



Fundamentos React - Projeto Clone Experiências AirBnb

O que vamos aprender

- props
- criar componentes a partir de um array

1. Criando seu app

Crie uma pasta chamada `react`. Recomendamos que esta pasta seja criada dentro da pasta que foi criada para o curso.

Dentro da pasta `react` crie uma pasta chamada `projeto2-airbnb-experiences`.

Execute `npm create-react-app airbnb-clone`, dê o comando `cd` em seu projeto e abra-o. No componente App.js Você pode excluir tudo na instrução `return` de forma que vai retornar apenas um elemento div vazio. Você também pode excluir todos os templates create-react-app fornecidos e apenas deixar `index.js` e `App.js` no diretório src.

Desafio 1:

- Crie um componente Navbar;
- Verifique o arquivo Figma correspondente para as especificações do projeto.

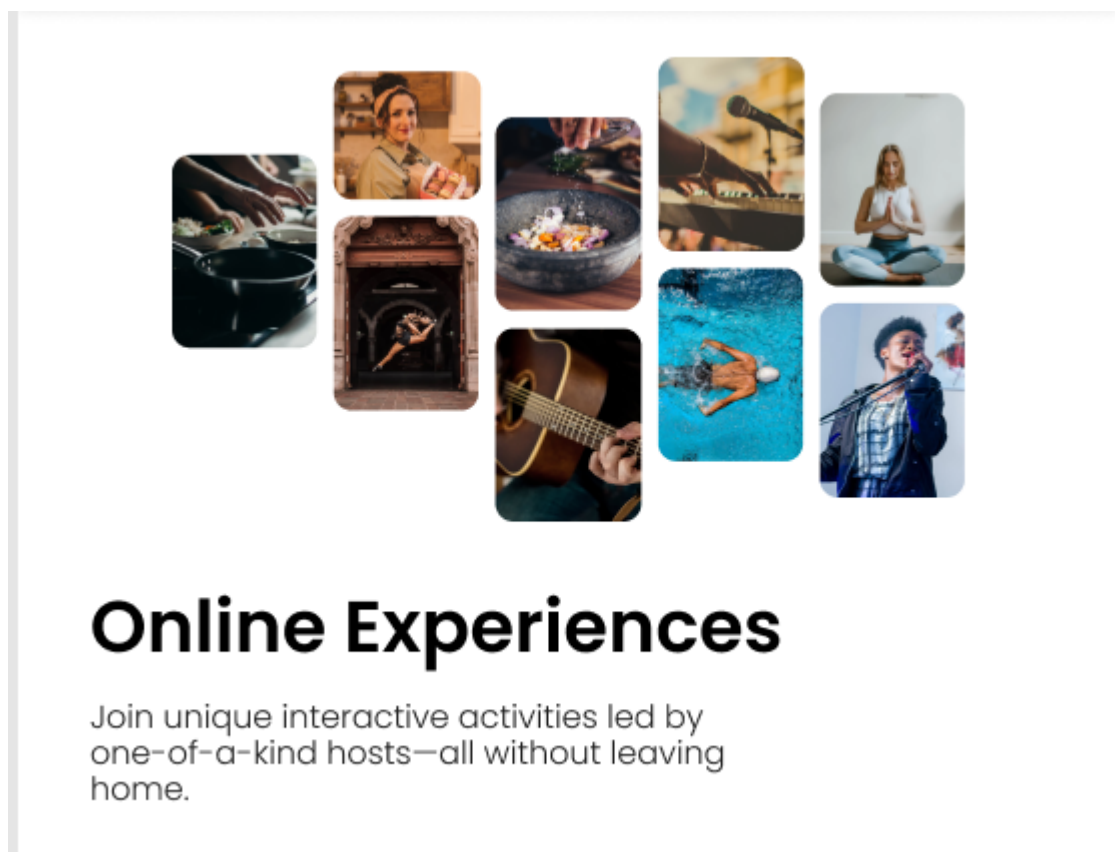
Resultado desejado:



Desafio 2:

- Crie um componente Hero;
- Verifique o arquivo Figma correspondente para as especificações do projeto.

