

The slide features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a dynamic, layered effect. The main title is centered in a large, green, sans-serif font.

Análise e Desenvolvimento de Software

Unified Modeling Language

Pedro Emanuel Cardoso de Sousa
pesousa@ipca.pt

Sobre UML

- ▶ O que é o UML?
- ▶ O que não é?
- ▶ Porque é importante?



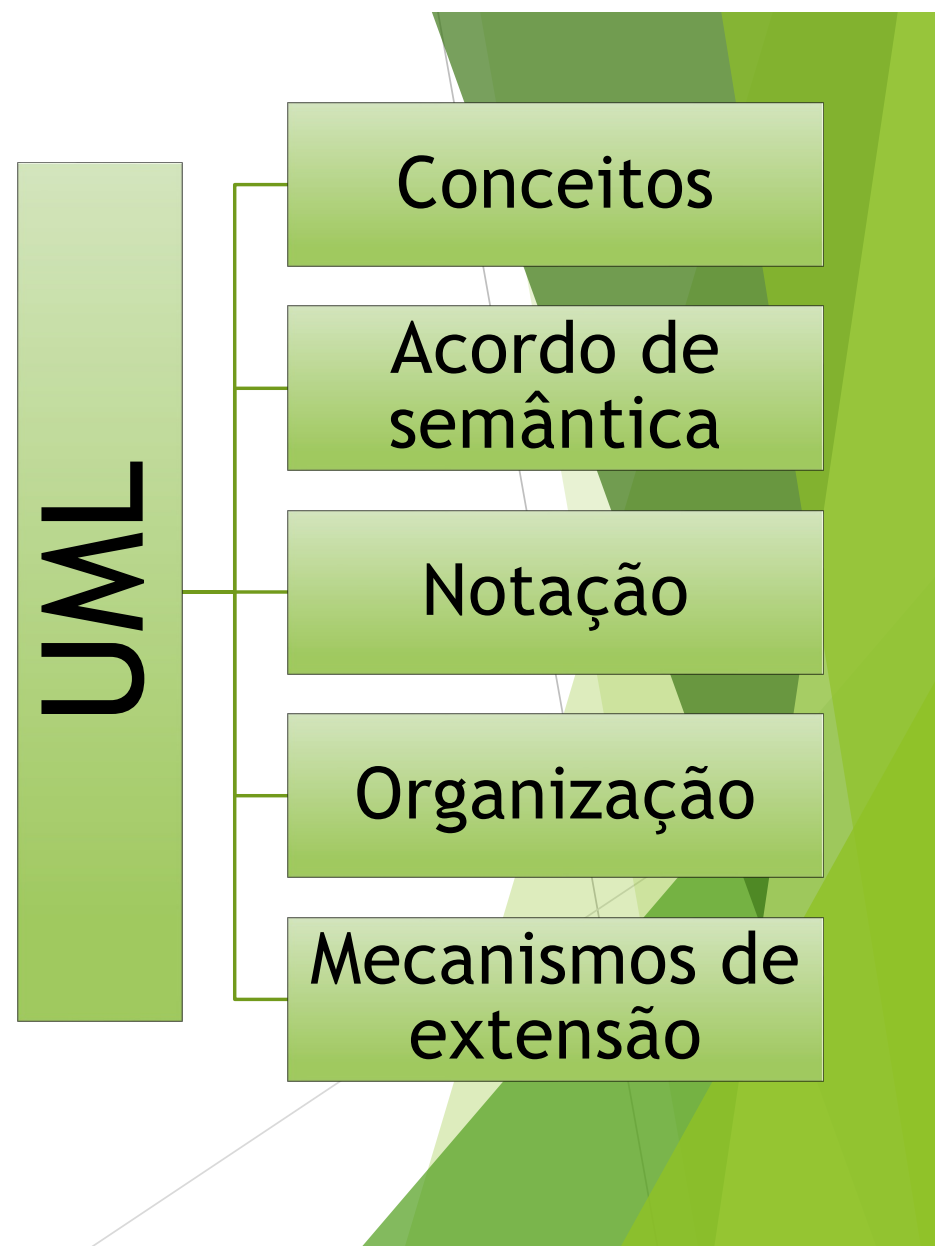
O que é o UML?

