



Urna Eletrônica

Grupo 5

1

Mathias Soares

2

Pablo Maciel

3

Fábio Bezerra

4

Alansidney da Silva

5

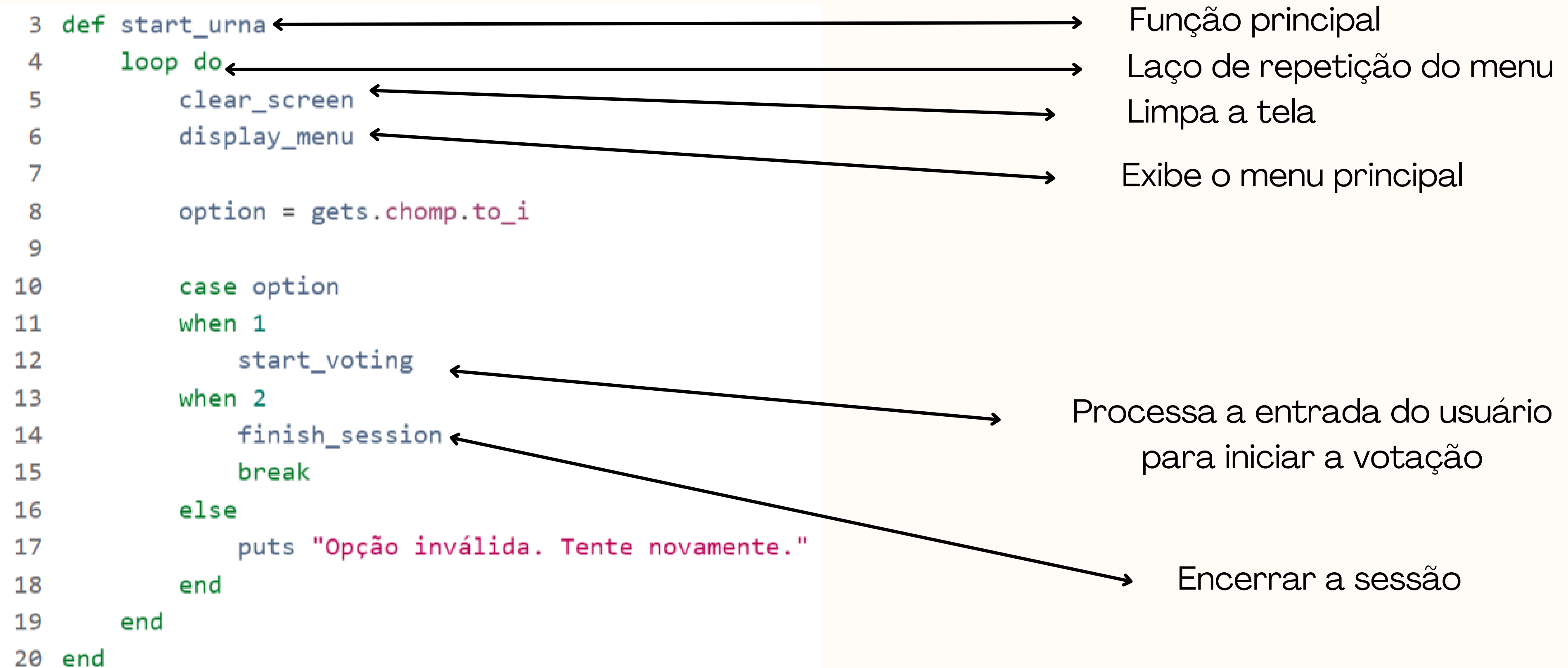
Milene Rayssa

Variáveis Globais

168 # Votos totais		
169 \$total_votes = 0	←————→	Contador total de votos.
170		
171 # Eleitores = CPFs que já		
172 \$voters = []	←————→	Array que armazena os CPFs dos eleitores que já votaram.
173		
174 # Votos por candidatos		
175 \$votes = {	←————→	Hash que armazena o número de votos recebidos por cada candidato, votos em branco, e votos nulos.
176 "JOAO" => 0,		
177 "MARIA" => 0,		
178 "PEDRO" => 0,		
179 "BRANCO" => 0,		
180 "NULO" => 0		
181 }		
182		
183 # Candidatos		
184 \$candidates = {	←————→	Hash que mapeia os números dos candidatos para seus respectivos nomes.
185 21 => "JOAO",		
186 23 => "MARIA",		
187 10 => "PEDRO",		
188 0 => "BRANCO"		

