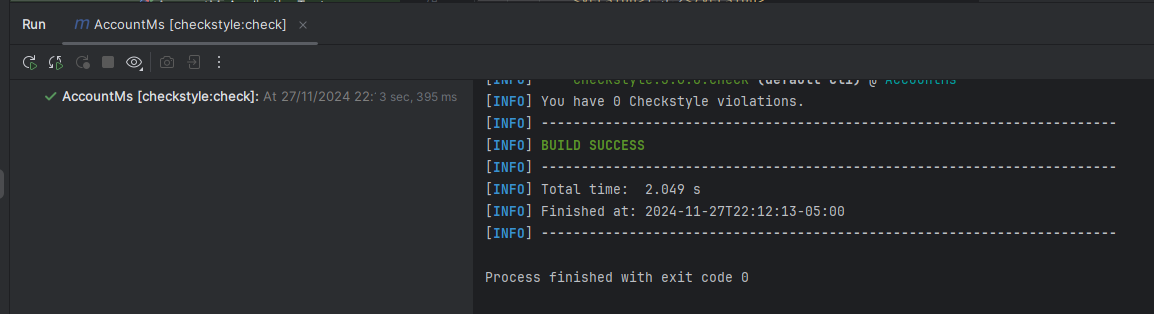
**Entregable 4 -Java Microservicios**

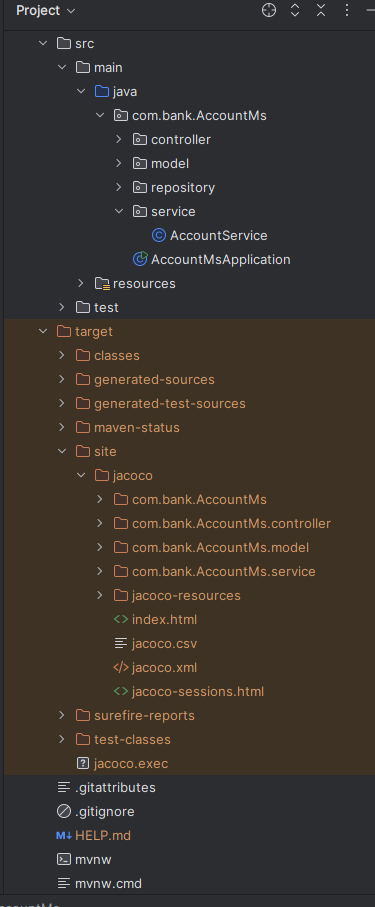
**Microservicio AccountMS**

**Requerimientos del entregable:**

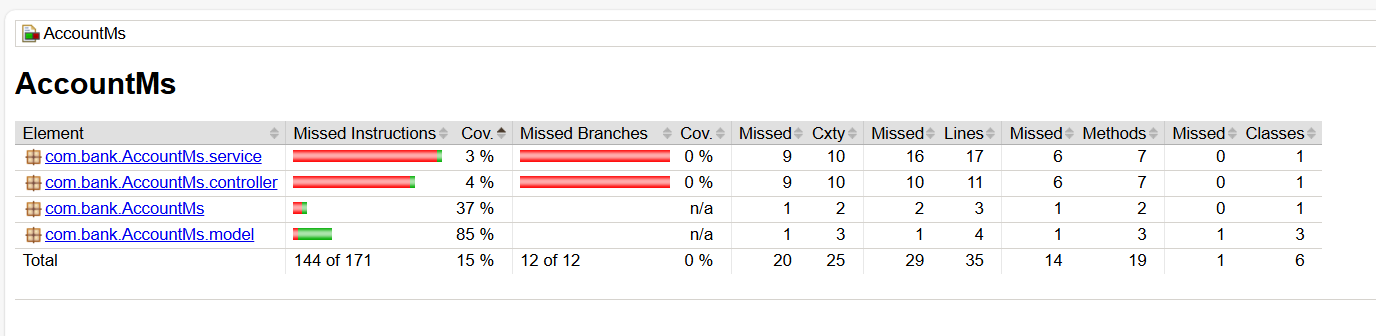
**Reglas del archivo checkstyle sin violaciones**



