



# Progressive Web App (PWA) e ReactJS



Fábio Soares



# Fábio Soares

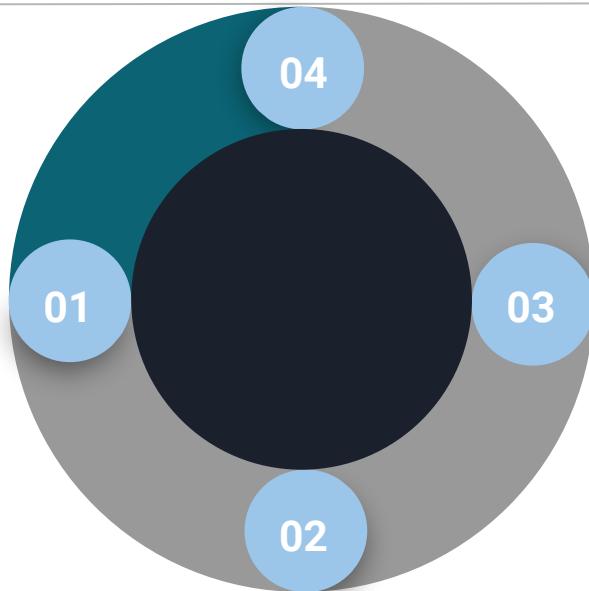
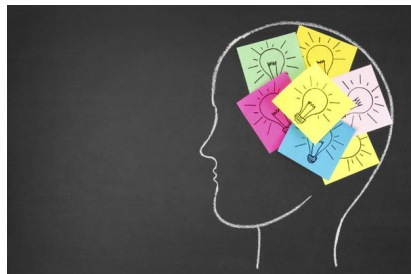
## Formação acadêmica/titulação

---



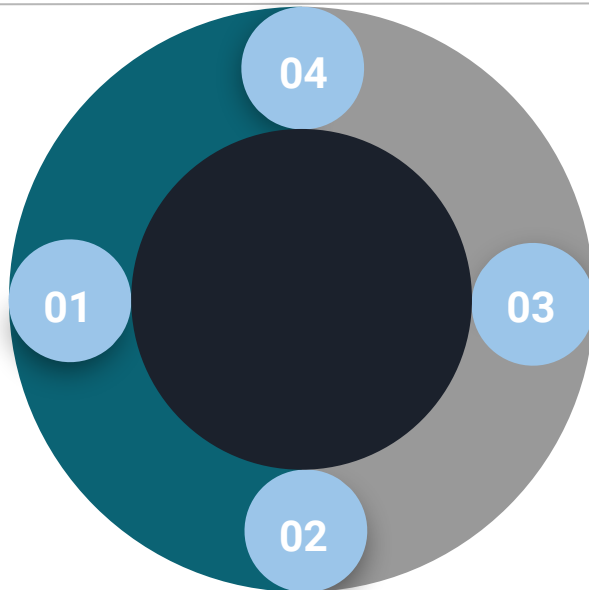
- 2014 - 2016** Mestrado em Ciências da Computação (Conceito CAPES 7).  
Universidade Federal de Pernambuco, UFPE, Brasil.  
Título: Um verificador de modelos em K para um subconjunto da linguagem Circus, Ano de Obtenção: 2016.  
Orientador: 🧐 Alexandre Mota.  
Coorientador: Adalberto Cajueiro.  
Bolsista do(a): Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco, FACEPE, Brasil.
- 2013 - 2013** Aperfeiçoamento em Análise de Testes (Residência). (Carga Horária: 552h).  
Universidade Federal de Pernambuco, UFPE, Brasil.  
Título: OracleKMS - A Tool for Knowledge Management. Ano de finalização: 2013.  
Orientador: Alexandre Mota.  
Bolsista do(a): Fundação de Apoio ao Desenvolvimento da Universidade Federal de Pernambuco, FADE, Brasil.
- 2009 - 2013** Graduação em Licenciatura em Computação.  
Universidade Federal Rural de Pernambuco, UFRPE, Brasil.

