

# Mentoria Seja Tester

## Academy – Aula 6

Níveis de Teste e Tipos de Teste



# Sumário

Quais assuntos vamos aprender nessa aula?

- Níveis de teste
- Tipos de teste
- Testes não funcionais
- Teste de Regressão
- Teste de manutenção



# NÍVEIS DE TESTE

Os níveis de teste são como etapas organizadas para testar um software. Em cada etapa, testamos uma parte diferente do sistema, desde os pedaços menores (como botões e funções separadas) até o sistema inteiro funcionando junto. Essas etapas são feitas conforme o software vai sendo desenvolvido, desde o começo até quando ele já está quase pronto para ser usado.

Esses níveis de teste fazem parte do processo de desenvolvimento do software. Em alguns tipos de projetos mais "linha reta" (como uma receita que você segue passo a passo), cada etapa de teste só começa depois que a anterior termina. Já em projetos mais flexíveis (onde o trabalho vai e volta), isso pode ser diferente, e as etapas de teste podem até acontecer ao mesmo tempo.

Além disso, temos os tipos de teste, que são testes voltados para verificar coisas específicas do software, como se ele é rápido, se é fácil de usar ou se é seguro. Esses tipos de testes podem ser feitos em qualquer uma das etapas (ou níveis) que mencionei antes.

