

Laboratório de Fundamentos em TIC

Introdução às Arquiteturas de *Software* e Dados

Prof. Gabriel Resende Machado

 gabrielmachado@unifeso.edu.br

 <https://www.linkedin.com/in/machadogabriel>

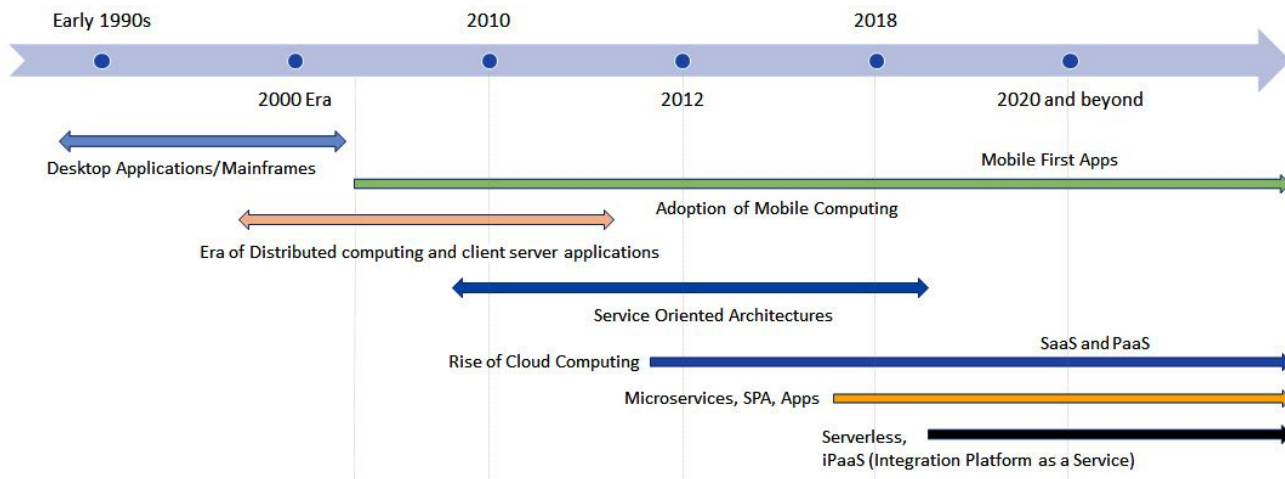
 <https://github.com/UNIFESO-Gabriel/fundamentos-em-tic>

Arquiteturas Físicas de *Software*

- *Softwares* podem serem implementados a partir de diferentes arquiteturas físicas que **busquem atender às necessidades dos usuários, plataformas e ambientes**.
- Atualmente, arquiteturas físicas de *softwares* podem ser do tipo:
 - **Desktop:** geralmente envolve a criação de aplicativos que são executados localmente em um computador. As linguagens de programação comumente utilizadas incluem C, C++, Rust, C#, Delphi (*Object Pascal*) e Java;
 - **Web:** aplicações acessadas por meio de um navegador (*browser*). Não dependem de recursos do computador local, pois o processamento e armazenamento são feitos em servidores remotos. Necessita de acesso à Internet;
 - **Mobile:** aplicativos destinados aos dispositivos móveis, como *smartphones e tablets*. Oferecem mobilidade e extensão de funcionalidade a partir de recursos como GPS, câmera e biometria;
 - **Cloud:** visam prover recursos de infraestrutura e componentes para facilitar o desenvolvimento de aplicações, sua operacionalidade, disponibilidade, segurança e integração. São serviços pagos, geralmente pelo método *pay-as-you-go*.

