|  |
| --- |
| C:\Users\Usuario\Desktop\1.png |
| Entrecine |
| Arquitectura del Software |
| **08/04/2013** |
|  |
|  |



Autores:

Jiménez Franco, Marta

Antomil Valledor, Benjamín

Barrantes Montes, Diego

Duarte Jiménez, Carlos Andrés

Gallego García, María

García Baldomir, Daniel

|  |
| --- |
|  |

Contenido

1. Especificación inicial

El sistema debe gestionar automáticamente tanto la venta de entradas a través de Internet como de las que se vendan en las taquillas del propio cine. Adicionalmente, la aplicación será utilizada por el administrador para configurar sus parámetros.

1. Metodología usada

Se va a realizar un estudio de arquitectura siguiendo el método de ADD[[1]](#footnote-2) (Atribute-Driven Design) y la norma del SEI (ANSI/IEEE 1471, 2000).

1. Identificación de Stakeholders

Se va a nombrar este proyecto como “Entrecine” (EC), de manera que se pueda centrar el contexto de referencias e interesados en el proyecto.

● Usuario página web

● Personal EC

● Administrador EC

● Pasarela de pago (pasivo)

● Equipo de desarrollo EC

● Gestores (dueños) del cine

● Equipo de mantenimiento EC

* 1. Usuario página web

Es el usuario (cliente) que realiza la compra de entradas desde la página.

Su objetivo será:

● Poder adquirir una entrada para la película y sesión que quiera y que esté disponible de forma no presencial.

● Poder ver las butacas disponibles (número de fila y asiento).

● Poder ver las películas que están en cartelera (película, horario, sinopsis).

* 1. Administrador EC

Encargado de la modificación de las películas y las sesiones de las mismas. Configura los parámetros de cada una.

Objetivos:

● Facilidad de manipulación de la base de datos, a poder ser a través de un interfaz gráfico (películas, horarios, precio de los tipos de las sesiones).

● Seguridad de que sólo él pueda acceder a la BD para verla y modificarla.

* 1. Equipo de mantenimiento EC

Equipo encargado del mantenimiento de la aplicación tras su desarrollo completo.

Objetivos:

● Tener una aplicación estable.

● Mantener la página web actualizada, tener los fallos de seguridad controlados.

● Contar con una documentación clara y completa.

* 1. Personal EC

Generalización de los empleados del cine, ya sean los encargados de las taquillas, mantenimiento, etc.

● En taquilla.

○ Encargados de la venta.

○ Verificación de los tickets obtenidos desde la página web.

● Que la aplicación funcione bien.

● La aplicación identifique correctamente los precios de cada sesión automáticamente.

* 1. Pasarela de Pago

Intermediario entre el cine y el banco. Recibe la información de cobro del cliente para efectuar el pago. Se pagará mediante tarjeta de crédito.

No tiene objetivos como tales (entidad pasiva) pero sí que habrá que asegurar una correcta comunicación entre la aplicación web y la pasarela de pago.

* 1. Equipo de desarrollo EC

Este equipo será el responsable de desarrollar el sistema resultante de la arquitectura.

Entre sus objetivos están:

● Proyecto rentable, esto es, que permite ser desarrollado por el precio establecido.

● Que el rendimiento sea óptimo (el esperado) tanto para uso normal como en momentos de una alta carga de trabajo.

● Que sea modificable en todo momento.

* 1. Gestor (dueño) del cine

Responsable del presupuesto y de las decisiones que comprometan dichos fondos.

Objetivo:

● Que el producto final cumpla correctamente con su función dentro del plazo de entrega.

1. Identificación de los atributos de calidad

Para el sistema se han identificado los siguientes atributos de calidad:

* 1. Disponibilidad

1. La página web debe estar siempre disponible 24/7
2. La aplicación de escritorio debe estar disponible durante el horario de apertura del cine
   1. Rendimiento
3. Que el servidor resista la carga de trabajo en las horas punta
4. Debe manejar correctamente la concurrencia a la hora de comprar.
   1. Usabilidad
5. El registro de la reserva en la página web sea intuitivo para el usuario, la página debe ser amigable.
6. Para el personal del cine también.
   1. Seguridad
7. Conexión segura para acceder a la pasarela de pago.
8. Proteger los datos de los usuarios registrados y en el propio registro de usuario.
   1. Modificabilidad
9. Posibles cambios en la aplicación.
10. Debe poder modificarse la cartelera así como sus atributos.
11. Deben poder modificarse los precios.
    1. Testabilidad
12. Los registrados en la web concuerden con el comprobante de pago recibido por el usuario.
13. Comprobar la reserva de butacas (acceso concurrente y registro de usuarios).
14. Debe funcionar en los equipos de los que disponga el cliente o en los que se piensen adquirir para la informatización.
15. Debe funcionar correctamente en las versiones actuales de al menos los navegadores más comunes.
    1. Escalabilidad
16. Es posible que en el futuro sea necesario añadir nuevos sistemas.
    1. Accesibilidad
17. Que cumpla con los estándares web.
18. Primer acercamiento de la solución

Con los requisitos iniciales planteados, se llega a una idea inicial de cómo tratar el problema. La solución planteada consiste en usar un servidor web, disponible 24/7, que se conecte a una base de datos para obtener la información sobre las películas y las sesiones del cine.

