

DOCUMENTO DE LA ARQUITECTURA DEL SOFTWARE

ADMINISTRADOR DE CONGRESOS ACADÉMICOS

**INSTITUTO TECNOLÓGICO DE COSTA RICA
SEDE DE CARTAGO**

**EQUIPO DE TRABAJO
BRANDON DINARTE CHAVARRÍA
JULIAN SALINAS ROJAS**

**FECHA
03/05/2018**

TABLA DE CONTENIDOS

Tabla de contenidos	2
Historial de revisiones	3
Introducción	4
Propósito	4
Alcance	4
Definiciones, acrónimos y abreviaturas	4
Referencias	5
Representación de la arquitectura	5
Metas de arquitectura y restricciones	6
Vista de casos de uso	7
Vista lógica	8
Vista general	8
Diseño de paquetes	8
Vista de procesos	9
Vista de despliegue	10
Vista de implementación	11
Visión general	11
Diagrama de clases	13
Vista de datos	13
Tamaño y rendimiento	14
Calidad	14
Apéndice A: Estructura de la base de datos	16

HISTORIAL DE REVISIONES

Nombre	Fecha	Razón de cambio	Versión del documento
Julian Salinas Rojas Brandon Dinarte Chavarría	02/05/2018	Redacción del documento	1.0.0
Julian Salinas Rojas Brandon Dinarte Chavarría	03/05/2018	Añadir diagramas UML	1.1.0

1. INTRODUCCIÓN

1.1. PROPÓSITO

El presente documento tiene como objetivo describir una visión inicial de la arquitectura del sistema, usando como punto de partida las necesidades especificadas en el Documento de Requerimientos de Software, versión *1.2.0*.

Se explica tanto el diseño lógico como físico del sistema. Además se describen las distintas decisiones arquitectónicas que han sido tomadas en beneficio de la funcionalidad, usabilidad, mantenibilidad y demás atributos de la calidad de software.

1.2. ALCANCE

Este documento involucra a todos los interesados en la arquitectura del sistema y abarca tanto la primera como la segunda iteración. Todos los requerimientos que han sido descritos hasta el momento se ven reflejados en este documento por medio de los distintos artefactos que se incluyen. Para más detalles acerca de cuales funcionalidades se toman en cuenta en el desarrollo de la arquitectura refiérase al Documento de Especificación de Requerimientos.

1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

La mayoría de términos técnicos que se utilizan ya han sido definidos en el Apéndice A del Documento de Especificación de Requerimientos. A continuación, se retoman algunos de los términos utilizados en este documento:

Término	Interpretación
API	Conjunto de definiciones de subrutinas, protocolos y herramientas para la creación de aplicaciones.
Base de datos	Colección de información digital estructurada y organizada.
MVC	Patrón de diseño que se sustenta en la división de los componentes del

	sistema en tres capas principales; modelo, vista y controlador.
Plataforma	Sistema base para hacer funcionar determinados módulos de software con los que es compatible.
Servidor	Aplicación en ejecución (software) capaz de atender las peticiones de un cliente mediante el envío de respuestas.

1.4. REFERENCIAS

El presente documento hace referencia a otros documentos y archivos que han sido presentados anteriormente, destacando entre los demás, el de “Especificación de Requerimientos de Software”. Estos documentos pueden ser solicitados nuevamente al administrador del proyecto cuando sea necesario. En futuras actualizaciones, se contará con referencias hacia otros documentos que sean necesarios para definir los requerimientos. A continuación, la lista de documentos y archivos a los que se hace referencia:

Última actualización	Nombre del documento
18/02/2018	Acta Constitutiva
25/02/2018	Visión y Alcance del Proyecto
01/03/2018	Prototipo del sistema
11/03/2018	Especificación de Requerimiento de Software

2. REPRESENTACIÓN DE LA ARQUITECTURA

El sistema es representado por medio de distintos artefactos que serán descritos posteriormente, entre ellos destacan el diagrama de clases y la vista de los datos, considerados como los más importantes dada la naturaleza del proyecto en cuestión. En general, la descripción de la arquitectura de software incorpora la vista de casos de uso, que describen los procedimientos que el usuario puede realizar con el sistema.

