**IMPLEMENTAR CONEXIÓN A BASE DE DATOS ORACLE**

* **Especificación de requisito**

Se debe incluir una nueva funcionalidad que permita conectar la aplicación a una base de datos Oracle además de la posibilidad actual de realizarlo contra MySQL. Para ello además de implementar el driver de Oracle, se deberán revisar y modificar si procede las distintas clases que tienen acceso a datos.

* **Estrategia de solución usada**

La solución entregada se estructura en dos partes bien diferenciadas. Por un lado, se ha modificado la clase que realiza la conexión a base de datos, para que permita conectarse contra una base de datos Oracle, esto se hará mediante un fichero de configuración.

Por otra parte, se han tenido que modificar las clases que implementan DAO tanto en el módulo de socios como en el de libros. La solución adoptada ha sido extender estas clases ya que se ha tenido que modificar el método de inserción debido a que la generación de identificadores es diferente en MySQL que en Oracle. Por lo tanto, se ha sobrescrito este método, adaptándolo a la nueva funcionalidad.

Para la generación de identificadores en Oracle se ha creado una nueva clase, la cual contiene un método estático que devuelve el valor de la secuencia que corresponde.

* **Modificaciones realizadas sobre código existente**

**Clase ConnectionManager (modificación)**

Esta clase es la encargada de gestionar las conexiones contra base de datos. Ha sido modificada para que realice la conexión dependiendo de la propiedad *dbms* que se encuentra dentro del fichero *Config.properties.* En caso de que este valor sea *mysql*, realizara la conexión del modo habitual. Si su valor fuese *oracle*, generara la cadena de conexión para Oracle, permitiendo realizarse tanto mediante *service name* como mediante *SID*.

**Clase CustomersManagementController (modificación)**

Se ha modificado el constructor de la clase para que instancie el objeto *ICustomersDAO* a partir de la clase *CustomersDAOImpl* o *CustomersDAOImplOracle*, dependiendo del valor de la propiedad *dbms* del fichero de configuración.

**Clase BooksManagementController (modificación)**

Se ha modificado el constructor de la clase para que instancie el objeto *IBooksDAO* a partir de la clase *BooksDAOImpl* o *BooksDAOImplOracle*, dependiendo del valor de la propiedad *dbms* del fichero de configuración.

**CustomersDAOImplOracle (nueva clase)**

Esta nueva clase extiende de *CustomersDAOImpl* y sobrescribe el método *insert* ya que como se ha comentado antes es diferente la inserción de datos, ya que en *MySQL* el identificador se autogenera, mientras que en Oracle este se debe obtener a partir del valor de un objeto secuencia. Por lo tanto, en este nuevo método, asignamos el Id Socio mediante el valor obtenido en *UtilsOracle.generaSecuencia* para *SEQ\_SOCIO*.

**BooksDAOImplOracle (nueva clase)**

Esta nueva clase extiende de *BooksDAOImpl* y sobrescribe el método *insert* ya que como se ha comentado antes es diferente la inserción de datos, ya que en *MySQL* el identificador se autogenera, mientras que en Oracle este se debe obtener a partir del valor de un objeto secuencia. Por lo tanto, en este nuevo método, asignamos el Id Socio mediante el valor obtenido en *UtilsOracle.generaSecuencia* para *SEQ\_LIBRO*.

**UtilsOracle (nueva clase)**

La creación de esta clase surge a partir de la necesidad de obtener los identificadores a través de secuencias (objeto Oracle). Por ello se crea un nuevo método estático al cual se le envía la conexión actual, así como el nombre de la secuencia sobre la cual se desea obtener el nuevo valor. Esta conectará con la base de datos y consultará el valor *NEXTVAL* de la misma, el cual será devuelto por la función.

* **Dificultades encontradas**

Para la realización de este cambio se tuvo que montar un servidor Oracle en el cual generar un esquema similar al proporcionado para *MySQL*.

El mayor problema surgió como se ha comentado anteriormente con la generación de los identificadores para las tablas de socios y libros. Se decidió resolver mediante la creación de Secuencias ya que es la herramienta que se utiliza habitualmente cuando se trabaja en bases de datos Oracle.

* **Pruebas Unitarias**

Para probar la implementación de este cambio se realizaron las siguientes pruebas mediante la utilización de *JUnit* y *Mockito*:

* Se realiza *Mock* sobre la clase *UtilsOracle* y se comprueba que el valor retornado por el método estático *generaSecuencia* es el esperado.
* Se comprueba que se accede una vez a través del método *generaSecuencia*.

Cabe destacar que ha sido necesaria la utilización de la librería PowerMockito, ya que la versión estándar no permite la realización de pruebas con métodos estáticos. Estas pruebas se han realizado tanto para *BooksDAOImplOracle* como para *CustomersDAOImplOracle*.