



**Questão 1 (2,0 pontos):** Considere a seguinte situação: uma empresa de comércio eletrônico deseja implementar um sistema para armazenar informações sobre seus produtos. Cada produto possui um código único, nome, preço e quantidade em estoque. Além disso, a empresa deseja ser capaz de realizar operações de busca e inserção eficientes. Como professor na área de estrutura de dados:

- 1.1. Proponha uma estrutura de dados adequada para resolver esse problema e explique os motivos da sua escolha. **(0,5)**
- 1.2. Apresente também um exemplo de como seria a estrutura de dados implementada para armazenar cinco produtos diferentes. **(0,5)**
- 1.3. Apresente um código desenvolvido na linguagem C para a solução proposta. **(0,5)**
- 1.4. Apresente a complexidade ciclomática da proposta. **(0,5)**

**Questão 2 (2,0 pontos):** Suponha que uma empresa esteja planejando implementar uma rede local (LAN) em seu escritório principal, que abrigará 100 funcionários. Cada funcionário precisa de acesso à internet, compartilhamento de arquivos e impressão em uma impressora de rede centralizada. Além disso, a empresa deseja garantir um alto nível de segurança para proteger seus dados confidenciais.

- 2.1. Como professor na área de redes de computadores, descreva uma solução técnica para atender a esses requisitos, considerando aspectos como a topologia de rede, os dispositivos necessários, a configuração de segurança e a escolha de protocolos de comunicação. **(1,0)**
- 2.2. Explique as vantagens da solução proposta e identifique possíveis desafios de implementação, bem como as estratégias para superá-los. **(1,0)**

**Questão 3 (2,0 pontos):** Leia atentamente o minitexto a seguir e responda aos subitens 1.1, 1.2 e 1.3.

Usuários do serviço de streaming CEFETflix serão cadastrados na base de dados, devendo ser registrados seu e-mail principal (login), senha de acesso, nome, telefone, CPF, endereço de cobrança e número de cartão de crédito para pagamento mensal. A cada mês, a cobrança da mensalidade será emitida automaticamente, debitada do cartão de crédito do usuário registrado no cadastro, e confirmada ao usuário por e-mail. Caso não seja possível realizar a cobrança (por exemplo, por bloqueio do cartão, etc.) o usuário será avisado por e-mail e a mensalidade ficará pendente. Com duas mensalidades pendentes, o usuário perderá o acesso ao serviço CEFETflix. Ao acessar o serviço, o usuário poderá consultar o catálogo de vídeos. Vídeos podem ser de três categorias: filmes, séries e documentários. Sobre filmes, armazena-se o título, ano de produção, e duração em minutos. Sobre cada episódio de séries, são registrados dados sobre título, ano de produção, duração em minutos, temporada e número do episódio. Para cada episódio, é armazenada também a identificação do próximo episódio da mesma série, caso exista. Sobre documentários, armazena-se o título, ano de produção, duração em minutos e nome da



produtora. Cada usuário poderá avaliar o vídeo que assistiu, dando a ele uma nota entre 0 e 10. Apenas uma nota poderá ser dada por vídeo. Uma lista de atores será também mantida, e associada ao catálogo de vídeos, de modo que o usuário possa procurar por vídeos em que tenha atuado algum ator ou atriz de sua preferência. Sobre cada ator, será armazenado seu nome, data e local de nascimento.

**1.1.** Descreva em detalhes o modelo conceitual e apresente o MER (Modelo Entidade Relacionamento). **(1,0)**

**1.2.** A partir do modelo ER desenvolvido na questão anterior, faça a transcrição para o modelo lógico. Considere destacar as chaves primárias e estrangeiras sempre que necessário. **(0,5)**

**1.3.** Transcreva as declarações DDL/SQL (Data Definition Language / Structured Query Language) necessárias para criar o banco de dados baseado no modelo lógico proposto no item anterior. Considere que o SGBD é do tipo relacional e centralizado. **(0,5)**

**Questão 4 (2,0 pontos):** Após uma análise criteriosa de candidatos, o Instituto de Pesquisa DataCEFET selecionou você para desenvolver um sistema para informatizar o processo de pesquisa eleitoral do Instituto. Após a reunião de *Kick-off*, o gerente de projetos do instituto, entregou-lhe o escopo do projeto com os principais requisitos técnicos (não funcional) do sistema.

