

FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y
COMPUTACIÓN.
UNIVERSIDAD NACIONAL DE CÓRDOBA.



“Desarrollo de una aplicación móvil basada en Android para el combate de incendios forestales y de interfaz”.

Autora: Utrera, Valeria.

Director: Lis, Diego (Lic. en Cs. de la Computación).

Director: Lanfri, Mario A. (Lic. en Física).

TRABAJO ESPECIAL PARA LA OBTENCIÓN DEL TÍTULO DE GRADO EN
LA LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.



«Desarrollo de una aplicación móvil basada en Android para el combate de incendios forestales y de interfaz» por Utrera, Valeria. Se distribuye bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 2.5 Argentina.

CÓRDOBA, ARGENTINA.

AGOSTO 2016

Resumen

La manipulación y control de incendios ha sido un reto muy grande para la humanidad desde su origen, debido a que están estrictamente ligados a las condiciones climáticas que no pueden ser controladas por el hombre; tales como las sequías, velocidad del viento, la temperatura y la humedad del suelo, entre otras. Contar con la información precisa y actualizada es vital para plantear una estrategia efectiva con el objetivo de contener y controlar los distintos focos de incendio.

Gracias al gran avance de la tecnología móvil en la actualidad, se ha dispuesto como propósito del presente trabajo crear una aplicación basada en la plataforma Android para teléfonos inteligentes. La aplicación actuará como cliente de servidores geográficos y consultará los datos requeridos, que luego serán expuestos a los encargados del manejo del fuego a través de una interfaz de mapas, brindando la información necesaria para llevar adelante cálculos y control respecto a diferentes variables en el momento de una crisis.

Palabras Clave: Aplicación Android, Incendios, Bomberos, Servidor Geográfico, Datos Georeferenciados, Manipulación de mapas, Desarrollo Iterativo.

Abstract

The manipulation and control of fires has been a big challenge for the humanity since its beginning, because they are strictly linked to weather conditions that can not be controlled by man; such as drought, wind speed, temperature and soil moisture, among others. Having accurate and current information is vital to make an effective strategy in order to contain and control the different sources of fire.

Thanks to the breakthrough of mobile technology today, the purpose of this work has been to create an application based on the Android platform for smartphones. The application will act as a client of geographical servers and will consult the required data, which will then be exposed to those responsible for fire management through a map interface, providing the necessary information to carry out calculations and control with respect to different variables at the time of a crisis.

Keywords: Android Application, Fires, Firemen, GeoServer, Georeferenced Data, Maps Handling, Iterative Development.

Agradecimientos

Quiero dedicar este trabajo a mi familia, muy especialmente a mis padres que siempre estuvieron a mi lado y me apoyaron incondicionalmente. Me dieron la confianza y la fortaleza para superarme cada día, permitiéndome progresar en esta maravillosa carrera profesional.

A mis amigos y compañeros de curso, con quienes compartí momentos únicos a lo largo de todos estos años, ayudándonos en cada trabajo y pasando horas interminables de estudio, siempre entre risas y un gran sentido del humor.

A mis directores por orientarme a lo largo de estos meses, mostrándome en cada momento una completa disposición para realizar las reuniones necesarias y responder a todas las dudas que me fueron surgiendo, brindando valiosas observaciones que guiaron el proceso de trabajo.

También agradezco muy profundamente a Fabián Freccia, bombero voluntario de la defensa civil de Río Ceballos, por su tiempo, dedicación, recomendaciones y ayuda incondicional aportada en todo momento durante el desarrollo de la aplicación.

De igual manera quería agradecer al profe Dr. Marcelo Scavuzzo por confiar en mí y recomendarme para la realización de este proyecto. Así también como a los profes Dra. Laura Brandán y el Dr. Nicolás Wolovick quienes generosamente accedieron a tomarse el tiempo para la evaluación, aportando sus correcciones y observaciones tenidas en cuenta al concluir con este trabajo.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Incendios	1
1.2.1. Tipos de incendios	2
1.2.2. Involucrados	2
1.2.3. Ataque del incendio	3
1.2.4. Guardia de ceniza	4
1.3. Objetivos del proyecto	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
2. Aplicación Implementada	7
2.1. Destinatarios	7
2.2. Funcionalidades Generales	7
2.3. Requerimientos Iniciales	8
2.4. Herramientas utilizadas	10
2.4.1. Entorno de trabajo	10
2.4.2. Bibliotecas Externas	11
2.4.3. Servidores	12
2.4.4. Almacenamiento de datos	13
2.5. Metodología y proceso de trabajo	13
3. Versiones de la aplicación	15
3.1. Prototipos iniciales	16
3.1.1. Apreciación final y correcciones	17
3.2. Versión 1	17
3.2.1. Análisis	17
3.2.1.1. Requerimientos a satisfacer	17
3.2.1.2. Historias de usuario	17
3.2.1.3. Casos de uso	18
3.2.2. Diseño	22
3.2.2.1. Arquitectura	22
3.2.2.2. Base de datos	23
3.2.3. Implementación	23
3.2.4. Resultados	24
3.2.4.1. Requisitos de instalación	24
3.2.4.2. Capturas	25
3.2.4.3. Pruebas	26

3.2.4.4.	Apreciación final y correcciones	27
3.3.	Versión 2	28
3.3.1.	Análisis	28
3.3.1.1.	Requerimientos a satisfacer	28
3.3.1.2.	Historias de usuario	28
3.3.1.3.	Casos de uso	30
3.3.2.	Diseño	38
3.3.2.1.	Base de datos	39
3.3.3.	Implementación	39
3.3.4.	Resultados	40
3.3.4.1.	Capturas	40
3.3.4.2.	Pruebas	48
3.3.4.3.	Apreciación final y correcciones	49
3.4.	Versión 3	50
3.4.1.	Análisis	50
3.4.1.1.	Requerimientos a satisfacer	50
3.4.1.2.	Historias de usuario	50
3.4.1.3.	Casos de uso	50
3.4.2.	Diseño	52
3.4.3.	Implementación	52
3.4.4.	Resultados	54
3.4.4.1.	Requisitos de instalación	54
3.4.4.2.	Capturas	54
3.4.4.3.	Pruebas	57
3.4.5.	Apreciación final y correcciones	57
3.5.	Versión 4	58
3.5.1.	Análisis	58
3.5.1.1.	Requerimientos a satisfacer	58
3.5.1.2.	Historias de usuario	58
3.5.1.3.	Casos de uso	59
3.5.2.	Diseño	62
3.5.2.1.	Base de datos	62
3.5.3.	Implementación	62
3.5.4.	Resultados	63
3.5.4.1.	Requisitos de instalación	63
3.5.4.2.	Capturas	63
3.5.4.3.	Pruebas	68
3.5.5.	Apreciación final y correcciones	69
3.6.	Versión 5	69
3.6.1.	Análisis	69
3.6.1.1.	Requerimientos a satisfacer	69
3.6.1.2.	Historias de usuario	70
3.6.1.3.	Casos de uso	71
3.6.2.	Diseño	74
3.6.2.1.	Arquitectura	74
3.6.2.2.	Base de datos	74
3.6.3.	Implementación	76
3.6.4.	Resultados	77
3.6.4.1.	Capturas	77
3.6.4.2.	Pruebas	82

3.6.4.3. Apreciación final y correcciones	82
4. Conclusión	85
4.1. Trabajos futuros	86
Bibliografía	89

Capítulo 1

Introducción

1.1. Motivación

La revolución en el mundo de la telefonía móvil es un hecho que está cambiando el mundo de la tecnología por completo. En todos los sectores, ya sean tecnológicos o no, se está intentando aprovechar esta revolución. Nuestra motivación principal fue llevar dicha tecnología al sector de control de incendios.

La Comisión Nacional de Actividades Espaciales (CONAE), responsable del programa espacial de Argentina, cuenta con sistemas para el procesamiento de datos geográficamente referenciados y que pueden ser accedidos de forma pública a través de su representación gráfica, como mapas y capas geográficas. Llevan a cabo proyectos para trabajar en la prevención y lucha contra incendios forestales y de interfaz.

El proyecto realizado, nace con la idea de dotar a los encargados del manejo del fuego, de datos georeferenciados e información general sobre el medio ambiente, de una forma rápida y en tiempo real. Para que les sea de utilidad al llevar a cabo la toma de decisiones frente a una emergencia y puedan ser capaces de comunicarse para realizar su trabajo con un mayor control del siniestro.

1.2. Incendios

Existen varias maneras de definir a los incendios forestales. Casi todas concluyen en que se trata de fuegos de grandes proporciones no programados o no controlados, que afectan de diversas formas, pudiendo provocar daños materiales, interrupción de los procesos de producción, pérdida de vidas humanas y afectación al ambiente, causando un gran daño ecológico en vastas extensiones de terreno, sobre la flora, la fauna, los recursos hídricos y los suelos.

El fuego es una reacción química que requiere de temperatura, oxígeno y material combustible para producirse y mantenerse en el tiempo. Si alguno de estos factores es removido, no podría existir el fuego, por lo tanto la tarea de los combatientes de un incendio es eliminar uno de los elementos mencionados. Ya sea cortando o barriendo el combustible, sofocando produciendo un vacío para desplazar el aire privando de oxígeno al fuego o rociándolo con agua para bajar su temperatura.

Cada incendio está formado por, una cabeza que es la parte delantera del incendio, la cual va avanzando y destruyendo con mayor intensidad; los flancos izquierdos y derechos, y la cola que va más lento quemando todo lo que no fue alcanzado anteriormente.

A continuación se explicarán diferentes términos y particularidades que ayudarán a enmarcar el presente proyecto en el contexto de los incendios, los cuales luego serán tenidos en cuenta para el desarrollo de la aplicación móvil. Se hará un resumen de las características relacionadas al incendio, la forma de trabajar de los bomberos y la información relevante que usualmente se recolecta en un siniestro, para tener una idea general del entorno en el cual se utilizará la aplicación y de los datos que serán de utilidad para los encargados del fuego.

1.2.1. Tipos de incendios

Existen diferentes tipos de incendios diferenciados según el lugar en el cual se producen. Ya sean en sectores urbanos, industriales o forestales. En el presente proyecto se tuvieron en cuenta aquellos incendios que se desarrollan en áreas forestales y los incendios producidos en áreas contiguas urbanas y rurales, o donde se entremezcla la vegetación con estructuras edilicias, como viviendas, establecimientos agrícolas, entre otros. A este último tipo de incendios se los denomina **Incendios de Interfaz**. Se pueden dividir en tres tipos de Interfase:

- Interfase Clásica: Involucra grandes urbanizaciones en contacto con áreas naturales, como las zonas de montaña.
- Interfase Cerrada: Se produce cuando en medio de la urbanización se encuentran espacios verdes aislados.
- Interfase Mixta: Incluye viviendas aisladas rodeadas de grandes áreas de vegetación.

Además los incendios pueden ser clasificados en tres categorías, dependiendo del estrato en el que se propague el fuego:

- Incendios de copa: El fuego se propaga sobre las copas de los árboles. Suelen avanzar con mayor velocidad puesto que el viento es más fuerte en las alturas y se producen llamas de gran tamaño.
- Incendios subterráneos o de subsuelo: Se propaga por debajo del combustible. Se producen en menor medida, no producen llamas y su propagación es muy lenta. Son los más peligrosos puesto que al no poder verlos, pueden desatar nuevos focos de incendio a mayores distancias.
- Incendio de superficie: Se extiende sobre la superficie del suelo. Son los más frecuentes y a menudo dan inicio a los otros tipos de incendios mencionados.

1.2.2. Involucrados

Las organizaciones de protección contra incendios forestales requieren anualmente una gran cantidad de trabajadores y empleados, cada uno de los cuales

desempeñan cargos y funciones que son críticos al momento de controlar los incendios forestales.

Cuando se desata un incendio, previamente al combate, se forma un centro de operaciones de emergencia (COE). Cuenta con un puesto de comando en donde se encuentra el jefe de bomberos, gente de defensa civil, de la policía y representantes de otros organismos involucrados. Se arma una mesa operativa donde se recibe y procesa la información necesaria, y se toman las decisiones para enfrentar la crisis. Se encargan de coordinar y comunicarse con los combatientes en campo, hacen manejo de la cartografía digital, se controla la dirección del relevamiento, se van registrando datos meteorológicos y así el supervisor puede tener una vista panorámica de la situación, para poder controlarla.

Los combatientes en campo llevan radios y un dispositivo GPS, para comunicarse con el COE y enviarles los datos de su ubicación, el tipo de incendio, dimensiones y velocidad de propagación, superficie afectada, combustible y topografía del lugar, entre otros datos que sean considerados relevantes para el correcto manejo del siniestro.

Cada defensa civil es autónoma y cada cuartel de bomberos cubre una jurisdicción. Los parques nacionales son cubiertos por un cuerpo aparte, que son los brigadistas forestales. Los cuarteles están dentro del plan provincial del manejo del fuego, que coordina todos los cuarteles de bomberos voluntarios, y el plan nacional es quien controla los cuarteles provinciales. Si se produce un incendio forestal de gran magnitud el plan provincial se encarga de trasladar al lugar, cuarteles de otros sitios alejados [27].

Al finalizar el incendio, se realiza un parte con toda la información recaudada. Estos datos pueden ser de utilidad para investigaciones futuras, por ejemplo para saber cómo fue avanzando el incendio y el tipo de vegetación quemada tanto autóctona como invasora.

1.2.3. Ataque del incendio

Al llegar a la ubicación del incendio, los combatientes envían información al COE para informarles sobre la situación y para recibir las instrucciones de actuación. Dependiendo del tipo de incendio deben estar propiamente equipados. Algunas de las herramientas que utilizan para el ataque son: rastrillos, palas, mochilas de agua, chicote, cuchillas, pico, entre otros. Además se utilizan vehículos contra incendios, como autobombas, motobombas, bulldozer, entre otros, pero solamente cuando pueden acercarse lo suficiente.

En incendios de gran magnitud también se utilizan aviones hidrantes que tiran una cortina de agua sobre el incendio, para bajar la temperatura, humectando el combustible y bajando el alto de llama, dando la posibilidad de que los bomberos en tierra puedan acercarse lo suficiente para combatirlo.

En primer lugar, los bomberos actúan sobre la cabeza del incendio si tienen acceso, para ir dividiéndolo, pues el incendio va generando mucho calor y viento, por lo tanto se intenta bajar la temperatura al dividirlo y luego van trabajando sobre los flancos.

Existen diferentes tipos de ataque:

- Ataque directo: Utilizando mochilas de agua, chicote, cuchillas, palas, en-

tre otras herramientas que ayuden a sofocar o bajar la temperatura.

- **Ataque indirecto:** Se realiza un cortafuego. Limpieza con azadones, hachas o motosierras, antes de la llegada del fuego. Se deja sin vegetación una franja y cuando llega el fuego ya no tiene combustible para quemar.
- **Ataque paralelo:** Se realiza un contrafuego. Se prende fuego controlado a los alrededores para que cuando llegue el fuego, ya se haya consumido todo el combustible, impidiendo la continuidad del incendio.

1.2.4. **Guardia de ceniza**

Una vez apagado el incendio por completo, se realiza la guardia de ceniza por todo el perímetro del área quemada. Consiste en recorrer el área, limpiarla y, cuidar que las cenizas y restos no se vuelvan a encender. Los bomberos deben separar lo que está quemado de lo que está sin quemar, lo cual equivale a un 80% aproximadamente, de que el incendio no se vuelva a iniciar. Por esto es muy importante la guardia de ceniza, si se realiza incorrectamente, se puede dar la situación donde una brasa se prenda nuevamente y con la ayuda del viento se traslade hacia sectores no quemados, teniendo como consecuencia el reinicio del incendio.

1.3. **Objetivos del proyecto**

1.3.1. **Objetivo General**

El objetivo del presente proyecto es brindar una herramienta móvil para la toma de decisión, por parte del personal de manejo de incendios, frente a emergencias relacionadas a incendios forestales y de interfaz.

1.3.2. **Objetivos Específicos**

Algunos objetivos puntuales planteados al comienzo del proyecto son:

- Desarrollar una aplicación para la plataforma Android, que permita al usuario un acceso rápido y preciso en todo momento, a datos relacionados con el manejo del fuego.
- Acceder a datos geoespaciales y proveer al usuario una interfaz de mapas para la visualización de capas GIS desde servicios web, siguiendo los protocolos WMS (Web Map Service), WFS (Web Feature Service), WCS (Web Coverage Service) y WPS (Web Processing Service) [35]. Por ejemplo capas de meteorología, de suelo, recursos hídricos, entre otras.
- Aprendizaje y manipulación de la plataforma Android, y de protocolos de comunicación web.
- Aprendizaje del lenguaje de programación Java y ampliación de los conocimientos en Python y bases de datos.

- Reunir y repasar lo visto durante el transcurso de la carrera, aplicando estos conocimientos junto con las distintas metodologías aprendidas para gestionar y lograr un buen desempeño, tanto en la organización como en la realización de un proyecto de mayor envergadura.

Capítulo 2

Aplicación Implementada

2.1. Destinatarios

El producto desarrollado está dirigido principalmente a los encargados del manejo del fuego, aunque también se planteó e implementó una serie de funcionalidades para que pueda ser utilizado por el público en general.

En cuanto a los encargados del fuego se ven incluidos, cuarteles y escuelas de bomberos, brigadas forestales, parques y zonas protegidas tanto regionales como nacionales, y organismos municipales, provinciales y nacionales con incumbencia en el manejo de incendios.

En la aplicación se destacan principalmente dos tipos de usuarios que la utilizarán, teniendo en cuenta el puesto y las tareas de cada uno. Por un lado se encuentran los supervisores encargados de planear y llevar a cabo el control del incendio; y por el otro se encuentran los combatientes quienes están a cargo de un supervisor. Cada combatiente es enviado al incendio para realizar el trabajo de campo y se encarga de extinguir el fuego obedeciendo las ordenes del supervisor.

2.2. Funcionalidades Generales

Teniendo en cuenta los puntos mencionados anteriormente sobre los incendios y la forma de actuar de los encargados del fuego, se plantearon diferentes necesidades que llevaron a la construcción de las funcionalidades de la aplicación desarrollada.

Una de las necesidades del personal involucrado en la prevención y extinción del fuego, es el estudio del suelo y las condiciones meteorológicas del área del incendio, antes y durante el siniestro para poder realizar una evaluación previa sobre el posible avance del fuego. Para esto, en la aplicación desarrollada se muestra como fondo principal un mapa, en el cual podrán ser agregadas por el usuario distintas capas temáticas, con datos sobre la cobertura del suelo, la sequía actual, heladas y precipitación actuales, entre otras. Además, desde la aplicación se le permite al usuario obtener datos de índices históricos¹ sobre un par de coordenadas del mapa, como el NDVI (*Normalized Difference Vegetation Index*), NDWI (*Normalized Difference Water Index*), la sequía y heladas

¹La documentación de cada uno de los índices se puede encontrar en el servidor de CONAE [40].

históricas. Y se los muestra mediante gráficas que contienen los valores de los índices a lo largo del año. Esta información puede ayudar a tomar las decisiones para la prevención de un siniestro, permitiendo al usuario analizar los sectores en los que se encuentra una mayor posibilidad de que se generen focos de calor.

Entre las capas temáticas, se proporciona una capa de focos de calor, en donde se pueden ver puntos que representan incendios potenciales en las regiones marcadas. También se ponen a disposición capas que contienen información sobre los ríos permanentes y transitorios, sectores urbanos y vegetación, rutas, localidades y cuarteles de bomberos, entre otras. Lo cual permite analizar el sector y ayuda a llevar adelante el proceso de planeación del ataque, considerando el ambiente cercano al incendio.

Por otro lado como los combatientes tienen la necesidad de recolectar datos de campo para el seguimiento y reporte de la situación en el lugar del incendio, la aplicación móvil ofrece la posibilidad de crear puntos directamente en el mapa con el fin de guardar dicha información. Así por ejemplo pueden crear puntos para señalar lugares relevantes como huellas, tranqueras, puestos de bomberos, largo de llama, velocidad del viento y otras variables meteorológicas medidas en una cierta coordenada geográfica.

También permite el ingreso de líneas y áreas sobre el mapa, mostrando el valor de las distancias y superficies respectivamente a medida que se van creando, con el objetivo de delimitar zonas destacadas. Como por ejemplo, para marcar un área en la que se encuentra un cierto tipo de combustible o el tipo de incendio en el lugar, permitiendo tomar las acciones necesarias y, determinar el tipo de ataque y medio necesario para el combate. Al finalizar la extinción del fuego y entrar en la etapa de guardia de cenizas, se pueden agregar líneas o áreas que encierren las superficies quemadas para poder hacer un relevamiento posterior de la vegetación perdida.

Para una mejor organización de los datos ingresados a través de las figuras mencionadas, se permite la creación de capas por parte del usuario, con la finalidad de agrupar todos los puntos, líneas y áreas que compartan características en común. De esta manera, se puede crear por ejemplo, una capa que guarde los puntos donde se encuentran fuentes de agua y otra capa que guarde las áreas marcadas durante la guardia de cenizas. Estas capas pueden ser compartidas entre los usuarios dando lugar a una comunicación visual de la información recolectada, entre los combatientes y sus supervisores, facilitando la evaluación de las acciones a tomar.

2.3. Requerimientos Iniciales

Como se mencionó anteriormente, se requiere implementar una aplicación para el sistema Android. El resultado debe ayudar al personal encargado del manejo del fuego para mejorar la toma de decisiones, proveyendo los datos suficientes para la prevención y actuación frente a un incendio.

Al comienzo del proyecto fueron planteados una serie de requerimientos generales que se pretendían cumplir al desarrollar la aplicación. Ellos son:

- El área de estudio considerada se encuentra ubicada en el área correspondiente a la provincia de Córdoba de la República Argentina.
- Posibilidad de autenticación de usuarios en el sistema.

- Interfaces GUI diferenciadas para las fases PRE, CRISIS y POST de la emergencia.
- Acceso a Internet vía Wi-Fi o 3G. Considerando la posibilidad de soportar fallas en la red.
- Acceso a GIS online, adición de capas temáticas desde servicios web (WMS, WFS, WCS) o archivos locales.
- Ingreso de información al sistema: localización de huellas, tranqueras, localización de incendios activos y extinguidos, posición del personal, área quemada y progreso del incendio, fotos etiquetadas y georeferenciadas, y anotaciones varias. Operaciones del personal, operaciones de los aviones.
- Incorporación de variables medidas por el usuario relativas al comportamiento del fuego, por ejemplo: largo de llama en una dada posición a un dado tiempo, velocidad y dirección de avance del frente de fuego.
- Posibilidad de simulación de escenarios de incendios.
- Gestión de alertas sobre un área definida por el usuario, disparada por la presencia de un foco de calor o cuando el riesgo de incendio supera un umbral especificado, etc. Y con la posibilidad de trazar sobre el mapa líneas y polígonos, calcular distancias y superficies para determinar el largo de una línea, las áreas con propensión al incendio o áreas incendiadas.

A continuación se listan las capas temáticas públicas que se esperaban mostrar inicialmente en la aplicación:

En Pre Incendio:

- Capa topográfica.
- Meteorología. Obtenida del INTA.
- Índice de riesgo de incendio. El FFDI (*Forest Fire Danger Index*) y el FWI (*Fire Weather Index*).
- Cobertura de suelo.
- Combustibilidad.
- Áreas protegidas.
- Base histórica acumulada de incendios, focos de calor.
- Instituciones.
- Jurisdicciones.

En Crisis:

- Recursos hídricos.
- Vías de acceso.

- Pistas de aterrizaje.
- Recursos para el combate.
- Posición de personal.
- Focos de calor.
- MODIS² Near Real Time.

En Post incendio:

- Área quemada y cantidad de hectáreas quemadas.

Dado que el presente es un trabajo final de grado y por ende es reducido tanto en personal como en tiempo, se decidió seleccionar algunos de los requerimientos mencionados, que se consideraban factibles y aptos para el proyecto, y se les asignaron prioridades.

Además durante el proceso de implementación de la aplicación se fueron considerando otros requerimientos y se modificaron algunas de las prioridades planteadas, como se verá más adelante.

2.4. Herramientas utilizadas

Dada la gran variedad de herramientas disponibles para llevar adelante la implementación de la aplicación, se realizó un estudio para determinar cuáles se adaptan mejor para el proyecto. Se debió seleccionar la metodología de trabajo, el lenguaje de programación a utilizar, el tipo de base de datos para el dispositivo móvil y para el servidor implementado en conjunto con la aplicación, editores para la programación, bibliotecas externas y el ambiente de desarrollo.

2.4.1. Entorno de trabajo

Se decidió utilizar la plataforma Android, pues es una de las plataformas con mayor crecimiento del mercado, es menos restrictiva que otras plataformas nativas considerando la libertad que ofrece al desarrollar una aplicación, permitiendo que esté disponible para cualquier usuario y dispositivo físico. Lo cual resultó ser de utilidad para poder compartir y mostrar la aplicación en varios dispositivos, y así poder ser evaluada por varios usuarios. Además luego de investigar las posibilidades del desarrollo en Android, se llegó a la conclusión de que permite implementar todas las funcionalidades planteadas, que fueron necesarias para el desarrollo de esta aplicación como se podrá ver a lo largo del documento. Finalmente una de las razones que llevó en primer lugar a considerar la plataforma Android fue que ya se disponía de un dispositivo móvil con este sistema operativo, el cual sirvió para la realización de las diferentes pruebas de funcionalidad.

Al momento de comenzar el proyecto, el sistema Android había sido actualizado a la versión 4.4 KitKat, con nuevas funciones que facilitaron el desarrollo

²MODIS (*Moderate-Resolution Imaging Spectroradiometer*): Sensor a bordo del satélite Terra-Aqua.

de la aplicación. Por este motivo se decidió implementar teniendo en cuenta desde la versión mencionada a las siguientes.

Se trabajó sobre el sistema operativo Windows 8.1 de 64 bits. En un principio se consideró utilizar un *framework*, como IONIC [16] o Intel XDK [21] pero por cuestiones de una mayor familiaridad con el lenguaje de programación Java por parte de la alumna, se prefirió seguir con las instrucciones de los desarrolladores de Android y trabajar con el Entorno de Desarrollo Integrado (IDE) Android Studio [14]. Haciendo uso además, de un emulador de Android AVD (*Android Virtual Device*) para realizar las pruebas de funcionamiento en distintos dispositivos móviles.

Finalmente para la implementación de la aplicación web en el servidor REST (creado para comunicarse con la aplicación móvil), se utilizó el IDE para programación en Python, llamado PyCharm [33], el cual posee soporte para desarrollo con Django (elegido para la implementación de una aplicación web en un servidor REST como se verá más adelante).

2.4.2. Bibliotecas Externas

Ha sido necesario la utilización de algunas bibliotecas externas, para ayudar a realizar una implementación rápida y eficiente.

Se tuvieron en consideración cuatro bibliotecas con el fin de mostrar el mapa interactivo en la aplicación, ellas son: Google API V2 [17], ArcGIS [8], Openlayers [30] y Leaflet [24]. Se realizaron varias pruebas con cada una de ellas con el fin de elegir alguna basándose en la eficiencia, facilidad de uso, rapidez y la comunidad de seguidores.

Google API V2, tiene la ventaja de que se adapta fácilmente a Android ya que existen bibliotecas que ayudan a su incorporación y a la interacción con el sistema. La desventaja es que existen condiciones para su uso que restringen el intercambio de datos a una cantidad fija de peticiones al servidor, y dado que nuestra aplicación está dirigida a una gran cantidad de personal, y por lo tanto gran cantidad de peticiones, finalmente se descartó.

ArcGIS, posee varias ventajas dado que es una biblioteca muy utilizada en el ambiente de imágenes satelitales, pero no es gratuita y como en este trabajo no se disponen de ingresos, también se descartó.

