

*disciplina de*

**p2 2 2 2d**

**A 2 ApA 2d 2  
e**

*Professor Jefferson de Oliveira Chaves  
jefferson.chaves@ifpr.edu.br*

## A 2 pA

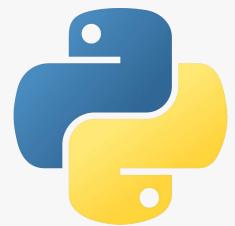
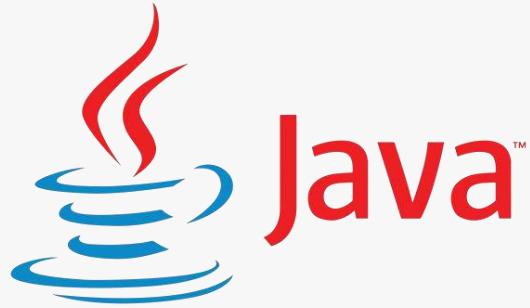
- ∴ Compreender arquitetura Cliente-Servidor;
- ∴ Conhecer o Protocolo HTTP;
- ∴ Endereços, portas e recursos;
- ∴ Interface Uniforme;
- ∴ HttpServlet Request e HttpServlet Response;

p2 2 p2

2 p2

A A

A



python™



android

A 2 Ae 2 2 2 p



Cliente



Regras de  
comunicação



Servidor

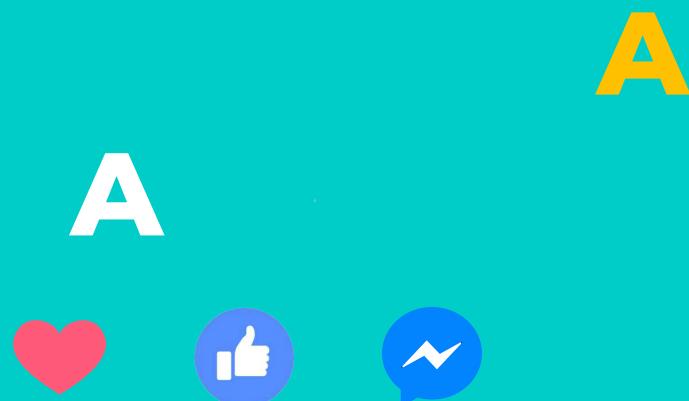


## 2 Ad2 2e2 p A e 2 X

- ∴ Nesse arquitetura, sempre é o cliente quem estabelece as conexões;
- ∴ Abrir uma conexão HTTP significa iniciar uma conexão na camada de transporte, geralmente TCP;
- ∴ Usando o protocolo TCP, a porta padrão é a porta 80, porém outras portas podem ser utilizadas, como a 8000 ou 8080;
- ∴ A URL da página requisitada contém o nome do domínio e o número da porta, quer pode ser omitida caso a porta seja a 80;