- A ferramenta de modelagem adotada para o sistema será a UML;
- O sistema deve ser desenvolvido em Java;
- O sistema deve rodar em plataforma operacional Windows em modo *desk top* em ambiente cliente/servidor;
- O banco de dados deve ser relacional e a ferramenta adotada deve ser o My SQL ou Postgree;
- Para separar a lógica de negócios (*Model*) da interface do usuário (*View*) e do fluxo da aplicação (*Controller*) será adotado a arquitetura MVC;
- O processo de desenvolvimento deve seguir a orientação do Processo Unificado (PU);
- Cada unidade do sistema deve ser devidamente testada.

Além dos requisitos técnicos, o escopo apresenta também um documento de visão do domínio do problema elaborado a partir de um levantamento inicial, de onde é possível identificar os principais requisitos funcionais do sistema.

O sistema deve possuir um banco de enquetes as quais devem ser utilizadas nas pesquisas. Existem dois tipos de enquetes: (1) Enquetes regionais; e (2) Enquetes nacionais. As enquetes regionais devem ser usadas em pesquisas locais/regionais e as enquetes nacionais devem ser usadas somente em pesquisas de âmbito nacional. O sistema deve ainda permitir cadastrar novas enquetes, consultar enquetes, excluir enquetes e cadastrar usuários/eleitor. Somente o coordenador do instituto pode abrir o sistema para votação, selecionar enquete para votação, cancelar votação e excluir enquete. O coordenador também pode realizar as funções comuns aos funcionários do instituto, como: cadastrar enquete, consultar enquete, gerar relatórios, cadastrar eleitor/usuário. Cada enquete, deve permitir que os usuários votem/escolha apenas



uma resposta dentro de um grupo de opções, com pelo menos duas opções (SIM ou NÃO, como exemplo). As escolhas dos usuários devem ser computadas em um relatório com:

- O voto de cada usuário bem como o total de votos por região e o total nacional.
- Os votos deverão ser contabilizados e exibidos de duas formas: (1) Tela com votos absolutos, que mostra os totais de votos para cada opção, por região e nacional; e (2) Tela com percentual de votos.
- Além da exibição em tela, o sistema deve permitir a extração de relatórios nos mesmos moldes dos resultados apresentados em tela.

Ainda na reunião de *Kick-off* o gerente de projetos solicitou a você que entregue no prazo máximo de 2 horas os artefatos enumerados nos subitens 4.1, 4.2 e 4.3, para que seja apresentado ao diretor do instituto e sua equipe, para que seja feita uma análise de viabilidade do projeto.

**4.1.** Identifique os principais requisitos funcionais do sistema e construa o Diagrama de Casos de Uso para o sistema de realização de enquetes. **(1,0)**

**4.2.** Construa o Diagrama de Classes para o sistema de realização de enquetes. Este modelo pode ser desenvolvido conceitualmente, ou seja, com as principais classes do sistema e seus métodos, mas sem a necessidade de listar os atributos destas classes. O DC deve apresentar as relações lógicas e emprego correto dos elementos notacionais da UML. **(0,5)**

**4.3.** Suponha que a arquitetura do sistema a ser desenvolvido utilizará o MVC (*Model, View, Controller*). Assim, construa um diagrama de sequência para o sistema de realização de enquetes com base no MVC. **(0,5)**

**Questão 5 (2,0 pontos):** Para o sistema de realização de enquetes modelado na **Questão 4**, desenvolva:

**5.1.** Um programa/classe em JAVA para o sistema de realização de enquetes. **(1,0)**

**5.2.** Uma classe de teste em JAVA (JUNIT) para testar unitariamente o programa/classe JAVA do sistema de realização de enquete construída no item 5.1. **(1,0)**



Prova Escrita (P. E.)

FOLHA DE RESPOSTA

**Questão 1 Subitem 1.1**

A estrutura de dados adequada para resolver o problema descrito é a árvore de busca binária.

Justificativa:

A árvore de busca binária é uma estrutura de dados que permite a busca eficiente de elementos, além de possibilitar a inserção ordenada dos mesmos. Essa estrutura é especialmente útil quando os elementos precisam ser mantidos em uma ordem específica, como é o caso da empresa de comércio eletrônico que deseja armazenar informações sobre produtos.

