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



Autores:

Antomil Valledor, Benjamín

Barrantes Montes, Diego

Duarte Jiménez, Carlos Andrés

García Baldomir, Daniel

|  |
| --- |
|  |

Contenido

[1 Especificación inicial 4](#_Toc353201208)

[2 Metodología usada 4](#_Toc353201209)

[3 Identificación de Stakeholders 4](#_Toc353201210)

[3.1 Usuario página web 4](#_Toc353201211)

[3.2 Administrador EC 4](#_Toc353201212)

[3.3 Equipo de mantenimiento EC 4](#_Toc353201213)

[3.4 Personal EC 5](#_Toc353201214)

[3.5 Pasarela de Pago 5](#_Toc353201215)

[3.6 Equipo de desarrollo EC 5](#_Toc353201216)

[3.7 Gestor (dueño) del cine 5](#_Toc353201217)

[4 Identificación de los atributos de calidad 5](#_Toc353201218)

[4.1 Disponibilidad 5](#_Toc353201219)

[4.2 Rendimiento 5](#_Toc353201220)

[4.3 Usabilidad 5](#_Toc353201221)

[4.4 Seguridad 5](#_Toc353201222)

[4.5 Modificabilidad 6](#_Toc353201223)

[4.6 Testabilidad 6](#_Toc353201224)

[4.7 Escalabilidad 6](#_Toc353201225)

[4.8 Accesibilidad 6](#_Toc353201226)

[5 Primer acercamiento de la solución 6](#_Toc353201227)

[5.1 Riesgos relacionados con la solución 6](#_Toc353201228)

[6 Lista actualizada de los Stakeholders 6](#_Toc353201229)

[7 Lista actualizada de atributos de calidad 7](#_Toc353201230)

[8 Atributos de calidad y stakeholders 8](#_Toc353201231)

[9 Descripción de negocio de la solución 8](#_Toc353201232)

[9.1 Servidor Web 9](#_Toc353201233)

[9.2 Aplicación local 9](#_Toc353201234)

[10 Escenarios de calidad 9](#_Toc353201235)

[11 Vistas 12](#_Toc353201236)

[11.1 System View 12](#_Toc353201237)

[11.1.1 Database 12](#_Toc353201238)

[11.1.2 Local App 13](#_Toc353201239)

[11.1.3 Server Web 13](#_Toc353201240)

[11.2 DataBase View 14](#_Toc353201241)

[11.2.1 Customer 14](#_Toc353201242)

[11.2.2 Place 14](#_Toc353201243)

[11.2.3 Movie 15](#_Toc353201244)

[11.2.4 Session 15](#_Toc353201245)

[11.2.5 Type 15](#_Toc353201246)

[11.2.6 SystemUser 16](#_Toc353201247)

[11.3 Local App View 16](#_Toc353201248)

[11.3.1 DatabaseControl 17](#_Toc353201249)

[11.3.2 DatabaseHandler 17](#_Toc353201250)

[11.3.3 DatabaseManagement 18](#_Toc353201251)

[11.3.4 LoginUtility 18](#_Toc353201252)

[11.3.5 SellingApplication 19](#_Toc353201253)

[11.3.6 SellerManagement 19](#_Toc353201254)

[11.3.7 TicketChecking 19](#_Toc353201255)

[11.3.8 TicketSelling 20](#_Toc353201256)

[11.4 Server Web View 20](#_Toc353201257)

[11.4.1 PaymentGateway 20](#_Toc353201258)

[11.4.2 ServerWeb 21](#_Toc353201259)

[11.4.3 Accept 21](#_Toc353201260)

[11.4.4 DataBase 22](#_Toc353201261)

[11.4.5 Index 23](#_Toc353201262)

[11.4.6 Login 23](#_Toc353201263)

[11.4.7 Register 24](#_Toc353201264)

[11.4.8 Reservation 25](#_Toc353201265)

[11.4.9 UserData 25](#_Toc353201266)

[11.5 Package model view 26](#_Toc353201267)

[11.6 Deployment view 28](#_Toc353201268)

# 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.

# Metodología usada

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

# 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

## 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).

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

# Identificación de los atributos de calidad

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

## 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

## Rendimiento

1. Que el servidor resista la carga de trabajo en las horas punta
2. Debe manejar correctamente la concurrencia a la hora de comprar.

## Usabilidad

1. El registro de la reserva en la página web sea intuitivo para el usuario, la página debe ser amigable.
2. Para el personal del cine también.

## Seguridad

1. Conexión segura para acceder a la pasarela de pago.
2. Proteger los datos de los usuarios registrados y en el propio registro de usuario.

## Modificabilidad

1. Posibles cambios en la aplicación.
2. Debe poder modificarse la cartelera así como sus atributos.
3. Deben poder modificarse los precios.

