



Urna eletrônica em Ruby

Projeto 1 - Grupo 5 - FAP 2024

Alansidney da Silva Júnior
Fábio Bezerra de Andrade
Mathias Ernandes Oliveira Soares
Pablo Jorge Maciel

Índice

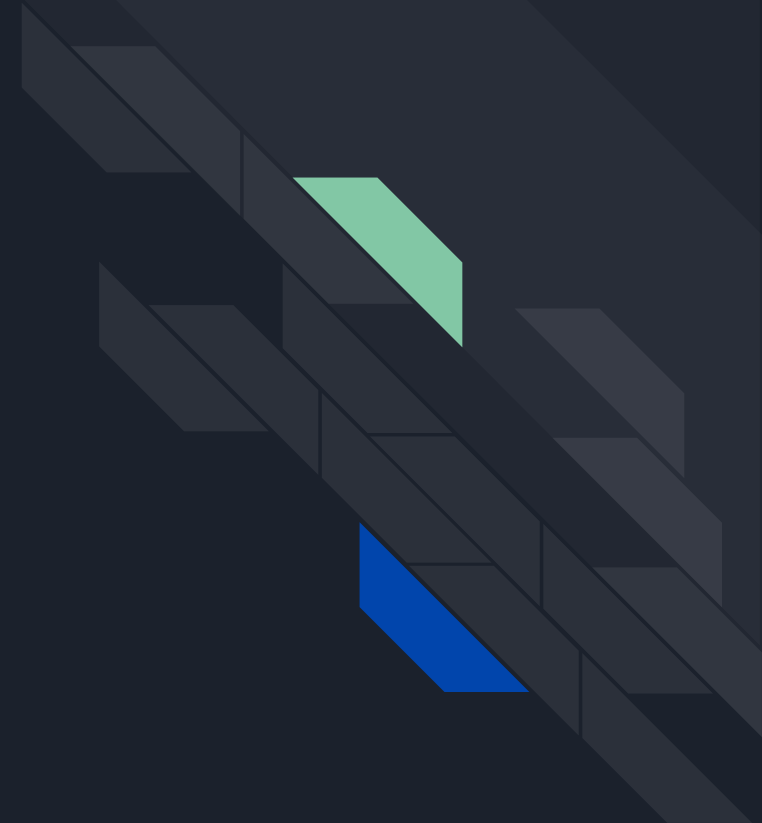
Visão geral

Fluxograma do projeto

Análise da implementação


Execução do programa

Normalização (1FN)





