

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

**DISEÑO Y DESARROLLO DE UNA APLICACIÓN MÓVIL PARA
GEOREFERENCIAR SITIOS DE INTERÉS**



Realizado por:

Andrew Davis Escalona

Tutor:

Prof. Antonio Leal

Caracas, Octubre de 2013

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por el Bachiller Andrew Olivier Davis Escalona, con el título: “Diseño y Desarrollo de una Aplicación Móvil para Georeferenciar Sitios de Interés”, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 24 de Octubre del 2013, para que su autor lo defendiera en forma pública, lo que hizo en el aula 1, de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente Acta, en Caracas a los veinticuatro días del mes de octubre del año dos mil trece.

Prof. Antonio Leal

Tutor

Prof. Rossana Díaz

Jurado

Prof. Robinson Rivas

Jurado

DEDICATORIA

Dedicado a la memoria de mi abuela materna María Paula Escalona. Quien fue para mí una segunda madre y cuyos consejos y cariños ayudaron a construir la persona que hoy soy en día. ¡Gracias Abuela! Donde quieras que estés, sé que estarás muy orgullosa de tu nieto.

AGRADECIMIENTOS

Quiero agradecer primeramente a mi madre Reina Antonia Escalona. La mujer que me dio la vida y que además también ha sido un padre para mí.

Agradezco también a la Universidad Central de Venezuela. La cual fue mi segunda casa por casi 8 años que llevo estudiando en esta magnífica institución.

Agradezco a la Facultad de Ciencias y a todos sus profesores con los que tuve la oportunidad de aprender. Además de todos esos compañeros y amigos que hice dentro de estas instalaciones.

Agradezco a la que fue co-tutora de seminario, la Lic. Lisbeth Mavarez, ya que sin sus guías y sabios consejos, el desarrollo de este trabajo hubiera sido mucho más complicado.

Agradezco a mi tía Zuleyma Escalona y al Profesor Alfredo Escalona, por haberme ayudado a ingresar a la que siempre será mi universidad.

Agradezco a mi tía Pilar Pérez y a su hijo Domingo Pérez (mi padrino) cuyos cariños y palabras de afecto y apoyo siempre estuvieron presentes para mí.

Agradezco a la Organización de Bienestar Estudiantil (OBE) por la pequeña (pero importante) ayuda económica que me brindaron mes tras mes.

Por último pero para nada menos importante quiero agradecer a dios todopoderoso, quien estoy seguro me acompañó en algunos momentos difíciles que tuve que pasar durante esta travesía.

¡Gracias a Todos!

ÍNDICE

RESUMEN	9
INTRODUCCIÓN	10
Objetivos Generales y Específicos	12
Objetivo General.....	12
Objetivos Específicos.....	12
CAPITULO 1: Servicios Basados en Localización	14
1.1 Definición y Antecedentes.....	14
1.2 Componentes de Servicios Basados en Localización	15
1.2.1 Localización por GPS	17
1.2.2 Localización por Wifi	19
1.2.3 Localización por Red Celular	19
1.3 Áreas de aplicación de Servicios Basados en Localización	20
1.3.1 Aplicaciones de redes sociales	20
1.3.2 Publicidad y Mercadeo	22
1.3.3 Servicios de Rastreo	23
CAPITULO 2: Framework PhoneGap	25
2.1 Definición y Antecedentes.....	25
2.2 Características de PhoneGap.....	27
2.2.1 Descripción del API.....	27
2.2.2 Implementación en Distintas plataformas	28
2.2.3 Servicio PhoneGap Build.....	30
2.2.4 Manejo de Geolocalización	31
CAPITULO 3: Librería OpenLayers.....	33
3.1 Definición y Antecedentes.....	33
3.2 Manejo de Capas Geográficas	34
3.2.1 Capas de Google Maps	34
3.2.2 Capas de Bing Maps.....	36
3.2.3 Capas de OpenStreetMap.....	37
3.3 Manejo de Objetos Geográficos	38
Capítulo 4: Análisis y diseño de la aplicación móvil	40

4.1	Casos de Uso.....	40
4.1.1	Actores	40
4.1.2	Nivel 1	41
4.1.3	Nivel 2	44
4.2	Diagramas de Clases	53
4.3	Diagramas de Secuencia	55
4.4	Diagrama de Implementación	58
CAPITULO 5: Implementación.....		59
5.1	Jquery Mobile	59
5.2	JSON.....	61
5.3	Pantalla Inicial	64
5.3.1	Crear Usuario	65
5.3.2	Iniciar Sesión	67
5.4	Menú Principal.....	68
5.4.1	Nuevo Sitio.....	68
5.4.2	Sitios.....	71
5.4.3	Checkpoints	72
5.4.4	Hacer Check-in.....	74
5.4.5	Mapa.....	75
5.4.6	Configuración.....	78
5.4.7	Alertas	81
5.4.8	Sincronización	82
5.5	Base de Datos Geoespacial	83
5.6	Pruebas de la aplicación	85
CONCLUSIONES		88
REFERENCIAS BIBLIOGRAFICAS		91

ÍNDICE DE FIGURAS Y TABLAS

Figura 1: SBL reflejado por la convergencia de distintas tecnologías [U-SNLBS].....	15
Figura 2: Componentes de un SBL	16
Figura 3: Representación de satélites GPS en su órbita sobre la tierra.....	18
Figura 4: Captura de la aplicación Foursquare.....	21
Figura 5: Captura de la aplicación Loopt.....	22
Figura 6: Captura del sistema Prey mostrando la ubicación de un dispositivo móvil reportado como perdido.....	24
Tabla 7: Características soportadas por PhoneGap en distintos SO móviles [U-PGAPW]	26
Figura 8: Modo de implementar PhoneGap en Android	29
Figura 9: Jerarquía de directorios de un Proyecto BlackBerry usando PhoneGap	30
Figura 10: Ficha de aplicación móvil subida a PhoneGap Build	31
Figura 11: Mapa Sencillo de OpenLayers	33
Figura 12: Mapa OpenLayers usando capa Google Maps	35
Figura 13: Mapa OpenLayers usando capa Bing Maps.....	36
Figura 14: Mapa OpenLayers usando capa OpenStreetMap (OSM)	38
Figura 15: Mapa OpenLayers mostrando objetos geográficos	39
Figura 16: Visión general de la aplicación móvil.....	41
Figura 17: Casos de Uso del Nivel 2.....	44
Figura 18: Diagrama de Clases de la aplicación móvil	53
Figura 19: Diagrama de Clases del servidor	54
Figura 20: Diagrama de secuencia de creación de Nuevo Sitio	55
Figura 21: Diagrama de secuencia de Visita en Checkpoint	56
Figura 22: Diagrama de secuencia de Sincronización.....	57
Figura 23: Diagrama de implementación.....	58
Figura 24: Código HTML de la aplicación usando JQuery Mobile	60
Figura 25: Resultado del código usado en la figura anterior.....	60
Figura 26: Código JavaScript que muestra la implementación de eventos JQuery Mobile	61
Figura 27: Código JavaScript perteneciente a la función de Sincronización de la aplicación móvil.....	62
Figura 28: Código JavaScript perteneciente la función de Sincronización (recepción de la respuesta).....	62
Figura 29: Código Java para transformar JSON a objetos Java.....	63
Figura 30: Código Java para transformar objetos Java a JSON.....	64
Figura 31: Vista de la pantalla inicial de la aplicación móvil	65
Figura 32: Formulario de creación de nueva cuenta de usuario	65
Figura 33: Mensaje de cuenta de usuario creada con éxito	66
Figura 34: Mensaje mostrado cuando el usuario introduce un dato incorrecto	66
Figura 35: Mensaje mostrado al iniciar sesión correctamente	67
Figura 36: Mensaje mostrado al fallar la autenticación de inicio de sesión.....	67
Figura 37: Menú principal de la aplicación móvil	68
Figura 38: Pantalla inicial de la opción “Nuevo Sitio”	68

Figura 39: Formulario de creación de nuevo sitio.....	69
Figura 40: Mensaje notificando el éxito de la creación de un nuevo sitio	69
Figura 41: Mensaje notificando la necesidad de indicar un nombre para el sitio a crear ..	70
Figura 42: Formulario para crear una nueva categoría	70
Figura 43: Pantalla inicial de la opción “Sitios” del menú principal	71
Figura 44: Pantalla inicial de la opción Sitios del menú principal con un renglón desplegado	71
Figura 45: Perfil de un sitio creado.....	72
Figura 46: Lista de Checkpoints.....	73
Figura 47: Perfil de un checkpoint.....	73
Figura 48: Lista de Checkpoints cercanos	74
Figura 49: Mensaje de confirmación de registro de checkpoint.....	75
Figura 50: Mapa principal de la aplicación móvil	75
Figura 51: Mapa mostrando la posición geográfica actual del dispositivo móvil	76
Figura 52: Lista de tipos de mapa disponibles	76
Figura 53: Ventana flotante de la opción “Mostrar”.....	77
Figura 54: Checkpoints desplegados sobre el mapa de la aplicación	77
Figura 55: Sitios de una categoría desplegados sobre el mapa de la aplicación	78
Figura 56: Opciones de configuración de la aplicación móvil	78
Figura 57: Formulario para cambiar contraseña.....	79
Figura 58: Opción para cambiar el Servicio de Mapas	79
Figura 59: Opción para configurar el servidor de sincronización	80
Figura 60: Opción para configurar las categorías.....	80
Figura 61: Bandeja de entrada de alertas	81
Figura 62: Contenido de una alerta.....	81
Figura 63: Mensaje de éxito de sincronización con el servidor.....	82
Figura 64: Mensaje de error de sincronización con el servidor.....	82
Figura 65: Instrucción SQL para crear tabla de checkpoints en la base de datos geoespacial	84
Figura 66: Instrucción SQL para insertar un registro en la tabla de checkpoints	84

RESUMEN

El presente Trabajo Especial de Grado es el resultado de una propuesta de combinar tecnologías de geolocalización con tecnologías móviles, para diseñar y desarrollar una aplicación móvil que permita georeferenciar sitios de interés para el usuario.

Durante el proceso de análisis en el desarrollo de este Trabajo Especial de Grado, se establecieron los casos de usos indispensables para una aplicación móvil que utilizando las capacidades de geolocalización del dispositivo, pueda georeferenciar sitios.

Las capacidades de geolocalización de los actuales dispositivos móviles permitirían a un usuario capturar las coordenadas geográficas (además de otros datos relacionados) del lugar donde se encuentra en un momento dado. Con los casos de uso establecidos en el proceso de análisis de este Trabajo Especial de Grado, el usuario puede con los datos obtenidos en la captura utilizar distintas funcionalidades de georeferenciación dentro de la aplicación móvil desarrollada.

La aplicación móvil está desarrollada en una combinación de los productos de software: PhoneGap¹, JQuery Mobile² y OpenLayers³. Estas tecnologías están basadas en los estándares de HTML5, CSS3 y JavaScript

En el desarrollo del presente Trabajo Especial de Grado, se comprueban las ventajas de utilizar las tecnologías descritas al implementar la aplicación móvil.

¹ PhoneGap: Framework para el desarrollo de aplicaciones móviles usando estándares web [<http://phonegap.com/>].

² JQuery Mobile: Framework para el desarrollo de interfaces de usuario móviles en entornos web [<http://jquerymobile.com/>].

³ OpenLayers: Librería para el despliegue y manejo de mapas en entornos web [<http://openlayers.org/>].

INTRODUCCIÓN

Las tecnologías móviles han tenido un gran avance en los últimos años. Los teléfonos celulares han pasado de ser simples dispositivos utilizados para recibir o hacer llamadas y enviar o recibir mensajes de texto, a ahora ser denominados dispositivos inteligentes, más cercanos a computadoras de bolsillo, que proveen variadas funciones de comunicación, entretenimiento y cómputo.

El éxito de las tecnologías móviles no ha estado reservado únicamente para teléfonos celulares, actualmente, el mercado de las tabletas electrónicas está muy activo. Las tabletas brindan a los usuarios una mejor experiencia móvil, ya que poseen mejores resoluciones de pantalla y mayor capacidad de procesamiento.

Las capacidades de geolocalización en los dispositivos móviles es una de las características más interesantes que incorporan ambos dispositivos móviles. La capacidad de geolocalización en el dispositivo móvil, brinda al usuario la capacidad de conocer en tiempo real su ubicación geográfica, que a su vez permite obtener información de los sitios de interés cercanos a su ubicación.

El presente Trabajo Especial de Grado, es el resultado de una propuesta de combinar las capacidades de geolocalización de los dispositivos móviles con frameworks de desarrollo de *app* móviles usando tecnologías web (HTML5). Esto con el fin de desarrollar una aplicación móvil multiplataforma que permita la georeferenciación de sitios de interés para el usuario. Las funciones de georeferenciación son básicamente dos. La primera permite guardar la ubicación actual del usuario, y la segunda permite registrar la visita (hacer *check-in*) en una ubicación específica conocida como “*checkpoint*”, una vez que el usuario se encuentre cercano a su posición. Los “*checkpoints*” son descargados desde una base de datos geoespacial localizada en el servidor y representan lugares que el usuario debe visitar ya que son de su interés.

Para esta experiencia fue necesaria la utilización de un teléfono celular dotado de mecanismos de geolocalización y capaz de conectarse a la red de internet. Básicamente, el dispositivo obtiene la ubicación geográfica actual del usuario, la

cual puede ser usada para crear un “sitio nuevo” o para registrar la visita en un “*checkpoint*”. Estos datos pueden ser visualizados geográficamente a través de servicios de mapas (si se cuenta con acceso a internet) y a su vez pueden ser respaldados en una base de datos geoespacial localizada en el *backend* del sistema (lado del servidor).

Se establecieron los casos de usos necesarios, para definir cada una de las funcionalidades que tiene la aplicación móvil. Así como los distintos artefactos necesarios para diseñar el funcionamiento de la aplicación.

En este Trabajo Especial de Grado, se establece el objetivo general y se proponen los objetivos específicos. A continuación, el primer capítulo trata del tema de los servicios basados en localización y sus componentes. El segundo capítulo aborda el framework PhoneGap para el desarrollo de aplicaciones móviles. El tercer capítulo trata sobre la librería de despliegue de mapas OpenLayers. El cuarto capítulo se aborda el análisis y diseño de la aplicación móvil y el servidor. El quinto capítulo aborda la implementación tanto de la aplicación móvil como la aplicación en el servidor, a continuación se ofrecen las conclusiones y seguidamente la bibliografía empleada para el desarrollo del Trabajo Especial de Grado

Objetivos Generales y Específicos

A continuación se establece el objetivo general y los objetivos específicos del Trabajo Especial de Grado.

Objetivo General

El objetivo general consiste en diseñar y desarrollar una aplicación móvil que permita la georeferenciación de sitios de interés por parte del usuario.

Objetivos Específicos

Para satisfacer el objetivo general, se plantean los siguientes objetivos específicos:

- Diseñar y desarrollar funcionalidades, que faculden a la aplicación móvil para obtener la localización del dispositivo (en forma de coordenadas geográficas), de manera que pueda permitirle al usuario guardar y visualizar esta localización con los atributos necesarios, los cuales pueden ser nombre, descripción, categoría entre otros.
- Diseñar y desarrollar un módulo de autenticación para el usuario en la aplicación móvil, de manera que la aplicación sólo cargue la información geográfica de interés del usuario autenticado.
- Diseñar y desarrollar funcionalidades de gestión de los datos georeferenciados en la aplicación móvil, permitiendo al usuario realizar tareas de creación, consulta, eliminación, modificación o eliminación de estos datos.
- Diseñar y desarrollar mecanismos de despliegues de mapas en la aplicación móvil, que pueda utilizar distintos servicios de información geográfica, de manera que el usuario pueda visualizar la información georeferenciada en estos mapas.
- Diseñar y desarrollar funcionalidades del lado del servidor que respondan a las peticiones de sincronización recibidas por los dispositivos móviles. El objetivo es proveer persistencia de datos geográficos entre la aplicación móvil y una base de datos geoespacial.

- Diseñar y desarrollar un mecanismo de comunicación entre la aplicación móvil y el servidor, que garantice la integridad de los datos durante la recepción y/o envío de información geográfica desde o hacia el servidor.

CAPITULO 1: Servicios Basados en Localización

En este capítulo se tratan los Servicios Basados en Localización (SBL) explicando sus características, los componentes que integra y algunas aplicaciones que usan esta tecnología.

En el capítulo 1.1 se definen los SBL y exponen sus antecedentes. Seguidamente, en el capítulo 1.2 se explicarán algunas características de estos servicios, describiremos sus componentes y distintas formas de hacer localización. En el capítulo 1.3 se describirán algunas aplicaciones del dominio de los SBL en distintas áreas.

1.1 Definición y Antecedentes

Tradicionalmente se ha definido un Servicio Basado en Localización (SBL) como un servicio que obtiene información relacionada a la ubicación geográfica de un dispositivo. Sin embargo, según los autores de [R&A2011] esta definición ha quedado algo desactualizada para lo que ha sido el auge de la definición Web 2.0 estos últimos años. La definición Web 2.0 plantea que el usuario es el protagonista de la creación de contenido en los servicios de información de acceso abierto. La definición actualizada de un servicio basado en localización es:

- El usuario es capaz de determinar su ubicación geográfica.
- La información proporcionada es espacialmente relacionada a la ubicación del usuario.
- El usuario tiene una interacción dinámica con la ubicación o contenido proporcionado y puede enviar u obtener información relacionada con la ubicación.

Con la masificación de dispositivos móviles con capacidades de geolocalización, se ha abierto un nuevo mundo de posibilidades en el diseño y desarrollo de aplicaciones móviles. El diseño y desarrollo de aplicaciones que hacen uso de

SBL y Sistemas de Información Geográfica (SIG⁴), pueden proveer a usuarios de aplicaciones interactivas que muestren contenido en tiempo real basada en la ubicación del usuario.

En la siguiente figura se muestra una representación en forma de intersección de conjuntos, como los SBL reflejan la convergencia de las tecnologías de Internet, dispositivos móviles y SIG.

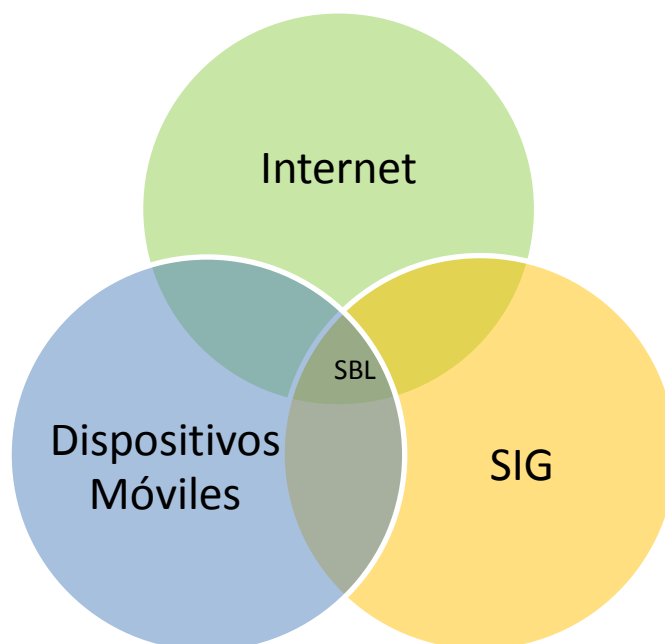


Figura 1: SBL reflejado por la convergencia de distintas tecnologías [U-SNLBS]

1.2 Componentes de Servicios Basados en Localización

Para que un SBL pueda proveer sus servicios, es necesario que cuente con una serie de componentes clave. Los autores de [R&A2011] definen cuatro componentes clave que son mostrados en la figura 2 y explicados a continuación.

