



# FUNDAMENTOS DO DESENVOLVIMENTO WEB

Prof. Esp. Wagner Ito de Cabral

# Visão Geral

Untitled - TextEdit



File Edit View Help

03	APRESENTAÇÃO
04	HISTÓRIA DA INTERNET
13	INFRAESTRUTURA DE REDES
15	PROTOCOLO HTTP
20	STATUS CODE
22	CLIENT SIDE X SERVER SIDE
24	TAREFA FINAL (T.F)



# **SOBRE O PROFESSOR**

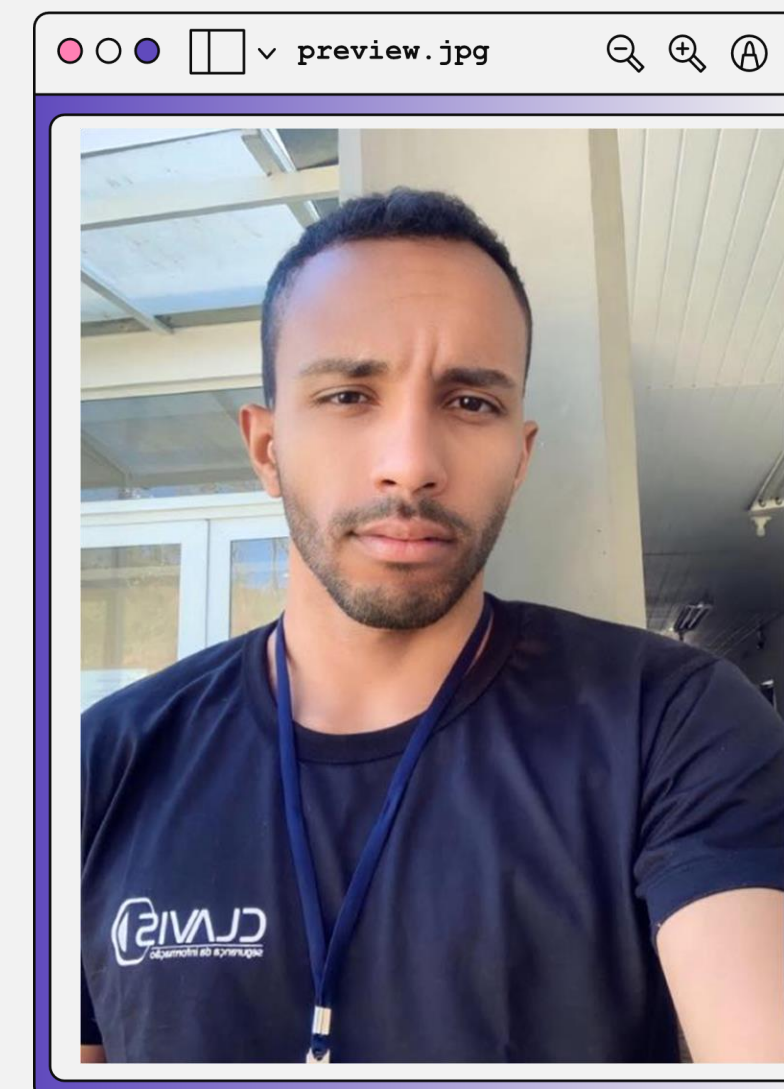


- **Técnico de Informática**
- **Análise e Desenvolvimento de Sistemas**
- **Redes de Computadores e Comunicação de Dados**
- **Cyber Segurança Ofensiva**
- **Engenharia da Computação**

