

Ingeniería de sistemas telemáticos

Proyecto de curso

1. Primera fase: *inception*

Esta primera fase se realiza en un único ciclo. Partimos de los siguientes requisitos funcionales:

- El cliente accede a la tienda con un navegador
- Catálogo de libros por temas, con autor, precio, foto portada, índice...
- Buscador de libros
- El cliente puede recopilar libros en un carro de la compra, siempre editable
- Los clientes se pueden registrar y hacer login
- Al hacer login se puede hacer un pedido y pagar con tarjeta de crédito
- Se puede añadir críticas de los libros comprados
- Consulta de pedidos pendientes, y cancelación de los no procesados
- Un encargado de envíos puede consultar los pedidos pendientes y procesarlos
- Al procesar un pedido, se cobra, actualiza en inventario y envía un e-mail
- Un encargado de inventario puede consultar y editar el inventario
- Un encargado de la tienda puede consultar y editar el catálogo
- Estadísticas de las compras realizadas
- Actualización automática de los balances contables
- Un administrador da de alta a los encargados de la tienda
- El acceso al sistema se define por medio de roles de seguridad que se asignan a grupos de usuarios.

De estos requisitos se identifican los actores y casos de uso, que da lugar al modelo general de casos de uso de la figura 1. Este es el principal resultado de la fase de *inception*, y sirve de referencia para planificar las siguientes fases del proyecto.

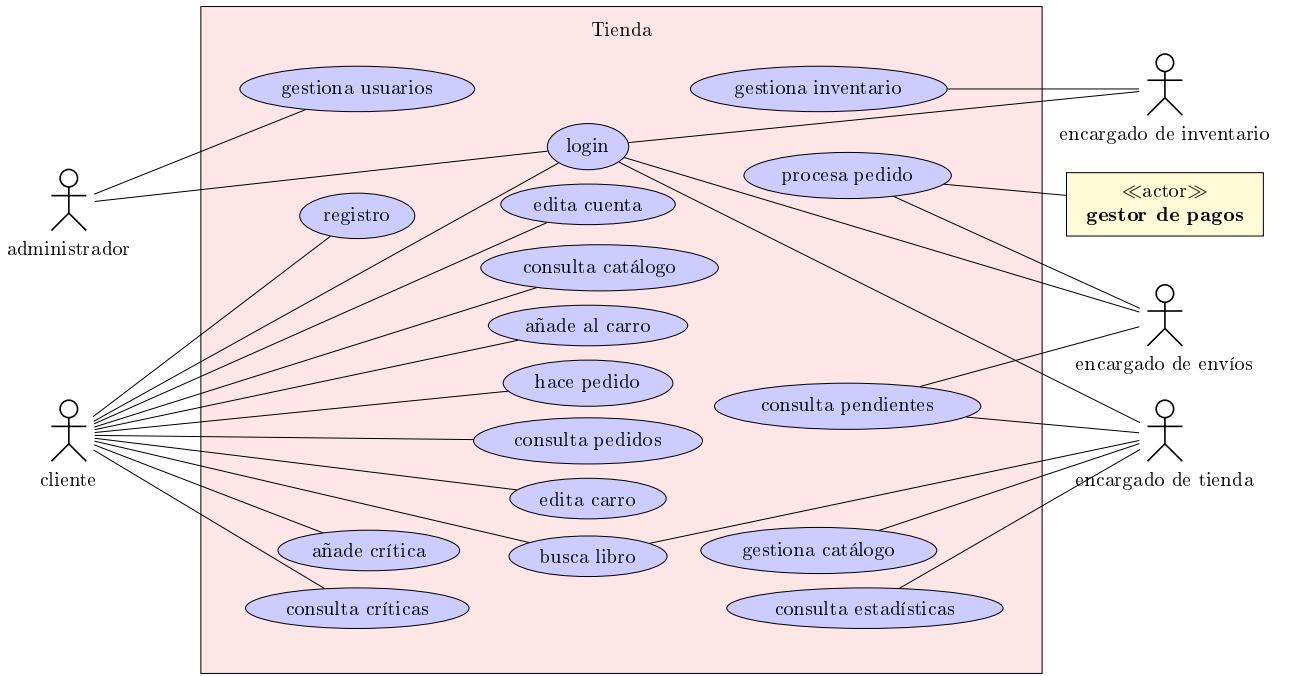


Figura 1: Modelo general de casos de uso

2. Segunda fase: *elaboration*

En esta segunda fase se desarrolla la ‘arquitectura ejecutable’ del sistema, que consiste en un conjunto de componentes que implementan la funcionalidad básica del sistema. Esta fase se puede desarrollar en varios ciclos, pero en este proyecto se hará en un único ciclo. El primer paso consiste en especificar la funcionalidad que implementará la arquitectura ejecutable. Esto se hace eligiendo un subconjunto de los casos de uso y actores del modelo general. En este caso, la funcionalidad elegida es la de la figura 2. Sobre estos casos de uso se realizan las actividades del ciclo: requisitos, análisis, diseño, implementación y prueba.

2.1. Requisitos

La captura de requisitos consiste en especificar de forma expandida los casos de uso del ciclo y estructurar los casos de uso determinando la dependencias que hay entre ellos. Las especificaciones expandidas son las de los cuadros 1, 2, 3, 4, 5, 6 y 7, y el diagrama de casos de uso estructurado es el de la figura 3.

