



INSTITUTO
KHIPU

Semestre III

Sesión 09

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**ANÁLISIS Y DISEÑO
DE SISTEMAS**

Tema:

**DIAGRAMA DE CLASES – CLASES
Y ATRIBUTOS.**

DIAGRAMA DE CLASES – CLASES Y ATRIBUTOS.

Definición de Diagrama de Clase | ¿Qué es un Diagrama de Clase?

Un diagrama de clase es un tipo de diagrama UML que describe un sistema visualizando los diferentes tipos de objetos dentro de un sistema y los tipos de relaciones estáticas que existen entre ellos. También ilustra las operaciones y atributos de las clases.

Suelen utilizarse para explorar los conceptos de dominio, comprender los requisitos de los programas informáticos y describir diseños detallados.

Notaciones del diagrama de clases con ejemplos

Hay varias notaciones de diagramas de clase que se utilizan cuando se dibujan diagramas de clase UML. Hemos enumerado a continuación las notaciones más comunes de diagramas de clase.

Las clases son el elemento principal del diagrama y representa, como su nombre indica, una clase dentro del paradigma de la orientación a objetos. Este tipo de elementos normalmente se utilizan para representar conceptos o entidades del «negocio». Una clase define un **grupo de objetos** que comparten características, condiciones y significado. La manera más rápida para encontrar clases sobre un enunciado, sobre una idea de negocio o, en general, sobre un tema concreto es buscar los **sustantivos** que aparecen en el mismo. Por poner algún ejemplo, algunas clases podrían ser: Animal, Persona, Mensaje, Expediente... Es un concepto muy amplio y **resulta fundamental identificar de forma efectiva estas clases**, en caso de no hacerlo correctamente se obtendrán una serie de problemas en etapas posteriores, teniendo que volver a hacer el análisis y perdiendo parte o todo el trabajo que se ha hecho hasta ese momento.

Bajando de nivel una clase está compuesta por tres elementos: **nombre de la clase, atributos, funciones. Estos elementos se incluyen en la representación** (o no, dependiendo del nivel de análisis).



Para representar la clase con estos elementos se utiliza una caja que es dividida en tres zonas utilizando para ello líneas horizontales:

- La primera de ellas se utiliza para el **nombre de la clase**. En caso de que la clase sea abstracta se utilizará su nombre en cursiva.
- La segunda, por otra parte, se utiliza para escribir los **atributos** de la clase, uno por línea y utilizando el siguiente formato:

visibilidad nombre_atributo : tipo = valor-inicial { propiedades }

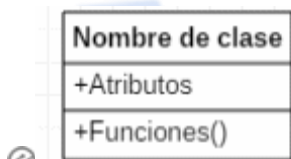


Aunque esta es la forma «oficial» de escribirlas, es común simplificar únicamente poniendo el nombre y el tipo o únicamente el nombre.

- La última de las zonas incluye cada una de las **funciones** que ofrece la clase. De forma parecida a los atributos, sigue el siguiente formato:

visibilidad nombre_funcion { parametros } : tipo-devuelto { propiedades }

De la misma manera que con los atributos, se suele simplificar indicando únicamente el nombre de la función y, en ocasiones, el tipo devuelto.



Notación de una clase

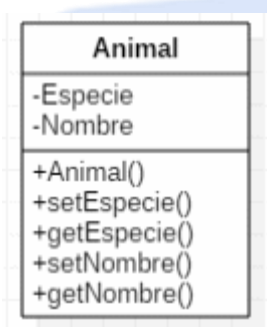
Tanto los atributos como las funciones incluyen al principio de su descripción la visibilidad que tendrá. Esta visibilidad se identifica escribiendo un símbolo y podrá ser:

- **(+) Pública.** Representa que se puede acceder al atributo o función desde cualquier lugar de la aplicación.
- **(-) Privada.** Representa que se puede acceder al atributo o función únicamente desde la misma clase.

