



Ciclo 2

Semana 2

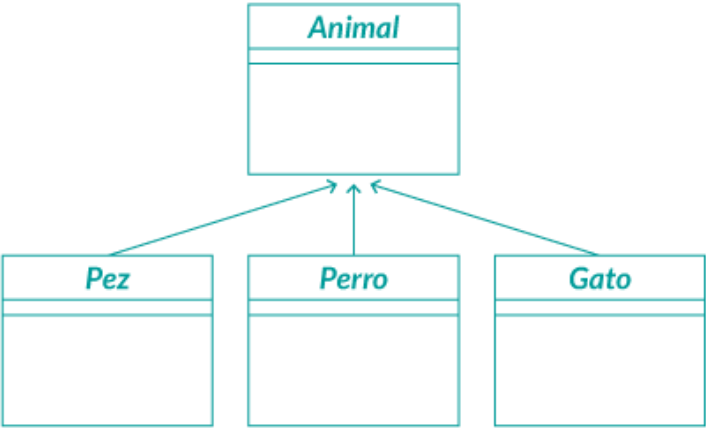
Modelamiento y Modelo Vista Controlador (MVC)

Lectura 3 - Diagrama de clases

| Diagrama de clases



Es el más ampliamente utilizado de los tipos de diagramas en el modelo UML. Está orientado a los lenguajes orientados a objetos, de modo que con este diagrama se pretende representar las clases de una aplicación, así como los métodos y atributos asociadas a estas, y lógicamente, la manera en que las clases se relacionan entre sí, simbolizadas por las flechas. Cada objeto lleva su nombre encima, a continuación, los atributos de esa clase y por último los métodos que contiene. Por convención estas constantes siempre van en mayúsculas y si el nombre de la constante tiene varias palabras, se separa con el carácter "_".





Definiciones del modelo UML en el diagrama de clases

Lo primero que debe hacer el programador es identificar las entidades del mundo, para luego documentar sus características. Luego se establecen las relaciones que existen entre estas entidades y finalmente determina la información adicional como reglas, restricciones, y demás que se tenga sobre las entidades.

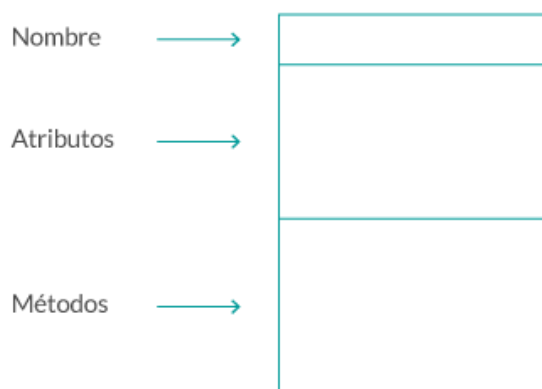
1. Identificar entidades y sus características

En esta primera etapa el objetivo es lograr identificar los elementos o entidades que intervienen en el mundo del problema, los cuales pueden clasificarse en concretos o abstractos: concretos aquellos tangibles como una casa, un animal, etc., o abstractos como puede ser una cuenta bancaria, una figura geométrica, etc. Estas entidades independientes de su clasificación se llaman clases en la Programación Orientada A Objetos y se establecen en un modelo UML así:

- **Clase:**

La clase en resumen es el elemento que define los atributos y el comportamiento que un objeto podría tener y puede contener definiciones de valores etiquetados de restricciones y estereotipos.

Se representa con un rectángulo dividido en tres áreas



En la primera sección se establece el **nombre** de la clase. Acepta cadena de caracteres por ejemplo Empleado y por convención este nombre comienza por mayúsculas. Si el nombre es compuesto como por ejemplo una cuenta bancaria, por

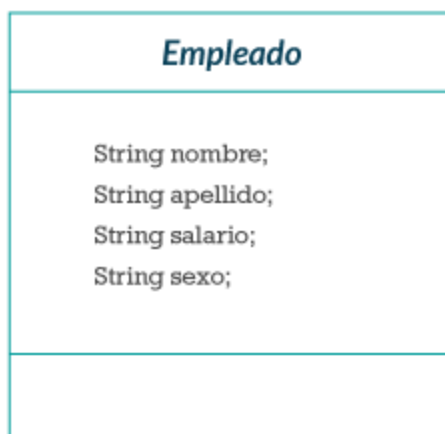


Semana 2

Modelamiento y Modelo
Vista Controlador (MVC)

convención se coloca una letra mayúscula al inicio cada palabra así CuentaBancaria, o todo en mayúscula si corresponde a una sigla. En Java no se pueden usar tildes en los nombres de las clases. Es muy importante que el nombre que se defina no solo sea claro sino significativos para que sea de fácil entendimiento.

