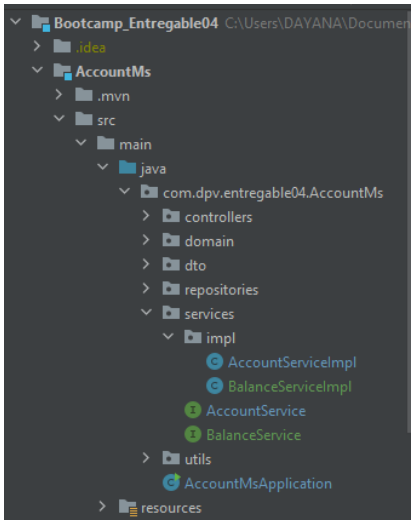
PRINCIPIO SOLID

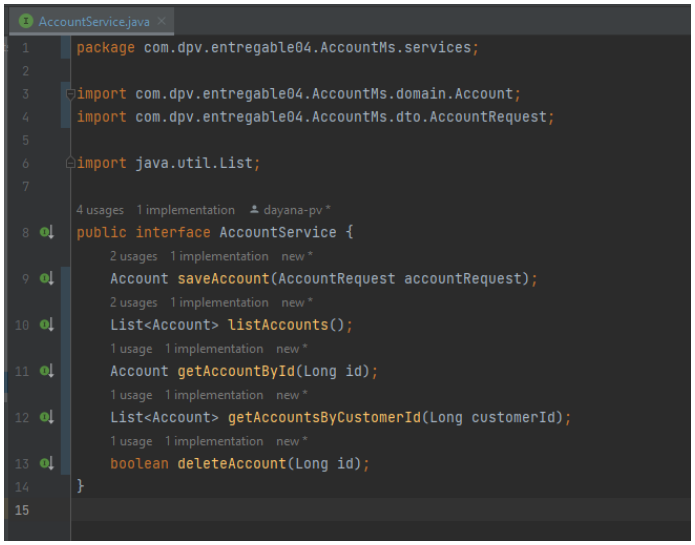
Microservicios:

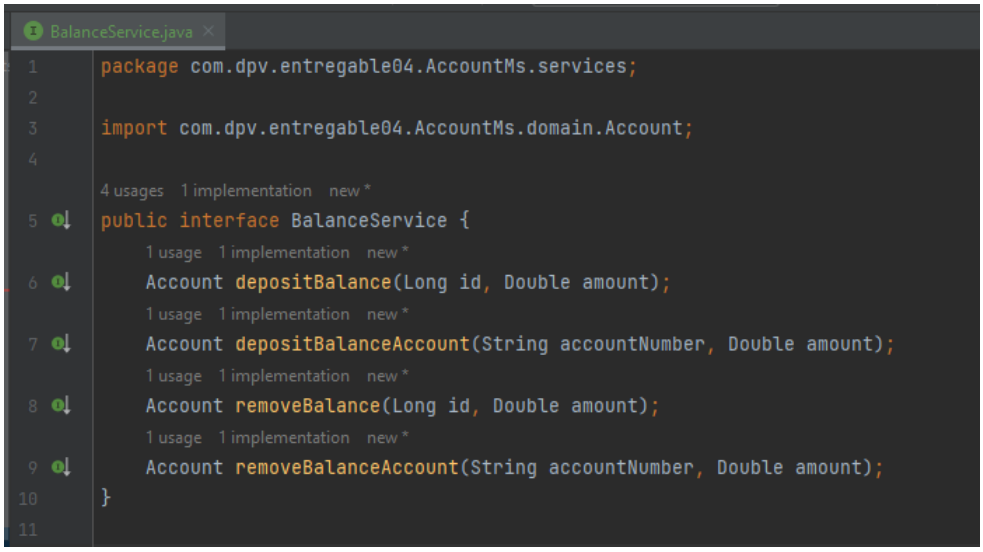
AccountMs

Principio de Responsabilidad Única

Se utiliza el Principio de Responsabilidad Única en los package de Service y en impl, creándose dos interfaces (AccountService y BalanceService) y dos clases (AccountServiceImpl y BalanceServiceImpl), en donde la clase de AccountService tiene las funcionalidades de registrar, listar y eliminar cuentas, mientras que la clase BalanceService presenta las funcionalidades de depositar y remover el saldo de las cuentas.

