Descripción breve

Estudio inicial de la arquitectura a seguir para el proyecto ‘Entrecine’

Proyecto Entrecine

v004

Claudio López Ardura; Diego Menéndez Díaz; David Santamaría Martin; Juan Manuel Sánchez Gómez; Alejandro Fernández Fernández; Miguel Ángel González Manchado

# Planteamiento del Problema

Se necesita construir un sistema que permita gestionar automáticamente la venta de entradas de un cine. La venta de las entradas se podrá hacer vía web o desde taquilla.

# Metodología usada

Se va a realizar un estudio de arquitectura siguiendo el método de ADD (Atribute-Driven Design) y la norma del SEI (ANSI/IEEE 1471, 2000).

# Identificación de los Stakeholders

El sistema tiene una serie de interesados que incurrirán en las decisiones del diseño.

|  |  |
| --- | --- |
| Código | Stakeholder |
| ST-01 | Empleados de taquilla |
| ST-02 | Usuario de aplicación web |
| ST-03 | Administrador del sistema |
| ST-04 | Directivos del cine |
| ST-05 | Banco |

## Empleados de taquilla

Trabajadores del cine. Se encargan de atender a los clientes desde la taquilla, vendiendo las entradas en metálico o por tarjeta de crédito. Sus intereses son:

* Manejo intuitivo de la aplicación de la *TPV.*
* Control y accesos rápidos entre sesiones y películas para despachar a los clientes en el menor tiempo posible.

## Usuario de aplicación web

Son los clientes propiamente dichos. Aunque tenga palabra directa en las decisiones sobre la arquitectura, es necesario tenerlos en cuenta porque forman parte una parte esencial del sistema. Los intereses que comúnmente se asocian con los clientes son:

* Fácil acceso a la cartelera del cine, sin navegaciones innecesarias ni información de difícil acceso.
* Facilidad en el pago mediante tarjeta de crédito.
* Control sobre sus datos registrados en el sistema Entrecine.

## Administrador del sistema

Es el trabajador del cine que se encarga de mantener actualizados los datos sobre las películas y sesiones, así como los precios. También tiene acceso a los datos de los usuarios y al control de las transacciones bancarias. Sus intereses son:

* Manejo CRUD sobre todos los datos del sistema.
* Configuración de los parámetros de la aplicación.

## Directivos

Se trata del equipo directivo del cine. Son responsables de los presupuestos y toman las decisiones que comprometen fondos de dicho presupuesto. Sus intereses son:

* Coste del desarrollo de la aplicación bajo.
* Tiempo de desarrollo de la aplicación corto.

## Banco

Aunque no actúe directamente sobre el sistema, es necesario tener relación con él porque es el que proporciona los métodos para los pagos vía tarjeta de crédito. Sus intereses son:

* Ofrece una pasarela de pago con la que es sistema debe comunicarse para la realización de los pagos.

# Atributos de calidad

Para el sistema se ha definido los siguientes atributos de calidad:

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo |
| AT-01 | Venta online de entradas 24x7 | Disponibilidad |
| AT-02 | Actualización de cartelera y precios inmediata | Modificabilidad |
| AT-03 | Robustez de la aplicación en horas putas | Rendimiento |
| AT-04 | Concurrencia de ventas | Rendimiento |
| AT-05 | Acceso rápido al sistema de ventas desde la TPV | Rendimiento |
| AT-06 | Conexión segura con la pasarela de pago | Seguridad |
| AT-07 | Protección de datos de los usuarios | Seguridad |
| AT-08 | Funcionalidad de la página web en los navegadores más comunes | Adaptabilidad |
| AT-09 | Fácil manejo de la web y de la compra de entradas | Usabilidad |
| AT-10 | Administración completa y de manejo sencillo | Usabilidad |
| AT-11 | Comprobaciones de que el comprobante de compra se corresponda con las ventas | Testabilidad |
| AT-12 | Tiempo de desarrollo corto | Time to market |
| AT-13 | Coste de desarrollo bajo | Coste-Beneficio |

# Relación entre Stakeholders y Atributos de Calidad

Los stakeholder y los atributos de calidad están relacionados directamente, siendo los intereses de los stakeholder la principal razón de inclusión de atributos de calidad que en un principio pudieron no tenerse en cuenta. Esta tabla muestra la relación entre ambos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ST-01 | ST-02 | ST-03 | ST-04 | ST-05 |
| AT-01 |  | **X** |  | **X** |  |
| AT-02 |  | **X** | **X** |  |  |
| AT-03 |  | **X** |  |  |  |
| AT-04 | **X** | **X** |  |  |  |
| AT-05 | **X** | **X** |  |  |  |
| AT-06 |  | **X** |  |  | **X** |
| AT-07 |  | **X** | **X** |  |  |
| AT-08 |  | **X** |  |  |  |
| AT-09 |  | **X** |  |  |  |
| AT-10 |  |  | **X** |  |  |
| AT-11 | **X** | **X** |  |  |  |
| AT-12 |  |  |  | **X** |  |
| AT-13 |  |  |  | **X** |  |

