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: En la clase Pokemon existe un método el cual trabaja con la vida del Pokemon. Por ello, la clase Pokemon no tiene como responsabilidad curarse, no es un método que deba manejar la clase Pokemon.
* OCP: Este principio indica que debe estar abierto a extensiones y cerrado a modificaciones, por ejemplo, si añado una herencia solo debe hacerse cambios en la nueva clase. Pero en el caso del proyecto, si tengo la necesidad de añadir un nuevo ataque de Pokemon, me veré obligado a hacer cambios en la clase ManejadorAtaque para colocar la vida respectiva del pokemon y de la misma forma con el Enum Ataque, debo añadir el nuevo tipo de ataque.
* LSP:
* ISP:
* 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.