**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**TAREA DE PRINCIPIOS SOLID**

DISEÑO DE SOFTWARE

**GRUPO 04:**

**BRYAN SEGOVIA MARISCAL**

**EDISON LÓPEZ ESPINOZA**

**LUIS DONOSO BRAVO**

**GÉNESIS CORTEZ GAVILANES**

**NOVIEMBRE 2021**

Tabla de contenido

[Principio SRP 2](#_Toc87393015)

[Principio OCP 5](#_Toc87393016)

[Principio LSP 6](#_Toc87393017)

[Principio ISP 6](#_Toc87393018)

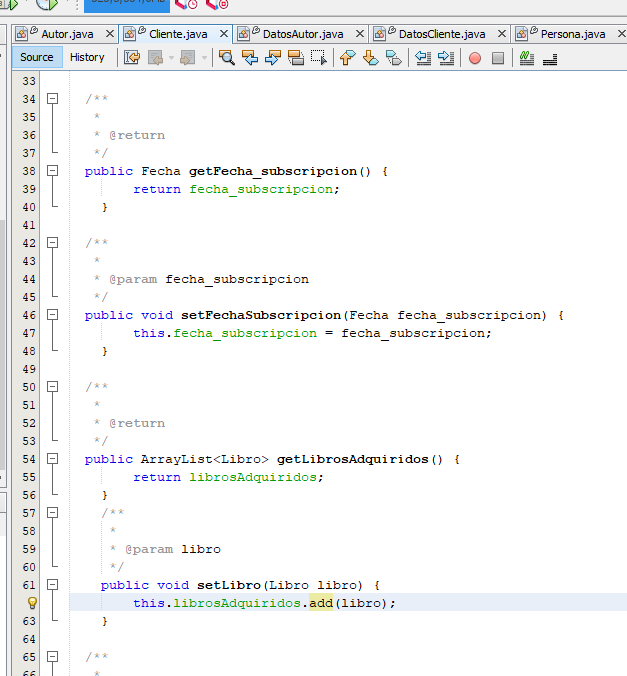
[Principio DIP 8](#_Toc87393019)

Principio SRP

1. **Primer ejemplo**

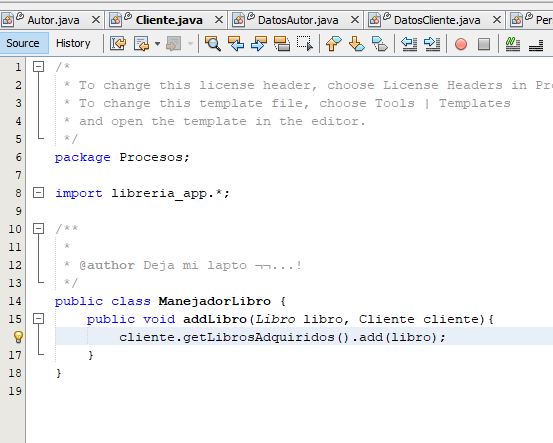
Repositorio: https://github.com/diegoaguilef/libreria\_app

Previo: La clase Cliente tiene una función llamada “setLibro”, que cumple la tarea de añadir más libros, lo cual no concuerda con el nombre de la clase y sus tareas a cumplir.



Justificación:

Se creó una nueva clase llamada ManejadorLibro, que se encarga de la función de agregar un libro a un cliente, de esta forma se libera a Cliente de realizar dicha tarea.



