



Planejamento - Teste de navegabilidade da página de login do site Game Mania

Autor: Tiago de Almeida Silva

➤ Plano de Testes

Modelo de teste cuidadosamente desenvolvido para o site Game Mania, contendo objetivo, escopo, elementos que serão testados, cronograma das atividades, a técnica e os recursos que serão utilizados, o responsável pelo monitoramento, as referências e as normas utilizadas.

Seção Plano de teste	Descrição da seção	Modelo básico de plano de teste	Modelo padrão de plano de teste	Modelo de plano de teste de software	Modelo de plano de teste agile	Modelo de plano de teste de sistema	Modelo de plano de teste de segurança do aplicativo da web	Modelo de plano de teste de software integrado	Modelo de plano de teste clássico
Resumo	Testar as funcionalidades da página de Login do site Game Mania.	X	X	X	X	X	X	X	X
Escopo do Plano de Teste	Para o teste foi utilizado a IDE Eclipse e a linguagem JAVA.			X					
Objetivos de Negócios	Melhorar a satisfação e experiência dos usuários.		X	X		X	X	X	X
Objetivos do Teste	Avaliar o correto funcionamento do login e verificar se há qualquer erro ou inconsistência no mesmo.		X	X		X	X	X	X
Documentos Normativos e Informativos	Normas e diretrizes da empresa foram seguidos, assim como as da ABNT.			X		X		X	

Seção Plano de teste	Descrição da seção	Modelo básico de plano de teste	Modelo padrão de plano de teste	Modelo de plano de teste de software	Modelo de plano de teste agile	Modelo de plano de teste de sistema	Modelo de plano de teste de segurança do aplicativo da web	Modelo de plano de teste de software integrado	Modelo de plano de teste clássico
Revisão Formal	Uma comissão de revisão será criada para manter a qualidade e eficiência do processo.		X	X		X	X	X	X
Requisitos	Uso dos softwares e ferramentas necessárias para os testes.	X		X		X	X	X	X
Links de Coleção de Requisitos	Não há links de coleção.		X	X	X	X		X	
Links de Plano de Desenvolvimento	Não há links de plano de desenvolvimento.		X		X				X
Avaliação de Risco	Riscos de sobrecarregamento da página e SQL Injection.		X	X		X		X	X
Planejamentos de Teste	O teste segue um rígido cronograma.	X	X	X		X	X	X	X
Estimativa de Teste	No mínimo um teste será realizado		X	X		X	X	X	X

Seção Plano de teste	Descrição da seção	Modelo básico de plano de teste	Modelo padrão de plano de teste	Modelo de plano de teste de software	Modelo de plano de teste agile	Modelo de plano de teste de sistema	Modelo de plano de teste de segurança do aplicativo da web	Modelo de plano de teste de software integrado	Modelo de plano de teste clássico
	para ao menos 30 usuários.								
Ambientes de Teste	Teste em ambiente web.		X	X	X	X	X	X	X
Detalhes do Ambiente de Teste de Software	Será usado a linguagem Java com o IDE do eclipse.			X					
Estratégia de Amostragem	Será testado centenas de senhas e logins para 30 usuários cadastrados.					X			
Instalação de Testes	Os testes serão instalados 5 dias antes do início dos mesmos.							X	
Condições de Teste.	Teste realizado em ambiente interno, atentando para as normas de segurança do trabalho.					X			

Seção Plano de teste	Descrição da seção	Modelo básico de plano de teste	Modelo padrão de plano de teste	Modelo de plano de teste de software	Modelo de plano de teste agile	Modelo de plano de teste de sistema	Modelo de plano de teste de segurança do aplicativo da web	Modelo de plano de teste de software integrado	Modelo de plano de teste clássico
Dados de Teste	Os dados serão coletados para a formulação do relatório final.							X	
Identificação do Teste	Teste realizado em fases.			X					
Estratégia de Teste	Teste realizado em fases.							X	
Equipe de Teste	Teste realizado pelo desenvolvedor: Tiago de Almeida Silva.		X	X		X	X	X	X
Instrumentos e Equipamento de Teste	Computadores e softwares de teste.					X			
Segurança do Aplicativo	Firewall e antivírus.						X		
Objetivos de Qualidade	Os objetivos de qualidade para uma liberação, no formato de tabela.		X	X		X	X	X	X

