

MODELAGEM DE SOFTWARE - APOO



Junio Figueiredo

Professor

junioinf@gmail.com

Aula 03

Diagrama de Classes e
Associação

MODELAGEM DE SOFTWARE - APOO



Associações

Engenharia de Software - APOO

Cada ocorrência dos meus conceitos recebe o nome de INSTÂNCIA ou OBJETO

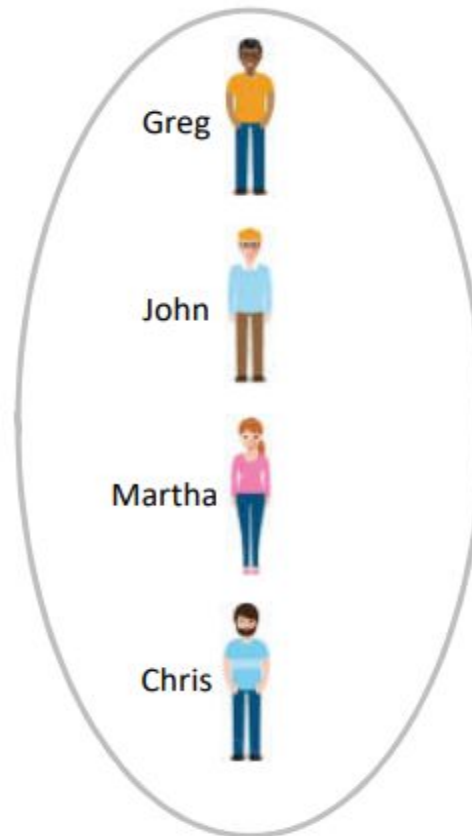
Instâncias

Exemplo:

Desejo criar um sistema para armazenar informações de pessoas e carros.

Conceitos:

- Pessoa
- Carro



Engenharia de Software - APOO



O que são associações

Associação é um relacionamento estático entre dois conceitos.

Exemplo:

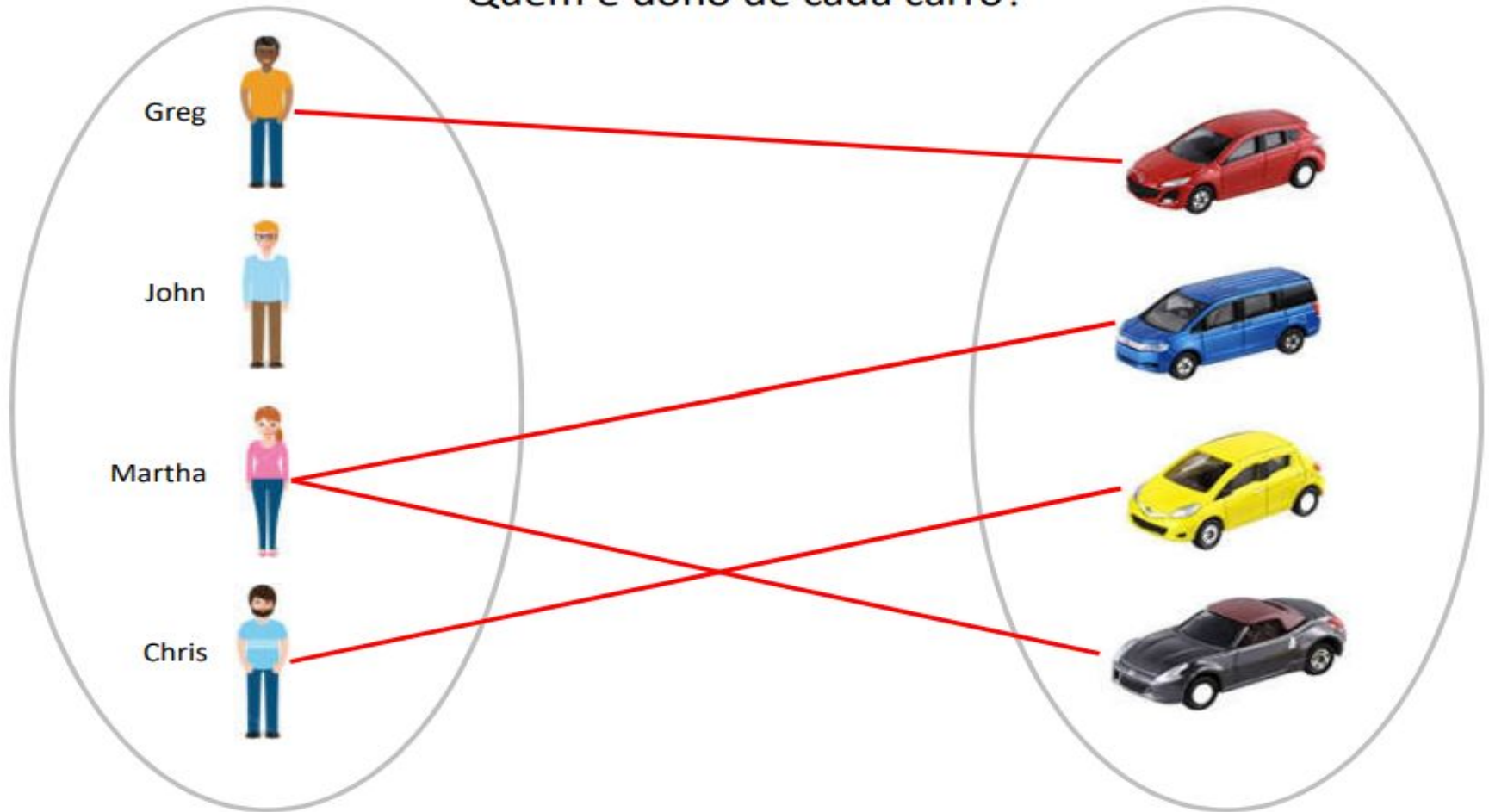
Desejo criar um sistema para armazenar informações de pessoas e carros.

Mas eu não tenho simplesmente a necessidade de saber quais são as pessoas e quais são os carros:

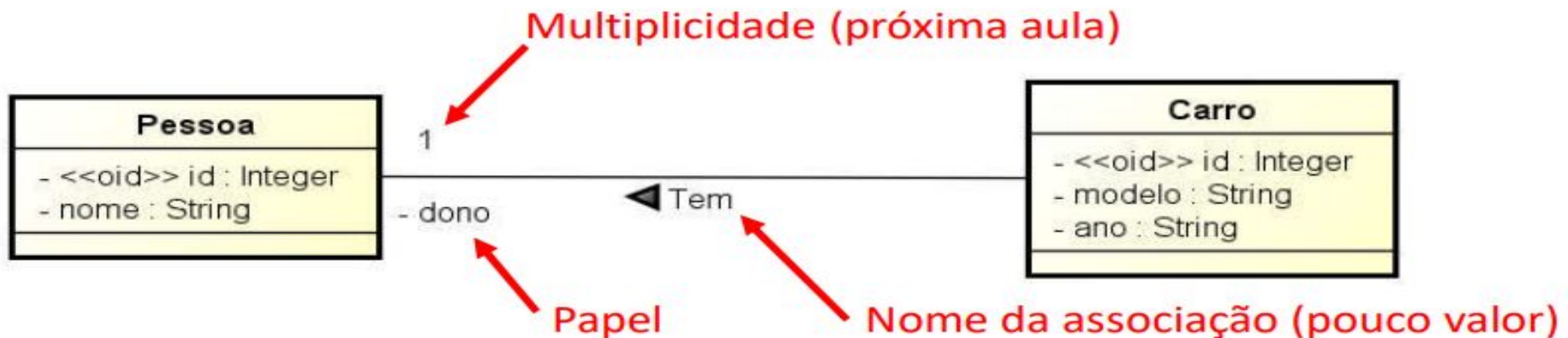
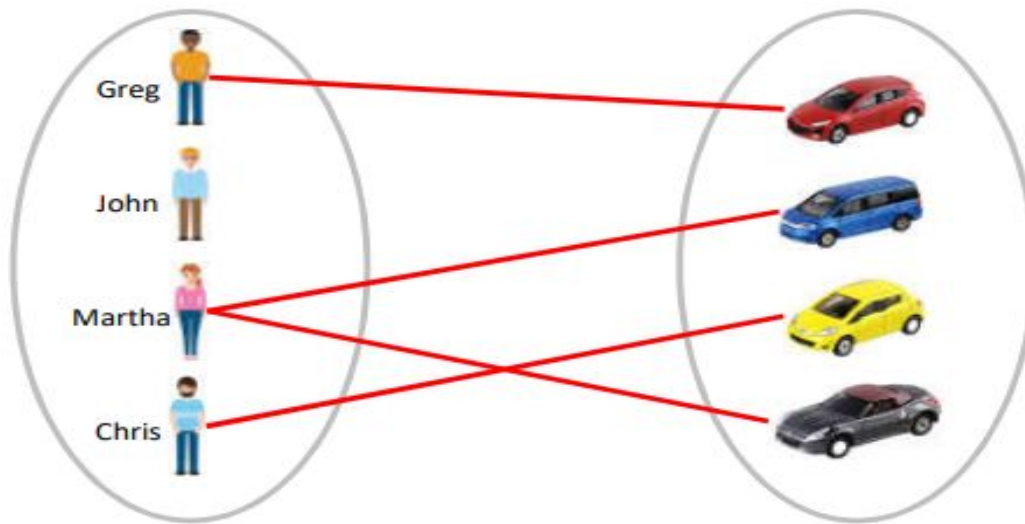
Também desejo saber quem é o dono de cada carro!

