

Escola de Férias

Node (Backend)

Geovane Fonseca de Sousa Santos
Dezembro 2019

Sumário

- Banco de Dados
- *Backend e Frontend*
- *API e Web Services*
- Principais Protocolos para *Web Services*
- O que iremos usar
- *Code and Teach*

Banco de Dados

Definição: Coleções organizadas de dados que se relacionam entre si de forma a criar algum sentido e ter eficiência para recuperação de informações.

Tipos:

- Bases de Dados relacionais
- Base de Dados não-relacionais

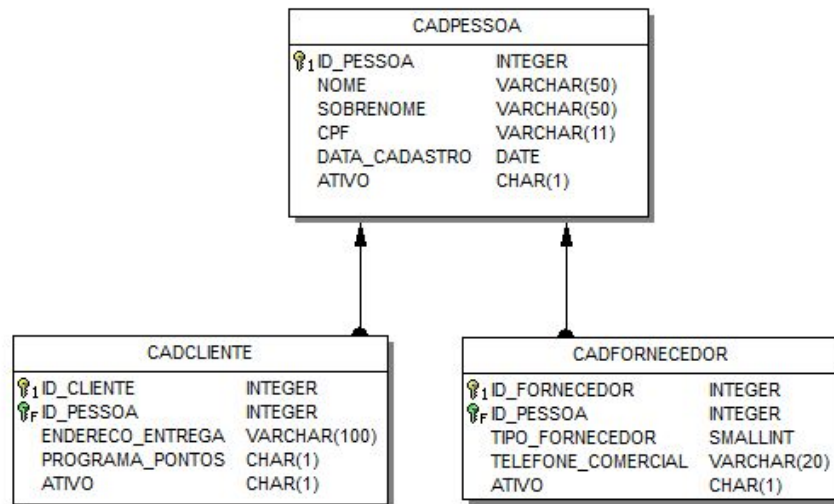
Base de Dados Relacionais

Definição: Os dados se relacionam entre si por meio dos seguintes componentes:

- Uma coleção de estruturas de dados (**Ex:** Relações, Tabelas)
- Uma coleção dos operadores relacionais (**Ex:** Tem vários, pertence a um)
- Uma coleção de restrições de integridade (**Ex:** Tipo, atributo, relvar (variável relacional) e restrição de base)

Base de Dados Relacionais

Exemplos: Postgresql, Mysql, Oracle, etc...



Base de Dados Não-Relacionais

Definição: Não utiliza a sintaxe SQL e são modelados de formas diferentes das tradicionais tabelas SQL

Motivações:

- Simplicidade
- Grande número de dados (usadas em Big Data)
- Velocidade

Problema:

- Fraca consistência de dados (necessita de middlewares)

