UML e Processo de desenvolvimento de Software

Diagrama de classes
Diagrama de casos de uso
Diagrama de sequência
Diagrama de actividade
Processo de desenvolvimento de software
Tipos de metodologias
Disciplinas de desenvolvimento

UML

- Linguagem de modelação/desenho orientada-a-objectos
  - Evoluiu a partir de vários linguagens do género e é actualmente a norma

- Proporciona diversos diagramas que descrevem o software de diversas perspectivas

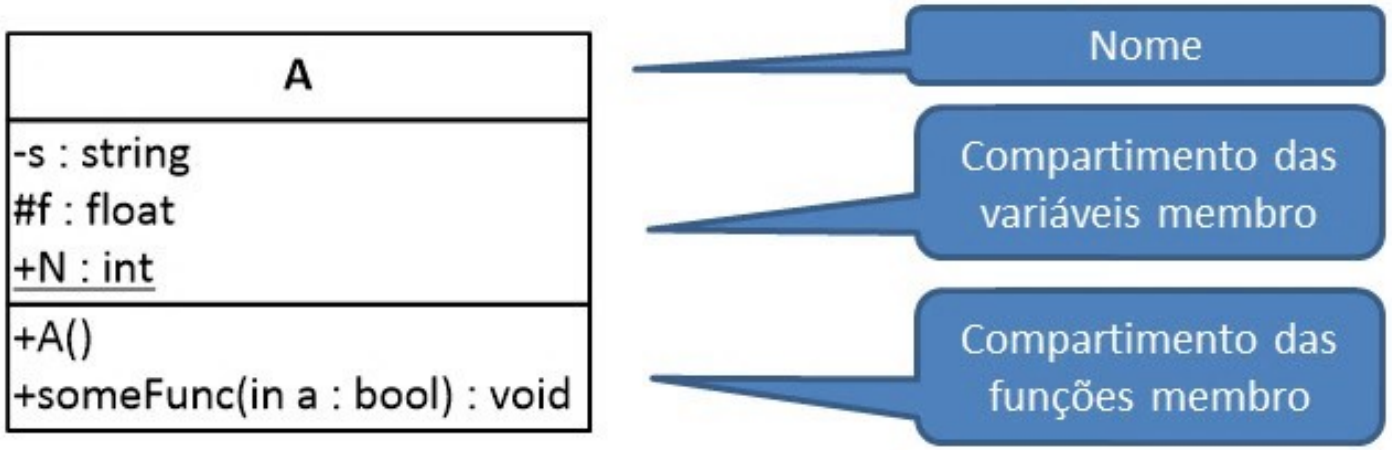
  - Perspectiva Estática ou Estrutural
    - Mostram os elementos estruturais (e.g., classes) e suas relações
    - E.g., diagrama de classes, objectos, pacotes, componentes

  - Perspectiva Dinâmica ou Comportamental (tempo de execução)
    - Mostram o comportamento dos elementos (e.g., objectos) em conjunto (interacção) ou de forma isolada
    - E.g., diagrama de caos de uso, sequência, actividade, estados (*state-charts*)

Diagrama de classes

- Descreve as classes e suas relações

- Classe
  - Representação



  - Descrição de variáveis membro
    <visibilidade> <nome> : <tipo>
  - Descrição de funções membro
    <visibilidade> <nome>(<argumentos>) : <tipo_de_retorno>

'+' = public
'#' = protected
'-' = private

Diagrama de classes

- Relação de Associação
  - Define-se entre classes, mas descreve a relação entre os objectos dessas classes
  - Existe quando os objectos de uma classe utilizam os objectos de outra

    Que conhecem através dos argumentos das suas funções

```
Line l(0.2, 1);
Point p(2, 2);
l.contains(p);
```