⁴ Sistema de Información Geográfica: es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar información geográficamente.

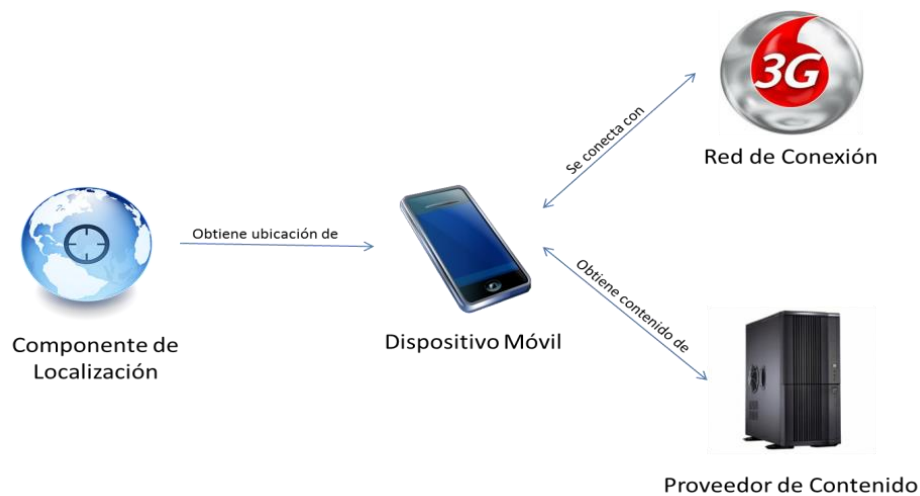


Figura 2: Componentes de un SBL

Dispositivo Móvil: Cuando se habla de dispositivos móviles, nos referimos a todo aquel dispositivo electrónico que es capaz de conectarse a una red móvil para transferir voz y/o datos. Estos dispositivos a su vez, deben contar con opciones de posicionamiento geográfico para que puedan hacer uso de los SBL. Actualmente, los teléfonos inteligentes o tabletas son los dispositivos más populares que utilizan estos servicios.

Proveedor de Contenido: Es la entidad encargada de generar y/o proveer el contenido suministrado a los dispositivos móviles. Por lo general, estos proveedores hacen uso de SIG para desplegar mapas e información geográfica en los dispositivos (en caso que no se encuentre precargada en el dispositivo). Los proveedores de contenido haciendo uso de la ubicación recibida del dispositivo, pueden suministrar información relacionada a esta localización.

Red de Comunicación: Es la red de telefonía móvil utilizada para el intercambio de información entre el dispositivo y los proveedores de contenido. Es la red que utiliza el dispositivo para comunicarse (por ejemplo, una red 3G). Para los desarrolladores de aplicaciones que usan SBL, el funcionamiento de esta red es transparente y el desarrollador no puede controlar su funcionamiento.

Componente de Posicionamiento: Es el componente encargado de suministrar la ubicación del dispositivo, el cual se encuentra en algún servidor expuesto a través de un API⁵ y disponible para otras aplicaciones. Este componente puede suministrar la ubicación a través de tres métodos que se explicarán en los siguientes sub-capítulos. Al igual que la red de comunicación, el Componente de Posicionamiento también está disponible para el diseño y desarrollo de aplicaciones y por lo general puede ser accedido a través de un API que provee el dispositivo.

Según el autor de [U-QMLBS] un SBL debe ser capaz de responder tres preguntas: “¿Dónde estoy?”, “¿Qué hay alrededor de mí?” y “¿Cómo puedo llegar a un lugar?”. Para hacer esto el dispositivo móvil debe poder obtener la ubicación geográfica del usuario. En las siguientes secciones se explican los tres mecanismos para obtener esta ubicación.

1.2.1 Localización por GPS

El sistema GPS (*Global Position System*⁶) consta de tres partes principales: los satélites, los receptores y el control terrestre. Este sistema se compone de 24 satélites distribuidos en seis órbitas polares diferentes, situadas a 2.169 kilómetros de distancia de la Tierra. Cada satélite circunvala la tierra dos veces cada 24 horas. Por encima del horizonte siempre están “visibles” por lo menos 4 satélites para los receptores, de forma tal que estos puedan operar correctamente desde cualquier punto de la Tierra donde se encuentren situados.

⁵ API: Acrónimo para *Application programming interface*. Es una especificación funcional pretendida para ser usada como interfaz de otros programas.

⁶ *Global Position System*: Sistema de Posicionamiento Global. Es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros

En la figura 3 se visualiza una representación de cómo se distribuyen los satélites GPS en la órbita de la tierra.

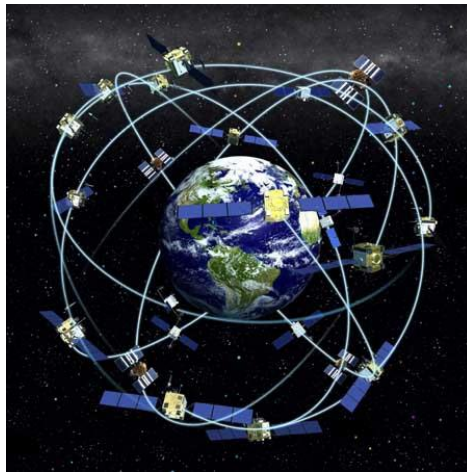


Figura 3: Representación de satélites GPS en su órbita sobre la tierra

El funcionamiento del sistema GPS se basa en el principio matemático de la triangulación, que plantea que para calcular la posición de un punto, es necesario que el receptor GPS determine con exactitud la distancia que lo separa de los satélites. Con la aplicación del principio matemático de la triangulación podemos conocer el punto o lugar donde nos encontramos situados a través de la órbita de los satélites, e incluso rastrear y ubicar el origen de una transmisión por ondas de radio. El sistema GPS utiliza el mismo principio, pero en lugar de emplear círculos o líneas rectas, crea esferas virtuales o imaginarias para lograr el mismo objetivo.

Para dispositivos móviles es el método más preciso de geolocalización (con una precisión de 4 a 40 metros dependiendo de la condición atmosférica). Sin embargo, posee algunas desventajas, como es la del alto consumo de batería y poca recepción en lugares cerrados (interiores de edificios y túneles) [EJ-GPS].

1.2.2 Localización por Wifi

La geolocalización por GPS es muy útil en aplicaciones web y móviles. Pero por razones físicas (como puede ser la de estar en un espacio cerrado sin vista al cielo) el GPS no siempre está disponible. Gracias a un esfuerzo de empresas como Google, Apple, Microsoft y Skyhook, es posible obtener coordenadas con sólo usar antenas Wifi; a continuación se explica su funcionamiento.

Cuando el dispositivo móvil tiene Wifi activado captura todas las redes inalámbricas cercanas. Esto sin importar que sean redes protegidas o publicas ya que no se usa el nombre de la red, sino la dirección MAC del router inalámbrico. Este es un número único del router que nunca cambia, incluso si el router es formateado o la red cambia de nombre, además que siempre va en texto plano entre las ondas.

Tras capturar una lista de direcciones MAC de los routers cercanos, el teléfono utilizando el GPS tradicional, obtiene las coordenadas geográficas y las manda a los servidores de geolocalización (*Google Geolocation*, *Skyhook*, *Microsoft Bing Geocode*, entre otros). Los *routers* rara vez cambian de lugar por lo que la información de estas bases de datos suelen ser bastante exacta. Cuando el dispositivo móvil intente obtener la ubicación actual y no tenga posibilidad de usar la localización por GPS, podrá entonces consultar los servidores de geolocalización y con las direcciones MAC de los routers inalámbricos cercanos hacer una triangulación de sus posiciones y obtener una ubicación geográfica relativamente precisa (aunque no tan precisa como la obtenida por GPS) [WF2012].

1.2.3 Localización por Red Celular

La localización por red celular (también conocida como localización GSM), se emplea el mecanismo de posicionamiento hiperbólico, el cual puede detectar la posición de un teléfono móvil, computando la diferencia en los tiempos de llegada de las señales que son emitidas desde el objeto a tres o más receptores.

Según [NA-GSM] existen tres métodos para lograr esto, los cuales son explicados a continuación:

Método basado en red: Usando este método se usa la infraestructura del proveedor de red para determinar la ubicación del dispositivo móvil. Los niveles de precisión usando este método pueden variar y por lo general tienen que ver con el número de bases celulares disponibles al momento. Mientras que el método de identificación de celda es el menos preciso, algunos métodos más avanzados como el AFLT (*Advanced Forward Link Trilateration*⁷) suelen tener una precisión bastante buena.

Método basado en dispositivos: Usando este método la ubicación del dispositivo es determinada en la identificación de la red celular y computando la fuerza de las señales provenientes de torres celulares cercanas y otros dispositivos móviles del vecindario. Para usar este método es necesaria la instalación de software especial en los dispositivos, lo cual es un gran inconveniente, ya que es difícil hacerlo sin consentimiento del usuario. Esta técnica únicamente es compatible con teléfonos inteligentes.

Método híbrido: Es posible combinar las bondades de los métodos basados en red y los basados en dispositivos. Esto hace que sea un método bastante preciso ya que combina la información que provee la red de servicio con los datos del dispositivo, sin embargo, también se obtienen las limitaciones de ambos métodos.

1.3 Áreas de aplicación de Servicios Basados en Localización

Los SBL cuentan con múltiples aplicaciones en distintas áreas, a continuación se explican alguna de ellas.

1.3.1 Aplicaciones de redes sociales

Los SBL han tenido un especial auge en el uso de redes sociales (también conocidas como redes “geosociales”). Esto ha generado que compañías

⁷ *Advanced Forward Link Trilateration*: Método de localización que utiliza la diferenciación de tiempos de la señal de llegada en el dispositivo receptor.

relevantes en el mercado de la informática como Facebook, Twitter y Google hayan incorporado elementos de geolocalización a sus servicios, sin embargo, sólo algunas compañías han decidido concentrar 100% en crear aplicaciones que aprovechan al máximo el uso de los SBL.

A continuación se describen algunas de las aplicaciones de redes sociales:

Foursquare: Iniciado en 2007 como un proyecto entre Dennis Crowley y Naveen Selvadurai, y lanzado al público el año 2009 en el festival *South by Southwest* de Austin, Texas. Esta red social de localización se ha convertido en una de las aplicaciones más populares de los últimos años, al alcanzar la no despreciable cifra de 20 millones de usuarios registrados a mediados de abril de 2012, según esta indicado en [U-SAFSQ]. El éxito de Foursquare está relacionado con la estrategia de mezclar el uso de redes sociales, con la capacidad de obtener puntos o “insignias” al visitar ciertos lugares, esto genera una condición de “lealtad” en los usuarios miembros de la comunidad, en la forma de que se ven interesados en explorar distintos lugares, para obtener más puntos. Además, los usuarios pueden conseguir puntos adicionales por dar opiniones o tips sobre los lugares donde hacen *check-in* (acción de confirmar la llegada a un sitio) y compartir sus logros con sus amigos de Facebook y/o Twitter. [http://foursquare.com]. La siguiente figura muestra la captura de una vista de la aplicación Foursquare.

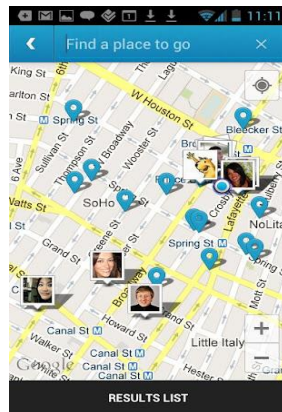


Figura 4: Captura de la aplicación Foursquare

Loopt: Siendo una apuesta parecida a la de Foursquare. Loopt se ha consolidado como una opción más agresiva a la ofrecida por su competidor. Orientada a saber “¿Dónde se encuentran mis amigos?”, esta aplicación ofrece alertas cuando un conocido se encuentra cerca. También ofrece sistema de recompensas similares a los ofrecidos por Foursquare y opción para dejar opiniones (llamados “Qs”) de los lugares visitados. [<http://www.loopt.com>]

En la siguiente figura se aprecia una captura de una vista de la aplicación Loopt.

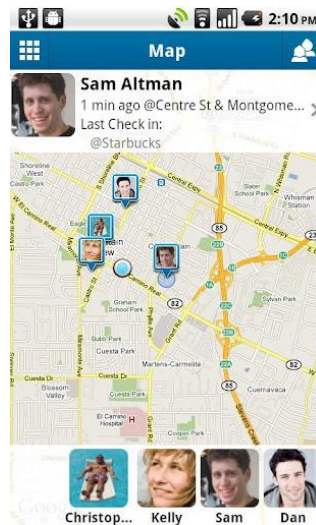


Figura 5: Captura de la aplicación Loopt

SCVNGR: A diferencia de Loopt y Foursquare, esta aplicación está más enfocada a la característica de “juego masivo” que a la de red social. Sus creadores lo definen así: “SCVNGR es un juego y jugarlo es fácil”. A diferencia de otras aplicaciones, SCVNGR ofrece la opción de realizar “retos” asignados por otros miembros de la comunidad. Estos retos no son más que actividades o tareas que los jugadores tienen que realizar en los lugares asignados. Si se cumplen, los jugadores obtienen recompensas virtuales y en algunos casos hasta reales. [<http://www.scvngr.com/>]

1.3.2 Publicidad y Mercadeo

El área de la publicidad y mercadeo también se han usado los SBL. Una aplicación puede aprovechar el interés que tienen los usuarios de conseguir

tiendas o restaurantes con ofertas y descuentos. En algunos países, los comercios, además de usar avisos tradicionales, envían sus ofertas a compañías de mercadeo y publicidad, éstos empleando SBL, pueden enviar mensajes de texto a usuarios cercanos de los lugares publicitados.

Otro uso de SBL en el campo del mercadeo, tiene como ejemplo la aplicación WeReward. Esta aplicación premia al usuario cada vez que visita un lugar promocionado por la aplicación, cargando a éste una pequeña cantidad de dinero real. El usuario luego de realizar diversas visitas a lugares promocionados logra acumular una cierta cantidad de dinero, que puede retirar a través de PayPal.

Otras aplicaciones móviles se basan en la recepción de cupones virtuales o códigos promocionales. Cada vez que un usuario está cercano a un lugar promocionado, este puede recibir el cupón virtual del comercio en cuestión, lo cual puede motivar al usuario de entrar al establecimiento y así canjear el cupón o código promocional. [E-2010]

1.3.3 Servicios de Rastreo

Rastrear la posición de un usuario también es otro uso de los SBL. Este uso ha sido considerado controversial, sin el consentimiento del usuario rastreado. Básicamente el dispositivo móvil posee una aplicación que envía su ubicación actual a través de la red celular a un sistema de rastreo, la ubicación rastreada puede ser visualizada a través de una aplicación web, móvil o de escritorio. Pese a las discusiones de privacidad son muchas las personas que utilizan estos sistemas, por ejemplo, rastrear donde se encuentran sus hijos menores de edad en un determinado momento o la ubicación de vehículos.

Otra aplicación comercial de los sistemas de rastreo a través de SBL, es la localización de un dispositivo en caso de robo o pérdida. Uno de los ejemplos destacables es el sistema de anti-robo y perdida *Prey*. El usuario puede instalar la aplicación en sus dispositivos móviles y una vez de haber declarado su dispositivo como “perdido” (a través de su cuenta de usuario en la plataforma web) este

puede activar a distancia las capacidades de geolocalización y así obtener reportes de la ubicación del dispositivo perdido. [<https://preyproject.com/es/>]

La siguiente figura muestra la captura de la plataforma web del sistema Prey, se puede visualizar la posición geográfica de un dispositivo reportado como perdido.

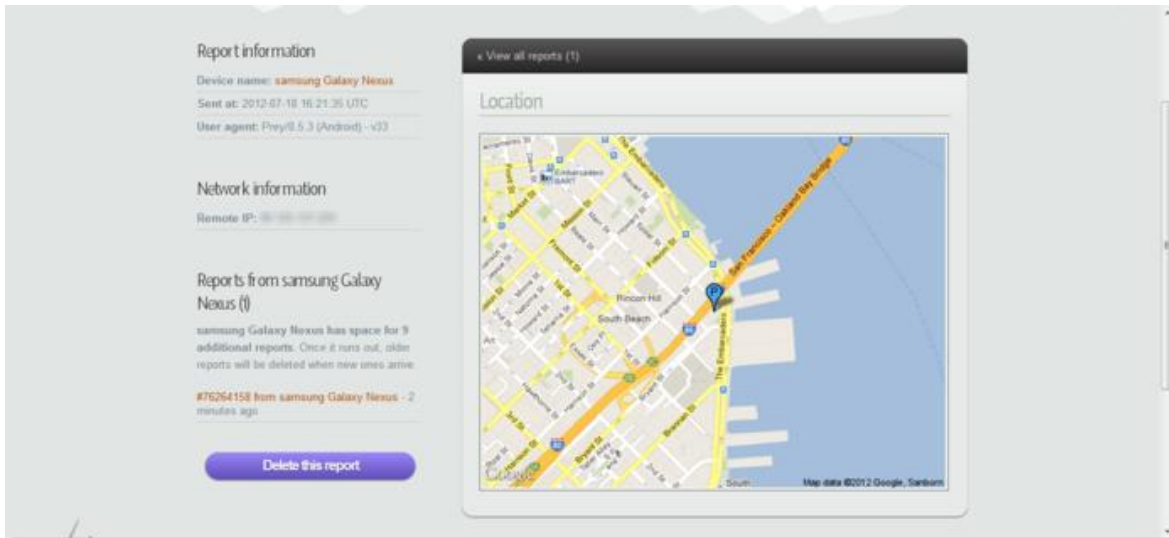


Figura 6: Captura del sistema Prey mostrando la ubicación de un dispositivo móvil reportado como perdido

CAPITULO 2: Framework PhoneGap

Actualmente el desarrollo de aplicaciones móviles basadas en tecnologías web, han facilitado la creación de aplicaciones multiplataforma. El uso de tecnologías ampliamente conocidas en el desarrollo web, como HTML5, JavaScript y CSS3, más la capacidad de los nuevos sistemas operativos móviles de ejecutar aplicaciones desarrolladas con estas tecnologías, ha traído como consecuencia que los desarrolladores pueden trasladar sus conocimientos en programación web al mundo de la programación móvil. En este capítulo se describirá el Framework de desarrollo de aplicaciones móviles PhoneGap. Concentrándonos en su definición, antecedentes y características.

