



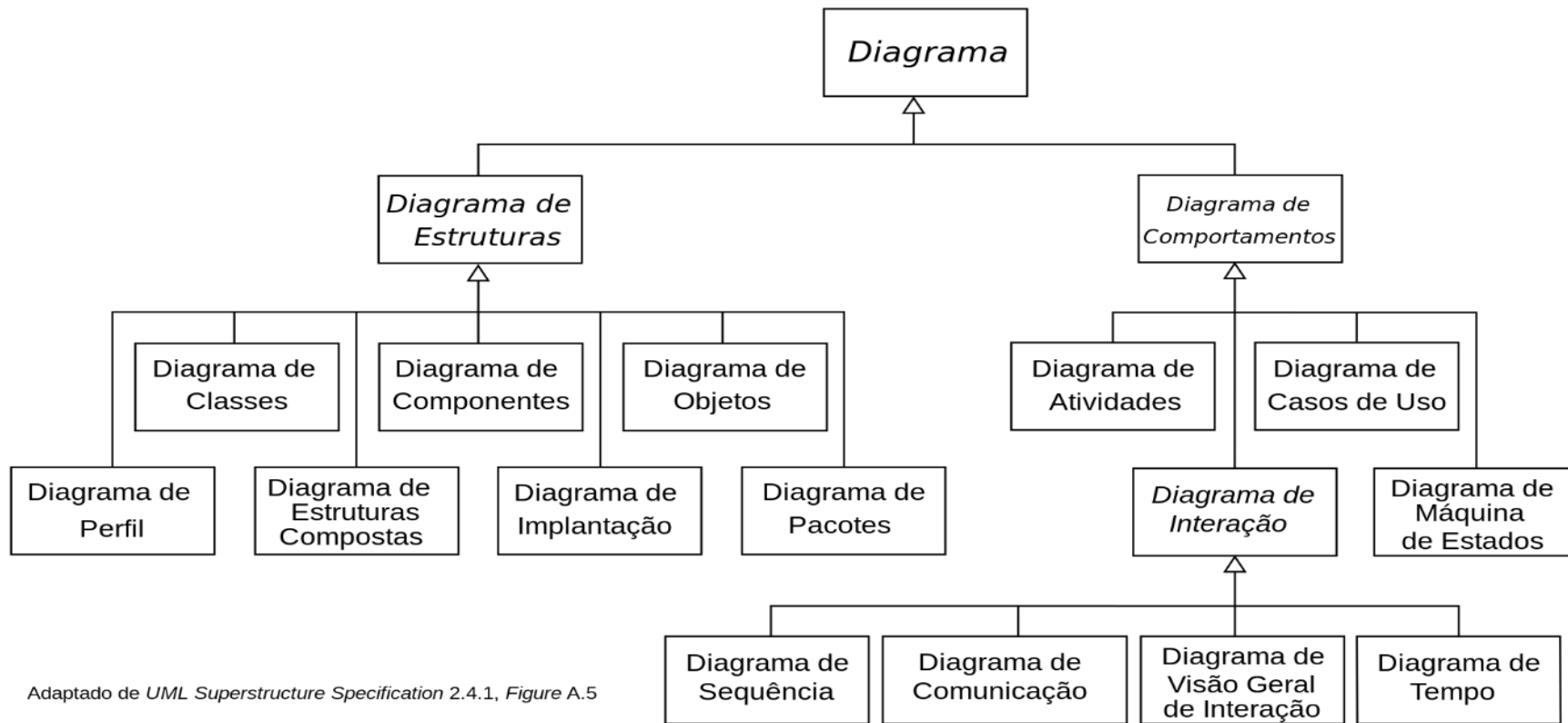
Quem se prepara, não para.

Análise de Sistemas

3º período

Professora: Michelle Hanne

Modelagem de Sistemas em UML



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

Diagrama de Atividades

- **Estado Inicial e Final**: Todos os diagramas de atividades possuem pelo menos um estado inicial e pelo menos um estado final.



