

# Plenário Virtual - Um Sistema Comparativo de Ideologia Política para Candidatos ao Legislativo Municipal de Belo Horizonte

Samuel S. C. Nascimento<sup>1</sup>, Thais R. M. B. Silva<sup>1</sup>

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas  
Universidade Federal de Viçosa - Campus UFV-Florestal (UFV)  
Caixa Postal 15.064 – 91.501-970 – Florestal – MG – Brazil

{samuel.nascimento, thais.braga}@ufv.br

***Resumo.** Este artigo apresenta um sistema Web que simula um Plenário Virtual para comparar as ideologias políticas de candidatos a posições legislativas municipais. O objetivo é fornecer uma plataforma interativa onde os usuários possam participar de uma sessão virtual do plenário e, ao final, comparar suas escolhas em projetos de lei com as de vereadores eleitos. Ao final, a comparação entre as escolhas do usuário com as dos vereadores fornece uma avaliação comparativa de sua ideologia política. O sistema inclui uma interface amigável, mecanismos de coleta e processamento de dados, e utiliza o algoritmo PCA na construção da análise comparativa. A eficácia e usabilidade do sistema foram avaliadas por meio de testes conduzidos com usuários reais.*

## 1. Introdução

A compreensão das questões políticas é fundamental no desenvolvimento da democracia e para a tomada de decisões informadas por parte da população. No entanto, essa tarefa costuma ser complexa e burocrática, o que dificulta a compreensão dos eleitores sobre a gestão de suas cidades e países. Muitas vezes é necessário um estudo aprofundado em ciência política para exercer o papel de cidadão de forma convicta [Monteiro et al. 2015]. Com o crescente volume de dados políticos disponíveis na Web, as questões políticas ligadas ao exercício democrático se tornaram ainda mais desafiadoras, já que o grande volume de dados dificulta ao usuário a filtragem ou o discernimento do conteúdo.

já que nem sempre as informações expostas são verídicas [Onoda and Neris 2019].

Nesse contexto, a Tecnologia da Informação (TI) pode desempenhar um papel importante no desenvolvimento de sistemas que facilitam a entrega de informações, podendo ajudar a melhorar a consciência política da população. Atualmente, devido ao enorme volume de dados oferecidos por sistemas distribuídos, o desenvolvimento de sistemas de análise política baseados em técnicas de raspagem de dados, mineração, processamento de linguagem natural, entre outras, têm entregue soluções promissoras e vem agregando cada dia mais à literatura.

O objetivo desse trabalho é o desenvolvimento de um sistema Web chamado Plenário Virtual, responsável por entregar, a partir de dados reais obtidos por meio de *Web Scrapping* e aplicações de algoritmos matemáticos, a proximidade de um possível candidato a vereador com vereadores de mandatos anteriores, utilizando como caso de

estudo a Câmara Municipal de Belo Horizonte no período de 2017 à 2020. A escolha pelo recorte temporal decorreu do parâmetro "Projetos de lei já tramitados". Isso quer dizer que, devido ao baixo número de projetos de lei efetivamente concluídos pelo legislativo da capital entre 2021 e 2023, optou-se por restringir a análise, ao tempo retratado.

O sistema desenvolvido neste artigo pode vir a ser uma ferramenta potencialmente valiosa para estudos voltados a área de análise política, contribuindo também para a literatura existente e ajudando na compreensão das questões políticas.

Este artigo é dividido em quatro seções. Os trabalhos relacionados, que abordam o desenvolvimento de sistemas que trabalham com dados políticos por meio de ferramentas de análise de dados, discutidos na seção 2. Em seguida, a seção 3 descreve os principais materiais utilizados, bem como todo o método de desenvolvimento deste trabalho. Logo após, a seção 4 aborda a metodologia utilizada para a obtenção dos resultados e explica toda a análise realizada com base nos resultados obtidos. Por último, a seção 5 demonstra pontos de melhoria a serem explorados em trabalhos futuros e conclui o artigo.

## 2. Trabalhos relacionados

Pesquisas recentes têm abordado o desenvolvimento de sistemas que utilizam das técnicas de análise de dados para facilitar a compreensão da população quanto aos feitos políticos que não são acessíveis por grande parte da população. É possível notar que os trabalhos abordados a seguir utilizam da construção de um sistema para atacar problemas relacionados a área da política, utilizando técnicas similares ao presente artigo para a obtenção dos dados. O presente estudo se diferencia dos relacionados porque procura simular um ambiente real, presente no dia a dia dos vereadores, com o intuito de testar os conhecimentos de candidatos a vereadores quanto a execução do trabalho que, se eleitos, terão que exercer. Além disso, o sistema aqui produzido, gera uma assimilação quanto a proximidade entre aquele candidato (participante da pesquisa) e os vereadores que já executaram tal trabalho no período de 2017 e 2020, para assim entregar à população por meio de um sistema Web, dados transparentes e reais sobre o posicionamento daquele candidato, podendo auxiliar na escolha dos representantes do legislativo do município.

[Costa and Gallegos 2021] desenvolveram um sistema para coleta e tratamento de textos envolvendo polarização política por meio da rede social Twitter. A análise envolveu verificação de textos quanto aos seus sentimentos, ou seja, positivos, negativos e neutros, em um momento sensível politicamente no Brasil, considerando a pandemia da COVID-19. Para realizar essa análise, foi utilizado o analisador léxico *Valence Aware Dictionary and sEntiment Reasoner* (VADER) [Hutto and Gilbert 2014]. Como resultado, concluíram que houve *tweets* em todos os estados da nação, sendo predominantemente originários do sudeste. Observou-se também que dentre os estados que mais se destacaram foram: Rio de Janeiro, São Paulo e Minas Gerais. Ao final, os autores identificaram que havia maior percentual de textos com sentimentos neutros.

*Democracy* é um jogo de simulação política que permite aos jogadores experimentarem as complexidades e os desafios de governar um país. Ele oferece uma experiência envolvente ao permitir que os jogadores tomem decisões políticas importantes e enfrentem as consequências delas, buscando equilibrar as necessidades da população, a estabilidade do país e sua própria popularidade como líder político [Martinelli 2022].

*Power & Revolution 2020 Edition*, também conhecido como *Masters of the World*,

é um jogo de simulação política desenvolvido pela Eversim [Fabris 2022]. Nele, os jogadores assumem o papel de líderes políticos e têm como objetivo governar um país, enfrentar desafios políticos e geopolíticos, e tomar decisões estratégicas para alcançar objetivos específicos.

Dito isso, o presente artigo se aproxima dos trabalhos referenciados pela temática política e tecnologias aplicadas, porém, se diferencia tendo em vista o público que irá utilizar o sistema.

