



**Centro Profesional**  
Universidad Europea Madrid  
LAUREATE INTERNATIONAL UNIVERSITIES

**ESCUELA  
POLITÉCNICA**

CICLO FORMATIVO DE GRADO SUPERIOR  
EN  
DESARROLLO DE APLICACIONES MULTIPLATAFORMA



PROYECTO FIN DE CICLO

**SISTEMA DE RESERVAS DE LIBROS**

**VERONICA GONZALEZ JIMENEZ**  
**CURSO 2013-2014**



**Centro Profesional**  
Universidad Europea Madrid  
**LAUREATE** INTERNATIONAL UNIVERSITIES

**ESCUELA  
POLITÉCNICA**



## **RESUMEN**

El proyecto consta de un sistema de reserva de libros para una librería. Este software está orientado para que el usuario pueda tramitar altas, modificaciones y bajas de las editoriales, libros y socios de la biblioteca o librería concreta.

Está diseñado para que un libro solo pueda ser reservado por un socio y no estará disponible hasta que proceda a devolverlo. Será entonces cuando nuevamente podrá ser reservado dicho libro por otro socio o por el mismo.

Este software no requiere login para acceder a él, ya que está destinado para cualquier usuario que vaya hacer uso del mismo.

Este software se resume, en un sistema básico de reservas y devoluciones de libros, que existe hoy en día en cualquier biblioteca.

## **ABSTRACT**

The project consists of a backup system for a library book. this software is oriented so that the user can handle, modifications and downs of editorials, books and library members or specific Seller.

It is designed so that a book can only be booked by a member and not be available to proceed with the return. That's when you will again be reserved the book by another partner or by yourself.

This software does not require login to access it as it is intended for any user who will use it.

This software is summarized in a basic reservation system and book returns, which exists today in any library.



## **AGRADECIMIENTOS**

Antes de comenzar con el contenido del proyecto, me gustaría expresar mi más sincero agradecimiento a aquellas personas sin las cuales me hubiera sido imposible llevarlo a cabo:

A mis profesores de ciclo.

A todos mis familiares y amigos. En especial a mi tío que se lo prometí

Y por supuesto a mis compañeros de clase, por haber tenido apoyo y ayuda durante estos dos años de ciclo.



RESUMEN.....	3
ABSTRACT .....	3
AGRADECIMIENTOS.....	4
<b>CAPITULO 1. INTRODUCCION .....</b>	<b>7</b>
<i>Introducción.....</i>	<i>7</i>
<i>Objetivos del Sistema .....</i>	<i>7</i>
<i>Metodología a Utilizar.....</i>	<i>8</i>
<i>Recursos.....</i>	<i>8</i>
<b>CAPITULO 2. IDENTIFICACION A LAS NECESIDADES .....</b>	<b>9</b>
<i>Introducción a las necesidades.....</i>	<i>9</i>
<i>Objetivos del sistema.....</i>	<i>9</i>
<i>Alcance del sistema .....</i>	<i>10</i>
<b>CAPITULO 3. ESPECIFICACIONES.....</b>	<b>11</b>
<i>Actores.....</i>	<i>11</i>
<i>Jerarquía Usuarios.....</i>	<i>11</i>
<i>Diagrama de Casos de Uso .....</i>	<i>12</i>
<i>Descripción de Casos de Uso .....</i>	<i>13</i>
<b>CAPITULO 4. MODELO LOGICO DEL SISTEMA .....</b>	<b>17</b>
<i>Gestión de Socios .....</i>	<i>17</i>
<i>Gestión de Libros.....</i>	<i>18</i>
<i>Gestión de Proveedores .....</i>	<i>19</i>
<i>Gestión de Reserva.....</i>	<i>20</i>
<b>CAPITULO 5. MODELO CONCEPTUAL DE DATOS.....</b>	<b>21</b>
<i>Introducción modelo conceptual de datos.....</i>	<i>21</i>
<i>Diccionario de datos.....</i>	<i>21</i>
<i>Diagrama entidad-relación.....</i>	<i>22</i>
<i>Relaciones Principales .....</i>	<i>22</i>
<i>Tabla de las Entidades.....</i>	<i>23</i>



<b>CAPITULO 6. PRUEBAS .....</b>	<b>24</b>
<i>Introducción a las Pruebas.....</i>	<i>24</i>
<i>Tipos de Pruebas .....</i>	<i>24</i>
<i>Pruebas.....</i>	<i>25</i>
<b>CAPITULO 7. CONCLUSIÓN.....</b>	<b>28</b>
<i>Conclusión.....</i>	<i>28</i>
<b>CAPITULO 8. FUTURAS MEJORAS .....</b>	<b>29</b>
<i>Futuras Mejoras.....</i>	<i>29</i>
<b>CAPITULO 9. BIBLIOGRAFÍA.....</b>	<b>30</b>
<i>Bibliografía .....</i>	<i>30</i>



## **CAPITULO 1. INTRODUCCION**

### **Introducción**

El mundo de las bibliotecas existe desde hace muchísimos años, debido a la gran demanda que existe en el mundo por el interés de los libros y la cultura. Hoy en día es un pilar fundamental, para los estudios básicos y para seguir aprendiendo a lo largo de toda una vida.

Actualmente internet es el pilar fundamental para la búsqueda de información, pero los libros siguen siendo algo muy importante para la sociedad y cultura de un país.

Por ese motivo, en todo centro público o privado, existe una biblioteca donde poder buscar la información necesaria, ya sea a través de un libro de papel, electrónico o documentación varia.

Por todo ello, he querido lanzarme a realizar un proyecto basado en el control de reservas y devoluciones que las personas realizan en cualquier librería o Biblioteca.

A pesar de las nuevas tecnologías, la gente sigue queriendo acudir a un lugar tranquilo y lleno de libros donde poder consultar lo que se quiera y además, la posibilidad de llevarte a casa un libro por un periodo de tiempo y ahorrar en costes de compra.

Este proyecto está realizado en Java y con ello no queremos competir con otros sistemas existentes en la actualidad, sino dar a conocer otro tipo de sistema creado con una interfaz sencilla y fácil de manejar para el usuario final.

A continuación os detallo en dicho documento el proceso llevado a cabo para la realización de este sistema, que base de datos se ha utilizado para el mismo, herramientas y recursos y conocimientos alcanzados.

