

# El Principio de Segregación de Interfaces (ISP)

FLOREZ VARGAS ANGIE VALENTINA  
BERMUDEZ GUTIERREZ SEBASTIAN  
QUINTERO RAMIREZ CESAR DAVID  
CRUZ PARDO JUAN CAMILO  
PEREZ BERMUDEZ LEONARDO FABIO  
SOLANO RUIZ IKER EDWIN SANTIAGO  
PUENTES GONZALEZ SEBASTIAN





# ¿Qué es el Principio de Segregación de Interfaces?

El Principio de Segregación de Interfaces (ISP), o Interface Segregation Principle por sus siglas en inglés, es el cuarto de los cinco principios SOLID para el diseño orientado a objetos. Fue introducido por Robert C. Martin (Uncle Bob) y aboga por la especialización de las interfaces.

**"Los clientes no deberían ser forzados a depender de interfaces que no utilizan."**

En esencia, sugiere que es mejor tener muchas interfaces pequeñas y específicas que una sola interfaz grande y general.



# Las "Interfaces Gordas": Una Carga Innecesaria

¿Qué sucede cuando una clase se ve obligada a implementar métodos que no le corresponden o que nunca utilizará? Esto es lo que llamamos una "interfaz gorda" o "contaminada".

Imaginemos una impresora multifuncional: puede imprimir, escanear, copiar y enviar faxes. Si creamos una única interfaz `IlmpresoraMultifuncional` con todos esos métodos, una clase que solo necesita imprimir (como una impresora básica) se vería forzada a implementar métodos como `Escanear()` o `EnviarFax()`, incluso si los deja vacíos o lanza excepciones.

## Acoplamiento Fuerte

Dependencia excesiva entre clases.

## Código Hinchado

Implementaciones vacías o inútiles.

## Mantenimiento Difícil

Cambios en un método afectan a muchas clases.

# La Filosofía de "Divide y Vencerás" en Interfaces

El ISP nos guía a descomponer interfaces monolíticas en múltiples interfaces de rol específico, permitiendo que las clases implementen solo lo que realmente necesitan.



En lugar de una única `IlmpresoraMultifuncional`, tendríamos `Ilmpresora`, `IEscanner`, `ICopiadora` y `IFax`. Una impresora básica solo implementaría `Ilmpresora`, mientras que la impresora multifuncional implementaría todas ellas.

# Ventajas de las Interfaces Pequeñas y Enfocadas

Evitar el "Efecto Dominó"

Mayor Facilidad de Testeo

Código Más Legible y Mantenible

Reducción del Acoplamiento

# Guía Paso a Paso para Implementar el ISP

## Identificar Métodos no Utilizados

Revisa tus interfaces actuales y las clases que las implementan. Busca métodos que algunas clases se ven forzadas a implementar sin hacer nada útil.

## Ajustar Clases y Dependencias

Modifica las clases para que implementen solo las interfaces que realmente necesitan. Actualiza las dependencias para que utilicen las nuevas interfaces.

## Refactorizar en Nuevas Interfaces

Crea nuevas interfaces más pequeñas, cada una agrupando un conjunto de métodos relacionados y coherentes con un único propósito o "rol".

## Considerar la Herencia de Interfaces

Si varias interfaces tienen métodos comunes, puedes usar la herencia de interfaces para crear una jerarquía clara y reutilizar definiciones de métodos.

# El Valor de Segregar Interfaces

## Sistemas Más Ágiles

Facilita la evolución y adición de nuevas funcionalidades sin romper las existentes.

## Diseño Cohesivo

Cada componente tiene una responsabilidad única y bien definida, mejorando la coherencia.

## Mayor Resistencia al Cambio

Los cambios localizados tienen un impacto mínimo en el resto de la base de código..