



El futuro digital  
es de todos

MinTIC

# Unidad 1

03 - Fundamentos de la  
programación orientada a  
objetos - POO





# Qué es programación orientada a objetos - POO?

- La programación orientada a objetos es una evolución lógica de la programación estructurada, en la que el concepto de variables locales a un procedimiento o función, que no son accesibles a otros procedimientos y funciones, se hace extensible a los propios subprogramas que acceden a estas variables.
- En la programación orientada a objetos, se definen **objetos** que conforman una aplicación.
- Estos objetos no están aislados en la aplicación, sino que se comunican entre ellos.





# Ventajas y desventajas

## Ventajas

- **Reusabilidad.** Se pueden usar en distintas partes del programa y en numerosos proyectos.
- **Mantenibilidad.** Son más sencillos de leer y comprender.
- **Modificabilidad.** Facilidad de añadir, suprimir o modificar nuevos objetos.
- **Fiabilidad.** Podemos probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que puedan surgir.

## Desventajas

- Cambio en la forma de pensar de la programación tradicional a la orientada a objetos.
- La ejecución de programas orientados a objetos es más lenta.
- La necesidad de utilizar bibliotecas de clases obliga a su aprendizaje y entrenamiento.





# Lenguajes Orientados a Objetos

- Simula (1967)
- Smalltalk (1980)
- Objective-C (1982)
- C++ (1983)
- Object Pascal (1988)
- Lisp CLOS (1989)
- Python (1991)
- Java (1995)
- C# (2000)
- Kotlin (2011)

Muchos de estos lenguajes no son puramente orientado a objetos, sino que son híbridos, que combinan la POO con otros paradigmas.

**Simula** (1967) es aceptado como el primer lenguaje orientado a objeto y fue creado para hacer programas de simulación.

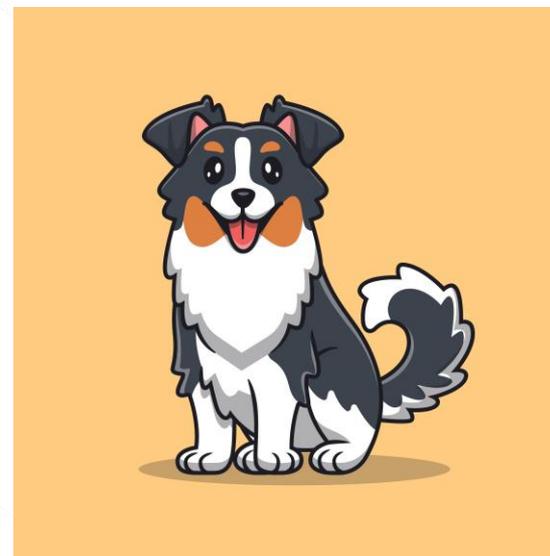
**Smalltalk** (1980) es el ejemplo canónico de un lenguaje puramente Orientado a Objeto.

[https://es.wikipedia.org/wiki/Historia\\_de\\_los\\_lenguajes\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programaci%C3%B3n)



# Qué es un Objeto?

- Un objeto es una persona, animal o cosa.
- Los objetos en POO son representaciones de entes físicos o abstractos que hacen parte del dominio del problema que se va a tratar.
  - Humano
  - Persona
- Los objetos están formados por una serie de *características* y *comportamientos* que se tienen los mismos.



- **Características:**
  - Raza
  - Color
  - Estatura
  - Edad
  - Etc.
- **Acciones posibles:**
  - Comer
  - Dormir
  - Sentarse
  - Ladrar
  - Etc.

<a href="https://www.freepik.es/fotos-vectores-gratis/perro">Vector de Perro creado por catalyststuff - www.freepik.es</a>

# Características del Objeto

- Las características son todos los datos que identifica, describe y representa el estado de un objeto en un momento determinado.
- En POO, las características son llamados *atributos* del objeto.
- Generalmente, los atributos se nombran con **sustantivos**.
  - color
  - marca
  - tamaño



# Comportamientos del Objeto

- Los comportamientos son la manera como actúan o reaccionan los objetos al momento de interactuar con él, dependiendo del valor en sus atributos o de los parámetros recibidos.
- En POO, los comportamientos son llamados *métodos* de objeto.
- Generalmente, los métodos se nombran con verbos.
  - avanzar
  - pausar
  - encender





