

AUTOMAÇÃO DE TESTES EM API REST

Objetivo: Capacitar os Alunos a se tornarem proficientes na automação de testes de APIs Rest ao compreenderem e aplicarem os princípios e práticas fundamentais para automatizar testes em APIs Rest.

Prof. MSc. Taynara Luana



Ementa

API	API - ESTRUTURA	AUTOMAÇÃO DE TESTES DE API	ENCERRAMENTO
<ul style="list-style-type: none">• Introdução a APIs• Estilo arquitetural REST	<ul style="list-style-type: none">• Métodos• Endpoints• JSON• Requisições• Respostas• Status Code	<ul style="list-style-type: none">• Fundamentos e Conceitos sobre automação de testes para APIs REST• Bibliotecas, primeiro código de testes de API• Logs e métodos• Headers e autenticações• Asserções e validações da estrutura da resposta em JSON• Boas práticas no desenvolvimento de testes automatizados	<ul style="list-style-type: none">• Quais forças externas podem impactar negativamente seu negócio?• Há empresas com potencial de se tornar concorrentes?

CALENDÁRIO

AULA 1

CONCEITOS API

- Definição de API;
- Papel no desenvolvimento de software;
- Impacto na maneira como o software é construído nos dias de hoje.

EXPLICAÇÃO PRÁTICA

- Como uma API FUNCIONA NA PRÁTICA?
- Fundamentos de API;
- Estilo arquitetural REST;
- Métodos HTTP no REST.

PROJETO PRÁTICO

- Apresentação das ferramentas que serão utilizadas durante a disciplina;
- Instalações necessárias;
- Conhecimento básico necessário;
- Executando o projeto;
- Primeiro teste de API utilizando POSTMAN.

O Papel do Analista de Qualidade no Desenvolvimento de Aplicações com APIs



01

crescimento do mercado e novas tecnologias

02

desenvolvimento de aplicações com a utilização de APIs

03

papel do analista de qualidade de software

04

mercado de Testadores de Software com Foco em Testes Automatizados de APIs

O que é uma API REST?

- Avanço do mercado digital;
- Desenvolvimento de novas plataformas de software;
- Plataformas Web.

Essas plataformas web, são divididas em **componentes**.

Por sua vez, estes componentes são conhecidos como **componentes de serviços web**. Que por sua vez, são conhecidos como **microserviços**.



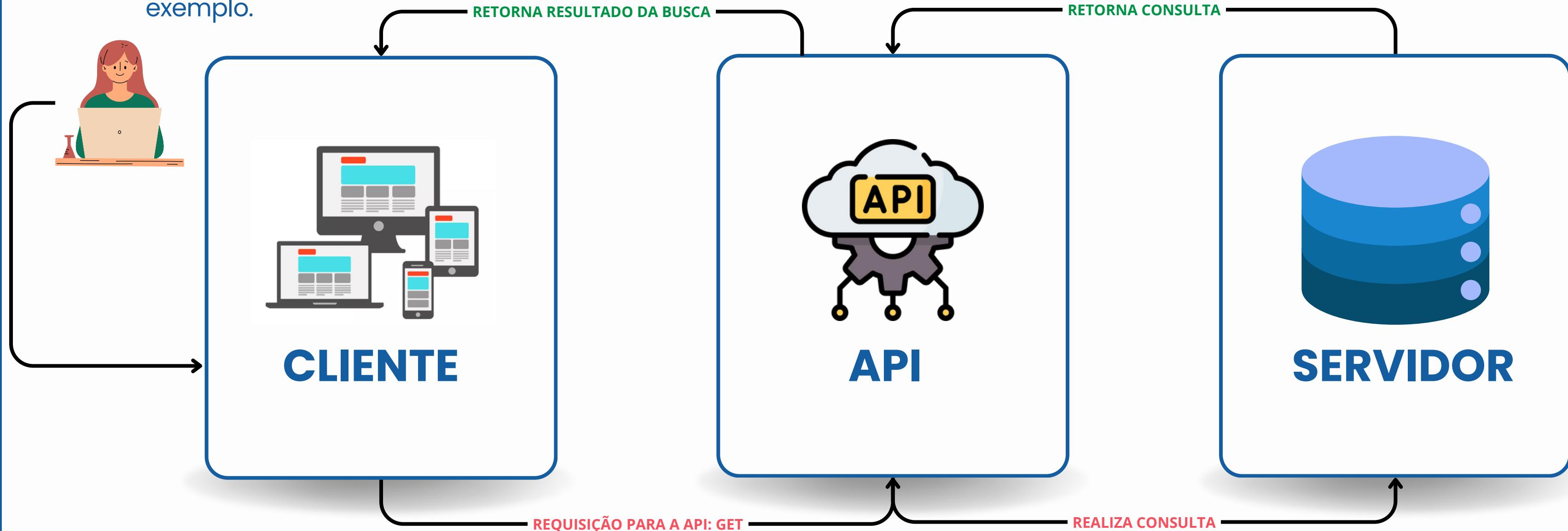
O conceito de API (Interface de Programação de Aplicações) é fundamental no desenvolvimento de software, pois permite que diferentes sistemas e aplicações se comuniquem entre si de forma eficiente e segura.

Um serviço WEB é construído de uma forma que suporte a interoperabilidade entre diferentes ambientes em uma rede na qual é baseada em padrões e protocolos WEB.

Por exemplo

Pesquisa um determinado produto no site da amazon por exemplo.

GET - Consultar produtos por filtro nome



COMO FUNCIONA ISSO TUDO NA PRATICA?

front end

Essa é a camada com a qual o usuário **interage diretamente**, geralmente em um **navegador web ou aplicativo móvel**.