- ▶ UML - Unified Modeling Language
- ▶ 3 métodos de modelação:
 - ▶ Object Oriented Software Engineering - Jacobson et al.
 - ▶ Object-Oriented Modeling and Design (OMT) - Rumbaugh et al.
 - ▶ Object-Oriented Analysis and Design with Applications - Booch

O que é o UML?



- ▶ UML é uma linguagem de modelação
- ▶ Disponibiliza:
 - ▶ Conceitos - O que interessa em modelação
 - ▶ Acordo de semântica - Os conceitos tem significado
 - ▶ Notação - Indica como representar o conceito
 - ▶ Organização - Estruturar e apresentar os modelos
 - ▶ Mecanismos de extensão - Modelar novos conceitos
- ▶ UML é um standard
- ▶ ISO 19505
- ▶ Object Management Group (www.omg.org)
- ▶ Usado para especificar outros standards



O que não é?

- ▶ Ferramenta / aplicação de modelação
 - ▶ Não é uma linguagem de programação
 - ▶ Não é uma metodologia, processo ou método de análise
-
- ▶ Conhecer o UML não significa que saiba aplicar.



Porque é importante

- ▶ Unifica as 3 notações originais
- ▶ Disponibiliza um standard para modelação que os consultores e fornecedores podem usar
- ▶ Define mecanismos de interoperabilidade para transportar entre aplicações
- ▶ É a base para definir outros standard → Meta Object Facility (MOF)
- ▶ Existe um ponto de vista onde apenas é necessário aprender UML, como sistema de notação
- ▶ É a notação de referência para a maioria das publicações e aplicações
- ▶ É possível ampliar a notação para cobrir outros domínios

MODELOS E DIAGRAMAS



Modelos e Diagramas

- ▶ Porque modelar sistemas?
- ▶ Modelos
- ▶ Modelação e desenhos
- ▶ UML → tipos de diagramas



Porque modelar sistemas

- ▶ Sistemas pequenos: provavelmente não necessitamos de modelar
- ▶ A maioria dos sistemas são grandes e complexos
- ▶ Os programas integram outros programas
- ▶ Modelar permite a analistas e programadores:
 - ▶ Ter a visão do sistema por inteiro
 - ▶ Perceber a razão dos aspetos existentes
 - ▶ Considerar alternativas
 - ▶ Analisar riscos
 - ▶ Comunicar com pessoas relevantes para o sistema



Porque modelar sistemas

- ▶ Modelos são diferentes das coisas que representam
 - ▶ O modelo é mais fácil e mais rápido de produzir
 - ▶ Modelo pode ser usado em simulações
 - ▶ Modelo pode evoluir, conforme aprendemos mais sobre o sistema ou problema
 - ▶ Podemos escolher o que incluir no modelo
 - ▶ O modelo pode representar realidade ou imaginação de determinado domínio



Modelos

- ▶ O modelo é uma abstração do mundo real
- ▶ UML define módulo assim:

“A model captures a view of a physical system. It is an abstraction of the physical system, with a certain purpose. This purpose determines what is to be included in the model and what is irrelevant. Thus the model completely describes those aspects of the physical system that are relevant to the purpose of the model, at the appropriate level of detail.”*