- **Atividades**: Retângulos com bordas arredondadas representam as atividades



Diagrama de Atividades

- **Transições**: Setas contínuas que representam fluxo de trabalho de uma atividade para outra.



- **Decisões**: Losango utilizado para controlar os desvios do fluxo de controle.

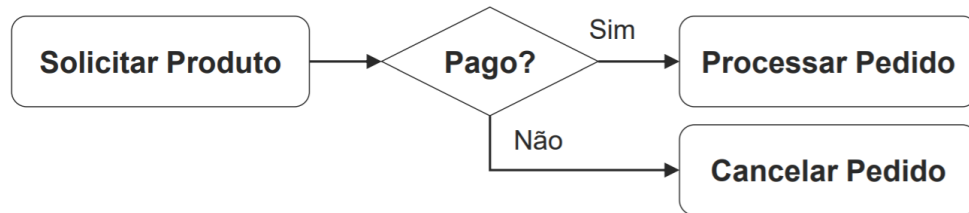


Diagrama de Atividades

- **Bifurcação e União:** barra sólida usada para atividades paralelas (concorrentes) Bifurcação: Divisão do fluxo de controle União: sincronização das atividades

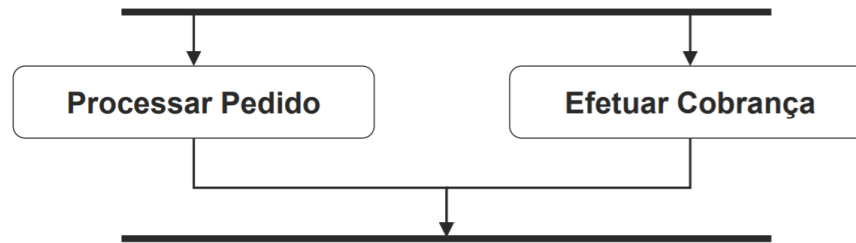


Diagrama de Atividades

- **Raias**: são uma forma de organização lógica das atividades, podem estar associadas a objetos, componentes do sistema ou a atores.

