

SPEL: Sistema de reporte participativo de eventos
basado en localización

Juan Camilo Ortiz Román

UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y
COMPUTACIÓN

BOGOTÁ, DC

NOVIEMBRE 2014

SPEL: Sistema de reporte Participativo de Eventos basado en
Localización

Juan Camilo Ortiz Román

Proyecto de Grado presentado como requisito para optar al título de Ingeniero de
Sistemas

Director: Claudia L. Jiménez Guarín, Ph.D.

UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

BOGOTÁ, DC

NOVIEMBRE 2014

Tabla de Contenido

1	Introducción.....	6
	Objetivos.....	8
2	Marco Teórico y Estado del Arte	9
2.1	Tecnologías posibilitadoras	9
2.1.1	Tecnologías de localización.....	9
2.1.1.1	Localización por GPS	9
2.1.1.2	Localización por redes inalámbricas.....	9
2.1.1.3	Localización por Bluetooth BLE	10
2.1.2	Sistemas de Información Geográfica	10
2.2	Participatory Sensing y crowdsourcing.....	11
2.3	Representación y recolección sintáctica y semántica de información	13
3	Estrategia de Solución.....	14
3.1	Modelo de Carga de Datos.....	15
3.2	Modelo de Validación de Usuarios	16
4	SPEL: Descripción de la Solución	18
4.1	Atributos de Calidad	18
4.2	Requerimientos Funcionales	20
4.3	Requerimientos No Funcionales	20
4.4	Casos de Uso.....	20
4.5	Arquitectura General	22
4.5.1	Arquitectura de la Aplicación	23
4.5.1.1	Componente GeoLoc	25
4.5.1.2	Componente PushValidation	26
4.5.1.3	Componente BroadcastReceiver	27
4.5.1.4	Componente EventRegistration	27
4.5.2	Arquitectura del Servidor SIEGE.....	27
4.5.2.1	Parser del Feed de Eventos y Eventos generados desde la aplicación a RDF	28

4.5.2.2	Sincronización de Datos con el Servidor de Mapas	29
4.5.3	Arquitectura del Servidor de Mapas.....	30
4.5.4	Arquitectura del Servidor de Validación de Eventos	32
	Comunicación por Google Cloud Messaging	33
4.6	Secuencia de Procesos	34
4.7	Autenticación de Usuarios.....	36
5	Desarrollo, implementación y Pruebas.....	37
5.1	Ambiente de desarrollo.....	37
5.2	Plataforma de Despliegue y Pruebas	37
5.3	Dificultades y problemas más significativos durante la implementación.....	37
5.4	Interfaz Gráfica de la aplicación	38
5.4.1	Interfaz Gráfica del prototipo de gestión de reportes de mantenimiento.....	39
5.4.2	SPEL: Interfaz Gráfica de la aplicación móvil	40
5.5	Estructura del Proyecto Java.....	42
5.6	Plan de Pruebas.....	44
5.6.1	Pruebas Funcionales	44
5.6.1.1	Prueba 1: Registro de un nuevo evento con datos sin formato.....	44
5.6.1.2	Prueba 3: Ubicación del usuario por GPS y Beacons	45
5.6.2	Pruebas No Funcionales	45
5.6.2.1	Prueba 4: Latencia en validación.....	45
5.6.2.2	Prueba 5: Latencia en consultas al servidor de mapas	46
5.6.3	Análisis de Resultados.....	47
6	Conclusiones y Trabajo Futuro.....	49
	Bibliografía	51

Índice de Ilustraciones

Ilustración 1 Contexto del sistema SIEGE y las nuevas implicaciones, modificaciones y oportunidades de SPEL	14
Ilustración 2 Atributos de Calidad Aplicación Móvil	18
Ilustración 3 Atributos de Calidad <i>Backend</i>	19
Ilustración 4 Casos de Uso	21
Ilustración 5 Arquitectura General del Sistema.....	22
Ilustración 6 Arquitectura de la aplicación	23
Ilustración 7 Detalle de la Aplicación sin el módulo GeoLoc	24
Ilustración 8 Componente de localización GeoLoc	25
Ilustración 9 Proceso ETL para actualización de los eventos en SIEGE.....	29
Ilustración 10 Detalle de la ampliación a las tablas incluidas en la base de datos.	31
Ilustración 11 Especificación de Servicios sobre la base de datos geográfica.....	32
Ilustración 12 Esquema de comunicación por GCM	33
Ilustración 13 Diagrama de secuencia del registro y validación de reportes de eventos	35
Ilustración 14 Diagrama de Secuencia de los Servicios de localización por GPS y Beacon	35
Ilustración 15 Diagrama de secuencia de recomendación de eventos.....	36
Ilustración 16 Estructura de archivos del código fuente de la aplicación móvil	42
Ilustración 17 Estructura de recursos de la aplicación móvil.....	42
Ilustración 18 Estructura de librerías y archivos de configuración de la aplicación móvil	43
Ilustración 19 Estructura del proyecto librería Google Play Services utilizado en SPEL	43
Ilustración 20 Estructura del proyecto librería Android Beacon Library utilizado en SPEL	43

Índice de tablas

Tabla 1 Ejemplo de traducción de un evento desde el Feed Institucional a RDF	38
Tabla 2 Documentación de la interfaz del prototipo de gestión de reportes de mantenimiento	40
Tabla 3 Documentación de la interfaz de la aplicación móvil SPEL.....	42

1 Introducción

Los sistemas de recomendación son aquellas herramientas tecnológicas encargadas de entregar contenido relevante a los usuarios en el contexto específico en el que se encuentren. Este objetivo tiene grandes implicaciones y retos de modelaje, tecnología y lógica que hacen de estos sistemas un complejo objeto de estudio. En el campo de aplicación de este tipo de tecnología, también deben considerarse aspectos como la ubicación del usuario en un momento específico y la frecuencia o patrones de uso o de recorrido que ellos puedan tener en un espacio delimitado. Todos estos aspectos constituyen información relevante para la creación de un modelo de recomendación útil y acorde con el contexto para el cual se desarrolla.

SIEGE (Lyons, 2014) es un sistema de recomendación de contenido basado en las preferencias del usuario y su localización en un momento determinado, que fue diseñado y desarrollado por Gastón Lyons en el 2004 para ser implementado en el campus de la Universidad de Los Andes en Bogotá, Colombia. A través de este sistema los usuarios obtienen notificaciones de eventos relacionados con su ubicación y con sus intereses. Este sistema marca el inicio de la solución a una estrategia gerencial de campus activo sobre la Universidad de los Andes, haciendo del espacio físico un ambiente inteligente y activo frente a las necesidades de estudiantes, profesores, personal administrativo y operativo y visitantes de la institución.

En el 2010, el campus de la Universidad de los Andes contaba con más de 99.000 m² de extensión, dentro de los cuales hay 156.000 m² de área construida en 80 edificaciones, 164 salones de clase, 144 laboratorios y 15 salas de cómputo, para un total de 323 espacios en los que no se incluyen áreas de circulación ni áreas comunes (Gerencia del Campus, Universidad de los Andes, 2010). Adicionalmente, la universidad tiene un flujo diario de 14.000 personas entre los que se encuentran estudiantes, personal académico, visitantes y personal administrativo. También cuenta con cerca de 45 eventos oficiales a la semana en donde se incluyen presentaciones corporativas, tertulias con los directivos, seminarios académicos, talleres y encuentros de grupos de investigación.

Aun así, en el contexto de un campus activo, existen decenas de eventos esporádicos que surgen durante la jornada en el campus de la universidad. Estos eventos incluyen desde improvisaciones de grupos culturales estudiantiles, hasta presentaciones espontáneas de varios expertos de diversas disciplinas académicas pasando por todo tipo de eventos categóricos intermedios para los intereses de la comunidad en general. Este tipo de actividades se convierten en ocurrencias de interés colectivo que no son difundidos exitosamente en la comunidad y por lo tanto se pierde el

valor cultural, intelectual e incluso emocional que estas podrían ofrecerle a potenciales usuarios interesados.

Con el apoyo del reporte participativo y de la tecnología de localización de usuarios es posible hacer de estos eventos esporádicos un insumo de importancia para los sistemas de recomendación existentes. En este documento se presenta la inclusión de la generación participativa de eventos en el sistema de recomendación de contenido SIEGE, a través del sistema SPEL (**Sistema de reporte Participativo de Eventos basado en Localización**). Mediante el uso de conceptos propios de colaboración abierta participativa y de sistemas de información geográfica, SPEL busca que los miembros de la Universidad de los Andes pueden generar contenido que es validado por la misma comunidad y que luego pasa a ser distribuido a la población interesada. Todo lo anterior se encuentra enmarcado en un contexto de computación ubicua, no intrusiva y transparente al usuario (Weiser, 1993).

Adicional a la generación de reportes de eventos en el campus por parte de la población uniandina en general, SPEL le permite a la comunidad encargada de la gestión de infraestructura del campus construir y administrar reportes de mantenimiento de la planta física de la universidad, facilitando las labores administrativas relacionadas con su gestión.

El *Capítulo 2* de este documento expone las tecnologías posibilitadoras, el estado del arte y una comparación de soluciones cercanas a esta problemática. En el *Capítulo 3* se presenta una propuesta de solución; el diseño en el *Capítulo 4*. La implementación y pruebas se muestran en el *Capítulo 5*. Finalmente, el Capítulo 6 expone conclusiones del proceso realizado y muestra las posibilidades de un trabajo futuro sobre SPEL.

Objetivos

Objetivo General

Diseñar e implementar una extensión al sistema de recomendación de contenido SIEGE a través de la colaboración participativa de la comunidad de la Universidad de los Andes en la creación y validación de eventos, en un contexto de computación ubicua, no intrusiva y consciente de la localización geográfica de los usuarios.

Objetivos Específicos

- Definir servicios de localización y consulta de posición para la ubicación del usuario utilizando información del campus físico de la Universidad de los Andes.
- Definir un modelo de reputación y validación de usuarios que asegure una alta calidad de la información generada por la comunidad.
- Implementar una aplicación móvil en capacidad de generar reportes de eventos asociados con la posición geográfica en la que se encuentra el usuario.

2 Marco Teórico y Estado del Arte

En este capítulo se presentan las tecnologías exploradas y utilizadas en la elaboración de SPEL así como los referentes más importantes y los trabajos relacionados.

2.1 Tecnologías posibilitadoras

Por la necesidad de ubicuidad y transparencia de las tecnologías para el usuario, SPEL se fundamenta sobre dos tipos de herramientas técnicas: las tecnologías de localización y los sistemas de información geográfica.

2.1.1 Tecnologías de localización

A través de las tecnologías de localización es posible conocer en un momento determinado, la posición geográfica de un dispositivo móvil o estático. Existen diversas técnicas para recuperar esta información. SPEL contempla la utilización del sistema de posicionamiento global GPS, la localización por red inalámbrica local y la utilización de dispositivos Bluetooth de baja intensidad LET para la ubicación a través de un dispositivo móvil.

2.1.1.1 Localización por GPS

EL sistema de posicionamiento global es una tecnología de localización por triangulación. Sus tres componentes principales son los satélites que actúan como emisores, las antenas receptoras y las estaciones de monitoreo y control. Las antenas receptoras, a través de la triangulación de las señales de por lo menos cuatro satélites, calculan su posición en el espacio en términos de la posición de los satélites y la hora de emisión y recepción de las señales que estos envían. (Zogg, 2002)

Según (Zandbergen & Barbeau, 2011) la precisión de estas antenas suele estar entre 5 y 8.5m en espacio abierto y de 30m en espacios cerrados. El auge de los dispositivos móviles ha contribuido a la inclusión de este tipo de antenas desde los dispositivos de gama media hasta los teléfonos de alta categoría lo que contribuye a la ubicuidad de esta tecnología.

2.1.1.2 Localización por redes inalámbricas

La apropiación de la tecnología de red local inalámbrica permite la localización de los dispositivos que se encuentran en el área de influencia de esta red. A través de la intensidad de la señal que está recibiendo la tarjeta de red inalámbrica de estos dispositivos, es posible estimar la distancia a la que se encuentra el usuario de los emisores de la señal de acuerdo con la distribución de probabilidad de las diferentes señales emitidas en el espacio (Ping, Rudys, Ladd, & Wallach, 2003).

Este proceso, conocido como *fingerprinting*¹, consiste en construir una grilla de esta distribución en puntos definidos en el espacio de manera tal que se pueda asociar una ubicación cartesiana específica (x, y, z) a una función de distribución de probabilidad que modela la intensidad de la señal de red. Para que esta estrategia sea efectiva se utilizan operaciones de similitud entre conjuntos pues estas señales no son constantes ya que un dispositivo emisor adicional puede afectar la distribución de la señal en el espacio. Adicionalmente, un cambio en la disposición de los objetos también genera variaciones u obstrucciones de la señal. Por lo tanto la utilización de este método requiere la generación de un modelo de regresión dinámico sobre los conjuntos de dispositivos emisores de señal para así verificar similitud de conjuntos o estados de señal en cada intento de ubicación del dispositivo móvil del usuario.

2.1.1.3 Localización por Bluetooth BLE

El Bluetooth BLE es un protocolo de transmisión de datos en redes de corto alcance. La especificación del Bluetooth 4.0 de bajo consumo de energía permite la generación de enlaces sencillos entre dispositivos. La ubicación del usuario se logra identificar por la medida de la intensidad de la señal en el dispositivo receptor bajo la premisa que esta intensidad es inversamente proporcional al cuadrado de la distancia del foco emisor de la señal. Estrategias similares a la triangulación o similitud de conjuntos han sido implementadas para esta tecnología (Bluetooth SIG, Inc.).

Los *beacons* son dispositivos emisores que utilizan el protocolo Bluetooth BLE para transmitir pequeños paquetes de información a cualquier dispositivo receptor que se encuentre en su radio de alcance. Este alcance depende de las posibles obstrucciones en el espacio y de la interferencia que pueda haber en el espacio; usualmente el alcance va desde 70 cm en *beacons* 'estándar' a 450 metros en dispositivos de mayor alcance. En el mercado existen varias implementaciones, como *Stick 'n Find*, *Gelo* o *iBeacons*, en donde conviven protocolos propietario cerrados y estándares de la industria.

En una red de *beacons*, cada dispositivo contiene una combinación única de dos identificadores *major* y *minor* para ser diferenciado del resto de nodos de la red. Estos identificadores son incluidos en los paquetes de información que los *beacons* emiten y pueden ser recuperados por los dispositivos receptores para identificar el nodo fuente de la información recibida.

2.1.2 Sistemas de Información Geográfica

Un sistema de información geográfica (SIG) es un sistema computacional que permite el manejo y representación de información geoespacial por medio de bases de datos relacionales. En estas

¹ En este documento se utilizan varios términos en inglés por no contar con una traducción apropiada al español.

bases de datos la información espacial se recopila información espacial que puede ser discriminada y representada a manera de capas sobrepuestas, utilizadas para construir vistas de la información relevante para el problema de aplicación. Estas capas le dan un sentido visual a la información relacional que se almacena en las bases de datos geográficas, usualmente cada capa se construye con información que proviene de diferentes fuentes, como el levantamiento topográfico, mapeo de temperatura o densidad poblacional.

El sobreponer las capas construidas a partir de información de diferentes fuentes permite obtener una visión holística de un punto geográfico específico, obteniendo una visión multidimensional de una ubicación espacial (Goodchild, 1991).

Los SIG requieren especial atención en el modelamiento de los datos y han existido diversas aproximaciones sobre los modelos de estructuras de datos para la información tabular, la información vectorial, que representa los puntos, líneas y áreas que construyen un espacio, y la información *raster*, que representa datos teselados en grillas.

