cand\_XXXX\_BRASIL.csv” na tabela “raw\_consulta\_candidato”:

| function load\_consulta\_candidato() {  year=$1   INGEST\_DATA\_CONSULTA\_CANDIDATO=$(cat <<EOF LOAD DATA LOCAL INFILE '/tmp/consulta\_cand\_${year}\_BRASIL.csv' INTO TABLE raw\_consulta\_candidato FIELDS TERMINATED BY ';' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS; EOF )   echo ${INGEST\_DATA\_CONSULTA\_CANDIDATO}  return 0 } |
| --- |

Função que imprime a instrução necessária para carregar os dados do arquivo “bem\_candidato\_XXXX\_BRASIL.csv” na tabela “raw\_bem\_candidato”:

| function load\_bem\_candidato() {  year=$1   INGEST\_DATA\_BEM\_CANDIDATO=$(cat <<EOF LOAD DATA LOCAL INFILE '/tmp/bem\_candidato\_${year}\_BRASIL.csv' INTO TABLE raw\_bem\_candidato FIELDS TERMINATED BY ';' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS; EOF )   echo ${INGEST\_DATA\_BEM\_CANDIDATO}  return 0 } |
| --- |

A realização da cópia dos arquivos do ano eleitoral desejado do repositório de dados para o disco local:

| mc alias set ${MINIO\_NAME} ${MINIO\_URL} ${MINIO\_ROOT\_USER} ${MINIO\_ROOT\_PASSWORD} >/dev/null 2>&1 mc cp ${MINIO\_NAME}/${MINIO\_BUCKET\_NAME}/bem\_candidato\_${YEAR}\_BRASIL.csv /tmp/bem\_candidato\_${YEAR}\_BRASIL.csv mc cp ${MINIO\_NAME}/${MINIO\_BUCKET\_NAME}/consulta\_cand\_${YEAR}\_BRASIL.csv /tmp/consulta\_cand\_${YEAR}\_BRASIL.csv |
| --- |

E por fim, verifica a existência dos arquivos e inicia a execução das instruções das cargas de dados dos arquivos para as tabelas do banco de dados com base no ano eleitoral desejado:

| [[ -f "/tmp/bem\_candidato\_${YEAR}\_BRASIL.csv" ]] && mysql -h db -u ${MYSQL\_USERNAME} -p"${MYSQL\_PASSWORD}" --database=${MYSQL\_DATABASE} --local-infile=1 <<< $(load\_bem\_candidato ${YEAR}) [[ -f "/tmp/consulta\_cand\_${YEAR}\_BRASIL.csv" ]] && mysql -h db -u ${MYSQL\_USERNAME} -p"${MYSQL\_PASSWORD}" --database=${MYSQL\_DATABASE} --local-infile=1 <<< $(load\_consulta\_candidato ${YEAR}) |
| --- |

# Limpeza e Transformação

Inicialmente, para a limpeza e o tratamento dos dados, foram removidos todos os registros inválidos da coluna **SQ\_CANDIDATO**. Registros inválidos são representados por valores nulos ou caracteres fora do padrão aceito para a chave que vincula os dados pessoais e bens declarados dos candidatos.

 Foi realizada a limpeza dos registros conforme o tipo de dados. Para as colunas com tipos numéricos foram removidas todas as ocorrências de caracteres A-Z, a-z e caracteres especiais. Para as colunas com tipo *string,* todos os valores foram padronizados em caixa alta (uppercase) e aplicado a limpeza de caracteres de espaço no início e no fim das palavras.

 Como foi escolhido uma modelagem multidimensional de fato-dimensão, para a criação das dimensões foram listados todos os registros únicos existentes para cada domínio de informação e assim criando as tabelas de dimensão. Abaixo um exemplo prático de como foi realizado essa operação:

| CREATE TABLE ${MYSQL\_DATABASE}.dim\_cargo (  cargo\_id INT NOT NULL AUTO\_INCREMENT,  CD\_CARGO INT,  DS\_CARGO VARCHAR(20),  created\_at TIMESTAMP,  PRIMARY KEY(cargo\_id) ) AS SELECT DISTINCT  CAST(CD\_CARGO AS DOUBLE) AS CD\_CARGO  ,UPPER(DS\_CARGO) AS DS\_CARGO  ,NOW() AS "created\_at" FROM ${MYSQL\_DATABASE}.vw\_raw\_consulta\_candidato rcc WHERE CD\_CARGO IN (1,2,3,4,5,6,7,8,9,10) ORDER BY CD\_CARGO ASC; |
| --- |

No exemplo acima a tabela dimensão **cargo** (aqui denominada de *dim\_cargo*) foi criada a partir de uma consulta buscando registros únicos nos dados brutos. Ao tempo de criação da tabela, os dados foram selecionados e transformados como demonstrado em UPPER(DS\_CARGO) AS DS\_CARGO.

