

Programação para WEB



Airton Bordin Junior
airtonbjunior@gmail.com



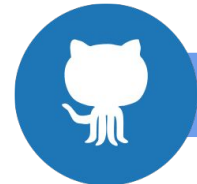
Quem sou eu?

Airton Bordin Junior

- Graduação
 - Ciência da Computação
 - Gestão Pública
- Especialização
 - Redes de Computadores
 - MBA Gerenciamento de Projetos
- Mestrado em Ciência da Computação - Inteligência Artificial



airtonbjunior@gmail.com
airton.junior@uniceub.br



github.com/airtonbjunior



linkedin.com/in/airtonbjunior

- Unidade 1: Conceitos iniciais
- Unidade 2: Programação WEB
- Unidade 3: Conceitos de Framework
- Unidade 4: Prototipação de Sistemas WEB
- Unidade 5: Conhecimento de Back-End

MEDEIROS, Ernani. Desenvolvendo Software com Uml Definitivo 2.0. Makron Books

PRESSMAN, Roger S.; Lowe. Engenharia Web. LTC

SILVA, Luciano Carlos. Banco de Dados para Web: Do Planejamento À Implementação. Érica

ELS, Diego. Guia do Front-End: O caminho das pedras para ser um programador Front-End. São Paulo: Casa do Código, 2015.

KRUG, Steve. Não me faça pensar. Rio de Janeiro: Alta Books, 2014.

LUBBERS, Peter. ALBERS, Brian. SALIM, Frank. Pro HTML5 Programming. New York: Springer, 2010.

MAZZA, Lucas. HTML5 e CSS3 Domine a web do futuro. São Paulo: Casa do código, 2012.

SAFFER, Dan. Designing For Interaction: Creating Innovative Applications and Devices. 2Ed. Berkeley: Voices That Matter, 2010.

Visual Studio C# - Fundamentos, Programação Com Asp.net, Windows Forms E Web Services - Autor:PETRONI, BENEDITO|Autor:LUHMANN, ANGELA|Autor:OLIVEIRA, CLAUDIO VIEIRA

Dinâmica das aulas

Unidade 1

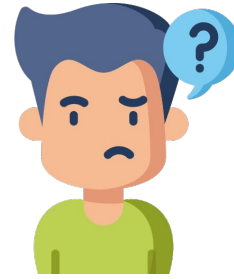
Conceitos iniciais

- Unidade 1: Conceitos iniciais
- Unidade 2: Programação WEB
- Unidade 3: Conceitos de Framework
- Unidade 4: Prototipação de Sistemas WEB
- Unidade 5: Conhecimento de Back-End

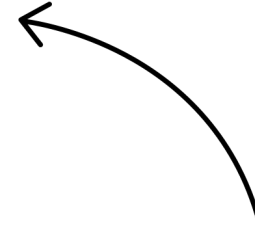
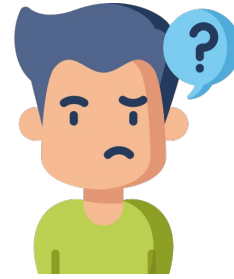
- Principais características do desenvolvimento WEB
- Linguagens e Padrões de Desenvolvimento WEB
- Protocolos de Comunicação

Programação para WEB

Programação para WEB



Programação para **WEB**



Programação para WEB

WEB \neq Internet

- WEB
 - Tim Berners-Lee
 - Resolver um problema de comunicação entre cientistas de diferentes regiões do mundo
 - 1989: projeto WorldWideWeb

Information Management: A Proposal

Tim Berners-Lee, CERN

March 1989, May 1990

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

- Primeiro website - Agosto de **1991** - Tim Berners-Lee

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

- Primeiro website - Agosto de **1991** - Tim Berners-Lee

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

<https://info.cern.ch/hypertext/WWW/TheProject.html>



- Primeiro website - Agosto de **1991** - Tim Berners-Lee



NeXT Computer, usado por Tim Berners-Lee no CERN. Tornou-se o primeiro servidor WEB

- Primeiro website - Agosto de **1991** - Tim Berners-Lee



Organização Europeia para a
Pesquisa Nuclear

NeXT Computer, usado por Tim
Berners-Lee no **CERN**. Tornou-se o
primeiro servidor WEB

