INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA RIO GRANDE DO NORTE

**Professor: Macêdo Firmino**
**Disciplina: Arquitetura de Rede**
**Prática 18: Sistema Web e Protocolo HTTP**

Na aula de hoje iremos discutirmos as funcionalidades do protocolo de aplicação utilizado para a transferência de dados na Web. Por último iremos realizar algumas experiência com um analisar de tráfego (Wireshark) sobre o respectivo protocolo.

## Sistema Web

A World Wide Web (WWW), comumente conhecida como web, é um sistema de armazenamento e distribuição de documentos hipermídia que utiliza a internet. Criada por Sir Tim Berners-Lee, um cientista da computação britânico, em 1989, a web revolucionou a maneira como as pessoas acessam, compartilham e interagem com informações em todo o mundo. A *Web* nasceu da necessidade de fazer com que grupos de cientistas de diferentes nacionalidades pudessem colaborar uns com os outros compartilhando conhecimento.

O conceito fundamental por trás da World Wide Web é a ideia de hipertexto, que permite a criação de documentos digitais interconectados através de links. Esses documentos, chamados de páginas da web (HTML), podem conter diversos tipos de conteúdo, incluindo texto, imagens, vídeos e outros recursos multimídia. Os links, ou hiperlinks, permitem que os usuários naveguem de uma página para outra, criando uma teia interconectada de informações.

Os componentes essenciais que compõem a World Wide Web incluem:

- URLs (Uniform Resource Locators): são endereços que identificam de maneira única recursos na web. As URLs permitem que os usuários acessem páginas específicas por meio de navegadores.
- HTTP (Hypertext Transfer Protocol): é um protocolo de comunicação que permite a transferência de informações, como texto e multimídia, entre o navegador do usuário e o servidor web. O HTTP é a base para a comunicação na web.
- HTML (Hypertext Markup Language): é uma linguagem de marcação utilizada para estruturar o conteúdo das páginas da web. Permite a criação de links, formatação de texto e incorporação de elementos multimídia.
- Navegadores Web: são aplicativos que permitem aos usuários acessar e visualizar páginas da web. Exemplos incluem Google Chrome, Mozilla Firefox e Safari.
- Servidores Web: são programas que armazenam e fornecem páginas da web quando solicitadas pelos usuários. Eles utilizam o protocolo HTTP para a entrega de conteúdo.

## URL

Um cliente que deseja acessar uma página Web precisa de seu endereço (URL). A URL (*Uniform Resource Locator*) é um padrão para a especificação de qualquer tipo de informação na Internet. Uma URL é constituída por quatro partes: protocolo, equipamento (IP), porta e caminho (pastas).


Cada parte de uma URL desempenha um papel específico na especificação do caminho para acessar esse recurso. Inicialmente, temos o protocolo de comunicação a ser utilizado, como “http://” para páginas da web não criptografadas ou “https://” para páginas seguras.

Na sequência, deverá ser informado o equipamento (host) que identifica o servidor onde o recurso está localizado. Pode ser um domínio, como “www.exemplo.com”, ou um endereço IP.

O número da porta é opcional e especifica a porta de comunicação no servidor. Se ausente, é assumido o valor padrão para o protocolo (por exemplo, 80 para HTTP, 443 para HTTPS).

Por último, temos o caminho que indica a localização específica do recurso no servidor. Pode incluir diretórios e subdiretórios, como “/pasta/subpasta/pagina.html”.

Alguns sites utilizam ainda passagens de parâmetros pela URL. Estes parâmetros podem fornecer informações adicionais sobre a requisição. Eles são separados do caminho por “?” e entre si por “&”. Por exemplo, “?id=123&modo=editar”.

## Processo de Acesso a um Site

Suponha que Bob liga seu notebook e o conecta por um cabo Ethernet ao switch da escola, o qual, por sua vez, está conectado ao roteador da instituição. Após obter corretamente as configurações de rede, seu notebook passa a ter acesso à Internet.