Informe de cobertura con Jacoco

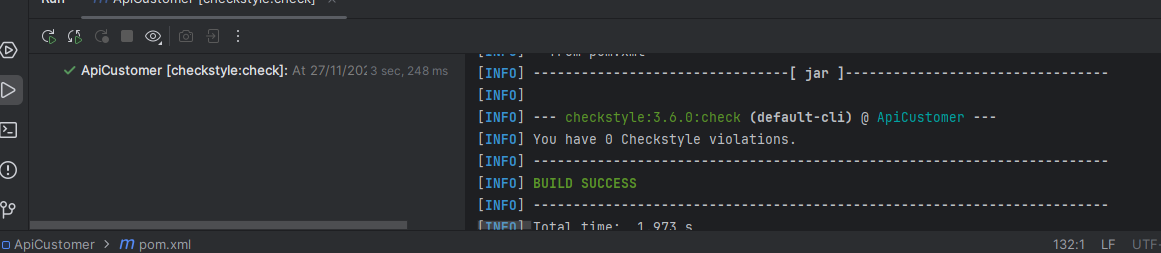


**Informe gráfico de cobertura de código**

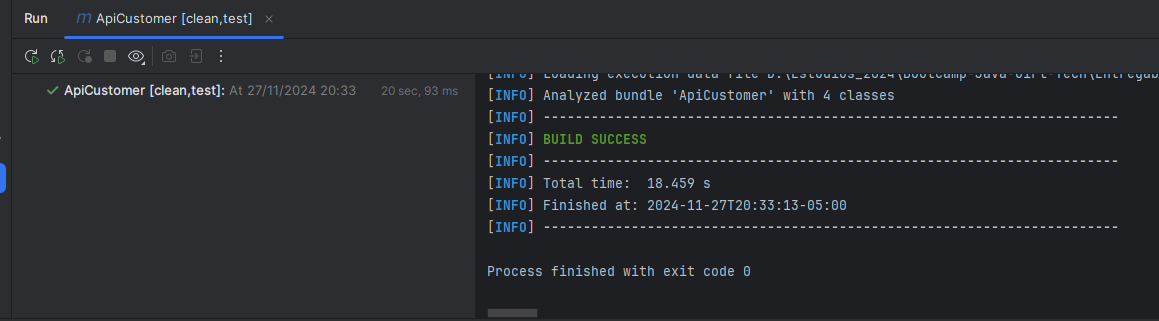


**Microservicio ApiCustomer**

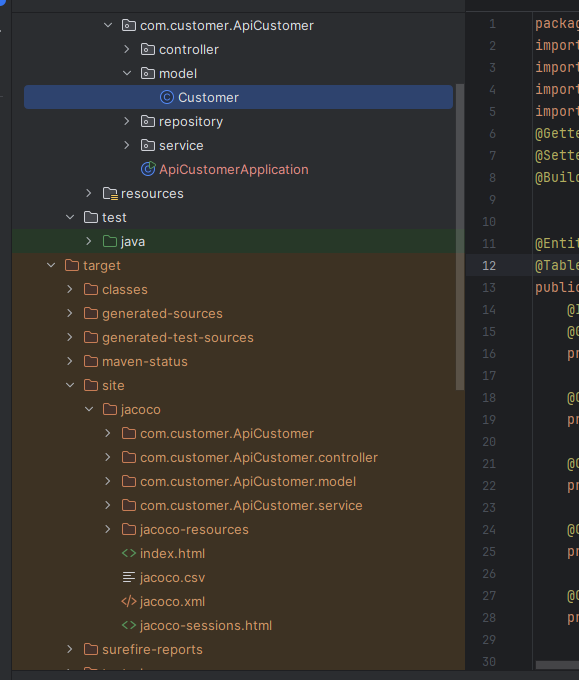
**Reglas del archivo checkstyle sin violaciones**