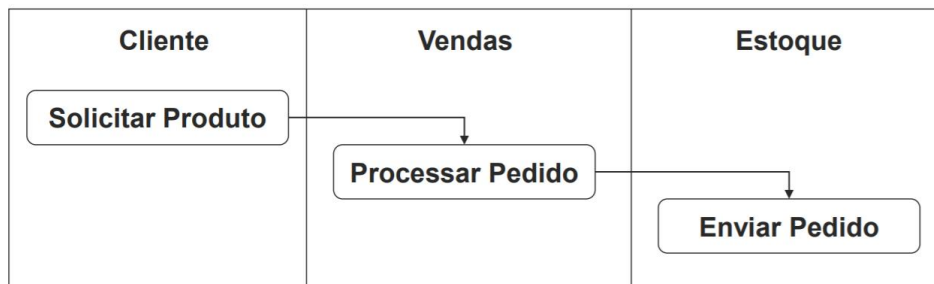


Diagrama de Atividades

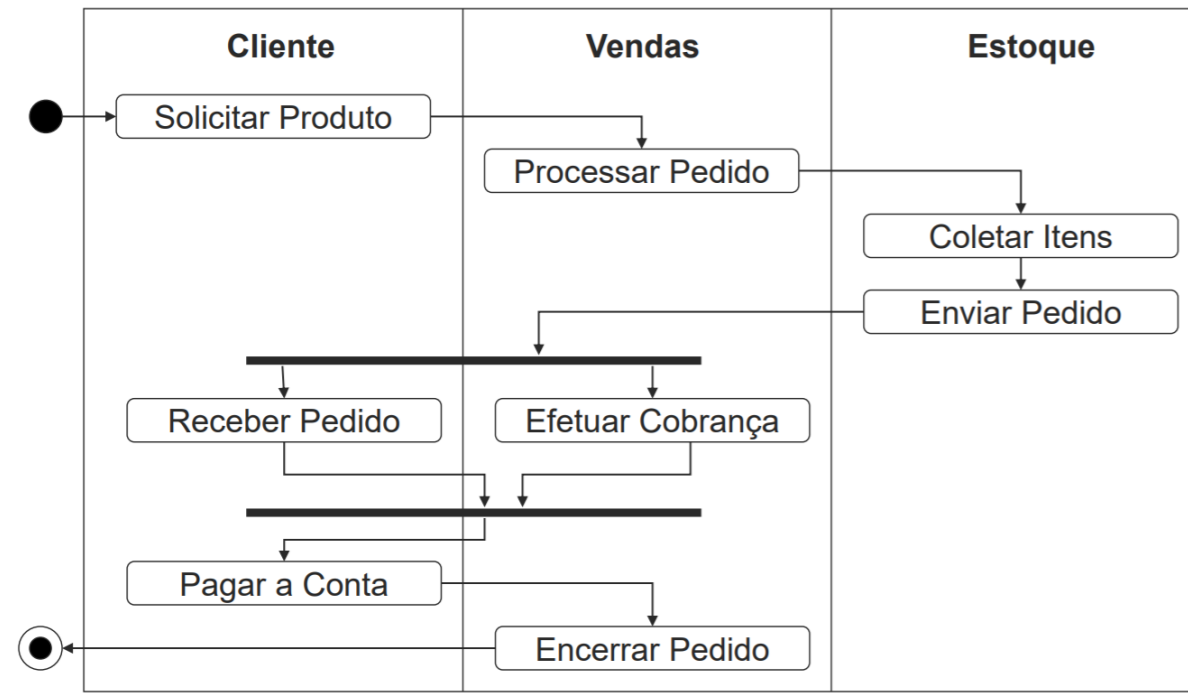
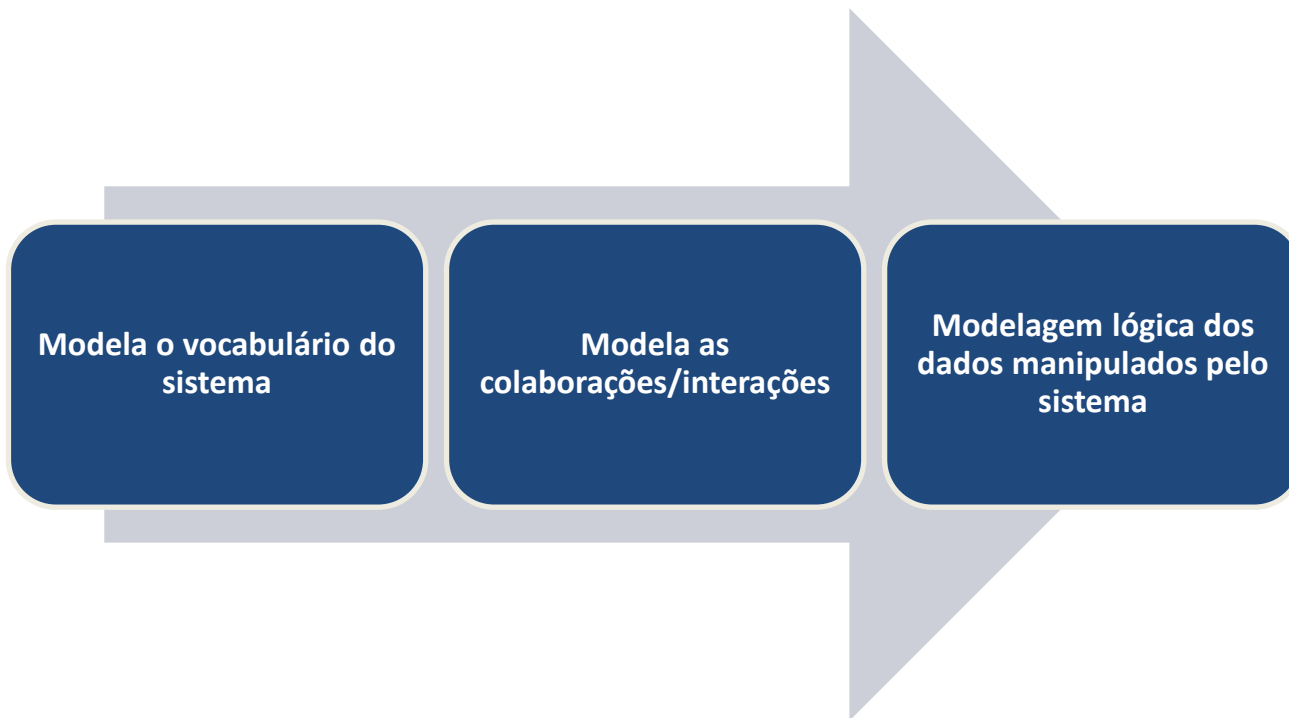