VISÃO GERAL



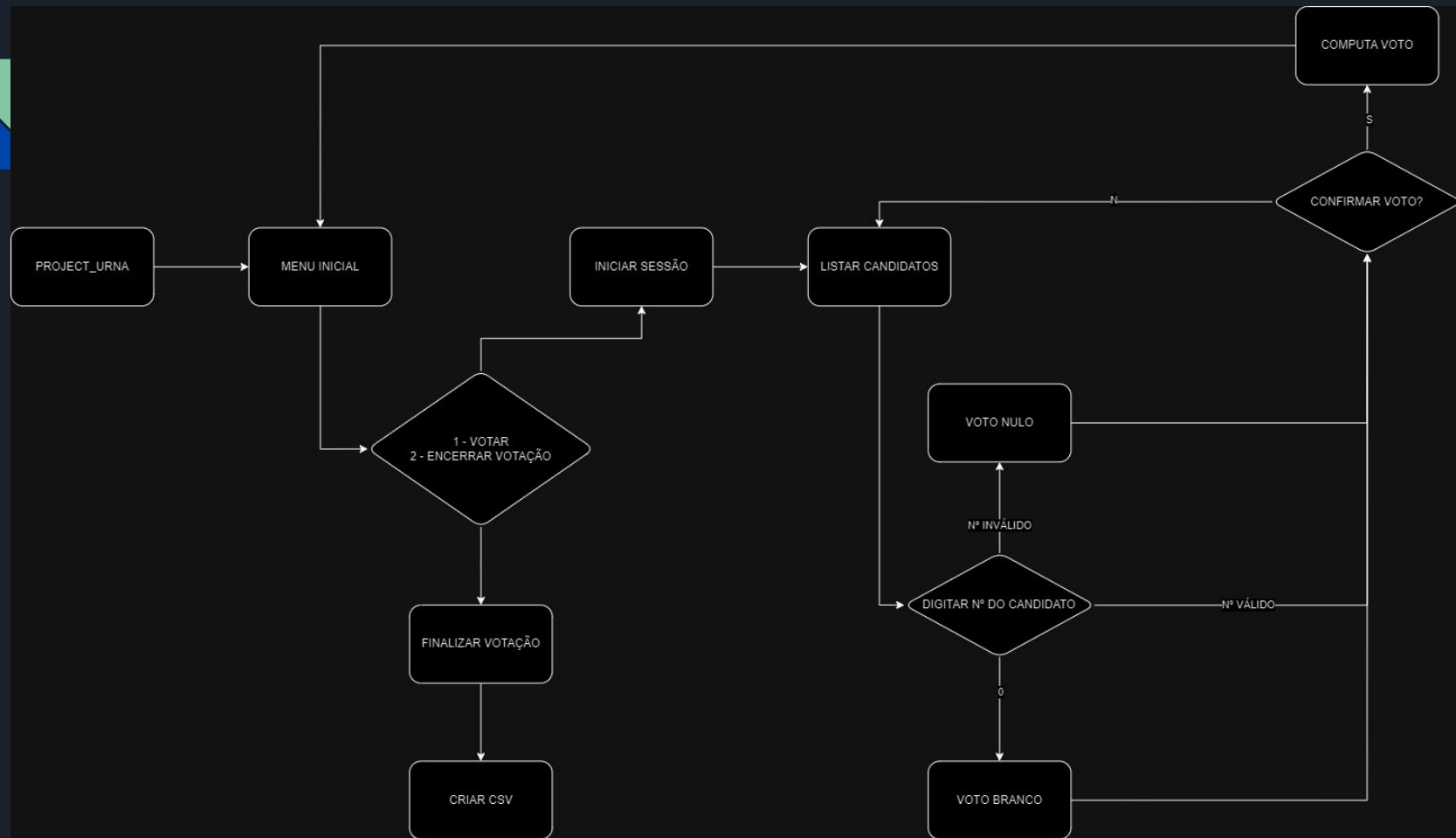
Este projeto implementa uma urna eletrônica simples em Ruby, permitindo que usuários votem em candidatos pré-definidos. O sistema contabiliza os votos e exibe o resultado da eleição.

Funcionalidades:

- Exibir lista de candidatos.
- Permitir que os usuários votem em um candidato.
- Contabilizar os votos.
- Exibir o resultado da eleição.



FLUXOGRAMA





ANÁLISE DA IMPLEMENTAÇÃO

SETUP INICIAL E DISPLAY DO MENU

```
urna.rb x
urna.rb
1  require 'csv'
2
3  candX = 0
4  candY = 0
5  candZ = 0
6  brancos = 0
7  nulos = 0
8
9  clear_command = Gem.win_platform? ? 'cls' : 'clear'
10
11  puts 'Bem Vindo ao sistema de Urna Eletronica do grupo 5 da Softex'
12
13  print "Digite a quantidade de eleitores para essa votação: "
14  voter_number = gets.chomp.to_i
15
16  # Loop principal para cada eleitor
17  for voter_count in 1..voter_number
18    system(clear_command)
19    puts "***** Menu: *****"
20    puts "=====
21    puts "1 - Votar."
22    puts "2 - Encerrar votação."
23
24    escolha = gets.chomp.to_i
25
```

Importação da biblioteca CSV

Inicialização de variáveis para contagem de votos

Recebe a quantidade de eleitores para votação e inicia o loop

Display de saudação e menu inicial

Recebe a escolha do usuário (1 - Votar ou 2 - Encerrar votação)

LISTAR CANDIDATOS

```
26 case escolha
27 when 1
28   system(clear_command)
29   loop do
30     puts "Escolha o(a) seu(sua) candidato(a): \n 89 - Pedro \n 47 - José
31
32     print "Digite o número do(a) candidato(a): "
33     voto = gets.chomp
34
35     if voto.match?(/\A\d+\z/) # Verifica se o input é um número
36       voto = voto.to_i
37
38       case voto
39       when 89
40         candidato = 'Candidato(a) Pedro'
41       when 47
42         candidato = 'Candidato(a) José'
43       when 51
44         candidato = 'Candidata Maria'
45       when 0
46         candidato = 'Branco'
47       else
48         candidato = 'Nulo'
49       end
50
51       system(clear_command)
52
```

