

# Diagrama de CLASSES

## Modelagem de software

Professor: Jonny I. Beckert

## ABRIL

S	T	Q	Q	S	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

## EVENTOS

- 7** Feriado de Sexta-Feira Santa
- 9** Feriado de Páscoa
- 21** Feriado de Tiradentes

**9, 10 e 11**

Avaliação - A1 em 1ª oportunidade

**13, 14 e 15**

Avaliação - A2 em 1ª oportunidade

**21 a 27**

Avaliação - A3

} **MAIO**

} **JUNHO**

# Contextualização

Faça um diagrama de classes que:  
Represente de maneira simplificada um banco qualquer.  
Para facilitar uma pequena contextualização:

Semana passada

1. Identificar nomes de classes
2. Distinguir os atributos
3. Distinguir os métodos
4. Distinguir as relações

Um determinado banco localizado num endereço central da cidade, possui vários empregados.

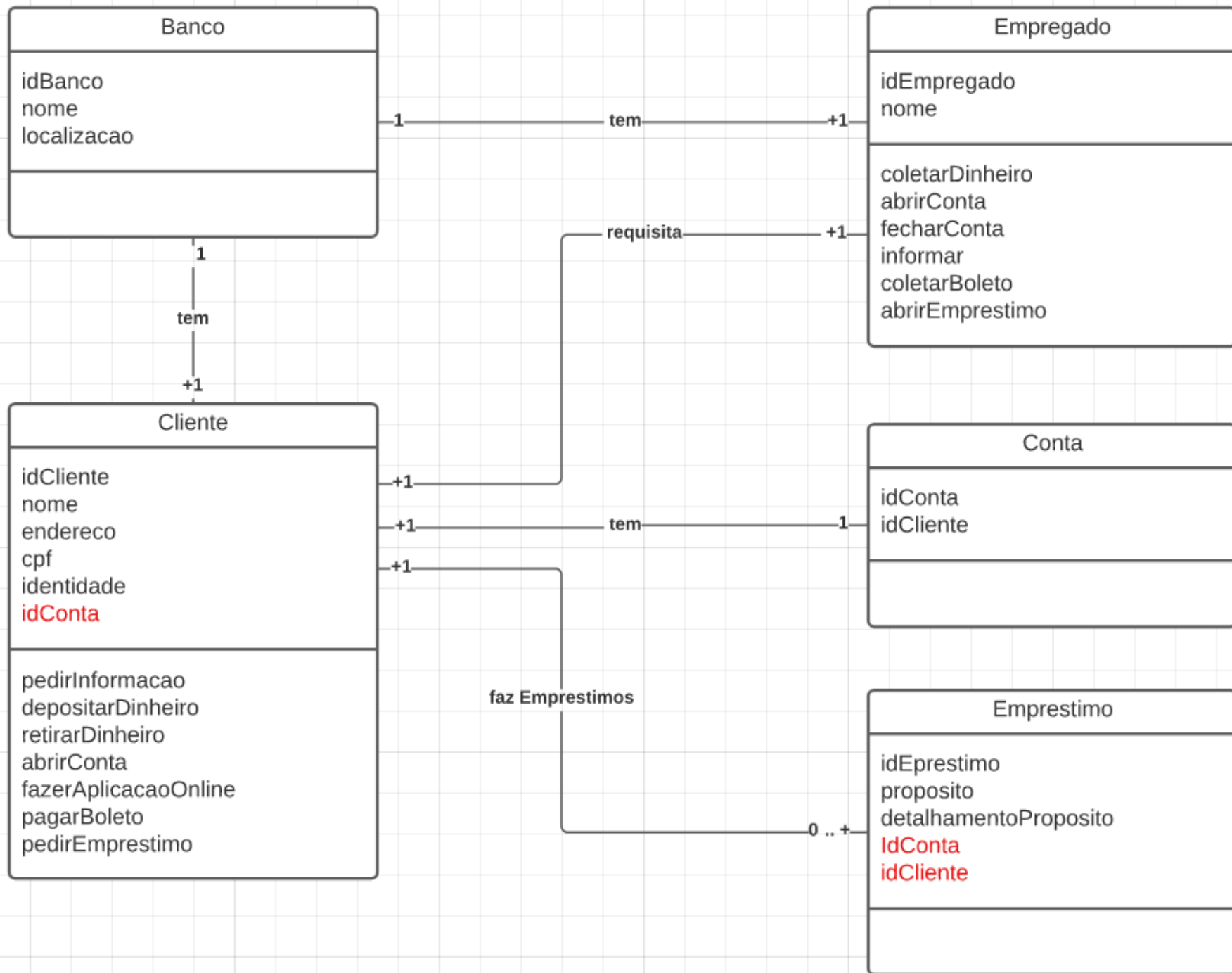
Por sua vez este mesmo banco também possui diversos clientes.