Envia solicitações para o back-end quando o usuário interage com a interface.



backend

Aqui fica - (**Lógica da Aplicação**): Esta camada processa as solicitações recebidas do front-end e **interage com o banco de dados** para obter ou salvar dados.

Interação com a base de dados para operações crud.



base de dados

Armazena todos os dados necessários para a aplicação. **O back-end se conecta a este banco de dados para recuperar ou armazenar informações conforme necessário.**



Sobre

A requisição é feita por meio de uma URL que aponta para o endpoint da API REST, incorporando operações CRUD, tais como (GET, POST, PUT, DELETE), juntamente com seus respectivos parâmetros.

Imagine que temos uma API REST para gerenciamento de usuários. Os endpoints e métodos suportados são:

- **GET /users:** Retorna uma lista de todos os usuários.
- **POST /users:** Cria um novo usuário.
- **GET /users/{id}:** Retorna os detalhes de um usuário específico.
- **PUT /users/{id}:** Atualiza os detalhes de um usuário específico.
- **DELETE /users/{id}:** Exclui um usuário específico.



APRESENTAÇÃO DA API

Durante nossa disciplina, faremos os testes em uma API simples porém com muitos insights de melhorias, segurança, construção e arquitetura de testes agregada.

A aplicação consiste em um sistema simples desenvolvido utilizando HTML, CSS, JavaScript, NodeJs + express e sqlite.

Teremos um processo de autenticação de usuários na APLICAÇÃO.
E contaremos com algumas operações simples de crud para os cadastros de:

- Usuários;
- Conteúdos;
- Atividades.

WELCOME TO TL CONTENT

Aplicação será apresentada na prática

RECURSOS E ENDPOINTS

Principais conceitos sobre recursos e endpoints de uma API REST

RECURSO

Um recurso pode ser qualquer **entidade** ou **objeto** que você deseja expor através da API, como usuários, tarefas ou conteudos.

Por exemplo:

Recurso: Usuário

- **GET /users:** Retorna uma lista de todos os usuários.
- **POST /users:** Cria um novo usuário.
- **GET /users/{id}:** Retorna os detalhes de um usuário específico.
- **PUT /users/{id}:** Atualiza os detalhes de um usuário específico.
- **DELETE /users/{id}:** Exclui um usuário específico.

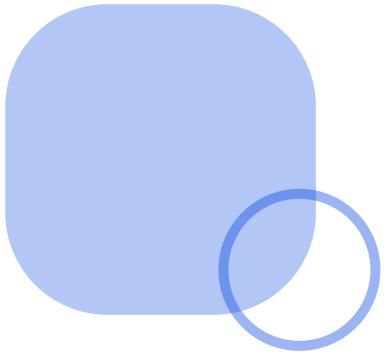
ENDPOINT

Se refere a uma **URL** em específico utilizada para acessar ou manipular esse recurso.

- **<https://localhost:3000/users>:** Retorna uma lista de todos os usuários.

FORMATOS E REPRESENTAÇÕES DE DADOS

JSON: Mais comum seu uso e escolha, devido a sua simplicidade e facilidade.



```
{  
    "id": 2,  
    "nome": "TESTE",  
    "telefone": "4495345345349707070",  
    "email": "teste@gmail.com"  
}
```

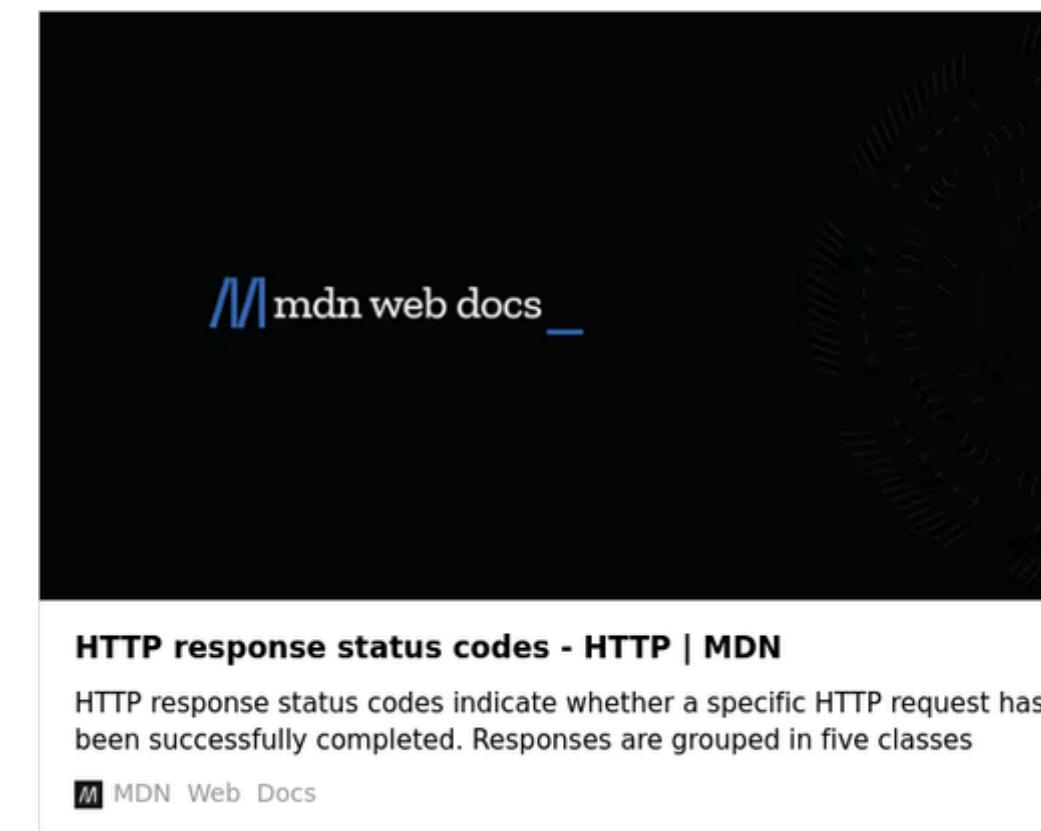
XML: também é um formato de representação de dados porém menos popular.

```
<book>  
    <id>123</id>  
    <nomeTeste</nome>  
    <telefone>4444444444</telefone>  
    <email>teste@gmail.com</email>  
</book>
```