Outro exemplo de tratamento de dados está demonstrado abaixo:

| CREATE TABLE ${MYSQL\_DATABASE}.dim\_candidatura (  candidatura\_id INT NOT NULL AUTO\_INCREMENT,  NR\_PROTOCOLO\_CANDIDATURA VARCHAR(255),  NR\_PROCESSO VARCHAR(255),  created\_at TIMESTAMP,  PRIMARY KEY(candidatura\_id) ) AS SELECT DISTINCT  NR\_PROTOCOLO\_CANDIDATURA  ,NR\_PROCESSO  ,NOW() AS "created\_at" FROM ${MYSQL\_DATABASE}.vw\_raw\_consulta\_candidato rcc WHERE  -- NOTE: remove números inválidos  SUBSTRING(NR\_PROCESSO, 1, 3) = '060'; |
| --- |

Nesse caso foi realizada a busca nos dados brutos e descartado todos os registros que não possuíam o **NR\_PROCESSO** no padrão formatado.

E também no exemplo:

| CREATE TABLE ${MYSQL\_DATABASE}.dim\_candidato\_numero (  candidato\_numero\_id INT NOT NULL AUTO\_INCREMENT,  NR\_CANDIDATO INT,  created\_at TIMESTAMP,  PRIMARY KEY(candidato\_numero\_id) ) AS SELECT DISTINCT  CAST(NR\_CANDIDATO AS DOUBLE) AS NR\_CANDIDATO  ,NOW() AS "created\_at" FROM ${MYSQL\_DATABASE}.vw\_raw\_consulta\_candidato rcc; |
| --- |

Neste caso há a conversão de dados obtidos no registro de dados brutos do tipo *String* para o tipo *Double*.

Foi identificado um problema com o arquivo da eleição de 2014, que não apresentava o resultado da eleição para presidente, todos os candidatos apareciam com a informação #NULO# na coluna SIT\_TOT\_TURNO. Os dados nesta coluna foram alterados conforme resultado eleitoral amplamente divulgado, modificando os valores para os candidatos não eleitos a presidente para “NÃO ELEITO”, e para os eleitos para “ELEITO”.

Na coluna DS\_SIT\_TOT\_TURNO, foram substituídos os valores “ELEITO POR MÉDIA” e “ELEITO POR QP” por apenas “ELEITO”, e o valor “SUPLENTE” para “NÃO ELEITO”. Estas substituições foram necessárias para facilitar a visualização dos dados, visto que a forma como os deputados foram eleitos não nos importa, e todos os candidatos a deputado que atingem certa quantidade de votos aparecem na base de dados como suplentes, tomando a maior parte dos registros.

Foi adicionada a coluna “Faixa etária”, identificando a faixa etária do candidato com base em sua idade na data da posse, e a coluna “Faixa de bens” em que a soma dos bens dos candidatos foi separada por faixas. Essas colunas foram criadas para facilitar a visualização do universo de dados.