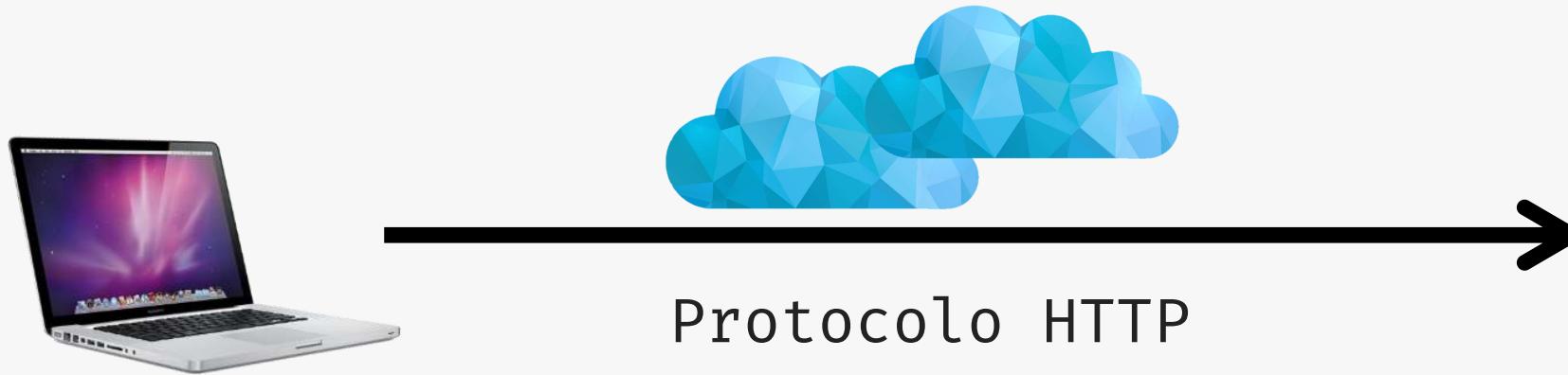
*Leia mais:*

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Session>



- O modelo cliente-servidor não permite o servidor enviar dados ao cliente sem uma requisição explícita
- Para contornar esse problema, os desenvolvedores usam várias técnicas:
  - Executar ping no servidor periodicamente por meio das APIs XMLHttpRequest;
  - Fetch, usando a WebSockets API ou protocolos semelhantes.