# Cronograma



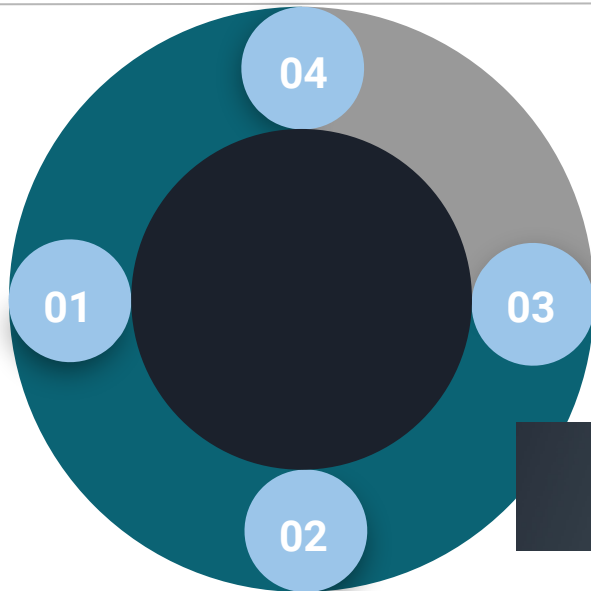


# Cronograma





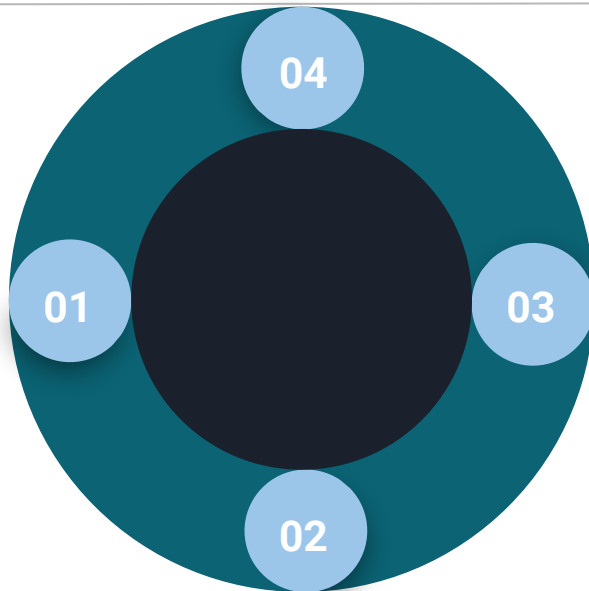
# Cronograma



 ReactJS



# Cronograma



## Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

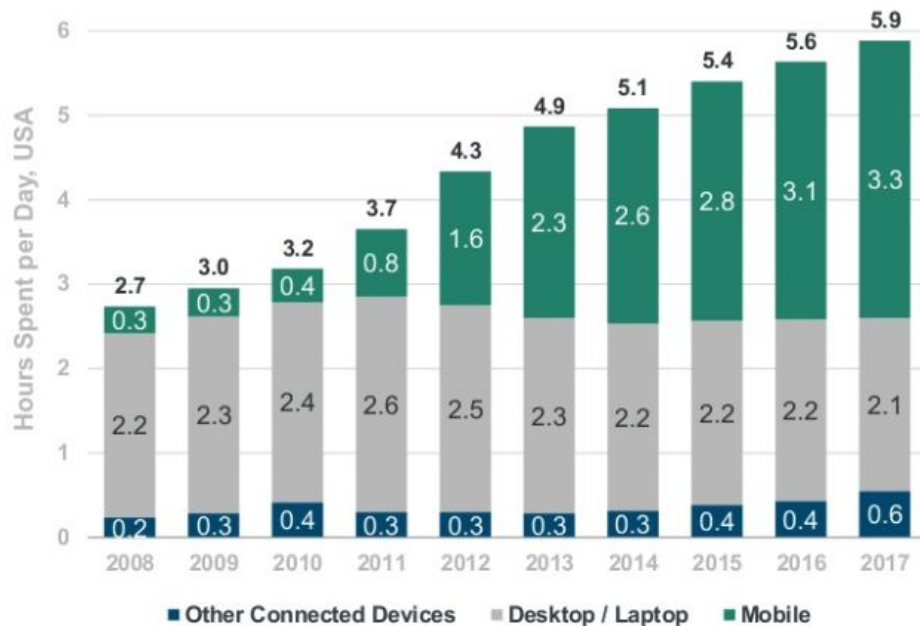
Partida encerrada!!!

Temos um vencedor.

RESETAR

## Digital Media Usage @ +4% Growth... 5.9 Hours per Day (Not Deduped)

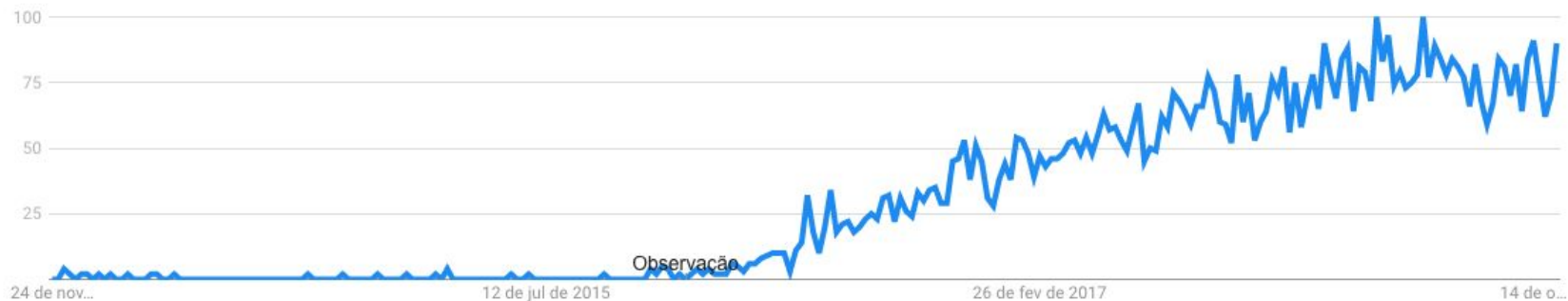
### Daily Hours Spent with Digital Media per Adult User





# PWA - Tendência

Interesse ao longo do tempo ?



Fonte: <https://trends.google.com/trends/explore?q=progressive%20web%20apps>





# PWA - Tendência

Assuntos relacionados ?

Em ascensão ▼



1 Progressive Web App - Assunto

Aumento  
repentino

2 React - JavaScript

Aumento  
repentino

3 Public Works Administration - Agência do go...

Aumento  
repentino

4 AngularJS - Assunto

Aumento  
repentino

5 Tutorial - Assunto

Aumento  
repentino

< Mostrando 1 a 5 de 25 assuntos >

Consultas relacionadas ?