Engenharia de Software - AP00

Quem é dono de cada carro?



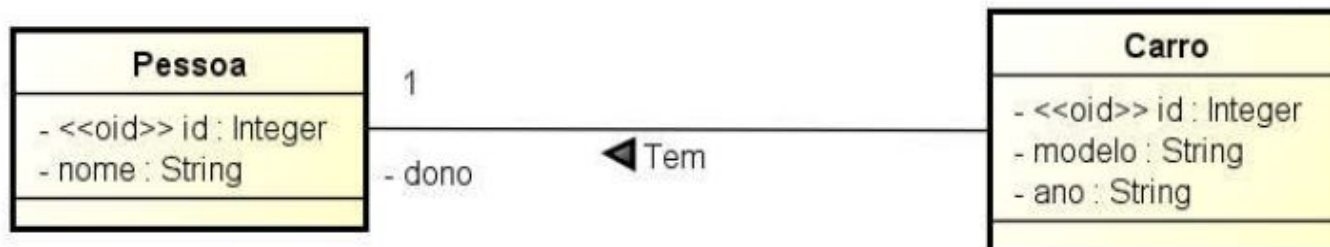
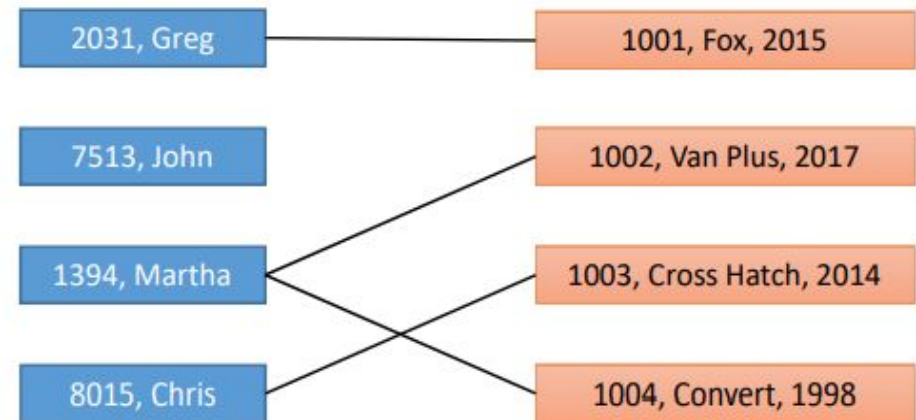
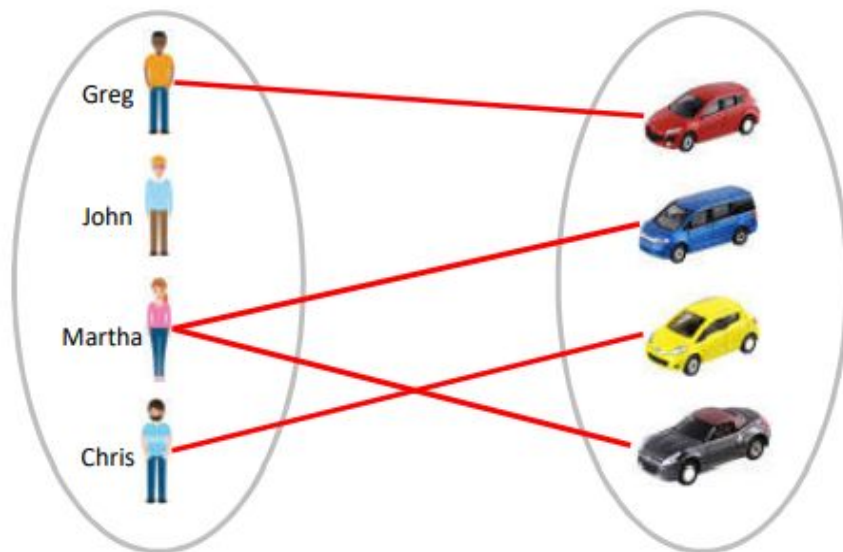
Engenharia de Software - AP00



Engenharia de Software - APOO

Mundo real

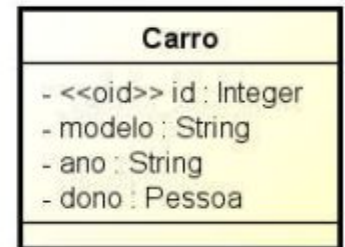
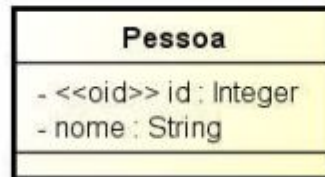
Objetos



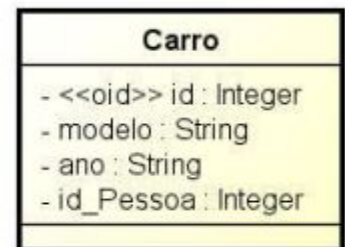
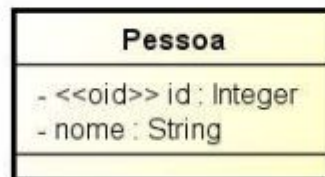
Engenharia de Software - AP00

Atenção!

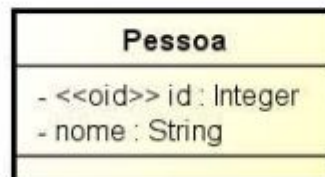
ERRADO



MUITO ERRADO

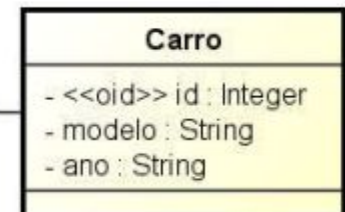


CORRETO



1

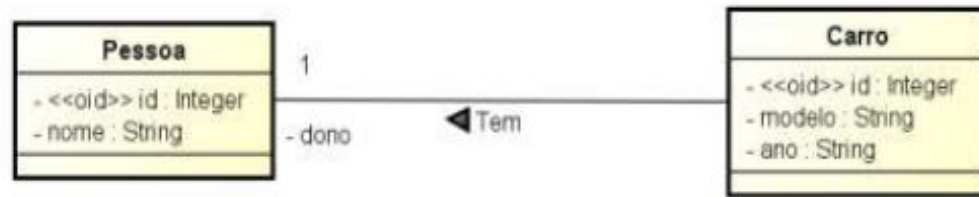
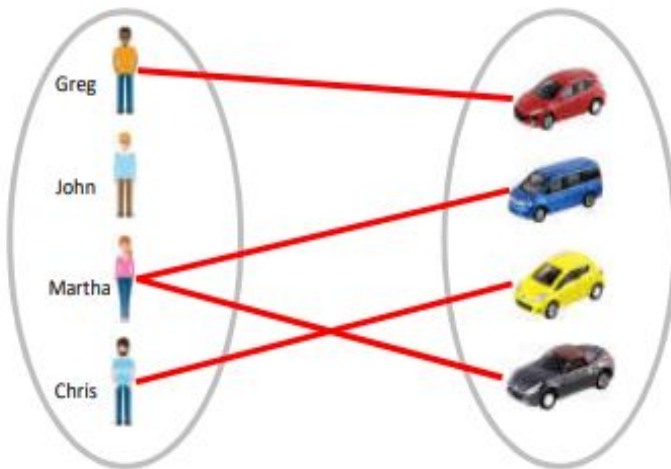
- dono



