**Forma

Descrição gerada automaticamente com confiança média**

**Trabalho de pesquisa sobre diagramas**

**Nome**: Victor Bernardo Gomes de Tatagiba-202211960

**Curso**: Engenharia de Software

**Disciplina:** Engenharia de Requisitos e Análise de Sistemas

**Professor:** Paulo Marcelo Paes Coelho Dantas de Goes

**UML (Linguagem de Modelagem Unificada | Unified Modeling Language)**

A Linguagem de Modelagem Unificada (UML) é a linguagem padrão que muitos engenheiros de software utilizam para criar uma visão geral de sistemas complexos.

**Abordaremos neste trabalho, os principais diagramas UML:**

* de Classes;
* de Objetos;
* de Componentes;
* de Implantação;
* de Pacotes;
* de Estrutura Composta;
* de Perfil;
* de Transição de Estados;
* de Atividade;
* de Interação e Sequência;
* de Colaboração (comunicação);
* de Tempo.

# DIAGRAMA DE CLASSES

Um dos tipos mais populares na UML é o diagrama de classes. Bastante usado por engenheiros de software para documentar arquiteturas de software, os diagramas de classes são um tipo de diagrama da estrutura porque descrevem o que deve estar presente no sistema a ser modelado.

Como as classes são os componentes básicos dos objetos, os diagramas de classes são os componentes básicos da UML. Os diversos componentes em um diagrama de classes podem representar as classes que serão realmente programadas, os principais objetos ou as interações entre classes e objetos.

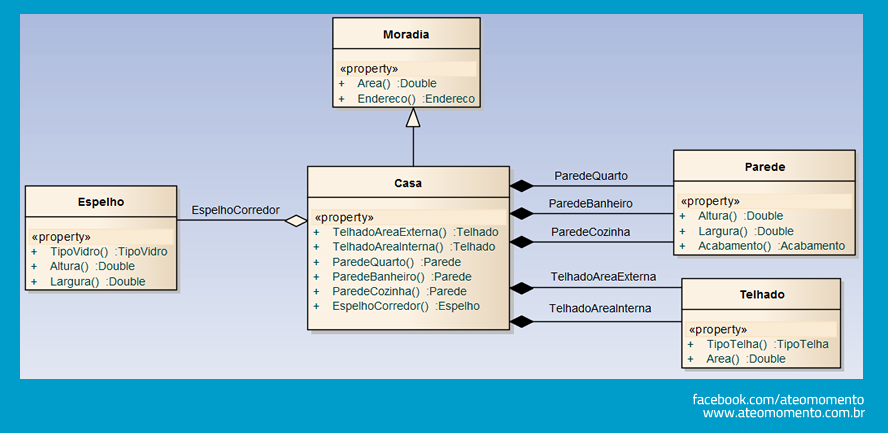
A forma de classe em si consiste em um retângulo com três linhas. A linha superior contém o nome da classe, a linha do meio, os atributos da classe e a linha inferior expressa os métodos ou operações que a classe pode utilizar. Classes e subclasses são agrupadas juntas para mostrar a relação estática entre cada objeto.

## Benefícios

Diagramas de classes oferecem uma série de benefícios para qualquer organização. Use diagramas de classes UML para:

* Ilustrar modelos de dados para sistemas de informação, não importa quão simples ou complexo.
* Entender melhor a visão geral dos esquemas de uma aplicação.
* Expressar visualmente as necessidades específicas de um sistema e divulgar essas informações por toda a empresa.
* Criar gráficos detalhados que destacam qualquer código específico necessário para ser programado e implementado na estrutura descrita.
* Fornecer uma descrição independente de implementação de tipos utilizados em um sistema e passados posteriormente entre seus componentes.

**Exemplo de diagrama de classe:**

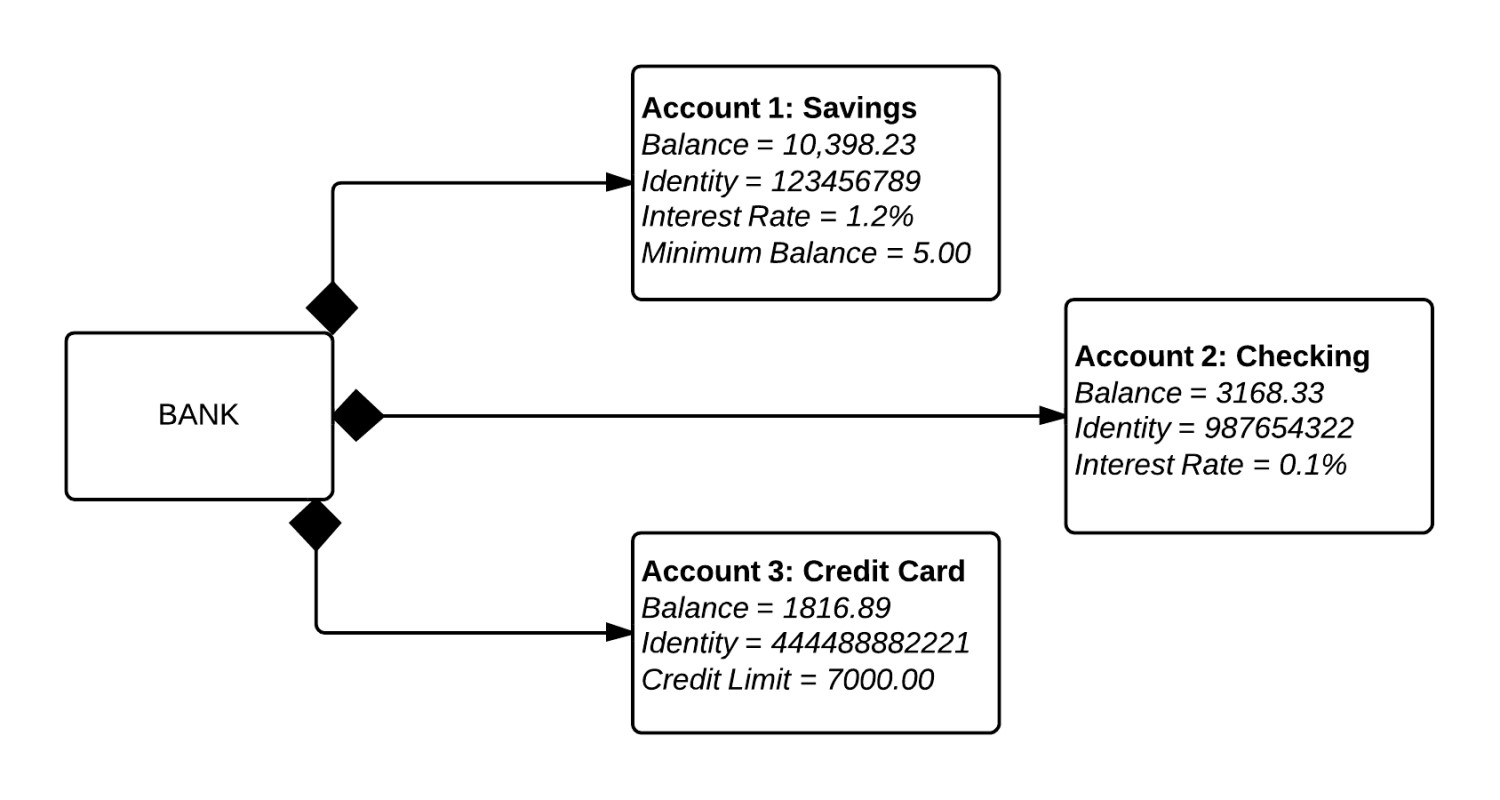
O diagrama abaixo contem além dos compartimentos com os nomes das classes e atributos, possui também um outro compartimento contendo as operações da classe.

# DIAGRAMA DE OBJETOS

Um diagrama de objetos incide sobre os atributos de um conjunto de objetos, e como eles se relacionam entre si.

Por exemplo, neste diagrama de objetos abaixo, as três contas bancárias estão relacionadas ao próprio banco. Os nomes da classe mostram os tipos de contas (poupança, corrente e de cartão de crédito) que um determinado cliente poderia ter neste banco. Os atributos de classe são diferentes para cada tipo de conta. Por exemplo, o objeto do cartão de crédito possui um limite de crédito, enquanto a poupança e a conta corrente possuem taxas de juros.

Diagramas de objetos não são usados apenas em casos de uso bancário. Você pode criar um diagrama de objetos para árvores genealógicas, departamentos corporativos ou qualquer outro sistema com partes inter-relacionadas.



Diagramas de objetos não são usados apenas em casos de uso bancário. Você pode criar um diagrama de objetos para árvores genealógicas, departamentos corporativos ou qualquer outro sistema com partes inter-relacionadas.

## Elementos dos diagramas de objetos

Diagramas de objetos são fáceis de criar: são feitos de objetos, representados por retângulos e ligados entre si por linhas. Confira os principais elementos de um diagrama de objetos.

* ***Objetos:*** Objetos são instâncias de uma classe. Por exemplo, se um “carro” for uma classe, um modelo Nissan Altima de 2007 é um objeto de uma classe.
* ***Títulos de classe:*** Títulos de classe são os atributos específicos de uma determinada classe. No diagrama de objetos de árvores genealógicas, títulos de classe incluem o nome, sexo e idade dos membros da família. Você pode listar títulos de classe como itens no objeto ou até mesmo nas propriedades do próprio objeto (tal como a cor).
* ***Atributos de classe:*** Atributos de classe são representados por um retângulo com duas abas que indicam um elemento de software.
* ***Links:*** Ligações são as linhas que conectam duas formas de um diagrama de objetos, uma a outra. O diagrama de objetos corporativo abaixo mostra como departamentos são ligados no organograma tradicional.