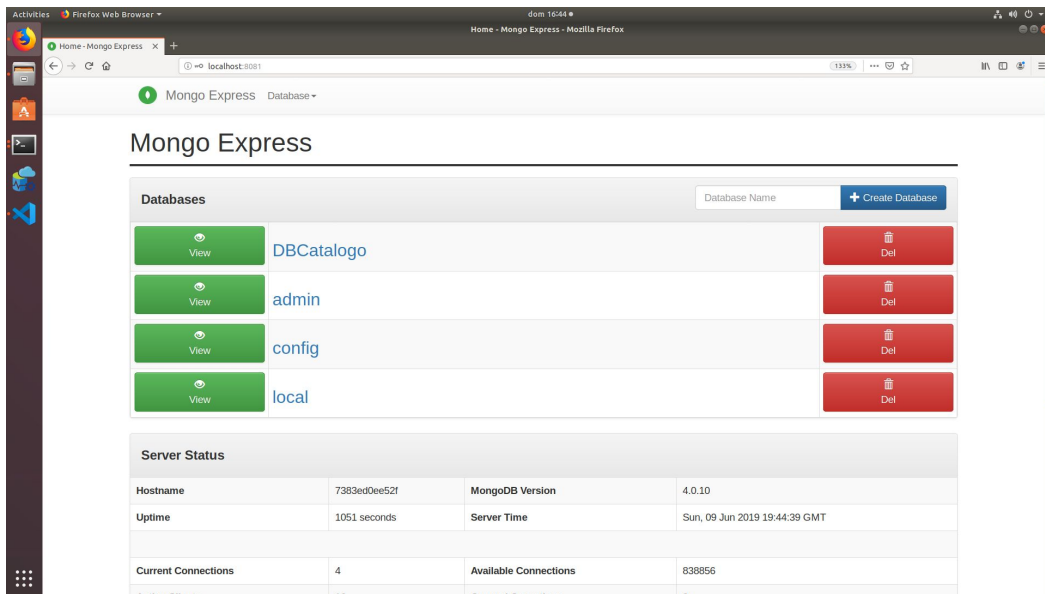
Base de Dados Não-Relacionais

Tipos:

- **Coluna:** Accumulo, Cassandra, Druid
- **Documento:** Cosmos DB, IBM Dominio, MongoDB
- **Chave-valor:** Oracle NoSQL Database, OrientDB, Virtuoso
- **Grafo:** AllegroGraph, ArangoDB, InfiniteGraph
- **Multi-modelo:** Apache Ignite, ArangoDB, MarkLogic

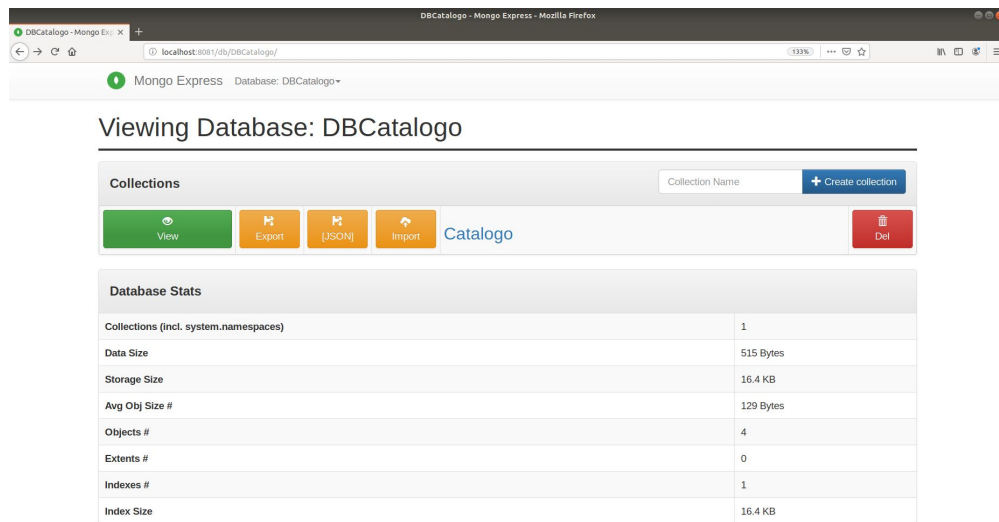
Base de Dados Não-Relacionais

Exemplo: MongoDB (Documento: Databases)



Base de Dados Não-Relacionais

Exemplo: MongoDB (Documento: Collections)



The screenshot shows the MongoDB Express web interface in a Mozilla Firefox browser. The address bar shows the URL `localhost:8081/db/DBCatalogo/`. The page title is "Mongo Express Database: DBCatalogo".

