<!--SOLID-->

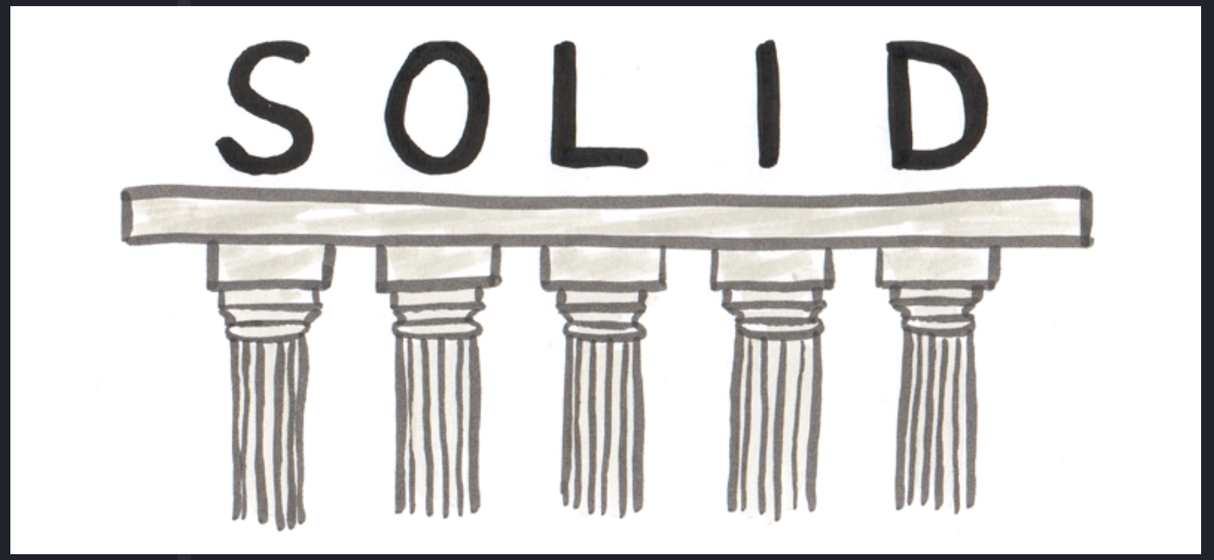
Introducción a SOLID

<Por="Raquel Martinez"/>

}

SOLID {

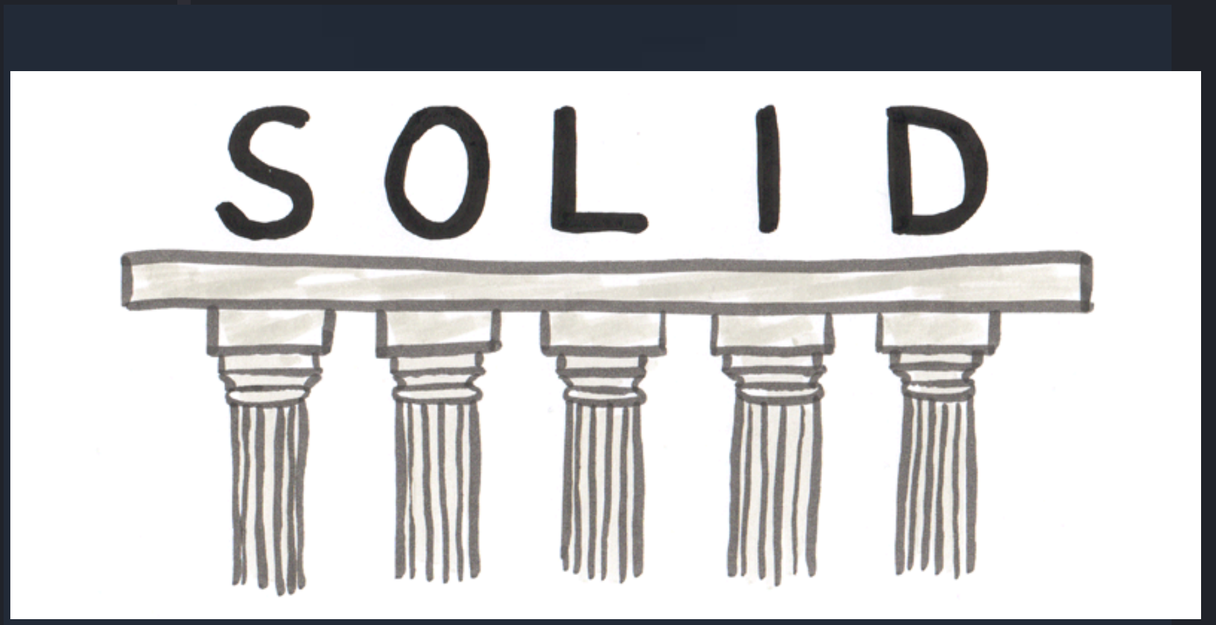
Son un conjunto de reglas y mejores prácticas a seguir al diseñar una estructura de clase.



