

Regras de Negócio e Funcionalidades

Person	https://github.com/kellen-xavier/qa-readme
Status	Review de Leitura 📖

O que é uma regra de negócio?

Regra de negócio refere-se a uma diretriz ou condição específica que governa o comportamento do sistema, baseada nas **necessidades e processos do negócio** para o qual o software está sendo desenvolvido. Essas regras definem como as funcionalidades do sistema devem operar de acordo com as particularidades da organização ou do domínio que o software está atendendo.

Exemplo de Regra de Negócio:

Imagine que você está desenvolvendo um sistema bancário.

Uma regra de negócio seria:

No caso de uma **locadora de carros**, as regras de negócio são todas as regras que ditam **como o aluguel de carros** deve acontecer, quais as condições para o aluguel, os prazos, as restrições e os comportamentos esperados.

Como distinguir regras de negócio, funcionalidades e dados?

- **Regra de negócio:** Descreve o **comportamento lógico** que o sistema precisa seguir de acordo com as necessidades do negócio.

Exemplo: "Um carro só pode ser alugado por um cliente que tenha uma carteira de motorista válida."

- **Funcionalidade:** Refere-se às **ações** ou **operações** que o sistema deve permitir, baseadas nas necessidades dos usuários.

Exemplo: "O sistema deve permitir que o cliente faça reservas de veículos."

- **Dados:** São as **informações** que o sistema armazena e gerencia.

Exemplo: Informações sobre os carros disponíveis, detalhes dos clientes, históricos de aluguel, etc.

Explicando com um exemplo de Sistema de Locadora de Carros

Imagine um sistema para uma locadora de carros que deve gerenciar o processo de reserva e aluguel de veículos. Vou dividir os conceitos para que fique fácil entender a diferença entre **regra de negócio, funcionalidade e dados**.

Regra de Negócio

O que é? : É uma condição ou restrição que a locadora definiu para que o processo de aluguel aconteça de forma adequada.

Exemplos de Regras de Negócio em uma Locadora de Carros:

- **Reserva de Carro:** "Um cliente só pode reservar um carro se ele não tiver dívidas pendentes."
- **Disponibilidade:** "O carro só pode ser alugado se estiver disponível no momento da reserva."

- **Período de Aluguel:** "O aluguel do carro deve ser feito por um período mínimo de 1 dia e máximo de 30 dias."
- **Manutenção:** "Carros em manutenção não podem ser alugados."

Essas regras de negócio definem **como o sistema se comporta e quais condições precisam ser atendidas** para que as funcionalidades funcionem corretamente.

Funcionalidade

O que é?: São as **ações** ou **operações** que o sistema permite o usuário realizar. Elas são os meios pelos quais as regras de negócio são aplicadas no sistema.

Exemplos de Funcionalidades em uma Locadora de Carros:

- **Cadastrar clientes:** O sistema deve permitir que novos clientes se cadastrem com suas informações pessoais.
- **Reservar carros:** O sistema deve permitir que clientes façam a reserva de carros online.
- **Verificar disponibilidade:** O sistema deve mostrar ao cliente os carros disponíveis em uma data específica.
- **Emitir relatórios:** O sistema deve permitir que a administração gere relatórios de uso dos carros (histórico de aluguéis, manutenção, etc).

Essas funcionalidades são **ações** que o sistema executa para **implementar as regras de negócio** e permitir a interação do usuário com o sistema.

Dados

O que é?: São as informações que o sistema armazena e manipula. Esses dados permitem que as funcionalidades funcionem e que as regras de negócio

sejam aplicadas corretamente.

Exemplos de Dados dessa Locadora de Carros:

- **Carros:** Informações sobre cada carro, como modelo, placa, status (disponível, alugado, em manutenção).
- **Clientes:** Informações sobre os clientes, como nome, CPF, carteira de motorista, histórico de aluguéis.
- **Aluguéis:** Registros dos aluguéis feitos, contendo informações como data de início, data de término, carro alugado, cliente.
- **Manutenção:** Dados sobre quando e quais carros passaram por manutenção.

Esses dados são

armazenados no banco de dados e usados para verificar as regras de negócio e realizar as funcionalidades do sistema.

Exemplo Prático: Aluguel de Carro

Vamos ver um exemplo completo para entender como tudo se conecta.

Cenário: Um cliente quer alugar um carro

Funcionalidade: O cliente acessa o sistema e faz uma solicitação para alugar um carro.

- O sistema tem a funcionalidade de **fazer reservas de carro**.

Dados: O sistema consulta as informações do carro e do cliente.

- Verifica se o carro está disponível (consulta os dados de disponibilidade dos carros).
- Verifica se o cliente está com a **carteira de motorista válida** e sem **dívidas pendentes** (dados do cliente).

Regra de Negócio: Se o carro estiver disponível e o cliente cumprir as condições, o sistema permite o aluguel.

- A regra de negócio exige que o carro esteja disponível e que o cliente atenda às condições (validade da CNH e sem dívidas).

Resultado: Se todas as condições forem atendidas, o carro é alugado. Caso contrário, uma **exceção de negócio** é lançada, informando o cliente sobre a restrição (por exemplo, dívida pendente ou CNH inválida).

Recapitulando:

- **Regra de Negócio:** São as condições ou políticas que governam o comportamento do sistema, diretamente ligadas aos processos do negócio (como "o carro deve estar disponível para ser alugado").
- **Funcionalidade:** São as operações que o sistema executa para permitir que o usuário interaja com o sistema e realize ações (como "permitir que o cliente faça reservas").
- **Dados:** São as informações armazenadas no sistema, que são usadas para verificar as regras de negócio e realizar as funcionalidades (como "os dados do cliente e dos carros").

▲ Referência para estudos ▲

Importante de entender esse conceito, pois dentro do desenvolvimento e engenharia de software vamos ter que lidar com isso no dia dia. Segue abaixo alguns vídeos e conteúdos focado nessa explicação para praticar e treinar.

Links abaixo:

[137 - Clean Architecture: Regras de Negócio de Domínio ou Aplicação?](#)

86 - Clean Architecture | Camada de Domínio vs. Camada de Casos de Uso

111 - Banco de Dados na Clean Architecture | theWiseDev + Node.js + TypeScript