



React, bota um cropped!

Aula 3 - Hard Skills



Wander Flaur / Unsplash



Todos os direitos reservados
©2023 Resilia Educapio

Códigos
do Amanhã
2023 EBANX RESILIA



Objetivos de aprendizagem

- ⇒ Expressar noções básicas da biblioteca React.js;
- ⇒ Demonstrar possíveis meios de instalação com o CRA e Vite;
- ⇒ Apresentar a sintaxe do JSX;
- ⇒ Expressar noções básicas de componentes;
- ⇒ Expressar noções básicas de props.



Todos os direitos reservados
©2023 Resilia Educação



Nesta aula, veremos:

- ⇒ React básico;
 - ⇒ Instalação com Create React App;
 - ⇒ Instalação com Vite;
 - ⇒ JSX;
 - ⇒ Componentes;
 - ⇒ Props.
-

Por que aprender isso?

- ⇒ React é uma das bibliotecas JavaScript mais populares e amplamente adotadas para o desenvolvimento de interfaces de usuário. Ter conhecimento básico de React significa estar alinhado com as tendências e demandas do mercado de desenvolvimento web.
- ⇒ A sintaxe familiar, a renderização eficiente, a criação de componentes reutilizáveis e a integração com JavaScript são razões convincentes para dominar o JSX e tirar o máximo proveito da biblioteca React.
- ⇒ Já os componentes permitem modularidade, reutilização de código, organização do código, separação de preocupações e eficiência no desenvolvimento.

Introdução ao React



Todos os direitos reservados
©2023 Realiz Educação

React

- Uma das bibliotecas mais populares e usadas no mercado de trabalho.
- Biblioteca JavaScript para front-end.
- Projeto *open-source* criado pela Meta (ex-Facebook).
- Em um modelo MVC, React é a camada da View.



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

React é uma biblioteca JavaScript de código aberto amplamente utilizada para a criação de interfaces de usuário interativas e reativas. O React permite que os desenvolvedores construam componentes reutilizáveis e os combinem para formar interfaces complexas.

O React se baseia no conceito de programação declarativa, o que significa que os desenvolvedores descrevem como a interface do usuário deve ser exibida em um determinado estado, e o React cuida automaticamente da atualização e manipulação eficiente do DOM.

Aprender React é recomendado pois é amplamente adotado pela indústria e oferece um conjunto de recursos robusto para criar aplicativos web modernos.

Configuração e Instalação

Para usufruirmos dessa biblioteca devemos ter instalados:

- NPM (gerenciador de pacotes do Node.js)
- Webpack
- Babel

Opcionais:

- React Dev Tools (*extensão do navegador*)
- Prettier, ES7 React/Redux/GraphQL (*extensão do VS code*)



Todos os direitos reservados
©2023 Realiza Educação

Configuração e Instalação

Para instalação do React em nosso projeto podemos:

- Instalar a partir de um arquivo **HTML estático**
- Utilizar o template do **Create React App** (CRA)
- Utilizar o build tool **Vite** (*fortemente recomendado*)



Todos os direitos reservados
©2023 Realiza Educação

Create React App

Cria um ambiente de desenvolvimento já **configurado** e **otimizado** para criação de aplicações React.



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

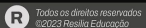
Create React App (CRA) é uma ferramenta desenvolvida pela equipe do React para facilitar a criação de projetos React de forma rápida e eficiente. É um ambiente pré-configurado que inclui todas as dependências e configurações necessárias para começar a desenvolver um aplicativo React sem a necessidade de configurações manuais complexas.

Com o CRA, os desenvolvedores podem se concentrar no código do aplicativo, pois o ambiente já cuida da estrutura básica, configurações de build, transpilação, hot-reloading e outras tarefas comuns. Isso torna o processo de início e desenvolvimento de projetos React mais simples e permite que os desenvolvedores se concentrem em criar aplicativos de alta qualidade.

Babel

É um **transpilador** JavaScript que permite a escrita de código JavaScript com sintaxes e recursos mais atuais, como do *ES6*, sendo responsável por **transformar esse código em um equivalente**, mas com uma sintaxe mais antiga, como a do *ES3*.

Isso nos traz vantagens de **suporte nos navegadores**.



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

Babel é uma ferramenta popular no ecossistema JavaScript que permite aos desenvolvedores converter o código JavaScript escrito em uma versão mais recente do ECMAScript (também conhecido como ES6+ ou ESNext) em uma versão compatível com navegadores e ambientes mais antigos.

O JavaScript evolui constantemente, introduzindo novos recursos e sintaxes mais modernas. No entanto, nem todos os navegadores e ambientes de execução suportam as versões mais recentes do JavaScript. É aí que o Babel entra em jogo.

Ao usar o Babel, você pode escrever código JavaScript usando as versões mais recentes do ECMAScript, como arrow functions, desestruturação, módulos ES6, entre outros recursos avançados. O Babel traduz esse código para uma versão mais antiga do JavaScript, geralmente ES5, que é amplamente suportada pelos navegadores.

Além de converter o código, o Babel também suporta a transformação de outras sintaxes, como o JSX usado no React, em código JavaScript padrão.