Engenharia de Software - APOO

Resumo da aula

- Associação é um relacionamento estático entre dois conceitos



- Nome da associação (pouco valor)
- Nome de papel
- Multiplicidade (próxima aula)

- Não confunda com Modelo Relacional

Engenharia de Software - AP00



Multiplicidade

Engenharia de Software - APOO

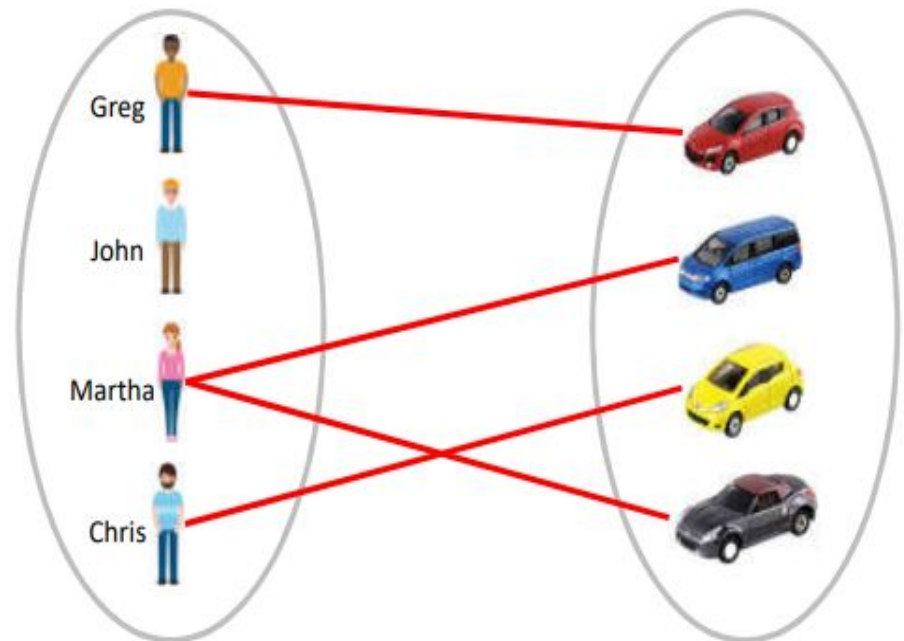
O que é multiplicidade?

É a quantidade mínima e máxima de objetos que uma associação permite em cada um de seus papéis.

Exemplo: um carro pode ter quantos donos?

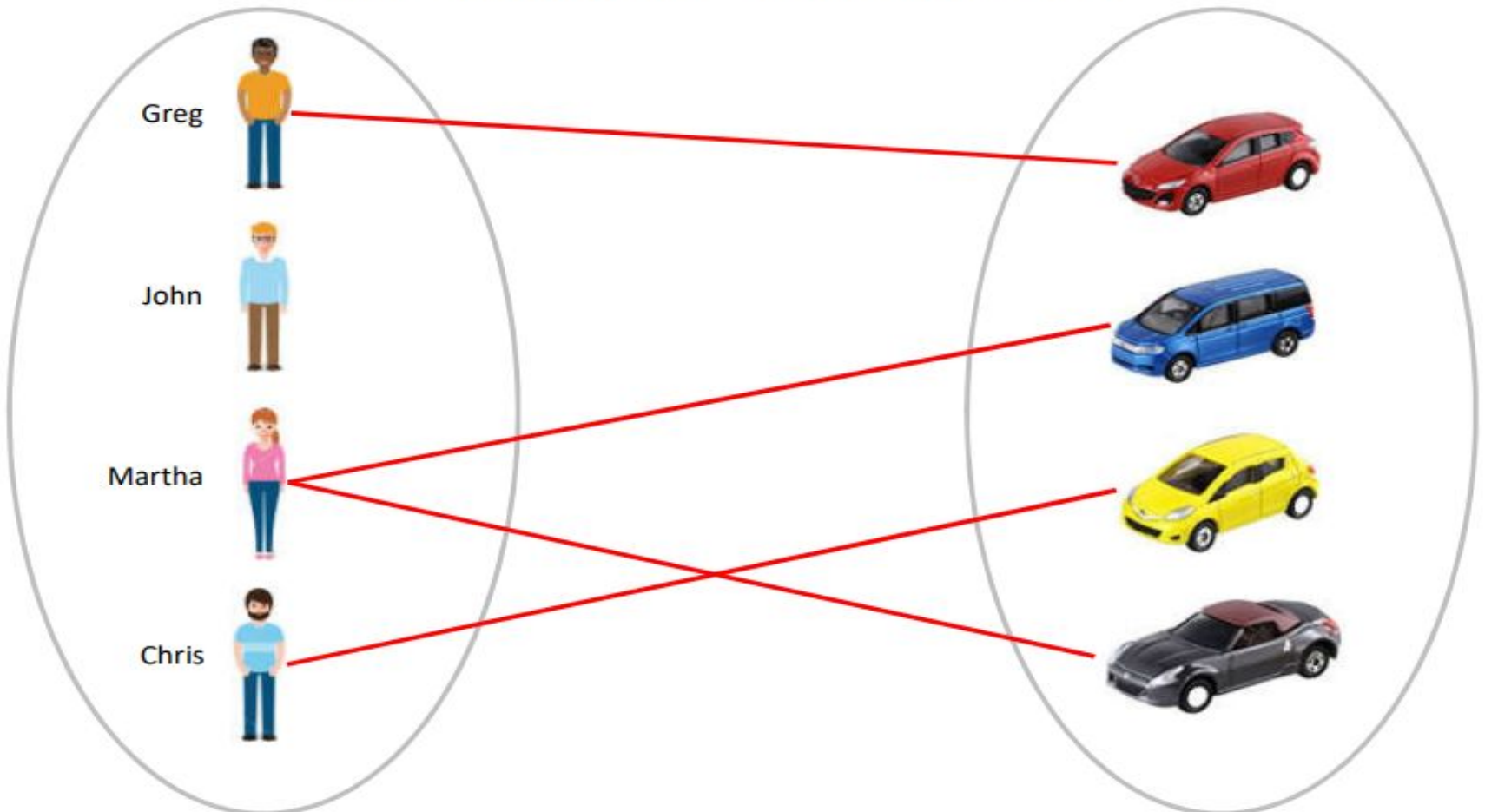
Mínimo: 1

Máximo: 1



Engenharia de Software - APOO

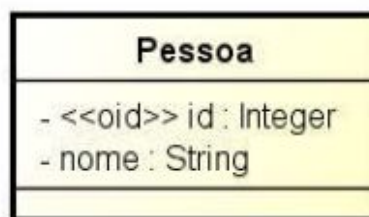
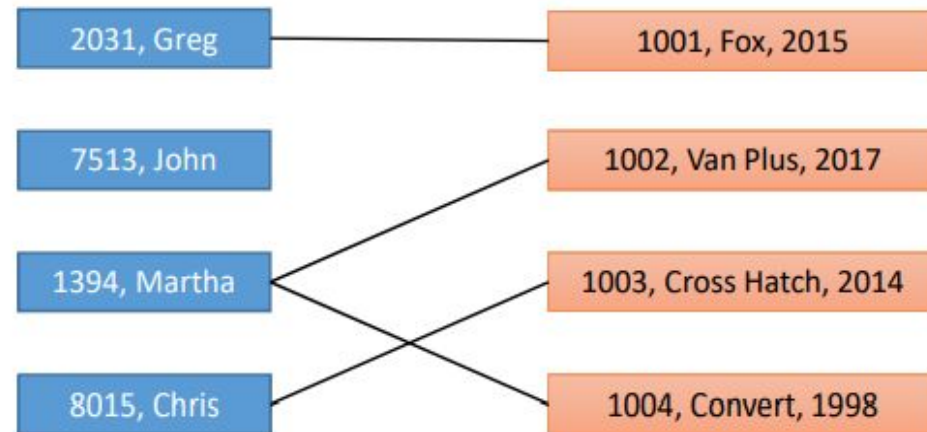
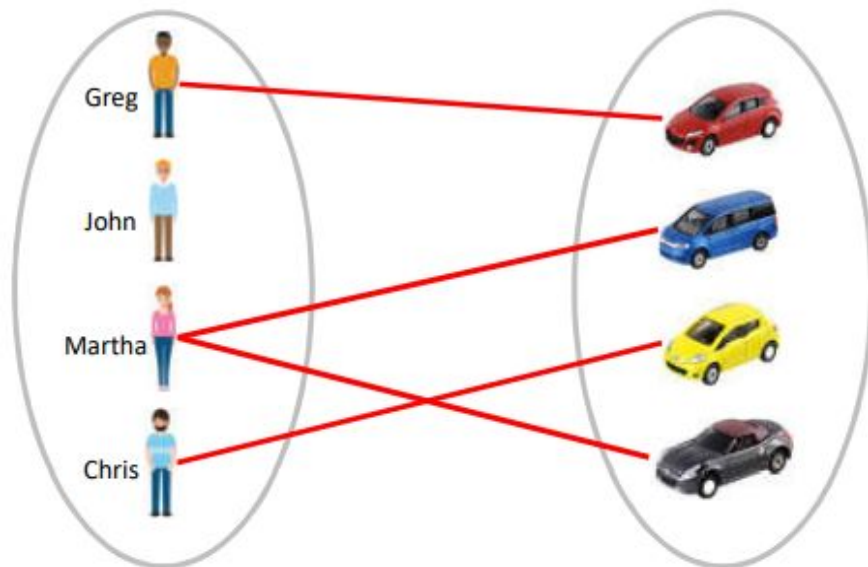
Recordando: quem é dono de qual carro?