### **Objetivos del Sistema**

Los objetivos que se desean cumplir con este proyecto son la gestión de datos de una librería o biblioteca y el procesamiento de esos datos para obtener información útil en la gestión de una biblioteca o librería. Para ello, se creará un sistema de gestión de una biblioteca que realice:

- Gestión de Socios
- Gestión de Libros
- Gestión de Proveedores
- Gestión de Reservas



Este sistema realizará el control de reservas y devoluciones de los libros que reservan los socios. Se manejarán todas las nuevas altas y bajas de los libros de la base de datos, además del control de los proveedores que sirven dichos libros.

## Metodología a Utilizar

La metodología a utilizar se basará en una de las estudiadas durante el ciclo, en concreto la metodología en cascada. Esta metodología consiste en que a la finalización de una fase comienza la siguiente, tomando como datos de entrada los resultados de la fase anterior.

En cada fase se llega a más detalle, hasta construir el código. Este sistema es el más extendido. El inconveniente es que no se ven los resultados hasta avanzado el proyecto.

## Recursos

Para llevar a cabo el proyecto se necesitarán los siguientes recursos hardware, software y de otros tipos:

- **Recursos Hardware.** Se utilizará un ordenador personal, el cual tendrá conexión a Internet, de esta manera se tendrá acceso a información útil.
- **Recursos Software.** Se utilizará Eclipse, para el desarrollo del código, que se hará en el lenguaje de programación Java. Como base de datos, se utilizará Mysql con Hibernate. También se utilizarán programas como Microsoft Word o Power Point, del paquete de Office 2007 para redactar la documentación, preparación de presentación.... Sistema operativo a utilizar será el Windows 7 Professional.
- **Otros Recursos.** Se utilizará documentación recopilada de internet, como por ejemplo teoría basada en Hibernate, cursos de programación orientada objetos, para la mejora de la estructura del programa.





## CAPITULO 2. IDENTIFICACION A LAS NECESIDADES

### Introducción a las necesidades

Es esta etapa se definen los problemas que mediante la aplicación se pretenden resolver. Además se establecen los límites del proyecto, los objetivos y antecedentes de la aplicación.

Información en la etapa de Identificación de Necesidades. Esta información consiste en:

- ❖ **Objetivos del sistema:** Son objetivos de tipo informático
- ❖ **Alcance del sistema:** Funciones que pretenden automatizarse.
- ❖ **Tipología de Usuarios:** Conocer el perfil de personas a quien va dirigida la aplicación.
- ❖ **Restricciones:** Deben considerarse aquellas restricciones que puedan afectar al plan de proyecto y su desarrollo.
- ❖ **Antecedentes:** Comprender como se realizan los procesos a tratar el proyecto para lograr entender cuáles han sido los motivos para optar por la implantación de este sistema.

### Objetivos del sistema

Los objetivos del sistema son los que quieren alcanzar en el entorno informático.

Con este proyecto se pretende conseguir una aplicación de gestión de una biblioteca, para que los empleados de la misma, puedan tener un control de todos los registros de clientes, libros, proveedores y reservas diarias, que se van a realizar en la propia biblioteca. Por otro lado, la aplicación deberá ser intuitiva y fácil de manejar para los empleados.

#### OBJETIVOS FUNCIONALES

- ❖ **Creación de una aplicación de fácil manejo.** Deberá ser de fácil manejo ya que será utilizado por personas no expertas en informática. Por lo que la información que reciban deberá ser breve, para que puedan usar el sistema rápidamente.
- ❖ **Creación de una base de datos.** Toda la información que maneja la biblioteca o librería, deberá estar recogida en una base de datos a la que podrán acceder mediante el sistema. De esta manera, al tener toda la información en un mismo lugar informatizada se tendrá un mejor control de la información.
- ❖ **Accesibilidad según el usuario.** Actualmente será programa para que cualquier usuario pueda utilizarlo, debido a que normalmente en una biblioteca, el empleado principal es el encargado de las reservas de libros y demás.



- ❖ **Facilitar la actualización de los contenidos.** En la actualidad, aun existen bibliotecas en centros públicos, que la ficha de los clientes esta en papel, por lo que cualquier modificación de los datos, se tiene que volver a copiar todo el documento, esto consume recursos y tiempo por parte del empleado que en ese momento no puede atender a un cliente que desea reservar un libro.

## **Alcance del sistema**

El alcance del sistema cubre las funciones de negocio que actualmente se hacen de manera manual y que pretenden automatizarse mediante la aplicación.

El sistema cubrirá las siguientes funcionalidades:

**Gestión de Socios:** creación de registros de nuevos socios, introduciendo los datos del socio en una tabla de socio.

**Gestión de Proveedores:** creación de registros de nuevos proveedores, introduciendo los datos del proveedor en una tabla de proveedores.

**Gestión de Libros:** Control de las unidades disponibles de libros de la librería o biblioteca, accediendo y manteniendo actualizada la tabla.

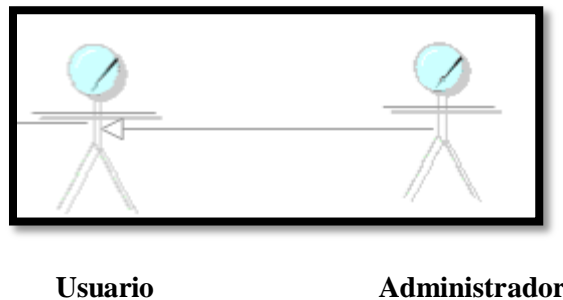
## CAPITULO 3. ESPECIFICACIONES

La fase de especificación es imprescindible para que un proyecto resulte exitoso. Por ello, se proporcionan diferentes modelos y diagramas para realizarla con el máximo detalle posible.

### Actores

Los actores son los usuarios que interactúan de forma directa con el sistema. En el sistema se encuentran los siguientes actores: Usuario Principal y Administrador

### Jerarquía Usuarios



#### Usuario

Se entiende por usuario aquella persona que utilizará el sistema para realizar todas las consultas, altas, bajas y modificaciones de los socios, libros y proveedores pertenecientes a la biblioteca.

