

Engenharia de Software Moderna

Cap. 4 - Modelos

Prof. Marco Tulio Valente

<https://engsoftmoderna.info>, @engsoftmoderna

Licença CC-BY; permite copiar, distribuir, adaptar etc; porém, **créditos devem ser dados ao autor dos slides**

Motivação

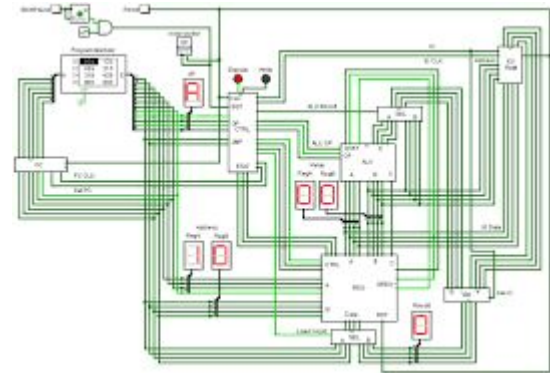
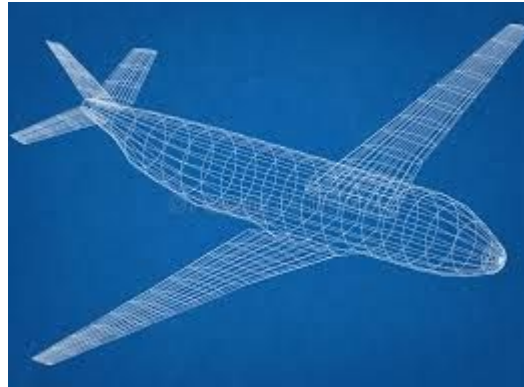
- Existe uma lacuna entre os seguintes mundos:
 - Requisitos: "o que" o sistema faz; possuem um nível de abstração alto
 - Código: "como" o sistema opera; possui nível de abstração baixo

Modelos de Software

- Objetivo: preencher essa "lacuna"
- Via uma notação com um nível de abstração intermediário
- Ajudar a conceber, especificar, entender e documentar uma solução para o problema delimitado pelos requisitos

Comuns em outras Engenharias

- Natural que fossem propostos também para software



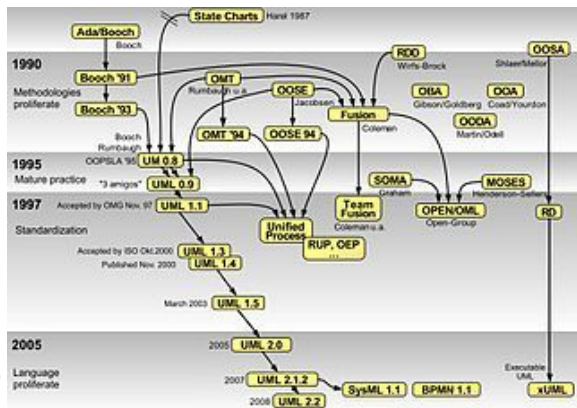
Modelos de Software

- Infelizmente, não são tão efetivos e largamente usados, como em outras engenharias
- Modelos de software podem ser:
 - Formais: menos comuns; não serão estudados aqui
 - Gráficos: UML é a notação mais comum

UML: Unified Modelling Language



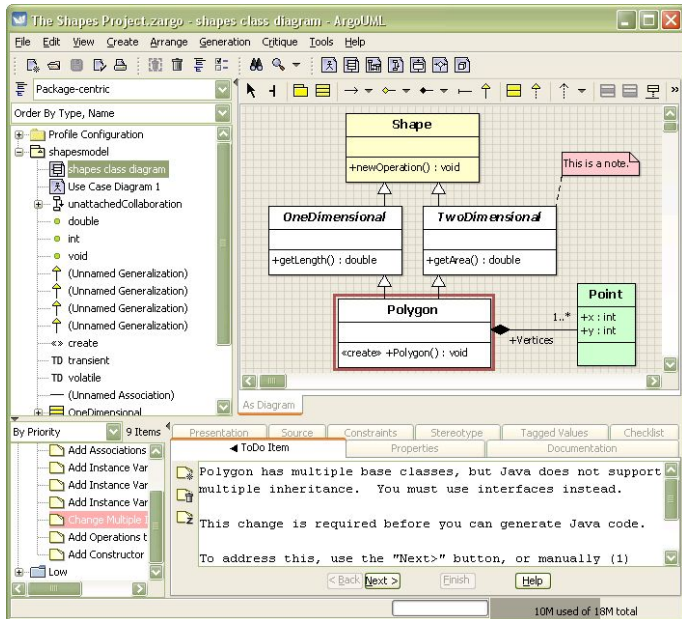
- Proposta em 1995, para fundir outras notações
- Processo mais comum na época: RUP
 - Documentação e planejamento detalhados
 - Código era escrito após meses ou anos de trabalho



Fonte: Wikipedia

Ferramentas CASE

- CASE: Computer-Aided Software Engineering
- Equivalente a ferramentas CAD, mas para Eng. Software



Como usar UML?

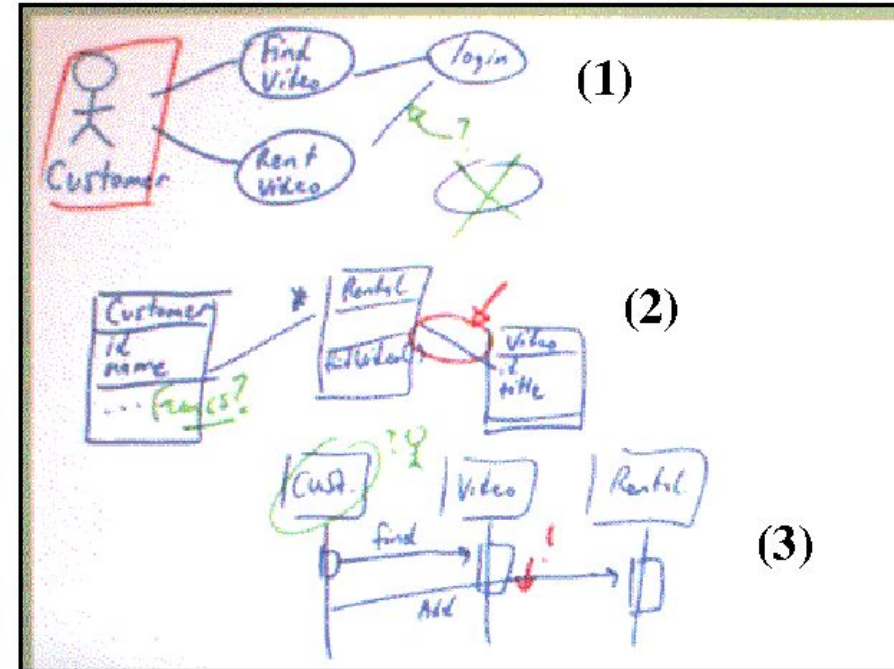
1. Blueprint (planta detalhada)
2. Linguagem de programação (geração automática código)
3. Sketches (esboços, rascunhos)

