

Nossa

história

TRANSFORMA

muitas

histórias

Modelagem de Sistemas

Apresentação e Diagnóstico

Objetivos do encontro de hoje:

1. Apresentação do professor
2. Fazer diagnóstico de aprendizagem (Quiz)
3. Revisar conteúdos ministrados (AGO-OUT)
4. Apresentar planejamento dos próximos encontros
5. Esclarecer dúvidas

Elisson Nadson Souza Marques

Professor Substituto de Modelagem de Sistemas



Formação Acadêmica

- Bacharelando em Engenharia de Software
- Técnico em Desenvolvimento de Sistemas (SENAI)
- Técnico em Informática (IETAAM)

Experiência Profissional

- 14 anos de atuação na área de TI
- Técnico em Informática na UEFS (Atual)
- Supervisor de Redes (9 anos de experiência)
- Desenvolvedor Frontend
- Ex-Analista MIS em Tel Contact Center



Onde estamos na jornada?

É fundamental
compreendermos o contexto
da nossa posição atual no
percurso formativo.

Vejamos o que já foi
consolidado e o que ainda
está por vir nos próximos
encontros:

1

Agosto a Outubro: 12 encontros realizados

- Regras de Negócio
- Requisitos Funcionais e Não-Funcionais
- Casos de Uso
- MER (Modelo Entidade-Relacionamento)
- Metodologias Ágeis

2

Novembro a Dezembro: 7 encontros planejados

- Continuação da modelagem UML
- Diagramas de Classes e Sequência
- Conectividade e APIs
- UX/UI, PDCA e Canvas
- Avaliação Final

Quiz de nivelamento: Atividade diagnóstica

O **quiz** tem como objetivo principal fazer um diagnóstico do conhecimento atual da turma em relação aos conteúdos abordados nos encontros anteriores. Responda com calma e atenção. Ao final, teremos um momento para revisar as respostas coletivamente e discutir os pontos mais relevantes.

O intuito é identificar os temas que precisam de maior reforço e alinhar o planejamento dos próximos encontros às necessidades da turma.

Estrutura do Quiz

43 questões de múltipla escolha sobre os conteúdos de agosto a outubro

Tempo Disponível

30 minutos para conclusão - tempo suficiente para reflexão cuidadosa

Objetivo Principal

Identificar **pontos de reforço** e ajustar o planejamento das próximas aulas

Acesse o quiz através do QR Code ou do link abaixo:



<https://forms.gle/K8XACWrCwDMxKm3y8>

ATENÇÃO: Não vale nota.

Esta é uma atividade diagnóstica para balizar os próximos encontros.

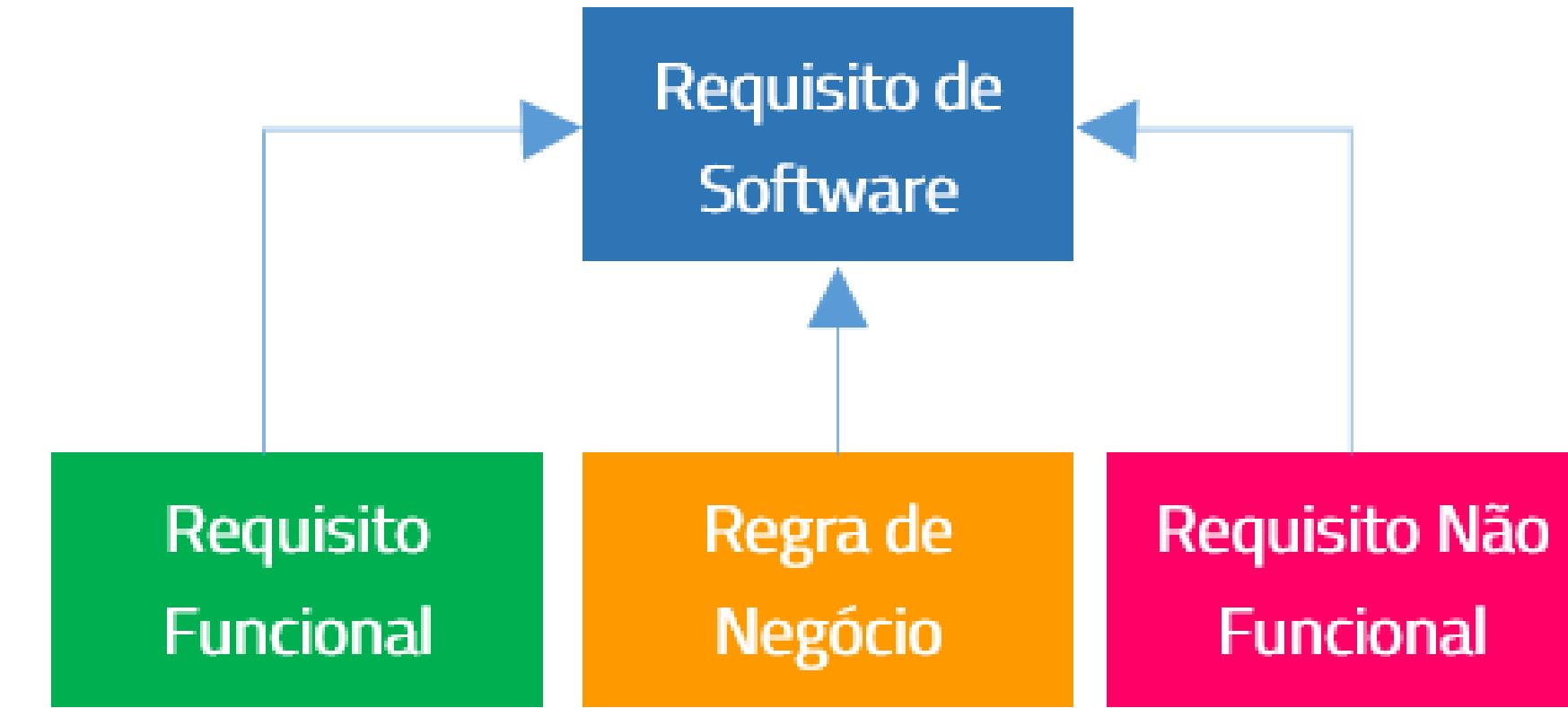
Revisão: O que é Modelagem?

A **Modelagem de Sistemas** é um processo de criação de representações abstratas e simplificadas de um sistema do mundo real. Essas representações, que podem ser diagramas, gráficos ou descrições textuais, ajudam a visualizar, especificar, construir e documentar o comportamento e a estrutura de um sistema.

É uma ferramenta essencial na engenharia de software para gerenciar a complexidade, facilitar a comunicação entre as partes interessadas e garantir que o sistema desenvolvido atenda aos requisitos.

"Um modelo não é a única solução possível, mas um conjunto de possibilidades"

Revisão: Requisitos de Software



- RN = Condições
- RF = Ações (o que o sistema faz)
- RNF = Qualidades (como o sistema se comporta)

são as especificações, restrições e necessidades que um sistema deve atender para ser bem-sucedido

Revisão: Regras de Negócio

Definição: Regras de Negócio são diretrizes que definem CONDIÇÕES ou CRITÉRIOS que precisam ser atendidos para que uma operação seja realizada.