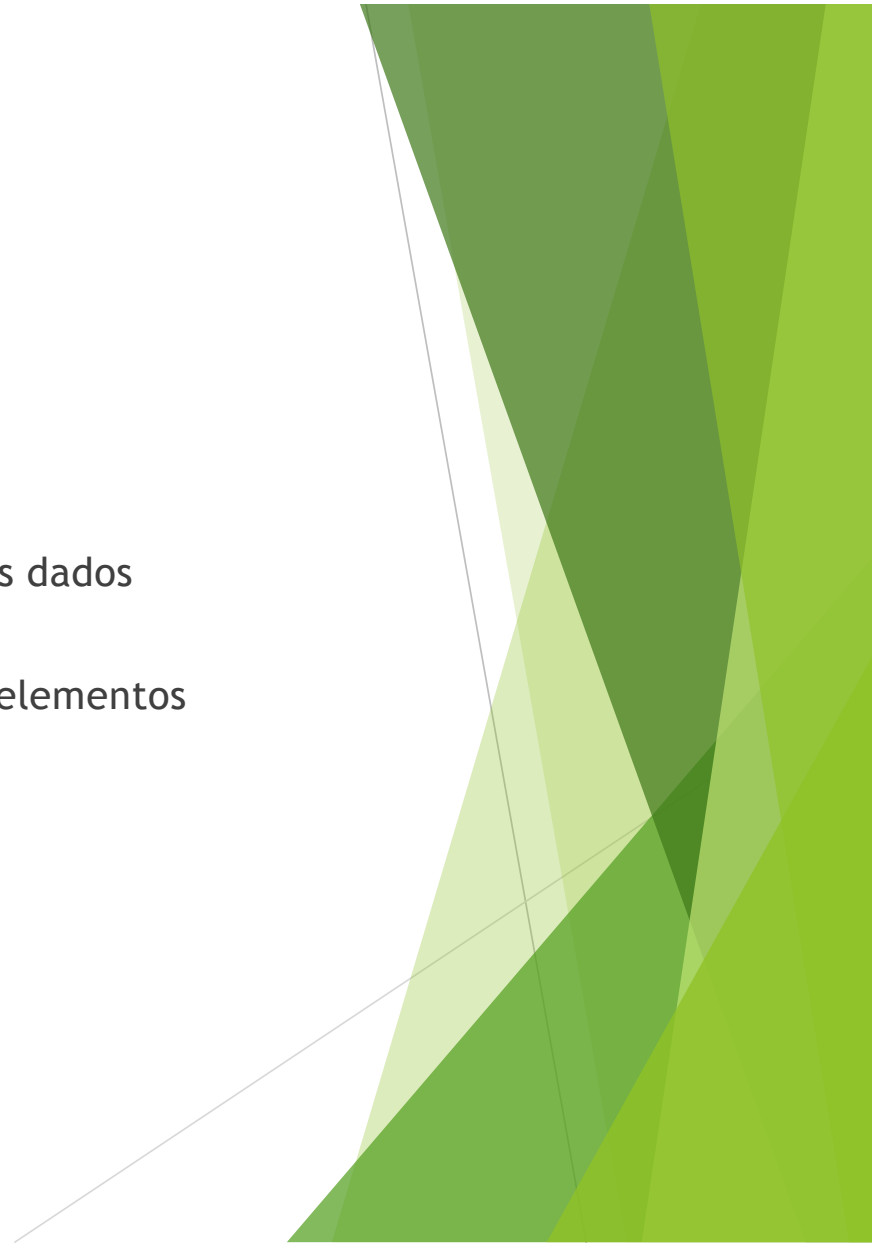
Modelos UML

- ▶ Contém dados sobre todos os elementos do modelo
- ▶ Contém dados sobre as relações entre os elementos do modelo
- ▶ Pode conter pontos de vista predefinidos do modelo
- ▶ Conter ou construir visualizações com base em pontos de vista
- ▶ Contém dados sobre a geometria de vistas de diagrama particulares



Modelação e Desenhos

- ▶ Em UML: um diagrama é uma vista perante o modelo
- ▶ Um diagrama isolado não tem a representação do modelo
- ▶ Se um diagrama deriva do modelo, é dinâmico e depende dos dados seleccionados
- ▶ Diagramas diferentes disponibilizam vistas sobre os mesmos elementos



UML Tipos de Diagramas

► Estruturais:

- Object
- Class
- Component
- Deployment
- Package
- Composite structure

► Comportamentos:

- Use cases
- Sequence
- Communication
- Interaction overview
- Timing
- State machine
- Activity



UML - ELEMENTOS COMUNS



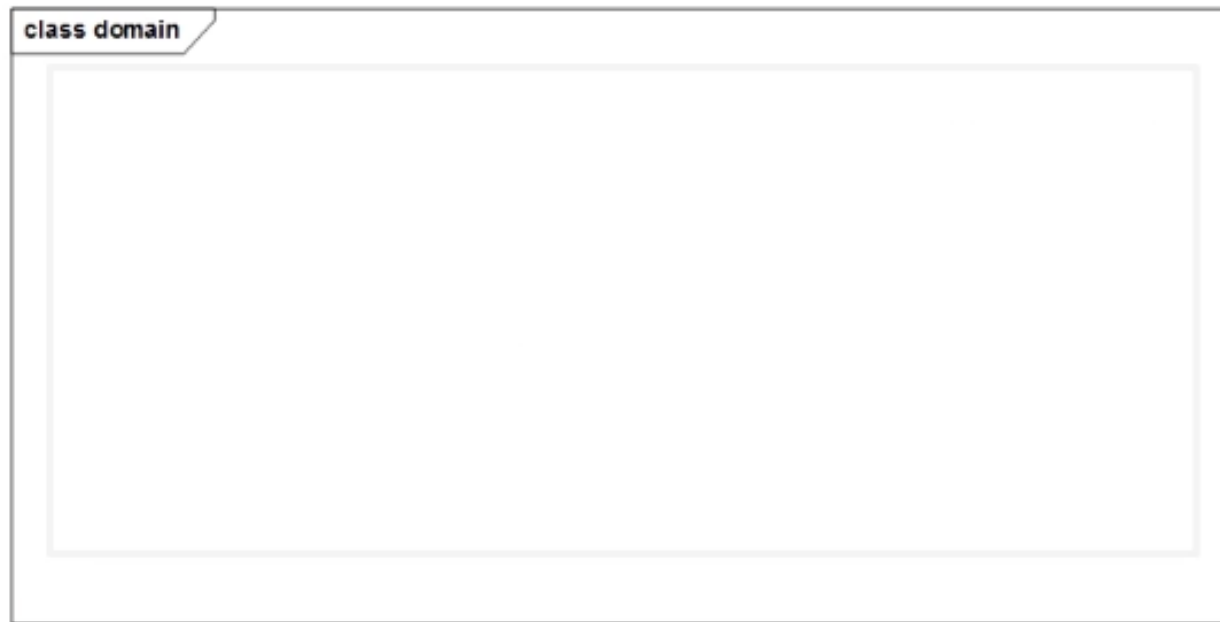
UML - Elementos comuns

- ▶ Frames
- ▶ Classifiers
- ▶ Features and Properties
- ▶ Comments/Notes and constrains
- ▶ Dependencies
- ▶ Stereotypes



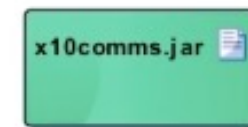
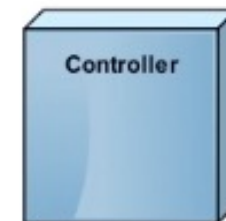
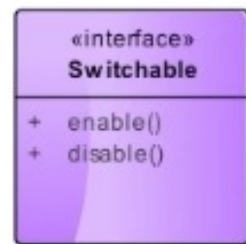
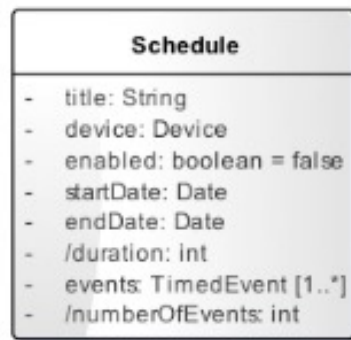
UML - Frames