### 3. Plenário Virtual

Definido todo o recorte e o problema que o sistema pensado para essa pesquisa precisaria resolver, o trabalho foi dividido em etapas para que se facilitasse todo o processo de desenvolvimento, obtenção dos dados, construção do banco de dados, redução de dimensionalidade e sistema *Web*.

Tendo em vista a referida necessidade de divisão, os próximos tópicos tratarão de maneira individualizada cada fase.

#### 3.1. Obtenção dos dados

O primeiro passo do projeto foi delimitar o escopo de trabalho. Foi definida a busca de projetos de leis com votos dos vereadores de Belo Horizonte/MG eleitos para o mandato de 2017 à 2020. Nesta oportunidade, após o filtro aplicado no site da Câmara Municipal de Belo Horizonte/MG (CMBH) <sup>1</sup> para encontrar projetos de leis no *status* "tramitados" foram identificados 1029.

Em seguida, a partir do contato com a assessoria da CMBH, se obteve a informação de que os dados de votação de cada vereador para os projetos de leis tramitados, informação crucial para o desenvolvimento da pesquisa, estavam na fase do procedimento legislativo denominado "Apreciação em Plenário". Esta fase é o processo onde os vereadores se reúnem em um Plenário, ou comissão, ou isoladamente, discutindo e votando em uma proposição, ou decidem sobre ela de outra forma.

A partir da triagem e identificação dos documentos que constavam da etapa de apreciação em plenário foi iniciado o processo de investigação de quais possíveis ferramentas poderiam ser utilizadas para se realizar a obtenção dos dados de forma automatizada através da técnica de *Web scrapping*. Foi escolhido o *framework* Puppeteer <sup>2</sup> por ser uma ferramenta que possui as funcionalidades básicas de *Web scrapping* necessárias ao trabalho, além de possuir e possuir comunidade ativa e uma documentação bem construída.

A ferramenta Puppeteer é uma Biblioteca Node.js que fornece uma API para um controle do navegador Google Chrome, que permite automatizar ações no navegador, sendo possível controlar o site da CMBH programaticamente. Assim, realizou-se tarefas como navegar por páginas da web, preencher filtros e capturar informações, ou seja, etapas da técnica de *scrapping* de dados. No entanto, importante ressaltar que, mesmo com a utilização da ferramenta, deparou-se com divergências de informações na interface

---

<sup>1</sup><https://www.cmbh.mg.gov.br/>

<sup>2</sup><https://pptr.dev/>

do site, porque apresentava um comportamento diferente quando acessado manualmente pelo usuário e quando o acesso ocorria via ferramenta de automatização.

O site da CMBH trabalha com uma estrutura de renderização *server side rendering*, onde o usuário faz uma requisição da página. A requisição é processada pelo servidor e entregue ao cliente sendo renderizada ao usuário final. Dentre essa estrutura, um ponto interessante a se ressaltar, é a utilização do Ajax <sup>3</sup>, uma ferramenta para comunicação assíncrona entre cliente e servidor, em que toda requisição feita pelo Ajax retorna um esqueleto HTML com todo o conteúdo, o que tornou em diversos momentos o tempo de resposta alto.

A falta de performance do site da CMBH foi um agravante na produção do algoritmo, pois em diversos momentos foi necessário inserir funções de espera de até 15 segundos a cada nova requisição realizada. Inclusive, muitas vezes as requisições levavam mais de 15 segundos e assim o algoritmo se encerrava imediatamente já que não era possível obter informações de hipertextos que ainda não foram carregados. Como é difícil mensurar qual o tempo maior de espera, bem como para cada segundo adicionado a cada etapa do processo aumentava ainda mais o tempo de execução do algoritmo, foi implementada uma lógica, qual seja, sempre que o programa se encerrasse de forma inesperada, a última lei consultada era retornada com o número da página onde ela se encontrava para que assim fosse possível continuar o processo a partir da página e lei retornada. Outro problema apresentado foi a indisponibilidade dos serviços de *downloads* dos arquivos que acontecia a medida que recebia várias requisições em um período pequeno de tempo, e novamente a solução para essa questão foi a inserção de funções de tempos para espaçar esses intervalos.

O algoritmo construído recebe como entrada a quantidade de projetos de leis por página e executava duas iterações, em que a primeira tinha como ponto de parada a quantidade de arquivos existentes (dado o filtro pesquisado) dividido pelo número de arquivos exibidos por página (7), esse resultado, caso fracionado, é arredondado para o valor acima. Dentro da primeira iteração uma outra iteração era feita tendo como ponto de parada o número de arquivos exibidos por página (7), a partir dessas iterações o algoritmo acessava lei a lei para baixar os arquivos que continham as votações. Porém, quando a página de uma lei específica era visitada, um botão de voltar a página anterior era exibido, botão esse que não era visível quando a página era visitada pelo algoritmo executado. Devido ao exposto, sempre que uma lei específica fosse visitada para se voltar a página anterior uma nova busca precisava ser feita e caso essa lei estivesse em páginas muito a frente, o algoritmo precisava chegar até essas páginas para assim continuar o processo, aumentando então consideravelmente o tempo de execução, o que foi considerado a principal dificuldade na obtenção de todos os dados necessários.

Após a execução do algoritmo, foram obtidos 410 arquivos, porém, ainda foi preciso consultar um a um, para verificar se em seu conteúdo continha as votações dos vereadores. Caso negativo, esse arquivo era excluído e, em caso positivo, era preciso mapear em qual página essa votação estava e renomear o nome desse arquivo para a seguinte maneira: "num\_projeto-ano\_projeto-pagina".

O papel de renomear os arquivos seguindo esse padrão foi pensado e efetivado

---

<sup>3</sup><https://api.jquery.com/category/ajax/>

para que o algoritmo construído posteriormente para a inserção dos dados em um banco de dados funcionasse da forma em que foi desenvolvido.

Ao final, a quantidade de arquivos foi para 262 e com todo o cenário preparado se deu início à fase de extração das informações e construção do banco de dados a ser utilizado pela aplicação.