Características:

- Definem RESTRIÇÕES
- Estabelecem LIMITES
- Determinam CONDIÇÕES para ações
- Não descrevem funcionalidades, mas REGEM as funcionalidades

EXEMPLO:

RN001 | Carga Horária Mínima / A matrícula só será aceita se a carga horária total for ≥ 20 horas.

Estrutura: ID | Nome | Descrição

Revisão: Regras de Negócio

Funcionalidades específicas que o sistema precisa ter

CARACTERÍSTICAS

- Sempre começa com verbo (ação)
- Descreve comportamento observável
- Pode ser testado e validado

TEMPLATE DE ESCRITA

“O sistema deve [VERBO] [OBJETO] [COMPLEMENTO]”

“Cada requisito funcional resolve um problema REAL do projeto”

Requisitos Funcionais:

Requisitos Funcionais:

Exemplo prático

O Sistema de padaria deve:



O requisito funcional (RF), tem relação com as funcionalidades do sistema, o que ele deve fazer.

- **RF01 - CADASTRO DE PRODUTOS:** Permitir cadastrar produtos com nome, código, preço, categoria e foto
- **RF02 - CONTROLE DE ESTOQUE:** Alertar quando estoque < 10 unidades e enviar notificação ao gerente
- **RF03 - GESTÃO DE PEDIDOS:** Registrar pedidos presenciais, por telefone e online com data de entrega
- **RF04 - PONTO DE VENDA (PDV):** Calcular total, aplicar descontos, registrar pagamento e imprimir recibo
- **RF05 - GESTÃO DE FUNCIONÁRIOS:** Cadastrar funcionários, definir horários e calcular salários
- **RF06 - RELATÓRIOS FINANCEIROS:** Gerar relatórios de vendas diárias, mensais e anuais com gráficos

Requisitos Não-Funcionais:

Já o requisito não funcional (RNF), é sobre como o sistema será, quais serão as suas restrições.

"Qualidades e restrições do sistema — não são funcionalidades, mas CARACTERÍSTICAS de como as funcionalidades devem operar"

DESEMPENHO (Performance)

- Tempo de resposta < 2 segundos
- Suportar 1.000 usuários simultâneos
- Processar 10.000 transações/hora

SEGURANÇA (Security)

- Criptografia de dados sensíveis
- Autenticação de dois fatores
- Backup automático diário

USABILIDADE (Usability)

- Criptografia de dados sensíveis
- Autenticação de dois fatores
- Backup automático diário

Desempenho (Performance)

- Código documentado
- Arquitetura modular
- Fácil adicionar novos módulos

DESEMPENHO (Performance)

- Funcionar em Windows, Linux, Mac
- Versão web + mobile
- Compatível com Chrome, Firefox, Safari

Requisitos Não-Funcionais:

TEMPLATE DE ESCRITA

"O sistema deve [QUALIDADE] [VALOR/MÉTRICA]"

Onde:

QUALIDADE= adjetivo (rápido, seguro, confiável...)

VALOR/MÉTRICA = número mensurável ou condição

Requisitos Não-Funcionais:

Exemplo prático

O Sistema de padaria:



- **RNF01** - O sistema deve funcionar 24/7 com disponibilidade de 99,9%
- **RNF02** - O sistema deve ser acessível via smartphone
- **RNF03** - Backup automático na nuvem a cada 6 horas
- **RNF04** - Interface em português com fonte mínima de 14pt

"O sistema deve [PROTEGER/GARANTIR]
[O QUÊ] usando/através de [MECANISMO]"

Requisito Funcional (RF)

O QUE o sistema faz
Funcionalidades

"Cadastrar cliente"
"Emitir relatório"
"Fazer login"

VERBO DE AÇÃO

Requisito Não-Funcional (RNF)

COMO o sistema faz
Qualidades

"Cadastrar em menos de 3 seg"
"Relatório em PDF/Excel"
"Login seguro com criptografia"

ADJETIVO/QUALIDADE

Exemplos Sistema de Biblioteca:

RF: "O sistema deve permitir reservar livros"

RNF: "A reserva deve ser processada em até 1 segundo"

RF: "O sistema deve enviar e-mail de aviso de devolução"

RNF: "O e-mail deve ser enviado em português ou inglês"

RF: "O sistema deve calcular multas por atraso"

RNF: "O cálculo deve seguir a Lei Municipal nº 1234/2024"

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve permitir que usuários façam upload de fotos" RF RNF
- "O sistema deve carregar a página inicial em menos de 2 segundos" RF RNF
- "O sistema deve enviar notificações por e-mail e SMS" RF RNF
- "O sistema deve funcionar em português, inglês e espanhol" RF RNF
- "O sistema deve gerar relatórios de vendas mensais" RF RNF
- "O sistema deve ter interface acessível para daltônicos" RF RNF

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **permitir que usuários façam upload de fotos**" [**X**] RF [] RNF

Por quê?

- Define uma **funcionalidade** (ação que o sistema realiza)
- Responde "O QUE o sistema faz?" → Permite upload
- É uma **ação observável e testável**

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **carregar a página inicial em menos de 2 segundos**" RF RNF

Por quê?

- Define uma qualidade (desempenho/tempo de resposta)
- Responde "COMO o sistema deve se comportar?" → Rápido (< 2s)
- Estabelece um **atributo de qualidade** mensurável
- Categoria: **DESEMPENHO**

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **enviar notificações por e-mail e SMS**" [X] RF [] RNF

****Por quê?****

- Define uma **funcionalidade** (envio de notificações)
- Responde "O QUE o sistema faz?" → Envia notificações
- É uma **ação concreta** que o sistema executa
- Os canais (e-mail/SMS) são parte da funcionalidade

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **funcionar em português, inglês e espanhol**" RF RNF

****Por quê?****

- Define uma ****qualidade**** (internationalização/portabilidade)
- Responde "COMO o sistema deve se comportar?" → Multilíngue
- Estabelece uma ****restrição de suporte****
- Categoria: ****PORTABILIDADE / USABILIDADE****

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **gerar relatórios de vendas mensais**" **[X] RF [] RNF**

****Por quê?****

- Define uma **funcionalidade** (geração de relatórios)
- Responde "O QUE o sistema faz?" → Gera relatórios
- É uma **ação observável** com resultado concreto

Desafio 1 minuto: Funcional ou Não-Funcional?

INSTRUÇÕES

Classifique cada requisito abaixo como RF ou RNF

LISTA DE REQUISITOS

- "O sistema deve **ter interface acessível para daltônicos**" RF RNF

****Por quê?****

- Define uma ****qualidade**** (acessibilidade)
- Responde "COMO o sistema deve se comportar?" → Acessível
- Estabelece um ****atributo de qualidade**** da interface
- Categoria: ****USABILIDADE / ACESSIBILIDADE****

DICA PRÁTICA PARA DIFERENCIAR

PERGUNTA-CHAVE

É Requisito Funcional (RF)?

Responda: "O sistema FAZ isso?" → SIM = RF

É Requisito Não-Funcional (RNF)?