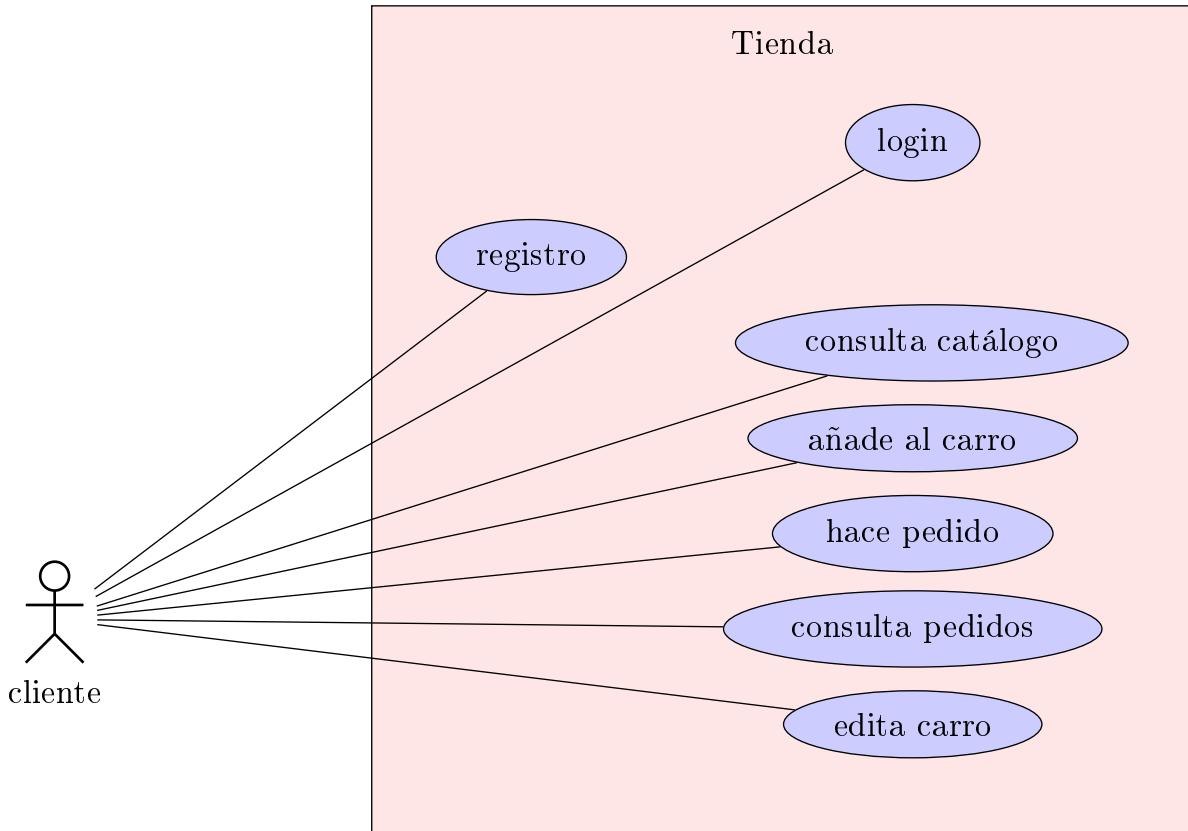


Figura 2: Casos de uso de la arquitectura ejecutable

consulta catálogo	
Actores:	cliente(iniciador)
Precondiciones:	Ninguna
Secuencia de eventos típica	
Acción de un actor	Acción del sistema
1. El cliente selecciona la opción 'catálogo'	2. El sistema presenta una lista de temas
3. El cliente selecciona un tema	4. El sistema presenta una lista de libros y la opción 'catálogo'
5. El cliente selecciona un libro	6. El sistema presenta los detalles del libro y la opción 'tema anterior'
<i><añade al carro></i>	
Secuencias alternativas	
Acción 5: Si el cliente selecciona 'catálogo', vuelve a 2	
Postcondiciones	
Ninguna	

Cuadro 1: Especificación expandida de 'consulta catálogo'

añade al carro	
Actores:	cliente(iniciador)
Precondiciones:	El sistema está presentando los detalles de un libro en estado ‘disponible’
Secuencia de eventos típica	
Acción de un actor	Acción del sistema
1. El cliente selecciona la opción ‘añadir al carro’	2. El sistema añade el libro al carro e inicia ‘edita carro’
Secuencias alternativas	
Ninguna	
Postcondiciones	El libro queda añadido al carro. Si el libro ya estaba en el carro, se incrementa la cantidad.

Cuadro 2: Especificación expandida de ‘añade al carro’

edita carro	
Actores:	cliente(iniciador)
Precondiciones:	Ninguna
Secuencia de eventos típica	
Acción de un actor	Acción del sistema
1. El cliente selecciona ‘edita carro’	2. El sistema presenta el contenido del carro, incluyendo la lista de libros y el importe total
<i><hace pedido></i>	
Secuencias alternativas	
Ninguna	
Postcondiciones	Ninguna

Cuadro 3: Especificación expandida de ‘edita carro’

hace pedido	
Actores:	cliente(iniciador)
Precondiciones:	El carro de la compra no está vacío
	Secuencia de eventos típica
Acción de un actor	Acción del sistema
1. El cliente selecciona la opción ‘hacer pedido’	2. El sistema presenta una descripción del pedido que incluye la lista de artículos del carro de la compra y el precio total, y las opciones ‘confirmar’ y ‘cancelar’
3. El cliente selecciona ‘confirmar’	4. El sistema informa que el pedido se ha registrado e inicia ‘consulta pedidos’
Secuencias alternativas	
Acción 2: Si el cliente no ha hecho login, inicia ‘login’	
Acción 3: Si el cliente selecciona la opción ‘cancelar’, se interrumpe el caso de uso	
Postcondiciones	El pedido queda registrado y anotado como ‘pendiente’ El inventario queda actualizado El carro de la compra se vacía

Cuadro 4: Especificación expandida de ‘hace pedido’

consulta pedidos	
Actores:	cliente(iniciador)
Precondiciones:	El cliente ha hecho login
	Secuencia de eventos típica
Acción de un actor	Acción del sistema
1. El cliente selecciona ‘consulta pedidos’	2. El sistema presenta una lista con un resumen de los pedidos del cliente
Secuencias alternativas	
Ninguna	
Postcondiciones	Ninguna

Cuadro 5: Especificación expandida de ‘consulta pedidos’

login	
Actores:	cliente(iniciador)
Precondiciones:	El cliente no ha hecho login
	Secuencia de eventos típica
Acción de un actor	Acción del sistema
1. El cliente selecciona la opción ‘login’	2. El sistema presenta un formulario de login, y las opciones ‘usuario nuevo’ y ‘cancelar’
3. El cliente introduce login y password	4. El sistema va a la página de inicio, indicando el nombre del cliente
	Secuencias alternativas
Acción 3: Si el cliente selecciona ‘usuario nuevo’, se inicia ‘registro’	
Acción 3: Si el cliente selecciona ‘cancelar’, finaliza el caso de uso	
Acción 3: Si el login o password no son correctos, vuelve a 1	
Postcondiciones	El cliente queda identificado en el correspondiente grupo de seguridad

Cuadro 6: Especificación expandida de ‘login’

registro	
Actores:	cliente(iniciador)
Precondiciones:	El cliente no ha hecho login
	Secuencia de eventos típica
Acción de un actor	Acción del sistema
1. El cliente selecciona la opción ‘registro’	2. El sistema presenta un formulario de registro, y la opciones ‘confirmar’ y ‘cancelar’
3. El cliente rellena el formulario y selecciona ‘confirmar’	4. El sistema valida los datos, registra el nuevo usuario, hace el login, y va a la página de inicio, indicando el nombre del cliente
	Secuencias alternativas
Acción 3: Si los datos de registro no son correctos, vuelve a 2	
Acción 3: Si el cliente selecciona ‘cancelar’, finaliza el caso de uso	
Postcondiciones	El sistema registra el nuevo cliente, y el cliente queda identificado en el correspondiente grupo de seguridad

Cuadro 7: Especificación expandida de ‘registro’

