**✅ 1. Testes Automatizados Avançados**

**📌 O que estudar:**

* **Design Patterns em automação**: Page Object Model (POM), Screenplay Pattern
* **Testes paralelos e distribuídos**: com ferramentas como Cypress Cloud, Playwright Test Runner, Selenium Grid
* **Integração com CI/CD**: GitHub Actions, GitLab CI, Jenkins
* **Testes com mocks e stubs**: interceptação de requisições (ex: cy.intercept)
* **Gerenciamento de dados de teste**: fixtures, factories, bancos de dados fake
* **Testes flakiness e estabilidade**: técnicas para reduzir falsos positivos/negativos

**✅ 2. Testes de API (nível básico/intermediário)**

**📌 O que estudar:**

* Métodos HTTP, status codes, headers
* Testes com **Postman** (coleções, variáveis, scripts)
* Introdução ao **RestAssured** (Java) ou **Supertest** (JavaScript)
* Automatização com **Newman** (CLI do Postman)
* Testes de contrato com **Swagger** ou **Pact**

**✅ 3. Testes com Foco em UX/UI (nível básico/intermediário)**

**📌 O que estudar:**

* **Heurísticas de Nielsen** aplicadas a testes exploratórios
* **Testes de usabilidade**: moderação, gravação de sessões, análise de comportamento
* **Ferramentas de prototipagem**: Figma, Adobe XD
* **Testes com protótipos**: validação antes do desenvolvimento

**✅ 4. Acessibilidade (nível básico)**

**📌 O que estudar:**

* Princípios do WCAG 2.1 (Perceptível, Operável, Compreensível, Robusto)
* Ferramentas: **axe DevTools**, **Lighthouse**, **NVDA**
* Testes manuais com teclado e leitores de tela
* Automatização de testes de acessibilidade com Cypress + axe-core

**✅ 5. Estratégia e Cultura de Qualidade (nível básico/intermediário)**

**📌 O que estudar:**

* Papel do QA em times ágeis (Scrum, Kanban)
* **TestOps** e qualidade contínua
* **BDD** com Gherkin + Cucumber
* Métricas de qualidade: cobertura, tempo de execução, bugs em produção

**✅ 6. Ferramentas e Práticas Complementares**

* **Versionamento com Git/GitHub**
* **Docker** (para ambientes de teste)
* **Testes em dispositivos reais**: BrowserStack, Sauce Labs
* **Testes mobile**: Appium (básico)

***--------------------------------------------------------------------------------------------------------------------------------***

***--------------------------------------------------Teste Avançado---------------------------------------------------------***

***--------------------------------------------------------------------------------------------------------------------------------***

**🚀 1. Performance e Testes de Carga (nível introdutório)**

Mesmo que não seja seu foco principal, entender o básico pode te diferenciar:

* Ferramentas: **JMeter**, **k6**, **Locust**
* Conceitos: throughput, latência, TPS, gargalos
* Casos de uso: testes de stress, carga, endurance

**🧩 2. Testes Baseados em Risco**

* Como priorizar testes com base em impacto e probabilidade de falha
* Técnicas de análise de risco (ex: matriz de risco)
* Aplicação prática em sprints e releases

**🧠 3. Pensamento Crítico e Análise de Requisitos**

* Como identificar inconsistências, ambiguidade e lacunas em histórias de usuário
* Técnicas de refinamento: **3 amigos (Dev, QA, PO)**, **exemplos concretos**
* Validação de critérios de aceitação com foco em valor de negócio

**🧪 4. Testes Baseados em Modelos (Model-Based Testing)**

* Criação de modelos de fluxo (ex: diagramas de estados, fluxogramas)
* Geração automática de casos de teste a partir de modelos
* Ferramentas: GraphWalker, TestCompass

**🧰 5. Ferramentas de Observabilidade e Logs**

* Como usar **logs, métricas e traces** para investigar bugs
* Ferramentas: **Grafana**, **Kibana**, **Datadog**, **Sentry**
* Monitoramento de qualidade em produção (shift-right testing)

**🧑‍🏫 6. Mentoria, Documentação e Compartilhamento**

* Criar **documentação de testes clara e acessível**
* Compartilhar conhecimento com o time (ex: tech talks, pair testing)
* Participar de comunidades de QA (ex: Ministry of Testing, Testers BR)

**🧭 7. Mentalidade de Qualidade Contínua**

* Atuar como **facilitador da qualidade**, não apenas executor de testes
* Participar desde o discovery até o deploy
* Promover cultura de **"testar cedo e frequentemente"**

***--------------------------------------------------------------------------------------------------------------------------------***

***-------------------------------------------------UX/UI Avançado---------------------------------------------------------***

***--------------------------------------------------------------------------------------------------------------------------------***

**🧠 1. Psicologia Cognitiva Aplicada ao Design**

* Teorias de percepção e atenção (ex: Lei de Hick, Lei de Fitts)
* Carga cognitiva e design de interfaces
* Tomada de decisão e comportamento do usuário

**📊 2. UX Research Avançado**

* Testes A/B e multivariados
* Análise de mapas de calor e gravações de sessão (Hotjar, Clarity)
* Entrevistas qualitativas e análise temática
* Card sorting e tree testing

**🧪 3. Testes de Usabilidade Profundos**

* Testes moderados vs. não moderados
* Testes remotos e com protótipos de alta fidelidade
* Análise heurística avançada (além de Nielsen, como Gerhardt-Powals)
* Testes com usuários com deficiência (acessibilidade real)

**🎨 4. Design Systems e Componentização**

* Atomic Design
* Criação e manutenção de Design Systems (ex: Material, Carbon, Fluent)
* Tokens de design e integração com código (Figma Tokens, Style Dictionary)

**🧩 5. UX Writing e Microcopy**

* Escrita orientada à ação (CTA eficaz)
* Tom de voz e consistência textual
* Testes A/B de microcopy

**📐 6. Arquitetura da Informação e Navegação**

* Design de fluxos complexos (ex: onboarding, checkout, dashboards)
* Hierarquia visual e escaneabilidade
* Design responsivo e mobile-first

**📈 7. Métricas e KPIs de UX**

* SUS (System Usability Scale)
* NPS, CSAT, CES
* Task success rate, time on task, error rate
* Ferramentas: Google Analytics, Mixpanel, Amplitude

**🧰 8. Ferramentas e Integrações Avançadas**

* Figma: auto layout, variantes, protótipos interativos complexos
* Plugins úteis: Stark (acessibilidade), FigJam (workshops)
* Integração com Devs: Figma → Storybook → Front-end