



# **Proyecto BiblioUOC**

José Manuel Saavedra González

# Índice

1	PlanDeTrabajo .....	3
1.1	Descripción .....	3
1.2	Objetivos .....	4
1.2.1	Específicos de la asignatura .....	4
1.2.2	Propios .....	4
1.3	Planificación y temporización .....	5
1.3.1	Análisis de requisitos .....	5
1.3.2	Diseño .....	6
1.3.3	Implementación .....	6
2	Especificación .....	7
2.1	Descripción .....	7
2.2	Funcionalidades .....	7
2.2.1	Subsistema cliente .....	7
2.2.2	Subsistema administrativo .....	9
2.3	Casos de uso .....	10
2.3.1	Casos de Uso del Cliente .....	10
2.3.2	Casos de Uso del Administrador .....	14
3	Diseño .....	19
3.1	Arquitectura .....	19
3.2	Patrones de diseño .....	20
3.2.1	Capa EJB .....	20
3.2.2	Capa Web .....	25
3.3	Diagramas e interfaz gráfica .....	26
3.3.1	Subsistema cliente .....	26
3.3.2	Subsistema administrativo .....	41
3.4	Diagrama De Clases .....	56
3.5	Diagrama E-R .....	57
4	Implementación .....	59
5	Conclusiones .....	61
6	Guía de instalación y prueba .....	62

# 1 PlanDeTrabajo

---

## 1.1 Descripción

Este trabajo de fin de carrera consiste en la elaboración de una aplicación para gestionar una tienda online de libros. Para llevar a cabo esta aplicación se utilizará Java 2Platform, Enterprise Edition (J2EE), una tecnología que permite el desarrollo rápido de aplicaciones distribuidas.

La aplicación estará formada por varios tipos de componentes que podrán estar ubicados en diferentes máquinas.

- Componentes que permitirán el acceso de los clientes.
- Componentes que se ejecutarán en el servidor J2EE.
- Componentes integrantes del sistema de información del negocio.

El desarrollo de este tipo de aplicaciones resulta complejo debido a la necesidad de gestionar las transacciones entre cliente y servidor, el acceso múltiple de clientes, la seguridad, el uso de los recursos y otros detalles de bajo nivel.

La plataforma J2EE ofrece un modelo de aplicación distribuida que consta de componentes reutilizables, y proporciona servicios que dan solución a las anteriores cuestiones de forma que es posible concentrarse en los problemas propios de nuestra aplicación.

La aplicación ofrecerá las siguientes funcionalidades básicas:

- Existirá un área publica donde accederán los clientes para visualizar la información de los libros clasificados por categorías.
- Los clientes podrán hacer pedidos e incluso tener un carrito de la compra.
- Existirá un área privada donde los administradores podrán ver los usuarios, gestionar los libros (precios, categorías, etc..) y consultar pedidos.

## **1.2 Objetivos**

### **1.2.1 Específicos de la asignatura**

- Adquirir los conocimientos básicos y los conceptos sobre la arquitectura J2EE necesarios para la creación de aplicaciones J2EE.
- Conocer las etapas necesarias para escribir y desplegar un programa en la plataforma J2EE.
- Conocer las tecnologías habitualmente utilizadas en las empresas.
  - JNDI (Java Naming and Directory Interface) y el almacenamiento, recuperación y manipulación de información para su utilización por componentes J2EE.
  - EJB (Enterprise JavaBeans) y sus diferentes tipos:
    - Session EJBs
    - Entity EJBs
    - Message-driven EJB
  - Servlets y la creación de aplicaciones J2EE orientadas a la Web.
  - JSP (JavaServer Pages) y la integración de contenido HTML con Java y la tecnología J2EE
  - Seguridad en las aplicaciones J2EE
- Estudiar los patrones de diseño para aplicaciones J2EE

### **1.2.2 Propios**

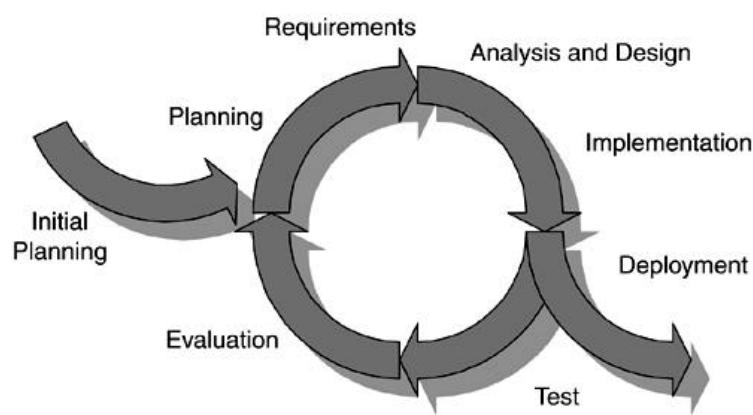
- Uso del IDE NetBeans para la codificación de aplicaciones J2EE.
- Almacenamiento y recuperación de información en la base de datos relacional MySQL.
- Conexión de aplicaciones J2EE a través de JDBC con bases de datos MySQL.

### 1.3 Planificación y temporización

El proceso de desarrollo en cascada ha sido uno de los más usados en el pasado y continúa siendo muy popular. En este modelo se divide el desarrollo en fases secuenciales (análisis de requisitos, diseño, implementación, test) y funciona bien para proyectos pequeños, en los que los requisitos son estables y de los que se conocen soluciones para proyectos similares.

Quizás el motivo de que siga siendo popular es que se ajusta bien a la planificación de las empresas. En el caso de esta universidad los plazos fijados para la entregas de las pruebas de evaluación continua hacen que en gran medida se siga este modelo.

De todas formas, siempre que sea posible es deseable seguir las prácticas de un modelo iterativo en el que la secuencia de actividades del modelo en cascada se revisita múltiples veces.



*Phillippe Kruchten. The Rational Unified Process: An Introduction. p. 7, Reading, MA: Addison-Wesley, 1999.*

Uno de los modelos iterativos más conocidos es el Proceso Unificado Racional o RUP (Rational Unified Process) que pretende implementar las mejores prácticas en ingeniería de software:

- **Desarrollo iterativo:** reduce el riesgo al afrontar las necesarias correcciones de una forma continua desde las primeras etapas. Los errores de diseño detectados en las últimas etapas del desarrollo tienen costes prohibitivos.
- **Administración de requerimientos:** En proyectos grandes los requisitos pueden variar con el tiempo y estos no se capturan en su totalidad desde el principio, se sigue un proceso de identificación gradual y refinamiento.
- **Uso de arquitectura basada en componentes:** El uso de componentes comerciales o desarrollados personalmente potencia la reutilización.
- **Control de cambios:** En proyectos en los que participan varios desarrolladores los conflictos producidos por los cambios que estos realizan pueden llevar al caos. Se hace imprescindible un control de cambios.
- **Modelado visual del software:** La construcción de modelos visuales lleva a un mejor entendimiento del problema.
- **Verificación de la calidad del software:** La realización de tests desde los primeros momentos resulta más efectivo y ayuda a determinar el estado del proyecto.

#### 1.3.1 Análisis de requisitos

En esta fase tendrá un papel muy importante el cliente que ha encargado el software, hecho que se materializará mediante las entrevistas que se le harán o los

cuestionarios que se le enviarán con el objetivo de aclarar dudas y garantizar que el documento de especificación reúne todas las funcionalidades que espera el cliente. Las tareas a realizar y que darán lugar al documento de especificación son:

- Una **descripción** en términos generales del sistema (software) que hay que desarrollar con la indicación de sus objetivos fundamentales, como qué problema resuelve, de cuántos subsistemas constará, dónde se instalará y se ejecutará (en red, etc.). Entrega: 17 de octubre.
- Una descripción precisa de las **funcionalidades** de cada subsistema por separado Entrega: 24 de octubre.
- Elaboración de los **casos de uso**. Se enumerarán y se especificarán con una precondition y una postcondición (normalmente en lenguaje natural) del caso y también con los diagramas de los casos de uso. En esta parte se indicarán también los principales actores del sistema. Entrega: 3 de noviembre.
- Elaboración del diagrama de paquetes para indicar las relaciones entre los paquetes (subsistemas) y entre los paquetes y el sistema (paquete principal). Entrega: 7 de noviembre.

### 1.3.2 Diseño

En esta fase se obtendrá el diseño para cada subsistema:

- **Fase exploratoria:** Identificar las clases y los objetos con sus responsabilidades y colaboraciones. Entrega: 16 de noviembre.
- **Diagramas en UML**
  - Diagramas de clases y jerarquías.
  - Diagramas de colaboración.
  - Diagrama de persistencia.Entrega: 30 de noviembre.
- **Diseño de la interfaz de usuario** (versión prototipo). Entrega 9 de diciembre.

### 1.3.3 Implementación

La implementación se hará del 9 de diciembre al 9 de enero. El 9 de enero será la entrega final y defensa del TFC.

## 2 Especificación

---

### 2.1 Descripción

La aplicación BiblioUOC permitirá la gestión de una biblioteca online. Deberá atender las necesidades de los clientes y también de los administradores de la biblioteca.

Los clientes podrán acceder a la biblioteca a través de Internet utilizando un navegador para consultar y comprar libros. Podrán hacer búsquedas por diferentes criterios: autor, editorial, título... y consultar información sobre los libros. Los libros seleccionados se podrán añadir a un carrito de compra para posteriormente formalizar la compra.

Existirá un grupo de administradores de la biblioteca que se encargarán de las tareas de mantenimiento: actualización de información sobre los libros (precios, nuevos libros...), información sobre los clientes y sobre los pedidos que estos realicen.

Se pueden distinguir dos subsistemas:

- **Subsistema cliente:** Los clientes de la biblioteca se conectarán a través de Internet con un navegador y tendrán a su disposición las funcionalidades que les ofrecerá este subsistema que estará ubicado en el servidor de la biblioteca.
- **Subsistema administrativo:** Este subsistema tratará de las tareas administrativas. Los administradores de la biblioteca se conectarán en la sede de la biblioteca, utilizarán para ello una aplicación cliente y podrán llevar a cabo todas las tareas necesarias para el mantenimiento de la biblioteca.

### 2.2 Funcionalidades

#### 2.2.1 Subsistema cliente

Este subsistema se ocupa de las funcionalidades que necesitan los clientes para hacer sus compras y consultas en la biblioteca.

##### 2.2.1.1 Búsquedas

Se podrán realizar búsquedas por diferentes criterios, lo que permitirá a los clientes encontrar en la base de datos de la librería los libros que deseen y visualizar la información relativa a estos.

Las opciones de que dispondrá el usuario para realizar las búsquedas serán:

- Por título
- Por autor
- Por ISBN
- Por editorial
- Por materia

##### 2.2.1.2 Registro online

El registro online permite al usuario almacenar su información en la base de datos de la librería. Así, en visitas posteriores a la librería, los usuarios facilitarán su

nombre de usuario y su contraseña para consultar el estado de sus pedidos y realizar compras de una forma más cómoda al no tener que proporcionar sus datos nuevamente.

En el registro los usuarios proporcionarán los siguientes datos:

- Nombre
- 1º Apellido
- 2º Apellido
- Dirección
- Población
- Cod Postal
- Provincia
- E-mail
- Tlf
- Contraseña

Como ya se ha comentado, se utilizará la contraseña en combinación con el e-mail para realizar la autenticación de los usuarios en posteriores visitas a la librería.

#### *2.2.1.3 Consultar/Editar información personal*

Los clientes una vez registrados y previa autenticación, podrán consultar el estado de sus pedidos y sus datos personales y si lo desean, hacer modificaciones.

La información que se almacenará en el sistema relativa a los pedidos y que se podrá visualizar es:

- Nº de pedido
- Cliente
- Relación de libros
- Estado (ENPROCESO, ENVIADO, ENTREGADO, CANCELADO)
- Fecha de pedido
- Fecha de envío
- Fecha de Entrega
- Fecha de cancelación

#### *2.2.1.4 Gestionar el carro de compra*

El carro de la compra es el instrumento que los clientes utilizarán para realizar las compras, su uso es similar al de los carros de los supermercados.

Cuando se visualice la información de un libro se tendrá la opción de añadir el libro al carro de la compra. En cualquier momento el usuario podrá consultar el contenido de este carro y modificarlo.

La pantalla de visualización del carrito mostrará los libros que se han añadido y la cantidad de ejemplares que se desean comprar, el precio de cada ejemplar, y los totales con el IVA detallado y el total a pagar.

El usuario podrá modificar el número de ejemplares que desea comprar o eliminar un libro del carrito. Después de hacer las modificaciones que crea convenientes podrá actualizar la pantalla para consultar nuevamente los precios y el total a pagar.

#### *2.2.1.5 Efectuar la compra*

Cuando así lo decida, el usuario podrá formalizar la compra de los artículos que se encuentren en su carrito.

Se solicitará al cliente que se autentifique o que se registre si no es todavía un usuario registrado.

En pantallas sucesivas se solicitará al usuario que confirme sus datos y se le ofrecerá la posibilidad de modificarlos. Finalmente el cliente deberá confirmar la



compra y se almacenará el pedido en el sistema para su tratamiento. Posteriormente el cliente podrá consultar el estado de su pedido.

## **2.2.2 Subsistema administrativo**

Este subsistema ofrecerá las funcionalidades que sean precisas para el mantenimiento de la biblioteca por parte de los administradores.

### ***Gestión de usuarios***

Se podrán dar de alta y de baja, así como modificar las características de los diferentes usuarios.

#### ***2.2.2.1 Alta de Usuarios***

Esta opción permitirá añadir nuevos usuarios al sistema.

El Administrador encargado de registrar el alta tendrá que indicar:

- Nombre
- 1º Apellido
- 2º Apellido
- Dirección
- Población
- Cod Postal
- Provincia
- E-mail
- Tlf
- Contraseña

#### ***2.2.2.2 Baja de Usuarios***

Esta opción permitirá dar de baja a los usuarios del sistema. El administrador deberá introducir la dirección de correo del cliente y confirmar la baja.

#### ***2.2.2.3 Modificaciones de Usuarios***

Con esta opción, el administrador podrá modificar los datos personales de un usuario.

### ***Gestión de libros***

Se podrán añadir nuevos libros, modificarlos y borrarlos.

#### ***2.2.2.4 Alta de Libros***

Esta opción permitirá añadir nuevos libros al sistema. De cada libro se almacenarán los siguientes datos:

- Título
- Autor
- Editorial
- Encuadernación
- ISBN
- Materia
- Precio
- Fotografía (opcional)
- Descripción (opcional)

#### *2.2.2.5 Baja de Libros*

Esta opción permitirá dar de baja libros. El administrador deberá introducir el ISBN del libro y confirmar la baja.

#### *2.2.2.6 Modificaciones de libros*

Con esta opción, el administrador podrá modificar los datos de un libro.

### ***Gestión de pedidos***

Se podrán consultar los pedidos realizados y modificarlos.

#### *2.2.2.7 Consulta de pedidos*

El administrador tendrá la opción de consultar los pedidos de un determinado usuario facilitando su e-mail. La información que se almacenará en el sistema y que se podrá visualizar es:

- Nº de pedido
- Cliente
- Relación de libros
- Estado (ENPROCESO, ENVIADO, ENTREGADO, CANCELADO)
- Fecha de pedido
- Fecha de envío
- Fecha de Entrega
- Fecha de cancelación

#### *2.2.2.8 Modificación de pedidos*

Una vez seleccionado el pedido, desde la ventana de visualización se podrán modificar sus datos.

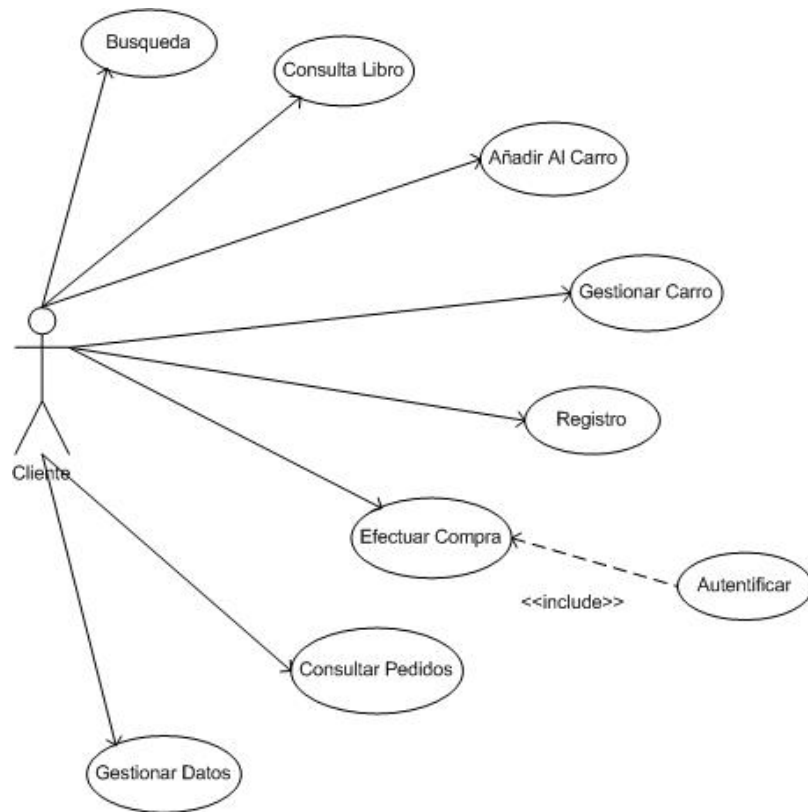
### ***Seguridad de acceso al subsistema administrativo***

El sistema de seguridad para acceder al subsistema administrativo se basa en una contraseña. En el momento de la conexión a este subsistema el administrador deberá introducir la contraseña correcta.

Existirá la posibilidad de cambiar la contraseña. El procedimiento de cambio de contraseña será convencional, pedirá la contraseña actual y después se introducirá dos veces la contraseña nueva.

