

<b>Curso: ENGENHARIA DE SOFTWARE</b>		<b>Série: 7 S</b>	<b>Turma: [ ] A / [ ] B</b>	<b>Turno: NOTURNO</b>
<b>Professor(a): JOÃO CHOMA NETO</b>		<b>Horário: 19:00 – 22:00</b>		
<b>Acadêmico (a):</b>			<b>RA:</b>	
<b>Disciplina: ARQUITETURA DE SOFTWARE</b>			<b>Data:</b>	
<b>Prova</b>	<b>Prova Prática</b>	<b>Atividades de estudo programadas (AEP)</b>	<b>Prova integrada</b>	<b>Nota final do bimestre</b>
5,0	3,0	1,0	1,0	
<b>INSTRUÇÕES PARA REALIZAÇÃO DA PROVA:</b> <ul style="list-style-type: none"> <li>Os dados do cabeçalho deverão ser preenchidos com letra maiúscula. E as questões deverão ser respondidas com letra legível.</li> <li>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.</li> <li>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.</li> <li>É obrigatória a permanência do acadêmico 1 (uma) hora em sala de aula após o início da prova.</li> <li>Não será permitida a entrada na sala de aula após 10 minutos do início da prova.</li> <li>É obrigatória a assinatura da lista de presença impressa na qual constam RA, nome e curso.</li> <li>O valor de cada questão está ao lado da mesma.</li> <li>Todas as respostas devem constar no espaço destinado e autorizado pelo professor, à resposta.</li> <li>Em caso de qualquer irregularidade comunicar ao Professor ou fiscal de sala.</li> <li>Ao término da prova, levante o braço e aguarde atendimento do professor ou do fiscal.</li> </ul>				
1ºbim.	2ºbim.	1ªsub.	3ºbim.	4ºbim.
				2ªsub.

**Commented [A1]:** ATENÇÃO: todas as questões discursivas possuem a mesma pontuação. Considere as recomendações do Manual de Elaboração de Questões. As questões precisam ser de níveis diferentes: fácil, médio e difícil, bem com suas pontuações.

**QUESTÃO 01 – (0,3 pontos)** Em um mundo cada vez mais dependente de sistemas complexos de software para operações críticas em negócios, saúde, educação e além, a capacidade de criar sistemas robustos, eficientes e escaláveis é mais valorizada do que nunca. A arquitetura de software serve como o esqueleto ou a estrutura fundamental que orienta o desenvolvimento de sistemas, garantindo que eles possam crescer e evoluir sem perder eficiência ou se tornar impossíveis de manter. Esta estrutura não apenas influencia o desempenho do sistema, mas também afeta diretamente a capacidade de satisfazer requisitos funcionais e não funcionais, como segurança, usabilidade e confiabilidade.

**Commented [A2]:** Referencia?

Fonte: Elaboração Própria (ano).



Diante deste contexto, **ASSINALE** a alternativa que apresenta corretamente o uma dos benefícios gerados pela arquitetura de software.

- A) Aumentar a velocidade da internet usada pelo software.
- B) Definir as especificações de hardware necessárias para o software.
- C) Estabelecer a estrutura fundamental sobre a qual o software é organizado, facilitando a manutenção e a escalabilidade.
- D) Ajudar a reduzir o custo dos jogos de computador.
- E) Determinar a cor e o estilo da interface do usuário.

**QUESTÃO 02 – (0,3 pontos)** Considere o desenvolvimento de um aplicativo móvel que visa melhorar a experiência de compras em supermercados, oferecendo funcionalidades como listas de compras compartilháveis, informações nutricionais, sugestões de produtos com base em preferências pessoais e promoções em tempo real. Para esse projeto, a equipe de desenvolvimento enfrenta várias decisões críticas de projeto que afetarão a usabilidade, desempenho e escalabilidade do aplicativo. Escolher o banco de dados, o algoritmo de escalonamento, a disposição dos componentes do software são decisões críticas.

Fonte: Elaboração Própria (2025).

Com base neste contexto, **ASSINALE** a alternativa que representa uma decisão de projeto crítica que a equipe de desenvolvimento deve considerar para garantir o sucesso do aplicativo.

