



AceleraDev Loadsmart Women Edition

Módulo 3

Camila Maia

Esta apresentação está disponível em:

<https://github.com/camilamaia/accelera-dev-loadsmart-women/>



Módulos

Python: Noções básicas

Primeiro programa, teste, estrutura de dados, condicionais, repetições, operações, funções, classes, objetos...

1

2

3

4

5

Django I: instalação, iniciando um projeto, arquitetura, modelos, views, django admin, URLs

REST APIs: O protocolo HTTP, Rest APIs e Django REST Framework

Um pouco mais de Python

Exceções, decorators, list e dict comprehensions, map, reduce, filter...

Django II: Templates, HTML e CSS básico, Forms, Autorização e Autenticação.



Loadsmart · GitHub

×

+

←

→

↺

🏠

🔒 GitHub, Inc. (US)

https://github.com/loadsmart

⋮

🔒

☆

☰

📄

🔗

☰

🐙

Why GitHub? ▾

Enterprise

Explore ▾

Marketplace


Pricing ▾

Search

/

Sign in

Sign up



Loadsmart

📍 New York City, NY


🔗 https://loadsmart.com/

Verified

📁 Repositories 12

👤 People 7


📁 Projects 0



Grow your team on GitHub

GitHub is home to over 31 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

Sign up



Pinned repositories

📁 danger-android lint

📁 danger-pen8



Internet

- É um sistema global de **redes de computadores** interligadas que utilizam um conjunto próprio de protocolos (Internet Protocol Suite ou TCP/IP)
- É uma rede que conecta milhões de computadores pelo mundo
- Tem o propósito de servir progressivamente usuários no mundo inteiro.

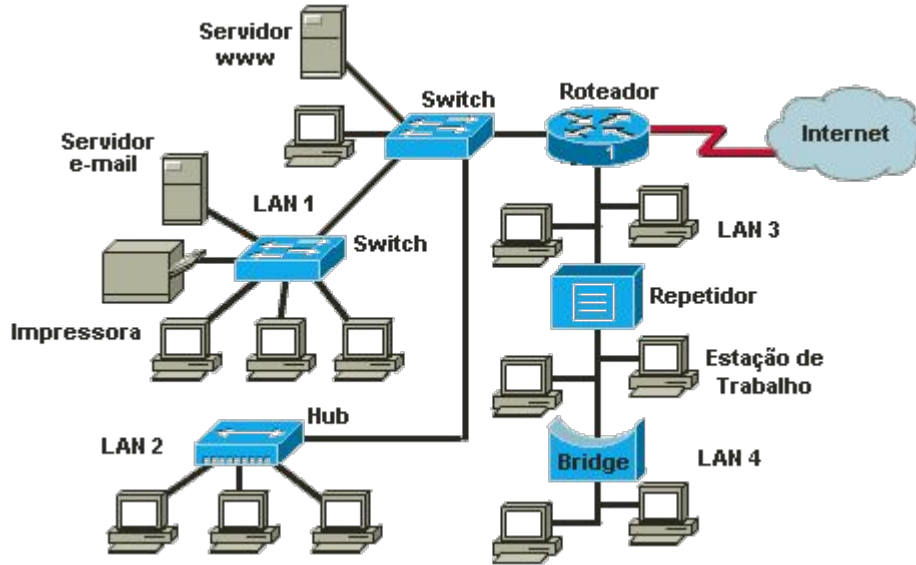


Internet

- É uma rede de **várias outras redes**, que consiste de milhões de empresas privadas, públicas, acadêmicas e de governo, com alcance local e global e que está ligada por uma **ampla variedade de tecnologias** de rede eletrônica, sem fio e ópticas



Internet



Web

- Também conhecida como World Wide Web (WWW)
- É um sistema de **documentos**, os quais esses são identificados por **URLs** (Uniform Resource Locators), que podem ser interligados por **hiperligação** e, são acessíveis via **Internet**.