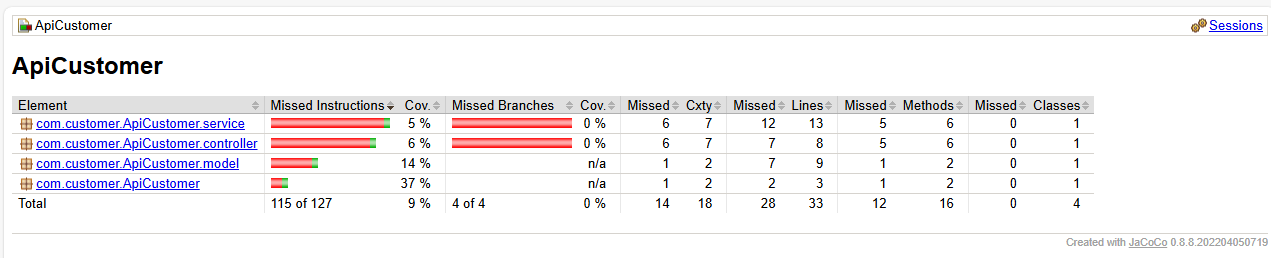
**Ejecución de Jacoco sin errores**



**Generación de carpeta site de Jacoco**



**Informe gráfico de cobertura de Código**

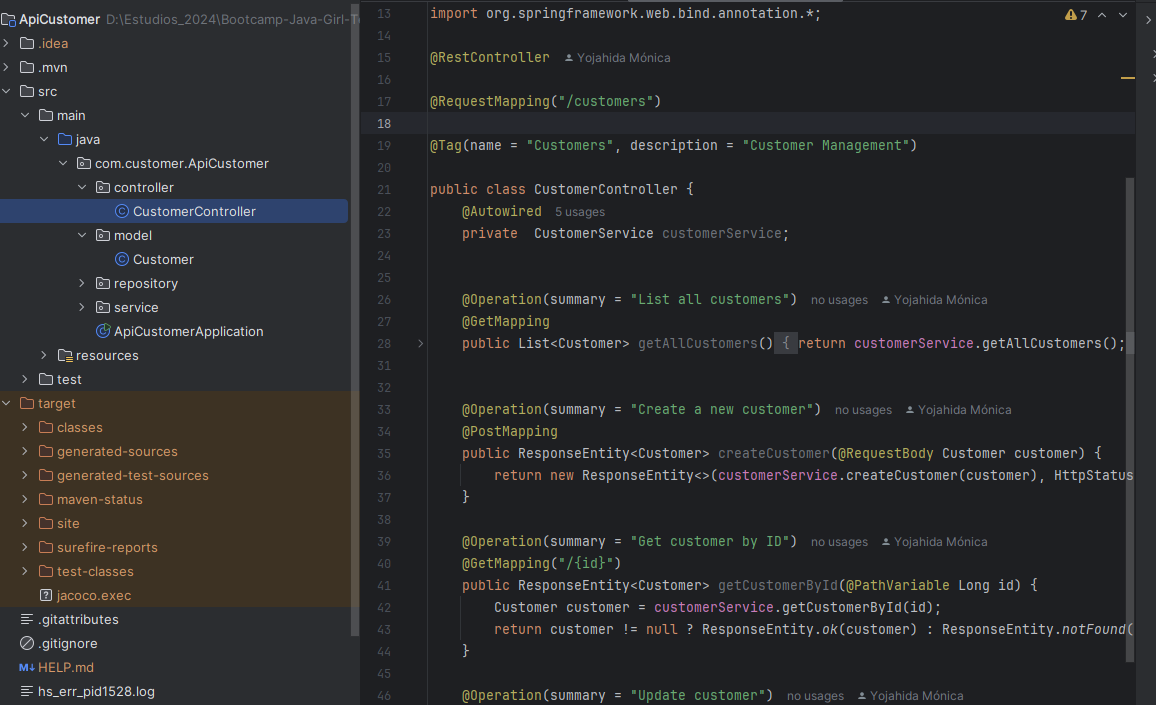


