



PROYECTO DE GRADO
INGENIERÍA EN COMPUTACIÓN

Plataforma de Alertas y Notificaciones Georreferenciadas

Integrantes

Gastón Rosas

Carlos Vélez

Agosto 2019

Supervisora

MSc. Ing. Raquel Sosa

Resumen

Con el paso de los años la cantidad de aplicaciones móviles existentes ha aumentado considerablemente y con ello la cantidad de notificaciones o alertas que reciben diariamente los usuarios. Por lo general, para el usuario es deseable que las notificaciones que recibe en su dispositivo sean relevantes y de su interés.

Frecuentemente las aplicaciones móviles que utilizan los usuarios en sus dispositivos tienen acceso a información valiosa como pueden ser los datos del GPS del dispositivo y en muchos casos, la relevancia de la información para el usuario está altamente vinculada a la ubicación geográfica donde se encuentre y según temas de interés que de alguna manera pueda elegir.

Los encargados de generar los avisos a los clientes, es decir los proveedores de datos, disponen de varios mecanismos para generar la información y enviarla. Esto implica que no siempre cuentan con las herramientas que mejor se adapten a sus necesidades y tampoco es posible compartir ciertas funcionalidades que les pueden ser de utilidad a otros proveedores de información.

Una forma de mejorar estos aspectos es centralizar el envío de notificaciones por parte de los proveedores de información y que cada usuario pueda optar que tipo de notificaciones desea recibir según una ubicación determinada y un tema de interés. Una solución a esto fue planteada en la Tesis de Maestría *Extensión Espacial de Middleware orientado a mensajería para dar soporte a servicios basados en geolocalización* de la Maestría en Sistemas de Información y Tecnologías de Gestión de Datos.

En esta tesis de referencia se plantea una solución gestionada por AGESIC que sea compatible con la Plataforma de Gobierno Electrónico para colaborar con el Estado Uruguayo en el envío de información relevante a la población, donde se utilice la geolocalización como base de las notificaciones.

Dado que la tesis de referencia no cuenta con un desarrollo completo de la solución planteada, este proyecto de grado se centra en la implementación de esta arqui-

tectura con todos los componentes necesarios para su utilización y la extensión con un componente de administración.

Se desarrolló un componente central en lenguaje Java llamado GeoMOM Central que contiene gran parte de las funcionalidades de la plataforma como por ejemplo, la interacción con los proveedores de información utilizando servicios web SOAP, pasando por la lógica de las suscripciones de los clientes y el envío de las notificaciones basadas en el elemento geográfico del evento respecto a la ubicación del cliente.

El otro conjunto de funcionalidades radica en el Broker de mensajería, que se encarga de la comunicación con los clientes utilizando colas de mensajes implementadas con ActiveMQ para el envío de mensajes. Toda comunicación entre los clientes y la plataforma se realiza a través de servicios web en tecnología Rest.

A la solución planteada en la tesis de referencia, se la extendió con un componente de administración web que se encarga de la gestión de topics, proveedores y clientes. Además brinda información estadística de la plataforma para la toma de decisiones por parte de los administradores con el fin de mantener el correcto funcionamiento de la plataforma.

Por último, en relación al caso de estudio, se implementó una aplicación móvil, compatible con el sistema operativo Android, con las funcionalidades principales que debería tener la aplicación para el cliente.

Palabras clave: Notificación, Geolocalización, Middleware, AGESIC, PGE, GeoMOM

Índice general

Resumen

1. Introducción	3
1.1. Contexto y motivación	3
1.2. Objetivos	5
1.3. Aportes realizados	6
1.4. Organización del documento	7
2. Marco conceptual	9
2.1. Proyectos relacionados	9
2.1.1. MedJelly	9
2.1.2. JobWatch y JourneyWatch	10
2.1.3. CityCop	11
2.1.4. INUMET	12
2.1.5. Sistema para peregrinos de Hajj	12
2.1.6. Sistema de alerta al viajero	14
2.2. Conceptos relacionados	14
2.2.1. Plataforma de Gobierno Electrónico - EGob	14
2.2.2. LBS - Location Based Services	16
2.2.3. MOM - Message Oriented Middleware	17
2.2.4. JMS	19
2.2.5. Broker de mensajería	21
2.2.6. Patrón Publish-Suscribe	21
2.3. Tesis de Maestría	22
2.3.1. Solución planteada	23
2.3.2. Propuesta de la solución	26
2.3.3. Arquitectura	28

3. Propuesta de solución	29
3.1. Actores	29
3.2. Requerimientos y casos de uso	30
3.2.1. Requerimientos funcionales	30
3.2.2. Requerimientos no funcionales	33
3.2.3. Diagrama de casos de uso	34
3.3. Arquitectura de la solución	35
4. Descripción de la solución	37
4.1. GeoMOM Central y Broker de mensajería	38
4.1.1. GeoMOM Central	39
4.1.2. Broker de mensajería	46
4.2. Base de datos	49
4.3. Web de Administración	50
4.3.1. Gestión de Topics y Proveedores	51
4.3.2. Estadísticas para la gestión	52
5. Caso de estudio y validaciones	59
5.1. Caso de Estudio	59
5.1.1. Datos utilizados	59
5.2. Aplicación cliente	61
5.2.1. Funcionalidades	61
5.2.2. Notificaciones push	67
5.2.3. API Google Maps	67
5.3. Verificaciones y validaciones	68
5.3.1. Publicar evento	69
5.3.2. Consultar alertas	71
5.3.3. Notificar alerta	72
5.3.4. Enviar ubicación	72
5.3.5. Validaciones Administración Web	72
6. Gestión de Proyecto	75
6.1. Descripción	75
6.2. Hitos	75
6.3. Repositorio Git	78

7. Conclusiones y trabajo a futuro	79
7.1. Conclusiones	79
7.2. Trabajo a futuro	80

Capítulo 1

Introducción

1.1. Contexto y motivación

Para cualquier sistema o producto generador de información, el principal objetivo es que los receptores de estos datos puedan obtener lo que buscan de forma fácil y concisa. Esto se ve incrementado si el acceso a la información se realiza desde dispositivos móviles, ya que en este contexto los usuarios esperan que la información esté disponible al instante y sea de utilidad.

Una de las principales formas de envío de información hacia los usuarios de dispositivos móviles, es a partir de notificaciones o alertas, sin que sea necesaria una acción explícita por parte del usuario para obtener dicha información.

Dado que actualmente existe alta disponibilidad y gran variedad de aplicaciones, la cantidad de notificaciones que son dirigidas a un usuario diariamente suele ser muy alta. Por esta razón es que en muchas ocasiones la información enviada en las notificaciones es ignorada por el usuario, dado que es difícil de clasificar o simplemente no es de su interés.

En muchos casos el contenido de estas notificaciones está altamente relacionado con la zona geográfica de las alertas y, sobre todo, con el lugar dónde se encuentre el usuario final que recibe la notificación quien determina, en definitiva, la utilidad de la alerta recibida. Además, en muchos casos el usuario no cuenta con la posibilidad de elegir la temática que desea recibir o no información.

Esta problemática fue abordado en la Tesis de Maestría denominada *Extensión Espacial de Middleware orientado a mensajería para dar soporte a servicios basados en geolocalización* [1] de la Maestría en Sistemas de Información y Tecnologías de Gestión de

Datos. En este trabajo se plantea una solución que pueda ser incluida en la PGE[2] como una herramienta que ayude al Estado Uruguayo en el envío de información relevante hacia la población. Esta solución permite enviar alertas de distintos temas, como accidentes de tránsito, alertas meteorológicas, nuevos locales comerciales, eventos relacionados a la salud, entre otros. En todos los casos, las alertas cuentan con un componente geográfico asociado a un evento, generado por un proveedor de datos registrado. Los ciudadanos, que serán potenciales clientes de esta plataforma, podrán suscribirse a ciertos tipos de eventos o temáticas indicando una zona de interés y así podrán recibir alertas que cumplan con esos requisitos. Esta tesis de maestría será tomada como referencia para nuestro proyecto de grado.

Un ejemplo de uso de este tipo de soluciones es el sistema de alertas de INUMET[3] que tiene como finalidad prestar los servicios públicos meteorológicos y climatológicos en Uruguay. Historicamente las alertas que emitía Inumet hacían referencia a regiones determinadas, es decir se daba una lista de departamentos o ciudades que eran afectadas por la alerta, sin precisión geográfica. En noviembre de 2017, implementaron un nuevo sistema de visualización espacial de las alertas meteorológicas [4] que mejoró ese aspecto, ahora se emiten alertas con un dibujo de la zona afectada. Con este cambio, un usuario puede saber de forma más precisa si la alerta emitida está en su ubicación o en determinada zona geográfica de su interés.

Aun así, la información de las alertas enviadas a través de la aplicación móvil de INUMET es recibida por todos los usuarios de la aplicación, sin importar la ubicación geográfica de la alerta ni la del usuario. Esta problemática de la aplicación ha sido notada por los usuarios en varias ocasiones. En la Figura 1.1 se pueden observar algunos comentarios realizados a la aplicación INUMET en Google Play [5] dónde se puede ver claramente la recomendación por parte de los usuarios de que se envíe información solamente cuando las alertas coinciden con su ubicación actual o con una zona de su interés.

Por otro lado están los proveedores de información, es decir los encargados de enviar información a los usuarios, que cuentan con sus propias herramientas para emitir los avisos o alertas. Esto implica que si un proveedor desea enviar información a distintos grupos poblacionales, es necesario adaptar los datos para que sean compatibles con las herramientas necesarias para llevar a cabo el envío.

Tampoco existe, por parte del Estado Uruguayo, una herramienta que unifique los sistemas utilizados por los distintos ministerios, intendencias o diversos sistemas

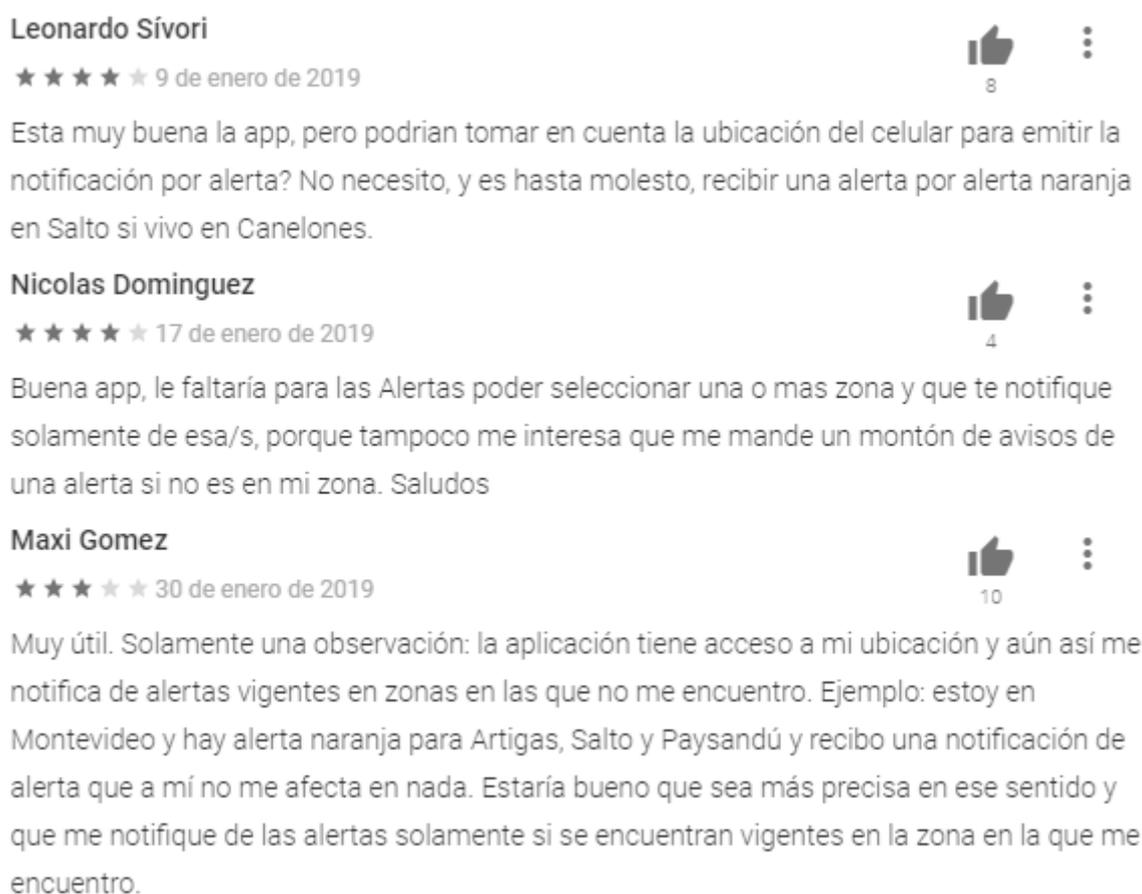


Figura 1.1: Comentarios de la aplicación INUMET en Google Play

encargados de notificar a la población de los sucesos que ocurran en el país.

Al desarrollar una solución como la planteada anteriormente e incluirla dentro de la Plataforma de Gobierno Electrónico, no solamente los usuarios se verían beneficiados, sino que también favorecería la comunicación entre los proveedores y sus clientes. En este escenario los proveedores dispondrían de una plataforma unificada donde publicar información de interés general desde la órbita del Estado, para notificar a los ciudadanos de ciertos eventos de relevancia tanto a nivel local como nacional.

1.2. Objetivos

El objetivo principal de este proyecto es implementar la solución completa propuesta por la tesis de referencia [1] y extenderla con un componente de administración.

Los objetivos específicos son:

- Estudiar la propuesta planteada en la tesis de referencia para diseñar una solución.

- Realizar una implementación completa de la arquitectura presentada en la tesis de referencia.
- Extender dicha implementación con la administración de la plataforma en tecnologías web.
- Realizar un ciclo completo de validación, con una aplicación cliente en versión móvil.
- Generar la documentación anexa necesaria para la puesta en funcionamiento de la plataforma.

Al finalizar el proyecto, se espera tener el diseño e implementación de una plataforma de middleware orientada a mensajes para el envío unificado de avisos en el contexto de la Plataforma de Gobierno Electrónico. Además se espera contar con el desarrollo del componente de administración web.

1.3. Aportes realizados

Para el desarrollo del marco conceptual se estudiaron propuestas de plataformas, informes académicos y software de mercado con uso de alertas georreferenciadas de naturaleza similar a la propuesta de este proyecto. Esto permitió generar una vasta visión del uso de este tipo de servicios y de su adecuación a la vida cotidiana.

Además se realizó un breve análisis de los conceptos teóricos que están relacionados de una manera u otra a la realización de este proyecto. Este análisis se puede encontrar en mayor extensión en el Anexo A: Marco teórico

Luego se plantearon los requerimientos para el diseño y construcción de la plataforma. Con base en estos requerimientos se diseñó la solución, se plantearon los casos de uso relevantes a la arquitectura y las interfaces necesarias. Este estudio se puede encontrar en el Anexo B: Arquitectura 4+1.

Una vez culminada la etapa de diseño, se desarrollaron los principales casos de uso para la construcción de la plataforma. Además se implementó el componente de la administración web para la gestión de la plataforma y la generación de estadísticas de su uso.

En relación al caso de estudio, se elaboró una aplicación para dispositivos móviles, que permitió generar el ciclo completo de pruebas de la solución, incluyendo el

envío de notificaciones georreferenciadas.

1.4. Organización del documento

El presente documento se organiza en siete capítulos.

En el primer capítulo se realiza una descripción del proyecto, la motivación, sus principales objetivos y aportes realizados.

En el segundo capítulo se describen los conceptos básicos necesarios para la realización y comprensión del proyecto, los trabajos relacionados, y una introducción de la tesis de referencia.

En el tercer capítulo se presenta la propuesta de solución del proyecto, los requerimientos y principales casos de uso, y la arquitectura de la plataforma.

En el cuarto capítulo se detalla la implementación de la solución y el funcionamiento de los principales componentes.

En el quinto capítulo se presenta el caso de estudio para llevar a cabo las pruebas necesarias, datos utilizados y resultados obtenidos.

En el sexto capítulo se muestran detalles de la gestión del proyecto e hitos importantes.

En el séptimo capítulo se presentan las conclusiones del proyecto y el trabajo a futuro.

Capítulo 2

Marco conceptual

En este capítulo se presenta un resumen de las distintas soluciones existentes en el mercado que tienen puntos en común con nuestro proyecto.

También se profundiza en algunos conceptos relacionados que serán de utilidad a lo largo del proyecto. Se definen algunos conceptos técnicos de la Plataforma de Gobierno Electrónico del Uruguay, servicios basados en localización, los sistemas de mensajería, entre otros lenguajes y tecnologías que podrían ser utilizados.

Por último se realiza una introducción de la tesis de referencia [1] donde se presentan los principales conceptos tomados de ese trabajo para el desarrollo de este proyecto.

2.1. Proyectos relacionados

2.1.1. MedJelly

El Proyecto MedJelly [7] se basa en una colaboración de más de 10 años entre el ICM-CSIC¹ y los municipios costeros de Cataluña que, a través de sus Servicios de Salvamento y Socorrismo, han logrado establecer una red de observadores a lo largo de todo el litoral catalán. Uno de los principales objetivos es consolidar la recolección de datos de presencia y/o ausencia de medusas de forma diaria y en tiempo real a nivel de playa. Los voluntarios que recogen y envían la información diaria, mayoritariamente los socorristas, reciben formaciones específicas y material de identificación de apoyo, facilitando con esto el desarrollo del trabajo y asegurando una correcta reco-

¹ICM-CSIC - Instituto de Ciencias del Mar de Barcelona - Consejo Superior de Investigaciones Científicas

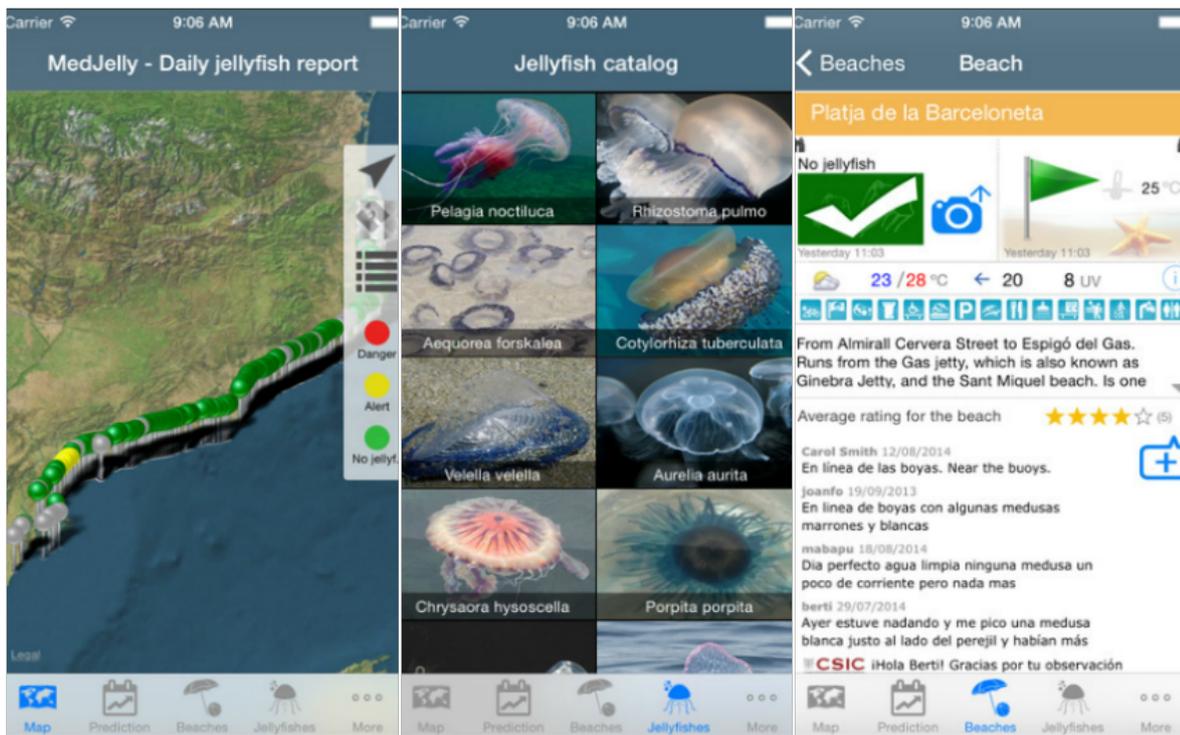


Figura 2.1: MedJelly. Tomado de [6]

lección de los datos. Estos datos luego son validados por expertos para poder entregar información contrastada a la ciudadanía.

La aplicación móvil iMedJelly entrega información científica actualizada sobre cada una de las especies de medusas que se pueden encontrar en la costa catalana y en el resto de la cuenca mediterránea, sumado a protocolos de actuación específicos frente a las picaduras de medusas. La última versión de la aplicación también invita a la ciudadanía a ser parte de esta red de observadores y contribuir con esto al mejor funcionamiento de la aplicación.

2.1.2. JobWatch y JourneyWatch

JobWatch [9] es una plataforma para la gestión de empresas, la cual se especializa en planificar, gestionar, programar y realizar un seguimiento del personal, y sus vehículos.

El seguimiento de personal en tiempo real se divide en dos partes: *Alertas* y *Seguimiento en Tiempo Real*, que aportan información georreferenciada con el fin de facilitar la gestión operativa y optimizar los recursos del personal móvil.

Las alertas le permiten mantenerse al tanto de lo que sucede y responder de una manera proactiva ya que todos los días surgen imprevistos que amenazan con poner en



Figura 2.2: JourneyWatch. Tomado de [8]

peligro lo programado, desde incidencias de tráfico y trabajos urgentes no planificados hasta visitas y entregas que se prolongan mucho más de lo estimado.

Journey Watch [8] permite identificar fácilmente los trabajos en riesgo, o posibles retrasos mediante alertas, permitiendo informar a los clientes y ajustar el plan en consecuencia.

Las alertas también pueden poner de manifiesto la existencia de jornadas de trabajo excesivas, usos fuera de las horas laborables e inquietudes relacionadas con la seguridad, como las directivas relativas a las horas de trabajo y el comportamiento de los conductores. En el Seguimiento en tiempo Real se comparte la ubicación de cada recurso móvil a través de sus smartphones.

2.1.3. CityCop

CityCop es una aplicación para realizar denuncias, enviar alertas o indicar acciones sospechosas, como por ejemplo: hurtos, robos u otras actividades delictivas.

El usuario baja la aplicación a su smartphone, se registra con sus datos personales y marca sus zonas de interés, como por ejemplo “casa” o “trabajo”.

