

# Skyforge Coding

Conquiste bibliotecas  
em projetos React.js



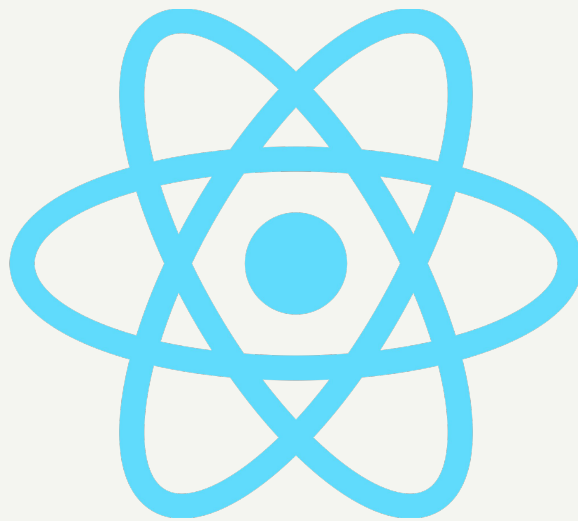
**Eric Santana**



# React.js melhorado

## Um Guia Abrangente para Bibliotecas Essenciais

Este ebook é seu guia para se tornar proficiente no desenvolvimento com React.js, com foco na utilização de bibliotecas essenciais como Vite, Immer, React Hook Form, Material UI e React Router. Cada capítulo fornecerá exemplos práticos e insights para ajudá-lo a compreender como essas bibliotecas aprimoram seus projetos React.js.





Capítulo 1

# Vite: acelere seu desenvolvimento



# Vite: acelere seu desenvolvimento

## Build Tool da Próxima Geração

Vite é uma ferramenta de construção inovadora, projetada para agilizar seu processo de desenvolvimento com React.js. Essa ferramenta aproveita as importações de módulos ES nativos para proporcionar uma experiência de desenvolvimento rápida.





# Vite: acelere seu desenvolvimento

O Vite elimina a necessidade de empacotamento durante o desenvolvimento, servindo seu código diretamente de um servidor de desenvolvimento otimizado. Vamos começar e ver como integrá-lo em seu projeto.

```
npm init @vitejs/app my-react-app --template react
cd my-react-app
npm install
npm run dev
```





Capítulo 2

# Domine a gestão de estado com Immer



# Domine a gestão de estado com Immer

## Simplifique a gestão de estado

Com o poder do Immer, a mutação de objetos não precisa mais ser feita ao longo de arrays.

```
const [state, setState] = React.useState([])
setState(produce(draft => {
  draft.push({ text: 'forge with Immer' })
}))
```

O Immer permite que você trabalhe com objetos imutáveis de uma maneira que parece mutável, eliminando a necessidade de criar novos objetos ao atualizar um estado.







# Domine a gestão de estado com Immer

## Comparação

Para comparar a forma antiga e a forma com o Immer de atualizar o estado, vamos supor que queremos revisar a quantidade de ferro em estoque:

```
Component.jsx

const [rawMetals, setRawMetals] =
  React.useState({
    iron: {
      Quantity: 10
    }
  })
```







# Domine a gestão de estado com Immer

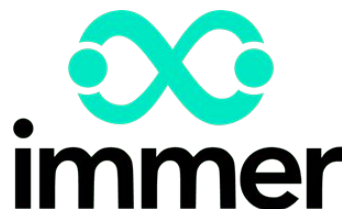
## Comparação

Sem o Immer:

```
const updatedRawMetalsWithoutImmer = {  
  ...rawMetals,  
  iron: {  
    ...rawMetals.iron,  
    Quantity: rawMetals.iron.Quantity + 5  
  }  
};  
setRawMetals(updatedRawMetalsWithoutImmer);
```

Com o Immer:

```
const updatedRawMetalsWithImmer =  
produce(rawMetals, draft => {  
  draft.iron.Quantity += 5;  
});  
setRawMetals(updatedRawMetalsWithImmer);
```





Capítulo 3

# Desbloqueie Caminhos com React Router

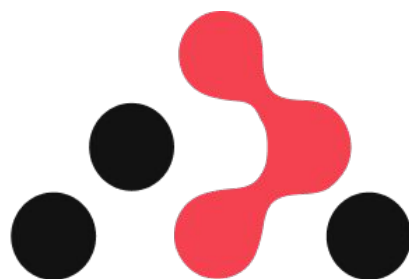


# Desbloqueie Caminhos com React Router

## Navegação sem Esforço

O React Router permite que você navegue entre diferentes partes do seu aplicativo sem recarregar a página.

Ele simplifica o roteamento em aplicativos React.js, permitindo uma navegação intuitiva e dinâmica entre diferentes componentes.



# React Router





# Desbloqueie Caminhos com React Router

## Navegação sem Esforço

Algumas vantagens de usar o React Router no seu projeto incluem:

- Suporte para Aplicações de Página Única (SPA).
- Definição de rotas de forma declarativa usando JSX.
- Roteamento Aninhado: permite criar estruturas de página complexas com vários níveis.
- Roteamento Dinâmico: possibilita a renderização dinâmica de rotas.





