

INSTITUTO SUPERIOR MINERO METALÚRGICO DE MOA

DR. ANTONIO NUÑEZ JIMENEZ

Facultad de Geología, Minas e Informática

Trabajo de Diploma para optar por el Título de Ingeniero Informático

**Aplicación Android de recolección de
datos para el Herbario Digital
Representativo del Distrito Florístico
Moense**

Autor: Abel Nicó Quintero

Tutores: M.Sc. Roiky Rodríguez Noa

Ing. Yisel Clavel Quintero

Consultante: Ing. Adrian Pierra Fuentes

Moa, Julio del 2015



Dedicatoria

A mis padres

Agradecimientos

A mi familia en general (mis padres, abuelos, tíos, mi Hermano, primos y algunos que no pueden estar con nosotros), por ayudarme y estar ahí cada vez que los necesité...

A mis amigos y amigos compañeros de cuarto, por escucharme y aceptarme como soy y por los necesarios y frecuentes momentos de terapia y descanso...

A mis compañeros de aula ya que estuvimos cinco años trabajando duro y ayudándonos...

A mis tutores, Roiky Rodríguez Noa y Yisel Clavel Quintero, por guiarme, preocuparse y ocuparse de que la tesis se desarrollara satisfactoriamente y con la calidad requerida...

A mi novia por escucharme, apoyarme, y en su momento aguantar mis descargas...

A mis profesores, por enseñarme casi todo lo que sé y prepararme para ser un profesional de éxito...

A la Revolución, por permitirme estar aquí y facilitarme los recursos para realizar esta tesis...

Y sobre todo, por todo el esfuerzo y dedicación, a mí.

*“Aquel que no es suficientemente fuerte como para
correr riesgos no podrá lograr nada en la vida”.*

Muhammad Ali

Resumen

El Centro de Estudios de Medio Ambiente del Instituto Superior Minero Metalúrgico de Moa, realiza investigaciones vinculadas fundamentalmente al desarrollo sostenible de los territorios minero-metalúrgicos y a la conservación de los recursos florísticos en estos. Para ello lleva a cabo un inventario de las especies florísticas del territorio; para lo cual utiliza un Herbario Digital con ambiente Web, que favorece en gran parte la gestión de información sobre la flora del Distrito Moense. Para recolectar los datos de las especies florísticas, los especialistas del centro deben trasladarse a localidades intrincadas cargando varios equipos. Por la lejanía y escasez de recursos económicos, se les dificulta la categorización de las especies y el desarrollo de un trabajo sostenible para conservar los recursos florísticos de esta región. La presente investigación propone el desarrollo de una Aplicación *Android* para favorecer la recolección de datos y la gestión de la información de especies florísticas en el Distrito Moense. Para esto se emplearon tecnologías y herramientas de *software* libre. Para guiar el proceso de desarrollo se utilizó la Metodología para el Desarrollo de Aplicaciones Móviles. Se realizó además, un estudio de la factibilidad de la aplicación propuesta.

Abstract

The Center for Environmental Studies of the Metallurgical Mining Institute of Moa, primarily performs research related to sustainable development of the mining-metallurgical territories and the conservation of plant resources in these. For it carried out an inventory of floral species of the territory; for which uses a Digital Herbarium Web environment, which largely favors the management of information on the flora of Moa District. To collect the data of the floristic species, the specialists of the center should be moved to locations intricate loading multiple computers. By remoteness and lack of economic resources, they are difficult to categorize the species and the development of a sustainable work to conserve plant resources of this region. This research proposes the development of an Android Application to facilitate data collection and information management of floristic species in the Moa District. For this technology and free software tools were used. To guide the process of developing the Methodology for the Development of Mobile Applications was used. A study of the feasibility of the proposed application is additionally performed.

Índice de contenido

Introducción.....	1
Capítulo 1: Fundamentos teóricos de la Aplicación <i>Android</i> para la recolección de datos de especies florísticas	6
Capítulo 2: Desarrollo de la Aplicación <i>Android</i> para la recolección de datos de especies florísticas.....	30
Capítulo 3: Estudio de factibilidad de la Aplicación <i>Android</i> para la recolección de datos de especies florísticas	45
Conclusiones	51
Recomendaciones	52
Referencias Bibliográficas	53

Índice de Tablas

Tabla 3.1: Casos de Uso sin ajustar y Actores.	47
Tabla 3.2: Factores de Complejidad Técnica.	47
Tabla 3.3: Factor de Complejidad Técnica.	48
Tabla 3.4: Factores de Complejidad Ambiental.	48
Tabla 3.5: Factor de Complejidad Ambiental.	48
Tabla 3.6: Estimación del esfuerzo y costo.	49

Índice de Figuras

Figura 1.1: Arquitectura de <i>Android</i>	11
Figura 1.2: Etapas de la metodología MDAM.	29
Figura 2.1: Posibles diagramas para el desarrollo de aplicaciones móviles.	34

Introducción

En el mundo actual es fundamental el cuidado y protección de nuestra flora y fauna, pues es un eslabón esencial para la supervivencia de las especies. La ciudad de Moa, ubicada en la provincia de Holguín, por su desarrollo en el campo de la minería es uno de los municipios que está expuesto al deterioro del Medio Ambiente (MA), debido a que el territorio cuenta con grandes reservas de minerales que se explotan desde mediados del siglo XX.

