



MEMORIA - GESTIÓN DE CINE

- v.1.0

9 de noviembre de 2022
Modelado de Software
Grupo E – Ingeniería del Software



INTEGRANTES DEL EQUIPO

Alberto Herrera García
Alfonso Burguete Gotor
Axel Kahou He Expósito
Beatriz Espinar Aragón
Daniel Barroso Casado
Jose María Gálvez Gómez
Miriam Choy Castillo
Nicolás Espinosa Mooser
Santiago Moral Santorio
Sara Sacó Baños
Steven Mallqui Aguilar
Unai Piris Ibáñez

CONTENIDO

1. Introducción.....	4
2. Desarrollo	5
2.1. Resumen.....	5
2.2. Proceso	6
2.3. Factores de calificación	18
3. Producto.....	19
3.1. Casos de uso	19
3.2. Interfaces externas	20
4. Arquitectura.....	25
4.1. Patrones.....	25
5. Modelo	27
5.1. Diagramas de clase	27
5.2. Diagramas de secuencia	35
5.3. Diagrama de despliegue	45
6. Estructura del código.....	46
7. Pruebas.....	48
8. Recursos	50

1. Introducción

La finalidad de este documento es presentar un informe sobre la primera parte del proyecto *Plateau* y cómo ha sido llevada a cabo. Esta primera parte consiste en una aplicación para gestionar un cine a nivel administrativo, tal y como se detalla en la Especificación de Requisitos Software (SRS).

El proyecto *Plateau* ha sido desarrollado durante los meses de septiembre, octubre y noviembre de 2022, y permite gestionar los pases que se realizan en un cine, con las películas que se emiten y sus correspondientes productoras. Además, permite gestionar compras y todo lo relativo a los clientes que las realizan.

Este documento queda dividido en secciones, cuyo contenido se explica brevemente a continuación:

- Desarrollo. En este apartado se describe el proceso de desarrollo del software, cómo se ha ido construyendo y cómo se ha realizado el reparto de tareas.
- Producto. En este apartado se muestra cómo ha quedado el producto final en comparación a lo establecido en la SRS, qué funcionalidades se han implementado y qué interfaces se han diseñado de cara al usuario.
- Arquitectura. En este apartado se especifica la arquitectura empleada, junto a los patrones que se han implementado para un correcto modelado del software.
- Modelo. En este apartado se presentan los distintos diagramas de clase, secuencia y despliegue que se han realizado antes de pasar al desarrollo del código, junto a algunos ejemplos visuales.
- Estructura del código. En este apartado se explica brevemente la estructura de carpetas seleccionada para mantener las clases Java ordenadas y dónde se encuentran otros archivos relevantes relacionados con la Base de Datos, entre otros.
- Pruebas. En este apartado se describen las distintas pruebas que se han llevado a cabo para asegurar el correcto funcionamiento del software tanto a nivel de requisitos como a nivel puramente funcional.
- Recursos. En este apartado se comentan los principales recursos software empleados para poder desarrollar el proyecto.

Así, el lector podrá tener una visión global y, a su vez, detallada, de lo que ha supuesto el desarrollo de la primera parte de *Plateau*, que se completará más adelante con el correspondiente desarrollo de la gestión de una tienda.

2. Desarrollo

2.1. Resumen

El proyecto comenzó tras la elección del dominio de la aplicación, al confeccionar la idea de *Plateau* como un gestor de cine, y con una ampliación posterior que permitiese gestionar la tienda de dicho cine. Así, el primer paso fue concretar el Modelo del Dominio, con las distintas entidades que resultan relevantes en este contexto, y las relaciones existentes entre ellas. De este modo, empezó entonces la elaboración de la Especificación de Requisitos Software del proyecto, para poder plantear de manera clara y concisa los requerimientos principales que debía cumplir la aplicación.

Una vez descritos con detalle los casos de uso a partir del Modelo del Dominio generado, y después de corregirlos tras el *feedback* recibido por parte del profesor de la asignatura, el siguiente paso era elaborar el modelo de la aplicación. Para ello, el equipo generó una lista de los diagramas necesarios para cubrir toda la funcionalidad de *Plateau*, y se hizo un reparto de tareas que se actualizaba conforme se iban terminando. Se empezó por la capa de Integración y Negocio, realizando primero los diagramas de clase y pasando después a los de secuencia. A continuación, se hizo el mismo proceso con la capa de Presentación, hasta que ya quedó todo el modelo configurado.

Así pues, se transformó el modelo en código y comenzó el desarrollo de este. En primer lugar, se desarrollaron las vistas, para facilitar el proceso de depuración posterior, y así no tener que hacer pruebas mediante comandos por consola. Se realizaron también los Command y los Transfers, que no requerían mucha complicación, y así la aplicación empezó a tomar forma.

El siguiente paso fue completar la capa de negocio, codificando los métodos correspondientes a los servicios de aplicación, para que empezasen a tenerse en cuenta las reglas de negocio relativas a los requisitos de *Plateau* especificados en la SRS. Una vez hecho esto, sólo quedaba completar la capa de integración para poder conectarse al almacén persistente y crear, para cada módulo, las operaciones CRUD (*Create, Read, Update, Delete*), además de otras necesarias para el correcto funcionamiento de la aplicación.

Por último, pero no menos importante, se realizaron todas las pruebas que se consideraron necesarias para poder aumentar todo lo posible el nivel de fiabilidad del software, y se comprobó con la SRS que todo actuaba de acuerdo a los requisitos especificados. Además, se hicieron diversos retoques para mejorar la estética de las interfaces de usuario y para mejorar la legibilidad del código, así como otras mejoras y correcciones.

Por supuesto, a lo largo de todo este proceso se han llevado a cabo diferentes reuniones que permitían al equipo estar al día del avance del proyecto, informar sobre posibles errores o contingencias que surgían, y planificar los próximos pasos. Aparte de esto, se han utilizado constantemente plataformas como *Whastapp* o *Discord* para que los integrantes se mantengan en contacto, puedan pedir ayuda y prestarla cuando sea necesario. Así, el equipo se ha asegurado de mantener una comunicación activa, constante y eficaz, aumentando la visibilidad y mejorando el proceso de desarrollo, y fomentando continuamente el trabajo en equipo.

Además, en lo relativo a los cambios, es relevante mencionar la naturaleza ágil que ha mantenido el proceso de desarrollo. Esto es, se ha mantenido actualizado todo el material que se ha ido generando,

tanto el Modelo del Dominio, como la SRS y los casos de uso, como el modelo y como cualquier otro producto generado. Así, los cambios se han aceptado como parte natural e inherente al desarrollo del software, y el equipo ha sido capaz de adaptarse a todo tipo de contratiempos y modificaciones.

El proyecto concluye con la revisión y corrección por parte del profesor de la asignatura, y cuyas indicaciones se tendrán en cuenta para el desarrollo de la segunda parte del proceso, además de servir como aprendizaje personal de cada integrante.

2.2. Proceso

En este apartado se incluye una tabla con todas las tareas que se han ido realizando durante el desarrollo de *Plateau*, con las fechas más relevantes y los encargados de dichas tareas. De este modo, el lector podrá visualizar con un alto nivel de detalle cómo ha evolucionado el proyecto a lo largo de estos meses y qué pasos se han ido dando hasta obtener el producto final que se presenta.

TAREA	AUTOR	FECHA
Desarrollo de la SRS		05/09-28/09
Discusión modelo del dominio	<i>TODOS</i>	-
Reunión planteamiento casos de uso	<i>TODOS</i>	-
Desarrollo casos de uso		-
Cliente	<ul style="list-style-type: none"> · Santiago Moral · Daniel Barroso 	-
Compra	<ul style="list-style-type: none"> · Steven Mallqui · Axel Kahou 	-
Película	<ul style="list-style-type: none"> · Nicolás Espinosa · Alfonso Burguete 	-
Sala	<ul style="list-style-type: none"> · Jose María Gálvez · Miriam Choy 	-
Productora	<ul style="list-style-type: none"> · Beatriz Espinar · Sara Sacó 	-
Pase	<ul style="list-style-type: none"> · Alberto Herrera · Unai Piris 	-
Redacción	<i>TODOS</i>	-
Corrección casos de uso	<ul style="list-style-type: none"> · Beatriz Espinar · Steven Mallqui 	-
Cambios en la SRS		04/10-05/11
Añadir nuevos casos de uso	<ul style="list-style-type: none"> · Beatriz Espinar 	-
Añadir productora a película		-
Quitar productora a película		-
Añadir proveedor a producto		-

TAREA	AUTOR	FECHA
Quitar proveedor a producto		-
Devolver compra		-
Devolver venta		-
Mostrar compra en detalle		-
Mostrar venta en detalle		-
Corrección de las precondiciones y postcondiciones	• Beatriz Espinar • Steven Mallqui	-
Correcciones en el modelo del dominio y lo correspondiente en los casos de uso	• Beatriz Espinar	-
Añadido atributo <i>nombre</i> a Sala		-
Añadido atributo <i>precio_actual</i> y <i>precio</i> en Pase y Compra		-
Añadido atributo <i>precio_actual</i> y <i>precio</i> en Producto y Compra		-
Añadido atributo <i>nomina</i> a Departamento		-
Añadido atributo <i>dni</i> a Cliente		-
El Cliente en la gestión de tienda pasa a llamarse ClienteJPA		-
Sólo “Cerrar compra” comprueba que los datos son correctos	• Beatriz Espinar • Steven Mallqui	-
Sólo “Cerrar venta” comprueba que los datos son correctos	• Beatriz Espinar • Steven Mallqui	-
Las descripciones de los casos de uso ahora son acciones	• Beatriz Espinar	-
Condiciones en los casos de uso revisadas y añadidas	• Beatriz Espinar • Steven Mallqui	-
Otros cambios	• Beatriz Espinar	-
Añadidos parámetros fechas a la query de Productora		-
La <i>primary key</i> de Pase es <i><horario, id_sala></i>		-
Al añadir un pase/producto a una compra/venta el usuario no introduce el precio		-
No se puede modificar el tipo de una entidad especializada		-
En los casos de uso de modificar se introducen todos los campos		-
En los casos de uso de listar se muestra el ID y si está activo		-
En los casos de uso de mostrar se muestra si está activo		-
Set-up de la tecnología		29/09
Instalación IBM y repositorio (SVN)	<i>TODOS</i>	-
Tutorial para hacer commits	• Alberto Herrera • Steven Mallqui	-

TAREA	AUTOR	FECHA
Organización carpetas	· Steven Mallqui	-
Acuerdo convenios (nombres métodos, CamelCase...)	<i>TODOS</i>	-
Diagramas de clase (Integración y Negocio)		30/09-02/10
Pensar modelos básicos	· Steven Mallqui	-
Desarrollo diagramas (uno de negocio y uno de integración por cada entidad)		-
Cliente	· Santiago Moral · Daniel Barroso	-
Compra	· Steven Mallqui · Axel Kahou	-
Película	· Nicolás Espinosa · Alfonso Burguete	-
Sala	· Jose María Gálvez · Miriam Choy	-
Productora	· Beatriz Espinar · Sara Sacó	-
Pase	· Alberto Herrera · Unai Piris	-
Diagramas DAOFactory y ASFactory	· Steven Mallqui · Santiago Moral	-
Diagrama QueryFactory	· Steven Mallqui · Santiago Moral	-
Diagrama transacciones	· Steven Mallqui · Santiago Moral	-
Corrección de todos los diagramas	· Steven Mallqui · Beatriz Espinar	-
“Recorrección” tras los cambios en la SRS	· Beatriz Espinar · Steven Mallqui	02/10-08/10
Diagramas de secuencia (Integración y Negocio)		04/10-08/10
Pensar modelos básicos	· Steven Mallqui	-
Diagramas secuencia integración Cliente	· Santiago Moral	-
Alta		-
Baja		-
Modificar		-
Mostrar		-
Listar		-

TAREA	AUTOR	FECHA
Corrección diagramas secuencia integración Cliente	· Beatriz Espinar	09/10-12/10
Diagramas secuencia integración Compra		-
Cerrar	· Steven Mallqui · Santiago Moral	-
Mostrar	· Alberto Herrera	-
Listar	· Unai Piris	-
Listar por cliente	· Daniel Barroso	-
Listar por pase	· Nicolás Espinosa	-
Mostrar en detalle	· Beatriz Espinar	-
Corrección diagramas secuencia integración Compra	· Beatriz Espinar	08/10-12/10
Diagramas secuencia integración Película		-
Alta	· Jose María Gálvez	-
Baja	· Miriam Choy	-
Modificar	· Alfonso Burguete	-
Mostrar	· Axel Kahou	-
Listar	· Santiago Moral	-
Listar por productora	· Alberto Herrera	-
Añadir productora	· Beatriz Espinar	-
Corrección diagramas secuencia integración Película	· Beatriz Espinar	08/10-12/10
Diagramas secuencia integración transacciones		-
New	· Jose María Gálvez	-
Start	· Axel Kahou	-
Commit	· Miriam Choy	-
Rollback	· Alfonso Burguete	-
Corrección diagramas secuencia integración transacciones	· Beatriz Espinar · Alberto Herrera	08/10
Diagramas secuencia negocio Cliente	· Santiago Moral	-
Alta		-
Baja		-
Modificar		-
Mostrar		-
Listar		-
Corrección diagramas secuencia negocio Cliente	· Beatriz Espinar	13/10-17/10

