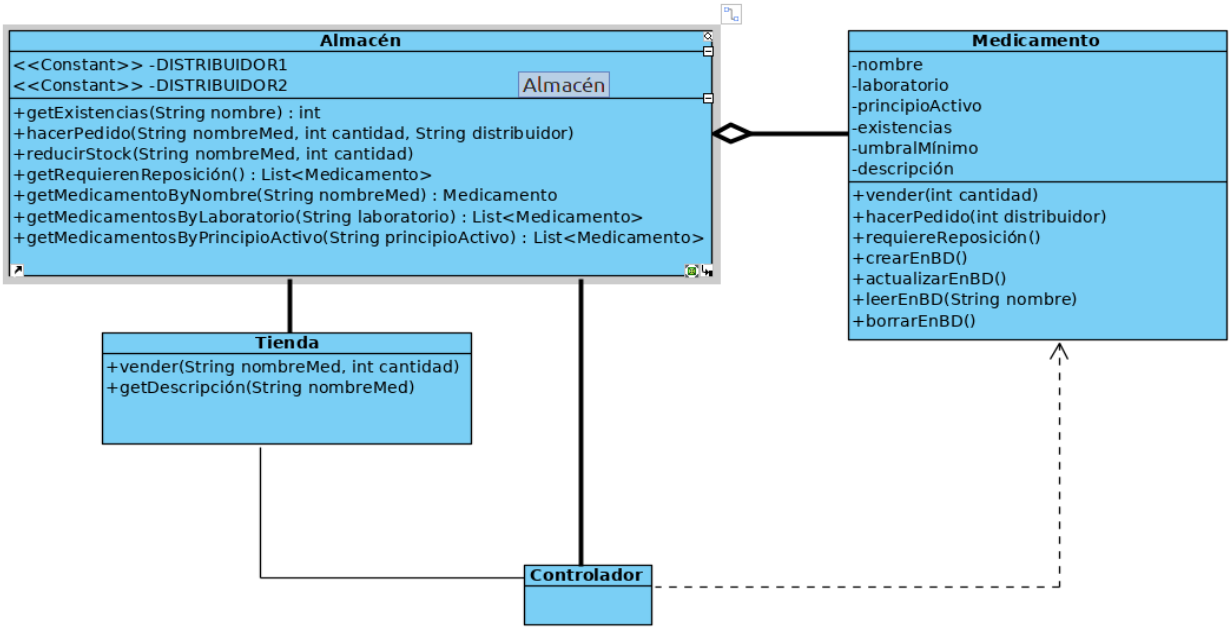
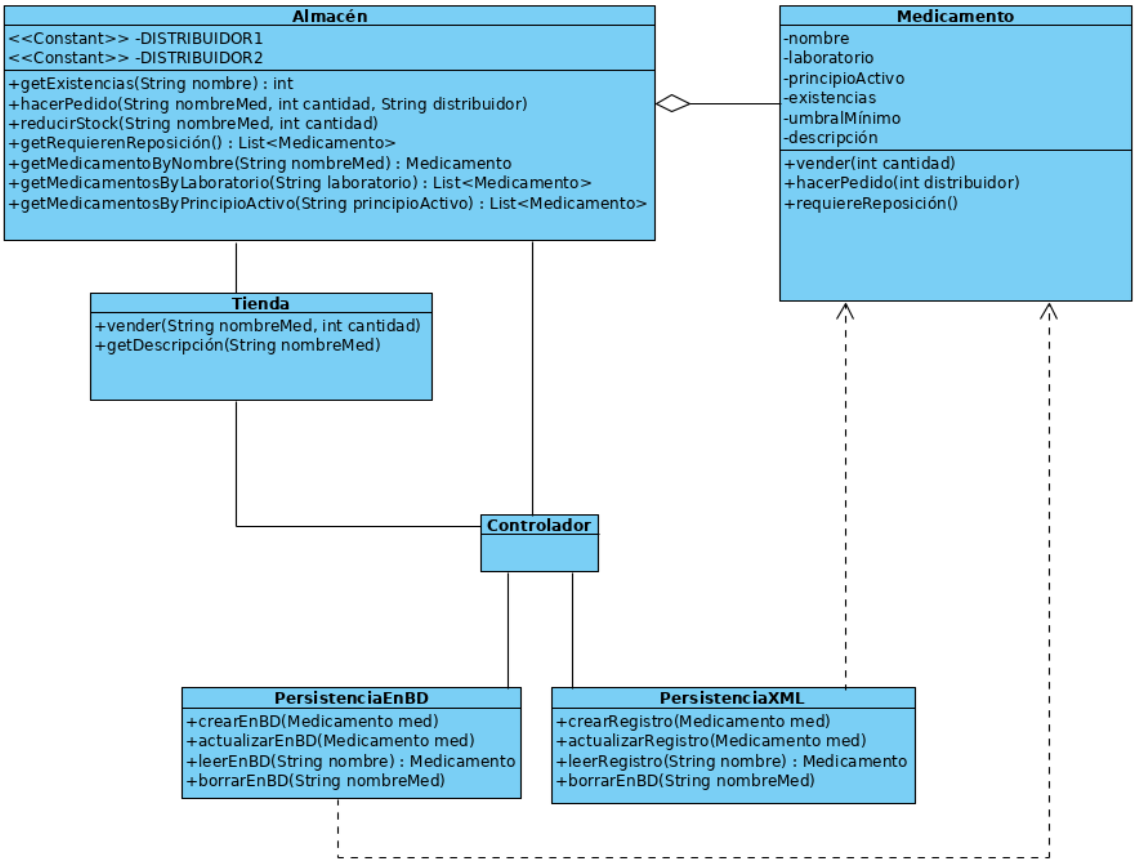
SRP