A árvore de busca binária é organizada de forma hierárquica, onde cada nó possui uma chave (no caso, o código do produto) e dois filhos (subárvores), que são menores ou maiores que a chave do nó pai. Dessa forma, a busca pode ser realizada de forma eficiente, percorrendo a árvore apenas pelo caminho necessário para encontrar o elemento desejado.

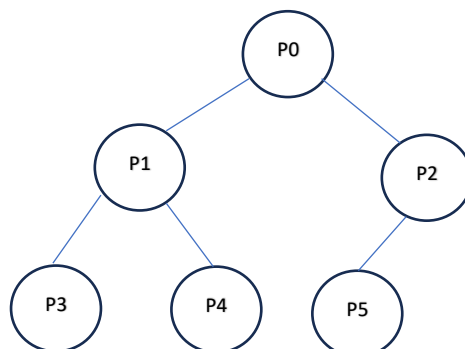
**Questão 1 Subitem 1.2**

Exemplo de implementação da árvore de busca binária para armazenar cinco produtos diferentes:

Considere os seguintes produtos:

Produto 1: Código: 12345 Nome: Camiseta Preço: R\$ 39,90 Quantidade em estoque: 10	Preço: R\$ 129,90 Quantidade em estoque: 7
Produto 2: Código: 98765 Nome: Calça Jeans Preço: R\$ 89,90 Quantidade em estoque: 5	Produto 4: Código: 13579 Nome: Óculos de Sol Preço: R\$ 59,90 Quantidade em estoque: 3
Produto 3: Código: 24680 Nome: Tênis	Produto 5: Código: 54321 Nome: Bolsa Preço: R\$ 79,90 Quantidade em estoque: 2

Exemplo de árvore de busca binária resultante:





Prova Escrita (P. E.)

FOLHA DE RESPOSTA

**Questão 1 Subitem 1.3**

```
// Definição da estrutura do nó da árvore
estrutura NoArvore:
    inteiro codigo
    texto nome
    decimal preco
    inteiro quantidade
    NoArvore esquerda
    NoArvore direita

// Função para inserir um produto na árvore
função inserirProduto(raiz, codigo, nome, preco,
quantidade):
    se raiz for nulo:
        novoNo <- criar nó com os valores passados
        retornar novoNo
    senão se codigo < raiz.codigo:
        raiz.esquerda <-
inserirProduto(raiz.esquerda, codigo, nome,
preco, quantidade)
    senão:
        raiz.direita <- inserirProduto(raiz.direita,
codigo, nome, preco, quantidade)
    retornar raiz

// Função para buscar um produto na árvore
função buscarProduto(raiz, codigo):
    se raiz for nulo ou raiz.codigo igual a codigo:
```

```
retornar raiz
se codigo < raiz.codigo:
    retornar buscarProduto(raiz.esquerda,
codigo)
senão:
    retornar buscarProduto(raiz.direita, codigo)
```

```
// Exemplo de inserção de produtos na árvore
raiz <- nulo
raiz <- inserirProduto(raiz, 12345, "Camiseta",
39.90, 10)
raiz <- inserirProduto(raiz, 98765, "Calça Jeans",
89.90, 5)
raiz <- inserirProduto(raiz, 24680, "Tênis",
129.90, 7)
raiz <- inserirProduto(raiz, 13579, "Óculos de
Sol", 59.90, 3)
raiz <- inserirProduto(raiz, 54321, "Bolsa", 79.90,
2)
```

```
// Exemplo de busca de um produto na árvore
produtoBuscado <- buscarProduto(raiz, 98765)
se produtoBuscado não for nulo:
    imprimir "Produto encontrado: " +
produtoBuscado.nome
senão:
    imprimir "Produto não encontrado"
```

**Questão 1 Subitem 1.4**

A complexidade ciclomática é:  $\log_2 n$



Prova Escrita (P. E.)

FOLHA DE RESPOSTA

### Questão 2 Subitem 2.1

Compreensão dos requisitos (20 pontos):

Compreende que os funcionários precisam de acesso à internet, compartilhamento de arquivos e impressão em uma impressora de rede centralizada.

Reconhece a importância da segurança para proteger os dados confidenciais da empresa.

Escolha da topologia de rede (20 pontos):

Propõe a utilização de uma topologia em estrela para conectar os 100 funcionários, com um switch centralizado.

Justifica a escolha, mencionando que a topologia em estrela oferece fácil gerenciamento, escalabilidade e bom desempenho.