**I. Professor**

**II. Gestor de T.I / Finanças**

**III. Analista de Infraestrutura de Redes**





**O Plano de estudos da disciplina pode ser acessado diretamente do portal do aluno.**

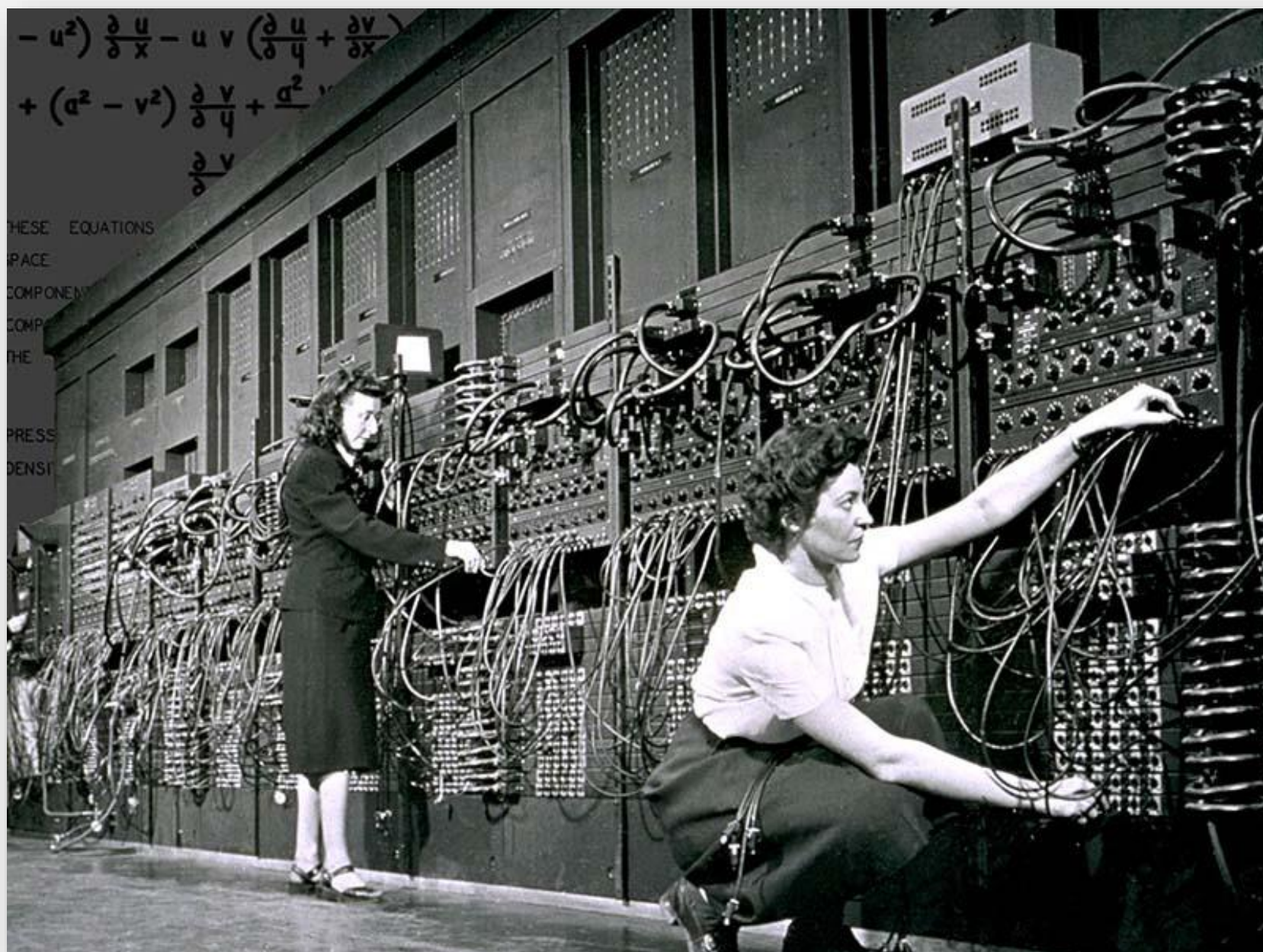


# Ferramentas Utilizadas





# O PRIMEIRO COMPUTADOR



Em fevereiro de 1946, o primeiro computador eletrônico da história era apresentado. O ENIAC (Electronic Numerical Integrator and Computer) tinha cerca de dois metros de altura, pesava 30 toneladas e ocupava 180 metros quadrados.

