

# MODELAGEM DE SOFTWARE

Prof. Saulo Popov Zambiasi  
Prof. Richard Henrique de Souza  
Prof. Ricardo Ribeiro Assink  
Prof. Edson Lessa



# Modelagem de Software



# Oracle Academy

Database Foundations  
Learner



Percurso de Aprendizagem - Currículo

**DFo Database Foundations  
do Aluno - Português**

[Detalhes](#)

Database Design  
Learner

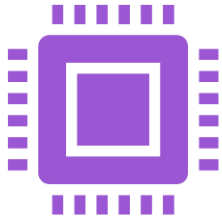


Percurso de Aprendizagem - Currículo

**DD Database Design do  
Aluno - Português**

[Detalhes](#)

Dúvidas ou acesso, enviar Email para:  
[richard.souza@animaeducacao.com.br](mailto:richard.souza@animaeducacao.com.br)



## Por que fazer Modelagem?

Para entregar um software que satisfaça ao propósito pretendido...

Para desenvolver software de Qualidade duradoura ...

Para desenvolver de forma rápida...



## Qual é o nosso principal produto?

O principal produto é um bom **software** capaz de satisfazer às necessidades de seus usuários e respectivos negócios.

A modelagem é uma parte central de todas as atividades que levam à implementação de um bom software.

*“Se você realmente quiser construir software equivalentes a uma casa ou a um prédio, o problema não se restringirá a uma questão de escrever uma grande quantidade de software – de fato, o segredo estará em criar o código correto e pensar em como será possível elaborar menos software. Isso faz com que o desenvolvimento de software de qualidade se torne uma questão de arquitetura, processo e ferramenta” (UML – Guia do Usuário)*

Um modelo é uma simplificação da realidade. Construimos modelos para compreender melhor o sistema que estamos desenvolvendo.



# Modelagem de software

1. Os modelos ajudam a visualizar;
2. Permitem especificar a estrutura ou o comportamento de um sistema;
3. Proporcionam um guia para a construção do sistema;
4. Documentam as decisões tomadas.



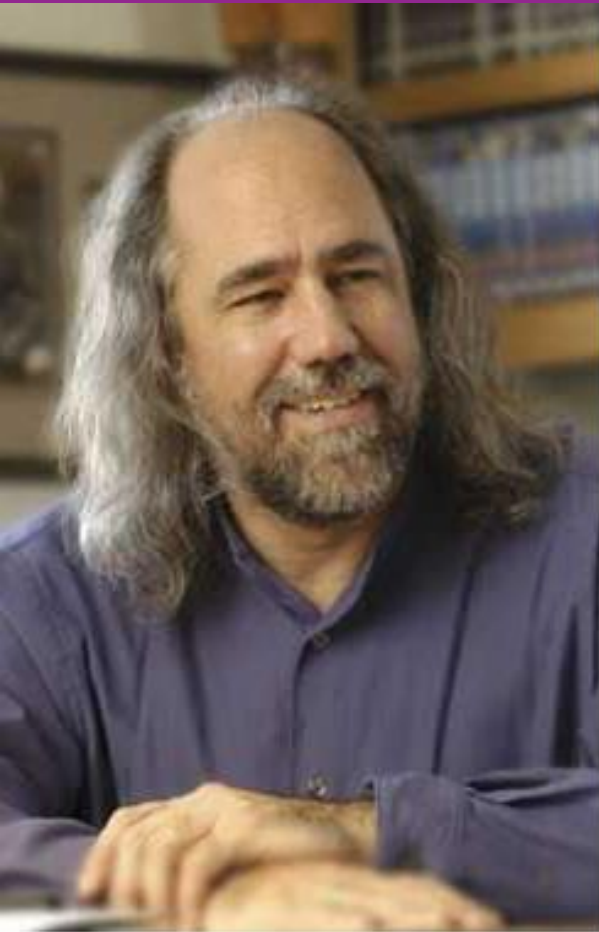
# Unified Modeling Language

UML

O que é?  
Motivadores.

# A história

A UML foi desenvolvida por Grady Booch, James Rumbaugh, e Ivar Jacobson. Sendo que a OMG (Object Management Group), em 1997, unificou na UML diferentes notações existentes na época.



grady booch



ivar jacobson



james rumbaugh

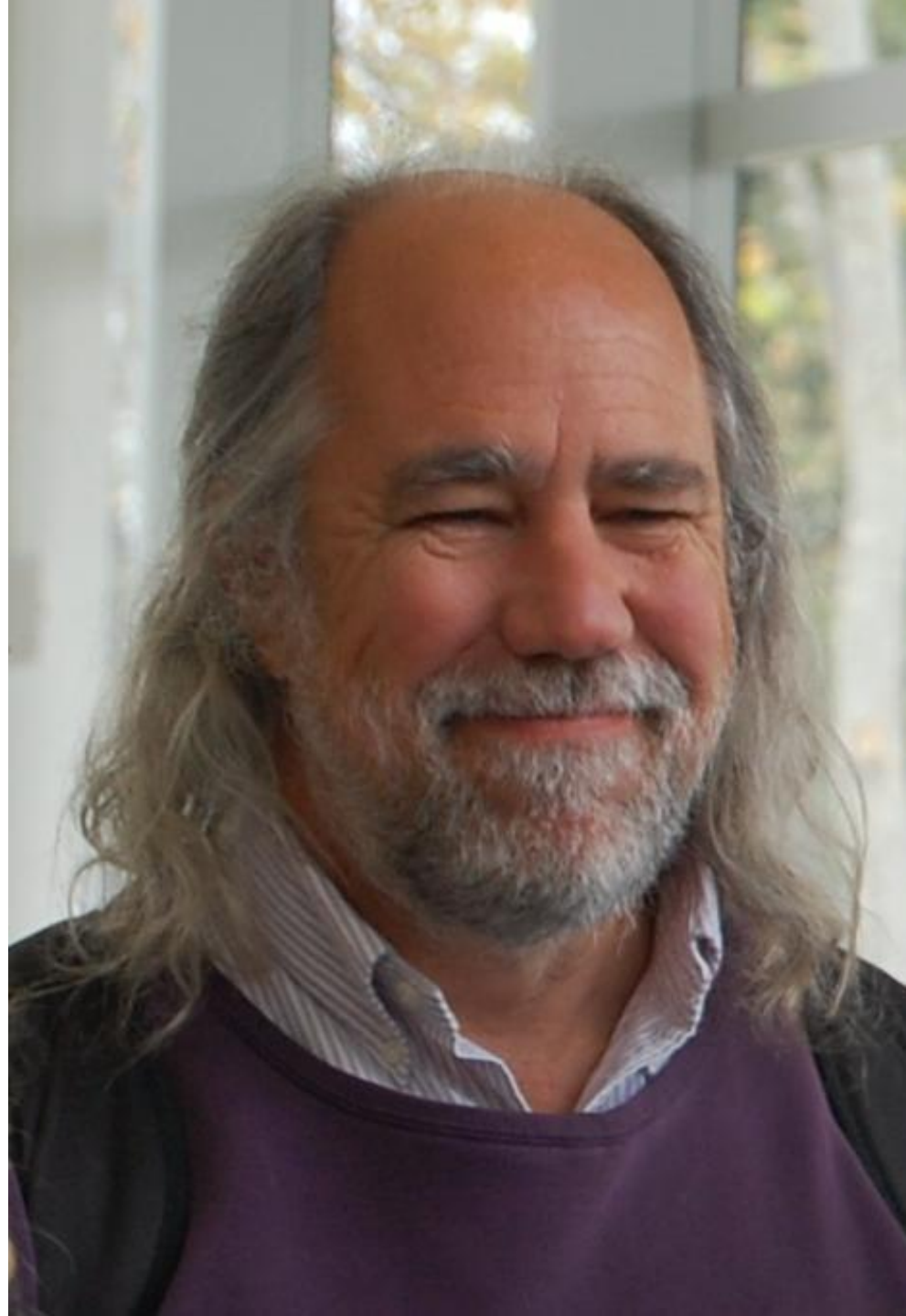


# Curiosidades

Principais metodologias que se tornaram populares nos anos 90:

## **Método de Booch**

Ele definiu a noção para que um sistema fosse analisado a partir de um número de visões, onde cada visão é descrita por um número de modelos e diagramas. O Método de Booch trazia uma simbologia complexa de ser desenhada a mão, continha também o processo pelo qual sistemas são analisados por macro e micro visões.