Dicha base de datos estará administrada a través de una aplicación de escritorio sólo accesible por el administrador.

Una vez accedido a los datos, el servidor proporcionará un sitio web desde el que trabajarán tanto los empleados del cine (desde una cuenta especial) como los clientes que compren entradas desde su casa. Además, contará con conexión, mediante pasarela de pago, con el banco designado por la gestión del sistema.

* 1. Riesgos relacionados con la solución

El sistema presenta algunos riesgos que es preciso identificar:

1. Los equipos usados por los empleados pueden sufrir de un bajo rendimiento al intentar acceder en las horas punta a las funciones necesarias desde la página web.

2. Dependencia de una buena y estable conexión de internet en la taquilla de venta.

1. Lista actualizada de los Stakeholders

La lista de interesados actualizada se detalla a continuación.

|  |  |  |
| --- | --- | --- |
| **Código** | **Stakeholders** | **Intereses** |
| **ST-01** | Usuario Pagina Web | Poder adquirir una entrada para la película y sesión que quiera y que esté disponible de forma no presencial.  Poder ver las butacas disponibles (número de fila y asiento).  Poder ver las películas que están en cartelera (peli, horario, sinopsis). |
| **ST-02** | Personal | Que la aplicación funcione bien.  La aplicación identifique correctamente los precios de cada sesión automáticamente. |
| **ST-03** | Administrador | Facilidad de manipulación de la base de datos, a poder ser a través de un interfaz gráfico (películas, horarios, precio de los tipos de las sesiones).  Seguridad de que sólo él pueda acceder a la BD para verla y modificarla. |
| **ST-04** | Pasarela de pago (pasivo) | Correcta comunicación entre la aplicación web y la pasarela de pago. |
| **ST-05** | Equipo de desarrollo | Proyecto rentable, esto es, que permite ser desarrollado por el precio establecido.  Que el rendimiento sea óptimo (el esperado) tanto para uso normal como en momentos de una alta carga de trabajo.  Que sea modificable en todo momento. |
| **ST-06** | Gestores (dueños) del cine | Que el producto final cumpla correctamente con su función dentro del plazo de entrega. |
| **ST-07** | Mantenimiento de la aplicación | Tener una aplicación estable.  Mantener la página web actualizada, tener los fallos de seguridad controlados.  Contar con una documentación clara y completa. |

1. Lista actualizada de atributos de calidad

La lista de atributos de calidad queda:

|  |  |  |
| --- | --- | --- |
| **Código** | **Descripción** | **Tipo de atributo** |
| **ATT001** | La página web debe estar siempre disponible 24/7 | Disponibilidad |
| **ATT002** | La aplicación de escritorio debe estar disponible durante el horario de apertura del cine. | Disponibilidad |
| **ATT003** | Que el servidor resista la carga de trabajo en las horas punta. | Rendimiento |
| **ATT004** | Debe manejar correctamente la concurrencia a la hora de comprar. | Rendimiento |
| **ATT005** | El registro de la reserva en la página web sea intuitivo para el usuario, la página debe ser amigable. | Usabilidad |
| **ATT006** | Aplicación local sea intuitiva y fácil de usar. | Usabilidad |
| **ATT007** | Conexión segura para acceder a la pasarela de pago. | Seguridad |
| **Código** | **Descripción** | **Tipo de atributo** |
| **ATT009** | Debe poder modificarse la cartelera así como sus atributos (sesiones, precios…). | Modificabilidad |
| **ATT010** | Los registrados en la web concuerden con el comprobante de pago recibido por el usuario. | Testabilidad |
| **ATT011** | Comprobar la reserva de butacas (acceso concurrente y registro de usuarios). | Testabilidad |
| **ATT012** | Debe funcionar en los equipos de los que disponga el cliente o en los que se piensen adquirir para la informatización. | Testabilidad |
| **ATT013** | Debe funcionar correctamente en las versiones actuales de al menos los navegadores más comunes. | Testabilidad |
| **ATT014** | Es posible que en el futuro sea necesario añadir nuevos sistemas. | Escalabilidad |
| **ATT015** | Que cumpla con los estándares web. | Accesibilidad |

