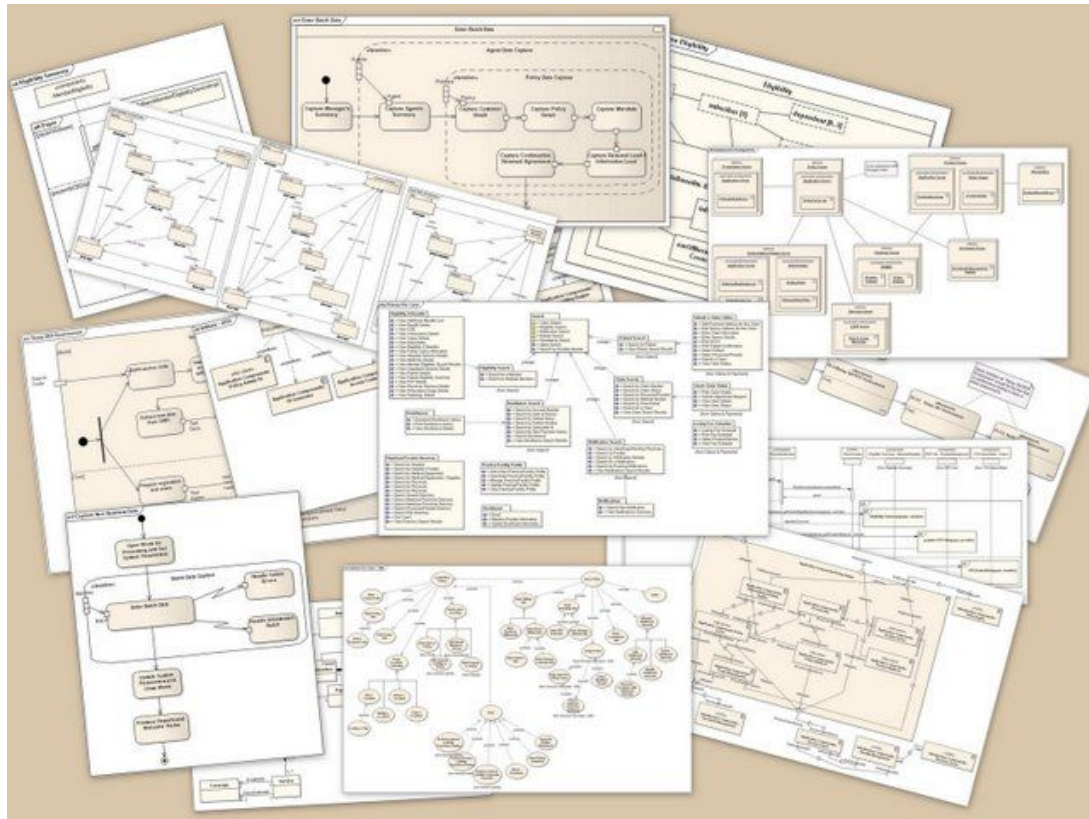

🔥 💀 🔥 UML diagrams 🔥 💀 🔥

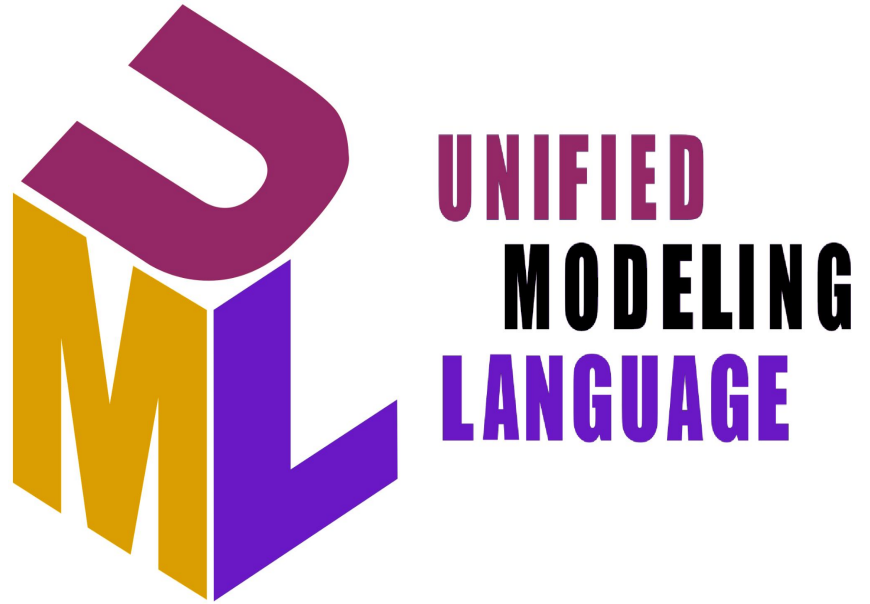


Introducción



UML

- Unified Modeling Language
- Es un lenguaje visual
- Nos permite modelar
 - Procesos
 - Sistemas
 - Software
- Es extensible, flexible y escalable

