

[Orientação](#) > [Diretrizes](#) > [Classe de Análise](#)

## Diretriz: Classe de Análise



Existem três Estereótipos de Classe de Análise: Limite, Controle e Entidade. Esta diretriz explica os respectivos significados e usos.

### Relacionamentos

#### Elementos Relacionados

- [Análise de Caso de Uso](#)
- [Classe de Análise](#)

[⬅ Voltar ao Início da Página](#)

### Descrição Principal

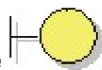
## Estereótipos de Classe de Análise

As classes de análise podem ser estereotipadas como:

- Classes de Fronteira
- Classes de Controle
- Classes de Entidade

Além de fornecer diretrizes mais específicas do processo para localizar classes, a criação de estereótipos é um poderoso modelo de objetos, pois as mudanças no modelo tendem a afetar somente uma área específica. As mudanças na interface do usuário, por exemplo, irão afetar somente as classes de fronteira. As mudanças no fluxo de controle irão afetar somente as classes de entidade. As mudanças em informações de longo prazo irão afetar somente as classes de entidade. Esses estereótipos são especialmente úteis para identificar classes em análise e design inicial. É muito provável que, em fases posteriores do design, você precise utilizar um conjunto de estereótipos ligeiramente diferente para fazer uma correlação melhor com o ambiente de implementação, com o tipo de aplicativo, e assim por diante.

### Classes de Limite



Uma **classe de limite** é utilizada para modelar a interação entre o ambiente do sistema e seus trabalhos internos. Essa interação envolve transformar e converter eventos, bem como observar mudanças na apresentação do sistema (como a interface).

As classes de fronteira modelam as partes do sistema que dependem do ambiente. As classes de entidade e de controle modelam as partes que são independentes de fatores externos ao sistema. Portanto, alterar a GUI ou o protocolo de comunicação significa alterar somente as classes de fronteira, e não as classes de entidade e de controle.

As classes de fronteira também facilitam a compreensão do sistema, pois definem suas fronteiras. Elas ajudam no design, fornecendo um bom ponto de partida para identificar serviços relacionados. Por exemplo, se você identificar uma interface de impressora logo no início do design, perceberá de imediato que também deve modelar a formatação das impressões.

Algumas classes de fronteira comuns são janelas, protocolos de comunicação, interfaces de impressora, sensores e terminais. Você não terá de modelar partes da interface de rotinas, como botões, como classes de limite separadas. Geralmente, a janela inteira é o objeto de fronteira mais refinado. As classes de fronteira também são úteis para capturar interfaces para APIs possivelmente não orientadas a objetos (como código mais antigo, por exemplo).

Você deve modelar as classes de fronteira de acordo com o tipo de fronteira que elas representam. A comunicação com outro sistema e a comunicação com um agente humano (através de uma interface do usuário) têm objetivos diferentes. Para comunicação com um agente humano, a maior preocupação é como a interface será apresentada ao usuário. Para comunicação com outro sistema, a maior preocupação é o protocolo de comunicação.

Um objeto de fronteira (uma instância de uma classe de fronteira) poderá durar mais que uma instância de caso de uso se, por exemplo, precisar ser exibido em uma tela entre o desempenho de dois casos de uso. Os objetos de fronteira, porém, costumam ter a mesma duração da instância de caso de uso.

### Localizando classes de limite

Uma classe de fronteira faz a intermediação entre a interface e algo fora do sistema. Objetos de limite isolam o sistema das mudanças no ambiente (mudanças nas interfaces com outros sistemas, mudanças nos requisitos do usuário, etc.), mantendo essas mudanças longe de afetar o restante do sistema.

Um sistema pode ter vários tipos de classes de fronteira:

- **Classes de interface com o usuário** - classes que intermedeiam a comunicação com usuários humanos do sistema
- **Classes de interface do sistema** - classes que intermedeiam a comunicação com outro sistema
- **Classes de interface de dispositivo** - classes que fornecem a interface para dispositivos (como sensores), que detectam eventos externos

### Localizar classes de interface com o usuário

Defina uma classe de limite para cada par de atores de caso de uso. Essa classe pode ser vista como tendo a responsabilidade de coordenar a interação com o agente. Você pode ainda definir classes de limite adicionais para representar classes **subsidiárias** às quais a classe de limite principal delega algumas de suas responsabilidades. Isso vale principalmente para aplicativos de GUI baseados em janelas, onde é possível modelar um objeto de limite para cada janela ou um para cada formulário. Modele somente as principais abstrações do sistema. Não modele todos os botões, listas e elementos da GUI. A meta da análise não é projetar cada detalhe, mas dar uma boa idéia de como o sistema é composto. Em outras palavras, identifique classes de limite somente para fenômenos do sistema ou para algo mencionado no **fluxo de eventos** da realização de casos de uso de análise.