Con Openlayers se realizaron pruebas exitosas. Tiene muy buena reputación y la ventaja de que es la elegida en la mayoría de proyectos donde se ven involucrados mapas. Tiene una gran cantidad de seguidores y una gran comunidad, pero para alguien que es principiante con la manipulación de mapas en aplicaciones, no se lo consideró muy amigable como Leaflet.

A continuación se muestra la lista final de las bibliotecas utilizadas:

- **Leaflet:** Openlayers tiene muchas ventajas y mayor cantidad de funcionalidades que Leaflet, pero se optó por elegir esta última biblioteca pues comprende las funcionalidades necesarias para nuestra aplicación, es amigable y sencilla de utilizar. Además posee varios *plugins* para ampliar la biblioteca, junto con una gran comunidad que proveen respaldo ante cualquier duda sobre su manejo y que además ayudan a la actualización continua del código.
- **Gson** [25]: Se utilizó para la conversión de objetos Java a código JSON [9] y viceversa. Se la utiliza junto a la clase de Android *SharedPreferences*

(explicada en la sección de almacenamiento de datos), para guardar los objetos que deben ser compartidos entre actividades³ de la aplicación.

- **AndroidPlot** [5]: Biblioteca de terceros, gratuita y de código abierto. Se utilizó para construir y mostrar gráficos de funciones implementadas en la aplicación.
- **jQuery** [22]: Se utilizó para poder manipular los documentos HTML creados para mostrar los mapas.

2.4.3. Servidores

Se hizo uso de un servidor geográfico que provee el servicio de mapas, generados a través de datos georeferenciados y almacenados en una base de datos. Aquellos elementos geográficos semejantes se agrupan en capas llamadas capas temáticas y se les asocian atributos convencionales que representan sus características. De este servidor se realizaron peticiones, por parte de la aplicación, para la obtención de capas temáticas públicas, como por ejemplo de meteorología, índice de vegetación, focos de calor, entre otras relacionadas al manejo del fuego y que son de utilidad para los usuarios. El servidor con el que se contó es propiedad de CONAE. El software que emplean es GeoServer⁴, de código abierto, instalado en un sistema LAMP (Linux, Apache [7], MySQL [29] y PHP). Implementa los estándares WFS, WMS, WPS y WCS [35], que permiten la obtención de datos geoespaciales del servidor. De ahora en adelante se utilizarán las denominaciones servidor geográfico, georeferenciado, de mapas o GeoServer indistintamente.

Además se realizaron peticiones al servidor web de la plataforma de mapas Mapbox [31], para la obtención de los mapas base que serán mostrados al usuario y sobre los cuales se agregarán las capas temáticas. Se eligió utilizar la plataforma Mapbox debido a que es de código abierto, gratuita y contiene un *plugin* que extiende funcionalidades de la biblioteca Leaflet, lo cual permite integrar esta última con los servicios y datos de Mapbox, dando la posibilidad de utilizarlo si fuera necesario en la implementación de la aplicación móvil.

Se implementó un servidor aparte, el cual tiene una relación directa con la aplicación, guarda los datos de los usuarios, incendios, objetos creados por el usuario, entre otros. Se tuvo a disposición un servidor físico con uso exclusivo para la aplicación implementada. Se montó una máquina virtual, utilizando el software «VMware Server» [39], en donde se instaló un sistema LAMPy (Linux, Apache, MySQL y Python [34]). Sobre él se instaló el *framework* Django [10] junto con la biblioteca DjangoREST [11], para llevar a cabo la comunicación con la aplicación móvil. Se eligió este *framework* pues además de ser amigable e intuitivo, se tenía experiencia previa de trabajo con el mismo en un curso de la carrera. Se utilizó «No-IP» [28] para la obtención de un DNS Dinámico y se realizaron las configuraciones pertinentes para poder acceder al servidor a través de una IP pública, desde la aplicación móvil.

³Una actividad de Android representa una unidad de interacción con el usuario, puede considerarse como una pantalla de la aplicación.

⁴GeoServer es un servidor de datos geográficos. Permite entre otras cosas, obtenerlos remotamente en el formato que se necesite como en imágenes estáticas, datos vectoriales, entre otros.

Se eligió emplear directamente el protocolo HTTP [4] (*Hypertext Transfer Protocol*) estándar para intercambiar datos entre la aplicación y el servidor, teniendo en cuenta el estilo de la arquitectura REST (*Representational State Transfer*) debido a lo siguiente. Se sigue el protocolo cliente/servidor sin estado, donde cada mensaje HTTP es independiente y contiene toda la información necesaria, evitando tener que guardar el estado de la comunicación en el cliente, ni en el servidor. Lo cual es una ventaja, pues en un incendio la comunicación con el servidor no se puede dar de forma permanente debido a posibles fallas de conectividad con la red. Además REST tiene una sintaxis universal para identificar cada recurso⁵, haciendo sencilla la navegación entre ellos y la utilización correcta de las operaciones estándar POST, GET, PUT y DELETE.

2.4.4. Almacenamiento de datos

Para el almacenamiento interno de datos en el dispositivo móvil, se utilizó una base de datos SQLite [37]. Se eligió SQLite, pues Android proporciona soporte para este tipo de bases de datos y posee implementados métodos que facilitan las consultas de creación o extracción de datos.

Para la lectura de información intercambiada entre el servidor REST y la aplicación Android a través de consultas HTTP, se utilizó el lenguaje JSON. Y para la lectura del archivo obtenido al realizar la petición «GetCapabilities» al GeoServer (explicada en secciones siguientes), se utilizó el lenguaje XML para obtener los datos de las capas públicas que se encuentran disponibles en el momento de la petición.

Se utilizó la clase «SharedPreferences» de Android, que permite compartir información dentro de la aplicación, guardando y obteniendo datos por medio de parejas formadas por una clave y un valor.

Para guardar recursos en el servidor REST, se optó por utilizar la base de datos MySQL de código abierto, por su facilidad de uso y, su velocidad y seguridad al realizar consultas a través de la web.

2.5. Metodología y proceso de trabajo

A la hora de llevar a cabo un proyecto, es muy importante tener en cuenta el proceso de trabajo, el tiempo disponible, el resultado que se quiere obtener y la capacidad del personal involucrado. En la actualidad, obtener un resultado rápido y un producto que se pueda ver, mostrar y utilizar, se ha vuelto crucial para el éxito. En consecuencia nacieron los procesos de desarrollo de software iterativos que intentan evitar las desventajas de las metodologías tradicionales.

Uno de ellos es la metodología ágil que se enfoca en el usuario y los resultados. Existen varias metodologías de este tipo pero todas ellas se guían por los mismos valores, definidos en el Manifiesto Ágil [26]. Se centran en el individuo y sus interacciones, más que en el proceso y las herramientas utilizadas; en desarrollar software que funcione en lugar de obtener una buena documentación; en la colaboración con el cliente y en responder a los cambios en lugar de seguir

⁵Llamaremos recurso en el servidor REST a elementos de información, que pueden ser accedidos utilizando un Identificador Universal de Recursos (URI)

una planificación.

Para llevar adelante el desarrollo del proyecto se siguió un proceso iterativo tomando como guía algunos principios de la metodología ágil. Entre ellos se encuentran, aceptar que los requerimientos cambien en cualquier etapa del desarrollo, realizar entregas tempranas y continuas de software y comunicar el progreso al equipo mediante reuniones cara a cara. Se aplicó este proceso, pues al ser un trabajo final de grado, se requiere completarlo en un tiempo limitado. Y si se implementa el proceso tradicional se corre el riesgo de que al final del desarrollo no se hayan implementado correctamente los requerimientos, y que el cliente quede insatisfecho teniendo que rehacer todo el proyecto, lo cual no es una opción posible. Además de esta manera los directores pueden estar involucrados en el proceso de trabajo y, pueden aconsejar y ver el resultado de cada entrega desde otro punto de vista que no haya sido considerado por la alumna. Por otro lado, con este se evitará la extensa documentación y un proceso demasiado complejo que conlleve mucho tiempo. En cambio, se podrá realizar el software en ciclos cortos que muestren un verdadero progreso.

Teniendo en cuenta que el proceso de implementación de la aplicación es realizado por una sola persona, los ciclos de trabajo se planificaron con una duración superior a la sugerida en el proceso iterativo, de aproximadamente un mes dependiendo de la complejidad de las tareas a realizar en cada uno. Por cada ciclo de trabajo finalizado, se realizó una reunión con el objetivo de observar y discutir las modificaciones realizadas. Y para plantear los pasos a seguir en el siguiente ciclo, tanto las mejoras a realizar en la aplicación, como las nuevas funcionalidades requeridas.

En la siguiente sección se desarrollan con mayor detalle los requerimientos y sus implementaciones en cada una de las versiones resultantes de los ciclos de trabajo.

Capítulo 3

Versiones de la aplicación

Siguiendo los puntos del proceso iterativo, antes de comenzar cada ciclo de trabajo, se tomó el tiempo necesario para discutir sobre los requerimientos a cumplir y funcionalidades de la aplicación que se agregarían o modificarían durante el mismo. Teniendo en cuenta luego las historias de usuario y los casos de uso creados, se continuó con la estimación aproximada de la duración del ciclo y de cada tarea a realizar. Luego del análisis y diseño correspondiente se pasó a la implementación y prueba de las funcionalidades finales. Al terminar cada ciclo se realizó una reunión entre los participantes del proyecto con el fin de observar el resultado obtenido y, discutir sobre los objetivos cumplidos y las modificaciones a realizar en el siguiente ciclo. Las versiones detalladas a continuación son el resultado de varias iteraciones del proceso de desarrollo, y serán presentadas al lector tomando la iteración en la cual se consideró un producto ejecutable y estable de acuerdo a las funcionalidades implementadas. A continuación se detallan las secciones que serán presentadas para cada versión.

En una primer instancia se analizan las funcionalidades que se requieren para la versión de la aplicación y se estudian los requisitos. En la sección de análisis, se incluyen los requerimientos generales, las historias de usuario y los respectivos casos de uso que especifican los requisitos planteados. En cada versión se representaron los casos de uso en una tabla en donde se describen los distintos escenarios posibles durante la interacción de los diferentes tipos de usuarios con la aplicación. Cada tabla contendrá:

- El nombre del caso de uso
- El actor primario, quien interactúa con el sistema cumpliendo los pasos del caso de uso, usualmente el usuario de la aplicación.
- Una pre-condición, que establece lo que se debe cumplir antes de comenzar con el primer paso del caso de uso.
- Un escenario normal, formado por una serie de pasos realizados por el actor primario. Y escenarios alternativos, formados por una serie de pasos realizados por el actor primario cuando se puede desviar del escenario normal.

- Una post-condición, que será el resultado esperado al cumplirse con todos los pasos detallados en el escenario normal. Determina el estado del sistema al terminar el caso de uso.

Cabe aclarar que los casos de uso fueron implementados por partes, como lo plantea el método iterativo; pero por motivos de simplicidad, en este informe se muestran los casos de uso completos a pesar de que no se hayan implementado todos los pasos en la versión que se los muestra. Es decir que si se comienza a plantear un caso en una de las versiones, puede suceder que se tengan pre-condiciones (como requerir el inicio de sesión en la aplicación) que no se hayan implementado todavía en la versión en la que es presentado el caso de uso.

Una vez se finaliza el análisis de los requerimientos de la versión en proceso, se prosigue con el diseño de la misma. En la sección de diseño, se presentan diferentes diagramas en UML que representan la arquitectura del sistema, que muestra cómo se relacionan los diferentes componentes del mismo y, la estructura de la base de datos que muestra cómo se relacionan los distintos objetos almacenados tanto en la aplicación móvil como en el servidor implementado, para cumplir con funcionalidades determinadas.

La arquitectura se diseñará analizando los componentes necesarios que deben desarrollarse o que serán empleados para cumplir con los objetivos, mencionados en secciones anteriores. El diagrama de la arquitectura se mostrará en la versión en la cual haya sido analizada la primera vez y será respetada para las versiones siguientes a menos que sea modificada. Y de la misma forma se mostrarán los diagramas de las bases de datos diseñadas.

Al finalizar con las fases anteriores, se comienza con la implementación de la aplicación siguiendo el análisis realizado. Se explicarán las distintas decisiones tomadas para llevar a cabo el desarrollo de la aplicación.

Por último se muestran los resultados obtenidos, tanto la interfaz gráfica de la versión como un resumen generalizado de las pruebas realizadas para corroborar el correcto funcionamiento de la aplicación y de las correcciones pertinentes que se realizaron de acuerdo al resultado de las pruebas.

3.1. Prototipos iniciales

En un principio se trabajó en el desarrollo de algunos prototipos, para tener una visión general de las funcionalidades que se podrían implementar, de acuerdo a los requerimientos iniciales planteados al momento de comenzar el proyecto.

Estos prototipos sirvieron para tener una primera visualización de cómo podría ser la aplicación y para saber si los requerimientos iniciales se habían interpretado correctamente.

Para una mejor visualización del prototipo, se utilizó la herramienta de escritorio *Justinmind Prototyper* [23], para poder así simular a través de una interfaz gráfica todas las funcionalidades que fueron posibles agregar. Como por ejemplo, ver una lista de capas que puedan ser seleccionadas, la diferenciación entre las etapas de un incendio, la interfaz de cómo se podrían crear puntos y áreas en el mapa, y el formulario para el inicio de sesión, entre otros.

3.1.1. Apreciación final y correcciones

Se realizó una reunión para entablar los requisitos de la aplicación que iban a ser implementados en general y los que debían ser implementados para el primer ciclo de trabajo. En ella se habló sobre las capas temáticas para cada una de las etapas del incendio, que serían obtenidas del servidor público de CONAE. Se dejaron en claro los destinatarios de la aplicación y se distinguieron los tipos de usuarios que serían aceptados por el sistema, y los requerimientos de cada uno. Estos últimos son expuestos en las secciones siguientes.

3.2. Versión 1

En esta primera versión se dedicó un tiempo al estudio de las diferentes herramientas a utilizar. Se tuvo como primer objetivo satisfacer el requisito de mostrar un mapa junto con las capas públicas. Se realizaron pruebas con las diferentes plataformas de mapas. Se trabajó con Openlayers, Google api v2 y Leaflet. Se realizaron ejemplos con cada uno, en donde se introdujo un mapa con cada plataforma en una página web, para luego colocarlo en la aplicación Android del dispositivo. Y luego de un análisis sobre el resultado se decidió trabajar con Leaflet.

3.2.1. Análisis

3.2.1.1. Requerimientos a satisfacer

Los requisitos planteados para completar en la presente versión son:

- Ver un mapa de la República Argentina, centrado en la provincia de Córdoba.
- Agregar capas temáticas públicas al mapa, obtenidas de un GeoServer.
- Ver GUI diferenciada en PRE CRISIS, CRISIS y POST CRISIS de un incendio.

3.2.1.2. Historias de usuario

A continuación se listan las historias de usuario, pensadas con el objetivo de cumplir los requisitos según cada uno de los tipos de usuario que utilizarán la aplicación.

Como usuario general quiero:

- Ver un mapa restringido a la República Argentina y centrado en la provincia de Córdoba, al entrar en la aplicación.
- Moverme por el mapa hacia los distintos puntos cardinales y no poder salir de los límites de Argentina.
- Realizar acercamiento y alejamiento en el mapa.
- Ver las capas públicas que tengo disponibles, luego de abrir un menú de capas.

- Seleccionar una capa en el menú, y que luego se muestre activada, y que aparezca en el mapa.
- Quitar la selección a una capa, y que luego se muestre desactivada, y que se deje de ver en el mapa.
- Ver los mapas de fondo que tengo disponibles abriendo un menú con capas activas, seleccionar uno y que se cambie el mapa de fondo.
- Ver las modalidades disponibles (PRE, CRISIS, POST), seleccionar una y, que en el menú de capas y en el mapa se muestren las capas asignadas a la modalidad elegida.
- Seleccionar una capa de una modalidad y que se vean solamente en el mapa de la modalidad respectiva.
- Al cambiar de modalidad, que se oculten del mapa las capas de la modalidad anterior.

3.2.1.3. Casos de uso

Los casos de uso que se desprenden de las historias de usuario anteriores son:

CASO DE USO: Desplazarse por el mapa.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario desliza su dedo por el mapa.	
2) El sistema actualiza el mapa desplazando la vista del mismo en la dirección que el usuario se deslizó.	2.a) Se alcanzó el límite de desplazamiento. No se puede seguir deslizando hacia esa dirección.
Post-Condición: Se actualizará la vista del mapa.	

CASO DE USO: Realizar acercamiento en el mapa.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario hace doble click en el mapa, o presiona en el mapa con dos dedos juntos y luego realiza movimiento de apertura.	
2) El sistema actualiza la vista del mapa y lo muestra con un nivel más de zoom.	2.a) Se alcanzó el máximo nivel de acercamiento. No se puede hacer más acercamiento.
Post-Condición: Se actualizará la vista del mapa.	

CASO DE USO: Realizar alejamiento en el mapa.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona en el mapa con dos dedos separados y luego realiza un movimiento juntando los dedos.	
2) El sistema actualiza la vista del mapa y lo muestra con un nivel menos de zoom.	2.a) Se alcanzó el mínimo nivel de alejamiento. No se puede hacer más alejamiento.
Post-Condición: Se actualizará la vista del mapa.	

CASO DE USO: Cambiar el mapa de fondo.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o entrado sin usuario.	
Escenario Normal	Escenarios Alternativos
1) El usuario abre el control de capas del mapa.	
2) El sistema muestra la lista de capas activas en el mapa y los tipos de mapas de fondo disponibles.	
3) El usuario selecciona un tipo de mapa de fondo.	3.a) El usuario presiona en el mapa. Se cierra el control de capas. 3.b) No hay otro tipo de mapa para seleccionar.
4) El sistema quita la selección de la capa base anterior y muestra la nueva capa seleccionada en el control de capas del mapa, y actualiza el mapa con la nueva capa base.	4.a) No se puede mostrar la capa por problemas internos. No se muestra el mapa.
5) El usuario, al terminar la selección, presiona en el mapa fuera del control de capas.	
6) El sistema cierra el control de capas y muestra el mapa con el tipo de capa base elegido.	
Post-Condición: Se mostrará la vieja capa como no seleccionada en el control de capas y se ocultará la capa en el mapa. Se mostrará la nueva capa en el mapa y se verá como seleccionada en el control de capas.	

CASO DE USO: Ver capas temáticas públicas disponibles.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o haber entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario abre el menú de capas.	
2) El sistema muestra una lista de capas separadas en grupos.	2.a) No hay capas para mostrar por problemas internos. Se muestran solamente los grupos sin las capas.
Post-Condición: Se mostrarán las capas temáticas públicas disponibles.	

CASO DE USO: Activar capa temática pública en el mapa.	
Actor Primario: Usuario.	
Pre-Condición: Debe haber capas para activar y se debe haber abierto el menú de capas.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona en una capa.	1.a) El usuario presiona el botón regresar atrás y no en una capa. Se cierra el menú de capas.
2) El sistema activa la capa, la muestra como seleccionada en la lista y la muestra tanto en el mapa como en el control de capas del mismo.	2.a) No se pudo mostrar la capa en el mapa por problemas internos. Se notifica al usuario.
3) El usuario, al terminar la selección, presiona botón regresar o sobre el mapa.	
4) El sistema cierra el menú de capas y muestra el mapa con la capa agregada.	
Post-Condición: Se mostrará la capa temática como seleccionada en el menú de capas y se la verá en el mapa.	

CASO DE USO: Desactivar capa en el mapa.	
Actor Primario: Usuario.	
Pre-Condición: Haber activado la capa y haber abierto el menú de capas.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona en la capa activa.	1.a) El usuario presiona el botón regresar atrás y no en una capa. Se cierra el menú de capas.
2) El sistema desactiva la capa, la muestra como no seleccionada en la lista y la oculta del mapa así como también del control de capas del mismo.	
3) El usuario, al terminar la selección, presiona botón regresar o sobre el mapa.	
4) El sistema cierra el menú de capas y muestra el mapa sin la capa seleccionada.	
Post-Condición: Se mostrará la capa como no seleccionada en el menú de capas y se la ocultará en el mapa.	

3.2.2. Diseño

3.2.2.1. Arquitectura

La arquitectura de la aplicación se basa en un diseño cliente/servidor. Modelo en el cual el cliente (en nuestro caso el teléfono móvil) realiza solicitudes a un servidor (un proveedor de recursos, en nuestro caso los servidores web) el cual procesa la solicitud y devuelve una respuesta.

Por un lado se tiene el móvil en el cual se ejecuta la aplicación Android, relacionada con su base de datos SQLite, para conservar el estado de la misma a largo plazo. Por otro lado el móvil debe interactuar con el servidor de Mapbox que contiene la biblioteca Leaflet. La comunicación se produce mediante el servicio WMS. La aplicación utiliza el código de Leaflet para realizar el pedido del mapa base a los servidores de Mapbox y mostrarlo en pantalla.

Por último se produce una comunicación con un servidor del tipo GeoServer, utilizando los protocolos WMS, WFS, WCS y WPS. La aplicación realiza las peticiones de las capas temáticas públicas, a este servidor. En la Figura 3.1 se pueden observar los componentes de la arquitectura y sus relaciones.

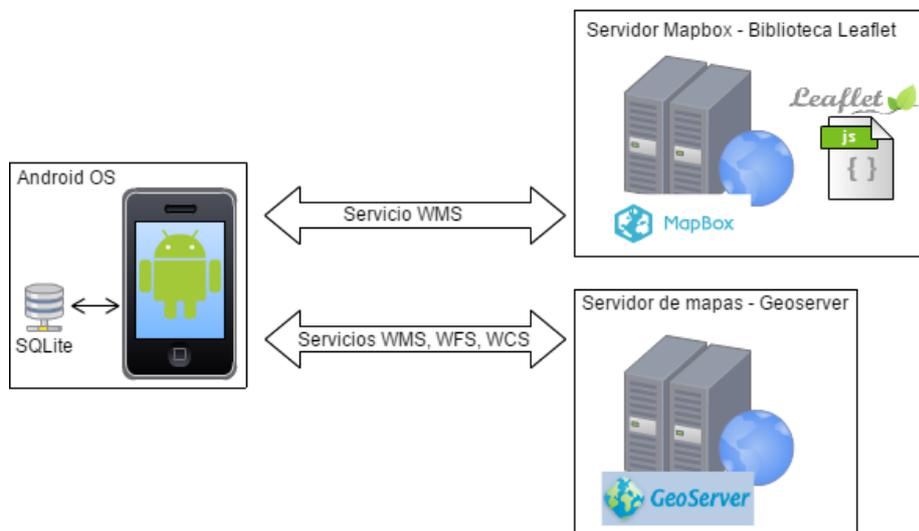


Figura 3.1: Arquitectura versión 1.

3.2.2.2. Base de datos

Como se muestra en la Figura 3.2, para la presente versión sólo se creó la tabla «ServerLayer». Que guarda los datos de las capas temáticas obtenidas del GeoServer público. La columna “id” guarda un identificador único con el cual se manipulará cada capa en la aplicación móvil. Las columnas “title” y “name” guardan el título que se mostrará al usuario y el nombre con el que se identificará la capa en el mapa respectivamente. “Submenu” guarda el grupo en el cual se incluirá y mostrará la capa al usuario. Finalmente las columnas “url” y “options” guardarán la dirección web y las opciones con las que se tomará cada capa del servidor geográfico.

ServerLayer	
*id	
*title	varchar(50)
*name	varchar(50)
*submenu	varchar(50)
*url	varchar(100)
*options	varchar(100)

Figura 3.2: Base de datos del móvil. Versión 1.

3.2.3. Implementación

El primer reto técnico que se presentó fue cómo incluir un mapa en la aplicación y poder comunicarse con el mismo para la agregación de capas temáticas. Partiendo de esta necesidad se realizaron los estudios mencionados anteriormente que llevaron a la elección de la biblioteca *Leaflet*. Como *Leaflet* está basado en *javascript* se optó por utilizar un archivo HTML e incluirlo en un objeto

«WebView»¹ de Android, en el cual se incluirá el código de la creación del mapa y su manipulación.

Para facilitar la creación de código *javascript*, a utilizar para la creación del mapa y las capas en el mismo, se decidió realizar un módulo aparte para tal fin. De esta manera si en un futuro se desea utilizar otra biblioteca distinta a *Leaflet*, se deberán hacer las modificaciones en este módulo.

Para la obtención de los mapas base, se optó por utilizar los disponibles en la plataforma proveedora de mapas *Mapbox*. Para esto se utilizó una cuenta privada de la alumna y una clave API, la cual debería ser cambiada en el futuro.

Se decidió tomar el menú lateral izquierdo² característico de las aplicaciones Android y utilizarlo para reunir todas las capas temáticas, y presentarlas en el mismo. A este menú lateral lo llamaremos menú de capas. En la presente versión se consideró la implementación de solamente un incendio y se dividió la interfaz en tres ventanas con un mapa cada una para seguir con el requerimiento de diferenciar la interfaz en las modalidades de PRE, CRISIS y POST CRISIS del incendio.

Las recomendaciones de diseño de los desarrolladores de Android es utilizar este menú lateral con información general que no cambie mientras se interactúa en una actividad de la aplicación, por lo tanto siguiendo con esta recomendación, al cambiar de una ventana a otra el menú debe permanecer con la misma información. Aquí se presentó una contradicción en nuestro diseño puesto que para cada modalidad se deben mostrar distintas capas temáticas. La solución que se implementó fue agrupar las capas en tres grupos correspondientes a cada modalidad. Así al seleccionar una capa por ejemplo de CRISIS, sólo se actualiza el mapa de la ventana de CRISIS y no se muestra en las otras ventanas. Una desventaja es que puede confundir al usuario dado que al estar visualizando una ventana y seleccionar una capa de otra ventana, no podrá ver la capa agregada hasta que el usuario abra la pestaña correspondiente. Luego de un re-diseño en la versión 2, se pudo solucionar este problema.

Esta versión puede considerarse el resultado de tres iteraciones, en las cuales primero se creó un mapa con una capa temática ya agregada, luego se creó el menú de capas para seleccionar las capas temáticas y agregarlas al mapa, y por último se crearon las tres ventanas que representan las etapas de PRE CRISIS, CRISIS y POST CRISIS de un incendio.

3.2.4. Resultados

3.2.4.1. Requisitos de instalación

Para poder utilizar esta versión de la aplicación, se deberán cumplir con los siguientes requisitos:

- El dispositivo móvil debe tener un sistema operativo Android, versión 4.4 KitKat o superior.
- Se debe tener conexión a Internet por Wi-Fi o 3G.

¹Una WebView de Android es una clase que despliega páginas web dentro de la aplicación.

²El menú lateral se puede ver en la dirección web <https://material.google.com/patterns/navigation-drawer.html#navigation-drawer-behavior>, junto con las recomendaciones de diseño de los desarrolladores de Android.

- Al instalar la aplicación, el usuario deberá aceptar el siguiente permiso para el funcionamiento correcto de la misma: Acceso a Internet sin límites.

3.2.4.2. Capturas

En las Figuras 3.3 y 3.4 se pueden observar imágenes de la interfaz resultante de la versión 1. Al iniciar la aplicación, se muestran los títulos de las tres ventanas *Inicio*, *Crisis* y *Guardia* (que representan la pre crisis, crisis y post crisis respectivamente, nombradas anteriormente), junto con el mapa de la ventana *Inicio* activa y su control de capas correspondiente en la esquina inferior derecha (Imagen 1). Al abrir el control de capas se pueden ver dos capas base del mapa: *Satelital* y *Estándar* seleccionada por defecto. En la Imagen 2 se muestra el resultado de seleccionar la capa *Satelital*.

En la Imagen 3 se puede ver el menú de capas luego de ser abierto, presionando en la flecha que se encuentra del lado derecho del título de la aplicación. Cuando se selecciona una capa de la lista de capas del *Inicio* (Imagen 4), la misma se agrega en el mapa y en el control de capas (Imagen 5). Al cambiar de modalidad por ejemplo a *Crisis* (Imagen 6) se puede observar que la capa seleccionada del *Inicio* no se encuentra seleccionada en el mapa de *Crisis*.