# Ejemplos de Objetos



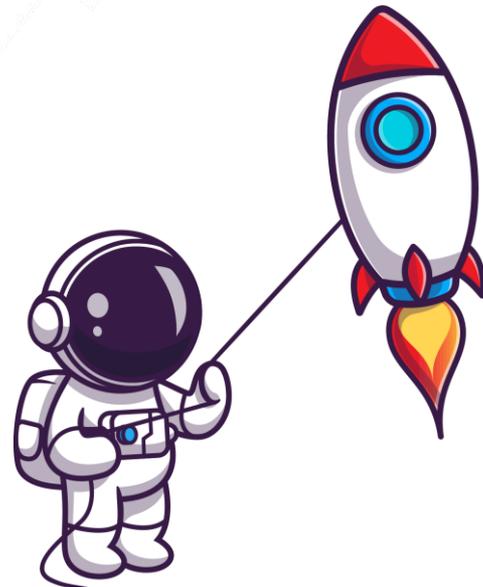
Objeto	Características	Comportamientos
Automóvil	Tipo de transmisión Fabricante Color	Girar Frenar Acelerar
León	Peso Color Hambriento o no Manso o salvaje	Rugir Dormir Cazar

Estos objetos podrían ser usados en juegos o programas interactivos para simular objetos del mundo real. Un ejemplo podría ser un objeto de software de un automóvil en un juego de carreras o objeto de software de un león en un software educativo interactivo de un zoológico para niños.



# La importancia de las palabras

- **Nombres:**
  - Objetos
  - Propiedades de objetos
- **Adjetivos:**
  - Valores de las propiedades
- **Verbos:**
  - Comportamiento de los objetos



“El **coche** tiene **color rojo** y se **mueve**”

“El **documento** tiene **letra grande** y se **muestra**”



# La importancia de las palabras

Una biblioteca contiene libros y revistas. Puede haber varias copias de un libro. Algunos de los libros son reservados sólo para préstamos a corto plazo. Todos los otros pueden ser prestados a cualquier miembro de la biblioteca por tres semanas. Los miembros de la biblioteca pueden normalmente solicitar hasta seis ítems de una vez, pero miembros del staff puede solicitar hasta doce ítems a la vez. Solamente miembros del staff pueden obtener prestado revistas.

El sistema debe conservar la pista de cuando los libros y revistas son prestados y retornados forzando las reglas de la biblioteca.





# La importancia de las palabras

Una **biblioteca** **contiene** **libros** y **revistas**. Puede haber varias **copias** de un **libro**. Algunos de los **libros** son **reservados** sólo para **préstamos a corto plazo**. Todos los otros pueden ser **prestados** a cualquier **miembro de la biblioteca** por **tres semanas**. Los **miembros de la biblioteca** pueden normalmente **solicitar** hasta **seis ítems** de una vez, pero **miembros del staff** puede **solicitar** hasta **doce ítems** a la vez. Solamente **miembros del staff** pueden **obtener** prestado **revistas**.

El **sistema** debe **conservar** la pista de cuando los **libros** y **revistas** son **prestados** y **retornados** forzando las **reglas** de la biblioteca.





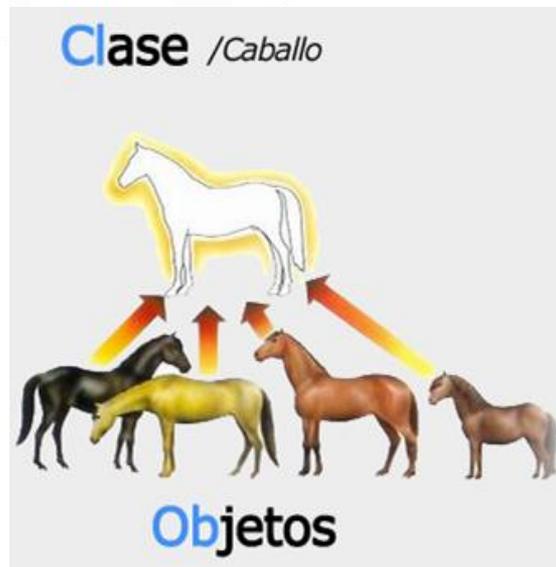
El futuro digital  
es de todos

MinTIC

# Conceptos Básicos de Programación Orientados a Objetos

# Qué es una Clase?

- Es la estructura fundamental en la programación orientada a objetos.
- Pueden ser pensadas como una **plantilla**, molde o prototipo de un objeto.
- En la definición de una clase, se especifican los **atributos** (con sus tipos de datos asociados) y los **métodos** que todos los objetos van a tener disponibles.



