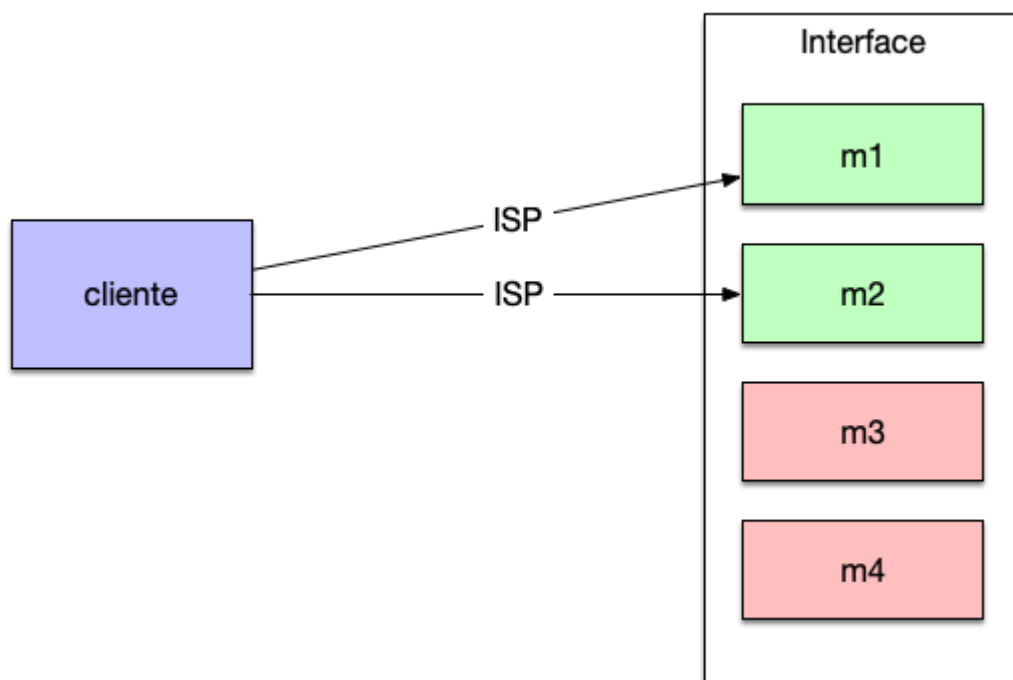
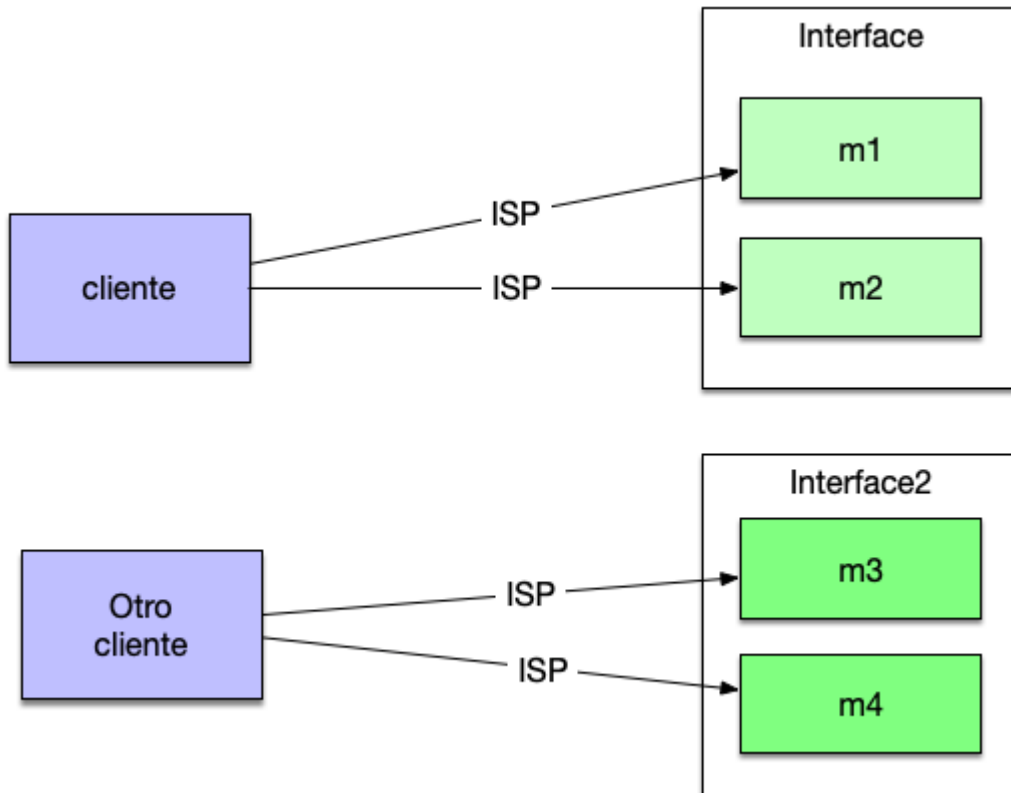


