



Linguagem de Programação para Internet

LUIZ CARLOS FELIX CARVALHO

Regras do Jogo

Primeiro momento (1 a 12/4)

- UNIUBE+ – 5 pontos
- Trabalho – 5 pontos
- Avaliação – 20 pontos

Segundo momento (20 a 29/5)

- UNIUBE+ – 5 pontos
- Trabalho – 5 pontos
- Avaliação – 20 pontos

Terceiro momento (19 a 25/6)

- UNIUBE+ – 5 pontos
- Trabalho – 5 pontos
- Avaliação – 20 pontos
- Simulado – 10 pontos
 - 18/6
- Segunda Chamada / Substitutiva / Recuperação
 - 24 a 28/6

Regras do Jogo

Primeiro momento

- **Trabalho – 5 pontos: 12/4**
- 2 Trabalhos avaliativos
- ✓ ◦ 10 pontos: 22/3
- **10 pontos: 12/4**

Segundo momento

- Trabalho – 10 pontos
- Avaliação – 10 pontos
- 1 Trabalho Avaliativo – 5 pontos

Terceiro momento

- Projeto – 20 pontos
- 1 Trabalho Avaliativo - 5 pontos

Trabalho 1

História da WEB

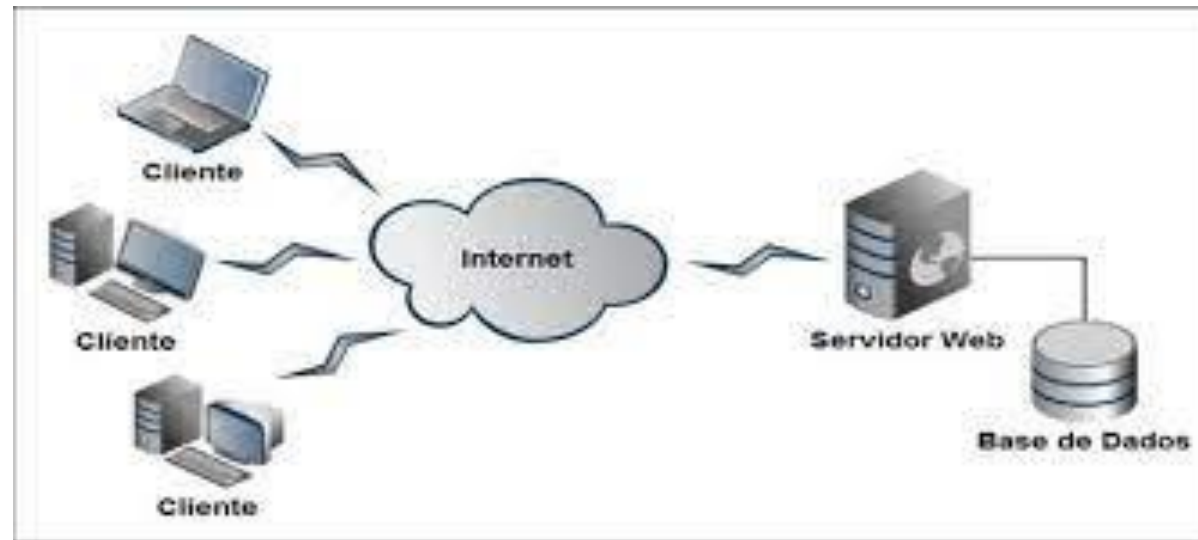
- História do protocolo TCP/IP
- História do protocolo HTTP/WWW e a Linguagem de marcação HTML
- Versões do HTML
- Gerações da WEB
 - Geração 1.0
 - Geração 2.0
 - Geração 3.0
- Mesmo grupo do trabalho já realizado em sala
- **Obs.: obrigatório apresentar bibliografia especificando as páginas. Se não colocar, não será avaliado.**

Tecnologias Web

INTRODUÇÃO

Tecnologias Web

Lado cliente x Lado Servidor



Tecnologias Web

Lado cliente x Lado Servidor

- São termos de desenvolvimento da web que descrevem onde o código do aplicativo é executado.
- Front-end X Back-end

