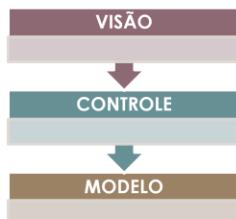
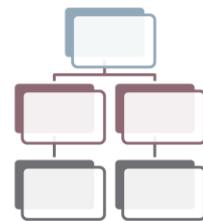


Arquitetura e Desenho de Software

AULA 07



Profa. Milene Serrano



Agenda



Considerações Iniciais

Abordagens Tradicionais

- Modelagem Orientada a Objetos
- UML (*Unified Modeling Language*)

Considerações Finais

- Conheceremos em tempo de aula, principalmente, os diagramas de classe e sequência.
- Demais diagramas estão colocados visando estudos extras sobre a notação da UML, por parte dos alunos.
- Qualquer dúvida, por gentileza, entrar em contato.

Considerações Iniciais



Considerações Iniciais

Existem várias propostas de abordagens anteriores à Orientação a Objetos.

Mas, nesse curso, iremos focar em modelos comumente utilizados no desenho de soluções orientadas pelo Paradigma Orientado a Objetos.

Nesses cenários, destaca-se a UML (**Unified Modeling Language**). Trata-se de uma linguagem (uma notação) de modelagem muito reconhecida e utilizada na comunidade de software.

A UML foi criada visando estabelecer uma linguagem visual comum no complexo mundo do desenvolvimento de software, que também poderia ser compreendida por usuários das áreas de negócio e demais envolvidos na elaboração de produtos de software.

Considerações Iniciais

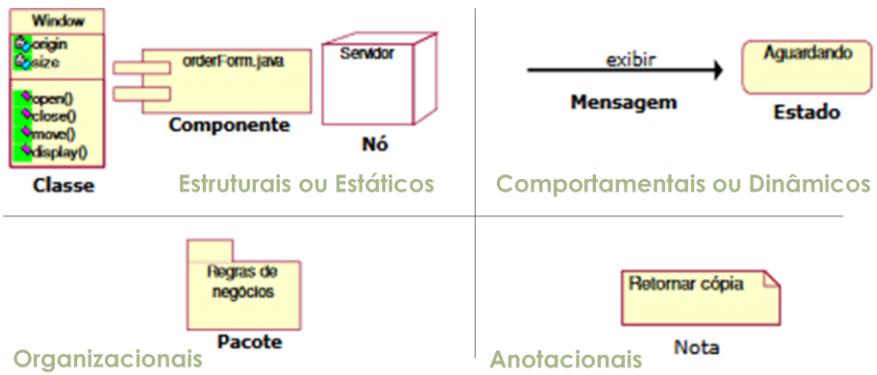
A UML não é, portanto:

- um processo;
- uma linguagem de programação;
- uma metodologia, ou
- algo nesse sentido.

É, na verdade, uma notação visual, com diagramas que ajudam a desenhar uma solução computacional em diferentes níveis de abstração e/ou perspectivas...

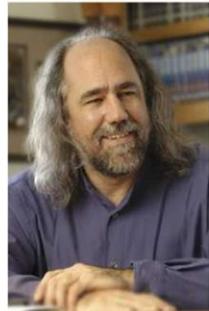
Muitos autores apresentam a UML, em uma visão mais otimizada, como um conjunto de diagramas para detalhamento estrutural (ou estático), dinâmico (ou comportamental); organizacional (ou em pacotes), e anotacional (ou explicativo, com uso de notas) no intuito de modelar uma solução computacional.

Considerações Iniciais



Como são muitos os diagramas oferecidos pela UML, iremos focar apenas em alguns, sendo esses os mais utilizados em modelagens estáticas e dinâmicas de soluções orientadas a objetos e associadas.

Considerações Iniciais



grady booch



ivar jacobson

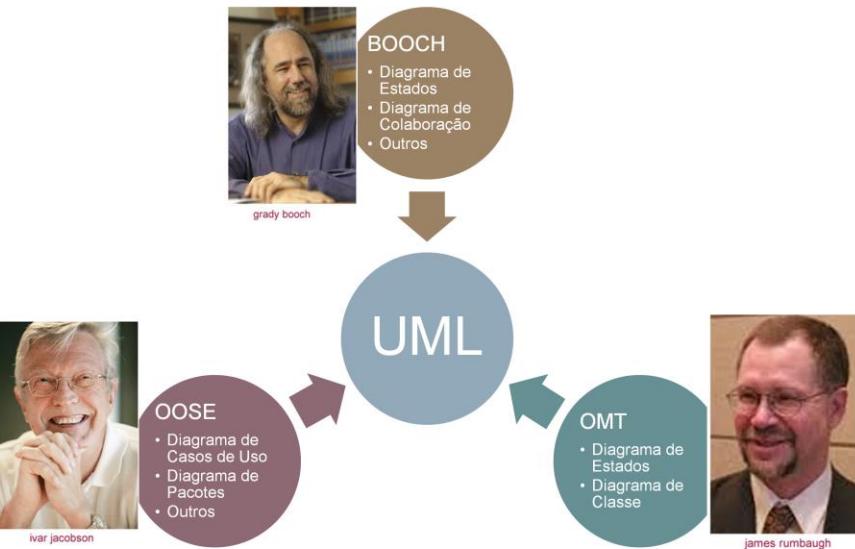


james rumbaugh

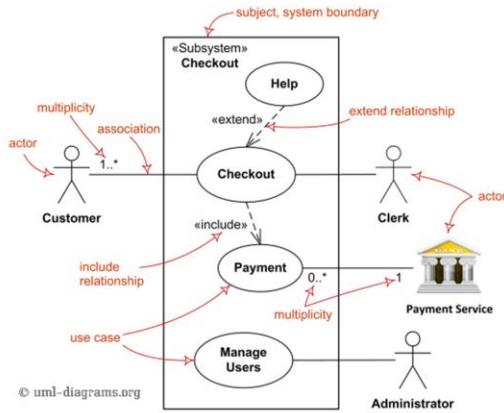
Vale ressaltar que a UML, em sua origem, procurou reunir as "melhores práticas de engenharia" que provaram ter sucesso na modelagem de sistemas grandes e complexos, na época.

Portanto, está baseada nos conceitos de Booch, OMT (Rumbaugh) e OOSE (Jacobson), fundindo-os em uma única linguagem de modelagem, comum e largamente utilizada.

