# Passo a Passo - Projeto Next.js + JSON-Server (CRUD Usuários)

## Introdução

Este material traz o passo a passo para a criação de um sistema completo de cadastro, edição, desativação e exclusão de usuários,  
utilizando Next.js com TypeScript e uma API simulada usando JSON-Server.

## Detalhamento de cada etapa

### 1. Criação do projeto Next.js

Comando:  
npx create-next-app@latest app-json-server --typescript  
  
√ Would you like to use ESLint? ... Yes

√ Would you like to use Tailwind CSS? ... No

√ Would you like your code inside a `src/` directory? ... No

√ Would you like to use App Router? (recommended) ... Yes

√ Would you like to use Turbopack for `next dev`? ... Yes

√ Would you like to customize the import alias (`@/\*` by default)? ... No

### 2. Instalação do Axios

Comando:  
 npm install axios

### 3. Criação do serviço API

Arquivo: services/api.tsx  
  
import axios from "axios";  
const api = axios.create({ baseURL: "http://localhost:3001" });  
export { api };

### 4. Configuração e inicialização do JSON-Server

Comando para instalar globalmente (se necessário):  
npm install -g json-server  
  
Arquivo db.json na raiz:  
  
{  
 "login": [],  
 "funcionarios": []  
}  
  
Comando para iniciar o servidor JSON:  
json-server --watch db.json --port 3001

### 5. Criação da tela de listagem de usuários

Arquivo: app/login/page.tsx

Código resumido que carrega e lista usuários, permitindo clicar para edição.

### 6. Criação da tela de edição, desativação e exclusão de usuário

Arquivo: app/login/[id]/page.tsx

Tela que permite editar dados, desativar ou excluir usuários existentes.

## Considerações Finais

O projeto segue boas práticas do Next.js mais atual, utilizando 'use client' e 'useParams' para tratar corretamente a navegação.  
Este passo a passo ensina a construir uma base sólida para projetos mais robustos de CRUD no futuro.