- Utilizado para conter diagramas



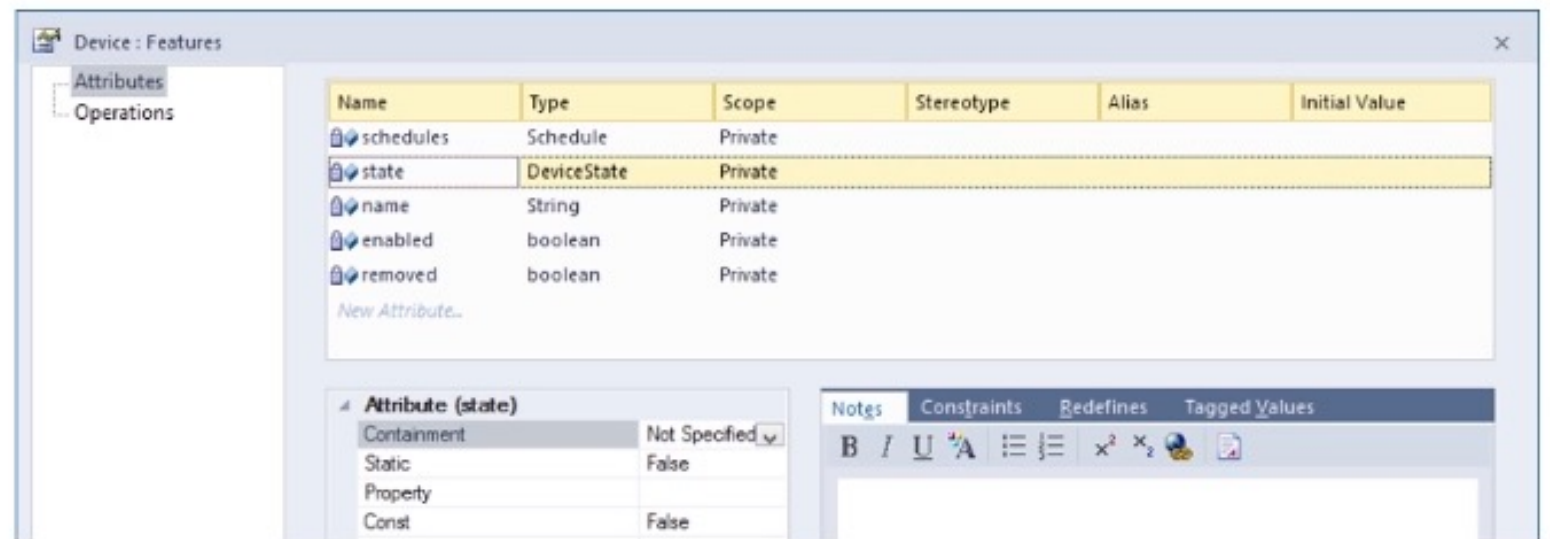
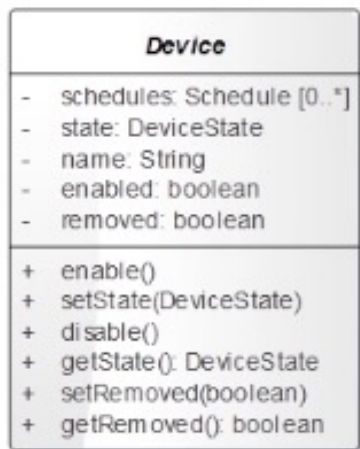
UML - Classifiers

- Descreve o tipo de objetos e usar num determinado domínio
- Generalização das classes, interfaces, componentes, etc.