Estos casos de uso están descritos en el documento de Especificación de Requerimientos de Software. También se incluye la vista de datos y una vista lógica por medio del diagrama de paquetes. En relación con la vista de los datos, esta se no presenta por medio de un diagrama de base de datos SQL, debido a que se trata de una del tipo documental. Por esta razón únicamente se muestra la estructura que tendrán cada uno de los documentos.

3. METAS DE ARQUITECTURA Y RESTRICCIONES

El objetivo principal de plantear una arquitectura adecuada es cumplir con todos los atributos de calidad descritos en la sección 5.3 del documento de Especificación de Requerimientos de Software, relacionados directamente con el estándar ISO/IEC 9126. Esto con el fin de proporcionar un medio de consulta adecuado tanto al cliente y los usuarios, como a las personas que en futuro darán mantenimiento al sistema.

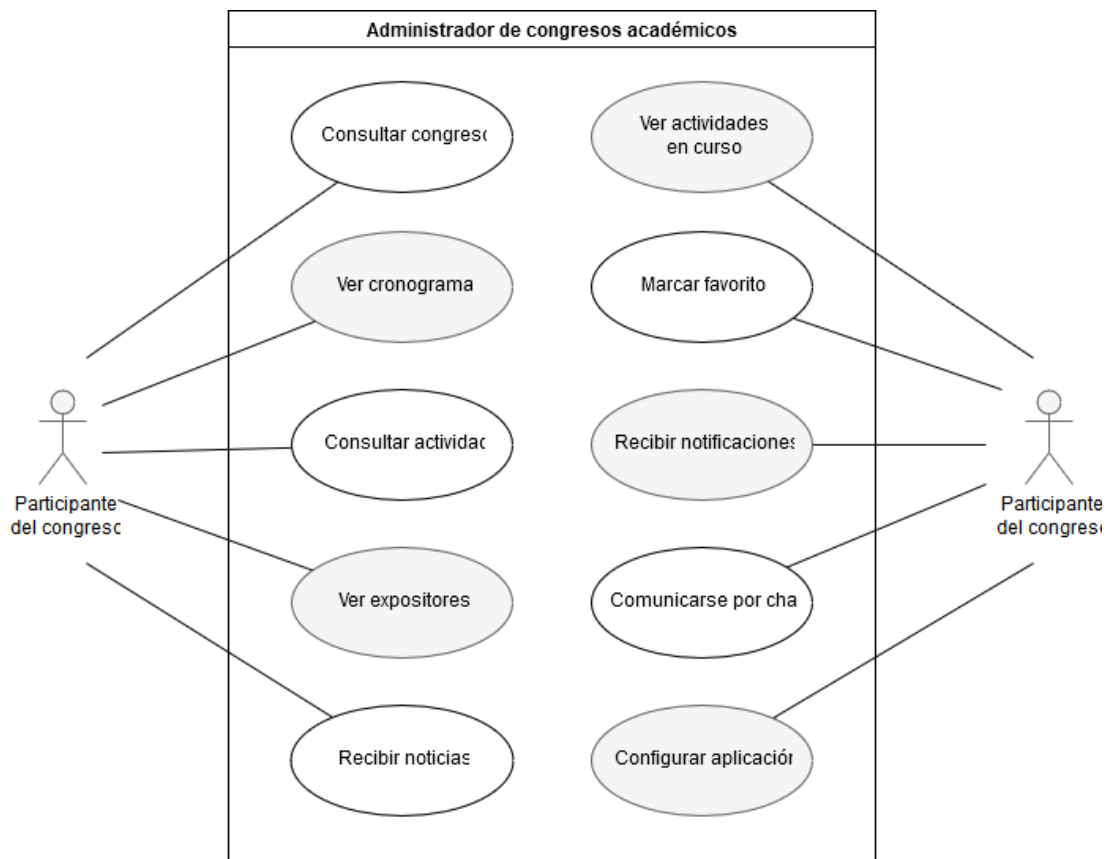
En cuanto a las restricciones de la arquitectura se deben rescatar las descritas en la sección 2.4, Ambiente Operativo y 2.5, Restricciones de Diseño e Implementación del mismo documento de Especificación de Requerimientos. Algunas de esas restricciones son:

- La aplicación móvil tendrá como objetivo el sistema operativo Android 8.0, retrocompatible hasta la versión 5.0 Lollipop.
- Es sistema tendrá un máximo 200 usuarios conectados de forma concurrente.
- Debido al sistema operativo en el se ejecutará la aplicación y a la experiencia del equipo de desarrollo, el lenguaje de programación usando será Java 8.
- Debido a la dimensión y disponibilidad de los datos, se utiliza la plataforma de servicios Firebase.
- La implementación del sistema a desarrollar estará bajo la licencia GNU General Public License.