Em ascensão ▼



1 pwa progressive web app

Aumento  
repentino

2 pwa

Aumento  
repentino

3 angular

Aumento  
repentino

4 angular progressive web app

Aumento  
repentino

5 progressive web app example

Aumento  
repentino

< Mostrando 1 a 5 de 25 consultas >

Fonte: <https://trends.google.com/trends/explore?q=progressive%20web%20apps>



## O que é? Quais os recursos?

---

Site Responsivo

App Nativo



Web App

PWA

---



# PWA - Recursos

Recursos	Website Responsivo	PWA	App Nativo
Usar quando estiver offline	Não	Sim	Sim
Precisa baixar em apps store	Não	Não	Sim
Implementa Design Responsivo	Sim	Sim	Sim
Indexável por mecanismos de busca	Sim	Sim	Não



# PWA - Recursos

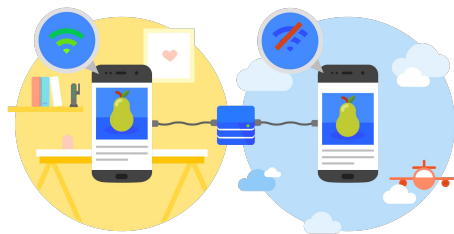
Recursos	Website	PWA	App Nativo
Notificações locais	Não	Sim	Sim
Notificações Push	Não	Sim	Sim
Precisa baixar para instalar	Não	Não	Sim
Atualizações rápidas e on-demand	Sim	Sim	Não



# PWA - Fundamentos

- **Confiável** - Carregue instantaneamente suas páginas/aplicações e nunca mostre o **downasaur**, mesmo quando estiver sem rede.
- **Rápido** - responda rapidamente às interações do usuário com animações suaves e sem rolagem.
- **Envolvente** - Sinta-se como se estivesse em um aplicativo nativo, com uma experiência de usuário imersiva.

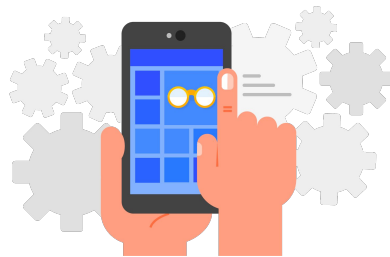
# PWA - Vantagens



- **Carregamento Progressivo** - Funciona para qualquer usuário, independentemente do navegador escolhido, pois é criado com aprimoramento progressivo (*Progressive Enhancement*) como princípio fundamental.
- **Responsivo** - Se adequa a qualquer formato: desktop, celular, tablet ou o que for inventado a seguir.
- **Independente de conectividade** - Aprimorado com service workers para trabalhar off-line ou em redes de baixa qualidade.



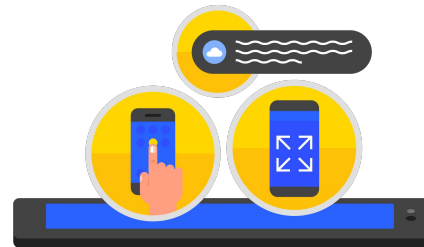
# PWA - Vantagens



- **Semelhante a aplicativos** - Parece com aplicativos para os usuários, com interações e navegação de estilo de aplicativos, pois é compilado no modelo de shell de aplicativo.
- **Atual** - Sempre atualizado graças ao processo de atualização do service worker.
- **Seguro** - Fornecido via HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.



# PWA - Vantagens



- **Descobrível** - Pode ser identificado como "aplicativo" graças aos manifestos W3C e ao escopo de registro do service worker, que permitem que os mecanismos de pesquisa os encontrem.
- **Reenvolvente** - Facilita o reengajamento com recursos como notificações push.
- **Instalável** - Permite que os usuários "guardem" os aplicativos mais úteis em suas telas iniciais sem precisar acessar uma loja de aplicativos.
- **Linkável** - Compartilhe facilmente por URL, não requer instalação complexa.





## PWA - Desvantagens

- **Play Store** - sua propriedade perde um significativo tráfego e fonte de descoberta.
- Dependendo do **navegador**, plugins como Facebook login e Google login não conseguem capturar dados dos Apps instalados.
- PWAs **não conseguem acessar alguns funções dos dispositivos** como Bluetooth e NFC.

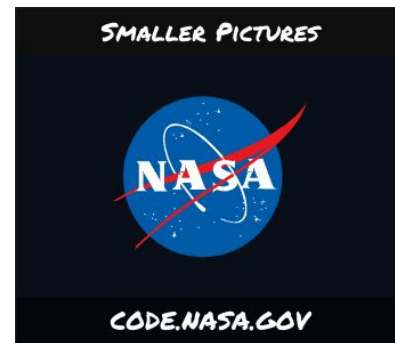
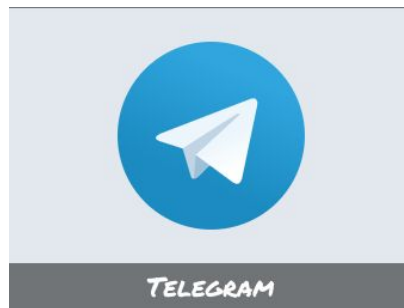


## PWA - Desvantagens

- Há uma série de **limitações para iOS**, como notificações e acesso offline, entre outros.
- **Não há integração com sensores**, o que inviabiliza seu uso com wearables, como **smart watches**, **phones wireless** e **fitness trackers**.