Engenharia de Software - APOO

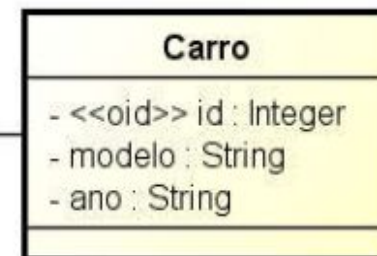
Mundo real

Objetos



1

- dono

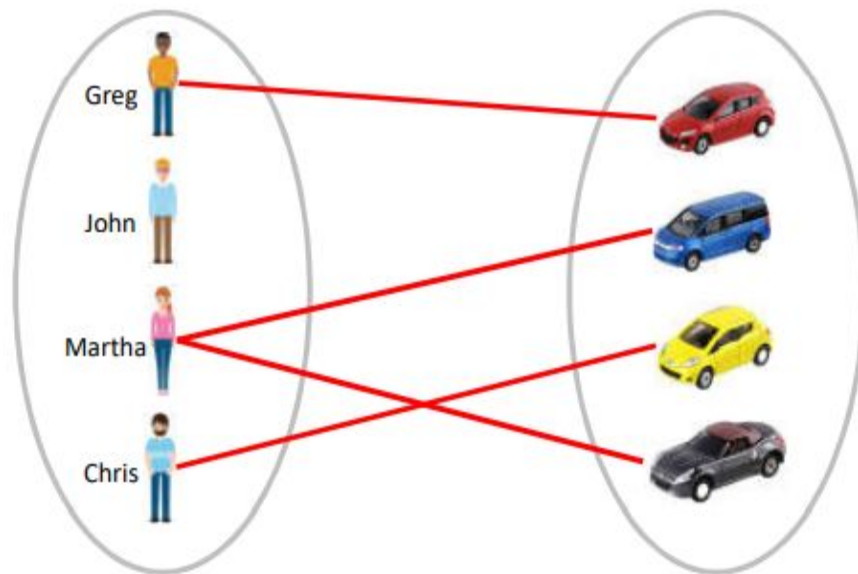


Engenharia de Software - APOO



Como encontrar as
multiplicidades?

Engenharia de Software - AP00



Pergunte para os dois lados:

1 <conceito> pode ter quantos <papel>?

1 carro pode ter quantos donos?

1 pessoa pode ter quantos carros?

Atenção:
sempre inicie
a pergunta
com "1"



Engenharia de Software - AP00

Multiplicidades possíveis

"," significa "ou"

".." significa "a"

"*" significa "vários" (sem limite específico)

a)	1	exatamente um
b)	2	exatamente dois
c)	0..1	zero a um
d)	0..*	zero ou mais
e)	*	zero ou mais
f)	1..*	um ou mais
g)	2..*	dois ou mais
h)	2..5	de dois a cinco
i)	2,5	dois ou cinco
j)	2,5..8	dois ou cinco a oito

Engenharia de Software - APOO

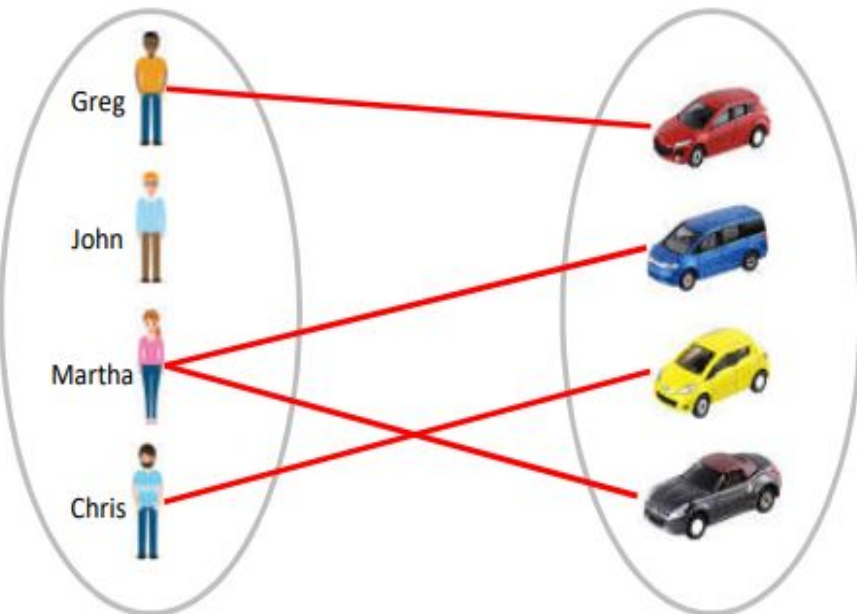


Associações comuns

Engenharia de Software - APOO

Um para muitos

Exemplo: quem é dono de cada carro?

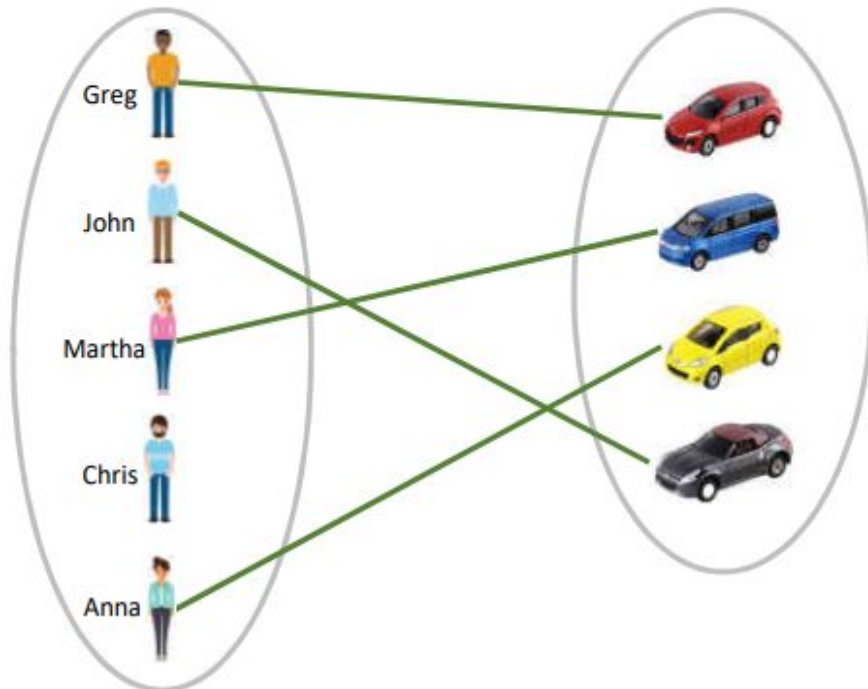


- Em um dos lados o máximo é 1
- No outro lado o máximo é "vários"

Engenharia de Software - APOO

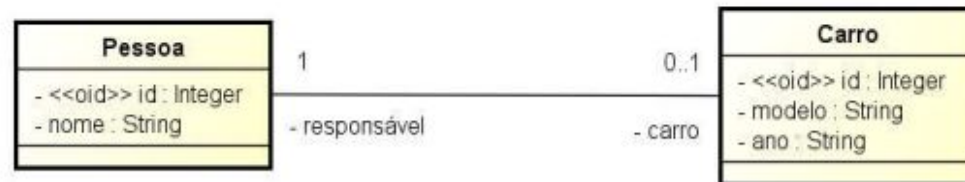
Um para um

Exemplo: quem é o responsável por cada carro?