- Primeiro website - Agosto de **1991** - Tim Berners-Lee

8MB RAM
256MB Disco



Organização Europeia para a
Pesquisa Nuclear

NeXT Computer, usado por Tim
Berners-Lee no **CERN**. Tornou-se o
primeiro servidor WEB

- Características do desenvolvimento WEB
 - Acessibilidade Global
 - Interatividade
 - Compatibilidade com Dispositivos
 - Padrões Abertos e Conformidade
 - Facilidade de Atualização e Manutenção

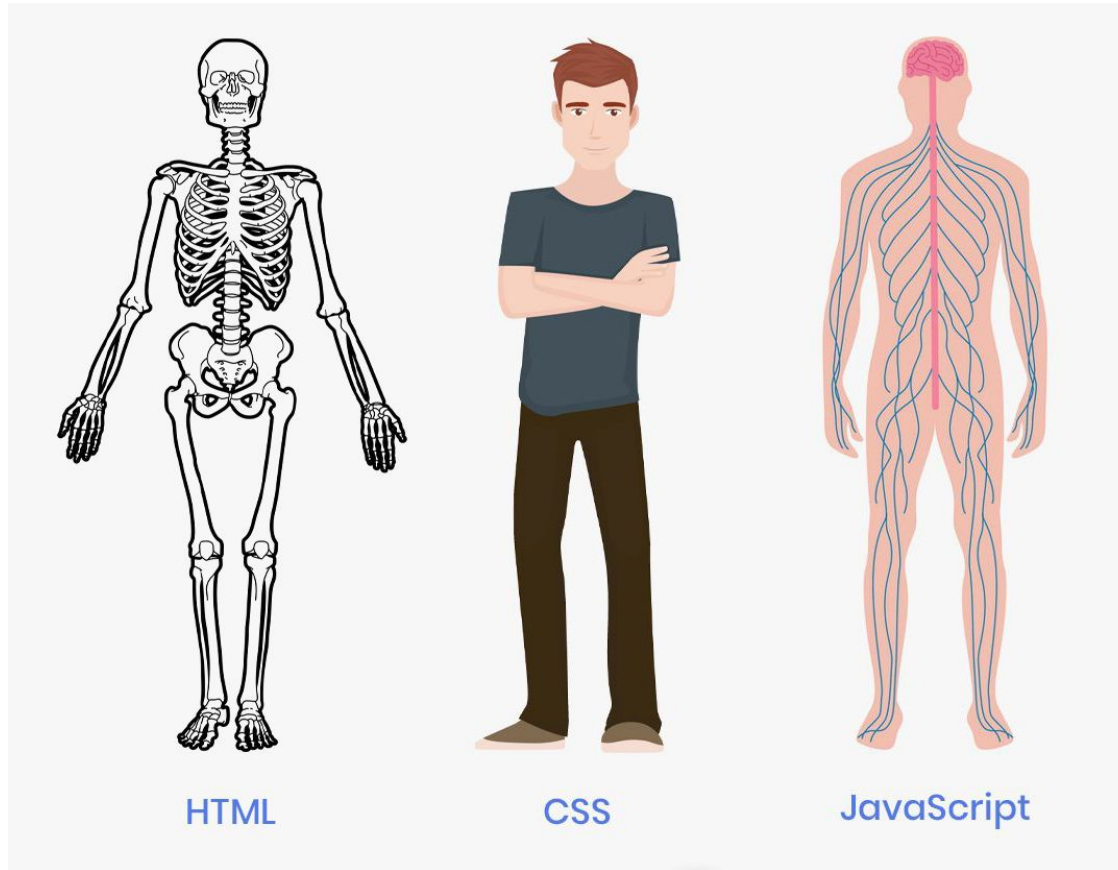
- Principais características do desenvolvimento WEB
- Linguagens e Padrões de Desenvolvimento WEB
- Protocolos de Comunicação

- Linguagens
 - HTML (é linguagem de programação?)
 - CSS
 - Javascript
 - ???

