

# Arquitectura Cliente-Servidor

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree

# Índice

1. Papel do cliente
2. Papel do servidor
3. Front-end? Back-end?

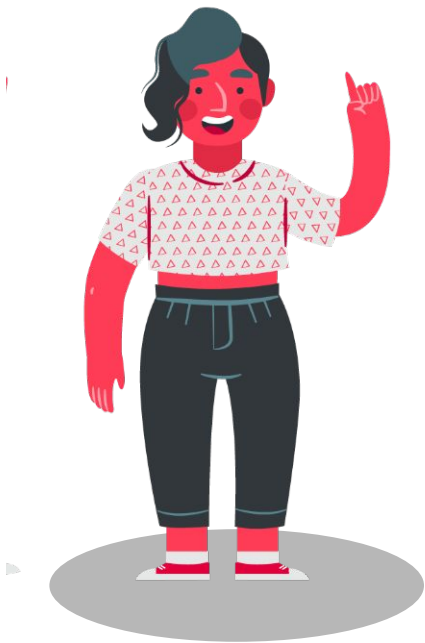
# 1 | Papel do Cliente

“

Arquitetura cliente-servidor é um modelo de comunicação que **conecta** dispositivos e servidores por meio da Internet.

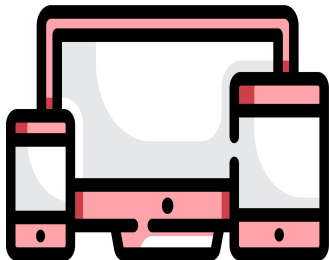
”





## O que chamamos de cliente?

Podemos definir clientes como **dispositivos** que **solicitam** serviços ou recursos a um servidor.



Esses dispositivos podem ser: computadores, smartphones, tablets, console de videogame, ou seja, qualquer dispositivo que se conecta a uma **rede**.



## **Como é o acesso do cliente?**

Na rede Internet, o cliente geralmente acessa os serviços e recursos por meio de um navegador da web.



# 2 | Papel do Servidor





## O que chamamos de servidor?

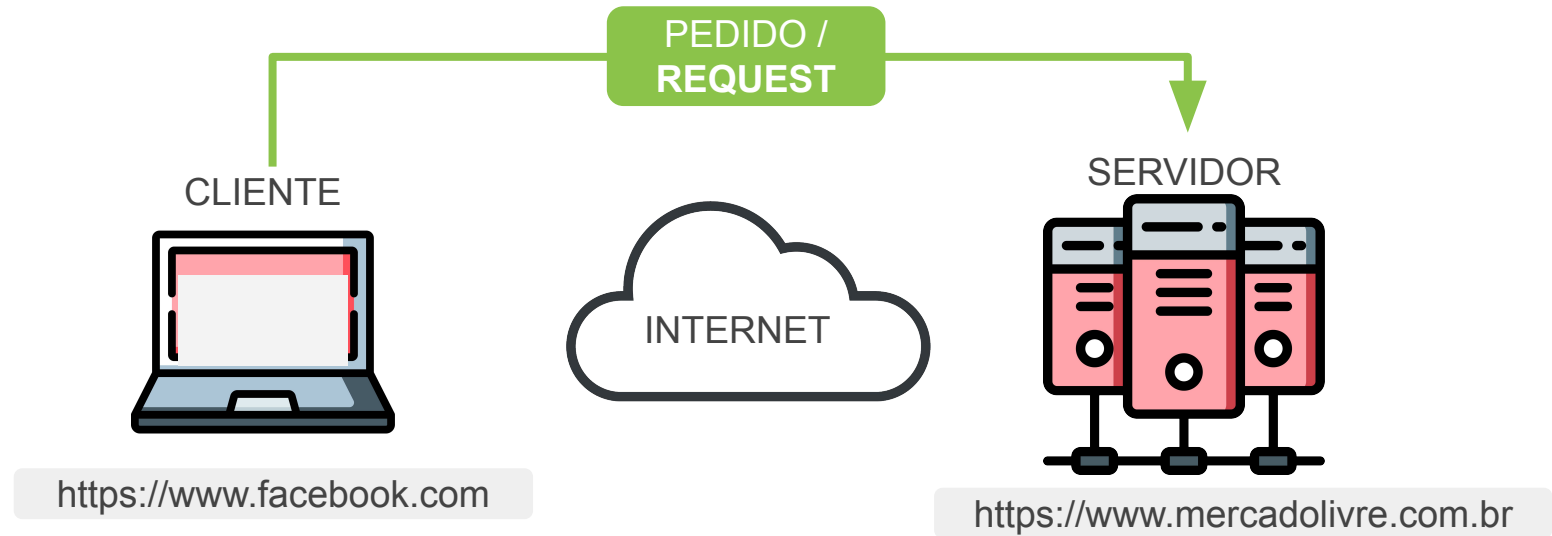
Dispositivos que **fornecem** os serviços, arquivos e recursos para os clientes. Em outras palavras: responde os pedidos dos clientes.