Acabamos de ver como usar Java equals y hashCode

El Interface Segregation Principle (ISP) o principio de segregación de interfaces es uno de los principios que esta bajo el grupo de los principios Solidos (SOLID) . Concretamente es el que hace la I. ¿Para qué sirve este principio y cómo se usa?. Este principio define que un cliente no puede depender de métodos de un interface que no usa. Es decir la siguiente figura no cumpliría el principio ISP.



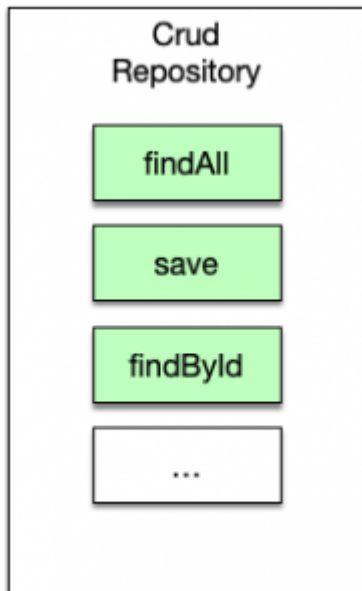
Hay varios métodos que el cliente del interface no usa. Para cumplir con el principio ISP (Interface Segregation Principle) debemos segregar el interface y dividirlo en dos . El primero dispondrá de los métodos m1 y m2 y el segundo de m3 y m4.



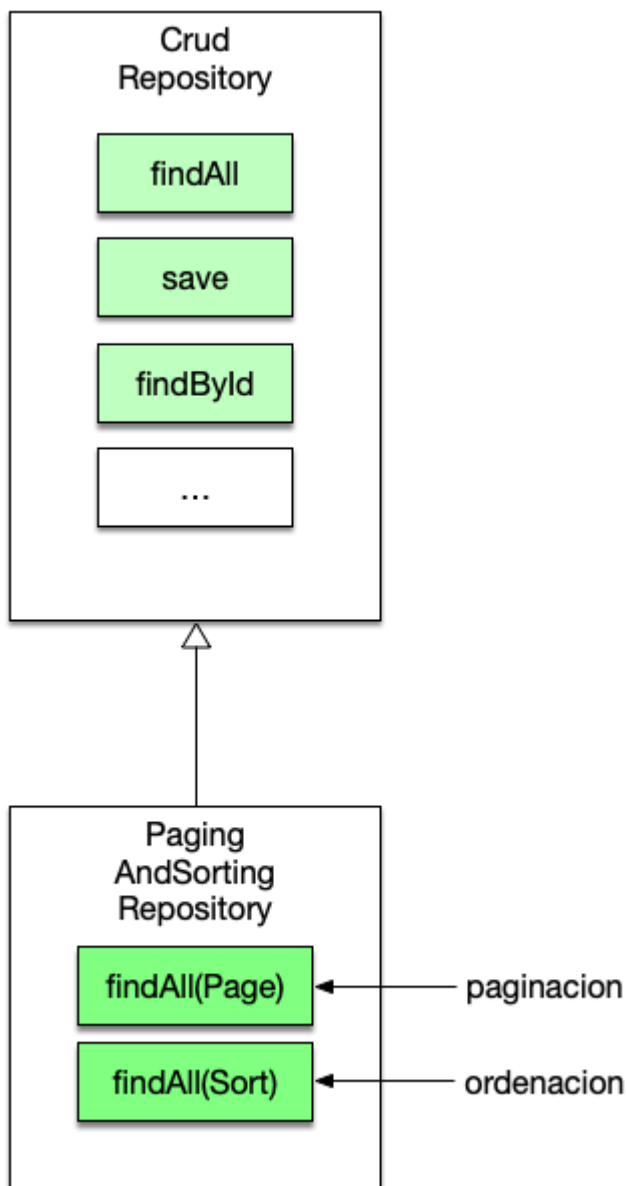
De esta manera las responsabilidades están mucho más organizadas y cada cliente accede a lo que necesita.