O Babel é altamente configurável e pode ser usado em conjunto com outras ferramentas, como Webpack, para automatizar o processo de transpilação. Ele é amplamente adotado na comunidade de desenvolvimento web para permitir o uso de recursos modernos do JavaScript sem se preocupar com a compatibilidade do navegador.

Webpack

Webpack é um **bundler**, responsável por agrupar todo o código e separá-lo em módulos, pode ser entendido como um **empacotador de módulos**.

Ele garante, por exemplo, que não tenhamos conflitos de código no nosso projeto, desde os mais simples, como nome de variáveis, até outros mais complexos.



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

Webpack é uma ferramenta de empacotamento (bundling) amplamente utilizada no desenvolvimento web. Ela é projetada para agrupar e otimizar vários arquivos JavaScript, CSS, imagens e outros recursos em um único pacote.

O principal objetivo do Webpack é lidar com as dependências e a modularidade do código em um projeto. Ele permite que você crie uma estrutura de arquivos bem organizada, onde os módulos JavaScript são importados e exportados uns dos outros, facilitando a divisão lógica e a reutilização do código.

Além disso, o Webpack oferece recursos avançados, como a capacidade de processar diferentes tipos de arquivos por meio de loaders. Por exemplo, ele pode transpilar o código JavaScript usando o Babel, converter estilos CSS em JavaScript, otimizar imagens e muito mais. Esses loaders permitem que você aplique transformações específicas em cada tipo de arquivo antes de serem incluídos no pacote final.

Outro recurso poderoso do Webpack é o sistema de plugins. Os plugins podem ser usados para executar tarefas adicionais durante o processo de empacotamento, como minificação de código, injeção de variáveis de ambiente, criação de cache, entre


outras otimizações.

Além do empacotamento e otimização de arquivos, o Webpack também fornece recursos para o desenvolvimento local, como o servidor de desenvolvimento embutido, recarregamento automático (live reloading) e suporte a HMR (Hot Module Replacement), que permite a atualização instantânea dos módulos modificados sem recarregar a página.

A hand holding a pen writing on a spiral notebook. The background is dark and blurred, focusing on the hand and the pen. The text is overlaid on the image.

Criando um projeto com CRA


[Live coding]

 Todos os direitos reservados
©2023 Resilix Educação

Vite

Vite é um **build tool**, serve para facilitar a configuração e instalação de vários projetos, com diversas tecnologias, como React e Vue.

Tecnologia que tem entrado em ascensão, por ser uma ferramenta de build (*build tool*) para diversas tecnologias e oferecer configurações iniciais melhores que outras do mercado.

 Todos os direitos reservados
©2023 Resilia Educação

NOTA PARA O ESTUDANTE

Vite é uma ferramenta de construção de aplicativos web rápida e leve, desenvolvida especialmente para projetos baseados em frameworks modernos como Vue.js e React. Ao contrário das ferramentas tradicionais de construção, como Webpack, Vite adota uma abordagem de desenvolvimento mais rápida e eficiente.

A principal vantagem do Vite é o tempo de inicialização extremamente rápido. Em vez de empacotar todo o código do aplicativo em um único arquivo durante o desenvolvimento, o Vite utiliza o recurso de módulos nativos do navegador para carregar os módulos do aplicativo diretamente no navegador. Isso permite um tempo de resposta instantâneo durante o desenvolvimento, com recarregamento automático rápido sempre que você faz uma alteração no código.

Além disso, o Vite oferece suporte nativo para recursos como HMR (Hot Module Replacement), que permite a atualização instantânea de módulos modificados sem a necessidade de recarregar a página inteira, resultando em um fluxo de desenvolvimento mais produtivo.

Outra característica interessante do Vite é a sua capacidade de processar arquivos de

forma otimizada. Ele utiliza carregadores (loaders) para processar diferentes tipos de arquivos, como JavaScript, CSS, JSON e imagens, de forma eficiente e sob demanda. Isso significa que apenas os recursos necessários são carregados, reduzindo o tamanho do pacote final e melhorando o desempenho da aplicação.

Embora o Vite seja comumente associado ao Vue.js, ele também oferece suporte a outros frameworks, como React e Preact. Isso o torna uma escolha viável para aqueles que trabalham com diferentes tecnologias de front-end.

Vite vs CRA

- Servidor **mais rápido** que do CRA.
- Processo de **build otimizado**.
- Oferece uma API de **Hot Module Replacement**, que oferece atualizações instantâneas e precisas sem recarregar a página ou destruir o estado da aplicação, integrado com o Fast-Refresh do React.

NOTA PARA O ESTUDANTE

Vite é uma ferramenta de construção de aplicativos web rápida e leve, desenvolvida especialmente para projetos baseados em frameworks modernos como Vue.js e React. Ao contrário das ferramentas tradicionais de construção, como Webpack, Vite adota uma abordagem de desenvolvimento mais rápida e eficiente.

