

¿Qué son los principios SOLID?



an NTT DATA Company

- Es un acrónimo creado por **Robert C.Martin**, para establecer los cinco principios básicos de la programación orientada a objetos y diseño.
- Cada una de las letras hace referencia a un principio. (Se verá en la siguiente diapositiva)
- Es imprescindible para crear clean code y que nuestro sistema sea fácil de mantener y ampliar con el tiempo.

¡Los 5 fantásticos!



an NTT DATA Company

Inicial	Acrónimo	Concepto
S	SRP	Principio de responsabilidad única (<i>Single responsibility principle</i>) la noción de que un objeto solo debería tener una única responsabilidad.
O	OCP	Principio de abierto/cerrado (<i>Open/closed principle</i>) la noción de que las “entidades de software ... deben estar abiertas para su extensión, pero cerradas para su modificación”.
L	LSP	Principio de sustitución de Liskov (<i>Liskov substitution principle</i>) la noción de que los “objetos de un programa deberían ser reemplazables por instancias de sus subtipos sin alterar el correcto funcionamiento del programa”. Ver también diseño por contrato .
I	ISP	Principio de segregación de la interfaz (<i>Interface segregation principle</i>) la noción de que “muchas interfaces cliente específicas son mejores que una interfaz de propósito general”. ⁵
D	DIP	Principio de inversión de la dependencia (<i>Dependency inversion principle</i>) la noción de que se debe “depender de abstracciones, no depender de implementaciones”. ⁵ La Inyección de Dependencias es uno de los métodos que siguen este principio.

Principio de responsabilidad única

Single responsibility principle

(SRP)



an NTT DATA Company

INTENTA EVITAR:

Intenta evitar que una clase tenga muchas responsabilidades forzando a dividirla en tantas clases como sea necesaria; teniendo cada clase una única responsabilidad.

APLICACIÓN Y CONSECUENCIA DE APLICAR EL PRINCIPIO:

Cada clase debe tener una única responsabilidad.

Se consigue mayor mantenibilidad y claridad en el código.

Ayuda a tener un código fácil de mantener y escalar.

DETECTAR SI INCUMPLIMOS SRP

The logo for 'everis' is located in the top right corner. It consists of the word 'everis' in a white, lowercase, sans-serif font, positioned inside a green, irregular, blob-like shape.

Si algunas de tus clases tiene más de un motivo para ser cambiada, es muy posible que estés incumpliendo este principio. Te puede dar una pista:

El número de métodos públicos:

Una clase tiene muchos métodos públicos (métodos que se muestran al exterior) puede ser un indicativo de que esa clase realiza más de una tarea.

Uso de los atributos en los métodos:

Si usamos un conjunto de atributos exclusivamente en unos métodos y otro conjunto de atributos en otro grupo de métodos, deberíamos replantearnos la posibilidad de dividir esos atributos y métodos en clases independientes.

DETECTAR SI INCUMPLIMOS SRP

The logo for everis, featuring the word "everis" in a white, lowercase, sans-serif font, positioned to the right of a green, irregular, blob-like shape.

an NTT DATA Company

Cantidad de imports:

Entre más imports necesitemos (inclusión de clases externas) mayor probabilidad de que nuestro código haga más de una sola acción.

Desarrollar una nueva funcionalidad implica cambios en la clase:

Si al desarrollar una nueva funcionalidad debemos modificar nuestras clases, es muy probable de que esté realizando más de una acción.

Número de líneas:

Si nuestra clase es muy grande, es muy probable que esté realizando más de una acción.

Déjate de historias y, ¡¡¡VAMOS AL CODIGO!!!

DESCRIPCIÓN:

Código de ejemplo donde tenemos una clase encargada del registro de usuarios y que además tiene la funcionalidad de mostrar el resultado del registro.

REPOSITORIO EN GITLAB:

https://github.com/EverisMovilidadZgz/SOLID_SRP

HASHES

e0fb5021a2c8510aec294af86f33564c7a3fa6b2 Problema.

b5e1549d7a88a65c9c3631ba659866a24b547b96 Solución.