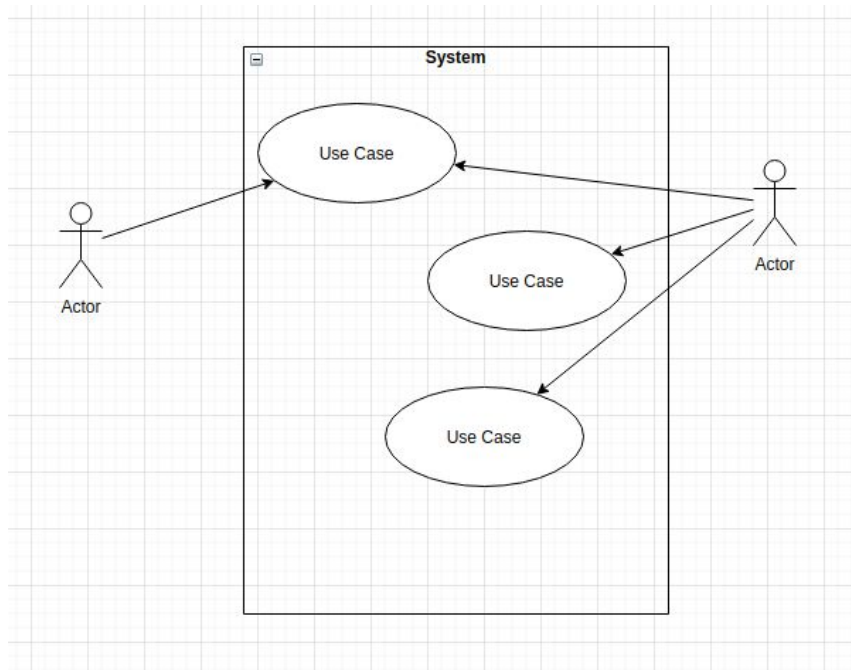


Use Case Diagram

Nos ayudan a capturar los requisitos funcionales del sistema a desarrollar

Elementos

- Caso de uso
- Actor
- Comunicación
- Entorno

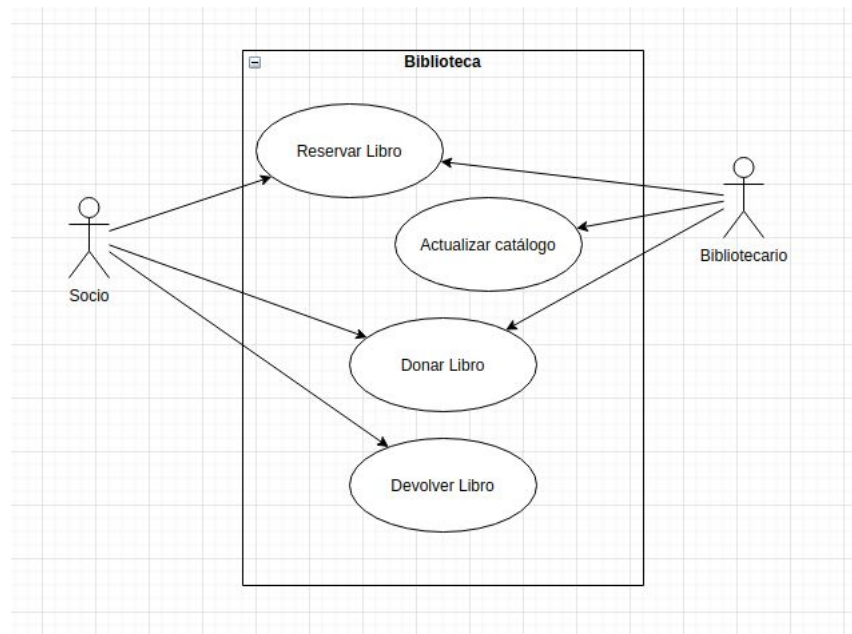


Use Case Diagram - Example