## Testabilidad

1. Los registrados en la web concuerden con el comprobante de pago recibido por el usuario.
2. Comprobar la reserva de butacas (acceso concurrente y registro de usuarios).
3. Debe funcionar en los equipos de los que disponga el cliente o en los que se piensen adquirir para la informatización.
4. Debe funcionar correctamente en las versiones actuales de al menos los navegadores más comunes.

## Escalabilidad

1. Es posible que en el futuro sea necesario añadir nuevos sistemas.

## Accesibilidad

1. Que cumpla con los estándares web.

# 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.

## 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.

# 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. |

# 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 |

# 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 |

# 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.

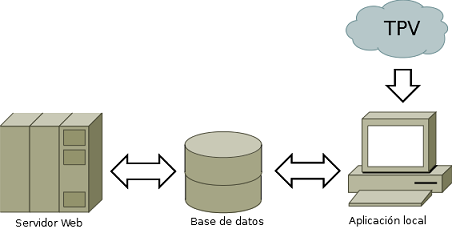


Figura 1 – Modelo de negocio

## 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.

## 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.

# 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 |

# 

# Vistas

## System View

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



Figura 2 – Vista general del sistema

### Database

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

#### *EmbeddedElements*

| **Element** | **Notes** |
| --- | --- |
| **ProvidedInterfacecheckAvailability** | Recibe la petición de consulta de la disponibilidad de cierta película |
| **ProvidedInterfacecheckTicket** | Recibe la petición de validación de entradas compradas por internet |
| **ProvidedInterfacegetAvailability** | Recibe la petición de consulta de las localidades disponibles para una determinada película |
| **ProvidedInterfacegetBillboard** | Recibe la petición de consulta de cartelera para cierta fecha |
| **ProvidedInterfacegetCustomer** | Recibe la petición de verificación de usuario |
| **ProvidedInterfacegetUser** | Recibe la petición de consultar la lista de usuarios |
| **ProvidedInterfacenewCustomer** | Recibe la petición de añadir un nuevo usuario |
| **ProvidedInterfacenewReservation** | Recibe la petición de añadir una nueva reserva de una localidad |
| **ProvidedInterfacesellTicket** | Recibe la información de la última venta realizada. |
| **ProvidedInterfacesetCustomer** | Recibe los datos para modificar los datos personales de un usuario |
| **ProvidedInterfaceUpdate** | Recibe la solicitud de actualización |

### Local App

Representa la aplicación de escritorio del cine.

#### *EmbeddedElements*

| **Element** |  | **Notes** |
| --- | --- | --- |
| **ProvidedInterfaceStart - Stop** |  | Pone en marcha la aplicación de escritorio |
| **RequiredInterfacecheckAvailability** |  | Lanza la petición para recibir los datos sobre asientos libres de la BD |
| **RequiredInterfacecheckTicket** |  | Lanza la petición de validación de entradas compradas por internet a la base de datos |
| **RequiredInterfacegetUser** |  | Recupera la información de usuarios de la BD |
| **RequiredInterfacesellTicket** |  | Manda a la BD la información de la última venta realizada. |
| **RequiredInterfaceUpdate** |  | Envia la información de actualización de la BD |

### Server Web

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

#### *EmbeddedElements*

| **Element** | **Notes** |
| --- | --- |
| **RequiredInterfacegetAvailability** | Consulta las localidades disponibles |
| **RequiredInterfacegetBillboard** | Consulta la cartelera del cine |
| **RequiredInterfacegetCustomer** | Verifica si un usuario es válido |
| **RequiredInterfacenewCustomer** | Envía los datos para añadir los datos personales de un nuevo usuario |
| **RequiredInterfacenewReservation** | Envía los datos correspondientes a la reserva de una localidad |
| **RequiredInterfacesetCustomer** | Envía los datos para modificar los datos personales de un usuario |
| **ProvidedInterfaceStart-Stop** | Pone en marcha el servidor web |
|  |  |

## DataBase View

La vista detalla los componentes de la base de datos.

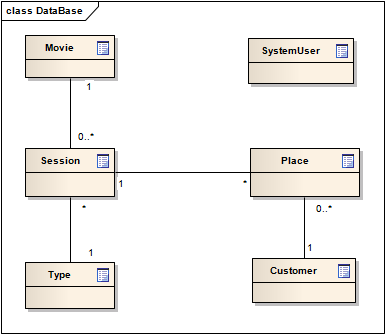


Figura – Vista sobre la base de datos

### Customer

* *Database*:Entrecine
* *Detail*: Guarda la información personal de los clientes registrados y sus respectiva cuenta bancaria
* *Notes*:

#### Relationships

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

### 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.

#### 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 |

### Movie

* *Database:* Entrecine
* *Detail*: Guarda toda la información relevante de las películas registradas en el sistema.
* *Notes*:

#### Relationships

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

### 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.

#### 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 |

### Type

* *Database:* Entrecine
* *Detail*: Establece el horario y precio de las sesiones que estén activas
* *Notes*:

#### Relationships

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

### SystemUser

* *Database:* Entrecine
* *Detail*: Almacena a los usuarios de LocalApp, tanto a los usuario del TPV como el administrador.
* *Notes*:

## Server Web View

La vista describe los procesos del servidor Web del EC.