- **(#) Protegida.** Representa que el atributo o función puede ser accedida únicamente desde la misma clase o desde las clases que hereden de ella (clases derivadas).

Estos tres tipos de visibilidad son los más comunes. No obstante, pueden incluirse otros en base al lenguaje de programación que se esté usando (no es muy común). Por ejemplo: (/) Derivado o (~) Paquete.

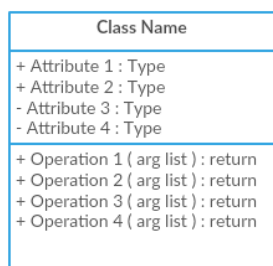
Un ejemplo de clase podría ser el siguiente:



Ejemplo de una clase

En caso de que un atributo o función sea estático, se representa en el diagrama subrayando su nombre. Una característica estática se define como aquella que es compartida por cada clase y no instanciada para cada uno de los objetos de esa clase. Es un concepto muy común.

Clase



Las clases representan los objetos centrales de un sistema. Está representado por un rectángulo con hasta 3 compartimentos.

El primero muestra el nombre de la clase, mientras que el segundo muestra los atributos de la clase que son las características de los objetos. El último enumera las operaciones de la clase, lo que representa el comportamiento de la clase.



Class Name

Clase simple

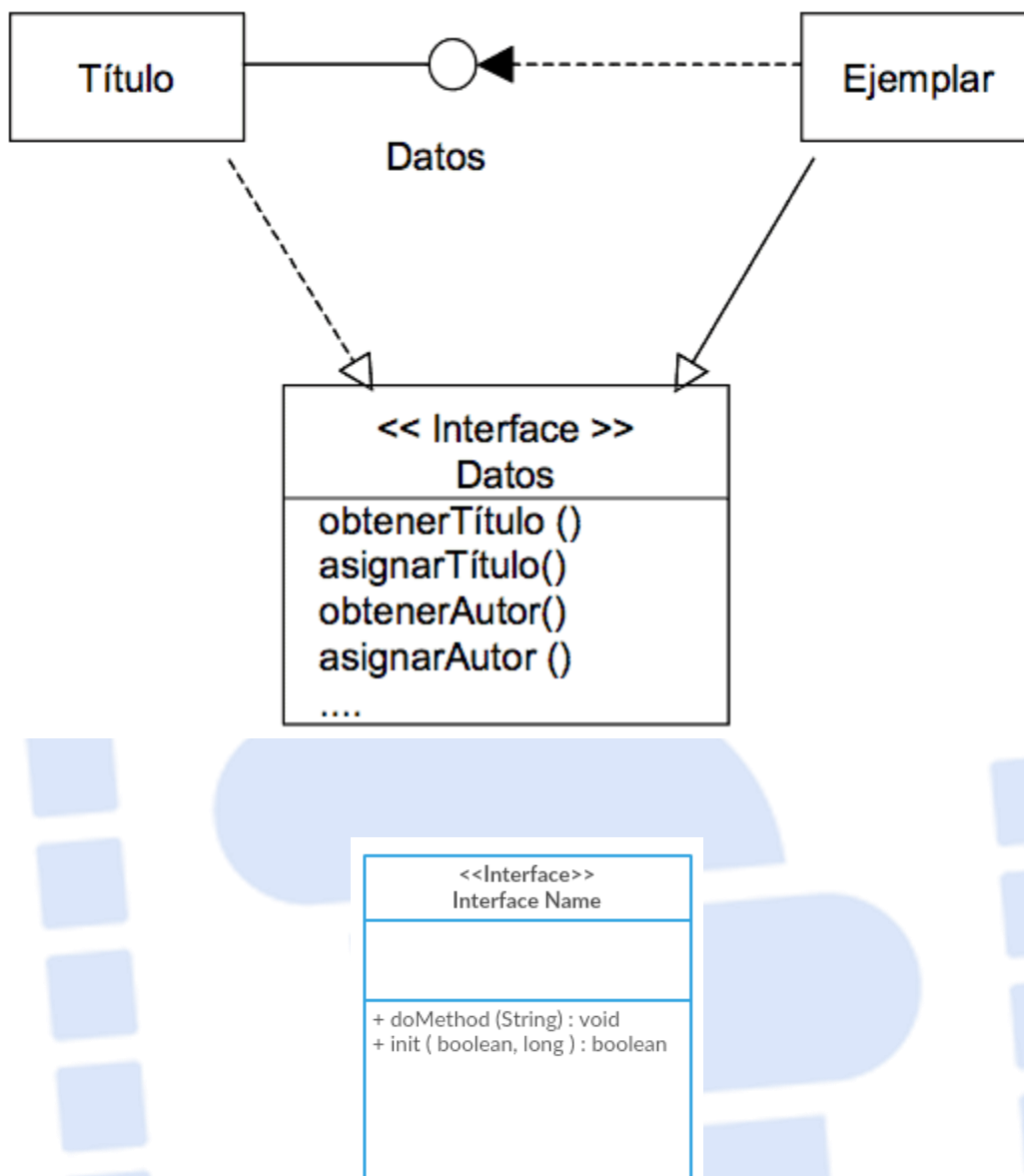
Clase simple dos últimos compartimentos son opcionales. La notación de clase sin los dos últimos compartimentos se llama clase simple y sólo contiene el nombre de la clase.