# Descripción de la solución

# Escenarios de calidad

A partir de toda la información anterior se definen los escenarios de calidad del sistema. La tabla siguiente muestra los escenarios y los atributos con los que están relacionados.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Escenario Nº | Fuente del estímulo | Estímulo | Entorno | Artefacto | Respuesta | Medición de la respuesta | Atributo de calidad asociado |
| 1 | Página web | Atención a peticiones | Explotación | Servidor | Atención 24x7 | Denegaciones de servicio < 1% | **AT-01** |
| 2 | Cambio de cartelera | Actualización de la cartelera | Explotación | Servidor, Sistema de Administración | Actualización de la cartelera y precios | Ventas en cartelera desactualizada: 0 | **AT-02** |
| 3 | Hora punta de ventas | Múltiples accesos al servicio web | Explotación | Servidor, Sistema de ventas | Activación de servidores espejo | Ralentizaciones y errores < 5% | **AT-03** |
| 4 | Venta de entrada | Acceso concurrente a misma butaca | Explotación | Servidor, Sistema de ventas | Venta de la butaca a un solo usuario | Errores en ventas: 0 | **AT-04** |
| 5 | TPV | Venta de entradas en taquilla | Explotación | TPV, Sistema de ventas | Cliente despacho rápidamente | Tiempo de venta en taquilla < 30s | **AT-05** |
| 6 | Venta de entrada | Pago con tarjeta de crédito | Explotación | Sistema de ventas | Transacción segura con el banco | Cargos inválidos: 0 | **AT-06** |
| 7 | Acceso a datos del usuario | Acceso a información | Explotación | Servidor | Acceso por contraseña | Los usuarios solo pueden acceder a sus datos | **AT-07** |
| 8 | Uso de página web | Envío de página web | Explotación | Servidor | Página web visualizada correctamente | Debe ser usable en IE, Firefox, Chrome… | **AT-08** |
| 9 | Página web | Acceso a la cartelera y ventas | Explotación | Servidor | Información del cine fácilmente accesible | Perdidas de clientes < 5% | **AT-09** |
| 10 | Sistema de administración | Acceso y cambios en el sistema | Explotación | Sistema de administración, sistema de persistencia | Actualizaciones rápidas | Tiempo en actualizar cartelera < 5min/película | **AT-10** |
| 11 | TPV | Comprobante de compra online | Explotación | TPV, Sistema de ventas | Validación y entrega de entrada | Errores de comprobación: 0 | **AT-11** |
| 12 | Sistema en desarrollo | Sistema en desarrollo | Desarrollo | Sistema | Tiempo bajo de desarrollo | Time to market < 2 meses | **AT-12** |
| 13 | Sistema en desarrollo | Sistema en desarrollo | Desarrollo | Sistema | Coste bajo de producción | Coste de desarrollo: 0€ | **AT-13** |

# Vistas

En este apartado se realizara la descripción de las vistas del sistema y de sus subsistemas, así como la vista de despliegue del sistema y su empaquetamiento para la implementación.

|  |  |  |
| --- | --- | --- |
| Vistas | Referencia | Escenarios |
| System |  | 2,5,6 |
| Server |  | 1,3,7,8,9 |
| Central System |  | 3,4,5,6,11 |
| Administration System |  | 2,10 |
| Database System |  | 2,7,10 |
| Package Model |  | 12,13 |
| Deployment |  |  |

## System View

LA vista del sistema describe los 4 subsistemas, *Server, Central System, Database System y Administration System,* y las interfaces con las que se comunican.

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Vista | Propiedades |
| Server | Subsistema encargado de interactuar con el cliente. |
| Central System | Subsistema que gestiona todo el sistema de ventas. |
| Administration System | Subsistema dedicado a la gestión de sistema. |
| Database System | Subsistema que protege los datos y los almacena. |

#### Relaciones

*Server* interactúa con *Central System* para obtener los datos necesarios para el usuario y realizar compras.

*Central System* se comunica con *Database System* para obtener y salvar toda la información procesada.

*Administration System* se encarga de modificar los datos con lo que trabaja *Database System*

#### Interfaces/Puertos

##### Server

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| HttpRequest | Proveída | Web Service | Recibe las peticiones del navegador del cliente. |
| Register | Requerida |  | Solicita la creación de un nuevo usuario registrado en el sistema. |
| Login | Requerida |  | Solicita el inició de sesión de un usuario. |
| RequestBillboard | Requerida |  | Solicita la actualización de la cartelera. Esta actualización se solicitará cuando se realice una modificación en la cartelera. |
| ServerPayment | Requerida |  | Solicita la realización de la compra de una entrada y del pago con tarjeta de crédito a través de la pasarela de pago. |
| ObtainTicket | Requerida |  | Solicita la obtención del comprobante de la compra de la entrada vía web. |
| Connect | Requerida |  | Solicita la conexión del servidor con las actualizaciones de la cartelera. |