- HTML
 - **Linguagem de marcação** padrão para criar a **estrutura e o conteúdo** das páginas web
 - Elementos e tags

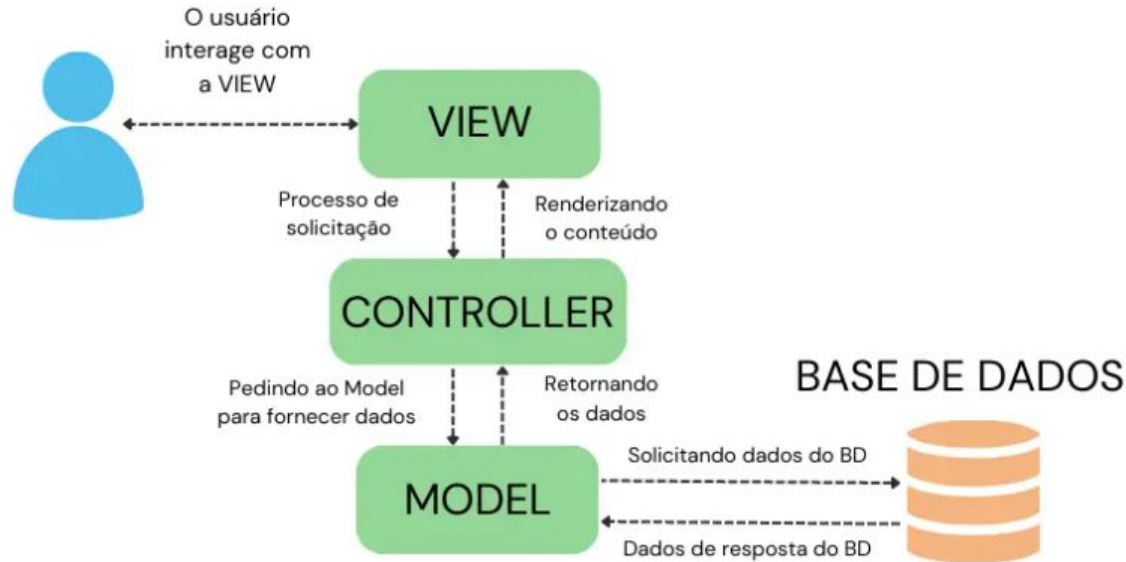
- CSS
 - CSS é uma **linguagem de estilo** usada para controlar a apresentação e o layout dos elementos HTML em páginas WEB

- Javascript
 - **Linguagem de programação** de alto nível usada para adicionar interatividade e dinamismo a páginas WEB



- Padrões de desenvolvimento WEB
 - Model-View-Controller (MVC)
 - Single Page Application (SPA)
 - Progressive Web Apps (PWAs)
 - Microsserviços
 - Representational State Transfer (REST)

Model-View-Controller (MVC): Separa a aplicação em 3 componentes principais: Modelo (representa os dados e a lógica de negócios), Visão (trata da interface do usuário) e Controlador (intermediário entre o Modelo e Visão, controlando as interações do usuário e atualizando o Modelo e Visão)



Single Page Application (SPA)

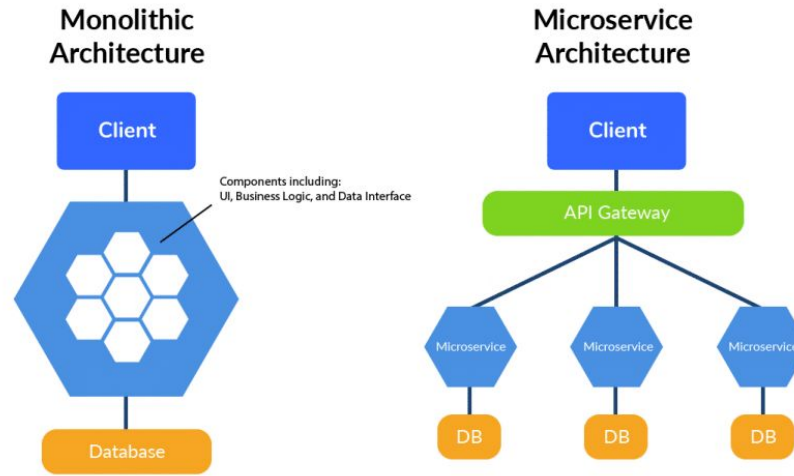
- Toda a aplicação é carregada em uma única página web;
- Conteúdo é dinamicamente atualizado conforme o usuário interage com a aplicação, sem a necessidade de recarregar a página inteira

Progressive Web Apps (PWAs)