STATELESS E STATUS CODE

stateless: cada solicitação enviada ao servidor contém todas as informações necessárias para que o servidor entenda e processe a solicitação, sem depender de nenhum estado mantido pelo servidor entre as solicitações.

statusCode: resultado de uma solicitação HTTP que fornecem informações sobre uma solicitação se ela foi bem-sucedida, redirecionada, falhou devido a um erro do cliente, falhou devido a um erro do servidor...



PAPEL DAS APIs NA INTEGRAÇÃO DE SISTEMAS E DESENVOLVIMENTO DE SOFTWARE

Integração de Sistemas

Permite que vários aplicativos se comuniquem entre si de forma eficiente. Fornecendo uma interface padronizada, onde podemos acessar e manipular os dados de dadas funcionalidades de um sistema, independentemente da sua implementação.

Desenvolvimento de Software

Desenvolvedores conseguem acessar e utilizarem funcionalidades de outros sistemas sem precisar entender a fundo todos os detalhes da implementação interna.

Desenvolvimento de API, fomenta a reutilização de código acelerando o processo de desenvolvimento, pois podemos reutilizar as APIS existentes ao inves de diversas funcionalidades serem implementadas do absoluto zero.

EXEMPLOS DE APIs

Diversas são as aplicações conhecidas que utilizam dessa arquitetura



PAGAMENTOS DIGITAIS

PagSeguro, paypal etc



SERVIÇOS EM NUVEM

API da Amazon Web Services (AWS), API do Google Cloud Platform (GCP), API do Microsoft Azure.

Um serviço WEB é construído de uma forma que suporte a interoperabilidade entre diferentes ambientes em uma rede na qual é baseada em padrões e protocolos WEB.

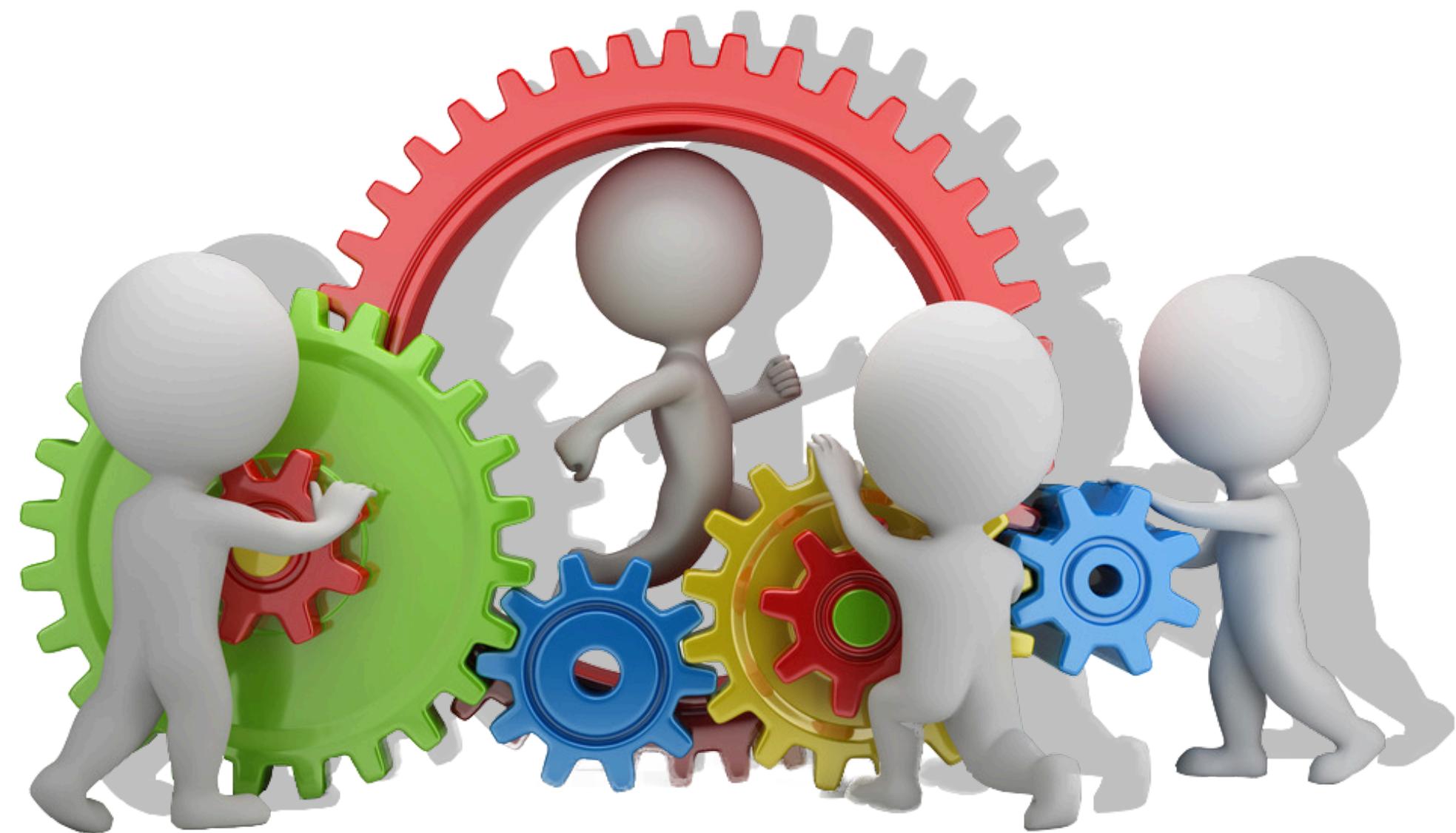
Interoperabilidade é a capacidade de diferentes sistemas, dispositivos ou componentes de software interagirem e funcionarem juntos de forma eficiente e eficaz.