Diagrama

Descripción generada automáticamente

1. Repositorio: Librería\_App

Las clases DatosCliente, DatosLibreria, DatosAutor, DatosPersona y DatosLibro, contienen métodos que cumplen más de una funcionalidad y que utiliza más de un *actor.*

Diagrama

Descripción generada automáticamente

Se propuso una división de las funciones en donde la primera permita buscar y la otra editar la base de datos:

Diagrama

Descripción generada automáticamente

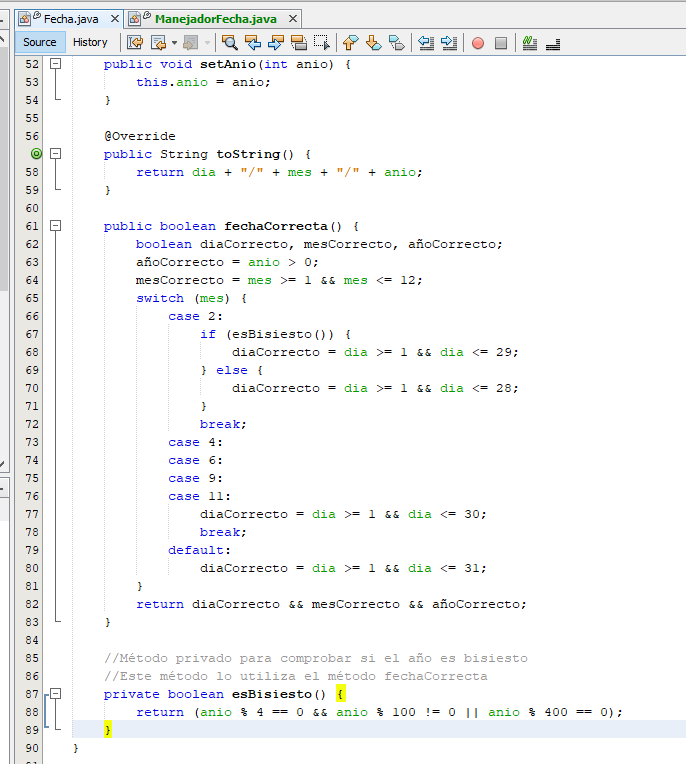
Rediseño del código:

Texto

Descripción generada automáticamente

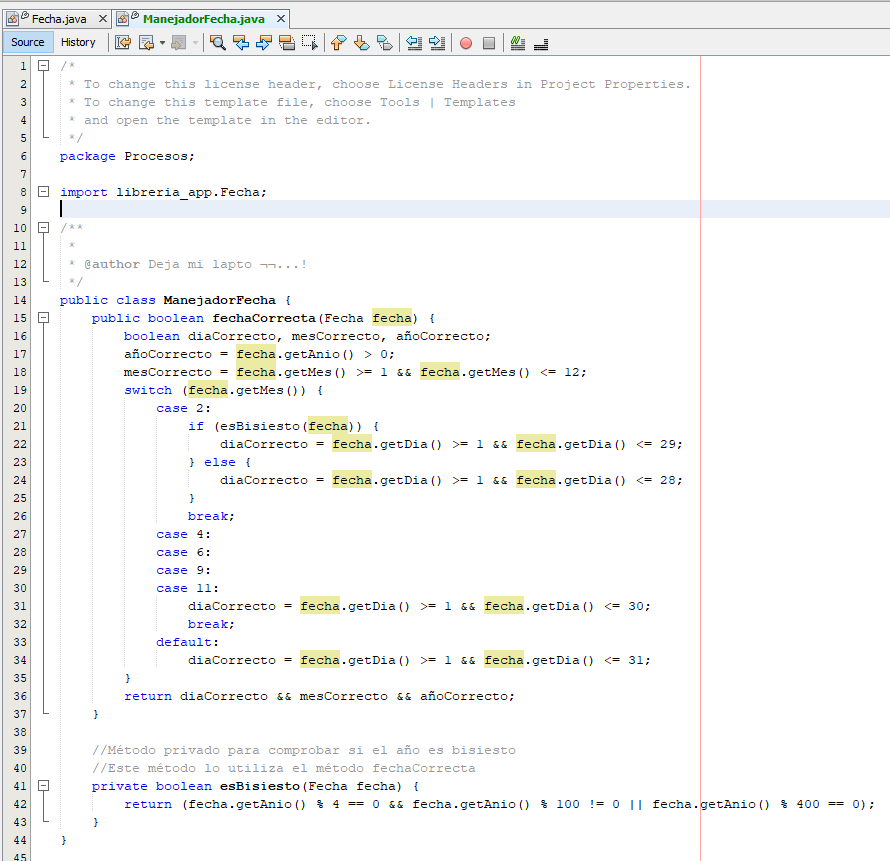
1. Tercer ejemplo SRP

Previo: La clase Fecha tiene que mantener solo los datos correspondientes a la fecha, cualquier calculo de alguna fecha no es un correcto uso de la clase.



