Taller de desarrollo de software 2

Integrantes

Jose David Martinez Meza

Carolina Blanquicett Devoz

David Reyes Padilla

10/08/2024

# Código Limpio: Tema 10 – Clases

## Organización de Clases

Las clases deben estar organizadas de una manera específica para facilitar la comprensión. Deben comenzar con una lista de variables. Primero las variables de clase (estáticas), seguidas de las variables de instancia. Luego, deben seguir las funciones públicas y, finalmente, las funciones privadas. Esta organización permite a los lectores del código entender rápidamente qué variables modifica una función.

## Encapsulación

La encapsulación es un principio fundamental de la programación orientada a objetos. Las variables y funciones de utilidad deben ser privadas para mantener la encapsulación. Esto significa que los detalles de implementación de una clase deben estar ocultos para los otros objetos. Solo los métodos públicos, que forman la interfaz de la clase, deben ser accesibles.

## Tamaño Reducido

Las clases deben ser pequeñas. Pero, ¿qué significa “pequeño”? En este contexto, pequeño no se refiere a la cantidad de líneas de código, sino a la cantidad de responsabilidades. Según el Principio de Responsabilidad Única (SRP), una clase debe tener una, y solo una, razón para cambiar. Esto significa que una clase debe tener una única tarea o responsabilidad.

## Cohesión

La cohesión se refiere a cuán estrechamente están relacionadas las responsabilidades de una clase. Las clases deben tener un número reducido de variables de instancia. Los métodos de una clase deben manipular una o más de esas variables. Esto ayuda a mantener la cohesión, lo que significa que los métodos y variables de una clase deben estar estrechamente relacionados y deben depender unos de otros.

## Conclusión

El objetivo de seguir estos principios es escribir un código que sea fácil de leer, entender y mantener. Un código limpio minimiza el costo de los cambios futuros y la posibilidad de introducir errores.