La explotación minera durante un tiempo prolongado ha producido una pérdida notable de la diversidad florística. Debido a lo anterior, es un riesgo evidente que en estos ecosistemas a lo largo del proceso evolutivo, las especies más sensibles en sus requerimientos biológicos y ecológicos van reduciendo sus poblaciones hasta desaparecer en la medida que el impacto en el medio y la fragmentación del hábitat se van acentuando.

El distrito florístico moense es bien conocido por la riqueza que contienen las formaciones vegetales, por la gran cantidad de especies endémicas que se han reportado para el mismo, con un total de 976, de ellas 373 que sólo crecen en el distrito. Dentro del mismo se encuentran áreas con reconocida importancia florística como El Toldo - Monte La Breña, La Melba, Cerro Miraflores, Yamanigüey, Playa La Vaca, entre otras; que han sido visitadas por numerosos naturalistas desde principios del siglo XIX [1].

En este tipo de estudio de la flora moense se ha destacado el Centro de Estudios de Medio Ambiente (CEMA), ubicado en el Instituto Superior Minero Metalúrgico de Moa (ISMMM), que está concebido para la implementación del programa de MA del Ministerio de Educación Superior. Este centro constituye un espacio de concertación interdisciplinaria común a todas las áreas del ISMM; enfoque que le permite la coincidencia de varias ramas de las ciencias como: geología, minería, metalurgia, botánica, veterinaria, forestal, entre otras. Las líneas de investigación están vinculadas fundamentalmente al desarrollo sostenible de los territorios minero-metalúrgicos, por lo que los temas de investigación se enmarcan en las prioridades nacionales y territoriales de la ciencia y la técnica. Es por esto la

necesidad de potenciar las investigaciones referidas a la conservación de los recursos florísticos dentro de un territorio que lleva a cabo una actividad tan destructiva como la explotación minera [2].

Es de vital importancia contar con un inventario de los recursos pues es imprescindible para la detección de las especies vulnerables y en peligro de extinción, para un posterior establecimiento de programas de conservación de la biodiversidad vegetal. Por tal razón, se desarrolló en el CEMA un Herbario Digital Representativo del Distrito Florístico Moense [1], que permitió favorecer en gran parte la gestión de información sobre la flora del distrito moense. Este *software* cuenta con una interfaz Web que permite el acceso desde cualquier máquina computadora conectada a la red.

El CEMA cuenta con un grupo de especialistas que dirigen proyectos de investigación que tributan a la conservación de la biodiversidad vegetal y animal. Sin embargo, es difícil para los especialistas trasladarse hasta localidades intrincadas para la categorización de las especies cargando un mapa, brújula, equipo de geolocalización (GPS), cámara de fotos digital, papel y lápiz. Teniendo en cuenta la lejanía y escasez de recursos económicos, se les dificulta el desarrollo de un trabajo sostenible para conservar los recursos florísticos de esta región. También los centros de referencia a nivel nacional se encuentran lejos de la localidad y de la provincia; limitándose así el trabajo profesional de los especialistas.

Por otra parte, en la actual era de la información, donde la sociedad está interconectada, se ha extendido el uso de dispositivos móviles para gestionar la información de forma más fácil y accesible. Cada vez son más las aplicaciones existentes para estos dispositivos para realizar diversas tareas como conectarse a internet, revisar correo electrónico, gestionar y compartir imágenes, videos, audios y documentos; siendo *Android* el sistema operativo más popular para estos dispositivos.

Actualmente los especialistas recogen periódicamente la información referente a las especies en el campo. Deben cargar hasta lo más intrincado de las localidades

un mapa, brújula, equipo de geolocalización (GPS), cámara de fotos digital, papel y lápiz; por tanto, este trabajo se hace muy engorroso a la hora de recoger dicha información y guardarla en la base de datos del Herbario Digital.

A partir de lo anteriormente planteado se define como **problema científico** de la investigación: ¿Cómo favorecer el proceso de gestión de información de especies florísticas en el Distrito Moense?

Se plantea como **objeto de estudio**: Aplicaciones para dispositivos móviles. Este objeto de estudio delimita como **campo de acción**: Aplicaciones *Android* para gestionar la información de especies florísticas en el Distrito Moense.