4. VISTA DE CASOS DE USO

Los casos de uso se ven reflejados significativamente en la arquitectura del sistema, por lo que es importante considerar la totalidad de estos. Los casos de uso ya han sido fijados en documentos anteriores, por lo que si desea obtener una descripción detallada de estos, refiérase al Apéndice B del documento de Especificación de Requerimientos de Software.

A continuación se muestra el diagrama de casos de uso de la aplicación móvil:



5. VISTA LÓGICA

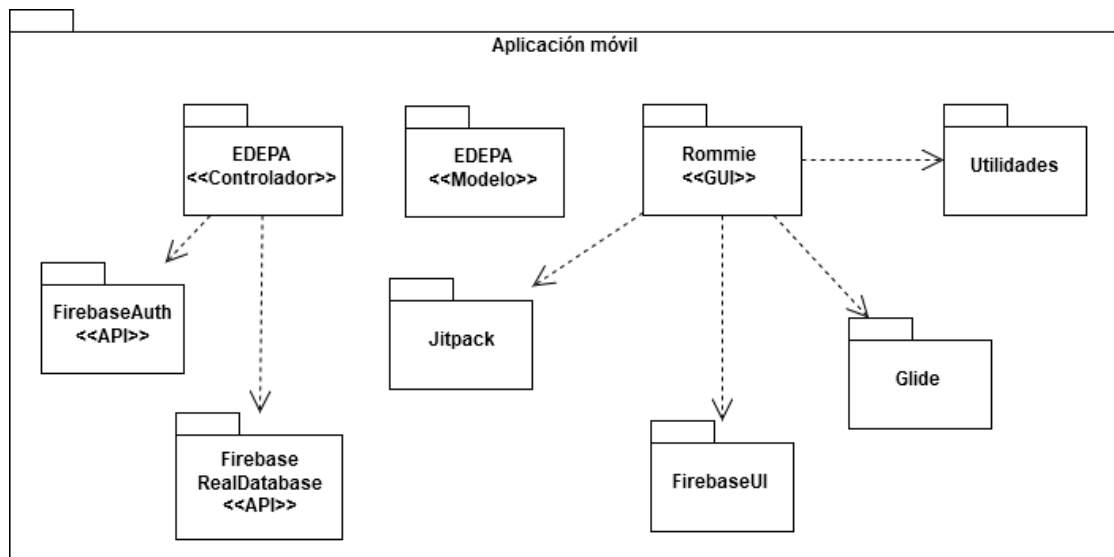
5.1. VISTA GENERAL

En esta sección se describen las partes significativas del diseño, como los subsistemas y las interfaces. Además, se presenta como se descompone el sistema en paquetes y clases, así como la relación que existen entre éstas. Además, se presenta de forma detallada la información que maneja el sistema, así como el host que da alojamiento.

5.2. DISEÑO DE PAQUETES

Como se ha mencionado en documentos anteriores, el sistema tiene dos partes, una aplicación móvil que puede ser usada por el usuario final y una aplicación web que solo puede ser usada por usuarios organizadores. Los tipos de usuarios, su papel en el sistema y las funcionalidades que utilizan han sido definidas de igual manera en la sección 5.2 del documento de Especificación de Requerimientos.

A continuación se define el diagrama de paquetes de la aplicación móvil:



La aplicación móvil cuenta con tres paquetes principales uno por cada capa del modelo MVC. Debido a que es poca la información que se maneja, el controlador es

el encargado de conectarse a las interfaces que proporciona la plataforma de servicios Firebase. Este mismo controlador es el encargado de sincronizar los datos que recibe de la base datos con el resto de la aplicación.