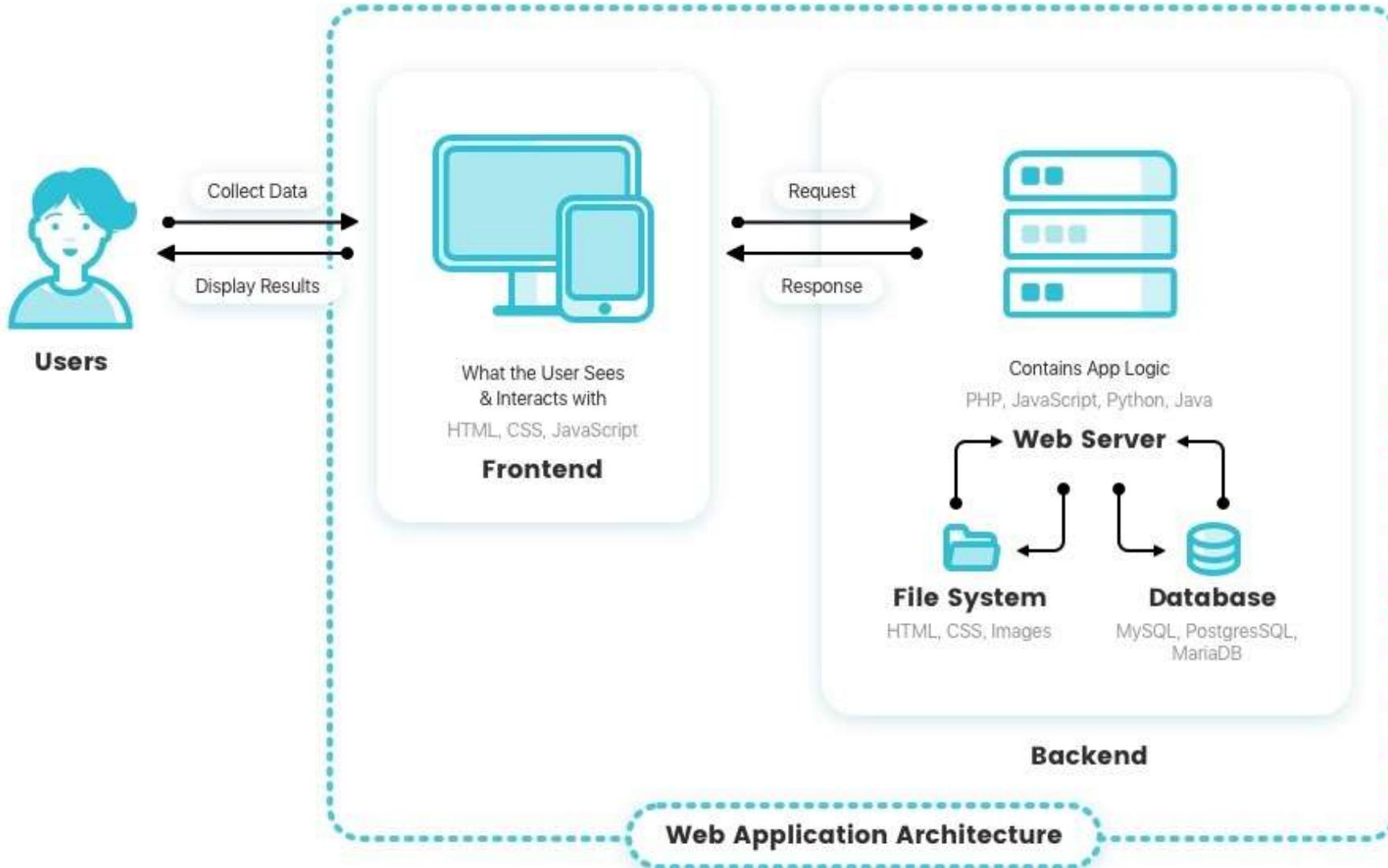
A 2 Ae 2 2 2 p



A 2 A e 2 2 2 p



A 2 Ae 2 2 2 p



A      2      A e    2    2    2      p



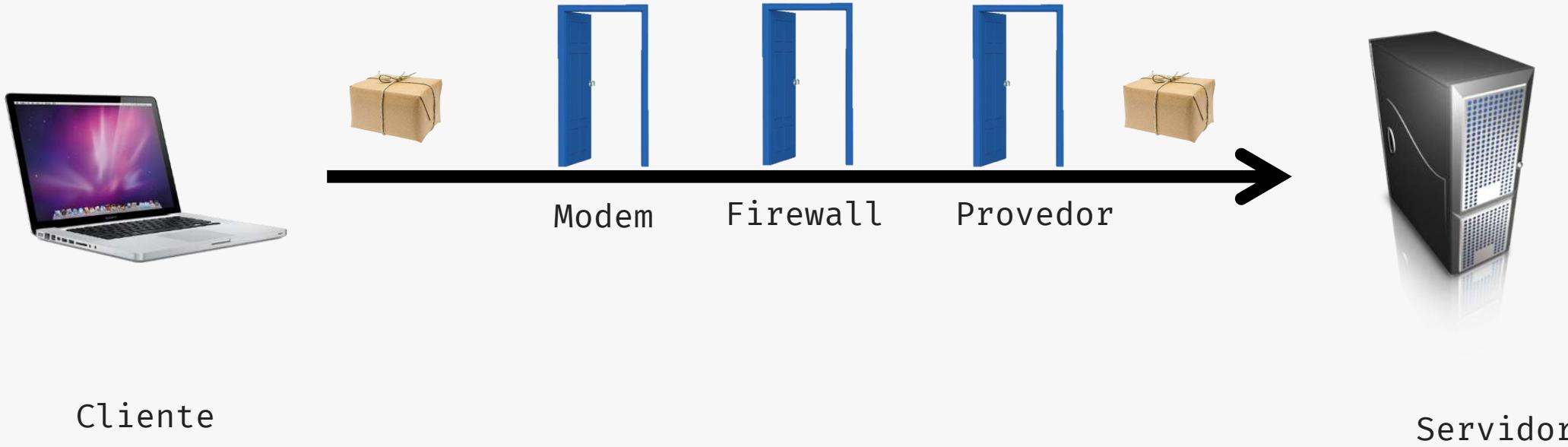
Cliente



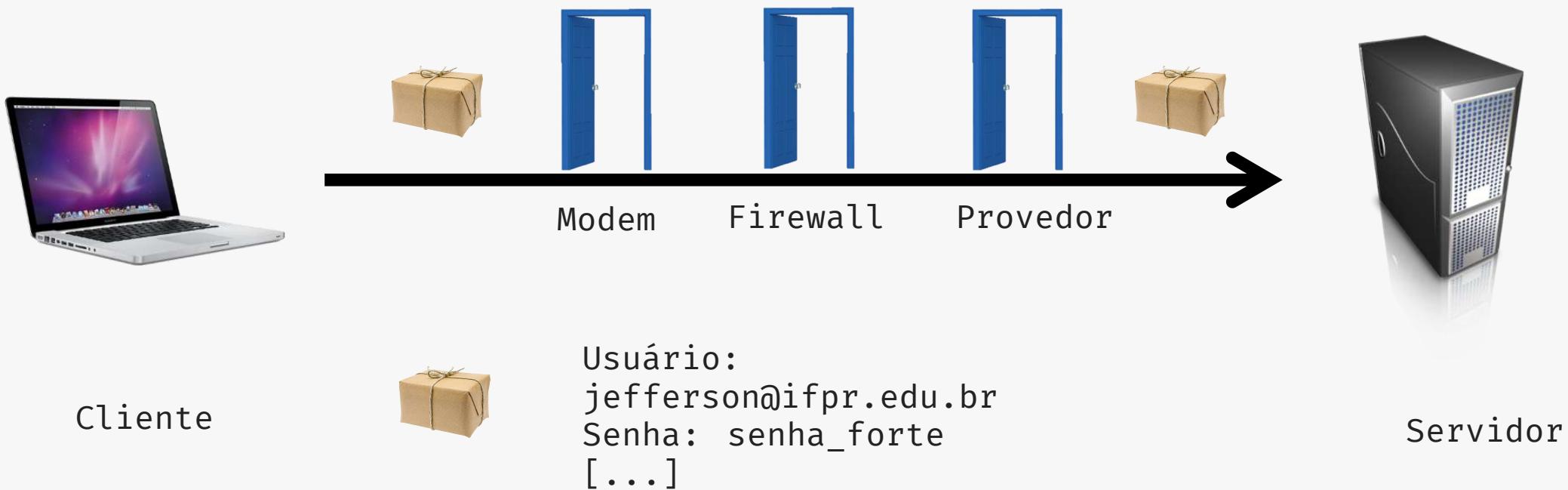
Servidor



A 2 Ae 2 2 2 p



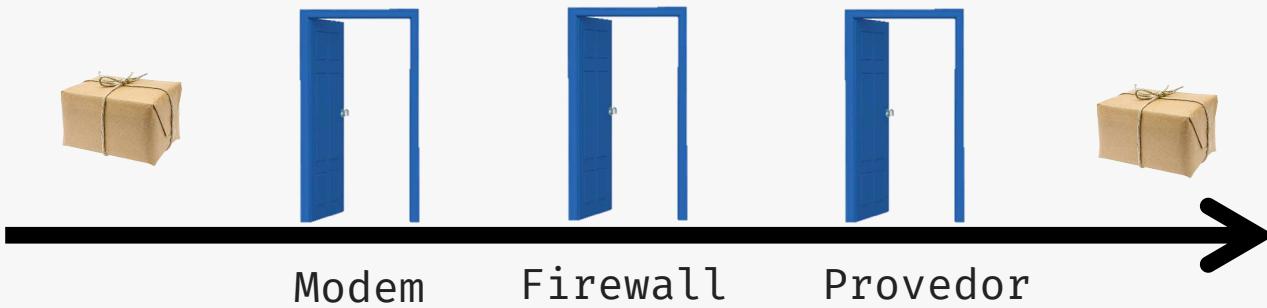
A 2 Ae 2 2 2 p





**A****e**

Cliente



\$2y\$10\$Q5iz729StvpUp2jXXsGU3  
uUdIrIAKKq5.FwTQXv.YAxUcbDC8  
UffK



Servidor

## 4 d Ad2

- ∴ O que acontece com nossos dados quando usamos HTTP , ou seja sem a letra S ao final?
- a. Os dados são transportados em texto puro para o servidor, visível para qualquer um.
  - b. Os dados são criptografados, para impedir a visualização por intermediários.
  - c. Usamos automaticamente um certificado digital para provar a identidade de um site.

## 2

- ∴ Por padrão, os dados são trafegados como **texto puro** na web (apenas com HTTPS a web é segura);
- ∴ O protocolo HTTPS nada mais é do que o protocolo HTTP mais uma camada adicional de segurança, a TLS/SSL;
- ∴ O tipo de criptografia utilizado na web é o de chave pública/chave privada;
- ∴ Certificados possuem identidade e validade;
- ∴ As chaves públicas estão no certificado, a chave privada fica apenas no servidor;

