

## RESUMEN DE CONCEPTOS

---

- Clase. Conjunto de elementos que comparten características y funcionalidades.
  - A cada uno de los elementos se les conoce con el nombre de **objeto**.
  - A las características se les conoce con el nombre de **atributos**
  - A las funcionalidades se les conoce con el nombre de **métodos**
- Un objeto es una instancia de una clase, una representación específica y se crea a partir de la clase.
- Un objeto tiene la facultad de llamar a los métodos que se definen en su clase, dado que definen su comportamiento.
- Los valores de los atributos de un objeto definen su estado, por ende, el estado de un objeto puede cambiar en el tiempo.
- Los métodos se clasifican en: GETTER, SETTER y métodos especializados.
- Cada clase cuenta con uno o varios constructores. Un constructor se caracteriza por llevar el mismo nombre de la clase y no cuenta con tipo de dato de retorno. La labor de un constructor es la de crear un objeto.
- Tipos de datos en Java: pueden ser primitivos o referencias.

PRIMITIVOS	REFERENCIAS (CLASES)
1. byte 2. short 3. int 4. long 5. float 6. double 7. boolean 8. char	1. String 2. Scanner 3. Integer (esta clase es para representar los int como objetos) 4. Float (esta clase es para representar los int como objetos) 5. Short (esta clase es para representar a los short como objetos) 6. Byte (esta clase es para representar a los byte como objetos) 7. Double (esta clase es para representar a los double como objetos) 8. Boolean (esta clase es para representar a los boolean como objetos) 9. Long (esta clase es para representar a los long como objetos) 10. Character (esta clase es para representar a los char como objetos) 11. Y muchas más que podrá encontrar en: <a href="https://docs.oracle.com/javase/8/docs/api/?xd_co_f=c5796437-7b20-400d-a182-85e041425308">https://docs.oracle.com/javase/8/docs/api/?xd_co_f=c5796437-7b20-400d-a182-85e041425308</a>

## Representación de una clase

---

Un rectángulo dividido en 3 secciones: nombre de la clase, lista de atributos, lista de métodos.

## Relaciones entre clases

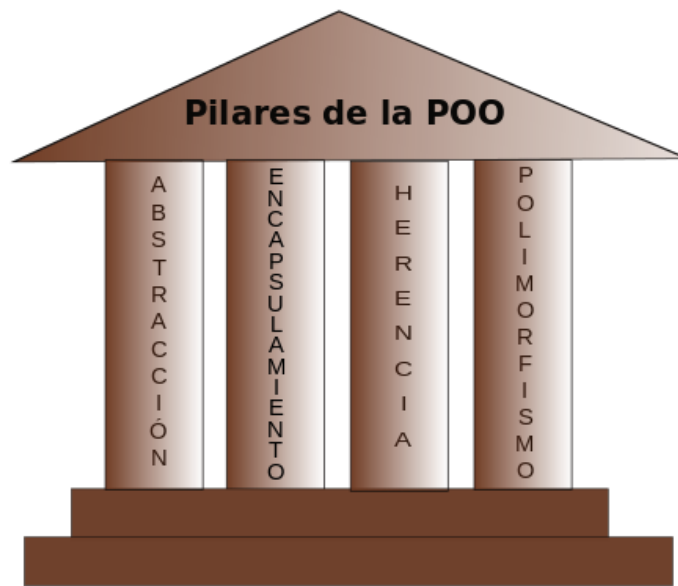
---

Las clases existen para poder crear objetos a partir de ellas (sirven de molde) y pueden relacionarse con otras clases. Existen 2 tipos de relaciones:

- Asociación (agregación o composición)
- Herencia

## Pilares de la orientación a objetos

---



**Abstracción.** Es la capacidad de poder considerar solo los detalles relevantes, dejando de lado lo que no tiene que ver con el contexto del problema.

**Encapsulamiento.** Proteger a los atributos/componentes de una clase, tiene relación con la visibilidad de los elementos. Ver Figura 1 - Explicación Encapsulación (fuente: extraída desde <https://informaciongeneralsite.wordpress.com/>)

**Herencia.** Tipo de relación entre clases que indica jerarquía. Aparecen los conceptos de super clase (clase padre), subclases (clases hijas)

**Polimorfismo.** Es la capacidad que tiene un objeto de cambiar su comportamiento dependiendo de su tipo (este concepto lo vamos a profundizar más adelante)



# Encapsulamiento

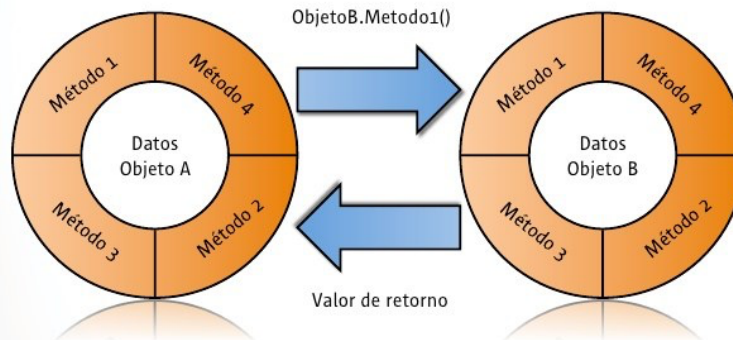


Figura 1 - Explicación Encapsulación (fuente: extraída desde <https://informaciongeneralsite.wordpress.com/>)

## Lecturas de la bibliografía

---

### ***Cómo programar en Java - Deitel & Deitel***

Sección 1.5 Introducción a la tecnología de los objetos

### ***Programación en Java – Luis Joyanes***

Sección 3.3 Tipos de datos en Java