Seleção de dispositivos e protocolos (25 pontos):

Sugere o uso de um switch de rede para a conexão dos funcionários, um roteador para o acesso à internet e uma impressora de rede centralizada.

Recomenda o uso do protocolo Ethernet para a comunicação local e o protocolo TCP/IP para a comunicação com a internet.

Explica que o uso desses dispositivos e protocolos oferece compatibilidade, desempenho adequado e uma infraestrutura padronizada.

Configuração de segurança (25 pontos):

Propõe a configuração de um firewall para proteger a rede local contra acessos indesejados.

Sugere a implementação de autenticação de usuários por meio de senhas ou outros mecanismos de identificação.

Destaca a importância do controle de acesso para restringir o compartilhamento de arquivos e o acesso à impressora somente aos funcionários autorizados.

Menciona a utilização de criptografia para garantir a confidencialidade dos dados transmitidos pela rede.

### Questão 2 Subitem 2.2

Análise de desafios e estratégias de superação (10 pontos):

Identifica o desafio de dimensionar adequadamente a capacidade da rede para atender às demandas dos 100 funcionários.

Sugere estratégias como o uso de switches gerenciáveis para monitorar o tráfego e o planejamento de uma largura de banda adequada para evitar congestionamentos.

Destaca a importância de considerar as restrições orçamentárias e o tempo necessário para a implementação da solução proposta.

Clareza e organização (10 pontos):

Apresenta a resposta de forma clara e organizada, com uso adequado da terminologia da área de redes de computadores.

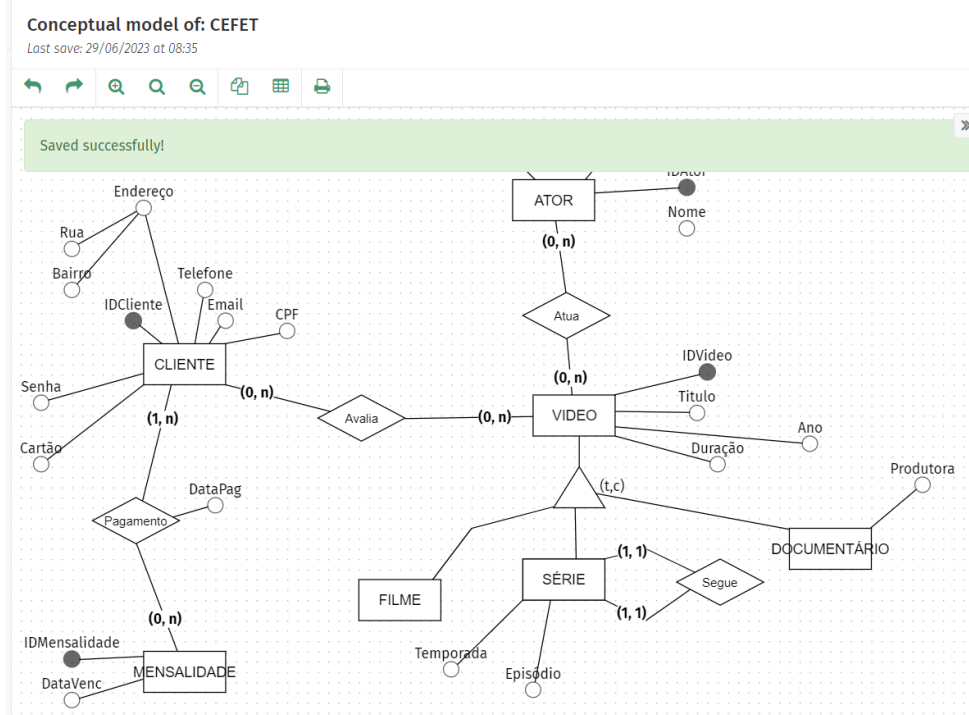
Demonstra habilidades de comunicação escrita, incluindo gramática, pontuação e estrutura de parágrafos.



