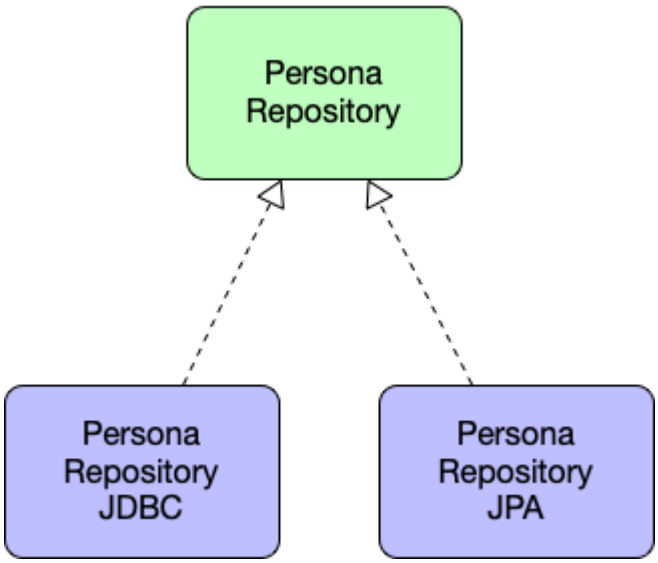
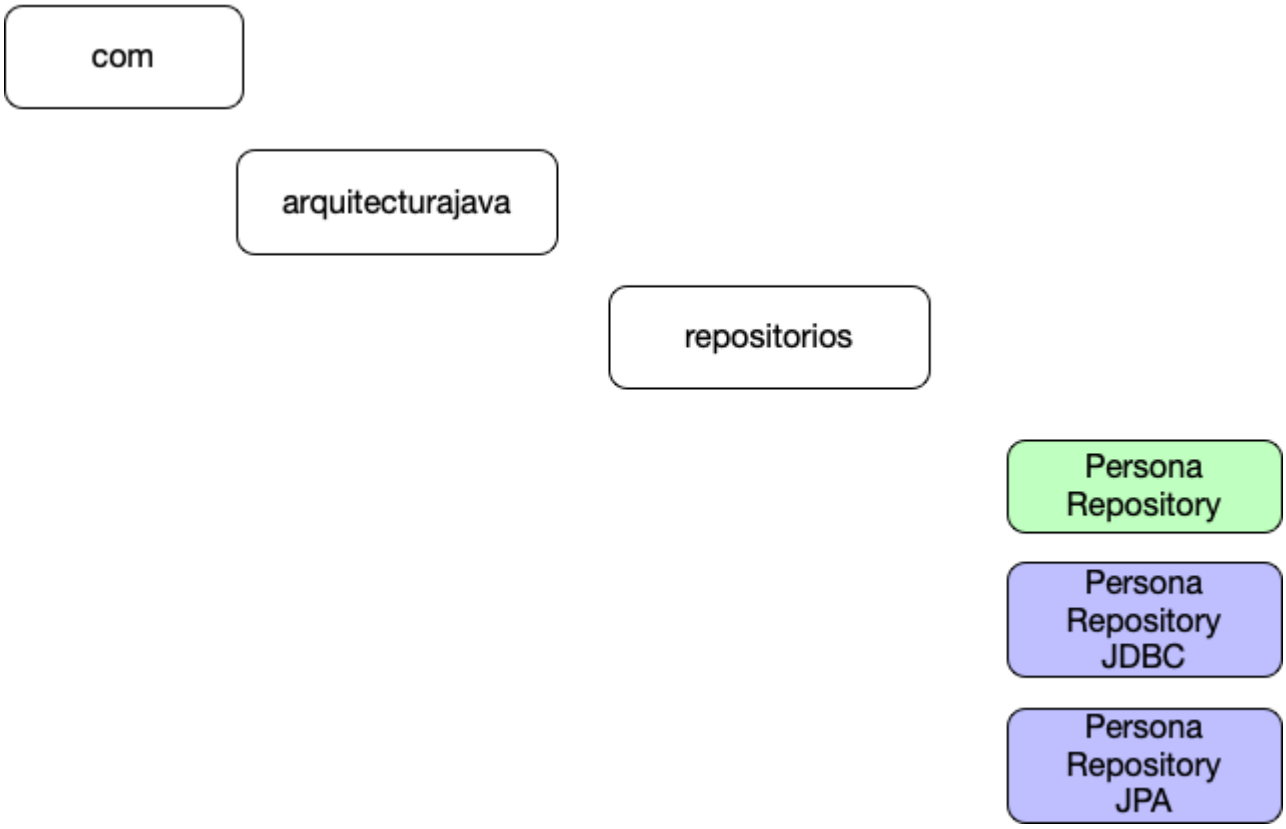
El concepto de Java Package es de sobra conocido por todos nosotros . Así pues todos usamos packages cuando programamos con el lenguaje pero muchas veces no pensamos demasiado en como organizar estos paquetes. Una de las dudas más habituales es como organizarlos a nivel de Interfaces e Implementaciones . Es decir cuando nosotros tenemos un Interface podemos disponer de una o más implementaciones. Un ejemplo clásico es el patrón Repositorio. En este patrón nosotros definimos una funcionalidad y la implementamos de cara a la capa de persistencia. Podemos tener persistencia JDBC y persistencia JPA.



