

Fundamentos de automação de testes

Carolina Santana Louzada

Analista QA - Venturus

Objetivo do curso

Introduzir conceitos importantes relacionados à automação de testes, entendendo seu propósito e os fatores essenciais para o sucesso de sua implantação juntamente ao ciclo de vida do software.

Pré-requisitos

- Fundamentos de qualidade de software
- Ciclo de Desenvolvimento de Software e Metodologias Ágeis

Percurso

Aula 1

Fundamentos para projetos de automação

Aula 2

Arquitetura genérica e design para automação

Aula 3

Padrões de projeto para testes E2E

Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

Aula 1

Fundamentos para projetos de automação

// Fundamentos de automação de testes

Objetivos

1. O propósito da automação no processo de teste
2. Fatores de sucesso para automação
3. A importância de ferramentas para suporte a testes automatizados

Aula 1 . Etapa 1

O propósito da automação no processo de teste

// Fundamentos de automação de testes

Revisando o ciclo de vida do teste



A automação como atividade de teste

→ Processo de design de um *testware*:

- ◆ Software
- ◆ Documentação
- ◆ Ambientes de teste
- ◆ Dados de teste

A automação como atividade de teste

→ Atividades de testes relacionadas:

- ◆ Implementação de casos de testes automatizados
- ◆ Monitoramento e controle de execução
- ◆ Interpretação, relatório e log de resultados

Objetivos da automação

- ★ Melhorar a eficiência do testes
- ★ Ampliar cobertura de testes
- ★ Reduzir custo total de teste
- ★ Diminuir tempo de execução
- ★ Aumentar frequência de teste

Vantagens da automação

- ★ Testes que não podem ser feitos de maneira manual
- ★ Complexidade pode aumentar -> tempo diminui
- ★ Reduz erros humanos na validação
- ★ Feedback mais rápido
- ★ Maior confiabilidade

Desvantagens da automação

- ★ Custos adicionais
- ★ Investimento inicial para configuração da arquitetura de testes
- ★ Investimento na arquitetura do software sob teste
- ★ Complexidade de codificação
- ★ Manutenção de testes

Aula 1 . Etapa 2

Fatores de sucesso para automação

// Fundamentos de automação de testes

Algumas reflexões...

- ★ Nem todos os testes podem ou devem ser automatizados!
- ★ A automação de testes não substitui os testes manuais!
- ★ A automação é limitada pela análise do sistema. A subjetividade do analista/engenheiro é importante!
- ★ Script de teste automatizado também é código assim como qualquer outro software, portanto, um bom analista deve se preocupar em um código limpo e com boas práticas!

Arquitetura do Projeto de Automação

- A arquitetura de um projeto de automação deve estar alinhada com a arquitetura do produto de software.
- Lembrar que existem requisitos não-funcionais para testes.
- Envolvimento com desenvolvedores do projeto é essencial para o sucesso da automação.

Testabilidade

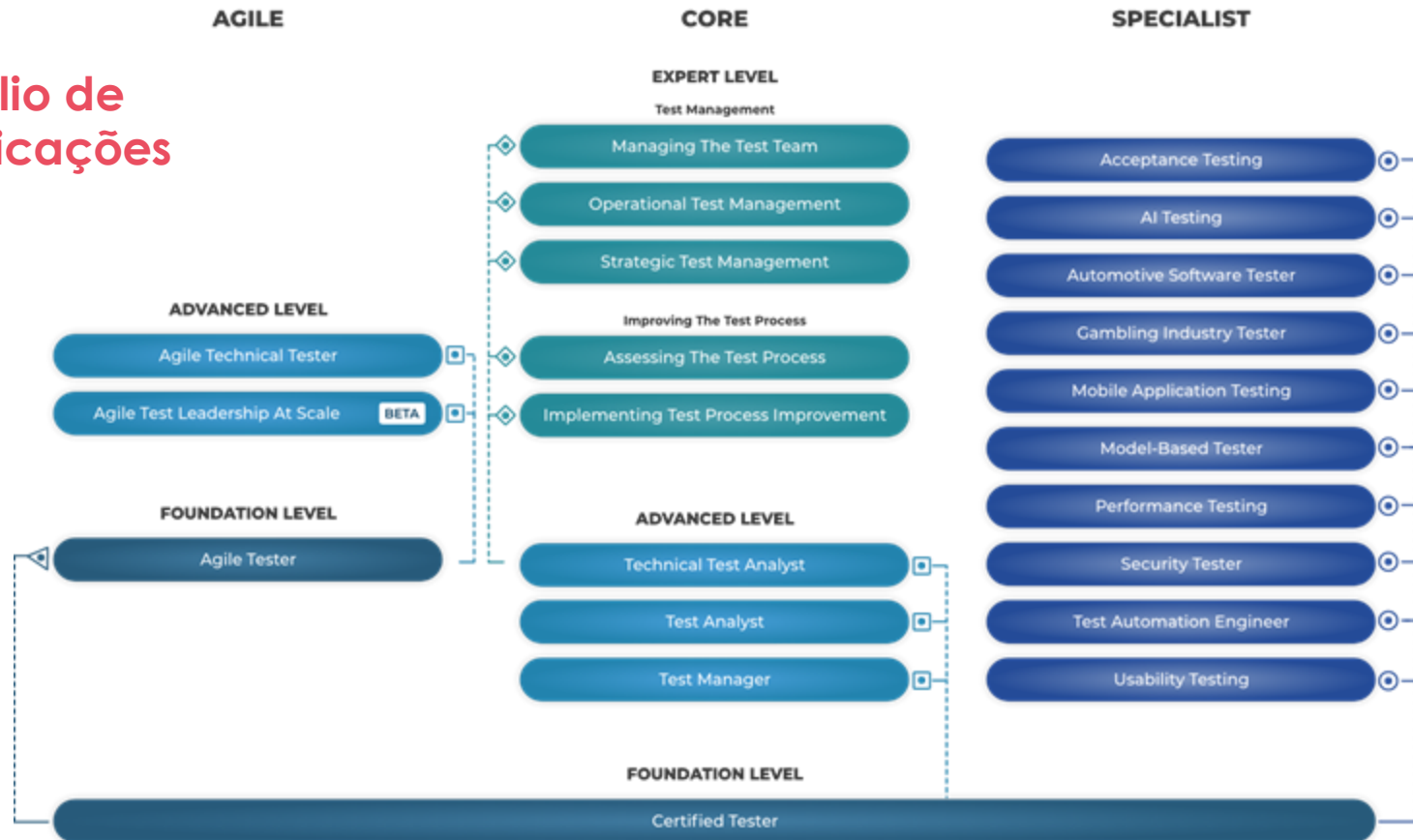
- O produto de software deve ser construído deve ser modelado pensando no suporte aos testes automatizados.
 - ◆ Desacoplamento entre camadas
 - ◆ Expor módulos/funções/classes para facilitar testes
- Identificar e analisar módulos que são testáveis faz parte do processo de decisão de automação.