TAREA	AUTOR	FECHA
Diagramas secuencia negocio Compra		-
Cerrar	· Steven Mallqui · Santiago Moral	-
Mostrar	· Alberto Herrera	-
Listar	· Unai Piris	-
Listar por cliente	· Daniel Barroso	-
Listar por pase	· Nicolás Espinosa	-
Mostrar en detalle	· Beatriz Espinar	-
Corrección diagramas secuencia negocio Compra	· Steven Mallqui · Beatriz Espinar	13/10-18/10
Diagramas secuencia negocio Película		-
Alta	· Jose María Gálvez	-
Baja	· Miriam Choy	-
Modificar	· Alfonso Burguete	-
Mostrar	· Axel Kahou	-
Listar	· Steven Mallqui	-
Listar por productora	· Unai Piris	-
Añadir productora	· Daniel Barroso	-
Corrección diagramas secuencia negocio Película	· Beatriz Espinar	13/10-17/10
Diagramas de clase (Presentación)		09/10-11/10
Pensar modelos básicos	· Steven Mallqui	-
Diagrama clase Command	· Steven Mallqui	-
Diagrama clase Controller (sin dependencias con paneles)	· Steven Mallqui	-
Diagrama clase Main GUI	· Steven Mallqui	-
Reunión aprender presentación y reparto de diagramas	<i>TODOS</i>	-
Diagramas clase Command de cada entidad		-
Cliente	· Santiago Moral · Daniel Barroso	-
Compra	· Steven Mallqui · Axel Kahou	-
Película	· Nicolás Espinosa · Alfonso Burguete	-
Sala	· Jose María Gálvez · Miriam Choy	-

TAREA	AUTOR	FECHA
Productora	· Sara Sacó · Beatriz Espinar	-
Pase	· Unai Piris · Alberto Herrera	-
Corrección diagramas clase Command	· Steven Mallqui	12/10
Diagramas clase paneles de cada entidad		-
Cliente	· Santiago Moral · Daniel Barroso	-
Compra	· Steven Mallqui · Axel Kahou	-
Película	· Nicolás Espinosa · Alfonso Burguete	-
Sala	· Jose María Gálvez · Miriam Choy	-
Productora	· Beatriz Espinar · Sara Sacó	-
Pase	· Alberto Herrera · Unai Piris	-
Corrección diagramas clase paneles	· Steven Mallqui · Beatriz Espinar	12/10-22/10
Añadir paneles de cada entidad al diagrama clase Controller		-
Cliente	· Santiago Moral · Daniel Barroso	-
Compra	· Steven Mallqui · Axel Kahou	-
Película	· Nicolás Espinosa · Alfonso Burguete	-
Sala	· Jose María Gálvez · Miriam Choy	-
Productora	· Beatriz Espinar · Sara Sacó	-
Pase	· Alberto Herrera · Unai Piris	-
Corrección paneles en el diagrama clase Controller	· Beatriz Espinar	19/10
Diagramas de secuencia (Presentación)		12/10-15/10
Pensar modelos básicos	· Steven Mallqui	-

TAREA	AUTOR	FECHA
Diagramas secuencia Command Cliente		-
Alta	· Santiago Moral	-
Baja	· Santiago Moral	-
Modificar	· Alberto Herrera	-
Mostrar	· Alberto Herrera	-
Listar	· Sara Sacó	-
Listar normales	· Jose María Gálvez	-
Listar VIP	· Sara Sacó	-
Corrección diagramas secuencia Command Cliente	· Beatriz Espinar	19/10
Diagramas secuencia Command Película		-
Alta	· Jose María Gálvez	-
Baja	· Unai Piris	-
Modificar	· Unai Piris	-
Mostrar	· Daniel Barroso	-
Listar	· Daniel Barroso	-
Listar por productora	· Miriam Choy	-
Añadir productora	· Miriam Choy	-
Quitar productora	· Nicolás Espinosa	-
Corrección diagramas secuencia Command Película	· Beatriz Espinar	19/10
Diagramas secuencia Command Compra		-
Cerrar	· Steven Mallqui	-
Listar	· Steven Mallqui	-
Listar por cliente	· Steven Mallqui	-
Listar por pase	· Beatriz Espinar	-
Mostrar	· Beatriz Espinar	-
Mostrar en detalle	· Beatriz Espinar	-
Diagramas secuencia paneles Cliente		-
Alta	· Santiago Moral	-
Baja	· Alberto Herrera	-
Modificar	· Jose María Gálvez	-
Mostrar	· Sara Sacó	-
Listar	· Unai Piris	-
Listar normales	· Daniel Barroso	-
Listar VIP	· Miriam Choy	-

TAREA	AUTOR	FECHA
Diagramas secuencia paneles Película		-
Alta	· Nicolás Espinosa	-
Baja	· Nicolás Espinosa	-
Modificar	· Axel Kahou	-
Mostrar	· Axel Kahou	-
Listar	· Axel Kahou	-
Listar por productora	· Alfonso Burguete	-
Añadir productora	· Alfonso Burguete	-
Quitar productora	· Alfonso Burguete	-
Diagramas secuencia paneles Compra		-
Crear	· Steven Mallqui	-
Listar	· Steven Mallqui	-
Listar por cliente	· Steven Mallqui	-
Listar por pase	· Steven Mallqui	-
Mostrar	· Steven Mallqui	-
Mostrar en detalle	· Steven Mallqui	-
Devolver	· Santiago Moral	-
Corrección tras tutoría		20/10-22/10
Añadir tipos de retorno en los métodos del diagrama de clase de transacciones	· Beatriz Espinar	-
Arreglar relación entre TransactionMySQL y Connection	· Beatriz Espinar	-
Diagrama secuencia getTransaction()	· Beatriz Espinar	-
Diagrama secuencia Controller	· Steven Mallqui	-
Añadir dependencias en los diagramas de clase de los Commands	· Beatriz Espinar	-
Diagramas secuencia queries	· Steven Mallqui · Beatriz Espinar	-
Añadir casos de uso devoluciones	· Santiago Moral	-
Añadir método (DAO) para comprobar si existe vinculación <Productora,Película>	· Beatriz Espinar	-
Añadir notas explicativas en diagramas secuencia	· Beatriz Espinar	-
Arreglar ConcurrentHashMap en diagrama clase transacciones	· Beatriz Espinar	-

TAREA	AUTOR	FECHA
Código (primera parte)		23/10-26/10
Transformar modelo en código	• Steven Mallqui	-
MainWindow, GeneralPanel, GeneralMainPanel	• Steven Mallqui	-
Command, CommandFactory, ContextEnum	• Steven Mallqui	-
Paneles principales		-
Cliente	• Steven Mallqui	-
Compra	• Beatriz Espinar	-
Película	• Steven Mallqui	-
Sala	• Beatriz Espinar	-
Productora	• Beatriz Espinar	-
Pase	• Steven Mallqui	-
Reunión para repartir tareas y poner fechas límite	<ul style="list-style-type: none"> • Steven Mallqui • Beatriz Espinar • Sara Sacó • Miriam Choy • Jose María Gálvez • Alberto Herrera • Axel Kahou 	23/10
Paneles sin ActionListener (sólo vistas), Commands y Transfers		-
Cliente	<ul style="list-style-type: none"> • Alberto Herrera • Unai Piris 	-
Compra	<ul style="list-style-type: none"> • Steven Mallqui • Beatriz Espinar 	-
Pase	<ul style="list-style-type: none"> • Nicolás Espinosa • Axel Kahou 	-
Película	<ul style="list-style-type: none"> • Alfonso Burguete • Daniel Barroso 	-
Productora	<ul style="list-style-type: none"> • Santiago Moral • Sara Sacó 	-
Sala	<ul style="list-style-type: none"> • Miriam Choy • Jose María Gálvez 	-
Corregir Commands	<ul style="list-style-type: none"> • Beatriz Espinar • Steven Mallqui 	26/10-27/10
Corregir Transfers	• Beatriz Espinar	26/10

TAREA	AUTOR	FECHA
Corregir paneles	• Beatriz Espinar • Nicolás Espinosa	26/10-29/10
Código (segunda parte)		26/10-29/10
Creación BD y configurar conexiones	• Steven Mallqui • Santiago Moral	-
ApplicationController, Dispatcher, Context	• Steven Mallqui • Santiago Moral	-
Métodos de los AS		-
Cliente	• Alberto Herrera • Unai Piris	-
Compra	• Steven Mallqui • Beatriz Espinar	-
Pase	• Nicolás Espinosa • Axel Kahou	-
Película	• Alfonso Burguete • Daniel Barroso	-
Productora	• Santiago Moral • Sara Sacó	-
Sala	• Miriam Choy • Jose María Gálvez	-
Corrección métodos de los AS	• Beatriz Espinar • Steven Mallqui	30/10
Código (tercera parte)		30/10-02/11
ErrorHandler y Message	• Steven Mallqui • Beatriz Espinar	-
Métodos de los DAOs y ActionListeners (y métodos <i>update</i> , <i>validation</i> y <i>hasError</i>)		-
Cliente	• Alberto Herrera • Unai Piris	-
Compra	• Steven Mallqui • Beatriz Espinar	-
Pase	• Nicolás Espinosa • Axel Kahou	-
Película	• Alfonso Burguete • Daniel Barroso	-
Productora	• Santiago Moral • Sara Sacó	-

TAREA	AUTOR	FECHA
Sala	· Beatriz Espinar	-
Corregir DAOPelícula	· Steven Mallqui · Beatriz Espinar	03/11-05/11
Testeo		03/11-09/11
Depurar	· Santiago Moral · Steven Mallqui	-
Test JUnit AS		-
Cliente	· Nicolás Espinosa	-
Compra	· Alfonso Burguete Gotor	-
Pase	· Nicolás Espinosa · Axel Kahou	-
Película	· Nicolás Espinosa · Alfonso Burguete Gotor	-
Productora	· Santiago Moral · Sara Sacó · Nicolás Espinosa	-
Sala	· Nicolás Espinosa	-
Test JUnit DAO		-
Cliente	· Axel Kahou	-
Compra	· Axel Kahou · Nicolás Espinosa	-
Pase	· Nicolás Espinosa · Axel Kahou	-
Película	· Santiago Moral · Nicolás Espinosa	-
Productora	· Santiago Moral · Sara Sacó · Nicolás Espinosa	-
Sala	· Nicolás Espinosa	-
Pruebas de aceptación	· Beatriz Espinar	-
Corrección tras pruebas de aceptación	· Beatriz Espinar · Steven Mallqui	-
Memoria		06/11-09/11
Apuntar progreso y reparto de tareas	· Beatriz Espinar	12/10-06/11
Redacción apartados 1, 2, 3, 5, 6 y 8	· Beatriz Espinar	-

TAREA	AUTOR	FECHA
Capturas ejemplos	<ul style="list-style-type: none"> • Miriam Choy • Beatriz Espinar • Sara Sacó • Alberto Herrera 	-
Redacción apartado 4	<ul style="list-style-type: none"> • Beatriz Espinar • Miriam Choy • Santiago Moral 	-
Redacción apartado 7	<ul style="list-style-type: none"> • Nicolás Espinosa • Beatriz Espinar 	-
Formulario de presentación	<ul style="list-style-type: none"> • Beatriz Espinar 	-
Otros		06/11-09/11
Diagrama de despliegue	<ul style="list-style-type: none"> • Steven Mallqui 	-
Diagrama clase Error Handler	<ul style="list-style-type: none"> • Alberto Herrera 	-
Corregir casos de uso modificar	<ul style="list-style-type: none"> • Beatriz Espinar • Nicolás Espinosa • Santiago Moral • Alberto Herrera • Unai Piris • Sara Sacó 	-
Diagramas secuencia negocio casos modificar	<ul style="list-style-type: none"> • Alberto Herrera • Sara Sacó 	-
Nuevo caso de uso VerResumenCompra en la SRS	<ul style="list-style-type: none"> • Beatriz Espinar 	-
Modelo VerResumenCompra	<ul style="list-style-type: none"> • Beatriz Espinar • Steven Mallqui 	-
Código VerResumenCompra	<ul style="list-style-type: none"> • Beatriz Espinar • Steven Mallqui 	-
Revisión modelo	<ul style="list-style-type: none"> • Alberto Herrera • Steven Mallqui • Sara Sacó 	-

2.3. Factores de calificación

A continuación, se indica el factor de calificación que el equipo de *Plateau* ha decidido asignar a cada miembro del equipo. Se cuenta con 12 puntos, a repartir entre los 12 integrantes, en función de la cantidad de trabajo realizado. Durante esta primera parte del proyecto, el equipo ha tomado la decisión por mayoría de otorgar 1 punto a cada uno.

Nombre y Apellidos	Factor de calificación
Alberto Herrera García	1
Alfonso Burguete Gotor	1
Axel Kahou He Expósito	1
Beatriz Espinar Aragón	1
Daniel Barroso Casado	1
Jose María Gálvez Gómez	1
Miriam Choy Castillo	1
Nicolás Espinosa Mooser	1
Santiago Moral Santorio	1
Sara Sacó Baños	1
Steven Mallqui Aguilar	1
Unai Piris Ibáñez	1

3. Producto

3.1. Casos de uso

El equipo de *Plateau* ha logrado completar¹ el 100% de las funcionalidades que se pretendían implementar tras la SRS. Estas funcionalidades corresponden a los siguientes casos de uso²:

Módulo Película

- | | |
|----------------------|-----------------------------------|
| ➤ Alta película | ➤ Listar películas |
| ➤ Baja película | ➤ Listar películas por productora |
| ➤ Modificar película | ➤ Añadir productora a película |
| ➤ Mostrar película | ➤ Quitar productora a película |

Módulo Productora

- | | |
|------------------------|-------------------------------------------|
| ➤ Alta productora | ➤ Listar productoras |
| ➤ Baja productora | ➤ Listar productoras por película |
| ➤ Modificar productora | ➤ Calcular la productora con más ingresos |
| ➤ Mostrar productora | |

Módulo Sala

- | | |
|------------------|---------------------------|
| ➤ Alta sala | ➤ Listar salas |
| ➤ Baja sala | ➤ Listar salas estándar |
| ➤ Modificar sala | ➤ Listar salas especiales |
| ➤ Mostrar sala | |

Módulo Cliente

- | | |
|---------------------|----------------------------|
| ➤ Alta cliente | ➤ Listar clientes |
| ➤ Baja cliente | ➤ Listar clientes normales |
| ➤ Modificar cliente | ➤ Listar clientes VIP |
| ➤ Mostrar cliente | |

Módulo Pase

- | | |
|------------------|-----------------------------|
| ➤ Alta pase | ➤ Listar pases |
| ➤ Baja pase | ➤ Listar pases por sala |
| ➤ Modificar pase | ➤ Listar pases por película |
| ➤ Mostrar pase | ➤ Listar pases por compra |

Módulo Compra

- | | |
|------------------------|-------------------------|
| ➤ Abrir compra | ➤ Quitar pase de compra |
| ➤ Añadir pase a compra | ➤ Cerrar compra |

¹ Completar se define como tener elaborado el modelo, tener un código funcional que implemente la funcionalidad, y haber pasado todas las pruebas descritas en el apartado 7 de este documento.

