

# Patrones Singleton

Los patrones Singleton son patrones de diseño que permiten crear una única instancia de objeto para una clase y que genera un punto de acceso global en esta misma. A pesar de que esto suena muy similar a las variables globales de Java, hay que recalcar algunas diferencias entre estos dos. En primer lugar los patrones Singleton son administrados por un método que se encarga de regular las instancias de la clase, mientras que una variable es lo que la misma palabra se refiere. (Pinzón and Paz, --)

## Ventajas

- Es más rápido y fácil de escribir que las variables globales, ya que éstas suelen estar en grandes cantidades y Singleton solamente en un método.
- Tiene un control más preciso sobre cómo y cuándo se accede a la clase.
- Puede derivarse mediante subclases, lo que le permite adaptarse con flexibilidad.

## Desventajas

- Al poder usarse de una manera espontánea, puede llevar a perjudicar el código al no ser del todo orientado a objetos.
- Pueden llegar a no ser muy compatibles con datos sensibles, ya que no todos sus cambios se pueden rastrear a otras partes del código.
- Dificultad para mantenimiento de software y para su rendimiento en casos de ser multiusuario.
- Puede llegar a ser difícil de eliminar por su única instancia y su relación con el código. (Ionos, 2021)

## Bibliografía

Ionos. "Patron singleton: una clase propia." *Digital Guide Ionos*, 19 febrero 2021,

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/patron-singleton/>.

Accessed 22 febrero 2022.

Pinzón, Mauricio Alberto, and Cristian Camilo Paz. "Singleton." *gitbooks*,

<https://fjimenezg.gitbooks.io/atributos-de-calidad-y-patrones-de-diseno/content/single-ton.html>. Accessed 22 febrero 2022.