Ejercicio 1

1. El primer ejercicio viola el patrón de ISP ya que existen implementaciones de la clase abstracta que no implementan algunos de los métodos propuestos por la clase abstracta, ya que no son aplicables a la clase de la implementación. Aplicado en este caso, no todos los animales vuelan.



Ejercicio 2

1. El segundo ejercicio viola el patrón SRP ya que la clase Impresora tiene múltiples responsabilidades que no todas deberían ser de ella.
2. 

Ejercicio 3

1. El tercer ejercicio incumple el principio SRP (Single Responsibility Principle) ya que la clase BaseDeDatos tiene tanto la responsabilidad de guardar los datos como de enviar un correo electrónico.



Ejercicio 4

El cuarto ejercicio viola el principio de segregación de interfaces (ISP) ya que el comportamiento de Robot podría estar dividido en más de una interfaz, para evitar tener una clase.



Ejercicio 5:

1. El quinto ejercicio no cumple el principio SRP, no es responsabilidad del cliente crear un pedido.
2. 

Estre creo q es el isp

Ejercicio 6:

1. Se viola el principio LSP, ya que PatoDeGoma hereda de IPato, pero PatoDeGoma no puede implementar Volar().





Ejercicio 7:

1. Este ejercicio viola el principio ISP, ya que ReadDatabase está obligado a implementar WriteData, lo cual no tiene ningún uso para la clase.
2. 



Ejercicio 9:

1. Este ejercicio viola el principio SRP, ya que en la clase User hay responsabilidades que no deberían ser de ella.
2. 

Ejercicio 10:

1. Este ejercicio viola el principio OCP, ya que para poder agregar una nueva forma de reproducir música se debe modificar la clase MusicPlayer, también podríamos decir que viola el principio SRP ya que la clase que se representa tiene varias responsabilidades que claramente se podría separar en diferentes clases

public class Mp3PLayer{

public void Play(){

//Play music

}

}

public class WavPLayer{

public void Play(){

//Play music

}

}

public class FlacPLayer{

public void Play(){

//Play music

}

}