### 3.2. Construção do Banco de Dados

Antes de se iniciar o desenvolvimento do banco de dados, foi necessário definir quais entidades seriam importantes para contemplar todas as funcionalidades do sistema. Sendo assim, com as entidades definidas se utilizou da versão gratuita do site dbdiagram <sup>4</sup>, para criar o diagrama de entidade relacionamento. Como mostra (Figura 1)

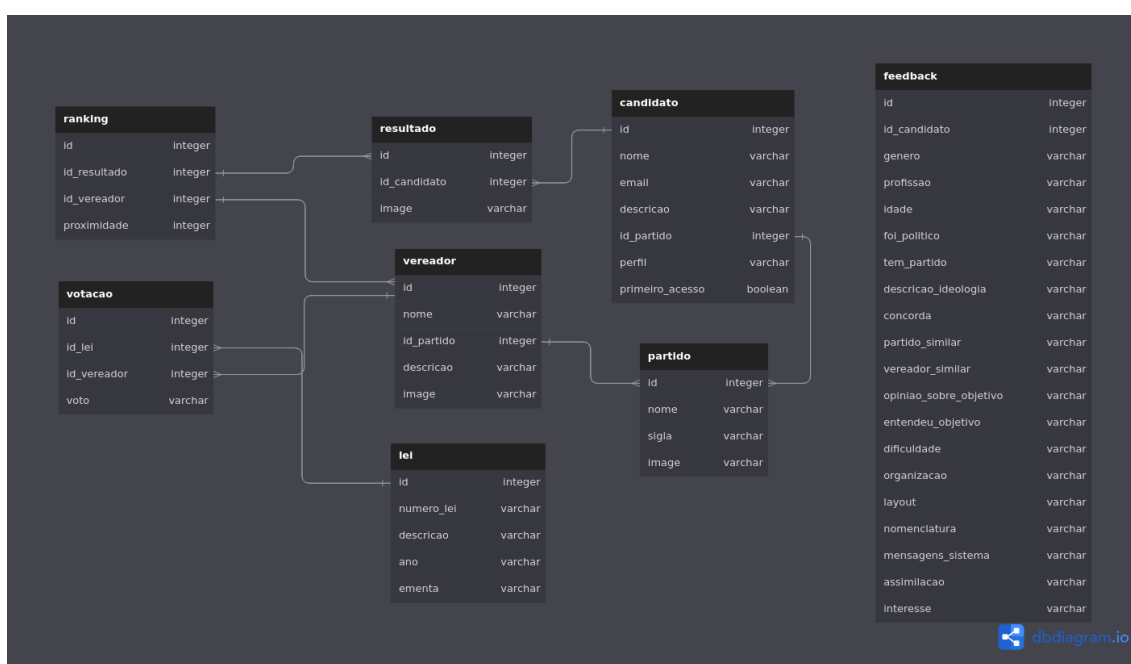


Figure 1. Diagrama Entidade Relacionamento

Com o diagrama estruturado a partir das entidades supracitadas, foi definida a utilização do banco em memória SQLite3 <sup>5</sup>. Como o foco principal estava na funcionalidade do sistema, optou-se por utilizar o SQLite3 que poupou bastante tempo já que ele é simples de instalar e configurar, não requer um servidor separado e não precisa de configurações complexas.

Para a obtenção dos valores, foi construído um novo algoritmo utilizando também a ferramenta Puppeteer, pois os dados relacionados à descrição e ementa precisavam ser obtidos no site da CMBH. O algoritmo acessa um diretório contendo todos os arquivos de votação e armazena em variáveis o referente ao número e ano do projeto de lei. Essas informações foram utilizadas no site da CMBH para aplicar os filtros e obter a descrição e ementa de cada uma das leis, e assim, a cada iteração uma inserção era feita na entidade

<sup>4</sup><https://dbdiagram.io/>

<sup>5</sup><https://www.sqlite.org/index.html>

**Lei**, obtendo como resultado final a entidade totalmente populada com o número, ano, descrição e ementa de todas as 262 leis.

Como o campo `idPartido` é uma chave estrangeira que referencia a entidade **Partido** foi necessário populá-la primeiro. (Figura 1)

Para a população dessa tabela foi realizada uma pesquisa no site do Tribunal Superior Eleitoral (TSE) <sup>6</sup>, em que para cada partido foi escrita uma consulta de inserção com nome, sigla e imagem (obtidas no google imagem).

Com a entidade **Partido** populada foi então possível popular a entidade **Vereador**, e para popular essa tabela as informações de nome, partido, descrição e imagem foram retiradas do site da CMBH sendo escritas as queries de inserção uma a uma.

Considerando o exposto, tendo as entidades **Lei**, **Vereador** e **Partido** populadas, foi então possível iniciar a inserção dos dados na entidade **Votação**. Os valores lei, vereador e voto estavam presentes dentro de cada um dos arquivos de votação, então, para recuperar as informações de dentro dos arquivos foi necessário utilizar a linguagem de programação Python. O motivo da utilização desta linguagem se deu pela facilidade em realizar manipulação de arquivos, já que o Python é muito utilizado na área de Ciência de Dados.

O primeiro passo para o desenvolvimento do algoritmo foi instalar as bibliotecas PyPDF2 <sup>7</sup> (biblioteca responsável por fornecer métodos para recuperação de informações em PDFs) e sqlite3 <sup>8</sup> (biblioteca do Python para manipulação de bancos SQLite). Utilizando a biblioteca PyPDF2 foi extraído de todos os projetos de lei o texto onde continham as votações. Com o texto extraído foi observado um padrão de caracteres antes e depois da tabela que continha os votos, e a partir deste padrão foi aplicada a lógica para pegar apenas os dados de votação. Após essa aplicação, o texto obtido foi tratado por meio de expressões regulares e construído um vetor de objetos, em que cada objeto obtido continha o nome do vereador, sigla do partido e voto (Sim, Não ou Abstenção).

Com o vetor de objetos construído foi utilizada a biblioteca sqlite3 para realizar a inserção dos dados na entidade **Votação**. Por fim, as entidades **Candidato**, **Resultado** e **Ranking** são utilizadas e preenchidas durante o uso do sistema *Web Plenário Virtual*.

### 3.3. Redução da dimensionalidade

Como descrito na introdução do presente artigo, o objetivo do sistema é entregar para o possível candidato a vereador da cidade de Belo Horizonte a proximidade com alguns vereadores que atuaram no mandato de 2017 e 2020. Essa proximidade se dá a partir da comparação entre a resposta do candidato a um formulário com projetos de leis diversos e as respostas dos vereadores.

Para uma melhor visualização dessas informações, foi necessário mostrar em um gráfico 2D o conteúdo de uma matriz com dimensão 42 X 15 que foi reduzida utilizando o algoritmo Principal component analysis (PCA) [Jolliffe 2002].

O algoritmo PCA é uma técnica de redução de dimensionalidade linear que pode

---

<sup>6</sup><https://www.tse.jus.br/>

<sup>7</sup><https://pypdf2.readthedocs.io/en/3.0.0/>

<sup>8</sup><https://docs.python.org/3/library/sqlite3.html>

ser utilizado para extrair informações de conjunto de dados muito grandes, reduzindo-o em um conjunto muito menor. A técnica no processo de redução procura manter as partes essenciais que possuem mais variação dos dados e remover as partes não essenciais com menor variação. Para trazer a solução ao problema apresentado por esse artigo, foi aplicado a técnica do PCA a partir da biblioteca do Python `sklearn`<sup>9</sup>.

