

## Trabalho - Pesquisa, Votação e Apuração Eleitoral

*quinta-feira, 10 de maio de 2012*

**Tópicos:** Classes, *arrays*, herança, polimorfismo, interface, *strings*, imagens gráficas, componentes GUI *Swing*, tratamento de eventos, modelagem UML, tratamento de exceção, arquivos e coleções.

---

Desenvolva um sistema em Java composto de 4 (quatro) módulos: **Candidatos**, **Pesquisa Eleitoral**, **Votação** e **Apuração Eleitoral**. A seguir uma breve descrição de cada módulo.

- **Candidatos:** permite o cadastro, a alteração, a consulta e a exclusão dos dados dos candidatos: nome, partido político e o número do candidato.
- **Pesquisa Eleitoral:** este módulo oferece as seguintes funcionalidades:

### 1. Cadastrar Pesquisa Eleitoral

Realiza a leitura dos seguintes dados:

- a) cargo público ao qual o candidato irá concorrer, por exemplo: "Presidente" (ver Figura 1);
- b) data de início da pesquisa no formato dd/mm/aaaa, onde dd = 1..31, mm = 1..12 e aaaa possui valor mínimo igual a 1995;
- c) data final da pesquisa no formato dd/mm/aaaa, onde dd = 1..31, mm = 1..12 e aaaa possui valor mínimo igual a 1995;
- d) o número de intenções de votos para cada candidato;
- e) o número de votos brancos e nulos;
- f) o número de eleitores indecisos, que não souberam ou não quiseram responder (ver Figura 2);
- g) o número de pessoas entrevistadas;
- h) o número de municípios em que a pesquisa foi realizada.

### 2. Relatório da Pesquisa Eleitoral

Exibe o resultado da pesquisa em formato gráfico, conforme os exemplos das Figuras 1 e 2. Usar as classes *Graphics* e/ou *Graphics2D* do pacote *java.awt*. O gráfico deve apresentar os seguintes dados:

- a) o cargo público;
  - b) o período da pesquisa eleitoral no formato mmm/aaaa, onde mmm é igual à abreviatura com três letras do nome do mês (jan, fev, mar... dez) e aaaa possui valor mínimo igual a 1995 (ver Figura 1);
  - c) o percentual das intenções de votos para cada candidato (ver Figura 1);
  - d) o percentual de votos brancos e nulos (ver Figura 2);
  - e) o percentual de pessoas que não souberam ou não quiseram responder (ver Figura 2);
  - f) o número de pessoas entrevistadas;
-

g) o número de municípios em que a pesquisa foi realizada.