Neste curso, vamos estudar o uso de UML
como sketches

UML como Sketch

- Uso mais comum de UML com métodos ágeis
- UML é usada para:
 - Conversar sobre uma parte do código ou do projeto
 - Documentar uma parte do código ou do projeto
- Uso mais informal e leve da notação
- Objetivo **não** é ter um modelo completo

UML como Sketch



Engenharia Avante vs Engenharia Reversa

- Engenharia Avante ("Forward")
 - Modelo é usado para discutir alternativas de projeto, antes de ter qualquer linha de código implementada
- Engenharia Reversa
 - Modelo é usado para explicar um código que já existe
- Sketches são úteis nos dois casos

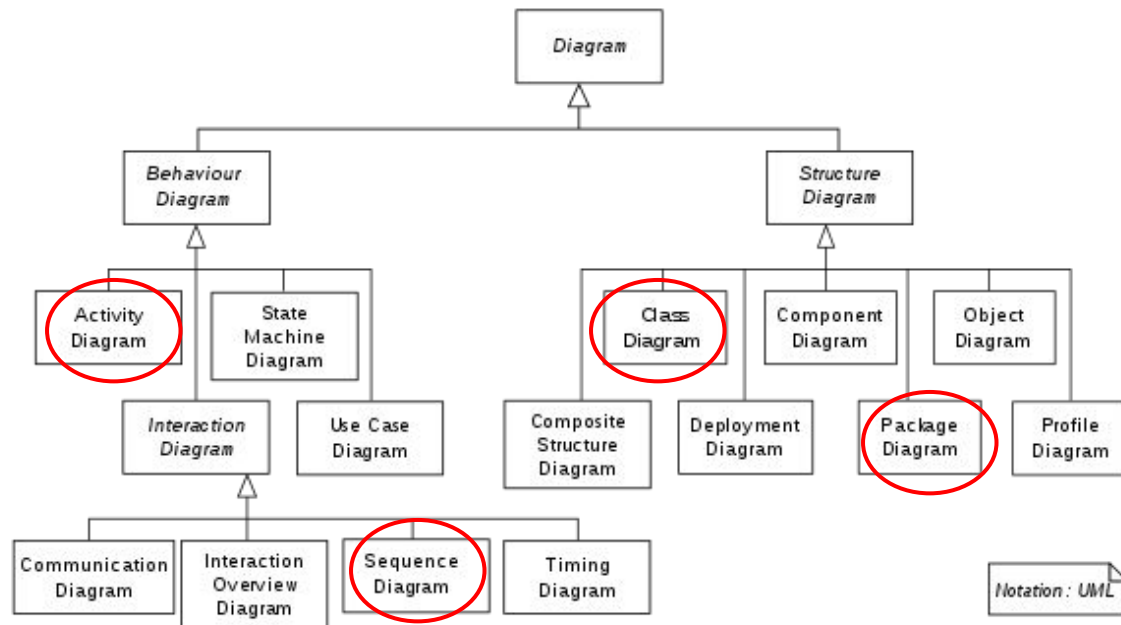
Diagramas UML

Diagramas UML

- UML é uma notação ou linguagem gráfica para modelagem de software
- Um conjunto de diagramas, organizados em 2 grupos:
 - Diagramas Estáticos: modelam a estrutura do código
 - Diagramas Dinâmicos: modelam a execução do código (o comportamento do sistema)

Diagramas UML

Em vermelho, os diagramas que vamos estudar



Versão de UML que iremos usar

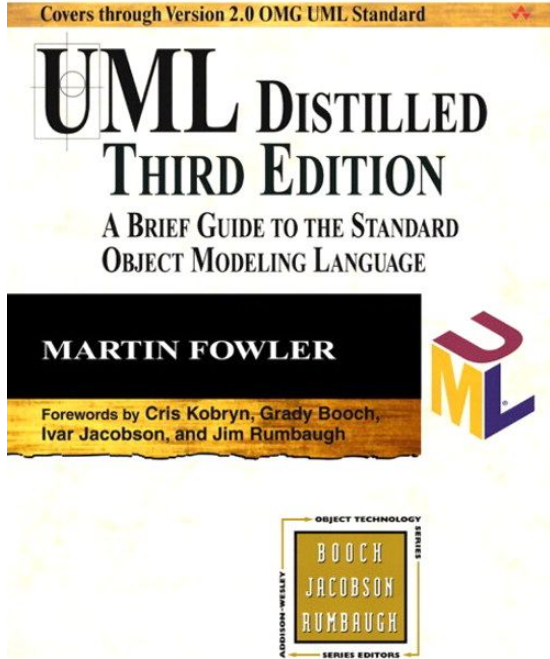


Diagrama de Classes

Formato genérico

[nome da classe]
[atributos]
[métodos]

Exemplo com duas classes

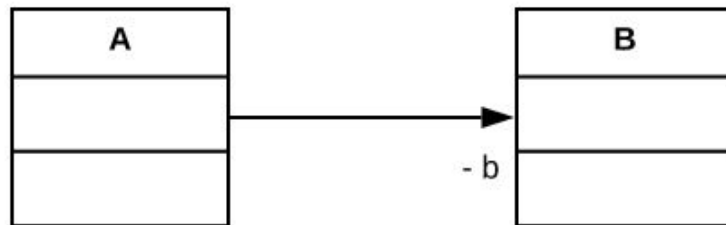
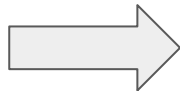
Pessoa
- nome: String - sobrenome: String - fone: Fone
+ setPessoa(nome, sobrenome, fone) + getPessoa(): Pessoa

Fone
- codigo: String - numero: String - celular: Boolean
+ setFone(codigo, numero, celular) + getFone(): String + isCelular(): Boolean