Resultado desejado:

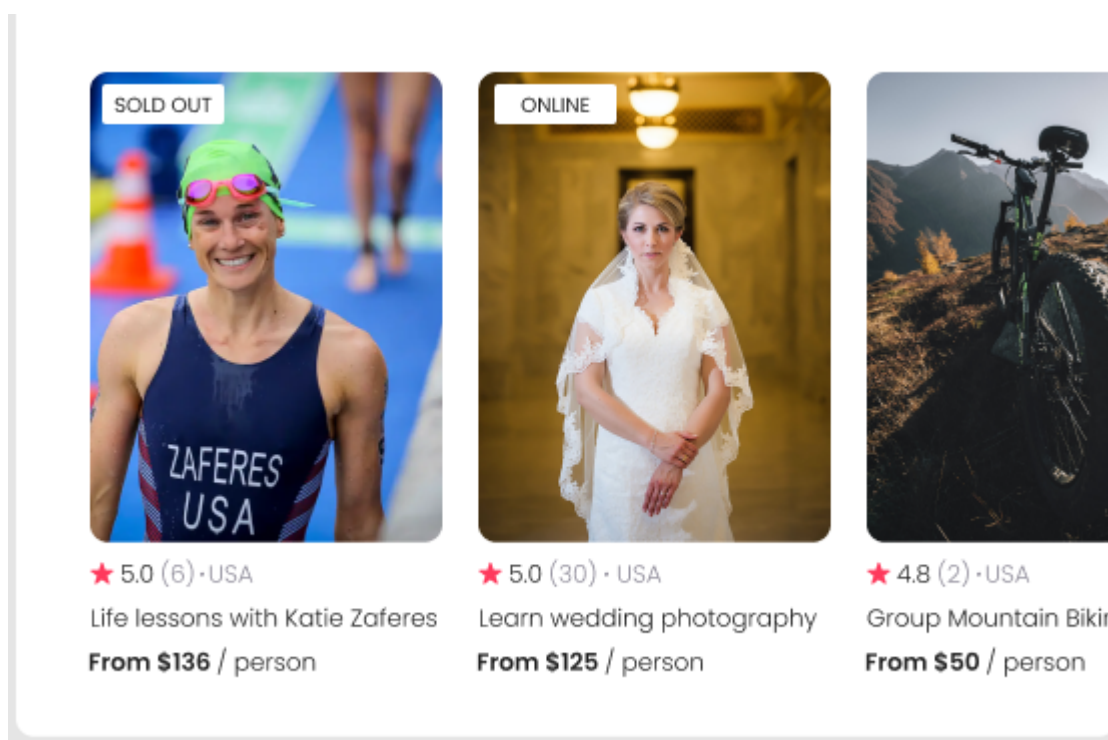


Desafio 3:

- Crie um componente Card. Por enquanto, codifique os dados (como a classificação (rating), título (title), preço (price), etc.);
- Verifique o arquivo Figma correspondente para as especificações do projeto.
- **Notas:**
 - Renderize apenas 1 instância;

- O ícone de estrela e a foto (katie-zaferes.png) estão nos arquivos e pasta de imagens.
- Certifique-se de incluir:
 - imagem(img)
 - ícone de estrela (star.png), classificação (rating) e contagem de resenhas (reviews)
 - título (title)
 - custo/pessoa
- O principal objetivo deste desafio é mostrar onde estão nossas limitações atualmente, então não se preocupe com o fato de que você está codificando todos esses dados no componente.

Resultado desejado:



Desafio 4:



- Crie um componente Contact.js em outro arquivo;
- Mova um dos divs de cartão de contato abaixo para esse arquivo;
- importe e renderize 4 instâncias desse cartão de contato
 - Pense bem: qual é o problema de fazer assim?