A

## 2 Ae2

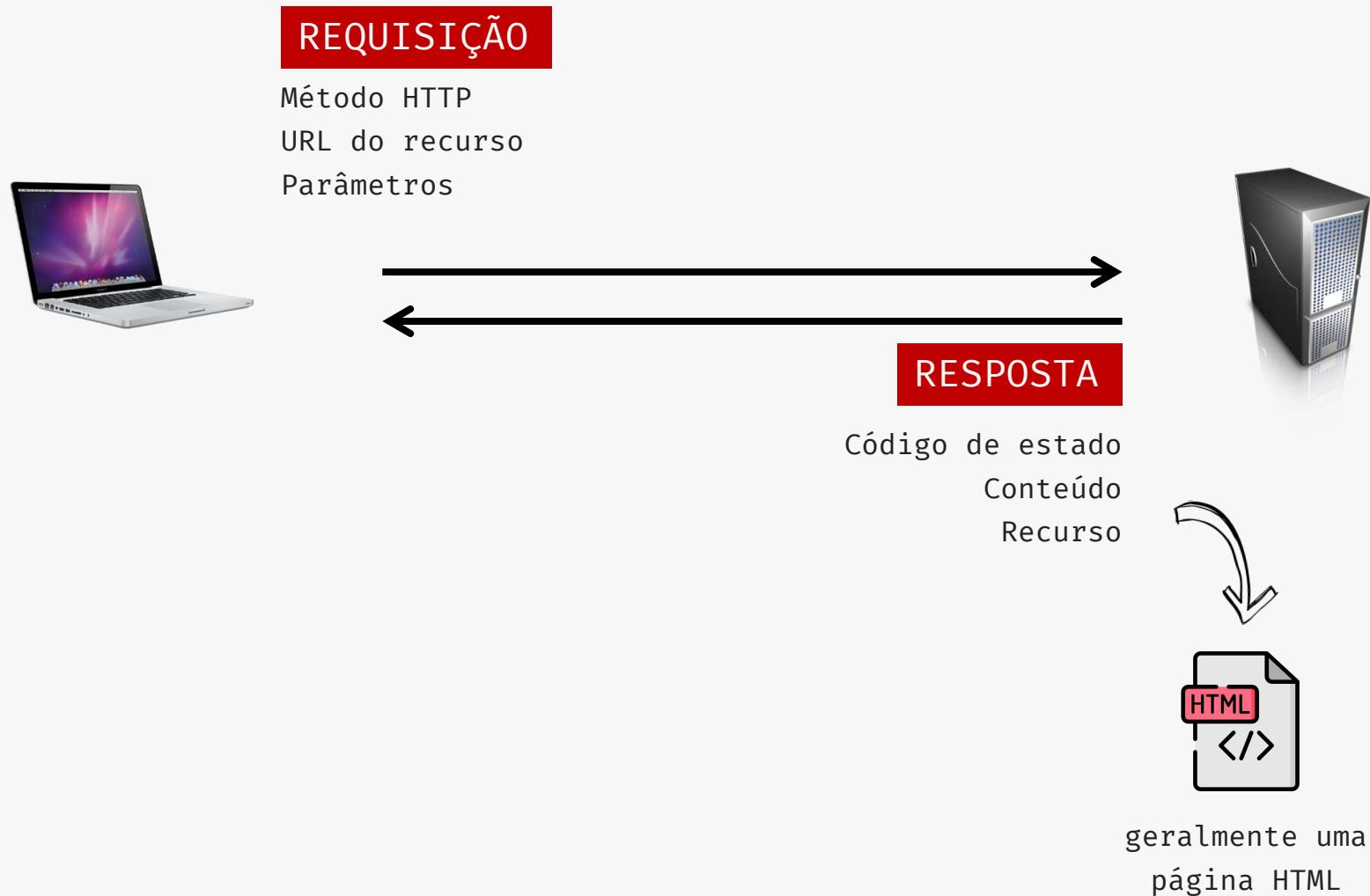
## 2

- ∴ O protocolo HTTP define uma série de interfaces para requisição e resposta;
- ∴ A requisição é padronizada por meio de **verbos**:
  - ∴ POST, GET, PUT, DELETE, PATCH, OPTINS, etc.
- ∴ A resposta é padronizada por meio de **código de resposta** ou **códigos de estados**:
  - ∴ 200, 301, 404, 500;

# 4 p p2 2 i X

- :: HTTP define um conjunto de métodos de requisição indicando a ação desejada a ser executada em um recurso.
- :: Embora eles também possam ser substantivos, essas solicitações são às vezes chamadas de verbos HTTP;
- :: As requisições mais comuns são GET e POST:
  - :: O método GET solicita uma representação de dados do recurso especificado.
  - :: Requisições usando GET só devem retornar dados.
  - :: O método POST envia dados para um servidor para que possa alterar seu estado.
  - :: Esse é o método geralmente usado para formulários HTML.