Se plantea como **Objetivo General**: Desarrollar una aplicación *Android* que favorezca el proceso de gestión de información de especies florísticas en el Distrito Moense.

Para guiar la investigación, se elaboraron las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos en cuanto al desarrollo de aplicaciones *Android* para gestionar la información de especies florísticas en el Distrito Moense?
- ¿Cómo desarrollar una aplicación *Android* que favorezca del proceso de gestión de información de especies florísticas en el Distrito Moense?
- ¿Será factible la solución que se propone?

Para dar respuesta a las preguntas científicas y cumplir el objetivo trazado, se realizaron las siguientes **tareas**:

1. Elaborar los fundamentos teóricos en cuanto al desarrollo de aplicaciones *Android* para gestionar la información de especies florísticas en el Distrito Moense
2. Determinar las herramientas, tecnologías y metodología a utilizar para el desarrollo de la aplicación *Android* propuesta
3. Realizar el estudio de factibilidad de la aplicación *Android* propuesta
4. Desarrollar la aplicación *Android* propuesta

Para dar cumplimiento a las tareas planteadas se emplearon **métodos teóricos y empíricos** de investigación científica. Los **métodos teóricos** utilizados son:

- **Análisis y síntesis:** Se utilizó con el fin de analizar la información sobre el desarrollo de aplicaciones *Android* y la relacionada con la gestión de la información de especies florísticas en el Distrito Moense, así como para elaborar los fundamentos teóricos de la investigación.
- **Histórico y lógico:** Permitió ordenar cronológicamente la información analizada. Posibilitó expresar, en forma teórica, la esencia del objeto y la necesidad de la investigación.
- **Modelación:** Permitió modelar la solución para representar el proceso informatizado y brindar una mejor comprensión, obteniendo un producto de mayor calidad.

Los **métodos empíricos** utilizados son:

- **Revisión de documentos:** Fue utilizado para la recopilación de la información sobre el desarrollo de aplicaciones *Android* y la gestión de la información de especies florísticas en el Distrito Moense.
- **Entrevista:** Se empleó para conocer en detalles cómo era el trabajo de los especialistas en el CEMA y además para determinar los requerimientos que debe cumplir la aplicación a desarrollar.

La Aplicación *Android* propuesta facilitará la recolección de los datos que se gestionan en el Herbario Digital. Permitirá a los especialistas insertar los datos de las especies florísticas en el lugar donde se encuentran, así como obtener automáticamente la geolocalización. Además, permitirá la sincronización con la Base de Datos del Herbario Digital.

La presente investigación consta de introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

El Capítulo 1: Fundamentos teóricos de la Aplicación *Android* para la recolección de datos de especies florísticas, expresa un estudio bibliográfico actualizado con el propósito de dar cumplimiento al objetivo trazado, incluyendo una descripción

del objeto de estudio de la investigación, las principales tendencias y tecnologías para la construcción de la solución propuesta y la fundamentación de la metodología de Ingeniería de *Software* empleada.

El Capítulo 2: Desarrollo de la Aplicación *Android* para la recolección de datos de especies florísticas, detalla el proceso de diseño, implementación y prueba de la solución propuesta según la metodología de desarrollo empleada.

El Capítulo 3: Estudio de factibilidad de la Aplicación *Android* para la recolección de datos de especies florísticas, muestra un estudio de factibilidad según los aspectos técnico, económico y operativo.

Capítulo 1: Fundamentos teóricos de la Aplicación *Android* para la recolección de datos de especies florísticas

En el presente capítulo se exponen los fundamentos del objeto y campo de la investigación, como el proceso de inventario de especies florísticas del Distrito Moense, así como en qué consisten las aplicaciones para dispositivos móviles. Se tienen en cuenta aplicaciones que existen que pueden dar parte de solución al problema. Se analizan las tecnologías y herramientas disponibles para la construcción de la solución propuesta, haciendo énfasis en sus características, ventajas y desventajas y, se identifican las que se proponen para el desarrollo de esta investigación. Además, se fundamenta la metodología de desarrollo utilizada.

1.1 Proceso de inventario de especies florísticas en Moa

El Centro de Estudios del Medio Ambiente (CEMA) está ubicado en la Facultad de Geología y Minería del ISMMM, formando parte del Consejo Científico y Consejo de Dirección de esta facultad. Además, forma parte del Consejo de Dirección, Consejo Asesor para la Defensa y Consejo Científico del ISMMM.