Estes clientes devem ter um identificador pessoal e um identificador de conta. Podem fazer diversas ações e inquéritos, bem como depositar dinheiro, sacar dinheiro, Abrir conta, Fechar conta, Fazer pagamentos de boletos, Fazer aplicações online, pedir empréstimo. Informações de nome, identificação e endereço do cliente devem estar sempre atualizados. Um cliente pode ter uma única conta neste banco.

Já os funcionários do banco também devem ter um identificador pessoal e executam os seguintes serviços: Coletar dinheiro, Abrir conta para clientes, Fechar contas para clientes, Providenciar informações, Coletar pagamentos de boletos, atender pedido de empréstimos.

As contas são identificadas por um identificados e podem ter mais de um cliente associado a esta conta.

Os pedidos de empréstimos devem ser associadas a uma conta e cliente, devem ter especificado o proposito da conta bem como o detalhamento deste proposito .



# Tarefa de casa

Fizeram ??

- Terminar e entregar o diagrama de classes iniciado na sala
- Fazer um diagrama de classes sobre a seguinte contextualização:
  - Uma empresa administra eventos.
  - Para isso possui várias salas com diferentes tamanhos e capacidades.
  - Cada sala onde ocorre um determinado evento possui um identificador e tem capacidade de lotação e recursos de mídia diferentes.
  - Os recursos de mídia são cadastrados e cada item possui um identificador, nome e descritivo da sua utilidade de uso.
  - Os eventos devem ser agendados para data e hora específica, bem como sua duração, recursos necessários (subtende-se aqui recursos como capacidade e mídia necessária).
  - Cada evento é solicitado por um palestrante que tem um cadastro prévio. Caso não tenha, deverá ser providenciado . Este palestrante tem identificador, nome, endereço e meio de entrar em contato.
  - O pagamento pelo uso dos recursos é realizado ao final do evento.
  - Recursos adicionais (comes e bebes) podem ser solicitados e devidamente identificados para posterior cobrança .
  - Cada evento possui um titulo chamativo.
  - Podem ocorrer vários eventos com o mesmo titulo simultaneamente.

Não será cobrado, mas tente informar o tipo de dado e sua visibilidade

# Revisão

## Classe

- **Atributo**
  - Representa características de uma classe (Jogador: nome,sexo,idade....).
- **Método**
  - Representa atividades que um objeto de uma classe pode executar (Jogador:correr,driblar,chutar ...).
- **Visibilidade**
  - Indica o nível de acessibilidade de um atributo ou método.
  - Tipos: Publica(+), Privada (-) e Protegida (#)

# Relacionamento

- Permite compartilhar informações e métodos.
- Descreve um vínculo que ocorre entre os objetos de uma ou mais classes.
- Tipos de relacionamentos:
  - Associação
    - Agregação.
    - Composição.
  - Especialização/Generalização.
  - Dependência.

