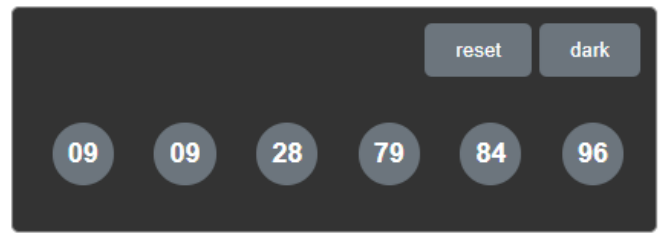
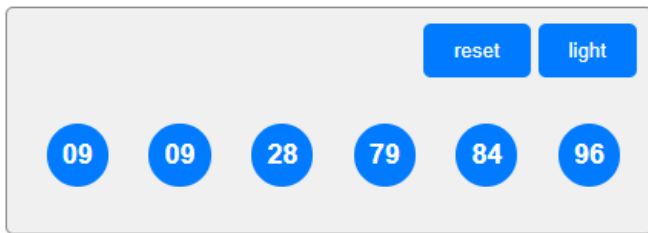


Descrição da atividade: fazer um aplicativo React TS para gerar 6 números aleatórios e exibir na tela. O usuário poderá escolher entre os temas light e dark.



Nota: 0,25 pts. na média final

Data de entrega: 07/maio

Forma de entrega: individual e presencial

Objetivos:

- Criar aplicações React TS;
- Context e hooks;
- Styled componentes e temas;
- Local storage do navegador.

Explicação do código:

Parte 1 – Estilos - <https://youtu.be/TEOfzj9h79g>

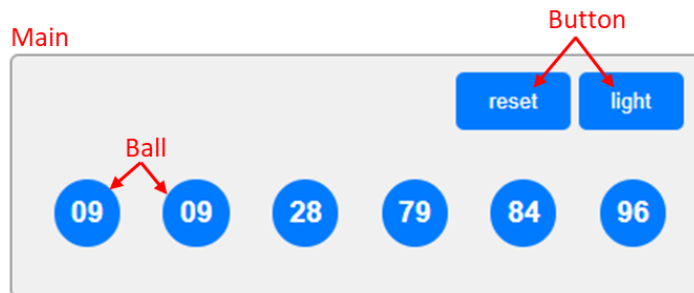
Parte 2 – Contexto - <https://youtu.be/erkmZFCD7tA>

Requisitos funcionais:

- Ao abrir a aplicação deverão ser gerados 6 números aleatórios no intervalo [0,99];
- Ao clicar no botão **reset** deverão ser gerados 6 números aleatórios no intervalo [0,99] e atualizar a tela;
- Ao clicar no botão **light/dark** o tema deverá ser trocado e o rótulo do botão deverá ser atualizado;
- Os números deverão ser apresentados em ordem crescente de valor;
- Ao clicar no número o seu valor deverá ser incrementado em 1 e atualizado na tela. Os números deverão ser apresentados em ordem crescente.

Requisitos não funcionais:

- O projeto deverá ter a estrutura mostrada ao lado;
- O layout deverá ser definido usando CSS Flexbox;
- A tela deverá ser estruturada usando os componentes destacados na figura a seguir. O componente Main deverá estar centralizado na vertical e horizontal na tela do navegador;



- Utilizar herança de estilo no componente **Ball**. O estilo do componente **Ball** deverá estender (herdar) do estilo do componente **Button**, visto que a maioria das propriedades do estilo são semelhantes;
- Deverão ser usadas as seguintes cores nos estilos. Os estilos deverão ser definidos no arquivo `src/themes/theme.ts` e propagados usando `ThemeProvider` do pacote `styled-components`;

Estilo	Tema light	Tema dark
Cor de fundo do painel	#f0f0f0	#333
Cor de fundo do botão	#007bff	#6c757d
Cor de texto do botão	#fff	#fff
Cor de fundo do botão com o mouse sobre	#0056b3	#495057

- Todos os tipos deverão ser definidos no arquivo `src/types/index.ts`, ou seja, as propriedades recebidas por cada componente e usada no contexto deverão ser definidas no arquivo de tipos;
- O contexto deverá propagar o estilo atual, o array de 6 números, a função para incrementar um número, a função para resetar os números e a função para mudar de tema;
- As propriedades do contexto deverão ser acessadas usando o hooks definido no arquivo `src/hooks/index.ts`.

```

▼ FRONT
  > node_modules
  > public
  ▼ src
    ▼ components
      ▼ Ball
        ⚙ index.tsx
      ▼ Button
        ⚙ index.tsx
      ▼ Menu
        ⚙ index.tsx
    TS index.ts
  ▼ contexts
    ⚙ RandomContext.tsx
  ▼ hooks
    TS index.ts
  ▼ pages\Main
    ⚙ index.tsx
  ▼ styles
    TS theme.ts
  ▼ types
    TS index.ts
  ⚙ App.tsx
  # index.css
  ⚙ index.tsx
  TS react-app-env.d.ts
  ⚙ .env
  .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md
  TS tsconfig.json
  
```