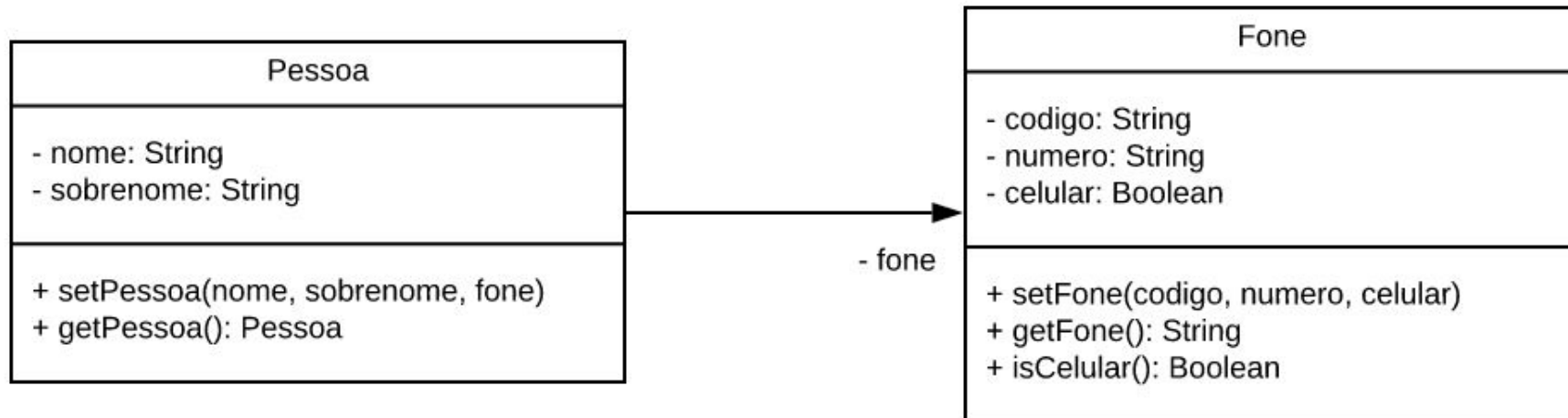
- : private
+: public

Associações

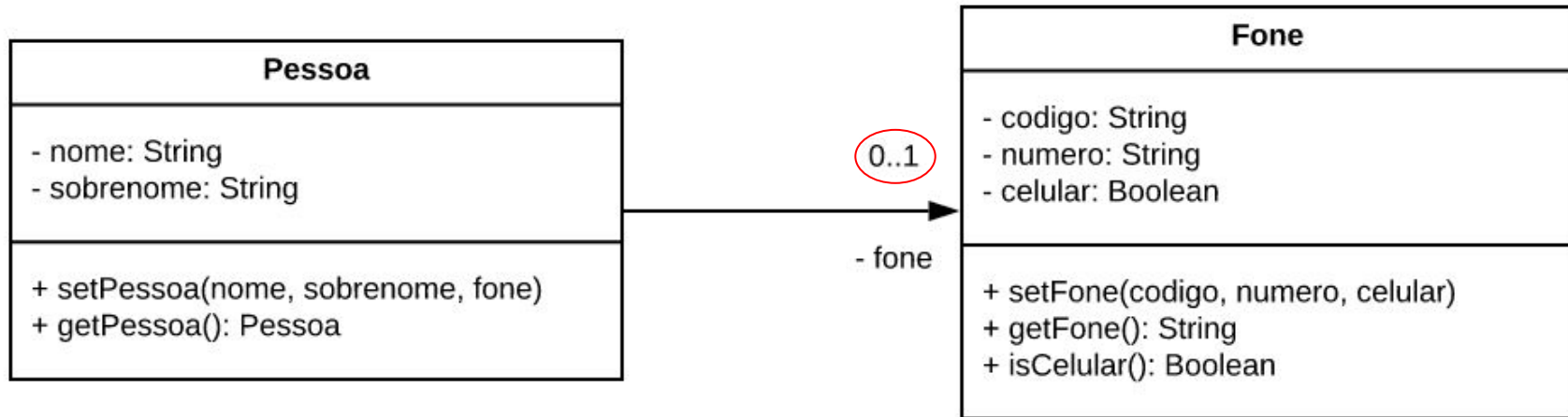
```
class A {  
    ...  
    private B b;  
    ...  
}  
  
class B {  
    ...  
}
```



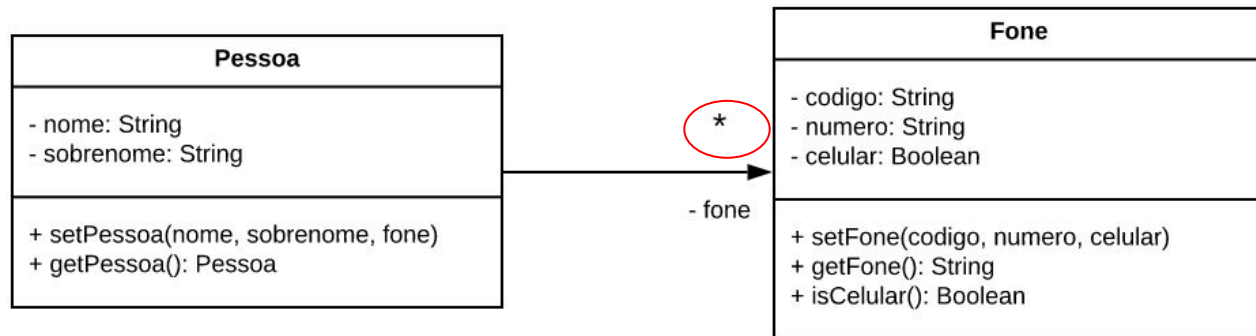
Associações



Multiplicidade (exemplo 1)

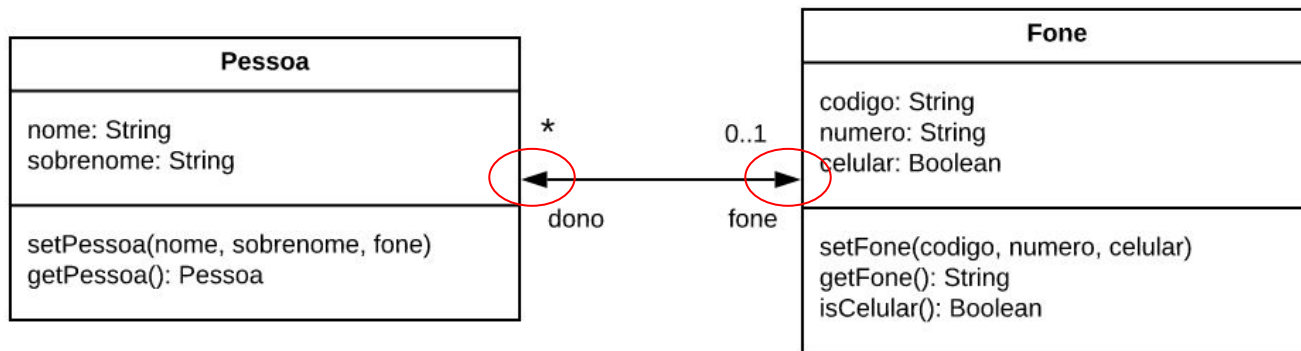


Multiplicidade (exemplo 2)



