

Camada de negócios



Rocketseat © 2025

Todos os direitos reservados

rocketseat.com.br

Introdução

O que seria a camada de negócios?

A camada de negócios é responsável por implementar as regras do domínio da aplicação. Ela atua como um intermediário lógico entre a entrada do usuário (View/Controller) e o acesso a dados (Model).



Responsabilidades

Responsabilidades

Validar dados: Certifica que entradas do usuário ou de APIs externas estão corretas.

- Exemplo: verificar se o valor de um desconto não excede 50%.

Responsabilidades

Aplicar regras:

- Implementa a lógica definida pelo domínio do negócio.
 - Exemplo: aplicar a política de frete grátis para pedidos acima de um valor específico.

Responsabilidades

Manter consistência:

- Garante que o sistema funcione de forma confiável e segura.
 - Exemplo: verificar o estoque disponível antes de aprovar um pedido.

Responsabilidades

Coordenar processos:

- Orquestra fluxos que envolvem várias etapas e decisões.
 - Exemplo: gerenciar o processo de aprovação de um empréstimo bancário.

Boas práticas para a camada de negócios:

Centralize a lógica: Mantenha as regras de negócio concentradas em métodos específicos, evitando duplicação.

Exemplo: um único método responsável por calcular preços com descontos.

Boas práticas para a camada de negócios:

Evite dependências excessivas: A camada de negócios deve depender apenas do Model e do Controller, sem incluir lógica de apresentação.

Exemplo: evitar formatações de strings dentro dessa camada.

Boas práticas para a camada de negócios:

Reutilize funcionalidades: Escreva funções genéricas que possam ser reaproveitadas em diferentes partes do sistema.

Exemplo: validações reutilizáveis, como CPF ou CNPJ.

Boas práticas para a camada de negócios:

Mantenha a testabilidade: Estruture a lógica em métodos pequenos, coesos e modulares, que possam ser testados de forma isolada.

Exemplo: Em um sistema de e-commerce, um único método centralizado na camada de negócios calcula o preço final com desconto, garantindo que clientes VIP recebam 10% de desconto sem ultrapassar o limite de 50%, mantendo a lógica reutilizável, testável e livre de formatações.

Como ela se comunica com outras camadas?

- **Transformar dados:** Pode transformar e formatar os dados retornados pelo Model antes de enviá-los ao cliente, além de padronizar as respostas para facilitar a integração com diferentes consumidores.
- **Tratamento de erros:** Deve apresentar mensagens de erro de forma clara e compreensível, evitando exposições técnicas que confundam o usuário.
- **Validação de entradas:** No backend, pode realizar validações básicas nos dados recebidos antes de repassá-los ao Controller ou ao Model, garantindo que entradas inválidas sejam tratadas antecipadamente.