

Delphi + React

Por onde começar ?

{ Palestrante

Luiz Sfolia

Embarcadero Conference 2023



Objetivo

Quer um módulo do ERP na nuvem ?

Tem dúvidas se é possível fazer em Delphi ?

Tem dúvidas se pode usar React ?

Agenda

- Server

- Delphi ?
- Frameworks
- Http x Https
- Autenticação
- Distribuição

- Site

- React ?
- Criando Projeto
- Hooks
- Distribuição

Luiz Sfolia

CTO – TMR

Graduado em Gestão de TI

MBA Gerenciamento de Projetos



TMR

Server Frameworks

Horse

<https://github.com/HashLoad/horse>

- Framework Web
- Minimalista
- Curva de aprendizagem baixa
- Código no Git
- Vários Middlewares
- Free

Marshmallow

<https://bitbucket.org/sglienke/spring4d/src/master/Marshmallow/>

- Framework ORM
- Trabalha com Spring4D
- Free

Tms

<https://www.tmssoftware.com/site/xdata.asp>

- Framework Web e ORM
- Documentação ampla
- Interfaces compartilhadas entre Server e Client
- Possibilidade de integração com ORM
- Pago

- Sempre que possível, use injeção de dependências
- Estruture usando arquitetura
Model
Controller
- Sempre que possível use interfaces
- Tenha uma forma de mapear os endpoints (swagger)
- Defina os retornos dos endpoints otimizados
- Faça testes unitários

Banco de Dados Principal

- Atualização em conjunto com o Software Principal
- Distribuição facilitada
- Api será vinculada ao software principal

Banco de Dados Específico

- Atualização independente do software principal
- Possibilidade de mais de um BD para api's diferentes
- Menor risco de quebra de aplicação
- Precisa de sistema de atualização

Integrador de informações pelo Banco de dados

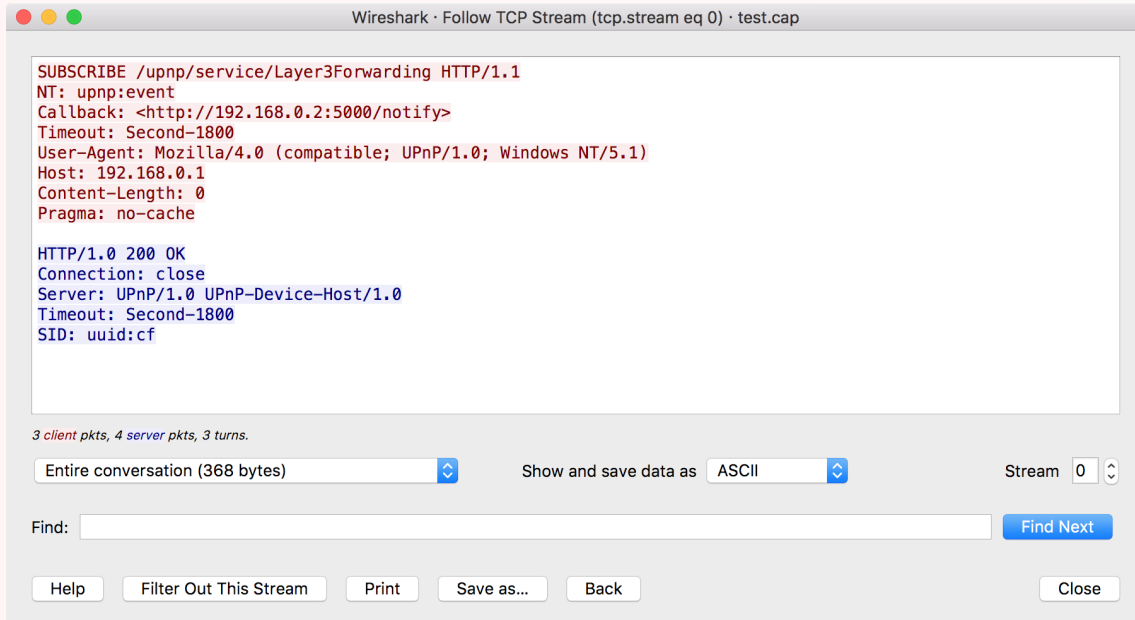
Integrador de informações por sistema integrador

Api's de integração de dados

Server

Http x Https

Porque Https ?