# DIAGRAMA DE COMPONENTES

O diagrama de componentes mostra o relacionamento entre diferentes componentes de um sistema. Para fins de UML 2.0, o termo "componente" refere-se a um módulo de classes que representa sistemas ou subsistemas independentes com capacidade de interagir com o restante do sistema.

Para isso, existe uma abordagem de desenvolvimento em torno de componentes: o desenvolvimento baseado em componentes (CBD). Nela, o diagrama de componentes identifica os diferentes componentes para que todo o sistema funcione corretamente.

O mais comum, em uma abordagem de programação OO, é o desenvolvedor sênior usar o diagrama de componentes para agrupar classes com base em um objetivo comum, para que o desenvolvedor e outras pessoas envolvidas analisem o projeto de desenvolvimento de software de forma generalizada.

## Benefícios do diagrama de componentes

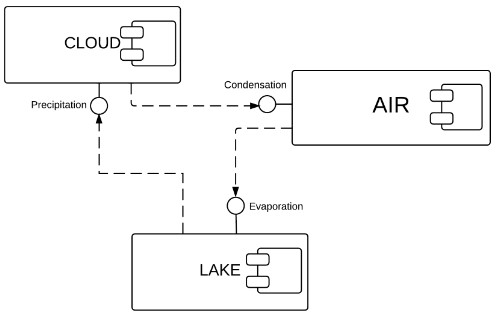
Embora o diagrama de componentes pareça complexo à primeira vista, ele é indispensável na criação do seu sistema. Com ele, sua equipe:

* Imagina a estrutura física do sistema;
* Presta atenção aos componentes do sistema e como eles se relacionam; ● Enfatiza o comportamento do serviço quanto à interface.

## Como usar o diagrama de componentes

O diagrama de componentes em UML dá uma visão geral do sistema de software. Compreender o comportamento exato do serviço fornecido por cada software fará de você um desenvolvedor melhor. O diagrama de componentes descreve sistemas de software implementados em qualquer linguagem ou estilo de programação.

No diagrama de componentes, a UML determina que componentes e pacotes sejam conectados a linhas que representam conectores de montagem e de delegação.



# DIAGRAMA DE IMPLEMENTAÇÃO EM UML

Fonte:https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-implementacao-uml

No contexto da Linguagem de modelagem unificada (UML), um diagrama de implementação faz parte da família de diagramação estrutural pois descreve um aspecto do sistema em si. Neste caso, o diagrama de implementação descreve a implementação física de informações geradas pelo programa de software em componentes de hardware. A informação gerada pelo software é chamada de artefato. Não se deve confundir este com o uso do mesmo termo em outras abordagens de modelagem, como BPMN.

Os diagramas de implementação são feitos de várias formas de UML. As caixas tridimensionais conhecidas como nódulos, representam os elementos básicos de software ou hardware, ou nódulos no sistema. As linhas de nódulo a nódulo indicam relacionamentos e as formas menores contidas dentro das caixas representam os artefatos de software empregados.

## Aplicações do diagrama de implementação

Os diagramas de implementação possuem várias aplicações de grande valor. Eles podem ser usados para:

* Mostrar quais elementos de software são implementados por quais elementos de hardware.
* Ilustrar o processamento do tempo de execução do hardware.

● Propiciar uma visão da topologia do sistema do hardware.

## Diretrizes do diagrama de implementação

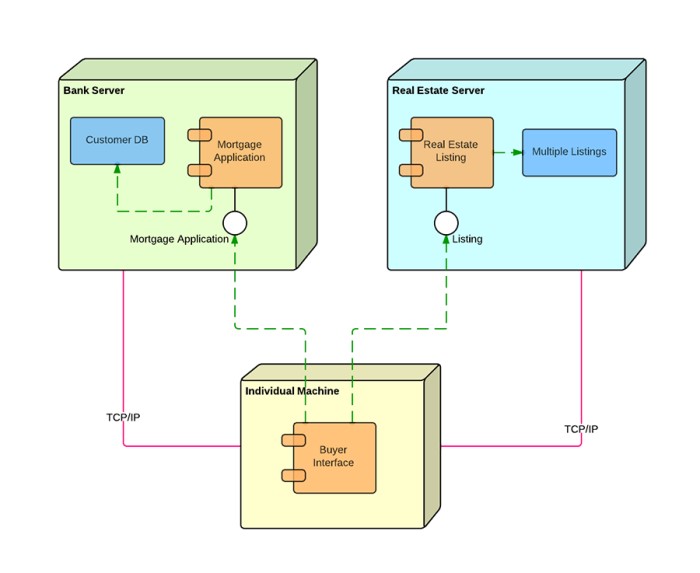
Antes de começar a criar diagramas, considere as seguintes perguntas:

1. **Você identificou o escopo de seu sistema?** *Por exemplo, é preciso saber se está diagramando um aplicativo único ou diagramando a implementação em toda uma rede de computadores.*
2. **Quais as limitações de seu hardware físico?** *Com quais sistemas legados você precisará interagir? Certifique-se de que conhece o software em operação e os protocolos com os quais trabalhará e qual sistema de monitoração será adotado.*
3. **Qual arquitetura de distribuição você está usando?** *É necessário saber quantos níveis seu aplicativo terá e em qual aplicativo você vai implantá-lo.*
4. **Você possui todos os nódulos necessários?** *Você sabe como eles todos estão conectados?*
5. **Você sabe quais componentes estão em quais nódulos?**

## Elementos do diagrama de implementação

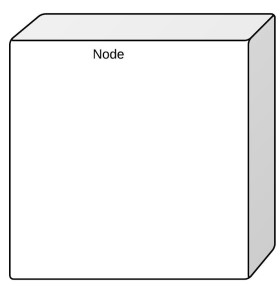
Diagramas de implementação são constituídos por uma variedade de formas. Esta lista oferece uma visão geral dos elementos básicos que podem ser encontrados. É possível ver a maioria destes itens ilustrados na imagem abaixo.

* **Artefato:** um produto desenvolvido pelo software, simbolizado por um retângulo com o nome e a palavra “artefato” entre setas duplas.
* **Associação:** uma linha que indica uma mensagem ou outro tipo de comunicação entre nós.
* **Componente:** um retângulo com duas abas que indica um elemento de software.
* **Dependência:** uma linha pontilhada finalizada em uma seta, indicando que um nó ou componente é dependente de um outro.
* **Interface:** um círculo que indica uma relação contratual. São aqueles objetos que notam que a interface deve realizar algum tipo de obrigação.
* **Nó:** um objeto de hardware ou software representado por uma caixa tridimensional.
* **Nó como contêiner:** um nó que contém outro nó em seu interior, tal como no exemplo abaixo onde os nós contêm componentes.
* **Estereótipo:** um dispositivo contido dentro do nó, apresentado na parte superior do nó, com o nome entre setas duplas.



## Símbolos e notação do diagrama de implementação

### Nódulos (Nós)

Existem dois tipos de nós em um diagrama de implementação: nós de dispositivos e nós de ambiente de execução. Nós de dispositivos são recursos de computação com capacidades de processamento e execução de programas. Alguns exemplos de nós de dispositivos são

PCs, laptops ou celulares.

Um nó de ambiente de execução, ou EEN, é qualquer sistema de

computação que resida dentro de um nó de dispositivo. Pode ser um sistema operacional, um JVM ou outro container web.