1. Atributos de calidad y stakeholders

Los diferentes atributos de calidad señalados anteriormente se correlacionan con los diferentes interesados. La siguiente tabla muestra la lista de intereses para el proyecto actual:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Atributos vs Interesados** | **ST-01** | **ST-02** | **ST-03** | **ST-04** | **ST-05** | **ST-06** | **ST-07** |
| **ATT001** | X |  |  | X |  | X | X |
| **ATT002** |  | X | X |  |  |  |  |
| **ATT003** |  | X | X |  | X |  | X |
| **ATT004** | X | X |  |  |  |  |  |
| **ATT005** | X |  |  |  |  |  |  |
| **ATT006** |  | X | X |  |  |  |  |
| **ATT007** |  |  |  | X | X |  | X |
| **ATT008** | X |  |  |  |  |  |  |
| **ATT009** |  |  | X |  |  |  |  |
| **ATT010** |  | X |  |  |  |  |  |
| **ATT011** | X | X |  |  |  |  |  |
| **ATT012** |  |  |  |  | X |  | X |
| **ATT013** | X |  |  |  |  |  |  |
| **ATT014** |  | X | X |  |  | X |  |
| **ATT015** |  |  |  | X | X |  | X |

1. Descripción de negocio de la solución

La aplicación a desarrollar constará de dos partes que interactúan con la base de datos, el servidor web que será accedido por los clientes y la aplicación local utilizada por los empleados del establecimiento y su respectivo administrador.

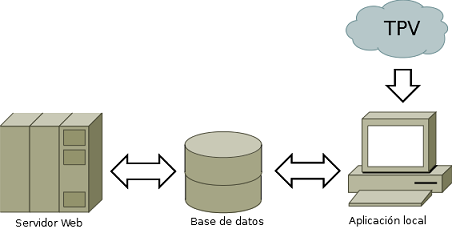


Figura – Modelo de negocio

* 1. Servidor Web

Permitirá a los clientes ver la cartelera actual, ver las butacas disponibles en el momento y reservarlas para las sesiones que deseen, además de que puedan registrarse en el sistema y así poder hacer sus compras con mayor rapidez al tener sus datos básicos guardados.

* 1. Aplicación local

Usada principalmente por el administrador y los empleados de la taquilla de venta, donde se modificará las sesiones, precios y las películas alojadas en la cartelera de la página web, también la aceptación de los comprobantes de pago obtenidos vía web y venta de los tickets de entrada a las salas de cine, acciones realizadas respectivamente por los actores interesados.

1. Escenarios de calidad

Con toda la información anterior se procederá a definir los escenarios de calidad que influencian esta arquitectura.

En las próximas páginas se muestra una tabla con la lista de escenarios identificados.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Escenario** | **Fuente de Estímulo** | **Estimulo** | **Entorno** | **Artefacto** | **Respuesta** | **Medición de la Respuesta** | **Atributo de calidad del Artefacto** |
| **1** | Usuario web | Servicio web | Explotación | Servidor web | Atención 24/7 | Denegación de servicio <1% | ATT001 |
| **2** | Personal del cine | Acceso a la aplicación | Explotación | Aplicación local | Funcionamiento correcto y disponible | Sistema 100% funcional | ATT002 |
| **3** | Usuario | Compra ticket (web/directa) | Comunicación con el servidor | Servidor web / aplicación local | Reserva y registro de la venta | Fallo en las operaciones de registro < 20% | ATT003 |
| **4** | Proceso de datos en el servidor | Petición de servicio | Explotación, desarrollo | Servidor | Tiempo efectivo de procesamiento | Tiempo efectivo de procesos > 99% | ATT003 |
| **5** | Usuario | Compra ticket (web/directa) | Explotación, desarrollo | Servidor | Prevención de riesgo de inconsistencias en concurrencia | Impedir transacción si la inconsistencia es >=1 | ATT004 |
| **6** | Usuario web | Registro | Explotación | Servicio web | Acceso fácil e intuitivo | Fallos de registro < 20% | ATT005 |
| **7** | Personal del cine | Acceso a la aplicación | Explotación | Aplicación local | Acceso fácil e intuitivo | Fallo en el correcto uso de la aplicación < 5% | ATT006 |
| **8** | Usuario web | Pago de tickets | Desarrollo | Servicio web | Acceso a conexión segura |  | ATT007 |
| **9** | Acceso a datos | Usuario web | Explotación | Servidor web | Conexión segura frente al acceso a los datos privados |  | ATT008 |
| **10** | Acceso a la base de datos | Modificación básica de información | Explotación | Aplicación local | Modificación correcta de la información implicada |  | ATT009 |
| **11** | Información correcta | Pago de ticket vía web | Desarrollo | Servidor web | Recibo de pago concuerde con la información del usuario implicado |  | ATT010 |
| **12** | Comprobación de recibo de pago | Cambio de comprobante por un ticket de entrada | Explotación, desarrollo | Aplicación local | Verificación de que la información mostrada concuerde con el registro del mismo |  | ATT011 |
| **13** | Soporte hardware | Informatización | Desarrollo | Equipos informáticos | Respuesta positiva a la aplicación implementada |  | ATT012 |
| **14** | Usuario web | Página web | Explotación | Servidor web | Funcionamiento correcto |  | ATT013 |
| **15** | Gestores / Mantenimiento | Incremento del presupuesto | Desarrollo | Servidor web / aplicación local | Mejoras o nuevos servicios |  | ATT014 |
| **16** | Usuario web | Página web | Explotación | Servidor web | Funcionamiento correcto y accesible |  | ATT015 |