Justificación:

La creación de una nueva clase llamada ManejadorFecha, en donde las funciones necesarias para realizar algún cálculo con respecto a la fecha se implementan en la nueva clase, dejando a la clase Fecha libre.



Diagrama

Descripción generada automáticamente

Principio OCP

1. Repositorio: Librería\_App

Como se mencionó en la violación del SRP, las clases de manejo de datos se separaron en clases dependiendo de su funcionalidad; esto también permite la disminución de errores en la extensión del código (adición de nuevos parámetros de búsqueda, etc) y la que su modificación no afecte a clases importantes como DatosCliente, que edita la base de datos.

Texto

Descripción generada automáticamente

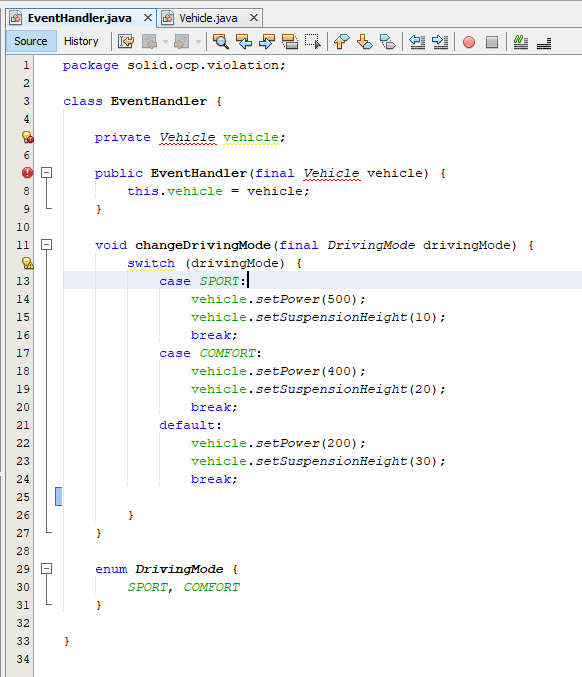
Rediseño de código:

Texto

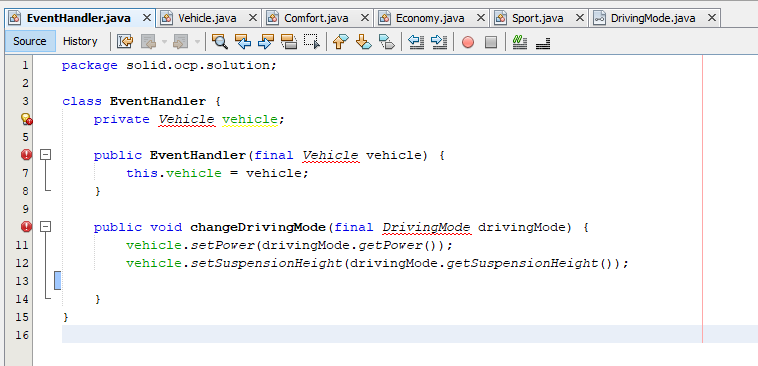
Descripción generada automáticamente

1. Repositorio: <https://github.com/alxgcrz/apuntes-principios-solid>

Previo: Cuando queremos añadir otro modo de conducción tendremos que seguir agregando dentro de ese método las demás opciones, por lo tanto viola OCP.



Solución: Lo que está en el switch, ahora se creó una clase para cada modo de conducción, y se modificó el método. Ahí se puede observar las demás clases creadas.