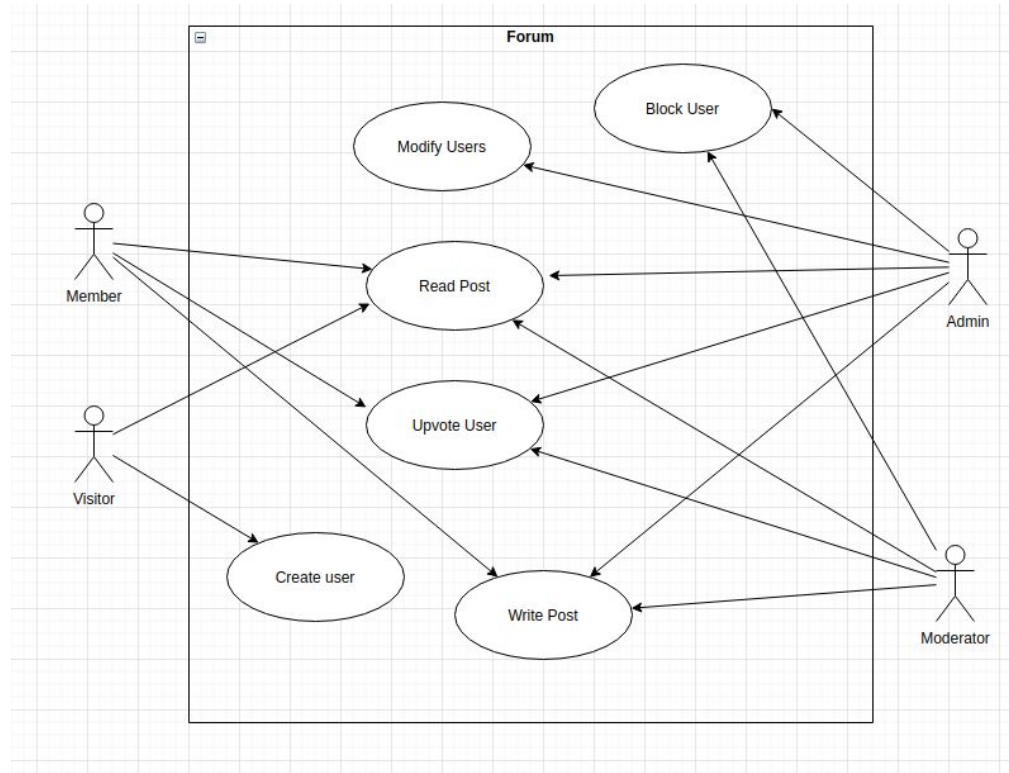
Nos ayudan a capturar los requisitos funcionales del sistema a desarrollar

Elementos

- Caso de uso
- Actor
- Comunicación
- Entorno



Use Case Diagram - Example



Use Case Diagram - Relationships

Relaciones entre casos de uso:

- Inclusión
- Extensión
- Herencia

Relaciones entre actores

- Herencia

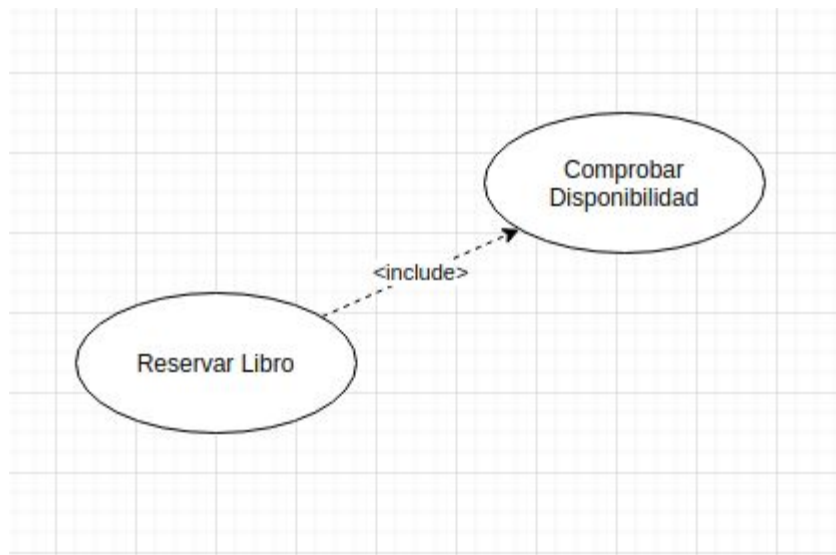
Use Case Diagram - Relationships

Relaciones entre casos de uso:

- **Inclusión**
- Extensión
- Herencia

Relaciones entre actores

- Herencia



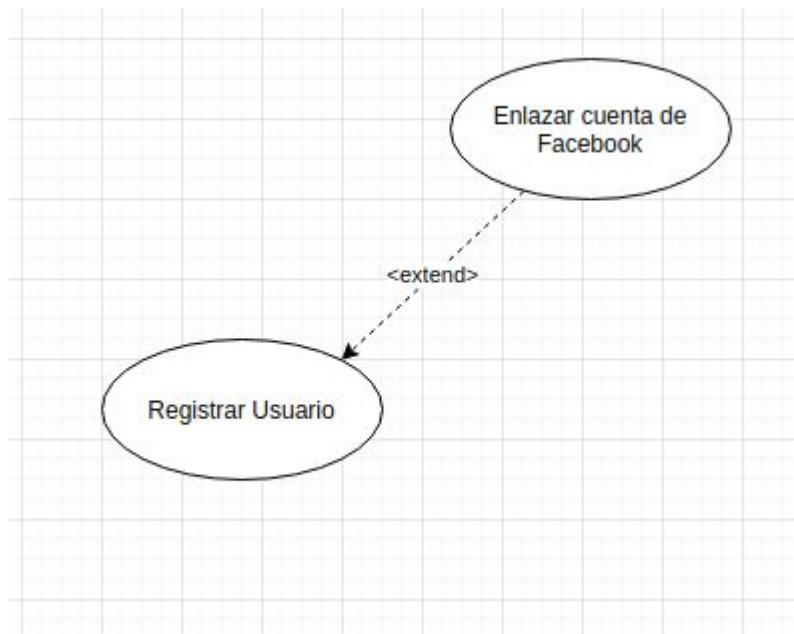
Use Case Diagram - Relationships

Relaciones entre casos de uso:

- Inclusión
- **Extensión**
- Herencia

Relaciones entre actores

- Herencia



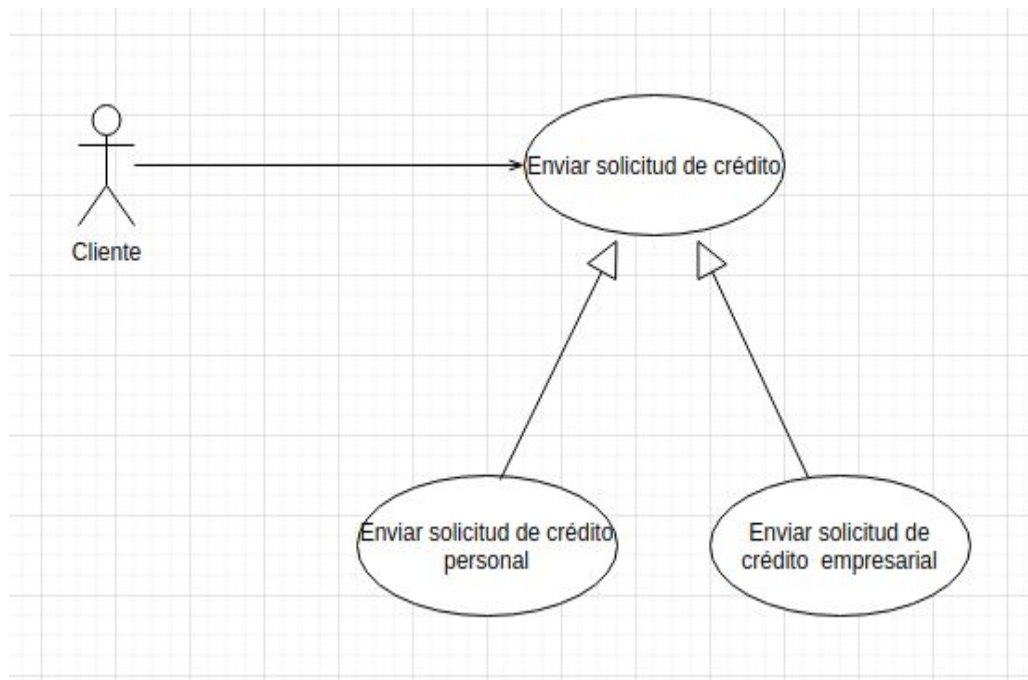
Use Case Diagram - Relationships

Relaciones entre casos de uso:

- Inclusión
- Extensión
- **Herencia**

Relaciones entre actores

- Herencia



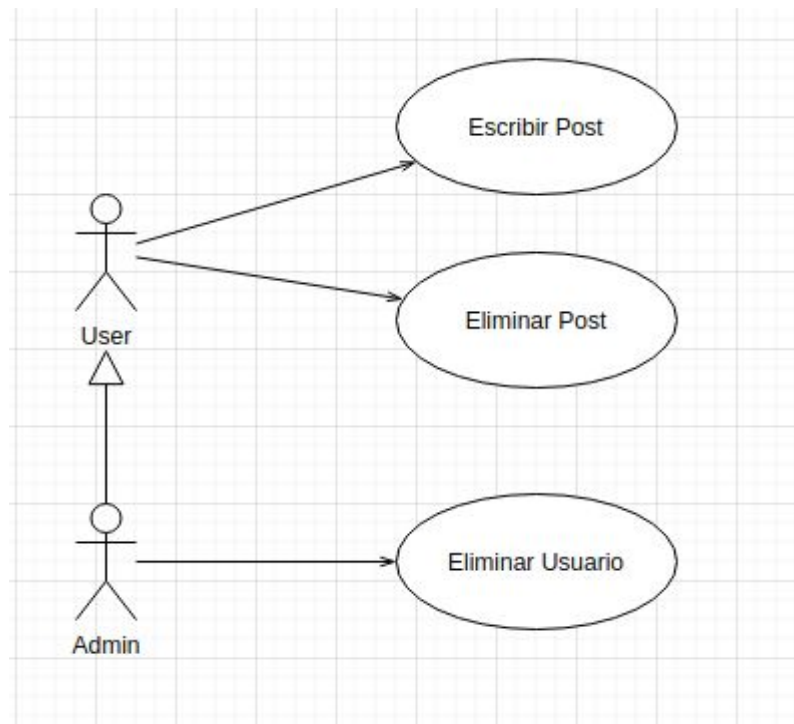
Use Case Diagram - Relationships

Relaciones entre casos de uso:

- Inclusión
- Extensión
- Herencia

Relaciones entre actores

- ***Herencia***



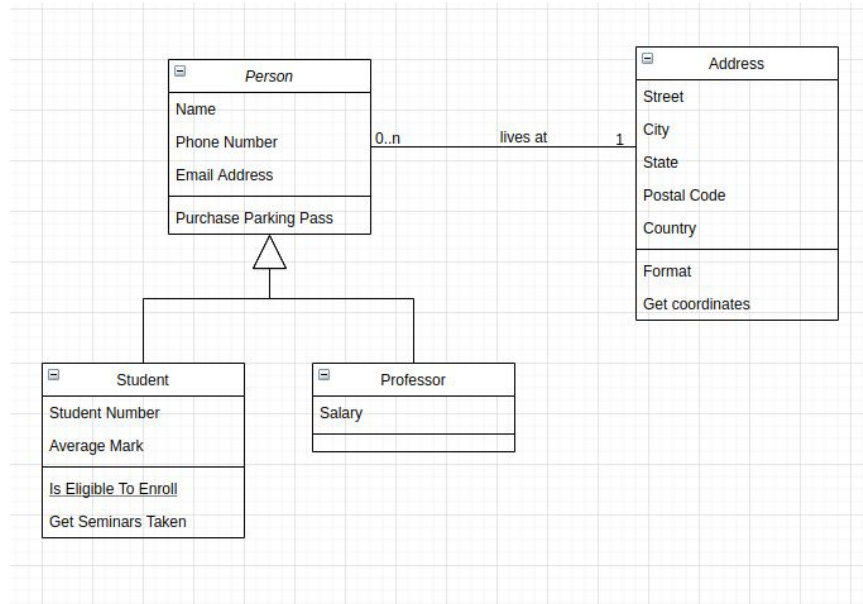


Class Diagram

Nos ayudan a modelar la estructura de un sistema.

Elementos

- Clase
- Asociación
- Agregación
- Composición
- Especialización



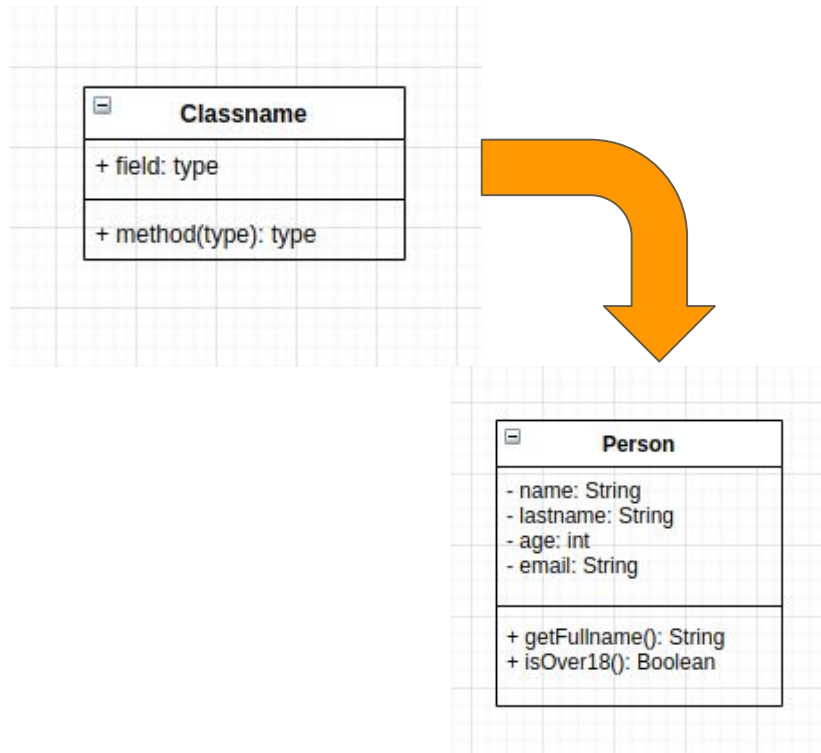
Class Diagram

Clases que conforman nuestro programa. Puede ser tan detallado como requiera el diagrama.

Podemos definir tanto atributos como métodos

Elementos

- **Clase**
- Asociación
- Agregación
- Composición
- Especialización



Class Diagram - Relationships

Relación estructural entre clases que especifica que ambas clases tienen algún tipo de relación

- Es bidireccional
- Se le asigna un nombre o nombres de rol
- Tiene multiplicidad en cada sentido

MULTIPLICIDAD

0..1 cero a uno (opcional)

n (cantidad específica)

