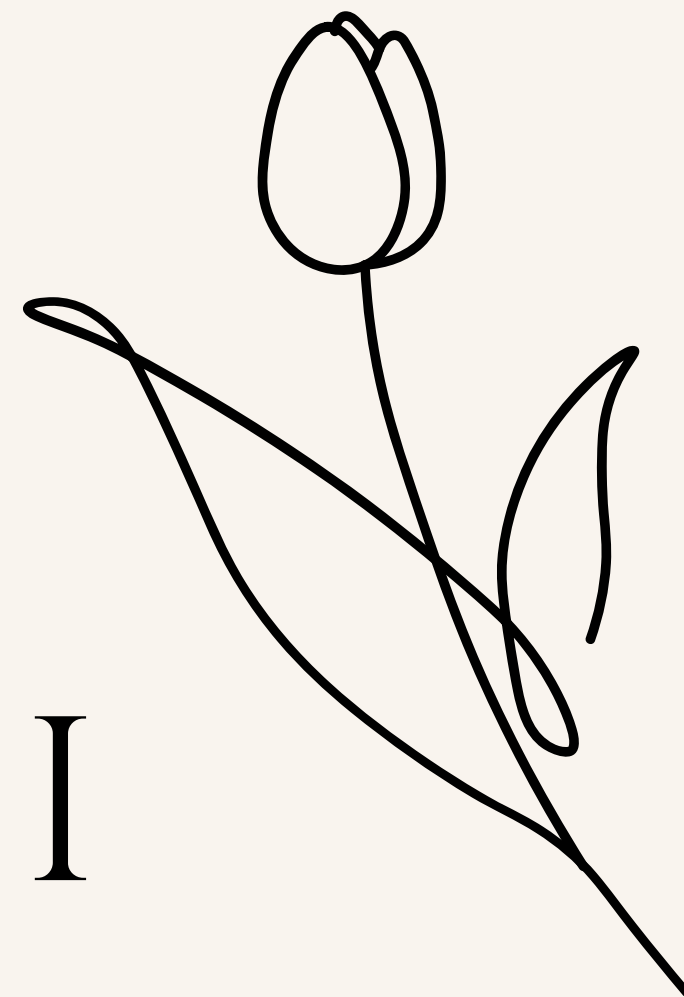


# TRABALHO API



---

LAURA LIVRAMENTO, GABRIEL.B E LAVÍNIA

# TIPOS DE API

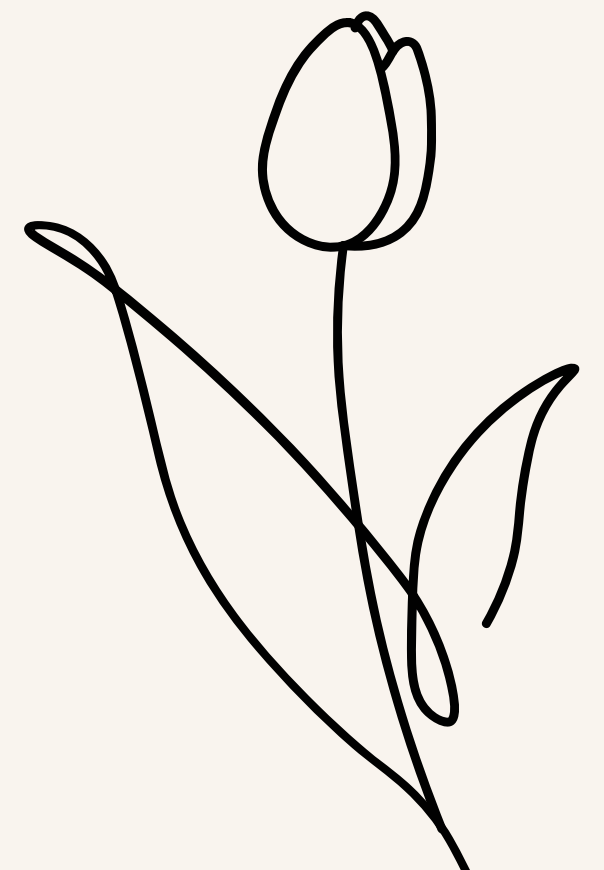
1.API RESTful: Baseado no estilo arquitetônico REST, utiliza métodos HTTP padrão para operações sobre recursos.

2. SOAP: Protocolo de comunicação padrão baseado em XML, comum em aplicações empresariais.

3.GraphQL: Linguagem de consulta para APIs que permite solicitar dados específicos, ganhando popularidade em aplicações modernas.

