**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**TAREA:** SOLID

**DISEÑO DE SOFTWARE**

**INTEGRANTES**

JOHANN RAMÍREZ

LUIS BARRERA

**PROFESOR:** DAVID JURADO

Tabla de contenido

[Primer Repositorio 3](#_Toc150723020)

[Incumplimiento: Principio de responsabilidad única (SRP) 3](#_Toc150723021)

[Incumplimiento: Principio Abierto/Cerrado (OCP) 3](#_Toc150723022)

[Incumplimiento: Principio de sustitución de Liskov (LSP) 4](#_Toc150723023)

[Incumplimiento: Principio de segregación de interfaz (ISP) 5](#_Toc150723024)

[Incumplimiento: Principio de inversión de dependencia (DIP) 6](#_Toc150723025)

[Solución planteada 6](#_Toc150723026)

[Evidencia reporte del primer repositorio 11](#_Toc150723027)

[Segundo Repositorio 12](#_Toc150723028)

[Incumplimiento: Principio de responsabilidad única (SRP) 12](#_Toc150723029)

[Incumplimiento: Principio Abierto/Cerrado (OCP) 12](#_Toc150723030)

[Incumplimiento: Principio de sustitución de Liskov (LSP) 13](#_Toc150723031)

[Incumplimiento: Principio de segregación de interfaz (ISP) 13](#_Toc150723032)

[Incumplimiento: Principio de inversión de dependencia (DIP) 14](#_Toc150723033)

[Solución planteada 14](#_Toc150723034)

[Evidencia reporte del segundo repositorio 16](#_Toc150723035)

[Tercer Repositorio 17](#_Toc150723036)

[Incumplimiento: Principio de responsabilidad única (SRP) 17](#_Toc150723037)

[Incumplimiento: Principio Abierto/Cerrado (OCP) 17](#_Toc150723038)

[Incumplimiento: Principio de sustitución de Liskov (LSP) 18](#_Toc150723039)

[Incumplimiento: Principio de inversión de dependencia (DIP) 18](#_Toc150723040)

[Solución planteada 19](#_Toc150723041)

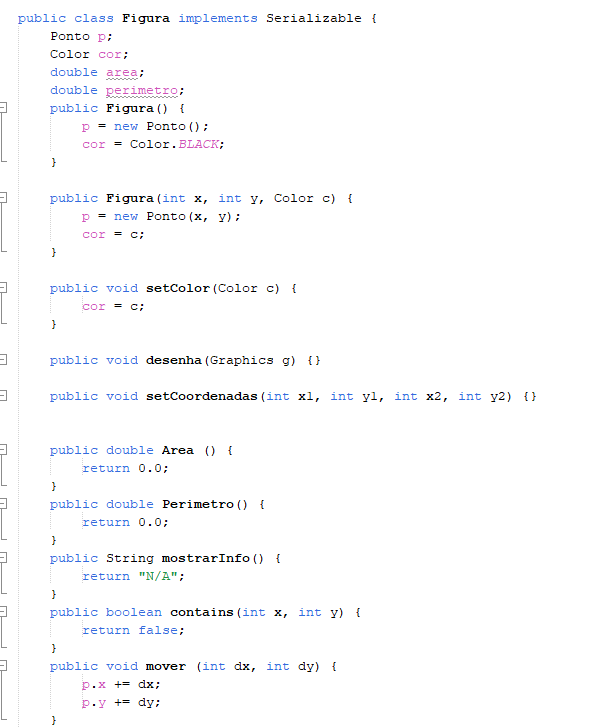
[Evidencia reporte del tercer repositorio 20](#_Toc150723042)

