

# Conceitos Fundamentais

//Para o desenvolvimento Web

IT BOARDING

**BOOTCAMP**



# Índice



**01** Arq. Cliente  
Servidor

**02** Protocolo HTTP

**03** Frameworks vs.  
Bibliotecas + Maven

**04** API REST

IT BOARDING

**BOOTCAMP**

## // Conceitos para recordar

### O que é Frontend?

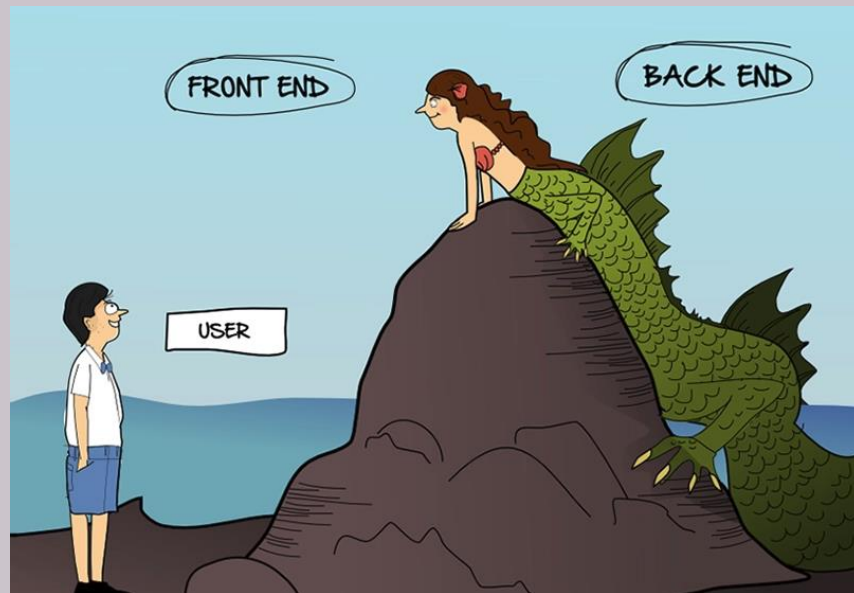
É a parte de um software que o **usuário pode acessar e interagir** diretamente (interface gráfica).

### O que é Backend?

Camada de **acesso aos dados** de um software ou qualquer dispositivo, **contém a lógica** da aplicação que trata esses dados.

### O que é FullStack?

É o nome dado aos desenvolvedores que podem realizar uma aplicação do **início ao fim**, ou seja, tanto em sua seção de front-end, como de back-end e todas as ferramentas complementares.

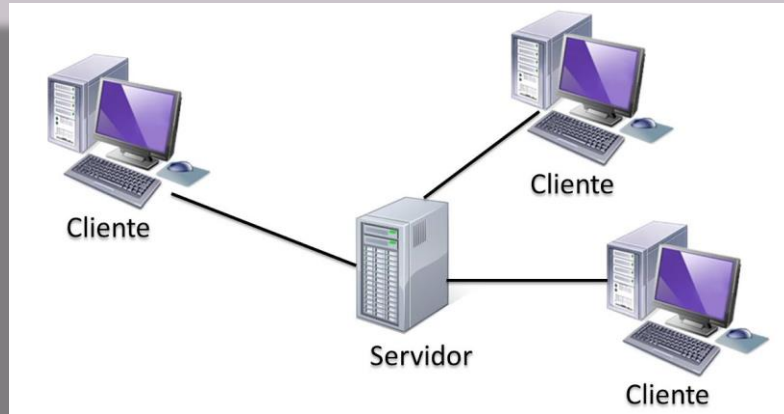


# Arquitetura Cliente - Servidor

IT BOARDING

**BOOTCAMP**





## Arquitetura Cliente-Servidor

É um modelo de design de software no qual as tarefas são compartilhadas entre provedores de recursos ou serviços, chamados **servidores**, e demandantes, chamados **clientes**.

- Os **clientes** fazem solicitações aos servidores (**request**).
- Os **servidores** fornecem uma resposta sobre o que o cliente solicita (**response**).

