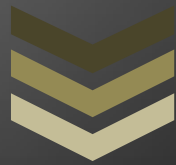


Entrecine



Álvaro García López

Adolfo Fernández Suárez

Herminio García González

Alberto Lorenzo Matilla

Daniel Loureda Arteaga

Entrega Parcial de Arquitectura del Software

Universidad de Oviedo

06/04/2013

Contenido

1	Planteamiento del Problema	3
2	Metodología Usada	3
3	Identificación de los interesados	3
3.1	Clientes del cine	3
3.2	Administrador del sistema	3
3.3	Comerciales del punto de venta	4
3.4	Directivos de la empresa	4
3.5	Equipo de desarrollo	4
4	Identificación inicial de Atributos de Calidad	4
4.1	Disponibilidad.....	4
4.2	Modificabilidad.....	4
4.3	Rendimiento	4
4.4	Seguridad	5
4.5	Testabilidad	5
4.6	Usabilidad	5
5	Primer acercamiento a la solución.....	5
6	Lista actualizada de los interesados	6
7	Lista de Atributos de Calidad	7
8	Atributos de Calidad e Interesados.....	7
9	Descripción de negocio de la aplicación	8
9.1	Funcionamiento de la aplicación web.....	8
9.2	Funcionamiento del cliente de escritorio (TPV).....	9
9.3	Funcionamiento del cliente administrador	9
10	Escenarios de Calidad.....	10
11	Vistas	12
11.1	System View.....	12
11.1.1	Presentación Principal.....	12
11.1.2	Catálogo de Elementos	12
11.1.3	Diagrama Contextual	14
11.1.4	Guía de puntos de variación	14
11.1.5	Justificación de las decisiones	14

11.2	TPV App View.....	14
11.2.1	Presentación Principal.....	14
11.2.2	Catálogo de elementos.....	15
11.2.3	Diagrama Contextual.....	17
11.2.4	Guía de puntos de variación.....	17
11.2.5	Justificación de decisiones.....	17
11.3	AdminApp View.....	17
11.3.1	Presentación Principal.....	17
11.3.2	Catálogo de Elementos.....	17
11.3.3	Diagrama Contextual.....	22
11.3.4	Guía de Puntos de Variación.....	22
11.3.5	Justificación de las decisiones.....	22
11.4	API View.....	22
11.4.1	Presentación Principal.....	23
11.4.2	Catálogo de Elementos.....	23
11.4.3	Diagrama Contextual.....	24
11.4.4	Guía de Puntos de Variación.....	24
11.4.5	Justificación de las decisiones.....	25
11.5	Sequence View.....	25
11.5.1	Presentación principal.....	25
11.5.2	Catálogo de elementos.....	25
11.5.3	Diagrama contextual.....	26
11.5.4	Guía de puntos de variación.....	26
11.5.5	Justificación de las decisiones.....	26
11.6	Deployment View.....	27
11.6.1	Presentación principal.....	27
11.6.2	Catálogo de elementos.....	27
11.6.3	Diagrama contextual.....	28
11.6.4	Guía de puntos de variación.....	28
11.6.5	Justificación de las decisiones.....	28
12	Documentación Adicional.....	29
13	Índice de figuras.....	29

1 Planteamiento del Problema

La empresa Entrecine necesita ofrecer un sistema de reserva de entradas de cine a través de la web, además de la compra directa en el propio cine (con un TPV). Adicionalmente también necesita de un sistema de administración que permita configurar todos los aspectos relativos al cine, las películas, las entradas, etc.

Los datos que el usuario introduzca en la web podrán ser guardados para que en sucesivas compras no tenga que volver a introducirlos, a excepción de los datos de la tarjeta de crédito, que deberán ser introducidos en todas las compras. El pago de las entradas se realizará a través de una pasarela de pago externa a los servicios que ofrece la empresa, luego el sistema que se cree se deberá integrar con dicho software.

En cuanto al TPV deberá trabajar con los mismos datos que la aplicación web y funcionará igual que cualquier otro TPV, permitiéndose el pago con tarjeta de crédito y en metálico.

2 Metodología Usada

La metodología empleada en este proyecto seguirá el método ADD (Attribute Driven Design) (Bass y otros, 2003).

3 Identificación de los interesados

- Clientes del cine
- Administrador del sistema
- Comerciales del punto de venta
- Directivos de la empresa
- Equipo de desarrollo

3.1 Clientes del cine

Son las personas que realizan la compra de entradas tanto presencialmente como vía telemática.

Sus principales objetivos son:

- Poder visualizar de forma sencilla y rápida toda la información que necesite (películas en cartelera, horarios, localidades libres...).
- Realizar la compra de entradas de manera segura y en cualquier momento.

3.2 Administrador del sistema

El administrador es la persona que configura todo el sistema, tanto las sesiones, como las películas o el precio de las entradas.

Sus principales objetivos son:

- Poder introducir los parámetros de forma sencilla.
- Que la actualización se realice inmediatamente.

3.3 Comerciales del punto de venta

Los taquilleros de toda la vida. Principales usuarios de la aplicación en su versión de escritorio.

Sus principales objetivos son:

- Gestionar la venta de forma rápida y eficaz.
- Que el sistema de escritorio que usa no provoque ningún conflicto con la versión web.

3.4 Directivos de la empresa

Se trata de las personas que van a financiar nuestro proyecto.

Sus principales objetivos son:

- Proyecto de rápido desarrollo y bajo coste, de manera que pueda solucionar su necesidad cuanto antes al menor precio posible.

3.5 Equipo de desarrollo

Grupo de 5 personas estudiantes de Ingeniería de Software.

Sus principales objetivos son:

- Uso de tecnología acorde a los conocimientos actuales de los integrantes para realizar el proyecto de manera rápida y eficaz.

4 Identificación inicial de Atributos de Calidad

4.1 Disponibilidad

- El sistema debe ser un 24/7 con backups del DB server (servidor espejo en caso de caída del servicio). Por ejemplo, backup del sistema todas las noches a las 4 de la madrugada, ya que es un periodo de completa inactividad casi siempre. Establecer en la interfaz web (en la aplicación de escritorio no hará falta ya que el cine estará cerrado) que el sistema está en mantenimiento durante esas horas.

4.2 Modificabilidad

- La estructura del cine debe de ser tomada a partir de una base de datos. Nunca fija en el programa, ya que la estructura del cine puede cambiar.
- Aplicación parametrizable, hay que ser flexibles a cambios e imprevistos que puedan aparecer.
- Proyecto sensible a cambios, pueden llegar nuevos requisitos durante el desarrollo.

4.3 Rendimiento

- El sistema debe estar preparado para afrontar cargas de trabajo medias de 1 petición por cada 10 minutos y con picos de hasta 10 peticiones por minuto.

4.4 Seguridad

- El número de identificador de la entrada debe ser generado por el sistema de manera pseudoaleatoria y ser único.
- Cumplimiento de la LOPD.

4.5 Testabilidad

- Cada vez que se tenga una interfaz con un subsistema por debajo, crear unos test que permitan ver que ante cualquier entrada que el sistema responde como debería.
- Crear además unos scripts que permitan probar el sistema una vez finalizado y que cada vez que se lance se ponga a reservar una entrada de cine. Lanzando, por lo tanto, 10 instancias del Script en un minuto se podrá comprobar el rendimiento del sistema.

4.6 Usabilidad

- La página web ha de ser usable ya que de no ser así se podrían perder potenciales clientes.
- La aplicación de escritorio debe tener una interfaz clara para su uso rápido y eficiente.
- Pasos cortos pero bien definidos y sin posibilidad de pérdida.
- El proceso de venta ha de ser transaccional, o bien se realiza la venta o no se realiza ningún cambio.
- Los datos entre el sistema físico de venta y virtual deben de ser consistentes.
- Base de datos creciente con el tiempo, ha de ser mantenible y consistente.

5 Primer acercamiento a la solución

Una vez estudiados los requisitos iniciales, se opta por 3 clientes, dos de escritorio y uno web, que utilizan una base de datos relacional centralizada.

El cliente web servirá para que los clientes efectúen la compra de entradas por Internet, por lo que deberá de contar con una pasarela de pago.

El primer cliente de escritorio servirá para que los usuarios que deseen realizar la compra de entradas de manera presencial puedan hacerlo. Debido a que el pago en este caso puede ser tanto en efectivo como con tarjeta se debe de contar también con una pasarela de pago.