}

SOLID {

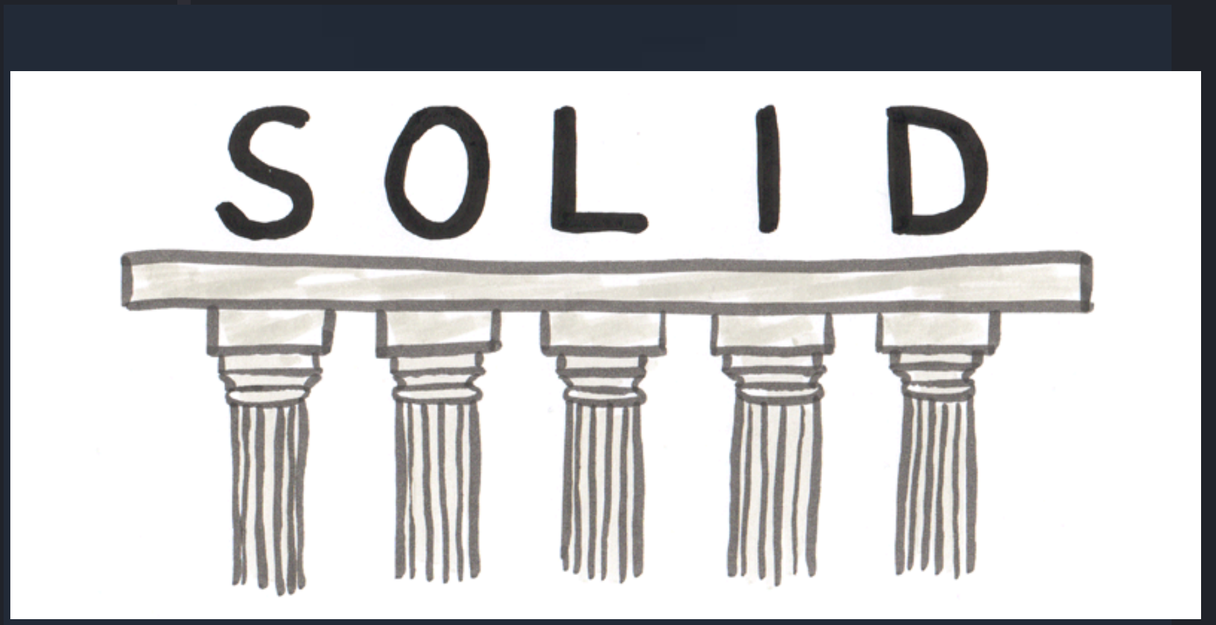
- Los principios SOLID fueron introducidos por primera vez por el famoso científico informático Robert J. Martin



