



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER


AULA 17 – REVISÃO

O QUE IREMOS APRENDER

- 01** TRY CATCH
- 02** CONSUMO DE API
- 03** ASYNC AWAIT
- 04** PROMISES
- 05** PROJETO

TRY CATCH


O bloco try-catch é uma estrutura em linguagens de programação que permite capturar e lidar com exceções (erros) que podem ocorrer durante a execução de um programa. O bloco "try" é usado para envolver o código que pode gerar uma exceção, enquanto o bloco "catch" é usado para especificar como lidar com essa exceção se ela ocorrer.



```
1  try{
2      // código a ser realizado que pode gerar uma exceção.
3      const resultado = 10 / 0
4  } catch (erro) {
5      // Aqui lidaremos com a exceção.
6      console.log("Ocorreu um erro: ", erro.message)
7  }
```

CONSUMO DE API

O consumo de API envolve fazer solicitações a um servidor remoto para obter ou enviar dados. Isso é comum em aplicativos da web e aplicativos móveis, onde os dados são obtidos de serviços externos por meio de APIs (Interfaces de Programação de Aplicativos).



```
1 fetch('https://api.exemple.com/data')
2   .then(response => response.json())
3   .then(data => console.log(data))
4   .catch(error => console.log("Erro ao buscar os dados: ", error));
```

ASYNC AWAIT

Async-Await é uma maneira de lidar com código assíncrono de forma mais síncrona e legível em linguagens como JavaScript. Async é usada para definir uma função assíncrona, e await é usada dentro dessa função para esperar que uma operação assíncrona seja concluída.



```
1  async function buscarDadosDaAPI() {  
2      try {  
3          const response = await fetch('https://api.exemple.com/data');  
4          const data = await response.json();  
5          console.log(data);  
6      } catch (error) {  
7          console.error("Erro ao buscar os dados: ", error);  
8      }  
9  }  
10  
11  buscarDadosDaAPI()
```


PROMISES

Promises são objetos em JavaScript que representam a conclusão (ou falha) de uma operação assíncrona. Elas são usadas com `async-await` para lidar com código assíncrono de forma mais estruturada.

```
1 function obterDadosDaAPI() {
2     return new Promise((resolve, reject) => {
3         fetch('https://api.exemple.com/data')
4             .then(response => response.json())
5             .then(data => resolve(data))
6             .catch(error => reject(error));
7     });
8 }
9
10 async function buscarDados() {
11     try {
12         const data = await obterDadosDaAPI();
13         console.log(data);
14     } catch (error){
15         console.error("Erro ao buscar dados: ", error);
16     }
17 }
18
19 buscarDados();
```

PROJETO

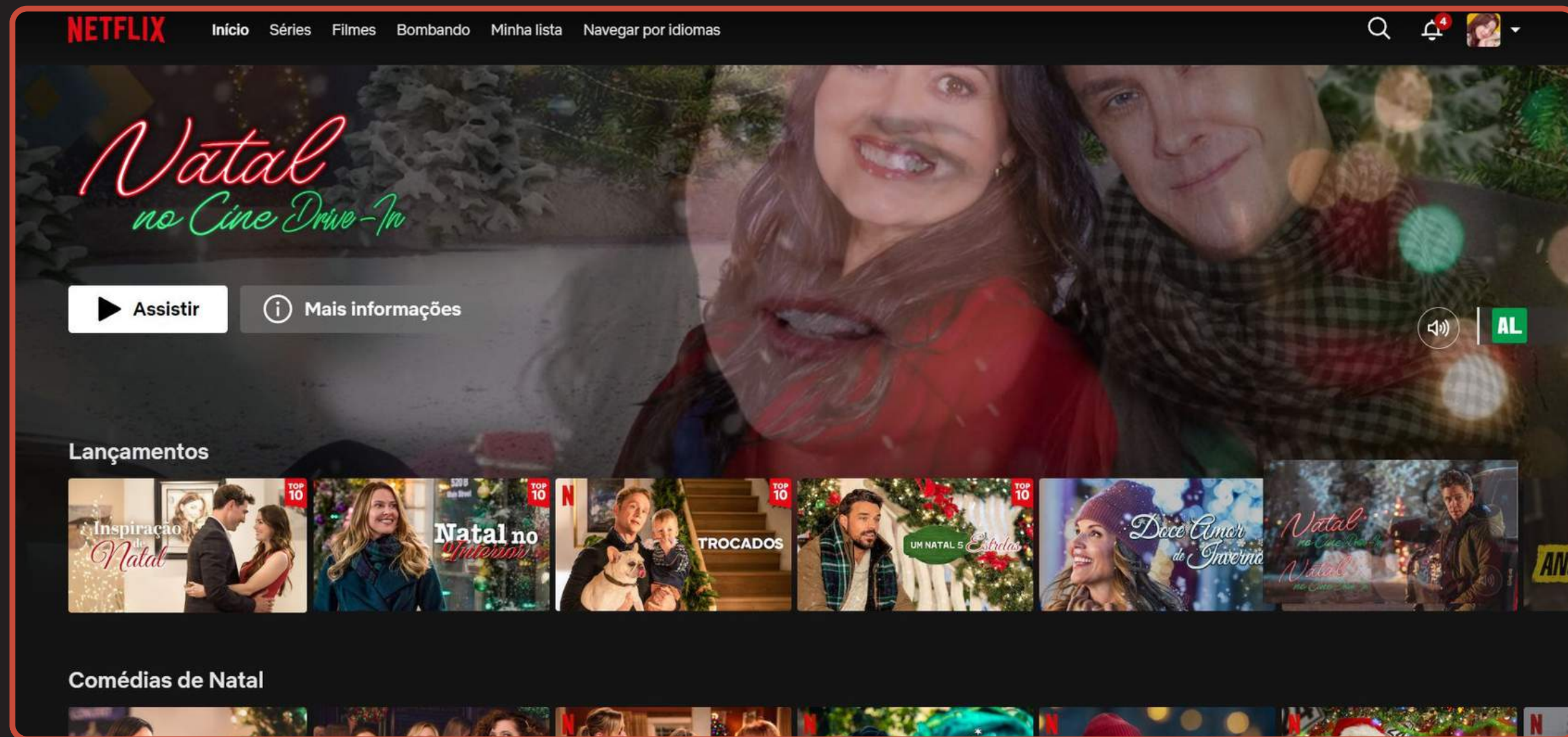
Projeto Netflix

Crie uma plataforma de streaming de vídeo semelhante à Netflix que utiliza uma API externa para obter informações sobre filmes e séries. Os usuários podem navegar por um catálogo de filmes e séries, e ver detalhes de cada título.

Link da Api: <https://www.themoviedb.org/documentation/api>



PROJETO



ATÉ A PRÓXIMA AULA!

AULA 18 DE JAVASCRIPT.
PROJETO



IN

INFINITY SCHOOL
VISUAL ART CREATIVE CENTER



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 17 – REVISÃO