Plano de Testes

SENAI-SP

Índice

1. Introdução 3

1.1 Objetivos 3

1.2 Escopo 3

2. Requisitos a Testar 4

2.1 Teste da Função de Busca de Cursos 4

3. Estratégia de Teste 5

3.1 Tipos de Teste 5

3.1.1 Teste da Interface do Usuário 5

3.2 Ferramentas 5

4. Recursos 6

4.1 Sistema 6

5. Cronograma 7

6. Fases de teste 8

6.1 Caso de uso: visualizar viabilidade 8

6.2 Tabela de casos de testes 8

7. Desenho do teste 8

7.1 Codificação com base nos casos de testes. 8

# Introdução

## Objetivos

Esse documento do Plano de Testes da plataforma SENAI-SP compõe-se dos seguintes objetivos:

• Identificar informações de projeto existentes e os componentes de software que devem ser testados.

• Listar os Requisitos a testar recomendados.

• Recomendar e descrever as estratégias de teste a serem empregadas.

• Identificar os recursos necessários e prover uma estimativa dos esforços de teste.

• Listar os elementos resultantes do projeto de testes.

## Escopo

Os testes que serão realizados na plataforma SENAI-SP serão os testes de navegabilidade, verificando a interação do usuário com a plataforma:

1. Verificar o comportamento da funcionalidade de busca de cursos

# Requisitos a Testar

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

## Teste da Função de Busca de Cursos

* Na página inicial da plataforma deve-se interagir com o campo de busca, procurar por um termo relacionado ao grupo de cursos, como por exemplo “gestão”, “TI”, “Informática”, entre outros, e verificar o retorno dessa busca, que deve retornar cursos com esta palavra-chave, sem anormalidades.

# Estratégia de Teste

## Tipos de Teste

Os tópicos abaixo são definidos como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

### Teste da Interface do Usuário

|  |  |
| --- | --- |
| Objetivo do Teste: | Verificar o seguinte:   * A navegação através dos alvos de teste acontece como o esperado, sem apresentar nenhum erro * Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões. |
| Técnica: | Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação. |
| Critério de Finalização: | É verificado se cada janela permanece consistente e dentro dos padrões aceitáveis. |

## Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

|  |  |  |
| --- | --- | --- |
|  | Ferramenta | Proprietário |
| Planejamento de Teste | Microsoft Office Word | Microsoft |
| Desenho de Teste | Eclipse IDE | Eclipse Foundation |
| Execução de Teste | Java | Oracle |

# Recursos

Essa seção apresenta os recursos recomendados para o projeto de teste, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

## Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

|  |
| --- |
| Recursos do Sistema |
| Código automatizador de Teste |
| — Java, com Selenium e JUnit |
| Repositório de Testes |
| —1 PC  1 PC para desenvolvimento dos Testes |
|  |

# Cronograma

|  |  |  |
| --- | --- | --- |
| **Tarefa** | **Data de Início** | **Data de Término** |
| Planejar Teste | 26/02/2022 | 26/02/2022 |
| Executar Teste | 26/02/2022 | 26/02/2022 |
| Avaliar Teste | 26/02/2022 | 26/02/2022 |

# Fases de teste

## Caso de uso: visualizar viabilidade

* Fluxo básico(#1):

- O usuário abre a página do sistema;

- O usuário navega pela página em busca de conteúdo;

- O usuário utiliza o campo de pesquisa para buscar um conteúdo específico;

- O usuário clica no botão de login;

- O usuário informa e-mail e senha;

- Se corretos, o sistema devolve a mensagem de usuário logado;

- Se incorretos, o sistema devolve mensagem de erro.

## Tabela de casos de testes

|  |  |  |  |
| --- | --- | --- | --- |
| Cenário | Caso de testes | Entradas | Resultados esperados |
| #1 | Acessar o sistema | Utilizar link de acesso | Exibir tela de usuário |
| Rolagem de tela para buscar conteúdo | Movimentar através de dispositivo externo | Conteúdo apresentado |
| Busca por meio de barra de pesquisa | Selecionar caixa de pesquisa e entrada de string de busca | Exibir resultados de busca com o número de ocorrências encontrdas |
| Acessar página de Login | Selecionar botão “login” | Acessar página de autenticação do usuário |
| Informar usuário e senha | Digitar strings referentes a login e senha | Acessar caixas para digitação de login e senha |
| Login “Usuário” | Informar dados do campo e-mail | Mensagem de usuário correto ou incorreto a depender do teto informado no campo |
| Login “Senha” | Informar dados do campo e-mail | Mensagem de usuário correto ou incorreto a depender do teto informado no campo |

# Desenho do teste

## Codificação com base nos casos de testes.

Linguagem Java

Software Eclipse

Código

package testessenaicursos;

import java.util.concurrent.TimeUnit;

package testessenaicursos;

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.chrome.ChromeDriver;

public class TesteBuscaCursos {

private WebDriver driver;

@Before

public void abrirNavegador() {

System setProperty("webdriver.chrome.driver" , "C:\\Progam Files\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

}

@Test

public void testeNavegador() {

driver.get("https://www.sp.senai.br/");

driver.manage().timeouts().implicitlyWait(3,TimeUnit.SECONDS);

driver.findElement(By.id("oqEstabuscando")).sendKeys("gest�o");

driver.findElement(By.className("btnBuscaJavaScript")).click();

}

}

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver; package testessenaicursos;

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.chrome.ChromeDriver;

public class TesteBuscaCursos {

private WebDriver driver;

@Before

public void abrirNavegador() {

System setProperty("webdriver.chrome.driver" , "C:\\Progam Files\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

}

@Test

public void testeNavegador() {

driver.get("https://www.sp.senai.br/");

driver.manage().timeouts().implicitlyWait(3,TimeUnit.SECONDS);

driver.findElement(By.id("oqEstabuscando")).sendKeys("gest�o");

driver.findElement(By.className("btnBuscaJavaScript")).click();

}

}

import org.openqa.selenium.chrome.ChromeDriver;

public class TesteBuscaCursos {

    private WebDriver driver;

    @Before

    public void abrirNavegador() {

        System setProperty("webdriver.chrome.driver" , "C:\\Progam Files\\chromedriver.exe");

        driver = new ChromeDriver();

        driver.manage().window().maximize();

    }

    @Test

    public void testeNavegador() {

        driver.get("https://www.sp.senai.br/");

        driver.manage().timeouts().implicitlyWait(3,TimeUnit.SECONDS);

        driver.findElement(By.id("oqEstabuscando")).sendKeys("gest�o");

        driver.findElement(By.className("btnBuscaJavaScript")).click();

    }

}