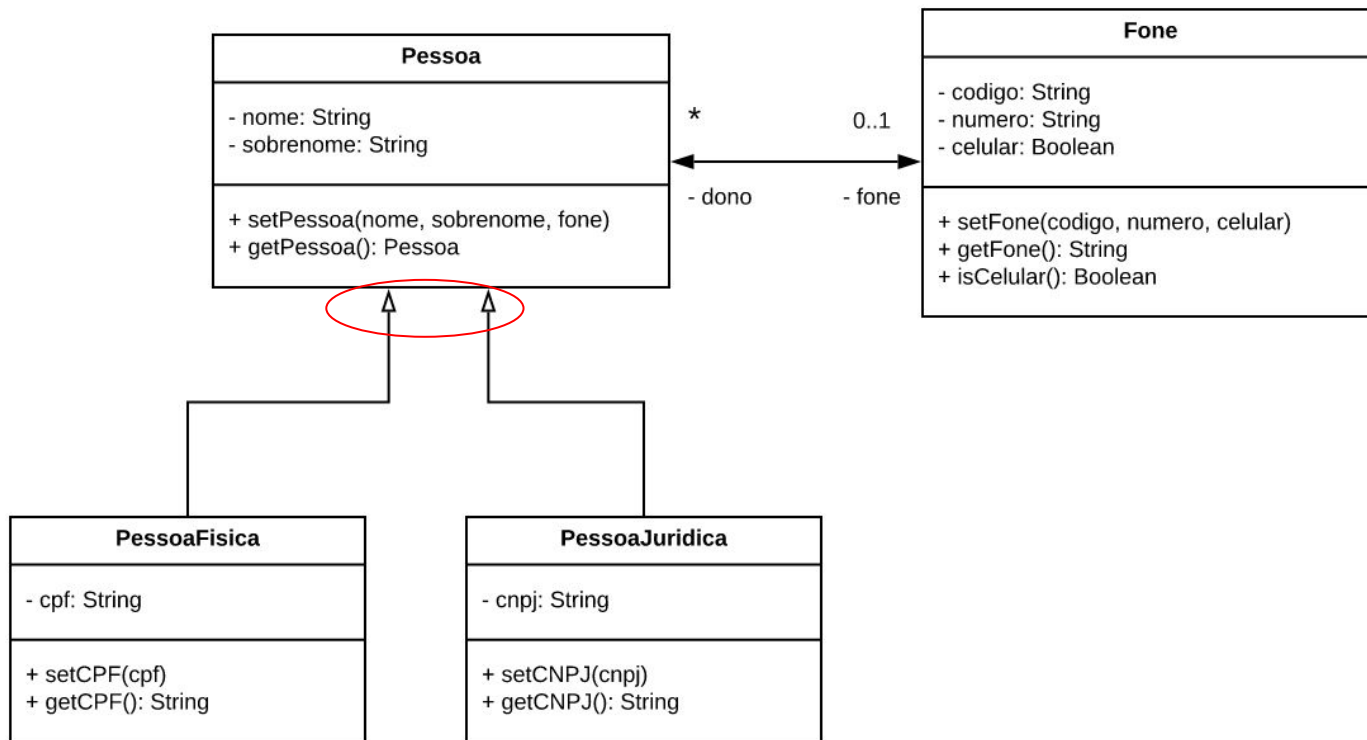
```
class Pessoa {  
    private Fone[] fone;  
    ...  
}  
class Fone {  
    ...  
}
```

Associação bidirecional



```
class Pessoa {  
    ...  
    private Fone fone;  
    ...  
}  
  
class Fone {  
    ...  
    private Pessoa[] dono;  
    ...  
}
```

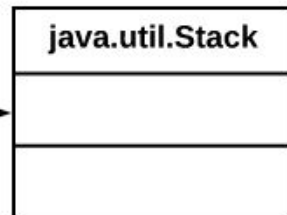
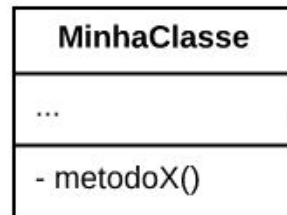

Herança



Dependências (setas tracejadas)

Relacionamento entre duas classes, mas que não é devido a associação ou herança

```
import java.util.Stack;  
  
class MinhaClasse {  
    ...  
    private void metodoX() {  
        Stack stack = new Stack();  
        ...  
    } ...  
}
```



Exemplo de Diagrama de Classes

(um pouco maior e mais completo)

