

Criação de um microserviço REST

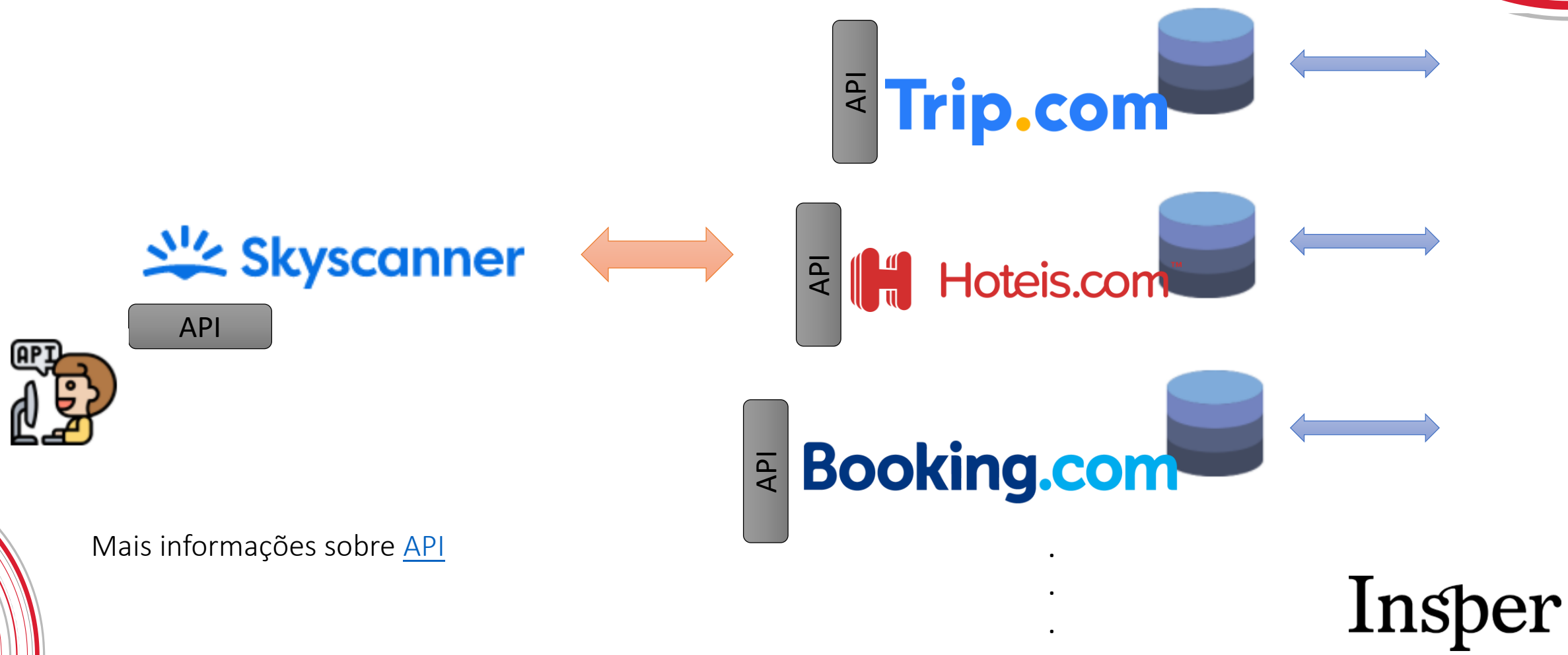
Aula teste para disciplina Tecnologias Web

Agenda

- Comunicação entre sistemas na web
- *Web services*
- HTTP
- Hands-on (Criando nosso *web service*)
- Arquitetura REST
- Modelo de maturidade de Richardson

Cenário

Busca de hotéis com [Skyscanner](#)



Famosos *web services*

Web service é um conjunto de métodos fornecidos e invocados por outros programas utilizando tecnologias Web.



- Protocolo: ?
- Meio de representar a informação: ?



API



Qual linguagem foi implementado?

Sistema Operacional?

Qual arquitetura do SW?
E da solução?

Resposta: POUCO IMPORTA!