Estratégias de testes

- Identificar o conjunto, tipos e níveis de testes automatizados que garantem consistência e cobertura do sistema.



Portfólio de Certificações



[Certifying Software Testers Worldwide - ISTQB®](#)
[International Software Testing Qualifications Board](#)

START HERE

Aula 1 . Etapa 3

Ferramentas de suporte a testes automatizados

// Fundamentos de automação de testes

Uso de Framework de Automação de Teste

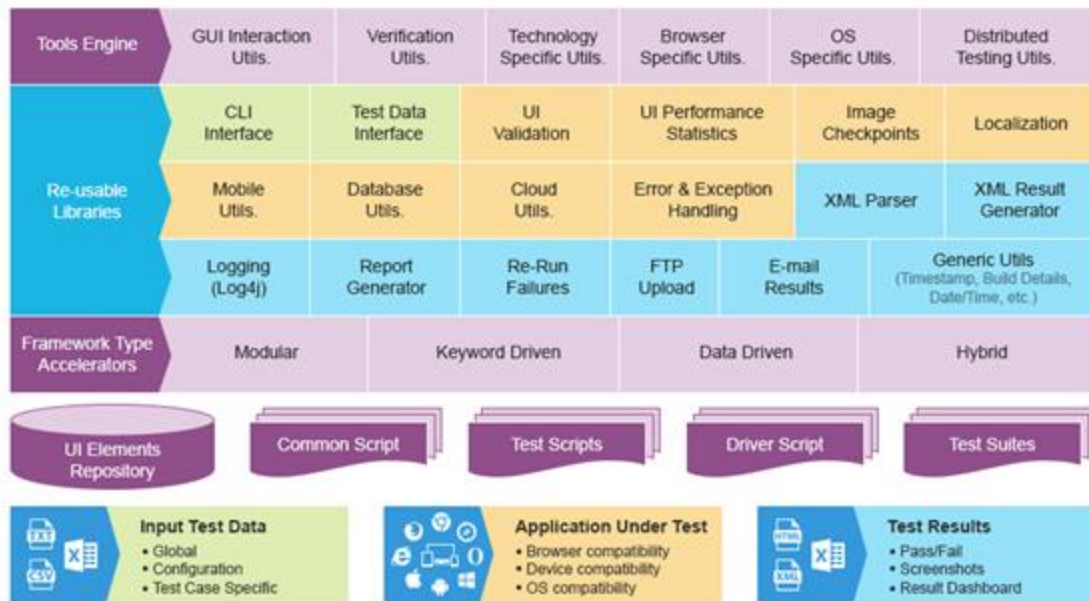
→ Avaliar um bom framework garante benefícios para:

- ◆ implementar relatórios de testes
- ◆ facilitar resolução de problemas
- ◆ facilitar configuração de ambiente
- ◆ documentar casos de teste
- ◆ rastrear etapas dos casos de teste
- ◆ facilitar manutenção dos testes
- ◆ atualização dos testes automatizados
- ◆ monitorar e restaurar execução dos testes em caso de falha