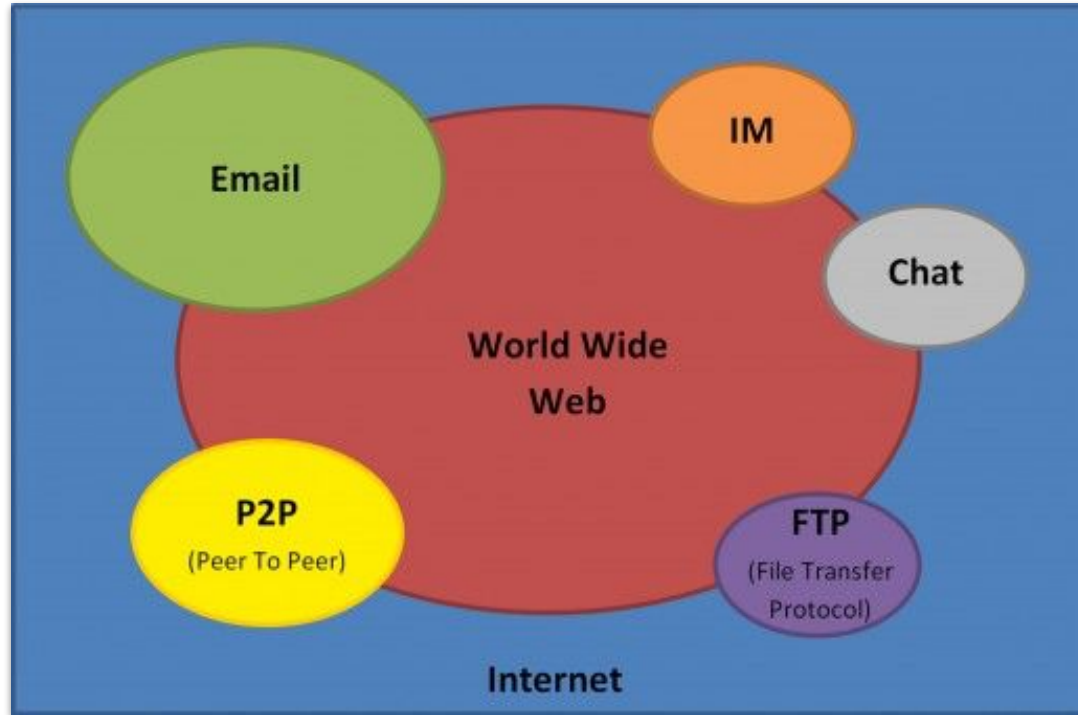


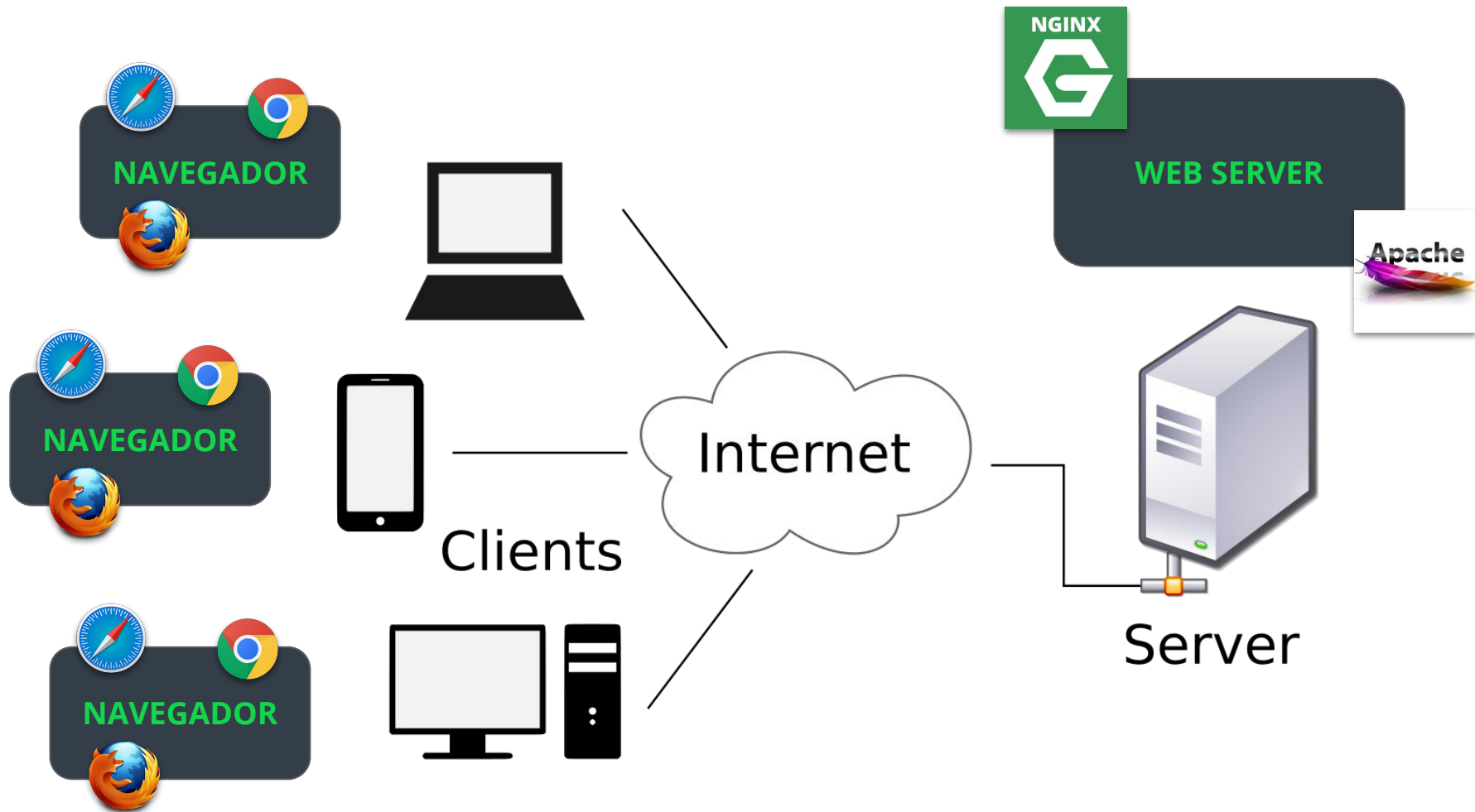
Web

- Os documentos podem estar na forma de vídeos, sons, hipertextos e imagens.
- Para consultar a informação, pode-se usar um programa de computador chamado **navegador** web para descarregar informações (páginas) de servidores web e mostrá-los na tela do usuário.



Internet e Web





Web

PROTOCOLO HTTP



Loadsmart · GitHub

×

+

← → ↺ 🏠

GitHub, Inc. (US) | https://github.com/loadsmart

⋮ 🛡️ ☆

☰ 📄 🗨️


🐙

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search /

Sign in

Sign up




Loadsmart

📍 New York City, NY 🔗 https://loadsmart.com/ Verified

📁 Repositories 12

👤 People 7