En el capítulo 2.1 se plantea la definición y los antecedentes de PhoneGap. Seguidamente en el capítulo 2.2 se presentan las características de PhoneGap como framework de desarrollo de aplicaciones móviles.

2.1 Definición y Antecedentes

PhoneGap es un *framework* para el desarrollo de aplicaciones móviles usando estándares basados en tecnologías web. Los cuales hacen de puente entre aplicaciones web y dispositivos móviles. En su página oficial [<http://http://phonegap.com>] se declaran así mismos como “una implementación de software de estándares abiertos. Lo que significa que desarrolladores y compañías pueden usar el framework PhoneGap para aplicaciones móviles que son gratis, comerciales, de código abierto o cualquier combinación de estas”. PhoneGap soporta una gran variedad de plataformas móviles, entre las que se incluye: Android, IOS, BlackBerry, WebOS, Symbian, entre otras. Sin embargo, no todos los sistemas operativos tienen soporte a todas las capacidades nativas del dispositivo.

PhoneGap en si es una distribución de Apache Cordova ; Un conjunto de APIs que permiten al desarrollador de la aplicación móvil acceder a funciones nativas del dispositivo, tales como la cámara, acelerómetro, almacenamiento interno, entre otras. Todo esto usando código JavaScript, lo cual combinado con frameworks de

interfaz gráfica como jQuery Mobile permiten al desarrollador crear aplicaciones móviles basadas en tecnologías web. En si se puede ver la librería Apache Cordova como el motor que soporta a PhoneGap, tal como los webkits son los motores que soportan a los navegadores web.

A continuación se muestra una tabla extraída de la página oficial de PhoneGap, que muestra el nivel de soporte que tiene el *framework* en las distintas plataformas móviles que soporta. Como se puede apreciar, las plataformas más populares de Android y iOS cuentan con completo soporte de las capacidades del dispositivo móvil que se tiene acceso a través del framework.

Capacidades	iOS, iPhone/ iPhone3G	iOS iPhone 3GS en adelante	Android	Blackberry 5.0	Blackberry 6.0	HP WebOS	Windows Phone 7	Symbian	Bada
Acelerómetro	Si	Si	Si	Si	Si	Si	Si	Si	Si
Cámara	Si	Si	Si	Si	Si	Si	Si	Si	Si
Brújula	No	Si	Si	No	No	No	Si	No	Si
Contactos	Si	Si	Si	Si	Si	No	Si	Si	Si
Archivo	Si	Si	Si	Si	Si	No	Si	No	No
Geolocalización	Si	Si	Si	Si	Si	Si	Si	Si	Si
Multimedia	Si	Si	Si	No	No	No	Si	No	No
Red	Si	Si	Si	Si	Si	Si	Si	Si	Si
Notificación Alerta	Si	Si	Si	Si	Si	Si	Si	Si	Si
Notificación Sonido	Si	Si	Si	Si	Si	Si	Si	Si	Si
Notificación Vibración	Si	Si	Si	Si	Si	Si	Si	Si	Si
Almacenaje Local	Si	Si	Si	Si	Si	Si	Si	Si	No

Tabla 7: Características soportadas por PhoneGap en distintos SO móviles [U-PGAPW]

La compañía Nitobi fue el creador original de PhoneGap y uno de sus principales colaboradores desde octubre de 2011. Con la adquisición de Nitobi por parte de Adobe se posibilitó al equipo desarrollador del *framework*, poder concentrarse puramente en el proyecto PhoneGap y continuar su trabajo de desarrollo eficiente multiplataforma.

PhoneGap cuenta con diferentes colaboradores de varias compañías. Entre las que se incluye IBM, Microsoft, RIM entre otras. Además posee una gran comunidad activa, colaborativa y abierta que puede ser consultada en la web. [<http://phonegap.com/community/>]

2.2 Características de PhoneGap

En los siguientes subcapítulos se describirá las capacidades de la API de Cordova/PhoneGap, dando especial enfoque a cómo se maneja la geolocalización usando este framework. Las referencias de esta sección pueden encontrarse en [PG-DOC].

2.2.1 Descripción del API

El API de Cordova/PhoneGap cuenta con una variedad de elementos que hace posible el acceso a las capacidades nativas del dispositivo a través de código JavaScript. Estos elementos se describen a continuación:

- **Acelerómetro:** Captura el movimiento del dispositivo en los ejes x, y y z.
- **Cámara:** Provee acceso a la aplicación de cámara (por defecto) del dispositivo.
- **Captura:** Provee acceso a las capacidades de captura de video y audio del dispositivo.
- **Brújula:** Obtiene la dirección a donde está apuntando el dispositivo.
- **Conexión:** Obtiene información del tipo de conexión a la que está conectado el dispositivo (3G/Wifi).
- **Contactos:** Permite realizar acciones en la base de datos de contactos del dispositivo.
- **Dispositivo:** Permite obtener información específica del software y hardware del dispositivo.
- **Eventos:** Permite manejar los distintos eventos que pueden ocurrir en el dispositivo. Por ejemplo: batería baja.
- **Archivo:** Permite leer, escribir o navegar sobre el sistema de archivos del dispositivo.
- **Geolocalización:** Provee acceso a la ubicación geográfica actual del dispositivo (véase capítulo 2.2.4).
- **Globalización:** Permite obtener y realizar operaciones específicas basadas en la región y lenguaje del dispositivo.

- **Navegador:** Permite abrir una URL web específica dentro de la aplicación móvil.
- **Multimedia:** Permite reproducir o grabar archivos de audio en el dispositivo.
- **Notificación:** permite desplegar notificaciones audibles y táctiles (vibración) en el dispositivo.
- **Almacenamiento:** Permite acceder a las capacidades de almacenamiento local (no confundir con sistema de archivos) de la aplicación móvil. Como por ejemplo una base de datos HTML5.

2.2.2 Implementación en Distintas plataformas

Para poder desarrollar efectivamente en PhoneGap se necesita integrar el framework con el respectivo SDK (kit de desarrollo de software) de la plataforma móvil. Por lo que es necesario instalar el respectivo SDK en el ambiente de desarrollo. Por supuesto, la forma de hacer esto varía de plataforma en plataforma. A continuación se explicara de forma breve la implementación de PhoneGap en plataformas Android y BlackBerry.

Android: Al crear una aplicación PhoneGap en Android, lo que se crea es una aplicación de Android normal. Con la notoriedad de que esta aplicación solo muestra una ventana web apuntando al archivo web de inicio en la aplicación (por lo general index.html). Los archivos web, así como la librería Cordova deben encontrarse todos en la carpeta `assets` del proyecto Android creado. En la siguiente imagen se puede apreciar cómo se carga el URL de inicio de la aplicación en la línea 11 del código. El archivo index.html debe contar con todas las referencias a la librería Cordova JavaScript, para que la aplicación PhoneGap pueda funcionar, además se debe enlazar el proyecto Android con la respectiva librería Java de Cordova, que es un archivo de tipo `.jar`.

```
1 package com.ucv.georef;
2
3 import android.os.Bundle;
4
5
6 public class MainActivity extends DroidGap {
7
8     @Override
9     public void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         super.loadUrl("file:///android_asset/www/index.html");
12     }
13
14 }
```

Figura 8: Modo de implementar PhoneGap en Android

BlackBerry: El proceso para implementar PhoneGap en BlackBerry es algo diferente al de Android. Primeramente no se utiliza el SDK tradicional para crear aplicaciones nativas. Sino que se utiliza un SDK especial llamado HTML5 WebWorks. Este es un kit de desarrollo especial para crear aplicaciones BlackBerry basadas en tecnologías web. Para poder usar este SDK es necesario tener instalado la herramienta de líneas de comando Apache Ant. Lo que básicamente se hace es copiar la carpeta base BlackBerry del paquete PhoneGap descargado, en una ubicación específica de nuestro computador, se modifica el archivo *project.properties* del directorio para que este apunte a las ubicación del SDK HTML5 Webworks, una vez hecho esto se procede a construir el proyecto PhoneGap utilizando el comando *build* de Apache Ant. Los archivos web deben de estar en el directorio *www* del proyecto BlackBerry creado. Al igual que en Android, la aplicación iniciara en el archivo *index.html*. Este archivo debe contar también con todas las referencias a la librería Cordova de JavaScript.

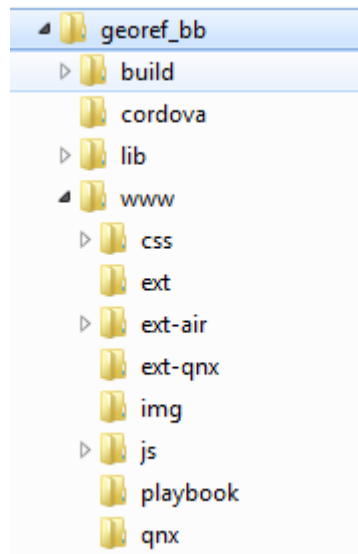


Figura 9: Jerarquía de directorios de un Proyecto BlackBerry usando PhoneGap

2.2.3 Servicio PhoneGap Build

Uno de los inconvenientes que posee PhoneGap como framework de desarrollo multiplataforma, es la necesidad de seguir dependiendo de los SDK de cada plataforma, para poder probar y generar las aplicaciones. Esto sin duda genera problemas, ya que es necesario crear copias del mismo proyecto PhoneGap en cada una de las plataformas a desarrollar. Por ejemplo; si fuera necesario crear una aplicación para Android y BlackBerry, se tendría que crear un proyecto Android y otro proyecto BlackBerry diferente que contenga la misma copia (los archivos web) del proyecto PhoneGap a desarrollar.

Debido a lo anterior, la empresa Adobe ha creado un servicio llamado PhoneGap Build. Con este servicio el usuario solo tiene que ocuparse por crear los archivos web necesarios de su aplicación (HTML5, CSS y JavaScript) y probarlos en un simulador como Ripple⁸. Para luego subirlos en el servicio PhoneGap Build, ya sea a través de un repositorio GitHub o un archivo comprimido tipo .zip. El servicio PhoneGap Build se encargara de generar los archivos ejecutables de cada plataforma. Incluso se pueden generar aplicaciones firmadas si se tiene las respectivas firmas para hacerlo.

⁸ Ripple emulator: es un simulador que permite probar aplicaciones y sitios web móviles a través del navegador Chrome usando una extensión.

En la figura 10 se muestra una captura de la ficha de una aplicación subida al servicio PhoneGap Build. Los cuadros azules que se notan en la parte inferior, son los enlaces de descarga de la aplicación para la respectiva plataforma móvil. En el caso de la plataforma iOS no se generó ninguna aplicación, ya que obligatoriamente se necesita una firma iOS certificada de Apple para hacerlo.

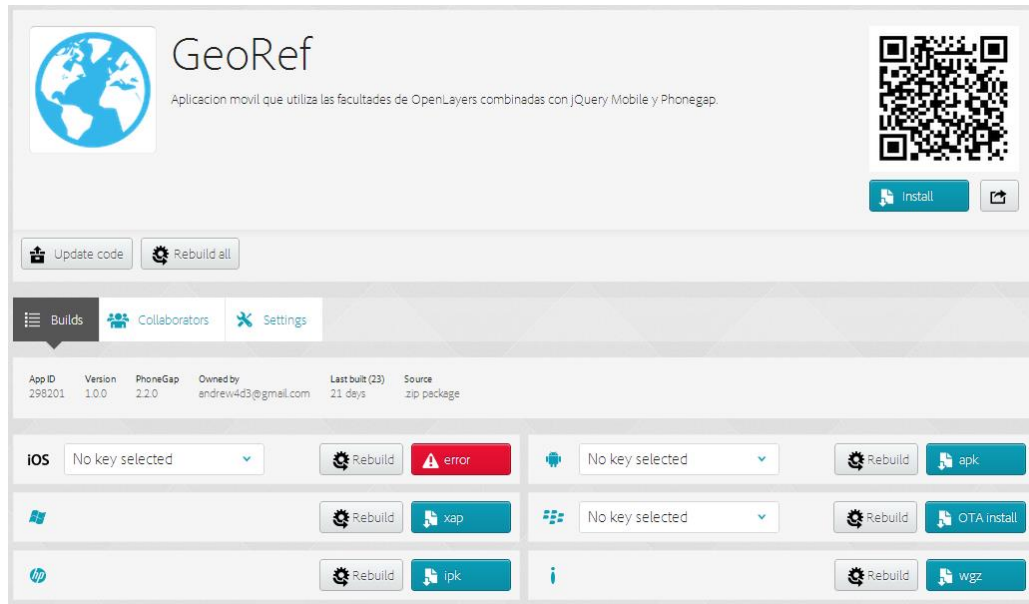


Figura 10: Ficha de aplicación móvil subida a PhoneGap Build

Para la correcta generación del archivo ejecutable en PhoneGap Build, es necesaria la correcta configuración de un archivo XML llamado config.xml. El cual debe estar dentro de la carpeta donde se encuentran los demás archivos web de la aplicación. En este archivo de configuración debe ir entre otras cosas: el nombre de la aplicación, la versión, la descripción, la ruta de acceso de la imagen del icono, distintos permisos y otros elementos opcionales o necesarios para la generación de la aplicación móvil.

2.2.4 Manejo de Geolocalización

Debido a la importancia que tiene la geolocalización en este trabajo, resulta conveniente describir el manejo de esta capacidad en un subcapítulo aparte.

La geolocalización en PhoneGap está basada en la especificación de la W3C *Geolocation* [W3CGC]. Usando el API de esta especificación el dispositivo puede

obtener atributos de posición geográfica tales como los valores de longitud y latitud. El dispositivo puede definir estos atributos a partir de algún método de geolocalización como GPS, Wifi o red celular. Debido a que no todos los métodos de geolocalización son igualmente precisos, no está garantizado que los datos geográficos que se obtengan sean asertivos.

Usando PhoneGap la geolocalización del dispositivo se maneja a través del objeto `navigator.geolocation`. Este objeto maneja dos métodos principales para obtener la actual ubicación geográfica. Los cuales se explican a continuación:

`getCurrentPosition`: Con este método se llama a dos funciones *callback*. Una en caso que se haya obtenido la posición geográfica con éxito y otra en caso que haya ocurrido un error. Si se pudo obtener la ubicación geográfica, la función *callback* recibe un objeto “posición” como parámetro. El cual tiene como atributos los valores de latitud y longitud de la actual ubicación geográfica. También el objeto “posición” tiene otros atributos tales como la precisión de la ubicación, la altura, la velocidad actual, y otros valores relacionados a la posición geográfica del dispositivo.

`watchPosition`: Este método es bastante similar al anterior. Posee dos funciones *callback* como parámetro (una en caso de éxito y otra en caso de error) y la función *callback* en caso de éxito recibe el mismo objeto “posición” del método anterior. La diferencia radica en que este método se mantendrá ejecutando ininterrumpidamente a menos que se indique su parada usando un método llamado `clearWatch`. Cada vez que el dispositivo cambie su ubicación geográfica, se ejecutará la función *callback* de éxito. Este método es especialmente útil para aplicaciones móviles que necesitan mantener un constante monitoreo de la ubicación geográfica del dispositivo.

CAPITULO 3: Librería OpenLayers

En este capítulo se abordara la librería web para el despliegue de mapas e información geográfica conocida como OpenLayers. Haciendo enfoque en lo que respecta al manejo de capas de mapas y objetos geográficos.

En el capítulo 3.1 se plantea la definición y los antecedentes de la librería, el capítulo 3.2 se enfoca en el manejo de capas geográficas usando distintos servicios de mapas y para finalizar el capítulo 3.3 se describe el manejo de objetos geográficos.

3.1 Definición y Antecedentes

OpenLayers [<http://openlayers.org/>] es un conjunto de librerías JavaScript que se utilizan para la inclusión de mapas dentro de entornos web. Actualmente es un proyecto de la Fundación Geoespacial de Código Abierto (Open Source Geospatial Foundation OsGeo) [<http://www.osgeo.org/>] que cuenta con el apoyo de un gran número de organizaciones alrededor del mundo, según hace constar la página oficial del proyecto. Es completamente gratis, *open source* (lanzado bajo la licencia FreeBSD⁹) y cuenta con una documentación extensa y bastante detallada. En la figura 11 se muestra un mapa sencillo desplegado con OpenLayers.



Figura 11: Mapa Sencillo de OpenLayers

⁹ Licencia FreeBSD: licencia que especifica distintas cláusulas relacionadas al sistema operativo Unix Berkeley.

OpenLayers cuenta con un API que permite acceder a las funciones que posee la librería de forma nativa [<http://trac.osgeo.org/openlayers/wiki/Documentation>]. OpenLayers no depende exclusivamente de una herramienta SIG o servicio de mapas específico, por lo que permite usar distintas fuentes de mapas tales como Google Maps, OpenStreetMap, Bing Maps y otros tantos. También se puede usar conjuntamente con un servidor SIG como MapServer, aunque se recomienda que su uso se haga a través de WMS¹⁰.

Otra de sus virtudes es la de poder incluir y manipular distintos servicios o fuentes de mapas de forma bastante ágil. Por lo que se pueden desarrollar interfaces de múltiples mapas, donde rápidamente se puede intercambiar de uno a otro.

La principal ventaja que tiene OpenLayers frente a otras opciones tales como la API de Google Maps, es que para usarlo en aplicaciones abiertas o comerciales no es necesario pagar por ninguna licencia.

3.2 Manejo de Capas Geográficas

Las capas en OpenLayers son la fuente de datos del mapa. Básicamente en OpenLayers puede existir dos tipos de capas; Una capa base la cual es la principal y que por lo general aloja el mapa base de la aplicación, y una capa no-base, que puede ser una capa secundaria que contenga información ante-puesta sobre la capa base. OpenLayers ofrece diversos controles que permiten manipular la visualización de capas base y no-base de forma sencilla.

A continuación en este subcapítulo se abordará el manejo de distintas capas en OpenLayers a través de distintos servicios de mapas web.

3.2.1 Capas de Google Maps

El servicio de mapas de Google [<http://maps.google.com/>] es sin duda alguna uno de los servicios más populares que tiene esta compañía. Es un servidor de aplicaciones de mapas en la Web que ofrece imágenes de mapas desplazables, fotos satelitales del mundo e incluso la ruta entre diferentes ubicaciones.

¹⁰ WMS: Es un protocolo estándar útil para la generación de mapas utilizando para ello una base de datos SIG.

Usando OpenLayers es posible incluir los mapas del servicio Google Maps dentro de nuestra aplicación, usando para ello el objeto `OpenLayers.Layer.Google`. Una instancia de esta clase contiene todo los métodos necesarios para la manipulación del mapa y sus propiedades. En la figura 12 se muestra un mapa OpenLayers con capa de GoogleMaps.



Figura 12: Mapa OpenLayers usando capa Google Maps

En general, es posible utilizar cuatro tipos de mapas con capas Google en OpenLayers. Estos son explicados a continuación:

Mapa Urbano: Este mapa solo contiene información de tipo urbano, relacionada a calles, vías, áreas administrativas, líneas de transporte, etc. No posee información de tipo satelital o física. Debido a que es el tipo de mapa por defecto, no es necesario especificar alguna propiedad para su generación.

