VOCÊ SABE MESMO DAR NOMES PARA SUAS FUNÇÕES?

```
//❌ ERRADO
const arrayUsers = fetchingUsers()
```

```
//✅ CORRETO
const { users } = getUsers()
```

```
// ✗ ERRADO
const arrayUsers =
```

Redundância

A tipagem já mostra que é um array.

Semântica ambígua

O nome não indica claramente o propósito

```
= fetchingUsers()
```

Ao utilizar algo como **fetchingUsers**, você acaba abrindo espaço para a criação de funções como...

```
retrieveProducts()
fetchPurchaseHistory()
```

```
getUsers()
retrieveProducts()
fetchPurchaseHistory()
```

Observe que todos realizam buscas, porém não há um padrão definido para a nomenclatura.

✅

```
getUsers()
getProducts()
getPurchaseHistory()
```

Agora, alteramos tudo para usar o prefixo do verbo "get", o que resultou em um código mais padronizado e claro

✅

```
const { users }
const { products }
const { userPurchaseHistory }
```

Agora, vamos alterar o retorno para um objeto, utilizando nomes que facilitem o seu uso e eliminem redundâncias como "users", "products", etc.

```
function getUsers() {}
export default getUsers

import fetchUser from './';
```

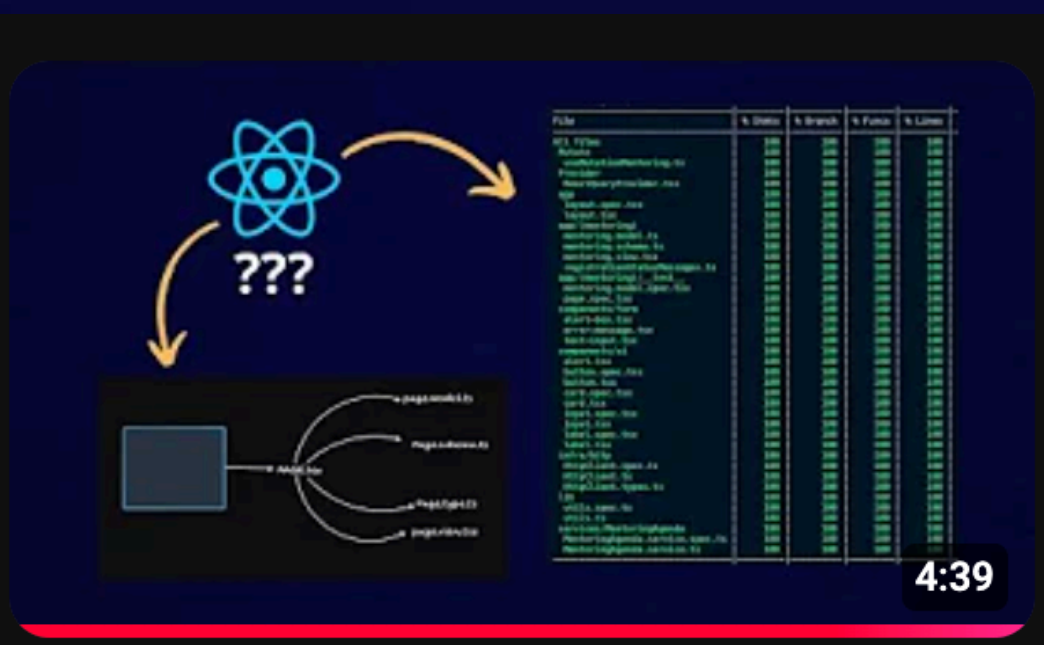
Para evitar a necessidade de renomeação, evite o uso de export default.

✅

```
export function getUsers() {}
import { getUsers } from './';
const { users } = getUsers()
```