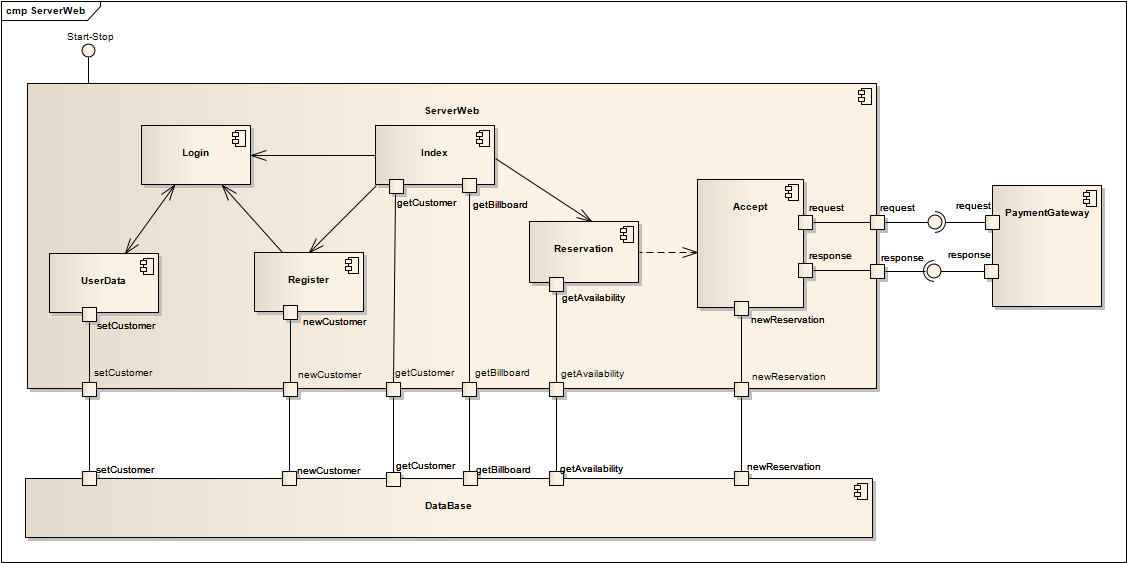


Figura – Vista sobre el servidor web

### 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.

#### Embedded Elements

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

### ServerWeb

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

#### 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 |

### Accept

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

#### 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 |

#### 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 |

### DataBase

Base de datos local con la que se interactúa.

#### 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 |

### Index

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

#### 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 |

#### Embedded Elements

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

### Login

Espacio personal de los usuarios registrados.