**Mapeo principios S.O.L.I.D**

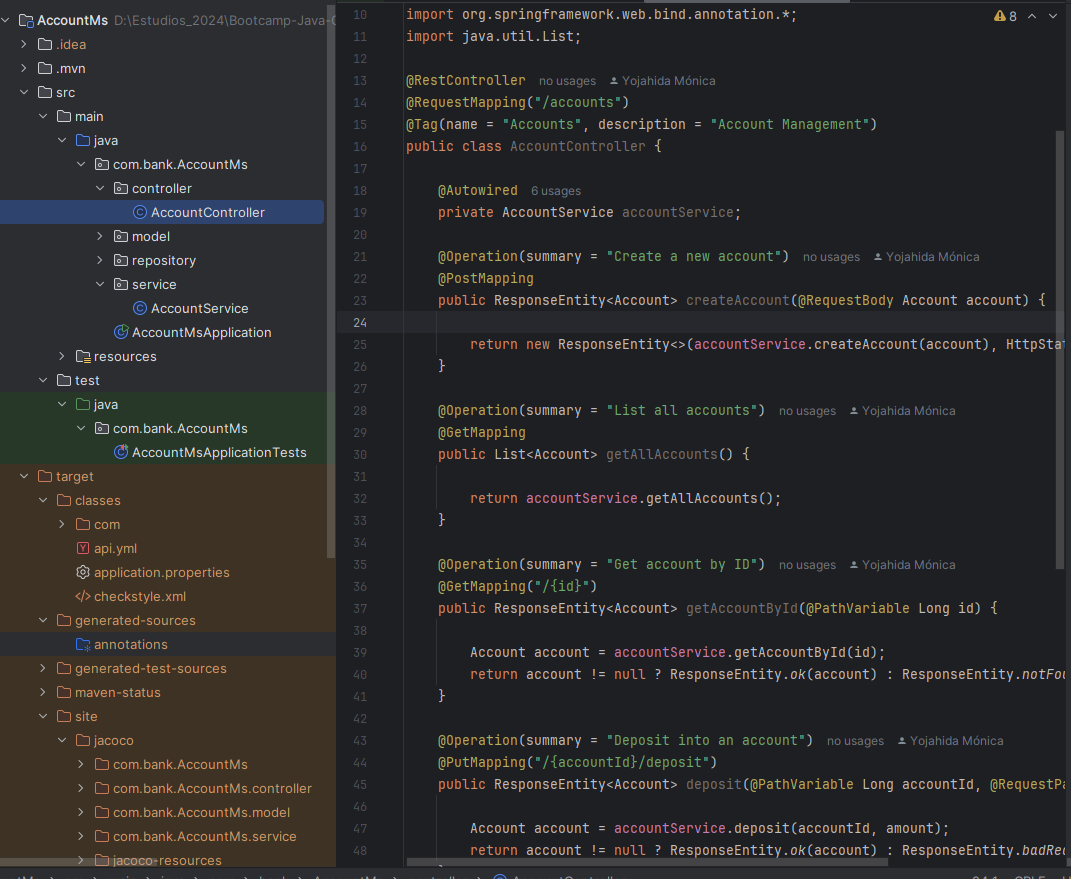
**Principio de responsabilidad única (SRP)**

Las clases CustomerController y Account Controller solo gestionan las solicicitudes HTTP, aquí no se realiza la lógica del del negocio.