1. Vistas
   1. System View

Esta vista describe los subsistemas en interacción: Local App, Database y Server Web.



Figura – Vista general del sistema

* + 1. Database

Representa la base de datos donde se almacenará la información necesaria para gestionar el cine y sus usuarios.

* + - 1. *Embedded Elements*

| **Element** | **Notes** |
| --- | --- |
| **ProvidedInterface checkAvailability** | Recibe la petición de consulta de la disponibilidad de cierta película |
| **ProvidedInterface checkTicket** | Recibe la petición de validación de entradas compradas por internet |
| **ProvidedInterface getAvailability** | Recibe la petición de consulta de las localidades disponibles para una determinada película |
| **ProvidedInterface getBillboard** | Recibe la petición de consulta de cartelera para cierta fecha |
| **ProvidedInterface getCustomer** | Recibe la petición de verificación de usuario |
| **ProvidedInterface getUser** | Recibe la petición de consultar la lista de usuarios |
| **ProvidedInterface newCustomer** | Recibe la petición de añadir un nuevo usuario |
| **ProvidedInterface newReservation** | Recibe la petición de añadir una nueva reserva de una localidad |
| **ProvidedInterface sellTicket** | Recibe la información de la última venta realizada. |
| **ProvidedInterface setCustomer** | Recibe los datos para modificar los datos personales de un usuario |
| **ProvidedInterface Update** | Recibe la solicitud de actualización |

* + 1. Local App

Representa la aplicación de escritorio del cine.

* + - 1. *Embedded Elements*

| **Element** |  | **Notes** |
| --- | --- | --- |
| **ProvidedInterface Start - Stop** |  | Pone en marcha la aplicación de escritorio |
| **RequiredInterface checkAvailability** |  | Lanza la petición para recibir los datos sobre asientos libres de la BD |
| **RequiredInterface checkTicket** |  | Lanza la petición de validación de entradas compradas por internet a la base de datos |
| **RequiredInterface getUser** |  | Recupera la información de usuarios de la BD |
| **RequiredInterface sellTicket** |  | Manda a la BD la información de la última venta realizada. |
| **RequiredInterface Update** |  | Envia la información de actualización de la BD |

* + 1. Server Web

Representa el servidor Web donde se podrán reservar entradas online y ver las disponibilidades de las películas ofertadas.

* + - 1. *Embedded Elements*

| **Element** | **Notes** |
| --- | --- |
| **RequiredInterface getAvailability** | Consulta las localidades disponibles |
| **RequiredInterface getBillboard** | Consulta la cartelera del cine |
| **RequiredInterface getCustomer** | Verifica si un usuario es válido |
| **RequiredInterface newCustomer** | Envía los datos para añadir los datos personales de un nuevo usuario |
| **RequiredInterface newReservation** | Envía los datos correspondientes a la reserva de una localidad |
| **RequiredInterface setCustomer** | Envía los datos para modificar los datos personales de un usuario |
| **ProvidedInterface Start-Stop** | Pone en marcha el servidor web |
|  |  |

* 1. DataBase View

La vista detalla los componentes de la base de datos.