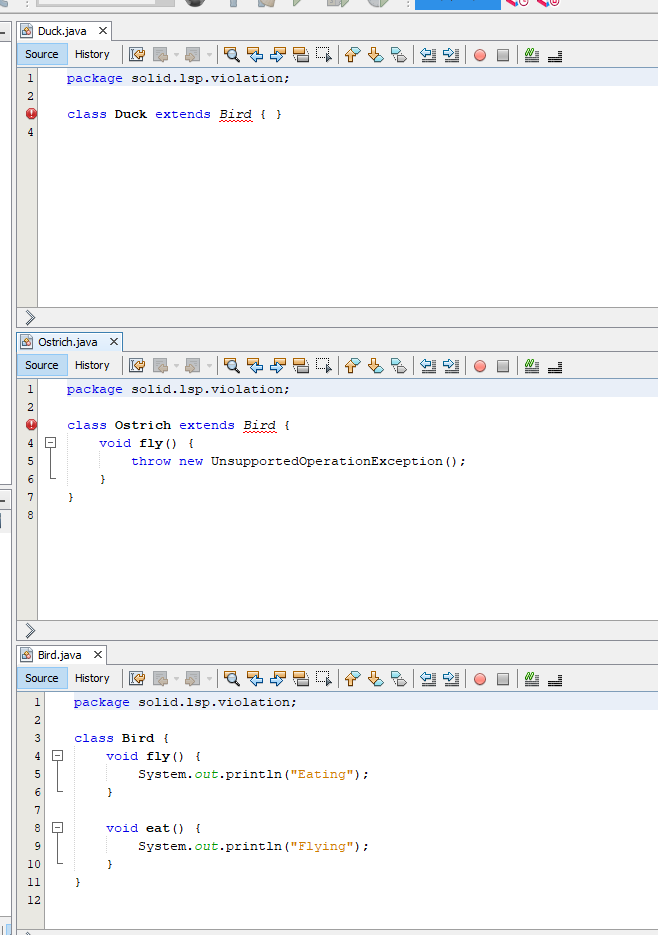
Pantalla de juego de computadora

Descripción generada automáticamente con confianza media

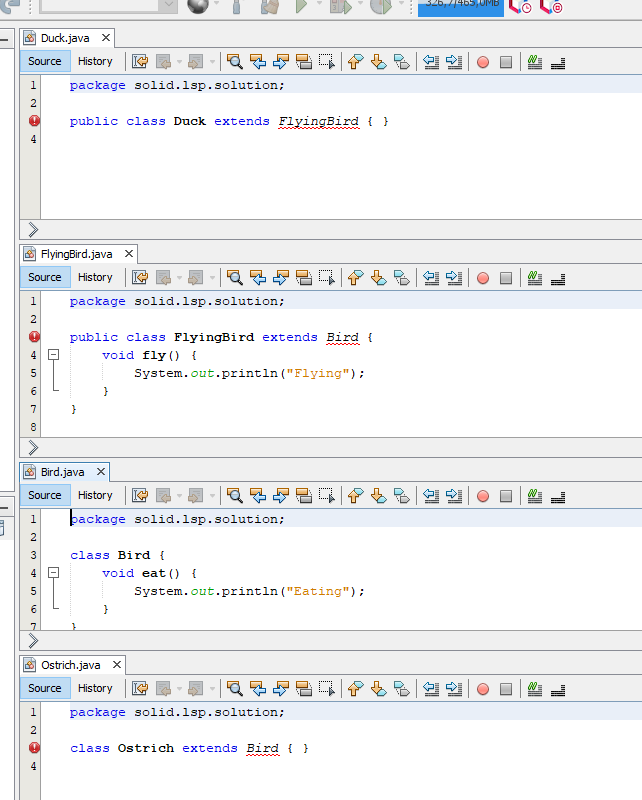
Principio LSP

Repositorio: <https://github.com/alxgcrz/apuntes-principios-solid>

Previo: En este caso las clases hijas no podrían ser reemplazadas por las clases padres, ya que no cumplen correctamente las funciones que tiene el padre.