Considerações Iniciais



Considerações Iniciais



Vocês já conhecem bem um diagrama dinâmico da UML, o qual revela uma parte comportamental da solução computacional. Trata-se do Diagrama de Casos de Uso! Lembram-se dele? :)

Diagrama de Classe



Diagrama de Classe

É um diagrama estático da UML que reúne os elementos mais importantes de um sistema orientado a objetos.

Nele são exibidos um conjunto de classes, interfaces e seus relacionamentos.

As classes especificam tanto as propriedades quanto os comportamentos dos objetos.

<https://www.uml-diagrams.org/class-diagrams-overview.html>

Diagrama de Classe

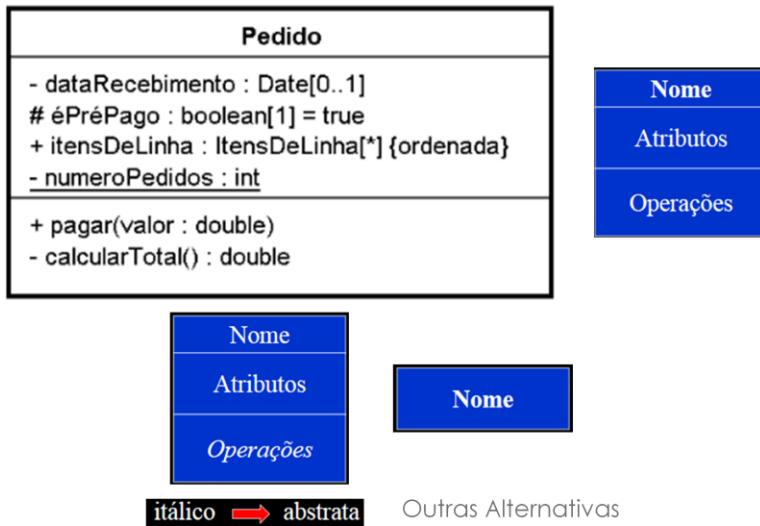
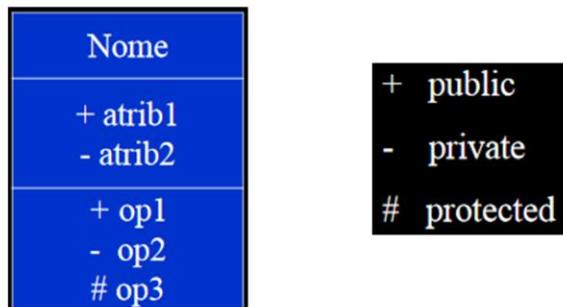


Diagrama de Classe



Especificações de Acesso

Diagrama de Classe

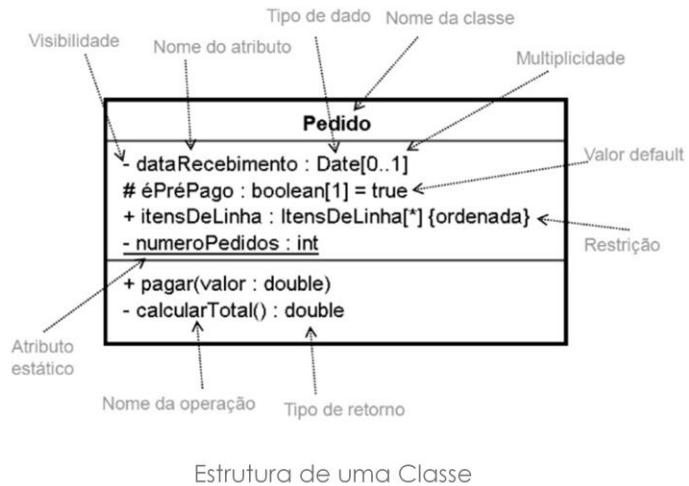


Diagrama de Classe

Relacionamentos:

- Dependência
- Associação
- Agregação
- Composição
- Generalização
- Realização

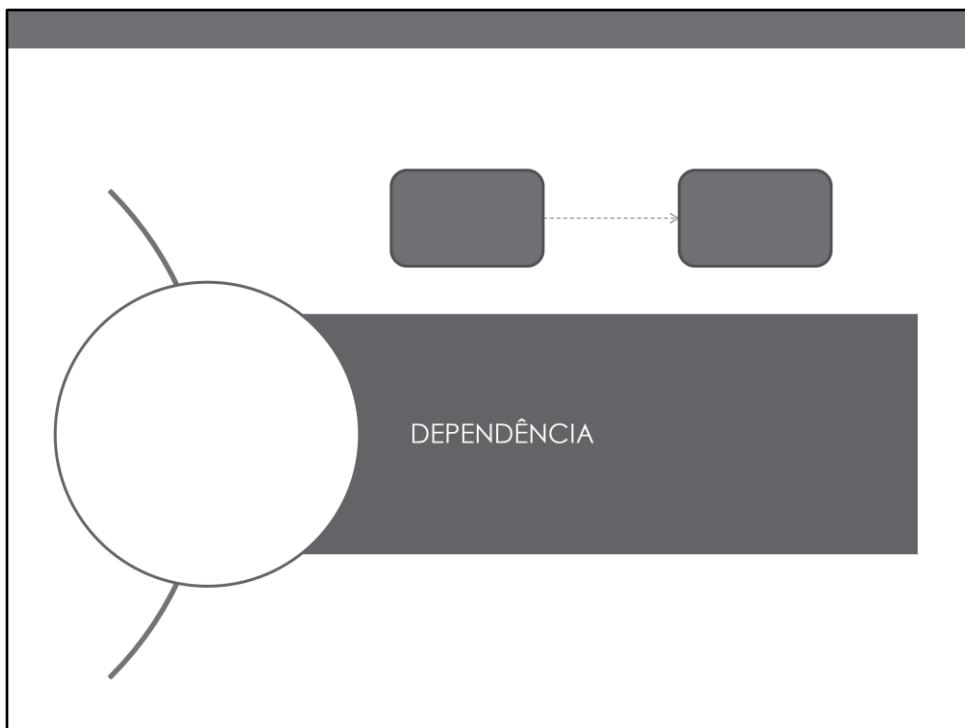
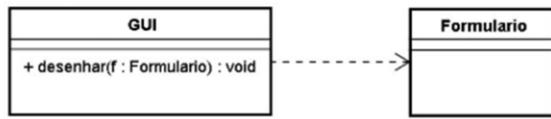


Diagrama de Classe



```
public class GUI {
    public void desenhar(Formulario f) {
        f.pintarBotao();
        f.pintarMenu();
        f.pintarJanelas();
        ...
    }
}
```