Encomendado pelo exército dos Estados Unidos, a máquina servia para fazer cálculos – a palavra “computador” vem de computar, sinônimo de calcular. Apesar de ser um gigante, o ENIAC tinha uma capacidade operacional menor do que qualquer calculadora de mão vendida atualmente.

Ele fazia cerca de 4,5 mil cálculos por segundo.

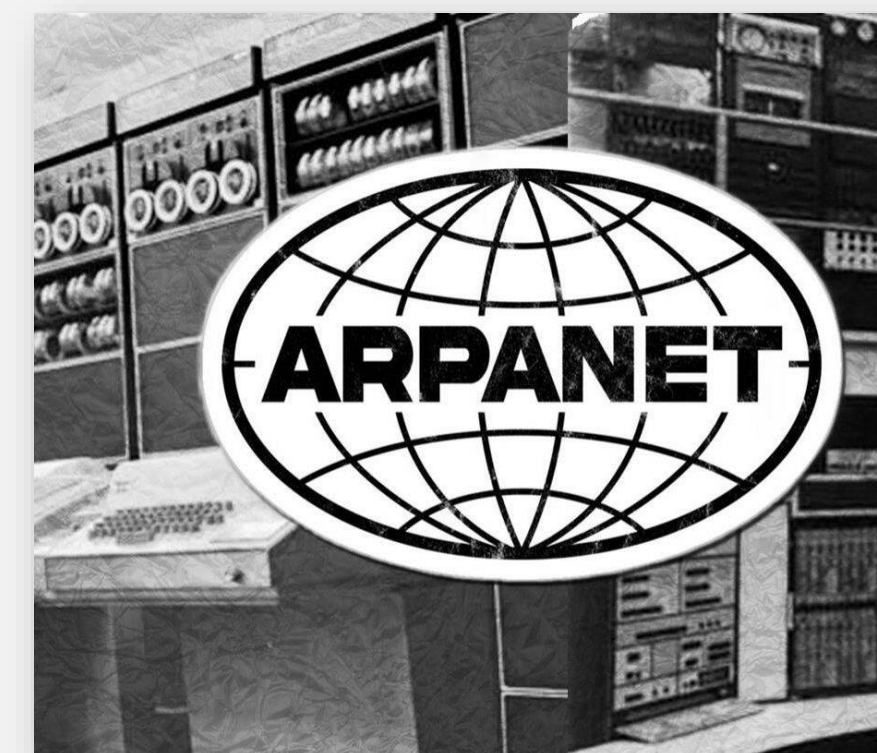


## Anos 1960 – A Origem: ARPANET

- ◆ Durante a Guerra Fria, os EUA buscavam uma forma de comunicação descentralizada para resistir a ataques nucleares.
- ◆ A **ARPANET** (Advanced Research Projects Agency Network) foi criada pelo **Departamento de Defesa dos EUA** em **1969**.
- ◆ Primeiro teste: uma mensagem foi enviada entre a **UCLA** e o **Stanford Research Institute**.

### ✓ Marcos importantes:

- ✚ 1969 – Primeira conexão entre computadores.
- ✚ 1971 – Ray Tomlinson cria o primeiro sistema de e-mail.
- ✚ 1973 – ARPANET começa a se expandir internacionalmente.







