

# EMSys

## Descripción de la Arquitectura

### Versión 2.2

#### Historia de revisiones

Fecha	Versión	Descripción	Autor
27/08/2016	2.1	Descripción inicial	Juan Manuel San Martín
28/08/2016	2.2	Revisión, cambio de orden de secciones, numeración.	Bruno Amaral

# Contenido

1.Introducción	3
1.1.Propósito	3
1.2.Alcance	3
1.3.Definiciones, siglas y abreviaturas.	3
1.4.Referencias	3
1.5.Visión general	3
2.Vista del Modelo de Casos de Uso	4
2.1.Diagrama de Casos de Uso relevantes a la Arquitectura	4
2.2.Casos de Uso relevantes a la Arquitectura	4
2.3.Recibir notificación	4
2.4.Listar eventos	4
2.5.Consumir servicio externo	4
3.Vista del Modelo de Distribución	5
3.1.Diagrama de Distribución	5
3.2.Nodos	5
3.2.1.Tablet (Android)	5
3.2.2.Servidor (WebApi)	5
3.2.3.Base de datos	6
3.2.4.Servidor cloud based para notificaciones	6
3.2.5.Servidor externo para consumo de datos	6
3.2.6.Conexiones	6
3.2.6.1.Web Services (REST)	6
3.2.6.2.Notificaciones Push (A Definir)	6
3.2.6.3.Driver de base de datos	6
3.2.6.4.Notificación de cambios	6
3.2.6.5.Consumo de datos externos (Web Services SOAP)	6
4.Secciones a definir	7
4.1.Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño	7
4.2.Vista del Modelo de Diseño	7
4.3.Trazabilidad desde el Modelo de Diseño al Modelo de Implementación	7
4.4.Vista del Modelo de Implementación	7

## **1.Introducción**

### **1.1.Propósito**

Este documento proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del mismo. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema. Y por último pretende justificar éstas últimas de manera correcta, utilizando requerimientos del cliente y técnicas arquitectónicas válidas.

### **1.2.Alcance**

Se pretende abarcar la arquitectura general de la solución que se plantea, junto a esta, se deben justificar las decisiones tomadas. No se pretende ahondar aún demasiado en algunas áreas ya que todavía no se tiene una visión total del sistema y quedan aún temas por resolver con el cliente.

### **1.3.Definiciones, siglas y abreviaturas.**

Ver Glosario.

### **1.4.Referencias**

- Documento de prototipo 1
- Documento de prototipo 2
- Documento de requisitos
- Documento de casos de uso
- Glosario

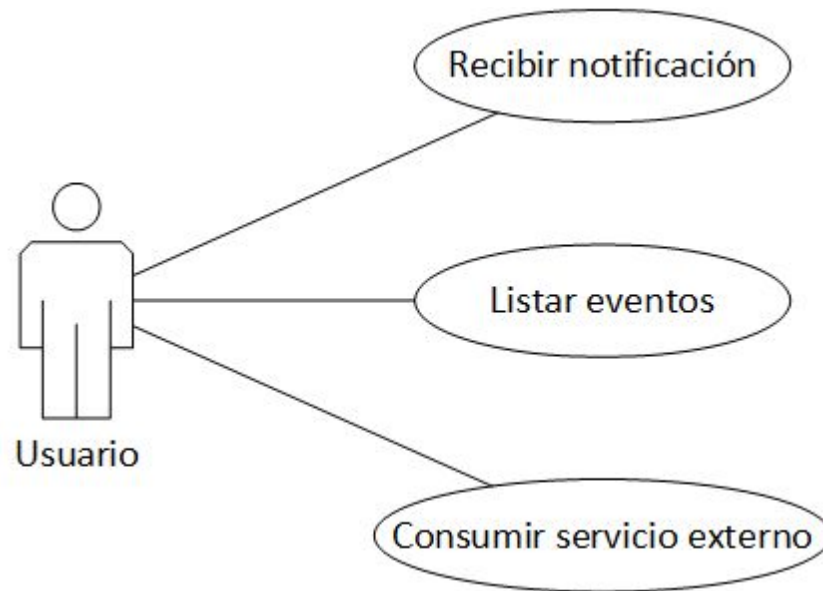
Todos estos documentos se entregarán en el mismo archivo .zip donde se encuentra este.

### **1.5.Visión general**

Se pretende presentar los casos de uso que definen la arquitectura como también la trazabilidad de estos hasta las decisiones arquitectónicas pertinentes.

## 2.Vista del Modelo de Casos de Uso

### 2.1.Diagrama de Casos de Uso relevantes a la Arquitectura



### 2.2.Casos de Uso relevantes a la Arquitectura

En particular, se tomaron estos 3 casos de uso, ya que son los que tienen requerimientos específicos de arquitectura y por tanto modelan el sistema.

Estos casos de uso aún no están definidos, por ello, a futuro puede cambiar su nombre.