Fonte: Martin Fowler. UML Distilled

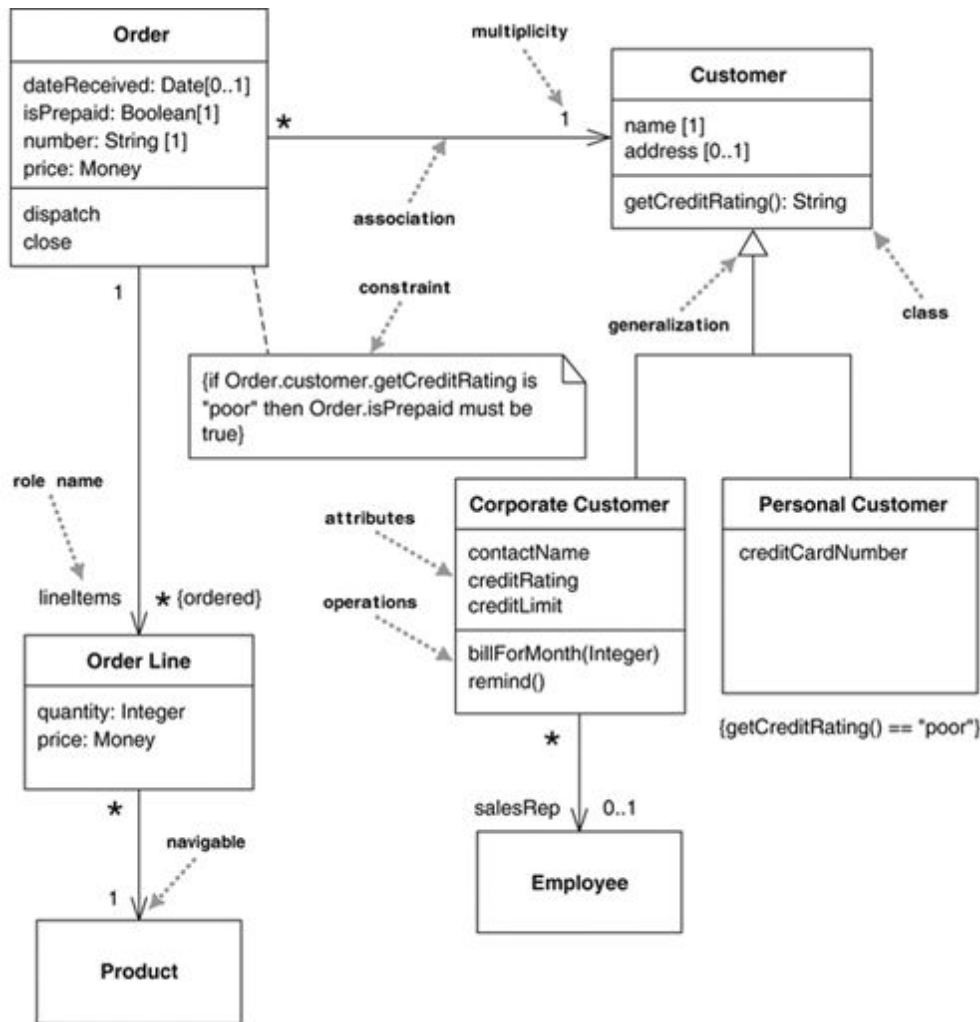


Diagrama de Pacotes

Diagrama de Pacotes



Diagrama de Pacotes

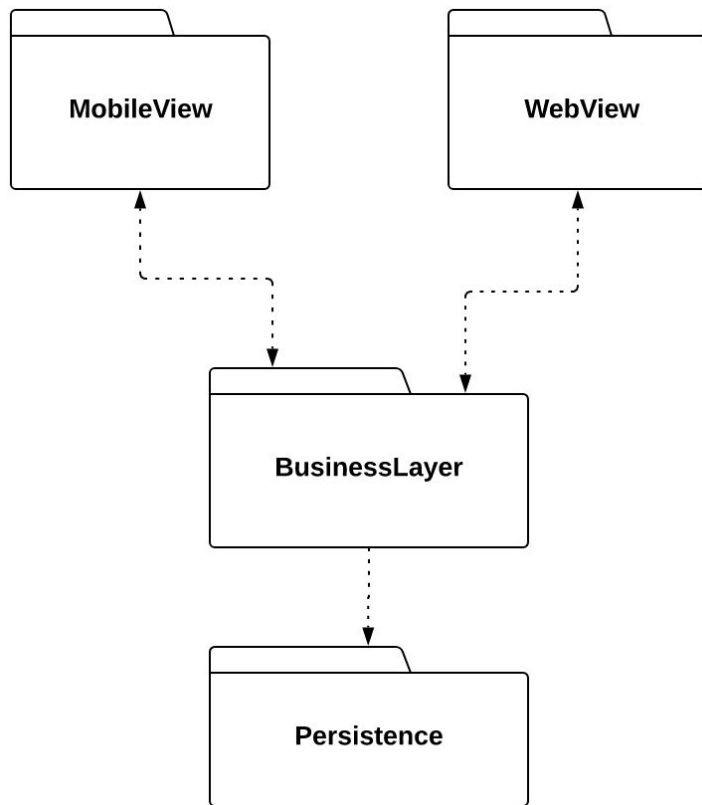


Diagrama de Sequência

Diagramas de Sequência

- São diagramas comportamentais ou dinâmicos
- Modelam:
 - Alguns objetos de um sistema
 - Métodos que eles executam em um determinado contexto

Diagrama de Sequência (exemplo 1)

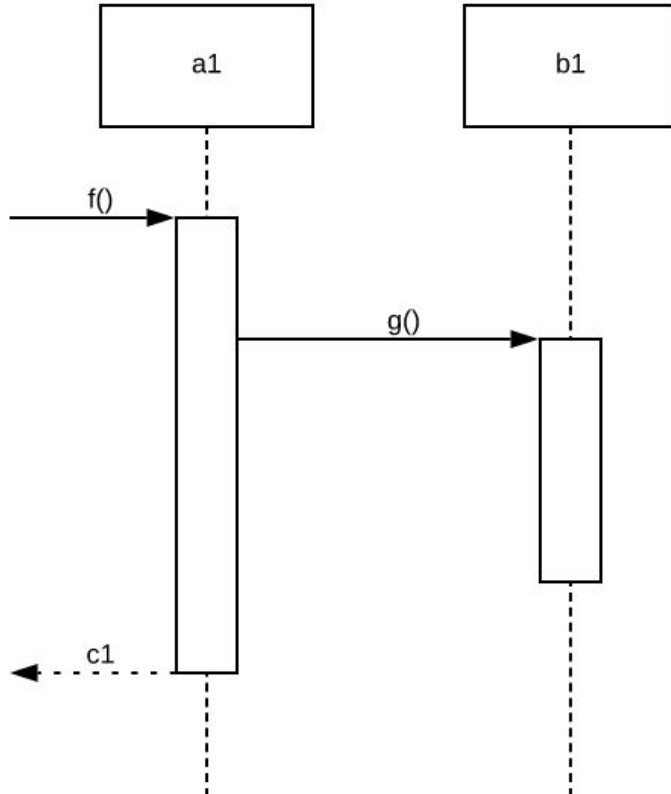


Diagrama de Sequência (exemplo 2)

```
class A {  
    void g() {  
        ...  
    }  
  
    void f() {  
        ...  
        g();  
        ...  
    }  
  
    main() {  
        A a = new A();  
        a.f();  
    }  
}
```

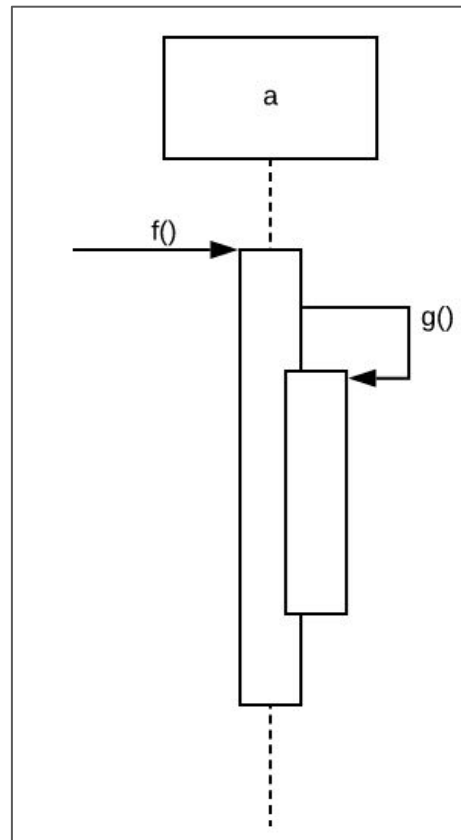


Diagrama de Sequência (exemplo 3)