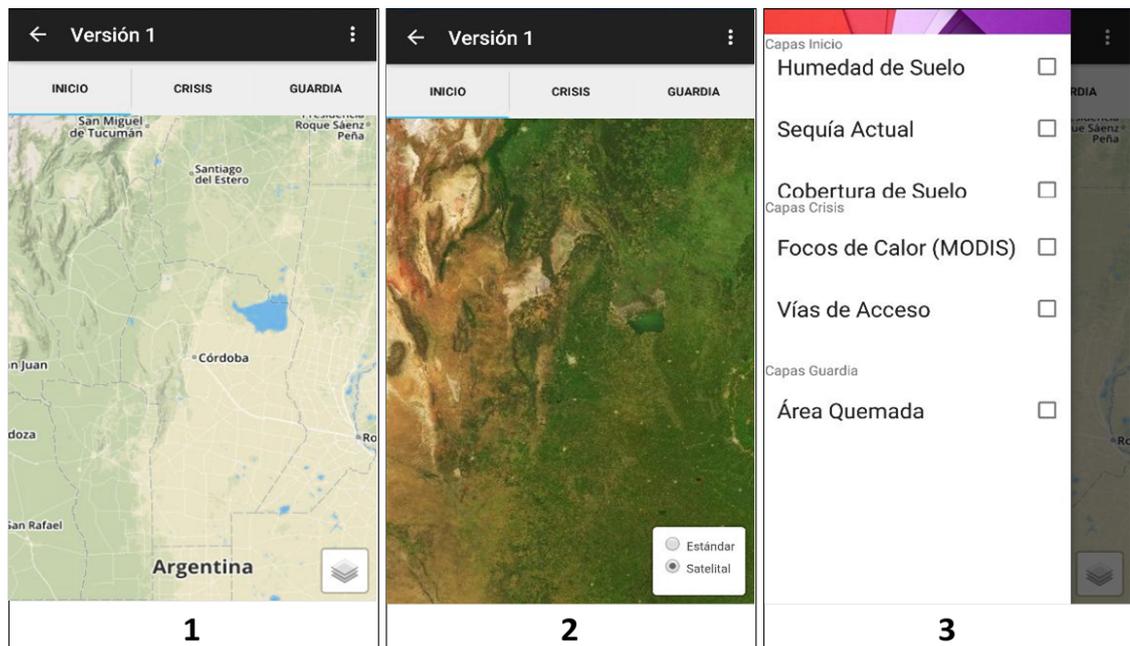


Figura 3.3: Capturas de Versión 1 (Continuación en Figura 3.4).

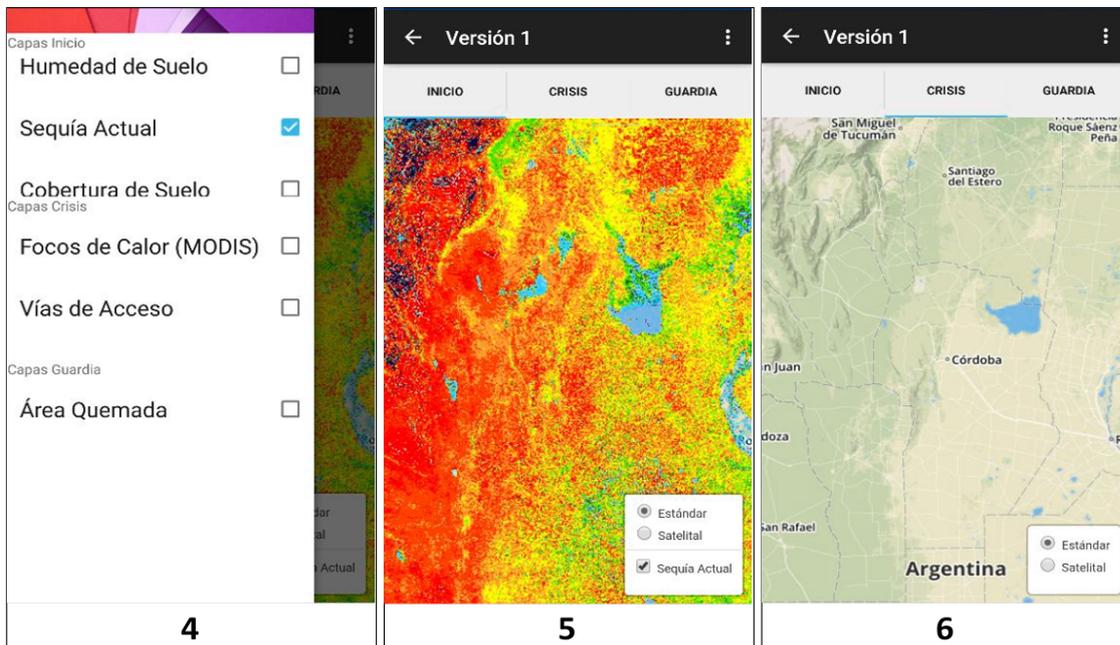


Figura 3.4: Continuación de Figura 3.3. Capturas de Versión 1.

3.2.4.3. Pruebas

Visualización de un mapa. Se creó un archivo HTML aparte, con un *script* para la ejecución de código Leaflet que realice la creación de un mapa. Además, se le agregó código para la obtención y agregación al mapa de una capa temática. Se abrió el archivo en un navegador web y se observó exitosamente un mapa con la capa agregada. Luego se procedió a la realización de una aplicación Android con una sola «WebView» y se incluyó el archivo HTML creado anteriormente. **Resultado:** Visualización y manipulación exitosa del mapa y de la capa agregada.

Agregar capa al mapa. Luego de seleccionar una capa temática del menú lateral, se verificó que el código *javascript* obtenido sea el correcto. Luego se lo envió a la «WebView» para ser ejecutado y se verificó que el código *javascript* sea el mismo. Además se verificó que la capa agregada se visualice en el mapa y sea la correcta, observando la capa directamente desde el GeoServer. **Resultado:** Se comparó exitosamente el código *javascript* y se visualizó la capa correcta en el mapa.

Inserción y actualización en la base de datos. Se hicieron las consultas de inserción y actualización de objetos a la base de datos y se verificó que fueran exitosas realizando manualmente la petición para obtener nuevamente los mismos objetos. Además se corroboraron los resultados utilizando la herra-

mienta de línea de comando que ofrece Android llamada *adb*³ (*Android Debug Bridge*) para la visualización de la base de datos de un emulador luego de ejecutar las consultas. **Resultado:** Comparación exitosa con los datos ingresados y actualizados.

Comprobación del estado de la aplicación. Para cada uno de los elementos de la aplicación, por ejemplo capas seleccionadas, ventana seleccionada, apertura de un menú, entre otros que muestren el estado de la aplicación en un momento dado, se fueron verificando que los valores sean los correctos a medida que se iba manipulando la aplicación en un emulador. **Resultado:** Finalmente se corroboró exitosamente el resultado esperado en cada interacción.

3.2.4.4. Apreciación final y correcciones

Se realizó una reunión con un bombero voluntario de la Defensa Civil de Río Ceballos para compartirle los objetivos de la aplicación y para que pueda aportar *feedback* al proyecto. Se habló sobre la actuación de los bomberos frente a un incendio, los protocolos que respetan y sobre la información que necesitan, y recaudan en cada uno (todo esto plasmado en el Capítulo 1 y en el Capítulo 2 Sección 2.2 del presente trabajo). También se habló de la información y herramientas que les serían de utilidad para un mejor manejo de sus actividades, dando sugerencias que fueron agregadas en el desarrollo de la aplicación. Entre ellas se encuentra la posibilidad de agregar información a través de figuras creadas directamente en el mapa, ya que en la actualidad utiliza *Google Earth* [19] al finalizar un incendio, para agregar datos que fueron anotados en papel durante el siniestro. De esta forma puede verlos nuevamente al presionar sobre las figuras, las cuales agrupa en capas teniendo en cuenta diferentes características y la etapa del incendio en la cual fueron tomados los datos.

Posteriormente, con la información proporcionada, al terminar la versión se realizó la correspondiente reunión de finalización del ciclo de trabajo, como se indica en el proceso iterativo.

En ella se plantearon distintos puntos observados entre los participantes del presente proyecto. Entre ellos se vio el problema de interactuar con tres mapas independientes (uno por cada ventana), pues no se sincronizaba la vista al cambiar de una ventana a otra. Esto lleva a que si el usuario ubica un punto en uno de los mapas, al cambiar de ventana tendría que volver a localizarlo. Además las pestañas muestran la misma interfaz y solamente se diferencian en las capas que se pueden observar. Por lo tanto se propuso que en lugar de tener tres mapas distintos, se muestre un solo mapa y se divida la aplicación en varios incendios que puedan ser accedidos por el usuario. Así al entrar en un incendio, el usuario podrá seleccionar entre las modalidades de dicho incendio sin cambiar de mapa. Esto simplificaría el diseño y siempre se tendrían las mismas funcionalidades, pero cambiaría la información que se pueda visualizar. Dejando en evidencia la ventaja de haber decidido llevar adelante un proceso de trabajo iterativo, debido a que en este punto se pudo reconocer a tiempo una falla de

³adb es una herramienta que permite la comunicación con un emulador o con un móvil real que posean el sistema operativo Android. Permite acceder y controlar el dispositivo Android desde una computadora [13].

diseño y se la pudo corregir en la sección siguiente, agregando además nuevos requerimientos para cumplir con el nuevo diseño.

Sumado a esto y teniendo en cuenta las sugerencias del bombero voluntario, también se discutió sobre la funcionalidad respectiva para permitir que los usuarios puedan agregar puntos, líneas y áreas con información geográfica para transferir información a otros usuarios. La comunicación sería bidireccional entre cada combatiente y su supervisor. Los combatientes enviarían información relevante solamente a su supervisor y el supervisor la enviaría solamente a todos sus combatientes. La idea planteada era que los usuarios puedan introducir las figuras directamente en el mapa y sean agregadas a una capa vectorial que las agrupe, para que sea la capa la que se comparte y así se puedan ocultar o mostrar al mismo tiempo todas las figuras que se relacionan. De esta forma el usuario también puede crear sus propias capas y agregarlas al mapa.

3.3. Versión 2

En la presente versión se realizó un nuevo diseño general teniendo en cuenta que ahora se crearán capas, figuras e incendios por parte de los usuarios. Se debieron reorganizar los elementos y se modificó la interfaz gráfica.

3.3.1. Análisis

3.3.1.1. Requerimientos a satisfacer

- Crear incendios con sus modalidades y al entrar en ellos ver las capas correspondientes a los mismos.
- Crear capas y, asignarlas a alguna modalidad de un incendio determinado.
- Crear figuras en el mapa (Puntos, Líneas y Áreas) y agregarlas a las capas creadas.

3.3.1.2. Historias de usuario

Como usuario combatiente quiero:

- Ver los incendios disponibles, creados por mi supervisor.
- Entrar en un incendio, y ver la modalidad CRISIS seleccionada por defecto.
- Al entrar y salir de un incendio, que se mantenga la vista del mapa anterior, tanto el desplazamiento como el acercamiento o alejamiento que se haya realizado.
- Ver las modalidades disponibles del incendio.
- Poder cambiar de modalidad en cada incendio.
- Ver las capas, en el menú de capas, correspondientes al incendio en una modalidad elegida.

- Al cambiar de modalidad quiero que se muestren, en el menú de capas, las capas pertenecientes a la misma y que se oculten las capas de la modalidad anterior.
- Al cambiar de modalidad quiero que se muestren, en el mapa, las capas que seleccioné de la misma y que se oculten las capas seleccionadas de la modalidad anterior.
- Ver una capa por defecto para las figuras a las cuales no quiera asignarlas a una capa en especial, tanto fuera de un incendio como dentro de cada incendio y modalidad.
- Crear una capa fuera de un incendio.
- Crear una capa en un incendio y modalidad elegida.
- Crear un punto y asignárselo a una capa.
- Crear una línea y asignársela a una capa.
- Crear un área y asignársela a una capa.
- Presionar en el mapa por un tiempo y que me aparezca una opción para poder crear un punto con esas coordenadas.
- Al presionar en un botón para crear una línea o un área, que se vea el mapa en modo de edición, mostrando solamente lo necesario para la creación de la figura.
- Para crear la línea, presionar en el mapa por cada punto que la forma y que se vaya viendo la línea marcada en el mapa. Y para finalizar la línea, presionar en el último punto agregado a la misma.
- Para crear un área, presionar en el mapa por cada punto que la forma y que se vaya viendo el área marcada en el mapa. Y para finalizar el área, presionar en el primer punto agregado a la misma.
- Tener la opción de cancelar la creación de un área o línea, y que al cancelar, se borren del mapa.
- Al terminar de marcar un punto, línea o área en el mapa, que se muestre un formulario para llenar con los datos de los mismos.
- Que los formularios para crear alguna figura, liste las capas a las que puedo asignarla.
- En el formulario a completar para cada figura, si no selecciono una capa en particular, que la figura sea asignada a una capa por defecto.
- Al crear una capa, si estoy fuera de un incendio, que se agregue al grupo *Mis Capas* y, que se vea siempre tanto fuera como dentro de cualquier incendio y modalidad
- Al crear una capa, si estoy en un incendio, que se agregue al grupo *Capas del Incendio* en la modalidad que estaba seleccionada al momento de crear la capa, y que no se vea fuera del incendio ni en otra modalidad del mismo incendio.

- Al crear una figura, si la capa a la cual fue asignada no se encuentra activada, que se active en el menú y se muestre en el mapa.
- Al activar una capa, que se muestren en el mapa todas las figuras asignadas a la misma.
- Al presionar sobre una figura en el mapa, que se muestre un cuadro con la información que ingresé en el formulario para su creación.
- Al presionar en un punto en el mapa, que se muestre el valor de sus coordenadas.
- Al presionar en un área en el mapa, que se muestre el valor de su superficie.
- Al presionar en una línea en el mapa, que se muestre el valor de su distancia.
- Que las capas y figuras que yo cree, tanto fuera como dentro de un incendio, pueda verlas yo solamente, es decir que no se compartan con los otros usuarios.

Como usuario supervisor quiero:

- Poder hacer lo mismo que el usuario combatiente.
- Tener una opción para crear un incendio.
- Al crear un incendio, que se muestre un formulario para llenar con los datos del mismo.
- Una vez creado el incendio, que se agregue a una lista con el resto de incendios creados.

3.3.1.3. Casos de uso

Los casos de uso que se desprenden de las historias de usuario anteriores son:

CASO DE USO: Ver capas disponibles.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o haber entrado sin usuario.	
Escenario Normal	Escenarios Alternativos
1) El usuario abre el menú de capas.	
2) El sistema muestra los grupos en los que se dividen las capas.	
3) El usuario presiona en un grupo.	3.a) El usuario presiona en el botón regresar atrás. Se cancela la operación.
4) El sistema muestra una lista de capas del grupo y oculta las listas de capas de los grupos no seleccionados.	4.a) No hay capas para mostrar, por problemas internos. Se muestran solamente los títulos de los grupos de capas.
5) Se repite desde el paso 3) para ver todas las capas disponibles en los distintos grupos.	
Post-Condición: Se mostrarán las capas disponibles.	

CASO DE USO: Crear capa nueva.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona el botón de crear capa nueva.	
2) El sistema muestra el formulario para llenar con los datos de la capa.	
3) El usuario completa el formulario y presiona el botón para aceptar el formulario.	3.a) El usuario presiona el botón regresar atrás. Se cancela la creación.
4) El sistema verifica los datos ingresados, crea la capa, la muestra seleccionada en la lista de capas y en el mapa.	4.a) Los datos ingresados no son válidos. Se le avisa al usuario para que ingrese nuevamente. 4.b) La capa no se pudo crear. Debido a problemas internos. Se notifica al usuario.
Post-Condición: Se agregará la nueva capa al sistema, se la mostrará en la lista de capas y en el mapa.	

CASO DE USO: Crear punto nuevo.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona por un tiempo sobre el mapa donde quiere crear el punto.	
2) El sistema muestra un ícono momentáneo en el lugar donde se presionó.	
3) El usuario presiona el ícono.	
4) El sistema muestra una opción para crear punto nuevo	
5) El usuario elige la opción de crear punto.	5.a) El usuario presiona el botón regresar atrás. Se cancela la creación y se sigue mostrando el ícono momentáneo.
6) El sistema muestra el formulario para llenar con los datos del punto, junto con la lista de capas a las cuales se podrá asignar el mismo.	
7) El usuario completa el formulario, elige una capa y presiona el botón para aceptar el formulario.	7.a) El usuario presiona el botón regresar atrás. Se cancela la creación y se sigue mostrando el ícono momentáneo.
8) El sistema verifica los datos ingresados, crea el punto, deja de mostrar el ícono momentáneo, marca como seleccionada la capa elegida y la muestra en el mapa junto con el punto agregado.	8.a) Los datos ingresados no son válidos. Se le avisa al usuario para que ingrese nuevamente. 8.b) El usuario no eligió una capa. Se le asigna una capa por defecto. 8.c) El punto no se pudo crear, debido a problemas internos. Se le notifica al usuario.
Post-Condición: Se agregará el nuevo punto al sistema, se mostrará en el mapa la capa a la cual pertenece el punto junto con el mismo y se verá seleccionada la capa en la lista de capas.	

CASO DE USO: Crear línea nueva.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón de crear línea nueva.	
2) El sistema prepara el mapa para que el usuario marque la línea.	
3) El usuario, seleccionando en el mapa, agrega un punto perteneciente a la línea a marcar.	3.a) El usuario presiona en el botón de cancelar. Se cancela la creación.
4) El sistema muestra el punto en el mapa y muestra la línea que se va formando.	
5) Se repite desde el paso 3) hasta que el usuario presione en el último punto marcado para terminar la línea.	5.a) No se ingresaron dos o más puntos para formar la línea. Se le informa al usuario para que siga ingresando puntos.
6) El sistema muestra el formulario para llenar con los datos de la línea, junto con la lista de capas a las cuales se la podrá asignar.	
7) El usuario completa el formulario, elige una capa y presiona el botón para aceptar el formulario.	7.a) El usuario presiona el botón regresar atrás. Se cancela la creación.
8) El sistema verifica los datos ingresados, crea la línea, marca como seleccionada la capa elegida y la muestra en el mapa junto con la línea agregada.	8.a) Los datos ingresados no son válidos. Se le avisa al usuario para que ingrese nuevamente. 8.b) El usuario no eligió una capa. Se le asigna una capa por defecto. 8.c) La línea no se pudo crear, debido a problemas internos. Se le notifica al usuario.
Post-Condición: Se agregará la nueva línea al sistema, se mostrará en el mapa la capa a la cual pertenece la línea junto con la misma y se verá seleccionada la capa en la lista de capas.	

CASO DE USO: Crear área nueva.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón de crear área nueva.	
2) El sistema prepara el mapa para que el usuario marque el área.	
3) El usuario, seleccionando en el mapa, agrega un punto perteneciente al perímetro del área a marcar.	3.a) El usuario presiona en el botón de cancelar. Se cancela la creación.
4) El sistema muestra el punto en el mapa y muestra el área que se va formando.	
5) Se repite desde el paso 3) hasta que el usuario presione en el primer punto marcado para cerrar el área.	5.a) No se ingresaron tres o más puntos para formar el área. Se le informa al usuario para que siga ingresando puntos.
6) El sistema muestra el formulario para llenar con los datos del área, junto con la lista de capas a las cuales se la podrá asignar.	
7) El usuario completa el formulario, elige una capa y presiona el botón para aceptar el formulario.	7.a) El usuario presiona el botón regresar atrás. Se cancela la creación.
8) El sistema verifica los datos ingresados, crea el área, marca como seleccionada la capa elegida y la muestra en el mapa junto con el área agregada.	8.a) Los datos ingresados no son válidos. Se le avisa al usuario para que ingrese nuevamente. 8.b) El usuario no eligió una capa. Se le asigna una capa por defecto. 8.c) El área no se pudo crear, debido a problemas internos. Se le notifica al usuario.
Post-Condición: Se agregará el área nueva al sistema, se mostrará en el mapa la capa a la cual pertenece el área junto con la misma y se verá seleccionada la capa en la lista de capas.	

CASO DE USO: Crear incendio nuevo.	
Actor Primario: Usuario Supervisor.	
Pre-Condición: Haber iniciado sesión como supervisor.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón de crear incendio nuevo.	
2) El sistema muestra el formulario para llenar con los datos del incendio.	
3) El usuario completa el formulario y presiona el botón para aceptar el formulario.	3.a) El usuario presiona el botón regresar atrás. Se cancela la creación.
4) El sistema verifica los datos ingresados, crea el incendio y lo muestra en la lista de incendios.	4.a) Los datos ingresados no son válidos. Se le avisa al usuario para que ingrese nuevamente. 4.b) El incendio no se pudo crear, debido a problemas internos. Se notifica al usuario.
Post-Condición: Se agregará el incendio nuevo al sistema y se lo mostrará en la lista de incendios.	

CASO DE USO: Entrar a un incendio.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona para listar incendios.	
2) El sistema muestra la lista de incendios disponibles.	2.a) No se pueden mostrar los incendios debido a problemas internos. No se muestran los incendios.
3) El usuario presiona en un incendio.	3.a) El usuario presiona el botón regresar atrás. Se cancela la operación.
4) El sistema entra en el incendio, muestra el mapa y las capas del incendio en una modalidad inicial.	
Post-Condición: Se ingresará en el incendio elegido y se mostrará el mapa con capas correspondientes al incendio en una modalidad inicial.	

CASO DE USO: Cambiar de modalidad.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión y haber entrado a un incendio.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona para listar las modalidades.	
2) El sistema muestra la lista de modalidades disponibles.	
3) El usuario presiona en una modalidad.	3.a) El usuario presiona el botón regresar atrás. Se cancela la operación.
4) El sistema entra en la modalidad, oculta las capas de la modalidad anterior, en el mapa y en el menú de capas, y muestra las capas del incendio en la modalidad elegida.	4.a) No se pueden mostrar las capas de la modalidad por problemas internos. No se muestran las capas.
Post-Condición: Se ingresará en la modalidad elegida, se modificará el menú de capas y el mapa, ocultando las capas de la modalidad anterior y mostrando las capas correspondientes al incendio en la modalidad seleccionada.	

CASO DE USO: Ver información de figuras en el mapa.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenario Alternativo
1) El usuario activa la capa con la figura que se quiere ver.	
2) El sistema muestra la capa con la figura en el mapa.	2.a) No se pudo mostrar la capa por problemas internos. Se notifica al usuario.
3) El usuario presiona en el mapa sobre la figura (punto, área o línea).	
4) El sistema muestra un cuadro en el mapa con los datos de la figura.	
5) El usuario presiona fuera de la figura para cerrar el cuadro.	
6) El sistema cierra el cuadro con los datos y muestra el mapa.	
Post-Condición: Se mostrarán los datos de la figura seleccionada.	

3.3.2. Diseño

Teniendo en cuenta también los requerimientos de manejo de usuarios, se diseñó el diagrama UML [38] de la Figura 3.5 que muestra cómo se relacionarían los distintos elementos de la aplicación. Basándonos en este diagrama se trabajó en la presente versión y en las siguientes. Ayudó a tener una visión general y, a obtener un diseño de la base de datos para el dispositivo móvil y el servidor (Este último expuesto en la sección ‘Versión 5’).

En el diagrama se distinguen por primera vez las capas creadas por el usuario («User Layer») y las capas temáticas, obtenidas de un GeoServer («Server Layer»). Se dividió el diagrama en tres módulos, para implementar cada uno por separado y luego poder relacionarlos. Por una parte un módulo que incluye los usuarios («User»), con los tipos de usuarios («User Type») y sus permisos («Permissions») (implementado en versiones siguientes). Por otro lado el módulo que incluye los incendios («Fire») y modalidades («Modality»), junto con las selecciones que los relacionan («Selection») y las capas tanto del usuario como del GeoServer. Por último el módulo que incluye las figuras («Figure»), tanto puntos («Point»), líneas («Line») y áreas («Area»). Los dos últimos módulos comenzados a implementar en la presente versión.

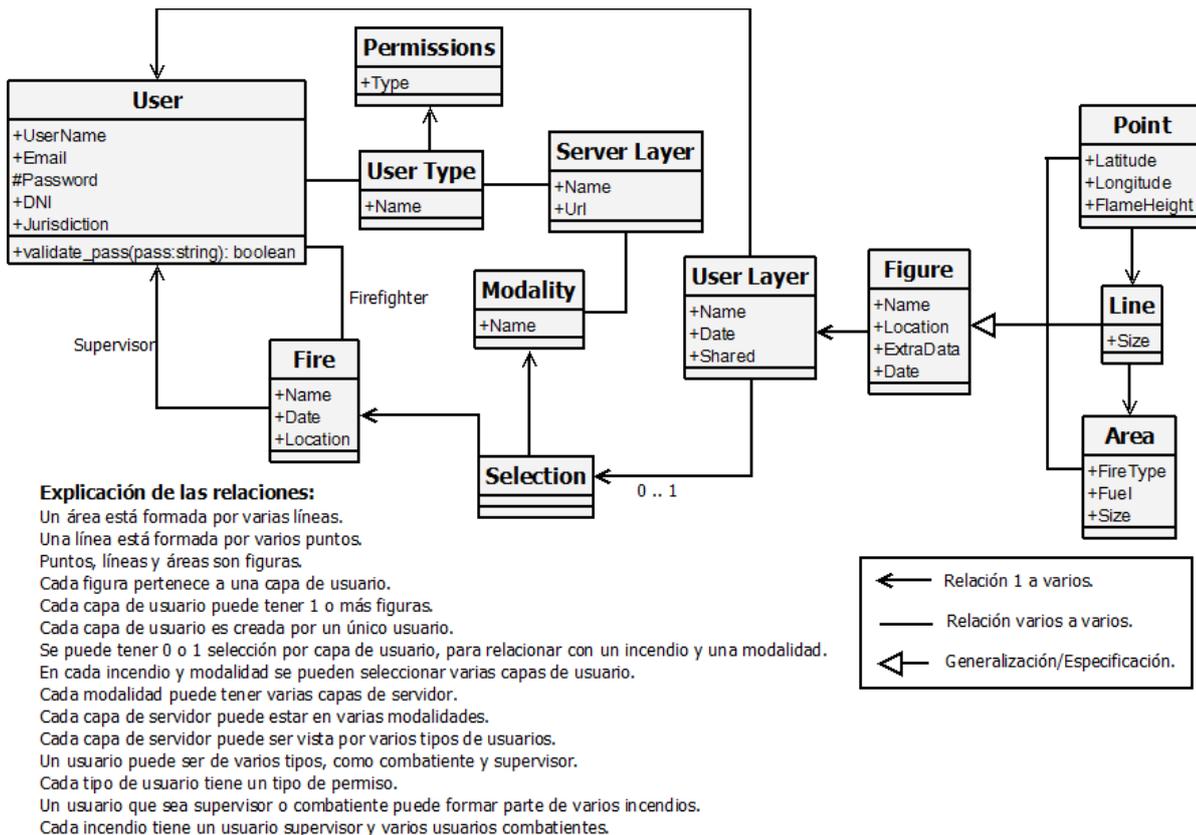


Figura 3.5: Diagrama de Clases Versión 2.

3.3.2.1. Base de datos

Luego de un análisis del diseño, se crearon las tablas para la base de datos del móvil, mostradas en la Figura 3.6. Las tablas «Point», «Line» y «Area» guardan los datos de puntos, líneas y áreas respectivamente creadas por el usuario. Las tablas «UserLayer» y «ServerLayer» guardan información de las capas creadas por el usuario y de las capas obtenidas del GeoServer respectivamente. La tabla «FireSelection» contiene los identificadores de las tablas «Fire» y «Modality» que guardan los datos de un incendio y una modalidad.