Por último, se debe proporcionar un cliente de escritorio para que el administrador del cine pueda gestionar las salas, añadir nuevas películas, gestionar el personal o modificar horarios.

Riesgos relacionados con la solución:

1. **Garantía de reservas concurrentes.** Este riesgo es especialmente importante cuando hay varios usuarios intentando hacer reservas de entradas en el sistema ya que se pueden producir fenómenos como reservas fantasma o Dead Locks (espera circular).

2. **Garantía de persistencia ante fenómenos adversos.** Debemos garantizar la persistencia de los datos y la posible recuperación ante ataque o averías del servidor.
3. **Garantía de Seguridad en pagos.** Es muy importante que los datos bancarios de los clientes no se vean comprometidos tanto si están haciendo un pago como si estos están almacenados.
4. **Protección ante uso no autorizado** de los clientes de escritorio. Los clientes de escritorio deben garantizar que es un empleado legítimo el que está manipulando el sistema.

Para resolver los riesgos anteriores se realizan una serie de trabajos que se describen a continuación.

1. **Garantía de reservas concurrentes.** El sistema será de tipo transaccional y estará convenientemente “testado”(probado) en un entorno concurrente para evitar los Dead Locks. Además, estará protegido contra lecturas fantasma en la base de datos.
2. **Garantía de persistencia ante fenómenos adversos.** Para evitar cualquier tipo de pérdida de datos se contará con un sistema automatizado de copias de seguridad (BackUps).
3. **Garantía de Seguridad en pagos.** En el caso de esta práctica se realizará una simulación de la pasarela de pago. Sin embargo, para el sistema real contaremos con un sistema seguro contratado con el banco que nos garantizará la fiabilidad de los pagos.
4. **Protección ante uso no autorizado de los clientes de escritorio.** Los clientes de escritorio deberán contar con un sistema de logueo contra la base de datos para evitar que alguien no autorizado lo use.

6 Lista actualizada de los interesados

Código	Stakeholder	Intereses
ST-01	Cientes del cine	Visualizar de forma rápida y sencilla películas, horarios, localidades libres. Realizar compra de entradas en cualquier momento y de forma segura.
ST-02	Administrador del sistema	Altas, bajas y modificaciones de películas, sesiones, usuarios. Actualización en base de datos inmediata.
ST-03	Comerciales del punto de venta	Consistencia entre aplicación web y de escritorio. Interfaz de aplicación clara, concisa y manejable.
ST-04	Directivos de la empresa	Coste de desarrollo bajo. Tiempo de desarrollo corto
ST-05	Equipo de desarrollo	Uso de tecnología acorde a los conocimientos del equipo, es decir, de fácil uso y rápido desarrollo.

7 Lista de Atributos de Calidad

En función de los trabajos anteriores se identifica la siguiente lista de atributos de calidad, en la que aparecen los nuevos atributos obtenidos tras analizar los riesgos del sistema.

Código	Descripción	Tipo de atributo
AT001	Sistema 24/7	Disponibilidad
AT002	Backup	Disponibilidad
AT003	Aplicación parametrizable sensible a cambios	Modificabilidad
AT004	Debe soportar picos de varios usuarios comprando concurrentemente	Rendimiento
AT005	Identificador de entrada único pseudoaleatorio	Seguridad
AT006	Test para cada caso con scripts para pruebas de concurrencia	Testabilidad
AT007	Página web y aplicación de escritorio usables teniendo en cuenta a la gente con discapacidades o dificultades	Usabilidad
AT008	Pagos seguros a través de pasarela de pago protegida	Seguridad
AT009	Login para acceso a las aplicaciones	Seguridad
AT010	Tiempo de desarrollo muy corto	Time-to-Market
AT011	Coste reducido	Coste
AT012	Cumplimiento de LOPD	Seguridad

8 Atributos de Calidad e Interesados

Atributos vs Interesados	ST-01	ST-02	ST-03	ST-04	ST-05
AT001	x			x	x
AT002	x	x		x	x
AT003					x
AT004	x			x	x
AT005	x				x
AT006					x
AT007	x	x	x		x
AT008	x			x	x
AT009	x	x	x		x
AT010				x	x
AT011				x	x
AT012	x			x	x

9 Descripción de negocio de la aplicación

Para describir la solución se utilizarán diagramas contextuales y texto.

El modelo de negocio de la aplicación estará constituido por un portal web donde se podrán comprar entradas así como un software para un terminal de punto de venta donde se podrán adquirir de manera presencial. Tanto la base de datos como la página web estarán alojados en un mismo servidor central ya que, a pesar de que puede haber varios usuarios a la vez comprando entradas, nunca se alcanzarán picos de consultas a base de datos tan elevados como para necesitar de un servidor adicional exclusivo para este servicio y se dispone de un presupuesto limitado.

El sistema consta de 3 aplicaciones: La aplicación Web, la aplicación de escritorio para el TPV (Terminal de Punto de Venta) y por último el cliente para el administrador del cine. Dado que los 3 clientes utilizan la misma base de datos, se creará una librería que permita su acceso y donde sólo se necesite hacer llamadas a la misma, en lugar de tener código triplicado para cada aplicación.

En caso de que una película exceda el tiempo establecido para la emisión se pondrá una emisión de otra película de duración más corta en la misma sala donde debería emitirse la película de larga duración.

9.1 Funcionamiento de la aplicación web

La aplicación web está alojada en un servidor común junto con la base de datos. Dicha aplicación obtiene los datos de las películas disponibles de la base de datos y se las presenta al usuario.

Cuando el usuario haya elegido la película que desea ver, el sistema le mostrará los días y sesiones en los que la película estará disponible. Una vez haya decidido la sesión a la que desea acudir se procederá a mostrar las butacas disponibles en dicha sesión (si hay más de una sala habilitada se ofrece la posibilidad de cambiar de una sala a otra). Tras seleccionar la butaca se inicia el proceso de reserva:

- En primer lugar el sistema bloqueará la butaca elegida para evitar que dos usuarios intenten reservar la misma simultáneamente. Adicionalmente se establecerá un timeout de forma que el cliente no pueda estar indefinidamente en la zona de reserva con la butaca bloqueada.
- Tras realizar el bloqueo se procede a tomar los datos del cliente (nombre, apellidos, email...).
- Cuando ya disponemos de los datos del usuario, le mandamos un ticket provisional para retirar la entrada y le advertimos de que debe completar el pago para que dicho ticket sea válido.
- El cliente pues realizará el pago a través de la pasarela de pago, y el sistema se quedará a la espera de que dicha pasarela le confirme que el pago se ha realizado. En esta fase el sistema detendrá el timeout que se había establecido en el primer paso e iniciará otro timeout específico para la pasarela de pago para asegurarnos de que no hay problemas durante el pago, si el timeout se excede volvemos al paso anterior, activamos el timeout que se detuvo al entrar en la pasarela de pago y el cliente deberá volver a iniciar el pago de su entrada.

- Cuando la pasarela nos confirme que la entrada ha sido pagada, activamos el ticket en nuestro sistema para dejar constancia de que es válido y se lo notificamos al usuario.
- Si en cualquier momento de la reserva el usuario decide que ya no quiere realizarla, el sistema desbloqueará la butaca para que otro usuario pueda reservarla.

El número máximo de reservas que se pueden hacer simultáneamente en el sistema a través de esta aplicación web es de 5, ya que así el sistema está protegido frente a un posible ataque DoS que intentase reservar todas las butacas del cine pero nunca llegase a pagarlas, por lo que dejaría todas las butacas bloqueadas.

9.2 Funcionamiento del cliente de escritorio (TPV)

El cliente de escritorio estará situado en cada ordenador del personal de venta presencial del cine. Constará de una pantalla de logueo para evitar que personal no autorizado pueda expender entradas, otra pantalla donde se podrá seleccionar la película, sesión, día y butacas. En el caso de que una película se esté mostrando en más de una sala en una misma sesión y día se mostrará la opción de elegir en qué sala se quiere hacer la reserva. En el caso de esta aplicación no habrá limitación ninguna para la venta de entradas a la vez, como ocurre en el cliente web. La interfaz constará de botones grandes para un mejor manejo en caso de que el TPV tenga una interfaz táctil.

Una vez realizado todo el proceso de reserva se procederá al pago a través de la “*pasarela de pago*”, que será proporcionada por el banco (en el caso de esta versión se contará con un simulador), y finalmente se procederá a expedir la entrada final.