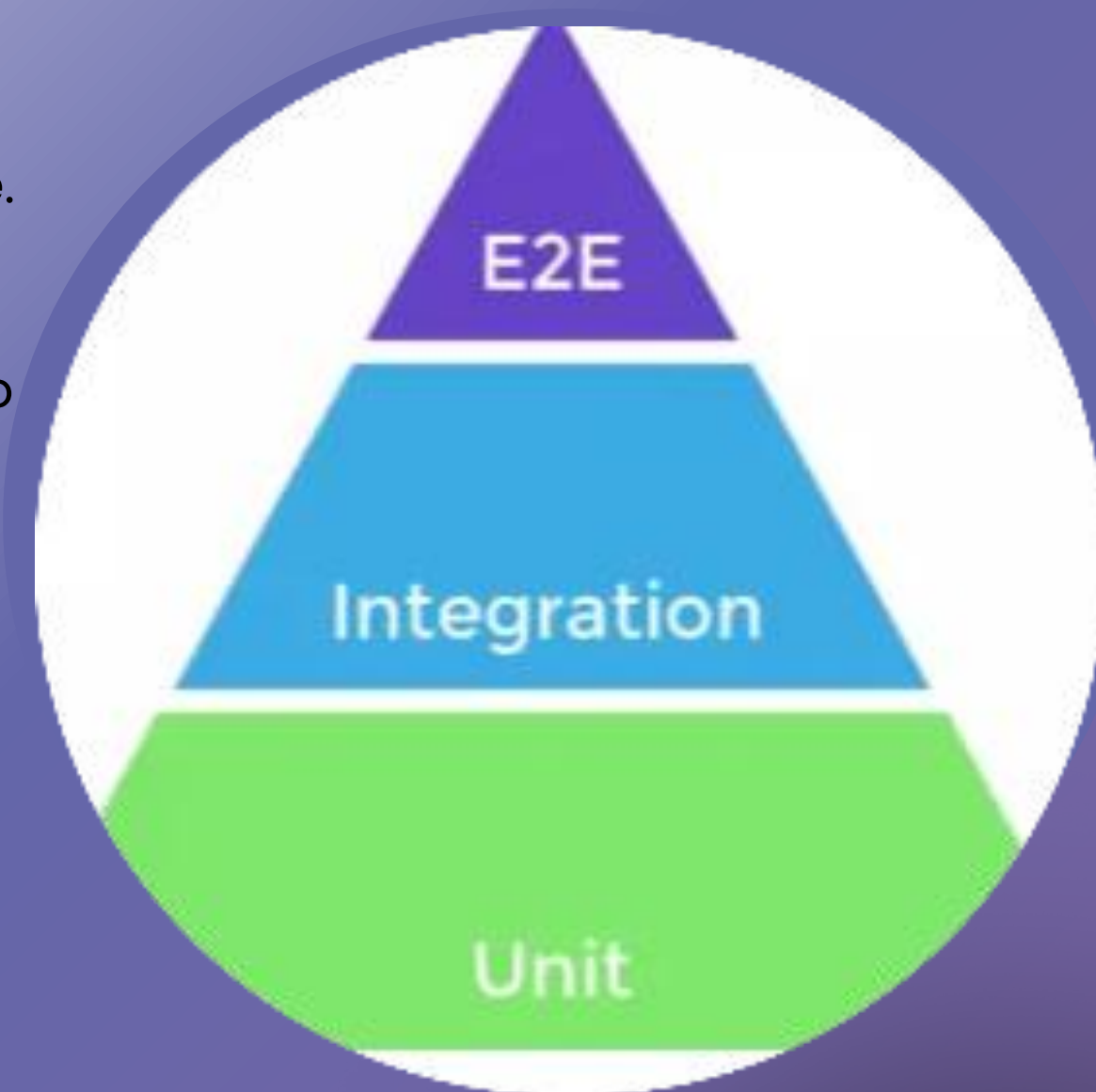




# NÍVEIS DE TESTE

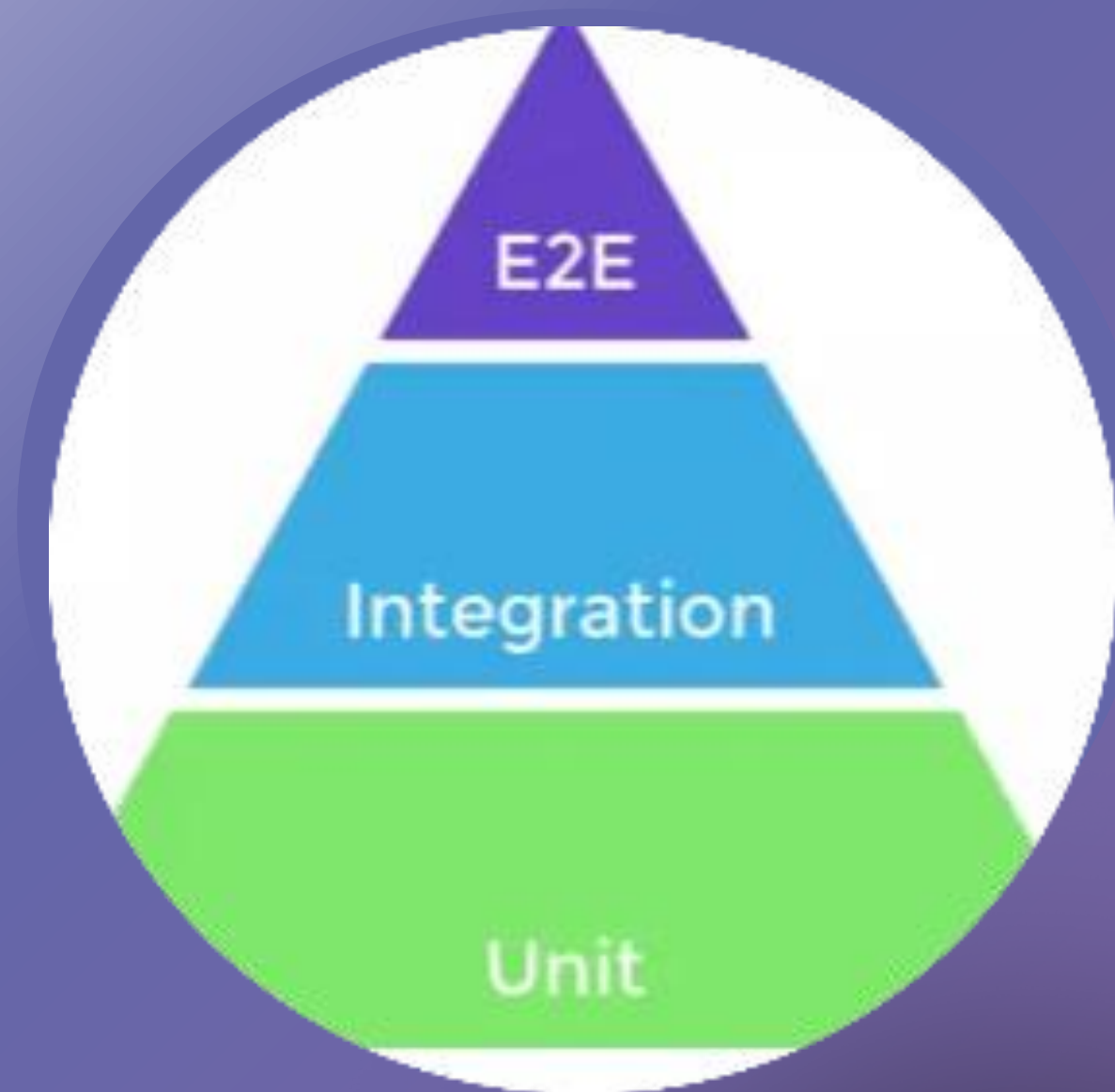
Existem cinco etapas principais para testar um sistema, e cada uma tem um foco diferente. Vou explicar cada uma de forma fácil:

- **Teste de Componente (ou Teste de Unidade):** Aqui, testamos os menores pedacinhos do sistema, como se estivéssemos testando peça por peça de um quebra-cabeça. Isso é feito geralmente pelos próprios programadores, direto onde eles estão escrevendo o código, usando ferramentas especiais para isso.
- **Teste de Integração de Componentes (ou Integração de Unidades):** Depois de testar as peças sozinhas, juntamos algumas delas para ver se funcionam bem juntas. É como montar partes do quebra-cabeça e ver se se encaixam direito. Dependendo da estratégia, podemos começar testando de baixo para cima, de cima para baixo ou tudo de uma vez.



# NÍVEIS DE TESTE

- **Teste de Sistema:** Agora testamos o sistema como um todo, como se ele já estivesse quase pronto para ser usado. Verificamos se ele faz tudo que deveria (como simular o uso real do começo ao fim) e também se ele é rápido, fácil de usar, seguro etc. Às vezes usamos simulações para partes do sistema, se ainda não estiverem prontas. Normalmente esse teste é feito por uma equipe separada, que segue o que foi combinado na documentação do projeto.
- **Teste de Integração de Sistema:** Aqui o foco é ver se o sistema funciona direitinho com outros sistemas ou serviços, como, por exemplo, quando ele precisa buscar dados em outro lugar. É importante que esse teste aconteça em um ambiente parecido com o que será usado na vida real.