1 carro pode ter quantos responsáveis?

1 pessoa pode ser responsável por quantos carros?

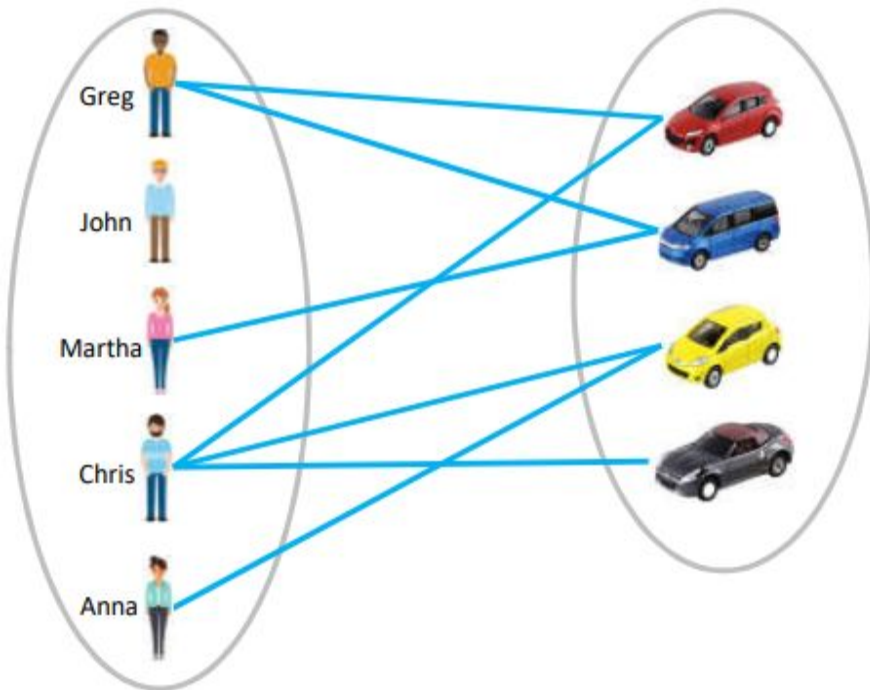


- Em ambos os lados o máximo é 1

Engenharia de Software - AP00

Muitos para muitos

Exemplo: quem dirige cada carro?



1 carro pode ter quantos motoristas?

1 pessoa pode dirigir quantos carros?



- Em ambos os lados o máximo é "vários"

Engenharia de Software - APOO



Resumo da aula

- Multiplicidade é a quantidade **mínima** e **máxima** de objetos que uma associação permite em cada um de seus papéis.
- Como encontrar as multiplicidades?
 - Pergunte para os dois lados: **1** <conceito> pode ter quantos <papel>?
- Multiplicidades possíveis
- Associações comuns
 - Um para muitos
 - Um para um
 - Muitos para muitos

Engenharia de Software - APOO



**Conceito dependente
Associações obrigatórias,
múltiplas e autoassociações**

Engenharia de Software - APOO

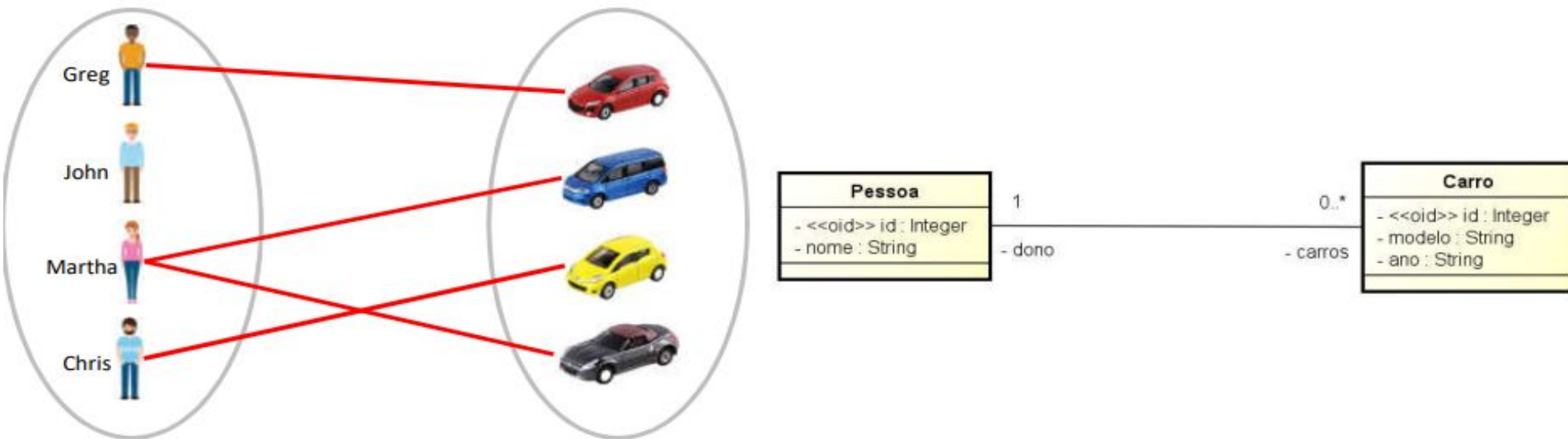


Associação obrigatória

Engenharia de Software - APOO

Definição

Uma associação é obrigatória se o conceito associado desempenha um papel de multiplicidade mínima maior que zero



- A associação de uma pessoa com carros não é obrigatória.
- A associação de um carro com dono é obrigatória.

Engenharia de Software - APOO

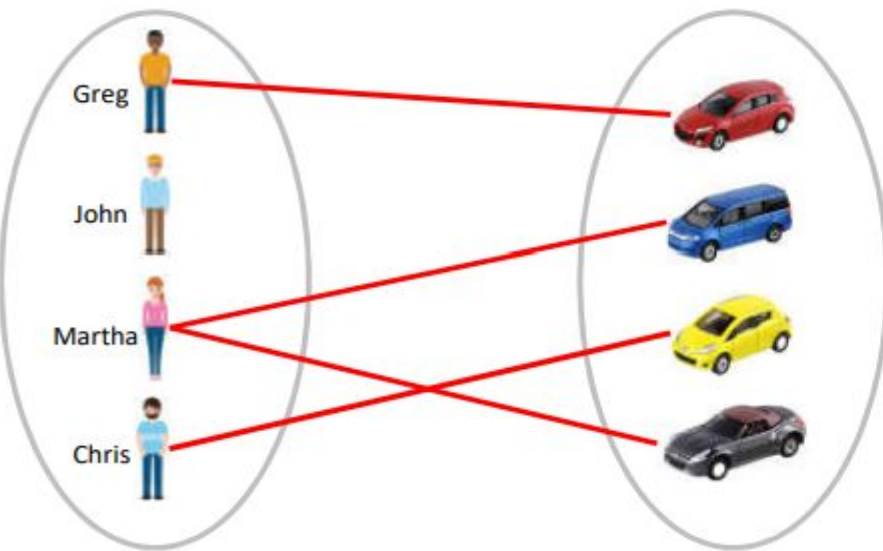


Conceito dependente

Engenharia de Software - APOO

Definição

Um conceito é dependente se ele possuir pelo menos uma associação obrigatória.