Relação mais **fraca** entre os objectos

  - Representação

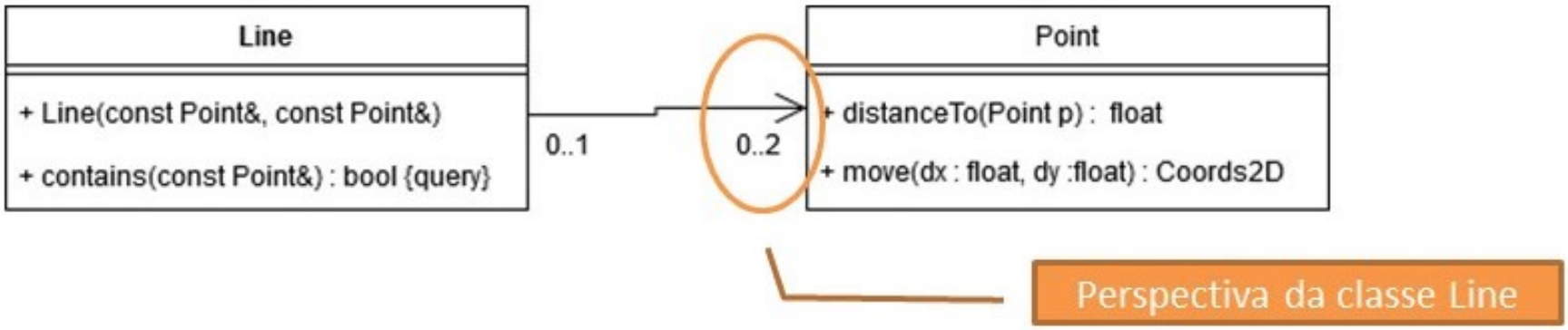


Diagrama de classes

- Extremidades de uma Associação

  - Multiplicidade
    - Número mínimo e máximo de objectos da classe próxima que são utilizados por cada objecto da classe distante
    - 0..N, * = zero ou mais, 1 por defeito
  - Navegabilidade
    - Ponta de seta que indica o sentido da relação (objectos da classe distante utilizam objectos da classe apontada)
    - Unidireccional ou bidireccional
  - Nome
    - Papel do(s) objecto(s) da classe próxima para os objectos da classe distante
    - Apenas quando não é transiente

Diagrama de classes

- Relação de Composição
  - É um tipo mais específico de Associação
  - Existe quando um objecto de uma classe contêm objecto(s) de outra(s)
    - Os objectos de uma classe são parte exclusiva de um objecto da outra
    - Tempos de vida iguais ou classe-todo cria e destrói objectos-parte

```
Point p(1, 2);
Circle c(p, 3);
```

Relação mais **forte** entre os objectos

  - Representação

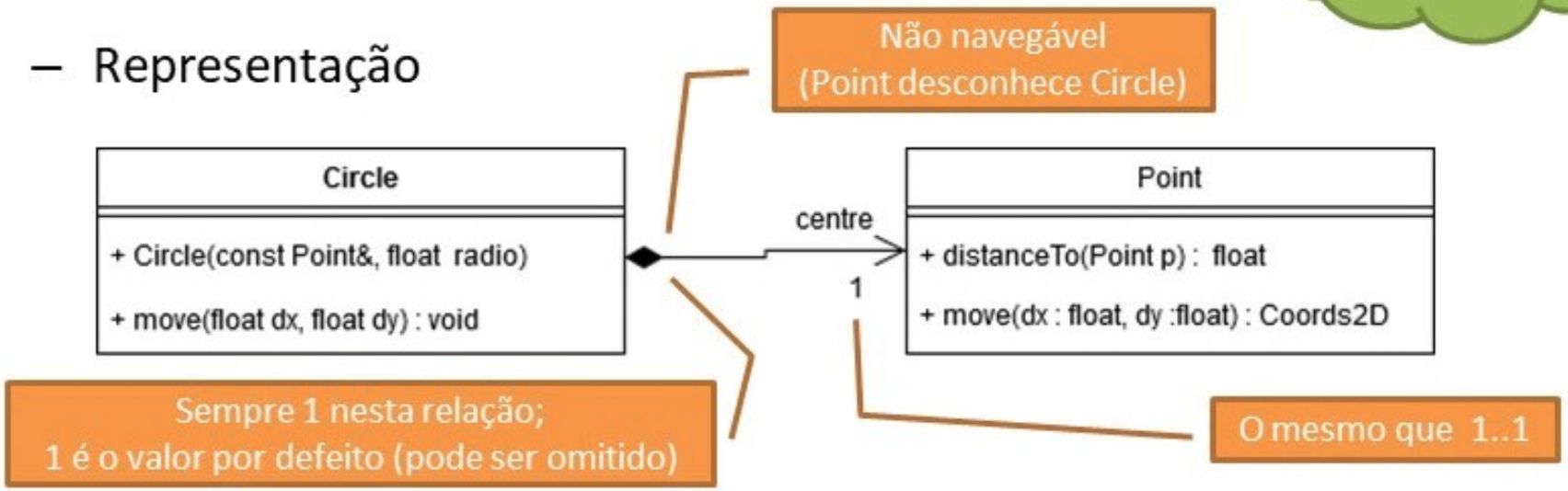
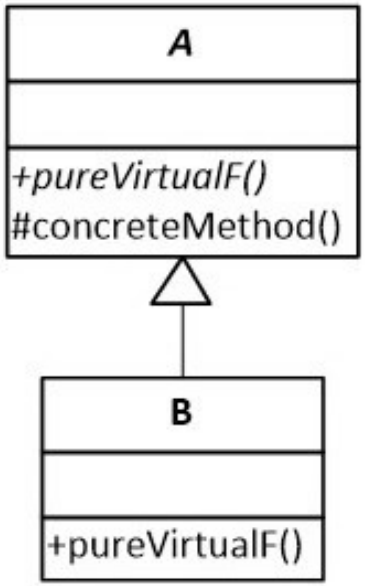