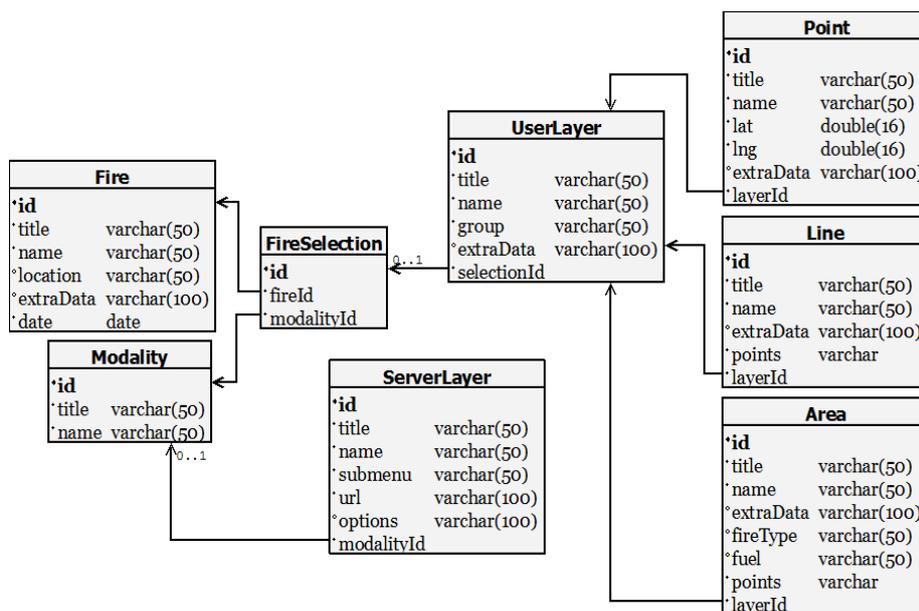


Figura 3.6: Base de datos del móvil para la Versión 2.

3.3.3. Implementación

La presente versión se realizó en dos iteraciones. La primera consistió en mostrar un prototipo de la interfaz e interacción posible del usuario al crear capas y figuras, y acceder a incendios. En ella se removieron las ventanas de la versión 1, dejando un solo mapa inicial y al acceder a un incendio se mostraron las modalidades de Crisis y Post Crisis relacionadas al mismo. Además si una capa se creaba dentro de un incendio, la misma solamente podía verse en ese incendio y no se podía ver fuera de él o en otro incendio.

Luego de observar el resultado de esta iteración, se realizó una reunión para discutir sobre la misma, dando como resultado el planteo final de los requerimientos y diagramas presentados anteriormente.

Dado que en esta versión existen varios estados de la aplicación en los cuales se pueden crear capas, tanto fuera como dentro de un incendio, y a su vez teniendo en cuenta las diferentes modalidades del incendio, se presentó el reto de organizar todas estas capas de manera que puedan ser accedidas fácilmente por el usuario. Para esto se plantearon varias opciones posibles y finalmente

se decidió presentarlas a todas en el menú de capas que se había creado en la versión 1 para las capas temáticas. Pero ahora se dividió el menú en diferentes grupos, que al ser desplegados muestran la lista de capas correspondiente. Los grupos son: *Capas Públicas* (las mismas que se mostraban en la versión 1), *Mis Capas* (creadas por el usuario fuera de un incendio) y *Capas del Incendio* (creadas por el usuario dentro de un incendio).

Esta decisión llevó a un proceso complicado debido a que el estilo de menú diseñado se tuvo que crear de cero, pues Android solamente provee un menú estático. Se tomó demasiado tiempo en la implementación del mismo, mucho más del propuesto para la iteración, por lo que se tuvieron que reorganizar los tiempos del ciclo. Finalmente se pudo realizar con éxito.

Además, para cada grupo de capas creadas por el usuario se optó por agregar una capa base por defecto, para darle la opción al usuario de guardar aquellas figuras que desee dejar sin clasificar.

Por otro lado, como ahora se tienen dos posibles estados en la aplicación (fuera y dentro de un incendio), al modificar el código se necesitaron enviar objetos compartidos de una actividad a otra de la aplicación (como capas en común, vista del mapa, entre otros). Una de las opciones consideradas para tal fin fue el uso de variables, pero debido a que esto puede llevar a fugas de memoria (en inglés *memory leaks*⁴), se optó por utilizar la clase «SharedPreferences» implementada por Android. Este se basa en almacenar datos mediante parejas clave/valor, donde el valor es de tipo primitivo. Como los datos a compartir entre las actividades de la aplicación son objetos complejos, se tuvo que recurrir a la biblioteca externa «Gson». La cual transforma los datos complejos a JSON y de esta forma se obtiene una cadena de caracteres que puede ser aceptada para guardar con la clase «SharedPreferences».

3.3.4. Resultados

Los requisitos de instalación para la presente versión continúan siendo los mismos que en la versión anterior.

3.3.4.1. Capturas

En la Figura 3.7 se puede ver la interfaz inicial de la aplicación en la presente versión. Al entrar en la aplicación se muestra el mapa en el estado de *Inicio* (que representa estar fuera de un incendio), junto con botones para la creación de áreas y líneas (Imagen 1).

Al abrir el menú de capas, ahora pueden observarse los grupos de capas separados en *Capas Servidor Público* (Son las mismas que en la versión anterior) y *Mis Capas* (Creadas por el usuario en el Inicio). Además se agregó el botón para la creación de capas (Imagen 2). En la Imagen 3 se ve desplegado el grupo *Mis Capas* y muestra por defecto la capa llamada *Figuras sin Clasificar*.

⁴Para ampliar la aclaración sobre la posible ocurrencia de fugas de memoria en Android, se recomienda el libro *Android Studio Cookbook* [1]

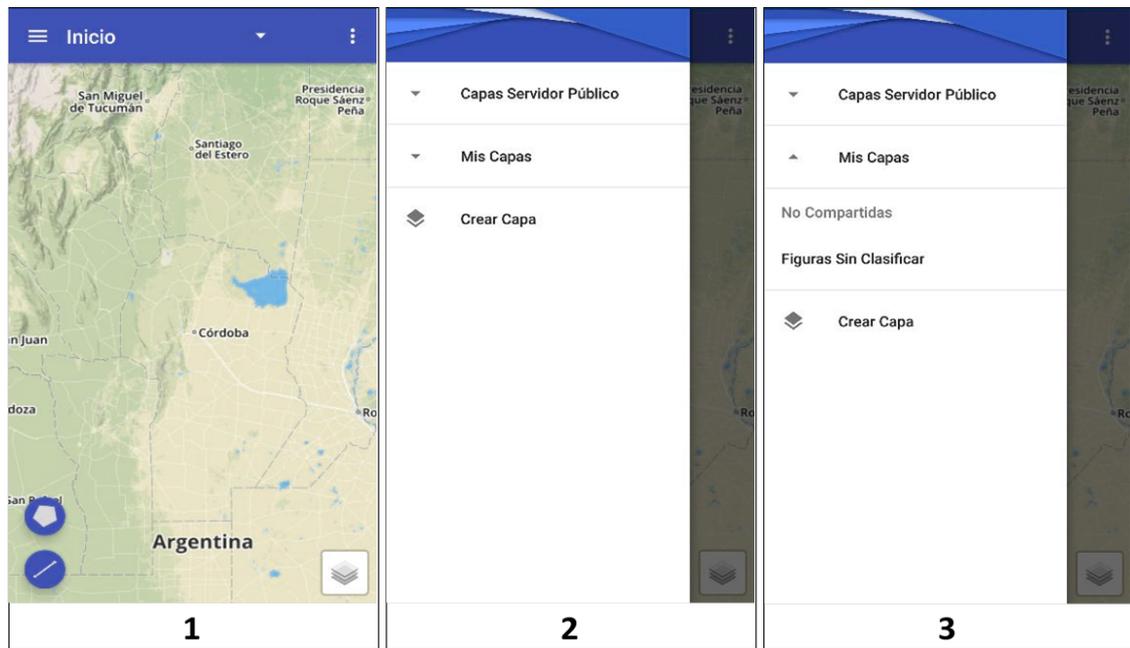


Figura 3.7: Iniciando la aplicación.

En la Figura 3.8 se puede ver la interfaz para la creación de capas. Cabe aclarar que solamente se muestran las pantallas para la creación de capas desde el *Inicio*, pero son exactamente las mismas para la creación de capas en un incendio y en cualquiera de sus modalidades, sólo cambia el menú en el que se la agrega.

Primero se debe abrir el menú de capas y presionar sobre el botón *Crear Capa* (Imagen 1). Luego se muestra una nueva pantalla con el formulario para llenar con los datos de la capa (Imagen 2). En la Imagen 3 se muestra el formulario relleno con los datos obligatorios y se presiona el botón *Crear Capa* del formulario para terminar la creación.

Una vez aceptado el formulario, se cierra y se muestra el menú de capas abierto con la nueva capa creada en el grupo correspondiente (Imagen 4), en el caso de ser creada en el *Inicio* se verá en el grupo de *Mis Capas* y si es creada desde un incendio, se verá en el grupo *Capas del Incendio*. Y como se puede ver en la Imagen 5 en ambos casos se activa la capa y se agrega en el mapa, apareciendo en el control de capas del mismo.

En la Figura 3.9 se muestra la interfaz para la creación de un punto. Para llevar a cabo esta acción, se debe mantener presionado sobre el mapa en el lugar donde se desea crear el punto, luego de unos segundos aparece un ícono rojo marcando el lugar seleccionado (Imagen 1). Se debe presionar sobre este ícono y se mostrará un cuadro con la opción para crear un punto (Imagen 2). Se presiona sobre el botón *Crear Punto* y se abre una pantalla con el formulario para su creación junto con una lista desplegable de capas en las que se lo puede agregar, con la capa *Figuras sin Clasificar* seleccionada por defecto (Imagen 3).

En la Imagen 4 se enseña otra posibilidad de creación de una capa. En la lista

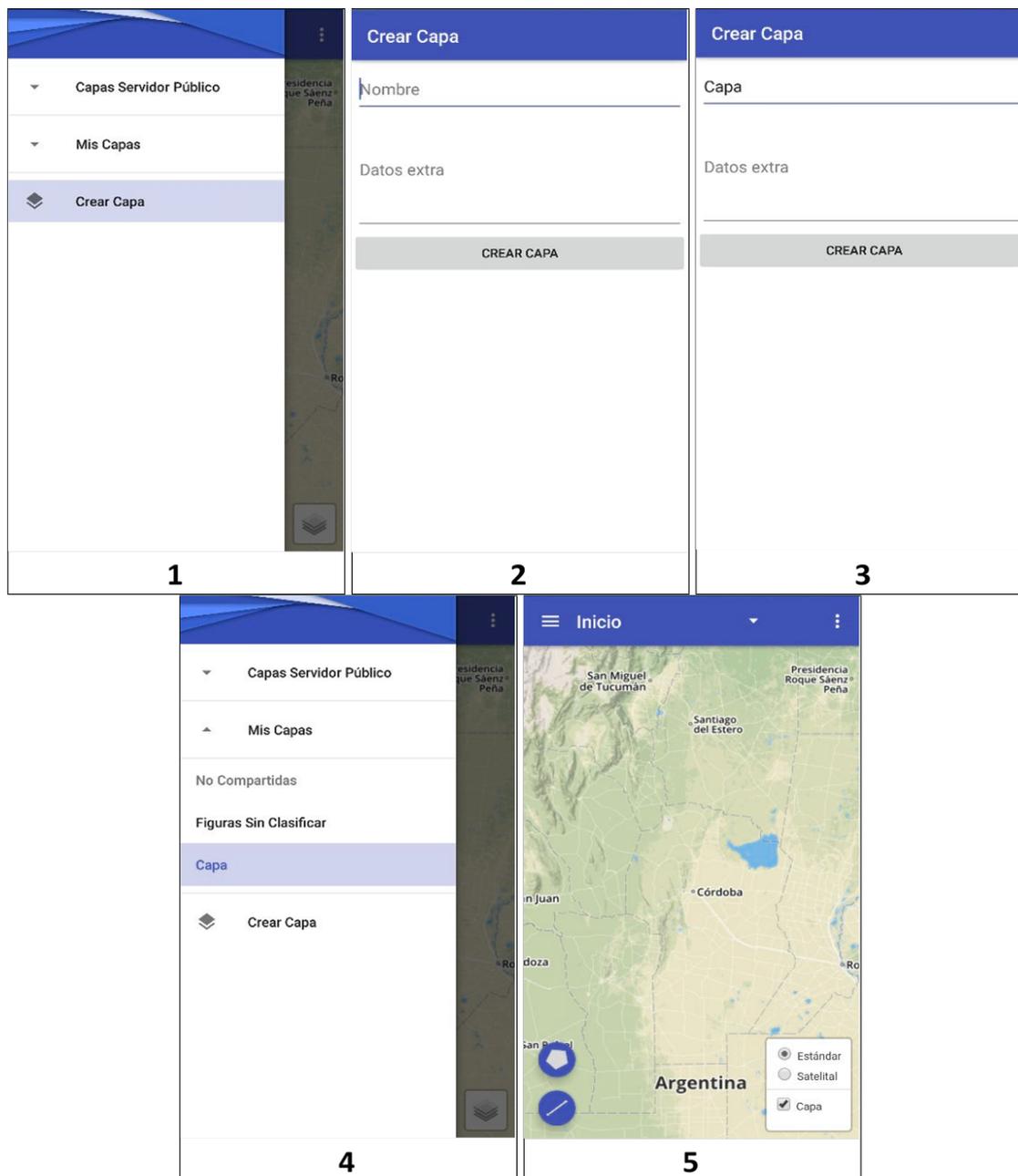


Figura 3.8: Creando Capa.

desplegable del formulario, se encuentra el botón *Crear Capa Nueva* que actúa como el botón del menú de capas mostrado anteriormente. Con la diferencia de que una vez que se completa el formulario para la creación de la capa, se vuelve a mostrar el formulario de creación de la figura, marcando como seleccionada en la lista desplegable la nueva capa. En este ejemplo se creó una capa con

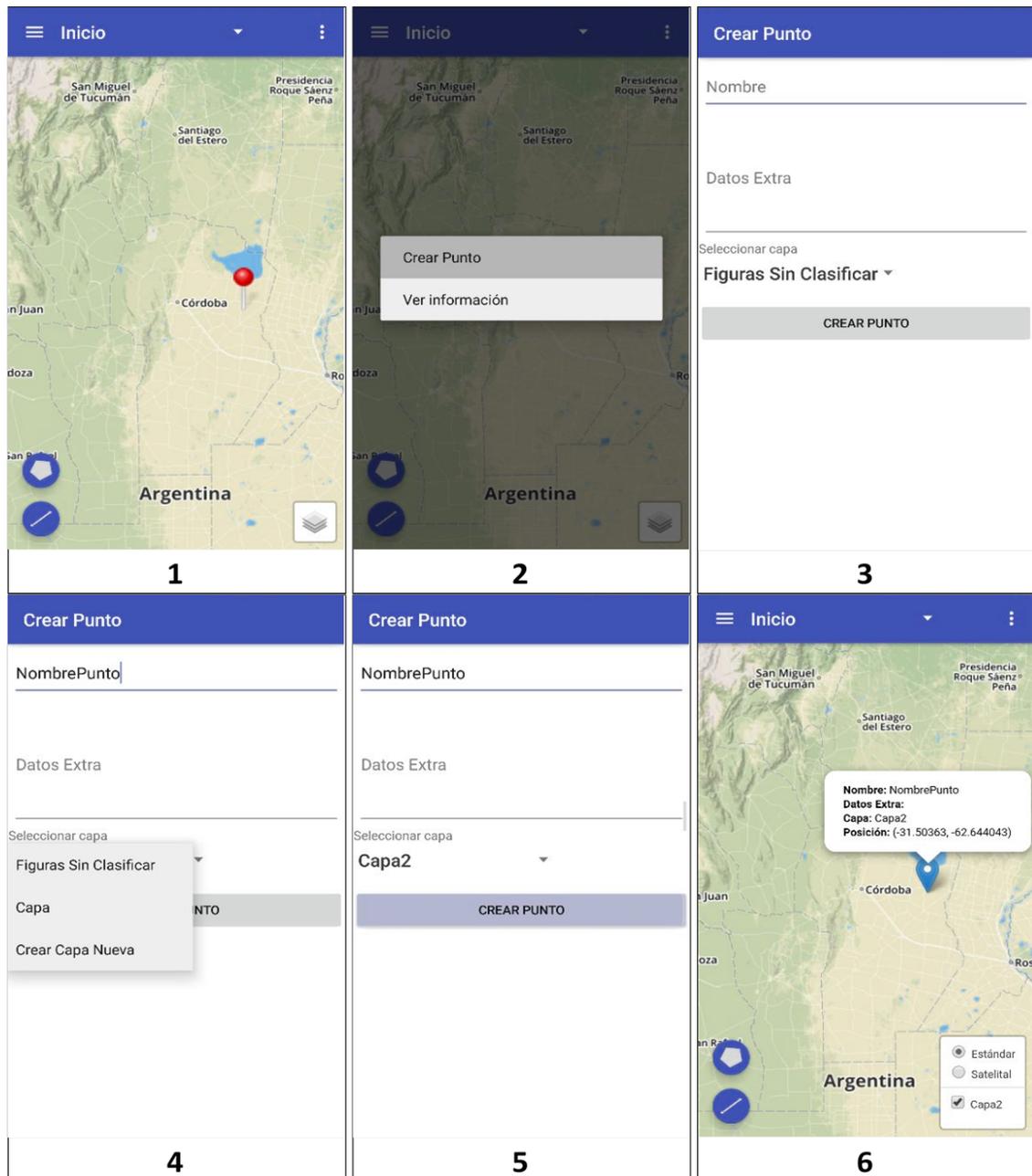


Figura 3.9: Creando Punto y viendo su información.

el nombre *Capa2*, se rellenó el formulario del punto con los datos obligatorios y se presionó el botón *Crear Punto* (Imagen 5). Como resultado se cierra el formulario y se muestra el mapa con la capa del punto activa en el control de capas. Además al presionar sobre el punto se despliegan los datos que le fueron asignados en el formulario (Imagen 6).

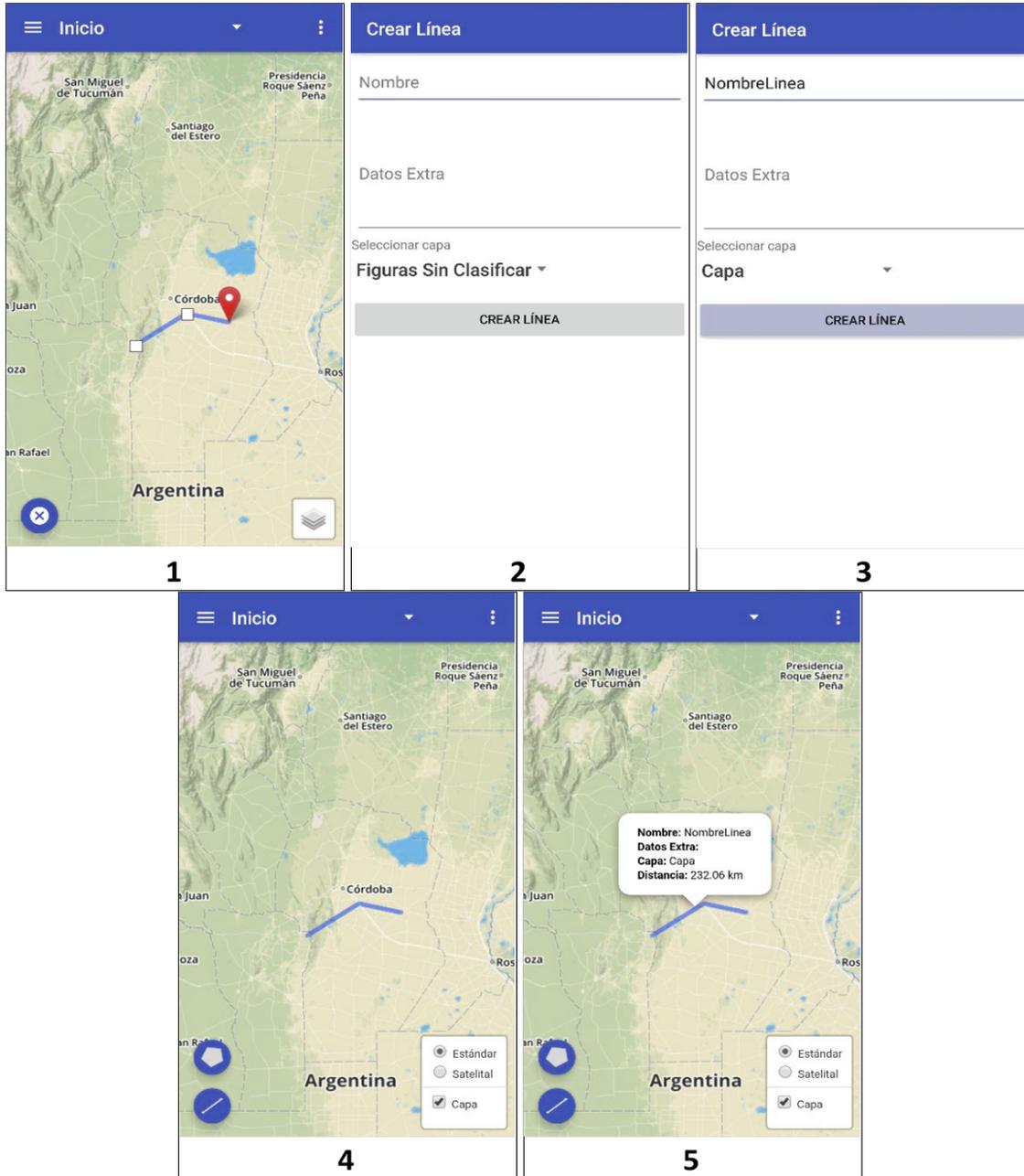


Figura 3.10: Creando Línea y viendo su información.

En la Figura 3.10 se muestra la interfaz gráfica para la creación de una línea. Se presiona en el botón del mapa con la imagen de una línea y el mapa se prepara en modo de edición para la introducción de la misma. Mientras se encuentre en estado de edición, se mostrará un botón con forma de cruz para

cancelar la creación en cualquier momento.

Para agregar una línea se debe presionar en el mapa e ir marcando los puntos que la forman. A medida que se van agregando los puntos, se va marcando la línea que los une. Una vez introducidos todos los puntos, para finalizar se debe presionar sobre el punto rojo que es el último que se creó (Imagen 1). Al hacer esto, se abre una pantalla con el formulario para completar con los datos de la línea, junto con la lista desplegable de capas a las cuales se la puede agregar (Imagen 2). En la Imagen 3 se puede ver el formulario rellenado con los datos obligatorios y con la capa de título *Capa*, creada anteriormente, seleccionada. Se presiona en el botón *Crear Línea* y se cierra el formulario, mostrando el mapa con la capa a la cual pertenece la línea, activa en el control de capas y la línea marcada en el mapa (Imagen 4). En la Imagen 5 se muestra el resultado de presionar sobre la línea, un cuadro con los datos ingresados en el formulario y el cálculo de la distancia total desde el primer punto agregado hasta el último.

En la Figura 3.11 se muestra la interfaz gráfica para la creación de un área. Se presiona en el botón del mapa con la imagen de un polígono y el mapa se prepara en modo de edición para la introducción del área. Mientras se encuentre en estado de edición, se mostrará un botón con forma de cruz para cancelar la creación en cualquier momento.

Para crear un área se debe presionar en el mapa e ir marcando los puntos que forman la misma. A medida que se van agregando los puntos, se va marcando el área rellenando el interior para distinguir su superficie. Una vez agregados todos los puntos, para cerrar el área se debe presionar sobre el punto rojo que es el primero que se creó (Imagen 1). Al hacer esto, se abre una pantalla con el formulario para llenar con los datos del área, junto con la lista desplegable de capas a las cuales se la puede agregar (Imagen 2).

En la Imagen 3 se puede ver el formulario rellenado con los datos obligatorios y con la capa *Figuras sin Clasificar* seleccionada por defecto. Se presiona en el botón *Crear Área* y se cierra el formulario, mostrando el mapa con la capa a la cual pertenece el área, activa en el control de capas y el área marcada en el mapa (Imagen 4). En la Imagen 5 se muestra el resultado de presionar sobre el área, un cuadro con los datos completados en el formulario y el cálculo de la superficie del misma.

En las Figuras 3.12 y 3.13 se muestra la interfaz gráfica para la creación de incendios y su acceso. Al presionar sobre la lista desplegable del título de la aplicación, se muestra la lista de incendios creados, por defecto vacía. Dentro de esta lista se encuentra el botón *Crear Incendio* (Imagen 1), que al ser pulsado abre una pantalla con un formulario para completar con los datos del incendio (Imagen 2).

En la Imagen 3 se muestra el formulario rellenado con los datos obligatorios y se presiona el botón *Crear Incendio* para aceptar el formulario. Una vez creado, se cierra el formulario y se muestra el mapa con la lista de incendios actualizada con el nuevo incendio (Imagen 4).

Para entrar en un incendio se debe seleccionar uno de la lista desplegable y como resultado, se abre una nueva pantalla (Imagen 5) con el título del incendio y *Crisis* como modalidad por defecto. Se muestra un mapa con la misma vista del mapa del *Inicio* y, los mismos botones para crear áreas y líneas.

Al abrir el menú de capas se puede ver el nuevo grupo de capas, *Capas del*

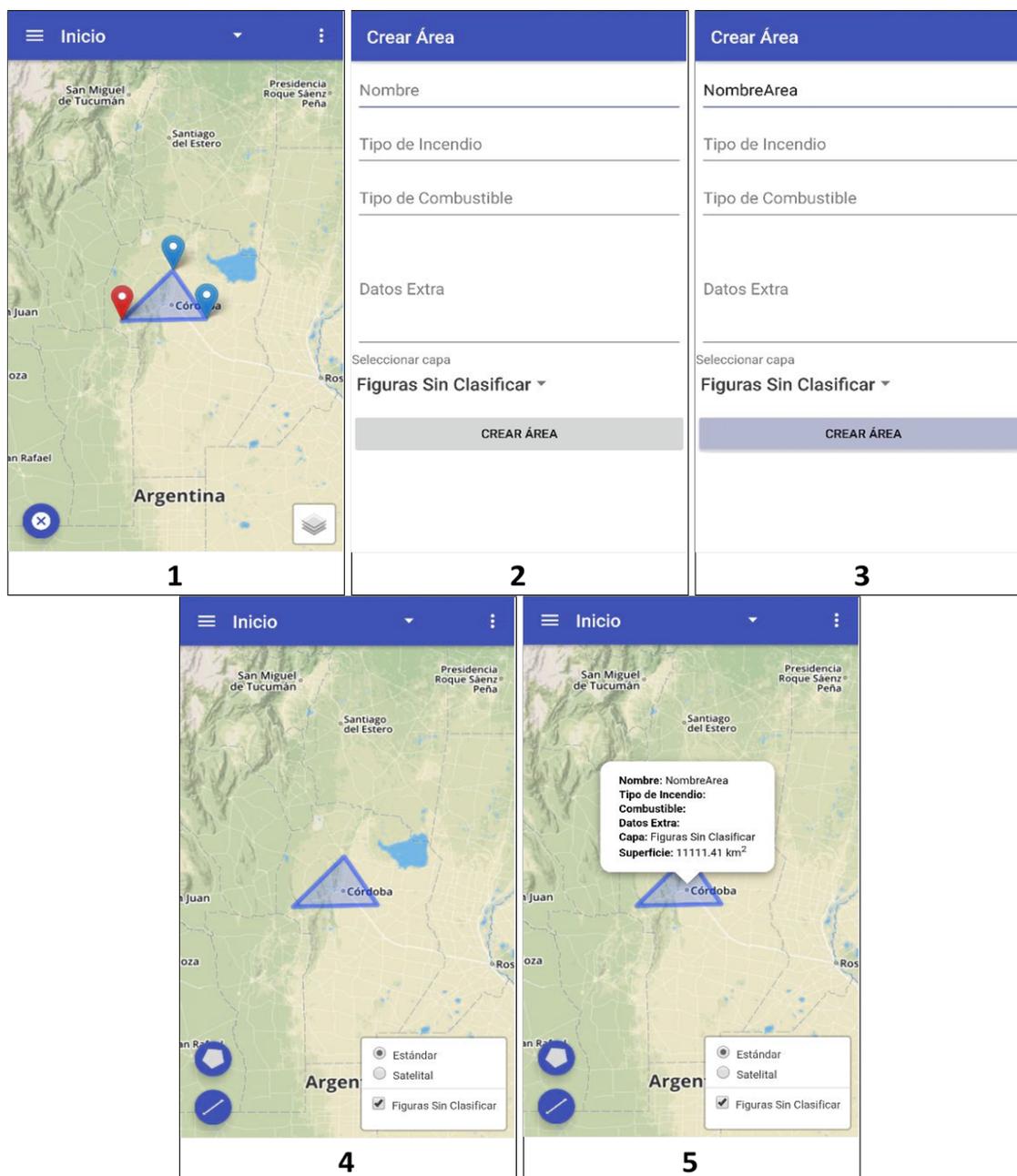


Figura 3.11: Creando Área y viendo su información.