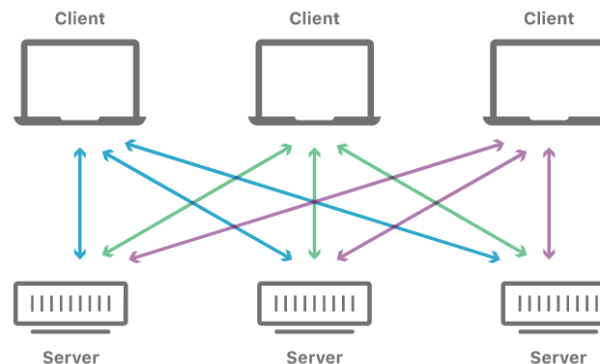
Arquitetura Cliente – Servidor

- Dispositivos do usuário se comunicam por meio de uma rede com servidores localizados centralmente.
- Cliente: notebooks, smartphones, computadores desktop etc.
- Os dispositivos clientes enviam solicitações aos servidores e os servidores fornecem respostas.

Tecnologias Web

Arquitetura Cliente – Servidor

- Servidores são normalmente mais poderosos e confiáveis do que os dispositivos dos usuários.
- São guardados e mantidos em ambientes controlados para garantir que estejam sempre ligados e disponíveis
- Servidores podem ficar inativos, mas, geralmente, há outros servidores servindo de apoio.
- Os dispositivos clientes podem ser desconectados, desligados...
- Servidores podem servir a vários dispositivos clientes ao mesmo tempo.
- Cada dispositivo cliente envia solicitações a vários servidores durante a navegação na internet.



Tecnologias Web

Lado Cliente

- Tudo em um aplicativo web que é exibido ou ocorre no cliente (notebook, smartphone etc.).
- Textos, imagens e o resto da interface.
- Ações que um aplicativo execute no **navegador** do usuário.
- Linguagens (de marcação) como HTML e CSS são interpretadas pelo navegador no lado do cliente.
- Migração de lógica para o lado cliente.
- Os processos do lado do cliente quase sempre são escritos em JavaScript.
- Front-end
 - O lado do cliente se refere apenas ao local onde os processos são executados;
 - Front-end se refere aos tipos de processos que são executados no lado do cliente.

Tecnologias Web

Lado Servidor

- Tudo o que acontece no servidor
- No passado, quase toda a lógica de negócios era executada no lado do servidor
 - Inclusive renderizar páginas da web dinâmicas
- Problema de centralizar toda lógica no servidor:
 - Cada solicitação tem que viajar desde o cliente até o servidor, todas as vezes.
 - Isso apresenta uma grande quantidade de latência.
- Hoje, aplicativos executam mais códigos no lado do cliente.
- Exemplo: renderizar páginas web dinâmicas em tempo real, executando scripts no navegador que fazem alterações no conteúdo que o usuário vê.
- Back-end

Tecnologias Web

Programação do Lado Cliente x Programação do Lado Servidor

- Script do lado do cliente significa simplesmente executar scripts, como JavaScript, no dispositivo do cliente, geralmente em um navegador.
- Os scripts do lado do servidor são executados no servidor, geralmente para entregar conteúdo dinâmico a páginas web em resposta às ações dos usuários.
- Várias linguagens:
 - Java
 - PHP
 - NodeJS
 - Python
 - C#





Tecnologias Web

Requisições HTTP e HTTPS

Tecnologias Web

HTTP

- HTTP (Hypertext Transfer Protocol)
- Protocolo projetado para a transferência de conteúdo na Web.
- Protocolo simples que utiliza os serviços de protocolo TCP para executar sua função de transferência de conteúdo.
- Um dos protocolos de aplicativo mais usados.
- Todas as operações na Web utilizam o protocolo HTTP.

HTTPS

- Versão segura do protocolo HTTP
- Implementa o HTTP usando TLS (segurança de camada de transporte)
- Configuração adicional necessária para configurar o TLS
- HTTPS é basicamente idêntico ao HTTP em uso

Tecnologias Web

HTTP

- O HTTP é um protocolo de camada de aplicação no modelo de comunicação de rede Open Systems Interconnection (OSI)
- HTTP transmite dados não criptografados
- Define vários tipos de solicitações e respostas
- Requisições: GET, PUT, POST etc.
- Respostas:
 - 200 – OK
 - 400 – Solicitação inválida
 - 404 – Recurso não encontrado
- Geralmente, requisições e respostas são invisíveis para os usuários

Tecnologias Web

HTTPS

- Possui camada de segurança à comunicação
- Combina solicitações e respostas HTTP com a tecnologia SSL e TLS
 - SSL (Secure Sockets Layer): precursor do TLS, depreciado em favor do TLS
 - TLS (Transport Layer Security):
 - Ambos os protocolos usam criptografia para proteger a transmissão de dados
 - TLS passou por desenvolvimento e melhorias mais rigorosos
- Para utilizar HTTPS:
 - Obter um certificado SSL/TLS de uma autoridade de certificação (CA) independente
 - O certificado é compartilhado com o navegador antes de trocar dados para estabelecer a segurança
 - No momento de criptografar e descriptografar o certificado é utilizado.

