INDICE

CAPÍTULO I 3

1.1 ANÁLISIS DE SITUACIÓN ACTUAL 3
1.1.1 ANTECEDENTES: 3
1.2 REQUERIMIENTOS Y SOFTWARE A UTILIZAR: 3
1.2.1 REQUERIMIENTOS: 3
1.2.2 SOFTWARE A UTILIZAR: 3
1.3 DIAGRAMAS DE PROCESOS: 4
1.3.1 MÓDULO DE LIBROS 5
1.3.2 MÓDULO DE MAPAS 14
1.3.3 MÓDULO DE PRESTAMOS 19
1.3.4 MÓDULO DE USUARIOS DEL SISTEMA 25
1.4 DIAGRAMAS DE CASOS DE USO 32
1.4.1 DIAGRAMA GENERAL 32
1.4.2 INGRESO AL SISTEMA 33
1.4.3 ADMINISTRACIÓN DE LIBROS Y EJEMPLARES 34
1.4.4 ADMINISTRACIÓN DE MAPAS 38
1.4.5 PRESTAMO DE LIBROS 40
1.4.6 ADMINISTRACIÓN DE USUARIOS 44
1.5 DIAGRAMA DE CLASES 46
1.6 DIAGRAMAS DE SECUENCIA DEL MÓDULO DE PRESTAMOS 48
1.6.1 PRESTAR LIBRO 48
1.6.2 DEVOLUCIÓN DE LIBRO 49
1.6.3 LIBROS PRESTADOS 50

CAPÍTULO II 51

1.1 MODELO DE DATOS 51
1.1.1 MODELO CONCEPTUAL: 51
1.2 MODELO FISICO 52
1.3 METODOLOGIA Y HERRAMIENTAS 53
1.3.1 METODOLOGIA DE DISEÑO Y DESARROLLO 53
1.3.2 HERRAMIENTAS DE DESARROLLO 58

CAPÍTULO V 60

1.1 CONCLUSIONES 60
1.2 RECOMENDACIONES 61

ANEXOS 63

CAPÍTULO I

1.1 Análisis de Situación Actual

1.1.1 Antecedentes:

El Instituto Tecnologico Bolivar esta ubicado al norte de la ciudad de Tulcán capital de la provincia del Carchi. La institución tiene cerca de 1500 estudiantes y cuenta con una Biblioteca con mas de 4000 ejemplares.

La Biblioteca ofrece el prestamo de libros dentro de la institución a todo tipo de publico, estudiantes, docentes pertenecientes a la institución y tambíen a estudiantes de otras instituciones.

1.2 Requerimientos y Software a utilizar:

1.2.1 Requerimientos:

La Biblioteca del Instituto Tecnologico Bolivar necesita realizar un control sobre el ingreso de los libros ya que cuenta con mas de 4000 ejemplares, La biblioteca no cuenta con un inventario consolidado de todos sus libros, el prestamo de los ejemplares se lleva de forma manual, el cual permite hacer un seguimiento para verificar los libros que estan prestados y conocer el nombre de la persona que lo solicito para resolver posibles inconvenientes que se presenten.
Se requiere entonces realizar un sistema que automatice el proceso de registro de libros y el proceso de prestamo de ejemplares.

1.2.2 Software a utilizar:

Para el desarrollo de este sistema se recomienda el uso de "Software Libre" término que es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en libre como en "libertad de expresión", no como en la recepción de alguna promoción o gratuidad.
El Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).

- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Siendo éstas las características que se buscan en el software y también analizando ciertos parámetros que se deben considerar el momento de implementar una aplicación como:

- La naturaleza del negocio, en este caso la Biblioteca es una entidad sin fines de lucro.
- Volumen de información: el volumen de información no es extremadamente elevada y encaja en los manejados por MySQL.
- Propósitos y niveles de seguridad.

Por todo esto se sugiere PHP[1] como lenguaje de programación y MySQL[2] como gestor bases de datos.

1.3 Diagramas de procesos:

A continuación se muestra cada uno de los procesos que se ejecutan en la Bibilioteca, los actores de los mismos y sus respectivas particularidades:

[1] Hypertext Preprocessor: http://www.php.net/
[2] http://www.mysql.com/

1.3.1 MÓDULO DE LIBROS

Numero de Definición: BIB01	**Fecha**: 15-Feb-2010

1. Objetivo

Permitir la administracion del ingreso de los Libros y Ejemplares que hacen parte de la Biblioteca.

2. Observaciones

- La Biblioteca cuenta con un metodo de organización basado en el código Dewey.
- La Biblioteca desea registrar la ubicación donde se encuentra cada uno de los ejemplares de cada libro.
- La Biblioteca desea registrar los autores y categorias de cada uno de los libros para facilitar su busqueda.
- La Biblioteca desea registrar el precio de los libros para realizar un reporte de gastos mensual.

3. Responsables

Nombre	Firma	Nombre	Firma
Rosaura Burbano			
Santiago Galvis			

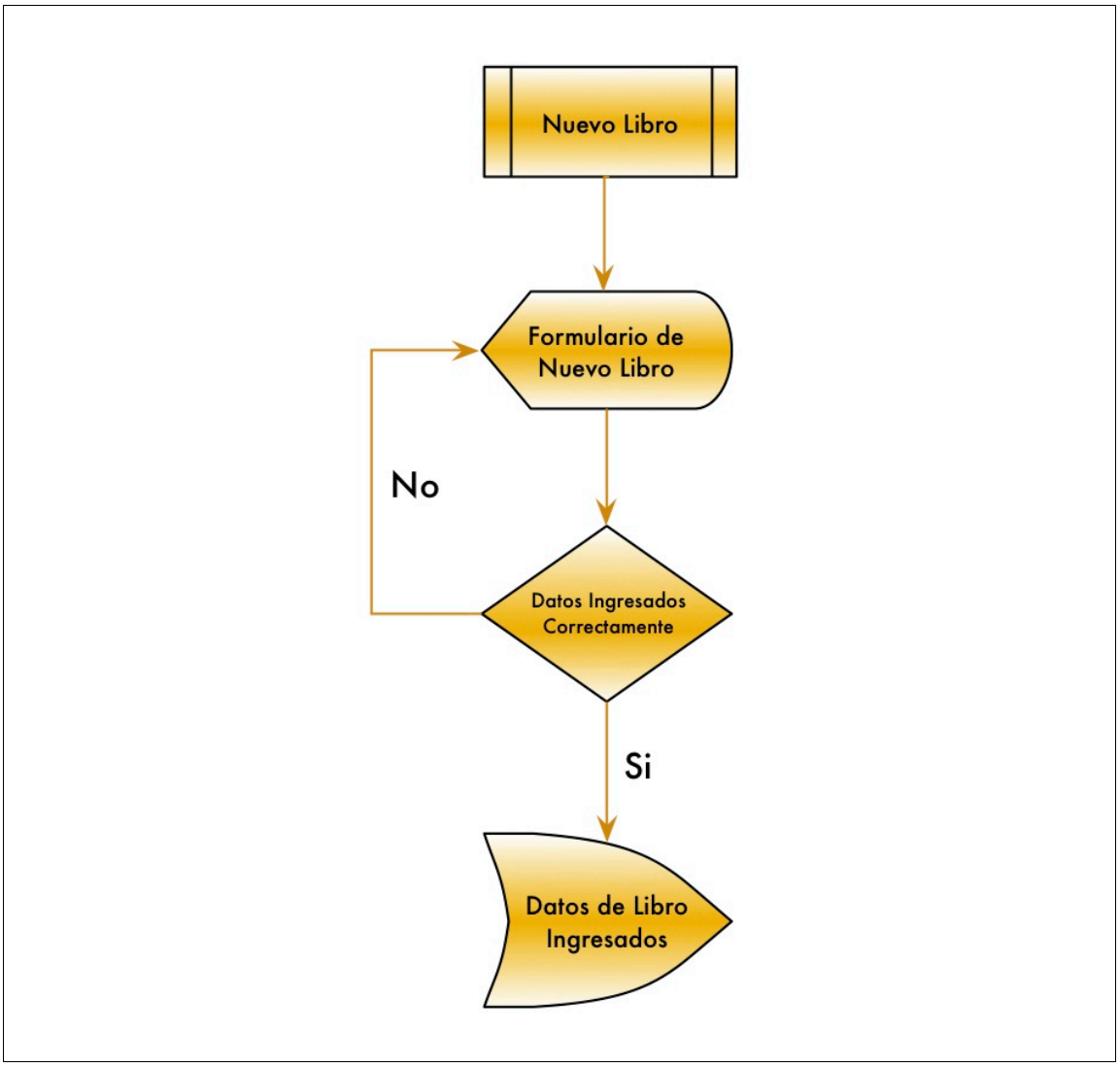
4. Definición de Procesos

Los procesos que se manejan en el módulo de registro de libros son:

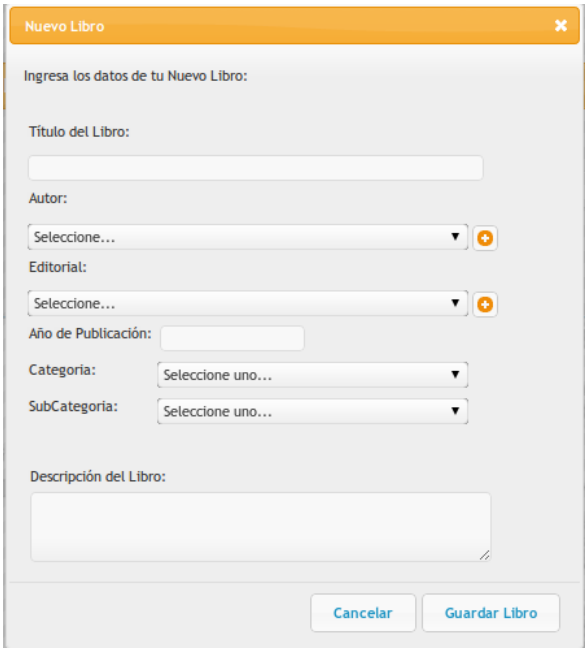
- Nuevo Libro
- Actualizar Libro.
- Nuevo Ejemplar
- Actualizar Ejemplar.

4.1. Nuevo Libro: Este proceso permite ingresar los datos de un libro nuevo que pertenezca a la Biblioteca. Los datos necesarios son:

- Titulo de Libro
- Autor (Se pueden registrar maximo 5 autores por libro).
- Editorial
- Año de Publicación
- Categoria y Subcategoria
- Descripción del Libro

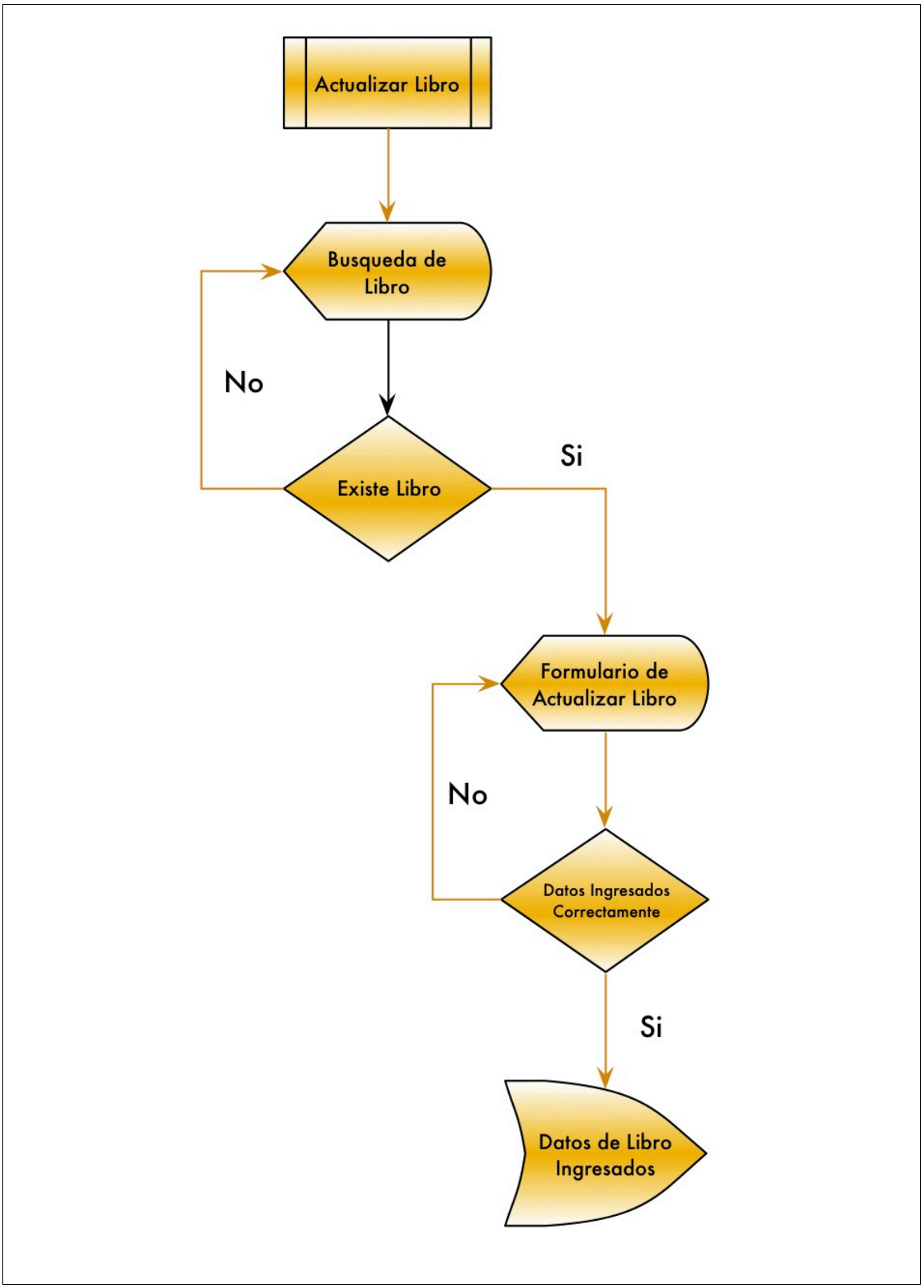

Nuevo Libro
Formulario de Nuevo Libro
No
Datos Ingresados Correctamente
Si
Datos de Libro Ingresados

Formulario de ingreso de nuevo Libro:

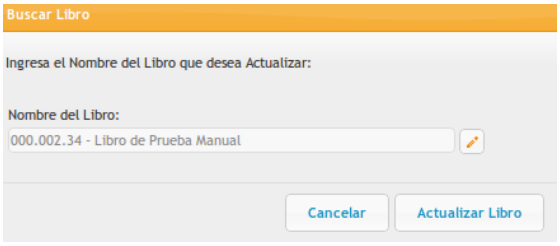


4.2. Actualización de Libro: Este proceso permite modificar la información de un libro cuando ya ha sido ingresado anteriormente :

- Los cambio del libro se deben reflejar en todos los ejemplares del mismo
- Se pueden modificar todos los datos de un libro.


Actualizar Libro
Busqueda de Libro
No
Existe Libro
Si
Formulario de Actualizar Libro
No
Datos Ingresados Correctamente
Si
Datos de Libro Ingresados

Pantalla de Busqueda de Libro:

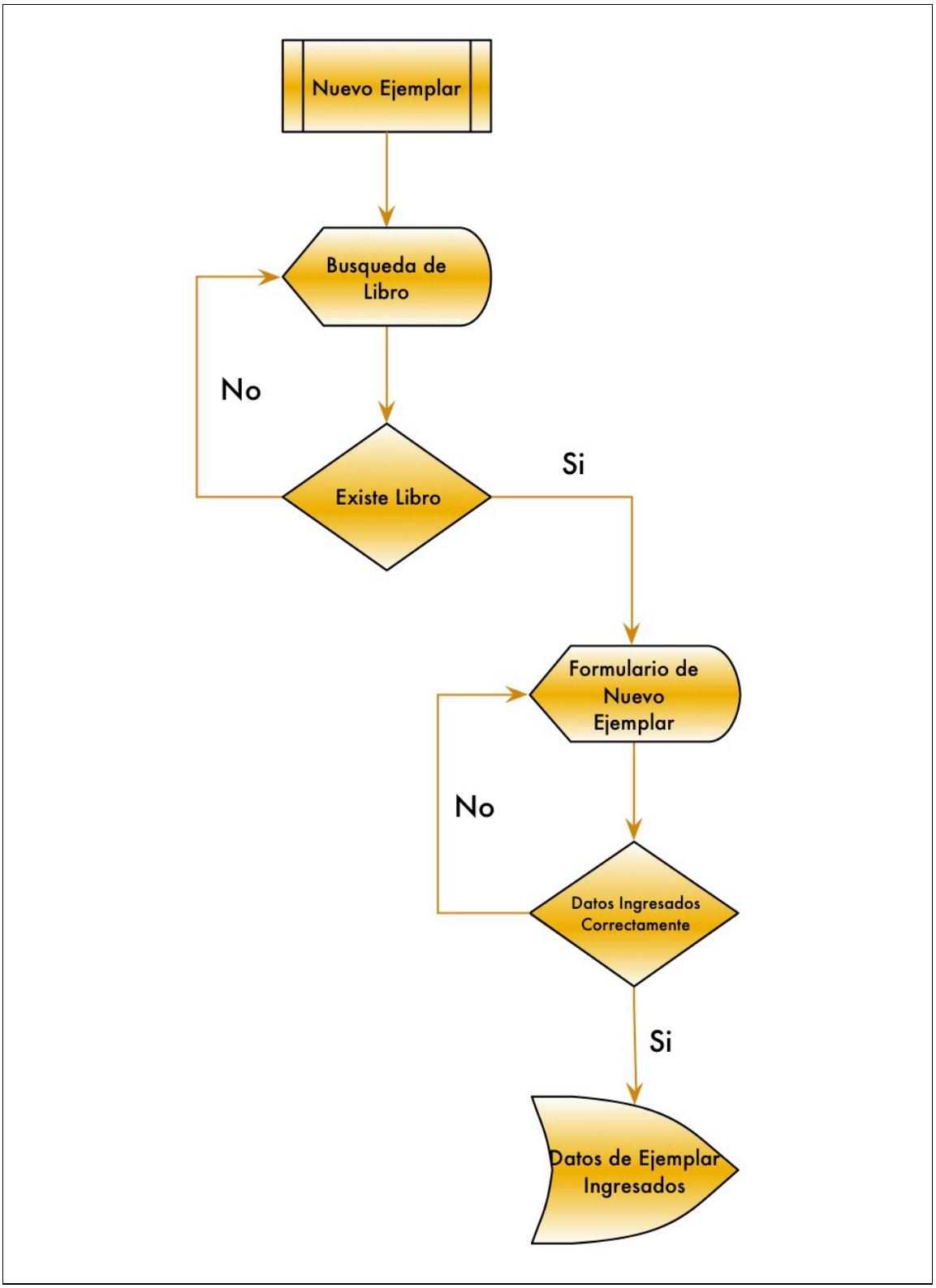


Formulario de Actualización de Libro:

