**Prueba Parcial 2**

Nombre: Cristhian Villamarin

Fecha: 22/07/2021

* **Single Responsability**

Al observar meticulosamente pude observar que la clase longitud tienes mas de una responsabilidad por lo cual se rompe el primer principio.

Para cumplir con este principio cree las clases de cada tipo de conversión de este modo cada clase tiene una sola cosa por la cual cambiar y todo el código no está almacenado en una sola clase.

* **Open/Closed Principle**

Luego de la solución del problema si se puede realizar la extencion de una clase, esto se realizo a través del uso de una interfaz la cual nos permite crear mas clases sin necesidad de modificar las ya creadas.

* **Liskov Substitution Principle**

Este principio lo cumplí mediante el uso de la interfaz ya que se realizó el programa de tal forma que se pueda llamar directamente al método sin necesidad de llamar a la interfaz.

* **Interface Segregation Principle**

En este caso no se rompió este principio ya que el programa propuesto solo tiene una funcion por lo cual no hubo necesidad de crear interfaces innecesarias.

Solo fue necesario implementar uno para cambiar las unidades de medidas correspondientes.

* **Dependency Inversion Principle**

Mediante el uso de la interfaz logre que las clases de bajo nivel no incidan en las clases de alto nivel(Longitud) cumpliendo así lo mencionada en el principio.