Un objeto es una **instancia** de una clase



# Clases y Objetos

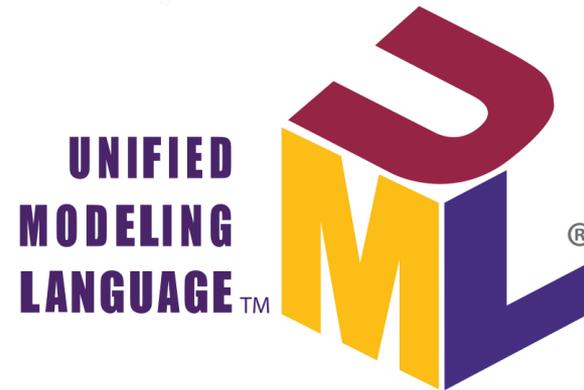
Clase Automóvil		Automóvil A	Automóvil B
Atributos	Placa	ABC111	XYZ123
	Color	Azul	Rojo
	Fabricante	Mitsubishi	Toyota
	Velocidad	50 Km/h	100 Km/h
Métodos	acelerar		
	girar		
	frenar		



# UML

## Lenguaje de Modelado Unificado

- Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).
- Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
- UML ofrece un estándar para describir un plano del sistema (modelo).



[https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)



# Diagramas en UML

## ● Estructurales

- Diagrama de clases
- Diagrama de componentes
- Diagrama de despliegue
- Diagrama de objetos
- Diagrama de paquetes
- Diagrama de perfiles
- Diagrama de estructura compuesta

## ● De comportamiento

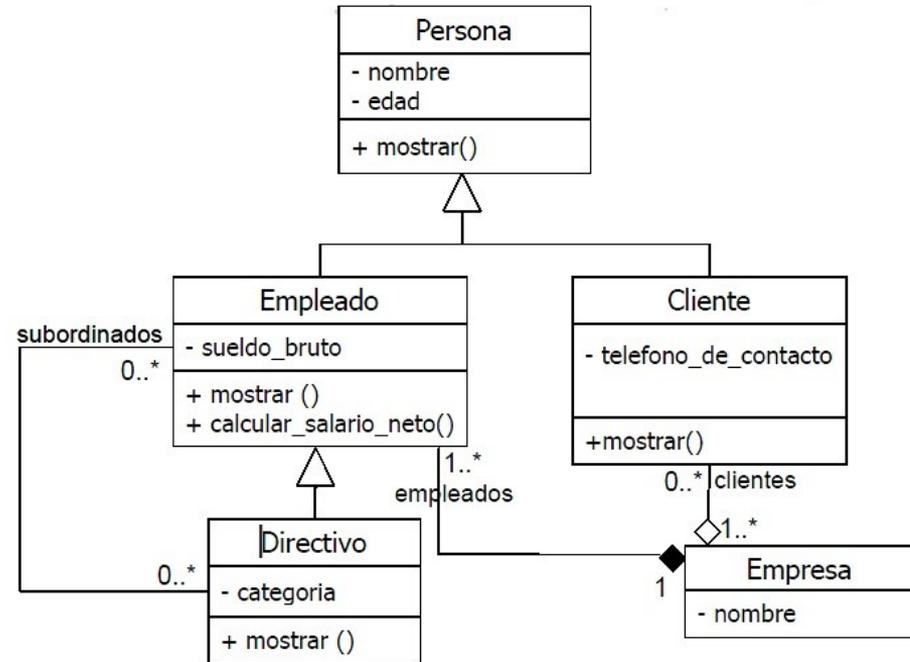
- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de máquina de estados
- Diagrama de interacción
  - Diagrama de secuencia
  - Diagrama de comunicación
  - Diagrama de tiempos
  - Diagrama global de interacciones