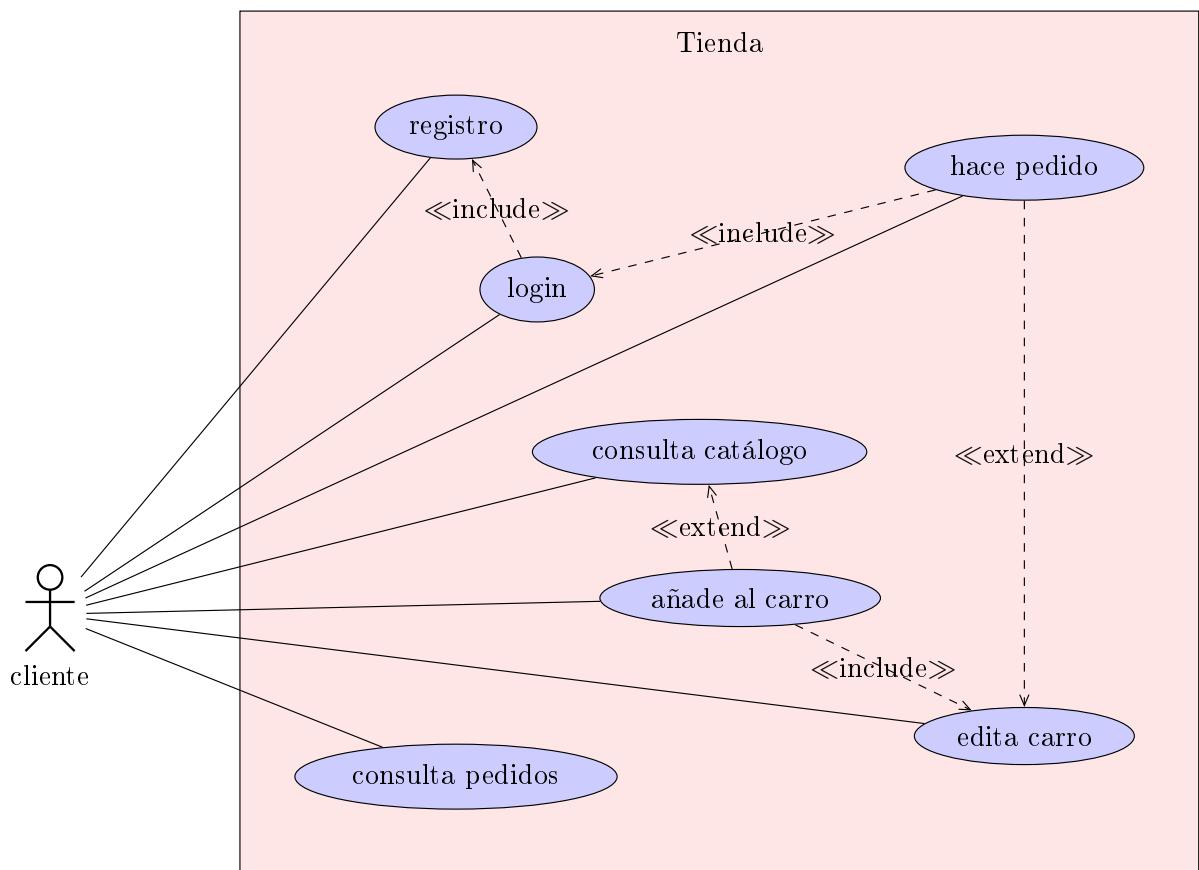


Figura 3: Diagrama de casos de uso estructurado de la arquitectura ejecutable

2.2. Análisis

Las clases del análisis se muestran en el diagrama de clases de figura 4.

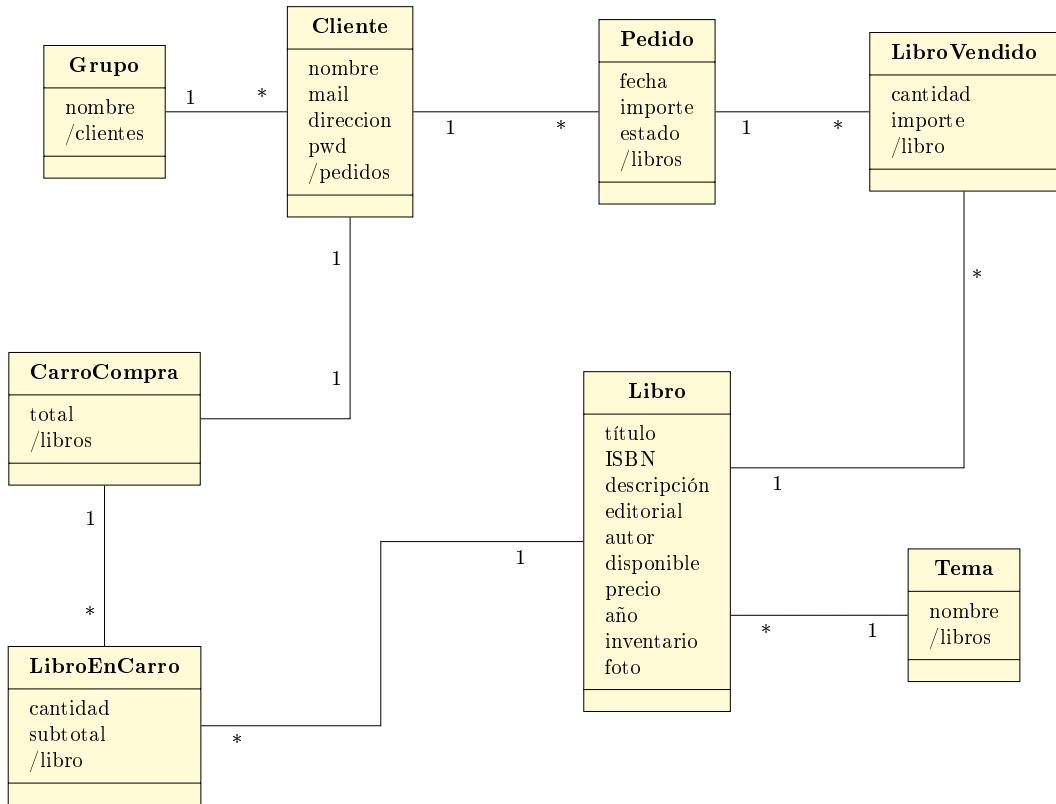


Figura 4: Diagrama de clases del análisis

Las responsabilidades a las clases del análisis se muestran en los diagramas de interacción (uno de por cada caso de uso) de las figuras 5, 6, 7, 8, 9, 10 y 11.

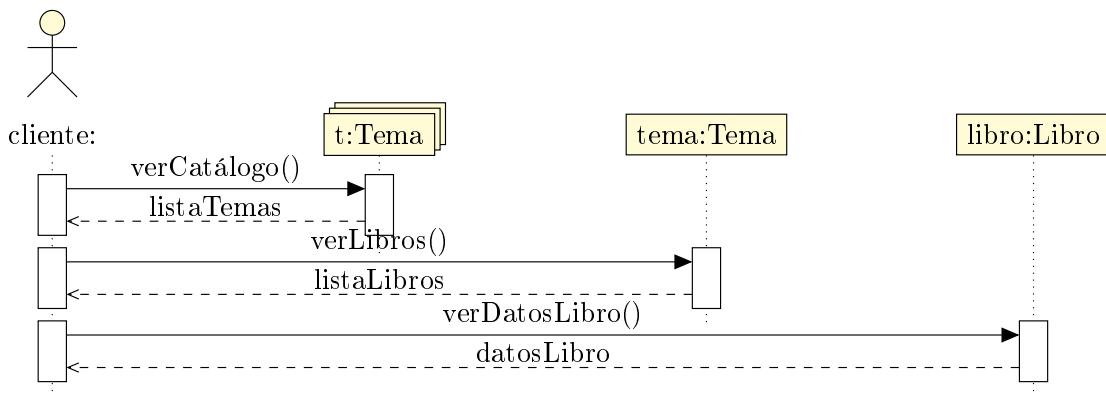


Figura 5: Diagrama de interacción del análisis de ‘consulta catálogo’

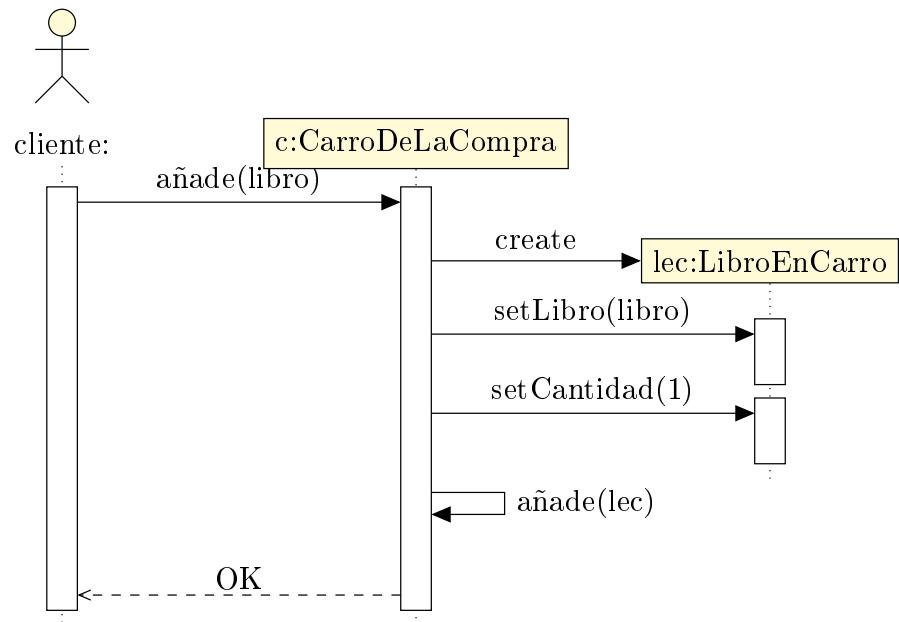


Figura 6: Diagrama de interacción del análisis de ‘añade al carro’