## Anos 1970 – A Invenção do TCP/IP

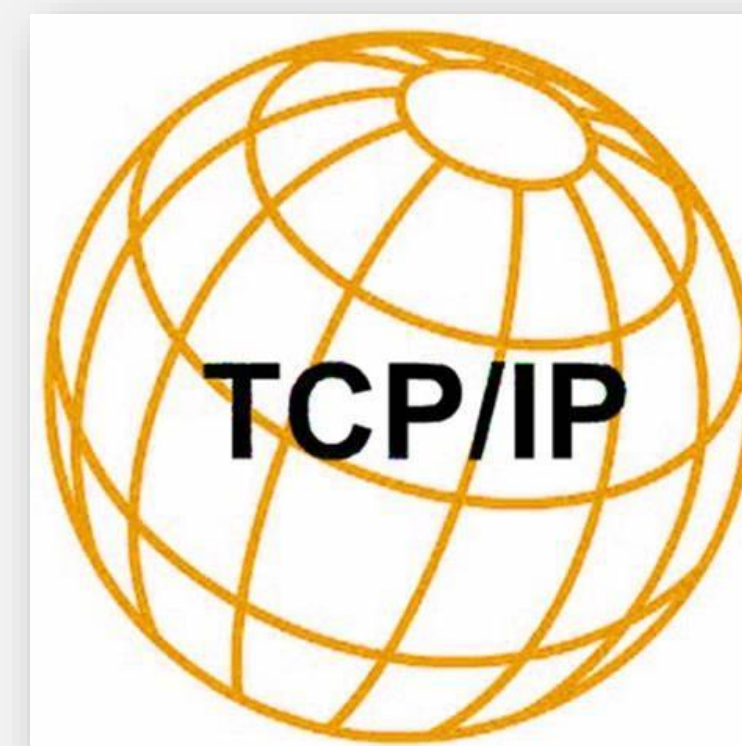
◆ **Vinton Cerf e Robert Kahn** desenvolveram o **protocolo TCP/IP** (Transmission Control Protocol/Internet Protocol).

◆ Em **1º de janeiro de 1983**, a ARPANET adotou oficialmente o TCP/IP, tornando-se a base da **Internet moderna**.

### ✓ Avanços na década de 70:

✂ 1974 – Primeira vez que o termo **"Internet"** foi usado.

✂ 1978 – Criação do primeiro **spam** (e-mail não solicitado).







## Anos 1980 – Expansão para Universidades

- ◆ A Internet começou a crescer além do uso militar.
- ◆ Redes como **NSFNET** (National Science Foundation Network) conectavam universidades.
- ◆ Surgiram os primeiros **domínios (.com, .org, .edu, .gov, .net)** em 1985.

### ✓ Destaques da década de 80:

- ✂ 1983 – ARPANET adota TCP/IP, tornando-se oficialmente "Internet".
- ✂ 1989 – **Tim Berners-Lee** propõe a **World Wide Web (WWW)** no CERN.



## Anos 1990 – O Boom da Internet 🌐

- ◆ **1991** – A **World Wide Web (WWW)** foi disponibilizada publicamente.
- ◆ **1993** – Surgiu o **Mosaic**, o primeiro navegador gráfico.
- ◆ **1994** – Empresas como Amazon e Yahoo nasceram.

### ✓ Marcos dessa era:

- ✚ 1995 – Microsoft lança o **Internet Explorer**.
- ✚ 1998 – Surge o **Google**.
- ✚ 1999 – Primeiros serviços de música online como o **Napster**.