Diagrama de classes

- Relação de Herança
  - Subclasse herda da classe base/mãe
  - Representação



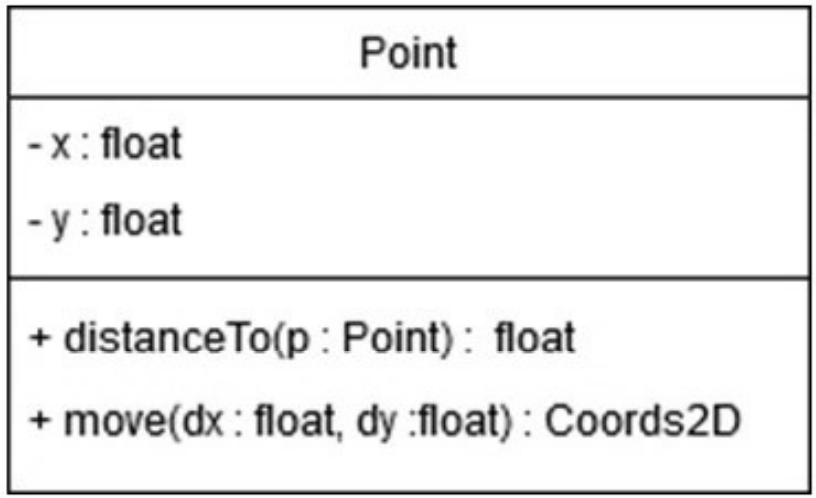

```
class B : public A {
    ...
}
```

- Classes abstractas em itálico
- Métodos abstractos em itálico (funções puramente virtuais)

Classe vs objecto

- Em UML

Classe



Objecto

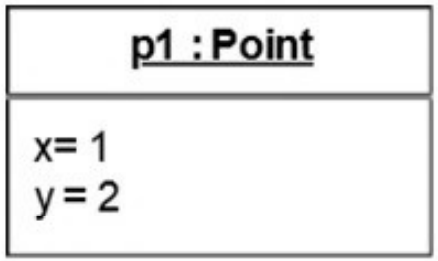


Diagrama de Casos de Uso

- Descrever a utilização do sistema
  - Captar os requisitos
- Elementos
  - Actor