##### Central System

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| Register | Proveída |  | Reciba la creación de un nuevo usuario en el sistema. |
| Login | Proveída |  | Recibe la solicitud de inicio de sesión. Si el usuario se mantiene inactivo más de 30 minutos se cierra la sesión. |
| RequestBillboard | Proveída |  | Recibe una petición de actualización de la cartelera. Si la cartelera ha sido actualizada se envía toda la nueva información. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ServerPayment | Proveída |  | Recibe la solicitud de compra de una entrada y del pago a través de la pasarela de pago.  Cuando se intente comprar una entrada de una butaca ya vendida se impedirá la venta.  Si el pago es realizado por un usuario registrado se asocia la venta al mismo. | |
| ObtainTicket | Proveída |  | | Recibe la solicitud de la entrega del comprobante de compra online. |
| Connect | Proveída |  | | Recibe la solicitud de conexión con las actualizaciones de la cartelera. |
| SaveUser | Requerida |  | | Solicita el almacenamiento del usuario recién registrado y creado. |
| SaveUser | Requerida |  | | Solicita el almacenamiento del usuario recién registrado y creado. |
| LoadUser | Requerida |  | | Solicita la carga de los datos de un usuario que inicia sesión. |
| GetTicket | Requerida |  | | Solicita la obtención del comprobante y la venta de entrada del código escaneado. |
| SaveTicket | Requerida |  | | Solicita el almacenamiento del comprobante recién creado y asociado a una venta online. |
| SaveSale | Requerida |  | | Solicita el almacenamiento de una venta realizada. |
| LoadBillboard | Requerida |  | | Solicita la carga de la cartelera. |
| ConnectBillboard | Requerida |  | | Solicita la conexión de un servidor que haya realizado con las actualizaciones de la cartelera. |
| TPVRequest | Proveída |  | | Recibe las solicitudes de la TPV para la realización de los pagos, obtención de cartelera y cancelaciones de comprobantes. |
| DoPayment | Requerida |  | | Solicita la realización de un pago de tarjeta de crédito a través de la pasarela de pago. |

##### Administration System

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetUsers | Requerida |  | Solicita la obtención de toda la información de los usuarios. |
| GetSales | Requerida |  | Solicita la obtención de las entradas vendidas |
| GetRawBillboard | Requerida |  | Solicita la cartelera en un formato editable. |
| UpdateBillboard | Requerida |  | Actualiza la cartelera con los datos nuevos o con los antiguos modificados. |

##### Database System

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| SaveUser | Proveída |  | Recibe la solicitud del almacenamiento de un nuevo usuario en la base de datos.  Antes de guardarlo encripta los datos para protegerlos. |
| LoadUser | Proveída |  | Recibe la solicitud de carga de los datos de un usuario de la base de datos.  AL obtener los datos los desencripta para que puedan ser entendidos. |
| GetTicket | Proveída |  | Recibe la solicitud de obtención del comprobante de una compra online. |
| SaveTicket | Proveída |  | Recibe la solicitud del almacenamiento del comprobante de una venta online. |
| SaveSale | Proveída |  | Recibe la solicitud del almacenamiento de una nueva venta.  Antes de guarda los datos los encripta para proteger las ventas. |
| LoadBillboard | Proveída |  | Recibe la solicitud de carga la información de la cartelera de la base de datos. |
| ConnectBillboard | Proveída |  | Recibe la solicitud de conectar un servidor con las actualizaciones de la cartelera |
| GetUsers | Proveída |  | Recibe la solicitud de obtención de la información de todos los usuarios.  Cuando los obtiene los desencripta. |
| GetSales | Proveída |  | Recibe la solicitud de obtención de información sobre las entradas vendidas.  Cuando obtiene las ventas las desencripta. |
| GetRawBillboard | Proveída |  | Recibe la solicitud de obtención de la cartelera en formato editable. |
| UpdateBillboard | Proveída |  | Recibe la solicitud de modificación de la cartelera. Cuando esta actualización ocurre se notifica a los servidores conectados la actualización. |

#### Comportamiento

##### Server

Se encarga de atender las peticiones de los usuarios a través de la web. Tiene una copia de la cartelera en una base de datos nativa, que se actualiza en tiempo real.

Cuando el usuario quiere comprar una entrada se conecta al sistema central para llevar a la venta, proveyendo al usuario la interfaz para la realización de un pago con tarjeta de crédito.

Al finalizar una venta obliga a la impresión del comprobante de la entrada comprada.

##### Central System

Es el sistema que rige todas las ventas. Tanto los servidores como las TPVs acceden a él a través de un adaptador que permiten adaptar numerosas peticiones al subsistema aliviando la carga de trabajo gestionándola con un pool de accesos.

Comprueba que las entradas vendidas no sean sobre una misma butaca.

Gestiona los accesos de los usuarios registrados y crea usuarios nuevos cuando es necesario.

Gestiona las peticiones de los servidores y TPVs para la actualización/obtención de la cartelera.

Se conecta con la pasarela de la pago para la realización de pagos con tarjeta de crédito con el banco.

##### Administration System

Es la interfaz de administración del sistema. Actúa directamente con Database System para modificar los parámetros de la cartelera, así como para obtener la información de las entradas vendidas y de los usuarios.

##### Database System

Es un sistema dedicado a proteger toda la información de la base de datos. Encripta ´y desencripta la información de los usuarios y las ventas para trabajar con ellas desde el resto de sistemas.