Dependência

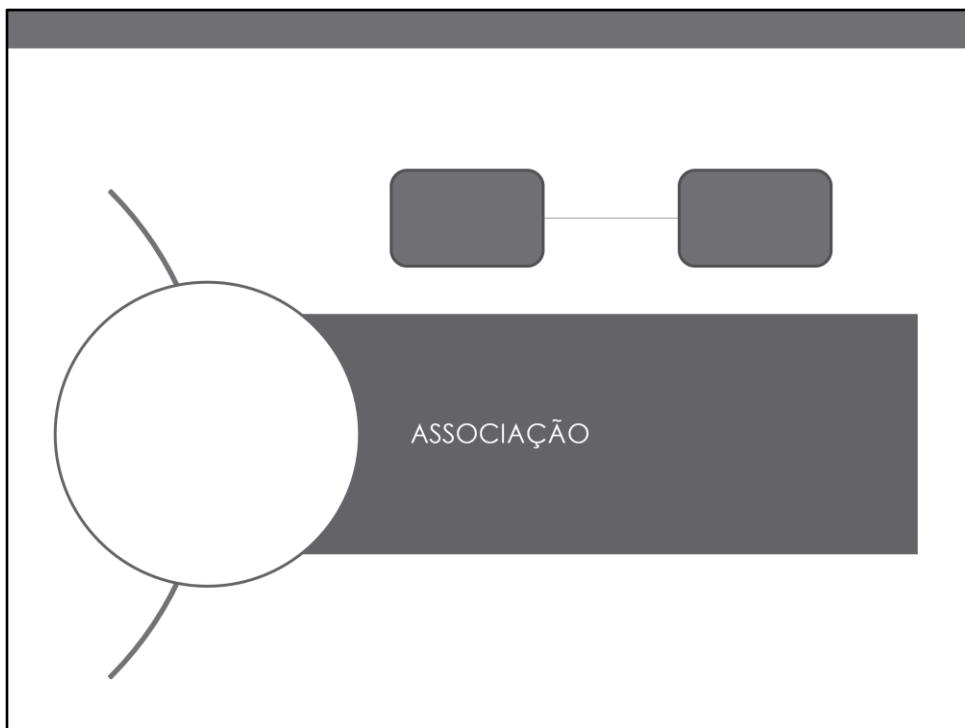
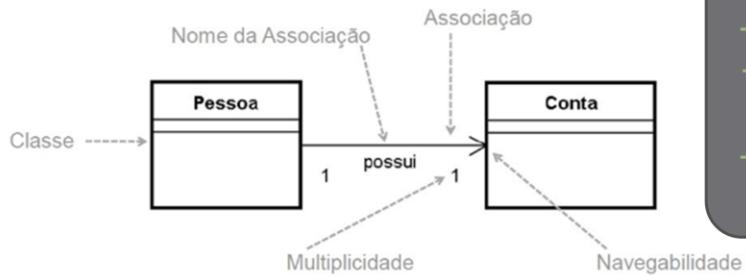


Diagrama de Classe



Associação

- PESSOA POSSUI CONTA (voz ativa, sentido da naveabilidade)
- CONTA É_POSSUÍDA PELA PESSOA (voz passiva, naveabilidade oposta)

Diagrama de Classe



UM CIDADÃO PODE SOLICITAR APENAS UM CPF (1 .. 1)
O MESMO CPF SÓ PODE SER SOLICITADO POR UM CIDADÃO (1 .. 1)
RESULTANTE: 1.. 1

Associação 1..1 (UM PARA UM)

- CIDADÃO SOLICITA CPF (voz ativa, sentido da naveabilidade)
- CPF É_SOLICITADO PELO CIDADÃO (voz passiva, naveabilidade oposta)

Diagrama de Classe



UM CLIENTE PODE FAZER VÁRIOS PEDIDOS (1 .. *)
O MESMO PEDIDO SÓ PODE SER FEITO POR UM CLIENTE (1 .. 1)
RESULTANTE: 1 .. *

Associação 1..N ou 1..* (UM PARA MUITOS)

- CLIENTE FAZ PEDIDO (voz ativa)
- PEDIDO É_FEITO PELO CLIENTE (voz passiva)

Diagrama de Classe



UM COMEDIANTE CONTA VÁRIAS PIADAS (1 .. *)
A MESMA PIADA PODE SER CONTADA POR VÁRIOS COMEDIANTES (1 .. *)
RESULTANTE: * .. *

Associação N .. N ou * .. * (MUITOS PARA MUITOS)

- COMEDIANTE CONTA PIADA (voz ativa)
- PIADA É_CONTADA PELO COMEDIANTE (voz passiva)

Diagrama de Classe

Existem outras cardinalidades ou multiplicidades....

- 1 .. 2
- 1 .. [0 .. 1]
- 1 .. [2 .. 5]
- e outras combinações...

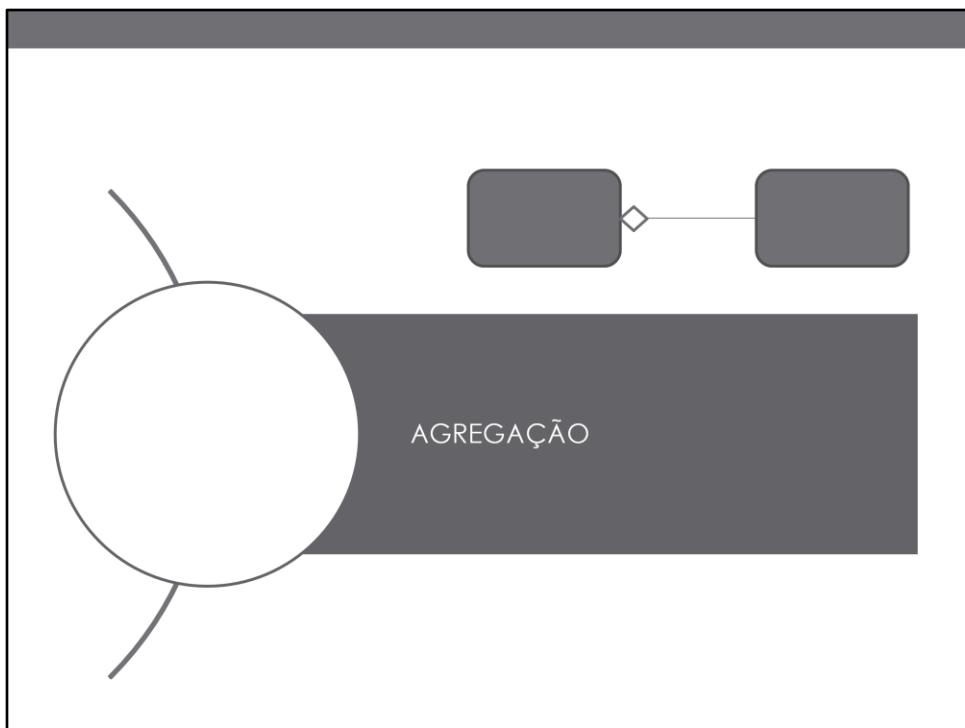
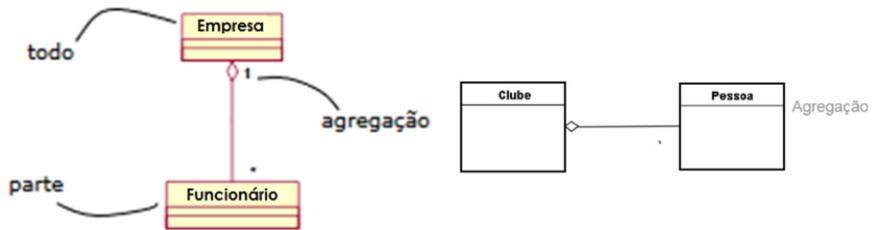