📁 Projects 0



Grow your team on GitHub

GitHub is home to over 31 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

Sign up

Pinned repositories

📁 danger-android lint

📁 danger-pen8



Web

DOMÍNIO

<https://github.com/loadsmart>

PROTOCOLO

CAMINHO DO ARQUIVO NO
SERVIDOR



Web Funcionamento

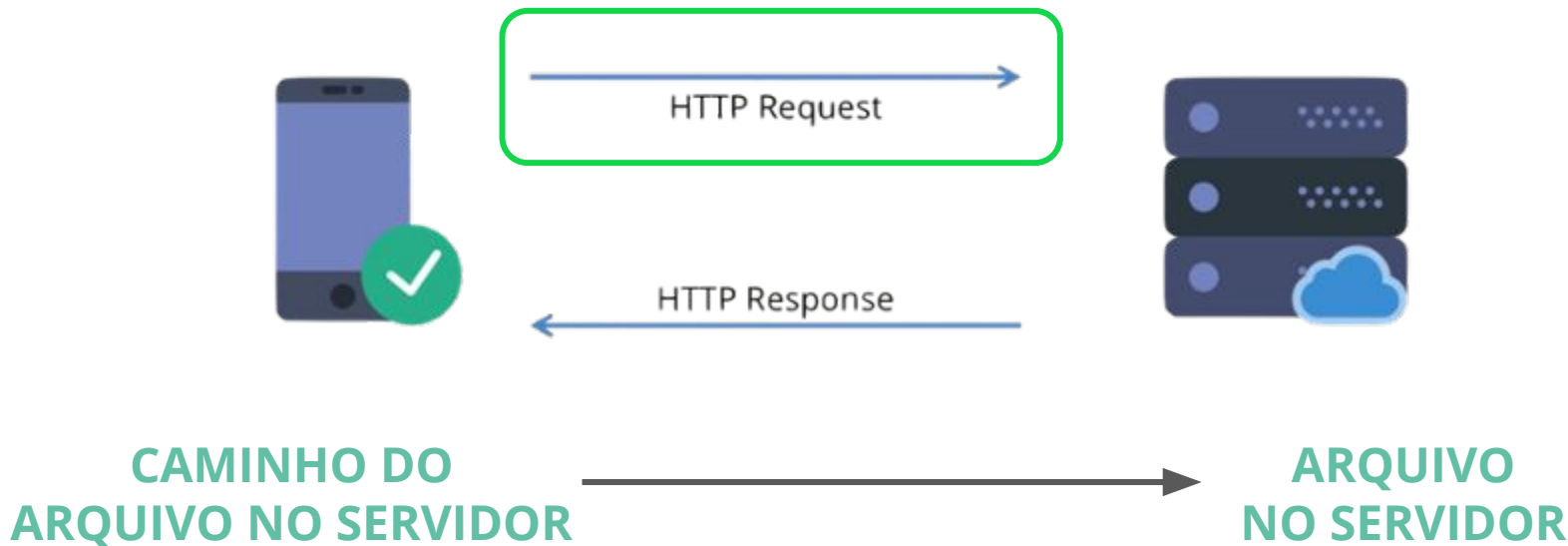
- Digitar uma URL no navegador ou seguindo (acessando) uma hiperligação.
- Primeiramente, a parte da URL referente ao servidor de rede é separada e transformada em um endereço IP, por um banco de dados da Internet chamado domain name system (**DNS**).
- O navegador estabelece, então, uma conexão com o servidor web localizado no endereço IP retornado.