Responda: "Isso define COMO/QUAL QUALIDADE o sistema tem?" → SIM = RNF

PALAVRAS-CHAVE

Indicam RF: Permitir, cadastrar, enviar, gerar, calcular, exibir, processar

- Ações concretas

Indicam RNF: Qualidades, restrições, atributos

- Em menos de X segundos, disponível X%, em X idiomas, acessível para

Revisão: Casos de Uso

O Que é Caso de Uso

Conjunto de passos que **descreve um cenário principal** e diferentes cenários **alternativos**

- Começa quando o usuário tem todas as condições já atendidas para iniciar o caso de uso
- Tudo o que deveria ser feito para conclusão da tarefa, foi feito sem necessidade de passo subsequente

Descreve o que o software deve fazer e não como será implementado

O método de casos de uso é composto por:

- Diagrama de casos de uso;
- Descrição dos atores;
- Especificação dos casos de uso;

Revisão: Casos de Uso

Diagrama de casos de uso

Compõe o produto em análise

É uma representação gráfica para os casos de uso

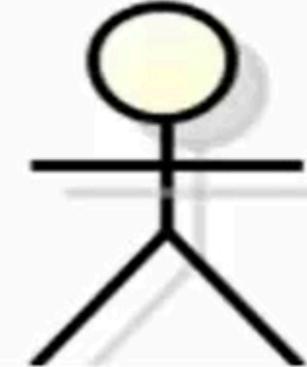
os papéis que os usuários (atores) desempenham nesses casos de uso
a inter-relação entre esses **elementos**

Deve **demonstrar as funcionalidades** que atenderão as necessidades do usuário apoando na verificação e validação

Também deve possibilitar a reformulação parcial ou total do diagrama com o mínimo de esforço

Revisão: Casos de Uso

Os elementos básicos de um Diagrama de Casos de Uso são

Elemento	Significado	Simbologia
Autor	Representa uma pessoa (ou um grupo de pessoas) que desempenha um papel ao interagir com o software . Não se limita a isso e pode ser qualquer "coisa" que interage com o software com a finalidade de cumprir um trabalho significativo, como outros produtos de software ou mesmo equipamentos.	
Caso de uso	<p>Representa uma funcionalidade que atende a um ou mais requisitos do cliente. Como nome, sugere-se usar um verbo no infinitivo com um complemento.</p> <p>Os casos de uso podem opcionalmente estar envolvidos por um retângulo que representa os limites do sistema.</p>	
Relacionamento (ou associação)	Um ator interage com um caso de uso e isso é representado por um relacionamento . Os casos de uso também podem se relacionar entre si .	

Revisão: MER (Modelo Entidade-Relacionamento)

MER (Modelo de Entidade e Relacionamento) é uma ferramenta gráfica usada para projetar e auxiliar na implementação de bancos de dados cuja finalidade é identificar as entidades e os seus relacionamentos.

ELEMENTOS:

- Entidades (tabelas)
- Atributos (campos)
- Relacionamentos (ligações entre entidades)

Revisão: MER (Modelo Entidade-Relacionamento)

Entidades: São abstrações do mundo real que contem um conjunto de informações inter-relacionadas e coerentes, estas informações são chamadas de **atributos**.

Nomeação:

Nome que a identifica, geralmente formada por um substantivo no singular.

Representação Gráfica:

Retângulo com o nome ao centro

Funcionario

Revisão: MER (Modelo Entidade-Relacionamento)

Atributo: Cada informação que compõe uma **Entidade**, possui um NOME, um TIPO e um TAMANHO (número de caracteres). De modo genérico o “tipo” pode nominado como “texto”, “número”, “data e hora”, entre outros.

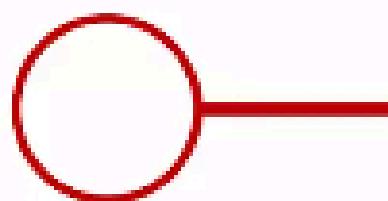
Nomeação:

- Começar com uma letra.
- NÃO conter espaço ou acentuação.
- Caracteres especiais apenas “_”.

Representação Gráfica:

Círculo com o nome ao lado
ou uma elipse com o seu nome

dataDemissao



Revisão: MER (Modelo Entidade-Relacionamento)

Tipos de Atributo:

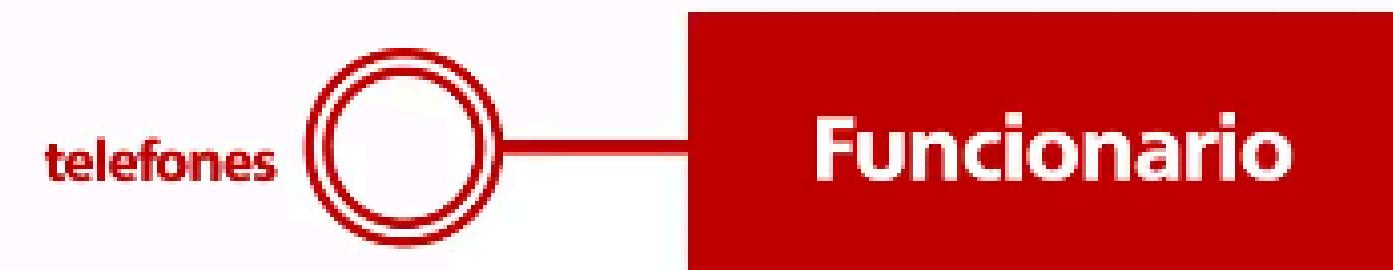
- Simples
- Composto
- Multivvalorado
- Opcional
- Derivado
- Identificador

Revisão: MER (Modelo Entidade-Relacionamento)

Atributo (multivvalorado): Pode possuir diversos valores para uma única entidade.

Representação Gráfica:

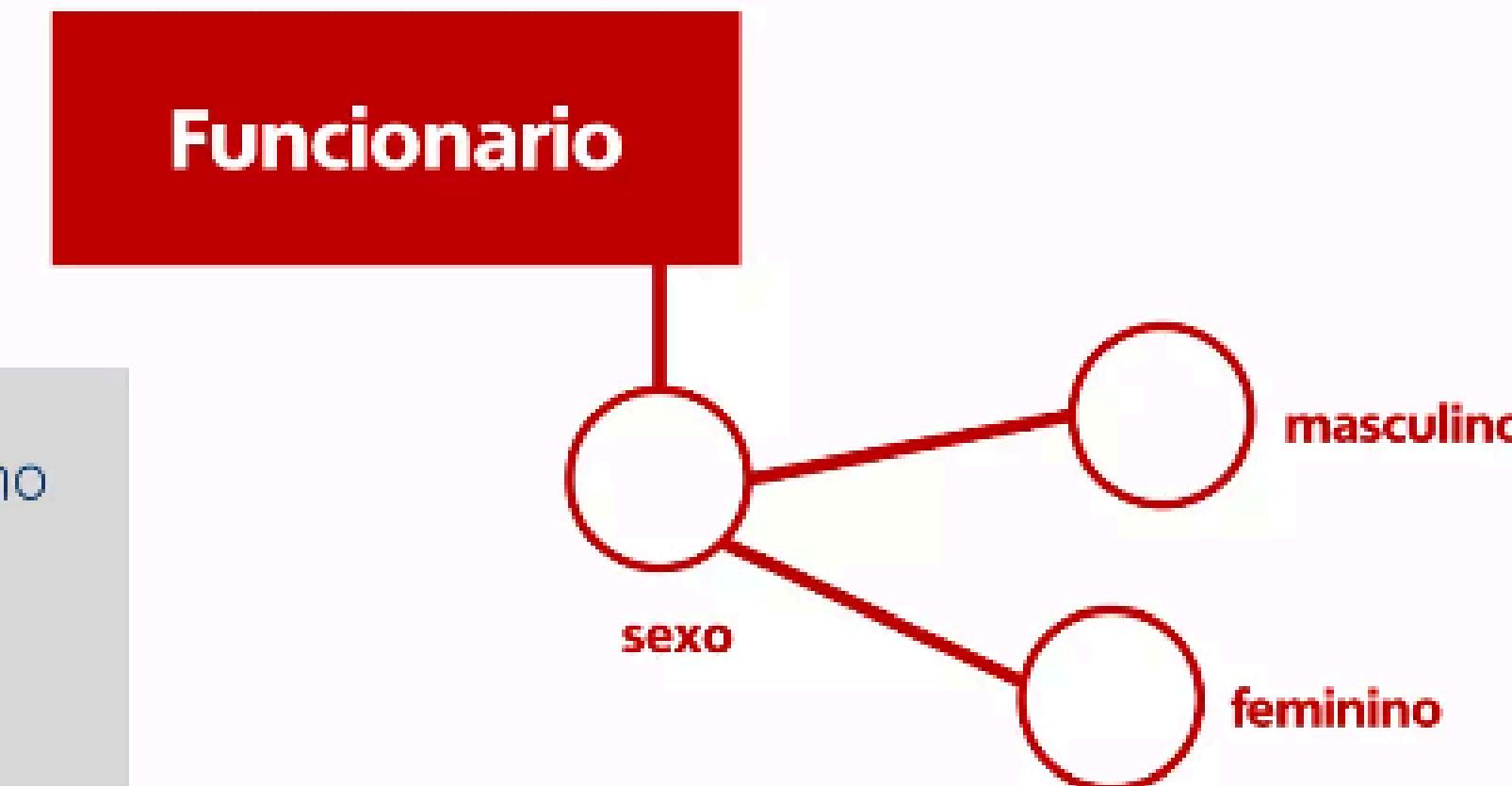
Um Circulo dentro do Outro



Revisão: MER (Modelo Entidade-Relacionamento)

- **Atributo (opcional):** Atributo que define opções de escolha.

Exemplo:



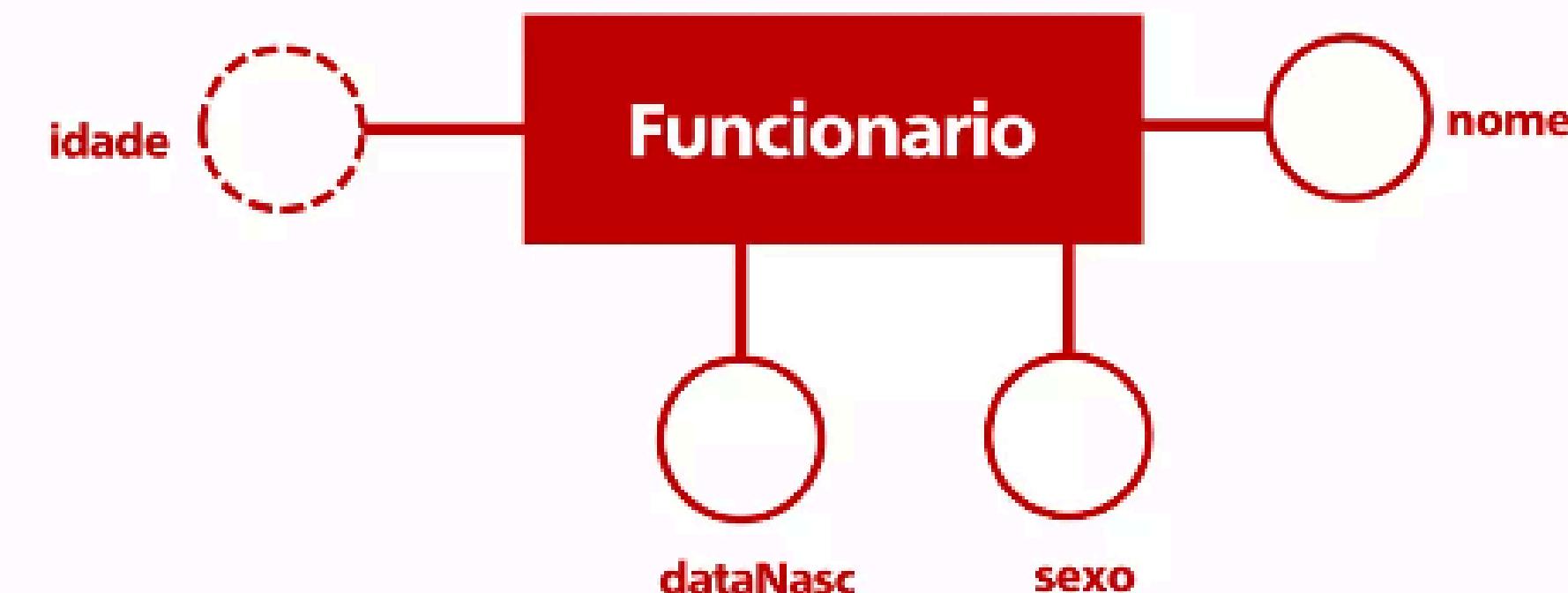
Um **Atributo Opcional**, ao mesmo tempo, pode ser um **Atributo Composto** ou **Multivalorado**.

Revisão: MER (Modelo Entidade-Relacionamento)

- **Atributo (derivado):** Atributo cujo valor pode ser derivado a partir de outro atributo

Representação Gráfica:

Um Circulo Pontilhado

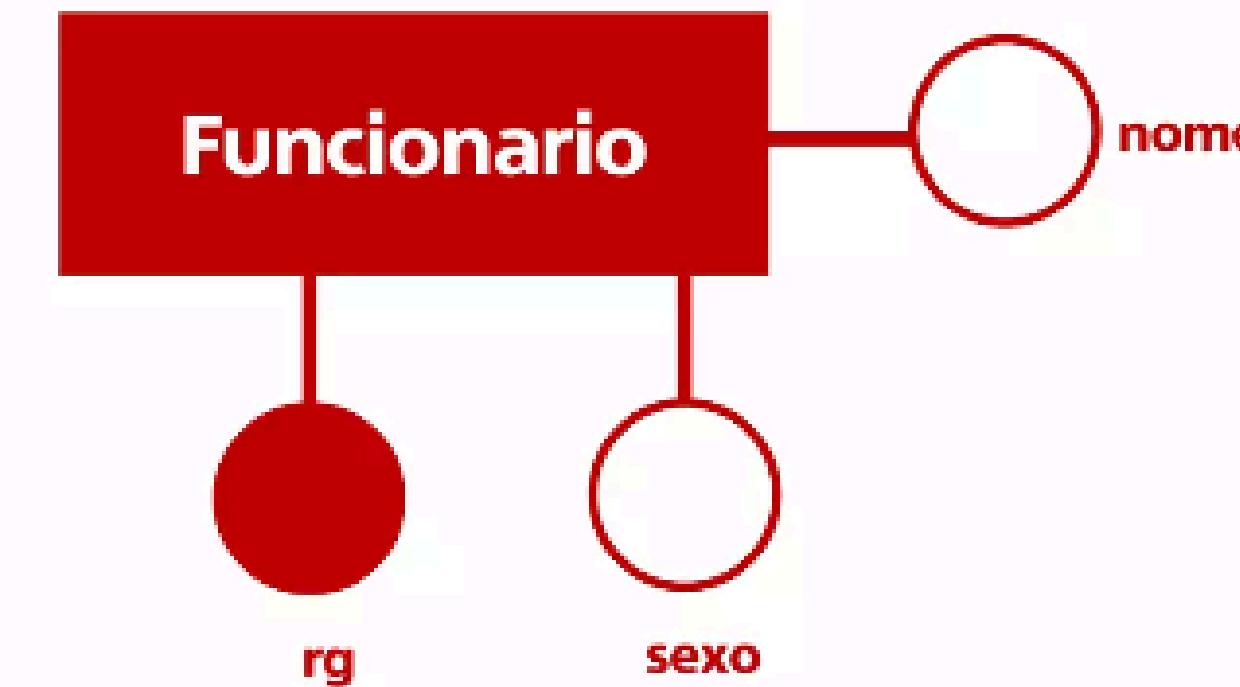


Revisão: MER (Modelo Entidade-Relacionamento)

- **Atributo (identificador):** Permite identificar univocamente cada entidade em um conjunto entidades.

Representação Gráfica:

Um Circulo Preenchido ou Com o seu nome Sublinhado



Revisão: MER (Modelo Entidade-Relacionamento)

Relacionamento: Elemento responsável por definir as características das ligações entre as entidades.

Nomeação:

Nome expresso por um verbo ou uma locução verbal.

Representação Gráfica:

Losango com o nome ao centro



Revisão: MER (Modelo Entidade-Relacionamento)

CARDINALIDADE

1 para **N**

N para **1**

N para **N**

1 para **1**



Revisão: MER (Modelo Entidade-Relacionamento)

CARDINALIDADE

1 para **N**

N para **1**

N para **N**

1 para **1**



Revisão: MER (Modelo Entidade-Relacionamento)

CARDINALIDADE

1 para **N**

N para **1**

N para **N**

1 para **1**



Revisão: MER (Modelo Entidade-Relacionamento)

CARDINALIDADE

1 para **N**

N para **1**

N para **N**

1 para **1**

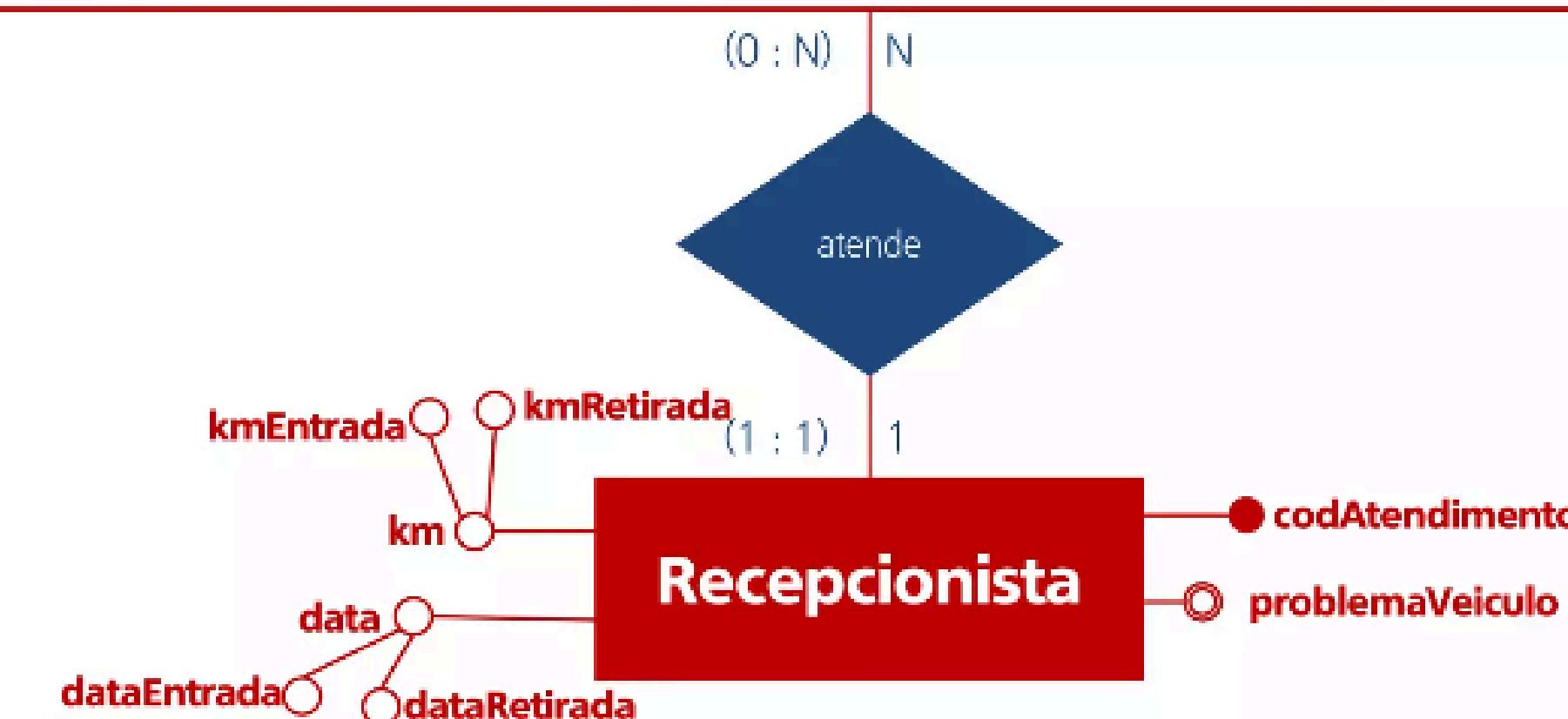
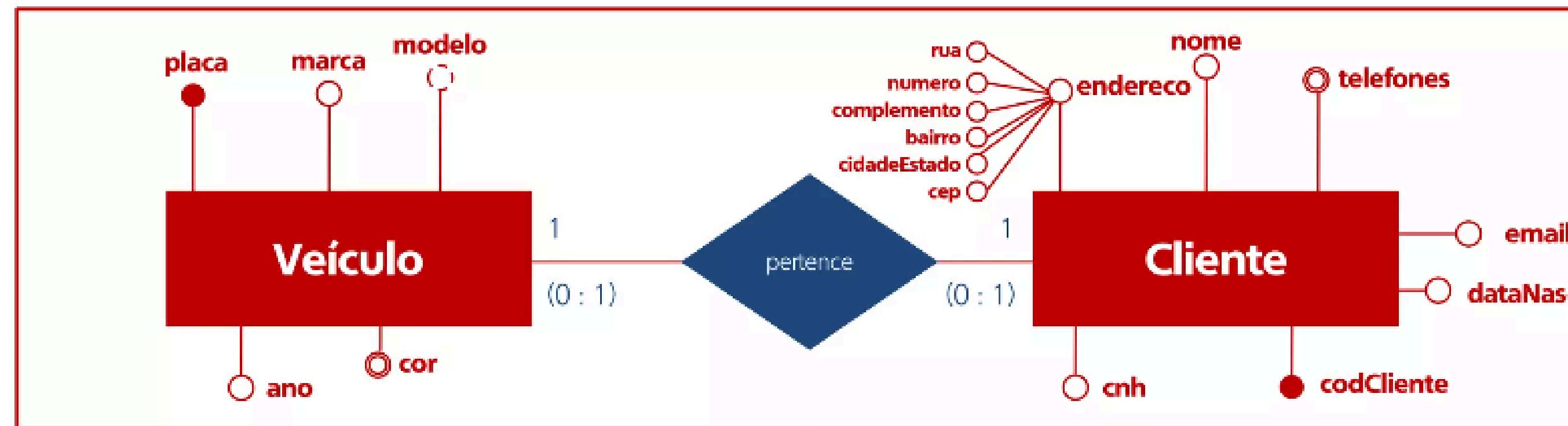