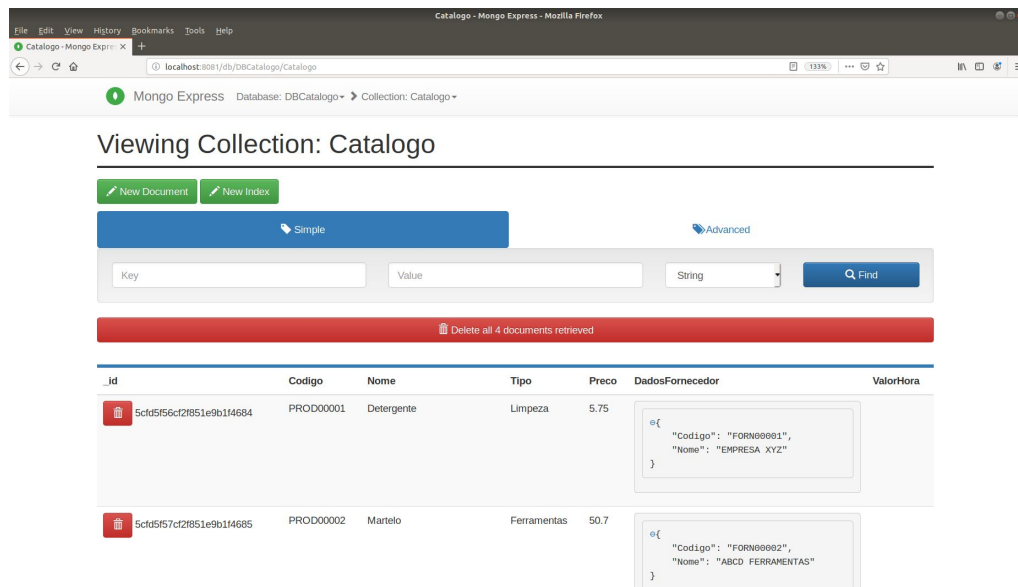
The main section is titled "Viewing Database: DBCatalogo". It features a "Collections" section with a search bar and a "+ Create collection" button. Below this, there are buttons for "View", "Export", "JSON", "Import", and "Catalogo" (highlighted in blue). A red "Def" button is also present.

Below the buttons is a "Database Stats" section with a table showing various statistics:



Database Stats	
Collections (incl. system.namespaces)	1
Data Size	515 Bytes
Storage Size	16.4 KB
Avg Obj Size #	129 Bytes
Objects #	4
Extents #	0
Indexes #	1
Index Size	16.4 KB

Base de Dados Não-Relacionais

Exemplo: MongoDB (Documento: Indexes)



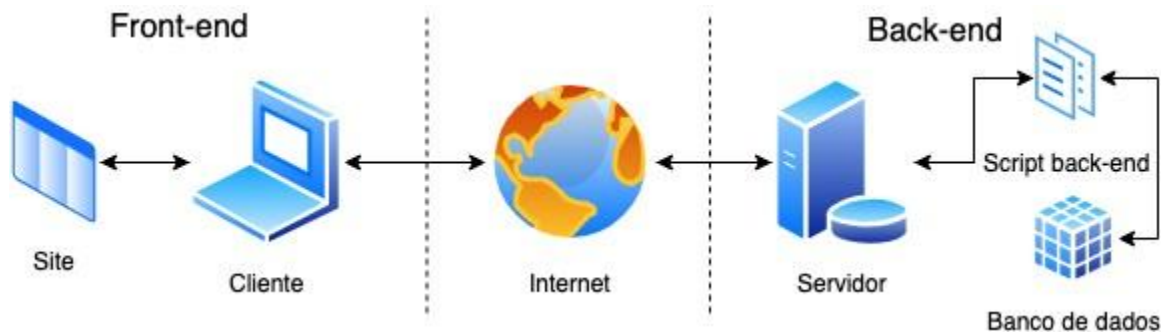
The screenshot shows the Mongo Express web interface in a Mozilla Firefox browser. The address bar shows the URL `localhost:8081/db/Catalogo/Catalogo`. The interface displays the 'Catalogo' collection with two documents. Above the table, there are buttons for 'New Document' and 'New Index', and a search bar with 'Key' and 'Value' fields. A red banner indicates 'Delete all 4 documents retrieved'. The table has columns: `_id`, `Codigo`, `Nome`, `Tipo`, `Preco`, `DadosFornecedor`, and `ValorHora`.

<code>_id</code>	<code>Codigo</code>	<code>Nome</code>	<code>Tipo</code>	<code>Preco</code>	<code>DadosFornecedor</code>	<code>ValorHora</code>
 5cd5f56cf2f851e9b114684	PROD00001	Detergente	Limpeza	5.75	<pre>{ "Codigo": "FORN00001", "Nome": "EMPRESA XYZ" }</pre>	
 5cd5f57cf2f851e9b114685	PROD00002	Martelo	Ferramentas	50.7	<pre>{ "Codigo": "FORN00002", "Nome": "ABCO FERRAMENTAS" }</pre>	