# Curiosidades

## Object Modelling Technique (OMT)

Um método desenvolvido pela GE (General Electric) onde James Rumbaugh trabalhava. O método é especialmente voltado para o teste dos modelos, baseado nas especificações da análise de requisitos do sistema. O modelo total do sistema baseado no método OMT é composto pela junção dos modelos de objetos, funcional e use-cases.



# Curiosidades

## OOSE/Objectory

Foram desenvolvidos baseados no mesmo ponto de vista formado por Ivar Jacobson. O método OOSE é a visão de Jacobson de um método orientado a objetos, já o Objectory é usado para a construção de sistemas tão diversos quanto eles forem. Ambos os métodos são baseados na utilização de use-cases, que definem os requisitos iniciais do sistema, vistos por um ator externo.

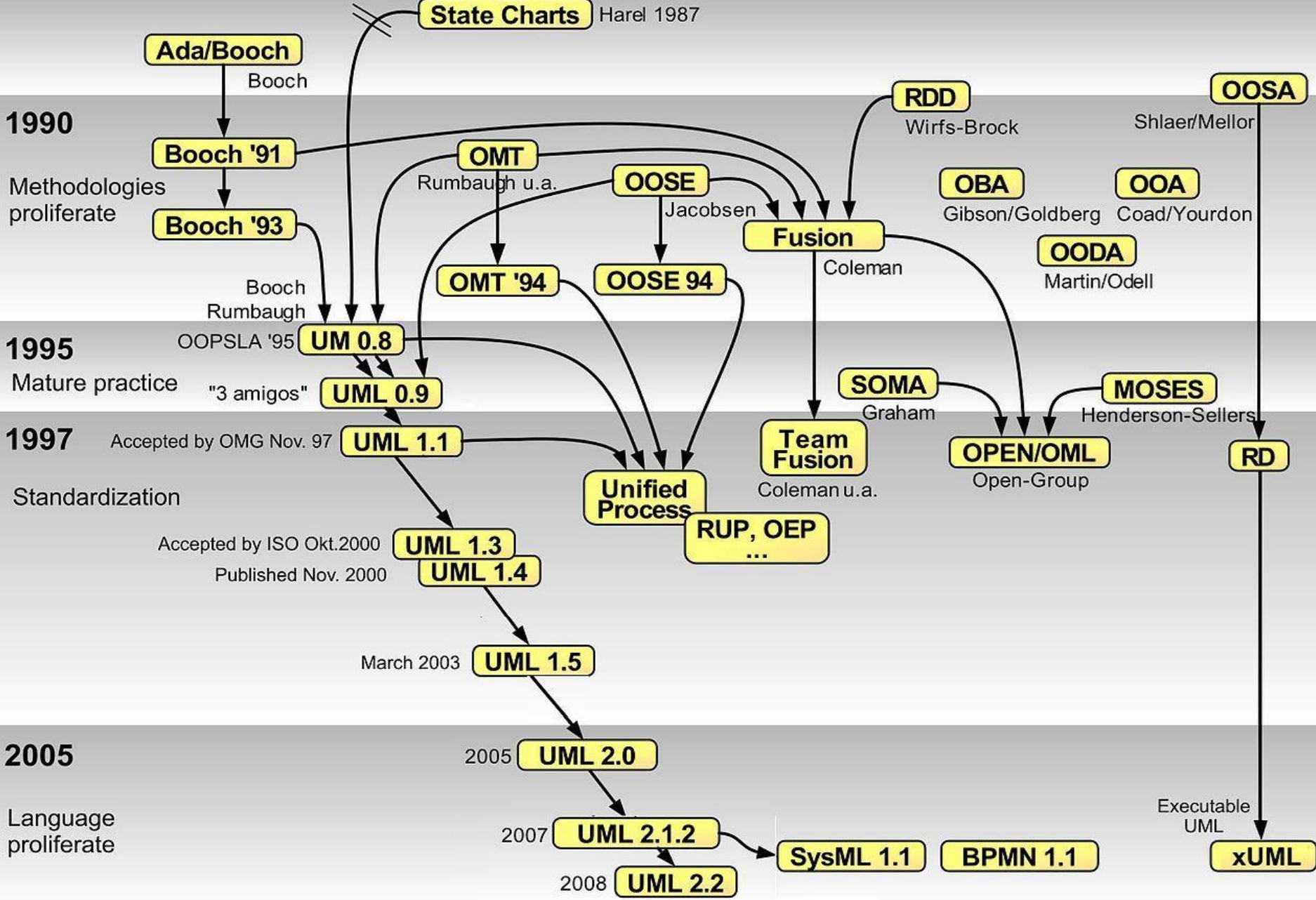


# A história

Diversas notação, processos, e ferramentas para descrição do projeto orientado a objetos foram proposta nos anos 80 e 90.

A UML era uma tentativa de padronizar a modelagem para que qualquer sistema possa ser descrito corretamente de forma consistente e clara. Existiam várias metodologias de modelagem orientada a objetos que causavam uma guerra entre a comunidade de desenvolvedores orientado a objetos.

A UML acabou com esta guerra trazendo as melhores ideias de cada uma destas metodologias, e mostrando como deveria ser a migração de cada uma para a UML.



Fonte: Guido Zockoll, Axel Scheithauer & Marcel Douwe Dekker (Mdd), 2009  
 Disponível em: <[http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)>



HOME

TRAINING

VENDORS

OMG WEB

MEMBERSHIP

CONTACTS

INTRODUCTION

# CELEBRATING 20 YEARS OF UML 1.1

One of the key figures in the software development community, Grady Booch describes the "grand journey" that led to the development of OMG UML standard.

WATCH VIDEO

20 YEARS  
ANNIVERSARY

WHAT IS  
UML?

UML  
VENDOR

UML  
RESOURCES

UML  
SPECIFICATIONS

OMG UML  
CERTIFICATION

Versão atual da UML <<http://www.uml.org>>



# Os objetivos da UML

Modelar sistemas (não apenas de software) usando os conceitos da orientação a objetos;

Estabelecer uma união fazendo com que métodos conceituais sejam também executáveis;

Criar uma linguagem de modelagem usável tanto pelo homem quanto pela máquina.

# Os objetivos da UML



A UML é dominante na indústria. Ela está totalmente baseada em conceitos e padrões extensivamente testados provenientes das metodologias existentes anteriormente, e também é muito bem documentada com toda a especificação da semântica da linguagem representada em meta-modelo.



# Partes que compõem a UML

## Visões

Mostram diferentes aspectos do sistema que está sendo modelado.

A visão é uma abstração consistindo em uma série de diagramas.

Definindo um número de visões, cada uma mostrará aspectos particulares do sistema, dando enfoque a ângulos e níveis de abstrações diferentes e uma figura completa do sistema poderá ser construída.