# Visualização de dados

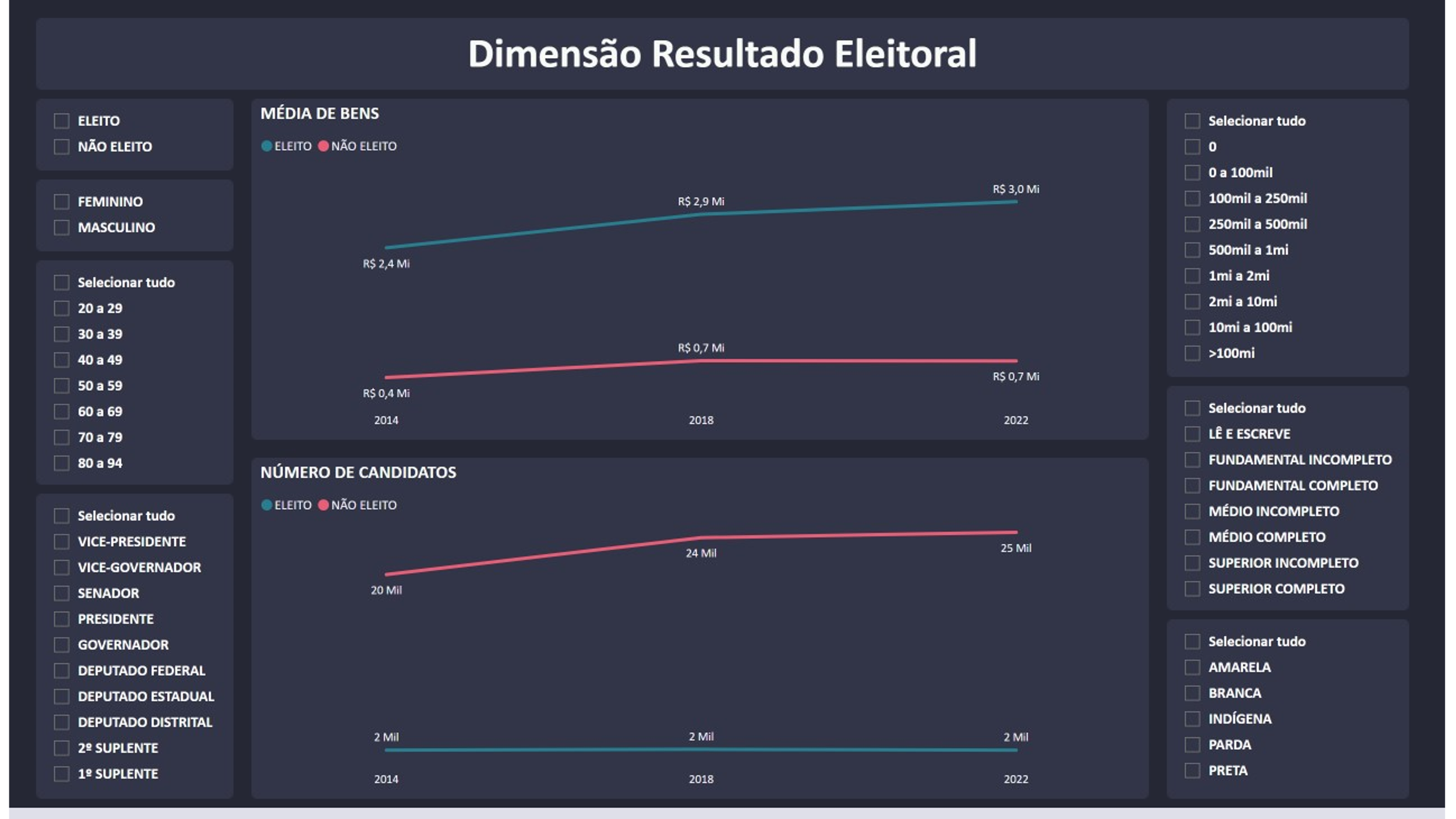
A partir dos dados transformados e limpos, foram criadas visualizações utilizando PowerBI permitindo verificar o impacto que o patrimônio declarado dos candidatos teve em relação à vitória nas urnas. Também foi possível verificar a distribuição do patrimônio pelos estados da federação, a distribuição do patrimônio declarado por cargo disputado e, utilizando os dados demográficos, verificar a influência do patrimônio pessoal em conjunção com fatores como cor, raça, gênero, grau de instrução e idade dos candidatos.

Para tanto, o primeiro passo foi realizar duas filtragens nos dados. A primeira, na coluna DS\_SITUACAO\_CANDIDATURA, pois apenas os candidatos que efetivamente participaram da eleição foram utilizados, cujo valor nesta coluna está indicado com “APTO”. A segunda, na coluna DS\_SIT\_TOT\_TURNO, já que uma das possibilidades de valores para esta coluna é o valor “2º TURNO”, que identifica candidaturas que seguiram no pleito até o segundo turno, ou seja, trata-se apenas do resultado do primeiro turno, duplicando os registros da candidatura. O interesse foi distinguir entre candidatos eleitos e não eleitos, dessa forma, foram excluídos da análise todos os registros que tinham a informação “2º TURNO”, visto que o resultado do segundo turno (“ELEITO” ou “NÃO ELEITO”), nos bastará para realizar a análise.