Um computador pode ser o cliente e servidor ao mesmo tempo. Na verdade, é bem comum no ambiente de desenvolvimento de um site ou aplicativo da web.



# Os pedidos (*requests*)

São as solicitações que fazemos por meio do navegador (cliente) a um servidor quando entramos em um site, por exemplo.



## As respostas (*responses*)

O servidor recebe nossa solicitação, processa e envia como resultado para o cliente.





**Por que é importante  
conhecer o fluxo de  
Request-Response?**

## Por quê?



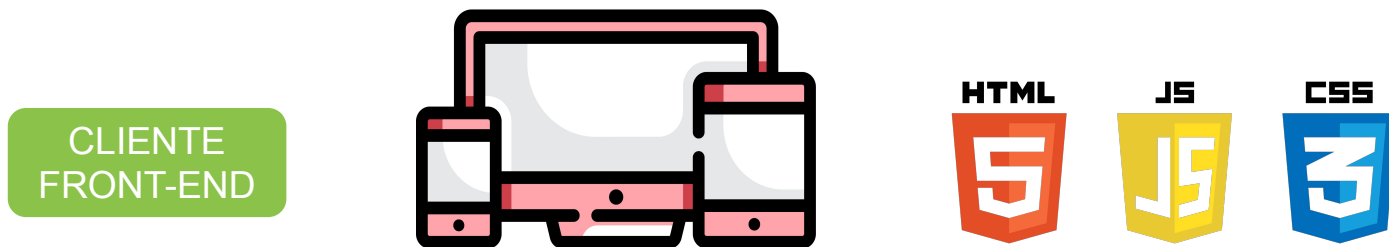
Dentro do mundo do desenvolvimento web, a maioria das aplicações tem duas frentes:

**Front-end** e **Back-end**

# 3 | Front-end? Back-end?

# Front-end

É tudo o que acontece no lado do **cliente**. Todos os elementos visuais que compõem a interface gráfica do site estão desse lado, além da parte de funcionalidade. As linguagens tratadas desse lado são **HTML** para a estrutura do site, **CSS** para os estilos visuais e **JavaScript** para as interações do site.