Visão Lógica  
(Logical View)

Visão de Processo  
(Process View)

Visão de caso de uso  
(Use Case View)

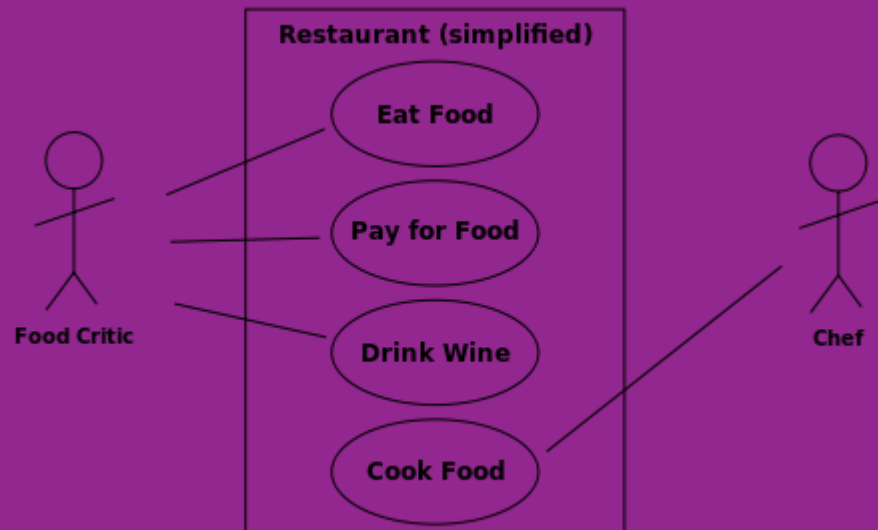
Visão Física  
(Physical View)

Visão de Desenvolvimento  
(Development View)

# Visões da UML

## Visão de use-case

Descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários). A visão use-case é central, já que seu conteúdo é base do desenvolvimento das outras visões do sistema.



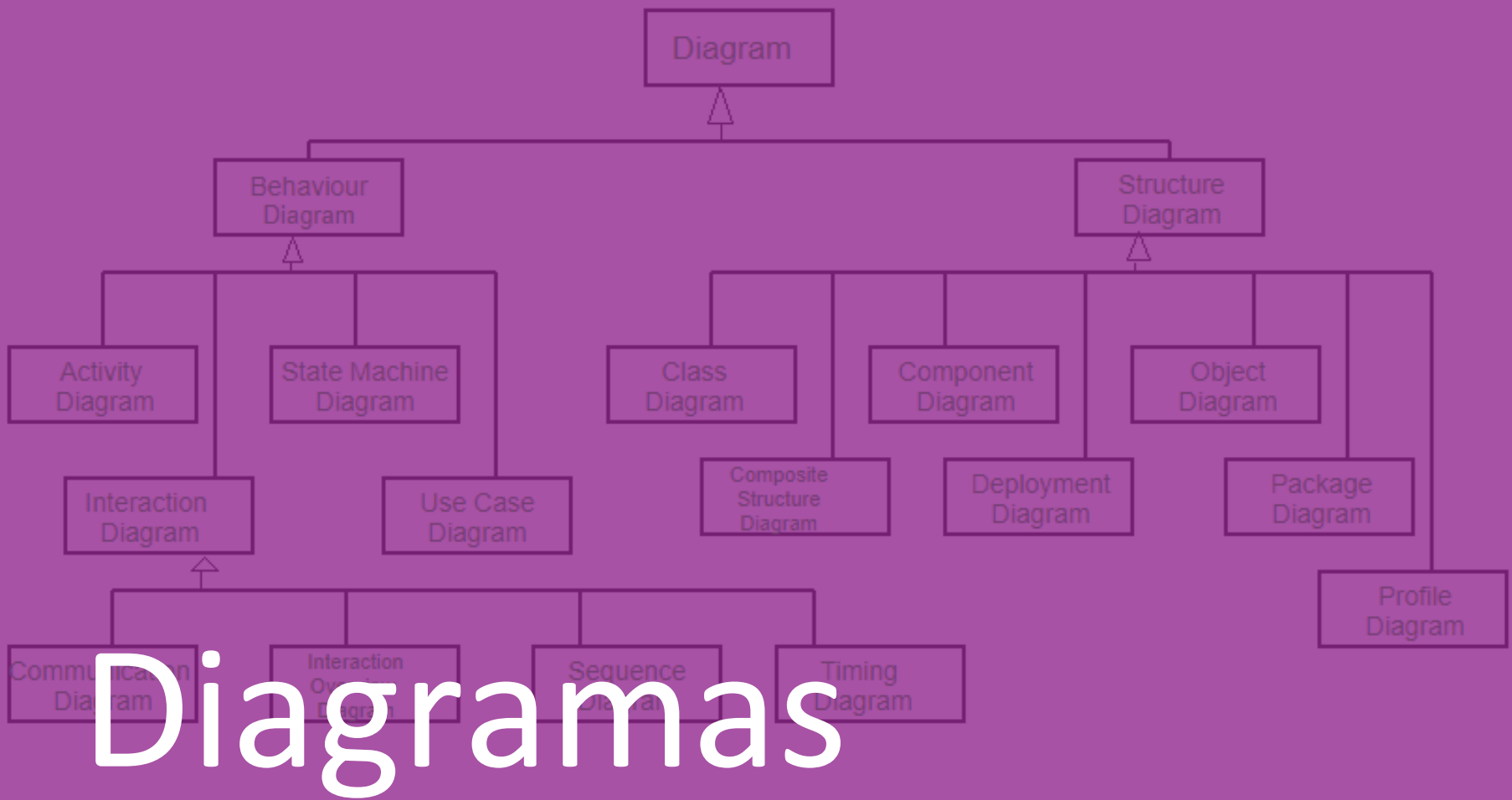
# Visões da UML

## Visão Lógica

Descreve como a funcionalidade do sistema será implementada. A visão lógica observa e estuda o sistema internamente.

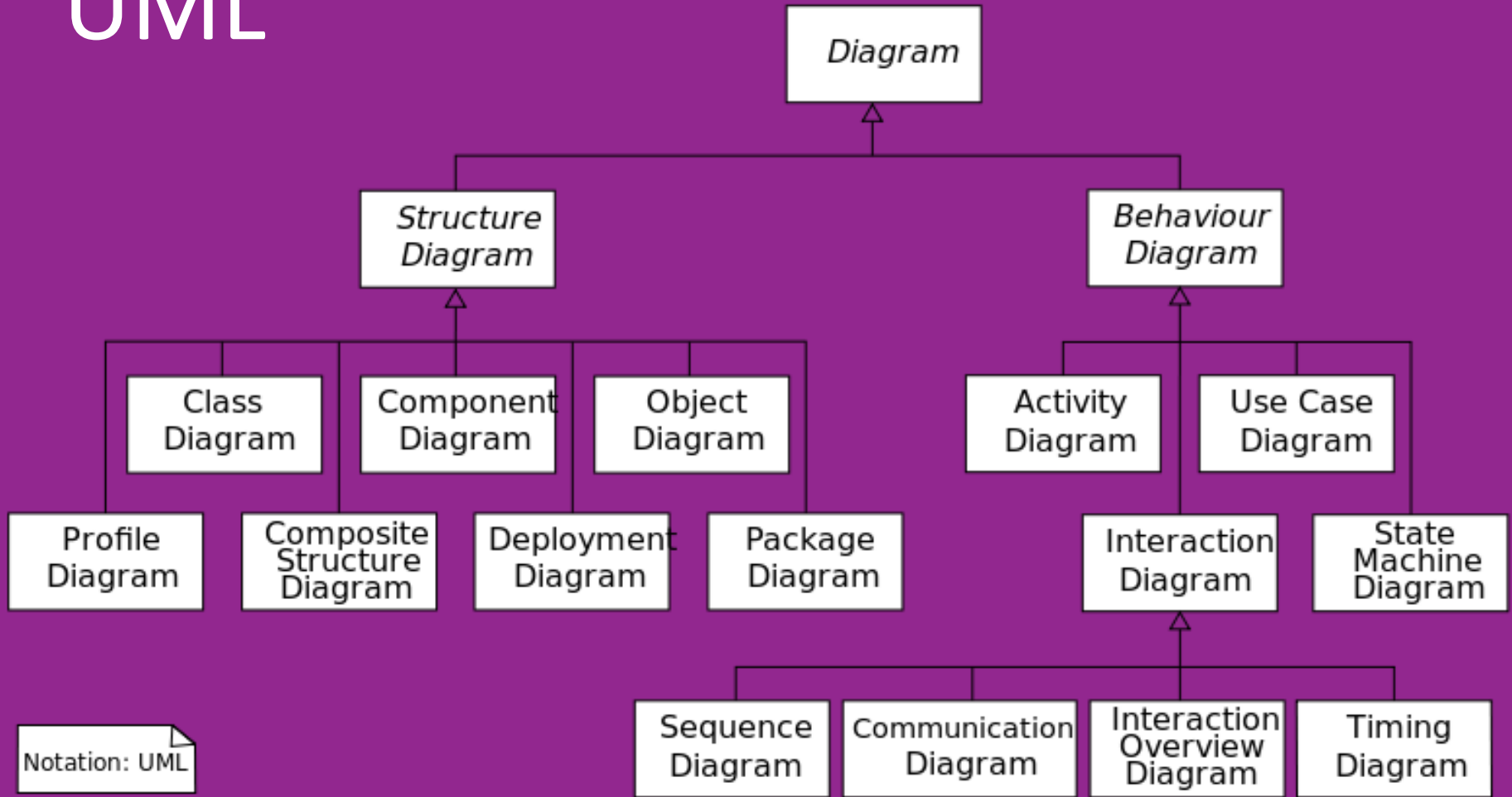
Ela descreve e especifica a estrutura estática do sistema (classes, objetos, e relacionamentos) e as colaborações dinâmicas quando os objetos enviarem mensagens uns para os outros para realizarem as funções do sistema.