A principal vantagem do Vite é o tempo de inicialização extremamente rápido. Em vez de empacotar todo o código do aplicativo em um único arquivo durante o desenvolvimento, o Vite utiliza o recurso de módulos nativos do navegador para carregar os módulos do aplicativo diretamente no navegador. Isso permite um tempo de resposta instantâneo durante o desenvolvimento, com recarregamento automático rápido sempre que você faz uma alteração no código.

Além disso, o Vite oferece suporte nativo para recursos como HMR (Hot Module Replacement), que permite a atualização instantânea de módulos modificados sem a necessidade de recarregar a página inteira, resultando em um fluxo de desenvolvimento mais produtivo.

Outra característica interessante do Vite é a sua capacidade de processar arquivos de

forma otimizada. Ele utiliza carregadores (loaders) para processar diferentes tipos de arquivos, como JavaScript, CSS, JSON e imagens, de forma eficiente e sob demanda. Isso significa que apenas os recursos necessários são carregados, reduzindo o tamanho do pacote final e melhorando o desempenho da aplicação.


Embora o Vite seja comumente associado ao Vue.js, ele também oferece suporte a outros frameworks, como React e Preact. Isso o torna uma escolha viável para desenvolvedores web júnior que trabalham com diferentes tecnologias de front-end.

Em resumo, o Vite é uma ferramenta de construção de aplicativos web rápida e eficiente, adequada para projetos baseados em frameworks modernos como Vue.js, React e Preact. Com seu tempo de inicialização rápido e suporte nativo ao HMR, o Vite oferece um fluxo de desenvolvimento ágil e produtivo, permitindo que os desenvolvedores se concentrem na criação de aplicativos sem preocupações com o desempenho durante o desenvolvimento.



Criando um projeto com Vite

[Live coding]

 Todos os direitos reservados
©2023 Resilite Educação

JSX

JavaScript XML / Extension é a **extensão de sintaxe do JavaScript**, introduz elementos de XML e HTML, que são convertidos em funções React, facilitando a sintaxe e a legibilidade do código.

JSX:

```
function App() {  
  return <button>Comprar</button>  
}
```

É transformado para:

```
function App() {  
  return React.createElement('button', null,  
    'Comprar')  
}
```



Todos os direitos reservados
©2023 Resilia Educação

NOTA PARA O ESTUDANTE

JSX (JavaScript XML) é uma extensão de sintaxe usada no React para escrever componentes de interface do usuário de forma declarativa. Ele permite que você misture HTML e JavaScript em um único arquivo, facilitando a criação e o gerenciamento de elementos da interface.

No JSX, você pode escrever tags HTML diretamente em seu código JavaScript, combinando a estrutura da interface do usuário com a lógica do aplicativo. Por exemplo, em vez de criar elementos HTML programaticamente usando métodos como `document.createElement()`, você pode escrever código JSX mais legível e expressivo.

Além disso, o JSX permite a incorporação de expressões JavaScript dentro do código, envolvendo-as em chaves `{}`. Isso permite que você insira dinamicamente valores, variáveis e até mesmo lógica condicional dentro dos componentes.

No entanto, é importante observar que o JSX não é interpretado diretamente pelo navegador. Antes de ser executado, ele precisa ser transpilado para código JavaScript puro usando ferramentas como o Babel. O processo de transpilação converte o código JSX em chamadas de função equivalentes que o React pode entender.

JSX

- Não é obrigatória sua sintaxe no React.
- É mais próximo de JavaScript do que HTML.
- Expressões JS podem ser colocadas dentro do JSX usando chaves { }.
- É mais fácil de escrever e entender do que criar e adicionar vários elementos com JavaScript puro.

JSX

Algumas diferenças importantes:

- Atributos podem ser passados como no HTML, mas existem casos especiais.
 - *exemplo*: class para a ser **className**, for para a ser **htmlFor**.
- Atributos com nome composto devem seguir o padrão camelCase.
- Podemos passar expressões (funções, variáveis) dentro de chaves { }.

```
function App() {  
  const nome = 'Marcelo'  
  return <p>{nome}</p>  
}
```

A close-up photograph of a large pile of yellow and white LEGO bricks, scattered and overlapping. The bricks are of various shapes and sizes, including standard 2x4 plates and 1x3 bricks. The lighting is soft, highlighting the texture of the plastic.

Componentes



Todos os direitos reservados
©2023 Realiz Educapão

Componentes

Podemos pensar em “*funções*” das quais podemos chamar e reutilizar diversas vezes desde que sejam passados os argumentos necessários. Isso torna o React poderoso e otimizado.

O ideal é dividir sua aplicação React em **diversos componentes**.



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

Em React, componentes são blocos de construção reutilizáveis e independentes que permitem dividir a interface do usuário em partes menores e mais gerenciáveis. Essas partes são representadas por elementos de interface (UI) que encapsulam lógica e renderização específicas. Componentes podem conter dados internos, chamados de estado (state), bem como aceitar entradas, chamadas de propriedades (props), que permitem a passagem de dados de um componente pai para um componente filho.

A abordagem baseada em componentes do React permite que os desenvolvedores criem interfaces complexas dividindo-as em componentes menores e mais fáceis de entender e manter. Esses componentes podem ser compostos juntos para criar uma aplicação inteira. Além disso, a reutilização de componentes em diferentes partes do aplicativo e até mesmo em projetos diferentes é uma prática comum, o que leva a um desenvolvimento mais eficiente e consistente.