#### 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 |

### Register

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

#### 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. |

#### Embedded Elements

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

### Reservation

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

#### 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. |

#### Embedded Elements

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

### UserData

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

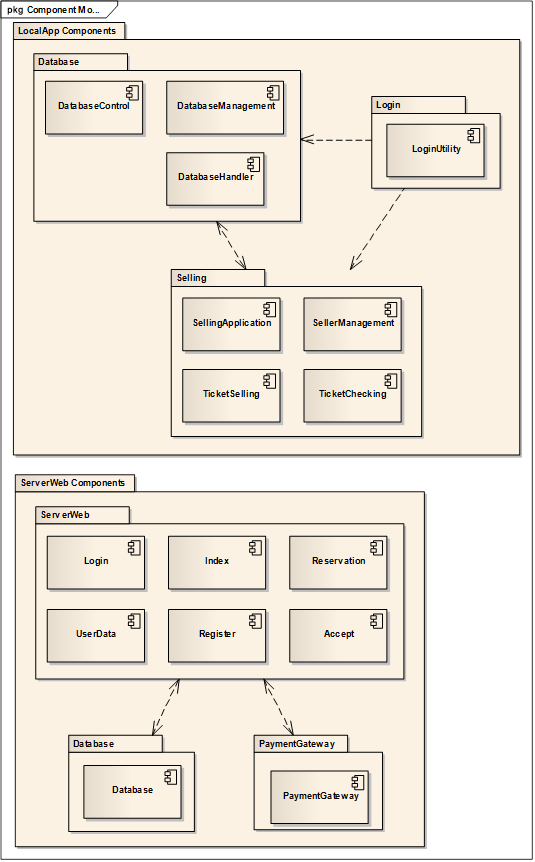
#### Connections

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

#### Embedded Elements

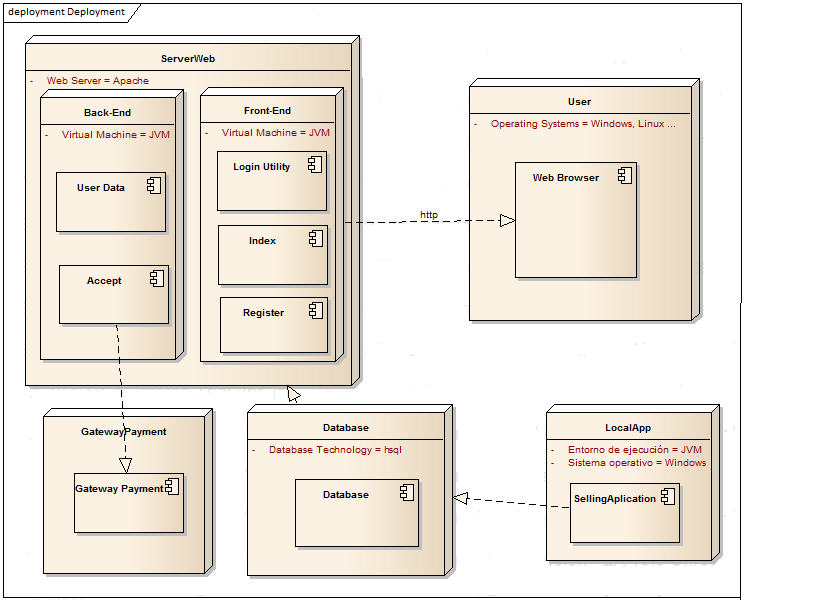
|  |  |
| --- | --- |
| **Element** | **Notes** |
| **Port setCustomer** | Envía los datos modificados para que sean guardados |

## Packagemodelview

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

## Deploymentview

La vista de Deploymentofrece 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 |

### Relations

Las comunicaciones entre el usuario y el servidor se realizan mediante http.

Las relaciones entre la base de datos y la aplicación local y el servidor serán mediante hsql.

### Interfaces/Puertos

N/A

### Compotamiento

#### User

Accede a través del navegador web.

#### ServerWeb

Soporta los componentes centrales de proceso.

Consta de Back-End y Front-End.

#### LocalApp

Soporta la aplicación de escritorio.

### Database

Da soporte a la persistencia de datos.

#### Front-End

Soporta las conexiones con los clientes.

#### Back-End

Hace de soporte para los módulos de gestión.

# Acercamiento final a la solución:

Tras haber planteado el sistema a la empresa Entrecine, se nos propone una serie de cambios en la realización de la aplicación. Aparte de lo que ya habíamos remarcado, se encarga un sistema de recomendaciones para los usuarios registrados. Además, la fecha de entrega se ve fijada, por lo que se añade el Time-to-market como requisito de la aplicación. Para ahorrar tiempo y costes de desarrollo se decide realizar todo el sistema a través de una única aplicación web.

## Riesgos asociados a la solución:

El sistema tiene los mismos riesgos que la especificación anterior, añadiendo además que la aplicación de gestión de la base de datos también sufre ahora el riesgo de que se pierda la conexión del servidor web.

# 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. |

# 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 |
| **Código** | **Descripción** | **Tipo de atributo** |
| **ATT006** | Aplicación local sea intuitiva y fácil de usar. | Usabilidad |
| **ATT007** | Conexión segura para acceder a la pasarela de pago. | Seguridad |
| **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 |
| **ATT016** | La aplicación debe servir recomendaciones a los usuarios registrados. | Usabilidad |
| **ATT017** | Time-to-Market | Disponibilidad |

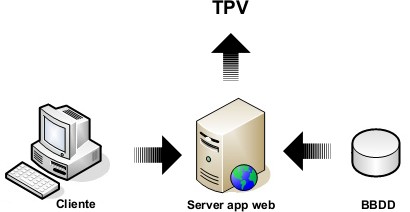
# Atributos de calidad y stakeholders

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 |
| **ATT016** | X |  | X |  |  | X | X |
| **ATT017** |  |  |  |  | X | X |  |

# Descripción del negocio de la solución:

La aplicación a desarrollar constará de dos bases de datos, una de ellas será la base de datos relacional, y la segunda se utilizará para gestionar las recomendaciones, además del servidor web que utilizarán tanto clientes como empleados.



## 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, lo que les permitirá recibir recomendaciones de películas y además les permitirá hacer sus compras con mayor rapidez al tener sus datos básicos guardados.