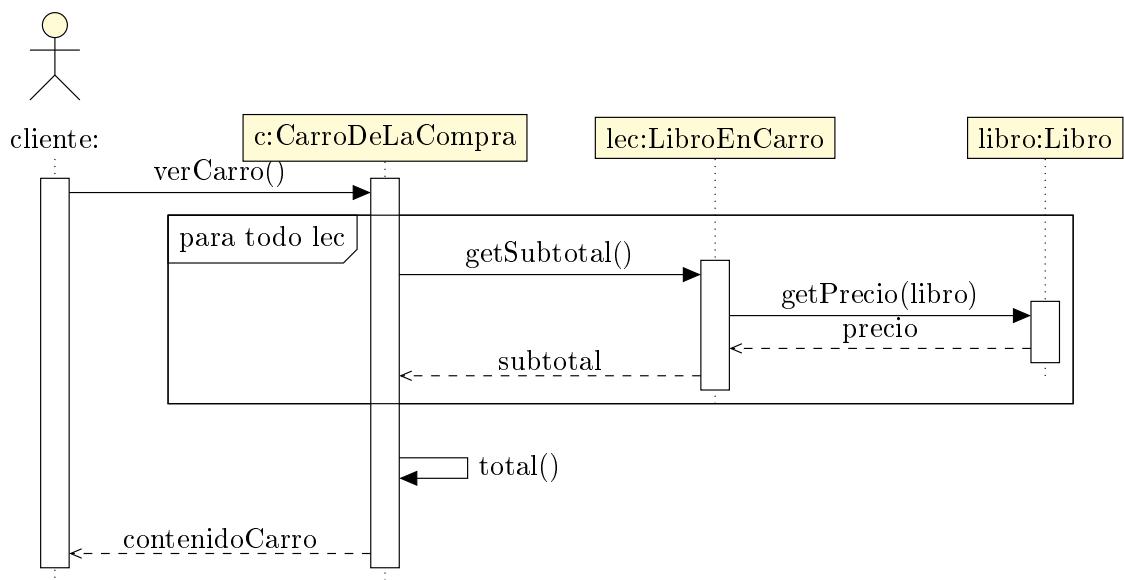


Figura 7: Diagrama de interacción del análisis de ‘edita carro’

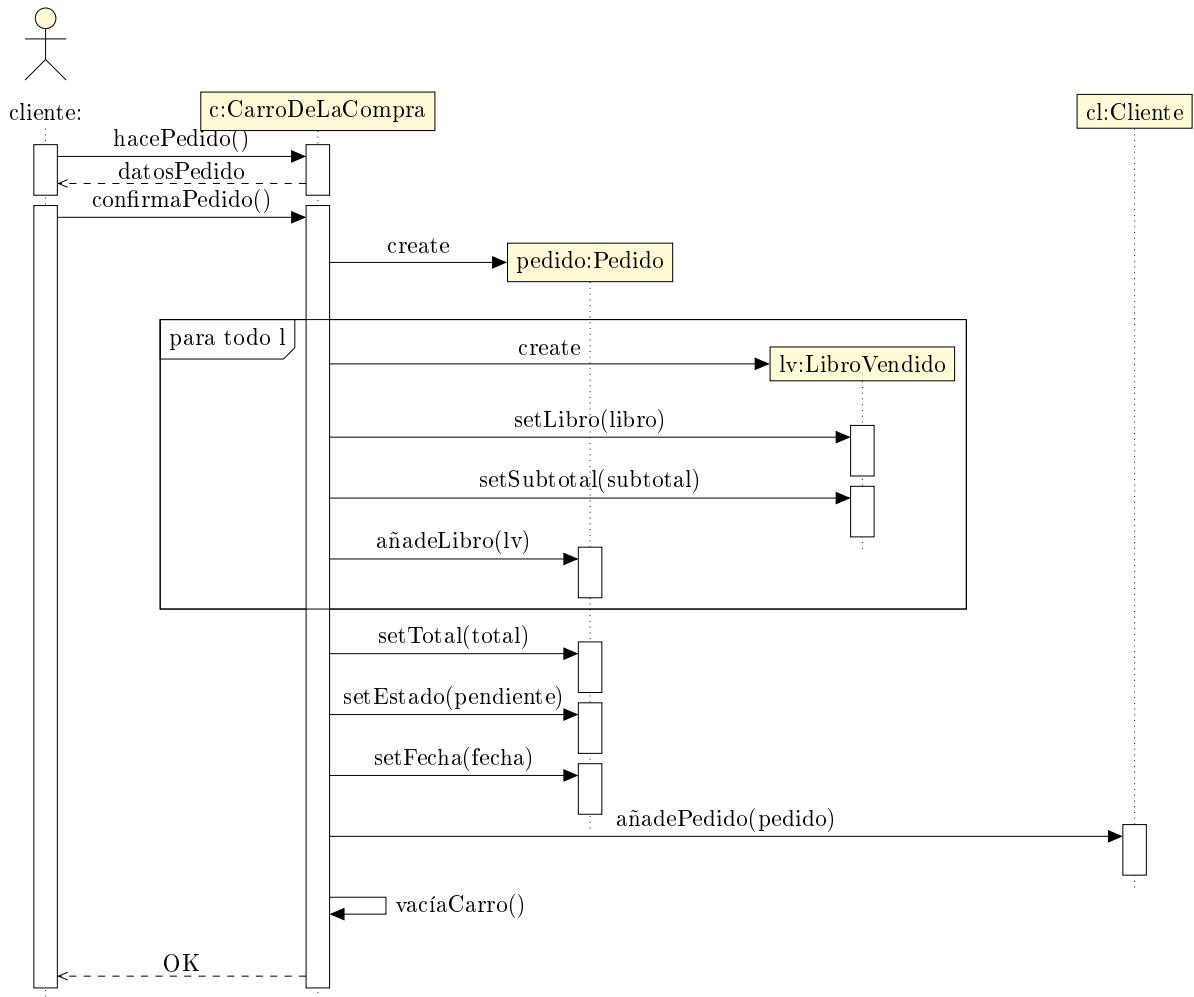


Figura 8: Diagrama de interacción del análisis de ‘hace pedido’

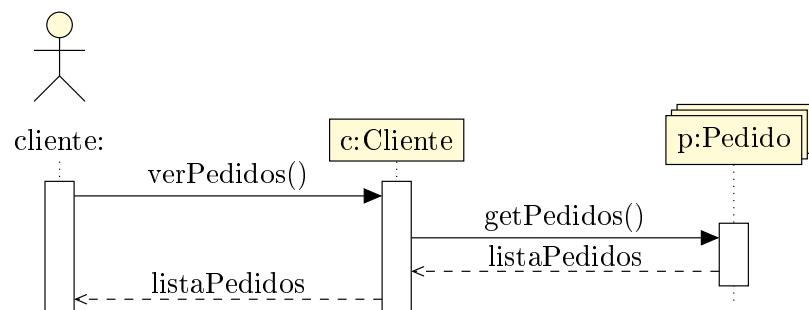


Figura 9: Diagrama de interacción del análisis de ‘consulta pedidos’