Incendio (Imagen 6) y los dos grupos de capas anteriores intactos. Al desplegar el grupo de capas del incendio (Imagen 7) se muestran las capas del mismo en la modalidad seleccionada.

Para cambiar de modalidad en el incendio, se debe desplegar la lista de modalidades en la barra de título de la aplicación (Imagen 8). Al cambiar de

modalidad se actualiza el menú de capas y en el mapa se muestran activas solamente las capas de la modalidad actual. En cada una de las modalidades se muestra una capa por defecto, en *Crisis* la capa *Figuras sin Clasificar-Crisis* (Imagen 7) y en *Post Crisis* la capa *Figuras sin clasificar-Post Crisis*.

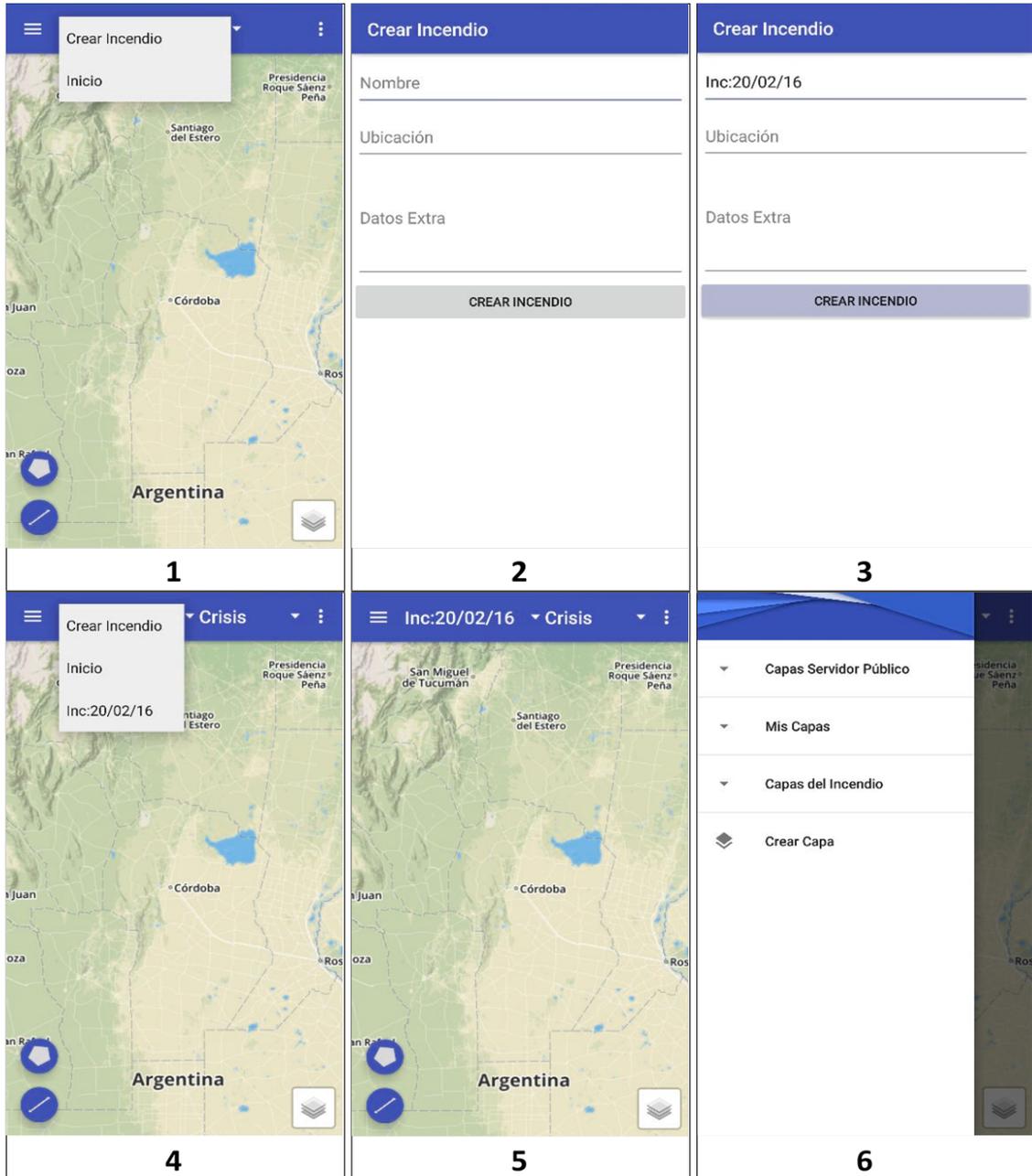


Figura 3.12: Creando Incendio y entrando en un incendio. (Continuación en Figura 3.13).

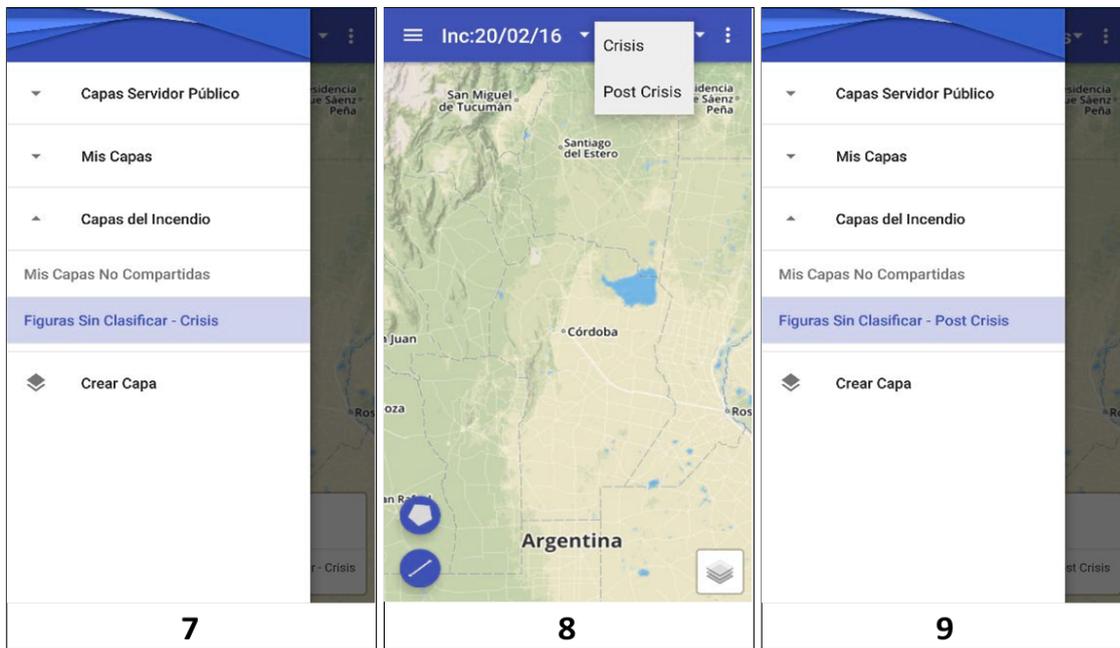


Figura 3.13: Continuación de Figura 3.12. Creando Incendio y entrando en un incendio.

3.3.4.2. Pruebas

Algunas de las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación fueron:

Asignación de figuras a una capa. Se verificó que las capas mostradas en el formulario de creación de una figura sean las correctas y luego se comprobó que la capa seleccionada sea la misma que existe en la base de datos. Luego se verificó que los valores ingresados en el formulario sean válidos y se hayan guardado exitosamente en la base de datos, realizando consultas a la misma. Además, también se comprobaron utilizando la herramienta *adb*. **Resultado:** Se obtuvieron exitosamente las capas correspondientes y se realizaron las asignaciones de datos correctamente.

Agregar figura al mapa. Se construyó un archivo HTML aparte con el código *javascript* para agregar una figura al mapa. Se corroboró que se agreguen exitosamente y luego se comprobó que en la aplicación se haya creado el mismo código al agregar un punto, una línea y un área en el mapa. Además se verificó que la figura agregada se visualice en el mapa y sea la correcta, observando los datos en el respectivo cuadro con la información de la figura. **Resultado:** Se comparó exitosamente el código *javascript* y se visualizó la figura correcta en el mapa.

Ingreso en un incendio. Se realizaron ingresos a diferentes incendios desde fuera y desde dentro de otro incendio. Se corroboró que el estado del sistema

sea el correcto en todo momento. Es decir, que se hayan creado correctamente las modalidades, capas, figuras, entre otras variables. **Resultado:** Se tuvieron que realizar varias pruebas y correcciones pues no se conseguía lo requerido y finalmente se logró un resultado exitoso.

Cambio de modalidad. Se realizaron cambios de modalidad en código y se corroboró que el estado de la aplicación sea el correcto, que las capas en el menú de capas hayan sido actualizadas y lo mismo con el mapa. **Resultado:** Actualización exitosa.

Actualización del menú de capas. Se realizaron creaciones de capas y se corroboró que las mismas se hayan incluido en el menú respectivo. Además se corroboró visualmente que las mismas se vieran en el grupo del menú correcto. **Resultado:** Actualización exitosa.

Inserción en la base de datos. Se realizaron creaciones de capas y se verificó que los datos sean los correctos, obteniéndolos nuevamente de la tabla de capas del usuario de la base de datos del móvil y comparando el resultado. Además, también se utilizó la herramienta adb para la visualización de la base de datos de un emulador ejecutando las pruebas. **Resultado:** Comparación exitosa con los datos ingresados y actualizados. La prueba se realizó para cada una de las tablas y objetos incorporados en la versión (capas, incendios, puntos, etc). Culminando con una comparación exitosa en todos los casos.

Comprobación del estado de la aplicación. Para cada uno de los elementos, por ejemplo capas seleccionadas, incendio y modalidad seleccionados, apertura de un menú, entre otros que muestren el estado de la aplicación en un momento dado; se fueron comparando sus valores con valores esperados a medida que se manipula la aplicación. **Resultado:** Finalmente se corroboró exitosamente el resultado esperado para cada interacción.

3.3.4.3. Apreciación final y correcciones

Cabe destacar que en la versión realizada, se planteó y se comenzaron a implementar las funcionalidades de eliminación de capas e incendios. Sin embargo debido a que esto agregaría una complejidad extra que no aportaría una variación considerable al procedimiento realizado en la creación de la aplicación, se decidió descartar la posibilidad de eliminar cualquier tipo de objetos del sistema de este proyecto. Dejando la posibilidad de que se pueda incluir la funcionalidad en versiones futuras.

Si se desea borrar los datos de la aplicación, se debe ir a la configuración de la misma y borrarlos. Esto hace que se vuelva al estado en el que se encontraba la aplicación al ser instalada por primera vez, borrando todos los incendios y todas las capas con sus figuras y otros datos creados por el usuario.

En las reuniones se revisó esta versión completada y se dio el consentimiento para seguir avanzando con otras funcionalidades. Se dieron las pautas para la implementación de una nueva funcionalidad prevista en un principio. Poder ver datos de índices históricos de un punto seleccionado en el mapa, como se verá en la siguiente sección.

3.4. Versión 3

3.4.1. Análisis

3.4.1.1. Requerimientos a satisfacer

- Ver gráficos de índices históricos⁵ (NDVI, NDWI, Sequía y Heladas) de un punto en el mapa.
- Ver ubicación del móvil en el mapa y una línea de seguimiento al cambiar de ubicación.

3.4.1.2. Historias de usuario

Como usuario general quiero:

- Presionar en el mapa por un tiempo y tener la opción de ver información de datos históricos en las coordenadas seleccionadas.
- Ver una lista de los tipos de índices históricos que puedo seleccionar.
- Seleccionar un tipo de índice y que me aparezca un gráfico con los valores históricos marcados.
- Que en el gráfico se pueda ver una curva con los valores del índice seleccionado, en relación a cada mes del año.
- Al cerrar el gráfico, poder seleccionar otro índice para ver sus datos.

Como usuario registrado quiero:

- Al tener activado el GPS que se muestre un punto en el mapa que señale mi ubicación.
- Al presionar el punto de mi ubicación, ver las coordenadas del mismo.
- Al moverme y tener la ubicación activada, que se marque una línea de seguimiento de mis movimientos.
- Poder presionar un botón que me lleve a mi ubicación en ese momento, es decir, que traslade la vista del mapa y haga un acercamiento a mi ubicación.
- Al presionar por un tiempo en el punto de mi ubicación, tener la opción de agregar el punto a una capa y la opción de ver datos históricos del mismo.

3.4.1.3. Casos de uso

Los casos de uso que se desprenden de las historias de usuario anteriores son:

⁵La documentación de cada uno de los índices se puede encontrar en el servidor de CONAE [40]

CASO DE USO: Ver datos históricos de un punto.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o haber entrado sin usuario.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona por un tiempo sobre el mapa en el punto que quiere ver la información.	
2) El sistema muestra un ícono momentáneo en el lugar donde se presionó.	
3) El usuario presiona el ícono.	
4) El sistema muestra un cuadro con el botón de ver datos históricos.	
5) El usuario presiona sobre el botón.	5.a) El usuario presiona el botón regresar atrás. Se cancela la operación y se sigue mostrando el ícono momentáneo.
6) El sistema muestra la longitud y latitud marcada, junto con botones para cada tipo de índice de datos históricos.	
7) El usuario presiona sobre el botón del índice que desea ver los datos.	7.a) El usuario presiona el botón regresar atrás. Se cancela la operación y se sigue mostrando el ícono momentáneo.
8) El sistema procesa los datos mientras muestra una barra de progreso. Al terminar habilita un botón para ver el gráfico.	8.a) No se pudieron encontrar datos. Debido a falta de conectividad a Internet. Se le notifica al usuario.
9) El usuario presiona sobre el botón de ver gráfico.	9.a) El usuario presiona el botón regresar atrás. Se cancela la operación y se sigue mostrando el ícono momentáneo.
10) El sistema muestra el gráfico con los datos históricos.	
Post-Condición: Se mostrará un gráfico con los datos históricos de un índice seleccionado para un punto en el mapa.	

CASO DE USO: Ver ubicación y línea de seguimiento.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona en el botón para activar el rastreo del móvil.	
2) El sistema muestra un cuadro de diálogo para que el usuario active el GPS.	2.a) El GPS ya está activado se salta al paso 6)
3) El usuario presiona el botón para dirigirse a la configuración del GPS.	3.a) El usuario presiona botón regresar atrás. Se cancela la operación.
4) El sistema abre la configuración de GPS.	
5) El usuario activa el GPS y presiona botón regresar atrás.	5.a) El usuario presiona el botón regresar atrás sin activar el GPS. Se cierra la configuración de GPS y se cancela la operación.
6) El sistema muestra un punto en el mapa en la ubicación del móvil, y actualiza el punto al cambiar de ubicación cada 3 segundos. Muestra una línea que va uniendo los puntos a medida que son actualizados.	
Post-Condición: Se irán mostrando puntos en el mapa a medida que se actualiza la ubicación del móvil y se unirán los puntos con una línea de seguimiento.	

3.4.2. Diseño

Para la presente versión no fue necesaria la modificación de la base de datos. Por lo cual se conserva el diseño de la versión anterior.

3.4.3. Implementación

La versión fue desarrollada en dos iteraciones. Se presentó una versión intermedia con la funcionalidad de ver índices históricos implementada, con el objetivo de asegurar la comprensión correcta de los requerimientos. Se debieron realizar algunas correcciones y luego se presentaron al terminar la segunda iteración, junto con la funcionalidad de ubicación del usuario, dando como resultado la presente versión.

Para obtener los datos de los índices históricos se tuvo que decidir el método a utilizar para realizar las peticiones al servidor de capas. Una de las opciones discutidas fue la utilización de llamadas AJAX [6] haciendo uso de la biblioteca Leaflet para peticiones WFS, pero esto causaba problemas de cruce de dominios

(*Access-Control-Allow-Origin*) y para solucionarlo se tenía que permitir acceso general a cualquier dominio en el proxy del GeoServer, ya que la aplicación se ejecuta en varios dispositivos y no contiene un identificador para que pueda ser identificada por el mismo. Habilitar el acceso general causa riesgos de seguridad respecto a la confidencialidad e integridad de los datos del GeoServer. Por lo que esta opción se descartó y se decidió hacer uso de peticiones HTTPRequest con direcciones URI públicas, para evitar los problemas de cruce de dominios utilizando un script CGI (*Common Gateway Interface*) que restringe el acceso a los datos del GeoServer.

Para comprensión del lector, a continuación se explica con brevedad el protocolo de comunicación con el servidor geográfico, que se utilizó para la obtención de los datos mencionados.

Del lado del servidor se encuentra una capa a la que llamaremos grilla, formada por celdas o polígonos, que juntos abarcan toda la región Argentina y son usados para ubicar rápidamente el trozo de una capa temática que corresponda a la zona deseada. En la Figura 3.14 se puede ver la grilla completa (Imagen 1) y la capa NDVI seleccionada como ejemplo, sólo en los polígonos que abarcan la zona de la provincia de Córdoba (Imagen 2). Dichos polígonos llevan el nombre «h12v11» y «h12v12», y son con los que se trabajó para restringir la búsqueda de datos históricos y capas públicas, sólo a la provincia de Córdoba.

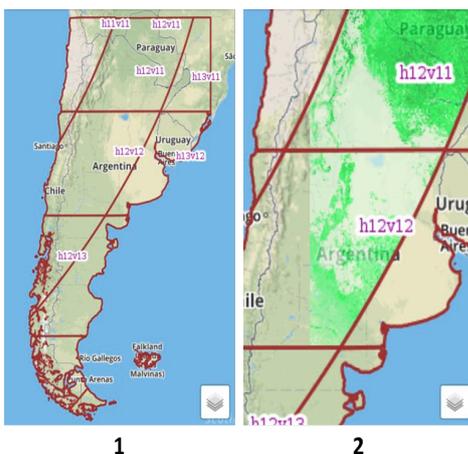


Figura 3.14: Grilla en mapa para consulta de datos.

En el servidor geográfico se genera una capa promedio por mes para cada índice histórico. Por lo tanto se tiene a disposición, doce capas por índice. Además teniendo en cuenta la grilla mencionada, las capas se dividen en varios trozos, uno por cada celda de la grilla.

Idealmente se podría realizar una sola consulta para cada índice histórico y que el GeoServer envíe todos los datos de las variables juntos en una lista, y así solamente realizar una consulta HTTP ahorrando recursos, y haciendo más liviana la aplicación. Lamentablemente la versión utilizada del protocolo WPS no permitía enviar de esta manera los datos, por lo que se tuvo que realizar una petición por cada variable. Es decir se realizaron las doce peticiones por cada índice.

Además de la carga que se le suma a la aplicación al llevar a cabo tantas consultas, se debieron realizar operaciones del lado del cliente para encontrar el cuadrante al cual corresponde el punto consultado, por otro lado los datos obtenidos no se encontraban en la escala del índice en cuestión (por ejemplo para el NDVI que sus valores van de -1 a 1, se obtenían datos de cinco cifras), por lo que se tuvieron que realizar más operaciones por cada valor para llevarlo a la escala deseada. Para actualizaciones futuras sería ideal poder realizar todas estas operaciones directamente en el GeoServer.

Para la creación de las gráficas se decidió modularizar creando un módulo de dibujo, que obtiene los datos de las variables a graficar y, luego se encarga de crear y mostrar el gráfico utilizando la biblioteca externa «AndroidPlot». Y por otro lado se creó un módulo que se encarga de realizar las peticiones al servidor geográfico para obtener todos los valores de los índices históricos. De esta manera en futuras actualizaciones se pueden agregar otros índices sin tener que modificar gran parte del código.

Teniendo en cuenta la ubicación del usuario, en la presente versión se pudo haber forzado al mismo para que antes de iniciar la aplicación, tenga el GPS activado y de esta manera siempre intentar obtener la ubicación del móvil en la que se esté ejecutando. Pero dado que puede suceder que el usuario abra la aplicación y solamente se mantenga fuera de un incendio, para darle privacidad sobre su ubicación en ese momento, se decidió dejarle la posibilidad al usuario de habilitar el GPS. Además de esta manera, el mismo puede decidir desactivar el GPS para ahorrar el consumo, tanto de la batería como de datos al utilizar conexión 3G, y seguir utilizando la aplicación sin inconvenientes.

3.4.4. Resultados

3.4.4.1. Requisitos de instalación

Para poder utilizar esta versión de la aplicación, se deberán cumplir con los requisitos de versiones anteriores, agregando lo siguiente:

- Posibilidad de obtención de la ubicación del móvil mediante el sistema GPS.
- Al instalar la aplicación, el usuario deberá aceptar los siguientes permisos para el funcionamiento correcto de la misma: Acceso a Internet sin límites, Ubicación aproximada (según la red) y Ubicación precisa (según el GPS y la red).

3.4.4.2. Capturas

En la Figura 3.15 se muestra la interfaz gráfica para la localización del móvil en el mapa.

Al presionar en el botón de localización, si no está habilitado el GPS en el dispositivo, se mostrará un cuadro para que el usuario pueda habilitarlo (Imagen 1). Una vez se tenga el GPS en funcionamiento, se mostrará en el mapa el punto en el que se encuentra el móvil en el momento y, al trasladarse se irá actualizando la posición y marcando la línea de seguimiento (Imagen 2). Si se presiona nuevamente en el botón de localización, se realiza un acercamiento al punto de la ubicación, y si se presiona sobre este último se podrá ver un cuadro

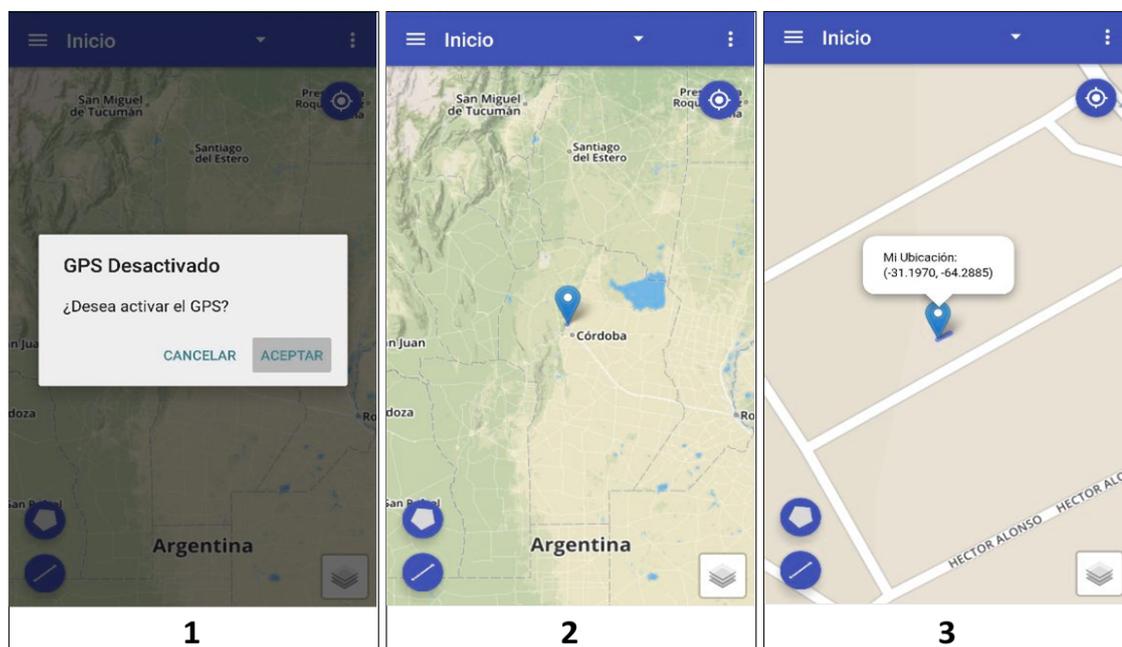


Figura 3.15: Viendo ubicación del móvil.

con las coordenadas del mismo (Imagen 3).

En la Figura 3.16 se muestra la interfaz gráfica para ver los gráficos de índices históricos.

Al igual que en la creación de un punto, se mantiene presionado en el mapa y luego de unos segundos se muestra un ícono rojo en el lugar seleccionado (Imagen 1), al presionarlo aparece un cuadro con la opción de ver la información del punto (Imagen 2). Luego de presionar el botón *Ver Información*, se abre una pantalla con una lista de botones, uno por cada índice histórico y las coordenadas del punto elegido en el mapa (Imagen 3).

Al presionar en uno de los botones, se mostrará una barra de progreso mientras se realizan las consultas al servidor (Imagen 4). Se pueden seleccionar más de uno a la vez y se tomarán los datos de cada índice seleccionado de forma paralela. Una vez que se obtengan los valores para ser graficados, se habilitarán los botones que permiten visualizar los gráficos (Imagen 5). Al presionar uno, se abre una pantalla con el diagrama que marca los valores del índice con respecto a los meses del año (Imagen 6).

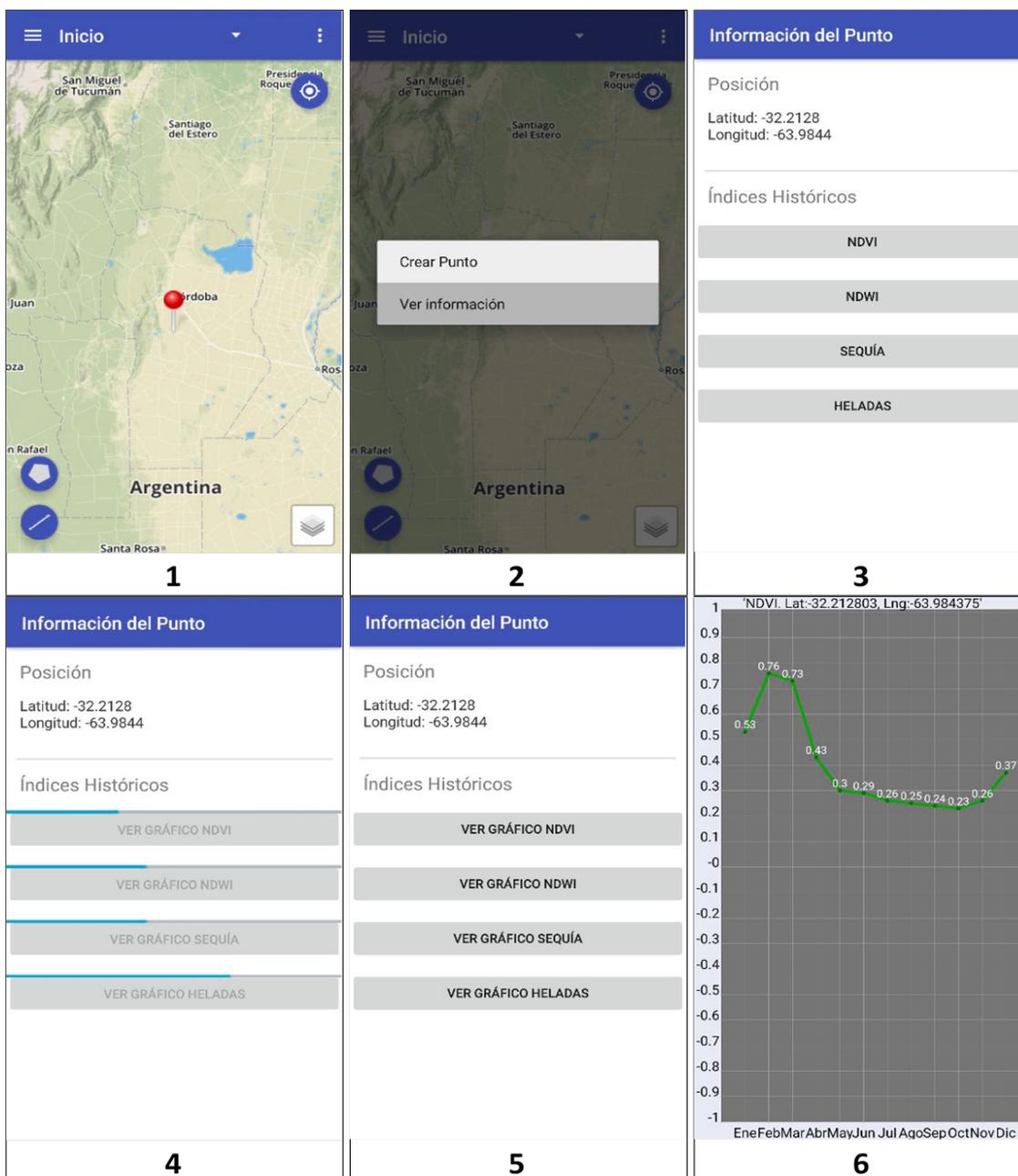


Figura 3.16: Viendo datos históricos.