Contiene la cartelera en la forma primitiva, de forma que la el tamaño de los datos que tiene sea mínimo, y se encarga de transformarla para enviarla a los otros sistemas.

### Diagrama Contextual

N/A

### Guía de puntos de variación

N/A

### Justificación de las decisiones

Las decisiones que han llevado a este diseño son:

|  |  |  |
| --- | --- | --- |
| Escenario | Atributo de calidad | Justificación |
| 2 | **AT-02** | Las interfaces ***“Connect “y*** ***“ConnectBillboard”*** permiten mantener actualizada la cartelera en tiempo real una vez un S*erver* se haya conectado a *Central System.* |
| 5 | **AT-05** | Las interfaces de acceso de la TPV permiten que varias puedan conectarse a *Central System* y poder realizar las ventas y cancelaciones de entradas. |
| 6 | **AT-06** | La interfaz ***“GatewayPayment”*** permite conectarsecon la pasarela de pago suministrada por el banco para la realización de pagos con tarjeta de crédito. |

## Server View

La vista de *Server* muestra el primer nivel de descripción de los componentes de *Server.*

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Vista | Propiedades |
| PresentacionManager | Es la capa de presentación del servidor web. |
| ServerLogicManager | Capa lógica del servidor |
| ServerBillboardManager | Encargado de actualizar la base de datos con la cartelera |
| ServerBillboardDB | Base de datos nativa con la cartelera |

#### Relaciones

*PresentationManager* se comunica con *ServerLogicManager* para pasarle las interacciones del usuario que necesiten de cómputos lógicos, y para obtener la cartelera.

*ServerLogicManager* pide a *ServerBillboardDB* la cartelera.

*ServerBillboardManager* se encarga de actualizar la cartelera almacenada en *ServerBillboardDB*.

#### Interfaces/Puertos

##### PresentationManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| HttpRequest | Proveída | WebService | Ver 8.1.1.3.1 |
| UserAction | Puerto (Out) | Method Call | Solicita computar las acciones del usuario que necesiten llevar a cabo de manera separada a la capa de presentación |
| ObtainBillboard | Puerto (In-Out) | Method Call | Recoge bajo demanda la cartelera del cine |

##### ServerLogicManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| Register | Requerida |  | Ver 8.1.1.3.1 |
| Login | Requerida |  | Ver 8.1.1.3.1 |
| ServerPayment | Requerida |  | Ver 8.1.1.3.1 |
| ObtainTicket | Requerida |  | Ver 8.1.1.3.1 |
| GetBillboard | Puerto (In-Out) | DAO (ODBC) | Solicita a la base de datos la obtención de la cartelera actualizada.  Si la no se han producido cambios no pasa nada. |
| UserAction | Puerto (In) | Method Call | Realiza las acciones del usuario necesite. |

##### ServerBillboardManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| RequestBillboard | Requerida |  | Ver 8.1.1.3.1 |
| Connect | Requerida |  | Ver 8.1.1.3.1 |
| SetBillboard | Puerto (Out) | DAO (ODBC) | Solicita modificar la base de dato en cuando se produce una actualización de la base de datos general. |

##### ServerBillboardDB

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetBillboard | Acceso DB | DAO (ODBC) | Devuelve los datos actualizados de la cartelera. SI no se han producido cambios desde el último acceso no se producen cambios. |
| SetBillboard | Acceso DB | DAO (ODBC) | Añade e introduce las modificaciones de la base de datos que se hayan producido. |

#### Comportamiento

##### PresentationManager

Este módulo se encarga de la presentación del que verá el usuario en su navegador.

En función del navegador usado, el sistema devolverá a través de la interfaz *HttpRequest*  la versión de la página que mejor se ajuste al navegador.

Para mostrar la página, necesita cargar la cartelera (*ObtainBillboard*) y con ella trabaja para que el usuario pueda interactuar con las distintas entidades que las conforman.

Cuando un usuario realiza una acción, como por ejemplo registrarse, traslada la petición con la interfaz *UserAction.*

##### ServerLogicManager

Se encarga de la lógica realizada por el servidor.

Cuando la capa de presentación necesita la cartelera (*ObtainBillboard*) pide, mediante un *DAO* a la base de datos toda la información que necesite (*GetBillboard*). Luego la prepara para que pueda ser entendida por *PresentationManager* y se la envía.

Cuando recibe acciones del usuario desde *PresentationManager* (*UserAction*)*,* puede desencadenar la necesidad de realizar acciones ajenas al módulo. Cuando se trata de una acción de registro, comprueba que los datos sean válidos y los envía al siguiente subsistema. Lo mismo pasa con *Login,* que preara los datos para un inicio de sesión; *ServerPayment,* validando que el número de tarjeta de crédito sea correcto; y ObtainTicket, que se desencadena tras *ServerPayment* y obliga a imprimir el comprobante tras obtenerlo.

##### ServerBillboardManager

Este módulo mantiene la base de datos al día.

Mediante la interfaz *Connect* se registra en el sistema de subscripciones de la actualización de la cartelera, permitiendo que esta se actualice cuando se realiza algún cambio.