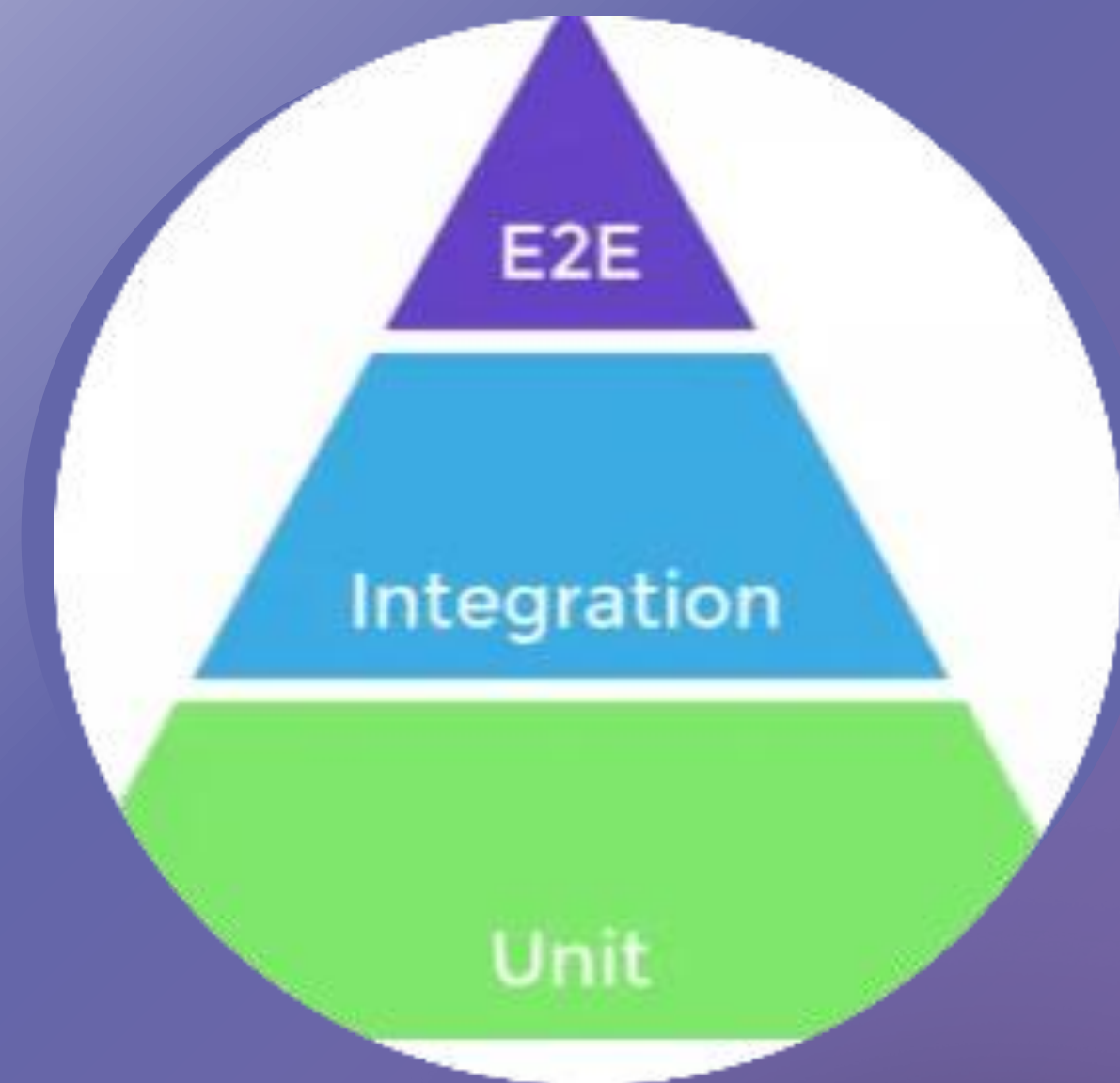


# NÍVEIS DE TESTE

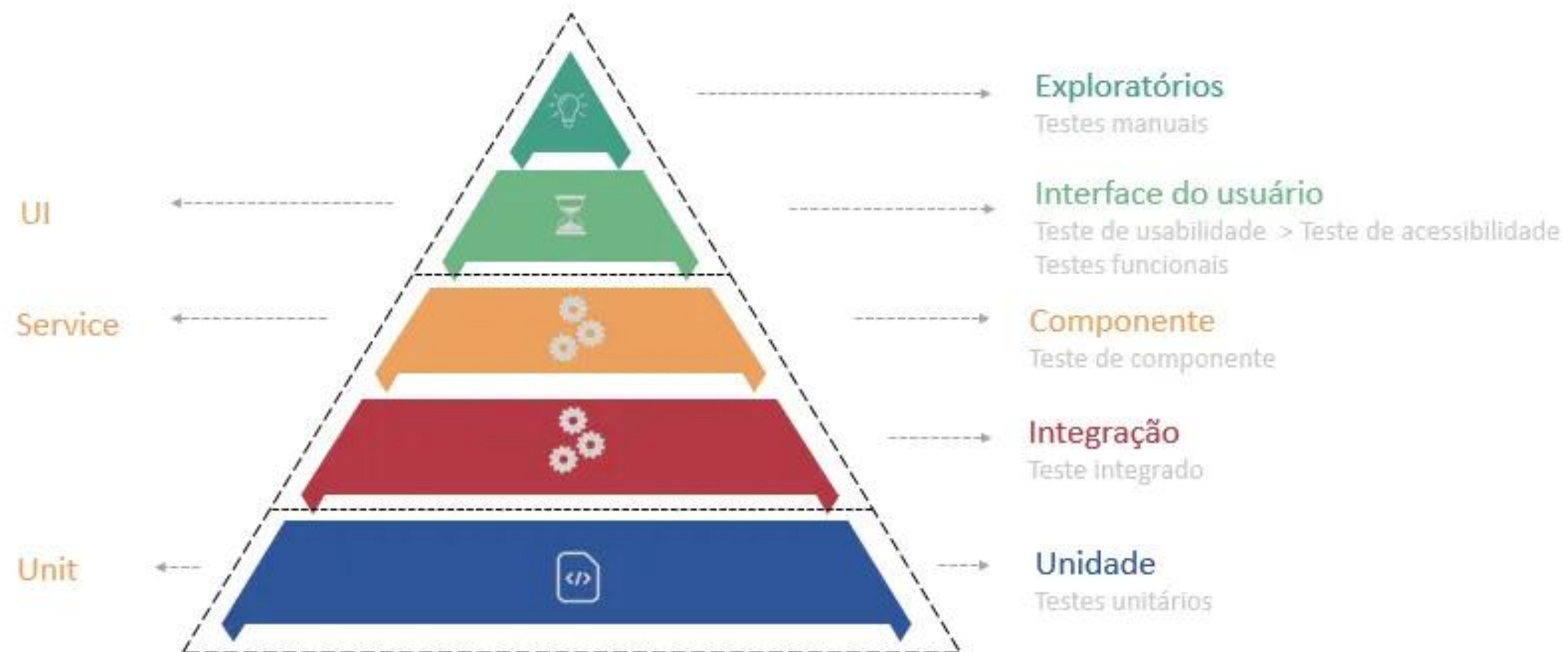
- **Teste de Aceite:** Essa é a última etapa, onde verificamos se o sistema está realmente pronto para ser usado pelas pessoas. A ideia é conferir se ele atende tudo o que foi pedido pelo cliente ou pelos usuários. Quem faz esse teste geralmente são os próprios usuários ou representantes deles. Existem várias formas de fazer isso: com testes feitos por usuários finais, por operadores, por exigências legais ou até com versões de teste antes do lançamento (como as versões alfa e beta).