- Recibir notificación
- Listar eventos
- Consumir servicio externo

### 2.3.Recibir notificación

Se eligió este caso de uso dentro de los existentes, ya que tiene asociados 2 riesgos importantes. La idea tras la recepción de la notificación es que hay un cambio en la información del evento, esto se puede dar por 2 razones, la primera es si el Backend recibe un cambio sobre un evento por parte de alguno de los dispositivos de Frontend, la segunda razón es por posibles modificaciones en la base. Un requisito no funcional indica que debemos dejar abierta la integración con otros sistemas

El primero es la utilización de un sistema de mensajería de notificaciones entre el Backend (de ahora en más WebApi) y el Frontend (de ahora en más aplicación Android).

El segundo y más severo, el cliente especificó que la base de datos está compartida con otro sistema externo, esto hace referencia a que el sistema debe integrarse a otro, y el punto de contacto con este otro, es la base de datos. Por tanto, es fundamental notificar en tiempo real al usuario de cambios en la base de datos, creados por sistemas externos.

### 2.4.Listar eventos

Listar eventos está dentro del conjunto estándar de casos de uso, y está simplemente listado aquí como un representante de los casos de uso que van a utilizar un mismo canal de comunicaciones entre la WebApi y la aplicación Android.

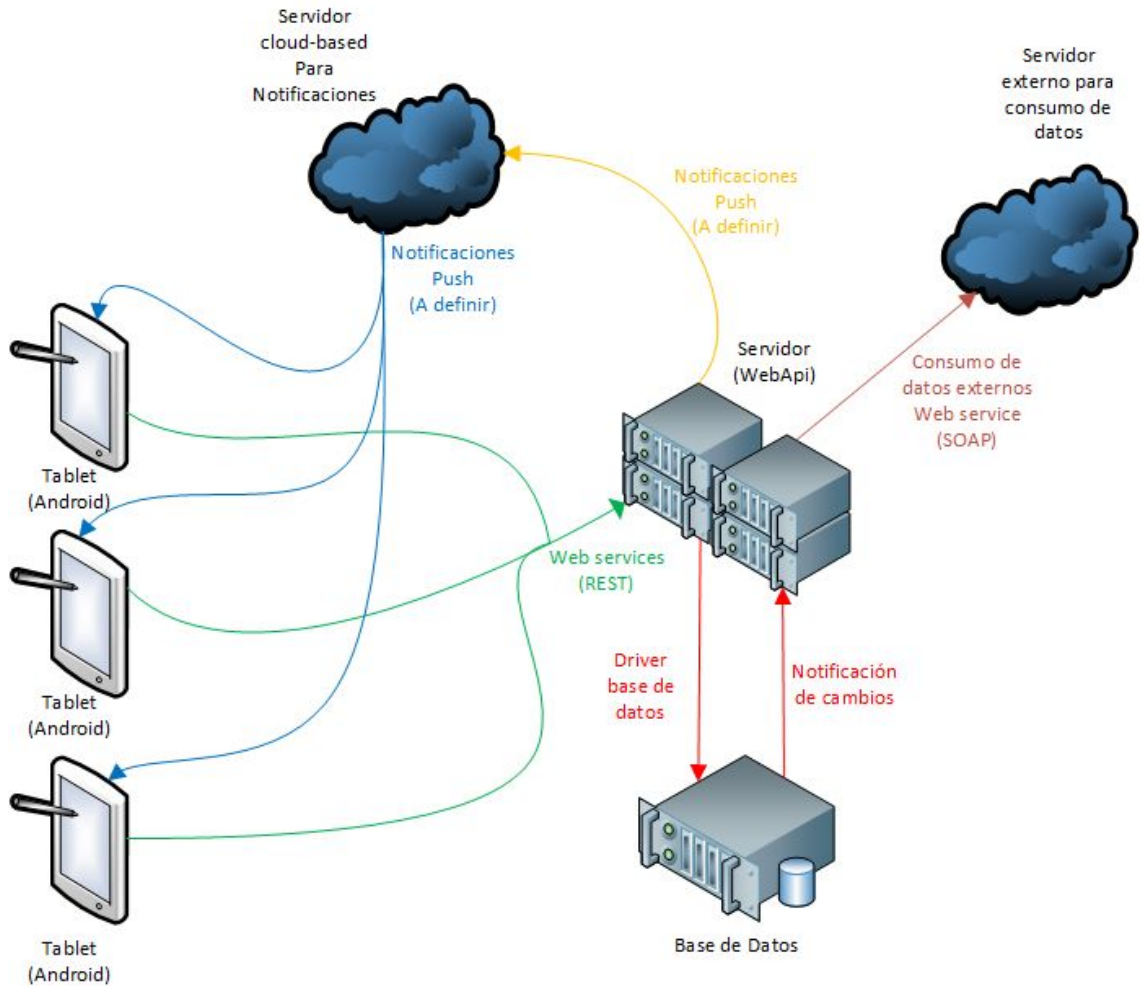
### 2.5.Consumir servicio externo

Este caso de uso hace referencia a un requisito especificado directamente por el cliente, el cual describe que se le debía ofrecer al usuario consumir un servicio provisto por una entidad externa. Sobre este servicio se especificó que se va a tratar de un Web Service de tipo SOAP con una firma fija aún no definida.

