

```
import React, { useState } from "react";
import { Text } from "react-native";
import Filho from "../Filho";
```

/\* COMUNICAÇÃO INDIRETA:

Comunicação Indireta é um conceito que faz parte da funcionalidade do React que também é importante de ser entendido.

Esse conceito explica como funciona a passagem de parâmetros de um Componente Filho para um Componente Pai.

PRIMEIRAMENTE O QUE É UM COMPONENTE PAI E UM COMPONENTE FILHO?

Um Componente Pai é um Componente que importa um Componente, usa uma instância dele no seu código e também passa parâmetros para esse Componente Filho.

Um Componente Filho é um Componente que é importado por um Componente Pai, que tem uma instância gerada em algum determinado Componente Pai e que recebe parâmetros do Componente Pai.

POR QUE RECEBE O NOME DE COMUNICAÇÃO INDIRETA?

É chamada assim por que essa comunicação acontece de forma escondida entre o Pai e o Filho, pois o filho não faz "import" do Componente Pai e nem gera instâncias dele. Mas ainda assim é possível que um Componente Filho passe parâmetros para um Componente Pai.

COMO ACONTECE A COMUNICAÇÃO INDIRETA?

Ela é feita sempre que um Componente Pai recebe um parâmetro entregue por um Componente Filho e faz uso desse parâmetro.

Para isso o Componente Pai precisa usar Hooks que recebam os dados do Componente Filho por através de Estados, para que possa renderizar esses dados. Para isso uma função "set" passada por através do Hook, deverá ser passada como parâmetro para o Componente Filho, irá usá-la e fará a atualização dos dados no Componente Pai.

A COMUNICAÇÃO INDIRETA SEMPRE ACONTECE QUANDO:

- \* É de Componente Filho para Pai;
- \* É escondida e é feita por através de mudança de Estados;
- \* O Componente Filho tem que receber a função "set" do Hook;
- \* O Componente Pai faz uso do valor que foi atualizado no Componente Filho;

Abaixo temos um exemplo de comunicação Indireta entre Componente Pai e Filho...

```
*/  
  
export default ({nome, letra}) => {  
  
  //Note que o componente pai está usando um useState onde o valor do nome do Filho será atualizado...  
  const [filho, setFilho] = useState('')  
  //Uma função que usa o setFilho é passada ao Filho...  
  const voltaNome = (letra) => setFilho(letra)  
  
  return (  
    <>  
      <Text>O nome do Pai é {nome}</Text>  
      <Filho pai={nome} letra={letra} funcao={voltaNome} />  
      {/* Note que quem faz uso do valor atualizado é o Pai... */}  
      <Text>O nome devolvido pelo meu filho é {filho}</Text>  
    </>  
  )  
}
```

ARQUIVO DO COMPONENTE FILHO...

```
import React from "react";  
import { Text } from "react-native";  
  
export default ({pai, letra, funcao}) => {
```

```

    //Aqui o Componente Filho faz uso da função que utiliza o set do useState e atualiza os valores, passando dados para o Componente
    Pai por através de mudança de Estados.
    const geraNome = (letra) => {
        const nomeCompleto = `${letra}esus`
        funcao(nomeCompleto)
    }

    geraNome(letra)

    return (
        <>
            <Text>Eu sou o Filho de {pai}.</Text>
            <Text>A primeira letra passada por Ele foi {letra}</Text>
        </>
    )
}

```

ARQUIVO DO APP.JS...

```

import React from 'react'
import ComunicacaoIndireta from './src/fundamentos/ComunicacaoIndireta'

export default () => {
    return (
        /* Coloque seus componentes abaixo...*/
        <ComunicacaoIndireta nome={"Jeová"} letra={"J"} />
    )
}

```

RESULTADO NO CELULAR...

The image shows a development environment with a mobile emulator on the left and a code editor on the right. The emulator displays the text: "O nome do Pai é Jeová", "Eu sou o Filho de Jeová.", "A primeira letra passada por Ele foi J", and "O nome devolvido pelo meu filho é Jesus". The code editor is Visual Studio Code, showing a file explorer on the left with a project structure including "componentes" and "App.js". The main editor displays the code for "App.js", which imports React and ComunicacaoIndireta, and exports a default function. The terminal at the bottom shows the command "node" and the output "Connecting to the development server...", "Starting the app on 'emulator-5554'...", and "Starting: Intent { cmp=com.componentes/.MainActivity }".

USESTATE - MODIFICANDO E SALVANDO ESTADOS DOS DADOS DOS COMPONENTES - Word (Product Activation Failed)

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

Appjs - componentes - Visual Studio Code

EXPLORADOR

COMPONENTES

- exercicios
- fundamentos
  - ComunicacaoDireta
    - JS Filho.js U
    - JS index.js U
  - ComunicacaoIndireta
    - JS Filho.js U
    - JS index.js U
- EstadosNoReact

.buckconfig  
.eslintrc.js  
.flowconfig  
.gitignore  
.node-version  
.prettierrc.js  
.ruby-version  
.watchmanconfig  
JS App.js M  
app.json  
babel.config.js  
Gemfile  
JS index.js  
metro.config.js  
package.json  
yarn.lock

JS App.js > ...

```
1 import React from 'react'
2 import ComunicacaoIndireta from './src/fundamentos/ComunicacaoIndireta'
3
4 export default () => {
5   return (
6     /* Coloque seus componentes abaixo... */
7     <ComunicacaoIndireta nome={"Jeová"} letra={"J"} />
8   )
9 }
10
```

PROBLEMAS 2 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

node + - x

info Connecting to the development server...  
info Starting the app on "emulator-5554"...  
Starting: Intent { cmp=com.componentes/.MainActivity }  
8 actionable tasks: 2 executed, 36 up-to-date

PAGE 4 OF 4 520 WORDS PORTUGUESE (BRAZIL)

master 2 0

Ln 10 Col 1 (238 selecionado) Espacos: 2 UTF-8 LF JavaScript Go Live

25°C 09:49 03/12/2022