

Aula 08

1. Criação do projeto next frontend api xadrez
2. Definição do layout
3. Implementação do GET
4. Implementação do POST

1. Criação do projeto next frontend api xadrez

Primeiro, crie um novo projeto Next.js executando o seguinte comando no terminal:

```
npx create-next-app@latest frontend-xadrez
```

Navegue até o diretório do projeto com o comando `cd frontend-xadrez`

2. Definição do layout

Altere o arquivo `src/app/layout.tsx` para ficar parecido com o seguinte trecho de código:

```
import './globals.css';

export const metadata = {
  title: 'Xadrez - Gerenciamento de Jogadores',
  description: 'App para gerenciar jogadores de xadrez',
};

export default function RootLayout({
  children,
}: {
  children: React.ReactNode;
}) {
  return (
    <html lang="en">
      <body className="bg-gray-100 text-gray-900">{children}</body>
    </html>
  );
}
```

Isso vai gerar um layout padronizado para todas as páginas da aplicação.

3. Implementação do GET

Vá ao arquivo `src/app/page.tsx` e altere para ficar parecido com o seguinte código:

```
'use client';
import { useState, useEffect } from 'react';
```

```

interface Jogador {
  id: number;
  nome: string;
}

export default function Home() {
  const [jogadores, setJogadores] = useState<Jogador[]>([]);

  const fetchJogadores = async () => {
    const res = await fetch('http://localhost:3000/jogadores');
    const data = await res.json();
    setJogadores(data);
  };

  useEffect(() => {
    fetchJogadores();
  }, []);

  return (
    <main className="p-6">
      <h1 className="text-2xl font-bold mb-4">Jogadores de Xadrez</h1>

      <ul className="list-disc pl-5">
        {jogadores.map((jogador) => (
          <li key={jogador.id} className="mb-2">
            {jogador.nome}
          </li>
        ))}
      </ul>
    </main>
  );
}

```

Esse código irá gerar uma lista de jogadores (que estão cadastrados na api que deve estar rodando na porta 3000). Dessa forma já estamos conseguindo executar uma chamada com o método GET para nosso backend e também recuperar os valores retornados.

4. Implementação do POST

Altere o código do arquivo `src/app/page.tsx` para ficar semelhante ao seguinte:

```

'use client';

import { useState, useEffect } from 'react';

```

```

interface Jogador {
  id: number;
  nome: string;
}

export default function Home() {
  const [jogadores, setJogadores] = useState<Jogador[]>([]);
  const [nome, setNome] = useState('');

  useEffect(() => {
    fetchJogadores();
  }, []);

  const fetchJogadores = async () => {
    const res = await fetch('http://localhost:3000/jogadores');
    const data = await res.json();
    setJogadores(data);
  };

  const addJogador = async () => {
    if (!nome) return;

    const res = await fetch('http://localhost:3000/jogador', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ nome }),
    });

    if (res.ok) {
      fetchJogadores();
      setNome('');
    }
  };

  return (
    <main className="p-6">
      <h1 className="text-2xl font-bold mb-4">Jogadores de Xadrez</h1>

      <div className="mb-6">
        <input
          type="text"
          value={nome}
          onChange={(e) => setNome(e.target.value)}
          className="border p-2 mr-2"
        />
      </div>
    </main>
  );
}

```

```

        placeholder="Nome do jogador"
      />
      <button
        onClick={addJogador}
        className="bg-blue-500 text-white px-4 py-2 rounded"
      >
        Adicionar
      </button>
    </div>

    <ul className="list-disc pl-5">
      {jogadores.map((jogador) => (
        <li key={jogador.id} className="mb-2">
          {jogador.nome}
        </li>
      ))}
    </ul>
  </main>
);
}

```

Nesse trecho foram adicionadas as tags HTML para criar um input e um botão para podermos digitar o nome do jogador que será inserido.

Ao clicar no botão será executada a função **addJogadores** que irá processar uma requisição com o método POST para o backend, e logo após resincronizar os dados do GET. De forma que teremos a inserção do novo jogador em nosso backend e ainda a atualização do get no frontend.