}

SOLID {

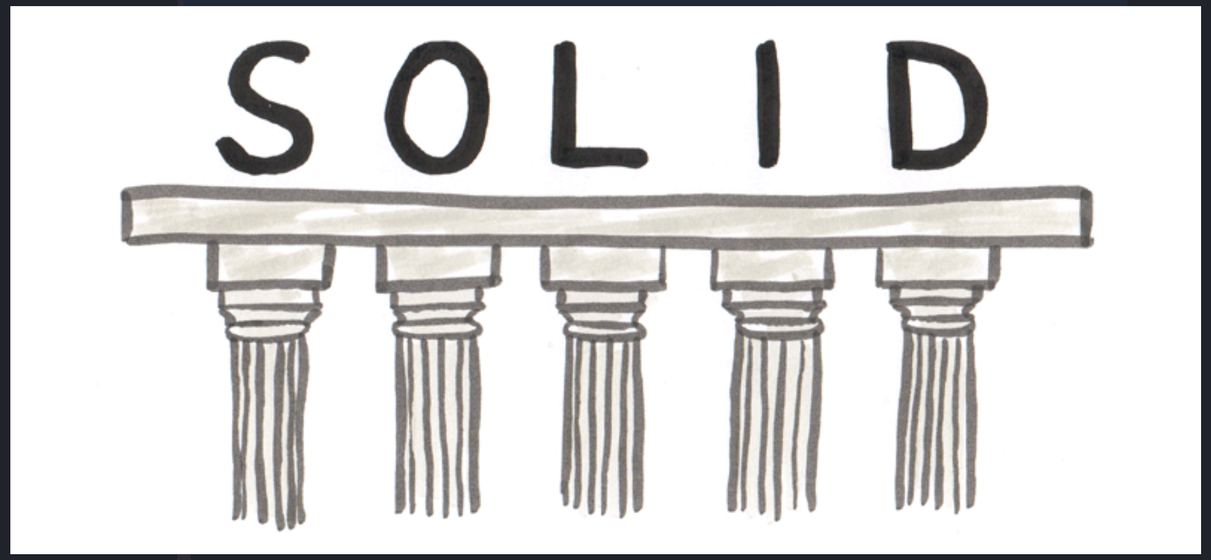
"Para crear código comprensible, legible y comprobable en el que muchos desarrolladores puedan trabajar en colaboración."