Diagrama de Classe



Agregação

- A EMPRESA EXISTE, INDEPENDENTEMENTE DE EXISITIR UM OU MAIS FUNCIONÁRIOS, nesse caso...

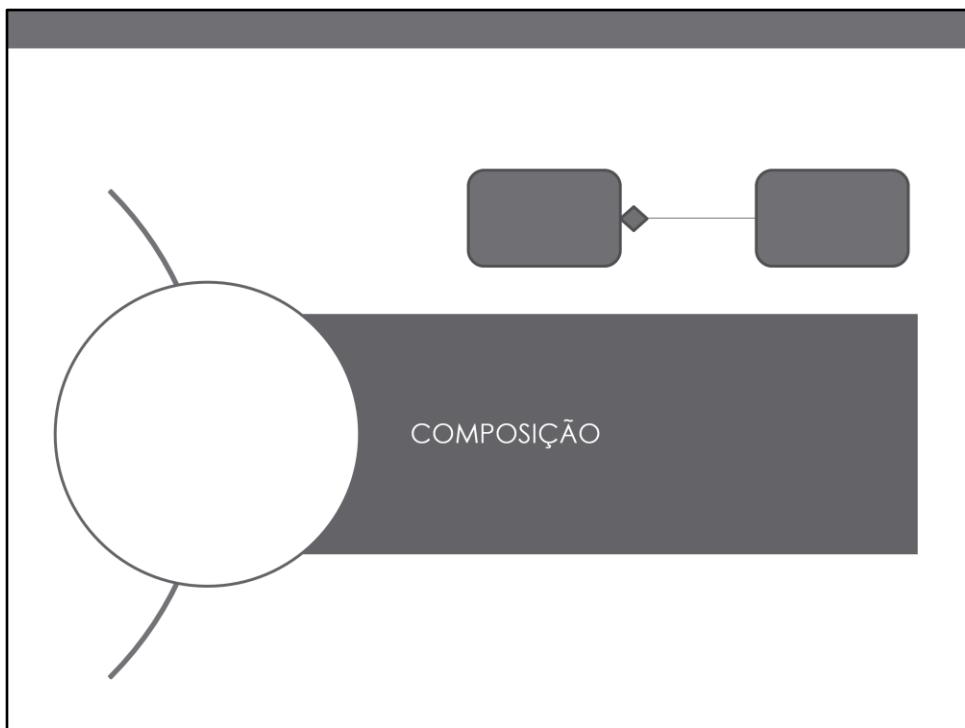


Diagrama de Classe



Composição

- NÃO EXISTE CARRO SEM MOTOR, nesse caso...

Diagrama de Classe

Tanto agregação, quanto composição podem ter diferentes cardinalidades...

- 1 .. 1
- 1 .. *
- e outros

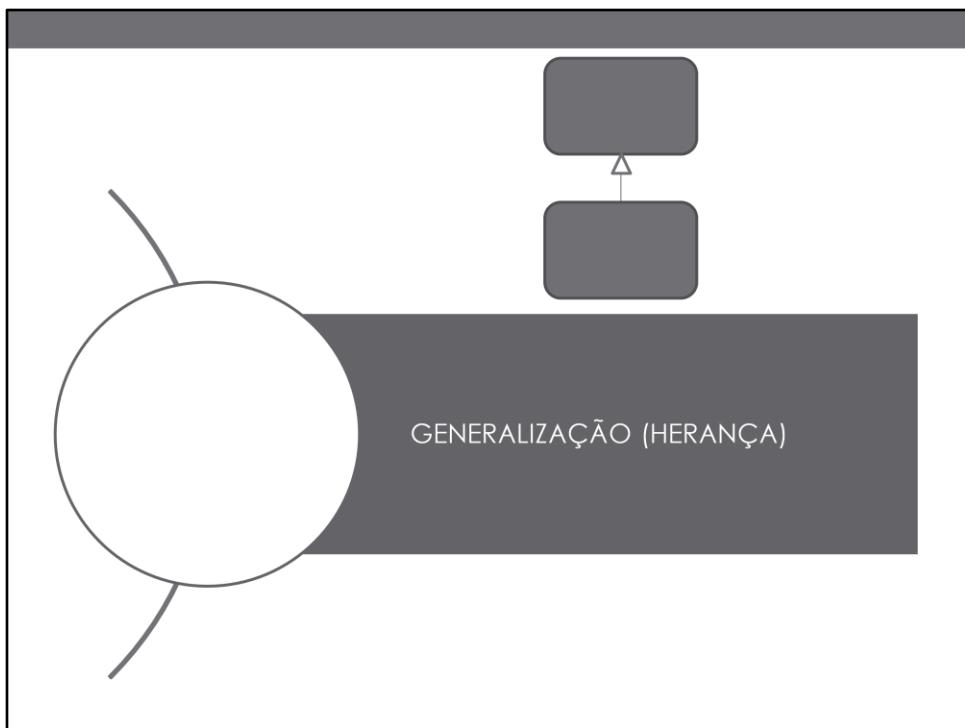
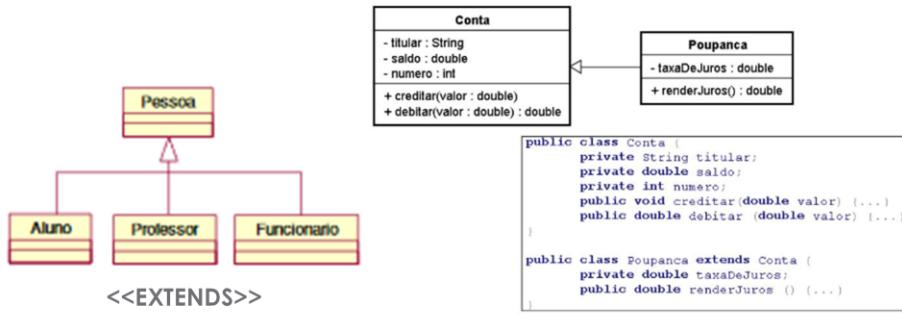