DOMÍNIIO → WEB SERVER



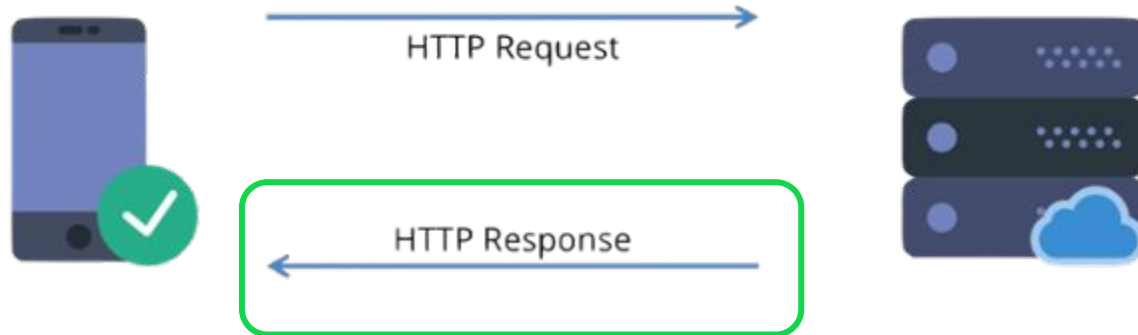
Web Funcionamento

- O próximo passo é o navegador enviar uma **requisição HTTP** ao servidor para obter o recurso indicado pela parte restante da URL