## **2.3 Casos de uso**

### **2.3.1 Casos de Uso del Cliente**



#### 2.3.1.1 Caso de Uso Búsqueda

- **Funcionalidad general:** Permite hacer búsquedas de los libros de la biblioteca por diferentes criterios.
- **Papel dentro del trabajo del usuario:** Es un paso previo a la compra. Una vez encontrado un libro, se podrá consultar la información asociada y comprarlo si se desea.
- **Actores:** Cliente
- **Casos de uso relacionados:** ConsultaLibro, AñadirAlCarrito.
- **Precondición:** El cliente ha accedido al sistema. No es preciso que se autentifique.
- **Postcondición:** Se devuelven los libros almacenados en el sistema que cumplen las condiciones de la búsqueda.
- **Descripción:** El cliente selecciona el tipo de búsqueda que quiere realizar:
  - Por título
  - Por título
  - Por autor
  - Por ISBN
  - Por editorial
  - Por materia

A continuación introducirá los datos relativos a cada criterio:

- Todo o parte del título
- Nombre o parte del nombre del autor
- ISBN exacto
- Editorial, o parte del nombre de una editorial
- Materia

Por último, iniciará la búsqueda.

#### *2.3.1.2 Caso de Uso Consulta Libro*

- **Funcionalidad general:** Permite consultar la información de los libros de la biblioteca.
- **Papel dentro del trabajo del usuario:** Es un paso previo a la compra. Desde la pantalla de visualización de la información se podrá añadir el libro al carrito para después formalizar la compra.
- **Actores:** Cliente
- **Casos de uso relacionados:** ConsultaLibro, AñadirAlCarrito.
- **Precondición:** El cliente ha realizado una búsqueda y ha seleccionado un libro.
- **Postcondición:** Se muestra la información del libro seleccionado.
- **Descripción:** El cliente solicita ver la información de un libro previamente seleccionado y esta se muestra con el formato adecuado.

#### *2.3.1.3 Caso de Uso Añadir Al Carro*

- **Funcionalidad general:** Añade un libro al carro de la compra para su posterior compra.
- **Papel dentro del trabajo del usuario:** Es un paso previo a la compra. Una vez añadido al carro se podrá proceder a la formalización de la compra de los libros del carro.
- **Actores:** Cliente
- **Casos de uso relacionados:** ConsultaLibro, AñadirAlCarrito.
- **Precondición:** El cliente ha seleccionado y está consultando un libro almacenado en el sistema.
- **Postcondición:** El libro se ha añadido al carro de la compra.
- **Descripción:** Desde la pantalla de visualización de la información de un libro el cliente solicita que se añada al carro de la compra

#### *2.3.1.4 Caso de Uso Gestionar Carro*

- **Funcionalidad general:** Permite ver al usuario el contenido del carro de la compra y modificarlo.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual en el proceso de compra.
- **Actores:** Cliente
- **Casos de uso relacionados:**
- **Precondición:** El cliente ha accedido al sistema. No es necesaria su autenticación.
- **Postcondición:** Se ha mostrado el contenido del carrito y se han guardado las modificaciones que se hayan realizado.
- **Descripción:** La pantalla de visualización del carrito mostrará los libros que se han añadido y la cantidad de ejemplares que se desean comprar, el precio de cada ejemplar, y los totales con el IVA detallado y el total a pagar. El usuario podrá modificar el número de ejemplares que desea comprar o eliminar un libro del carrito. Después de hacer las modificaciones que crea convenientes podrá actualizar la pantalla para consultar nuevamente los precios y el total a pagar.

#### *2.3.1.5 Caso de Uso Registro*

- **Funcionalidad general:** Registrándose, el cliente guarda sus datos en el sistema.
- **Papel dentro del trabajo del usuario:** El Cliente debe registrarse una vez para poder autenticarse posteriormente y tener acceso a todas las funcionalidades del sistema.
- **Actores:** Cliente
- **Casos de uso relacionados:** Autenticar

- **Precondición:** El Cliente ha accedido al sistema.
- **Postcondición:** Se han guardado los datos personales del cliente en el sistema.
- **Descripción:** Se le solicitará al Cliente que proporcione la siguiente información:
  - Nombre
  - 1º Apellido
  - 2º Apellido
  - Dirección
  - Población
  - Cod Postal
  - Provincia
  - E-mail
  - Tlf
  - Contraseña

Se comprobará que el e-mail no está ya registrado en el sistema y que la contraseña es válida (entre 4 y 9 caracteres). Si no se cumplen estas condiciones se pedirá al Cliente que introduzca nuevamente los datos.

#### *2.3.1.6 Caso de Uso Autenticar*

- **Funcionalidad general:** Permite la autenticación de los clientes
- **Papel dentro del trabajo del usuario:** Es un paso previo necesario para la formalización de la compra, y para la consulta/edición de los datos personales y de pedidos.
- **Actores:** Cliente
- **Casos de uso relacionados:** Registro
- **Precondición:** El Cliente ha accedido al sistema.
- **Postcondición:** El cliente ha sido autenticado y tiene acceso a todas las funcionalidades del subsistema cliente.
- **Descripción:** Se le solicitará al Cliente que proporcione su e-mail y su contraseña. Si los datos introducidos no son correctos se le dará nuevamente la opción de identificarse.

#### *2.3.1.7 Caso de Uso Efectuar Compra*

- **Funcionalidad general:** Formaliza la compra de los artículos que se hayan añadido al carro.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual.
- **Actores:** Cliente
- **Casos de uso relacionados:** Autenticar
- **Precondición:** El cliente ha accedido al sistema y ha añadido artículos al carro. En caso contrario se le advertirá de que el carro está vacío.
- **Postcondición:** Se genera un pedido con los artículos del carro de la compra que se enviarán al cliente.
- **Descripción:** Se solicitará al cliente que se autentique o que se registre si no es todavía un usuario registrado. En pantallas sucesivas se solicitará al usuario que confirme sus datos y se le ofrecerá la posibilidad de modificarlos. Finalmente el cliente deberá confirmar la compra y se almacenará el pedido en el sistema para su tratamiento.

#### *2.3.1.8 Caso de Uso Consultar Pedidos*

- **Funcionalidad general:** Da la posibilidad al cliente de consultar el estado de sus pedidos.
- **Papel dentro del trabajo del usuario:** Es un caso de uso poco habitual.
- **Actores:** Cliente

- **Casos de uso relacionados:** Autenticar, Registro
- **Precondición:** El cliente ha accedido al sistema, es un Cliente registrado y se ha autenticado.
- **Postcondición:** Se ha mostrado la información sobre sus pedidos al cliente.
- **Descripción:** Se mostrará al cliente la información relativa a sus pedidos:
  - N° de pedido
  - Cliente
  - Relación de libros
  - Estado (ENPROCESO, ENVIADO, ENTREGADO, CANCELADO)
  - Fecha de pedido
  - Fecha de envío
  - Fecha de Entrega
  - Fecha de cancelación

#### 2.3.1.9 Caso de Uso Gestionar Datos

- **Funcionalidad general:** Hace posible que un usuario consulte y modifique sus datos personales.
- **Papel dentro del trabajo del usuario:** Es un caso de uso poco habitual. El Cliente sólo necesita registrarse una vez y sólo en caso de que alguno de sus datos cambie necesitará recurrir a esta funcionalidad.
- **Actores:** Cliente
- **Casos de uso relacionados:** Autenticar, Registro
- **Precondición:** El cliente ha accedido al sistema, es un Cliente registrado y se ha autenticado.
- **Postcondición:** Las modificaciones realizadas por el Cliente se han guardado en el sistema.
- **Descripción:** El cliente consultará sus datos personales y si lo desea procederá a modificarlos. Si el dato a modificar es la contraseña se le solicitará que la introduzca dos veces y se comprobará que es válida.

### 2.3.2 Casos de Uso del Administrador



#### *2.3.2.1 Caso de Uso Verificar Entrada*

- **Funcionalidad General:** Verifica que la contraseña sea correcta.
- **Función dentro del trabajo del usuario:** Es el primer caso de uso que se encuentra el usuario.
- **Actores:** Administrador
- **Casos de Uso relacionados:** Cambio contraseña
- **Precondición:** El usuario es conoce su contraseña.
- **Postcondición:** El sistema ha comprobado la contraseña y, si es el caso, ha permitido la entrada.
- **Descripción:** El sistema pide la contraseña, se comprueba que la contraseña sea correcta.
- **Observaciones:** Esta es la primera pantalla que se muestra la primera vez que se ejecuta el programa.

#### *2.3.2.2 Caso de Uso Cambio Contraseña*

- **Funcionalidad General:** Permite cambiar la contraseña.
- **Papel dentro del trabajo del usuario:** Es un caso de uso secundario para el actor Administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:** Verificar entrada.
- **Precondición:** El usuario ha introducido la contraseña correcta.
- **Postcondición:** El sistema ha cambiado la contraseña.
- **Descripción:** El sistema muestra una pantalla donde el usuario tendrá que indicar la contraseña actual y, introducir dos veces la nueva. Al apretar Aceptar se registra el cambio.

#### *2.3.2.3 Caso de Uso Alta usuarios*

- **Funcionalidad General:** Permite al administrador dar de alta usuarios
- **Papel dentro del trabajo del usuario:** Es un caso de uso secundario para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se ha dado de alta a un cliente en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce los siguientes datos:
  - Nombre
  - 1º Apellido
  - 2º Apellido
  - Dirección
  - Población
  - Cod Postal
  - Provincia
  - E-mail
  - Tlf
  - Contraseña

Se comprobará que el e-mail no está ya registrado en el sistema y que la contraseña es válida (entre 4 y 9 caracteres). Si no se cumplen estas condiciones se informará al administrador y no se llevará a cabo el alta.

#### *2.3.2.4 Caso de Uso Baja usuarios*

- **Funcionalidad General:** Permite al administrador dar de baja usuarios
- **Papel dentro del trabajo del usuario:** Es un caso de uso secundario para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se ha dado de baja a un cliente en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce el e-mail de un usuario. El sistema muestra la información del usuario y solicita confirmación para darlo de baja.

#### *2.3.2.5 Caso de Uso Modificación usuarios*

- **Funcionalidad General:** Permite al administrador modificar los datos de los usuarios.
- **Papel dentro del trabajo del usuario:** Es un caso de uso secundario para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se han modificado los datos de un cliente en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce el e-mail de un usuario. El sistema muestra la información del usuario y da la posibilidad de modificarla. Una vez modificados los datos se confirman las modificaciones y se realizan las comprobaciones oportunas:
  - Si se modifica la contraseña, esta debe cumplir la condición de tener entre 4 y 9 caracteres.
  - Si se modifica el e-mail, se comprobará que este no esté ya registrado en el sistema.Si las modificaciones son válidas, estas quedan registradas en el sistema. En caso contrario se informa al administrador del motivo.

#### *2.3.2.6 Caso de Uso Alta libros*

- **Funcionalidad General:** Permite al administrador dar de alta libros.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se ha dado de alta un libro en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce los siguientes datos:
  - Título
  - Autor
  - Editorial
  - Encuadernación
  - ISBN
  - Materia
  - Precio
  - Fotografía (opcional)
  - Descripción (opcional)



#### *2.3.2.7 Caso de Uso Baja libros*

- **Funcionalidad General:** Permite al administrador dar de baja libros.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se ha dado de baja un libro en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce el ISBN del libro. El sistema muestra la información del libro y solicita confirmación para darlo de baja.

#### *2.3.2.8 Caso de Uso Modificación libros*

- **Funcionalidad General:** Permite al administrador modificar los datos de los libros.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:**
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se han modificado los datos de un libro en el sistema.
- **Descripción:** El administrador escoge la opción adecuada en el menú e introduce el ISBN de un libro. El sistema muestra la información del libro y da la posibilidad de modificarla. Una vez modificados los datos se confirman las modificaciones y estas quedan registradas en el sistema.

#### *2.3.2.9 Caso de Uso Consulta pedidos*

- **Funcionalidad General:** Permite al administrador consultar los datos de los pedidos.
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:** Modificar pedidos
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta.
- **Postcondición:** Se han mostrado los datos de los pedidos.
- **Descripción:** El administrador la opción de consultar los pedidos de un determinado usuario facilitando su e-mail. La información que se almacenará en el sistema y que se podrá visualizar es:
  - N° de pedido
  - Cliente
  - Relación de libros
  - Estado (ENPROCESO, ENVIADO, ENTREGADO, CANCELADO)
  - Fecha de pedido
  - Fecha de envío
  - Fecha de Entrega
  - Fecha de cancelación

#### *2.3.2.10 Caso de Uso Modificación pedidos*

- **Funcionalidad General:** Permite al administrador modificar los datos de los pedidos.

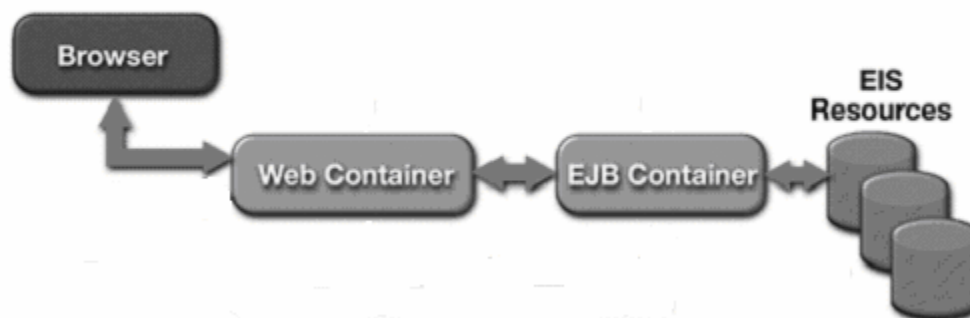
- **Papel dentro del trabajo del usuario:** Es un caso de uso habitual para el actor administrador
- **Actores:** Administrador
- **Casos de Uso relacionados:** Consultar pedidos
- **Precondición:** El administrador ha accedido al sistema introduciendo la contraseña correcta. Ha seleccionado un pedido después de hacer una consulta.
- **Postcondición:** Se han modificado los datos del pedido.
- **Descripción:** Una vez seleccionado un pedido, desde la ventana de visualización el administrador accede a la modificación de sus datos.

## 3 Diseño

### 3.1 Arquitectura

La plataforma J2EE está diseñada para proporcionar soporte para el desarrollo de aplicaciones distribuidas multicapa. Las aplicaciones de este tipo habitualmente constan de una capa cliente que proporciona la interfaz de usuario, una o más capas intermedias que proporcionan servicios a los clientes y la lógica de negocio y una última capa en la que reside el sistema de información de la empresa (EIS enterprise information system) que proporciona la gestión de datos.

Nuestra aplicación seguirá este modelo multicapa. En la capa cliente se utilizarán navegadores web para conectarse a una capa intermedia que consistirá en un Web-container y un EJB-container. Finalmente los datos se almacenarán en una base de datos MySQL.



La lógica de negocio residirá en la capa formada por EJB (Enterprise Java Beans). Los EJB permiten al desarrollador centrarse en la lógica de negocio mientras las complejidades de proporcionar un servicio confiable y escalable son manejadas por el container.

La lógica de negocio estará disponible en la WWW gracias a la capa web. Esta capa será la encargada de gestionar las comunicaciones entre clientes y la aplicación invocando la lógica de negocio y transmitiendo datos en respuesta a las peticiones entrantes. Se ocupará también del control de flujo de pantallas, de forma que la lógica que determina qué pantalla o qué página es la próxima que se visualizará residirá en esta capa.

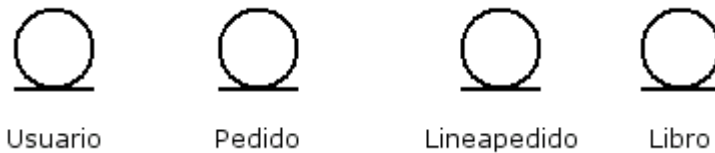
Los factores que determinan la elección de una arquitectura multicapa con web-container y EJB-container son los siguientes:

- Particionar la aplicación en presentación y lógica de negocio hace que se incremente la modularidad, esto permite por ejemplo que se apliquen cambios en el aspecto de la aplicación, que son los más habituales, sin que afecte a otras partes de la aplicación.
- Esta estructuración hace posible una asignación de recursos humanos con la preparación adecuada para cada tarea (diseñadores gráficos, especialistas en bases de datos, programadores... ) y el trabajo podrá realizarse de una manera relativamente independiente.
- Se tendrá la posibilidad de ensamblar componentes de diferentes fuentes, y de desplegar la aplicación en diferentes tipos de hardware y plataformas software.

## 3.2 Patrones de diseño

### 3.2.1 Capa EJB

Necesitamos almacenar los datos relativos a los clientes, a los libros y a los pedidos de libros que realizan los clientes. Esto dará lugar a cuatro entity beans que gestionarán la persistencia de los datos.



Se utilizará CMP (Container Managed Persistence). Hay varias razones para preferir CMP antes que BMP (Bean Managed Persistence).

- Con CMP el mantenimiento de la persistencia y de las relaciones entre entity beans se deja al container esto da lugar a aplicaciones con menos líneas de código que se desarrollan en menos tiempo y son más fáciles de mantener.
- Con BMP el desarrollador tiene que escribir sentencias SQL que pueden no ser portables a otras bases de datos. Con CMP, es la herramienta de despliegue la que genera las clases apropiadas en el momento del despliegue asegurando así un alto grado de portabilidad.
- Estas herramientas de despliegue podrán generar código altamente especializado para la base de datos seleccionada. De esta forma los desarrolladores no tienen que aprender APIs específicas para mejorar el rendimiento.
- Con CMP la gestión de la integridad y de las relaciones entre entity beans así como el borrado en cascada es automático.