Pero la gran utilidad de los SIG es la posibilidad de visualizar información que no se restringe a descripciones espaciales del ambiente. El uso de técnicas estadísticas de procesamiento de datos es lo que permite la visión multidimensional de las locaciones geográficas representadas en el nivel más básico de un SIG. Esto abre la posibilidad de priorizar la información allí presente y de obtener contenido relevante para el contexto del problema o para el usuario al que está dirigida la propuesta de solución (Dempsey, 2012)

2.2 Participatory Sensing y crowdsourcing

La Percepción Participativa (*Participatory Sensing*) es un concepto en el que una comunidad es la encargada de incorporar información a un cuerpo de conocimiento. En ese sentido, el conocimiento de un sistema es generado por medio de la adquisición de información en cada dispositivo individual que luego es recolectada y procesada de forma tal que se le pueda atribuir algún significado relevante para la misma comunidad que la registró o para un grupo de estudio tercero.

Esta metodología ha estado apoyada sobre los dispositivos móviles de última generación y por la conexión ubicua. La gran cantidad de sensores que contienen estos dispositivos y la posibilidad de contar con un alto nivel de conexión para la transmisión de información facilitan la documentación del ambiente por parte de los usuarios, un fenómeno heredado de la gran acogida de la Web 2.0, asociado a la colaboración participativa (*crowdsourcing*) en la que labores o procesos son delegados a los usuarios o colectivos en las comunidades poblacionales.

La Percepción Participativa aumenta significativamente el alcance y cubrimiento que tiene la adquisición de información en un sistema. El permitir que las unidades móviles adquieran los datos necesarios para que un sistema sea alimentado, asegura una expansión del área de cubrimiento del espacio sobre el que se desea capturar. Aunque en muchos casos, la calidad de la información recolectada baja por cuanto no se tienen las especificaciones de gestión de calidad que se tendría en una recolección centralizada.

Según (De Cristofaro & Soriente, 2012) una infraestructura de Percepción Participativa típicamente cuenta con cuatro componentes principales.

- Nodos móviles: son los dispositivos móviles e inalámbricos de los usuarios, con más de un sensor y en capacidad de generar información.
- Consumidores: dispositivos o servicios suscritos a la información registrada por los nodos móviles y a los reportes que de allí puedan ser generados. Puede ser que, en algunos casos, los nodos móviles sean los mismos consumidores de la información que ellos mismos generan.
- Operadores de red: encargados del canal de transmisión física de la información de los sensores por GSM, 3G o 4G.
- Proveedores de servicio: compañías que actúan como intermediarias entre los nodos móviles y los consumidores

Aun así, las grandes posibilidades de esta tecnología han arrojado retos de dimensiones similares. De Cristofaro y Soriente identifican amenazas latentes a la privacidad en estas situaciones en donde los usuarios registran constantemente su posición y sus mediciones. Frente a amenazas latentes como filtrado de información por la transmisión de información existen técnicas para asegurar el canal de comunicación o incluso el cifrado de la misma información. Ellos también identifican amenazas menos evidentes pero igualmente existentes, como la posesión de información por parte de los proveedores de servicios o por terceros en la medida en que el sistema de percepción participativa lo permita.

Cristofaro y Soriente presentan un panorama de disponibilidad de información a agentes externos que debe ser analizado muy minuciosamente al momento de diseñar la aplicación para no comprometer la privacidad del usuario. En ese orden de ideas, es relevante en este tipo de aplicaciones definir qué información que viaja por la red y cuál debe ser retenida en el dispositivo sin ser expuesta a agentes externos.

2.3 Representación y recolección sintáctica y semántica de información

Para la recolección de información relevante contenida en texto plano se cuenta con dos aproximaciones diferentes, una sintáctica y otra semántica. La recolección sintáctica se vale de elementos como el etiquetado, las palabras clave o metadata sobre el texto para obtener aquellos elementos sintácticos que dan sentido a una frase u oración, que son aprovechados por los sistemas de información para modelar comportamientos, entidades o procesos de acuerdo a lo definido por la sintaxis extraída. Esta aproximación da lugar a estándares de representación como *XML* o *JSON*, que estructura la información de acuerdo a marcadores y etiquetas presentes en los textos.

La otra aproximación es por una estrategia semántica en donde, herramientas de procesamiento de texto como Open Calais, realizan una clasificación automática de frases o párrafos por medio de sus unidades estructurales. Así, es posible obtener las grandes categorías sobre las que se emite algún juicio o enunciado de valor en el texto procesado.

El estándar *Resource Description Framework*, utilizado por SIEGE, permite la representación semántica de la información procesada. *RDF* estructura esta información en tripletas de la forma <sujeito><predicado><objeto>. En donde se representa una relación de tipo predicado entre el sujeto y el objeto de la tripleta.

Con esta ontología, o modelo del conocimiento, es posible generar nueva información con un motor de inferencias aprovechando las propiedades transitivas de la estructura en tripletas. Así mismo, la facilidad de representación facilita la traducción de entidades entre lenguajes como *JSON* o *XML* al estándar *RDF*, lo que resulta conveniente en sistemas multilinguaje.

3 Estrategia de Solución

SPEL se construye sobre la base tecnológica expuesta por Gastón Lyons en (Lyons, 2014), por lo cual la estrategia planteada responde a las restricciones, limitaciones y requerimientos de lo establecido en SIEGE. Aun así, se contemplan algunas extensiones y eventuales modificaciones, en la medida en que los objetivos de ambos proyectos así concuerden.

En ese sentido, el diagrama general de arquitectura presentado extiende las funcionalidades del sistema sin necesidad de recompilar sus componentes. Es por esto que la arquitectura del sistema se articula en una aproximación orientada a servicios, para así facilitar su acoplamiento con otros sistemas y extender las funcionalidades que otros proyectos puedan llegar a tener o necesitar.

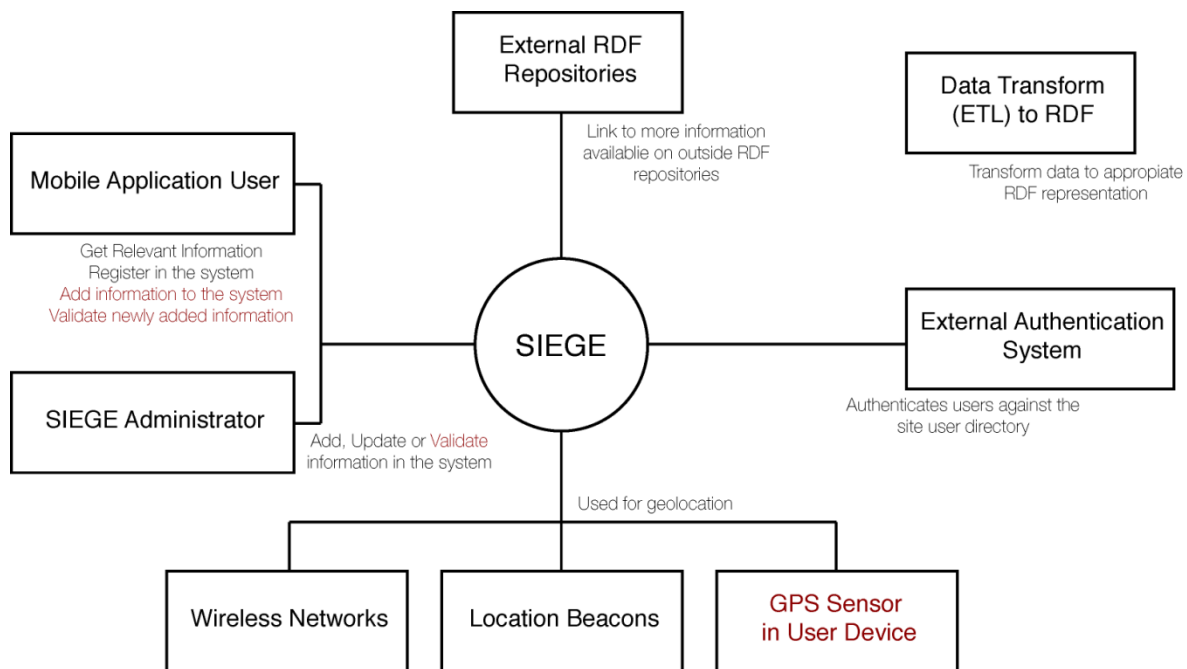


Ilustración 1 Contexto del sistema SIEGE y las nuevas implicaciones, modificaciones y oportunidades de SPEL

La estructura general se divide en funcionalidades de localización y funcionalidades de registro de eventos. Desde el dispositivo móvil, la aplicación realiza la petición a los servicios *RESTful* del servidor de mapas sobre la localización geográfica del usuario en el campus de la universidad. Una vez obtenida esta ubicación, el usuario está en capacidad de generar eventos asociados a esa posición. Desde la aplicación se realiza el reporte del evento que es traducido a lenguaje RDF por un *parser* para ser luego introducido a SIEGE y, de esta manera, alimentar el sistema de recomendación.

Posteriormente, a través de colaboración participativa de los usuarios que se encuentran cerca a la ubicación desde la que es generada el reporte, se valida la existencia del evento. Finalmente, al ubicar a un usuario que se encuentre circulando por la ubicación desde la que fue generado el reporte de evento, si dentro de los intereses de ese usuario se encuentra el tema del evento ya validado, el usuario recibe en su dispositivo la recomendación de este nuevo evento.

Si bien la aplicación móvil está en capacidad de obtener datos geográficos del GPS, los *beacons* ubicados en el campus y las redes WiFi existentes en la universidad, estos datos deben ser contextualizados en información útil para el usuario y para el problema. Por ejemplo, en el caso del sensor GPS, el dispositivo móvil recibe sus coordenadas de latitud y longitud pero estas deben ser llevadas a la especificación de un bloque, edificio o salón en la universidad para ser entendidas por el usuario. Esta labor la realiza el servidor de mapas, la aplicación actúa como un colector de datos que el servicio *RESTful* del servidor debe interpretar y emitir el bloque o edificio correspondiente de acuerdo al nivel de precisión y granularidad que cada una de las tecnologías de ubicación permite.

3.1 Modelo de Carga de Datos

Inicialmente, SIEGE realizaba una carga de eventos definidos por un administrador del sistema desde diversas fuentes de información que incluyen información de los usuarios e información de eventos oficiales administrados por la oficina de Relaciones Públicas de la Universidad de los Andes. Este proceso se realizaba antes de ejecutar la instancia del servidor que atiende las solicitudes de los dispositivos móviles para la recomendación de contenido.

Con la propuesta presentada en este documento, el usuario puede generar reportes de eventos desde la aplicación, que alimentan el sistema de recomendación SIEGE. Bajo esta premisa es necesario establecer un modelo de carga de los datos que se generan en la aplicación hacia el servidor. A través de la traducción de los eventos generados a un formato *RDF* utilizable por SIEGE, es posible introducir la información de los eventos generados por los usuarios para alimentar el sistema.

De igual forma, al contemplar la utilización de un servidor de mapas que también persiste información relacionada con los eventos de un espacio, es necesario contar con un esquema de carga, limpieza, extracción y sincronización entre los eventos registrados en SIEGE y los eventos registrados en el servidor de mapas para evitar inconsistencia en el estado del sistema.

El proceso de extracción de la información se realiza desde la fuente oficial de eventos de la universidad. A través de la consulta del *feed* se extraen los metadatos que lo construyen e inicialmente se obtiene la información de los eventos expuestos en ese *feed* de acuerdo al modelo

de la ontología RDF en SIEGE en donde se contemplan el título, la descripción, la locación, la hora de inicio y la hora de finalización del evento.

Con esta información se construye la entidad evento en el servidor SIEGE. Actualmente, la estructura de la información del evento en el *feed* sólo contiene la descripción y el título del evento. La locación, la hora de finalización y la hora de inicio están contenidas en la descripción. Por lo tanto, se realiza un procesamiento sintáctico de este campo para extraer dicha información. Las heurísticas de extracción están definidas por las palabras precedentes a la información relevante. Es así como la extracción de la hora de inicio y la hora de finalización se obtiene mediante la búsqueda de la expresión regular de la ecuación 1.

$$'[0-9]{2}[V][0-9]{2}[V][0-9]{4} ([0-9]+:[0-9]{2} [A|P]M)?' \quad (1)$$

La identificación de la locación se realiza mediante la aplicación de la expresión regular en la ecuación 2.

$$'Lugar: ((([A-Za-z]*([]+)[A-Za-z0-9]+){1,2})|(((salón|Salón)? *([a-zA-Z]+)[- _][0-9]+)))' \quad (2)$$

Posteriormente, se deben obtener las categorías temáticas del evento. Para esto se utiliza la librería *OpenCalais*, que procesa semánticamente la descripción y establece los temas sobre los cuales se trata para incluir esta información en el modelo ontológico de SIEGE. Finalmente se realiza la escritura de esta información en un archivo en formato *RDF* del cual se alimenta el motor de inferencias para poblar el sistema de recomendación.

3.2 Modelo de Validación de Usuarios

Uno de los grandes inconvenientes de la colaboración activa y la percepción participativa es la disminución en la calidad de los datos por la introducción de datos 'basura' o la inadecuada utilización de la plataforma por parte de los usuarios. Para mitigar los inconvenientes que se puedan presentar en este aspecto SPEL cuenta con un modelo de reputación de usuarios en donde se atribuye una calificación al comportamiento de registro de eventos de un usuario a través de una validación con aproximación participativa. Cada vez que un usuario registra un evento, el sistema identifica a los usuarios que se encuentran próximos a la locación desde donde fue generado el evento y solicita la confirmación de la existencia de dicho evento.

La confirmación o negación de la existencia del evento, así como la cantidad total de eventos reportados por el usuario, influyen en la calificación generada sobre el usuario. Esta calificación es persistida localmente en el dispositivo pero es incluida junto con la firma del usuario en el momento

en que este genera un evento. Esto con el fin de notificar al sistema de la credibilidad de dicho usuario y proceder acordemente. Si el usuario tiene poca credibilidad, esto les será informado a los demás usuarios al momento de realizar la recomendación del evento.

Esta medida de credibilidad del usuario está definida por la fórmula 3.

$$C_u = \frac{\sum_{i=0}^N C_{ri}}{N} \quad (3)$$

En donde C_u es la calificación acumulada del usuario, N es la cantidad total de reportes realizados por el usuario y C_{ri} es la calificación individual de cada uno de los reportes generados por ese usuario. La calificación de cada reporte está expresada en la fórmula 4.

$$C_r = \frac{(\#confirmados - \#negados)}{total\ respuestas} \quad (4)$$

La calificación de cada reporte individual depende del número de usuarios que validar el evento como confirmado y el número de usuario que lo validaron como erróneo. Siendo así, la calificación del reporte se encuentra en el intervalo definido en la fórmula 5.

$$-1 \leq C_r \leq 1 \quad (5)$$

De esta forma se maneja una calificación acumulativa luego de cada reporte realizado en donde el cálculo luego de reportar el evento es casi inmediato y sólo depende de la prontitud en la validación de los usuarios que se encuentran cerca al lugar. Si la calificación del usuario C_u disminuye y llega a ser menor a un umbral K , el usuario queda inhabilitado para seguir reportando eventos, aunque recibe las recomendaciones generadas por el sistema o por otros usuarios. Para asegurarse que la penalización sea por errores en la generación de reportes por parte del usuario y no por falta de usuarios validadores, este umbral K debe ser menor a 0.

Para que el usuario pueda volver a usar el sistema el sistema debe contar con una función de perdón que depende exclusivamente del tiempo transcurrido desde la penalización del usuario. Luego de transcurrido este tiempo el usuario puede volver a realizar recomendaciones y su calificación acumulada, así como el conteo de los reportes que entran a afectar su calificación, vuelve a ser 0.

$$EPerdon_{u,p,i} = \begin{cases} f_i = 1 \\ f_{i+k} = 0, \\ f_{i+p} = 1 \end{cases}, \quad 0 < k < p \quad (6)$$

La fórmula 6 expresa la ecuación de perdón para el usuario u , con un lapso de penalización p , a partir del instante i en el que es penalizado. $EPerdon_{u,p,i}$ es un valor binario que define si el usuario está habilitado para realizar reportes dado un tiempo k luego del instante i .