En la segunda sección se listan los **atributos** o características de la entidad. A cada atributo se le debe asignar un nombre que sea significativo y debe ser una cadena de caracteres, iniciando con una letra minúscula, sin espacios en blanco y sin tildes. Por convención, el nombre de los atributos comienza por una letra. Si el nombre es compuesto, se inicia cada palabra simple con letra mayúscula. Los nombres de los atributos dentro de una clase deben ser únicos por tanto no se puede dar el mismo nombre a dos atributos. Luego del nombre se coloca la descripción del conjunto de valores que puede tomar, ya sean cadenas de caracteres, enteros, etc.



En la tercera sección se definen los algoritmos de la entidad llamados métodos los cuales tienen como objetivo resolver un problema puntual, dentro del contexto planteado. Estos métodos también pueden ser servicios que prestar una clase a si misma o a las otras clases del modelo. El encabezado del método se denomina como signatura. Primero se establece un nombre claro del método, luego se define la lista valores o parámetros y su tipo de dato, que se necesitan para resolver el problema.

Estos parámetros se definen sabiendo la información que necesita para dar solución y que no tiene el objeto. Si no hay una respuesta, se indica el tipo void. Una clase puede tener entre cinco y veinte métodos cada uno que puede resolver un tema puntual de la clase y que está relacionado con la información de la clase.



Semana 2

Modelamiento y Modelo
Vista Controlador (MVC)

La notación que antecede el nombre del atributo u operación define el tipo de visibilidad del elemento:

- símbolo + si el atributo y la operación público de visibilidad
- símbolo - el atributo u operación tienen un nivel privado de visibilidad.
- símbolo # define una operación o atributo como protegido
- símbolo ~ indica la visibilidad del paquete.

2. Relaciones Entre Entidades



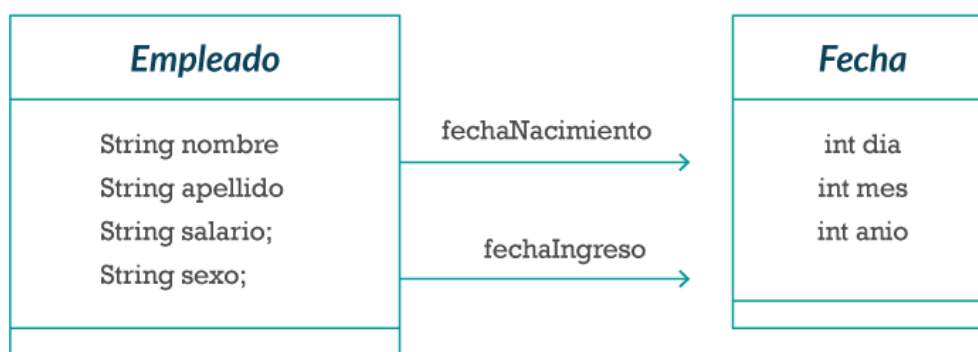
En esta etapa se identifican las relaciones o **asociaciones** que hay entre las entidades establecidas, y que se pueden describir como una característica que tiene una entidad pero que su valor está representado por otra entidad.



Semana 2

Modelamiento y Modelo
Vista Controlador (MVC)

Se debe asignar un nombre a esta relación que sigue las mismas convenciones establecidas para los nombres de los atributos. El nombre debe reflejar como una clase usa a otra como parte de sus características. Se pueden encontrar varias relaciones entre dos clases, de ahí la importancia de encoger muy bien el nombre de cada asociación. Se debe tener en cuenta que dirección de la flecha indica la entidad que "contiene" a la otra.



En el modelo UML estas asociaciones se representan con una flecha que une las cajas de las clases y según el tipo de flecha se representa el tipo de asociación. Una vez se tenga el diagrama de clases representando sus entidades y las relaciones entre ellas se dice que se tiene un **modelo conceptual**, que describe la estructura y las relaciones de los elementos en el mundo del problema.