Contact.js

```
import React from "react"

export default function Contact() {
  return (
    <div className="contact-card">
      
      <h3>Mr. Whiskerson</h3>
      <div className="info-group">
        
        <p>(212) 555-1234</p>
      </div>
      <div className="info-group">
        
        <p>mr.whiskaz@catnap.meow</p>
      </div>
    </div>
  )
}
```

App.js

```
import React from "react";
import Contact from "../components/Contact";

export default function App() {
  return (
```



```
<div className="contacts">
  <Contact />
  <Contact />
  <Contact />
  <Contact />
</div>
);
}
```

Desafio 5:

- Adicione todo o resto do dados para as instâncias de cartão de contato

App.js

```
import React from "react";
import Contact from "../components/Contact";

export default function App() {
  return (
    <div className="contacts">
      <Contact
        img="./images/mr-whiskerson.png"
        name="Mr. Whiskerson"
        phone="(212) 555-1234"
        email="mr.whiskaz@catnap.meow"
      />
      <Contact
        img="./images/fluffykins.png"
        name="Fluffykins"
        phone="(212) 555-2345"
        email="fluff@me.com"
      />
      <Contact
        img="./images/felix.png"
        name="Felix"
      />
    </div>
  );
}
```



```
    phone="(212) 555-4567"
    email="thecat@hotmail.com"
  />
  <Contact
    img="./images/pumpkin.png"
    name="Pumpkin"
    phone="(0800) CAT KING"
    email="pumpkin@scrimba.com"
  />
</div>
);
}
```

Desafio 6:

- Corrigir o código do componente Contact.js para usar os `props`;
- Coloque os valores de objeto (props) no lugar dos valores codificados.

Contact.js

```
import React from "react"

export default function Contact(props) {
  return (
    <div className="contact-card">
      <img src={props.img} alt="imagem Mr Whiskerson"/>
      <h3>{props.name}</h3>
      <div className="info-group">
        
        <p>{props.phone}</p>
      </div>
      <div className="info-group">
        
        <p>{props.email}</p>
      </div>
    </div>
  )
}
```



```
    </div>
  )
}
```

Desafio 7:

- O aplicativo deve renderizar de 4 a 5 componentes <Citacoes /> (Componente *Citacoes* definido em seu próprio arquivo também)
- Cada citação deve receber um prop "Citação" e um prop de "Autor" e renderize-os como quiser.
- Use suas citações favoritas ou verifique o arquivo citacoes.md para alguns exemplos.

Citacoes.js

```
import React from "react"

export default function Citacoes(props) {
  return (
    <div>
      <h3>Citação: {props.citacao}</h3>
      <p>Autor: {props.autor}</p>
      <hr />
    </div>
  )
}
```

App.js

```
import React from "react"
import Citacoes from "../components/Citacoes"

export default function App() {
  return (
    <div>
      <Citacoes
        citacao="Creio no riso e nas lágrimas como
antídotos contra o ódio e o terror."
      />
    </div>
  )
}
```

```
        autor = "Charles Chaplin"
    />
    <Citacoes
        citacao="Seja como for o que penses, creio que
é melhor dizê-lo com boas palavras."
        autor = "William Shakespeare"
    />

    <Citacoes
        citacao="Creio que tenho prova suficiente de
que falo a verdade: a pobreza."
        autor = "Sócrates"
    />

    <Citacoes
        citacao="Creio que quase sempre é preciso um
golpe de loucura para se construir um destino."
        autor = "Marguerite Yourcenar"
    />

    <Citacoes
        citacao = "Eu creio haver corações que
poderiam cortar diamantes."
        autor = "Emanuel Wertheimer"
    />
</div>
)
}
```

Desafio 7:

- Passe props para o componente Card do projeto e exiba esses dados.
 - img ("katie-zaferes.png")
 - rating ("5.0")
 - reviewCount (6)
 - country (Whatever you want)
 - title ("Life Lessons with Katie Zaferes")
 - price (136)



Card.js

```
import React from "react";
import star from "../images/star.png";

export default function
Card({img, rating, reviewCount, country, title, price} ) {
  console.log(img)
  return (
    <div className="card">
      <img src= {`/images/${img}`} className="card-photo"
alt="imagem card" />
      <div className="card-rating">
        <img src={star} className="star" alt="imagem card" />
        <p>{rating} ({reviewCount}).{country}</p>
      </div>
      <p>{title}</p>
      <p>
        <strong>From ${price}</strong> / person
      </p>
    </div>
  );
}
```