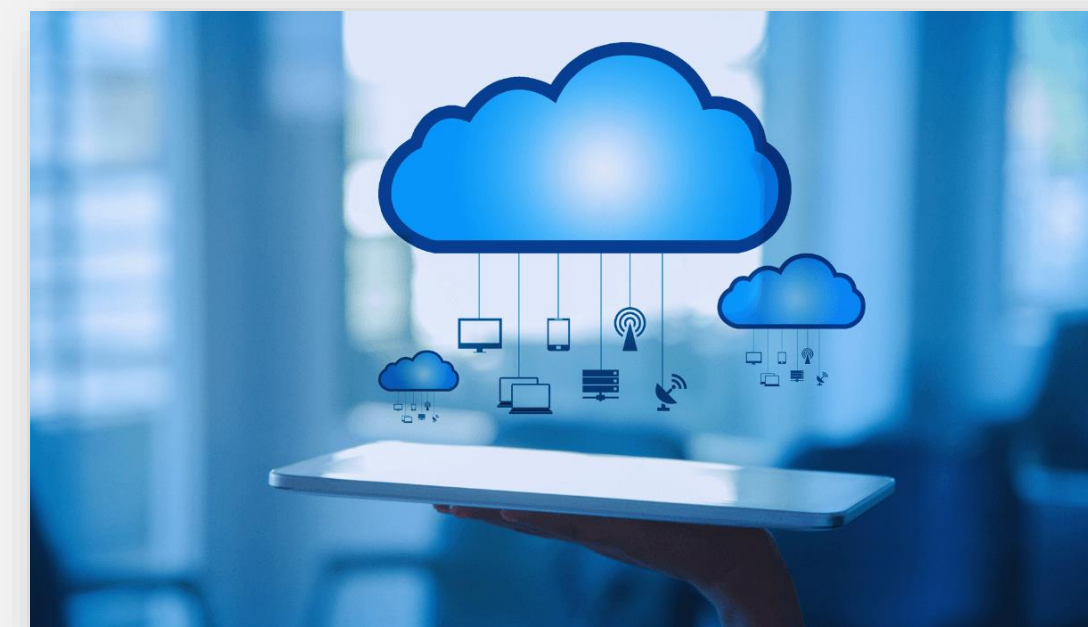


## Anos 2010 – Internet das Coisas (IoT) e a Nuvem

- ◆ A computação em nuvem permitiu o armazenamento de dados online.
- ◆ Aplicativos como WhatsApp, Instagram e TikTok dominaram o mercado.
- ◆ A **Internet das Coisas (IoT)** conectou dispositivos do dia a dia (smart TVs, assistentes virtuais, carros inteligentes).

### ✓ Eventos importantes:

- 📌 2010 – Instagram é lançado.
- 📌 2015 – Inteligência Artificial começa a ser integrada à Internet (assistentes como Alexa e Google Assistant).





## Presente e Futuro – 5G, IA e Web3 🚀

- ◆ O **5G** promete velocidades ultra rápidas.
- ◆ O conceito de **Web3** traz uma Internet descentralizada com blockchain.
- ◆ A **Inteligência Artificial** está cada vez mais integrada à navegação online.

### ✓ **Tendências:**

- ✚ Expansão do **Metaverso**.
- ✚ Mais segurança e privacidade com **criptografia**.
- ✚ Desenvolvimento da **Internet espacial** (como o Starlink).