Revisão: MER (Modelo Entidade-Relacionamento)

EXERCÍCIO

Criar um Diagrama de Entidade e Relacionamento para um Banco de Dados que armazenará informações de um Sistema para uma Mecânica. O Diagrama deverá ter no mínimo 3 entidades.

Revisão: MER (Modelo Entidade-Relacionamento)



Metodologias Ágeis

O que são Metodologias Ágeis?

- Metodologias Ágeis são formas de gerenciar projetos que focam em entregas rápidas, flexibilidade e colaboração da equipe.
- Indivíduos e interações mais que processos.
- Software em funcionamento mais que documentação.
- Colaboração com o cliente mais que negociação.
- Responder a mudanças mais que seguir um plano.

Metodologias Ágeis

Frameworks Populares: Scrum vs. Kanban

Scrum

Um framework iterativo focado em gerenciar projetos complexos. O trabalho é dividido em ciclos curtos e fixos (Sprints).

- Ritmo: Iterativo (Sprints de 1-4 semanas).
- Papéis: Product Owner, Scrum Master, Time.
- Foco: Entrega de incremento de valor ao final de cada Sprint.

Kanban

Um método visual focado em gerenciar e otimizar o fluxo de trabalho. Foca em visualizar o trabalho e limitar o "Trabalho em Progresso" (WIP).

- Ritmo: Fluxo Contínuo (contínuo).
- Papéis: Flexível (geralmente sem papéis prescritos).
- Foco: Melhoria contínua do fluxo e redução de gargalos.

Próximos Passos

13/11 - Hoje

Apresentação, diagnóstico e revisão

04/12

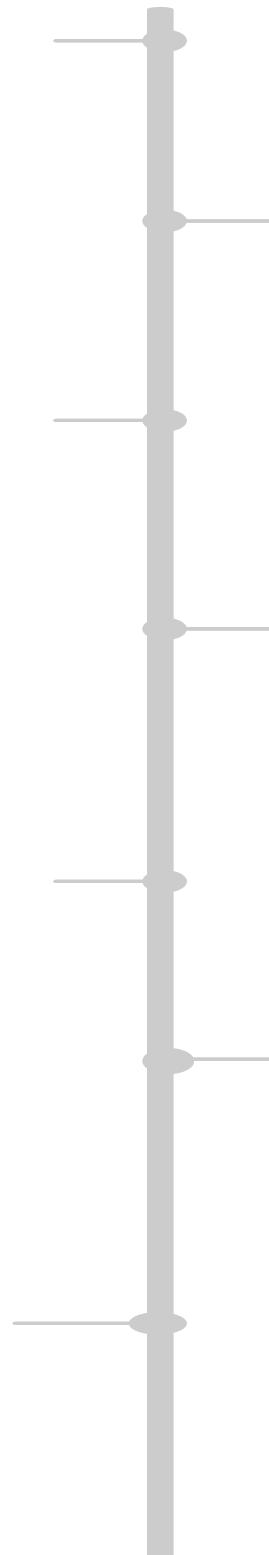
UML - Diagrama de Classes (Parte 2) +
Sequência

12/12

Conectividade, APIs, PDCA e Canvas

19/12

Avaliação Final



14/11

UML - Diagrama de Classes (Parte 1)

05/12

UML - Diagramas de Sequência e
Atividades

18/12

UX/UI + Revisão Geral

Critérios de Avaliação

COMPOSIÇÃO DA NOTA:

- Atividades práticas em sala: 30%
- Trabalhos/exercícios: 20%
- Avaliação Final (19/DEZ): 50%

Recuperação disponível para quem obtiver nota entre 5,0 e 6,9

Próximos Passos

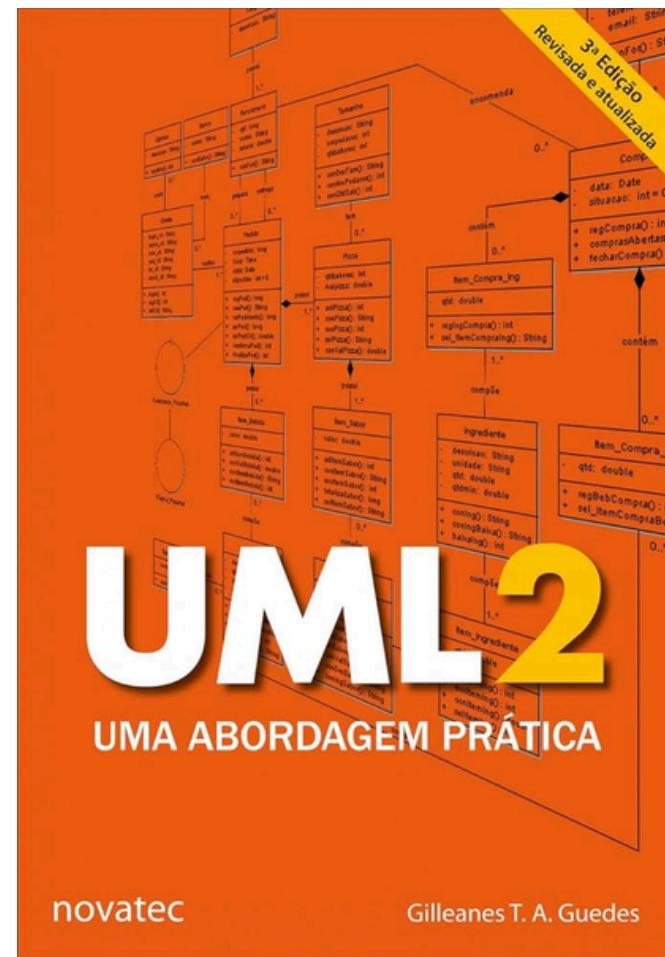
PARA O PRÓXIMO ENCONTRO (14/NOV)

TEMA: Diagrama de Classes UML (Parte 1)

O QUE VEREMOS:

- Conceitos de Orientação a Objetos
- Estrutura de uma classe (atributos e métodos)
- Tipos de relacionamentos entre classes
- Visibilidade (public, private, protected)