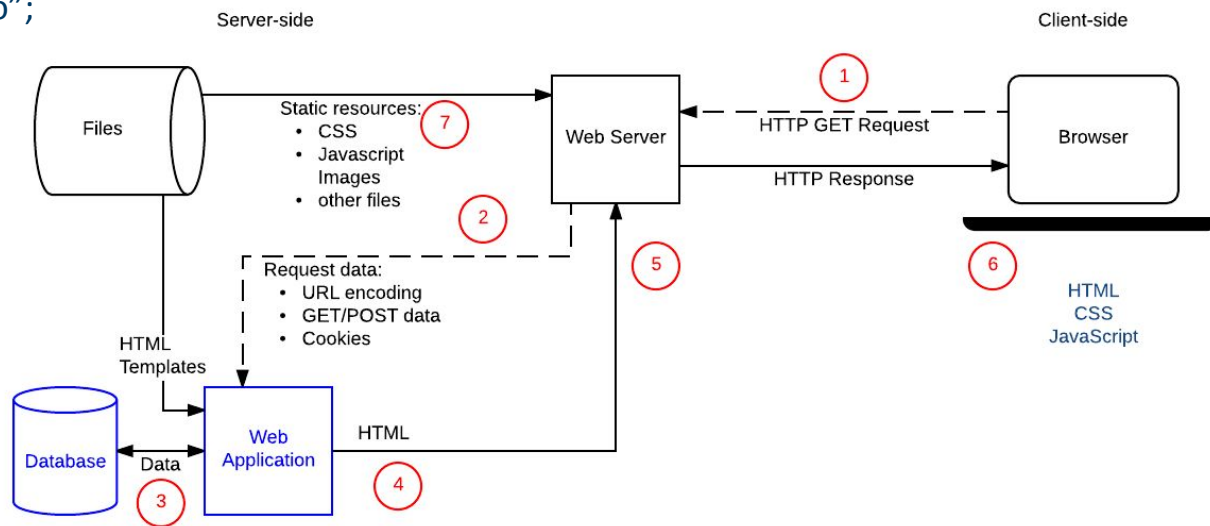
Arquiteturas Físicas de *Software* - Aplicações *Desktop*

- Geralmente são aplicações monolíticas, desenvolvidas para serem executadas em um computador pessoal. Frequentemente persistirem os dados no próprio PC ou em um servidor local;
- Variam desde aplicações pessoais como calculadoras, editores de texto, reprodutores multimídia, até aplicações gerenciais, como PDVs, ERPs, CRMs, entre outros;
- Dificilmente deixarão de existir, porém o desenvolvimento desses aplicativos têm decaído muito nos últimos anos, principalmente com a popularização da Internet e aplicações *Web* e *Mobile*.



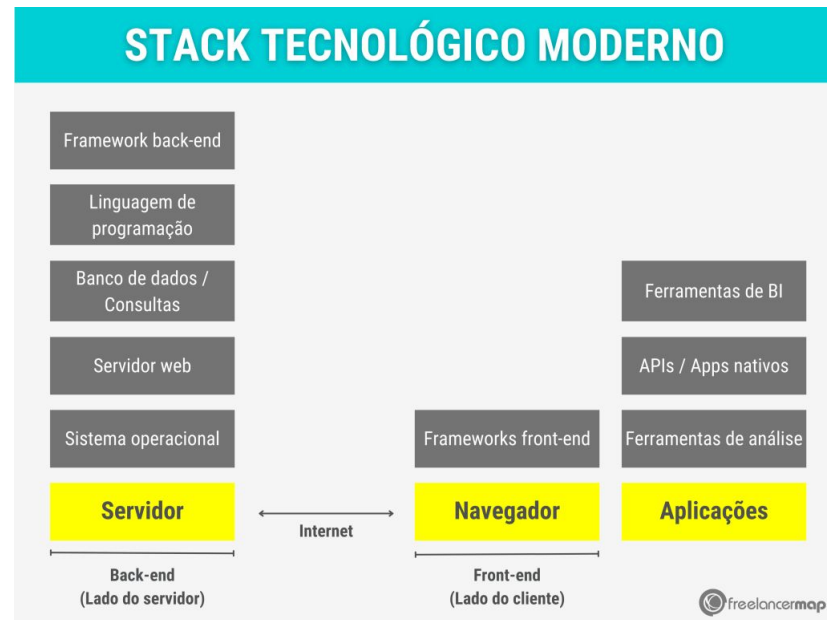
Arquiteturas Físicas de *Software* - Aplicações *Web* I

- Adotam a arquitetura **cliente-servidor**: a máquina cliente faz uma requisição à máquina servidora, que responde à requisição fornecendo os dados solicitados pela máquina cliente;
- **São imprescindíveis**: (i) uma conexão com a Internet, (ii) um *browser* (cliente), (iii) um ou mais servidores, (iv) protocolos de comunicação (HTTP) e (v) políticas de segurança da informação;
- São, atualmente, as aplicações mais populares no ramo de desenvolvimento de *software*. Não necessitam “instalação”;



Arquiteturas Físicas de *Software* - Aplicações Web II

- Aplicações Web geralmente são divididas em (i) **back-end** e (ii) **front-end**:
- O **back-end** é a parte do software que é executada no servidor. Gerencia a lógica de negócio, o armazenamento e recuperação de dados, como também a integração com outros sistemas. Atualmente, são utilizadas como principais linguagens de programação C#, Node.js, Python, PHP, Ruby e Java;
- O **front-end** é a parte da aplicação executada no lado do cliente, resultante de uma requisição. O **front-end** é administrado pelo *browser*, responsável por renderizar o (i) HTML, (ii) estilos CSS e (iii) scripts JavaScript;
- Atualmente, há vários *frameworks* para desenvolvimento Web, tanto para o **back-end**, como para o **front-end**. Combinações desses *frameworks* resultam em stacks.