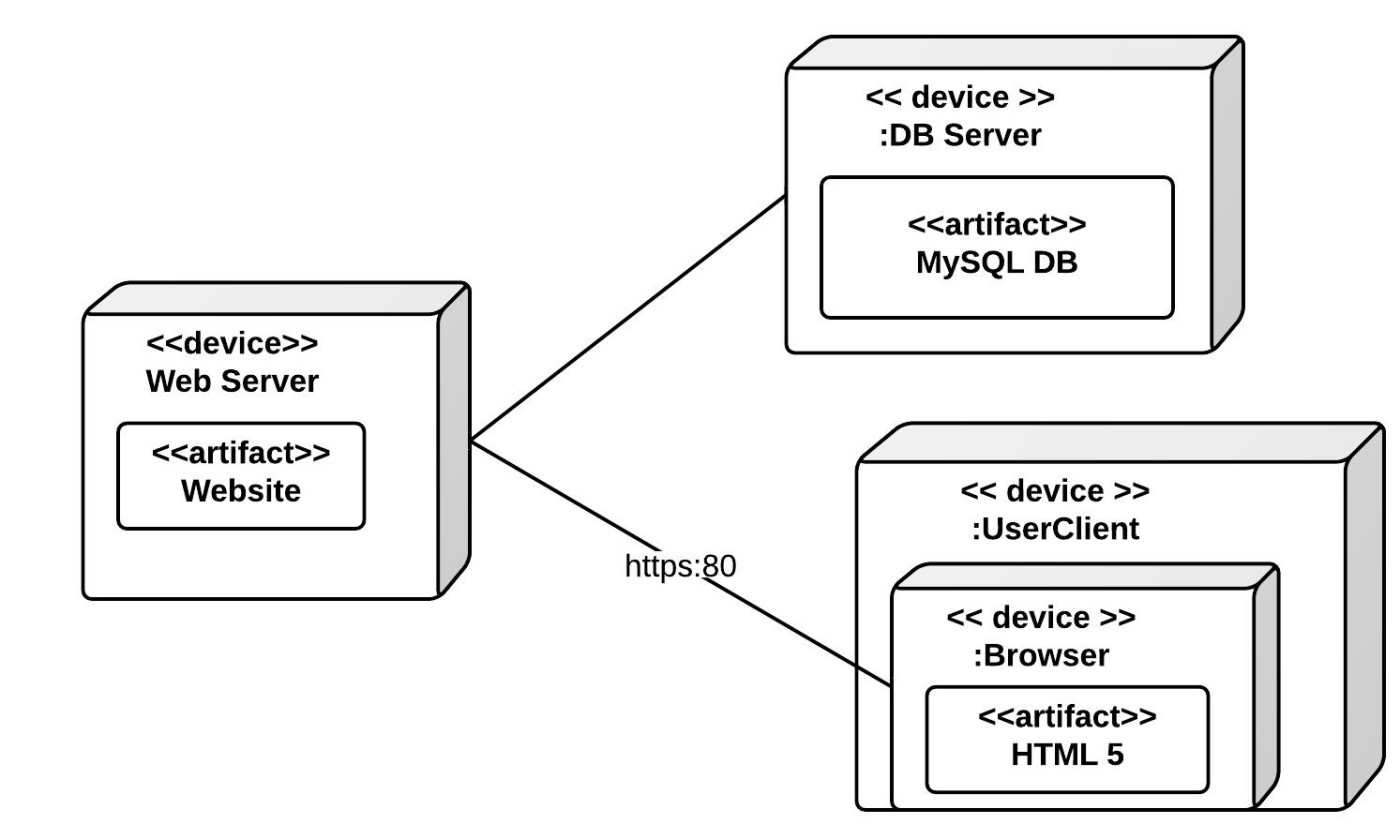
### Base de dados

Bancos de dados representam qualquer dado armazenado pelo sistema implementado. Em alguns casos, o banco de dados é representado como um outro nó, no entanto, às vezes você verá esta forma representando um banco de dados.

#### Outras formas

* **Caminho da comunicação:** uma linha reta que representa a comunicação entre dois nós de dispositivos.
* **Artefatos:** uma caixa com o cabeçalho “>” e o nome do arquivo.
* **Pacote:** uma caixa em forma de arquivo que agrupa todos os nós de dispositivos para integrar toda a implementação.
* **Componente:** uma entidade necessária para executar uma função de estereótipo. Consulte este guia para notação de componentes UML.

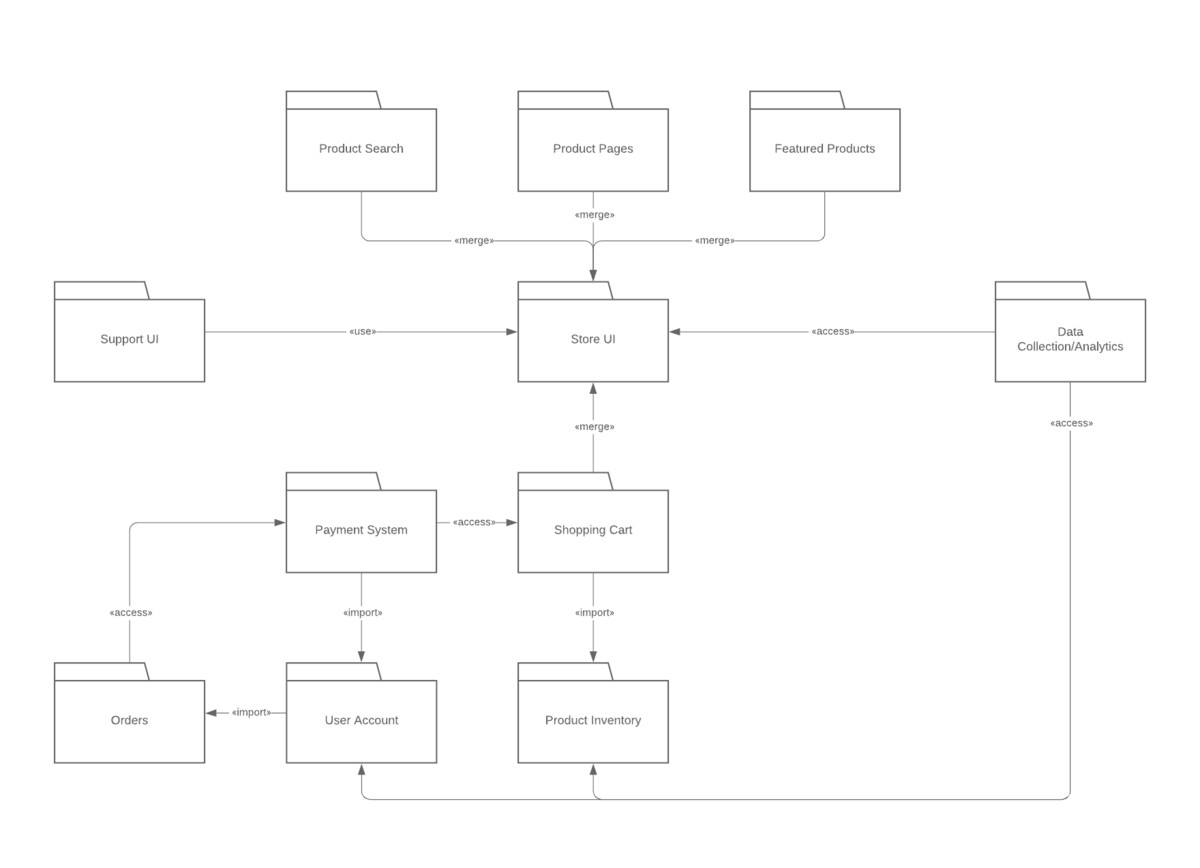
## Exemplo de diagrama de implementação

Este exemplo mostra um diagrama básico de implementação. Há um servidor web, um servidor de banco de dados e o aparelho de onde o usuário visualiza o site. É possível acrescentar uma maior complexidade mostrando as diferentes partes do servidor web e como o Javascript funciona no Usuário/Cliente.

# DIAGRAMA DE PACOTES EM UML

https://www.lucidchart.com/pages/pt/diagrama-de-pacotes-uml

Diagramas de pacotes são diagramas estruturais usados para mostrar, em uma forma de pacotes, a organização e disposição de vários elementos de modelos. Um pacote é um agrupamento de elementos UML relacionados, como diagramas, documentos, classes ou até mesmo outros pacotes. Cada elemento é colocado dentro do pacote e é representado como uma pasta de arquivo dentro do diagrama, e depois organizado hierarquicamente no diagrama. Diagramas de pacotes são usados para proporcionar uma organização visual de uma arquitetura em camadas de qualquer classificador UML, por exemplo, um sistema de software.



## Benefícios de um diagrama de pacotes

Um diagrama de pacotes bem projetado oferece inúmeros benefícios para quem precisa de uma visualização de seu sistema ou projeto UML.

* O diagrama fornece uma visão clara da estrutura hierárquica dos variados elementos UML dentro de um determinado sistema.
* Esses diagramas podem simplificar diagramas de classes complexos, criando elementos visuais organizados.
* Eles oferecem uma ótima visibilidade geral de projetos e sistemas de grande escala.
* Os diagramas de pacotes UML podem ser usados para esclarecer de forma visual uma grande variedade de projetos e sistemas.
* Os elementos visuais podem ser atualizados com facilidade conforme a evolução dos sistemas e projetos.

## Componentes básicos do diagrama de pacotes

A composição do diagrama de pacotes é relativamente simples. Cada diagrama contém apenas dois símbolos:

|  |  |  |
| --- | --- | --- |
| **Imagem do símbolo** | **Nome do símbolo** | **Descrição** |
|  | Pacote | Agrupa elementos comuns com base em dados, comportamento ou interação do usuário |
|  | Dependência | Mostra a relação entre um elemento (pacote, elemento nomeado etc.) e outro |

Esses símbolos podem ser usados de várias maneiras para representar diferentes iterações de pacotes, dependências e outros elementos dentro de um sistema. Veja quais são os componentes básicos encontrados em um diagrama de pacotes:

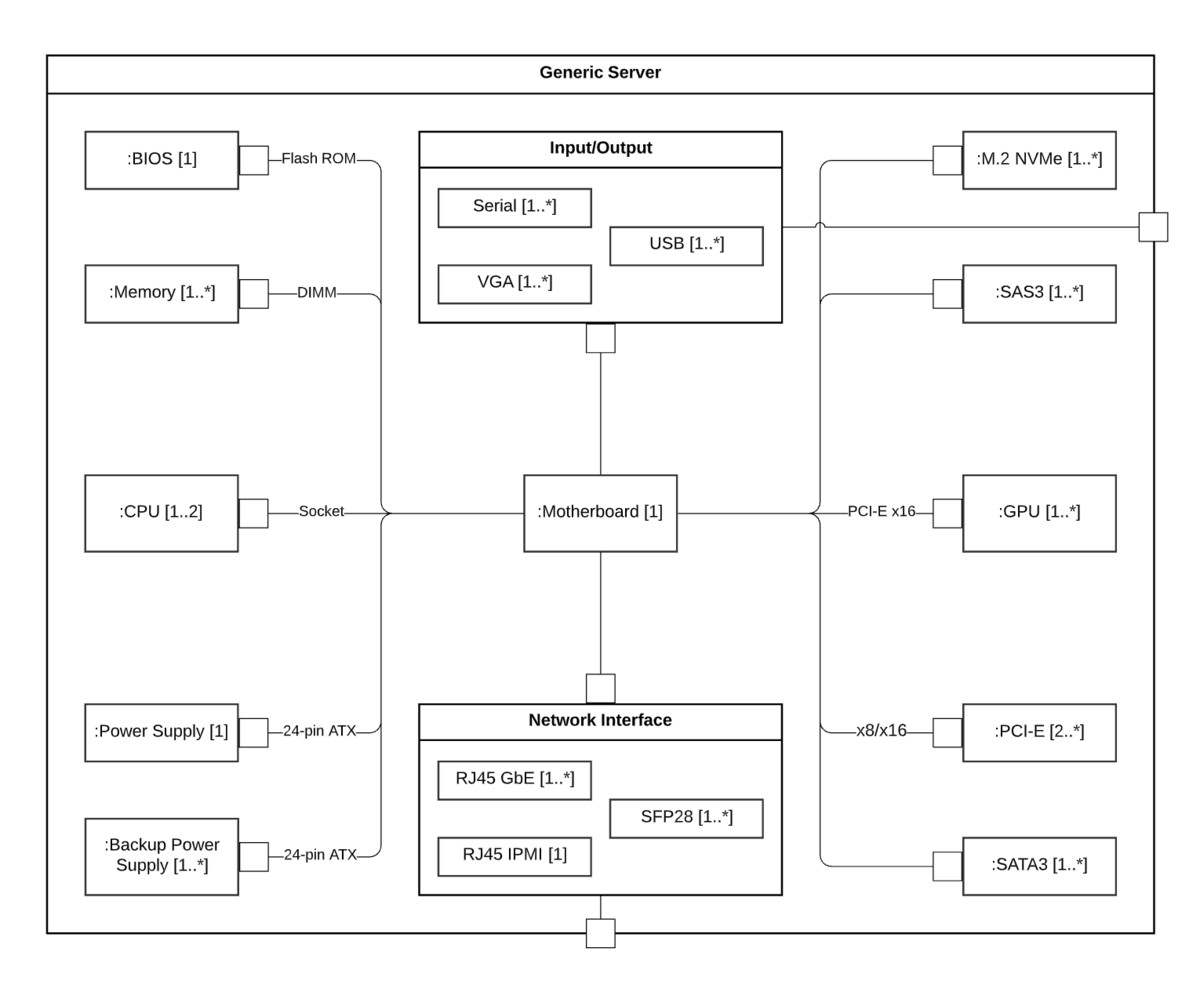
* **Pacote:** um espaço de nomes usado para agrupar elementos logicamente relacionados dentro de um sistema. Cada elemento contido no pacote deve ser um elemento empacotável e ter um nome exclusivo;
* **Elemento empacotável:** um elemento nomeado, possivelmente de propriedade direta de um pacote. Pode incluir eventos, componentes, casos de uso e os próprios pacotes. Elementos empacotáveis também podem ser renderizados como um

retângulo dentro de um pacote, com um rótulo de um nome apropriado;

* **Dependências:** uma representação visual de como um elemento (ou conjunto de elementos) depende ou influencia outro. As dependências são divididas em dois grupos: dependências de acesso e de importação. (Veja a próxima seção para mais informações.);
* **Importação de elemento:** um relacionamento direcionado entre um espaço de nomes de importação e um elemento empacotável importado. É usada para importar elementos individuais selecionados sem recorrer a uma importação de pacote e sem torná-lo público dentro do espaço de nomes;
* **Importação de pacote:** um relacionamento direcionado entre um espaço de nomes de importação e um pacote importado. Este tipo de relacionamento direcionado adiciona os nomes dos membros do pacote importado ao próprio espaço de nomes dele;
* **Mesclagem de pacotes:** uma relação direcionada em que o conteúdo de um pacote é estendido pelo conteúdo de outro. Essencialmente, o conteúdo de dois pacotes é juntado para produzir um novo pacote.

# DIAGRAMA DE ESTRUTURA COMPOSTA EM UML

https://www.lucidchart.com/pages/pt/diagrama-de-estrutura-composta-uml

O diagrama de estrutura composta é um diagrama estrutural UML que oferece uma visão geral lógica de todo ou parte de um sistema de software. O diagrama atua como uma perspectiva dentro de um determinado classificador estruturado, e define suas classes de configurações, interfaces, pacotes e as relações entre eles a partir de um nível detalhado.

## Os benefícios de diagramas de estrutura composta

O diagrama de estrutura composta permite ao usuário ver exatamente o que está contido em um objeto, e especifica como diferentes propriedades se encaixam para produzir um determinado comportamento. As diferentes relações em um sistema de software complexo podem ser difíceis de entender, no entanto, ao decompor a funcionalidade de um sistema, você pode descobrir informações valiosas sobre como as estruturas estão interligadas, como as informações são comunicadas e muito mais.

## O diagrama de estrutura composta também oferece os seguintes benefícios

* Ajuda o usuário a entender o estado atual de seu sistema;
* Detalha a estrutura interna de diversas classes, interfaces ou componentes e suas interações;
* Descreve arquiteturas de tempo de execução e padrões de uso não encontrados em diagramas estáticos;
* Fornece ao usuário informações para otimizar e solucionar problemas em seu sistema.

## Componentes básicos de um diagrama de estrutura composta

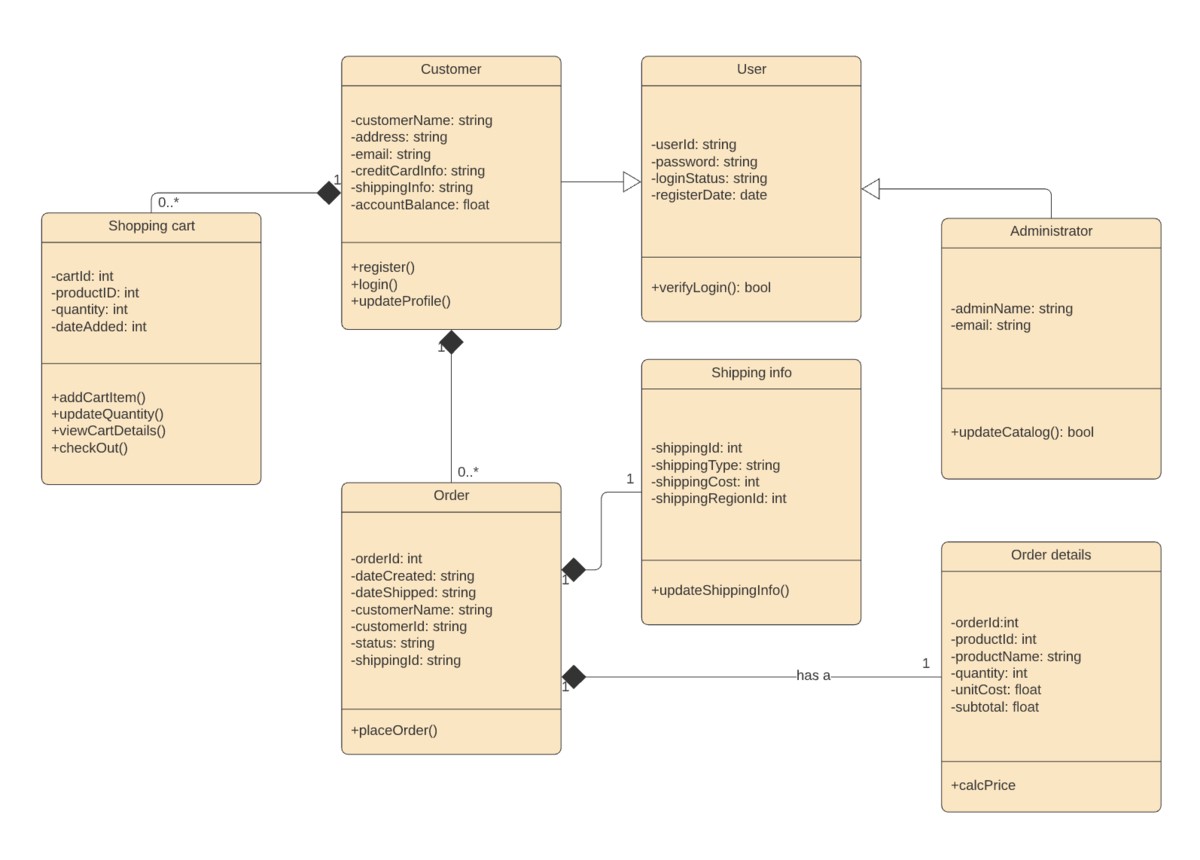
O diagrama de estrutura composta é feito de diversos símbolos UML que representam partes individuais de um sistema, bem como as relações entre eles.

