SystranHorizonte

Architecture Notebook

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Propósito

El Documento de Arquitectura de Software (SAD) proporciona una visión arquitectónica integral del Sistema de transporte Horizonte. Presenta una serie de puntos de vista diferentes de arquitectura para representar diferentes aspectos del sistema. Se tiene la intención de captar y transmitir las decisiones arquitectónicas significativas que se hayan introducido en el sistema.

# Los objetivos de arquitectura y filosofía

En esta sección se describen los requisitos de software y los objetivos que tienen algún impacto significativo en la arquitectura.

# Plataforma Técnica

El sistema diseñado será desplegado en un ordenador con sistema operativo Windows 7, IIS7 y con una base de datos SQL2012Express.

# Transacción

De acuerdo a la solicitud de la empresa, el sistema web, se ha diseñado de tal forma que se pueda obtener acceso desde cualquier parte de su local conectado a la red local.

# Seguridad

El sistema contará con seguridad, para el acceso:

* Autenticación: Inicia sesión usando el usuario y clave.
* Autorización: De acuerdo a su perfil, al usuario en línea se debe conceder o no para llevar a cabo algunas acciones específicas (ventas, consultas, reportes, etc.).
* Integridad de datos: Los datos enviados de la red no pueden ser modificados.

# Persistencia

Para la persistencia de datos se abordará el uso de una base de datos relacional.

# Fiabilidad / disponibilidad

La disponibilidad del sistema es un requisito clave por la naturaleza, ya que es un sistema de ventas. La arquitectura debe garantizar capacidades de conmutación por error. La disponibilidad será abordada a través de la plataforma IIS. Disponibilidad dirigida es 12/7: 12 horas al día, 7 días a la semana, y 1 hora se reserva para las actividades de mantenimiento.

# Rendimiento

Los procesos de ventas (venta de pasajes, encomiendas y reservas), debe tener menos de 5 segundos.

# Supuestos y dependencias

* Arquitectura separada por capas: Todo lo relacionado con la interfaz del usuario va en una capa, las reglas del negocio en otra y el manejo de datos en una tercera capa. No se mezcla en una capa código correspondiente a otra. El código correspondiente a una capa puede ser reutilizado desde varias partes de la capa inmediatamente superior. Sabiendo dónde está el código correspondiente a cada capa, pueden realizarse modificaciones dentro de una capa para mejorar o aumentar el tamaño del sistema de software, con un mínimo impacto en las capas restantes.
* Unity (Inyección de Dependencias): Unity brinda todas las herramientas necesarias para poder desarrollar aplicaciones con bajo acoplamiento, por tanto, un Framework de inyección de dependencias o DI Container.
* Entity Framework: Es de ayuda para las conexiones, el costo de abrir una conexión se distribuye entre todo el grupo. El proveedor de datos .NET para SQL Server admite la agrupación de conexiones.
* Bootstrap: Es un framework de css, en otras palabras es un conjunto de archivos CSS que se incluyes en la página web y se puede empezar a maquetar el sitio web en minutos, sin tocar una sola linea de CSS.
* Android Studio: Es un IDE para el trabajo en SO Android.

# Requisitos Arquitectónicamente significativos

* El poder ser accesible al sistema en todo el negocio
* El sistema debe responder dentro de los 5 segundos.
* El sistema debe desplegar en SO Windows 7
* El sistema debe mandar alertas de las reservas y ventas si no hay presencia de los clientes.
* El sistema debe mantener un registro de los movimientos sobre los usuarios del sistema.
* El sistema debe poder ser accesible desde un ordenador, Tablet, celular.

# Decisiones, las limitaciones, y justificaciones

* Entorno Web, pueden haber diferentes inconvenientes con las configuraciones del servidor, firewall.
* Servidor, el hardware o el sistema operativo pueden fallar, por lo cual se optará por usar backup de la base de datos en la nube.
* Topología de red, en caso de no haber un buen diseño de red, puede fallar el acceso al sistema desde la aplicación android o desde otros ordenadores.
* Aplicación Android, se usará android studio para la elaboración de la aplicación.
* La data mostrada en la aplicación android, va a ser obtenida del servidor, gracias a que este brinda servicios JSON, con la data necesaria.

# Mecanismos de arquitectura

## Persistencia de datos

Estado: Diseño

Para el almacenamiento se datos se opta por la utilización de SQLServer2012Express

## Comunicación

Estado: Diseño

La comunicación entre las diferentes capas se realiza a través de inyección de dependencias con el framework Unity, para la comunicación del sistema con la base de datos se utiliza el framework Entity Framwork.

## Plataforma

Estado: Diseño

La plataforma para el despliegue del sistema es IIS, el cual va a servir como medio de comunicación entre el servidor y los clientes

## Construcción del Sistema

Estado: Diseño

El IDE para el desarrollo del sistema es Visual Studio 2012 Express, para la parte móvil se usa el IDE Android Studio.