Função/Papel

- Caso de uso

  - Relações entre
    - Actores e Casos de uso

    - Casos de uso

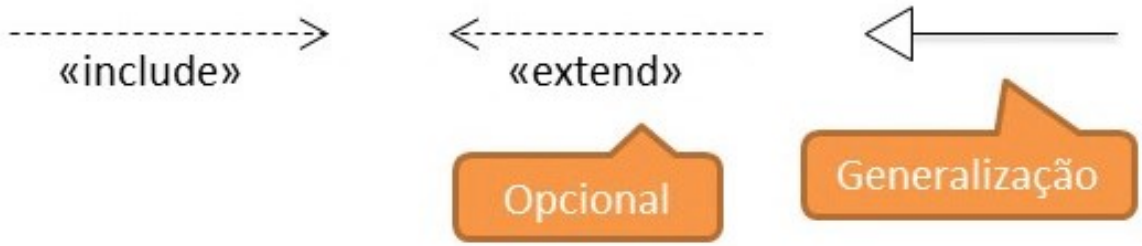


Diagrama de Casos de Uso

- Exemplo: Sistema de elevador(es)

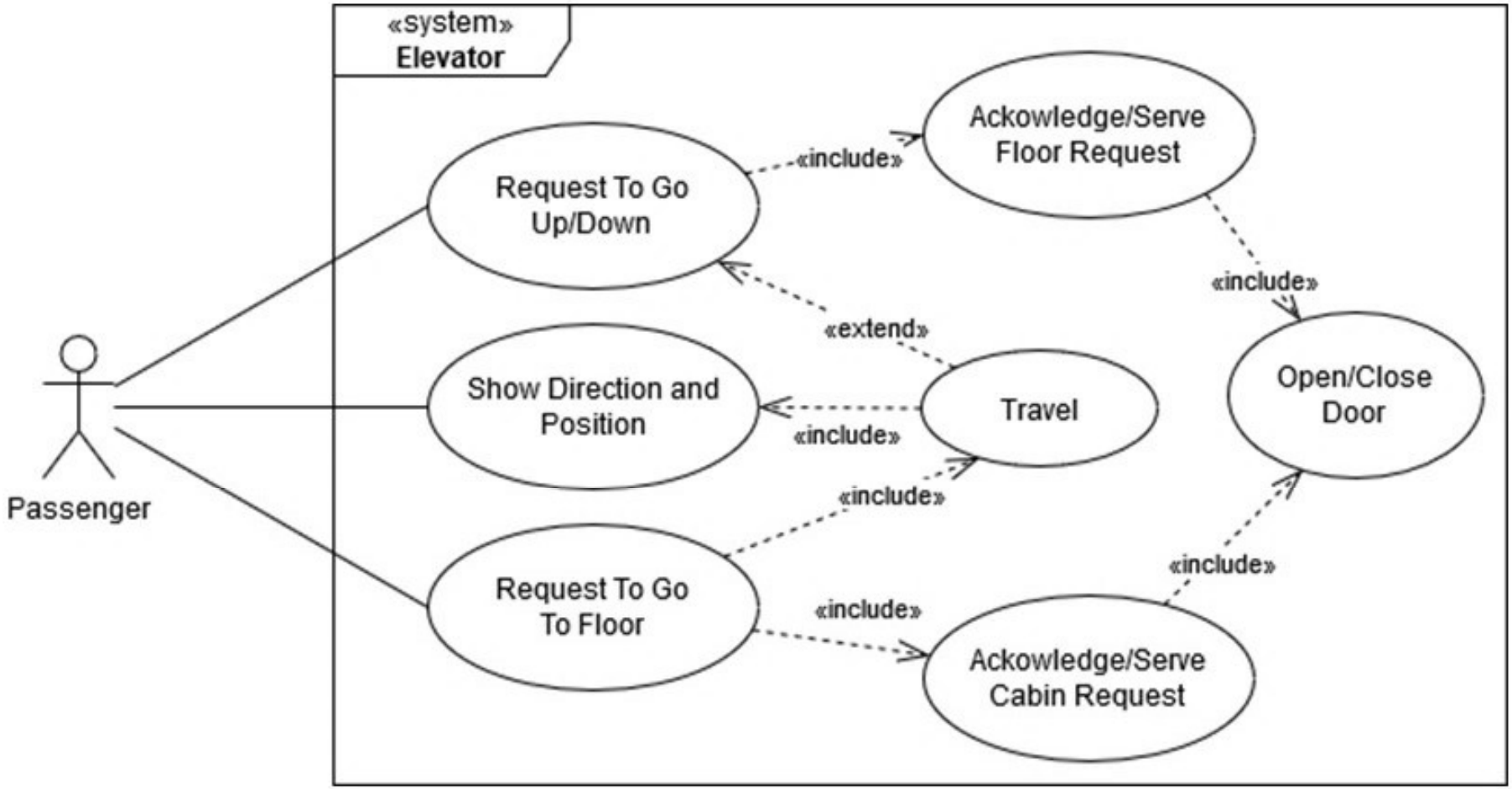


Diagrama de sequência

- Descrever o comportamento do sistema
  - Interacção entre os objectos
  - Sequência (particular) de mensagens