A continuación, recibirá información relevante sobre lo que esté sucediendo en

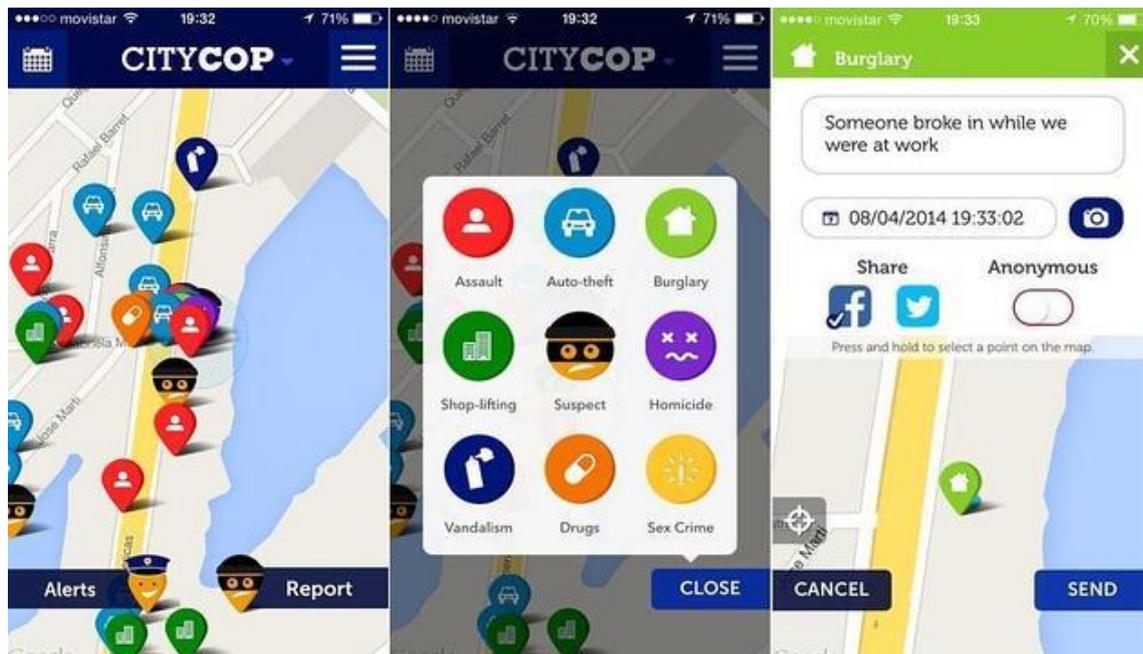


Figura 2.3: CityCop. Tomado de [10]

las zonas marcadas, pudiendo recibir incluso alertas generadas por otros usuarios, o realizar él mismo denuncias de hechos delictivos. A su vez, la aplicación permite indicar la fecha y la hora exacta del suceso, ingresar fotos y compartir las alertas por redes sociales.

2.1.4. INUMET

El Instituto Uruguayo de Meteorología (INUMET) [3] es el organismo encargado de prestar los servicios públicos meteorológicos y climatológicos del Uruguay, con el fin de asegurar la seguridad de las personas y bienes. Dicho instituto lanzó la aplicación homónima, con el fin de notificar las alertas climáticas a sus usuarios, brindar la información del pronóstico del tiempo, entre otros. Se pueden recibir distintos tipos de alertas categorizadas como roja, naranja y amarilla. En la Figura 2.4 se pueden ver algunas capturas de pantalla de esta aplicación.

Cuando existe una alerta meteorológica vigente, la aplicación notifica a todos sus usuarios registrados, mostrando en un mapa las zonas afectadas.

2.1.5. Sistema para peregrinos de Hajj

La congregación Hajj [11] es un evento anual dónde se aglomeran cientos de personas en un mismo lugar. Esto trae muchos desafíos para garantizar que todos los

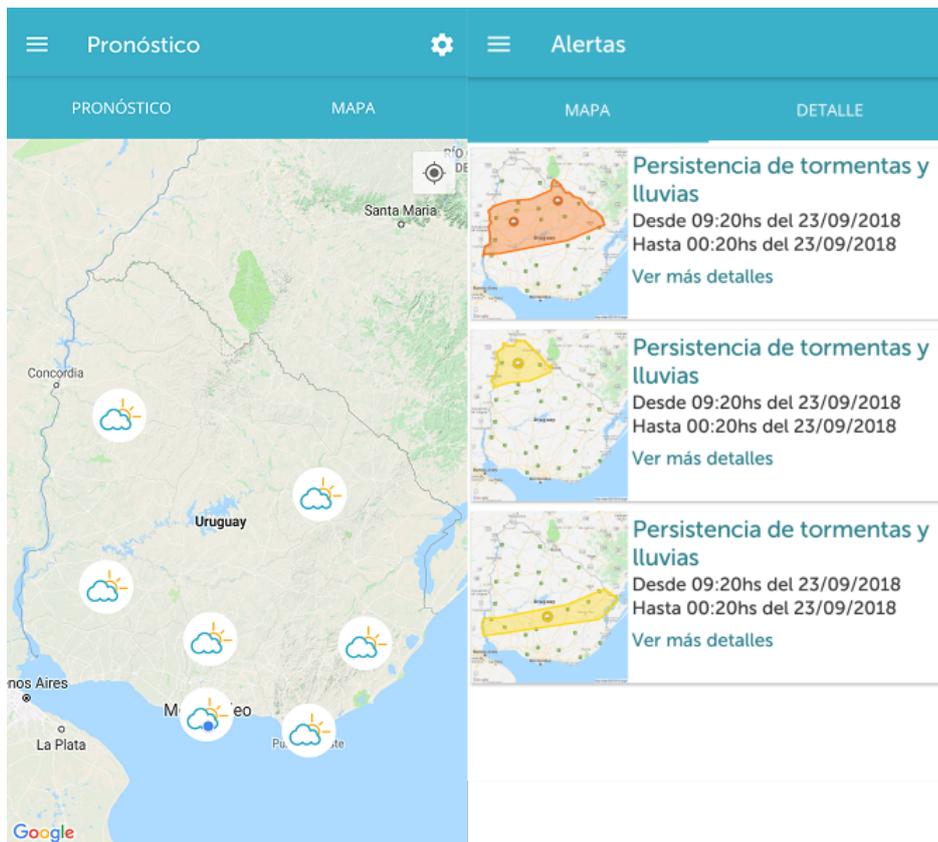


Figura 2.4: Aplicación Android de INUMET

peregrinos estén seguros en la realización de los rituales del Hajj, por lo tanto la tecnología de la información se puede usar para ayudar a manejar esa situación.

En este documento[12], se presenta un diseño arquitectónico del sistema de alertas de emergencia por zonas para peregrinos del Hajj. Usando el sistema, un peregrino puede enviar una alerta informando su ubicación y cual es su emergencia. La alerta se distribuye a otros miembros del grupo que se encuentren a cierta distancia del peregrino, incluyendo el líder de su grupo y su agente de viajes Hajj. Esta alerta les permite localizar al peregrino rápidamente y ofrecerle la ayuda necesaria.

La arquitectura planteada utiliza teléfonos móviles con GPS² como dispositivos de rastreo. El diseño está pensado no solo para entregar la funcionalidad esperada sino que también para manejar una alta demanda de usuarios activos con algoritmos para balancear la carga de los sistemas.

²GPS - Sistema de Posicionamiento Global

2.1.6. Sistema de alerta al viajero

Los desastres naturales y los provocados por el hombre son difíciles de predecir y a la vez, difíciles de manejar si se tratan con métodos y tecnologías de comunicación tradicionales. El incidente en Nepal [13] ha demostrado una vez más los desafíos que tienen los departamentos nacionales cuando intentan reunir con éxito información sobre el paradero de sus ciudadanos en un país extranjero, cuantificando con precisión sus números o incluso ponerse en contacto con ellos para proporcionarles información instantánea donde se puede obtener ayuda, o ayudar a su evacuación.

Este sistema [14] se centra en cómo es posible localizar y rastrear a las víctimas, encontrarlas, poder organizar una evacuación rápida o gestionar equipos de rescate locales. Todo ello basado en infraestructuras existentes combinadas con nuevas alertas y métodos de recopilación de información. El sistema se ha utilizado con éxito durante la primavera árabe en varios países, después del terremoto de Chile en 2010, y también proporcionó información valiosa sobre el paradero y las cantidades exactas de los viajeros tras el terremoto de Nepal en 2015.

2.2. Conceptos relacionados

2.2.1. Plataforma de Gobierno Electrónico - EGob

El desarrollo del Gobierno Electrónico del Estado Uruguayo [2] es una actividad permanente que en forma progresiva va incrementando y perfeccionando los servicios que brinda a la población y al propio Estado. Dicho desarrollo se basa en la existencia de una plataforma que proporcione el contexto tecnológico sobre el que se desarrollará el Gobierno Electrónico.

Los usuarios de ésta plataforma pueden ser tanto los ciudadanos como también los propios organismos que componen el Estado Uruguayo. En cuanto a los servicios que dispone, pueden ser ofrecidos de forma individual o conjunta entre varios organismos y además existen servicios de infraestructura propios de la plataforma

El aspecto de la seguridad es muy importante para el desarrollo de esta plataforma, ya que de esta manera genera la confianza indispensable para el desarrollo del Gobierno Electrónico. En consecuencia se ha de tener especial cuidado en asegurar tanto la privacidad de la información de los ciudadanos, como el derecho al acceso a la información pública.

En la Figura 2.5 se pueden observar los distintos componentes de la Plataforma de Gobierno Electrónico e identificar los principales:

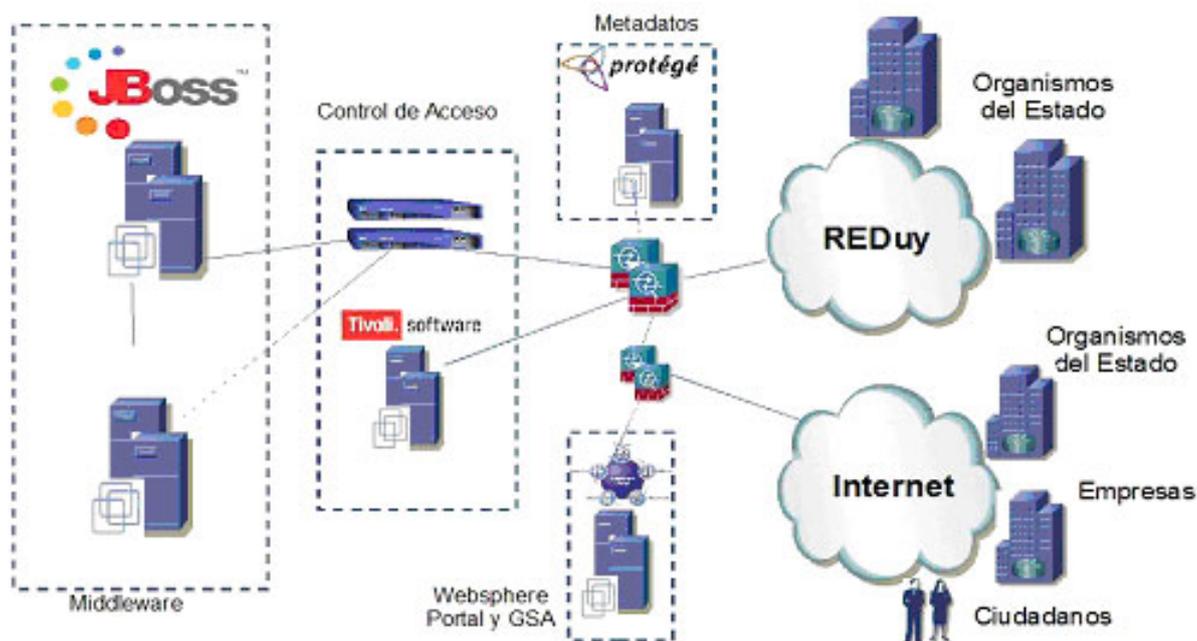


Figura 2.5: Esquema PGE. Tomado de [2]

- Middleware:** La plataforma de Middleware compuesta por dos ESB³, uno con tecnología Java EE⁴ y otro Microsoft, de manera de lograr acercarse a los distintos organismos con el menor esfuerzo posible y contemplando las tecnologías con las que la mayoría de estos trabajan. Además cuenta con la implementación de los estándares de WS-*[15], la implementación nativa de distintos patrones de diseño y el manejo de mensajería sincrónica y asincrónica.
- Sistema de Seguridad:** El sistema de seguridad de la plataforma de eGov está compuesto por el Sistema de control de acceso, servicios periféricos de seguridad y Sistema de auditoría.
- Metadatos:** La estrategia que AGESIC utilizará en el corto plazo para lograr implementar los metadatos en el Estado uruguayo, será la creación de un Referencial de Modelo de Datos (RMD), mediante el cual se definirán los estándares necesarios para lograr la interoperabilidad. AGESIC recomienda a los organismos

³ESB - Enterprise Service Bus

⁴Java EE - Java Platform, Enterprise Edition

que el diseño de nuevas bases, se haga considerando la definición establecida para facilitar el trabajo futuro.

2.2.2. LBS - Location Based Services

Los servicios de locación (LBS) [16] son básicamente, información, alertas y servicios de entretenimientos, accesibles con pc, dispositivos móviles a través de la red OTA (Over the air). Facilita hacer uso de la posición geográfica del dispositivo para diversos servicios. Los LBS se refieren a los servicios que hacen uso de la ubicación del usuario, para agregar valor al servicio como un todo. Los terminales pueden ser fijos o móviles, donde ambos reciben o transmiten datos. Incluyen dispositivos inalámbricos, computadoras portátiles, dispositivos de navegación portátiles y sistemas integrados. La información de ubicación del usuario consiste en coordenadas X,Y generadas por cualquier tecnología de determinación de ubicación (LTD), éstas tecnologías generalmente requieren modificaciones en las redes o teléfonos móviles, y en algunos casos ambos. Las principales categorías de servicios LBS incluyen, servicios de emergencia y seguridad, comunidades y entretenimiento, información y navegación, seguimiento y monitoreo, y comercio móvil. Los sistemas LBS ha generado mucho interés en los últimos años, como para que los operadores móviles mejoren su oferta de servicios, las predicciones de uso han generado mucho interés y expectativas, para nuevos desarrolladores de aplicaciones y servicios. Estos servicios se pueden dividir en varias categorías, como los servicios Pull vs Push; Servicios orientados a personas frente a dispositivos; activos vs pasivos.

Los servicios Push realizan un seguimiento de la ubicación actual del usuario para servicios tales como alertas de seguridad, actualizaciones de noticias, etc. Dichos servicios requieren la actualización de la ubicación en tiempo real del servidor de ubicación. Los servicios Pull, se basan en la demanda del usuario y, por lo tanto, no requieren la actualización continua de la ubicación real del usuario. Por ejemplo búsqueda de puntos de interés (POI), geocodificación o geocodificación inversa.

Los servicios orientados a la persona, comprende todas las aplicaciones en la que un servicio se basa en el usuario. El enfoque de la aplicación es posicionar una persona o utilizar la posición para mejorar un servicio. Los servicios orientados a dispositivos son externos al usuario, donde en lugar de enfocar la demanda de una persona, se podría ubicar un objeto (automóvil, ómnibus) o un grupo de personas (ejemplo una

flota). En las aplicaciones orientadas a dispositivos, la persona u objeto que se encuentra normalmente no controla el servicio, por ejemplo, rastreo de automóviles para la recuperación de robos. En servicios activos, el usuario inicia la solicitud de servicio, mientras que en los servicios pasivos un tercero localiza un usuario (locate) a petición de otro usuario (localizador). Los servicios típicos de localización pasiva son, búsqueda de amigos, juegos basados en la ubicación, etc.

Los sistemas de navegación y las aplicaciones informativas LBS, han sido útiles en navegación paso a paso, búsqueda de puntos de interés, búsqueda de ubicación, visualización de mapas, etc.

2.2.2.1. Arquitectura LBS

Debido a la variedad de tecnologías de posicionamiento, el modelo debe ser independiente de la implementación de determinación de ubicación. También debería poder funcionar cuando el usuario cambia de una tecnología de posicionamiento a otra. La Figura 2.6 muestra una arquitectura genérica del ecosistema de LBS.

Los componentes incluyen el posicionamiento, la red de comunicación y los proveedores de contenido como requisitos básicos de infraestructura. El posicionamiento puede hacerse desde el dispositivo del cliente utilizando un GPS basado en satélite o un servicio de posicionamiento de red. Posteriormente, el cliente móvil envía la solicitud de servicio, que contiene los objetivos del servicio y la posición a través de la red de comunicación. El proveedor de contenido / datos es el motor de ubicación real que brinda varios servicios de LBS.

Además, el middleware se puede agregar a la arquitectura genérica para garantizar la privacidad de la ubicación, la gestión de suscripciones y la responsabilidad de las transacciones.

2.2.3. MOM - Message Oriented Middleware

Un middleware orientado a mensajería(Message-Oriented-Middleware o MOM) [18] se puede definir como un tipo de middleware que soporta envío y recepción de paquetes de datos de forma sincrónica o asincrónica entre diferentes participantes. Un MOM soporta los modelos de comunicación sincrónica (de RPC) y asincrónica. En el último caso, el cliente no debe esperar a que esté finalizado el pedido. El cliente puede continuar la ejecución sin tener en cuenta el estado del servicio o procedimiento invo-

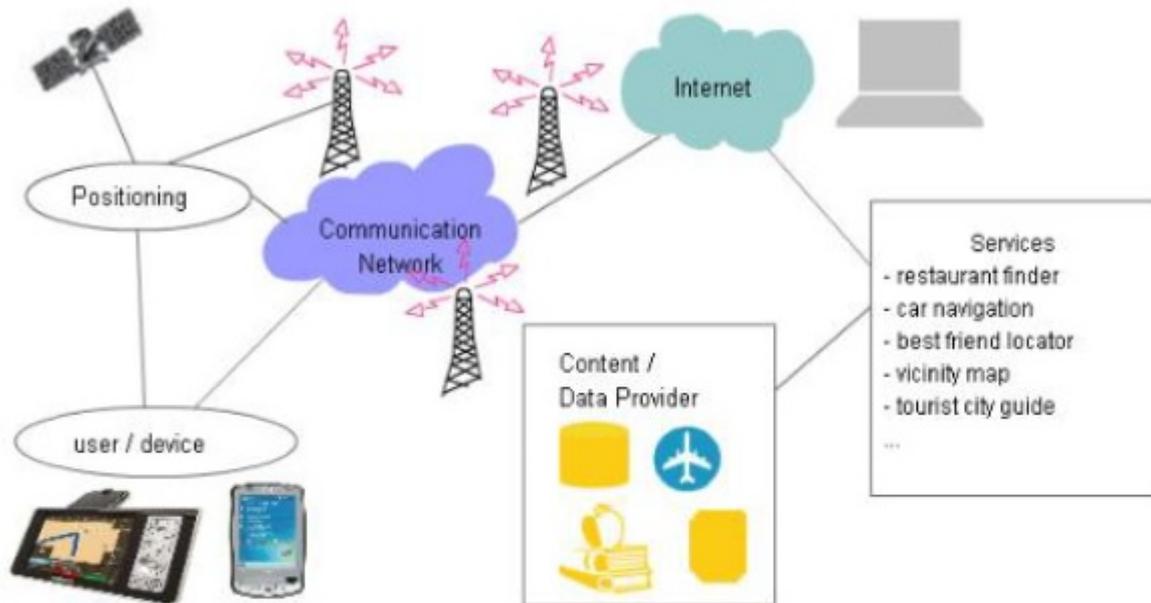


Figura 2.6: Arquitectura LBS. Tomado de [17]

cado. Puede eventualmente obtener el mensaje de respuesta en cualquier momento de la ejecución (y si está listo el pedido) mediante una operación pull o push.

En rasgos generales, provee diversas características:

- Transformación de formatos de datos
- Bajo acoplamiento de las aplicaciones. La capacidad de conectar aplicaciones sin tener que adaptar los sistemas origen ni destino.
- Procesamiento en paralelo de los mensajes
- Soporte para niveles de prioridad de los mensajes.
- Alta cohesión. Es consecuencia del bajo acoplamiento.
- Confiabilidad

- Alta disponibilidad. MOM no requiere que haya disponibilidad en todos los subsistemas al mismo tiempo. La falla en uno de ellos no hace que se caiga todo el sistema.
- Alta velocidad

2.2.4. JMS

La API JMS [19] forma parte de la plataforma de Java EE. En este contexto, la mensajería es un método de comunicación entre dos pares de componentes de software o aplicaciones. Esto difiere del correo electrónico ya que este es un método de comunicación entre personas y no entre aplicaciones.

Se dice que el sistema de mensajería sigue una arquitectura punto a punto ya que un cliente de la mensajería puede enviar mensajes y recibir mensajes de cualquier otro cliente, para esto a cada cliente se le debe brindar un mecanismo para crear, enviar, recibir, y leer mensajes.

Este tipo de comunicación por mensajería tiene la ventaja de estar débilmente acoplado, es decir que un emisor envía un mensaje al receptor y este lo puede obtener. El hecho de que estén débilmente acoplados implica que no es necesario que ambos estén disponibles para poder intercambiar los mensajes. Incluso el emisor no tiene por qué saber características particulares del receptor y viceversa, únicamente requieren conocer cuál es el tipo o formato de los mensajes que están intercambiando.

Esto difiere de los sistemas altamente acoplados, como Remote Method Invocation (RMI), donde una de las aplicaciones requiere conocer los métodos de la aplicación remota.

Además de ser débilmente acoplado, la API JMS es:

- **Asíncrona:** Un proveedor de mensajes puede entregar los mensajes a sus destinatarios en cuanto el mensaje llega. Además un cliente no necesita solicitar el mensaje para que le llegue.
- **Segura:** La API garantiza que un mensaje llega una y por única vez a un destinatario. Asegura que los mensajes llegan y no son repetidos.

Existen varios escenarios donde es recomendado o por lo menos se adapta el uso de mensajería como la de JMS. Entre ellos, lo más relevantes son:

- Un proveedor de información no necesita tener a los clientes conectados para poder enviar notificaciones. A cada cliente suscripto le va a llegar la notificación correspondiente cuando se conecte y esté disponible.
- El proveedor de información puede enviar una notificación y continuar con su ejecución, no es necesario que este reciba una respuesta por parte del cliente.
- Un proveedor que implemente los métodos necesarios para la comunicación con la API puede comenzar a enviar mensajes y no necesita realizar grandes cambios en su arquitectura. Además existe también compatibilidad del lado del cliente, solamente requiere consumir los servicios expuestos.

2.2.4.1. Arquitectura de la Api JMS

Una aplicación JMS usualmente se compone de las siguientes partes:

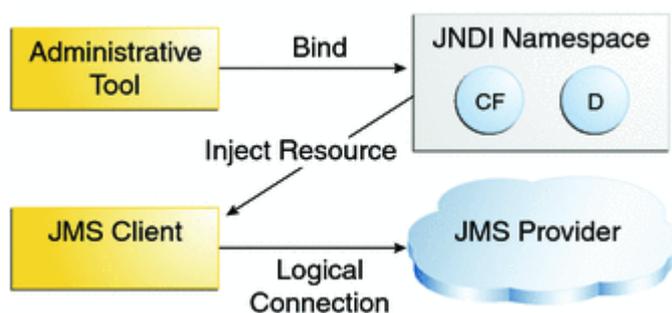


Figura 2.7: Arquitectura JMS[19]

- **Proveedor JMS:** Se trata de una aplicación de mensajería que implementa las interfaces de la API JMS y provee características administrativas y de control.
- **Cliente JMS:** Son programas o componentes que producen o consumen mensajes.
- **Mensaje:** Objeto que se utiliza para enviar en la comunicación
- **Objetos de administración:** Objetos JMS preconfigurados por un administrador que son utilizados por los clientes.