El paquete de modelo contiene las clases correspondientes al dominio de la aplicación, como el congreso, eventos, expositores, entre otros. Estas clases carecen de métodos por lo que son usadas a su vez como clases de transferencia de datos.

Como es de esperarse, el paquete correspondiente a la interfaz de usuario maneja una gran cantidad de clases. Algunas de ellas implementan un mecanismo que permite auto-ajustarse según los cambios que ocurran en los datos. Otras son adaptadores que sirven para visualizar la información, mientras que otras solamente funcionan para pequeños detalles y animaciones.

6. VISTA DE PROCESOS

El sistema está compuesto de múltiples procesos que trabajan constantemente en segundo plano. A nivel de la aplicación móvil se identifican los siguientes:

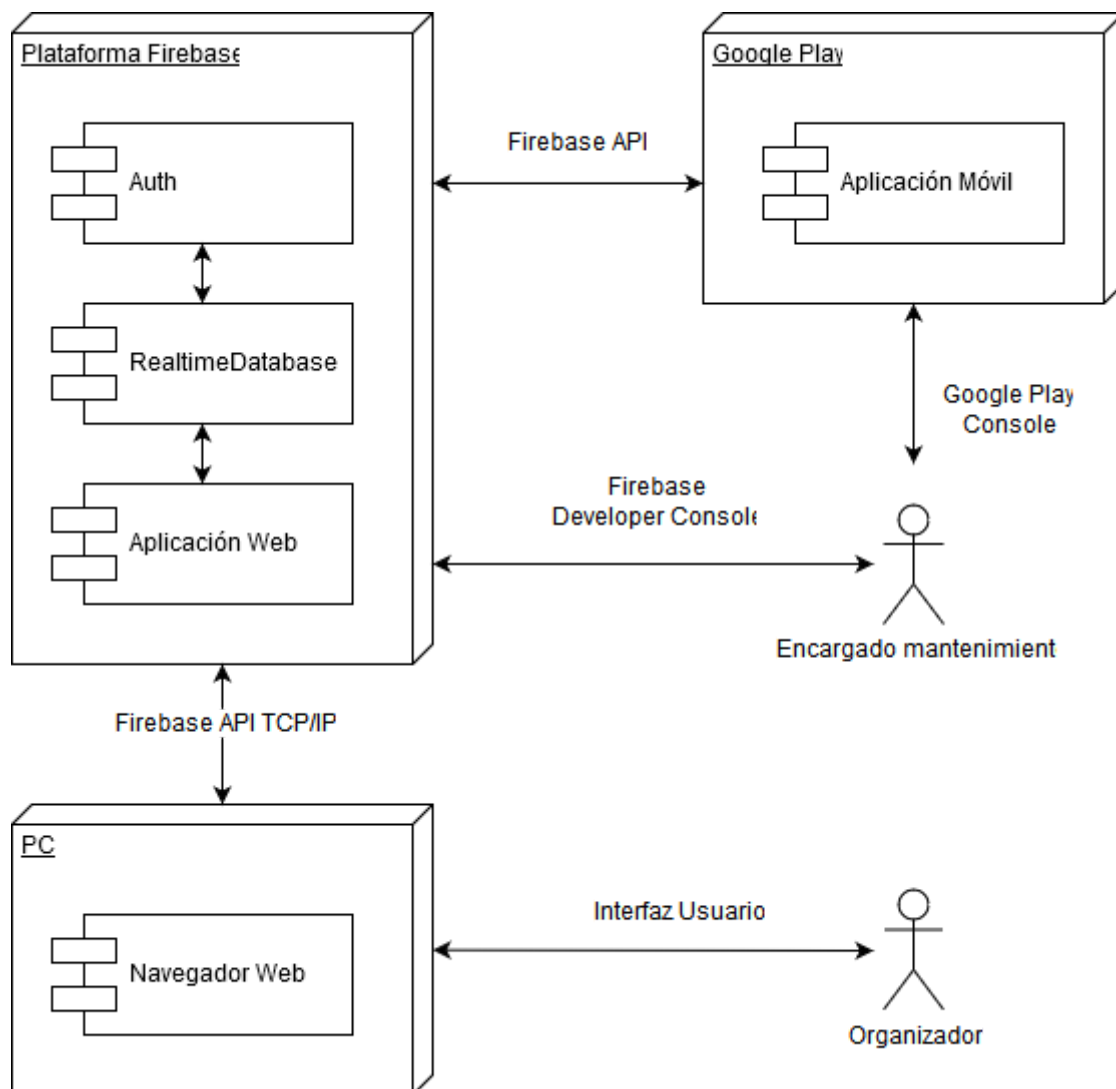
- Control principal de la aplicación. Se encarga de la interacción con el usuario, recibe las entradas del mismo, despliega las vistas que este solicita y ejecuta las operaciones correspondientes de navegación y ejecución de la aplicación. Este proceso interactúa a su vez con los siguientes procesos.
 - Control de notificaciones. Revisa constantemente el servicio de administración de notificaciones, esto con el fin de desplegar las ocurrencias de las mismas mediante la interacción con el sistema operativo del dispositivo móvil.
 - Control de actualización de información en tiempo real. Actualiza la información desplegada en la aplicación según ésta es editada en la base de datos en la nube, este proceso se mantiene activo únicamente para las vistas actualmente instanciadas de cada una de las secciones de la aplicación.