Disponível em <https://shorturl.at/gmJT3>. Acesso em 13/05/2024.

Arquiteturas Físicas de *Software* - Aplicações *Mobile*

- Possuem uma inspiração semelhante às aplicações *Web*, **porém visando a portabilidade e mobilidade**;
- São aplicações desenvolvidas para dispositivos móveis, atualmente concentrados em dois sistemas operacionais: **(i) Android** (Flutter, Kotlin, React Native, Xamarin) e **(ii) iOS** (Swift, Objective-C, Xcode);
- Têm ganhado muito espaço no campo de desenvolvimento de *software* nos últimos anos, principalmente pela popularização dos dispositivos móveis e da Internet Wi-Fi e móvel (5G);
- Além da mobilidade, se beneficiam dos dispositivos acoplados aos aparelhos, como GPS, camera, *bluetooth* e o NFC para estenderem suas funcionalidades, como o reconhecimento biométrico;
- **Pode-se dizer que os dispositivos *mobile* têm se tornado “parte do corpo humano”?**



Disponível em <https://shorturl.at/eoKRV>.
Acesso em 13/05/2024.

Arquiteturas Físicas de *Software* - Aplicações *Cloud* I

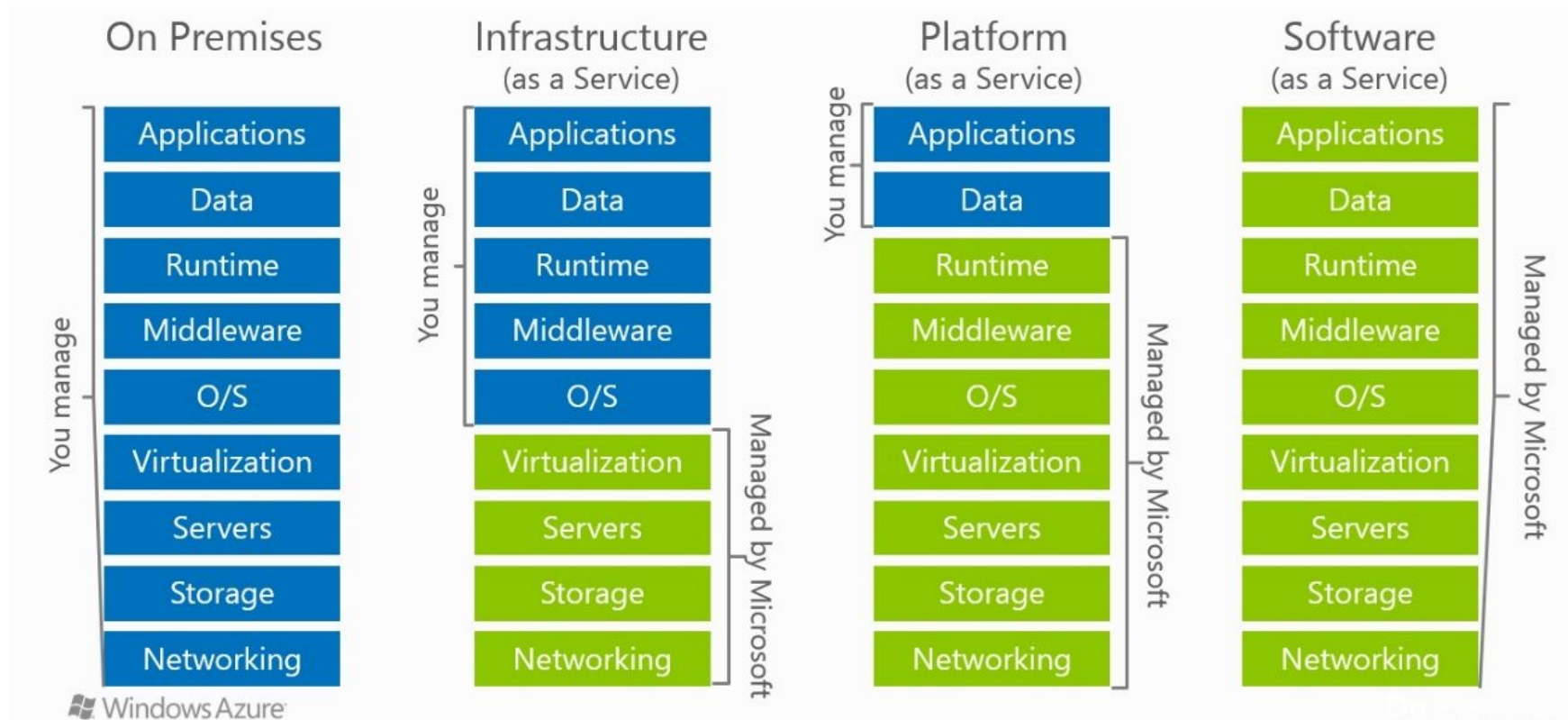
- Imagine que você é um empreendedor de uma startup que, segundo você, produzirá muitos lucros a partir dos seus produtos inovadores;
- Contudo, você não possui o capital necessário para investir em equipamentos e profissionais de infraestrutura;
- Ainda, há a probabilidade do negócio não prosperar e você perder todo seu investimento...
- Solução: investir em uma solução em nuvem! Os principais serviços hoje são (i) **Amazon Web Services (AWS)**, (ii) **Microsoft Azure** e (iii) **Google Cloud (GCP)**.



