

LISTA DE EXERCÍCIO 2

1) Sabe-se que existem vários tipos de teste. Abaixo, quatro são listados:

- i) Teste de unidade
- ii) Teste de aceitação
- iii) Teste de estresse
- iv) Teste de integração

Relacionando esses tipos de teste com as abordagens caixa branca e caixa preta, marque a alternativa correta:

- a) Os tipos (i) e (ii) são caixa branca e (iii) e (iv) são caixa preta.
- b) Os tipos (ii) e (iii) são caixa branca e (i) e (iv) são caixa preta.
- c) Os tipos (i) e (iv) são caixa branca e (ii) e (iii) são caixa preta.
- d) Os tipos (iii) e (iv) são caixa branca e (i) e (ii) são caixa preta.
- e) Os tipos (i) e (iii) são caixa branca e (ii) e (iv) são caixa preta.

2) Sobre os testes de software, analise os itens a seguir:

- i) O processo de teste de software é definido como um processo separado, mas intimamente ligado, ao processo de desenvolvimento do sistema. Porém, eles têm metas e medidas de sucesso diferentes.
- ii) A taxa de defeitos de um software é calculada como sendo a divisão entre o número de casos de teste que falham e o total de casos de teste executados.
- iii) Quanto maior a taxa de defeitos de um software, menos sucedido é o processo de teste do respectivo software.
- iv) O documento de requisitos e os modelos de casos de uso servem de entradas para o processo de reportar o resultado de testes.

Quais alternativas são verdadeiras?

- a) Apenas I e II.
- b) Apenas III e IV.
- c) Apenas I e III.
- d) Apenas II e IV.
- e) Apenas I, II e III.

3) Entende-se como sendo _____ do sistema quando ele _____. O programador pode cometer _____ que geram _____ no sistema.

Qual das alternativas abaixo preenche correta e respectivamente as lacunas?

- a) bug; gera saídas incoerentes com as entradas; bug; falha.
- b) erro; trava; engano; entradas erradas.
- c) falha; erra; defeito; erro.
- d) bug; gera bug; falha; erro.
- e) erro; falha; engano; defeito.

4) O ciclo de vida de um defeito envolve os seguintes *stakeholders*:

- a) Líder da equipe de desenvolvimento do sistema, Testador, Grupo de usuários finais, Gerente de Projeto.
- b) Desenvolvedor, Testador, Analista de Requisitos, Cliente.
- c) Usuário final, Testador, Desenvolvedor, Líder de Projeto.
- d) Testador, Desenvolvedor, Analista de Teste, Gerente de Projeto.
- e) Gerente de Projeto, Desenvolvedor, Testador, Patrocinador do Projeto.

5) Leia o requisito de um sistema de e-commerce abaixo:

“Ao inserir um código de desconto, o sistema deve verificar se ele é válido e conceder no mínimo 5% e no máximo 20% de desconto no produto”

Marque a opção que descreve os melhores valores de desconto (em percentagem) para criação de casos de testes capazes de cobrir a maioria dos cenários possíveis do referido requisito:

- a) 1, 2, 3, 4 e 5 .
- b) 5 e 20.
- c) 4, 5, 10, 20 e 21.
- d) 19, 20 e 21.
- e) -1, 0, 5, 10 e 20.

Baseado nos códigos-fonte a seguir, responda as questões 6, 7 e 8:

| Classe Equacao2Grau : | Classe de teste da classe Equacao2Grau: |
|--|--|
| <pre>package projetoteste; public class Equacao2Grau { private double a, b, c; public void setA(double a) { this.a = a; } public void setB(double b) { this.b = b; } public void setC(double c) { this.c = c; } public double[] calcRaizes() { double raizes[] = new double[2]; raizes[0] = (-b + Math.sqrt(this.calcDelta())) / 2 * a; raizes[1] = (-b - Math.sqrt(this.calcDelta())) / 2 * a; return raizes; } public double calcDelta() { return b*b - (4 * a * c); } }</pre> | <pre>import org.junit.After; import org.junit.Before; import org.junit.Test; import static org.junit.Assert.*; import projetoteste.Equacao2Grau; public class Equacao2GrauTeste { Equacao2Grau eq1 = new Equacao2Grau(); public Equacao2GrauTeste() { } @Before public void setUp() { eq1.setA(1); eq1.setB(12); eq1.setC(-13); } @After public void tearDown() { } @Test public void testeDelta() { assertEquals(196.0, eq1.calcDelta(), 0.00001); } @Test public void testeRaizes() { double raizesCalculadas[] = eq1.calcRaizes(); assertEquals(1.0, raizesCalculadas[0], 0.00001); assertEquals(-13.0, raizesCalculadas[1], 0.00001); } }</pre> |