# Desbloqueie Caminhos com React Router

## Navegação sem Esforço

Aqui estão exemplos que demonstram a implementação do React Router no seu código.

```
Navigation.jsx

import React from 'react';
import {
  BrowserRouter as Router,
  Route,
  Link }
from 'react-router-dom';

const Home = () => <h1>Home</h1>;
const About = () => <h1>About</h1>;

const App = () => {
  return (
    <Router>
      <nav>
        <Link to="/">Home</Link>
        <Link to="/about">About</Link>
      </nav>
      <Route path="/" exact component={Home} />
      <Route path="/about" component={About} />
    </Router>
  );
};
```





Capítulo 4

# Otimize Formulários com React Hook Form



# Otimize Formulários com React Hook Form

## Melhorando o Manejo de Formulários

No seu cerne, React Hook Form é uma biblioteca que aproveita o poder dos hooks do React para otimizar o manuseio de formulários.

Utilize o poder dos hooks para capturar dados de formulário sem esforço, validar entradas e lidar com a submissão de formulários com habilidade.







# Otimizar Formulários com React Hook Form

## Melhorando o Manejo de Formulários

Let's initialize the React hook form with the resources inside his useForm hook:

```
import React from 'react'
import { useForm } from 'react-hook-form';

function MyForm() {
  const {
    register,
    handleSubmit,
    formState: { errors },
    setValue,
    watch
  } = useForm();
  ...
}
```

The returned values of this hook will be used for basic use, validation and controlled components.





# Otimize Formulários com React Hook Form

## Uso Básico

Uso em um formulário Básico:

```
function MyForm() {  
  ...  
  const onSubmit = data => {  
    console.log(data);  
  };  
  return (  
    <form onSubmit={handleSubmit(onSubmit)}>  
      <input  
        {...register("firstName")}  
        placeholder="First Name" />  
      <input  
        {...register("lastName")}  
        placeholder="Last Name" />  
      <button type="submit">Submit</button>  
    </form>  
  );  
}  
  
export default MyForm;
```





# Otimize Formulários com React Hook Form Validação

Formulário com validação:

```
function MyForm() {  
  ...  
  const onSubmit = data => {  
    console.log(data);  
  };  
  return (  
    <form onSubmit={handleSubmit(onSubmit)}>  
      <input  
        {...register("email", { required: true })}  
        placeholder="Email" />  
        {errors.email &&  
        <span>Email is required</span>}  
  
      <input  
        {...register("password", {  
          required: true, minLength: 6  
        })}  
        type="password" placeholder="Password" />  
        {errors.password &&  
        <span>Password must be at least 6  
        characters long</span>}  
  
      <button type="submit">Submit</button>  
    </form>  
  );  
}  
  
export default MyForm;
```





# Otimize Formulários com React Hook Form

## Componentes controlados

Formulário com componentes controlados:

```
function MyForm() {  
  ...  
  const onSubmit = data => {console.log(data)};  
  return (  
    <form onSubmit={handleSubmit(onSubmit)}>  
      <input  
        {...register("email")}  
        placeholder="Email" />  
  
      <input  
        {...register("password")}  
        type="password"  
        placeholder="Password" />  
  
      <input {...register("confirmPassword")}  
        type="password" placeholder="Confirm Password"  
        onChange={e => {  
          setValue("confirmPassword", e.target.value);  
        }} />  
      {password !== watch("confirmPassword")  
        && <span>Passwords do not match</span>}  
  
      <button type="submit">Submit</button>  
    </form>  
  );  
}
```

export default MyForm;





Capítulo 5

# Criando Excelência: Material-UI para React



# Criando Excelência: Material-UI para React UI Aprimorada

O Material-UI simplifica o processo de desenvolvimento da interface do usuário, fornecendo componentes prontos para uso que seguem as diretrizes de design do Material Design. Isso garante consistência e aprimora a experiência do usuário em todas as aplicações.

Gender

☒ Female

☐ Male

TEXT

CONTAINED

OUTLINED

+

NAVIGATE

Age \*

Ten

None

Ten

Twenty

Thirty

ITEM ONE

ITEM TWO

Deletable

Deletable

4

Slider







# Criando Excelência: Material-UI para React UI Aprimorada

Vamos mergulhar em um exemplo de código para ilustrar como usar componentes do Material-UI em uma aplicação React.

```
import React from 'react';
import {
  Button,
  Typography
} from '@mui/material';

function App() {
  return (
    <div>
      <Typography variant="h1">
        Welcome to Material-UI
      </Typography>
      <Button variant="contained" color="primary">
        Get Started
      </Button>
    </div>
  );
}
export default App;
```







# Criando Excelência: Material-UI para React Tematização

Se você deseja alterar a aparência de um componente, você pode personalizá-lo de várias maneiras. No código abaixo, o Overriding Styles:

```
import React from 'react';
import { Button } from '@mui/material';

function CustomButton() {
  return (
    <Button
      sx={{ backgroundColor: 'red',
        borderRadius: '10px' }}>
      Click Me
    </Button>
  );
}

export default CustomButton;
```





# Criando Excelência: Material-UI para React UI Aprimorada

Lembre-se, dominar o Material-UI leva tempo e prática. Comece experimentando com projetos pequenos e gradualmente incorpore recursos mais avançados à medida que se sentir mais confortável. Com dedicação e perseverança, logo você se tornará proficiente em aproveitar o Material-UI para construir interfaces de usuário incríveis.





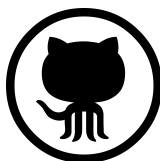
# Finalização



# Obrigado por ler até aqui.

Este livro foi produzido por um ser humano em colaboração com uma IA, com várias seções geradas por inteligência artificial e editadas e diagramadas por um ser humano. Este livro foi produzido para fins educacionais e leitura gratuita, não deve ser revendido.

## Conecte-se comigo



<https://github.com/efms25>



<https://www.linkedin.com/in/eric-santana-955812159/>



Skyforge Coding - Eric Santana