Em seguida, foram criadas duas medidas, uma para a média dos bens, realizando a média da coluna SOMA\_BEM\_CANDIDATO, utilizando a função “*AVERAGE*” do DAX, e outra para a contagem do número de candidatos, realizando a contagem da coluna SQ\_CANDIDATO, utilizando a função “*COUNTA*”. Essas medidas foram utilizadas para apresentar os dados nos gráficos para comparar o total de candidatos e a média de bens.

Foi desenvolvido um padrão utilizado para criar seis visualizações diferentes, consistente em um grupo de sete diferentes formas de segregação de dados (resultado eleitoral, gênero, faixa de etária, cargo disputado, faixa de bens, grau de instrução e cor/raça), localizadas nas laterais das visualizações, e, ao centro, gráficos que reagem às seleções do usuário. Diferentes estilos de gráficos foram selecionados para representar cada dimensão de dados, de forma que a visualização dos dados pudesse ser potencializada:

## Dimensão resultado eleitoral



Para apresentar a dimensão resultado eleitoral, foram utilizados dois gráficos de linhas que representam os candidatos eleitos e não eleitos, um com a média de bens por ano e outro com o número de candidatos por ano, ambas sensíveis às segmentações de dados disponibilizadas.

Esses gráficos foram utilizados para mostrar com clareza a diferença entre a média de bens dos candidatos eleitos e dos não eleitos, demonstrando que a média de bens declarados pelos candidatos eleitos é maior do que a média dos não eleitos para a maioria das segmentações de dados proposta.

Foi verificado que em alguns casos, em especial quando o universo de candidatos é menor, como no caso das eleições para o executivo, a soma de bens de um ou dois candidatos não eleitos que seja muito maior do que a dos outros pode gerar uma distorção grande no gráfico, como no caso da eleição de 2018 para presidente, quando dois candidatos não eleitos declararam possuir cada um mais de duas vezes a soma dos bens declarados pelos outros onze candidatos juntos.

Combinando diversas segmentações também é possível chegar a cenários em que o número de candidatos eleitos e não eleitos se aproxima muito, casos em que a média de bens dos não eleitos pode ser maior do que dos eleitos, porém são casos muito específicos, que fogem ao escopo do trabalho.