- A) Escolher a fonte dos textos no aplicativo para garantir legibilidade em dispositivos de baixa resolução.
- B) Determinar se o aplicativo deve ser desenvolvido nativamente para sistemas operacionais específicos ou se deve ser um aplicativo web responsivo.
- C) Decidir o tema musical que será tocado ao abrir o aplicativo.
- D) Selecionar o tipo de animações de transição entre as telas para maximizar a estética visual.
- E) Escolher a paleta de cores e o design dos ícones do aplicativo para atrair um público mais jovem.

[www.unicesumar.edu.br](http://www.unicesumar.edu.br)

Commented [A3]: Referencia?

Se for texto próprio, deve constar ao final do texto  
Fonte: Elaboração Própria (ano).

**QUESTÃO 03 – (0,4 pontos)** Considere o desenvolvimento de um sistema de gestão empresarial que precisa oferecer uma visão clara da estrutura organizacional, processos de negócios, e fluxo de dados entre diferentes departamentos. O uso de diferentes visões arquiteturais é crucial para abordar os variados interesses dos stakeholders, desde a equipe de desenvolvimento até os gerentes de negócios.

Fonte: Elaboração Própria (2025).

Commented [A4]: Referencia?

Se for texto próprio, deve constar ao final do texto  
Fonte: Elaboração Própria (ano).

Com base nisso, **INDIQUE** qual alternativa responde à seguinte pergunta: Qual das seguintes visões arquiteturais é mais relevante para entender e representar o fluxo de informações e a interação entre os diferentes sistemas e componentes em um sistema de gestão empresarial?

- A) Visão de Implementação: Detalha como o código é organizado no sistema de desenvolvimento.
- B) Visão de Desenvolvimento: Foca na organização do software no ambiente de desenvolvimento.
- C) Visão Física: Representa o sistema do ponto de vista do hardware ou topologia de rede.
- D) Visão Lógica: Mostra o sistema do ponto de vista dos usuários finais e como os recursos do sistema são concebidos por eles.
- E) Visão de Processo: Destaca as dinâmicas do sistema durante sua execução, incluindo processos e concorrência.

**QUESTÃO 04 e 05)** Considere que você é o arquiteto de software de uma equipe de desenvolvimento encarregada de construir um sistema de informação hospitalar que deve ser capaz de manejar um número crescente de registros de pacientes, interagir com outros sistemas hospitalares e garantir a privacidade e segurança dos dados dos pacientes. A escolha da arquitetura de software adequada para este sistema é crítica, pois influenciará todos os aspectos do desenvolvimento, desde a escolha de tecnologias até a facilidade de integração e manutenção futura. Dada a complexidade e as demandas críticas do projeto de sistema de informação hospitalar





descrito, é necessário que o arquiteto tome decisões assertivas para alcançar um produto de qualidade. Com base na problematização responda as questões 04 e 05.

Fonte: Elaboração Própria (ano).

**QUESTÃO 04 – (0,6 pontos) DESCREVA** quais seriam as principais considerações ao escolher a arquitetura de software para o sistema hospitalar.

---

---

---

---

---

---

---

---

---

---

---

**QUESTÃO 05 – (0,5 pontos) LISTE** os fatores que devem ser levados em conta para garantir a escalabilidade, a integração com outros sistemas e a segurança dos dados.

---

---

---

---

---

---

---

---

---

---

Commented [A5]: É recomendado separar as questões.

Referencia?

Se for texto próprio, deve constar ao final do texto  
Fonte: Elaboração Própria (ano).



**QUESTÃO 06 – (0,5 pontos)** No desenvolvimento de software, enfrentamos uma variedade de desafios que vão desde a organização do código até a otimização de processos e a implementação de funcionalidades complexas de maneira eficaz. Para superar esses desafios, os desenvolvedores recorrem a soluções comprovadas, conhecidas como padrões de projeto (design patterns). Estes padrões oferecem um esquema reutilizável para resolver problemas comuns no projeto de software. A importância desses padrões reside na sua capacidade de facilitar o desenvolvimento de software, tornando-o mais modular, flexível e adaptável a mudanças. Os padrões de projeto são tipicamente classificados em três categorias principais, Padrões Criacionais, Padrões Estruturais e Padrões Comportamentais.

Fonte: Elaboração Própria (ano).

Com base nos seus conhecimentos sobre padrões de projeto **EXPLIQUE** cada uma das três categorias citadas no texto.