}

SOLID {

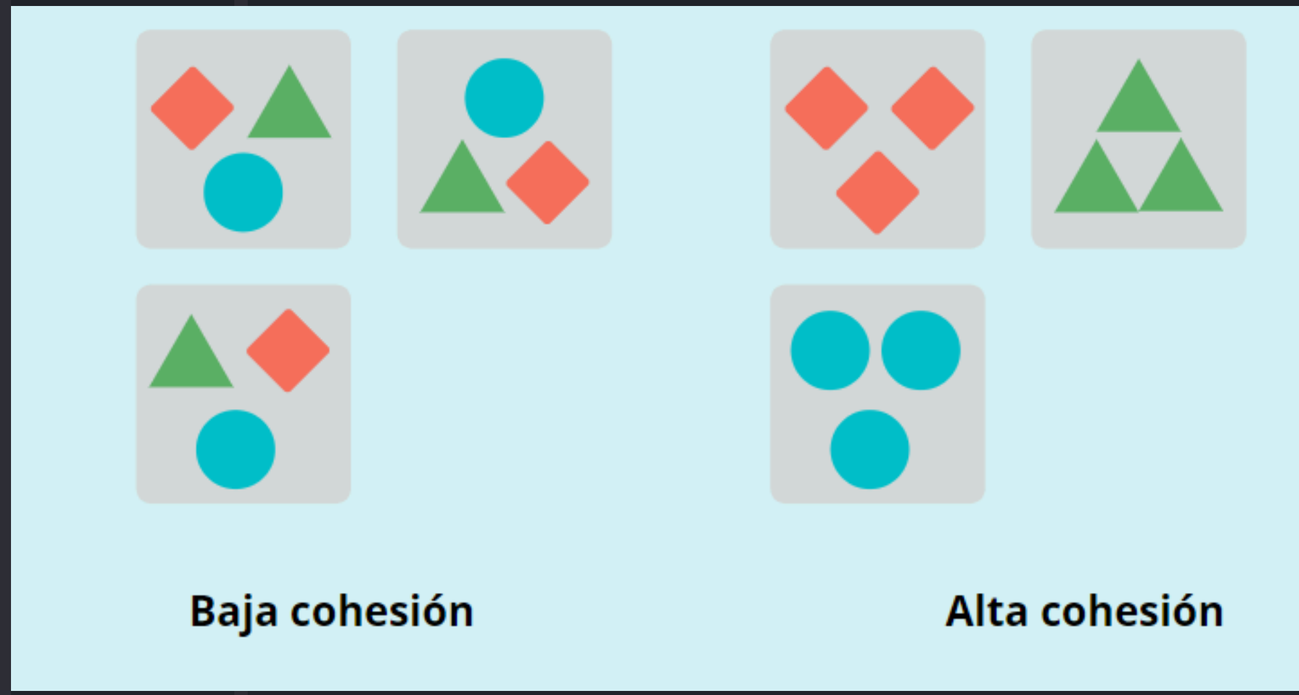
1. Cohesión
2. Acoplamiento



}

Cohesión {

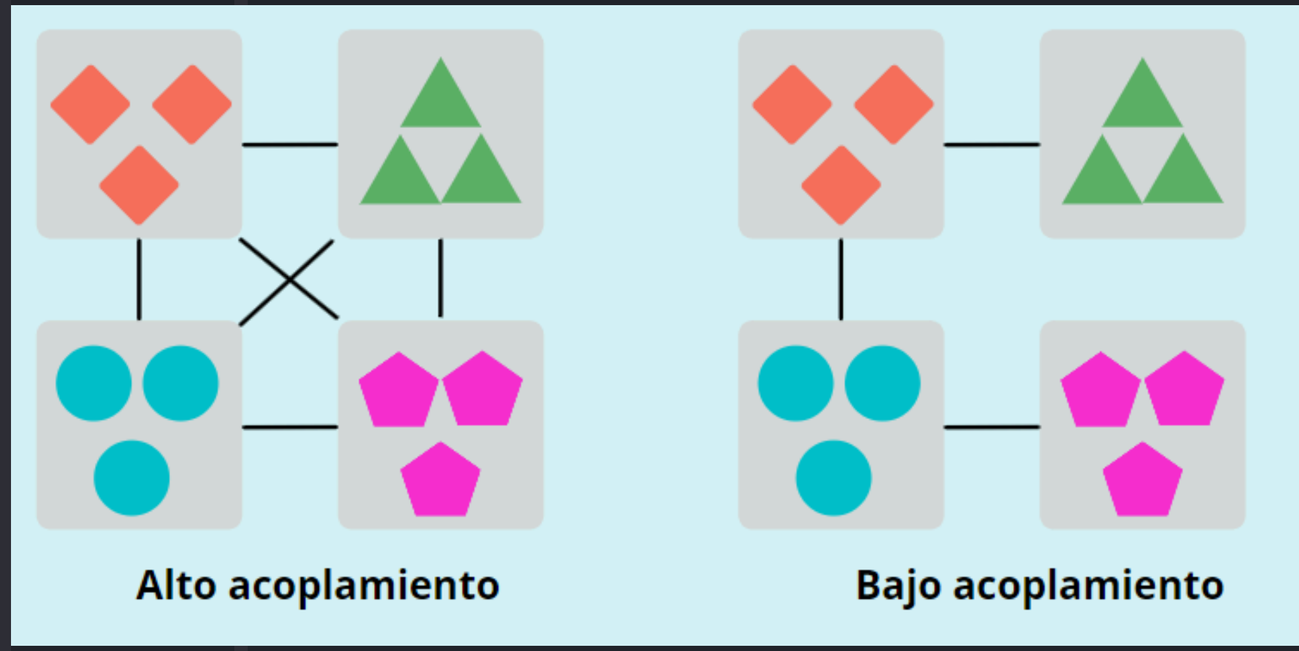
- La cohesión se refiere a la medida en que las responsabilidades y funciones de un módulo (clase, método o componente) están relacionadas y enfocadas en una única tarea o propósito.



