

Após ter configurado o ambiente e criado o projeto com base no [tutorial](#). Abra o projeto no VSCode e siga os passos a seguir para continuar o desenvolvimento:

1. Primeiro vamos rodar a nossa aplicação, digite no terminal:

```
npm start
```

- Acesse: <http://localhost:3000/> e veja como a aplicação está agora.
- O que você está vendo é a tela padrão de um projeto React, ela está definida em **src/App.js**

2. Crie um arquivo para gerenciar as chamadas à API:

- Crie o arquivo **src/services/api.js**
- **axios**: Utilizado para fazer requisições HTTP.
- O arquivo ficará parecido com isso:

```
import axios from 'axios';

// Configurar a URL base da sua API
const api = axios.create({
  baseURL: 'https://angular-api.xwhost.com.br/api/', // URL real da sua API
  headers: {
    'Content-Type': 'application/json',
  },
});

export default api;
```

3. Agora vamos criar um arquivo para gerenciar as operações de clientes:

- Crie o arquivo **src/services/ClienteService.js**
- O arquivo ficará parecido com isso:

```
import api from './api';

// Função para buscar todos os clientes
export const getClientes = async () => {
  try {
    const response = await api.get('/clientes');
    return response.data;
  } catch (error) {
    console.error('Erro ao buscar clientes:', error);
    throw error;
  }
};

// Função para criar um cliente
export const createCliente = async (cliente) => {
  try {
    const response = await api.post('/clientes/', cliente);
    return response.data;
  } catch (error) {
    console.error('Erro ao criar cliente:', error);
    throw error;
  }
}
```

```

};

// Função para atualizar um cliente
export const updateCliente = async (id, clienteAtualizado) => {
  try {
    const response = await api.put(`/clientes/${id}`, clienteAtualizado);
    return response.data;
  } catch (error) {
    console.error('Erro ao atualizar cliente:', error);
    throw error;
  }
};

// Função para deletar um cliente
export const deleteCliente = async (id) => {
  try {
    await api.delete(`/clientes/${id}`);
  } catch (error) {
    console.error('Erro ao deletar cliente:', error);
    throw error;
  }
};

```

4. Agora, vamos criar o componente para listar os clientes:

- Crie o arquivo **src/components/ListaClientes.js**
- O arquivo deverá ficar parecido com isso:

```

import React, { useEffect, useState } from 'react';
import { getClientes, deleteCliente } from '../services/ClienteService';

const ListaClientes = ({ onEdit }) => {
  const [clientes, setClientes] = useState([]);

  useEffect(() => {
    const fetchClientes = async () => {
      try {
        const data = await getClientes();
        setClientes(data);
      } catch (error) {
        console.error('Erro ao carregar clientes:', error);
      }
    };

    fetchClientes();
  }, []);

  const handleDelete = async (id) => {
    try {
      await deleteCliente(id);
      setClientes(clientes.filter((cliente) => cliente.id !== id));
    } catch (error) {
      console.error('Erro ao deletar cliente:', error);
    }
  };
};

```

```

return (
  <div>
    <h2>Clientes</h2>
    <table>
      <thead>
        <tr>
          <th>Nome</th>
          <th>CPF</th>
          <th>Email</th>
          <th>Ações</th>
        </tr>
      </thead>
      <tbody>
        {clientes.map((cliente) => (
          <tr key={cliente.id}>
            <td>{cliente.nome}</td>
            <td>{cliente.cpf}</td>
            <td>{cliente.email}</td>
            <td>
              <button onClick={() => onEdit(cliente)}>Editar</button>
              <button onClick={() =>
handleDelete(cliente.id)}>Excluir</button>
            </td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
);
};

export default ListaClientes;

```

5. Agora, vamos criar o componente de formulário que permitirá adicionar e editar clientes:

- Crie o arquivo **src/components/FormCliente.js**
- O arquivo deverá ficar parecido com isso:

```

import React, { useState, useEffect } from 'react';
import { createCliente, updateCliente } from '../services/ClienteService';

const FormCliente = ({ clienteAtual, onSave }) => {
  const [cliente, setCliente] = useState({
    nome: '',
    cpf: '',
    email: '',
    observacoes: '',
    ativo: true,
  });

  useEffect(() => {
    if (clienteAtual) {
      setCliente(clienteAtual);
    }
  }, [clienteAtual]);

```

```

const handleChange = (e) => {
  const { name, value } = e.target;
  setCliente({ ...cliente, [name]: value });
};

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    if (cliente.id) {
      await updateCliente(cliente.id, cliente);
    } else {
      await createCliente(cliente);
    }
    onSave();
  } catch (error) {
    console.error('Erro ao salvar cliente:', error);
  }
};

return (
  <form onSubmit={handleSubmit}>
    <label>
      Nome:
      <input type="text" name="nome" value={cliente.nome}
onChange={handleChange} />
    </label>
    <label>
      CPF:
      <input type="text" name="cpf" value={cliente.cpf}
onChange={handleChange} />
    </label>
    <label>
      Email:
      <input type="email" name="email" value={cliente.email}
onChange={handleChange} />
    </label>
    <label>
      Observações:
      <textarea name="observacoes" value={cliente.observacoes}
onChange={handleChange}></textarea>
    </label>
    <label>
      Ativo:
      <input type="checkbox" name="ativo" checked={cliente.ativo}
onChange={() => setCliente({ ...cliente, ativo: !cliente.ativo })} />
    </label>
    <button type="submit">Salvar</button>
  </form>
);
};

export default FormCliente;

```

6. Agora, vamos criar a página que permitirá a visualização e edição de clientes:

- Crie o arquivo **src/pages/Clientes.js**
- O arquivo deverá ficar parecido com isso:

```

import React, { useState } from 'react';
import ListaClientes from '../components/ListaClientes';
import FormCliente from '../components/FormCliente';

const Clientes = () => {
  const [clienteEdit, setClienteEdit] = useState(null);

  const handleEdit = (cliente) => {
    setClienteEdit(cliente);
  };

  const handleSave = () => {
    setClienteEdit(null);
  };

  return (
    <div>
      <ListaClientes onEdit={handleEdit} />
      <FormCliente clienteAtual={clienteEdit} onSave={handleSave} />
    </div>
  );
};

export default Clientes;

```

7. Adicione a navegação entre clientes e contas:

- Vá para o arquivo **src/App.js**
- O arquivo deverá ficar parecido com isso:

```

import './App.css';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import Clientes from './pages/Clientes';

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Clientes />} />
        <Route path="/clientes" element={<Clientes />} />
      </Routes>
    </Router>
  )
}

export default App;

```