Automation Tools



Third Party Tools and Frameworks



Aula 2

Arquitetura genérica de automação

// Fundamentos de automação de testes

Objetivos

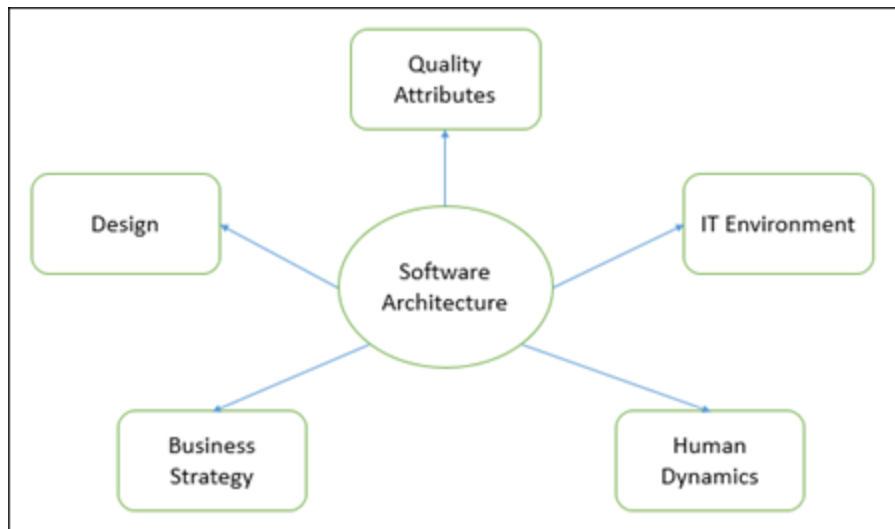
- 1 - Conceitos sobre arquitetura de software
- 2 - Arquitetura genérica para projetos de automação
- 3 - O que considerar antes de iniciar um projeto?
- 4 - Abordagens para automação de casos de teste

Aula 2 . Etapa 1

Revisitando conceitos de arquitetura de software

// Fundamentos de automação de testes

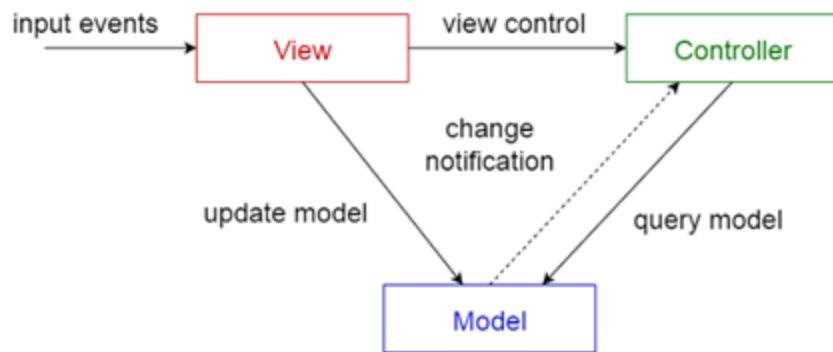
O que é arquitetura de software?



- ★ Descrição e compreensão dos componentes do sistema, suas estrutura e relações.
- ★ Considerar fatores externos na definição da arquitetura de um sistema.

Pensando nos padrões arquiteturais

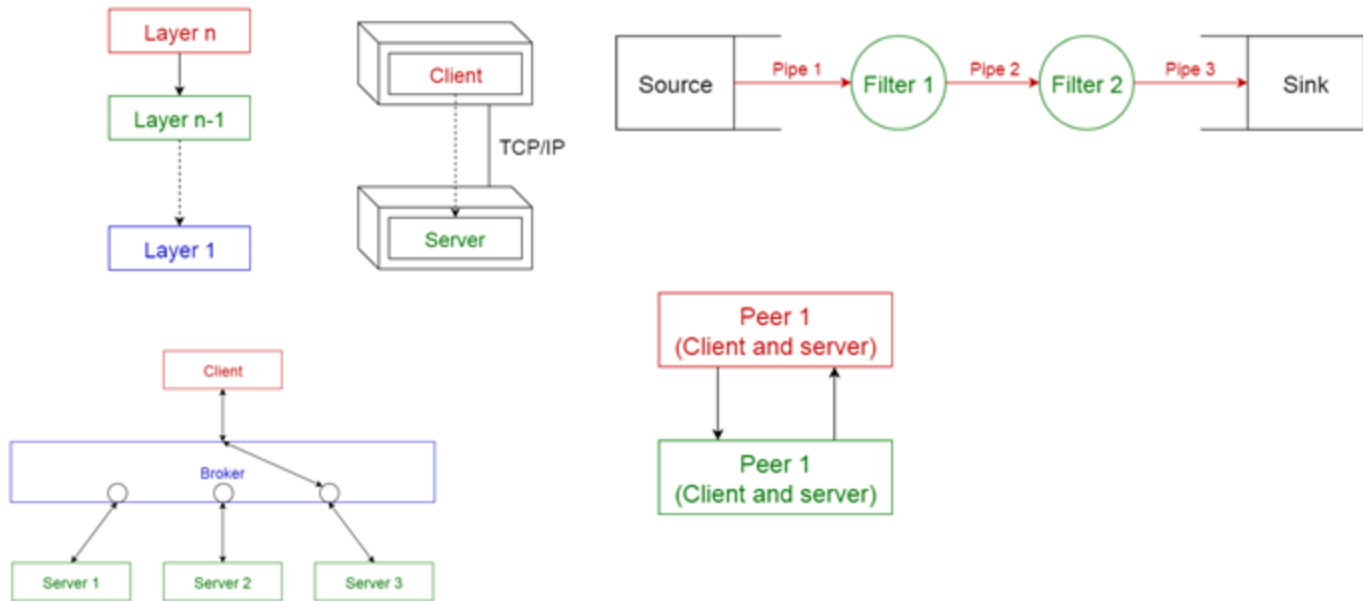
- ★ São soluções gerais e reutilizáveis para resolver problemas recorrentes dentro de certo contexto.
- ★ Estão inseridos em um escopo maior do sistema.