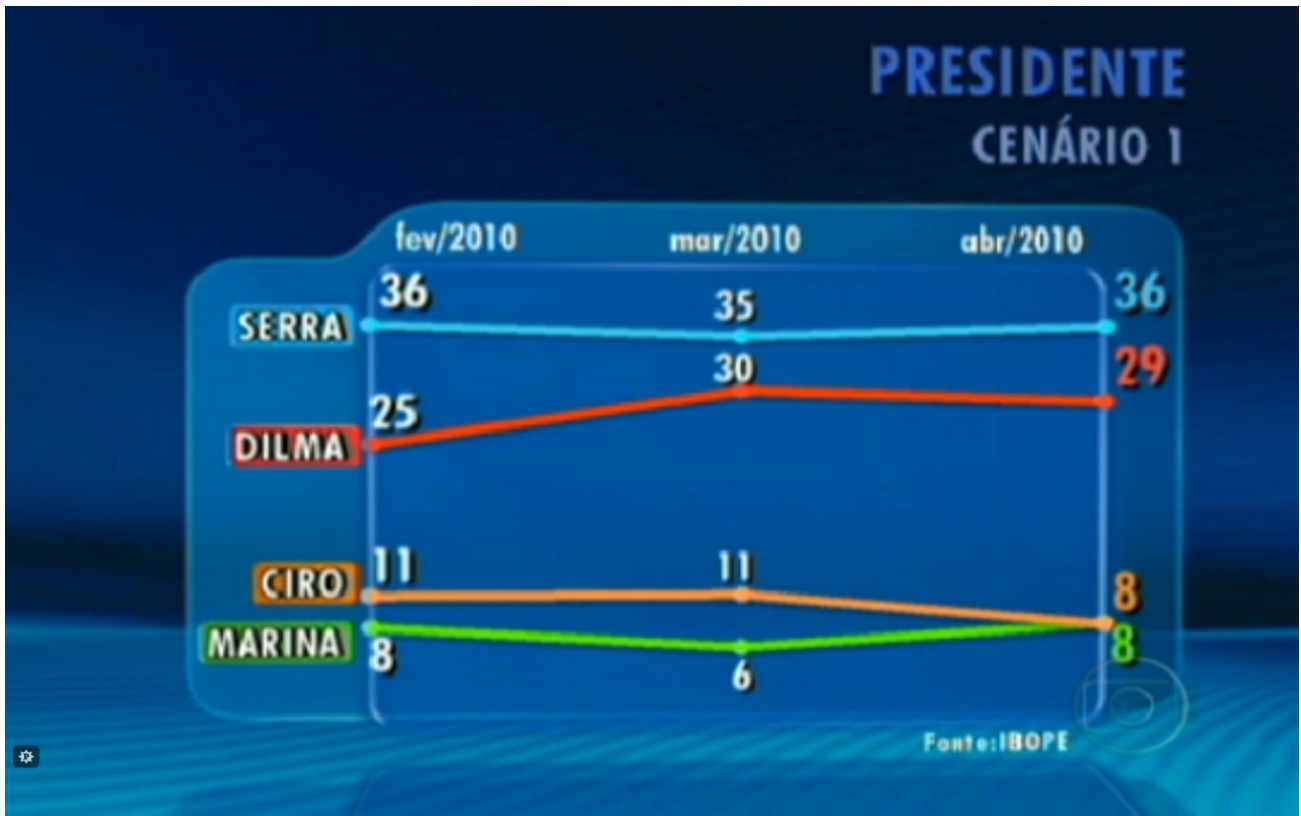


Figura 1 - Intenções de voto para presidente no primeiro turno.

Fonte: [www.g1.com.br](http://www.g1.com.br)