Propriedades como persistência e concorrência são definidas nesta fase, bem como as interfaces e as estruturas de classes.



Os diagramas são os gráficos que descrevem o conteúdo em uma visão. UML possui nove tipos de diagramas que são usados em combinação para prover todas as visões do sistema.

# UML



# Exemplos de diagramas

Diagrama de Use Case

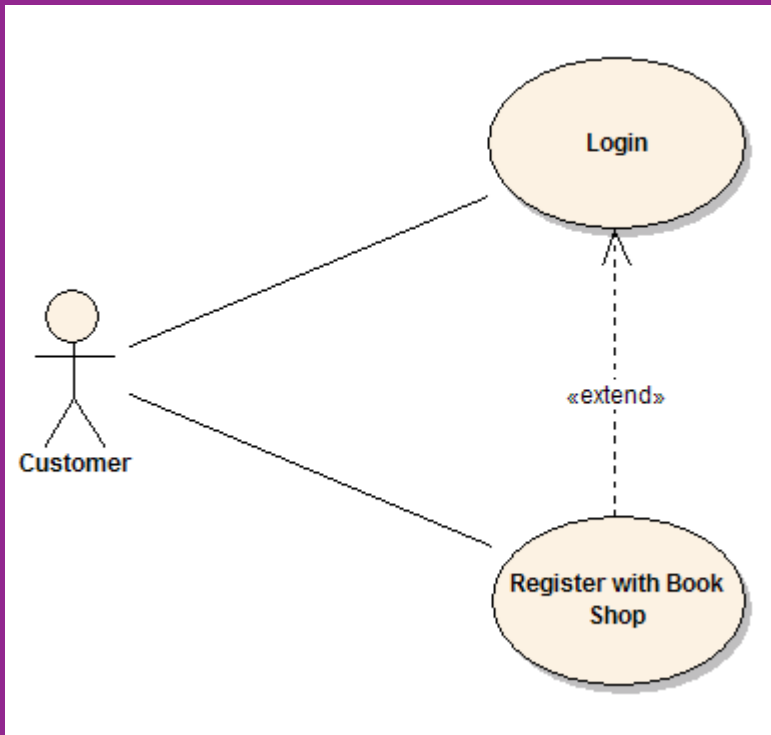
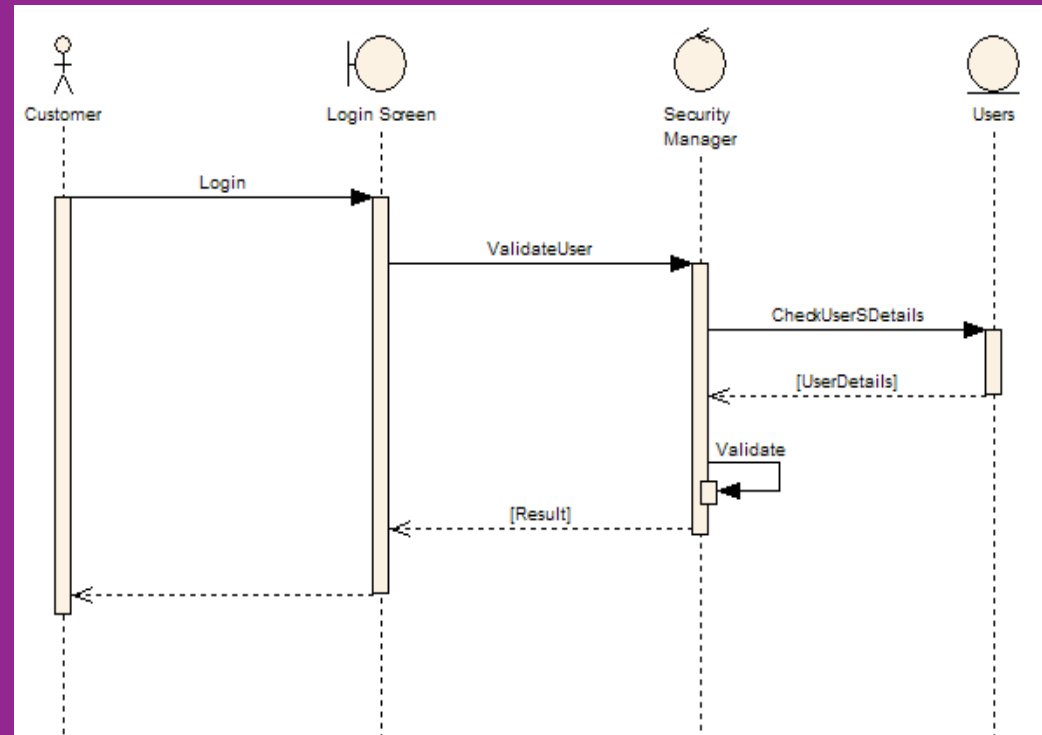


Diagrama de Sequência



# Em resumo

1. É o padrão para a modelagem Orientada a Objetos.
2. Pode ser usada para especificação, construção, visualização e documentação de sistemas de software
3. Pode ser usado durante todo o ciclo de vida de um software.
4. Pode ser usado com diferentes tecnologias de implementação.
5. Oferece uma notação gráfica baseada em vários diagramas que permitem a modelagem visual de programas orientados a objeto independente de linguagem de programação.







Casos de uso

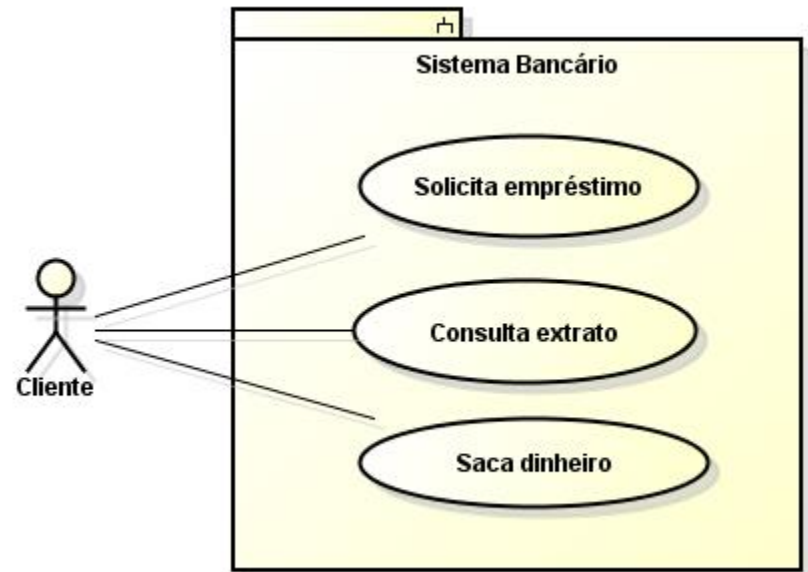
Como fazer?