Interfaz del diagrama de clase

Una interfaz se representa como una caja con compartimentos, igual que las clases. En la zona superior se incluye el nombre y el estereotipo `<>`. La lista de operaciones se coloca en la zona inferior, igual que en las representaciones de clases. La zona en la que se listan los atributos estará vacía o puede omitirse.

Existe una representación más simple para la interfaz: un círculo pequeño asociado a una clase con el nombre de la interfaz debajo. Las operaciones de la interfaz no aparecen en esta representación; si se quiere que aparezcan, debe usarse la primera notación.

Entre una clase que implementa las operaciones que una interfaz ofrece y esa interfaz se establece una relación de realización que, dependiendo de la notación elegida, se representará con una línea continua entre ellas cuando la interfaz se representa como un círculo y con una flecha hueca discontinua apuntando a la interfaz cuando se represente como una clase.



El símbolo de la interfaz en los diagramas de clase indica un conjunto de operaciones que detallarían la responsabilidad de una clase.

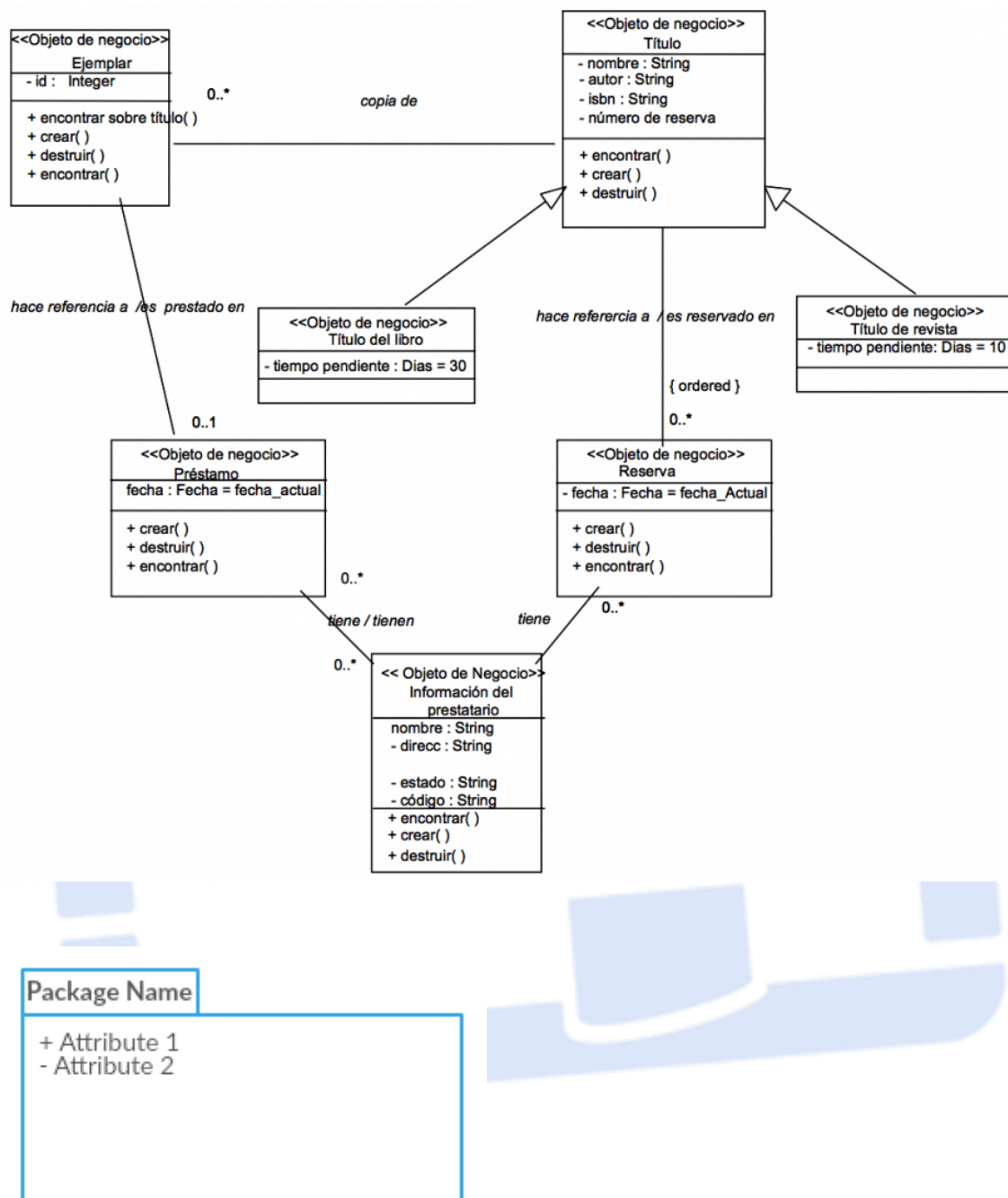
Paquete

Los paquetes se representan mediante un icono con forma de carpeta y las dependencias con flechas discontinuas entre los paquetes dependientes (ver [Diagrama de paquetes](#)).

Ejemplo

Estudio del sistema encargado de la gestión de préstamos y reservas de libros y revistas de una biblioteca. Dependiendo del momento del desarrollo el diagrama estará más o

menos detallado. Así, el diagrama tendría la siguiente estructura en el proceso de análisis:



El símbolo del paquete se utiliza para agrupar clases o interfaces de naturaleza similar o relacionadas. La agrupación de estos elementos de diseño utilizando los símbolos del paquete mejora la legibilidad del diagrama

Diagrama de clases (Restaurante)