Componentes de Classe

Antes da versão 16.8 do React ser lançada, todos os componentes eram baseados em classes.

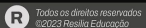
```
class Button extends React.component {  
  render() {  
    return <button>Comprar</button>  
  }  
}
```



Componentes de Classe

Esses componentes tinham vários métodos herdados de `React.component`, mas o **único obrigatório é o `render()`**, já que é o responsável pela **renderização dos elementos**.

Dica: salvar o arquivo do componente com a extensão `.jsx` trará vantagens como *snippets* (sugestões de código ao você digitar)



Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

Em React, os componentes de classe eram uma forma de definir componentes antes da introdução das funções de componente (também conhecidas como componentes funcionais) e dos hooks. Componentes de classe são classes JavaScript que estendem a classe `React.Component` e possuem um método obrigatório chamado `render()`, que retorna o código JSX que representa o componente.

A principal característica dos componentes de classe é que eles podem conter um estado interno (`state`) e métodos de ciclo de vida, como `componentDidMount`, `componentDidUpdate`, `componentWillUnmount`, entre outros. O estado permite que o componente reaja a mudanças e atualize sua renderização com base nessas alterações.

No entanto, com o avanço do React, os componentes de classe estão sendo gradualmente substituídos pelos componentes funcionais. Isso se deve ao fato de que os componentes funcionais são mais concisos, legíveis e têm melhor desempenho em muitos casos. Além disso, a introdução dos hooks permitiu que os componentes funcionais também tivessem estado e usassem métodos de ciclo de vida, tornando-os uma alternativa mais moderna e preferível aos componentes de classe.

Componentes Funcionais

Desde o lançamento da versão 16.8 do React, os componentes funcionais foram adotados e são fortemente sugeridos. Eles trazem diversas vantagens.

```
function Button() {  
  return <button>Comprar</button>  
}
```

NOTA PARA O ESTUDANTE

Em React, os componentes funcionais são uma forma mais simples e concisa de definir componentes. Eles são funções JavaScript que retornam elementos de interface (JSX) para representar a UI do componente. Ao contrário dos componentes de classe, os componentes funcionais não possuem estado interno (state) e não usam métodos de ciclo de vida.

Com o advento dos hooks no React, os componentes funcionais ganharam mais funcionalidades, permitindo que eles tenham estado (state) e usem métodos de ciclo de vida, caso necessário. Os hooks, como useState, useEffect, useContext e outros, possibilitam o gerenciamento de estado e a adição de funcionalidades de ciclo de vida aos componentes funcionais sem a necessidade de convertê-los em classes.

Os componentes funcionais são preferidos sempre que possível, porque são mais legíveis, mais fáceis de entender e têm melhor desempenho do que os componentes de classe. Além disso, a abordagem de usar hooks torna o desenvolvimento mais eficiente, permitindo que os desenvolvedores compartilhem lógica e reutilizem código de forma mais elegante. Como resultado, os componentes funcionais se tornaram a escolha padrão para criar componentes em React.

Componentes Funcionais

A sintaxe é bem mais curta e não precisamos estender ao `React.component`, o método `render()` já não é mais necessário, já que faz sentido que todo componente retorne elementos HTML.

```
function Button() {  
  return <button>Comprar</button>  
}
```



Todos os direitos reservados
©2023 Resilia Educação

Componentes Funcionais

Também podemos criar componentes funcionais com arrow functions.

```
function Button() {  
  return <button>Comprar</button>  
}  
→  
const Button = () => {  
  return <button>Comprar</button>  
}
```



Componentes Funcionais

É válido ressaltar que **todo nome de componente inicia com a letra maiúscula**. Essa foi uma convenção que permaneceu dos tempos do React com abordagens de componentes de classe.



Todos os direitos reservados
©2023 Realiza Educação

Atividade: Criar componente de formulário

→ O QUE É PRA FAZER?

Criar um formulário

→ COMO FAZER?

Em grupos:

- Criar uma pasta *components* na pasta *src*
- Criar um arquivo *Form.jsx*
- Esse arquivo terá:
 - 2 inputs (nome e e-mail)
 - Uma área de texto
 - Um botão para envio do formulário
- O componente deve ser chamado no componente *App.js*



Todos os direitos reservados



FECHAMENTO

Compartilhar as dúvidas e dificuldades encontradas

Props

As props são propriedades que podemos passar para componentes React, elas são somente de leitura e passamos elas via atributo.

```
function Button(props) {  
  const { text } = props  
  return (  
    <button>  
      {text}  
    </button>  
  )  
}
```



Todos os direitos reservados
©2023 Resilia Educativa

NOTA PARA O ESTUDANTE

Props, abreviação de "propriedades", são um mecanismo fundamental no React para passar dados de um componente pai para um componente filho. As props são utilizadas para transmitir informações e configurar o comportamento dos componentes.