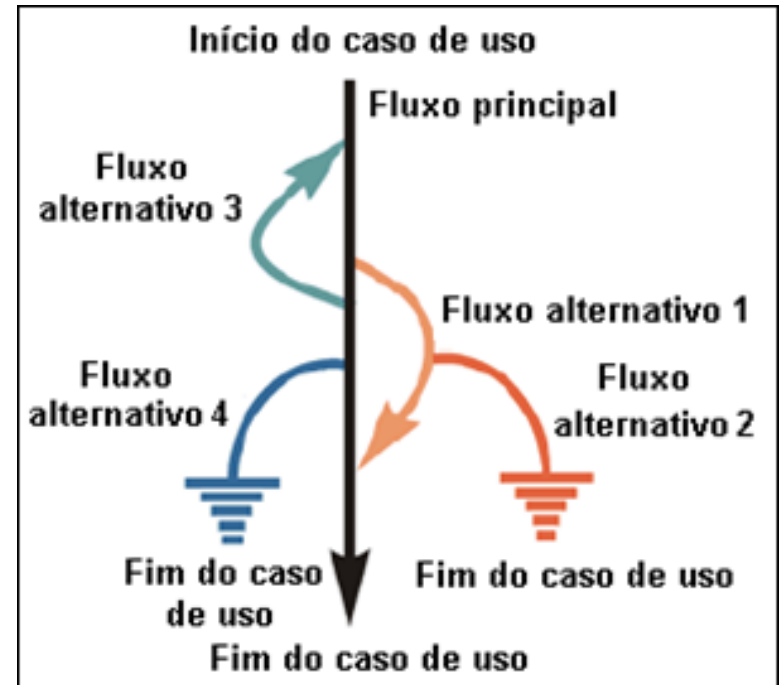
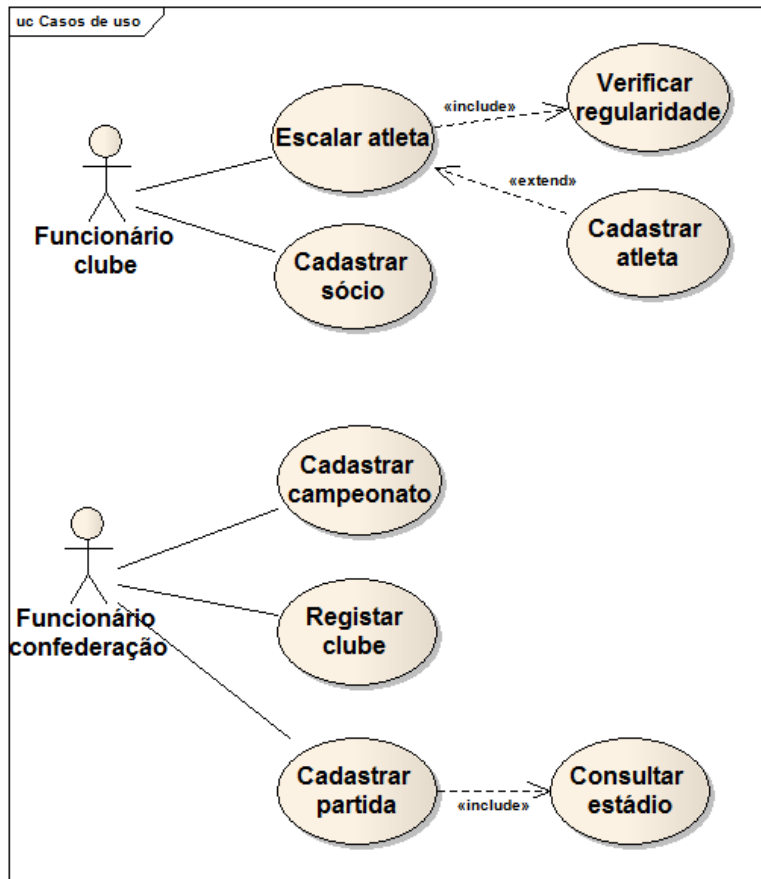
# O que é um caso de uso?

Descreve como um usuário interage com o sistema proposto para executar uma determinada funcionalidade

O termo **usuário** pode significar um usuário humano, um equipamento ou mesmo outro sistema de software

**Interação** que tem significado para o usuário final e que deve terminar (sair do sistema) com um **estado completo**, sendo concluída ou retornando ao estado inicial





# Usuário = Ator

Um ator é **qualquer coisa** que interage com o sistema (ou com um subsistema)

- Os atores não fazem parte do sistema e podem receber e/ou fornecer informações ao mesmo
- O nome que identifica o ator deve ser o **papel** que ele desempenha na utilização do sistema
- Além disso, certifique-se de que não haja risco de confundir o nome de um ator com outro



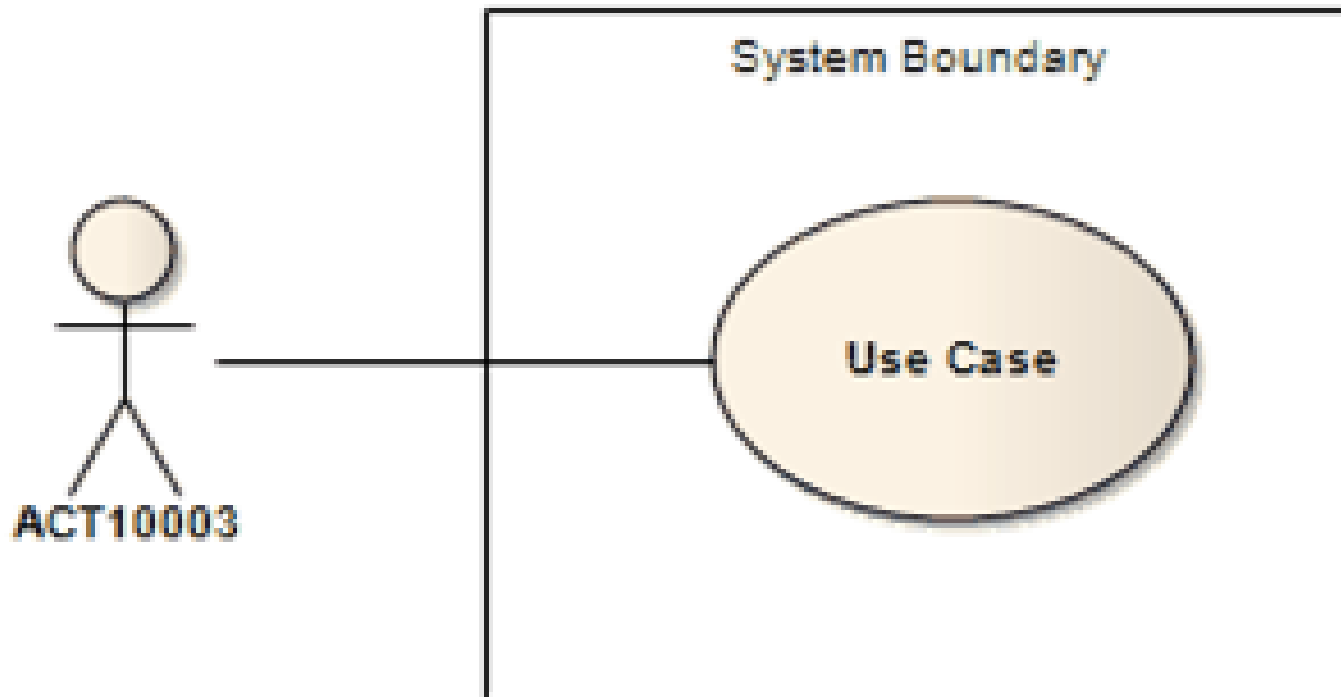
# Atores x Sistema

O Sistema ou Subsistema modelado nunca poderá ser um ator dele mesmo

Atores representam entidades externas

UML

Limite do sistema (*system boundary*) Ou Assunto (*subject*)



# Identificando Atores

Atores sempre são externos ao sistema.