## Arquitetura Cliente-Servidor

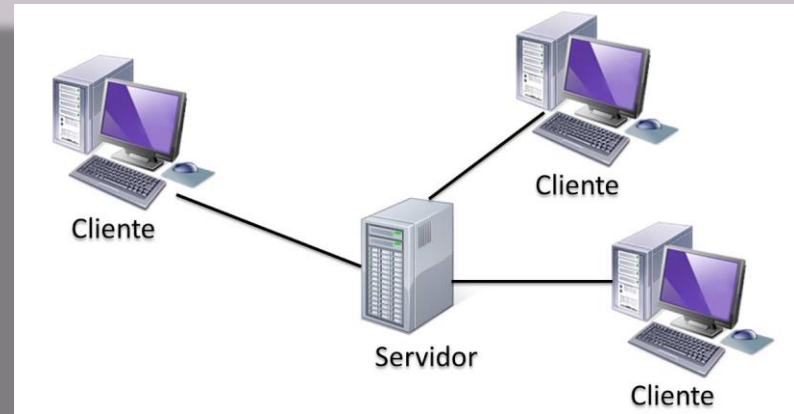
### //Cliente

Pode ser representado como **um computador comum e atual** que tenta acessar um servidor ou serviço oferecido pela Internet ou por uma rede interna.

Os clientes mais conhecidos são os **navegadores web** (Google Chrome, Mozilla Firefox, Opera, etc)

### //Servidor

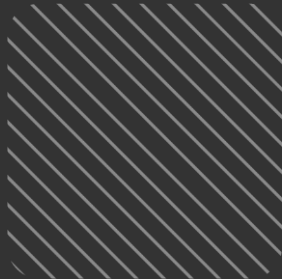
Tal como o cliente, também é **um computador**, mas com a diferença que tem uma **grande capacidade de processamento, armazenamento** ou manuseio de aplicações em simultâneo que lhe permite atender ao número de pedidos que recebe dos clientes.



# Protocollo HTTP

IT BOARDING

**BOOTCAMP**





## // O que é HTTP?

- Por sua sigla em inglês significa: “**H**ypertext **T**ransfer **P**rotocol” (Protocolo de transferencia de hipertexto).
- É um protocolo que visa a **comunicação entre o cliente e o servidor**
- É um protocolo **orientado a transações**, onde o cliente faz uma solicitação enviando ao servidor uma “mensagem” (**request**) com determinado formato e o servidor, posteriormente, envia uma “mensagem” de resposta. (**response**).