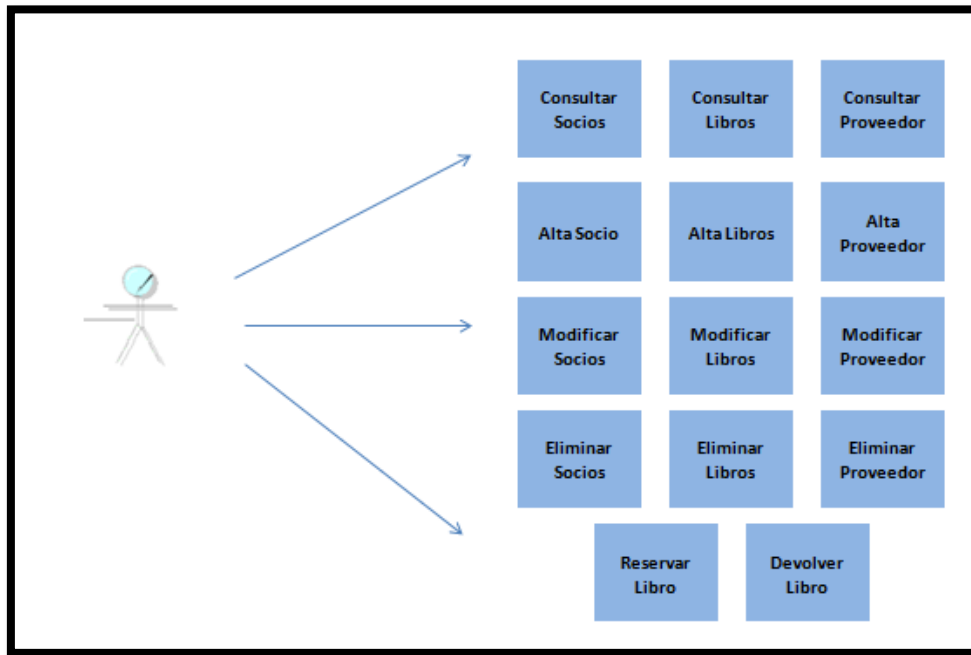
Es el encargado de usar el sistema para gestionar todas las reservas y devoluciones de los socios de la biblioteca

#### Administrador

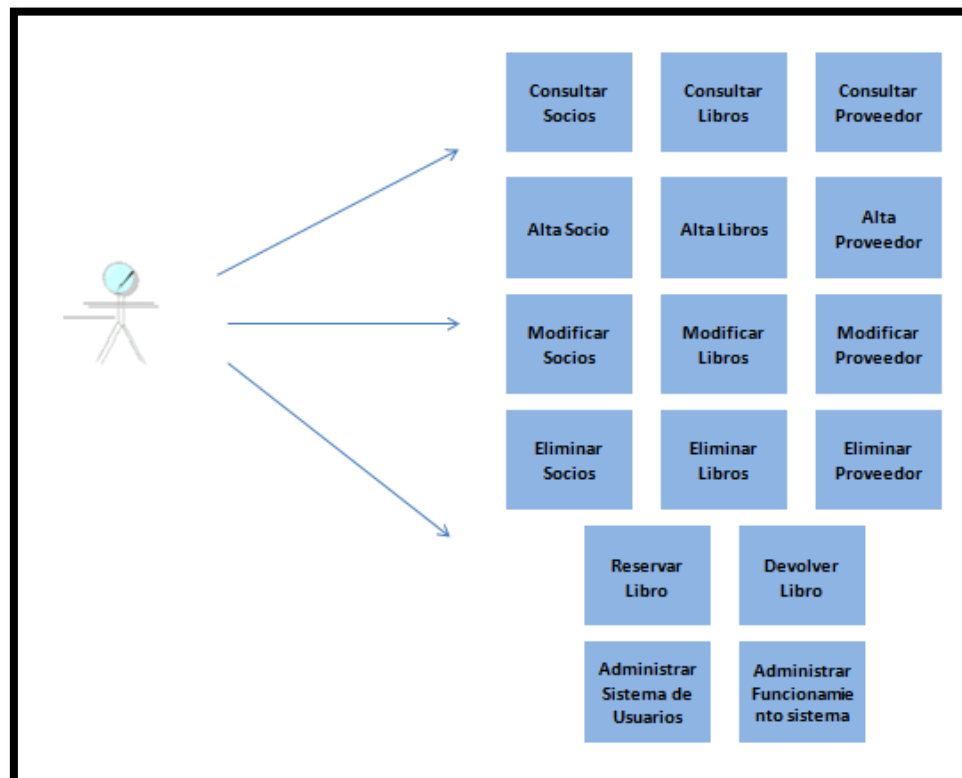
Se entiende por Administrador, aquella persona que maneja el funcionamiento de todo el sistema y tiene el permiso de modificar cualquier estructura o parte del sistema para su correcto funcionamiento y dota de todos los permisos existentes en el programa

## Diagrama de Casos de Uso

### Usuario



### Administrador





## Descripción de Casos de Uso

Para una mayor claridad se han clasificado los casos de uso según el actor principal del mismo.

### USUARIO

#### --Consultar Socios--

- Descripción: El usuario consulta información detallada de un socio en concreto o varios.
- Precondición: Conexión a la BBDD y buscar el socio previamente
- PostCondicion: El sistema muestra la información del socio
- Flujo Normal: La consulta se realizará correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna consulta

#### --Alta Socio--

- Descripción: El usuario introduce los datos de un nuevo socio para darlo de alta en el sistema.
- Precondición: Conexión a la BBDD e introducir los datos previamente
- PostCondicion: El sistema muestra da de alta a un socio nuevo
- Flujo Normal: El alta del socio se realizara correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá el alta del socio en el sistema

#### --Modificar Socio--

- Descripción: El usuario modifica los datos de un socio ya existente en el sistema
- Precondición: Conexión a la BBDD y buscar socio previamente para poder modificarlo
- PostCondicion: El sistema guarda los nuevos datos en el sistema
- Flujo Normal: La modificación se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna modificación en el sistema

#### --Eliminar Socio--

- Descripción: El usuario elimina un socio del sistema
- Precondición: Conexión a la BBDD y buscar socio previamente para poder eliminarlo
- PostCondicion: El socio es eliminado del sistema
- Flujo Normal: La eliminación del socio se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna eliminación en el sistema



#### **--Consultar Libros--**

- Descripción: El usuario consulta información detallada de un libro en concreto o varios.
- Precondición: Conexión a la BBDD y buscar el libro previamente
- PostCondicion: El sistema muestra la información del libro
- Flujo Normal: La consulta se realizará correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna consulta

#### **--Alta Libro--**

- Descripción: El usuario introduce los datos de un nuevo libro para darlo de alta en el sistema.
- Precondición: Conexión a la BBDD e introducir los datos previamente
- PostCondicion: El sistema muestra da de alta a un libro nuevo
- Flujo Normal: El alta del libro se realizara correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá el alta del libro en el sistema

#### **--Modificar Libro--**

- Descripción: El usuario modifica los datos de un libro ya existente en el sistema
- Precondición: Conexión a la BBDD y buscarlibro previamente para poder modificarlo
- PostCondicion: El sistema guarda los nuevos datos en el sistema
- Flujo Normal: La modificación se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna modificación en el sistema

