**PLANO DE TESTES**

O presente plano de testes tem por objetivo alinhar as estimativas e expectativas quanto aos testes e qualidade de software além de servir também como embasador dos casos de testes.

**POR QUE TESTAR**

Os testes são essenciais pois será através dos testes que conseguiremos garantir com algum grau de certeza de que o desenvolvimento foi feito da forma correta, ou seja, alinhado às expectativas do cliente/stakeholders.

**O QUE TESTAR**

Diante da IMPOSSIBILIDADE de testar tudo, os testes serão realizados com base na análise de risco potencial para o negócio, dessa forma tudo que é de mais importante será priorizado, exemplo o "carrinho" em um e-commerce.

**COMO TESTAR**

Os testes poderão ser feitos de forma manual ou automatizada, a melhor forma que garante um maior grau na análise são ambos os testes sendo realizados, dessa forma é possível garantir uma maior validação no que está desenvolvido.

**TESTAR X GARANTIA DE QUALIDADE**

É importante destacar que os testes são diferentes da garantia de qualidade “QA”, enquanto o teste é uma ação, a garantia de qualidade é considerada um processo e por isso diversos fatores serão levados em consideração nos indicadores da gestão de qualidade do que está sendo desenvolvido.

**ESTRATÉGIA DE TESTES**

Como estratégias de testes será usada como parâmetro a pirâmide de testes, esse é um dos modelos de padrão de gestão de qualidade mais usado, observe que na medida que a pirâmide vai da base para a ponta os testes ficam mais caros e levam mais tempo porém muitas vezes é justamente na parte de UI e Service onde se encontram a maiorias dos erros.



**COBERTURA DE TESTES**

**UI.**

**Teste funcional**

O teste funcional, que também é conhecido como “teste de técnica de caixa - preta”, são os testes definidos de acordo com os requisitos funcionais do software.

**Teste regressivo**

Esse teste deve ser realizado sempre que uma nova funcionalidade for implementada, pois todos os testes funcionais devem ser executados e dessa forma será possível verificar se a nova funcionalidade não quebrou alguma outra.

**SERVICE.**

**Testes de API**

Nos testes de API é testado como um micro serviços se comporta com diferentes entradas, é usado uma client API como o Postman para fazer as Request e validação dos Response e status code HTTP.

**Teste de integração**

Nos testes de integração são realizados os testes e validação de diferentes componentes da aplicação, exemplo disso é a validação via banco de dados se as informações salvas pelo cliente estão sendo persistidas de forma correta no BD.

**UNIT.**

**Testes de unidade**

Esses testes são para validar a menor parte de um sistema, como uma função ou classe.

**Testes de componentes**

Similar ao testes de unidade, esse teste é realizado com algum tipo de dado para validação da função.

**TESTES AUTOMATIZADOS**

Os testes automatizados são muito necessários pois com esses testes conseguimos validar a jornada do cliente na aplicação, além de ser a melhor forma de executar o teste regressivo, pois como já foi dito, nos testes regressivos precisamos validar que uma nova entrega não quebrou uma entrega anterior e única forma de validar isso é testando TUDO que foi desenvolvido anteriormente, esse teste feito de forma manual é extremamente demorado porém feito de forma automatizada é rápido, e o feedback no que deu certo e o que deu errado muito preciso.

**ESTRATÉGIA PARA OS TESTES AUTOMATIZADOS**

Para os testes automatizados a ideia é usar as tecnologias mais robustas e usadas no mercado, que são as seguintes: **Linguagem de programação Java + Frameworks JUnit + Selenium WebDriver + Cucumber,** através dessas ferramentas é possível automatizar os testes de ponta a ponta, e assim conseguimos também garantir de uma maneira mais eficiente nossas entregas.