Na primeira etapa do algoritmo é feito a junção das respostas do candidato com as respostas dos demais vereadores para as leis que o candidato respondeu no formulário. Após a junção é obtido uma matriz  $m \times n$ , onde  $m$  é o número de vereadores mais o candidato, ou seja  $41+1$  que é igual a 42 e  $n$  a quantidade de leis respondidas no formulário, ou seja 15 (esse número é definido nas regras de negócio do sistema), chegando então em uma matriz  $42 \times 15$ . Porém, os valores dentro de cada posição da matriz é uma *string* com os valores Sim, Não ou Abstenção, e como o algoritmo PCA executa a partir de valores numéricos foi necessário determinar valores para cada uma das respostas, sendo assim foi definido da seguinte forma:

valor em texto	valor em número
Sim	1
Não	-1
Abstenção	0

Definidos os valores numéricos uma iteração entre a matriz é aplicada alterando os valores em *string* para o seu respectivo valor numérico, e logo em seguida estabelecida a configuração de redução para tamanho 2, ou seja, reduzir a dimensão da matriz de 15 para 2 componentes, e assim o algoritmo é executado, retornando uma matriz  $42 \times 2$  onde na coluna 1 representa o valor onde o candidato/vereador está em relação ao eixo  $x$  e a coluna 2 representa o valor onde o candidato/vereador está em relação ao eixo  $y$ . Portanto, através da matriz retornada pelo algoritmo PCA um gráfico 2D é produzido tendo como legenda a sigla dos partidos antecedendo do símbolo que ele representa no gráfico. A cor cinza com opacidade para facilitar a visualização caso os pontos se sobreponham é definida para os vereadores e para o candidato o seu ponto foi aumentado e sua cor definida como vermelha para que fosse fácil o suficiente de identificar em qual local e a quais vereadores o candidato está próximo. Sendo assim ao final da execução do algoritmo uma imagem de extensão *jpeg* contendo o resultado é salva no diretório imagem do projeto, para posteriormente ser salvo em um servidor na nuvem.

### 3.4. Desenvolvimento do Sistema *Web*

O Plenário Virtual desenvolvido para resolver o problema abordado nesse artigo, consiste de um sistema distribuído que simula a atuação de um plenário real. O sistema *Web* foi pensado para trabalhar com o tipo de renderização SPA (*single page application*) por parte do *front-end* e para isso foi utilizado o *framework* `VueJS`<sup>10</sup>.

Por parte do *Back-end* a decisão foi baseada na linguagem utilizada para solucionar o problema da redução de dimensionalidade, portanto o Python juntamente do

<sup>9</sup><https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

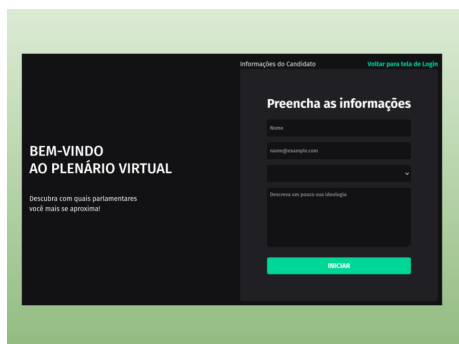
<sup>10</sup><https://vuejs.org/>

*micro-framework* Flask <sup>11</sup> que traz facilidade no desenvolvimento de uma API Rest - a interface utilizada para construção do servidor.

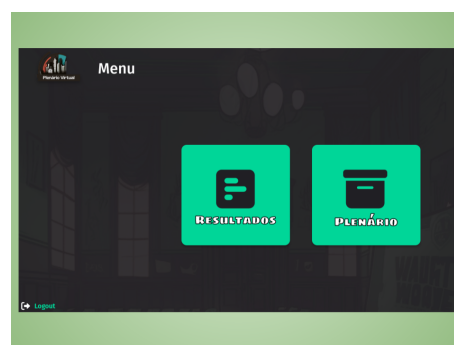
Além da construção do *front-end* e *back-end*, para que o sistema fosse acessível aos usuários, utilizou-se os serviços da *Amazon Web Service* (AWS) <sup>12</sup>, configurando um *bucket* S3 para armazenamento das imagens geradas pelo algoritmo de redução da dimensionalidade, armazenamento dos arquivos de *front-end* gerados pelo *build* do VueJS para acesso via url pelos usuários e a configuração de um EC2 (um serviço da AWS que disponibiliza uma máquina virtual Linux para hospedagem de aplicações) para armazenar o *back-end* e o banco SQLite.

A escolha da utilização dos serviços da AWS se deu pelo fato de que por 12 meses é oferecido gratuitamente 750 horas/mês do uso de instâncias EC2 e 5GB de armazenamento no S3, contendo o necessário para todas as necessidades do sistema. Um problema enfrentado na construção do sistema foi a retro-compatibilidade do *front-end* com as plataformas *mobile*, ou seja, o sistema é acessível por um navegador, porém a sua interface não se adequa aos dispositivos móveis sendo utilizável de fato apenas em *desktops*, ou *notebooks*.

A aplicação desenvolvida possui, no primeiro acesso do usuário, um cadastro prévio, portanto, a primeira tela visualizada - a ser reproduzida a seguir - entrega a funcionalidade de autenticação e cadastro, onde caso o usuário já possua seu e-mail cadastrado ele apenas precisa preencher o campo de e-mail para ter acesso as funcionalidades do sistema. A utilização da funcionalidade de cadastro e autenticação se deu pela demanda de se armazenar os resultados e o *feedback* dos usuários para uma análise e uso destas para a construção dos resultados do artigo. (Figura 2)



**Figure 2. Tela de formulário dos projeto de leis**



**Figure 3. Tela de resultados**

Já dentro do sistema, o usuário é levado a um menu (Figura 3) com as opções "resultado" e "plenário". A opção plenário demonstra uma breve explicação sobre como o formulário com os projetos de leis devem ser respondidos para aumentar a acurácia dos resultados, e após a confirmação, há o encaminhamento para a tela de formulário com os projetos de leis propostos a serem respondidos, enquanto a opção resultado envia o usuário a uma tela com todas as informações preenchidas no *feedback* e aos resultados obtidos após respondido o formulário.

<sup>11</sup><https://flask.palletsprojects.com/en/2.3.x/>

<sup>12</sup><https://aws.amazon.com/pt/>



Entrando mais a fundo sobre a interface onde o questionário é construído, é importante ressaltar a forma de construção da lógica aplicada a essa funcionalidade do sistema. Para construir o formulário (Figura 4), uma requisição para a API executa o caso de uso "gerar formulário de leis". Esse caso de uso tem como regra de negócio a entrega de 15 projetos de leis aleatórios provindos da entidade "Lei" do banco de dados SQLite. O número 15 foi escolhido, pois durante a fase de testes da aplicação (onde o número inicial era de 30 leis), percebeu-se um tempo considerável para a resposta de um projeto de lei, pois cada projeto demanda uma leitura prévia do arquivo que contém o texto inicial disponibilizado no site da CMBH.

