

# *Comissão Eleitoral Calanguística 2024*

Ata de Resultados - Eleições 2024 1º Turno

Luiz Mario Andrade, Larissa Tunes e Carluxo Moura

13 de maio de 2024

## Sumário

<b>1</b>	<b>Introdução - Descrição do Processo e Escolha Social</b>	<b>2</b>
1.1	Início do Processo . . . . .	2
1.2	Segundo Turno . . . . .	8
1.3	Escolha do Público . . . . .	8
<b>2</b>	<b>Metodologia</b>	<b>8</b>
2.1	Introdução - Variável, Distribuição e Medidas . . . . .	8
2.1.1	Variável . . . . .	8
2.1.2	Distribuições de Frequência . . . . .	9
2.1.3	Medidas de Posição e Medidas de Dispersão . . . . .	9
2.2	Critério de Contagem dos Votos e Escolha das Músicas Vencedoras . . . . .	11
2.3	Simulação da Eleição . . . . .	12
2.3.1	Primeiro Passo: . . . . .	12
2.3.2	Segundo Passo: . . . . .	14
2.3.3	Terceiro Passo: . . . . .	15
<b>3</b>	<b>Análise da Votação e Resultado</b>	<b>15</b>
3.1	Resultados Lista Geral . . . . .	17
3.2	Resultados - Lista Brasília . . . . .	18
<b>4</b>	<b>Apêndice</b>	<b>20</b>
4.1	O que é o Python e como ele Funciona . . . . .	20

4.2	Códigos em Python . . . . .	24
4.2.1	Códigos da Simulação . . . . .	24
4.2.2	Códigos da Eleição . . . . .	25
4.3	Variáveis Aleatórias Contínuas . . . . .	31
<b>5</b>	<b>Recursos Impostos durante a Eleição</b>	<b>33</b>
5.1	Primeiro Turno . . . . .	33
5.1.1	Recurso I - Vandrê Monteiro . . . . .	33
5.1.2	Recurso II - Karen Duque . . . . .	34
5.1.3	Recurso III - Vandrê Monteiro . . . . .	35
5.1.4	Recurso IV - Phydia . . . . .	36
<b>6</b>	<b>Tabelas Completas de Resultado</b>	<b>37</b>
6.1	Lista Geral . . . . .	37
6.2	Lista Brasília . . . . .	37
<b>7</b>	<b>Referências Bibliográficas</b>	<b>37</b>

# 1 Introdução - Descrição do Processo e Escolha Social

Primeiramente, é preciso dizer que esse documento foi elaborado pela CEC ( *Comissão Eleitoral Calanguística*) com o objetivo de documentar o processo eleitoral para a escolha de 3 novas músicas pra o repertório do Bloco do Calango Careta em 2024.

Desta forma, por meio das próximas subseções, iremos detalhar e explicar como se deu o processo em cada etapa do pleito, desde a escolha das músicas até os resultados brutos e quais critérios utilizamos para eleger as músicas para o 2° Turno.

## 1.1 Início do Processo

Durante muitas conversas entre os membros do bloco, além de debates dentro do grupo de *Whatsapp* foi detectado que existia uma demanda por músicas que representassem Brasília dentro do nosso repertório, não só a cidade em si, mas serem músicas compostas por artistas locais ou que possuem uma história com a cidade.

Dessa forma, foi estabelecido que neste processo eleitoral seriam consideradas duas listas distintas, uma para escolher uma música de forma "Geral" considerando qualquer artista e qualquer música e uma exclusiva para Brasília.

Para a montagem dessas duas listas, o coletivo foi consultado e solicitado que enviassem para a Larissa suas sugestões, cada membro devia dar duas sugestões de músicas sendo elas aceitas nas seguintes possibilidades:

- 1 música para a lista geral e uma para a lista de Brasília
- 1 música para as duas listas

Após o envio das solicitações, foram montadas duas *playlists* na plataforma Spotify, considerando as devidas solicitações e para quais listas elas iriam. Assim foram montadas as seguintes listas

Porém, durante o processo eleitoral, foram aplicados alguns recursos que resultaram em mudanças na formação das listas, dessa forma as listas atualizadas se encontram em:

Para conferir os recursos, ver 5

Vale ressaltar também que, internamente, a CEC decidiu por colocar as seguintes regras para a eleição:

## Primeiro Turno

- Cada votante terá direito à 3 votos em cada lista, não sendo permitida a repetição
- Se uma mesma música for ganhadora nas duas listas, valerá a seguinte regra como critério de desempate: Seja  $x$  uma música presente em duas listas distintas, e  $NV_{x,y}^i$  a contagem de votos para a música  $x$  ou  $y$  na lista  $i$ , considerando  $y$  como qualquer outra música. Então, a escolha final da música  $x$  é determinada por:

$$\text{Música escolhida} = \begin{cases} x & \text{se } NV_x^i > NV_y^i \quad \forall \quad x, y \in i, \forall \quad i \in \{1, 2\} \\ y & \text{se } y \neq x \quad \text{com } NV_x^i < NV_x^{i^*} \quad \text{para a lista } i^*, \quad i^* \neq i \end{cases} \quad (1)$$

Música	Artista
Start Wearing Purple	Gogol Bordello
Nitro Bong (Radio Mix)	Webra
Jogo De Calçada	Os Mutantes
Olhos Coloridos	Sandra De Sá
Cumbia da Praia	Academia da Berlinda
Libertango	Yo-Yo Ma
Last Nite	The Strokes
Sandra Rosa Madalena "A Cigana"	Sidney Magal
Magical Mystery Tour	The Beatles
Kilario	Di Melo
Caçada	Chico Buarque
Funkytown / I Feel Good	Lucky Chops
Frevo Sanfonado	Spok Frevo Orquestra
Alma Sebosa	Johnny Hooker
Quem Sabe	Los Hermanos
Bumbum de Ouro	Gloria Groove
COISA BOA - AO VIVO	Gloria Groove
Rapunzel	Daniela Mercury
Que Beleza	Tim Maia
Tumbalacatumba	Galinha Pintadinha
Smells Like Teen Spirit	Nirvana
Ob-La-Di, Ob-La-Da	The Beatles
Chevette Azul	Thiago França, A Espetacular Cha- ranga do França
Bombinha	Juliana Linhares
Sexy Yemanjá	Pepeu Gomes
Iko Iko	The Gramophone Allstars
Abre Alas	Ivan Lins
Confidências	Petrúcio Amorim, Jorge De Altinho

Tabela 1: Lista Geral, fonte: Spotify

Música	Artista
Pra Enfeitar	Saci Wèrè
Tempestade	Maskavo Roots
Meu Erro	Os Paralamas Do Sucesso
Juriti	Liga Tripa
O Carcará E A Rosa	Natiruts
Suite Brasília: Theme 1	Renato Vasconcellos
Tambores do Paranoá	Martinha do Coco
Seria o rolex?	Móveis Coloniais de Acaju
Óculos	Os Paralamas Do Sucesso
Sol no Rosto	Remobília, André Gonzales, Beto Mejía, Esdras Nogueira, Rodrigo Balduino, Gustavo Dreher, fernando jato, Mateo Piracés-Ugarte, Thiago Cunha
Presente De Um Beija-Flor	Natiruts
Esquilo não samba	Móveis Coloniais de Acaju
O Beco	Os Paralamas Do Sucesso
Perfume Dela	Chinelo de Couro, Martinha do Coco
Ícaro	Joe Silhueta
Pretin (Verão, Pt. 1)	Flora Matos, Stereodubs
Travessia Do Eixão	Legião Urbana
Coco do Cerrado	Martinha do Coco, Escola Buriti Vermelho
Perfume Dela	Martinha do Coco
Piloto	NATTAN
Baião do Cerrado	Kirá, Kirá e a Ribanceira, Jéssica Carvalho
Xote dos Sonhos	Pé de Cerrado, Índios Tingui-Botó

Tabela 2: Lista de Brasília, fonte: Spotify

Música	Artista
Start Wearing Purple	Gogol Bordello
Nitro Bong (Radio Mix)	Webra
Jogo De Calçada	Os Mutantes
Olhos Coloridos	Sandra De Sá
Cumbia da Praia	Academia da Berlinda
Libertango	Yo-Yo Ma
Last Nite	The Strokes
Sandra Rosa Madalena “A Cigana”	Sidney Magal
Magical Mystery Tour - Remastered 2009	The Beatles
Kilario	Di Melo
Caçada	Chico Buarque
Funkytown / I Feel Good	Lucky Chops
Alma Sebosa	Johnny Hooker
Quem Sabe	Los Hermanos
Bumbum de Ouro	Gloria Groove
COISA BOA - AO VIVO	Gloria Groove
Rapunzel	Daniela Mercury
Que Beleza	Tim Maia
Smells Like Teen Spirit	Nirvana
Ob-La-Di, Ob-La-Da - Remastered 2009	The Beatles
Chevette Azul	Thiago França, A Espetacular Cha- ranga do França
Bombinha	Juliana Linhares
Sexy Yemanjá	Pepeu Gomes
Iko Iko	The Gramophone Allstars
Abre Alas	Ivan Lins
Confidências	Petrúcio Amorim, Jorge De Altinho
Emmenez-moi	Charles Aznavour
Isquenta Muié	Orquestra Terremoto
Deixe-se Acreditar	Chinaina, Felipe S
Lindo Lago Do Amor	Gonzaguinha
Perdido Em Abbey Road	14 Bis
Proud Mary	Ike e Tina Turner

Tabela 3: Lista Geral após Recursos, fonte: Spotify

Música	Artista
Pra Enfeitar	Saci Wèrè
Tempestade	Maskavo Roots
Meu Erro	Os Paralamas Do Sucesso
Juriti	Liga Tripa
O Carcará E A Rosa	Natiruts
Suite Brasília: Theme 1	Renato Vasconcellos
Tambores do Paranoá	Martinha do Coco
Seria o rolex?	Móveis Coloniais de Acaju
Óculos	Os Paralamas Do Sucesso
Sol no Rosto	Remobília, André Gonzales, Beto Me- jía, Esdras Nogueira, Rodrigo Bal- duino, Gustavo Dreher, fernando ja- toba, Mateo Piracés-Ugarte, Thiago Cunha
Presente De Um Beija-Flor	Natiruts
Esquilo não samba	Móveis Coloniais de Acaju
O Beco	Os Paralamas Do Sucesso
Oração	Pé de Cerrado
Toda em Si	Ana Reis
Sorri, sou Rei	Natiruts e Claudia Leite
Natiruts Reggea Power	Natiruts
Ícaro	Joe Silhueta
Pretin (Verão, Pt. 1)	Flora Matos, Stereodubs
Travessia Do Eixão	Legião Urbana
Coco do Cerrado	Martinha do Coco, Escola Buriti Ver- melho
Perfume Dela	Martinha do Coco
Piloto	NATTAN
Baião do Cerrado	Kirá, Kirá e a Ribanceira, Jéssica Car- valho
Xote dos Sonhos	Pé de Cerrado, Índios Tingui-Botó
Deixa o Menino Jogar	Natiruts
Veneno de Cobra	Emília Monteiro e Dona Onete
Esperança no Vermelho	A Tuba Atômica do Planalto

Tabela 4: Lista de Brasília após Recursos, fonte: Spotify

## 1.2 Segundo Turno

- Será composto pelas músicas mais votadas de cada lista. O número de músicas que vai para o 2º turno depende da distribuição dos votos no 1º turno e será feito por meio de um método estatístico adequado e cuidadosamente selecionado pela CEC (A ser especificado na seção de Metodologia)

## 1.3 Escolha do Público

- As músicas classificadas em 2º e 3º lugar de cada lista formarão uma lista de 4 músicas que irão para votação popular.