Caso opção 1, inicia loop

Lista candidatos e aguarda N° do usuário

Verifica se a entrada é um dígito

Atribui a variável "candidato" o nome correspondente ao N° digitado

VALIDAÇÃO DE VOTO

```
53 puts "Confirma voto para #{candidato}? (S/N)"
54 confirmacao = gets.chomp.upcase
55
56 if confirmacao == 'S'
57     if voto == 89
58         candX += 1
59     elsif voto == 47
60         candY += 1
61     elsif voto == 51
62         candZ += 1
63     elsif voto == 0
64         brancos += 1
65     else
66         nulos += 1
67     end
68     break
69 else
70     system('clear_command')
71     puts "Opção inválida. Digite S para confirmar ou
72     sleep(2)
73 end
74 else
75     puts "Entrada inválida. Por favor, digite apenas n
76     sleep(2)
77 end
78 end
```

Solicita confirmação de voto

Incrementa o voto à variável do candidato/branco/nulo

Caso opção inválida ou não confirmação do voto, volta a listar os candidatos

VALIDAÇÃO DE VOTO

```
89 # Determina o vencedor
90 if candX > candY && candX > candZ
91     vitorioso = 'Candidato(a) Pedro'
92 elif candY > candX && candY > candZ
93     vitorioso = 'Candidato(a) José'
94 elif candZ > candX && candZ > candY
95     vitorioso = 'Candidata Maria'
96 else
97     vitorioso = 'Aconteceu um Empate'
98 end
99
100 system(clear_command)
101
102 # Exibe os resultados
103 total_votos = candX + candY + candZ + brancos + nulos
104 puts "O ganhador desta eleição é:\n\n ***** #{vitorioso} *****\n"
105 puts "\n#{total_votos} votos totalizados, sendo:"
106 puts "\n Candidato(a) Pedro: #{candX} votos."
107 puts " Candidato(a) José: #{candY} votos."
108 puts " Candidata Maria: #{candZ} votos."
109 puts " Brancos e nulos: #{brancos + nulos} votos.\n Sendo:"
110 puts " Brancos: #{brancos}"
111 puts " Nulos: #{nulos}"
112
113 system(clear_command)
```

Calcula qual variável com maior valor e determina o candidato vencedor

Mostra o resultado da votação



EXECUÇÃO DO CÓDIGO



NORMALIZAÇÃO DE BD

1ª FORMA NORMAL (1FN)



NORMALIZAÇÃO

A normalização de banco de dados é um processo essencial no design de bases de dados relacionais, que organiza tabelas e colunas para reduzir a redundância e melhorar a integridade dos dados. É dividida em várias formas normais, cada uma com regras e requisitos específicos.

Objetivos da normalização:

- Eliminar redundância dos dados
- Assegurar integridade dos dados
- Facilitar manutenção no banco de dados
- Melhorar a flexibilidade da estrutura



1ª FORMA NORMAL

É a primeira etapa no processo de normalização de bancos de dados.

Para uma tabela estar em 1FN, ela deve atender aos seguintes requisitos:

1. Eliminar Grupos Repetidos: Cada tabela deve representar uma entidade única, e não deve haver múltiplos valores ou grupos de valores em uma única célula.
2. Garantir Atomicidade dos Dados: Cada campo deve conter apenas um valor indivisível e único.

EXEMPLO DE BD NÃO-NORMALIZADO

CANDIDATOS		
NOME	NUMERO	ENDEREÇO
PEDRO	89	RUA SAO JOAO, 54, BOA VIAGEM, RECIFE, PE
JOSÉ	47	RUA SANTO ANTONIO, 55, VILA RICA, JABOATAO, PE
MARIA	51	RUA SAO PEDRO, 57, BAIRRO NOVO, OLINDA, PE



EXEMPLO DE BD NORMALIZADO

CANDIDATOS						
NOME	NUMERO	LOGRADOURO	NUMERO	BAIRRO	CIDADE	ESTADO
PEDRO	89	RUA SAO JOAO	54	BOA VIAGEM	RECIFE	PE
JOSÉ	47	RUA SANTO ANTONIO	55	VILA RICA	JABOATAO	PE
MARIA	51	RUA SAO PEDRO	57	BAIRRO NOVO	OLINDA	PE



OBRIGADO PELA
ATENÇÃO