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Nome** | **Descrição** |
|  | Exterminador | Indica pontos de início e término |
|  | Nó (circular) | Representa eventos ou marcos, e contém números |
|  | Nó (retangular) | Representa eventos ou marcos, e contém números |
|  | Ator | Interage com o sistema de fora do sistema (p. ex., pessoa, equipamento etc.) |
|  | Classe | Agrupa objetos com propriedades ou  comportamentos em comum  (p. ex., operações, parâmetros, atributos etc.  em comum) |
|  | Parte | Atua como uma instância de tempo de execução de classes ou interfaces |
|  | Porta | Atua como um ponto de interação entre uma  instância de classificador  (ou seu comportamento) e seu ambiente |
|  | Interface | Especifica o comportamento que o implementador concorda em atender |
|  | Conector | Mostra a comunicação entre as partes |

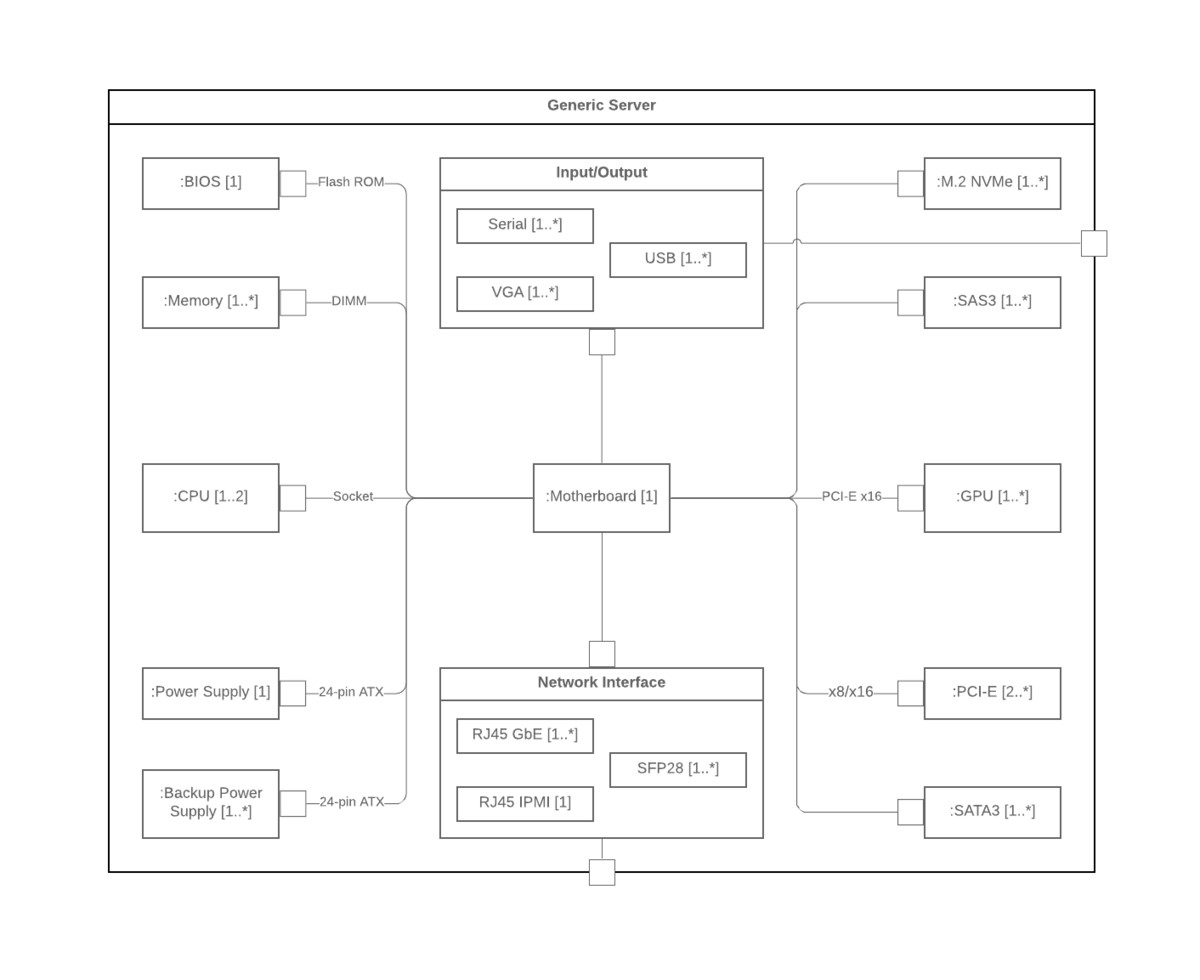
## Diagrama de estrutura composta x diagrama de classes

Por serem diagramas UML, tanto os diagramas de estrutura composta quanto os diagramas de classes são usados para visualizar e organizar os atores, interações e artefatos dentro de um sistema. No entanto, embora os diagramas de estrutura composta e diagramas de classes tenham significados semelhantes, eles diferem na forma como expressam esses significados. Em outras palavras, os diagramas de estrutura composta são mais específicos e menos ambíguos do que os diagramas de classes.

O diagrama de estrutura composta permite ao usuário modelar as implementações da atividade de um artefato dentro de um tempo de execução com mais clareza. Ele também consegue mostrar melhor a decomposição em contexto e descrever a estrutura interna de várias classes e as relações definidas entre elas. Em outras palavras, o diagrama de estrutura composta é a melhor opção para transmitir informações concretas e explícitas sobre os comportamentos e as relações dentro do seu sistema.



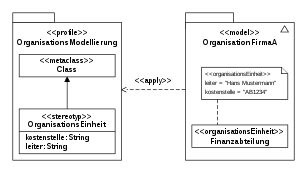
## Exemplo de diagrama de estrutura composta



# DIAGRAMA DE PERFIL EM UML

https://pt.frwiki.wiki/wiki/Diagramme\_de\_profils

Um diagrama de perfil é um diagrama de estrutura que permite o uso de perfis para um determinado metamodelo. Apresentado com UML 2.2, este diagrama fornece uma representação dos conceitos usados na definição de perfis (pacotes, estereótipos, aplicação de perfis, etc.).



## O diagrama de perfil permite o uso dos seguintes elementos

● Estereótipo (representado como uma classe com o estereótipo <<stereotype>>); ● Metaclasse (representada como uma classe com o estereótipo <<metaclass>>); ● Perfil (representado como um pacote com o estereótipo <<profile>>).

## Relações entre elementos

Os elementos de um diagrama de perfil podem ser relacionados usando os seguintes elementos:

* Extensão (de um estereótipo a uma metaclasse, representada por uma seta sólida);
* Aplicação de perfil (de um perfil para um pacote de metamodelo, mostrado como uma seta pontilhada com o estereótipo <<apply>>);
* Referência (de um perfil a um pacote ou metaclasse referenciado pelo perfil, representado como uma seta pontilhada com o estereótipo <<reference>>).

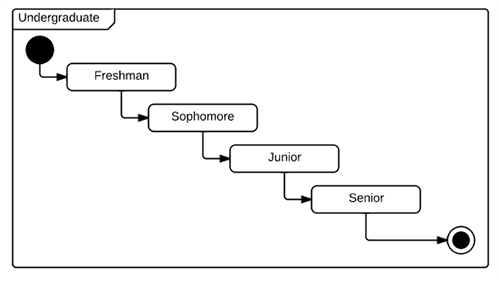
# DIAGRAMA DE TRANSIÇÃO DE ESTADOS EM UML

https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml

Uma transição de estados é qualquer dispositivo que armazena o status de um objeto em um determinado momento e pode mudar de status ou causar outras ações baseado na entrada que recebe. Estados indicam as diferentes combinações de informações que um objeto pode conter, e não como ele se comporta. Para entender os diferentes estados de um objeto, você pode visualizar todos os possíveis estados e mostrar como um objeto chega a determinado estado; e isso pode ser feito com um diagrama de estados UML.

Cada diagrama de estados começa, geralmente, com um círculo escuro que indica o estado inicial, e termina com um círculo contornado, indicando o estado final. No entanto, apesar de ter pontos iniciais e finais claros, os diagramas de estados não são necessariamente a melhor ferramenta para registrar a progressão geral de eventos. Em vez disso, eles ilustram tipos específicos de comportamento, principalmente mudanças de um estado para outro.

Diagramas de estados retratam principalmente estados e transições. Estados são representados por retângulos com cantos arredondados e rotulados com o nome do estado. As transições são marcadas com setas que fluem de um estado para outro, mostrando como os estados mudam. Abaixo, você pode ver esses dois elementos em ação por meio de um diagrama básico retratando a vida de um estudante.



## Aplicações de diagramas de estados

Como a maioria dos diagramas UML, diagramas de estados têm diversos usos. As principais aplicações são as seguintes:

* Descrever objetos orientados a eventos em um sistema reativo;
* Ilustrar cenários de caso de uso em um contexto de negócios;
* Descrever como um objeto se move por vários estados em seu tempo de vida;
* Mostrar o comportamento geral de uma máquina de estados ou o comportamento de um conjunto relacionado de máquinas de estados.

## Símbolos e componentes de diagramas de estados

Você pode incluir muitas formas diferentes em um diagrama de estados, principalmente se você decidir combiná-lo com outro diagrama. Esta lista resume as formas mais comuns que podem ser encontradas.

## Estado composto

Um estado que contém subestados em sua estrutura. Veja o exemplo de diagrama de estados para universidades abaixo. Neste exemplo, a “Matrícula” é o estado composto porque ela engloba diversos subestados no processo de matrícula.

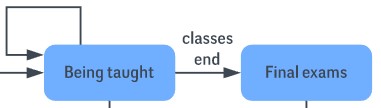
## Pseudoestado de escolha

Um símbolo de diamante que indica uma condição dinâmica com resultados potenciais ramificados.