*Clase CustomerController*

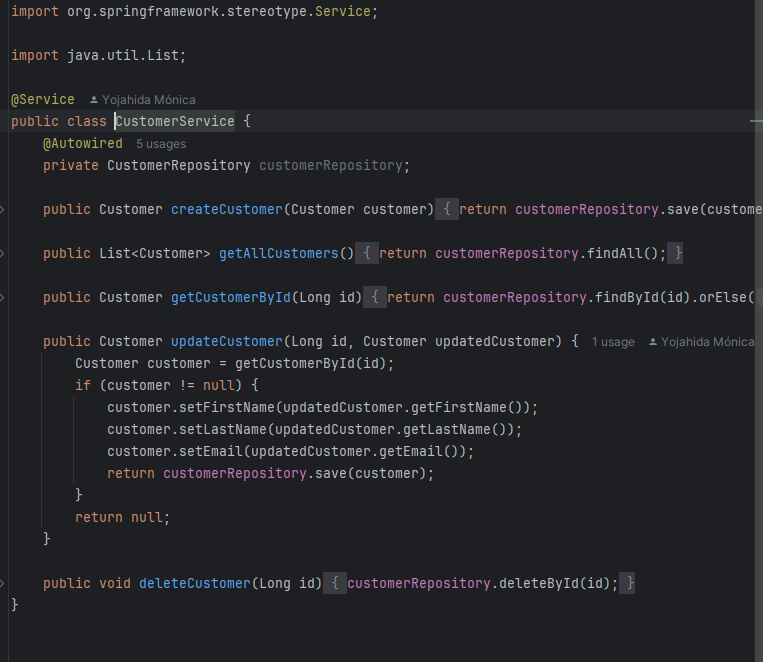


*Clase AccountController*

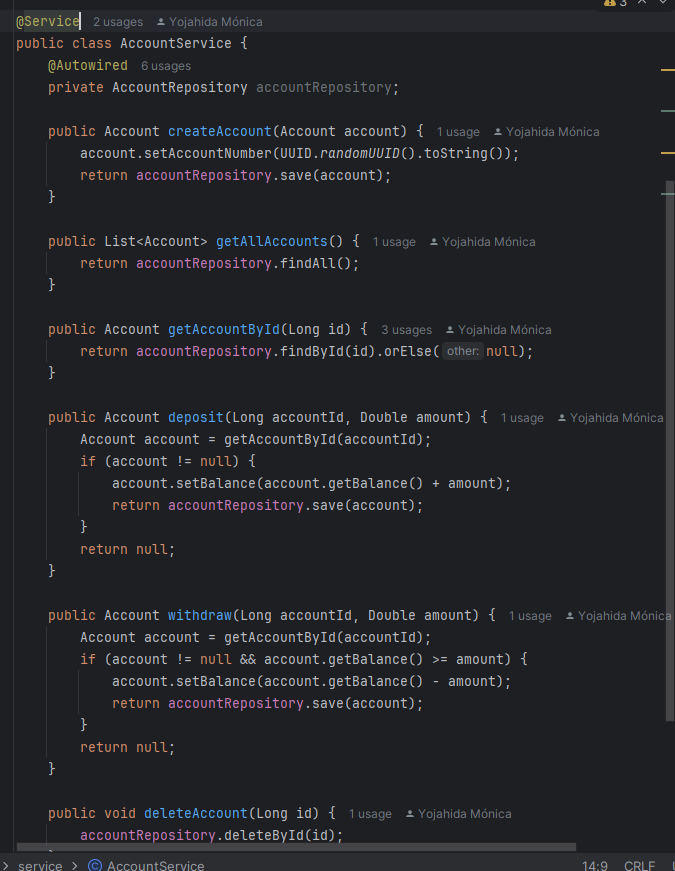


Las clases CustomerService y AccountService contienen la lógica de negocio, no interactúan directamente con la base de datos.

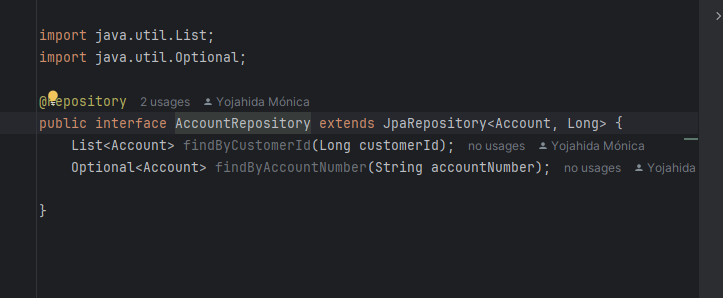
*Clase CustomerService*

**

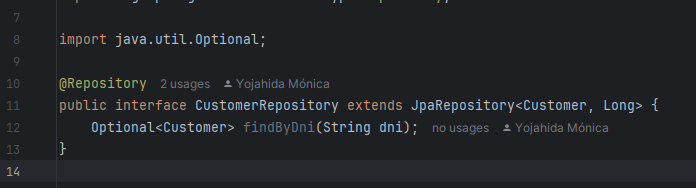
*Clase AccountService*

**

Las interfaces CustomerRepository y AccountRepository están dedicadas exclusivamente a interactuar con la base de datos y no manejan ni las solicitudes HTTP ni la lógica del negocio.