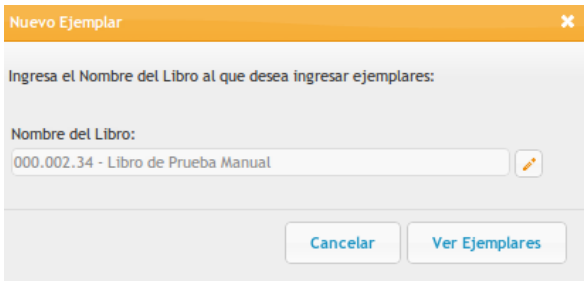


4.3. Nuevo Ejemplar: Este proceso permite ingresar un numero n de ejemplares de un libro que haya sido ingresado anteriormente. Los datos nesesarios son:

- Codigo
- # de Inventario
- Estado en que se ecuentra el libro (nuevo, usado, deteriorado, retirado, etc)
- Ubicación del Ejemplar
- Precio
- Observaciones (En prestamo, Disponible)
- Disponibilidad


Nuevo Ejemplar
Busqueda de Libro
No
Existe Libro
Si
Formulario de Nuevo Ejemplar
No
Datos Ingresados Correctamente
Si
Datos de Ejemplar Ingresados

Pantalla de busqueda del libro al que pertenece el ejemplar:

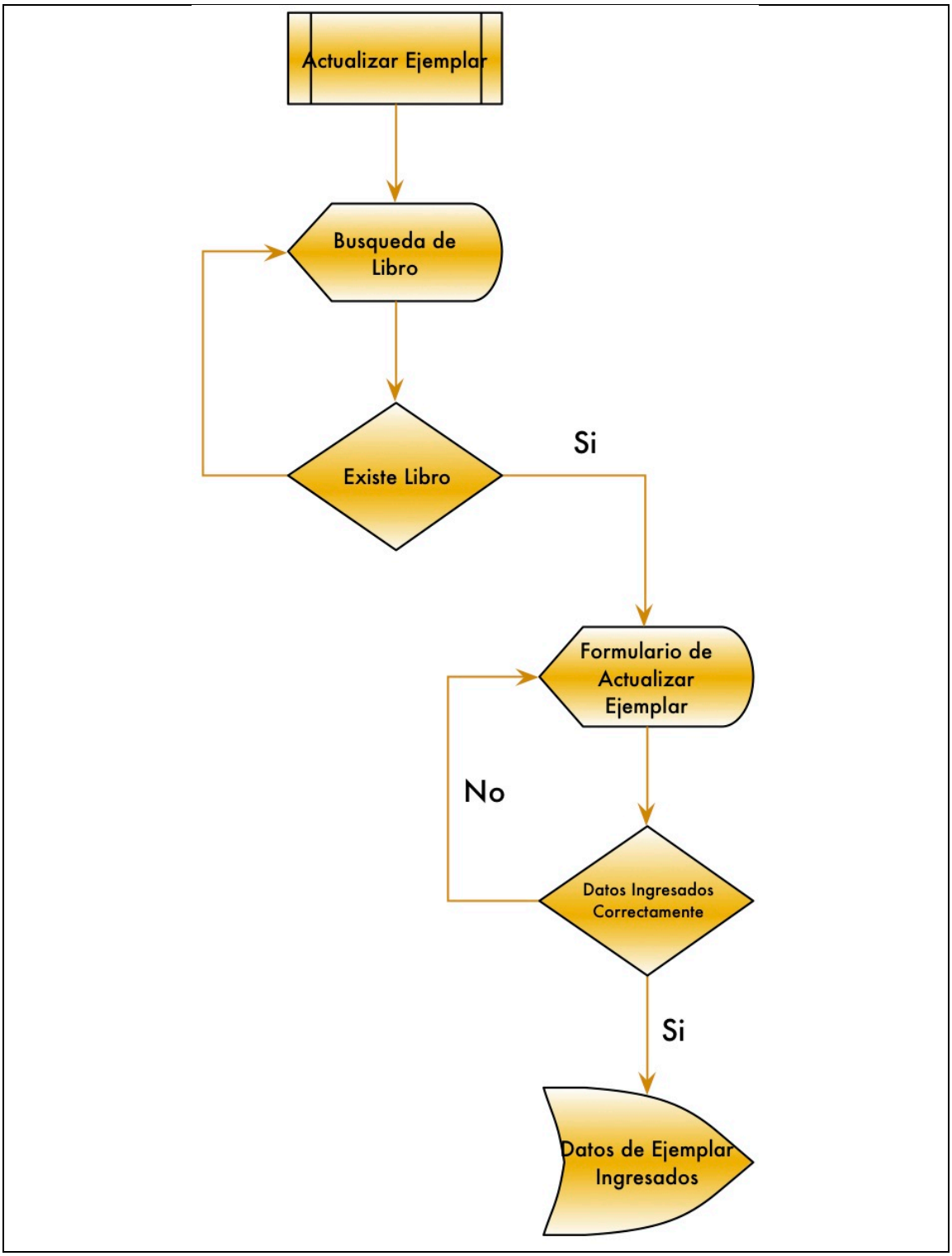


Formulario de Ingreso de Ejemplares:

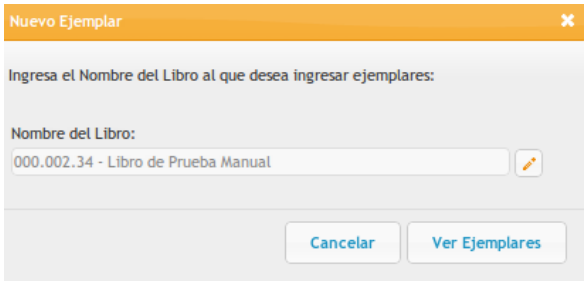


4.2. Actualización de Ejemplar: Este proceso permite modificar los ejemplares que pertenecen a un libro

- Se pueden cambiar todos los datos del ejemplar a escepcion del código.
- Se puede eliminar el ejemplar de un libro.


Actualizar Ejemplar
Busqueda de Libro
Existe Libro
Si
Formulario de Actualizar Ejemplar
No
Datos Ingresados Correctamente
Si
Datos de Ejemplar Ingresados

Pantalla de busqueda del libro al que pertenece el ejemplar a actualizar:



Formulario de Actualización de Ejemplares:



Seguridades: Seguridad establecida nivel de módulo

1.3.2 MÓDULO DE MAPAS

Numero de Definición: BIB02	**Fecha**: 10-Jul-2008

1. Objetivo

Permitir la administración de la mapoteca con la cual cuenta la Biblioteca.

2. Observaciones

- Los mapas cuentan con un codigo unico y son de uso exclusivo de la institución.

3. Responsables

Nombre	Firma	Nombre	Firma
Rosaura Burbano			
Santiago Galvis			

4. Definición de Procesos

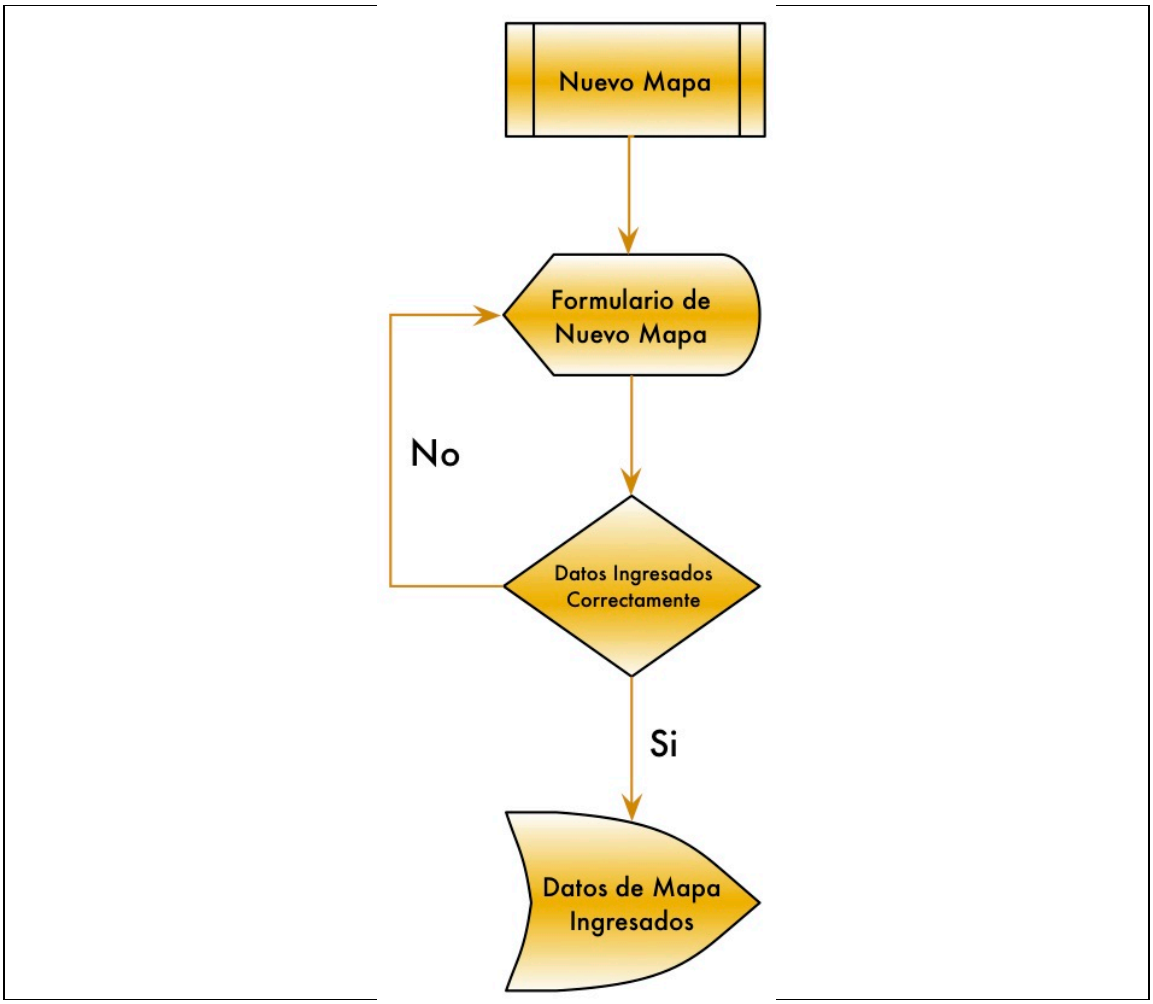
Los procesos que se manejan en el módulo de lecturas son:

- Nuevo Mapa
- Actualizar Mapa

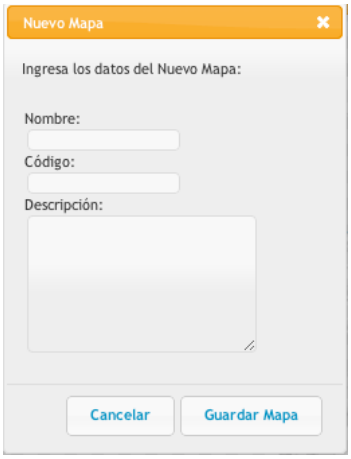
4.1. Nuevo Mapa: Este proceso permite registrar toda la información correspondiente a los mapas que pertenecene a la Biblioteca.

Los datos que se necesitan son:

- Codigo
- Nombre
- Descripción

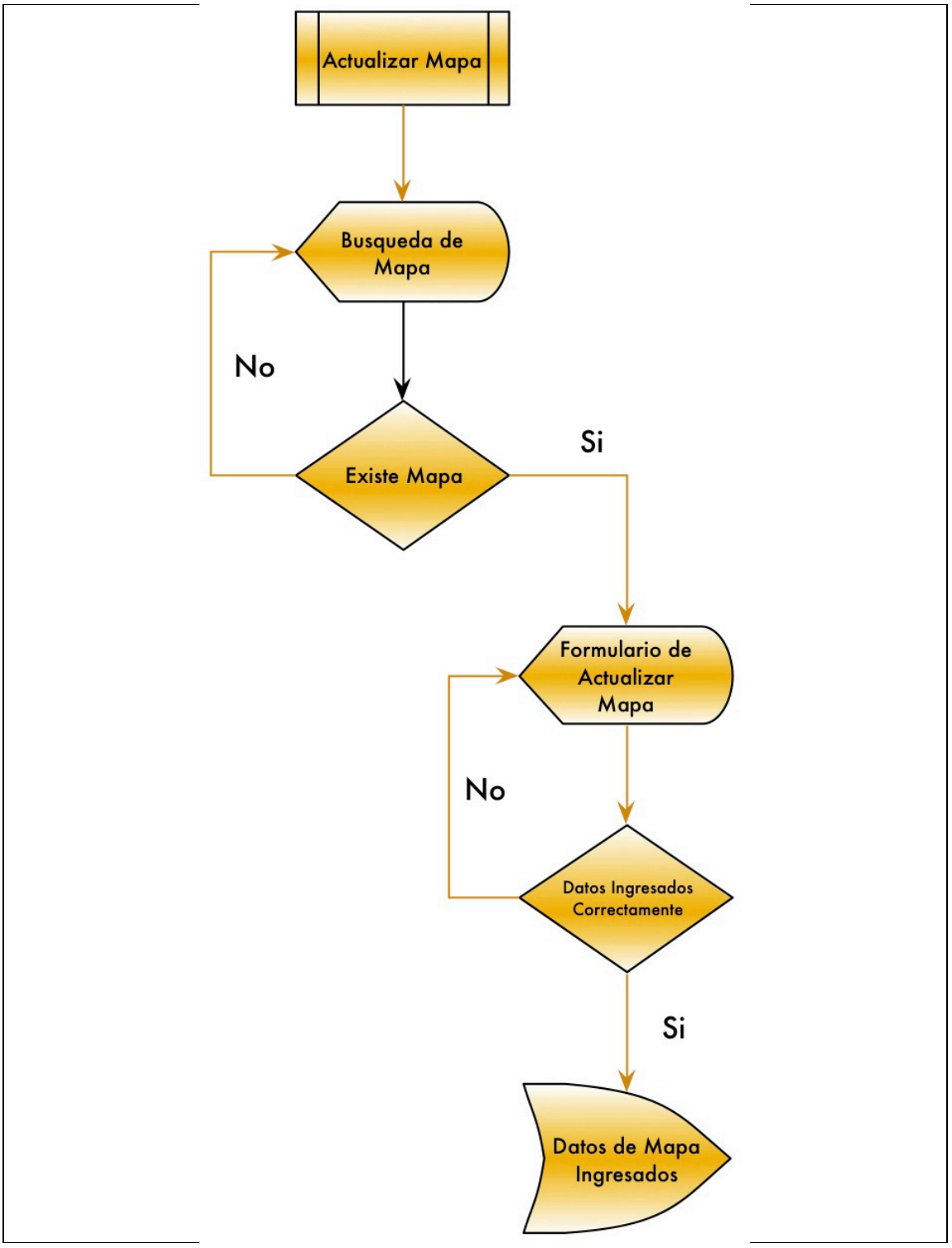

Nuevo Mapa
Formulario de Nuevo Mapa
No
Datos Ingresados Correctamente
Si
Datos de Mapa Ingresados

Formulario de registro de nuevo Mapa:

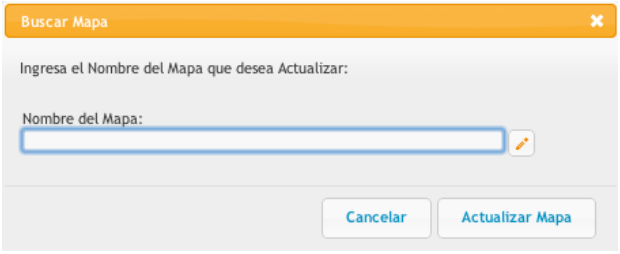


4.2. Actualizar Mapa: Esta opción debe permitir al usuario modificar los datos de una mapa anteriormente ingresado.

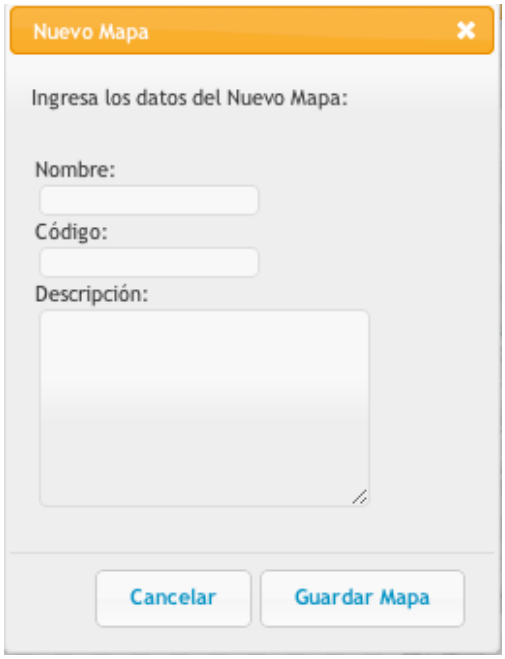
- Se pueden cambiar todos los datos ingresados de una mapa


Actualizar Mapa
Busqueda de Mapa
No
Existe Mapa
Si
Formulario de Actualizar Mapa
No
Datos Ingresados Correctamente
Si
Datos de Mapa Ingresados

Pantalla de búsqueda de Mapa:



Formulario de modificación de Mapa:



Seguridades: Seguridad establecida nivel de módulo

1.3.3 MÓDULO DE PRESTAMOS

Numero de Definición: BIB03	**Fecha**: 10-Jul-2008

1. Objetivo

Permitir realizar el prestamo de libros a los usuarios de la biblioteca de manera que exista un seguimiento de cada uno de los ejemplares y se pueda conocer de forma agil y rapida los ejemplares que se encuentran prestados y a que usuario se le realizo el prestamo.