**Essas etapas são diferentes entre si em vários aspectos, como:**

- O que está sendo testado (por exemplo, uma parte ou o sistema inteiro);
- O objetivo de cada teste (descobrir erros, ver se está funcionando como esperado etc.);
- De onde vêm as informações usadas para os testes (documentos, regras do negócio etc.);
- Que tipo de problema pode aparecer em cada etapa;
- Quem é responsável por cada tipo de teste e como ele é feito.



# NÍVEIS DE TESTE



# TIPOS DE TESTE

Existem vários tipos de testes que podemos fazer em um projeto de software, dependendo do que queremos verificar. Neste caso, vamos focar em quatro tipos principais. Aqui vão os dois primeiros explicados de forma simples:

**1. Teste Funcional:** Esse teste serve para ver se o sistema faz o que ele deveria fazer.

Por exemplo: se estamos testando um aplicativo de compras, verificamos se ele consegue adicionar um produto no carrinho, calcular o valor total ou finalizar a compra corretamente.

Esses testes respondem à pergunta: 👉 “O sistema está fazendo o que foi combinado?”

“O objetivo aqui é conferir se tudo está funcionando direitinho, sem erros e do jeito certo.





# TIPOS DE TESTE

**2. Teste Não Funcional:** Esse tipo de teste não foca no que o sistema faz, mas sim em como ele se comporta.

Por exemplo: o sistema está rápido? É fácil de usar? Funciona bem em diferentes tipos de celular ou computador? É seguro?

Esses testes respondem à pergunta: 🙌 “O sistema está funcionando bem?”

Segundo uma norma internacional (ISO/IEC 25010), os pontos mais importantes que avaliamos nesse tipo de teste são:

- Velocidade e desempenho (ele é rápido o suficiente?);
- Compatibilidade (funciona em vários dispositivos e sistemas?);
- Facilidade de uso (qualquer pessoa consegue usar sem dificuldade?);
- Confiabilidade (ele funciona sem travar ou dar erro?);
- Segurança (os dados estão protegidos?);
- Facilidade de manutenção (é fácil de corrigir problemas ou fazer melhorias?);
- Portabilidade (pode ser usado em outros lugares ou sistemas com facilidade?).