# 2 Metodologia

Para a escolha das músicas ser a mais precisa possível, foi determinado como metodologia de escolha o uso dos conceitos de Medidas de Dispersão e de Posição para a distribuição dos votos na referida eleição.

## 2.1 Introdução - Variável, Distribuição e Medidas

### 2.1.1 Variável

Para o pleito, iremos utilizar como variável os votos que cada música recebeu durante os 2 turnos da eleição, de acordo com (BUSSAB; MORETTIN, 2010) para cada elemento investigado em uma pesquisa, tem-se um, ou mais de um, resultado correspondendo à realização de uma característica, a esse resultado, atribuímos o nome de Variável.

Como por exemplo, estamos estudando a distribuição de alturas para duas amostras, homens e mulheres, temos aqui nesse exemplo 3 variáveis, uma discreta e quantitativa (Altura) e duas variáveis categóricas (Homem ou Mulher).

Define-se variável discreta como aquela cujos os possíveis valores formam um conjunto finito de resultados possíveis, assim podemos definir variável categórica ou dicotômica como aquela que só aceita como resposta dois tipos distintos de classificação, sucesso ou fracasso, Homem ou mulher e por ai vai. No caso da eleição, estaremos lidando somente com variáveis discretas, ou seja, o quantitativo de votos representa um conjunto finito de resultados.



### 2.1.2 Distribuições de Frequência

Quando estudamos variáveis dentro das suas mais diversas classificações, o principal objetivo do pesquisador com esse estudo é compreender o comportamento dessas variáveis dentro da sua amostra, ou população. Analisando a maior quantidade de ocorrência das mais diversas características dentro de suas possíveis realizações, a essa quantidade de ocorrência damos o nome de Frequência e ao conjunto de frequências damos o nome de Distribuição, uma visão global desses acontecimentos.

### 2.1.3 Medidas de Posição e Medidas de Dispersão

Quando estudamos variáveis e observamos o seu comportamento por meio das distribuições de frequência. Em muitas situações, queremos resumir os dados dispostos, apresentando um ou mais valores que sejam representativos da distribuição como um todo. Podemos classificar esses valores como sendo:

1. Medidas de Posição: São valores que determinam a tendência de concentração dos dados observados. Dentro desse grupo temos: Moda, Média e Mediana
2. Medidas de Dispersão: São valores que mostram o quanto uma certa observação se distanciou da média dos dados observados. Dentro desse grupo temos: A variância e o desvio padrão

#### Medidas de Posição - Moda

A moda se refere de forma empírica como a observação que mais aparece dentro da nossa distribuição.

#### Medidas de Posição - Mediana

A Mediana ( $Md$ ) se refere à observação, ou valor, que ocupa a posição central da nossa distribuição, a Mediana ( $Md$ ) de um conjunto de observações  $x$  com tamanho  $n$  pode ser definida como:

$$Md(x) = \begin{cases} x_{\frac{n+1}{2}} & \text{se } n \text{ é ímpar} \\ \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} & \text{se } n \text{ é par} \end{cases} \quad (2)$$

## Medidas de Posição - Média

A Média, medida de posição mais usual, representa a concentração dos dados observados em um ponto bem definido, em uma expressão mais formal, definimos a média como a soma ponderada dos valores observados. A forma mais usual de compreender esta medida é a sua forma aritmética, onde os pesos da soma é 1. Assim, iremos definir a média como:

$$\bar{x} = \mu = \sum_{i=1}^N \frac{x_i}{N} \quad (3)$$

Em que  $N$  é o tamanho da nossa amostra.

## Medidas de Dispersão - Variância e Desvio Padrão

Definimos a variância e o desvio padrão como sendo indicadores baseados na média do quadrado dos desvios, considerando desvio como a diferença entre a  $i$ -ésima observação  $x_i$  da média amostral da nossa distribuição, sendo assim:

$$\Delta X = x_i - \mu \quad (4)$$

Dessa forma podemos definir a variância ( $\sigma^2$ ) como:

$$\sigma^2 = \sum_{i=1}^N \frac{(x_i - \mu)^2}{N} \quad (5)$$

Já o desvio padrão ( $\sigma$ ) é a raiz quadrada da variância, servindo para resolver uma problemática da medida anterior, ele coloca os dados em sua dimensão original. Dessa forma, temos:

$$\sigma = \sqrt{\sum_{i=1}^N \frac{(x_i - \mu)^2}{N}} \quad (6)$$

## Percentis - Quantis Empíricos

Seguindo as ideias de (BUSSAB; MORETTIN, 2010), tanto a média quanto o desvio padrão podem não ser medidas interessantes para representar um conjunto de dados, por conta dos seguintes fatores:

- São medidas afetadas por valores extremos de forma exagerada

- Apenas com esses dois valores não temos ideia se a nossa distribuição é assimétrica ou simétrica.

Devido a isso, podemos considerar uma nova medida, a qual chamaremos de quantil de ordem  $p$  ou  $p$ -quantil, dada por  $q(p)$ , onde  $p$  é uma proporção qualquer,  $0 < p < 1$  tal que  $100p\%$  das observações da amostra sejam menores que  $q(p)$ .

Imagine dividir seus dados em cem partes iguais. Cada parte, chamada de percentil, representa uma proporção da amostra. O  $p$ -quantil, por sua vez, é o valor que divide os dados em duas partes:

1.  $p\%$  das observações são **menores** que o  $p$ -quantil.
2.  $(100 - p)\%$  das observações são **maiores** que o  $p$ -quantil.

Ao analisar os quantis, você obtém uma visão mais completa da distribuição dos seus dados:

- **Identificando assimetrias:** Se os quantis médios (como o Q2, ou mediana) estiverem deslocados para um dos extremos, a distribuição é assimétrica.
- **Desvendando outliers:** Os quantis extremos (como Q1 e Q99) podem indicar a presença de **valores atípicos** que influenciam a média.
- **Comparando distribuições:** Os quantis facilitam a comparação entre conjuntos de dados, mesmo com escalas diferentes.

## 2.2 Critério de Contagem dos Votos e Escolha das Músicas Vencedoras

Como critério de contagem dos votos, a CEC decidiu que para serem definidas as músicas vencedoras será utilizado o seguinte critério no **Primeiro Turno**:

- As músicas a serem candidatas a vencedoras do 1o turno e vencedoras gerais deverão estar acima da média aritmética ( $\mu$ ) da distribuição de votos de sua referida lista.
- Para serem ordenadas, serão escolhidas aquelas que mais se distanciaram da média ( $\mu$ ), sendo assim, usaremos o Percentil de 75% para conseguirmos extrair as músicas mais votadas da distribuição, isto é, as músicas que estão acima do percentil

## 2.3 Simulação da Eleição

Para facilitar a compreensão dos critérios adotados para classificação das músicas, realizamos uma simulação usando Python, ver 4.1 em 4, ferramenta escolhida para apurar os votos, e explicaremos passo a passo nesta seção.

Para realizar todas as operações necessárias usamos as seguintes bibliotecas da linguagem:

1. Numpy (Numerical Python) (HARRIS et al., 2020) para operações envolvendo geração de números aleatórios e cálculo de estatísticas descritivas (Média, Percentil)
2. Pandas (TEAM, 2020) e (MCKINNEY, 2010) para operarmos com dados tabulares, isso envolve a construção de Tabelas, também chamadas de *Dataframes*, para armazenar os dados em Linhas e Colunas
3. Scipy (VIRTANEN et al., 2020) para identificarmos a distribuição dos nossos dados.
4. Matplotlib (HUNTER, 2007) gerar gráficos envolvendo nossas variáveis.

Com isso, iremos agora descrever e analisar passo a passo o que foi feito na simulação, considerando que os métodos aplicados nela, serão os métodos aplicados na eleição real. Todo esse passo a passo pode ser conferido em 4.2, Assim:

### 2.3.1 Primeiro Passo:

Como primeiro passo para a simulação, iremos gerar variáveis aleatórias correspondentes aos votos, para realizar essa tarefa, montamos uma lista fictícia com 31 músicas (quantidade correspondente à lista 1, por exemplo), a partir dessa lista criamos uma variável correspondente ao número de votantes nessa eleição, ou seja, o total atual de membros do calango.

Após isso, criamos uma matriz (array) para armazenar os votos de cada música, com todo esse aparato montado, criamos um loop for para passar de voto a voto dentro do limite estabelecido pelo número de votantes, dentro desse loop existe a variável *voto\_pessoa* que corresponde aos 3 votos de cada votante, após isso temos o loop que adiciona o voto para cada uma das músicas escolhidas pela pessoa.