Mais informações sobre [web services](#)

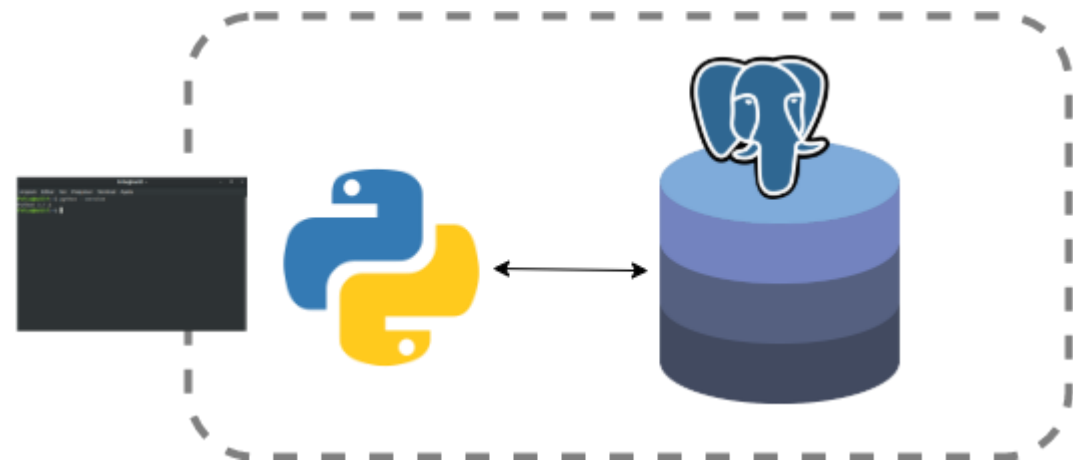
O que sabemos até aqui?

JavaScript

Python (procedural)

Banco de Dados (DML)

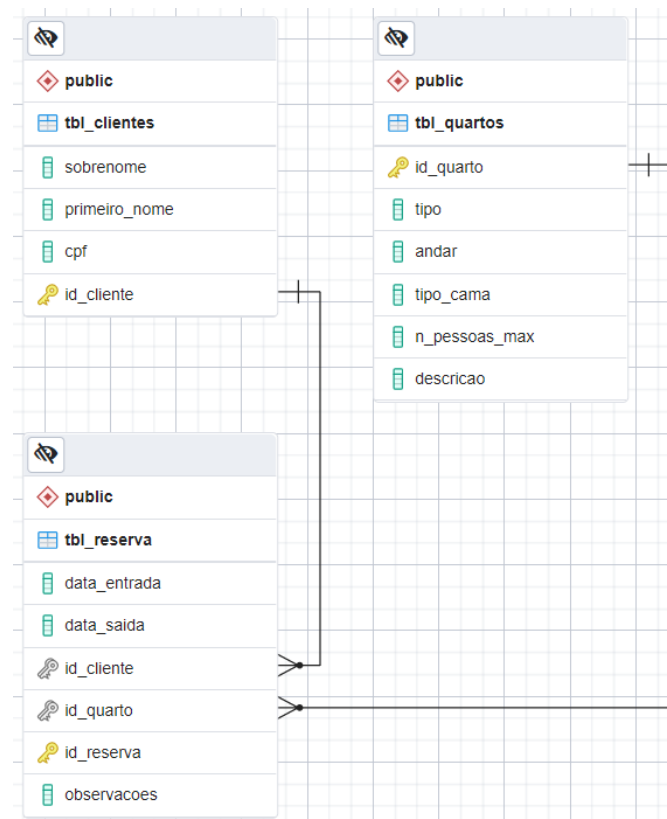
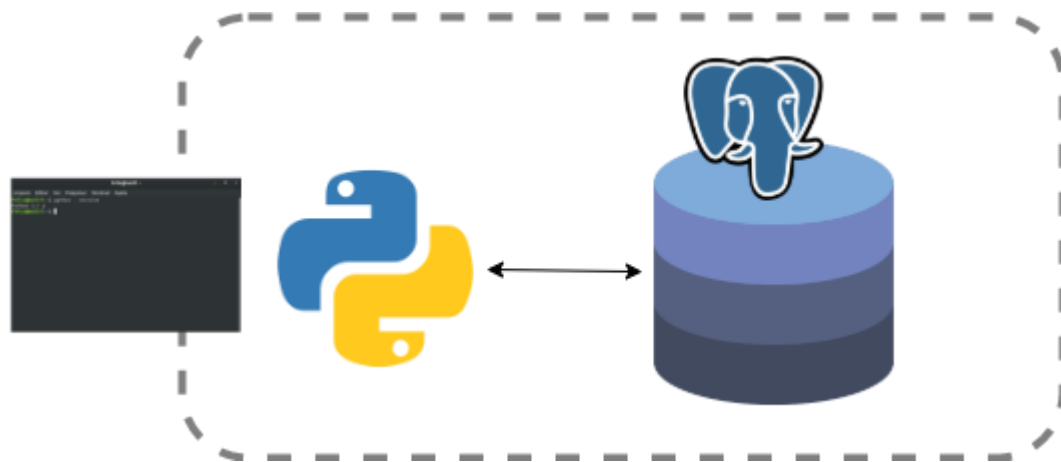
Básico de web (HTML, CSS e JS)