Associação

Herança

Dependência

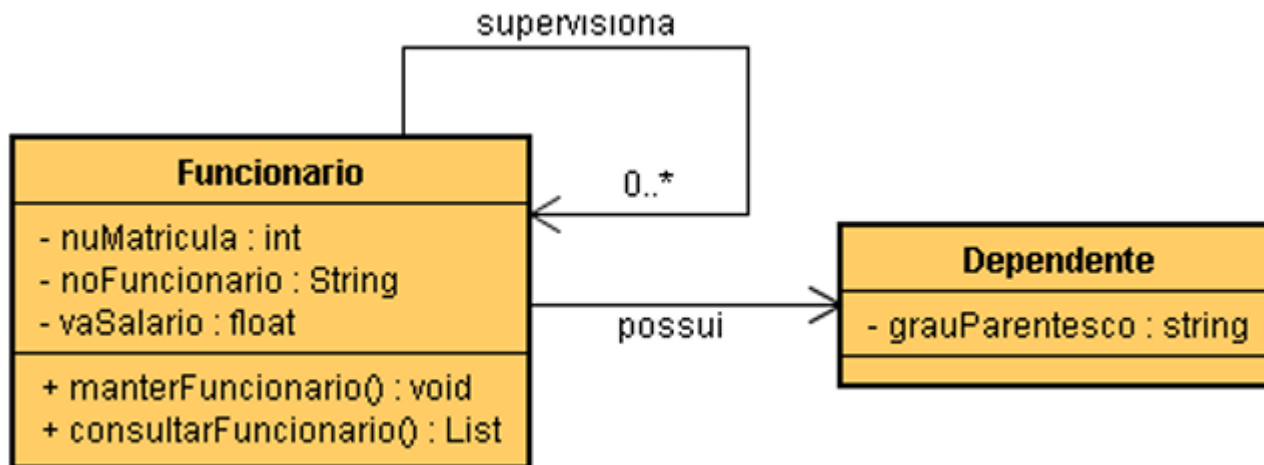
Agregação

Associação

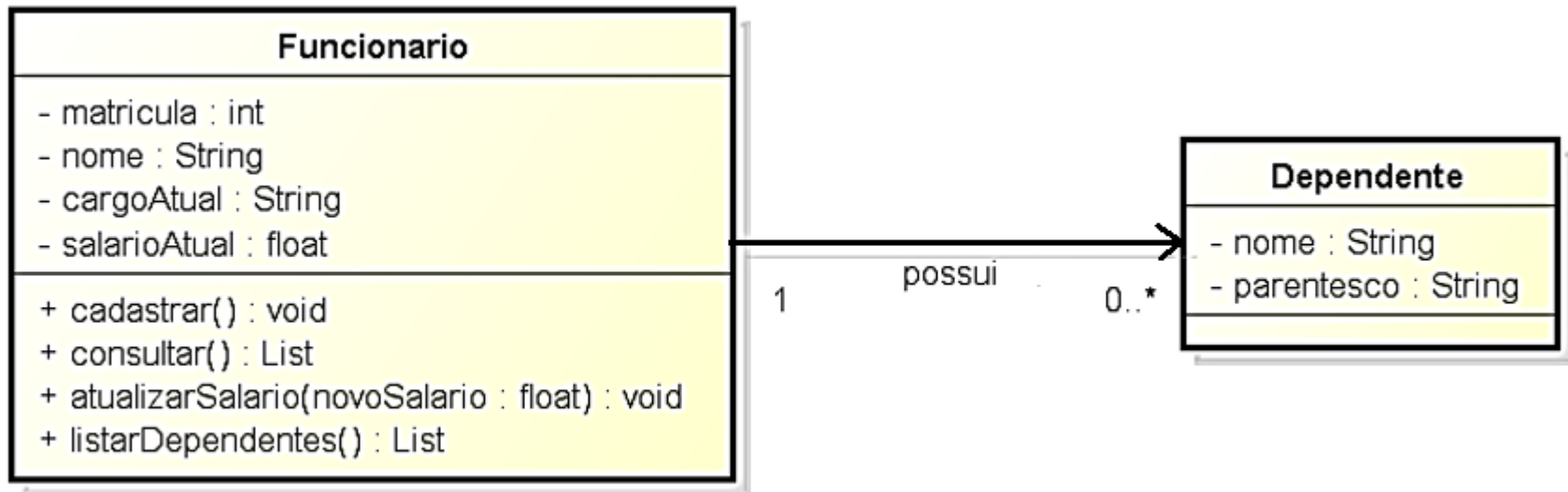


# Associação

- Vinculo entre elementos de um modelo.
- Especifica objetos de um item conectados a objetos de outro item
  - Binária
    - Relacionamento entre duas classes.
    - Podem possuir títulos para determinar o tipo de vinculo.
  - Unária
    - Relacionamento de uma classe consigo mesma.