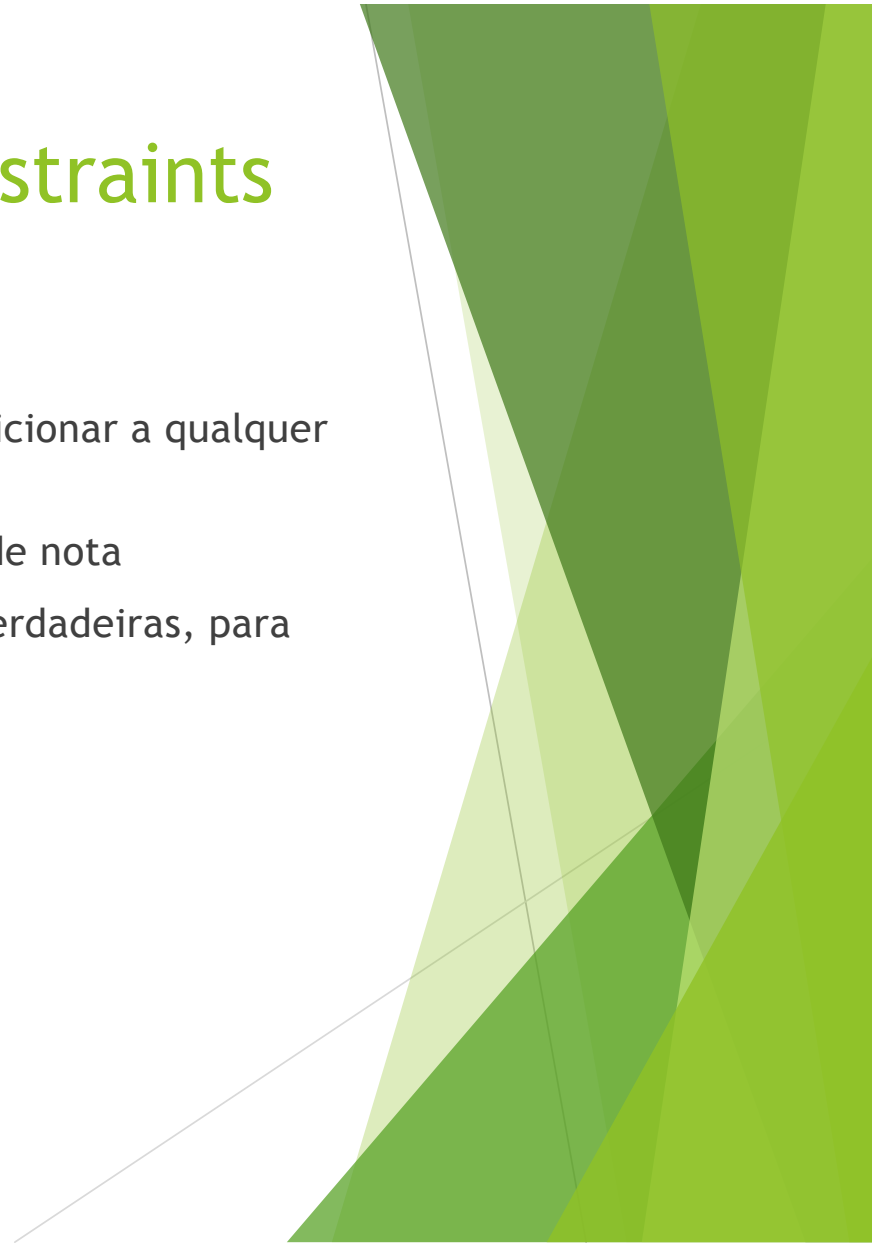
UML - Features and Properties

- Características estruturais ou comportamentais dos elementos que pertencem ao modelo, tais como atributos e operações das classes.