**Avalie e vote as propostas de leis a seguir como se estivesse participando do plenário Câmara de Vereadores**

**1) PL 474-2018** [Link para tramitação CMBH](#)

**Ementa:**  
Torna obrigatória a instalação de grades ou redes de proteção em janela, sacada, mezanino e varanda de clínica, consultório, hospital e alas pediátricas.

**Assunto:**  
Obrigatoriedade, instalação, grade, rede, segurança, proteção, vão, varanda, estabelecimento de saúde, clínica, hospital, criança, requisito, infração, penalidade, multa, cassação,

Sim  
 Não  
 Abstenção

**2) PL 716-2019** [Link para tramitação CMBH](#)

**Ementa:**  
Promove a revisão anual da remuneração dos servidores da Câmara Municipal de Belo Horizonte.

**Assunto:**  
Organização administrativa, servidor, revisão, remuneração, reajustamento, percentual, benefício, vale refeição, vencimento, remuneração | Câmara Municipal de Belo Horizonte

**Figure 4. Tela de formulário dos projeto de leis**

Como forma de agilizar a leitura desse arquivo, foi adicionado à interface um botão que redireciona o usuário para a página onde ele se encontra. Visto que a leitura prévia e cautelosa de todas as informações impacta diretamente na eficácia do resultado final, a quantidade de 30 projetos de leis levou a um tempo muito maior do que esperado. Isso obrigou a redução dos projetos pela metade, já que esse volume implicaria na desistência dos usuários quanto à utilização da aplicação, ou respostas aleatórias para simplesmente finalizar o formulário.

A escolha por recuperar projetos aleatórios a cada execução se deu pelo motivo de tornar o processo mais dinâmico e evitar a quebra do caráter científico dos resultados, ao permitir que o sistema fosse utilizado diversas vezes por um mesmo usuário entregando resultados diversos para cada formulário respondido evitando respostas enviesadas.

Ao finalizar o preenchimento de todo o formulário, uma requisição é feita para o servidor que executa o caso de uso "gerar resultado", esse caso de uso foi construído a partir do algoritmo de redução de dimensionalidade desenvolvido em passos anteriores, e como a parte mais complexa já estava pronta foi preciso realizar pequenos ajustes como: trocar os dados fictícios pelos dados reais entregues pelo *front-end*, armazenar os resulta-

dos obtidos na entidade **Resultado**, fazer *upload* no S3 da imagem gerada pelo algoritmo PCA e também adicionar uma nova função responsável pela construção de um ranking com os vereadores mais próximos, sendo do primeiro mais próximo até o decimo mais próximo, como mostra a (Figura 5)



Figure 5. Tela de resultados

Nesse caso de uso, é fundamental destacar a concordância entre a lógica do método de construção do ranking e o planejamento realizado. A matriz de pontos resultante do algoritmo PCA foi utilizada para calcular a distância euclidiana, permitindo assim obter a distância entre o ponto referente ao usuário no plano cartesiano e os demais pontos no gráfico.

Portanto, a partir da aplicação dessa teoria, o método de criação do ranking resulta em um vetor ordenado com os dez vereadores que mais se aproximaram dos votos do candidato. Com esses dados em mãos, o caso de uso os salva na entidade **Ranking** e os integra ao objeto resultante. Dessa forma, tanto a imagem quanto essa informação são disponibilizadas na tela de resultados no *front-end*.

Neste ponto do sistema ao usuário resta clicar no botão "Realizar Feedback" para enfim responder um questionário (Figura 6) com perguntas referentes a usabilidade e eficácia do sistema.

**Nos dê seu Feedback**

Formulário de Feedback

**Preencha as informações**

Gênero  
Selecione o gênero

Profissão  
Profissão

Idade  
Selecione a idade

Já atuou como político?  
 Sim  
 Não

É filiado em algum partido político?  
 Sim  
 Não

Qual ideologia você entende que o sistema apontou para você?  
 Descreva sobre...

**Figure 6. Formulário de feedback**

Ao fim do questionário o usuário é retornado para a tela de menu, podendo então responder novamente à outros formulários com novos projetos de lei, ou acessar seus resultados. Caso o usuário acesse a tela de resultados, o questionário de feedback com as respostas são visualizados, como também uma lista com o link para cada um dos resultados atingidos (Figura 7).

**Descrição sobre ideologia**  
Ideologia de esquerda

**Samuel Silva**  
samuel55@gmail.com  
Masculino  
Desenvolvedor  
Entre 18 e 30 anos

Historico de resultados	Feedback	Feedback - Sobre Usabilidade
Resultado 0	Já atuou como político? Não	Entendeu facilmente o propósito do sistema? Não
Resultado 1	É filiado em algum partido político? Não	Encontrou facilmente as opções pretendidas (facilidade de utilização)? Medio
Resultado 2	Qual ideologia você entende que o sistema apontou para você? De direita	Organização das informações Medio
Resultado 3	Qual partido você considera mais próximo da sua ideologia política? PT	Mensagens do sistema Confuso
Resultado 4	Qual vereador você considera mais próximo de sua ideologia política? Aurea Carolina	Layout das telas Claro
	Você acha que o programa atingiu o objetivo para o qual foi desenvolvido? Sim, eu faria de forma diferente	Nomenclatura utilizada nas telas (nome de comandos, títulos, campos, etc.) Claro
		Assimilação das informações Facil
		No geral, o uso do sistema foi Interessante

**Figure 7. Tela de resultados atingidos**

#### 4. Avaliação e Resultados

A aplicação *Web* desenvolvida com o objetivo de entregar ao candidato a vereador da cidade de Belo Horizonte a proximidade dele com os demais vereadores que atuaram no mandato de 2017 à 2020, foi avaliado em termos de usabilidade e eficácia. Para o desenvolvimento desse projeto considera-se usabilidade a facilidade e eficiência com que os usuários podem interagir com o sistema. Isso envolve a clareza da interface, a intuitividade na navegação, a disponibilidade de recursos e a facilidade de compreensão das funcionalidades disponíveis.

Por outro lado, a eficácia diz respeito à capacidade do sistema de cumprir seu propósito e fornecer resultados compreensíveis e de concordância dos usuários. No caso de um plenário virtual, a eficácia envolve aspectos como a correta simulação dos processos e procedimentos de um plenário real.

##### 4.1. Metodologia de Análise Qualitativa do Software

A metodologia adotada para este estudo envolveu a seleção de usuários para a realização do teste aplicado pelo Sistema Web Plenário Virtual. Para isso, foram convidados participantes de grupos diversos, levando em consideração a diversidade de gênero, profissão e idade. O grupo foi composto por 20 pessoas, porém apenas 16 delas realizaram os testes dentro do prazo estabelecido. (Tabela 1).