Faça esboços, ou utilize dumps de tela de um [Protótipo de Interface com o Usuário](#), que ilustrem o comportamento e a aparência das classes de limite.

### Localizar classes de interface do sistema

Uma classe de fronteira que se comunica com um sistema externo é responsável pelo gerenciamento do diálogo com esse sistema. Ela fornece a interface entre o sistema externo e o sistema que está sendo criado.

#### Exemplo

Em um Caixa Eletrônico, a retirada de dinheiro deve ser verificada através da Rede do Caixa Eletrônico, um agente (que, por sua vez, faz a verificação da retirada com o sistema de contabilidade do banco). Um objeto chamado Interface da Rede do Caixa Eletrônico pode ser identificado para fazer a comunicação com aquela Rede.

A interface com um sistema existente pode já estar bem definida. Se estiver, as responsabilidades serão derivadas diretamente da definição da interface. Se existir uma definição formal de interface, ela pode ser obtida por meio de engenharia reversa e isso precisa estar formalmente definido aqui. Simplesmente anote o fato de que a interface existente será reutilizada durante o design.

### Localizar classes de interface de dispositivo

O sistema pode conter elementos que agem como se fossem externos (seus valores são alterados espontaneamente, sem que algum objeto do sistema os tenha afetado); por exemplo, um equipamento sensor. Embora seja possível representar esse tipo de dispositivo externo utilizando agentes, os usuários do sistema podem achar isso "confuso", já que o processo tende a colocar

dispositivos e agentes humanos no mesmo "nível. Depois de reunir os requisitos, no entanto, precisamos considerar a origem de todos os eventos externos e verificar se dispomos de meios para fazer com que o sistema detecte esses eventos.

Se o dispositivo estiver representado como um agente no modelo de casos de uso, será fácil justificar a utilização de uma classe de fronteira para intermediar a comunicação entre o dispositivo e o sistema. Se o modelo de caso de uso não incluir esses "agentes de dispositivo", esse será o momento adequado para incluí-los, atualizando as Descrições Suplementares dos Casos de Uso, quando apropriado.

Para cada "agente de dispositivo", crie uma classe de limite para capturar as responsabilidades do dispositivo ou do sensor. Se já existir uma interface bem definida para o dispositivo, anote-a para fazer referência posteriormente durante o design.

## Classe de Controle



Uma **classe de controle** é uma classe utilizada para modelar um comportamento de controle específico para um ou alguns casos de uso. Como objetos de controle (instâncias de classes de controle) geralmente controlam outros objetos, o comportamento de objetos de controle é do tipo coordenador. As classes de controle encapsulam um comportamento específico de caso de uso.

O comportamento de um objeto de controle está estreitamente relacionado à realização de um caso de uso específico. Em muitos cenários, é possível até dizer que os objetos de controle "executam" as realizações de casos de uso de análise. Entretanto, se as tarefas de caso de uso estiverem intrinsecamente relacionadas, alguns objetos de controle poderão participar de mais de uma realização de caso de uso de análise. Além disso, vários objetos de controle de diferentes classes de controle podem participar de um único caso de uso. Nem todos os casos de uso exigem um objeto de controle. Por exemplo, se o fluxo de eventos em um caso de uso estiver relacionado a um objeto de entidade, um objeto de fronteira poderá realizar o caso de uso em cooperação com o objeto de entidade. Comece identificando uma classe de controle para cada realização de caso de uso de análise e, em seguida, refine-a à medida que identifica mais realizações de casos de uso de análise e o respectivo compartilhamento de características comuns.

As classes de controle podem ajudar a entender o sistema, pois representam a dinâmica do sistema, controlando as principais tarefas e os fluxos de controle.

Quando o sistema executar o caso de uso, um objeto de controle será criado. Os objetos de controle geralmente desaparecem após a execução do correspondente caso de uso.

Observe que uma classe de controle não manipula **tudo** o que é necessário em um caso de uso. Em vez disso, ela coordena as tarefas de outros objetos que implementam a funcionalidade. A classe de controle delega trabalho aos objetos aos quais foi atribuída a responsabilidade pela funcionalidade.