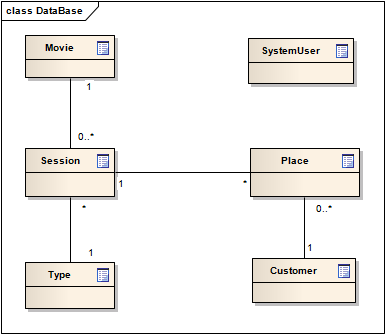


Figura – Vista sobre la base de datos

* + 1. Customer
* *Database*:Entrecine
* *Detail*: Guarda la información personal de los clientes registrados y sus respectiva cuenta bancaria
* *Notes*:
  + - 1. Relationships

|  |  |
| --- | --- |
| **Association** | **Notes** |
| **1 Customer.**  **0..\* Place.** | Representa las compras realizadas por los clientes, esta puede efectuarse localmente o vía web |

* + 1. Place
* *Database:* Entrecine
* *Detail*: Tabla que lleva el registro de todas las ventas realizadas por la TPV local y web.
* *Notes*: En caso de que sea una venta vía web deberá llevar el cliente con su respectivo número de tarjeta usada en ese momento.
  + - 1. Relationships

|  |  |
| --- | --- |
| **Association** | **Notes** |
| **1 Session.**  **\* Place.** | Localidades reservadas para una sesión específica |
| **1 Customer.**  **0..\* Place.** | Reserva hecha por un cliente desde la página web |

* + 1. Movie
* *Database:* Entrecine
* *Detail*: Guarda toda la información relevante de las películas registradas en el sistema.
* *Notes*:
  + - 1. Relationships

|  |  |
| --- | --- |
| **Association** | **Notes** |
| **1 Pelicula.**  **0..\* Session.** | Sesión o sesiones a las que fue asignada una película |

* + 1. Session
* *Database:* Entrecine
* *Detail*: Son puntos en los que está prevista el comienzo de las películas.
* *Notes*: Lleva el control del número de butacas disponibles.
  + - 1. Relationships

|  |  |
| --- | --- |
| **Association** | **Notes** |
| **1 Type.**  **\* Session.** | El tipo y horario que representa la sesión |
| **1 Movie.**  **0..1 Session.** | Película que se emitirá en la sesión |
| **1 Session.**  **\* Place.** | Reservas asociadas a la sesión |

* + 1. Type
* *Database:* Entrecine
* *Detail*: Establece el horario y precio de las sesiones que estén activas
* *Notes*:
  + - 1. Relationships

|  |  |
| --- | --- |
| **Association** | **Notes** |
| **1 Type.**  **\* Session.** | Establece un tipo y horario a las sesiones |

* + 1. SystemUser
* *Database:* Entrecine
* *Detail*: Almacena a los usuarios de LocalApp, tanto a los usuario del TPV como el administrador.
* *Notes*:
  1. Local App View

La vista describe los procesos de la aplicación de escritorio del EC.

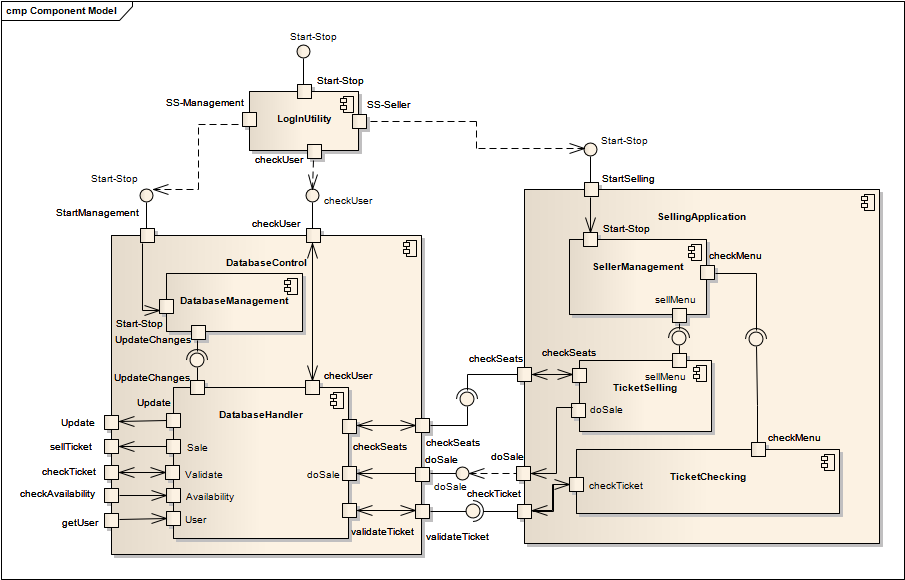


Figura – Vista sobre la aplicación local

* + 1. DatabaseControl

Componente que contiene los programas que interactúan con la base de datos.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkAvailability** | Lanza la petición para recibir los datos sobre asientos libres de la BD |
| **Port checkSeats** | Recibe la petición de buscar asientos libres por parte de la zona de ventas. |
| **Port checkTicket** | Lanza la petición de validación de entradas compradas por internet a la base de datos**.** |
| **Port checkUser** | Recibe una petición de validar usuarios de la aplicación a través de la BD. |
| **Port doSale** | Recibe la información que se ha de pasar a BDsobre las entradas vendidas. |
| **Port getUser** | Recupera la información de usuarios de la BD. |
| **Port sellTicket** | Manda a la BD la información de la última venta realizada. |
| **Port StartManagement** | Recibe la petición de comenzar a administrar la base de datos |
| **Port Update** | Actualiza la BD con la nueva información |
| **Port validateTicket** | Recibe la petición de validar entradas compradas por internet. |