El administrador y los empleados de la taquilla de venta modificarán las sesiones, precios y las películas alojadas en la cartelera de la página web, también se encargarán de 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.

## Base de datos relacional

Guardará los datos de todo el sistema; tanto sesiones como las películas, así como usuarios del sistema y de la web. Será donde se realicen la mayoría de las peticiones de datos.

## Base de datos No-SQL

Guardará la información de películas y usuarios, y los relacionará para establecer un sistema de recomendaciones para la página web.

### Descripción de la realización de las recomendaciones en la web:

Para cumplir el nuevo requisito encargado por la organización Entrecine se ha decido utilizar una segunda base de datos No-SQL, en éste caso la base Neo4j, que utiliza esquema de grafo, por lo que permite obtener las recomendaciones en un tiempo muy superior a los que se podrían obtener en una BD relacional.

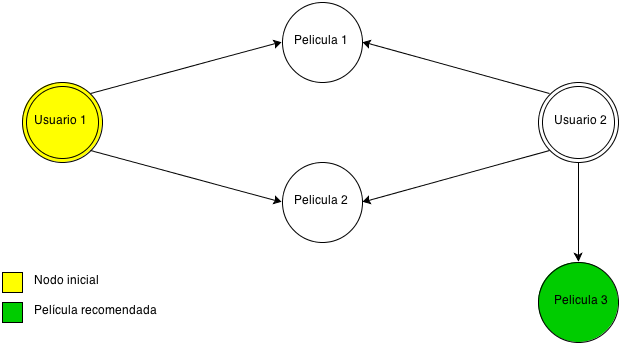
#### Llenar la BD con la información.

Para obtener las recomendaciones se almacenan en la base de datos la información de los usuarios registrados y las películas que ven a la hora de hacer la reserva.

#### Obtener las recomendaciones.

Siendo el nodo N del grafo el usuario registrado, y existiendo otro usuario M, que tiene en común al menos dos películas en común con N. Si M ha visto la película K, y N no la ha visto, K se recomendaría.

**Explicación gráfica:**

****

# 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 | Servidor web | Funcionamiento correcto y disponible | Sistema 100% funcional | ATT002 |
| **3** | Usuario | Compra ticket (web/directa) | Comunicación con el servidor | Servidor web | 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 | Servidor web | 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 | Servidor web | 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 | Servidor web | 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 | Mejoras o nuevos servicios |  | ATT014 |
| **16** | Usuario web | Página web | Explotación | Servidor web | Funcionamiento correcto y accesible |  | ATT015 |
| **17** | Usuario web | Página web | Explotación | Servidor web | Servir recomendaciones de películas a los usuarios registrados | Se devuelven películas que el usuario no ha visto y se corresponden con sus gustos. | ATT016 |

# Vistas:

## System View:

Ésta vista describe los subsistemas finales de la aplicación.



Figure: 1

### Database

Base de datos relacional que contendrá los datos del sistema.

*Embedded Elements*

| **Element** | **Notes** |
| --- | --- |
| ProvidedInterface checkTicket | Comprueba que la reserva online existe. |
| ProvidedInterface getAvailability | Comprueba los asientos disponibles en las distintas sesiones del cine |
| ProvidedInterface getBillboard | Recupera los datos de la cartelera del cine y sus sesiones. |
| ProvidedInterface getCostumer | Recupera los datos de un usuario registrado en BD |
| ProvidedInterface getSystemUser | Recupera a un usuario del sistema de BD |
| ProvidedInterface newCostumer | Crea un nuevo usuario en la BD |
| ProvidedInterface newReservation | Confirma una nueva reserva online de asientos. |
| ProvidedInterface sellTicket | Vende en taquilla una entrada |
| ProvidedInterface updateData | Actualiza información de la base de datos. |

### webApp

Aplicación web que contendrá toda la lógica y la presentación del proyecto.

*Embedded Elements*

| **Element** | **Notes** |
| --- | --- |
| ProvidedInterface Start-Stop | Comienza la aplicación |
| RequiredInterface checkTicket | Comprueba que la reserva online existe. |
| RequiredInterface getAvailability | Comprueba los asientos disponibles en las distintas sesiones del cine |
| RequiredInterface getBillboard | Recupera los datos de la cartelera del cine y sus sesiones. |
| RequiredInterface getCostumer | Recupera los datos de un usuario registrado en BD |
| RequiredInterface getSystemUser | Recupera a un usuario del sistema de BD |
| RequiredInterface newCostumer | Crea un nuevo usuario en la BD |
| RequiredInterface newReservation | Confirma una nueva reserva online de asientos. |
| RequiredInterface sellTicket | Vende en taquilla una entrada |
| RequiredInterface updateData | Actualiza información de la base de datos. |

## Server Web View

**WebServer** - *(Component diagram)*



Figure: 1

### DataBase

Base de datos usada por medio de persistencia empleando jdbc.

