Taller Principios SOLID

# 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 los grupos del proyecto.
* Usted trabajará con el proyecto en un repositorio de Github. El proyecto está compuesto por 10 paquetes (2 por cada principio, mal y bien).
* Realizar un Fork al repositorio del grupo, desde el repositorio llamado: <https://github.com/djurado/solid>
* Cada integrante debe haber solucionado por lo menos uno de principios y debe verse reflejado en el usuario de los commits al repositorio.
* Los paquetes que contienen **‘mal’** 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.
* Analice el código y reestructure las clases, de modo que el principio SOLID ya no sea violado. Agregue su propuesta de solución en sub-paquetes **‘bien’** respectivos.

# Conteste

Explique brevemente como cada principio es violado en el código analizado.

* **SRP:**
* **OCP:**
* **LSP:** La clase MotoAcuatica implementa la interfaz VehiculoAcuatico y posee el comportamiento para navegar, que es propio también de Velero, sin embargo, una moto acuática tiene como atributo el estar encendido o apagado. El método sobrescrito navegar para una moto acuática lanza una excepción, lo cual viola el principio de sustitución de Liskov dado que una clase derivada presenta un comportamiento diferente al esperado. Por lo tanto, para poder obtener un comportamiento adecuado sin que sea estrictamente necesario saber a qué tipo de vehículo acuático nos referimos se debe de modificar el método navegar para que en caso de que la moto esté apagada, modifique ese comportamiento a encendido y pueda navegar.
* **ISP:** La interfaz EntrenamientoPokemon contiene código para entrenar Pokemones en general, y para habilidades de fuego y agua que no todos los Pokemones tienen. Por lo tanto, si un Pokemon quiere implementar el entrenamiento general, también tiene que implementar los entrenamientos de agua y fuego, que no necesita. La solución es separar las interfaces en habilidades generales, de agua y de fuego (en general, una interfaz por cada habilidad diferente) y hacer que cada Pokemon implemente las interfaces de las habilidades que tiene.
* **DIP:**

# Entregable

* El enlace al repositorio de GitHub donde colocaron su solución a cada principio y este archivo Word con su análisis grupal.
* Subir el enlace a Sidweb.