Mapa Satelital: Este mapa contiene imágenes satelitales, las cuales son cargadas a medida que nos acercamos a un área específica. No posee ninguna etiqueta de identificación o información urbana. Para utilizar este mapa es necesario especificar el valor `google.maps.MapTypeId.SATELLITE` dentro de la propiedad `type`.

Mapa Híbrido: Este mapa es una combinación del mapa urbano y del mapa satelital visto anteriormente. Simplemente muestra la información urbana ante puesta sobre las imágenes satelitales. Lo cual resulta muy útil a la hora de

identificar visualmente referencias posicionales. Para generar este mapa es necesario especificar el valor `google.maps.MapTypeId.HYBRID` dentro de la propiedad `type`.

Mapa Físico: Este mapa muestra las características físicas del terreno, como puede ser el relieve y la concentración de vegetación. También muestra etiquetas de identificación de distintos lugares y vías de comunicación, pero no al nivel de detalle del mapa urbano. Para generar este mapa es necesario especificar el valor `google.maps.MapTypeId.TERRAIN` dentro de la propiedad `type`.

3.2.2 Capas de Bing Maps

Bing Maps es otro servicio de mapas web, el cual fue desarrollado por Microsoft para competir contra el servicio Google Maps. Ofrece muchas similitudes con su competidor de Google, aunque con diferencias en la información que despliega. En la figura 13 se muestra un mapa OpenLayers con capa de Bing Maps.



Figura 13: Mapa OpenLayers usando capa Bing Maps

Con OpenLayers se pueden utilizar tres tipos de mapas Bing, los cuales se explican a continuación:

Mapa Urbano: Con este mapa se pueden visualizar vías de comunicación y etiquetas de ciudades y áreas administrativas. También se pueden visualizar características del relieve y la vegetación del terreno (dependiendo del nivel de

zoom del mapa). Para generar este mapa se debe especificar el valor "Road" en la propiedad `type`.

Mapa Aéreo: Este mapa es equivalente al mapa satelital de Google Maps. Sin embargo el nivel de detalles de las imágenes satelitales puede variar dependiendo del área que se esté observando. Para generar este mapa es necesario especificar el valor "Aerial" en la propiedad `type`.

Mapa Híbrido: Este mapa es equivalente al mapa híbrido de Google Maps. Para generarlo es necesario especificar el valor "AerialWithLabels" en la propiedad `type`.

3.2.3 Capas de OpenStreetMap

OpenStreetMap es una iniciativa de crear y proveer información geográfica gratis (mapas) a cualquiera que lo necesite. La fundación OpenStreetMap es una organización sin fines de lucro que apoya (pero no controla) al proyecto OpenStreetMap (OSM). Esta organización está dedicada a promover el crecimiento, desarrollo y distribución de información gratis geoespacial para su uso o reparto. [OSMF]

Con OpenLayers podemos desplegar un capa especial que use la fuente de mapa estándar de OSM (para fuentes OSM secundarias es necesario realizar algunas configuraciones adicionales). Todo esto usando un objeto `OpenLayers.Layers.OSM`, el cual posee los atributos y métodos necesarios para manipular y/o configurar el mapa OSM.

La figura 14 se muestra un mapa OpenLayers usando capa de OpenStreetMap (OSM).



Figura 14: Mapa OpenLayers usando capa OpenStreetMap (OSM)

El mapa que se despliega usando OpenLayers en conjunto con OSM es un mapa de tipo urbano, que contiene información de vías de comunicación, delimitaciones de áreas administrativas, edificios o zonas de interés. Esta información está en constante cambio ya que los mapas no son creados por un ente específico, sino por una comunidad de usuarios que constantemente está actualizando la información geográfica. La ventaja de usar OSM sobre otras opciones propietarias como la de Google Maps o Bing Maps, es su uso libre de licencias comerciales.

3.3 Manejo de Objetos Geográficos

La forma de representar objetos en sistemas de información geográficos en general, es usando objetos geográficos. Estos objetos son puntos, líneas o polígonos que representan elementos geográficos en un área del mapa, por ejemplo: un punto pudiera representar una localidad específica como la ubicación de una estación de metro, mientras que un polígono puede representar un área que abarca varios metros o kilómetros, como un parque o área de construcción.

En OpenLayers la forma de crear y desplegar objetos geográficos en mapas es usando las clases `OpenLayers.Feature.Vector` y `OpenLayers.Geometry` conjuntamente.

A continuación se explica en qué consiste cada una:

OpenLayers.Geometry: Con esta clase se crea la geometría del objeto geográfico deseado. Es necesario llamar a una subclase que especifica el tipo de

geometría a crear. Por ejemplo: si se desea crear una geometría de tipo punto es necesario instanciar una clase de `OpenLayers.Geometry.Point`. La cual reciba como parámetro las coordenadas geográficas del punto a crear.

OpenLayers.Feature.Vector: Con esta clase creamos el objeto geográfico deseado. Para eso pasamos como parámetros la geometría creada usando la clase pasada y otros atributos adicionales que se deseen agregar; Como puede ser descripciones del objeto, opciones de estilo, entre otras.

La figura 15 muestra un mapa OpenLayers con objetos geográficos contruidos con las clases recientemente explicadas.

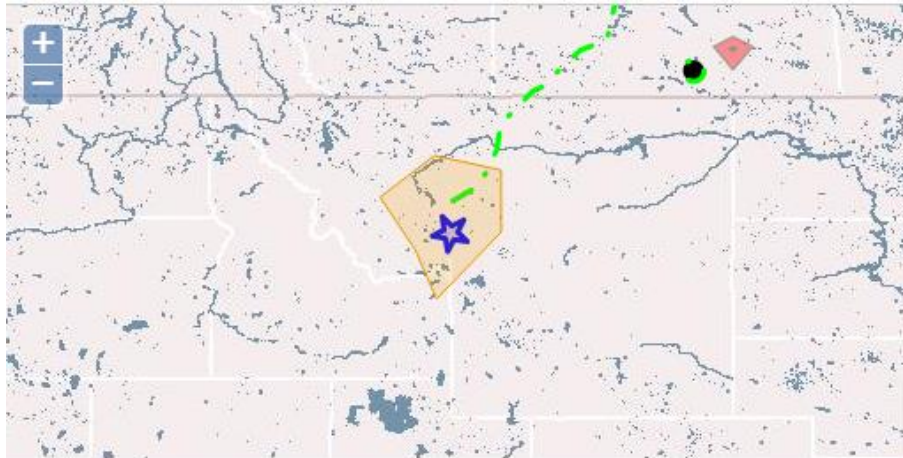


Figura 15: Mapa OpenLayers mostrando objetos geográficos

Una vez creado el objeto geográfico deseado, para incluirlo en el mapa se usa una capa especial (no-base) para alojar objetos geográficos. Esta capa es instanciada usando la clase `OpenLayers.Layer.Vector`. Para agregar los objetos geográficos dentro de la capa instanciada se utiliza el método `addFeatures`, el cual pasa como parámetro un arreglo de los objetos geográficos que queramos agregar en el mapa.

Capítulo 4: Análisis y diseño de la aplicación móvil

Para el análisis y diseño de la aplicación móvil se utilizaron las especificaciones del proceso unificado de desarrollo UP (acrónimo de *Unified Process*), debido a que el mismo provee un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software y diferentes áreas de aplicación. UP propone actividades y artefactos para las diferentes etapas del proceso de análisis y diseño.

Usando el proceso unificado, se desarrollaron actividades para el levantamiento de requerimientos y diseño de la solución. Se emplearon casos de uso para describir las funcionalidades de la aplicación móvil. Para describir el diseño de la solución se emplearon diagramas de clase, secuencia e implementación. Los diagramas empleados en este documento están basados en la especificación UML¹¹.

La sección 4.1 contiene los casos de uso y sus descripciones, la sección 4.2 contiene los diagramas de clases tanto para la aplicación móvil como para la aplicación en el servidor. La sección 4.3 contiene los diagramas de secuencia y la sección 4.4 se describe el diagrama de implementación del sistema en general (aplicación móvil y servidor).

4.1 Casos de Uso

Esta sección describe los actores y los casos de uso de la aplicación móvil.

4.1.1 Actores

A continuación se describe el actor que interactúa con las funcionalidades de la aplicación móvil desarrollada.

Usuario Móvil: Es el actor responsable de realizar las siguientes acciones dentro de la aplicación móvil:

- Crear un usuario que podrá iniciar sesión en la aplicación móvil.

¹¹ UML: acrónimo en inglés de *Unified Modelling Language*, lenguaje de modelado unificado.
http://www.omg.org/gettingstarted/what_is_uml.htm

- Iniciar sesión dentro la aplicación móvil con un usuario creado.
- Crear, consultar, editar o eliminar los sitios asociados al usuario en sesión.
- Activar la sincronización de la aplicación móvil con el servidor.
- Consultar y registrar visitas en los checkpoints descargados desde el servidor.
- Visualizar el mapa de la aplicación móvil.
- Configurar elementos de la aplicación móvil.

4.1.2 Nivel 1

El diagrama del nivel 1 presenta una visión funcional de alto nivel de la aplicación móvil, con el actor descrito que interactúa con la aplicación. Así como los casos de uso generales dentro de la frontera de la aplicación móvil, tal como se muestra en la figura 16.

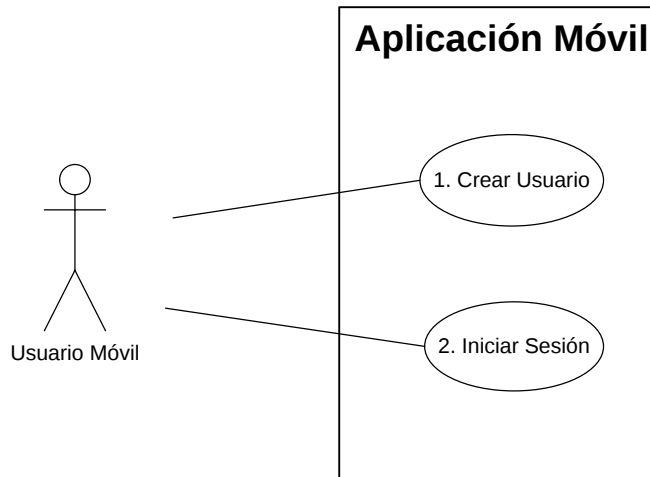


Figura 16: Visión general de la aplicación móvil

A continuación las descripciones de los casos de uso de este nivel.

Caso de Uso	CU #1 Crear Usuario.
Actor	Usuario Móvil.
Descripción	El actor ingresa a la aplicación móvil y crea una cuenta de usuario ingresando los datos correspondientes a la misma (nombre de

	usuario y contraseña).
Flujo Básico	1. El caso de uso inicia cuando el usuario móvil ingresa a la aplicación móvil instalada en el dispositivo y presiona el botón de “Crear Usuario”. 2. El actor ingresa los datos solicitados para crear la cuenta de usuario (nombre de usuario y contraseña) en los campos correspondientes y luego presiona el botón “Crear Usuario”. 3. La aplicación valida los datos ingresados y verifica si el nombre de usuario ingresado se encuentra disponible (no se encuentra aún creado). 4. Si los datos son correctos se crea la cuenta y se muestra al usuario móvil un mensaje de “Usuario creado satisfactoriamente”. Posteriormente se devuelve al actor a la pantalla de inicio de sesión para que este pueda ingresar con su cuenta.
Flujo Alternativo 1	1. En caso que el usuario ingrese algún dato incorrecto o que corresponda a un nombre de usuario no disponible (debido a que ha sido creado por otra cuenta). Se le muestra al usuario móvil un mensaje indicando el respectivo error.
Pre-condiciones	El usuario móvil debe haber ingresado satisfactoriamente a la aplicación y no haber activado la opción “Conectar automáticamente” en un acceso anterior.
Post-condiciones	Se crea la cuenta de usuario con los datos ingresados. Esta cuenta ya puede ser accedida a través del inicio de sesión.

Caso de Uso	CU #2 Iniciar Sesión.
Actor	Usuario Móvil.

Descripción	El actor ingresa a la aplicación móvil e inicia sesión ingresando su nombre de usuario y contraseña de una cuenta de usuario creada.
Flujo Básico	1. El caso de uso inicia cuando el usuario móvil ingresa a la aplicación móvil instalada en el dispositivo, teclea su nombre de usuario y contraseña en los respectivos campos y presiona el botón de “Iniciar Sesión”. 2. La aplicación verifica que los datos ingresados correspondan a los de una cuenta de usuario creada y muestra un mensaje de bienvenida indicando que el inicio de sesión ha sido exitoso. 3. Una vez iniciado sesión se muestran las distintas opciones que el usuario en sesión puede comenzar a hacer uso.
Flujo Alternativo 1	1. En caso que el usuario móvil ingrese un nombre de usuario y/o contraseña incorrecto (que no corresponda a ninguna cuenta de usuario creada) se muestra un mensaje de “Nombre de Usuario y/o contraseña incorrecto”.
Pre-condiciones	El usuario debe haber ingresado a la aplicación móvil.
Post-condiciones	Se muestra el menú principal de la aplicación móvil.

4.1.3 Nivel 2

El diagrama del nivel 2, se muestra en la figura 17, presenta el refinamiento de los casos de uso que tiene acceso el usuario móvil autenticado y con sesión activa.

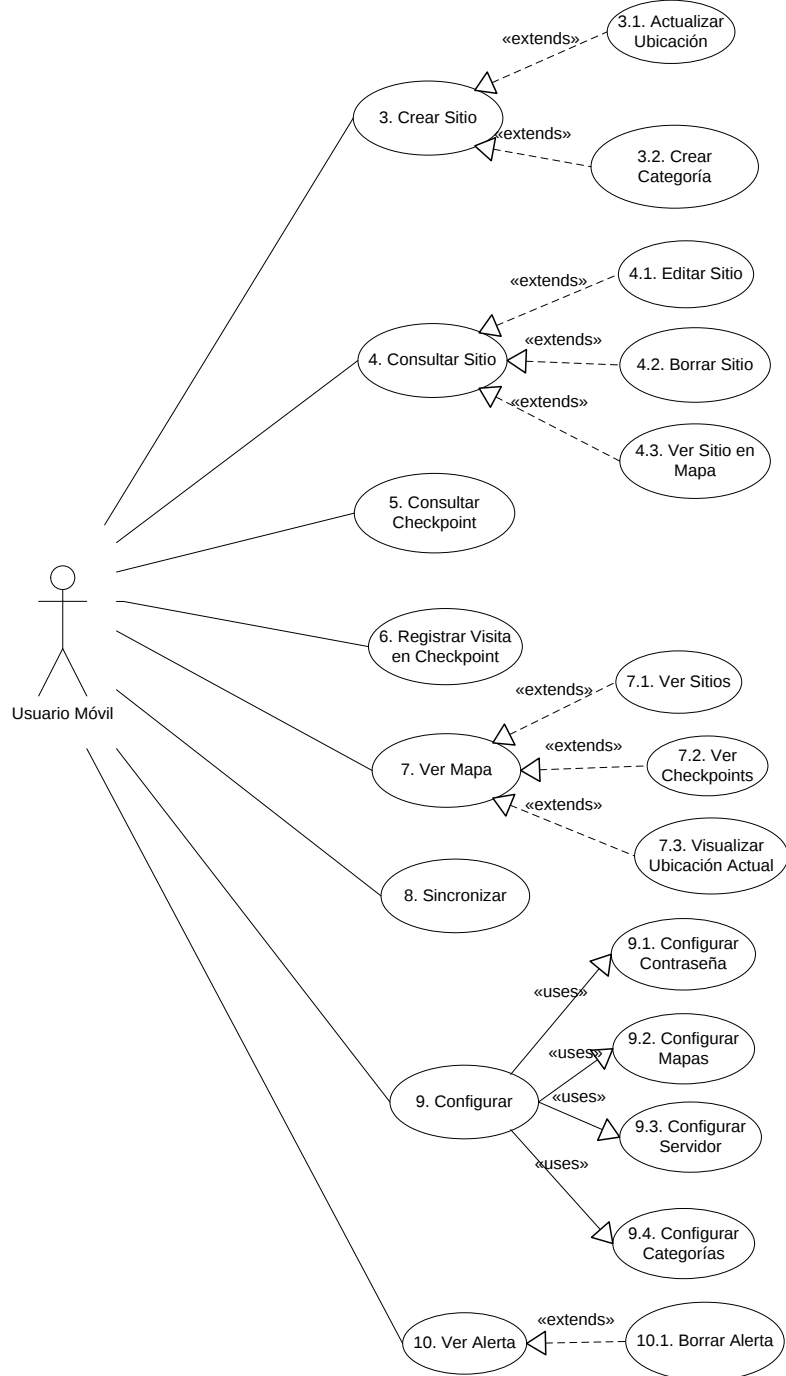


Figura 17: Casos de Uso del Nivel 2

A continuación se describen las especificaciones de los casos de uso más importantes de este nivel.

Caso de Uso	CU #3 Crear Sitio.
Actor	Usuario Móvil.
Descripción	Permite al actor crear un nuevo sitio en su cuenta.
Flujo Básico	1. El caso de uso inicia cuando el usuario ingresa en la opción “Nuevo Sitio” del menú principal. 2. La aplicación procede a obtener las coordenadas geográficas actuales y se las muestra al usuario móvil. 3. El usuario móvil procede a presionar el botón “Continuar” y así pasar a la pantalla donde puede ingresar los datos del sitio a crear: nombre del sitio, descripción (opcional) y categoría (opcional). 4. Una vez ingresado los datos, el usuario móvil podrá presionar el botón “Crear Sitio”, el cual crea un nuevo sitio en la cuenta del usuario.
Flujo Alternativo 1	1. En caso que el dispositivo por alguna razón no sea capaz de recibir las coordenadas geográficas actuales o unas coordenadas con poca precisión, un mensaje es desplegado al usuario describiendo el caso.
Flujo Alternativo 2	1. En caso que el usuario desee asignar al sitio creado una nueva categoría, este podrá crearla especificando la opción “Otra (Crear Categoría)” en el formulario. 2. La aplicación despliega un formulario donde el usuario podrá ingresar los datos de la nueva categoría en los respectivos campos. 3. Una vez ingresado los datos, el usuario podrá crear dicha categoría presionando el botón “Crear Categoría”. Esta acción crea una nueva categoría en la cuenta.