e

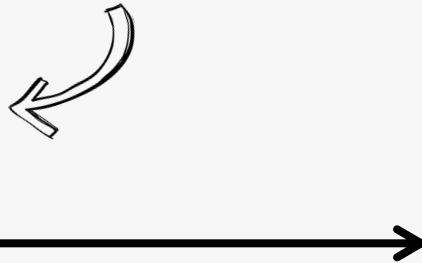


e

O usuário clica num **link** ou **digita a URL** de algum recurso no browser



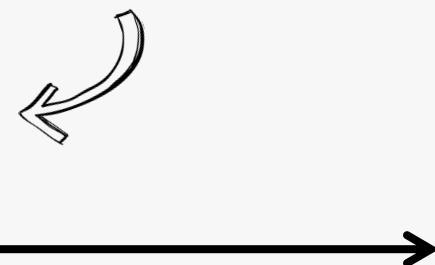
REQUISIÇÃO  
MÉTODO GET



O usuário preenche um **formulário** e envia os dados para serem processados



REQUISIÇÃO  
MÉTODO POST



e

i X pA 2

i X

[MÉTODO] [URI] HTTP/[Versão]  
[Cabeçalhos]

[CORPO/PAYLOAD]

POST /produtos HTTP/1.1  
Content-Type: application/json  
Accept: application/json

```
{  
    "nome": "Notebook i7",  
    "preco": 2100.0  
}
```

# e i X pA 2 A

HTTP/[Versão] [STATUS]  
[Cabeçalhos]  
[CORPO]

HTTP/1.1 201 Created  
Location: /produtos/331  
Content-Type: application/json

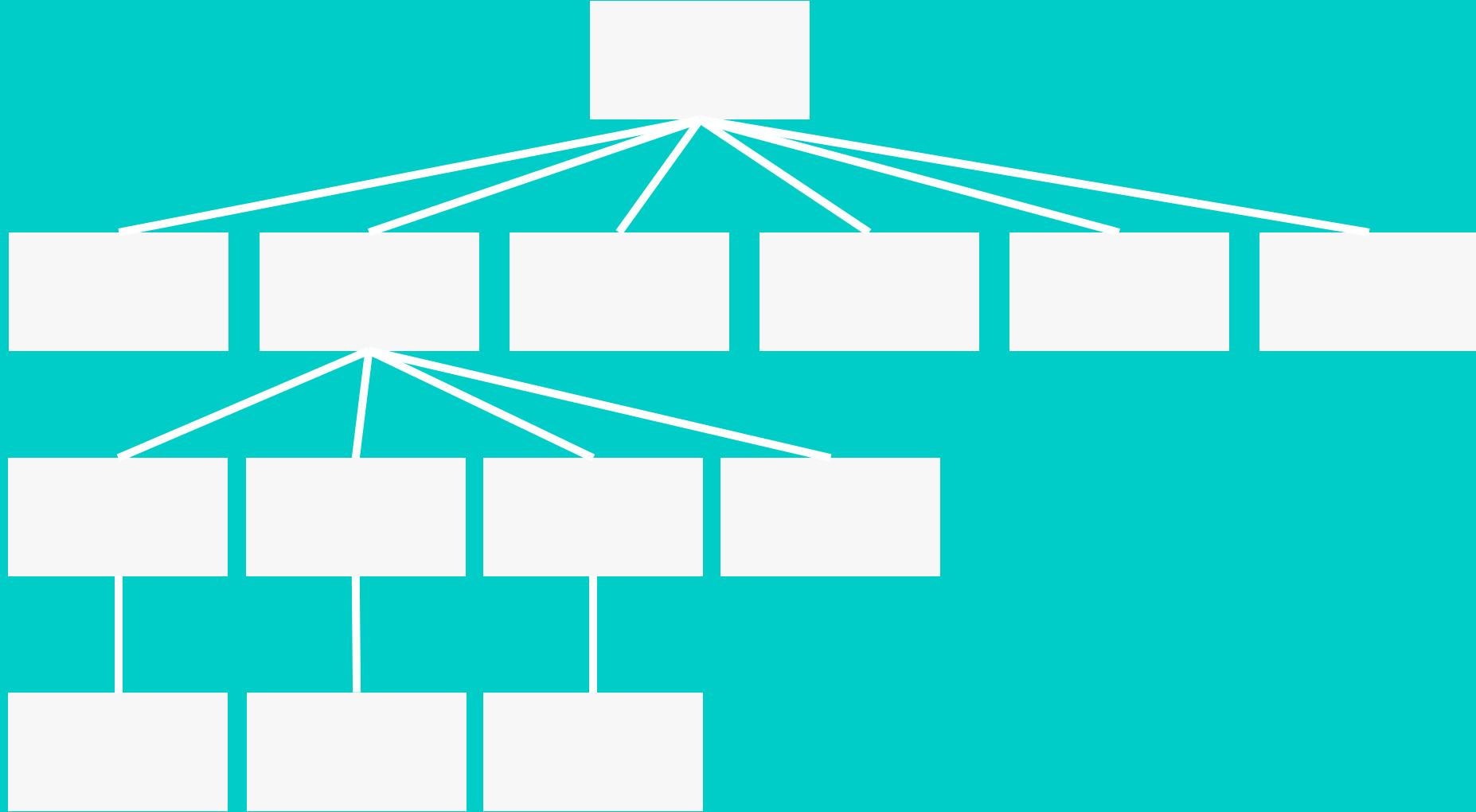
```
{  
  "codigo": 331,  
  "nome": "Notebook i7",  
  "preco": 2100.0  
}
```