Backend e Frontend

Frontend: Interface de Interação com o usuário

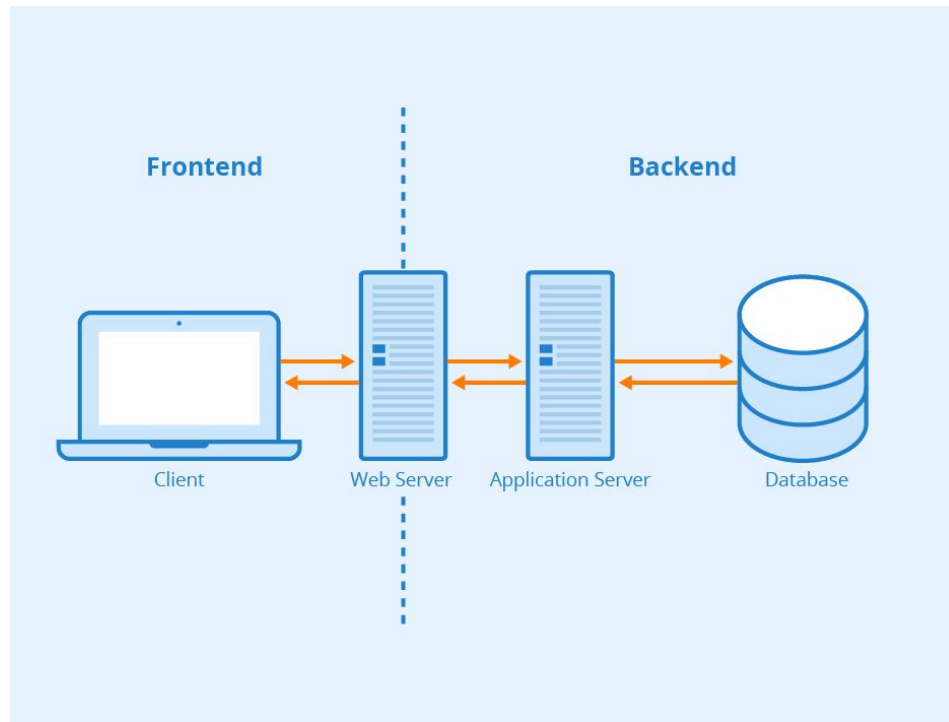
Backend: Sistema responsável pela Regra de Negócios, WebServices e APIs de uma aplicação



Backend e Frontend

Por que separar?