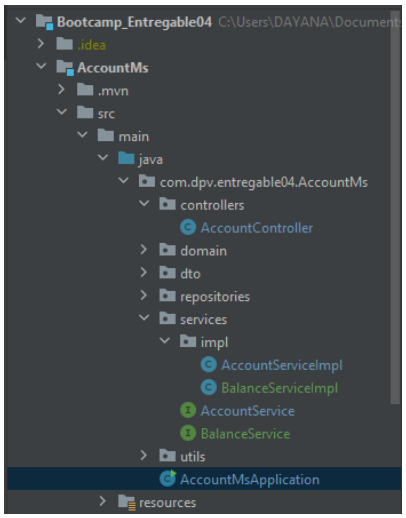


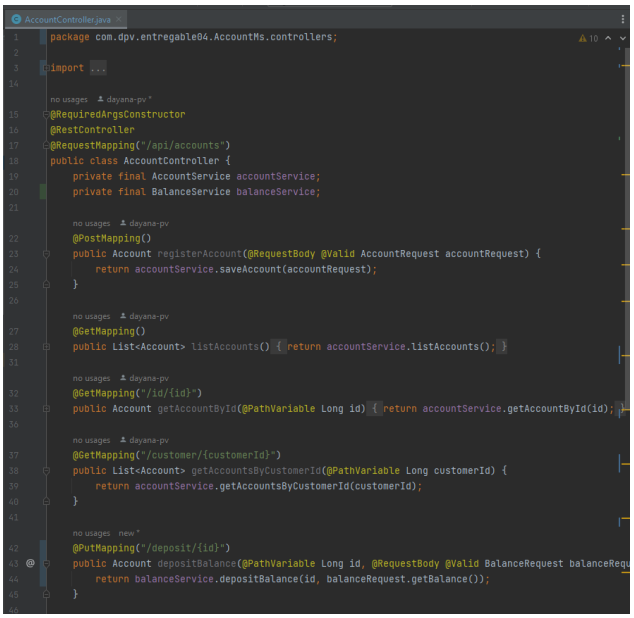



Principio de Inversión de Dependencias

Se utiliza el Principio de Inversión de Dependencias en el Controller AccountController, debido a que el controlador no tiene conocimiento de la implementación concreta de AccountServiceImpl, ya que se conecta con la interfaz AccountService.