A

A

**2 p2 2i**

p



**2 p2 2i**

**nslookup ifpr.edu.br**

# 2 p2 2i

- Base URL

`https://api.vinhos.com`

- Resource (recurso) ou Path

`https://api.vinhos.com/vinhos`

- Query Params

`https://api.vinhos.com/vinhos?pais=brasil&regiao=sul`

- Route Params

`https://api.vinhos.com/vinhos/brasil/sul`

A A

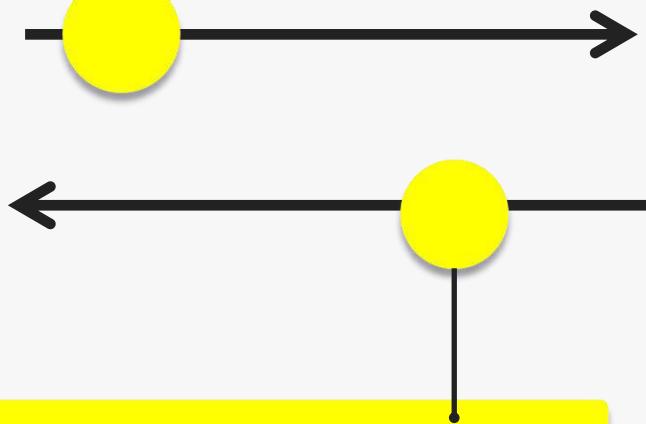
2 p 2 p2A eAi X 2 2 e A 2

- ∴ Servidor de aplicação:

- ∴ Suporta o conjuntos de especificação do JEE (Processamento distribuido, gerenciamento de aplicações, programação em cluster, JPA, EJB) (Jboss);

- ∴ Servlet Container

- ∴ Servidor que suporta apenas as parte das especifcações JEE relacionadas a web .



2 O módulo de container web “transforma” a requisição em um objeto **HttpServletRequest**.

Servidor web  
(Glassfish, Jboss, Tomcat, etc)

Container web  
(Tomcat, Jetty, Grizzly )

Componentes web  
(Servlet, JSP, etc)

5 O container “transforma” o objeto **HttpServletResponse** em um resposta HTTP e o envia para o cliente

4 O componente pode, ou gerar o objeto **HttpServletResponse**, ou despachar a requisição para outro componente processar.

3 O componente interage com um SGBD, Bean, etc, para gerar o conteúdo de em resposta a requisição.



# e A d pAp2

| TOMCAT VERSION | JAVA EE* | SERVLET | JSP | EL  | WEBSOCKET | MIN. JAVA SE VERSION |
|----------------|----------|---------|-----|-----|-----------|----------------------|
| 3.3.x          | 1.2      | 2.2     | 1.1 | -   | -         | 1.1                  |
| 4.1.x          | 1.3      | 2.3     | 1.2 | -   | -         | 1.3                  |
| 5.5.x          | 1.4      | 2.4     | 2.0 | -   | -         | 1.4                  |
| 6.0.x          | 5        | 2.5     | 2.1 | 2.1 | -         | 5.0                  |
| 7.0.x          | 6        | 3.0     | 2.2 | 2.2 | -         | 6                    |
| 8.0.x          | 7        | 3.1     | 2.3 | 3.0 | 1.0       | 7                    |

e A d pAp2

A A → OA



# d Ap

perguntas?

[Jefferson.chaves@ifpr.edu.br](mailto:Jefferson.chaves@ifpr.edu.br)

45 998508359

[github.com/jeffersonchaves](https://github.com/jeffersonchaves)