La misión del Centro de Estudios está definida como:

“La integración del potencial universitario en la implementación de acciones coadyuven a crear la base científica para el desarrollo sostenible del territorio minero-metalúrgico a través de la actividad investigativa y la introducción de resultados, así como la incorporación de la dimensión ambiental en la esfera de la educación de posgrado, contribuyendo también a la formación del profesional y extensión universitaria.”[2]

El CEMA cuenta con un grupo de especialistas que dirigen proyectos de investigación que tributan a la conservación de la biodiversidad vegetal y animal. Para esto, los mismos se encargan de recoger información sobre las especies florísticas de Moa. Para gestionar y almacenar la información el Centro usa un Herbario Digital desarrollado en el año 2014 [1].

Un herbario digital es una página fundamentada con información de varias especies. Cada espécimen virtual está acompañado de información sobre el lugar y fecha de colección, autor, el nombre científico correcto, el nombre común y, en general, con información de las especies asociadas y preferencias ecológicas. Los herbarios virtuales nacieron como una herramienta de consulta gratuita a disposición de todos aquellos que quieran conocer las distintas plantas de una determinada región, su ecología, distribución, nomenclatura, taxonomía, y está dirigida a estudios científicos, organismos públicos, grupos ecologistas, asociaciones vinculadas a la naturaleza, o simplemente, a quienes quieran identificar las plantas a través de sus nombres [1].

1.1.1 Recolección de datos

Para recolectar los datos de las especies florísticas, los especialistas deben trasladarse a localidades intrincadas. Sin embargo, en las últimas décadas se hace cada vez más difícil para los especialistas cubanos y extranjeros, trasladarse hasta estas localidades por la lejanía y escasez de recursos económicos; dificultándoles la categorización de las especies y el desarrollo de un trabajo sostenible para conservar los recursos florísticos de esta región. También, los centros de referencia a nivel nacional se encuentran lejos de la localidad y de la provincia; limitándose así el trabajo profesional tanto de científicos cubanos como extranjeros.

Actualmente los especialistas recogen periódicamente la información referente a las especies en el campo. Deben cargar hasta lo más intrincado de las localidades un mapa, brújula, equipo de geolocalización (GPS), cámara de fotos digital, papel y lápiz; por tanto, este trabajo se hace muy engorroso a la hora de recoger dicha información.

1.2 Aplicaciones para dispositivos móviles

La tecnología móvil se está convirtiendo en una revolución dentro de nuestra sociedad. La adopción de las tecnologías móviles por parte de los ciudadanos y

las empresas es altísima y crece cada día, y esto provoca que la demanda de desarrolladores para dispositivos móviles en estos momentos sea muy elevada [3].

Una aplicación móvil es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tablets¹ y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como *Android*, *iOS*², *BlackBerry OS*, *Windows Phone*, entre otros. Existen aplicaciones móviles gratuitas u otras de pago, donde en promedio el 20-30% del costo de la aplicación se destina al distribuidor y el resto es para el desarrollador. El término *apk* se volvió popular rápidamente, tanto que en 2010 fue listada como Palabra del Año por la *American Dialect Society*.

El desarrollo de aplicaciones para dispositivos móviles requiere tener en cuenta las limitaciones de estos dispositivos. Los dispositivos móviles funcionan con batería y tienen procesadores menos poderosos que los ordenadores personales. Los desarrollos de estas aplicaciones también tienen que considerar una gran variedad de tamaños de pantalla, datos específicos de *software* y configuraciones.

El desarrollo de aplicaciones móviles requiere el uso de entorno de desarrollo integrado. Las aplicaciones móviles suelen ser probadas primero usando emuladores y más tarde se ponen en el mercado en período de prueba. Actualmente un gran número de empresas se dedica a la creación profesional de aplicaciones.

Las aplicaciones móviles tienen como ventajas que:

- generan una base de datos de clientes,
- permiten vender productos y servicios,
- brindan información directa a clientes,
- permite que la empresa esté disponible las 24 horas por los 365 días del año, en cualquier lugar,

¹ Tablet, tipo de computadora portátil con pantalla táctil.

² Sistema Operativo para *iPhone*

- brinda comodidad al cliente al adquirir los servicios de la empresa,
- facilita al cliente acceder a la información rápida y oportunamente [4].

1.3 Tecnologías y herramientas utilizadas

Para el desarrollo de la aplicación *Android* propuesta se analizaron tecnologías y herramientas que fueran libres y de código abierto para facilitar el proceso de desarrollo y en concordancia con la política del país. Se escogieron las más adecuadas para la implementación de la aplicación teniendo en cuenta las ventajas y desventajas y la experiencia del desarrollador, las mismas se describen en este epígrafe.

1.3.1 *Android*

Android es una solución completa de *software* de código libre para teléfonos y dispositivos móviles. Es un paquete que engloba un sistema operativo, un *runtime* de ejecución basado en *Java*, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final (todas ellas desarrolladas en *Java*). *Android* se distribuye bajo una licencia libre permisiva (Apache) que permite la integración con soluciones de código propietario.

