**Daniel Cavalcante Dourado**

Universidade Estadual de Feira de Santana(UEFS)

Engenharia de Computação

CEP 44077-040 – Rua Saracura, Santa Mônica

Feira de Santana – BA – Brasil

[daniel10dourado@gmail.com](mailto:daniel10dourado@gmail.com)

1.**Introdução**

Com a chegada das eleições majoritárias de 2018, o TSE( Tribunal Superior Eleitoral) está necessitando de um mecanismo capaz de realizar as votações que serão responsáveis por eleger os políticos vencedores, com o objetivo de auxiliar na segurança das eleições, torna-la mais simples e manifestar os resultados de maneira mais rápida, foi designado aos estudantes do primeiro de semestre de Engenharia de Computação da UEFS o trabalho de criar uma urna eletrônica.

Foi requerido um algoritmo capaz de suprir todas as necessidades que uma urna possui, bem como incrementar detalhes úteis, pensados pelo criador do software.

Para a criação desse algoritmo, foi utilizado estruturas condicionais, laços de repetição, entrada/saída de dados, operações matemáticas e contadores, todos esses recursos algorítmicos são utilizados a partir da entrada de dados, afim de contemplar tudo que foi planejado.

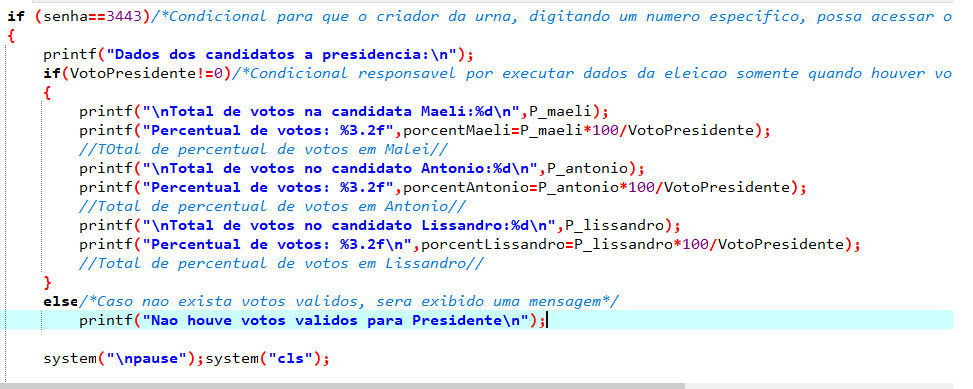
**2.Metodologia**

Para a criação do algoritmo, teve de ser feito um estudo sobre linguagem “C” de programação, requerida pelos idealizadores. A partir desse estudo, foi decidido criar primeiro o software que solucionasse o problema e depois o fluxograma que demonstrasse a estrutura do código em outra linguagem

Foram utilizados recursos dessa linguagem, com as funções “printf” e “scanf” foi possível a saída e entrada de dados, que controlaram as ações feitas pelo código; as estruturas condicionais “if”, “else if”, “else” e “ switch”, possibilitaram: a incrementação de valores às variáveis, quando a condição para isso fosse atendida, utilização de uma senha, conhecida pelo criador, para acessar devidamente os dados da votação e evitar erros no código.

Com o uso do laço de repetição “do while”, a votação pôde receber quantidades indeterminadas de votos, armazenando os dados de todas elas, assim como oferecer a correção do voto, tais laços são controlados a partir de certas entradas de dados, feitas pelo eleitor, de forma que o mesmo tenha mais autonomia para realizar seu voto; a inicialização de variáveis, com valor inicial zero e contadores dentro dos “switchs”, permitiram, ao final da votação, um relatório contendo informações requeridas pelos idealizadores, total e percentual dos votos nulo, branco, nos candidatos do sexo masculino e feminino, se haverá necessidade de Segundo turno e o total de votos.

A partir do desenvolvimento do fluxograma, foi montado todo o formato do algoritmo, quais componentes seriam utilizados, a ordem entre eles, qual função cada componente iria desempenhar e a forma como que os requisitos seriam concluídos. A figura 1 mostra a utilização de condicionais controlados por uma senha que, se digitada corretamente, segue para a verificação da existência de votos validos, não nulos e brancos, o uso de diversas saídas de dados, demonstrando valores importantes, e cálculos matemáticos feitos para conseguir porcentagens.

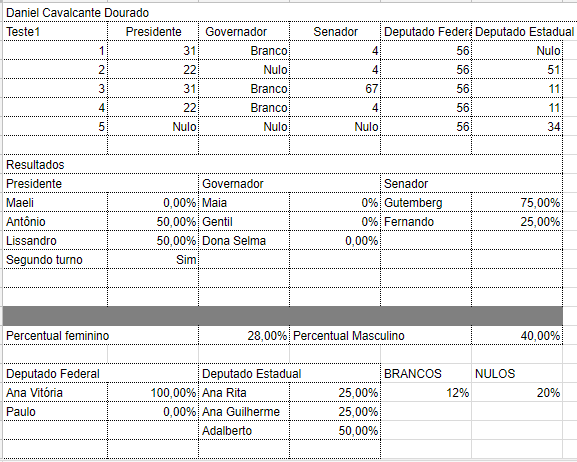


**Figura 1. Estrutura condicional, saida de dados e operações**

Para ter acesso a essa parte do código, o usuário deve digitar, quando requerido, uma senha especifica, após a digitação correta, o código entra em uma condicional para averiguar a existência de votos validos pois, sem essa estrutura, poderia haver divisões por 0 dentro das operações matemáticas, gerando um erro no programa. Após a condição ser atendida, as variáveis responsáveis por conter os dados dos candidatos são exibidas sendo, uma delas, feita através de um calculo matemático, ex: porcentagem de votos na candidata Maeli(porcentMaeli) equivale aos votos que ela teve (P\_Maeli)\*100/ votos para presidente (VotoPresidente).

**3. Resultados e Discussões**

Para verificar o desempenho e exatidão do algoritmo, foram realizados testes, com entradas de dados específicas para encontrar qualquer erro/falha que venha a existir. Esses testes foram feitos por meio de tabelas que, após todas as entradas, arquivam as estatísticas conseguidas, um desses testes é demonstrado na figura 2:



**Figura 2. Plano de Testes**

**4. Conclusão**

O algoritmo desenvolvido supriu todos os requisitos, funcionando sem erros e com um formato simples e explicativo, sem ser necessário um conhecimento a respeito de programação, ou seja, tanto alguém da área de programação, como um cidadão qualquer que irá votar, poderão realizar seus votos da mesma maneira.

Foram disponibilizados, ao longo do código, opções de correção de voto caso aconteça alguma digitação incorreta ou confusão com os números dos candidatos. Não só a execução do código ficou intuitiva, o código fonte em si possui diversos comentários que, mesmo sem conhecer as funções utilizadas, o individuo que venha a observa-lo ira compreender o uso de cada componente.

O programa poderia além de computar votos e fazer cálculos com eles, ter inteligência para declarar os vencedores de cada cargo no final das eleições, mesmo havendo segundo turno, coletar informações não so dos candidatos mas como dos próprios eleitores, gênero, região, etnia.

**5. Bibliografia consultada**

**Não foi utilizada nenhuma bibliografia para a criação do relatório.**