Padrão arquitetural != Padrão de projeto

Pensando nos padrões arquiteturais

- N-camadas
- Cliente-servidor
- Broker
- Peer-to-peer
- Event bus
- Pipe-filter



E os padrões de projeto?

- ★ São soluções gerais e reutilizáveis para problemas comuns relacionados à construção do projeto de software independente da linguagem.
- ★ São divididos em:
 - padrões de criação: *factory method, abstract factory, singleton...*
 - padrões estruturais: *adapter, composite, facade, proxy...*
 - padrões comportamentais: *command, observer, mediator...*

[O que é um padrão de projeto? \(refactoring.guru\)](https://refactoring.guru/)

SOLID

- ★ **Responsabilidade única:** Cada componente deve ter uma única responsabilidade e esta deve ser encapsulada.
- ★ **Extensão:** Cada componente deve estar aberto para extensão, mas fechado para modificação.
- ★ **Substituição:** Cada componente deve ser substituível sem afetar o comportamento geral da solução.

SOLID

- ★ **Segregação:** É melhor ter componentes específicos do que um componente geral multiuso.
- ★ **Inversão de dependência:** Os componentes devem depender de abstrações e não de detalhes de mais baixo nível.

[O que é SOLID: O guia completo para você entender os 5 princípios da POO | by João Roberto da Paixão | Desenvolvendo com Paixão | Medium](#)

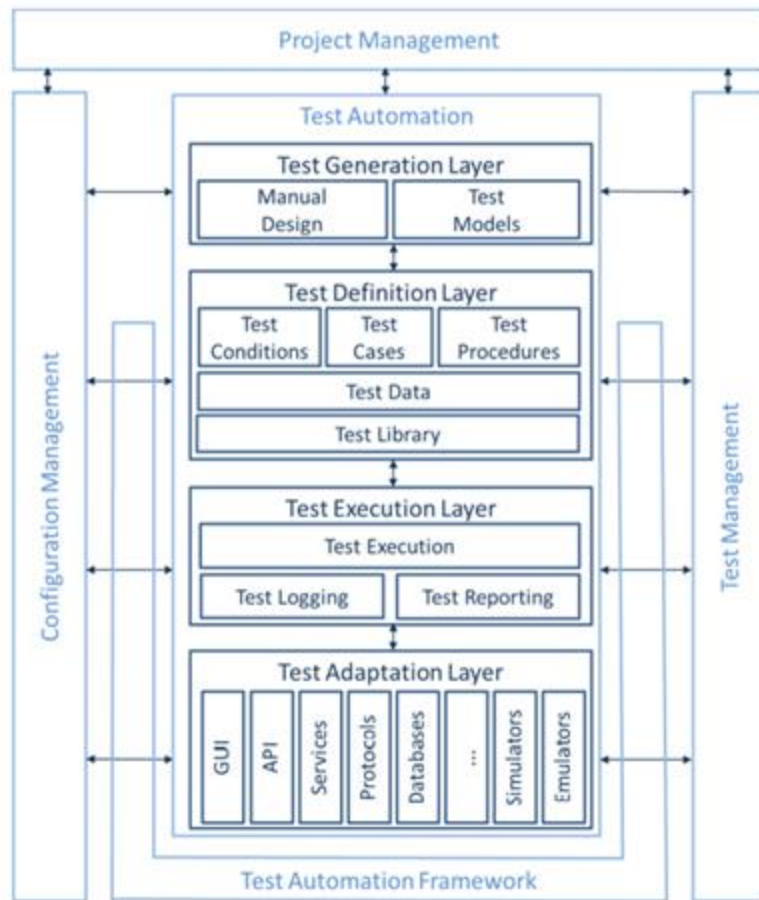
[Amazon.com.br eBooks Kindle: Código Limpo, Martin, Robert C.](#)

