Curso: Engenharia de Software	Série: 6S	Turma: A[ ]/B[ ]/C[ ]	Turno:
Professor(a): João Choma Neto	Horário: 19:00 - 22:10		
Acadêmico (a):			RA:
Disciplina: PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE			Data:

Prova	Prova Prática	Atividades de estudo programadas (AEP)	Prova integrada	Nota final do bimestre
5,0	3,0	1,0	1,0	

INSTRUÇÕES PARA REALIZAÇÃO DA PROVA:

⇒ Os dados do cabeçalho deverão ser preenchidos com letra maiúscula. E as questões deverão ser respondidas com letra legível.

⇒ É vedado, durante a prova, o porte e/ou o uso de aparelhos sonoros, fonográficos, de comunicação ou de registro eletrônico ou não, tais como: notebooks, celulares, tablets e similares.

⇒ A prova é individual e sem consulta, deverá ser respondida a caneta azul ou preta. Prova escrita a lápis não dá direito à revisão. Não é permitido o uso de corretivo.

⇒ É obrigatória a permanência do acadêmico 1 (uma) hora em sala de aula após o início da prova.

⇒ Não será permitida a entrada na sala de aula após 10 minutos do início da prova.

⇒ É obrigatória a assinatura da lista de presença impressa na qual constam RA, nome e curso.

⇒ O valor de cada questão está ao lado da mesma.

⇒ Todas as respostas devem constar no espaço destinado e autorizado pelo professor, à resposta.

⇒ Em caso de qualquer irregularidade comunicar ao Professor ou fiscal de sala.

⇒ Ao término da prova, levante o braço e aguarde o atendimento do professor ou do fiscal.

1ºbim.	xxxx	2ºbim.		1ªsub.		3ºbim.		4ºbim.		2ªsub.	

Orientação do Professor:

Abaixo, você encontrará a tabela do Gabarito para resposta e inserção das alternativas das questões objetivas. Preencha esta tabela de acordo com a sua resposta.

GABARITO RESPOSTAS OBJETIVAS		
Q01	Q02	Q03

Questão 01 (0,1 pontos) Nos testes funcionais, o sistema é entendido com uma caixa-preta, a qual são fornecidas entradas ao sistema e avaliadas as saídas apresentadas. Entretanto, testar exaustivamente todas as entradas possíveis, pode ser uma tarefa impraticável, e por tal, técnicas de Particionamento do conjunto de entradas possíveis são utilizadas para viabilizar a atividade de teste (Pressman, 2011).

Considere, assim, uma função em um sistema de vendas online, que deve verificar a aplicação de um desconto de desconto, “verificaDesconto(valor, codigoPromo)” que deve retornar desconto válido apenas para valores maiores ou iguais a cem [≥ 100] e códigos de promoção válidos[valido]. O objetivo é testar a sensibilidade a combinações de entrada.

Com base na função acima descrita, **ANALISE** as combinações abaixo e **ASSINALE** aquela que melhor cobre testes de fronteira e combinação relevante para detectar falhas funcionais.

a) {valor=99, codigoPromo=”invalido”}
b) {valor=100, codigoPromo=”valido”}
c) {valor=150, codigoPromo=”invalido”}
d) {valor=100, codigoPromo=”invalido”}
d) {valor=120, codigoPromo=”valido”}

Questão 02 (0,1 pontos) Segundo o apresentado por Pressman (2011), qualquer produto de engenharia pode ser testado a partir de duas perspectivas diferentes. Uma aborda a lógica interna, e outra os requisitos e funcionalidades.