4 SPEL: Descripción de la Solución

En este capítulo se presenta la especificación de la solución planteada con base en la identificación de requerimientos funcionales y no funcionales y los roles de usuario en el contexto del problema.

El sistema se integra con las soluciones existentes de recomendación de contenido, SIEGE y (Melani, 2014), y de ubicación y densidad de usuario en (Rojas, 2014) y (Díaz, 2004). SPEL consume servicios expuestos por Rojas y Díaz sobre las tablas de registro de las ubicaciones compartidas por los usuarios y la visualización del mapa de densidad y aporta a la solución expuesta por Melani en la generación de eventos por parte de la comunidad.

4.1 Atributos de Calidad

A partir de la problemática planteada y el contexto en el cual se desarrolla SPEL, se definen los siguientes atributos de calidad para la aplicación móvil y los servicios *backend* que componen SPEL. En las ilustraciones 2 y 3 se muestran los atributos de calidad de la aplicación móvil y los servidores respectivamente. Se muestran los atributos de calidad relevantes para el contexto del problema, las subcategorías críticas en el diseño del sistema y las tácticas de diseño que se implementan para asegurar los atributos de calidad seleccionados. Los atributos presentados en este documento son los avalados por el estándar IEEE 1061 Extendido (IEEE Computer Society, 2009).

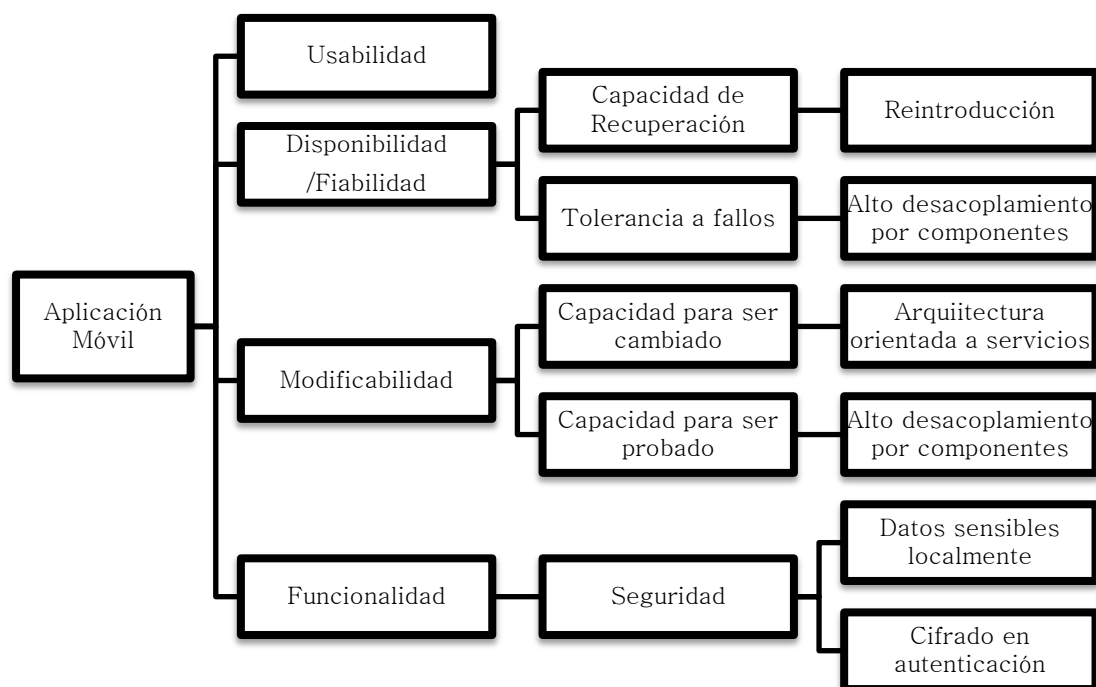


Ilustración 2 Atributos de Calidad Aplicación Móvil

El contexto *pervasive* en el que se encuentra SPEL implica el diseño e implementación de sistemas con una interacción intuitiva y sencilla, casi transparente al usuario. Es por esto que el atributo de calidad más relevante en el diseño de la aplicación móvil es la Usabilidad y la Fiabilidad. Al ser SPEL un sistema desarrollado en un contexto académico de investigación en soluciones basadas en localización, la modificabilidad del sistema es altamente deseable. Otro atributo importante en la aplicación es el manejo de la información privada, sensible, como las credenciales de acceso a los sistemas de información de la Universidad de los Andes y la ubicación del usuario en un momento específico. Para esto el diseño no contempla la transmisión de datos del usuario desde el dispositivo hacia un servidor remoto, el único dato sensible que es enviado al servidor de autenticación es la contraseña del usuario para la autenticación por LDAP por lo cual se utiliza una llave simétrica sólo disponible en el dispositivo del usuario y en el servidor.

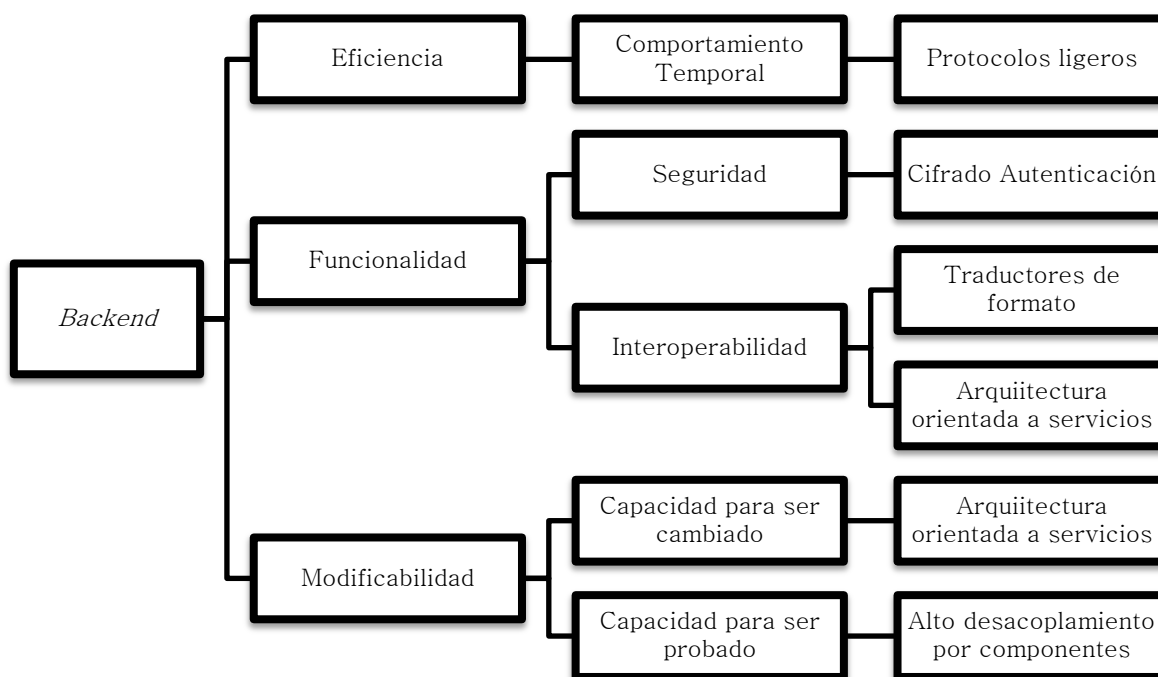


Ilustración 3 Atributos de Calidad *Backend*

Los servidores de SPEL requieren de un corto tiempo de respuesta de las peticiones realizadas desde la aplicación móvil, por lo tanto se utilizan protocolos de comunicación que intercambian mensajes ligeros, sin multimedia asociada. La interoperabilidad con los otros sistemas es necesaria, lo que se logra, no sólo a través de la arquitectura orientada a servicios, sino también por medio de la existencia de traductores de formato *parser* en las comunicaciones que así lo requieren.

4.2 Requerimientos Funcionales

- El sistema debe determinar la ubicación geográfica del usuario con una de 10 metros radiales de su ubicación real.
- El sistema debe permitirle al usuario registrar un evento asociado a la posición geográfica en la que se encuentra en el horario de funcionamiento normal del campus de la universidad, que actualmente es de 6:00am a 10:00 pm.
- El sistema debe permitir la validación colaborativa de varios usuarios sobre los eventos generados en el campus para asegurar la calidad de la información reportada.

4.3 Requerimientos No Funcionales

- El sistema debe ser escalable para atender las múltiples solicitudes y registros que pueden darse por parte de la comunidad uniandina que puede llegar a utilizar la aplicación.
- El sistema debe ofrecer y consumir funcionalidades externas por lo que estará diseñado en una arquitectura orientada a servicios sobre la plataforma Android, que comprende la mayor cantidad de usuarios en el mercado.

4.4 Casos de Uso

Se identifican tres roles diferentes para este sistema:

- Visitante: usuario que no hace parte de la comunidad de los Andes pero que, por invitación de algún miembro de la comunidad, ingresa al campus. Este usuario debe estar en capacidad de consumir el contenido generado por los miembros de la comunidad.
- Usuario de la comunidad: es un miembro activo de la comunidad (profesor, estudiante o trabajador de acuerdo a la definición dada en (Lyons, 2014)). Este usuario debe estar en capacidad de generar, validar y consumir contenido de acuerdo a sus intereses ya su locación.
- Administrador del sistema: este rol está encargado del mantenimiento del sistema y de la actualización de la información de espacios del campus y de eventos oficiales de Relaciones Públicas. Este rol también es el encargado de correr el motor de inferencias para ingresar la nueva información al sistema de recomendación SIEGE.

Los casos de uso considerados para SPEL, para los roles definidos previamente, están representados en la ilustración 2.

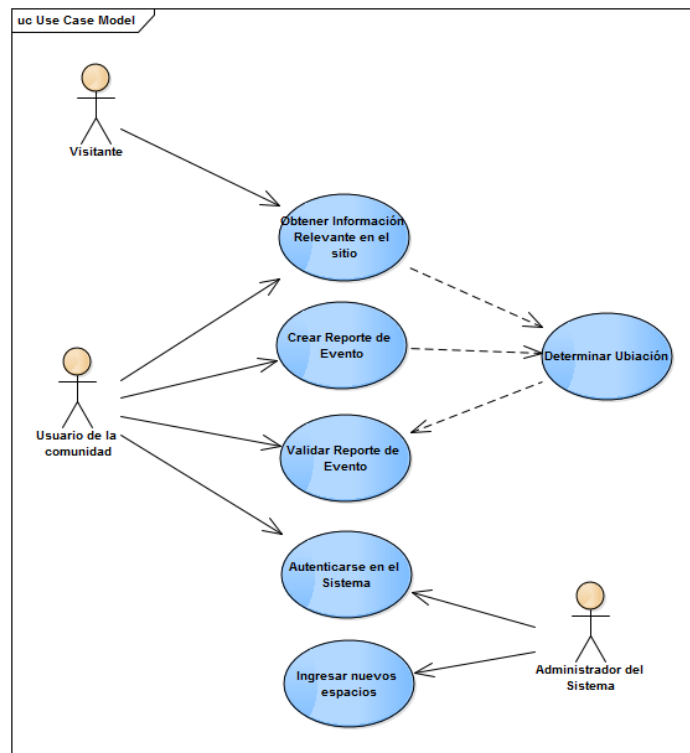


Ilustración 4 Casos de Uso

- **Crear Reporte de Evento:** Permite crear a un usuario que hace parte de la comunidad un evento asociado al lugar en el que se encuentra. El sistema obtiene su locación geográfica y recibe el evento reportado por el usuario.
- **Validar Reporte de Evento:** una vez generado un reporte de evento, este caso de uso permite realizar la validación del evento registrado. El sistema recibe el evento generado y remite su validación a los usuarios cuyo último registro de ubicación es cercano al sitio desde donde fue registrado el nuevo evento. Los usuarios envían su respuesta a la solicitud y el sistema actualiza el estado del evento en el servidor.
- **Obtener Información Relevante en el sitio:** para un usuario de la comunidad o visitante, el sistema detecta su ubicación geográfica y le recomienda contenido de su interés asociado al sitio geográfico en el que se encuentra.
- **Ingresar nuevos Espacios:** el administrador del sistema puede actualizar la información del campus de la universidad agregando nuevos espacios o actualizando los ya existentes.
- **Determinar Ubicación:** el sistema, a través de la posición GPS del dispositivo del usuario y de la red de *beacons* ubicados en el campus, determina la ubicación geográfica del usuario.
- **Autenticarse en el sistema:** permite a los usuarios de la comunidad y al administrador del sistema ingresar al sistema para que, luego de la validación de sus credenciales, sean

capaces de generar contenido o de realizar modificaciones de acuerdo con su rol de usuario.

4.5 Arquitectura General

SPEL está diseñado bajo una arquitectura orientada a servicios, para lo cual se identificaron tres servidores diferentes: SIEGE, un servidor de mapas que contiene la base de datos geográfica de la Universidad de los Andes y expone e servicios sobre los datos ahí almacenados, y un servidor de validación de eventos que también estará encargado de la sincronización de datos entre los servidores.

El otro macrocomponente importante para el sistema es la aplicación móvil desde la cual se obtienen datos geográficos del dispositivo y que actúa como terminal de interacción de los usuarios con las funcionalidades del sistema.

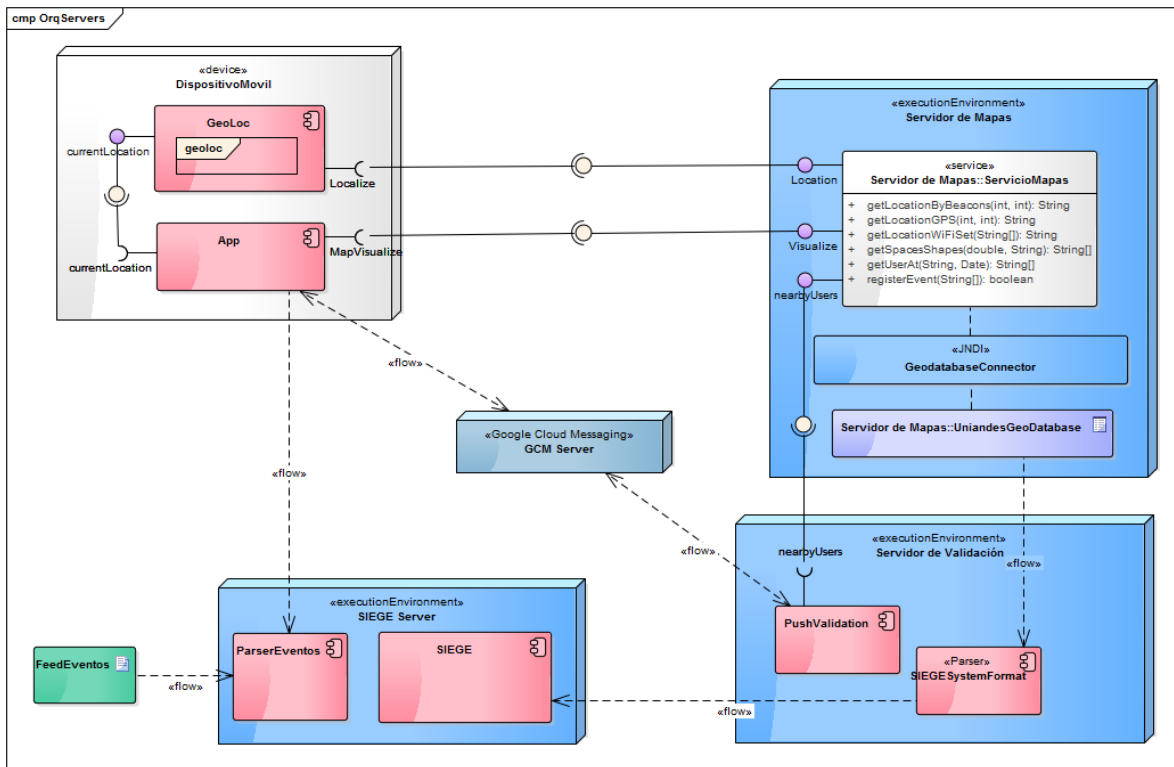


Ilustración 5 Arquitectura General del Sistema

Para este caso concreto se considera la utilización de dos artefactos externos a los servidores y a la aplicación. El *feed* de eventos, que constituye una fuente de información importante en el proceso, cuyo papel será descrito a cabalidad en la sección 4.5.2.1, y el servidor de servicios de Google Inc. *Google Cloud Messaging*, utilizado en la comunicación entre el componente de validación de

eventos en la aplicación y el servidor correspondiente. El uso de este servicio es abordado en la sección 4.4.4.1.