Diagrama de Classe

- Mostra a **estrutura do sistema**, subsistema ou componente projetado como **classes e interfaces** relacionadas, com seus **recursos, restrições e relacionamentos** - associações, generalizações, dependências, etc. (Sommerville, 2011).

Diagrama de Classe



Notação da Classe

Nome da Classe
Atributos
Operações

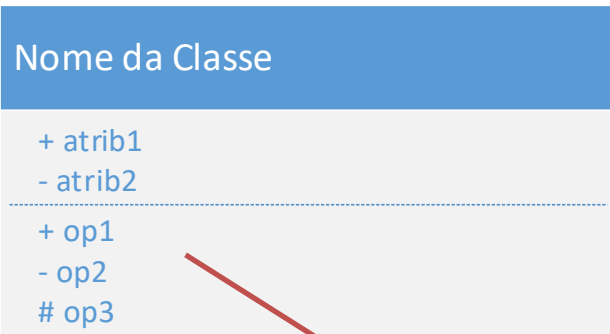
Nome da Classe
+ atrib1 - atrib2
+ op1 - op2 # op3

Consulta
Médicos Data Horário Clínica Motivo Medicação prescrita Tratamento prescrito Anotações de voz Transcrições
Novo() Preescrever() RegistrarAnotações() Transcrever()

Fonte: Sommerville (2011), p. 91.

Classe é uma abstração de um conjunto de objetos com características similares.

Notação da Classe



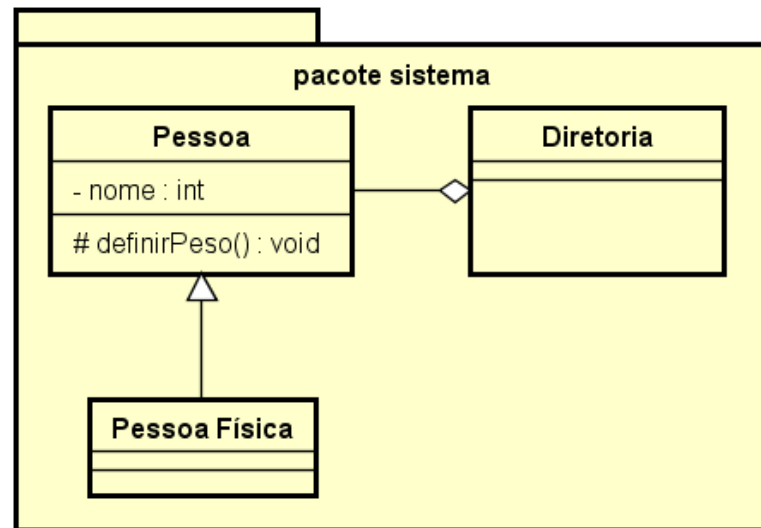
Indica o nível de acessibilidade de um atributo ou método:

- + Público**
- Privado**
- # Protegido**

Notação da Classe

- **(private)**: Atributos e métodos declarados como *private* são acessíveis somente pela classe que os declara. Métodos e atributos com o modificador *private* **não são herdados**.