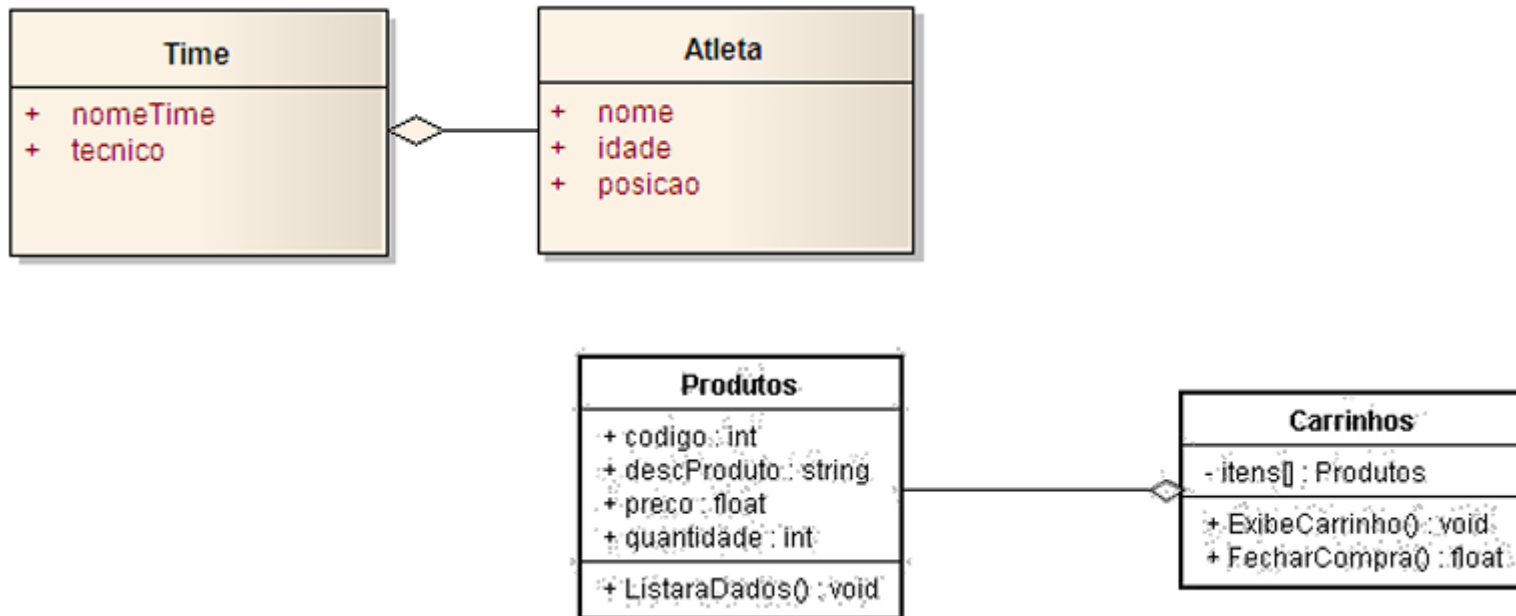


Multiplicidade	Significado
0..1	No mínimo zero e no máximo um. Os objetos não precisam estar relacionados, porém se houver relacionamento deve ser de no máximo 1.
1..1	Um e somente um
0..*	No mínimo nenhum e no máximo muitos.
*	Muitos
1..*	No mínimo um e no máximo muitos.
3..5	No mínimo 3 e no máximo 5.

# Agregação

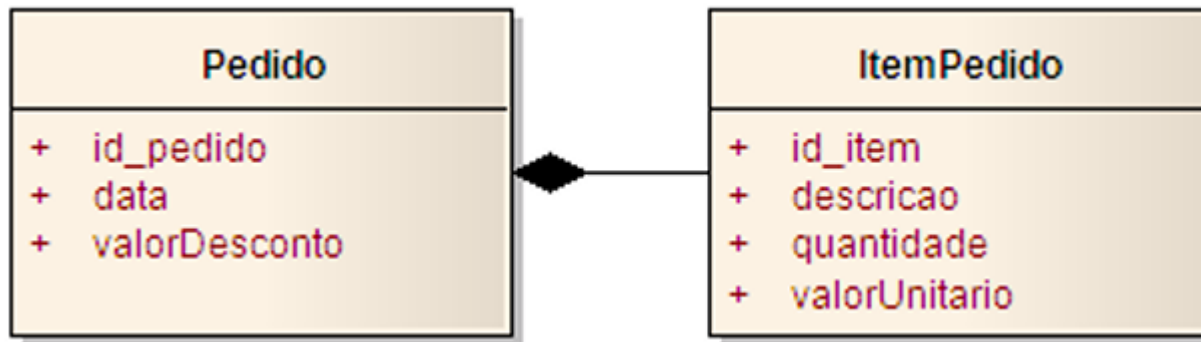


- Informações de um objeto precisam ser complementadas pelas informações contidas em outro objeto.
- A existência do **objeto-parte** faz sentido mesmo se não existir o objeto-todo.