## Evento

Uma instância que aciona uma transição, rotulada acima da seta de transição apropriada. Neste caso, o “término das aulas” é o evento que aciona o fim do estado “Sendo ensinado” e o início do estado de “Provas finais”.



## Ponto de saída

O ponto em que um objeto sai de um estado composto ou uma máquina de estados, identificado por um círculo com um X no meio. O ponto de saída é geralmente usado se o processo não está concluído mas tem que ser encerrado por algum erro ou outro problema.



## Primeiro estado

Um marcador para o primeiro estado no processo, representado por um círculo escuro com uma seta de transição.



## Proteção

Uma condição booleana que permite ou impede uma transição, escrita acima da seta de transição.

## Estado

Um retângulo com cantos arredondados que indica a natureza atual de um objeto.



## Subestado

Um estado contido dentro da região de um estado composto. No diagrama de máquina de estados para universidades encontrado abaixo, “Matrículas abertas” é um subestado no estado composto maior de “Matrícula”.

## Exterminador

Um círculo com um ponto no meio, indicando que um processo foi terminado.



## Transição

Uma seta que flui de um estado a outro, indicando um estado em mudança.

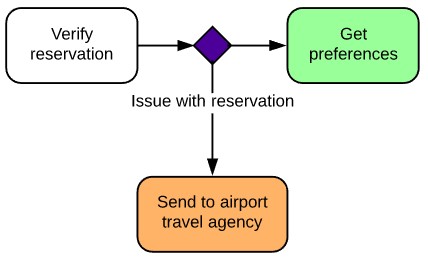


## Comportamento transicional

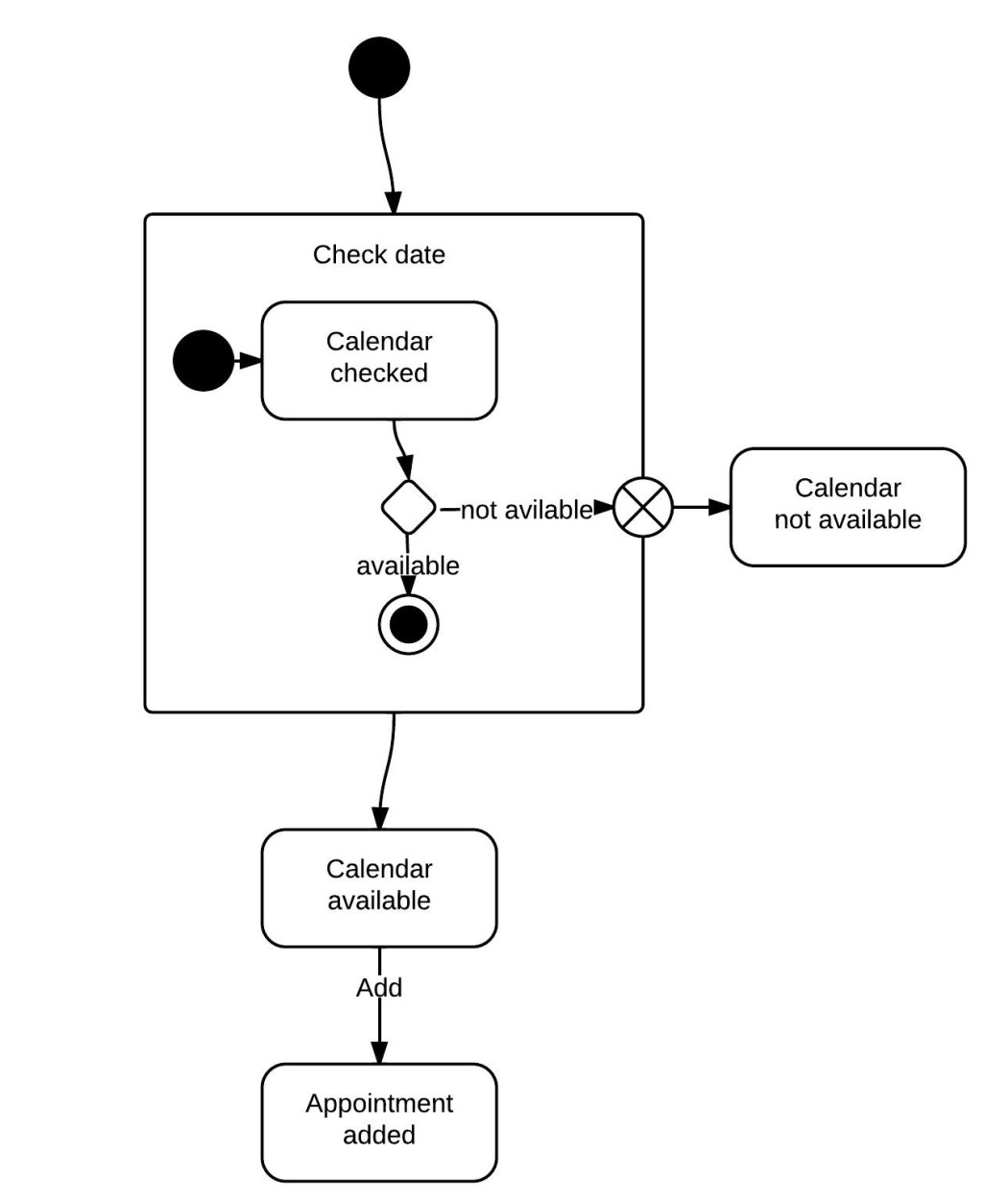
Um comportamento resultante que ocorre quando um estado entra em transição, escrito acima da seta de transição.

## Acionador

Um tipo de mensagem que move um objeto de forma ativa de estado para estado, escrito acima da seta de transição. Neste exemplo, “Problema com a reserva” é o acionador que mandaria a pessoa para a agência de viagens do aeroporto em vez do próximo passo no processo.

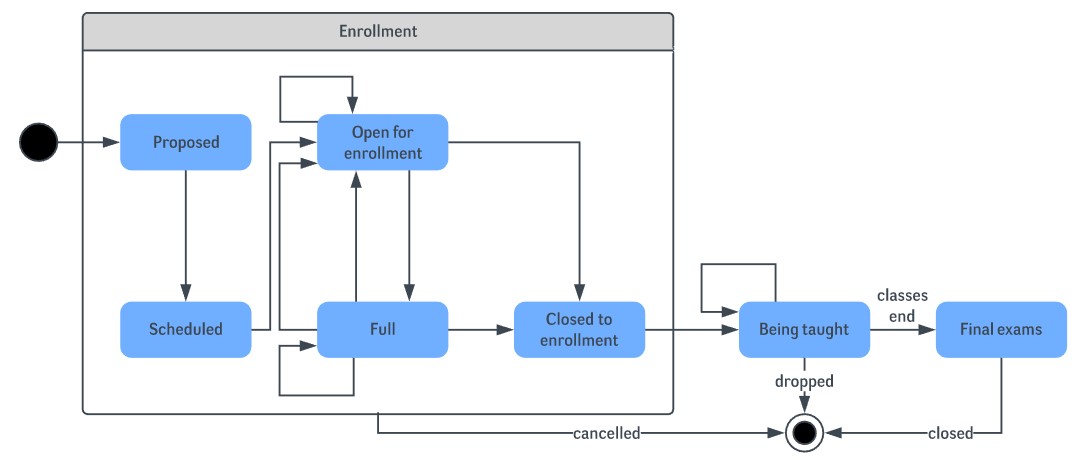


## Exemplo de diagrama de estados de disponibilidade de calendário

Este exemplo de diagrama de máquina de estados mostra o processo pelo qual uma pessoa define um compromisso em seu calendário. No estado composto “Verificar data”, o sistema verifica a disponibilidade no calendário em alguns subestados. Se o horário não estiver disponível no calendário, o processo será encerrado. Se houver disponibilidade, no entanto, o compromisso será adicionado ao calendário.

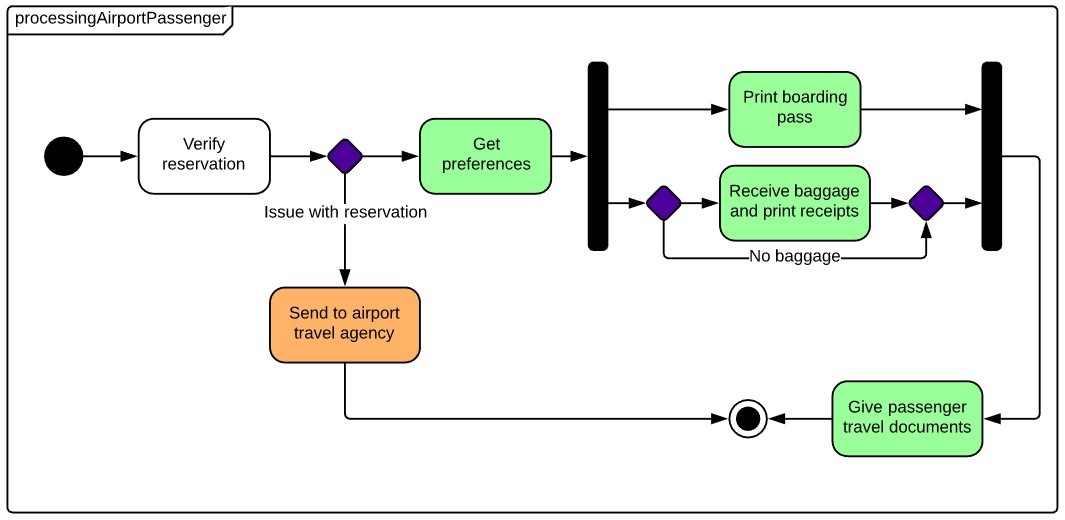
## Exemplo de diagrama de estados para universidades

Este diagrama de estados mostra o processo de matrícula e as aulas em uma universidade. O estado composto de “Matrícula” é formado por diversos subestados que orientarão os alunos no processo de matrícula. Com o aluno matriculado, serão “Ensinados” e, por fim, passarão pelas “Provas finais”.