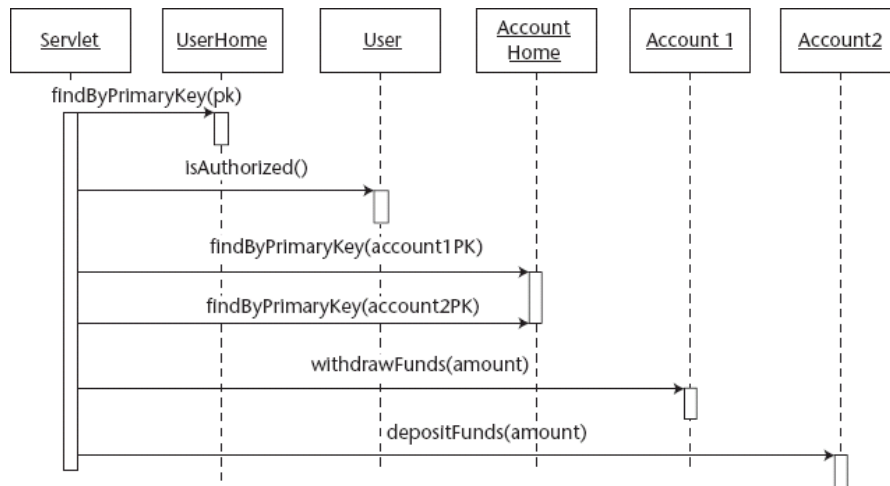
Por supuesto también existen contrapartidas:

- Existe la posibilidad de heredar bugs inherentes a las clases de persistencia generadas por la herramienta de despliegue.
- Con BMP se tiene un mayor control sobre la persistencia.
- La curva de aprendizaje tiene una pendiente más pronunciada en el caso de CMP ya que muchos desarrolladores están familiarizados con las consultas SQL y les resulta más sencillo aprender a implementar BMP entity beans.

#### 3.2.1.1 Session Façade

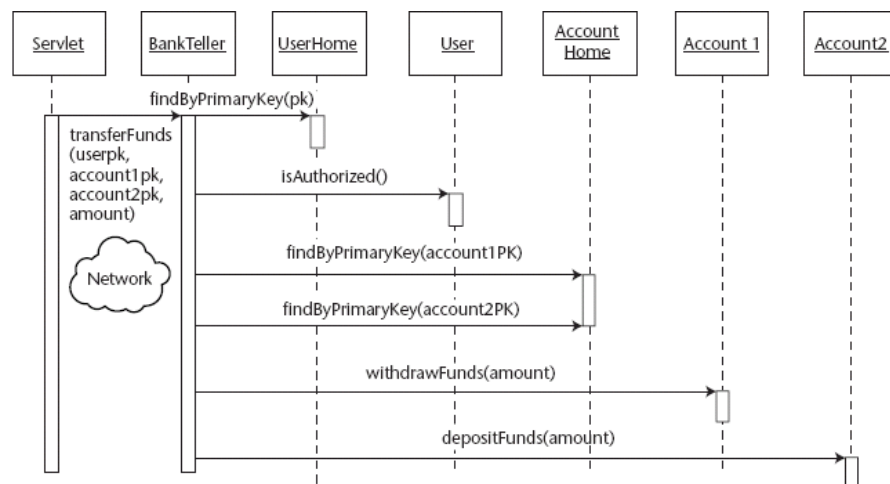
En la ejecución de la lógica que se deriva de los casos de uso, deberemos acceder y modificar varios objetos en el servidor (tales como session beans o entity beans). El problema es que realizar múltiples invocaciones de objetos supone una sobrecarga de llamadas a través de la red.

Pensemos en un banco online en el que un servlet recibe una petición para hacer una transferencia de una cuenta a otra.



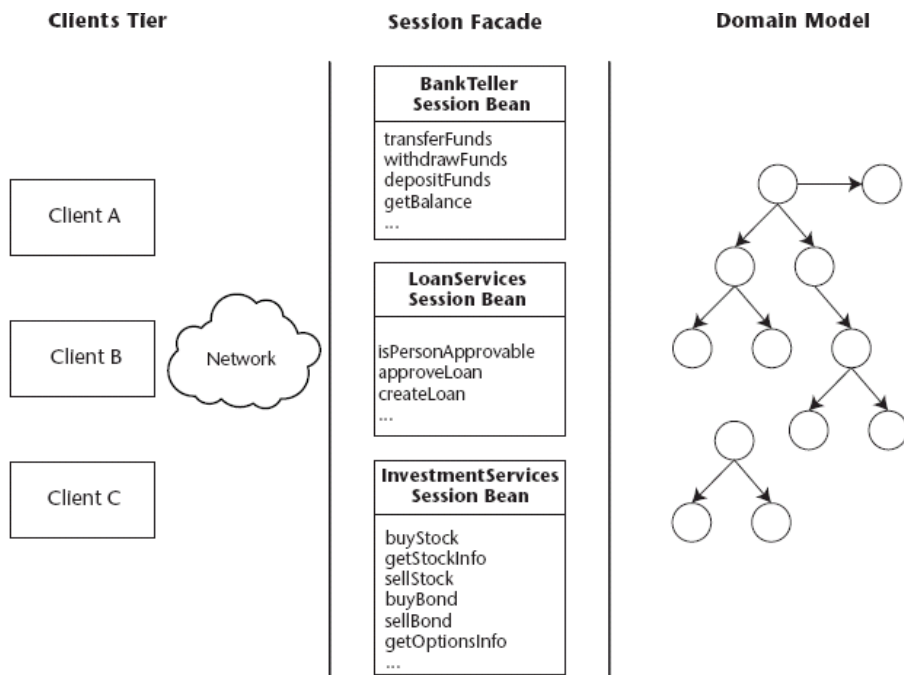
Esta aproximación a la solución del problema no es escalable bajo cargas de trabajo serias, ya que se necesitan al menos seis llamadas a través de la red. Lo que es peor, en esta aproximación no se garantiza la seguridad del dinero del cliente. Si algo va mal con el depósito, el dinero del cliente habrá sido retirado y se perderá.

La solución es poner la lógica para el caso de uso de la transferencia en un session bean, el BankTeller session bean que se sitúa con los bean User y Account y se programa para comunicarse a través de las interfaces locales reduciéndose así la carga de la red a una única llamada.



Este patrón de diseño no sólo proporciona beneficios en el comportamiento de la aplicación, si no que además, sugiere una arquitectura estándar para las aplicaciones J2EE de forma que los límites entre el cliente y el servidor están separados por una capa de session beans, llamada Session Façade, cuyos métodos se corresponden con los casos de uso de la aplicación.

Usando este patrón de diseño se crean session beans que agrupan casos de uso con funcionalidades similares en un solo bean.



Los beneficios del patrón de diseño Session Façade son:

- **Una carga de red baja.**
- **Una separación clara de la lógica de negocio de la lógica de la capa de presentación.** La lógica de negocio está contenida por completo en los métodos de los session bean y los clientes sólo necesitan preocuparse por la lógica de la capa de presentación y nunca deberían ejecutar más de un método en un EJB para realizar una unidad de trabajo.
- **Integridad de las transacciones.** Las transacciones están demarcadas en el nivel de los métodos de los session bean y se pueden configurar en los deployment descriptors.
- **Nivel de acoplamiento bajo.** Los session beans forman un buffer entre las peticiones de los clientes y los entity beans. Si la capa de entity beans necesita cambios en el futuro, podremos evitar cambios en los clientes gracias a la capa de session beans.
- **Una separación clara de verbos-nombres.** La capa de session beans es el modelo para los verbos en nuestra aplicación, mientras que la capa de entity beans modela los nombres. Esta arquitectura hace muy fácil la correspondencia entre los casos de uso del documento de requerimientos y la arquitectura EJB.

El patrón Sesión Façade funciona tan bien que es fácil abusar de él. Los usos incorrectos de este patrón más habituales son:

- **Crear un session bean único.** Poner todos los casos de uso en un único bean da lugar a un bean sobredimensionado que reduce la productividad del equipo de desarrollo por que todos los desarrolladores necesitan acceder a esta única clase. Los session bean deberían dividirse en grupos de casos de uso relacionados.
- **Situar lógica de dominio en session beans.** Un modelo de dominio con un buen diseño orientado a objetos debería contener toda la lógica de negocio de los casos de uso de la aplicación (Fowler, 2001). La mayoría de los métodos de la Session Façade deberían simplemente delegar en el entity bean apropiado a no ser que la lógica del caso de uso dé lugar a un flujo de

trabajo que necesite operar en beans diferentes que no puedan relacionarse directamente.

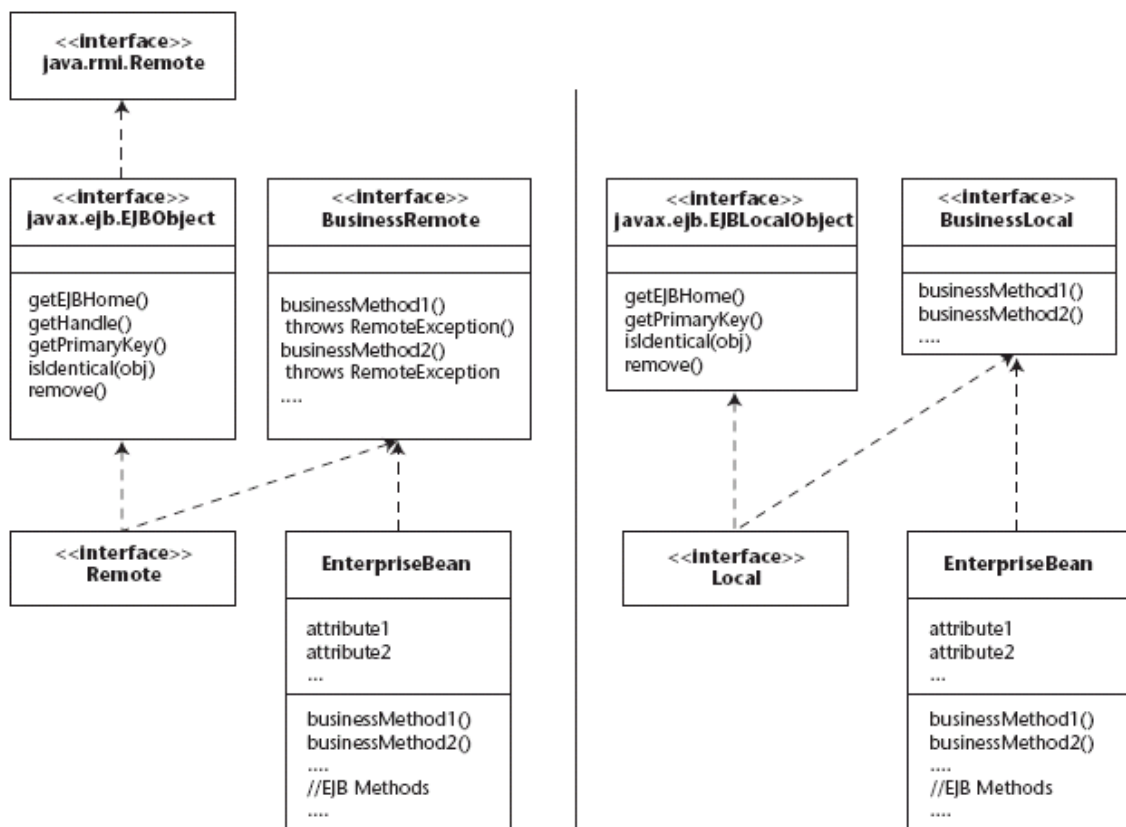
- **Duplicación de la lógica de trabajo en la façade.** A medida que un proyecto crece, es frecuente que métodos de session beans diferentes contengan código duplicado. La solución es añadir una capa de servicios implementados como session beans o como clases Java simples, que encapsulen esta lógica de negocio reutilizable e independiente de un caso de uso en concreto.

En combinación con el patrón de diseño Session Façade utilizaremos otros dos patrones, Business Interface y Data Transfer Object

### 3.2.1.2 Business Interface

Otro patrón de diseño que utilizaremos en esta capa es el conocido como business interface. La especificación EJB obliga a que los enterprise beans implementen todos los métodos declarados en la interfaz local o remota, pero el bean no puede implementar directamente esas interfaces. Esto hace que surja el problema de cómo se pueden descubrir las inconsistencias entre los métodos en las interfaces y los métodos del bean en tiempo de compilación.

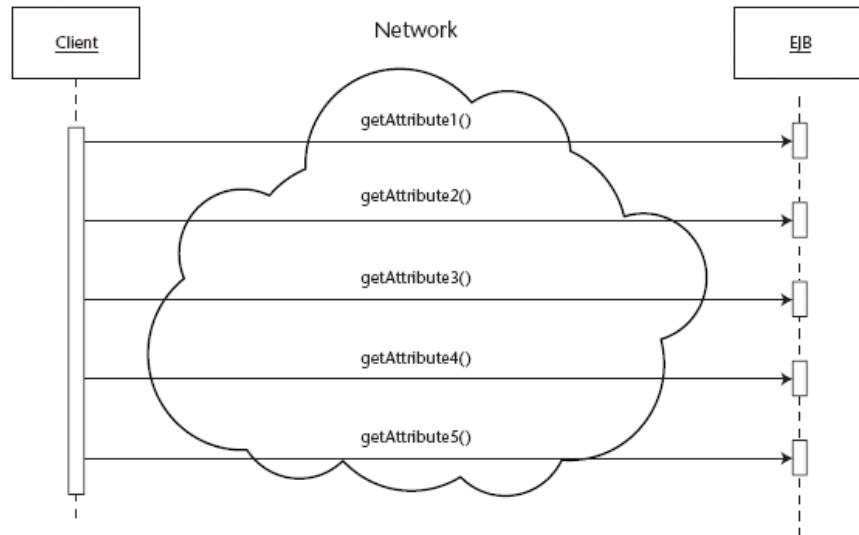
La solución es crear una interfaz llamada business interfaz en la que se definen todos los métodos de negocio. Tanto las interfaces remota y local como el bean implementarán esta interfaz forzándose así el chequeo de la consistencia en tiempo de compilación.



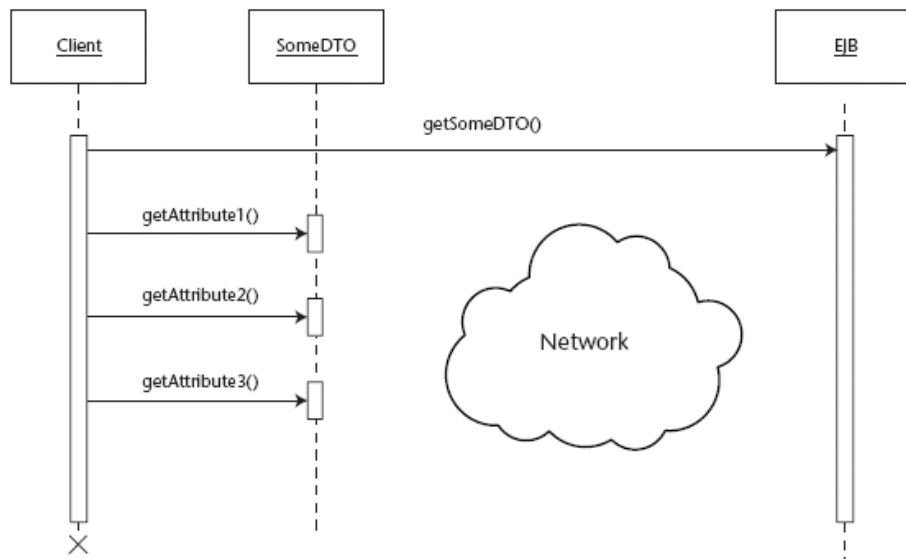
### 3.2.1.3 Data transfer Object

Los clientes en un sistema EJB necesitan una forma de transferir grandes volúmenes de datos desde y hacia el servidor.

Imaginemos que la interfaz de usuario de un cliente necesita mostrar un conjunto de atributos que se encuentran en el servidor. Esos atributos podrían encontrarse en un entity bean o ser accesibles a través de un session bean, una posibilidad sería ejecutar múltiples llamadas al servidor como se muestra en la siguiente figura.



Ejecutar múltiples llamadas de esta forma contribuiría significativamente a la degradación del comportamiento de la aplicación. La solución es crear una clase Java simple que contenga y encapsule el volumen de datos en un único lote transportable a través de la red.



Se pueden distinguir dos casos:

- **Domain Data Transfer Object**

Este tipo de DTO mantiene una correspondencia 1 a 1 con los entity beans del servidor. El diseño de DTO como copias de los entity beans del servidor tiene los siguientes beneficios:

- Las estructuras de datos del modelo de dominio se obtienen en una sola llamada a través de la red.
- Es fácil y rápido construir una aplicación usando este tipo de DTO como intercambio de información.

Los inconvenientes son:



- Acoplamiento entre el cliente y el modelo de dominio del servidor. Si el entity bean cambia, su DTO correspondiente también debe cambiar y por tanto todos los clientes que usen ese DTO también deben cambiar.
- No siempre se adapta bien a las necesidades de los clientes.
- Da lugar a una jerarquía paralela de DTOs y por tanto a una duplicación de atributos y métodos

- **Custom Data Transfer Objects**

Como hemos visto en el apartado anterior, en ocasiones los DDTO no se ajustan a las necesidades de los clientes. En este caso es necesario diseñar DTOs completamente desligados de la capa del modelo de negocio, este tipo de DTO se utiliza habitualmente para operaciones de lectura desde la interfaz de usuario.