4.5.1 Arquitectura de la Aplicación

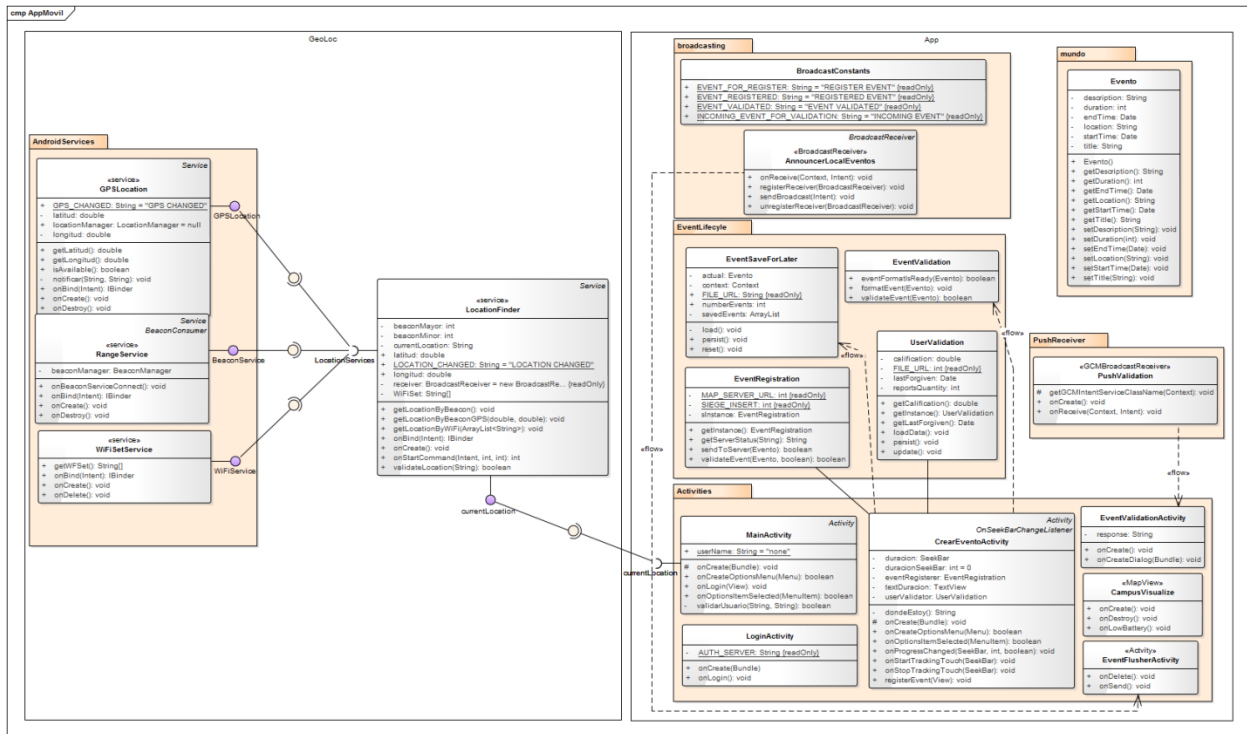


Ilustración 6 Arquitectura de la aplicación

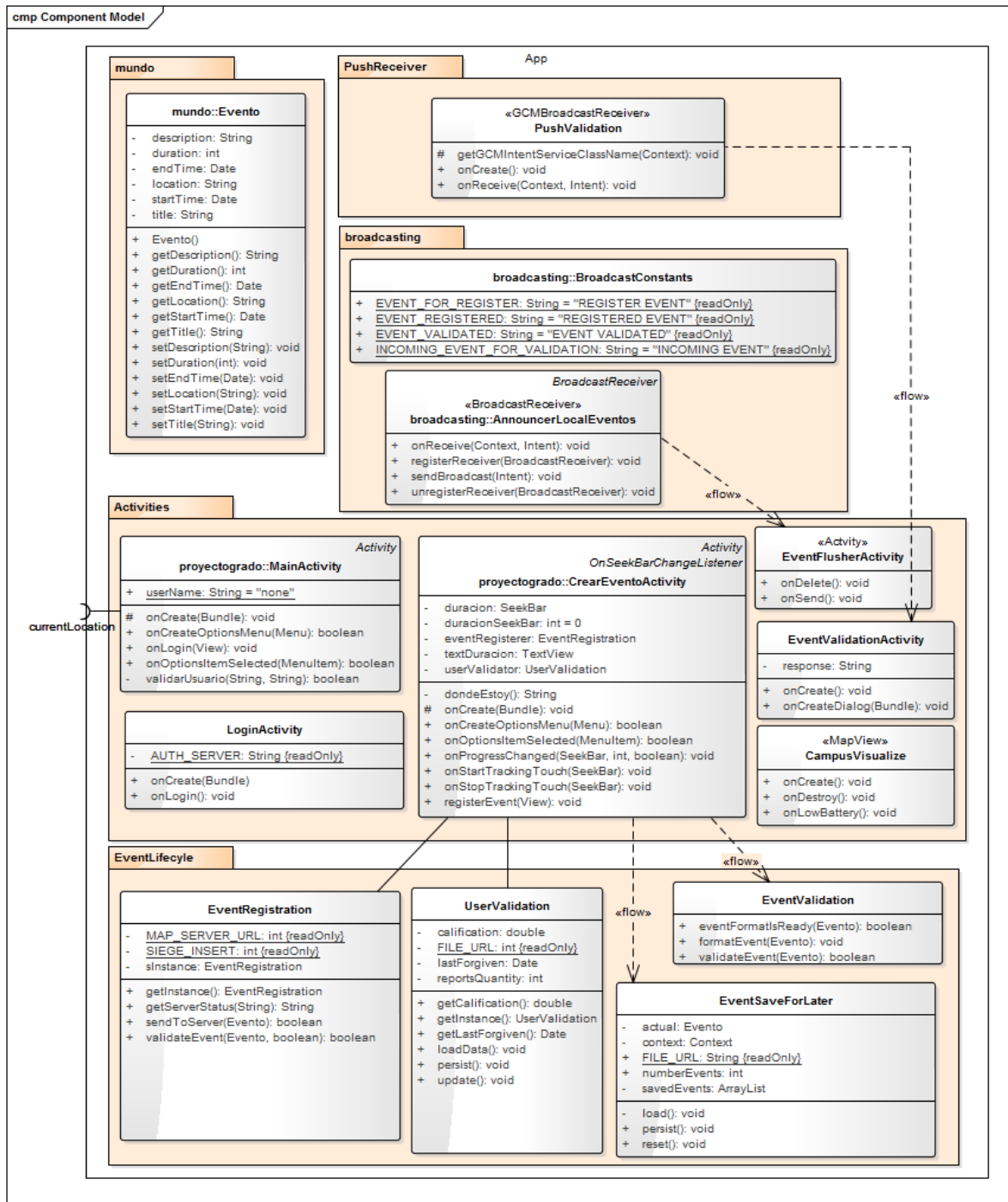


Ilustración 7 Detalle de la Aplicación sin el módulo GeoLoc

La aplicación está diseñada con una arquitectura orientada a servicios con tres componentes principales adicionales a las actividades propias de la aplicación. El componente `GeoLoc` es el encargado de manejar los sensores de localización del dispositivo y, a través de servicios, recuperar la ubicación del usuario en el campus de la universidad. El componente `PushValidation` es el encargado de realizar la comunicación con el servidor de validación de los eventos creados por

otros usuarios. Finalmente, el componente `BroadcastReceiver` es el encargado de redirigir las peticiones y manejar el flujo de eventos dentro de la aplicación y la conexión entre ella y otros eventos recibidos.

4.5.1.1 Componente GeoLoc

El componente `GeoLoc` consta de cuatro servicios principales orquestados para responder a la necesidad de ubicar al usuario. Este componente realiza la obtención de los datos de los sensores de localización, traduce esos datos a un contexto particular en el campus y le da a la aplicación la respuesta del contexto con el nivel de precisión asociado.

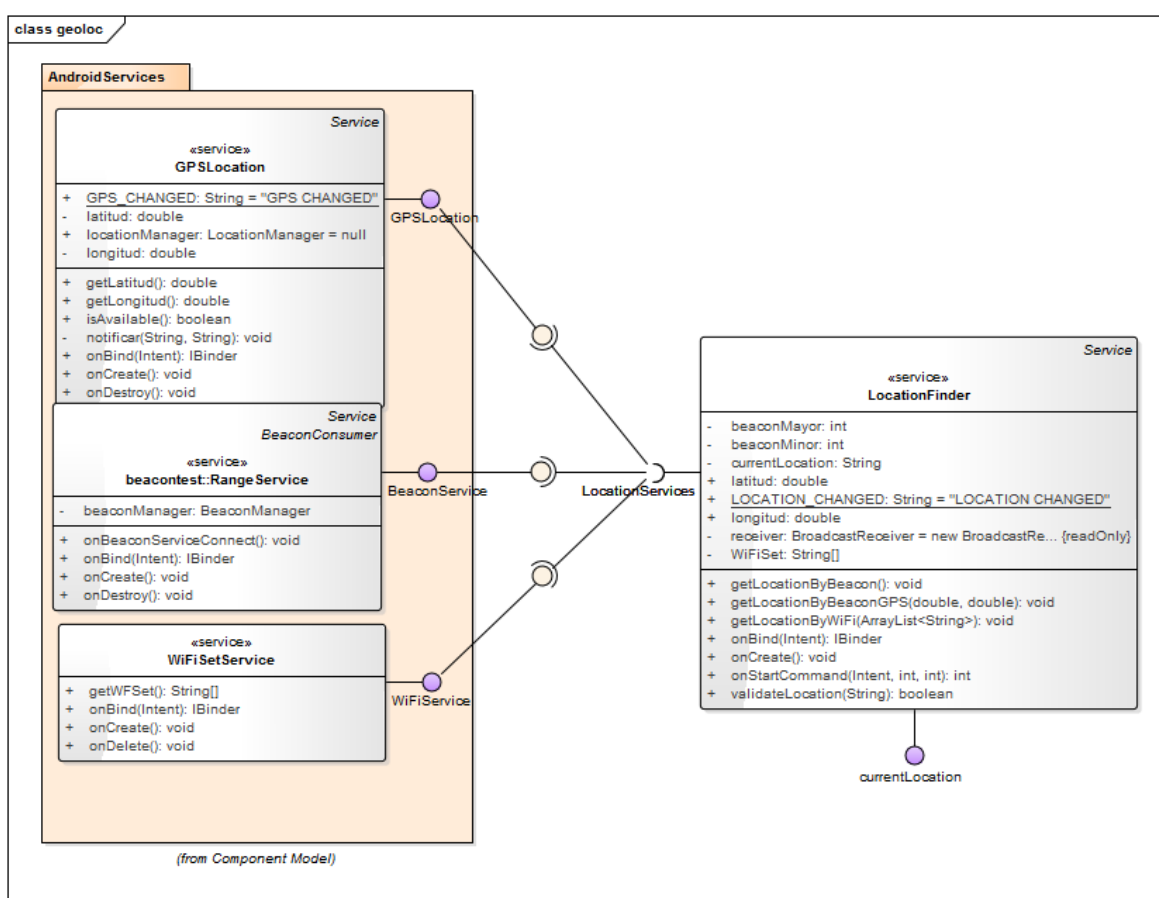


Ilustración 8 Componente de localización GeoLoc

La clase `LocationFinder` es realiza la orquestación de las clases `GPSLocation`, `RangingService` y `WiFiLocationService` encargadas de la obtención de los datos GPS, *iBeacons* y Redes WiFi correspondientemente. `LocationFinder` verifica la disponibilidad de estos sensores y servicios y realiza la consulta en el servidor de mapas de acuerdo a los datos registrados

por las tres clases de localización. Con esto el servidor traduce esos datos de bajo nivel a un lugar concreto en el campus de la universidad. Al obtener la respuesta, `LocationFinder` es capaz de asociar un nivel de precisión dependiendo de las condiciones en las que se encuentre el dispositivo (en espacio abierto, intensidad de la señal, entre otras) e informar el lugar y el nivel de precisión asociado.

`GPSLocation` hace uso de la antena GPS del dispositivo para registrar cambios en la posición geográfica del usuario. Esta clase implementa un `LocationListener` para obtener cambios de posición mayores a una medida de distancia o de tiempo determinadas. Para el caso de este proyecto, la medida de interés es el cambio en la distancia, pues esto indica que el usuario tuvo un desplazamiento y que por ende puede encontrarse cerca de una zona o edificio con un conjunto diferente de eventos asociados.

`RangingService` hace uso de la librería (Young, 2012) mediante la cual es posible obtener un conjunto de *beacons* cercanos al dispositivo y, a través de la intensidad de la señal recibida por cada uno de estos *beacons*, la clase establece cuál es el más cercano y lo reporta consecuentemente al `LocationFinder`.

`WiFiLocationService` es la clase encargada de construir el conjunto de direcciones *MAC* de las antenas emisoras WiFi que están siendo observadas por el dispositivo. Esta clase reporta el conjunto construido a `LocationFinder` quien envía dicha información al servidor de mapas en donde un algoritmo de semejanza de conjuntos devuelve el posible lugar en el que se encuentra el usuario con un porcentaje de precisión calculado a partir de la medida de semejanza. Esta aproximación se explica con más detalle en la tesis de Arturo Henao (Henao, 2013).

En escenarios de no conectividad no es posible obtener la ubicación actualizada del usuario, para lo cual la clase `LocationFinder` maneja un caché de los últimos datos recibidos por cada uno de los servicios de localización. De esta forma, el usuario está en capacidad de generar reportes y editar la locación para hacerla coincidir con su lugar actual de reporte.

4.5.1.2 Componente PushValidation

Este componente es el encargado de recibir las peticiones generadas desde el servidor a través de *Google Cloud Messaging* (GCM) para la validación de los eventos generados por otros usuarios. El componente está compuesto por la clase `PushReceiver` y la clase `EventValidation`.

`PushReceiver` es el encargado de recibir los eventos provenientes de GCM que deben ser validados. Una vez recibido el evento, `PushReceiver` lo envía a `EventValidation` que es la

clase encargada de la interacción con el usuario. Esta clase genera el diálogo de validación del evento y retorna la respuesta a `PushValidation` quien la remite al servidor GCM.

4.5.1.3 Componente *BroadcastReceiver*

Por la arquitectura de servicios planteada, `BroadcastReceiver` es el componente encargado de realizar la orquestación correspondiente entre los diferentes servicios y funcionalidades de la aplicación. La orquestación y comunicación entre los diferentes componentes se realiza a través de *broadcast* locales y externos (en la comunicación con el servidor de mapas, el servidor GCM y SIEGE).

