**SENAI – DESENVOVIMENTO DE SISTEMAS**

**UNIDADE CURRICULAR – TESTE DE SISTEMAS**

**KAIO GOMES DO NASCIMENTO MAZZA**

**GUSTAVO WENDT**

**TIPOS DE TESTE DE SISTEMA**

**JOINVILLE**

**2025**

1. **Testes de caixa branca:**

Os testes de caixa branca são tipos de teste que buscam trazer um conhecimento interno do sistema, como o sistema está funcionando em seu código fonte, como a estrutura do sistema roda, é basicamente isso que os testes de caixa branca buscam trazer. O objetivo é garantir o funcionamento do código fonte, parte a parte.

1. **Teste de Unidade:**

Um teste unitário é um tipo de teste de software que verifica se uma unidade individual do código (como uma função, método ou classe) está funcionando corretamente, isoladamente de outros componentes do sistema. É uma etapa crucial no processo de desenvolvimento de software para garantir a qualidade e confiabilidade do código. Este resgata blocos de códigos maiores para então testar os códigos menores.

**EXEMPLO:**

Suponha que você está desenvolvendo um sistema e uma das funções do sistema é verificar se um número é **primo**.

Você escreveu a seguinte função (em qualquer linguagem de programação):

**Função:** Verifica se um número inteiro é primo.  
**Lógica interna da função:**

* Se o número for menor ou igual a 1, retorna **falso** (não é primo).
* Para números maiores que 1, verifica se existe algum divisor entre 2 e a raiz quadrada do número.
* Se encontrar um divisor, retorna **falso**.
* Se não encontrar nenhum divisor, retorna **verdadeiro** (o número é primo).

**Casos de teste que seriam escritos:**

1. **Teste com número negativo**
   * Entrada: -3
   * Saída esperada: falso
   * Justificativa: Números negativos não são primos.
2. **Teste com zero e um**
   * Entrada: 0 e 1
   * Saída esperada: falso
   * Justificativa: Zero e um não são considerados primos.
3. **Teste com números primos pequenos**
   * Entrada: 2, 3, 5, 13
   * Saída esperada: verdadeiro
   * Justificativa: São números primos conhecidos.
4. **Teste com números não primos**
   * Entrada: 4, 9, 15
   * Saída esperada: falso
   * Justificativa: São divisíveis por outros números além de 1 e eles mesmos.
5. **Teste de Integração:**

Os testes de integração servem para verificar como duas partes diferentes do sistema interagem entre si, como por exemplo: Como o banco de dados vai recolher os dados digitados pelo cliente em um sistema da WEB. Este garante não somente que partes isoladas do sistema funcionem, mas também que várias partes dos sistemas funcionem em conjunto, que o sistema trabalhe junto de maneira eficaz.

**EXEMPLO:**

No sistema X, são necessários um envio e um recebimento de dados rápido do banco de dados. Então, para isso podemos realizar o seguinte teste:

Precisamos de uma requisição da API para auxiliar o cliente e então enviar esses dados ao banco de dados.;

1. **O que é Teste de Caixa-Preta?**

O teste de caixa-preta é uma técnica em que o testador verifica o sistema analisando apenas as entradas e saídas, sem conhecimento da estrutura interna do código. Ou seja, ele avalia o que o software faz, e não como ele faz.

1. **Teste Funcional**

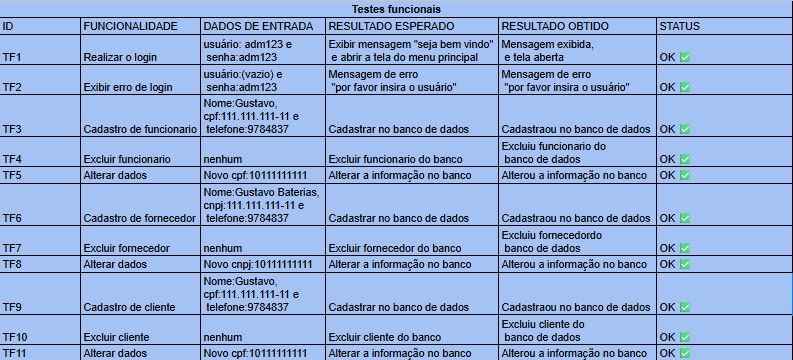
São testes focados nos requisitos e funcionalidades do sistema. O objetivo é garantir que cada funcionalidade funcione conforme o esperado, verificando os resultados com base nos dados de entrada e saídas esperadas.