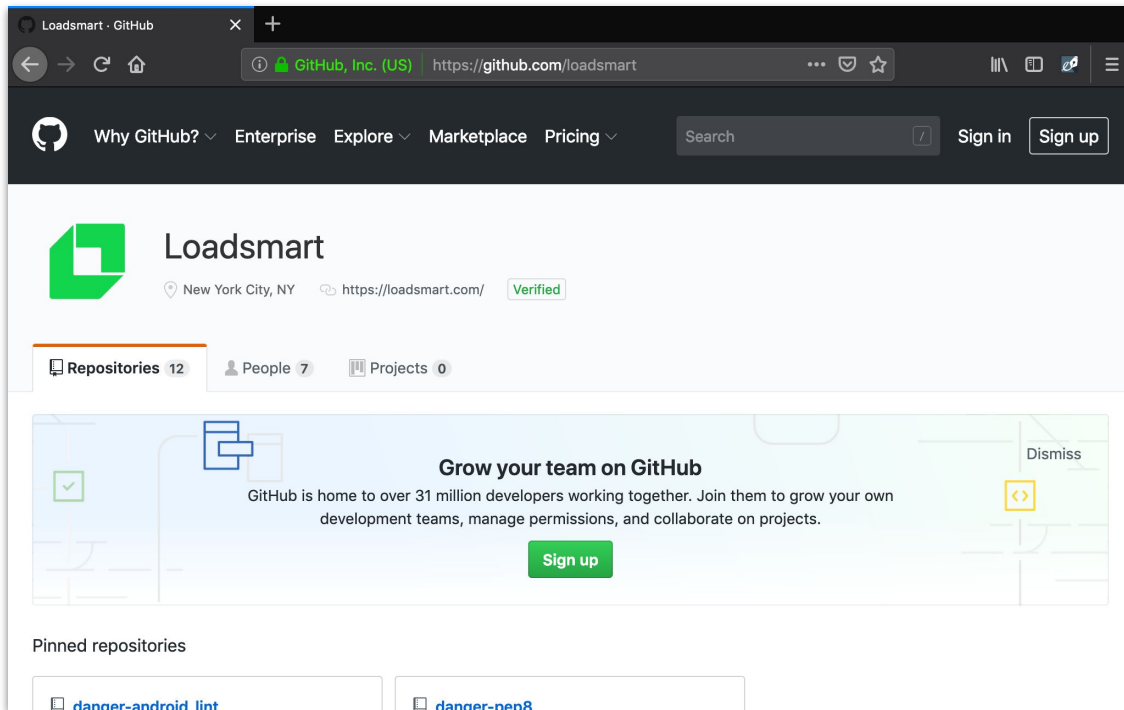
Web Funcionamento

- No caso de uma página web típica, o texto HTML é recebido e interpretado pelo navegador, que realiza então requisições adicionais para figuras, arquivos de formatação, arquivos de script e outros recursos que fazem parte da página.

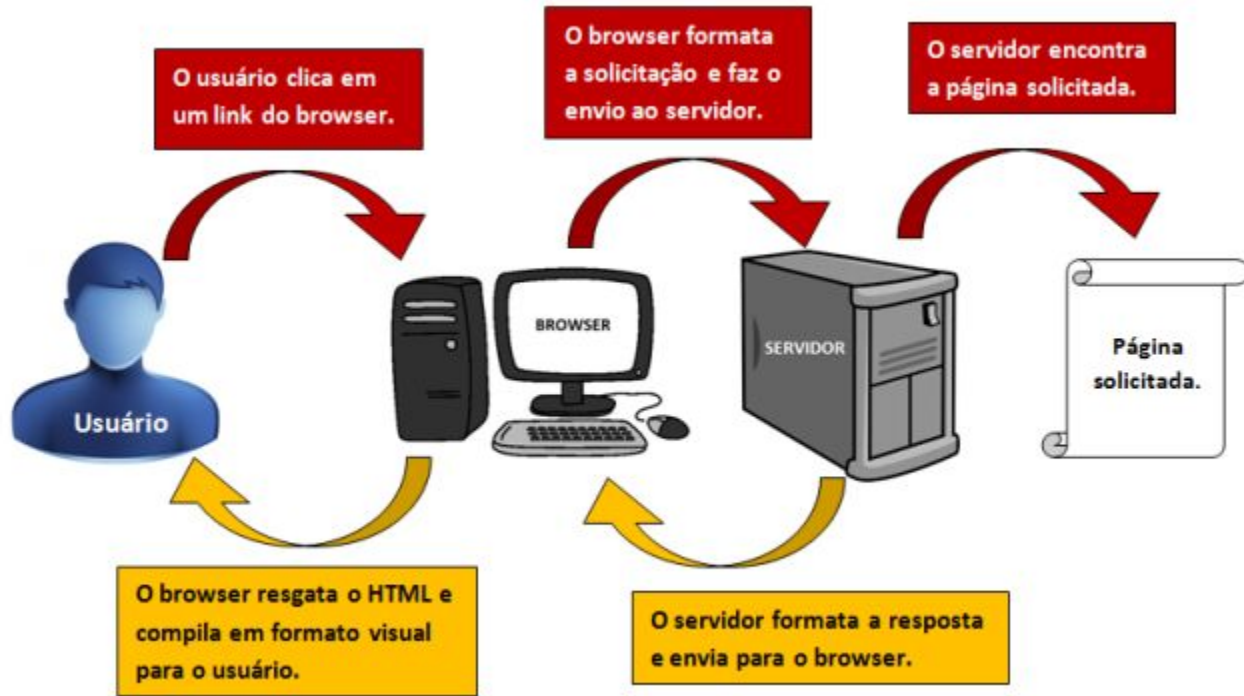


Web Funcionamento

- O navegador, então, renderiza (reconstitui) a página na tela do usuário (utilizador), assim como descrita pelos arquivos que a compõe.

