# | \EXT\_Js 14

DESENVOLVIMENTO TURBO

ANDER SILVA

Next.js 14 traz uma revolução no desenvolvimento de aplicativos web. Aqui, exploraremos as funcionalidades chave que permitem criar páginas robustas e modernas, acompanhadas de exemplos práticos de códigos.

# RENDERIZAÇÃO HÍBRIDA FLEXÍVEL

## RENDERIZAÇÃO HÍBRIDA FLEXÍVEL

No Next.js, você não precisa escolher entre Server-Side Rendering (SSR) e Static Site Generation (SSG). Você pode usar ambos no mesmo projeto!

Exemplo de Código:

Esse é o poder da flexibilidade: escolha a melhor estratégia para cada página, otimizando a performance e a experiência do usuário.

# CARREGAMENTO DE IMAGENS OTIMIZADO

### CARREGAMENTO DE IMAGENS OTIMIZADO

O Next.js oferece um componente Image que otimiza automaticamente suas imagens.

Exemplo de Código:

Com o componente Image, suas páginas carregam mais rápido, consumindo menos dados.

## API ROUTES

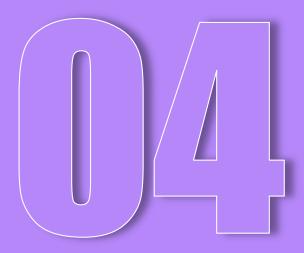
### **API ROUTES**

Crie APIs diretamente dentro do seu projeto Next.js. Isso simplifica o processo de construção de back-ends para suas aplicações web.

Exemplo de Código:

```
// Página API em pages/api/user.js
export default function handler(req, res) {
  res.status(200).json({ nome: 'Anderson' })
}
```

Aqui está a conveniência: um endpoint API pronto com apenas algumas linhas de código.



## ROTEAMENTO DINÂMICO

### **ROTEAMENTO DINÂMICO**

Com o Next.js, criar rotas dinâmicas é um processo simples e intuitivo.

Exemplo de Código:

```
import { useRouter } from 'next/router'

const Post = () => {
  const router = useRouter()
  const { pid } = router.query

  return Post: {pid}
}

export default Post
```

Rotas dinâmicas permitem que você crie páginas personalizadas rapidamente, baseadas nos parâmetros da URL.

## MIDDLEWARE

### **MIDDLEWARE**

Next.js 14 introduziu um novo conceito de Middleware, permitindo que você execute códigos antes de uma resposta ser completada.

#### Exemplo de Código:

```
export function middleware(request) {
  const { pathname } = request.nextUrl

  // Redirectiona se não estiver logado
  if (pathname == '/dashboard') {
    return new Response(null, { status: 401, headers: { 'Location': '/login' } })
  }
}
```

O Middleware abre um leque de possibilidades para segurança e manipulação de requisições, tudo isso na borda da rede.

## **CONCLUSÃO: NEXT.JS 14 E VOCÊ**

As funcionalidades do Next.js 14 permitem construir páginas web não apenas modernas e robustas, mas também otimizadas para o melhor desempenho. Experimente esses exemplos em seu próprio projeto e veja a diferença que o Next.js pode fazer.

Cada capítulo deste eBook foi feito para guiá-lo pelo mundo do Next.js, transformando ideias em realidade digital. O próximo passo? Mãos no código e começar a criar!



### **SOBRE O AUTOR**



**ANDER SILVA**FULL STACK DEVELOPER

Formado em Engenharia da Computação, mergulhei de cabeça no mundo do desenvolvimento web e descobri no Next.js 14 a linguagem da eficiência moderna.

Através deste eBook, compartilho com vocês não só o poder do Next.js, mas a filosofia por trás de um código limpo e uma arquitetura pensada. Para mim, programar é mais do que uma profissão, é uma forma de moldar o futuro, uma linha de código de cada vez.