Github e vscode live share

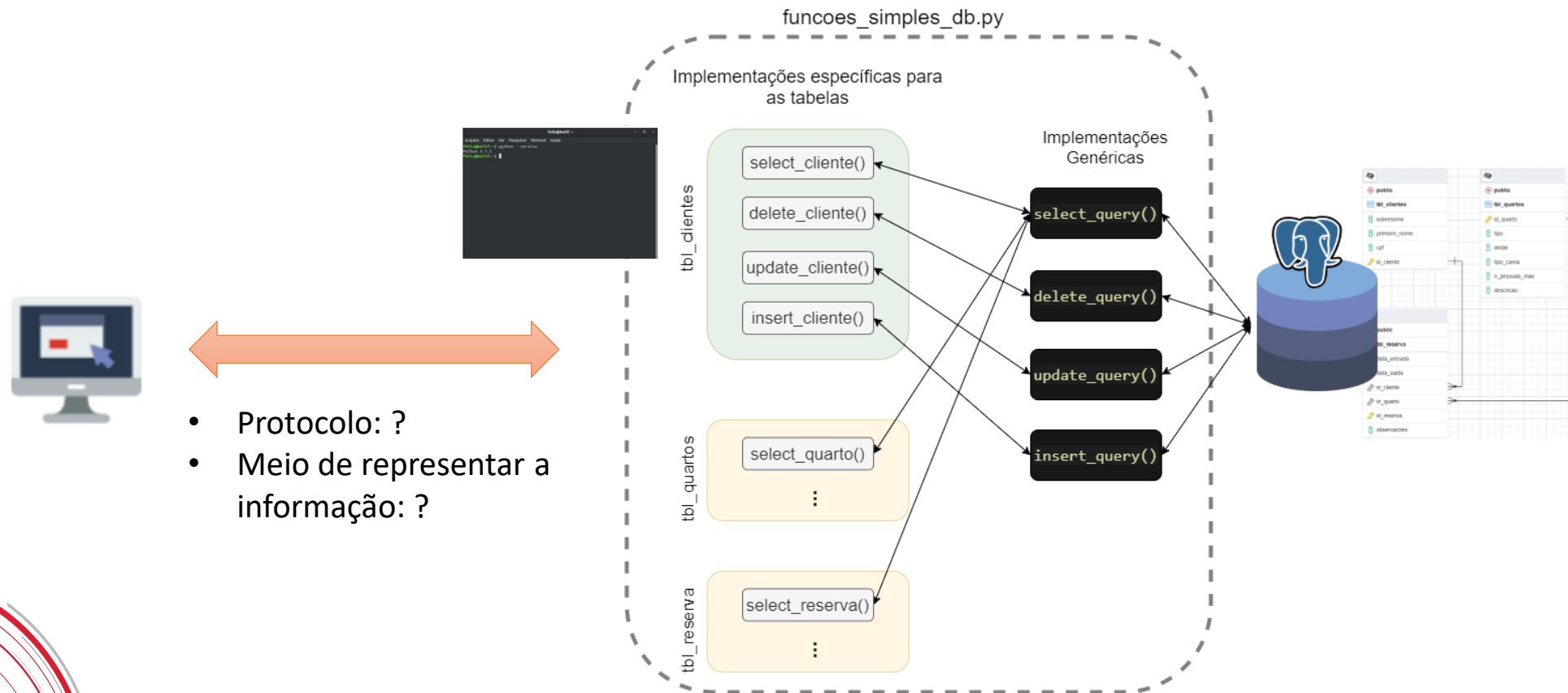
Repositório da aula de hoje: [link](#)