## Dimensão faixa de bens

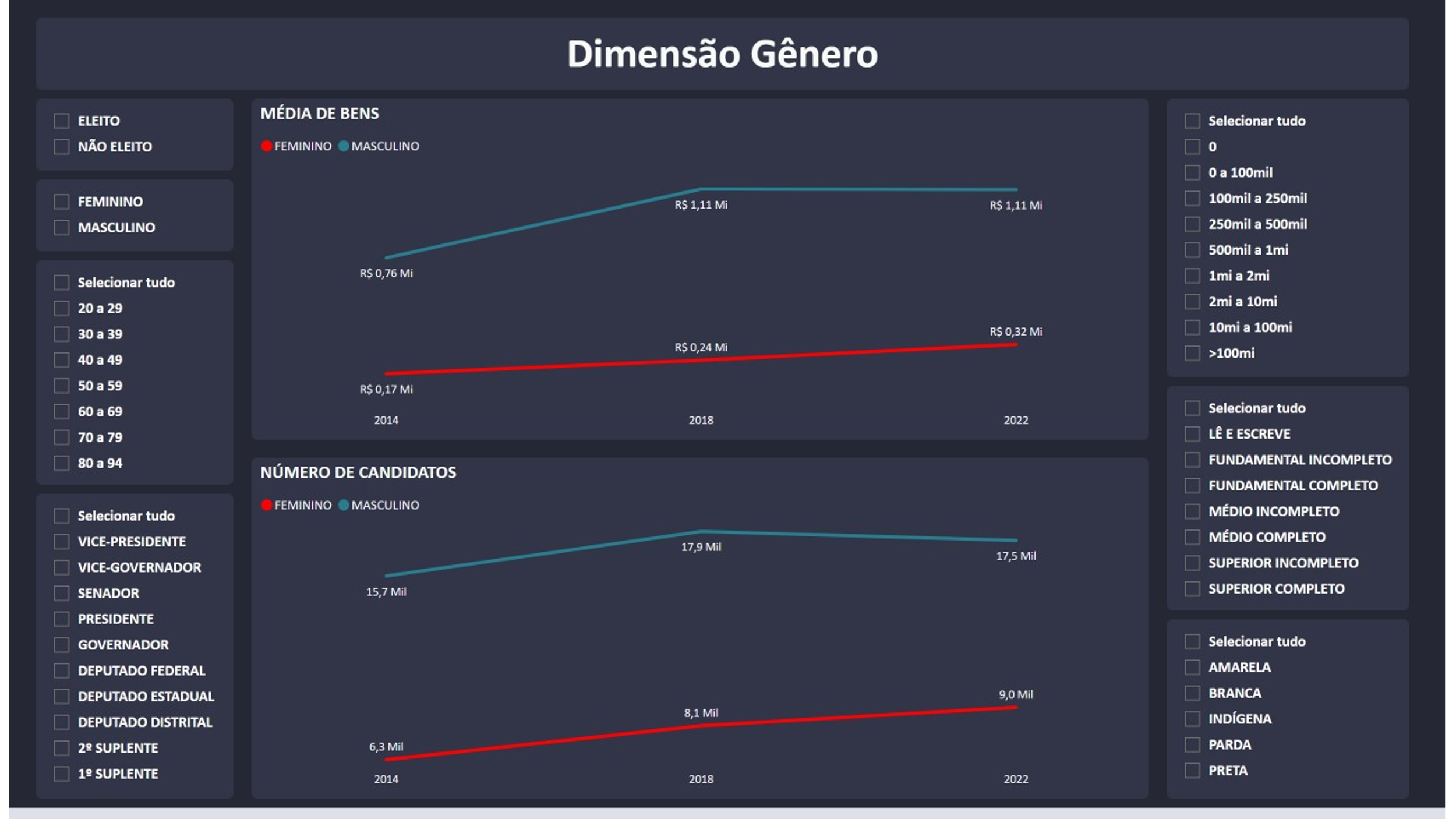
Graphical user interface, application

Description automatically generated

Para apresentar a dimensão faixa de bens, foram utilizados dois gráficos de colunas clusterizados, representando o número de candidatos por faixa de bens em cada ano, sensível às segmentações de dados disponibilizadas, e outro com o total de candidatos por faixa de bens em cada ano fixo, que não se modifica com a seleção de qualquer segmentação de dados.

Esses gráficos foram utilizados para comparar as segmentações propostas com o total geral por ano, de forma a entender, por exemplo, que cerca de 60% dos candidatos eleitos declara possuir entre 500 mil e 10 milhões de reais, demonstrando ainda mais claramente a diferença na distribuição das colunas dos dois gráficos quando comparados sob a segmentação de dados “Eleitos” para o primeiro gráfico.