}

Acoplamiento {

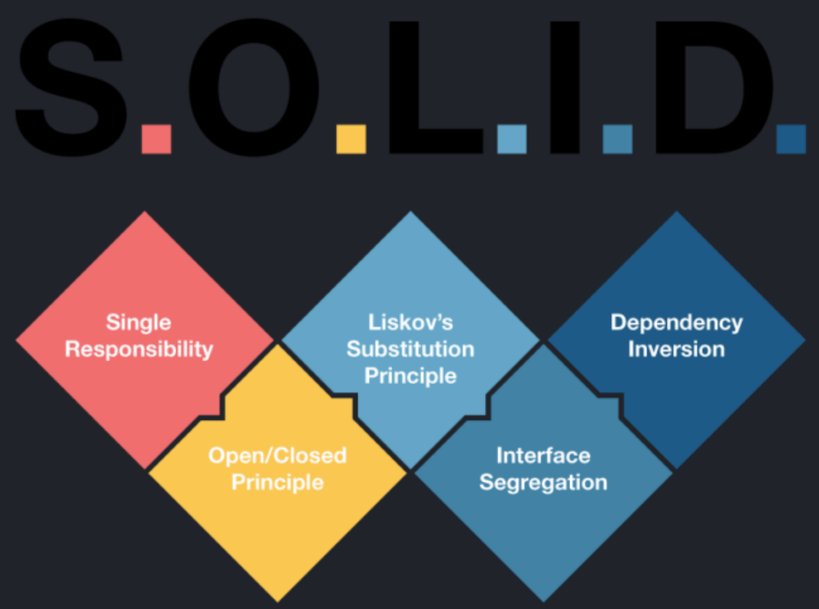
- Grado de interdependencia entre los diferentes módulos de un sistema.



}

Single Responsibility Principle - Principio de responsabilidad única {

- Una clase debe hacer una cosa y, por lo tanto, debe tener una sola razón para cambiar.



}

Single Responsibility Principle - Principio de responsabilidad única {

A design that fails to conform to the SRP would look like this:

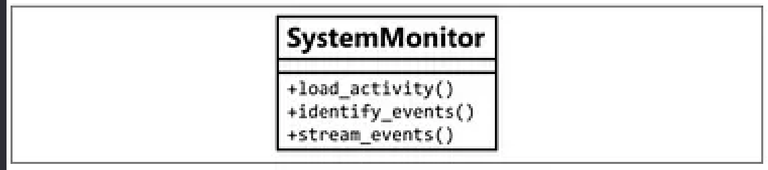


Figure 4.1: A class with too many responsibilities

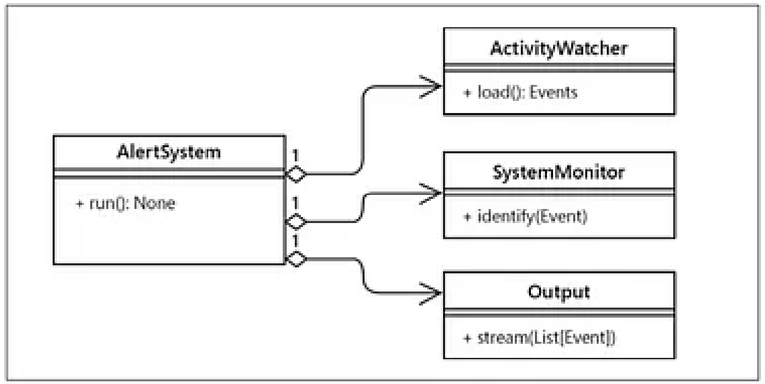
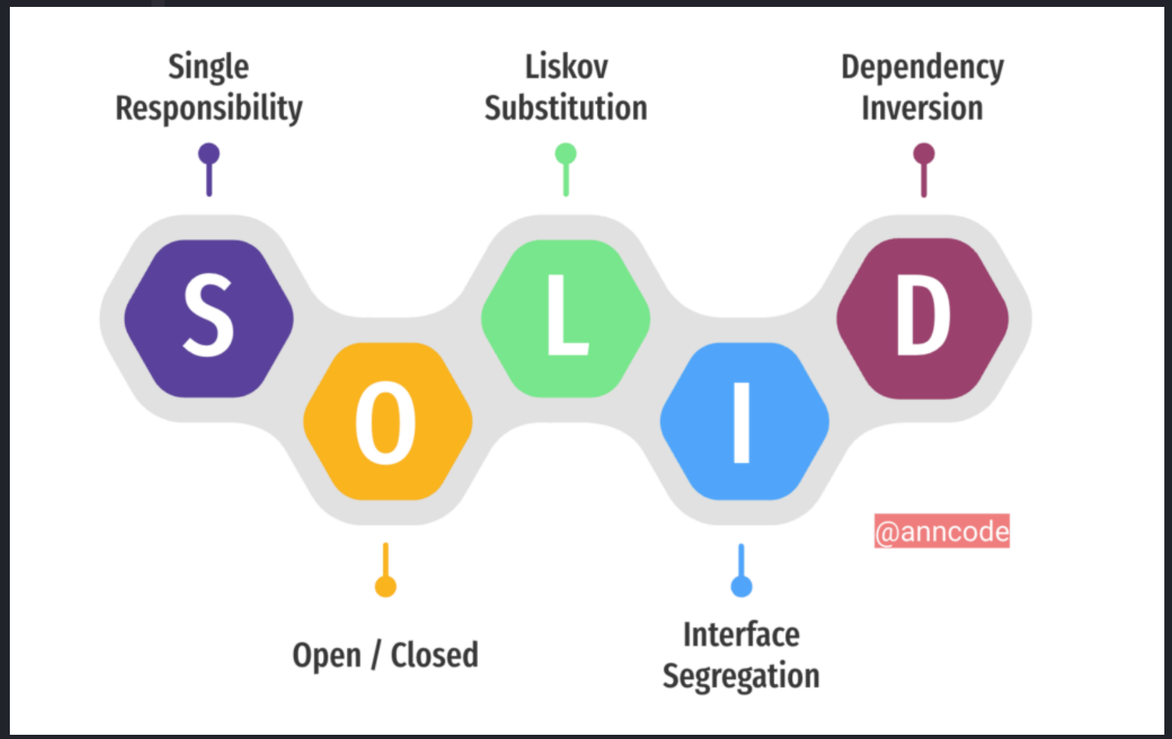