Ao digitar o URL www.google.com no navegador, Bob desencadeia uma série de etapas que resultarão na exibição da página inicial do Google. As principais etapas são:

**01.** Criação e envio da consulta DNS: o sistema operacional do notebook gera uma mensagem de consulta DNS, inserindo "www.google.com" no campo de pergunta. Essa consulta é encapsulada em um segmento UDP destinado à porta 53 e, em seguida, colocada em um datagrama IP direcionado ao servidor DNS configurado.

**02.** Encaminhamento da consulta pela rede: a mensagem deixa o notebook e passa pelos dispositivos intermediários da rede (switch e roteador), seguindo a lógica de encaminhamento IP, até chegar ao servidor DNS.

**03.** Processamento da consulta pelo servidor DNS: o servidor DNS recebe o datagrama, extrai a consulta e procura pelo nome www.google.com em sua base de dados. Encontrando o registro correspondente, com o endereço IP (por exemplo, 64.233.169.105), ele cria uma mensagem DNS de resposta contendo o mapeamento nome em IP. Essa resposta é encapsulada em um segmento UDP e enviada de volta ao notebook de Bob.

**04.** Recebimento da resposta DNS: a resposta chega ao notebook, que extrai o endereço IP (64.233.169.105) do servidor www.google.com.

**05.** Estabelecimento da conexão TCP: de posse do endereço IP, o notebook cria um socket TCP e inicia o processo de conexão enviando um segmento TCP SYN para a porta 80 (HTTP) do servidor, dentro de um datagrama IP destinado ao endereço 64.233.169.105. Este é o início do three-way handshake.

**06.** Resposta do servidor web: o servidor www.google.com recebe o SYN, demultiplexa o segmento para a porta 80, cria um socket para a nova conexão e responde com um segmento SYN-ACK, novamente encapsulado em um datagrama IP de volta para Bob.

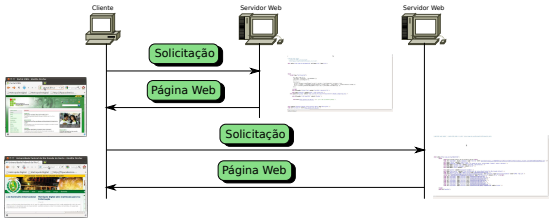
**07.** Conclusão do handshake: o datagrama com o SYN-ACK percorre as redes da Internet até chegar ao notebook de Bob. O sistema operacional demultiplexa o segmento para o socket correspondente, envia um ACK ao endereço do Google e entra no estado de conexão estabelecida.

**08.** Envio da requisição HTTP: com a conexão TCP pronta, o navegador cria uma mensagem HTTP GET contendo a URL requisitada. Essa mensagem é enviada pelo socket, tornando-se a carga útil do segmento TCP, que é encapsulado em um datagrama IP e enviado ao servidor Google.

**09.** Processamento da requisição pelo servidor HTTP: o servidor www.google.com lê o HTTP GET recebido, gera uma mensagem de resposta HTTP e inclui no corpo da resposta o conteúdo da página solicitada. Essa resposta é enviada pelo socket TCP.

**09.** Recebimento e exibição da página: o datagrama contendo a resposta HTTP chega ao notebook de Bob. O navegador lê o corpo da mensagem, extrai o código HTML e monta a página Web.

**10.** Execução de scripts da página: caso a página contenha scripts (como JavaScript), o navegador os executa localmente para adicionar interatividade, validação de formulários, animações ou atualizações dinâmicas.




O navegador pode armazenar dados localmente, como ccookies e cache, para melhorar a experiência do usuário e manter informações entre diferentes sessões.

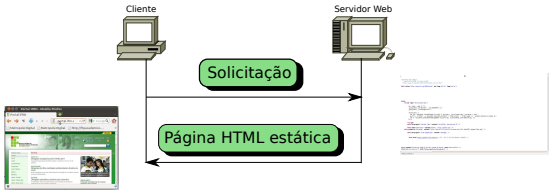
A Web utiliza a arquitetura cliente-servidor. Nela existem em dois personagens principais: o cliente, que solicita serviços ou recursos, e o servidor, que fornece esses serviços em resposta às solicitações. O cliente, que neste caso é um navegador web, oferece a interface de usuário e gera solicitações, enquanto o servidor, que executa processamento mais intensivo, aguarda essas solicitações, processa-as e retorna as respostas correspondentes. Essa abordagem é essencial em redes e sistemas distribuídos, proporcionando eficiência na comunicação e compartilhamento de recursos em ambientes como a World Wide Web.