## Dimensão gênero

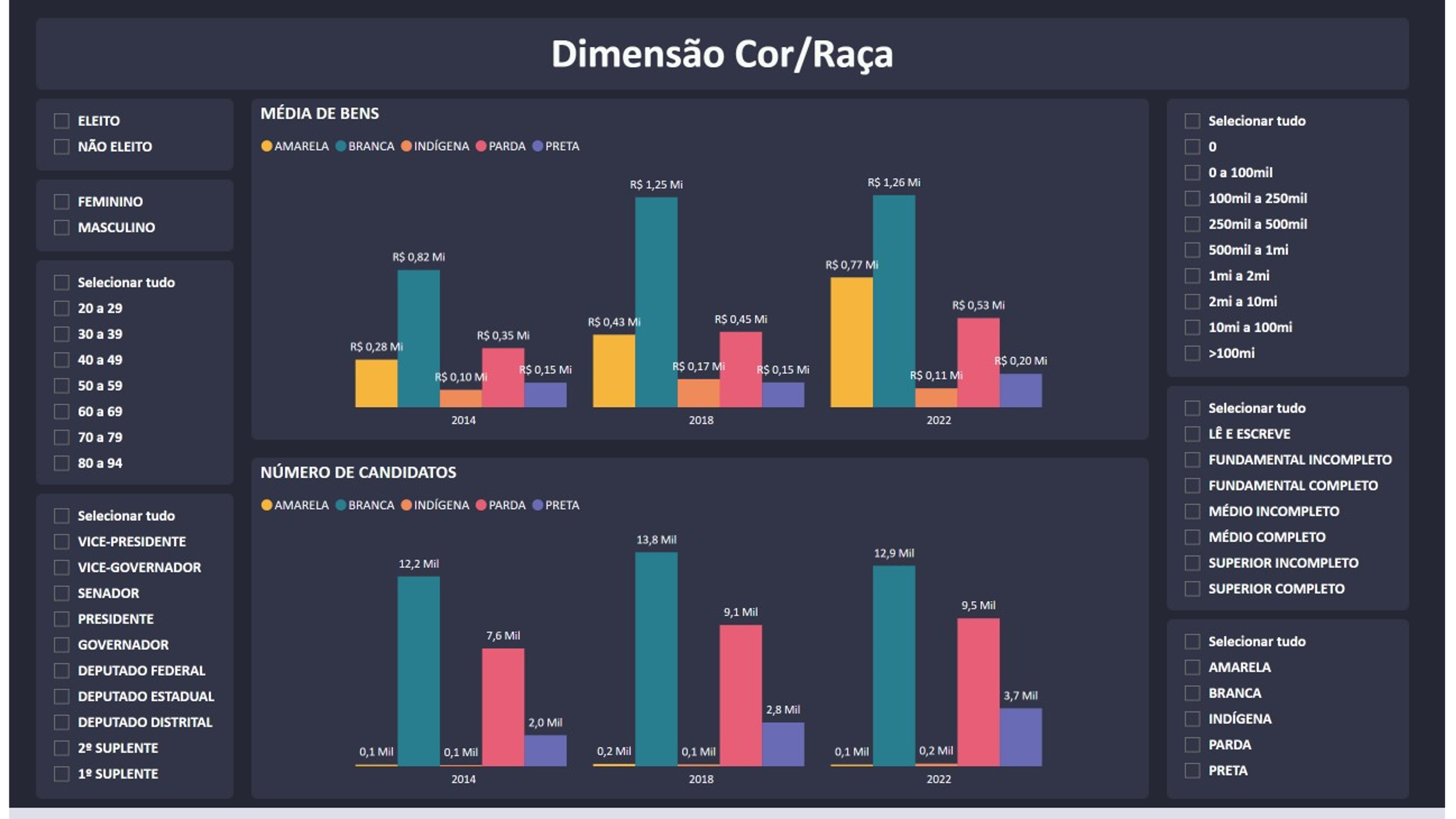


Para apresentar a dimensão gênero, foram utilizados dois gráficos de linhas que representam os candidatos do sexo masculino e feminino, um com a média de bens por ano e outro com o número de candidatos por ano, ambas sensíveis às segmentações de dados disponibilizadas.

Esses gráficos foram utilizados para demonstrar que o número total e a média de bens declarados por candidatos do sexo masculino são sempre maiores do que de candidatas do sexo feminino, porém nas duas últimas eleições a diferença tem se tornado menor, impacto, talvez, do aumento do número de candidatas de quase 50% ao longo destas eleições.

Quando os dados são segmentados pelos candidatos eleitos, verifica-se que a maioria é do sexo masculino, entretanto o número de candidatas bem crescendo nos últimos anos, porém a média de bens dos eleitos é muito maior entre candidatos do sexo masculino.