Por fim, criamos um dataframe para armazenar esses dados, obtendo como resultado da simulação o seguinte perfil:

Música	Votos
Música 1	2
Música 2	7
Música 3	3
Música 4	4
Música 5	9
Música 6	8
Música 7	6
Música 8	9
Música 9	5
Música 10	6
Música 11	3
Música 12	11
Música 13	8
Música 14	12
Música 15	10
Música 16	9
Música 17	3
Música 18	7
Música 19	9
Música 20	11
Música 21	6
Música 22	5
Música 23	11
Música 24	5
Música 25	6
Música 26	10
Música 27	5
Música 28	6
Música 29	5
Música 30	8
Música 31	7

### 2.3.2 Segundo Passo:

Certo, com a nossa distribuição de votos definida, plotamos o gráfico do histograma dos votos para compreender como é o seu comportamento, o gráfico pode ser visto na figura 1

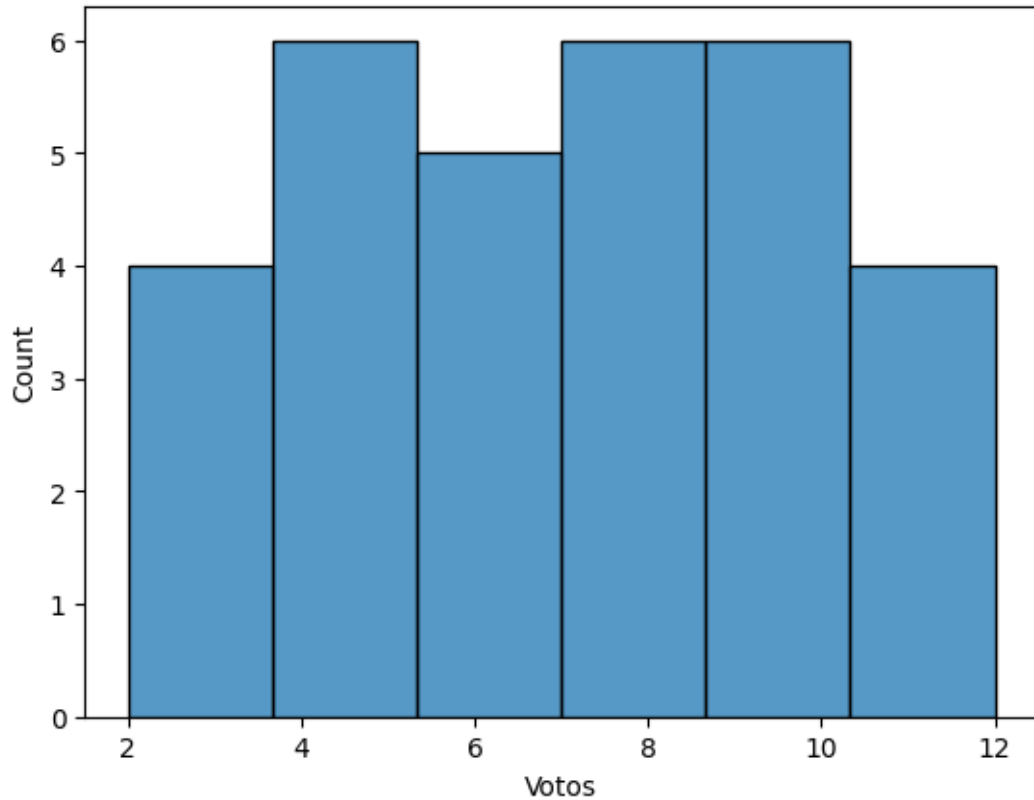


Figura 1: Histograma dos Votos

Dessa forma, podemos deduzir que a nossa distribuição dos votos é aproximadamente uma Normal, tal distribuição pode ser melhor compreendida na seção 4.3 .

Dessa forma, calculamos as estatísticas descritivas dessa amostra de votos simulada, obtendo o seguinte resultado:

	Votos
Total	31.000000
Média de Votos ( $\mu$ )	6.967742
Desvio Padrão ( $\sigma$ )	2.689286
Mínimo	2.000000
Percentil 25%	5.000000
Percentil 50%	7.000000
Percentil 75%	9.000000
Máximo	12.000000

### 2.3.3 Terceiro Passo:

Após os resultados das estatísticas descritivas, iremos classificar as nossas músicas com base nos resultados da Média e do Percentil.

Para isso, iremos inserir na nossa tabela duas colunas, a coluna "Critério Média" que será composta por duas possíveis respostas: Aprovada e Reprovada, o critério aqui é verificar se a quantidade de votos recebida pela música é inferior ou esta dentro/acima da média, sendo reprovada se estiver com votos inferiores a essa quantia e aprovada caso contrário.

Já a coluna percentil, localiza as músicas que estão com quantidade de votos acima do percentil 75% da nossa distribuição, recebendo "True (Verdadeiro)" caso for maior e "False (Falso)" caso contrário. São selecionadas em primeiro turno as músicas que tiverem sido aprovadas pelo critério da média e estão acima do percentil 75%. Dessa forma, chegamos no seguinte resultado com todas as músicas da amostra, como mostra a tabela 10. Posteriormente, filtramos e deixamos só as que seguem os critérios adotados, representada pela tabela 6, tais músicas estão aprovadas no 1º turno e classificadas para o 2º turno:

## 3 Análise da Votação e Resultado

Para contar os votos durante o 1º turno da eleição, os códigos presentes em 4.2.2 foram utilizados. Dessa forma, foram contados os votos um a um em cada linha da base de dados, considerando que em uma linha temos 3 colunas para votos, afinal uma mesma pessoa escolheu 3 músicas distintas nesse turno, sendo que o mesmo código foi usado para as

Música	Votos	Critério Média	Percentil
Música 1	2	Reprovada	False
Música 2	7	Aprovada	False
Música 3	3	Reprovada	False
Música 4	4	Reprovada	False
Música 5	9	Aprovada	False
Música 6	8	Aprovada	False
Música 7	6	Reprovada	False
Música 8	9	Aprovada	False
Música 9	5	Reprovada	False
Música 10	6	Reprovada	False
Música 11	3	Reprovada	False
Música 12	11	Aprovada	True
Música 13	8	Aprovada	False
Música 14	12	Aprovada	True
Música 15	10	Aprovada	True
Música 16	9	Aprovada	False
Música 17	3	Reprovada	False
Música 18	7	Aprovada	False
Música 19	9	Aprovada	False
Música 20	11	Aprovada	True
Música 21	6	Reprovada	False
Música 22	5	Reprovada	False
Música 23	11	Aprovada	True
Música 24	5	Reprovada	False
Música 25	6	Reprovada	False
Música 26	10	Aprovada	True
Música 27	5	Reprovada	False
Música 28	6	Reprovada	False
Música 29	5	Reprovada	False
Música 30	8	Aprovada	False
Música 31	7	Aprovada	False

Tabela 5: Tabela dos Resultados Totais

Música	Votos	Critério Média	Percentil
Música 12	11	Aprovada	True
Música 14	12	Aprovada	True
Música 15	10	Aprovada	True
Música 20	11	Aprovada	True
Música 23	11	Aprovada	True
Música 26	10	Aprovada	True

Tabela 6: Tabela das Aprovadas



duas listas distintas de músicas, só era necessário alterar as funções de filtro.

Com base nas respostas do forms enviado para o bloco, o perfil dos eleitores por naipe pode ser descrito usando o seguinte gráfico de barras, que demonstra que o naipe mais engajado no processo foi o de Flautas, seguido pela Trupe e pelo naipe de Saxofones e Clarinetes. As tabelas completas de votação estarão disponíveis em 4 na seção 6

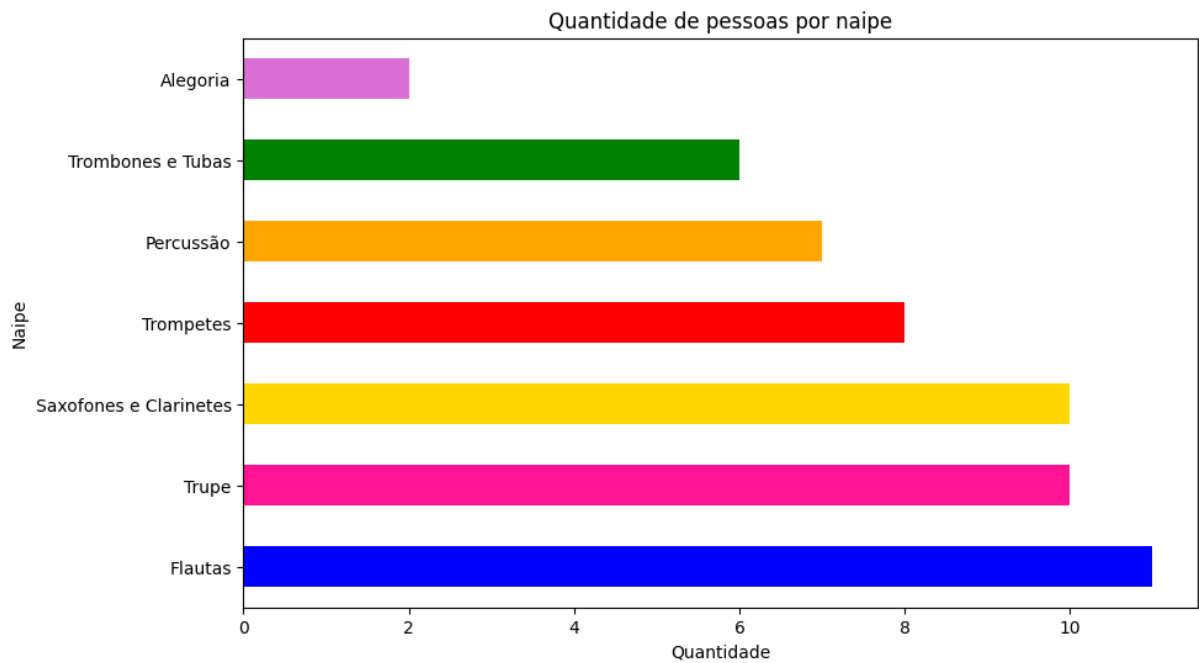


Figura 2: Perfil dos Eleitores por Naipe

### 3.1 Resultados Lista Geral

Primeiramente, foi realizada a contagem dos votos para músicas presentes nas listas expostas anteriormente, com isso extraímos as estatísticas descritivas da distribuição de votos, obtendo o seguinte resultado:

	Votos
Total de Votos Computados	162
Contagem Única	29.000
Média de Votos ( $\mu$ )	5.586207
Desvio Padrão ( $\sigma$ )	5.321904
Mínimo	1
Percentil 25%	2.000000
Percentil 50%	3.000000
Percentil 75%	8.000000
Máximo	24

Com isso, temos que a média ( $\mu$ ) da lista geral foi de 5,586207 e o valor correspondente ao Percentil de 75% foi 8,000000. A partir desses dados fizemos o filtro para declarar as músicas vencedoras no primeiro turno, obtendo por fim:

Música	Votos	Critério Média	Percentil
Kilario	24	Aprovada	True
Libertango	13	Aprovada	True
Ob-La-Di,Ob-La-Da	12	Aprovada	True
Chevette Azul	12	Aprovada	True
Bumbum de Ouro	11	Aprovada	True
Olhos Coloridos	10	Aprovada	True
Funkytown/ I Feel Good	10	Aprovada	True

Tabela 7: Tabela dos Resultados 1o Turno - Lista Geral

Analisando temporalmente o desempenho das músicas, podemos ver um destaque maior para Kilario dado o aumento de votos repetino no decorrer dos dias da eleição, o gráfico 3 mostra essa relação.