Cumple el siguiente modelo SRP. ¿Qué cambios se pueden hacer?



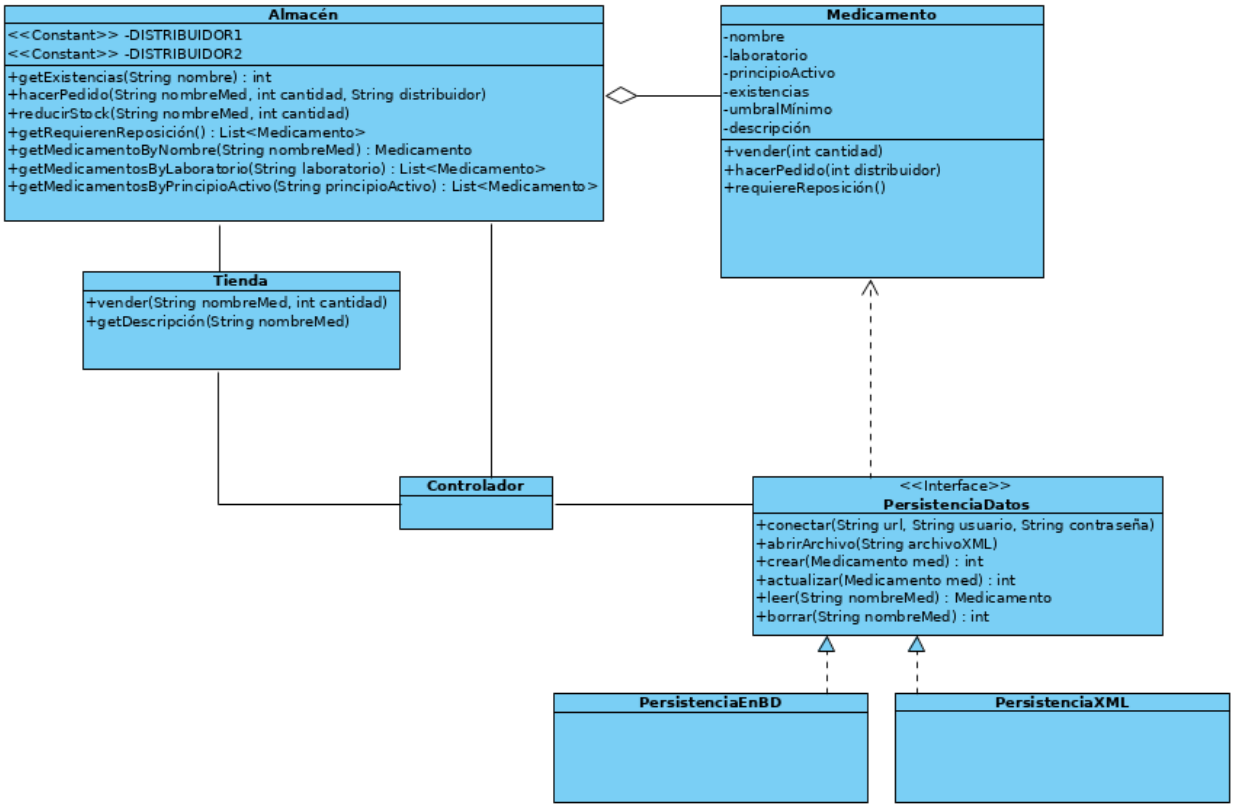
OCP

Actualmente se utiliza como motor de persistencia una base de datos SQL o bien un documento CSV, dependiendo de la configuración. Pero esto es algo que puede variar, añadiendo más tipos de métodos de almacenamiento ¿Qué inconvenientes tiene el siguiente diseño?