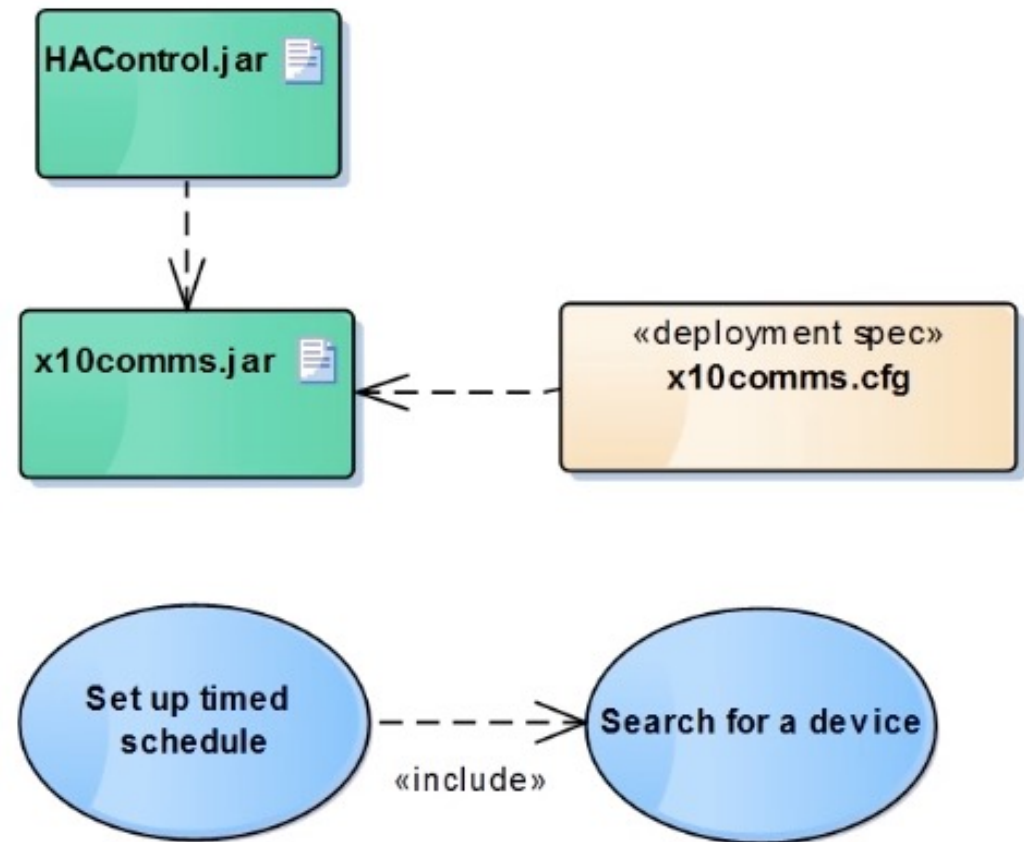
UML - Comments/notes and Constraints

- ▶ Comentários/notas são anotações em texto que podemos adicionar a qualquer elemento do modelo
- ▶ Tipicamente aparecem dentro de uma caixa com o símbolo de nota
- ▶ Constraints são elementos ou propriedades que devem ser verdadeiras, para se verificarem



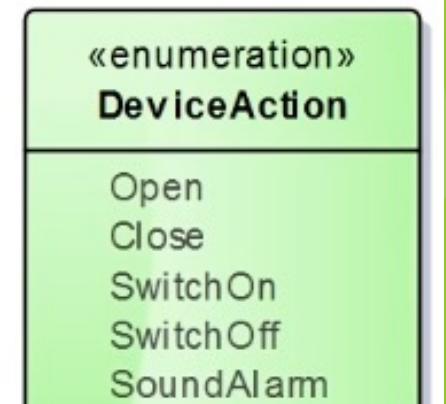
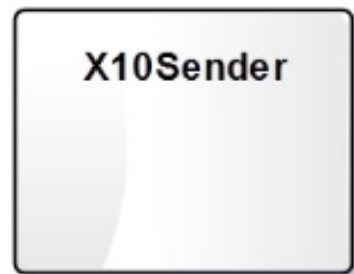
UML - Dependencies

- ▶ São apresentadas como setas picotadas
- ▶ Indicam relação de dependência entre elementos
- ▶ Também pode ser visto como um estereótipo



UML - Stereotypes

- Define a extensão de determinada classe (meta-classe), para um tipo especial de elemento no nosso modelo



Obrigado !!!