2.2.5. Broker de mensajería

Un broker de mensajería es un patrón arquitectónico que permite el desacoplamiento entre la aplicación que genera mensajes y las que los reciben. Se encarga además de la validación, transformación y ruteo de los mensajes.

La utilización de este patrón permite disminuir el conocimiento que deben tener entre las aplicaciones que se comunican, incluso no siempre será posible comunicarse con una aplicación, por lo que puede llegar a ser necesario el envío de mensajes por otros medios.

La utilización de un broker de mensajería brinda funcionalidades como:

- Dirigir un mismo mensajes a uno o más destinos diferentes. Esto nos permite agrupar varios clientes de una misma categoría y enviarles un mismo mensaje a todos.
- Permite interactuar con distintas bases de datos que permitirá almacenar información de los clientes y/o obtener datos de estos antes de enviar una notificación.
- Responder a eventos o errores
- Proveer un ruteo de los mensajes basado en su contenido o en sus tópicos empleando el modelo de publish-suscribe

2.2.6. Patrón Publish-Suscribe

En escenarios donde es necesario enviar información entre entidades, la mensajería asíncrona es una forma eficaz de desacoplar a los remitentes de los consumidores y evitar bloquear al remitente para que espere una respuesta. Sin embargo, el uso de una cola de mensajes dedicada para cada consumidor no es efectivo para muchos consumidores.

Una forma de solucionar esto es utilizar el patrón Publish-Suscribe [20], es decir introducir un subsistema de mensajería asíncrona que incluya lo siguiente:

- Un canal de mensajería de entrada utilizado por el remitente. El remitente (publicador) empaqueta los eventos en mensajes, mediante un formato de mensaje conocido, y envía estos mensajes a través del canal de entrada.
- Un canal de mensajería de salida por consumidor. Los consumidores se conocen como suscriptores.

- Un mecanismo para copiar cada mensaje del canal de entrada a los canales de salida para todos los suscriptores interesados en ese mensaje.

En la Figura 2.8 se muestran los componentes lógicos de este patrón:

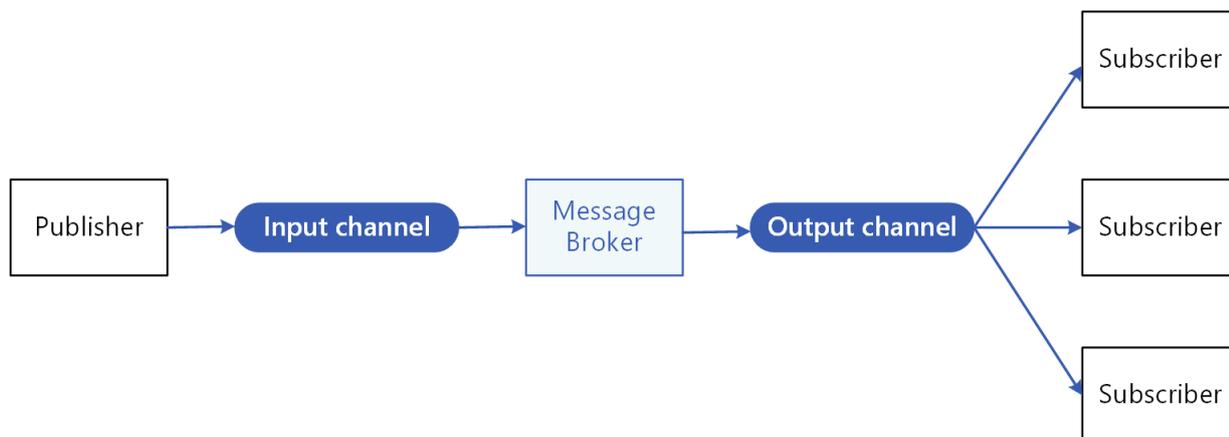


Figura 2.8: Diagrama patrón Publish-Suscribe[20]

Las principales ventajas de utilizar un patrón como este son:

- Desacopla los subsistemas que necesitan comunicarse, los mensajes se pueden administrar correctamente incluso si uno o más receptores están desconectados.
- El remitente puede enviar rápidamente un solo mensaje al canal de entrada y luego volver a sus responsabilidades de procesamiento principales.
- Permite una integración más sencilla entre sistemas que utilizan diferentes plataformas, lenguajes de programación o protocolos de comunicación.

2.3. Tesis de Maestría

El desarrollo de este proyecto de grado basa su arquitectura en la solución propuesta en la Tesis de Maestría denominada *Extensión Espacial de Middleware orientado a mensajería para dar soporte a servicios basados en geolocalización* [1]. A continuación se muestran los principales puntos de dicho trabajo.

Esta tesis se centra en aprovechar las potencialidades de los dispositivos móviles, en particular el uso del GPS ya que es una herramienta que permite personalizar el envío de información a las personas según su ubicación. Se plantea proveer de un LBS

enmarcado en la Plataforma de Gobierno Electrónico de Uruguay, mantenida y gestionada por la AGESIC que ayude al Estado uruguayo a satisfacer mejor las necesidades de la población, brindando información a partir de la ubicación de los ciudadanos mediante el envío de notificaciones utilizando algún mecanismo de mensajería confiable.

Las notificaciones pueden abarcar diferentes categorías como por ejemplo: emergencias, meteorología, cursos temáticos, accidentes de tránsito, locales comerciales, comida, inseguridad, cultura, medio ambiente, vivienda, etc. Los encargados de enviar estas notificaciones pueden ser muchos y muy variados. Se plantean varios ejemplos que van desde proveedores públicos pertenecientes al estado uruguayo pero también proveedores privados. Algunos ejemplos pueden ser el Ministerio del Interior, INUMET, SINAIE, Cámara de Comercio, Intendencias departamentales, PedidosYa, CityCop entre otros.

Como propuesta de solución plantea un middleware orientado a mensajería o MOM, definido como una infraestructura de middleware que soporta envío y recepción de paquetes de datos de forma sincrónica o asincrónica entre diferentes participantes. También se detallan los actores que interactúan, los formatos de los mensajes y patrones de mensajería que definen mecanismos de distribución y entrega de los mensajes.

El objetivo general de la tesis fue proponer una plataforma tipo MOM extendida con capacidades geoespaciales que brinde soporte al desarrollo de LBS en un contexto de e-gov. Para esto fue necesario realizar un análisis de los requerimientos generales, actores, y escenarios posibles de uso de la plataforma.

2.3.1. Solución planteada

La plataforma a construir debe permitir la publicación de eventos con datos de carácter geográfico por parte de organismos del Estado uruguayo, que disponen de datos de interés para la población. Los datos pueden tener asociada una geometría específica que se puede tratar localización geográfica puntual con coordenadas específicas (ejemplo: latitud y longitud); es el caso de un siniestro vial. En otros casos pueden tener un área de influencia o de incidencia, representada por un polígono con puntos en el espacio geográfico. Es el caso de un incendio forestal, que abarca un área amplia y con límites específicos. Y, por último, otros datos pueden representarse mediante una polilínea en el espacio, por ejemplo, si se refiere a una ruta o camino de una ciudad.

Asimismo, el evento puede contener un timeout o tiempo de vencimiento según el cual, pasado dicho tiempo desde la emisión del evento, pierda validez y deba ser eliminado.

Por otra parte, existen usuarios que desean recibir notificaciones o eventos, que pueden depender de su localización actual o que se produzcan en cierta área geográfica de interés. En el primer caso, dicha área puede comprender un buffer de distancia a la ubicación del usuario (ejemplo: recibir incidentes de inseguridad que ocurran en un rango de 300 metros de mi posición actual) o un área con límites fijos y donde su posición pertenezca al interior del área (ejemplo: mi departamento actual o mi barrio actual). En el segundo caso, dicha área es arbitraria, permitiéndole al usuario definir un polígono sobre un mapa y recibir notificaciones que procedan de esa área independientemente de si su ubicación se encuentra en ella (ejemplo: deseo recibir eventos que ocurran en el LATU).

Esta plataforma debe contar con las capacidades de un MOM, es decir, un sistema de mensajería que permita entregar de forma fiable y segura los mensajes de los productores a los usuarios, bajo diversos mecanismos de comunicación. La plataforma debe recibir los eventos de los organismos y alojarlos temporalmente hasta que los clientes estén disponibles para ser notificados, utilizando el patrón Publish-Suscribe. En este esquema, las suscripciones de los usuarios comprenden las restricciones espaciales que imponen a las notificaciones que pretenden recibir. El sistema de mensajería debe cruzar la ubicación puntual o área geográfica de los eventos y las zonas de interés de los usuarios, y si la intersección es diferente a vacío, realizar las notificaciones correspondientes.

Del análisis de los actores se desprenden tres tipos de actores:

- **Proveedores de datos:** Se pueden categorizar entre proveedores públicos o privados. En el Estado uruguayo existen múltiples organismos que generan datos de diversa índole, para ser consumidos por otros organismos o por el público en general. Además de los organismos del Estado, podrían agregarse otros proveedores de datos de la órbita privada que publiquen datos de interés público y de naturaleza geoespacial para generar notificaciones oficiales
- **Usuarios finales:** Los usuarios finales de la plataforma pueden ser cualquier usuario del público general. El único requisito del usuario es registrarse en el sistema e ingresar sus preferencias personales de las notificaciones que desea re-

cibir. Se requiere como mínimo que el usuario este registrado a nivel del Estado con su cédula de identidad.

- **Desarrollador de aplicaciones:** Es un usuario intermedio entre la plataforma y los usuarios finales y es el encargado de implementar la aplicación para el cliente móvil, de forma de permitir la entrada y lectura de datos utilizando interfaces acordes y bien definidas.

Los tipos de eventos que los proveedores de datos pueden generar son muy variados. Con el fin de dar una idea general de este aspecto, se tiene el Cuadro 2.1 extraído de la tesis de maestría donde se pueden ver los distintos tipos de eventos, junto con los posibles proveedores de datos para estos eventos y la clasificación de los eventos según la prioridad.

Tipo de evento	Proveedor de datos	Prioridad
Meteorológico	INUMET, SINAЕ, intendencias departamentales	Baja o Media: consulta de pronostico. Alta: alertas meteorológicas.
Criminalidad de inseguridad	Ministerio del Interior, City-Cop	Alta: episodio cercano de inseguridad, circulación en zona de riesgo.
Rutas y caminos	Dirección Nacional de Policía Caminera, intendencias departamentales	Alta: desvió de ruta, alta densidad de tráfico, exceso de velocidad.
Incendio	SINAЕ	Media o Alta según las dimensiones del incendio
Publico	Intendencias departamentales, Socio Espectacular	Baja: información futura de cursos, obras teatrales, conciertos, etc. Media: evento ocurriendo en cierto lugar y con alta densidad de personas (ejemplo: partido de futbol). Alta: Cortes importantes de caminos (carreras o marchas de manifestación).
Accidente vial	Dirección Nacional de Policía Caminera, UNASEV, AGESIC	Media: accidentes en zona especifica. Alta: accidentes a corta distancia del usuario.
Locomoción	Intendencias departamentales	Baja: horarios de líneas de ómnibus por parada. Media: móviles de taxis según buffer de usuario.
Locales comerciales	Intendencias departamentales, Cámara de Comercio	Media: apertura o cierre de local comercial, ubicación más cercana a sucursal.

Vivienda	Intendencias departamentales, Cámara de Comercio	Baja: venta de inmuebles, planes de vivienda estatales.
Estacionamiento	Intendencia departamental	Media o alta: búsqueda de un lugar disponible cercano al usuario.
Medio ambiente	OAN	Media: notificación de variables medioambientales. Alta: violación de parámetros medioambientales en cierto padrón rural.
Comidas	PedidosYa	Baja: nuevos platos o restaurantes.

Cuadro 2.1: Tabla 2: Clasificación de eventos. Tomado de [1]

Otro aspecto en el que se centró el planteo de la solución es definir cuales serían las operaciones a realizar sobre la plataforma y que actores serán los encargados de efectuar estas operaciones. Estas operaciones se deben hacer mediante interfaces bien definidas y con estructura, parámetros y resultados bien establecidos. Se pueden clasificar las distintas operaciones según el actor que las invoca:

- **Operaciones de proveedores**

- Registrar proveedor
- Registrar tipo de evento
- Publicar evento

- **Operaciones de usuarios**

- Registrar usuario
- Login de usuario
- Suscribirse a eventos
- Notificar alerta

2.3.2. Propuesta de la solución

En base a los requerimientos planteados se propone una arquitectura de solución que se compone de dos grandes componentes, GeoMOM y broker de mensajería. Se

Proveedores de datos

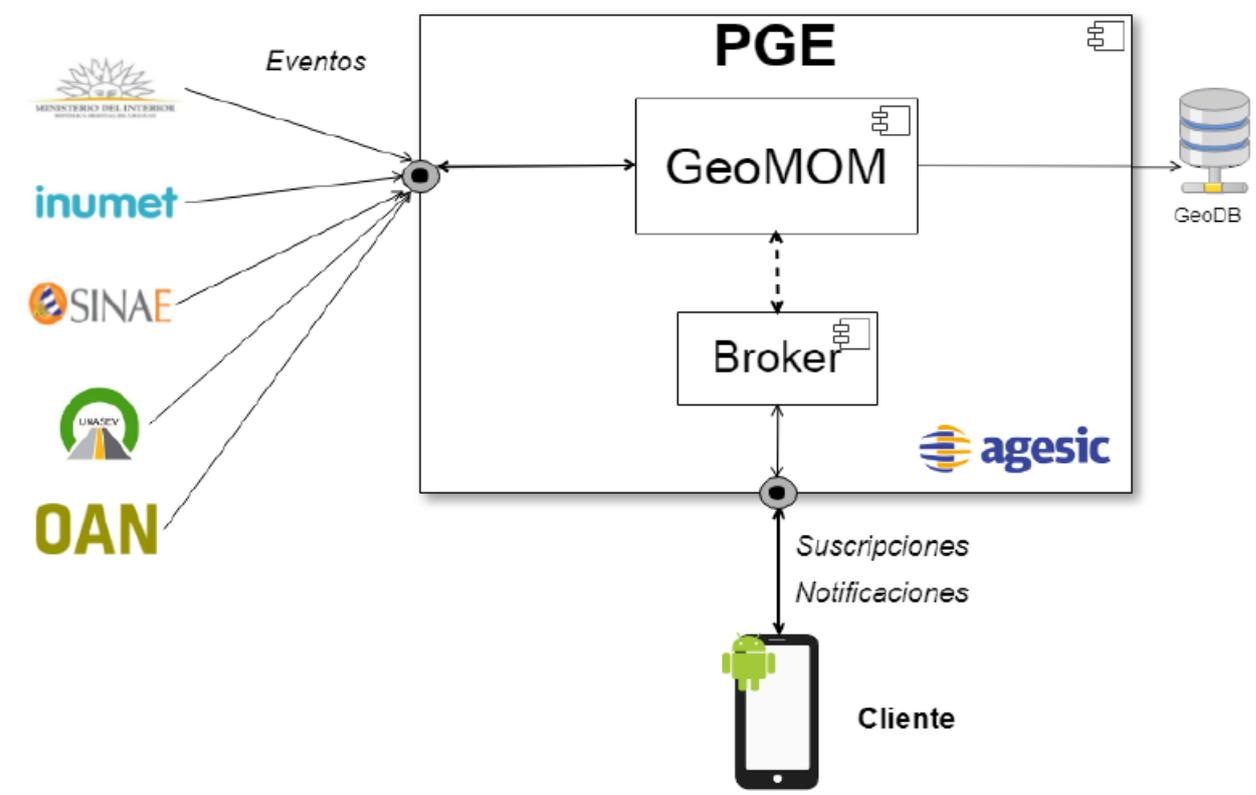


Figura 2.9: Arquitectura general. Tomado de [1]

propone el esquema arquitectónico en la Figura 2.9 en el contexto de la Plataforma de Gobierno Electrónico (PGE).

Se puede observar algunos de los ejemplos vistos de proveedores de datos y como sería la distribución de los componentes de la plataforma y las interfaces para los proveedores y clientes.

En esta propuesta, los Proveedores de datos se comunican a través de una interfaz con el GeoMOM para enviar los eventos que deseen. El GeoMOM cuenta con una base de datos que contiene información georreferenciada de las suscripciones de los clientes. El Broker de mensajería se encarga de la comunicación con los clientes dónde recibe las suscripciones de estos. Los clientes, además de realizar suscripciones a Topics a través del Broker de mensajería, también recibe desde el Broker las notificaciones que le correspondan según estas suscripciones.

2.3.3. Arquitectura

Luego de planteada la propuesta de la solución se introduce la arquitectura de la plataforma.

En la Figura 2.10, se pueden apreciar las estructuras del sistema de mensajería que forman parte del GeoMOM.

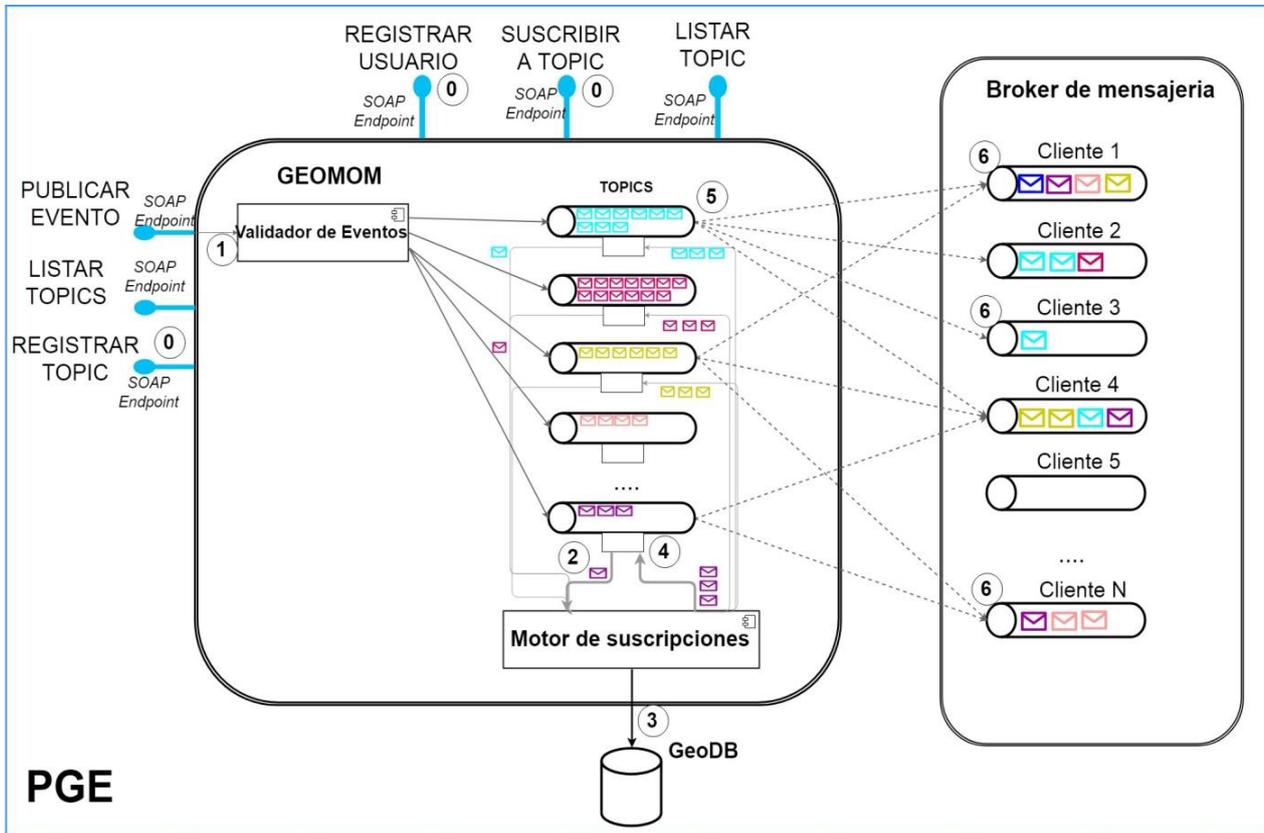


Figura 2.10: Arquitectura del GeoMOM. Tomado de [1]

El GeoMOM se compone de varios topics que reciben los mensajes entrantes de eventos de los organismos que viajan a través de la PGE y un motor de suscripciones que procesa los mensajes de los topics. Cada sobre de distinto color en la figura representa un tipo de evento distinto. Por último el Validador de Eventos se encarga de validar la correctitud y correspondencia del mensaje.

Capítulo 3

Propuesta de solución

En este capítulo se introduce el diseño de la propuesta de solución del proyecto, tomando como base la tesis de referencia [1] que fue presentado en la Sección 2.3 y extendiéndola con un componente de administración.

La propuesta de solución toma varios de los componentes definidos en la tesis de referencia, pero incorpora cambios a las denominaciones de los componentes planteados. Por esta razón, la plataforma a diseñar e implementar consta de dos módulos separados: por un lado, el módulo GeoMOM que se compone por el Broker de mensajería y el GeoMOM Central; por otro lado, el segundo módulo que se corresponde con la Administración Web de la plataforma. Ambos módulos son los definidos en la solución e integrados a la Plataforma de Gobierno Electrónico.

A continuación se definen los componentes centrales de la plataforma, sus requerimientos generales que se traducen en los principales casos de uso, así como la definición de los requerimientos no funcionales. Se definen los actores involucrados y se muestran los diagramas correspondientes para una mejor comprensión del diseño.

3.1. Actores

En la Sección 2.3.1 se presentó la solución planteada por la tesis de referencia y se mencionaron los tres tipos de actores que se identificaban para la plataforma.

Con la propuesta de extensión se incorpora un actor más, los Administradores de la plataforma. Por lo tanto los actores de nuestra propuesta de solución son los siguientes:

- **Proveedores de datos:** serán los encargados de generar los eventos en la plata-

forma con el fin de enviar información a los Usuarios. Se comunicarán con la plataforma a través de servicios web.

- **Clientes o Usuarios finales:** cada proveedor de información tendrá una aplicación, generalmente móvil, donde los Usuarios podrán suscribirse a los temas existentes y así recibir las alertas generadas por los Proveedores en los casos que corresponda.
- **Administrador:** la plataforma tendrá un componente con funcionalidades de administración. El Administrador podrá realizar tareas de gestión de los datos existentes (Proveedores, Topics, Clientes) y además contará con listados y gráficos estadísticos que le permita tomar decisiones.
- **Desarrolladores de aplicaciones:** los proveedores que deseen utilizar la plataforma deberán adaptar sus aplicaciones existentes o generar nuevas para que sus Clientes reciban los eventos generados por ellos. Es por esto que los Desarrolladores serán actores momentáneos de la plataforma durante el proceso de desarrollo de la aplicación para el proveedor.

3.2. Requerimientos y casos de uso

A continuación se detallan los requerimientos tanto funcionales como no funcionales que deberá cumplir la solución diseñada.

3.2.1. Requerimientos funcionales

Se definen los requerimientos funcionales los que dan lugar a los casos de usos. Cada uno está definido según el actor que lo ejecuta y se listan a continuación.

Cliente

3.2.1.1. Listar topics

El cliente lista los topics existentes en la plataforma.