# Primer Repositorio

1er repositorio url: <https://github.com/miguelsoeiro/Editor-GeometricalFigures>

## Incumplimiento: Principio de responsabilidad única (SRP)

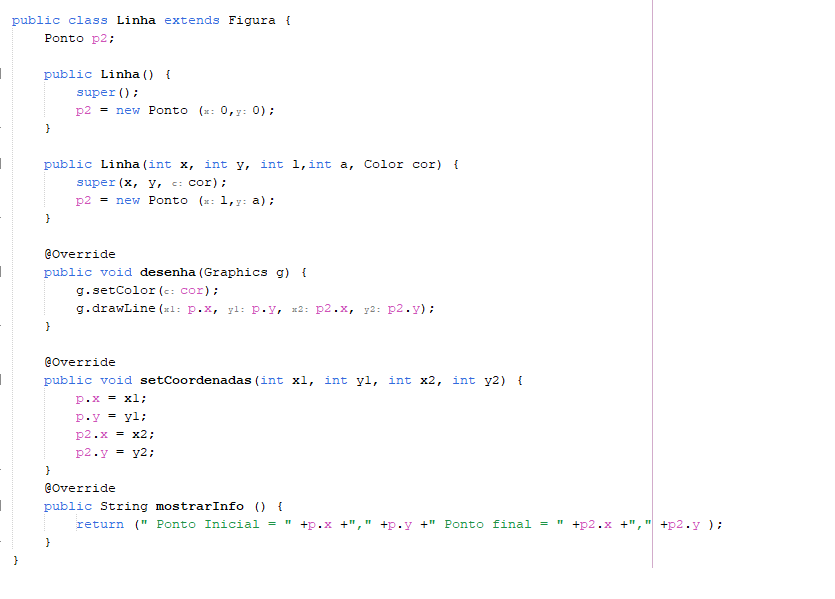
## Incumplimiento: Principio Abierto/Cerrado (OCP)



Argumentación SRP: Se incumple porque la clase figura solo debería tener el trabajo de la creación de objetos tipo figuras, sin embargo, se puede observar que tiene diferentes métodos que hace que la clase tenga más de un trabajo tales como Area(), mostrarInfo(). Desenha(), etc.

Argumentación OCP: Se incumple porque la clase figura esta siendo modificada agregando otros métodos cuando su objetivo principal es establecer los atributos básicos de una figura, esos métodos no deberían estar ubicados dentro de la clase figura, sino crear nuevas clases y estas deben ser heredadas o implementadas. Por ejemplo, el método de área(), podría ser una clase que es heredada por todas las figuras geométricas

## Incumplimiento: Principio de sustitución de Liskov (LSP)

.Argumentación: La clase Linha (Linea) es una clase derivada de la clase Figura, en la clase Figura se implementa el método Area(), el cuál es un método que no esta implementado en la clase Linha, por lo tanto, la clase derivada no es completamente sustituible por la clase base.

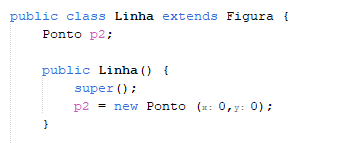
## Incumplimiento: Principio de segregación de interfaz (ISP)

A screenshot of a computer program

Description automatically generated

Argumentanción: Muchas clases heredan de la clase Figura, sin embargo, muchas de estas clases no necesitan implementar todos los métodos de esta clase, por ejemplo, la clase línea no debería implementar el método área o perímetro, por eso se deberían crear diferentes interfaces con métodos específicos.

## Incumplimiento: Principio de inversión de dependencia (DIP)



Blue text on a white background

Description automatically generated

Argumentanción: Se puede observar que la clase Linha y Figura tienen como atributo una instancia de la clase Ponto, pero ponto es una concreción, no es una abstracción, por lo que estaría incumpliendo con el principio.

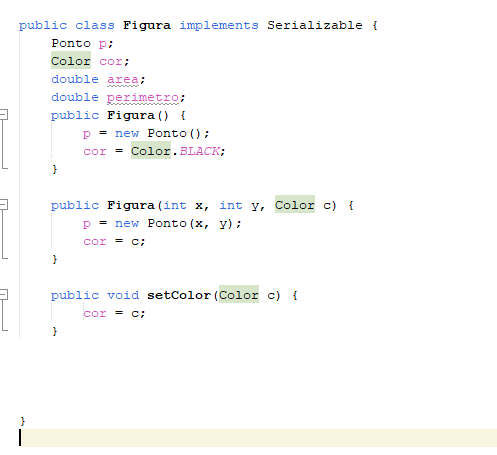
## Solución planteada

**UML planteado**

A diagram of a computer

Description automatically generated

**Rediseño SRP**



**Rediseño OCP**

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

**Rediseño LSP**

A screenshot of a computer program

Description automatically generated

**Rediseño ISP**

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A black and white image of a figure

Description automatically generated

**Rediseño DIP**

A black circle with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

## Evidencia reporte del primer repositorio

A screenshot of a computer

Description automatically generated

# Segundo Repositorio

2do repositorio url: <https://github.com/wafaaelmaandy/Car-Store.git>

## Incumplimiento: Principio de responsabilidad única (SRP)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

No cumple con principio de responsabilidad única debido a que la clase CarStoreApp posee métodos no integrales como randomBoolean(), randomName(), etc…