Em desenvolvimento, pretende-se distinguir testes que derivam das especificações (caixa-preta) daqueles que dependem da estrutura interna (caixa-branca). Sobre este assunto, **ANALISE** as afirmativas abaixo e em seguida **ASSINALE** a aquela que apresenta corretamente uma característica exclusiva (ou típica) do teste caixa-preta.

a) Avalia fluxo de controle interno e cobertura de branch.
b) Deriva casos de teste a partir da especificação e requisitos, sem exame do código.
c) Requer instrumentação do código para medir cobertura.

d) Serve apenas para testes unitários.
e) Substitui a necessidade de testes de integração

Questão 03 (0,1 pontos) O teste caixa-preta não é uma alternativa às técnicas caixa-branca, e vice-versa. Em vez disso, são abordagens complementares, com possibilidade de descobrir uma classe de erros diferente daquela obtida com métodos caixa-branca (Pressman, 2011).
Com isso podemos inferir as proposições apresentadas abaixo:

> **A**: "O teste caixa-preta detecta erros em interfaces e requisitos faltantes que muitas vezes escapam aos testes caixa-branca."
> **R**: "Porque o teste caixa-branca sempre proporciona cobertura completa do comportamento externo observado pelo usuário."

ANALISE as assertivas acima e **ASSINALE** a opção correta.
a) **A** verdadeira e **R** explica **A**.
b) **A** verdadeira e **R** não explica **A**.
c) **A** falsa e **R** verdadeira.
d) **A** e **R** são falsas.
e) **A** verdadeira e **R** irrelevante

Questão 04 (1,0 ponto) A atividade de teste, utiliza de uma série de artefatos para gerenciar e documentar o processo de teste de software. Tal documentação é essencial para a execução correta da atividade e para a rastreabilidade do software (Pressman, 2011).

Diante dessa característica a equipe de qualidade de software do TecTabajara uma empresa das Organizações Tabajara, organizou a documentação do EstoqueitorTabajara, um sistema revolucionário no processo de cadastro de dados falsos em organizações bancárias. Assim a aplicação tem como principais stakeholders, um grupo de bancários de reputação duvidosa.

Considere que você é parte do time Tabajara, e ficou responsável por elaborar dois casos de teste para orientar seus programadores.

Sendo assim **ELABORE** os dois casos de teste, conforme apresentado em aula, os casos de teste devem avaliar funcionalidades fundamentais do sistema.

Questão 05 (1,0 ponto) Testes de caixa branca são essenciais para garantir a que os caminhos percorridos na execução de um software são corretos. Neles a estrutura do presente programa pode ser avaliada através de uma série de técnicas e representações (Pressman, 2011)

Diante disso, considere que uma equipe quer garantir cobertura de decisões e cobertura de caminhos em um módulo crítico, que contém múltiplos if/else aninhados com estruturas de laço.

Sendo assim **DESENHE** um grafo de fluxo de controle para o seguinte código:

```
public class BubbleSort {
    public static void main(String[] args) {
        int[] arr = {5, 3, 1, 6, 2, 8, 4};
        boolean swapped;
        do {
            swapped = false;
            for (int i = 0; i < arr.length - 1; i++) {
                if (arr[i] > arr[i + 1]) {
                    // Troca os elementos de posição
                    int temp = arr[i];
                    arr[i] = arr[i + 1];
                    arr[i + 1] = temp;
                    swapped = true; // Indica que houve uma troca
                }
            }
        } while (swapped);
        // Imprime o array ordenado
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }
}
```

Questão 06 (0,5 ponto) A documentação do processo de teste de software fornece à pessoa do testador, uma base sólida para a realização dos testes. Está contida nesta documentação uma série de artefatos que descrevem detalhadamente os processos papeis e norteiam as atividades de teste daquele software (Pressman, 2011).

Dentre estes artefatos, encontramos, por exemplo, os Casos de Teste, que desenham detalhadamente o processo de teste a ser realizado em um software. Incluindo os dados e informações a serem testadas as saídas esperadas, e o procedimento a ser executado na execução do caso de teste.

A partir dos conhecimentos adquiridos, considere que a tabela abaixo, apresenta um exemplo de caso de teste para um software comercial de Frente de Caixa (PDV), com seus respectivos campos e informações.