9.3 Funcionamiento del cliente administrador

El cliente administrador constará de una pantalla de logueo, especialmente importante en este caso, ya que tendrá poder absoluto para modificar el cine. Una vez superada esta pantalla se verá una pantalla donde se mostrará qué aspectos se quiere modificar entre los siguientes:

- Películas
 - Añadir Película
 - Modificar Película
 - Eliminar Película
- Sala
 - Añadir Sala
 - Modificar Sala
 - Eliminar Sala
- Personal Cine
 - Añadir usuario
 - Modificar Usuario
 - Eliminar Usuario
- Incidencias
 - Bloquear sala por incidencia

10 Escenarios de Calidad

Escenario N°	Fuente de Estímulo	Estímulo	Entorno	Artefacto	Respuesta	Medición de la Respuesta	Atributo de Calidad Afectado
1	Clientes comprando entradas	Venta de entradas	Explotación	Sistema en funcionamiento	Tiempo efectivo de venta	Tiempo efectivo > 95%	AT001
2	Recuperación de datos	Avería o ataque	Explotación	Base de datos	Recuperación de la copia de seguridad realizada	Pérdida de datos < 2%	AT002
3	Nueva forma de venta de entradas	Cambio forma venta de entradas	Explotación	Aplicación Web, Aplicación de Escritorio	Actualización del diseño y vuelta a la venta	Número de operaciones con sistemas antiguos: 0	AT003
4	Nueva distribución del cine	Cambio forma venta de entradas	Explotación	Aplicación Web, Aplicación de Escritorio	Actualización del diseño y vuelta a la venta	Número de operaciones con sistemas antiguos: 0	AT003
5	Incremento del número de usuarios	Mayor número de datos a procesar por Venta de entradas	Explotación	Venta de entradas	Incremento de la capacidad de procesamiento del servidor	Número de caídas del sistema < 5%	AT004
6	Evitar generación de tickets fraudulentos	Evitar que Usuarios no legítimos vendan entradas	Explotación	Aplicación Web	Algoritmo de generación de códigos de entradas pseudoaleatorio	Que no haya un número de usuarios en la sala mayor que el número de entradas	AT005

						expedidas	
7	Bugs	Evitar el paso a explotación con fallos en el sistema	Desarrollo	Aplicación Web, Aplicación de Escritorio, Aplicación de Administración	Pruebas unitarias con éxito, pruebas de concurrencia satisfactorias	Caídas del sistema por bugs < 50%, pruebas exitosas: 100%	AT006
8	Web del cine	Usuarios incapaces de usar la web	Explotación	Página web	Modificación de las partes poco usables y test de usabilidad	Reservas fallidas < 5%	AT007
9	Pasarela de pago	Pago mediante tarjeta de crédito/débito	Explotación	Pasarela de pago	Contratación de pasarela de pago con el banco	Total de números de tarjetas robadas < 1	AT008
10	Control de acceso	Evitar uso no autorizado	Explotación	Aplicación de Escritorio, Aplicación de Administración	Accesos no autorizados	Número de accesos no autorizados: 0	AT009
11	Sistema en desarrollo	Fin del desarrollo	Desarrollo	Sistemas y equipos de desarrollo	Tiempo bajo de desarrollo	Time to market <= 1 mes	AT010
12	Sistema en desarrollo	Fin del desarrollo	Desarrollo	Sistemas y equipos de desarrollo	Coste bajo de desarrollo	Coste de desarrollo: 0€	AT011
13	Base de datos	Uso de datos	Explotación	Base de datos y aplicaciones	Cumplimiento de la ley vigente	Denuncias de usuarios = 0	AT012

11 Vistas

En los próximos párrafos se describirán algunas de las vistas identificadas y se documentarán.

VISTA	REFERENCIA	ESCENARIOS
System	11.1	
TPV App	11.2	1, 6, 7, 10, 11
AdminApp	11.3	4, 7, 10
API	11.4	1, 3, 6, 7, 9, 10
Sequence	11.5	1, 3, 6, 7, 9
Deployment	11.6	2, 5

11.1 System View

La vista del sistema describe los dos subsistemas principales: Las aplicaciones y el sistema central, así como sus interfaces.

11.1.1 Presentación Principal

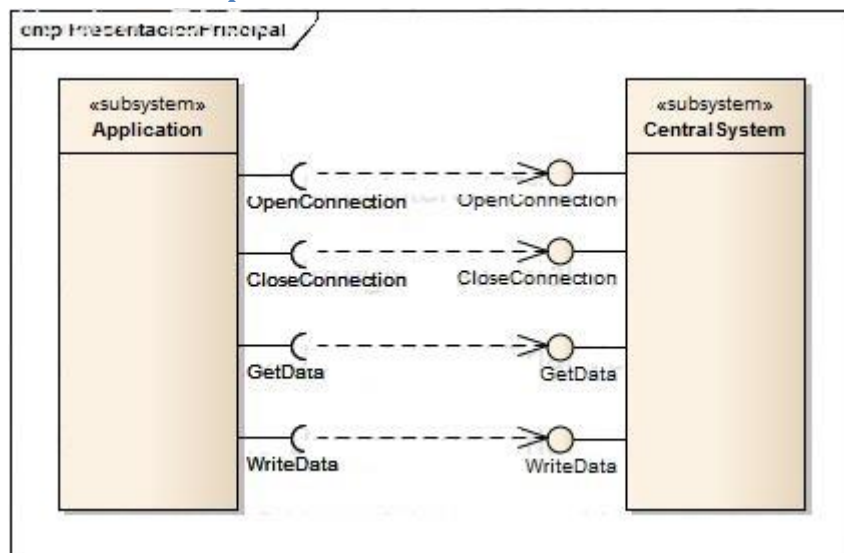


Figura 1. System View

11.1.2 Catálogo de Elementos

11.1.2.1 Elementos

Elemento	Propiedades
Application	Subsistema desde donde se venden entradas o se administra el sistema
Central System	Subsistema residente en el servidor central que posee los datos

11.1.2.2 Relaciones

Ambos subsistemas se relacionan por medio de un .jar que permite que cada una de las distintas aplicaciones puedan interactuar con el servidor central.

11.1.2.3 Interfaces / Puertos

11.1.2.3.1 Application

Interface	Tipo	Tecnología	Propiedades
OpenConnection	Requerida	.jar con librería jdbc para conexión a BBDD MySQL	Abre la conexión entre la base de datos MySQL residente en el servidor central y la aplicación
CloseConnection	Requerida	.jar con librería jdbc para conexión a BBDD MySQL	Cierra la conexión entre la base de datos MySQL residente en el servidor central y la aplicación
GetData	Requerida	.jar con librería jdbc para conexión a BBDD MySQL	Recibe datos almacenados en la base de datos MySQL residente en el servidor central y se los pasa a la aplicación
WriteData	Requerida	.jar con librería jdbc para conexión a BBDD MySQL	La aplicación escribe datos en la base de datos MySQL residente en el servidor central

11.1.2.3.2 Central System

Interface	Tipo	Tecnología	Propiedades
OpenConnection	Proveída	.jar con librería jdbc para conexión a BBDD MySQL	Abre la conexión entre la base de datos MySQL residente en el servidor central y la aplicación
CloseConnection	Proveída	.jar con librería jdbc para conexión a BBDD MySQL	Cierra la conexión entre la base de datos MySQL residente en el servidor central y la aplicación
GetData	Proveída	.jar con librería jdbc para conexión a BBDD MySQL	Envía datos almacenados en la base de datos MySQL residente en el servidor central y se los pasa a la aplicación
WriteData	Proveída	.jar con librería jdbc para conexión a BBDD MySQL	Recibe datos de la aplicación y los guarda en la base de datos MySQL residente en el servidor central

11.1.2.4 Comportamiento

11.1.2.4.1 Application

Es un sistema que constará de tres aplicaciones finales (WebApp, TPV y Cliente de escritorio de administración) donde cada una de ellas se conectará por medio de un .jar común al servidor central (que es el que contiene la base de datos MySQL con toda la información) y en función del tipo de aplicación requerirá unos datos u otros.

La WebApp se realizará con JSF y ambas aplicaciones de escritorio se realizarán en Java con una interfaz Swing para que sean más sencillas de manejar.