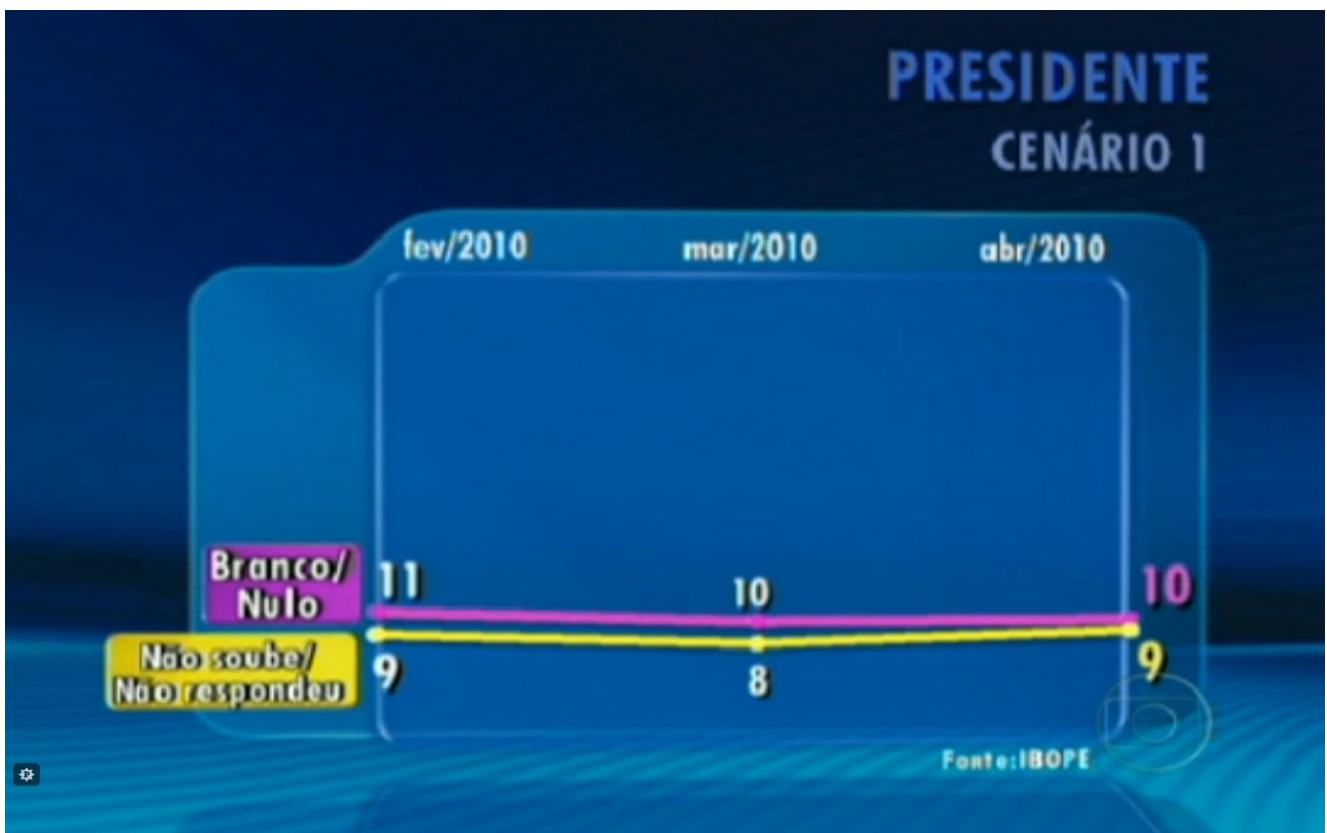
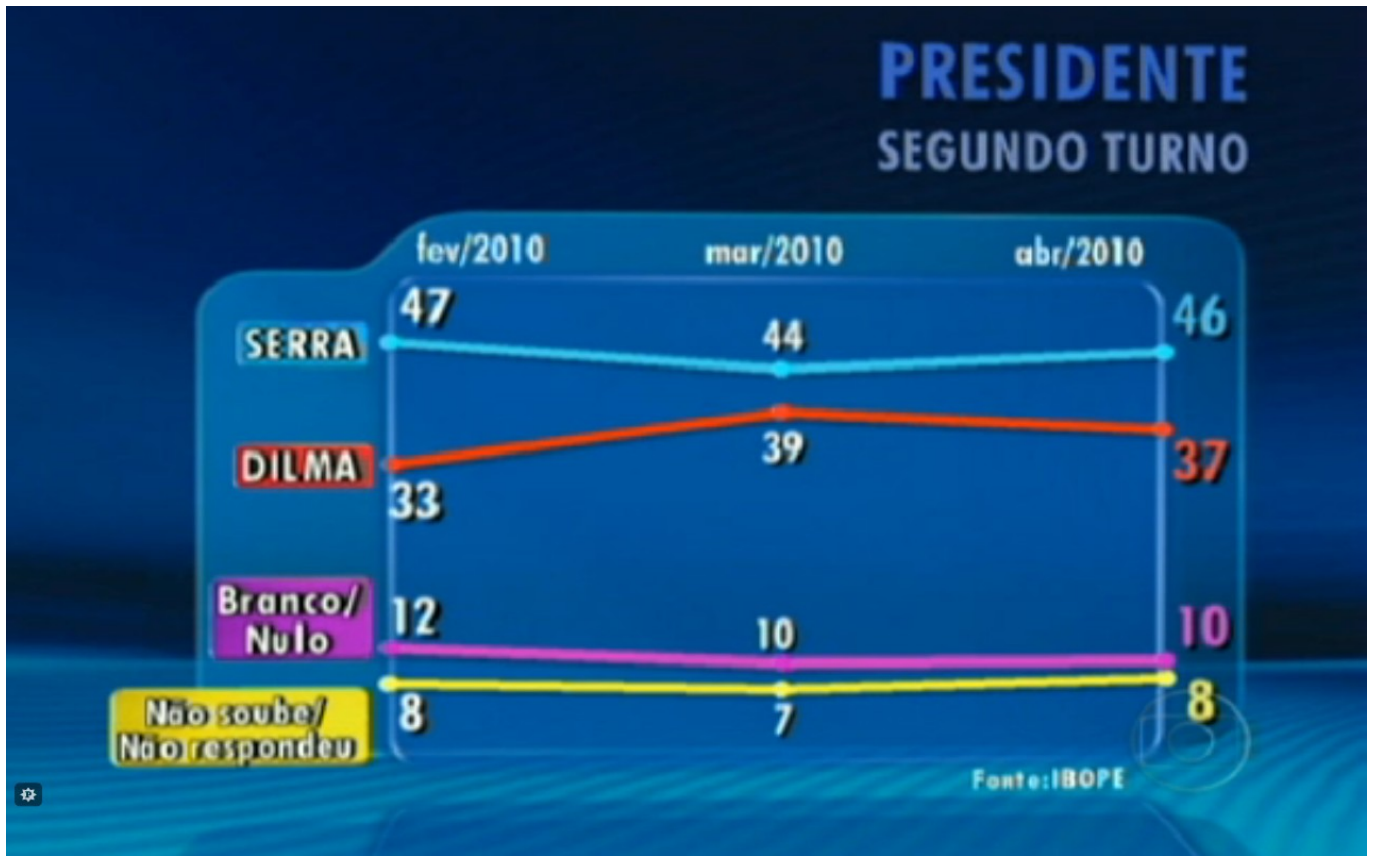


Figura 2 - Número de votos brancos e nulos pra presidente e de eleitores indecisos.