#### **--Eliminar Libro--**

- Descripción: El usuario elimina un libro del sistema
- Precondición: Conexión a la BBDD y buscar el libro previamente para poder eliminarlo
- PostCondicion: El libro es eliminado del sistema
- Flujo Normal: La eliminación del libro se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna eliminación en el sistema



#### **Consultar Proveedor--**

- Descripción: El usuario consulta información detallada de un proveedor en concreto o varios.
- Precondición: Conexión a la BBDD y buscar el proveedor previamente
- PostCondicion: El sistema muestra la información del proveedor
- Flujo Normal: La consulta se realizará correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna consulta

#### **--Alta Proveedor--**

- Descripción: El usuario introduce los datos de un nuevo proveedor para darlo de alta en el sistema.
- Precondición: Conexión a la BBDD e introducir los datos previamente
- PostCondicion: El sistema muestra da de alta a un proveedor nuevo
- Flujo Normal: El alta del proveedor se realizara correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá el alta del proveedor en el sistema

#### **--Modificar Proveedor--**

- Descripción: El usuario modifica los datos de un proveedor ya existente en el sistema
- Precondición: Conexión a la BBDD y buscar proveedor previamente para poder modificarlo
- PostCondicion: El sistema guarda los nuevos datos en el sistema
- Flujo Normal: La modificación se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna modificación en el sistema

#### **--Eliminar Proveedor--**

- Descripción: El usuario elimina un proveedor del sistema
- Precondición: Conexión a la BBDD y buscar el proveedor previamente para poder eliminarlo
- PostCondicion: El proveedor es eliminado del sistema
- Flujo Normal: La eliminación del proveedor se realiza correctamente en el sistema
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna eliminación en el sistema



#### **--Reservar Libro--**

- Descripción: El usuario reserva un libro o libros a un socio determinado
- Precondición: Conexión a la BBDD y buscar el socio y libro previamente
- PostCondicion: El libro es reservado por un socio en concreto
- Flujo Normal: Se realiza la reserva correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna reserva en el sistema

#### **--Devolver Libro--**

- Descripción: El usuario realiza la devolución que realiza un socio de un libro determinado
- Precondición: Conexión a la BBDD y buscar al socio previamente, indicando el libro a devolver
- PostCondicion: El libro es devuelto en el sistema correctamente
- Flujo Normal: Se realiza la devolución correctamente
- Flujo Alternativo: Si no hay conexión a BBDD, no se producirá ninguna devolución en el sistema

### **ADMINISTRADOR**

Realizara todas las operaciones mencionadas anteriormente realizadas por el usuario y además estas otras:

#### **--Administrar Sistema de Usuarios--**

- Descripción: El administrador realiza todo el control de acceso y permisos de los usuarios del sistema
- Precondición: No hay
- PostCondicion: La administración del control y permisos se realiza sin problema
- Flujo Normal: Se realiza el control de usuarios correctamente
- Flujo Alternativo: No hay

#### **--Administrar Funcionamiento del Sistema--**

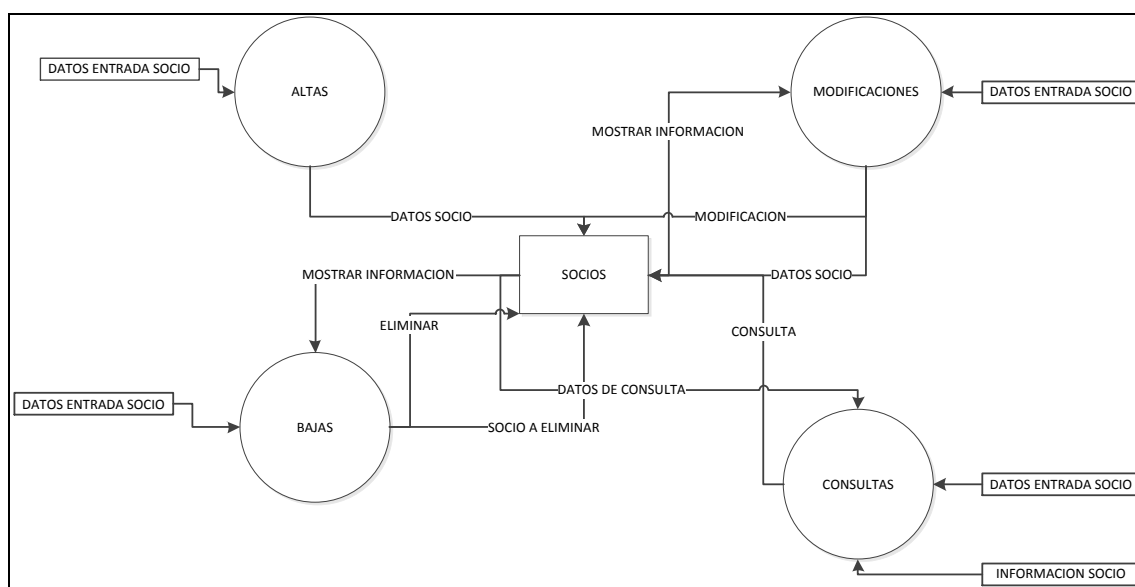
- Descripción: El administrador realiza todo el mantenimiento, control y supervisión del sistema
- Precondición: No hay
- PostCondicion: La administración y funcionamiento del sistema se realiza sin problema
- Flujo Normal: Se realiza el control la administración del sistema correctamente
- Flujo Alternativo: No hay