3.2.1.2. Registrar usuario

Se registran usuarios a la plataforma. Se solicita su correo electrónico y una contraseña para ingreso. El usuario está habilitado para registrarse a los topics disponibles.

3.2.1.3. Suscribir a topic

El cliente registrado lista los topics disponibles y selecciona uno que sea de su interés. Debe indicar la forma en que será notificado (notificación Push, email, SMS) y además una zona de interés geográfico marcando en un mapa o su ubicación en tiempo real.

3.2.1.4. Dar localización

En caso de existir una suscripción por ubicación en tiempo real, la aplicación que utiliza el cliente indica, cada cierto intervalo de tiempo, la localización exacta mediante el GPS del dispositivo móvil.

3.2.1.5. Cancelar suscripción topic

El cliente elige uno los topics al que está suscrito y elimina su suscripción.

3.2.1.6. Consultar Alertas

El cliente consulta todas las alertas disponibles, es decir, las leídas y las no leídas.

Proveedor

3.2.1.7. Publicar evento

El proveedor de información publica un evento en un topic existente en el sistema. Además deberá indicar datos como fecha, fecha de vencimiento, descripción y zona de incidencia del evento.

3.2.1.8. Listar topics

El cliente lista los topics existentes en la plataforma.

Administrador

3.2.1.9. Registrar topic

El proveedor solicita al administrador de la plataforma el registro de un topic. El administrador ingresa un nombre y una descripción del mismo. El topic queda disponible para el envío de notificaciones y para el registro de usuarios.

3.2.1.10. Modificar topic

El administrador ingresa a la administración web selecciona un topic y lo modifica.

3.2.1.11. Listar topics

Se listan los topics existentes en la plataforma.

3.2.1.12. Obtener clientes inactivos

El administrador ingresa a la administración web lista los clientes inactivos con un filtro de fecha. El sistema le indica la lista de clientes inactivos desde la fecha que indicó en el filtro

3.2.1.13. Eliminar cliente

Si un cliente está inactivo por mucho tiempo, es decir que recibe alertas pero no las lee, se puede determinar que este cliente no está interesado en recibir alertas o no sigue utilizando la plataforma. En este caso, el Administrador puede eliminar el cliente y así liberar los recursos de la plataforma que fueron reservados para él.

3.2.1.14. Obtener mensajes enviados

El administrador ingresa a la administración web, y lista los mensajes enviados. El sistema retorna la lista de mensajes enviados, mensajes leídos, datos correspondientes a los clientes y topics involucrados.

3.2.1.15. Obtener mensajes entregados

El administrador ingresa a la administración web, lista los mensajes efectivamente entregados a los clientes junto a la información correspondiente al cliente, topic in-

volucrado, fecha de enviado y fecha de entregado.

3.2.1.16. Cantidad de matching geométrico

El usuario administrador ingresa a la administración web y obtiene información de los eventos generados, cantidad de suscripciones y cantidad de mensajes enviados por topic. Además se grafica el porcentaje de matching geográfico según la fórmula 3.1.

$$\%Matching = \frac{mensajesEnviados}{suscripciones * eventos} * 100 \quad (3.1)$$

3.2.1.17. Registrar proveedor

El administrador ingresa a la administración web y registra un proveedor nuevo. El proveedor no debe estar ingresado previamente al sistema.

3.2.1.18. Eliminar proveedor

El administrador ingresa a la administración web, selecciona un proveedor y lo da de baja. El proveedor debe existir en el sistema.

3.2.1.19. Modificar proveedor

El administrador ingresa a la administración web, selecciona un proveedor y lo modifica.

3.2.1.20. Extracción de datos

Se extrae toda la información registrada en la plataforma en formato estándar.

3.2.2. Requerimientos no funcionales

3.2.2.1. Portabilidad

Uno de los principales objetivos del proyecto es que las alertas sean recibidas en dispositivos móviles. Por lo tanto, las suscripciones, el manejo de la ubicación geográfica y las alertas deben gestionarse en estos dispositivos.

3.2.2.2. Conectividad

Para el envío de las notificaciones se deberá contemplar la falta de conectividad de los usuarios en dispositivos móviles. Los mensajes serán notificados cuando el usuario adquiera conexión, si corresponde. Esto se debe a que las notificaciones tienen un tiempo de vida limitado para cada topic, con lo cual el mensaje puede estar disponible por tiempo limitado.

3.2.2.3. Seguridad

Dado que intervienen distintos tipos de actores que deben acceder a la plataforma y además esta debe ubicarse dentro de una infraestructura común a otros servicios, la seguridad es un punto importante a tener en cuenta. Deberá utilizarse certificados digitales exigidos por AGESIC, además de autenticación de servicios SOAP para las comunicaciones con los proveedores.

3.2.2.4. Interoperabilidad

La plataforma debe permitir integrar proveedores y clientes de cualquier tipo, de manera rápida y sencilla, implementando simplemente las interfaces correspondientes.

3.2.2.5. Rendimiento y escalabilidad

Se deberá poder extender las funcionalidades provistas en los puntos anteriores, además de poder integrar nuevas, de manera sencilla a la plataforma. Debe manejar una cantidad masiva de proveedores y gran concurrencia de usuarios, y aún así brindar tiempos de respuesta acordes.

3.2.3. Diagrama de casos de uso

Con la finalidad de mostrar todos los casos de uso generados a partir de los requerimientos funcionales, se presenta el diagrama de casos de uso de la Figura 3.1.

En este diagrama se muestra que actores ejecutan las distintas funcionalidades de la plataforma.

Se omitió agregar en el diagrama a los Desarrolladores de aplicaciones ya que no serán actores activos en la plataforma y las funcionalidades que estos ejecutan son las

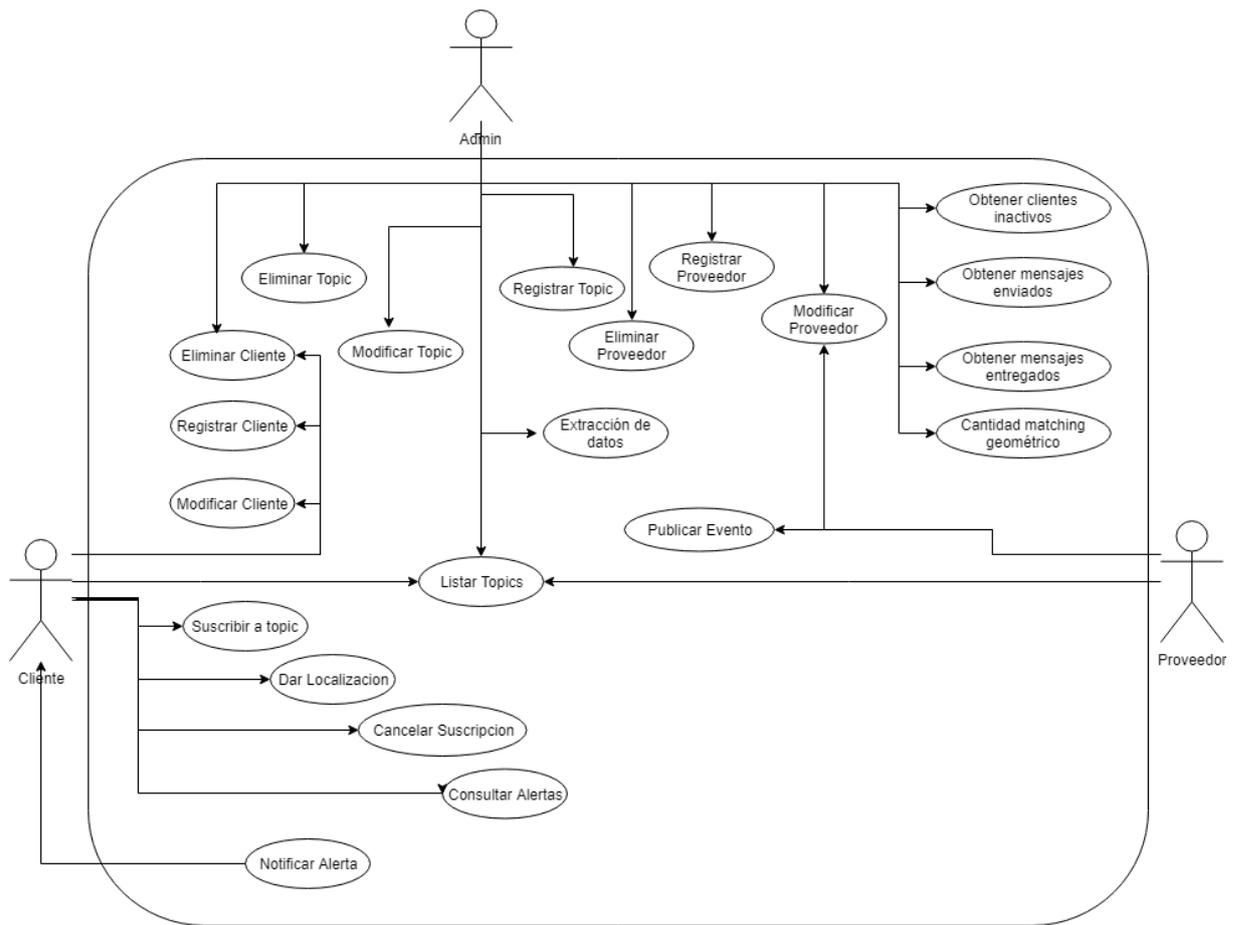


Figura 3.1: Diagrama de casos de uso

mismas que los clientes.

3.3. Arquitectura de la solución

En esta sección se detallan los componentes que conforman la arquitectura. A continuación se describen a nivel general todos los componentes de la solución planteada y su función dentro de la plataforma GeoMOM.

Como se puede observar en la Figura 3.2, la plataforma GeoMOM se compone por el GeoMOM Central y el Broker de mensajería que centralizan las funcionalidades de la plataforma.

El componente GeoMOM Central se encarga de recibir los eventos publicados por los proveedores y validarlos. La comunicación entre el GeoMOM Central y los proveedores de información se realiza a través de servicios web dónde el proveedor dispone de la estructura necesaria para enviar las alertas. El GeoMOM Central, en base

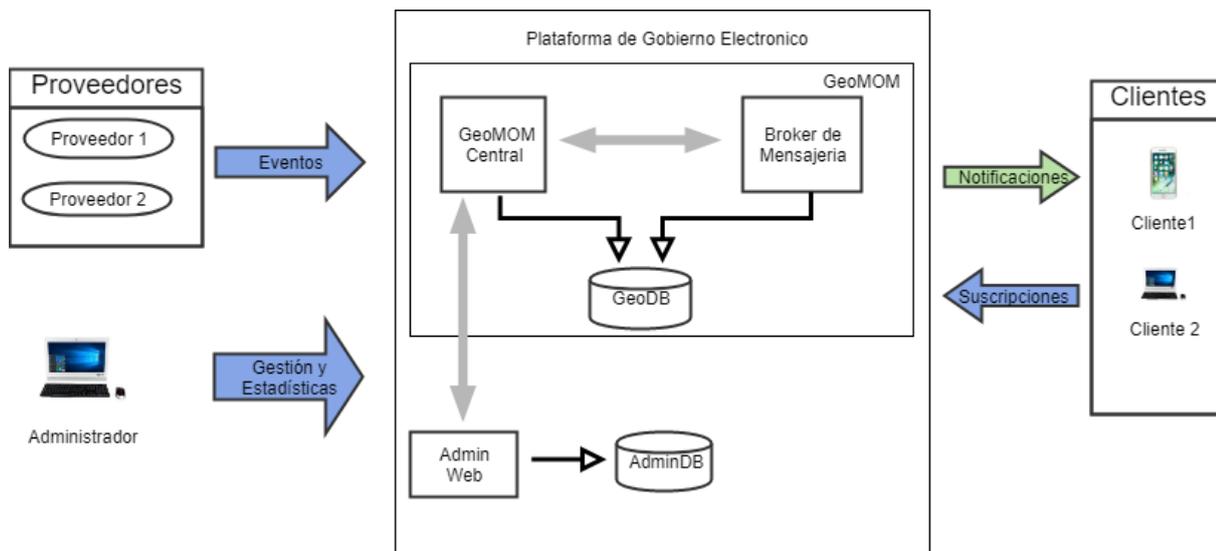


Figura 3.2: Planteo de la solución

al matching geográfico entre el componente geográfico del evento y la suscripción del cliente, determina a qué cliente le corresponde enviar la alerta.

Luego, el Broker de mensajería toma esos mensajes y los envía a la cola de mensaje del cliente que corresponda. Además se encarga de mantener las colas de mensajes de cada cliente y con esto se puede saber en todo momento si los mensajes están siendo recibidos por los clientes y sobre todo asegurarnos que el mensaje será enviado. Está claro que un cliente puede estar recibiendo, en un mismo momento, alertas de múltiples proveedores de información, por lo tanto es importante que exista una depuración activa de las colas de mensajes.

El Broker de mensajería es encargado además de toda la comunicación con el cliente, es decir, recibe las suscripciones, envía las alertas y recibe información de la ubicación en tiempo real.

Tanto el GeoMOM Central como el Broker de mensajería utilizan la misma base de datos para la persistencia de todos los datos necesarios para el funcionamiento de la plataforma, los eventos generados y también los mensajes enviados a los clientes.

Por otro lado, se encuentra la Administración Web que se integra a la Plataforma de Gobierno Electrónico, pero como un componente independiente de la plataforma GeoMOM, con el fin de separar los componentes centrales de los que son puramente de administración.

Para lograr esta separación, el componente de administración cuenta con su propia base de datos y se conecta al GeoMOM a través de servicios web.

Capítulo 4

Descripción de la solución

En el capítulo anterior se planteó la propuesta de solución para la plataforma GeoMOM sobre la base de la tesis de referencia[1]. A continuación se describe en detalle la arquitectura planteada y la descripción de los distintos componentes.

En la Figura 4.1 se puede observar cada componente implementado y la comunicación que existe entre ellos.

En el modelo de distribución de la Figura 4.1 se puede observar la ubicación de cada componente que hace a la solución del sistema. En la parte central de la plataforma se encuentra el GeoMOM Central y el Broker de mensajería ubicados en el mismo servidor de aplicaciones con el fin de lograr una comunicación directa entre ellos. Cada uno de ellos publica distintos servicios para brindar sus funcionalidades a los diferentes actores. El GeoMOM Central publica servicios en tecnologías Rest para la administración y servicios SOAP para los proveedores. Por otro lado, el Broker de mensajería brinda servicios Rest para los clientes y para los desarrolladores de aplicaciones. Tanto el GeoMOM Central como el Broker de mensajería utilizan el mismo esquema de base de datos llamado GEOMOM.

Por otro lado existe otro servidor de aplicaciones que contiene el componente de administración web con su propia base de datos en un esquema llamado GEOMOMADMIN. Este componente utiliza los servicios Rest del GeoMOM Central para obtener los datos necesarios.

A continuación se describe en detalle la arquitectura de los componentes principales de la solución.

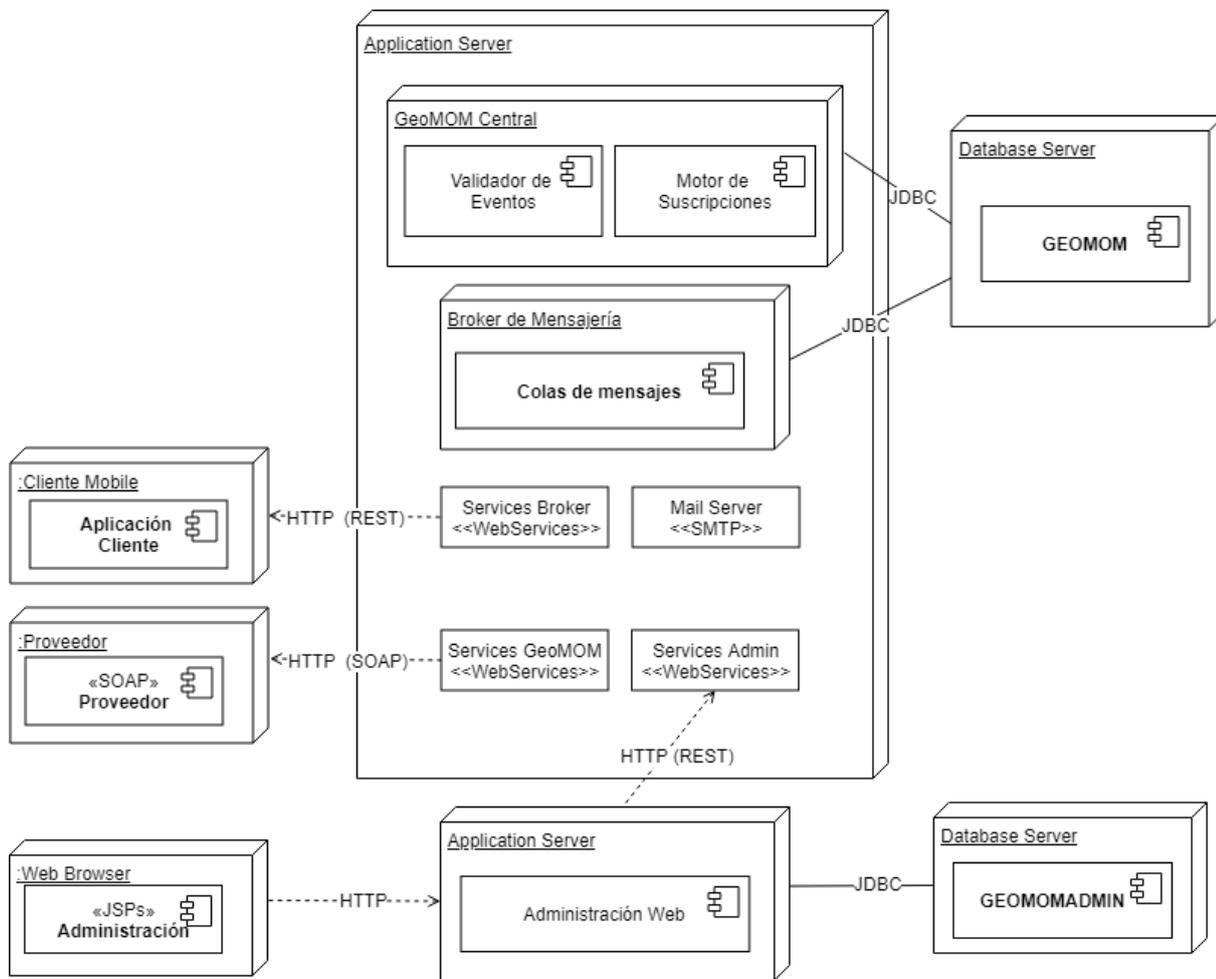


Figura 4.1: Modelo de distribución

4.1. GeoMOM Central y Broker de mensajería

El GeoMOM Central y el Broker de mensajería son los dos componentes principales de la plataforma, donde se centralizan todas las funcionalidades y servicios. En la Figura 4.2 se presenta la arquitectura de estos dos componentes con el fin de apreciar las interacciones que existen entre ellos al momento de procesar una solicitud.

Cada nuevo evento enviado por un Proveedor es validado por el componente Validador de eventos que comprueba que el evento haya sido correctamente generado y cuenta con todos los parámetros necesarios. Dentro del GeoMOM Central existe una cola de mensajes para cada Topic almacenado en el sistema, por lo tanto, cada vez que un proveedor envía un nuevo evento, este se almacena en el tópico correspondiente.

Luego, el Motor de Suscripciones toma el evento y según las suscripciones de los clientes decide a qué clientes debe enviarle el evento. Para los casos en que corresponda, el evento es almacenado en la cola de mensajes correspondiente a cada cliente

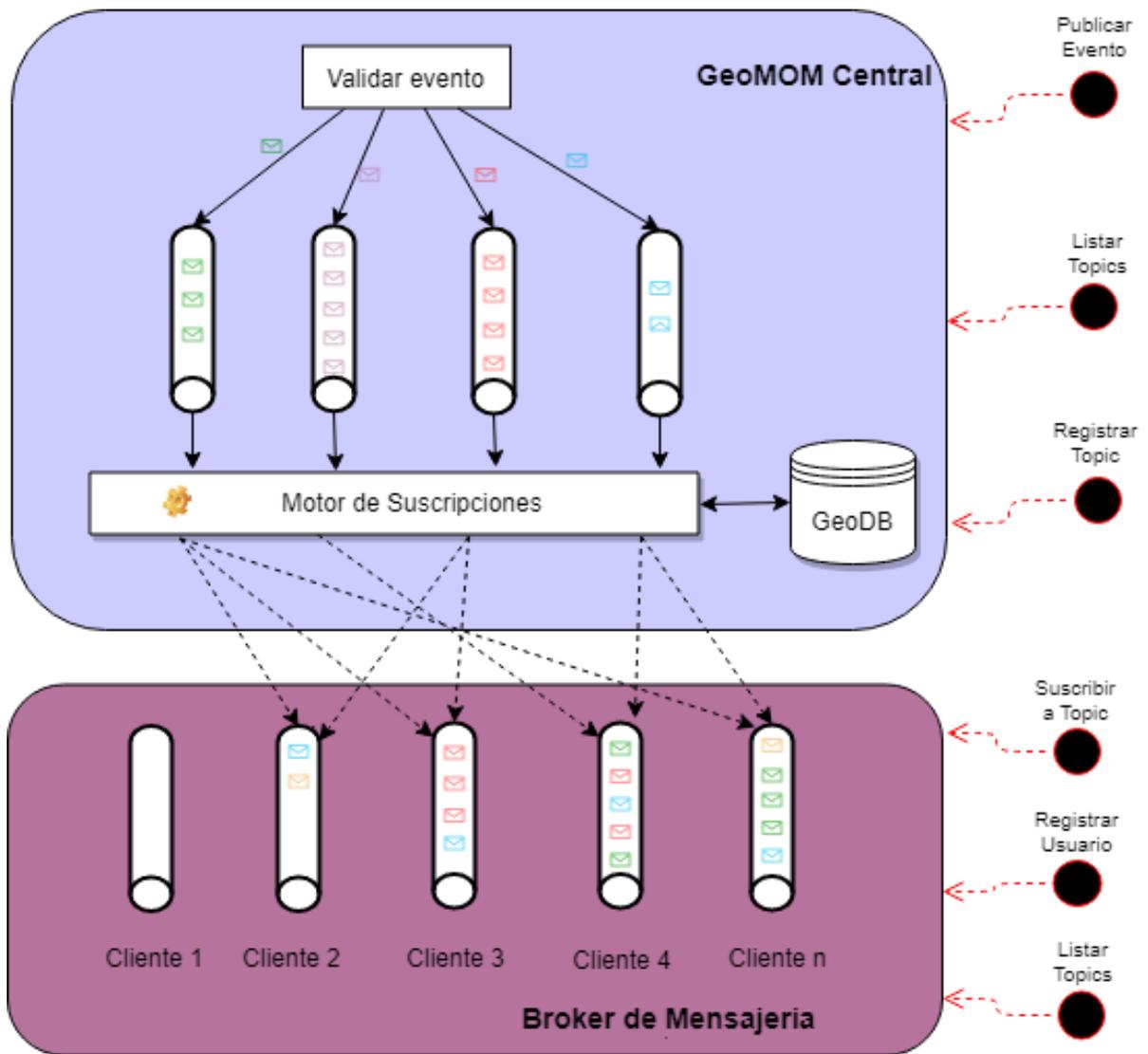


Figura 4.2: Arquitectura del GeoMOM Central y broker de mensajería

dentro del Broker de mensajería.

