

# FullStack

## Exercícios da Semana

### [M1S9] Ex 1 – Projeto *Trindade Places*

#### Contexto:

O projeto ***Trindade Places*** é uma iniciativa importante da prefeitura de Trindade para fornecer informações precisas e atualizadas sobre as instituições que prestam serviços públicos na cidade. Através da API desenvolvida em Node.js, os usuários poderão pesquisar, criar, atualizar e excluir informações sobre essas instituições, como nome do lugar e localização.

A escolha das tecnologias *express* e *sequelize* para o desenvolvimento do projeto permite uma implementação eficiente e escalável da API, enquanto a integração com o banco de dados garante que os dados sejam armazenados de forma segura e facilmente acessíveis para consulta e atualização.

O principal objetivo do projeto é fornecer uma base de dados confiável para a cidade de Trindade, visando melhorar a gestão dos serviços públicos e aumentar a transparência das informações sobre as instituições públicas.

Com a API desenvolvida, a prefeitura poderá fornecer aos cidadãos uma fonte confiável e atualizada de informações sobre os serviços públicos disponíveis na cidade, contribuindo para uma melhor qualidade de vida para a população.

#### Projeto:

Para iniciar um novo projeto Node.js + Express e contribuir para o desenvolvimento da API do projeto *Trindade Places*, siga os seguintes passos:

- Certifique-se de ter o Node.js instalado em sua máquina.
- Abra um terminal e crie uma nova pasta para o seu projeto.
- Navegue até a pasta criada e inicialize o npm com o comando `"npm init"`.
- Instale o Express utilizando o comando `"npm install express"`.
- Crie um arquivo chamado `"index.js"` na raiz do seu projeto.
- Abra o arquivo `"index.js"` e importe o Express com o comando `"const express = require('express')"`.
- Inicie o servidor Express e defina a porta 3333

## [M1S9] Ex 2 – Modelo *Sequelize*

Após criar a estrutura inicial, é necessário criar um novo modelo *Sequelize* chamado **Place** para representar um local. Para isso, é importante definir corretamente seus campos para que possa ser utilizado adequadamente. Os campos que devem ser definidos são:

- **id**: um número que serve como identificador único para o local.
- **name**: uma *string* que representa o nome do local.
- **contact**: uma *string* que representa o número de telefone do local.
- **opening hours**: uma *string* que representa o horário de funcionamento do local.
- **description**: uma *string* que representa a descrição do local.
- **latitude**: um número que representa a latitude do local.
- **longitude**: um número que representa a longitude do local

## [M1S9] Ex 3 – Rota POST

Após criar o modelo **Place** no exercício anterior, é necessário implementar uma rota do tipo **POST** com o *path* `"/places"` que receba valores por meio de BODY *params*. Os seguintes valores devem ser recebidos:

```
{
  "name": "Nome do Local",
  "telefone": "(00) 0000-0000",
  "opening hours": "Segunda a Sexta: 9h às 18h",
  "description": "Descrição do Local",
  "latitude": -23.5505,
  "longitude": -46.6333
}
```

## [M1S9] Ex 4 – Rota GET

Aprimore o projeto *Trindade Places* implementando uma rota **GET** com o *path* `"/places"`, que retorne a lista de lugares cadastrados na base de dados.

Essa rota permitirá aos usuários consultar todas as instituições públicas disponíveis na cidade de Trindade, fornecendo informações relevantes sobre cada uma delas.

Essa implementação é fundamental para a transparência e eficiência na gestão dos serviços públicos, permitindo aos cidadãos acessar informações sobre as instituições de forma fácil e rápida. Com isso, será possível melhorar a qualidade dos serviços públicos na cidade.

## [M1S9] Ex 5 – Rota DELETE

Para melhorar ainda mais a API do projeto *Trindade Places*, é necessário implementar uma rota DELETE. Essa rota deve permitir aos usuários excluir uma instituição pública específica da base de dados. Para isso, é preciso definir um identificador único para cada instituição, que será utilizado na requisição de exclusão.

A rota **DELETE** pode ser implementada com o *path* `"/places/:id"`, onde o parâmetro `:id` representa o identificador único da instituição. Ao receber uma requisição **DELETE** com esse *path*, a API deve verificar se o identificador corresponde a uma instituição válida na base de dados e, em caso positivo, excluí-la.

A implementação da rota **DELETE** é essencial para permitir que os usuários tenham controle sobre as informações presentes na base de dados, garantindo a integridade e a atualização dos dados. Além disso, essa rota é importante para manter a transparência e a eficiência na gestão dos serviços públicos, permitindo que as instituições sejam atualizadas ou excluídas quando necessário.

## [M1S9] Ex 6 – Rota PUT

A rota **PUT** é outra funcionalidade importante que deve ser implementada na API do projeto *Trindade Places*. Essa rota permite que os usuários atualizem as informações de uma instituição pública específica na base de dados.

Para implementar a rota **PUT**, é preciso definir um identificador único para cada instituição, que será utilizado na requisição de atualização. Além disso, é necessário definir quais campos da instituição poderão ser atualizados pelos usuários.

A rota **PUT** pode ser implementada com o *path* `"/places/:id"`, onde o parâmetro `:id` representa o identificador único da instituição. Ao receber uma requisição **PUT** com esse *path*, a API deve verificar se o identificador corresponde a uma instituição válida na base de dados e, em caso positivo, permitir a atualização dos campos definidos.

A rota permitirá atualizar todos os campos cadastrados, exceto o ID.