Cuando se produce una actualización el módulo realiza una petición (*RequestBillboard*) para obtener la información necesaria de cartelera y la almacena en mediante una *DAO*  en la base de datos del servidor (*SetBillboard*)*.*

##### ServerBillboardDB

Gestiona el almacenamiento de datos de cartelera en una base de datos nativa.

Permite modificar y añadir elementos a la cartelera (*SetBillboard*) y recuperar los datos de la misma (*GetBillboard*).

### Diagrama contextual

Esta vista es el desarrollo del subsistema *Server.* (Ver 8.1.1)

### Guía de puntos de variación

*PresentationManager* y *ServerLogicManager* pueden estar implementados mediante un Modelo Vista-Controlador y actuando como un módulo único, pero con una clara separación de las capas.

El ServerBillboardManager debería esta implementado por un patrón *Observer* que permita estar suscrito a la actualización de la cartelera, utilizando un *Listener* como elemento de paso por la interfaz *Connect*  y que al ser notificado cause la llamada de *RequestBillboard.*

### Justificación de las decisiones

Las decisiones que han llevado a este diseño son:

|  |  |  |
| --- | --- | --- |
| Escenario | Atributo de calidad | Justificación |
| 1 | **AT-01** | El subsistema *Server* está pensado de manera que puedan existir múltiples servidores en el sistema y en caso de que una fallo haya otro para atender las peticiones de los usuarios |
| 3 | **AT-03** | Debido a la independencia de este subsistema frente al sistema, en caso de necesidad, se pueden añadir servidores para poder soportar una mayor carga de trabajo. Además debido a la disposición de la base de datos nativa, no necesita de acceder al resto del sistema para mostrarla al usuario, por lo que disminuye también los cuellos de botella. |
| 7 | **AT-07** | Un usuario en sesión tiene acceso a sus datos, pero estos se encuentran en el servidor, de manera que solo ese usuario en esa sesión puede acceder a ellos. En caso de que una sesión se encuentre inactiva mucho rato esta se cierra, protegiendo los datos del usuario. |
| 8 | **AT-08** | Los navegadores actuales son muy heterogéneos, puesto que no es lo mismo acceder desde un ordenador que desde un dispositivo móvil, por lo que manteniendo una capa de presentación que sea capaz de identificar el navegador y responder en consecuencia, permitiendo una mejor visualización en una mayor gama de dispositivos. |
| 9 | **AT-09** | En la capa de presentación debe tenerse en cuenta la usabilidad del sistema, por lo que manteniéndola aislada en un módulo es posible modificarla en función de la necesidad sin tener que modificar otros módulos. |

## Central System View

### Presentación principal



### Catálogo de elementos

#### Elementos

#### Relaciones

#### Interfaces/Puertos

#### Comportamiento

### Diagrama contextual

### Guía de puntos de variación

### Justificación de las decisiones

## Administration System View

La vista de *Administration System* muestra el primer nivel de descripción de los componentes de *Administration System.*

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Vista | Propiedades |
| AdminPresentation | Es la capa de presentación de la aplicación del administrador. |
| AdminLogicManager | Capa lógica del administrador. |
| CommitManager | Encargado de llevar a cabo las actualizaciones de la cartelera sin sobrecargar el sistema. |
| InfoReader | Obtiene la información del sistema que necesite el administrador |

#### Relaciones

AdminPresentation pide a AdminLogicManager la información que necesite en un momento determinado. InfoReader recibe la petición desde AdminLogicManager y la lleva a cabo, tras lo cual devuelve los datos recogidos.

AdminPresentación lleva a cabo unos cambios en la cartelera que son guardados y se lo comunica a AdminLogicManager. Este pasa lo cambios a CommitManager que lleva a cabo la petición de actualizar la cartelera tras un periodo de tiempo.

#### Interfaces/Puertos

##### AdminPresentation

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| EditBillboard | Puerto (Out) | Method Call | Solicita la salvaguarda de las modificaciones hechas en la cartelera.  También puede solicitar la cancelación de una actualización. |
| Billboard | Puerto (In-Out) | Method Call | Pide y recoge los datos de la última versión de la cartelera en un formato editable. |
| Sales | Puerto (In-Out) | Method Call | Pide y recoge los datos de todas las entradas vendidas en un ámbito determinado por el administrador. |
| Users | Puerto (In-Out) | Method Call | Pide y recoge los datos de todos los usuarios. |

##### AdminLogicManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| UpdateBillboard | Requerida |  | Ver 8.1.1.3.3 |
| GetRawBillboard | Requerida |  | Ver 8.1.1.3.3 |
| GetSales | Requerida |  | Ver 8.1.1.3.3 |
| GetUsers | Requerida |  | Ver 8.1.1.3.3 |
| EditBillboard | Puerto (In) | Method Call | Recibe los peticione para la realización de cambios en la cartelera y los pasa a *CommitManager* |
| Billboard | Puerto (In-Out) | Method Call | Recibe la petición de obtención de la cartelera y la traslada a *InfoReader.* Cuando contesta, devuelve los datos. |
| Sales | Puerto (In-Out) | Method Call | Recibe la petición de los datos de ventas y los parámetros con los que se buscan.  Pasa la petición a *InfoReader* y devuelve los datos recogidos cuando contesta. |
| Users | Puerto (In-Out) | Method Call | Recibe la petición de obtener todos los datos de los usuarios.  Pasa la petición a *InfoReader* y devuelve los datos recogidos cuando contesta. |

##### CommitManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| UpdateBillboard | Puerto (Out) |  | Actualiza los datos del sistema con modificados de la cartelera que tenga guardados tras un periodo de tiempo |
| EditBillboard | Puerto (In) | Method Call | Recibe la solicitud de salvaguardar los datos de una actualización de cartelera.  Los guarda en memoria secundaria hasta que se haga una actualización.  Puede recibir la cancelación de la actualización de los datos que se encuentren en memoria secundaria. |

##### InfoReader

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetRawBillboard | Requerida |  | Ver 8.1.1.3.3 |
| GetSales | Requerida |  | Ver 8.1.1.3.3 |
| GetUsers | Requerida |  | Ver 8.1.1.3.3 |
| Billboard | Puerto (In-Out) | Method Call | Recibe la petición de obtención los datos de la última versión de la cartelera en un formato editable.  Cuando los obtiene los devuelve. |
| Sales | Puerto (In-Out) | Method Call | Recibe la petición de obtener los datos de las ventas de entradas en un ámbito.  Cuando recibe los datos los devuelve a quien los pidió. |
| Users | Puerto (In-Out) | Method Call | Recibe la petición para obtener la información de todos los usuarios.  Cuando la recibe la devuelve. |

#### Comportamiento

##### AdminPresentation

Es el módulo encargado de proporcionar la interfaz gráfica con la que trabaja el administrador.

Cuando el administrador quiere ver las ventas de una determinada entrada de la cartelera deberá indicar los parámetros por los que se busca (*Sales*).

También puede visualizar los usuarios registrado en el sistema (*Users*).

Para realizar una modificación de la cartelera, primero es necesario obtenerla (*Billboard).* Desde ella se pueden realizar los cambios pertinentes y después se pueden guardar para que se actualicen en el sistema (*EditBillboard*).

##### AdminLogicManager

Se encarga de encapsular las peticiones del administrador para que las lleven a cabo los módulos que los componen.

Las interfaces *Billboard, Sales* y *Users*  reciben las peticiones de datos y las trasladan a *InfoReader* cuyas peticiones internas se llevan a través de las interfaces *GetRawBillboard, GetSales* y *GetUsers.*

Cuando llega una petición de editar la cartelera por la interfaz *EditBilboard* la pasa a *CommitManager.* Cuando *CommitManager* realiza una actualización, pasa a través de la interfaz *UpdateBillboard.*

##### CommitManager

Este módulo se encarga de gestionar las actualizaciones de la cartelera de manera que no se produzcan sobrecargas en el sistema por hacer demasiados cambios en poco tiempo.

Cuando recibe la cartelera modificada (*EditBillboard*) la almacene en una memoria secundaria. Cuando ha pasado un período determinado de tiempo o la cantidad de información alcanza cierto nivel, lleva a cabo la actualización de la cartelera en el sistema mediante una transacción para evitar que se produzcan problemas por el medio y deje a medio actualizar la cartelera.

Si la actualización no se ha producido es posible borrar todos los cambios producidos a través del puerto *EditBillboard*.

##### InfoReader

Es el encargado de llevar a cabo las peticiones para obtener la información y datos que el administrador necesite.

Los puertos *GetRawBillboard y Billboard* se encargan de pedir, recibir y enviar los datos relacionados con la cartelera del sistema central al administrador.

Los puertos *GetSales* y *Sales* hacen lo mismo con las ventas de entradas, pero a través de unos parámetros indicados por el administrador para recibir las ventas de un ámbito de la cartelera.

Los puertos *GetUsers* y *Users* funcionan de igual manera pero con la información de los usuarios.

### Diagrama contextual

Esta vista es el desarrollo del subsistema *Administration System.* (Ver 8.1.1)

### Guía de puntos de variación

El módulo A*dminPresentation* de puede implementarse de muchas maneras, por lo que no existen restricciones al respeto, siempre y cuando cumpla lo requerimientos para conectarse y mostrar la información.

*CommitManager* puede tener una pequeña base de datos donde almacenar los cambios antes de trasladarlos, o puede almacenarlos en memoria.

InfoReader está presentado como un solo módulo, pero puede estar implementado con una división que separe los 3 tipos de peticiones de manera que los cambios en uno no afecten a los otros.