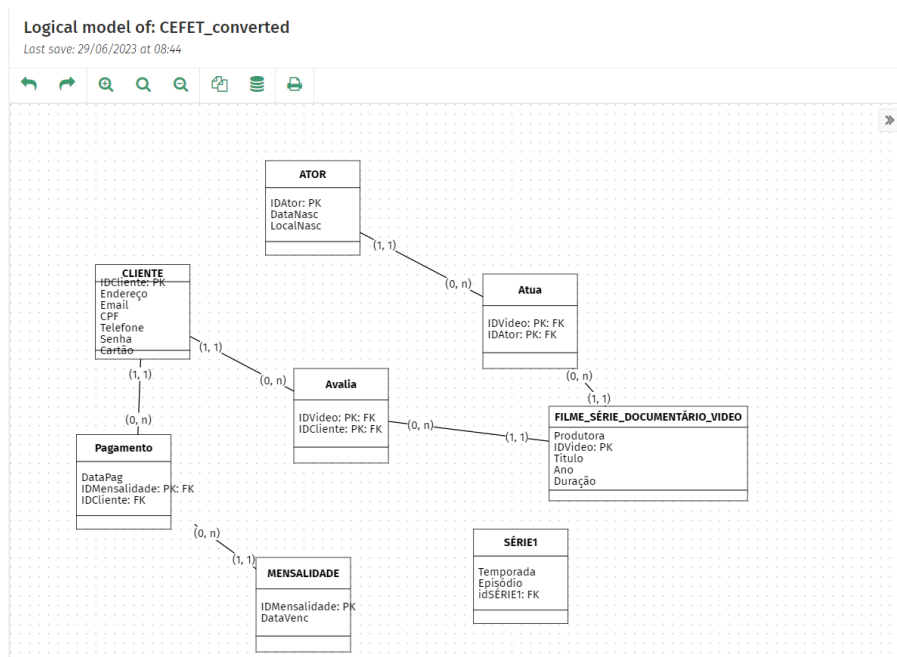
Prova Escrita (P. E.)

FOLHA DE RESPOSTA

Questão 3 Subitem 3.1



Questão 3 Subitem 3.2





Prova Escrita (P. E.)

FOLHA DE RESPOSTA

Questão 3 Subitem 3.3

```
CREATE TABLE ATOR
(
  IDAtor INT PRIMARY KEY,
  DataNasc INT,
  LocalNasc INT,
);
CREATE TABLE CLIENTE
(
  IDCliente INT PRIMARY KEY,
  Endereço INT,
  Email INT,
  CPF INT,
  Telefone INT,
  Senha INT,
  Cartão INT,
);
CREATE TABLE MENSALIDADE
(
  IDMensalidade INT PRIMARY KEY,
  DataVenc INT,
);
CREATE TABLE SÉRIE1
(
  Temporada INT,
  Episódio INT,
  idSÉRIE1 INT,
);
CREATE TABLE FILME_SÉRIE_DOCUMENTÁRIO_VIDEO
(
  Produtora INT,
  IDVideo INT PRIMARY KEY,
  Titulo INT,
  Ano INT,
  Duração INT,
);
CREATE TABLE Atua
(
  IDVideo INT PRIMARY KEY,
  IDAtor INT PRIMARY KEY,
);
CREATE TABLE Avalia
(
  IDVideo INT PRIMARY KEY,
  IDCliente INT PRIMARY KEY,
);
CREATE TABLE Pagamento
(
  DataPag INT,
  IDMensalidade INT PRIMARY KEY,
  IDCliente INT,
);

ALTER TABLE SÉRIE1 ADD FOREIGN KEY(idSÉRIE1) REFERENCES SÉRIE1 (idSÉRIE1)
ALTER TABLE Atua ADD FOREIGN KEY(IDVideo) REFERENCES FILME_SÉRIE_DOCUMENTÁRIO_VIDEO (IDVideo)
ALTER TABLE Atua ADD FOREIGN KEY(IDAtor) REFERENCES ATOR (IDAtor)
ALTER TABLE Avalia ADD FOREIGN KEY(IDVideo) REFERENCES FILME_SÉRIE_DOCUMENTÁRIO_VIDEO (IDVideo)
ALTER TABLE Avalia ADD FOREIGN KEY(IDCliente) REFERENCES CLIENTE (IDCliente)
ALTER TABLE Pagamento ADD FOREIGN KEY(IDMensalidade) REFERENCES MENSALIDADE (IDMensalidade)
ALTER TABLE Pagamento ADD FOREIGN KEY(IDCliente) REFERENCES CLIENTE (IDCliente)
```