### 3.2.2 Capa Web

Se utilizará una arquitectura Model 2 que introduce un servlet controlador entre el navegador y las páginas jsp o el contenido que se envía al cliente. El controlador centraliza la lógica para gestionar las peticiones del cliente y determinar cuál es la próxima página que se debe mostrar. (Model 1 y Model 2 hacen referencia a la presencia o ausencia de este controlador. En el Model 1 toda la lógica residiría en las páginas jsp)

Existen varios marcos de trabajo Web que implementan el modelo 2. Se ha tomado la decisión de utilizar JSF JavaServer Faces™. JSF proporciona:

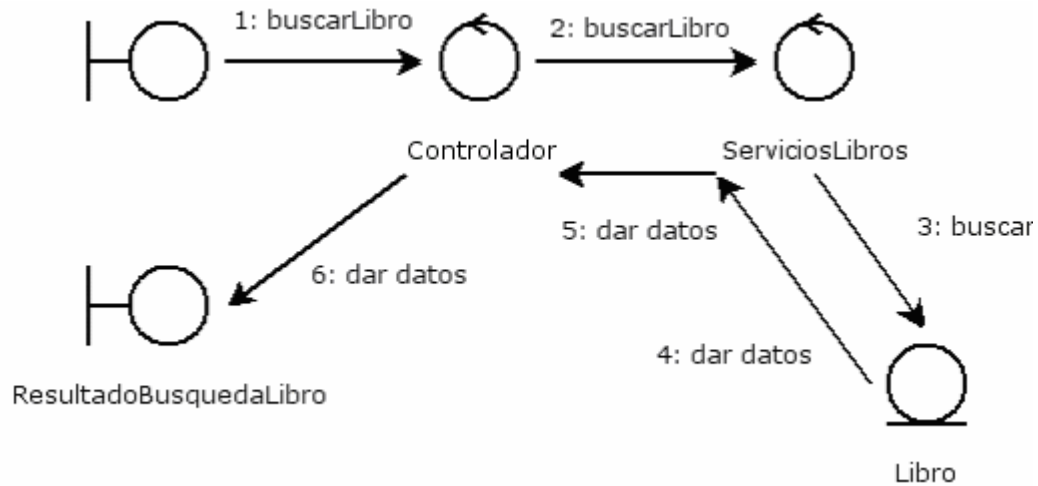
- **Una arquitectura Model-View-Controller** Todas las aplicaciones permiten al usuario manipular determinados datos. Estos datos son el modelo. El desarrollador de software genera una visión o vista de ese modelo, en una aplicación web, se utiliza HTML para generar esa visión. JSF conecta la visión con el modelo, cada componente de la vista se liga con una propiedad de un bean del modelo y JSF hace las veces de controlador que reacciona ante las acciones del usuario y ante los eventos que dan lugar a cambios en los valores de modelo. JSF utiliza el código apropiado en cada caso para actualizar la vista o el modelo.
- **Conversión de datos** Los usuarios introducen datos en formularios web en forma de texto. Los objetos de la lógica de negocio utilizan números, fechas u otro tipo de datos. JSF facilita la elaboración de reglas de conversión
- **Validación y gestión de errores** JSF facilita la asignación de reglas de validación tales como "este campo es obligatorio" o "este campo debe ser un número". Si el usuario introduce datos inválidos JSF se encarga de mostrar los mensajes de error apropiados.
- **Internacionalización** JSF gestiona las tareas de internacionalización tales como la utilización de los recursos adecuados para un lenguaje determinado.
- **Componentes** JSF proporciona un modelo de componentes que permite a terceras partes proporcionar componentes adicionales
- **Visualizadores alternativos** Por defecto, JSF genera páginas HTML pero es fácil extender JSF para generar otro tipo de lenguaje. Por ejemplo, se podría utilizar JSF para comunicarse con teléfonos móviles.
- **Soporte para herramientas** JSF está optimizado para usar con herramientas de automatización de forma que sea posible diseñar interfaces Web arrastrando componentes en una ventana y relacionándolos con objetos Java. Una herramienta de este tipo es Java Studio Creator, que es la herramienta elegida para este proyecto.

### 3.3 Diagramas e interfaz gráfica

#### 3.3.1 Subsistema cliente

##### 3.3.1.1 Caso de Uso Búsqueda

- Diagrama de colaboración

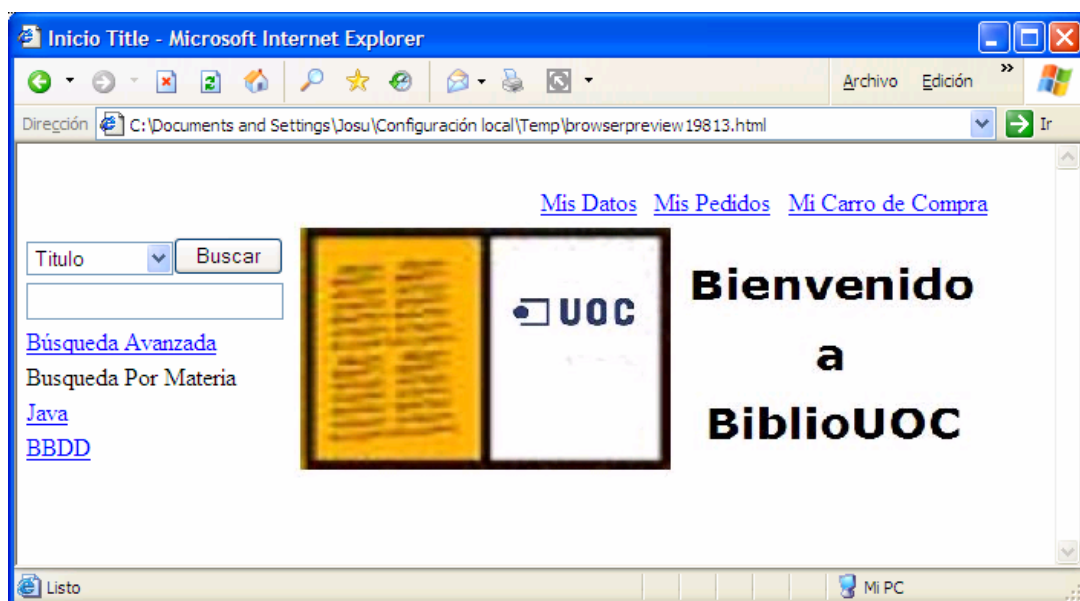


- DTO

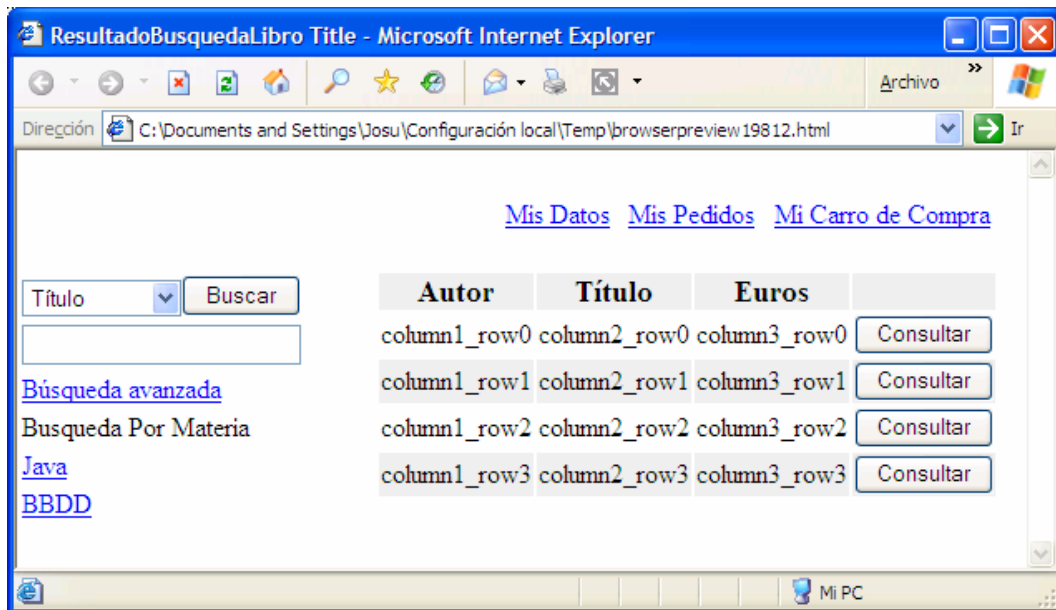
ResumenLibroDTO
-isbn
-autor
-titulo
-Euros

- Pantallas

Las búsquedas pueden realizarse desde la pantalla inicial:

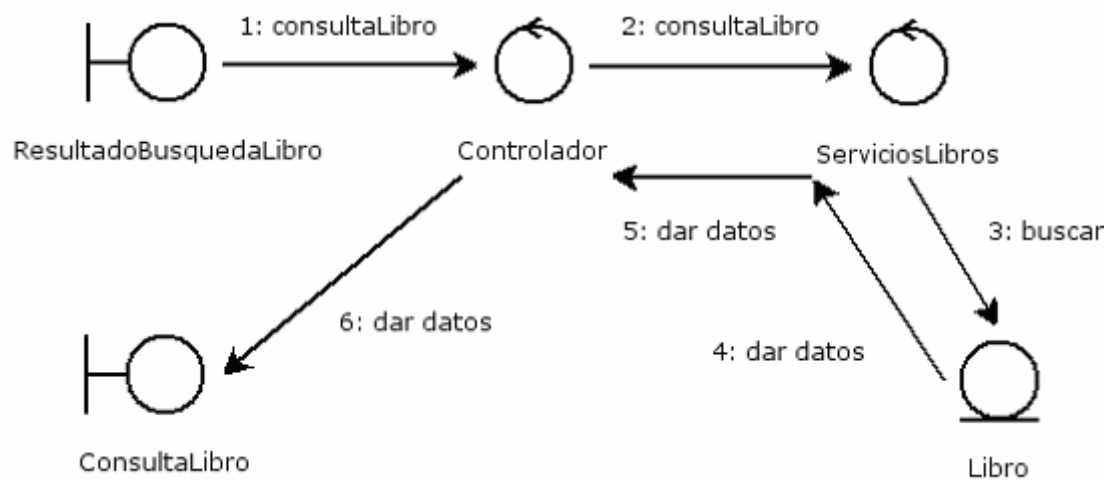


O también puede iniciarse una nueva búsqueda desde la pantalla de resultados de una búsqueda anterior:



### 3.3.1.2 Caso de Uso Consulta Libro

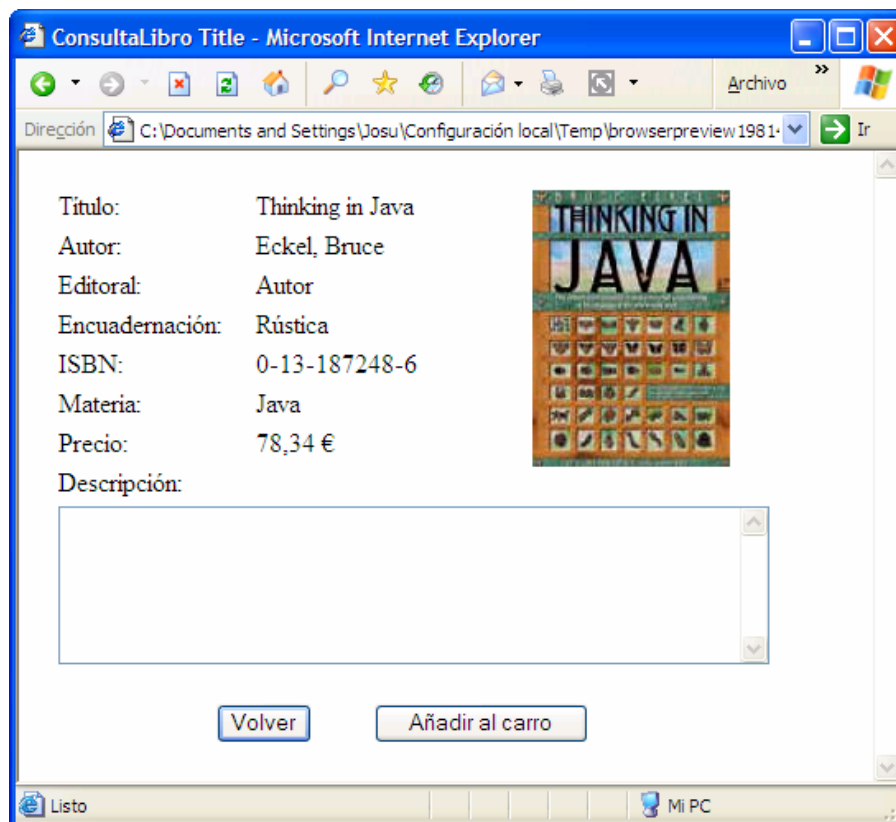
- *Diagrama de colaboración*



- *DTO*

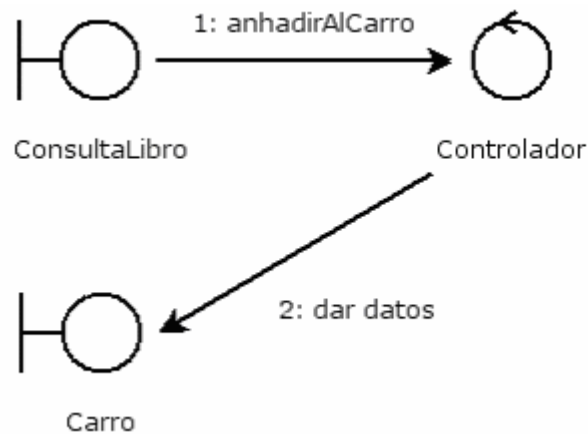
LibroDTO
-isbn
-autor
-titulo
-Euros
-encuadernacion
-materia
-precio
-descripcion
-foto

- *Pantallas*

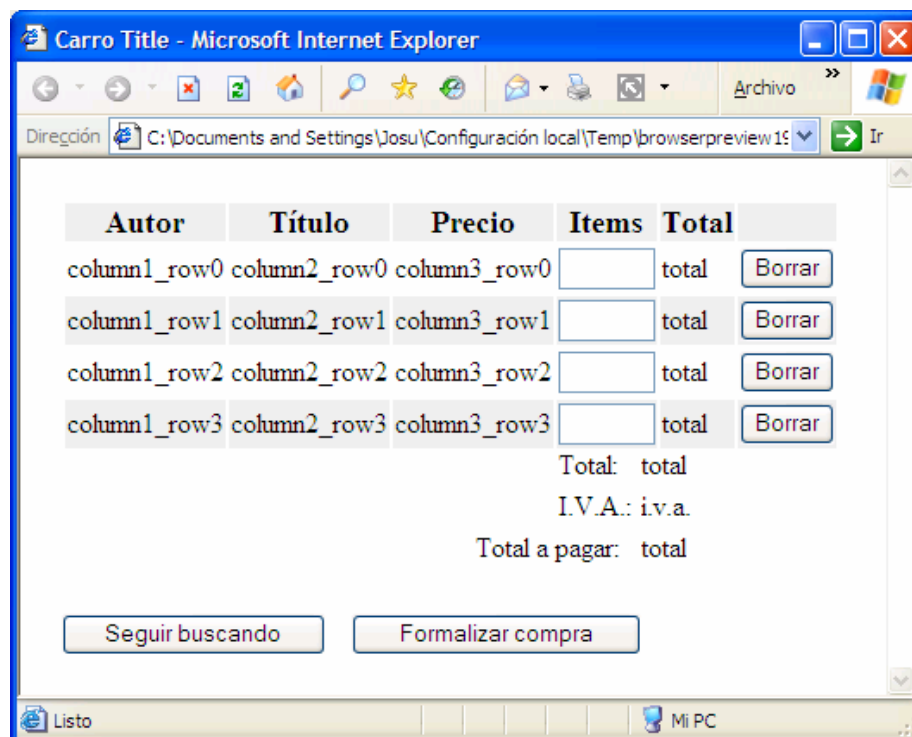


### 3.3.1.3 Caso de Uso Añadir Al Carro

- *Diagrama de colaboración*

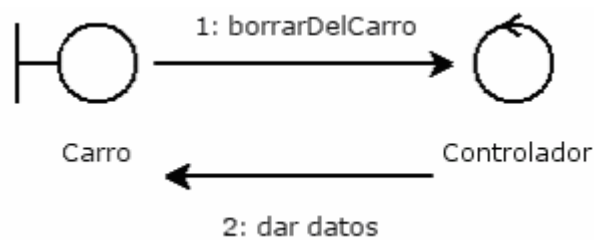


- *Pantallas*



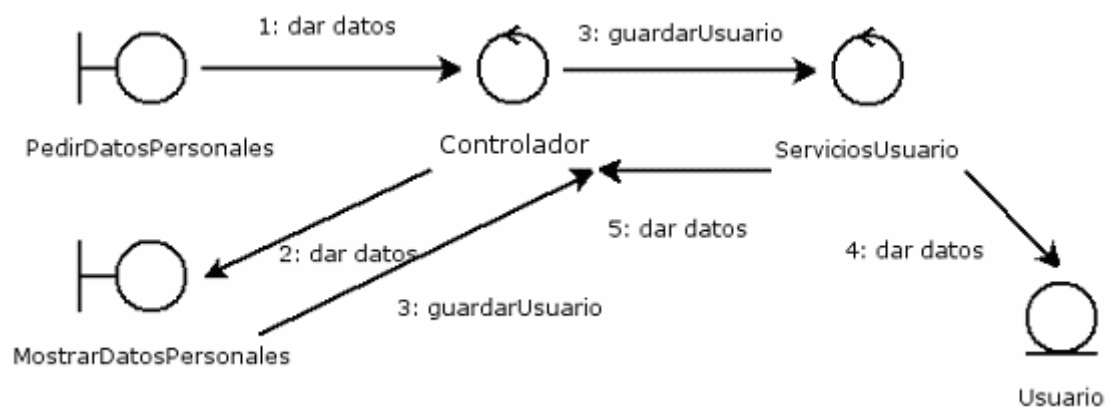
### 3.3.1.4 Caso de Uso Gestionar Carro

- *Diagrama de colaboración*



### 3.3.1.5 Caso de Uso Registro

- Diagrama de colaboración



- DTO

UsuarioDTO
-nombre
-apellido1
-apellido2
-direccion
-poblacion
-codPostal
-provincia
-email
-tlf
-contraseña

- *Pantallas*

PedirDatosPersonales Title - Microsoft Internet Explorer

Dirección <http://localhost:18080/bibliouocliente/> Ir

Por favor introduzca la siguiente información personal

Nombre

1º Apellido

2º Apellido

Dirección

Población

Cod Postal

Provincia

E-mail

Tlf

Contraseña

Verificar Contraseña

Listo Intranet local

ConfirmarRegistro Title - Microsoft Internet Explo...