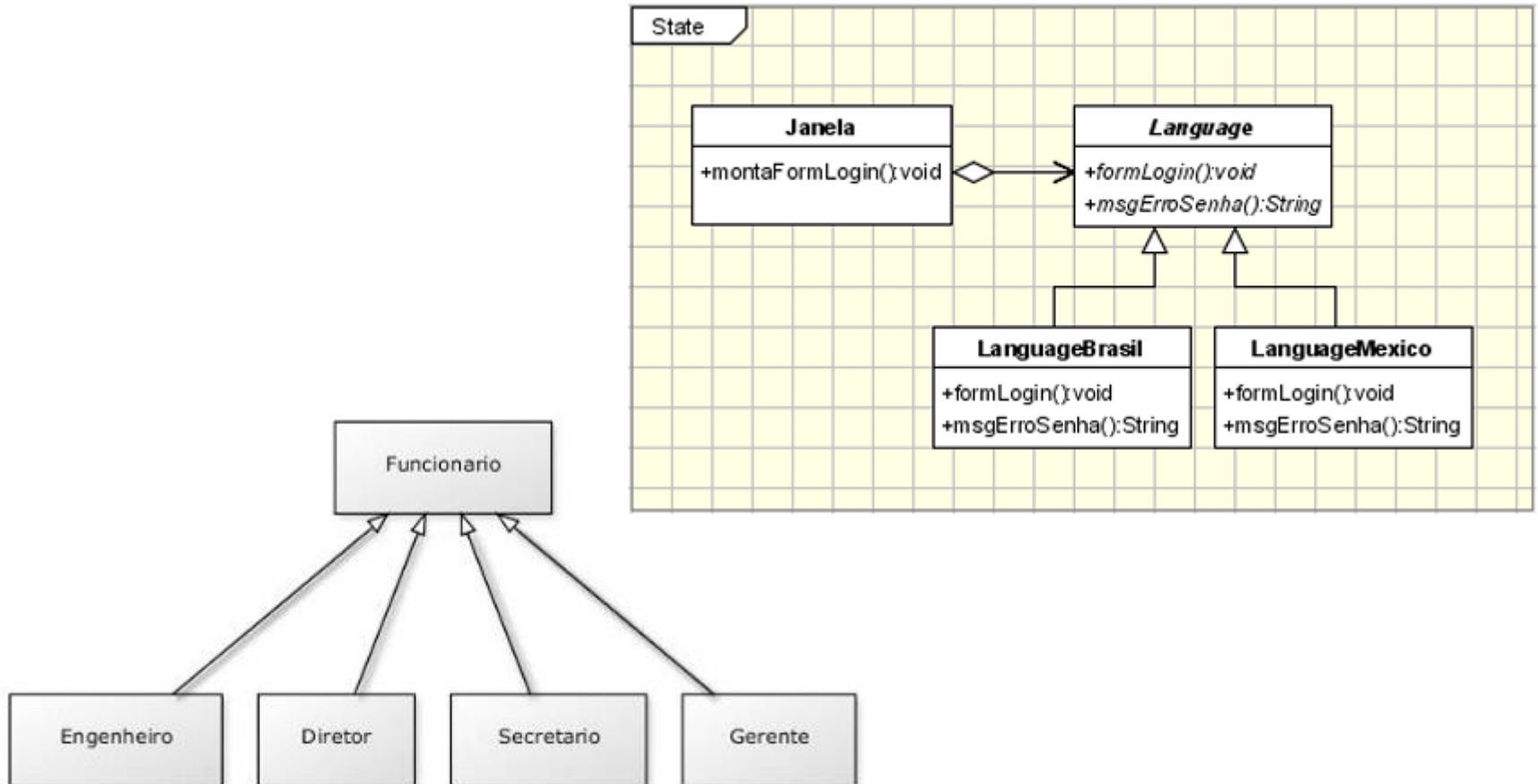
# Composição

- Variante de agregação considerada mais forte.
- A existência do **objeto-parte NÃO faz sentido se não existir o objeto-todo.**
- Se o objeto-todo for destruído, o objeto-parte também será.