### Exemplo de diagrama de estados para check-in em aeroportos

O exemplo a seguir simplifica os passos necessários para fazer check-in no aeroporto. Para as companhias aéreas, um diagrama de estados pode ajudar a otimizar processos e eliminar passos desnecessários.



# DIAGRAMA DE ATIVIDADE EM UML

https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml

Diagramas de atividade, junto com diagramas de caso de uso e de máquina de estados, são considerados diagramas de comportamento porque descrevem o que é necessário acontecer no sistema sendo modelado.

As partes interessadas lidam com muitas questões, portanto, é importante se comunicar com clareza e concisão. Diagramas de atividade ajudam a unir as pessoas das áreas de negócios e de desenvolvimento de uma organização para entender o mesmo processo e comportamento. Para criar um diagrama de atividade, é necessário um conjunto de símbolos especiais, incluindo aqueles para dar partida, encerrar, fundir ou receber etapas no fluxo.

## Benefícios de diagramas de atividade

Diagramas de atividade oferecem uma série de benefícios para os usuários.

Considere criar um diagrama de atividade para:

* Demonstrar a lógica de um algoritmo;
* Descrever as etapas realizadas em um caso de uso UML;
* Ilustrar um processo de negócio ou fluxo de trabalho entre usuários e o sistema;
* Simplificar e melhorar qualquer processo ao esclarecer casos de uso complicados; ● Modelar elementos de arquitetura de software, como método, função e operação.

## Componentes básicos de um diagrama de atividade

Antes de criar um diagrama de atividade, é necessário primeiro entender sua composição. Alguns dos componentes mais comuns de um diagrama de atividade incluem:

* **Ações:** É uma etapa da atividade em que o usuário ou software realiza uma determinada tarefa;
* **Nó de decisão:** Um ramo condicional no fluxo representado por um diamante. Inclui uma única entrada e duas ou mais saídas;
* **Fluxos de controle:** Outro nome dado aos conectores que mostram o fluxo entre as etapas no diagrama;
* **Nó inicial:** Simboliza o início da atividade. É representado por um círculo preto;
* **Nó final:** Representa a etapa final da atividade. É representado por um círculo preto delineado.

## Símbolos de diagramas de atividade

Estas formas e estes símbolos de diagramas de atividade são alguns dos tipos mais comuns que você encontrará em diagramas UML.

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Nome** | **Descrição** |
|  | Símbolo de início | Representa o começo de um processo ou fluxo de  trabalho em um diagrama  de atividade. Ele pode ser  usado por si só ou com um  símbolo de nota que explica o ponto de partida. |
|  | Símbolo de atividade | Indica as atividades que compõem um processo  modelado. Estes símbolos, que incluem descrições  breves dentro da forma, são  os principais componentes de um diagrama de atividade. |
|  | Símbolo de conector | Mostra o fluxo de direção, ou fluxo de controle, da atividade. Uma seta de  entrada inicia um passo de uma atividade. Uma vez  concluído o passo, o fluxo continua com a seta de saída. |
|  | Símbolo de junta / Barra de sincronização | Combina duas atividades simultâneas e as reintroduz em um fluxo onde apenas uma atividade ocorre por  vez. Representado por uma linha espessa vertical ou horizontal. |
|  | Símbolo de garfo | Divide um único fluxo de atividade em duas  atividades simultâneas. É simbolizado com várias  linhas com setas de uma junção. |
|  | Símbolo de decisão | Representa uma decisão e sempre tem pelo menos  dois caminhos ramificados e com texto de condição, permitindo aos usuários |

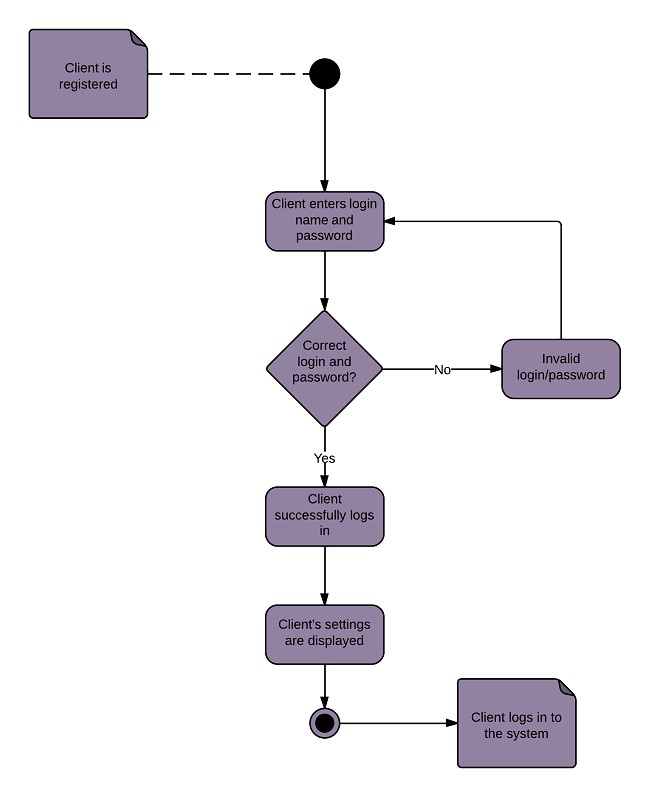
|  |  |  |
| --- | --- | --- |
|  |  | visualizarem opções. Este símbolo representa a  ramificação ou fusão de diferentes fluxos, com o  símbolo atuando como um quadro ou contêiner. |
|  | Símbolo de nota | Permite aos criadores ou colaboradores do diagrama comunicar mensagens  adicionais que não se  encaixam dentro do próprio diagrama. Deixe  observações para uma maior clareza e especificação. |
|  | Símbolo de enviar sinal | Indica que um sinal está sendo enviado a uma atividade recebedora. |
|  | Símbolo de receber sinal | Demonstra a aceitação de um evento. Após o evento  ser recebido, o fluxo que vem desta ação é concluído. |
|  | Símbolo de história rasa pseudoestado | Representa uma transição que invoca o último estado ativo. |
|  | Símbolo de opção em loop | Permite ao criador modelar uma sequência repetitiva  dentro do símbolo de opção em loop. |
|  | Símbolo de final de fluxo | Representa o final de um fluxo de processo  específico. Este símbolo  não deve representar o fim de todos os fluxos em uma atividade. Nesse caso, use o símbolo de término. O  símbolo final do fluxo deve  ser colocado no final de um  processo em um fluxo único de atividade. |
|  | Texto de condição | É colocado ao lado de um marcador de decisão para  avisar em qual condição um fluxo de atividade deve se separar nesse sentido. |
|  | Símbolo de término | Marca o estado final de uma atividade e representa a conclusão de todos os fluxos de um processo. |

## Exemplos de diagramas de atividade

Diagramas de atividade mapeiam fluxos de processos de maneira fácil de entender. Considere os dois exemplos abaixo quando se trata de criar diagramas de atividade UML.

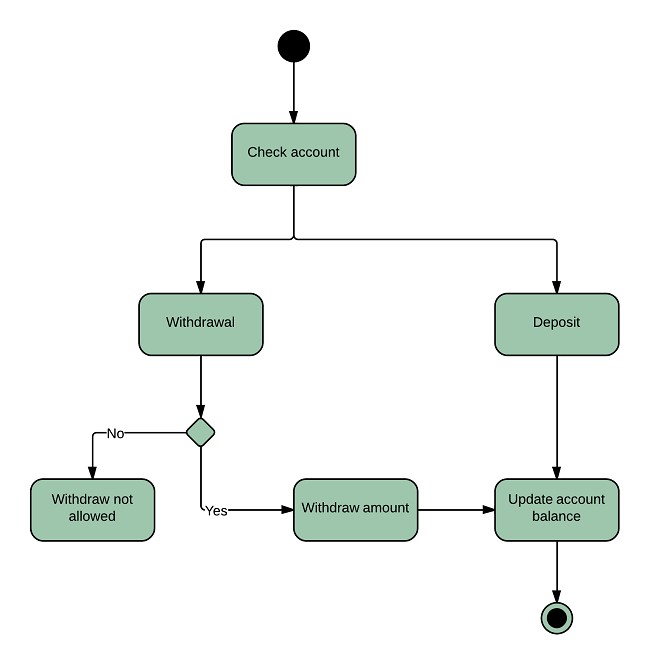
## Diagrama de atividade para uma página de login

Muitas das atividades que as pessoas querem realizar on-line — acessar e-mail, administrar suas finanças, comprar roupas etc. — as obriga a fazer login em um site. Este diagrama de atividade mostra o processo de fazer login em um site, desde a inserção de nome de usuário e senha até entrar no sistema com sucesso. O diagrama utiliza diferentes formas de contêiner para atividades, decisões e notas.



## Diagrama de atividade para um sistema bancário

Este diagrama mostra o processo de saque ou depósito de dinheiro em uma conta bancária. A vantagem de representar o fluxo de trabalho visualmente em UML é a possibilidade de exibir saques e depósitos em um único gráfico.