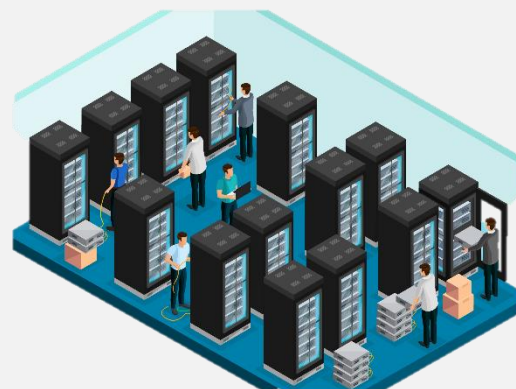




# Principais Componentes



**ROTEADOR/GATEWAY**



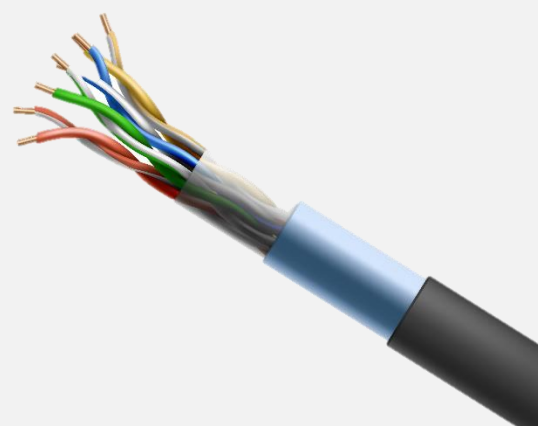
**SERVIDOR/DATACENTER**



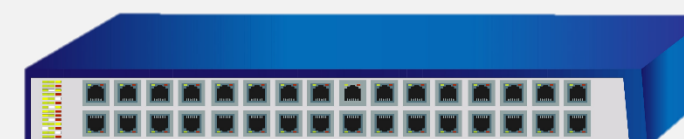
**ACCES POINT**



**FIREWALL**

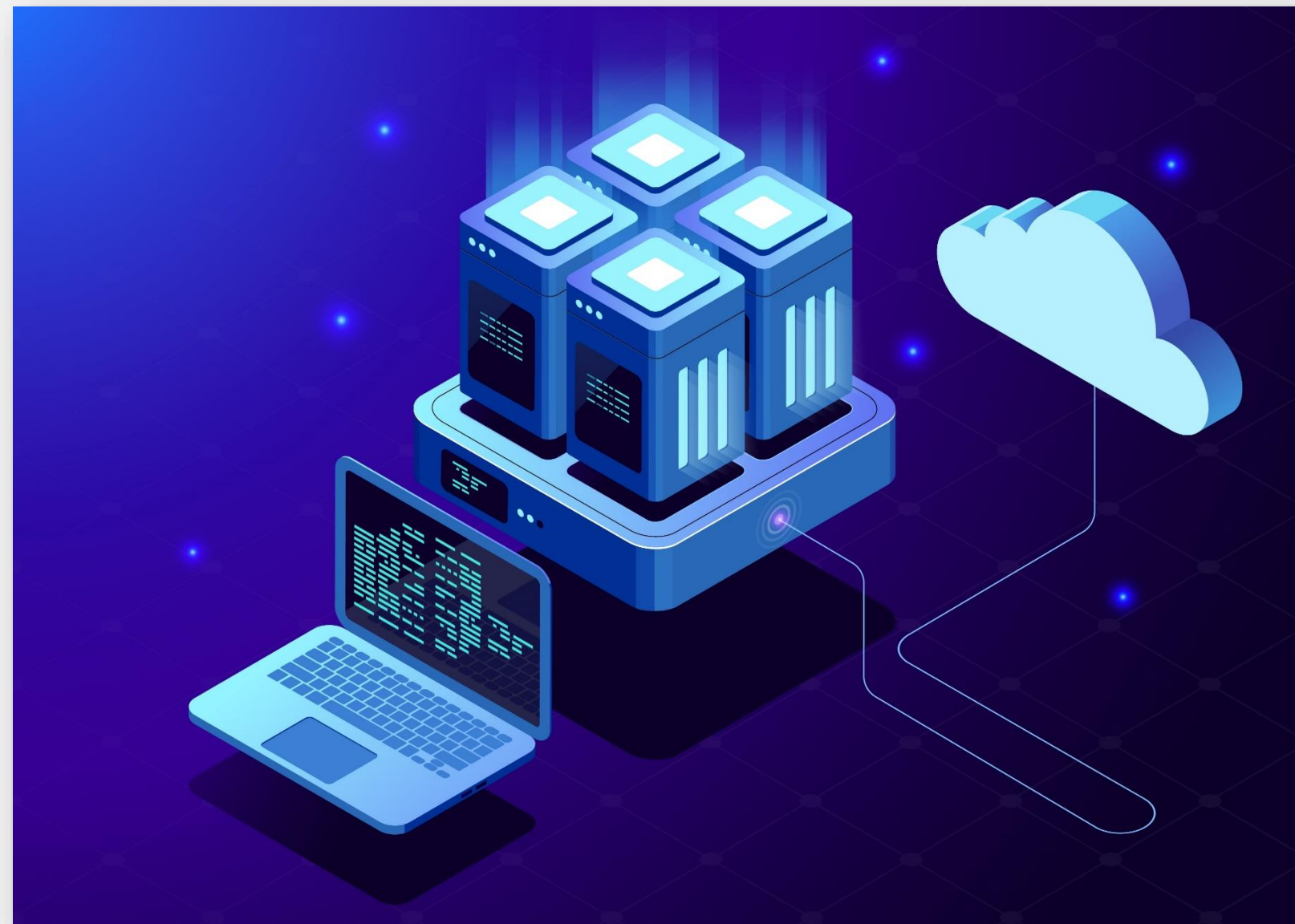


**CABOS DE REDE**



**SWITCH**

**O QUE VOCE ENTENDE SOBRE CLOUD (NUVEM)?**





# PROTOCOLO HTTP

Um **protocolo** é um conjunto de regras e padrões que define como a comunicação entre dispositivos deve acontecer.  
Ele estabelece como os dados são formatados, transmitidos, recebidos e interpretados em uma rede.

