



## Ciclo 2

# Semana 3

## *Herencia, Abstracción y Polimorfismo*

---

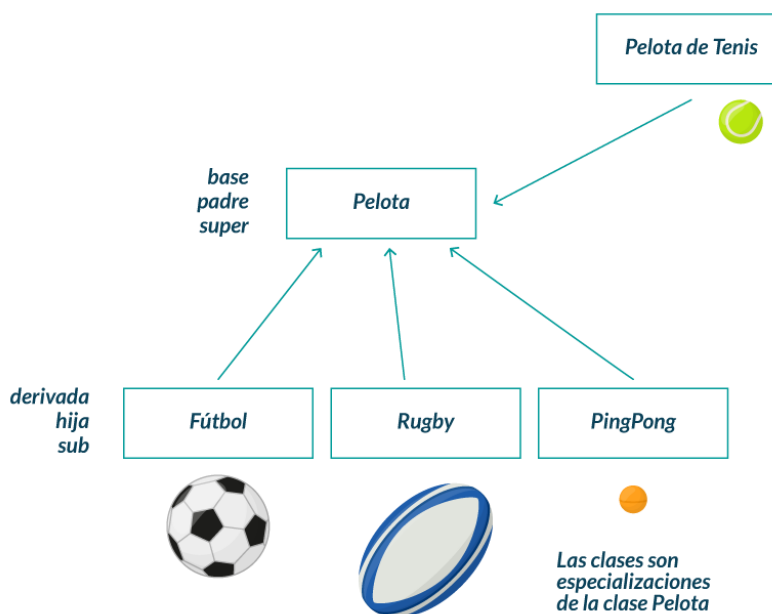
### Lectura 1 - Herencia - Clases Derivadas



## | Herencia - Clases derivadas

### Definición

En algunos casos existe una relación entre dos clases, una clase derivada que se crea de una que ya existe llamada clase base y que cuentan con un código y datos comunes. A esta relación se le llama herencia y puede darse tanto en clases como en objetos.

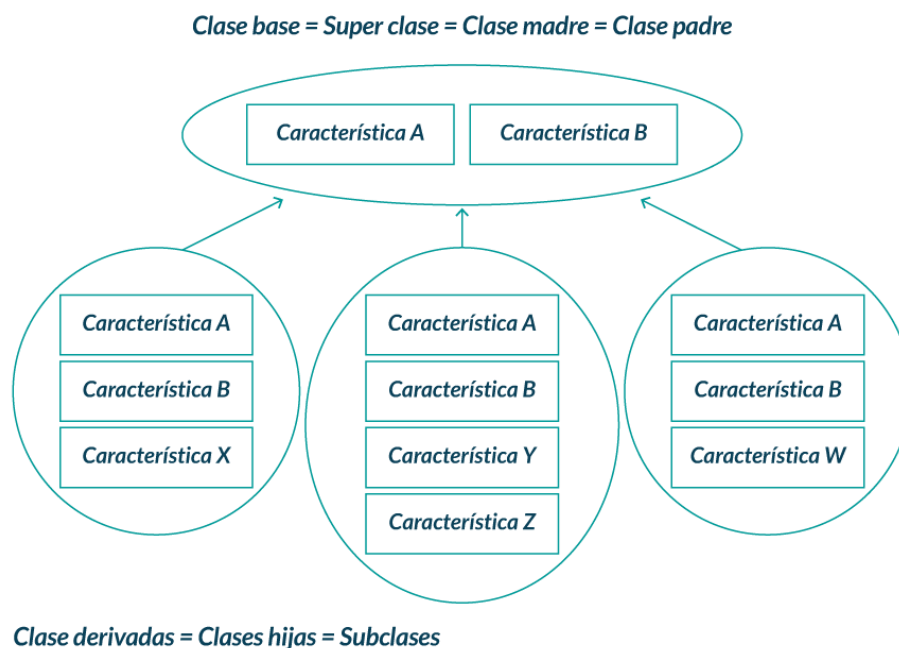


La clase original y de la cual se derivan las nuevas clases se llama padre o superclase y las clases derivadas se les denomina hijas o subclases.

En la programación orientada a objetos la herencia es una manera de organizar objetos en modo jerárquico pasando de lo general a lo específico. En términos de clase la herencia se usa para extender una clase en otra con una ejecución análoga, permitiendo que una clase se comporte igual que otra ampliando y adaptando así su comportamiento.