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getAvailability | Version: 1.0 | verifica si la silla está disponible para la reserva |
| Port getBillboard | Version: 1.0 | trae de la base de datos las películas en la cartelera |
| Port getCustomer | Version: 1.0 | comprueba si es un usuario correcto |
| Port getMovies | Version: 1.0 | trae las películas en cartelera Tvp |
| Port getPlacesBySession | Version: 1.0 | trae las sillas disponibles para la sesión Tvp |
| Port getSessionsByMovie | Version: 1.0 | trae las sesiones de una película Tvp |
| Port newCustomer | Version: 1.0 | crea un nuevo usuario |
| Port newMovie | Version: 1.0 | crea una nueva película controlPanel |
| Port newReservation | Version: 1.0 | reserva un lugar en una sesion |
| Port newReserve | Version: 1.0 | reserva un lugar en una sesión Tvp |
| Port newSession | Version: 1.0 | crea una nueva sesión controlpanel |
| Port setCustomer | Version: 1.0 | modifica los datos de un cliente |
| Port updateMovie | Version: 1.0 | actualiza una pelicula |
| Port updateSession | Version: 1.0 | actualiza una sesion |

### PaymentGateway

Pasarela de pago externa que comprueba los pagos con tarjeta de crédito.

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port request | Version: 1.0 | envía la trasferencia para que se procese el pago |
| Port response | Version: 1.0 | respuesta si el pago fue éxito o fallo |

### WebServer

Servidor dedicado a la venta online

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getAvailability | Version: 1.0 | pide confirmar la disponibilidad de una silla |
| Port getBillboard | Version: 1.0 | pide la cartelera del cine |
| Port getCustomer | Version: 1.0 | pide confirmar el inicio de sesión de un cliente |
| Port newCustomer | Version: 1.0 | guarda un nuevo cliente |
| Port newReservation | Version: 1.0 | guarda una nueva reservacion |
| Port request | Version: 1.0 | comunicación con la pasarela de pago |
| Port response | Version: 1.0 |  |
| Port setCustomer | Version: 1.0 |  |
| ProvidedInterface Start-Stop | Version: 1.0 |  |

### Accept

Pide la confirmacion el pago a la pasarela de pago y guarda en la bd

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Dependency  Source -> Destination | Public  Reservation | Public  Accept | recibe la información para reservar en una sesion |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port newReservation | Version: 1.0 | guarda la reservación |
| Port request | Version: 1.0 | envía información para el pago |
| Port response | Version: 1.0 | recibe la respuesta del pago |

### Index

Pantalla principal

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Delegate  Source -> Destination | Public  Index | Public  Reservation | muestra las sesiones disponibles de una pelicula |
| Delegate  Source -> Destination | Public  Index | Public  Register | muestra un formulario para el registro de clientes nuevos |
| Delegate  Source -> Destination | Public  Index | Public  Login | loguea al usuario registrado |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getBillboard | Version: 1.0 | trae la cartelera de la base de datos |
| Port getCustomer | Version: 1.0 | comprueba el inicio de sesión de los usuarios |

### Login

Espacio personal de los usuarios

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Delegate  Source -> Destination | Public  Register | Public  Login | login automatico luego del registro |
| Delegate  Bi-Directional | Public  Login | Public  UserData | modificación de los datos personales |
| Delegate  Source -> Destination | Public  Index | Public  Login | login por parte del usuario registrado |

### Register

Guarda un Nuevo usuario en el sistema

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Delegate  Source -> Destination | Public  Register | Public  Login | inicio de sesión luego de completar el registro |
| Delegate  Source -> Destination | Public  Index | Public  Register | reacion de un nuevo usuario |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port newCustomer | Version: 1.0 | crea un nuevo usuario |

### Reservation

Muestra las sesiones de una pelicula ya seleccionada

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Dependency  Source -> Destination | Public  Reservation | Public  Accept | luego de elegir las sillas se procede con el pago |
| Delegate  Source -> Destination | Public  Index | Public  Reservation | muestra las sesiones para una pelicula |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getAvailability | Version: 1.0 | pregunta sobre la disponibilidad de una silla |

### UserData

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Delegate  Bi-Directional | Public  Login | Public  UserData | modifica la información personal de un usuario |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port setCustomer | Version: 1.0 | modifica los datos del usuario |

## ControlPanel

Espacio personal para los usuarios de sistema



Figure: 2

### MoviesEdit

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port newMovie | Version: 1.0 | crea una nueva pelicula |
| Port updateMovie | Version: 1.0 | modifica una pelicula |