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/Implementation	
Dependency	
Aggregation	
Composition	

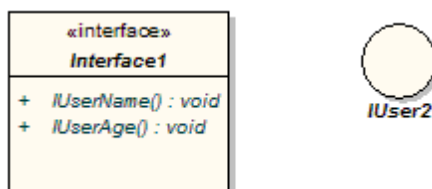


3. Información adicional

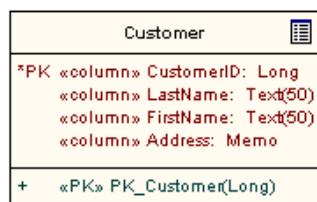
Finalmente se determina la información adicional como reglas, restricciones, y demás que se tenga sobre las entidades.

Existen otros elementos que se especifica en el diagrama de clases, pero se especificaran a detalle más adelante.

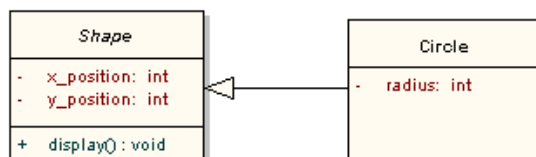
- **Interface:** Es la parte de la solución que permite que el usuario interactúe con el programa y se dibuja de manera similar al de una clase, o también como un círculo con ninguna operación detallada.



- **Tablas:** es una clase estereotipada y se dibuja con icono pequeño en la esquina superior derecha. Los atributos son «columnas» estereotipadas.

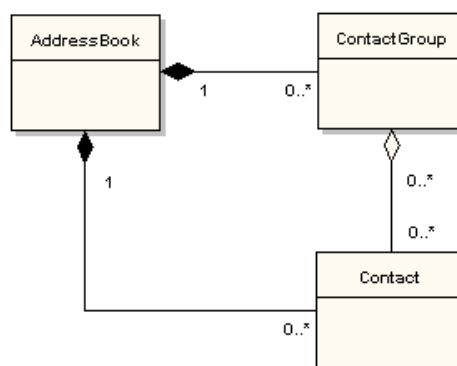


- **Generalizaciones:** Usada para indicar herencia. Se diagrama desde un clasificador específico a uno general, donde el origen hereda las características del destino.

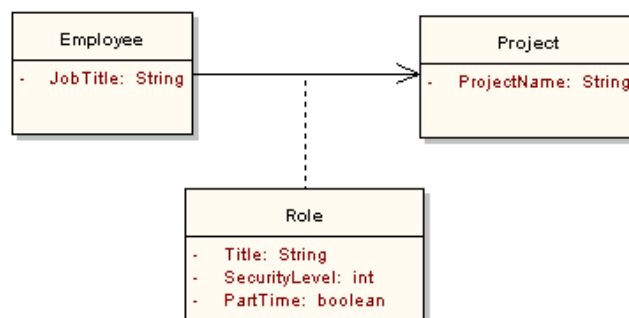




- **Agregaciones:** Las agregaciones describen elementos compuestos de otros más pequeños. Estas relaciones se definen con una punta de flecha con forma de diamante apuntando hacia el destino o clase padre.



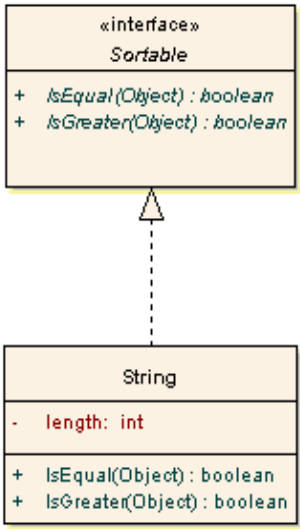
- **Clase Asociación:** Estructura que permite una conexión de asociación para tener conexiones y atributos.



- **Dependencias:** modela el rango de relaciones dependientes entre los elementos del modelo.
- **Trazado:** es una especialización de una dependencia, que vincula elementos a través de los modelos. Son usados normalmente rastrear cambios de requisitos y cambios en el modelo.
- **Realizaciones:** expresar la trazabilidad e integridad en el modelo y se muestra con una línea de trazos con una punta de flecha sólida y el estereotipo.

Semana 2

Modelamiento y Modelo Vista Controlador (MVC)



- **Anidamientos:** conector que muestra como un elemento se anida dentro del elemento destino.