Los mensajes de los servidores son recibidos por el componente encargado de procesarlo debido a que la funcionalidad de cada servicio individual depende fuertemente de la respuesta generada por los servidores por lo que no es recomendable desasociarlos de dichos componentes. Además, el juntar todos los *broadcast* externos e internos en un solo componente con un solo proceso lo convierte en un punto único de falla.

`BroadcastReceiver` procesa los diversos mensajes publicados por los componentes de la aplicación y los sensores del dispositivo para remitir la respuesta o solicitud al componente que la necesita.

4.5.1.4 Componente *EventRegistration*

El componente `EventRegistration` es el encargado de darle formato a los reportes creados por el usuario. `EventRegistration` realiza la conexión al servidor SIEGE y al servidor de validación para la persistencia y validación de los eventos creados por el usuario. Este incluye los módulos de calificación de usuario en donde realiza el procesamiento local del modelo de validación descrito en la sección 3.2.

En escenarios de no conectividad, `EventRegistration` maneja un módulo llamado `EventSaveForLater`, que persiste el evento en el dispositivo del usuario y, cuando el dispositivo vuelve a tener conectividad de red, le permite al usuario enviar o suprimir los eventos que quedaron guardados en la aplicación.

4.5.2 Arquitectura del Servidor SIEGE

SIEGE cuenta ya con una arquitectura definida sobre la cual no se realizan modificaciones mayores. El reto reside en la integración con dicha plataforma. Para esto se realiza la modificación del método de carga de datos a SIEGE y de sincronización con la información del servidor de mapas, que será presentado en la sección 4.5.4. La mayor modificación realizada sobre SIEGE para el acoplamiento de las plataformas está descrita en la sección 4.5.2.1.

La comunicación entre la aplicación en el dispositivo móvil y SIEGE se realiza a través de peticiones HTTP de tipo POST al servidor php en donde se encuentran alojados los scripts de SIEGE y el *parser* de eventos. La comunicación con este punto es unidireccional por lo que sólo se plantea el envío de la información desde la aplicación sin ninguna respuesta emitida desde el servidor. El script que recibe el evento enviado realiza su traducción para que esta información resida en el archivo de carga del motor de inferencias, como será expuesto en la siguiente sección.

4.5.2.1 *Parser del Feed de Eventos y Eventos generados desde la aplicación a RDF*

Originalmente, SIEGE carga los datos a través de la extracción de los eventos registrados por Relación Públicas en el portal web. A través de una modificación al *parser* de carga de eventos se extiende la funcionalidad para agregar eventos generados desde la aplicación en los dispositivos móviles de los usuarios.

Este *parser* traduce los datos en XML a un formato RDF del que se alimenta el motor de inferencias de SIEGE para construir el contenido de recomendación que es utilizado por el sistema. La información de ubicación y fecha del evento debe extraerse de la descripción de este en el formato XML por cuanto se ha modificado el formato del *feed* y ya no es posible extraerlo directamente de las etiquetas XML. Adicionalmente, se utiliza la librería *OpenCalais* para extraer los temas de interés del evento y utilizar esta información en la recomendación de contenido por parte de SIEGE. *OpenCalais* extrae las entidades del texto por medio de procesamiento de lenguaje natural y enriquecimiento semántico.

Existe una limitación a la ejecución del motor de inferencias. Por restricciones tecnológicas, el resultado de la ejecución de este motor no puede ser utilizado mientras el sistema de recomendación se encuentre activo. Es decir, para que las inferencias producidas por el motor tengan efecto, el sistema de recomendación debe volverlas a cargar. Por lo tanto, para ingresar nuevos datos al sistema de recomendación la ejecución debe detenerse.

En el contexto del problema, esto es una desventaja considerable pues los eventos generados por los usuarios se envían en tiempo real y el sistema debe estar en capacidad de realizar recomendaciones inmediatas de este nuevo contenido a otros usuarios una vez ha sido validado.

Para afrontar esta situación se desarrollan dos soluciones paralelas. En primer lugar, conscientes de que el reporte de eventos es mayor en horas de alta fluidez y movilidad en el campus (cambios de clase y hora de almuerzo), se propone la realización de carga de estos eventos en lapsos acordes a estos movimientos. De esta forma, el sistema pararía su ejecución luego de unos minutos de recibir los eventos en las horas de cambio de clase (a saber, 8:30, 10:00, 11:30, 13:00, 14:00, 15:30, 17:00 y 18:30). Al transcurrir 15 minutos después de esas horas, la actividad en el campus

disminuye y por lo tanto un administrador del sistema puede detener la ejecución de SIEGE y cargar los nuevos datos, proceso que no toma más de 5 minutos. El reinicio del servidor por parte de un administrador del sistema no es un proceso complejo, todo lo contrario, por medio de comandos sencillos en la consola de administrador este proceso no toma más de dos minutos. Durante el tiempo que el servidor SIEGE no está disponible, el usuario está en capacidad de generar los eventos de la aplicación pero, como en el caso de no conectividad, estos son persistidos localmente en el dispositivo mientras el servidor vuelve a estar en línea.

Lo que es crucial que se realice casi inmediatamente después de reportado un nuevo evento es la validación por parte de usuarios cercanos al lugar desde donde es generado. Por esta razón se delega esta responsabilidad al servidor de validación de eventos.

4.5.2.2 Sincronización de Datos con el Servidor de Mapas

Debido a los requerimientos de los proyectos de (Rojas, 2014) y (Díaz, 2004) es necesario mantener un registro de la presencia de los usuarios que han compartido su ubicación a una hora determinada. En el contexto del problema abordado en este documento, esto es útil para la ubicación de usuarios en capacidad de valorar un evento reportado sobre una ubicación. De esta manera, es posible identificar los usuarios que se encuentran a una pequeña distancia de la ubicación en donde transcurre el evento y pueden ser ellos los que verifiquen y validen que el evento efectivamente está ocurriendo.

En esta medida es importante realizar una sincronización entre los eventos registrados en SIEGE y los eventos y ocurrencias que se registran en el Servidor de Mapas. La integridad de la información asegura una correcta visualización y extracción para los casos abordados por Díaz y Rojas y mantiene la consistencia en el sistema.

El siguiente diagrama muestra el proceso ETL que se realiza sobre estas fuentes y su información:

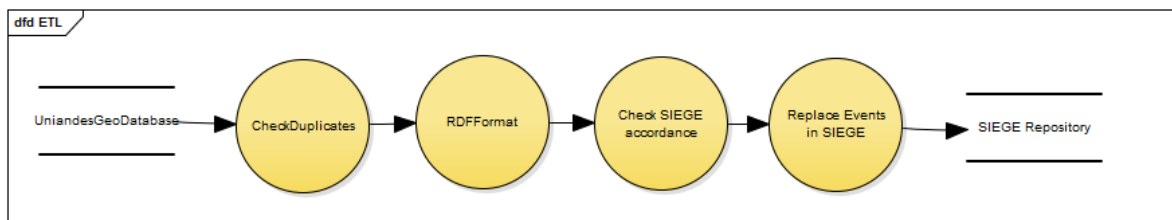


Ilustración 9 Proceso ETL para actualización de los eventos en SIEGE

El módulo `SIEGESystemFormat` extrae los registros de tipo evento de las tablas que se encuentran en el servidor de mapas. Luego de extraer los eventos de las tablas relacionales, este módulo reconstruye los eventos en el mismo formato utilizado por el *parser* del *feed* de Eventos.

Luego el módulo realiza la verificación de la correspondencia entre los eventos extraídos y los eventos existentes en SIEGE. A través de la comparación del título del evento, de la fecha de inicio y del lugar en el que ocurre, el módulo establece la correspondencia entre los eventos de los dos servidores y posteriormente reemplaza los registros en SIEGE con los eventos extraídos del servidor de mapas actualizando la información del motor de inferencias. De esta manera se logra que los eventos generados por los usuarios y los eventos provenientes del *feed* de Relaciones Públicas sean cargados diariamente al sistema para así actualizar el estado de confirmación y mantener la integridad del sistema.

4.5.3 Arquitectura del Servidor de Mapas

El servidor de mapas cumple múltiples funciones también utilizadas por (Rojas, 2014), (Díaz, 2004) y (Melani, 2014). Este servidor es una base de datos con información geográfica de los bloques, espacios y salones de la Universidad de los Andes. A través de la herramienta ArcGIS y ArcMap se definen una serie de servicios de localización utilizando los datos provenientes de las tres técnicas consideradas en la aplicación móvil: *beacons*, GPS y red WiFi.

En este servidor se registran todas las ocurrencias de un usuario, esto es la ubicación en la que se encuentra en un momento del día. También se registran los eventos creados por los usuarios asociados a un espacio y fecha específicos. En los eventos se especifica el estado de validación del evento dependiendo de si esta ha sido confirmado, negado o si todavía se espera una respuesta de validación por parte de los otros usuarios.

Las tablas *OCURRENCIAS*, *USUARIOS*, *EVENTOS* y *ESTADO_EVENTOS* son incluidas al modelo ya existente de Bloques y espacios. En la tabla *OCURRENCIAS* se registra la última ubicación en la que se encuentra un usuario al ser solicitada por la aplicación móvil. La tabla *USUARIOS* mantiene un registro de todos los miembros de la comunidad que se han registrado en la aplicación, y su seudónimo en caso de haberlo especificado. La tabla *ESTADO_EVENTOS* mantiene los posibles estados que puede tener un evento. SPEL considera tres estados:

- *Esperando Validación*: menos del 50% de usuarios a los que se les solicitó la validación del evento ha respondido. Cabe mencionar que responder la solicitud incluye tres posibles respuestas: confirmación, negación o desconocimiento del evento. En caso de desconocimiento, estos usuarios no entran en el cálculo del estado del evento ni de la calificación del usuario.
- *Confirmado*: más del 50% de los usuarios que no ignoraron la solicitud de validación, confirmaron que el evento efectivamente está sucediendo en el sitio de registro.

- Negado: más del 50% de los usuarios que no ignoraron la solicitud de validación, negaron que el evento estuviera sucediendo en el sitio de registro.

En la tabla *EVENTOS* se registran los eventos de acuerdo a lo descrito en la sección 4.5.1.4.

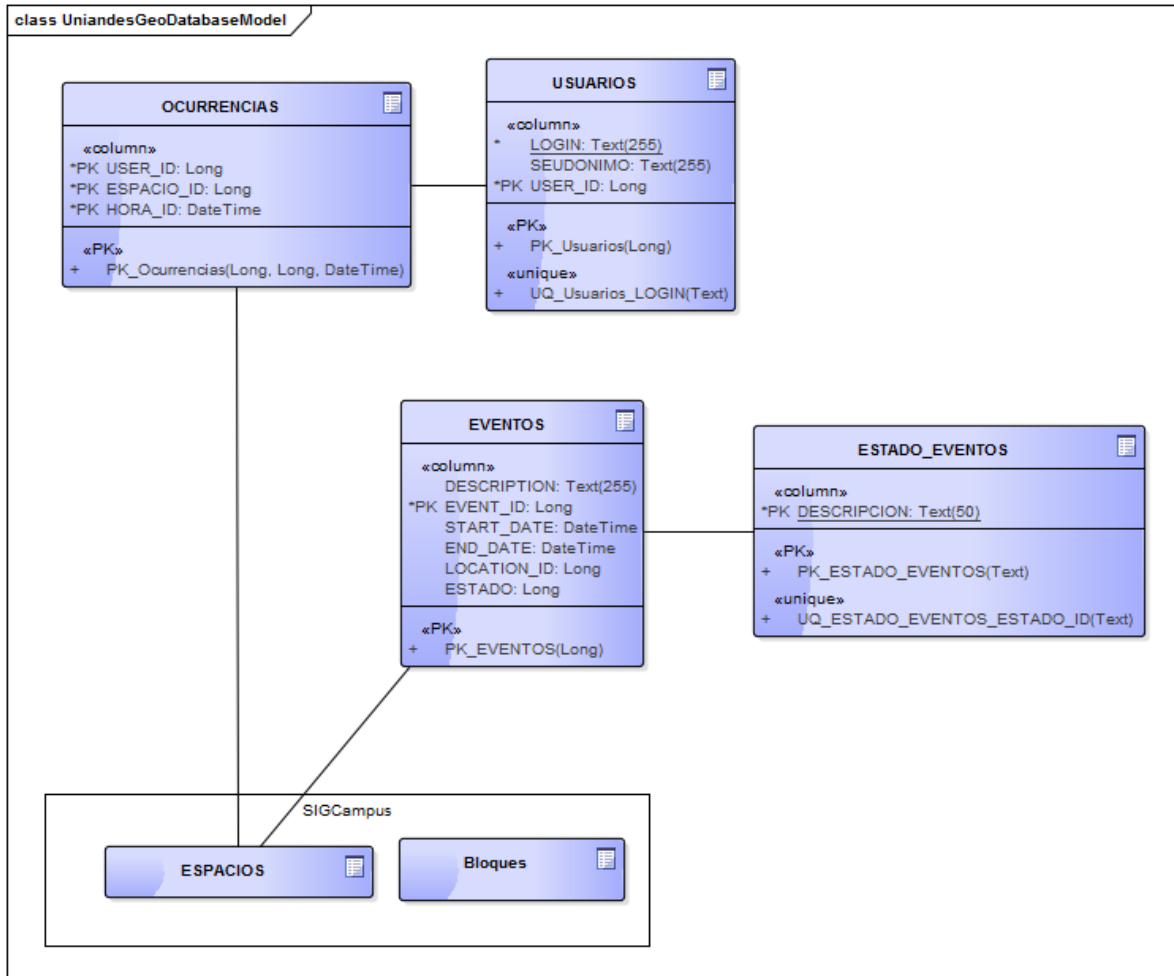


Ilustración 10 Detalle de la ampliación a las tablas incluidas en la base de datos.

Sobre esta base de datos se generan servicios *RESTful* que responden a las solicitudes de ubicación del usuario en el campus de la universidad y que también le permiten al sistema encontrar usuarios cercanos a un sitio determinado.

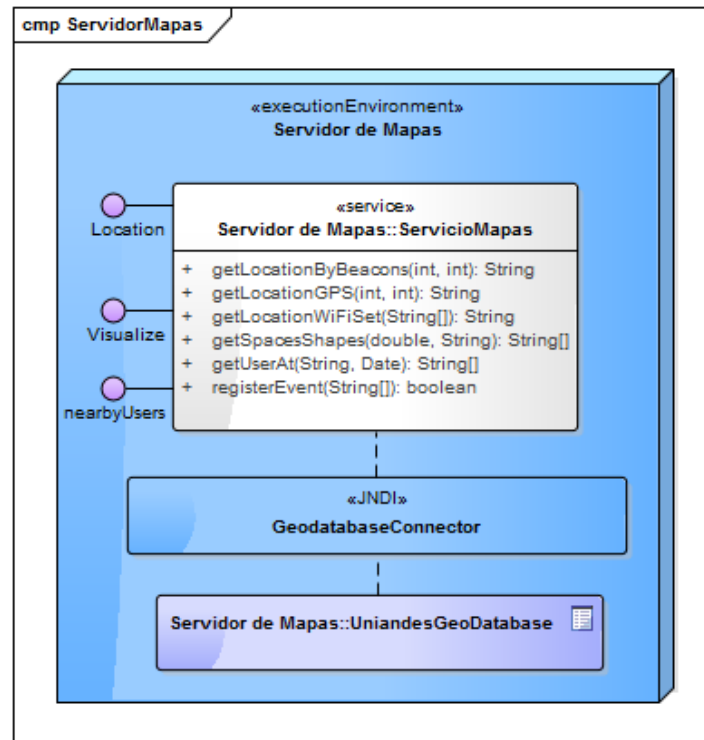


Ilustración 11 Especificación de Servicios sobre la base de datos geográfica

A través de una conexión por JNDI, el servidor extrae la información solicitada por los servicios de las tablas existentes y provee la información necesaria para realizar la sincronización de los eventos con SIEGE a través del módulo `SIEGESystemFormat`.