Aula 2 . Etapa 2

Arquitetura do projeto de automação

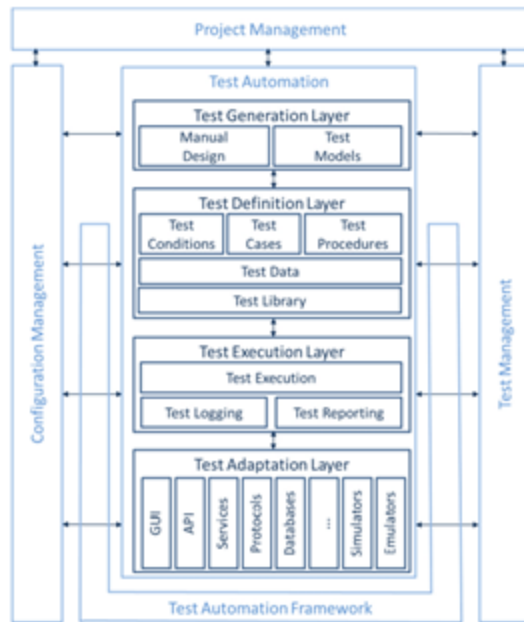
// Fundamentos de automação de testes

Uma arquitetura genérica para automação



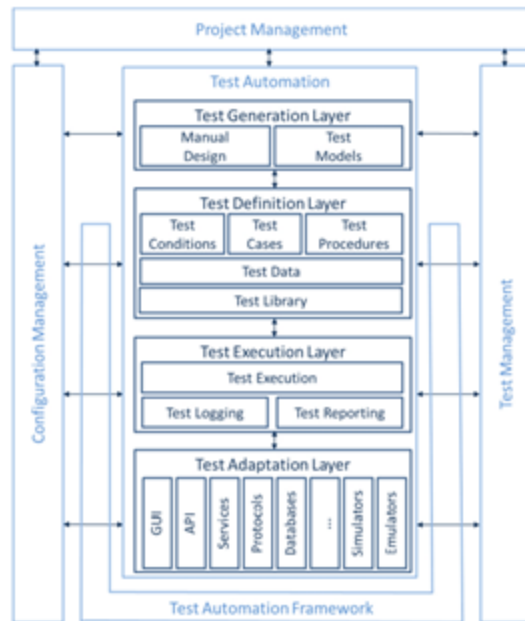
Camada de geração de teste

- ★ Criação manual de casos de teste
- ★ Desenvolvimento, captura ou obtenção de dados de teste
- ★ Geração automática de casos de teste a partir de modelos



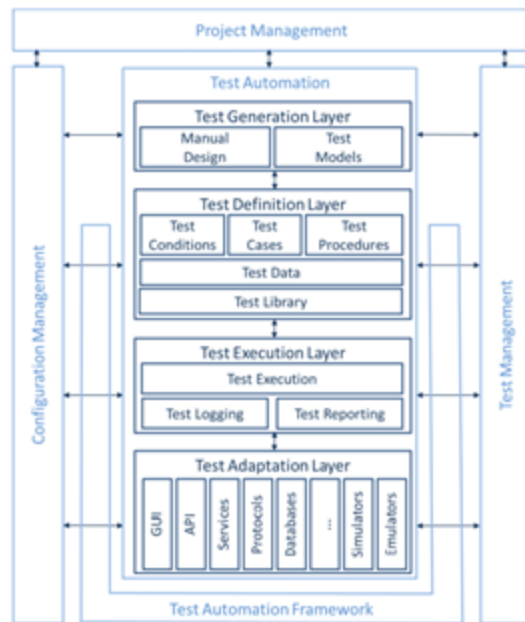
Camada de definição de teste

- ★ Especificação de casos de teste
- ★ Definição de dados de teste
- ★ Especificação de procedimento de testes
- ★ Definição de scripts para execução dos casos de teste
- ★ Conceder acesso às bibliotecas de teste conforme necessário



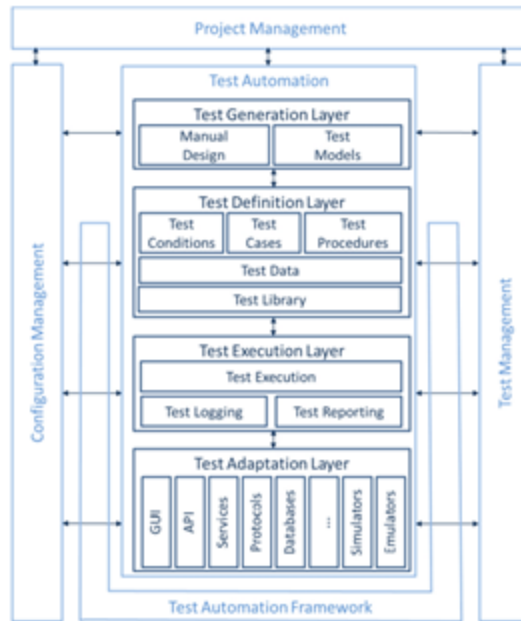
Camada de execução de teste

- ★ Executar casos de teste automaticamente
- ★ Registrar execuções
- ★ Relatar execuções



Camada de adaptação de teste

- ★ Controlar ambiente de teste
- ★ Interagir com sistema a ser testado
- ★ Monitorar o sistema sendo testado
- ★ Simular ou emular o ambiente sendo testado



Gerenciamento de projeto de solução de testes automatizados

- ★ Prestar atenção ao gerenciamento de configuração do projeto
- ★ Versionamento deve seguir boas práticas
- ★ A extração de métricas deve ser analisada da mesma forma que em uma abordagem manual

Aula 2 . Etapa 3

O que devemos considerar antes de iniciar um projeto?

// Fundamentos de automação de testes

No que devemos pensar ao iniciar um projeto de automação?

- ★ Qual atividade ou fase do processo de testes queremos automatizar?
- ★ Qual nível de teste queremos ter suporte?
- ★ Qual o tipo de teste?
- ★ Quem executará e implementará o teste?
- ★ Quais projetos existem gratuitos e quais podem ser comprados?
- ★ Quais tecnologias queremos ter suporte? E a complexidade?
- ★ Que linguagens queremos utilizar?

No que devemos pensar ao iniciar um projeto de automação?

- ★ Qual modelo de ciclo de vida do software estamos inseridos?
- ★ Qual a complexidade de implementação e como afeta o projeto?
- ★ Será necessário treinamento?

Aula 2 . Etapa 4

Abordagens para automação

// Fundamentos de automação de testes

Abordagens para automação de casos de teste

- Sabemos que os casos de testes correspondem à sequência de ações executadas sobre o sistema sendo testado.
- Abordagens:
 - 1 .A solução implementa casos de teste diretamente em scripts de teste automatizados.