Java Package Interfaces e Implementaciones

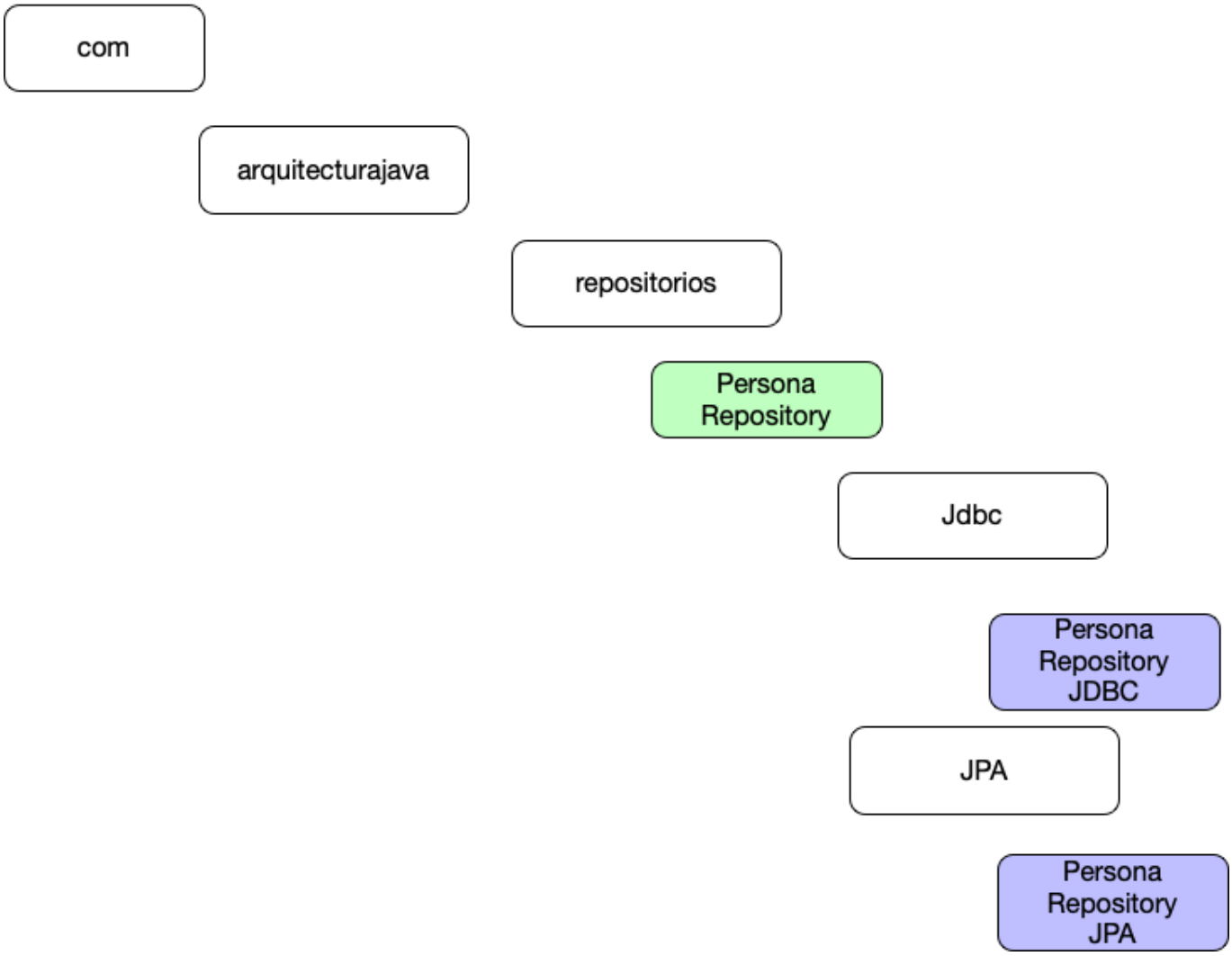
¿Cómo organizamos estas tres clases a nivel de Java Package ? . La realidad es que existen varias opciones y ninguna de ellas es la mejor sino que la organización se basa en principios de reutilización de las clases. La primera opción nos puede parecer demasiado sencilla y es todas las clases e interfaces en el mismo Package.