	4. Luego el usuario puede seleccionar en el formulario principal (el de creación de sitio) la categoría que acaba de crear u otra opción que sea de su preferencia.
Flujo Alternativo 3	1. En caso que el usuario móvil desee actualizar la ubicación geográfica actual, podrá hacerlo al presionar el botón “Actualizar Ubicación”. 2. La aplicación solicita nuevamente las coordenadas geográficas al dispositivo y son desplegadas en pantalla.
Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	Se crea un nuevo sitio en la cuenta del usuario en sesión.
Extiende	3.1 Actualizar Ubicación 3.2 Crear Categoría

Caso de Uso	CU #4 Consultar Sitio.
Actor	Usuario Móvil.
Descripción	Permite al actor consultar un sitio ya creado en su cuenta.
Flujo Básico	1. El caso de uso inicia cuando el usuario en sesión selecciona la opción “Sitios” del menú principal. 2. La aplicación despliega los sitios creados organizados por categoría. Los sitios sin categoría los despliega en un renglón llamado “Sin Categoría”. 3. El usuario móvil ingresa a alguno de los sitios creados al presionar sobre él. 4. Se muestra en detalle los datos del sitio en que se ha ingresado. Así como las opciones para editarlo, eliminarlo o

	visualizarlo en el mapa de la aplicación.
Flujo Alterno 1	1. En caso que el usuario no tenga ningún sitio creado en su cuenta se muestra una lista vacía.
Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	Se despliegan los datos del sitio consultado, así como las opciones para editarlo, eliminarlo y verlo en el mapa.
Extiende	4.1 Editar Sitio 4.2 Borrar Sitio 4.3 Ver sitio en mapa

Caso de Uso	CU #5 Consultar Checkpoint.
Actor	Usuario Móvil.
Descripción	El actor consulta un checkpoint ya descargado desde el servidor.
Flujo Básico	1. El caso de uso inicia cuando el usuario ingresa en la opción "Checkpoints" del menú principal. 2. La aplicación despliegue los checkpoints descargados en una lista. 3. El usuario móvil ingresa a alguno de los checkpoints descargados al presionar sobre él. 4. Se muestra en detalle los datos del checkpoint en que se ha ingresado.
Flujo Alterno 1	1. En caso de que el usuario no tenga ningún checkpoint descargado desde el servidor, se mostrara una lista vacía.

Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	Se despliegan los datos del sitio consultado, así como las opciones para editarlo, eliminarlo y verlo en el mapa.
Notas	• Al igual que en sitios el usuario móvil también podrá ver el checkpoint en el mapa principal de la aplicación. • El usuario móvil podrá registrar su visita en el checkpoint si se encuentra cercano a su posición.

Caso de Uso	CU #6 Registrar Visita en Checkpoint.
Actor	Usuario Móvil.
Descripción	El actor registra su visita en algún checkpoint cercano a su posición.
Flujo Básico	1. El caso de uso inicia cuando el usuario móvil ingresa en la opción “Hacer Check-in” del menú principal. 2. La aplicación despliegue los checkpoints cercanos a la posición del dispositivo móvil. 3. El usuario móvil ingresa a alguno de los checkpoints cercanos al presionar sobre él. 4. Se muestra en detalle los datos del checkpoint en que se ha ingresado. 5. El usuario móvil presiona el botón “Hacer Check-in” del checkpoint ingresado. Lo cual registra la visita en este determinado checkpoint.
Flujo Alternativo 1	1. En caso que el usuario no tenga ningún checkpoint cercano a su posición, se despliega un mensaje indicando esto.

Flujo Alternativo 2	1. En caso que el dispositivo por alguna razón no sea capaz de recibir las coordenadas geográficas actuales o unas coordenadas con poca precisión, se despliega un mensaje al usuario describiendo el problema.
Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	Se registra la visita en el determinado checkpoint.
Notas	Al igual que en sitios el usuario móvil también podrá ver el checkpoint cercano en el mapa principal de la aplicación.

Caso de Uso	CU #7 Ver Mapa.
Actor	Usuario Móvil.
Descripción	El actor puede visualizar y manipular el mapa principal de la aplicación.
Flujo Básico	1. El caso de uso inicia cuando el usuario ingresa en la opción “Mapa” del menú principal. 2. La aplicación despliegue el mapa principal. Ahora el usuario podrá hacer uso de las funcionalidades del mapa (hacer zoom, ver sitios, ver checkpoints y visualizar ubicación actual).
Pre-condiciones	• El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación. • El dispositivo móvil debe contar con acceso a internet, de lo contrario el mapa no se desplegará correctamente.
Post-condiciones	El mapa es desplegado en la aplicación con todas sus funcionalidades y opciones correspondientes.

Extendido	7.1 Ver sitios 7.2 Ver checkpoints 7.3 Visualizar ubicación actual
------------------	---

Caso de Uso	CU #8 Sincronizar.
Actor	Usuario Móvil.
Descripción	El actor sincroniza los datos de su cuenta en la aplicación móvil con los datos de su cuenta en el servidor.
Flujo Básico	1. El caso de uso inicia cuando el usuario presiona la opción “Sincronizar” del menú principal. 2. La aplicación se comunica con el servidor y actualiza todos los datos necesarios en él (sitios nuevos, sitios modificados, nuevos registros de visita, etc). A su vez si hay datos en el servidor que requieren descargarse (nuevos checkpoints y/o alertas) estos son descargados dentro de la aplicación. 3. Una vez que los datos han sido sincronizado se despliega un mensaje al usuario informando el éxito de la sincronización.
Flujo Alternativo 1	1. En caso que no haya comunicación con el servidor se despliega un mensaje notificando esto.
Flujo Alternativo 2	1. En caso que la autenticación en el servidor haya fallado (verificación del usuario y la contraseña) se despliega un mensaje informando el fallo de la autenticación.
Pre-condiciones	• El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación. • El dispositivo móvil debe contar con acceso a internet, de lo contrario no podrá existir comunicación con el servidor.

Post-condiciones	Los datos entre la aplicación móvil y el servidor se encontraran completamente sincronizados.
-------------------------	---

Caso de Uso	CU #9 Configurar.
Actor	Usuario Móvil.
Descripción	El actor puede configurar distintos elementos de la aplicación.
Flujo Básico	1. El caso de uso inicia cuando el usuario presiona la opción “Configurar” del menú principal. 2. La aplicación despliega todas las opciones que puede configurar el usuario móvil. 3. El usuario móvil puede ahora cambiar la configuración de la aplicación. Como por ejemplo cambiar la contraseña. 4. Una vez configurada la opción determinada, el usuario deberá confirmar los cambios al presionar un botón de “Guardar” o “Cambiar” de la respectiva opción. 5. Un mensaje es desplegado al usuario notificando que los cambios de configuración han sido realizados con éxito.
Flujo Alternativo 1	1. En caso que algunos de los datos de configuración ingresados no sea correcto, se despliega un mensaje al usuario móvil indicando el error de validación.
Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	Los cambios de configuración realizados quedan guardados.
Incluye	9.1 Configurar contraseña 9.2 Configurar mapas 9.3 Configurar servidor

	9.4 Configurar categorías
--	---------------------------

Caso de Uso	CU #10 Ver Alerta.
Actor	Usuario Móvil.
Descripción	El actor puede ver una alerta (mensaje) recibida desde el servidor.
Flujo Básico	1. El caso de uso inicia cuando el usuario presiona la opción “Alertas” del menú principal. 2. La aplicación despliega todas las alertas descargadas desde el servidor. 3. El usuario móvil puede ahora abrir una determinada alerta y visualizar el mensaje. 4. Una vez visualizada la alerta el usuario podrá borrarla o ver otra alerta disponible.
Flujo Alternativo 1	2. En caso de que no haya ninguna alertas se despliega el mensaje: “Tiene 0 alertas”.
Pre-condiciones	El usuario móvil debe haber iniciado satisfactoriamente sesión dentro de la aplicación.
Post-condiciones	La alerta visualizada se marca como “vista” dentro de la cuenta del usuario.
Extendido	10.1 Borrar alerta

4.2 Diagramas de Clases

Esta sección contiene los diagramas de clase para la aplicación móvil y el servidor.

La figura 18 muestra el diagrama de clases de la aplicación móvil.

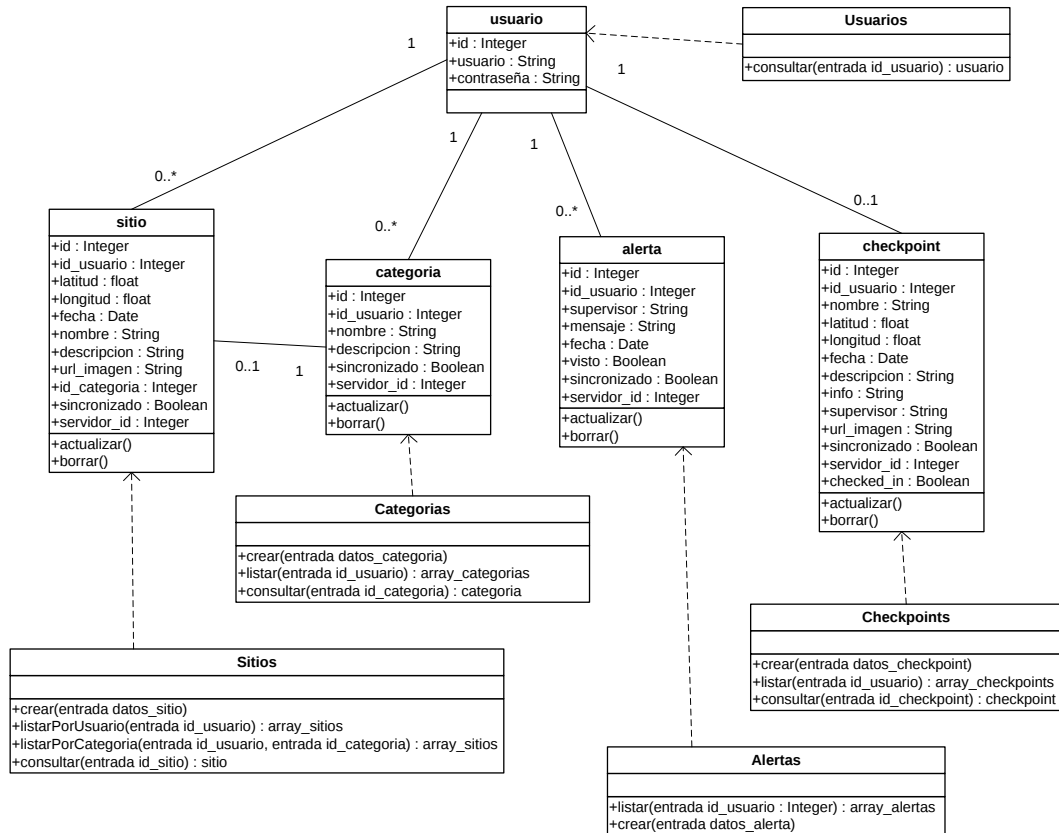


Figura 18: Diagrama de Clases de la aplicación móvil

La figura 19 muestra el diagrama de clases de la aplicación en el servidor.

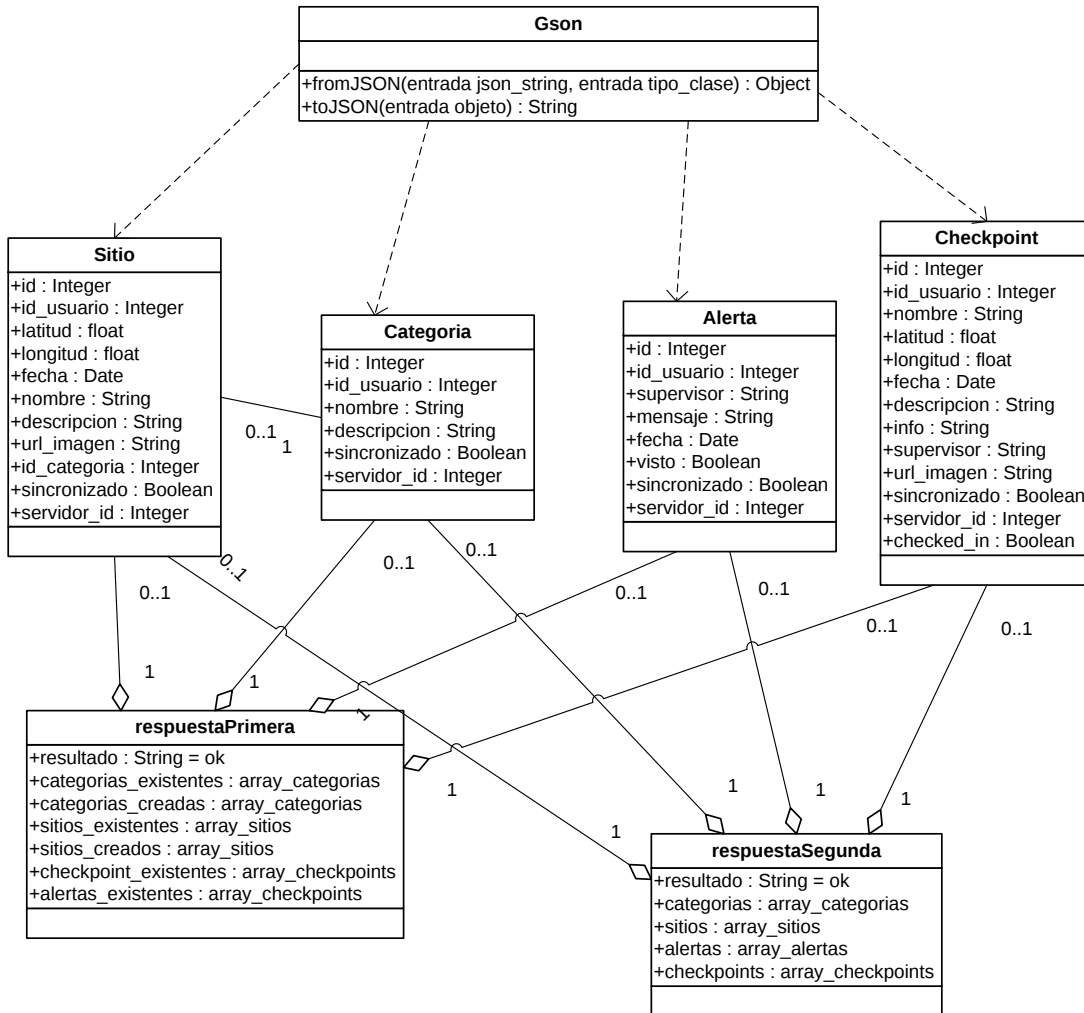


Figura 19: Diagrama de Clases del servidor

4.3 Diagramas de Secuencia

Esta sección contiene los diagramas de secuencia de las interacciones más importantes de la aplicación móvil y el servidor.

La figura 20 muestra el diagrama de secuencia de la creación de un nuevo sitio.

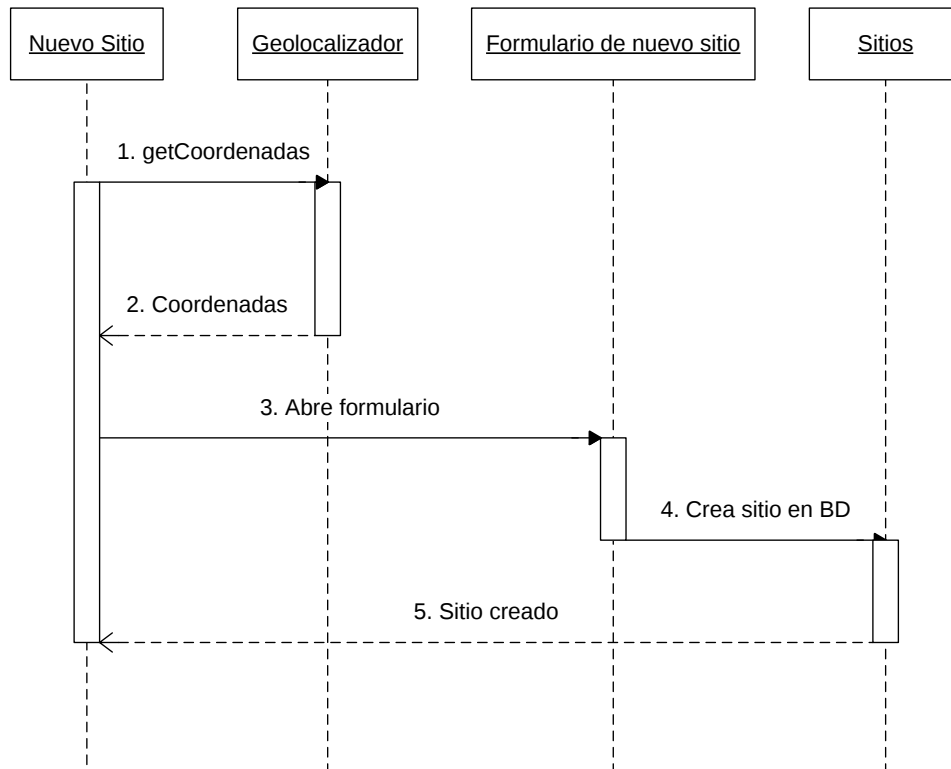


Figura 20: Diagrama de secuencia de creación de Nuevo Sitio

La figura 21 muestra el diagrama de secuencia de visita en checkpoint.

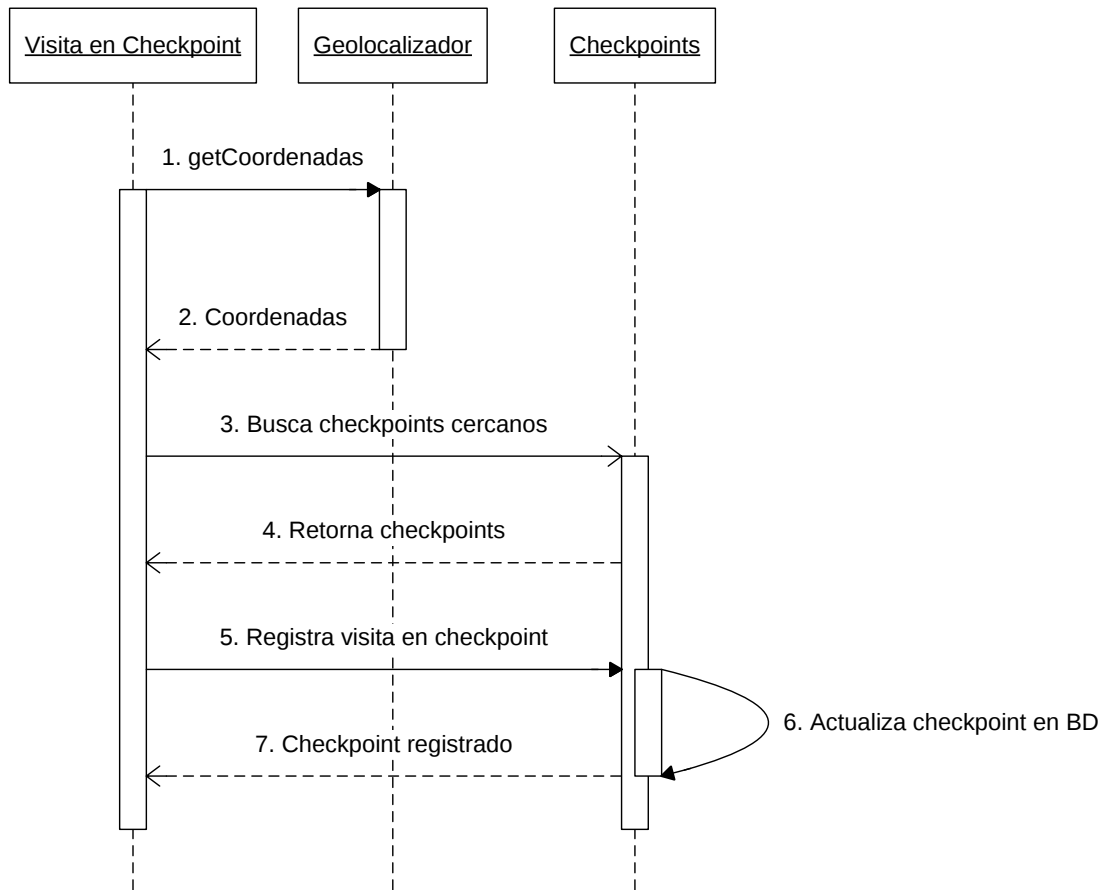


Figura 21: Diagrama de secuencia de Visita en Checkpoint

La figura 22 muestra el diagrama de secuencia de Sincronización.