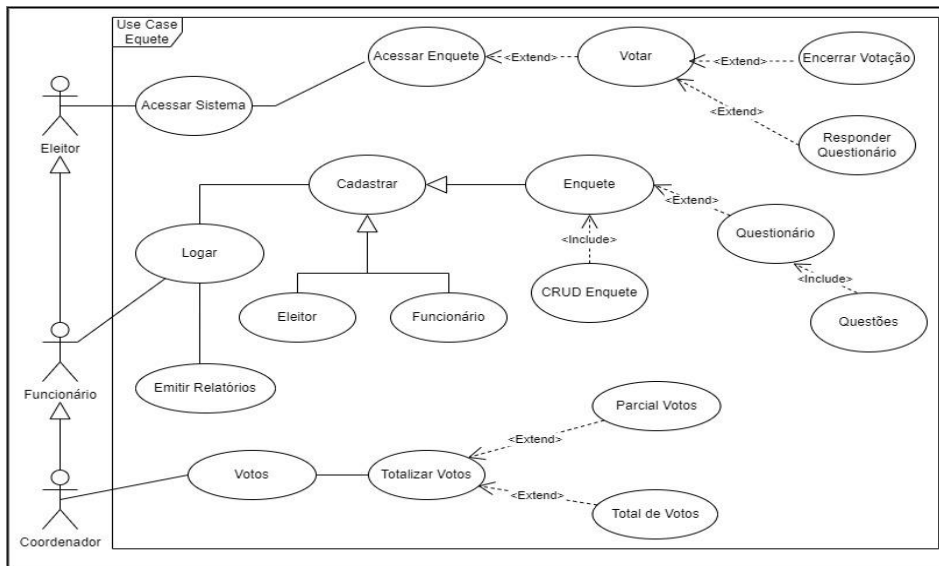


Prova Escrita (P. E.)

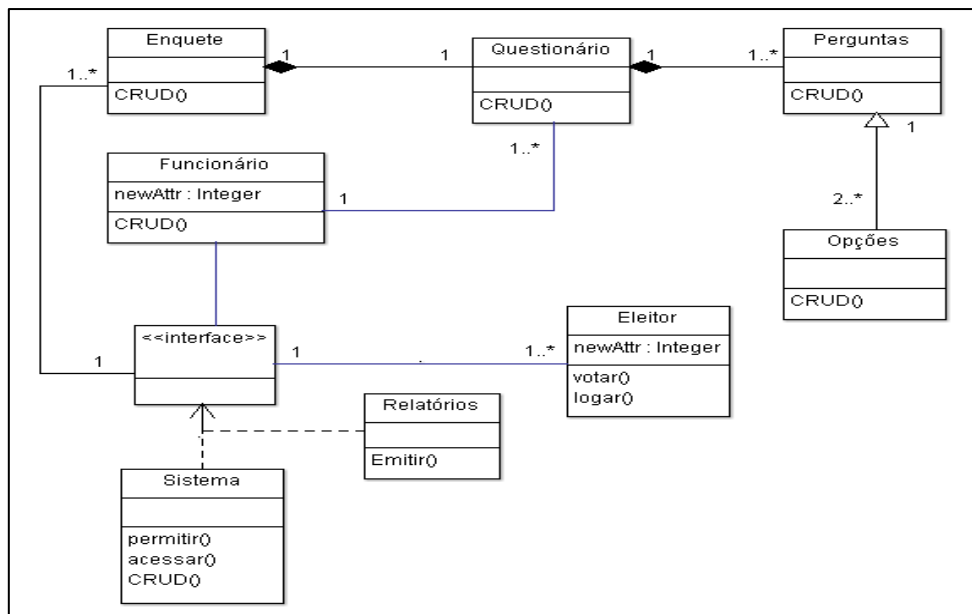
FOLHA DE RESPOSTA

**Questão 4 Subitem 4.1**

Lista dos Requisitos / Use Case do Sistema								
Funcionário	Enquete	Período	Questionário	Opções	Votos	Eleitor	Interface	Relatório
Gerar	Encerrar	Definir	Responder	Marcar	Totalizar	Acessar	Permitir	Emitir
CRUD	CRUD		CRUD	CRUD		Votar	Acessar	