### Justificación de las decisiones

Las decisiones que han llevado a este diseño son:

|  |  |  |
| --- | --- | --- |
| Escenario | Atributo de calidad | Justificación |
| 2 | **AT-02** | Las actualizaciones que se lleven a cabo se hacen de manera transaccional, de manera que un usuario solo tendrá la última versión válida de la cartelera. Además los cambios que se haga en la cartelera se hacen en grupos gracias al *CommitManager*, por lo que el sistema no tiene que actualizar los servidores cada vez que se hagan pequeños cambios. |
| 10 | **AT-10** | La interfaz se presenta de manera que permite realizar cambios en la cartelera en la cartelera. Cuando uno de esos cambios se produce se guarda en *CommitManager,* pero no realiza la actualización inmediatamente, por lo que el administrador puede deshacer los cambios antes de que afecten a todo el sistema. |

## Database System View

La vista de *Database System* muestra el primer nivel de descripción de los componentes de *Database System.*

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Vista | Propiedades |
| UserDBManager | Módulo gestor de encriptación y almacenamiento de usuarios. |
| SalesDBManager | Módulo gestor de encriptación y almacenamiento de venta de entradas. |
| BillboardDBManager | Módulo gestor de almacenamiento de la cartelera. |
| GeneralDB | Base de datos general que contiene todos los datos del sistema. |

#### Relaciones

*UserDBManger*  se conecta con *GeneralDB* para almacenar los usuarios tras encriptar sus datos y para obtenerlos y desencriptarlos para su uso en otras partes del sistema.

*SalesDBManager* esta relacionada de forma similar con *GeneralDB* pero añadiendo la gestión de los comprobantes de venta.

*BillboardDBManager* está relacionada con *GeneralDB* para la manipulación y obtención de la cartelera.

#### Interfaces/Puertos

##### UserDBManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| SaveUser | Proveída |  | Ver 8.1.1.3.4 |
| LoadUser | Proveída |  | Ver 8.1.1.3.4 |
| GetUsers | Proveída |  | Ver 8.1.1.3.4 |
| Save EncryptedUser | Puerto (Out) | DAO (ODBC) | Guarda los datos encriptados de un nuevo usuario. |
| Get EncryptedUser | Puerto (In-Out) | DAO (ODBC) | Carga los datos del usuario buscado por identificador de inicio de sesión para desencriptarlo y enviarlo. |

##### SalesDBManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| SaveSale | Proveída |  | Ver 8.1.1.3.4 |
| SaveTicket | Proveída |  | Ver 8.1.1.3.4 |
| GetTicket | Proveída |  | Ver 8.1.1.3.4 |
| GetSales | Proveída |  | Ver 8.1.1.3.4 |
| Save EncryptedSale | Puerto (Out) | DAO (ODBC) | Guarda los datos encriptados de un nueva entrada vendida y indicando la butaca elegida y su tipo (tarjeta o metálico). |
| Get EncryptedSales | Puerto (In-Out) | DAO (ODBC) | Carga todas las ventas de entradas realizadas para desencriptarlas y enviarlas. |
| Save EncryptedTicket | Puerto (Out) | DAO (ODBC) | Guarda encriptado el comprobante de venta online asociado a una determinada entrada. |
| Get EncryptedTicket | Puerto (In-Out) | DAO (ODBC) | Obtiene el comprobante asociado a un determinada venta lo desencripta y lo envía |

##### BillboardDBManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| ConnectBillboard | Proveída |  | Ver 8.1.1.3.4 |
| LoadBillboard | Proveída |  | Ver 8.1.1.3.4 |
| UpdateBillboard | Proveída |  | Ver 8.1.1.3.4 |
| GetRawBillboard | Proveída |  | Ver 8.1.1.3.4 |
| SaveBillboard | Puerto (Out) | DAO (ODBC) | Guarda los datos nuevos y modificados de la cartelera en la base de datos. |
| GetBillboard | Puerto (In-Out) | DAO (ODBC) | Carga la cartelera para enviarla. |

##### GeneralDB

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| Save EncryptedUser | Acceso DB | DAO (ODBC) | Escribe en la base de datos un nuevo usuario. |
| Get EncryptedUser | Acceso DB | DAO (ODBC) | Obtiene los datos encriptados del usuario buscado. |
| Save EncryptedSale | Acceso DB | DAO (ODBC) | Escribe en la base de datos una nueva venta de entrada. |
| Get EncryptedSales | Acceso DB | DAO (ODBC) | Obtiene los datos encriptados de todas las ventas. |
| Save EncryptedTicket | Acceso DB | DAO (ODBC) | Escribe en la base de datos el comprobante de la venta de una entrada. |
| Get EncryptedTicket | Acceso DB | DAO (ODBC) | Obtiene los datos encriptados del comprobante asociado a la entrada vendida. |
| SaveBillboard | Acceso DB | DAO (ODBC) | Escribe en la base de datos la cartelera con las últimas modificaciones. |
| GetBillboard | Acceso DB | DAO (ODBC) | Obtiene todos los datos de la cartelera. |

#### Comportamiento

##### UserDBManager

Este módulo se encarga de la encriptación y desencriptación de la información sobre los usuarios de la base de datos.

Cuando recibe los datos de un nuevo usuario registrado en el sistema (*SaveUser*) encripta los datos para luego almacenarlos en la base de datos a través de un *DAO* (*SaveEncryptedUser*).

Cuando necesita acceder a los datos de un usuario que acaba de iniciar sesión (*LoadUser*) primero los carga de la base de datos (*GetEncryptedUser*), luego los desencripta y lo envía a quien los pidió.

De manera similar ocurre cuando el administrador del sistema pide los datos de los usuarios (*GetUsers*), pero cargando todos los usuarios y desencriptandolos antes de enviarlos.

##### SalesDBManager

Este módulo se encarga de encriptar las ventas de entradas y los comprobantes asociadas a estas.