11.1.2.4.2 Central System

El sistema central constará de una Base de Datos MySQL común a las 3 aplicaciones. Esta base de datos registrará toda la información necesaria así como datos de los usuarios de la web o de las aplicaciones de escritorio. Además, llevará a cabo un control de incidencias en caso de que no se pueda usar una sala.

11.1.3 Diagrama Contextual

N/A

11.1.4 Guía de puntos de variación

N/A

11.1.5 Justificación de las decisiones

Las decisiones que han llevado a cabo este diseño son:

Escenario	Atributos de Calidad	Justificación
3	AT003	Tener una sola librería con todas las conexiones necesarias a la BBDD para todas las aplicaciones permite que, en caso de necesitar hacer un cambio en la BBDD, sólo haya que adjuntar la nueva librería a la aplicación
8	AT007	Las aplicaciones con interfaz gráfica permite que sea mucho más sencillo su manejo, especialmente en el caso de personas con discapacidades
11 y 12	AT010	Una vez realizada la librería, no hay que implementar código redundante en cada una de las aplicaciones, por lo que se ahorra tiempo de desarrollo y se reducen costes

11.2 TPV App View

La vista de TPV App muestra el primer nivel de descripción de los componentes de la aplicación de escritorio del terminal en el punto de venta.

11.2.1 Presentación Principal

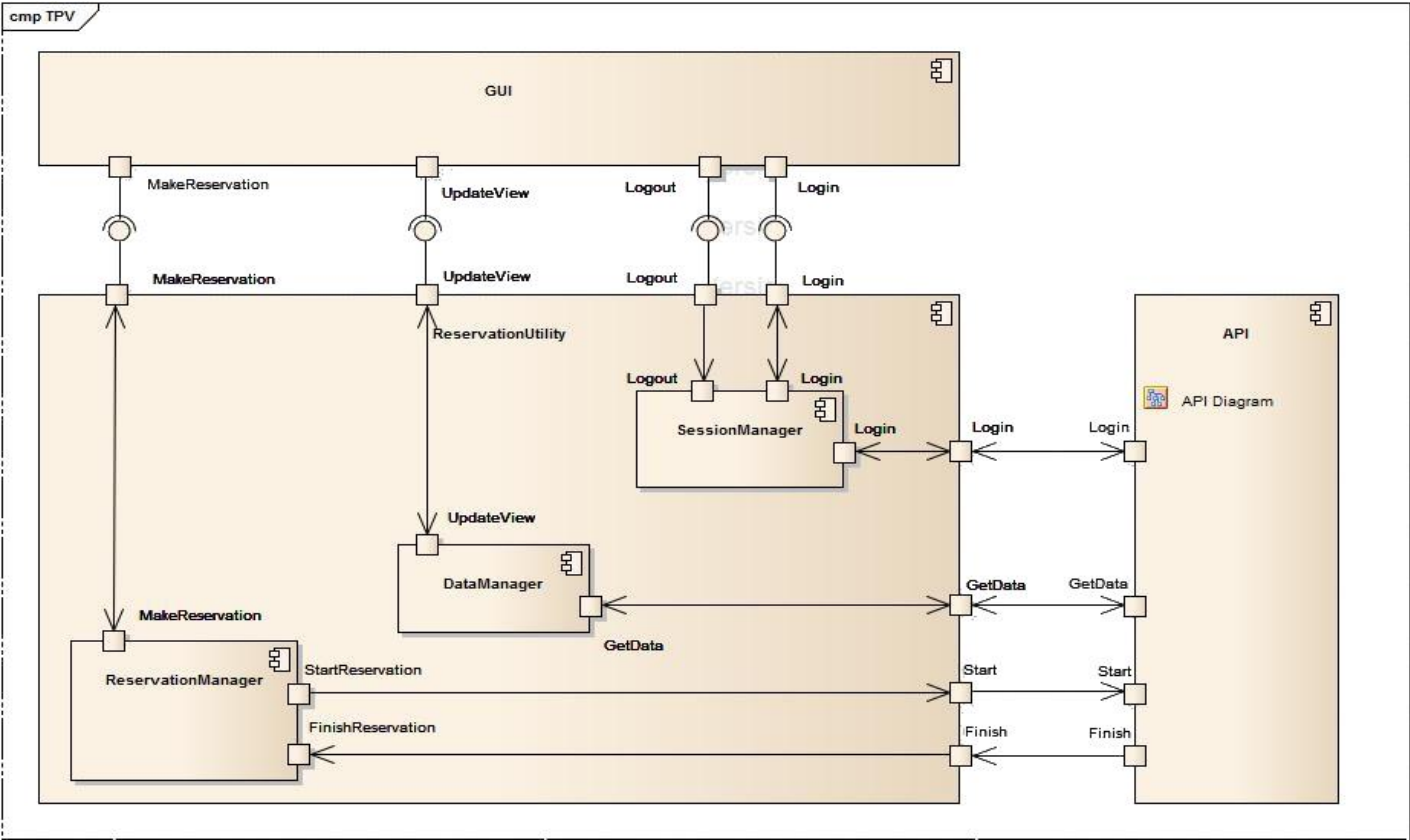


Figura 2. TPVApp View

11.2.2 Catálogo de elementos

11.2.2.1 Elementos

Elemento	Propiedades
GUI	Interfaz gráfica de la aplicación.
ReservationUtility	Contenedor de toda la lógica de la aplicación.
SessionManager	Componente interno de ReservationUtility. Gestiona la sesión del empleado.
DataManager	Componente interno de ReservationUtility. Se encarga de actualizar la interfaz gráfica con los datos de la BD.
ReservationManager	Componente interno de ReservationUtility. Se encarga de gestionar las reservas y actualizar la BD.

11.2.2.2 Relaciones

El usuario interactuará con la aplicación a través de la interfaz gráfica, introducirá sus datos de identificación y SessionManager a través de la API comprobará que sean correctos y le da acceso al sistema.

El usuario navegará por la aplicación y DataManager se encargará de proporcionar los datos necesarios para mostrar en pantalla, finalmente ReservationManager se encargará de realizar la reserva.

11.2.2.3 Interfaces / Puertos

11.2.2.3.1 GUI

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call	Llama a ReservationUtility para iniciar una sesión.
Logout	Puerto (Out)	Method Call	Llama a SessionManager para terminar una sesión.
UpdateView	Puerto (In-Out)	Method Call	Controla la navegación entre pantallas y llama a ReservationUtility para obtener los datos que se muestran.
MakeReservation	Puerto (In-Out)	Method Call	Llama a ReservationUtility para realizar una reserva.

11.2.2.3.2 Reservation Utility

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call	Método fachada que redirige la petición a la API pasando por el SessionManager para hacer los cambios necesarios en la interfaz.
Logout	Puerto (Out)	Method Call	Método fachada que redirige la petición a SessionManager.
UpdateView	Puerto (In-Out)	Method Call	Método fachada que redirige la petición a DataManager.
MakeReservation	Puerto (In-Out)	Method Call	Método fachada que redirige la petición a ReservationManager.
GetData	Puerto (In-Out)	Method Call	Recibe datos almacenados en la base de datos MySQL residente en el servidor central y se los pasa a la aplicación.
Start	Puerto (Out)	Method Call	Método para iniciar la reserva de las entradas que el usuario quiera a través de la API.

Finish	Puerto (In)	Method Call	Método que manda a la aplicación el resultado de la reserva.
--------	-------------	-------------	--

11.2.2.3.3 SessionManager

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call	Llama al método Login de la API para realizar el inicio de sesión.
Logout	Puerto (Out)	Method Call	Cierra la sesión.

11.2.2.3.4 DataManager

Interface	Tipo	Tecnología	Propiedades
UpdateView	Puerto (In-Out)	Method Call	Devuelve los datos solicitados por la vista.
GetData	Puerto (In-Out)	Method Call	Llama al método de la API para acceder a los datos.

11.2.2.3.5 ReservationManager

Interface	Tipo	Tecnología	Propiedades
MakeReservation	Puerto (In-Out)	Method Call	Con los datos proporcionados se encarga de realizar la reserva e informa si hubo éxito.
Start	Puerto (Out)	Method Call	Llama a la API para comenzar el proceso de reserva de entradas.
Finish	Puerto (In)	Method Call	Recibe la respuesta del proceso de reserva, es decir, si ha habido éxito o no.