Suponha que você tenha desenvolvido um serviço de pagamento online com base nos princípios de interoperabilidade e padrões da web. Os endpoints da sua API seguem o estilo **RESTful**, e os dados são trocados no formato **JSON**. Além disso, você implementou autenticação e autorização usando tokens **JWT** (JSON Web Tokens), que são amplamente suportados e compatíveis com uma variedade de bibliotecas e frameworks.



Exposição prática

Mão na Massa!



APLICAÇÕES QUE VOCÊ PRECISA TER ANTES DE COMEÇAR

- GIT HUB;
- VISUAL STUDIO CODE;
- NODEJS;
- SQLITE3 - <https://sqlitebrowser.org/dl/>
- POSTMAN.

Downloads - DB Browser for SQLite

Free code signing provided by SignPath.io, certificate by SignPath Foundation.

 sqlitebrowser.org

SIGA OS SEGUINTE PASSOS:

1. Crie uma pasta em seu computador, de a ela o nome que achar melhor;
2. **Acesse o repositório:** https://github.com/taynaraluanacaetano/crud_clientes_node
3. Vamos clonar o projeto na pasta que você acabou de criar em seu computador.

EXPLICAÇÃO DO PROJETO

1. Explicação do FRONT-END;
2. Explicação do BACK-END.

Exposição prática

TUDO PRONTO, COMO FAÇO PARA EXECUTAR?

- Através do seu terminal, acesse a pasta do projeto;
- Crie uma nova branch para que você possa fazer modificações com o tempo.
- git checkout -b nomeBranch;
- Navegue até a pasta do backend, e execute o seguinte comando:
- npm install;
- Após executar o comando, podemos iniciar o nosso projeto.

EXECUÇÃO

- Abra um terminal secundário e navegue até a pasta do backend e execute o seguinte comando: node app.js
- No seu terminal primário, navegue até a pasta frontend, e execute o arquivo login. htt-server

DESAFIO 1

Dante do que aprendemos sobre como funciona uma API, quais seus métodos, e também a importância da sua documentação, você poderá se juntar a mais outro colega, para executar o primeiro desafio.

1. Explore a documentação fornecida inicialmente na aula 1, examine a documentação da API, e entenda quais os endpoints que temos disponíveis, quais são os parâmetros aceitos e também os tipos de respostas esperadas.
2. Com base na documentação e análise realizada na mesma, crie uma lista de casos de testes para diferentes processos e funcionalidades da API, deve-se incluir casos de testes para cada um dos endpoints. Isso inclui a validação da regra de negócio.
3. Após análise e execução dos casos de testes realizados via POSTMAN, identifique todos os possíveis bugs encontrados na API e:
 - a. Registre os bugs encontrados;
 - b. Desenvolva uma descrição detalhada do problema;
 - c. Coloque o passo a passo para a reprodução desse bug;
 - d. Descreva o impacto que esse bug pode causar na funcionalidade da API.

Enviar o desafio no formato PDF

- Nome completo dos integrantes
- Enviar: taynaraluanac@gmail.com
- Espaço reservado para apresentação de 2 grupos.

CALENDÁRIO

AULA 2 PARTE 1

CONCEITOS API

- Princípios do rest p2.
- Recursos e URI.

CONCEITOS TESTES AUTOMATIZADOS

- Introdução aos testes automatizados de API REST.
- Tipos de testes de API;
- Ferramentas, frameworks;
- Boas práticas para escrever testes de API;
- Conhecendo os frameworks que iremos utilizar.