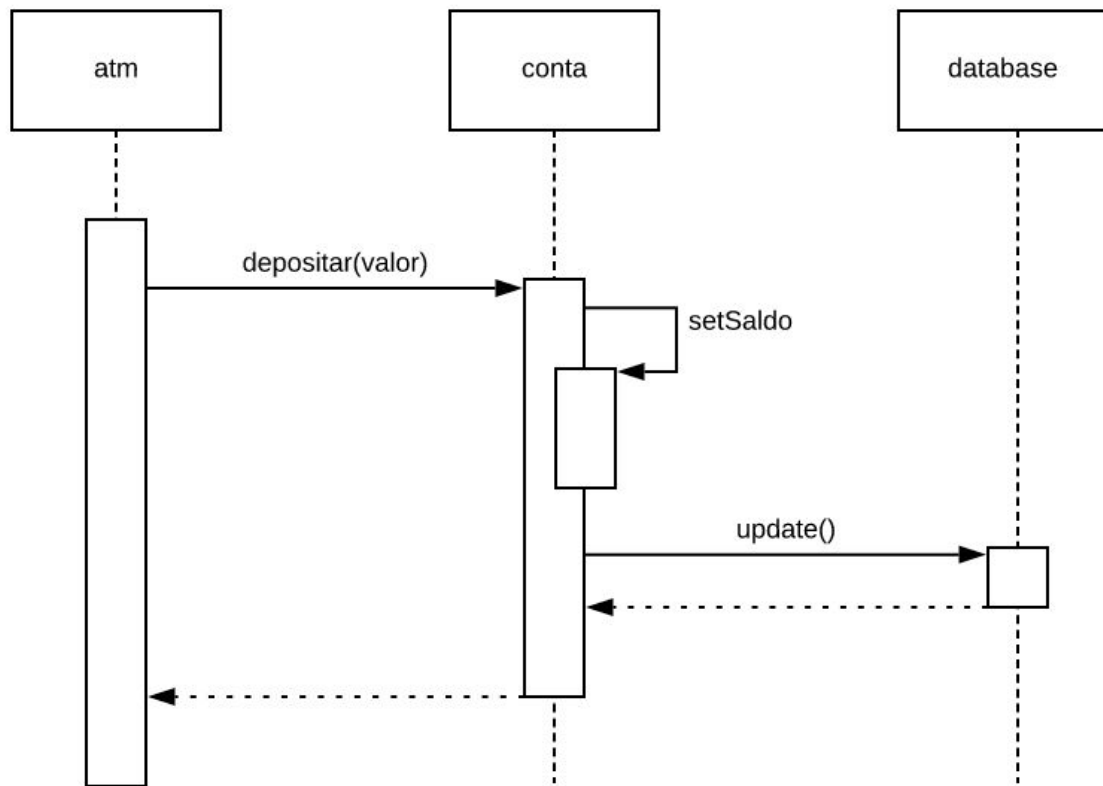
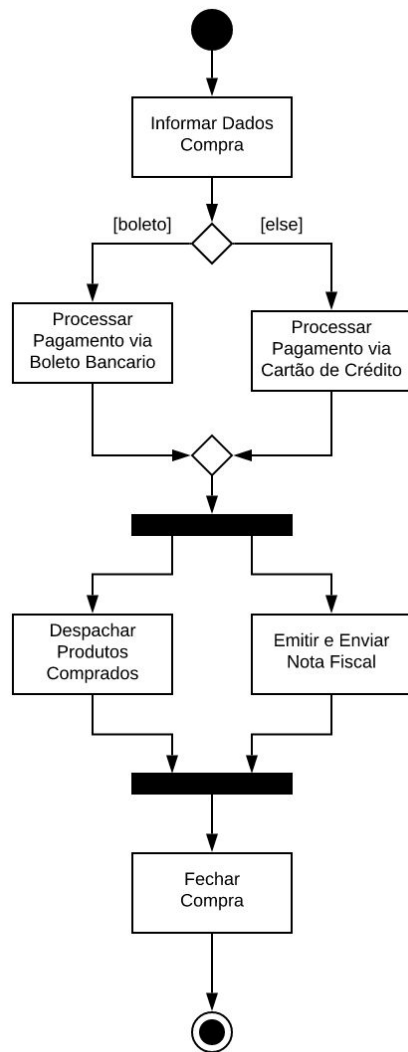