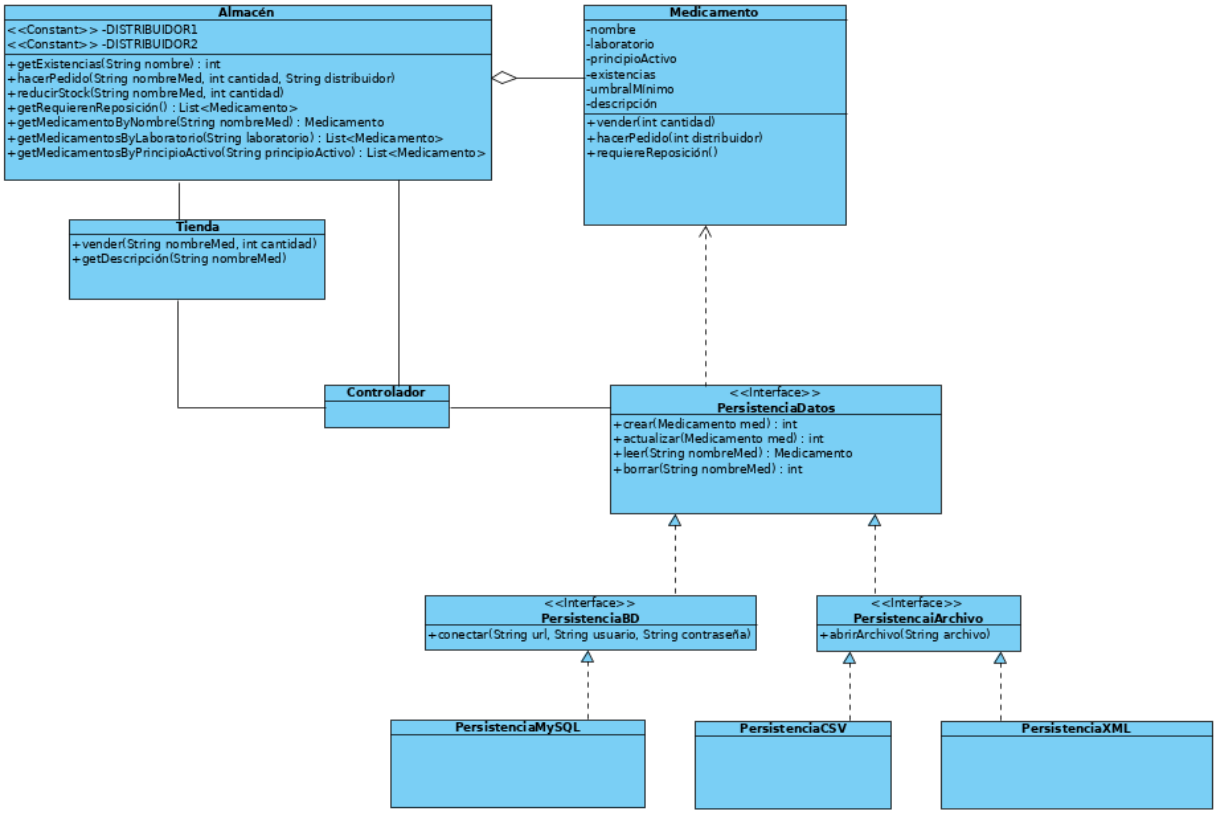
LSP

El siguiente diseño pretende generalizar la persistencia, para garantizar el principio OCP. ¿Cumple LSP?