A nivel del sistema web se pueden identificar los siguientes procesos:

- Control de actualización de información en tiempo real. Actualiza la información que se almacena en la base de datos en tiempo real, esto implica la interacción indirecta con el proceso de la misma función pero en la aplicación móvil.
- Generación de notificaciones. La fuente que origina las notificaciones es un proceso del sistema web de administración pues es en este donde se generan los cambios en los datos que son desplegados en la aplicación móvil al usuario final.

7. VISTA DE DESPLIEGUE

El despliegue del sistema es descrito por el siguiente diagrama:



8. VISTA DE IMPLEMENTACIÓN

8.1. VISIÓN GENERAL

El sistema de administración de congresos está dividido en dos subsistemas principales. El primero siendo una aplicación móvil con funciones principalmente de despliegue de información, así como otras operaciones pequeñas a nivel local dentro del dispositivo móvil. El segundo es una aplicación web para la edición de los datos mostrados dentro de la aplicación móvil y administración de algunas funcionalidades y características de la misma.

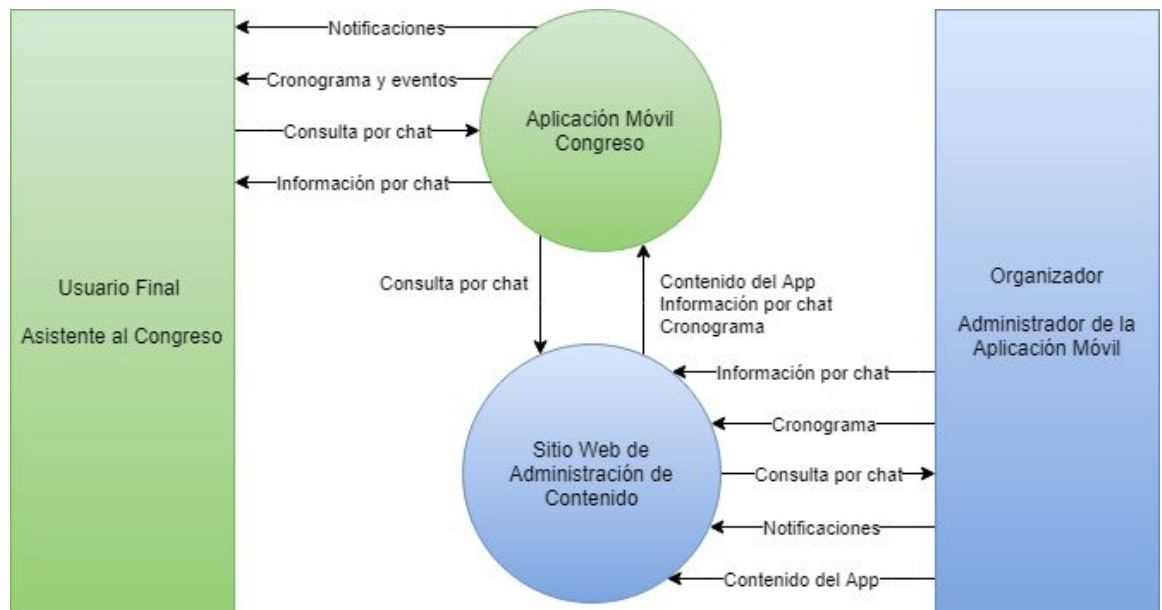
Ambos subsistemas están diseñados en base al patrón de MVC. Por lo cual se han definido tres capas principales correspondientes a dicho patrón. Ambos subsistemas están divididos según estas tres capas pero las características para agruparse dentro de una u otra capa son las mismas sin importar el sistema.