# DIAGRAMA DE INTERAÇÃO E SEQUÊNCIA EM UML

https://www.lucidchart.com/pages/pt/diagrama-de-interacao-uml

Como o nome já indica, o diagrama de interação é um tipo de diagrama UML usado para mostrar o comportamento interativo de um sistema. Os diagramas de interação descrevem o fluxo de mensagens e fornece contexto para uma ou mais linhas da vida dentro de um sistema. Além disso, os diagramas de interação podem ser usados para representar as sequências organizadas dentro de um sistema e servir como um meio para visualizar dados em tempo real por UML.

## Benefícios do diagrama de interação

Os diagramas de interação podem ser implementados em diversos cenários para fornecer um conjunto exclusivo de informações. Eles podem ser usados para:

* Modelar um sistema como uma sequência de eventos organizada por tempo;
* Fazer engenharia reversa ou avançada de um sistema ou processo;
* Organizar a estrutura de vários eventos interativos;
* Mostrar, de forma simples, o comportamento de mensagens e linhas da vida dentro de um sistema;
* Identificar possíveis conexões entre elementos de linhas da vida.

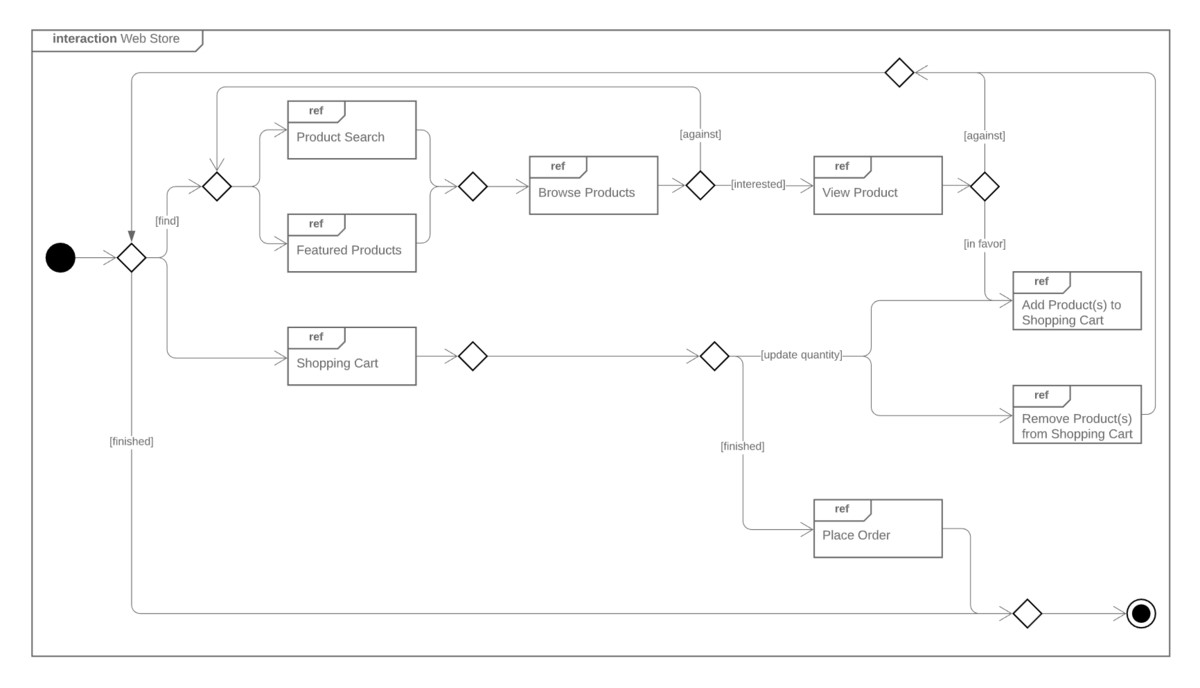
## Tipos de diagramas de interação em UML

Diagramas de interação são divididos em quatro tipos principais de diagramas:

* Diagrama de comunicação;
* Diagrama de sequência;
* Diagrama de tempo;
* Diagrama de visão geral de interação.

## Exemplo de diagrama de interação

Veja um exemplo simples de diagrama de interação que pode ser usado para modelar as interações entre os variados elementos de um aplicativo web básico. Você pode modificar este modelo para visualizar o fluxo de controle de um sistema e mostrar as interações entre os objetos dentro dele.



# DIAGRAMA DE COLABORAÇÃO (COMUNICAÇÃO) EM UML

https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/interacao/colaboracao/diagrama\_d e\_colaboracao.htm

O diagrama de comunicação, também conhecido como diagrama de colaboração, como um dos quatro tipos de diagrama de interação, mostra a interação existente num conjunto de objetos e seus relacionamentos, dando ênfase à organização estrutural dos objetos. O diagrama mostra os objetos das classes que participam da interação, mostrando os vínculos entre os mesmos, descrevendo as mensagens que os objetos recebem e enviam. Os diagramas de comunicação diferenciam-se dos diagramas de seqüências nestes aspectos:

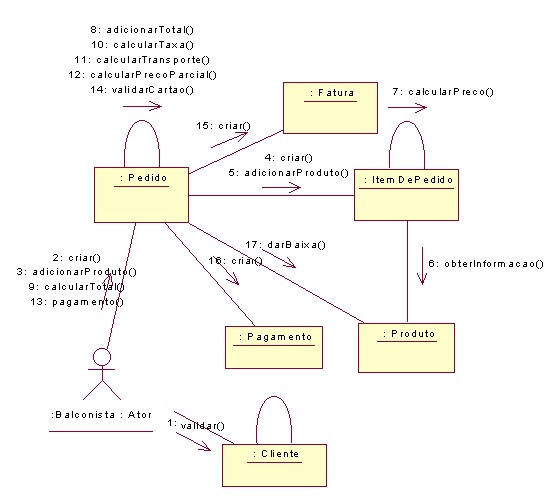
* Existe um caminho que indica como o objeto está vinculado a outro.
* Existe um número de seqüência para indicar a ordem temporal de uma mensagem:

○ 1:validar()

○ 2:criar()

○ …

Abaixo é mostrado o diagrama de comunicação, para o caso de uso Processar Pedido da Virtual LTDA, onde pode-se acompanhar o caminho de interação e as mensagens que são trocadas.



# DIAGRAMA DE TEMPO EM UML

https://medium.com/documenta%C3%A7ao-uml/diagrama-de-tempo-505aed0e8c6f

O diagrama de tempo é um diagrama de interação da UML que tem o propósito de demonstrar as interações do sistema em razão do tempo. Isso auxilia a que o mesmo seja utilizado em processos onde o tempo é um fator crítico. Ele se concentra nas mudanças de estado e condições para tais mudanças dentro do tempo de vida ao longo de um eixo de tempo linear.

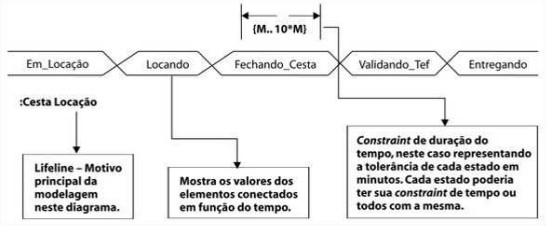
É possível criar diversos diagramas de tempo, em que cada um focalize em uma linha de vida ou visualização diferente da mesma interação. É possível usá-lo para examinar e modelar ainda mais as restrições de tempo e duração.

Na área da programação, o diagrama de tempo pode servir como especificação para os desenvolvedores cumprirem requisitos do sistema como tempo de resposta, performance, processamento paralelo e etc.

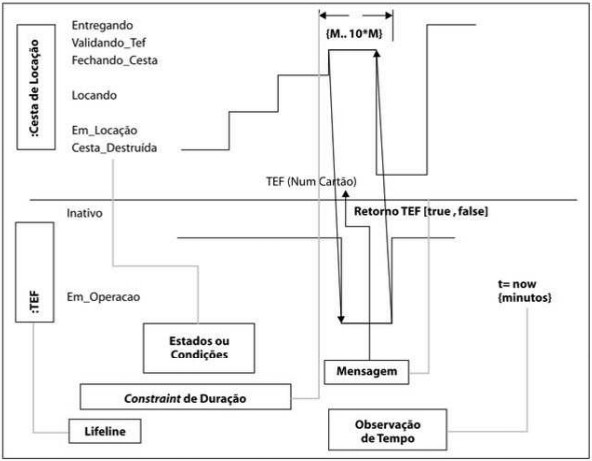
Enquanto o diagrama de sequência é mais adequado a situações onde a cronometragem não é o fator chave, o diagrama de tempo é extremamente útil para processos que envolvem respostas e eventos do negócio que tem limitações de horário, planejamento de tarefas (schedule), sistemas em que a concorrência e a velocidade de acesso sejam de fator crítico e devem ser bem analisados.

O mesmo tem pouca aplicação na modelagem de aplicações comerciais, porém, pode ser de grande ajuda em sistemas que trabalham em tempo real ou que utilizam recursos de multimídia , onde o tempo que levar para executar algo é importante.

## Exemplo de fluxo do contexto do elemento foco



## Exemplo de detalhamento do elemento foco do diagrama



**Referencias:**

UML Class Diagram Tutorial: https://www.lucidchart.com/pages/uml-class-diagram