- Pessoa é um conceito independente
 - Carro é um conceito dependente
- Só pode existir se existir uma pessoa dona dele
- Se o dono deixar de existir, o carro também deixa de existir

Engenharia de Software - APOO

Nota

A UML tem um símbolo que denota dependência de um modo geral, mas que não acrescenta valor prático à modelagem conceitual:



"Carro depende de pessoa"

Engenharia de Software - APOO

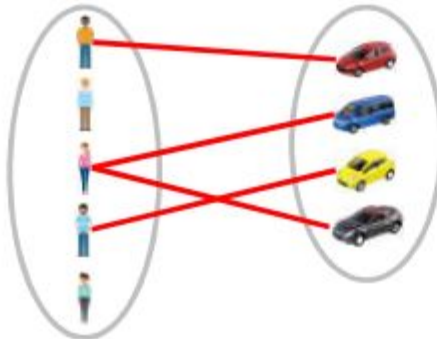


Associações múltiplas

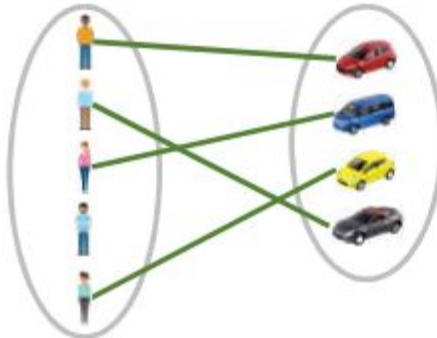
Pode haver mais de uma associação entre dois conceitos?

Engenharia de Software - APOO

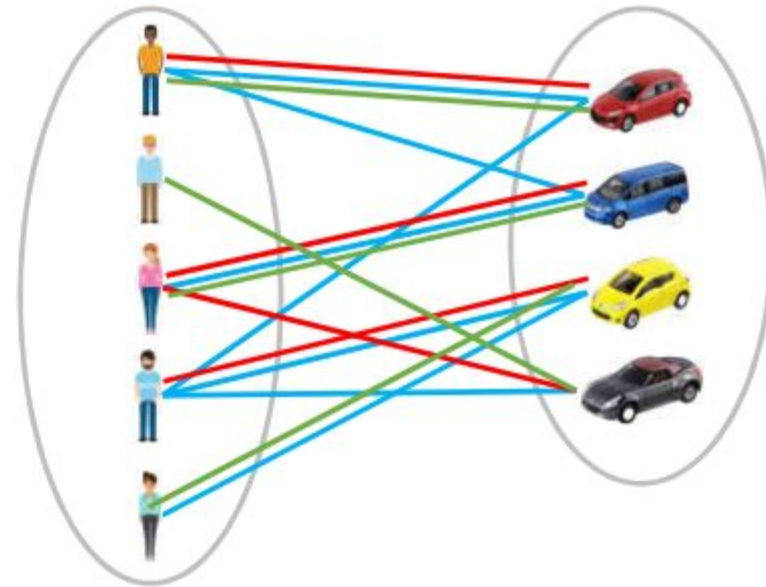
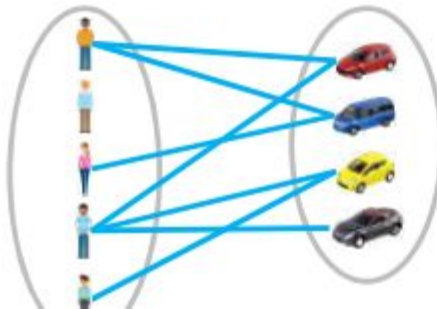
Quem é dono de cada carro?



Quem é responsável por cada carro?



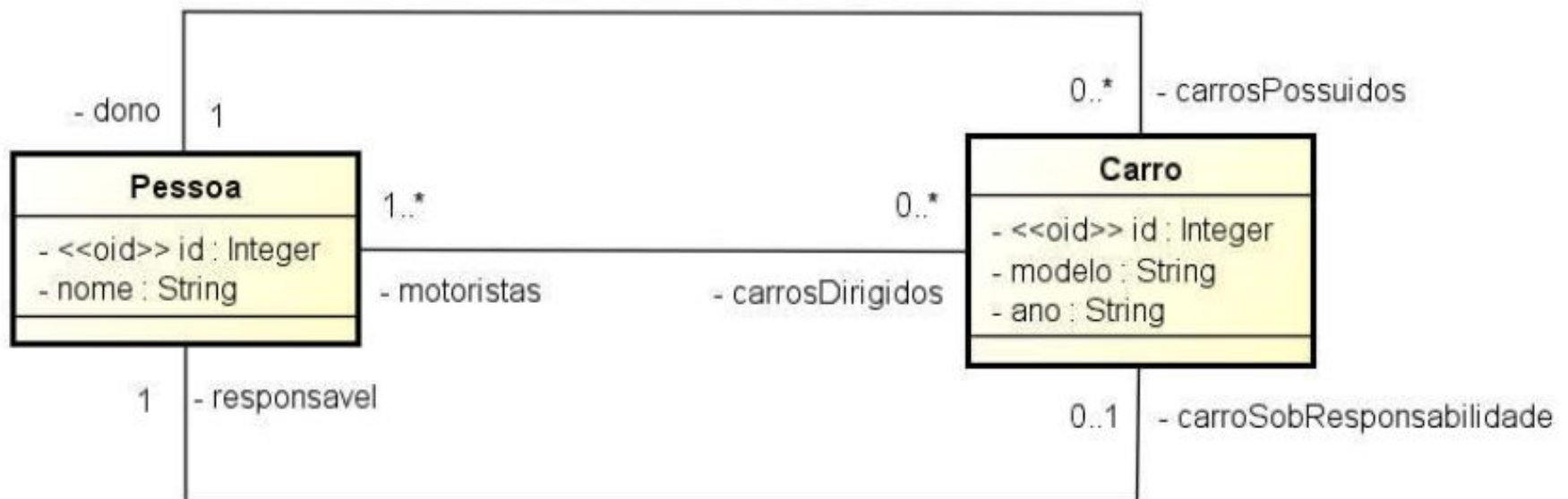
Quem dirige cada carro?



É possível um modelo de conceitual com todas as três informações ao mesmo tempo?

Engenharia de Software - APOO

Sim. Mas os nomes de papel devem ser únicos.



Engenharia de Software - APOO

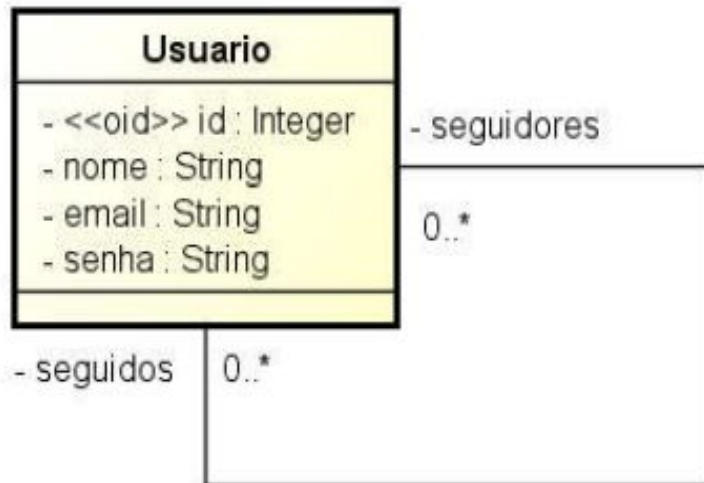


Autoassociações

Engenharia de Software - APOO

Autoassociações

Quando um conceito é associado com ele próprio.



1 usuário pode ter quantos seguidores?

1 usuário pode ter quantos seguidos?

Engenharia de Software - APOO



**Desenhando instâncias com
diagrama de objetos da UML**

Engenharia de Software - APOO

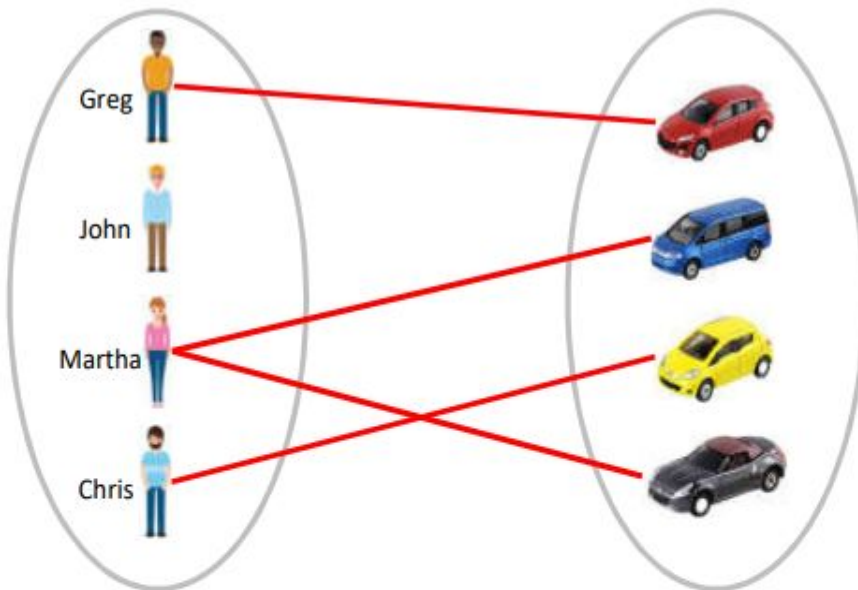


Relembrando:

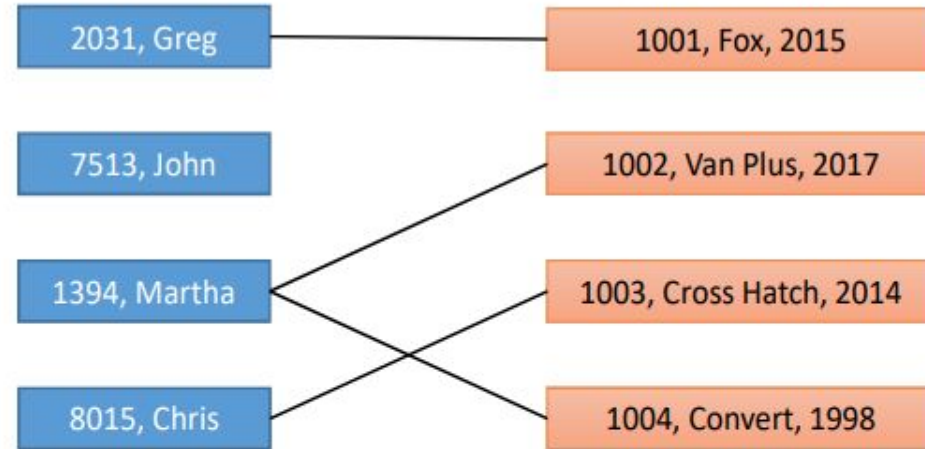
- O Modelo Conceitual representa a estrutura dos dados
 - Conceitos/atributos e como eles se inter-relacionam entre si
- Cada ocorrência de um conceito é chamada de instância ou objeto

Engenharia de Software - APOO

Mundo real



Objetos



Engenharia de Software - APOO

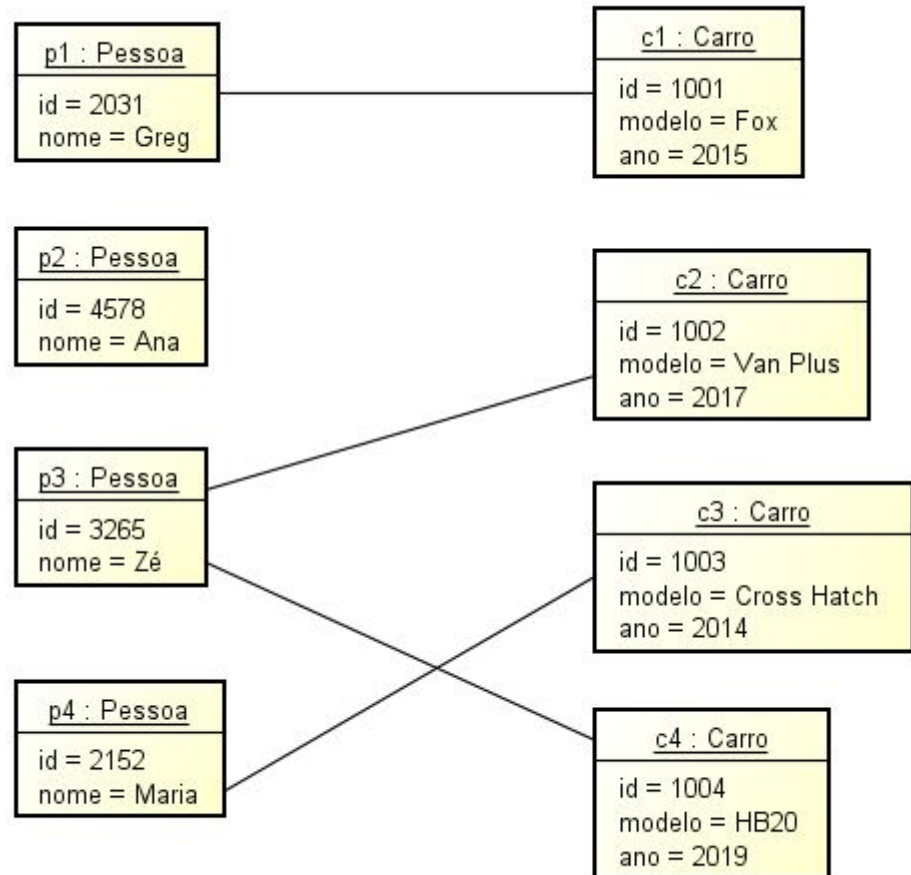


Pra quê visualizar as instâncias (ou objetos)?

- Ajuda a compreender
- Ajuda a descobrir problemas
- Ferramenta UML: diagrama de objetos

Engenharia de Software - APOO

Diagrama de Objetos:



Engenharia de Software - APOO



Exercício1

- 1 - Desenhar o Modelo Conceitual
- 2 - Esboçar uma instância atendendo os requisitos mínimos pedidos

Deseja-se construir um sistema para manter um registro de artistas musicais e seus álbuns. Cada álbum possui várias músicas, as quais poderão ser consultadas pelo sistema.

O sistema também deve permitir a busca de artistas por nome ou nacionalidade. O sistema também deve ser capaz de exibir um relatório dos álbuns de um artista, o qual pode ser ordenado por nome, ano, ou duração total do álbum. Um álbum pode ter a participação de vários artistas, sem distinção. Já a música pode possuir um ou mais autores e intérpretes (todos considerados artistas).

Instância mínima: 2 artistas, 3 álbuns, 4 músicas

Engenharia de Software - APOO



Exercício1

Exercício 1 (RESOLVIDO): Deseja-se construir um sistema para manter um registro de artistas musicais e seus álbuns. Cada álbum possui várias músicas, as quais poderão ser consultadas pelo sistema. O sistema também deve permitir a busca de artistas por nome ou nacionalidade. O sistema também deve ser capaz de exibir um relatório dos álbuns de um artista, o qual pode ser ordenado por nome, ano, ou duração total do álbum. Um álbum pode ter a participação de vários artistas, sem distinção. Já a música pode possuir um ou mais autores e intérpretes (todos considerados artistas).