2. Observaciones

- Los usuarios que deseen solicitar un ejemplar deben presentar un documento de identificación de preferencia la cédula de ciudadania o el carné de la institución a la que pertenece.
- Los usuario no pueden sacar los libros fuera de las instalaciones de la biblioteca y lo deben entregar antes que finalice el día.

3. Responsables

Nombre	Firma	Nombre	Firma
Rosaura Burbano			
Usuarios de la biblioteca			
Santiago Galvis			

4. Definición de Procesos

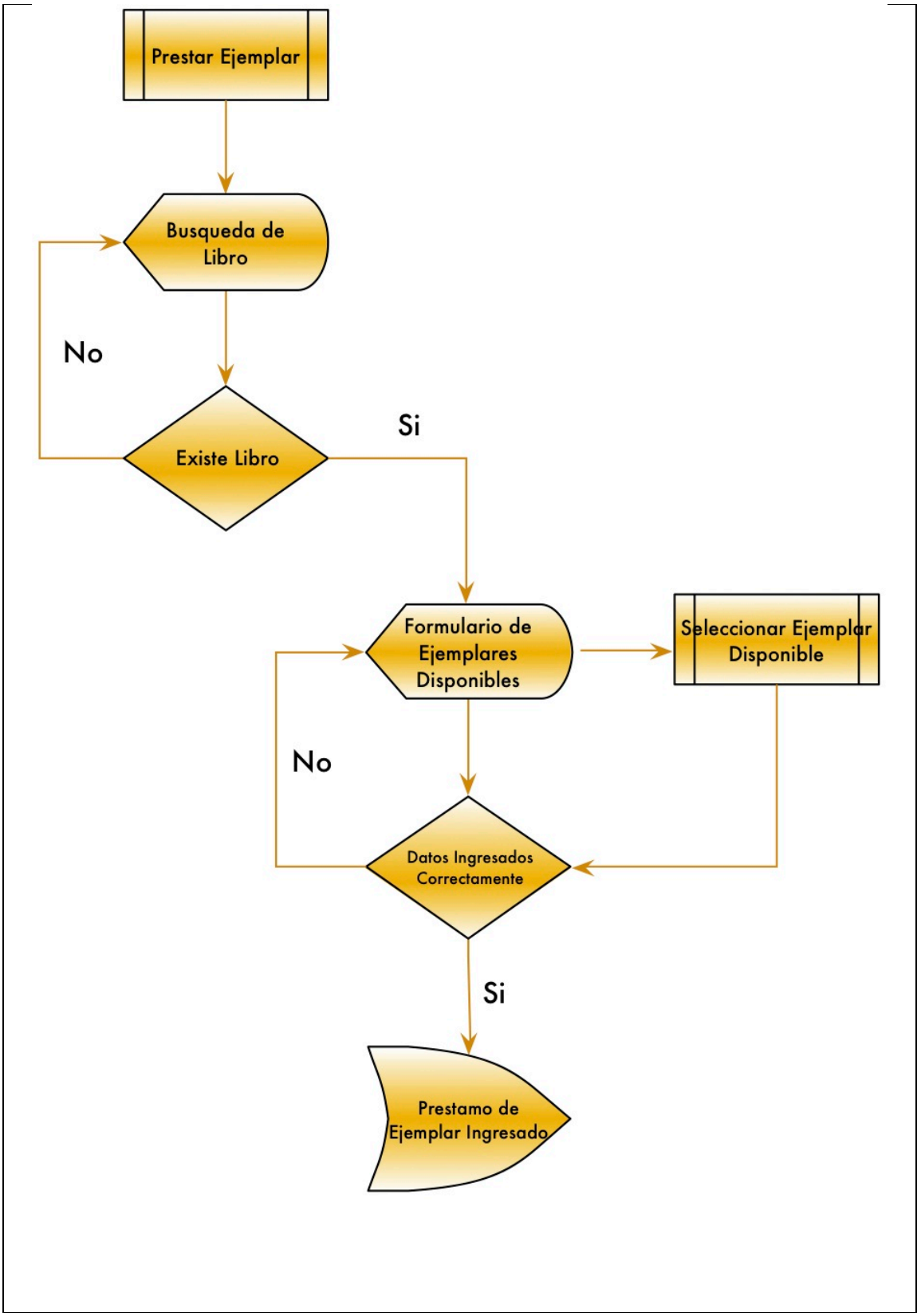
Los procesos que se manejan en el módulo de Prestamos son:

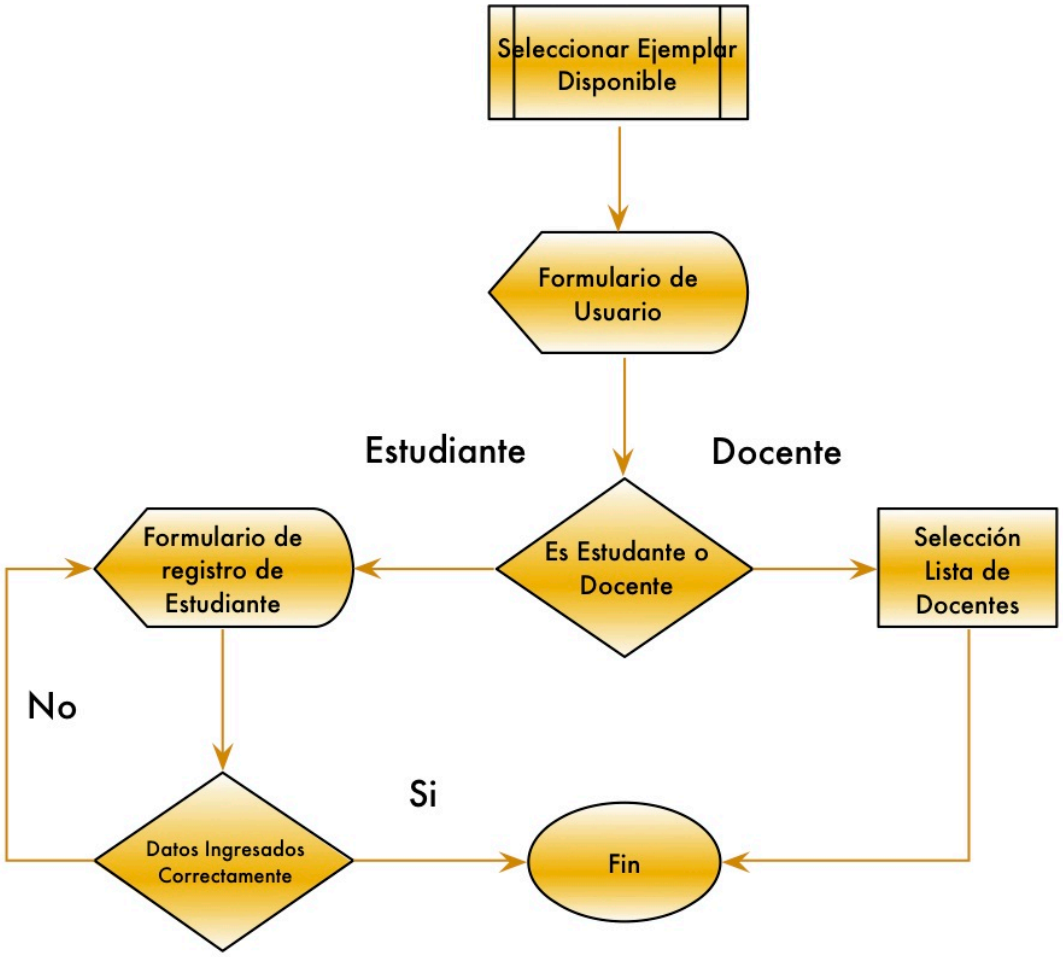
- Prestar Ejemplar
- Devolución de Ejemplar

4.1. Prestar Ejemplar: Este proceso permite registrar el prestamo de un ejemplar a un usuario que lo solicite. Se deberá escoger un ejemplar disponible y se podrá registrar la información del usuario o seleccionarlo si es usuario ya existente para proceder finalmente con la entrega del ejemplar.

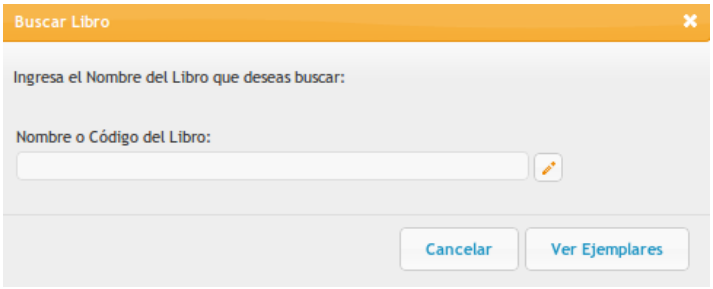
Los datos que se necesitan son:

- Nombre del Libro
- Cédula o Carne del usuario
- Nombre Completo del usuario
- Curso o Nivel (Opcional)

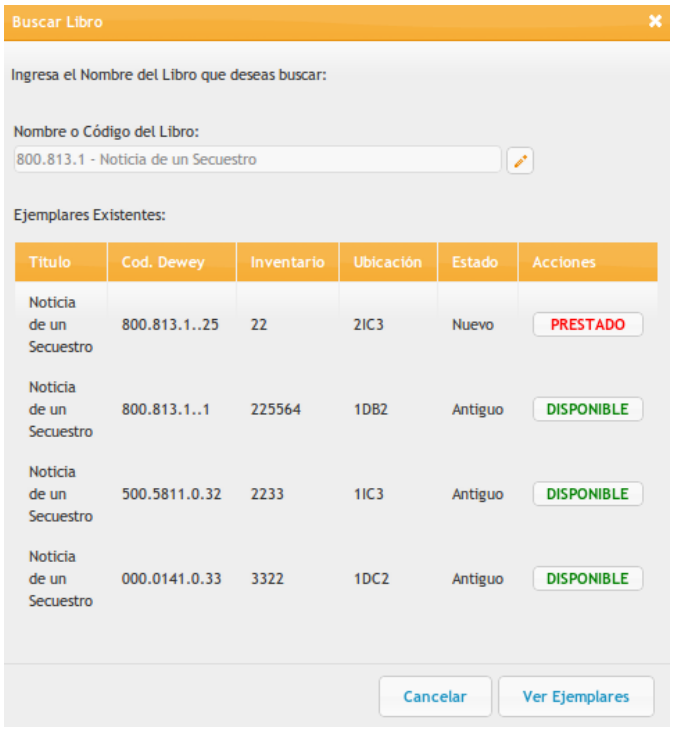

Prestar Ejemplar
Busqueda de Libro
No
Existe Libro
Si
Formulario de Ejemplares Disponibles
Seleccionar Ejemplar Disponible
No
Datos Ingresados Correctamente
Si
Prestamo de Ejemplar Ingresado



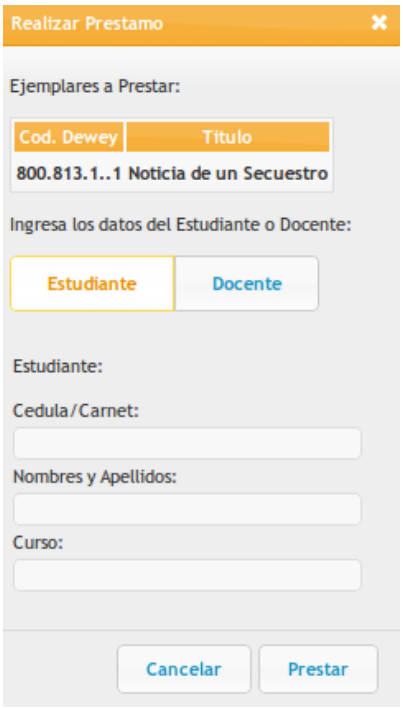
Pantalla de Busqueda del Libro:



Pantalla de Selección de Ejemplar disponible:



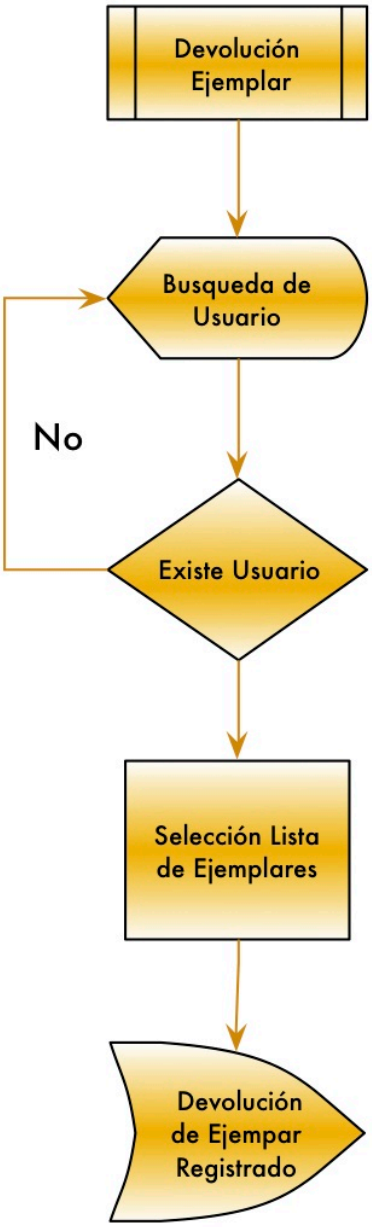
Formulario de Prestamo de Libro:



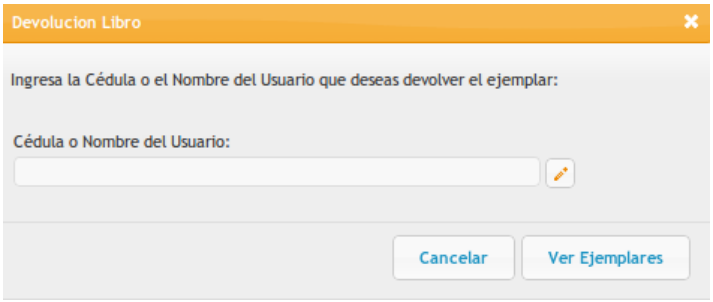
4.2. Devolución de Ejemplar: Este proceso permite registrar la entrega de un libro porparte del usuario a la Biblioteca. Se debe solicitar el numero de identificacion del usuario que solicito el libro en prestamo y verificar que el codigo del libro que se recibe sea el mismo codigo del libro que presto.

Los datos que se necesitan son:

- Codigo de Usuario.



Pantalla de busqueda de usuario:



Formulario de devolución de libro:



Seguridades: Seguridad establecida nivel de módulo

1.3.4 MÓDULO DE USUARIOS DEL SISTEMA

Numero de Definición: BIB04	**Fecha**: 10-Jul-2008

1. Objetivo

Permitir la administración de los Usuarios de la biblioteca, entre eso se encuentra la inscripción de un nuevo Usuario y actualización de datos.

2. Observaciones

- El Sistema a realizarse debe tener una interfaz de ingreso al Sistema en la cual se debe exigir una contraseña para identificar al usuario y que pueda acceder a los módulos de su perfil asociado.

- Este modulo esta relacionado con el prestamo y devolución de libros.