Perguntas que auxiliam na hora de definir um ator:

Quem usa o sistema?

Quem instala o sistema?

Quem inicia o sistema?

Outro sistema utiliza este sistema?

Quem recebe informações do sistema?

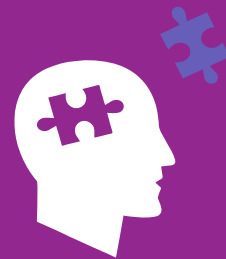
Quem fornece informações ao sistema?

# Exemplo

Ferramenta: draw io (<https://app.diagrams.net/>)

## Vamos identificar os Atores?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Esta promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no seu cadastro o preço, categoria e tamanho.



As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o cliente deve realizar o seu cadastro com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



**Observação: O sistema tem que funcionar em Firefox e Chrome.**



## Vamos identificar os Atores?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Está promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no seu cadastro o preço, categoria e tamanho.

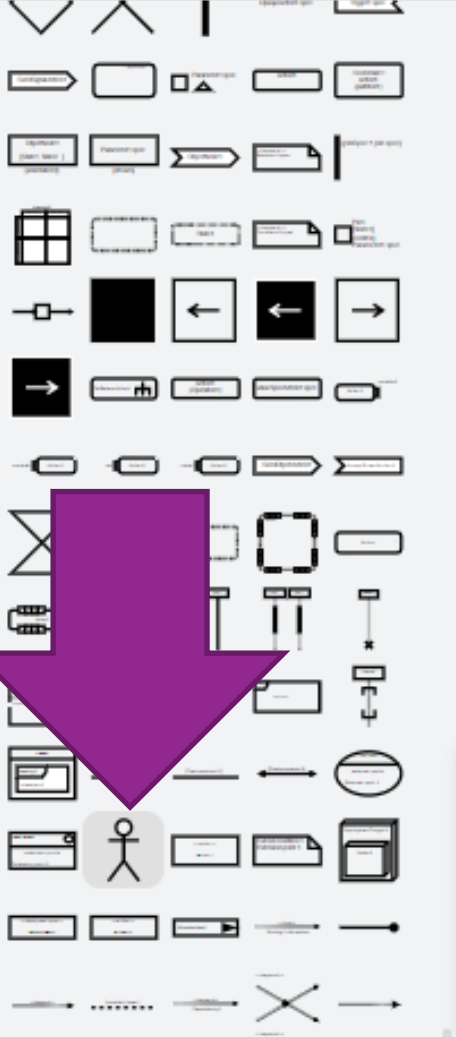
As categorias e os descontos são mostrados abaixo:



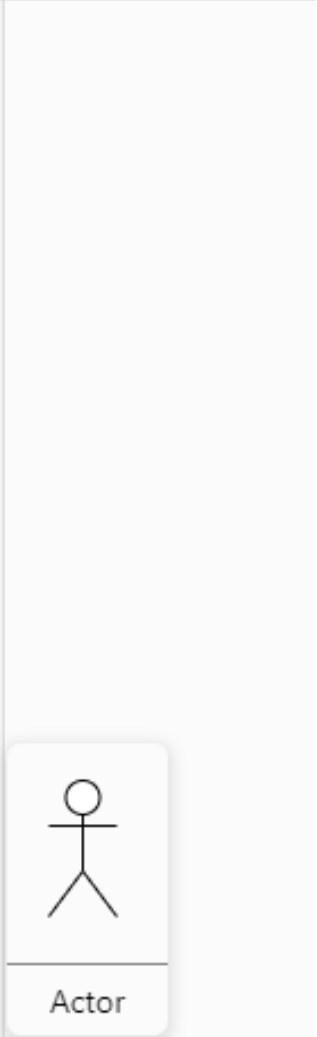
Para entrar na outlet o **cliente deve realizar o seu cadastro** com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



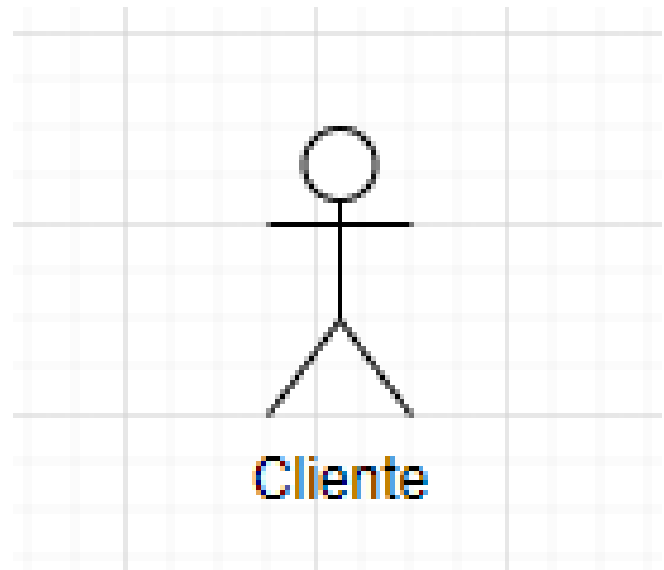
**Observação: O sistema tem que funcionar em Firefox e Chrome.**



A vertical toolbar containing various UML diagram symbols such as rectangles, ovals, arrows, and stick figures. A large purple arrow points to the stick figure icon.



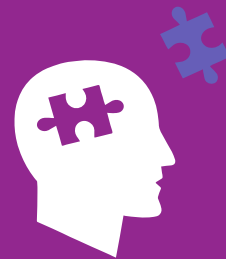
A preview of the selected actor symbol, showing a stick figure icon above the text "Actor".



# Vamos identificar os Atores?

## E os Demais ?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Esta promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no **seu cadastro** o preço, categoria e tamanho.



As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o **cliente deve realizar o seu cadastro** com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



**Observação: O sistema tem que funcionar em Firefox e Chrome.**

# Vamos identificar os Atores?

## E os Demais ?

- ❖ **Alguém tem que cadastrar (editar) os produtos e as categorias.**
- ❖ **Mas quem ?**
- ❖ **Lembra das Técnicas de Levantamento de Requisitos ?**

Vamos identificar os Atores?

E os Demais ?



- ❖ E se você fosse o dono da empresa, quem iria cadastrar os produtos ?

# Vamos identificar os Atores?

## E os Demais ?

### Está faltando ALGO ?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Está promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no **seu cadastro** o preço, categoria e tamanho.

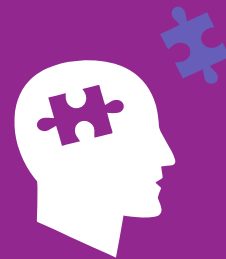
As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o **cliente deve realizar o seu cadastro** com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



**Observação: O sistema tem que funcionar em Firefox e Chrome.**



Vamos identificar os Atores?

E os Demais ?

Está faltando ALGO ?