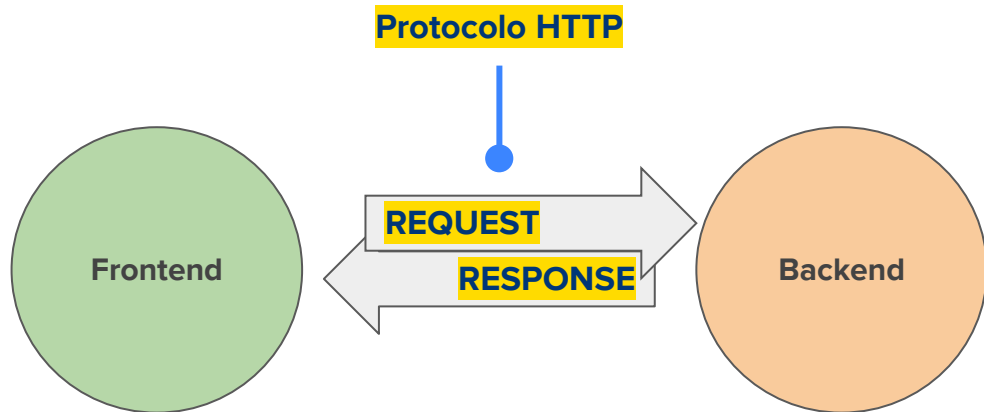






## // Como isso se reflete em um aplicativo?

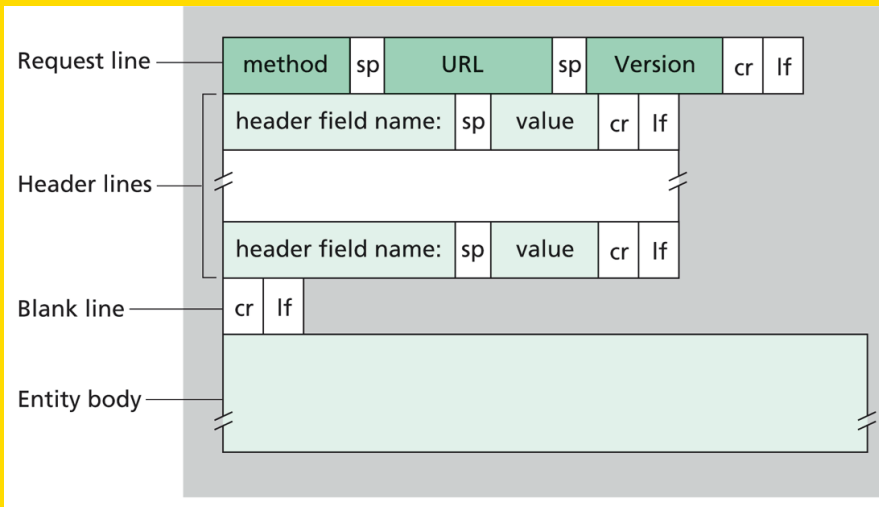
- Geralmente, o **backend** é refletido como **servidor**, enquanto o **frontend** é o **cliente**.
- Para trocar mensagens entre o cliente e o servidor. O cliente faz “requests” (requisições) e o servidor devolve “responses” (respostas).
- Cada uma das mensagens entre o cliente e o servidor possui uma **estrutura** com a qual são enviadas através da rede existente.





## // Estrutura de um REQUEST

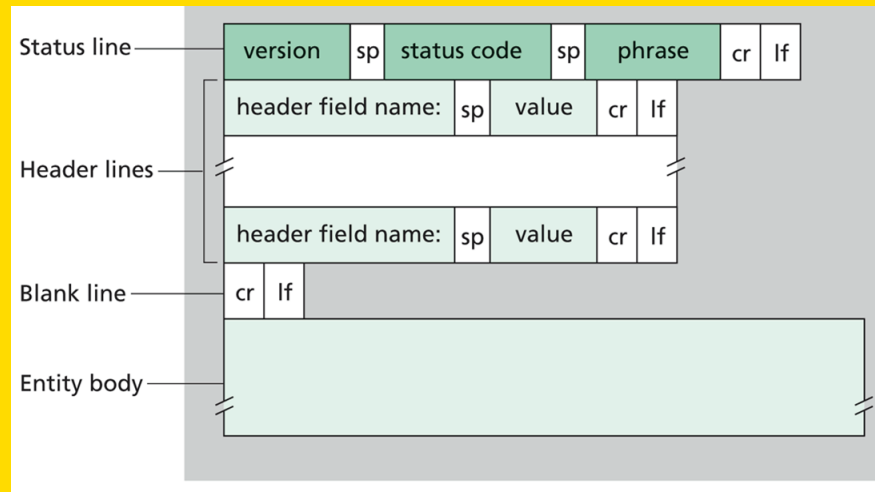
- **METHOD:** Indica qual método está sendo utilizado para o envio da mensagem. Existem diferentes métodos, alguns dos mais conhecidos são: GET, POST, PUT e DELETE.
- **URL:** Endereço para onde o pedido está sendo enviado.
- **HEADER (cabeçalho):** Atributos gerais ou essenciais incluídos (por exemplo, Auth).
- **BODY:** Corpo da mensagem utilizada nas solicitações para poder transmitir dados.





## // Estrutura de uma RESPONSE

- **STATUS:** É um código que representa o que acontece com uma solicitação. Existem diferentes “Status Code”:
  - **1xx:** Respostas informativas. Indica que a solicitação foi recebida e está sendo processada.
  - **2xx:** Respostas corretas. Indica que a solicitação foi processada corretamente. (Exemplo: 200 Sucesso)
  - **3xx:** Redirecionar respostas.
  - **4xx:** Erros causados pelo cliente. (Exemplo: 404 não encontrado)
  - **5xx:** Erros causados pelo servidor. (Exemplo: 500 Erro Interno do Servidor)
- **HEADER (cabeçalho):** Atributos gerais ou essenciais incluídos (por exemplo, Auth).
- **BODY:** Resposta da mensagem



# Frameworks, Bibliotecas & Maven

IT BOARDING

**BOOTCAMP**





“Uma **biblioteca** é um conjunto de implementações de código funcional em uma linguagem de programação que está disponível para uso em qualquer aplicativo / programa.”