# Abstracciones clave

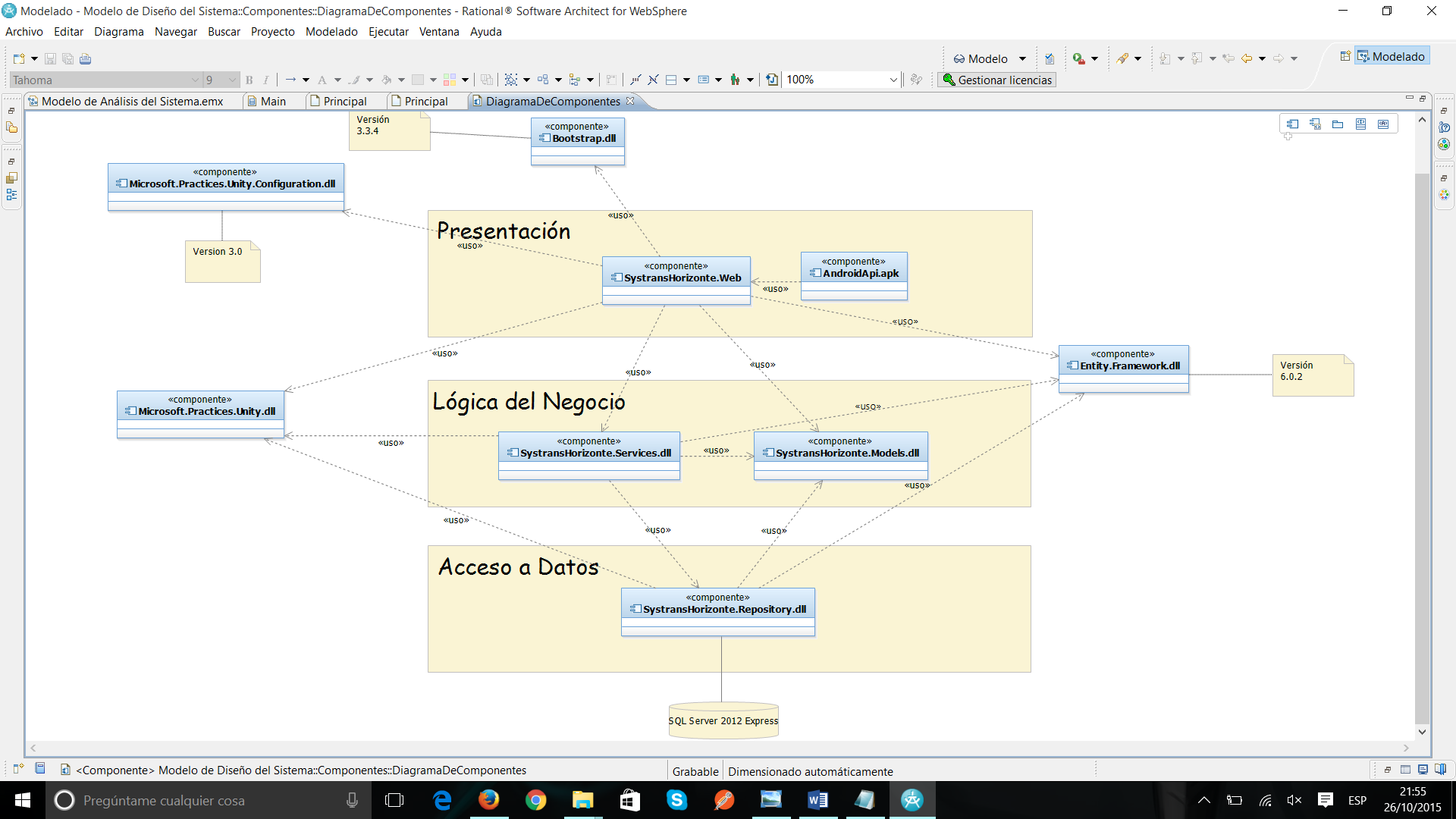
* Pasajero: Persona que viaja en un vehículo, especialmente de un avión, barco o tren, sin conducirlo.
* Pasaje: Derecho que se paga por pasar por un lugar.
* Comprador: Que compra.
* Reserva: Acción de reservar una plaza o localidad para un transporte público.