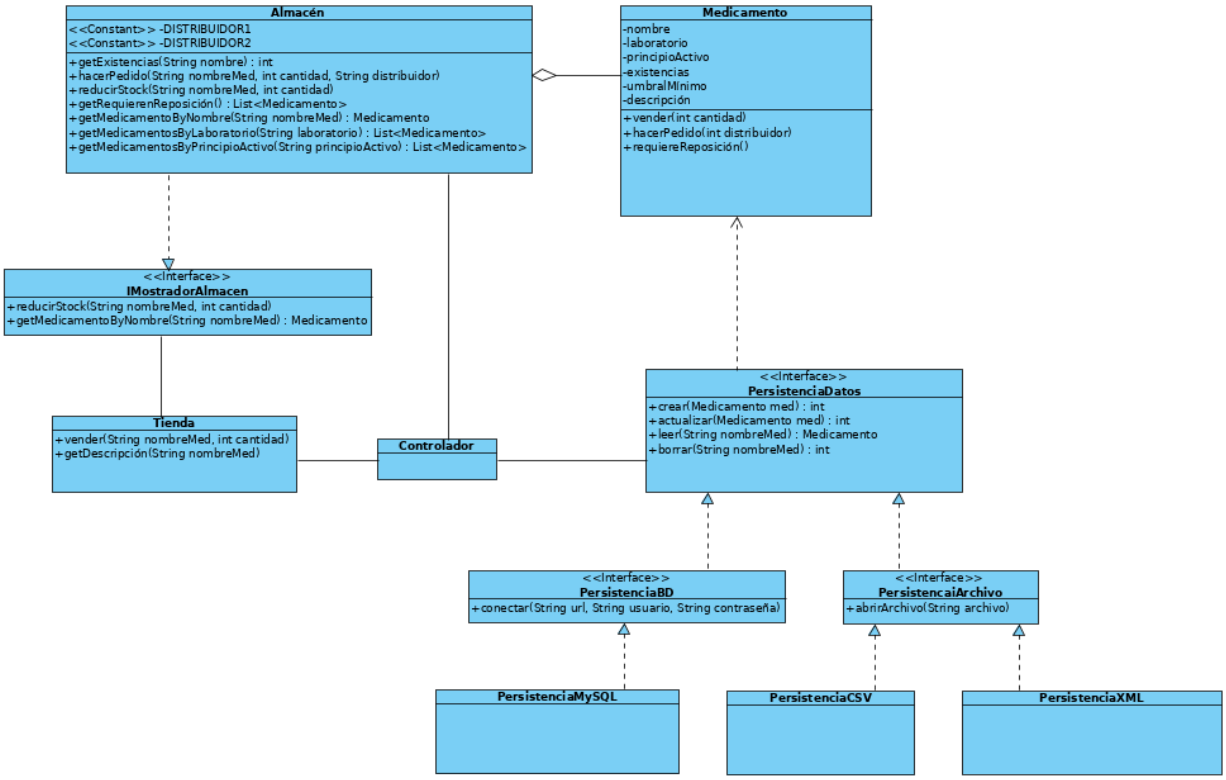
ISP

El nuevo diseño ha eliminado el problema en la interfaz *PersistenciaDatos*. Sin embargo, se incumple ISP en alguna parte del diseño. Hay una clase que tiene acceso a más métodos de los necesarios. ¿De qué clase se trata?



DIP

Tras segregar la interfaz *ImostradorAlmacen* el diseño queda así:

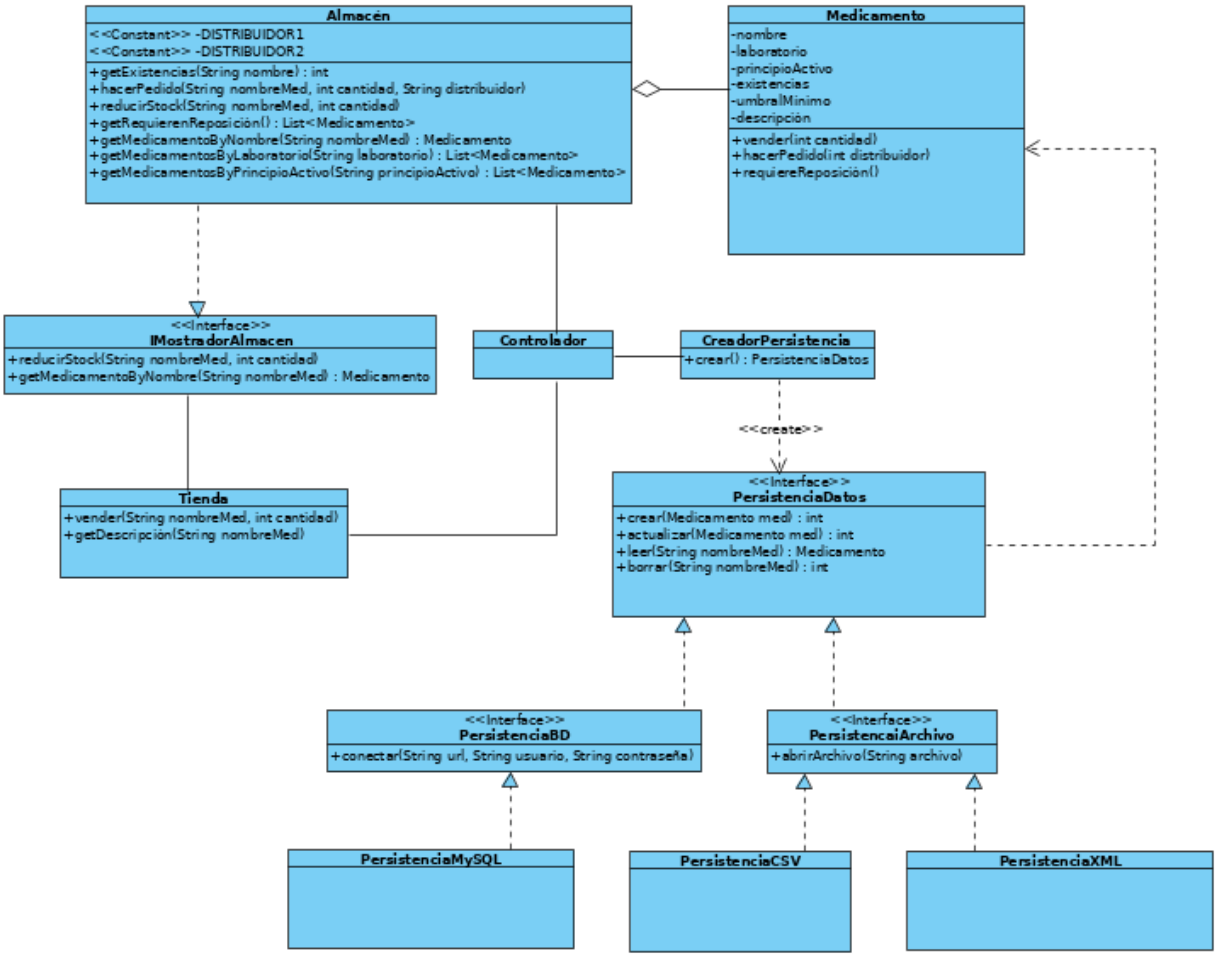


El controlador crea un clase para gestionar la persistencia directamente, con un código similar a este:

```
PersistenciaDatos motorPersistencia;

switch (tipoPersistencia){
        case “mysql”:
                motorPersistencia = new PersistenciaMySQL();
                String[] datosConexion = getDatosConexionMySQL();
                ((PersistenciaDB) motorPersistencia).conectar(datosConexion[0],
                                                              datosConexion[1],
                                                              datosConexion[2]);
                break;
        case “csv”:
                motorPersistencia = new PersistenciaCSV();
                String nombreXML = getNombreArchivoXML();
                ((PersistenciaArchivo) motorPersistencia).abrirArchivo(nombreXML);
                break;
        // ...
}
```

¿Crees que cumple DIP? ¿Cómo se puede solucionar?


Almacén
<<Constant>> -DISTRIBUIDOR1
<<Constant>> -DISTRIBUIDOR2
+getExistencias(String nombre) : int
+hacerPedido(String nombreMed, int cantidad, String distribuidor)
+reducirStock(String nombreMed, int cantidad)
+getRequierenReposición() : List<Medicamento>
+getMedicamentoByNombre(String nombreMed) : Medicamento
+getMedicamentosByLaboratorio(String laboratorio) : List<Medicamento>
+getMedicamentosByPrincipioActivo(String principioActivo) : List<Medicamento>
Medicamento
-nombre
-laboratorio
-principioActivo
-existencias
-umbralMinimo
-descripción
+vender(int cantidad)
+hacerPedido(int distribuidor)
+requiereReposición()
<<Interface>>
IMostradorAlmacen
+reducirStock(String nombreMed, int cantidad)
+getMedicamentoByNombre(String nombreMed) : Medicamento
Controlador
CreadorPersistencia
+crear() : PersistenciaDatos
<<create>>
Tienda
+vender(String nombreMed, int cantidad)
+getDescripción(String nombreMed)
<<Interface>>
PersistenciaDatos
+crear(Medicamento med) : int
+actualizar(Medicamento med) : int
+leer(String nombreMed) : Medicamento
+borrar(String nombreMed) : int
<<Interface>>
PersistenciaBD
+conectar(String url, String usuario, String contraseña)
<<Interface>>
PersistencaiArchivo
+abrirArchivo(String archivo)
PersistenciaMySQL
PersistenciaCSV
PersistenciaXML