

AVALIAÇÃO PRÁTICA – QA SÊNIOR (Automação de API – Java + RestAssured | Postman - Manual)

Orientação

- Criar testes automatizados de API em **Java + RestAssured**
 - Criar coleção de testes em **Postman** e executá-la com **Newman**
 - Documentar corretamente o projeto (README)
 - Aplicar boas práticas de testes, organização, versionamento e estrutura de código
 - Link sugerido ou de livre escolha <https://jsonplaceholder.typicode.com/jsonplaceholder.typicode.com+2GitHub+2>
 - Também há a documentação da API mais moderna: **JSONPlaceholder.dev** — <https://jsonplaceholder.dev/jsonplaceholder.dev+1>
-

ATIVIDADE PRÁTICA – Automação de API

Tecnologias exigidas:

Java (RestAssured)

Postman (Collection + Environment)

✓ Newman (CLI + relatório JUnit) - diferencial

API alvo:

https (JSONPlaceholder API)

Endpoints usados (com base no site acima):

- /posts
- /posts/{id}
- /users

Automação com Java + RestAssured

1. Criar projeto Maven

Requisitos obrigatórios:

- Usar RestAssured com RequestSpecification
- Usar JUnit ou TestNG
- Criar POJO ou Map para payloads
- Reutilização de código (BaseTest) se possível

Criar e automatizar cenários obrigatórios

1. GET /posts — validar estrutura do retorno

Critérios:

- Status 200
- Retorno é lista com 100 posts
- Validar campos: userId, id, title, body [exemplo]

2. GET /posts/{id} — validar dado específico

Testar:

- /posts/1
 - Status 200
 - Campo id == 1
 - userId e title não podem ser nulos **[exemplo]**
1. Validar Contrato JSON (JSON Schema Validator)
 2. Criar schema post-schema.json:
 3. Validações mínimas:
 - id inteiro
 - userId inteiro
 - title string
 - body string

Acima é apenas um modelo segundo o site sugerido, mas QA pode usar da melhor forma, dentro dos padrões obrigatórios já informados

Automação no Postman

Tarefa 2 — Criar coleção Postman

Coleção:

JSONPlaceholder_API.postman_collection.json [sugerido]

Cenários obrigatórios:

1. GET /posts
2. GET /posts/{id}

Validações com pm.expect:

- Status code
- Campos obrigatórios
- Tipos dos campos
- Valor do campo id ao consultar post específico

Regras:

- Criar variável de ambiente postId e usá-la nos testes
- Incluir scripts no “Tests” (não apenas no “Pre-Request”)

Tarefa 3 - Manual

Criar e documentar os principais cenários encontrados

Modelar os cenários encontrados, visando que outra pessoa usaria para replicá-los

Nota importante: Seguindo conforme orientação acima, o ambiente acima é uma sugestão, porém pode utilizar o site ou o ambiente que se sentir à vontade.

1. Disponibilizar o perfil do github em modo público ou compartilhar com o email: isabele.aquino@finaya.tech
2. Caso encontre algum bug, registrar em um documento pontuando o erro e uma possível sugestão de correção.
3. O Objetivo é analisar tecnicamente, então ao iniciar este teste, reserve um lugar tranquilo
4. Total de dias para entregar: 04