## Localizando classes de controle

As classes de controle fornecem comportamento de coordenação no sistema. O sistema pode executar alguns casos de uso sem objetos de controle (simplesmente utilizando objetos de limite e de entidade), principalmente os casos de uso que envolvem somente a simples manipulação de informações armazenadas.

Em geral, casos de uso mais complexos exigem uma ou mais classes de controle para coordenar o comportamento de outros objetos no sistema. Exemplos de objetos de controle incluem programas como gerenciadores de transações, coordenadores de recursos e identificadores de erros.

As classes de controle realmente dissociam objetos de fronteira de objetos de entidade (e vice-versa), tornando o sistema mais tolerante a mudanças em sua fronteira. Elas também dissociam o comportamento específico de casos de uso dos objetos de entidade, tornando-os mais reutilizáveis em casos de uso e sistemas.

As classes de controle fornecem um comportamento que:

- Independe de fatores externos (o comportamento não se altera quando o ambiente é alterado),
- Define a lógica de controle (ordem entre os eventos) e as transações em um caso de uso,
- É pouco alterado se a estrutura interna ou o comportamento das classes de entidade se altera,
- Usa ou define o conteúdo de várias classes de entidade e, portanto, precisa coordenar o comportamento dessas classes.
- Não é executado da mesma maneira toda vez que é ativado (o fluxo de eventos apresenta vários estados).

### Determine se uma classe de controle é necessária

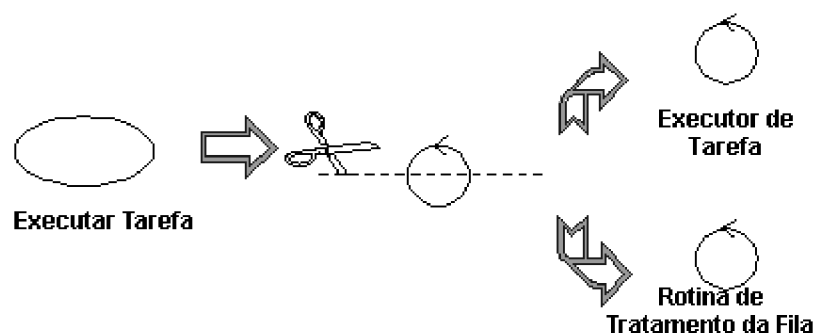
O fluxo de eventos de um caso de uso define a ordem em que diferentes tarefas são executadas. Comece investigando se o fluxo pode ser gerenciado pelas classes de fronteira e de entidade já identificadas. Para **fluxos de eventos** simples, que basicamente inserem, recuperam e exibem ou modificam informações, normalmente não justifica ter uma classe de controle separada. As classes de limite serão responsáveis por coordenar o caso de uso.

Os **fluxos de eventos** devem ser encapsulados em uma classe de controle separada quando forem complexos e consistirem em comportamento dinâmico, que possa ser alterado sem levar em consideração as interfaces (classes de limite) ou armazenamentos de informações (classes de entidade) do sistema. Por meio do encapsulamento dos **fluxos de eventos**, a mesma classe de controle possivelmente pode ser reutilizada para vários sistemas, que podem ter diferentes interfaces e armazenamentos de informações (ou, pelo menos, as estruturas de dados subjacentes).

### Exemplo: Gerenciando uma Fila de Tarefas

É possível identificar uma classe de controle no caso de uso Executar Tarefa no Sistema para Administração de Depósito. Essa classe de controle gerencia uma fila de Tarefas, assegurando a execução das Tarefas na ordem correta. Ela executa a próxima Tarefa da fila assim que um equipamento de transporte adequado é alocado. O sistema pode, portanto, executar várias Tarefas ao mesmo tempo.

Fica mais fácil descrever o comportamento definido pelo objeto de controle correspondente se você dividi-lo em duas classes de controle, o Executor de Tarefas e o Gerenciador de Filas. O objeto Gerenciador de Filas irá gerenciar somente a ordem da fila e a alocação dos equipamentos de transporte. Um único objeto Gerenciador de Filas é necessário para toda a fila. Cada vez que uma Tarefa é executada, o sistema cria um novo objeto Executor de Tarefas, que executará a próxima Tarefa. Precisamos, portanto, de um objeto Executor de Tarefas para cada Tarefa que o sistema executar.



Classes complexas devem ser divididas segundo um modelo de responsabilidades semelhantes

A principal vantagem dessa divisão é obter responsabilidades de identificação da fila separadas (algo genérico em muitos casos de uso) a partir das tarefas específicas de gerenciamento de tarefas, que são específicas para esse caso de uso. Isso torna as classes mais fáceis de entender e de adaptar, à medida que o design se desenvolve. Também traz vantagens ao equilibrar a carga do sistema, já que vários Executores de Tarefas podem ser criados conforme a necessidade de gerenciar a carga de trabalho.