Certificado Digital no Server

SSL

TLS

Gerar certificado grátis



<https://letsencrypt.org/pt-br/>

Win-acme

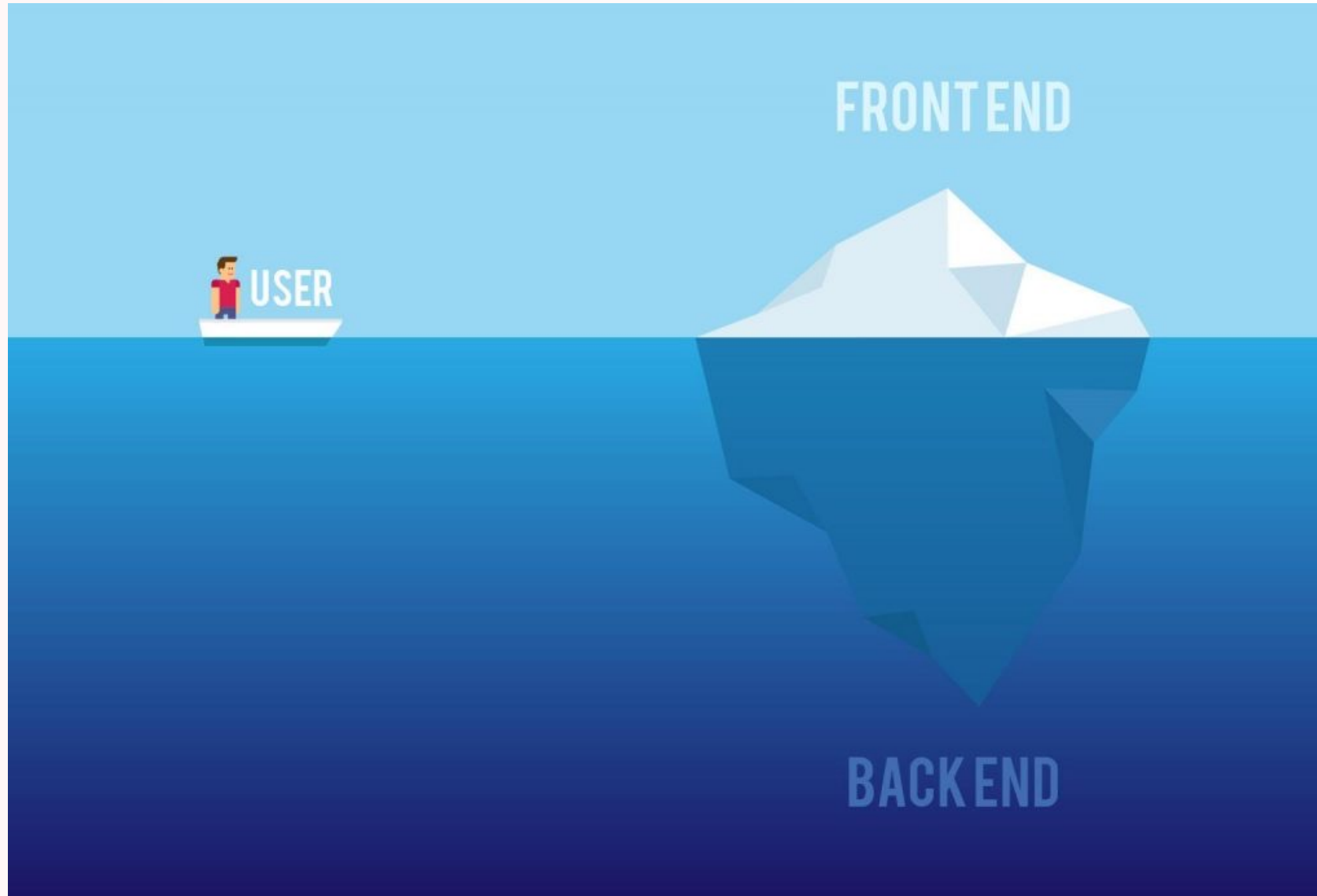
<https://github.com/win-acme/win-acme>

Server Distribuição

Como distribuir ?