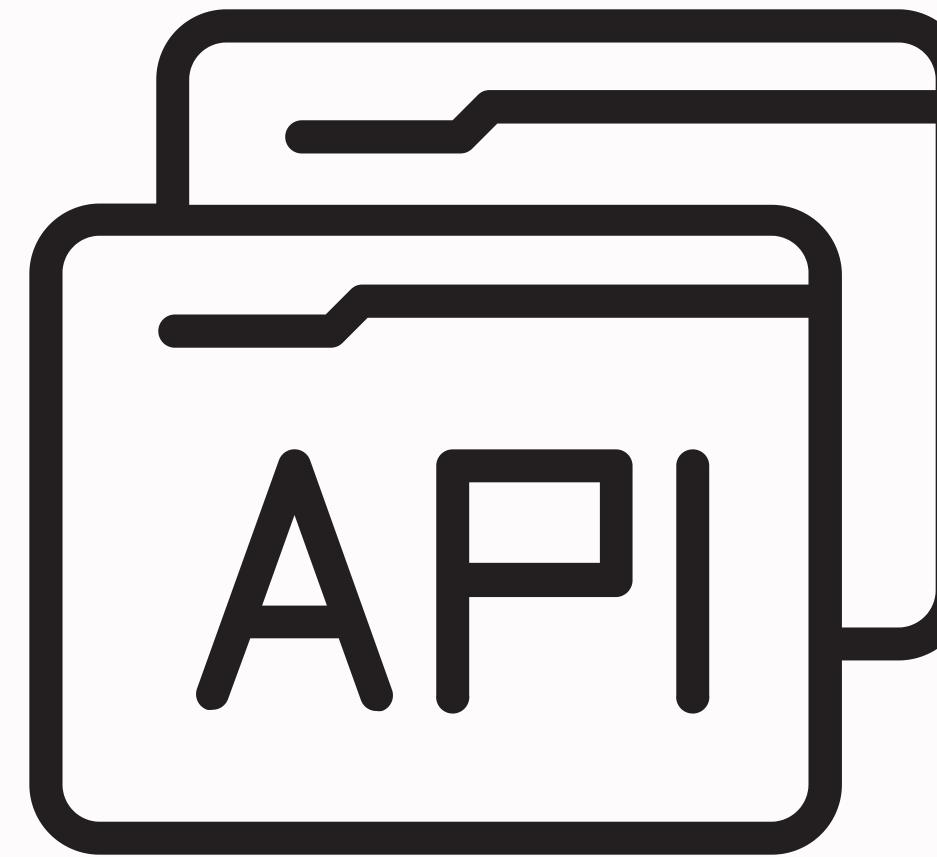
Atividades Práticas

- Executando os primeiros testes na API.
- Configuração do ambiente de desenvolvimento.

Atividades entregáveis

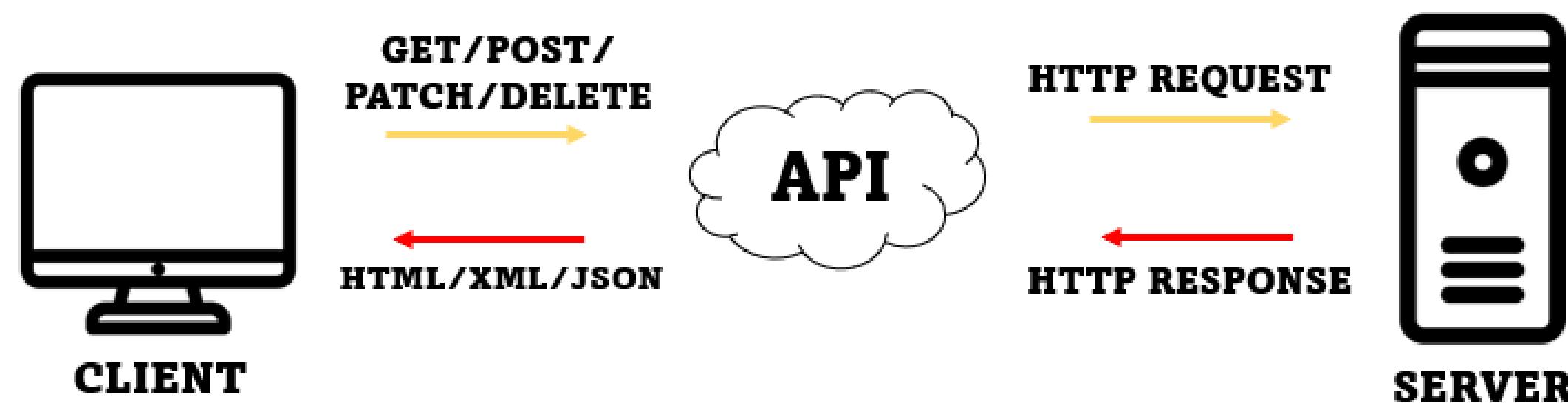
- Desafio 02

APIs são ferramentas poderosas no mundo profissional, permitindo acesso fácil e rápido a informações valiosas que podem impulsionar os negócios e melhorar a eficiência operacional.



PRINCÍPIOS DO REST:

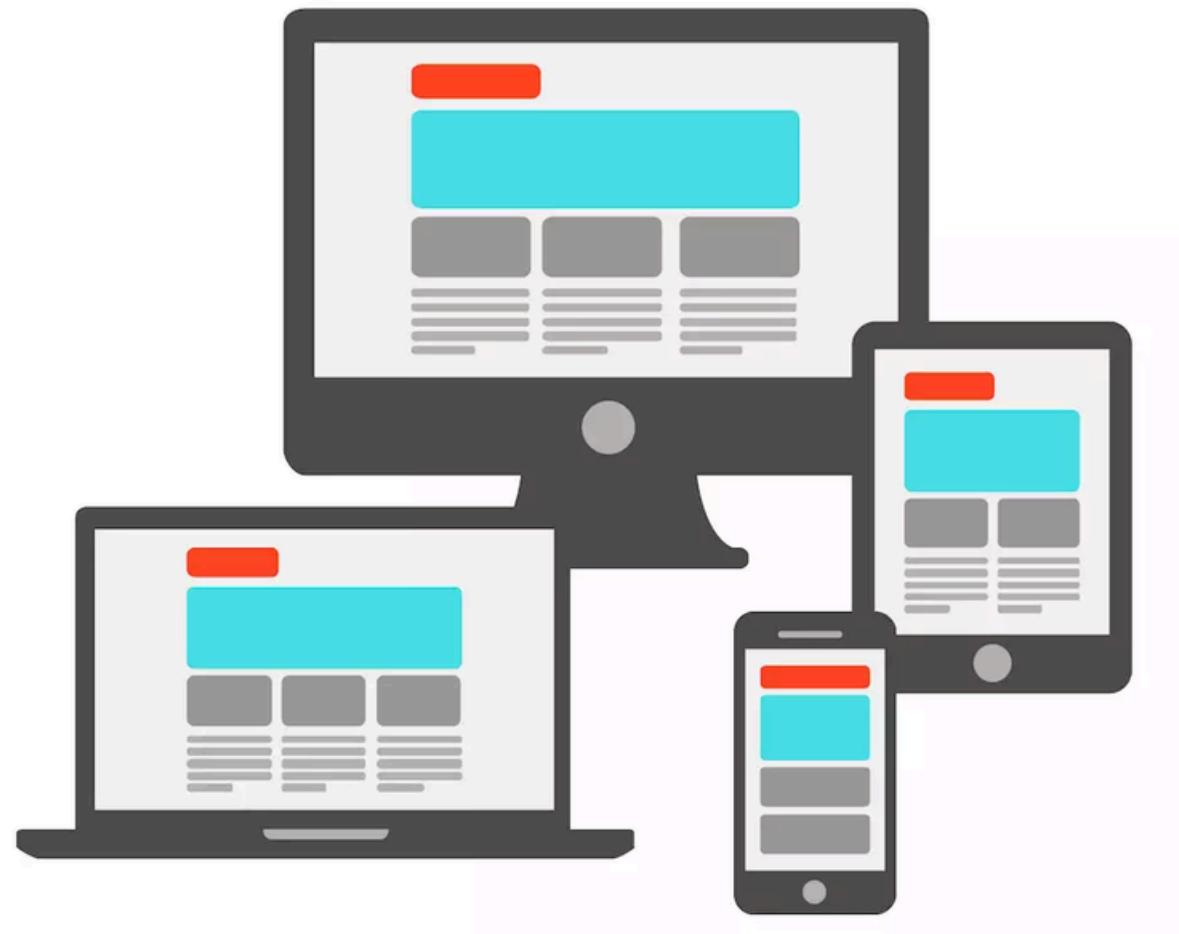
- Arquitetura Cliente-Servidor;
- Stateless (Sem Estado);
- Cacheabilidade;
- Interface Uniforme;
- Sistema em Camadas.



Fonte da Imagem: <https://velog.io/@wlgus2134/TIL14REST-API%EB%9E%80-%EB%AC%B4%EC%97%87%EC%9D%B8%EA%B0%80>

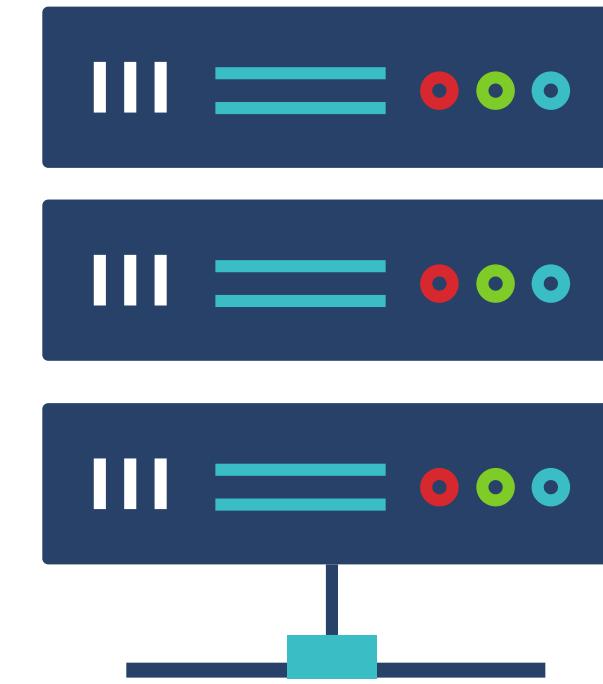
Cliente e servidor

Dois modelos de entidades: cliente e servidor



CLIENTE

Enviamos solicitações para uma API através de navegadores web, aplicativos móveis etc.



SERVIDOR

É onde a API é executada, recebe as solicitações do cliente, processa essas solicitações e retorna as respostas correspondentes.

Em nosso estudo o Express.js atua como o servidor que recebe as solicitações HTTP dos clientes e fornece as respostas correspondentes.

ELEMENTOS-CHAVE DO REST

- **Recursos e URI:**

- Recursos são entidades ou conceitos abstratos que são acessados via URI (Uniform Resource Identifier);
- Cada recurso é identificado por um URI único, que geralmente segue uma estrutura hierárquica.

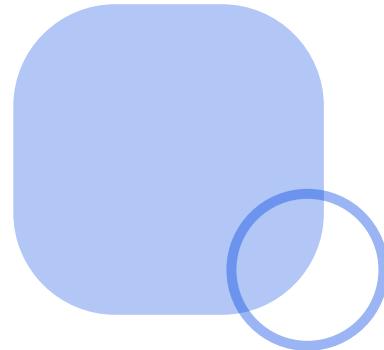
Exemplo de URI: <http://localhost:3000/api/users>

SANTOS, Taynara Luana Caetano dos. Aderência de técnicas de testes automatizados em uma api rest: abordagem em uma aplicação de hub integrador de marketplaces. 2023. Dissertação (Mestrado em Inovações Tecnológicas) - Universidade Tecnológica Federal do Paraná, Campo Mourão, 2023.

Mão na Massa!



TESTES QUE PODEMOS EXECUTAR NA API EM QUESTÃO



**Quais os testes que podemos executar na API que
estamos atualmente testando?**





Testes de Integração na API de Cadastro de Usuários

Objetivo:

Realizar testes de integração na API de cadastro de usuários para garantir que as funcionalidades de cadastro e listagem de usuários estejam funcionando corretamente.