Dirección <http://localhost:18080/bibliouoccliente/faces/PedirDato> Ir

Confirme sus datos

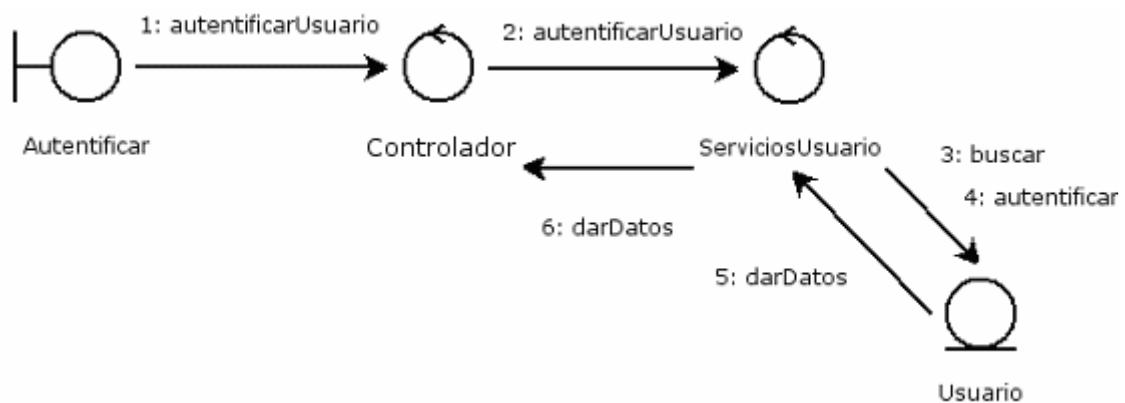
Nombre	José Manuel
1º Apellido	Saavedra
2º Apellido	González
Dirección	Avenida Castrelos
Población	Vigo
Cod Postal	36210
Provincia	Pontevedra
E-mail	jsaavedrag@uoc.edu
Tlf	444444444

Guardar Datos      Modificar Datos

Listo      Intranet local

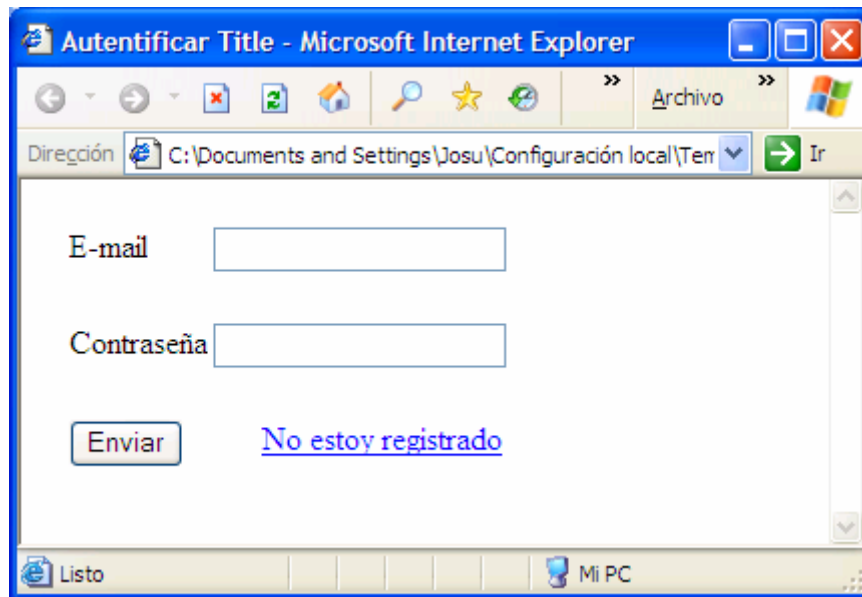
### 3.3.1.6 Caso de Uso Autenticar

- Diagrama de colaboración





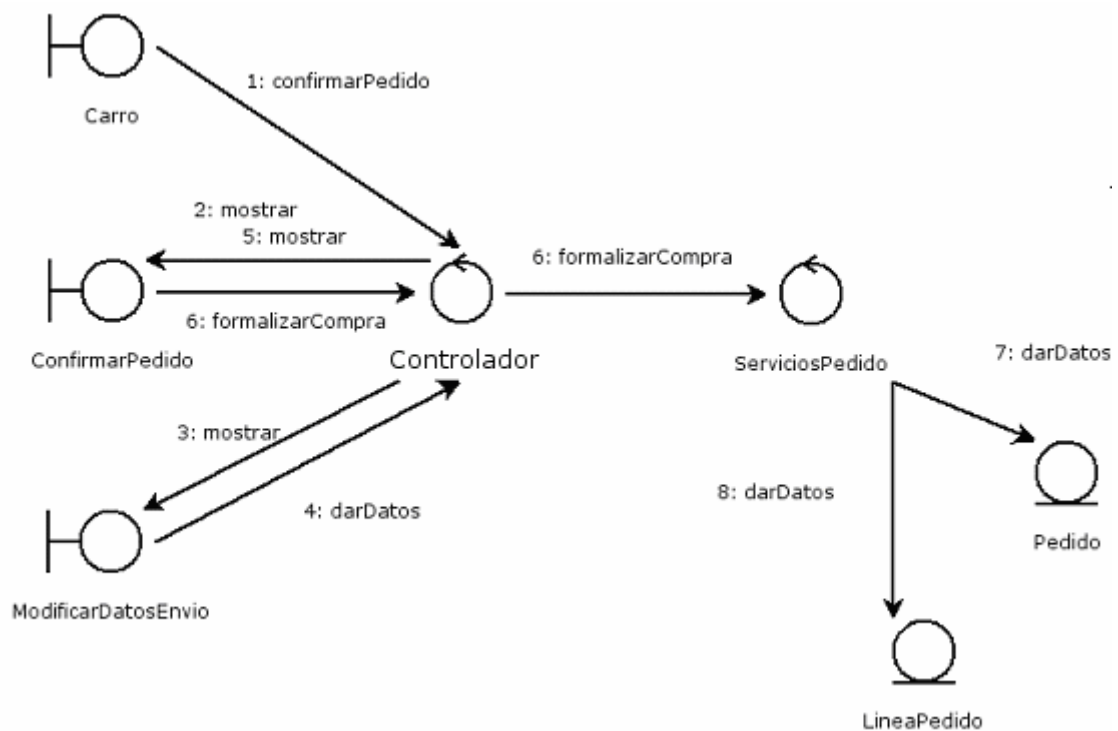
- *Pantallas*



### 3.3.1.7 Caso de Uso Efectuar Compra

El Usuario será autenticado y registrado si es necesario antes de formalizar la compra.

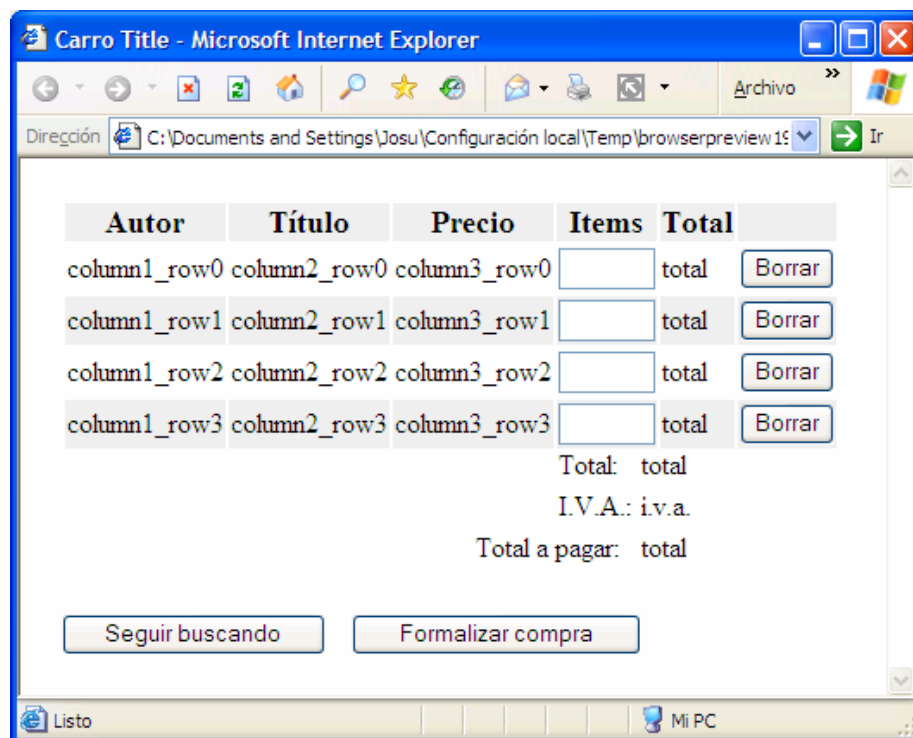
- *Diagrama de colaboración*



- *DTO*

PedidoDTO	LineaPedidoDTO
<ul style="list-style-type: none"> <li>-idpedido</li> <li>-fechaentrada</li> <li>-fechaenvio</li> <li>-fechaentrega</li> <li>-fechacancelacion</li> <li>-direccion</li> <li>-poblacion</li> <li>-codPostal</li> <li>-provincia</li> <li>-email</li> </ul>	<ul style="list-style-type: none"> <li>-items</li> <li>-isbn</li> <li>-idpedido</li> </ul>

- *Pantallas*



ConfirmarPedido Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview49599.html Ir

Autor	Título	Precio	Items	Total
Text	Text	Text	<input type="text"/>	total <input type="button" value="Borrar"/>

Total: total  
I.V.A.: i.v.a.  
Total a pagar: total

**Datos de envío**

Nombre: Text  
1º Apellido Text  
2º Apellido Text  
Dirección Text  
Población Text  
Cod Postal Text  
Provincia Text  
E-mail Text  
Tlf Text

Listo Mi PC

**ModificarDatosEnvio Title - Microsoft Internet Explorer**

Dirección

Población

Cod Postal

Provincia

E-mail

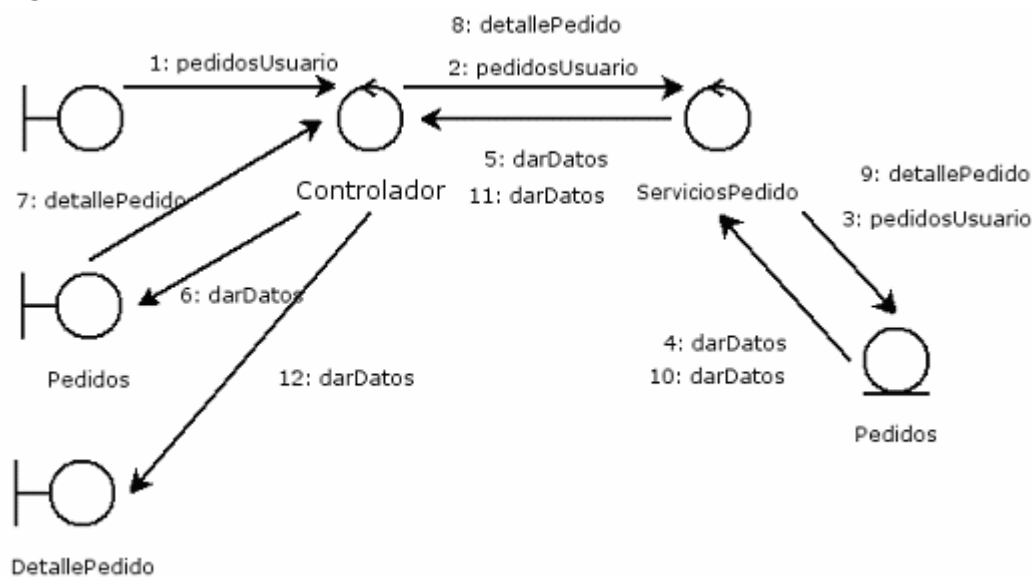
Tlf

Listo Mi PC

### 3.3.1.8 Caso de Uso Consultar Pedidos

El Usuario será autenticado y registrado si es necesario antes de formalizar la compra.

- *Diagrama de colaboración*

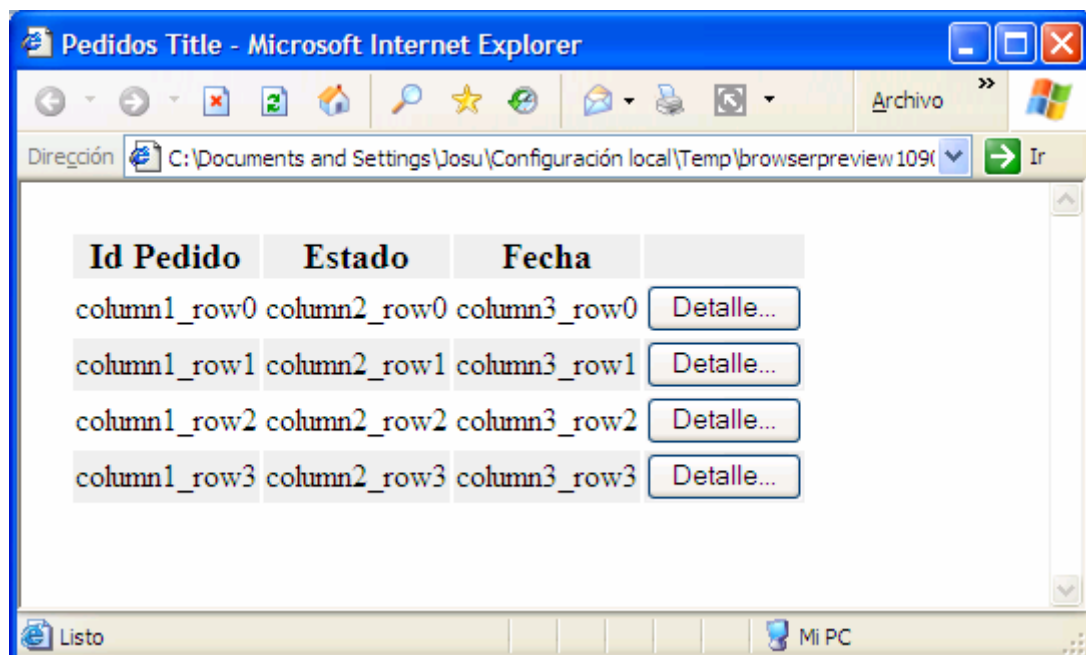


- *DTO*

PedidoDTO
-idpedido
-fechaentrada
-fechaenvio
-fechaentrega
-fechacancelacion
-direccion
-poblacion
-codPostal
-provincia
-email

LineaDetallePedidoDTO
-autor
-titulo
-precio
-items

- *Pantallas*



**DetallePedido Title - Microsoft Internet Explorer**

Dirección: C:\Documents and Settings\Josu\Configuración local\Temp\brows

Autor	Título	Precio	Items	Total
Text	Text	Text	Text	total

Total: total  
I.V.A.: i.v.a.  
Total a pagar: total

**Dirección de envío**

Dirección: Text  
Población: Text  
Cod Postal: Text  
Provincia: Text  
E-mail: Text  
Tlf: Text

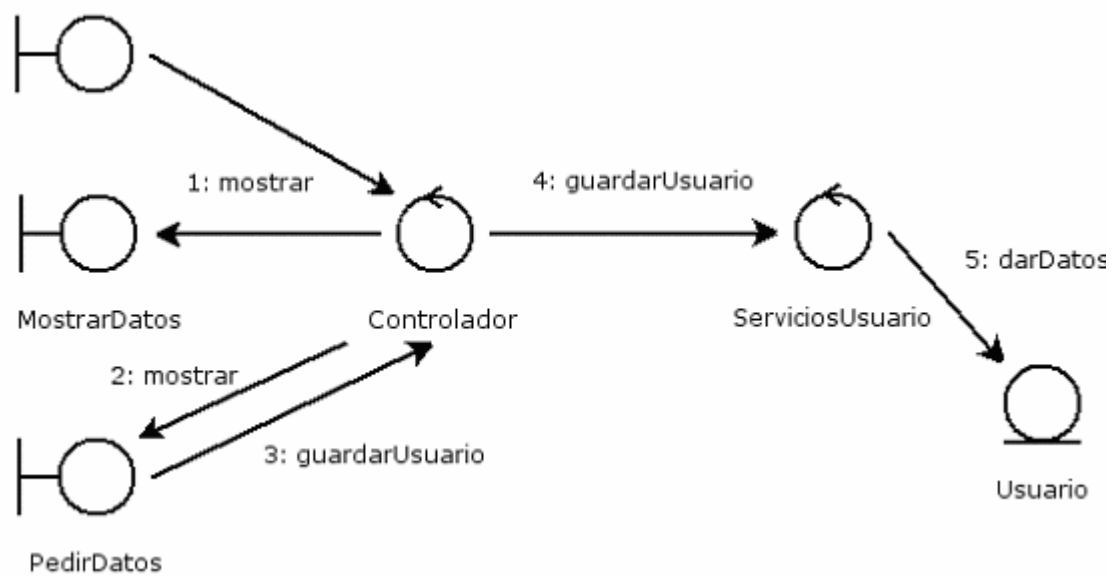
[Volver](#)

Barra de estado: Listo, Mi PC

### 3.3.1.9 Caso de Uso Gestionar Datos

El Usuario será autenticado y registrado si es necesario antes de gestionar los datos

- *Diagrama de colaboración*



- *DTO*

UsuarioDTO
-nombre
-apellido1
-apellido2
-direccion
-poblacion
-codPostal
-provincia
-email
-tlf
-contraseña