App.js

```
import React from 'react';
import Navbar from './components/Navbar';
import Hero from './components/Hero';
import Card from './components/Card';

export default function App() {
  return (
    <div>
      <Navbar />
      <Hero />
    </div>
  );
}
```

```
<Card
  img = "katie-zaferes.png"
  rating = "5"
  reviewCount = {6}
  country = "USA"
  title = "Life Lessons with Katie Zaferes"
  price = {136}
/>
</div>
);
}
```

Desafio 8:

- Veja se você consegue passar corretamente os endereços necessários para o Componente Citacoes no .map() (e renderize o array elementData) então as citações voltam a aparecer na página

App.js

```
import React from "react"
import Citacoes from "../components/Citacoes"
import citacoesData from "../citacoesData"

export default function App() {

  const elementData = citacoesData.map(elemento =>{
    return <Citacoes citacao = {elemento.citacao} autor =
{elemento.autor}/>
  })
  return (
    <div>
      {elementData}
    </div>
  )
}
```



Desafio 9:

- Importe a matriz de dados de data.js
- Mapeie o array para criar componentes <Card />
- Exiba o array de componentes do cartão abaixo no componente Navbar (no lugar do componente <Card /> atual)

Observação: ainda não estilizamos o grupo de componentes, então eles ainda serão elementos de bloco, empilhados verticalmente. Adicionaremos estilo mais tarde.

Card.js

```
import React from "react";
import star from "../images/star.png";

export default function Card(props) {
  let badgeText
  if (props.openSpots === 0) {
    badgeText = "SOLD OUT"
  } else if (props.country === "Online") {
    badgeText = "ONLINE"
  }
  return (
    <div className="card">
      {badgeText} && <div
className="card-badge">{badgeText}</div>
      <img src= `{}/images/${props.img}`
className="card-photo" alt="imagem card" />
      <div className="card-rating">
        <img src={star} className="star" alt="imagem card" />
        <p>{props.rating}
({props.reviewCount}). {props.country}</p>
      </div>
      <p>{props.title}</p>
      <p>
        <strong>From ${props.price}</strong> / person
      </p>
    </div>
  )
}
```



```
</div>
);
}
```

styles.css

```
@import
url("https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap");

* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: "Poppins", sans-serif;
}

nav {
  width: 550px;
  height: 70px;
  padding-left: 30px;
  box-shadow: 0px 2.98256px 7.4564px rgba(0, 0, 0, 0.1);
  display: flex;
  align-items: center;
}

.nav-logo {
  max-width: 100px;
}

section {
  padding-top: 10px;
  max-width: 550px;
}
```



```
.hero {  
  display: flex;  
  flex-direction: column;  
}  
  
.hero-texto {  
  padding-left: 36px;  
}  
  
.hero-texto h1 {  
  font-size: 36px;  
  font-weight: 600;  
  line-height: 40px;  
}  
  
.hero-texto p {  
  font-size: 16px;  
  font-weight: 300;  
  line-height: 17.6px;  
}  
  
.hero-grid {  
  max-width: 400px;  
  margin: auto;  
}  
  
.card-list {  
  display: flex;  
  margin-left: 37px;  
  gap: 20px;  
}  
  
.card {  
  display: flex;  
  flex-direction: column;  
  padding-bottom: 30px;
```



```
max-width: 176px;
font-size: 12px;
font-weight: 300;
line-height: 0px;
position: relative;
}

.card-photo {
  max-width: 176px;
  border-radius: 9px;
}

.card-rating {
  display: flex ;
  align-items: center;
}

.star {
  width: 14px;
  height: 14px;
}

.card-badge {
  position: absolute;
  top: 8px;
  left: 6px;
  width: 80px;
  background-color: white;
  padding: 10px 8px;
  border-radius: 2px;
  font-weight: bold;
}
```