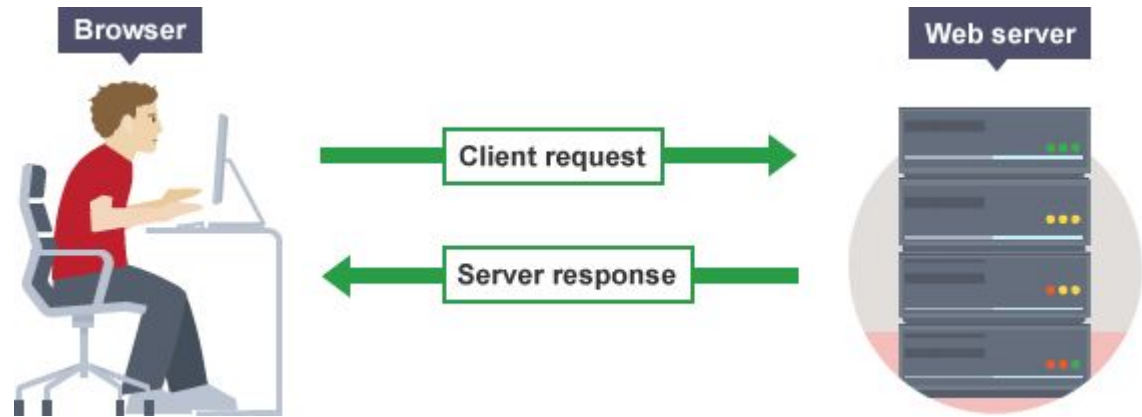


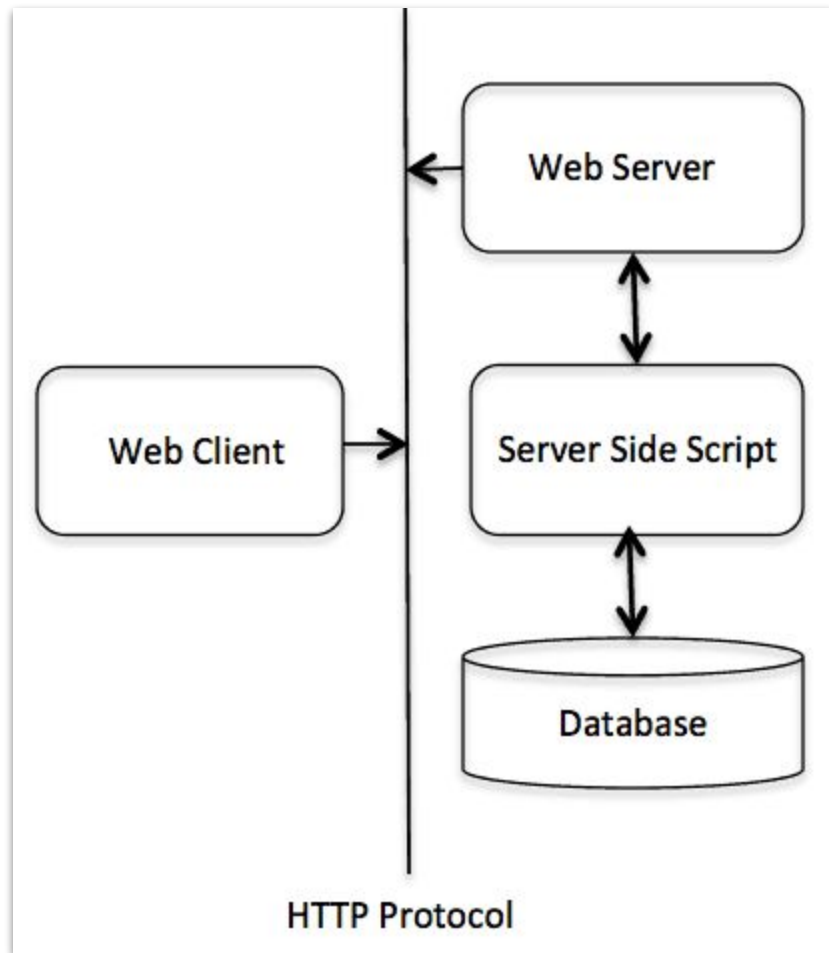
Web Funcionamento



Protocolo HTTP

- Hypertext Transfer Protocol
- Connectionless





Requests

Método

GET, POST, PUT, DELETE

URI: **github.com/loadsmart**

Header

- User-Agent

Boby (Opcional)



Método HTTP

GET

Retrieve resources from the Server

- The GET method is used to retrieve information from the given server using a given URI.
- Requests using GET should only retrieve data and should have no other effect on the data.