* + 1. DatabaseHandler

Programa encargado de realizar las interacciones directas con la base de datos.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkSeats** | Manda a base de datos una petición para comprobar que asientos están ocupados |
| **Element** | **Notes** |
| **Port doSale** | Recibe los datos de la venta realizada |
| **Port Sale** | Vuelca los datos de la venta a BD |
| **Port Update** | Actualiza la BD con la nueva información |
| **Port UpdateChanges** | Recibe la petición de actualización de la BD |
| **Port User** | Recibe la información de todos los usuarios registrados en el sistema |
| **Port validateTicket** | Recibe la petición de validar una entrada comprada por internet |
| **Port validateTicket** | Recibe de BD la información relacionada con las entradas vendidas por internet |
| **Port Availability** | Recupera los asientos disponibles de BD |

* + 1. DatabaseManagement

Componente utilizado por el administrador, se encarga de realizar las actualizaciones de datos.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port Start-Stop** | Recibe la petición de iniciar el programa por parte del login. |
| **Port UpdateChanges** | Avisa al database handler de que tiene que volcar la información a la BD |

* + 1. LoginUtility

Pantalla de log para controlar si entran empleados de taquilla o el administrador.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkUser** | Pide a BD que compruebe si el usuario es correcto. |
| **Port SS-Management** | Implementará las acciones que inicien sesión al administrador. |
| **Port SS-Seller** | Llamará al Start del módulo que gestiona la venta de entradas. |
| **Port Start-Stop** | Recibe la petición de inicio de la aplicación. |

* + 1. SellingApplication

Encapsula todas las acciones que se relacionan con vender entradas de cine por parte de los taquilleros.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkSeats** | Manda una petición para comprobar qué asientos están libres. |
| **Port checkTicket** | Manda una petición para validar una entrada comprada online. |
| **Port doSale** | Envía los datos de una venta a la BD |
| **Port StartSelling** | Recibe la petición de comenzar la aplicación de venta de entradas desde el login. |

* + 1. SellerManagement

Menú que trata las opciones a las que puede optar un vendedor en taquilla.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkMenu** | Manda una petición para comprobar entradas compradas por internet. |
| **Port sellMenu** | Manda una petición de vender una o más entradas |
| **Port Start-Stop** | Recibe la petición de inicio del login. |

* + 1. TicketChecking

Encapsula las acciones que comprueban si las entradas compradas por internet son verdaderas.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkMenu** | Recibe la llamada para comprobar una entrada comprada por internet. |
| **Port checkTicket** | Llama a base de datos para saber si el ticket es válido |

* + 1. TicketSelling

Encapsula las acciones necesarias para vender una entrada.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port checkSeats** | Pide a BD los asientos disponibles para la sesión. |
| **Port doSale** | Concluye la venta y manda a BD los datos |
| **Port sellMenu** | Recibe la petición de iniciar una venta en taquilla. |

* 1. Server Web View

La vista describe los procesos del servidor Web del EC.

