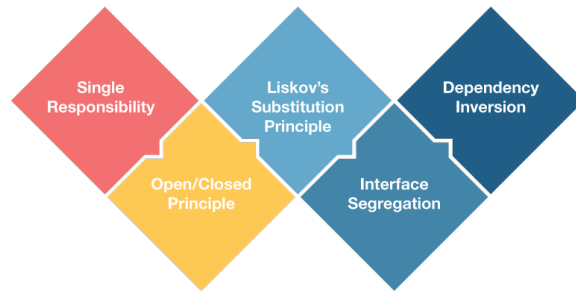


# Investigación de Solid

**S.O.L.I.D.**



# Índice

1. Pregunta: ¿Qué significa la letra "S" en SOLID y cómo se aplica en el diseño de software?.....	3
2. Pregunta: ¿Cuál es el significado de la letra "O" en SOLID y cómo se ilustra con un ejemplo práctico?.....	3
3. Pregunta: Explique el significado de la letra "L" en SOLID y proporcione un ejemplo de su aplicación en el desarrollo de software.....	3
4. Pregunta: ¿Qué implica la letra "I" en SOLID y cómo puede ilustrarse mediante un caso de uso en programación?.....	3
5. Pregunta: ¿Cuál es el significado de la letra "D" en SOLID y cómo puede aplicarse en el desarrollo de software?.....	4
6. Pregunta: ¿Por qué es importante aplicar los principios SOLID en el desarrollo de software y cómo contribuyen a escribir un código de mejor calidad?.....	4
7. Pregunta: ¿Quiénes son algunos autores o figuras prominentes que han contribuido significativamente a la popularización e implementación de los principios SOLID en el campo de la programación?.....	4
8. Pregunta: ¿Cómo pueden los principios SOLID ayudar a los equipos de desarrollo a trabajar de manera más eficiente y colaborativa?.....	5
9. Pregunta: ¿Cuáles son algunos desafíos comunes que enfrentan los desarrolladores al intentar aplicar los principios SOLID en proyectos de software existentes?.....	5
10. Pregunta: ¿Cómo pueden los principios SOLID ayudar a mitigar el riesgo de introducir errores en el código durante el desarrollo de software?.....	5

## Actividad: Investigación sobre los Principios SOLID

*Investigar sobre los principios SOLID. Relaciona las explicaciones a la programación relacionada a objetos.*

### 1. Pregunta: ¿Qué significa la letra "S" en SOLID y cómo se aplica en el diseño de software?

Single Responsibility | Principio de Responsabilidad Única

Esto hace referencia a que una clase en programación, ha de ser usada para una cosa, y, principalmente está ha de cambiar cuando cambiemos algo de sus derivados, o clases padres.

### 2. Pregunta: ¿Cuál es el significado de la letra "O" en SOLID y cómo se ilustra con un ejemplo práctico?

Open-Closed Principle | Principio Abierto-Cerrado

Básicamente, este principio nos propone que, toquemos lo menos posible del código original a través de interfaces, herencias y esas cosas.

### 3. Pregunta: Explique el significado de la letra "L" en SOLID y proporcione un ejemplo de su aplicación en el desarrollo de software.

Liskov's Substitution Principle | Principio de Sustitución de Liskov

En resumidas cuentas y a mi entender, esto a lo que se refiere es que las subclases derivadas de otras, no modifiquen en código original, más que nada para que no den conflicto más tarde.

Básicamente, dejar el código original limpio, abstracto, que los hijos sean quienes son definidos.

### 4. Pregunta: ¿Qué implica la letra "I" en SOLID y cómo puede ilustrarse mediante un caso de uso en programación?

Interface Segregation | Principio de segregación de interfaces

Este principio nos incita a que usemos múltiples interfaces concretas para una sola clase, más que nada por la flexibilidad a la hora de implementar el código.

Seré sincero, en la práctica no se explica exactamente esa flexibilidad que da el uso de esto, pero más o menos lo trato de asumir.

A mi entender, cuanto más segreguemos el código de interfaces, más podremos simplificarlo para que quede todo mucho más limpio.

**5. Pregunta: ¿Cuál es el significado de la letra "D" en SOLID y cómo puede aplicarse en el desarrollo de software?**

Dependency Inversion | Principio de inversión de dependencia

Esté es un enfoque de la programación que nos incita a que nuestro código no dependa de variables y cosas especificadas, si no de variables, abstracciones y interfaces para que, si a largo plazo queremos modificar cosas, se nos haga más sencillo y no tan engorroso.

**6. Pregunta: ¿Por qué es importante aplicar los principios SOLID en el desarrollo de software y cómo contribuyen a escribir un código de mejor calidad?**

Básicamente, son buenas praxis que nos ayudan a que nuestro código sea más cohesivo entre si mismo, hasta cierto punto, algo más fácil de leer (no siempre), y por extensión, más sencillo de modificar a largo plazo.

Simplemente, son una serie de prácticas que te ayudan a ser algo más eficiente a largo plazo a la hora de programar, hacer pruebas y demás.

**7. Pregunta: ¿Quiénes son algunos autores o figuras prominentes que han contribuido significativamente a la popularización e implementación de los principios SOLID en el campo de la programación?**

- Robert J. Martin | tío Bob – Autor de el concepto de SOLID.
- Michael Feathers – Es quien promovió el propio concepto de SOLID y lo popularizo bajo ese nombre.
- Barbara Liskov – No trabajo directamente en los principios SOLID, fue una figura fundamentales sobre la que, indirectamente, tío Bob agarro muchas de sus bases de programación orientada a objetos para los principios SOLID.
- Eric Evans
- Martin Fowler
- Kent Beck
- Sandi Metz

**8. Pregunta: ¿Cómo pueden los principios SOLID ayudar a los equipos de desarrollo a trabajar de manera más eficiente y colaborativa?**

En general, bastante, lo dicho previamente, es una manera más cohesiva de trabajar en la programación orientada a objetos, en esta ocasión, en trabajos grupales ayuda mucho en el aspecto tanto de problemas a la hora de ejecutar cosas.

Básicamente, que no hayan conflictos y demás a la hora de buildear archivos.

**9. Pregunta: ¿Cuáles son algunos desafíos comunes que enfrentan los desarrolladores al intentar aplicar los principios SOLID en proyectos de software existentes?**

Seh, a veces pueden suceder cosas como que, en lo que implementas un principio, alomejor sin darte cuenta estás chafando otro principio distinto, puede costar figurarte el ver como separas responsabilidades de interfaces o abstracciones, dificultar el como escribir el código, implementarlo en uno ya existente o incluso, tratar de aplicar estos principios a un equipo entero.

Pues, si bien el uso de estos principios se consideran como buenas praxis, no todo el mundo está familiarizado con el y aprenderlo y inculcartelo tampoco es la tarea más sencilla que pueda haber.

A parte de conflictos de enfoques y demás.

**10. Pregunta: ¿Cómo pueden los principios SOLID ayudar a mitigar el riesgo de introducir errores en el código durante el desarrollo de software?**

Pues con lo dicho previamente, ayudan tanto en el aspecto de cohesividad entre archivos, lo que viene a significar una flexibilidad más gorda para implementar y meter cosas.

Y también en el aspecto de no dejar nada al aire.

Las cosas se definen con variables por si acaso para que, cosas que dependan de otras, si una cosa se cambia no pete todo lo de 'alante.