### 3.2 Resultados - Lista Brasília

Seguindo o mesmo processo da seção anterior, ou seja, extraímos as estatísticas descritivas da distribuição de votos, obtendo o seguinte resultado:

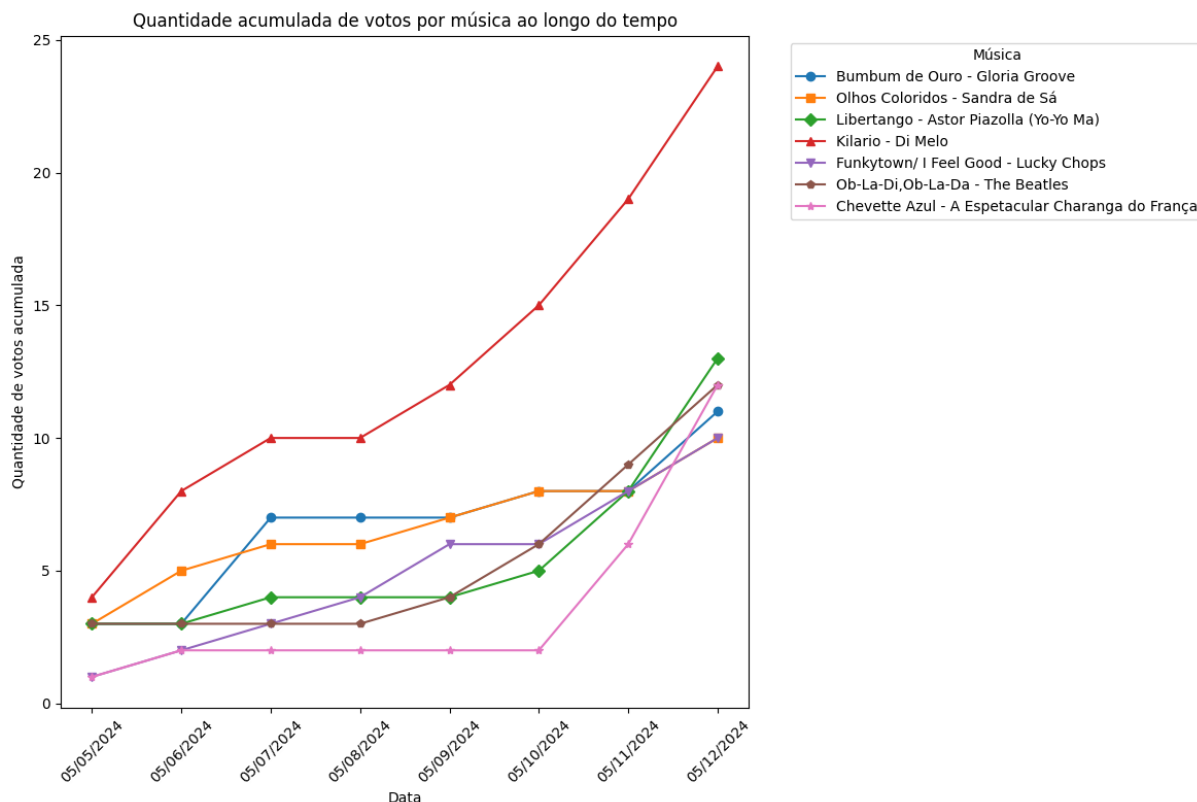


Figura 3: Desempenho das Músicas Vencedoras do 1o turno em relação ao Tempo - Lista Geral

	Votos
Total de Votos Computados	162
Contagem Única	26.000
Média de Votos ( $\mu$ )	6.230769
Desvio Padrão ( $\sigma$ )	4.510501
Mínimo	1
Percentil 25%	3.000000
Percentil 50%	5.500000
Percentil 75%	9.000000
Máximo	16

Com isso, temos que a média ( $\mu$ ) da lista de Brasília foi de 6,230769 e o valor correspondente ao Percentil de 75% foi 9,000000. A partir desses dados fizemos o filtro para declarar as músicas vencedoras no primeiro turno, obtendo por fim:

Verificando o desempenho temporal das músicas, podemos ver um crescimento de Pretin e Seria o Rolex, resultando em um empate.

Música	Votos	Critério Média	Percentil
Seria o rolex ?	16	Aprovada	True
Pretin	16	Aprovada	True
Pra Enfeitar	12	Aprovada	True
O Beco	11	Aprovada	True
Piloto	11	Aprovada	True
Natiruts	10	Aprovada	True
Reggea Power			

Tabela 8: Tabela dos Resultados 1o Turno - Lista Brasília

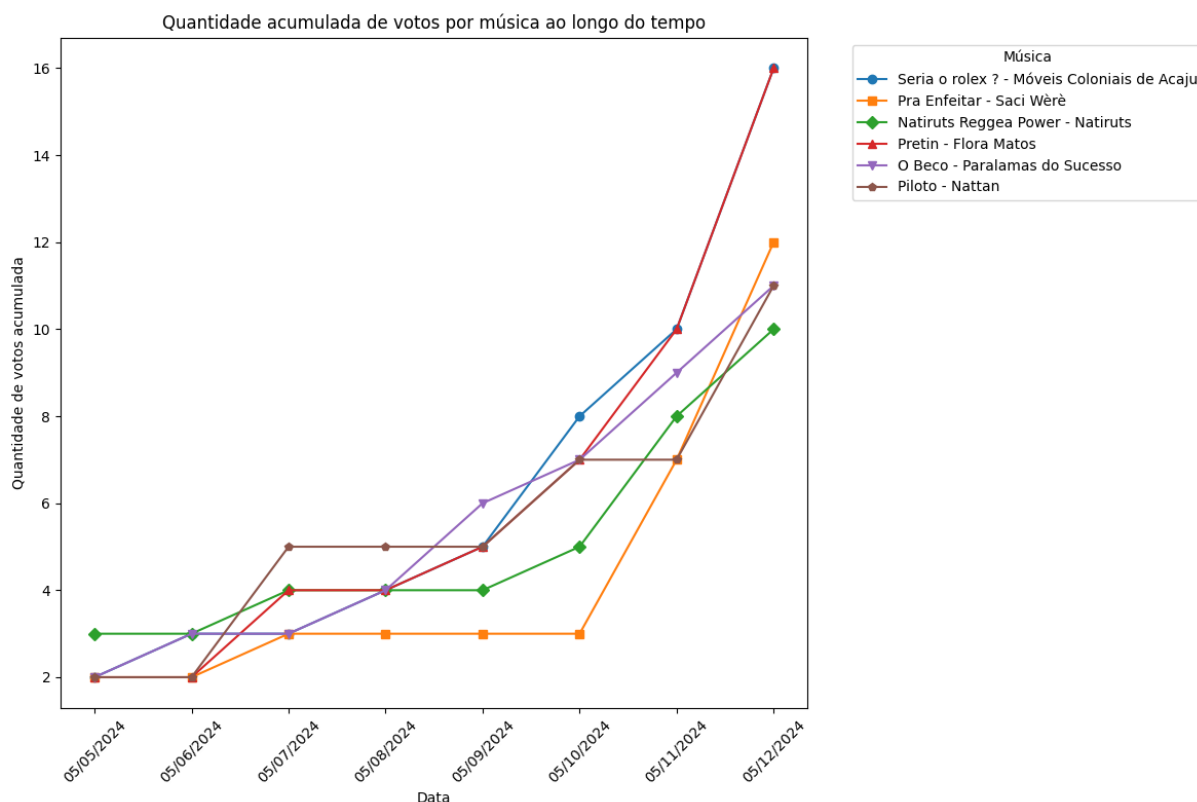


Figura 4: Desempenho das Músicas Vencedoras do 1o turno em relação ao Tempo - Lista Brasília

## 4 Apêndice

### 4.1 O que é o Python e como ele Funciona

*Python* é uma linguagem de programação interpretada de software livre, muito utilizada e reconhecida em diversos tipos de ambientes, sejam em ambientes corporativos (desenvolvimento de *games* e *software*) ou em ambientes científicos para simulações numéricas, análise de dados e muitas outras aplicações.

A linguagem foi desenvolvida por Guido Van Rossumem, recebendo seu nome como uma homenagem ao grupo inglês *Monty Python*, com a finalidade de ser uma linguagem

simples e de fácil compreensão. Apesar de simples, é uma linguagem muito poderosa, que pode ser usada para desenvolver e administrar grandes sistemas.

Sendo considerada uma linguagem de programação de alto nível, ou seja, a forma de escrevermos o código do nosso programa e a forma como nós, seres humanos, nos expressamos é próxima, não possuindo assim uma linguagem própria de máquina, como é o caso da linguagem de baixo nível assembly.