## Incumplimiento: Principio Abierto/Cerrado (OCP)

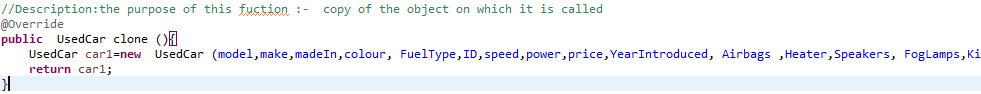


Ilustración 1: En la clase "UsedCar"

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Ilustración 2: En la clase "NewCar"

El método clone() se repite varias veces

## Incumplimiento: Principio de sustitución de Liskov (LSP)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

La clase derivada UsedCar puede lanzar una excepción que newCar no puede.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Incumplimiento: Principio de segregación de interfaz (ISP)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Hay un problema en que no todos los métodos de ICar que son necesarios en UsedCar, lo son en newCar.

## Incumplimiento: Principio de inversión de dependencia (DIP)

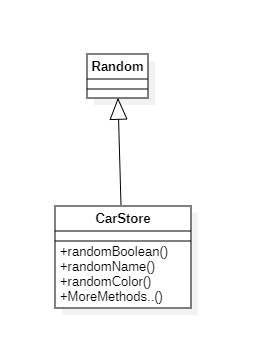
## Solución planteada

**Rediseño SRP**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se crea una nueva clase para todos los elementos random y se quitan todos los métodos random de la clase CarStoreApp.



**Rediseño OCP**

Texto

Descripción generada automáticamente

Se crea una nueva clase Clone para todas las clases que usan el método clone().

A diagram of a computer code

Description automatically generated

**Rediseño LSP**

**Texto

Descripción generada automáticamente**

Ponemos ese if statement para que cuando el auto es llamado en el main, en cualquier caso se cumpla el método.

**Rediseño ISP**

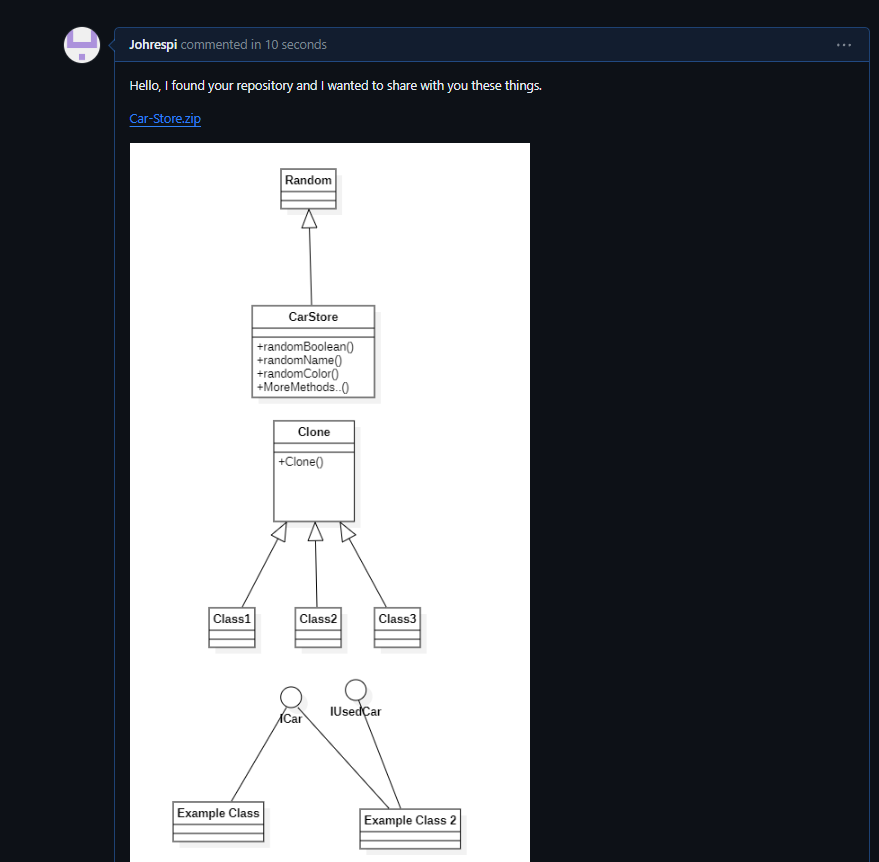
Texto, Carta

Descripción generada automáticamente

Se separa la interfaz en dos para que newCar no tenga que usar los métodos que no necesitaA diagram of a car

Description automatically generated

## Evidencia reporte del segundo repositorio



# Tercer Repositorio

3er repositorio url: https://github.com/iagoizi/poo-petshop-java.git

## Incumplimiento: Principio de responsabilidad única (SRP)



Aquí, los métodos logIn() y logOut() pueden ser removidos a otra clase para que la clase “PetShop.java” tenga menos responsabilidades.