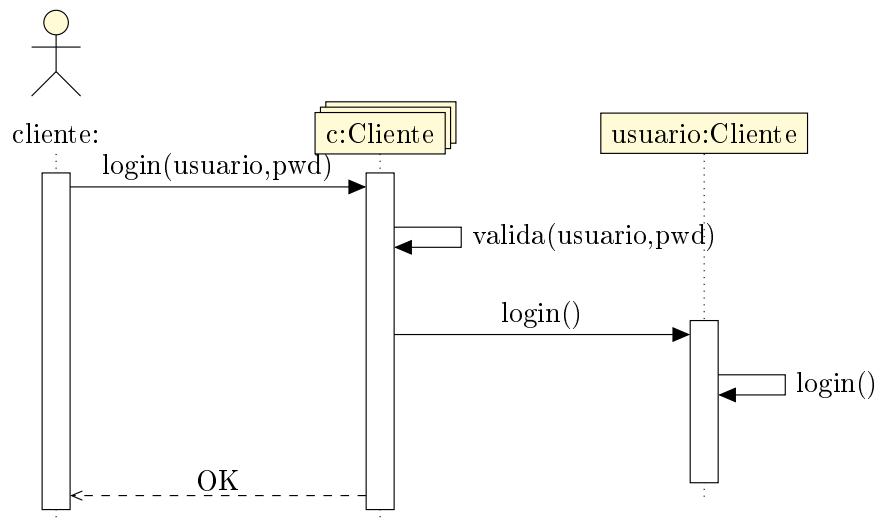


Figura 10: Diagrama de interacción del análisis de 'login'

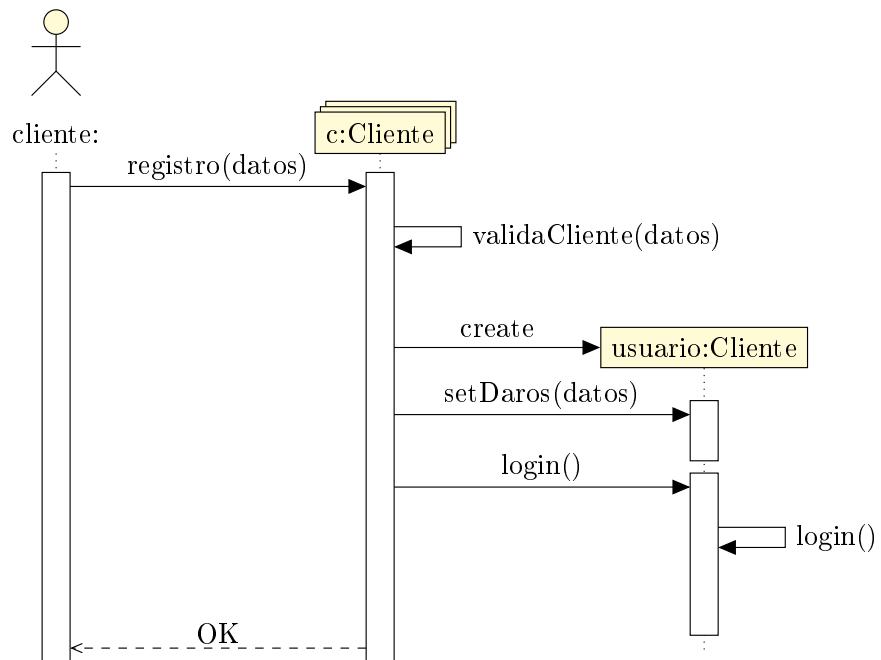


Figura 11: Diagrama de interacción del análisis de 'registro'

2.3. Diseño

El diseño se ha realizado con los siguientes componentes de JEE:

- Páginas java server face (JSF). Se ha incluido una JSF por cada vista identificada en los casos de uso. Éstas son:
 - inicioJSF: presenta las opciones de inicio.
 - listaPedidosJSF: presenta la lista de pedidos.
 - loginJSF: presenta el formulario de login.
 - registroJSF: presenta el formulario de registro.
 - registroErrorJSF: presenta los mensajes de error durante el registro.
 - listaTemasJSF: presenta la lista de temas.
 - listaLibrosJSF: presenta la lista de libros de un tema.
 - detallesLibroJSF: presenta los detalles de un libro.
 - carroCompraJSF: presenta el contenido del carro.
 - pedidoJSF: presenta los detalles del pedido.
- Entidades. Una por cada clase del análisis con datos persistentes. Éstas son:
 - Libro
 - Tema
 - Cliente
 - Pedido
 - LibroVendido
 - Grupo
- Enterprise java beans (EJB). Se han incluido EJBs para aquellas operaciones que requieren procesar entidades. Éstas son:
 - ClienteEJB: busca y crea entidades Cliente durante el login y el registro.
 - CarroCompraEJB: crea las entidades Pedido y LibroVendido durante la creación de un pedido.
 - CatalogoEJB: obtiene la lista de temas.
- Clases Java planas (POJO). La clase LibroEnCarro se ha implementado como una clase interna de CarroCompraEJB.
- Managed beans (MB). Se ha incluido una MB de nombre SesionMB para que sirva de enlace entre las JSF y las EJBs.

El resultado del diseño es un diagrama de secuencia por cada caso de uso (figuras 12, 13, 14, 15, 16, 17 y 18).