## Tipos de Documentos Web

Os documentos na WWW podem ser agrupados em três grandes categorias: estáticos, dinâmicos ou ativos.

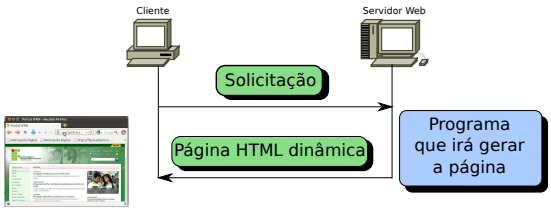
### Estáticos

Páginas estáticas são arquivos HTML simples armazenados em um servidor web. O conteúdo é fixo e não muda com base em interações do usuário ou em variáveis específicas. São utilizados em sites simples de uma única página ou sites institucionais com informações que não mudam frequentemente.




### Dinâmico

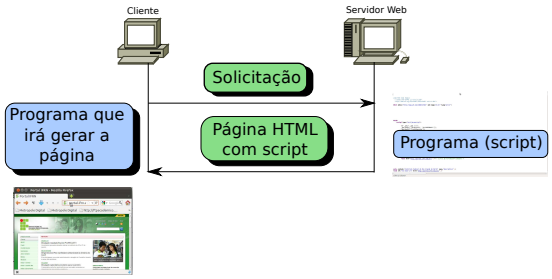
Páginas dinâmicas são geradas no momento em que são solicitadas pelo navegador. O conteúdo pode variar com base em parâmetros, dados de usuário ou outras condições. Quando chega uma solicitação, o servidor Web roda um programa aplicativo ou um script que cria o documento dinamicamente. Podem utilizar as tecnologias: PHP, Python (Django), Ruby (Ruby on Rails), entre outras. Estas páginas são utilizadas em redes sociais, blogs, webmail, Internet Banking, e-commerces, onde o conteúdo pode ser personalizado para usuários específicos ou alterado dinamicamente com base em eventos ou dados em tempo real. Podem utilizar banco de dados para gerar o conteúdo dinâmico.




### Ativo

As páginas ativas oferecem interatividade em tempo real, usando tecnologias (flash, applets Java e JavaScript) do lado do cliente. Suponha, por exemplo, que necessitamos rodar um programa de jogo ou um programa que interage com o usuário. O programa deve ser executado no cliente em que ocorre a animação ou a interação.

Quando um browser solicita um documento ativo, o servidor transmite uma cópia do programa ou de um script. Em seguida, o programa é executado no cliente (browser).



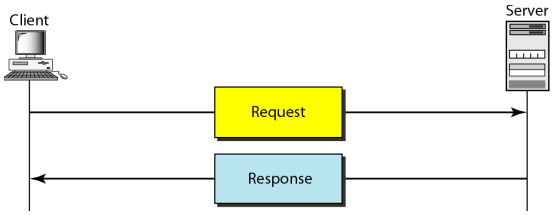

## Protocolo HTTP

O principal protocolo de transferência utilizado em toda a *World Wide Web* é o HTTP (*HyperText Transfer Protocol*). Ele especifica as mensagens que os clientes podem enviar aos servidores e que respostas eles receberão. Tanto as requisições quanto as respostas consistem em cabeçalhos (*headers*) e, opcionalmente, um corpo de mensagem.

O modo habitual de um navegador (*browser*) entrar em contato com um servidor HTTP é estabelecer uma conexão TCP para a porta 80 da máquina servidora.

A idéia do HTTP é muito simples. Um cliente envia um pedido, na forma de uma mensagem, ao servidor. O servidor envia uma resposta, também na forma de mensagem, ao cliente.

O cliente inicia a transação enviando uma mensagem (por exemplo, GET, POST, PUT, HEAD, DELETE) de solicitação (um pedido). Em seguida, o servidor responde enviando a mensagem resposta.