Frontend



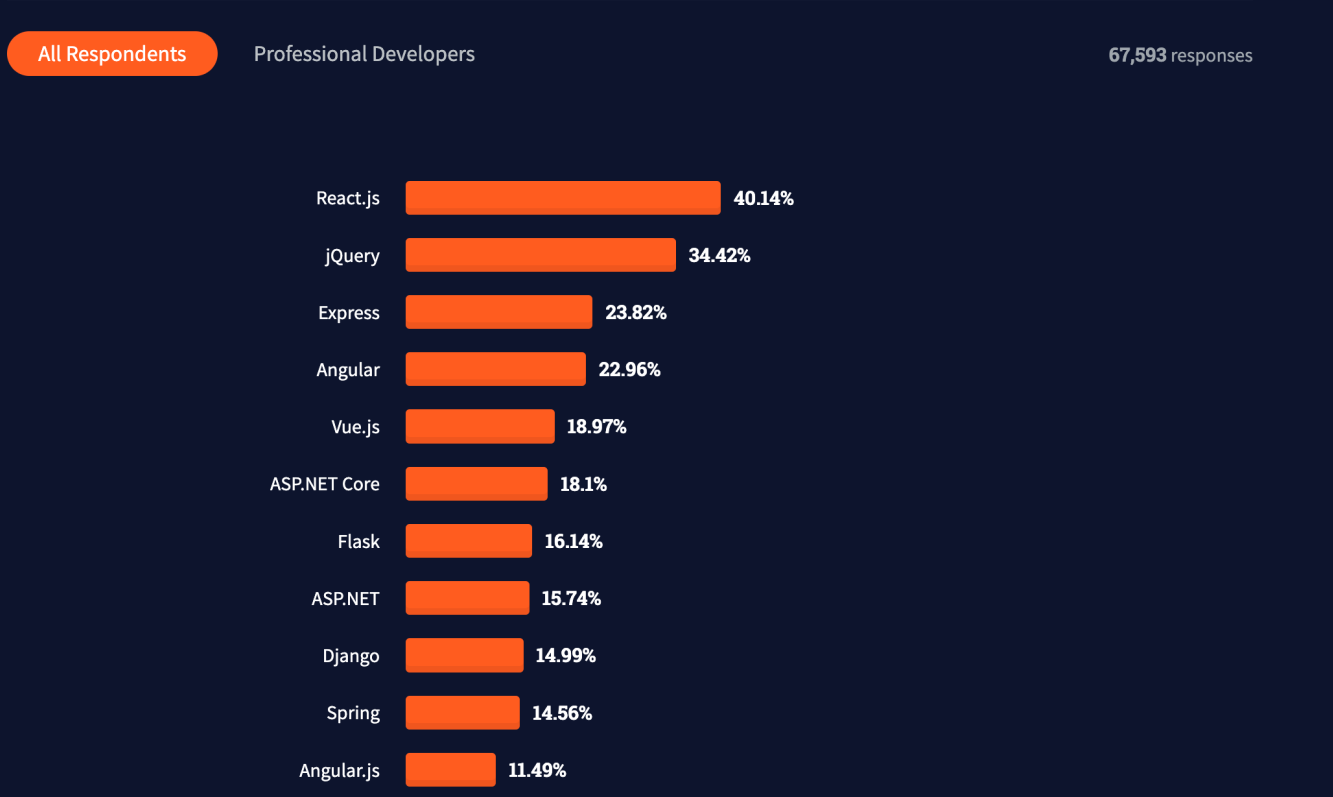
React x Angular



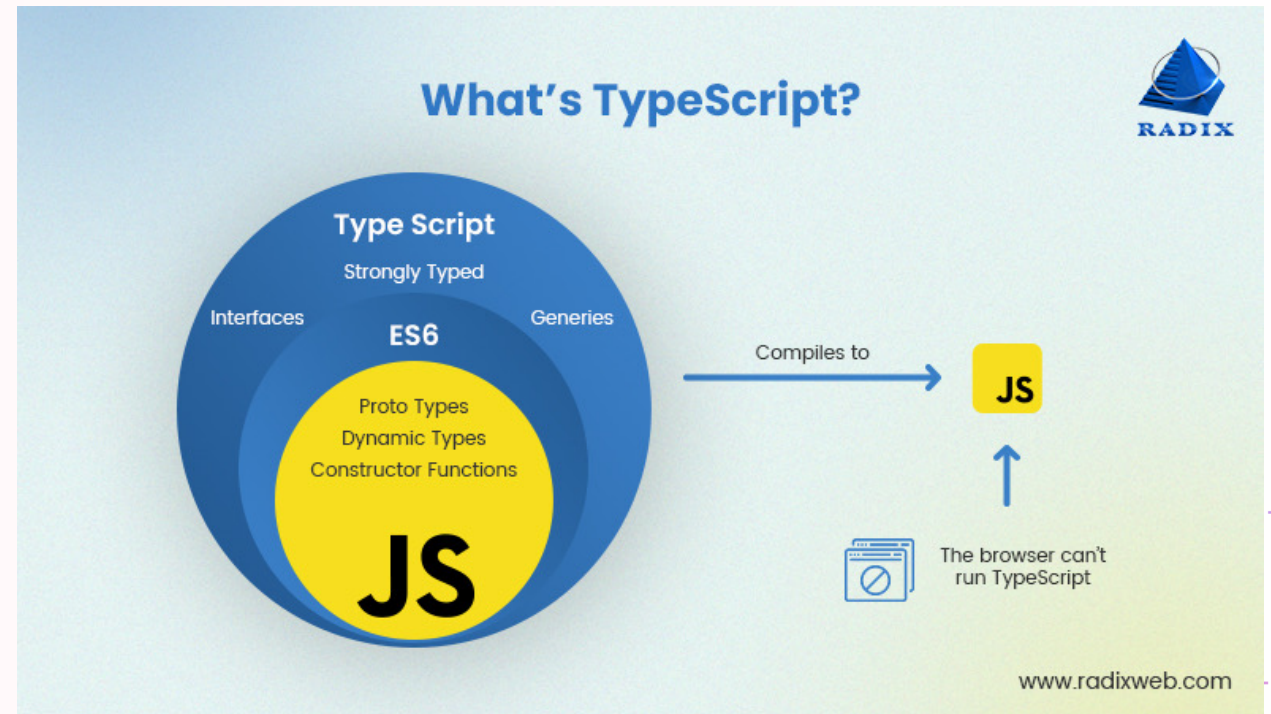
<https://insights.stackoverflow.com/survey/>

Web frameworks

This year, React.js surpassed jQuery as the most commonly used web framework.



JavaScript x TypeScript



Criando app React

<https://create-react-app.dev/docs/getting-started/>
`npm create-react-app {nome_projeto}`



<https://vitejs.dev/guide/>
`npm create vite@latest`



```
luizsfolia@MacBook-Pro-de-Luiz ~ % sudo npm create vite@latest
Password:
✓ Project name: ... teste_conf_2023_1
? Select a framework: > - Use arrow-keys. Return to submit.
  Vanilla
  Vue
  > React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```


Classes x Hooks

```
1 export function ShowCount(props) {  
2   const [count, setCount] = useState();  
3  
4   useEffect(() => {  
5     setCount(props.count);  
6   }, [props.count]);  
7  
8   return (  
9     <div>  
10      <h1> Count : {count} </h1>  
11    </div>  
12  );  
13 }
```

```
1 export class ShowCount extends React.Component {  
2   constructor(props) {  
3     super(props);  
4     this.state = {  
5       count: 0  
6     };  
7   }  
8   componentDidMount() {  
9     this.setState({  
10      count: this.props.count  
11    })  
12  }  
13  
14  render() {  
15    return (  
16      <div>  
17        <h1> Count : {this.state.count} </h1>  
18      </div>  
19    );  
20  }  
21  
22 }
```

useState

useEffect

useContext

useRef

<https://pt-br.legacy.reactjs.org/docs/hooks-overview.html>

Hooks

useState

```
import React, { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Hooks

useEffect

```
import React, { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  // Similar to componentDidMount and componentDidUpdate:
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Hooks

useContext

```
<MyContext.Consumer>  
  {value => /* render something based on the context value */}  
</MyContext.Consumer>
```

Hooks

useRef