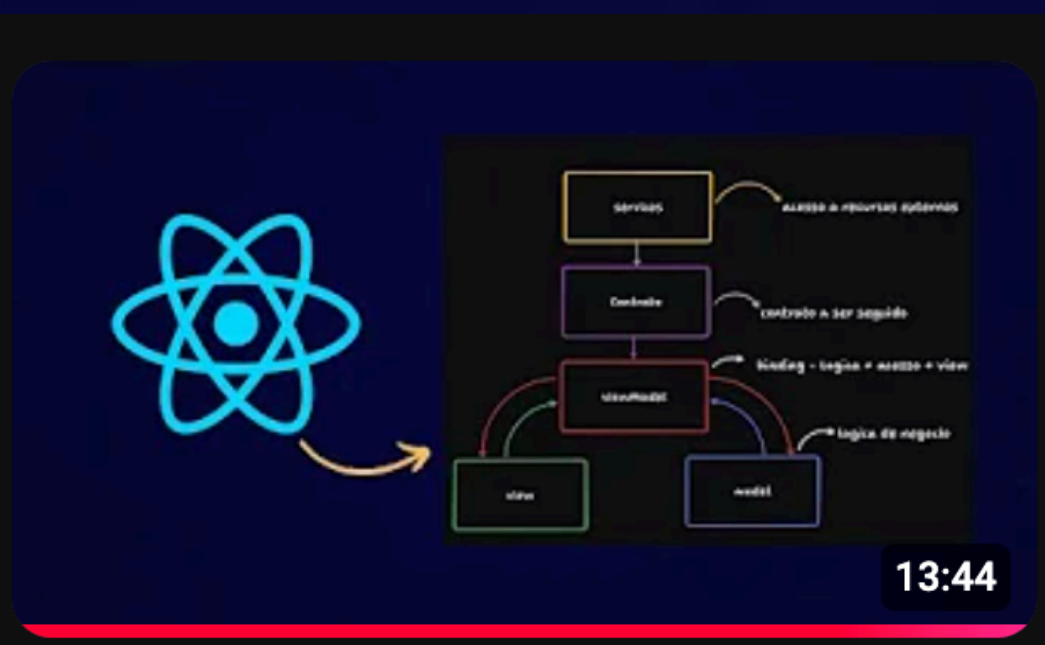
Agora, além de utilizarmos o export, obtemos um retorno mais claro e objetivo.

FIQUE LIGADO NO CANAL PARA MAIS CONTEÚDOS.



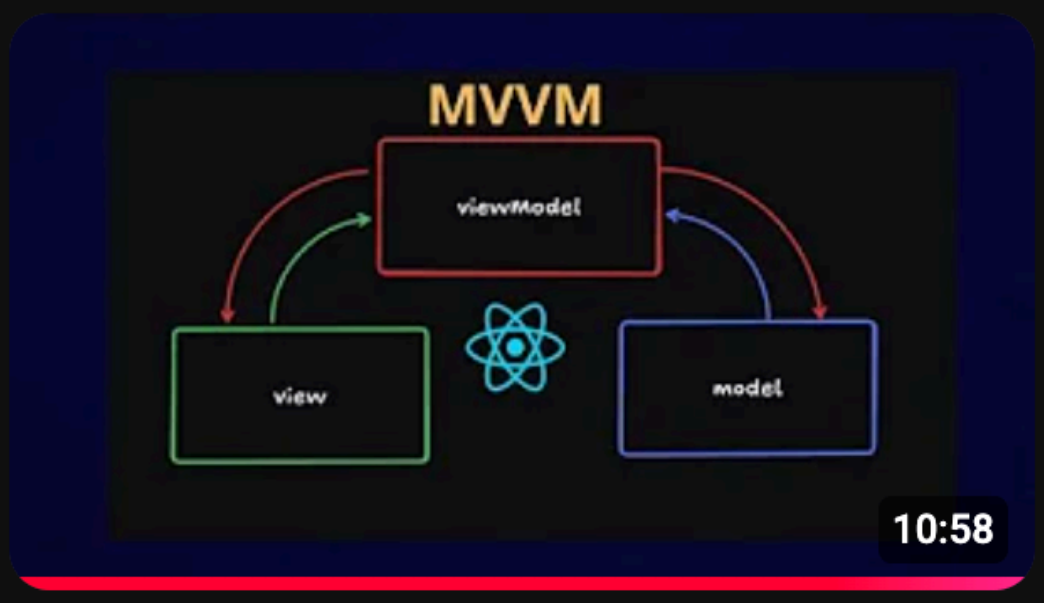
Arquitetura Front-end: Construindo um Arquitetura Testável em React.js

2,1 mil visualizações • há 1 mês



Arquitetura Front-end: Inversão de Dependência em React.js (abstraindo...

1,8 mil visualizações • há 1 mês



Arquitetura Front-end: Aplicando Arquitetura MVVM no Front-end (React.js)

1,6 mil visualizações • há 1 mês

Por Que Todo Mundo Cresce e Eu Não?

822 visualizações • há 12 dias

Gostou?

Curta

Compartilhe

Salve