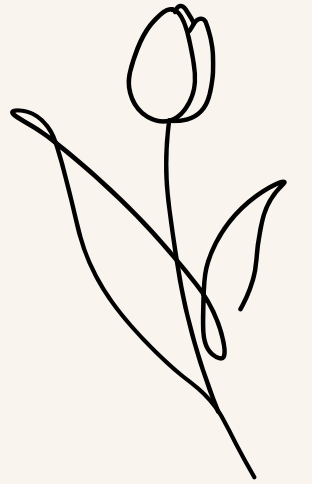
4.WebSocket: Fornece conexões bidirecionais persistentes para comunicação em tempo real.

5.gRPC: Sistema eficiente de chamada de procedimento remoto desenvolvido pelo Google, ideal para alto desempenho entre serviços distribuídos.



# Ferramentas (Postman, Swagger);

---



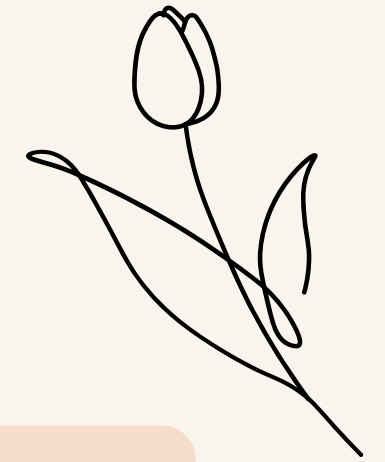
**Postman:** É uma plataforma de colaboração para desenvolvimento de APIs. Ele permite aos usuários criar, testar, documentar e compartilhar APIs de forma rápida e eficiente. O Postman oferece uma interface amigável para enviar solicitações HTTP para APIs, testar diferentes endpoints, gerenciar variáveis de ambiente e automatizar fluxos de trabalho.

**Swagger:** é uma estrutura de código aberto que permite descrever, documentar e consumir APIs de forma fácil e padronizada. Ele usa a especificação OpenAPI para definir a estrutura e os detalhes de uma API RESTful, incluindo endpoints, parâmetros, respostas e esquemas de dados. O Swagger UI, uma ferramenta integrada ao Swagger, gera uma interface interativa baseada na especificação OpenAPI, permitindo que os desenvolvedores visualizem e testem endpoints diretamente no navegador.

# VERBOS HTTP

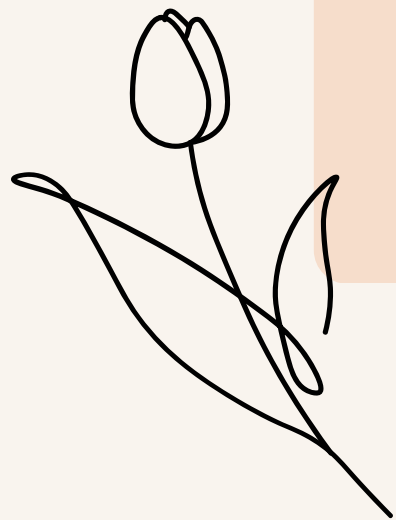
- 1.GET: Solicita dados de um recurso especificado.
2. POST: Submete dados para serem processados por um recurso especificado.
- 3.PUT: Atualiza um recurso especificado com novos dados.
- 4.DELETE: Exclui um recurso especificado.
5. PATCH: Aplica modificações parciais a um recurso.
- 6.HEAD: Solicita apenas os cabeçalhos de um recurso sem o conteúdo do corpo.
7. OPTIONS: Solicita informações sobre as opções de comunicação disponíveis para um recurso.
- 8.TRACE: Realiza um teste de retorno de mensagem ao longo do caminho até o recurso de destino.

# DIFERENÇAS DE HTTP E HTTPS



## Http

É um protocolo padrão para transferência de dados na web. Os dados são transmitidos em texto simples, o que significa que podem ser facilmente interceptados por terceiros. Isso pode representar um risco de segurança ao enviar informações sensíveis, como senhas e dados pessoais.



## Https

É uma versão segura do HTTP que utiliza criptografia para proteger os dados durante a transferência. O HTTPS adiciona uma camada de segurança chamada SSL/TLS (Secure Sockets Layer/Transport Layer Security) aos dados, criptografando as informações antes de serem enviadas pela internet. Isso torna muito mais difícil para os hackers interceptarem e lerem os dados transmitidos.

EM RESUMO, O HTTPS OFERECE UMA CAMADA ADICIONAL DE SEGURANÇA EM COMPARAÇÃO COM O HTTP, GARANTINDO QUE AS INFORMAÇÕES TRANSMITIDAS PELA INTERNET ESTEJAM PROTEGIDAS CONTRA INTERCEPTAÇÃO E MANIPULAÇÃO POR TERCEIROS.

# FORMATO DE DADOS (JSON E XML)

JSON e XML ambos formatos de dados amplamente utilizados para estruturar e armazenar informações. JSON é mais leve e fácil de ler, enquanto XML é mais expressivo e flexível.