### 3.Vista del Modelo de Distribución

#### 3.1.Diagrama de Distribución

De acuerdo a los casos de uso relevantes a la arquitectura se definió la siguiente.



#### 3.2.Nodos

Se pasarán a detallar los nodos relevantes que se visualizan en el Modelo de distribución

##### 3.2.1.Tablet (Android)

Consiste en un dispositivo móvil tipo Tablet, corriendo el sistema operativo Android, sobre el cual se implementará una aplicación cliente, con la cual los usuarios tendrán interacciones. Esta aplicación tiene un contenido lógico mínimo, donde se ofrecerán acciones que impactarán directamente en el Backend (WebApi).

##### 3.2.2.Servidor (WebApi)

El servidor consiste en un equipo, corriendo alguna versión de Microsoft Windows e I.I.S., sobre el cual se ejecuta el segundo componente que debemos implementar de tipo WebApi. Esta es la capa lógica del sistema, donde se publican las operaciones que ejecutan las Tablets, se disparan las notificaciones de tipo Push según corresponda, se ejecutan operaciones sobre la base de datos y se consumen los servicios externos que proveen datos.

### **3.2.3.Base de datos**

La base de datos es relacional, aún no se ha definido el motor, dentro de las posibilidades mencionadas por el cliente se manejan MySQL o SQL Server, de este depende la notificación hacia el Servidor de los cambios en la misma. En el prototipo 2 se están investigando métodos para realizar esto, por lo cual, la salida del prototipo probablemente establecerá qué tipo de base de datos se utilizará.

### **3.2.4.Servidor cloud based para notificaciones**

Este componente consiste en un servicio online encargado de repartir las notificaciones a las distintas aplicaciones que corran sobre las tablets, según disponga el servidor. Este servicio se encarga de manejar el nivel de indirectión necesario, para que el servidor no tenga la responsabilidad de obtener una conexión saliente para cada dispositivo que corra la aplicación Android.

### **3.2.5.Servidor externo para consumo de datos**

Sobre este servidor el cliente no ha dado grandes especificaciones. Se sabe que este ofrecerá una interfaz de tipo Web Service SOAP, donde el servidor, desarrollado como parte de este proyecto, debe consumir elementos de esta interfaz y luego enviar esta información a las tablets que corran la aplicación Android.

### **3.2.6.Conexiones**

Se describirán brevemente las conexiones que unen varios de estos nodos.

#### **3.2.6.1.Web Services (REST)**

Los Web Services de tipo REST son un método de comunicación entre 2 partes, en este caso, se publicará la interfaz de los Web Services en el Servidor y se consumirá desde la tablet que corra la aplicación Android. Estos Web Services ofrecerán toda la información que la aplicación Android necesite de manera de ejecutar todos los casos de uso que comiencen con el actor Usuario.

#### **3.2.6.2.Notificaciones Push (A Definir)**

Para esta conexión aún no se ha establecido una tecnología, consta de mensajes que se envían desde el Servidor hacia las distintas Tablets. Es necesaria para poder contemplar el requisito no funcional de tiempo real. El cual consiste en que las Tablets sean notificadas de cambios en el Servidor sin necesidad de estar realizando operaciones sobre este.

#### **3.2.6.3.Driver de base de datos**

Estos drivers se llaman más comúnmente conectores, y consisten en un software encargado de hacer de interfaz entre una base de datos y una aplicación que desea utilizar la misma. En nuestro caso, al no tener establecido que motor de base de datos se utilizará, no se puede saber que driver es necesario aún.

#### **3.2.6.4.Notificación de cambios**

Esta conexión tampoco está definida aún, y se creó el prototipo 2 para poder investigar qué opciones existen. Esta conexión es necesaria para que el Servidor pueda ser notificado de cambios externos sobre la base de datos, sin necesidad de estar realizando operaciones sobre esta.

#### **3.2.6.5.Consumo de datos externos (Web Services SOAP)**

Esta conexión consiste en el consumo de Web Services de tipo SOAP, estos son un tipo de Web Services, los cuales existen para comunicar 2 entidades, y funcionan publicando métodos en una de ellas, mientras que la otra debe consumir estos últimos. Sobre estos el cliente sólo ha especificado que son de tipo SOAP y que probablemente la firma tenga un formato sencillo.

## **4.Secciones a definir**

**4.1.Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño**

**4.2.Vista del Modelo de Diseño**

**4.3.Trazabilidad desde el Modelo de Diseño al Modelo de Implementación**

**4.4.Vista del Modelo de Implementación**