13

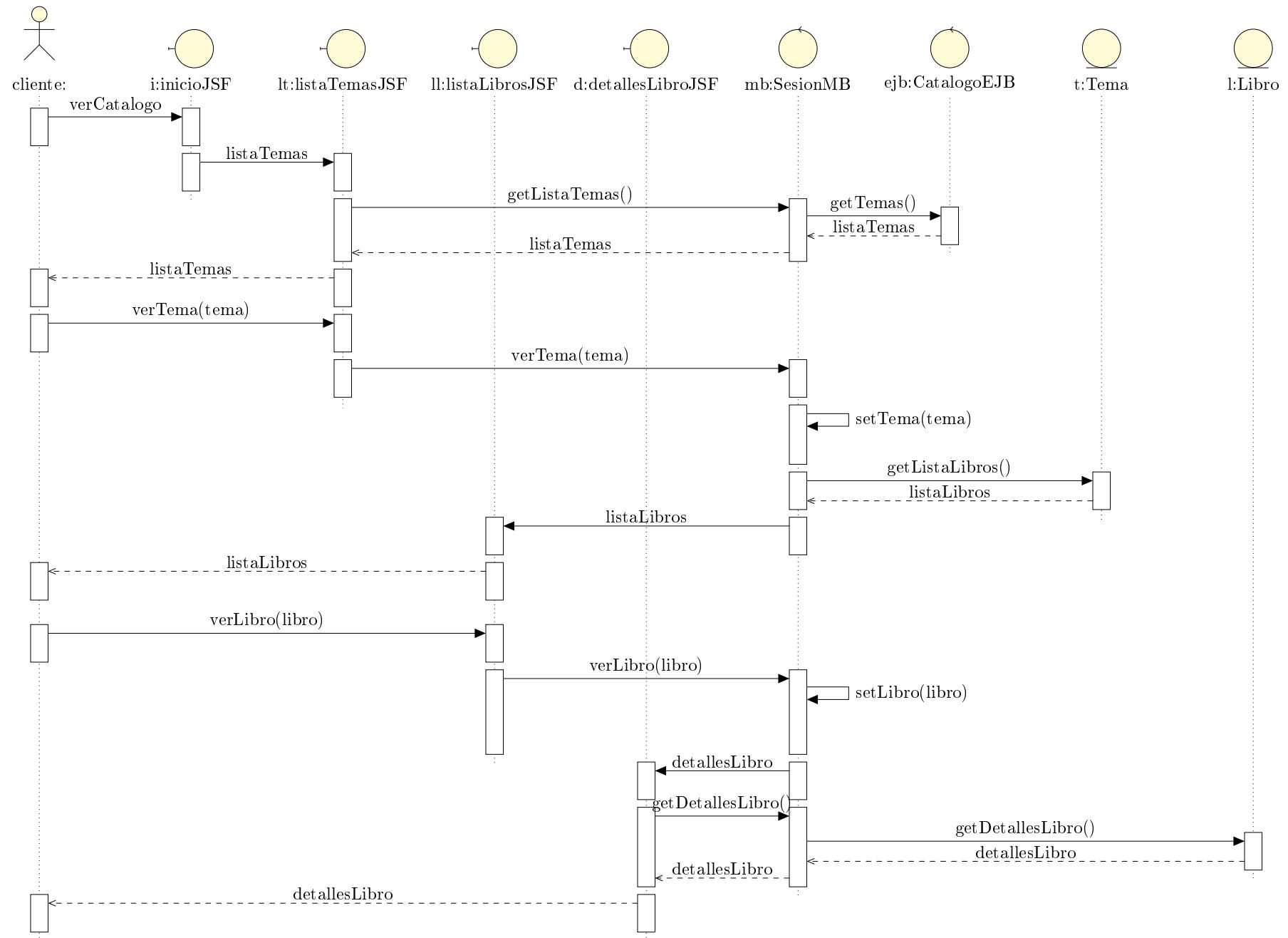


Figura 12: Diagrama de interacción del diseño de 'consulta catálogo'

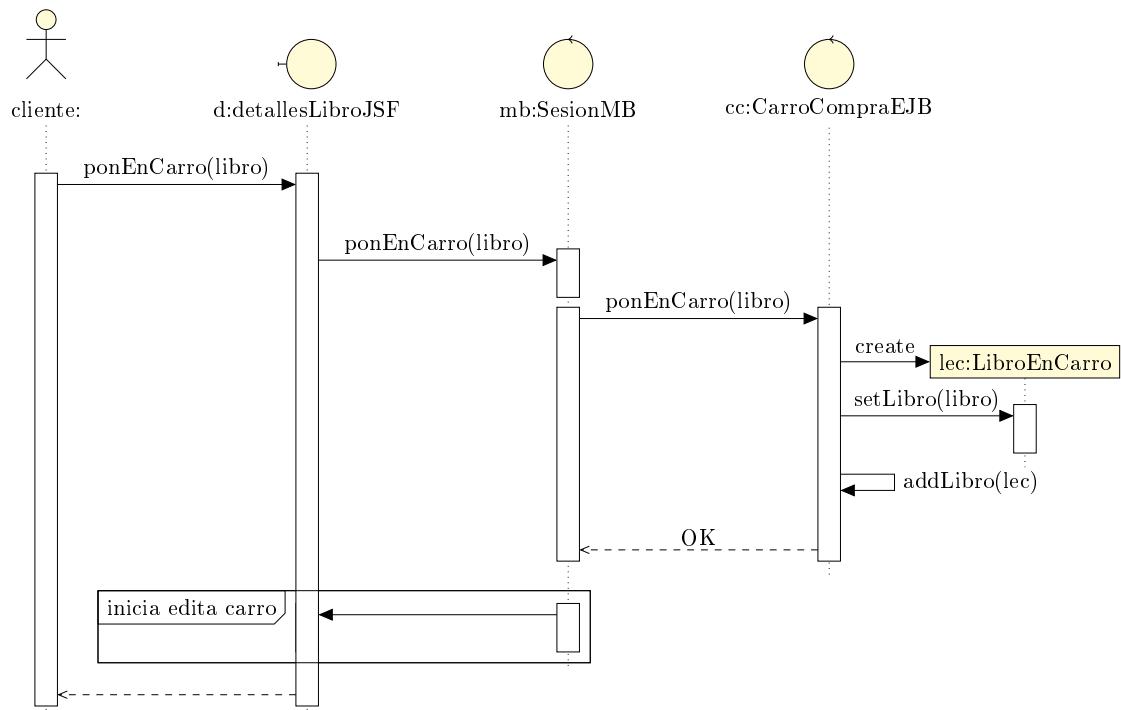


Figura 13: Diagrama de interacción del diseño de ‘añade al carro’

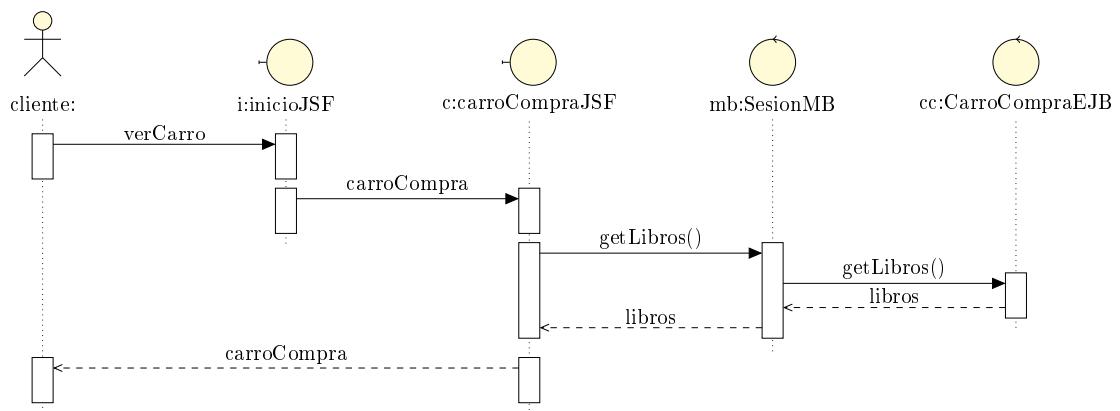


Figura 14: Diagrama de interacción del diseño de ‘edita carro’