Abordagens para automação de casos de teste

- Sabemos que os casos de testes correspondem à sequência de ações executadas sobre o sistema sendo testado.
- Abordagens:
 1. A solução projeta procedimentos de teste e os transforma em scripts automatizados.
 2. A solução projeta procedimentos de teste e os transforma em scripts automatizados.

Abordagens para automação de casos de teste

- Sabemos que os casos de testes correspondem à sequência de ações executadas sobre o sistema sendo testado.
- Abordagens:
 1. A solução utiliza uma ferramenta para gerar os casos de teste.
 2. A solução utiliza uma ferramenta para gerar os casos de teste a partir de uma especificação.
 3. A solução utiliza uma ferramenta para traduzir os procedimentos de teste em scripts de teste automatizados.

Abordagens para automação de casos de teste

- Sabemos que os casos de testes correspondem à sequência de ações executadas sobre o sistema sendo testado.
- Abordagens:
 - 4. A solução utiliza uma ferramenta que gera procedimentos automatizados de teste ou traduz os scripts diretamente dos modelos. Maior grau de automação.

Abordagem de captura e reprodução

- ★ As ferramentas são usadas para capturar a interação com o sistema enquanto executa a sequência de ações, registrando as saídas. A verificação da saída pode ser manual ou automatizada.
 - o Prós: Fácil de usar e configurar
 - o Contras: Difícil de evoluir e manter



Geração de Scripts lineares

- ★ As ferramentas são usadas para capturar a interação com o sistema enquanto executa a sequência de ações, registrando as saídas e gerando scripts de testes lineares.
 - Prós: Fácil uso e configuração
 - Contras: Difícil de evoluir e manter

Geração de Scripts estruturados

- ★ Geração de bibliotecas de scripts reutilizáveis para os casos de teste.
 - Prós: Menor manutenção e menor custo para novos testes.
 - Contras: Custo maior para criação dos scripts compartilhados. Boas habilidades de programação.

Orientado a dados

- ★ Abordagem na qual os scripts utilizam arquivos com diversas massas de dados.
 - Prós: Reduz custo para adição de testes. Aumenta variações com dados de testes.
 - Contras: Gerenciar bem os arquivos a serem lidos pela solução.

Uso de palavras-chaves

- ★ O script baseado em palavras-chaves utiliza arquivos de definições a partir de palavras mais descritivas além da possibilidade de uso de arquivos de dados
 - **Prós:** Reduz custo para adição de testes. Aumenta variações com dados de testes.
 - **Contras:** A implementação das palavras-chaves vem com custo. Boas palavras-chaves são mais reutilizadas.

Testes baseado em modelos

- ★ Geração automatizada de casos de teste em oposição à execução de casos de testes. Pode derivar qualquer tipo de script.
 - **Prós:** Abstração garante foco no objetivo do teste. Manutenção somente nos modelos.
 - **Contras:** É necessário bom conhecimento para modelos de testes além de garantir que as ferramentas de geração sejam maduras e gerem bons relatórios.

Orientado a comportamento (BDD)

- ★ Uso de linguagem natural para definir o comportamento da funcionalidade.
 - **Prós:** Maior compreensão de negócio. Facilidade na criação da modelagem dos casos.
 - **Contras:** Ambiguidade. Pode gerar código repetido se não for bem pensado.

Aula 3

Padrões de projeto para automação E2E

// Fundamentos de automação de testes

Objetivos

- 1 - Conhecer o Page Object Model
- 2 - Conhecer o Business-layer Page Object
- 3 - Conhecer o ScreenPlay Pattern
- 4 - Conhecer o App Actions

Aula 3 . Etapa 1

Conhecendo o Page Object Model

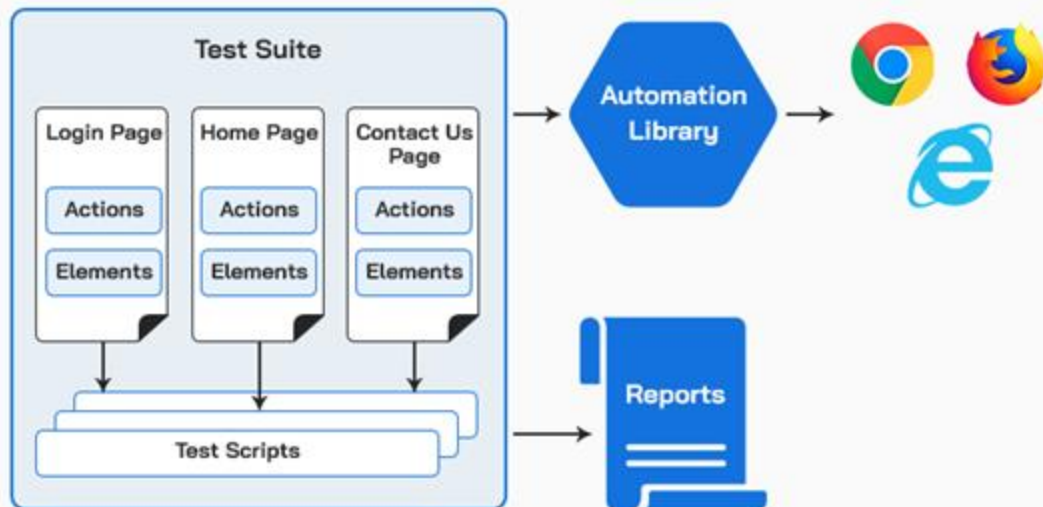
// Fundamentos de automação de testes

Padrões de projetos para testes?