As props são passadas para os componentes como atributos na sua declaração. No componente pai, você pode definir as props como valores estáticos ou passar valores dinâmicos através de expressões JavaScript. Dentro do componente filho, as props podem ser acessadas como propriedades do objeto `props`.

As props são imutáveis, o que significa que não devem ser modificadas diretamente dentro do componente filho. Elas são usadas principalmente para fornecer dados de um componente superior para um componente inferior, permitindo que o componente filho seja reutilizável e agnóstico em relação à fonte dos dados.

Além disso, as props podem ser usadas para transmitir funções ou callbacks dos componentes pais para os componentes filhos. Isso permite que os componentes filhos interajam com o componente pai, comunicando eventos ou atualizando o estado do componente pai.

As props são uma maneira de criar componentes flexíveis e modulares no React. Elas

ajudam a estabelecer uma hierarquia de componentes bem definida, facilitando o compartilhamento de dados e comportamentos entre os componentes.

Props

Outra forma de desestruturar as props é diretamente nos parênteses dos parâmetros.

```
function Button({ text }) {  
  return (  
    <button>  
      {text}  
    </button>  
  )  
}
```



Atividade: Criar componente de botão

→ O QUE É PRA FAZER?

Criar um componente de botão

→ COMO FAZER?

Em grupos:

- Dentro da pasta *components* criar um arquivo *Button.jsx*.
- Criar um objeto com as cores disponíveis no PDF de aula.
- O componente Button terá uma *prop* chamada *color*.
- A cor de fundo (*background color*) deve ser alterada de acordo com o valor da *prop color*.
- Substituir o botão do componente Form pelo novo componente Button.

→ FECHAMENTO

Compartilhar dúvidas e dificuldades encontradas

Atividade / Projeto de aula: Finanças+

Ao longo deste módulo as atividades de aula vão consolidar a realização de um projeto.

- ⇒ O projeto será um **aplicativo web** de gerenciamento financeiro pessoal, onde os usuários poderão acompanhar suas finanças, criar orçamentos, categorizar despesas e visualizar seus gastos ao longo do tempo.
- ⇒ O projeto será **construído ao longo das aulas**, cada atividade será um complemento do projeto.

NOTA PARA O ESTUDANTE

Link do figma:

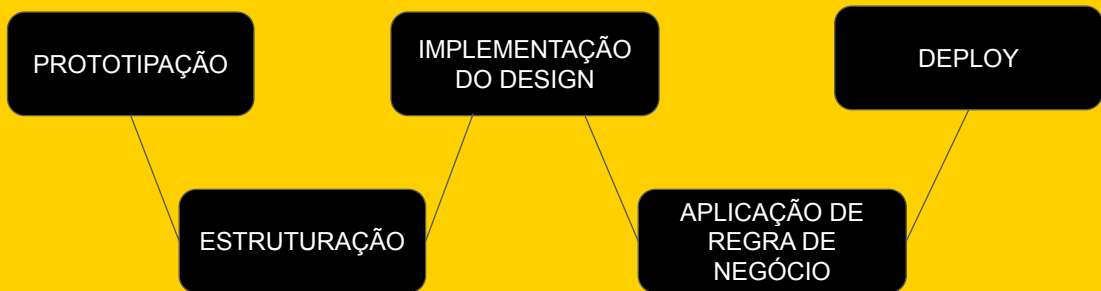
<https://www.figma.com/file/pt90KkTffFctZgWpzdiiBb/Finan%C3%A7as%2B?type=design&node-id=0%3A1&mode=design&t=jntRt6Wxib6AxSCs-1>

Tecnologias aplicadas

- ⇒ React.js
- ⇒ Vite
- ⇒ Styled Components
- ⇒ React Router DOM
- ⇒ Axios
- ⇒ TypeScript
- ⇒ Figma



Fases de desenvolvimento



NOTA PARA O ESTUDANTE

PROTOTIPAÇÃO: No dia a dia de trabalho como front-end você tem um designer que cuida de toda parte de prototipação (Desenho do projeto), aqui não será diferente, o protótipo do projeto está pronto. Nessa fase do projeto nós vamos entender como os devs usam o figma para facilitar o dia a dia de trabalho.

ESTRUTURAÇÃO: Fase em que criaremos a base do nosso projeto como o repositório, a estrutura React.js e a organização das pastas.

IMPLEMENTAÇÃO DO DESIGN: Essa é a fase em que construiremos toda parte visual da aplicação, estilo global, tema, telas e componentes.

APLICAÇÃO DE REGRA DE NEGÓCIO: Esse é o momento que damos vida a nossa aplicação, trazendo todas as funcionalidades através do javascript e API's.

DEPLOY: Com tudo pronto, esse é o momento de hospedar o nosso site em uma plataforma para que possamos compartilhar com as pessoas.

Vantagens e Objetivos

- ⇒ Ter um projeto completo (Back e front) no portfólio
- ⇒ Entender todas as fases de desenvolvimento de uma aplicação web
- ⇒ Aplicação dos conteúdos aprendidos em aula em um projeto real

- ⇒ Entender a rotina de um dev front-end
- ⇒ Ter como base o projeto de integração para o projeto final
- ⇒ Revisar e aplicar os conceitos dos módulos 1 e 2



Atividade: Construindo a base

→ O QUE É PRA FAZER?