*interface AccountRepository*

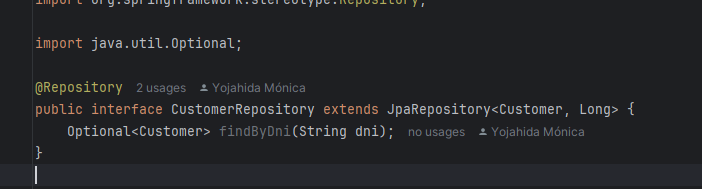
*Interface CustomerRepository*



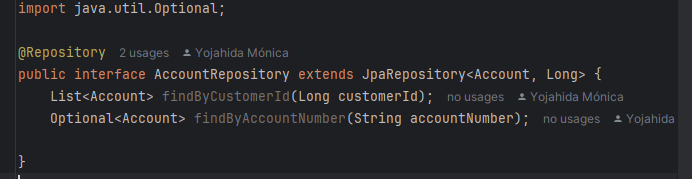
**Principio de abierto/cerrado(OCP)**

Las clases CustomerRepository y AccountRepository, son interfaces que extienden de JpaRepository,lo que permite agregar más funcionalidades, es decir nuevas consultas sin modificar las clases existentes.

*Interface CustomerRepository*

****

*interface AccountRepository*



**Principio de sustitución de Liskov(LSP)**

En este principio en que se establece que los objetos de una clase derivada deben poder reemplazar a los objetos de su clase base sin alterar el comportamiento del programa. Las clases clase que extienden de JpaRespository aseguran que lo métodos CRUD pueden ser sustituidos por implementaciones personalizadas en caso necesario sin romper el comportamiento esperado.

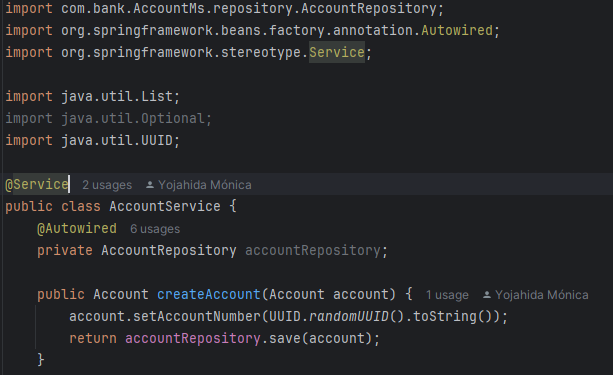
**Principio de segregación de interfaces(ISP)**

JpaRepository es una interfaz que proporciona los métodos necesarios para CRUD, sin agregar más métodos que obliguen a las clases a implementar funcionalidades innecesarias**.**

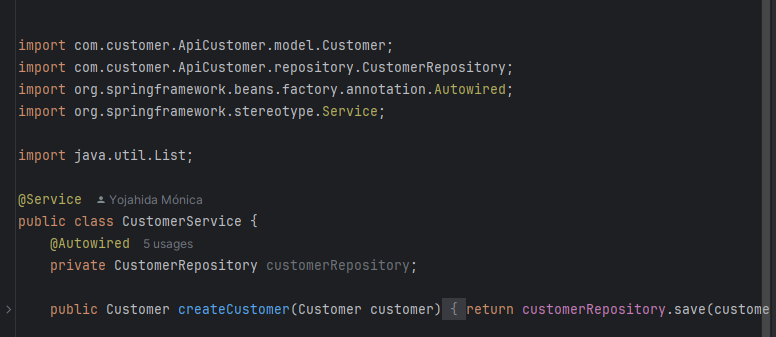
**Principio de inversión de dependencias(DIP)**

La aplicación se evidencia en las clases CustomerService y AccountService que dependen de las interfaces CustomerRepository y AccountRepository.

*Clase AccountService depende de la interfaz AccountRepository*

****

*Clase CostumerService depende de la interfaz CustomertRepository*

**