Tecnologias Web

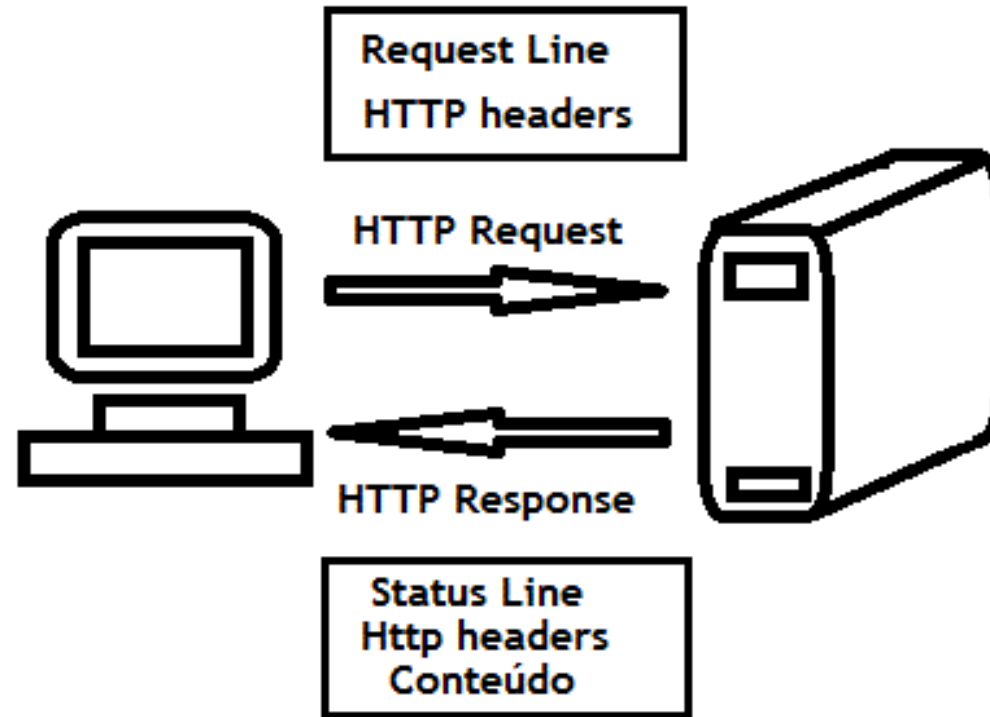
HTTPS - Funcionamento

- Acessar site HTTPS digitando `https://`, indicando tal protocolo
- O navegador tenta verificar a autenticidade do site solicitando o certificado SSL do servidor.
- O servidor envia o certificado SSL que contém uma chave pública como resposta.
- O certificado SSL do site comprova a identidade do servidor. Quando o navegador estiver satisfeito, ele usará a chave pública para criptografar e enviar uma mensagem que contém uma chave de sessão secreta.
- O servidor web usa sua chave privada para descriptografar a mensagem e recuperar a chave de sessão. Em seguida, ele criptografa a chave da sessão e envia uma mensagem de confirmação ao navegador.
- Agora, o navegador e o servidor da Web mudam para usar a mesma chave de sessão para trocar mensagens com segurança.

Tecnologias Web

HTTPS vs HTTP

- Segurança:
 - HTTP são de texto simples
 - HTTPS transmite todos os dados em formato criptografado
- Autoridade:
 - Mecanismos de pesquisa geralmente classificam o conteúdo do site HTTP abaixo das páginas da Web HTTPS
 - O navegador torna a conexão HTTPS visível para seus usuários (cadeado na barra de endereço)



Tecnologias Web

Estrutura de uma requisição HTTP

- Dois tipos de mensagens:
 - Requisição HTTP (HTTP request)
 - Resposta HTTP (HTTP response)
- São compostas de informação textual codificada em ASCII.
- Possuem por múltiplas linhas.
- HTTP/1.1: mensagens eram abertamente enviadas através da conexão.
- HTTP/2: mensagens divididas em quadros HTTP, resultando em otimização e melhora de desempenho.
- Obs.: mensagens HTTP são transparentes para os desenvolvedores.