6) Sobre a classe de teste Equacao2GrauTeste, julgue os itens a seguir:

- i) Possui dois casos de testes diferentes.
- ii) O método assertEquals serve para verificar se o valor calculado pela classe é igual ao valor esperado no caso de teste. Se forem iguais, o teste é aprovado (verde); senão, o teste falha (vermelho).
- iii) Utiliza o framework de teste gratuito do Java conhecido como JUnit e é um exemplo de teste unitário automatizado.

É correto o que se afirma nos itens:

- a) I apenas.
- b) III apenas.
- c) III apenas.
- d) I e III apenas.
- e) I, II e III.

7) Quando a classe _____ for executada, o caso de teste testeDelta vai _____ e o testeRaizes vai _____.

Marque a alternativa que preenche as lacunas acima de maneira correta e em ordem.

- a) Equacao2GrauTeste, passar, passar.
- b) Equacao2GrauTeste, falhar, passar.
- c) Equacao2Grau, falhar, falhar.
- d) Equacao2Grau, passar, passar.
- e) Equacao2Grau, falhar, passar.

8) Sobre as duas classes, julgue os itens abaixo como verdadeiro ou falso:

- () O comando `return b*b - (4 * a * c);` pode ser substituído por `return Math.pow(b,2) - (4 * a * c);`
- () O método `assertEquals(196.0, eq1.calcDelta(), 0.00001);` pode ser substituído por `assertEquals(196.0, eq1.calcDelta());`
- () Os métodos setA, setB e setC não precisam ser testados, porque, no contexto da matemática, podem receber qualquer valor real (negativo, positivo ou zero).

Qual alternativa representa a sequência correta de V para verdadeiro e F para falso?

- a) V-V-F.
- b) V-F-V.
- c) F-V-F.
- d) V-V-V.
- e) F-F-F.

9) Existem vários tipos de testes de software, um deles é o unitário. Quando eles são automatizados é possível

- a) testar mais lentamente o sistema.
- b) verificar o software como um todo.
- c) estimar se os requisitos não funcionais serão atendidos.
- d) garantir a alta qualidade do sistema.
- e) mostrar o código dos testes para o usuário com o objetivo de explicar os casos de teste.

10) Associe a ferramenta da esquerda com o descrição na direita tendo em vista o processo de teste de software.

| Ferramentas: | Descrição: |
|--------------|--|
| I. JMeter | () Permite reduzir artificialmente as configurações do computador para testar o sistema em ambiente com recursos reduzidos. |
| II. JUnit | () Armazena o código-fonte do projeto, sendo possível controlar as versões e a relação de bugs do mesmo. |
| III. Github | () Largamente utilizado para testes unitários de métodos das classes programadas na linguagem Java. |
| IV. Excel | () Software que serve para realizar testes de performance em aplicações web. |
| V. WinStress | () Ferramenta muito usada para projetar e controlar os casos de testes em nível de projeto. |