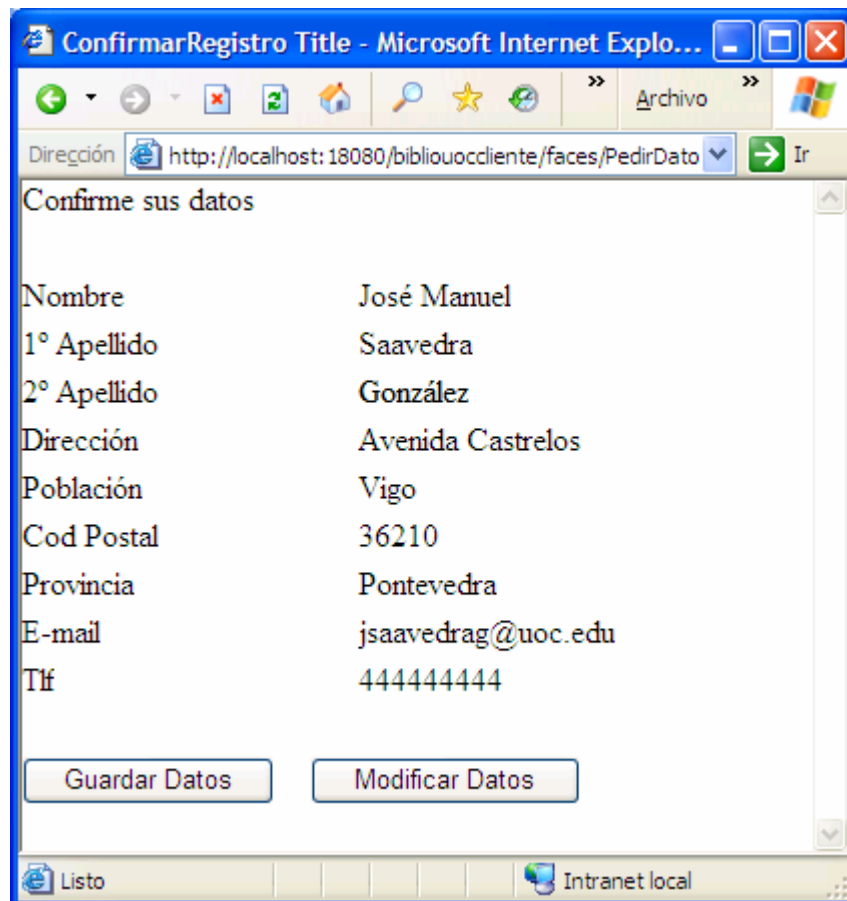
- *Pantallas*

The screenshot shows a Microsoft Internet Explorer window titled "PedirDatosPersonales Title - Microsoft Internet Explorer". The address bar displays "http://localhost:18080/bibliouoccliente/". The page content begins with the instruction "Por favor introduzca la siguiente información personal". Below this, there is a form with the following fields and labels:

- Nombre
- 1º Apellido
- 2º Apellido
- Dirección
- Población
- Cod Postal
- Provincia
- E-mail
- Tlf
- Contraseña
- Verificar Contraseña

At the bottom of the form is a button labeled "Continuar". The browser's status bar at the bottom shows "Listo" and "Intranet local".



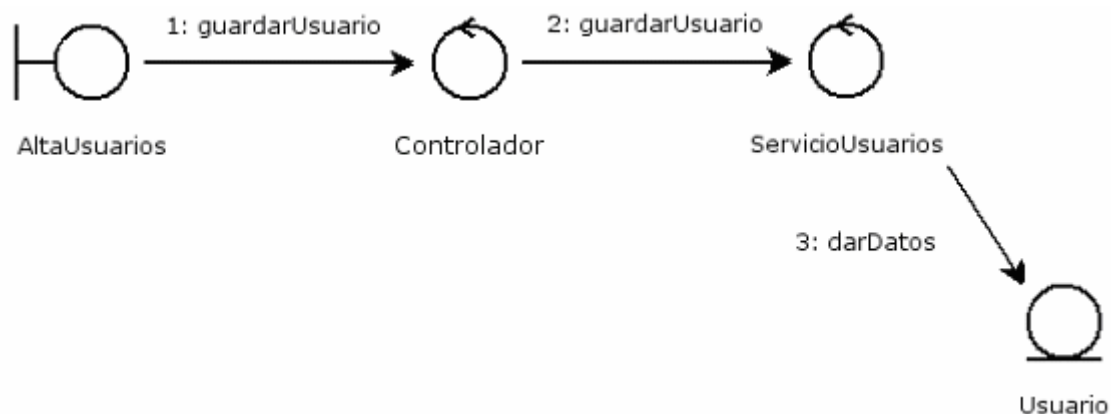


### 3.3.2 Subsistema administrativo

Ya que muchas de las funcionalidades de este subsistema se encuentran ya en el subsistema cliente, se reutilizarán las clases de este subsistema.

#### 3.3.2.1 Caso de Uso Alta Usuarios

##### ▪ Diagrama de colaboración



- *DTO*

UsuarioDTO
-nombre
-apellido1
-apellido2
-direccion
-poblacion
-codPostal
-provincia
-email
-tlf
-contraseña

- *Pantallas*

AltaUsuarios Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview 10908.html Ir

[Usuarios](#) [Libros](#) [Pedidos](#) [Seguridad](#)

[Alta](#) [Modificación](#) [Baja](#)

Introduzca los siguientes datos personales

Nombre

1º Apellido

2º Apellido

Dirección

Población

CodPostal

Provincia

E-mail

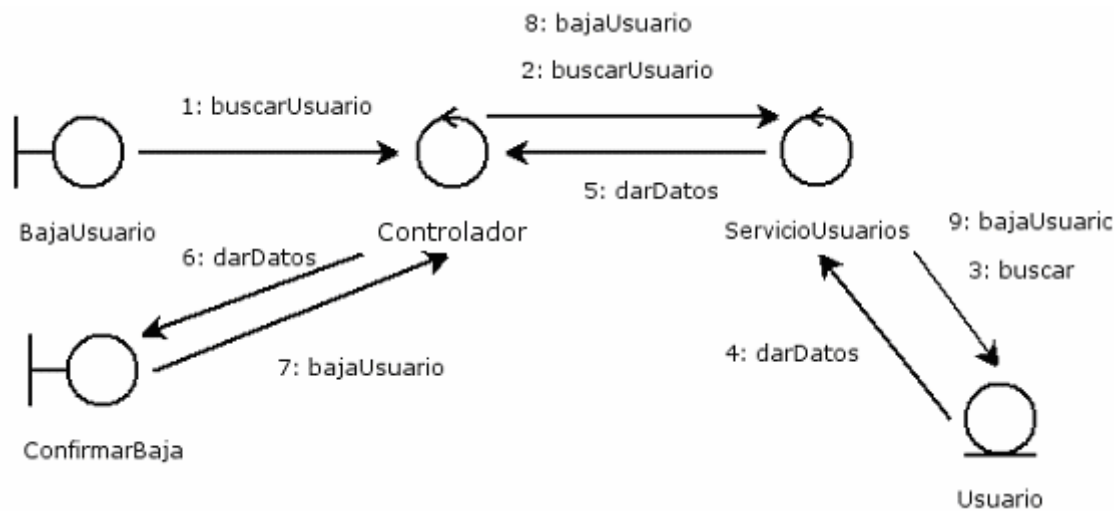
Tlf

Contraseña

Listo Mi PC

### 3.3.2.2 Caso de Uso Baja Usuario

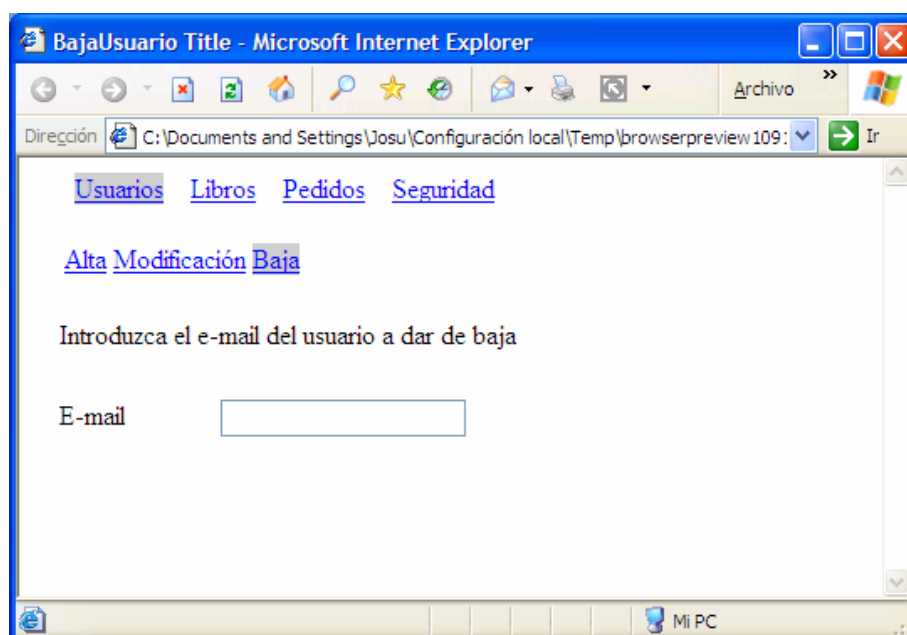
- Diagrama de colaboración

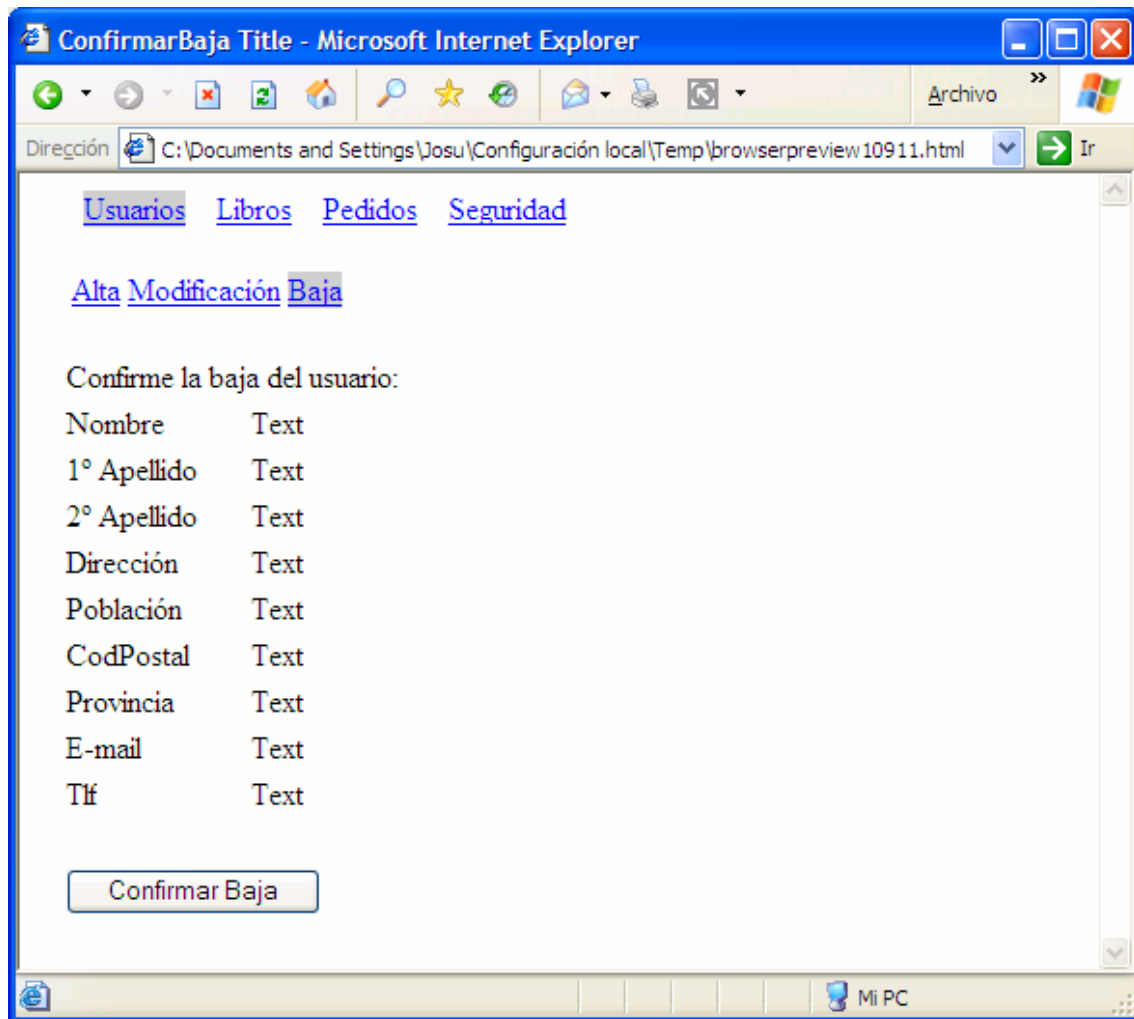


- DTO

UsuarioDTO
-nombre
-apellido1
-apellido2
-direccion
-poblacion
-codPostal
-provincia
-email
-tlf
-contraseña

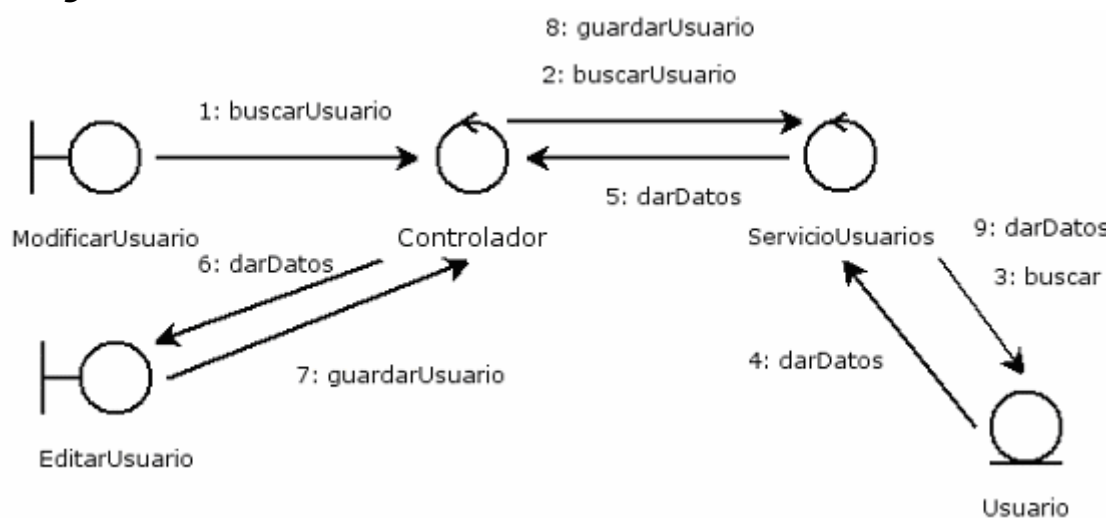
- Pantallas





### 3.3.2.3 Caso de Uso Modificación Usuario

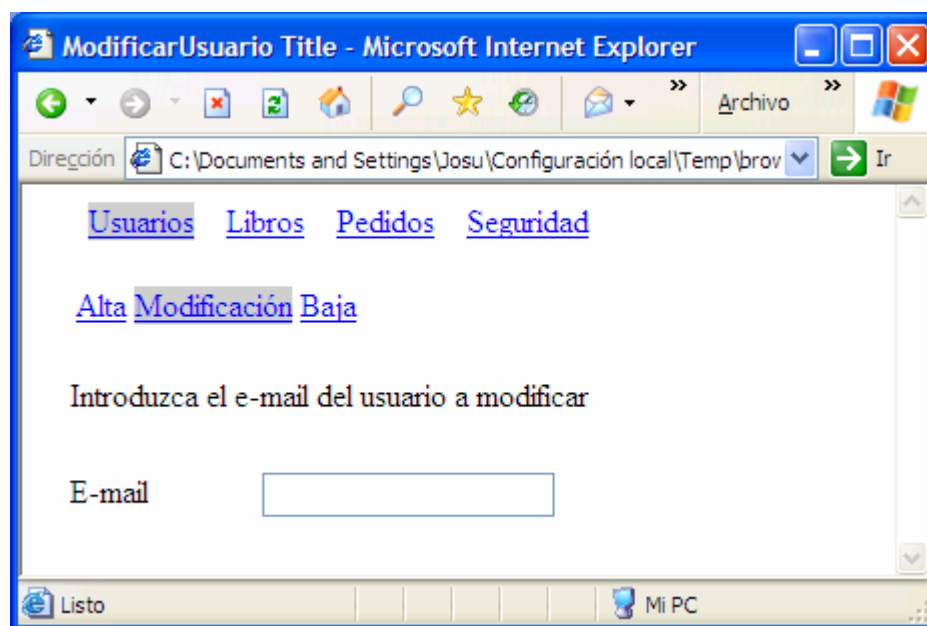
- Diagrama de colaboración



- *DTO*

UsuarioDTO
-nombre
-apellido1
-apellido2
-direccion
-poblacion
-codPostal
-provincia
-email
-tlf
-contraseña

### 3.3.2.4 Pantallas



EditarUsuario Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview10913.ht Ir

[Usuarios](#) [Libros](#) [Pedidos](#) [Seguridad](#)

[Alta](#) [Modificación](#) [Baja](#)

Edite los siguientes datos personales

Nombre

1º Apellido

2º Apellido

Dirección

Población

CodPostal

Provincia

E-mail

Tlf

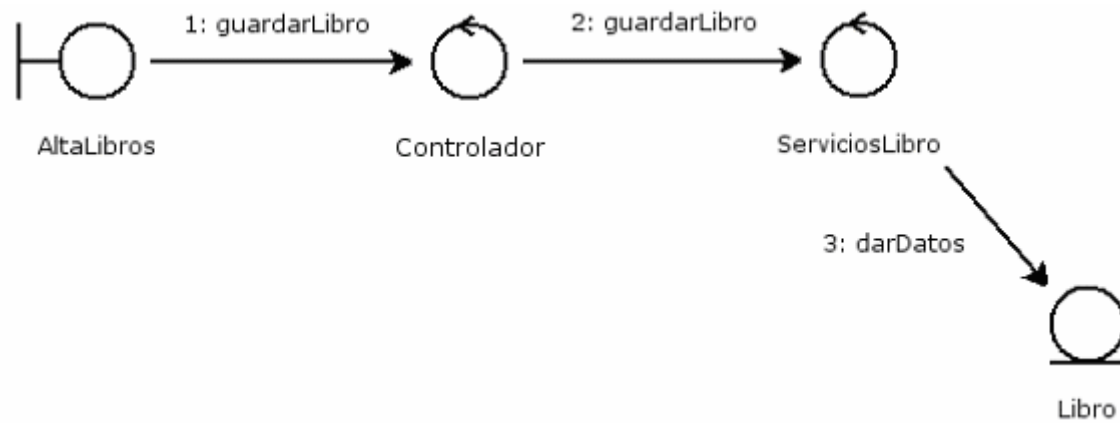
Contraseña

Contraseña

Listo Mi PC

### 3.3.2.5 Caso de Uso Alta Libro

- *Diagrama de colaboración*



- *DTO*

LibroDTO
-isbn -autor -titulo -Euros -encuadernacion -materia -precio -descripcion -foto

- *Pantallas*

The screenshot shows a Microsoft Internet Explorer window titled "AltaLibros Title - Microsoft Internet Explorer". The address bar displays the path: "Dirección | Documents and Settings\Josu\Configuración local\Temp\browserpreview10914.html". The page content includes a navigation menu with links: [Usuarios](#), [Libros](#) (highlighted), [Pedidos](#), and [Seguridad](#). Below this is a sub-menu with links: [Alta](#), [Modificación](#), and [Baja](#). The main heading is "Introduzca los datos del libro". It contains several input fields: "Titulo", "Autor", "Editorial", "Encuadernación", "ISBN", "Materia", and "Precio". At the bottom is a "Descripción" label followed by a large text area for input. The Windows taskbar at the bottom shows the "Mi PC" icon.