- ★ O conhecimento sobre padrões de projeto é muito importante quando precisamos lidar com projetos de alta complexidade.
- ★ Os padrões voltados para testes servem para resolver problemas comuns relacionados à coerência, coesão, isolamento e responsabilidades dos testes.

Vamos conhecer ou revisar padrões de projetos de testes funcionais E2E?

POM - Page Object Model



this API is about the application

```
selectAlbumWithTitle()  
getArtist()  
updateRating(5)
```

Page Objects

Album Page

Album List Page

this API is about HTML

```
findElementsWithClass('album')  
findElementsWithClass('title-field')  
getText()  
click()  
findElementsWithClass('ratings-field')  
setText(5)
```

HTML Wrapper

title: Whiteout
artist: In the Country
rating:

title: Ouro Negro
artist: Moacir Santos
rating:

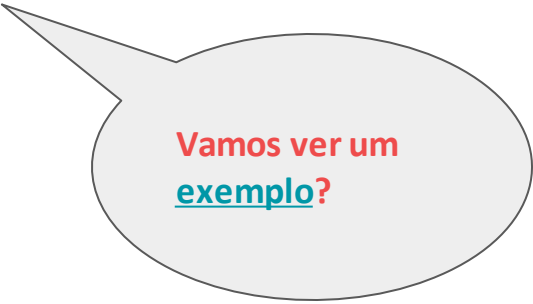
POM - Page Object Model

→ Vantagens:

- ◆ Reaproveitamento de código
- ◆ Código mais limpo
- ◆ Facilidade de manutenção
- ◆ Independência de testes

→ Desvantagens

- ◆ Necessária boa compreensão de programação e do padrão em si
- ◆ Elementos ficam isolados a uma página específica
- ◆ É necessária uma boa modelagem, caso contrário as refatorações serão constantes
- ◆ Projetos complexos demandam mais tempo para automação



Vamos ver um
exemplo?