Fonte: [www.g1.com.br](http://www.g1.com.br)



**Figura 3** - Intenções de voto pra presidente no segundo turno. Número de votos brancos e nulos e de eleitores indecisos.

Fonte: [www.g1.com.br](http://www.g1.com.br)

- **Votação:** este módulo permite ao eleitor votar nos candidatos cadastrados no sistema, usando o mesmo sistema de votação das urnas eletrônicas brasileiras.



**Figura 4** - Urna eletrônica usada nas eleições brasileiras.

Fonte: <http://pt.wikipedia.org>

O sistema deve exibir uma interface gráfica que mostre um teclado numérico de 10 dígitos e 3 (três) botões - **BRANCO**, **CORRIGE** e **CONFIRMA** - semelhante ao da urna eletrônica (ver Figuras 4 e 5).



Figura 5 - Botões da urna eletrônica.

Fonte: <http://pt.wikipedia.org>

☞ **Atenção:** O mecanismo de votação do sistema deve ser exatamente o mesmo usado na urna eletrônica da Justiça Eleitoral. Para detalhes consulte a página do TSE, disponível em <http://www.tse.jus.br/eleicoes/eleicoes-anteriores/eleicoes-2010/eleicoes-2010/eleicoes-2010-simulacao-de-votacao>.

O eleitor digita o número do candidato e o sistema exibe o cargo ao qual o candidato está concorrendo, o número, o nome e o partido político do candidato. Além das mensagens “SEU VOTO PARA” antes da descrição do cargo público e “Aperte a Tecla: VERDE para CONFIRMAR e LARANJA para CORRIGIR”, conforme está apresentado na Figura 6.

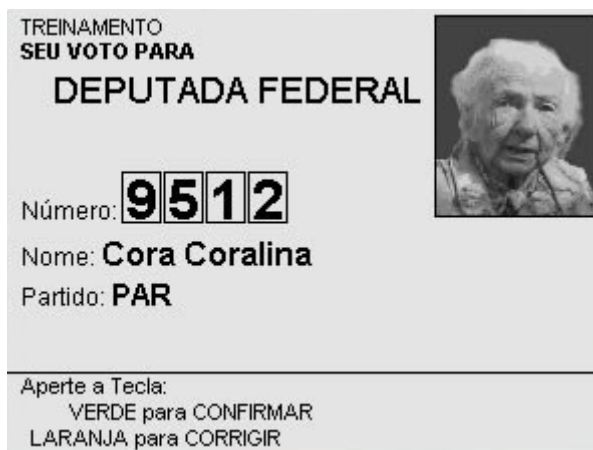


Figura 6 - Imagem apresentada pela urna eletrônica em uma sessão de treinamento de votação.

Após a confirmação o sistema deve concluir a votação exibindo a palavra “FIM”. Se o usuário desejar corrigir, ele deve pressionar o botão **CORRIGE** e digitar o número do candidato novamente. Para votar em branco o usuário pressiona o botão **BRANCO** e depois o botão **CONFIRMA**.

Na página do Tribunal Superior Eleitoral, existe um simulador de urna eletrônica disponível em <http://www.tse.jus.br/eleicoes/eleicoes-anteriores/eleicoes-2010/eleicoes-2010-simulacao-de-votacao/simulador-da-urna-eletronica-nacional-eleicao-2010>.

Este simulador foi desenvolvido em Java. Acessem para ver o funcionamento.

- **Apuração Eleitoral:** este módulo realiza a contagem dos votos. O resultado eleitoral deve ser apresentado em um gráfico semelhante ao da Figura 7.