```
function TextInputWithFocusButton() {  
  const inputEl = useRef(null);  
  const onClick = () => {  
    // `current` points to the mounted text input element  
    inputEl.current.focus();  
  };  
  return (  
    <>  
      <input ref={inputEl} type="text" />  
      <button onClick={onClick}>Focus the input</button>  
    </>  
  );  
}
```

Rotas

Clientes = Abre a listagem de Clientes

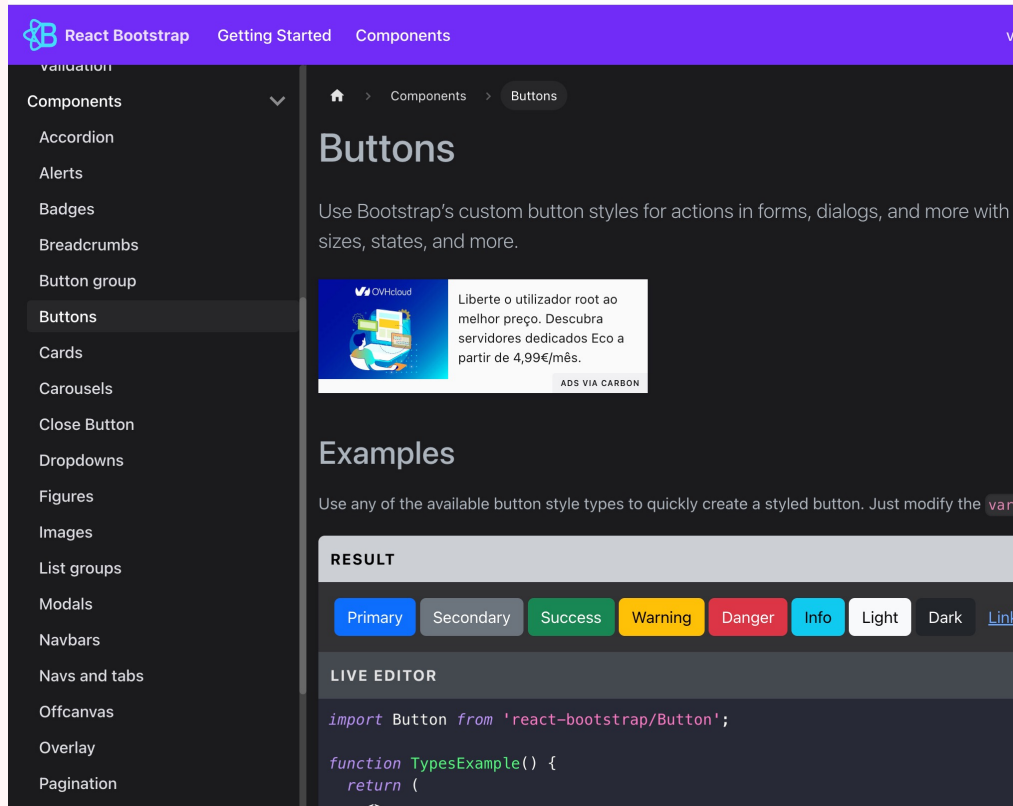
Clientes/1 = Visualiza Cliente 1

Clientes/editar/1 = Editar Cliente 1