² El detalle de lo que hace cada caso se puede consultar en el documento de Especificación de Requisitos Software.

- Devolver compra
- Mostrar compra
- Mostrar compra en detalle
- Listar compras
- Listar compras por cliente
- Listar compras por pase
- Calcular nº de compras que incluyen una película dada
- Ver resumen de compra

3.2. Interfaces externas

A continuación, se adjuntan capturas del producto final para algunos de sus casos de uso, incluyendo la vista inicial al seleccionar el módulo Pase (Ilustración 1), el caso de uso Alta Sala, en el caso particular de una sala especial (Ilustración 2), el caso de uso Modificar Cliente, en el caso particular de un cliente VIP (Ilustración 3), el caso de uso Listar Productoras (Ilustración 4) y el caso de uso Mostrar Compra en detalle (Ilustración 5).

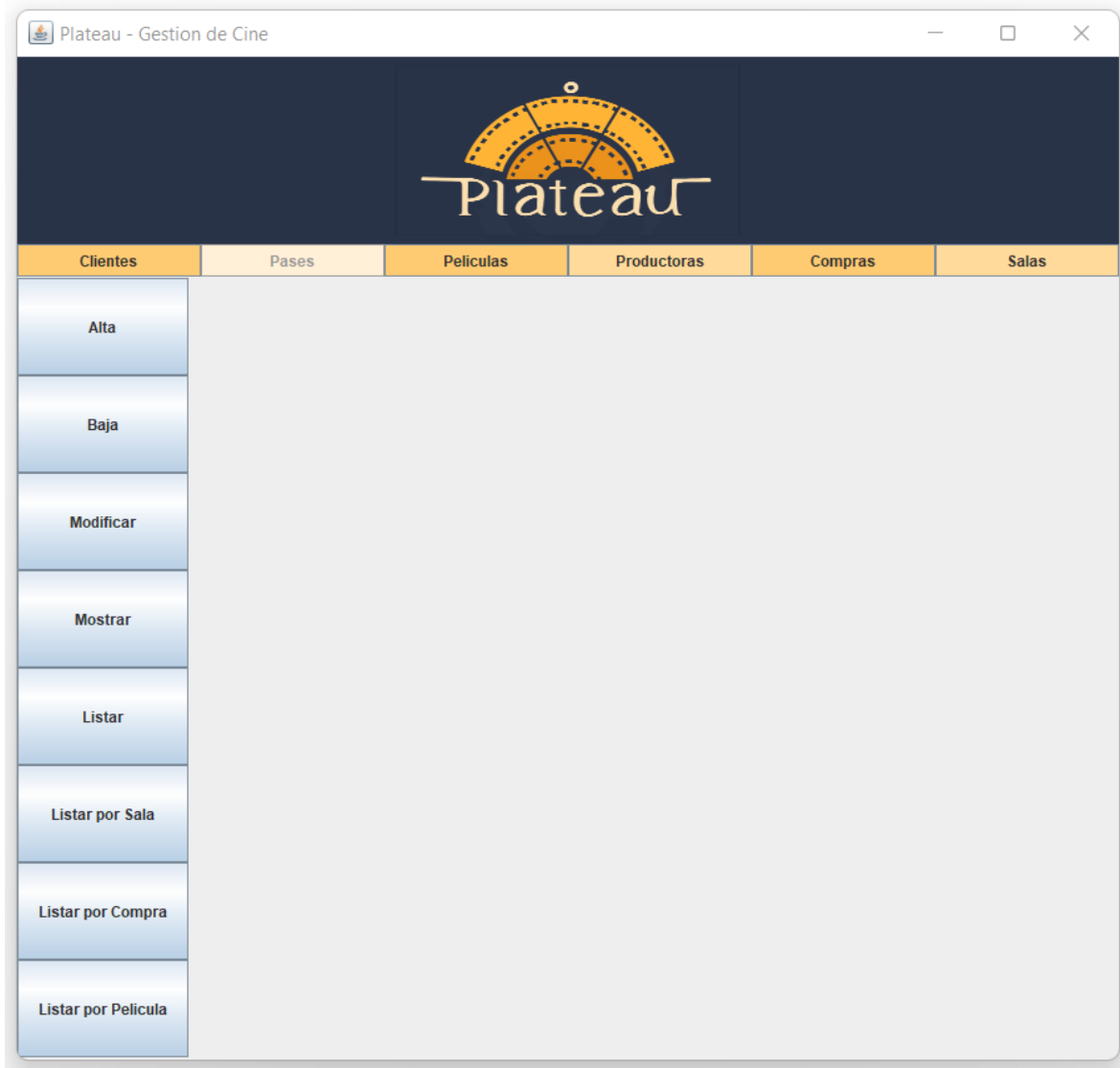



Ilustración 1. Interfaz principal Pase



Cientes

Pases

Peliculas

Productoras

Compras

Salas

Alta

Baja

Modificar

Mostrar

Listar

Listar Estandar

Listar Especiales

ALTA SALA

Nombre:

Tipo:

ESPECIAL

Aforo:

50

☐ 3D
 ☒ VO

Aceptar

Ilustración 2. Interfaz Alta Sala

Ilustración 3. Interfaz Modificar Cliente VIP

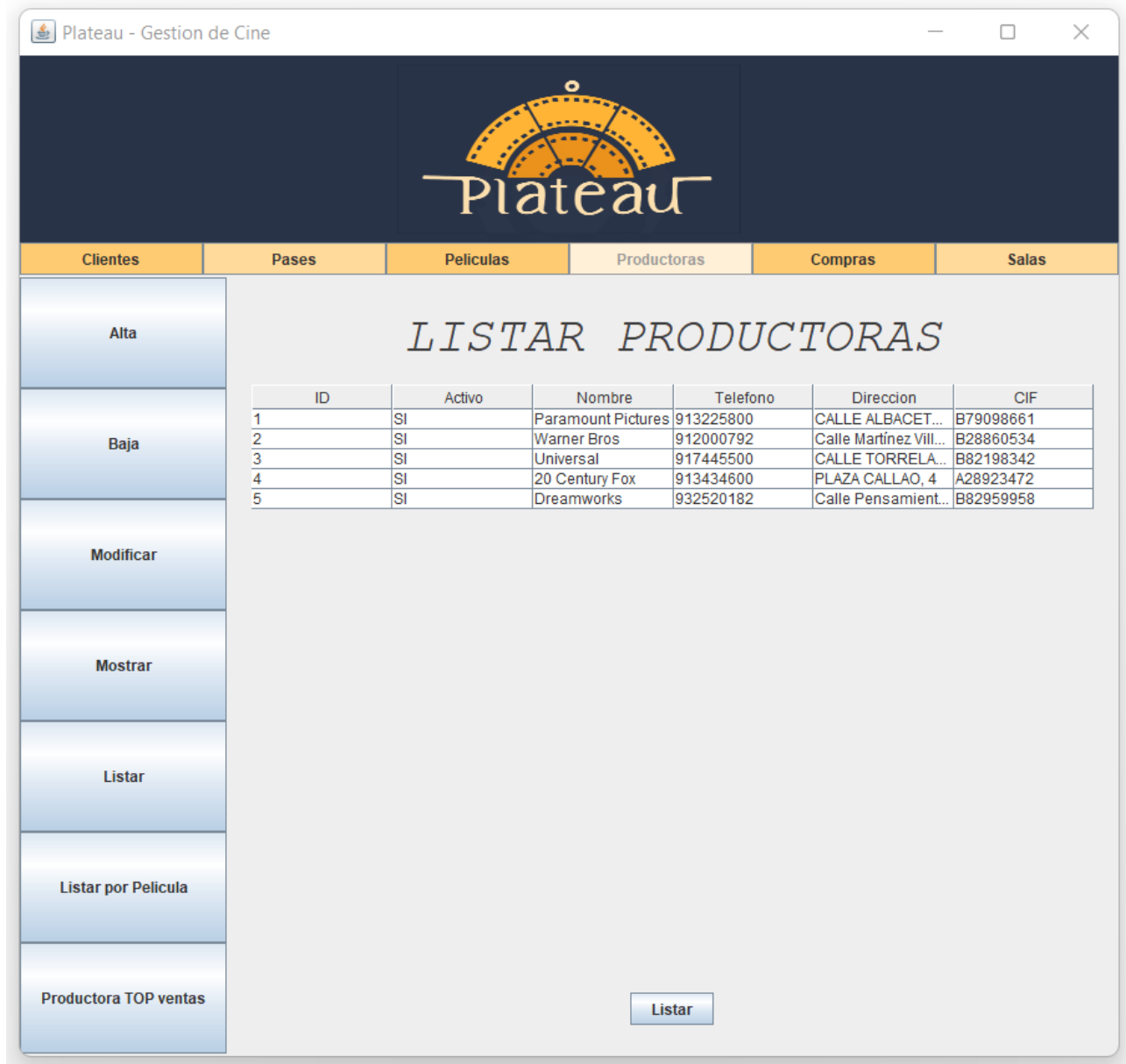


Ilustración 4. Interfaz Listar Productoras

Ilustración 5. Interfaz Mostrar Compra En Detalle

4. Arquitectura

La arquitectura empleada en el proyecto ha sido multicapa, que divide el software en 3 capas principales (aunque también podrían considerarse la de clientes y la de recursos):

1. Capa de presentación. Encapsula toda la lógica de presentación necesaria para dar servicio a los clientes que acceden al sistema. En esta capa se implementa la interfaz gráfica y los correspondientes *ActionListener* de los componentes que crean los *transfers* y los envían al controlador, también incluido en esta capa y que sirve como intermediario entre esta capa y la de negocio. Además, se incluyen en esta capa los comandos, encargados de invocar al servicio de aplicación, traduciendo el contexto que recibe del controlador.
2. Capa de negocio. Proporciona los servicios del sistema e implementa la lógica del negocio (procesamiento, *workflow*, reglas de negocio y datos). En esta capa se encuentran los *transfers* que representan las entidades del sistema, y los servicios de aplicación, encargados de implementar dicha lógica y asegurar que los datos de entrada cumplen las reglas de negocio y los requisitos del sistema. Para poder realizar estas operaciones hará uso de la capa de integración, que se comunicará con el almacén persistente.
3. Capa de integración. Responsable de la comunicación con los recursos y sistemas externos. En el caso de *Plateau*, interactúa con la base de datos, para lo cual hace uso de los DAOs, que extraen y guardan los datos en el almacén persistente. También se implementa en esta capa toda la gestión de transacciones, así como las queries, que se explicarán en el siguiente apartado al igual que otros patrones mencionados anteriormente.

Así, mediante esta arquitectura, se promueve el aislamiento entre las capas, cada una encargada de su ámbito, de forma que los cambios, como modificar una regla de negocio, no afectan a las demás. También favorece la encapsulación y la reutilización del código, y proporciona mayor seguridad y un sistema con mayor rendimiento y muy empaquetado.

4.1. Patrones

Seguidamente se describen brevemente los patrones software empleados en este proyecto, algunos de los cuales fueron ya mencionados en la Especificación de Requisitos Software, pues se trataban de requerimientos imprescindibles a nivel lógico del sistema.

❖ Patrón *Data Access Object* (DAO)

Utilizamos este componente software como interfaz entre nuestro proyecto y el SGBDR, se utiliza en todos los módulos y soporta las operaciones básicas (Create, Read, Update, Delete) y alguna adicional necesaria (véase listar, listar por parámetro..), Así, encapsulamos en un objeto Transfer la información de la Base de Datos (de la cual el resto de capas no tienen por qué tener información de su implementación).

❖ Patrón *Transfer Object*

Es un conocido patrón j2ee que porta datos sobre alguna entidad del modelo entre capas, encapsulando sus atributos. Es utilizado (con operaciones mutadoras y accesoras) en todo nuestro proyecto, para guardar en objetos nuestras entidades del modelo.

❖ Patrón *Transfer Object Assembler* (TOA)

Se trata de un patrón que compone objetos Transfer a partir de otras fuentes de datos (comúnmente, otros transfers), así, simplifica algunas peticiones costosas haciendo menos llamadas. En nuestro proyecto, el módulo Compra hace uso de este patrón para evitar concatenaciones de Transfers, como proponía Fowler.

❖ Patrón *Application Service* (AS)

Se trata de un patrón que hemos aplicado en la capa de negocio, para centralizar la lógica de nuestro programa. Evalúa las condiciones estipuladas en nuestros casos de uso, a partir de un transfer generado en los ActionListener de cada panel. Los casos de uso tienen asociada una operación de negocio en cada Servicio de aplicación particular (particular a su módulo). Además, añade una capa más a negocio y permite un código reutilizable y no duplicado.

❖ Patrón *Service to Worker*

La finalidad de este patrón es llevar a cabo el manejo de peticiones y la invocación de lógica de negocio antes de pasar el control a la vista. *Service to Worker* articula diversos patrones de presentación (controlador frontal, controlador de aplicación, ayudante de vista). Sin embargo, en esta parte del proyecto no se ha implementado la versión original del patrón, sino una adaptada. Por ejemplo, no se utiliza el patrón *ViewHelper* ni el *FrontController*.