## Dimensão cor/raça

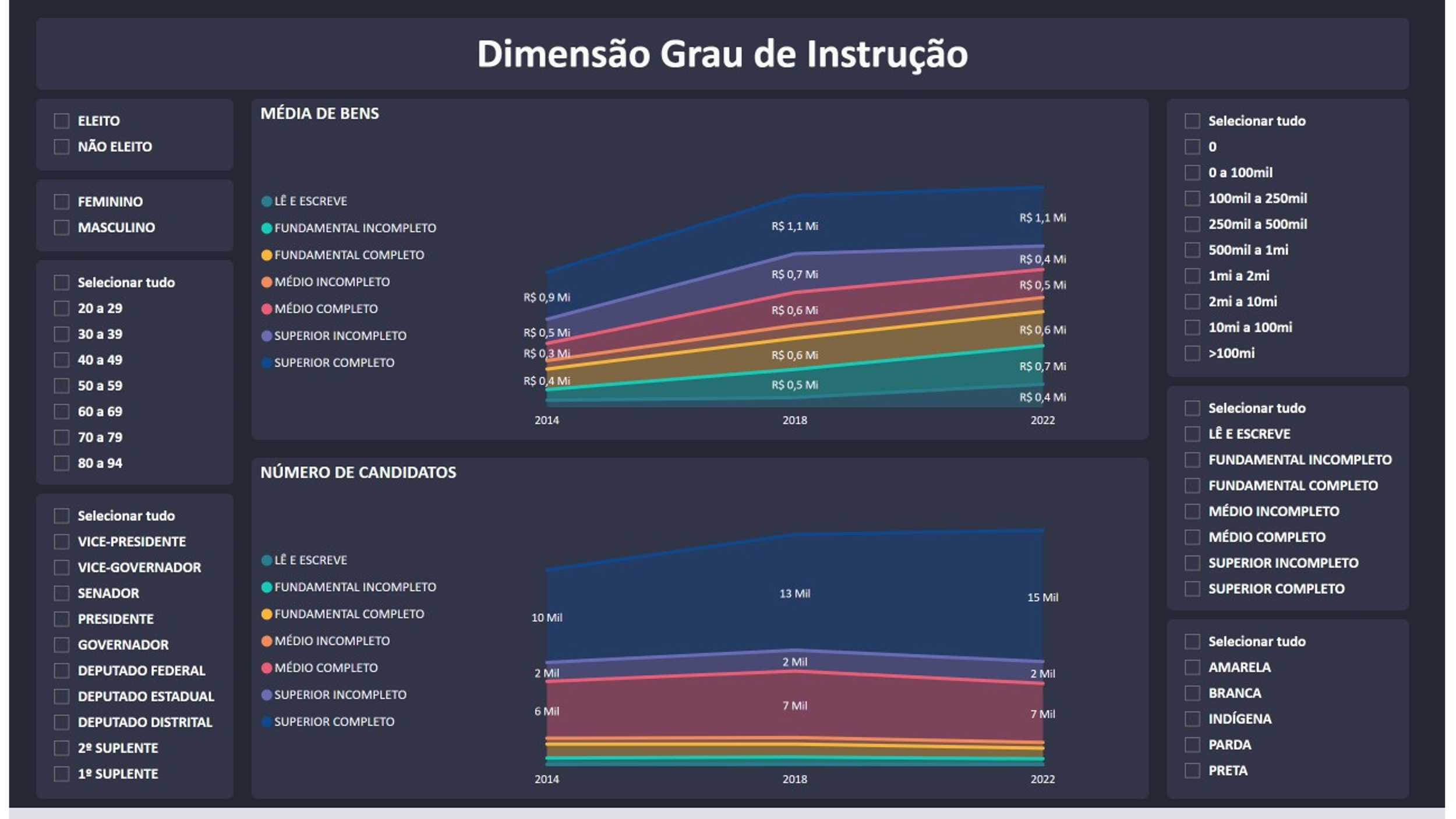


Para apresentar a dimensão cor/raça, foram utilizados dois gráficos de colunas clusterizados, representando o número de candidatos por cor/raça que se declararam em cada ano, um com a média de bens por ano e outro com o número de candidatos por ano, todas sensíveis às segmentações de dados propostas.

Esses gráficos mostraram que a maior parte dos candidatos se autodeclara branco, com uma crescente quantidade entre pardos e pretos, entretanto, a média de bens se mostra maior entre brancos e amarelos, apesar do baixo número de candidatos autodeclarados amarelos.

Quando os dados são segmentados pelos candidatos eleitos, verifica-se que a esmagadora maioria se autodeclara branco, e que a média de bens dos pardos eleitos é bem maior do que a média geral deste grupo.

## Dimensão grau de instrução



Para apresentar a dimensão grau de instrução, foram utilizados dois gráficos de área empilhados, ordenados pelo grau de instrução, um com a média de bens por ano e outro com o número de candidatos por ano, todas sensíveis às segmentações de dados propostas.

Estes gráficos mostraram que a quantidade de candidatos que possui nível superior completo é maior do que todos os outros níveis de instrução juntos, e vem aumentando ano a ano, o que mostra que o acesso à educação superior pode ter aumentado nos últimos anos, porém a média de bens desses candidatos, apesar de maior, não o é na mesma proporção.

Quando os dados são segmentados pelos candidatos eleitos, verifica-se que a esmagadora maioria possui o ensino superior completo.

## Dimensão localidade

Map

Description automatically generated

Para apresentar a dimensão localidade, foi utilizado um gráfico de mapa com bolhas representando a média de bens por localidade, sensível às segmentações de dados propostas.

Esse gráfico mostra os estados nos quais a média de bens declarados pelos candidatos é maior, mostrando que a região centro-oeste do Brasil é onde se concentram os candidatos com a maior média de bens, e entre os candidatos eleitos, o Mato Grosso se destaca, com candidatos com média de bens superior, porém há uma grande distorção quando desconsideramos os candidatos a governador, já que os candidatos daquele estado são os que apresentam maior média de bens.

Quando segmentamos os dados para candidatos ao legislativo estadual verificamos uma distribuição diferente, com Santa Catarina tomando a frente na média de bens.

