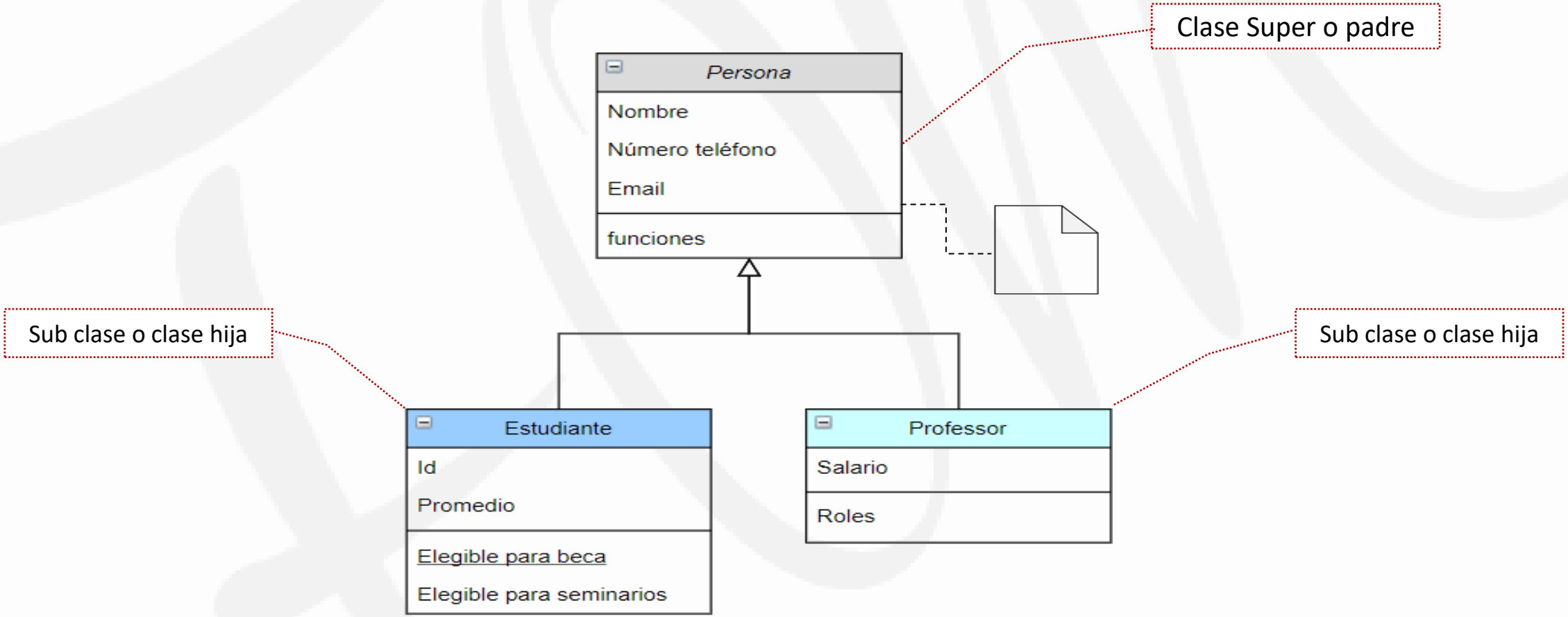
Modelamiento

Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

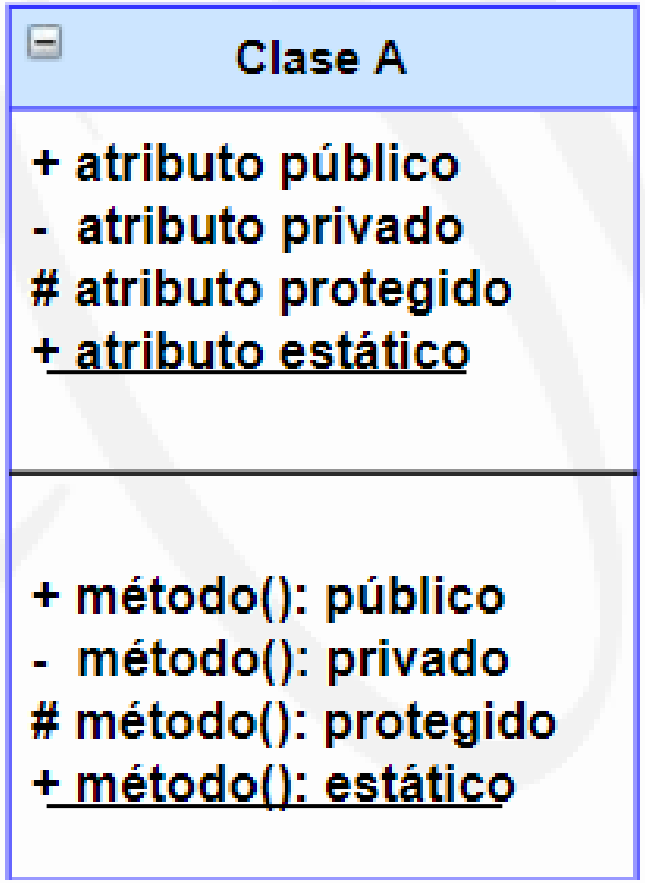
(Unified Modeling Language)



Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

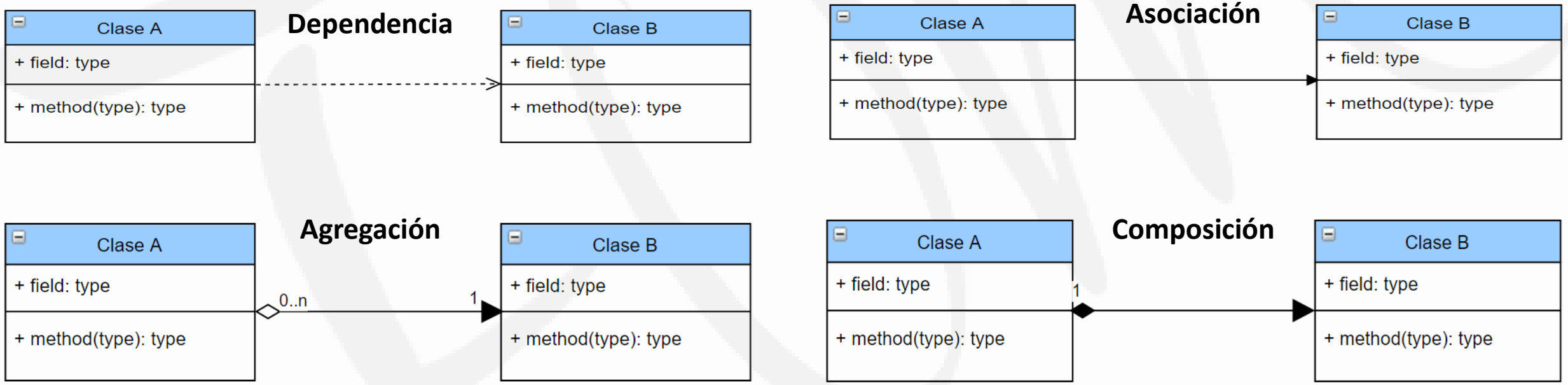
(Unified Modeling Language)



Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

(Unified Modeling Language)

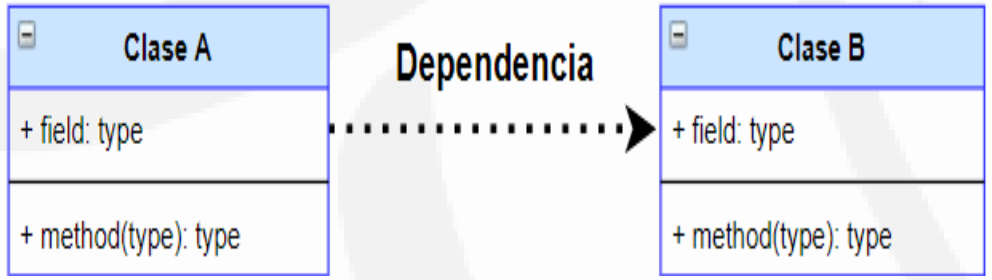


Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

(Unified Modeling Language)

Una clase incluye atributos o métodos de otra clase



Las clases son independientes, pero en algún momento la clase A necesita un objeto de la clase B.

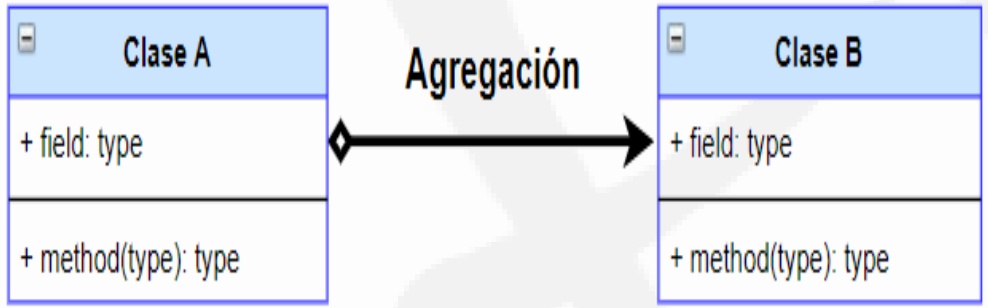
Ejemplo:

La **clase A Ventas** necesita el atributo forma_de_pago de la **clase B**

Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

(Unified Modeling Language)



Las clases son independientes entre si. La clase A necesita uno o más objetos de la clase B. La relación es unidireccional

Ejemplo:

La **clase Línea** necesita varios objetos **punto** de la **clase B**

El objeto **punto** existe por si sólo

Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

(Unified Modeling Language)

La clase A necesita uno o más objetos de la clase B y la clase B puede usar uno o más objetos de la clase A. La relación puede ser bidireccional

Ejemplo:

La relación profesor - alumno

Bidireccional

Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

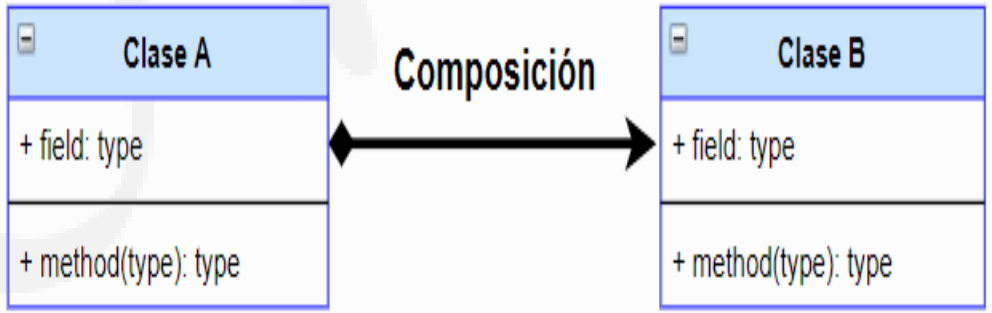
(Unified Modeling Language)

Un objeto está compuesto de otro objeto. Son dependientes: la clase B no puede existir sin el objeto que lo contiene.

Ejemplo:

Carro – Motor

Hotel - Habitación

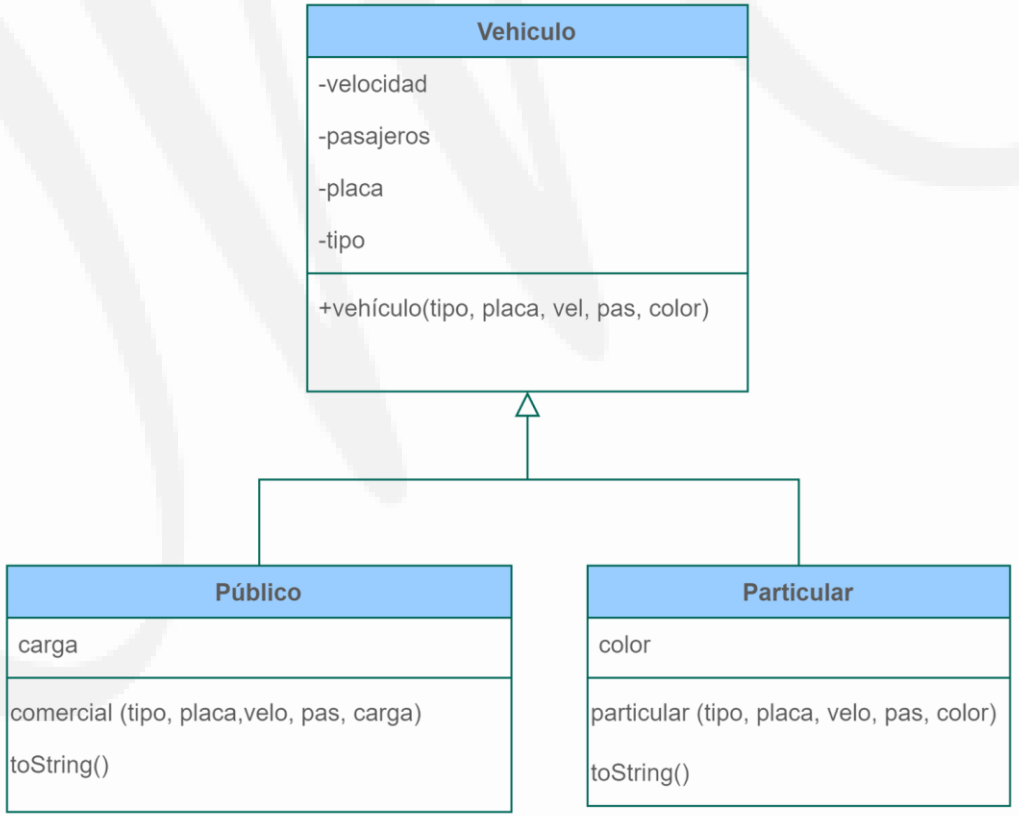


Ing. Fernando Ospina Marín
fospinam@gmail.com

Diagrama de clases

(Unified Modeling Language)

La generalización o especialización se refiere a la Herencia



Ing. Fernando Ospina Marín
fospinam@gmail.com

Patrones

"Un patrón de diseño (design pattern) es una **solución general y reutilizable para problemas típicos y recurrentes.**

Las reglas de diseño se basan en patrones (patterns).

Uno de los más difundidos es MVC (Modelo Vista Controlador)

Ing. Fernando Ospina Marín
fospinam@gmail.com

Patrones

Patrones creacionales

Sirven para solucionar problemas derivados de la creación de nuevos objetos, los más conocidos son:

- Abstract Factory
- Builder
- Factory Method
- Lazy initialization
- Prototype
- Singleton

Patrones estructurales

Sirven para ensamblar objetos y clases en estructuras más grandes, manteniendo la flexibilidad y eficiencia de la estructura:

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Ing. Fernando Ospina Marín
fospinam@gmail.com

Patrones

Otros Patrones o Frameworks

Comportamiento: 1. Incrementales 2.fuentes de datos

- Command
- Iterator
- Mediator
- Memento
- Observer
- State

- React
- Vue
- Django
- Flask
- Bootstrap

Ing. Fernando Ospina Marín
fospinam@gmail.com