Diagrama de Classe



Generalização

- ALUNO É_UMA PESSOA
- PROFESSOR É_UMA PESSOA
- FUNCIONÁRIO É_UMA PESSOA

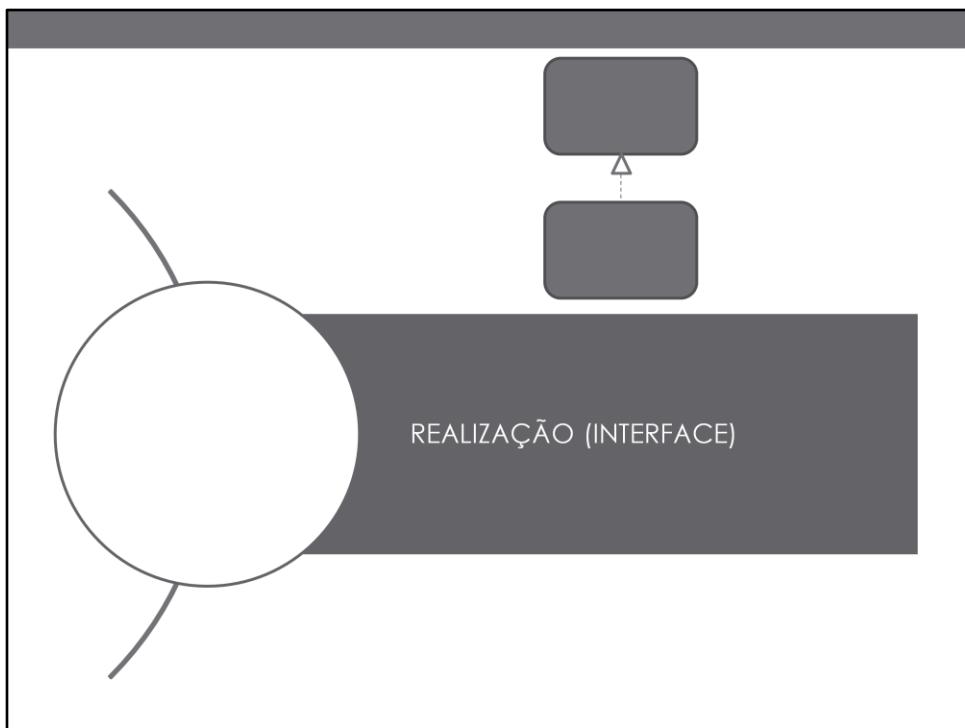


Diagrama de Classe

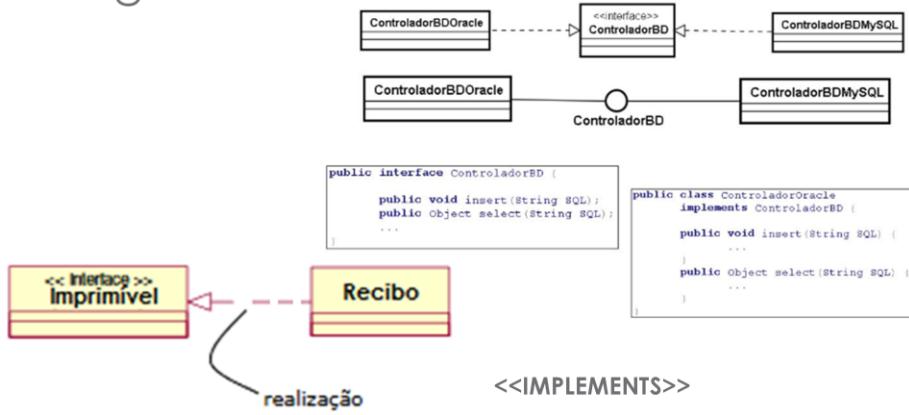


Diagrama de Classe

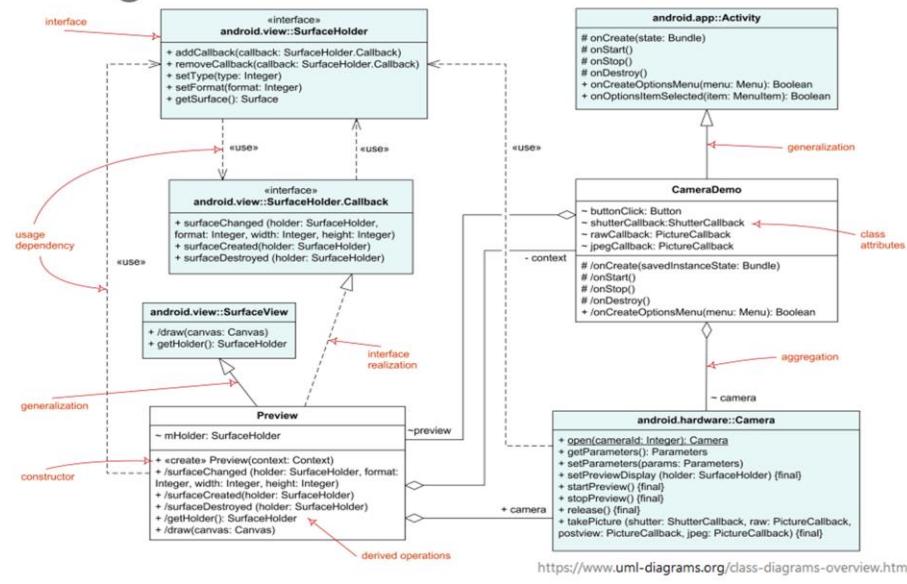
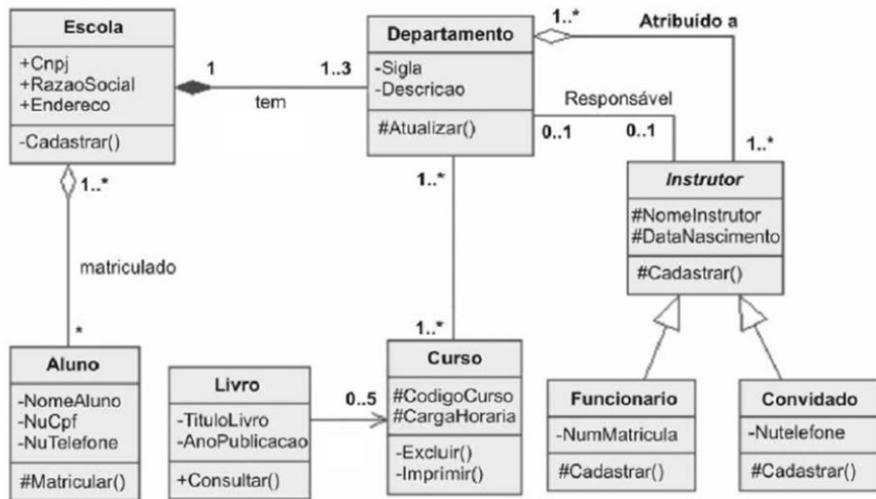