# Apresentação e Conclusão

A partir da análise dos dados divulgados pelo TSE, pode-se concluir que a média de bens declarados pelos candidatos eleitos é, em geral, maior do que a média dos bens do universo total de candidatos. A realidade brasileira apresentada no *Global Wealth Report 2022* do banco *Credit Suisse*, pode-se verificar cerca de 77% dos brasileiros possuem patrimônio menor que 50 mil reais, cerca de 20% entre 50 mil e 500 mil reais e apenas 3% da população possui valores acima de 500 mil reais, entretanto, entre os candidatos eleitos a proporção é invertida, com apenas 8% de eleitos entre 0 e 50 mil, 27% entre 50 e 500 mil e 65% dos eleitos com mais de 500 mil reais, o que mostra que o patrimônio pessoal é um fator crucial para que o candidato seja eleito, perpetuando a desigualdade social.

Foi observado também que a média de bens de candidatos do sexo masculino é maior do que a média de candidatas do sexo feminino, e que a média de bens e candidatos brancos é maior do que a média de bens de candidatos de outras cores/raças, demonstrando um grande problema de desigualdade de gênero e representatividade no Brasil. Quanto ao grau de instrução, há uma prevalência na média de bens de candidatos que possuem o Ensino Superior completo, porém os demais graus de instrução, distribuição na média de bens é parecida.

Entretanto, foi observado que as informações disponibilizadas pelos TSE são prestadas pelos próprios candidatos, o que torna os dados inconsistentes. Pode-se verificar situações em que o candidato descreve o bem como um carro ou terreno cujo valor seria alto, porém, ao declarar o valor, utiliza valores simbólicos, como R$ 1,00. Não há penalidade para o candidato que faz tal declaração, a Justiça Eleitoral não faz uma avaliação dos dados para garantir sua correção, porém é de interesse do candidato declarar patrimônio suficiente para realizar o autofinanciamento de sua campanha. Uma solução para este problema seria juntar os dados da base do Imposto de Renda no momento do registro da candidatura, já que os dados apresentados à Receita Federal são efetivamente fiscalizados.

Como sugestão para evolução do trabalho ou trabalhos futuros, a base de dados de prestação de contas eleitorais poderia ser utilizada para verificar os valores efetivamente gastos pelos candidatos em cada eleição, e, a partir do cruzamento dos dados de patrimônio declarado, poder-se-ia analisar se os candidatos que possuem maior patrimônio efetivamente gastam mais, além de poder-se verificar qual o gasto médio de candidaturas bem-sucedidas, o “preço” de uma cadeira no legislativo por exemplo.

# Referências Bibliográficas

Candidatos - 2014: banco de dados. <https://dadosabertos.tse.jus.br/dataset/candidatos-2014>. Acesso em: 06 de dezembro de 2022.

Candidatos - 2018: banco de dados. <https://dadosabertos.tse.jus.br/dataset/candidatos-2018>. Acesso em: 06 de dezembro de 2022.

Candidatos - 2022: banco de dados. <https://dadosabertos.tse.jus.br/dataset/candidatos-2022>. Acesso em: 06 de dezembro de 2022.

Credit Suisse Research Institute**. Global Wealth Report 2022**. Disponível em: <https://www.credit-suisse.com/about-us/en/reports-research/global-wealth-report.html>. Acesso em: 06 de dezembro de 2022.

Duarte, Luiz, **O que é Scrum?** Disponível em: <https://www.luiztools.com.br/post/o-que-e-scrum/>. Acesso em: 12 de novembro de 2022

Siqueira, Daniel. Modelagem de banco de dados: entidades, relacionamentos e atributos. **Alura,** 06 de maio de 2022. Disponível em: <https://unibb.alura.com.br/course/modelagem-banco-dados-entidades-relacionamentos-atributos>. Acesso em 10 de outubro de 2022.

Siqueira, Daniel. Dashboard com Power BI: visualizando dados. **Alura,** 06 de julho de 2022. Disponível em: <https://unibb.alura.com.br/course/ power-bi-visualizando-dados>. Acesso em 15 de novembro de 2022.

Espinha, Roberto Gil. **O guia definitivo da metodologia ágil Scrum**. Disponível em: <https://artia.com/scrum/>. Acesso em :12 de novembro de 2022.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit:** The Definitive Guide to Dimensional Modeling. ,3ª edição, Wiley, 2013.