**APIs**

- É um acrônimo para Interface de Programação de Aplicações (Application Programming Interface), não tem telas propriamente ditas, o front-ed está sendo separado do back-end, é esperado a resposta da interface com base no que solicitamos;

- Uma forma simples de comunicação entre aplicações; não necessariamente o software se conecta a apenas uma API, pode ser conectado a várias;

- Não necessita de telas, respostas geralmente são em JSON;

Ex.: É enviado um texto que contém um usuário (email, nome e senha) e API dá a resposta do usuário cadastrado com sucesso.

- Independente do front-end, a API não possui ligação com o front (isso é uma regra, seguindo a padronização da API em qualquer ambiente);

Ex.: - quero adicionar produto

- Api responde com a cor do botão(btn-color:red)

- Isso não faria sentido **em uma loja**

- A **api só trataria de adicionar o produto**

- Baseada em requisições e resposta;

- Pode ser criada uma API com o Express.

**O que é REST e RESTful**

- **REST** significa Representational State Transfer ou Transferência Representacional de Estado, quando transferimos o estado da nossa aplicação, seja da requisição para a API ou vice-versa, trabalhando só com informações;

**REST** - Padrão que se aplica a uma API

- Estilo de arquitetura que define como as APIs devem ser criadas;

- **Seguindo todos** esses **padrões** a **API é** considerada uma **RESTful**;

- Alguns dos pontos são: respostas uniformes(mensagem de erros e de sucesso padronizadas), sem ligação de estado(front e back sem ligação), cache de respostas e outros.

**SPA**

- Significa Single Page Application;

- Quando a aplicação possui um **front-end separado do back-end**;

- Ou seja, uma **API para back-end**;

- E um framework **front-end JS**, como: React, Vue e Angular;

- Aplicações deste tipo estão dominando o mercado;

- As seções seguintes tem como objetivo trabalhar focadas em SPA, consumir api no front-end;

- A arquitetura anteriormente utilizada é a conhecida como Monolith (front, back-end e banco de dados).

**Verbos HTTP**

- Andam de mãos dadas com as APIs e **fazem parte do REST**;

- Temos métodos como: **GET**(resgatando informação), **POST**(postar, enviar dados), **PUT**(atualização completa de um registro (todos os dados)), **DELETE**(remove, deleta dados), **PATCH**(atualização parcial dos dados) e etc;

- Cada um representa uma ação na API, ou seja, são essenciais;

- É muito importante que as nossas funcionalidades usem métodos corretos, os métodos essenciais para aquela função;

- Enviamos o método através da requisição.

**Criando uma API com Express**

- Para isso, teremos uma tarefa comparada a estrutura com **Handlebars**;

- Basta **instalar o Express** e ele fará tudo sozinho neste ponto;

- Depois será criada uma **rota** **que responde em JSON**, este é o **dado de comunicação** entre aplicação e API;

- É importante a definição do verbo correto, com **GET** ou **POST** por exemplo.