## Incumplimiento: Principio Abierto/Cerrado (OCP)

Logotipo

Descripción generada automáticamente

Ilustración 3:Tomado de la clase "Servico"

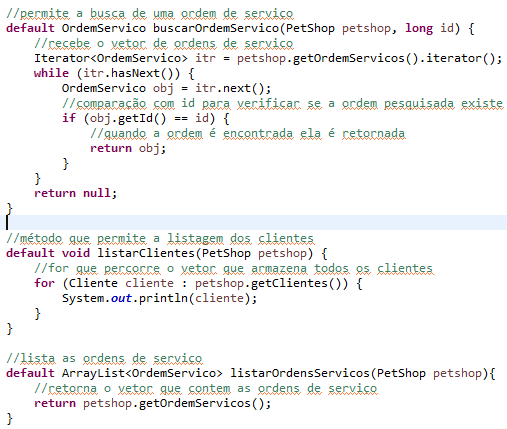
Imagen que contiene Forma

Descripción generada automáticamente

Ilustración 4:Tomado de la clase "Produto"

Hay varios métodos de clonar diferentes, se los puede poner como métodos de una clase que solo tenga el deber de implementar el método clone y las otras clases podrán heredar de esta clase.

## Incumplimiento: Principio de sustitución de Liskov (LSP)



El método listarClientes() puede separarse de esa interfaz para hacer interfaces más específicas.

## Incumplimiento: Principio de inversión de dependencia (DIP)

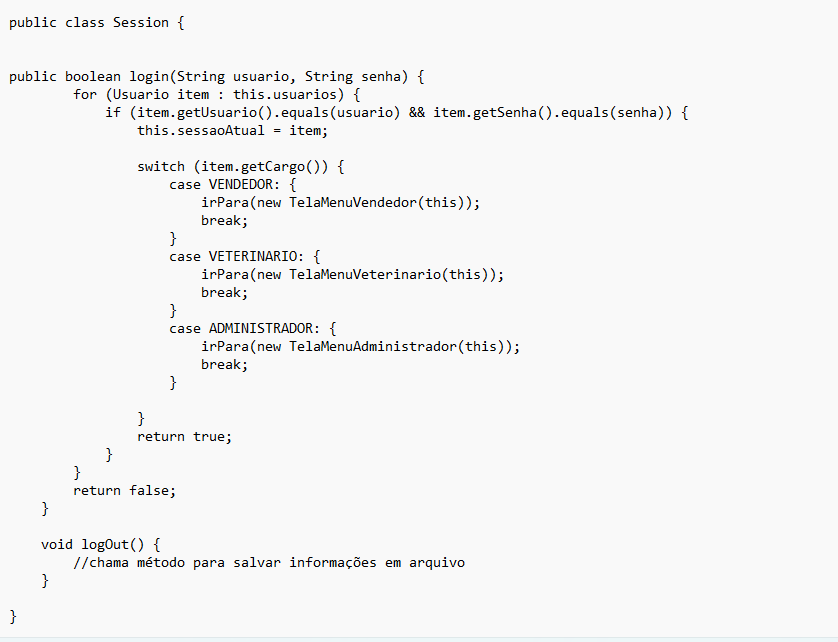
Imagen que contiene Logotipo

Descripción generada automáticamente

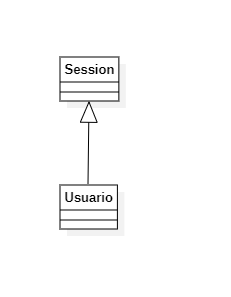
La clase orden de servicio depende de una concreción (en este caso la clase “Servicio”) cuando podría depender de una abstracción.

## Solución planteada

**Rediseño SRP**



Se crea una nueva clase para específicamente el log in y log out.

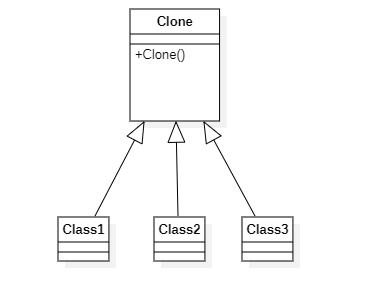


**Rediseño OCP**

Texto

Descripción generada automáticamente

Se crea una nueva clase Clone para todas las clases que usan el método clone().



## Evidencia reporte del tercer repositorio

