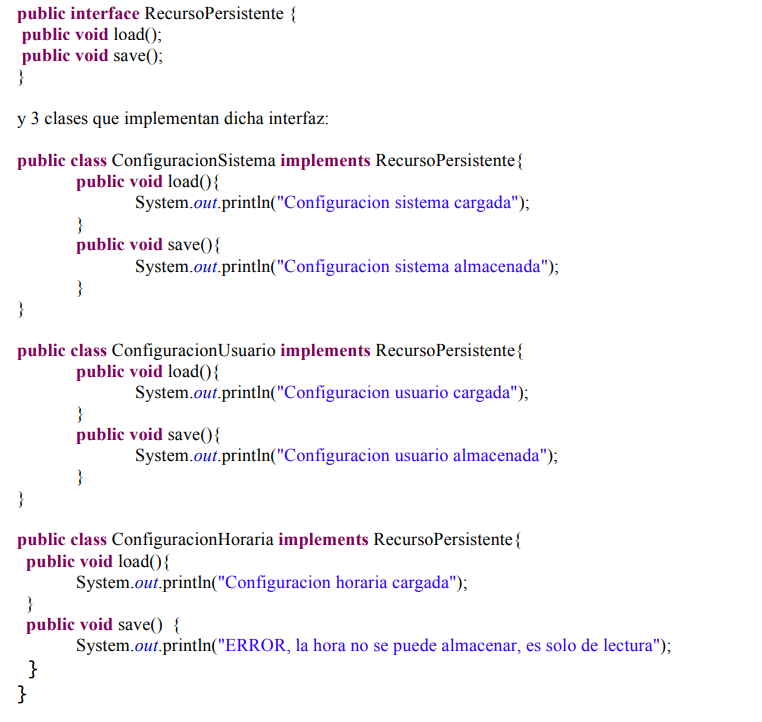
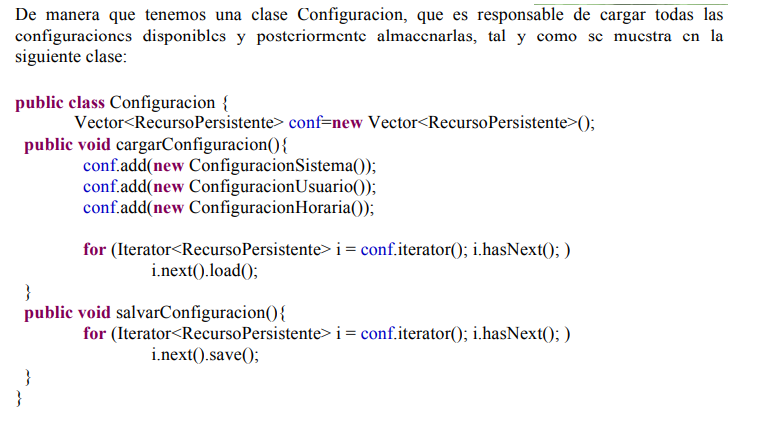
**Nombre:** Sergio Fernando Roncancio Carrero

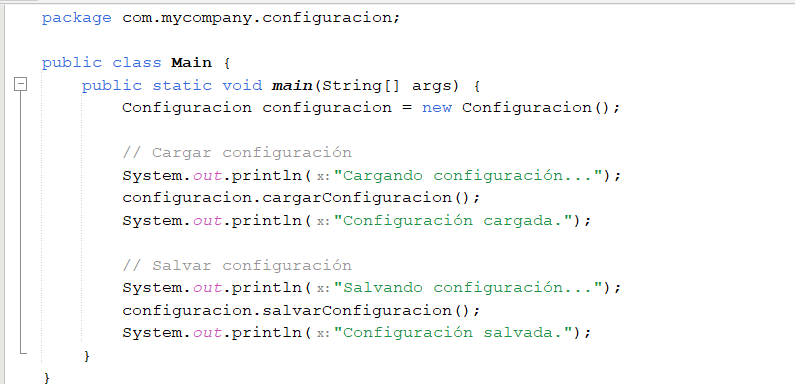
**Código:** 67000771

**Taller principios Solid**

**** ****

**Consultas:**

**Crea un programa principal que ejecute los métodos de la clase Configuración**



**Cumple la clase Configuración en Principio OCP. Justifica la respuesta.**

* El código de configuraciones no cumple el principio OCP, puesto que la razón es que cada vez que se agrega un nuevo tipo de configuración, se modifica directamente de la clase configuración para agregar la creación una nueva instancia de este mismo tipo. A lo que habla y se define que la clase configuración está abierta a modificaciones cada vez que se introduce un nuevo tipo de configuración

**Cumple la clase Configuracion el Principio de Liskov. Justifica la respuesta.**

* El código cumple con el principio de Liskov crea instancias de ConfiguracionSistema, ConfiguracionUsuario y ConfiguracionHoraria, que son subtipos de RecursoPersistente, y luego las utiliza en los métodos cargarConfiguracion() y salvarConfiguracion(). Si estas subclases cumplen con las expectativas de la superclase RecursoPersistente, entonces se cumple el principio

**Explica de forma general (independientemente del ejemplo) cual es el problema y la solución propuesta.**

* La solución aquí es adaptar la interfaz en función de lo que cada cliente necesita ( [Principio de segregación de interfaz o ISP](http://www.objectmentor.com/resources/articles/isp.pdf) )