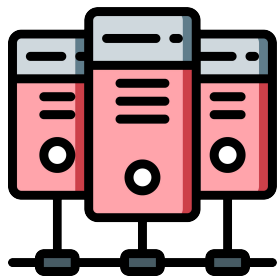




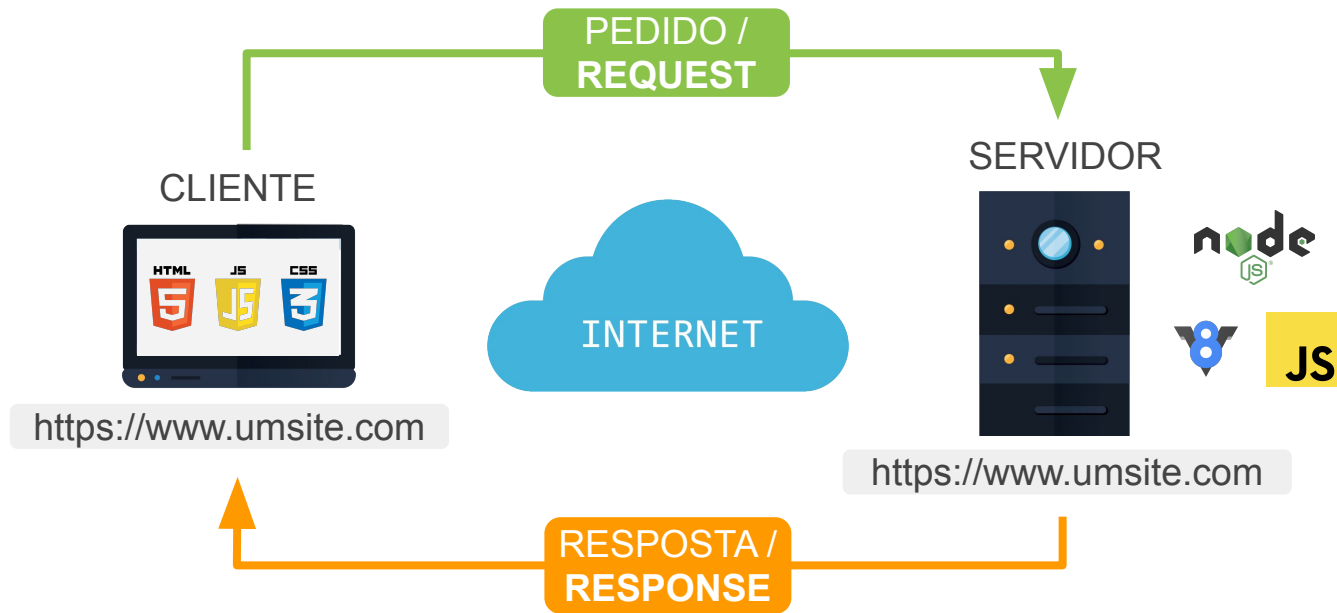
# Back-end

É tudo o que acontece do lado do **servidor**. Aqui se inclui todo o funcionamento **interno e lógico do site**. É o que permite que todos os pedidos solicitados pelos clientes sejam atingidos. Tratamos: bases de dados como MySQL, linguagens como PHP, JavaScript para sites dinâmicos e frameworks como Express e Laravel.

SERVIDOR  
BACK-END



# Fluxo Cliente-Servidor

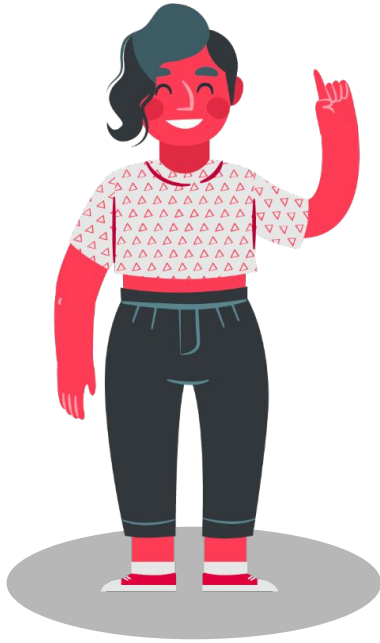


# Sobre Javascript

Hoje existe a possibilidade de rodar o JavaScript do lado do servidor, o que permite programar na mesma linguagem tanto no front como no back, tornando o processo de desenvolvimento mais fluido. Mas isso é assunto para outra aula.



# Protocolo de Transferência de Hipertexto

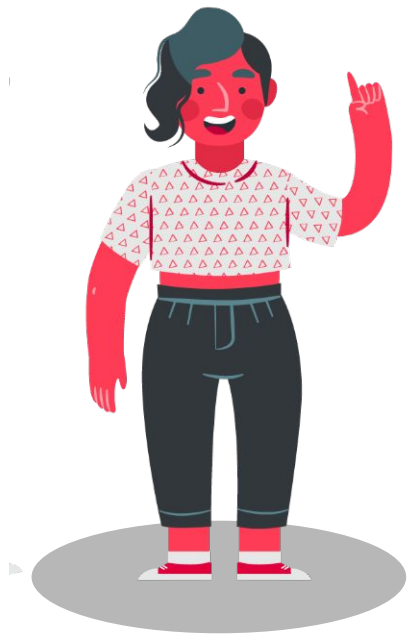


“

Para que esse modelo de comunicação aconteça, é necessário uso do protocolo HTTP.

”

# Protocolo de Transferência de Hipertexto



O protocolo HTTP utiliza códigos de status, que indicam se uma requisição HTTP foi concluída da forma correta. Esses códigos são agrupados em cinco classes:

# Protocolo de Transferência de Hipertexto

## Códigos de Status

- **100 a 199** - Respostas de informação
- **200 a 299** - Respostas de sucesso
- **300 a 399** - Redirecionamentos
- **400 a 499** - Erros do cliente
- **500 a 599** - Erros do servidor



**Já tentou acessar algum  
site e seu navegador  
retornou esse resultado:**

{ Página no encontrada... }



[Ir al inicio](#)



# Protocolo de Transferência de Hipertexto

- 1 Essa é uma mensagem automática que o servidor envia ao cliente, informando que a página solicitada não foi encontrada, ou que não existe.
- 2 Provavelmente isso aconteceu devido ao fato de o usuário digitar o site incorreto na URL, ou que a página foi removida, ou que não existe.



DigitalHouse>  
Coding School