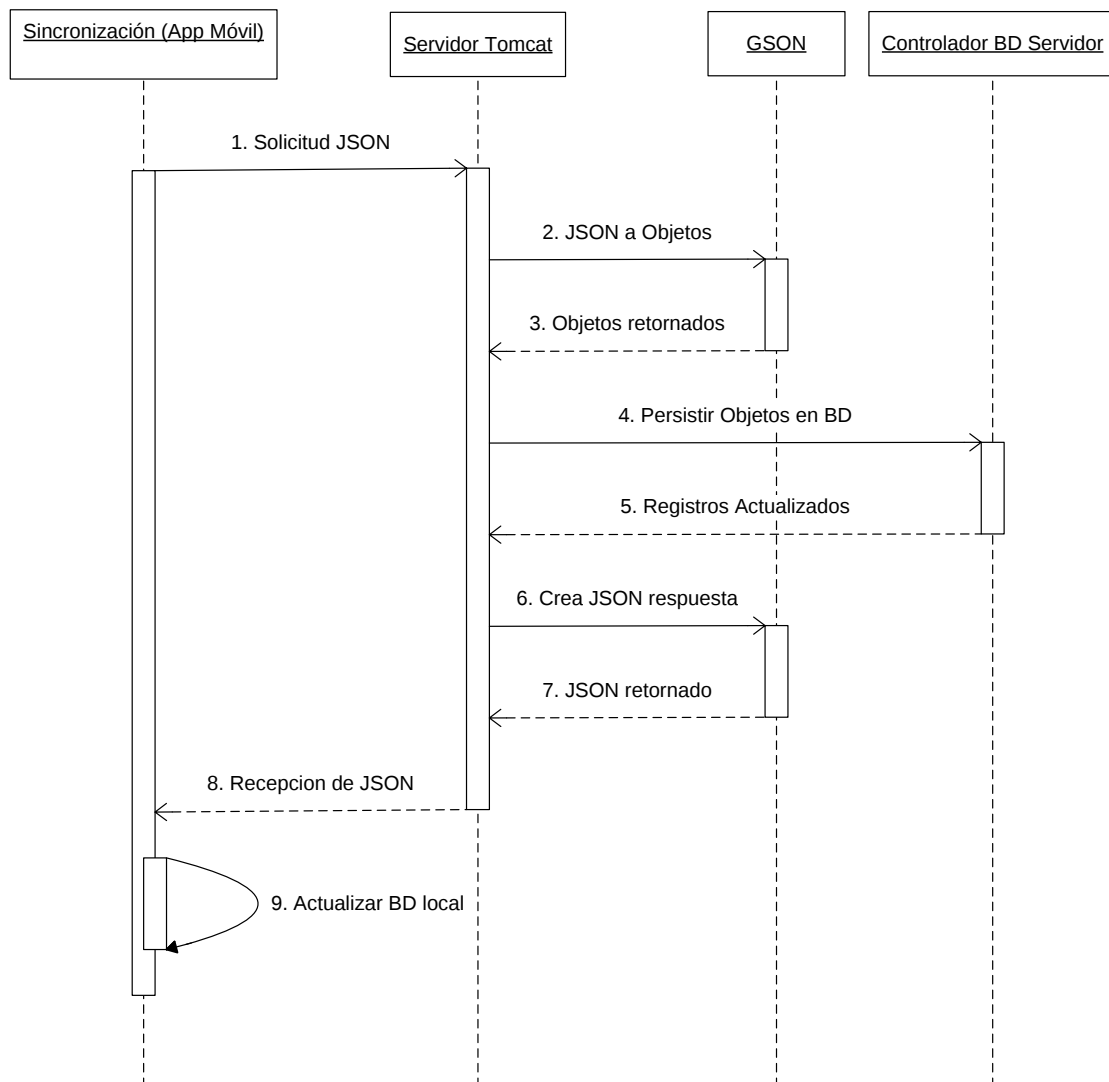


Figura 22: Diagrama de secuencia de Sincronización

4.4 Diagrama de Implementación

Esta sección contiene el diagrama de implementación del sistema en conjunto. La figura 23 muestra los nodos del dispositivo móvil y el servidor, los componentes que contienen y sus relaciones entre sí. Se muestra como la aplicación móvil depende del componente de geolocalización del dispositivo móvil el cual puede obtener las coordenadas geográficas usando los tres métodos ya explicados (véase capítulo 1.2). Mientras que en el servidor se encuentra un componente de software, que en entre otros expone (a través de un servicio web RESTful¹²) obtiene de la aplicación móvil las peticiones para realizar las operaciones de persistencia en la Base de datos geoespacial.

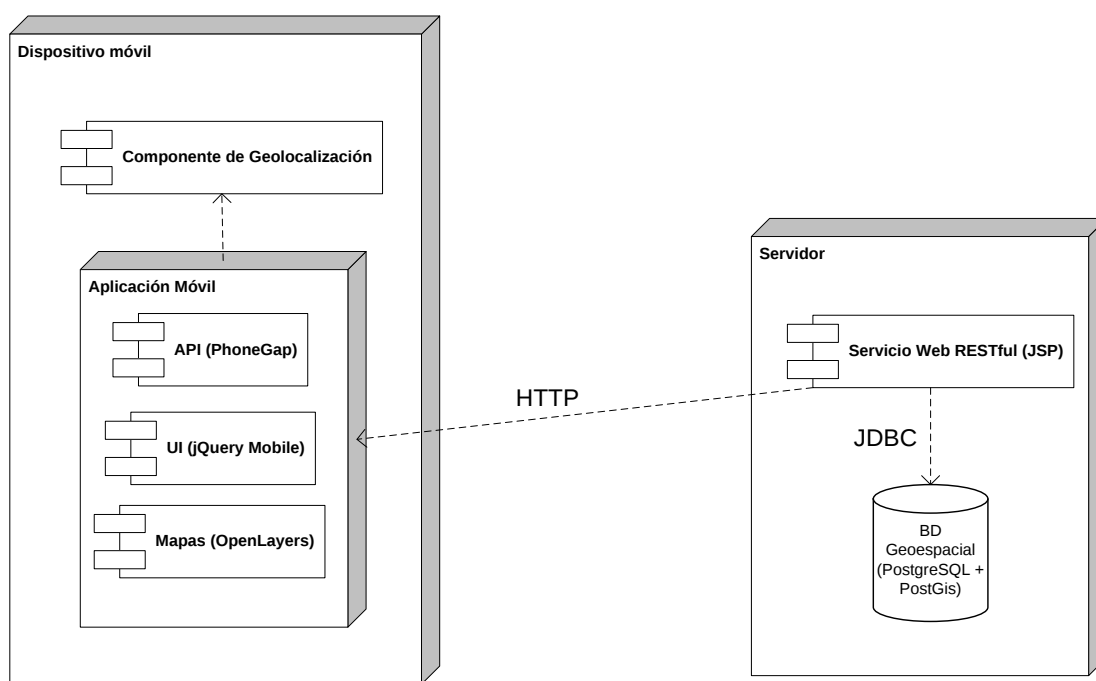


Figura 23: Diagrama de implementación

¹² RESTful: cualquier interfaz de servicio web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes.

CAPITULO 5: Implementación

En este capítulo se describirá la implementación de la solución, así como las tecnologías usadas para cumplir exitosamente los objetivos de dicha implementación.

En la sección 5.1 se describe como fue la implementación de JQuery Mobile, en la sección 5.2 como fue la implementación de JSON, en la sección 5.3 se describe como se implementaron las funcionalidades de la pantalla inicial de la aplicación móvil, en la sección 5.4 como se implementaron las opciones del menú principal y en la sección 5.5 se describe como fue la implementación de la base de datos geoespacial.

5.1 JQuery Mobile

JQuery Mobile [<http://jquerymobile.com/>] es un framework de desarrollo que funciona sobre las plataformas móviles más populares y que se encuentra basado en los mismos métodos y elementos de interfaz que introduce JQuery y JQuery UI respectivamente. Todo esto para dar apoyo en la creación de ricas interfaces de usuario construidas usando poco código. Para lograr este acercamiento, todas las páginas en JQuery Mobile están construidas con un limpio y semántico código HTML, que asegura la compatibilidad con cualquier dispositivo móvil capacitado para entender código web.

Jquery Mobile fue utilizado para construir la interfaz de usuario de la aplicación móvil del presente trabajo especial de grado.

En la figura 24 se aprecia una muestra de código HTML (de la aplicación desarrollada) soportado por JQuery Mobile.

```

253     <div data-role="page" id="sitios">
254         <div data-role="header">
255             <a href="#home" class="regresar" data-icon="back">Regresar</a>
256             <h1>Sitios</h1>
257         </div>
258
259         <div id="lista-sitios" data-role="collapsible-set">
260
261
262         </div>
263
264     </div>

```

Figura 24: Código HTML de la aplicación usando JQuery Mobile

En la línea 253 del código mostrado en la figura, el atributo `data-role` del contenedor `div` contiene el valor `"page"` el cual le indica al motor del framework que debe construir el contenido de este contenedor con formato de página simple. En la subsiguiente línea el atributo `data-role` contiene el valor `"header"` el cual le indica al motor de framework que debe formatear este contenedor como una cabecera (lo que sería la cabecera de la página). En la línea 259 el atributo `data-role` contiene el valor `"collapsible-set"`, lo cual le indica al motor del framework que el contenido debe ser formateado como una lista de elementos colapsables. Este contenedor se encuentra vacío ya que el contenido es generado dinámicamente por el controlador de la aplicación móvil.

En la figura 25 se visualiza el resultado de utilizar el código de la figura anterior.



Figura 25: Resultado del código usado en la figura anterior

Jquery Mobile también fue utilizado para definir el comportamiento de ciertos eventos dentro de la aplicación móvil desarrollada. En la siguiente figura se aprecia una muestra de código JavaScript usando métodos propios de Jquery Mobile.

```
$('#nuevo_sitio').live('pageshow',nuevoSitio);

$('#check-in').live('pageshow',listarCheckins);

$('#sitios').live('pagebeforeshow',listarSitios);

$('#checkpoints').live('pagebeforeshow',listarCheckpoints);

$('#alertas').live('pagebeforeshow',listarAlertas);

$('#configuracion').live('pagebeforeshow',cargarConfiguracion);
```

Figura 26: Código JavaScript que muestra la implementación de eventos Jquery Mobile

En el código anterior se aprecia el método `live`. El cual es un método *listener* (semejante al método `bind` de Jquery) que se activa cuando la página referenciada es desplegada. El método recibe dos parámetros; el primero es el tipo de evento que se espera y el segundo la función *callback* que se ejecutará al activarse ese evento. El evento `'pageshow'` ocurre en el momento que la página ha sido desplegada, mientras que el evento `'pagebeforeshow'` ocurre justo antes del despliegue de la página.

5.2 JSON

JSON es el acrónimo en inglés para *JavaScript Object Notation* (Notación de Objetos de JavaScript) el cual es un formato ligero de intercambio de datos e independiente del lenguaje o plataforma. Leerlo y escribirlo es simple para humanos, y para un desarrollador es simple interpretarlo con un parser y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript (estándar ECMA-262 3era Edición de Diciembre 1999). Estas propiedades hacen que JSON sea un formato ideal para el intercambio de datos.

[I-JSON]

Para el desarrollo de la solución de este Trabajo Especial de Grado, JSON fue seleccionado como formato para intercambio de datos entre la aplicación móvil y el servidor.

Para implementar JSON en la aplicación móvil se emplearon las funciones `JSON.stringify` y `JSON.parse` de JavaScript. En la figura 27 se aprecia un trozo del código utilizado en la función de sincronización de la aplicación móvil.

```
function(categorias){

    var json_categorias = JSON.stringify(categorias);

    s = new Sitios();
    s.listarU(usuario.id,
        function(sitios){

            var json_sitios = JSON.stringify(sitios);
```

Figura 27: Código JavaScript perteneciente a la función de Sincronización de la aplicación móvil

En la figura anterior se aprecia cómo se utiliza la función `JSON.stringify` en la segunda y última línea del código. La primera recibe un arreglo de categorías y la segunda un arreglo de sitios. Estos dos objetos JavaScript son transformados usando `JSON.stringify` en formato JSON y almacenados en las variables `json_categorias` y `json_sitios`, que son enviadas al servidor usando el protocolo HTTP y el método de acceso POST.

En la siguiente figura se aprecia cómo es procesada la respuesta recibida desde el servidor.

```
success: function(json_r){
    respuesta=JSON.parse(json_r);

    if(respuesta.resultado!="ok"){
        if(respuesta.resultado=="inexistente"){
            alert("El usuario no existe");
        }
        else{
            alert("La contraseña es incorrecta");
        }
    }
}
```

Figura 28: Código JavaScript perteneciente la función de Sincronización (recepción de la respuesta)

En el código anterior la variable `json_r` es el que contiene la respuesta en formato JSON recibida desde el servidor. Esta respuesta es pasada a la función `JSON.parse` la cual la transforma el mensaje JSON en un objeto JavaScript convencional.

En el lado del servidor, para poder manipular mensajes JSON de forma eficiente fue necesario utilizar la librería GSON de Java. Una librería creada por Google que permite transformar de forma sencilla los datos JSON en objetos Java y viceversa. Para hacer esto es necesario crear una clase Java que posea la estructura del elemento JSON a transformar. La siguiente figura es una captura del código implementado en el servidor, donde se transforma JSON a objetos Java.

```
Gson gson = new Gson();
Modelo.Categoria[] categorias= gson.fromJson(json_categorias, Modelo.Categoria[].class);
```

Figura 29: Código Java para transformar JSON a objetos Java

En la primera línea se instancia el objeto GSON. Este objeto es utilizado en la segunda línea para crear un arreglo en Java de las categorías recibidas (en formato JSON) desde la aplicación móvil. Para eso es necesario pasar como parámetro el String que contiene el mensaje en formato JSON recibido desde la aplicación móvil y la clase Java que posee la estructura de datos a generar (en este caso un arreglo de categorías).

Para hacer el proceso inverso de transformar objetos Java a formato JSON, simplemente basta con tener el objeto que se quiere transformar a JSON. La siguiente figura muestra un trozo del código en donde se crea la respuesta que será enviada de regreso a la aplicación móvil.

```
String json_respuesta="";
Gson gson = new Gson();
respuestaSegunda respuesta = new respuestaSegunda();

respuesta.categorias=categorias;
respuesta.sitios=sitios;
respuesta.alertas=alertas;
respuesta.checkpoints=checkpoints;
respuesta.alertas_nuevas=alertas_nuevas;
respuesta.checkpoints_nuevos=checkpoints_nuevos;

json_respuesta = gson.toJson(respuesta);
```

Figura 30: Código Java para transformar objetos Java a JSON

Como se puede apreciar en el anterior código, una vez instanciado y completado el objeto a transformar (en este caso el objeto es `respuesta`) este simplemente es pasado como parámetro al método `toJson` del objeto GSON. El cual retorna un *String* con el objeto en formato JSON (este es almacenado en la variable `json_respuesta`).

5.3 Pantalla Inicial

En esta sección se describirá el flujo de la aplicación móvil desde la pantalla inicial.

5.3.1 Crear Usuario

La figura 31 muestra la pantalla inicial de la aplicación móvil. Se puede visualizar los botones de inicio de sesión y creación de usuario, así como el formulario para ingresar el nombre de usuario y la contraseña. También se puede encontrar un “interruptor” que activa el inicio de sesión automático sin necesidad de ingresar nuevamente el usuario y la contraseña.

La imagen muestra la interfaz de usuario de la aplicación GeoRef. En la parte superior hay una barra de título negra con el texto "GeoRef" en blanco. Debajo, hay un formulario con dos campos de entrada: "Usuario" y "Contraseña". Entre estos campos hay un interruptor etiquetado "Conectar Automáticamente" con la opción "No" seleccionada. Al final del formulario, hay dos botones: "Iniciar Sesión" y "Crear Usuario".

Figura 31: Vista de la pantalla inicial de la aplicación móvil

Para ingresar al formulario de creación de una nueva cuenta de usuario, es necesario presionar el botón “Crear Usuario” de la pantalla inicial de la aplicación móvil. Esto abrirá un nuevo formulario que puede ser visualizado en la figura 32.

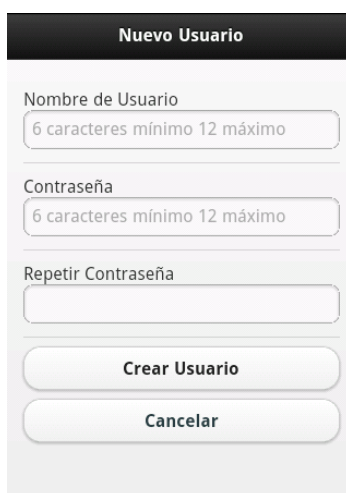
La imagen muestra el formulario "Nuevo Usuario". Tiene un título "Nuevo Usuario" en una barra superior. El formulario contiene tres campos de entrada: "Nombre de Usuario" (con una pista de "6 caracteres mínimo 12 máximo"), "Contraseña" (con una pista de "6 caracteres mínimo 12 máximo") y "Repetir Contraseña". Al final, hay dos botones: "Crear Usuario" y "Cancelar".

Figura 32: Formulario de creación de nueva cuenta de usuario

El formulario de la figura anterior cuenta con los campos donde el usuario debe introducir los datos de la cuenta de usuario a crear. Si los datos ingresados son correctos, aparecerá un mensaje de éxito de la operación. Este mensaje puede ser visualizado a continuación en la figura 33.

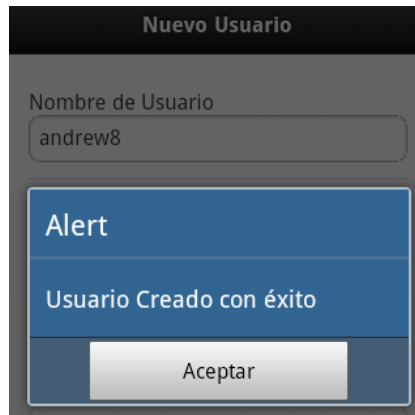


Figura 33: Mensaje de cuenta de usuario creada con éxito

En caso que el usuario haya introducido algún dato incorrecto o haya dejado de introducir alguno, la aplicación despliega un mensaje informando el respectivo error. Un ejemplo de este mensaje puede ser encontrado en la figura 34.