- Aplicativos WEB que oferecem funcionalidades e experiências semelhantes às de aplicativos nativos
- Construídos usando tecnologias WEB padrão, como HTML, CSS e JavaScript.
- Chamados de "progressivos" porque podem ser implementados de forma progressiva, ou seja, podem ser acessados por meio de um navegador da web como qualquer outro site, mas também oferecem a opção de serem instalados na tela inicial de dispositivos móveis ou computadores e acessados como um aplicativo nativo

Microserviços

- Arquitetura de desenvolvimento de software que divide uma aplicação em serviços pequenos, independentes e interconectados
- Cada serviço é focado em uma função específica e pode ser desenvolvido, implantado e dimensionado de forma independente



REST (Representational State Transfer)

Estilo arquitetural que define um conjunto de princípios para projetar serviços WEB escaláveis e interoperáveis

Baseados na ideia de recursos, acessados e manipulados usando operações HTTP padrão, como GET, POST, PUT e DELETE

- Principais características do desenvolvimento WEB
- Linguagens e Padrões de Desenvolvimento WEB
- Protocolos de Comunicação

O que é um Protocolo?



O que é um Protocolo?

Conjunto de regras convencionadas que controla e possibilita a conexão, comunicação e, transferência de dados entre dois ou mais sistemas computacionais

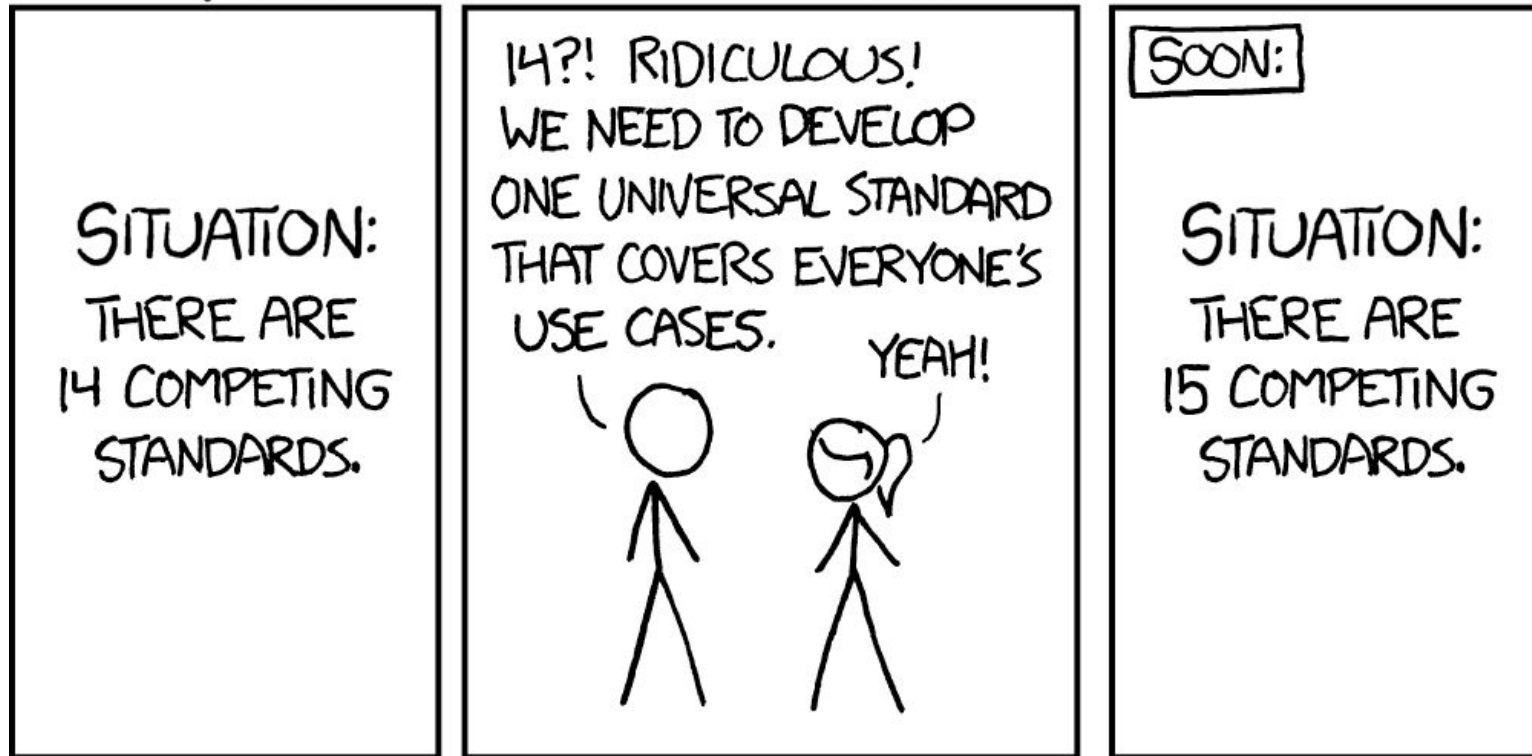
Por que criar um Protocolo?