❖ Patrón *Context*

La finalidad de este patrón es evitar utilizar información del sistema específica del protocolo fuera de su contexto relevante, se encarga de encapsular el estado de una forma independiente del protocolo para ser compartida por toda la aplicación

❖ Patrón *Command*

La finalidad de este patrón es encapsular una petición en un objeto, permitiendo así parametrizar a los clientes con diferentes peticiones, hacer cola o llevar un registro de las peticiones, y poder deshacer las operaciones.

❖ Patrón *Query*

Es un objeto que representa una consulta de base de datos, el beneficio que nos ofrece este patrón es que abstrae por completo el lenguaje de consulta de la base de datos subyacente, así mantiene las preocupaciones de la infraestructura de la persistencia y la recuperación de datos fuera de la capa de negocio.

❖ Patrón *Singleton*

La finalidad de este patrón es garantizar que una clase solo tenga una instancia (o un número controlado de instancias) proporcionando un punto de acceso global a ella.

❖ Patrón *Abstract Factory*

La finalidad de este patrón es proporcionar una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

5. Modelo

En esta sección se explica el modelo realizado, con un listado de los diagramas que se han elaborado en el proyecto y algún ejemplo visual.

5.1. Diagramas de clase

Por un lado, se han realizado diagramas de clase. Estos se han dividido en 3 carpetas, correspondientes a las 3 capas de la arquitectura multicapa: Integración, Negocio y Presentación.

➤ Integración

Se ha creado una carpeta para cada módulo, para tener organizados los diagramas. Aparte de los 6 módulos (Cliente, Compra, Pase, Película, Productora y Sala) se han añadido carpetas adicionales para el DAOFactory, la QueryFactory y las transacciones, quedando la estructura de carpetas que puede verse en la Ilustración 6.

Así, se ha elaborado un diagrama de integración para cada módulo, quedando los diagramas que se listan a continuación. Se adjuntan posteriormente una captura del diagrama de Sala (Ilustración 7) y el diagrama de transacciones (Ilustración 8), a modo de ejemplo.

- *Modelos/Modelo de Diseño/Integración/Cliente/Diagrama de Integración – Cliente*
- *Modelos/Modelo de Diseño/Integración/Compra/Diagrama de Integración – Compra*
- *Modelos/Modelo de Diseño/Integración/DAOFactory/Diagrama de Integración – DAOFactory*
- *Modelos/Modelo de Diseño/Integración/Pase/Diagrama de Integración – Pase*
- *Modelos/Modelo de Diseño/Integración/Pelicula/Diagrama de Integración – Pelicula*
- *Modelos/Modelo de Diseño/Integración/Productora/Diagrama de Integración – Productora*
- *Modelos/Modelo de Diseño/Integración/QueryFactory/Diagrama de Integración – QueryFactory*
- *Modelos/Modelo de Diseño/Integración/Sala/Diagrama de Integración – Sala*
- *Modelos/Modelo de Diseño/Integración/Transactions/Diagrama de Integración – Transactions*



Ilustración 6. Estructura Diagramas Integración

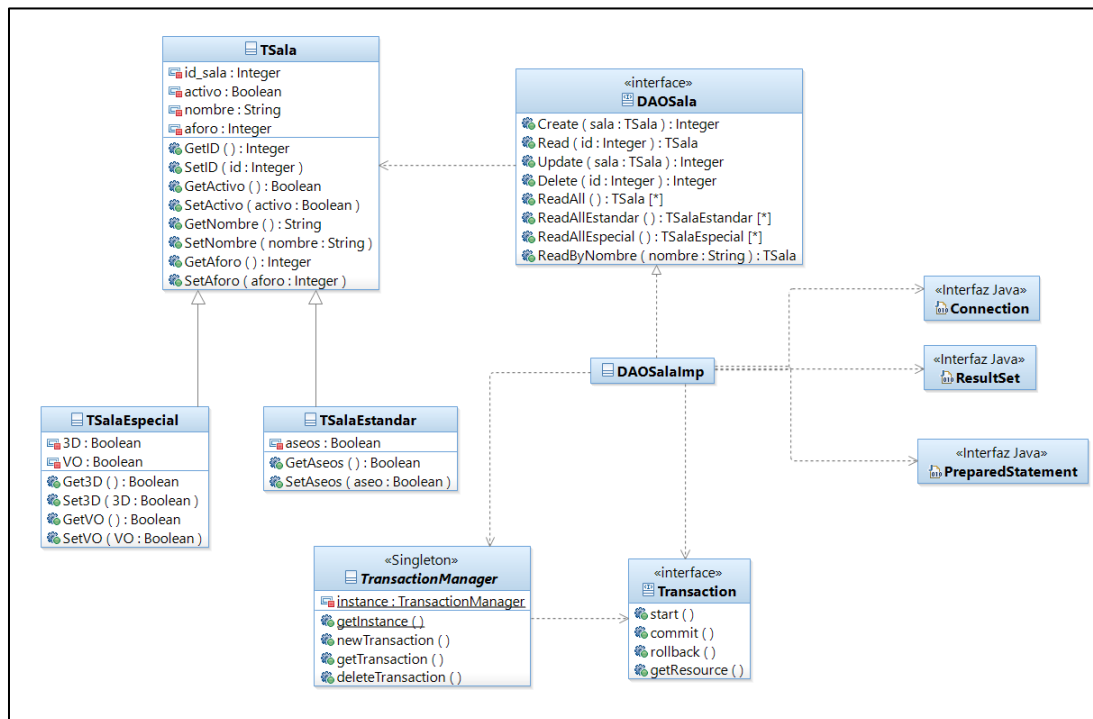


Ilustración 7. Diagrama Clase Integración Sala

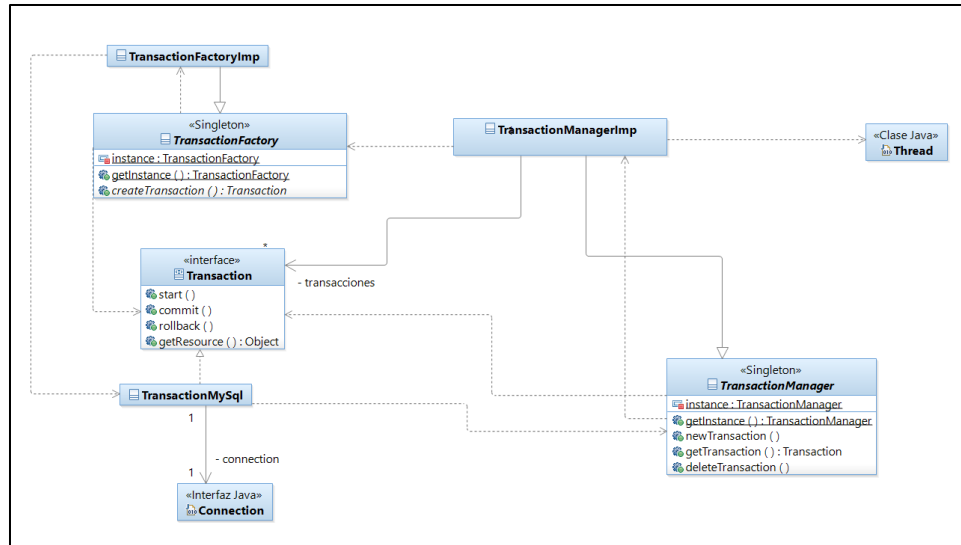


Ilustración 8. Diagrama Clase Transactions

➤ Negocio

Se ha creado una carpeta para cada módulo, para tener organizados los diagramas. Aparte de los 6 módulos (Cliente, Compra, Pase, Película, Productora y Sala) se ha añadido una carpeta adicional para el ASFactory, quedando la estructura de carpetas que puede verse en la Ilustración 9.

Así, se ha elaborado un diagrama de negocio para cada módulo, quedando los diagramas que se listan a continuación. Se adjuntan posteriormente una captura del diagrama de Película (Ilustración 10) y el diagrama del ASFactory (Ilustración 11), a modo de ejemplo.

- *Modelos/Modelo de Diseño/Negocio/ASFactory/Diagrama de Negocio – ASFactory*
- *Modelos/Modelo de Diseño/Negocio/Cliente/Diagrama de Negocio – Cliente*
- *Modelos/Modelo de Diseño/Negocio/Compra/Diagrama de Negocio – Compra*
- *Modelos/Modelo de Diseño/Negocio/Pase/Diagrama de Negocio – Pase*
- *Modelos/Modelo de Diseño/Negocio/Pelicula/Diagrama de Negocio – Pelicula*
- *Modelos/Modelo de Diseño/Negocio/Productora/Diagrama de Negocio – Productora*
- *Modelos/Modelo de Diseño/Negocio/Sala/Diagrama de Negocio – Sala*