## CAPITULO 4. MODELO LOGICO DEL SISTEMA

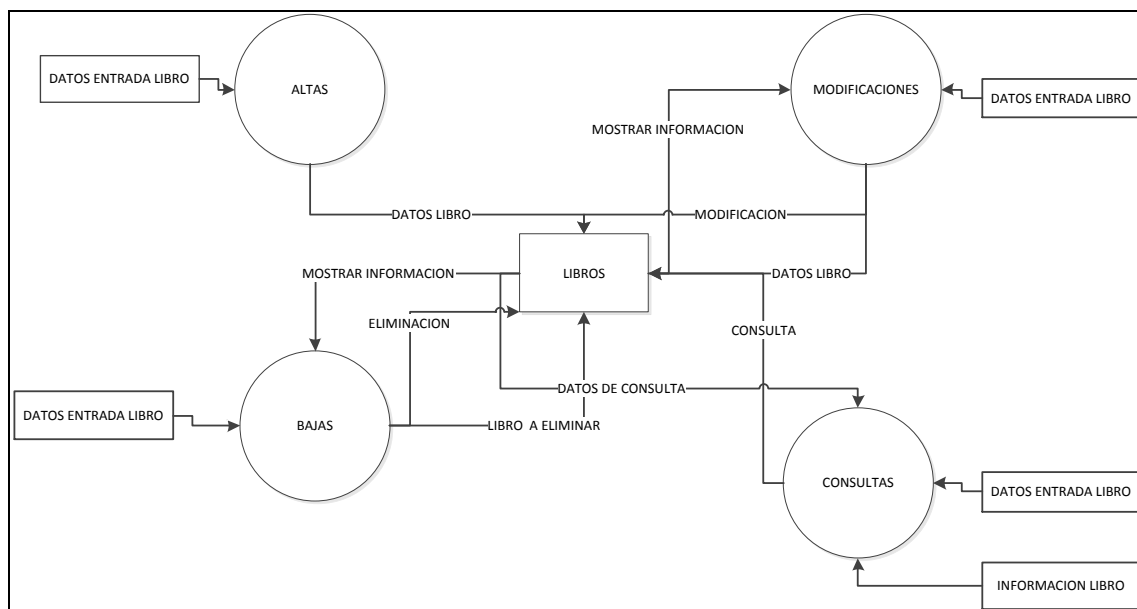
### Gestión de Socios

En gestión de socios se tratarán los datos de los socios, como las altas, modificaciones y bajas, que lo realizará el empleado de turno, en ningún caso el propio socio de la biblioteca o librería. Estos datos se almacenarán en la tabla SOCIO a la cual se podrá acceder para cualquier tipo de modificación o consulta.



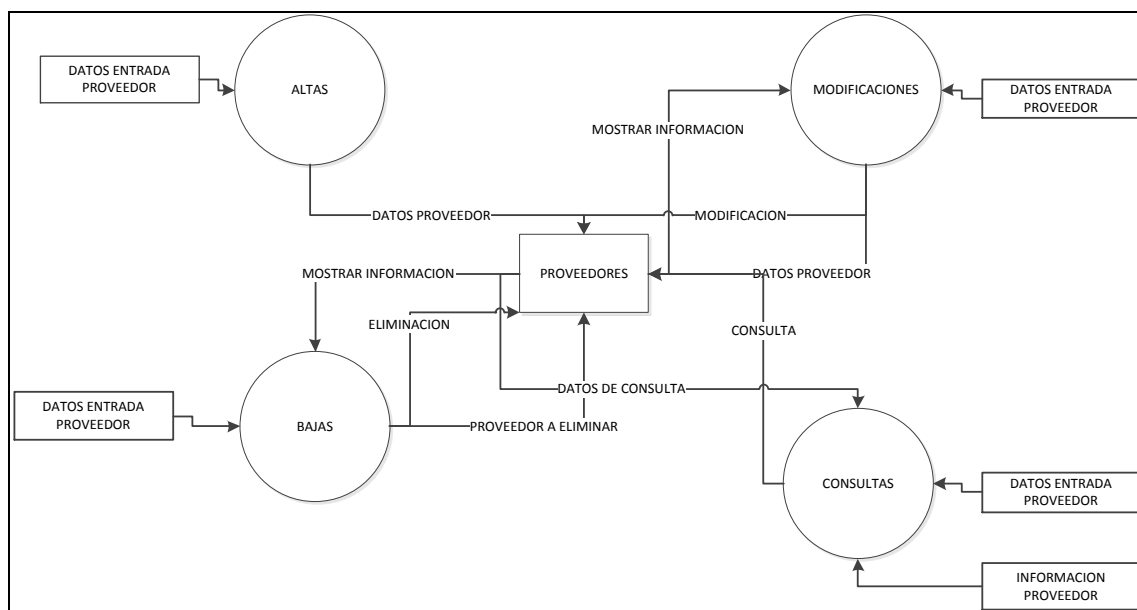
## Gestión de Libros

En gestión de Libros se tratarán los datos de los Libros, los datos para el alta de libros, modificaciones y bajas los realizará el empleado de turno. Estos datos se almacenarán en la tabla de LIBRO, a la cual se podrá acceder para realizar cualquier tipo de modificación o consulta.



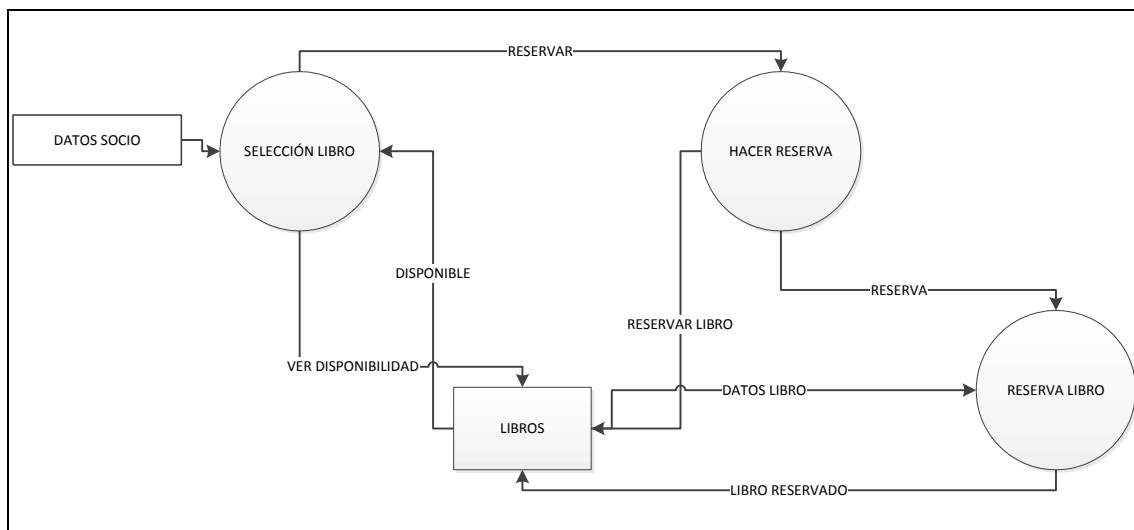
## Gestión de Proveedores

En gestión de Proveedores se tratarán los datos de los Proveedores, los datos para el alta de proveedores, modificaciones y bajas los realizará el empleado de turno. Estos datos se almacenarán en la tabla de PROVEEDOR, a la cual se podrá acceder para realizar cualquier tipo de modificación o consulta.



## Gestión de Reserva

Se realizaran las reservas, para ello son necesarios los datos del socio, para saber para quien es la reserva y los datos del libro que se reserva. En este caso el empleado procederá a realizar el trámite de reserva, comprobando su disponibilidad y si esta disponible, el empleado tramitara la reserva. Estos datos se almacenaran en la tabla de RESERVA.