## Principais Características de um Protocolo

- ✓ **Definição de regras** → Especifica como a comunicação ocorre.
- ✓ **Interoperabilidade** → Permite que sistemas diferentes se comuniquem.
- ✓ **Organização em camadas** → Normalmente seguem modelos como o **OSI** e o **TCP/IP**.
- ✓ **Controle de erros** → Alguns protocolos garantem a entrega correta dos dados.



## Exemplos de Protocolos

- ✚ **HTTP/HTTPS** → Comunicação na web.
- ✚ **TCP/IP** → Base da Internet.
- ✚ **FTP** → Transferência de arquivos.
- ✚ **SMTP/POP3/IMAP** → Envio e recebimento de e-mails.
- ✚ **DNS** → Conversão de domínios em endereços IP.
- ✚ **Wi-Fi (802.11)** → Comunicação sem fio.





## Fundamentos do HTTP (Hypertext Transfer Protocol)

O **HTTP (Hypertext Transfer Protocol)** é um protocolo de comunicação utilizado para a transferência de dados na web. Ele define como as mensagens são formatadas e transmitidas entre clientes (navegadores) e servidores web.

## Características Principais do HTTP

### ✚ 1.1. Baseado em Cliente-Servidor

- O cliente (geralmente um navegador) inicia uma requisição para um servidor web.
- O servidor processa a solicitação e retorna uma resposta ao cliente.

### ✚ 1.2. Protocolo Sem Estado (Stateless)

- Cada requisição HTTP é independente, ou seja, o servidor não mantém informações sobre requisições anteriores.
- Para armazenar sessões, utiliza-se **cookies, tokens ou mecanismos como o HTTP Sessions**.

### ✚ 1.3. Texto Simples e Legível

- As mensagens HTTP são baseadas em texto, o que facilita a depuração e análise.

### ✚ 1.4. Utiliza o Modelo de Requisição-Resposta

- O cliente envia uma **requisição** e o servidor retorna uma **resposta**.



# PROTOCOLO HTTP



Os métodos HTTP definem a ação a ser executada na requisição.

Os principais são:

Método	Descrição
<b>GET</b>	Solicita um recurso sem modificar dados.
<b>POST</b>	Envia dados para o servidor (ex.: formulários).
<b>PUT</b>	Atualiza um recurso existente ou cria se não existir.
<b>DELETE</b>	Remove um recurso específico.
<b>PATCH</b>	Modifica parcialmente um recurso existente.
<b>HEAD</b>	Igual ao GET, mas retorna apenas os cabeçalhos.



# PROTOCOLO HTTP

## Códigos de Status HTTP (STATUS CODE)

Os códigos de status indicam o resultado da requisição:

**1xx (Informativo):** Processamento em andamento.

**2xx (Sucesso):** Requisição bem-sucedida.

- **200 OK** → Requisição bem-sucedida.
- **201 Created** → Recurso criado com sucesso.

**3xx (Redirecionamento):** Cliente precisa tomar ação adicional.

- **301 Moved Permanently** → Página movida permanentemente.
- **302 Found** → Redirecionamento temporário.

**4xx (Erro do Cliente):** Ocorreu um erro na solicitação.

- **400 Bad Request** → Requisição inválida.
- **401 Unauthorized** → Requer autenticação.
- **403 Forbidden** → Acesso negado.
- **404 Not Found** → Recurso não encontrado.

**5xx (Erro do Servidor):** Erro interno do servidor.

- **500 Internal Server Error** → Erro geral do servidor.
- **503 Service Unavailable** → Servidor temporariamente indisponível.