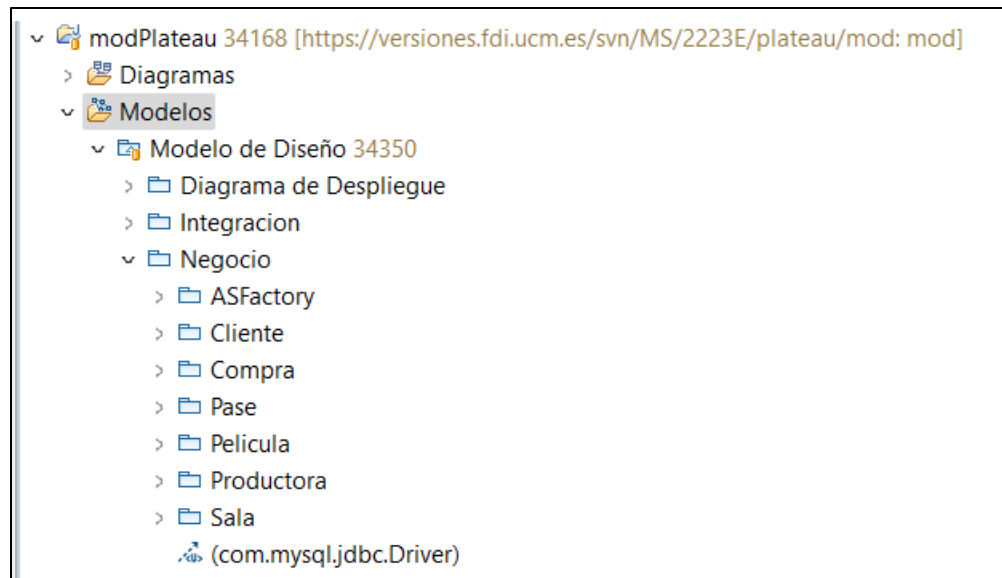


Ilustración 9. Estructura Diagramas Negocio

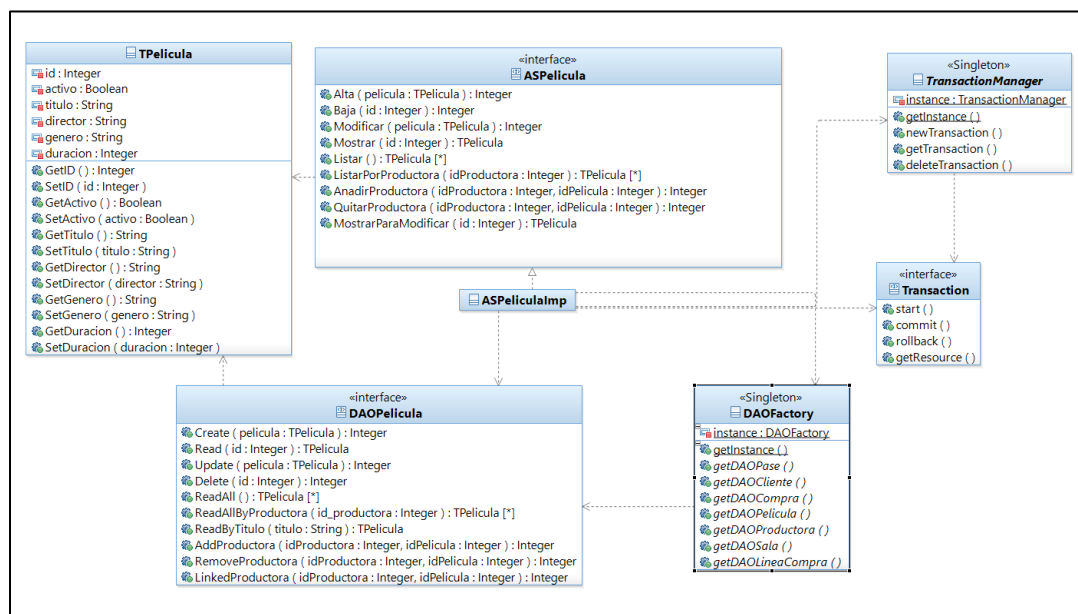


Ilustración 10. Diagrama Clase Negocio Pelicula

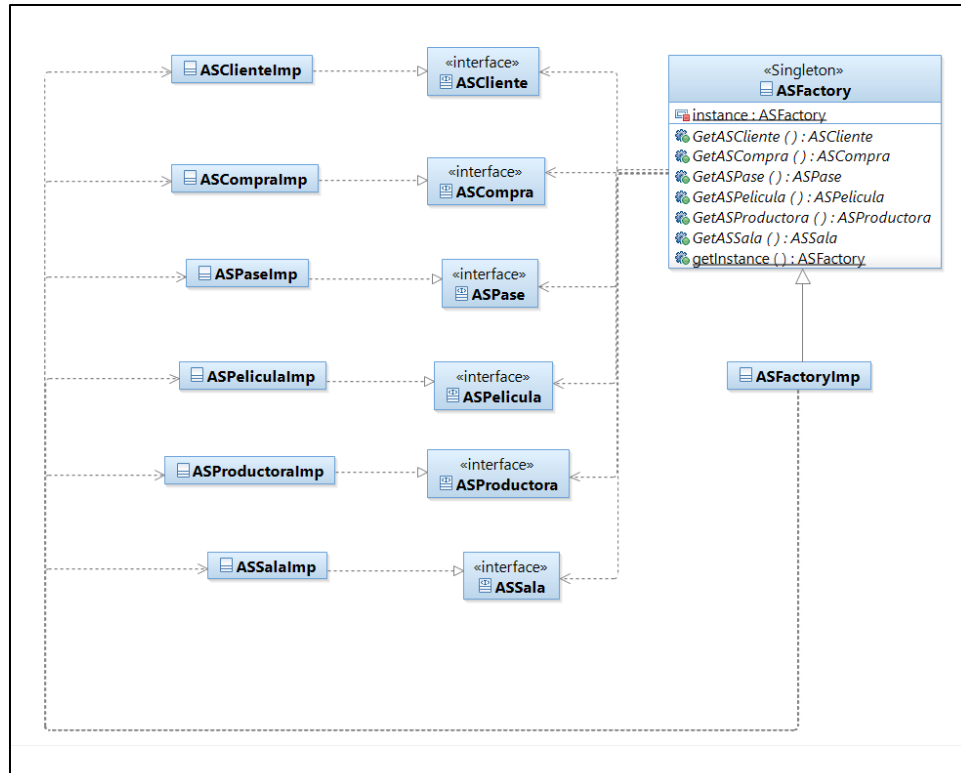


Ilustración 11. Diagrama Clase ASFactory

➤ Presentación

En primer lugar, se han creado 3 carpetas principales: Command, Controller y GUI. En la carpeta Command, aparece una carpeta por cada uno de los 6 módulos, con sus correspondientes diagramas, y el diagrama de presentación de Command. Además, en la carpeta Controller aparece su respectivo diagrama. Por último, en GUI, aparece el diagrama Main, una carpeta con el modelo del ErrorHandler, y otra llamada Panels que, de nuevo, se subdivide en 6 carpetas, cada una correspondiente a un módulo. Así pues, queda la estructura de carpetas que se puede ver en las ilustraciones 12 y 13, y se han realizado los diagramas que se listan a continuación. Se adjuntan posteriormente una captura del diagrama de CommandCliente (Ilustración 14), el diagrama del Controller (Ilustración 15), el diagrama Main (Ilustración 16), el diagrama del ErrorHandler (Ilustración 17) y el diagrama de PanelCompra (Ilustración 18), a modo de ejemplo.

- Modelos/Modelo de Diseño/Presentacion/Command/Diagrama de Presentación – Command
- Modelos/Modelo de Diseño/Presentacion/Command/Cliente/Diagrama de Presentación – CommandCliente
- Modelos/Modelo de Diseño/Presentacion/Command/Compra/Diagrama de Presentación – CommandCompra
- Modelos/Modelo de Diseño/Presentacion/Command/Pase/Diagrama de Presentación – CommandPase
- Modelos/Modelo de Diseño/Presentacion/Command/Pelicula/Diagrama de Presentación – CommandPelicula

- *Modelos/Modelo de Diseño/Presentacion/Command/Productora/Diagrama de Presentación – CommandProductora*
- *Modelos/Modelo de Diseño/Presentacion/Command/Sala/Diagrama de Presentación – CommandSala*
- *Modelos/Modelo de Diseño/Presentacion/Controller/ Diagrama de Presentación – Controller*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Main*
- *Modelos/Modelo de Diseño/Presentacion/Gui/ErrorHandler/ErrorHandler*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Cliente/Diagrama de Presentación – PanelsCliente*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Compra/Diagrama de Presentación – PanelsCompra*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Pase/Diagrama de Presentación – PanelsPase*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Pelicula/Diagrama de Presentación – PanelsPelicula*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Productora/Diagrama de Presentación – PanelsProductora*
- *Modelos/Modelo de Diseño/Presentacion/Gui/Panels/Sala/Diagrama de Presentación – PanelsSala*

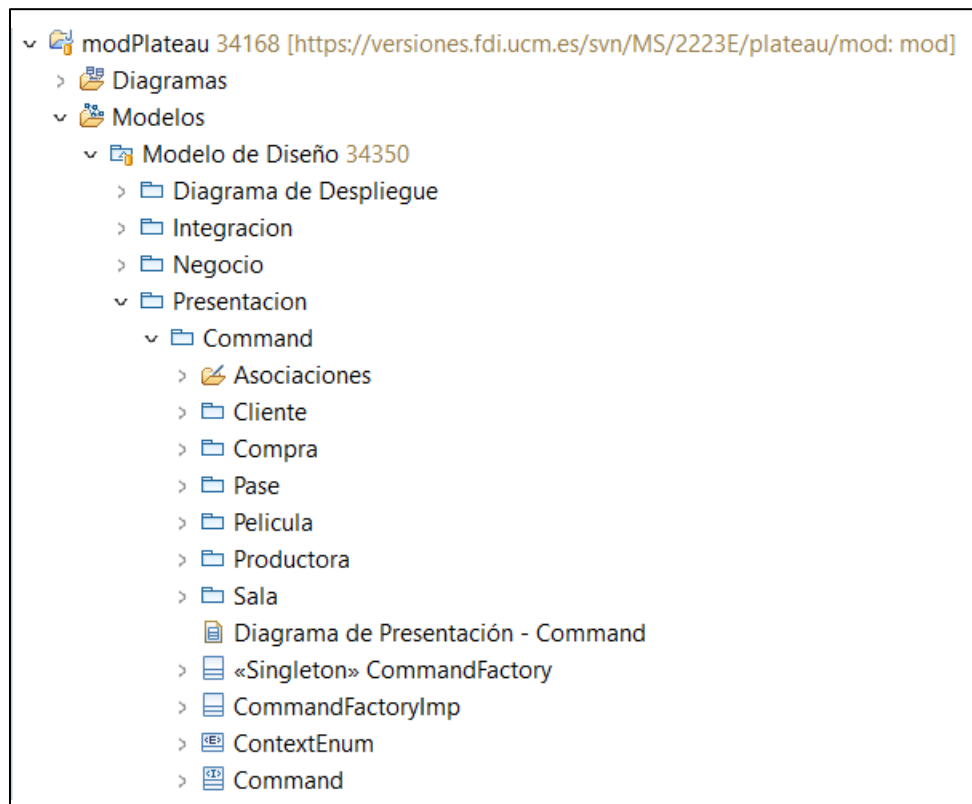


Ilustración 12. Estructura Diagramas Presentación (1)

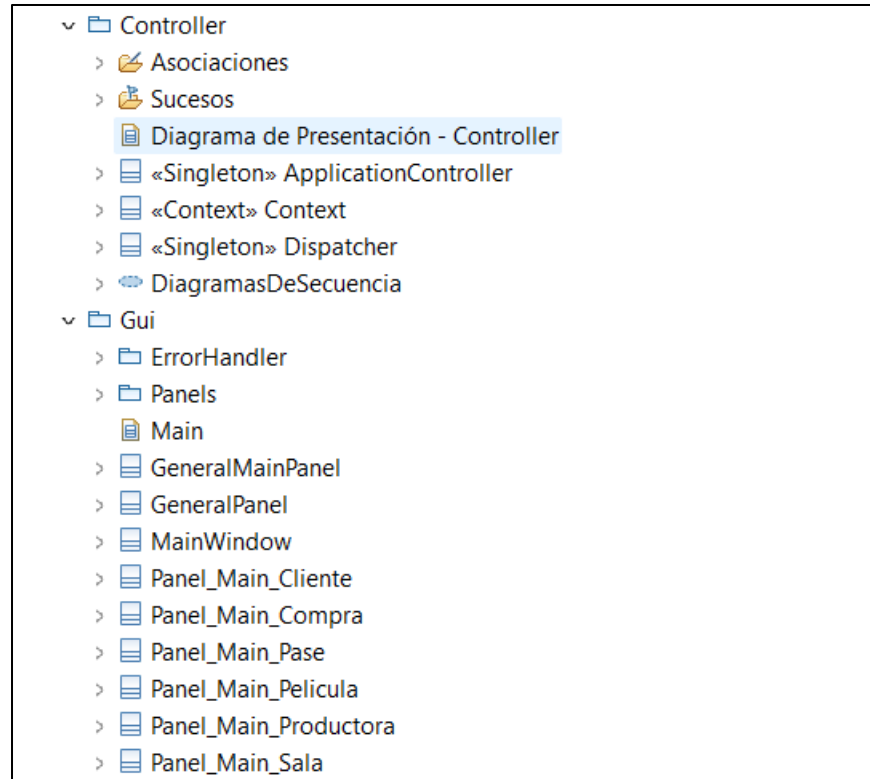


Ilustración 13. Estructura Diagramas Presentación (2)