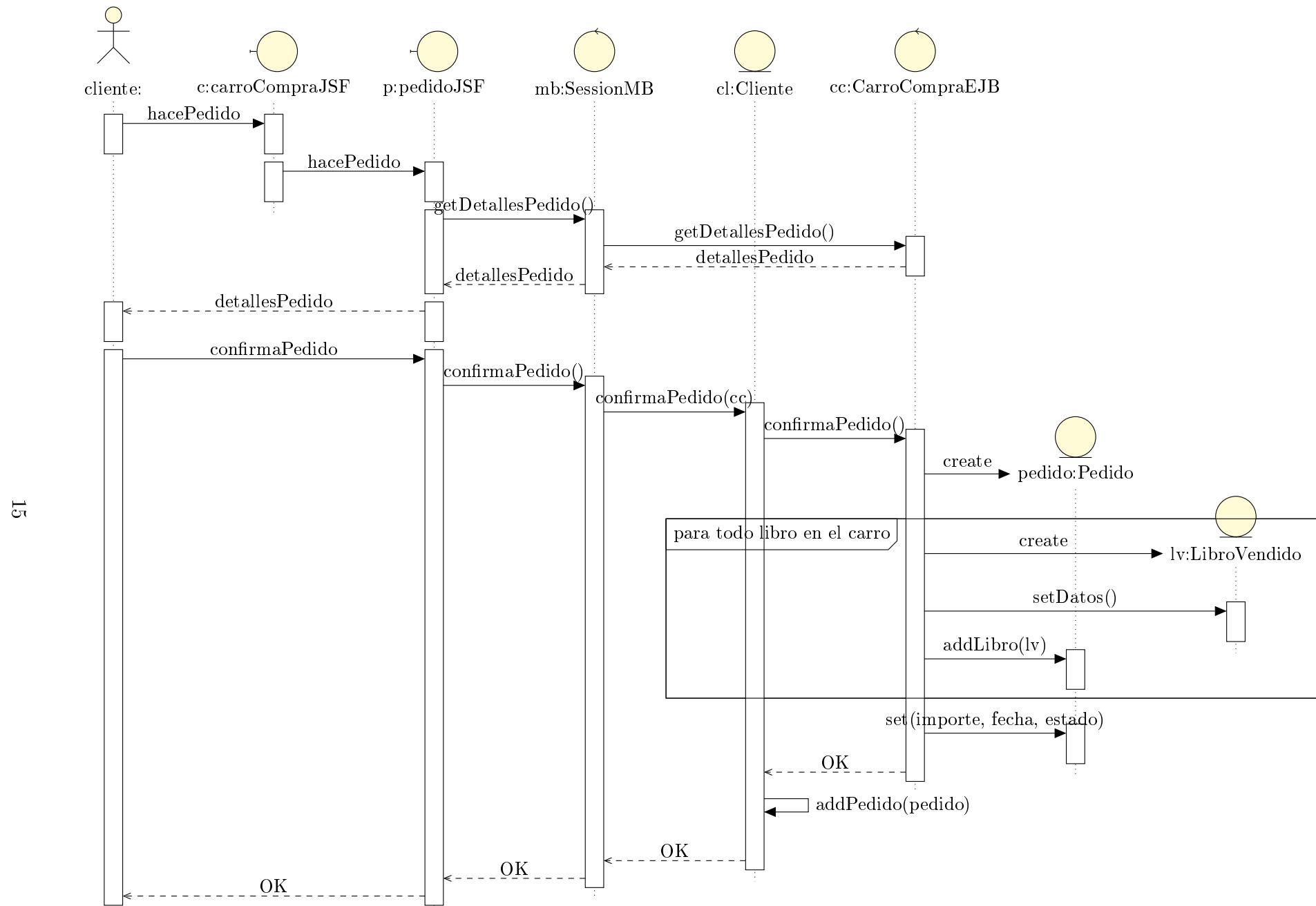


Figura 15: Diagrama de interacción del diseño de 'hace pedido'

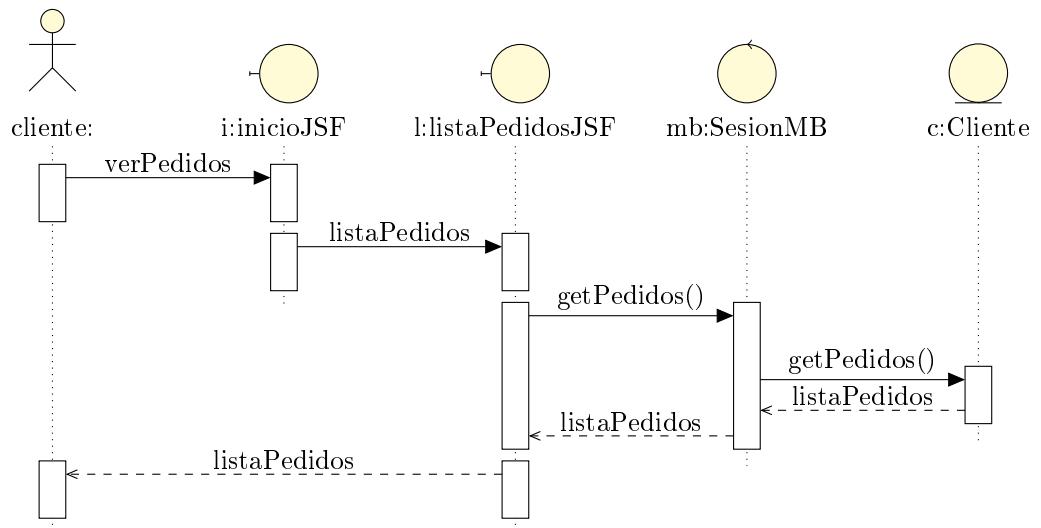


Figura 16: Diagrama de interacción del diseño de ‘consulta pedidos’

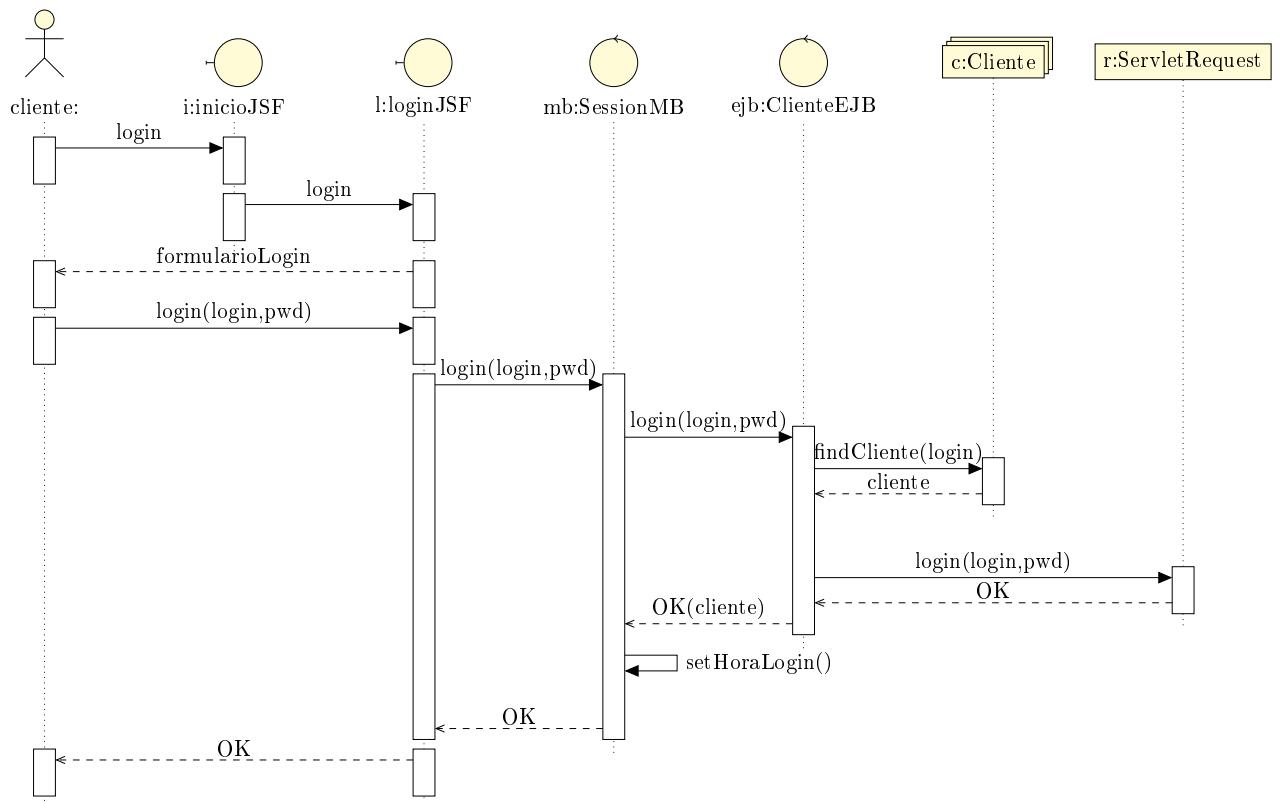


Figura 17: Diagrama de interacción del diseño de ‘login’