Figure 4.2: Distributing responsibilities throughout classes

}

S DE SOLID {

- Claridad de la responsabilidad
- Facilita el mantenimiento
- Promueve la cohesión
- Facilita la reutilización del código



}

SOLID {

```java
public class Factura {

  private Libro libro;
  private int cantidad;
  private double tasaDescuento;
  private double tasaImpuesto;
  private double total;

  public Factura(Libro libro, int cantidad, double tasaDescuento, double tasaImpuesto) {
    this.libro = libro;
    this.cantidad = cantidad;
    this.tasaDescuento = tasaDescuento;
    this.tasaImpuesto = tasaImpuesto;
    this.total = this.calculaTotal();
  }

  public double calculaTotal() {
          double precio = ((libro.precio - libro.precio * tasaDescuento) * this.cantidad);

    double precioConImpuestos = precio * (1 + tasaImpuesto);

    return precioConImpuestos;
  }

  public void imprimeFactura() {
            System.out.println(cantidad + "x " + libro.nombre + " " + libro.precio + "$");
            System.out.println("Tasa de Descuento: " + tasaDescuento);
            System.out.println("Tasa de Impuesto: " + tasaImpuesto);
            System.out.println("Total: " + total);
  }

        public void guardarArchivo(String nombreArchivo) {
  // Crea un archivo con el nombre dado y escribe la factura.
  }

}
```

}

SOLID {

```java
public class FacturaImpresion {
    private Factura factura;

    public FacturaImpresion(Factura factura) {
        this.factura = factura;
    }

    public void imprimir() {
        System.out.println(factura.cantidad + "x " + factura.libro.nombre + " " + factura.libr
        System.out.println("Tasa de Descuento: " + factura.tasaDescuento);
        System.out.println("Tasa de Impuesto: " + factura.tasaImpuesto);
        System.out.println("Total: " + factura.total + " $");
    }
}
```

}

SOLID {

```
public class FacturaPersistencia {
    Factura factura;

    public FacturaPersistencia(Factura factura) {
        this.factura = factura;
    }

    public void guardarArchivo(String nombreArchivo) {
        // Crea un archivo con el nombre dado y escribe la factura.
    }
}
```

}

<!--SOLID-->

Codifiquemos! {

<Ejemplo/>

}

<!--SOLID-->

Gracias {

<Por="Raquel Martinez/>

}