11.2.2.4 Comportamiento

11.2.2.4.1 GUI

Es la interfaz de usuario, constará de botones grandes dado su uso en pantallas táctiles y su diseño será sencillo para un fácil aprendizaje y alta eficiencia de uso. Será desde este componente desde donde se llamará a los demás y se realizará la navegación entre las distintas pantallas.

11.2.2.4.2 ReservationUtility

Componente que aúna todo la lógica de la aplicación, la interfaz de usuario solo interactuará con este componente que consta de una interfaz a modo de fachada para facilitar el desacoplamiento de los demás componentes.

11.2.2.4.3 SessionManager

El componente recibirá una solicitud para realizar el login, con los datos recibidos realiza una llamada a la API para comprobar que estos sean correctos y devuelve el resultado de esta comprobación. Además inicializará los datos necesarios para el mantenimiento de la sesión.

11.2.2.4.4 DataManager

El componente recibirá una solicitud para obtener unos datos determinados, una vez que la procese, realizará una llamada a la API para obtenerlos de la base de datos y los devolverá a la interfaz para que los muestre.

11.2.2.4.5 ReservationManager

El componente recibirá una solicitud para realizar una reserva, comprobará que los datos recibidos son correctos, realizará una llamada a la API para comenzar la reserva y finalmente recibirá la respuesta de ésta.

11.2.3 Diagrama Contextual

N/A

11.2.4 Guía de puntos de variación

N/A

11.2.5 Justificación de decisiones

N/A

11.3 AdminApp View

Esta vista describe el principal funcionamiento del cliente de escritorio para el administrador del cine.

11.3.1 Presentación Principal

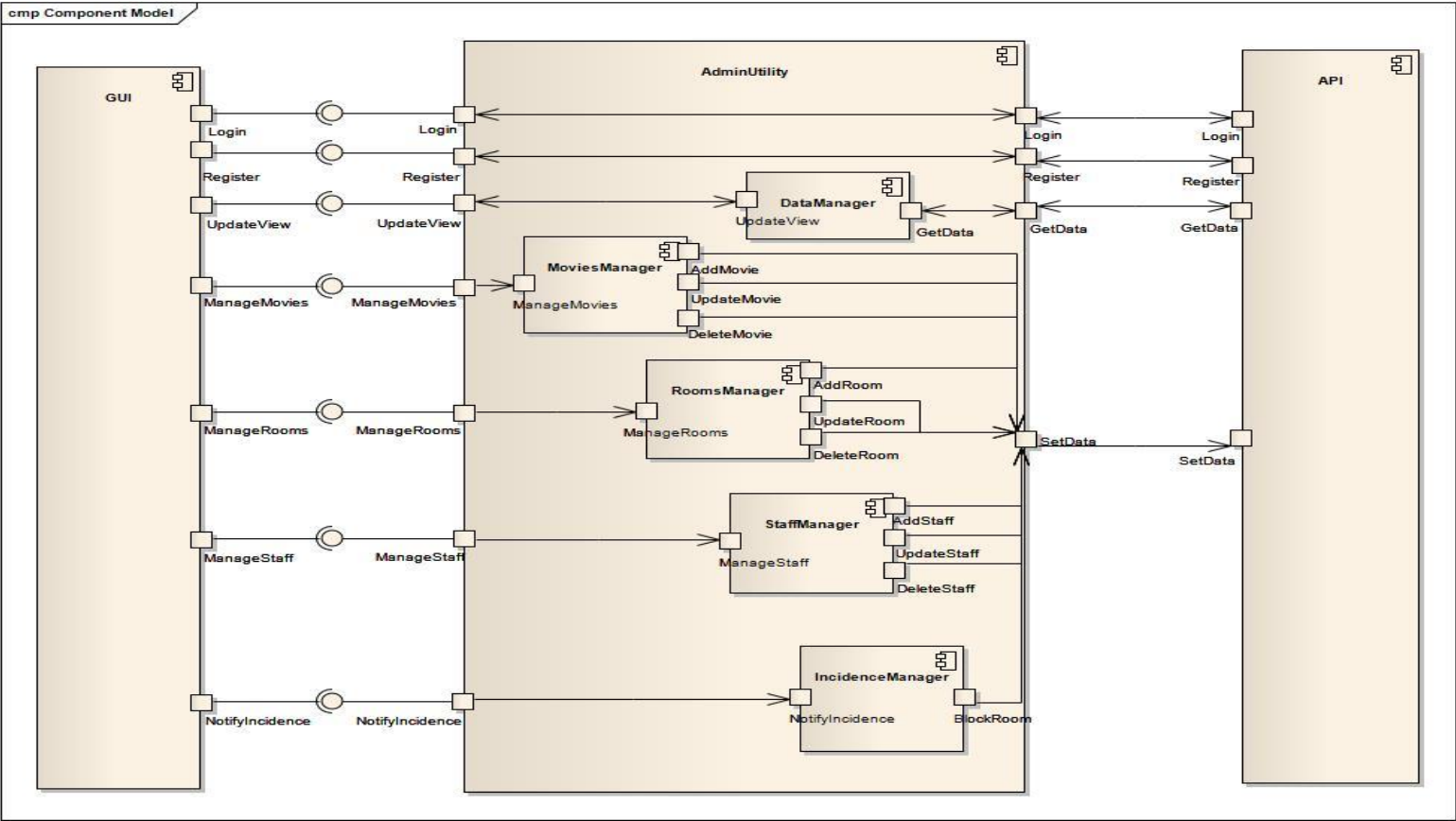


Figura 3. AdminApp View

11.3.2.1 Elementos

Elemento	Propiedades
GUI	Interfaz gráfica de la aplicación.
AminUtility	Contenedor de toda la lógica de la

	aplicación.
MoviesManager	Componente interno de AdminUtility. Gestiona el CRUD de películas.
DataManager	Componente interno de AdminUtility. Se encarga de actualizar la interfaz gráfica con los datos de la BBDD.
RoomsManager	Componente interno de AdminUtility. Gestiona el CRUD de salas.
StaffManager	Componente interno de AdminUtility. Gestiona el CRUD de personal.
IncidenceManager	Componente interno de AdminUtility. Gestiona las incidencias que se producen en las salas.

11.3.2.2 Relaciones

El usuario (en este caso el administrador) interactúa con la GUI (interfaz gráfica). Deberá autenticarse (Login) contra la base de datos, de lo que se encargará la API. Además, se dará opción a registrar otro administrador.

El usuario también podrá gestionar todo lo respectivo a películas, personal, salas o incidencias, a través de los respectivos managers que se proporcionan (MoviesManager, StaffManager, RoomsManagers e IncidenceManager, respectivamente). Dichos managers se comunicarán con la API, que se encargará de acceder a la BBDD.

11.3.2.3 Interfaces / Puertos

11.3.2.3.1 GUI

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call	Llama a AdminUtility para iniciar una sesión.
Register	Puerto (In-Out)	Method Call	Llama a AdminUtility para registrar un administrador.
UpdateView	Puerto (In-Out)	Method Call	Controla la navegación entre pantallas y llama a AdminUtility para obtener los datos que se muestran.
ManageMovies	Puerto (Out)	Method Call	Llama a AdminUtility para añadir, modificar o borrar películas.
ManageStaff	Puerto (Out)	Method Call	Llama a AdminUtility para

			añadir, modificar o borrar personal.
ManageRooms	Puerto (Out)	Method Call	Llama a AdminUtility para añadir, modificar o borrar salas del cine.
NotifyIncidence	Puerto (Out)	Method Call	Llama a AdminUtility para notificar una incidencia de una sala.

11.3.2.3.2 AdminUtility

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call	Redirige la petición de Login a la API.
Register	Puerto (In-Out)	Method Call	Redirige la petición de Register a la API.
UpdateView	Puerto (In-Out)	Method Call	Redirige la petición de UpdateView al DataManager.
ManageMovies	Puerto (In)	Method Call	Redirige la petición de ManageMovies al MoviesManager.
ManageStaff	Puerto (In)	Method Call	Redirige la petición de ManageStaff al StaffManager.
ManageRooms	Puerto (In)	Method Call	Redirige la petición de ManageRooms al RoomsManager.
NotifyIncidence	Puerto (In)	Method Call	Redirige la petición de NotifyIncidence al IncidenceManager.
GetData	Puerto (In-Out)	Method Call	Recibe datos almacenados en la base de datos MySQL residente en el servidor central.
SetData	Puerto (Out)	Method Call	La aplicación escribe datos en la base de datos MySQL residente en el servidor central.