4.5.4 Arquitectura del Servidor de Validación de Eventos

El Servidor de Validación de Eventos cumple la función de distribuir la información de un evento registrado recientemente para ser confirmado por otros usuarios cercanos a la ubicación a la que el evento está asociado. El componente `PushValidation` en el servidor recibe los eventos creados desde la aplicación móvil en el dispositivo del usuario. Este componente utiliza un servicio generado sobre las tablas del servidor de mapas llamado `nearbyUsers` que, dado una magnitud radial y una ubicación, otorga la colección de usuarios cuya última ocurrencia se encuentra dentro del radio de la ubicación.

Luego de obtener la colección de usuarios cercanos a la locación, `PushValidation` remite un mensaje a los dispositivos móviles de estos usuarios solicitando la confirmación o negación del evento. El usuario puede afirmar, negar o desconocer la existencia del evento y envía esa respuesta al componente `PushValidation` que realiza el cálculo de la calificación del evento y del usuario de acuerdo a lo descrito en la sección 3.2 de este documento. Finalmente el componente le notifica a la aplicación móvil el resultado del cálculo para ser persistido en el dispositivo del usuario. Este

componente contiene un microcomponente de cifrado y descifrado de datos con una llave simétrica que sólo se encuentra disponible en los componentes PushValidation del servidor y PushValidation de la aplicación móvil.

Aunque SPEL no contempla el registro de ocurrencias de usuarios en los espacios del campus de la universidad, se debe tener en cuenta que el servidor de validación debe verificar que dentro del conjunto de usuarios, obtenidos de la tabla de ocurrencias, a los que les será enviada la solicitud de validación, no se encuentre el usuario que generó el evento. Esta comprobación es realizada por el componente PushValidation en el servidor antes de enviar la solicitud por GCM.

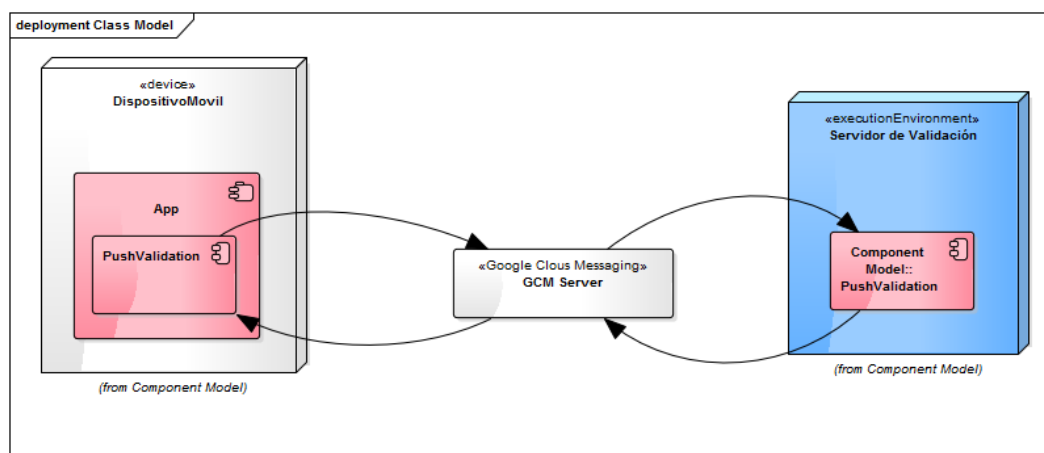


Ilustración 12 Esquema de comunicación por GCM

Los datos de la calificación del usuario son registrados localmente en la aplicación al terminar la validación del evento por parte de los otros usuarios. El componente del servidor PushValidation calcula y envía la calificación del evento individual generada de acuerdo al modelo descrito en la sección 3.2 y la aplicación es la encargada de agregar esta información a su cálculo acumulativo local de eventos realizado por el componente UserValidation. El modelo de reputación es manejado en el dispositivo del usuario, allí se lleva el registro del tiempo de penalización y la calificación del usuario por la sumatoria de la calificación de los eventos individuales generados por el usuario.

Comunicación por Google Cloud Messaging

La comunicación entre el servidor de validación y la aplicación móvil se realiza a través de *Google Cloud Messaging*. Este servicio de Google Inc., permite enviar mensajes ligeros (4kb) entre servidores y dispositivos móviles con sistema operativo Android 4.2 o mayor, a través de una conexión HTTP que requiere la autenticación de credenciales por ambos extremos de la comunicación.

Por el contenido de los mensajes, es necesario realizar el cifrado de los datos de ubicación del usuario antes de que estos sean enviados al servidor. La arquitectura utilizada es similar a la aproximación por *broadcast* utilizada localmente en la aplicación. En esta arquitectura el servidor envía los datos empaquetados en una acción acorde a lo esperado en la aplicación, el servidor GCM reenvía estos mensajes al dispositivo; en el caso en el que se encuentra apagado, el servidor GCM encola el mensaje y lo envía cuando el dispositivo esté disponible. Una vez recibidos los datos con la acción que la aplicación debe realizar, el dispositivo lo filtra para asegurar que el mensaje sea recibido por la aplicación. En caso de que la aplicación no esté activa, es despertada y será la encargada de procesar el mensaje.

4.6 Secuencia de Procesos

Es de especial importancia el registro y validación de los eventos creados por los miembros de la comunidad. La ilustración 13 explica la dinámica de registro y validación y la actualización del estado del evento en los servidores del sistema y la ilustración 14 muestra la interacción de los servicios de ubicación por GPS y *beacons* y su relación con el servidor de mapas.

En el registro y validación de eventos, una vez definida la ubicación del usuario, este puede generar, a través de la aplicación móvil, un reporte de evento que es recibido por el servidor de validación. El servidor de validación está encargado de persistir los eventos en el servidor de mapas y el servidor SIEGE. Posteriormente el servidor de validación solicita al servidor de mapas el conjunto de usuarios cercanos al lugar desde donde fue generado el reporte para solicitarles la validación de dicho evento y al recibir la respuesta de los usuarios cambia el estado de los eventos persistidos en SIEGE y en el servidor de mapas.

Para encontrar la ubicación del usuario en el campus de la universidad, los servicios de GPS y *beacons* están constantemente emitiendo actualizaciones de posición si existe un cambio significativo (mayor a 10 metros) o si se detectan nuevos *beacons* en el espacio. Estas emisiones son recibidas por la clase `LocationFinder` que realiza la solicitud de traducción de la información de los sensores al servidor de mapas para obtener el bloque o espacio en el que se encuentra el usuario. Posteriormente, `LocationFinder` realiza la persistencia local en el caché de la última posición recibida del servidor y esta es la que remita a la clase principal de la aplicación.

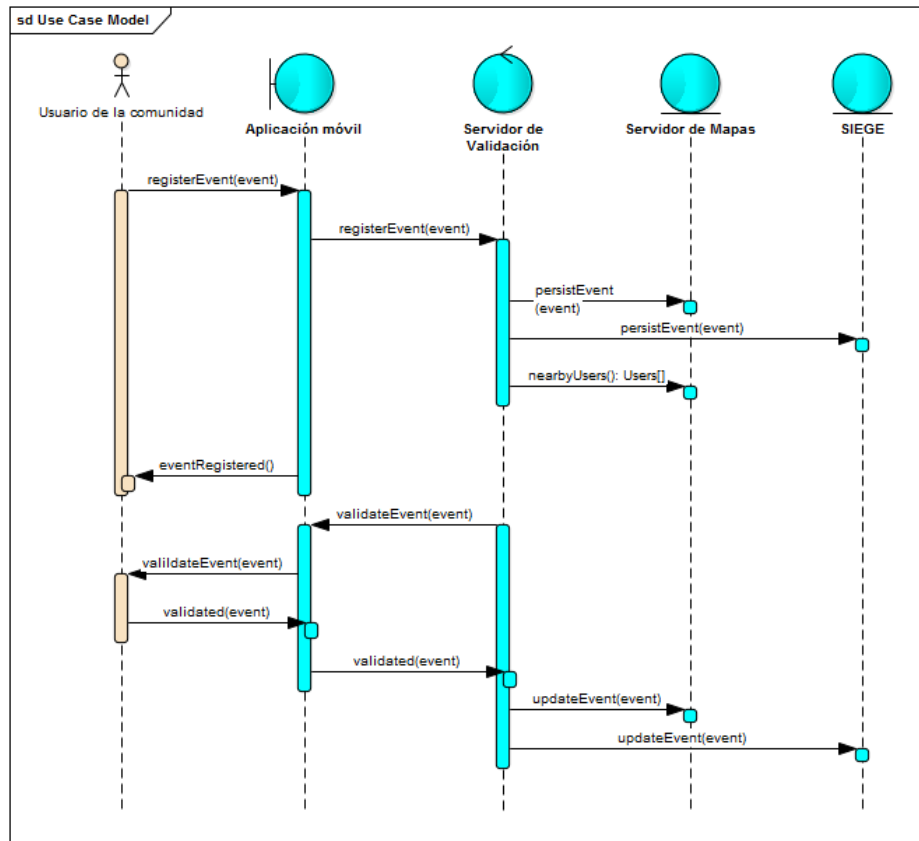


Ilustración 13 Diagrama de secuencia del registro y validación de reportes de eventos

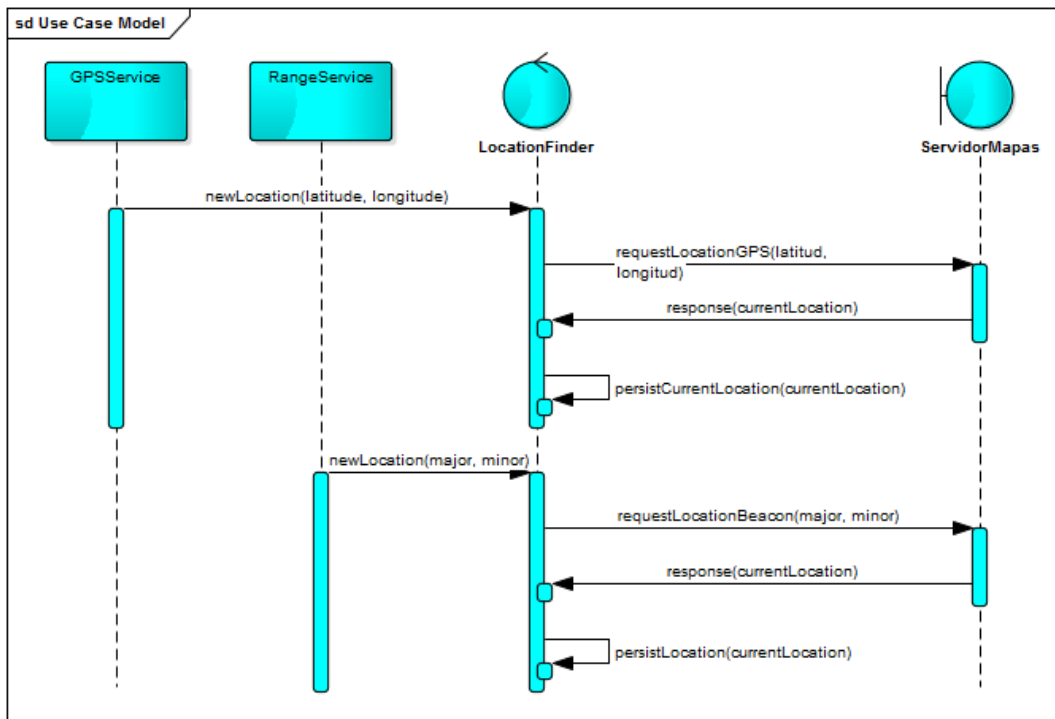


Ilustración 14 Diagrama de Secuencia de los Servicios de localización por GPS y Beacon

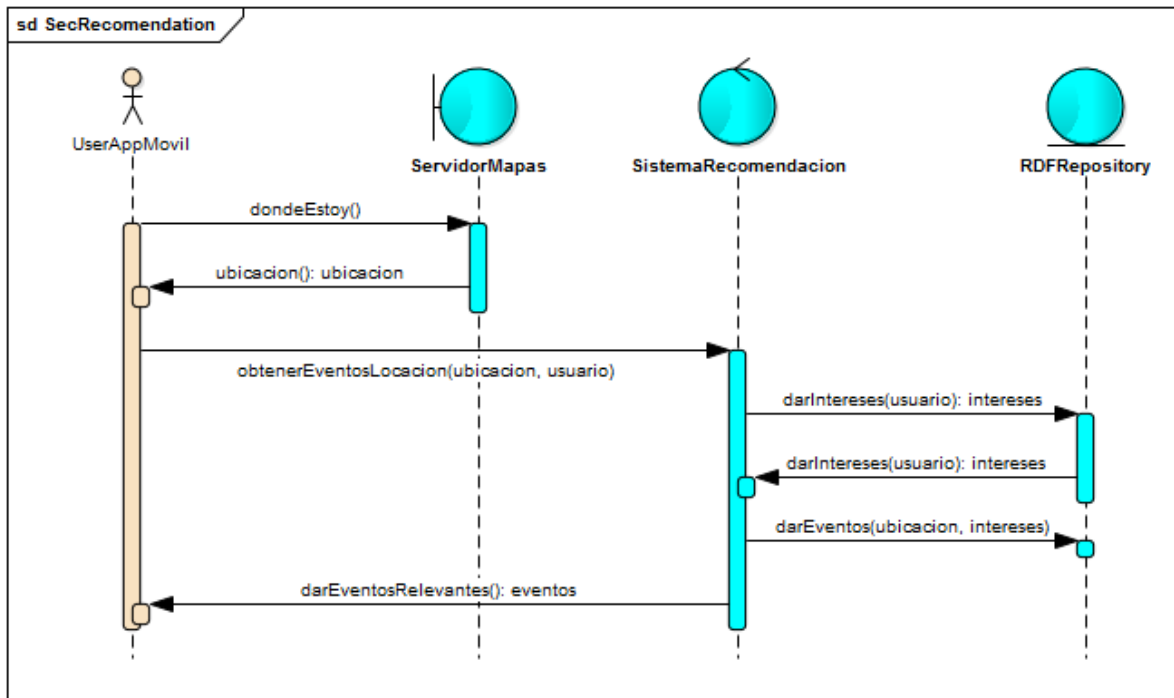


Ilustración 15 Diagrama de secuencia de recomendación de eventos

4.7 Autenticación de Usuarios

La autenticación de usuarios se realiza a través de la plataforma LDAP. Al momento de ejecutar la aplicación por primera vez, la aplicación solicita la introducción de las credenciales Uniandes para verificar que el usuario hace parte de la comunidad. También solicita la introducción de un seudónimo o nombre de usuario para la aplicación. Una vez realizada la autenticación, la aplicación persiste el nombre de usuario Uniandes y el seudónimo introducido por el usuario y envía estos dos campos al servidor de mapas en donde se encuentra la tabla de usuarios que almacena estos datos en sus registros.

La persistencia en el dispositivo no exige cifrado, por cuanto no se almacenan claves privadas sino solamente el nombre de usuario que se utilizará al enviar un reporte de evento. A este nombre están asociadas las validaciones que otros usuarios hacen sobre el evento generado, y sobre este se genera la calificación de usuario.

5 Desarrollo, implementación y Pruebas

5.1 Ambiente de desarrollo

La aplicación móvil fue desarrollada en Java utilizando el ambiente de desarrollo *Eclipse Android SDK Bundle* versión 20140702, para *Android 4.4 KitKat*, versión 19 del API por la compatibilidad con la librería de localización por *beacons*.

Los componentes del servidor SIEGE fueron realizados en php utilizando *SublimeText2*. Los servicios del servidor de mapas fueron generados desde la herramienta ArcMap. El servidor fue desplegado en una máquina *Ubuntu 12.04* con una instancia *Tomcat* para el despliegue de los servicios SIEGE.