Iniciar um projeto react, chamado Finanças+



Todos os direitos reservados
©2023 Resilia Educapio

Atividade: Construindo a base

→ COMO FAZER?

- ⇨ Crie um novo projeto vite react chamado **FinancasMais**
- ⇨ Instale a biblioteca **phosphor icons**
- ⇨ Exclua os arquivos que não são necessários (**App.jsx**, **app.css**, **index.css**, pasta **Assets**, **vite.icon**)
- ⇨ Dentro de **SRC** crie uma pasta **components** e uma pasta **pages**
- ⇨ Dentro da pasta **pages** crie uma pasta com o nome **LandingPage** e dentro da pasta crie um componente com o mesmo nome
- ⇨ Crie uma tag **h1** com o texto *"Hello Word"* no componente **LandingPage**
- ⇨ Import o componente **landingPage** no arquivo **main.jsx**
- ⇨ Rode o projeto


→ FECHAMENTO





Para ir além

- ⇒ [Creating a React App Using Vite.](#)
- ⇒ [Guide - ViteJS.](#)
- ⇒ [Hot Module Replacement.](#)
- ⇒ [Components and Props. ReactJS.](#)
- ⇒ [Introduzindo JSX. ReactJS.](#)

 **Dica:** Quando o artigo estiver em inglês, você pode utilizar a opção "Traduzir essa página", que fica na barra de ferramentas do Google Chrome.



Todos os direitos reservados
©2023 Resilia Educação





React, bota um cropped!

Aula 3 - Hard Skills



Wander Flor / Unsplash



Todos os direitos reservados
©2023 Resilia Educapio

Códigos
do Amanhã
2023 EBANX RESILIA



Objetivos de aprendizagem

- ⇒ Defender a importância de estruturar um negócio centrado no cliente.
- ⇒ Organizar perguntas para investigar perfil e necessidades do cliente.
- ⇒ Usar o mapa da empatia para definir perfil de cliente a partir de estudo de caso.



Todos os direitos reservados
©2023 Resilia Educação



Nesta aula, veremos:

- ⇒ O cliente no centro;
 - ⇒ Mapa de empatia;
 - ⇒ Aplicação prática para o contexto do dev.
-

Por que aprender isso?

- ⇒ Ao conhecer as necessidades e desejos do cliente por meio do mapa de empatia, o desenvolvedor pode projetar interfaces mais intuitivas, melhorar a usabilidade e a acessibilidade, além de personalizar a interação. Isso resulta em maior engajamento, fidelização e satisfação do cliente, levando ao crescimento do negócio e fortalecendo a imagem da empresa.
- ⇒ Colocar o cliente no centro das decisões de desenvolvimento garante produtos e serviços mais relevantes, alinhados com suas expectativas, e aumenta a competitividade no mercado digital.

“

**Ser a empresa mais focada no
cliente que existe no mundo.**

Missão da Amazon.com

”



Todos os direitos reservados
©2023 Resilia Educapio

O **cliente** no centro do negócio.



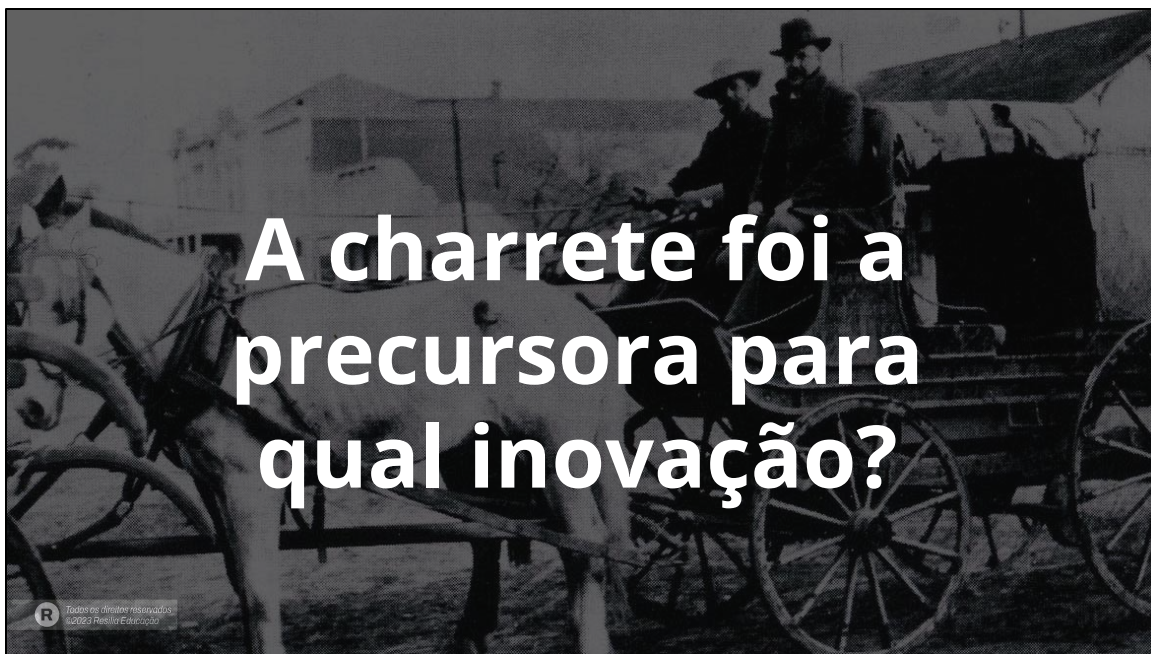
Todos os direitos reservados
©2023 Realiza Educação