3. Responsables

Nombre	Firma	Nombre	Firma
Rosaura Burbano			
Santiago Galvis			

4. Definición de Procesos

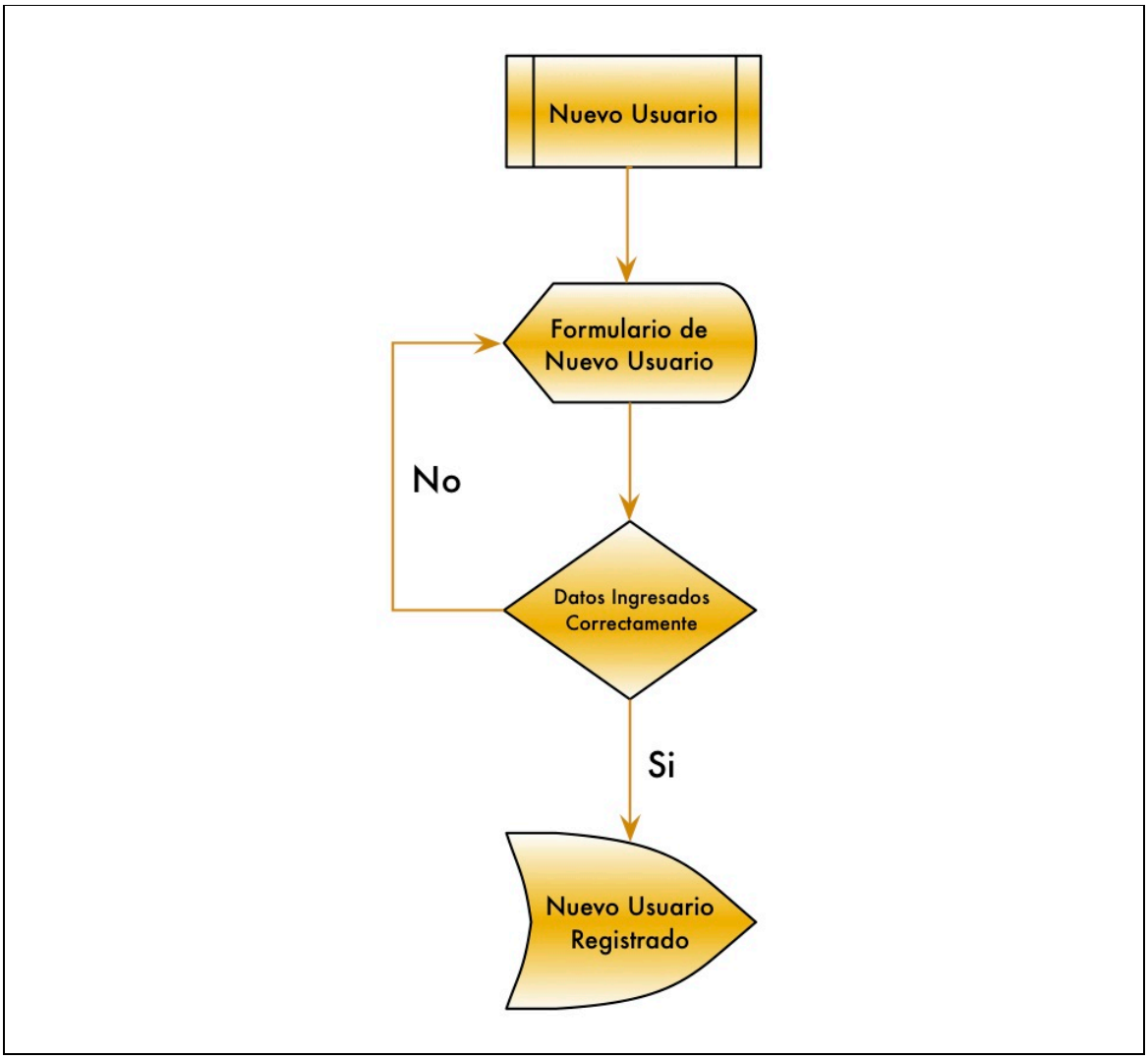
Los procesos que se manejan en el módulo de usuarios son:

- Nuevo Usuario.
- Modificar de Usuario.

4.1. Ingresar Nuevo Usuario: Este proceso nos permite la creación de un nuevo usuario para el ingreso al sistema.

Los datos que se necesitan son:

- Carnet o Cédula.
- Nombres y Apellidos.
- Teléfono.
- Dirección
- Perfil (Estudiante, Bibliotecario y Administrador).


Nuevo Usuario
Formulario de Nuevo Usuario
No
Datos Ingresados Correctamente
Si
Nuevo Usuario Registrado

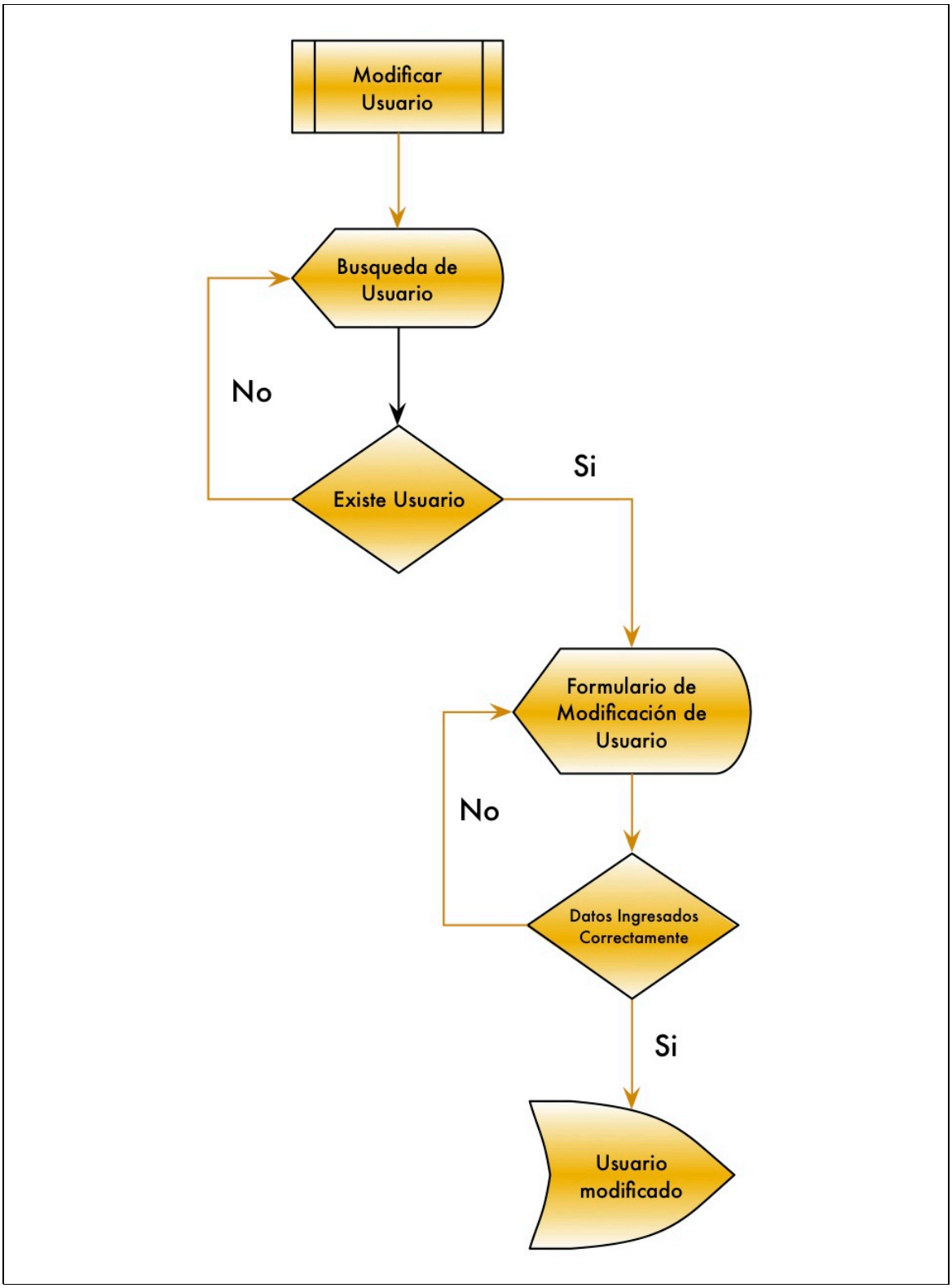
Formulario de registro de nuevo usuario del sistema:



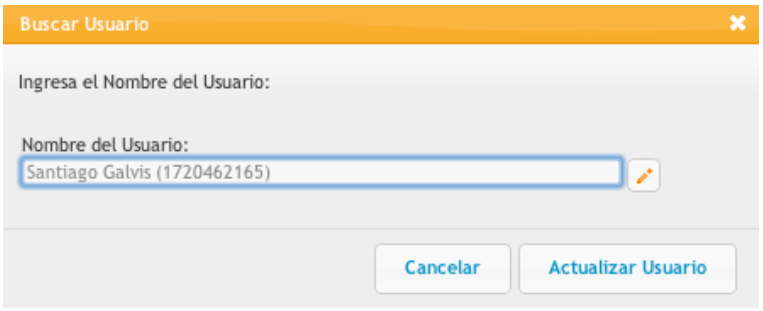
4.2. Modificar Usuario: Este proceso nos permite modificar los datos de un usuario del sistema.

Los datos que se necesitan son:

- Nombre del usuario.
- Contraseña.
- Perfil.


Modificar Usuario
Busqueda de Usuario
No
Existe Usuario
Si
Formulario de Modificación de Usuario
No
Datos Ingresados Correctamente
Si
Usuario modificado

Formulario de selección de usuario del sistema a modificar:



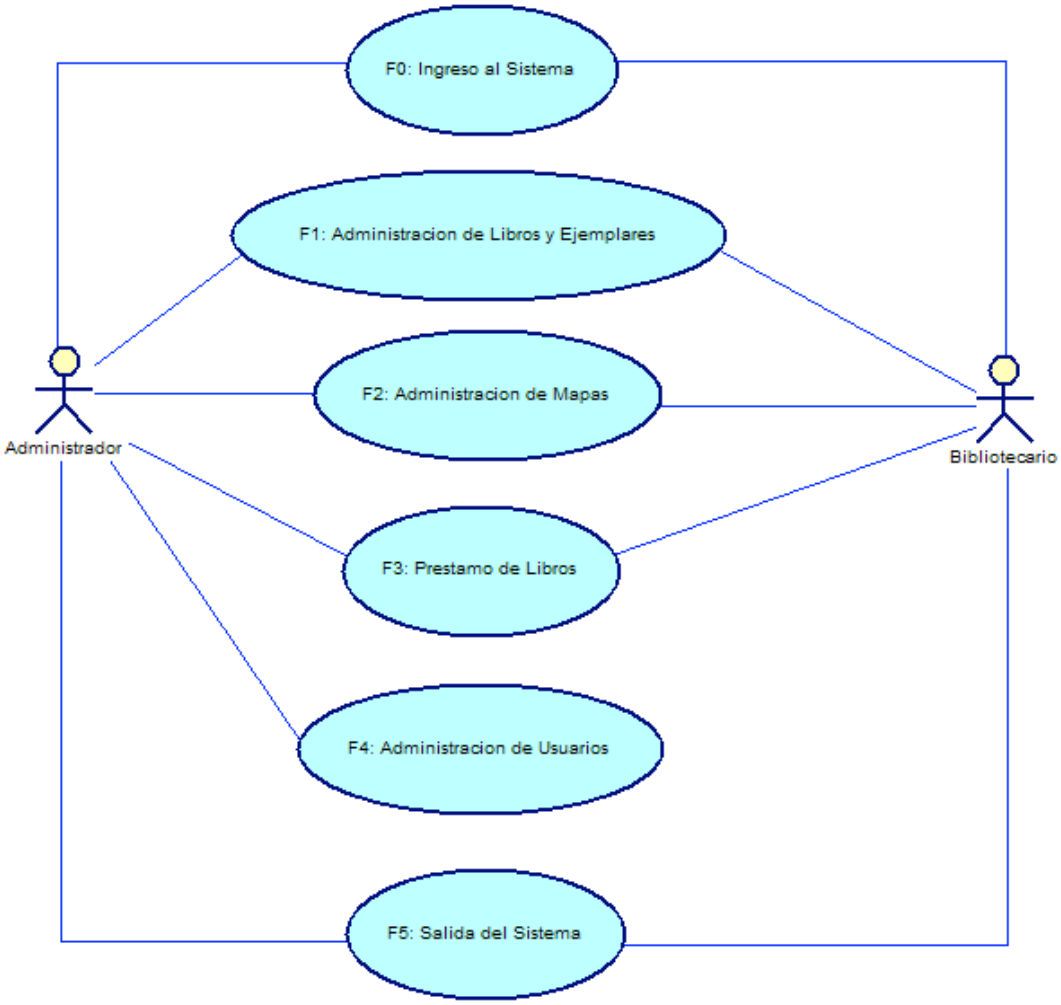
Formulario de modificación de usuario del sistema:



Seguridades: Seguridad establecida nivel de módulo

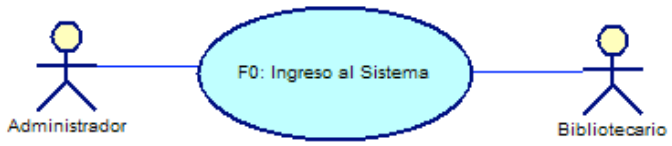
1.4 Diagramas de Casos de Uso

1.4.1 Diagrama General



1.4.2 Ingreso al Sistema

Caso de Uso: Ingreso al Sistema
Identificador: F0



Descripción: Este caso de uso sirve para la validación de usuarios del sistema y para el acceso a los módulos del mismo. Este caso de uso será utilizada por todos los actores del sistema.

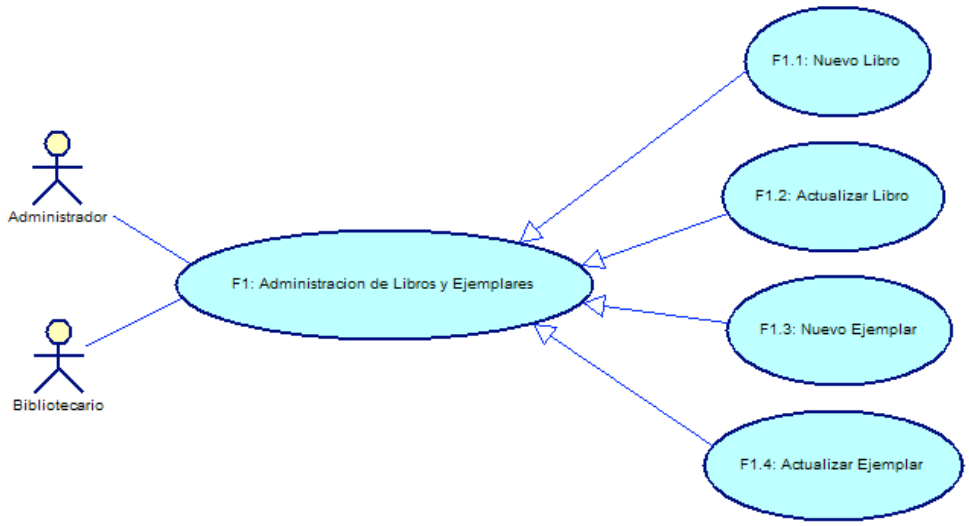
Flujo:

Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Inicia la aplicación web	2	Presenta el formulario de ingreso al sistema	
		3	Carga usuarios del sistema	
4	Ingresa los datos de validación de ingreso	5	Verifica los datos	E1
		6	Presenta página principal del sistema	

Excepciones:

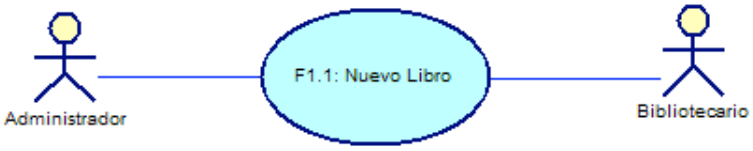
id	*Nombre*	*Acción*
E1	Contraseña Incorrecta	Emite un mensaje de error sobre la contraseña inválida.

1.4.3 Administración de Libros y Ejemplares



Caso de Uso: Nuevo Libro

Identificador: F1.1



Descripción: Este caso de uso sirve para el registro de un nuevo libro. Esta opción solo podrá hacerla el administrador y el Bibliotecario.

Flujo:

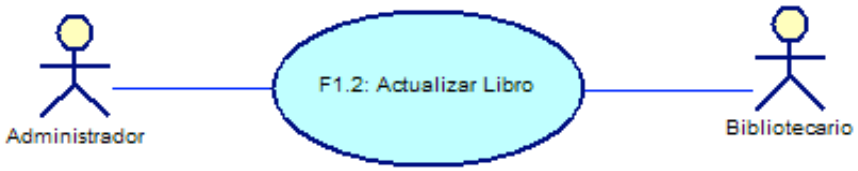
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Nuevo Libro	2	Presenta formulario de Nuevo Libro	
3	Ingreso de datos necesarios			
4	Notifica al sistema que el llenado de información ha sido completado	5	Validación de datos	E1
		6	Almacena información	

Excepciones:

id	*Nombre*	*Acción*
E1	Validación de datos	Emite un mensaje de error sobre el formato de los datos ingresados.

Caso de Uso: Actualizar Libro

Identificador: F1.2



Descripción: Este caso de uso sirve para la modificación de los datos de un Libro registrado. Esta opción solo podrá hacerla el administrador y el Bibliotecario.

Flujo:

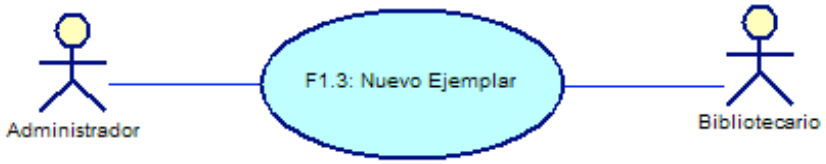
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Actualizar Libro	2	Presenta formulario de busqueda de libro	
		3	Carga libros registrados	E1
4	Ingreso de datos de busqueda	5	Presenta formulario de actualizar libro	E2
6	Modifica datos del libro			
7	Notifica al sistema que el llenado de información ha sido completado	8	Validación de Datos	E3
		9	Alamcena Información	

Excepciones:

id	*Nombre*	*Acción*
E1	No hay información	No existen Libros Registrados
E2	Libro no encontrado	Emite mensaje de error de no existencia del libro
E3	Validación de datos	Emite error de datos incorrectos

Caso de Uso: Nuevo Ejemplar

Identificador: F1.3



Descripción: Este caso de uso sirve para el registro de un nuevo ejemplar perteneciente a un libro. Esta opción solo podrá hacerla el administrador y el Bibliotecario

Flujo:

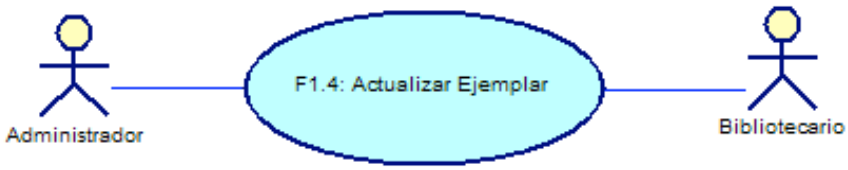
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Nuevo Ejemplar	2	Presenta formulario de busqueda de libro	
		3	Carga libros registrados	E1
4	Ingreso de datos de busqueda	5	Presenta formulario de Nuevo Ejemplar	E2
6	Ingreso de datos necesarios			
7	Notifica al sistema que el llenado de información ha sido completado	8	Validación de Datos	E3
		9	Alamcena Información	

Excepciones:

id	*Nombre*	*Acción*
E1	No hay información	No existen Libros Registrados
E2	Libro no encontrado	Emite mensaje de error de no existencia del libro
E3	Validación de datos	Emite error de datos incorrectos

Caso de Uso: Actualizar Ejemplar

Identificador: F1.4



Descripción: Este caso de uso sirve para la modificación de los datos de los ejemplares de un Libro registrado. Esta opción solo podrá hacerla el administrador y el Bibliotecario.

