**1. Compare la arquitectura inicial con la final en términos de acoplamiento y cohesión**

En la arquitectura inicial el código es fuertemente acoplado ya que en la clase de área calculator calcula las áreas sin embargo eso hace que se cambien muchas cosas y responsabilidades de acuerdo al tipo de figura implementada algo que no permite la “s” en los principios solid ya que una clase debería tener una razón para cambiar y en cuanto a la cohesión, pues como se mencionó antes si quiero calcular el área de una figura entonces tendré que especificarlo en la clase área calculator pero al hacerlo hace que se pierda legibilidad y escalabilidad, cuando seria mejor implementar una interfaz con los objetos concretos

**2. Explique cómo el diseño refactorizado utiliza el polimorfismo para cumplir con el principio Abierto/Cerrado.**

Para utilizar y visualizar el método de sumar las áreas de acuerdo a la figura en la clase areacalculator, en ese momento se necesita en el parámetro digamos el tipo de figura que al momento de llamarlo no importa si es un triángulo, cuadrado o circulo, ese método lo calcula ya que las clases concretas lo implementan a su manera, cambiando la forma en que se utiliza el método, y para cumplir el principio de Abierto y Cerrado, dice que debe ser abierto para la extensión y cerrado para la modificación, en este caso si quiero crear un rombo no tendré que modificar la interfaz shape, simplemente utiliza la estructura ya creada sin dañar otras figuras.

**3. Analice cómo este nuevo diseño ejemplifica el Principio de Inversión de**

**Dependencias (DIP), donde los módulos de alto nivel no dependen de los de bajo**

**nivel, sino que ambos dependen de abstracciones**

En ese caso al momento de implementar la interfaz shape, el area calculator no le interesa los detalles de como se calcula ese area, es decir no depende de las clases concretas como circulo, triangulo o cuadrado, entonces lo que interesa es la figura que se llame, ya que no es su responsabilidad calcular áreas sino que llamarlas, de ahí que dependan de abstracciones como Shape, puesto que al ser de alto nivel entonces van separadas en cuanto a la lógica de negocio