AltaLibros Title - Microsoft Internet Explorer

Dirección | Documents and Settings\Josu\Configuración local\Temp\browserpreview10914.html

[Usuarios](#) [Libros](#) [Pedidos](#) [Seguridad](#)

[Alta](#) [Modificación](#) [Baja](#)

Introduzca los datos del libro

Titulo

Autor

Editorial

Encuadernación

ISBN

Materia

Precio

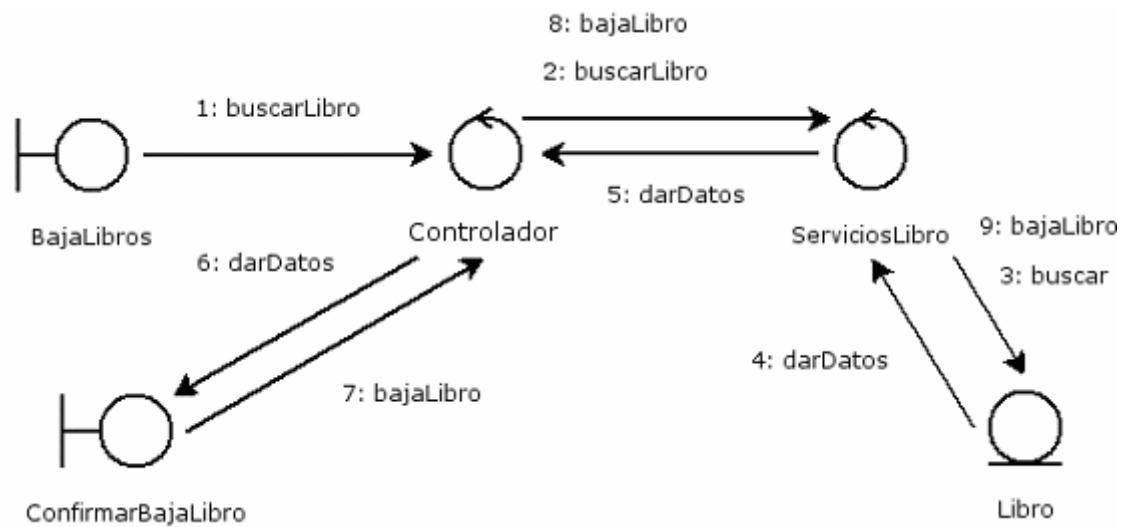
Descripción

Mi PC



### 3.3.2.6 Caso de Uso Baja Libro

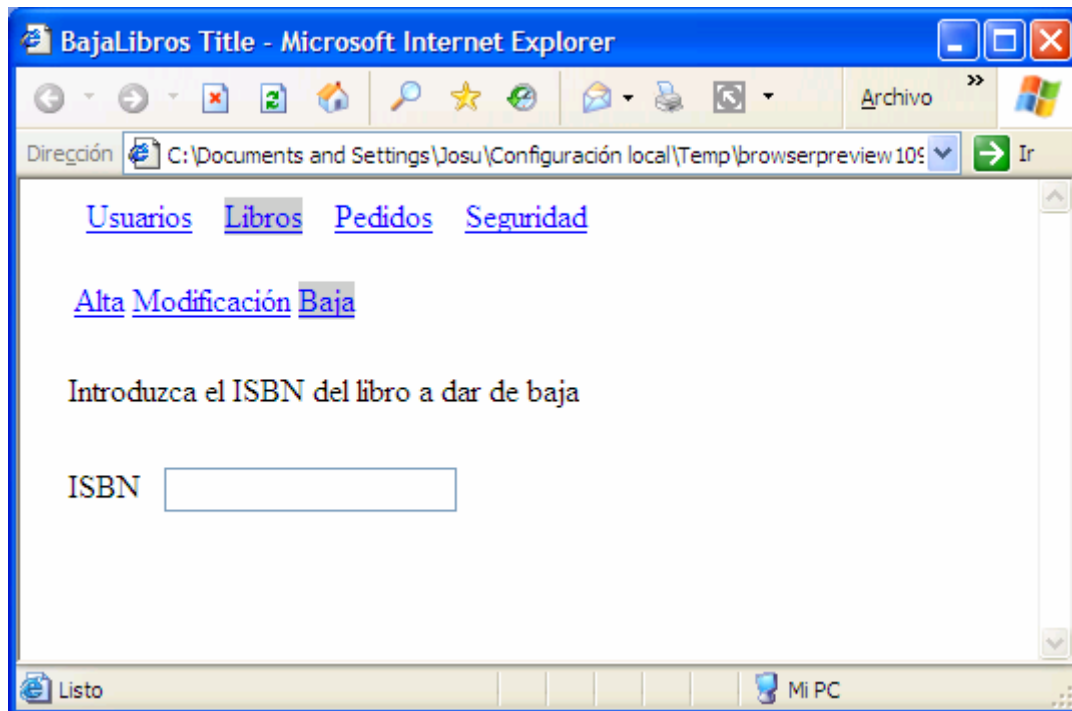
- Diagrama de colaboración



- DTO

LibroDTO
-isbn
-autor
-titulo
-Euros
-encuadernacion
-materia
-precio
-descripcion
-foto

- *Pantallas*



ConfirmarBajaLibro Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview 10916.html Ir

[Usuarios](#) [Libros](#) [Pedidos](#) [Seguridad](#)

[Alta](#) [Modificación](#) [Baja](#)

Confirme la baja del libro:

Título	Text
Autor	Text
Editorial	Text
Encuadernación	Text
ISBN	Text
Materia	Text
Precio	Text

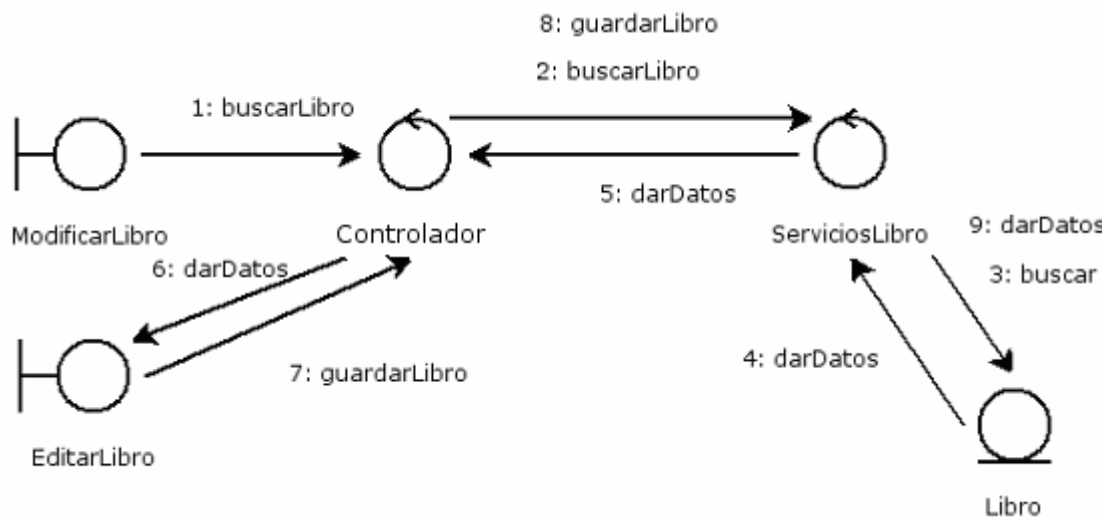
Descripción

Confirmar Baja

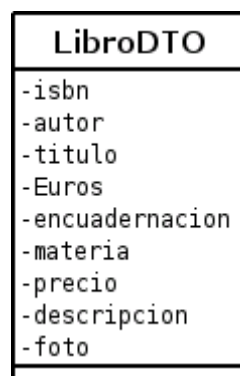
Listo Mi PC

### 3.3.2.7 Caso de Uso Modificar Libro

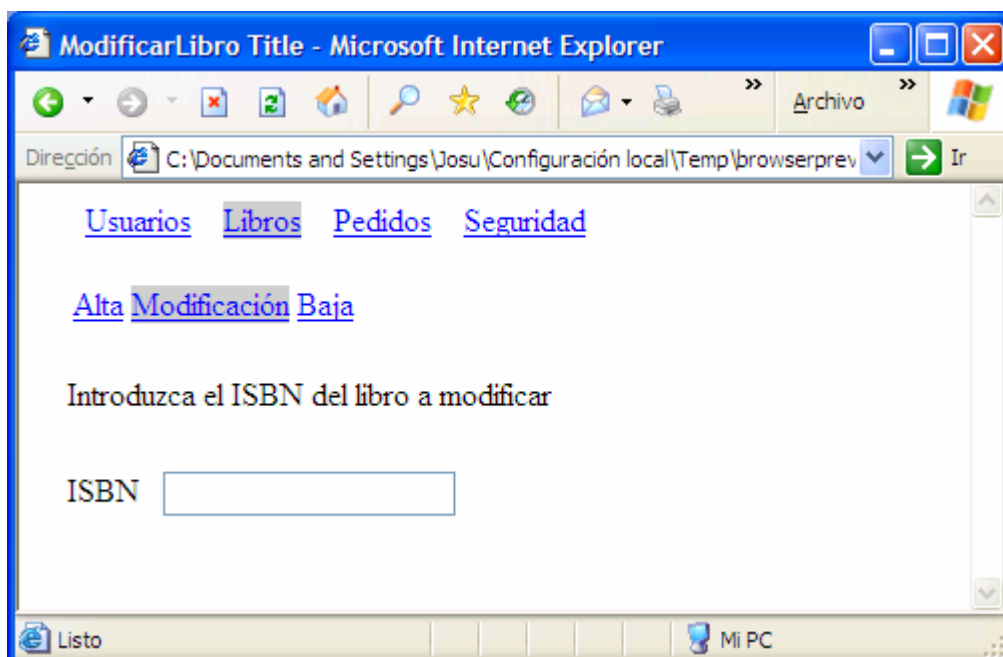
- Diagrama de colaboración



- DTO



- Pantallas



EditarLibro Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview 10' Ir

[Usuarios](#) [Libros](#) [Pedidos](#) [Seguridad](#)

[Alta](#) [Modificación](#) [Baja](#)

Edite los datos del libro

Título

Autor

Editorial

Encuadernación

ISBN

Materia

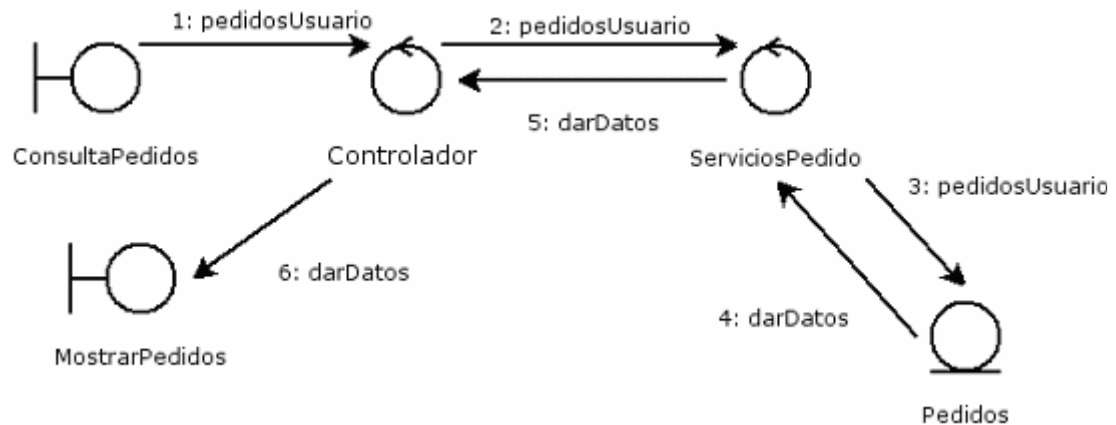
Precio

Descripción

Listo Mi PC

### 3.3.2.8 Caso de Uso Consulta Pedidos

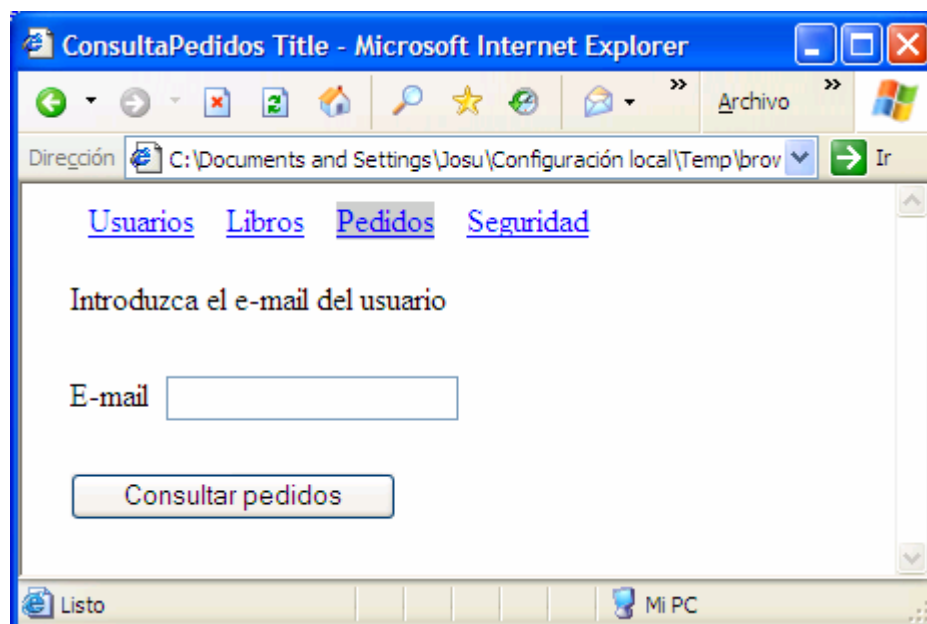
- Diagrama de colaboración

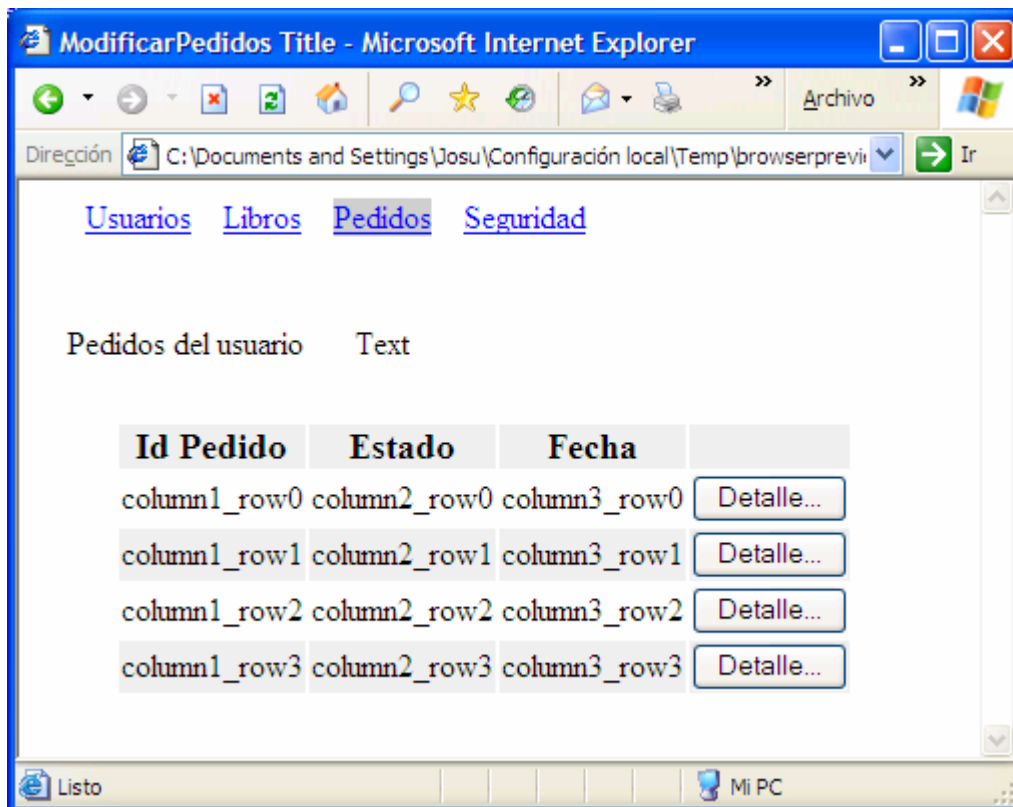


- DTO

PedidoDTO
-idpedido
-fechaentrada
-fechaenvio
-fechaentrega
-fechacancelacion
-direccion
-poblacion
-codPostal
-provincia
-email