Con esta arquitectura se logra un mayor control de los recursos de la plataforma y mantener una distribución de los componentes de forma desacoplada.

4.1.1. GeoMOM Central

Gran parte de la lógica de la plataforma se encuentra en el GeoMOM Central, dónde se implementan los casos de uso de los Proveedores y de la Administración, además se recibe la información proporcionada por los clientes a través del Broker de mensajería. Este componente está desarrollado en lenguaje Java.

A continuación se describe el funcionamiento de dos componentes presentes en

el GeoMOM Central.

4.1.1.1. Validador de eventos

Es esencial que los eventos generados por los proveedores estén bien contruidos y cuenten con todos los datos necesarios. Esa es la función que cumple el Validador de eventos. Este toma el evento recibido desde el servicio web publicado para los proveedores y valida que se cumplan los siguientes requisitos de datos:

- **Identificador de Proveedor:** el proveedor que envía el evento debe estar previamente registrado en el sistema.
- **Identificador del Topic del evento:** el evento debe estar asociado a un Topic que exista en el sistema. Los Topics son creados únicamente por el Administrador.
- **Descripción del evento:** es necesario que cada evento cuente con una descripción del evento. Esta información debe ser concisa y de utilidad, ya que es información que luego recibe el usuario.
- **Punto o zona de interés para el evento:** si la zona de interés enviada por el proveedor es una geometría, la lista de puntos que la determinan debe cumplir que la primer coordenada y la última coincidan.
- **Prioridad:** la prioridad seleccionada debe ser una de las existentes en el sistema.
- **Fecha timeout:** la fecha/hora del vencimiento del evento debe ser mayor a la fecha actual.

4.1.1.2. Motor de suscripciones

Una vez que se verifica que el evento está correctamente generado, es necesario procesar ese evento para enviarlo a los usuarios. Aquí es donde interviene el «Motor de suscripciones» que se encarga de comparar el punto o geometría del evento con las suscripciones de todos los usuarios.

Para lograr el «matching» primero se realiza un buffering [21] de la geometría enviada por el proveedor y el valor de bufferMatching enviado en el evento en caso de que no se envíe dicho valor, se toma el valor seteado en el topic. Luego se compara si dicha geometría se intersecta con la geometría de cada suscripción existente en el

sistema para dicho topic y el resultado de este «matching» genera el listado de los clientes a los que se le debe enviar la alerta.

La función que se encarga de comparar dos geometrías, es el método «Intersect» de la librería JTS [22]. De esta manera se compara la geometría recibida en el evento con la zona de interés marcada en la suscripción del usuario o la última ubicación recibida para los casos de suscripciones por ubicación en tiempo real.

4.1.1.3. Adaptador de geometrías

Las geometrías proporcionadas por los proveedores y por los clientes es una tema relevante al momento de manejar los datos. Para esto se generó un modulo encargado de la gestión de estos datos geométricos para facilitar las operaciones que se realicen sobre ellos.

El manejo geográfico en la lógica de los módulos GeoMOM Central y el Broker de mensajería se implementó utilizando la biblioteca [23] que implementa la librería JTS. Esta librería cuenta, entre otros, con estructuras de datos para representar puntos y geometrías en el espacio, que son las clases «Point» y «Geometry».

Para la persistencia de datos y todos los servicios con componentes geográficos asociados se definieron las clases «Punto» y «Geometria». La clase Punto contiene dos atributos, latitud y longitud, mientras que la clase «Geometria» contiene una lista de valores de la clase «Punto».

De esta forma, el adaptador de geometrías toma un tipo de datos y lo convierte al correspondiente según el Cuadro 4.1:

Entrada	Salida
Point	Punto
Punto	Point
Geometria	Geometry
Geometry	Geometria

Cuadro 4.1: Adaptador de geometría

El uso de los tipos de datos «Punto» y «Geometria» nos permite publicar servicios con parámetros estándar, facilitando el uso al momento de consumirlos. Además es

posible almacenar fácilmente estos datos en cualquier motor de base de datos.

4.1.1.4. Servicios SOAP del GeoMOM Central para Proveedores

Todas las funcionalidades para los proveedores son provistas mediante servicios web SOAP y es el GeoMOM Central el encargado de publicar estos servicios. En esta sección se detalla cada servicio con su cometido en dicho componente.

Estos servicios web fueron implementados con el protocolo SOAP dado que sobre estos se pueden aplicar mecanismos de seguridad adicionales como HTTPS y WS-Security.

El servicio Publicar Evento es uno de los principales servicios que brinda el GeoMOM Central, ya que es el encargado de recibir los eventos publicados por los Proveedores. Los proveedores también tienen a disposición el servicio de Listar Topics. Estos servicios están publicados en la ruta: `GEOMOM/ws-geomom?wsdl`

En el Cuadro 4.2 se presenta el detalle de estos servicios junto a los parámetros requeridos para utilizarlos y su respuesta.

Publicar Evento	
Parámetros	Respuesta
<p>idProveedor: Identificador del Proveedor.</p> <p>idTopic: Identificador del Topic.</p> <p>fecha: Fecha en la que ocurrió el evento.</p> <p>fechaTimeout: Fecha de vencimiento de la alerta.</p> <p>prioridad: Indica la prioridad que tendrá el evento.</p> <p>punto: Coordenadas Latitud y Longitud del punto.</p> <p>geometria: Lista de coordenadas Latitud y Longitud que representan la geometría del evento.</p> <p>bufferMatching: Indica el buffer que se le aplicará al punto o geometría ingresada. El atributo debe ser ingresado en metros.</p> <p>eventoDesc: Se describe de qué se trata el evento ingresado.</p>	<p>Resultado del evento enviado. Mensaje adecuado en caso de error.</p>
Listar Topics	
No tiene	Lista de los Topics existentes en el sistema.

Cuadro 4.2: Descripción de los servicios SOAP del GeoMOM Central para los Proveedores

El valor de geometría recibido por la plataforma es un punto o una geometría. Luego de validado el evento y para su posterior tratamiento dentro de la plataforma, se procesa en el adaptador de geometrías para convertirlo en Point o Geometry según corresponda.

Dicho evento se envía al Topic correspondiente, para luego ser procesado por el matching geográfico, realizando las comparaciones geográficas mencionadas en 4.1.1.2. Inmediatamente identificados los clientes suscritos y con matching geográfica positivo, se procede a enviar la notificación mediante el Broker de Mensajería, según el canal de suscripción, y se agrega el mensaje a la cola de mensajes del cliente.

Con el fin de entender cómo es la comunicación entre los distintos componentes que participan cuando se procesa un evento, se presenta un diagrama DSS¹ en la Figura 4.3:

En este diagrama se puede ver que el evento es recibido por el GeoMOM Central y teniendo en cuenta las suscripciones de los clientes se genera una lista de todos los clientes que se obtuvo un matching positivo. Este matching se da entre la geometría de la suscripción y la geometría del evento.

Luego se envía al Broker de mensajería todos los MensajeEvento que contiene la información necesaria para almacenar las alertas en las colas de mensajes de los clientes que corresponda.

A continuación, el Broker de mensajería toma esta lista de clientes y le envía la notificación por los canales que seleccionó al momento de la suscripción.

4.1.1.5. Servicios REST del GeoMOM Central para Administración

Con el fin de diferenciar la lógica central del GeoMOM y las funcionalidades administrativas, la web de administración se implementó en un sistema independiente y comunicado con servicios web. Es por esto que se definen una serie de servicios web Rest para satisfacer todas las funcionalidades requeridas en la web.

¹DSS - Diagrama de Secuencia del Sistema

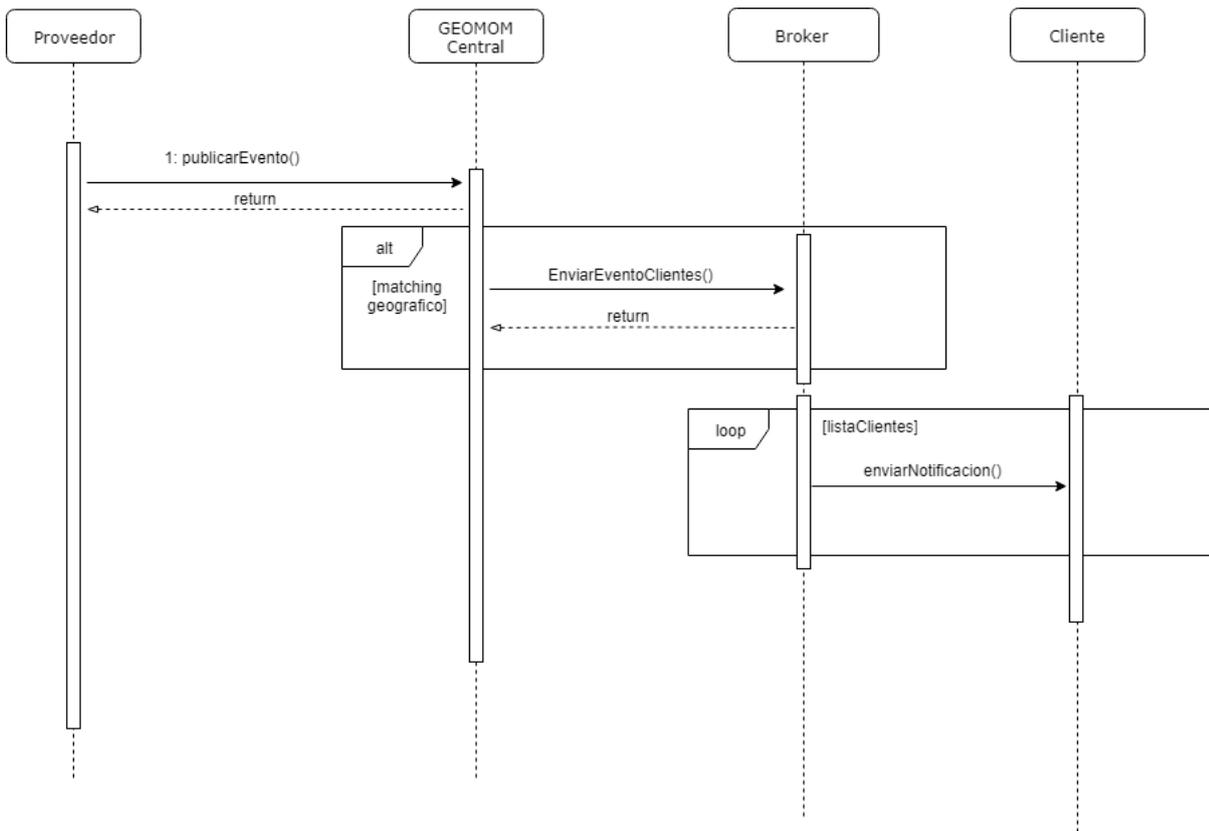


Figura 4.3: DSS - Publicar Evento

La elección de servicios Rest se debe a la simplicidad de su implementación y a los tipos de datos que se transfiere entre el GeoMOM Central y la Administración. Estos servicios están publicados en la ruta: `GEOMOM/rest-geomom/web/servicios/`

Se pueden agrupar en tres grandes grupos, por un lado los referidos a la administración de Topics, Proveedores y Clientes; luego están los necesarios para los listados y gráficos estadísticos; y por último los necesarios para la extracción de datos que también reutiliza servicios de los grupos anteriores. En el Cuadro 4.3 se detallan todos los servicios disponibles con los atributos y datos necesarios para su utilización.

Método	Ruta	Atributo	Cuerpo	Descripción
GET	topics			Lista los topics existentes en el sistema.
GET	topics	idTopic		Obtiene el Topic con idTopic enviado como parámetro en la ruta.
POST	topics		Topic	Crea el Topic enviado en el cuerpo de la consulta.
PUT	topics		Topic	Modifica el Topic enviado en el cuerpo de la consulta.

GET	proveedores			Lista los proveedores existentes en el sistema.
GET	proveedores	idProveedor		Obtiene el Proveedor con idProveedor enviado como parámetro en la ruta.
POST	proveedores		Proveedor	Crea el Proveedor enviado en el cuerpo de la consulta.
PUT	proveedores		Proveedor	Modifica el Proveedor enviado en el cuerpo de la consulta.
GET	clientes			Lista los clientes existentes en el sistema.
GET	clientes	idCliente		Obtiene el Cliente con idCliente enviado como parámetro en la ruta.
DELETE	clientes	idCliente		Elimina el Cliente con idCliente enviado como parámetro en la ruta. Además descarta sus mensajes, suscripciones, ubicaciones y la cola de mensajes en el Broker de mensajería
GET	mensajesclientes			Lista los mensajes enviados a los clientes. Se utiliza para los listados estadísticos.
GET	eventosproveedores			Lista los eventos generados por los proveedores. Se utiliza para los listados estadísticos.
GET	matchinggeografico			Retorna los eventos generados por los proveedores, las suscripciones de los clientes y los mensajes enviados a los clientes. Se utiliza para los listados estadísticos.
GET	clientesactivos			Lista todos los clientes en el sistema junto a la fecha del último mensaje enviado, último mensaje leído y si está o no suscrito a algún topic. Se utiliza para los listados estadísticos.
GET	canales			Lista los canales existentes en el sistema. Se utiliza para la extracción de datos.
GET	eventos			Lista los eventos existentes en el sistema. Se utiliza para la extracción de datos.

GET	mensajeeventos			Lista los MensajeEvento existentes en el sistema. Se utiliza para la extracción de datos.
GET	ubicaciones			Lista las Ubicaciones existentes en el sistema. Se utiliza para la extracción de datos.

Cuadro 4.3: Descripción de los servicios Rest del GeoMOM Central para la Administración

4.1.2. Broker de mensajería

El broker de mensajería es parte central de la arquitectura de servicios orientada a mensajes. Este componente centra su principal función en la comunicación con los clientes, tanto para el envío como para la recepción de información.

El diseño de este componente se basa en el uso de colas de mensajes para la gestión de la información de cada cliente. Esto permite tener una cola de mensajes con las alertas pendientes de envío, así como también permite consultar la cantidad de mensajes encolados, desencolados y demás estadísticas. La implementación se hizo utilizando Apache ActiveMQ [24], que se integra de manera sencilla con nuestra solución mediante las librerías correspondientes en Java.

Cuando se da un evento que debe ser enviado a un cliente, se genera una cola de mensajes identificada por el atributo «email» del Cliente y a partir de ese momento todos los mensajes dirigidos al cliente se van a almacenar en esta cola. Cuando el cliente consulta por sus alertas, el servicio correspondiente se encarga de obtener los mensajes de esta cola y entregarlos al cliente.

4.1.2.1. Envío de notificaciones

Cuando existe una alerta que debe ser enviada a un cliente, primero se le envía una notificación según los canales que haya seleccionado en suscripción correspondiente y luego se guarda esta alerta en su cola de mensajes en el Broker de mensajería.

El envío de notificaciones al cliente está centralizado en el Broker de mensajería y se implementó mediante dos canales: envío de notificaciones Push y envío de emails.

En la Figura 4.4 se muestra el flujo de envío de notificaciones desde el Broker de mensajería hacia el cliente para las suscripciones cuyo canal es email o notificaciones push. Una vez que se da un matching positivo para el evento generado por el Proveedor, el Broker de mensajería se encarga de enviarle la notificación al cliente, en el paso 1, según el canal que haya elegido en la suscripción correspondiente. En el paso 2, el cliente recibe la notificación y si es una alerta de su interés, puede consultarlas en el paso 3. Cuando el cliente consulta sus alertas disponibles, en el paso 4, el Broker de mensajería se encarga de leer los mensajes de la cola de mensaje del cliente y revisa que la alerta no haya vencido por timeout. Una vez que obtiene las alertas nuevas y las ya leídas por el cliente, se las retorna en dos listas separadas.

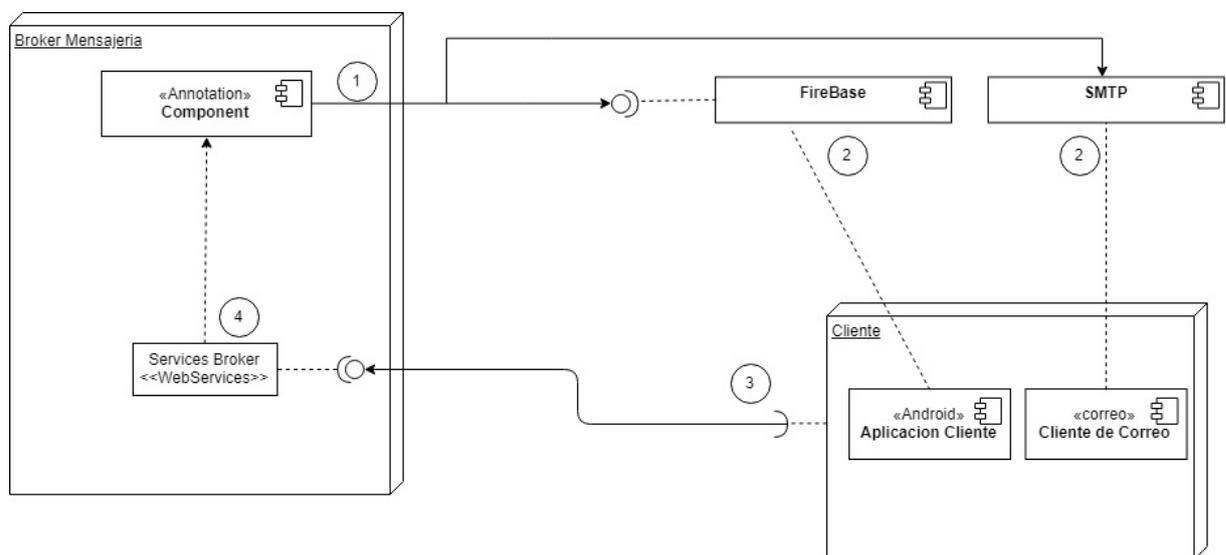


Figura 4.4: Envío de notificaciones y eventos al cliente

4.1.2.2. Dar localización

Contar con la ubicación del cliente actualizada es tarea fundamental para que los eventos sean de utilidad para los clientes. Es tarea del Broker de mensajería procesar todas las solicitudes de actualización de localizaciones de todos los dispositivos móviles o cualquier plataforma que permita obtener su geolocalización.

Al momento de recibir una localización, el Broker procesa dicha solicitud guardando la ubicación enviada y la fecha en que fue recibida. Además actualiza el dato de la última ubicación conocida en el cliente que se utilizará para matching en caso necesario.

4.1.2.3. Consultar alertas

El cliente consulta cuáles son las alertas que tiene disponibles en la plataforma. Al momento de ejecutar este servicio, el Broker consulta los mensajes disponibles en la cola de mensaje del cliente con el fin de verificar si tiene alertas sin leer y que no hayan expirado por timeout. Al cliente se le retorna la lista de las nuevas alertas y también las alertas que ya tenía almacenadas en la plataforma.

4.1.2.4. Servicios REST del Broker de mensajería

El Broker de mensajería publica ciertos servicios para cumplir con las funcionalidades que realiza dentro de la plataforma. Al tratarse de clientes que utilizan dispositivos móviles, los recursos con los que cuentan son limitados y es por eso que los servicios se implementan con tecnología Rest ya que son servicios que consumen pocos recursos. Estos se encuentran publicados en la ruta: `GEOMOM/rest-geomom/broker/servicios`

En el Cuadro 4.4 se encuentra un listado de todos los servicios prestados, los parámetros necesarios para consumirlos y una pequeña descripción de su funcionamiento.

Método	Ruta	Atributo	Cuerpo	Descripción
GET	topics			Lista los topics existentes en el sistema.
GET	topics	idTopic		Obtiene el Topic con idTopic enviado como parámetro en la ruta.
POST	topics/suscripto	idTopic	Cliente	Retorna si el cliente enviado en el cuerpo de la consulta está suscripto al Topic enviado como parámetro en la ruta. En caso de estarlo retorna la suscripción correspondiente.
POST	topics/suscribir		Suscripcion	Se realiza la suscripción según los datos enviados en el cuerpo de la consulta.
POST	topics/cancelarsuscripcion	idTopic	Cliente	Cancela la suscripción del cliente al topic con idTopic enviado como parámetro en la consulta.

POST	eventoscliente		Cliente	Envía todas las alertas anteriores almacenadas para el cliente enviado en el cuerpo de la consulta.
POST	eventosnuevoscliente		Cliente	Consulta si existe alertas sin leer en la cola de mensajes del cliente enviado en el cuerpo de la consulta y las retorna.
POST	ubicacioncliente		Cliente	Recibe la actualización de la ubicación actual del cliente enviado en el cuerpo de la consulta y la almacena en la tabla de ubicaciones, junto a la fecha de recibida.
POST	logincliente		Cliente	Si el cliente recibido en el cuerpo de la consulta no existe, lo crea con los datos recibidos. En caso que se trate de un cliente existente, únicamente actualiza el valor de TokenApp.

Cuadro 4.4: Descripción de los servicios Rest del Broker de mensajería

4.2. Base de datos

Los esquemas de bases de datos utilizados en la solución son dos: GEOMOM y GEOMOMADMIN. El primero es el que se utiliza para la persistencia de todos los datos de la lógica de la plataforma, es decir el esquema que utilizan los componentes GeoMOM Central y Broker de mensajería. El segundo es utilizado por el componente de administración.

La persistencia se generó mediante JPA [25] con su implementación mediante Hibernate [26] para el mapeo de objetos a la base de datos.

Los detalles que siguen a continuación hacen referencia a la base de datos del GeoMOM Central y del Broker de mensajería, dado que la administración web contie-

ne solamente la información de los usuarios administradores.

En la Figura 4.5 se describe a través de un modelo relacional generado con Db-Visualizer [27] la estructura de tablas y restricciones existentes en la base de datos GEOMOM.

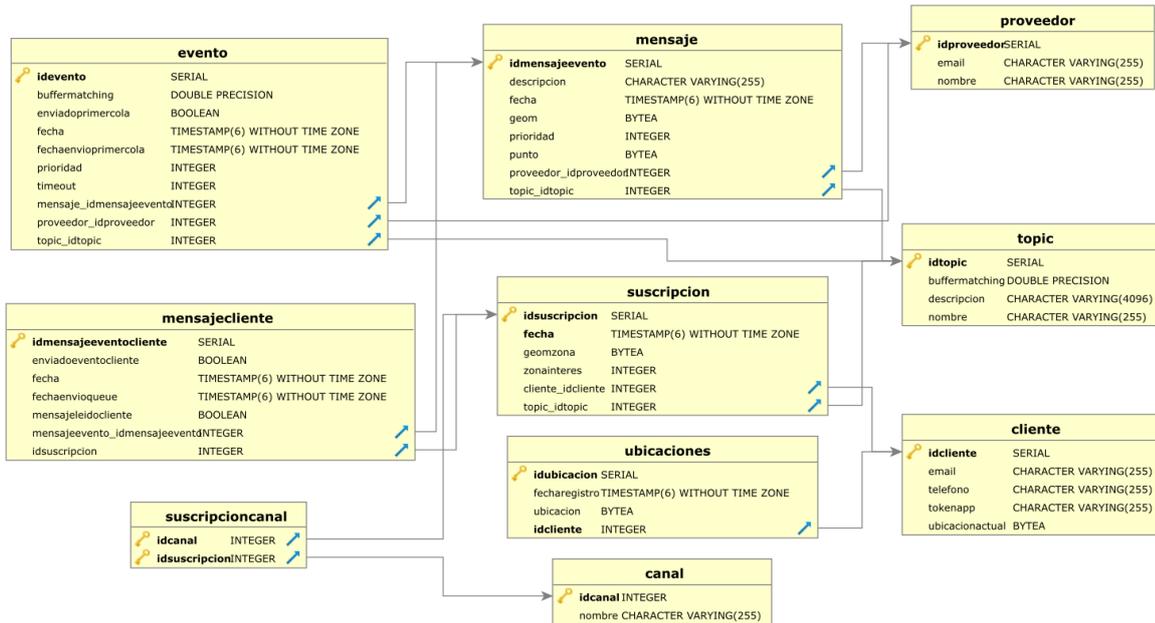


Figura 4.5: Base de datos

Los eventos generados por los proveedores son almacenados en la tabla «eventos» y, a su vez, cada evento enviado a los topics es persistido en la tabla «mensaje».