### SessionsEdit

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port newSession | Version: 1.0 | crea una sesión nueva |
| Port updateSession | Version: 1.0 | modifica una sesión existente |

### TVP

#### Accept

Confirma la reservacion de una silla

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Assembly  Source -> Destination | Public  Accept | Public  Places | envía las sillas a reservar |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port newReserve | Version: 1.0 | guarda la reserva |

#### Movies

Lista de peliculas que estan en la actual cartelera

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Assembly  Source -> Destination | Public  Sessions | Public  Movies | sesiones disponibles con dicha pelicula |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getMovies | Version: 1.0 | devuelve la cartelera del cine |

#### Places

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Assembly  Source -> Destination | Public  Accept | Public  Places | envio de sillas para el pago |
| Assembly  Source -> Destination | Public  Places | Public  Sessions | plazas de las sesiones |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getPlacesBySession | Version: 1.0 | plazas disponibles de una sesion |

#### Sessions

Sessiones de reproduccion de las pellicles

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Assembly  Source -> Destination | Public  Places | Public  Sessions |  |
| Assembly  Source -> Destination | Public  Sessions | Public  Movies |  |

*Embedded Elements*

| **Element** | **Detail** | **Notes** |
| --- | --- | --- |
| Port getSessionByMovie | Version: 1.0 | trae las sesionde de una pelicula |
| ProvidedInterface ProvidedInterface1 | Version: 1.0 |  |

## DataBase View



Figure: 1

### customer

*Database:* , *Stereotype:* «table», Package: Data Model

Clientes registrados en el sistema

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | name | varchar | False | False |  |  |  |  |  |
| False | surnames | varchar | False | False |  |  |  |  |  |
| False | email | varchar | False | False |  |  |  |  |  |
| False | creditCard | varchar | False | False |  |  |  |  |  |
| False | login | varchar | False | False |  |  |  |  |  |
| False | passw | varchar | False | False |  |  |  |  |  |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_customer | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** customer.  **0..\*** place. | representa las reservas de sillas hechas por los clientes |

### movie

*Database:* , *Stereotype:* «table», Package: Data Model

Películas registradas en el sistema

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | name | varchar | False | False |  |  |  |  |  |
| False | category | varchar | False | False |  |  |  |  |  |
| False | synopsis | varchar | False | False |  |  |  |  |  |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_movie | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** movie.  **0..\*** session. | Representa las sesiones en las que pertenece la película |

### place

*Database:* , *Stereotype:* «table», Package: Data Model

Lugares/sillas reservadas en las sesiones impuestas en el cine

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | id\_session | integer | False | False |  |  |  |  |  |
| False | seat | integer | False | False |  |  |  |  | numero de silla reservada |
| False | id\_customer | integer | False | False |  |  |  |  |  |
| False | creditCard | varchar | False | False |  |  |  |  | número de tarjeta de crédito con la que se realizo la reserva |
| False | amount |  | False | False |  |  |  |  | monto de la reserva |
| False | paymentDate |  | False | False |  |  |  |  | fecha de la reserva |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_place | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** customer.  **0..\*** place. | reserva hecha por un cliente |
| **1** session.  **1..\*** place. | lugares reservados de una sesión especifica |

### room

*Database:* , *Stereotype:* «table», Package: Data Model

Salas del cine en donde se ubican las sesiones del cine

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | access | varchar | False | False |  |  |  |  | nombre identificativo de la sala |
| False | seatingCapacity |  | False | False |  |  |  |  | capacidad de la sala |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_room | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** room.  **1..\*** session. | lugar donde se establece las sesiones |

### session

*Database:* , *Stereotype:* «table», Package: Data Model

Intervalos de tiempo en los que se lleva a cabo la reproducción de una pelicula

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | id\_room | integer | False | False |  |  |  |  |  |
| False | id\_movie | integer | False | False |  |  |  |  |  |
| False | id\_sessionType | integer | False | False |  |  |  |  |  |
| False | startDate | date | False | False |  |  |  |  | fecha desde la que estará activa en la cartelera |
| False | endDate | date | False | False |  |  |  |  | fecha de caducidad para mostrarse en la cartelera |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_session | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** session.  **1..\*** place. | lugares reservados en la sesion |
| **1** room.  **1..\*** session. | lugar donde se llevara a cabo la sesion |
| **1** sessionType.  **1..\*** session. | tipo al que pertenece la sesion |
| **1** movie.  **0..\*** session. | película que se reproducirá |