GET github.com/loadsmart



Métodos HTTP POST / PUT

Create/Edit resources in the Server

- A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
- PUT: Replaces all the current representations of the target resource with the uploaded content.

POST `www.exemplo.com/user/new`

Body:

nome: Pedro

sobrenome: Da Silva



Método HTTP

DELETE

Remove resources from the Server

- Removes all the current representations of the target resource given by URI.

DELETE www.exemplo.com/user/1



Responses

Status Code

1xx, **2xx**, 3xx, 4xx, 5xx

Header

- User-Agent

Body

- HTML document



Responses

Code and Description

1xx: Informational

It means the request was received and the process is continuing.

2xx: Success

It means the action was successfully received, understood, and accepted.

3xx: Redirection

It means further action must be taken in order to complete the request.

4xx: Client Error

It means the request contains incorrect syntax or cannot be fulfilled.

5xx: Server Error

It means the server failed to fulfill an apparently valid request.



django

Django

- É um framework para desenvolvimento web escrito em Python
- Free and Open Source

Por que Django?

- Rápido de fazer uma aplicação
- Possui várias ferramentas
- Versátil
- Escalável



Arquitetura

- MTV Framework: Model, Template and View

View

- Descreve **qual** dado será apresentado

Template

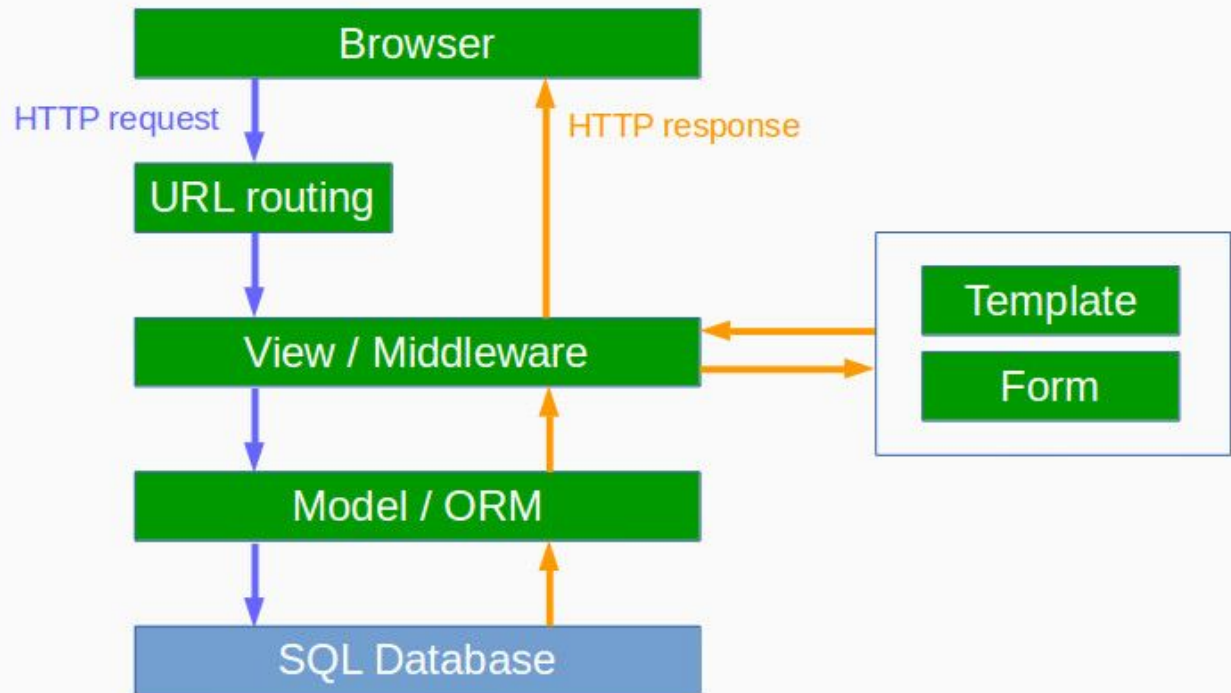
- Descreve **como** o dado será apresentado

