

pulsati<sup>+</sup>

---

express





# O que é o express

Express é um framework web para Node.js que fornece um conjunto de recursos para construir aplicativos da web.

Ele é simples e flexível, permitindo criar aplicativos da web de maneira rápida e fácil, com uma sintaxe concisa e intuitiva.

O Express é projetado para lidar com rotas, solicitações e respostas HTTP, manipulação de sessões, integração com vários mecanismos de modelo, entre outras funcionalidades.



# Como eu inicio?

Vamos iniciar do início, instalando a biblioteca  
`$ npm install express`





# npm install??

O NPM (Node Package Manager) é um gerenciador de pacotes para o Node.js. Ele permite instalar, gerenciar e compartilhar pacotes ou módulos de software Node.js em um projeto. Esses pacotes contêm bibliotecas de código pré-escritas que podem ser reutilizadas em seus próprios projetos Node.js.

Acesse: <https://www.npmjs.com/>





# Como inicio meu servidor?

```
const express = require('express'); // importo a biblioteca do express

const app = express(); // crio uma instância de um servidor server

app.use(express.json()); // configuro para o servidor trabalhe somente com JSON

// na linha abaixo, estou adicionando o tratamento para leitura de uma chamada GET
// para a rota 'hello-world' e retornando como resultado a mensagem que executou com sucesso
app.get('/hello-world', async (req, res) => {
  res.json({ data: 'Executado com sucesso' });
});

app.listen(8000); // configuro para que o servidor utilize a porta 8000 da minha máquina

// pronto!! Com essas linha eu iniciei um servidor em express
```



# Prática

Crie um servidor express que:

- Adicione um usuário
- Leia um usuário
- Atualize um usuário
- Delete um usuário

0/  
Exercícios  
+



# Criar um aplicação de registro de passageiros

## Atividade 1:

- Crie um servidor com express que receba uma requisição POST para a rota /aeroportos com o nome do aeroporto
- Você deve criar uma classe Aeroporto que tem os campos código e nome
- No tratamento da requisição POST, você deve criar uma instância da classe Aeroporto (o campo código deve ser criado pela própria implementação no servidor) e o objeto Aeroporto deve ser salvo em uma lista

## Atividade 2:

- Utilizando o mesmo servidor criado na Atividade 1
- Adicione o tratamento para uma requisição GET para a rota /aeroportos, como resposta para esta requisição, deve retornar um JSON com todos os aeroportos salvos até o momento

## Atividade 3:

- Utilizando o mesmo servidor criado na Atividade 1
- Adicione o tratamento para uma requisição PUT para a rota /aeroportos/:codigo, atualizando o nome do Aeroporto
- Nessa requisição PUT, você irá receber o código do Aeroporto que deve ser atualizado

## Atividade 4:

- Utilizando o mesmo servidor criado na Atividade 1
- Adicione o tratamento para uma requisição DELETE para a rota /aeroportos/:codigo, onde deve deletar o Aeroporto de acordo com o código recebido como parâmetro





# Criar um aplicação de registro de passageiros

Atividade 5:

- Utilizando o mesmo servidor criado na atividade 1
- De acordo com as atividades anteriores, adicione tratamento para atender as requisições POST/GET/PUT/DELETE para a rota /empresas-aereas

# Obrigado!

**pulsati<sup>+</sup>**

---

[pulsati.com.br](https://pulsati.com.br)