PROTOCOLO HTTP



## TTP vs. HTTPS

A principal diferença entre HTTP e HTTPS está na segurança:

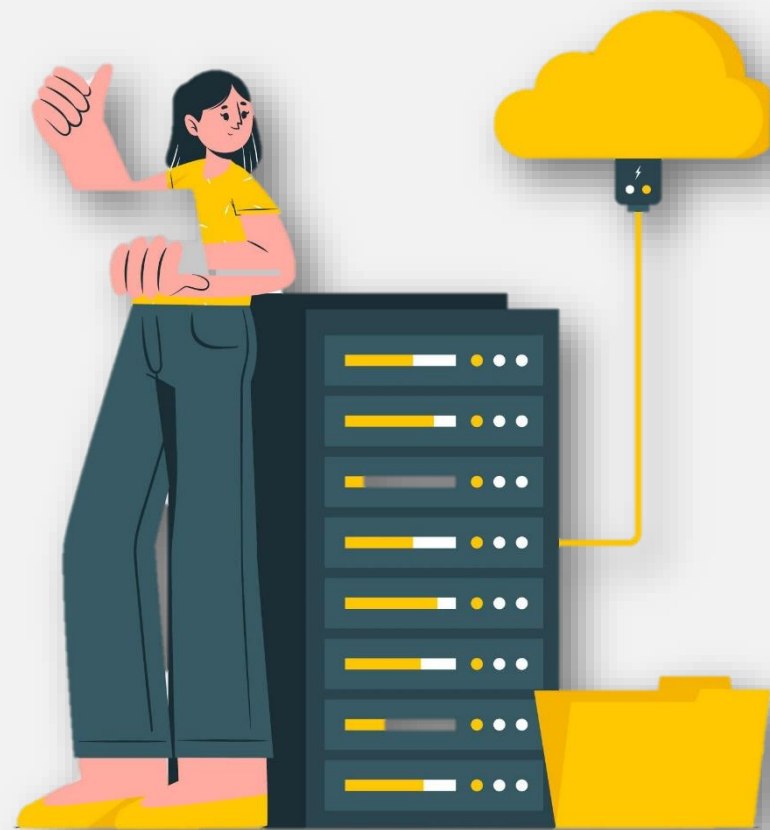
Protocolo	Segurança	Porta Padrão
HTTP	Não utiliza criptografia	80
HTTPS	Utiliza criptografia (SSL/TLS)	443



# Client-Side vs. Server-Side

## O que é Client-Side?

- ◆ **Significa "lado do cliente"** – Refere-se a tudo que acontece no dispositivo do usuário (como um navegador ou um aplicativo).
- ◆ **É executado no navegador** – O código é baixado e processado localmente.
- ◆ **Usa tecnologias como:**
  - **HTML, CSS** (estrutura e estilo da página).
  - **JavaScript** (interatividade e manipulação de elementos).
  - **Frameworks como React, Angular, Vue.js** (melhoria na experiência do usuário).
- ◆ **Exemplos de funcionalidades Client-Side:**
  - ✓ Validação de formulários no navegador.
  - ✓ Animações e interatividade com JavaScript.
  - ✓ Atualização de conteúdo sem recarregar a página (AJAX).



## Client-Side vs. Server-Side

### O que é Server-Side?

◆ **Significa "lado do servidor"** – O processamento acontece no servidor antes de enviar os dados ao cliente.

### ◆ Usa tecnologias como:

• **Linguagens:** PHP, Python, Node.js, Java, Ruby, C#.

• **Banco de Dados:** MySQL, PostgreSQL, MongoDB.

• **Frameworks:** Laravel (PHP), Django (Python), Express (Node.js), Spring (Java).

### ◆ Exemplos de funcionalidades Server-Side:

- ✓ Autenticação de usuários.
- ✓ Processamento de dados em banco de dados.
- ✓ Geração de páginas dinâmicas (como em um blog ou e-commerce).



- 1. GRUPOS DE ATÉ 4 PESSOAS.**
- 2. ACESSE O SITE SUGERIDO PELO PROFESSOR.**
- 3. ESCOLHA UM NICHOS (Ex. Culinária, Contabilidade, Games)**
- 4. DESENVOLVA O LAYOUT DA PAGINA E EM SEGUIDA MANDE UM PRINT AO PROFESSOR.**
- 5. ENVIO INDIVIDUAL POR R.A NO PORTAL DO ALUNO.**