**Ejercicio 1.**

Implemente el ejercicio 4 de la Practica 1 usando el patrón *Doble Dispatching* (Visitor). Realice un modelo de clases que permita representar estos requerimientos adecuadamente. Piense qué patrón de diseño puede ayudarle y describa por qué.

**Ejercicio 2.**

Imaginemos que hemos desarrollado una aplicación en Windows que puede tener diferentes usuarios de base de datos (por ejemplo, un usuario usa Oracle y otro puede usar MySQL). Ahora, cada vez que necesitamos insertar datos en nuestra base de datos, necesitamos crear primero una conexión MysqlDataSource o una OracleConnection para luego, poder realizar consultas sobre el recurso creado. El arquitecto de software nos solicita que la lógica de instanciación de las conexiones de base de datos debe estar centralizada en un único recurso, el cual debe permitir la selección de la conexión deseada por el usuario.

Realice un modelo de clases que permita representar estos requerimientos adecuadamente. Es decir, piense qué patrón de diseño puede ayudarle y describa por qué.

**Ejercicio 3.**

Una empresa aeronáutica nos solicita modelar el proceso de la construcción de un avión, el cual implica los tres pasos en orden:

1. fabricación de la cabina.
2. el motor y
3. finalmente, las alas.

En la empresa aeronáutica, cada avión requiere al menos los tres pasos anteriores. Sin embargo, un avión de pasajeros puede tener un paso adicional de hacer baños en el avión. Los pasos representan el proceso de construcción. El producto es un avión, pero puede tener diferentes representaciones, como un F-16 o un Boeing-747. Utilizando el mismo proceso de construcción, deberíamos poder producir tanto F-16 como Boeings.

Realice un modelo de clases que permita representar estos requerimientos adecuadamente. Es decir, piense qué patrón de diseño puede ayudarle y describa por qué.