Inicialmente fue desarrollado por *Android Inc.*, empresa que *Google* respaldó económicamente y más tarde, en 2005, compró. Fue presentado en 2007 como resultado de la *Open Handset Alliance*, un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en los dispositivos móviles y un compromiso de comercializar estos con este sistema operativo. Entre las compañías encontramos compañías de *software* (*Ebay*, *LivingImage*, etc.), operadores (*Telefónica*, *Vodafone*, *T-Mobile*, etc.), fabricantes de móviles (*Motorola*, *Samsung*, *Acer*, *LG*, *HTC*, etc.) o fabricantes de *hardware* (*nVidia*, *Intel* o *Texas Instruments*).

El sistema operativo *Android* está basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o *tablets*; y también para relojes inteligentes, televisores y automóviles. El primer móvil con el sistema operativo *Android* fue el *HTC Dream* y

se vendió en octubre de 2008. Los dispositivos de *Android* venden más que las ventas combinadas de *Windows Phone* e *iOS*. La versión básica de *Android* es conocida como *Android Open Source Project* (AOSP).

Android presenta una arquitectura basada en 4 niveles (Figura 1.1), que detallamos a continuación por orden ascendente:

- Un *kernel* Linux versión 2.6 que sirve como base de la pila de *software* y se encarga de las funciones más básicas del sistema: gestión de drivers, seguridad, comunicaciones, etc.
- Una capa de bibliotecas de bajo nivel en C y C++, como *SQLite* para persistencia de datos; la librería *OpenGL* para gestión de gráficos 3D, con aceleración 3D opcional y *Webkit* como navegador Web embebido y motor de renderizado HTML³.
- Un *framework* para el desarrollo de aplicaciones, dividido en subsistemas para gestión del sistema como el "Administrador de paquetes", el "Administrador de telefonía" (para la gestión del *hardware* del teléfono anfitrión) o el acceso a APIs⁴ sofisticadas de geolocalización o mensajería XMPP. Los desarrolladores tienen acceso completo a las mismas APIs del *framework* usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del *framework*). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario. También incluye un sistema de vistas para manejar el interfaz de usuario de las aplicaciones, que incluye posibilidad de visualización de mapas o renderizado HTML directamente en la interfaz gráfica de la aplicación.

³ *HyperText Markup Language* (Lenguaje de Etiquetación de Hipertexto)

⁴ *Application Programming Interface* (Interfaz de Aplicación de Programación)

- Aplicaciones: Las aplicaciones base incluyen un teléfono, cliente de *email*, programa de envío de SMS⁵, calendario, mapas, navegador, contactos, etc. que pueden a su vez ser usados por otras aplicaciones.