5.2 Plataforma de Despliegue y Pruebas

La aplicación móvil fue desplegada en un *Samsung Galaxy S5* con Android versión 4.4.2 y Kernel Linux 3.4.0. El despliegue del servidor SIEGE se realizó utilizando una máquina virtual Ubuntu versión 12.04 LTS sobre un servidor Apache 2.4. En esta misma máquina se realizaron las pruebas funcionales del componente PushValidation del servidor de validación.

La mayor dificultad en el despliegue de los componentes fue la orquestación y conexión entre la aplicación y los diferentes servicios de los servidores debido, en gran parte, a los múltiples desarrollos en paralelo y las modificaciones que se iban presentando en cada uno lo que implicaba el desmontaje y redespliegue de los servidores. Por esta razón se realizaron pruebas simulando el comportamiento de los servidores una fueron requeridas las pruebas sobre la aplicación móvil. Se utilizó la herramienta SpaceBrew para simular el comportamiento del servidor de mapas y del servidor de validación en las etapas de prueba de la aplicación móvil.

5.3 Dificultades y problemas más significativos durante la implementación

Durante la implementación, uno de los grandes problemas fue el cambio de formato de la publicación de eventos por parte de la oficina de Relaciones Públicas. El cambio en el estándar que se había manejado requirió el cambio de la extracción de la información semántica del evento y de las heurísticas sintácticas utilizadas para la obtención de la locación y las horas de inicio y de finalización. Por lo tanto, la solución construida responde al momento puntual en el que se desarrolló esta implementación. En caso de que el formato de las publicaciones cambie, el *parser* del *feed* de eventos también debe ser modificado acordeamente. Aun así, los eventos generados desde la aplicación móvil mantienen el formato *RDF* necesario en SIEGE por lo que no es evidente

la necesidad de realizar también un ajuste en el módulo `EventRegistration`, encargado del registro de eventos desde la aplicación móvil. La tabla 1 muestra el resultado de la traducción de eventos del *feed* institucional a formato *RDF* para SIEGE.

Evento extraído del Feed Institucional en formato XML

```
<item>
  <title><![CDATA[Seminario: Propuesta de un nuevo sistema penitenciario y carcelario]]></title>
  <description><![CDATA[<span style="font-weight: bold;">Date: </span> 17/09/2012 11:30 AM
Esta es una invitaci&oacute;n de la
Direcci&oacute;n de Investigaciones de la Facultad de Derecho de la Universidad de los Andes.<br />
&nbsp;<br />
<strong>Entrada libre - cupo limitado.</strong><br />
<br />
<strong>Para apartar un cupo es necesario inscribirse en <a href="mailto:relexderecho@uniandes.edu.co"
class="ApplyClass">relexderecho@uniandes.edu.co</a>.</strong>]]></description>
  <link><![CDATA[http://eventos.uniandes.edu.co?sid=1384&gid=26&pgid=252&cid=3103&ecid=3103&ciid=5
150&crid=0]]></link>
  <enclosure
url="http://eventos.uniandes.edu.co/s/1384/images/content_images/boceto_mail_2393_634831273284704230.jpg"
length="68039" type="image/jpeg" />

  <pubDate>Thu, 13 Sep 2012 16:06:36 GMT</pubDate>
  <guid isPermaLink="false">ID1384-1556-5150</guid>
</item>
```

Evento producido por el Parser en formato RDF

```
<rdf:Description rdf:about="http://URI/SIEGE#ID1384-1556-5150">
  <rdf:type rdf:resource="http://URI/SIEGE#Event"/>
  <rdfs:label> Seminario: Propuesta de un nuevo sistema penitenciario y carcelario a</rdfs:label>
  <rdfs:comment>Date: 17/09/2012 11:30 AM Esta es una invitación de la Dirección de Investigaciones de la Facultad
de Derecho de la Universidad de los Andes.
Entrada Libre - cupo limitado.
Para apartar un cupo es necesario inscribirse en relaxderecho@uniandes.edu.co
</rdfs:comment>
  <SIEGE:startDateTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">17/09/2012 11:30 AM
</SIEGE:startDateTime>
  <SIEGE:endDateTime rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"></SIEGE:endDateTime>
  <SIEGE:isHeldAt rdf:resource="http://URI/SIEGE#"/>
  <SIEGE:isAbout rdf:resource="http://URI/SIEGE#orientacionprofesional@uniandes.edu.co"/>
  <SIEGE:isAbout rdf:resource="http://URI/SIEGE#Centro de Trayectoria Profesional"/>
</rdf:Description>
```

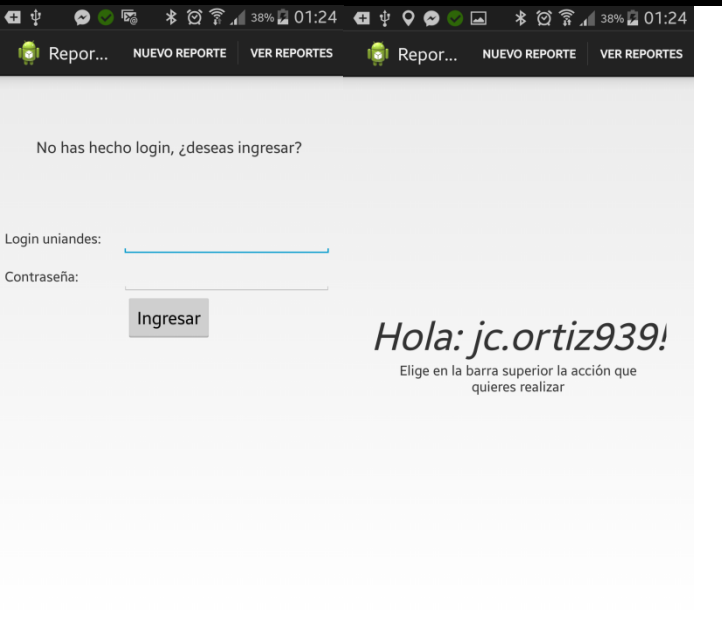
Tabla 1 Ejemplo de traducción de un evento desde el Feed Institucional a RDF

5.4 Interfaz Gráfica de la aplicación

En esta sección se presenta la interfaz gráfica implementada para la aplicación y para un primer prototipo de SPEL utilizado en la gestión de reportes de mantenimiento en el campus de la Universidad de los Andes

5.4.1 Interfaz Gráfica del prototipo de gestión de reportes de mantenimiento

Durante el desarrollo de este proyecto se implementó un primer prototipo con el cual se maneja la gestión de reportes de mantenimiento de infraestructura física del campus, entregado a la dependencia Planta Física de la oficina de Gerencia del Campus de la Universidad de los Andes.

Acción o Actividad	Interfaz Implementada
<p>Registro en la aplicación: el usuario debe ingresar sus credenciales para registrarse en la aplicación. El nombre de usuario es persistido localmente y los reportes realizados desde la aplicación son asociados a dicho usuario explícitamente.</p>	
<p>Creación de un nuevo reporte: interfaz de la actividad encargada de la generación de nuevos.</p>	

Creación de un nuevo reporte: mensajes de error cuando el reporte no contiene la información completa o confirmación cuando se ha registrado el reporte.

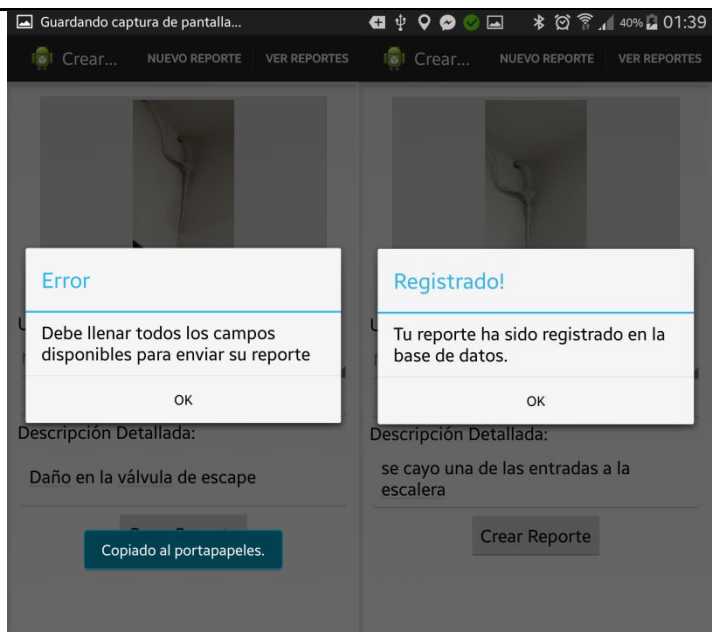


Tabla 2 Documentación de la interfaz del prototipo de gestión de reportes de mantenimiento

5.4.2 SPEL: Interfaz Gráfica de la aplicación móvil

La interfaz de la aplicación móvil mantiene los elementos estructurales y de control mínimos para facilitar la interacción del usuario con el sistema de una forma intuitiva y natural, en la medida de lo posible.

Acción o Actividad	Interfaz Implementada
Interfaz de registro en la aplicación por autenticación con LDAP y actividad principal de la aplicación con persistencia del nombre de usuario.	

Interfaz de creación del reporte de evento: campos del formulario y resultado cuando este ha sido creado exitosamente.

Reportar un... NUEVO REPORTE TEST

¿Qué está pasando?

Cuéntanos más...¿de qué se trata?

¿En dónde es?

Fuera del campus Map

¿Cuánto crees que dure?

0:00 hr

Selecciona el tipo de evento:

Cultural

Registrar Evento

Hola: jc.ortiz939!

Elige en la barra superior la acción que quieres realizar

Tu Evento ha sido reportado

Interfaz de creación del reporte de evento: resultado cuando el reporte contiene errores de formato.

Reportar un... NUEVO REPORTE TEST

Concierto de violin

Violinista escandinava en el campito

¿En dónde es?

No Disponible Map

¿Cuánto crees que dure?

Error

Elige una duracion aproximada

OK

Registrar Evento

Reportar un... NUEVO REPORTE TEST

Concierto de arpa

Música llanera en el campo to con un cuarteto de cuerdas

¿En dónde es?

Map

Error

Llena todos los campos

OK

Registrar Evento

Visualización de la información geográfica del campus.

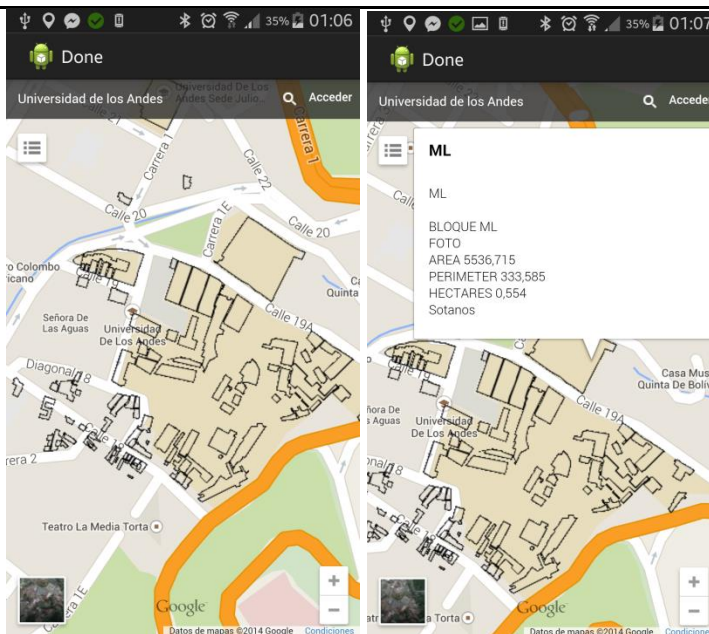


Tabla 3 Documentación de la interfaz de la aplicación móvil SPEL

5.5 Estructura del Proyecto Java

En esta sección se presenta la estructura del proyecto Android para SPEL. Se muestran los recursos, las librerías utilizadas y la estructura del código fuente en los paquetes especificados.

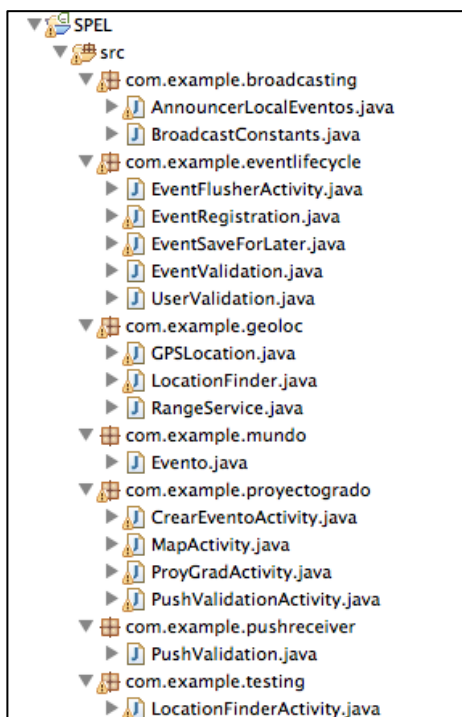


Ilustración 16 Estructura de archivos del código fuente de la aplicación móvil

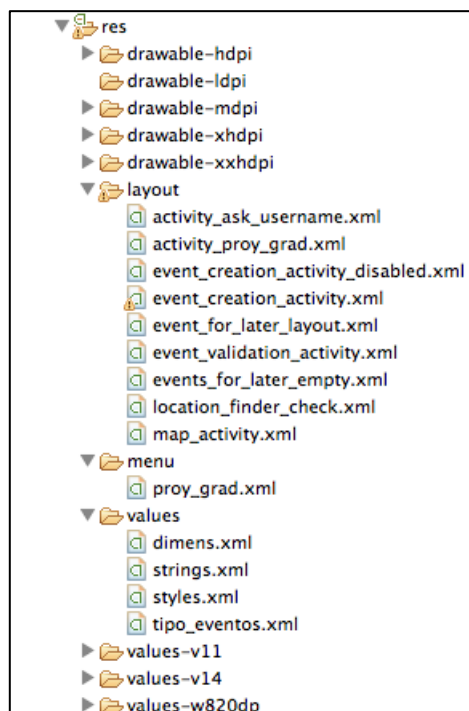


Ilustración 17 Estructura de recursos de la aplicación móvil

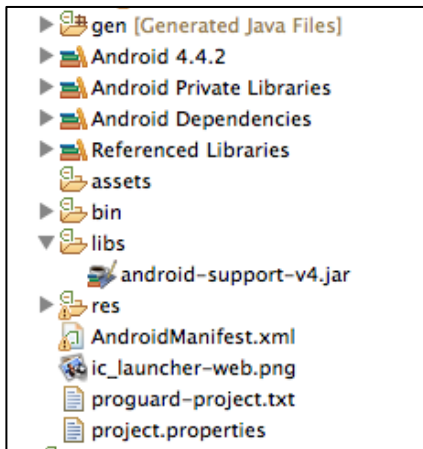


Ilustración 18 Estructura de librerías y archivos de configuración de la aplicación móvil

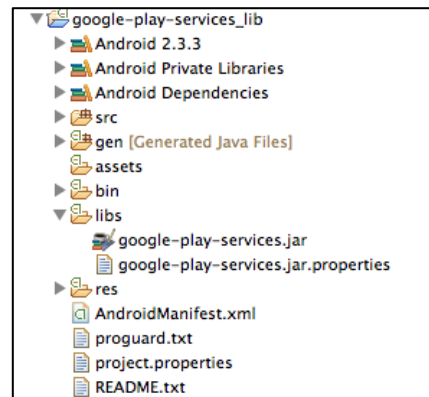


Ilustración 20 Estructura del proyecto librería Google Play Services utilizado en SPEL

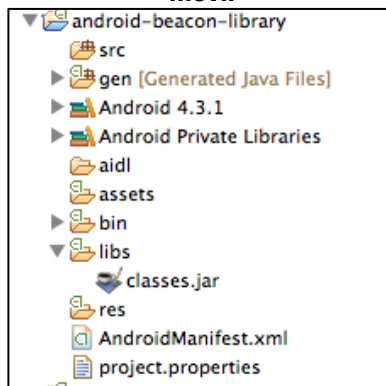
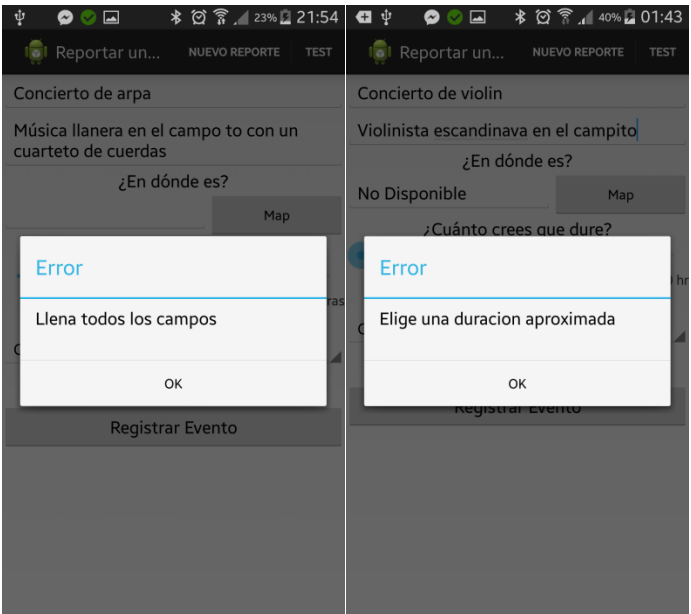


Ilustración 19 Estructura del proyecto librería Android Beacon Library utilizado en SPEL

5.6 Plan de Pruebas

En esta sección se describen las pruebas funcionales y no funcionales realizadas sobre los componentes del sistema de acuerdo al objetivo y el escenario especificados para cada prueba.