## PWA - Cases



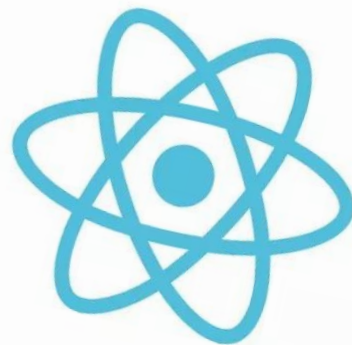
Fonte: <https://pwa.rocks/>



# PWA - Checklist

- **HTTPS** (Implement HTTPS );
- **Páginas responsivas** tanto para tablets quanto para celulares;
- Todas as URLs devem ser possíveis carregar quando offline (**Service Worker**);
- Metadata que possibilite o **Add to Home screen** (Web App Manifest)
- **Carregamento rápido** mesmo em 3G;
- Site cross-browser (**Suporte à múltiplos navegadores**);
- Cada página deve ter uma **URL**.

# Implementando uma aplicação PWA



**React JS**



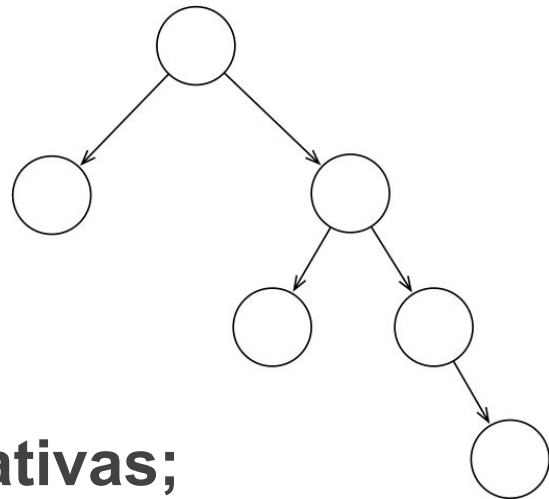
**Vue.js**



# React - Princípios

## Declarativo

- React **facilita a criação de UIs interativas;**
- React **atualizará e renderizará** com eficiência os componentes corretos quando os dados forem alterados.





# React - Princípios

## Baseado em componentes

Crie **componentes encapsulados** que gerenciam seu próprio estado e, em seguida, componha-os para criar interfaces de usuário complexas.

# React - Syntax

```
1 |
2 | import React, { Component } from 'react';
3 | import './App.css';
4 |
5 | export default class App extends Component {
6 |   constructor(props) {
7 |     super(props);
8 |     this.state = { count: 0 }
9 |   }
10 |
11 |   render() {
12 |
13 |   }
14 | }
```

```
1 | import React, { Component } from 'react';
2 | import './App.css';
3 |
4 | export default class App extends Component {
5 |   constructor(props) {
6 |     super(props);
7 |     this.state = { count: 0 }
8 |   }
9 |
10 |   incFunc = () => {
11 |     let { count } = this.state
12 |     this.setState({ count: count + 1 })
13 |   }
14 |
15 |   decFunc = () => {
16 |     let { count } = this.state
17 |     this.setState({ count: count - 1 })
18 |   }
19 |
20 |   render() {
21 |     return (
22 |       <div className="App">
23 |         <h1>Minicurso React</h1>
24 |         <button onClick={() => this.incFunc()}>
25 |           Clique aqui para incrementar
26 |         </button>
27 |         <p>{this.state.count}</p>
28 |         <button onClick={() => this.decFunc()}>
29 |           Clique aqui para decrementar
30 |         </button>
31 |       </div>
32 |     );
33 |   }
34 | }
```



# React - Syntax

```
9
10 incFunc = () => {
11   let { count } = this.state
12   this.setState({ count: count + 1 })
13 }
14
15 decFunc = () => {
16   let { count } = this.state
17   this.setState({ count: count - 1 })
18 }
19
```

```
1 import React, { Component } from 'react';
2 import './App.css';
3
4 export default class App extends Component {
5   constructor(props) {
6     super(props);
7     this.state = { count: 0 }
8   }
9
10   incFunc = () => {
11     let { count } = this.state
12     this.setState({ count: count + 1 })
13   }
14
15   decFunc = () => {
16     let { count } = this.state
17     this.setState({ count: count - 1 })
18   }
19
20   render() {
21     return (
22       <div className="App">
23         <h1>Minicurso React</h1>
24         <button onClick={() => this.incFunc()}>
25           Clique aqui para incrementar
26         </button>
27         <p>{this.state.count}</p>
28         <button onClick={() => this.decFunc()}>
29           Clique aqui para decrementar
30         </button>
31       </div>
32     );
33   }
34 }
```

# React - Syntax

```
render() {  
  return (  
    <div className="App">  
      <h1>Minicurso React</h1>  
      <button onClick={() => this.incFunc()}>  
        Clique aqui para incrementar  
      </button>  
      <p>{this.state.count}</p>  
      <button onClick={() => this.decFunc()}>  
        Clique aqui para decrementar  
      </button>  
    </div>  
  );  
}
```

```
1  import React, { Component } from 'react';  
2  import './App.css';  
3  
4  export default class App extends Component {  
5    constructor(props) {  
6      super(props);  
7      this.state = { count: 0 }  
8    }  
9  
10   incFunc = () => {  
11     let { count } = this.state  
12     this.setState({ count: count + 1 })  
13   }  
14  
15   decFunc = () => {  
16     let { count } = this.state  
17     this.setState({ count: count - 1 })  
18   }  
19  
20   render() {  
21     return (  
22       <div className="App">  
23         <h1>Minicurso React</h1>  
24         <button onClick={() => this.incFunc()}>  
25           Clique aqui para incrementar  
26         </button>  
27         <p>{this.state.count}</p>  
28         <button onClick={() => this.decFunc()}>  
29           Clique aqui para decrementar  
30         </button>  
31       </div>  
32     );  
33   }  
34 }
```