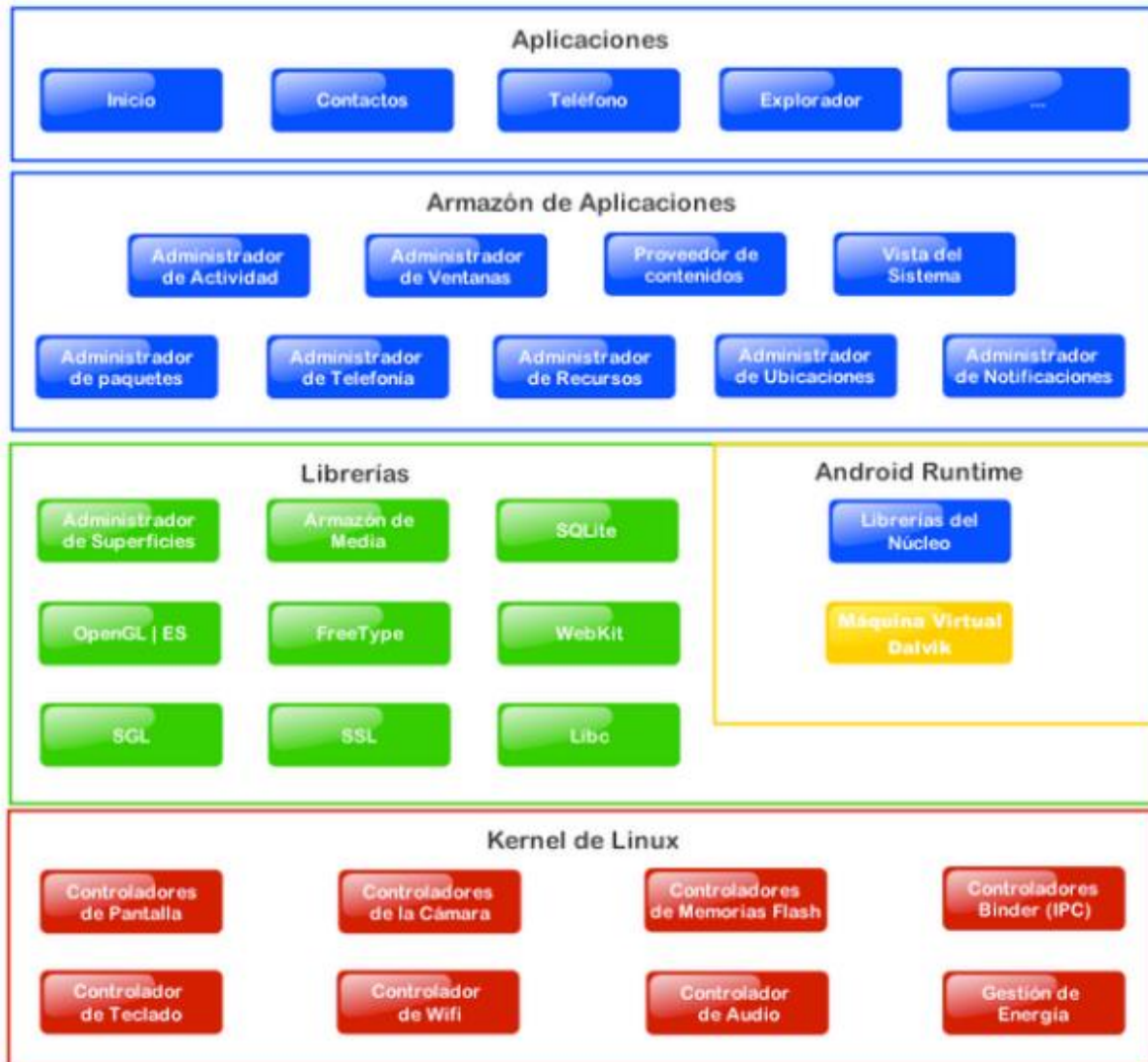


Figura 1.1: Arquitectura de *Android* [5].

Las aplicaciones *Android* están programadas en *Java*, pero no corriendo sobre la plataforma *Java Micro Edition* (*JavaME* o *J2ME*)⁶, sino sobre Dalvik, una máquina virtual (VM) *Java* desarrollada por Google y optimizada para dispositivos

⁵ *Short Message Service*

⁶ Plataforma *Java* para desarrollar aplicaciones para dispositivos móviles.

empotrados. La creación de una VM propia es un movimiento estratégico que permite a *Google* evitar conflictos con *Sun* por la licencia de la máquina virtual, así como asegurarse el poder innovar y modificar esta. Cada aplicación *Android* corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente [5].

1.3.2 Lenguaje de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones [6].

Java

Java es un lenguaje de programación y la primera plataforma informática creada por *Sun Microsystems* en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. *Java* se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.

Es un lenguaje independiente de plataforma con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

Ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable. Proporciona un conjunto de clases potente y flexible con facilidades de reutilización de código. Además, pone al alcance de cualquiera la utilización de aplicaciones que se pueden incluir directamente en páginas Web (aplicaciones denominadas *applets*). Fue diseñado

para crear *software* altamente fiable, para lo cual proporciona numerosas comprobaciones en compilación y en tiempo de ejecución [7].

1.3.3 Bases de Datos Móviles

La transición de los ficheros como almacenes de datos a la tecnología de Base de Datos (BD) dio lugar a diferencias tangibles. Henry F. Koth define BD como “una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar esos datos” [8].

Las BD móviles surgen en el siglo XXI como respuesta a las nuevas necesidades de asequibilidad y eficiencia en los dispositivos portátiles. Una base de datos móvil es una base de datos que puede ser instalada en un dispositivo de computación móvil a través de una red de este mismo tipo. El cliente y el servidor tienen conexiones inalámbricas, la memoria caché se mantiene para almacenar los datos frecuentes y transacciones de manera que no se pierdan debido a un fallo de conexión [9]. Se puede definir como una base de datos portable y físicamente independiente del servidor corporativo de bases de datos, pero que es capaz de comunicarse con ese servidor desde sitios remotos, permitiéndose compartir los datos corporativos [10].

Los Sistemas Gestores de Base de Datos Móviles (SGBDM) se encargan de recuperar la información de los sistemas de computación y/o repositorios de información sobre dispositivos móviles en cualquier momento en cualquier lugar; además de introducir o actualizar información en los sistemas principales de forma remota desde el dispositivo móvil. Esto da toda la libertad al usuario de manipular toda la información desde lejos. Son un sistema distribuido que soporta conectividad móvil, posee todas las capacidades de un sistema de base de datos y permiten a las unidades móviles una completa movilidad espacial por medio de la tecnología inalámbrica.

