

Por que preciso separar meu projeto por responsabilidades?



Rocketseat © 2025

Todos os direitos reservados

rocketseat.com.br

Introdução

O que seria a separação
de responsabilidades?



Introdução

O que seria a separação de responsabilidades?

- Dividir o sistema em partes, cada uma com um único objetivo.



Introdução

O que seria a separação de responsabilidades?

- Dividir o sistema em partes, cada uma com um único objetivo.
- Cada componente é responsável por uma função específica, facilitando a organização e a manutenção.



Introdução

O que seria a separação de responsabilidades?

- Dividir o sistema em partes, cada uma com um único objetivo.
- Cada componente é responsável por uma função específica, facilitando a organização e a manutenção.
- Foco na modularidade e na redução do acoplamento.



Introdução

O que seria a separação de responsabilidades?

- Dividir o sistema em partes, cada uma com um único objetivo.
- Cada componente é responsável por uma função específica, facilitando a organização e a manutenção.
- Foco na modularidade e na redução do acoplamento.
- A modularidade permite que os componentes sejam independentes, minimizando dependências entre eles.



Introdução

O que seria a separação de responsabilidades?

- Dividir o sistema em partes, cada uma com um único objetivo.
- Cada componente é responsável por uma função específica, facilitando a organização e a manutenção.
- Foco na modularidade e na redução do acoplamento.
- A modularidade permite que os componentes sejam independentes, minimizando dependências entre eles.
- Exemplo: camadas de Apresentação, Negócios e Dados (Essa separação clara entre interface, lógica de negócios e persistência garante um sistema mais robusto e fácil de evoluir.



O problema ao separar por responsabilidade

O problema ao separar por responsabilidade

- **Dificuldade de manutenção e expansão:** Quando o código não é organizado ou modularizado, pequenas mudanças em uma funcionalidade podem exigir alterações em várias partes do sistema.

O problema ao separar por responsabilidade

- **Dificuldade de manutenção e expansão:** Quando o código não é organizado ou modularizado, pequenas mudanças em uma funcionalidade podem exigir alterações em várias partes do sistema.
- **Alto risco de quebras inesperadas:** Em sistemas com alto acoplamento, modificar um componente pode impactar outros diretamente dependentes, gerando bugs e falhas críticas.

O problema ao separar por responsabilidade

- **Dificuldade de manutenção e expansão:** Quando o código não é organizado ou modularizado, pequenas mudanças em uma funcionalidade podem exigir alterações em várias partes do sistema.
- **Alto risco de quebras inesperadas:** Em sistemas com alto acoplamento, modificar um componente pode impactar outros diretamente dependentes, gerando bugs e falhas críticas.
- **Código duplicado:** A ausência de organização e reutilização resulta em repetição de código, aumentando o tamanho do sistema.

O problema ao separar por responsabilidade

- **Dificuldade de manutenção e expansão:** Quando o código não é organizado ou modularizado, pequenas mudanças em uma funcionalidade podem exigir alterações em várias partes do sistema.
- **Alto risco de quebras inesperadas:** Em sistemas com alto acoplamento, modificar um componente pode impactar outros diretamente dependentes, gerando bugs e falhas críticas.
- **Código duplicado:** A ausência de organização e reutilização resulta em repetição de código, aumentando o tamanho do sistema.
- **Código difícil de entender:** O excesso de duplicação e falta de estrutura tornam o código confuso, elevando o esforço e o custo para manutenção.

Benefícios da organização



Benefícios da organização

Facilidade de manutenção: Alterações são localizadas, afetando apenas a parte relevante do sistema e reduzindo o risco de erros em áreas não relacionadas.



Benefícios da organização

Facilidade de manutenção: Alterações são localizadas, afetando apenas a parte relevante do sistema e reduzindo o risco de erros em áreas não relacionadas.

Escalabilidade: Partes independentes podem crescer separadamente, permitindo que cada componente seja otimizado ou ampliado sem impactar o restante do sistema.



Benefícios da organização

Facilidade de manutenção: Alterações são localizadas, afetando apenas a parte relevante do sistema e reduzindo o risco de erros em áreas não relacionadas.

Escalabilidade: Partes independentes podem crescer separadamente, permitindo que cada componente seja otimizado ou ampliado sem impactar o restante do sistema.

Reutilização: Componentes bem projetados podem ser reaproveitados em diferentes sistemas, com pouca ou nenhuma modificação.



Benefícios da organização

Facilidade de manutenção: Alterações são localizadas, afetando apenas a parte relevante do sistema e reduzindo o risco de erros em áreas não relacionadas.

Escalabilidade: Partes independentes podem crescer separadamente, permitindo que cada componente seja otimizado ou ampliado sem impactar o restante do sistema.

Reutilização: Componentes bem projetados podem ser reaproveitados em diferentes sistemas, com pouca ou nenhuma modificação.

Testabilidade: Cada camada pode ser testada isoladamente, facilitando a identificação e correção de erros.