3.4.4.3. Pruebas

Agregación del punto en el mapa Se creó un código *javascript* en una página HTML independiente, para agregar un punto y actualizar su ubicación manualmente. Se comprobó que la línea de seguimiento se vaya creando correctamente. Luego de incluir el código en la aplicación, se comprobó que sea el mismo que el de la página HTML y se aseguró la obtención del mismo efecto.

Resultado: Actualización correcta del punto y la línea en el mapa, tanto en la página HTML como en la aplicación.

Localización del móvil Se realizó una prueba con un emulador ejecutando la aplicación. Se simuló una ubicación del móvil, se obtuvo la posición mediante el código realizado y al simular cambios de posición se comprobó que los valores resultantes sean los mismos que los ingresados. Luego se agregó la manipulación del punto en el mapa y se comprobó visualmente el correcto funcionamiento.

Resultado: Actualización exitosa de la ubicación.

Obtención de datos históricos del servidor Se realizaron las peticiones al GeoServer con la aplicación, se comprobó que se devuelva una respuesta exitosa y luego se ejecutaron manualmente las mismas consultas en un navegador web para luego comparar los resultados. **Resultado:** Se recibieron los datos correctamente y se logró una comparación exitosa.

Creación de los gráficos Se realizó un proyecto aparte solamente para probar el uso de la biblioteca «AndroidPlot», tomando valores ingresados manualmente. Posteriormente se incluyó la funcionalidad en la aplicación tomando los valores obtenidos del GeoServer para un índice en particular. Se comprobó visualmente que el diagrama sea el correcto. **Resultado:** Se construyeron los gráficos correctamente.

3.4.5. Apreciación final y correcciones

Luego de la reunión final se vieron algunos detalles que causaban lentitud al interactuar con la aplicación y que se debían solucionar. En una primera implementación de las peticiones de valores históricos, cada una de las consultas al servidor demoraban demasiado tiempo en completarse, haciendo muy tediosa la espera para poder ver un gráfico. Por lo que se consultó a personal de CONAE con el objetivo de encontrar una posible optimización. Se llegó a la conclusión de que se estaba utilizando como valor del BBOX⁶ la extensión total del cuadrante, causando una sobrecarga innecesaria.

Finalmente la solución sugerida fue reducir el valor del BBOX en cada consulta, restringiéndolo sólo a las coordenadas del punto a consultar, y de esta manera el tiempo de respuesta por cada petición se redujo considerablemente.

Otro problema encontrado fue la demora excesiva al cargarse la aplicación cuando se ingresaba en ella, causada por el tiempo que conllevan las peticiones de capas temáticas al GeoServer. El menú de capas era creado luego de la obtención de todas las capas, lo que causaba que la interfaz gráfica no estuviese

⁶BBOX (Bounding box): es un arreglo que contiene los límites de la extensión geográfica en la que se encuentra el punto a consultar.

disponible para el usuario hasta que se hayan realizado todas las actualizaciones. La solución fue acotar la búsqueda de capas solamente a la provincia de Córdoba y, realizar la creación del menú de capas al iniciar la aplicación y posteriormente actualizarlo a medida que se vayan obteniendo las capas deseadas.

3.5. Versión 4

La siguiente versión incluye el manejo de usuarios solamente en la aplicación cliente.

3.5.1. Análisis

3.5.1.1. Requerimientos a satisfacer

- Iniciar sesión y ver interfaz gráfica diferenciada según el rol del usuario.
- Crear combatientes a cargo de un supervisor.
- Compartir capas entre usuarios. (Un supervisor compartirá capas solamente a sus combatientes a cargo y un combatiente solo compartirá capas a su supervisor.)

3.5.1.2. Historias de usuario

Como usuario general quiero:

- Poder entrar en la aplicación sin iniciar sesión.

Como usuario combatiente quiero:

- Iniciar sesión con un nombre de usuario y contraseña.
- Poder cerrar sesión cuando desee.
- Al estar iniciada la sesión, si cierro y vuelvo a abrir la aplicación, que siga estando la sesión iniciada.
- Al iniciar sesión quiero ver la interfaz gráfica adaptada a mi ocupación, en mi caso la interfaz de combatientes.
- Al crear capas y figuras que solamente pueda verlas yo, a menos que quiera compartirlas.
- Poder seleccionar para compartir una o más capas a mi supervisor.
- Ver una lista de capas que puedo compartir, teniendo en cuenta si me encuentro fuera o dentro de un incendio y modalidad.
- Ver la lista de las capas que mi supervisor comparte conmigo.
- Si mi supervisor me comparte una capa dentro de una modalidad de un incendio, yo puedo verla solo en la lista de capas de esa modalidad y dentro de ese incendio.

- Poder seleccionar una capa compartida por mi supervisor y, que se active y se vea en el mapa, como cualquiera de mis capas.

Como usuario supervisor quiero:

- Tener las mismas opciones a realizar que los combatientes.
- Tener la opción de crear un combatiente a mi cargo.
- Al crear un combatiente, que se muestre un formulario para que pueda llenar con sus datos.
- Al crear un combatiente, quiero que sea asignado a todos los incendios que creé y los que crearé en el futuro.
- Al crear un incendio, quiero que todos mis combatientes sean asignados a él.
- Ver una lista de combatientes a mi cargo.
- Al seleccionar un combatiente de la lista, quiero ver sus datos que ingresé en el formulario de creación.
- Al compartir una capa, que puedan verla solo mis combatientes.
- Quiero ver en el menú de capas, las capas compartidas por mis combatientes.

3.5.1.3. Casos de uso

Los casos de uso que se desprenden de las historias de usuario anteriores son:

CASO DE USO: Iniciar sesión.	
Actor Primario: Usuario.	
Pre-Condición: No haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario abre la aplicación por primera vez o luego de cerrar una sesión.	
2) El sistema muestra el formulario para llenar con los datos para ingresar.	
3) El usuario completa el formulario y presiona el botón para iniciar sesión.	3.a) El usuario presiona el botón entrar sin iniciar sesión. Se ingresa en la aplicación sin usuario.
4) El sistema verifica los datos ingresados, inicia sesión y muestra la pantalla inicial correspondiente al tipo de usuario que inició la sesión.	4.a) Los datos ingresados no son válidos. Se avisa al usuario para que ingrese nuevamente.
Post-Condición: Se iniciará sesión y se mostrará la interfaz de acuerdo al tipo de usuario.	

CASO DE USO: Cerrar sesión.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenario Alternativo
1) El usuario abre el menú principal.	
2) El sistema muestra el menú con el botón de cerrar sesión.	
3) El usuario presiona el botón de cerrar sesión.	3.a) El usuario presiona el botón regresar atrás. Se cancela el procedimiento.
4) El sistema cierra sesión y muestra el formulario para iniciar sesión.	
Post-Condición: Se cerrará sesión y se mostrará el formulario para iniciar sesión nuevamente.	

CASO DE USO: Compartir capas.	
Actor Primario: Usuario registrado.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón de compartir capa.	
2) El sistema muestra una lista de capas disponibles para compartir.	2.a) No hay capas disponibles para compartir. Se muestra la lista vacía.
3) El usuario selecciona las capas que quiere compartir, y presiona el botón para compartirlas.	3.a) El usuario presiona el botón regresar atrás. Se cancela el procedimiento y no se comparte ninguna capa.
4) El sistema toma las capas seleccionadas y las comparte, cierra la lista y se muestra el menú de capas, con las capas recién compartidas dentro de la lista de capas compartidas.	4.a) No se seleccionó ninguna capa. Se le avisa al usuario para que seleccione alguna. 4.b) Las capas no se pudieron compartir por problemas internos. Se notifica al usuario.
Post-Condición: Se compartirán las capas en el sistema y se las mostrará en la lista de capas compartidas.	

CASO DE USO: Crear combatiente nuevo.	
Actor Primario: Usuario Supervisor.	
Pre-Condición: Haber iniciado sesión y ser supervisor.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón de crear nuevo combatiente.	
2) El sistema muestra el formulario para llenar con los datos del combatiente.	
3) El usuario completa el formulario y presiona el botón para aceptar el formulario.	3.a) El usuario presiona el botón regresar atrás. Se cancela la creación.
4) El sistema verifica los datos ingresados, crea el combatiente y lo muestra en la lista de combatientes.	4.a) Los datos ingresados no son válidos. Se avisa al usuario para que ingrese nuevamente. 4.b) El combatiente no se pudo guardar debido a problemas internos. Se notifica al usuario.
Post-Condición: Se agregará el combatiente nuevo al sistema y se lo mostrará en la lista de combatientes.	

CASO DE USO: Ver información de combatientes.	
Actor Primario: Usuario Supervisor.	
Pre-Condición: Haber iniciado sesión y ser supervisor.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona el botón para ver combatientes.	
2) El sistema muestra una lista de combatientes agregados por el supervisor.	2.a) No hay combatientes agregados. Se muestra la lista vacía.
3) El usuario selecciona el combatiente que quiere ver.	3.a) El usuario presiona el botón regresar atrás. Se cancela el procedimiento.
4) El sistema muestra los datos ingresados al agregar el combatiente seleccionado.	4.a) No se pudieron mostrar los datos, debido a problemas internos. Se notifica al usuario.
Post-Condición: Se mostrarán los datos del combatiente elegido.	

3.5.2. Diseño

3.5.2.1. Base de datos

Para la presente versión se preservó la estructura de la base de datos de versiones anteriores y, se agregaron las tablas «User» (que guarda los datos de un usuario) y «SupFiref» (que guarda los identificadores de supervisores con combatientes para relacionar los combatientes a cargo de un supervisor determinado), para llevar a cabo las nuevas funcionalidades. Además se modificaron las tablas «Fire» y «UserLayer», agregando una nueva columna en cada caso para relacionarlas con la nueva tabla de usuarios, guardando el identificador del usuario creador del objeto, como se muestra en la Figura 3.17.

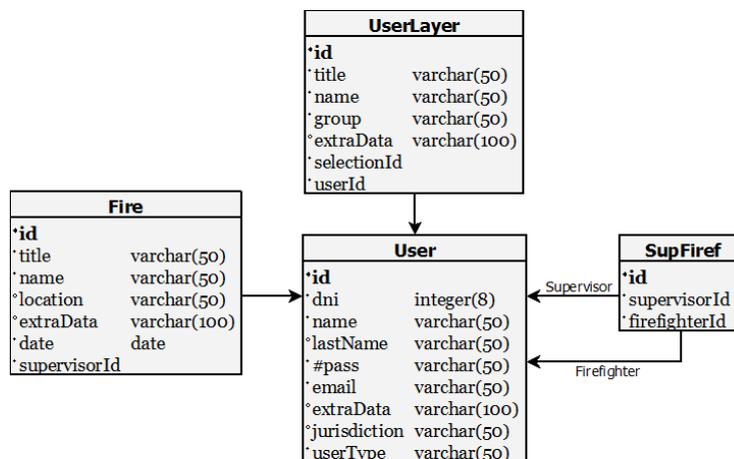


Figura 3.17: Base de datos Versión 4.

3.5.3. Implementación

Por simplicidad, para la presente versión se optó por realizar el manejo de usuarios sólo del lado del cliente y dejar para la siguientes iteraciones el desarrollo del servidor web.

Se simulan las creaciones de datos como si se tuviese un servidor. Por lo tanto información que debería encontrarse en el mismo, son guardados en la aplicación móvil, como lo es la contraseña del usuario. Esto causa que los datos creados en la aplicación, solamente sean válidos para el dispositivo en el que se efectuó la creación. Por ejemplo, cuando se agrega un combatiente por parte de un supervisor, la acción se realiza satisfactoriamente pues el combatiente puede iniciar sesión, pero solamente en el móvil en el que fue creado, dado que los datos no son guardados en un contenedor en común y por ende no pueden ser accedidos desde otros teléfonos inteligentes. Esta posibilidad fue agregada en versiones siguientes luego de implementar el servidor REST junto con la comunicación con la aplicación móvil.

Para tener una buena organización de la información, se decidió que solamente se puedan compartir capas y no figuras por separado. Además se optó que una vez compartida una capa, la misma no se pueda modificar, es decir que se impidió la creación de nuevas figuras en las capas compartidas. También se

prefirió que solamente se le de al usuario la posibilidad de compartir las capas que no sean por defecto, como las de “Figuras sin Clasificar”, debido a que el usuario se quedaría sin estas capas para guardar información no clasificada. Si esto se quiere implementar, una posibilidad podría ser copiar la información a una nueva capa y así poder compartirla. Todas estas decisiones mencionadas se realizaron meramente por simplicidad en el desarrollo de la presente versión, dejando para actualizaciones futuras la incorporación de las diferentes posibilidades planteadas.

Uno de los retos que surgieron, fue cómo presentar al usuario las capas compartidas y no compartidas. Se optó por separar en subgrupos cada uno de los grupos de capas creadas por el usuario del menú lateral. De esta forma se dividen en “Capas Compartidas” y “Capas no Compartidas”. Permitiendo al usuario diferenciarlas fácilmente.

Por otro lado, se tuvo que realizar una modificación en la obtención de capas temáticas, debido a que algunas capas cambian de nombre según la fecha en la que se crean en el GeoServer y a veces no se actualizaban todos los días, y por lo tanto no se veían en la aplicación. Para solucionar esto, se decidió realizar una consulta del protocolo WMS, llamada «GetCapabilities», al GeoServer y leer el XML de respuesta para así obtener todas las capas que se encuentran disponibles en el momento de la petición. De esta forma no se necesita realizar del lado del cliente el mismo algoritmo que se utiliza en el GeoServer para nombrar las capas y correr el riesgo de que no se hayan actualizado. Para implementar esta actualización, se creó un módulo encargado de formar las direcciones web para construir las peticiones al GeoServer, otro módulo para efectuar las consultas y finalmente otro para leer la respuesta en formato XML. De esta forma se simplifica el proceso de actualización del código en caso de necesitar agregar nuevas capas de este estilo.

3.5.4. Resultados

3.5.4.1. Requisitos de instalación

Para poder utilizar esta versión de la aplicación, se deberán cumplir con los requisitos de versiones anteriores agregando lo siguiente:

- Al instalar la aplicación, el usuario deberá aceptar los siguientes permisos para el funcionamiento correcto de la misma: Acceso a Internet sin límites, Ubicación aproximada (según la red), Ubicación precisa (según el GPS y la red), Leer datos de contacto, Leer tarjeta contacto propia y buscar cuentas en el dispositivo.

3.5.4.2. Capturas

Al abrir por primera vez la aplicación o en caso de cerrar una sesión anterior, se mostrará un formulario que el usuario que desee iniciar sesión, deberá completar con sus datos y elegir su puesto entre *Supervisor* o *Combatiente*. La interfaz gráfica se puede ver en el Figura 3.18. En la Imagen 1 se puede ver que por defecto se encuentra seleccionado el tipo de usuario supervisor, y se muestra un botón para ingresar sin inicio de sesión (para usuario público). En la Imagen 2 y 3 se muestran ejemplos de formularios rellenos por un supervisor y un

combatiente respectivamente.

The figure shows three sequential screenshots of a login form, labeled 1, 2, and 3. Each screenshot has a blue header with the word 'Aplicación'.
 Screenshot 1: Shows the form with empty input fields for 'Email' and 'Contraseña'. The role dropdown is set to 'supervisor'. Below the fields are two buttons: 'INICIAR SESIÓN' and 'ENTRAR SIN USUARIO'.
 Screenshot 2: Shows the form with 'sup1@mail.com' in the email field and a masked password '.....' in the password field. The role dropdown is still 'supervisor'.
 Screenshot 3: Shows the form with 'comb1@mail.com' in the email field and a masked password '.....' in the password field. The role dropdown is now set to 'combatiente'.

Figura 3.18: Formulario de inicio de sesión.

Luego de presionar el botón *Iniciar Sesión* y los datos del formulario sean válidos, se entrará en la aplicación y se mostrará la interfaz gráfica con el mapa, como en versiones anteriores.

En la Figura 3.19 se muestran las capturas de la interfaz gráfica vista por un supervisor para compartir capas a sus combatientes. Cabe aclarar que la interfaz gráfica para un combatiente es la misma, diferenciándose en la presentación de las capas en el menú luego de ser compartidas, como se verá más adelante.

En la presente versión las capas que pueden ser compartidas son únicamente las creadas por el usuario en un incendio (en cualquiera de sus modalidades). Por lo tanto el botón para compartir capas se encuentra en el menú del incendio en el grupo de capas creadas en el mismo, como se puede ver en la Imagen 1. Al presionar este botón, se abre una nueva ventana con la lista de capas no compartidas que se encuentran en el grupo de capas del incendio en la modalidad seleccionada (Imagen 2).

El usuario debe seleccionar de la lista, todas las capas que desea compartir y presionar el botón *Compartir Capas*. Si se compartieron satisfactoriamente, como resultado se mostrará al usuario un mensaje de éxito y se cerrará la lista de capas, mostrando nuevamente el menú con las capas compartidas en una nueva sección denominada *Mis Capas Compartidas* dentro del grupo de capas del incendio (Imagen 3).

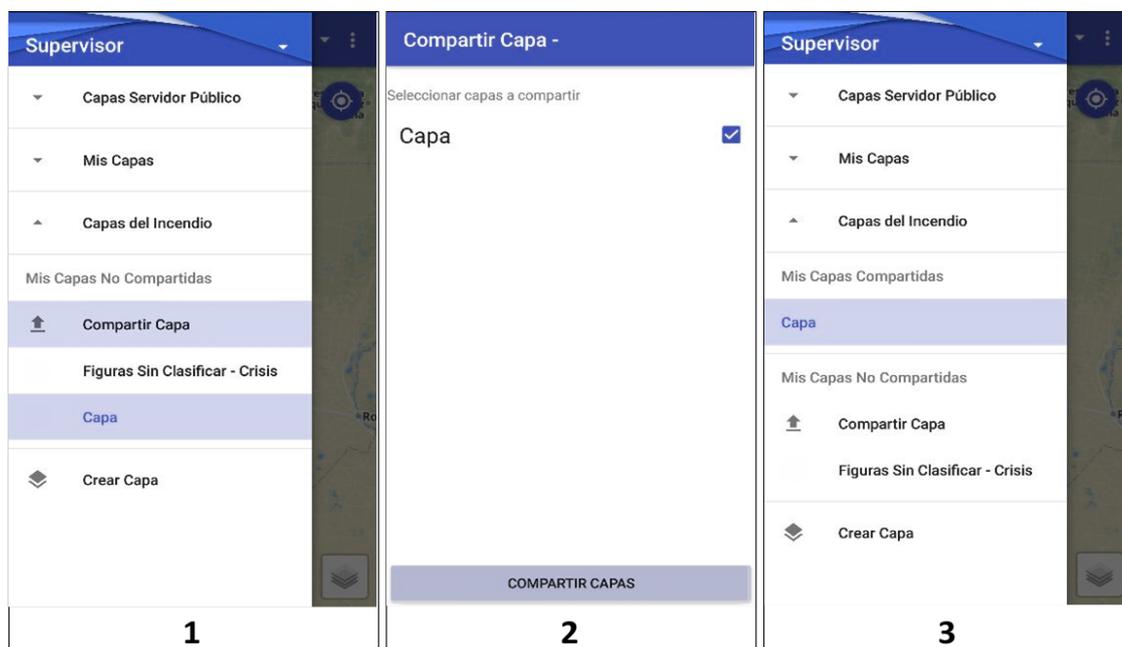


Figura 3.19: Compartiendo Capas.

En la Figura 3.20 se muestran capturas de la interfaz gráfica de un supervisor para la creación de un combatiente. Al abrir el menú principal, se muestra un botón para cerrar la sesión y un botón con nombre *Combatientes* (Imagen 1) que al ser presionado, abre una nueva ventana con la lista de combatientes creados anteriormente y un botón para crear uno nuevo (Imagen 2).

Una vez se presiona en el botón *Crear Combatiente*, se abre el formulario para completar con los datos del mismo (Imagen 3). En la Imagen 4 se muestra el formulario completado. Luego, al presionar el botón para aceptar el formulario, este se cierra y se vuelve a mostrar la lista, actualizada con el nuevo combatiente (Imagen 5). Como se puede ver en la Imagen 6, si se presiona en un combatiente de la lista, se muestran sus datos introducidos al ser creado.

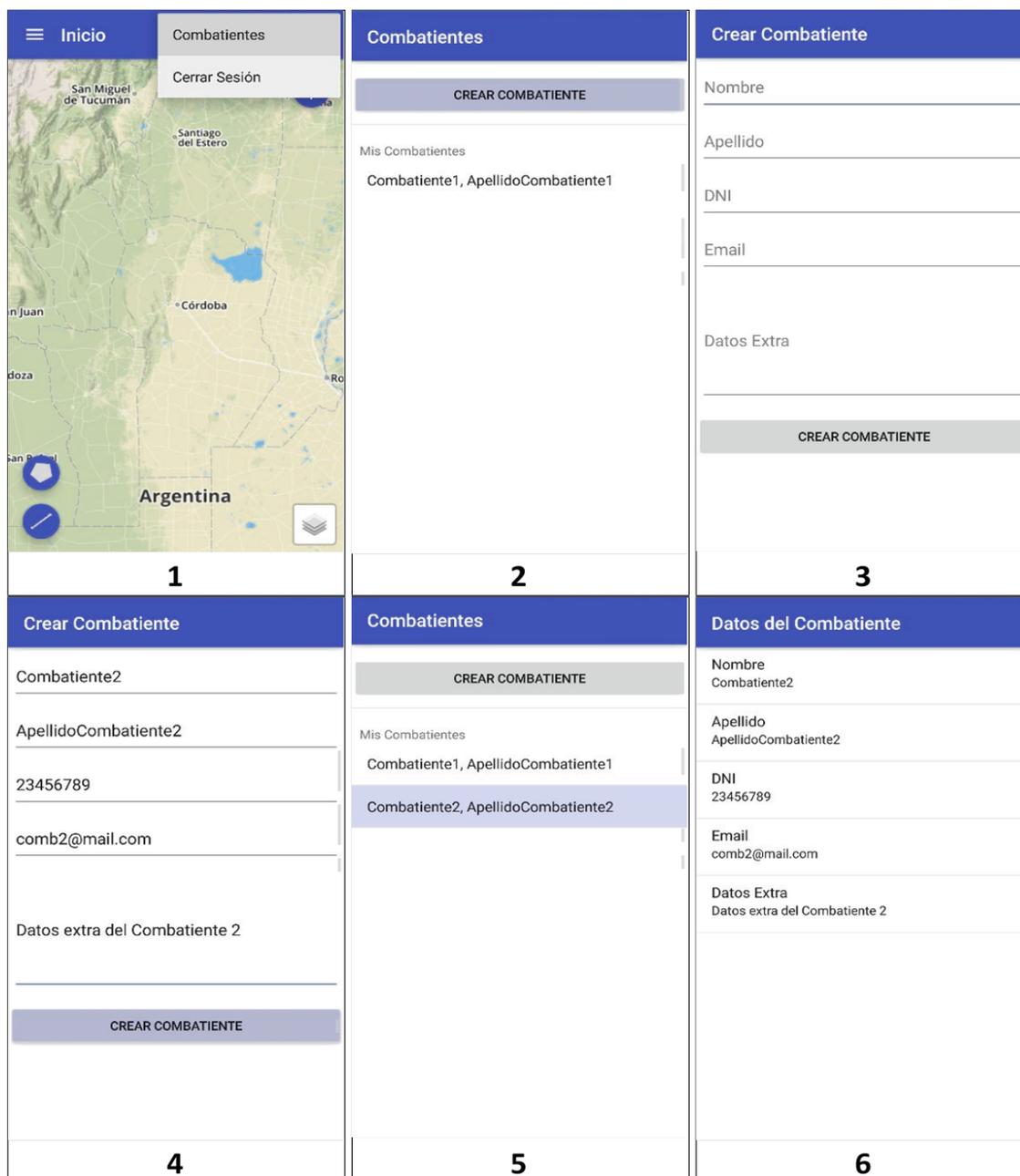


Figura 3.20: Creando combatiente y viendo datos de combatientes.

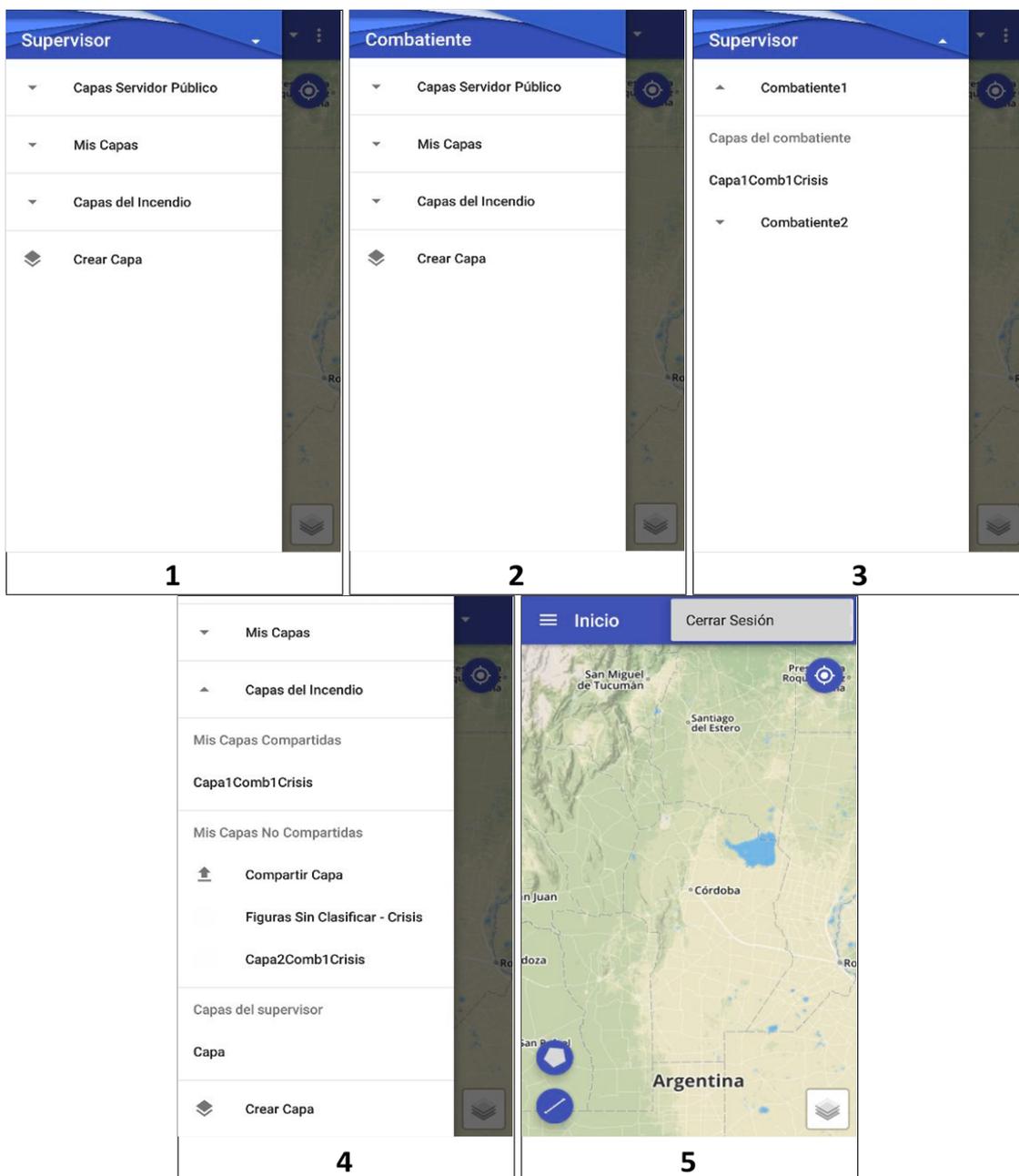


Figura 3.21: Comparando interfaces entre usuarios registrados.