Por que criar um Protocolo?

Padronização da Comunicação,
interoperabilidade, segurança, eficiência etc

HOW STANDARDS PROLIFERATE: (SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)



- **RFC**

- *Request For Comments*
- Documentos que pesquisadores, engenheiros, cientistas de computação e outras pessoas de áreas correlatas disponibilizam para revisão e possível criação de um padrão.
- Definem os padrões de muitas coisas fundamentais para a internet que conhecemos hoje
 - RFC #2616: HTTP 1.1
 - RFC #7540: HTTP/2
 - RFC #9110: HTTP Semantics

Protocolo HTTP

- **HTTP**

- Opera no modelo **cliente-servidor**
- Cliente, como um navegador da WEB, envia requisições para um servidor WEB e espera por respostas
- Cliente é geralmente o iniciador da comunicação, solicitando recursos como páginas da web, imagens, vídeos, etc

- **Ciclo de vida de uma requisição**
 - **Inicialização da requisição:** Cliente inicia a requisição especificando o **método de requisição**, o **URI** (Uniform Resource Identifier) do recurso desejado e outros parâmetros, como cabeçalhos e dados do corpo

- **Ciclo de vida de uma requisição**

- **Inicialização da requisição:** Cliente inicia a requisição especificando o método de requisição, o URI (Uniform Resource Identifier) do recurso desejado e outros parâmetros, como cabeçalhos e dados do corpo



*O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a **ação** a ser executada para um dado recurso*

Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como Verbos HTTP

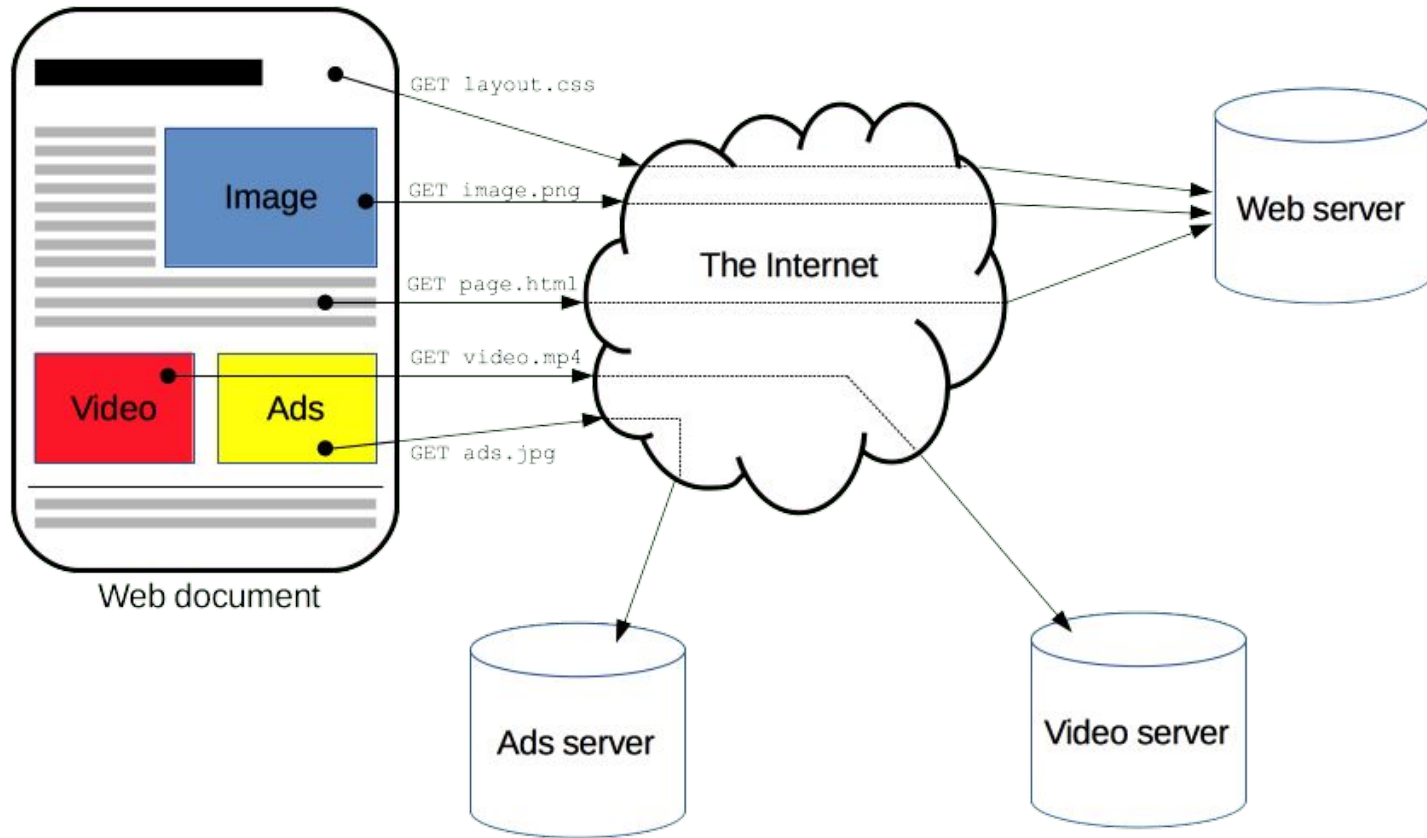
GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH

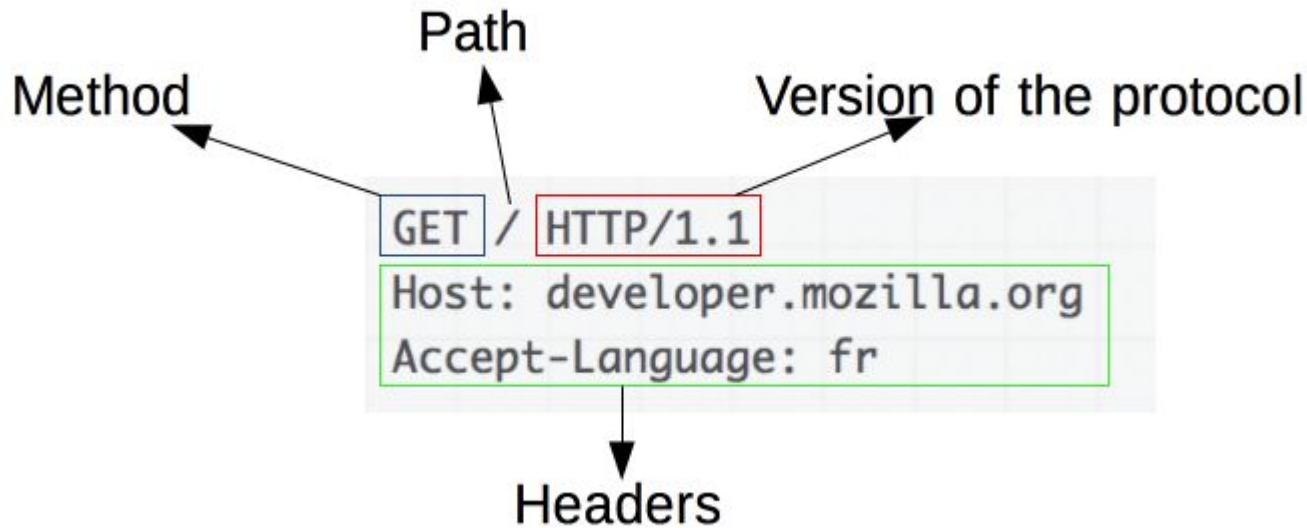
- **Ciclo de vida de uma requisição**
 - **Inicialização da requisição:** Cliente inicia a requisição especificando o **método de requisição**, o **URI (Uniform Resource Identifier)** do recurso desejado e outros parâmetros como cabeçalhos e dados do corpo