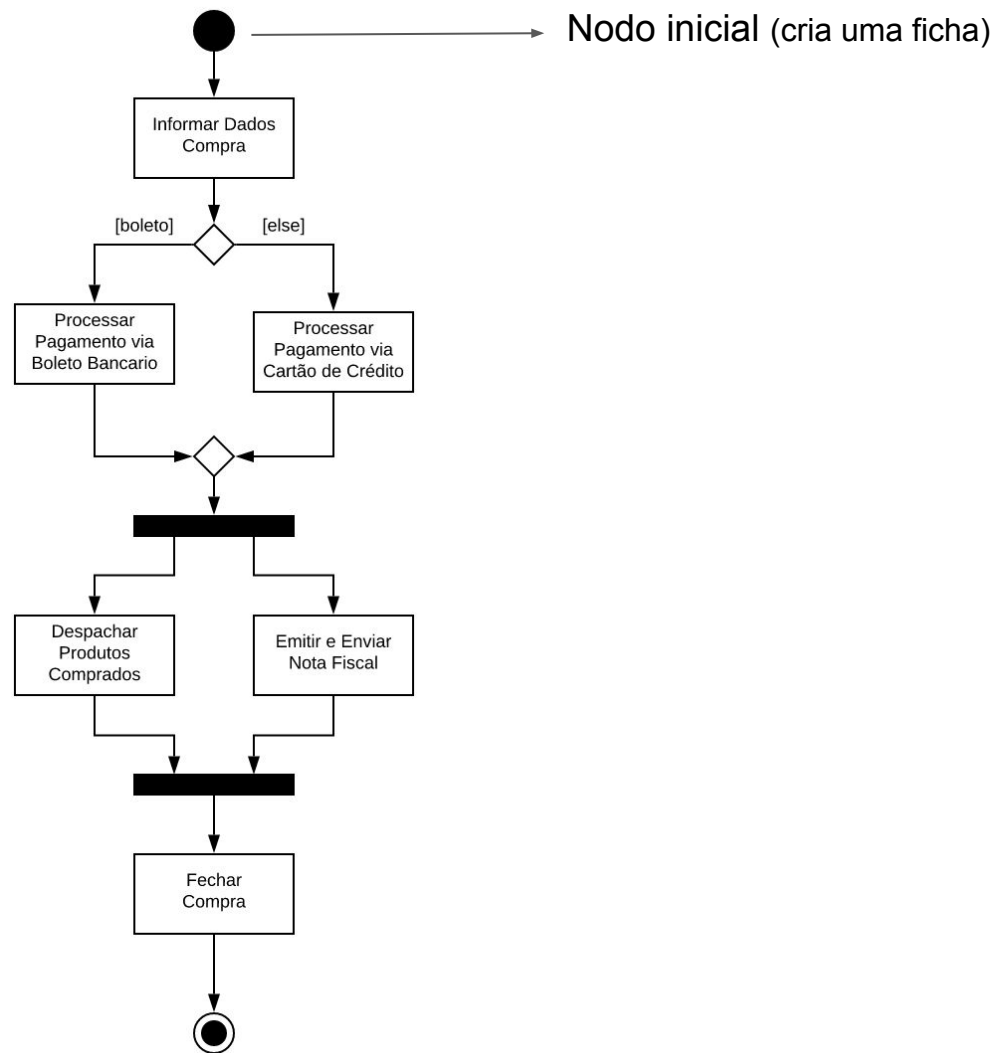
Diagrama de Atividades

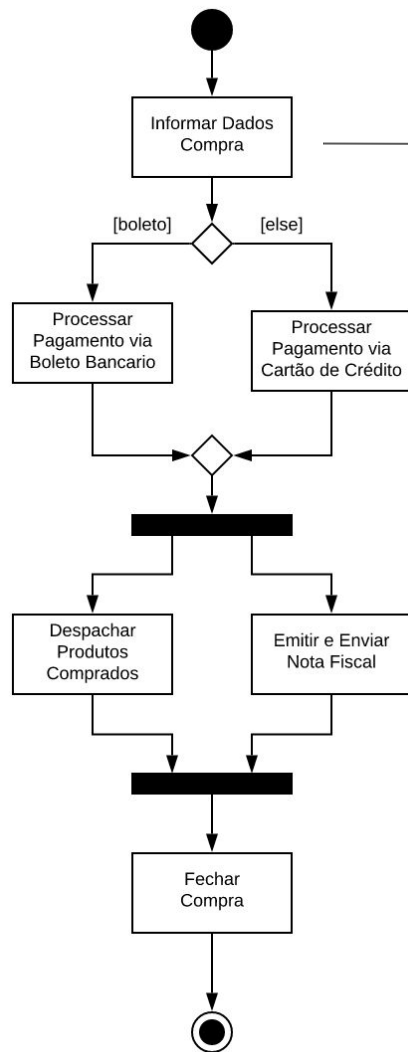
Diagramas de Atividades

- São também diagramas comportamentais ou dinâmicos
- Modelam em alto nível um processo ou fluxo de negócio

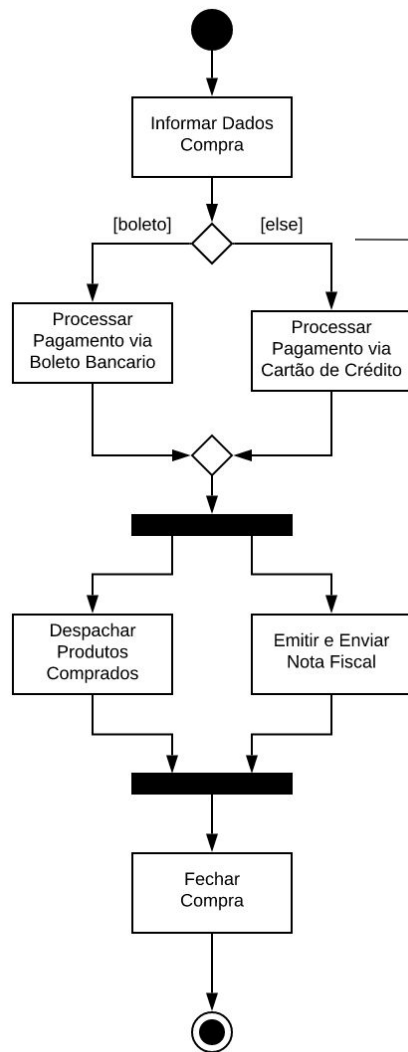


Imagine que existe uma ficha (token) que caminha pelos nodos do diagrama de atividades.

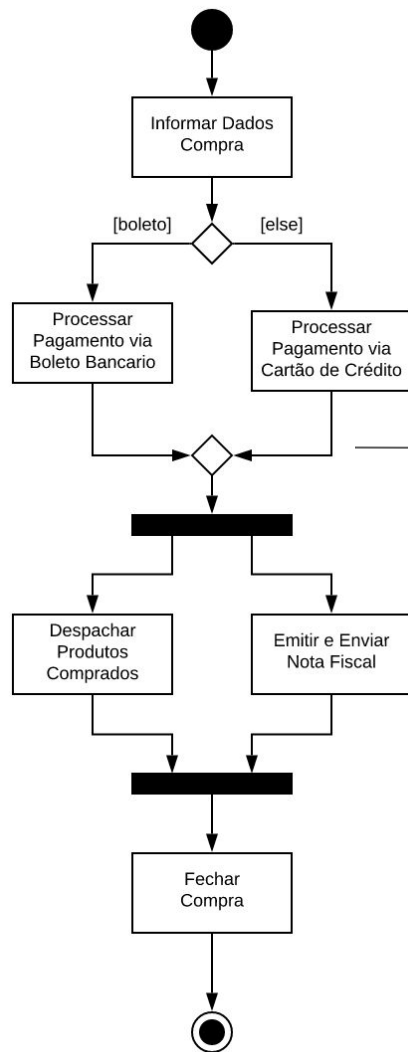




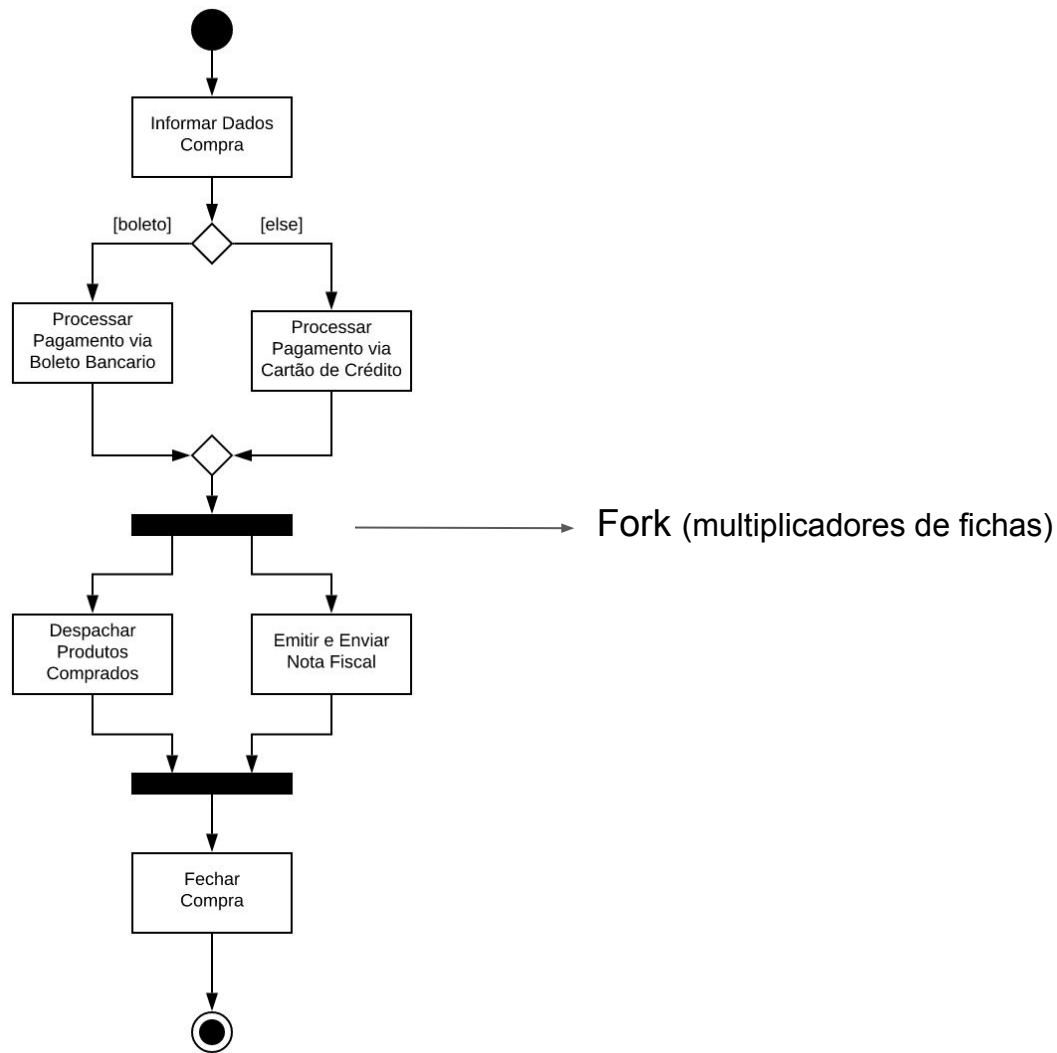
Ação (repassa ficha do fluxo de entrada para saída)