localhost:3000



# Minicurso React

Clique aqui para incrementar

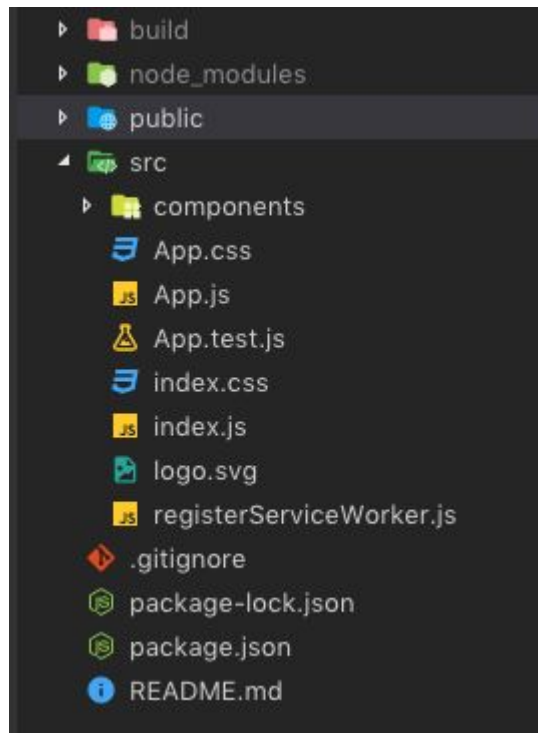
0

Clique aqui para decrementar



# React - Implementando Hello World

1. `npm install -g create-react-app`
2. `create-react-app hello-world`
3. `cd hello-world`
4. `npm start` ou `yarn start`
5. implementar no arquivo `src/App.js`
6. <https://gist.github.com/fss6>





localhost:3000



Edit `src/App.js` and save to reload.

[Learn React](#)



# React - Implementando Hello World

1. Implementar hello world no arquivo `src/App.js`
2. Link para o código completo:  
<https://gist.github.com/fss6>

## Minicurso React

Clique aqui para incrementar

0

Clique aqui para decrementar

Como passar este texto por props?





## Minicurso React

Clique aqui para incrementar

0

Clique aqui para decrementar

index.js — hello-wor

App.js

JS index.js ✕

```
1 import React from 'react'; 32
2 import ReactDOM from 'react-dom'; 33
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
```

```
ReactDOM.render(<App text='Minicurso React' />, document.getElementById('root'));
```

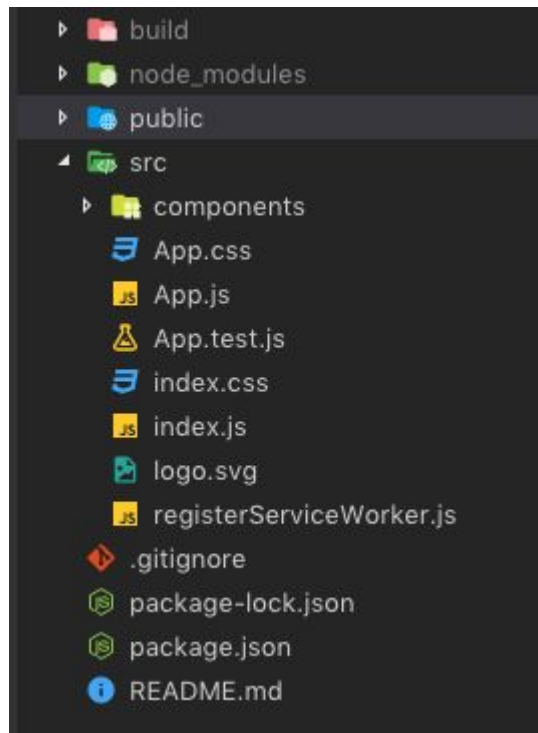
```
20 render() {
21   return (
22     <div className="App">
23       <h1>{this.props.text}</h1>
24       <button onClick={() => this.incFunc()}>
25         Clique aqui para incrementar
26       </button>
27       <p>{this.state.count}</p>
28       <button onClick={() => this.decFunc()}>
29         Clique aqui para decrementar
30       </button>
31     </div>
32   );
33 }
```





# React - Implementando TODO List

1. **npm install -g create-react-app**
2. **create-react-app todo-list**
3. **cd todo-list**
4. **npm start** ou **yarn start**
5. implementar no arquivo **src/App.js**
6. <https://gist.github.com/fss6>





localhost:3000



## TODO List

O que você precisa fazer?

```
export default class App extends Component {  
  constructor(props) {  
    super(props);  
    this.state = { items: [], text: '' };  
    this.handleChange = this.handleChange.bind(this);  
    this.handleSubmit = this.handleSubmit.bind(this);  
  }  
}
```

```
handleChange = (event) => {  
  this.setState({ text: event.target.value });  
}
```

```
handleSubmit = (event) => {  
  event.preventDefault();  
  
  if (!this.state.text.length) return;  
  
  const newItem = {  
    text: this.state.text,  
    id: Date.now()    
  };  
  this.setState(state => ({  
    items: state.items.concat(newItem),  
    text: ''  
  }));  
}
```

```

31 render() {
32   return (
33     <div className="App">
34       <h3>TODO List</h3>
35       <form onSubmit={this.handleSubmit}>
36         <label htmlFor="new-todo"> O que você precisa fazer? </label>
37         <input
38           id="new-todo"
39           onChange={this.handleChange}
40           value={this.state.text}
41           placeholder="Digite uma tarefa"
42         />
43       </form>
44       <TodoList items={this.state.items} />
45     </div>
46   );
47 }
48 }

```

```

50 class TodoList extends React.Component {
51   render() {
52     return (
53       <ul>
54         {this.props.items.map(item => (
55           <li key={item.id}>{item.text}</li>
56         ))}
57       </ul>
58     );
59   }
60 }
61

```



# React - Implementando TODO List



## TODO List

O que você precisa fazer?

1. Implementar TODO List no arquivo `src/App.js`
2. Link para o código completo:  
<https://gist.github.com/fss6>



# Implementando o Jogo da Velha




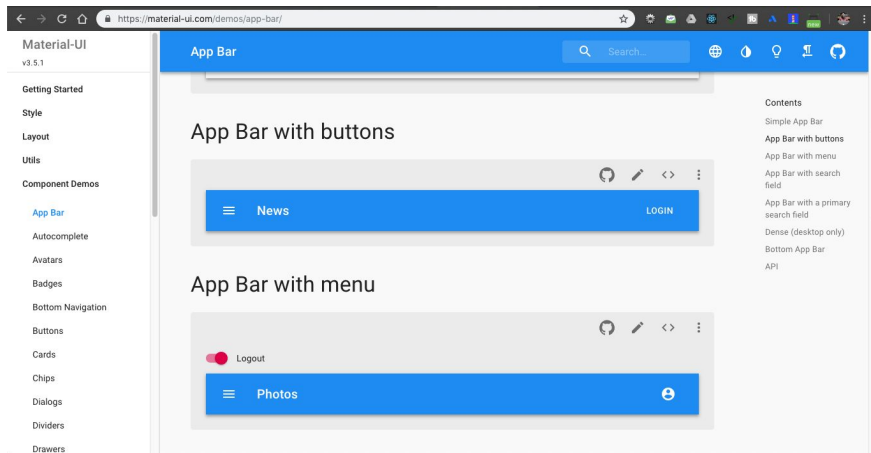


## MATERIAL-UI

## React components that implement Google's Material Design.

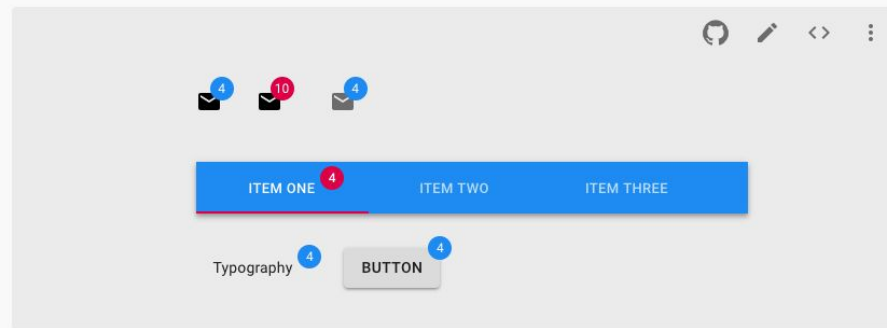
[GET STARTED](#)

 Get Professionally Supported Material-UI



## Simple Badges

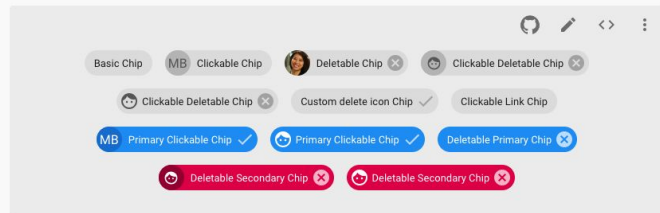
Examples of badges containing text, using primary and secondary colors. The badge is applied to its children.



## Chip

Examples of Chips, using an image Avatar, SVG Icon Avatar, "Letter" and (string) Avatar.

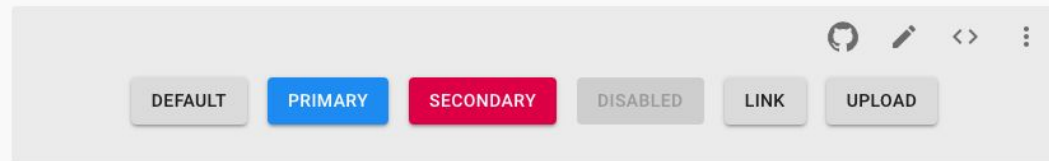
- Chips with the `onClick` property defined change appearance on focus, hover, and click.
- Chips with the `onDelete` property defined will display a delete icon which changes appearance on hover.



## Contained Buttons

**Contained buttons** are high-emphasis, distinguished by their use of elevation and fill. They contain actions that are primary to your app.

The last example of this demo show how to use an upload button.







# Quantos componentes temos?

Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

Partida encerrada!!!  
Temos um vencedor.

RESETAR



# Quantos componentes temos?

## Game.jsx

Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

Partida encerrada!!!  
Temos um vencedor.

RESETAR

## Board.jsx

Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

Partida encerrada!!!  
Temos um vencedor.

RESETAR

## Square.jsx

Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

Partida encerrada!!!  
Temos um vencedor.

RESETAR

## Panel.jsx

Jogo da Velha (PWA)

X	O	O
X	O	X
O	X	

Partida encerrada!!!  
Temos um vencedor.

RESETAR



# Implementando o Jogo da Velha

1. `npm install -g create-react-app`
2. `create-react-app jogo-da-velha`
3. `cd jogo-da-velha`
4. `npm install @material-ui/core`
5. `npm install @material-ui/icons`
6. `npm start`



# Implementando o Jogo da Velha

- index.html
- static files (imagens):  
<https://github.com/fss6/jogo-da-velha-pwa/raw/master/public/static/images.zip>
- App.js
- Game.jsx
- Board.jsx
- Square.jsx
- Panel.jsx
- <https://gist.github.com/fss6>



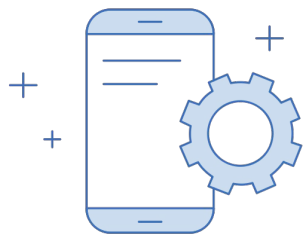
# Manifest.json

```
1  {
2    "short_name": "Jogo da Velha",
3    "name": "Jogo da Velha - PWA",
4    "icons": [
5      {
6        "src": "static/images/logo-192x192.png",
7        "sizes": "192x192",
8        "type": "image/png"
9      },
10     {
11       "src": "static/images/logo-512x512.png",
12       "sizes": "512x512",
13       "type": "image/png"
14     }
15   ],
16   "start_url": "./index.html",
17   "display": "standalone",
18   "orientation": "portrait",
19   "theme_color": "#000000",
20   "background_color": "#ffffff"
21 }
```

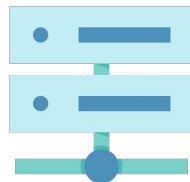
<https://developer.mozilla.org/pt-BR/docs/Web/Manifest>



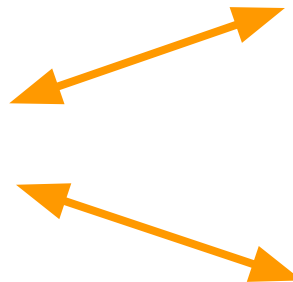
# Service Work



**Aplicação**



**Service Worker**



**Cache**

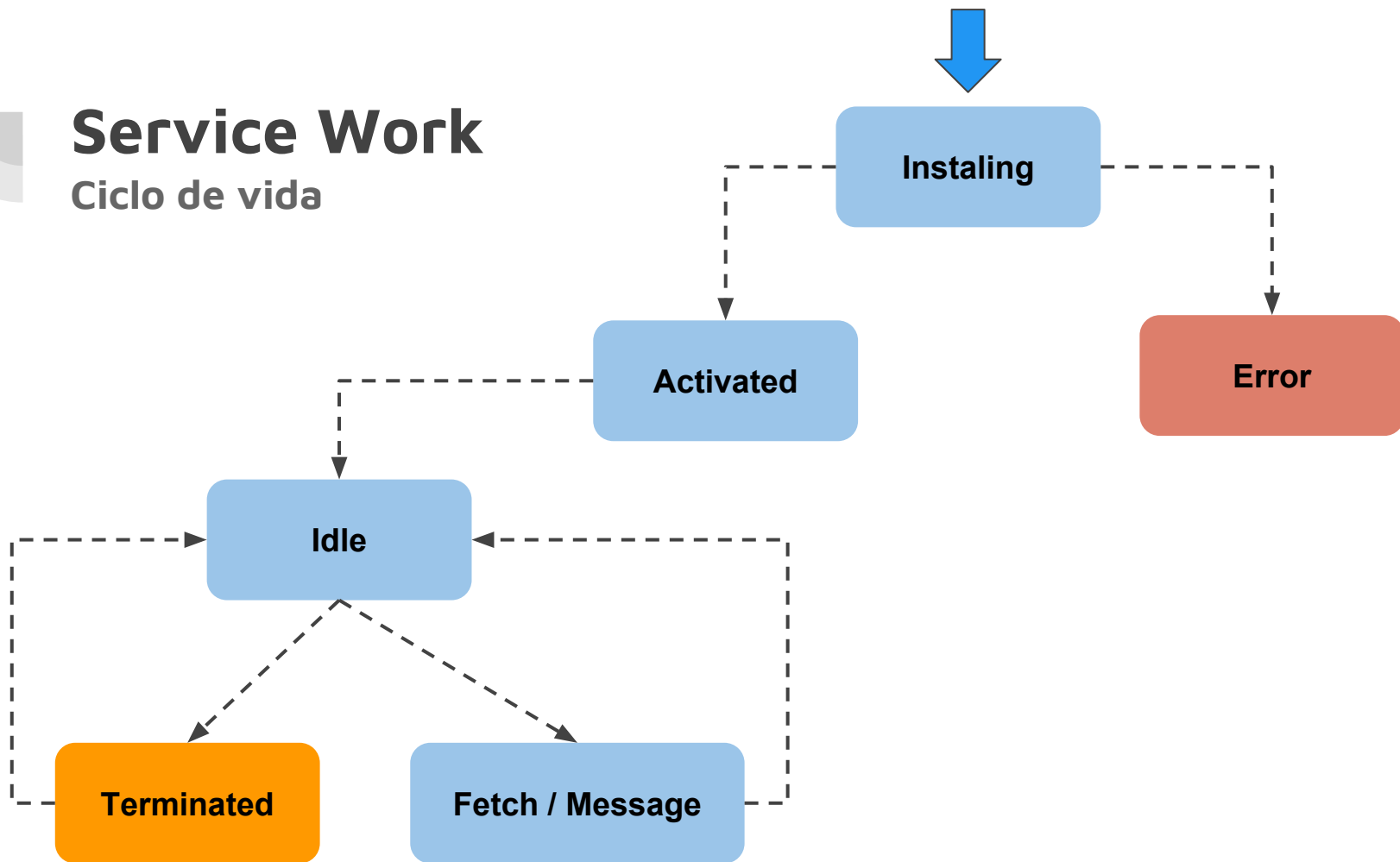


**Network**



# Service Work

Ciclo de vida





## Service Work

- `serviceWorker.unregister();`
- to
- `serviceWorker.register();`





# Light House - Teste PWA





# Pwa - Deploy

<https://gist.github.com/fss6>

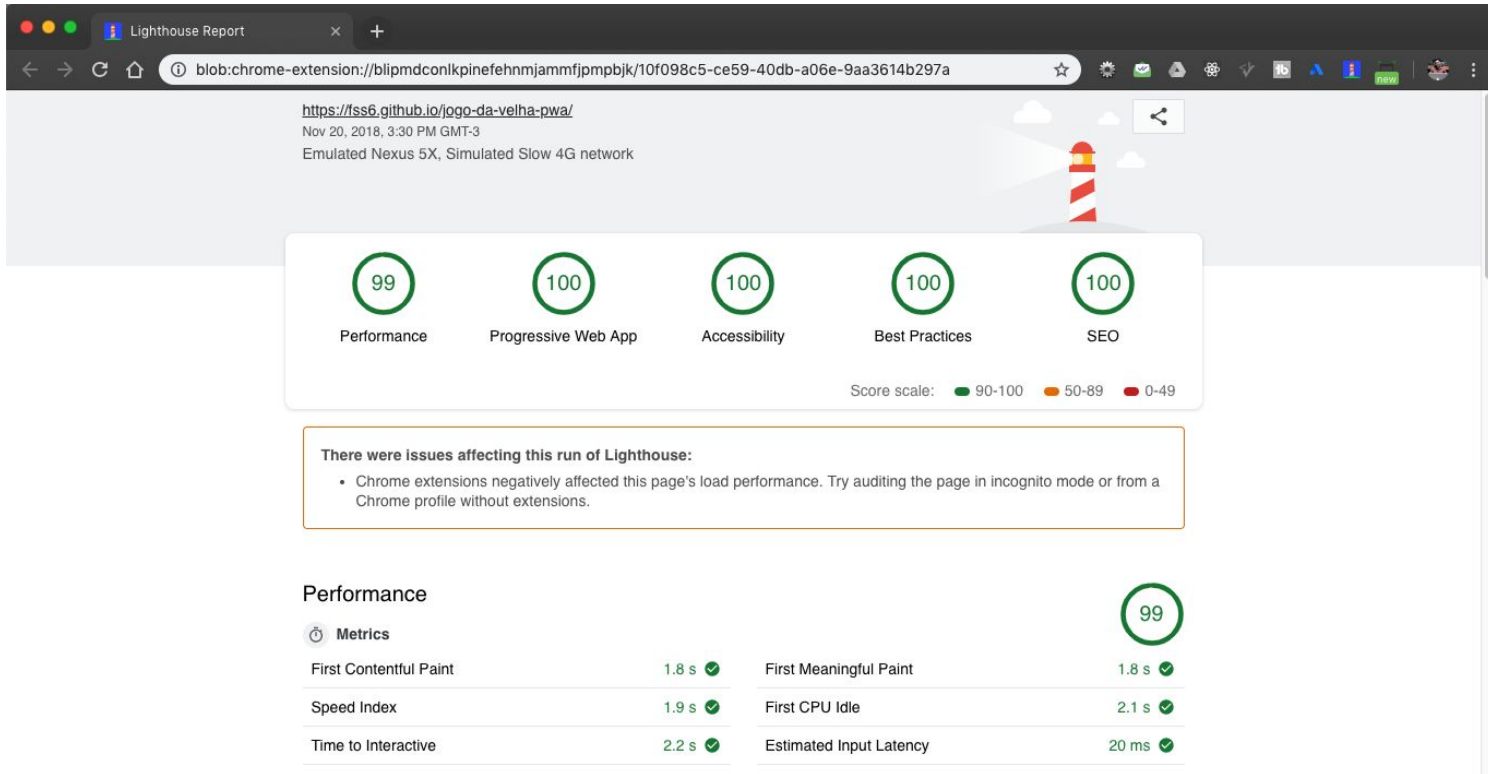
```
{
  "name": "react-jogodavelha",
  "version": "0.1.0",
  "homepage": "https://fss6.github.io/jogo-da-velha-pwa",
  "private": true,
  "dependencies": {
    "@material-ui/core": "^3.5.1",
    "@material-ui/icons": "^3.0.1",
    "gh-pages": "^1.1.0",
    "husky": "^0.14.3",
    "lint-staged": "^6.0.0",
    "prettier": "^1.9.1",
    "react": "^16.1.1",
    "react-dom": "^16.1.1",
    "react-scripts": "1.0.17"
  },
  "scripts": {
    "predeploy": "npm run build",
    "start": "react-scripts start",
    "deploy": "gh-pages -d build",
    "build": "react-scripts build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject"
  }
}
```



## Pwa - Deploy

- Criar ou logar na conta do GitHub
- Criar novo repositório
- `git init`
- `git add .`
- `git commit -m "Jogo da Velha"`
- `git add remote`
- `git push origin master`
- `npm run build`
- `npm run deploy`

# Light House - Teste PWA





# Obrigado!