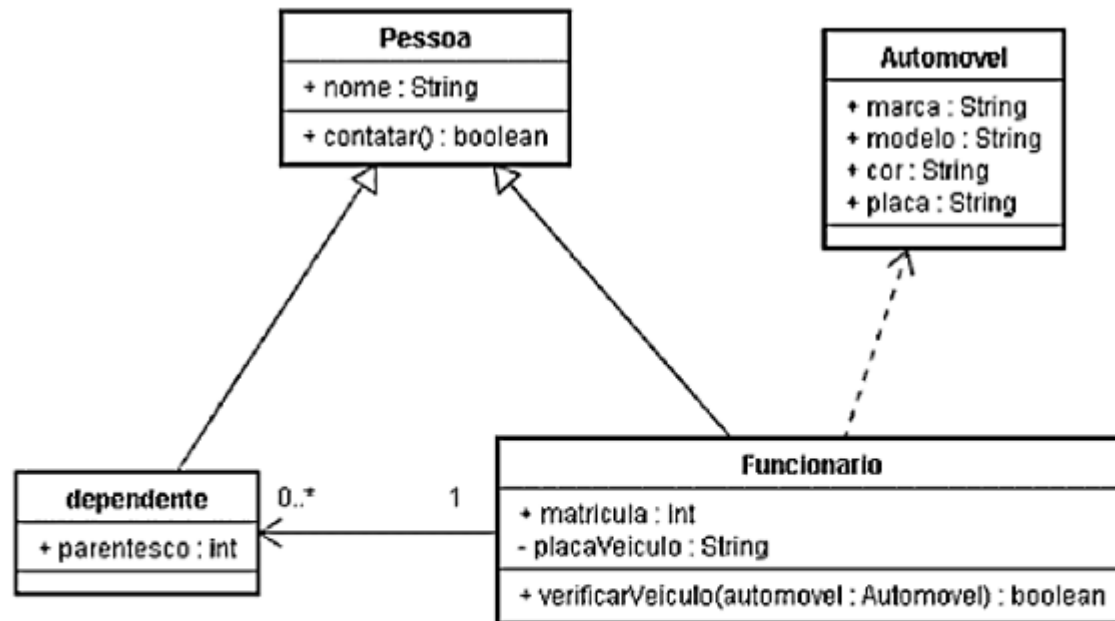
# Especialização/Generalização

- Identifica classes-mães (gerais) e classes-filhas (especializadas).
- São chamadas de relacionamentos “é um tipo de”.



# Dependência

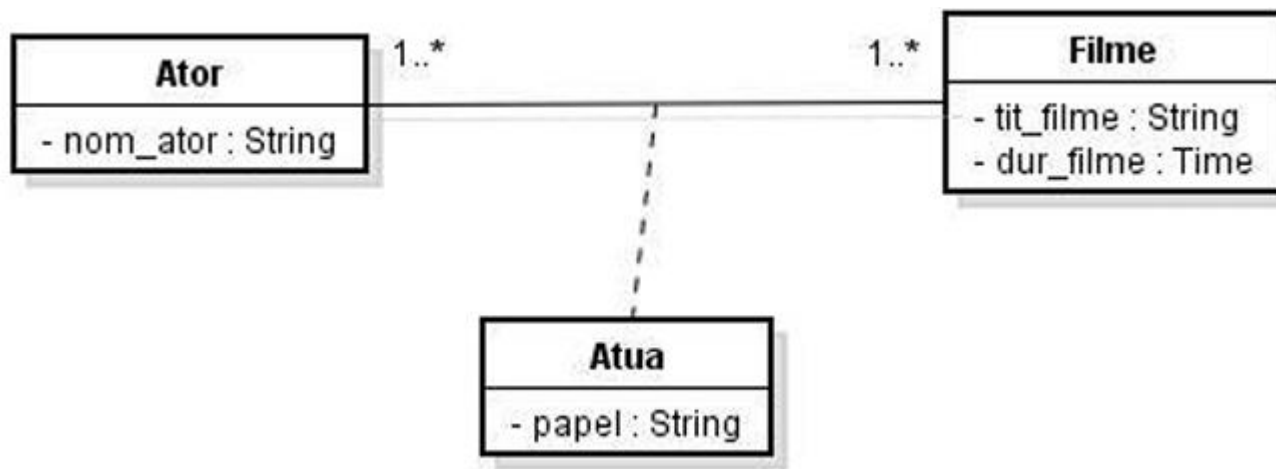
- Indica um grau de dependência entre 2 classes.
- Difere de associação porque a conexão é temporária.
- Seta tracejada.



Funcionário não instancia um Automóvel, apenas usa-o como parâmetro de um método.

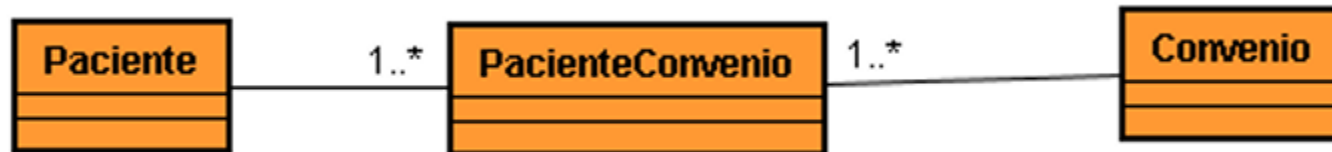
# Classe associativa

- Usada quando temos associações muito para muitos em ambas as extremidades.
- Seta tracejada partindo do meio da associação até a classe associativa.
- Armazena os atributos da associação.
- Pode possuir seus próprios atributos.



