Diagramas de clases y patrones de diseño

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(Información del tutor**

**Kevin Zeledón, 8487-6944,** [**magneticok@gmail.com**](mailto:magneticok@gmail.com)**)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Etapa de análisis (Entender que es lo que quiero/quieren)

Etapa de modelaje/Diseño (Proponer una serie de soluciones para el problema)

Etapa de implementación (Programar el código)

# UML (Unified modeling language)

Se utiliza durante todo el ciclo de vida de desarrollo de software (SDLS). Pero se utiliza principalmente en la etapa de modelaje

Es una notación formal que utiliza la gente que desarrolla software.

## ¿para qué modelar software?

* Comunicar

Brinda la estructura y el comportamiento deseado del sistema que se va a construir

* Visualizar y controlar

Manejar la arquitectura del sistema en desarrollo (darle seguimiento, y manejar los cambios)

* Entender

Cuando se modela se entiende tanto el problema a resolver como la manera que se está implementando (Entender problema y solución(sistema))

* Gestionar Riegos

Identificar, analizar y prevenir (evitar) o aceptar riesgos (con la idea de mitigar) o delegar (asignar dicho riego a un tercero) el riesgo

No modelar es un riego que se podría correr para cosas pequeñas

Un modelo permite crear una abstracción, es decir representar algo que existe en la vida real. También puede ser considerado como una simplificación.

## Diagramas del UML

### Diagrama de clases (Vista estática)

Estático: porque no se puede ver el programa en tiempo de ejecución. Permite ver la solución al problema.

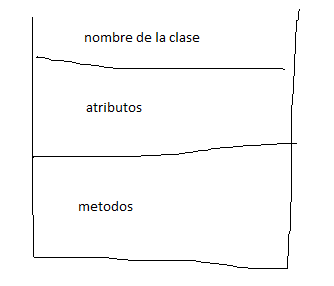
Incluye todas las estructuras de datos y todas las operaciones para solucionar el problema.

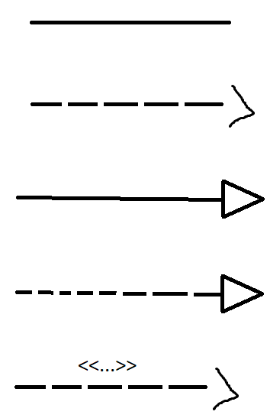
Incluye todas las clases que requiere la implementación de la solución.

Tiene clases con nombre, métodos y atributos, etc…

(modelo dominio: entender el problema, es más general, solo incluyen objetos que forman parte del problema)

Representación de una clase:



En un diagrama de clases, existen 5 tipos de unión:

Asociación

Dependencia

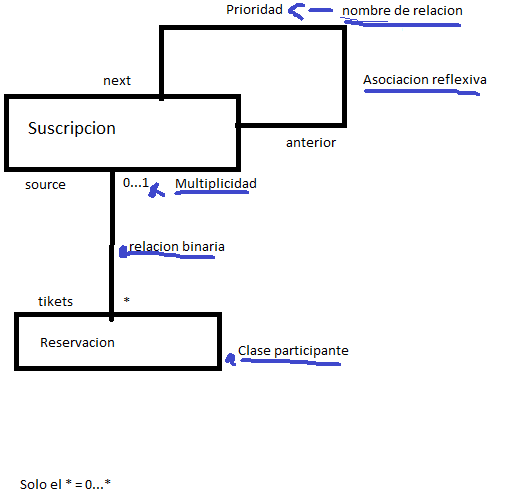
Generalización (de quien heredo)

Realización

(es como dependencia, pero indirecta (algo como sobreescritura)

Uso (explica uso)

# Tipos de relaciones de asociación



La multiplicidad indica la cantidad posible de cada objeto.