Las suscripciones de los clientes son registradas en la tabla «suscripcion» que contiene datos relevantes como la zona geométrica de interés de la suscripción, así como el topic al que pertenece.

Se utiliza la tabla «ubicaciones» para almacenar el histórico de todas las ubicaciones que envía cada cliente. Además, cada cliente tiene almacenado su última ubicación en la tabla «cliente» junto sus restantes atributos.

La tabla «mensajecliente» contiene los mensajes que deberán ser enviados a cada cliente, la relación con la suscripción, la relación al evento generado, valores de fecha necesarios y banderas de control.

4.3. Web de Administración

Para realizar una correcta gestión de los Topics y Proveedores de la plataforma, se implementó una web de administración que se conecta al GeoMOM a través de

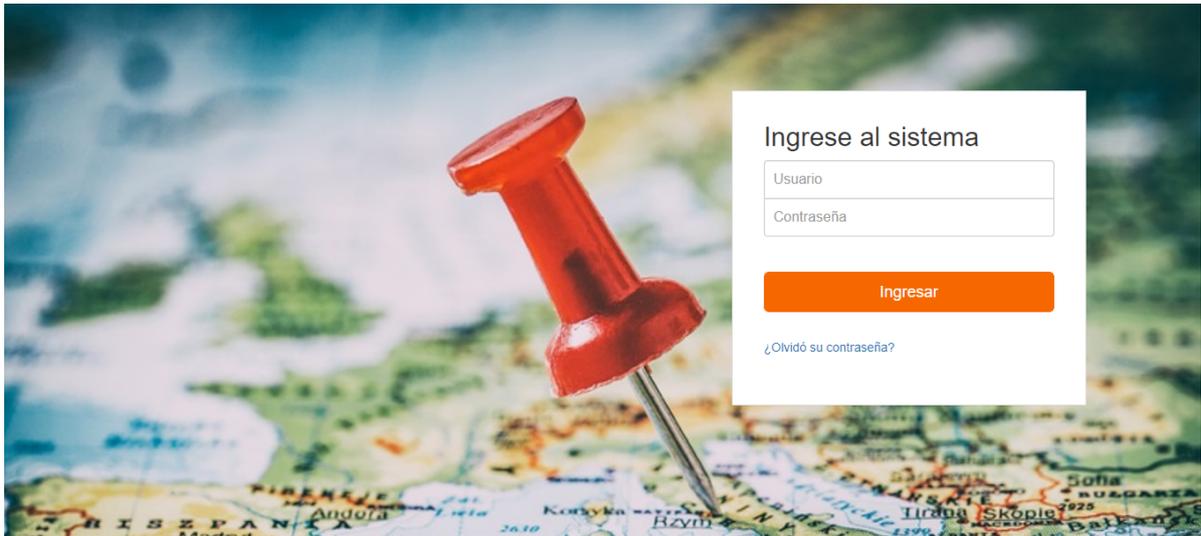


Figura 4.6: Página de login de la Web de Administración

servicios web Rest. Esta web tiene las funcionalidades necesarias para dar de alta y realizar modificaciones sobre los Topics y los Proveedores. En cuanto a los Clientes, se pueden realizar listados con distintos tipos de filtros, ver información de ellos y eliminarlos en los casos que corresponda.

Además se muestran estadísticas y listados de cantidad de eventos generados, alertas enviadas, porcentaje de matching geográfico entre otros. Utilizando estos listados se pueden tomar decisiones y mejorar el funcionamiento de la plataforma, como puede ser eliminar clientes inactivos o asignar recursos a determinados componentes con alto estrés.

Para el desarrollo de la implementación web se utilizó Bootstrap [28] para el diseño. Esta librería permite realizar sitios responsivos, facilita la maquetación y la generación de las distintas páginas. Además se utiliza Spring Framework [29] que brinda funcionalidades importantes en la generación de formularios con validaciones, presentación de la información y facilita el vínculo entre las vistas y los datos de la lógica.

Por otro lado, con el fin de ejecutar completamente un caso de estudio, se implementó una interfaz dentro de la web de administración para simular los casos de uso del proveedor para el envío de alertas georreferenciadas.

4.3.1. Gestión de Topics y Proveedores

Con el objetivo de cumplir con estas funcionalidades se brindan listados de todos los Topics y Proveedores existentes en la plataforma. Además, es posible dar de

alta nuevas instancias o modificar existentes, todo esto a través de formularios con validaciones en los campos.

Para facilitar el manejo de los datos en la web y la comunicación con los servicios web del GeoMOM se generó la misma estructura de clases existente en el GeoMOM con todos los atributos necesarios para el funcionamiento de la web. De esta forma se pueden mapear fácilmente los datos en las respuestas de los web services a las clases correspondientes. Utilizar las mismas clases que el GeoMOM facilita el proceso de mostrar y recolectar datos desde los formularios web utilizando Spring Framework.

En la Figura 4.7 se muestra un ejemplo de los listados de Topics.

Es posible ordenar los datos por cualquiera de las columnas y además realizar búsquedas de palabras donde automáticamente los datos se filtran según ese criterio. En la segunda figura se muestra un ejemplo del formulario para visualizar o ingresar datos de los Topics.

Este listado y forma de visualizar los datos es igual para el caso de los datos de los Proveedores.

4.3.2. Estadísticas para la gestión

Una ventaja de contar con una sitio de administración para una plataforma de la naturaleza del GeoMOM es disponer con insumos para la toma de decisiones y mantener en buenas condiciones los servicios prestados. Por ejemplo, es importante para los Proveedores saber si los eventos generados por ellos están siendo enviados a sus clientes y si no lo están, saber por qué razón. En el caso de los Administradores, es de utilidad saber si existen Clientes que no están recibiendo las alertas que le son enviadas. Esto puede suceder por varias razones, como por ejemplo, que el Cliente ya no esté utilizando el dispositivo o que ya no cuente con la aplicación para recibirlos. En los casos en los que el Cliente ya no esté activo, es de utilidad poder descartarlos de la plataforma para liberar los recursos reservados de ese cliente.

Con el fin de generar estadísticas que ayuden a la gestión de la plataforma, se definen las siguientes funcionalidades.

4.3.2.1. Listar todos los mensajes

Se lista la cantidad de mensajes generados en la plataforma, donde se especifican los mensajes enviados por alertas y los mensajes efectivamente leídos por los clientes.

GEOMOM-ADMIN - Web de administración del GEOMOM admin

Menu Tareas

- Topics
- Listar topics
- Cientes
- Proveedores
- Extraer Datos

Listado de Topics

Ver 10 resultados por página Buscar:

Nombre	Descripción	Buffer Matching
Alertas meteorológicas	Pueden ser brindadas por el INUMET y SINAЕ. La mision de INUMET es prestar servicios públicos meteorológicos y climatológicos, con el objeto de contribuir a la seguridad de las personas y sus bienes, actuando como autoridad meteorologica en el territorio nacional.	0.01
Incendios	Los incendios de diversa magnitud (forestales, vivienda, vehículos, etc.) pueden ser reportados por el SINAЕ que es el organismo cuya responsabilidad es la gestión integral de riesgo de desastres en	0.01

GEOMOM-ADMIN - Web de administración del GEOMOM admin

Menu Tareas

- Topics
- Cientes
- Proveedores
- Extraer Datos

Ver Topic

Datos

Nombre
Incendios

Descripción
Los incendios de diversa magnitud (forestales, vivienda, vehículos, etc.) pueden ser reportados por el SINAЕ que es el organismo cuya responsabilidad es la gestión integral de riesgo de desastres en el Uruguay.

Buffer matching
0.01

[Editar](#)
[Listar topics](#)
[Eliminar](#)

Figura 4.7: Listado de Topics y Ver Topic

Además se grafica el porcentaje de mensajes enviados por cada Topic en relación al total de mensajes como se puede observar en la Figura 4.8.

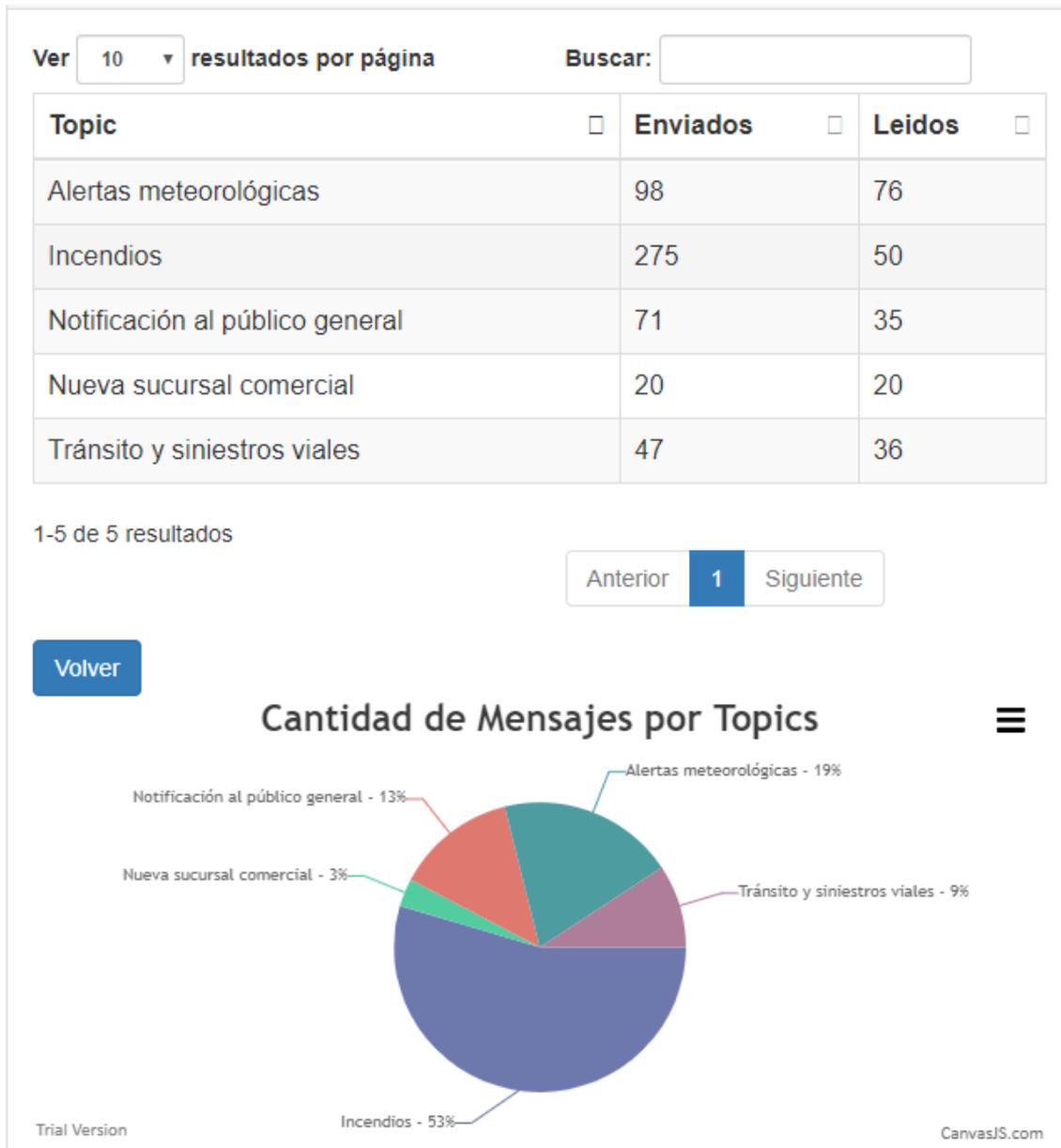


Figura 4.8: Mensajes por topics

4.3.2.2. Clientes por Topic

Se indica la cantidad de clientes que están suscritos a los distintos Topics del sistema y un gráfico con la distribución de suscripciones por Topic como se puede observar en la Figura 4.9.



Figura 4.9: Clientes por topics

4.3.2.3. Eventos por proveedor

Se listan todos los proveedores existentes en el sistema junto a la cantidad de eventos que generó cada uno como se puede observar en la Figura 4.10.

Ver 10 resultados por página

Buscar:

Proveedor	Eventos generados
Dirección Nacional de Medio Ambiente (DINAMA)	250
Instituto Uruguayo de Meteorología (Inumet)	661
Intendencia de Montevideo (IM)	183
Sistema Nacional de Emergencias (SINAE)	100

1-4 de 4 resultados

Anterior 1 Siguiente

Volver

Cantidad de Eventos por Proveedor

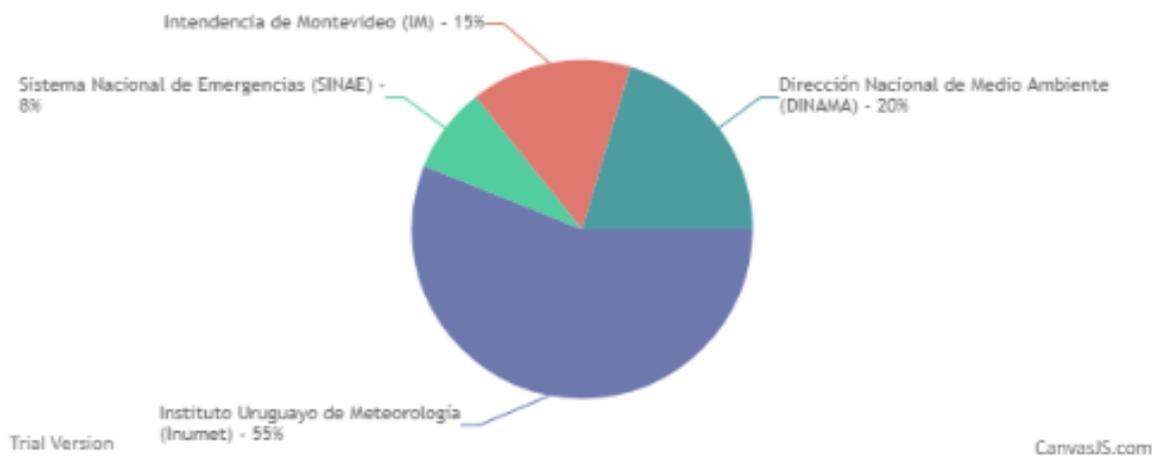


Figura 4.10: Eventos por proveedor

4.3.2.4. Matching geográfico

El panel de matching geográfico indica el total de eventos generados, la cantidad de suscripciones y la cantidad de mensajes efectivamente enviados.

El cálculo del porcentaje de matching geográfico utilizado para las gráficas se realiza con la Fórmula 3.1 presentada en la sección 3.2.1 referente a los requerimientos funcionales.

En la Figura 4.11 se encuentra el listado de los distintos Topics del sistema junto a la cantidad de eventos generados por los proveedores, suscripciones de los clientes y

la cantidad de mensajes que fueron enviados.

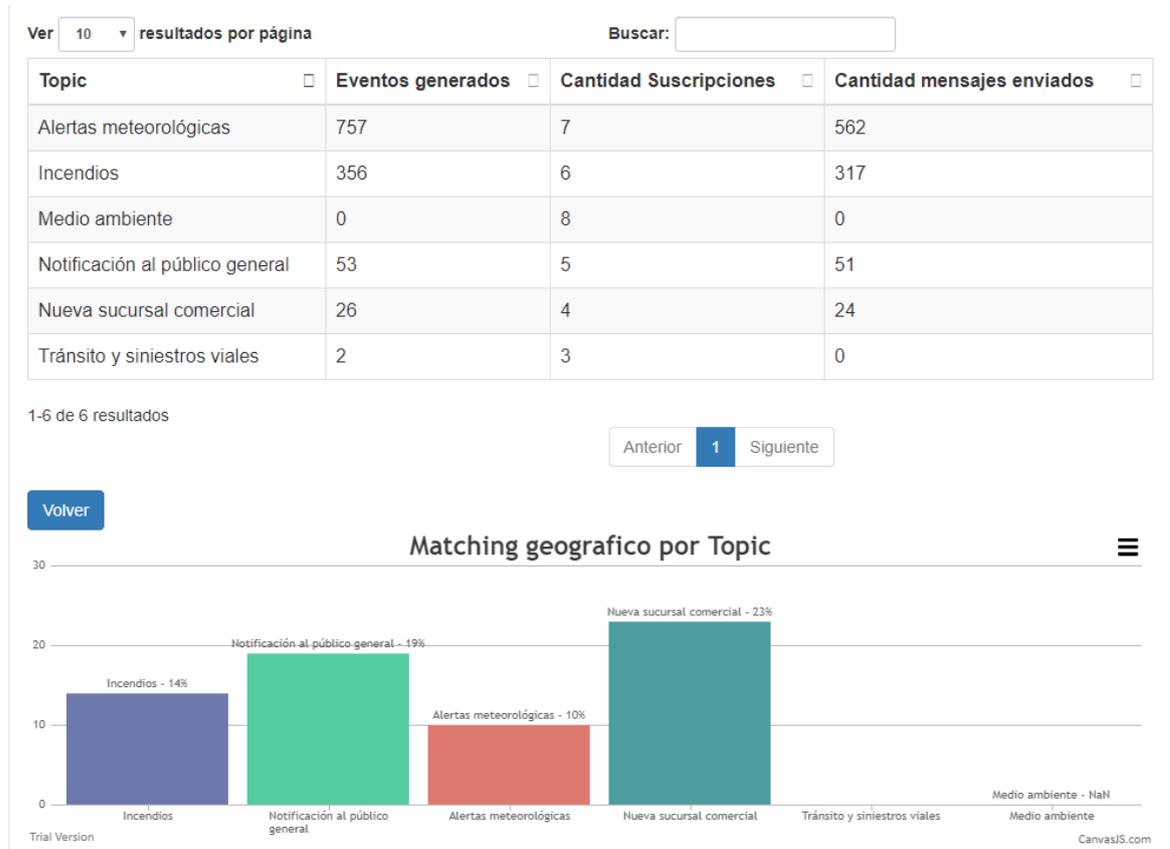


Figura 4.11: Matching geográfico por topic

4.3.2.5. Clientes activos e inactivos

En este listado el Administrador tiene a disposición todos los clientes existentes en el sistema como se puede observar en la Figura 4.12. Para cada cliente se indica si está o no suscrito a algún Topic, la fecha del último mensaje leído y la fecha del último mensaje enviado.

Ver resultados por página Buscar:

Cliente <input type="checkbox"/>	Mail <input type="checkbox"/>	Suscripto <input type="checkbox"/>	Fecha ultimo mensaje Leido <input type="checkbox"/>	Fecha ultimo mensaje Enviado <input type="checkbox"/>
1	carlozve@gmail.com	SI	Sat Jul 27 19:31:00 GFT 2019	Sat Jul 27 19:33:00 GFT 2019
2	test1@geomom.com	NO		
3	gaston231@gmail.com	SI	Sat Jul 27 19:04:00 GFT 2019	Sat Jul 27 19:32:00 GFT 2019
4	ggonzalez@geomom.com	SI		
5	nmelgar@geomom.com	SI		Sat Jul 27 19:32:00 GFT 2019
6	rperazza@geomom.com	SI		Sat Jul 27 19:32:00 GFT 2019
7	rperez@geomom.com	SI		Sat Jul 27 19:32:00 GFT 2019
8	mperdomo@geomom.com	SI		
9	phein@geomom.com	SI	Sat Jul 27 19:45:00 GFT 2019	Sat Jul 27 19:33:00 GFT 2019
10	vcravino@geomom.com	SI		
11	mclavero@geomom.com	SI		
12	rcaraballo@geomom.com	SI		
13	fdelgado@geomom.com	SI		
14	jaserta@geomom.com	SI		

1-14 de 14 resultados

Figura 4.12: Clientes activos

Con esta información es posible determinar, según algún criterio que considere pertinente, cuales son los clientes que se encuentran inactivos. El listado le permite filtrar y ordenar los datos según lo que considere necesario.

Si algún cliente debe ser eliminado, ingresa al que corresponda y lo elimina. Esto implica eliminar los mensajes del cliente previamente almacenados, las suscripciones, la cola de mensajes en el Broker de mensajería y toda la información referente al cliente.

Capítulo 5

Caso de estudio y validaciones

En este capítulo se presenta el caso de estudio llevado a cabo para las pruebas de la plataforma y la validación de la propuesta implementada. Se describe la aplicación móvil desarrollada con el fin de simular las interacciones de los clientes con la plataforma a través de su dispositivo móvil. Por último se incluye una sección con las pruebas de verificación y validaciones realizadas.

5.1. Caso de Estudio

Con respecto al caso de estudio, en primer lugar se determinan los datos a utilizar, ejemplos de Proveedores, Topics, Canales y diferentes datos de administración necesarios para el correcto funcionamiento de la plataforma. Luego se detallan todas las funcionalidades implementadas.

5.1.1. Datos utilizados

Se definen los datos utilizados en el caso de estudio con el fin de validar la implementación. Como base de pruebas se define un conjunto de datos que son necesarios al momento de verificar los principales requerimientos funcionales, entre los que se encuentran los datos de Proveedores, Topics y Canales.

5.1.1.1. Datos de Proveedores

Los proveedores de información que se definen para la generación de eventos son los siguientes:

- Instituto Uruguayo de Meteorología (INUMET)

- Sistema Nacional de Emergencias (SINAE)
- Intendencia de Montevideo (IM)
- Dirección Nacional de Medio Ambiente (DINAMA)

5.1.1.2. Datos de Topics

Los Topics hacen referencia a las temáticas a las que se pueden suscribir los Clientes, por lo tanto, se definen los siguientes:

- Incendios
- Alertas meteorológicas
- Tránsito y siniestros viales
- Nueva sucursal comercial
- Notificación al público general
- Medio ambiente

5.1.1.3. Datos de Canales

Cuando un cliente se suscribe a un Topic, determina por qué canal desea recibir las notificaciones de alertas nuevas. Para esto se definen tres tipos de Canales.

- SMS¹
- EMAIL - Correo electrónico
- NOTIFICACIÓN - Notificaciones push para dispositivos móviles

Para el caso del SMS, se definió como una opción que puede elegir el Cliente pero no se implementó el envío de mensajes de texto.