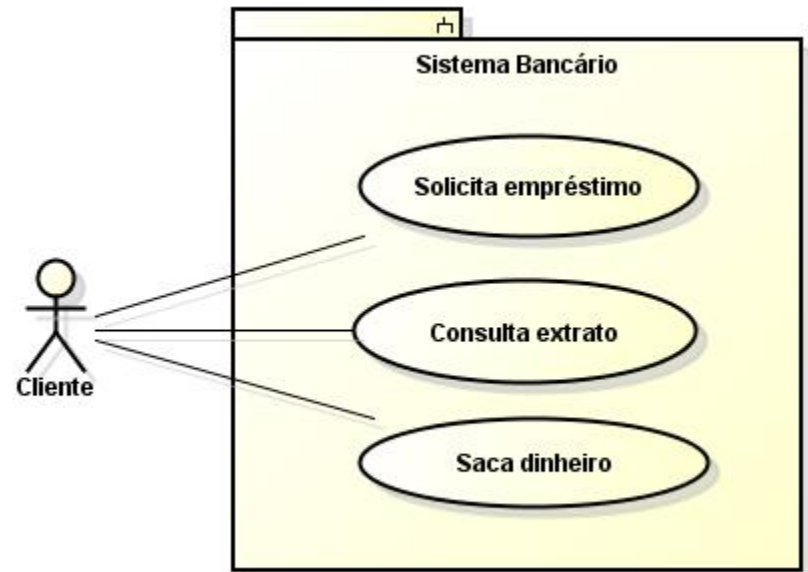


5 Minutos

❖ O que você acha que está faltando ?

# Como identificar os casos de uso?

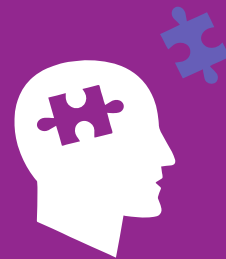
- Quais são as tarefas principais que um ator deverá executar?
- Um ator deseja consultar ou até mesmo modificar alguma informação que existe no sistema?
- Algum ator precisa informar o sistema sobre mudanças que ocorreram em outros sistemas?
- Algum ator precisa ser informado sobre eventos inesperados dentro do sistema?





# Vamos identificar os casos de uso?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Esta promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no seu cadastro o preço, categoria e tamanho.



As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o cliente deve realizar o seu cadastro com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.

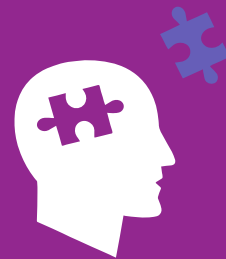


**Observação: O sistema tem que funcionar em Firefox e Chrome.**

# Vamos identificar os casos de uso?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Esta promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. **Cada produto deve ter no seu cadastro o preço, categoria e tamanho.**

As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o **cliente deve realizar o seu cadastro com nome e telefone.** **Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho.** **O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra.** Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



**Observação: O sistema tem que funcionar em Firefox e Chrome.**

# Vamos identificar os casos de uso?

## Tem mais ?

Uma loja que fazer uma grande outlet de promoções para aumentar as vendas. Esta promoção deve ser feita pela internet e com transmissão ao vivo devido ao Covid. Será dado desconto para seus produtos de acordo com a categoria de cada produto durante a transmissão. Cada produto deve ter no seu cadastro o preço, categoria e tamanho.

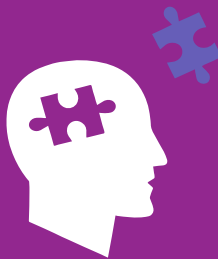
As categorias e os descontos são mostrados abaixo:



Para entrar na outlet o cliente deve realizar o seu cadastro com nome e telefone. Para cada produto apresentado deve ser exibido o preço, preço final, categoria e tamanho. O cliente terá direito ao produto quando for o primeiro a sinalizar a intenção de compra. Caso o cliente não efetue o pagamento em até 2 dias úteis terá direito a compra o próximo cliente da fila.



**Observação: O sistema tem que funcionar em Firefox e Chrome.**



Vamos identificar os Atores?

Tem mais ?



5 Minutos

❖ O que você acha que está faltando ?

# E agora ?

- ❑ Falta fazer os relacionamentos.
  - ❑ Ou seja, quem faz o que no sistema.

Algo está errado ?

O que está faltando ?



# Leitura Recomendada

## Busca Ativa

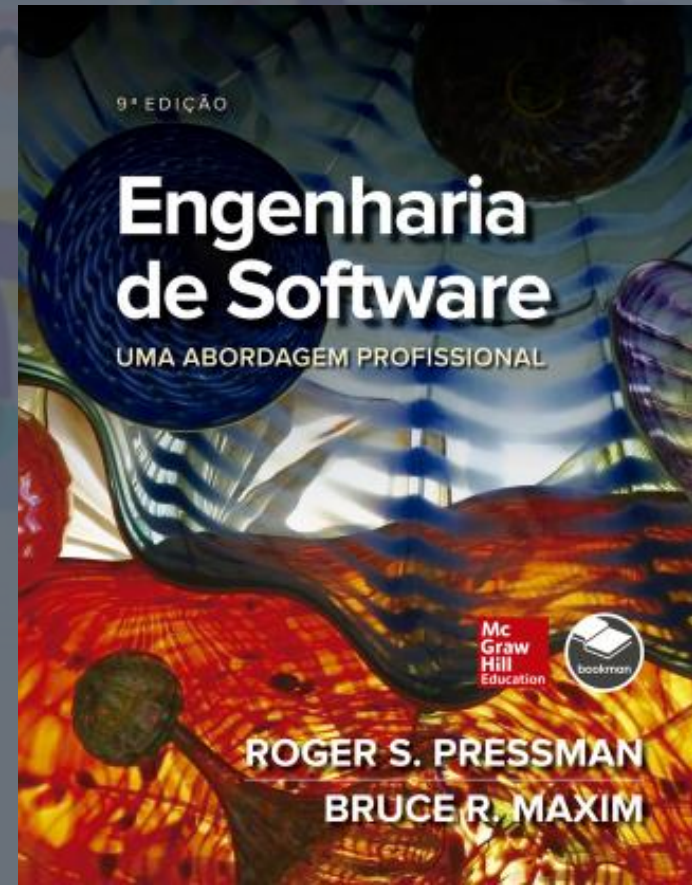
- ✓ Da página 33 até a página 107 do livro
  - ✓ REINEHR, Sheila. Engenharia de Requisitos . Grupo A, 2020. 9786556900674. Disponível em:
    - ✓ <https://integrada.minhabiblioteca.com.br/#/books/9786556900674/>. Acesso em: 04 mar. 2022.
    - ✓ No ULIFE, link da Minha biblioteca



# Leitura Recomendada

## Busca Ativa

- ✓ Da página 84 até a página 135 do livro
- ✓ Pressman, Roger, S. e Bruce R. Maxim. Engenharia de software. Disponível em: Minha Biblioteca, (9th edição). Grupo A, 2021..
- ✓ No ULIFE, link da Minha biblioteca