Las interfaces gráficas de los usuarios registrados, tanto supervisores como combatientes, se diferencian principalmente en el menú de capas. En la Figura 3.21 se muestran las interfaces gráficas del tipo de usuario *Supervisor* y *Combatiente*, comparándolas luego de que ambos hayan compartido capas.

En las Imágenes 1 y 2 se puede ver el menú de un incendio para un supervisor y un combatiente respectivamente, y se diferencian en que el supervisor contiene un botón en forma de “flecha” al lado del título del menú. Al presionarlo se ocultan los grupos de capas y se muestra una lista de capas compartidas por todos los combatientes a su cargo (Imagen 3). Si se presiona nuevamente la flecha, entonces se vuelven a mostrar los grupos como en la Imagen 1.

En cambio, como el combatiente únicamente comparte sus capas con su supervisor, es decir, que solamente podrá ver las capas de este y de ningún otro usuario, las capas compartidas por el supervisor se podrán visualizar en el grupo de capas del incendio en una nueva sección denominada *Capas del Supervisor*. (Imagen 4). Como se puede observar la capa que compartió el supervisor se ve en el menú del combatiente y viceversa. Además en la Imagen 5, se puede ver que en el menú principal del combatiente solamente se encuentra el botón para cerrar sesión, a diferencia del supervisor que también posee el botón para listar combatientes como se vio anteriormente.

3.5.4.3. Pruebas

Compartir una capa Luego de simular compartir una capa, se verificó que se haya actualizado correctamente la base de datos, corroborando que la capa sea la elegida y que haya cambiado su estado a compartida. Además se verificó que en las otras capas no se hayan producido cambios. **Resultado:** Se realizaron correctamente los cambios de capas no compartidas a compartidas y se conservó la integridad de las demás capas.

Inicio y cierre de sesión Se realizó un proyecto por separado, con el objetivo de únicamente iniciar y cerrar sesión. Se corroboró que al ingresar un nombre de usuario y contraseña, si son válidos ambos entonces se inicie la sesión, si alguno es incorrecto que muestre el respectivo error. Y al cerrar sesión, se comprobó que las variables del sistema se restablezcan correctamente. Luego de agregar la funcionalidad a la aplicación, se realizaron nuevamente las pruebas. **Resultado:** Verificación exitosa en ambos casos tanto para el inicio como para el cierre de sesión.

Creación de combatiente nuevo Luego de la creación de un combatiente por parte de un supervisor, se comprobó realizando las consultas a la base de datos, que los datos del usuario y la relación con el supervisor junto con las capas por defecto del nuevo combatiente hayan sido creadas exitosamente. Luego se comprobó que al seleccionar en un combatiente, se obtengan los datos correspondientes comparándolos con los de la tabla. **Resultado:** Actualización correcta de la base de datos y obtención exitosa de datos.

Lectura y parseo de un archivo XML. Se realizó manualmente la consulta *GetCapabilities* para una capa, se obtuvo el archivo y se utilizó el módulo de lectura implementado para leerlo. Luego se corroboró que el resultado de

la lectura sea el correcto. **Resultado:** Comparación exitosa con los datos del archivo.

3.5.5. Apreciación final y correcciones

Al finalizar la versión, se discutieron a cerca de algunas mejoras y correcciones, que fueron implementadas luego en la versión siguiente.

Uno de los temas que se discutieron, fue la necesidad que podría tener el usuario de compartir las capas creadas por él fuera de un incendio, de la misma forma que ya lo hace dentro de un incendio. Se planteó la idea de darle la posibilidad al usuario de compartir sus capas desde cualquier lugar en la aplicación.

Además se planteó la necesidad de que al cerrar y volver a abrir la aplicación, las capas que hayan sido activadas antes de cerrar, vuelvan a aparecer como seleccionadas. Así el usuario no tendría que perder su tiempo en seleccionar nuevamente las capas que necesita.

También se discutieron mejoras en la interfaz gráfica, con el objetivo de hacerla más amigable e intuitiva para el usuario. Una de las modificaciones propuestas para realizar en la próxima versión, fue borrar de la lista desplegable de incendios la opción “Inicio” y sólo listar los incendios disponibles. Debido a que puede confundir al usuario, pues “Inicio” no sería un incendio.

Por último se hablaron sobre nuevas posibles funcionalidades como por ejemplo, ver las coordenadas de un punto correspondiente al presionar en el mapa, agregar fotos georeferenciadas por parte de los usuarios, y la opción de que el supervisor pueda ver la ubicación de sus combatientes. La primera fue implementada en la siguiente versión. La dos últimas fueron dejadas para actualizaciones futuras. Una opción que se analizó para agregar las fotos, fue por ejemplo crear puntos como los que ya se pueden agregar a las capas, y asignarle una opción en el formulario de creación del mismo para incluir una fotografía. La opción de ver la ubicación del combatiente, también podría llevarse a cabo pues ya se tiene implementada la localización del lado del usuario, sólo se debería crear una capa que guarde esta información y sea compartida al supervisor.

Se habló sobre la idea original de poder utilizar la aplicación en varios dispositivos. Por lo tanto el manejo de usuarios debe hacerse a través de un servidor en común, del cual se puedan obtener los datos y así poder iniciar una sesión en cualquier móvil, conservando siempre la información creada. Por lo tanto se analizaron los requerimientos tanto de la siguiente versión como los del servidor a realizar, expuestos a continuación.

3.6. Versión 5

En la siguiente sección se explica el procedimiento realizado para agregar a la aplicación móvil, la posibilidad de conectarse a un servidor web para la obtención de datos desde cualquier dispositivo Android.

3.6.1. Análisis

3.6.1.1. Requerimientos a satisfacer

- Poder utilizar la aplicación en varios dispositivos, para esto guardar los datos de la aplicación en un servidor web y comunicarse con el mismo.

- Ver capas activadas al abrir la aplicación, si fueron seleccionadas en aperturas anteriores.
- Ver latitud y longitud al presionar un punto en el mapa.
- Al ir creando áreas y líneas, ver el valor temporal marcado de la superficie y longitud respectivamente.
- Desplegar las leyendas de capas temáticas, con la simbología asociada a las mismas para una mejor interpretación por parte del usuario.

3.6.1.2. Historias de usuario

Como usuario general quiero:

- Poder iniciar sesión en la aplicación en cualquier dispositivo móvil.
- Al activar una capa y cerrar la aplicación, si vuelvo a abrirla quiero ver esa capa activa nuevamente.
- Ver en el mapa las coordenadas cuando presiono sobre él.
- Si presiono en un punto del mapa, que se actualice el texto con las del puntero seleccionado.

Como usuario registrado quiero:

- Tener las mismas funcionalidades que un usuario general.
- Al cerrar sesión en un dispositivo e iniciar sesión en otro, que toda mi información sea recuperada.
- Al crear un área, que se muestre en el mapa el valor de la superficie marcada temporalmente. Es decir, al agregar un punto en el mapa que se vuelva a calcular la superficie incluyendo este punto, y se muestre en el mapa.
- Al crear una línea, que se muestre en el mapa el valor de la distancia marcada temporalmente. Es decir, al agregar un punto en el mapa que se vuelva a calcular la distancia incluyendo este punto, y se muestre en el mapa.
- Al terminar de crear una área o cancelar su creación, que se oculte del mapa el valor de la superficie temporal.
- Al terminar de crear una línea o cancelar su creación, que se oculte del mapa el valor de la distancia temporal.
- Ver una lista de capas temáticas que poseen leyendas con simbología asociada.
- Seleccionar una capa temática con leyenda y que se despliegue sobre el mapa una imagen con la misma, y al presionar sobre ella que se oculte.
- Si una imagen con una leyenda aparece en el mapa y comienzo a crear una línea o un área, que la imagen se oculte.

3.6.1.3. Casos de uso

Los casos de uso que se desprenden de las historias de usuario anteriores son:

CASO DE USO: Ver capas previamente seleccionadas.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o haber entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario activa una capa en el menú de capas.	
2) El sistema guarda la capa como seleccionada.	
3) El usuario cierra y vuelve a abrir la aplicación	3.a) El usuario cierra sesión. Se desactivan las capas activadas en la sesión anterior.
4) El sistema muestra como seleccionadas las capas activadas anteriormente.	
Post-Condición: Se mostrarán como seleccionadas las capas activadas en un ingreso anterior a la aplicación.	

CASO DE USO: Ver latitud y longitud de un punto.	
Actor Primario: Usuario.	
Pre-Condición: Haber iniciado sesión o haber entrado sin usuario.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona en el mapa.	1.a) El usuario presiona fuera del mapa. No se modifican las coordenadas.
2) El sistema muestra la latitud y longitud en una esquina del mapa.	
Post-Condición: Se mostrará la latitud y longitud en el mapa.	

CASO DE USO: Ver medición temporal al crear figura.	
Actor Primario: Usuario supervisor o combatiente.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona botón para crear la figura.	
2) El sistema muestra en el mapa un mensaje con el valor de la medición en cero.	
3) El usuario agrega un punto perteneciente a la figura.	3.a) El usuario presiona en el botón de cancelar. Se oculta el mensaje con la medición.
4) El sistema actualiza en el mapa la medición generada hasta el momento.	
5) Se repite desde el paso 3) hasta que el usuario termine la creación de la figura.	
Post-Condición: Se mostrará temporalmente, en el mapa, la medición al crear las figuras.	

CASO DE USO: Ver imagen con leyenda de una capa temática.	
Actor Primario: Usuario supervisor o combatiente.	
Pre-Condición: Haber iniciado sesión.	
Escenario Normal	Escenarios Alternativos
1) El usuario presiona botón para mostrar leyendas.	
2) El sistema muestra un cuadro con una lista de capas temáticas que contienen leyendas.	2.a) No hay capas con leyendas. No se muestra el cuadro y se le avisa al usuario.
3) El usuario presiona sobre una capa temática del cuadro.	3.a) El usuario presiona en el botón de regresar atrás. Se oculta el cuadro y se vuelve a mostrar el botón para ver leyendas.
4) El sistema oculta el cuadro y el botón de ver leyendas, y muestra sobre el mapa, una imagen con la leyenda de la capa seleccionada.	4.a) No se encuentra la imagen. Se notifica al usuario y se cancela la operación.
Post-Condición: Se mostrará una imagen en el mapa con la leyenda de una capa temática seleccionada.	

CASO DE USO: Ocultar imagen con leyenda de una capa temática.	
Actor Primario: Usuario supervisor o combatiente.	
Pre-Condición: Haber iniciado sesión y haber seleccionado una leyenda para mostrar.	
Escenario Normal	Escenario Alternativo
1) El usuario presiona sobre la imagen con la leyenda.	1.a) El usuario presiona botón para crear línea o área. Se oculta la imagen y al finalizar se muestra el botón para ver leyendas.
2) El sistema oculta la imagen y vuelve a mostrar el botón para ver leyendas.	
Post-Condición: Se ocultará la imagen en el mapa con la leyenda de una capa temática seleccionada.	

3.6.2. Diseño

Para completar el diseño de la aplicación, se tuvo que realizar primero un diseño del lado del servidor web. Se plantearon el uso de diferentes tipos de servidores, tanto el software a utilizar como el tipo de comunicación con el dispositivo.

Se decidió utilizar un servidor LAMPy, con el framework Django [10] junto con Django REST [11] para realizar una comunicación de tipo REST con la aplicación móvil.

3.6.2.1. Arquitectura

En la Figura 3.22 se puede observar la nueva arquitectura general del sistema. Se modificó la de versiones anteriores, agregando la comunicación con el nuevo servidor. Este último se relaciona con su base de datos «MySQL» para guardar la información que se necesita conservar. Para la inserción y obtención de datos del servidor, se deben utilizar peticiones HTTP siguiendo el protocolo REST.

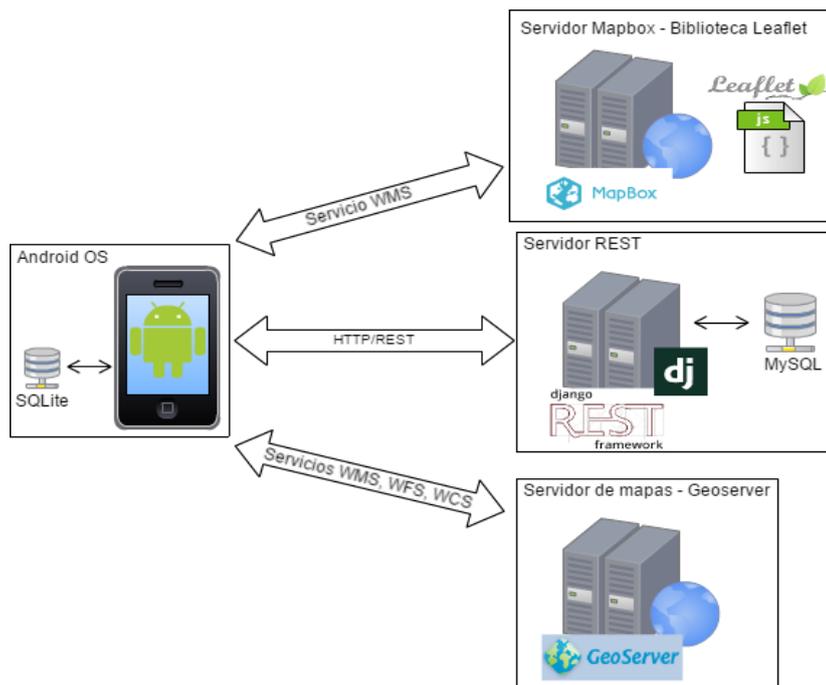


Figura 3.22: Arquitectura versión 5.

3.6.2.2. Base de datos

En la Figura 3.23 se muestra la estructura de la base de datos creada para guardar los datos en el servidor. Las tablas con el título «(Default)» agregado, son tablas por defecto que ofrece Django para guardar los datos del usuario, los grupos (tipo de usuario) y sus permisos. Como se puede observar se creó una

tabla extra llamada «AppUser», agregando los campos para la información que la aplicación móvil necesita guardar además de los que ofrece la tabla «User» de Django.

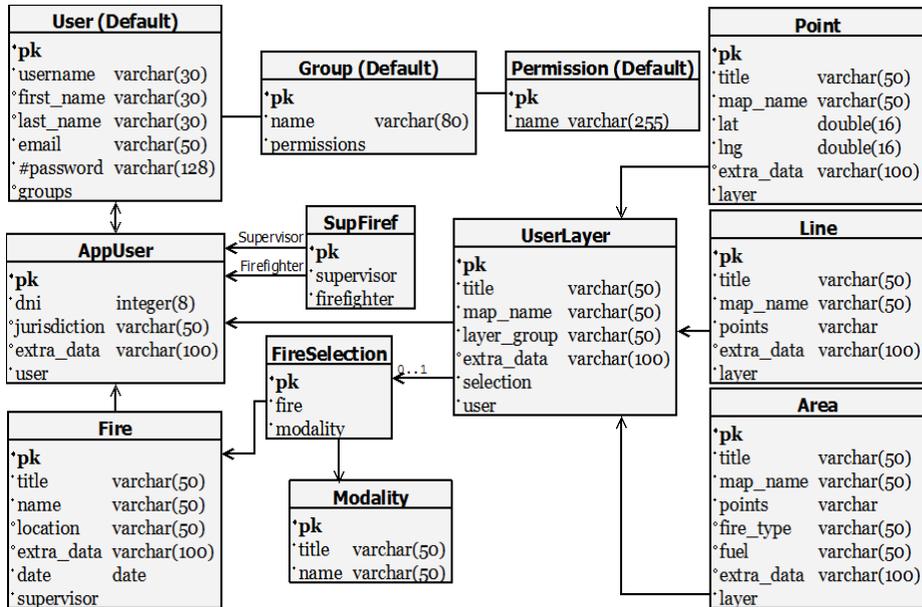


Figura 3.23: Base de datos del servidor.

En la Figura 3.24 se puede observar la estructura completa de la base de datos de la aplicación Android. En donde se modificaron todas las tablas, para guardar los identificadores con los cuales son guardados los mismos datos en el servidor. La tabla «User» guardará los datos de las tablas «AppUser», «User» y «Group» del servidor, tomando sólo el identificador de «AppUser» y exceptuando la contraseña, que ahora se encuentra solamente en el servidor y se usará para validar el usuario en el mismo.

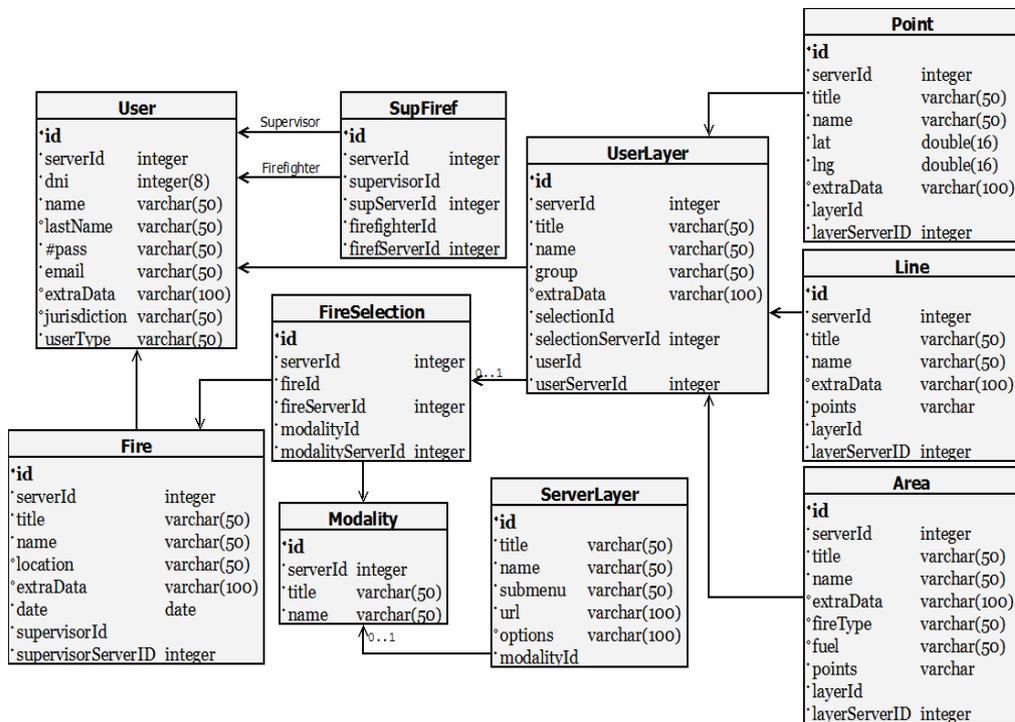


Figura 3.24: Base de datos del móvil.

3.6.3. Implementación

Para cumplir con los requerimientos de esta versión, se implementaron módulos separados que se encargan de construir y efectuar las peticiones HTTP al servidor REST. De esta manera si en un futuro se desea realizar otro tipo de comunicación con un servidor web, los cambios en el código se reducen básicamente a modificar estos módulos.

Se optó por utilizar el formato JSON para la transmisión de información entre el servidor web y la aplicación móvil, debido a que se manejan tipos de datos simples y como consecuencia ofrece mayor seguridad que XML, dado que no se pueden compartir ejecutables que podrían comprometer la seguridad del servidor. Sin embargo, si en actualizaciones futuras se desean guardar este tipo de datos, se tendrían que realizar las modificaciones correspondientes tanto en el servidor como en la aplicación móvil, para poder utilizar el formato XML.

Se decidió conservar una copia de los objetos del servidor en el dispositivo, para reducir la cantidad de consultas por cada uno y poder utilizarlos libremente en la aplicación móvil. Esto tiene la desventaja de que se ocupa memoria en el dispositivo, pero por otro lado si se corta la conexión a Internet luego de obtener los datos, se puede seguir accediendo a ellos sin necesidad de volver a pedirlos al servidor. Esta es la razón por la cual se tuvieron que modificar las tablas de la base de datos del móvil como se enseñó en la sección de diseño.

Además se tuvo que decidir cómo tratar con las capas y figuras creadas por el usuario, si guardarlas primero en el dispositivo y luego cuando se posea señal

enviarlas al servidor, o primero enviarlas al servidor y luego de una respuesta exitosa guardarlas en el dispositivo. La primera opción tiene la ventaja que la aplicación se puede utilizar sin conexión a Internet, pero impide que se puedan compartir los datos en tiempo real. En esta versión se optó por la segunda opción. Una desventaja de esta decisión es que se depende de la comunicación con el servidor y de la conexión a Internet al momento de la creación de objetos en el mapa. Pero por otro lado, de esta forma siempre se tiene la misma versión del recurso en el servidor y si se desean compartir con otros usuarios, estos siempre tendrán la última actualización.

En la presente versión las peticiones de datos al servidor se realizan solamente al abrir la aplicación. Y no se incluye una actualización automática al ser compartidas capas y figuras de un usuario a otro. Es decir, si por ejemplo un supervisor comparte una capa a sus combatientes, si ellos poseen la aplicación abierta en ese momento, no se actualizará. Pero si se cierra y vuelve a abrir la aplicación se verá la capa compartida.

Para efectuar tal actualización, se analizó la posibilidad de utilizar notificaciones push⁷. Se realizaron pruebas con el paquete *django-push-notifications*, configurando Django para utilizar el servicio *Google Cloud Messaging* (GCM) [18] para Android, consiguiendo resultados exitosos al compartir capas pero con algunas fallas al actualizar incendios. Debido a que en este punto se llegó al límite de tiempo planeado para finalizar el trabajo, se optó por dejar para versiones futuras la implementación completa de esta característica.

3.6.4. Resultados

Los requisitos de instalación continúan siendo los mismos que en la versión anterior.

3.6.4.1. Capturas

En la presente versión se modificó la interfaz gráfica con el objetivo de hacerla más amigable e intuitiva para el usuario.

Si se observa en las Figuras 3.25 y 3.26, en la Imagen 1 se muestra el formulario de inicio de sesión modificado para ingresar solamente el nombre de usuario y la contraseña. Se eliminó la lista desplegable para elegir tipo de usuario pues cuando el servidor realiza la autenticación, comprueba el tipo de usuario dando el permiso correspondiente.

Al iniciar sesión, se puede ver el mapa como en versiones anteriores junto con las coordenadas que marcan el centro del mapa, en una esquina superior (Imagen 2). Cuando el usuario presiona en algún punto del mapa, el texto anterior se reemplaza con las coordenadas del punto presionado. Además se puede observar que en el título de la aplicación, ya no se muestra la sección *Inicio* sino que se muestra la opción *Seleccionar Incendio*. De esta forma el usuario sabe que debe presionar en la lista para seleccionar el incendio que desee. Cuando el usuario presiona en la lista de incendios solamente se mostrarán aquellos a los cuales pertenece, y si fuera supervisor también se mostrará el botón para la creación de un incendio (Imagen 8, para supervisor. Imagen 9, para combatientes).

⁷Mensajes que un servidor envía a una aplicación cliente alertándola sobre información nueva.

Volviendo a la Imagen 3, se puede ver el menú de capas de un supervisor fuera de un incendio. Se puede notar que para la presente versión se implementó la capacidad de compartir las capas que no pertenecen a un incendio, a las que ahora llamamos *Capas generales*. Se muestra desplegado el grupo de las capas mencionadas donde se ven separadas las compartidas y las no compartidas por el usuario. En la Imagen 4 se puede ver el mismo menú pero visto por un combatiente. En este caso se muestran también las capas compartidas por su supervisor fuera de un incendio. El supervisor puede ver las capas compartidas por sus combatientes seleccionando en el botón del título del menú, denominado *Capas de combatientes*. Este abre un nuevo menú con la lista de sus combatientes y las capas que le compartieron (Imagen 6).

En la Imagen 5 se puede ver la lista de capas públicas que se pueden obtener fuera de un incendio para la presente versión.

Como se puede observar en la Figura 3.26, se modificó el formulario para creación de incendios (Imagen 7). Se eliminó el nombre de un incendio y se lo reemplazó por la fecha de creación. De esta manera los incendios se distinguen ahora por fecha y la estructura del título de cada uno es *Inc:FECHA* como se muestran en las Imágenes 8 y 9.