Uma das principais características que diferencia a linguagem Python das outras é a legibilidade dos programas, ou scripts, escritos. Isto ocorre porque, em outras linguagens, é muito comum o uso excessivo de marcações (ponto ou ponto e vírgula), de marcadores (chaves, colchetes ou parênteses) e de palavras especiais (begin/end), o que torna mais difícil a leitura e compreensão dos programas, tendo assim esse recurso reduzido, se tornando uma linguagem mais ampla e de fácil compreensão, como pode ser vista nas figuras 5 e 6

A screenshot of a Python 3.8.5 Shell window. The window has a title bar that says "Python 3.8.5 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window is a text editor showing the following text:

```
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello, World!')
Hello, World!
>>> |
```

The text is color-coded: "Type" is blue, "help", "copyright", "credits", and "license()" are in quotes and black, "for more information." is black, ">>>" is green, "print" is red, "Hello, World!" is black, and the cursor is a vertical bar. At the bottom right of the window, a status bar shows "Ln: 5 Col: 4".

Figura 5: Programa inicial do Python: Hello World

O conceito de "livre" não se refere apenas ao preço, mas sim à liberdade de acesso ao código-fonte e à capacidade de modificar o software de acordo com as necessidades do usuário. Isso dá aos usuários a liberdade de entender como o programa funciona e oferece a oportunidade de colaborar com a comunidade de desenvolvimento. A história do software livre remonta ao final da década de 1970 e está profundamente ligada ao

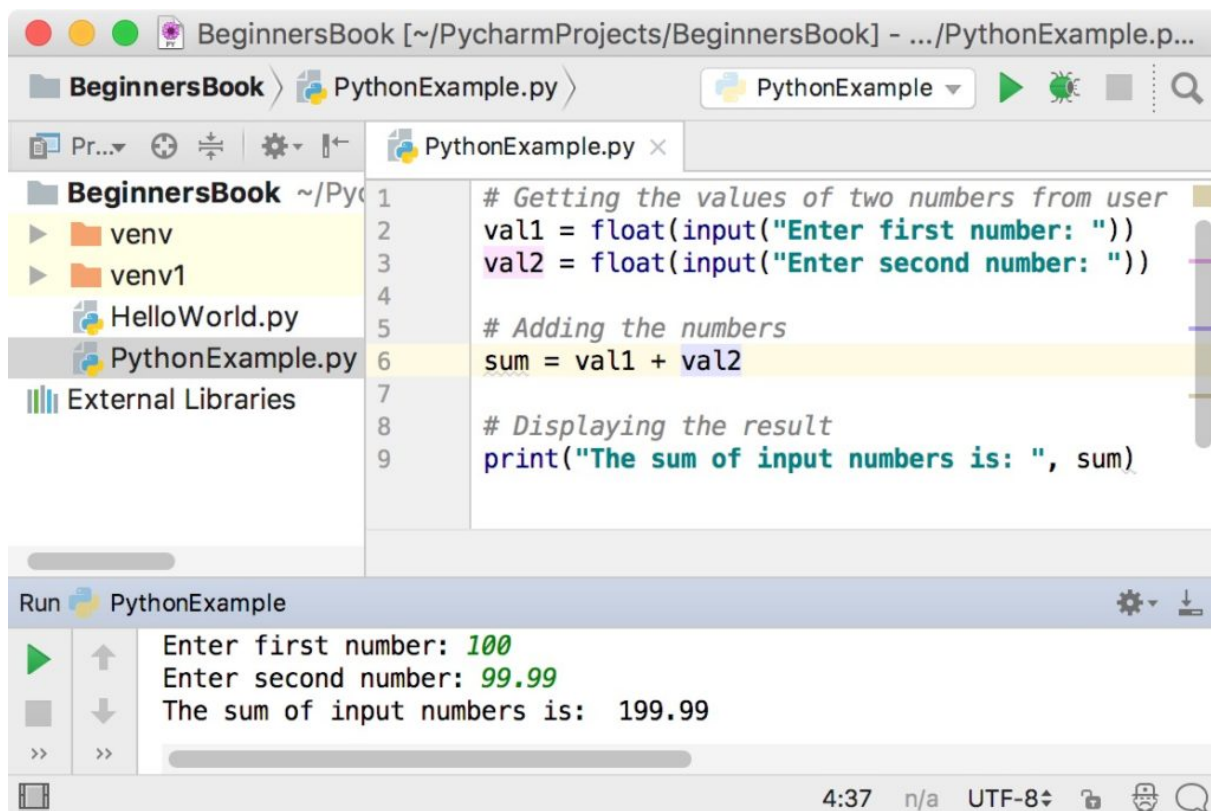


Figura 6: Soma de dois números em Python

movimento do software livre e ao desenvolvimento da cultura hacker. Aqui estão alguns marcos importantes:

- GNU Project: Em 1983, Richard Stallman lançou o Projeto GNU (GNU's Not Unix) com o objetivo de criar um sistema operacional completamente livre e de código aberto. O projeto visava oferecer um conjunto completo de ferramentas e programas de software para que os usuários tivessem total liberdade em suas atividades de computação.
- Licença GPL: Em 1989, Stallman criou a Licença Pública Geral GNU (GPL), uma licença de software que garantia as quatro liberdades essenciais dos softwares livres. A GPL permitiu que os desenvolvedores compartilhassem suas modificações e melhorias com a comunidade, garantindo que o software permanecesse livre, mesmo após ser modificado.
- Linux: Na década de 1990, o núcleo do sistema operacional GNU estava incompleto. Linus Torvalds, em 1991, criou o kernel Linux como parte do projeto GNU, combinando-o com o software GNU existente, formando um sistema operacional

funcional conhecido como Linux GNU.

- Open Source Initiative: Em 1998, o termo "código aberto" foi cunhado e a Open Source Initiative (OSI) foi fundada por Eric S. Raymond e Bruce Perens. Eles promoveram uma abordagem mais pragmática do software livre, enfatizando os benefícios práticos do código aberto para as empresas.
- Expansão do Movimento: O movimento do software livre e de código aberto (FOSS - Free and Open Source Software) continuou a crescer e atrair a atenção de empresas, organizações e governos em todo o mundo. Grandes projetos de software livre, como o Apache, MySQL, Mozilla Firefox e muitos outros, ganharam popularidade e ampla adoção.

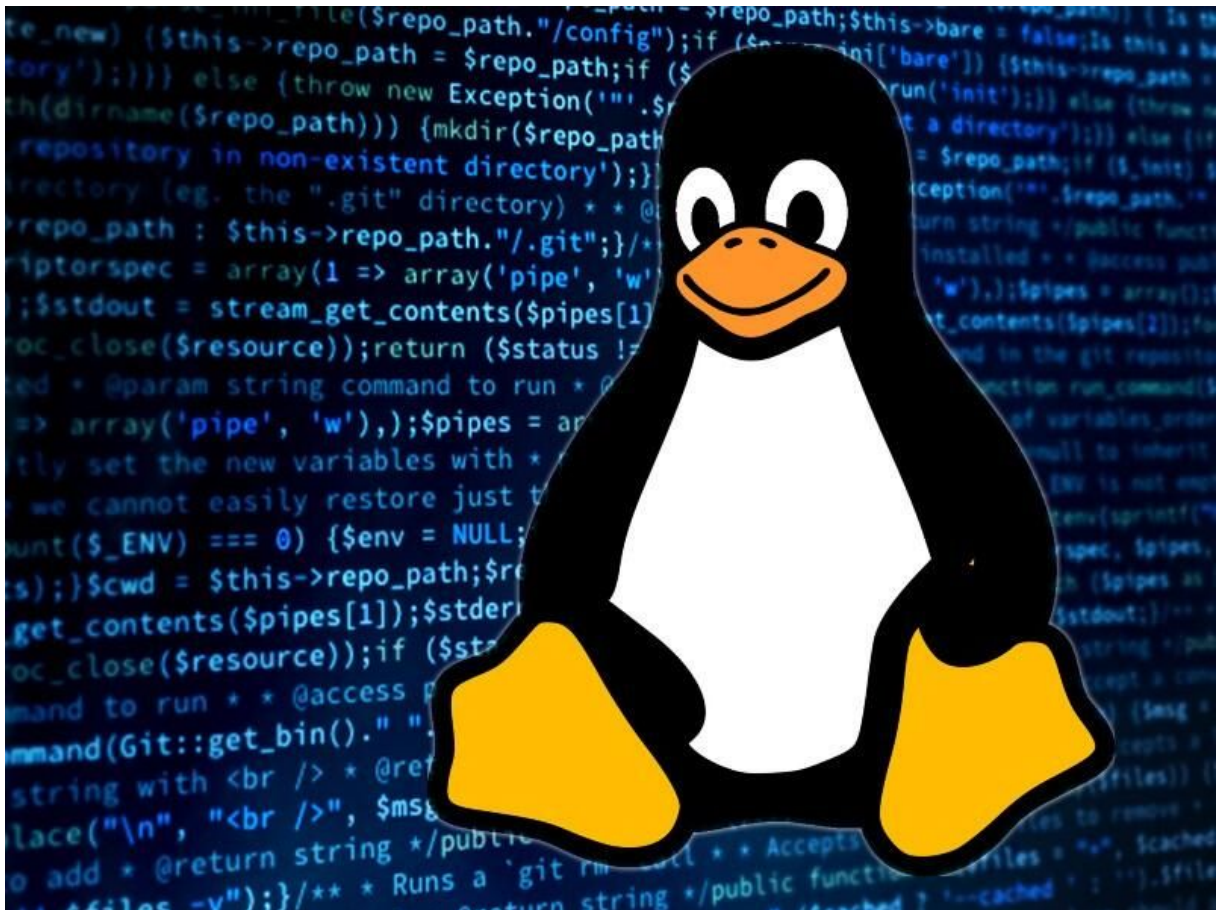


Figura 7: Mascote do sistema operacional Linux, o pinguim Tux.

O Python, assim como outras linguagens, possui uma gama de Bibliotecas para as mais diversas atividades, Bibliotecas são um conjunto de módulos e módulos, por sua vez, é um comando necessário para realizar uma tarefa ou operação.