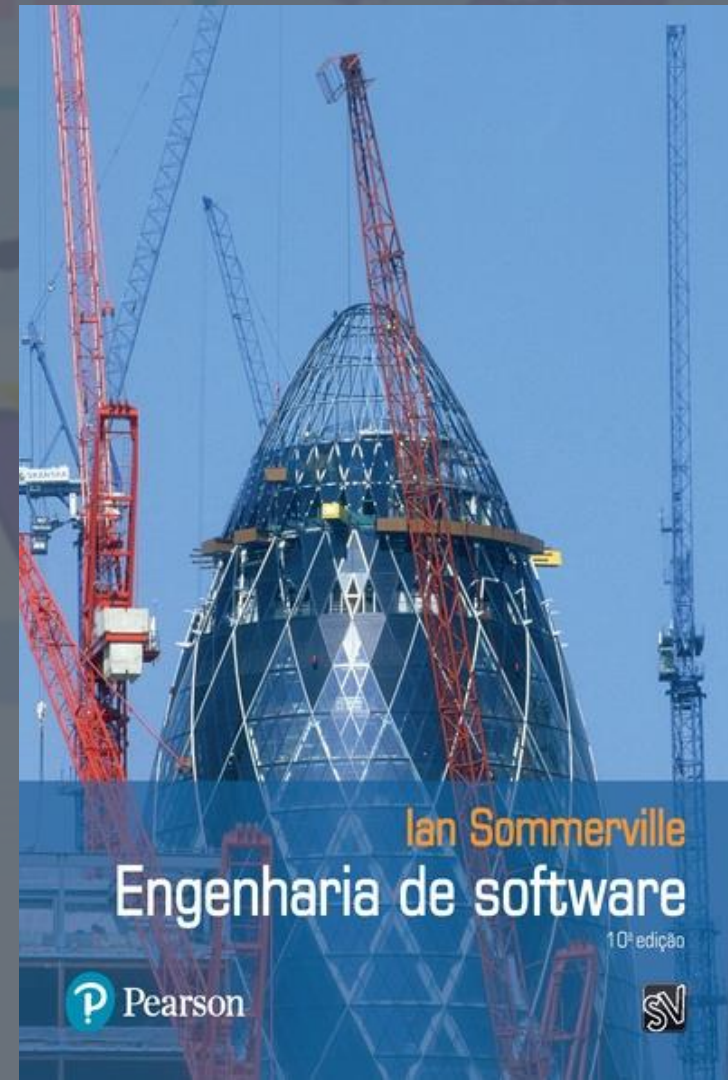




# Leitura Recomendada

## Busca Ativa

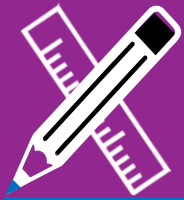
- ✓ Da página 85 até a página 128 do livro
- ✓ Engenharia de Software. Ian Sommerville. 2018.
- ✓ <https://plataforma.bvirtual.com.br/Leitor/Publicacao/168127/pdf/142>.
- ✓ No ULIFE, link da biblioteca Pearson



# DICAS

- ✓ Trilha: Fundamentos do Desenvolvimento de Sistemas
  - ✓ <https://www.ev.org.br/areas-de-interesse/programacao>
- ✓ Cursos
  - ✓ **Introdução à Programação Orientada a Objetos (POO)**
  - ✓ **Ética no Desenvolvimento de Sistemas**
  - ✓ **Projetos de Sistemas de TI**

## EXERCÍCIOS



Faça o diagrama de casos de uso para os problemas a seguir:





## Case 1: Lanchonete

- A Lanchonete MC Donald é conhecida por fazer os hamburger`s em 30 segundos áreas de “Pedido”, “Produção”. A área de “Pedido” contempla o atendimento ao cliente para registro da solicitação. O atendimento ocorre de forma presencial. O atendente registra o pedido e o número do celular. Ao finalizar a solicitação, o atendente envia para a área de “Produção”. Na “Produção”, a solicitação deve seguir uma fila. A produção prepara o Pedido. Ao concluir, o produto é entregue ao solicitante, o tempo total de produção é de 30s. O Dono da lanchonete deseja atender via delivery, e quer garantir que o pedido chegue a casa do Solicitante em 30min. O Pedido deve ser feito sem a necessidade de ligar para o atendente.





## Case 2: Locadora de carro

Uma pequena locadora de carros possui em torno de 2.000 carros, cujo a locação deve ser controlado. Cada carro possui um código. Para cada carro, é necessário saber sua categoria (Passeio, SUV, Esportivo...). Cada carro recebe um identificador próprio.

Os clientes podem desejar encontrar os carros pela sua marca favorita. Por isso, é necessário manter a informação das marcas de cada carros. Para cada marca o cliente pode desejar procurar o nome real do carro e o ano de fabricação.

A locadora possui muitos clientes cadastrados. Somente clientes cadastrados podem locar carros. Para cada cliente é necessário saber seu nome completo, seu(s) telefone(s) e seu endereço. Além disso, cada cliente recebe um número de associado.

Finalmente, desejamos saber que carros cada cliente tem emprestado. Um cliente pode ter vários carros ao mesmo tempo. Devem ser mantidos registros históricos de cada aluguel realizado pelos clientes.

O cliente pode realizar a reserva do carro de forma on-line;

A locadora só mantém carros com no máximo 5 anos de uso.

O sistema deve ser executado de forma on line, o tempo de processamento de cada tela deve ocorrer em no máximo 10 segundos. Cada tela não podem ter mais que 10 campos.





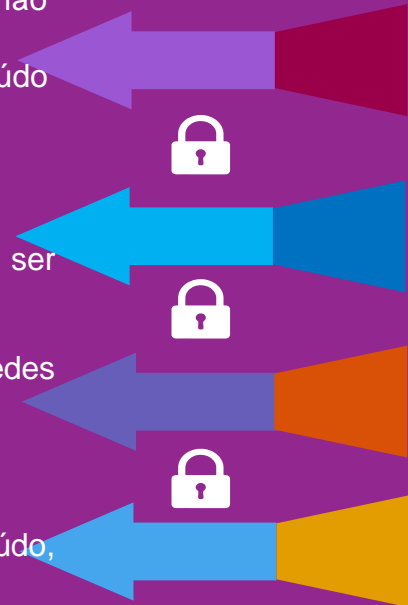
## Case 3: Portal Web

O presente projeto trata da construção de um Portal Web de gestão de informação destinado a atender ao público corporativo que atualmente tem dificuldade em encontrar informação por não estar centralizada e organizada e perde muito tempo nesta procura. O portal será disponibilizado aos usuários com livre acesso ao seu conteúdo e será alimentado com conteúdo postado pelos próprios usuários cadastrados. Portanto, o portal deve oferecer as funcionalidades abaixo.

### Escopo do projeto

Modelar um Portal Web para a gestão de informações corporativas onde devem ser consideradas minimamente as funcionalidades:

- O portal deve oferecer os recursos inclusão de comentários, compartilhamento em redes sociais, e inclusão como favoritos e cadastro de conteúdo para os colaboradores;
- O conteúdo publicado no portal é classificado como de gestão ou de tecnologia;
- O portal deve permitir que o usuário salve o conteúdo durante o cadastro.
- O portal deve oferecer uma área administrativa para publicar agendamento de conteúdo, visualização das estatísticas de acesso e cadastro de usuários.
- A publicação pode ser agendada ou ser publicada no momento do cadastro.
- Somente administradores podem publicar conteúdo mesmo que agendado.
- O portal deve disponibilizar uma ferramenta de busca para conteúdo interno.
- O portal deve funcionar em desktop e tablet.





## Case4 : Clínica médica

Uma clínica possui 2 consultórios de atendimento e 4 médicos que dividem o consultoria em dois turnos, ou seja, dois médicos atendem no turno matutino (um em cada consultório) e dois no turno vespertino. A atendente da clínica realiza o cadastro de médicos, pacientes e agenda as consultas. Os cadastros são:

1. O Cadastro de médicos deve possuir CRM, Nome, especialidade ,telefone e período de atendimento.
2. Cadastro de paciente deve possuir Nome, telefone, endereço e data de nascimento
3. O agendamento deve ser de 1 hora por atendimento e não pode ter agendamento em horários já marcados.
4. Deve permitir realizar agendamento para horários disponíveis,
5. O atendente deve visualizar os agendamentos com o horário e paciente.

Para resolver esse problema a clínica contratou os alunos para construir um sistema que possa garantir que não tenha consultas no mesmo horário, atualmente ocorrem muitos erros de agendamento.



Vamos  
trabalhar?





Dúvidas?



# Referência bibliográfica

A stack of five books is shown, with the top book open. The books are of various colors (blue, red, yellow, blue, red) and are stacked on a dark purple surface. The background is a gradient of purple.

FOWLER, Martin e SCOTT, Kendall. Uml Essencial. 2a. Edição. Bookman. Porto Alegre, 2000.

SCHNEIDER, Geri. Applying use case: a practical guide. Addison-Wesley, 1998.

OESTEREICH, Bernd. Developing Software with UML. Addison-Wesley, 1999.