La prioridad es un verbo simple que describe la acción que se realiza

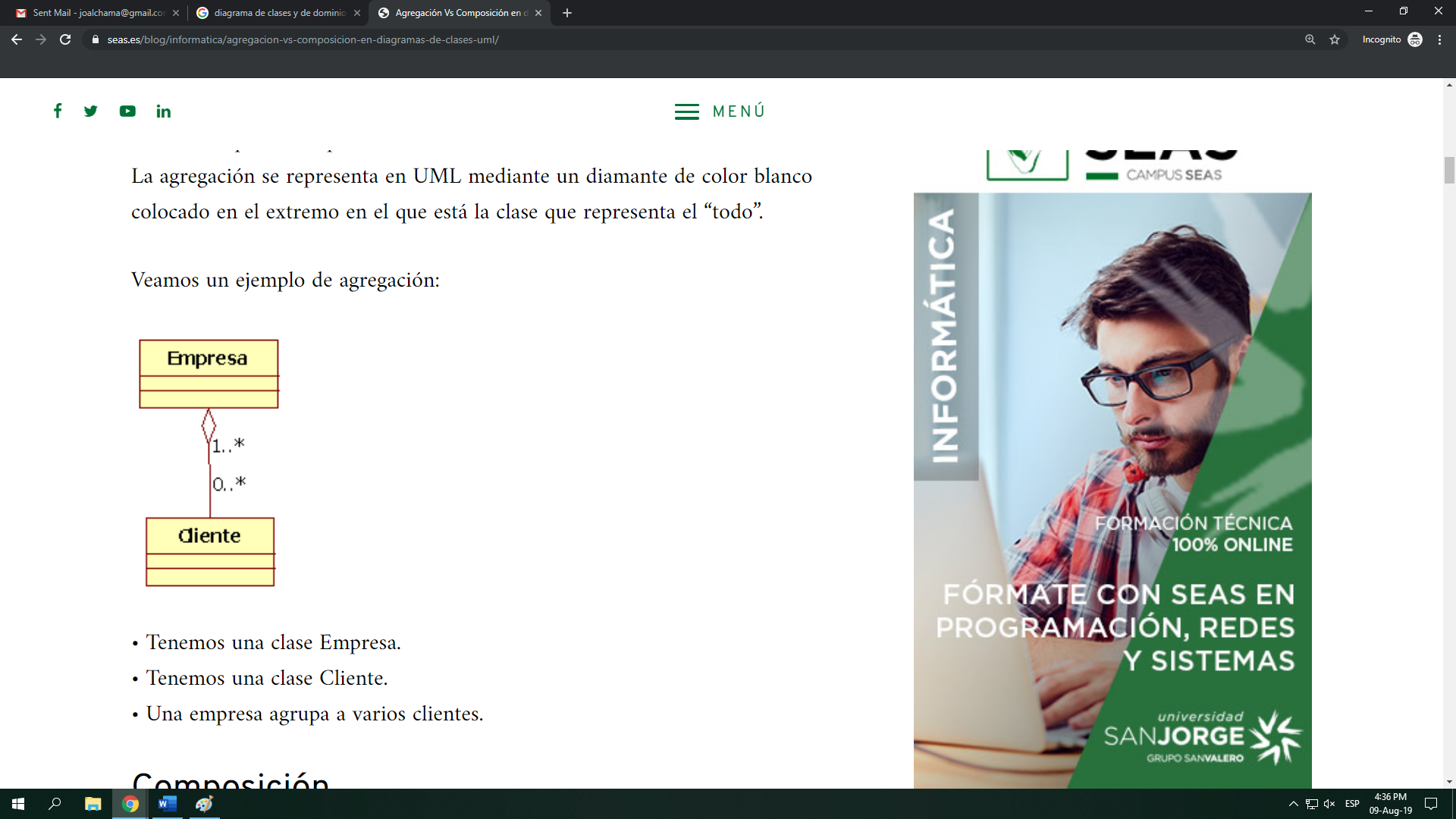
(ver imagen)

# Composición y agregación

Tienen un contenedor y los contenidos

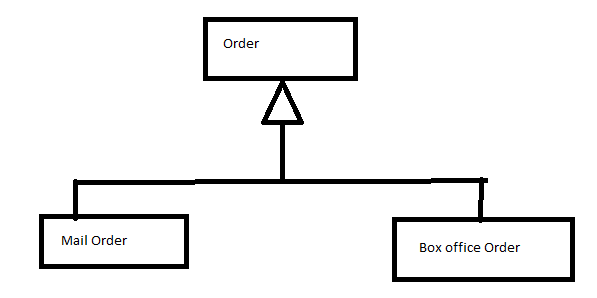
Agregación: relación débil. Si el contenedor deja de existir los contenidos pueden seguir existiendo

Composición: relación fuerte. Si el contenedor deja de existir los contenidos NO pueden seguir existiendo





# Relaciones de generalización



Es una relación de herencia

# Realización

(Buscar en internet)

El tipo bloque de selección depende de lo que voy a pedir. (ejemplo de radio botón o check box)

# Foward engineering (normal) y reverse engineering

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*Programas para crear diagramas de clase\*

Microsoft Visio

Lucid Chart

IBM rational software architect

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_