Exemplo: Micro Hotel

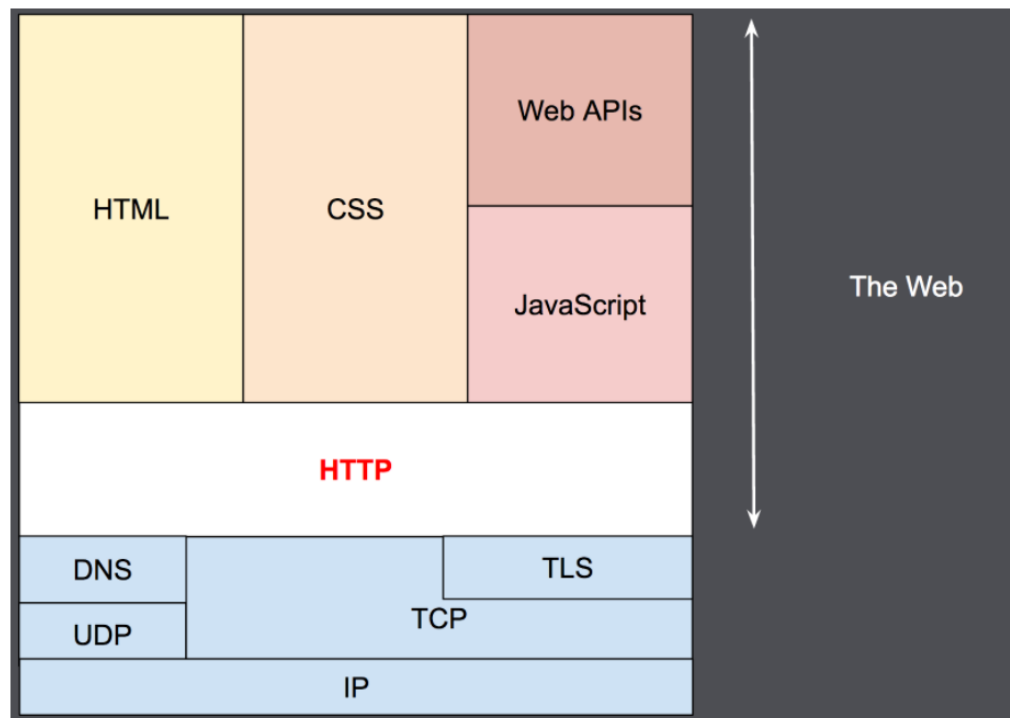


Github e vscode live share

Pasta: [2. app-Python-PostgreSQL](#)



HTTP (HyperText Transfer Protocol)