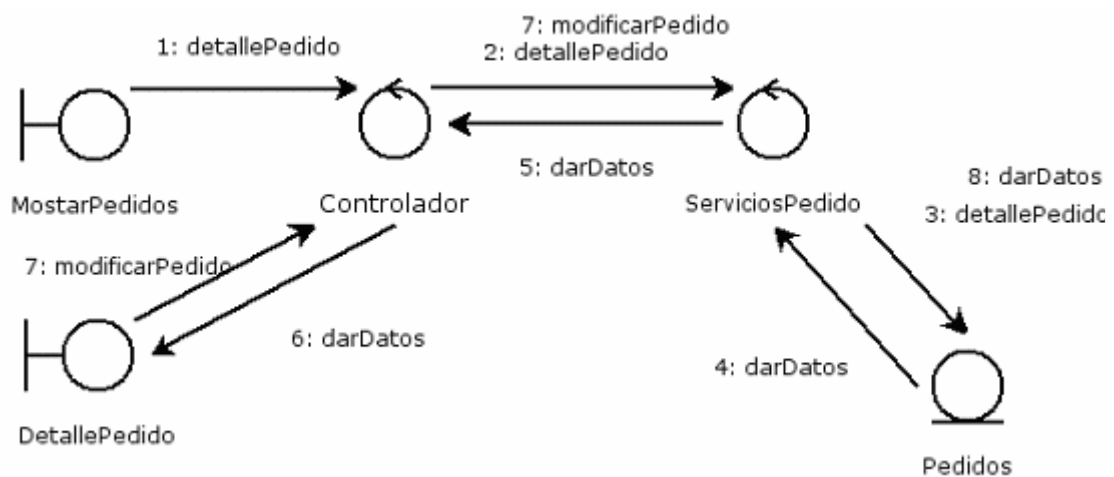
- Pantallas





### 3.3.2.9 Caso de Uso Modificar Pedidos

- Diagrama de colaboración



- DTO

LineaDetallePedidoDTO
-autor
-titulo
-precio
-items

### 3.3.2.10 Pantallas

MostrarDetallePedido Title - Microsoft Internet Explorer

Dirección C:\Documents and Settings\Josu\Configuración local\Temp\browserpreview10922.html Ir

Detalle del pedido:

Id Pedido Text

Fecha entrada  Fecha envío

Fecha entrega  Fecha cancelación

Autor	Título	Precio	Items	Total
column1_row0	column2_row0	column3_row0	Text	Text
column1_row1	column2_row1	column3_row1	Text	Text
column1_row2	column2_row2	column3_row2	Text	Text
column1_row3	column2_row3	column3_row3	Text	Text

Total Text

I.V.A. Text

Total a pagar Text

Dirección de envío

Dirección

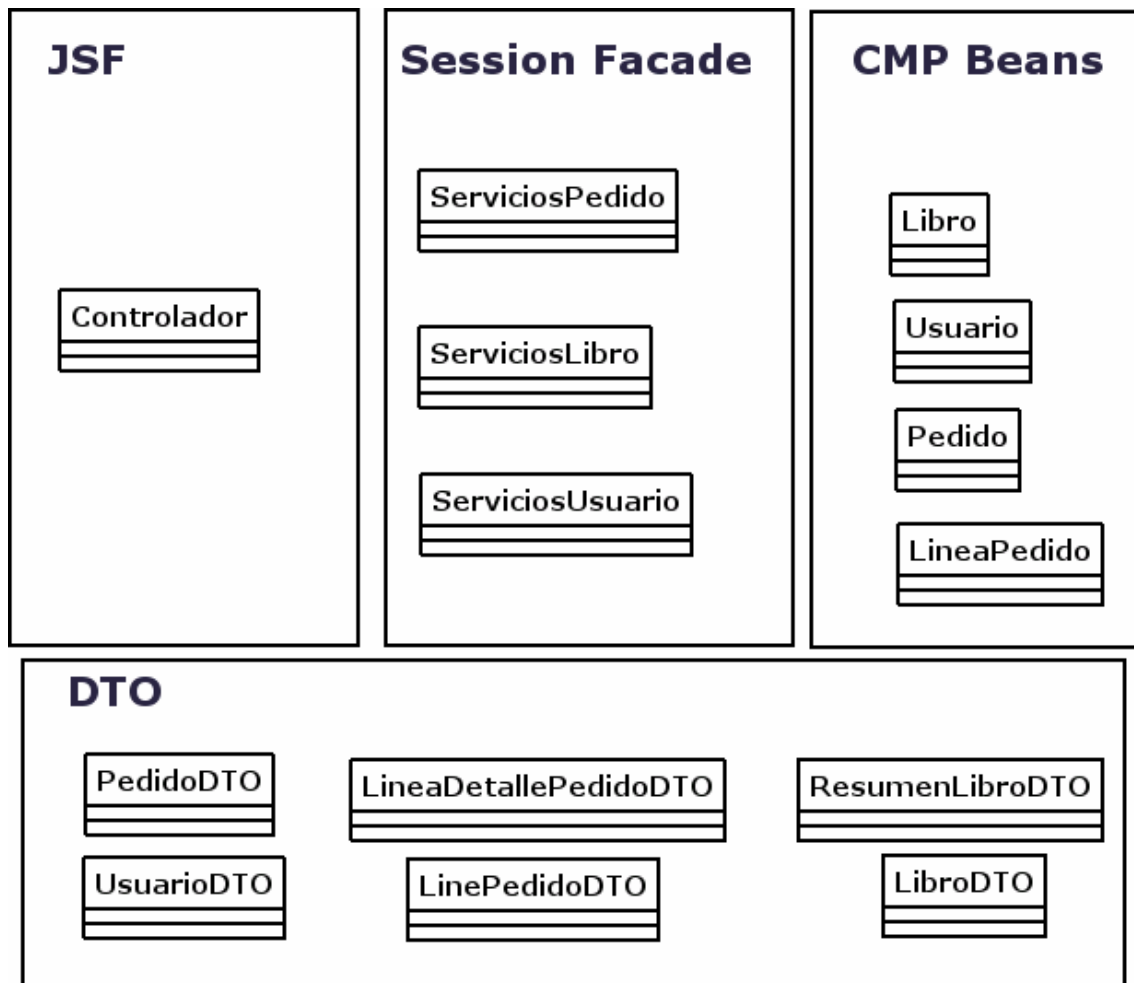
Población

CodPostal

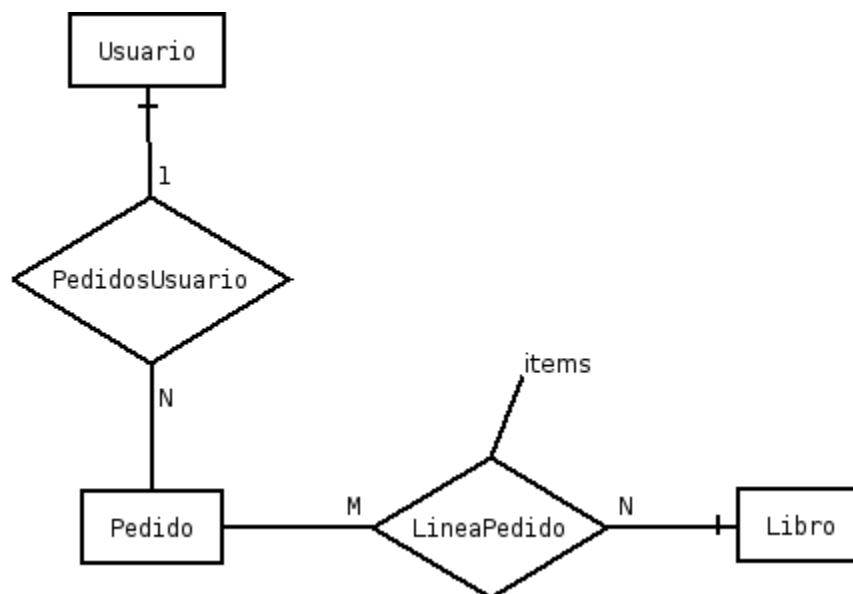
Mi PC

## 3.4 Diagrama De Clases





### 3.5 Diagrama E-R



#### Usuario:

email, nombre, 1apellido, 2apellido, tlf, contraseña, direccion, poblacion, codpostal, provincia

**Pedido:** (email es una clave foránea a Usuario. Se debe tener en cuenta que la dirección de envío no tiene por que coincidir con la dirección del usuario que hizo el pedido)

idpedido, fechaentrada, fechaenvio, fechaentrega, fechacancelacion, direccion, población, codpostal, provincia, email

**Libro:**

isbn, titulo, autor, editorial, encuadernacion, precio, foto, descripcion

**Lineapedido:** (isbn es clave foránea a Libro, idpedido es clave foránea a Pedido)  
ítems, isbn, idpedido

## 4 Implementación

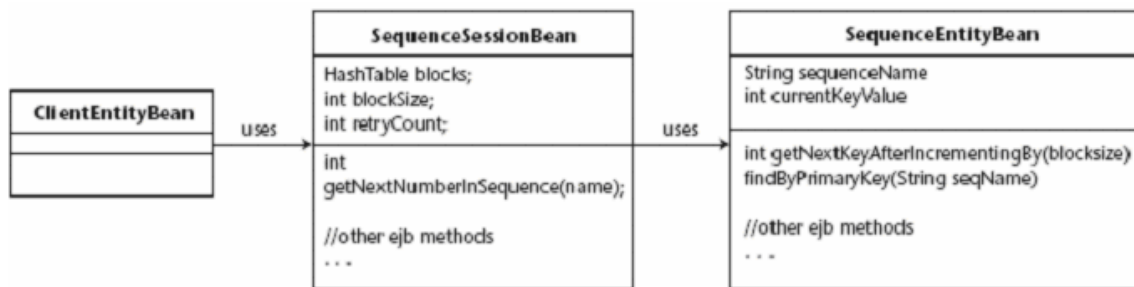
Los datos de la biblioteca se almacenarán en una base de datos MySQL que llamaremos "bibliouoc". Se crearán las tablas: libro, lineapedido, pedido, secuencias y usuario.

La tabla secuencias se utiliza para implementar el patrón de diseño Sequence Block. Se ha decidido recurrir a este patrón ya que aunque el servidor que se utilizará (Sun Java System Application Server) soporta la generación automática de claves primarias tiene limitaciones:

*[...If the database schema is not created during deployment, the primary key column in the mapped table must be of type NUMERIC with a precision of 19 or more, and must not be mapped to any CMP field. The Sun Java System Application Server generates unique values for the primary key column at runtime...]*

<http://www.netbeans.org/kb/41/generated-primary-keys.html>

El patrón **Sequence Block** utiliza un entity bean (SequenceEntityBean) para incrementar un contador que se guarda en una tabla (sequences) en una base de datos. El entity bean se utiliza en combinación con un session bean (SequenceSessionBean) que hace las veces de generador de la clave y que mantiene un bloque de claves en memoria, de esta forma no será necesario acceder al entity bean (y por tanto a la base de datos) cada vez que se necesita una clave primaria.



**SEQUENCES TABLE**

name	value
Account	80
Person	30
Country	100

Cuando un entity bean cliente necesita una clave hace una llamada a **SequenceSessionBean** y solicita la clave con la función *getNextNumberInSequence* el pseudocódigo para esta función es:

1. Comprueba si en el caché local hay un bloque de claves correspondiente a la secuencia solicitada
2. Si no existe o el bloque ya no tiene claves disponibles, entonces **SequenceSession** llama a **SequenceBean** y coge el siguiente bloque de claves disponible para la secuencia.
3. Al coger un nuevo bloque se deben capturar (catch) rollbacks y reintentarlo si es necesario.
4. Finalmente se pasa la clave al entity bean cliente.

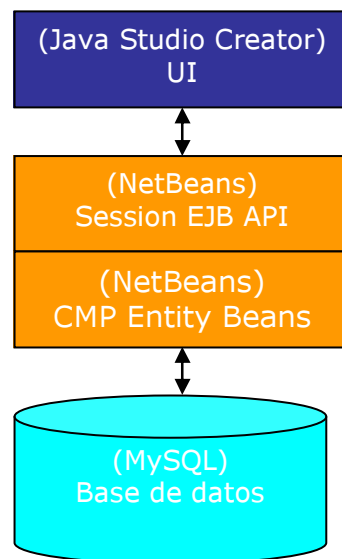
Se utilizarán herramientas distintas para generar la interfaz de usuario y para generar la lógica de negocio y el acceso a la base de datos. El acceso a la base de datos y la lógica de negocio se generará con NetBeans y la UI con Creator.

NetBeans permite construir automáticamente una capa de entity beans con CMP a partir de la base de datos. En versiones no actualizadas de este IDE existe un bug conocido y la configuración resultante debe corregirse manualmente.

*[...When you create CMP beans from a database, the IDE automatically creates a connection pool and data source for the database connection. A known bug in NetBeans 4.1 is that the IDE generates connection pools for MySQL databases with the wrong datasource class name. You have to change the class name yourself...]*

<http://www.netbeans.org/kb/41/generated-primary-keys.html>

Para acceder a la capa de persistencia se usará una capa de Session EJBs que se generará también aprovechando las facilidades de Netbeans. De esta forma se dispondrá de una API que se podrá desplegar y utilizar con Java Studio Creator



Los detalles de la integración de Creator pueden consultarse en la siguiente serie de artículos:

<http://www.developer.com/java/ent/article.php/3489531>

[http://www.developer.com/java/ent/article.php/10933\\_3491091\\_1](http://www.developer.com/java/ent/article.php/10933_3491091_1)

<http://www.developer.com/java/ejb/article.php/3494801>

## 5 Conclusiones

---

J2EE ha establecido un nuevo modelo para el desarrollo rápido de aplicaciones distribuidas. Este modelo se basa en componentes que pueden utilizar los servicios de sofisticadas plataformas, pueden ser desplegados en diferentes servidores y se pueden combinar y reutilizar para una mayor productividad.

Cuando se diseñan aplicaciones distribuidas, los sistemas deben diseñarse de forma que puedan gestionar fácilmente las interacciones de múltiples clientes. Para garantizar la escalabilidad se necesitan mecanismos para una gestión eficiente de recursos y servicios tales como conexiones a bases de datos y transacciones o el soporte de clusters. Para conseguir estos objetivos es fundamental la noción de contenedor. Se pueden distinguir contenedores Web y contenedores EJB.

- Los contenedores Web proporcionan soporte para responder a las peticiones de los clientes, además de la gestión de las sesiones de usuarios.
- Los contenedores EJB proporcionan soporte automático para las transacciones y para el mantenimiento del ciclo de vida de los componentes EJB, así como la búsqueda de beans en directorios y otros servicios.

Los contenedores facilitan también un mecanismo para seleccionar el comportamiento de las aplicaciones en el momento de su despliegue. Mediante el uso de "deployment descriptors" (ficheros XML que especifican el comportamiento del componente), los componentes se pueden configurar para que se ajusten al entorno de un determinado contenedor.

Tanto los EJBs como las páginas JSP han sido diseñadas para ser manipuladas por herramientas gráficas de desarrollo que permiten la automatización de gran parte de las tareas. Muchos proveedores de servidores, así como terceras empresas han desarrollado herramientas que se ajustan a los estándares de J2EE. De esta forma existe la libertad de elección de servidores herramientas y componentes.

J2EE goza de otras cualidades como la integración con los sistemas de información existentes, un modelo flexible de seguridad o la posibilidad de que los servidores pueden ejecutarse en múltiples sistemas hardware o software.

## 6 Guía de instalación y prueba

---

En primer lugar deben crearse las tablas en la base de datos "bibliuoc", para esto puede utilizarse el script "tablas.sql". Además la tabla secuencias debe contener un registro que puede crearse con el script "insert.sql"

Una vez creada la base de datos se despliega el módulo EJB biblioUOC.jar. El servidor utilizado para probar el producto fue Sun Java™ System Application Server Platform Edition 8, la configuración de este servidor para acceder a la base de datos bibliuoc es como sigue:

1. Instalación del driver JDBC: El driver puede descargarse de:

<http://dev.mysql.com/downloads/connector/j/3.1.html>

Una vez descargado debe hacerse accesible en el servidor. Los pasos a seguir para Sun System Application Server pueden consultarse en su página web:

<http://docs.sun.com/source/817-6088/jdbc.html#wp1018096>

2. Creación de un Connection Pool: Los parámetros para este Connection Pool son:

- a. **Name:** mysqlConnectionPool
- b. **Data Source Class Name:**  
com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
- c. **Resource Type:** javax.sql.DataSource
- d. **URL:** jdbc:mysql://host:3306/bibliuoc
- e. **Password:** Password
- f. **User:** User

Se puede consultar una descripción más detallada en:

<http://docs.sun.com/source/817-6088/jdbc.html#wp1017956>

3. Creación de un recurso JDBC:

- a. **JNDI Name:** jdbc/mysql
- b. **Pool Name:** mysqlConnectionPool

<http://docs.sun.com/source/817-6088/jdbc.html#wp1019258>

Para probar el módulo biblioUOC.jar puede ejecutarse la aplicación clienteBiblio.jar. Esta aplicación llamará las funciones del módulo EJB para comprobar su correcto funcionamiento e insertará en la base de datos tres libros y un usuario que podrán utilizarse desde las aplicaciones cliente bibliuoccliente.war y bibliuocadmon.war. Estos clientes deben desplegarse en alguno de los siguientes servidores:

- Sun Java System Application Server 8
- Apache Tomcat 5.x
- BEA Weblogic 8.1
- IBM Websphere Application Server 5