

Exercícios sobre Requisitos e Testes

1) Sobre **testes de software**, identifique verdadeiras (V) ou falsas (F) as seguintes afirmativas:

- () Um teste bem-sucedido identifica defeitos.
- () Casos de teste são especificações das entradas para o teste e da saída esperada do sistema.
- () Um teste caixa-preta é um teste estrutural, em que partes específicas de componentes são testadas.
- () Um teste de aceitação foca em cada unidade do software, ou seja, seu código-fonte.

Assinale a alternativa que apresenta a sequência correta, de cima para baixo.

- A) V – V – F – F B) F – F – V – V C) V – F – F – V
- D) F – V – V – F E) V – V – V – V

2) Qual das opções abaixo contém apenas **níveis de teste de software**?

- A) Teste de unidade, Teste de estresse, Teste de aceitação.
- B) Teste de integração, Teste de estresse, Teste de sistema.
- C) Teste de aceitação, Teste Chinês, Teste de caixa branca.
- D) Teste de unidade, Teste de integração, Teste de aceitação.
- E) Teste de sistema, Teste de caixa branca, Teste de caixa preta.

3) Usando o teste de particionamento de equivalência para um intervalo de número inteiros de 400 a 600 qual a melhor combinação de casos de testes:

- A) 400, 600, 399, 601 B) 400, 401, 600, 599
- C) 380, 650, 520 D) 380, 400, 600

4) A gestão de requisitos é um conjunto de atividades que tem como principal objetivo ajudar a equipe de projeto:

- A) utilizar ferramentas de engenharia de software para modelar os requisitos do sistema, através da UML.
- B) identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue.
- C) construir um modelo técnico refinado de funções, características e restrições do software.
- D) avaliar os requisitos quanto à qualidade, garantindo que ambiguidades, inconsistências, omissões e erros tenham sido detectados e corrigidos.

5) São cenários potencialmente dificultadores no processo de **engenharia de requisitos**:

- A) os usuários e desenvolvedores fazerem uso de vocabulários distintos e os usuários possuírem a mesma visão sobre todos os requisitos
- B) o usuário saber com precisão o que esperar do software e novos envolvidos poderem vir a participar do processo
- C) os requisitos não mudarem com o tempo e os usuários possuírem percepções conflitantes sobre os requisitos
- D) o escopo do software não poder ser definido e as mudanças nos requisitos ocorrerem diariamente.
- E) as mudanças nos requisitos possuírem frequência diária e os usuários possuírem a mesma visão sobre todos os requisitos.

6) Na análise e projeto orientado a objetos com **UML (Unified Modeling Language)** o programador necessita de um diagrama que apresente os requisitos funcionais do sistema. O diagrama desejado será o:

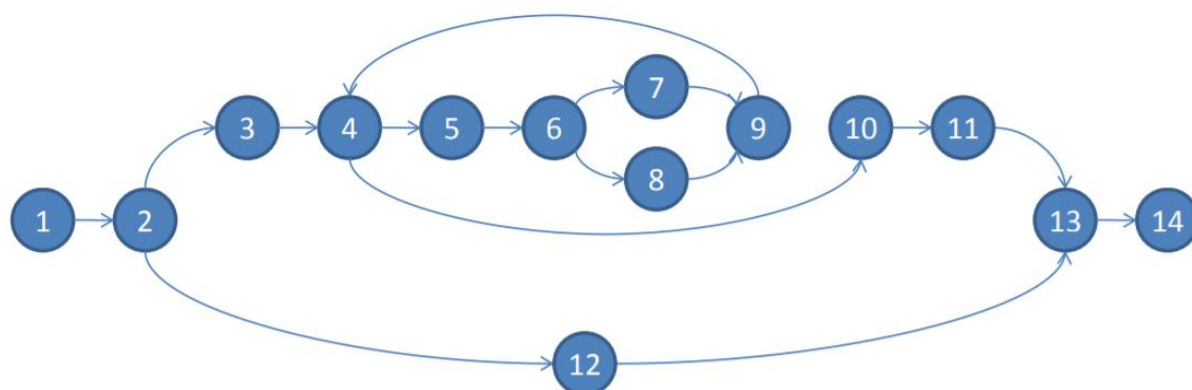
- A) Diagrama de Interação
- B) Diagrama de Caso de Uso
- C) Diagrama de Atividades
- D) Diagrama de Sequência
- E) Diagrama de Implantação

7) Você foi convidado a descrever requisitos de forma padronizada e adequada para um sistema web de locação de carros. Tomando como base o site apresentado pede-se:

- a) Dois requisitos funcionais
- b) Dois requisitos não funcionais de categorias diferentes
- c) Duas regras de negócios



8) Utilize a técnica do caminho básico para definir quantos caminhos devem ser testados para o código representado pelo grafo a seguir. Apresente claramente duas fórmulas de cálculo usadas.



9) Apresente um conjunto de casos de teste apropriado para os atributos sublinhados usando a técnica de particionamento de equivalência: Um sistema de avaliação nutricional de pessoas do sexo feminino recebe, como entradas: nome, endereço eletrônico, idade, peso e classe de renda. O nome é obrigatório e deve conter pelo menos um sobrenome, o endereço eletrônico é obrigatório e deve conter o símbolo @ antes do domínio. O sistema deve estar habilitado para aceitar apenas mulheres com idade entre 16 e 76 anos, com peso entre 42 e 110 quilos. As classes de renda possíveis são A – alta, M – média e B - baixa.

10) Em processos de desenvolvimento de software ágil, é comum a utilização de **Histórias de Usuário** e **Casos de Uso** para capturar requisitos e descrever funcionalidades do sistema. Considerando suas características e aplicabilidade, qual das seguintes afirmações melhor descreve a adoção apropriada desses elementos?

- A) Histórias de Usuário são detalhadas e técnicas, enquanto Casos de Uso são breves descrições do que um usuário espera do sistema
- B) Ambos, Histórias de Usuário e Casos de Uso, são destinados primariamente para desenvolvedores e servem para descrever como o código deve ser escrito
- C) Casos de Uso tendem a ser mais detalhados, descrevendo interações específicas com o sistema, enquanto Histórias de Usuário focam em necessidades do usuário e podem ser mais abstratas e concisas
- D) Histórias de Usuário e Casos de Uso são intercambiáveis e têm exatamente o mesmo propósito e nível de detalhe
- E) Histórias de Usuário são estruturas fixas que não podem ser modificadas após sua criação, enquanto Casos de Uso podem ser alterados frequentemente