Diagrama de Classe



www.ricardobarcelar.com.br/aulas/eng_sw/mod3-uml.pdf

Diagrama de Seqüênci

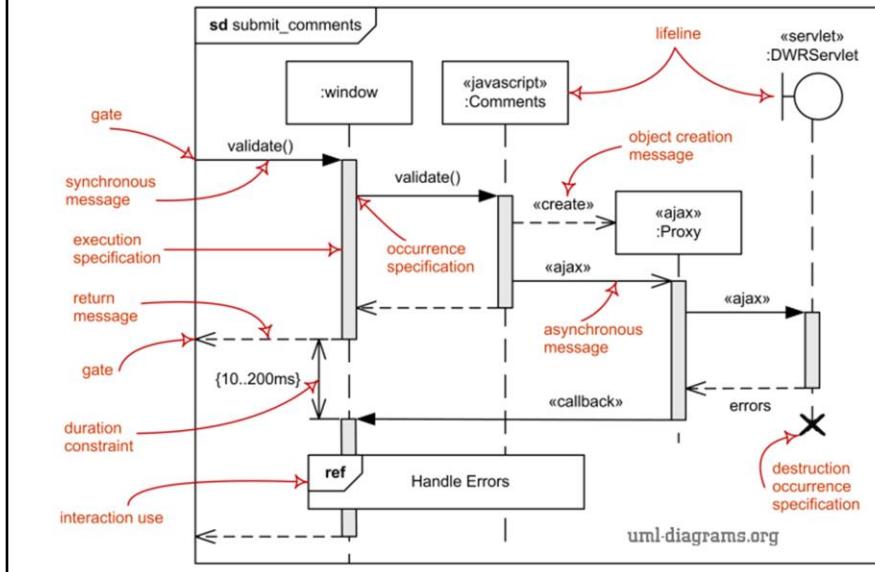


Diagrama de Seqüência

Trata-se do diagrama dinâmico mais relevante na UML, em especial, para fase de projeto do software. Confere, dentre outras particularidades, a representação das interações entre os objetos, distribuídas entre várias linhas de vida.

<https://www.uml-diagrams.org/sequence-diagrams.html>

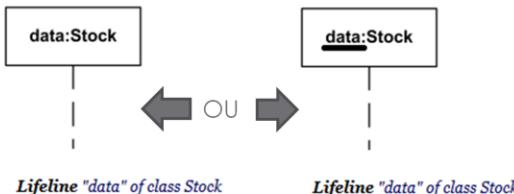
Diagrama de Seqüência



uml-diagrams.org

Diagrama de Seqüência

© uml-diagrams.org



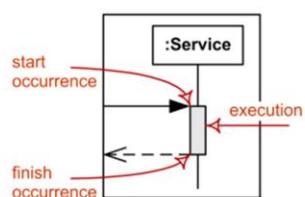
Anonymous lifeline of class User Lifeline "data" of Anonymous class

Representação de Objeto e Classe

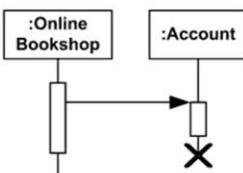
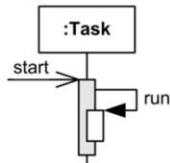
Representação de Objeto e Classe

Diagrama de Seqüência

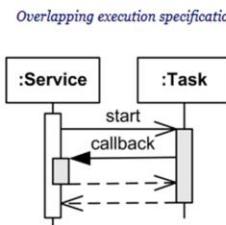
© uml-diagrams.org



Duration of an execution is represented by two execution occurrences - start and finish.



Account lifeline is terminated



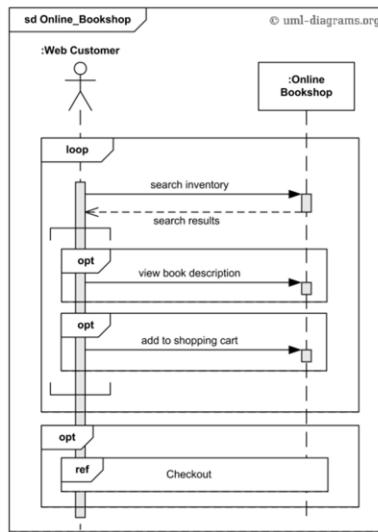
Overlapping execution specifications on the same lifeline - callback message.

Linha de Execução ou de Vida ou Caixa de Ativação (detalhes)

Linha de Execução ou de Vida ou Caixa de Ativação (detalhes)

Diagrama de Seqüênciа

Loop



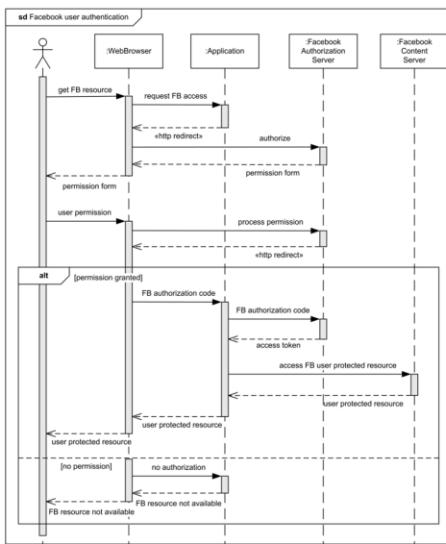
© uml-diagrams.org

Loop

Diagrama de Seqüência

Fluxo Alternativo

© uml-diagrams.org



Loop

Diagrama de Pacotes



Diagrama de Pacotes

Trata-se de mais um diagrama estrutural, estático, o qual permite organizar o sistema como se representasse uma visão em módulos.