Estrutura Geral

O código é estruturado em diversas funções que lidam com diferentes aspectos do processo de votação. O fluxo principal é controlado pela função **start_urna**, que exibe um menu e processa a entrada do usuário para iniciar a votação ou encerrar a sessão.





Função display_menu

Exibe o menu principal com as opções de iniciar a votação ou encerrar a sessão.

```
21
22 def display_menu
23     puts "-----URNA-ELETRONICA-----"
24     puts "1 - INICIAR VOTACAO"
25     puts "2 - ENCERRAR SESSAO"
26     print "Digite: "
27 end
28
```



Função start_voting

Solicita o CPF do eleitor

Verifica a validade
do CPF

Exibe a lista de
candidatos

Valida o voto
do eleitor

```
29 def start_voting
30     cpf = get_cpf
31     if cpf && check_cpf(cpf)
32         show_candidates
33         validate_vote(cpf)
34     else
35         puts "ESSE CPF É INVALIDO OU JA FOI USADO NESTA VOTACAO!\nTENTE NOVAMENTE"
36         start_voting
37     end
38 end
39
```

Função `get_cpf` e Função `check_cpf`

Solicita e valida o CPF do eleitor. Se o CPF for válido (**valid_cpf?**), ele é retornado; caso contrário, exibe uma mensagem de erro.

```
40 def get_cpf
41   print "Digite seu CPF (apenas números): "
42   cpf = gets.chomp
43   if valid_cpf?(cpf)
44     cpf
45   else
46     puts "CPF inválido."
47     nil
48   end
49 end
50
51 def check_cpf(cpf)
52   !$voters.include?(cpf)
53 end
54
```

Verifica se o CPF fornecido já foi utilizado na votação.



Função `valid_cpf`?

Verifica se o CPF é válido

Checa se o CPF tem
exatamente 11 dígitos.

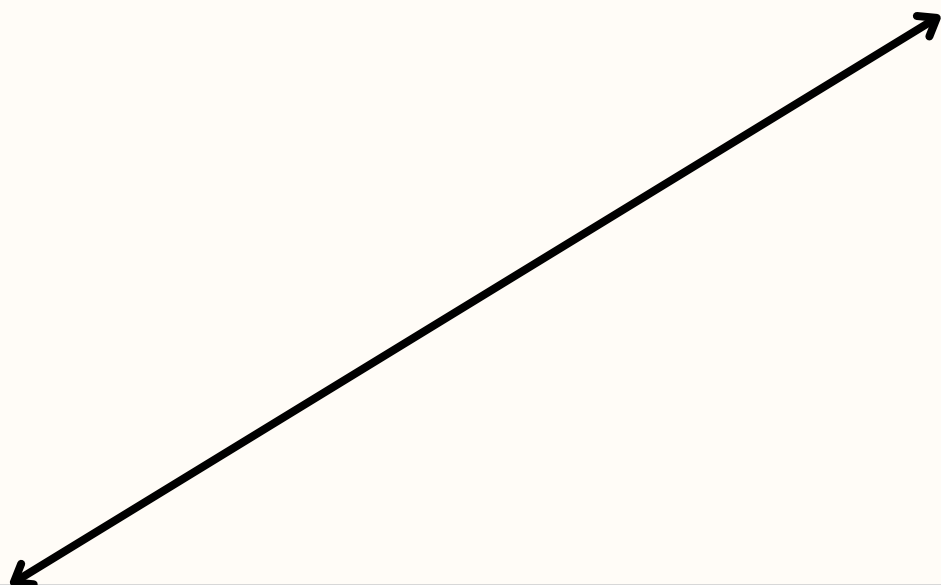
Calcula e verifica os
dígitos verificadores
usando a função função
calculate_verifier

```
55 def valid_cpf?(cpf)
56   return false unless cpf.match?(/^\\d{11}$/)
57
58   digits = cpf.chars.map(&:to_i)
59   first_verifier = calculate_verifier(digits, 9)
60   second_verifier = calculate_verifier(digits, 10)
61
62   first_verifier == digits[9] && second_verifier == digits[10]
63 end
```