Teste de Cadastro de Usuário:

- Envie uma requisição POST para o endpoint /api/users com dados válidos de usuário (nome, e-mail, senha).
- Verifique se a resposta retorna um status de sucesso (status code 201).
- Verifique se os dados do usuário cadastrado estão corretos na resposta.
- Valide a persistencia dos dados no banco.



Testes de Integração na API de Cadastro de Usuários

Objetivo:

Realizar testes de integração na API de cadastro de usuários para garantir que as funcionalidades de cadastro e listagem de usuários estejam funcionando corretamente.

Teste de Listagem de Usuários:

- Envie uma requisição GET para o endpoint **/api/users**.
- Verifique se a resposta retorna um status de sucesso (status code 200).
- Verifique se a lista de usuários não está vazia.



Testes de Integração na API de Cadastro de Usuários

Objetivo:

Realizar testes de integração na API de cadastro de usuários para garantir que as funcionalidades de cadastro e listagem de usuários estejam funcionando corretamente.

Teste de Cadastro de Usuário com Dados Inválidos:

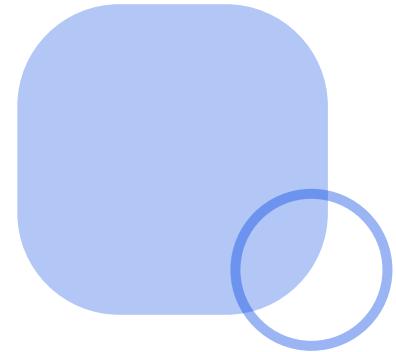
- Envie uma requisição POST para o endpoint **/api/users** com dados inválidos de usuário (por exemplo, nome em branco, e-mail inválido).
- Verifique se a resposta retorna um status de erro apropriado (status code 400 ou 422).
- Verifique se a resposta contém uma mensagem de erro explicativa.

TESTES QUE PODEMOS EXECUTAR NA API EM QUESTÃO

Em resumo:

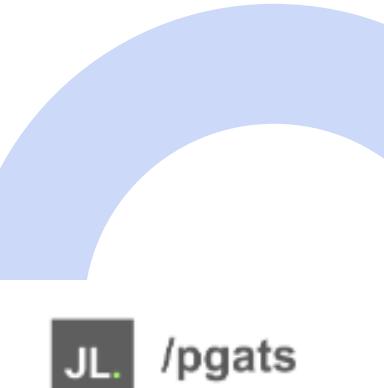
1. Podemos testar as requisições básicas;
2. Testar o contrato;
3. Testar as respostas e os formatos;
4. Testar a validação dos dados de entrada;
5. Testar a atualização dos registros já existentes;
6. Realizar testes para manipular erros;
7. Realizar testes para a validação de desempenho e segurança;
8. Testar a remoção de registros e retornos.

TESTES AUTOMATIZADOS EM UMA API REST

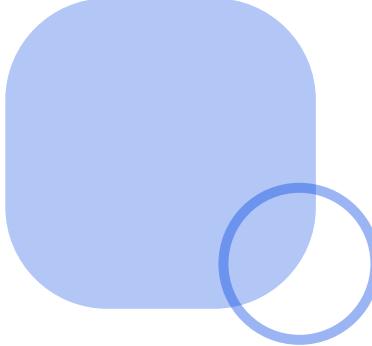


Possibilita que:

- O software obtenha um nível de **qualidade** elevado;
- Possíveis problemas sejam identificados de forma precoce;
- Mesmo após uma alteração, o software funcione como o esperado, pois estamos verificando o que estamos enviando e o que estamos recebendo.



IMPORTÂNCIA DOS TESTES AUTOMATIZADOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

- 
- Detecção de bugs rapidamente;
 - Promover a entrega contínua;
 - Aumenta a confiança no código consequentemente nas entregas.

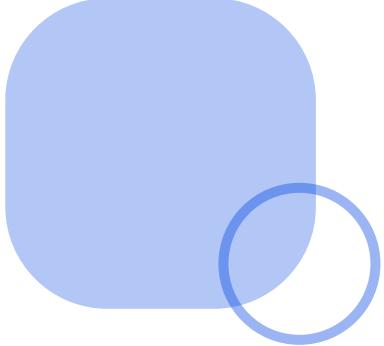
TIPOS DE TESTES AUTOMATIZADOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

- Testes **unitários** com a utilização de JUnit;
- Testes de **integração** com a utilização do Postman;
- Testes de **aceitação** com RestAssured;
- Testes **regressivos** com Selenium.

Mão na Massa!