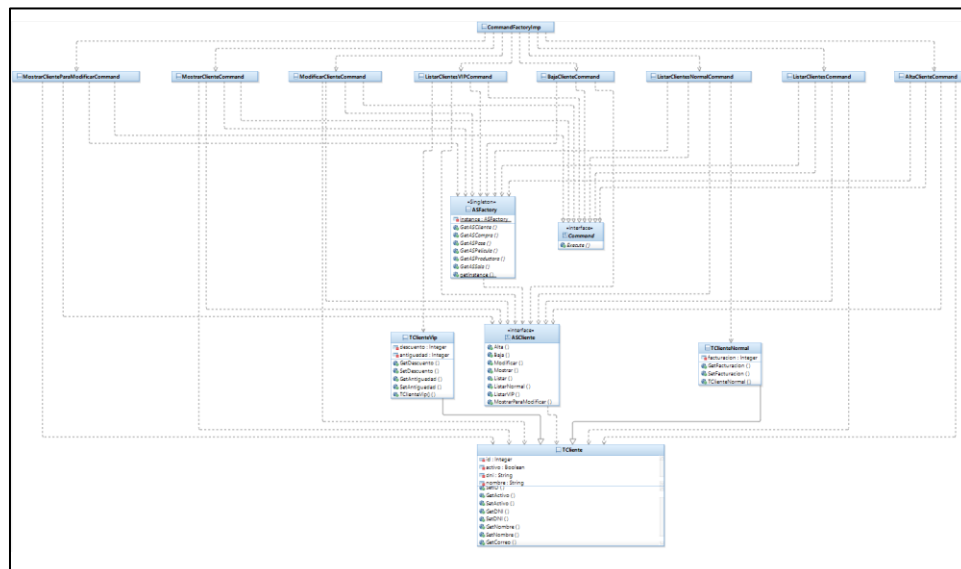


Ilustración 14. Diagrama Clase CommandCliente

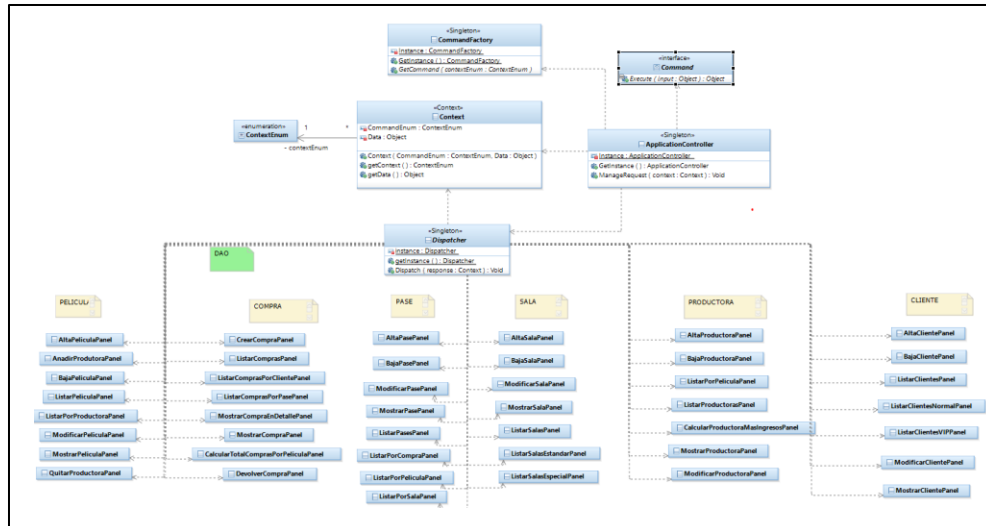


Ilustración 15. Diagrama Clase Controller

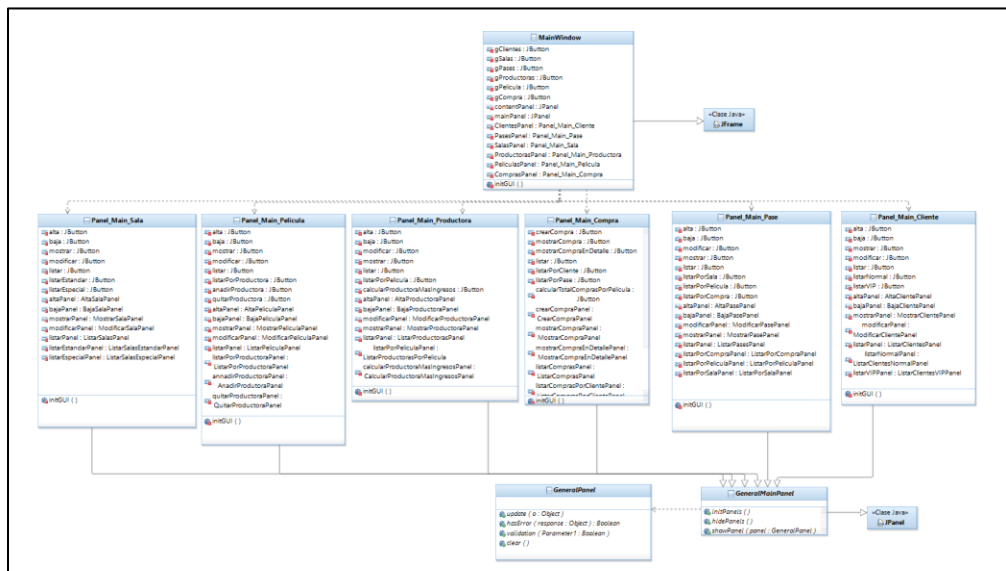


Ilustración 16. Diagrama Clase Main

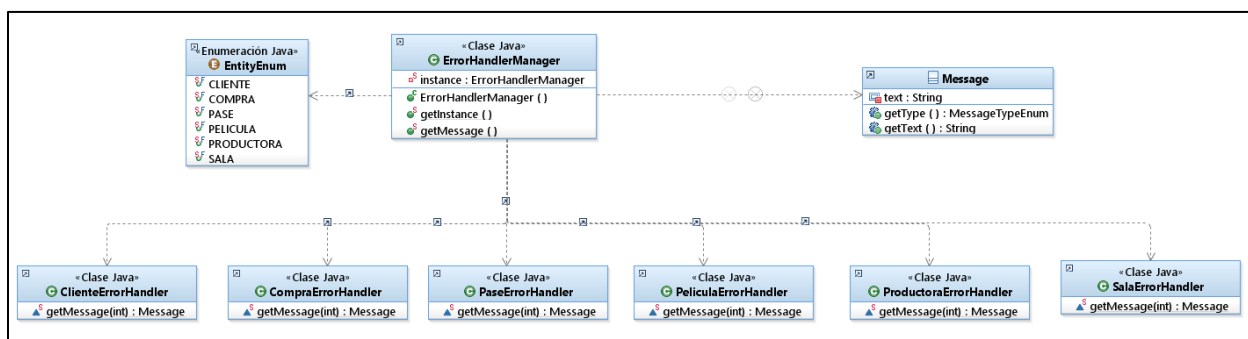


Ilustración 17. Diagrama Clase ErrorHandler

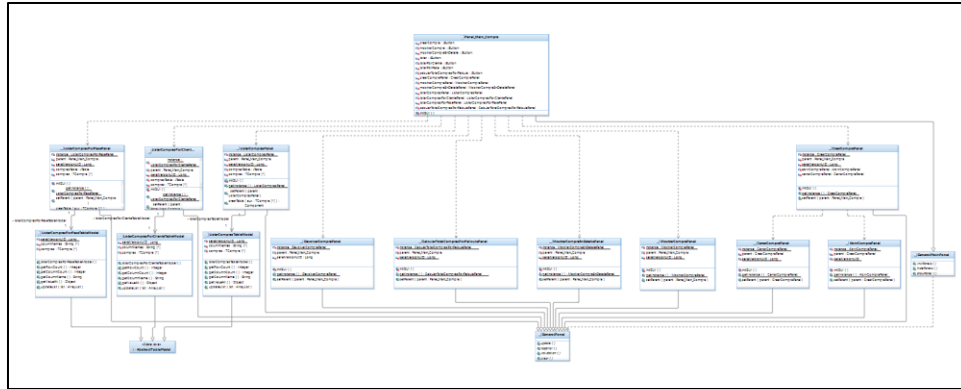


Ilustración 18. Diagrama Clase CompraPanel

5.2. Diagramas de secuencia

Para los diagramas de secuencia también se ha realizado la misma división en 3 carpetas, correspondientes a las 3 capas de la arquitectura multicapa: Integración, Negocio y Presentación.

► Integración

En este caso, no se han realizado los diagramas de secuencia de integración de todos los módulos, puesto que varios resultaban innecesarios al tener excesivas similitudes con los demás. Por ello, se han seleccionado 3 módulos que destacaban por los siguientes motivos: en primer lugar, se han realizado diagramas de Película, pues esta es una entidad básica y sirve como modelo para las demás. Por otro lado, se han realizado los diagramas de Cliente³, como ejemplo de entidad especializada, sirviendo como modelo para Sala, que también es especializada. Por último, se han realizado los de Compra⁴, por ser una entidad claramente distinguible de las demás, que opera de otro modo.

Aparte de estos diagramas correspondientes a los módulos, se han elaborado los diagramas de las queries y de las transacciones, quedando entonces el listado de diagramas que sigue. Se adjuntan posteriormente una captura del diagrama de AltaCliente (Ilustraciones 19, 20 y 21), el diagrama de ListarPeliculas (Ilustraciones 22 y 23) y el diagrama de NewTransaction (Ilustración 24), a modo de ejemplo.

- *Modelos/Modelo de Diseño/Integracion/Cliente/DiagramasDeSecuencia/AltaCliente*
- *Modelos/Modelo de Diseño/Integracion/Cliente/DiagramasDeSecuencia/BajaCliente*
- *Modelos/Modelo de Diseño/Integracion/Cliente/DiagramasDeSecuencia/ListarClientes⁵*
- *Modelos/Modelo de Diseño/Integracion/Cliente/DiagramasDeSecuencia/ModificarCliente*
- *Modelos/Modelo de Diseño/Integracion/Cliente/DiagramasDeSecuencia/MostrarCliente*
- *Modelos/Modelo de Diseño/Integracion/Compra/DiagramasDeSecuencia/ListarCompras*

³ No se ha elaborado el diagrama correspondiente al método del DAO ReadByDNI porque es similar a otros (sólo es necesario añadir la cláusula “WHERE dni = ?” a la query).

⁴ No se han elaborado los diagramas correspondientes a los métodos Create, Delete o Update porque son análogos a otros casos (AltaPelícula, BajaPelícula, ModificarPelícula).

⁵ No se han elaborado los diagramas de ListarClientesNormales y ListarClientesVIP porque son análogos a este.

- *Modelos/Modelo de Diseño/Integracion/Compra/DiagramasDeSecuencia/ListarComprasPorCliente*
- *Modelos/Modelo de Diseño/Integracion/Compra/DiagramasDeSecuencia/ListarComprasPorPase*
- *Modelos/Modelo de Diseño/Integracion/Compra/DiagramasDeSecuencia/MostrarCompra*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/AltaPelicula*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/AnadirProductoraAPelicula⁶*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/BajaPelicula*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/ListarPeliculas*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/ListarPeliculasPorProductora*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/ModificarPelicula*
- *Modelos/Modelo de Diseño/Integracion/Pelicula/DiagramasDeSecuencia/MostrarPelicula*
- *Modelos/Modelo de Diseño/Integracion/QueryFactory/DiagramasDeSecuencia/CalcularNumComprasPorPelicula*
- *Modelos/Modelo de Diseño/Integracion/QueryFactory/DiagramasDeSecuencia/CalcularProductoraConMasIngresos*
- *Modelos/Modelo de Diseño/Integracion/Transactions/DiagramasDeSecuencia/Commit*
- *Modelos/Modelo de Diseño/Integracion/Transactions/DiagramasDeSecuencia/GetTransaction*
- *Modelos/Modelo de Diseño/Integracion/Transactions/DiagramasDeSecuencia/NewTransaction*
- *Modelos/Modelo de Diseño/Integracion/Transactions/DiagramasDeSecuencia/Rollback*
- *Modelos/Modelo de Diseño/Integracion/Transactions/DiagramasDeSecuencia/Start*

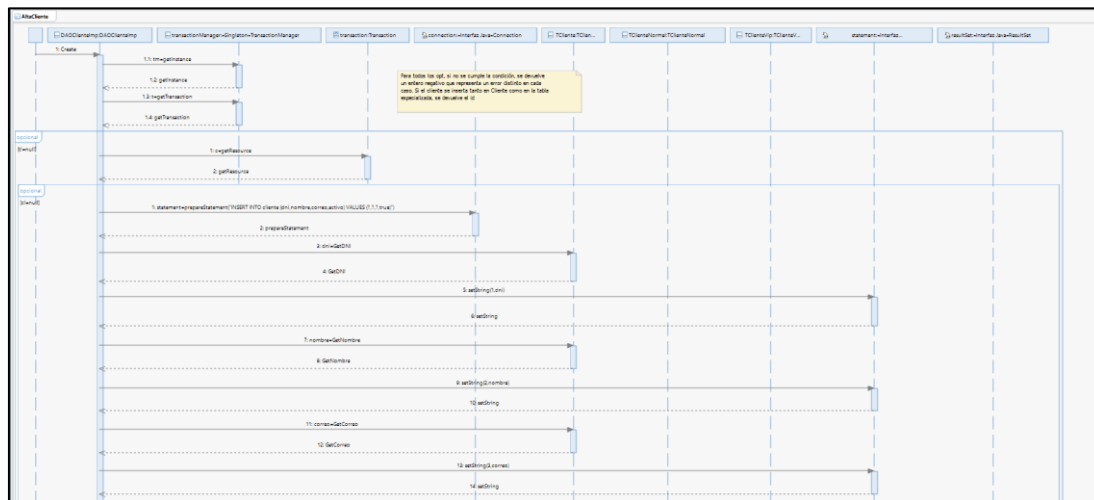


Ilustración 19. Diagrama Secuencia Integración AltaCliente (1)

⁶ No se ha elaborado el diagrama de QuitarProductoraAPelicula porque es análogo a este.



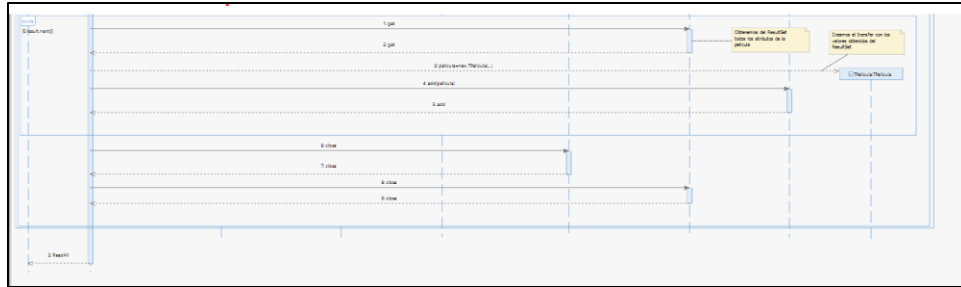


Ilustración 23. Diagrama Secuencia Integración ListarPelículas (2)

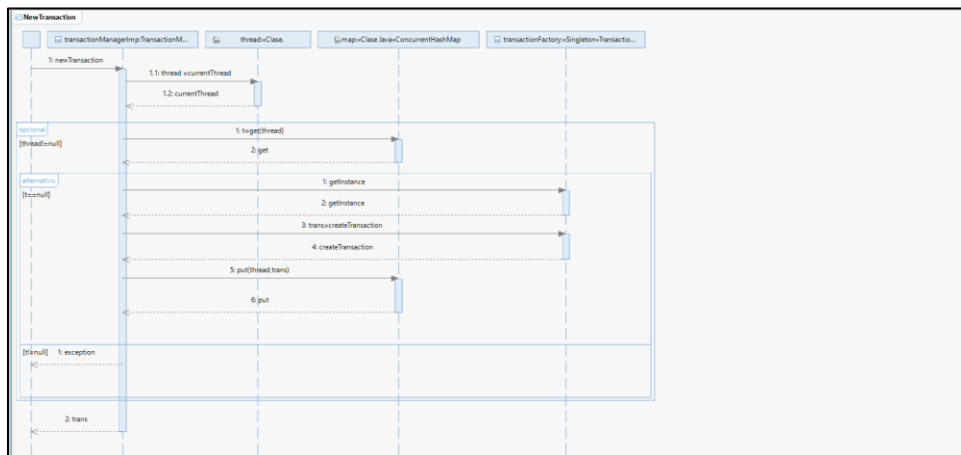


Ilustración 24. Diagrama Secuencia NewTransaction

➤ Negocio

Para esta capa se ha tomado la misma decisión. Se han tomado los módulos Película, Cliente⁷ y Compra como referentes para los demás. Aparte de estos diagramas, se ha creado también el de la query correspondiente al módulo Productora, quedando entonces el listado de diagramas que sigue. Se adjuntan posteriormente una captura del diagrama de AltaCliente (Ilustraciones 25, 26 y 27), el diagrama de DevolverCompra (Ilustraciones 28 y 29) y el diagrama de ListarPelículasPorProductora (Ilustraciones 30 y 31), a modo de ejemplo.

- Modelos/Modelo de Diseño/Negocio/Cliente/DiagramasDeSecuencia/AltaCliente
- Modelos/Modelo de Diseño/Negocio/Cliente/DiagramasDeSecuencia/BajaCliente
- Modelos/Modelo de Diseño/Negocio/Cliente/DiagramasDeSecuencia/ListarClientes⁸
- Modelos/Modelo de Diseño/Negocio/Cliente/DiagramasDeSecuencia/ModificarCliente
- Modelos/Modelo de Diseño/Negocio/Cliente/DiagramasDeSecuencia/MostrarCliente
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/CalcularTotalComprasPorPelícula
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/CerrarCompra

⁷ No se ha elaborado el diagrama correspondiente al método MostrarParaModificar (ni en Cliente ni en Película) porque es análogo al de Mostrar, añadiendo la condición de que la entidad esté activa

⁸ No se han elaborado los diagramas de ListarClientesNormales y ListarClientesVIP porque son análogos a este.

- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/DevolverCompra
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/ListarCompras
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia /ListarComprasPorCliente
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/ ListarComprasPorPase
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/MostrarCompra
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/MostrarCompraEnDetalle
- Modelos/Modelo de Diseño/Negocio/Compra/DiagramasDeSecuencia/VerResumenCompra
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/AltaPelicula
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/ AnadirProductora⁹
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/BajaPelicula
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/ListarPeliculas
- Modelos/Modelo de Diseño/Negocio /Pelicula/DiagramasDeSecuencia/
ListarPeliculasPorProductora
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/ModificarPelicula
- Modelos/Modelo de Diseño/Negocio/Pelicula/DiagramasDeSecuencia/MostrarPelicula
- Modelos/Modelo de Diseño/Negocio/Productora/DiagramasDeSecuencia/
CalcularProductoraConMasIngresos

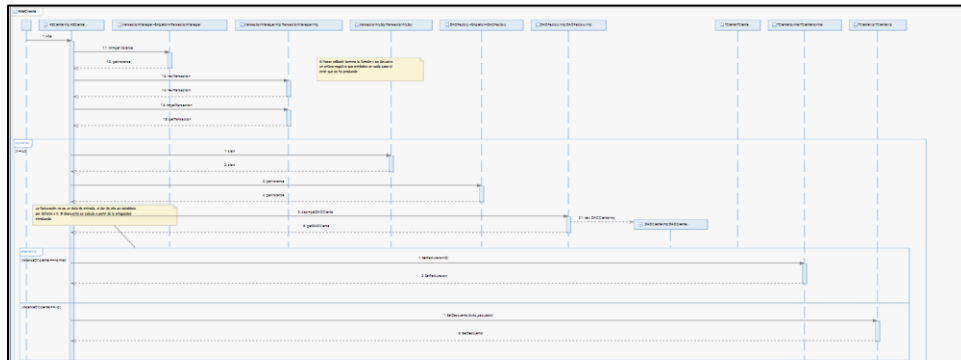


Ilustración 25. Diagrama Secuencia Negocio AltaCliente (1)

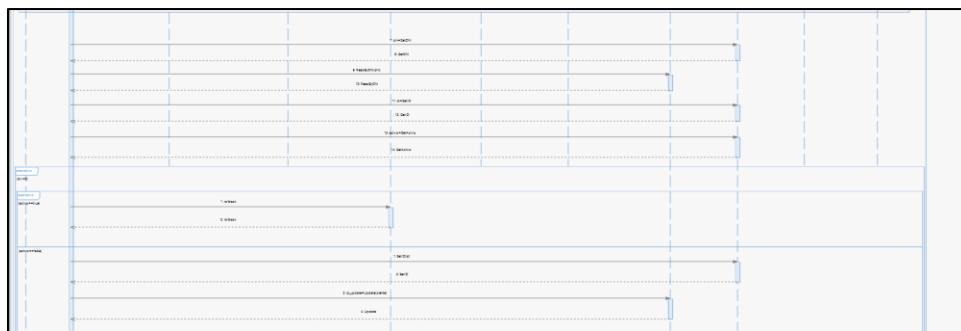


Ilustración 26. Diagrama Secuencia Negocio AltaCliente (2)

⁹ No se ha elaborado el diagrama de QuitarProductora porque es análogo a este.

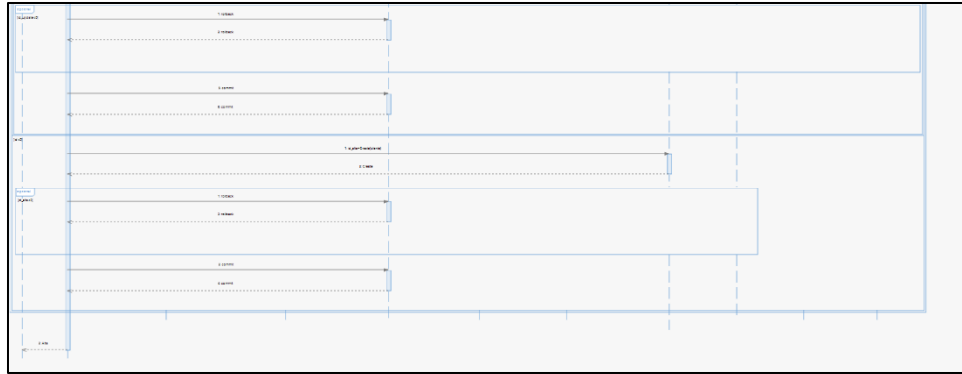


Ilustración 27. Diagrama Secuencia Negocio AltaCliente (3)

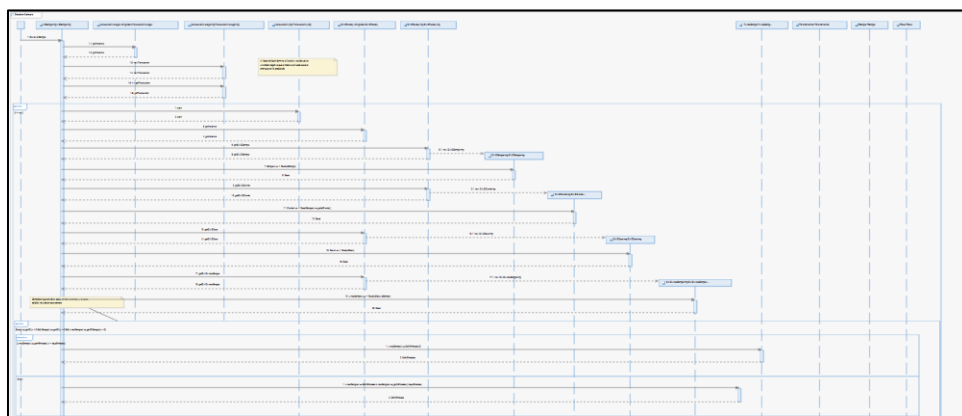


Ilustración 28. Diagrama Secuencia Negocio DevolverCompra (1)

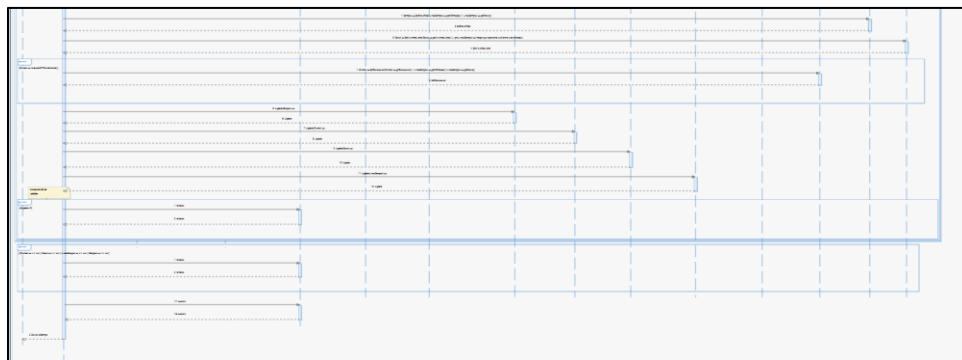


Ilustración 29. Diagrama Secuencia Negocio DevolverCompra (2)

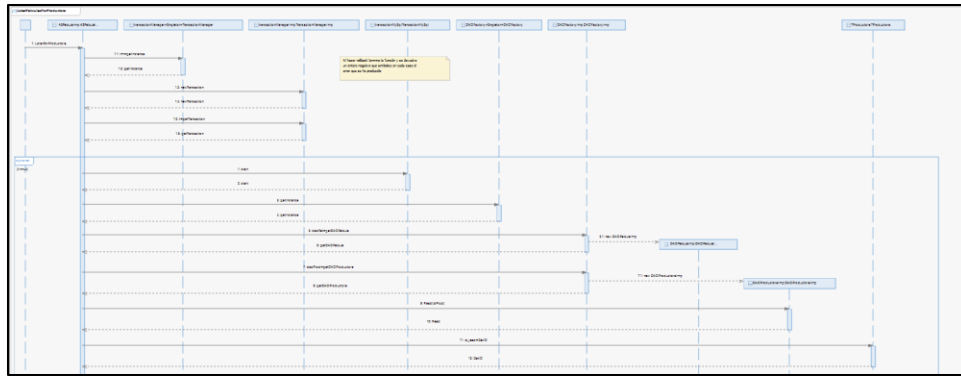


Ilustración 30. Diagrama Secuencia Negocio ListarPeliculasPorProductora (1)

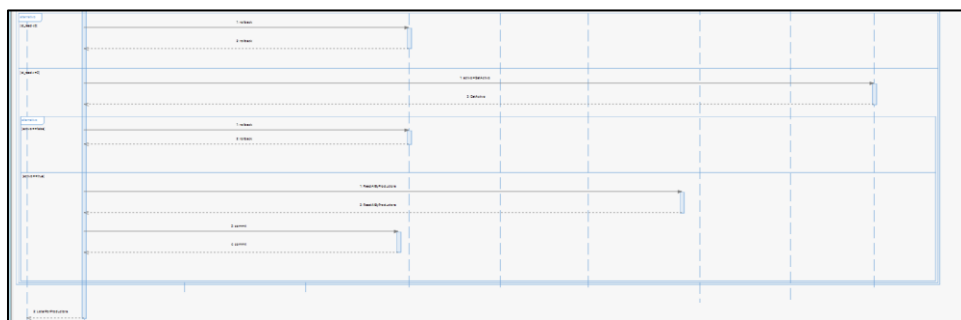


Ilustración 31. Diagrama Secuencia Negocio ListarPeliculasPorProductora (2)

➤ Presentación

Por un lado, en la carpeta Command, se han elaborado los diagramas de los mismos módulos que las otras capas (Cliente, Compra y Película¹⁰). Se ha realizado también el diagrama de secuencia del ApplicationController, y en la carpeta Gui los diagramas de los 3 módulos mencionados. Quedan por tanto los diagramas que se listan a continuación. Se adjuntan posteriormente una captura del diagrama de MostrarCompraEnDetalleCommand (Ilustración 32), el diagrama del ApplicationController (Ilustración 33) y el diagrama de BajaPeliculaPanel (Ilustración 34), a modo de ejemplo.

- Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/AltaClienteCommand
- Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/BajaClienteCommand
- Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/ListarClientesCommand
- Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/ListarClientesNormalCommand
- Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/ListarClientesVipCommand

¹⁰ Excepto el MostrarParaModificarCommand (tanto en Cliente como en Película)

- *Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/ModificarClienteCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Cliente/DiagramasDeSecuencia/MostrarClienteCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/CerrarCompraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/DevolverCompraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/ListarComprasCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/ListarComprasPorClienteCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/ListarComprasPorPaseCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/MostrarCompraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/MostrarCompraEnDetalleCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Compra/DiagramasDeSecuencia/VerResumenCompraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/AltaPeliculaCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/AnadirProductoraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/BajaPeliculaCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/ListarPeliculasCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/ListarPeliculaPorProductoraCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/ModificarPeliculaCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/MostrarPeliculaCommand*
- *Modelos/Modelo de Diseño/Presentación/Command/Pelicula/DiagramasDeSecuencia/QuitarProductoraCommand*
- *Modelos/Modelo de Diseño/Presentación/Controller/ DiagramasDeSecuencia/ ApplicationController*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/ AltaClientePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/ BajaClientePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/ ListarClientesPanel*

- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/ListarClientesVipPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/ModificarClientePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Cliente/DiagramasDeSecuencia/MostrarClientePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/CrearCompraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/DevolverCompraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/ListarComprasPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/ListarComprasPorClientePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/ListarComprasPorPasePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/MostrarCompraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/MostrarCompraEnDetallePanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Compra/DiagramasDeSecuencia/VerResumenCompraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/AltaPeliculaPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/AnadirProductoraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/BajaPeliculaPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/ListarPeliculasPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/ListarPorProductoraPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/ModificarPeliculaPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/MostrarPeliculaPanel*
- *Modelos/Modelo de Diseño/Presentación/Gui/Panels/Pelicula/DiagramasDeSecuencia/QuitarProductoraPanel*

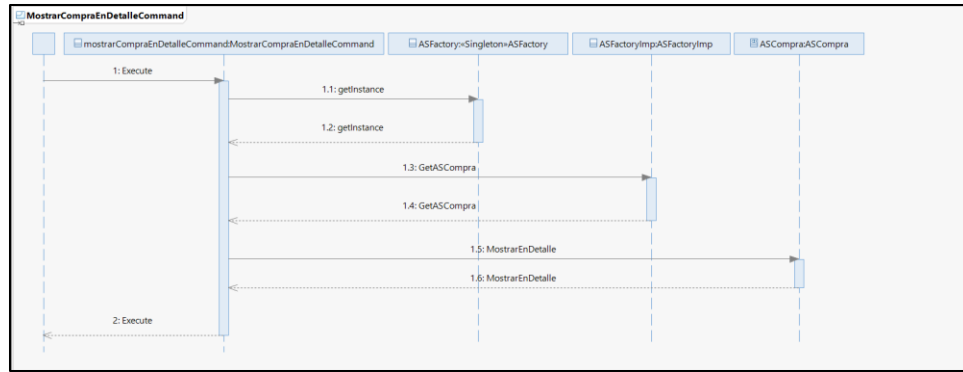


Ilustración 32. Diagrama Secuencia MostrarCompraEnDetalleCommand

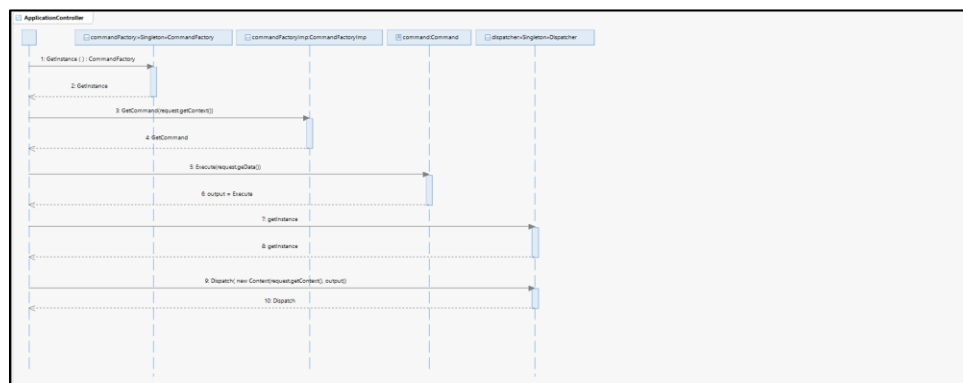


Ilustración 33. Diagrama Secuencia ApplicationController

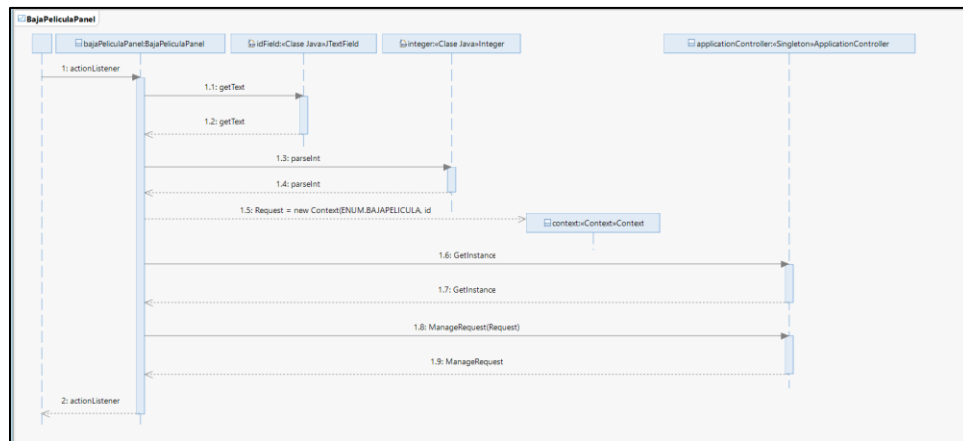


Ilustración 34. Diagrama Secuencia BajaPelículaPanel

5.3. Diagrama de despliegue

El diagrama de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Se adjunta a continuación (Ilustración 35) el diagrama de despliegue correspondiente *Plateau*, mostrando la arquitectura de ejecución de este proyecto.