11.3.2.3.3 DataManager

Interface	Tipo	Tecnología	Propiedades
UpdateView	Puerto (In-Out)	Method Call	Devuelve los datos solicitados por la vista.

GetData	Puerto (In-Out)	Method Call	Llama al método de la API para acceder a los datos.
---------	-----------------	-------------	---

11.3.2.3.4 MoviesManager

Interface	Tipo	Tecnología	Propiedades
ManageMovies	Puerto (In)	Method Call	Recibe los datos necesarios para poder realizar la petición a la API.
AddMovie	Puerto (Out)	Method Call	Llama al método de la API para introducir una película en la BBDD.
UpdateMovie	Puerto (Out)	Method Call	Llama al método de la API para modificar una película de la BBDD.
DeleteMovie	Puerto (Out)	Method Call	Llama al método de la API para eliminar una película de la BBDD.

11.3.2.3.5 StaffManager

Interface	Tipo	Tecnología	Propiedades
ManageStaff	Puerto (In)	Method Call	Recibe los datos necesarios para poder realizar la petición a la API.
AddStaff	Puerto (Out)	Method Call	Llama al método de la API para introducir un miembro de personal en la BBDD.
UpdateStaff	Puerto (Out)	Method Call	Llama al método de la API para modificar un miembro de personal de la BBDD.
DeleteStaff	Puerto (Out)	Method Call	Llama al método de la API para eliminar un miembro de personal de la BBDD.

11.3.2.3.6 RoomsManager

Interface	Tipo	Tecnología	Propiedades
ManageStaff	Puerto (In)	Method Call	Recibe los datos necesarios para poder realizar la petición a la API.
AddRooms	Puerto (Out)	Method Call	Llama al método de la API para introducir una sala en la BBDD.
UpdateRooms	Puerto (Out)	Method Call	Llama al método de la API para modificar una sala de la BBDD.
DeleteRooms	Puerto (Out)	Method Call	Llama al método de la API para eliminar una sala de la BBDD.

11.3.2.3.7 IncidenceManager

Interface	Tipo	Tecnología	Propiedades
NotifyIncidence	Puerto (In)	Method Call	Recibe los datos necesarios para poder realizar la petición a la API.
BlockRoom	Puerto (Out)	Method Call	Llama al método de la API para bloquear una sala del cine.

11.3.2.4 Comportamiento

11.3.2.4.1 GUI

Se trata del principal canal de comunicación entre el administrador y la aplicación. Estará formada por grandes botones, con formato usable e intuitivo, ya que es el principal punto desde el que un administrador gestionará todo lo relacionado con el cine, ya sean salas, personal, películas o incidencias.

11.3.2.4.2 AdminUtility

“Empaqueta” toda la lógica de negocio del cliente de administrador. Se puede decir que sirve como medio de comunicación entre la interfaz de usuario y la API.

11.3.2.4.3 DataManager

Este componente recibirá una petición de la interfaz de usuario, procesará los datos recibidos, y a su vez, llamará a la API para obtener los datos de la BBDD que son requeridos. Una vez obtenidos, son devueltos a la GUI, que los mostrará por pantalla.

11.3.2.4.4 MoviesManager

La función de este componente es recibir datos de la interfaz de usuario, procesarlos, y en función de dichos datos, realizar una llamada a la API, ya sea para añadir, modificar o eliminar una película de la BBDD.

11.3.2.4.5 RoomsManager

La función de este componente es recibir datos de la interfaz de usuario, procesarlos, y en función de dichos datos, realizar una llamada a la API, ya sea para añadir, modificar o eliminar una sala de la BBDD.

11.3.2.4.6 StaffManager

La función de este componente es recibir datos de la interfaz de usuario, procesarlos, y en función de dichos datos, realizar una llamada a la API, ya sea para añadir, modificar o eliminar un miembro del personal de la BBDD.

11.3.2.4.7 IncidendeManager

Este componente tiene como misión procesar los datos que recibe de la interfaz gráfica, con el fin de identificar la sala a la que se le está tratando de añadir una incidencia, para poder llamar a la API y bloquear dicha sala. De este modo consigue que no se proyecten más películas en la misma.

11.3.3 Diagrama Contextual

N/A

11.3.4 Guía de Puntos de Variación

N/A

11.3.5 Justificación de las decisiones

N/A

11.4 API View

La vista de la API describe como este componente utilizado en cada aplicación simplificará la manera en que las aplicaciones interactúan con la base de datos sita en el servidor central y como aumentarán todas ellas su nivel de modificabilidad debido a la abstracción que proporciona este componente.

11.4.1 Presentación Principal

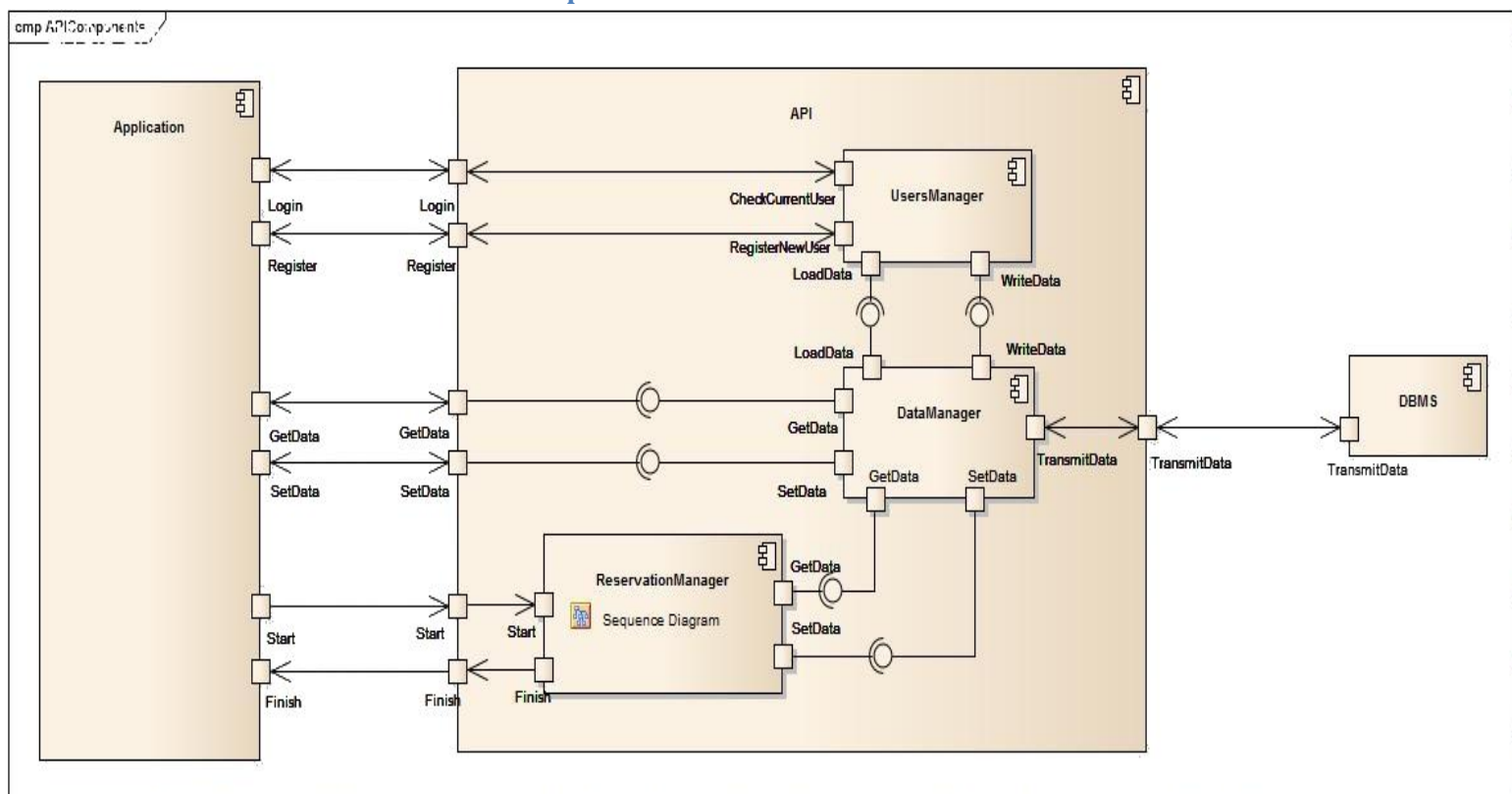


Figura 4. API View

11.4.2 Catálogo de Elementos

11.4.2.1 Elementos

Elementos	Propiedades
Application	Esto será cualquier aplicación que use la api jar
API	Esto será el jar que contiene toda la lógica de negocio y el acceso a datos.
UsersManager	Se encarga de la gestión de usuarios, tanto de los registrados en la web como de los empleados.
ReservationManager	Se encarga de la gestión de las reservas (bloqueo de butacas, gestión del pago, etc.)
DataManager	Se encarga del acceso a la base de datos
DBMS	Esto vendrá dado por el sistema de base de datos que se use