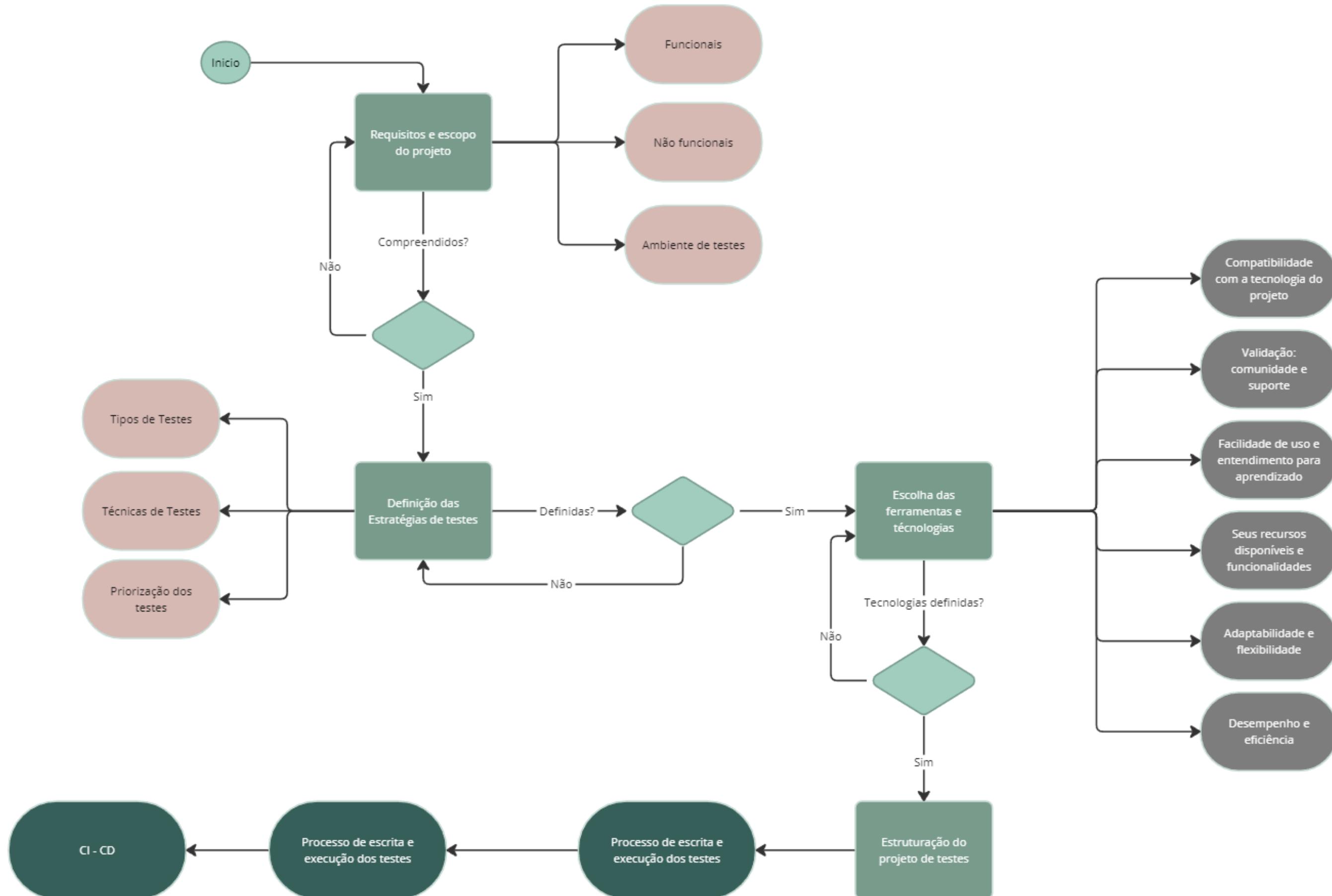


Objetivos

- 
- Ferramentas, frameworks e linguagem de programação;
 - Construindo um projeto de testes automatizados do zero;
 - Instalações e dependências;
 - Primeiro script de testes automatizados;
 - A importância dos Logs.
- 

Arquitetura de um projeto de automação de testes

Definindo a Arquitetura de um Projeto de Testes Automatizados

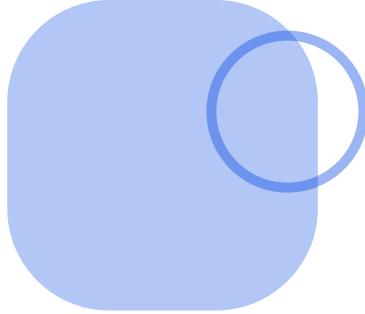


Mão na Massa!



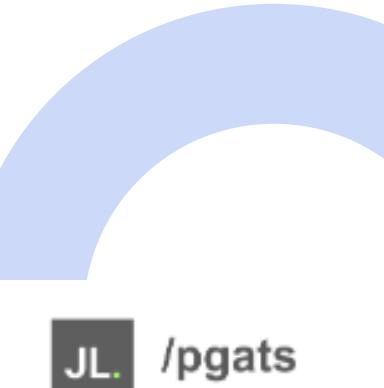
Primeiro script de teste com jest io

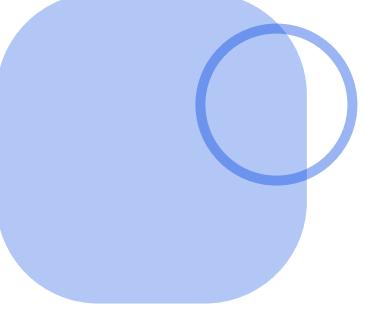




Praticando os primeiros scripts de testes na API

- Validação de statusCode;
- Validação do corpo da resposta;
- Validação da existência de propriedades.





Melhorando o nosso script utilizando boas práticas

- Inserindo arquivos .env;
- Inserindo arquivo gitignore;
- Otimizando payloads reutilizáveis.

DESAFIO 2

Diante do que aprendemos sobre como podemos criar uma pequena arquitetura de um projeto de automação de testes, boas práticas, e também como escrever testes de API REST utilizando o jest:

De acordo com a documentação de execução e evidências de testes que vocês desenvolveram, utilize os casos de testes de cadastro de um usuário, para testar os cenários onde estamos recebendo resultados de erro, e aproveite para inserir também os erros que vocês encontraram, pois se forem corrigidos, iremos alterar o código e pedir que vocês executem somente o teste automatizado para verificar que de fato o problema foi devidamente corrigido.