

Qué es la encapsulación: Es la agrupación de datos con los métodos que operan en esos datos, o la restricción del acceso directo a algunos de los componentes de un objeto.

[Fuente de Referencia](#)

Significado: La encapsulación en un lenguaje de programación es un principio fundamental de la programación orientada a objetos que consiste en agrupar los datos (atributos) y los métodos (comportamientos) relacionados con un objeto dentro de una sola unidad llamada clase

Para lograr la encapsulación en un lenguaje de programación, se utilizan modificadores de acceso, como "public", "private" y "protected", para determinar qué partes del código pueden acceder y modificar los atributos y métodos de una clase.

Public: Los miembros (atributos y métodos) marcados como públicos son accesibles desde cualquier parte del programa. No existen restricciones en su acceso.

Private: Los miembros marcados como privados solo son accesibles dentro de la misma clase. Otros objetos o clases no pueden acceder directamente a ellos.

Protected: Los miembros marcados como protegidos son accesibles dentro del mismo paquete y también por clases derivadas (herencia) fuera del paquete.

[Fuente de Referencia](#)

Origen: Se originó en el paradigma de programación orientada a objetos (OOP). La OOP es un enfoque de diseño de software que organiza la lógica del programa alrededor de "objetos", que son instancias de clases que encapsulan datos y los métodos que operan sobre esos datos. David Parnas introdujo por primera vez el concepto de ocultación de información alrededor de 1972.

Ejemplos:

```
public class Empleado{

    private String nombre;
    private double salario;

    public Empleado(){
        nombre="";
        salario=0.0;
    }

    public Empleado(String nombre, double salario){
        this.nombre=nombre;
        this.salario=salario;
    }

    public String getNombre(){
```

```
        return nombre;
    }

    public void setNombre(String nombre){
        this.nombre=nombre
    }

    public double getSalario(){
        return salario;
    }

    public void setSalario(double salario){
        this.salario=salario;
    }
}
```