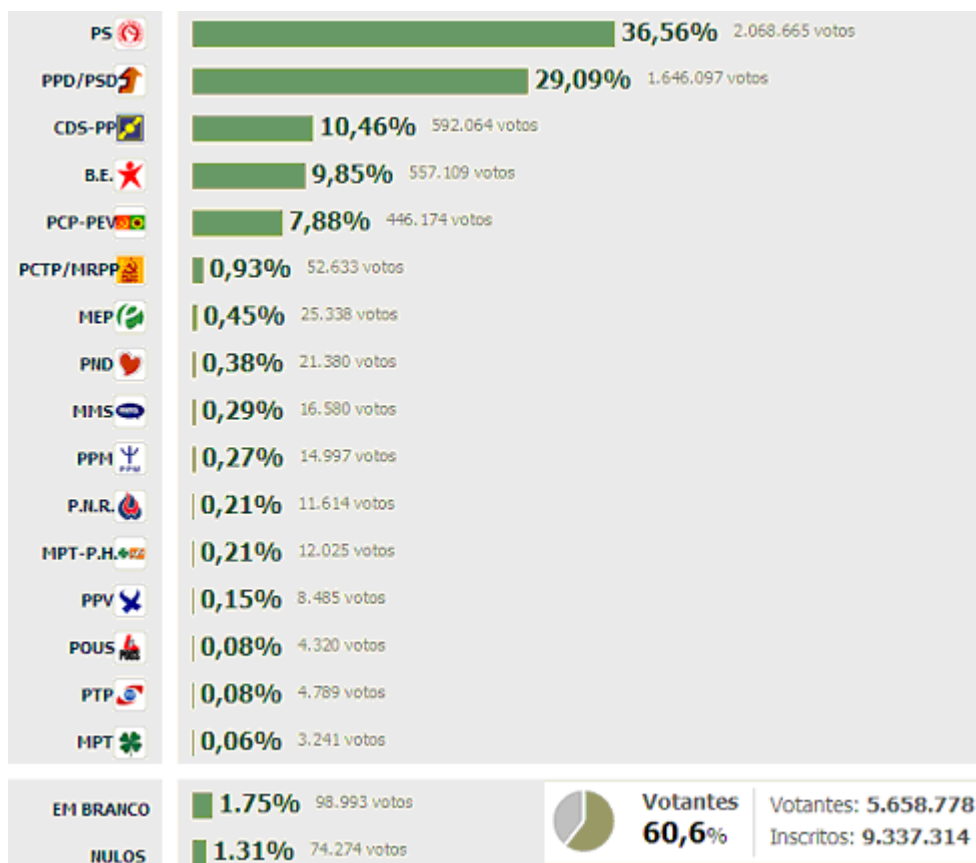


Figura 7 - Exemplo de gráfico com os resultados.

O gráfico deve apresentar os seguintes dados:

- a) o percentual de votos obtidos por cada candidato;
- b) o número de votos obtidos por cada candidato;
- c) o percentual de votos brancos e nulos;
- d) o número de votos brancos e nulos;
- e) o número de eleitores.

### • Critérios de Avaliação

1. Tratamento das situações de erro, validando todos os dados de entrada e os cálculos que possam abortar a execução do programa. Usar tratamento de exceção.
2. O trabalho será corrigido considerando:
  - 2.1 - a lógica empregada na solução do problema;
  - 2.2 - o funcionamento do programa;
  - 2.3 - a usabilidade<sup>1</sup> do programa e da interface gráfica;
  - 2.4 - o conhecimento da linguagem de programação;

<sup>1</sup> A usabilidade está diretamente ligada a interface e a capacidade do *software* em permitir que o usuário alcance suas metas de interação com o sistema. Ser de fácil aprendizagem, permitir uma utilização eficiente e apresentar poucos erros, são os aspectos fundamentais para a percepção da boa usabilidade por parte do usuário. Mas a usabilidade pode ainda estar relacionada com a facilidade de ser memorizada e ao nível de satisfação do usuário. **Fonte:** <http://pt.wikipedia.org/wiki/Usabilidade>

- 2.5 - o uso do paradigma de orientação a objetos.
3. Todo o código fonte deve estar legível, indentado, organizado e comentado.
  4. Todo o código fonte Java deve usar os comentários de documentação para a criação da documentação do sistema através da ferramenta *javadoc*.
  5. O trabalho pode ser desenvolvido em equipe composta por no máximo 2 (dois) alunos.

#### • Artefatos de *Software*

A equipe deve entregar os seguintes artefatos de *software*:

- a) o projeto Java desenvolvido na IDE Eclipse;
- b) o arquivo JAR da aplicação;
- c) a documentação HTML do sistema Java gerado com o *javadoc*.

#### • Data e Forma de Apresentação

- Quarta-feira, 4 de julho de 2012, segundo os horários definidos abaixo.

Equipe	Horário
1	7:30
2	8:00
3	8:30
4	9:00
5	9:30
6	10:00
7	10:30
8	11:00

- Entrevista com a equipe sobre o desenvolvimento e o funcionamento do sistema.
- Compactar todos os artefatos em um arquivo e nomeá-lo usando o nome da equipe, por exemplo, "Equipe 1.ZIP".

#### • Valor do Trabalho

25,0 (vinte e cinco) pontos.

Prof. Márton Oliveira da Silva  
[marlon.silva@ifsudestemg.edu.br](mailto:marlon.silva@ifsudestemg.edu.br)