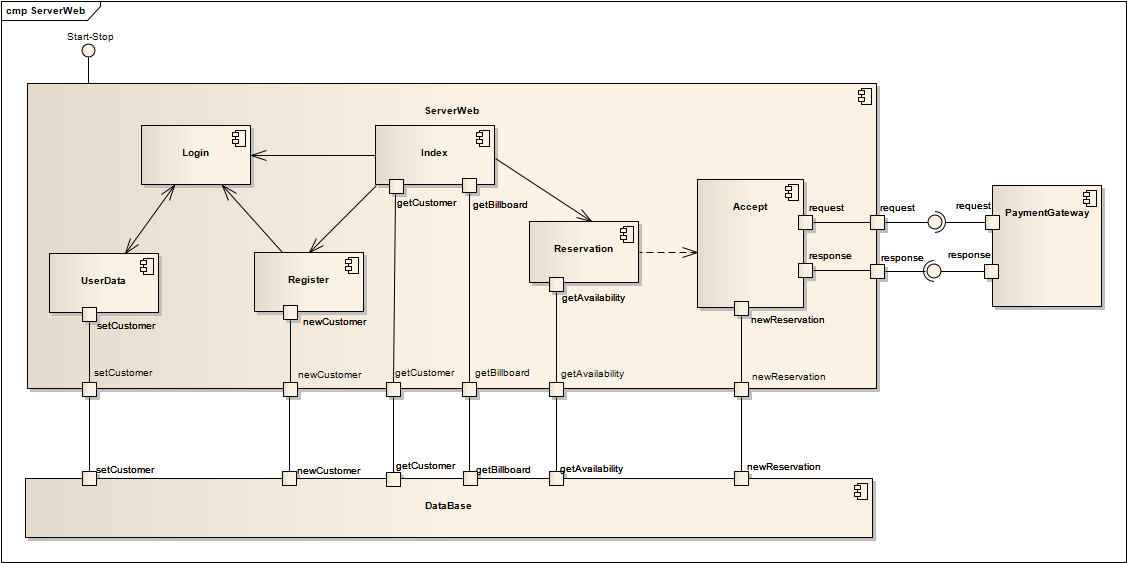


Figura – Vista sobre el servidor web

* + 1. PaymentGateway

Pasarela de pago contratada la cual es ajena a la aplicación. Recibe la información bancaria del cliente y envía a la aplicación si la operación se realizó con éxito o no.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port request** | Recibe una petición de pago |
| **Port response** | Retorna si la operación fue realizada con éxito |

* + 1. ServerWeb

Servidor que estará disponible 24/7 y es el encargado de gestionar las transacciones web.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port getAvailability** | Consulta las localidades disponibles |
| **Port getBillboard** | Consulta la cartelera del cine |
| **Port getCustomer** | Verifica si un usuario es válido |
| **Port newCustomer** | Añade un usuario a la base de datos |
| **Port newReservation** | Guarda la reserva de una localidad |
| **Port request** | Envía la petición para pagar una reserva |
| **Port response** | Si el pago se realizó con éxito |
| **Port setCustomer** | Modifica los datos personales de un usuario |
| **ProvidedInterface Start-Stop** | Pone en marcha el servidor web |

* + 1. Accept

Encargada de la comunicación con la pasarela de pago contratada y modificación de la base de datos.

* + - 1. Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Dependency**  **Source -> Destination** | Public  Reservation | Public  Accept | Recibe la información de las localidades que se desean reservar para su proceso respectivo |

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port newReservation** | Guarda una reservación al ser validado el pago |
| **Port request** | Envía el importe a la pasarela de pago |
| **Port response** | Verifica si la transacción fue realizada |

* + 1. DataBase

Base de datos local con la que se interactúa.

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port getAvailability** | Devuelve el número de localidades disponibles |
| **Port getBillboard** | Devuelve las sesiones de la cartelera actual del cine |
| **Port getCustomer** | Devuelve el usuario si la verificación es válida |
| **Port newCustomer** | Guarda el nuevo usuario en la base de datos |
| **Port newReservation** | Guarda la nueva reserva de las localidades seleccionadas |
| **Element** | **Notes** |
| **Port setCustomer** | Modifica los datos de un usuario |

* + 1. Index

Página de inicio de la plataforma web, muestra la cartelera del cine y comprueba el inicio de sesión de los usuarios registrados.

* + - 1. Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Login | Una vez que fue comprobado el usuario valido, permite el acceso a su espacio personal |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Register | Accede a la página que permite registrarse como usuario |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Reservation | Accede a la página que permite reservar las localidades |

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port getBillboard** | Muestra la cartelera previamente pedida |
| **Port getCustomer** | Envía los datos de usuario para comprobar si es válido |

* + 1. Login

Espacio personal de los usuarios registrados.

* + - 1. Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Login | Recibe un usuario ya validado previamente |
| **Delegate**  **Bi-Directional** | Public  Login | Public  UserData | Accede a un espacio para modificar sus datos personales y bancarios |
| **Delegate**  **Source -> Destination** | Public  Register | Public  Login | Acceso al espacio personal luego de haberse registrado en el sistema |

* + 1. Register

Permite registrar los nuevos clientes en el sistema, esto permitirá agilizar las reservas.

* + - 1. Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Register | Espacio para registrarse en el sistema y guardar sus datos bancarios personales |
| **Delegate**  **Source -> Destination** | Public  Register | Public  Login | Al terminar el registro se permite acceder a su espacio personal. |

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port newCustomer** | Envía la información con los datos del nuevo usuario registrado |

* + 1. Reservation

Lugar donde se reservan las localidades de una sesión ya previamente escogida.

* + - 1. Connections

|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Dependency**  **Source -> Destination** | Public  Reservation | Public  Accept | Paso al pago de la reservación escogida. |
| **Delegate**  **Source -> Destination** | Public  Index | Public  Reservation | Acceso desde la página inicial para hacer una reservación definida. |

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port getAvailability** | Verifica si hay localidades disponibles para la sesión seleccionada |

* + 1. UserData

Lugar en el que se permite al usuario cambiar su información personal o bancaria.