### Encapsulando o fluxo de eventos principal e os fluxos de eventos alternativos/especiais em classes de controle separadas

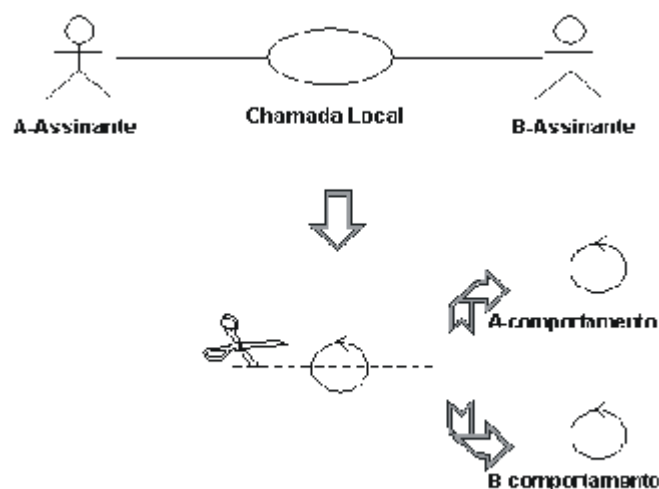
Para simplificar as mudanças, encapsule o fluxo de eventos principal e os alternativos em classes de controle diferentes. Se os fluxos alternativos e excepcionais forem completamente independentes, separe-os também. Isso facilitará a ampliação e manutenção do sistema ao longo do tempo.

### Dividir classes de controle nas quais dois atores compartilham a mesma classe de controle

Também pode ser necessário dividir as classes de controle quando vários atores usam a mesma classe de controle. Ao fazer isso, isolamos as mudanças nos requisitos de um agente a partir do restante do sistema. Quando o custo da mudança for alto ou as consequências forem terríveis, você deverá identificar e dividir todas as classes de controle que estejam relacionadas a mais de um agente. Em uma situação ideal, cada classe de controle deve interagir (através de algum objeto de fronteira) com um ou nenhum agente.

Exemplo: gerenciamento de chamadas

Considere o caso de uso **Chamada Local**. Inicialmente, podemos identificar uma classe de controle para gerenciar a chamada em si.



A classe de controle que manipula as chamadas telefônicas locais em um sistema telefônico pode ser rapidamente dividida em duas classes de controle, **comportamento A** e **comportamento B**, uma para cada agente envolvido.

Em uma chamada telefônica local, há dois agentes: o **assinante A** que inicia a chamada e o **assinante B** que recebe a chamada. O **assinante A** tira o telefone do gancho, ouve o tom de discagem e discar alguns números, que o sistema armazena e analisa. Depois de receber todos os números, o sistema envia um tom de chamada para o **assinante A** e um sinal de chamada para o **assinante B**. Quando o **assinante B** responde, o tom e o sinal param e a conversa entre os assinantes pode começar. A chamada é concluída quando os dois assinantes desligam.

Dois comportamentos devem ser controlados: o que acontece onde o assinante A se encontra e o que acontece onde o assinante B se encontra. Por esse motivo, o objeto de controle original foi dividido em dois objetos de controle, **comportamento A** e **comportamento B**.

Não será necessário dividir uma classe de controle se:

- Você puder ter certeza suficiente de que o comportamento dos atores relacionados com os objetos da classe de controle nunca será alterado ou será muito pouco alterado.
- O comportamento de um objeto da classe de controle em relação a um agente for insignificante quando comparado ao seu comportamento em relação a outro agente; um único objeto pode conter todo o comportamento. A combinação de comportamento dessa maneira terá um efeito insignificante na mutabilidade.

## Classe de Entidade



Uma **classe de entidade** é uma classe utilizada para modelar as informações e o comportamento associado que devem ser armazenados. Os objetos de entidade (instâncias de classes de entidade) são usados para manter e atualizar informações sobre alguns fenômenos, como um evento, uma pessoa ou algum objeto real. Esses objetos geralmente são persistentes, precisando de atributos e relacionamentos durante muito tempo, algumas vezes durante todo o ciclo de vida do sistema.