NOTA PARA O ESTUDANTE

Colocar o cliente no centro do negócio é uma abordagem estratégica que coloca a satisfação e as necessidades do cliente como prioridade em todas as decisões e ações da empresa. É uma filosofia que reconhece que o sucesso de um negócio depende da criação de valor para o cliente e do estabelecimento de relacionamentos duradouros com eles. Aqui estão alguns aspectos importantes de ter o cliente no centro do negócio:

1. **Compreensão profunda do cliente:** Colocar o cliente no centro do negócio começa com uma compreensão profunda de quem são seus clientes, quais são suas necessidades, desejos, motivações e expectativas. Isso requer uma análise cuidadosa do mercado, pesquisa de mercado e interação direta com os clientes. Quanto melhor você conhecer seus clientes, mais eficazmente poderá atender às suas necessidades.
2. **Personalização da experiência do cliente:** Cada cliente é único e valoriza experiências personalizadas. Ao colocar o cliente no centro do negócio, você deve se esforçar para oferecer experiências personalizadas que atendam às

1. necessidades e preferências individuais de cada cliente. Isso envolve o uso de tecnologias e ferramentas de análise de dados para coletar informações sobre os clientes e adaptar suas ofertas e interações de acordo.
2. **Melhoria contínua da experiência do cliente:** Ter o cliente no centro do negócio significa estar comprometido com a melhoria contínua da experiência do cliente. Isso envolve ouvir ativamente o feedback dos clientes, analisar os pontos de dor e identificar áreas de oportunidade para aprimorar seus produtos, serviços e processos. A empresa deve se esforçar para superar as expectativas do cliente e fornecer um serviço excepcional em todos os pontos de contato.
3. **Construção de relacionamentos de longo prazo:** Colocar o cliente no centro do negócio envolve a construção de relacionamentos de longo prazo com os clientes. Em vez de se concentrar apenas em uma transação única, é importante investir em relacionamentos duradouros, cultivando a confiança e a fidelidade do cliente.





“

Se eu perguntasse aos meus
clientes o que eles queriam,
teriam me dito ‘um cavalo mais
rápido’.

Henry Ford

”



Todos os direitos reservados
©2023 Resilia Educaçáo

IMPORTANTE

O desafio da inovação é desenvolver uma compreensão mais profunda dos clientes e não simplesmente perguntar o que eles querem. Muitas vezes ele sabe o que quer mas não sabe o que precisa ou o que de fato vai resolver o seu problema.

COMO ENTENDER A REAL DOR DO CLIENTE?



Todos os direitos reservados
 2023 Resilia Educa o

Atividade: Mapa da empatia

→ O QUE É PRA FAZER?

Identificar qual solução podemos entregar para atender a uma expectativa ou necessidade de um cliente

→ COMO FAZER?

Em grupos, discutam e respondam:

Se vocês tivessem um negócio, quais seriam as perguntas que fariam para conhecer o seu cliente e suas necessidades?