¹SMS - Short Message Service

5.1.1.4. Datos de Prioridades

Cada evento generado por un Proveedor debe indicar cual es la prioridad para ese evento. Para esto se definen cuatro tipos de prioridades:

- DATO_INFORMATIVO
- ADVERTENCIA
- ALERTA
- ALERTA_URGENTE

5.2. Aplicación cliente

La aplicación que van a utilizar los clientes, para hacer uso de la plataforma, deberá ser desarrollada por los distintos proveedores de información. El objetivo es que cada proveedor que ya disponga de una aplicación móvil pueda actualizarla para adaptarse a las funcionalidades que brinda la plataforma y así utilizar sus servicios.

Cada desarrollador de aplicaciones de los proveedores dispone de los servicios web necesarios para acceder a las distintas funcionalidades y así hacer uso completo de la plataforma.

En este proyecto, el desarrollo de la aplicación no formaba parte del alcance de la plataforma pero sí fue necesario realizarlo para generar las pruebas correspondientes.

Se implementó una aplicación móvil en lenguaje Android [30] utilizando Android Studio [31] como IDE² para la programación.

Esta aplicación cuenta únicamente con las funcionalidades mínimas necesarias para cumplir con los requisitos del caso de estudio planteado. La Figura 5.1 ilustra la pantalla de bienvenida a la aplicación y el menú principal.

5.2.1. Funcionalidades

Las funcionalidades de la aplicación móvil son únicamente las necesarias para testear los servicios de la plataforma GeoMOM. Dicho esto, se implementaron algunos de los servicios publicados por la plataforma para los clientes, con el fin de lograr un ciclo completo de pruebas. Los servicios implementados son:

²IDE - Integrated Development Environment

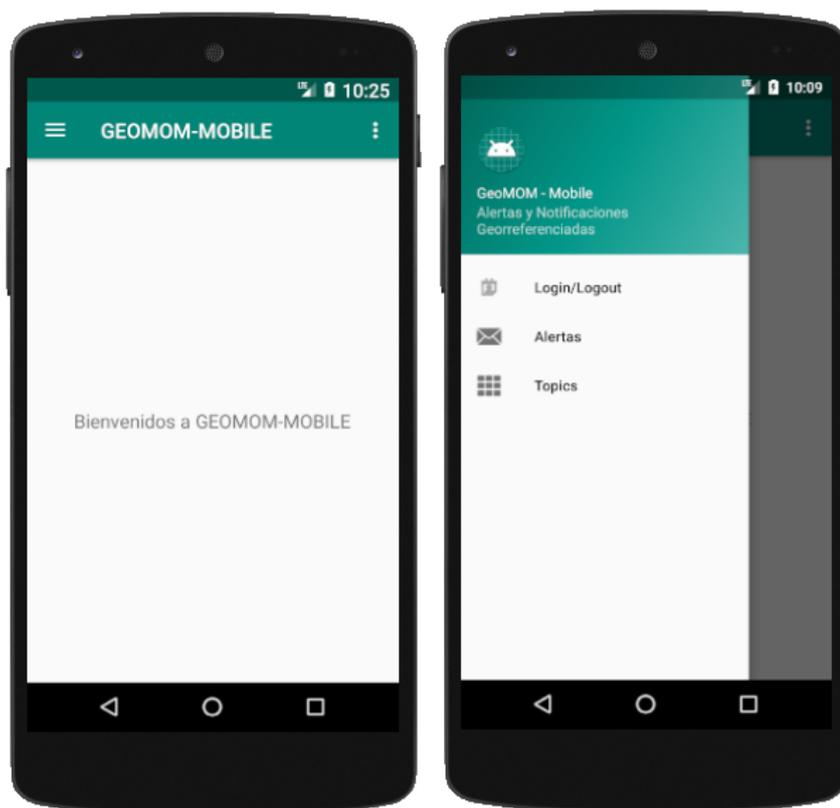


Figura 5.1: Aplicación cliente

- Login / Logout
- Listar Topics
- Ver Topic
- Suscribirse a Topic
- Enviar ubicación
- Cancelar suscripción a Topic
- Listar alertas
- Ver alerta

Para consumir los servicios REST publicados por el GeoMOM, que fueron detallados en la sección 4.3, se utilizó la librería Retrofit [32] que se trata de un cliente HTTP para Android y Java. La gran ventaja que tiene el uso de esta librería es que la respuesta de los servicios se pueden transformar en clases Java y así facilitar el manejo de los datos obtenidos.

Para el manejo de datos se definen clases simétricas a las clases del GeoMOM para que las respuestas de los servicios puedan ser leídas fácilmente. Algo similar a lo realizado para la Administración.

A continuación se indica la llamada a cada servicio utilizando Retrofit y se describe brevemente como interactúa la aplicación con cada uno de los servicios:

5.2.1.1. Listar Topics y ver Topic

```
@GET("topics")
Call<MensajeServicio> getTopics();

@GET("topics/{idTopic}")
Call<MensajeServicio> getTopicById(@Path("idTopic") int idTopic);
```

Se listan todos los topics existentes en la plataforma con el nombre y descripción de cada uno. Para acceder a esta funcionalidad se requiere que el usuario ingrese a la opción *Topics* en el menú principal.



Figura 5.2: Listar topics

En la Figura 5.2 se puede ver la lista de Topics en el sistema y también el detalle

de cierto Topic cuando es seleccionado.

5.2.1.2. Suscribirse a topic

```
@POST ("topics/suscribir")
```

```
Call<MensajeServicio> suscribirTopic(@Body Suscripcion suscripcion);
```

Luego de que el cliente elige un Topic y quiere suscribirse a él, debe seleccionar si lo desea hacer mediante *Ubicación en tiempo real (GPS)* o *Zona de interés en mapa*. Si la suscripción se realiza mediante ubicación en tiempo real, al cliente se le muestra en un mapa su ubicación actual y se le habilita el botón *Confirmar suscripción*. En caso de que sea una suscripción mediante una zona de interés personalizada, se le habilita la opción de definir la zona en una nueva pantalla. Ahí puede seleccionar todos los puntos que definen su zona de interés, deshacer el último punto agregado (en caso de que fuera necesario) y, cuando tiene la zona definida, confirmar su suscripción.

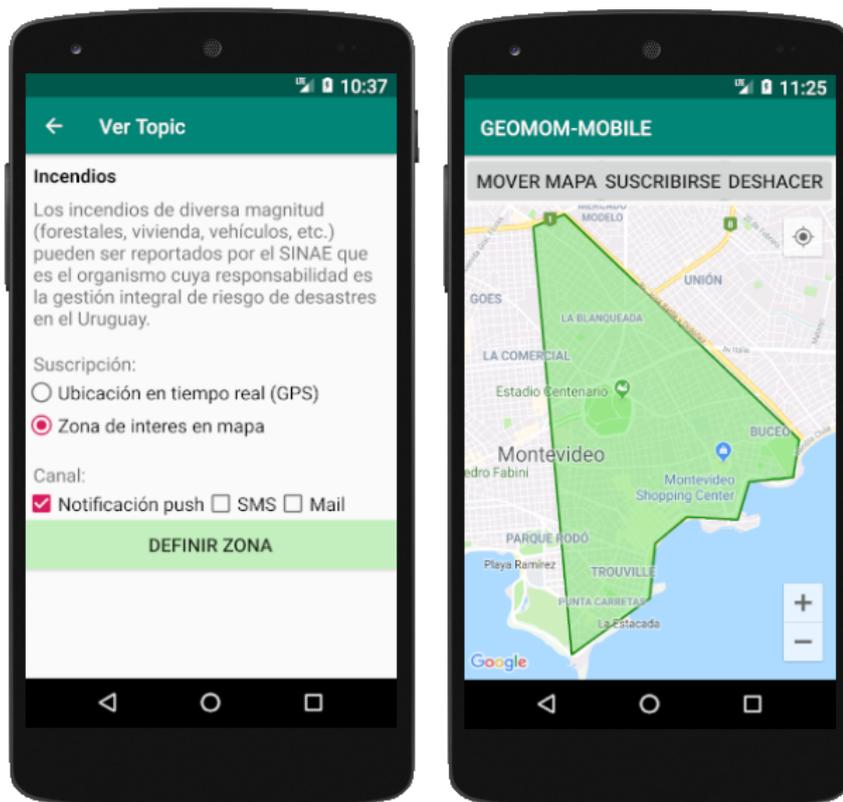


Figura 5.3: Suscribirse a Topic

En la Figura 5.3 se puede ver un ejemplo de una suscripción mediante zona de interés al Topic «Incendios».

5.2.1.3. Enviar ubicación

```
@POST("ubicacioncliente")
Call<MensajeServicio> enviarUbicacion(@Body Cliente cliente);
```

Cuando la suscripción del cliente es mediante ubicación en tiempo real, la aplicación debe enviar su ubicación al Broker de mensajería para que el GeoMOM disponga de esa información al momento de realizar el matching geográfico.

Una vez que el cliente está suscripto a algún Topic con esta opción, se dispara un proceso que envía la ubicación utilizando el servicio Rest del Broker de mensajería correspondiente.

El proceso se encarga de enviar los datos de Latitud y Longitud del GPS cada cierto tiempo. Para poder acceder a los datos del GPS del dispositivo fue necesario agregar los permisos de acceso en el archivo Manifest de la aplicación:

```
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
```

5.2.1.4. Cancelar suscripción topic

```
@POST("topics/cancelarsuscripcion/{idTopic}")
Call<MensajeServicio> cancelarSuscripcionClienteTopic(@Body Cliente
    cliente, @Path("idTopic") int idTopic);
```

Si el usuario ingresa a ver la información de un Topic al que ya está suscripto, la aplicación le muestra los detalles de esa suscripción y le permite cancelarla como se muestra en la Figura 5.4. Se implementó de forma que cada cliente pueda tener solamente una suscripción activa para cada Topic.

5.2.1.5. Listar alertas y Ver alerta

```
@POST("eventoscliente/")
Call<MensajeServicio> recibirEventosCliente(@Body Cliente cliente);
```



Figura 5.4: Cancelar suscripción

El cliente puede acceder a sus alertas recibidas en la opción *Alertas* del menú principal. En la primera imagen de la Figura 5.5 se muestra un listado de todas las alertas ya recibidas para ese cliente. En ese listado se indica el nombre del Topic, la prioridad de la alerta, su descripción y la fecha de enviada. Los distintos colores para cada alerta corresponden al nivel de prioridad de cada una.

En caso que se liste una alerta que no haya sido leída por el cliente, se la destaca con texto en negrita.

Posteriormente, el cliente puede seleccionar una alerta y se muestran los detalles como se puede observar en la segunda imagen de la Figura 5.5. El detalle incluye el Topic correspondiente, el Proveedor, algunos datos de la suscripción y datos ingresados por el Proveedor. En todos los casos se muestra un mapa con la ubicación de la alerta, marcada en rojo, y la zona de interés marcada en la suscripción para ese Topic en color verde. De esta manera el cliente puede saber de qué forma le afecta ese evento.

En el ejemplo ilustrado se puede apreciar que las zonas no se intersectan, esto es así porque es un caso de un evento que fue enviado con un valor de buffer. Por esta razón, la zona que afecta la alerta es mayor a la dibujada, provocando que se intersecte con la zona seleccionada en la suscripción del cliente.

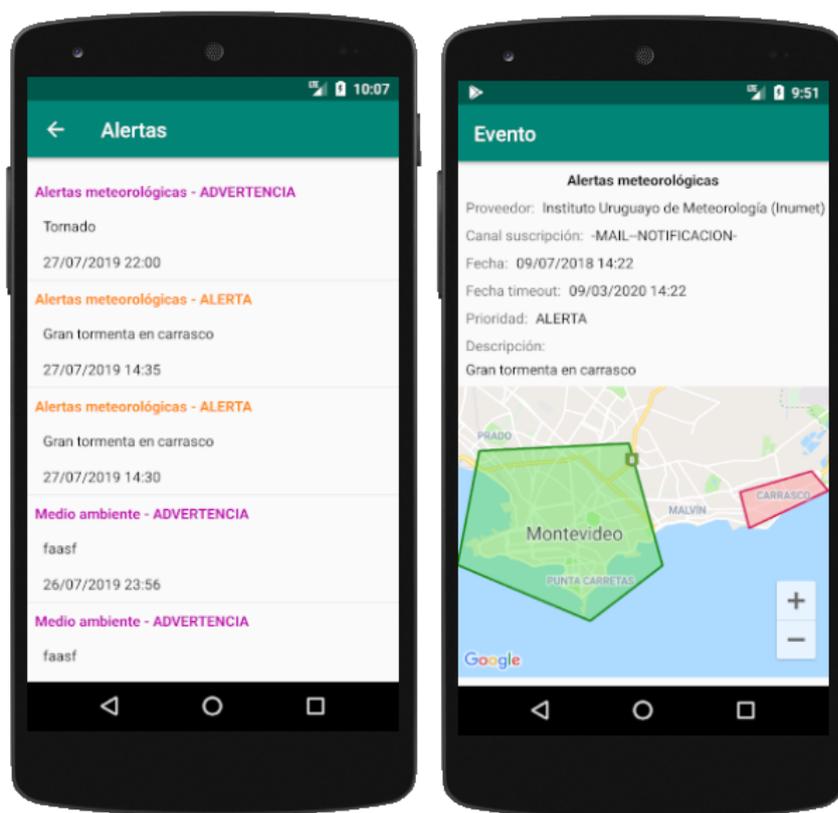


Figura 5.5: Alertas

5.2.2. Notificaciones push

Si a un cliente le corresponde que se le envíe una alerta, es de interés que esa alerta sea leída por el destinatario. Una de las formas más efectivas de notificar a los usuarios de dispositivos móviles es a través de notificaciones push. Para el envío de estas notificaciones se utiliza Firebase Cloud Messaging [33], que se trata de uno de los componentes que integra la plataforma Firebase. Esto permite enviar notificaciones push utilizando su API³ desde el Broker de mensajería. El uso de esta herramienta permite realizar un seguimiento de todas las notificaciones enviadas y tener estadísticas en general.

En la Figura 5.6 se puede ver un ejemplo de una notificación recibida con la información referente a una alerta de incendios.

5.2.3. API Google Maps

Para todos los casos en que se muestran mapas, se utilizan los provistos por la API de Google Maps [34] que permite fácilmente la integración de mapas con aplicaciones

³API - Application Programming Interface

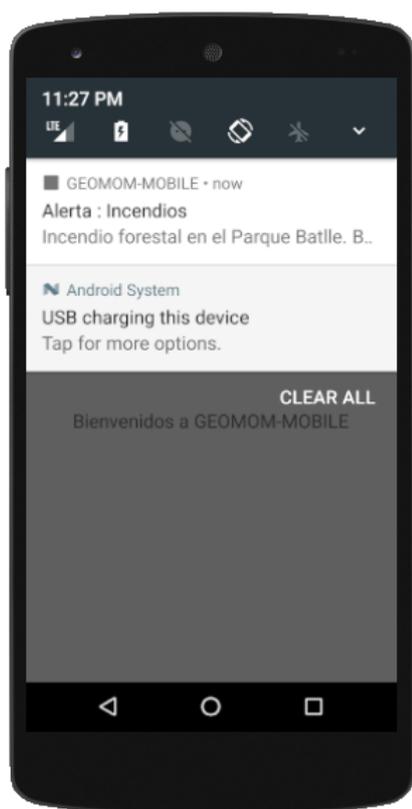


Figura 5.6: Notificación push

desarrolladas en Android.

Se utilizan funcionalidades para generar polígonos con el fin de delimitar las zonas de interés o zonas que afectan las alertas. Para delimitar las zonas de interés de los clientes al momento de suscribirse a un Topic, se capturan los puntos que el cliente va seleccionando en el mapa y luego se dibuja el polígono resultante.

5.3. Verificaciones y validaciones

Para garantizar el correcto funcionamiento de la plataforma en todos sus componentes se realizaron pruebas teniendo en consideración los casos de uso relevantes a la arquitectura.

En primer lugar se realizan pruebas unitarias validando los principales servicios que brinda la plataforma.

La validación de la administración web corresponderá con los casos de uso relevantes a la arquitectura, y el correcto funcionamiento de los datos estadísticos de envío y recepción de mensajes.

Finalmente, se requiere una prueba general de la plataforma, por lo tanto se eje-

cutarán pruebas que exijan al sistema en su totalidad y a todas las funcionalidades propuestas.

A continuación se explican, en cada una de las secciones, los puntos anteriormente mencionados según el caso de uso que se está validando.

5.3.1. Publicar evento

Se crearon casos de prueba mediante la herramienta SoapUI [35]. Se generaron tres modelos: el primero es la verificación unitaria, el segundo es la prueba de estrés y el tercero es una prueba de carga de datos significativa.

Las pruebas realizadas corresponden a:

- Cuatro casos de prueba con distintos tipos de geometrías en los eventos enviados. Dos con puntos y dos con polígonos.
- Pruebas de estrés, en la que se generaran ciclos de 200 ejecuciones enviando eventos cada cierto tiempo.
- Pruebas de carga, en la que se construye un polígono con 1530 puntos, ejecutando la prueba 20 veces con 1 segundo de demora entre cada ejecución.

Para el primer modelo, se visualiza la ejecución correcta en la Figura 5.7 en los cuatro casos generados.

Para el caso de la prueba de estrés, se realiza un total de 200 ejecuciones de envío de evento. Como se puede visualizar en la Figura 5.8 se ejecutó el test con un total de 200 eventos obteniendo un promedio de respuesta de 305,75 ms por evento y un total de ejecución de 1 minuto y 9 segundos.

Para este test se registraron dos clientes con suscripción al Topic «Incendio». En la Figura 5.9 se puede visualizar los mensajes generados en las colas de mensaje de cada cliente dentro del Broker de mensajería. Además es posible observar la cantidad de mensajes leídos y pendientes.

Las pruebas se completaron con éxito y todos los clientes recibieron las notificaciones y alertas correspondientes. Con este test, también se pudo probar los envíos de notificaciones push. Se notó una demora de aproximadamente dos segundos entre que un evento ingresa a la plataforma y se envía al cliente mediante la notificación. Se observa que los tiempos aumentan cuando la cantidad de suscripciones al Topic incrementa.

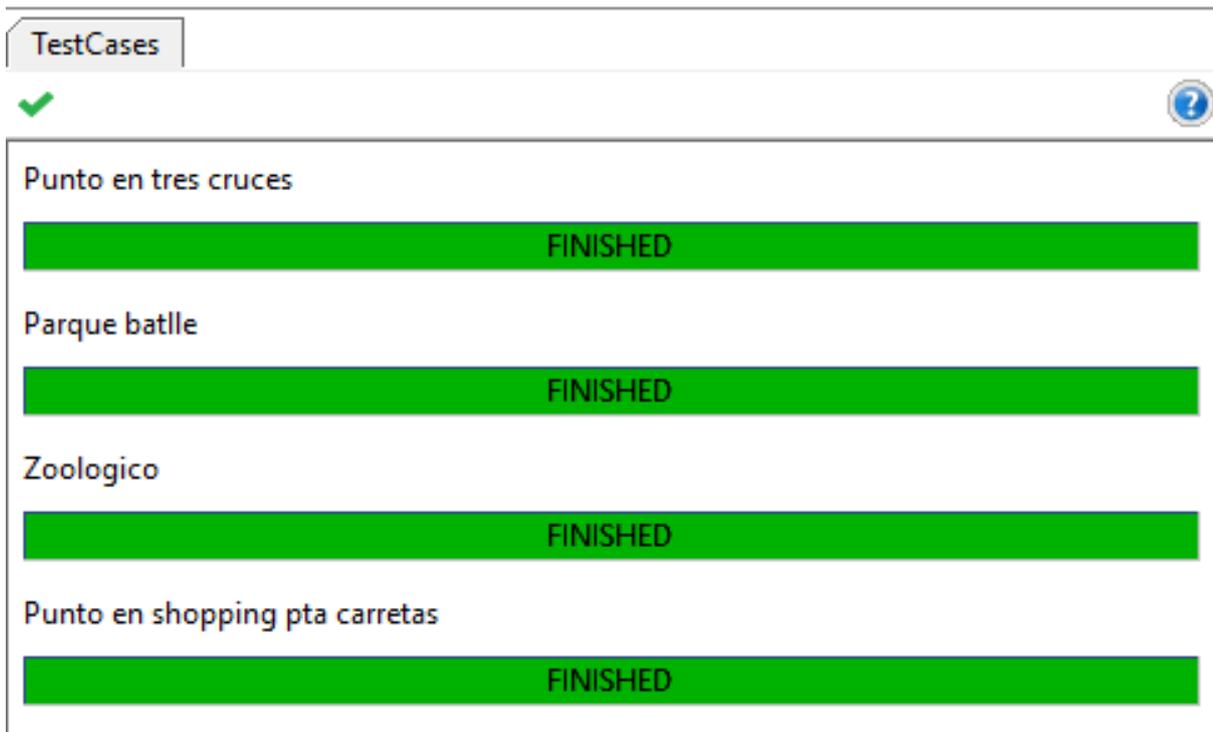


Figura 5.7: Test Eventos.

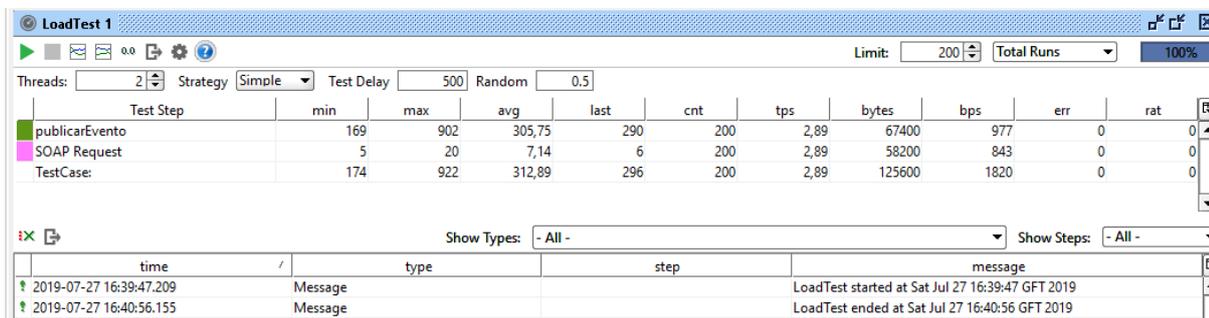


Figura 5.8: Test de envío de 200 Eventos.

Queues:

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued
carlozve@gmail.com	200	0	200	0
gaston231@gmail.com	0	1	200	200

Figura 5.9: Envío a colas de mensajes de clientes.

Para la prueba de carga se generó un evento con una geometría de 1530 puntos y se efectuaron un total de 20 ejecuciones con una demora de 100 ms entre cada una, en un tiempo total de ejecución de 8 segundos, como se puede observar en la Figura 5.10. Allí también se puede visualizar las 20 ejecuciones con la cantidad de bytes enviados y el promedio de cada ejecución que fue de 304,54 ms.