Tecnologias Web

Estrutura de uma requisição HTTP

Requisições e respostas HTTP:

- Linha inicial (start-line): requisições a serem implementadas, ou seu status de sucesso ou falha.
- Um conjunto opcional de cabeçalhos HTTP especificando a requisição, ou descrevendo o corpo incluso na mensagem.
- Uma linha em branco (empty line) indicando que toda meta-informação para a requisição já foi enviada.
- Um corpo (body) contendo dados associados à requisição (como o conteúdo de um formulário HTML), ou o documento associado à resposta.

Tecnologias Web

Estrutura de uma requisição HTTP

Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,..., */*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345
```

```
-12656974
(more data)
```

Responses

```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

start-
line

HTTP
headers

empty
line

body

Tecnologias Web

Estrutura de uma requisição HTTP – Linha inicial

- Contêm três elementos:

- Um método HTTP

- Alvo da requisição: uma URL

- Um caminho absoluto, seguido de um '?' e o texto da consulta.

- URL completa

- Versão HTTP

Tecnologias Web

Estrutura de uma requisição HTTP – Cabeçalho

- Um cabeçalho:

- <chave>:<valor>
- Chave: palavra *case insensitive* (não diferencia maiúsculo e minúsculo)
- Valor: um valor, cujo a estrutura depende da chave
- Ocupa uma única linha
- Pode possuir muitos caracteres

- Vários cabeçalhos disponíveis:

- Cabeçalhos gerais: se aplicam à mensagem como um todo.
- Cabeçalhos de requisição: (Accept-Type) modificam a requisição, especificando-a mais (como Accept-Language), dando-a contexto (como Referer), ou restringindo-a condicionalmente (como If-None).
- Cabeçalhos de entidade, como Content-Length que se aplicam ao corpo da mensagem.

Tecnologias Web

Estrutura de uma requisição HTTP – Cabeçalho

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,...,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

The diagram illustrates the structure of an HTTP request header. It features a red background for the first four lines (Request headers), a green background for the next two lines (General headers), and a blue background for the last two lines (Entity headers). Arrows point from the labels on the right to the corresponding sections of the header.

- Request headers
- General headers
- Entity headers

Tecnologias Web

Estrutura de uma requisição HTTP – Corpo

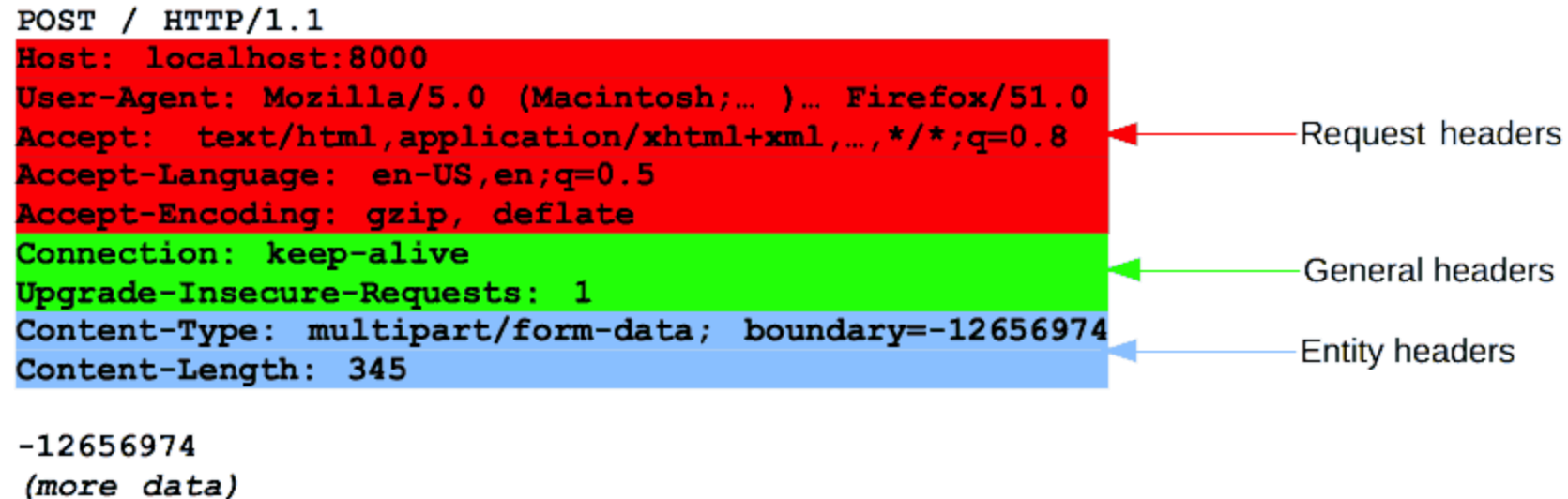
- Parte final da requisição.
- Nem todas as requisições tem um.
 - Não tem: GET, HEAD, DELETE, OPTIONS
- São dados a serem enviados para o servidor.