Corrección: Se crea una clase intermedia llamada FlyingBird para que así esta extienda solamente a Duck. Y así haya una concordancia con clase, se soluciona este problema aplicando Liskov.



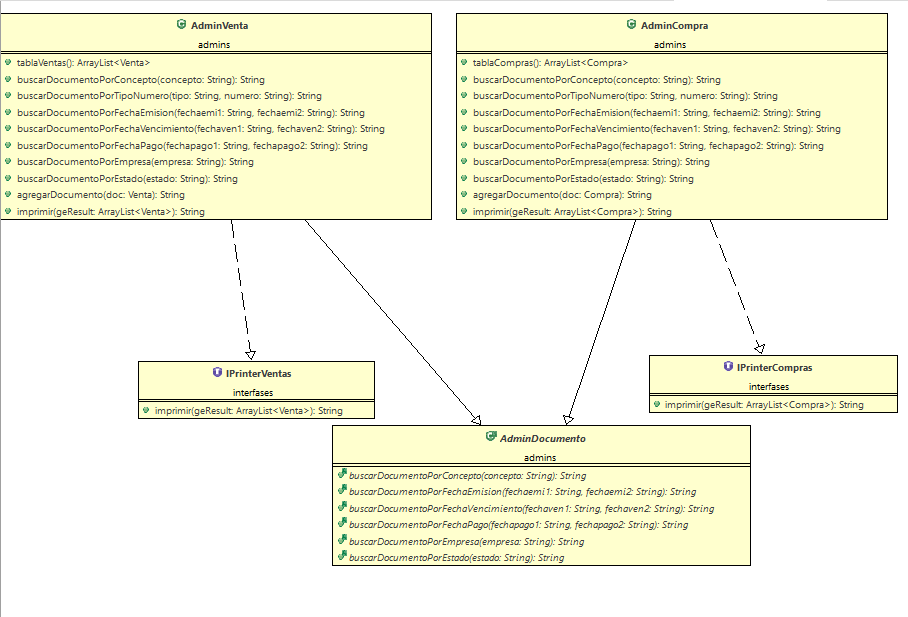
Interfaz de usuario gráfica, Diagrama, Aplicación, Teams