- Elementos principais
  - Objecto
    - E linha de vida
    - A activação (execução de funções)
    - Opcionalmente, o fim (destruição)

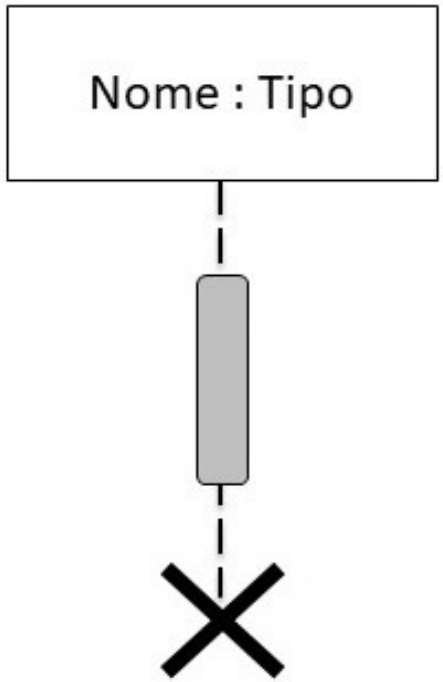
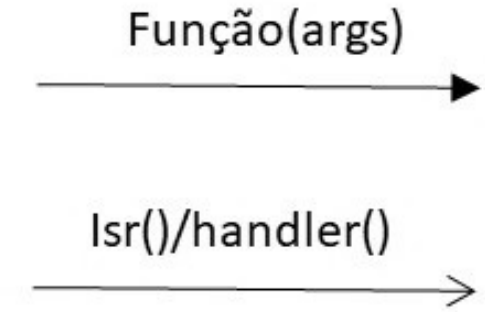
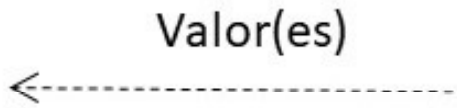


Diagrama de sequência

- Elementos principais
  - Mensagem
    - Síncrona
      (invocação de função)
    - Assíncrona
      (rotina interrupção,
      handler de evento/sinal)



  - Retorno
    (de mensagem síncrona)



  - Criação de objecto

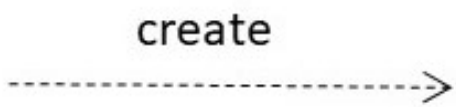
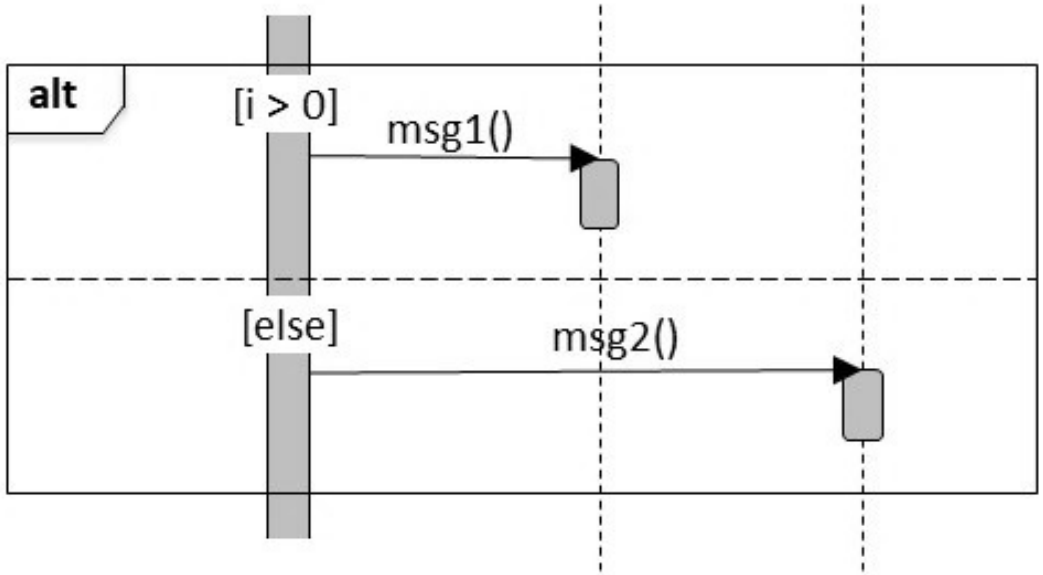