## CAPITULO 5. MODELO CONCEPTUAL DE DATOS

### Introducción modelo conceptual de datos

En este apartado se describen las características principales de los datos del sistema. El modelo conceptual de datos se describe las entidades, los atributos y las relaciones de interés para la aplicación. Este modelo es independiente del hardware y software utilizado para el manejo de los datos y de las aplicaciones que utilicen dichos datos.

- **Entidades:** Objetos que tienen una existencia propia conforme a las decisiones de gestión de la empresa. Es aquello de interés duradero para la empresa, sobre lo cual se pueden almacenar datos e identificar de un modo único.
- **Relaciones:** Representación de asociaciones entre entidades, acorde con las decisiones de gestión de la empresa. Las relaciones establecen el grado de asociación entre dos estructuras de datos diferentes.
- **Atributos:** Datos elementales conforme a las decisiones de gestión de la empresa y las relaciones. Son características o propiedades de una entidad y sirven para definir, describir y clasificar. Los atributos llamados clave o identificadores permiten identificar sin ambigüedad cada valor de la entidad.
- **Clave:** La superclave de menor número de atributos
- **Superclave:** Dada una relación, un atributo o subconjunto de atributos es clave si permite distinguir unívocamente las tuplas de la relación.

### Diccionario de datos

A continuación se detalla el diccionario de datos de la aplicación.

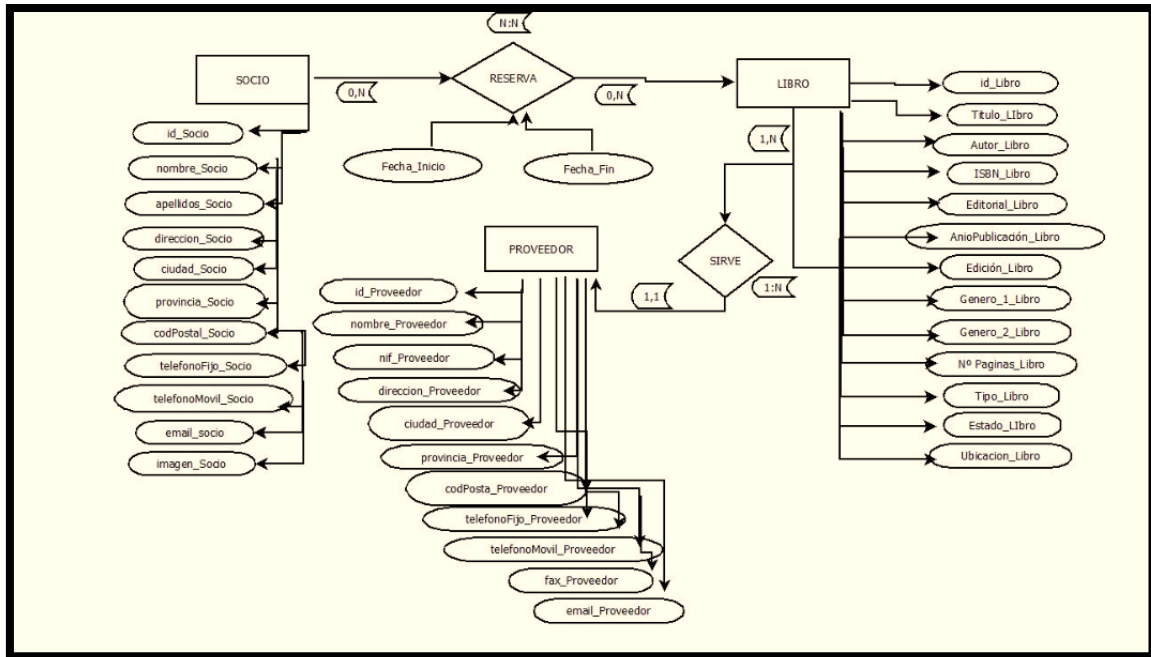
**SOCIO**={ID\_SOCIO + DNI\_SOCIO + NOMBRE\_SOCIO + APELLIDOS\_SOCIO + DIRECCION\_SOCIO + CIUDAD\_SOCIO + PROVINCIA\_SOCIO + CODPOSTAL\_SOCIO + TELEFONOFIJO\_SOCIO + TELEFONOMOVIL\_SOCIO + EMAIL\_SOCIO + IMAGEN\_SOCIO}

**PROVEEDOR**={ID\_PROVEEDOR + NOMBRE\_PROVEEDOR + NIF\_PROVEEDOR + DIRECCION\_PROVEEDOR + CIUDAD\_PROVEEDOR + PROVINCIA\_PROVEEDOR + CODPOSTAL\_PROVEEDOR + TELEFONOFIJO\_PROVEEDOR + TELEFONOMOVIL\_PROVEEDOR + FAX\_PROVEEDOR + EMAIL\_PROVEEDOR}

**LIBRO**={ID\_LIBRO + ID\_PROVEEDOR + TITULO\_LIBRO + AUTOR\_LIBRO + ISBN\_LIBRO + EDITORIAL\_LIBRO + ANIOPUBLICACION\_LIBRO + EDICION\_LIBRO + GENERO1\_LIBRO + GENERO2\_LIBRO + NPAGINAS\_LIBRO + TIPO\_LIBRO + ESTADO\_LIBRO + UBICACIÓN\_LIBRO}

