ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

**Taller de Principios SOLID – Diseño de Software**

**MSc. David Jurado**

# Objetivos:

* Identificar violaciones a los principios SOLID.
* Corregir código que viole principios SOLID.
* Utilizar Git como herramienta de apoyo al trabajo colaborativo.

# Instrucciones

* Trabajar en grupos de máximo 5 integrantes.
* Debe utilizarse un repositorio de Github para este nuevo grupo.
* En el paquete desacargado hay un proyecto compuesto por 10 paquetes (2 por cada principio, mal y bien).
  + Los paquetes que terminan con **‘violation’** en su nombre, contiene clases e interfaces que viola un principio SOLID. Las tres primeras letras del nombre del paquete le indican el principio que se está violando.
* Dentro de este documento, deben indicar como cada subpaquete está violando el principio SOLID correspondiente. Todos deben estar de acuerdo en el problema y como solucionarlo.
* Cada integrante debe corregir por lo menos un subpaquete, colocando la solución en el subpaquete ‘**solution’** respectivo y subirlo al repositorio remoto de Github con su propio usuario.

# Conteste

Explique brevemente como cada principio es violado en el código analizado y cuál sería la posible solución.

* SRP: Se viola el principio de single responsibilty porque la clase ‘MaquinaHaceTodo’ debería enfocarse en hacer una sola responsabilidad y no todo; por lo que la solución seria crear una clase padre ‘MaquinaGenerica’ de la cual heredaran sus clases hijas las cuales tienen una responsabilidad especifica.
* MaquinaHaceCarro se encarga de crear carros
* MaquinaHaceBicicleta se encarga de crear bicicletas
* MaquinaControlaHabitacion gestiona una habitación por ejemplo: la temperatura, luces.
* MaquinaOficinaUtil se encarga de imprimir informes y servir café.
* OCP: La violacion de este principio ocurre debido a que cuando se desea implementar un nuevo tipo de giro la clase “testeadorDeGiros” se modificara. Para esto es mejor crear una interfaz denominada “TiposGiros” la cual sera implementada por todas las clases de giros para cada vehiculo.
* LSP:
* ISP:
* DIP:

# Entregable

* El enlace al repositorio de GitHub con las soluciones y este archivo Word con el análisis grupal.
* Subir el enlace a Sidweb.