[https://es.wikipedia.org/wiki/Lenguaje\\_unificado\\_de\\_modelado](https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado)



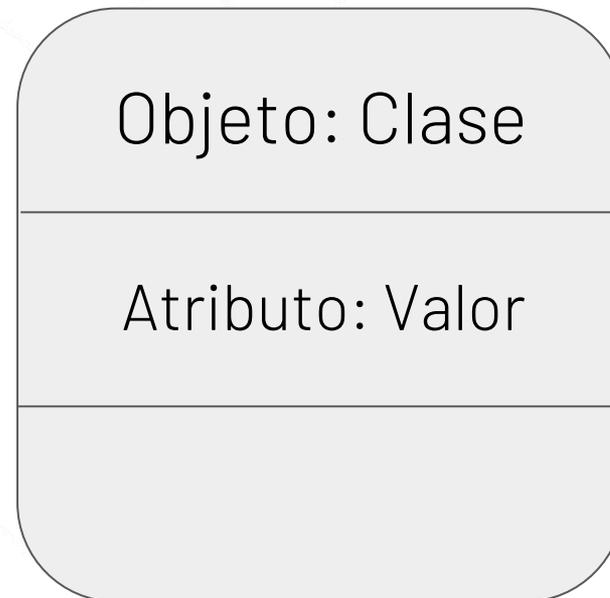
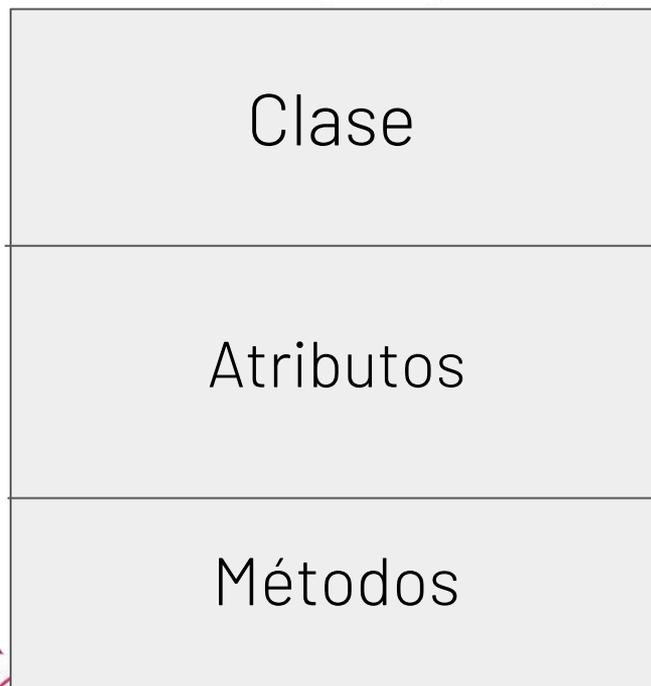
# Diagrama de clases

- Es el bloque de construcción principal de cualquier solución orientada a objetos.
- Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase.
- En sistemas grandes con muchas clases relacionadas, las clases se agrupan para crear diagramas de clases.





# Representación en UML

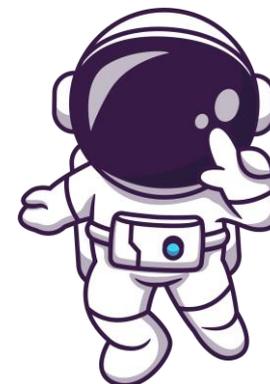




# Visibilidad

Para especificar la visibilidad de un miembro de la clase (es decir, cualquier atributo o método), se coloca uno de los siguientes signos delante de ese miembro:

+	Público
-	Privado
#	Protegido
/	Derivado (se puede combinar con otro)
~	Paquete





# Representación de Clases

Automóvil A	
Atributos	Placa
	Color
	Fabricante
	Velocidad
Métodos	acelerar
	girar
	frenar

Automóvil
- placa: String - color: String - fabricante: String - velocidad: Double
+ acelerar() + girar() + frenar()



# Representación de Objetos

Clase Automóvil		A
Atributos	Placa	ABC111
	Color	Azul
	Fabricante	Mitsubishi
	Velocidad	50 Km/h
Métodos	acelerar	
	girar	
	frenar	