<https://www.uml-diagrams.org/package-diagrams-overview.html>

Diagrama de Pacotes

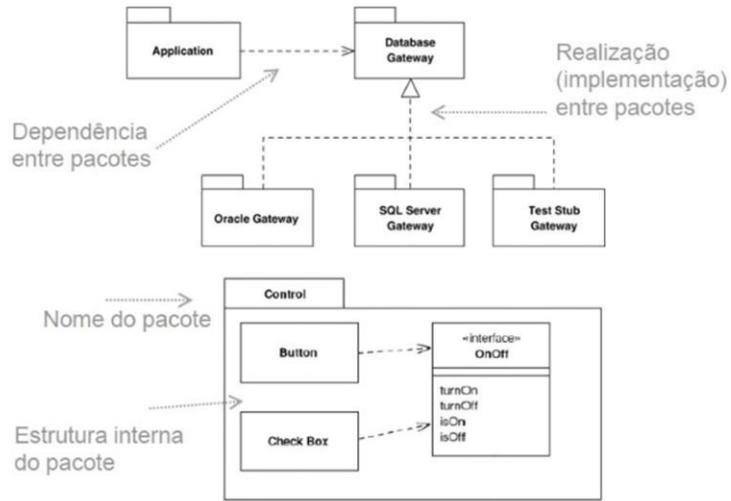


Diagrama de Pacotes

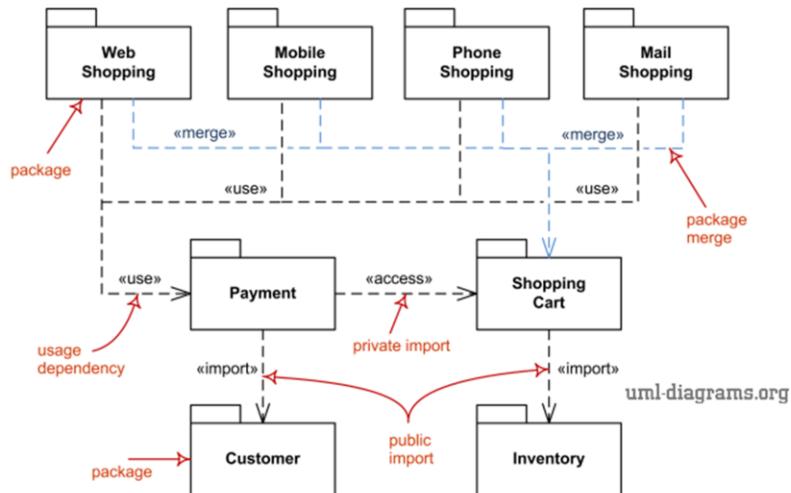


Diagrama de Pacotes

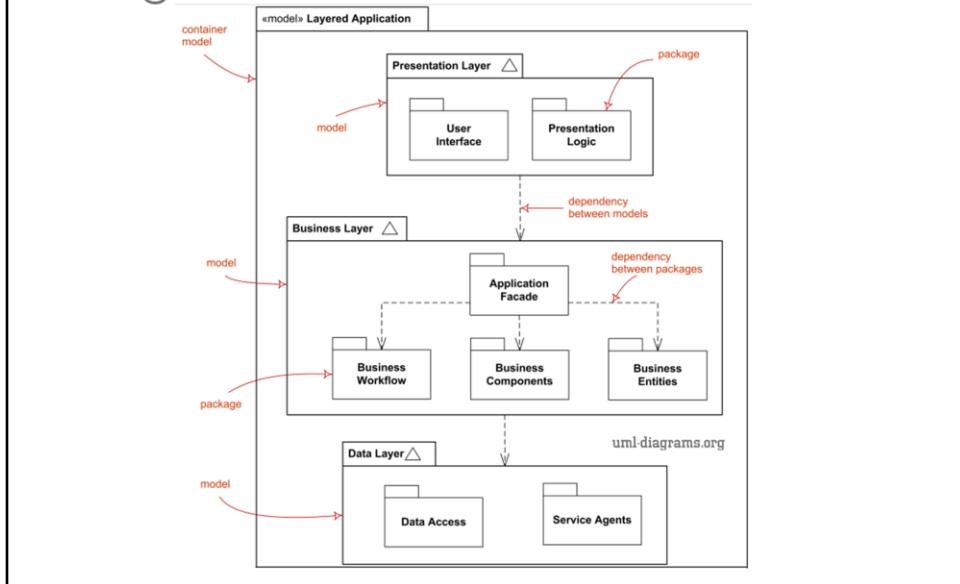


Diagrama de Colaboração



Diagrama de Colaboração

Um diagrama dinâmico que mostra a interação entre objetos e/ou partes usando mensagens sequenciais. Ênfase na ordem das mensagens, normalmente, com foco em um cenário de uso.