Um método HTTP é uma instrução que define a ação a ser realizada em um recurso identificado em uma solicitação. Em outras palavras, métodos HTTP indicam o tipo de operação que deve ser executado no recurso especificado. Os métodos mais comuns são:

- GET: solicita dados do servidor. Usado visualizar uma página da web.
- POST: envia dados ao servidor para processamento. É comumente usado para enviar formulários.
- PUT: substitui completamente o recurso identificado ou cria um novo recurso se não existir. Geralmente usado para atualizar informações.
- DELETE: remove o recurso identificado no URI. Usado para excluir informações no servidor.

As respostas do servidor web inclui informações sobre o resultado da solicitação e, quando apropriado, o conteúdo solicitado. As respostas seguem um formato específico e contêm diversos elementos essenciais, os principais são:

- Linha de Status: indica o resultado da solicitação em termos de sucesso, erro ou outro status. Por exemplo, "200 OK" indica sucesso, enquanto "404 Not Found" indica que o recurso solicitado não foi encontrado.
- Cabeçalhos (Headers): fornece informações adicionais sobre a resposta, como o tipo de conteúdo, data de modificação, tamanho do conteúdo, e outros detalhes importantes.
- Corpo da Resposta: contém o conteúdo real da resposta, que pode ser HTML, texto, imagens, ou qualquer outro tipo de dados dependendo da natureza da solicitação.

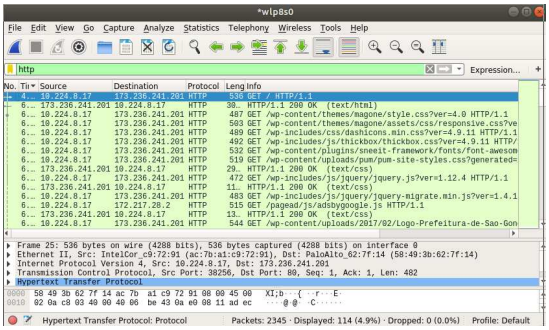
## Prática com Wireshark

Agora iremos realizar uma análise prática de troca de mensagens HTTP entre o navegador e um servidor web utilizando a ferramenta Wireshark. Para isso, siga os seguintes passos:

1. Abra o Wireshark e selecione a interface de rede pela qual a comunicação ocorrerá (por exemplo, Ethernet) e inicie a captura de pacotes;
2. Acesse o site `http://example.com/`.
3. Volte para o Wireshark e clique no ícone "Stop Capture" para encerrar a captura. No campo de filtro (parte superior do Wireshark), digite http e pressione "Enter". Isso filtrará apenas os pacotes relacionados ao protocolo HTTP.
4. Observe a lista de pacotes filtrados e encontre as mensagens HTTP relacionadas à comunicação com o site `http://example.com/`. Utilize o filtro: dns.qry.name == "www.example.com" || http || ip.addr == 2.22.53.149 || ip.addr == 2.22.53.143.
5. No painel de detalhes, expanda a seção "Hypertext Transfer Protocol" para visualizar os cabeçalhos HTTP e o corpo da mensagem. Observe as informações trocadas entre o navegador e o servidor.
6. Utilize a opção "Follow" no menu "HTTP Stream" para visualizar o fluxo completo de dados entre o cliente e o servidor para uma mensagem específica.

Neste exemplo as mensagens HTTP capturadas foram: *GET* (solicita ao servidor que envie a página) e a mensagem de resposta do servidor para seu navegador (200 OK). De acordo com estas duas mensagens responda as seguintes perguntas:

Na sequência será apresentado uma tela com os pacotes capturados na prática.




## Atividade

Formem duplas e responda os seguintes questionamentos abaixo:

1. O que significa a sigla HTTP? Explique brevemente a finalidade principal do Protocolo de Transferência de Hipertexto.
2. Selecione um pacote de solicitação (GET HTTP) na captura. Descreva os principais campos desse pacote, tais como: host, user-agent, accept-Language.
3. Analise uma resposta do servidor. Identifique o código de status HTTP e explique o que esse código indica sobre o resultado da solicitação.
4. Ainda com a mensagem de resposta, cite os principais campos, tais como: Date, content-type, Server e Last modified.
5. O protocolo HTTP utiliza qual protocolo de Transporte?
6. Escrevam as respostas dos quesitos anteriores num documento de texto e insira no Google Sala de Aula.