**RESERVA**={ID\_RESERVA + ID\_SOCIO + ID\_PROVEEDOR + ID\_LIBRO +  
FECHA\_INICIO + FECHA\_FIN}

## Diagrama entidad-relación



## Relaciones Principales

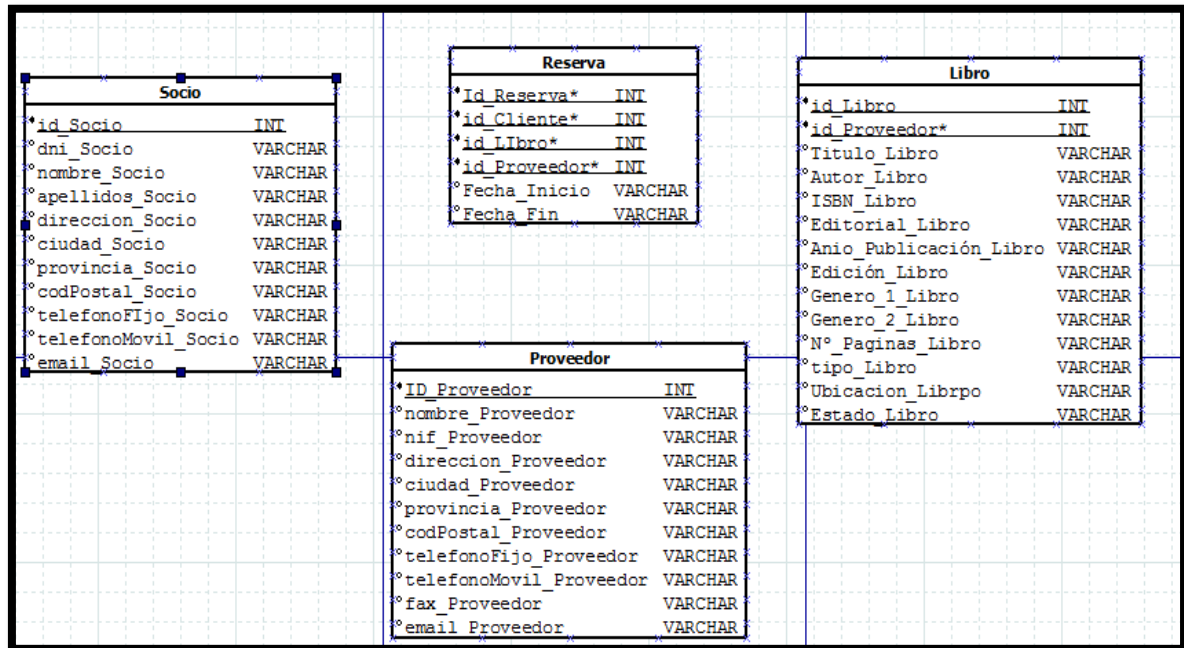
RESERVADOS = {ID\_RESERVA + ID\_LIBRO}

RESERVADORES = {ID\_RESERVA + ID\_SOCIO}

SUMINISTRAR = {ID\_PROVEEDOR + ID\_LIBRO}



## Tabla de las Entidades





## CAPITULO 6. PRUEBAS

### Introducción a las Pruebas

Una vez desarrollados y probados cada uno de los programas que forman el software, deben realizarse una serie de pruebas para conseguir integrar todo el sistema.

El objetivo global de esta fase es someter al sistema desarrollado y a sus componentes a una serie de verificaciones encaminadas a garantizar un nivel de fiabilidad aceptable.

Si los resultados de las pruebas son satisfactorios, se procederá a la aceptación y a la implantación del sistema. En caso de no ser satisfactorios habrá que corregir las anomalías hasta que el resultado sea satisfactorio.

Como consecuencia de las pruebas realizadas en un entorno de características similares al de producción, se desarrollará el manual de instalación y configuración.

### Tipos de Pruebas

En esta etapa se realizan diversas pruebas, cada una con diferentes objetivos. Se realiza un bloque de pruebas con el objetivo de comprobar la funcionalidad y rendimiento exigido en los requisitos. Para ello, previamente se habrán realizado las pruebas unitarias de cada componente software.

Los tipos de pruebas pueden clasificarse de la siguiente manera:

- **Pruebas de Encadenamiento:** Verifican las llamadas entre componentes.
- **Pruebas de Integración:** Verifican la funcionalidad de todo el sistema integrado, y el rendimiento de los recursos utilizados.
- **Pruebas de Explotabilidad:** Verifican la correcta operación del sistema.
- **Pruebas de Usabilidad:** Certifican, por parte de los usuarios, la manejabilidad y amigabilidad del sistema.



## Pruebas

- Pruebas de Interfaz Gráfica

ID PRUEBA	DESCRIPCION	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO PRUEBA
001_INTF	El metodo hace invisibles las ventanas internas anteriores, para poder mostrar la nueva ventana con otro metodo distinto	Hace invisible la ventana interna anterior	CORRECTO	FINALIZADA
002_INTF	El metodo controla la visibilidad de las ventanas a las que el usuario se dirige al presionar los eventos de los componentes	Hace visible la ventana que le mando por parametros	CORRECTO	FINALIZADA
003_INTF	Este metodo reinicia todos los textField de mi ventana a traves de un JButton, dejandolas vacias, para volver a rellenarlas.	Limpiar todas las cajas de texto o reiniciar los componentes de la ventana a vacio o null.	CORRECTO	FINALIZADA
004_INTF	Este metodo va a crear una tabla vacia con el nombre de las columnas indicadas y con un scroll incluido	Crea un Jtable con las características correspondientes	CORRECTO	FINALIZADA
005_INTF	Este metodo vacia la tabla a traves de un for que va a ir eliminando cada fila de la tabla	Reinicia la tabla, vaciado las filas	CORRECTO	FINALIZADA
006_INTF	Este metodo leer la cadena de texto que contienen las filas de la tabla, para luego trasladar la informacion a los textField de la propia ventana	lee los datos contenidos en las filas de la tabla y los muestra en cajas de texto	CORRECTO	FINALIZADA
007_INTF	Este metodo evalua primero que opcion del comboBox es la elegida y en funcion de la misma, leera la fila seleccionada en la tabla y rellenara los textField	realiza la lectura de la fila seleccionada, en funcion de la opcion elegida en el combobox	CORRECTO	FINALIZADA
008_INTF	Este metodo recorre una lista de objetos en funcion de la busqueda realizada, para poder acceder a sus propiedades y guardarlas en una matriz para luego utilizar dichos datos como se quiera.	recorre una lista con el objeto buscado, para mostrar las propiedades del mismo y guardarlas en una matriz	CORRECTO	FINALIZADA
009_INTF	Este metodo guardara los datos de los textfield de una clase en concreta para pasarlos a otra clase diferente y poder usarlos	guarda los datos de los textField en un vector	CORRECTO	FINALIZADA
010_INTF	Este metodo deshabilita las cajas de texto para no ser modificadas ni poder escribir en ellas	deshabilita todos los textfield de la ventana	CORRECTO	FINALIZADA
011_INTF	Este metodo habilita las cajas de texto para modificarlas o escribir en ellas	habilita todos los textfield de la ventana	CORRECTO	FINALIZADA
012_INTF	Este metodo limpia todos los componentes del panel, dejandolos de inicio	reinicia todos los componentes del panel	CORRECTO	FINALIZADA