<b>Profissão</b>	<b>Qtd Gênero Masculino</b>	<b>Qtd Gênero Feminino</b>	<b>Qtd Outros</b>
Advogado(a)	1	2	0
Arquiteto(a)	0	1	0
Autônomo(a)	1	0	0
Bancário(a)	1	0	0
Consultor(a) de vendas	1	0	0
Desenvolvedor(a)	3	0	0
Estudante	1	0	0
Professor(a)	0	2	0
Secretário(a)	0	1	0
Analista	1	0	0
Aposentado(a)	1	0	0
<b>Idade</b>	<b>Qtd Gênero Masculino</b>	<b>Qtd Gênero Feminino</b>	<b>Qtd Outros</b>
Entre 18 e 30	6	3	0
Entre 30 e 40	3	2	0
Entre 50 e 60	0	1	0
Mais de 60	1	0	0
<b>Total</b>	10	6	0

**Table 1. Tabela com os recortes de gênero, profissão e idade**

Após a seleção desse grupo de pessoas, um grupo no WhatsApp foi formado e as seguintes instruções foram enviadas:

”Para a realização do teste, solicitamos que reserve de 20 a 30 minutos em um ambiente calmo e livre de interferências. O teste deve ser realizado em computadores

(desktops ou notebooks). Certifique-se de que seu computador esteja conectado à internet. É importante ler com cuidado cada um dos projetos de leis apresentados e responder com o máximo de veracidade possível. E por fim, acesse o link enviado no final desta mensagem para ter acesso ao sistema web.”

O teste com as 16 pessoas foi conduzido de maneira remota e individual. A escolha das pessoas seguiu os recortes mencionados e o uso do sistema consistiu na resposta dos candidatos a 15 projetos de leis tendo como resposta para cada questionamento os valores **sim**, **não** e **abstenção**. Após houve um questionário final que traz perguntas sobre o uso geral do sistema.

Essas perguntas foram aplicadas para avaliar a usabilidade e eficácia do sistema, sendo disponibilizadas 18 questões de caráter obrigatório. Dessas, 8 estavam relacionadas à usabilidade e foram na modalidade “fechada”. As outras 10 questões abordavam a eficácia, sendo 8 “fechadas” e 2 “abertas”. É importante destacar que as duas questões “abertas” tinham como objetivo obter o entendimento dos usuários sobre a adequação da ideologia apresentada a eles pelo sistema, bem como coletar opiniões e sugestões relacionadas ao sistema.

Para a coleta de dados, utilizou-se da ferramenta DBeaver<sup>13</sup>, que é uma interface de banco de dados SQL de código aberto, amplamente utilizada para gerenciar e consultar bancos de dados relacionais. Por meio dessa ferramenta, foi possível ler os dados disponíveis pelo arquivo gerado pelo SQLite e, a partir da tabela “Resultado”, retirar as informações relevantes sobre o desempenho e a usabilidade do Sistema Web Plenário Virtual durante os testes realizados pelos usuários.

Dessa forma, a combinação da disponibilização do sistema web aos usuários para acesso remoto e a utilização da ferramenta DBeaver para a coleta de dados contribuiu para uma avaliação eficiente e detalhada sobre um comparativo da ideologia política dos candidatos ao legislativo municipal.

## 4.2. Teste de Usabilidade

Conforme critério de metodologia, foi possível identificar os resultados da (Figura 8).

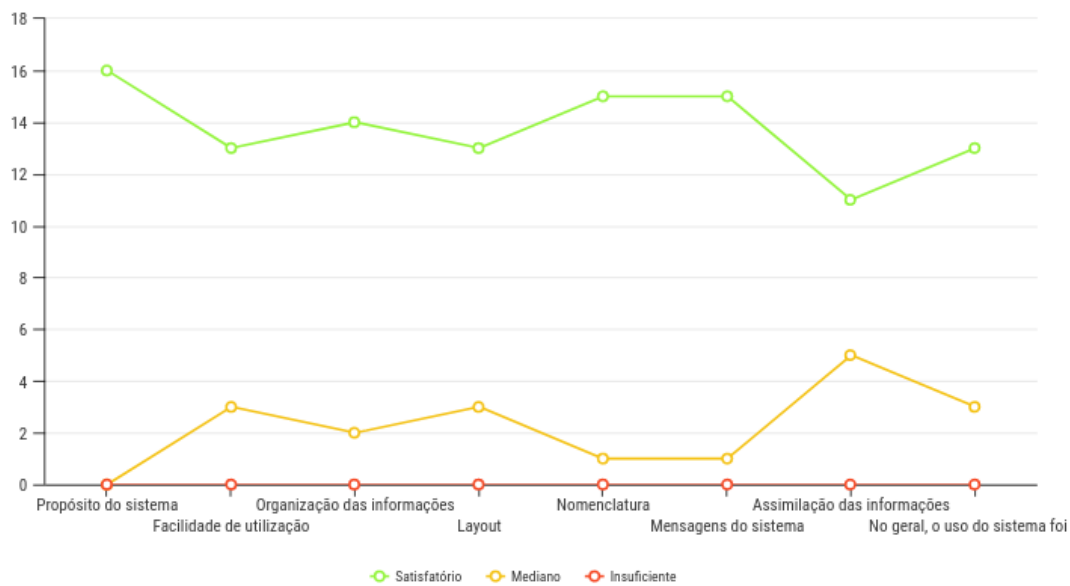
Os resultados obtidos no teste de usabilidade revelaram que a todos os 16 participantes compreenderam facilmente o propósito do sistema, já que responderam “Sim” à pergunta sobre a compreensão do propósito (ou seja, “Entendeu facilmente o propósito do sistema?”).

No quesito a facilidade de utilização, 13 pessoas responderam que encontraram as opções pretendidas de forma fácil, enquanto 3 participantes avaliaram como “Médio”. Em relação à organização das informações, a maioria respondeu positivamente, com 14 participantes considerando a organização como “Bom” e 2 avaliando como “Médio”. Em relação ao layout das telas, 13 pessoas consideraram como “Claro” e 3 como “Médio”. A nomenclatura utilizada nas telas também foi bem avaliada, com 15 participantes considerando como “Claro” e apenas 1 como “Médio”.

No que diz respeito às mensagens do sistema, 15 participantes consideraram como “Claro” e 1 como “Médio”. A assimilação das informações apresentou um resultado um

---

<sup>13</sup><https://dbeaver.io/>



**Figure 8. Gráfico de resultados de usabilidade**

pouco diferente, já que 11 pessoas avaliaram como "Fácil" e 5 como "Médio". Em geral, o uso do sistema foi bem avaliado, com 13 participantes considerando-o "Interessante" e 3 avaliando como "Legal". Não houve relatos de monotonia ou desinteresse no uso do sistema.

Esses resultados indicam que o sistema apresentou uma boa usabilidade, com a maioria dos participantes conseguindo compreender facilmente o seu propósito. As avaliações positivas em relação à organização das informações, nomenclatura e mensagens também indicam que houve uma boa experiência geral de uso por parte dos participantes do teste.