- Manutenção
- Legibilidade
- Velocidade
- Escalabilidade

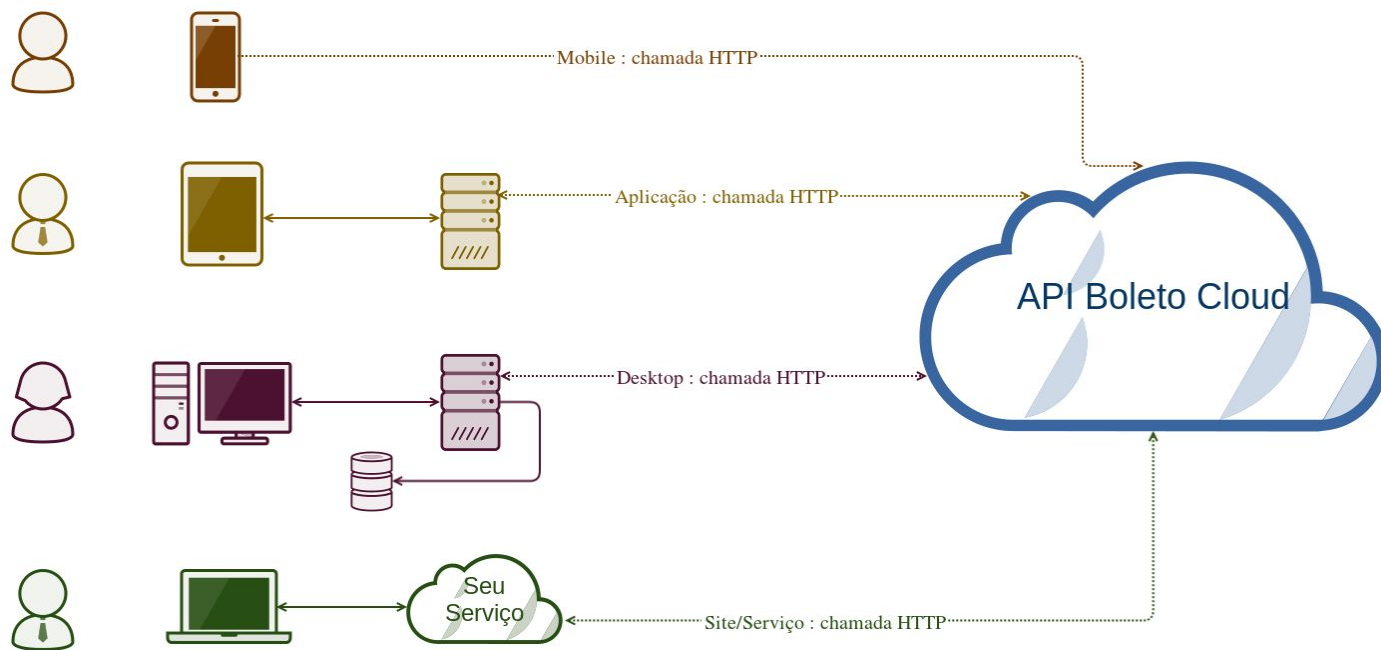


API e Web Services

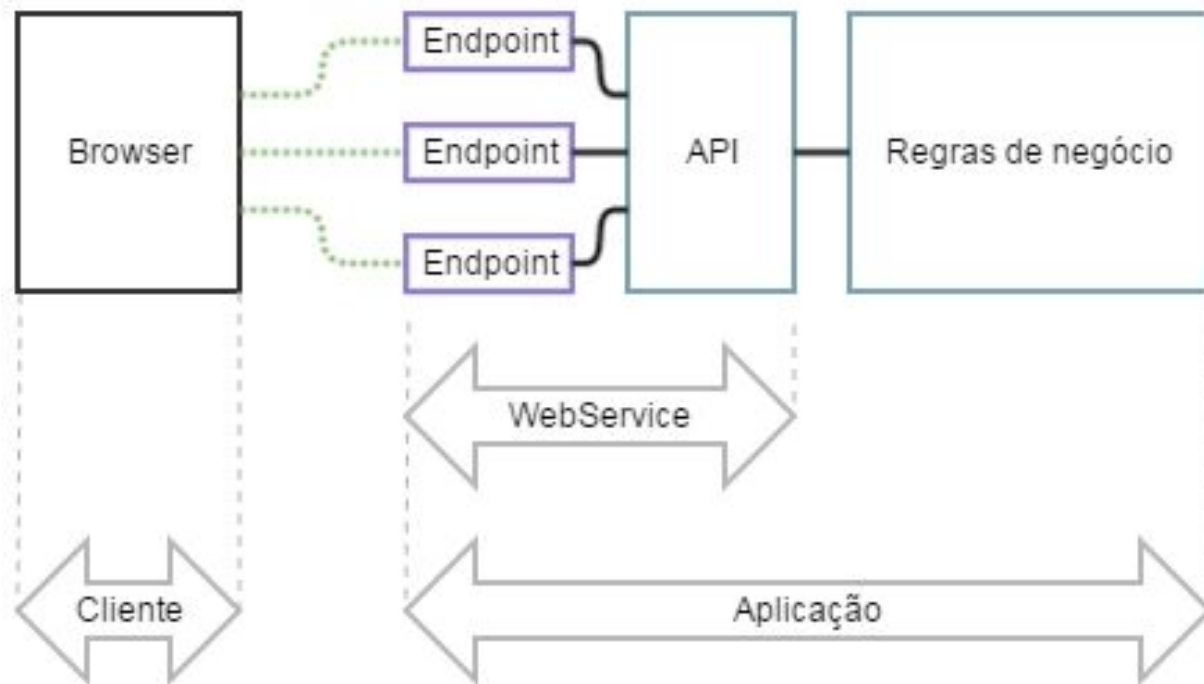
API (Application Programming Interface): Conjunto de rotinas e padrões de programação para acesso a algum aplicativo ou plataforma na *Web*