A ordem da associação correta é

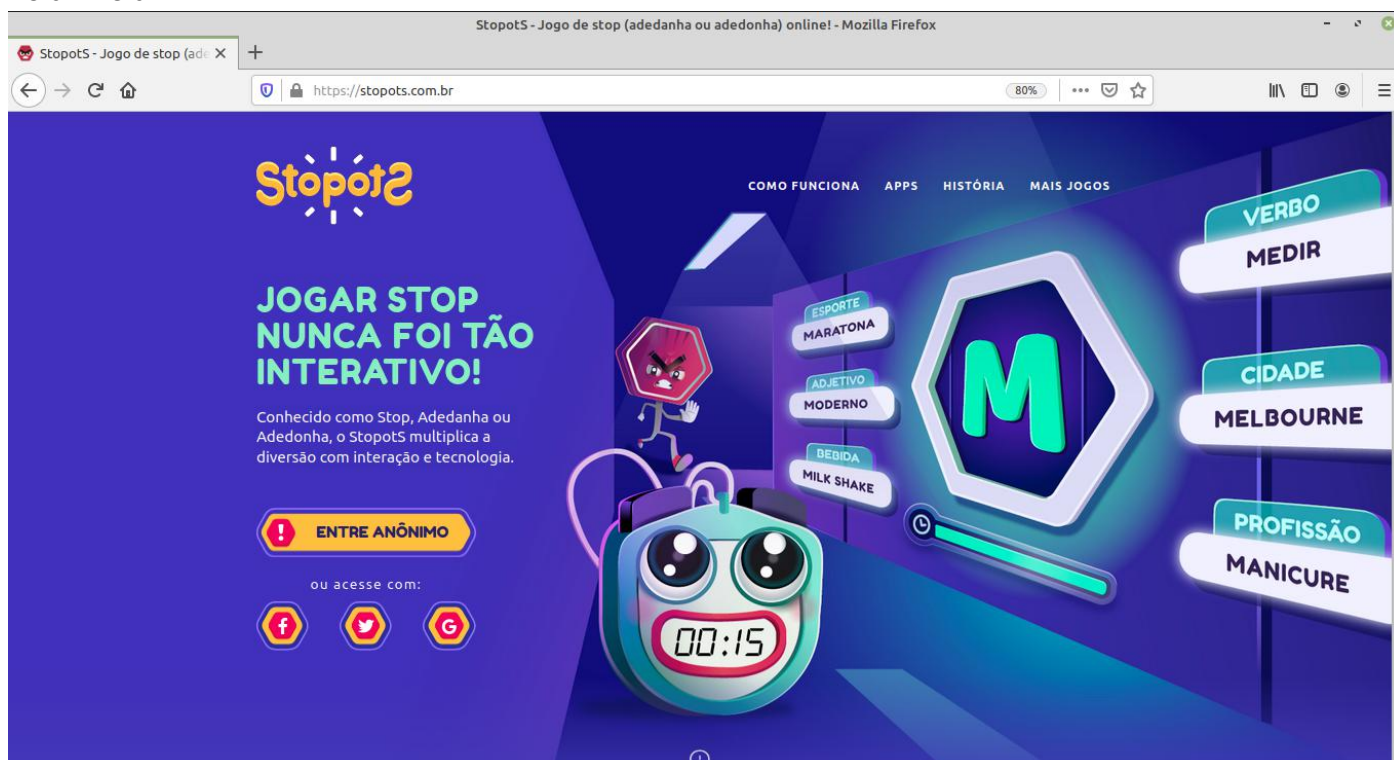
- a) III, I, II, V, IV.
- b) V, III, II, IV, I.
- c) I, III, II, V, IV.
- d) I, II, III, IV, V.
- e) V, III, II, I, IV.

11) Explique como o processo de gerenciamento de erros pode ser realizado em um software de grande porte e complexo.

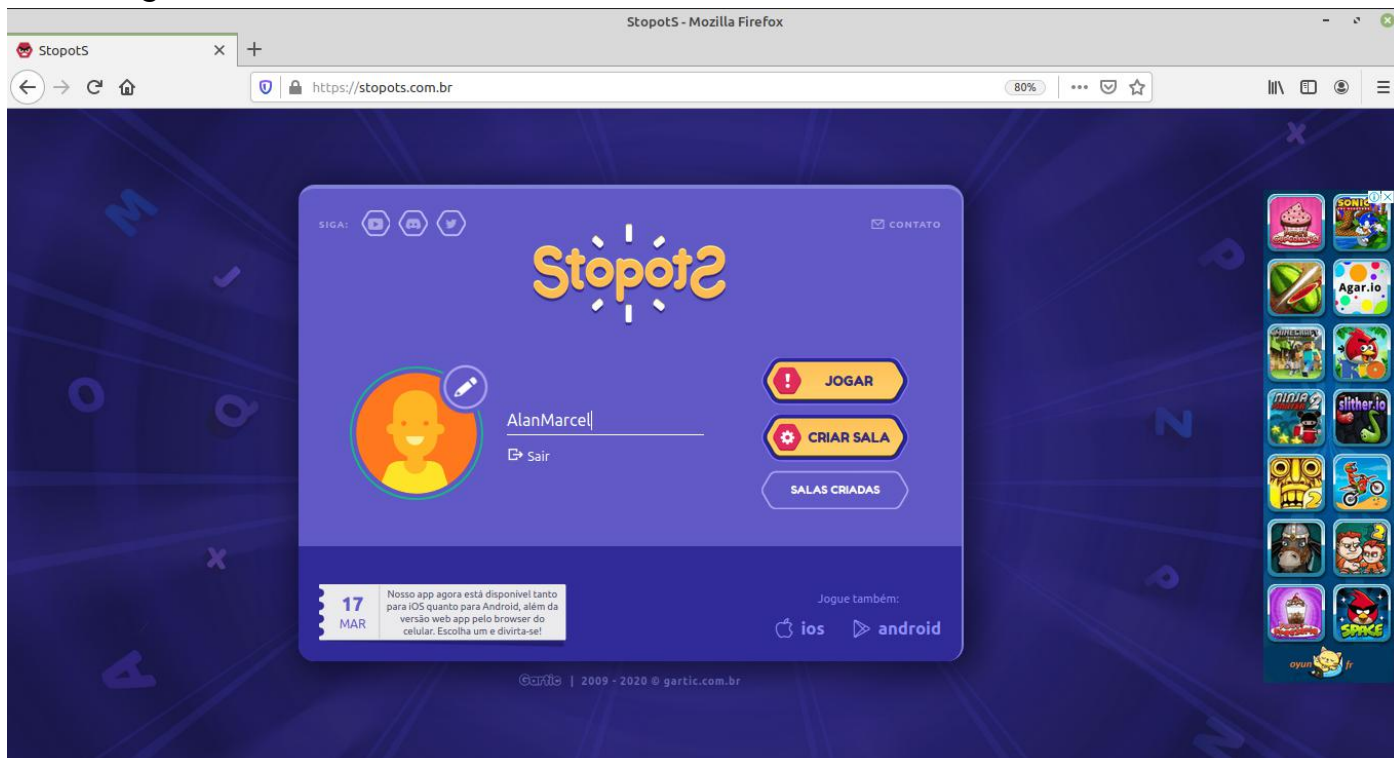
12) Analise as telas do software <https://stopots.com.br/>, que é uma versão digital do “Jogo da Adedonha”, jogado, antigamente, com papel e caneta. Em seguida, cite e descreva:

- a) três casos de testes unitários que podem ser realizados no software;
- b) dois testes de performance;
- c) dois testes de integridade de dados.

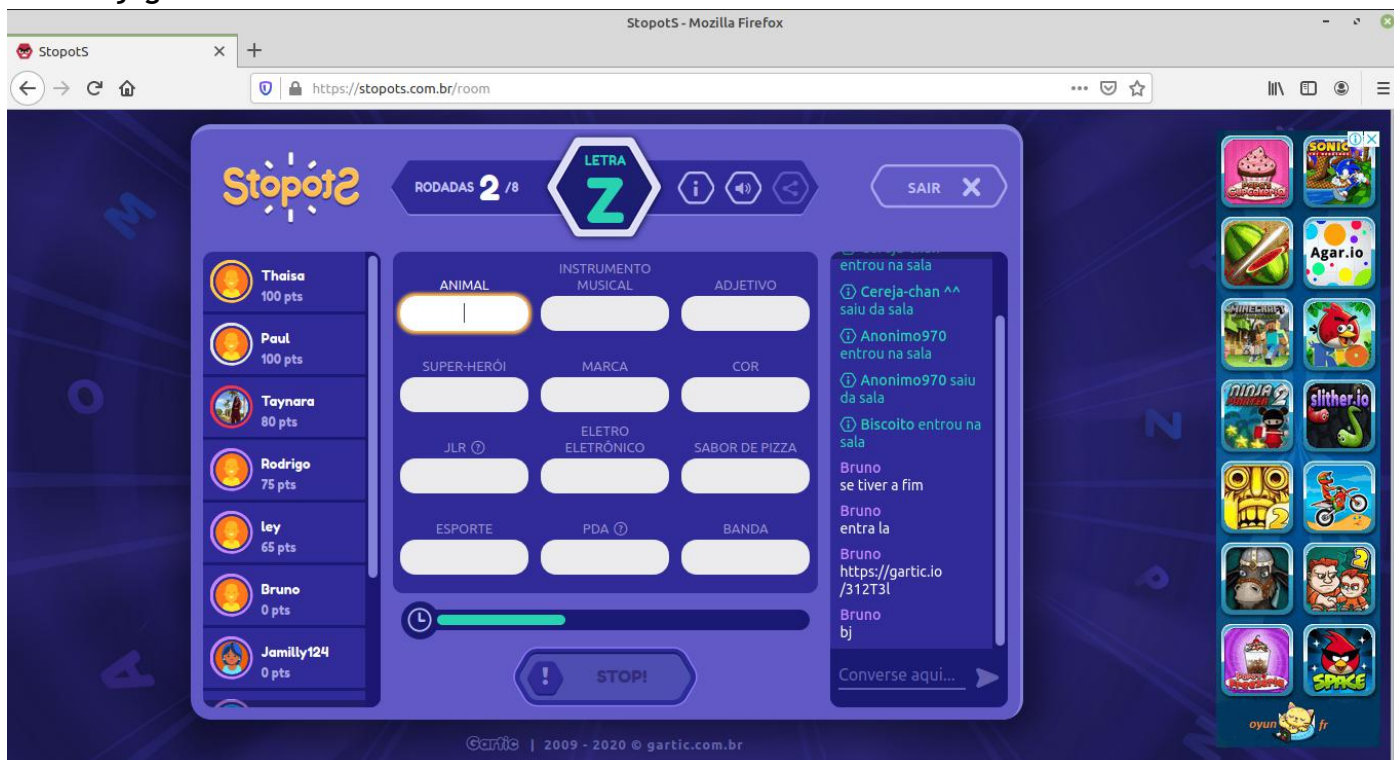
Tela inicial:



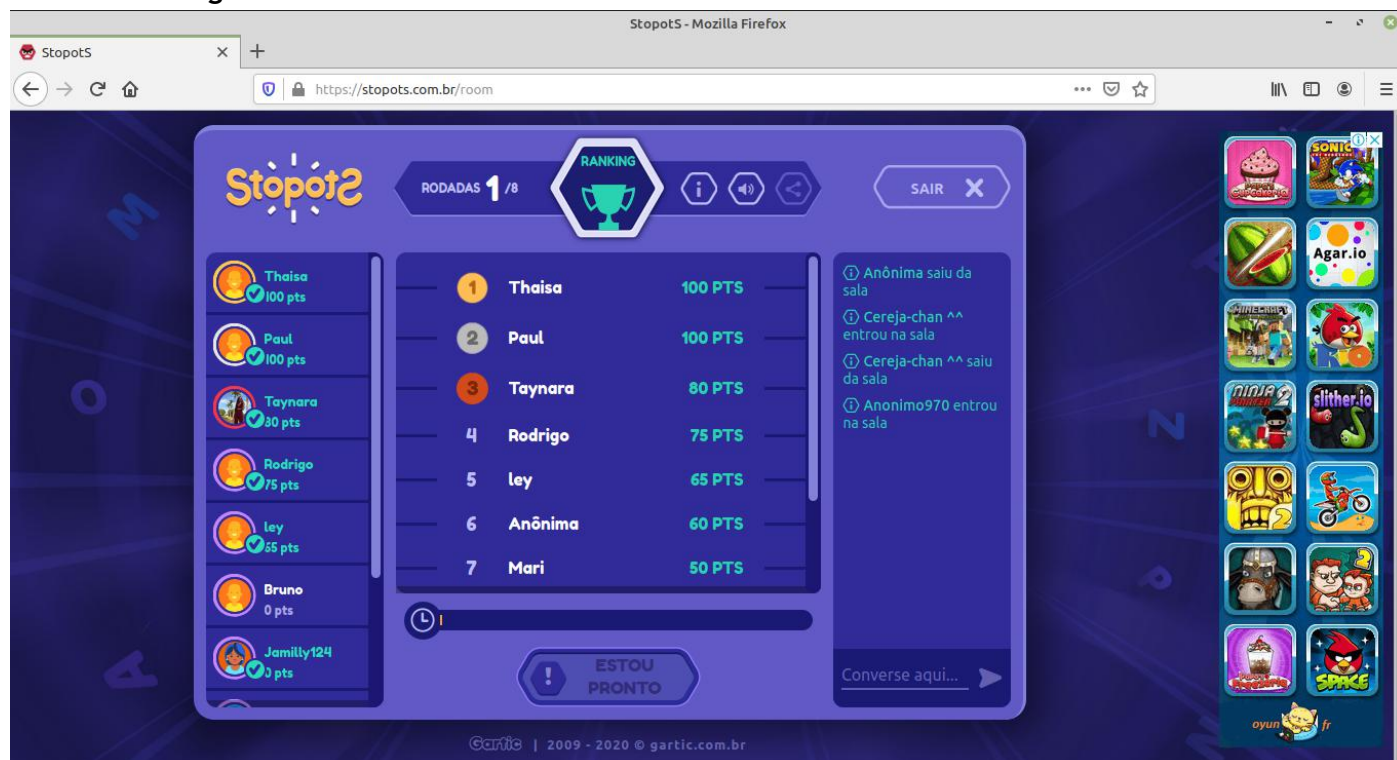
Tela de login:



Tela do jogo:



Tela de ranking:



Tela de criação de sala:

