



Leonardo



Matheus



Diogo



Pêrsio



Yuri



g-28





g-28



> Problema:
Construir um aplicativo de perguntas
e respostas com React-Redux

Projeto Trivia

TRYBE

Trocar Tema

TRIVIA

EMAIL DO GRAVATAR

NOME

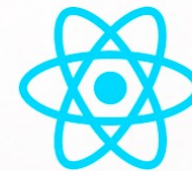
Entrar

Configurações

```

1  import styled, { createGlobalStyle } from 'styled-components';
2
3  export const GlobalStyle = createGlobalStyle`
4  body{
5    background-color: var(--secondary-color);
6  }`;
7
8  export const Container = styled.div`
9  display: flex;
10 flex-direction: column;
11 width: 80%;
12 margin: auto;
13 margin-top: 100px ;
14
15 & h1 {
16   text-align: center;
17   color: var(--accent-color);
18   font-size: 2.5rem;
19 }
20
21 & div {
22   display: flex;
23   justify-content: space-between;
24   align-items: center;
25   border: 1px solid var(--azul-light) ;
26 }

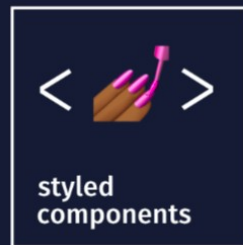
```



React



Redux



g-28

**Ferramentas
utilizadas**



Trello



slack



g-28



DESAFIOS:

- COMUNICAÇÃO
- PLANEJAMENTO
- LÓGICA DE PROGRAMAÇÃO
- NOVAS FERRAMENTAS

> CÓDIGOS



```

1 <S.ContainerAlt
2   data-testid="answer-options"
3 >
4   {alternatives.sort(() => Math.random() - NUMBER_SORT)
5     .map((element) => (
6       <S.Buttons
7         onClick={ this.handleClick }
8         key={ element }
9         type="button"
10        name={ element }
11        disabled={ isDisabled }
12        data-testid={
13          element === correctAnswer
14            ? CORRECT
15            : `wrong-answer-${incorrectAnswers.indexOf(element)}`
16        }
17        id={
18          element === correctAnswer
19            ? CORRECT
20            : `wrong-answer-${incorrectAnswers.indexOf(element)}`
21        }
22      >
23        {element}
24      </S.Buttons>
25    )}
26 </S.ContainerAlt>

```

```

1 startTimer = () => {
2   this.sortListofQuestions();
3   const THIRTY = 30000;
4   const { dispatch } = this.props;
5   this.timeoutID = setTimeout(() => {
6     dispatch(timerAction());
7     this.setState({
8       isDisabled: true,
9       nextQuestion: false,
10      firstQuestion: false });
11   }, THIRTY);
12 }
13
14 handleClick = (event) => {
15   if (event.target.id === 'correct-answer') {
16     this.handleScore();
17   }
18
19   const { dispatch } = this.props;
20
21   dispatch(timerAction());
22   clearTimeout(this.timeoutID);
23 }

```

```

1 import styled, { css } from 'styled-components';
2
3 export const ContainerAlt = styled.div`
4   display: flex;
5   flex-direction: column;
6   margin: 0 20px;
7   font-size: 600px;
8   align-items: center;
9 `;
10
11 export const Buttons = styled.button`
12   width: 550px;
13   height: 50px;
14   background-color: white;
15   border-color: #f9f9f9;
16   margin-top: 3px;
17   cursor: pointer;
18   border: 2px solid lightgrey;
19
20   &.disabled {
21     cursor: default;
22     border: ${({props}) => (`${props.id}` === 'correct-answer'
23       ? css`solid green 2px` : css`solid red 2px`)};
24   }
25 `;

```




```
1 import styled, { css } from 'styled-components';
2
3 export const ContainerAlt = styled.div`
4   display: flex;
5   flex-direction: column;
6   margin: 0 20px;
7   font-size: 600px;
8   align-items: center;
9 `;
10
11 export const Buttons = styled.button`
12   width: 550px;
13   height: 50px;
14   background-color: white;
15   border-color: #f9f9f9;
16   margin-top: 3px;
17   cursor: pointer;
18   border: 2px solid lightgrey;
19
20   &:disabled {
21     cursor: default;
22     border: ${({ props }) => ( `${props.id}` === 'correct-answer'
23       ? css`solid green 2px` : css`solid red 2px` )};
24   }
25 `;
```





```
1  startTimer = () => {
2    this.sortListOfQuestions();
3    const THIRTY = 30000;
4    const { dispatch } = this.props;
5    this.timeoutID = setTimeout(() => {
6      dispatch(timerAction());
7      this.setState({
8        isDisabled: true,
9        nextQuestion: false,
10       firstQuestion: false });
11    }, THIRTY);
12  }
13
14  handleClick = (event) => {
15    if (event.target.id === 'correct-answer') {
16      this.handleScore();
17    }
18
19    const { dispatch } = this.props;
20
21    dispatch(timerAction());
22    clearTimeout(this.timeoutID);
23  }
```


> em produção



OBSERVAÇÕES

Precisamos melhorar em:

- Confiança interna
- Comunicação
- Planejamento

Precisamos nos guiar:

- Para ser mais assertivo
- Assumir sentimento de dono





Diogo Martini Pantaleão, o jogo começou. Você tem 30s por pergunta...

Pontos: _ 0

Trocar Tema

Entertainment: Video Games

Tempo: 19

League of Legends, DOTA 2, Smite and Heroes of the Storm are all part of which game genre?

First Person Shooter (FPS)

Real Time Strategy (RTS)

Role Playing Game (RPG)

Multiplayer Online Battle Arena (MOBA)



g-28



Obrigado !