Web Services: Solução utilizada para integrar sistemas e na comunicação de aplicações diferentes

API



Web Services



Principais Protocolos para *Web Services*

Os principais protocolos para criação dos Web Services atualmente são:

- SOAP
- REST
- GraphQL (Facebook)

SOAP (*Simple Object Access Protocol*)

Para cada serviço um *endpoint*

Formato de mensagem: XML

Transferência: HTTP

Requisição:

```
1 <?xml version="1.0"?>
2 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3   <soap:Header>
4   </soap:Header>
5   <soap:Body>
6     <m:GetRevista xmlns:m="http://www.example.org/revista">
7       <m:RevistaNome>Java Magazine</m:RevistaNome>
8     </m:GetRevista>
9   </soap:Body>
10 </soap:Envelope>
```

REST (*Repesentional State Transfer*)

Para cada serviço, um *endpoint*

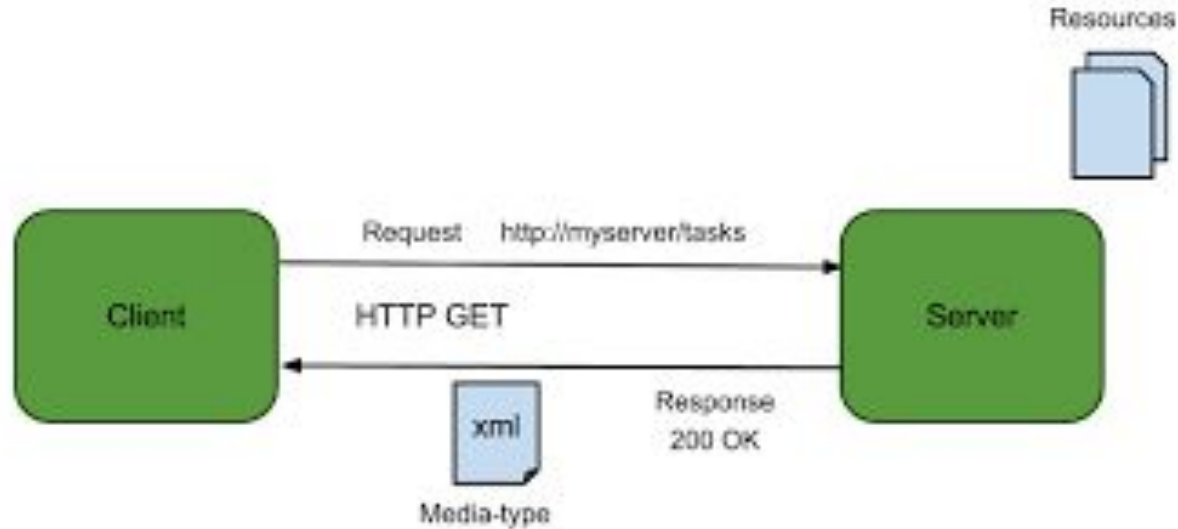
Formato da mensagem: HTML, XML ou JSON

Transferência: HTTP

Métodos HTTP: GET, HEAD, POST, PUT, PATCH, DELETE,
CONNECT, OPTIONS e TRACE

REST (*Repesentational State Transfer*)

Requisição:



GraphQL

Um endpoint para todos os serviços

Formato da mensagem: Query

Transferência: HTTP

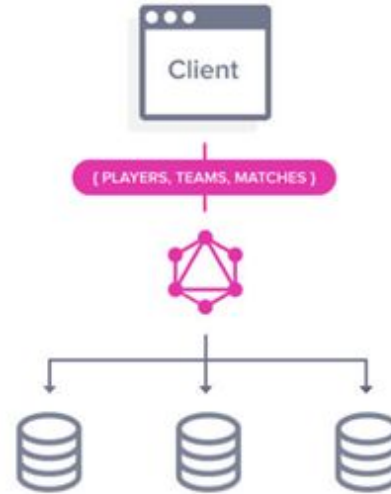
Requisição: Um endpoint como no REST, mas com uma Query associada ao corpo da requisição