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
publicarEvento	216	828	375,8	218	20	2,14	6740	721	0	0
SOAP Request	5	13	7,6	5	20	2,14	5820	623	0	0
TestCase:	221	841	383,4	223	20	2,14	12560	1344	0	0

time	type	step	message
2019-07-27 18:08:57.163	Message		LoadTest started at Sat Jul 27 18:08:57 GFT 2019
2019-07-27 18:09:06.475	Message		LoadTest ended at Sat Jul 27 18:09:06 GFT 2019

Figura 5.10: Envío de polígonos con múltiples puntos.

5.3.2. Consultar alertas

En este caso, se verifica que se envíe a los clientes las notificaciones correspondientes y que puedan ser leídas mediante la aplicación móvil o el correo electrónico.

Utilizando las alertas generadas en la Sección 5.3.1 se pudo probar la funcionalidad de «Consultar Alertas» dado que se había generado gran volumen de datos.

En todos los casos, el cliente pudo ver tanto las alertas recibidas previamente como las nuevas. Además, se verificó que las alertas se vieran reflejadas en los listados de la administración web 4.3.2.1.

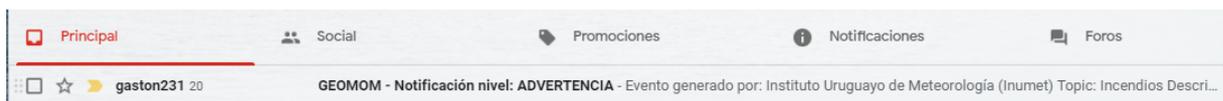


Figura 5.11: 20 notificaciones por email

En la Figura 5.11 se aprecian las veinte notificaciones por email recibidas para el caso de la Figura 5.10.

5.3.3. Notificar alerta

Mediante las pruebas anteriormente mencionadas se verifica que la plataforma genere las alertas a clientes según corresponda por matching geográfico y se comprueba que las notificaciones hayan sido generadas según el canal de la suscripción. Al momento de obtener todas las alertas, la plataforma filtra correctamente las alertas que hayan expirado.

5.3.4. Enviar ubicación

Uno de los servicios ofrecidos a los clientes que más exige a la plataforma es el de «Enviar ubicación». Este servicio es exigente porque cuando los clientes se suscriben a un Topic mediante «Ubicación en tiempo real», la aplicación envía periódicamente su ubicación al Broker de mensajería. Según la implementación de cada aplicación cliente, el envío se puede dar cada pocos segundos y por lo tanto la plataforma debe estar preparada para recibir ráfagas de muchas ubicaciones en pocos segundos.

Para el testing de este servicio se realizaron pruebas de estrés en SoapUI que simula diez clientes enviando, de forma simultánea, su ubicación.

En la Figura 5.12 se pueden ver los diez hilos que genera el SoapUI dónde cada uno consume el servicio de «Enviar ubicación con un cliente distinto».

Esta prueba se ejecutó por 46 segundos y generó 1000 registros de ubicaciones divididas entre los diez clientes. El tiempo promedio de respuesta de cada cliente oscila entre 378 ms y 413 ms. En conclusión, no se detectaron problemas durante la prueba y la plataforma reaccionó correctamente.

5.3.5. Validaciones Administración Web

Luego de haber generado gran número de datos para la plataforma mediante los casos de prueba anteriormente descritos, se pueden verificar los casos de uso referentes a la administración web. De esta manera, se accede a los distintos reportes generados a través de las funcionalidades, verificándose la corrección de los datos generados, el envío y recepción de mensajes a clientes, así como las gráficas correspondientes a cada listado.

También se verificó el caso de uso que elimina los clientes inactivos. En estos casos se descartaron correctamente los mensajes de los clientes eliminados, sus suscripciones

LoadTest 2

Limit: 10 Runs per Thread 100%

Threads: 10 Strategy Burst Burst Delay 5 Burst Duration 60

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
cliente1	321	607	413,22	366	100	2,47	8500	209	0	0
cliente10	133	513	378,67	133	100	2,47	8500	209	0	0
cliente2	328	636	404,5	484	100	2,47	8500	209	0	0
cliente3	318	587	400,09	332	100	2,47	8500	209	0	0
cliente4	319	555	403,13	350	100	2,47	8500	209	0	0
cliente5	301	586	400,5	324	100	2,47	8500	209	0	0
cliente6	307	552	397,93	307	100	2,47	8500	209	0	0
cliente7	201	600	393,66	201	100	2,47	8500	209	0	0
cliente8	159	513	393,88	159	100	2,47	8500	209	0	0
cliente9	147	573	400,76	148	100	2,47	8500	209	0	0

Show Types: - All - Show Steps: - All -

time	type	step
2019-07-27 17:00:06.839	Message	
2019-07-27 17:00:52.364	Message	

Figura 5.12: Test Enviar Ubicación en SoapUI

y la cola de mensaje en el Broker de mensajería.

Se verificó el caso de uso de Extracción de datos, comprobando que se encuentran todos los datos generados por la plataforma.

Capítulo 6

Gestión de Proyecto

6.1. Descripción

En el presente capítulo se describe la gestión del proyecto, los hitos relevantes, el cronograma presentado al inicio del proyecto y el presentado al final, así como una estimación de el tiempo dedicado a las diferentes etapas del proyecto.

6.2. Hitos

Parte relevante de la gestión es identificar los hitos y enmarcarlos en las etapas del proyecto.

- 09/03/2018 - Postulación al proyecto de grado.
- 16/04/2018 - Presentación de Tesis de maestría y generación del calendario tentativo.
- 07/08/2018 - Primera versión del marco teórico y documento de Arquitectura.
- 04/12/2018 - Primera versión del GeoMOM Central, Broker de Mensajería y aplicación Android.
- 20/12/2018 - Recepción del documento final de la Tesis de maestría.
- 19/03/2019 - Envío de calendario actualizado.
- 09/04/2019 - Presentación de avances del proyecto en el curso Taller de Sistemas de Información Geográficos Empresariales.

- 08/05/2019 - Primera versión del informe final.

A continuación se presenta en la Figura 6.1 el diagrama de Gantt establecido al inicio del proyecto y su correspondiente diagrama luego de finalizado en la Figura 6.2.

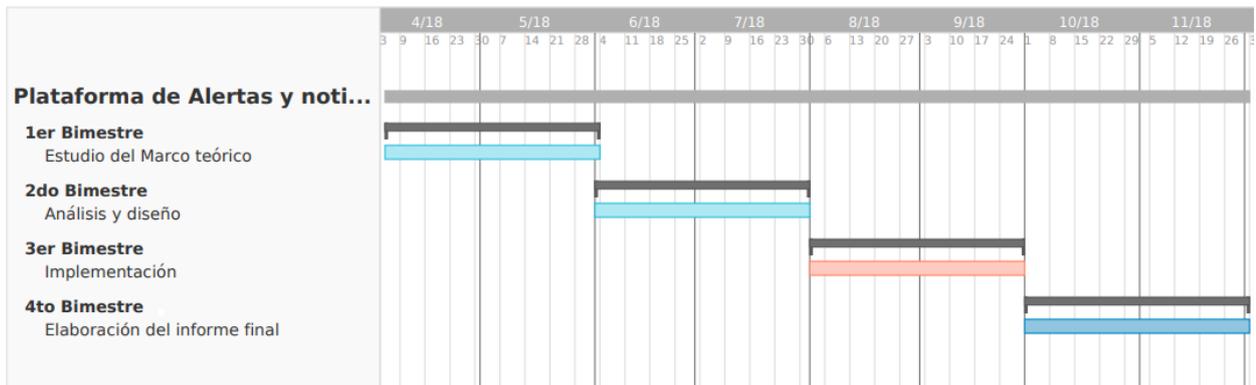


Figura 6.1: Gantt Inicial

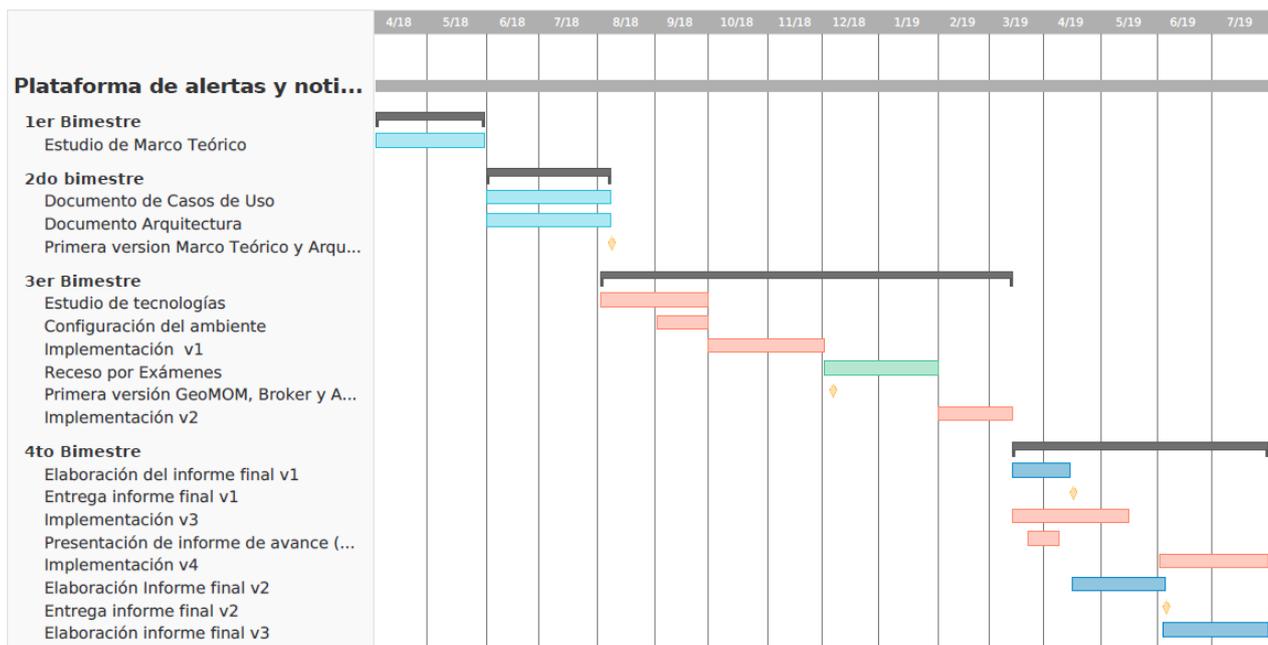


Figura 6.2: Gantt final

Se puede apreciar en los diagramas que hasta el segundo bimestre no hubo atrasos significativos, cumpliendo con el estudio del marco teórico, el análisis y el diseño en base a lo presentado en la tesis de referencia.

El tercer bimestre fue dedicado a el estudio de tecnologías y generación del ambiente de trabajo para la implementación, proceso que llevó dos meses.

La implementación comenzó a principios del mes de octubre y se extendió por un total de 30 semanas.

Las 9 semanas de recesos fueron marcados en los meses de diciembre y enero por exámenes rendidos.

El informe final se comenzó a elaborar a mitad de marzo y extendiéndose durante 20 semanas hasta la finalización del proyecto.

La duración total del proyecto, sin contabilizar las semanas de receso, fue de 66 semanas divididas de la siguiente forma:

- Marco teórico: 8 semanas
- Documentación de Casos de uso y Arquitectura: 8 semanas
- Implementación: 30 semanas
- Elaboración del informe final: 20 semanas

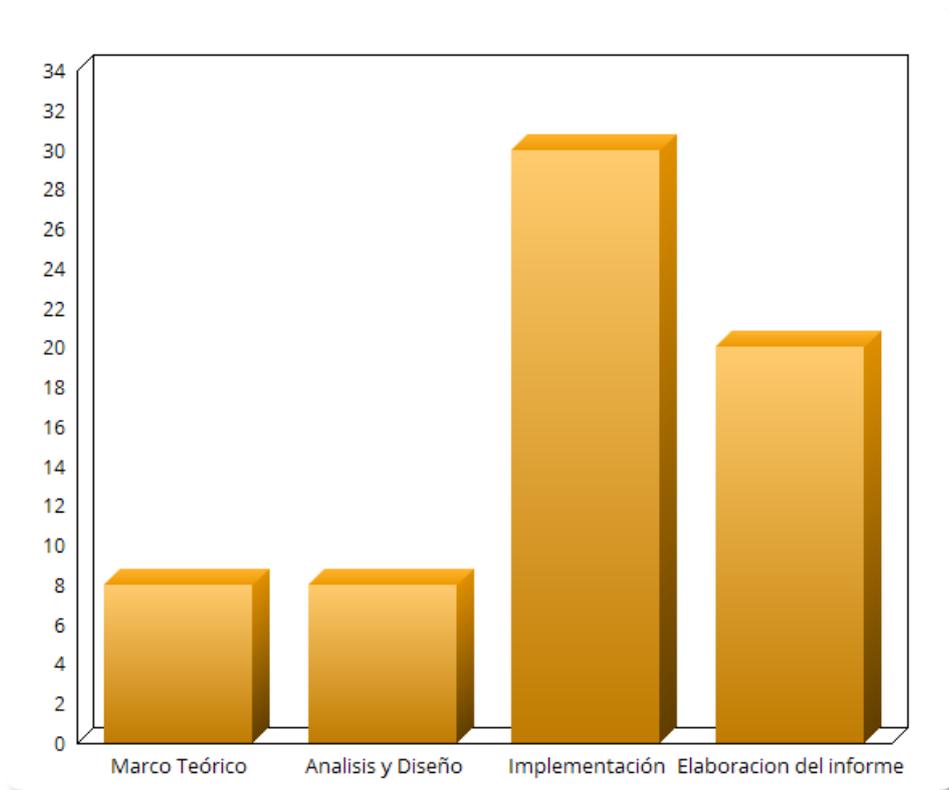


Figura 6.3: Cantidad de semanas trabajadas por área

6.3. Repositorio Git

Dado que mucho del trabajo de implementación se hizo a distancia, fue necesario la utilización de un repositorio para el código generado. El uso de esta herramienta permitió realizar un versionado del proyecto y mitigar el riesgo ante eventuales pérdidas del trabajo realizado.

Se optó por utilizar Bitbucket [36] como repositorio Git, ya que es una herramienta previamente utilizada por los integrantes de este proyecto.

Capítulo 7

Conclusiones y trabajo a futuro

En el presente capítulo se presentan las conclusiones finales de proyecto, así como algunas consideraciones generales sobre posibles trabajos a futuro.

7.1. Conclusiones

El proyecto se centró en la implementación de la plataforma propuesta en la tesis de referencia [1]. Para esto fue necesario realizar un análisis en profundidad de la propuesta, teniendo en cuenta los requerimientos funcionales y no funcionales para así determinar los componentes y funcionalidades a desarrollar.

Se tomaron como base los casos de uso propuestos en la tesis de referencia y se extendieron con casos de uso de Administración para la plataforma. Es de suma importancia la implementación de esta web de administración, ya que con estas funcionalidades se logra gestionar los recursos de la plataforma, los Clientes y los Proveedores.

El proyecto incluyó el estudio de trabajos relacionados con la temática en distintas áreas, como la académica, la industrial y la comercial. Una vez analizada dicha información, se diseñó el documento de arquitectura y de casos de uso que fue el principal insumo para la implementación de la plataforma.

Luego de trazados los requerimientos centrales de la arquitectura y de definido el alcance de los mismos, se implementaron dichos requerimientos de forma satisfactoria, cumpliendo con los objetivos planteados.

Con el desarrollo de esta plataforma se logró generar un espacio centralizado en la órbita del Estado Uruguayo. Allí, los Proveedores de información pueden generar eventos dirigidos a toda la población; sin embargo, los usuarios disponen de opciones

para elegir el tipo de alertas que desean recibir en función de sus necesidades e intereses. De esta manera se evita abrumar a los ciudadanos con alertas o notificaciones de eventos que podrían no serles provechosas.

7.2. Trabajo a futuro

Si bien se logró cumplir con los objetivos principales y secundarios planteados al principio del proyecto, luego de un análisis exhaustivo y de la puesta en práctica de la plataforma, se identificaron ciertos aspectos a mejorar y posibles lineamientos de trabajo a futuro:

- Evaluar la escalabilidad de la plataforma en cuanto al diseño de su arquitectura de colas de mensajes por usuario. Evaluar el rendimiento en escenarios con miles de usuarios.
- Implementar la prioridad de envío de mensajes según su topic correspondiente, si bien se diseñó de esa forma, no se implementó una prioridad al momento de procesar los mensajes.
- Implementar el envío de notificaciones a los usuarios por SMS (mensaje de texto) u otras opciones que se consideren oportunas.
- Incluir criterios de seguridad en los servicios web SOAP y REST, tanto del Geo-MOM Central como en el Broker de mensajería.
- Evaluar la viabilidad de permitir el acceso de los Proveedores a la Web de Administración con el fin de que puedan hacer uso de los listados estadísticos.

Bibliografía

- [1] Miguel Merlino. «Extensión Espacial de Middleware orientado a mensajería para dar soporte a servicios basados en geolocalización». En: (2018).
- [2] AGESIC. *Plataforma de Gobierno Electrónico*. https://www.agesic.gub.uy/innovaportal/v/1496/1/agesic/plataforma_de_gobierno_electronico_.html. [Accedido Abril 2019]. 2019.
- [3] Instituto Uruguayo de Meteorología. <https://www.inumet.gub.uy/>. [Accedido Mayo 2019]. 2019.
- [4] Instituto Uruguayo de Meteorología. *Nuevo sistema de visualización espacial de las alertas meteorológicas*. <https://www.inumet.gub.uy/index.php/sala-de-prensa/noticias/nuevo-sistema-de-visualizacion-espacial-de-las-alertas-meteorologicas>. [Accedido Abril 2019]. 2017.
- [5] Google Play. *INUMET*. <https://play.google.com/store/apps/details?id=uy.gub.meteorologia.inumet&hl=es>. [Accedido Mayo 2019]. 2019.
- [6] La Vanguardia. *Caza medusas con el móvil cuando vayas a la playa*. <https://www.lavanguardia.com/natural/observados-cameras-naturaleza/20160724/403398854328/medusas-app-ciencia-ciudadana.html>. [Accedido Julio 2019].
- [7] ICM-CSIC. *MedJelly*. <http://medjelly.com/es/>. [Accedido Abril 2019]. 2012.
- [8] BigChange. *JourneyWatch - La solución de seguimiento de vehículos*. <https://www.bigchangeapps.com/es/productos/la-solucion-de-seguimiento-de-vehiculos-que-va-mas-alla/>. [Accedido Abril 2019].

- [9] BigChange. *JobWatch - La solución de gestión integral del personal móvil*. <https://www.bigchangeapps.com/es/productos/la-manera-digital-de-planificar-gestionar-programar-y-realizar-un-seguimiento-de-su-personal-movil-jobwatch/>. [Accedido Abril 2019].
- [10] ABC. *Citycop: una app para denunciar delitos*. <https://www.abc.es/tecnologia/moviles-aplicaciones/20140512/abci-citycop-aplicacion-delitos-seguridad-201405121305.html>. [Accedido Julio 2019].
- [11] IslamicHelp. *Hajj*. <https://www.islamichelp.org.uk/what-we-do/seasonal/qurbani/what-is-hajj/>. [Accedido Abril 2019].
- [12] «An Architectural Design of Geo-Fencing Emergency Alerts System for Hajj Pilgrims.» En: *2018 8th International Conference on Computer Science and Information Technology (CSIT), Computer Science and Information Technology (CSIT), 2018 8th International Conference on* (2018). [Accedido Abril 2019], pág. 1. ISSN: 978-1-5386-4152-1. URL: <http://proxy.timbo.org.uy:443/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsee&AN=edsee.8486189&lang=es&site=eds-live>.
- [13] Diario El País. *Un potente terremoto causa más de 1.800 muertos en Nepal*. https://elpais.com/internacional/2015/04/25/actualidad/1429950325_883537.html. [Accedido Mayo 2019]. 2016.
- [14] «Traveller Alert System.» En: *2015 2nd International Conference on Information and Information and Communication Technologies for Disaster Management (ICT-DM)* (2015). [Accedido Mayo 2019], pág. 285. ISSN: 978-1-4799-9923-1. URL: <http://proxy.timbo.org.uy:443/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsee&AN=edsee.7402018&lang=es&site=eds-live>.
- [15] innoQ. *Ws**. <http://www.innoq.com/soa/ws-standards/poster/>. [Accedido Julio 2019].
- [16] JD Zimmer Scott. «Location-based services (LBS).» En: *Salem Press Encyclopedia* (2018). URL: <http://proxy.timbo.org.uy:443/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=109057067&lang=es&site=eds-live>.

- [17] CartouCHe. *Location Based System Architecture*. http://www.e-cartouche.ch/content_reg/cartouche/LBStech/en/html/unit_LBStechU4.html. [Accedido Mayo 2019].
- [18] E. Curry. «Message Oriented Middleware, in Middleware for communications». En: (2018).
- [19] Oracle. *The Java EE 6 Tutorial - Java Message Service Concepts*. <https://docs.oracle.com/javase/6/tutorial/doc/bncdq.html>. [Accedido Mayo 2019].
- [20] Microsoft Azure. *Patrón de publicador y suscriptor*. <https://docs.microsoft.com/es-es/azure/architecture/patterns/publisher-subscriber>. [Accedido Junio 2019].
- [21] JTS Topology Suite. *Function buffer*. <http://www.tsusiatsoftware.net/jts/javadoc/com/vividsolutions/jts/geom/Geometry.html>. [Accedido Mayo 2019]. 2016.
- [22] OSGeo. *Java Topology Suite*. <https://www.osgeo.org/projects/jts/>. [Accedido Mayo 2019].
- [23] Javadoc. *Package com.vividsolutions.jts.geom*. <http://javadoc.com/com.vividsolutions/jts/1.13/com/vividsolutions/jts/geom/package-summary.html>. [Accedido Mayo 2019].
- [24] Apache Software Foundation. *Apache ActiveMQ*. <https://activemq.apache.org/>. [Accedido Julio 2019].
- [25] Oracle. *Java Persistence Api*. <https://docs.oracle.com/javase/6/tutorial/doc/bnbpz.html>. [Accedido Julio 2019].
- [26] Hibernate. *Hibernate Framework*. <http://hibernate.org/>. [Accedido Julio 2019].
- [27] DbVis Software. *DbVisualizer*. <https://www.dbvis.com/>. [Accedido Julio 2019].
- [28] Bootstrap. *Introduction*. <https://getbootstrap.com/docs/4.3/getting-started/introduction/>. [Accedido Julio 2019].
- [29] Spring. *Spring Framework*. <https://spring.io/projects/spring-framework>. [Accedido Julio 2019].

- [30] Google. *Android*. <https://www.android.com/>. [Accedido Junio 2019].
- [31] Google. *Android Studio*. <https://developer.android.com/studio>. [Accedido Junio 2019].
- [32] Square Inc. *Retrofit*. <https://square.github.io/retrofit/>. [Accedido Junio 2019].
- [33] Google. *Firebase Cloud Messaging*. <https://firebase.google.com/products/cloud-messaging/>. [Accedido Mayo 2019].
- [34] Google. *Maps SDK for Android*. <https://developers.google.com/maps/documentation/android-sdk/intro>. [Accedido Julio 2019].
- [35] SmartBear. *SoapUI*. <https://www.soapui.org/>. [Accedido Julio 2019].
- [36] Atlassian. *Bitbucket*. <https://bitbucket.org/>. [Accedido Julio 2019].