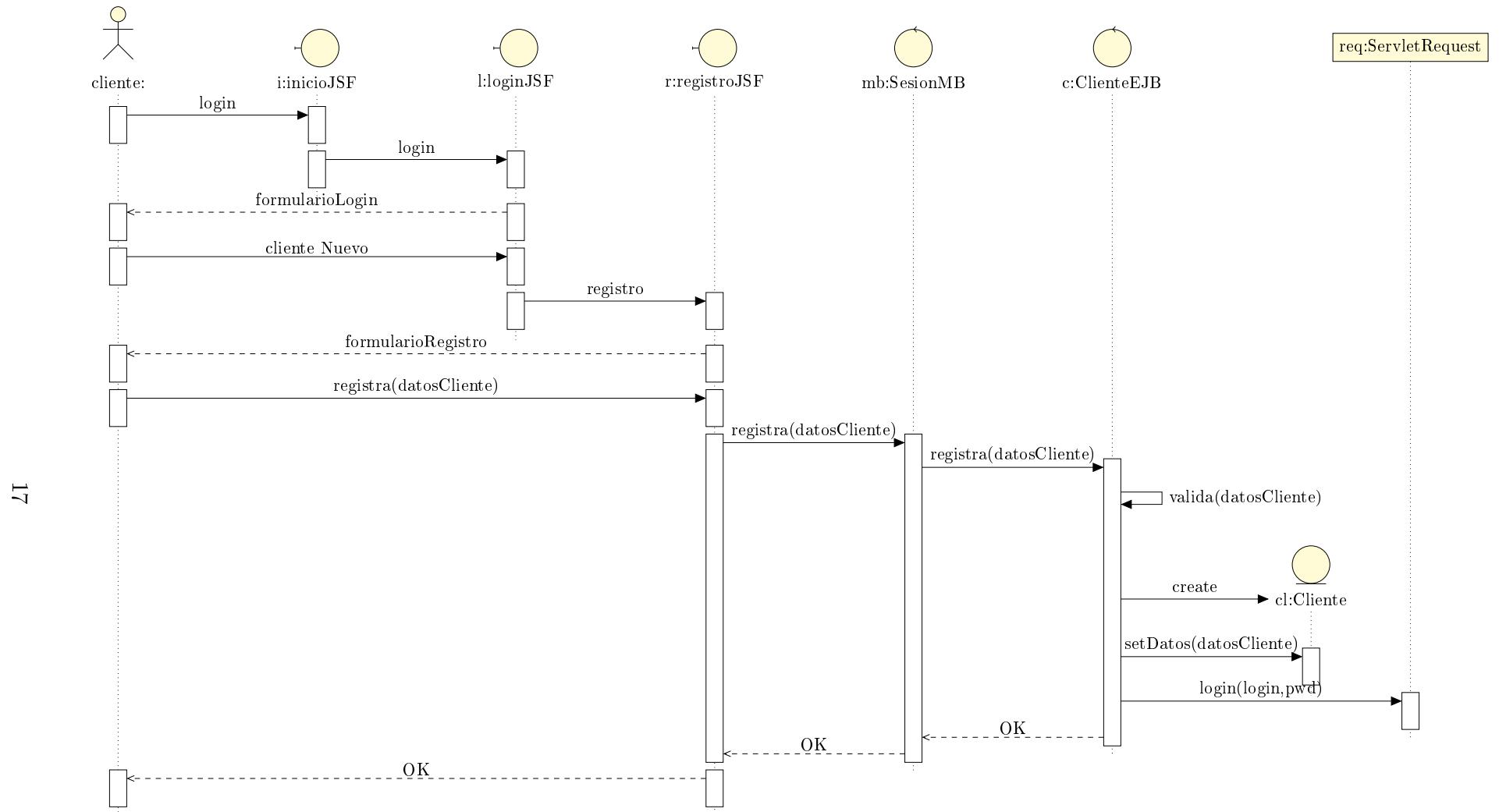


Figura 18: Diagrama de interacción del diseño de 'registro'

2.4. Implementación y despliegue

Las clases que implementan este diseño están en `proyectoIST.zip`. Para desplegar el sistema sigue los siguientes pasos:

1. Descomprime `proyectoIST.zip`
2. Crea un proyecto ‘Web Application with Existing Sources’.
3. Crea una base de datos. No dejes en blanco el login y el password.

Si quieres que los archivos de la base de datos se guardan en un lugar conocido, por ejemplo en la carpeta del proyecto, en la ventana en la que introduces el nombre de la base de datos, pulsa en ‘Properties ...’ y selecciona la carpeta del proyecto en ‘Database Location:’. De este modo dentro de la carpeta del proyecto se creará una carpeta con el mismo nombre que la base de datos, y en ella se guardarán todos los archivos de la base de datos.

4. Crea una unidad de persistencia de nombre ‘tiendaPU’, con una fuente de datos nueva de nombre ‘tiendaDS’ y conecta la fuente a la base de datos.
5. Ejecuta el proyecto y comprueba que se han creado las tablas.
6. Pon algún dato en las tablas ‘TEMA’ y ‘LIBRO’, ejecuta la aplicación y comprueba que se puede consultar el catálogo y llenar el carro de la compra.
7. Configura el dominio de seguridad en el servidor, en los siguientes pasos:

- a) Abre el ‘View Domain Admin Console’ del GlassFish Server.
- b) Despliega la pestaña ‘Configuraciones→server-config→Seguridad’.
- c) Selecciona ‘Dominios’, y crea un nuevo dominio con los siguientes parámetros:

```
Nombre de clase: com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm  
Contexto JASS: jdbcRealm  
JNDI: java:app/tiendaDS  
Tabla de usuarios: cliente  
Columna de Nombres de Usuario: login  
Columna de Contraseñas: pwd  
Tabla de Grupos: cliente_grupo  
Columna de Nombres de Usuario de la Tabla de Grupos: cliente_login  
Columna de Nombres de Grupo: grupos_nombre  
Algoritmo de Cifrado de Contraseña: none  
Algoritmo de Resumen: SHA-512
```

- d) Selecciona la pestaña ‘Seguridad’ y pon este dominio como ‘Dominio por defecto’.
8. Añade a la tabla ‘GRUPO’ el registro ‘clientes’.
9. Añade el rol ‘clientes’ en el archivo de configuración `web.xml` (desde ‘Security Roles’ en la pestaña ‘Security’).
10. Asigna el rol ‘clientes’ al grupo ‘clientes’ en el archivo de configuración `glassfish-web.xml` (desde ‘Security Role Mappings’ en la pestaña ‘Security’). Si el archivo `glassfish-web.xml` no está entre los archivos de configuración del proyecto, añádelo creando un ‘GlassFish Descriptor’ dentro de ‘WEB-INF’.
11. Reinicia el GlassFish Server.
12. Vuelve a ejecutar el proyecto y prueba el registro, el login y los pedidos.

3. Tercera fase: *construction*

Durante la fase de *construction* se incorpora, en ciclos sucesivos, el resto de la funcionalidad especificada en el modelo general de casos de uso. El proyecto del curso consistirá en desarrollar uno de estos ciclos. Para ello has de elegir un subconjunto de casos de uso del modelo general (figura 1) e incorporarlo a los casos de uso de la arquitectura ejecutable desarrollada en la fase de *elaboration* (figura 2). A continuación, has de desarrollar todas las actividades del ciclo (requisitos, análisis, diseño, implementación y prueba) para incorporar la funcionalidad elegida. El resultado será una memoria con los siguientes diagramas:

- Un diagrama de casos de uso con la funcionalidad del ciclo (los de la fase anterior más los nuevos).
- Una especificación expandida de cada uno de los casos de uso nuevos.
- Un diagrama de casos de uso estructurado con la funcionalidad del ciclo.
- Un diagrama de clases del análisis que incorpore las clases nuevas identificadas.
- Un diagrama de secuencia de objetos del análisis por cada caso de uso nuevo.
- Un diagrama de secuencia de componentes del diseño por cada caso de uso nuevo.