Arquiteturas Físicas de *Software* - Aplicações *Cloud* II

- Serviços em nuvem são soluções que visam a economia de recursos ao terceirizar serviços para os *vendors*;
- Com os serviços em nuvem, é possível publicar aplicações, ingerir, armazenar e processar grandes quantidades de dados, alocar máquinas virtuais conforme a necessidade, gerenciar usuários, entre outros serviços;
- As principais características da nuvem incluem:
 - **Escalabilidade/Elasticidade:** capacidade de dimensionar o sistema para lidar com variações nas cargas de trabalho e demanda. Scale-out: inclusão de mais instâncias de recursos, como servidores ou vertical; scale-up: aumento da capacidade dos recursos existentes, como CPU e RAM;
 - **Disponibilidade:** as arquiteturas em nuvem são projetadas para fornecer alta disponibilidade e confiabilidade, principalmente via contratos de SLA (*Service Level Agreement*);
 - **Tolerância a falhas:** as aplicações em nuvem são replicadas em diferentes *racks* ou até mesmo *data centers*, visando alta disponibilidade e segurança contra acidentes.
 - **Segurança:** soluções em nuvem contam com soluções de ponta baseadas em criptografia de dados, autenticação de usuários e controle de acesso.

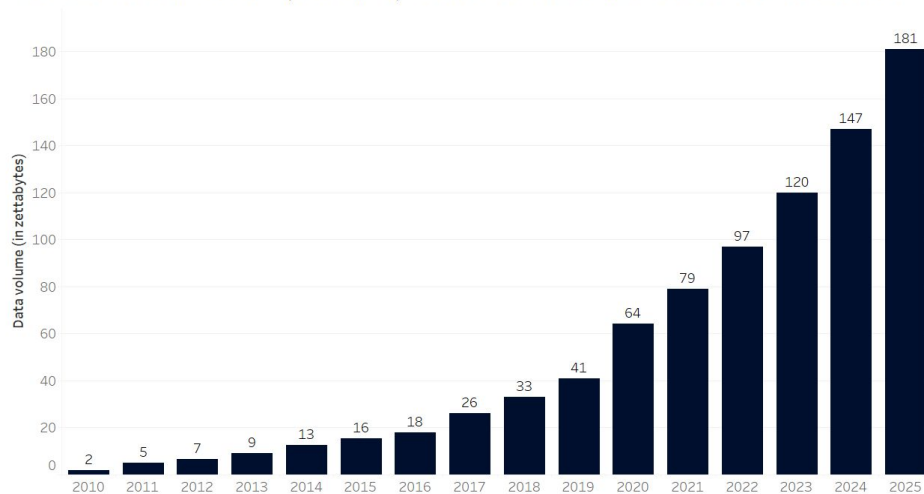
Arquiteturas Físicas de *Software* - Aplicações *Cloud* III



Disponível em: <https://shorturl.at/BHO28>. Acesso em 13/05/2024.

Dados: O Ouro do Século XXI

Volume of data created, captured, copied and consumed worldwide from 2010 to 2025



Disponível em <https://bit.ly/3WFQGoR>.