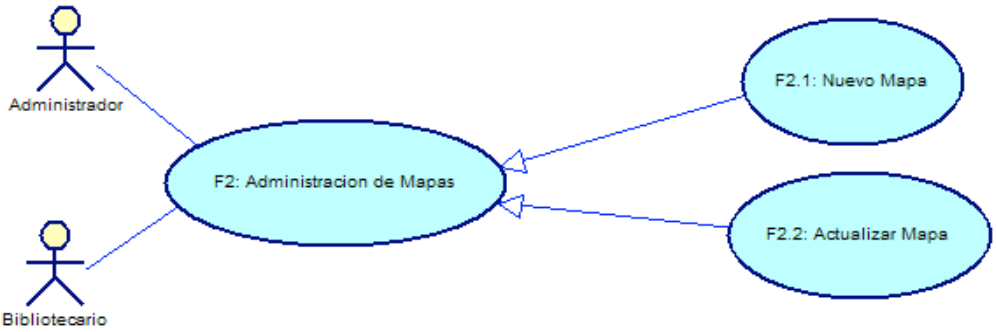
Flujo:

Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Actualizar Ejemplar	2	Presenta formulario de busqueda de libro	
		3	Carga libros registrados	E1
4	Ingreso de datos de busqueda	5	Presenta formulario de Actualizar Ejemplar	E2
6	Ingreso de datos necesarios			
7	Notifica al sistema que el llenado de información ha sido completado	8	Validación de Datos	E3
		9	Alamcena Información	

Excepciones:

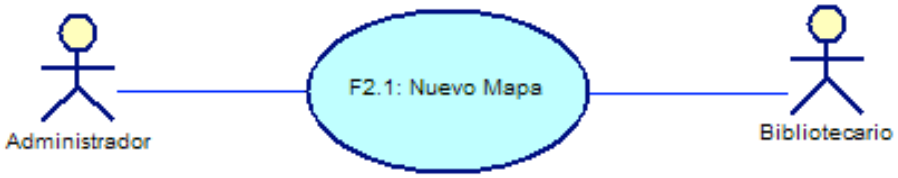
id	*Nombre*	*Acción*
E1	No hay información	No existen Libros Registrados
E2	Libro no encontrado	Emite mensaje de error de no existencia del libro
E3	Validación de datos	Emite error de datos incorrectos

1.4.4 Administración de Mapas



Caso de Uso: Nuevo Mapa

Identificador: F2.1



Descripción: Este caso de uso sirve para el registro de un nuevo mapa. Esta opción podrá hacerla el administrador y el bibliotecario.

Flujo:

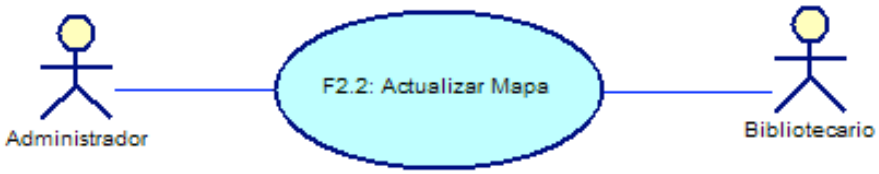
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Nuevo Mapa	2	Presenta formulario de Nuevo Mapa	
3	Ingreso de datos necesarios			
4	Notifica al sistema que el llenado de información ha sido completado	5	Validación de datos	E1
		6	Almacena información	

Excepciones:

id	*Nombre*	*Acción*
E1	Validación de datos	Emite un mensaje de error sobre el formato de los datos ingresados.

Caso de Uso: Actualizar Mapa

Identificador: F2.2



Descripción: Este caso de uso sirve para la modificación de los datos de los mapas registrados. Esta opción solo podrá hacerla el administrador y el Bibliotecario.

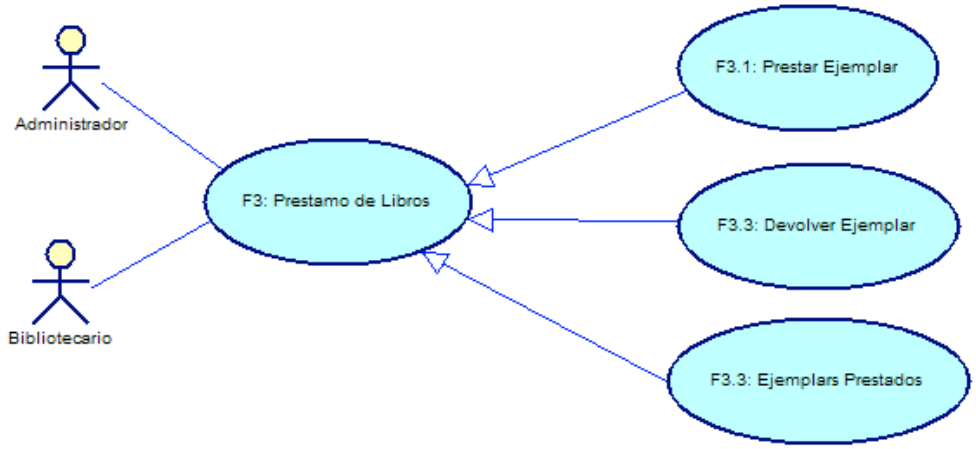
Flujo:

Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Actualizar Mapa	2	Presenta formulario de busqueda de Mapa	
		3	Carga Mapas registrados	E1
4	Ingreso de datos de busqueda	5	Presenta formulario de Actualizar Mapa	E2
6	Ingreso de datos necesarios			
7	Notifica al sistema que el llenado de información ha sido completado	8	Validación de Datos	E3
		9	Alamcena Información	

Excepciones:

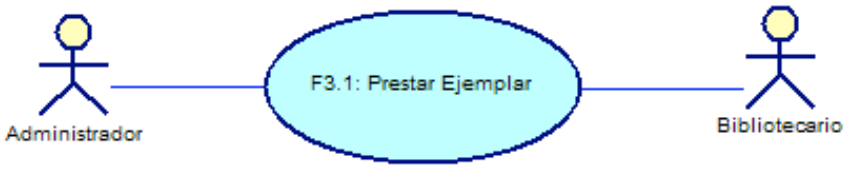
id	*Nombre*	*Acción*
E1	No hay información	No existen Mapas Registrados
E2	Mapa no encontrado	Emite mensaje de error de no existencia del Mapa
E3	Validación de datos	Emite error de datos incorrectos

1.4.5 Prestamo de Libros



Caso de Uso: Prestar Ejemplar

Identificador: F3.1



Descripción: Este caso de uso sirve para el prestamo de un ejemplar disponible a un usuario de la biblioteca. Esta opción podrá hacerla el administrador y el bibliotecario.

Flujo:

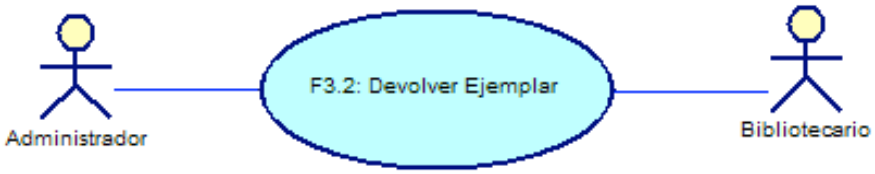
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Prestar Ejemplar	2	Presenta formulario de busqueda de libros	
		3	Carga de ejemplares disponibles	E1
4	Seleccionar el ejemplar a prestar	5	Presenta formulario de datos del usuario	
6	Ingreso de datos necesarios			
7	Notifica al sistema que el llenado de información ha sido completado	8	Validación de datos	E2
		9	Almacena información	

Excepciones:

id	*Nombre*	*Acción*
E1	No hay ejemplares disponibles	Mensaje de error de no existencia de ejemplares disponibles
E2	Validación de datos	Emite error de datos incorrectos

Caso de Uso: Devolución de Ejemplar

Identificador: F3.2



Descripción: Este caso de uso sirve para la devolución de un ejemplar por parte de un usuario a la biblioteca. Esta opción podrá hacerla el administrador y el bibliotecario.

Flujo:

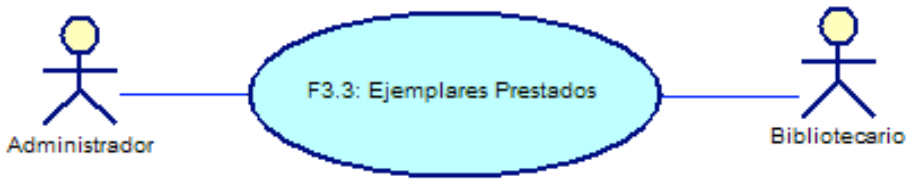
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de devolución de ejemplar	2	Presenta formulario de busqueda de usuario	
		3	Carga de usuarios con ejemplares en prestamo	E1
4	Ingreso de datos de busqueda	5	Presenta formulario de devolución de ejemplar	
6	Notifica al sistema el ejemplar a devolver	7	Almacena información	

Excepciones:

id	*Nombre*	*Acción*
E1	No hay usuarios	Mensaje de error de no existencia de usuario

Caso de Uso: Ejemplares Prestados

Identificador: F3.3



Descripción: Este caso de uso sirve para consultar los ejemplares que se encuentran prestados a los usuarios de la biblioteca. Esta opción podrá hacerla el administrador y el bibliotecario.

Flujo:

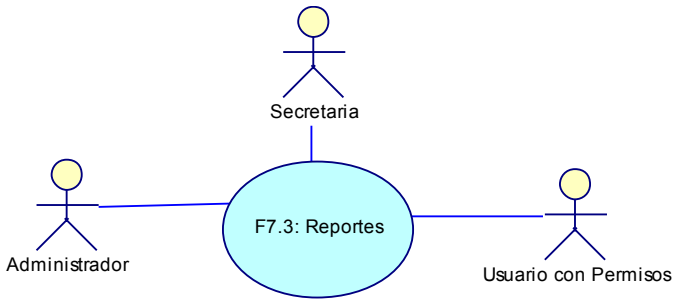
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de Ejemplares prestados	2	Genera reporte	E1

Excepciones:

id	*Nombre*	*Acción*
E1	No hay registros	Mensaje de ausencia de Ejemplares prestados.

Caso de Uso: Reportes

Identificador: F3.4



Descripción: Este caso de uso sirve para visualizar los reportes de los libros de la Biblioteca. Esta opción podrá hacerla el administrador y el bibliotecario

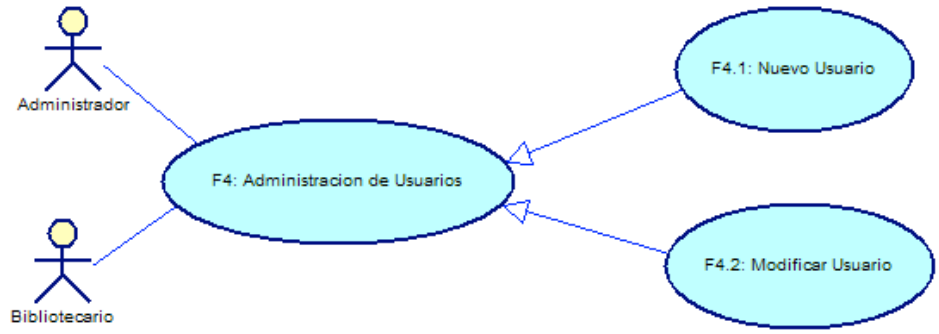
Flujo:

Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de ver reportes	2	Presenta formulario de reportes	
3	Ingreso de datos necesarios			
4	Notifica al sistema que el llenado de información ha sido completado	5	Genera reporte	E1

Excepciones:

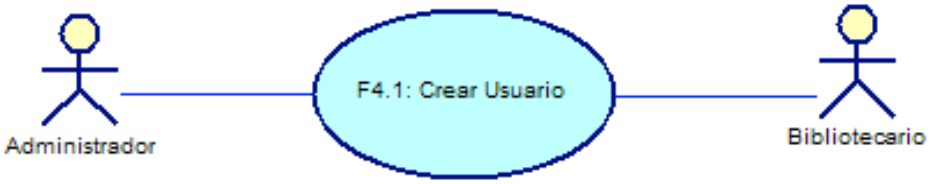
id	*Nombre*	*Acción*
E1	No hay información	No existen libros de la Junta

1.4.6 Administración de Usuarios



Caso de Uso: Crear usuario

Identificador: F8.1



Descripción: Este caso de uso sirve para la creación de un usuario del sistema. Esta opción solo podrá hacerla el administrador.

Flujo:

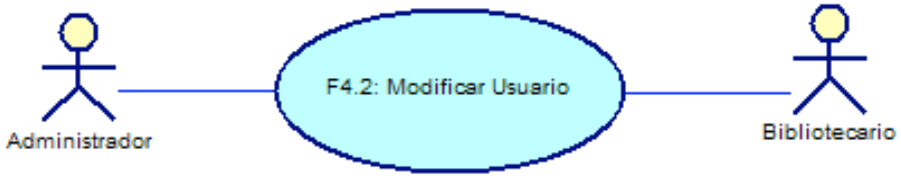
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de crear nuevo usuario del sistema	2	Presenta formulario de creación de nuevo usuario del sistema	
		3	Carga perfiles	
4	Ingreso de datos necesarios			
5	Notifica al sistema que el llenado de información ha sido completado	6	Validación de datos	E1
		7	Almacena información	

Excepciones:

id	*Nombre*	*Acción*
E1	Validación de datos	Emite un mensaje de error sobre el formato de los datos ingresados.

Caso de Uso: Modificar usuario

Identificador: F8.2



Descripción: Este caso de uso sirve para la modificación de los datos de un usuario del sistema. Esta opción solo podrá hacerla el administrador.

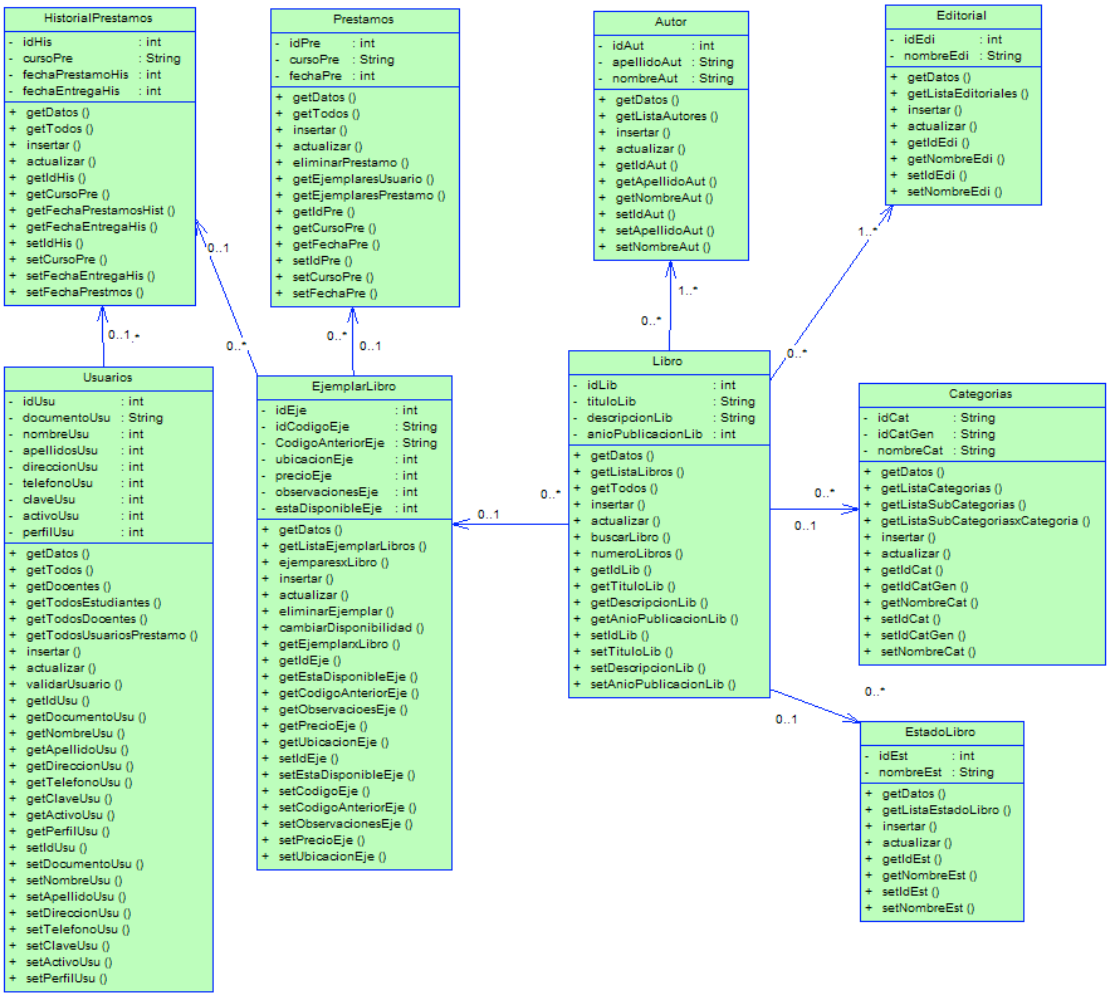
Flujo:

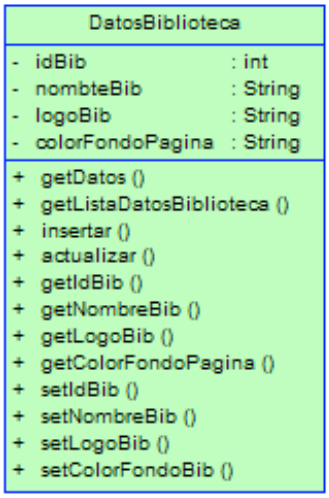
Actor		*Sistema*		
Paso	*Acción*	*Paso*	*Acción*	*Exc*
1	Emite evento de modificar usuario del sistema	2	Presenta formulario selección de usuario de sistema a modificar	
		3	Carga usuarios del sistema	
4	Selecciona usuario del sistema a modificar	5	Presenta formulario de modificación de usuario del sistema	
		6	Carga perfiles	
		7	Carga información del usuario del sistema a modificar	
8	Modifica los datos necesarios			
9	Notifica al sistema que la modificación ha sido completada	10	Validación de datos	E1
		11	Almacena información	

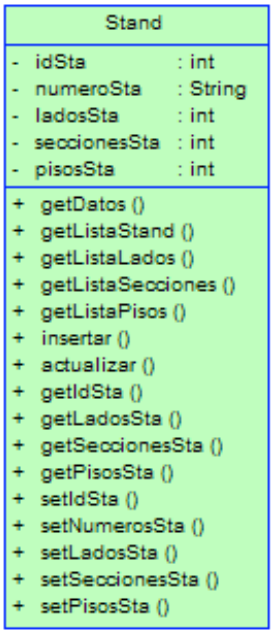
Excepciones:

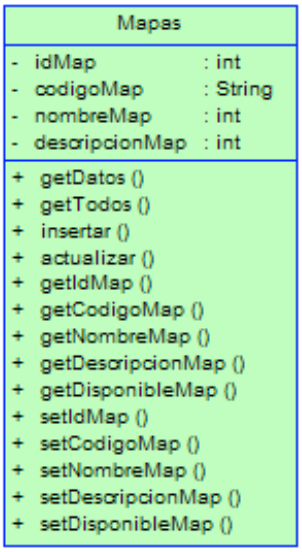
id	*Nombre*	*Acción*
E1	Validación de datos	Emite un mensaje de error sobre el formato de los datos modificados.

1.5 Diagrama de Clases



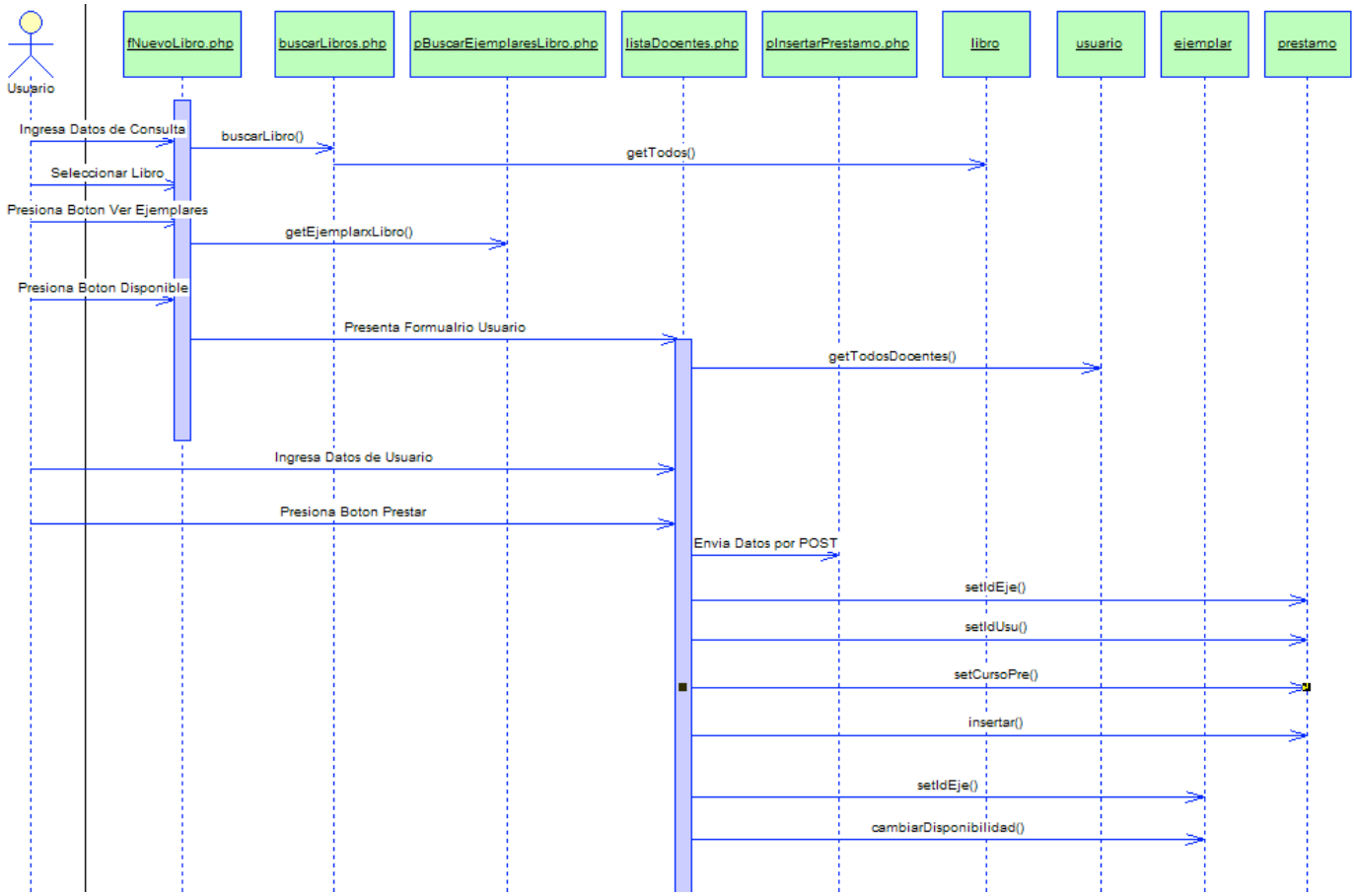

DatosBiblioteca
- idBib : int
- nombteBib : String
- logoBib : String
- colorFondoPagina : String
+ getDatos ()
+ getListaDatosBiblioteca ()
+ insertar ()
+ actualizar ()
+ getIdBib ()
+ getNombreBib ()
+ getLogoBib ()
+ getColorFondoPagina ()
+ setIdBib ()
+ setNombreBib ()
+ setLogoBib ()
+ setColorFondoBib ()


Stand
- idSta : int
- numeroSta : String
- ladosSta : int
- seccionesSta : int
- pisosSta : int
+ getDatos ()
+ getListaStand ()
+ getListaLados ()
+ getListaSecciones ()
+ getListaPisos ()
+ insertar ()
+ actualizar ()
+ getIdSta ()
+ getLadosSta ()
+ getSeccionesSta ()
+ getPisosSta ()
+ setIdSta ()
+ setNumerosSta ()
+ setLadosSta ()
+ setSeccionesSta ()
+ setPisosSta ()

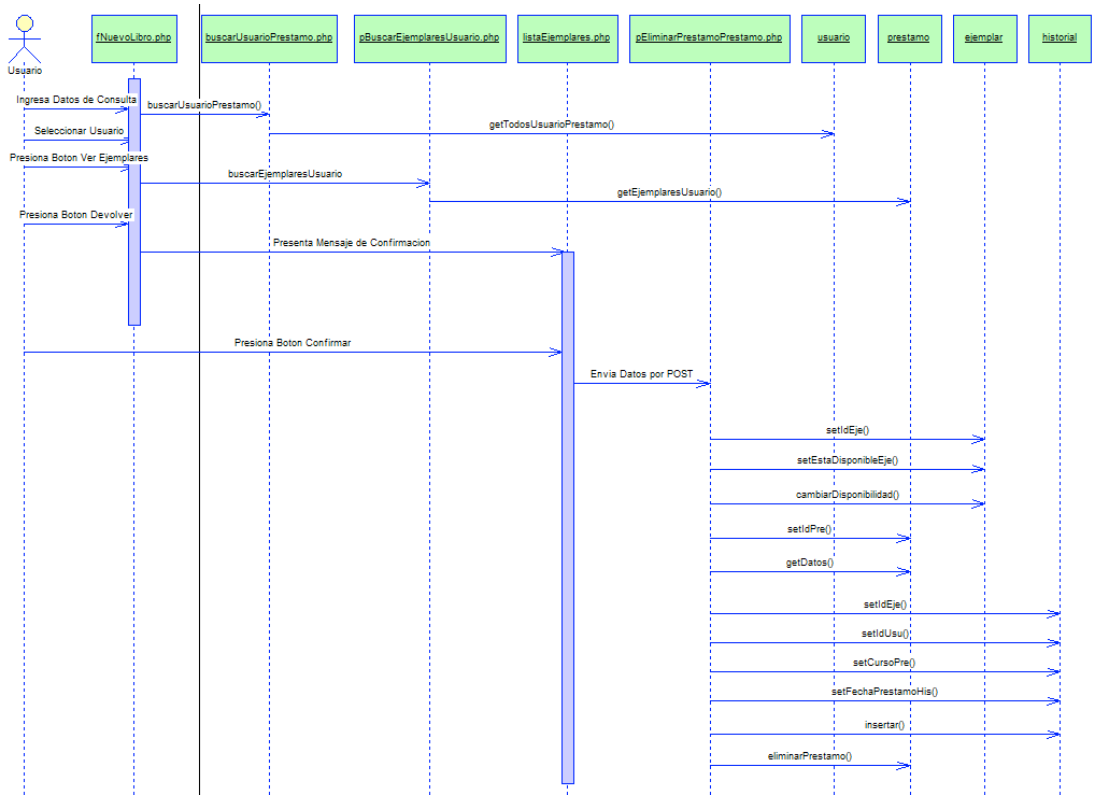

Mapas
- idMap : int
- codigoMap : String
- nombreMap : int
- descripcionMap : int
+ getDatos ()
+ getTodos ()
+ insertar ()
+ actualizar ()
+ getIdMap ()
+ getCodigoMap ()
+ getNombreMap ()
+ getDescripcionMap ()
+ getDisponibleMap ()
+ setIdMap ()
+ setCodigoMap ()
+ setNombreMap ()
+ setDescripcionMap ()
+ setDisponibleMap ()

1.6 Diagramas de Secuencia del Módulo de Prestamos

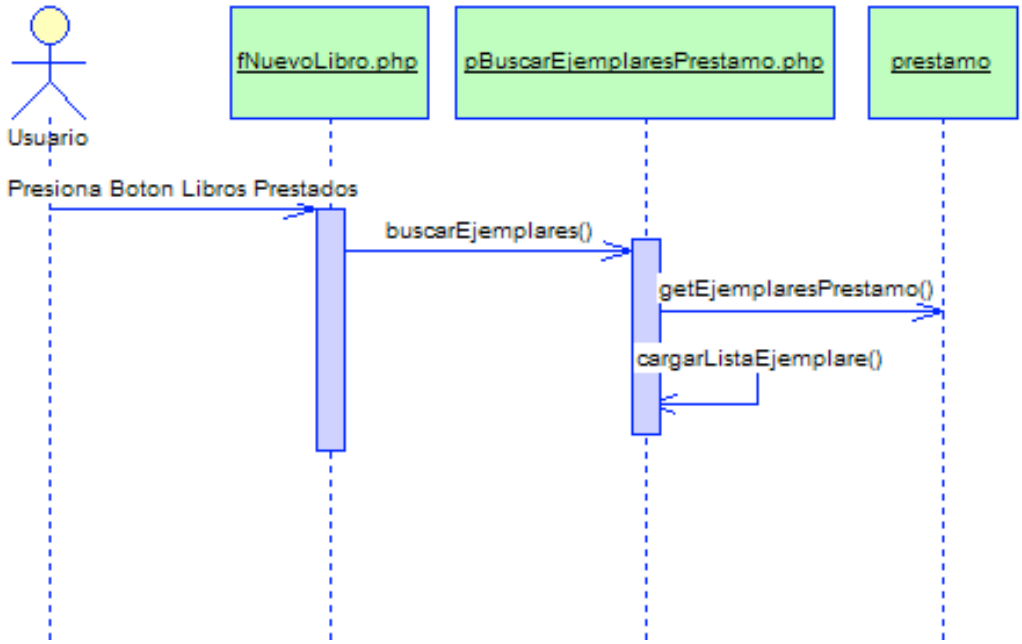
1.6.1 Prestar Libro



1.6.2 Devolución de Libro



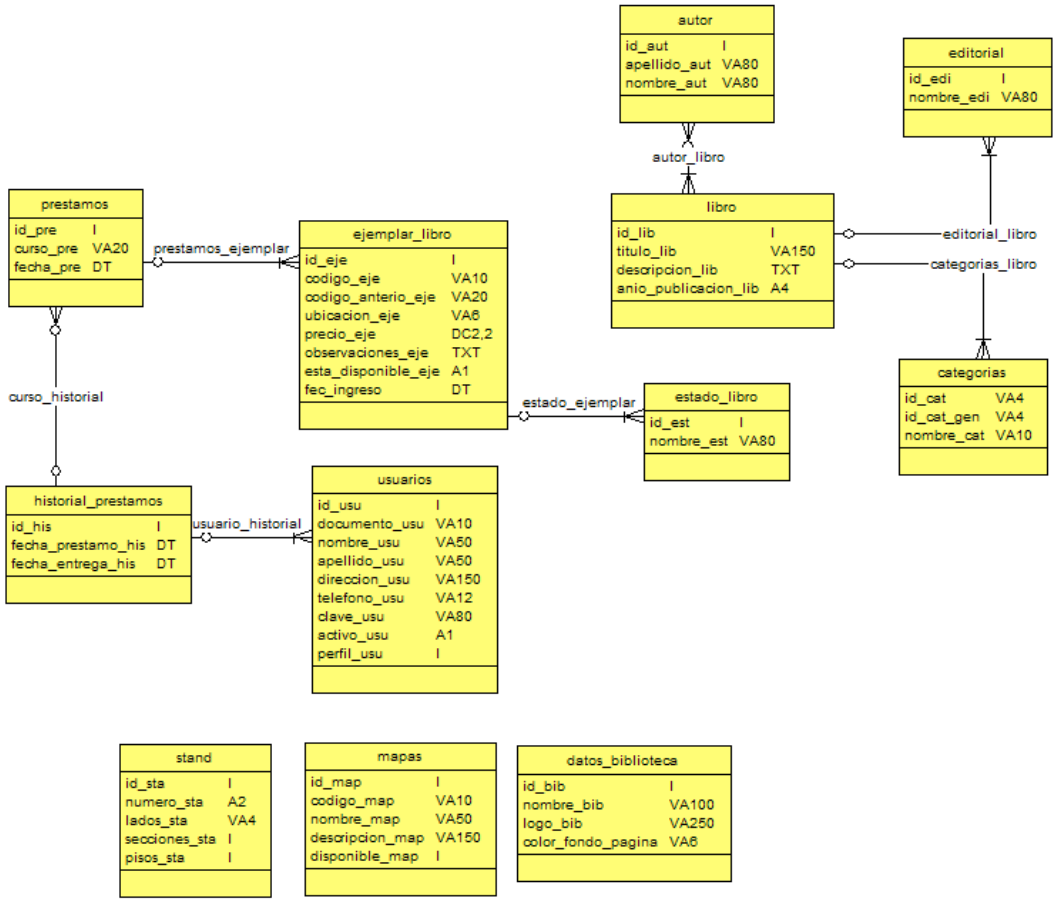
1.6.3 Libros Prestados



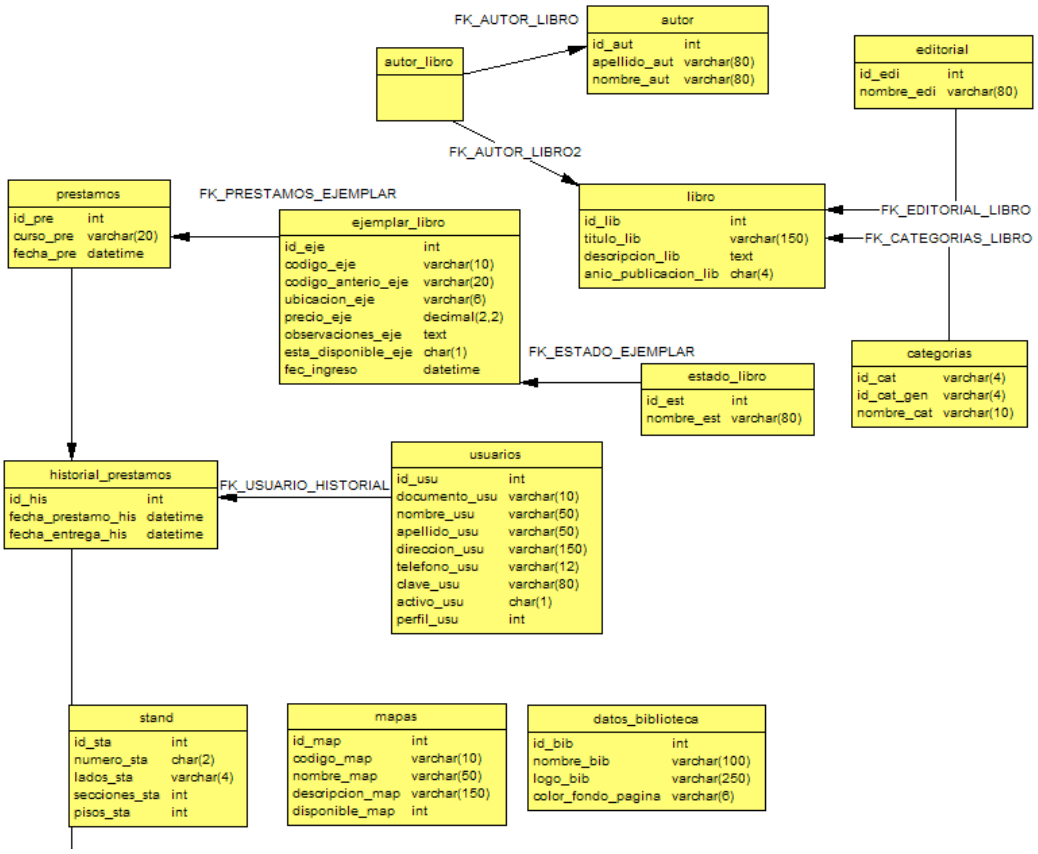
CAPÍTULO II

1.1 Modelo de Datos

1.1.1 Modelo Conceptual:



1.2 Modelo Fisico



1.3 Metodologia y Herramientas

1.3.1 Metodologia de Diseño y Desarrollo

Extreme programming es una nueva disciplina del desarrollo de software que se basa en la comunicación, simplicidad, retroalimentación y valor.