* + - 1. Connections

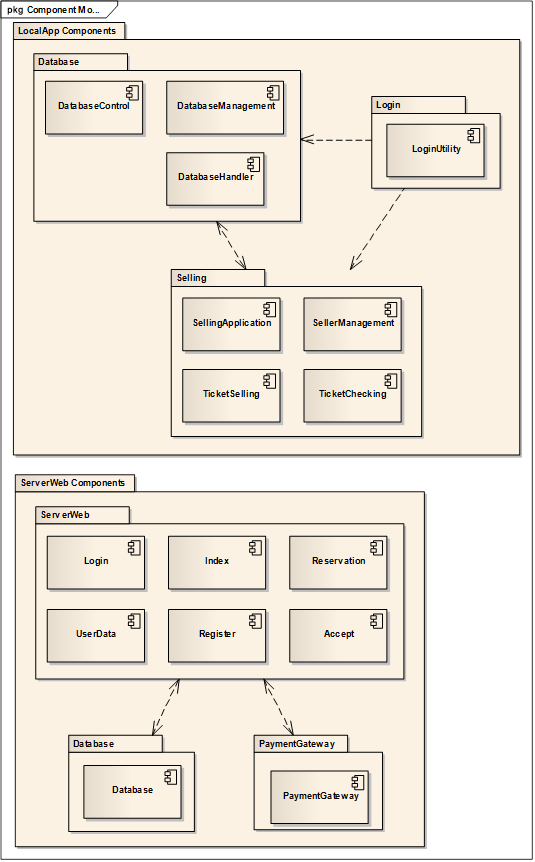
|  |  |  |  |
| --- | --- | --- | --- |
| **Connector** | **Source** | **Target** | **Notes** |
| **Delegate**  **Bi-Directional** | Public  Login | Public  UserData | Solamente accedida desde el espacio de un usuario registrado |

* + - 1. Embedded Elements

|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port setCustomer** | Envía los datos modificados para que sean guardados |

* 1. Package model view

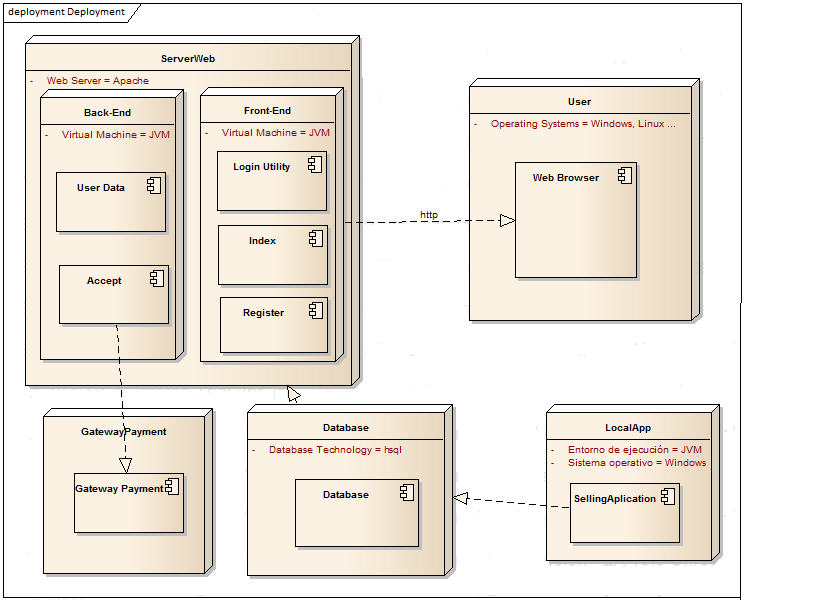
A continuación de muestra la vista de paquete muestra información importante para el equipo de desarrollo del sistema: el empaquetamiento de los componentes.



* 1. Deployment view

La vista de Deployment ofrece una vista de los módulos en ejecución y sobre los equipos

previstos.



|  |  |
| --- | --- |
| **Element** | **Notes** |
| **LocalApp** | Ordenador doméstico.   * Windows 8 * JVM |
| **ServerWeb** | Servidor Apache |
| **Database** | Las bases de datos irán sobre máquinas reales en el back-end del sistema físico. |
| **User** | Ordenador doméstico. Requisitos mínimos:   * Sistema Operativo: Windows, Linux o Mac * Navegador web. |
| **Front-End** | Estará sobre máquinas virtuales JVM |
| **Back-End** | Estará sobre máquinas virtuales JVM. Sobre Windows y hsql |

1. **Bass, Len, Clements, Paul y Kazman, Rick. 2003.** *Software Architecture in Practice, Second Edition.* Boston : Addison Wesley, 2003. ISBN: 0-321-15495-9. [↑](#footnote-ref-2)