Model

- Descreve as **entidades** da lógica do seu problema



Django's architecture



URLs

- Rotas do programa
- `urls.py`



View

- Descreve **qual** dado será apresentado

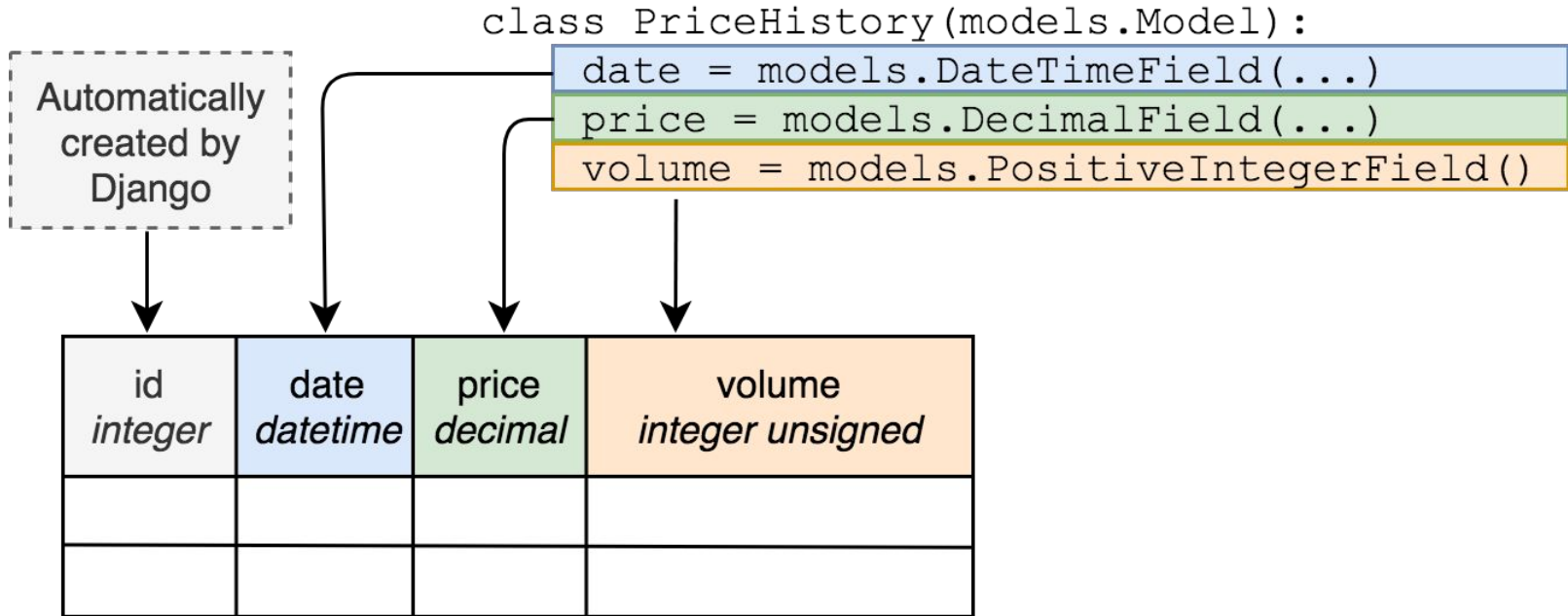


Model

- Descreve as entidades da lógica do seu problema
- Podem ou não ser uma tabela no banco de dados



Model



Migrações

- Change your models (in models.py).
- Run `python manage.py makemigrations` to create migrations for those changes
- Run `python manage.py migrate` to apply those changes to the database.



Shell

- `python manage.py shell`



Django Admin

- Django provides an admin area automatically





Dúvidas?

Conhecimento Passado

Legal, mas
não
aprendi
nada
novo, não.



Aprendi
muita
coisa
nova!

Conhecimento: 0-10



Velocidade

5: Velocidade
ideal.

ZzzZzzz,
pode
acelerar
isso aí.



Muito
rápido, tô
assimilando
o primeiro
slide ainda.

Velocidade: 0-10



Conteúdos

- <https://djangobook.com/mdj2-django-structure/>
- <https://www.dummies.com/education/internet-basics/dns-what-it-is-and-what-it-does/>
- <https://www.twoscoopspress.com/products/two-scoops-of-django-1-11>
- <https://www.w3.org/Protocols/rfc2616/rfc2616.html>
- <https://www.tutorialspoint.com/http/index.htm>
- <https://docs.djangoproject.com/en/1.11/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>
- [Desconstruindo a Web](#)
- [How Django URLs work with Regular Expressions](#)





MUITO OBRIGADA!