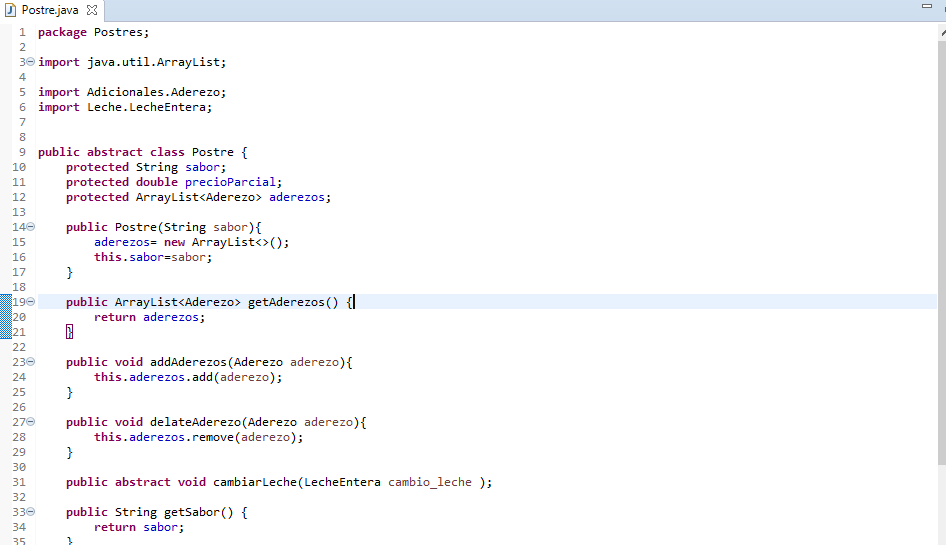
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

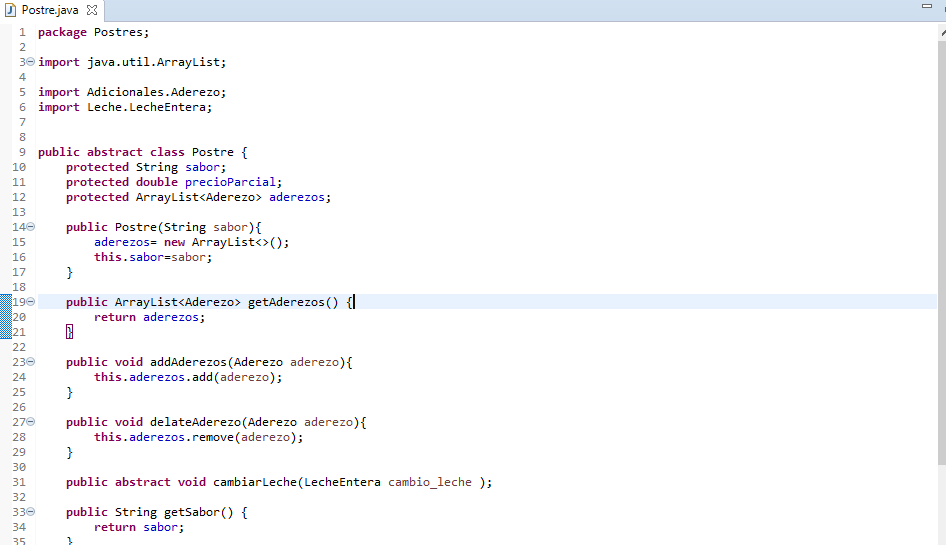
**Taller de Principios SOLID – Diseño de Software**

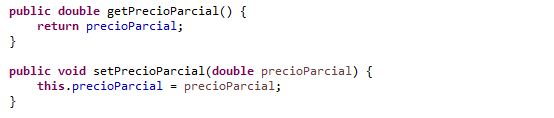
1. **Clases Helado y Pastel. Tienen mucha similitud, se debería crear una clase padre llamada Postre.**

No se está violando ningún principio SOLID ya que solamente se está repitiendo el código en clases diferentes. El problema de la repetición de código se resolvió agregando la clase padre Postre.



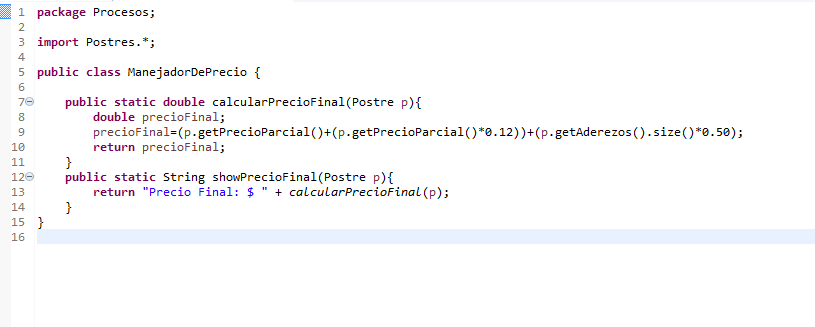
1. **Clases Procesos.OperacionesAderezo y Postre. ¿Es necesaria la clase OperacionesAderezo?. Se puede incluir dentro de postre un método para agregar un aderezo y para quitar un aderezo.**

Se está violando el principio de Responsabilidad única ya que se están usando métodos específicos propios de la clase postre en otras clases. No es necesaria la clase OperacionesAderezo por lo que se procedió a eliminar la clase.