Para la capa de modelo, se incluyen las definiciones de clases y objetos requeridos como parte del diseño de la solución. Esto incluye objetos únicamente para almacenamiento básico de datos, conocidos como clases planas, así como objetos más complejos con métodos requeridos para el correcto flujo del sistema.

Para la capa de vista, se incluyen únicamente los componentes necesarios para el despliegue de información hacia el usuario. Esto incluye todos los recursos de los que hacen disposición las clases y métodos de la capa de controlador, así como las plantillas y esqueletos para pintar información.

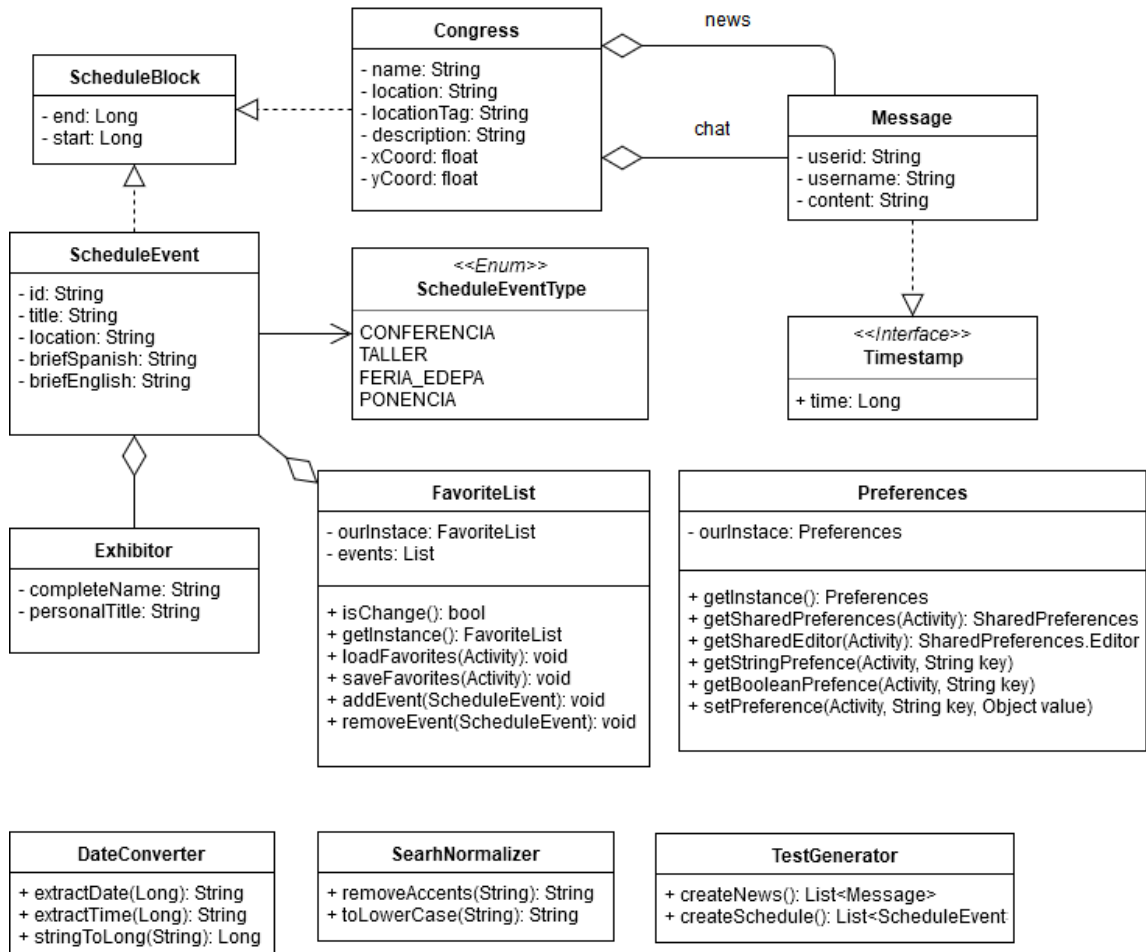
Para la capa de controlador, se incluyen las clases principales de manejo del flujo del sistema. Esto incluye en general la instanciación de los objetos de la capa de modelo, así como la construcción de las vistas y el poblamiento de los contenedores de información. De igual forma esta capa contiene los componentes encargados de realizar cálculos o “tomar decisiones” con respecto al flujo del sistema.