Descripción generada automáticamente

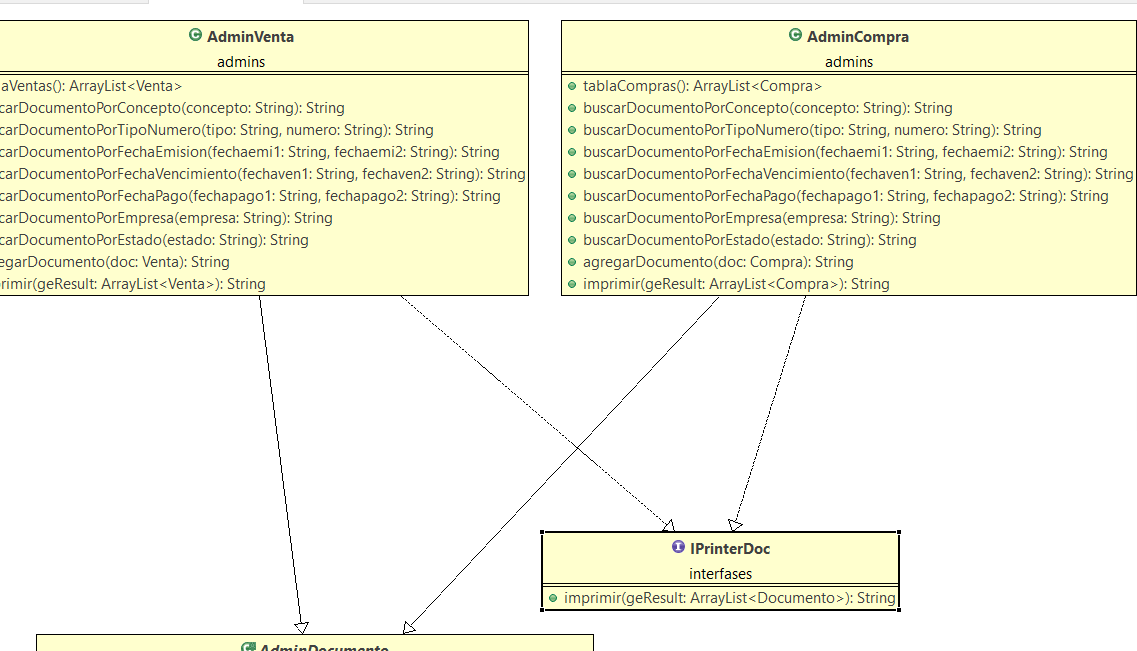
Principio ISP

1)Repositorio: FPPOODJJ

Las clases AdminCompra y AdminVentas tenian interfaces separadas que realiza la mismas acciones(iPrinterCompra e iPrinterVenta) que imprimir las compras y ventas



Al ser hijas de documentos y mostrar la misma informacion en la clase padre, se puede usar una interfaz que se llame iPrinterDoc que imprima como documentos ambas, las compras y las ventas



1. Repositorio: Librería\_App

Diagrama

Descripción generada automáticamente

Rediseño de código:

Texto

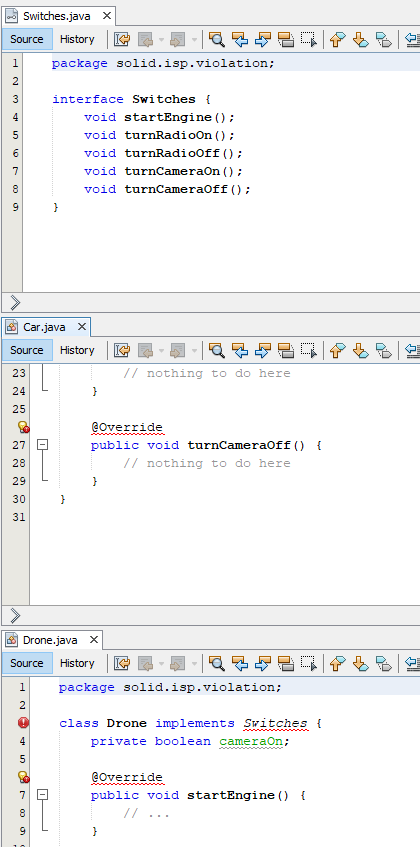
Descripción generada automáticamente

Texto

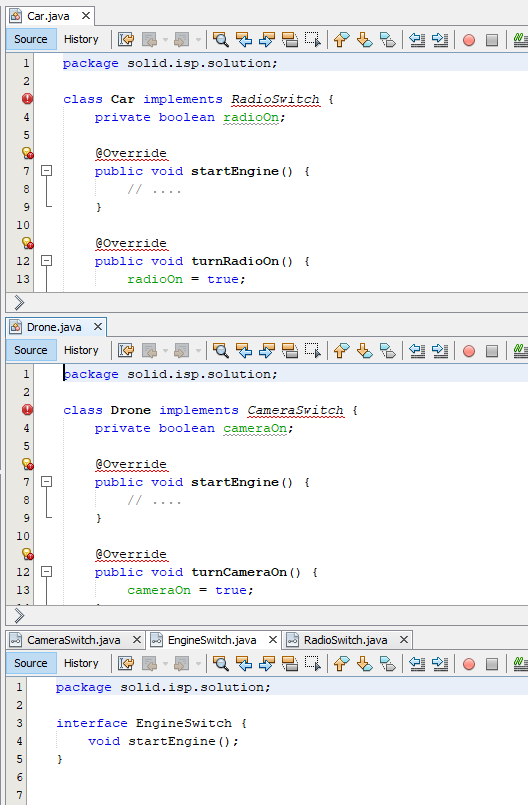
Descripción generada automáticamente

Repositorio: <https://github.com/alxgcrz/apuntes-principios-solid>

Previo: Se observa que las clases Car y Drone no usan todas las funciones de la interface Switches



Corrección: Ahora las clases Car y Drone implementan otras interfaces en donde ya solo implementan las funciones que sí van a usar, en la imagen en la parte se puede observar la creación de las nuevas interfaces.