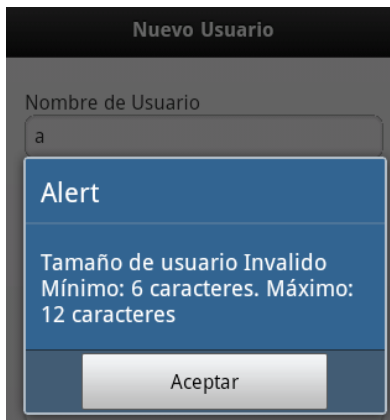


Figura 34: Mensaje mostrado cuando el usuario introduce un dato incorrecto

5.3.2 Iniciar Sesión

El usuario inicia sesión al introducir sus datos de usuario y contraseña en los respectivos campos y luego presiona el botón “Iniciar Sesión” (ver figura 30). Si los datos son autenticados correctamente, un mensaje de bienvenida será mostrado al usuario, indicando que el inicio de sesión ha sido exitoso. En la figura 35 se puede visualizar el respectivo mensaje mostrado.

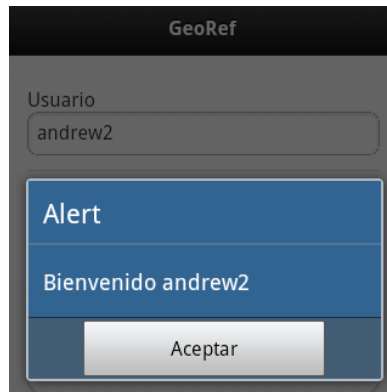


Figura 35: Mensaje mostrado al iniciar sesión correctamente

En caso de que la autenticación haya fallado (ya que se ha ingresado un nombre de usuario y/o contraseña incorrecto) será mostrado un mensaje indicando este error. En la figura 36 se visualiza este mensaje.

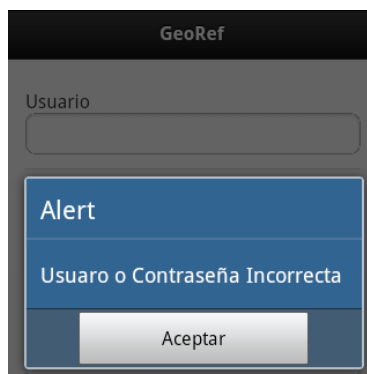


Figura 36: Mensaje mostrado al fallar la autenticación de inicio de sesión

En la figura 37 se visualiza el menú principal de la aplicación móvil, el cual se despliega después de que el usuario ha iniciado sesión correctamente y se ha mostrado el mensaje de bienvenida.

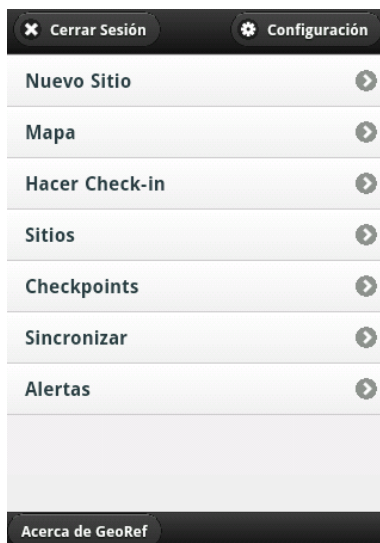


Figura 37: Menú principal de la aplicación móvil

5.4 Menú Principal

En esta sección se describirán las opciones a las que se puede tener acceso el usuario en el menú principal de la aplicación móvil.

5.4.1 Nuevo Sitio

Con la funcionalidad “Nuevo Sitio” el usuario puede crear un nuevo sitio en su cuenta. El usuario accede a esta funcionalidad al seleccionarla en el menú principal, el dispositivo procede a obtener la posición, cuyos datos serán mostrados en pantalla, así como se visualiza en la figura 38.



Figura 38: Pantalla inicial de la opción “Nuevo Sitio”

El usuario en este punto podrá presionar el botón “Actualizar Posición” de la esquina inferior izquierda, lo que hará que el dispositivo refresque la posición geográfica del usuario. En caso de que el usuario presione el botón “Continuar” de la esquina inferior derecha, pasará a la siguiente pantalla que contiene el formulario de creación de sitio, el cual se puede visualizar en la figura 39.

El formulario tiene un encabezado con un botón de retroceso y el título "Nuevo Sitio". Contiene tres campos de texto: "Nombre:" con un placeholder "Campo requerido", "Descripción:" con un placeholder "Campo opcional", y "Categoría:" con un menú desplegable que muestra "(Sin Categoría)". En la parte inferior hay un botón "Crear Sitio".

Figura 39: Formulario de creación de nuevo sitio

Este formulario contiene los campos que el usuario debe llenar para crear un nuevo sitio. Una vez completado estos campos el usuario deberá presionar sobre el botón “Crear Sitio”.

Si los datos ingresados son correctos se mostrara un mensaje notificando el éxito de la operación, el cual puede ser visualizado en la figura 40.

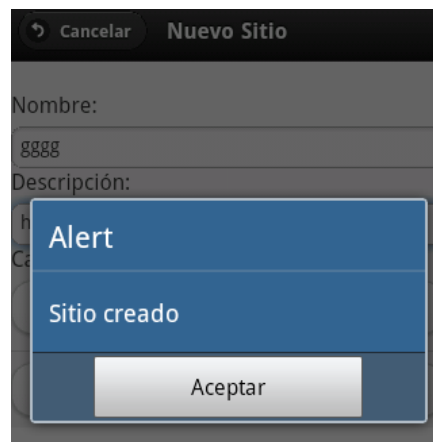


Figura 40: Mensaje notificando el éxito de la creación de un nuevo sitio

En caso que el usuario no indique un nombre para el sitio, se despliega un mensaje mostrando el respectivo error. Este mensaje se visualiza en la figura 41.

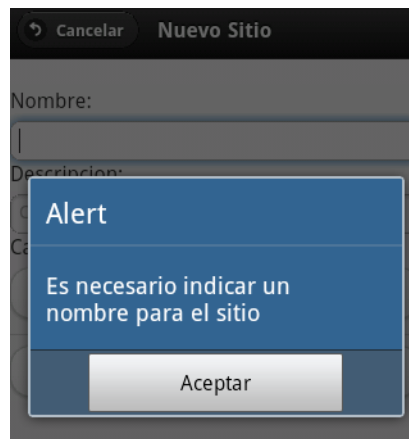


Figura 41: Mensaje notificando la necesidad de indicar un nombre para el sitio a crear

El usuario también puede seleccionar una nueva categoría para el sitio a crear. Para hacer esto debe seleccionar la opción “Otra (Nueva Categoría)” de la lista desplegable de categorías (en el formulario de “Nuevo Sitio”).

Un formulario nuevo se desplegará sobre el formulario actual, donde el usuario podrá especificar los datos de la categoría a crear. En la figura 42 se visualiza este formulario.

Una captura de pantalla de un formulario modal con el título "Nueva Categoría" y un botón de cerrar (X) en la esquina superior derecha. El formulario contiene dos campos de texto: "Nombre de categoría:" con el placeholder "Requerido" y "Descripción de categoría:" con el placeholder "Opcional". En la parte inferior del formulario hay un botón "Crear Categoría".

Figura 42: Formulario para crear una nueva categoría

Una vez que el usuario haya ingresado los datos de la categoría a crear, deberá presionar el botón “Crear Categoría” lo que producirá que una nueva categoría sea creada y esté disponible para ser seleccionada en el formulario de “Nuevo Sitio”.

5.4.2 Sitios

En la opción “Sitios” del menú principal el usuario puede consultar los distintos sitios que el haya creado usando la opción “Nuevo Sitio”. El usuario al entrar en esta opción vera en primera instancia una lista de las categorías asociadas a su cuenta (en caso que las haya), así como se visualiza en la figura 43.



Figura 43: Pantalla inicial de la opción “Sitios” del menú principal

En caso que existan sitios sin categoría, aparecerán en un renglón especial llamado “(Sin Categoría)”. Al presionar sobre alguno de los renglones se desplegara(n) lo(s) sitio(s) asociado(s) a la categoría. En la figura 44 se ha presionado en la categoría “Cat” (último renglón) y se muestra el único sitio asociado a esta.



Figura 44: Pantalla inicial de la opción Sitios del menú principal con un renglón desplegado

Al presionar sobre un sitio se muestra el perfil del sitio ingresado. En la figura 45 se visualiza el perfil del sitio identificado como “Sitio 1”.



Figura 45: Perfil de un sitio creado

El perfil de un sitio muestra toda la información relacionada a él: Nombre, los datos de longitud y longitud, la fecha y hora de creación, la descripción, la categoría asociada y un mini-mapa indicando la posición geográfica del sitio. También en la parte inferior se muestran botones con los cuales el usuario puede ejecutar acciones específicas sobre el sitio. Al presionar sobre el botón “Editar” se despliega un formulario similar al de la opción “Nuevo Sitio” (ver figura 37) con la diferencia que los campos ya tiene cargados los datos del sitio a editar. Al presionar el botón “Ver en Mapa” la posición del sitio se despliega en el mapa principal de la aplicación. Al presionar el botón “Borrar” se borra el sitio en cuestión.

5.4.3 Checkpoints

En la opción “Checkpoints” del menú principal el usuario puede consultar todos los checkpoints asociados a su cuenta (los cuales fueron descargados desde el servidor en una sincronización previa).

El usuario al entrar en esta opción primero visualiza una lista de todos los checkpoints asociados a su cuenta, tal como se aprecia en la figura 46.



Figura 46: Lista de Checkpoints

En la figura anterior los checkpoints están identificados con una bandera a cuadros. Esta bandera posee un icono en la parte inferior derecha; el cual puede ser un símbolo de visto indicando que el Checkpoint fue visitado o una “X” en rojo que indica que no ha sido.

Al presionar sobre uno de los checkpoints de la lista, se despliega el perfil del checkpoint. El cual puede ser visualizado en la figura 47.

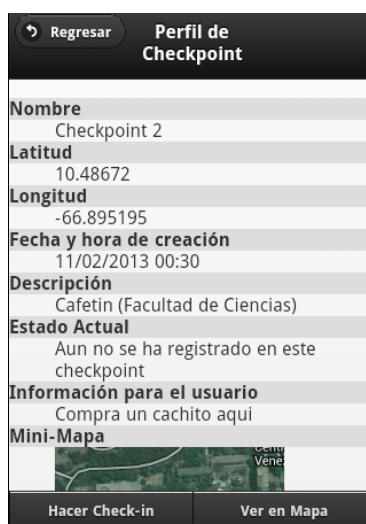


Figura 47: Perfil de un checkpoint

El perfil del checkpoint muestra la misma información que muestra el perfil de un sitio, además de la información específica de interés para el usuario. En la parte inferior se visualizan dos botones; el primero es “Hacer Check-in”, el cual permite registrar la visita al checkpoint (si el usuario está cercano a su ubicación) y el segundo es “Ver en Mapa” que permite visualizar la posición geográfica del checkpoint en el mapa principal de la aplicación.

5.4.4 Hacer Check-in

En la opción “Hacer Check-in” del menú principal el usuario visualiza los checkpoints cercanos a su posición geográfica (en caso que los haya) para así registrar su visita en ellos. Una vez ingresado a esta opción, la aplicación primero consulta la posición geográfica del dispositivo, una vez obtenida, despliega una lista de los checkpoints cercanos a la posición actual. En la figura 48 se visualiza una lista con el único checkpoint cercano a la posición actual del dispositivo móvil.

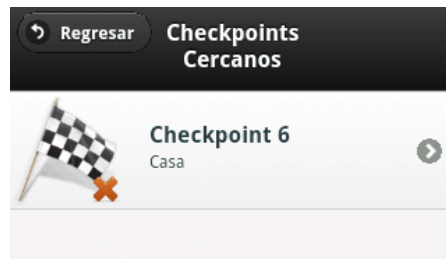


Figura 48: Lista de Checkpoints cercanos

Como se puede apreciar en la figura anterior, el checkpoint posee un icono de “X” en color rojo, indicando que no ha sido visitado aun. Al hacer clic sobre él, la aplicación nos traslada al perfil de este checkpoint (ver figura 47).

En este perfil, el usuario puede presionar el botón “Hacer Check-in”, lo que genera que la aplicación solicite la confirmación de la acción, como muestra la figura 49.

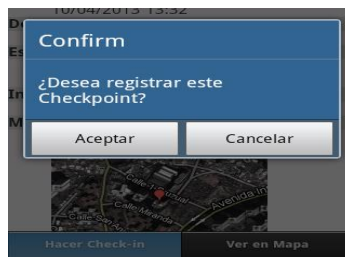


Figura 49: Mensaje de confirmación de registro de checkpoint

El usuario al presionar sobre el botón “Aceptar” del cuadro de confirmación, habrá registrado su visita al Checkpoint.

5.4.5 Mapa

En la opción “Mapa” del menú principal el usuario podrá visualizar y manipular el mapa principal de la aplicación, como se muestra en la figura 50.



Figura 50: Mapa principal de la aplicación móvil

El usuario al presionar sobre el botón “Mi posición” de la esquina superior derecha, provoca que la aplicación móvil despliegue la posición geográfica actual del dispositivo en el mapa.

Esta posición es indicada con una marca en forma de aro, justo en el centro del mapa, tal como se puede apreciar en la figura 51.

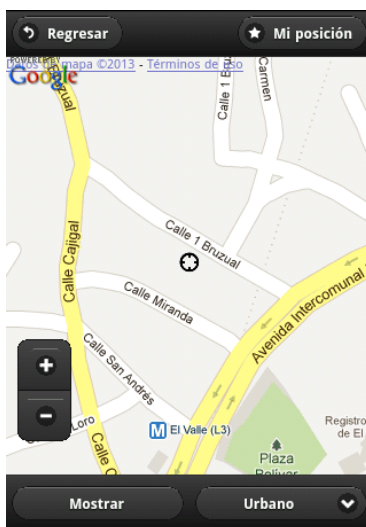


Figura 51: Mapa mostrando la posición geográfica actual del dispositivo móvil

Al presionar sobre la lista desplegable de la esquina inferior derecha se despliega la lista de los tipos de mapas disponibles, tal como se aprecia en la figura 52. El usuario puede seleccionar alguna opción de la lista para cambiar el tipo de mapa actual.



Figura 52: Lista de tipos de mapa disponibles

Al presionar sobre el botón “Mostrar” de la esquina inferior izquierda, se despliega una ventana flotante donde el usuario puede seleccionar que información desea ver sobre el mapa. Esta información puede ser los checkpoints o los sitios de una categoría. En la figura 53 se visualiza la ventana flotante de la opción “Mostrar”.



Figura 53: Ventana flotante de la opción “Mostrar”

En la figura 54 se visualiza la información mostrada en caso de que el usuario seleccione “Checkpoints” con la opción “Mostrar”. Los checkpoints visitados son indicados con banderas en color azul mientras que los no visitados son indicados con banderas en color rojo.

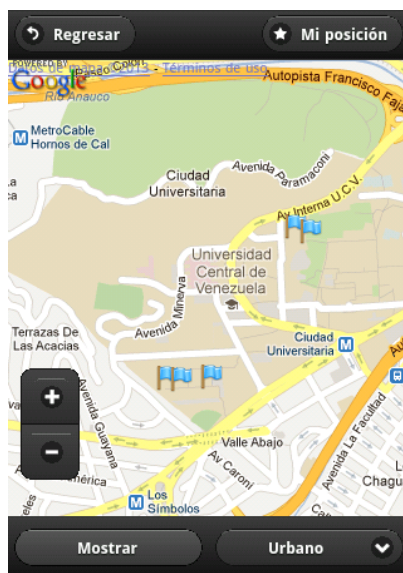


Figura 54: Checkpoints desplegados sobre el mapa de la aplicación

En la figura 55 se visualiza la información mostrada en caso que el usuario seleccione ver los sitios de una determinada categoría. Como se aprecia los sitios son identificados con un sujetador azul.



Figura 55: Sitios de una categoría desplegados sobre el mapa de la aplicación

5.4.6 Configuración

En la opción “Configuración” del menú principal el usuario puede configurar distintos elementos de la aplicación móvil. Las distintas opciones de configuración son mostradas en la figura 56.



Figura 56: Opciones de configuración de la aplicación móvil

En la figura 57 se visualiza el formulario para cambiar la contraseña actual. Este formulario se despliega cuando el usuario presiona sobre el renglón de “Nueva Contraseña”.

El formulario se encuentra dentro de un panel con el título "Configuración" y un botón "Regresar" con una flecha hacia atrás. El panel contiene una sección titulada "Nueva Contraseña" con un icono de menos. Debajo de este título, hay un campo de texto etiquetado "Contraseña:" con el texto de ayuda "6 caracteres mínimo 12 máximo". A continuación, hay otro campo de texto etiquetado "Repita Contraseña:". Al final del formulario, hay un botón "Guardar Nueva Contraseña".

Figura 57: Formulario para cambiar contraseña

En la figura 58 se visualiza la opción para configurar el “Servicio de Mapas”. Aquí el usuario podrá cambiar la fuente de mapas que usa el mapa principal de la aplicación. En caso que seleccione como fuente de mapas un servicio WMS¹³, el usuario deberá indicar la dirección URL del servidor WMS y el parámetro de capas que recibe.

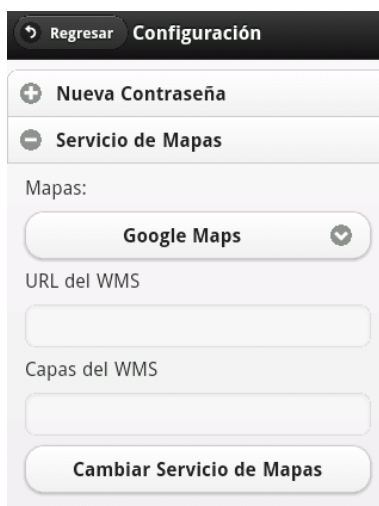
El formulario está dentro de un panel con el título "Configuración" y un botón "Regresar" con una flecha hacia atrás. El panel muestra una lista de opciones: "Nueva Contraseña" (con un icono de plus) y "Servicio de Mapas" (con un icono de menos). La opción "Servicio de Mapas" está seleccionada. Debajo de ella, hay un campo de texto etiquetado "Mapas:" con un menú desplegable que muestra "Google Maps" y un icono de flecha hacia abajo. A continuación, hay un campo de texto etiquetado "URL del WMS". Después, hay un campo de texto etiquetado "Capas del WMS". Al final, hay un botón "Cambiar Servicio de Mapas".

Figura 58: Opción para cambiar el Servicio de Mapas

¹³ Web Map Service (WMS): Tipo de servicio web el cual produce mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica.

En la figura 59 se visualiza la opción para configurar el servidor de sincronización. Aquí el usuario podrá indicar la dirección URL que la aplicación móvil utilizará para comunicarse con el servidor.



The screenshot shows a mobile application interface with a dark header bar containing a back arrow and the text 'Regresar Configuración'. Below the header is a list of settings options: 'Nueva Contraseña', 'Servicio de Mapas', and 'Servidor de Sincronización'. The 'Servidor de Sincronización' option is selected, indicated by a minus sign icon. Below this option, the text 'URL del servidor' is displayed above a text input field containing the URL 'https://190.78.86.97:8443/georef/'. At the bottom of the screen is a button labeled 'Guardar URL'.