Função calculate_verifier

[Voltar para o slide de Temas](#)

Calcula os dígitos verificadores do CPF.
É uma função auxiliar usada em
valid_cpf?

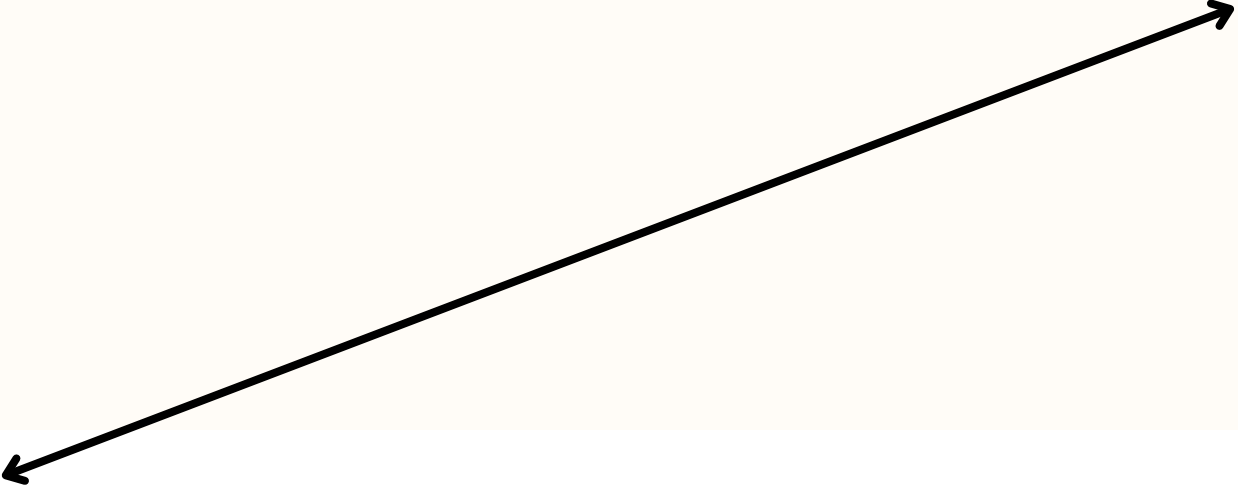


```
65 def calculate_verifier(digits, length)
66   sum = digits[0...length].each_with_index.reduce(0) do |acc, (digit, index)|
67     acc + digit * (length + 1 - index)
68   end
69
70   remainder = sum % 11
71   remainder < 2 ? 0 : 11 - remainder
72 end
73
```



Função show_candidates

Exibe a lista de candidatos e solicita que o eleitor digite o número do candidato em quem deseja votar.



```
74 def show_candidates
75     clear_screen
76     puts "|-----LISTA DE CANDIDATOS-----|"
77     puts "|      DIGITO      |           NOME          |"
78     puts "|-----|-----|"
79     puts "|      21      |           JOAO          |"
80     puts "|      23      |           MARIA          |"
81     puts "|      10      |           PEDRO          |"
82     puts "|       0      |           BRANCO         |"
83     puts "|-----|\n\n|"
84     print "DIGITE O NUMERO DO CANDIDATO QUE DESEJA VOTAR: "
85 end
```



Função `validate_vote`

Valida o voto digitado
pelo eleitor

Verifica se o voto é um
número.