Con la herencia se representan objetos con diferencias y similitudes en su funcionamiento. Aquí la clase base puede contener la funcionalidad común de los objetos, para luego definir subclases con una funcionalidad propia y personalizada.



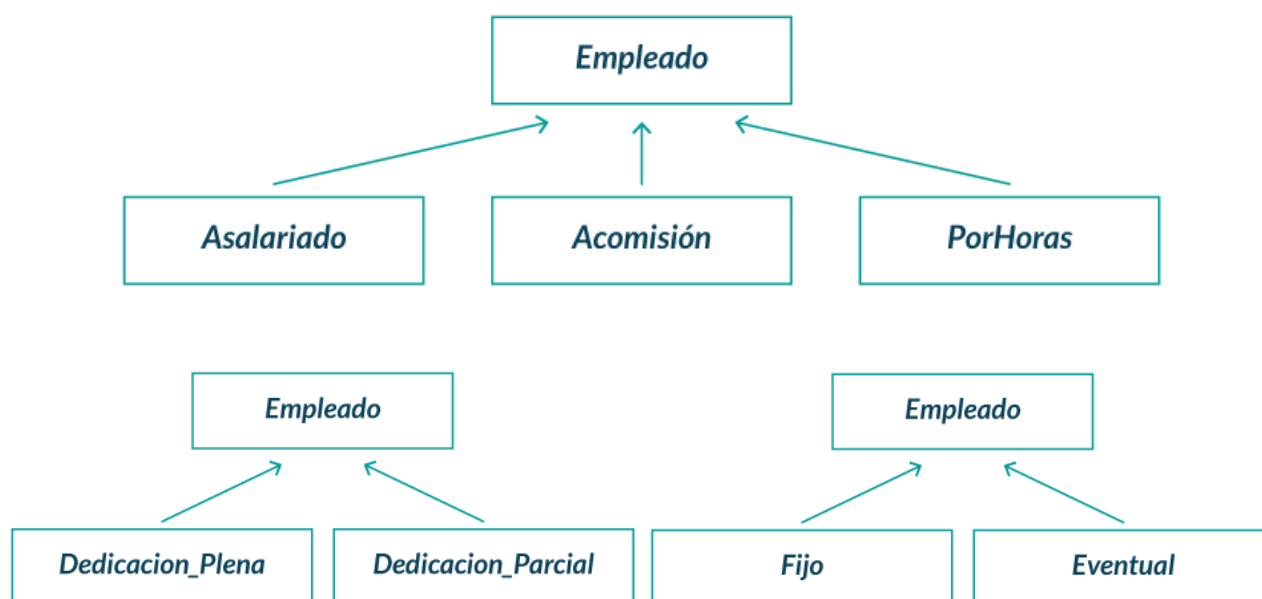
En los lenguajes orientados a objetos que se basan en clases, normalmente el objeto que se crea con la herencia se denomina objeto hijo y tendrá los comportamientos y propiedades del objeto padre.

Con la herencia las clases no están aisladas, sino que están relacionándose entre sí y conformando jerarquías de clasificación. Así la herencia facilita el polimorfismo y el encapsulamiento haciendo que los objetos sean definidos y creados como tipos especializados de objetos ya preexistentes. Si se estructuraran las clases con un diseño tipo árbol genealógico, se tendría lo que se conoce como una estructura jerarquizada de clases.

Al diseñar una aplicación en Lenguaje Orientado A Objetos, normalmente no es tan sencillo lograr establecer una relación de herencia óptima entre sus clases, pues depende del criterio de selección o discriminador que se tenga el enfoque puede cambiar. En una empresa no es lo mismo clasificar a sus empleados basados en el modo de pago, que clasificarlos por dedicación laboral o duración del contrato, los empleados se dividen en hijos. Adicional un empleado podría



pertenecer a más de una de estas clasificaciones en donde por ejemplo un empleado con dedicación plena tiene salario mensual.



## Tipos de herencia

- Herencia única



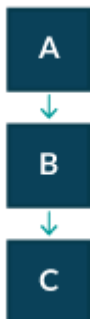
Se denomina herencia única a la relación en la que una clase hija hereda las propiedades y el comportamiento de una sola clase padre, o sea una clase se extiende solo a otra clase.