- De acordo com <https://bit.ly/44l6vxn>, em 2019, por dia, em média:
 - 500 milhões de *tweets* foram postados;
 - 294 bilhões de *emails* foram enviados;
 - 95 milhões de fotos foram compartilhadas no *Instagram*;
 - 5 bilhões de buscas foram realizadas, sendo 3,5 bilhões no *Google*;
 - 4 *petabytes* de dados foram criados pelo *Facebook*.

Abreviação	Unidade	Valor
b	bit	0 or 1
B	bytes	8 bits
KB	kilobytes	1,000 bytes
MB	megabyte	1,000 ² bytes
GB	gigabyte	1,000 ³ bytes
TB	terabyte	1,000 ⁴ bytes
PB	petabyte	1,000 ⁵ bytes
EB	exabyte	1,000 ⁶ bytes
ZB	zettabyte	1,000 ⁷ bytes
YB	yottabyte	1,000 ⁸ bytes



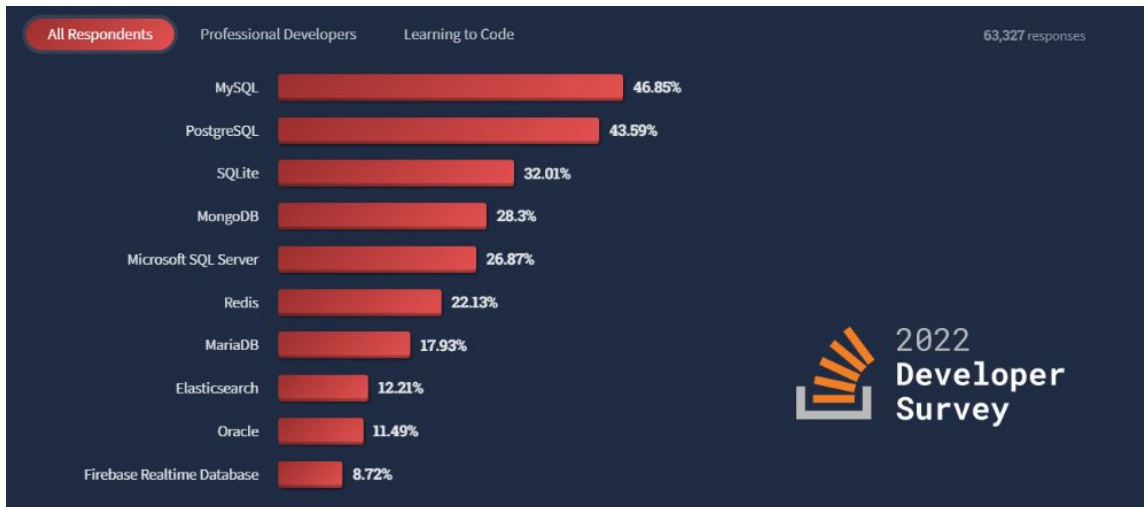
1 ZB equivale a
1.073.741.824 HDs 1TB

Banco de Dados e SGBDs I

- Praticamente, qualquer aplicação desenvolvida nos dias de hoje possui alguma interação com dados;
- **Um banco de dados** é um conjunto organizado de dados que são armazenados e gerenciados de forma estruturada para serem acessados e manipulados de maneira eficiente;
- Por sua vez, um **Sistema Gerenciador de Banco de Dados (SGBD)** é um software que permite a criação, manutenção e gerenciamento de bancos de dados. Ele fornece uma interface entre os usuários e o banco de dados, permitindo que os usuários criem, modifiquem e excluam dados, bem como executem consultas e gerem a segurança e a integridade dos dados;
- Muito é investido em infraestrutura para que, a partir dos dados, sejam descobertas tendências, padrões e realizadas previsões que venham a alavancar os lucros de uma empresa, otimizar e automatizar serviços e diminuir gastos;
- Profissões relacionadas à análise, ciência e engenharia de dados têm crescido exponencialmente, porém ainda há grande defasagem profissional no mercado.