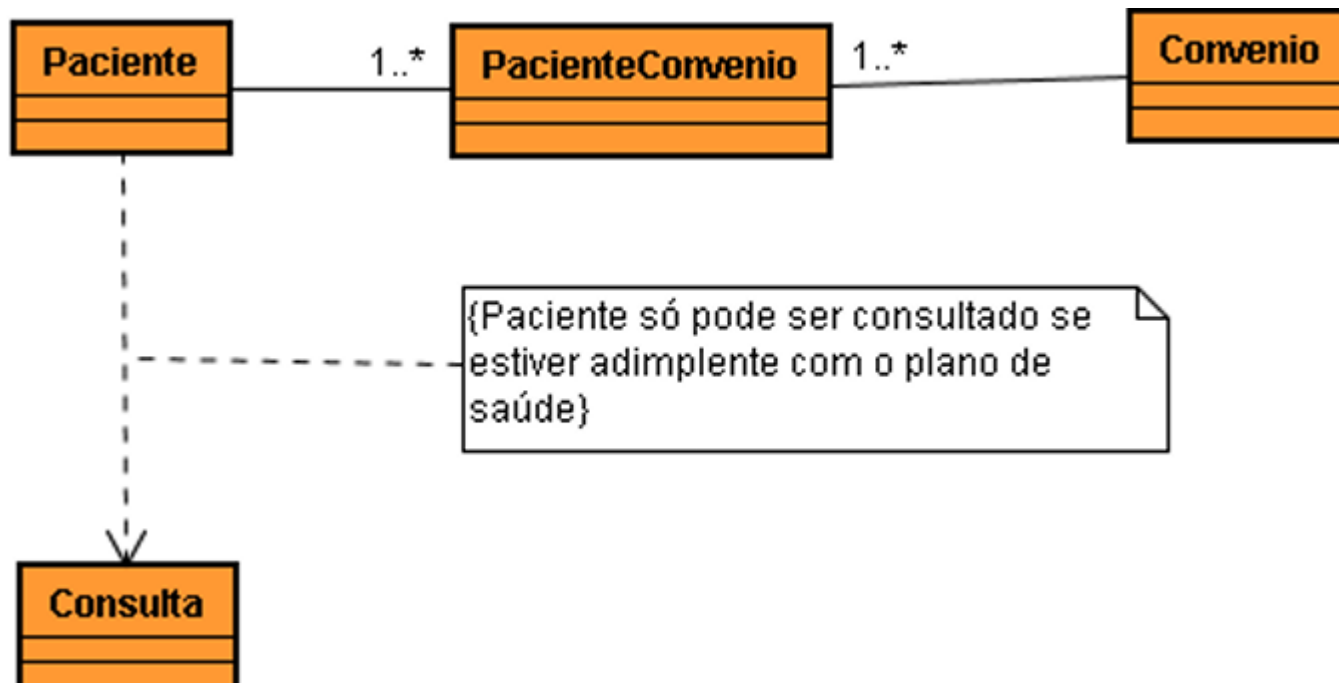
# Classe intermediária

- Substitui as classes associativas.
- Tem a mesma função da classe associativa.
- Possui seus próprios atributos.