Esta nos puede parecer una solución pobre ya que estamos hablando de repositorios pero la realidad es que el propio API de Java la usa en muchos lugares como por ejemplo con el interface List y ArrayList ambas clases se encuentran ubicadas en el mismo package java.util. ¿Porque? . Porque se supone que siempre se van a utilizar de forma agrupada.

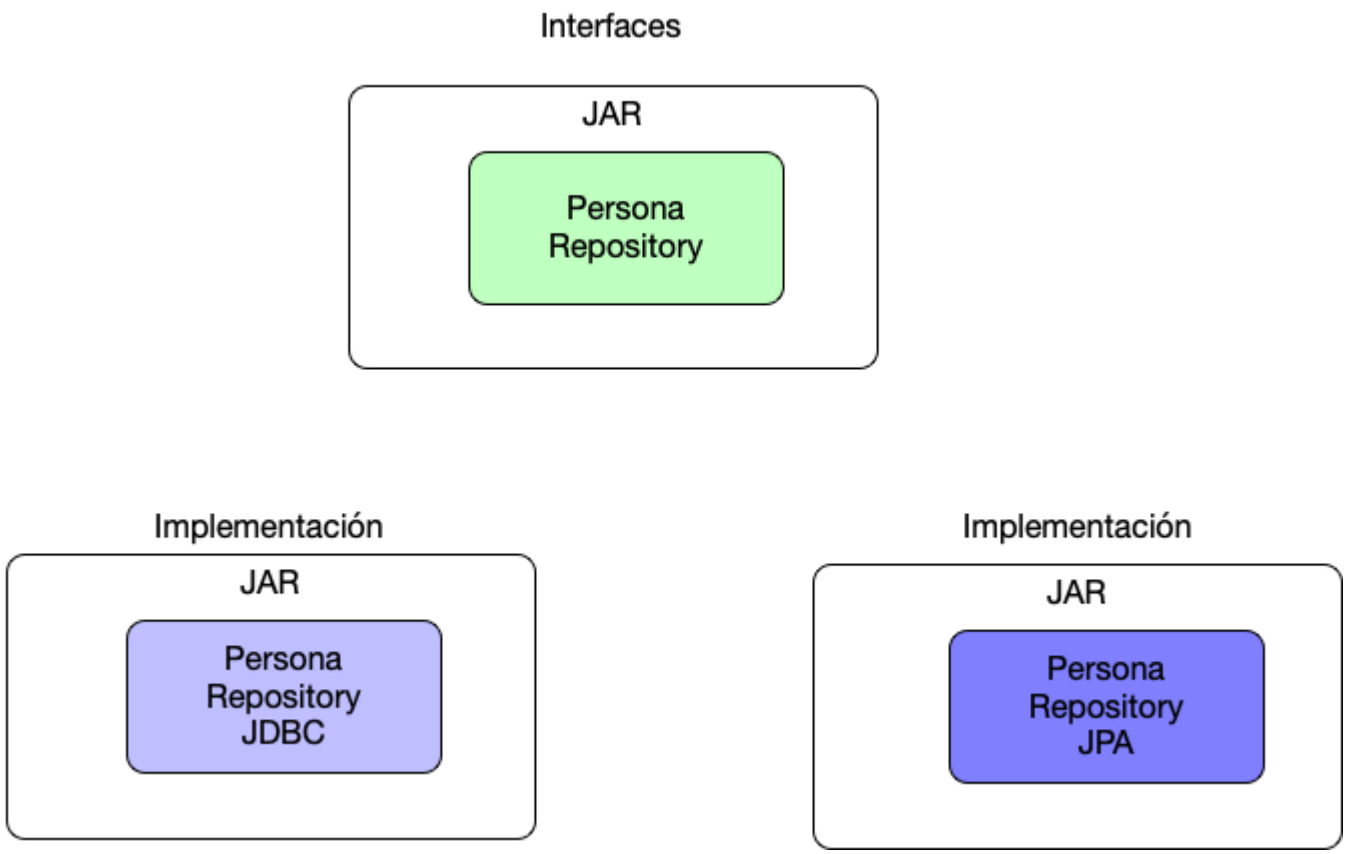
Java Packages y Jerarquías

La segunda opción que es bastante habitual es dividir en dos Packages jerárquicos la definición de los interfaces y sus diferentes implementaciones . Una opción podría ser :