IniCesumar

**Commented [A6]:** Referencia?



**QUESTÃO 07 – (0,6 pontos)** | Imaginemos o desenvolvimento de um aplicativo de processamento de imagens que ofereça várias operações, como ajuste de brilho, contraste, aplicação de filtros e redimensionamento. Cada uma dessas operações pode ser implementada de diferentes maneiras, dependendo do tipo de imagem, do resultado desejado ou da eficiência necessária. Além disso, novas operações ou métodos podem ser adicionados ao aplicativo no futuro sem alterar o código existente substancialmente. Este é um cenário ideal para aplicar o padrão de projeto Strategy. Ao utilizar o padrão Strategy, você pode definir uma interface comum para todas as operações de processamento de imagem, como `ImageOperation`.

Fonte: Elaboração Própria (ano).

**APONTE** como as suas decisões de implementação do padrão Strategy beneficiam a construção do aplicativo de processamento de imagens.

IniCesumar

**Commented [A7]:** Referencia?

Se for texto próprio, deve constar ao final do texto  
Fonte: Elaboração Própria (ano).



**QUESTÃO 08 – (0,5 pontos)** Padrões de projeto (ou *Design Patterns*) são soluções reutilizáveis para problemas recorrentes no desenvolvimento de software orientado a objetos. Eles não são pedaços de código prontos para copiar e colar, mas **modelos de solução**, que devem ser adaptados ao contexto específico de um sistema.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J.

***Design Patterns: Elements of Reusable Object-Oriented Software.***

Addison-Wesley. 1994

Considere o desenvolvimento de um sistema de gerenciamento de logs para uma aplicação web que precisa armazenar registros de eventos importantes, como erros, informações de depuração e alertas. Cada parte da aplicação pode precisar acessar o logger para registrar eventos, e é crucial que o arquivo de login se mantenha consistente ao longo do tempo.

Com base no seu conhecimento sobre design patterns no contexto apresentado acima (logs):

- (0,1 pontos) APRESENTE** qual design pattern poderia ser utilizado como solução para o problema dos logs.
- (0,4 pontos) DESCREVA** como seria a implementação desse design pattern.

---

---

---

---

---

---

---

---

---

---



**QUESTÃO 09 – (0,5 pontos)** Imaginemos o desenvolvimento de um aplicativo de comércio eletrônico robusto, com interfaces interativas para os usuários, processamento complexo de dados nos bastidores e a necessidade de apresentar informações atualizadas sobre produtos, preços e disponibilidade de estoque. À medida que o aplicativo cresce, sua base de código se torna cada vez mais complexa, exigindo uma estrutura que facilite a manutenção, a atualização e a expansão do código de maneira eficiente. Neste contexto, a aplicação do padrão de projeto Model-View-Controller (MVC) apresenta-se como a solução ideal. O padrão MVC divide o aplicativo em três componentes principais, facilitando a separação de responsabilidades e a gestão do desenvolvimento.

Fonte: Elaboração Própria (2025).

**EXPLIQUE** quais são os três componentes do padrão MVC.

**Commented [A8]:** Referencia?

Se for texto próprio, deve constar ao final do texto  
Fonte: Elaboração Própria (ano).



**QUESTÃO 10 – (0,8 pontos)** A utilização de padrões de projeto facilita o desenvolvimento de software, uma vez que esses padrões fornecem soluções para alguns problemas de design de software. Um desses padrões de projeto é o padrão de projeto *Factory Method*. Segundo Gamma et al. (1994), o *Factory Method* tem como principal objetivo “Definir uma interface para criar um objeto, mas permitir que as subclasses decidam qual classe instanciar. O *Factory Method* permite adiar a instanciação para as subclasses.”

Gamma, E., Helm, R., Johnson, R., & Vlissides, J.

***Design Patterns: Elements of Reusable Object-Oriented Software.***

Addison-Wesley. 1994

Com base no seu conhecimento sobre o padrão de projeto *Factory Method*:

- (0,1 pontos) LISTE** os elementos que compõem o padrão *Factory Method*.
- (0,5 pontos) DESCREVA**, detalhadamente, um exemplo de funcionalidade em que o padrão *Factory Method* poderia ser utilizado, justificando seu exemplo.
- (0,2 pontos) APRESENTE** quais classes deveriam ser criadas no seu exemplo para implementar cada elemento do padrão *Factory Method*.

[illegible]

**Commented [A9]:** Inserir linhas para resposta