**FATEC ZONA LESTE**

**AMS ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Danilo Costa Rodrigues**

**Gabriel da Silva Mendes**

**Henrique da Silva Machado**

**João Vitor de Sousa Rodrigues**

**Lucas Pereira Carvalho**

**Paulo Enrick Santos Silva**

**Pesquisa: API’s e procedimentos de banco de dados**

**São Paulo**

**2024**

Sumário

[1 API’s 3](#_Toc164974718)

[1.1 Introdução 3](#_Toc164974719)

[1.2 Buscando dados do servidor usando a *Fetch* API 3](#_Toc164974720)

[1.3 Envio de formulários via JavaScript 4](#_Toc164974721)

[*1.4* *Client-side storage* 4](#_Toc164974722)

[2 Procedimentos em banco de dados 5](#_Toc164974723)

[Referencias 3](#_Toc164974724)

# API’s

## Introdução

“Uma API é um conjunto de regras e protocolos que permite a comunicação entre diferentes softwares.”. Uma API é vista como uma interface de comunicação entre dois sistemas. Por exemplo, um sistema de e-commerce não cria todo um sistema de pagamento do zero, ele se comunica com um sistema pronto através de uma API. Graças às API’s, não é necessário saber todos os detalhes da implementação de um *software*, apenas as regras para usá-lo corretamente.

A API pode estar integrada com uma linguagem, como as API’s do JavaScript, que são usadas no lado do cliente (DOM, *Fetch* etc.), e podem ser API’s de terceiros, que buscam dados em um servidor para retornar para o cliente (API do Twitter, Twitch, Github etc.).

Figura 1 - Funcionamento de uma API

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor.

## Buscando dados do servidor usando a *Fetch* API

Como dito anteriormente, *Fetch* é uma API do JavaScript, ou seja, ela á integrada à linguagem. Ela é usada para a utilização de API’s de terceiros, ou seja, ela vai fazer a requisição para uma API, que vai pegar os dados no servidor de um sistema, e vai retornar os dados solicitados.

“A API *Fetch* fornece uma interface JavaScript para acessar e manipular partes do pipeline HTTP, tais como os pedidos e respostas. Ela também fornece o método global *fetch* que fornece uma maneira fácil e lógica para buscar recursos de forma assíncrona através da rede.”.

Para um exemplo usando a função *fetch* para buscar dados de API’s de terceiros, vemos criar um *script* que retorna dados de usuários do GitHub.

Figura 2 - Requisição *Fetch*

Texto

Descrição gerada automaticamente

Fonte: do próprio autor.

A função *fetch* funciona da seguinte maneira:

* Ela recebe o endereço da API como parâmetro (linha 48).
* Caso tenha sucesso, os dados são retornados em forma de uma *promise*, então é necessário converter os dados para um objeto JavaScript através da função *response.json()* (linha 49).
* Os dados retornados são impressos no console (linha 50).
* Caso haja algum erro no servidor do GitHub, ele será impresso no console (linha 51).

Como a função *fetch* retorna uma *promise* (promessa), é necessário tratar tanto o sucesso quanto o erro da requisição feita à API do GitHub.

Basicamente, usamos a função *fetch* para fazer o nosso sistema se comunicar com o sistema do GitHub, sem a necessidade de implementar um novo sistema de versionamento de código, e essa é a principal vantagem da utilização de API’s.

## Envio de formulários via JavaScript

## *Client-side storage*

# Procedimentos em banco de dados

# Referencias

LUCAS, Humberto. Uma visão de um iniciante sobre APIs. [S. l.], 12 jul. 2023. Disponível em: https://www.dio.me/articles/uma-visao-de-um-iniciante-sobre-apis. Acesso em: 25 abr. 2024.

MDN CONTRIBUTORS (EUA). Usando Fetch. [S. l.], 3 ago. 2023. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch\_API/Using\_Fetch. Acesso em: 25 abr. 2024.