Mais informações sobre <http>

Requisição

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

Resposta

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (here comes the 29769 bytes
```


HTTP (HyperText Transfer Protocol)

| | |
|--------|---|
| GET | <ul style="list-style-type: none">• Buscar recursos• Cache |
| POST | <ul style="list-style-type: none">• Criar um novo recurso |
| PUT | <ul style="list-style-type: none">• Atualizar um recurso existente |
| DELETE | <ul style="list-style-type: none">• Remover um recurso |

Requisição

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

Resposta

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

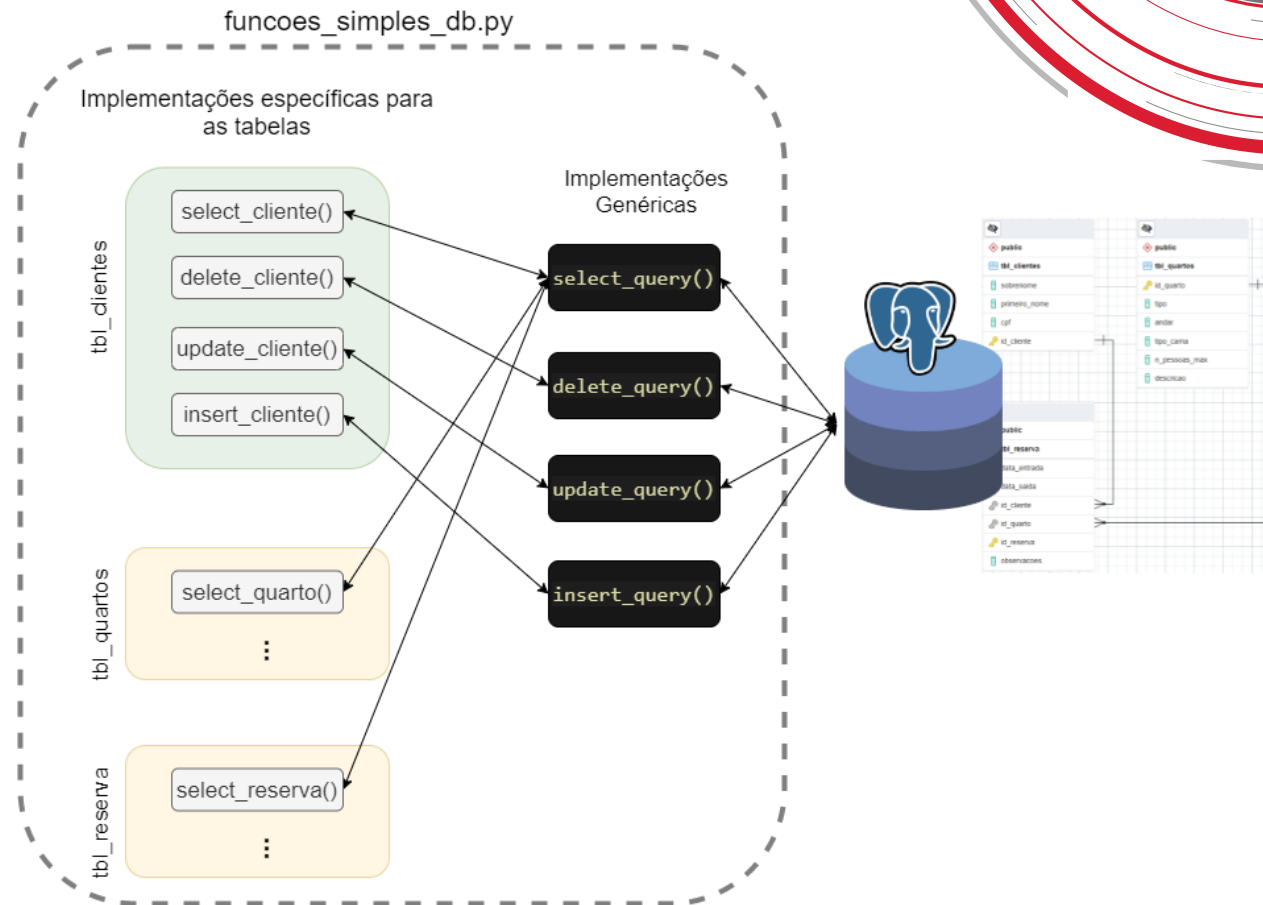
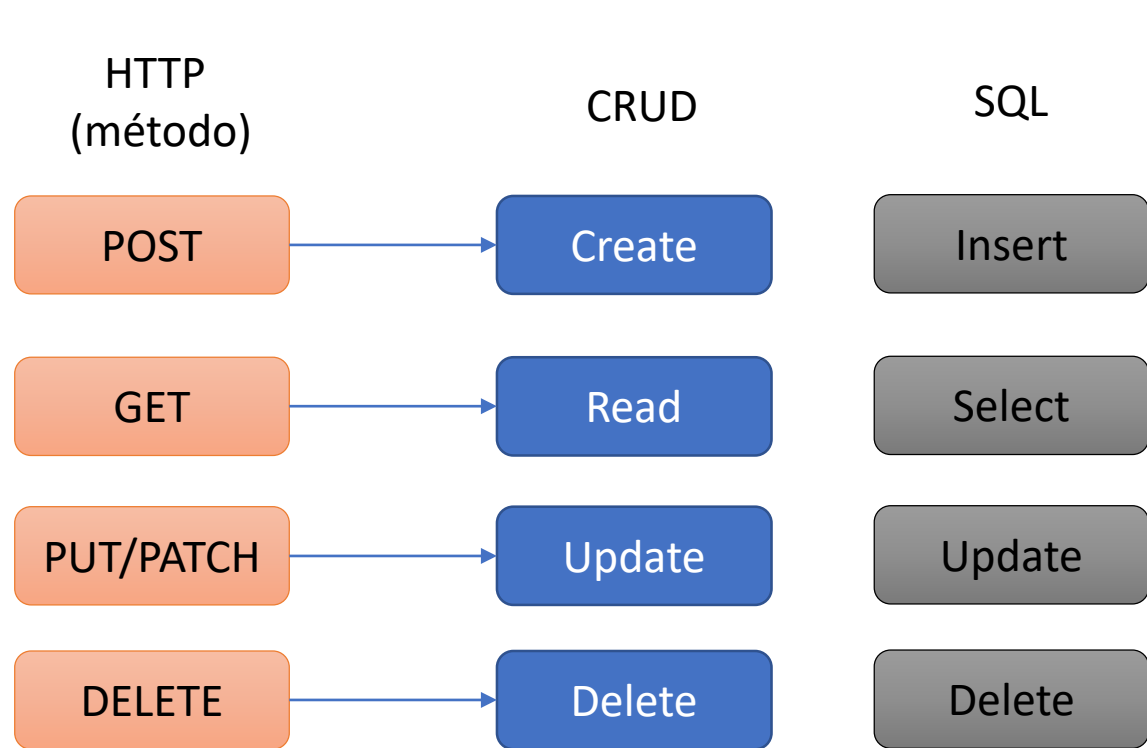
<!DOCTYPE html... (here comes the 29769 bytes
```