5.6.1 Pruebas Funcionales

5.6.1.1 Prueba 1: Registro de un nuevo evento con datos sin formato	
Tipo	Funcional
Objetivo	Determinar si es posible que un nuevo evento sea generado con datos en un formato erróneo
Escenario	En ejecución
Estímulo	El usuario registra un evento con datos 'basura'.
Resultados Esperados	El evento no puede ser registrado. Se le notifica al usuario el error de formato de los datos.
Resultados Obtenidos	<p>Exitosa</p> <p>El evento no es creado y un diálogo de error le informa al usuario el campo que debe ser cambiado para que el formato del evento sea correcto.</p> 
Observaciones	Este es el primer mecanismo de control en el contexto colaborativo del sistema. A través de esta prueba se asegura que la calidad de la información a difundir no sea baja.

5.6.1.2 Prueba 3: Ubicación del usuario por GPS y Beacons	
Tipo	Funcional
Objetivo	Determinar la ubicación del usuario a través de GPS y Beacons con una precisión de 10 metros.
Escenario	La aplicación móvil se encuentra instalada en un dispositivo y el servidor de mapas se encuentra activo
Estímulo	El componente <code>LocationFinder</code> solicita una actualización de la ubicación del usuario a los servicios <code>GPSLocation</code> y <code>RangingService</code>
Resultados Esperados	Se obtiene la ubicación del usuario por medio de ambos servicios con una precisión no menor a 10 metros de la ubicación real, en términos de espacios del campus.
Resultados Obtenidos	<p>EXITOSA</p> <p>En la ubicación por <i>beacons</i> se utilizaron dos <i>beacons</i> que fueron identificados exitosamente por la aplicación y traducidos correctamente por el componente de prueba utilizado.</p> <p>En la ubicación por GPS la aplicación identificó el cambio de edificaciones mientras se daba el desplazamiento.</p>
Observaciones	<p>Se mantuvo el nivel de granularidad de cada una de las estrategias de localización planteadas. Siendo así, la localización por <i>beacons</i> funciona a nivel de salón o espacio, mientras que la localización por GPS.</p> <p>Para el momento de realización de esta prueba el servidor no estaba disponible. Por lo tanto se utilizó un componente ajeno al servidor, que simula las respuestas del servidor utilizando la misma información que este contiene.</p>

5.6.2 Pruebas No Funcionales

5.6.2.1 Prueba 4: Latencia en validación	
Tipo	No Funcional
Objetivo	Determinar la velocidad del servidor de validación para enviar las solicitudes de validación a los usuarios cercanos a la locación desde

	la que ha sido registrado un nuevo evento.
Escenario	El servidor de validación se encuentra disponible. Se reportado un nuevo evento y la última ocurrencia de uno o más usuarios ha sido en el sitio desde el que se ha registrado el evento.
Estímulo	Un nuevo evento ha sido registrado por un usuario.
Resultados Esperados	El sistema solicita la validación del evento a los usuarios en un tiempo no mayor a 5 segundos. El usuario que generó el evento no recibe la solicitud de validación.
Resultados Obtenidos	Exitosa, con limitaciones
Observaciones	<p>Por cuanto las tablas de ocurrencia de usuarios no fueron implementadas, el componente <code>PushValidation</code> no tiene en cuenta el tiempo de consulta en las tablas, que, en caso de tener un número significativo de registros, puede llegar a ser del orden de segundos.</p> <p>La prueba se realizó utilizando un solo dispositivo</p>

5.6.2.2 Prueba 5: Latencia en consultas al servidor de mapas	
Tipo	No Funcional
Objetivo	Determinar la velocidad del servidor de mapas para enviar las solicitudes de localización desde la aplicación.
Escenario	El servidor de mapas se encuentra disponible. Se requiere la ubicación del dispositivo del usuario en el campus de la universidad.
Estímulo	Existe una solicitud de ubicación geográfica por GPS
Resultados Esperados	La aplicación solicita la ubicación geográfica del usuario por coordenadas GPS. El servidor es capaz de responder la solicitud en un tiempo no mayor a 4 segundos.
Resultados Obtenidos	<p>EXITOSA</p> <p>El servidor es capaz de responder las cinco solicitudes realizadas en diferentes espacios dentro del campus de la Universidad de los Andes.</p>
Observaciones	Esta prueba fue realizada utilizando el servicio de Google Maps y Fusion Tables de la información de límites geográficos de los bloques.

5.6.3 Análisis de Resultados

Con las pruebas realizadas es posible verificar la confiabilidad del sistema para ubicar al usuario en el campus de la Universidad de los Andes y generar recomendaciones de eventos relacionado a la ubicación geográfica encontrada. Al contar con la información geográfica de los bloques de la universidad, la ubicación por GPS resulta ser la más deseable al no requerir instalación de red de dispositivos, como si lo requiere la ubicación por *beacons*. Aunque el nivel de granularidad no pueda ser mayor actualmente por la precisión de la tecnología y la información geográfica que se posee, llega a ser suficiente para la extensión del campus y la cantidad de bloques que posee la universidad.

Se desarrollaron dos macrocomponentes desacoplados pero interdependientes que muestran el potencial de la arquitectura orientada a servicios para este sistema con componentes de alta granularidad y claridad en sus funciones y en donde las fuentes de información son los usuarios desde sus dispositivos móviles o repositorios de otros sistemas externos.

Por esta decisión de diseño se realizó la implementación en sistema operativo Android 4.4, lo que facilitó el desarrollo de servicios y la comunicación entre componentes. En caso de llevar el diseño de la aplicación móvil a la plataforma iOS, aunque se puede mantener el esquema de comunicación entre componentes, debe revisarse la viabilidad de las emisiones realizadas desde cada uno de los componentes del sistema y asegurar que estos puedan acoplarse a las restricciones de la plataforma. Por su parte, la comunicación con los servidores no debe resultar un inconveniente en caso de que se desee realizar una implementación de la aplicación en esta otra plataforma.

Fue posible verificar, aunque no era la intención inicial, que la migración de servicios a otras plataformas, como en el caso del servidor de mapas de Google o del *mock* del servidor de validación de eventos, no requiere de una gran transformación de los procesos definidos. Esto le da al sistema un carácter de portabilidad e interoperabilidad deseable en contextos académicos y experimentales, bajo los cuales se pueden desplegar sus servicios en diferentes escenarios.

También se confirma parcialmente la baja latencia que tiene el sistema para responder a las solicitudes realizadas, a pesar del alto nivel de desacoplamiento entre sus componentes. Los servidores del sistema y la aplicación móvil están contruidos con una arquitectura orientada a servicios con protocolos de comunicación ligeros lo que resulta ser efectivo para el contexto de computación ubicua en la percepción que tiene el usuario del sistema.

Por la ubicuidad de los dispositivos y la cantidad de sensores incluidos en ellos, es posible evidenciar la gran ventaja que ofrecen los dispositivos móviles en la generación colectiva de contenido; En este sentido, las pruebas realizadas resaltan la pertinencia de las tácticas de

validación de la información reportada incluso antes de que el dispositivo envíe los datos generados a los servidores para su divulgación.

También se encontró una limitación en la persistencia del contenido generado en el sistema de recomendación SIEGE, más específicamente en la generación de inferencias sobre la nueva información recolectada. Aunque se propone el reinicio de la ejecución del servidor de recomendaciones, esta solución es solamente parcial pues lo ideal es que el sistema esté siempre en línea. Una alternativa está en el desarrollo de un módulo o sistema que reciba el evento generado y validado y genere las inferencias necesarias por su propia cuenta. Este resultado debe ser después persistido en ejecución en el repositorio de la ontología de la cual se está alimentando el sistema de recomendación.

Dado el alcance del proyecto, se cuenta con una solución parcialmente desacoplada y los datos de validación recibidos en la aplicación móvil no corresponden a los procesos por los módulos de validación en el servidor correspondiente. De igual forma, los módulos de validación no cuentan con los recursos necesarios para su operación. En particular los registros de ocurrencia de los usuarios en los espacios del campus en los que compartieron su ubicación no existen. Este recurso es crucial para el correcto funcionamiento del servidor de validación y por su ausencia no se produce un juicio de valor sobre el esquema de validación colaborativa.

6 Conclusiones y Trabajo Futuro

En este documento se presenta el diseño y la implementación de SPEL, un sistema de recomendación de eventos basado en localización a partir de la construcción participativa de reportes de actividades y ocurrencias, con un esquema de validación colaborativa para un contexto de computación ubicua y transparente al usuario.

En un campus activo, en donde existe una gran oferta de eventos oficiales y espontáneos que pueden resultar pertinentes para un subconjunto miembro de la comunidad, la magnitud del espacio y del flujo de personas diariamente en la universidad representan retos considerables para la notificación de información oportuna y relevante para los usuarios. Fue construida una aproximación a la solución de este problema utilizando técnicas de localización geográfica, de extracción de información de eventos de fuentes oficiales y fuentes colaborativas, y de la validación participativa de la existencia de dichos eventos.

Utilizando sensores de GPS y Bluetooth BLE en el dispositivo móvil del usuario, fue posible realizar la identificación de su posición geográfica en términos de espacios y bloques en el contexto del problema. Se encontró una gran conveniencia en la utilización de los sensores GPS de los dispositivos móviles con el sistema de información geográfica de la Universidad de los Andes. El consumo de la información espacial, en conjunto con la adquisición de la posición geográfica del usuario, asegura una alta confiabilidad en las respuestas obtenidas. Aun así, esta tecnología no permite tener una mayor granularidad que el bloque en el que se encuentra el usuario, cuando lo ideal en el contexto específico del problema es tener una respuesta a nivel de espacio concreto.

Por otra parte, la aproximación colaborativa asegura una alta generación de contenido pero pone en riesgo la calidad de la información recolectada. Aunque la validación inicial de la información antes de ser divulgada resultó ser conveniente, para el contexto del problema esto es aún un nivel bajo de validación de contenido y se requieren técnicas de mayor rigurosidad. En este proyecto se diseñó un modelo de validación participativo que bien podría responder a este problema pero, al no haber sido sometido a un plan de pruebas, no se emite ningún juicio del valor sobre la conveniencia de dicho modelo. Aun así el presente documento propone este modelo como una interesante herramienta de validación para una futura prueba e implementación en este mismo contexto.

La información extraída de otros sistemas y la información generada por los usuarios alimentó la construcción semántica de categorías sobre los eventos registrados. Este proceso, realizado por medio de extracción de entidades y procesamiento de lenguaje natural, resulta ser satisfactorio, aunque no en todos los casos se obtiene un elemento de calidad para ser asociado a los intereses de los usuarios. Es probable que con técnicas de *machine learning* se logre refinar el proceso y

obtener categorías ontológicas de mayor calidad que puedan ser usados más efectivamente. Aun así este enriquecimiento resulta necesario para la realización de recomendaciones relevantes para el usuario.

En el ecosistema tecnológico de la Universidad de los Andes, la arquitectura orientada a servicios bajo la cual se definió el diseño de SPEL resulta ser increíblemente conveniente. El desacoplamiento de los servidores y componentes, junto con la alta integración que se tiene de la información proveniente de diversas fuentes de datos, institucionales y colaborativas, facilita la interoperabilidad de los servicios ofrecidos por los sistemas institucionales y permite una fácil integración de sus diferentes funcionalidades.

SPEL representa un interesante caso de aplicación de un sistema de recomendación que se apoya en la identificación de la ubicación geográfica del usuario, capaz de generar recomendaciones relevantes y de permitir el reporte colaborativo de evento y la validación participativa de dichos reportes, en un contexto de campus activo bajo la premisa de computación ubicua no intrusiva.

Bibliografía

- Bluetooth SIG, Inc. (s.f.). *Bluetooth Technology Specification*. Recuperado el 16 de Noviembre de 2014, de <https://www.bluetooth.org/en-us/specification/adopted-specifications>
- De Cristofaro, E., & Soriente, C. (2012). Participatory Privacy: Enabling Privacy in Participatory Sensing. *IEEE Network*.
- Dempsey, C. (1 de Marzo de 2012). *GISLounge*. Recuperado el 22 de Noviembre de 2014, de What is GIS?: www.gislounge.com/what-is-gis
- Diaz, L. (2004). *Attendace Uniandes: sistema pervasivo para el registro de asistencia y manejo de evacuaciones*. Bogotá: Universidad de los Andes.
- Gerencia del Campus, Universidad de los Andes. (2010). *Facts and Figures*. Bogotá, Colombia.
- Goodchild, M. (1991). Geographic information systems. *Progress in Human Geography*, 194-200.
- Henao, A. (2013). *Budgie System: Identifying Individuals Behavioral-Patterns to generate Pertinent Alerts in a Pervasive System at Scale*. Bogotá: Universidad de los Andes.
- IEEE Computer Society. (2009). *1061-1998 - IEEE Standard for a Software Quality Metric Methodology*.
- Lyons, G. (2014). *SIEGE: Delivering relevant information through environment aware pervasive computing system*. Bogota: Universidad de los Andes.
- Melani, A. (2014). *Generación de un campus activo a través de una solución basada en movilidad tipo pervasiva*. Bogotá: Universidad de los Andes.
- Ping, T., Rudys, A., Ladd, A. M., & Wallach, D. S. (2003). *Wireless LAN Location-Sensing for Security Applications*. Houston, TX: Rice University.
- Rojas, N. (2014). *Limbus: sistema para monitoreo pervasivo de tráfico y ocupación de espacios*. Bogotá: Universidad de los Andes.
- Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 75-84.
- Young, D. (2012). Android Beacon Library. *An Android library providing APIs to interact with beacons*. Washington, DC.

- Zandbergen, P., & Barbeau, S. (2011). Positional Accuracy of Assisted GPS Data from High-Sensitivity GPS-enabled Mobile Phones. *Journal of Navigation*, 381-399.
- Zogg, J.-M. (2002). *GPS Basics: Introduction to the system. Application Overview*. Thalwil, Suiza: ublox.