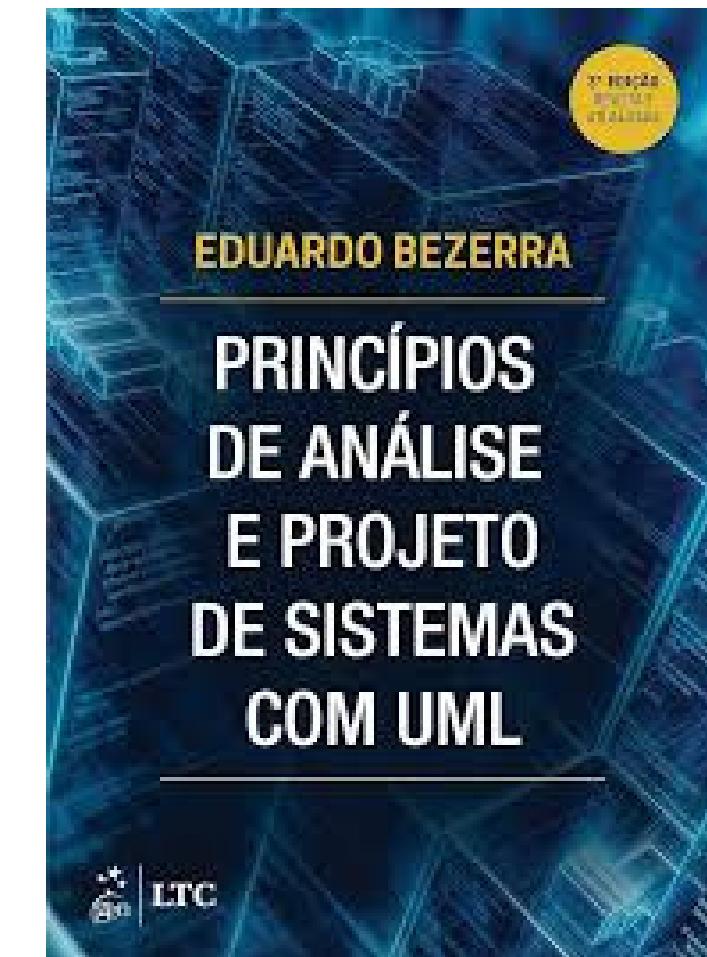
Referências e Materiais de Apoio



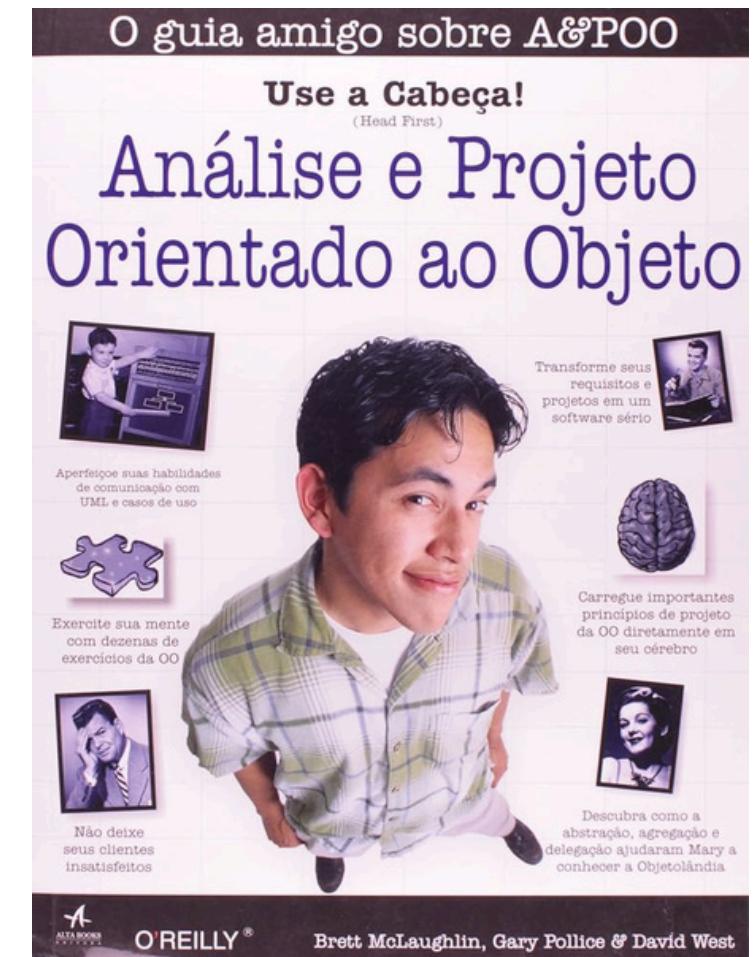
**UML 2 - Uma Abordagem
Prática -**
Gilleanes T. A. Guedes



