Assincronicidade

Algo que não ocorre ou não se efetiva ao mesmo tempo

**JavaScript** por padrão roda de forma **síncrona**, acontece alguma coisa e quando ele termina acontece outro processo

Promises

Objeto de processamento assíncrono

Inicialmente, seu valor é desconhecido. Ela pode, então, ser resolvido ou rejeitado

A promises possui 3 estados

Pending - Pendente

Fulfilled - Completada

Rejectd – Rejeitada

Estrtura

Const myPromise = new Promises((resolve, reject)) => {

Windows.setTimeout(( ) => {

Resolve(console.log(‘Resolvido!’));

}, 2000);

});

Manipulação

Await myPromise

.then

O que tiver dentro do await e .then é que será feito com aquilo depois que for resolvido

Async/await

Funções assíncronas precisam dessas duas palavras chave

Funções assíncronas também retornam Promises.

Consumindo APIs

Application Programming Interface

Uma API é uma forma de intermediar, os resultados do back-end com o que é apresentado no front-end. Você consegue acessá-la por meio de URLs

JSON: JavaScript Object Notation

É muito comum que APIs retornem seus dados no formato .json, portanto precisamos tratar esses dados quando os recebemos.

Fetch

Consumindo APIs

Fetch(url, options)

.then(response => response.json( ))

.then(json => console.log(json))

// fetch retorna uma Promise

Operações no banco (POST, GET, PUT, DELETE, etc)