| Status code | Status code | Status code |
|-------------|------------------|---------------------------|
| 200 OK | 400 Bad Request | 500 Internal Server Error |
| 201 Created | 401 Unauthorized | |
| | 403 Forbidden | |
| | 404 Not Found | |

Mais informações sobre [verbos http](#)

Mais informações sobre [status code](#)

CRUD (Create, Read, Update e Delete)



Hands-on (talk is cheap...)

Ingressar na VS Code Live Session do professor.

REST (Representational State Transfer)

É uma arquitetura de *web servisse*

Core de um web servisse REST:

- Recursos
- Uso explícito dos verbos HTTP
- Sem estado (*stateless*)
- Respostas HTTP coerentes (*status code*)
- Representação (normalmente JSON)

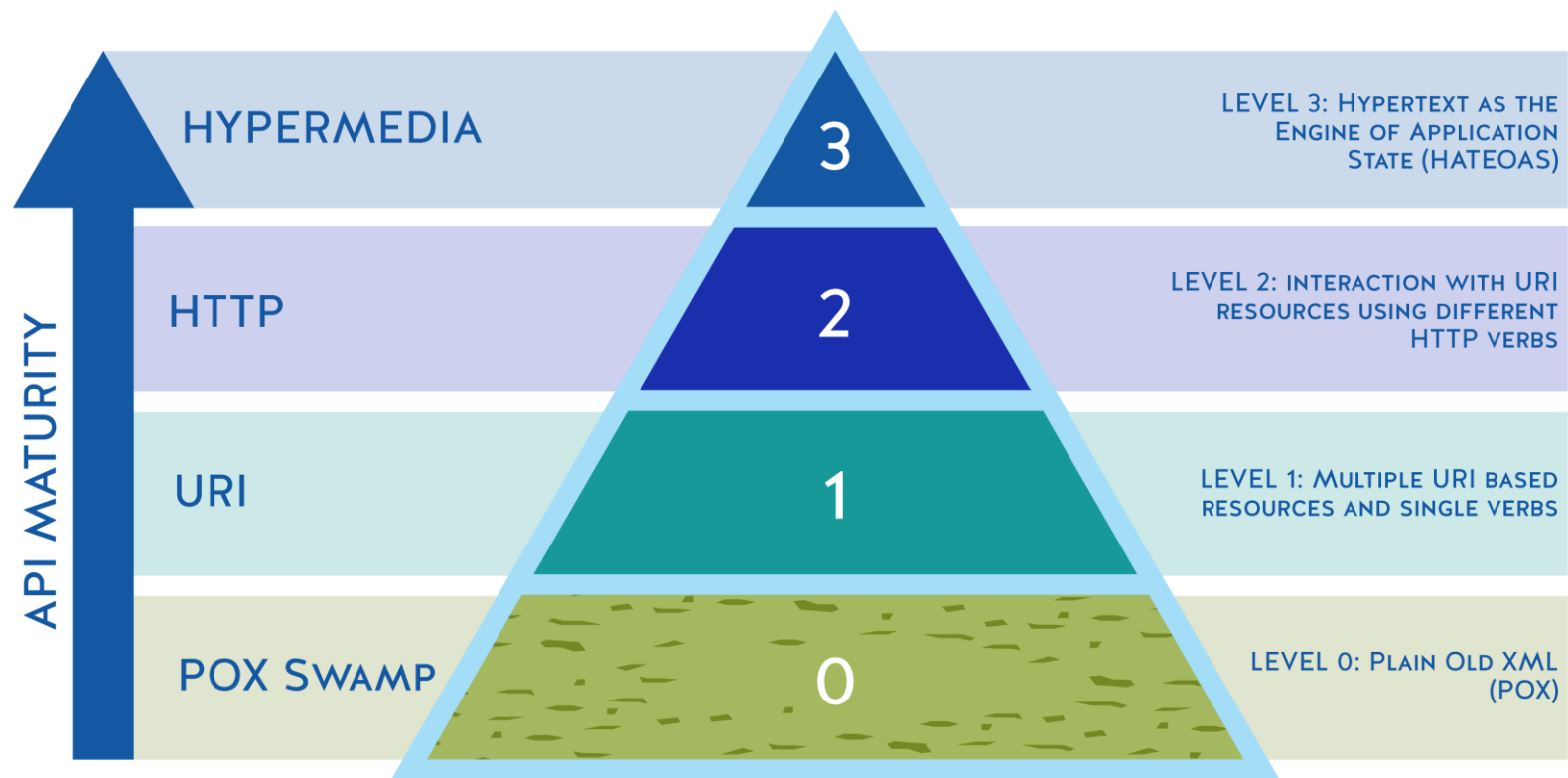
Completaremos a lista nas aulas que seguirão