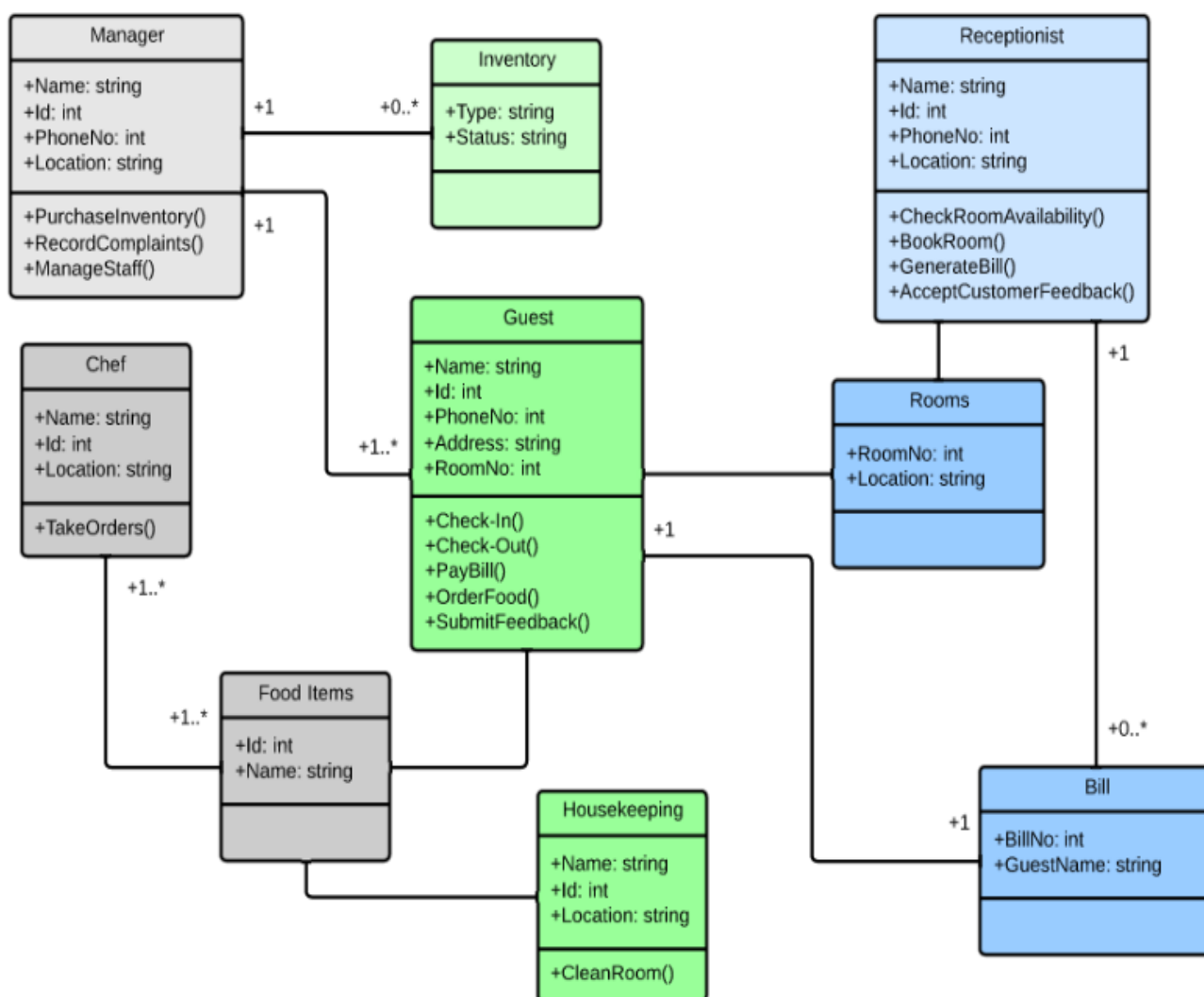
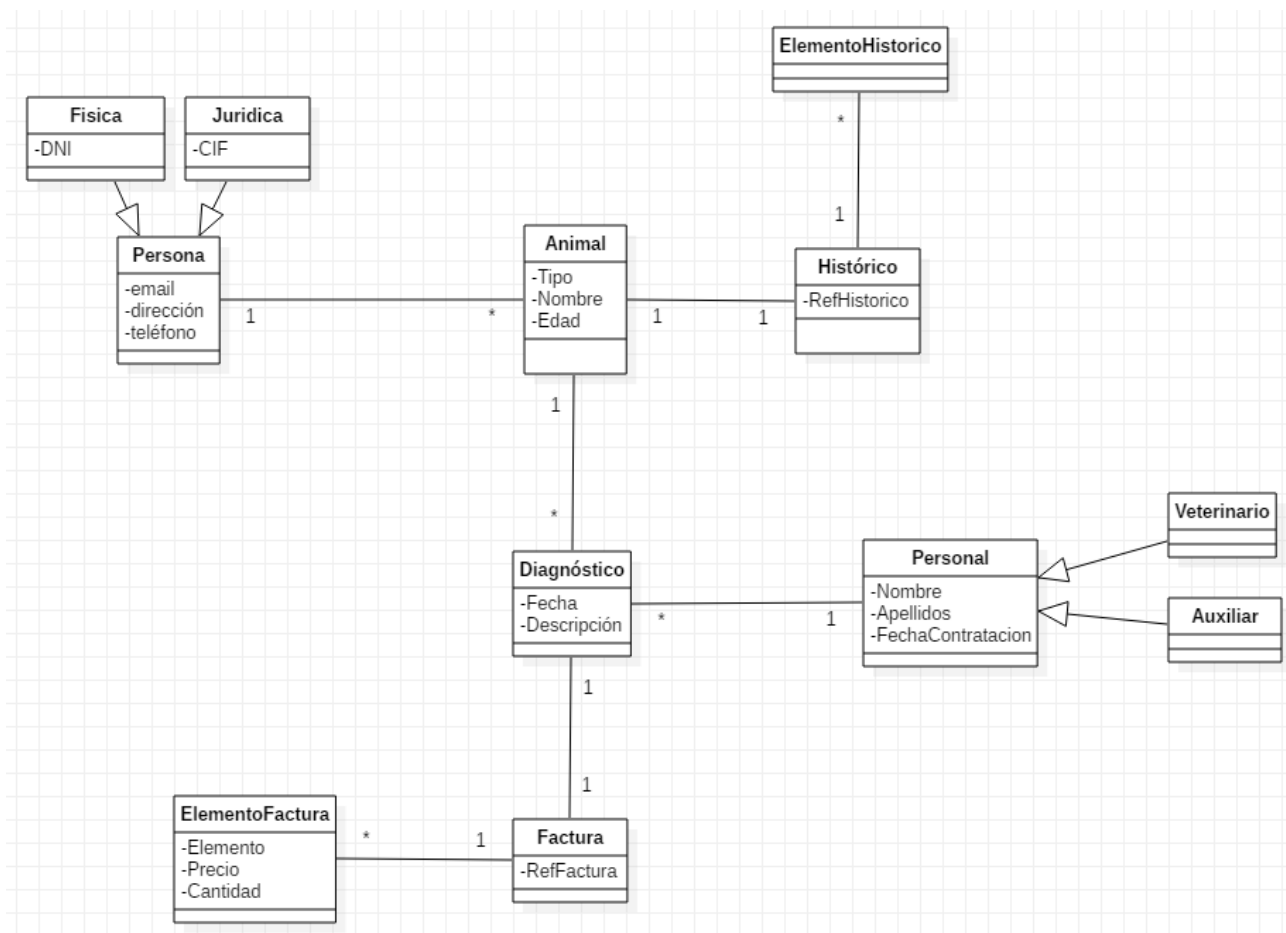


Diagrama de Clases (veterinaria)



FUENTE:

- <https://www.monografias.com/trabajos94/analisis-y-diseno-sistemas-informacion/analisis-y-diseno-sistemas-informacion>
- <https://diagramasuml.com/diagrama-de-clases/>
- <https://repositorio.grial.eu/bitstream/grial/353/1/DClase.pdf>
- <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
- <https://diagramasuml.com/diagrama-de-clases/>
- <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/>
- <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-clases/>
- <http://sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto.pdf>
- BURCH, John; GRUDNISKY, Gary. "Diseño de Sistemas de Información", Grupo Noriega editores.
- SENN, James A. "Análisis y diseño de sistemas de información", 2da. ed., McGraw-Hill.



INSTITUTO
KHIPU