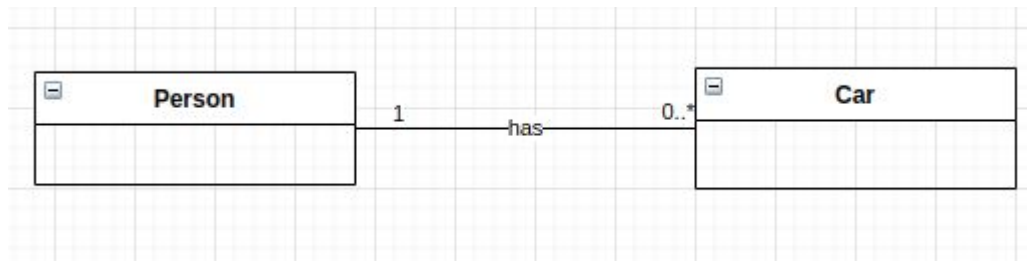
0..* cero a muchos

1..* uno a muchos

m..n rango específico

Elementos

- Clase
- **Asociación**
- Agregación
- Composición
- Especialización



Class Diagram - Relationships

La agregación es una variante de la asociación que representa una relación de tipo parte-todo o parte-de

- Es bidireccional
- Tiene multiplicidad en cada sentido

Elementos

- Clase
- Asociación
- **Agregación**
- Composición
- Especialización

MULTIPLICIDAD

0..1 cero a uno (opcional)

n (cantidad específica)

0..* cero a muchos

1..* uno a muchos

m..n rango específico



Class Diagram - Relationships

En la composición intentamos representar el todo y sus partes. Si eliminamos el contenedor el contenido también es eliminado

- Es bidireccional
- Tiene multiplicidad en cada sentido

MULTIPLICIDAD

0..1 cero a uno (opcional)

n (cantidad específica)

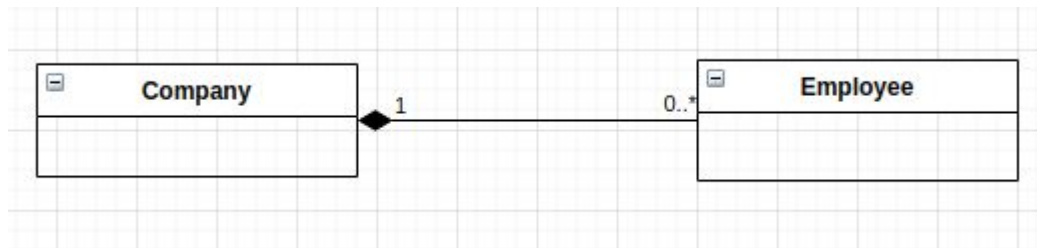
0..* cero a muchos

1..* uno a muchos

m..n rango específico

Elementos

- Clase
- Asociación
- Agregación
- **Composición**
- Especialización

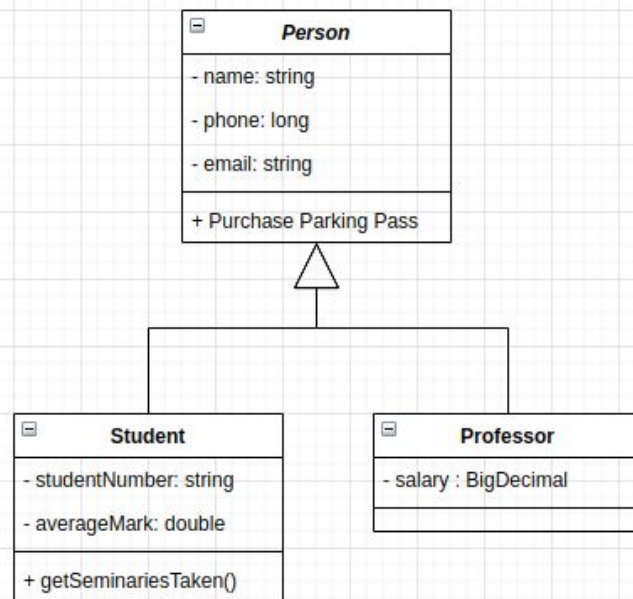


Class Diagram - Relationships

- La especialización se usa para indicar la herencia.

Elementos

- Clase
- Asociación
- Agregación
- Composición
- **Especialización**



Class Diagram