# Restrição

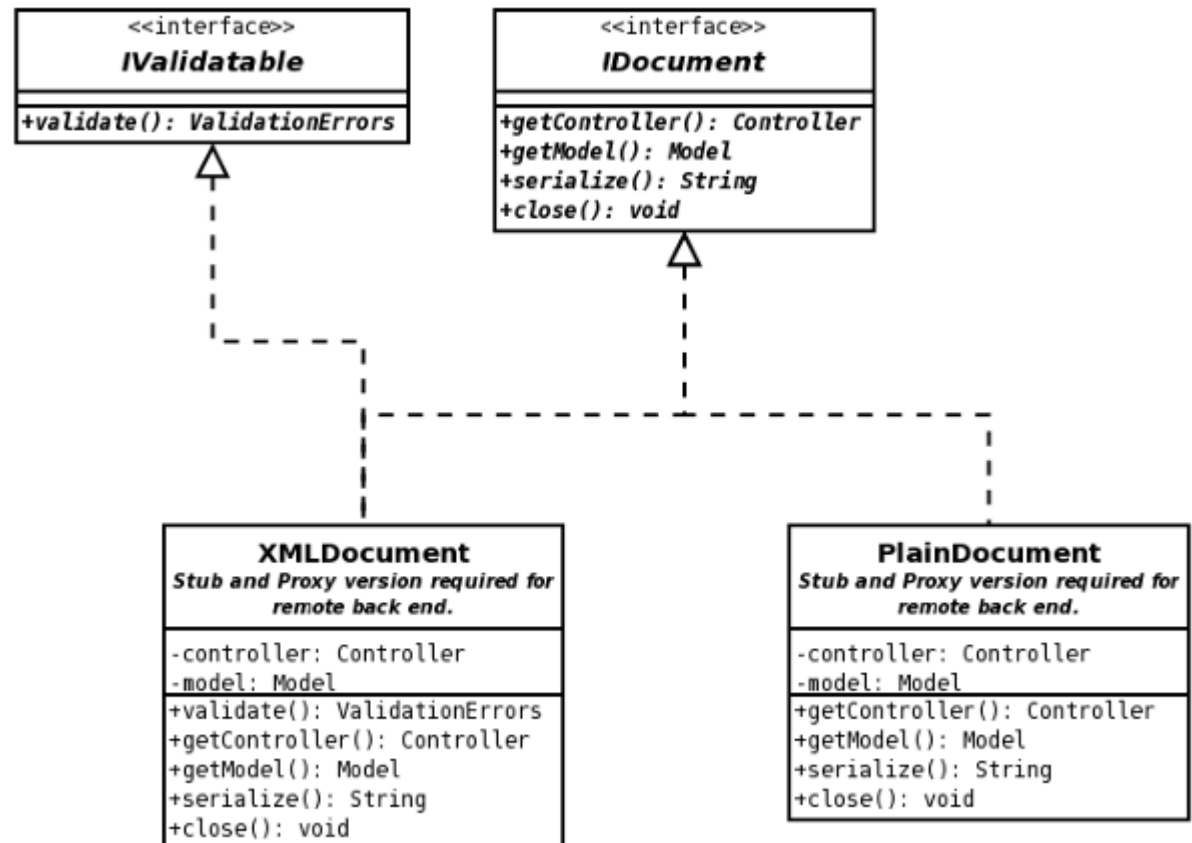
- Informações extras que definem validação durante a implementação dos métodos.
- Representado por texto entre chaves.



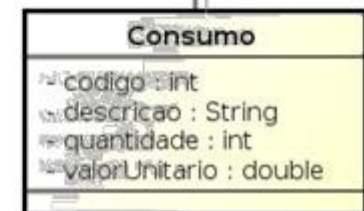
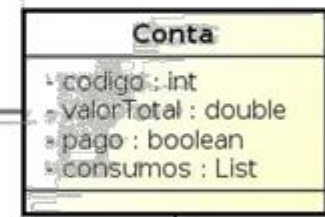
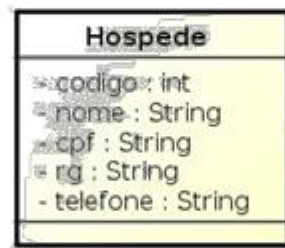
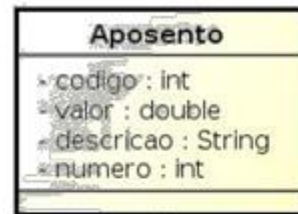


# Interface

- Tipo especial de classe que não pode ser instanciada.
- Especifica operações para outra classe implementar.



# Exemplos de Diagrama de Classes





# Referências



SIERRA, Katy; BATES, Bert. **Use a cabeça JAVA**. Ed 2, Editora Altabooks.

GUEDES, Gilleanes. **UML Uma Abordagem Prática**. Editora Novatec. São Paulo, 2007.

FURLAN, José. **Modelagem de Objetos através da UML**. Editora Makron Books.

CASTRO, Maurício. **Orientação a Objetos**. Solis/Univates (internet).

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Editora Campus.

LIMA, Gleydson. **Diagrama de Classes**. Curso de Especialização em Sistemas Corporativos, FARN/2008.

MACEDO, José Alexandre. **Modelando objetos com cores**. Disponível em: <http://jamacedo.com/tag/uml-em-cores/>. Acessado em: 16 mai. 2011.

MENDES, Ricardo. **UML: composição x agregação**. Disponível em: [http://imasters.com.br/artigo/18901/uml/uml\\_composicao\\_x\\_agregacao/](http://imasters.com.br/artigo/18901/uml/uml_composicao_x_agregacao/). Acessado em: 26 mai. 2011

TONSIG, Sérgio Luiz. **Engenharia de Software: Análise e Projeto de Sistemas**. 2ª edição. Rio de Janeiro: Editora Ciência Moderna, 2008.

# Tarefa prática

- Apresentar dois exemplos de cada um dos tipos de relacionamentos apresentados.
- Pode ser só o exemplo e ou um diagrama de classes maior e você identificar neste diagrama os tipos de relacionamentos (aqui pode ser 1 de cada).