Banco de Dados e SGBDs II

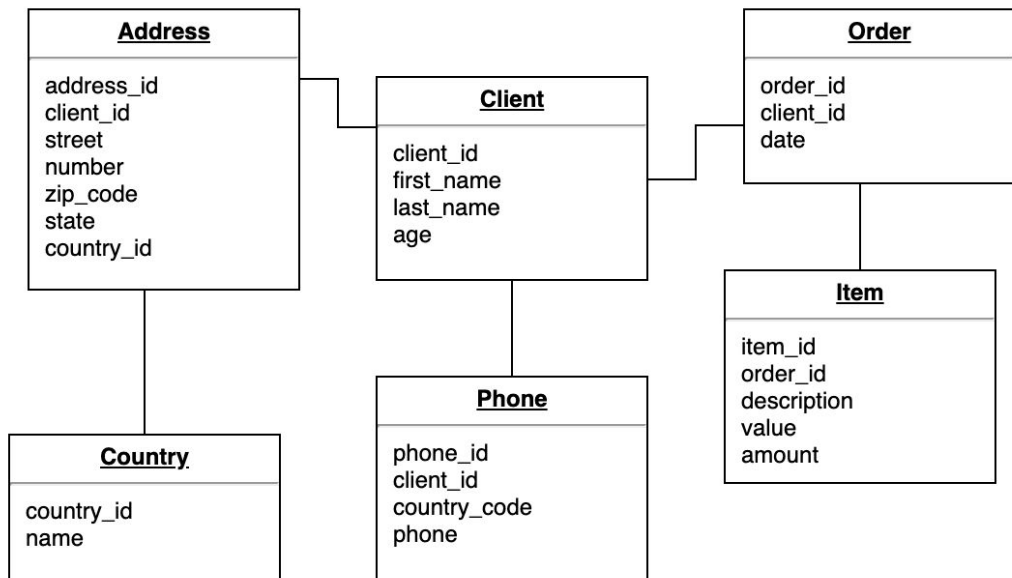
- Os dados podem ser classificados em (i) **estruturados**, (ii) **semi-estruturados** ou (iii) **desestruturados**;
 - Os dados estruturados são mais conhecidos como dados relacionais, ou tabulares;
 - Bancos de dados relacionais trabalham com esquemas pré-definidos em tabelas, relações via chave-primária e estrangeira e garantia de consistência.
 - Por muitos anos, até os dias atuais, os bancos de dados relacionais representam a solução mais utilizada nas aplicações de *software*. Por esse motivo, serão abordados com um pouco mais de detalhes neste curso.



Disponível em: <https://bit.ly/3wwYXky>. Acesso em 13/05/2024.

Banco de Dados e SGBDs III - Modelo Relacional

- Um modelo relacional é composto por tabelas e relações entre essas tabelas, visando a consistência entre os dados.



Disponível em: <https://tinyurl.com/4rw9477v>. Acesso em 13/05/2024.

Banco de Dados e SGBDs III - SQL e NoSQL

- **Definições:**
 - **SQL (*Structured Query Language*):** linguagem declarativa utilizada para realizar consultas em dados estruturados;
 - **NoSQL (*Not Only SQL*):** bancos de dados voltados para o gerenciamento de dados semi-estruturados e desestruturados.
- **Estrutura:** bancos de dados SQL são relacionais, com esquema e relacionamentos definidos. Bancos de dados NoSQL são mais flexíveis e podem ser baseados em diferentes formatos, como documentos, colunas, chave-valor ou grafos;
- **Desempenho:** bancos de dados SQL são otimizados para consultas complexas e transações com alta integridade de dados, enquanto os bancos de dados NoSQL são otimizados para leituras e escritas rápidas de grandes volumes de dados;
- **Escalabilidade:** bancos de dados SQL são projetados para escalar verticalmente, adicionando mais recursos de *hardware* a um único servidor. Bancos de dados NoSQL são projetados para escalar horizontalmente, adicionando mais nós de servidor a um *cluster*.

Próxima Aula - *Hands-on*

- Na próxima aula, será criado do zero em um ambiente Linux, uma estrutura baseada no banco relacional **PostgreSQL**, utilizando **Docker**. Em suma:
 - Será instalado e configurado o **Docker**, uma ferramenta para virtualização de ambientes de desenvolvimento;
 - Em seguida, serão criados dois contêineres em *Docker*: (i) instalação e configuração do banco de dados; (ii) instalação e configuração do SGBD **pgAdmin**;
 - Será criado uma base de dados contendo tabelas e relações a partir de *scripts* SQL;
 - Serão inseridos dados *dummy* nessa base, simulando uma aplicação real;
 - Serão apresentados os principais comandos SQL, como *select*, *insert*, *update* e *delete*.



Laboratório de Fundamentos em TIC

Introdução às Arquiteturas de *Software* e Dados

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/fundamentos-em-tic>