Centro Universitário de Anápolis- Unievangélica

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

*Plano de Teste*

Neste plano de teste iremos detalhar cada teste desenvolvido no projeto, o que cada um é, quais ferramentas irá ser utilizado em cada teste, e o que irá ser verificado. Os testes que irá conter no projeto são: Casos de Teste;Teste Unitário; Teste Integrado; Teste Funcional e Teste não-funcional.

\*O projeto nextstar é feito utilizando o desenvolvimento do front, e utilizando o framework spring boot para desenvolver a API(backEnd)

* **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. A única ferramenta utilizada irá ser o Excel.

Será cenários que verificaram possíveis retornos do caso de uso cadastrar modelo.

* **Teste Unitário:** É a forma de se testar unidades individuais de código fonte. Unidades podem ser métodos,classes,funcionalidades,módulos,etc.

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. JAVA; TYPESCRIPT/HTML/CSS são as ferramentas utilizadas.

Será verificado se os resultados estão coerente com o que foi codificado e implementado no caso de uso.

* **Teste Integrado:** 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. As ferramentas são: (Código: Angular- FrontEnd/ Java-BackEnd/ Banco de dados: PostgreSQL/ Node.js/JDK/Visual Studio Code/ IntelliJIDEA2019.1 x64/ Eclipse Jee 2019-03).

Será verificado o módulo de integração do código.

* **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. As ferramentas utilizadas são o Browser(versão 66.0.3- 64 bits) e com a extensão (katalon- versão: 3.7.0)

Será verificado o caso de uso cadastrar modelo, visando automatizar a funcionalidade e verificar se os campos de acordo como foi especificado.

* **Teste não-funcional:** São testes que verificam atributos de um componente de sistema que não se relacionam com a funcionalidade(confiabilidade, eficiencia, usabilidade, manutenabilidade e portabilidade). As ferramentas utilizadas neste teste são: Java (Java 8-versão:8.0); Jmeter(versão: 5.1.1); Browser(Firefox-versão: 66.0.3-64bits)

Será verificado no teste não funcional: Teste carga com 100 usuários acessando a aplicação simultaneamente e com o tempo apropriado de resposta de 01 segundo. Será feito apenas uma requisição por usuário.