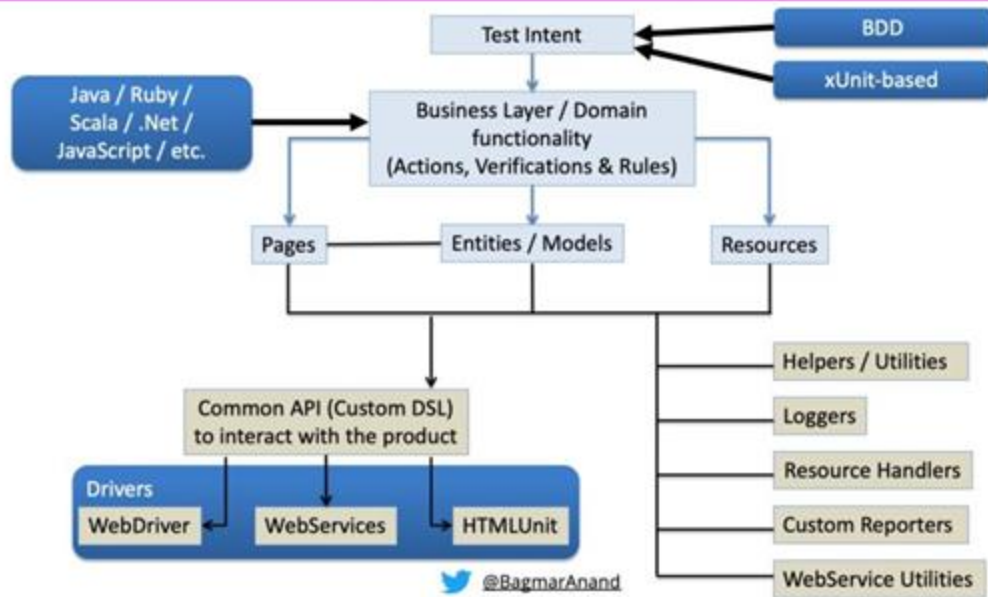
Aula 3 . Etapa 1

Conhecendo o Business-Layer Page Object

// Fundamentos de automação de testes

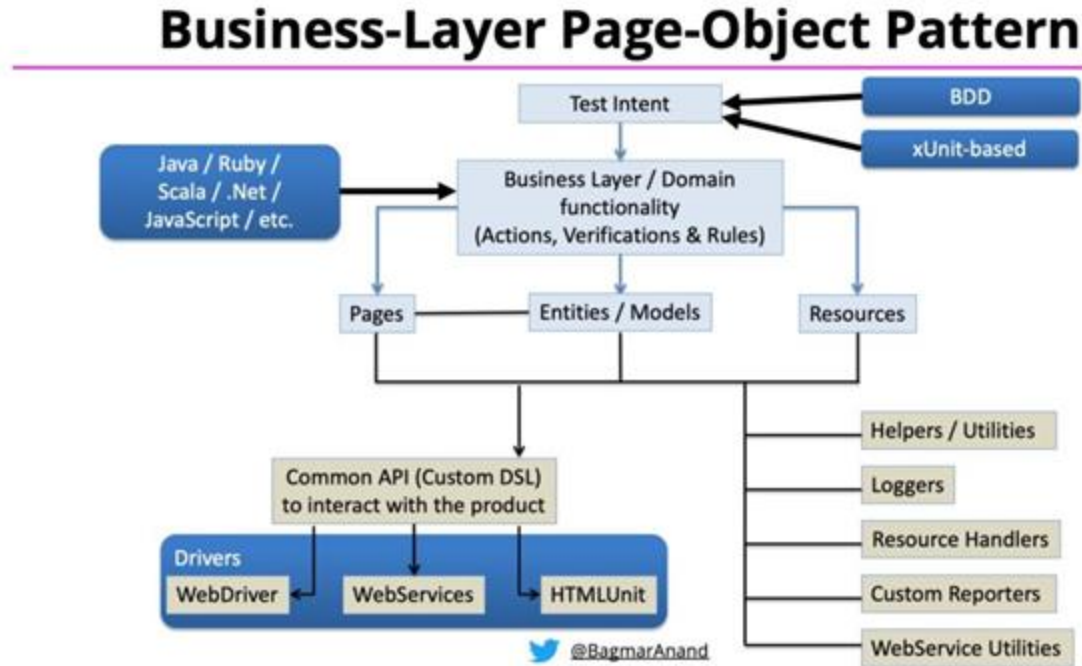
Business-Layer Page Object Pattern

Business-Layer Page-Object Pattern



Vamos ver um [exemplo?](#)

Business-Layer Page Object Pattern



Vamos ver um [exemplo?](#)

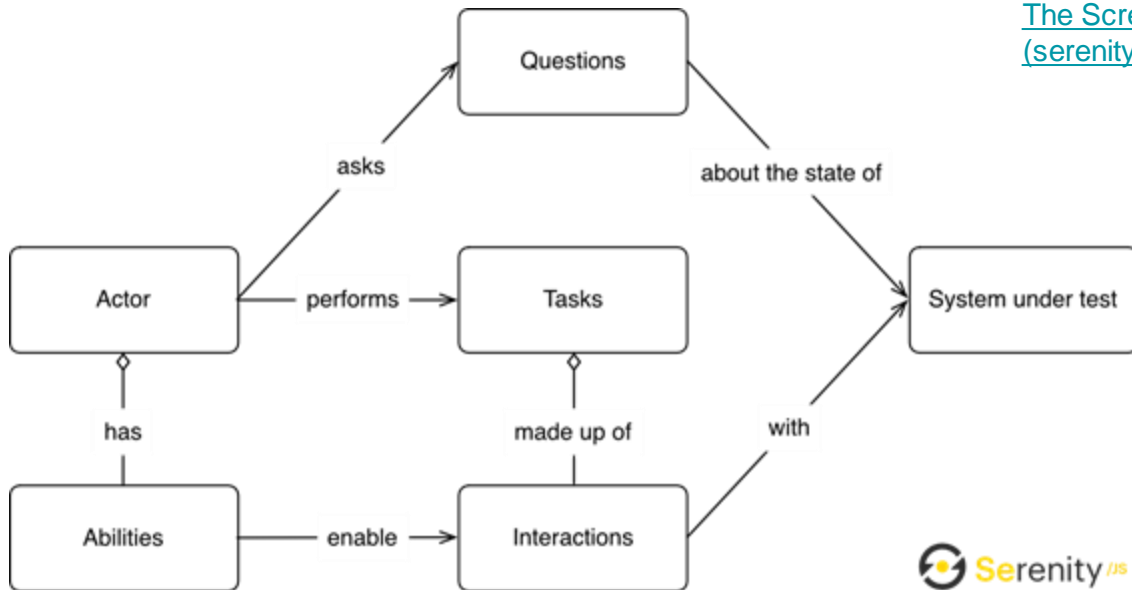
Aula 3 . Etapa 3

Conhecendo o ScreenPlay Pattern

// Fundamentos de automação de testes

ScreenPlay Pattern

[The Screenplay Pattern | Serenity/JS](https://serenity-js.org)
(serenity-js.org)



Aula 3 . Etapa 4

Conhecendo o App Actions

// Fundamentos de automação de testes

App Actions

- Objetivo do App Actions é utilizar a Arquitetura do Cypress a favor do desacoplamento entre a camada da aplicação e o HTML.
- Utiliza-se o acesso à própria aplicação para configurar o estado do teste sem depender da UI, melhorando ,dessa forma, a performance.

[Page objects vs. App actions in Cypress - Automated Visual Testing | AppliTools](#)

[Stop using Page Objects and Start using App Actions \(cypress.io\)](#)

Fundamentos de automação de testes

Carolina Santana Louzada

Analista QA - Venturus

Percurso

Aula 1

Fundamentos para projetos de automação

Aula 2

Arquitetura genérica e design para automação

Aula 3

Padrões de projeto para testes E2E

Para saber mais

<https://www.istqb.org/certifications/certification-list>

<https://www.primesoft.net/portfolio-items/test-automation/>

[O que é um padrão de projeto? \(refactoring.guru\)](#)

[Design Patterns in Test Automation - TestProject](#)

[PageObject \(martinfowler.com\)](#)

Dúvidas durante o curso?

- > Fórum do curso
- > LinkedIn: Carolina Santana Louzada
- > Comunidade online (Discord)

