Centro Universitário de Anápolis- Unievangélica

Cíntia Galvão;Isabella R Canedo; Ricardo Brendhon; Samara de Lourdes

**Plano de Teste**

O plano de teste se faz necessário para detalhar cada teste desenvolvido no projeto, o que cada um é, quais ferramentas serão utilizadas em cada teste, e o que será verificado.

Os testes que serão executados são: Teste Funcional, Teste não funcional (Teste de Carga), Teste Unitário, e Teste de Integração.

O projeto que será testado é o NextStar, este projeto foi desenvolvido utilizando o framework angular 5 no front-end, e utilizando o framework Spring boot para desenvolver a API (back end). As linguagens de programação utilizadas foram Java e typescript/html/css.

* **Casos de Teste**

Pode ser elaborado para identificar defeitos na estrutura interna do software por meio de situações que exercitem utilizadas na codificação, ou ainda garantir que os requisitos do software que foi construído sejam plenamente atendidos.

Podemos Utilizar a ferramenta de casos de Uso para criar e rastrear um caso de teste, facilitando assim a identificação de possíveis falhas.

* Ferramenta: Excel

Os cenários de teste devem verificarar os possíveis retornos do caso de uso cadastrar modelo.

* **Teste Funcional**

São testes que avaliam o comportamento da aplicação. Dados de entrada são fornecidos o teste e executado, e o resultados são obtidos e comparados. O teste funcional é aplicável a todas as fases do teste(unitário, integração, sistema e aceitação). Testa as funcionalidades, requerimento, regras de negócio presentes na documentação.

* Ferramentas: Browser(versão 66.0.3- 64 bits) / Extensão (katalon- versão: 3.7.0)
* O que será verificado: O funcionamento do caso de uso cadastrar modelo, visando automatizar a funcionalidade e verificar se os campos estão de acordo como foi especificado.
* **Teste Unitário**

Teste unitário se refere a testar a menor unidade do código ou seja o menor trecho de código, por exemplo o método de uma classe.

O objetivo é mostrar que cada unidade atende corretamente sua especificação. Os benefícios é encontrar problemas o quanto antes, facilitando a mudança da unidade, e simplificando a integração melhorando a documentação. Uma unidade é a menor parte estável de um programa de computador. Uma unidade pode ser uma função individual ou um procedimento. Cada teste de unidade é independente dos demais, podendo testar cada módulo isolado.

* Ferramenta: IntelliJ IDEA 2019.1 x64
* Máquina: Intel(R) Core(TM) i5-7200U CPU 2.71GHz 4GB SSD 240GB.
* O que será verificado: Pequenas partes do caso de uso Cadastrar Modelo com base em suas possíveis entradas e saídas esperadas.
* **Teste de Integração**

É a forma de se testar a combinação das unidade em conjunto. Neste casom a idéia é encontrar falhas na junção destas unidades. Pode ser que a classe X funcione bem sozinha mas ao ser utilizada pela classe Y, ela deixe de funcionar.

Os benefícios são que além de testar funcionalmente, pode assegurar performance e confiabilidade. Ajudam a garantir que o trabalho de um desenvolvedor não está afetando o trabalho de outro e em equipes grandes isso pode fazer toda a diferença se forem realizados com frequencia.

* Ferramenta: IntelliJ IDEA 2019.1 x64
* Máquina: Intel(R) Core(TM) i5-7200U CPU 2.71GHz 4GB SSD 240GB.
* O que será verificado: O módulo de integração do código.
* **Teste Não Funcional**

São testes que verificam atributos de um componente de sistema que não se relacionam com a funcionalidade com por exemplo: teste de performance, teste de carga, teste de estresse, teste de usabilidade.

* Ferramentas: Java (Java-8 v8.0) / Jmeter(v5.1.1); / Browser (Firefox-v66.0.3-64bits)
* O que será verificado: Teste carga com 100 usuários acessando a aplicação simultaneamente e com o tempo de resposta de 01 segundo, com apenas uma requisição por usuário.