**TUTORIAL: COMO INSTALAR A BIBLIOTECA REACT NO LINUX**

**Passo 1: Instalar o Node.js e o npm**

1. Abra um terminal no seu sistema Linux.
2. Para instalar o Node.js, você pode usar o gerenciador de pacotes do seu sistema. Se você estiver usando o Ubuntu ou o Debian, você pode usar o seguinte comando:

**sudo apt update**

**sudo apt install nodejs**

Se você estiver usando o Fedora, você pode usar o seguinte comando:

**sudo dnf install nodejs**

Este comando instalará tanto o Node.js quanto o npm (Node Package Manager).

1. Verifique se o Node.js e o npm foram instalados corretamente executando os seguintes comandos no terminal:

**node -v**

**npm -v**

Isso mostrará as versões instaladas do Node.js e do npm, respectivamente.

**Passo 2: Instalar o Create React App**

O Create React App é uma ferramenta oficial para criar aplicativos React facilmente. Para instalá-lo, execute o seguinte comando no terminal:

**npm install -g create-react-app**

Isso instalará o Create React App globalmente no seu sistema.

**INSTALANDO O POSTGRE SQL**

**Passo 1: Atualize o Sistema**

Antes de começar, é importante garantir que seu sistema esteja atualizado. Abra o terminal e execute o seguinte comando:

**sudo apt update**

**Passo 2: Instale o PostgreSQL**

Agora, instale o pacote do PostgreSQL junto com o pacote -contrib, que adiciona algumas utilidades e funcionalidades extras:

**sudo apt install postgresql postgresql-contrib**

Quando solicitado, pressione **Y** para confirmar a instalação.

**Passo 3: Inicie o PostgreSQL**

Inicie o serviço do PostgreSQL com o seguinte comando:

**sudo systemctl start postgresql**

Habilite o PostgreSQL na inicialização: Para garantir que o PostgreSQL seja iniciado automaticamente sempre que o sistema for reiniciado, execute o seguinte comando:

**sudo systemctl enable postgresql**

**Passo 4: Configure o PostgreSQL**

O PostgreSQL armazena suas configurações no arquivo postgresql.conf. Você pode editá-lo usando qualquer editor de texto, como nano ou vim:

**sudo nano /etc/postgresql/16/main/postgresql.conf**

No arquivo postgresql.conf, defina listen\_addresses como **\*** para permitir conexões remotas:

**listen\_addresses = '\*'**

Configure a autenticação de senha: Abra o arquivo pg\_hba.conf para configurar a autenticação de senha:

**sudo nano /etc/postgresql/16/main/pg\_hba.conf**

Substitua as linhas relevantes para usar autenticação MD5:

**host all all 0.0.0.0/0 md5**

**local all all trust**

Reinicie o PostgreSQL para aplicar as alterações:

**sudo systemctl restart postgresql**

Permita a porta do PostgreSQL no firewall: Se você estiver usando um firewall, permita a porta 5432 (padrão do PostgreSQL):

**sudo ufw allow 5432/tcp**

**Passo 5: Conecte-se ao Servidor PostgreSQL**

Primeiro, conecte-se ao servidor PostgreSQL usando o usuário postgres:

**sudo -u postgres psql**

Defina uma senha para o usuário postgres: Execute o seguinte comando dentro do psql para definir uma senha:

ALTER USER postgres PASSWORD '<sua\_senha>';

Substitua **<sua\_senha>** pela senha desejada.

Saia do psql: Digite **\q** e pressione Enter para sair do psql.

**O que é Fast Api?**

FastAPI é um framework de desenvolvimento web para construir APIs (Interfaces de Programação de Aplicações) em Python de forma rápida, fácil e eficiente. Ele foi projetado para ser fácil de usar, enquanto ainda oferece alto desempenho, graças à sua implementação assíncrona baseada em Python 3.7 e posterior.

O FastAPI utiliza a tipagem de dados do Python através do sistema de annotations (anotações) para definir os tipos de dados das requisições e respostas, o que facilita a validação automática e a geração automática de documentação interativa (Swagger UI e ReDoc). Além disso, ele suporta o uso de padrões modernos como o OpenAPI.

Uma das características notáveis do FastAPI é a sua capacidade de lidar com código assíncrono de forma eficiente, permitindo a construção de APIs altamente concorrentes e escaláveis. Com uma sintaxe declarativa e intuitiva, o FastAPI torna possível criar APIs de maneira rápida e segura, ao mesmo tempo em que aproveita ao máximo as funcionalidades mais recentes do Python.

Em resumo, o FastAPI é uma escolha poderosa para desenvolvedores que buscam criar APIs de forma eficiente, com suporte a documentação automática, validação de tipos e desempenho otimizado.

**INSTALANDO O PYTHON**

Antes de começar a trabalhar com o FastAPI, é necessário garantir que o Python esteja instalado no ambiente de desenvolvimento. O FastAPI é compatível com Python 3.7 e versões posteriores. Aqui estão os passos básicos para instalar o Python:

Faça o download do Python:

Visite o site oficial do Python em python.org e vá para a seção de downloads. Escolha a versão mais recente estável para o seu sistema operacional (Windows, macOS ou Linux) e faça o download do instalador.

Instale o Python:

Execute o instalador baixado e siga as instruções do assistente de instalação. Certifique-se de marcar a opção "Adicionar Python ao PATH" durante a instalação para facilitar o acesso ao interpretador Python a partir do terminal ou prompt de comando.

Verifique a instalação:

Abra um terminal ou prompt de comando e digite o seguinte comando para verificar se o Python foi instalado corretamente no cmd:

**python --version**

Isso deve exibir a versão do Python que você instalou.

Com o Python devidamente instalado, você estará pronto para instalar o FastAPI e começar a desenvolver suas APIs. Você pode usar o gerenciador de pacotes pip, que geralmente é instalado automaticamente com o Python. Para instalar o FastAPI.

1. Acesse o site e localize a seção de downloads no seguinte link:

https://fastapi.tiangolo.com/

1. Crie um projeto em Python e digite a seguinte linha de instalação no prompt:

**pip install fastapi**

1. Após concluir esse passo, copie o comando a seguir:

**pip install "uvicorn[standard]"**