O que é Teste Automatizado?

📘 Artigo – O que é Teste Automatizado?

🟩 Resumo por tópicos:

🔍 O que é teste automatizado?

- Técnica que **valida funcionalidades de software** de forma automatizada, sem intervenção humana.
- Ideal para testes **repetitivos**, **amplos** ou **impossíveis de realizar manualmente**.
- Usado em: **unitários, API, regressão, sistema, smoke, UI**.
- Suporte essencial para **CI/CD** e **Continuous Testing**.

⚙️ Como funciona o teste automatizado?

1. Escolher a ferramenta de teste
2. Definir o escopo de automação
3. Planejar, projetar e desenvolver os testes
4. Executar os testes e coletar relatórios
5. Fazer manutenção dos scripts com novas versões

🧰 Frameworks e abordagens

- **Lineares:** simples, mas pouco reutilizáveis
- **Modular:** scripts independentes, melhor manutenção
- **Data-driven:** um script, múltiplos dados
- **Keyword-driven:** usa palavras-chave para facilitar criação por não programadores
- **Híbridos:** combina abordagens

🛠️ Ferramentas mencionadas

- **Selenium:** testes web multi-browser e multi-linguagem
- **Robotium:** testes em dispositivos Android
- **Cypress:** testes fim-a-fim, integração e unitários no navegador

✅ Benefícios do teste automatizado

- **Mais rápido** que testes manuais
- **Maior cobertura e repetibilidade**
- **Menos erros humanos**
- **Feedback rápido**
- **Melhor ROI**
- **Testes reutilizáveis e escaláveis**

❌ Limitações

- Nem todos os testes devem ser automatizados (ex: **exploratórios** e **visuais** são melhores manualmente)

🧠 Equívocos comuns sobre testes automatizados

- "Automação dá mais tempo livre" → na verdade, **realoca o esforço**
- "Automação é melhor que testes manuais" → **ambos se complementam**
- "Automação desestimula a colaboração" → **pode fortalecer a comunicação**

- "É caro" → o custo inicial compensa com o tempo
- "Scripts funcionam para qualquer build" → exigem **manutenção**

🏆 Boas práticas de automação

- Testar cedo e com frequência
- Escolher ferramentas adequadas
- Criar testes resistentes a mudanças na UI
- Separar testes automatizados por contexto
- Medir métricas (ex: taxa de falhas, tempo de execução)
- Automatizar a execução com alertas de falha

🔄 Testes contínuos com CI/CD

- Integra testes ao longo de todo o pipeline de desenvolvimento
- Permite **entregas frequentes**, **sem interrupções**
- CI: mudanças pequenas testadas automaticamente
- CD: código testado vai para staging ou produção automaticamente

🧪 Teste automatizado vs. Teste unitário

- **Testes unitários** focam na menor parte do código
- Podem ser **automatizados** para reduzir erros humanos e acelerar execuções

🧪 Teste automatizado vs. Teste manual

- Manual: mais observação e análise humana, **ideal para etapas subjetivas**
- Automático: repetição em larga escala, **ideal para testes longos e técnicos**
- Ambos são **complementares**, não substitutos

🟦 Tabela Resumo – Principais Conceitos e Aplicações:

Conceito	Detalhes
Definição	Teste de software automatizado com scripts e ferramentas
Áreas aplicáveis	Unit, API, UI, regressão, integração, sistema
Processos	Seleção de ferramenta → desenvolvimento → execução → manutenção
Frameworks comuns	Lineares, modulares, data-driven, keyword-driven, híbridos
Ferramentas	Selenium, Robotium, Cypress
Benefícios	Velocidade, repetibilidade, ROI, cobertura, menor erro humano
Limitações	Exploratórios e visuais ainda são melhores manualmente
Boas práticas	Testar cedo, usar dados externos, monitorar falhas, manter scripts atualizados

Equívocos	Custo, automação como substituto, “esforço zero”
CI/CD	Automatiza testes no fluxo de entrega contínua