IT BOARDING

**BOOTCAMP**



“Um **framework** é um **ambiente de desenvolvimento**, com o qual podemos resolver problemas comuns mais rapidamente, **usando diferentes bibliotecas** para isso.

IT BOARDING

**BOOTCAMP**



## // O que é Maven?

- É uma ferramenta de software para gerenciamento e construção de projetos Java.
- **Maven** nos permite instalar bibliotecas dentro do nosso projeto de uma forma fácil e escalonável..
- **IntelliJ IDEA** é integrado ao Maven e facilita a instalação de plug-ins e bibliotecas.

IT BOARDING

**BOOTCAMP**

# maven



# API REST

//O que são as API?

IT BOARDING

**BOOTCAMP**

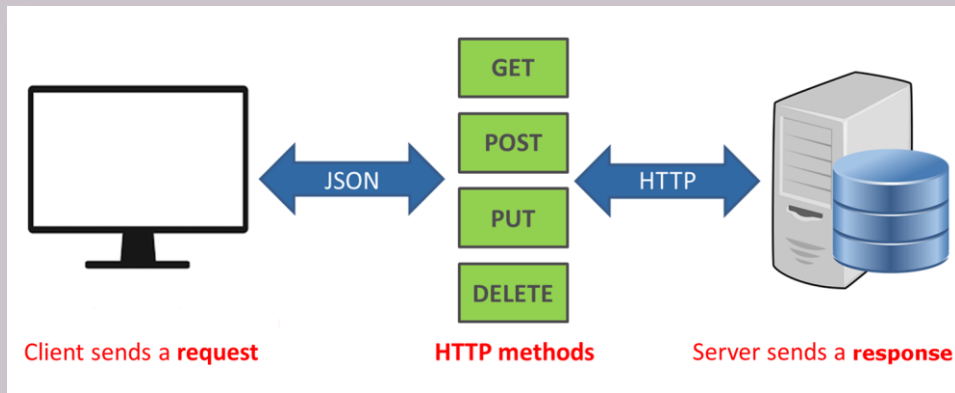






## // O que é uma API REST?

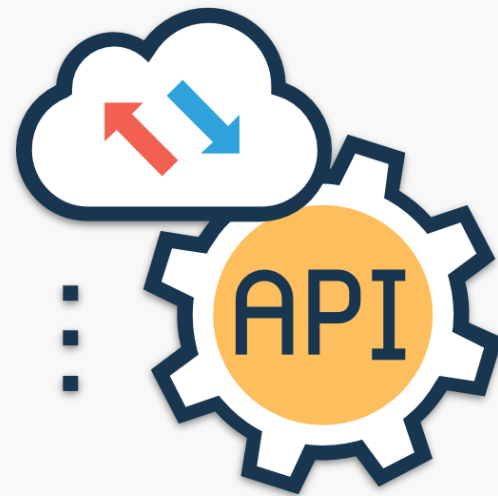
- Uma **API** (interface de programação de aplicativo) é um conjunto de funções e procedimentos (métodos) usados para projetar e integrar o software de diferentes aplicativos.
- **REST (REpresentational State Transfer)** é um tipo de serviço que não tem estado.
- É qualquer interface (interconexão) entre sistemas que usa HTTP como protocolo para obter dados ou gerar operações sobre ele em formatos como XML e JSON.



# O que uma API nos permite fazer?



- Uma API nos permite **interconectar sistemas independentemente da linguagem do idioma** em que são desenvolvidos.
- Uma API permite oferecer um determinado serviço / aplicação para que possa ser consumido por vários clientes ao mesmo tempo.
- Uma das principais funções das APIs é **facilitar o trabalho dos desenvolvedores, permitindo realizar o conhecido reaproveitamento de recursos / softwares.**
- **Exemplo de uso:** Suponha que você esteja criando um aplicativo que é uma loja online, usando o API's, você não precisaria criar um sistema de pagamento do zero para receber pagamentos pelas compras de seus clientes; Você pode facilmente usar a API de um serviço de pagamento existente, por exemplo MercadoPago ou PayPal.





# Dúvidas?

IT BOARDING

**BOOTCAMP**