Identificação	CT_001	
Itens a Testar	Caso de uso Abertura do Caixa	
Entradas	**Campo**	**Valor**
	Nome	Vendedor_A
	Login	VD_A
	Senha	ADM123
Saídas Esperadas	**Campo**	**Valor**
	Nome	Ademirson Luiz da Silva
	Grupo do Usuário	Vendedor
	Tela	Abertura de PDV
Ambiente	Banco de Teste	
Procedimento	Entrada de Usuário - ET-001-PT-01	
Dependência	Banco de dados vazio	

Outro artefato de suma importância, é o procedimento (Roteiro) de Teste, e que sempre aparece, e é referenciado na tabela de organização do caso de teste, estão conforme a tabela anterior “**Inclusão de Usuário - ET-001-PT-01”.**

Considerando o caso de teste apresentado, **PREENCHA** abaixo a tabela que descreve procedimentos de teste que serão usados para executar o caso de teste **CT_001.**

Identificação	ET-001-PT-01
Objetivo	
Requisitos	
Fluxo	

Questão 07 (1,0 ponto) Teste de caixa-preta ou testes funcionais são essenciais para encontrar falhas em categorias específicas no domínio das funcionalidades do software (Pressman, 2011).

A função de testador de software e líder de equipe de testes, você foi designado para elaborar um relatório a ser apresentado à gerência da empresa. Para cumprir essa responsabilidade com excelência, espera-se que demonstre domínio sólido dos conceitos fundamentais de teste de software, aplicando-os de forma estruturada e fundamentada.
Sendo assim, **DESCREVA** abaixo os conceitos de defeito, erro e falha, estabelecendo também a correlação existente entre eles.

Questão 08 (0,5 ponto) - Testes de caixa branca são essenciais para complementar lacunas não cobertas por testes de caixa preta. Essa nova perspectiva permite que o testador avalie o sistema de software por meio de critérios como McCabe (Pressman, 2011).

Imagine que você é um engenheiro de segurança encarregado de verificar a resistência de um cofre de alta segurança. Sua tarefa é determinar se o cofre é realmente à prova de arrombamento, de modo que nenhum invasor possa acessar seu conteúdo protegido. Para fazer isso, você não apenas deseja verificar a resistência da porta do cofre, mas também entender o funcionamento interno do mecanismo de travamento para identificar quaisquer pontos fracos que um invasor poderia explorar. Esse processo de avaliação minuciosa interna é semelhante ao que acontece no teste caixa branca. **EXPLIQUE** o teste caixa branca e **APRESENTE** um exemplo.

Questão 09 (0,4 ponto) - Os critérios de teste são essenciais para interpretação do sistema de software e para garantir que o sistema atende as expectativas do usuário. Uma forma de aplicação desses critérios é por meio do teste unitário (Pressman, 2011).

Para criar um teste unitário eficaz é muito importante fornecer um nível de cobertura adequado aos seus propósitos. Por exemplo, podemos testar todas as linhas de código, todas as classes ou todos os métodos. Além disso, na orientação a objeto ainda é necessário testar um objeto em todos os seus estados possíveis.

RELATE como um teste unitário deve ser implementado para que tenha sucesso no seu objetivo.

Questão 10 (0,3 ponto) Atualmente existem diversos frameworks de desenvolvimento de software, em um time ágil, documentos de teste (plano, casos, roteiro, relatórios) devem ser práticos, atualizáveis e rastreáveis para suportar entregas contínuas (Pressman, 2011). Como parte de um time de software, que aplica a metodologia ágil, você está responsável por construir os artefatos de teste de software para cada Sprint. Onde em cada entrega um plano de tese deve conter: (1) objetivo mensurável; (2) Abordagem (Estratégia) (3) responsabilidades; (4) ambiente de teste (5) Itens a testar (conjunto dos itens cobertos pelo plano); (6) Cronograma (Agenda) e Tarefas de Teste;

Para facilitar o processo descrito, **ELABORE** um **modelo enxuto de Plano de Teste** (máx. 1 página) para uma entrega incremental de um módulo de cadastro (funcionalidade básica).