(Protected): Atributos e métodos declarados como *protected* são acessíveis pela classe que os declara, suas subclasses em outros pacotes e outras classes dentro do mesmo pacote.



powered by Astah 

Neste exemplo o método `definirPeso()` é visível na subclasse “Pessoa Física” e na classe “Diretoria” que está no mesmo pacote.

Notação da Classe

+ (Public): Atributos, métodos e classes declarados como public são acessíveis por qualquer classe do Java. Todos os métodos e atributos declarados como public são herdados pelas subclasses. Métodos e atributos declarados como public devem se manter public em todas as subclasses.

~ (Default): Modificador de acesso padrão, usado quando nenhum for definido. Neste caso os atributos, métodos e classes são visíveis por todas as classes dentro do mesmo pacote.

Atributo

- Permite a identificação de cada objeto de uma classe.
- Os valores dos atributos podem variar de instância para instância.
- Atributos devem conter o tipo de dados a ser armazenado: Byte, boolean, int, double, char, String, etc

Método

- São apenas declarados, não define a implementação.
- Outros diagramas permitem modelar o comportamento interno dos métodos: Diagrama de Sequência e Diagrama de Atividades.

Método

- Uma classe é representada por um retângulo com três divisões:
 - Nome da Classe
 - Atributos da Classe
 - Métodos da Classe

Pessoa
nome email
enviarMensagem()

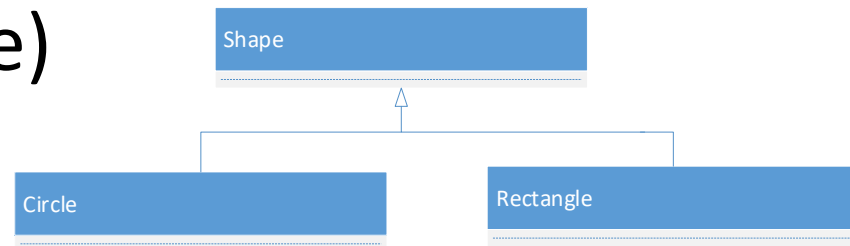
Relacionamentos

São conexões entre classes:

1. Dependência - uma classe usa a outra.



2. Generalização - geral (superclasse) e uma coisa mais específica (subclasse)



3. **Associação** - classes ou objetos estão interconectados.

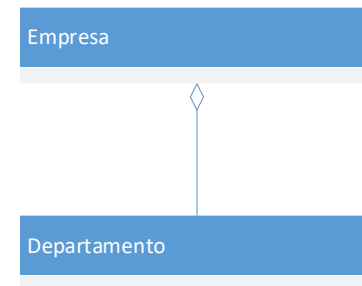


Ornamentos para Associações

- **Multiplicidade**

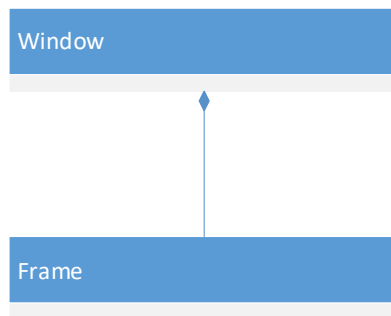


- **Agregação:** É uma relação do tipo “todo/parte” ou “possui um”