# TESTES NÃO FUNCIONAIS

- Às vezes, é bom começar a verificar certos detalhes técnicos logo no começo do projeto, como por exemplo quando estamos conferindo partes menores do sistema ou o sistema inteiro. Muitos desses testes são feitos com base nas funções principais do sistema, só que ao invés de só ver se a função funciona, a gente também vê como ela funciona, por exemplo, se ela é rápida o suficiente ou se dá para usar em outro tipo de computador ou celular.
- Se a gente só descobrir esses problemas no final do projeto, pode ser um grande risco e atrapalhar muito. Em alguns casos, esses testes precisam ser feitos em lugares ou com ferramentas bem específicas, como uma sala especial onde as pessoas testam se o sistema é fácil de usar.





# TESTE DE REGRESSÃO

**Teste de Regressão:** Esse teste é feito para garantir que nenhuma outra parte do sistema foi prejudicada por causa da mudança ou correção feita. Por exemplo, corrigir um erro em uma parte do programa pode sem querer quebrar outra parte, ou até afetar outro sistema que está conectado.

Antes de fazer esse tipo de teste, é importante analisar o que pode ter sido afetado com a mudança, para saber o que exatamente precisa ser testado. Esse processo se chama **análise de impacto**.

Os testes de regressão são feitos muitas vezes durante o desenvolvimento de um sistema, e a cada nova versão, o número de testes costuma aumentar. Por isso, é muito comum automatizar esses testes, ou seja, usar programas para rodá-los automaticamente. Isso ajuda a economizar tempo e evitar erros humanos.







# TESTE DE REGRESSÃO

Se a equipe usa uma prática chamada Integração Contínua (CI), que é comum em projetos modernos como os que seguem o modelo DevOps, é recomendado que os testes de regressão já comecem a ser automatizados desde o início do projeto.

Esses testes automatizados podem acontecer em diferentes partes do sistema, dependendo do que foi alterado. Por fim, sempre que algo for corrigido ou alterado no sistema, é importante fazer testes de confirmação e/ou de regressão em todos os níveis do sistema, desde as menores partes até o sistema completo.





# TESTE DE MANUTENÇÃO

Existem diferentes tipos de **manutenção em sistemas** (ou seja, quando algo precisa ser ajustado ou atualizado em um programa).

Essa manutenção pode ter objetivos diferentes, como:

- Corrigir erros;
- Se adaptar a mudanças no ambiente (como mudanças na internet ou nos computadores onde o sistema roda);
- Melhorar o desempenho ou tornar o sistema mais fácil de manter no futuro.

Essas mudanças podem ser planejadas, com data e organização para acontecerem, ou podem ser feitas às pressas, quando surge um problema urgente, essas últimas são chamadas de hotfixes.

Antes de mudar algo no sistema, é importante fazer uma análise de impacto, que é basicamente estudar se essa mudança pode causar problemas em outras partes do sistema. Isso ajuda a decidir se vale a pena ou não fazer a alteração.



# TESTE DE MANUTENÇÃO

Quando algo é alterado em um sistema que já está funcionando (em produção), é importante fazer dois tipos de verificação:

- Ver se a alteração deu certo;
- Ver se outras partes do sistema que não foram alteradas continuam funcionando normalmente, essa segunda parte é o famoso teste de regressão.

O que influencia os testes de manutenção? A quantidade e o tipo de testes que precisam ser feitos dependem de:

- O quanto a mudança pode ser arriscada;
- O tamanho do sistema;
- O tamanho da mudança que foi feita.







# TESTE DE MANUTENÇÃO

## Tipos de manutenção e situações que exigem testes

### **1. Modificações:**

- Melhorias planejadas, como novas versões com novos recursos;
- Correções de erros (como os hotfixes, feitos com urgência).

### **2. Mudanças no ambiente:**

- Por exemplo, quando o sistema precisa funcionar em um novo tipo de computador ou plataforma. Nesses casos, é preciso testar não só a parte que foi alterada, mas também como o sistema se comporta nesse novo ambiente;
- Às vezes também é necessário migrar dados de um sistema para outro, e isso também exige testes para garantir que os dados foram transferidos corretamente.

### **3. Aposentadoria do sistema:**

- Quando o sistema não vai mais ser usado (chegou ao fim da vida útil), é necessário guardar os dados de forma segura, se eles precisarem ser mantidos por muito tempo;
- Também é importante testar se os dados podem ser recuperados facilmente no futuro, caso alguém precise acessá-los de novo.



- DÚVIDAS?
- SUGESTÕES?
- RECLAMAÇÃO?
- ELOGIO?