No entanto, as respostas variadas sobre a assimilação das informações sugerem que ainda existem pontos de melhoria para tornar o sistema ainda mais intuitivo e de fácil entendimento para todos os usuários. Uma das razões para essas respostas variadas foi a dificuldade em encontrar o botão de "realizar feedback". Durante a análise dos dados coletados, foi observado que das 16 pessoas que realizaram o teste, 5 não responderam inicialmente o formulário de feedback.

As 5 pessoas foram questionadas sobre os motivos pelos quais não encerraram corretamente a atividade e todas elas relataram que não tinham percebido a existência do botão de "realizar feedback". Assim, após esclarecimentos retornaram ao sistema para encerrar a pesquisa. Esse resultado ressalta a importância de aprimorar a visibilidade e a usabilidade do botão de "feedback", a fim de garantir que todos os usuários tenham conhecimento e acesso a essa etapa crucial de avaliação do sistema.

Além disso, ao analisar as opiniões e feedbacks fornecidos pelos usuários por meio do campo aberto disponibilizado, identificamos alguns desconfortos (Figura 9) que serão abordados como pontos de melhoria em futuras atualizações do sistema.

---

Atingiu sim. Poderia salvar progresso quando começou a responder, pois perdi meu progresso anterior.

---

Em parte sim, mas acredito que uma curadoria melhor nos projetos de lei iria melhorar os resultados.

---

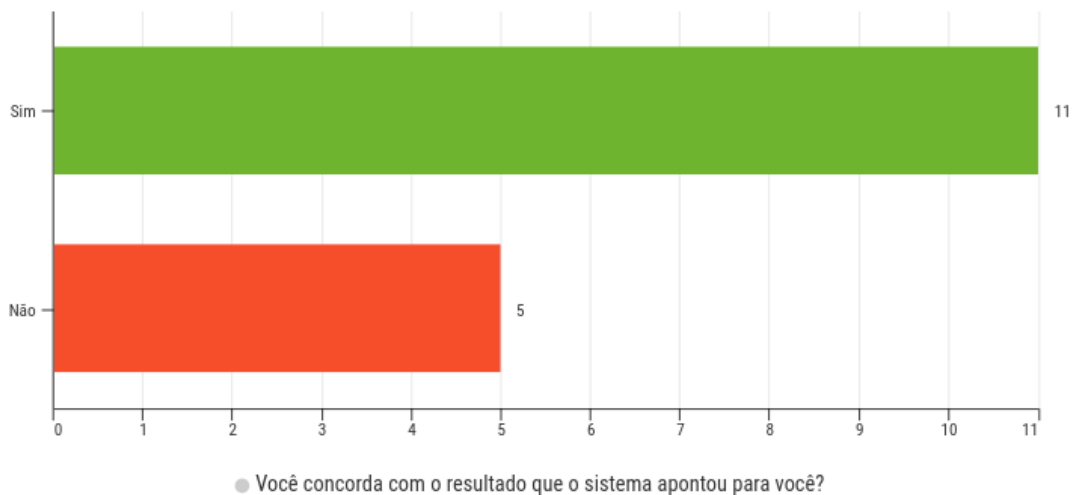
**Figure 9. Recorte do campo aberto de opinião e feedback**

Ao analisar os recortes de gênero, profissão e idade, observou-se que a única variação significativa estava relacionada às respostas sobre a assimilação das informações. Foi possível perceber que as pessoas que relataram dificuldade na visualização do botão "realizar feedbacks" pertencem à faixa etária de 18 a 30 anos, que inclusive foi majoritária durante o curso da utilização do sistema.

### 4.3. Eficácia

Os resultados mostraram que o sistema foi capaz de identificar com sucesso a coerência com as ideologias políticas dos candidatos previamente apontadas, com uma precisão média de 68.75%, uma vez que dentre as 16 pessoas participantes, 11 marcaram "Sim" para a pergunta "Concorda com o resultado que o sistema apontou?".

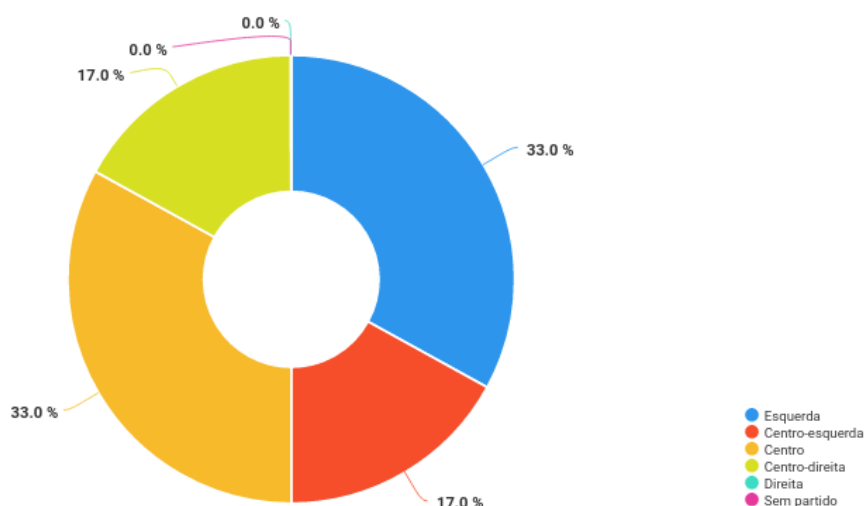
Paralelamente ao exposto, em análise dos tópicos "Descrição da ideologia" com "Partido similar" o mesmo resultado foi obtido, ou seja, 11 das 16 pessoas tiveram compatibilidade entre aquilo que entende enquanto sua ideologia política, com o resultado do sistema. (Figura 10)



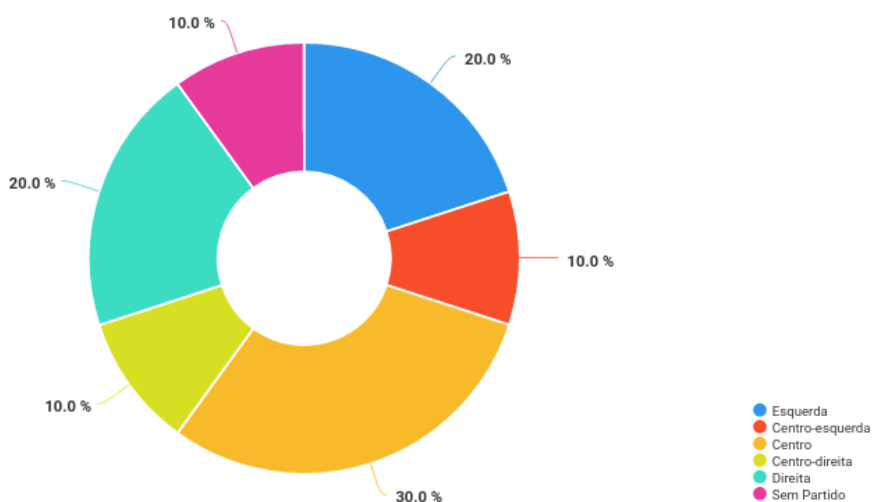
**Figure 10. Gráfico de coerência dos dados de resultado do sistema**

Além disso, o sistema foi capaz de identificar uma diferença ideológica entre o recorte de gênero feminino e masculino dentre os participantes, uma vez que as mulheres apresentaram aproximadamente 33% de proximidade a vereadores de partidos da esquerda, 17% de proximidade de vereadores centro-esquerda, 33% de vereadores de centro e 17% de centro direita (Figura 11).