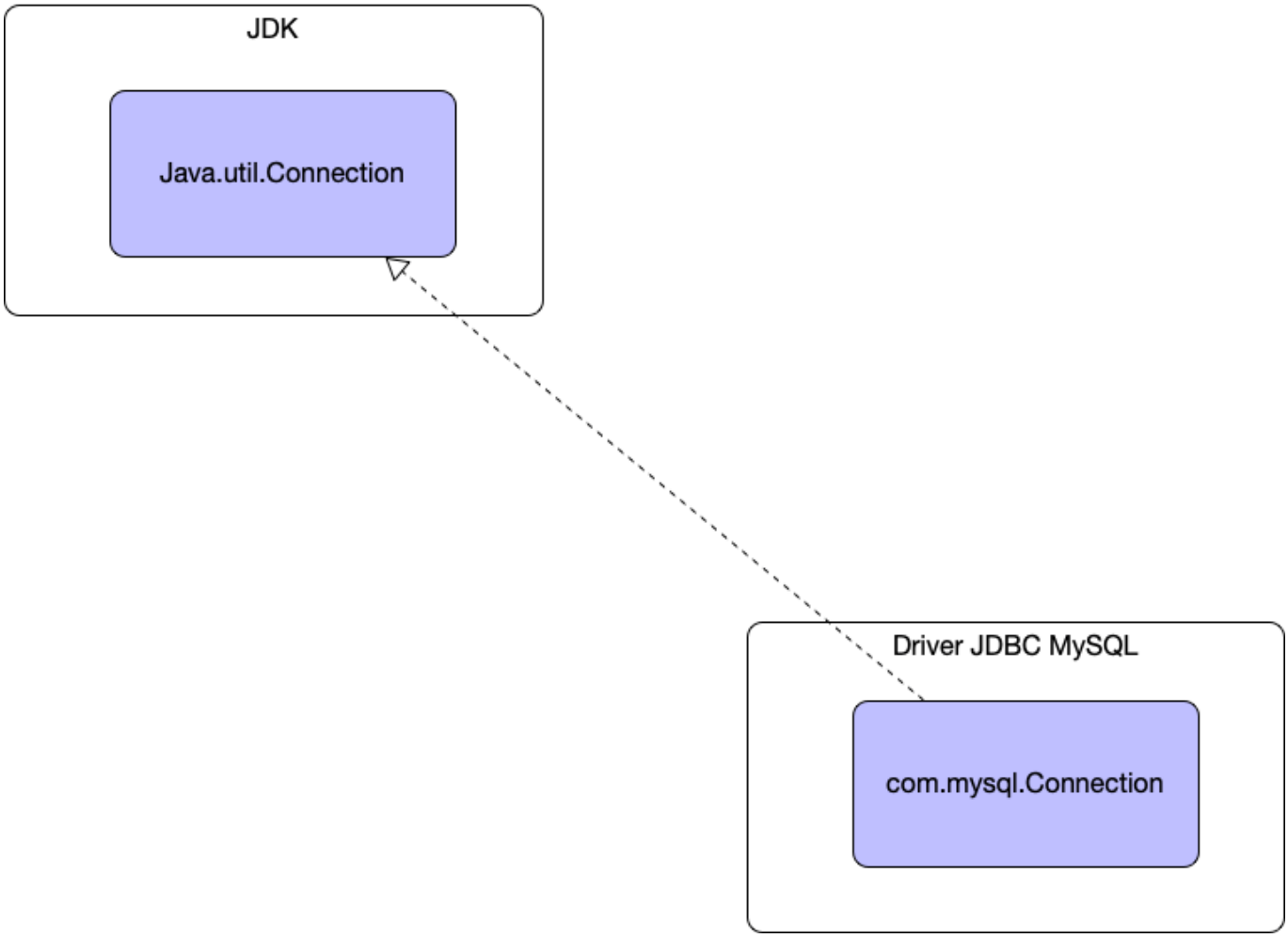
Java Packages y Jars

Esta es quizás la opción más clásica ya que incluso nos permitiría de una forma bastantes sencilla separar la definición de la implementación ya que en muchos casos estos nos puede resultar útil a nivel de Maven y gestión de dependencias muchas aplicaciones pueden necesitar solo los interfaces.



Java JARs y reutilización

Nos puede parecer que estas son las únicas dos formas de organizar los Packages a nivel de interfaces e implementaciones . Pero realmente existen otras más complejas y que siempre están basadas en el concepto de “reutilización y jars” .Un ejemplo clásico es el interface java.sql.Connection que se encuentra en el JDK y su package es java.sql . ¿Cual es su implementación para MySQL? . Esa implementación viene dada en el JAR que nos bajamos de conector y se encuentra en un package totalmente diferente.



Tengamos en cuenta sobre todo el concepto de reutilización a nivel de Packages a la hora de gestionar las estructuras en las cuales nosotros organizamos el código:

Otros artículos relacionados

1. El concepto de Java Package Encapsulation
2. Java Package Visibility
3. Java Packages vs JARs y su reutilización
4. Java Util Packages y sus clases