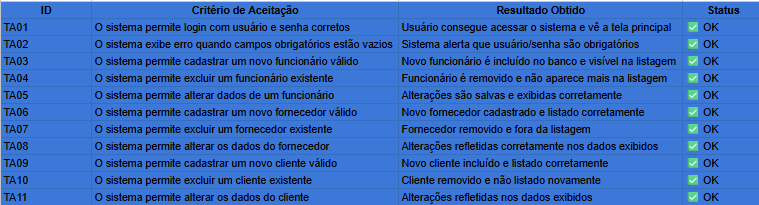
1. **Teste de Aceitação**

É o teste realizado para validar o sistema junto ao usuário final, assegurando que ele atende às necessidades e critérios definidos. Ou seja, verifica se o que foi desenvolvido realmente resolve o problema do usuário e está pronto para uso.

1. **Teste Exploratório**

Esse tipo de teste ocorre quando o testador explora livremente o sistema para encontrar falhas que não foram previstas. É um teste menos estruturado, baseado na experiência e intuição do testador para descobrir comportamentos inesperados.

**EXEMPLO SOBRE TESTE FUNCIONAL:**

**EXEMPLO SOBRE TESTE DE ACEITAÇÃO:**

**EXEMPLO SOBRE TESTE EXPLORATÓRIO:**

Nos testes exploratórios, foram executadas interações livres com o sistema para identificar falhas que não foram previstas nos casos de teste.

***Observações:***

* Ao preencher o campo “Telefone” com texto ao invés de números, o sistema aceitou o valor e o armazenou incorretamente.
* O botão “Voltar” da tela de cliente não limpa o campo de “idcliente”.
* Não houve problemas visuais, porém o layout não se adaptou bem a diferentes tamanhos de tela.
* O design não seguiu o mesmo modelo em todas as telas.
* O campo “CNPJ” da tela de fornecedor não diferencia CPF de CNPJ.

1. **O que é Teste de Caixa-Cinza?**

Os testes de caixa-cinza combinam características dos testes de caixa-preta e caixa-branca. Ou seja, o testador tem algum conhecimento parcial da estrutura interna do sistema (código, arquitetura, integrações), mas testa focando nos resultados externos. Exemplo: validar a segurança da API sabendo o endpoint e formato dos dados, mas sem ver o código completo.

* 1. **Testes de Regressão**

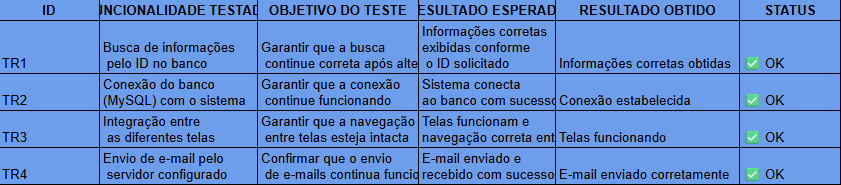
Os testes de regressão servem para garantir que alterações recentes (novas funcionalidades, correções de bugs, etc.) não quebrem o que já funcionava antes. Depois de cada mudança no sistema, reexecutamos testes anteriores para ter certeza que as outras partes do software permanecem corretas.

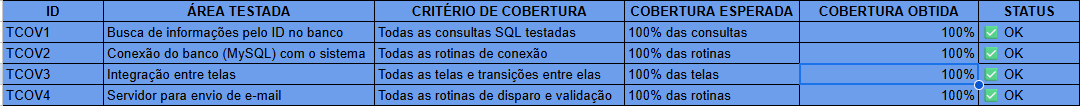
* 1. **Testes de Cobertura**

Os testes de cobertura analisam o quanto do código-fonte foi realmente executado pelos testes. Por exemplo:

* Cobertura de Linhas: que porcentagem das linhas do programa foram testadas.
* Cobertura de Caminhos ou Ramos: que porcentagem dos fluxos lógicos foram executados.

A ideia é medir a abrangência da sua suíte de testes e identificar partes que ainda não foram testadas.

**EXEMPLO SOBRE TESTE DE REGRESSÃO:**

**EXEMPLO SOBRE TESTE DE COBERTURA:**