Interface Segregation Principle y Spring Data

Un ejemplo clásico del uso de este principio lo podemos ver en Spring Data . Spring Data es el framework de Spring que se encarga de persistir información de una forma lo más transparente al programador posible. Este framework dispone de un interface fundamental que se denomina CrudRepository.



Este interface se encarga de los métodos de persistencia más habituales. Prácticamente todos los clientes necesitan este tipo de operaciones a la hora de usar el concepto de repositorio. Buscar todos los elementos , salvarlos , o buscarlos por Id. Sin embargo este no es el único interface soportado por Spring Data . Existe un interface hijo que se denomina PagingAndSortingRepository

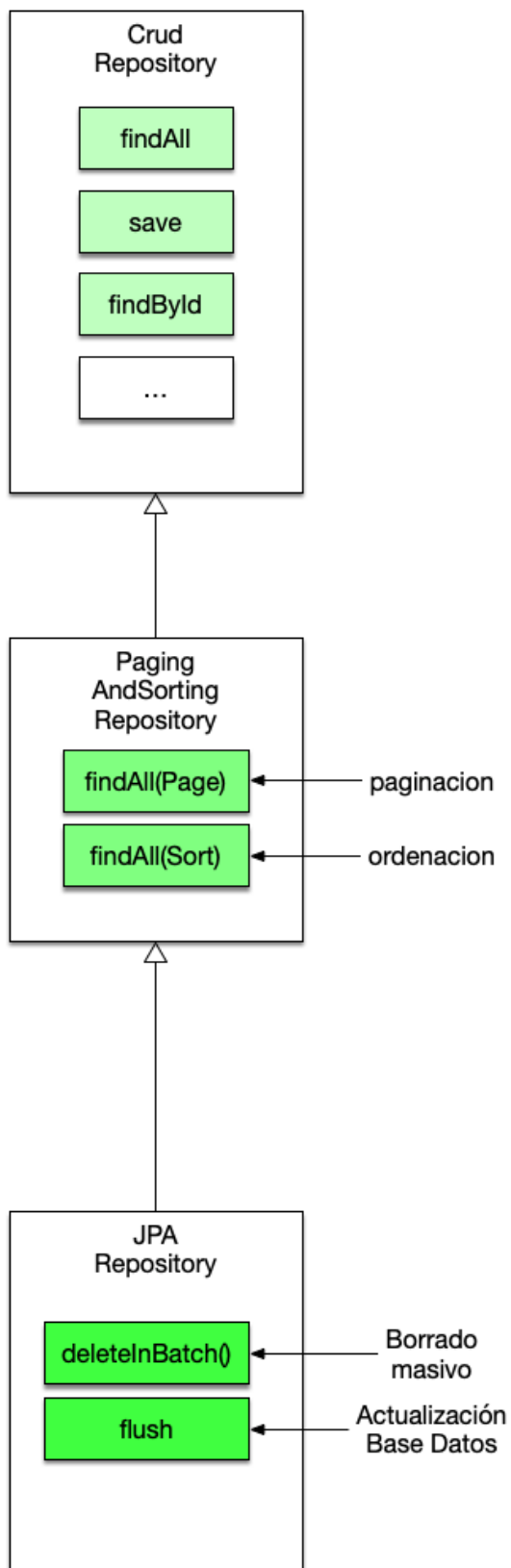


Este interface aglutina una funcionalidad que no todos los clientes necesitan de un repositorio. Es decir una funcionalidad de paginación y ordenación que solo “algunos” van a necesitar. Por lo tanto los diseñadores de Spring lo han separado.

Spring Data y JpaRepository

Para complementar otro de los interfaces más habituales de Spring Data es el `JpaRepository`

que añade métodos fuertemente ligados a JPA y que extiende el interface de `PagingAndSortingRepository`.



En este caso son métodos que solo aquellos clientes que usan JPA utilizan. Spring Data se encarga de dividir las responsabilidades a nivel de interfaces usando ISP (Interface Segregation Principle) ya que cada tipo de cliente necesitará unas capacidades.

Otros artículos relacionados:

1. [Introducción a Spring Data y JPA](#)
2. [@RepositoryRestResource y Spring Framework](#)
3. [Spring Boot JPA y su configuración](#)
4. [Spring Data](#)