Checa se o número
corresponde a um
candidato ou se é um voto
em branco (0).

Confirma o voto com o
eleitor

```
87 def validate_vote(cpf)
88   vote = gets.chomp
89   if vote.match?(/^\\d+$/) # apenas dígitos
90     vote = vote.to_i
91     if $candidates.key?(vote) || vote == 0
92       confirm_vote(vote, cpf)
93     else
94       puts "Número inválido. Voto considerado como nulo."
95       confirm_vote(1, cpf) # Voto nulo
96     end
97   else
98     puts "Entrada inválida. Por favor, digite apenas números."
99     validate_vote(cpf)
100  end
101 end
102
```

Função confirm_vote e Função compute_vote

[Voltar para o slide de Temas](#)

```
103 def confirm_vote(vote, cpf)
104     candidate = $candidates[vote] || "NULO"
105     print "CONFIRMAR VOTO EM #{candidate}? (S/N): "
106
107     confirm = gets.chomp.upcase
108     if confirm == 'S'
109         compute_vote(candidate, cpf)
110         puts "VOTO FEITO COM SUCESSO!"
111     else
112         puts "Voto não confirmado. Iniciando novamente."
113         validate_vote(cpf)
114     end
115 end
116
117 def compute_vote(candidate, cpf)
118     $votes[candidate] += 1
119     $voters << cpf
120     $total_votes += 1
121 end
122
```

Solicita confirmação
do voto

Registra o voto se confirmado;
caso contrário, reinicia o
processo de validação do voto.

Incrementa o contador de votos do
candidato ou do voto nulo.

Adiciona o CPF do eleitor à lista de
eleitores que já votaram.



Função clear_screen

Função finish_session

Função display_results

Limpa a tela do terminal. Utiliza um comando diferente dependendo do sistema operacional

Encerra a sessão de votação

```
123 def clear_screen
124   clear_command = Gem.win_platform? ? 'cls' : 'clear'
125   sleep(1)
126   system(clear_command)
127 end
```

```
128
129 def finish_session
130   clear_screen
131   puts "Votação encerrada. Resultados finais:"
132   display_results
133   save_results
134 end
```

```
135
136 def display_results
137   puts "\nResultado da votação:"
138   $votes.each do |candidate, votes|
139     puts "#{candidate}: #{votes} votos"
140   end
141   puts "Total de votos: #{$total_votes}"
142 end
```

Exibe os resultados finais

Salva os resultados em um arquivo CSV

Exibe os resultados da votação, incluindo o número de votos para cada candidato e o total de votos.



Função `save_results`

Salva os resultados da votação em um arquivo CSV. Se o arquivo já existir, pergunta ao usuário se deseja sobrescrever o arquivo ou salvar com um novo nome.

```
144 def save_results
145   arquivo_existente = "resultados_votacao.csv"
146   if File.exist?(arquivo_existente)
147     puts "O arquivo 'resultados_votacao.csv' já existe. Deseja sobrescrever? (S/N)"
148     resposta = gets.chomp.upcase
149     if resposta == 'N'
150       puts "Digite um novo nome para o arquivo (sem extensão):"
151       novo_nome = gets.chomp
152       arquivo_existente = "#{novo_nome}.csv"
153     end
154   end
155
156   CSV.open(arquivo_existente, "w") do |csv|
157     csv << ["Candidato", "Votos"]
158     $votes.each do |candidate, votes|
159       csv << [candidate, votes]
160     end
161   end
162
163   puts "Os resultados foram salvos no arquivo '#{arquivo_existente}'."
164 end
165
```

Abre o arquivo CSV para escrita

Escreve o cabeçalho

Escreve os resultados de cada candidato