# Teoría:

Cuando hablamos de herencia en programación no nos referimos precisamente a que algún familiar lejano nos ha podido dejar una fortuna, ya nos gustaría. En realidad, se trata de uno de los pilares fundamentales de la programación orientada a objetos. Es el mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase.

La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

La herencia en programación orientada a objetos es un mecanismo que permite a una clase (llamada clase derivada o subclase) heredar propiedades y métodos de otra clase (llamada clase base o superclase). Este concepto es fundamental en la programación orientada a objetos porque promueve la reutilización del código y la creación de jerarquías de clases más claras y organizadas.

# Principales Características de la Herencia:

Reutilización de Código: Permite que las subclases utilicen el código definido en la superclase sin necesidad de volver a escribirlo. Esto ahorra tiempo y reduce errores.

Jerarquía de Clases: Facilita la organización y estructuración del código en una jerarquía, donde las subclases pueden especializarse a partir de una superclase más genérica.

Extensibilidad: Las subclases pueden extender (añadir nuevas funcionalidades) o modificar (sobrescribir métodos) el comportamiento de la superclase, adaptándolo a necesidades específicas.

Polimorfismo: La herencia permite el polimorfismo, donde un objeto de una subclase puede ser tratado como un objeto de la superclase, lo que facilita la creación de código más flexible y reutilizable.