**Engenharia de
Software -**
Ian Sommerville

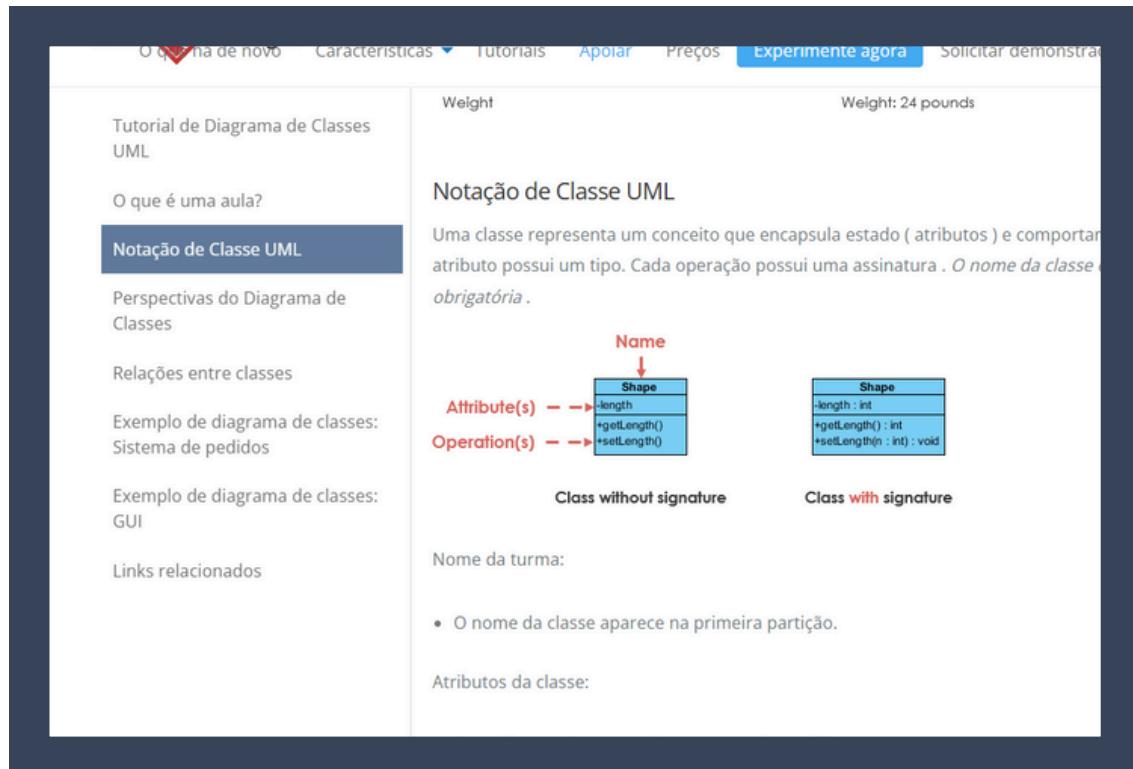
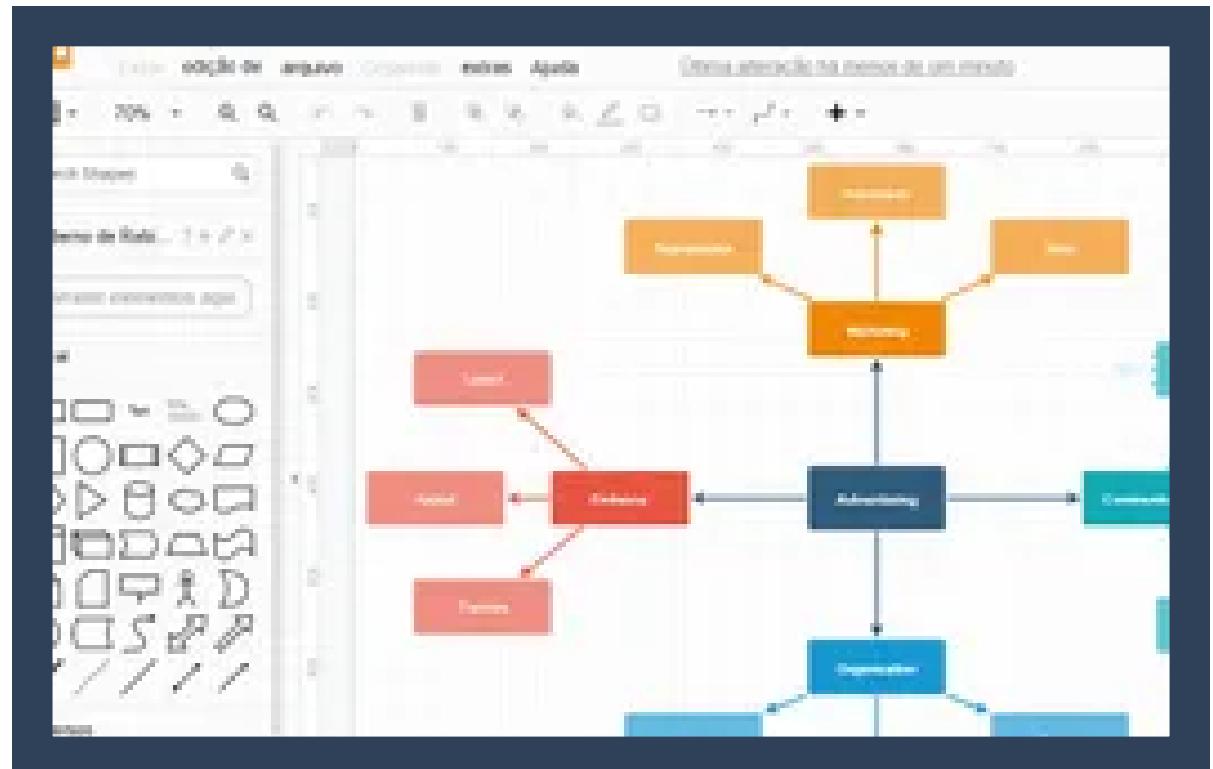
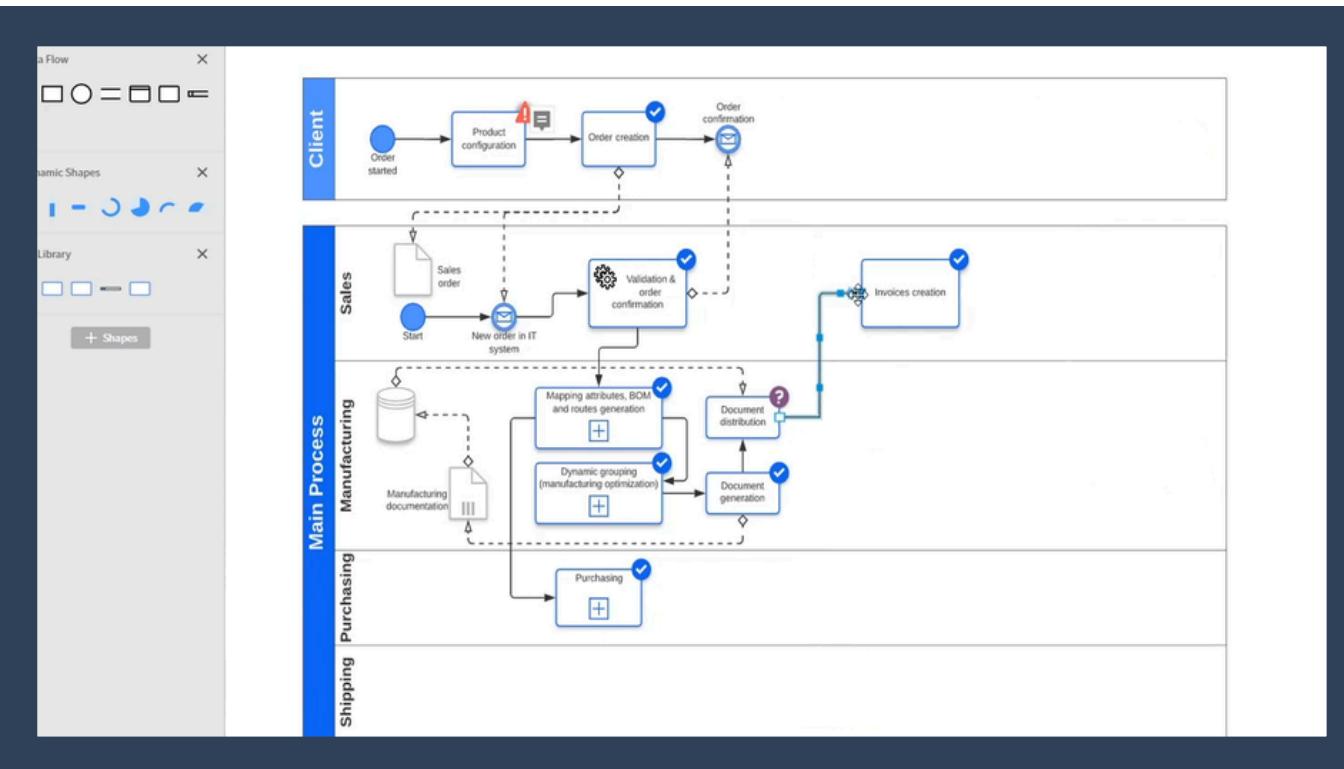


**Análise e Projeto de
Sistemas**
- Eduardo Bezerra



**• Use a Cabeça! Análise e
Projeto Orientado ao Objeto**
- Brett McLaughlin

SITES E FERRAMENTAS:



- [lucidchart.com](https://www.lucidchart.com) - Criação de
diagramas

- draw.io - Diagramas
gratuitos

- [visual-paradigm.com](https://www.visual-paradigm.com) -
Tutoriais UML



SENAI