La simplicidad ayuda a que los desarrolladores de software encuentren soluciones más simples a problemas, según el cliente lo estipula. Los desarrolladores también crean características en el diseño que pudieran ayudar a resolver problemas en un futuro.

La comunicación prevalece en todas las prácticas de extreme programming. Comunicación cara a cara es la mejor forma de comunicación, entre los desarrolladores y el cliente. Método muy ágil. Gracias a esto el equipo esta pude realizar cambios que al cliente no le gustaron. También apoya agilidad con la extensión del conocimiento tácito dentro del equipo del desarrollo, evitando la necesidad de mantener la documentación escrita.

La Retroalimentación continua del cliente permite a los desarrolladores llevar y dirigir el proyecto en una dirección correcta hacia donde el cliente quiera.

El valor requiere que los desarrolladores vayan a la par con el cambio, por que sabemos que este cambio es inevitable, pero el estar preparado con una metodología ayuda a ese cambio.

La metodología se diseña para entregar el software según las necesidades de cliente y cuando sea necesaria.

Extreme programming promueve el trabajo del equipo. Cada integrante del proyecto (cliente, desarrolladores, etc.) forman parte integral del equipo encargado de desarrollar software de calidad.

13 prácticas incorporan estos valores y se describen así:

- **El Juego de Planificación (The planning game).** Son una serie de actividades planificadas que son llamados juegos planificados por extreme programming, son asumidos durante el transcurso del proyecto. Un juego inicial planeado se emprende durante el inicio del proyecto e implica una sesión con el cliente de reflexión con algunos personas claves como desarrolladores para determinar los requisitos iniciales y el contorno, conocidos en la metodología como historias. Si el proyecto continúa, se sigue con un emprende “Planning game de lanzamiento” donde el cliente escribe otras historias en tarjetas y los desarrolladores buscan términos de Unidades Ideales de Ingeniería. El cliente también planea un calendario iterativo de desarrollo, cada iteración funciona entre dos y cuatro semanas,

y subdivide historias en base a la prioridad del negocio. En el comienzo de cada iteración el cliente puede agregar otras historias, y cambiar el horario de la historia. En realidad pretende un permanente dialogo entre la parte empresarial y técnica del proyecto, en la que los empresarios deciden el alcance, la prioridad, el contenido de las versiones y la fecha de estas. Y la parte técnica, son los responsables de establecer la duración para implementar las funciones deseados por el cliente, de desarrollar la planificación de cada versión, entre otras.

- **Desarrollo de Probar-Conducir. (Test-driven development).** Esta es la forma de prueba de unidad y de prueba de aceptación, donde las pruebas se escriben antes de la implementación del código. Las pruebas de aceptación, son escritas por el cliente, se automatizan y prueban generalmente la funcionalidad del sistema. Las pruebas de unidad son escritas por los desarrolladores y prueba que las unidades del código estén a un nivel según lo esperado.

- **Programación del par (Pair programming).** Extreme programming requiere que todo el código de la producción sea producido por un par desarrolladores compartiendo un monitor y un teclado en una estación de trabajo. Uno actúa como conductor y escribe código mientras que el otro, el navegador, observa qué está pasando y proporciona consejos estratégicos para resolver los problemas de manera eficiente. El conductor tiene que estar explicando que es lo que esta haciendo y se pueden intercambiar los roles con frecuencia. Por otro lado, para facilitar el trabajo entre los miembros del equipo, estos pueden interactuar con otros individuos para realizar la tarea.

- **El reorganizado sin piedad (Merciless refactoring**). Los desarrolladores requieren mejorar el código sin que este cambie su función para realizar un código simple. Esto dará oportunidad a modificarlo cuando haya necesidad de cambiar una característica.

- **Diseño simple (Simple Design)**. Los desarrolladores necesitan seguir el principio de YAGNI (You Are not Going to Need It)(usted no lo va a necesitar) al producir un diseño para una historia dada. El diseño no debe no incluir otra cosa que no sea necesaria para ejecutar la historia y pasar las pruebas de unidad. Y lo que en realidad busca extreme programming es buscar soluciones y diseñar en cada momento para las decisiones necesitadas en ese momento.

- **Propiedad colectiva del código (Colective code ownership)**. Todos los desarrolladores son propietarios del código. Cualquier desarrollador puede modificar el código base en cualquier momento aunque haya sido hecho por otro par de desarrolladores, por ejemplo, si un desarrollador encuentra otra forma de hacer el código de manera mas simple lo puede hacer sin

ningún inconveniente, claro siempre y cuando no altere su funcionalidad. El objetivo es quitar dependencias en individuos.

- **Integración continua (Continuos Integration).** Al cabo de un día el sistema deberá de ser integrado por una máquina, cada vez que un para de programadores tengan una clase ya probada unitariamente. Si al añadir la clase el sistema completo sigue funcionando correctamente, la tarea fue realizada con éxito. De lo contrario, esto permite que los desarrolladores detecten errores en la integración del sistema en una etapa antes ejecutando las pruebas de unidad del proyecto. Si después de esto no pueden realizar el código, lo borraran y comenzaran desde cero.

- **Cliente en el sitio (On-site Costumer).** Los desarrolladores tienen acceso continuo con el cliente para poner en claro las historias y discutir el desarrollo de ediciones y proporcionar retroalimentación. Y un cliente real debe permanecer junto al equipo de desarrollo, para cualquiera duda o aclaración.

- **Lanzamientos pequeños (Small releases)**. Al final de cada iteración el sistema es lanzado y el cliente lo observa. Se lanza primero unos meses antes de estar completamente terminado, las otras versiones serán mas frecuentes entre un día y un mes. La mayoría de estos lanzamientos es para conseguir retroalimentación del cliente.

- **Semana de 40 horas (40 hours week).** Si queremos que sea un trabajo de calidad , los trabajadores tienen que estar bien descansados , no superando la regla de las 40 horas semanales y que tengan otras cosas que hacer y no estén obsesionados con el trabajo. El objetivo no es el trabajo extra, sino mantener el balance entre el trabajo y la vida. Esto está en los intereses del proyecto.

- **Aplique los estándares de la codificación (Apply coding standards)**. El código es la forma principal de documentación y por lo tanto debe de estar escrito de una manera clara y constante, que pueda ser identificado fácilmente por cualquier programador de otro equipo.

- **Metáfora del sistema (System metaphore)**. Un lenguaje de metáforas se utiliza para describir la arquitectura del sistema. Esto ayuda a la comunicación entre los mismos desarrolladores y entre los desarrolladores y el cliente.

- **Reunión (Stand-up meeting)**. Una reunión de 15 minutos en el comienzo de cada día para discutir problemas encontrados el día anterior y los problemas que se resolverán durante el día.

Extreme programming es como un rompecabezas .Hay muchos pedazos pequeños. Los pedazos individualmente no tienen ningún sentido, pero cuando están combinados pueden ser considerados un cuadro completo. Esto es una salida significativa de métodos tradicionales y del desarrollo del software en un cambio en la manera que programamos.

Extreme programming establece cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance.

También establece, que de estas cuatro variables que tenemos, solo tres de ellas podrán ser fijadas o indicadas por el cliente o jefe del proyecto, mientras que una variable quedará libre.

Un problema de esto es la calidad, por que muchas veces se ignora, por que nadie es capaz de trabajar bien cuando esta sometido a mucha presión.

Las variables entre sí, no guardan una relación tan obvia. Extreme programming, hace énfasis en equipos de desarrollo pequeños entre diez y doce personas, que podrán ir incrementando.

El costo del proyecto se incrementa cuando se necesita máquinas más rápidas, mas especialistas técnicos en determinadas áreas o mejores oficinas para el equipo de desarrollo.

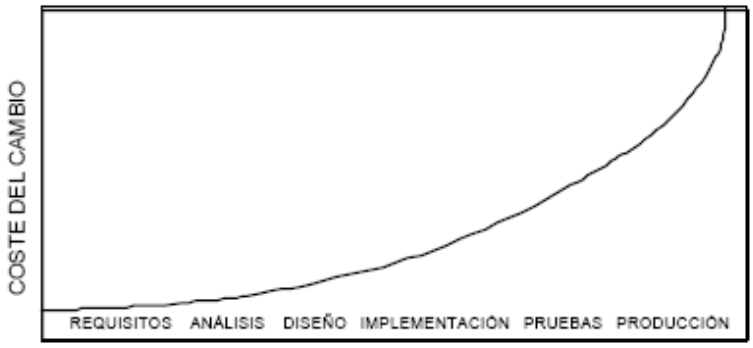
La calidad puede representar un cambio extraño; debido a que a mayor calidad menor tiempo de realización del proyecto. Por lo tanto el equipo de desarrolladores esta encargado de la tarea de hacer las pruebas con los mejores resultados posibles para así tener una idea de cual es el problema y como lo van a resolver de una manera simple y eficiente, para que la calidad del proyecto se mantenga al 100% y tener una facilidad de adaptarse a los cambios del código lo que hace este proceso más rápido.

No tener en cuenta que la gente trabaja mejor cuando la dejan hacer trabajo de calidad y sin presión puede llegar a que el proyecto no se desarrolle a su tiempo y con una calidad por los suelos.

La ultima variable es la que se encuentra libre y como ya mencioné no es establecida por los directores, sino que ya que se tiene una buen idea de las otras tres variables, se continúa a realizar el proceso de alcance del proyecto, en la cual el equipo determina: la estimación de las tareas a realizar, que es lo que el cliente quiere, la implementación de los requisitos mas importantes de manera que este siempre sea funcional.

En extreme programming el costo del cambio maneja un papel muy importante, por que comparado con otras metodologías para implementar software, es mucho mas barato, debido a que las pruebas se van haciendo según las versiones liberadas, no es como una metodología normal, que primero se realiza el análisis,

después el diseño, implementación, pruebas y finalmente producción, mientras que en la extreme programming siempre estas implementando, probando y produciendo.



*.Costo del cambio en la ingeniería de software tradicional.



*Costo del cambio en extreme programming.

Los largos ciclos de desarrollo de las metodologías tradicionales son incapaces de adaptarse al cambio, tal vez una solución puede ser ciclos de desarrollo más cortos como la extreme programming.

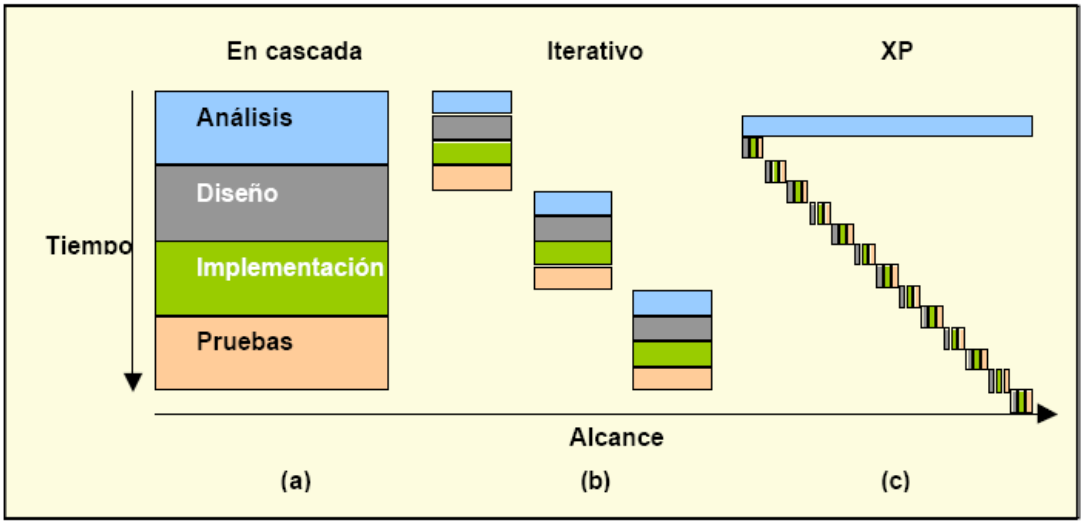


*Diagrama de los ciclos de desarrollo de software

1.3.2 Herramientas de Desarrollo

NETBEANS 6.8

NetBeans IDE es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++.

El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de terceros.

Funciones Destacadas para PHP

- Compatibilidad total con PHP 5.3: espacios de nombre, funciones y clausuras lambda, adiciones a la sintaxis: NOWDOC, condiciones ternarias, etiquetas de salto, __callStatic()
- Compatibilidad con Symfony Framework: proyectos de Symfony, comandos de Symfony, métodos abreviados de teclado, resaltado de sintaxis PHP en archivos YAML
- Creación de proyectos PHP desde aplicaciones PHP remotas
- PHPUnit, Código de cobertura, mejoras en la integración de FTP/SFTP, exclusión de las carpetas del proyecto PHP de exploraciones e indexaciones

CAPÍTULO V

1.1 Conclusiones

1. El lenguaje de programación PHP es un lenguaje muy versátil que permite desarrollar desde paginas web dinámicas hasta aplicaciones web de gran escala gracias a su multitud de funciones y su corta curva de aprendizaje.

2. El uso de librerías JavaScript Avanzadas como jQuery, Prototype entre otras, han permitido que las interfaces de las aplicaciones web se acerquen más a las aplicaciones de escritorio mejorando su desempeño y la facilidad de uso para los usuarios de las mismas.

3. El desarrollo de una aplicación web requiere de una buena metodología de programación y de una buena organización en el código de lo contrario el mantenimiento y la escalabilidad de la aplicación web se convertirán en una tarea muy complicada.

4. Las aplicaciones web permiten la implementación de interfaces para diferentes tipos de dispositivos de manera fácil y rápida brindando a los usuarios un mejor acceso a las mismas dependiendo de cada una de sus necesidades.

5. Es de gran importancia establecer un diálogo cordial y constante con el cliente y los usuarios del sistema para lograr un adecuado levantamiento de procesos donde se identifiquen completamente los requerimientos de los mismos y para facilitar el proceso de implantación ya que los usuarios se identificaran con el uso del sistema en el desempeño de sus labores diarias.

1.2 Recomendaciones

- ✓ Es recomendable instalar la aplicación web desarrollada con PHP y MySQL en un servidor con ambiente GNU/Linux con servidor web APACHE para poder aprovechar todas las capacidades de rendimiento que cada una de estas herramientas brinda para tiene el mejor desempeño.

- ✓ No es recomendable hacer un uso excesivo de AJAX dentro de la aplicación web ya que el desempeño de la misma dependera en un porcentaje muy alto del cliente (PC y navegador web) que la este ejecutando.

ANEXOS

INSTITUTO TECNOLÓGICO "BOLÍVAR"
Tulcán - Ecuador
Telf. 2980327

Msc. Rodrigo Sánchez V. **RECTOR DEL INSTITUTO TECNOLÓGICO "BOLÌVAR"** a petición verbal de parte interesada y en legal forma:

C E R T I F I C A

Que, el Señor **NOEL SANTIAGO GALVIS DUARTE**, portador de la cédula de ciudadanía No. 1720462165 desarrolló e implemento satisfactoriamente el sistema de Administración para el Departamento de Biblioteca del "Instituto Tecnológico Bolívar" de la ciudad de Tulcán, Cantón Tulcán, Provincia del Carchi.

Que, el sistema cumple a cabalidad con las expectativas y necesidades que nuestra Institución puso en este sistema; demostrando profesionalismo, y destreza en el conocimiento del sistema informático desarrollado.

Que, el Instituto Tecnológico "Bolívar" utilizará en beneficio de los estudiantes que se preparan en nuestra Institución el sistema de Administración Bibliotecario el cual facilitará los procesos estudiantiles.

Que es todo cuanto puedo certificar en honor a la verdad.

Faculto al peticionario hacer legal uso de esta certificación.