```
<Routes>  
  <Route path="/clientes" element={<ListagemClientes/>} />  
  <Route path="/clientes/:id" element={<VisualizacaoCliente/>} />  
  <Route path="/clientes/editar/:id" element={<EdicaoCliente/>} />  
</Routes>
```


Estilos

<https://react-bootstrap.netlify.app/>

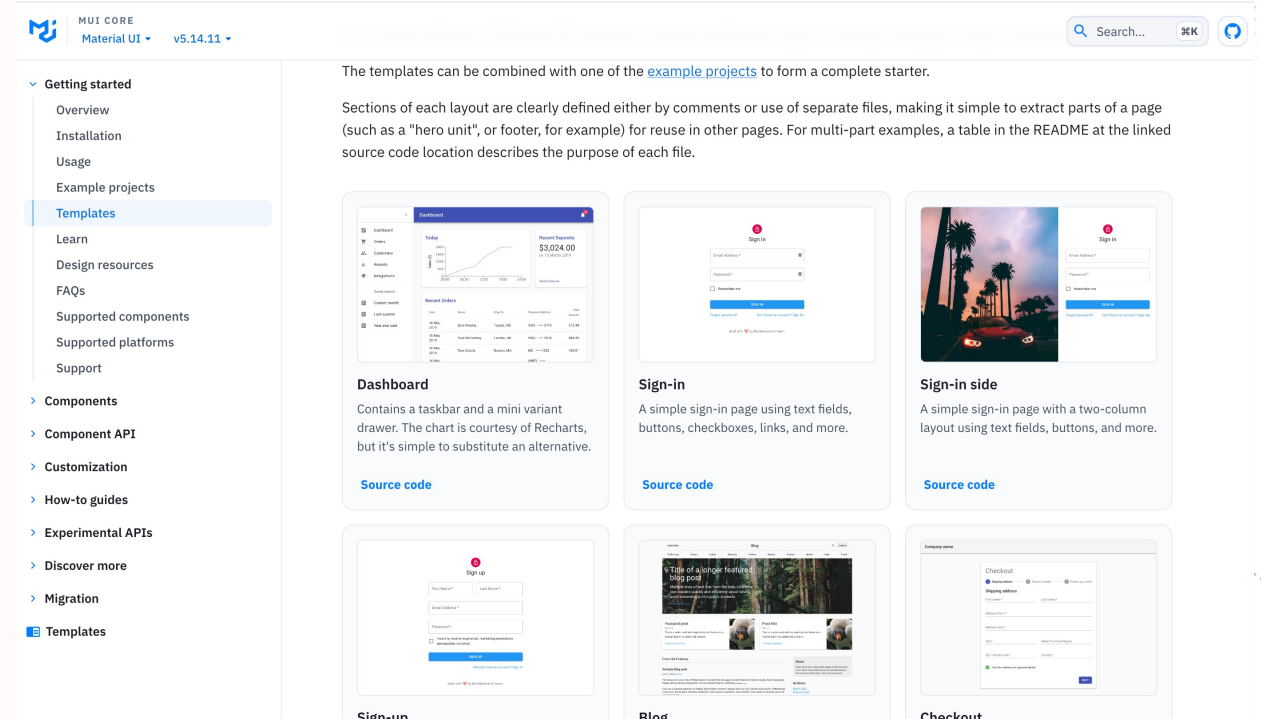


The screenshot shows the React Bootstrap website with a dark theme. The left sidebar lists various components, with 'Buttons' selected. The main content area is titled 'Buttons' and includes a description: 'Use Bootstrap's custom button styles for actions in forms, dialogs, and more with sizes, states, and more.' Below this is an advertisement for OVHcloud. Further down, there's a section titled 'Examples' with the text: 'Use any of the available button style types to quickly create a styled button. Just modify the var.' Below the examples is a 'RESULT' section showing a row of buttons: Primary, Secondary, Success, Warning, Danger, Info, Light, and Dark. At the bottom is a 'LIVE EDITOR' section with a code snippet:

```
import Button from 'react-bootstrap/Button';

function TypesExample() {
  return (
```

<https://mui.com/material-ui/>



The screenshot shows the Material UI website with a light theme. The left sidebar lists various sections, with 'Templates' selected. The main content area is titled 'Templates' and includes a description: 'The templates can be combined with one of the [example projects](#) to form a complete starter.' Below this is a paragraph: 'Sections of each layout are clearly defined either by comments or use of separate files, making it simple to extract parts of a page (such as a "hero unit", or footer, for example) for reuse in other pages. For multi-part examples, a table in the README at the linked source code location describes the purpose of each file.' Below the text are six preview cards for different templates: Dashboard, Sign-in, Sign-in side, Sign-up, Blog, and Checkout. Each card includes a preview image and a 'Source code' link.

Teste

Testes podem ser feitos fazendo leitura da página, esperando algum resultado

```
test('renders learn react link', () => {  
  render(<App />);  
  const linkElement = screen.getByText(/learn react/i);  
  expect(linkElement).toBeInTheDocument();  
});
```

Distribuição

1. Vercel
2. Firebase
3. Netlify
4. GitHub Pages
5. Heroku
6. Surge
7. Render
8. GitLab Pages

Netlify

- Deploy vinculado ao gitlab, github, etc..
- Configs por branch
- Vários projetos
- Variáveis de ambiente
- Versão free e paga
- Deploy automático

Dúvidas

Por onde começar

- Tela ?
- Banco de dados ?
- Api's ?

O que definir

- estrutura de endpoints
- estrutura de server
- Autenticação
- injeção de dependências

Decisões antes de começar

- Usar o mesmo banco de dados ?
- Deploy único ou em cada cliente ?
- Autenticação no mesmo serviço ?
- Quem vai filtrar, Server ou cliente ?
- Quem vai ordenar ?
- Quem vai paginar ?

O que você achou da palestra?

Acesse o link do QR Code ao lado e responda a pesquisa.



Obrigado

Luiz Sfolia

{ CTO TMR



Luiz.sfolia@tmrti.com.br

(51) 99822-2731



Embarcadero Conference 2023