A: Automóvil

placa: ABC111

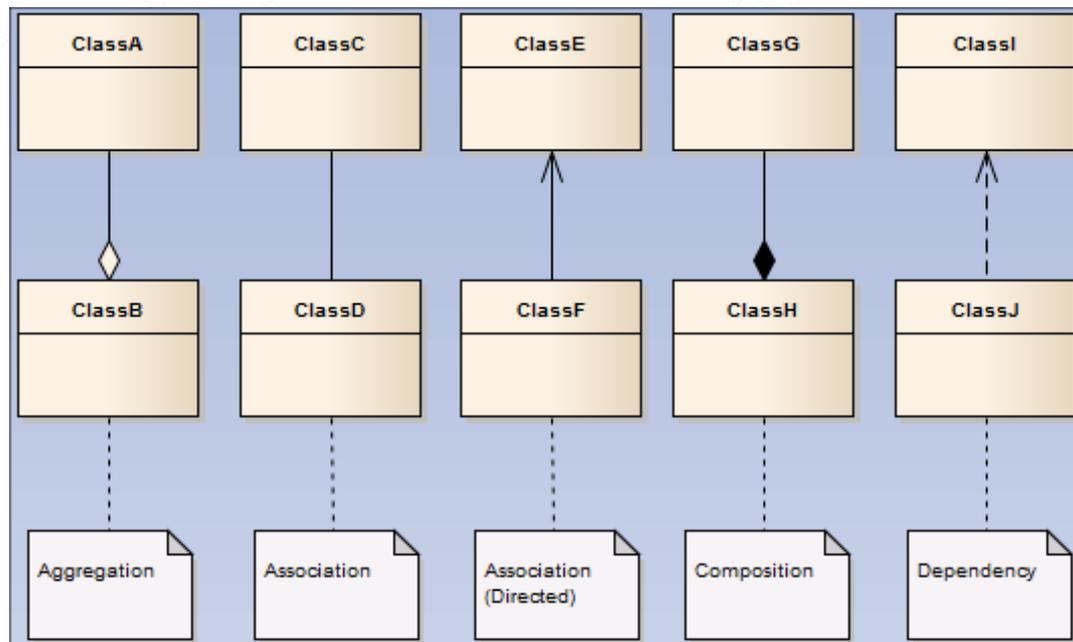
color: Azul

fabricante: Mitsubishi

velocidad: 50.0

# Relaciones entre clases

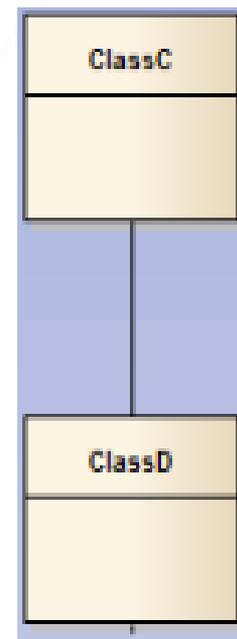
- Asociación
- Agregación
- Composición
- Dependencia
- Generalización





# Asociación

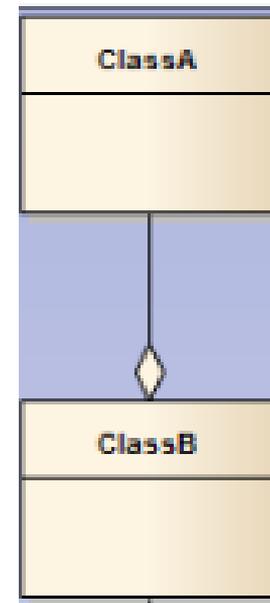
- Es una conexión física o conceptual entre clases.
- Son bidireccionales.
- El nombre puede ser omitida si la tiene un claro entendimiento de su relación.
- Es representado con una línea.
- La asociación entre objetos se llama **enlace**