# Capas o Framework arquitectónico

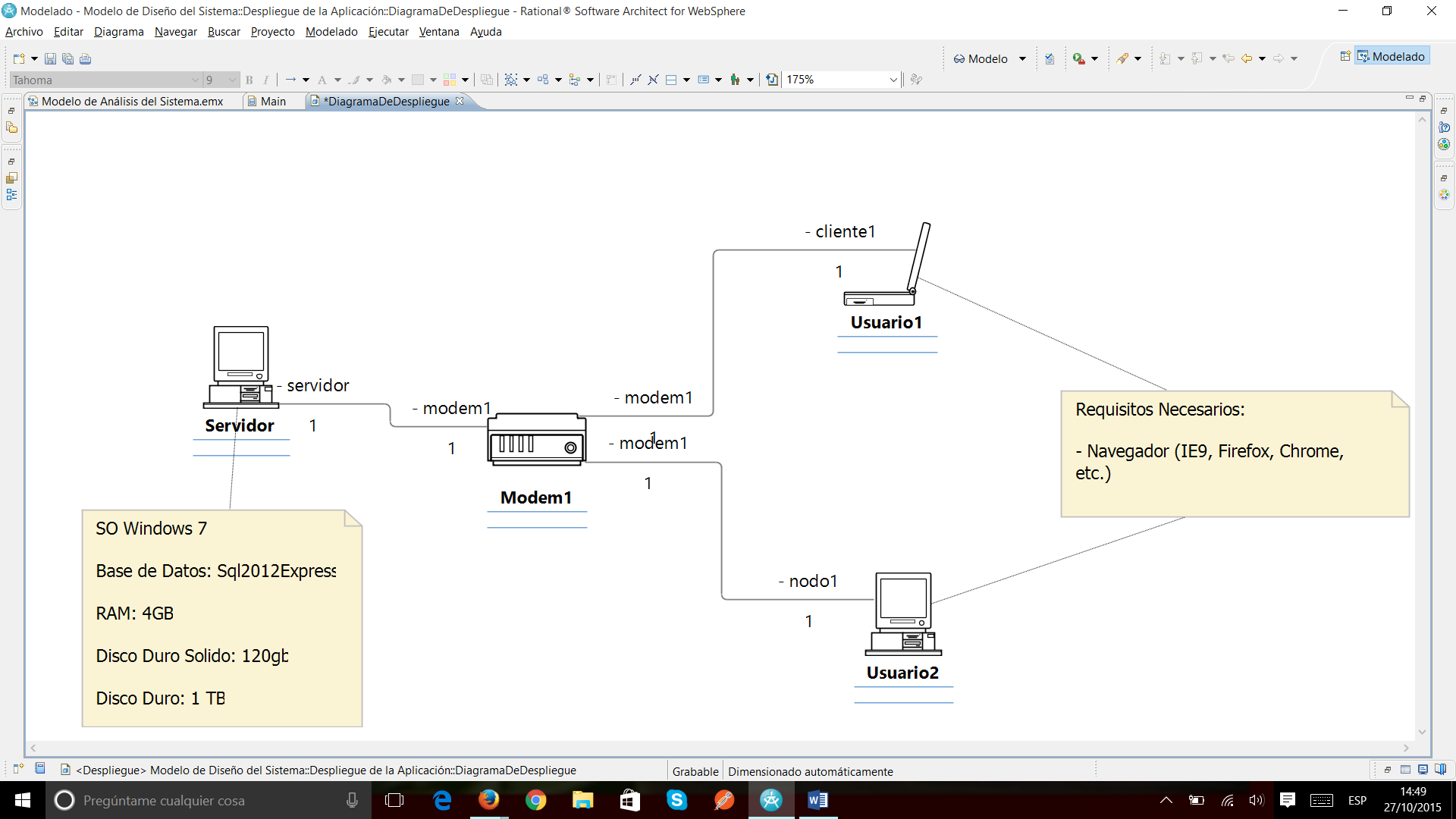
La arquitectura del Sistema se basa en el patrón de diseño MVC, Este patrón plantea la separación del problema en tres capas: la capa modelo, que representa la realidad; la capa controlador, que conoce los métodos, atributos del modelo, recibe y realiza lo que el usuario quiere hacer; y la capa vista, que muestra un aspecto del modelo y es utilizada por la capa anterior para interaccionar con el usuario. Además se desarrollará una aplicación móvil con información sobre el sistema, para lo cual se trabajará con datos JSON desde la capa vista del sistema con su respectivo controlador.

# Vistas de arquitectura

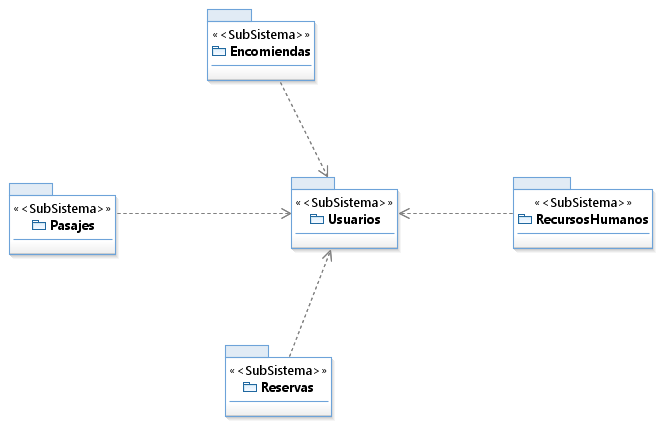
**Diagrama de Componentes**



**Diagrama de Despliegue**



**Diagrama de Sub-Sistemas**

****