GraphQL

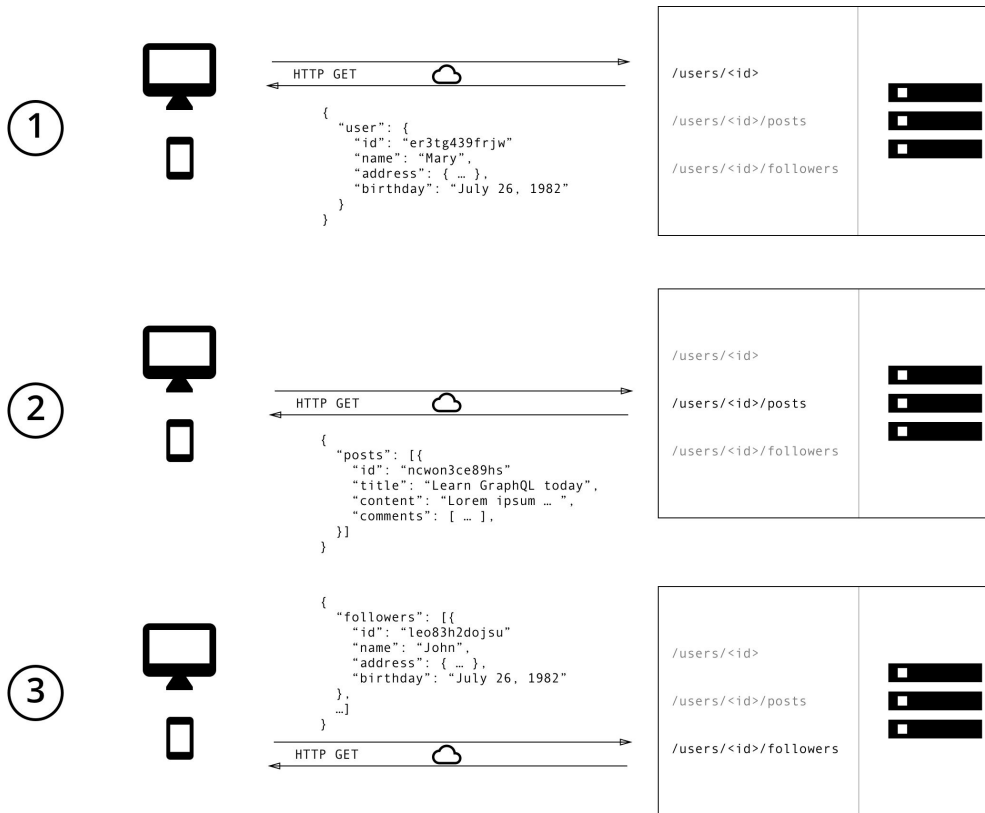
Rest API



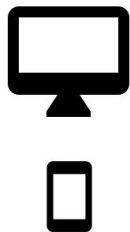
GraphQL API



GraphQL (Exemplo REST)



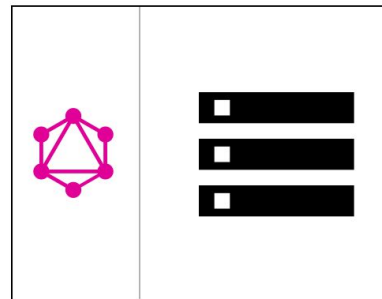
GraphQL (Exemplo)



```
query {  
  User(id: "er3tg439frjw") {  
    name  
    posts {  
      title  
    }  
    followers(last: 3) {  
      name  
    }  
  }  
}
```

HTTP POST

```
{  
  "data": {  
    "User": {  
      "name": "Mary",  
      "posts": [  
        { title: "Learn GraphQL today" }  
      ],  
      "followers": [  
        { name: "John" },  
        { name: "Alice" },  
        { name: "Sarah" },  
      ]  
    }  
  }  
}
```



O que usaremos

Banco de Dados: MongoDB

Gerenciador de dependências: npm

API: Nodejs

Web Service: REST (RESTful)

Para Testar API: Postman

Code and Teach

Material Aula (Slide):

https://docs.google.com/presentation/d/1_Q33xIPC54dA42BN_ojwDKNnK2oSHmXDb2ZzIq9Fj0CU/edit?usp=sharing

Código Aula:

<https://github.com/GeovaneF55/edf-code>