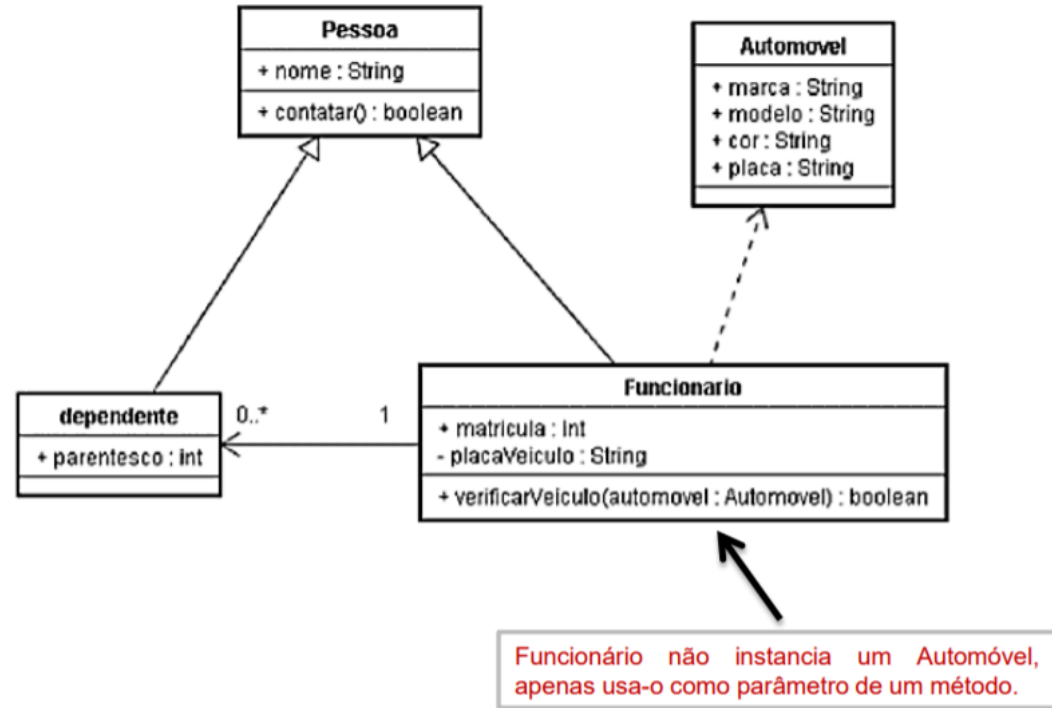
Ornamentos para Associações

- **Composição:** Um tipo de agregação na qual as partes são inseparáveis do todo.



Dependência

- Uma dependência difere de uma associação porque a conexão entre as classes é temporária.



