

Nomes: Elizabeth, Emily Cristina, Giovanna Braz, Julia Leite, Bárbara Nascimento
Desenvolvimento de Sistemas
03/09/2024

Plano de Teste - Software para Sistema Web de Manutenção de Máquinas

1. Objetivo do Plano de Testes

O objetivo deste plano de testes é definir as estratégias e os critérios para verificar a funcionalidade do cadastro de administradores, a geração de relatórios em PDF das últimas manutenções e o cadastro de manutenções no sistema. Os testes garantirão que o sistema opere corretamente e sem falhas, tanto na interface do usuário quanto na lógica de backend.

2. Escopo do Plano de Testes

Inclusões:

- Frontend:

- Validação da interface de cadastro de administradores.
- Geração e visualização de relatórios em PDF.
- Cadastro e exibição de manutenções.

- Backend:

- Funcionalidade dos endpoints para cadastro de administradores e manutenções.
- Geração de relatórios em PDF a partir dos dados armazenados.
- Integridade e segurança dos dados no banco de dados.

Exclusões:

- Testes de outros módulos que não envolvem o cadastro de administradores, relatórios e cadastro de manutenções.

3. Abordagem de Testes

Frontend

- Testes Funcionais/Integração: Utilizar ferramentas como Cypress ou Selenium para verificar se as funcionalidades do frontend operam conforme o esperado.
- Testes de Interface/Usabilidade: Realizar testes de usabilidade para garantir que a interface do usuário seja intuitiva e que os formulários funcionem corretamente.

Backend

- Testes Unitários: Cada função do backend que lida com o cadastro de administradores e manutenções será testada usando frameworks como `unittest` ou `pytest`.
- Testes de Integração: Testar a interação entre os módulos do backend (por exemplo, a API de cadastro e o banco de dados) usando Postman ou pytest-django.

4. Casos de Teste

Frontend

- Verificação de Layout e Responsividade:
 - Testar a interface de cadastro em diferentes tamanhos de tela para garantir que os elementos estejam corretamente alinhados.
- Testes de Formulários:
 - Testar se o formulário de cadastro de administradores aceita dados válidos e retorna a mensagem de sucesso ao ser submetido.
 - Garantir que erros sejam exibidos corretamente ao tentar enviar dados inválidos (como e-mail já cadastrado).
- Geração de Relatórios em PDF:
 - Testar a geração de relatórios em PDF a partir da lista de manutenções e garantir que o PDF é formatado corretamente.

Backend

- Cadastro de Administradores:
 - Validar se o endpoint de cadastro de administradores aceita dados corretos e cria um registro no banco de dados.
 - Testar o tratamento de erros ao tentar cadastrar um administrador com dados duplicados ou inválidos.
- Cadastro de Manutenções:
 - Testar o endpoint que permite o cadastro de manutenções e verificar se as manutenções são registradas corretamente no banco de dados.
- Geração de Relatórios em PDF:
 - Validar a geração do PDF das últimas manutenções, garantindo que os dados estejam corretos e completos.

5. Ambiente de Testes

- Sistema Operacional: Windows/Linux
- Navegadores: Google Chrome, Mozilla Firefox, Microsoft Edge
- Versões do Python: 3.8 ou superior
- Frameworks: Django para backend
- Banco de Dados: SQLite para testes ou PostgreSQL
- Infraestrutura: Servidores locais ou em nuvem para testes.

6. Ferramentas de Testes

- Frontend: Selenium
- Backend: Unittest, Pytest, Postman
- Integração Contínua: GitHub Actions ou Jenkins para automação de testes.

7. Critérios de Aceitação

- A geração de PDFs deve ser realizada sem erros e dentro de 3 segundos.

- O cadastro de administradores deve ser bem-sucedido com dados válidos e deve apresentar mensagens de erro para dados inválidos.

- As manutenções devem ser corretamente registradas no banco de dados, e os relatórios devem conter informações precisas e completas.

8. Relatórios de Testes

Os resultados dos testes serão documentados utilizando ferramentas de gerenciamento de testes como:

- TestRail ou Jira para gerenciar casos de teste e registrar resultados.
- Relatórios gerados pelas ferramentas de testes automáticos, com logs detalhados sobre falhas e erros ocorridos durante os testes.

Exemplo de Casos de Teste

Frontend

ID Teste	Descrição	Esperado
F001	Teste de Responsividade	Layout correto 1024x768
F002	Teste de Formulário de Cadastro de ADM	Cadastro bem-sucedido
F003	Teste de Geração de PDF	PDF gerado corretamente
F004	Teste de mensagens de Erro	Mensagem exibida para dados inválidos

Backend

ID Teste	Descrição	Esperado
B001	Teste de API de Cadastro de ADM	Registro criado com sucesso
B002	Teste de Endpoint de Cadastro de Manutenção	Manutenção registrada com sucesso
B003	Teste de Geração de PDF de Manutenções	PDF gerado com dados corretos
B004	Teste de Erro ao Cadastrar Administrador Duplicado	Erro de duplicação retornado