# Agregación (o Agrupación)

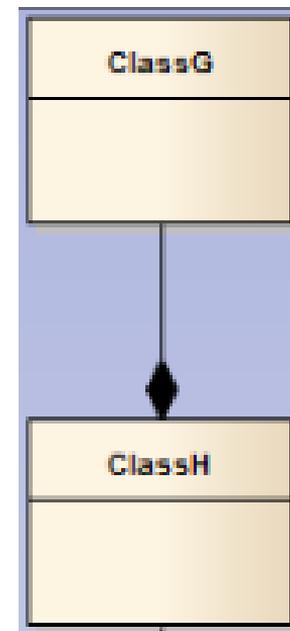
- Es una variante de la relación de asociación **tiene-un**.
- Se trata de una asociación que representa una relación de tipo **parte-todo** o **parte-de**.
- Es representado con un rombo blanco al lado del contenedor y una línea.
- Si el contenedor es eliminado, el contenido usualmente no es destruido.





# Composición

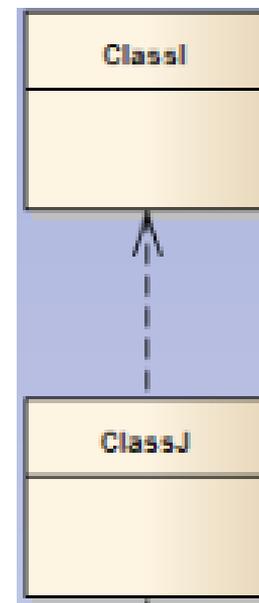
- Representa una clase que es compuesta por una o más clases. Intentamos representar un todo y sus partes.
- Se trata de una asociación que representa una relación de tipo **está-compuesto-por**, **parte-todo** o **parte-de**.
- Los componentes existen mientras la clase que contenedora exista.
- Es representado con un rombo negro al lado del contenedor y una línea.





# Dependencia

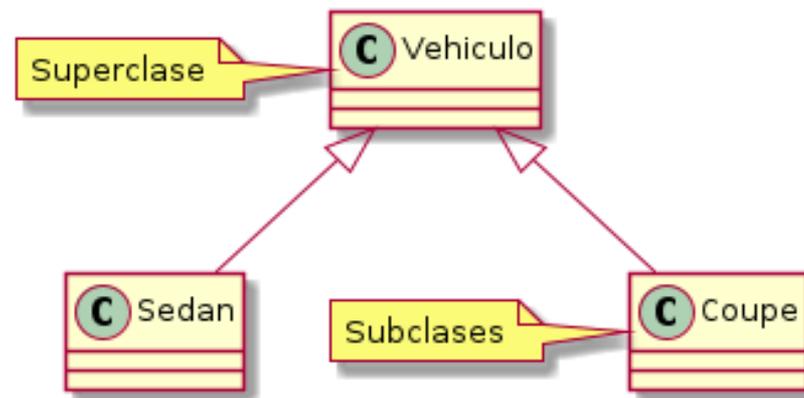
- Es una relación de uso, es decir que una clase utiliza a otra. Y si esta última se altera, la anterior se puede ver afectada.
- Las asociaciones de dependencia ocurren cuando una clase necesita otra clase para existir o cuando el objeto es usado para retornar una variable local, o el argumento de un método.
- Se representa con una línea discontinua con una flecha simple.