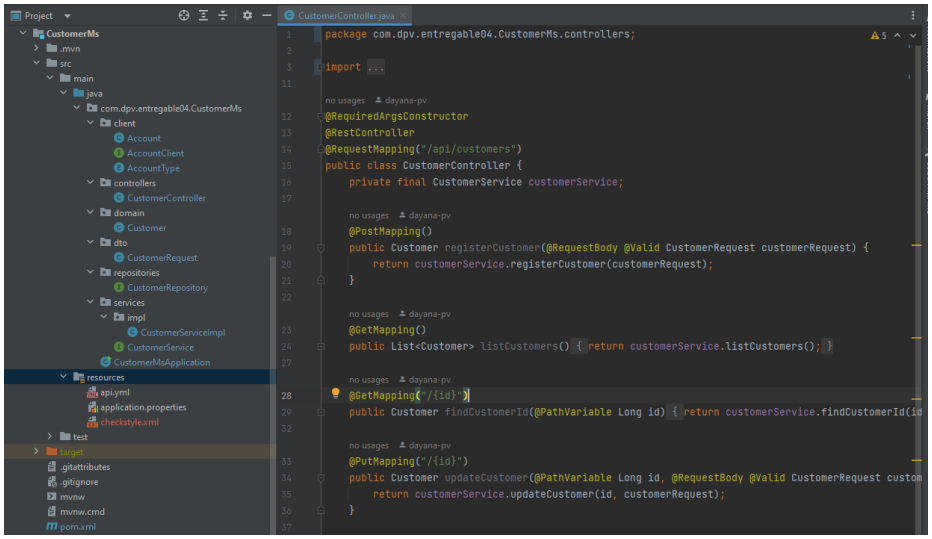



CustomerMS

Principio de Inversión de Dependencias

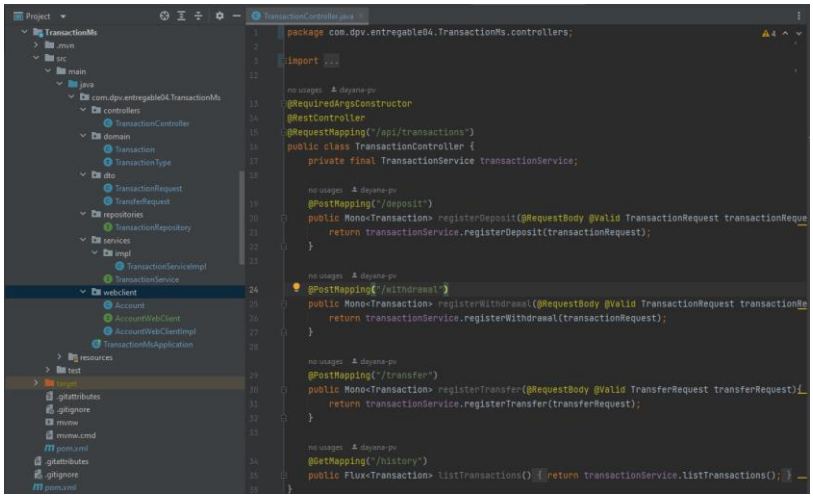
Se utiliza el Principio de Inversión de Dependencias en el Controller CustomerController, debido a que el controlador no tiene conocimiento de la implementación concreta de  CustomerServiceImpl, ya que se conecta con la interfaz CustomerService.



TransactionMS

Principio de Inversión de Dependencias

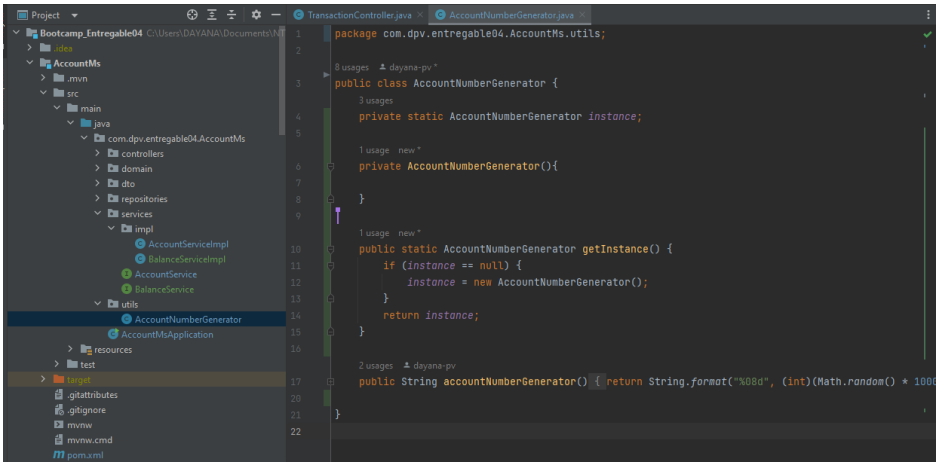
Se utiliza el Principio de Inversión de Dependencias en el Controller TransactionController, debido a que el controlador no tiene conocimiento de la implementación concreta de  TransactionServiceImpl, ya que se conecta con la interfaz TransactionService.




PATRONES DE DISEÑO

SINGLETON

Se utiliza el Patrón de Diseño Singleton en las clases AccountNumberGenerator y AccountServiceImpl, garantizando que AccountServiceImpl esté funcionando correctamente para obtener la instancia única de AccountNumberGenerator.


Project
Bootcamp_Entregable04 C:\Users\DAYANA\Documents\N
.idea
AccountMs
.mvn
src
main
java
com.dpv.entregable04.AccountMs
controllers
domain
dto
repositories
services
impl
AccountServiceImpl
BalanceServiceImpl
AccountService
BalanceService
utils
AccountNumberGenerator
AccountMsApplication
resources
test
target
.gitattributes
.gitignore
mvnw
mvnw.cmd
pom.xml
TransactionController.java
AccountNumberGenerator.java
package com.dpv.entregable04.AccountMs.utils;
8 usages dayana-pv *
public class AccountNumberGenerator {
3 usages
private static AccountNumberGenerator instance;
1 usage new *
private AccountNumberGenerator(){
}
1 usage new *
public static AccountNumberGenerator getInstance() {
if (instance == null) {
instance = new AccountNumberGenerator();
}
return instance;
}
2 usages dayana-pv
public String accountNumberGenerator() { return String.format("%08d", (int)(Math.random() * 100
}