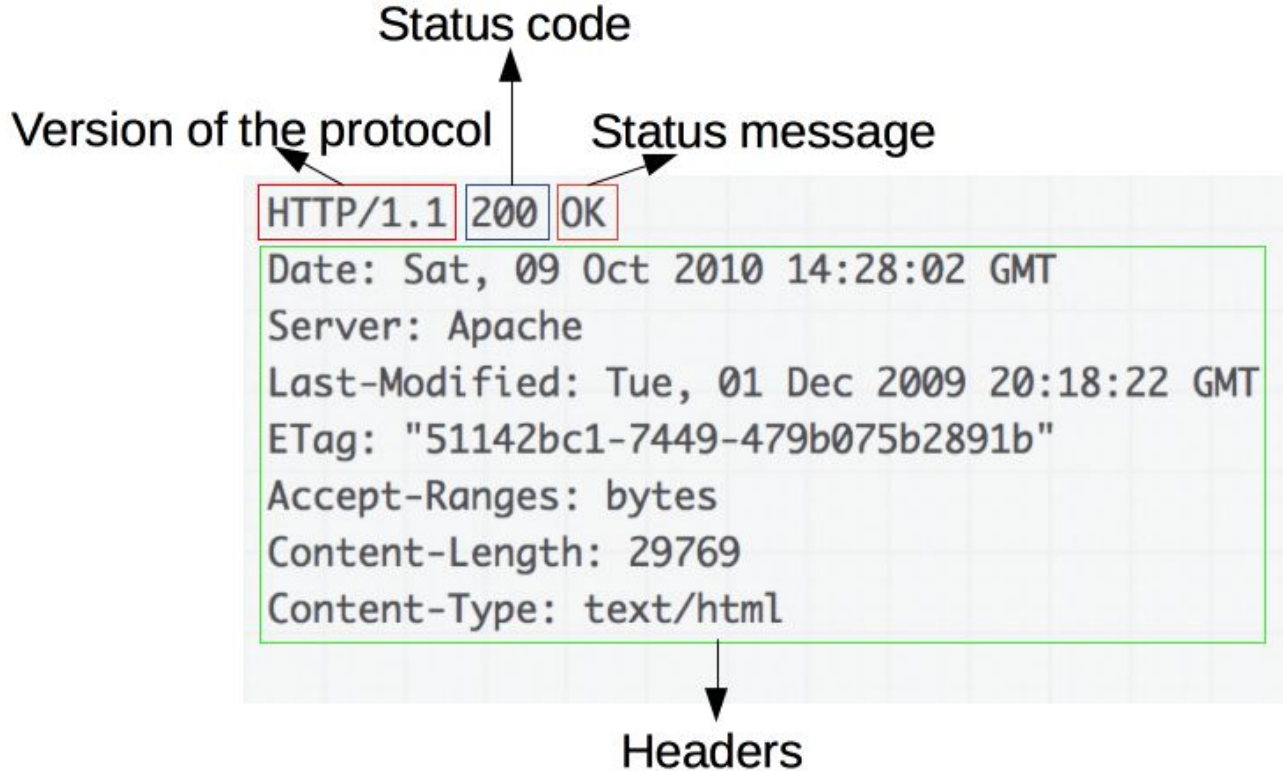
Uma URI pode ser definida como um localizador, enquanto uma URL provê também informações sobre como este recurso pode ser acessado

- O HTTP é um protocolo **sem estado**
- Cada requisição é tratada de forma independente, sem nenhuma memória de requisições anteriores
- Simplifica a implementação de servidores e aumenta a escalabilidade do sistema, mas também requer o uso de mecanismos adicionais, como **cookies e sessões**, para manter o estado entre requisições.





HTTP Request



HTTP Response

- Códigos de status HTTP
 - Indicam se uma solicitação foi concluída com êxito
 - São agrupadas em cinco classes
 - Respostas Informativas (100 – 199)
 - Respostas bem-sucedidas (200 – 299)
 - Mensagens de redirecionamento (300 – 399)
 - Respostas de erro do cliente (400 – 499)
 - Respostas de erro do servidor (500 – 599)

- Códigos de status HTTP



[https://developer.mozilla.org/pt-BR/
docs/Web/HTTP/Status](https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status)