**Questão 4 Subitem 4.2**

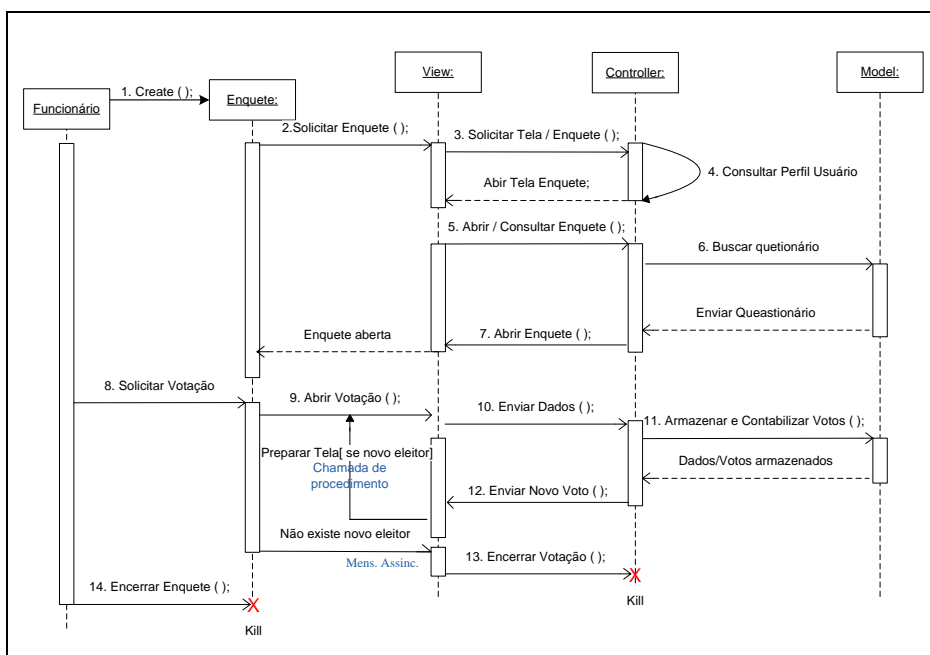
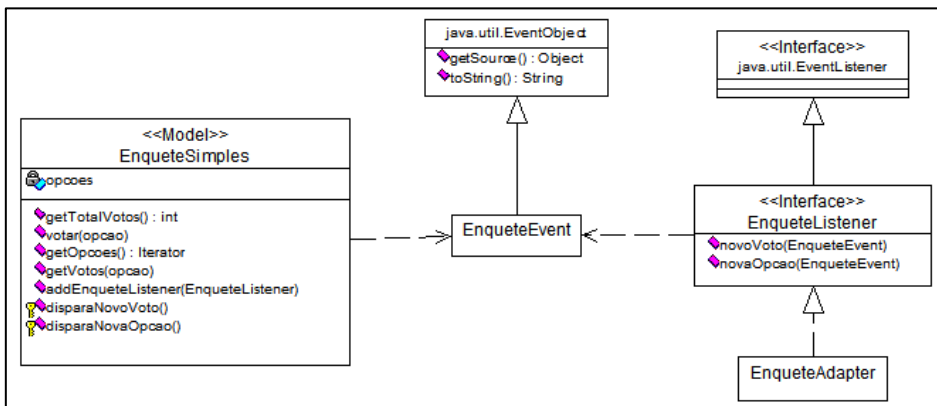




Prova Escrita (P. E.)

FOLHA DE RESPOSTA

Questão 4 Subitem 4.3





Prova Escrita (P. E.)

FOLHA DE RESPOSTA

Questão 5 Subitem 5.1

```
import java.util.List;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.Map;
import java.util.Set;

public class EnqueteSimples {
    private Map <String,Integer>opcoes;
    private List <EnqueteListener>enqueteListeners = new LinkedList();

    public EnqueteSimples(){
        opcoes = new HashMap<String, Integer>();
    }
    /**
     * Adiciona uma opção para ser votada na enquete
     * @param opcao nome da opção
     */
    public void addOpcao(String opcao){
        opcoes.put(opcao,new Integer(0));
        this.disparaNovaOpcao(opcao);
    }
    /**
     * Retorna um iterador de opções disponíveis na enquete
     * @return Iterator opções disponíveis na enquete
     */
    public Set <String> getOpcoes(){
        return opcoes.keySet();
    }
    /**
     * Incrementa um voto para opção
     * @param opcao opção que receberá voto
     */
    public void votar(String opcao){
        int votoAtual = (opcoes.get(opcao)).intValue();
        opcoes.put(opcao,new Integer(++votoAtual));
        this.disparaNovoVoto(opcao);
    }
    /**
     * Retorna a soma dos votos de todas as opções da enquete
     * @return int soma dos votos de todas as opções da enquete
     */
    public int getTotalVotos(){
        int total = 0;
        for(Integer votos : opcoes.values()){
            total+= votos.intValue();
        }
        return total;
    }
    /**
     * Retorna a quantidade de votos de uma opção individual
     * @param opcao opção que se quer o voto
     */
}
```



Prova Escrita (P. E.)