Diagrama de sequência

- Fragmentos
  - Alternativas



  - Ciclo

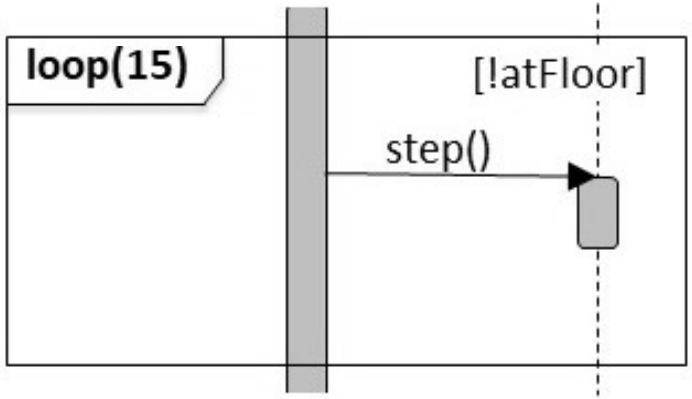


Diagrama de sequência

- Exemplo: chamar o elevador

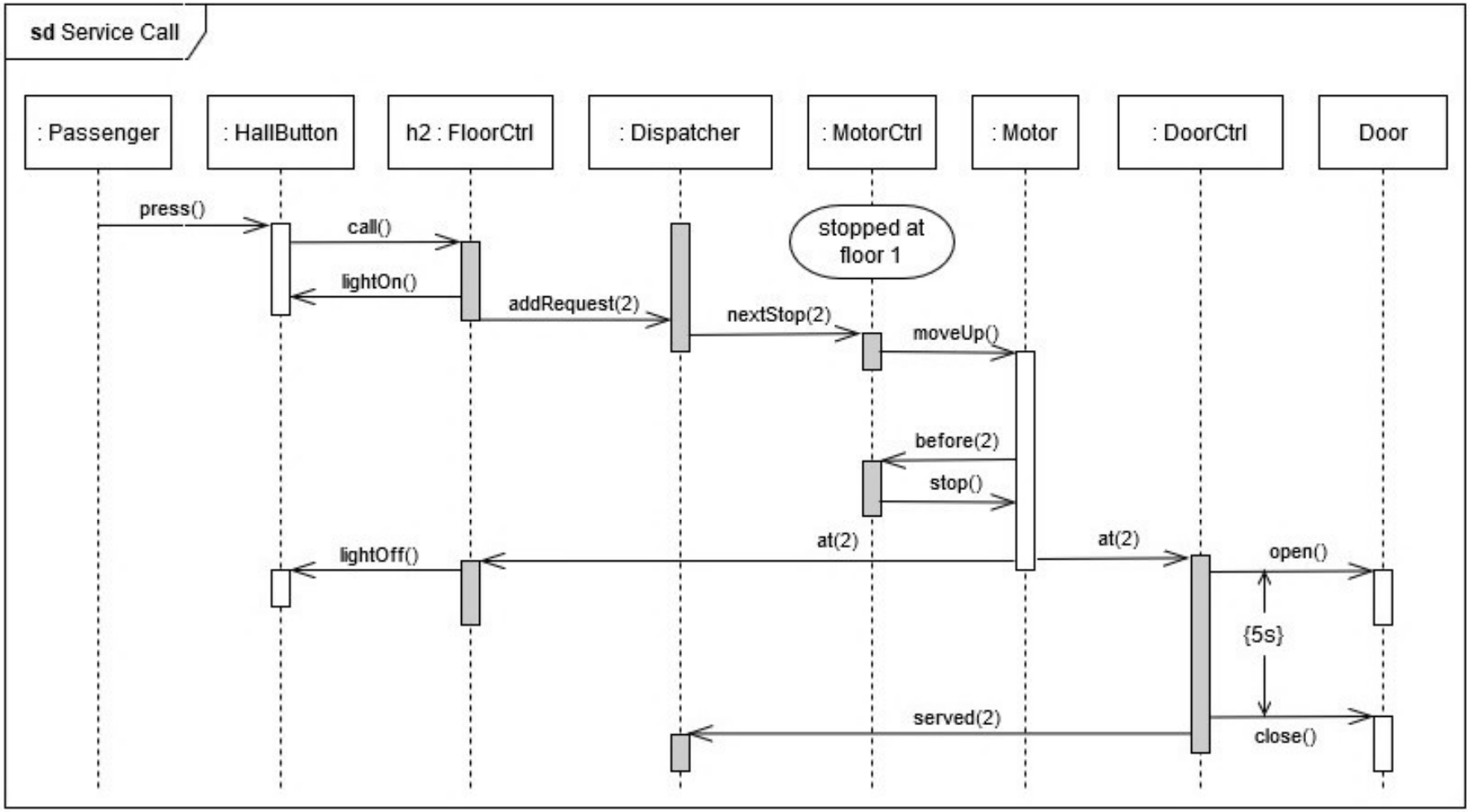
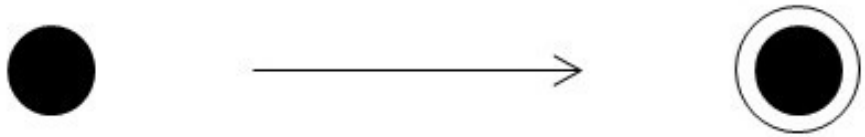