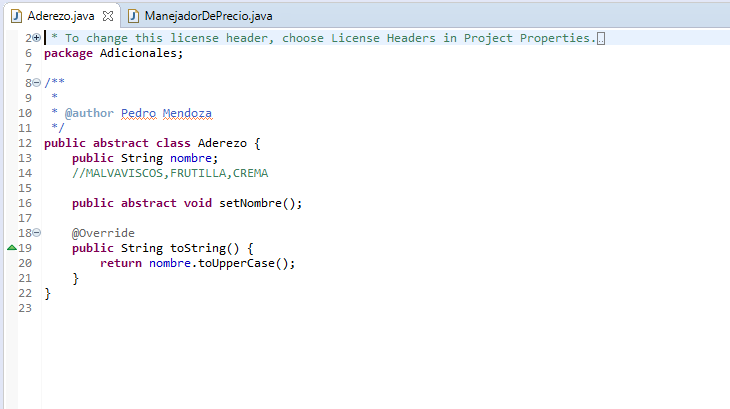
1. **Métodos calcularPrecioFinal() y showPrecioFinal() están muy relacionados, deben estar en otra clase por si cambia la fórmula de cálculo. La clase nueva debe llamarse Procesos.ManejadorDePrecio.**

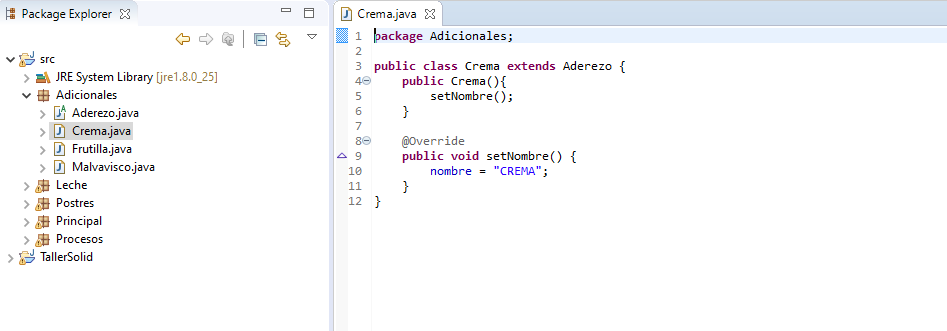
SRP (Single Responsability Principle) se viola este principio porque la responsabilidad de la clase postre y helado no debe incluir los métodos que calcularPrecioFinal() y showPrecioFinal(), se lo arregla creando la clase manejadorDePrecio.

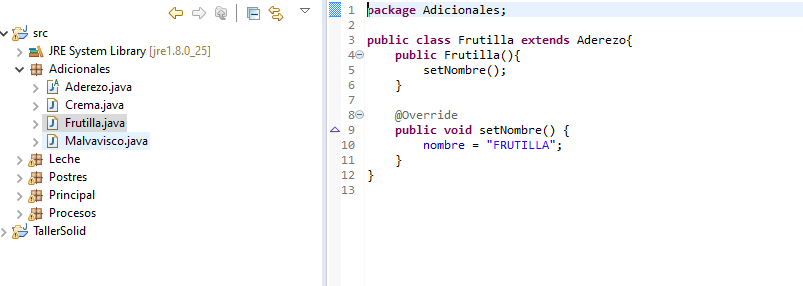


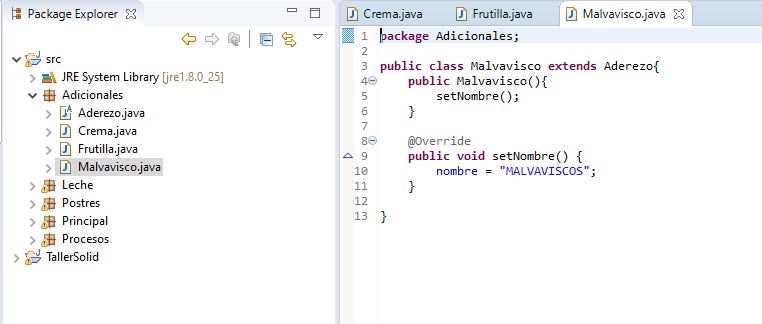
1. **Enum Adicionales.Aderezo es muy estático, debe convertirse en clase abstract con un atributo nombre y un método abstracto setNombre para que cada tipo de aderezo sea una subclase de Aderezo e implemente dicho método. También, sobrescriba el método toString() en la clase Aderezo, para que devuelva el nombre del aderezo en mayúsculas.**

Dependency Inversion Principle, al depender de abstracciones las clases aprovechan la interfaz de las abstracciones, convertimos la clase Aderezo en abstracta y convertimos cada tipo de aderezo sea una subclase de aderezonono









1. **Paquete Leche y la clase Procesos.ManejadorDeLeche. En el main descomente las instrucciones para realizar el cambio del tipo de leche utilizada en cada postre, luego analice como solucionar el error generado en la clase Leche.LecheDeslactosada.**

Se solucionó el error en la clase LecheDeslactosada quitando la excepción y retornando un mensaje para saber que el proceso no se completó, también se agregó un condicional en la clase pastel para validar esa condición cumpliendo el principio OCP, ya que si existe otro tipo de leche solo debe de cumplir con la clase LecheEntera o si existe otro postre solo debe cumplir con la clase Postre

