



# Sente a resposta

Módulo 5 - Aula 4 - Hard



Fonte: Pixabay



Todos os direitos reservados  
©2022 Resilia Educação

A person is sitting at a desk, writing on a document with a pen. They are wearing a red sweater and a watch. A brown leather bag is on the desk next to them. The background is slightly blurred, showing a computer monitor and some papers.

# Problema da responsabilidade



# Engenharia de software - S.R.P.

---

O princípio de responsabilidade única diz que **uma classe/arquivo/função** deve fazer **uma coisa e apenas uma coisa!**

Essa abordagem **diminui o acoplamento** e aumenta a possibilidade de reutilização de código!

# Problema da responsabilidade

---

Hoje no nosso projeto **TODO app API** temos **um arquivo** tratando **duas rotas** de **entidades distintas**.

Em termos de **entendimento e legibilidade** do código isso pode ser um dificultador.

# Recap MVC



Todos os direitos reservados  
©2022 Resilia Educação

Todos os direitos reservados ©2022 Resilia Educação



# Controllers



A hand is holding a camera lens, with the lens's viewfinder showing a blurred landscape of a lake and mountains. The word "Views" is written in large, bold, yellow letters across the center of the lens.

# Views





# Models



Todos os direitos reservados  
©2022 Resilia Educação



A hand holding a camera lens is shown against a dark, blurred background. A large, bright red 'X' is drawn across the entire image, crossing out the word 'Views' which is written in a bold, yellow, sans-serif font in the center of the lens.

# Views



Todos os direitos reservados  
@Brasil Educação

A dark, high-contrast image showing a person in a camouflage hoodie on the left, throwing a brick towards the head of a muscular, bald man on the right. The man has a pained or aggressive expression, with his hand near his ear. The background is a dark, textured wall. The title text is overlaid in the center.

# Quebrando em arquivos



Todos os direitos reservados  
©2022 Resilia Educação

# Node - Exportando módulos

---

Para exportar módulos no node, podemos utilizar:

`module.exports = <aquilo que queremos exportar>`

Ou então trabalhar com a exportação no formato ESM:

`export default <aquilo que queremos exportar>`

# Node - Importando módulos

---

Para **importar conteúdos** (funções/classes/objetos) criados por nós mesmos em outros arquivos (com **module.exports**) utilizamos:

```
const nomeVariavel = require('path/caminho do arquivo');
```

Enquanto isso para **importar** com ESM utilizamos:

```
import OQueEstamosImportando from "path/caminho do arquivo";
```

A background image showing a hand placing a wooden block on a Jenga tower. The scene is dimly lit, with a warm, yellowish light source in the background. The Jenga tower is made of light-colored wooden blocks, and the hand is positioned at the top, about to place another block. The overall mood is one of focus and precision.

# Deconstructing





# Node - Destructuring

---

Destructuring (desestruturação) é um método de **importação de campos de objetos**. Quando um objeto é exportado em um módulo podemos importar apenas alguns atributos dele utilizando a notação abaixo:

```
const {appName, appPort} = require('./app-config');
```

```
import { appName, appPort } from './app-config';
```

# Atividade: Organizando o projeto

Na pasta do projeto TODO app API crie as seguintes pastas e arquivos seguindo a estrutura fornecida:

- ⇒ Pasta src - Onde todo código da aplicação será criado
- ⇒ Pastas models e controllers dentro da pasta src
- ⇒ Mova o arquivo com o app/servidor para a pasta src
- ⇒ Crie os arquivos usuario-controller.js e tarefa-controller.js dentro da pasta controller
- ⇒ Transfira as rotas de usuario e tarefa presentes no app/servidor para os arquivos correspondentes de controller
- ⇒ Crie nos arquivos do controller uma função que recebe como parâmetro o app e configura as rotas
- ⇒ Importe os controllers no app/servidor e utilize a função importada para configurar as rotas



# Atividade em pares