- Pruebas de Base de Datos

ID PRUEBA	DESCRIPCION	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO PRUEBA
013_BBDD	Busqueda de un Socio en BBDD a traves del Id_Socio	realiza la busqueda del id_Socio	CORRECTO	FINALIZADA
014_BBDD	Busqueda de un Socio en BBDD a traves de cualquier otro campo que no sea el Id_Socio	realiza la busqueda del id_Socio realizando la busqueda de cualquier campo	CORRECTO	FINALIZADA
015_BBDD	Busqueda de un Libro en BBDD a traves del Id_Libro	realiza la busqueda del id_Libro	CORRECTO	FINALIZADA
016_BBDD	Busqueda de un Libro en BBDD a traves de cualquier otro campo que no sea el Id_Libro	realiza la busqueda del id_Libro realizando la busqueda de cualquier campo	CORRECTO	FINALIZADA
017_BBDD	Busqueda de un Proveedor en BBDD a traves del Id_Proveedor	realiza la busqueda del id_Proveedor	CORRECTO	FINALIZADA
018_BBDD	Busqueda de un Proveedor en BBDD a traves de cualquier otro campo que no sea el Id_Proveedor	realiza la busqueda del id_Proveedor realizando la busqueda de cualquier campo	CORRECTO	FINALIZADA
019_BBDD	Busqueda de las reservas de un Id_Socio en concreto	realiza la busqueda de todas las reservas de un Socio en concreto	CORRECTO	FINALIZADA
020_BBDD	Guarda un socio nuevo en BBDD, con la informacion introducida en los textField	realiza la introduccion en BBDD de los datos del Socio	CORRECTO	FINALIZADA
021_BBDD	Guarda un Libro nuevo en BBDD, con la informacion introducida en los textField	realiza la introduccion en BBDD de los datos del Libro	CORRECTO	FINALIZADA
022_BBDD	Guarda un Proveedor nuevo en BBDD, con la informacion introducida en los textField	realiza la introduccion en BBDD de los datos del Proveedor	CORRECTO	FINALIZADA
023_BBDD	Modifica los datos de un Socio en BBDD, con la informacion modificada en los textField, con el evento del boton modificar.	realiza la actualizacion de datos en BBDD de los datos de un socio	CORRECTO	FINALIZADA
024_BBDD	Modifica los datos de un Libro en BBDD, con la informacion modificada en los textField, con el evento del boton modificar.	realiza la actualizacion de datos en BBDD de los datos de un Libro	CORRECTO	FINALIZADA
025_BBDD	Modifica los datos de u Proveedor en BBDD, con la informacion modificada en los textField, con el evento del boton modificar	realiza la actualizacion de datos en BBDD de los datos de un Proveedor	CORRECTO	FINALIZADA
026_BBDD	Elimina un socio en BBDD, con el evento del boton eliminar	realiza la eliminacion de un socio en BBDD, eliminando el objeto como tal de la base de datos	CORRECTO	FINALIZADA
027_BBDD	Elimina un Libro en BBDD, con el evento del boton eliminar	realiza la eliminacion de un Libro en BBDD, eliminando el objeto como tal de la base de datos	CORRECTO	FINALIZADA
028_BBDD	Elimina un Proveedor en BBDD, con el evento del boton eliminar	realiza la eliminacion de un Proveedor en BBDD, eliminando el objeto como tal de la base de datos	CORRECTO	FINALIZADA



ID PRUEBA	DESCRIPCION	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO PRUEBA
029_BBDD	Inserta nuevas reservas de libros de un socio en concreto y establece la fecha de reserva y lo guarda en BBDD	realiza la reserva de un Libro en concreto y de un socio en concreto, indicando la fecha del día de su reserva	CORRECTO	FINALIZADA
028_BBDD	Inserta devoluciones de libros de un socio en concreto y establece la fecha de devolución y lo guarda en BBDD	realiza la devolución de un Libro en concreto y de un socio en concreto, indicando la fecha del día de su reserva	CORRECTO	FINALIZADA
030_BBDD	Muestra todos los libros reservados de un Socio en concreto	Los muestra correctamente en una tabla, incluyendo id reserva, id Libro, id Socio y fecha reserva	CORRECTO	FINALIZADA

## CAPITULO 7. CONCLUSIÓN.

### Conclusión

Una vez completado este proyecto, he podido hacer uso de muchos de los conocimientos aprendidos durante el ciclo formativo, tales como el desarrollo de interfaces, haber profundizado en base de datos con Mysql y Hibernate y programación orientado a Objetos puro y duro.

He conseguido manejar bastante bien la programación orientada a objetos con la utilización de métodos para el control de las ventanas internas de mi programa y además mejorar el uso de matrices y vectores para almacenar la información.

Cabe destacar, que con dicho proyecto he conseguido estructurar en mi cabeza la estructura de este programa, utilizando diferentes clases y métodos, que anteriormente no conseguía hacer en ejercicios de clase.

Como dificultad tengo que destacar el uso de tablas dinámicas a la hora de manejar la información, como la introducción y extracción de los datos de las celdas con eventos.

Finalmente, mencionar que con la creación de este programa en Java, he conseguido utilizar principalmente, conocimientos adquiridos en las siguientes asignaturas del ciclo:

Desarrollo de Interfaces, Bases de Datos y Programación orientada a Objetos

Así mismo, a pesar de las mejoras indicadas en el capítulo 7, considero que la realización de este sistema es satisfactoria para mí.



## **CAPITULO 8. FUTURAS MEJORAS.**

### **Futuras Mejoras**

En un futuro a dicha aplicación se podría implementar un sistema que identifique el tiempo que un socio puede tener un libro reservado y en qué fecha exacta tendrá que tramitar su devolución. Es decir, tener un control de las devoluciones de los socios, para que cada libro tenga un tiempo determinado para poder reservarse.

También como mejoras, incluiremos mejor diseño en los eventos de visualización de los datos de las tablas, con solo pulsar la fila seleccionada, directamente la información de la misma se muestre en las cajas de texto, ya que actualmente se hace a través de la presión de un botón.

Mejorar el apartado de ayuda para el usuario, para que pueda acceder al manual o información de ayuda a través de la red y no en local como esta actualmente.

Por último, también incluiríamos, el acceso al sistema a través de un login previo, donde el empleado tenga que identificarse y en función del empleado que acceda, conceder una serie de permisos en la aplicación.



## **CAPITULO 9. BIBLIOGRAFÍA**

### **Bibliografía**

*Página Community Hibernate*

<http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/queryhql.html>

*Página Curso de Hibernate*

<http://cursohibernate.es/doku.php>

*Página de Api de Java*

<http://docs.oracle.com/javase/7/docs/api/>

*Página de Programación Orientada a Objetos*

<http://www.monografias.com/trabajos/objetos/objetos.shtml>

Apuntes de Hibernate. Unidad 3. Acceso a datos

[Documento PDF](#)