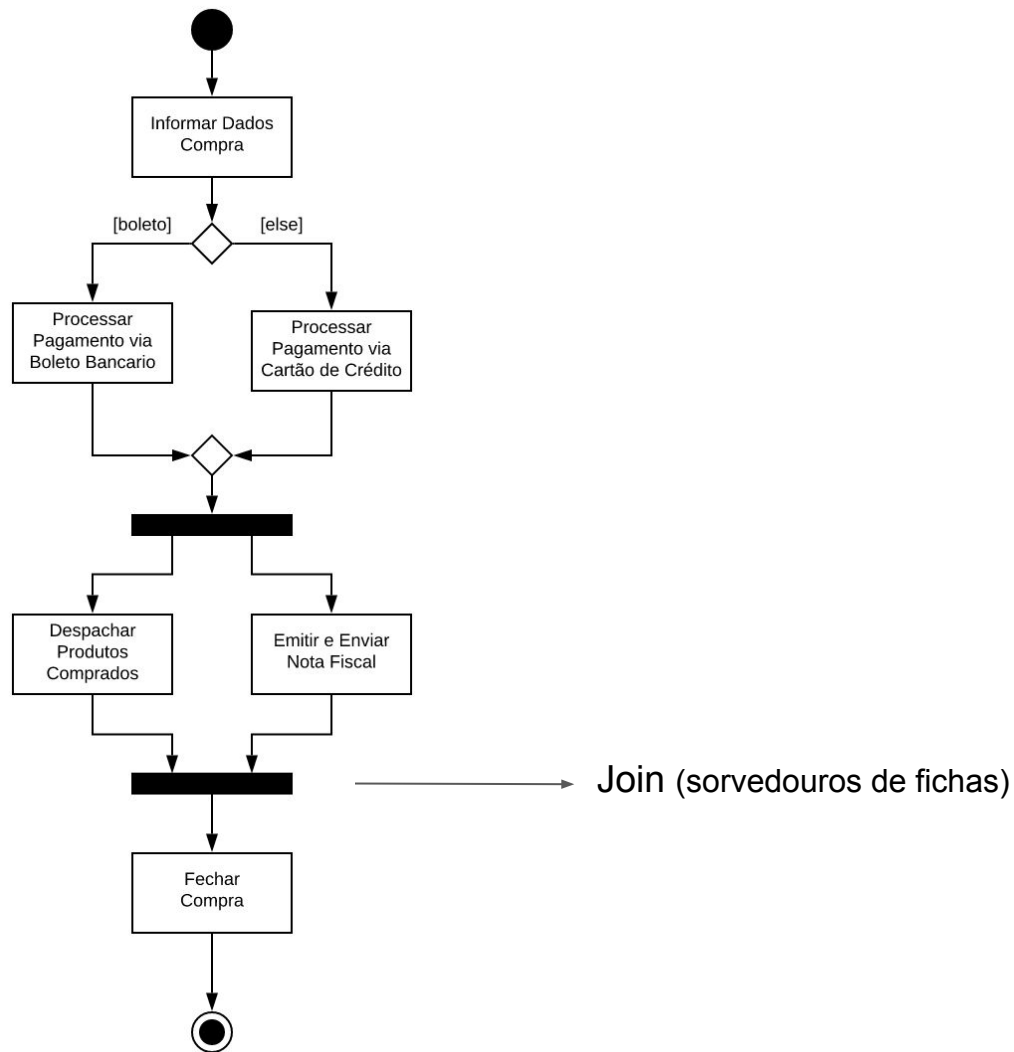


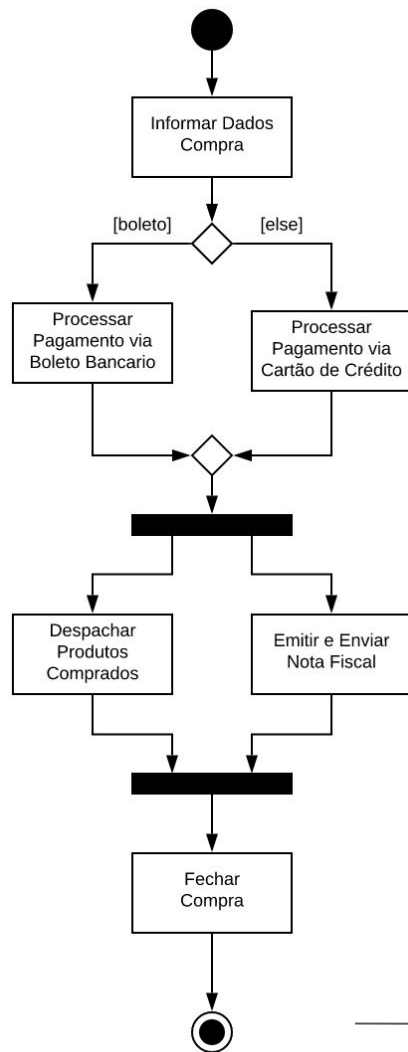
Decisão (decide para qual fluxo de saída repassará a ficha)



Merge (quando ficha chega em uma das entradas, repassa para saída)







Fim