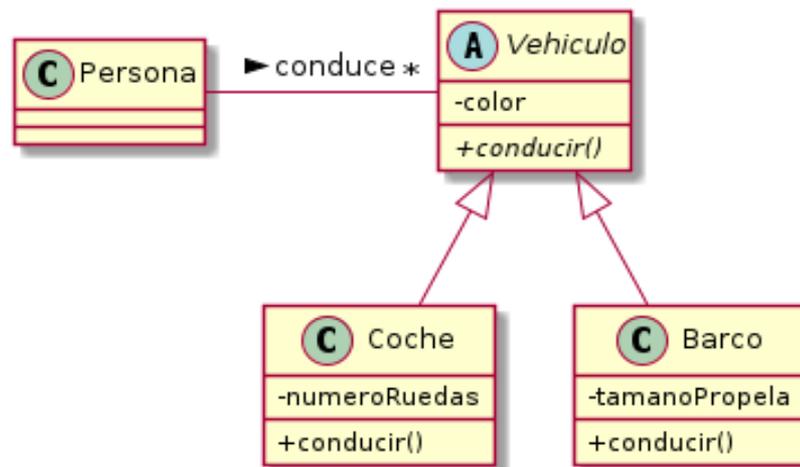
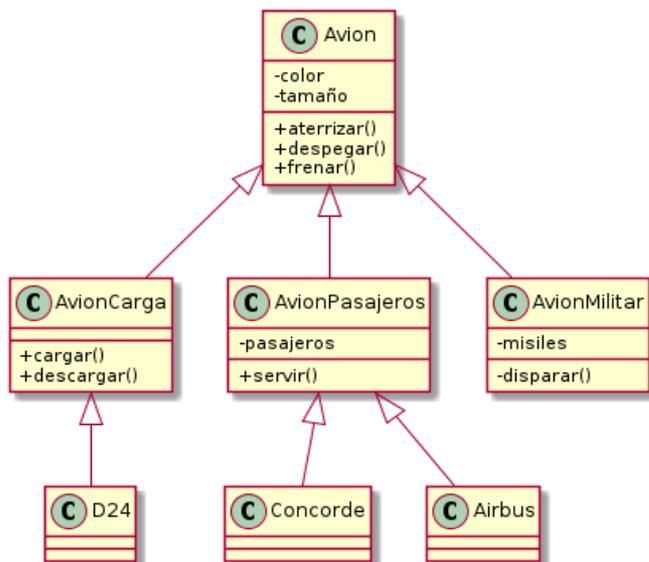


# Generalización / Especificación

- Representa una relación entre clases de tipo **es-un** o **un-tipo-de**.
- La clase que se está especializando se llama **superclase** y la especializada es la **subclase**.
- Los atributos y métodos de la superclase son compartidos por todas las subclases.
- Toda instancia de una subclase, es instancia de la superclase.



# Generalización / Especificación





# Multiplicidad en relaciones

- La multiplicidad de una asociación determina cuántos objetos de cada tipo intervienen en la relación

Multiplicidad	Significado
1	Uno y solo uno
0..1	Cero o uno
m..n	Entre m y n
*	Cero o varios
0..*	Cero o varios
1..*	Uno o varios (al menos uno)





# La importancia de las palabras

Una biblioteca contiene libros y revistas. Puede haber varias copias de un libro. Algunos de los libros son reservados sólo para préstamos a corto plazo. Todos los otros pueden ser prestados a cualquier miembro de la biblioteca por tres semanas. Los miembros de la biblioteca pueden normalmente solicitar hasta seis ítems de una vez, pero miembros del staff puede solicitar hasta doce ítems a la vez. Solamente miembros del staff pueden obtener prestado revistas.

El sistema debe conservar la pista de cuando los libros y revistas son prestados y retornados forzando las reglas de la biblioteca.





# Clases candidatas

- Biblioteca
- Libro
- Revista
- Copia
- PréstamosACortoPlazo
- MiembroDeBiblioteca
- Semana
- Item
- Tiempo
- MiembroDeStaff
- Sistema
- Regla

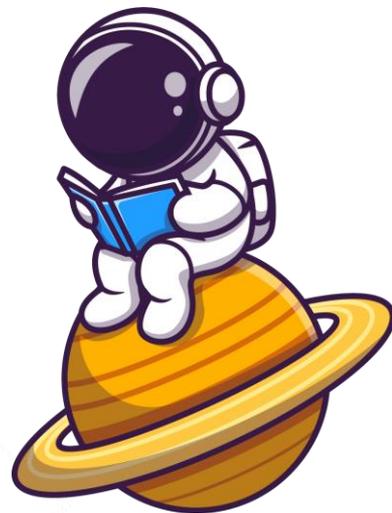
- Biblioteca
- Libro
- Revista
- Copia
- PréstamosACortoPlazo
- MiembroDeBiblioteca
- Semana
- Item
- Tiempo
- MiembroDeStaff
- Sistema
- Regla





