

Construção do modelo de classes de análise (modelo de domínio)

Conceitos, técnicas, exemplos práticos e aplicação

1. O que é o modelo de classes de análise?

O modelo de classes de análise, também chamado de modelo de domínio, é a primeira representação estrutural do sistema, construída logo após (ou junto com) o levantamento de requisitos. Ele identifica os principais conceitos do negócio — chamados de classes de análise — e como eles se relacionam entre si, sem se preocupar ainda com detalhes técnicos de implementação, como linguagem de programação, banco de dados ou padrões de arquitetura.

Construir esse modelo é uma das atividades centrais da fase de análise orientada a objetos, servindo como ponte entre os requisitos levantados junto ao cliente e as decisões técnicas que virão nas fases de projeto (design) e desenvolvimento.

2. Tipos de classes de análise

Uma técnica amplamente utilizada (originada no processo unificado, RUP) classifica as classes de análise em três estereótipos, cada um com um papel diferente no sistema:

2.1 Classes de entidade (Entity)

Representam os conceitos e dados persistentes do negócio, ou seja, informações que precisam ser armazenadas e que normalmente sobrevivem além de uma única execução do sistema.

Exemplo prático: em um sistema de locação de veículos, as classes Cliente, Veículo e Contrato de Locação são classes de entidade, pois representam dados que precisam ser guardados mesmo depois que o sistema for reiniciado.

2.2 Classes de fronteira (Boundary)

Representam a interface entre o sistema e o mundo externo, seja um ator humano (telas, formulários) ou outro sistema (interfaces de integração, APIs).

Exemplo prático: no mesmo sistema de locadora, a classe TelaDeReserva representa a interface onde o atendente registra uma nova locação, enquanto InterfaceGatewayPagamento representa a comunicação com um serviço externo de processamento de cartão de crédito.

2.3 Classes de controle (Control)

Representam a lógica e a coordenação de um caso de uso específico, funcionando como uma espécie de "maestro" que orquestra a comunicação entre classes de fronteira e classes de entidade.

Exemplo prático: a classe `ControladorDeReserva` coordena o processo de registrar uma nova locação: recebe os dados da `TelaDeReserva`, verifica a disponibilidade do `Veículo`, cria o `Contrato de Locação` e aciona a `InterfaceGatewayPagamento` para confirmar o pagamento.

3. Etapas para construir o modelo de classes de análise

3.1 Extração de substantivos e conceitos-chave

A partir dos requisitos, histórias de usuário ou casos de uso já escritos, a equipe identifica substantivos e expressões que representam conceitos importantes do negócio, gerando uma lista inicial de classes candidatas.

Exemplo prático: da frase "o atendente registra a locação de um veículo para um cliente, definindo o período e a forma de pagamento", extraem-se as classes candidatas `Atendente`, `Locação`, `Veículo`, `Cliente`, `Período` e `Pagamento`.

3.2 Classificação das classes por estereótipo

Cada classe candidata é analisada e classificada como entidade, fronteira ou controle, o que já ajuda a organizar melhor as responsabilidades de cada uma dentro do sistema.

Exemplo prático: `Cliente`, `Veículo` e `Locação` são classificadas como entidade; a tela usada pelo atendente é modelada como uma classe de fronteira (`TelaDeLocacao`); e a lógica do processo de locação é atribuída a uma classe de controle (`ControladorDeLocacao`).

3.3 Definição de atributos relevantes

Para cada classe de entidade, são definidos os atributos que representam informações relevantes para o negócio, mantendo o nível de abstração da análise (sem tipos técnicos de banco de dados ainda).

Exemplo prático: a classe `Veículo` recebe os atributos `placa`, `modelo`, `categoria` e `disponibilidade`; a classe `Locação` recebe `data de início`, `data de devolução prevista` e `valor total`.

3.4 Identificação dos relacionamentos

São definidas as associações entre as classes de entidade, incluindo a cardinalidade (quantas instâncias de uma classe se relacionam com quantas instâncias de outra).

Exemplo prático: um `Cliente` pode realizar várias `Locações` ao longo do tempo (1 para muitos), mas cada `Locação` está associada a exatamente um `Veículo` em um determinado período (1 para 1 naquele intervalo).

3.5 Validação com casos de uso e especialistas

O modelo construído é confrontado com os casos de uso do sistema (por meio de diagramas de interação) e revisado junto a especialistas do negócio, garantindo que ele realmente suporta todos os cenários previstos.

Exemplo prático: ao simular o caso de uso "Cancelar locação", a equipe percebe que faltava um atributo "status" na classe `Locação` para representar se ela está ativa, cancelada ou finalizada, e o modelo é ajustado.

4. Exemplo simplificado do modelo construído

Representação textual simplificada do modelo de classes de análise da locadora de veículos, já organizado por estereótipo:

```
[Boundary] TelaDeLocacao
[Control] ControladorDeLocacao
[Entity] Cliente (nome, CPF, telefone)
[Entity] Veiculo (placa, modelo, categoria, disponibilidade)
[Entity] Locacao (dataInicio, dataDevolucaoPrevista, valorTotal, status)
[Boundary] InterfaceGatewayPagamento
```

```
Cliente 1 ---- * Locacao * ---- 1 Veiculo
```

5. Resumo dos estereótipos de classes de análise

Estereótipo	Papel no sistema	Exemplo
Entidade (Entity)	Armazena dados persistentes do negócio	Cliente, Veículo, Locação
Fronteira (Boundary)	Faz a interface com atores ou sistemas externos	TelaDeLocacao, GatewayPagamento
Controle (Control)	Coordena a lógica de um caso de uso	ControladorDeLocacao

6. Boas práticas na construção do modelo

- Evitar incluir, nesta fase, detalhes técnicos como métodos de acesso a banco de dados, chaves estrangeiras ou anotações de frameworks — isso pertence ao modelo de projeto (design), não ao modelo de análise.
- Usar sempre o vocabulário do negócio, validando os nomes das classes e atributos junto a especialistas do domínio.
- Utilizar técnicas complementares, como cartões CRC (Classe-Responsabilidade-Colaborador), para discutir em equipe as responsabilidades de cada classe de forma simples e colaborativa.
- Construir o modelo de forma incremental, especialmente em projetos ágeis, refinando-o à medida que novos casos de uso ou histórias de usuário são detalhados.

7. Aplicação prática no dia a dia dos projetos

A construção do modelo de classes de análise é aplicada em diferentes momentos e por diferentes profissionais:

- **Analistas de sistemas** constroem o modelo logo após o levantamento de requisitos, usando-o como base para detalhar casos de uso e especificações funcionais.
- **Arquitetos de software** partem desse modelo para tomar decisões de projeto, como a divisão do sistema em camadas ou microsserviços, já com uma visão clara dos principais conceitos do negócio.

- **Desenvolvedores** utilizam o modelo como referência inicial para nomear classes, tabelas e módulos do sistema de forma consistente com a linguagem do negócio.
- **Equipes ágeis** aplicam versões simplificadas dessa técnica, como mapas de domínio ou cartões CRC, para alinhar rapidamente o entendimento do time sobre os principais conceitos de uma nova funcionalidade.

8. Considerações finais

A construção do modelo de classes de análise é uma etapa fundamental para traduzir o entendimento do negócio em uma estrutura organizada, que servirá de base sólida para as decisões técnicas das fases seguintes. Ao separar claramente conceitos de negócio (entidades), pontos de interação externa (fronteiras) e lógica de coordenação (controle), a equipe constrói um modelo mais claro, coeso e fácil de evoluir conforme o projeto avança.