Tecnologias Web

Estrutura de uma requisição HTTP – Corpo

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,...,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```



The diagram illustrates the structure of an HTTP request. The request is divided into three main sections, each highlighted with a different color and labeled with an arrow:

- Request headers:** Indicated by a red arrow pointing to the red-highlighted section containing the following headers:
 - Host: localhost:8000
 - User-Agent: Mozilla/5.0 (Macintosh;...)... Firefox/51.0
 - Accept: text/html,application/xhtml+xml,...,*/*;q=0.8
 - Accept-Language: en-US,en;q=0.5
 - Accept-Encoding: gzip, deflate
- General headers:** Indicated by a green arrow pointing to the green-highlighted section containing:
 - Connection: keep-alive
 - Upgrade-Insecure-Requests: 1
- Entity headers:** Indicated by a blue arrow pointing to the blue-highlighted section containing:
 - Content-Type: multipart/form-data; boundary=-12656974
 - Content-Length: 345

The body of the request is shown below the headers, starting with the boundary string `-12656974` and followed by `(more data)`.

Tecnologias Web

Estrutura de uma requisição HTTP – Métodos HTTP

- **GET:** solicita a representação de um recurso específico.
- **POST:** utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.
- **PUT:** substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.
- **PATCH:** utilizado para aplicar modificações parciais em um recurso.
- **DELETE:** remove um recurso específico.
- **Outros: OPTIONS, HEAD, TRACE, CONNECT**

Tecnologias Web

Estrutura de uma requisição HTTP – Métodos HTTP

- **GET:** pesquisa.
- **POST:** inserção.
- **PUT:** atualização total.
- **PATCH:** atualização parcial.
- **DELETE:** remoção.

Tecnologias Web

Respostas HTTP – Linha de status

- Linha inicial: linha de status
- Três informações:
 - A versão do protocolo: HTTP/1.1 e HTTP/2.
 - Um código de status, indicando o sucesso ou falha da requisição. Códigos de status comuns são 200, 404, ou 302.
 - Um texto de status: uma descrição textual breve, puramente informativa.
- Exemplo: HTTP/1.1 404 Not Found.

Tecnologias Web

Respostas HTTP – Cabeçalhos

- Mesma estrutura do cabeçalho de requisição
- Vários cabeçalhos disponíveis:
 - Cabeçalhos gerais: aplicam-se à toda mensagem.
 - Cabeçalhos de resposta: dão informação adicional sobre o servidor.
 - Cabeçalhos de entidade: aplicam-se ao corpo da resposta (ex.: Content-Length).

Tecnologias Web

Respostas HTTP – Cabeçalhos

HTTP/1.1 200 OK

Access-Control-Allow-Origin: *

Connection: Keep-Alive

Content-Encoding: gzip

Content-Type: text/html; charset=utf-8

Date: Wed, 10 Aug 2016 13:17:18 GMT

Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"

Keep-Alive: timeout=5, max=999

Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT

Server: Apache

Set-Cookie: csrftoken=.....

Transfer-Encoding: chunked

Vary: Cookie, Accept-Encoding

X-Frame-Options: DENY

(body)

Response headers

Entity headers

General headers

Tecnologias Web

Respostas HTTP – Corpo

- Última parte de uma resposta.
- Nem toda resposta tem um.
 - Ex.: Código de status 201 ou 204 normalmente não tem.

Tecnologias Web

Respostas HTTP – Códigos de Respostas

63 códigos divididos em:

- Respostas Informativas (100 – 199)
 - Ex.: 102 Processing
- Respostas bem-sucedidas (200 – 299)
 - Ex.: 200 OK
- Mensagens de redirecionamento (300 – 399)
 - 304 Not Modified
- Respostas de erro do cliente (400 – 499)
 - 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
- Respostas de erro do servidor (500 – 599)
 - 500 Internal Server Error