## Semana 3

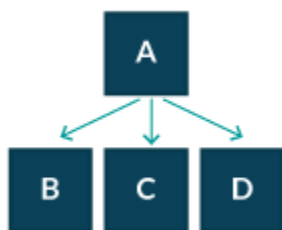
Herencia, abstracción  
y polimorfismo

- Herencia multinivel



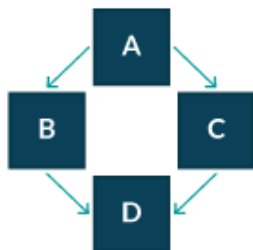
En esta herencia se crea una nueva clase a partir de una clase ya derivada, la cual se vuelve ahora una clase base para una nueva clase.

- Herencia jerárquica



La herencia jerárquica es aquella donde muchas subclases tienen herencia de una sola clase, por tanto, una sola clase base hereda a más de una clase derivada.

- Herencia híbrida



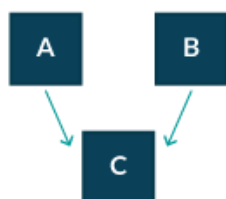
Es la combinación de las tres herencias anteriores, la herencia única, la jerárquica y a la multinivel.



## Semana 3

Herencia, abstracción  
y polimorfismo

- Herencia múltiple



Corresponde a la herencia en donde una clase hereda de más de una superclase, aunque en esta herencia la clase derivada tendrá que diligenciar la dependencia que tiene con sus dos clases base.

Existe también tres tipos de herencia para la clase base

- Herencia pública

Cuando la clase derivada viene de una clase base pública, los miembros públicos de esta clase base se volverán miembros públicos de la clase derivada y los protegidos de la clase base se convierten también en miembros protegidos de la clase derivada, así los miembros privados de la superclase no podrán ser asequibles desde una clase derivada directamente, pero si se puede acceder mediante llamadas que hagan a los miembros públicos y protegidos de la clase base.

- Herencia protegida

Al crear una clase que es heredada desde una superclase de tipo protegida, miembros públicos y protegidos de la superclase pasaran a ser miembros protegidos de la clase derivada.

- Herencia privada

Al definirse una subclase desde una superclase privada, los miembros públicos y protegidos de la superclase cambiarán a miembros privados de la subclase.





Especificador Acceso	Accesible desde su propia clase	Accesible desde la clase derivada	Accesible desde objetos fuera_clase
public	si	si	si
private	si	si	no
protected	si	no	no

Al derivar clases de otras existentes, no es necesario escribir desde cero, por lo que la herencia ofrece una gran característica de reutilización de código. Una clase derivada hereda los métodos de la base, excepto constructores, destructores, operadores sobrecargados de la clase base y funciones amigas de la clase base. La herencia se da con los atributos y métodos `public` y `protected` y no con los `private`. Si un método de la clase hija redefine un atributo o un método de su clase padre, se puede hacer referencia al miembro redefinido anteponiendo la palabra reservada `super`. Así los constructores no se heredan, sino que son invocados con `super`.

## Constructores y herencia

Los constructores se usan para inicializar atributos de clases, dándoles su estado inicial. En las clases derivadas o subclases de la superclase, los constructores, no solo dan el estado inicial de su propia subclase, sino que además dan el estado inicial de la superclase, por tanto, los constructores de las subclases pueden invocar a los constructores de su superclase.

Como se indicaba anteriormente, a diferencia de otros miembros de una superclase, los constructores no se heredan y son llamados usando `super()`. En caso de usarse, obligatoriamente debe ser la primera instrucción en el cuerpo del constructor de la subclase, pero debe tener en cuenta que si un constructor de una subclase no utiliza la instrucción `super`, el compilador podría insertar de forma automáticamente la instrucción `super()`, generando

posibles errores si en la superclase se definió un constructor y no el constructor `super()`, pues el compilador no encontrará este constructor

Cuando una clase derivada crea un objeto, estos tendrán la porción que corresponde a su clase base y sus miembros propios, por tanto, primero se crea la parte de su clase base, haciendo un llamado a su constructor, y luego se inicializan los miembros propios de la clase derivada.

Se debe de tener en cuenta:

- El constructor de la clase base se invoca antes del constructor de la clase derivada.
- Si una clase base es también una clase derivada, se invocan en orden primero el constructor base, y luego el constructor derivada.
- Los métodos que implementan a los constructores no se heredan.
- Si no se indica el constructor de la clase base, se hará automáticamente sin argumentos.