### sessionType

*Database:* , *Stereotype:* «table», Package: Data Model

Representa el tipo de las session, considerando su horario y precio

Columns

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | name | varchar | False | False |  |  |  |  |  |
| False | startTime | integer | False | False |  |  |  |  | hora de inicio de la sesion |
| False | cost | double | False | False |  |  |  |  | coste de la sesion |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_sessionType | Public | id |  |  |

*Relationships*

| **Association** | **Notes** |
| --- | --- |
| **1** sessionType.  **1..\*** session. | horario y precio para las sesiones |

### systemUser

*Database:* , *Stereotype:* «table», Package: Data Model

Representa la información de los usuarios que usan el sistema

Columns

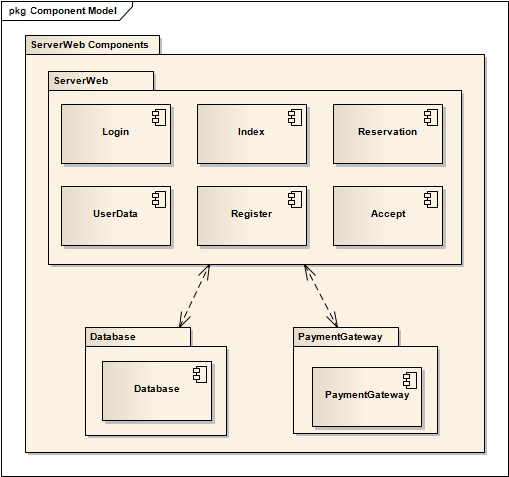
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** | **Name** | **Type** | **Not Null** | **Unique** | **Len** | **Prec** | **Scale** | **Init** | **Notes** |
| True | id | integer | True | False |  |  |  |  |  |
| False | login | varchar | False | False |  |  |  |  |  |
| False | passw | varchar | False | False |  |  |  |  |  |
| False | type | varchar | False | False |  |  |  |  | user, admin |
| False | mSale | boolean | False | False |  |  |  |  | indica si el usuario puede vender |
| False | mSession | boolean | False | False |  |  |  |  | indica si el usuario puede crear y modificar sesiones |
| False | mMovies | boolean | False | False |  |  |  |  | indica si el usuario puede crear y modificar peliculas |

Constraints

| **Name** | **Type** | **Columns** | **Initial Code** | **Notes** |
| --- | --- | --- | --- | --- |
| PK\_systemUser | Public | id |  |  |

## Packagemodelview

Al eliminar la parte de la aplicación destinada a la ejecución en local, la vista de los paquetes queda reducida esencialmente a una.



## Deployment View

Vista que representa la configuración de despliegue de la aplicación.



Figure: 2

### Bank

Banco que nos proporciona sus pasarelas de pago.

#### PaymentGateway

Pasarela de pago provista por el banco

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation  Source -> Destination | Public  PaymentHandler | Public  PaymentGateway |  |

### DatabaseServer

Base de datos de la aplicación, que contiene la información relevante del sistema.

#### SGBD

Sistema de gestión de la base de datos

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation  Source -> Destination | Public  DatabaseHandler | Public  SGBD |  |

### Employee

Empleado del cine que se encarga de la taquilla o de utilizar el sistema de gestion de información.

#### WebBrowser

Tecnología de navegador que utilice la empresa para sus empleados.

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation http  Source -> Destination | Public  FrontEnd | Public  WebBrowser |  |

### User

El usuario que se encuentra en su casa dispuesto a comprar entradas online

#### WebBrowser

Sistema que utilizará el usuario para conectarse a la pagina web

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation http  Source -> Destination | Public  FrontEnd | Public  WebBrowser |  |

### WebServer

Servidor web que contiene la aplicación web desarrollada

#### BackEnd

Contiene la logica de la aplicación, que se comunicara con la base de datos

##### DatabaseHandler

Componente encargado de gestionar la comunicación con la base de datos relacional.

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation  Source -> Destination | Public  DatabaseHandler | Public  SGBD |  |

##### GraphDatabase

Base de datos de grafo utilizada por la aplicación para servir recomendaciones.

##### PaymentHandler

Componente encargado de comunicarse con la pasarela de pagos bancarios.

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation  Source -> Destination | Public  PaymentHandler | Public  PaymentGateway |  |

#### FrontEnd

Contiene la zona pública de la aplicación, con sus interfaces de usuario.

*Connections*

| **Connector** | **Source** | **Target** | **Notes** |
| --- | --- | --- | --- |
| Realisation http  Source -> Destination | Public  FrontEnd | Public  WebBrowser |  |
| Realisation http  Source -> Destination | Public  FrontEnd | Public  WebBrowser |  |

##### EmployeeWebSide

Zona de la web dedicada al uso por parte de los empleados

##### Index

Indice de la web.

##### UserWebSide

Conjunto de páginas web dedicadas al cliente.

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-1)