Figura 59: Opción para configurar el servidor de sincronización

En la figura 60 se visualiza la opción para configurar las categorías asociadas a la cuenta. Aquí el usuario podrá modificar o borrar cualquier categoría que tenga asociada a su cuenta.



The screenshot shows the same mobile application interface as Figure 59, but with the 'Opciones de Categoría' option selected, indicated by a minus sign icon. Below this option, the text 'Categorías Guardadas:' is displayed above a list of saved categories. The first category listed is 'Parques', which has a checkmark icon to its right. Below the list of categories are two buttons: 'Modificar' and 'Borrar'.

Figura 60: Opción para configurar las categorías

5.4.7 Alertas

En la opción “Alertas” del menú principal el usuario podrá visualizar cualquier alerta que se haya descargado del servidor en una sincronización previa. Estas alertas son simples mensajes enviados desde el servidor, indicando (por ejemplo) la agregación de nuevos checkpoints o un reinicio de la contraseña. En la figura 60 se visualiza la bandeja de entrada de alertas.

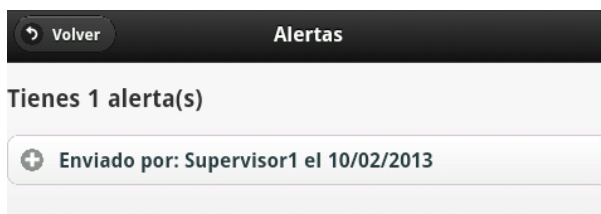


Figura 61: Bandeja de entrada de alertas

La alerta indica la fecha de creación y por quien fue enviada. Al presionar sobre esta alerta se despliega el mensaje contenido en ella y un botón para poder borrarla, tal como se visualiza en la figura 62.



Figura 62: Contenido de una alerta

5.4.8 Sincronización

En la opción de “Sincronización” el usuario podrá sincronizar los datos de la aplicación móvil con los datos en el servidor. Descargando y/o subiendo cualquier información necesaria. En caso que la sincronización con el servidor haya sido efectiva, la aplicación móvil mostrara un mensaje de éxito de la operación, como el que puede ser visualizado en la figura 63.

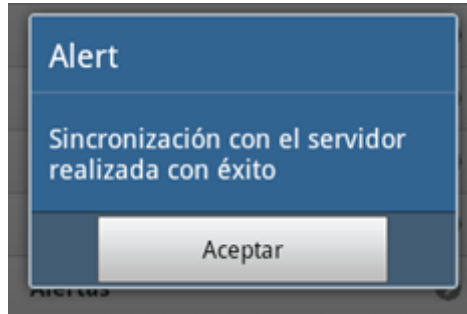


Figura 63: Mensaje de éxito de sincronización con el servidor

En caso que la sincronización no haya sido exitosa, ya sea porque no ha sido posible comunicarse con el servidor o ha habido algún fallo en la autenticación en el servidor, se despliega un mensaje como el visualizado en la figura 64.

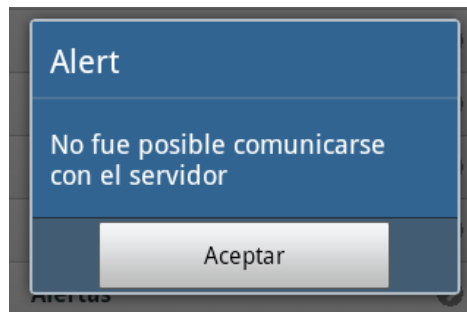


Figura 64: Mensaje de error de sincronización con el servidor

5.5 Base de Datos Geoespacial

Una base de datos espacial según [U-GRSDB] es una base de datos que es capaz de almacenar y manejar de forma optimizada objetos en el espacio, estos objetos pueden ser puntos, líneas o polígonos y en algunos casos hasta objetos tridimensionales. Una base de datos espacial, llevada al ámbito geográfico pasaría a llamarse “Base de datos Geoespacial”, en este caso los objetos almacenados representarían lugares o sitios como ciudades, países, municipios, ríos, lagos y cualquier otro lugar geográfico que pueda ser representado de forma espacial.

Para el desarrollo del presente trabajo especial de grado fue necesaria la implementación de una base de datos geoespacial, donde se pudieran persistir los datos de los usuarios que usen la aplicación móvil. Para esto se optó por usar una base de datos de tipo PostgreSQL con extensión PostGIS.

PostgreSQL [<http://www.postgresql.org/>] es un manejador de base datos relacional de código abierto. Con más de 15 años en el mercado, es una de las bases de datos más usadas. Disponible para la mayoría de los sistemas operativos existentes (Linux, Windows, MacOS), ofrece soporte para la mayoría de los tipos de datos y funciones SQL, esto entre otras características propias de este manejador de base de datos.

PostGIS [<http://postgis.refrations.net/>] es una extensión de PostgreSQL, la cual agrega soporte a objetos geográficos. Teóricamente hablando PostGIS “habilita espacialmente” al servidor PostgreSQL, permitiendo que este sea usado como una Base de Datos espacial de tipo *backend*, que albergue información relacionada a un SIG.

Únicamente los sitios y checkpoints de las aplicaciones móviles fueron almacenados como objetos geográficos en la base de datos geoespacial. Para lograr esto fue necesaria la creación de tablas habilitadas espacialmente que pudieran almacenar estos objetos. En la figura 65 se muestra la instrucción necesaria para crear la tabla checkpoints de la base de datos geoespacial.

```

CREATE TABLE checkpoints
(
    id serial NOT NULL,
    nombre character varying,
    fecha timestamp without time zone NOT NULL,
    descripcion character varying,
    id_usuario integer NOT NULL,
    sincronizado boolean NOT NULL,
    id_administrador integer NOT NULL,
    info character varying,
    checked_in boolean NOT NULL,
    coordenadas geometry(Point,4326),
    CONSTRAINT checkpoints_pkey PRIMARY KEY (id),
    CONSTRAINT checkpoints_id_administrador_fkey FOREIGN KEY (id_administrador)
        REFERENCES administradores (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT checkpoints_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES usuarios (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

Figura 65: Instrucción SQL para crear tabla de checkpoints en la base de datos geoespacial

La columna `coordenadas` es la que contiene los atributos geoespaciales del checkpoint. Para crear esta columna es necesario usar la función `geometry` de PostGIS, la cual recibe como parámetros el tipo de geometría (en este caso es *Point* de “punto”) y la proyección que usara la geometría (en este caso 4326).

Para agregar un nuevo registro en la tabla creada, se utiliza una sentencia SQL común y corriente, sin embargo para establecer el atributo geográfico, es necesario utilizar otra función especial de PostGIS, la cual permite establecer la geometría en un formato en específico. En la figura 66 se muestra la sentencia SQL para insertar un nuevo registro en la tabla `checkpoints`.

```

INSERT INTO
checkpoints(nombre, fecha, descripcion, id_usuario, sincronizado, id_administrador, info, checked_in, coordenadas)
VALUES
('Checkpoint 5', now(), 'Tierra de Nadie (UCV)', 2, FALSE, 1, '', FALSE, ST_GeomFromText('POINT(-66.890329 10.491742)', 4326))

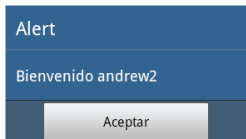
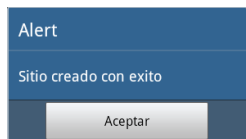
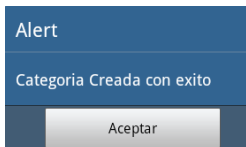
```

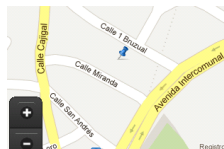
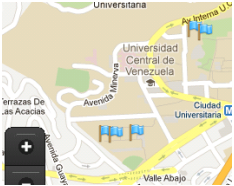
Figura 66: Instrucción SQL para insertar un registro en la tabla de checkpoints

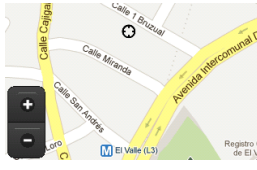
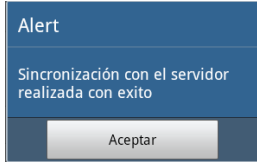
Como se puede apreciar en la captura, para definir la geometría se usó la función `ST_GeomFromText`, la cual recibe como parámetro la geometría en formato texto (con sus valores de longitud y latitud) y la proyección usada (en este caso 4326).

5.6 Pruebas de la aplicación

A continuación se listan las pruebas realizadas a cada una de las funcionalidades descritas en la sección de casos de uso. La siguiente tabla identifica la funcionalidad por nombre, el resultado esperado y el resultado obtenido, así como observaciones de la prueba por funcionalidad.

Nombre	Resultado esperado	Resultado obtenido	Observaciones
1. Crear Usuario	Se crea la cuenta de usuario con los datos ingresados	Usuario creado con éxito.	
2. Iniciar Sesión	Se muestra el menú principal de la aplicación móvil.	Sesión iniciada correctamente. 	
3. Crear Sitio	Se crea un nuevo sitio en la cuenta del usuario en sesión.	Sitio creado con éxito. 	Por aspectos del dispositivo y de la red, a veces es necesario más de un intento para obtener exitosamente la ubicación.
3.1. Actualizar Ubicación	Se refresca la ubicación del sitio a crear.	Nueva ubicación mostrada con éxito. 	Por aspectos del dispositivo y de la red, a veces es necesario más de un intento para obtener exitosamente la ubicación.
3.2. Crear Categoría	Se crea una nueva categoría.	Categoría creada con éxito. 	
4. Consultar Sitio	Se visualizan los atributos de un sitio.	Perfil de sitio desplegado.	
4.1. Editar Sitio	Se cambian algunos atributos de un sitio.	Cambios realizados con éxito.	

4.2. Borrar Sitio	Se elimina un sitio.	Sitio eliminado con éxito.	
4.3. Ver Sitio en Mapa	Se visualiza el sitio en el mapa principal de la aplicación.	Sitio desplegado sobre el mapa. 	
5. Consultar Checkpoint	Se visualizan los atributos de un checkpoint.	Perfil de checkpoint desplegado con éxito.	
6. Registrar Visita en Checkpoint	Se registra la visita en un checkpoint.	Visita registrada con éxito.	Por aspectos del dispositivo y de la red, a veces es necesario más de un intento para obtener exitosamente la ubicación.
7. Ver Mapa	Se visualiza el mapa de la aplicación.	Mapa desplegado. 	
7.1. Ver Sitios (en mapa)	Se visualizan los sitios de una categoría seleccionada en el mapa.	Sitios de la categoría seleccionada desplegados sobre el mapa. 	
7.2. Ver Checkpoints (en mapa)	Se visualizan los checkpoints en el mapa.	Checkpoints desplegados sobre el mapa. 	
7.3. Visualizar	Se visualiza una marca en el mapa indicando la ubicación	Ubicación actual mostrada con una marca sobre el	Por aspectos del dispositivo y

Ubicación Actual	actual.	mapa. 	de la red, a veces es necesario más de un intento para obtener exitosamente la ubicación.
8. Sincronizar	Se sincroniza la información de la aplicación con el servidor y viceversa.	Sincronización realizada con éxito. 	
9.1. Configurar Contraseña	Se cambia la contraseña.	Contraseña cambiada con éxito.	
9.2. Configurar Mapas	Se cambia el servicio de mapas usado por la aplicación.	Servicio de mapa cambiado con éxito.	
9.3. Configurar Servidor	Se cambia o asigna la dirección del servidor de sincronización.	Dirección del servidor cambiada o asignada con éxito.	
9.4. Configurar Categorías	Se modifican o eliminan categorías del usuario en sesión.	Modificación o eliminación de categoría realizada con éxito.	
10. Ver Alerta	Se visualiza una alerta.	Alerta visualizada.	
10.1. Borrar Alerta	Se borra una alerta.	Alerta eliminada con éxito.	

Los resultados de las pruebas demuestran la validez funcional de la aplicación diseñada y desarrollada.

CONCLUSIONES

A continuación se presentan las conclusiones del Trabajo Especial de Grado.

Se cumplió el objetivo del Trabajo Especial de Grado de diseñar y desarrollar una aplicación móvil que permita la georeferenciación de sitios de interés por parte del usuario.

Los objetivos específicos también fueron alcanzados. Se diseñaron y desarrollaron funcionalidades que fueron capaces de obtener la localización del dispositivo (en forma de coordenadas geográficas). Esto se pudo lograr gracias al método **watchPosition** que ofrece PhoneGap para el manejo de la geolocalización. La razón de escoger este método sobre **getCurrentPosition** es que el dispositivo al no obtener la ubicación geográfica inmediatamente, fue necesario monitorearla durante una ventana de tiempo específica, hasta que se obtuviera una ubicación con buena precisión. Es necesario resaltar que no es posible saber a través de que método se está obteniendo la ubicación geográfica

También se diseñó y desarrolló la funcionalidad de autenticación para el usuario en la aplicación móvil. Este módulo solo realiza la autenticación de forma local, sin acceder a ningún servicio remoto. Se hizo de esta forma con la intención de que la aplicación fuera lo más independiente posible de una conexión de red. La aplicación solo carga la información geográfica relacionada al usuario en sesión, recuperando esos datos de la base de datos local del dispositivo.

Se diseñaron y desarrollaron funcionalidades de gestión de los datos georeferenciados en la aplicación móvil, usando para esto las herramientas provistas por el framework JQuery Mobile. Estas herramientas facilitaron el desarrollo de distintas interfaces de usuario, que posibilitaron al usuario móvil realizar las tareas de gestión de los datos georeferenciados.

Utilizando la librería OpenLayers se pudieron diseñar y desarrollar mecanismos de despliegues de mapas en la aplicación móvil, que a su vez utilizaron distintos servicios de información geográfica (Google Maps, Bing Maps, OpenStreetMap y WMS). Gracias a las funciones desarrolladas usando los métodos que ofrece la

librería para la creación y despliegue de objetos geográficos, el usuario puede visualizar la información georeferenciada de su interés usando el servicio de mapas de su preferencia.

Se diseñaron y desarrollaron funcionalidades del lado del servidor empleando el lenguaje Java y el contenedor web que soporta la ejecución de Java servlets, que respondieron exitosamente a las peticiones de sincronización recibidas por los dispositivos móviles.

Las funcionalidades del lado del servidor realizan tareas de persistencia en una base de datos PostgreSQL, la cual está habilitada geoespacialmente usando PostGIS. Debido a que solo se manejaron objetos geográficos de tipo “Punto”, el proceso de implementar las distintas tablas y los datos dentro de la base de datos geoespacial, fue bastante sencillo.

Se escogió a JSON como formato de datos para las interacciones entre la aplicación móvil y el servidor, debido a su facilidad de entenderlo y procesarlo. Gracias a la librería GSON de Java, la transformación de datos JSON a objetos java (y viceversa) no tuvo mayor inconveniente.

Como esquema de comunicación entre la aplicación móvil y el servidor, se optó por diseñar y desarrollar un esquema RESTful. Usando este esquema (al estar basado en el protocolo HTTP) la integridad de los datos está garantizada en las interacciones entre la aplicación móvil y el servidor. Para las peticiones HTTP desde la aplicación móvil, se utiliza el método POST el cual envía los datos en formato JSON.

La aplicación móvil fue desarrollada tanto para dispositivos Android como para dispositivos BlackBerry, usando el *framework* PhoneGap en combinación con el *framework* JQuery Mobile (para crear la interfaz gráfica). Para el despliegue de mapas e información georeferenciada se utilizó la librería OpenLayers. Debido a que todas estas son tecnologías web, su integración fue bastante sencilla por lo que no hubo ningún inconveniente.

El framework PhoneGap es una distribución de la librería Apache Cordova, que contiene un conjunto de APIs que permiten al desarrollador de la aplicación móvil acceder a las capacidades nativas del dispositivo. Por lo tanto se puede considerar la librería Apache Cordova como el motor que soporta PhoneGap.

En cuanto a los Servicios Basados en Localización (SBL) podemos concluir lo siguiente:

Los SBL están basados en la posición geográfica actual del usuario, por lo que es imprescindible que el dispositivo móvil que use éstos servicios, contenga capacidades de geolocalización que permitan obtener de alguna forma la posición geográfica actual.

El método que ofrece la geolocalización más precisa es el Sistema de Posicionamiento Global (GPS). Sin embargo, a veces no es posible usar este método debido a que el dispositivo puede no tener vista directa al cielo. Por lo que se recomienda siempre tener Wifi activado a la hora de usar SBL (no importa si se está usando o no una red inalámbrica). Esto permite que el dispositivo utilice la geolocalización por Wifi, la cual puede ofrecer una precisión relativamente buena.

Los SBL tienen una gran variedad de usos y aplicaciones en distintos campos. Entre los cuales se pueden citar el campo de las redes sociales, la publicidad o mercadeo y los servicios de rastreo. El punto en común de como los SBL son utilizados en estas y otras áreas, es que en todas ellas se le ofrece al usuario información relacionada a la ubicación del dispositivo.

REFERENCIAS BIBLIOGRAFICAS

[R&A2011] Ferraro, Richard. Aktihanoglu, Murat. (2011). *Location-Aware Applications*. Manning Publications. ISBN: 978-1935182337.

[U-SNLBS] Shah, Neil. *Location Based Services Part I: Technologies in Wireless Networks*. Disponible en: <http://technowizz.wordpress.com/2010/01/03/lbs-technologies-part-1/>

[U-QMLBS] Mahmoud, Qusay H. (2004). *J2ME and Location-Based Services*. Disponible en: <http://developers.sun.com/mobility/apis/articles/location/>

[EJ-GPS] Garcia, Jose. *Así funciona el GPS*. Disponible en: http://www.asifunciona.com/electronica/af_gps/af_gps_8.htm

[WF2012] Vega, Freddy. (2012). *Cómo funciona la geolocalización por WiFi*. Disponible en: <http://www.cristalab.com/blog/como-funciona-la-geolocalizacion-por-wifi-c107677/>

[NA-GSM] Naik, Abhijit. (2012). *GSM Mobile Phone Tracking System*. Disponible en: <http://www.buzzle.com/articles/gsm-mobile-phone-tracking-system.html>

[E-2010] Eydie. (2010). *LBS: No BS, It's Mobile Marketing's Game-Changer*. Disponible en: <http://www.mobilemarketingwatch.com/lbs-no-bs-its-mobile-marketings-game-changer-7883/>

[PG-DOC] PhoneGap API Reference. Disponible en: <http://docs.phonegap.com/en/2.5.0/index.html>

[OSMF] OpenStreetMap Foundation. Disponible en: http://wiki.osmfoundation.org/wiki/Main_Page

[W3CGC] Popescu, Andrei. (2012). *Geolocation API Specification*. Disponible en: <http://dev.w3.org/geo/api/spec-source.html>

[U-JQMO] jQuery Mobile Overview. Disponible en: <http://jquerymobile.com/demos/1.1.0/docs/about/intro.html>