

Atividade de Revisão - Testes de Software

Total de pontos **0/0**

O intuito dessa atividade é revisar o que foi visto em sala de aula, com base no material complementar disponibilizado no Classroom, assim como no que foi abordado e discutido em sala.

O e-mail do participante (**erick.vier@edu.unipar.br**) foi registrado durante o envio deste formulário.

✓ 1. O que é teste de software? *

- ☐ O processo de adicionar novas funcionalidades a um software para garantir que ele atenda aos requisitos dos usuários.
- ☐ Um processo de modificar o código-fonte de um software para eliminar bugs antes da entrega ao cliente.
- ☐ Um procedimento realizado apenas após a entrega do software ao cliente para corrigir possíveis problemas.
- ☐ A análise exclusiva de código-fonte para garantir que não há erros de sintaxe e o software funcione corretamente.
- ☒ Um conjunto de atividades que visa verificar e validar se o software funciona conforme o esperado, identificando erros ou comportamentos indesejados. ✓

✗ 2. Quais os objetivos do teste de software? *

- ☐ Desenvolver uma nova aplicação.
- ☐ Testar para que o software não precise mais receber manutenções e novas implementações.
- ☒ Garantir que o software funcione como esperado, e atenda aos requisitos definidos. ✓
- ☐ Corrigindo possíveis defeitos falhas ou comportamentos indesejados.
- ☐ Garantir que o software funcione somente em alguns navegadores ou sistemas operacionais.
- ☐ Garantir qualidade.

Resposta correta

- ☒ Garantir que o software funcione como esperado, e atenda aos requisitos definidos.
- ☒ Corrigindo possíveis defeitos falhas ou comportamentos indesejados.
- ☒ Garantir qualidade.

✓ 3. Sobre a repetibilidade em testes de software. *

- ☐ Os testes devem ser realizados apenas uma vez para economizar tempo.
- ☒ Testes bem definidos podem ser repetidos várias vezes com resultados consistentes. ✓
- ☐ Os testes devem ser automatizados para serem considerados repetíveis.
- ☐ A repetibilidade não é importante, pois cada teste é único.

✓ 4. Por que a documentação é fundamental nos testes de software? *

- ☐ Para que os testadores possam esquecer os testes realizados.
- ☒ Para rastreamento, auditoria e comunicação eficaz entre a equipe de desenvolvimento e os testadores. ✓
- ☐ Para aumentar a quantidade de trabalho realizado pelos testadores.
- ☐ Para evitar a necessidade de automatização dos testes.

✓ 5. Quais são as limitações no teste de software? *

- ☐ Não existe limitações, pois os testes de software realizam a cobertura total.
- ☒ Falsos positivos e negativos. ✓
- ☒ Dependência de dados. ✓
- ☐ O teste de software, economiza tempo e custos.
- ☐ Os teste de software são altamente eficazes e rápidos, pois não são complexos.
- ☒ Dependência de ambiente. ✓
- ☒ Mudanças no software. ✓

✓ 6. Qual das alternativas abaixo descreve corretamente o teste unitário? *

- ☐ Um teste realizado para avaliar o desempenho do software em um ambiente de produção.
- ☒ Um teste que verifica o comportamento de uma unidade específica do código, como uma função ou método, isoladamente. ✓
- ☐ Um teste que envolve a integração de múltiplos componentes do software para verificar se eles funcionam juntos.
- ☐ Um teste que se concentra exclusivamente na interface gráfica do usuário.
- ☐ Um teste que é feito apenas após o desenvolvimento completo do software.

✓ 7. Qual das alternativas abaixo descreve corretamente o teste de integração? *

- ☐ Um teste que verifica a funcionalidade de uma única unidade de código isoladamente.
- ☒ Um teste que avalia como diferentes módulos ou componentes do software interagem entre si. ✓
- ☐ Um teste focado exclusivamente na interface do usuário.
- ☐ Um teste realizado apenas após a entrega do software ao cliente.
- ☐ Um teste que é executado para medir o desempenho do software em ambientes de produção.

✓ 8. Qual das alternativas abaixo representa uma etapa fundamental do processo de teste de software? *

- ☐ Apenas a documentação dos requisitos do software.
- ☐ A realização de testes apenas após o software ser entregue ao cliente.
- ☒ O planejamento, que inclui a definição da estratégia, escopo e recursos necessários para os testes. ✓
- ☐ O desenvolvimento de novas funcionalidades sem considerar os testes.
- ☐ A execução de testes em um ambiente de produção sem preparação.

✓ 9. Qual das alternativas abaixo descreve corretamente o que o seguinte código em JUnit faz? *

```
@Test no usages
public void testAdd() {
    Calculator calculator = new Calculator();
    assertEquals(5, calculator.add(2, 3));
}
```

- ☐ O código testa se a classe Calculator tem um método chamado add.
- ☐ O código verifica se a soma de 2 e 3 na classe Calculator retorna 6.
- ☐ O código implementa a classe Calculator e o método add.
- ☒ O código testa se a soma de 2 e 3 na classe Calculator retorna 5, usando uma asserção. ✓
- ☐ O código compila a classe Calculator para uso em produção.

✓ 10. Interprete o código a seguir *

```
private UserService userService; 11 usages

@Before
public void setUp() {
    userService = new UserService();
}

@Test no usages
public void testCriarUsuario() {
    User usuario = userService.createUser("Alice");
    assertNotNull(usuario);
    assertEquals("Alice", usuario.getName());
    assertEquals(1, usuario.getId());
}

@Test no usages
public void testObterUsuarios() {
    userService.createUser("Alice");
    userService.createUser("Bob");

    List<User> usuarios = userService getUsers();
    assertEquals(2, usuarios.size());
}

@Test no usages
public void testAtualizarUsuario() {
    User usuario = userService.createUser("Alice");
    userService.updateUser(usuario.getId(), "Alice Atualizada");

    User usuarioAtualizado = userService.getUserById(usuario.getId());
    assertEquals("Alice Atualizada", usuarioAtualizado.getName());
}

@Test no usages
public void testExcluirUsuario() {
    User usuario = userService.createUser("Alice");
    assertTrue(userService.deleteUser(usuario.getId()));
    assertNull(userService.getUserById(usuario.getId()));
}
```

- ☒ O código verifica se a classe UserService consegue realizar operações básicas de CRUD (Criar, Ler, Atualizar e Excluir) em usuários. ✓
- ☐ O código apenas testa se os métodos da classe UserService lançam exceções.
- ☐ O código valida a ordem de criação dos usuários na lista.
- ☐ O código verifica se a classe UserService possui um método de busca por nome de usuário.
- ☐ O código está testando a integração entre a classe UserService e um banco de dados.