De maneira diversa, para os homens o resultado foi: 20% apresentaram proximidade com vereadores de esquerda, também 20% apresentaram proximidade com os vereadores de direita, 10% apresentaram proximidade com vereadores de centro-esquerda e centro-direita (10%), enquanto o sistema detectou que 30% se aproximava de vereadores de centro. No caso do recorte masculino observou-se que 10% dos votantes se abstiveram de responder a questão.(Figura 12)



**Figure 11. Gráfico de resultados da aproximação ideológica para o gênero feminino**



**Figure 12. Gráfico de resultados da aproximação ideológica para o gênero masculino**

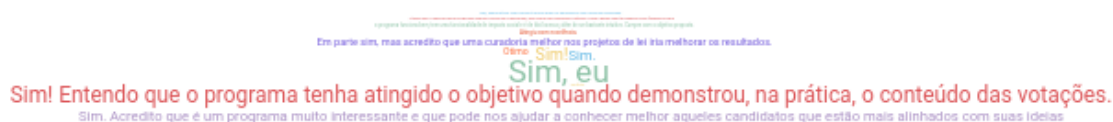
Quanto a análises sobre o recorte de idade, é possível perceber que no questionamento que traz "Qual partido você mais se identifica?", o Partido dos Trabalhadores (PT) recebeu 50% de proximidade enquanto 10% firmaram proximidade com o Partido Liberal (PL), 10% Podemos (PODE), 10% União, 10% Partido Socialista e Liberdade (PSOL) e



10% alegaram não se identificaram com nenhum partido, para a faixa etária de 18 a 30 anos.

No tocante as seguintes perguntas: "Já atuou como político?" e "É filiado em algum partido político?", 100% dos participantes alegaram "Não" para ambas as questões. E, para a pergunta "Qual vereador você considera mais próximo de sua ideologia política?", a maioria das pessoas, no percentual aproximado de 31%, afirmaram se tratar de Áurea Carolina que, a época, era filiada ao PSOL.

Quanto a eficácia do sistema é possível observar os resultados conforme (Figura 13), em que boa partes dos participantes comentaram no campo aberto boas opiniões quanto aos objetivos do sistema, bem como sugestões conforme (Figura 9).



**Figure 13. Nuvem de palavras com os comentários deixados pelos participantes**

Os resultados do teste de eficácia aplicado no sistema mostram que ele cumpre com o objetivo para que foi desenvolvido, sendo capaz de fornecer informações úteis sobre o alinhamento ideológico do candidato apresentado em suas eleições com vereadores da cidade de Belo Horizonte. Entende-se assim, que o sistema desenvolvido pode ser útil para os formuladores de políticas, tomadores de decisões e também ajudar os eleitores a tomar decisões mais informadas nas eleições municipais e promover a transparência na política local.

Em resumo, os resultados deste artigo indicam que o sistema web Plenário Virtual desenvolvido é fácil de usar e entender, e é eficaz na identificação de similaridade ideológicas entre candidatos e vereadores, muito embora presentes pontos de melhoria.

## **5. Conclusão e Trabalhos Futuros**

Este artigo teve como objetivo desenvolver um sistema *Web* responsável por entregar, a partir de dados reais obtidos através de processos de mineração de dados da *Web*, e aplicações de algoritmos matemáticos, um resultado sobre a proximidade a partir de votações reais de um candidatos a vereador do município de Belo Horizonte com vereadores eleitos no município no mandato de 2017 á 2020, sendo avaliado a partir dos resultados a eficácia quanto a alinhamento ideológico e usabilidade do sistema. Os resultados indicam que o sistema desenvolvido cumpriu com alta eficácia o que pretendia e é fácil de usar e entender.

A facilidade no uso do sistema é um marco importante para sua utilidade prática. Essa afirmação é comprovada a partir dos resultados apresentados na seção anterior, o que sugere que o sistema é acessível para um público amplo, incluindo cientistas políticos, analistas políticos e até mesmo a eleitores. Além disso, como os resultados do questionário de eficácia trouxeram uma alta precisão na coerência ideológica entre candidato e vereadores, pode-se afirmar que o sistema é capaz de entregar dados importantes para análises políticas e oportunas.

O sistema *Web Plenário Virtual* desenvolvido neste estudo pode ser uma ferramenta interessante para cientistas políticos e tomadores de decisões na compreensão das tendências políticas. No entanto, este artigo tem algumas limitações que precisam ser consideradas. Em primeiro lugar, os resultados foram obtidos a partir de um grupo de pessoas que não eram candidatos a vereadores. Portanto, para trabalhos futuros é sugerido avaliar a eficácia do sistema em grupo de futuros candidatos a vereadores, inclusive em épocas eleitorais, sendo interessante também incrementar ainda mais a base de dados com projetos de leis mais recentes e até de mandatos mais antigos.

## References

- Costa, P. and Gallegos, L. (2021). Sistema para Coleta e Tratamento Textos Brasileiros sobre Polarização Política. Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação (SBSI), 17<sup>o</sup> edition.
- Fabris, P. (2022). Jogos Políticos – Os Melhores Jogos De Simulação Política No PC 2022. Married Games.DICAS/GUIAS. 07 de abril de 2022. Disponível em: <https://marriedgames.com.br/dicas-guias/jogos-politicos/>. Acesso em: 29/06/2023.
- Hutto, C. and Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216-225.
- Jolliffe, I. (2002). Principal Component Analysis. 2nd ed. Springer.
- Martinelli, F. (2022). Democracy 4 — Game de simulação política em estado da arte. Folha Vitoria. Games. 17 de Janeiro de 2022. Disponível em: <https://www.folhavitoria.com.br/games/noticia/01/2022/democracy4-steam>. Acesso em: 29/06/2023.
- Monteiro, L., Moura, J., and Lacerda, A. (2015). Teorias da democracia e a práxis política e social brasileira: limites e possibilidades. SciELO.
- Onoda, C. and Neris, P. (2019). Estado, democracia e tecnologia: conflitos políticos e vulnerabilidade no contexto do big-data, das fake news e das shitstorms. SciELO.