

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

```
/*  COMUNICAÇÃO DIRETA:
```

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

Esse conceito explica como funciona uma comunicação de ligação e passagem de parâmetros entre de um Componente Pai para um Componente Filho.

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 DIRETA?

Quando falamos de Comunicação Direta estamos nos referindo a Comunicação que acontece de um Componente Pai para um Componente Filho. Ela é direta por que ela acontece de forma explícita entre o Pai e o Filho, o "import" não deixa dúvidas de que a comunicação está acontecendo e a passagem de parâmetros é feita também de forma explícita, fica óbvio que ela está acontecendo.

A COMUNICAÇÃO DIRETA SEMPRE ACONTECE QUANDO:

- \* É de Componente Pai para Filho;
- \* É explícita por através do import e da passagem de parâmetros;

Abaixo temos um exemplo de comunicação direta entre elemento Pai e Filho...

```
*/
```

```
//ComunicacaoDireta é um elemento Pai, por que ele importa Filho, gera uma instância dele e passa parâmetros para ele...
export default ({pai, filho}) => {
  return (
    <>
      <Text>Eu sou o Pai, meu nome é {pai}. Meu filho vai se apresentar...</Text>
      <Filho nome={filho} pai={pai} />
    </>
  )
}
```

ARQUIVO FILHO...

```
import React from "react";
import {Text} from 'react-native';

//Veja que filho não importa nenhum componente, ele na verdade só passa componentes...
export default ({nome, pai}) => {
  return (
    <Text>Eu sou filho de {pai}, meu nome é {nome}, esse nome foi dado pelo meu pai.</Text>
  )
}
```

ARQUIVO APP.JS

```
import React from 'react'
import ComunicacaoDireta from './src/fundamentos/ComunicacaoDireta'

export default () => {
  return (
    /* Coloque seus componentes abaixo...*/
    <ComunicacaoDireta pai={"Jeová"} filho={"Jesus Cristo"} />
  )
}
```

## 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 a simple text-based interface with two lines of text: "Eu sou o Pai, meu nome é Jeová. Meu filho vai se apresentar..." and "Eu sou filho de Jeová, meu nome é Jesus Cristo, esse nome foi dado pelo meu pai." The code editor is Visual Studio Code, showing a project named "Appjs" with a file explorer on the left. The file explorer shows a directory structure with "componentes" and "fundamentos" folders. The "fundamentos" folder is expanded, showing "ComunicacaoDireta" and "Filho.js". The "ComunicacaoDireta" file is selected, and its content is displayed in the editor. The code is a React component that imports "React" and "ComunicacaoDireta" and renders a simple UI with a "pai" and "filho" prop. The terminal at the bottom shows the output of the application, including the message "Connecting to the development server..." and "Starting the app on 'emulator-5554'...".

Visual Studio Code interface showing the project structure and the code for the "ComunicacaoDireta" component.

Project Structure (EXPLORADOR):

- componentes
  - xtend
  - y18n
  - yallist
  - yargs
  - yargs-parser
  - ycto-queue
  - .yarn-integrity
- src
  - componentes
  - estilos
  - exercicios
  - fundamentos
    - ComunicacaoDireta
      - Filho.js
      - index.js
    - EstadosNoReact
  - .buckconfig
  - .eslintrc.js
  - .flowconfig
  - .gitignore
  - .node-version
  - .prettierrc.js
  - .ruby-version
  - .watchmanconfig

Code in "ComunicacaoDireta" (JS Appjs):

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

Terminal Output:

```
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
```