Diagrama de actividade

- Descrever o comportamento do sistema
  - Fluxo completo de acções e procedimentos
    - Completo ≠ detalhado
    - "Fluxograma" mais sofisticado

- Elementos principais
  - Controlo
    - Inicio, Fluxo, Fim

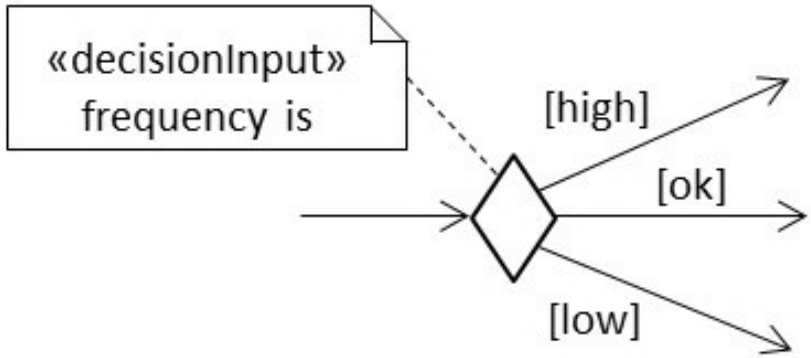


  - Acção
    (função, bloco)

Diagrama de actividade

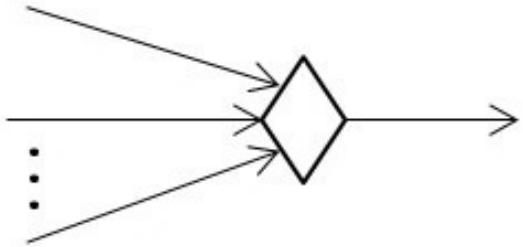
- Elementos principais
  - Decisão

    (**início** de if-else, switch-case)



  - Fusão

    (**fim** de if-else, switch-case)