Engenharia de Software - APOO



Exercício1

Artista
- <<oid>> id : Integer
- nome : String
- nacionalidade : String

Album
- <<oid>> id : Integer
- nome : String
- ano : Integer

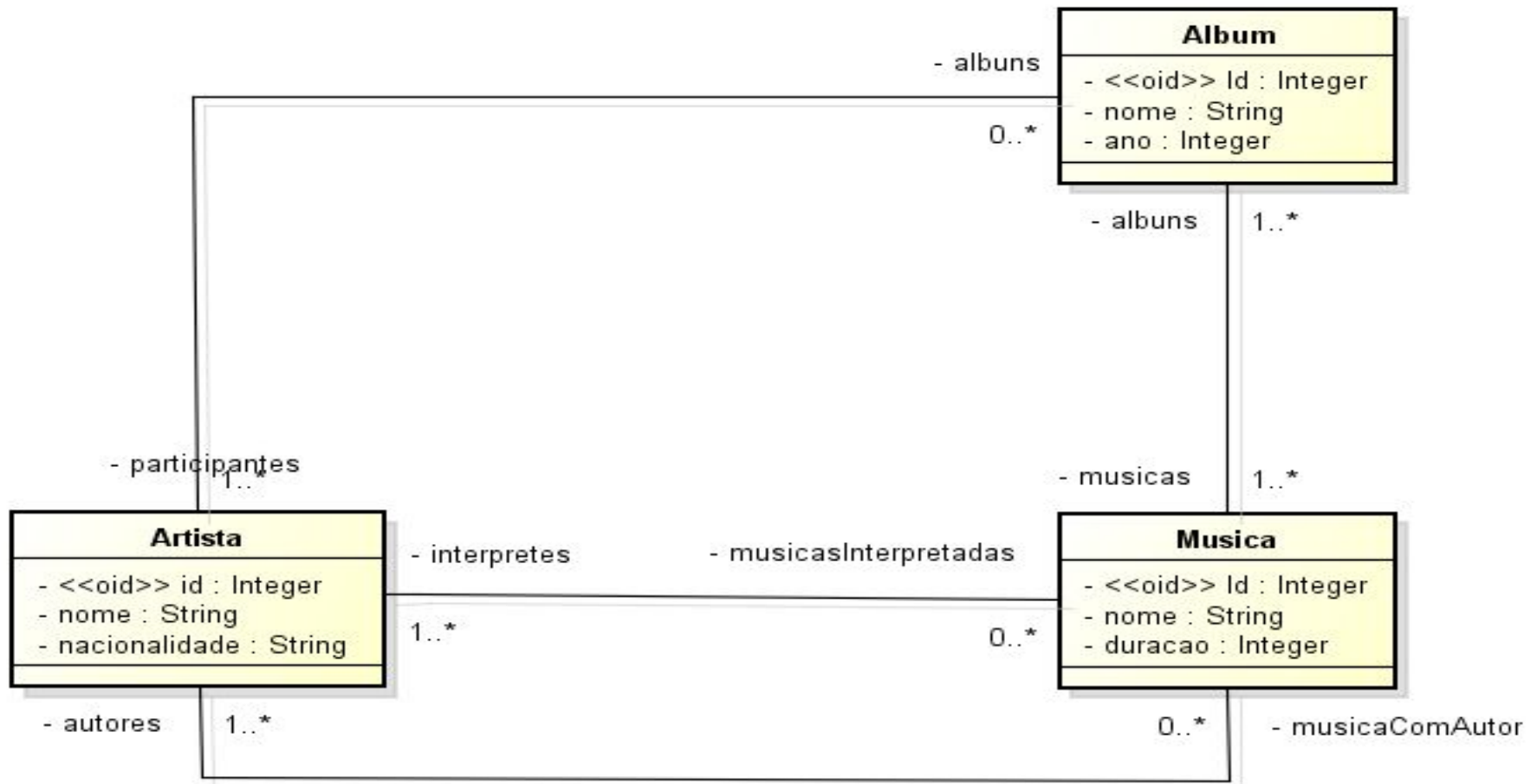
Musica
- <<oid>> id : Integer
- nome : String
- duracao : Integer

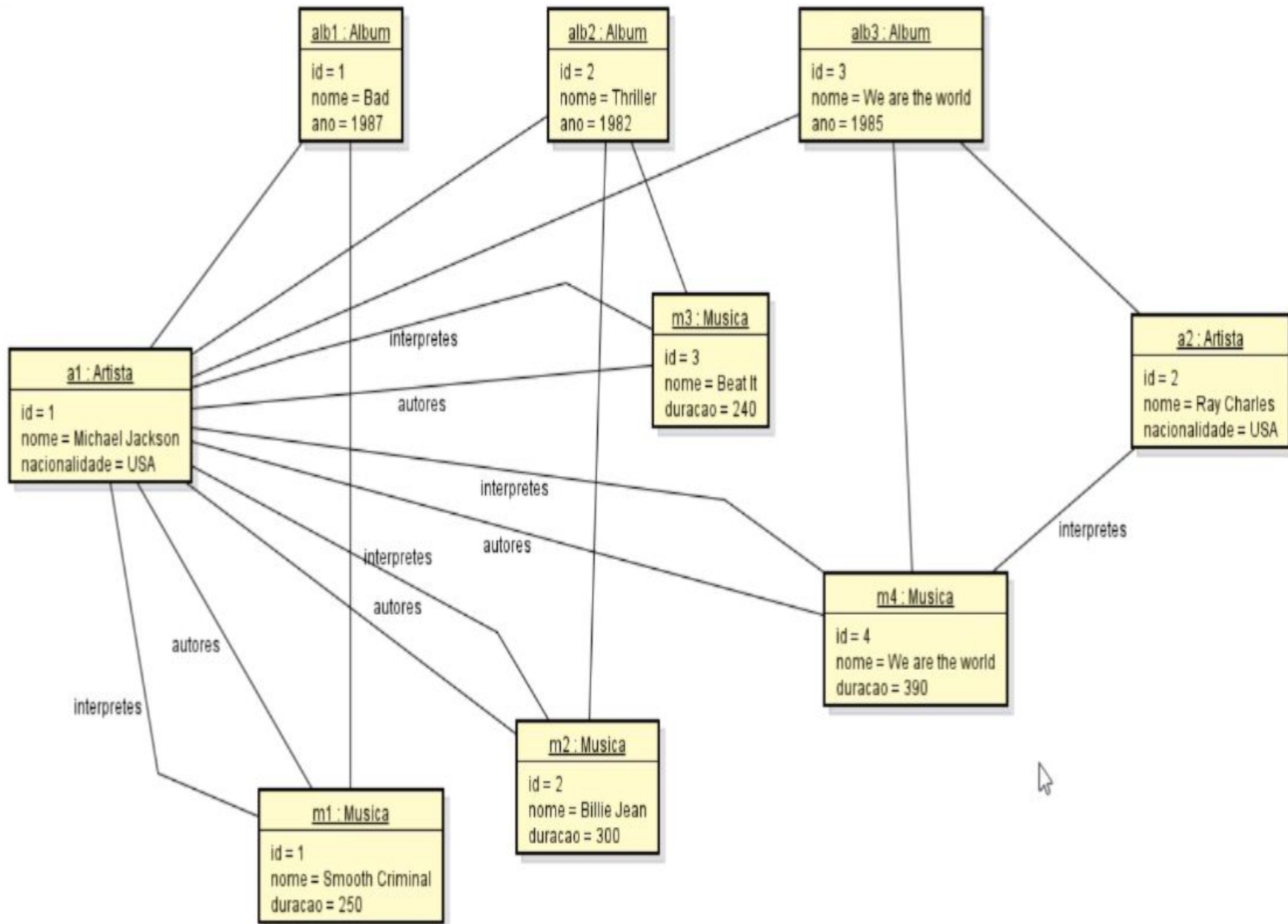
Engenharia de Software - APOO



Exercício1

Engenharia de Software - APOO





Engenharia de Software - APOO



Exercício 2

Deseja-se construir um sistema para gerenciar as informações de campeonatos de handebol, que ocorrem todo ano. Deseja-se saber nome, data de nascimento, gênero e altura dos jogadores de cada time, bem qual deles é o capitão de cada time. Cada partida do campeonato ocorre em um estádio, que possui nome e endereço. Cada time possui seu estádio-sede e, assim, cada partida possui um time mandante (anfitrião) e o time visitante. O sistema deve ser capaz de listar as partidas já ocorridas e não ocorridas de um campeonato. O sistema deve também ser capaz de listar a tabela do campeonato, ordenando os times por classificação, que é calculada em primeiro lugar por saldo de vitórias e em segundo lugar por saldo de gols.

Engenharia de Software - APOO



Exercício 2

Deseja-se construir um sistema para gerenciar as informações de **campeonatos** de handebol, que ocorrem todo **ano**. Deseja-se saber **nome**, **data de nascimento**, **gênero** e **altura** dos **jogadores** de cada **time**, bem qual deles é o **capitão** de cada time. Cada **partida** do campeonato ocorre em um **estádio**, que possui **nome** e **endereço**. Cada time possui seu **estádio-sede** e, assim, cada partida possui um time **mandante** (anfitrião) e o time **visitante**. O sistema deve ser capaz de listar as partidas já **ocorridas e não ocorridas** de um campeonato. O sistema deve também ser capaz de listar a **tabela** do campeonato, ordenando os times por **classificação**, que é calculada em primeiro lugar por **saldo de vitórias** e em segundo lugar por **saldo de gols**.

Engenharia de Software - APOO

Resposta - Exercício 2

Campeonato
- <<oid>> id : Integer
- ano : Integer

Jogador
- <<oid>> id : Integer
- nome : String
- nascimento : Date
- genero : String
- altura : Double

Time
- <<oid>> id : Integer
- nome : String

Partida
- <<oid>> id : Integer
- data : Date
- pontuacaoMandante : Integer
- pontuacaoVisitante : Integer

Estadio
- <<oid>> id : Integer
- nome : String

Endereco
- <<oid>> id : Integer
- logradouro : String
- numero : Integer
- complemento : String
- bairro : String