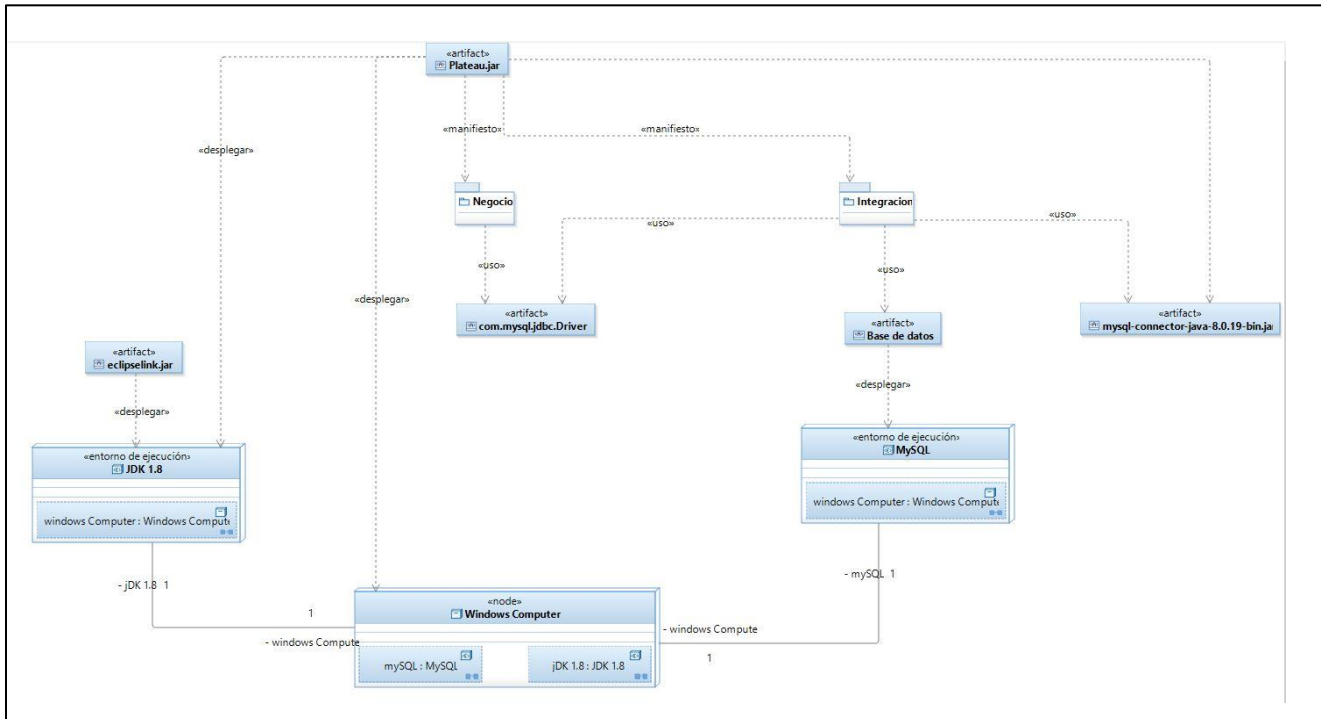


Ilustración 35. Diagrama de despliegue

6. Estructura del código

En primer lugar, es relevante aclarar que el código fue generado a partir del modelo una vez terminado este, y gracias a la herramienta de transformación que proporciona el entorno de desarrollo.

Todo el código de *Plateau* ha sido desarrollado en el repositorio de «cod» y dentro de la carpeta «src». Antes de explicar su contenido, existe también dentro de este repositorio una carpeta «images», donde se almacena el logo de la aplicación, «lib», donde se encuentra el .jar correspondiente al SQL-connector, y un archivo .sql para que cada miembro del equipo pueda generar la base de datos en su SQL Workbench.

Dentro de la carpeta «src», existe una subcarpeta por capa (Integración, Negocio y Presentación), además de una carpeta «test», donde se incluyen los tests unitarios implementados con JUnit, y la carpeta «utilities» con una clase auxiliar necesitada en varios módulos. Dentro de las carpetas correspondientes a cada capa, existen subcarpetas iguales a las explicadas en el apartado del [Modelo](#). Se adjunta a continuación una captura (Ilustración 36) de los distintos paquetes que existen para facilitar al lector la visualización de esta estructura.



Ilustración 36. Estructura código

7. Pruebas

Para garantizar el funcionamiento e implementación de las reglas de negocio de manera correcta, se hicieron pruebas en 3 categorías: pruebas unitarias DAO, pruebas unitarias SA, y comprobaciones manuales.

Las pruebas inicialmente se enfocaron en comprobar el funcionamiento de la capa de integración a través de las pruebas unitarias DAO. Por cada módulo se creó un conjunto de pruebas, cada prueba enfocada en un método distinto del DAO. Las pruebas unitarias DAO aseguran que los datos en la base de datos siempre sean correctos y que no existan duplicados, pero no de que se cumplan las reglas de negocio.

Una vez se hubo garantizado el funcionamiento básico a través de las pruebas unitarias DAO, las pruebas unitarias SA se enfocan en el funcionamiento correcto de la capa de negocio y la implementación de las reglas lógicas. Igual que en las pruebas unitarias DAO, por cada módulo se creó un conjunto de pruebas, cada prueba enfocada en un método distinto del SA. Las pruebas unitarias SA aseguran que no se rompen las reglas de negocio, probando todos los casos posibles.

Además, desde que el código empezó a ser funcional, se han realizado continuas comprobaciones manuales desde la interfaz de usuario para testear los módulos en su totalidad. Estas pruebas han dado lugar a depuraciones exhaustivas pasando por todos métodos de todas las capas hasta localizar los errores que se producían.

Por último, una vez realizadas todas las pruebas mencionadas, se han realizado lo que se ha denominado pruebas de aceptación, en las cuales un integrante del equipo, actuando como cliente, ha testeado la aplicación probando todos los posibles casos que se pueden dar para cada funcionalidad. Así, se ha ido comprobando cada caso de uso comparando con los requisitos establecidos en la SRS, apuntando qué casos fallan para arreglarlos posteriormente, y qué detalles de usabilidad podrían mejorarse. Toda esta información se incluye a continuación. Además, se han apuntado observaciones que se consideran relevantes para la presentación de la aplicación.

- ✓ AltaCliente debería fallar si no tiene tipo seleccionado
- ✓ ModificarCliente se bloquea si se introduce un DNI repetido
- ✓ La reactivación de una productora (el método Update en DAO) debe devolver el ID, no un entero cualquiera
- ✓ AltaSala debería fallar si no tiene tipo seleccionado
- ✓ La reactivación de una sala (el método Update en DAO) debe devolver el ID, no un entero cualquiera
- ✓ Cuando se reactiva una sala su atributo activo no se está actualizando a true
- ✓ Al pulsar “Buscar” en ModificarSala no aparecen los campos correspondientes a los atributos especializados
- ✓ Al pulsar “Buscar” en ModificarSala el ID desaparece, lo que genera un error al pulsar “Modificar”
- ✓ MostrarSala no rellena el campo “Tipo”
- ✓ En BajaCliente no se valida que el ID introducido sea positivo
- ✓ En MostrarCliente no se valida que el ID introducido sea positivo

- ✓ El método clear() no funciona correctamente en ModificarPelicula
- ✓ El mensaje de error al ListarPeliculasPorProductora dice “Pase” en lugar de “Productora”
- ✓ En AltaProductora no se valida que el CIF, el nombre y la dirección sean no vacíos
- ✓ En BajaProductora no se valida que el ID sea positivo
- ✓ El método clear() no funciona correctamente cuando hay un error en ListarPeliculasPorProductora
- ✓ Error ortográfico en el mensaje de PeliculaErrorHandler cuando la película está inactiva
- ✓ En ModificarProductora no se valida que el ID sea no vacío, y que sea un entero positivo
- ✓ En ModificarProductora cuando se repite el CIF o el nombre se envía un mensaje correspondiente a otro tipo de error
- ✓ En ModificarProductora no se comprueba que el CIF, el nombre y la dirección sean no vacíos
- ✓ ModificarProductora se bloquea si se introduce un teléfono vacío o una cadena de caracteres, y si se introduce un entero no comprueba que no sea negativo
- ✓ En MostrarProductora no se comprueba que el ID sea positivo
- ✓ En MostrarProductora el campo “Activo” se rellena con true/false en lugar de con SI/NO
- ✓ El método clear() no funciona correctamente en CalcularProductoraMasIngresos
- ✓ El método clear() no funciona correctamente si se produce un error en AltaSala
- ✓ El método clear() no funciona correctamente en ModificarSala
- ✓ ModificarPelicula se bloquea si se introduce un ID que no existe
- ✓ La reactivación de un pase (el método Update en DAO) debe devolver el ID, no un entero cualquiera
- ✓ Cuando se reactiva un pase su atributo activo no se está actualizando a true
- ✓ Los casos de ListarEntidad1PorEntidad2 se bloquean si se introduce una Entidad2 sin Entidades1
- ✓ El método clear() no funciona correctamente en ListarPasesPorCompra
- ✓ Los casos de ListarEntidad1PorEntidad2 no deben mostrar el atributo activo, dado que siempre es true por los requisitos
- ✓ En ModificarPase no se valida que el ID sea positivo
- ✓ Los mensajes de error en AbrirCompra y TotalComprasPorPelicula no están creados en relación al panel padre
- ✓ CerrarCompra no aplica el descuento si el cliente es VIP
- ✓ MostrarCompra muestra la fecha con hora y es irrelevante
- ✓ MostrarCompraEnDetalle no obtiene el cliente correcto
- ✓ En MostrarCompraEnDetalle al mostrar una compra desaparece el ID
- ✓ CerrarCompra se bloquea si la compra incluye más de un pase
- ✓ En TotalComprasPorPelicula el texto que se muestra no es legible
- ✓ En ModificarPase, si se modifica la sala, no se comprueba que no se haya vendido ninguna entrada de ese pase (condición especificada en la SRS)
- ✓ En los casos de ListarEntidad1PorEntidad2 al pulsar el botón desaparece el ID
- ✓ Los casos CalcularProductoraConMasIngresos y CalcularTotalComprasPorPelicula no funcionan correctamente

Al ser un desarrollo continuo, las pruebas unitarias y manuales permitieron la comprobación del seguido cumplimiento de los requisitos funcionales tras hacer cualquier modificación y parcheo del código.

8. Recursos

El principal recurso utilizado para llevar a cabo el proyecto ha sido el entorno de desarrollo *IBM Rational Software Architect Designer*. Esta herramienta nos ha permitido tanto generar y almacenar código fuente, como elaborar los diagramas de clase, secuencia y despliegue. Se trata de un entorno de modelado y desarrollo que utiliza el lenguaje de modelado unificado (UML) para diseñar arquitectura para aplicaciones y servicios web C++ y Java EE (JEE). *Rational Software Architect* se basa en el marco de software de código abierto Eclipse e incluye capacidades centradas en el análisis de código arquitectónico, C++ y el desarrollo basado en modelos (MDD) con el UML para crear aplicaciones y servicios web.

Por otro lado, para la gestión de la base de datos se ha utilizado MySQL, y como entorno de trabajo el MySQL Workbench, uno de sus primeros productos. Este software incluye entre sus funcionalidades el modelado de datos, el desarrollo de SQL y otras herramientas de administración. Esto permite la depuración del código y la realización de todo tipo de pruebas software.

La aplicación ha sido desarrollada en Java, y el equipo de *Plateau* ha hecho uso del sistema de control de versiones de la Facultad de Informática (Universidad Complutense de Madrid) para gestionar tanto la documentación como el propio código, utilizando SVN para la administración del repositorio.

Aparte de estas herramientas, se han utilizado plataformas como *Whatsapp* y *Discord* para la comunicación del equipo, *Microsoft Word Office 365* para la elaboración de la documentación, y la plataforma *One Drive* para almacenar las distintas versiones del Modelo del Dominio y de la SRS, y otros documentos borradores para facilitar el reparto de tareas y la organización del equipo.