FOLHA DE RESPOSTA

```
* @return int quantidade de votos da opção
*/
public int getVotos(String opcao){
    return (opcoes.get(opcao)).intValue();
}
/**
 * Adiciona um EnqueteListener, um objeto interessado em
 * receber eventos lançados pela Enquete
 * @see EnqueteListener
 * @param listener objeto interessado em receber eventos
 */
public synchronized void addEnqueteListener(EnqueteListener listener){
    if(enqueteListeners.contains(listener)){ return; }
    this.enqueteListeners.add(listener);
}
/**
 * Informa aos objetos interessados nos eventos lançados
 * pela Enquete que um novo voto foi contabilizado.
 */
private synchronized void disparaNovoVoto(String opcao){
    for(EnqueteListener listeners : this.enqueteListeners){
        listeners.novoVoto(new EnqueteEvent(this,opcao));
    }
}
/**
 * Informa aos objetos interessados nos eventos lançados
 * pela Enquete que uma nova opção foi adicionada.
 */
private synchronized void disparaNovaOpcao(String opcao){
    for(EnqueteListener listeners : this.enqueteListeners){
        listeners.novaOpcao(new EnqueteEvent(this,opcao));
    }
}
}
```

**Questão 5 Subitem 5.2**

```
Public class TestaEnquete {
    Enquete enquete = new Enquete ( );
    Enquete enquetesimplesDAO = new EnqueteSimplesDAO ( );
    @test // alguns metodos a serem testados
    public void teste cadastrar ( ), test consultar ( ), test votar ( ) {
        enquete.cadastrar (“entradas de enquete”);
        enquete.consultar (“consulta de enquetes criadas”);
        enquete.votar (“testa entrada do voto, testa contagem do voto”);
        Enquete enquete = enqueteDAO.consultarEnquete (“Enquete”);
        Assert.assertEquals ( “Cadastro”, .getNome Enquete ());
        Assert.assertEquals ( “Cadastro”, .getVoto Enquete ());
        Assert.assertEquals ( “Cadastro”, .getDados Enquete ()); }
    ...
    @Test
```



Prova Escrita (P. E.)

---

FOLHA DE RESPOSTA

```
Public void Validar ( ) {  
    Enquete .setNome Enquete ( );  
    Enquete .setVoto Enquete ( );  
    Enquete .setDados Enquete ( );  
    }  
}
```



## Critérios Adotados na Correção

1. As questões foram corrigidas considerando os seguintes critérios: Corretude e Completude da resposta.
2. Para respostas completa e correta foram atribuídos os **2,0** , **somando** cada subitem da cada questão.
3. Resposta completa e parcialmente correta foi atribuído **1,5 pontos** somando cada subitem de cada questão.
  - a. Neste quesito o avaliador pode considerar, mediante interpretação, uma variação da nota, que pode chegar até **1,75 pontos**, somando cada subitem de cada questão.
4. Resposta incompleta e parcialmente correta foi atribuído **1,0** ponto somando cada subitem de cada questão.
  - a. Neste quesito o avaliador pode considerar mediante a interpretação da resposta, uma variação da nota com os seguintes valores: **0,25, 0,5, 0,75 e 1,0 pontos**, somando cada subitem de cada questão.
5. Resposta incorreta foram atribuídos **0,0 pontos**, independente da “completude” ou do tamanho da resposta
6. Questão não respondida: Atribuído 0,0 pontos.

### Observação importante:

1. As respostas do gabarito foram desenvolvidas pelo membro da banca que elaborou a questão.
2. Cada membro da banca elaborou pelo menos uma questão e desenvolveu a respostas (que está no gabarito).
3. As questões foram corrigidas pelo membro da banca que elaborou a questão, obedecendo os critérios previamente estabelecidos e descritos acima.