Mais informações sobre [REST](#)

Dissertação [Roy Fielding](#)

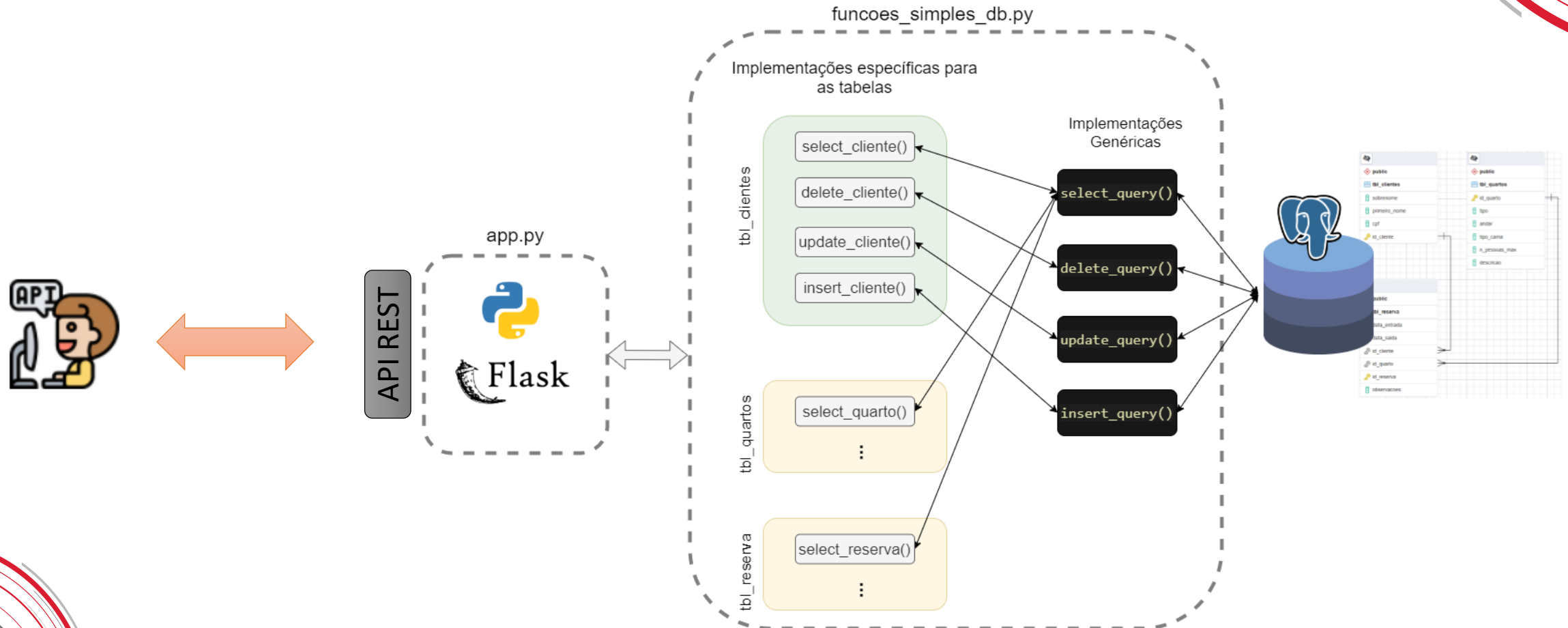
Maturidade de Richardson

RICHARDSON MATURITY MODEL

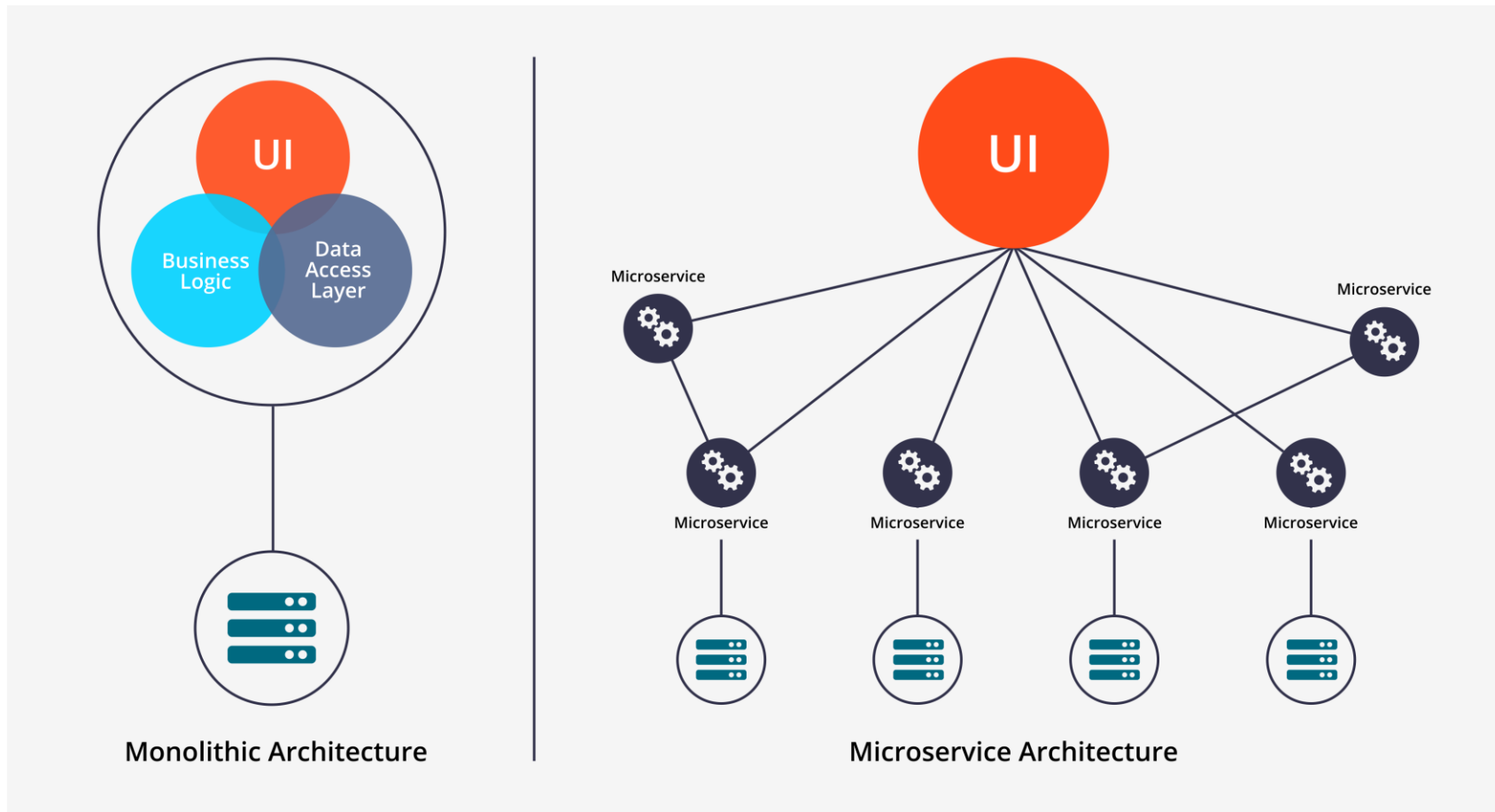


Fonte

Visão final do nosso sistema (consumo com JS)



Monolítico x Microserviços



Fonte

Consolidando

- Utilidade dos Web Services
- Princípios da arquitetura REST
- Framework Flask (REST em python)
 - Flask não é servidor HTTP
- A importância do protocolo HTTP
- Utilização de ferramentas REST Client para testar a API durante o desenvolvimento
- Modelo de Maturidade de Richardson



FIM

Tiago Sanches da Silva

Insper