  - Decisão-Fusão combinadas

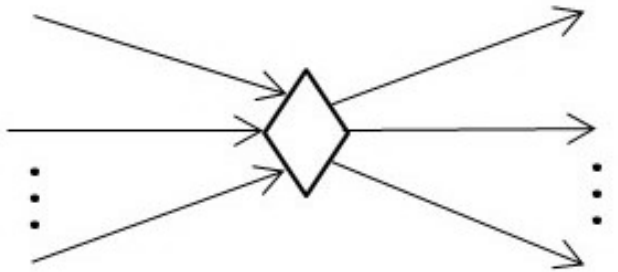
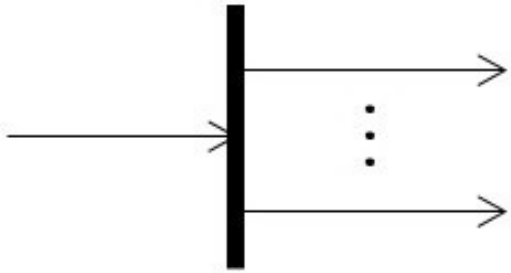


Diagrama de actividade

- Elementos principais
  - Multiplicação

    (do fluxo de controlo; *fork*)

    (e.g., *threads*)



  - Junção

    (sincronização)

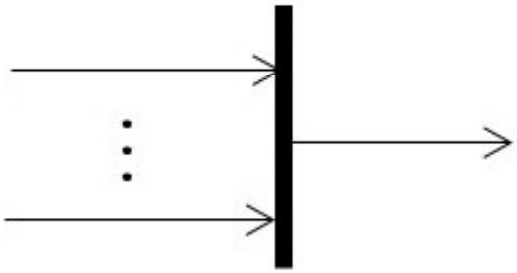


Diagrama de actividade

- Elementos principais
  - Partições
    - Agrupam nós com características comuns
      (e.g., pertencentes a uma classe)
    - Na vertical, horizontal ou ambas (multidimensional)

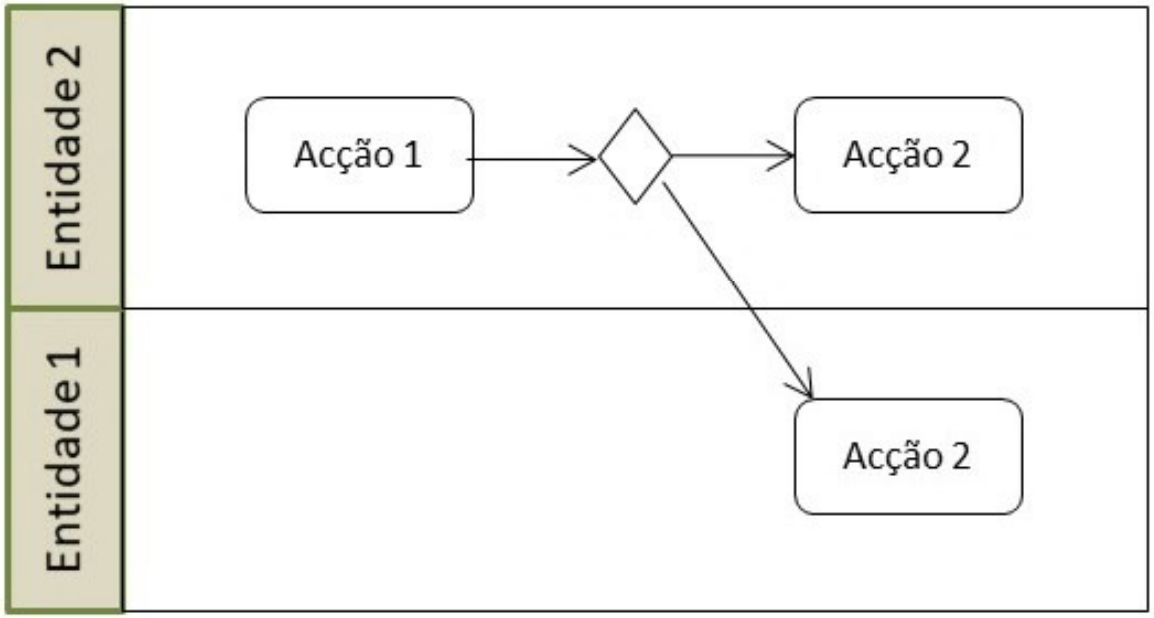


Diagrama de actividade