

Atenção: Como a AD é individual, caso seja constatado que provas de alunos distintos são cópias uma das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final das respostas para as questões da prova tem que ser individual!

Os tópicos tratados como pesquisa na AD2 podem ser questionados na AP2 ou AP3!

1. Ao discutirmos os princípios de projeto de software, observamos um ciclo onde requisitos geram a demanda por mais software, que entra em operação, provoca mudanças no ambiente de trabalho, mudanças estas que acabam gerando novos requisitos. Explique porque este ciclo ocorre e como normalmente é encerrado. (Valor: 1,0 ponto; máximo: 10 linhas)
2. Quais são os componentes de um diagrama de sequência da UML? Mostre um diagrama de sequência e explique como estes componentes se relacionam e o quê cada um representa no diagrama. (Valor 1,0 ponto; máximo: 10 linhas)
3. O que representa o conceito de coesão em software. Que níveis de coesão podem ser observados no software? Quais os adequados? Como medir coesão? (Valor: 1,0 ponto; máximo 10 linhas)
4. Relacione as afirmações com os conceitos associados a elas (1.0 ponto):

(a) Estilos Arquiteturais	(1) Preocupações com o desenvolvimento de um produto de software, incluindo a concepção, implementação, entrega, uso e manutenção.
(b) Componentes	(2) Reduzir as consequências com eventos inesperados no desenvolvimento ou manutenção do software.
(c) Gerenciamento de Riscos	(3) Solução em alto nível de abstração envolvendo a definição da estrutura e do comportamento sistêmico do software.
(d) Casos de Uso	(4) Abstração correspondente a processos computacionais ou de estruturas de armazenamento.
(e) Ciclo de vida	(5) Explicita a interação a interação com o sistema por meio de cenários e passos bem definidos.

5. Atividade de pesquisa (não vale copiar e colar! Você deve pesquisar e explicar com suas palavras. Indique fontes alternativas que usar!): (valor 2,0 pontos)

O que diz o Código de ética do Profissional de Informática da Sociedade Brasileira de Computação (<http://www.sbc.org.br/institucional-3/codigo-de-etica>) ? Como ele se compara ao Código de ética e de Prática Profissional da Engenharia de Software, segundo as recomendações da *ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices* (<https://www.computer.org/cms/Computer.org/professional-education/pdf/doc.pdf>)? O que tem em um que não tem no outro?

6. Marque verdadeiro (V) ou falso (F) para as afirmações abaixo (valor até 1 ponto – 0,1 por marcação correta. Atenção: cada (2 erros) OR (2 brancos) OR (1 erro + 1 branco) eliminam 1 acerto!);

- a. Especificação de Requisitos incompleta representa uma das causas mais comuns de problemas em projetos de software.
- b. A deterioração da qualidade do software é consequência das alterações de hardware realizadas no ambiente de produção.
- c. Os defeitos que mais aparecem em especificações de requisitos são Informação Estranha e Fato Incorreto.
- d. Sistemas de software devem apresentar alto acoplamento e baixa coesão para reduzir a possibilidade de problemas em tempo de execução e facilitar a manutenção.
- e. O espectro de gerenciamento dos projetos de software deve envolver as pessoas, o problema e o processo.
- f. O custo para correção de um defeito muda muito pouco ao longo do ciclo de vida do software.
- g. Definição, Desenvolvimento e Manutenção representam as fases básicas da engenharia do software.
- h. O projeto é o processo criativo de transformar o problema em uma solução. Entretanto, a descrição de uma solução não pode ser chamada de projeto.
- i. Na orientação a objetos, a estrutura e o comportamento dos objetos estão descritos na representação da classe.
- j. O número de linhas de comunicação em um projeto depende do número de participantes na equipe. Uma equipe com 10 desenvolvedores implica em 52 linhas de comunicação que necessitam ser gerenciadas.

7. Uma abordagem orientada a objetos pode ser utilizada para desenvolver qualquer sistema? Quais são os “pontos fortes” da orientação a objetos? Quais são os “pontos fracos”? Dê um exemplo de um sistema em que a orientação a objetos não seria uma estratégia de desenvolvimento apropriada. (Valor até 1,0 ponto, máximo 15 linhas)

8. Explique a diferença entre teste funcional (caixa fechada), teste estrutural (caixa aberta) e teste baseado em erros, apresentando pelo menos dois exemplos de critérios que podem ser utilizados para projetar casos de teste em cada uma destas perspectivas. (Valor: 1,0 ponto)

9. Calcule a métrica Complexidade Ciclomática para o grafo de fluxo de controle abaixo (mostre como você calculou!) e indique como seu valor pode ser usado para apoiar a tomada de decisão no gerenciamento e desenvolvimento do projeto de software? (valor: 1,0 ponto)

(fonte: <https://guimaraesdani.wordpress.com/tecnicas-de-teste-parte-ii/>)

```
1. { int n, lastc, c; n = 1; lastc = getchar();  
2. while(lastc != ENDFILE)  
3.     {if ((c = getchar()) == ENDFILE) {  
4.         if ((n > 1 || (lastc == warning))  
5.             putrep (n, lastc);  
6.         Else putchar (lastc);  
7.     }  
8.     Else {if (c == lastc)  
9.         n+ +;  
10.        elseif ((n > 1) || (lastc == warning)) {  
11.            putrep (n, lastc) ; n = 1; }  
12.        Else putchar (lastc);  
13.    }  
14.    lastc = c;  
15. }  
16. }
```