Pantalla de vídeo juego

Descripción generada automáticamente

Principio DIP

1. Repositorio: Librería\_App

La clase CrudLibro contiene las interfaces visuales y métodos que manejan los eventos consecuentes de la interacción del usuario con el sistema; esta clase contiene dependencias de otras funciones concretas. Esto también elimina la duplicación de código en las clases CrudAutor y LibreriaFrame.

Texto

Descripción generada automáticamente

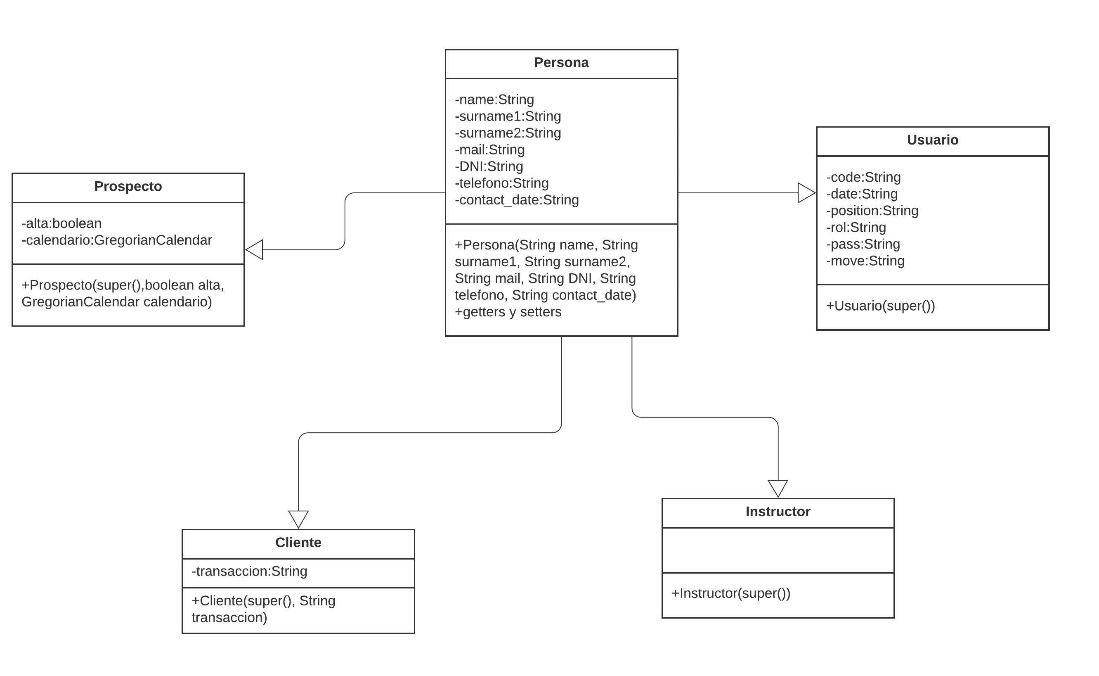
Se propone implementar una clase abstracta, ManejadorFecha, que implemente los métodos validar y parsearFecha:

Texto

Descripción generada automáticamente

2)Repositorio: FPPOODJJ

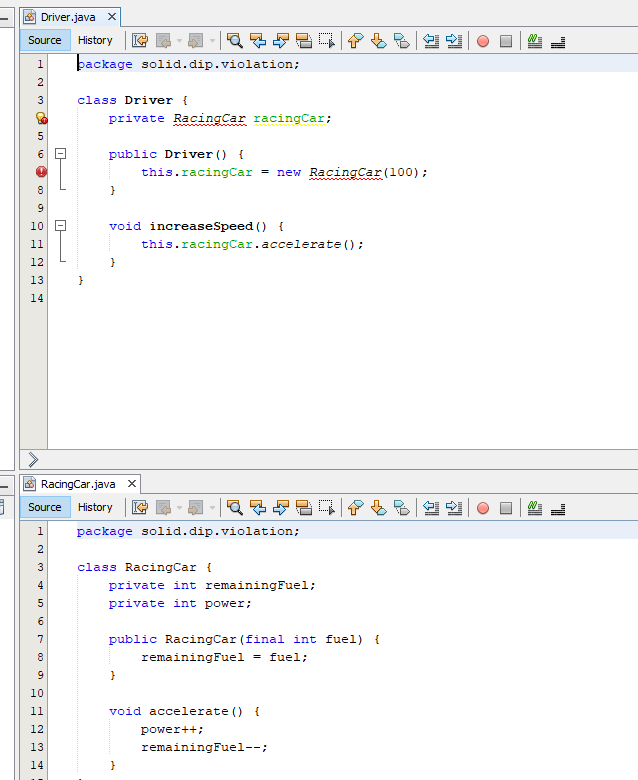
Las clases Usuario, Cliente, Instructor y Prospecto comparten atributos similares por lo cual se realiza a crar una clase padre llamada Persona que hereden de ella y así estaríamos corrigiendo este principio, dado que, estas clases solo se enfocarían en cumplir sus requerimientos sin tener que crear métodos innecesarios.



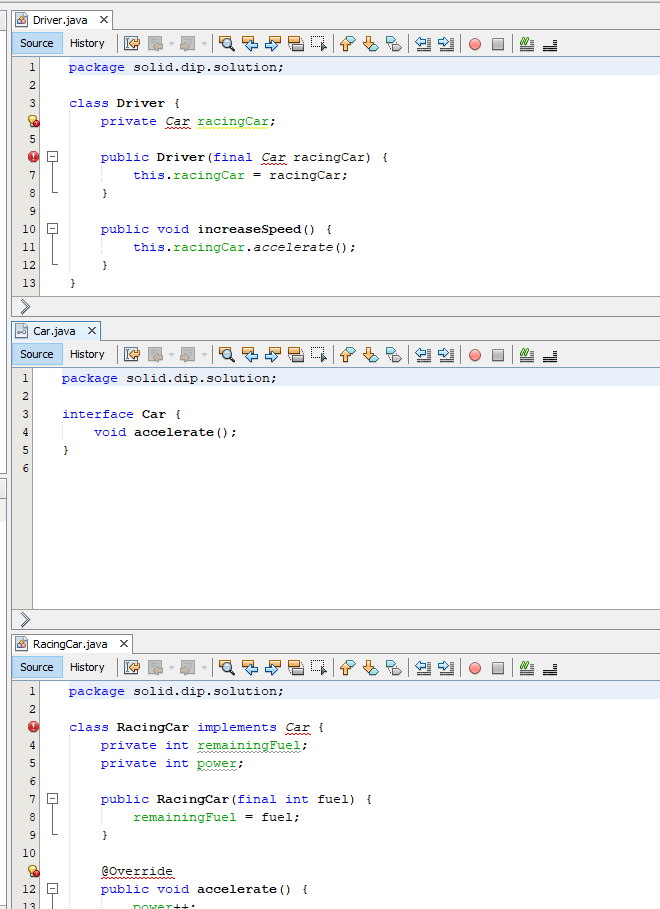
3)

Repositorio: <https://github.com/alxgcrz/apuntes-principios-solid>

Previo: No están bien reducidas las dependencias entre la clase Drives y Rancingcar



Correción: Se creó una interface llamada Car, para que así haya una mejor sinergia entre todo.



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Repositorios:

* Librería\_App: <https://github.com/diegoaguilef/libreria_app>
* <https://github.com/Dannylost/PFPOODJJ>
* <https://github.com/bardes/T4-POO>

Anexos:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media