Muchos fabricantes ofrecen SGBDM capaces de comunicarse con los principales sistemas gestores de BD relacionales. Estos SGBDM están adaptados a los

recursos limitados de las unidades móviles y proporcionan una serie de funcionalidades adicionales:

- Comunicación con el servidor centralizado de base de datos mediante técnicas de comunicación inalámbrica.
- Replicación de datos en el servidor centralizado de base de datos y en el dispositivo móvil.
- Sincronización de datos entre el servidor centralizado de base de datos y el dispositivo móvil.
- Gestión de datos en el dispositivo móvil.
- Análisis de los datos almacenados en el dispositivo móvil [9].

En una base de datos móvil, la comunicación entre los dispositivos es una parte importante, pues es imprescindible una buena comunicación para el acceso a los datos. Esto se evidencia en su arquitectura, la cual se muestra en la Figura 1.2.

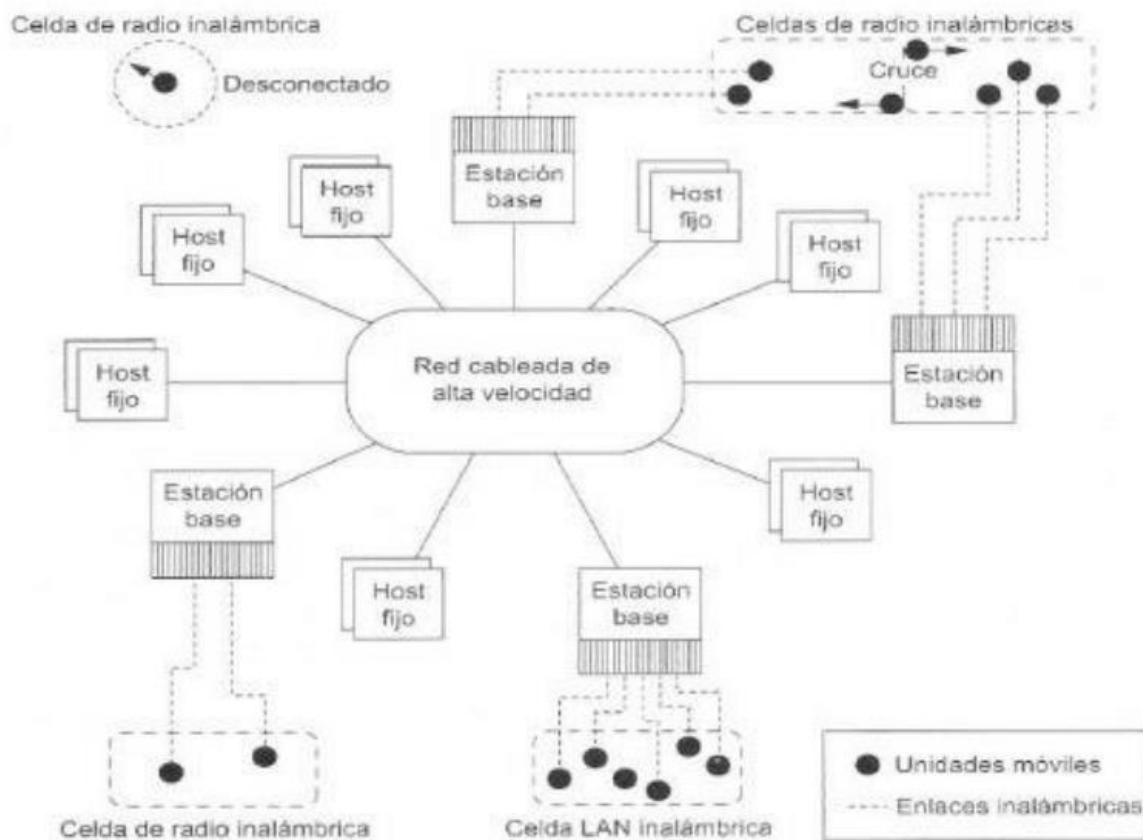


Figura 1.2: Arquitectura de Bases de Datos Móviles [10].

En algunos casos el usuario de un dispositivo móvil puede conectarse a un servidor de base de datos corporativos gracias a los agentes móviles y trabajar allí con los datos, mientras que en otros el usuario puede descargar los datos y trabajar con ellos en un dispositivos móvil, es decir, varias aplicaciones pueden tener acceso simultáneo a la información compartida. Esta característica permite a los usuarios estar en una sincronización con la base de datos corporativa en diferentes ubicaciones geográficas.

La arquitectura general de una plataforma móvil es un modelo distribuido formado por computadores fijos, estaciones base y unidades móviles. La complejidad de este tipo de sistemas está dada por el sistema gestor de base de datos móvil y el enlace de comunicaciones, como se muestra en la Figura 1.3.