## 4.2 Códigos em Python

### 4.2.1 Códigos da Simulação

```
1 #Importando/Chamando as bibliotecas
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import scipy.stats as st #Gerar distribuicoes
6 # Lista de 31 músicas
7 lista_musicas = [
8     "Música 1", "Música 2", "Música 3", "Música 4", "Música 5",
9     "Música 6", "Música 7", "Música 8", "Música 9", "Música 10",
10    "Música 11", "Música 12", "Música 13", "Música 14", "Música 15",
11    "Música 16", "Música 17", "Música 18", "Música 19", "Música 20",
12    "Música 21", "Música 22", "Música 23", "Música 24", "Música 25",
13    "Música 26", "Música 27", "Música 28", "Música 29", "Música 30",
14    "Música 31"
15 ]
16
17 # Número de votos
18 num_votos = 72
19
20 # Array para armazenar os votos de cada música
21 votos = {musica: 0 for musica in lista_musicas}
22
23 # Simulação dos votos das 72 pessoas
24 for voto in range(num_votos):
25     # Escolher 3 músicas distintas para cada pessoa
26     votos_pessoa = np.random.choice(lista_musicas, 3, replace=False)
27     # Adicionar 1 voto para cada uma das músicas escolhidas pela pessoa
28     for musica in votos_pessoa:
29         votos[musica] += 1
30
31 # Exibir os resultados dos votos
32 for musica, num_votos in votos.items():
33     print(f"{musica}: {num_votos} votos")
34
35 # Criar DataFrame a partir do dicionário de votos
```



```
36 df = pd.DataFrame(list(votos.items()), columns=['Música', 'Votos'])
```

Listing 1: Gerando a variável aleatória para o Voto

```
1 # Gerando gráfico
2 sns.histplot(x = "Votos", data = df)
3 #Estatísticas Descritivas
4 df_est_descritiva = df.describe()
```

Listing 2: Gerando o Gráfico e Calculando as Estatísticas Descritivas

```
1 # Critérios de Classificação
2 media = np.mean(df['Votos']) #Cálculo da Média
3 df['Critério Média'] = np.where(df['Votos'] <=
4                                 media, 'Reprovada', 'Aprovada')
5 #Percentil
6 percentil_75 = np.quantile(df['Votos'], 0.75)
7 df['Percentil'] = np.where(df['Votos'] > percentil_75, True, False)
```

Listing 3: Critérios de Classificação

```
1 #Filtrando
2 df_1_turno = df[
3     (df['Critério Média'] == 'Aprovada') & (df['Percentil'] == True)]
4 #Ordenando dados do mais votado para o menos votado
5 df_1_turno.sort_values(by = 'Votos', ascending = False)
```

Listing 4: Filtrando para o Resultado Final

#### 4.2.2 Códigos da Eleição

```
1 #Importando as bibliotecas
2 import pandas as pd
3 import numpy as np
4 import seaborn as sns
5 import matplotlib.pyplot as plt
```

Listing 5: Importando das bibliotecas necessárias

```
1 #Lendo o dataframe dos resultados
2 df = pd.read_csv('/content/Eleiç es Calango - 2024 (respostas) -
3                 Respostas ao formulário 1.csv')
```

Listing 6: Lendo a base

```

1 # Crie uma contagem dos valores únicos na coluna 'Qual é o seu naipe?'
2 contagem_naipe = df['Qual seu Naipe ?'].value_counts()
3
4 # Crie o gráfico de barras
5 # Lista de Cores
6 colors = ['blue', 'deeppink', 'gold', 'red', 'orange', 'green']
7 plt.figure(figsize=(10, 6))
8 contagem_naipe.plot(kind='barh', color= colors)
9
10 # Adicione rótulos e título
11 plt.xlabel('Quantidade')
12 plt.ylabel('Naipe')
13 plt.title('Quantidade de pessoas por naipe')
14
15 # Mostrar o gráfico
16 plt.show()

```

Listing 7: Montando Gráfico da quantidade de votantes por Naipe

```

1 #split de votos
2 def separar_votos(df):
3     # Inicializar as listas para cada coluna de votos
4     voto_lista1_geral = []
5     voto_lista2_geral = []
6     voto_lista3_geral = []
7
8     # Iterar sobre os votos na coluna do DataFrame
9     for voto in df['Quais são os seus votos para a Lista Geral ?\nVocê
10     tem 3 votos!']:
11         # Dividir os votos com base na vírgula
12         votos_separados = voto.split(", ")
13
14         # Adicionar cada música a coluna correspondente
15         for i, musica in enumerate(votos_separados):
16             if i == 0:
17                 voto_lista1_geral.append(musica)
18             elif i == 1:
19                 voto_lista2_geral.append(musica)
20             elif i == 2:
21                 voto_lista3_geral.append(musica)

```

```

21
22     return voto_lista1_geral, voto_lista2_geral, voto_lista3_geral
23 # Chamar a função para separar os votos
24 voto_lista1_geral, voto_lista2_geral, voto_lista3_geral = separar_votos(
    df)
25 #Armazenando votos em uma coluna para cada
26 df['Voto 1 - Lista Geral'] = voto_lista1_geral
27 df['Voto 2 - Lista Geral'] = voto_lista2_geral
28 df['Voto 3 - Lista Geral'] = voto_lista3_geral

```

Listing 8: Montando Split de Votos para determinar os 3 votos de cada pessoa

```

1 # Função para agregar votos por música
2 def contar_votos(voto_lista1_geral, voto_lista2_geral, voto_lista3_geral
    ):
3     # Criar um dicionário para armazenar os votos por música
4     votos_por_musica = {}
5
6     # Iterar sobre as listas de votos e contar os votos para cada música
7     for lista in [voto_lista1_geral, voto_lista2_geral,
    voto_lista3_geral]:
8         for musica in lista:
9             if musica in votos_por_musica:
10                 votos_por_musica[musica] += 1
11             else:
12                 votos_por_musica[musica] = 1
13
14     return votos_por_musica
15 # Agregar os votos por música
16 votos_por_musica = contar_votos(voto_lista1_geral, voto_lista2_geral,
    voto_lista3_geral)
17
18 # Criar um DataFrame a partir do dicionário de votos por música
19 df_votos_agregados = pd.DataFrame(list(votos_por_musica.items()),
    columns=['Música', 'Total de Votos'])
20
21 # Imprimir o DataFrame de votos agregados
22 print(df_votos_agregados)

```

Listing 9: Agregando Votos por Música

```

1 #Critério da média
2 media = np.mean(df_votos_agregados['Total de Votos']) #Cálculo da Média
3 df_votos_agregados['Critério Média'] = np.where(df_votos_agregados['
    Total de Votos'] <=
4
5         media, 'Reprovada', 'Aprovada')
6 #Percentil
7 percentil_75 = np.quantile(df_votos_agregados['Total de Votos'],0.75)
8 df_votos_agregados['Percentil'] = np.where(df_votos_agregados['Total de
    Votos'] > percentil_75, True, False)

```

Listing 10: Critérios de Classificação

```

1 #Filtrando para mostrar os vencedores
2 df_1_turno = df_votos_agregados[
3     (df_votos_agregados['Critério Média'] == 'Aprovada') & (
4         df_votos_agregados['Percentil'] == True)]
5 df_1_turno.sort_values(by = 'Total de Votos',ascending = False)

```

Listing 11: Cálculo Final

```

1 #Gerando Lista das Musicas vencedoras do primeiro turno
2 musicas_1_turno = list(df_1_turno['Música'])
3 # Filtrar o DataFrame original para incluir apenas as linhas com as mú
    sicas da lista
4 df_plot_musicas = df[df['Voto 1 - Lista Geral'].isin(musicas_1_turno) |
5     df['Voto 2 - Lista Geral'].isin(musicas_1_turno) |
6     df['Voto 3 - Lista Geral'].isin(musicas_1_turno)]

```

Listing 12: Criando Lista para as Vencedoras e preparando base para gráficos

```

1 # Definir marcadores para cada naipe
2 naipes = df_plot_musicas['Qual seu Naipe ?'].unique()
3 markers = ['o', 's', 'D', '^']
4 default_marker = 'o' # Marcador padrão para um naipe não reconhecido
5 naipe_markers = {naipe: marker for naipe, marker in zip(naipes, markers)
6     }
7
8 # Para cada música do primeiro turno
9 for musica in musicas_1_turno:
10     # Filtrar o DataFrame para incluir apenas as linhas com a música
    atual

```

```

10     df_musica = df_plot_musicas[df_plot_musicas[['Voto 1 - Lista Geral',
11         'Voto 2 - Lista Geral', 'Voto 3 - Lista Geral']].isin([musica]).any(
12         axis=1)]
13
14     # Converter a coluna 'Carimbo de data/hora' para o tipo datetime
15     df_musica['Carimbo de data/hora'] = pd.to_datetime(df_musica['
16         Carimbo de data/hora'])
17
18     # Agrupar as músicas por data/hora e naipes e contar o número de
19     vezes que aparecem
20     df_votos_por_dia_naipes = df_musica.groupby([df_musica['Carimbo de
21         data/hora'].dt.strftime('%d/%m/%Y'), 'Qual seu Naipes ?']).size().
22     unstack(fill_value=0)
23
24     # Calcular os totais acumulados de votos por data/hora e naipes
25     df_totais_acumulados_naipes = df_votos_por_dia_naipes.cumsum()
26
27     # Plotar os totais acumulados de votos por Carimbo de data/hora e
28     naipes
29     plt.figure(figsize=(12, 8))
30     for naipes in df_totais_acumulados_naipes.columns:
31         marker = naipes_markers.get(naipes, default_marker)
32         plt.plot(df_totais_acumulados_naipes.index,
33             df_totais_acumulados_naipes[naipes], label=naipes, marker=marker)
34
35     plt.title(f'Quantidade acumulada de votos por dia para a música "{
36         musica}"')
37     plt.xlabel('Dia/Mês')
38     plt.ylabel('Quantidade de votos acumulada')
39     plt.xticks(rotation=45)
40     plt.legend(title='Naipes', bbox_to_anchor=(1.05, 1), loc='upper left',
41         )
42     plt.tight_layout()
43     plt.show()

```

Listing 13: Gráfico de Evolução temporal das músicas por naipes

```

1 # Filtrar o DataFrame para incluir apenas as linhas com as músicas do
2 primeiro turno
3 df_filtrado = df_plot_musicas