Um objeto de entidade geralmente não é específico para uma realização de caso de uso de análise. Às vezes, um objeto de entidade não é nem mesmo específico para o próprio sistema. Os valores de seus atributos e relacionamentos costumam ser fornecidos por um agente. Um objeto de entidade também pode ajudar a executar tarefas internas do sistema. Seu comportamento pode ser tão complicado quanto o de outros estereótipos de objeto. No entanto, ao contrário de outros objetos, esse comportamento está intrinsecamente relacionado ao fenômeno que o objeto de entidade representa. Os objetos de entidade independem do ambiente (os atores).

Os objetos de entidade representam os conceitos-chave do sistema que está sendo desenvolvido. Exemplos típicos de classes de entidade em um sistema financeiro são **Conta** e **Cliente**. Em um sistema de manipulação de redes, os exemplos são **Nó** e **Link**.

Se o fenômeno que você deseja modelar não for usado por outras classes, será possível modelá-lo como um atributo de uma classe de entidade ou mesmo como um relacionamento entre classes de entidade. Por outro lado, se o fenômeno for usado por qualquer outra classe do modelo de design, será preciso modelá-lo como uma classe.

As classes de entidade fornecem um outro ponto de vista do sistema, pois mostram a estrutura lógica dos dados, que pode ajudá-lo a compreender o que o sistema deve oferecer aos usuários.

### Localizando classes de entidade

As classes de entidade representam os armazenamentos de informações no sistema e geralmente são usadas para representar os conceitos-chave que o sistema gerencia. Os objetos de entidade costumam ser passivos e persistentes. Suas principais responsabilidades são armazenar e gerenciar informações no sistema.

O Glossário (desenvolvido durante os requisitos) e um modelo de domínio de negócios (desenvolvido durante a modelagem de negócios, se esta tiver sido executada) são fontes freqüentes de inspiração para as classes de entidade.

## Restrições de Uso do Estereótipo de Associação

### Restrições para Classes de Limite

As seguintes situações são permitidas:

- Comunicar associações entre duas classes de Fronteira, por exemplo, para descrever como uma janela específica está relacionada a outros objetos de fronteira.
- Comunicar ou assinar associações de uma classe de Fronteira para uma classe de Entidade, pois os objetos de fronteira podem precisar controlar certos objetos de entidade entre ações no objeto de fronteira ou ser informados de mudanças de estado no objeto de entidade.
- Comunicar associações de uma classe de Fronteira para uma classe de Controle, para que um objeto de fronteira possa disparar um determinado comportamento.

### Restrições para Classes de Controle

As seguintes situações são permitidas:

- Comunicar ou assinar associações entre classes de Controle e classes de Entidade, pois os objetos de controle podem precisar controlar certos objetos de entidade entre ações no objeto de controle ou ser informados de mudanças de estado no objeto de entidade.

- Comunicar associações entre classes de Fronteira e de Controle, permitindo que os resultados do comportamento disparado sejam comunicados ao ambiente.
- Comunicar associações entre classes de Controle, permitindo a construção de padrões de comportamento mais complexos.

### Restrições para Classes de Entidade

As classes de entidade devem ser apenas a origem de associações (comunicar ou assinar) com outras classes de entidade. Os objetos da classe de entidade normalmente têm longa duração, já os objetos de classe de fronteira e de controle costumam durar pouco. Do ponto de vista arquitetural, é sensato limitar a visibilidade que um objeto de entidade tem do ambiente; dessa maneira, o sistema fica mais receptivo a mudanças.

### Resumo de Restrições

De/Para (navegabilidade)	Fronteira	Entidade	Controle
Fronteira	comunicar	comunicar assinar	comunicar
Entity		comunicar assinar	
Controle	comunicar	comunicar assinar	comunicar

Combinações de Estereótipos de Associação Válidas

### Aplicando Consistência

- Quando um novo comportamento for identificado, verifique se há uma classe existente que tenha responsabilidades semelhantes e, sempre que possível, reutilize as classes. Só crie novas classes se tiver certeza de que nenhum objeto existente pode executar o comportamento.
- À medida que as classes forem identificadas, examine-as para verificar se elas têm responsabilidades consistentes. Quando as responsabilidades das classes forem desconexas, divida o objeto em duas ou mais classes. Atualize os diagramas de interação da forma adequada.
- Se uma classe for dividida porque responsabilidades desconexas foram identificadas, examine as colaborações nas quais a classe desempenha um papel a fim de verificar se a colaboração precisa ser atualizada. Atualize a colaboração, se necessário.
- Uma classe com uma única responsabilidade não é um problema por si só, mas levanta questões sobre o motivo pelo qual ela é necessária. Esteja preparado para questionar e justificar a existência de todas as classes.

[🏠 Voltar ao Início da Página](#)