11.4.2.2 Interfaces / Puertos

11.4.2.2.1 Puertos Aplicación

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call from jar library	Envía la solicitud de login en la aplicación
Register	Puerto (In-Out)	Method Call from jar library	Envía la solicitud de registro de un cliente en el sistema
GetData	Puerto (In-Out)	Method Call from jar library	Envía solicitud de datos para mostrar al usuario
SetData	Puerto (In-Out)	Method Call from jar library	Envía solicitud de salvado de datos
Start	Puerto Out	Method Call from jar	Envía solicitud de inicio de reserva

		library	
Finish	Puerto In	Method Call from jar library	Recibe la respuesta del proceso de reserva

11.4.2.2.2 Puertos API

Interface	Tipo	Tecnología	Propiedades
Login	Puerto (In-Out)	Method Call from jar library	Recibe la solicitud de autentificacion de la aplicación y la manda al UserManager
Register	Puerto (In-Out)	Method Call from jar library	Recibe la solicitud de registro de la aplicación y la manda al UserManager
GetData	Puerto (In-Out)	Method Call from jar library	Recibe la solicitud de datos de la aplicación y la manda al DataManager
SetData	Puerto (In-Out)	Method Call from jar library	Recibe la solicitud de de salvado de datos de la aplicación y la manda al DataManager
Start	Puerto (In-Out)	Method Call from jar library	Recibe la solicitud de de inicio de reserva de la aplicación y la manda a ReservationManager
Finish	Puerto (In-Out)	Method Call from jar library	Recibe la respuesta de fin de reserva del ReservationManager y la manda a la aplicación
TransmitData	Puerto (In-Out)	JDBC	Transmite los datos a la base de datos

11.4.2.2.3 Puertos DataManager

Interface	Tipo	Tecnología	Propiedades
LoadData	Puerto (In-Out)	Method Call	Recibe la solicitud de datos y la pasa a la base de datos
WriteData	Puerto (In-Out)	Method Call	Recibe la solicitud de escritura de datos y la pasa a la base de datos
GetData	Puerto (In-Out)	Method Call	Recibe la solicitud de datos y la pasa a la base de datos
SetData	Puerto (In-Out)	Method Call	Recibe la solicitud de escritura de datos y la pasa a la base de datos
TransmitData	Puerto (In-Out)	DAO	Transmite los datos a la base de datos

11.4.2.2.4 Puertos ReservationManager

Interface	Tipo	Tecnología	Propiedades
Start	Puerto In	Method Call	Recibe la solicitud de empezar la reserva
Finish	Puerto Out	Method Call	Envía la respuesta del proceso de reserva
GetData	Puerto (In-Out)	Method Call	Manda la solicitud de datos
SetData	Puerto (In-Out)	Method Call	Manda la solicitud de escritura de datos

11.4.2.2.5 Puertos DBMS

Interface	Tipo	Tecnología	Propiedades
TransmitData	Puerto (In-Out)	JDBC	Recibe las solicitudes y devuelve los datos solicitados

11.4.3 Diagrama Contextual

N/A

11.4.4 Guía de Puntos de Variación

N/A

11.4.5 Justificación de las decisiones
N/A

11.5 Sequence View

La vista de Sequence proporciona una vista detallada del funcionamiento del sistema de compra de las entradas a través de la web.

En el caso de los pagos en la TPV el proceso es similar pero simplificado ya que no se generará código alguno ni tampoco se guardarán datos del cliente.

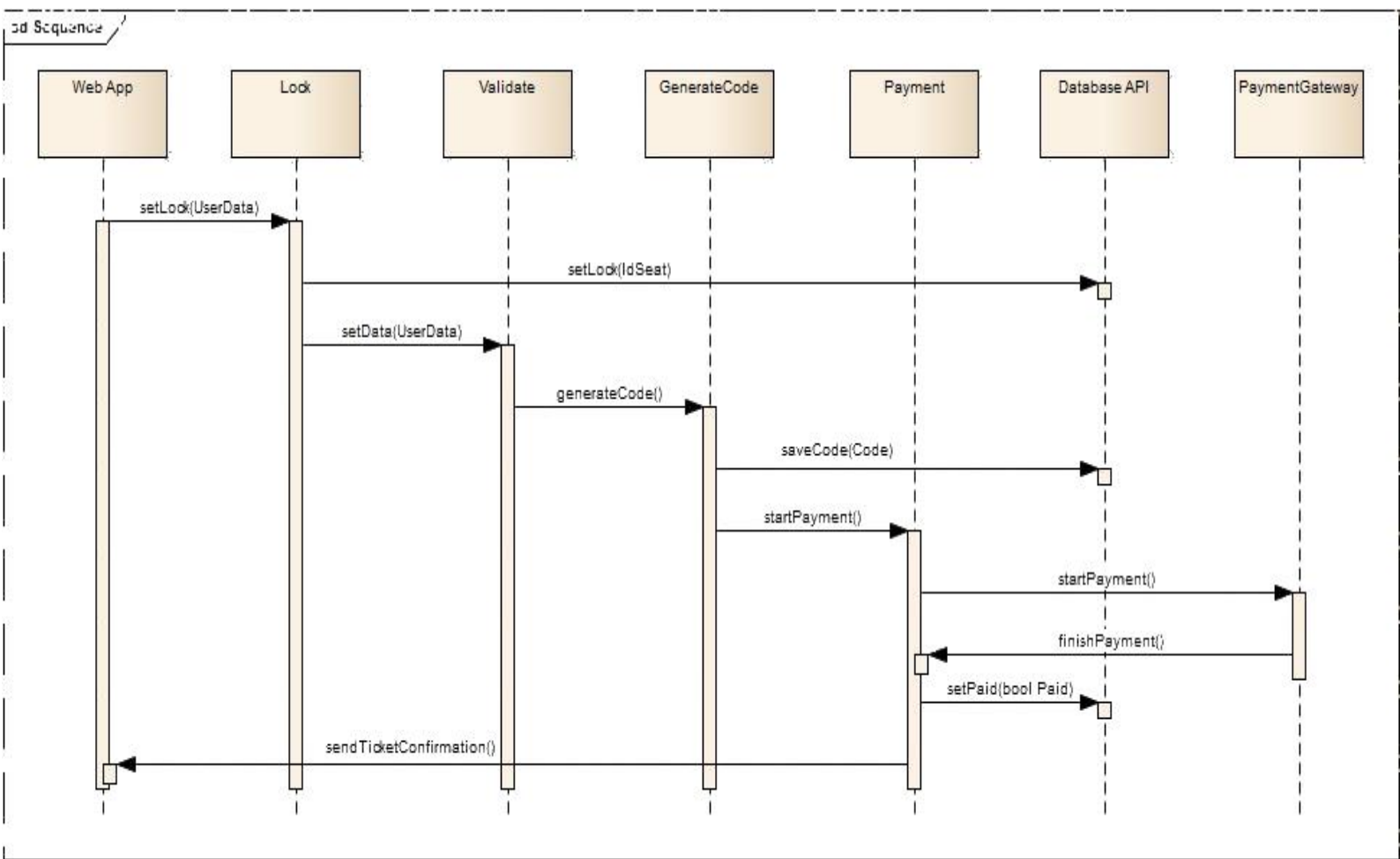


Figura 5. Sequence View

11.5.2 Catálogo de elementos

11.5.2.1 Elementos

Elemento	Propiedades
Web App	Aplicación web desde la que se iniciará la compra de las entradas.
Lock	Se encarga de bloquear el asiento elegido por el usuario para que no pueda ser reservado por otro usuario a la vez.
Validate	Se encarga de realizar la validación de los datos del usuario que reserva la entrada.
GenerateCode	Se encarga de generar el código para retirar la entrada en el cine y almacenarlo.