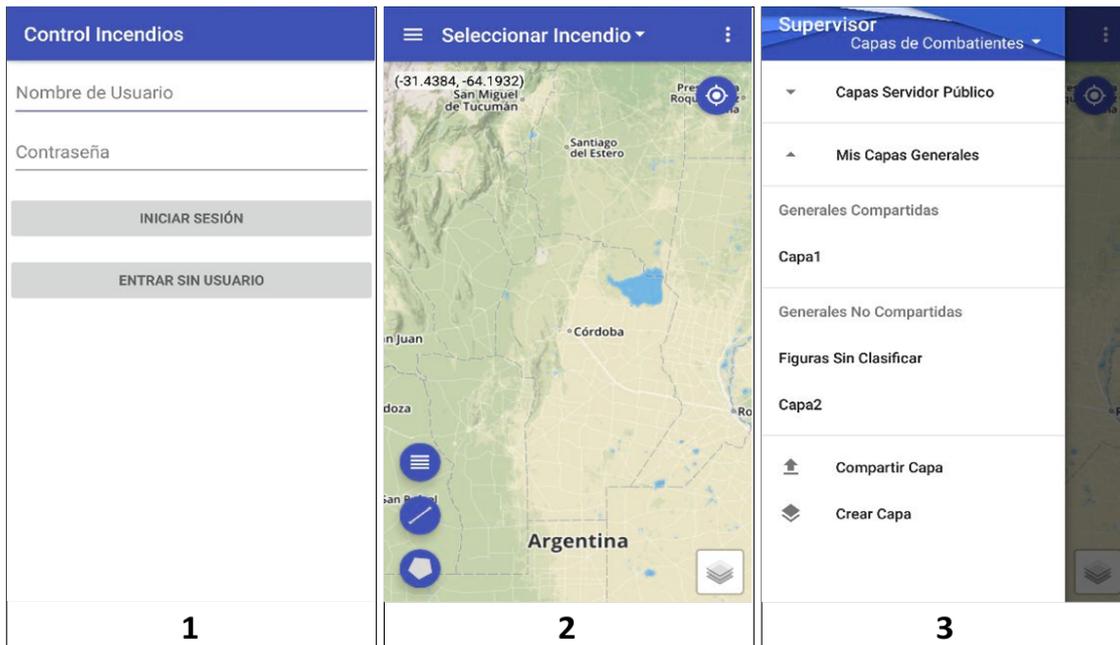


Figura 3.25: Inicio de sesión y menús de versión 5. (Continuación en Figura 3.26)

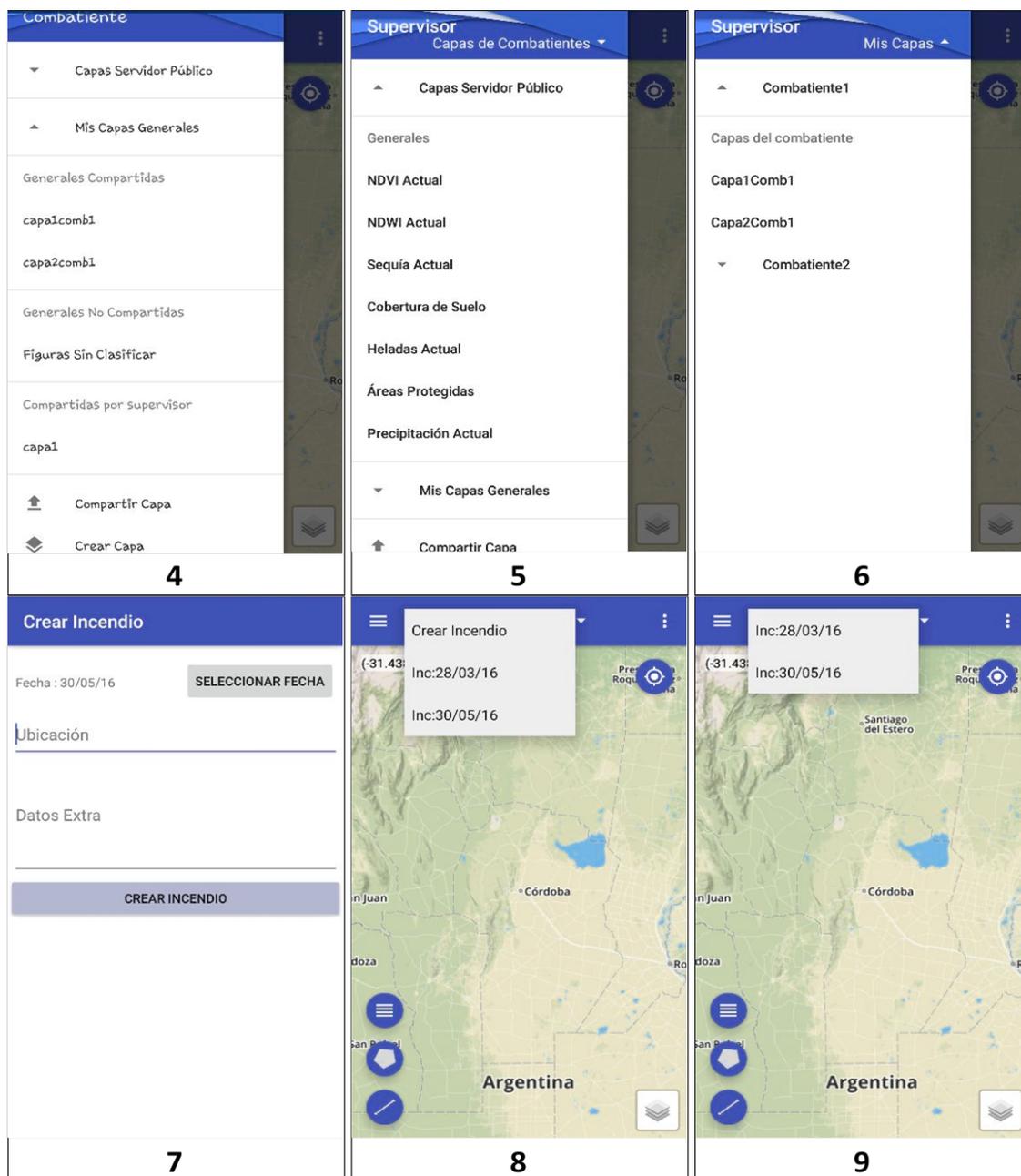


Figura 3.26: Continuación de Figura 3.25. Inicio de sesión y menús de versión 5.

Si se observa las Figuras 3.27 y 3.28 se puede observar que al entrar a un incendio, se muestra el título con la fecha y además se muestran las mismas coordenadas que se veían en la esquina superior antes de entrar en el incendio, y se cumple con la misma funcionalidad (Imagen 1).

En las Imágenes 2 y 3 puede verse el menú de capas del incendio para

un supervisor y un combatiente respectivamente. Como se puede observar, la diferencia es la misma que en el menú de capas afuera de un incendio, en el combatiente se muestran las capas compartidas por el supervisor en el mismo grupo desplegable de capas del incendio y en el supervisor se debe presionar en el botón del título del menú denominado *Capas de Combatientes*. Si este último es presionado entonces se abre un nuevo menú con la lista de combatientes del supervisor y, las capas que le comparten en el incendio y en la modalidad seleccionada. Para volver al menú anterior se debe presionar en el botón del título del menú llamado *Mis Capas* (Imagen 4).

Para cumplir con las funcionalidades de ver los valores temporales al crear un área o una línea, se agregó en una esquina superior un texto que se muestra en modo edición del mapa y se va actualizando a medida que se van insertando los puntos de cada una de las figuras, como se muestra en las Imágenes 5 y 6. Al salir del modo de edición del mapa, es decir, cuando se terminan de crear las figuras, el texto se oculta.

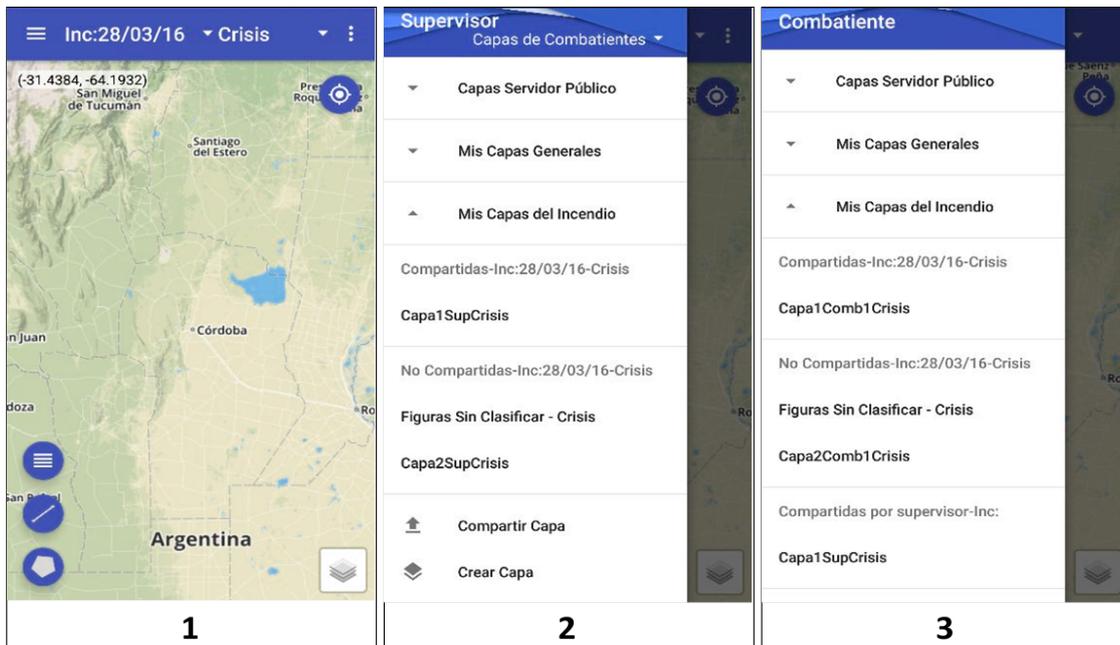


Figura 3.27: Menú incendios y creación de figuras.

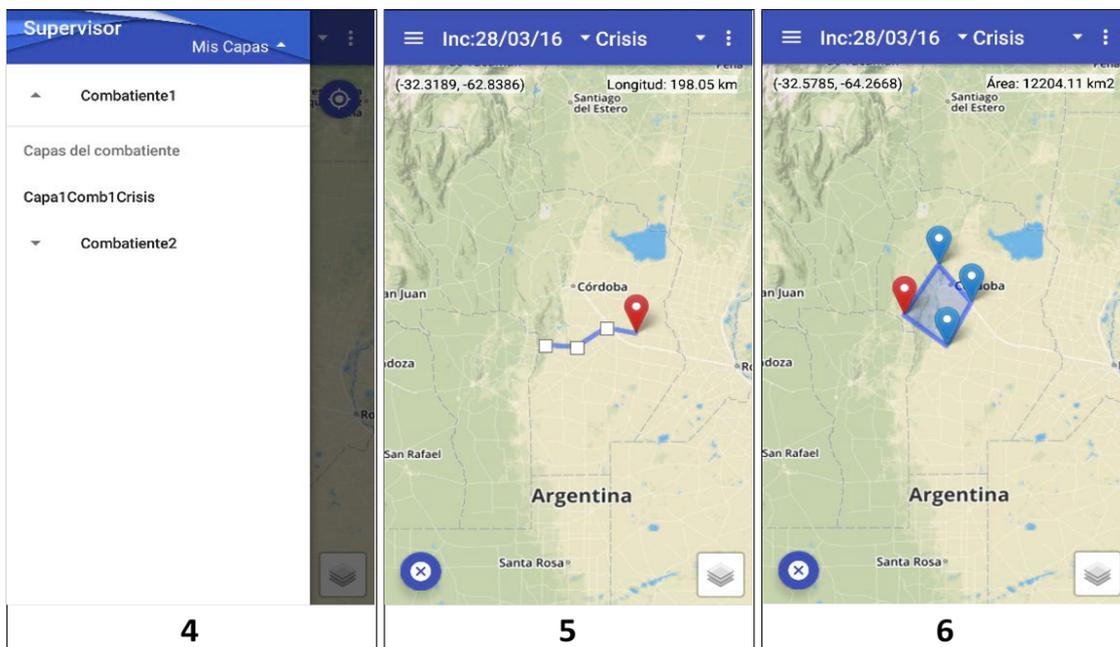


Figura 3.28: Continuación de Figura 3.27. Creación de figuras.

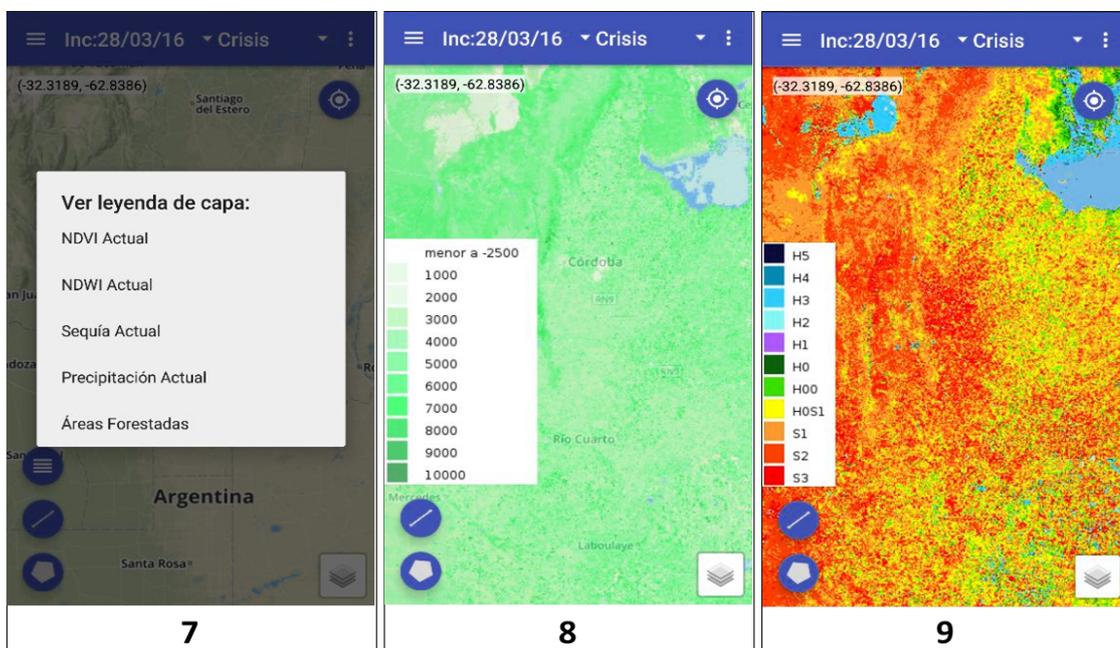


Figura 3.29: Viendo leyendas de capas temáticas.

Por último para mostrar las leyendas que posean las capas temáticas, se agregó un botón sobre los botones para la creación de áreas y líneas. Al presionar sobre él, se abre un cuadro con la lista de capas temáticas que poseen leyendas, como se muestra en la Figura 3.29 en la Imagen 1. Luego de elegir una capa, se cierra el cuadro y se muestra la imagen con la leyenda. En la Imagen 2 se puede ver el resultado de presionar la capa temática NDVI y en la Imagen 3 se puede ver la leyenda de la capa de Sequía.

3.6.4.2. Pruebas

Vista de Coordenadas. Se creó una página HTML aparte con el código *javascript* para mostrar las coordenadas presionadas y se la ejecutó en un navegador web con éxito. Luego de agregar la funcionalidad en la aplicación, se comprobó la obtención del mismo código. Finalmente se probó la actualización correcta en el cuadro de texto del mapa. **Resultado:** Comparación exitosa entre los códigos y correcta actualización del texto de las coordenadas.

Vista de distancia y superficie. Se creó una página HTML aparte con el código *javascript* para mostrar los valores calculados de líneas y áreas ingresadas manualmente, y luego al ir creándolas en el mapa. Se la ejecutó en un navegador web exitosamente. Luego de agregar la funcionalidad en la aplicación, se comprobó la obtención del mismo código. Posteriormente se probó la actualización correcta en el cuadro de texto del mapa para mostrar los valores calculados. **Resultado:** Comparación exitosa entre los códigos y correcta actualización del texto en el mapa.

Recepción de datos y respuesta del servidor. Se realizaron peticiones manualmente, utilizando la herramienta incorporada en PyCharm llamada *Test RESTful webservice*. Se construyeron manualmente los encabezados y cuerpos de cada tipo de consulta, se ejecutaron y posteriormente se comprobó que el servidor las reciba de forma correcta, y que efectúe una respuesta según corresponda. **Resultado:** Actualización de la base de datos del servidor correcta para cada tipo de petición que lo requiera, y respuesta adecuada según la consulta realizada.

Envío de datos al servidor. Se creó una aplicación aparte en la que se enviaban un nombre de usuario y contraseña al servidor. Se revisó que el servidor realice correctamente la consulta a su base de datos y responda con éxito si era correcta la combinación de los datos. Luego se integró esta funcionalidad en la aplicación y se realizaron nuevamente las pruebas. **Resultado:** Envío y recepción de información en la aplicación, y en el servidor de manera correcta.

3.6.4.3. Apreciación final y correcciones

Esta versión es el resultado de cuatro iteraciones, las dos primeras con implementación del servidor y correcciones de errores, y las otras realizadas en ciclos reducidos en tiempo con el objetivo de sumar las funcionalidades para ver los valores de las distancias y superficies marcadas, y para ver las leyendas de capas temáticas.

El cambio llevado a cabo en la forma de nombrar a los incendios, utilizando la fecha de inicio, se dio gracias a una consulta realizada a un bombero voluntario, sobre el modo en el que distingue los incendios en la documentación que realiza. De esta forma el usuario no tiene que inventar un nombre arbitrario, solamente se limita a colocar la fecha de inicio del incendio. Se debe tener en cuenta que pueden ocurrir varios incendios en un mismo día, por lo tanto también se deberían agregar para versiones futuras, la hora y minutos de comienzo de cada uno.

En esta versión se realizan demasiadas peticiones al servidor REST, debido a que ya se había implementado la lógica de gestión de los datos y de las funcionalidades del lado de la aplicación cliente, y no del lado del servidor. Por ejemplo cuando se piden los incendios, se realiza una petición de las capas por cada una de las modalidades mientras que con una misma consulta se podrían obtener todas las capas juntas. Por lo tanto para versiones futuras se podría implementar la lógica del lado del servidor reduciendo la cantidad de peticiones realizadas.

Finalmente se realizó una reunión con un bombero voluntario nombrado anteriormente, para presentarle la aplicación terminada en la versión 5. Se le mostró cada funcionalidad y se planteó un escenario de uso para cada una, dando como resultado una aprobación de su parte y garantizando la utilidad del trabajo realizado.

Como, gracias al proceso iterativo, se podría seguir avanzando y creando varias versiones con nuevos requerimientos quizás no considerados, se decidió concluir el trabajo con la versión 5. Ya que contempla la mayor cantidad de funcionalidades planteadas originalmente y además luego de ser presentada a un posible usuario, se la considera funcional y apta para un posible caso de prueba.

Capítulo 4

Conclusión

En este trabajo se desarrolló una herramienta útil para el personal encargado del manejo de incendios. Se tuvo como objetivo principal que les permitiera un acceso rápido a información geoespacial y a datos generales de la emergencia. De esta forma se les brinda una visión global de la situación en la que se encuentran, ayudándoles a evaluar las acciones a tomar durante las diferentes etapas de un siniestro.

Al finalizar el proyecto se realizó una reunión con un bombero voluntario de la localidad de Río Ceballos, quien avaló la utilidad de la aplicación señalando lo beneficioso de cada funcionalidad de acuerdo a diferentes escenarios en los cuales podría ser utilizada.

Para llevar a cabo el proyecto, se ha diseñado un sistema cliente/servidor en donde la aplicación Android implementada actúa como cliente, que interactúa con servidores geográficos para la obtención de mapas y con un servidor REST para guardar información compartida por la aplicación móvil, también desarrollado en conjunto con la misma.

Se debieron estudiar los diferentes protocolos de comunicación con cada tipo de servidor. Se utilizaron los servicios WMS, WFS, WCS y WPS para peticiones al servidor de mapas, y se utilizaron consultas HTTP estándar siguiendo el estilo de la arquitectura REST para peticiones al servidor web implementado.

Durante el desarrollo del proyecto se han tenido que tomar diferentes decisiones y superar una serie de retos técnicos en cada fase de desarrollo. En un principio se debieron elegir las diferentes herramientas a utilizar y cómo se llevaría a cabo el proceso de trabajo. Esto fue un gran reto pues cada herramienta utilizada era totalmente desconocida para la alumna, por lo que se tuvo que dedicar un tiempo considerable al estudio del funcionamiento de las mismas.

Ayudó mucho, el hecho de que antes de aceptar llevar a cabo el proyecto en cuestión, se tuviera una idea inicial del tipo de aplicación que se debía realizar, junto con requisitos específicos con los cuales empezar a trabajar. Con esta idea se comenzó a planear y a diseñar la aplicación siguiendo un desarrollo iterativo tomando algunos puntos de la ideología de la metodología ágil para llevar a cabo el proceso de trabajo.

Se implementaron varias prácticas recomendadas por el proceso iterativo, se realizaron ciclos de trabajos cortos permitiendo la evolución del proyecto con nuevos requerimientos al comienzo de cada uno y, con reuniones al finalizar cada iteración, para analizar las funcionalidades implementadas. Se realizaron los casos de uso recomendados para el análisis de los requisitos y se llevaron a cabo las pruebas pertinentes para comprobar el correcto funcionamiento de cada versión de la aplicación desarrollada.

A modo de resumen, a continuación se listan las diferentes versiones de la aplicación, realizadas siguiendo objetivos específicos:

- **Version 1:** Ver un mapa de Argentina y agregar capas temáticas.
- **Version 2:** Agregar varios incendios distinguiendo sus distintas modalidades y crear capas con diferentes tipos de figuras marcadas en el mapa.
- **Version 3:** Ver gráficos con datos de índices históricos tomando un par de coordenadas seleccionadas en el mapa.
- **Version 4:** Manejo de usuarios en la aplicación cliente.
- **Version 5:** Posibilidad de compartir información entre usuarios desde diferentes dispositivos, mediante un servidor web de intermediario.

La experiencia de realizar un trabajo desde cero y ver que los conocimientos adquiridos en la carrera han permitido llevar adelante un proyecto de desarrollo de software con resultados verdaderamente útiles, que podrían ayudar a personas en la vida cotidiana, ha sido realmente grata y satisfactoria. Además sumando los nuevos conocimientos adquiridos durante el autoaprendizaje sobre las distintas herramientas, lenguajes de programación y, desarrollo y gestión de proyectos, finalmente se consideran satisfechos los objetivos propuestos.

4.1. Trabajos futuros

Como se ha comentado en secciones anteriores, durante el desarrollo del proyecto han ido surgiendo ideas que han sido integradas en el mismo pero otras han sido descartadas, pues no imponían un reto diferente al de las funcionalidades ya planteadas y solamente aumentaban la complejidad del proyecto, o por no disponer del tiempo necesario para implementarlas. A continuación se destacan algunas líneas de trabajo futuras, sobre diferentes mejoras y nuevos desarrollos que podrían ser interesantes de estudiar y realizar:

- Notificación instantánea a los usuarios, sobre actualizaciones realizadas por otros usuarios. Por ejemplo cuando un usuario comparte sus capas o se crean nuevos incendios.
- Vista de la ubicación y seguimiento de combatientes por parte del supervisor.

- Posibilidad de modificar los datos compartidos. Poder seguir agregando figuras a las capas ya compartidas.
- Agregar áreas o líneas utilizando el GPS. Ir marcando el recorrido con el GPS y al terminar guardar el área o línea marcadas.
- Lanzar notificaciones al usuario sobre posibles focos en un área marcada por él o cuando se acerque a una cierta distancia de un posible foco de incendio. Además permitir que cualquier usuario pueda subir información sobre un incendio desatado en sus cercanías.
- Permitir la descarga de mapas al dispositivo móvil, para disminuir la transferencia de datos del usuario. Y poder utilizar la aplicación sin conexión a Internet.
- Esta aplicación sólo está abocada a incendios, pero podría ser llevada a otro nivel, considerando otros tipos de catástrofes y emergencias como las inundaciones.
- Extender la obtención de datos y mapas a todo el país o a hasta a nivel mundial. E incluir las referencias que explican el contenido de las capas temáticas.
- Posibilidad de unir capas creadas por el usuario para crear una sola con todas las figuras. Y también poder eliminar datos como incendios, capas o figuras.
- Distinguir los puntos creados por el usuario en distintos tipos de puntos cada uno con su propio tipo de información. Además distinguirlos por medio de un color o ícono diferente. De esta forma se pueden crear puntos de ubicación de personal para compartir si el supervisor quiere que se ordenen de una forma en particular, lugar donde se encuentran los camiones, tranqueras, puntos de referencia, focos de incendio, etc.
- Posibilidad de filtrar información y realizar estadísticas, como por ejemplo cantidad de incendios en un mes y una zona determinada, cantidad de áreas y tipo de vegetación quemadas, entre otras.
- Mejorar el diseño de la interfaz gráfica del menú de capas. Eliminar los subgrupos que separan las capas compartidas de no compartidas y en su lugar distinguirlas de otra forma como por ejemplo con íconos al lado del título de cada capa.
- Desarrollo de la aplicación cliente para diferentes plataformas nativas o en forma de aplicación web multiplataforma.
- Medidas de seguridad más rigurosas tanto en la aplicación cliente como del lado del servidor. Además aumentar la descripción para el usuario sobre el uso de la aplicación, con mayor cantidad de mensajes sobre el estado de la misma e incluir un tutorial para su uso.

Bibliografía

- [1] Drongelen, M. *Android Studio Cookbook*. Packt Publishing. ISBN: 978-1-78528-068-9. Octubre 2015.
- [2] Nader, J. *Modelización con Diagramas de Casos de Uso*.
- [3] Peralta, G. *Trabajo Especial Licenciatura en Ciencias de la Computación: “Desarrollo de tecnología móvil para la prevención y lucha contra los incendios en la provincia de Córdoba”*. Diciembre 2005.
- [4] Wong, C. *HTTP: The Definitive Guide*. O’Reilly, 2002
- [5] AndroidPlot. Dirección web: <http://androidplot.com/>. Mayo 2016.
- [6] AJAX. Dirección web: <http://www.w3schools.com/ajax/>. Agosto 2016.
- [7] Apache. Dirección web: <http://www.apache.org/>. Mayo 2016.
- [8] ArcGIS. Dirección web: <https://www.arcgis.com>. Mayo 2016.
- [9] Código JSON. Dirección web: <http://www.json.org/>. Mayo 2016.
- [10] Django. Dirección web: <https://www.djangoproject.com/>. Mayo 2016.
- [11] Django REST framework. Dirección web: <http://www.django-rest-framework.org/>. Mayo 2016.
- [12] Curso de programación en Android. Dirección web: <http://www.sgoliver.net/>. Mayo 2016.
- [13] Desarrollo en Android. Dirección web: <http://developer.android.com/>. Mayo 2016.
- [14] Descarga de Android SDK y Android Studio IDE. Dirección web: <http://developer.android.com/intl/es/sdk/index.html>. Mayo 2016.
- [15] Explorador de Coordenadas. Dirección web: <http://isithackday.com/geoplanet-explorer/>. Mayo 2016.
- [16] Ionic Framework. Dirección web: <http://ionicframework.com/>. Mayo 2016.
- [17] Google API v2. Dirección web: <https://developers.google.com>. Mayo 2016.
- [18] Google Cloud Messaging (GCM). Dirección web: <https://developers.google.com/cloud-messaging/>. Mayo 2016.

- [19] Google Earth. Dirección web: <https://www.google.com/earth/>. Agosto 2016.
- [20] Instrucciones para montar un servidor web. Dirección web: <http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=662>. Mayo 2016.
- [21] Intel XDK. Dirección web: <https://software.intel.com/es-es/html5/tools>. Mayo 2016.
- [22] jQuery. Dirección web: <https://jquery.com/>. Mayo 2016.
- [23] Justinmind. Dirección web: <http://www.justinmind.com/>. Mayo 2016.
- [24] Leaflet. Dirección web: <https://leafletjs.com>. Mayo 2016.
- [25] Biblioteca Gson. Dirección web: <https://github.com/google/gson> . Mayo 2016.
- [26] Manifiesto Ágil. Dirección web: <http://agilemanifesto.org/>. Mayo 2016.
- [27] Ministerio de Ambiente y Desarrollo Sustentable. Dirección web: <http://www.ambiente.gov.ar/>. Mayo 2016.
- [28] No-IP, DNS Dinámico. Dirección web: <http://www.noip.com/>. Mayo 2016.
- [29] MySQL. Dirección web: <https://dev.mysql.com/doc/>. Mayo 2016.
- [30] Openlayers. Dirección web: <https://openlayers.org/>. Mayo 2016.
- [31] Plataforma de mapas Mapbox. Dirección web: <https://www.mapbox.com/>. Agosto 2016.
- [32] Protocolo HTTP. Dirección web: <https://www.w3.org/Protocols/>. Mayo 2016.
- [33] PyCharm. Dirección web: <https://www.jetbrains.com/pycharm/>. Mayo 2016.
- [34] Python. Dirección web: <https://docs.python.org>. Mayo 2016.
- [35] Servicios WMS, WFS, WCS y WPS. Dirección web: <http://docs.geoserver.org/>. Mayo 2016.
- [36] Stack Overflow, Comunidad de programadores. Dirección web: <http://stackoverflow.com/>. Mayo 2016.
- [37] SQLite. Dirección web: <https://www.sqlite.org/>. Mayo 2016.
- [38] UML. Dirección web: <http://www.uml.org/>. Mayo 2016.
- [39] VMware. Dirección web: <http://www.vmware.com/>. Mayo 2016.
- [40] Web-GIS GeoServer de CONAE. Dirección web: <http://ambiente.caearte.conae.gov.ar/geoexplorer/composer/>. Mayo 2016.