Cuando se realiza un venta en el sistema es enviada a este módulo para almacenarla en la base de datos (*SaveSale*), donde se encripta y se manda a guardar (*SaveEncryptedSale*). Si la venta ha sido a través del servicio Web, paralelamente se creó un comprobante asociado a la entrada, que también es enviado a este módulo (*SaveTicket*) para encriptarlo y guardarlo en la base de datos (*SaveEncryptedTicket*).

Si se realiza la cancelación desde una TPV se requiere conocer entrada a la que está asociada (*GetTicket*) por lo que se busca (*GetEncryptedTicket*) en la base de datos y se desencripta antes de enviarla.

Cuando un administrador necesita comprobar las entradas vendidas (*GetSales*) el módulo obtiene todas las ventas según los parámetros del administrador (*GetEncryptedSales*).

##### BillboardDBManager

Este módulo se encarga de gestionar los datos y actualizaciones de la cartelera en el sistema.

Cuando un servidor se suscribe (*ConnectBillboard*) permite al módulo indicarle cuando se producen actualizaciones.

Cuando un servidor necesita obtener la nueva cartelera (*LoadBilboard*) se conecta a la base de datos (*GetBillboard*) para obtener la última versión de la cartelera guardada.

De forma similar ocurre cuando un administrador accede a los datos de la cartelera (*GetRawBillboard*) pero en este caso lo envía en un formato editable.

Cuando el administrador añade o modifica algo de la cartelera (*UpdateBillboard*) se realiza la actualización de la base de datos (*SaveBillboard*) y se desencadena la actualización de los servidores.

##### GeneralDB

Es módulo es la base de datos general del sistema. En ella se almacena toda la información usada por el sistema.

Guarda (*SaveEncryptedUser)* y recupera (*GetEncryptedUser*) la información encriptada de los usuarios registrados y.

Guarda (SaveEncryptedSale) la información de las entradas vendidas y recupera todas las ventas según los parámetros buscados (*GetEncryptedSales*).

Almacena los comprobantes de una entrada vendida a través de la Web (*SaveEncryptedTicket*) y recupera el comprobante para cancelar la entrada (*GetEncryptedTicket*).

Guarda las modificaciones y añadidos a la cartelera hechas por un administrador (*SaveBillboard*) y recupera toda la cartelera de la base de datos (*GetBillboard*).

### Diagrama contextual

Esta vista es el desarrollo del subsistema *Database System.* (Ver 8.1.1)

### Guía de puntos de variación

*GeneralDB* puede estar separada en varias bases de datos, pero deberían estar interrelacionadas, puesto que las entradas vendidas a un usuario registrado deberían estar asociadas a él.

*BillboardDBManager* implementa la otra parte del patrón *Observer*  donde se suscriben los servidores para recibir actualizaciones (Ver 8.2.4) metiendo los *Listener* de los servidores en una lista que se recorrería para notificarles que deberían hacer la petición de actualizar la cartelera.

### Justificación de las decisiones

|  |  |  |
| --- | --- | --- |
| Escenario | Atributo de calidad | Justificación |
| 2 | **AT-02** | El módulo *BillboardDBManager* está pensando para que cuando se haga un cambio en la cartelera este cambio se transmita a los servidores y estos obtengan la última versión de la cartelera. |
| 7 | **AT-07** | Las ventas y los usuarios son encriptados para proteger los datos según las leyes de protección de datos de manera que solo los elementos internos del sistema sepan cómo se encripten/desencripten. |
| 10 | **AT-10** | Cuando se realiza una actualización de la cartelera de parte del administrador, esta se cambia en la base de datos rápidamente. |

## PaymentGateway Adapter View

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Vista | Propiedades |
| IGatewayPayment | Encapsula los métodos de pago de las pasarelas de pago de manera que solo se comunique mediante un método de pago. |
| BankGatewayPayment | Clase que contiene los métodos específicos de pasarela de pago y que lleva a cabo el pago propiamente dicho a través del método especificado por su interfaz. |
| IDoPayment | Encapsula el pago de la entrada vendida. |
| DoPayment | Esta clase en la implementación del pago para el tipo de pasarela de pago que ofrezca el banco. |
| Sale | Contiene toda la información relacionada con la venta, como puede el número de tarjeta de crédito, cantidad a pagar… |

#### Relaciones

N/A

#### Interfaces/Puertos

N/A

#### Comportamiento

Ver *Propiedades* en 8.6.2.1

### Diagrama contextual



### Guía de puntos de variación

La pasarela de pago es ofrecida por el banco, por lo que es conveniente mantener separada la pasarela de todo el sistema, de manera que si esta cambia pueda modificarse adaptándose a la interfaz DoPayment y que permita conectarse a la interfaz del gateway, a la cual se adapta el gateway específico del banco.

### Justificación de las decisiones

|  |  |  |
| --- | --- | --- |
| Escenario | Atributo de calidad | Justificación |
| 6 | **AT-06** | El adaptador permite llevar a cabo una transacción segura con el banco y también permite modificar la pasarela de pago en caso de que esta sea necesaria cambiarla sin complicar la implementación. |