

Patrón de diseño “Singleton”

Ventajas y desventajas

EL patrón de diseño *singleton* consta de utilizar una sola instancia de una clase durante la ejecución del programa impidiendo mediante diseño la creación de otro objeto del mismo tipo. Esto tiene sus ventajas al momento de optimizar los recursos de el equipo en donde se ejecute, además de evitar posibles errores por tener múltiples objetos del mismo tipo.

Entre las desventajas de este patrón de diseño está el no poder utilizar herencia ni técnicas ligadas a ello, ya que la implementación de clases hijas de una que implemente *singleton* infiere que habrán muchas instancias del mismo tipo de la clase en el programa. Además, el uso del patrón *singleton* requiere la responsabilidad de crear una instancia de las clases en que se implementa, en caso contrario se convertiría en recursos mal utilizados y código muerto.

Uso en este programa

El uso del patrón de diseño *singleton* está justificado debido a que no es necesario instanciar más de una vez la clase Calculadora (a la que se le aplica el patrón de diseño), además se implementa un tipo de programación defensiva que no permite en ningún momento crear dos objetos de dicha clase.