[illegible]

➤ Caso de Testes

Definição das unidades do site Game Mania que serão testadas, os dados de entrada, resultados e os fluxos básicos e alternativos de uma ação do usuário.

Definição da tabela de casos de teste:

Cenário	Caso de Teste	Entradas	Resultados Esperados
Cenário 1 (e-mail correto)	Entrar na página de Login	Verificar área de Login	Funcionalidades do Login funcionando corretamente
	Digitar o e-mail de usuário (modo autônomo)	Verificar se o e-mail é o correto	E-mail correto (adicionado no código) incluído no campo de e-mail
	Digitar a senha do usuário (modo autônomo)	Esperar o código inserir a primeira senha do array	Todas as senhas do array inseridas uma por vez
	Verificar se o login foi bem sucedido ou se possíveis erros e/ou inconsistências ocorreram.	Atentar para possíveis erros e/ou inconsistências	Sistema logado com a senha correta referente ao e-mail do usuário inserido, e mensagens de erros mostradas para as senhas incorretas

Cenário 2 (senha correta)	Entrar na página de Login	Verificar área de Login	Funcionalidades do Login funcionando corretamente
	Digitar o e-mail de usuário (modo autônomo)	Esperar o código inserir o primeiro e-mail do array	Todos os e-mails do array inseridos um por vez
	Digitar a senha do usuário (modo autônomo)	Verificar se a senha é a correta	Senha correta (adicionada no código) incluída no campo de senha
	Verificar se o login foi bem sucedido ou se possíveis erros e/ou inconsistências ocorreram.	Atentar para possíveis erros e/ou inconsistências	Sistema logado com a senha correta referente ao e-mail do usuário inserido, e mensagens de erros mostradas para as senhas incorretas

➤ **Desenho do Teste**

Criação da codificação para testar a aplicação, com base nos casos de testes, utilizando a linguagem Java. O código foi criado através da linguagem Java utilizando a IDE do Eclipse, e o mesmo visa testar se o sistema está capturando corretamente os e-mails e senhas dos usuários cadastrados no banco de dados para posteriormente efetuar o login dos mesmos. As funcionalidades dos inputs de e-mail/senha, botão de login e API Rest do JSON Server foram testados com o seguinte código:

```
package gamemania;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.junit.Before;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class TesteLoginGameMania {
```

```
private WebDriver driver;
```

```
@Before
```

```
public void abrirNavegador() {  
    System.setProperty("webdriver.chrome.driver", "C:\\Arquivos de  
Programas\\chromedriver\\chromedriver.exe");  
    driver = new ChromeDriver();  
    driver.manage().window().maximize();  
}
```

```
@Test
```

```
public void testeNavegador() {  
    driver.get("http://localhost:4200/login");  
    driver.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);  
    WebElement inputEmail = driver.findElement(By.id("emailLogin"));  
    WebElement inputSenha = driver.findElement(By.id("senhaLogin"));  
    WebElement botaoLogin = driver.findElement(By.className("enviar-form"));
```

```
String[] listaSenhas = {"casa123", "praiaABC", "cinema123", "bar123"};
```

```
for(int tentativas = 0; tentativas < listaSenhas.length; tentativas++) {  
  
    try {  
        inputEmail.clear();  
        inputSenha.clear();  
  
        inputEmail.sendKeys("tiago@email.com");  
        inputSenha.sendKeys(listaSenhas[tentativas]);  
        botaoLogin.click();  
  
        Thread.sleep(3000);  
  
    }catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}  
}  
}
```