```

```

3
4 # Converter a coluna 'Carimbo de data/hora' para o tipo datetime e
   extrair apenas o dia/mês/ano
5 df_filtrado['Carimbo de data/hora'] = pd.to_datetime(df_filtrado['
   Carimbo de data/hora']).dt.strftime('%d/%m/%Y')
6
7 # Inicializar um DataFrame para armazenar os totais acumulados de votos
   por data/hora para cada música
8 df_totais_acumulados = pd.DataFrame(index=df_filtrado['Carimbo de data/
   hora'].unique(), columns=musicas_1_turno).fillna(0)
9
10 # Calcular os totais acumulados de votos por data/hora para cada música
11 for musica in musicas_1_turno:
12     df_musica = df_filtrado[(df_filtrado['Voto 1 - Lista Geral'] ==
13         musica) |
14         (df_filtrado['Voto 2 - Lista Geral'] ==
15             musica) |
16         (df_filtrado['Voto 3 - Lista Geral'] ==
17             musica)]
18     df_votos_por_dia = df_musica.groupby('Carimbo de data/hora').size()
19     df_totais_acumulados[musica] = df_votos_por_dia.cumsum()
20
21 # Preencher valores ausentes com a quantidade acumulada do dia anterior
   apenas se o valor do dia atual for zero
22 df_totais_acumulados = df_totais_acumulados.fillna(method='ffill')
23
24 # Definir uma lista de marcadores diferentes para cada música
25 marcadores = ['o', 's', 'D', '~', 'v', 'p', '*', 'X', 'h', 'd']
26
27 # Plotar os totais acumulados de votos por Carimbo de data/hora para
   cada música
28 plt.figure(figsize=(12, 8))
29 for i, musica in enumerate(musicas_1_turno):
30     plt.plot(df_totais_acumulados.index, df_totais_acumulados[musica],
31         label=musica, marker=marcadores[i])
32
33 plt.title('Quantidade acumulada de votos por música ao longo do tempo')
34 plt.xlabel('Data')
35 plt.ylabel('Quantidade de votos acumulada')

```

```

32 plt.xticks(rotation=45)
33 plt.legend(title='Música',bbox_to_anchor=(1.05, 1), loc='upper left')
34 plt.tight_layout()
35 plt.show()

```

Listing 14: Gráfico de Evolução temporal das músicas

### 4.3 Variáveis Aleatórias Contínuas

As variáveis aleatórias contínuas, como mostra (ROSS, 2009), recebem esse nome por conta de que são variáveis cujo o conjunto de valores possíveis é incontável. Dessa forma dizemos que  $\mathcal{X}$  é uma variável aleatória contínua se existir uma função não negativa  $f$ , definida para todo real  $x \in (-\infty, \infty)$ , que tenha a propriedade de que, para qualquer conjunto  $B$  de números reais:

$$P\{\mathcal{X} \in B\} = \int_B f(x)dx \quad (7)$$

Em que  $f(x)$  é definida como função densidade de probabilidade da variável aleatória  $x$ .

**Observação:** A equação acima é válida apenas para os conjuntos mensuráveis de  $B$ , que, para tal finalidade, incluem todos os conjuntos de interesse prático.

Traduzindo a equação para português, ela diz que a probabilidade de que  $\mathcal{X}$  esteja em  $B$  pode ser obtida integrando-se a função densidade de probabilidade ao longo do conjunto  $B$ .

Como  $\mathcal{X}$  deve assumir algum valor,  $f(x)$  deve satisfazer:

$$1 = P\{\mathcal{X} \in (-\infty, \infty)\} = \int_{-\infty}^{\infty} f(x)dx \quad (8)$$

Assim tudo o que se deseja saber sobre  $\mathcal{X}$  pode ser respondido em termos de  $f$

Por exemplo, fazendo  $B = [a, b]$ , temos:

$$P\{a \leq \mathcal{X} \leq b\} = \int_a^b f(x)dx \quad (9)$$

Se fizermos  $a = b$ :

$$P\{\mathcal{X} = a\} = \int_a^a f(x)dx = 0 \quad (10)$$

Colocando em palavras, essa equação diz que a probabilidade de que uma variável aleatória contínua assuma qualquer valor específico é zero. Por fim, para uma variável aleatória contínua:

$$P\{\mathcal{X} < a\} = P\{\mathcal{X} \leq a\} = F(a) = \int_{-\infty}^a f(x)dx \quad (11)$$

## Distribuição Normal

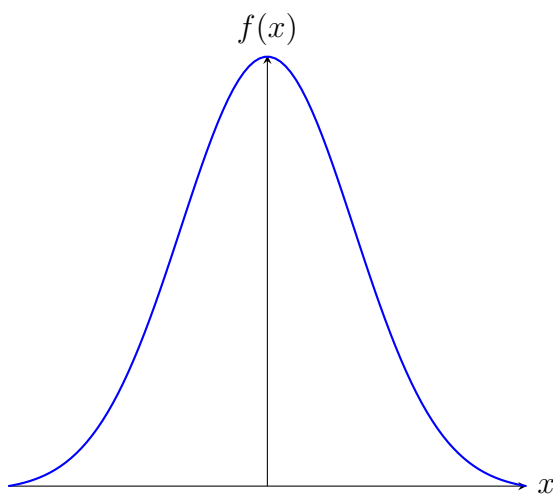
A distribuição normal, também conhecida como distribuição gaussiana, é uma das mais importantes e amplamente utilizadas distribuições de probabilidade contínuas. Possuindo dois parâmetros principais, o desvio padrão ( $\sigma$ ) e a média ( $\mu$ ), ela pode ser descrita matematicamente como:

$$\mathcal{X} \sim N(\mu, \sigma) \rightarrow P(\mathcal{X} = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right]} \quad (12)$$

Essa distribuição é simétrica em torno da média  $\mu$ , com a forma de sino. O parâmetro  $\sigma$  controla a dispersão dos dados ao redor da média. Quanto maior o  $\sigma$ , mais dispersos os dados estarão.

Essa distribuição é amplamente utilizada em estatística devido à sua simplicidade e ao fato de que muitos fenômenos naturais e artificiais têm distribuições que se assemelham à distribuição normal. Por exemplo, na modelagem de fenômenos naturais, como a altura de uma população, ou em processos artificiais, como erros de medição.

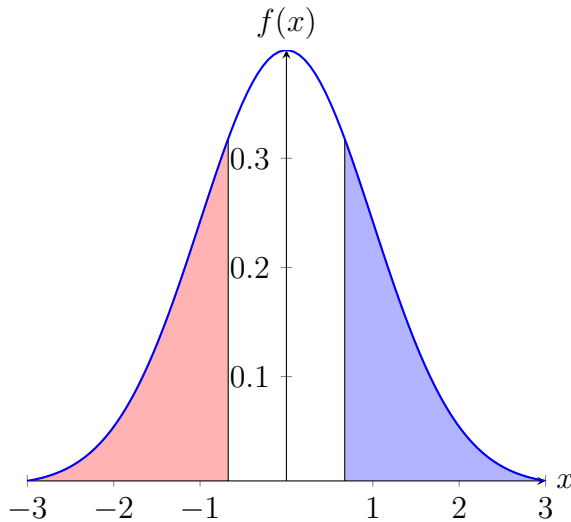
Tal distribuição possui o seguinte comportamento gráfico:



A linha central, em preto, se refere à média ( $\mu$ ) da nossa distribuição, a Normal é frequentemente usada para descrever o comportamento de muitas variáveis existentes no mundo, como peso, altura e até votos de uma determinada amostra ou população.



A Normal também nos ajuda a compreender porque estamos usando o Percentil 75% para determinarmos as músicas mais votadas, isso porque nela graficamente fica claro que os números acima dele são os 25% mais votados. Como pode ser visto neste gráfico, na região em azul:



## 5 Recursos Impostos durante a Eleição

### 5.1 Primeiro Turno

#### 5.1.1 Recurso I - Vandr  Monteiro

**Ac rd o n  001/2024 da Comiss o Eleitoral Calangu stica - Resposta a recurso**

**Impetrante:** *Vandr  Monteiro*

Data: 29/04/2024

**Ementa:** inadequa  o de m sica a crit rios previamente estabelecidos

**Relat rio:**

Trata-se de recurso impetrado por *Vandr  Monteiro* no sentido de invalidar a candidatura da m sica "Partido Alto", de Chico Buarque, interpretada por C ssia Eller, na lista de Bras lia, sob os seguintes argumentos: 1. O autor/compositor da m sica n o   de Bras lia (em que pese o impetrante ser f  de verdade do Chico Buarque)

**Voto:** Trecho do voto de um dos membros da **CEC** (*Carluxo Moura*):

"O que torna uma m sica ser de X lugar? Poderia ser o compositor, poderia ser o int rprete. Mas neste caso, Partido Alto j  foi gravado por diferentes pessoas antes da

Cássia. Ela não foi A interprete que lançou a música".

**Decisão:**

Após deliberação, a **CEC**, à unanimidade, acorda em dar provimento ao recurso. Retire-se a música da lista. Notifiquem-se os envolvidos.

**Encaminhamento:**

A **CEC** já notificou o autor da referida sugestão e ofereceu a ele a possibilidade de substituição da música por uma outra até amanhã, se assim desejar. Até o momento, ele disse que está deveras tranquilo.

### 5.1.2 Recurso II - Karen Duque

**Acórdão nº 002/2024 da Comissão Eleitoral Calanguística - Resposta a consulta**

**Autora:** *Karen Duque*

Data: 29/04/2024

Ementa: **duplicidade de música na lista de Brasília**

**Relatório:**

Trata-se de consulta realizada por *Karen Duque* questionando o fato de a música "Perfume dela", de Martinha do Coco, constar duas vezes na lista de Brasília, ainda que em versões diferentes (uma da própria Martinha e outra do Chinelo de Couro). Considera a interessada que:

1. "Se os votos forem divididos entre as duas músicas isso pode dividir votos de pessoas que querem a mesma coisa".

**Decisão:**

Após deliberação, a **CEC** decidiu por consultar as autoras das sugestões em duplicidade sobre a manutenção ou não das duas versões e, em caso de exclusão de uma, oferecer a possibilidade de incluir mais uma música no lugar da música excluída.

**Encaminhamento:**

As autoras das sugestões, em acordo, decidiram manter apenas uma versão (a da própria Martinha), e uma delas encaminhou uma nova música para ser acrescentada à lista no lugar da excluída. A **CEC** já realizou as mudanças.

### 5.1.3 Recurso III - Vandr  Monteiro

**Ac rd o 003/2024**

**Requerente:** *Procurador da Frente Ampla PopoLamas (Vandr  Monteiro)*

Data: 7/03/2024

**Ementa:**

Trata-se de requerimento de autoriza  o para fus o das chapas Meu Erro e  culos em uma  nica candidata, doravante denominada "pot-pourri Meu Erro/ culos". Argumenta o requerente que a fus o ser  ben fica para o pleito, pois permitir  maior precis o na escolha dos eleitores al m de orientar o trabalho a ser executado no caso de vit ria da candidata em quest o.

**Votos:**

Em primeiro lugar,   importante observar o cuidado na elabora  o do requerimento com o mais elevado n vel de vern culo fofiano burocr tico. T  de parab ns.

A CEC, depois de longa delibera  o, com v rios votos divergentes, analisou ponto a ponto argumentos apresentados nas discuss es sobre esse assunto pol mico. Deixando as formalidades de lado, chegamos  s seguintes conclus es:

No m rito, a proposta   v lida. Entendemos que um pot-pourri   um candidato t o v lido quanto qualquer outro. Mesmo que n o seja do agrado de todos, isso n o   diferente de qualquer outra m sica que n o seja um pot-pourri, que tamb m pode ser boa ou ruim.

Com rela  o a dificuldades t cnicas que porventura surjam ao juntar m sicas em pot-pourri, tamb m entendemos que n o s o diferentes de dificuldades t cnicas que possam surgir em adapta  es de qualquer outra m sica. De fato, j  temos pot-pourris no repert rio, e n o cabe a n s estabelecer nenhum crit rio excludente quanto a isso. N o existe um limite para isso, sendo essa uma decis o subjetiva. Quando uma pessoa decide inscrever um pot-pourri como candidata, ela pode ganhar votos por contemplar mais de uma m sica, mas tamb m pode perder votos de pessoas que acham que as m sicas n o combinam, ou que s  gostam de uma das m sicas envolvidas. Essa an lise de custo/benef cio cabe exclusivamente a quem sugere a m sica.

A CEC tamb m compreende que as m sicas envolvidas no processo fizeram sua inscri  o no per odo correto e da forma correta. Ent o elas podem ter interesse em se unir em uma chapa, e esse interesse   leg timo e pode ser contemplado.

Na forma, por m, temos um problema, que foi o prazo. Embora a CEC de fato n o

tenha previsto com antecedência essa possibilidade, e foi uma discussão que surgiu depois da liberação das playlists, nada teria impedido a chapa solicitante de ter se articulado e feito o requerimento depois de abertas as playlists, mas antes de abriremos o período de votação. De fato, mudarmos os candidatos agora vai nos obrigar a fazer um novo forms e cancelar todos os votos feitos até o momento (que já tinham sido muitos), o que vai nos dar um certo trabalho. E ainda teríamos que verificar se a Frente Ampla Natiruts quer fazer o mesmo. É de se destacar que entre o lançamento das candidatas e a abertura da votação tivemos 6 dias, tempo este suficiente para que correções, sugestões ou mesmo trocas fossem realizadas (o que de fato ocorreu). Estando a votação em curso, não é razoável trocar as candidaturas. Então, repetindo, consideramos o pedido justo e válido, mas por causa dessa questão prática de prazo e de retrabalho (que em parte tem a ver com o fato de não termos a regra claramente prevista com antecedência), vamos indeferi-lo.

No entanto, a CEC coloca aqui o compromisso de, na próxima eleição, assim que divulgada a playlist, fazer uma chamada, com a devida publicidade, para os candidatos que desejam fazer fusões. Divulgaremos a regra, o prazo, tudo certinho, e abriremos o forms para votação apenas depois de esgotado esse prazo.

Para este pleito que já está acontecendo, a nossa sugestão para as duas Frentes Amplas que se manifestaram até este momento (PopoLamas e PopoRuts) permanece sendo a de que escolham uma música para concentrar os votos e depois façam um lobby para um pout-pourri. É claro que nesse caso não existe uma garantia 100% de que isso vá acontecer, mas é uma possibilidade real e factível.

Acreditamos que uma regra clara sobre as alianças e fusões vai tornar os próximos pleitos mais eficazes e transparentes, e vamos nos empenhar nisso.

Nos colocamos à disposição para dúvidas e desabafos, e também, sempre, para subornar elogios.

Atenciosamente,

CEC (Carluxo, Lalá e Luizinho)

#### 5.1.4 Recurso IV - Phydia

**Acórdão 004/2024**

Data: 09/05/2024

**Requerente: *Phydia***

**Ementa:**

Trata-se de solicitação de extensão do prazo de votação em 24h ou 48h, considerando os recentes acontecimentos no Rio Grande do Sul e o surgimento de novas tocadass, fatos que estão ocupando muito a mente dos eleitores.

**Voto:**

A CEC, à unanimidade, deliberou pelo deferimento do pedido. Não houve argumentos em contrário.

**Encaminhamento:**

Fica estendido o prazo de votação até as 18h de domingo, 12 de maio.

## 6 Tabelas Completas de Resultado

### 6.1 Lista Geral

### 6.2 Lista Brasília

## 7 Referências Bibliográficas

BUSSAB, W. d. O.; MORETTIN, P. A. Estatística básica. In: **Estatística básica**. [S.l.: s.n.], 2010. p. xvi–540.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COUNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfán van der; MILLMAN Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 56 – 61.

ROSS, S. **Probabilidade: um curso moderno com aplicações**. [S.l.]: Bookman Editora, 2009.

Música	Votos	Critério Média	Percentil
Kilario - Di Melo	24	Aprovada	True
Libertango - Astor Piazzolla (Yo-Yo Ma)	13	Aprovada	True
Ob-La-Di,Ob-La-Da - The Beatles	12	Aprovada	True
Chevette Azul - A Espetacular Charanga do França	12	Aprovada	True
Bumbum de Ouro - Gloria Groove	11	Aprovada	True
Olhos Coloridos - Sandra de Sá	10	Aprovada	True
Funkytown/ I Feel Good - Lucky Chops	10	Aprovada	True
Alma Sebosa - Johnny Hooker	8	Aprovada	False
Que Beleza - Tim Maia	8	Aprovada	False
Last Nite - The Strokes	7	Aprovada	False
Confidências - Petrúcio Amorim	6	Aprovada	False
Rapunzel - Daniela Mercury	6	Aprovada	False
Deixe-se Acreditar - Chinaina,Felipe S	5	Reprovada	False
Sandra Rosa Madalena - Sidney Magal	4	Reprovada	False
Sexy Yemanjá - Pepeu Gomes	3	Reprovada	False
Magical Mystery Tour - The Beatles	3	Reprovada	False
Coisa Boa - Gloria Groove	3	Reprovada	False
Smells Like Teen Spirit - Nirvana	2	Reprovada	False
Jogo de Calçada - Os Mutantes	2	Reprovada	False
Iko Iko - The Gramophone Allstars	2	Reprovada	False
Cumbia da Praia - Academia da Berlinda	2	Reprovada	False
Jorge de Altinho	2	Reprovada	False
Quem Sabe - Los Hermanos	1	Reprovada	False
Caçada - Chico Buarque	1	Reprovada	False
Start Wearing Purple - Gogol Bordello	1	Reprovada	False
Nitro Bong (Radio Mix) - Webra	1	Reprovada	False
Abre Alas - Ivan Lins	1	Reprovada	False
Proud Mary - Ike e Tina Turner	1	Reprovada	False
Lindo Lago do Amor - Gonzaguinha	1	Reprovada	False

Tabela 9: Tabela dos Resultados Completos 1o Turno - Lista Brasília

Música	Votos	Critério Média	Percentil
Pretin - Flora Matos	16	Aprovada	True
Seria o rolex ? - Móveis Coloniais de Acaju	16	Aprovada	True
Pra Enfeitar - Saci Wèrè	12	Aprovada	True
O Beco - Paralamas do Sucesso	11	Aprovada	True
Piloto - Nattan	11	Aprovada	True
Natiruts Reggea Power - Natiruts	10	Aprovada	True
Óculos - Paralamas do Sucesso	9	Aprovada	False
Perfume Dela - Martinha do Coco	9	Aprovada	False
Travessia do Eixão - Liga Tripa (Versão Legião Urbana)	8	Aprovada	False
O Carcará e a Rosa - Natiruts	8	Aprovada	False
Veneno de Cobra - Emília Monteiro e Dona Onete	7	Aprovada	False
Juriti - Liga Tripa	6	Reprovada	False
Meu Erro - Paralamas do Sucesso	6	Reprovada	False
Sorri,sou Rei - Natiruts e Claudia Leite	5	Reprovada	False
Bião do Cerrado - Kirá e a Ribanceira	4	Reprovada	False
Tambores do Paranoá - Martinha do Coco	4	Reprovada	False
Presente de um Beija Flor - Natiruts	4	Reprovada	False
Deixa o Menino Jogar - Natiruts	3	Reprovada	False
Tempestade - Maskavo Roots	3	Reprovada	False
Oração - Pé de Cerrado	3	Reprovada	False
Esquilo Não Samba - Móveis Coloniais de Acaju	2	Reprovada	False
Ícaro - Joe Silhueta	1	Reprovada	False
Coco do Cerrado - Martinha do Coco	1	Reprovada	False
Esperança no Vermelho - A Tuba Atômica do Planalto	1	Reprovada	False
Sol no Rosto - Remobília	1	Reprovada	False
Xote dos Sonhos - Pé de Cerrado e Dominginhos	1	Reprovada	False

Tabela 10: Tabela dos Resultados Completos 1o Turno - Lista Brasília

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. Zenodo, 2020.  
Disponível em: <<https://doi.org/10.5281/zenodo.3509134>>.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J.; van der Walt, S. J.; BRETT, M.; WILSON, J.; MILLMAN, K. J.; MAYOROV, N.; NELSON, A. R. J.; JONES, E.; KERN, R.; LARSON, E.; CAREY, C. J.; POLAT, İ.; FENG, Y.; MOORE, E. W.; VanderPlas, J.; LAXALDE, D.; PERKTOLD, J.; CIMRMAN, R.; HENRIKSEN, I.; QUINTERO, E. A.; HARRIS, C. R.; ARCHIBALD, A. M.; RIBEIRO, A. H.; PEDREGOSA, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. **Nature Methods**, v. 17, p. 261–272, 2020.