# Operaciones

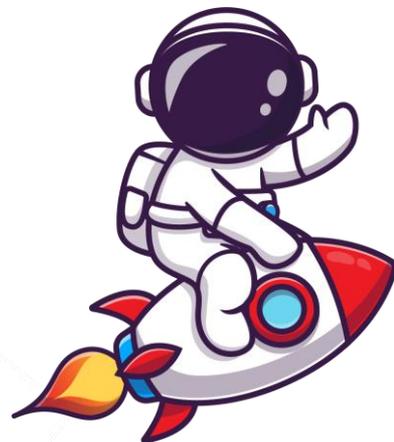
- **MiembroDeBiblioteca pide prestado Copia**
- **MiembroDeBiblioteca devuelve Copia**
- **MiembroDeStaff pide prestado Copia**
- **MiembroDeStaff devuelve Copia**





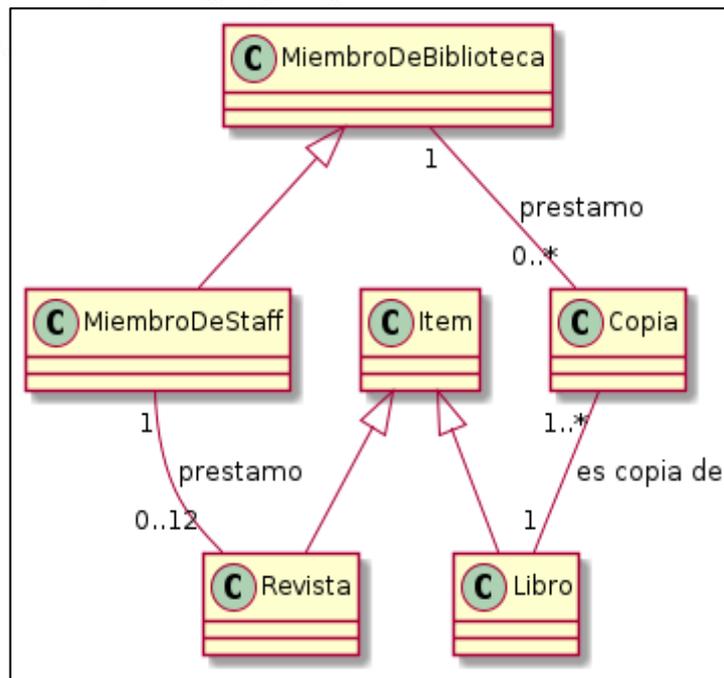
# Relaciones entre clases

- Libro es un Ítem
- Revista es un Ítem
- Copia es copia de Libro
- MiembroDeStaff pide prestado Revista
- MiembroDeStaff es un MiembroDeBiblioteca
- MiembroDeBiblioteca pide prestado Copia





# Modelo de Clases





# Ejercicio

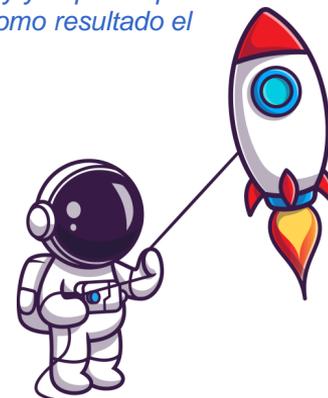
## Realizar el diagrama de clases para el juego del Ajedrez

*El ajedrez es un juego entre dos contrincantes en el que cada uno dispone al inicio de 16 piezas móviles que se colocan sobre un tablero, dividido en 64 casillas o escaques.*

*Se juega sobre un tablero cuadrulado de 8x8 casillas (llamadas escaques), 2 alternadas en colores blanco y negro, que constituyen las 64 posibles posiciones de las piezas para el desarrollo del juego. Al principio del juego cada jugador tiene dieciséis piezas: un rey, una dama, dos alfiles, dos caballos, dos torres y ocho peones. Se trata de un juego de estrategia en el que el objetivo es «derrocar» al rey del oponente. Esto se hace amenazando la casilla que ocupa el rey con alguna de las piezas propias sin que el otro jugador pueda proteger a su rey interponiendo una pieza entre su rey y la pieza que lo amenaza, mover su rey a un escaque libre o capturar a la pieza que lo está amenazando, lo que trae como resultado el jaque mate y el fin de la partida.*

...

**Leer resto de información:** <https://es.wikipedia.org/wiki/Ajedrez>





# Para la próxima sesión...

- Realizar los ejercicios de la unidad
- Realizar el diagrama de clases del juego del Ajedrez
- Revisar el material
  - Diagramas Estáticos en UML