<https://www.uml-diagrams.org/communication-diagrams.html>

Diagrama de Colaboração

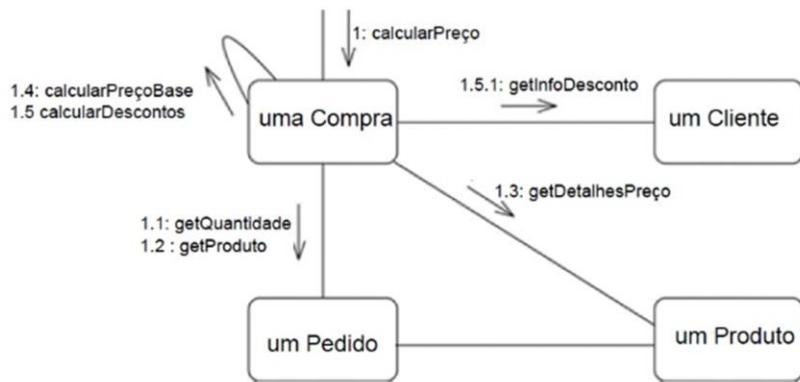
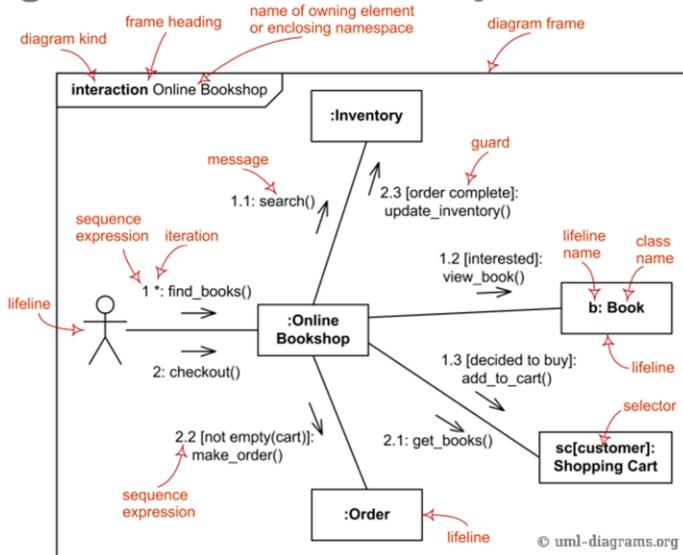


Diagrama de Colaboração



© uml-diagrams.org

Diagrama de Atividades



Diagrama de Atividades

Também um diagrama dinâmico, mas esse mostra o fluxo entre objetos, ou seja, a parte comportamental, representando-a como um fluxo de ações. O foco, portanto, está nos procedimentos, processos de negócio e fluxo de trabalho.

<https://www.uml-diagrams.org/activity-diagrams.html>

Diagrama de Atividades



Diagrama de Atividades

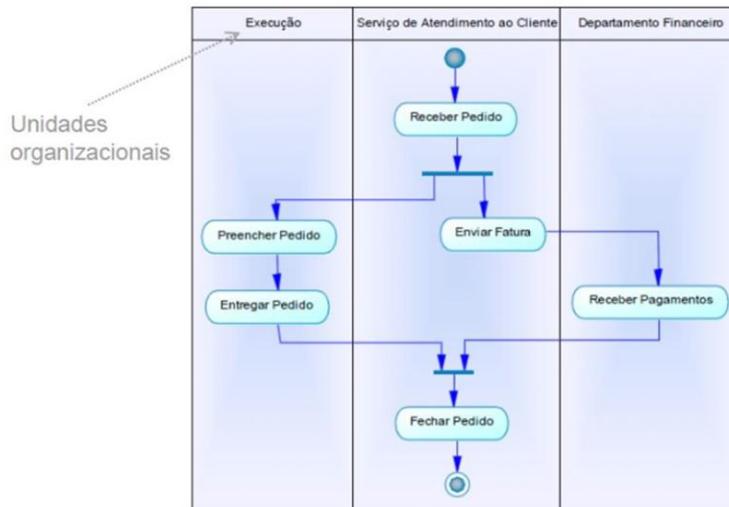


Diagrama de Estados

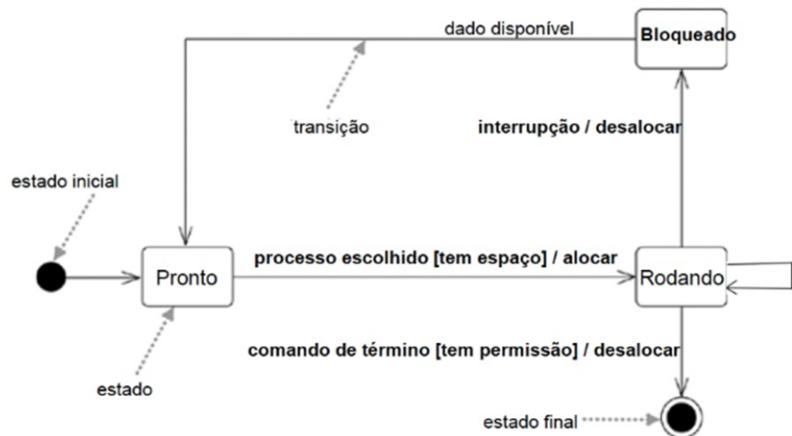


Diagrama de Estados

Também um diagrama dinâmico. Esse procura apresentar os vários estados pelos quais um objeto pode passar. Ao longo do tempo, um objeto muda de estado quando acontece algum evento interno ou externo ao sistema.

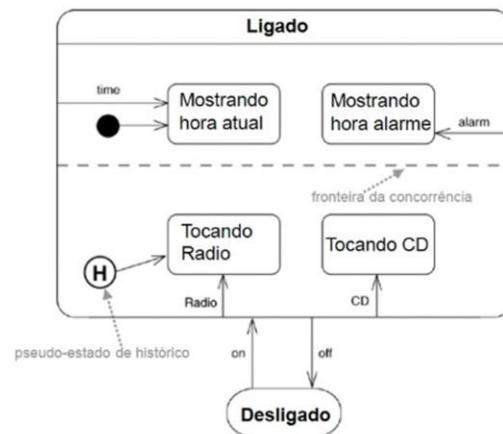
<https://www.uml-diagrams.org/state-machine-diagrams.html>

Diagrama de Estados



www.ricardobarcelar.com.br/aulas/eng_sw/mod3-uml.pdf

Diagrama de Estados



www.ricardobarcelar.com.br/aulas/eng_sw/mod3-uml.pdf

Considerações Finais



Considerações Finais

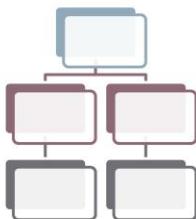
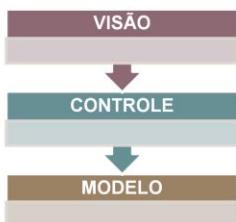
- Nessa aula, apresentamos os principais diagramas da notação UML.
- Existem vários outros artefatos de requisitos e desenho/projeto de software. Acessem:
 - <https://www.uml-diagrams.org/>
- Continuem os estudos!

Referências



Referências

- UML:
 - <https://www.uml-diagrams.org/>
 - https://docs.kde.org/trunk4/pt_BR/kdesdk/umbrello/uml-basics.html



FIM

Dúvidas?

CONTATO:
mileneserrano@unb.br
ou
mileneserrano@gmail.com

Sugestões?