Payment	Se encarga de dirigir al usuario a la pasarela de pago y en función de si se realiza el pago o no, marca la entrada como pagada y activa el código generado para dicha compra.
Database API	Librería para acceder a la base de datos.
PaymentGateway	Pasarela de pago.

11.5.2.2 Relaciones

La aplicación web se relacionará a través de la API con el subsistema encargado de los pagos, el cual comenzará el proceso de bloqueo, validación, etc.

El subsistema de pagos se relacionará con la base de datos a través de la API de la misma, y se relacionará con la pasarela de pago que será proveída.

11.5.2.3 Interfaces/Puertos

N/A

11.5.2.4 Comportamiento

Un usuario desde la aplicación web comenzará la reserva de una entrada para la película y sesión que elija. Una vez lo haya elegido, la aplicación web solicitará al sistema que comience el proceso de reserva.

Lo primero que se hace es bloquear la butaca elegida, accediendo para ello a la base de datos y marcando dicha butaca como no disponible.

Tras esto se mandan los datos del usuario al sistema de validación para comprobar que los datos sean correctos y por tanto no haya fallos a la hora de hacer la reserva.

Tras verificar los datos, se genera un código pseudoaleatorio con el que el cliente podrá retirar su entrada en el cine, y será almacenado en la base de datos, aunque no será efectivo hasta que se pague la entrada.

Cuando ya se ha guardado, se pasa a la realización del pago. En este paso es en el que se dirige al usuario hacia la pasarela de pago, y el sistema queda a la espera de la respuesta de la pasarela. En cuanto se haya pagado la entrada, se accede a la base de datos para marcarla como pagada (y por tanto se activa el código de recogida), y se manda la confirmación al usuario de que su compra ha sido realizada con éxito.

11.5.3 Diagrama contextual

Esta vista es la explicación del funcionamiento del componente *TransaccionPago* (Ver 11.4.1).

11.5.4 Guía de puntos de variación

En cuanto a la TPV el funcionamiento es similar, con la excepción de que tras realizar el bloqueo de la entrada, se pasa directamente al pago de la entrada, saltándose pues los apartados de validación y generación del código.

11.5.5 Justificación de las decisiones

Las decisiones que han llevado a este comportamiento han sido:

Escenario	Atributos de calidad	Justificación
6	AT005	Al usar un código pseudoaleatorio evitamos que se puedan generar códigos fraudulentos. Para generarlo por tanto se pueden usar los datos del usuario como parte no aleatoria.
14	AT013	La reserva de la entrada debe ser realizada por completo o no realizarse, es decir en caso de algún fallo, el sistema debe quedar en la misma situación que antes de comenzar la reserva.

11.6 Deployment View

La vista de Deployment muestra los módulos que forman la solución propuesta al cliente.

11.6.1 Presentación principal

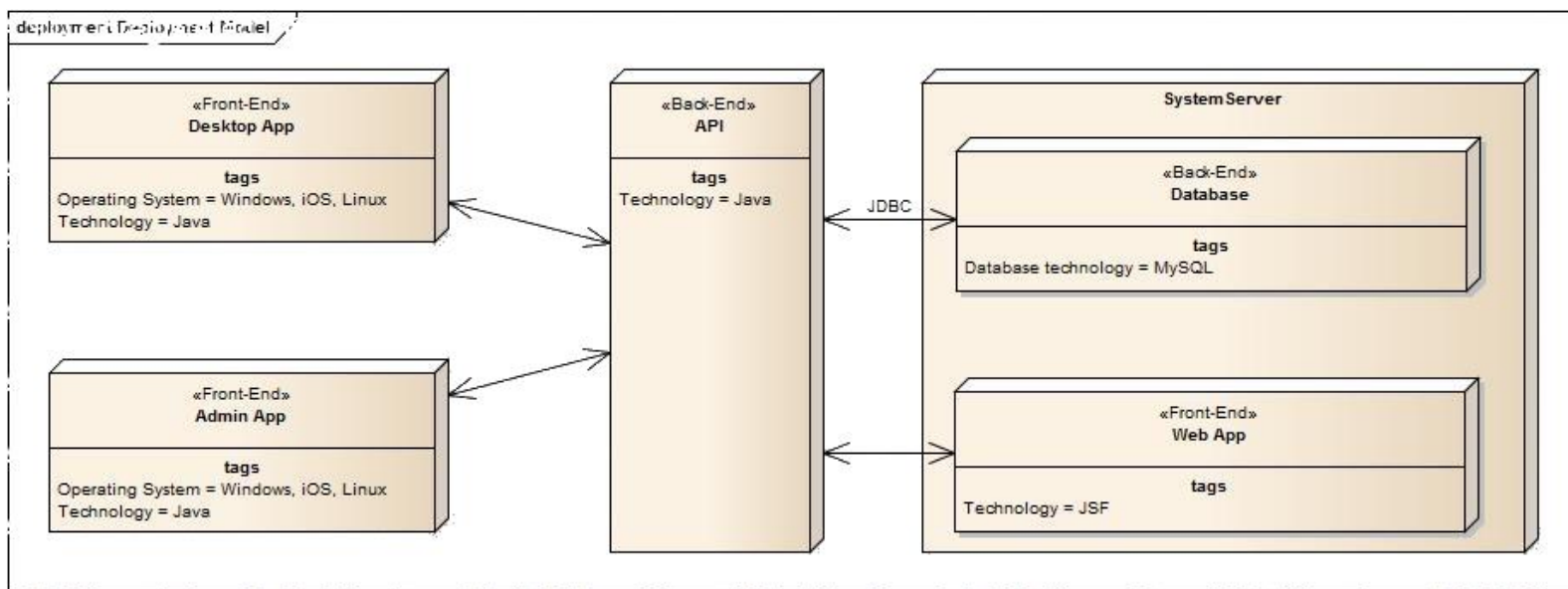


Figura 6. Deployment View

11.6.2.1 Elementos

Elemento	Propiedades
Desktop App	TPV multiplataforma que usarán los empleados del cine para la venta de entradas. Desarrollada con Java.
Admin App	Aplicación de administración también multiplataforma y también desarrollada con Java, a través de la cual se podrán realizar cambios en el cine, las películas, empleados, etc.
API	Librería que contiene los métodos de

	acceso a la base de datos, así como los métodos de autenticación, validación, pagos, etc. Estará presente en todas las aplicaciones.
SystemServer	Servidor central que albergará tanto la base de datos como la aplicación web.
Web App	Aplicación web desarrollada mediante JSF.
Database	Base de datos MySQL que contendrá todos los datos necesarios para el funcionamiento de las diferentes aplicaciones.

11.6.2.2 Relaciones

Las comunicaciones con la base de datos se realizarán a través de JDBC.

11.6.2.3 Interfaces/Puertos

N/A

11.6.2.4 Comportamiento

11.6.2.4.1 Desktop App

Soporta el TPV que estará presente en el cine, y por tanto las interacciones del empleado con el sistema.

11.6.2.4.2 Admin App

Soporta el cliente de administración del cine, y por tanto las interacciones del administrador con el sistema.

11.6.2.4.3 Web App

Soporta la aplicación web y por tanto la interacción de cualquier usuario con el sistema.

11.6.2.4.4 API

Soporta la comunicación entre las diferentes aplicaciones y la base de datos, además de proveer a las aplicaciones de los servicios que precisen, como por ejemplo la autenticación de usuarios o la verificación de datos del cliente.

11.6.2.4.5 Database

Almacena los datos que necesitan el resto de aplicaciones para su correcto funcionamiento.

11.6.3 Diagrama contextual

N/A

11.6.4 Guía de puntos de variación

N/A

11.6.5 Justificación de las decisiones

Escenario	Atributos de calidad	Justificación
4	AT003	Al contar con una aplicación de administración, los cambios son realizados con facilidad e inmediatamente.
10	AT009	Al contar con la API que

	<p>proporciona el servicio de autenticación, podemos evitar accesos no autorizados al sistema, así como llevar el control de los usuarios que acceden a él.</p>
--	---

12 Documentación Adicional

N/A

13 Índice de figuras

Figura 1. System View.....	12
Figura 2. TPVApp View.....	14
Figura 3. AdminApp View.....	17
Figura 4. API View	23
Figura 5. Sequence View	25
Figura 6. Deployment View	27