→ FECHAMENTO

Registrar as respostas no menti

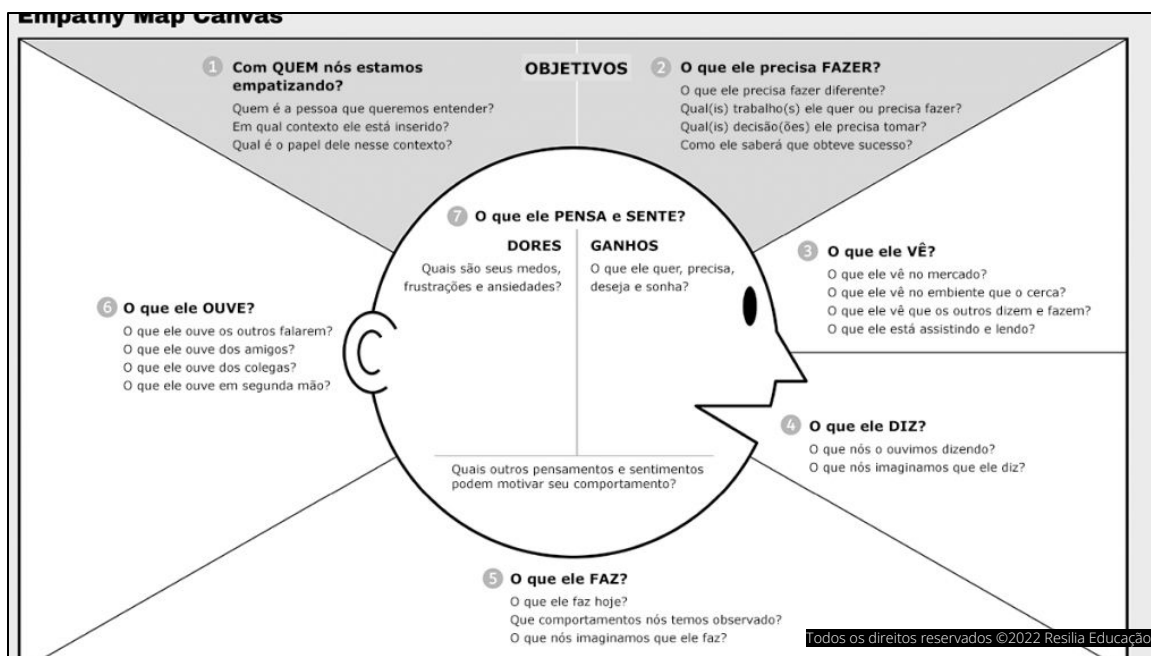


GLOSSÁRIO

Um mapa de empatia é uma ferramenta visual que ajuda a compreender melhor o cliente, suas necessidades, desejos, motivações e preocupações. É uma representação gráfica das características e emoções do cliente, permitindo que as empresas obtenham uma visão mais profunda de como o cliente pensa e sente. O mapa de empatia é geralmente dividido em quatro seções principais:

1. **O que o cliente vê:** Esta seção concentra-se no ambiente em que o cliente está imerso, incluindo o contexto social, físico e tecnológico em que ele opera. Isso pode incluir coisas como os lugares que o cliente frequenta, as pessoas com quem ele interage e as influências culturais e sociais que o afetam.
2. **O que o cliente ouve:** Nesta seção, considera-se o que o cliente escuta de outras pessoas, como amigos, familiares, colegas de trabalho, influenciadores ou especialistas. Também abrange a comunicação de marketing, opiniões de especialistas e experiências anteriores que podem moldar a perspectiva e as expectativas do cliente.
3. **O que o cliente pensa e sente:** Aqui, é importante compreender as necessidades, desejos, motivações e preocupações do cliente. Quais são

1. seus objetivos, aspirações e medos? Quais são suas principais preocupações e frustrações relacionadas ao produto ou serviço? Entender os pensamentos e sentimentos do cliente é fundamental para criar uma oferta que responda às suas necessidades emocionais e racionais.
2. **O que o cliente diz e faz:** Esta seção aborda as ações e comportamentos observáveis do cliente. O que ele diz publicamente? Quais são suas preferências declaradas? Como ele se comporta ao tomar decisões de compra? Isso ajuda a entender como o cliente se expressa e como suas ações podem estar alinhadas ou divergentes em relação às suas crenças e sentimentos.



NOTA PARA O ESTUDANTE

O mapa da empatia é uma ferramenta valiosa no contexto de negócios, que auxilia as empresas a compreenderem profundamente o perfil e as necessidades dos clientes. É uma representação visual que engloba quatro áreas-chave: pensar, sentir, ouvir e fazer do cliente, com o objetivo de criar empatia com eles.

Ao explorar as motivações, desejos, frustrações e pontos de dor dos clientes, as empresas podem projetar produtos e serviços mais alinhados com suas expectativas.

Essa abordagem centrada no cliente resulta em melhorias na experiência do usuário, aumentando a satisfação, lealdade e sucesso dos negócios no mercado competitivo.

Perguntas que podem ser respondidas

1

Estamos resolvendo o principal problema do cliente?

2

Nossa experiência atual desperta emoções positivas?

3

Como os clientes percebem o valor do nosso negócio?



Todos os direitos reservados
©2023 Realiza Educação

Mão na Massa: O cliente sempre tem razão!

→ O QUE É PRA FAZER?

Em grupos, vocês irão montar o mapa de empatia do cliente, no caso do Seu José.

→ COMO FAZER?

Ler o caso do Seu José apresentado a seguir e construir o mapa da empatia

→ FECHAMENTO

Compartilhar as ideias com a turma

Mão na Massa: O cliente sempre tem razão!

Caso:

Seu José é dono de uma empresa de e-commerce e contratou a sua equipe para resolver um problema: o seu serviço de suporte ao cliente.

José relatou que muitos clientes enviaram e-mails e até ligaram tentando resolver seus problemas, mas os profissionais responsáveis pelo suporte e o armazém onde estão os produtos se encontram em outro local, que não possui telefone.

Uma das sugestões de seu José para resolver esta necessidade, o cliente deveria se encaminhar fisicamente até lá, ou deixar sua reclamação por telefone, o que acabou gerando ainda mais frustração e reclamação para a empresa. E-mails se acumulavam na caixa de entrada da companhia e nunca eram lidos, respondidos ou resolvidos.



Para ir além

⇒ [7 passos para fazer um Mapa de Empatia para Negócios.](#)



Todos os direitos reservados
©2023 Resilia Educação





Até a próxima e
#confianoprocesso



Todos os direitos reservados
©2023 Resilia Educação

Códigos
do Amanhã
2023 EBANX RESILIA