

ANALISIS Y DISEÑO DE SISTEMAS – UML PUDS.

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

# Diagrama de clases

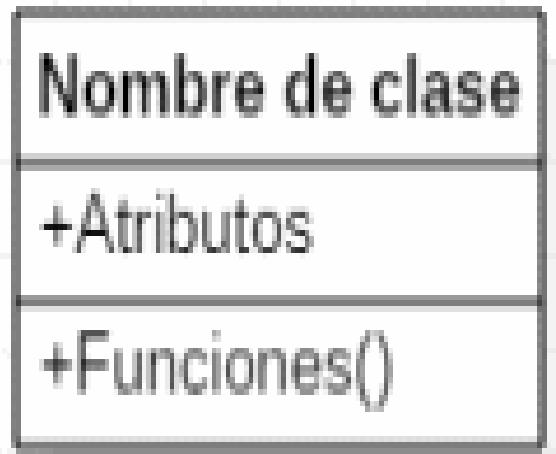
- El **diagrama de clases** es uno de los diagramas incluidos en UML 2.5 clasificado dentro de los diagramas de estructura y, como tal, se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático.
- El diagrama de clases es un **diagrama puramente orientado al modelo de programación orientado a objetos**, ya que define las clases que se utilizarán cuando se pase a la fase de construcción y la manera en que se relacionan las mismas. Se podría equiparar, salvando las distancias, al famoso diagrama de modelo Entidad-Relación (E/R), no recogido en UML, tiene una utilidad similar: la representación de datos y su interacción. Ambos diagramas muestran el modelo lógico de los datos de un sistema.

# ELEMENTOS DE UN DIAGRAMA DE CLASES

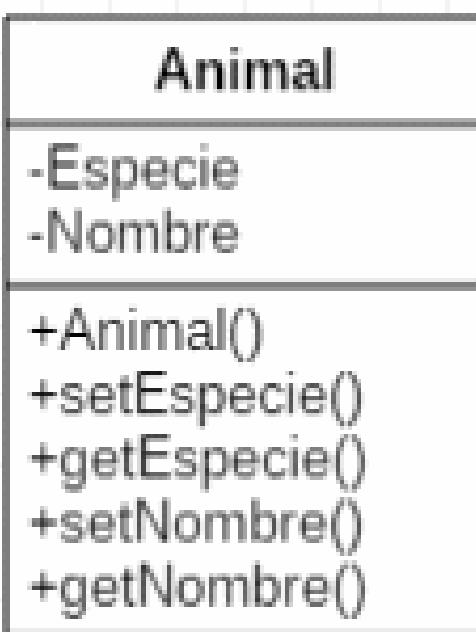
## \* CLASES

- 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 simplificando 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 }*

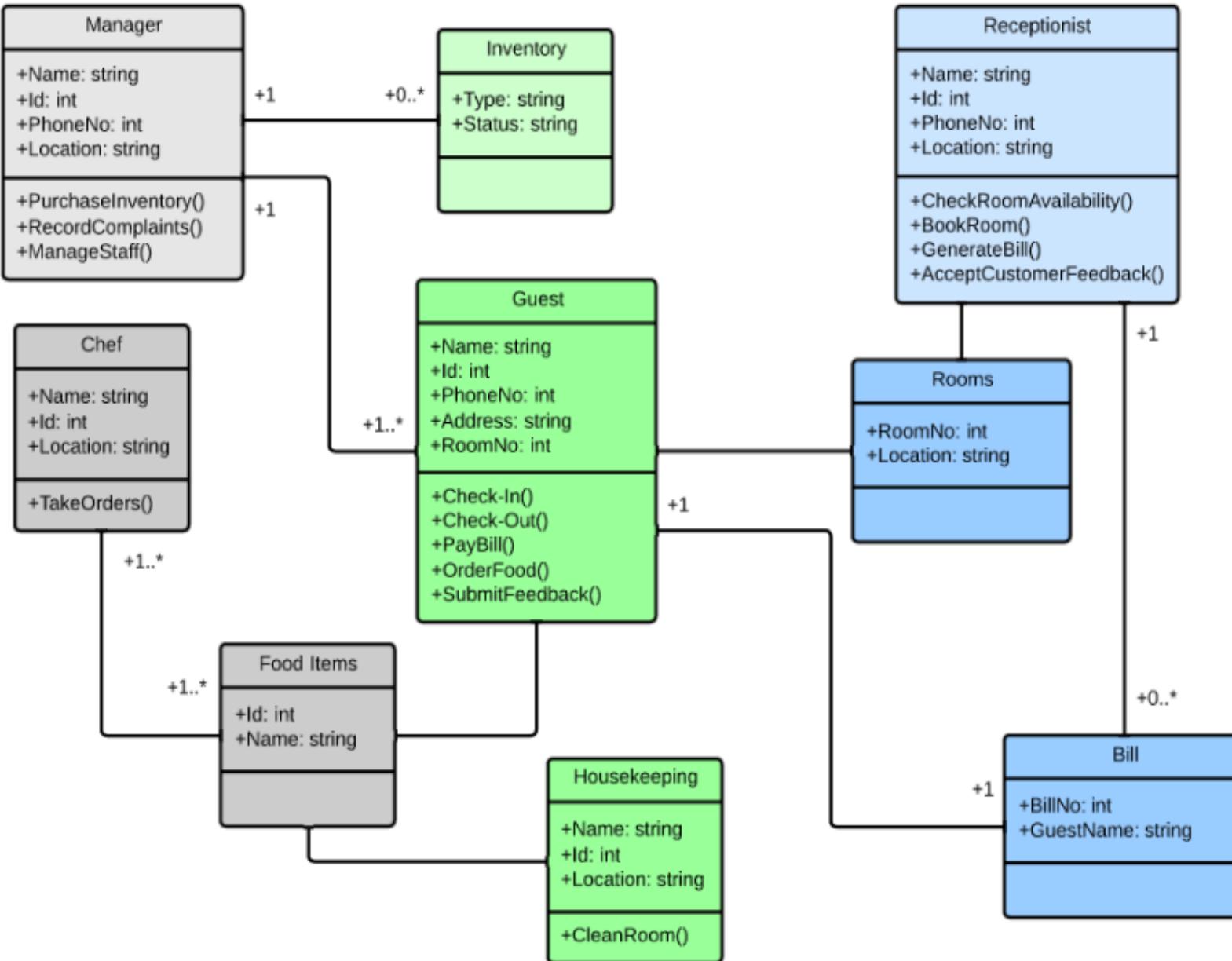


- Tanto los atributos como las funciones incluyen al principio de su descripción la visibilidad que tendrá. Esta visibilidad se identifica escribiendo un símbolo y podrá ser:
- **(+)** **Pública.** Representa que se puede acceder al atributo o función desde cualquier lugar de la aplicación.
- **(-)** **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.

# Ejemplo: diagrama de clases



# 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>