Tulcán, 12 de noviembre de 2010

Msc. Rodrigo Sánchez V.
RECTOR

INSTITUTO TECNOLOGICO "BOLIVAR" RECTORADO TULCAN - ECUADOR

Diccionario de Datos

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **autor**

Table: autor

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_aut	int(11)	Not Null		Yes	Yes	
Description:	Identificador Unico del Autor					
apellido_aut	varchar(80)	Not Null				
Description:	Apellidos del Autor					
nombre_aut	varchar(80)	Not Null				
Description:	Nombres del Autor					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_aut`	Yes		

Description

Datos de los Autores

DDL

```
CREATE TABLE `autor` (
  `id_aut` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador Unico del
Autor',
  `apellido_aut` VARCHAR(80) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Apellidos del Autor',
  `nombre_aut` VARCHAR(80) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Nombres del Autor',
  PRIMARY KEY (`id_aut`)

)ENGINE=InnoDB
AUTO_INCREMENT=35 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Datos de los Autores';
```

Date / Time:	**24 February 2011**	User:	**root**
Database:	**biblioteca**	Table:	**autor_libro**

Table: autor_libro

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_aut	int(11)					
Description:	Identificador Unico del Autor					
id_lib	int(11)	Not Null				
Description:	Identificador Unico del Libro					

Indices

Index Name	On Field	Unique	Full Text	Sorting
autor_libro_libro_fk	`id_lib`			
autor_libro_autor_fk	`id_aut`			

Description

Autores por libro

DDL

```
CREATE TABLE `autor_libro` (
  `id_aut` INTEGER(11) DEFAULT NULL COMMENT 'Identificador Unico del Autor',
  `id_lib` INTEGER(11) NOT NULL COMMENT 'Identificador Unico del Libro',
  KEY `autor_libro_libro_fk` (`id_lib`),
  KEY `autor_libro_autor_fk` (`id_aut`),
  CONSTRAINT `autor_libro_autor_fk` FOREIGN KEY (`id_aut`) REFERENCES `autor`
(`id_aut`),
  CONSTRAINT `autor_libro_libro_fk` FOREIGN KEY (`id_lib`) REFERENCES `libro`
(`id_lib`)

)ENGINE=InnoDB
CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Autores por libro';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **categorias**

Table: categorias

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_cat	varchar(4)	Not Null		Yes		
Description:	Identificador Unico de la Categoria					
id_cat_gen	varchar(4)					
Description:	Identificador de la Subcategoria					
nombre_cat	varchar(100)	Not Null				
Description:	Nombre de la Categoria o Subcategoria					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_cat`	Yes		
id_cat_gen	`id_cat_gen`			

Description

Categorias para organizar los libros segun Dewey

DDL

```
CREATE TABLE `categorias` (
  `id_cat` VARCHAR(4) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Identificador Unico de la Categoria',
  `id_cat_gen` VARCHAR(4) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'Identificador de la Subcategoria',
  `nombre_cat` VARCHAR(100) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Nombre de la Categoria o Subcategoria',
  PRIMARY KEY (`id_cat`),
  KEY `id_cat_gen` (`id_cat_gen`),
  CONSTRAINT `categorias_fk` FOREIGN KEY (`id_cat_gen`) REFERENCES `categorias`
(`id_cat`)

)ENGINE=InnoDB
CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Categorias para organizar los libros segun Dewey';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **datos_biblioteca**

Table: datos_biblioteca

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_bib	int(11)	Not Null		Yes	Yes	
Description:	Identificador Unico					
nombre_bib	varchar(100)	Not Null				
Description:	Nombre de la Biblioteca					
logo_bib	varchar(250)	Not Null				
Description:	Ubicacion de la imagen del logo para la biblioteca					
color_fondo_pagina	varchar(6)	Not Null				
Description:	Color definido en hexadecimal para el sistema					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_bib`	Yes		

Description

Datos de la Biblioteca

DDL

```
CREATE TABLE `datos_biblioteca` (
  `id_bib` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador Unico',
  `nombre_bib` VARCHAR(100) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Nombre de la Biblioteca',
  `logo_bib` VARCHAR(250) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Ubicacion de la imagen del logo para la biblioteca',
  `color_fondo_pagina` VARCHAR(6) COLLATE utf8_unicode_ci NOT NULL DEFAULT ''
COMMENT 'Color definido en hexadecimal para el sistema',
  PRIMARY KEY (`id_bib`)

)ENGINE=InnoDB
AUTO_INCREMENT=1 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Datos de la Biblioteca';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **editorial**

Table: editorial

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_edi	int(11)	Not Null		Yes	Yes	
Description:	Identificador Unico de la Editorial					
nombre_edi	varchar(80)	Not Null				
Description:	Nombre de la Editorial					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_edi`	Yes		

Description

Editoriales de los Libros

DDL

```
CREATE TABLE `editorial` (
  `id_edi` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador Unico de la
Editorial',
  `nombre_edi` VARCHAR(80) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Nombre de la Editorial',
  PRIMARY KEY (`id_edi`)

)ENGINE=InnoDB
AUTO_INCREMENT=14 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Editoriales de los Libros';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **ejemplar_libro**

Table: ejemplar_libro

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_eje	int(11)	Not Null		Yes	Yes	
Description:	Identificador Unico del Ejemplar					
id_lib	int(11)	Not Null				
Description:	Identificador de libro al que pertenece					
id_est	int(11)	Not Null				
Description:	Identificador del Estado					
codigo_eje	varchar(10)	Not Null				
Description:	Codigo de la categoria del Ejemplar					
codigo_anterior_eje	varchar(20)					
Description:	Codigo antiguo del Ejemplar					
ubicacion_eje	varchar(6)	Not Null				
Description:	Ubicacion fisica del Ejemplar					
precio_eje	decimal(5,2)					
Description:	Precio del Ejemplar					
observaciones_eje	varchar(100)	Not Null				
Description:	Observaciones del Ejemplar					
esta_disponible_eje	char(1)	Not Null				
Description:	Esta disponible o no el Ejemplar					
fec_ingreso	datetime					
Description:	Fecha de ingreso del Ejemplar					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_eje`	Yes		
id_lib	`id_lib`			
id_est	`id_est`			

Description

La cantidad de copias que se tienen de los libros

DDL

```
CREATE TABLE `ejemplar_libro` (
  `id_eje` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador Unico del
Ejemplar',
  `id_lib` INTEGER(11) NOT NULL COMMENT 'Identificador de libro al que pertenece',
  `id_est` INTEGER(11) NOT NULL COMMENT 'Identificador del Estado',
  `codigo_eje` VARCHAR(10) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Codigo de la categoria del Ejemplar',
  `codigo_anterior_eje` VARCHAR(20) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'Codigo antiguo del Ejemplar',
  `ubicacion_eje` VARCHAR(6) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Ubicacion fisica del Ejemplar',
  `precio_eje` DECIMAL(5,2) DEFAULT NULL COMMENT 'Precio del Ejemplar',
  `observaciones_eje` VARCHAR(100) COLLATE utf8_unicode_ci NOT NULL DEFAULT ''
COMMENT 'Observaciones del Ejemplar',
  `esta_disponible_eje` CHAR(1) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Esta disponible o no el Ejemplar',
  `fec_ingreso` DATETIME DEFAULT NULL COMMENT 'Fecha de ingreso del Ejemplar',
  PRIMARY KEY (`id_eje`),
  KEY `id_lib` (`id_lib`),
  KEY `id_est` (`id_est`),
```

```
  CONSTRAINT `ejemplar_libro_estado_fk` FOREIGN KEY (`id_est`) REFERENCES
`estado_libro` (`id_est`),
  CONSTRAINT `ejemplar_libro_fk` FOREIGN KEY (`id_lib`) REFERENCES `libro`
(`id_lib`)

)ENGINE=InnoDB
AUTO_INCREMENT=45 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='La cantidad de copias que se tienen de los libros';
```

Table: estado_libro

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_est	int(11)	Not Null		Yes	Yes	
Description:	Identificador unico del Estado					
nombre_est	varchar(80)	Not Null				
Description:	Descripcion del Estado					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_est`	Yes		

Description

Estado de los Ejemplares

DDL

```
CREATE TABLE `estado_libro` (
  `id_est` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador unico del
Estado',
  `nombre_est` VARCHAR(80) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Descripcion del Estado',
  PRIMARY KEY (`id_est`)

)ENGINE=InnoDB
AUTO_INCREMENT=4 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Estado de los Ejemplares';
```

Date / Time:	**24 February 2011**	User:	**root**
Database:	**biblioteca**	Table:	**historial_prestamos**

Table: historial_prestamos

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_his	int(11)	Not Null		Yes	Yes	
Description:	Id					
id_eje	int(11)					
Description:	Ejemplar					
id_usu	int(11)					
Description:	Usuario					
curso_pre	varchar(20)					
Description:	Curso					
fecha_prestamo_his	datetime					
Description:	Fecha Prestamo					
fecha_entrega_his	datetime					
Description:	Fecha Entrega					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_his`	Yes		
id_eje	`id_eje`			
id_usu	`id_usu`			

Description

Historial de Prestamo de los Ejemplares

DDL

```
CREATE TABLE `historial_prestamos` (
  `id_his` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Id',
  `id_eje` INTEGER(11) DEFAULT NULL COMMENT 'Ejemplar',
  `id_usu` INTEGER(11) DEFAULT NULL COMMENT 'Usuario',
  `curso_pre` VARCHAR(20) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT 'Curso',
  `fecha_prestamo_his` DATETIME DEFAULT NULL COMMENT 'Fecha Prestamo',
  `fecha_entrega_his` DATETIME DEFAULT NULL COMMENT 'Fecha Entrega',
  PRIMARY KEY (`id_his`),
  KEY `id_eje` (`id_eje`),
  KEY `id_usu` (`id_usu`),
  CONSTRAINT `historial_prestamos_fk` FOREIGN KEY (`id_eje`) REFERENCES
`ejemplar_libro` (`id_eje`),
  CONSTRAINT `historial_prestamos_fk1` FOREIGN KEY (`id_usu`) REFERENCES `usuarios`
(`id_usu`)

)ENGINE=InnoDB
AUTO_INCREMENT=10 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Historial de Prestamo de los Ejemplares';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **libro**

Table: libro

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_lib	int(11)	Not Null		Yes	Yes	
Description:	ID					
id_edi	int(11)	Not Null				
Description:	Editorial					
id_cat	varchar(4)	Not Null				
Description:	Dewey					
id_enc	int(11)					
titulo_lib	varchar(150)	Not Null				
Description:	Titulo					
descripcion_lib	varchar(250)	Not Null				
Description:	Descripcion					
anio_publicacion_lib	char(4)	Not Null				
Description:	Publicacion					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_lib`	Yes		
id_edi	`id_edi`			
id_cat	`id_cat`			
id_enc	`id_enc`			

Description

Informacion del Libro

DDL

```
CREATE TABLE `libro` (
  `id_lib` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'ID',
  `id_edi` INTEGER(11) NOT NULL COMMENT 'Editorial',
  `id_cat` VARCHAR(4) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT 'Dewey',
  `id_enc` INTEGER(11) DEFAULT NULL,
  `titulo_lib` VARCHAR(150) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Titulo',
  `descripcion_lib` VARCHAR(250) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Descripcion',
  `anio_publicacion_lib` CHAR(4) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Publicacion',
  PRIMARY KEY (`id_lib`),
  KEY `id_edi` (`id_edi`),
  KEY `id_cat` (`id_cat`),
  KEY `id_enc` (`id_enc`),
  CONSTRAINT `libro_categoria_fk` FOREIGN KEY (`id_cat`) REFERENCES `categorias`
(`id_cat`),
  CONSTRAINT `libro_editorial_fk` FOREIGN KEY (`id_edi`) REFERENCES `editorial`
(`id_edi`)

)ENGINE=InnoDB
AUTO_INCREMENT=33 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Informacion del Libro';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **mapas**

Table: mapas

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_map	int(11)	Not Null		Yes	Yes	
Description:	Identificador unico del Mapa					
codigo_map	varchar(10)	Not Null				
Description:	Codigo del Mapa					
nombre_map	varchar(50)	Not Null				
Description:	Nombre del Mapa					
descripcion_map	varchar(150)	Not Null				
Description:	Descripcion del Mapa					
disponible_map	int(1)	Not Null				
Description:	Disponible o no el Mapa					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_map`	Yes		

Description

Informacion de los Mapas

DDL

```
CREATE TABLE `mapas` (
  `id_map` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador unico del
Mapa',
  `codigo_map` VARCHAR(10) COLLATE utf8_bin NOT NULL DEFAULT '' COMMENT 'Codigo del
Mapa',
  `nombre_map` VARCHAR(50) COLLATE utf8_bin NOT NULL DEFAULT '' COMMENT 'Nombre del
Mapa',
  `descripcion_map` VARCHAR(150) COLLATE utf8_bin NOT NULL DEFAULT '' COMMENT
'Descripcion del Mapa',
  `disponible_map` INTEGER(1) NOT NULL DEFAULT '1' COMMENT 'Disponible o no el
Mapa',
  PRIMARY KEY (`id_map`)

)ENGINE=InnoDB
AUTO_INCREMENT=6 CHARACTER SET 'utf8' COLLATE 'utf8_bin'
COMMENT='Informacion de los Mapas';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **prestamos**

Table: prestamos

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_pre	int(11)	Not Null		Yes	Yes	
Description:	Identificador unico de los prestamos					
id_eje	int(11)					
Description:	Identificador del Ejemplar					
id_usu	int(11)					
Description:	Identificador del Usuario					
curso_pre	varchar(20)					
Description:	Curso al que pertenece el Usuario					
fecha_pre	datetime					
Description:	Fecha del Prestamo					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_pre`	Yes		
id_eje	`id_eje`			
id_usu	`id_usu`			

Description

Informacion de los Prestamos de Ejemplares

DDL

```
CREATE TABLE `prestamos` (
  `id_pre` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador unico de los
prestamos',
  `id_eje` INTEGER(11) DEFAULT NULL COMMENT 'Identificador del Ejemplar',
  `id_usu` INTEGER(11) DEFAULT NULL COMMENT 'Identificador del Usuario',
  `curso_pre` VARCHAR(20) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT 'Curso al que
pertenece el Usuario',
  `fecha_pre` DATETIME DEFAULT NULL COMMENT 'Fecha del Prestamo',
  PRIMARY KEY (`id_pre`),
  KEY `id_eje` (`id_eje`),
  KEY `id_usu` (`id_usu`),
  CONSTRAINT `prestamos_fk` FOREIGN KEY (`id_eje`) REFERENCES `ejemplar_libro`
(`id_eje`),
  CONSTRAINT `prestamos_fk1` FOREIGN KEY (`id_usu`) REFERENCES `usuarios` (`id_usu`)

)ENGINE=InnoDB
AUTO_INCREMENT=17 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Informacion de los Prestamos de Ejemplares';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **stand**

Table: stand

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_sta	int(11)	Not Null		Yes	Yes	
Description:	Identificador unico del Stand					
numero_sta	char(2)	Not Null				
Description:	Numero de Stands					
lados_sta	varchar(4)	Not Null				
Description:	Lados del Stand					
secciones_sta	int(11)	Not Null				
Description:	Secciones del Stand					
pisos_sta	int(11)	Not Null				
Description:	Pisos del Stand					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_sta`	Yes		

Description

Los stands donde puedes poner los libros para ubicarlos

DDL

```
CREATE TABLE `stand` (
  `id_sta` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador unico del
Stand',
  `numero_sta` CHAR(2) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT 'Numero
de Stands',
  `lados_sta` VARCHAR(4) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT 'Lados
del Stand',
  `secciones_sta` INTEGER(11) NOT NULL COMMENT 'Secciones del Stand',
  `pisos_sta` INTEGER(11) NOT NULL COMMENT 'Pisos del Stand',
  PRIMARY KEY (`id_sta`)

)ENGINE=InnoDB
AUTO_INCREMENT=4 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Los stands donde puedes poner los libros para ubicarlos';
```

Date / Time: **24 February 2011** User: **root**
Database: **biblioteca** Table: **usuarios**

Table: usuarios

Fields

Name	Type	Not Null	Unique	P/K	A/I	Binary
id_usu	int(11)	Not Null		Yes	Yes	
Description:	Identificador unico del Usuario					
documento_usu	varchar(10)	Not Null				
Description:	Documento de Identidad del Usuario					
nombre_usu	varchar(50)	Not Null				
Description:	Nombre del Usuario					
apellido_usu	varchar(50)	Not Null				
Description:	Apellido del Usuario					
direccion_usu	varchar(150)	Not Null				
Description:	Direccion del Usuario					
telefono_usu	varchar(12)	Not Null				
Description:	Telefono del Usuario					
clave_usu	varchar(60)	Not Null				
Description:	Clave de acceso para el Usuario					
activo_usu	char(1)	Not Null				
Description:	Estado del Usuario					
perfil_usu	int(11)	Not Null				
Description:	Perfil del Usuario					

Indices

Index Name	On Field	Unique	Full Text	Sorting
PRIMARY	`id_usu`	Yes		

Description

Datos de los usuarios del sistema

DDL

```
CREATE TABLE `usuarios` (
  `id_usu` INTEGER(11) NOT NULL AUTO_INCREMENT COMMENT 'Identificador unico del
Usuario',
  `documento_usu` VARCHAR(10) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Documento de Identidad del Usuario',
  `nombre_usu` VARCHAR(50) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Nombre del Usuario',
  `apellido_usu` VARCHAR(50) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Apellido del Usuario',
  `direccion_usu` VARCHAR(150) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Direccion del Usuario',
  `telefono_usu` VARCHAR(12) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT
'Telefono del Usuario',
  `clave_usu` VARCHAR(60) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT 'Clave
de acceso para el Usuario',
  `activo_usu` CHAR(1) COLLATE utf8_unicode_ci NOT NULL DEFAULT '' COMMENT 'Estado
del Usuario',
  `perfil_usu` INTEGER(11) NOT NULL COMMENT 'Perfil del Usuario',
  PRIMARY KEY (`id_usu`)

)ENGINE=InnoDB
AUTO_INCREMENT=12 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci'
COMMENT='Datos de los usuarios del sistema';
```