Exemplo de Diagrama de Classe

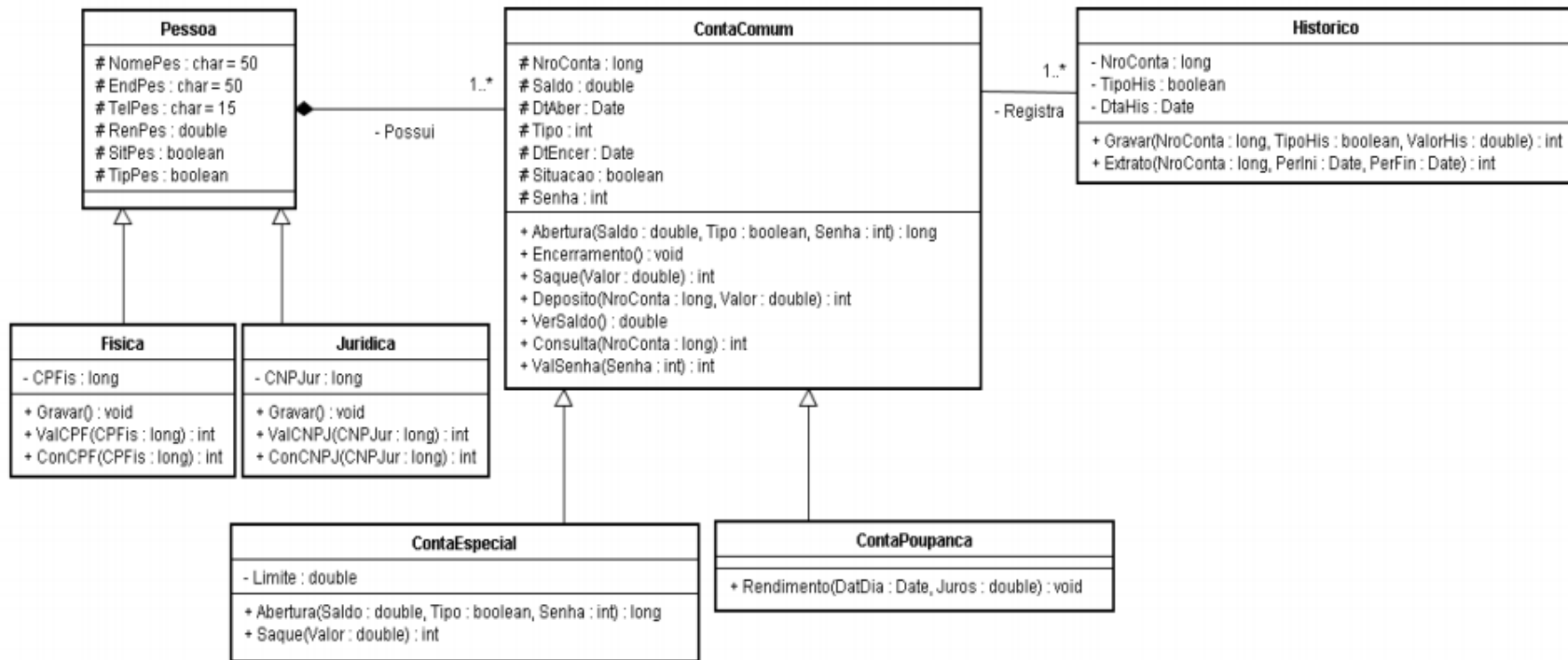
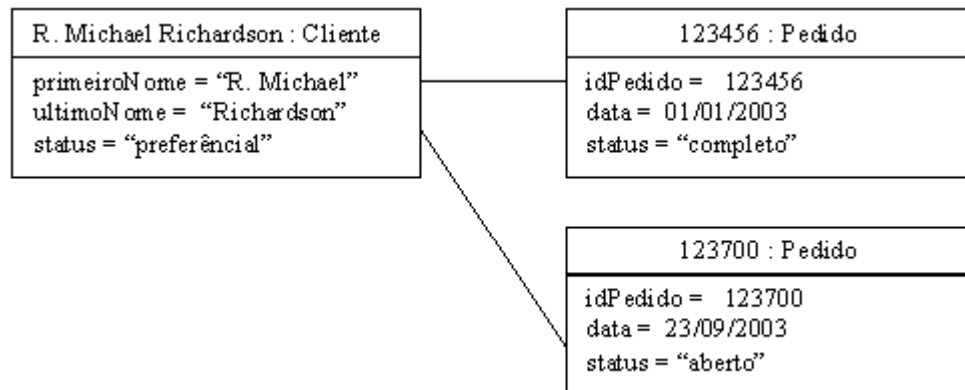


Diagrama de Objeto

Os diagramas de objetos fornecem uma captura instantânea das instâncias em um sistema e os relacionamentos entre as instâncias.

Os diagramas de objetos usam notação semelhante à usada nos diagramas de classe. No entanto, enquanto os diagramas de classe mostram os classificadores reais e seus relacionamentos em um sistema, os diagramas de objetos mostram instâncias específicas desses classificadores e os links entre essas instâncias em um determinado momento.

Diagrama de Objeto



O objeto R. Michael Richardson da classe Cliente está associado a ambos os objetos 123456 e 123700 da classe Pedido.

Usa-se o diagrama de objetos para modelar a visão estática de um sistema.

Referências

- OMG, UnifiedModelingLanguage2.5.1, 2017. Disponível em
<<https://www.omg.org/spec/UML/2.5.1/PDF>>. Acesso em 04 de fevereiro de
2020
- PRESSMAN, Roger S. MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem
Profissional**. 8.ed. Porto Alegre: Amgh Editora, 2016. 968p. ISBN 9788580555332.
- SOMMERVILLE, Ian. **Engenharia de software**. 8.ed. São Paulo: A. Wesley publishing
company, 2010.

.