

Principio de responsabilidad única

El **principio de responsabilidad única** o *SRP* (siglas del inglés, *Single Responsibility Principle*) en ingeniería de software establece que cada módulo o clase debe tener responsabilidad sobre una sola parte de la funcionalidad proporcionada por el software y esta responsabilidad debe estar encapsulada en su totalidad por la clase. Todos sus servicios deben estar estrechamente alineados con esa responsabilidad. Este principio está incluido en el acrónimo mnemotécnico SOLID. Robert C. Martin expresa el principio de la siguiente forma:¹

Una clase debe tener solo una razón para cambiar.

En programación orientada a objetos, se suele definir como principio de diseño que cada clase debe tener una única responsabilidad, y que esta debe estar contenida únicamente en la clase. Así:

- Una clase debería tener solo una razón para cambiar
- Cada responsabilidad es el eje del cambio
- Para contener la propagación del cambio, debemos separar las responsabilidades.
- Si una clase asume más de una responsabilidad, será más sensible al cambio.
- Si una clase asume más de una responsabilidad, las responsabilidades se acoplan.

Referencias

1. Martin, Robert C. (2003). *Agile Software Development: Principles, Patterns, and Practices* (https://books.google.com/books?id=0HYhAQAAIAAJ&redir_esc=y) (en inglés). Pearson Education. ISBN 9780135974445. Consultado el 7 de agosto de 2018.

Obtenido de «https://es.wikipedia.org/w/index.php?title=Principio_de_responsabilidad_única&oldid=130604500»

Esta página se editó por última vez el 3 nov 2020 a las 14:29.

El texto está disponible bajo la Licencia Creative Commons Atribución Compartir Igual 3.0; pueden aplicarse cláusulas adicionales. Al usar este sitio, usted acepta nuestros términos de uso y nuestra política de privacidad. Wikipedia® es una marca registrada de la Fundación Wikimedia, Inc., una organización sin ánimo de lucro.