Diagrama de contexto del sistema



8.2. DIAGRAMA DE CLASES

El 75% de las clases de la aplicación móvil está relacionadas con la interfaz de usuario, por lo que se hace exhaustivo realizar un diagrama que represente a todas, por esta razón solo se presenta el diagrama de clases de la capa del modelo:



9. VISTA DE DATOS

El sistema utiliza una base de datos no relacional, Firebase Real Time Database. Al ser una base de datos en la nube que actúa en tiempo real, se restringe la cantidad de información que se puede almacenar, lo cual, no es problema para el sistema en cuestión. A su vez, proporciona una interfaz que permite al usuario tener siempre disponible la información más actualizada, requerido por el cliente.

Tratándose de una base de datos no relacional, un diagrama de bases de datos tradicional no se ajusta a la abstracción que se desea comprender. Por tanto, en el Apéndice A del presente documento, se muestra cómo la información almacenada en un documento con múltiples nodos.

10. TAMAÑO Y RENDIMIENTO

La aplicación móvil tiene un peso aproximado 50 MB. Además, utiliza espacio adicional para guardar información de los eventos, permitiendo de esta forma un modo offline. La información se actualiza una vez que la conexión a internet se restablece.

Como se mencionó anteriormente, la plataforma donde se almacenan los datos se comporta en tiempo real, por lo que posee un límite en la cantidad de usuarios que puede manejar. Esto dependerá de la versión del servicio que sea contratado al momento del despliegue, lo cual, aún está por definir.

El despliegue de la información de cada una de las secciones de la aplicación no está limitada por tiempos específicos, sin embargo, la aplicación es fluida desde la perspectiva del usuario.

11. CALIDAD

Como se describe en la Especificación de Requerimientos de Software, la calidad del sistema se utiliza el estándar internacional para la evaluación de calidad de software ISO/IEC 9126. Entre de los atributos de calidad definidos en este estándar, el sistema prioriza los siguientes: :

1. Correctitud: La implementación del sistema está de acuerdo con requerimientos previamente descritos.
2. Usabilidad: La aplicación es lo suficientemente intuitiva para el usuario.
3. Portabilidad: La aplicación puede ser instalada en más del 80.7% de los teléfonos móviles actuales.

4. Extensibilidad: El sistema está modularizado de tal forma que permite agregar nuevas funcionalidades sin mayores inconvenientes.
5. Mantenibilidad: La cantidad de esfuerzo requerido para conservar su funcionamiento normal o para restituir una vez se ha presentado un evento de falla es baja.

APÉNDICE A: ESTRUCTURA DE LA BASE DE DATOS

Nombre-Congreso

- Chat
 - Identificador-Mensaje
 - Contenido
 - Hora-Envio
 - Id-Usuario
 - Nombre-Usuario
- Config
 - Tema
 - Url-Mapa
- General
 - Nombre
 - Fecha-Inicio
 - Fecha-Fin
 - Descripción
 - Ubicación
 - Edificio
 - Coordenada-X
 - Coordenada-Y
- Noticias
 - Identificador-Noticia
 - Contenido
 - Hora-Envio
- Cronograma
 - Id-Evento
 - Título
 - Fecha-Inicio
 - Fecha-Fin
 - Tipo-Evento
 - Resumen-Español
 - Resumen-Inglés

- Ubicación
- Expositores
 - Id-Expositor
 - País
 - Nombre-Completo

AUTORIZACIÓN

Aprobado por el patrocinador del proyecto:

Greivin Ramírez Arce

Prof. Escuela de Matemáticas del Instituto Tecnológico de Costa Rica

Fecha: _____