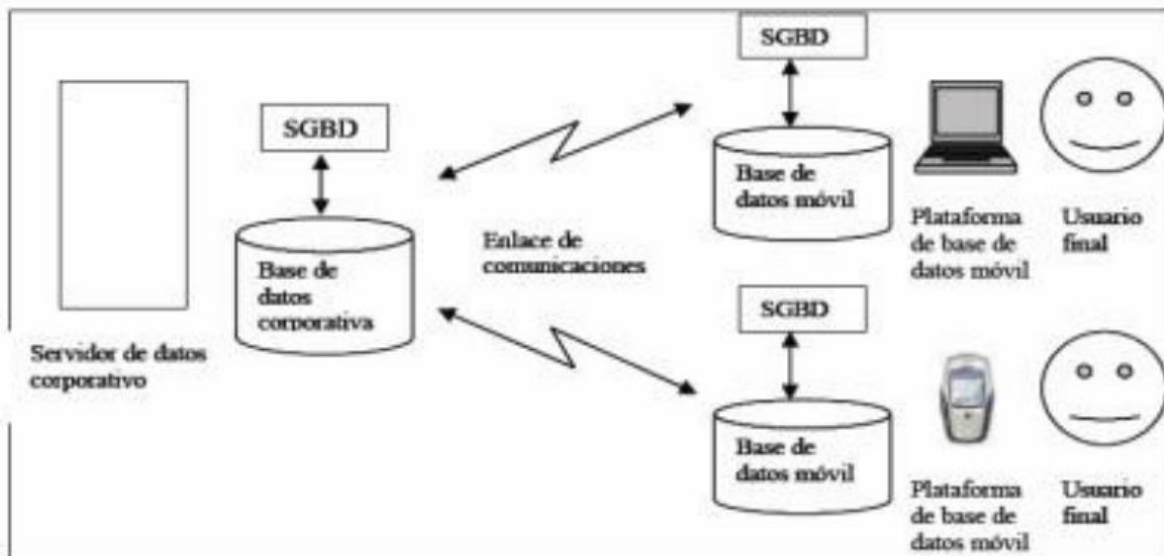


Figura 1.3: Arquitectura de Bases de Datos Móviles mostrando al Sistema Gestor de Base de Datos Móvil y al enlace de comunicaciones [11].

Algunos ejemplos de los SGBDM más utilizados son: *SQL Anywhere*, *DB2 Everyplace*, *SQL Server Compact* y *Oracle Database Lite 10g*.

SQL Anywhere, de *Anywhere Solutions*, empresa filial de *Sybase*, lidera el ranking del mercado de bases de datos móviles. Este paquete proporciona bases de datos que pueden utilizarse tanto a nivel de servidor como a nivel de dispositivo móvil. *SQL Anywhere* se compone de las siguientes tecnologías:

- *SQL Anywhere Server*: sistema gestor de bases de datos relacionales para los sistemas de bases de datos móviles.
- *Ultralite*: sistema gestor de bases de datos que puede embeberse en dispositivos móviles.
- *Mobilink*: tecnología de sincronización para el intercambio de datos entre bases de datos relacionales y bases de datos no relacionales. *QAnywhere*: facilita el desarrollo de aplicaciones móviles robustas y seguras.
- *SQL Remote*: permite a los usuarios de dispositivos móviles sincronizar sus datos con otras bases de datos *SQL Anywhere*.

DB2 Everyplace, de IBM⁷, es una base de datos relacional y un servidor de sincronización que permite extender las aplicaciones y los datos empresariales a dispositivos móviles. Gracias a un consumo de recursos reducido, esta base de datos puede integrarse en dispositivos como PDAs⁸ y teléfonos móviles.

SQL Server Compact, de *Microsoft*, es un motor de bases de datos que permite desarrollar aplicaciones en cualquier plataforma *Windows* incluyendo *Tablets*, *PocketPCs*⁹, *Smartphones*¹⁰ y equipos de escritorio.

Oracle Database Lite 10g es la solución de *Oracle* para desarrollar aplicaciones en entornos móviles. Proporciona un cliente que permite la realización de consultas SQL¹¹ para acceder a los datos locales del dispositivo y un servidor para gestionar los datos de forma centralizada.

Entre las BD móviles con licencia libre y de código abierto se encuentran *SQLite* y la BD de *Java Micro Edition*.

⁷ *International Business Machines*, Compañía Internacional de Máquinas de Oficina, fabrica y comercializa *hardware* y *software* para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet y consultoría informática.

⁸ *Personal Digital Assistant*, computadora de bolsillo.

⁹ También llamado PDA

¹⁰ Teléfonos inteligentes

¹¹ *Structured Query Language*, Lenguaje Estructurado de Consulta para Bases de Datos.

La plataforma *JavaME* incluye un mecanismo de almacenamiento de datos llamado *Record Management System* (RMS), BD formada por almacenes de registros (Record Store), donde un registro es un conjunto de bytes. Presenta algunos problemas como que un almacén de registros sólo puede ser utilizado por el programa que lo crea; no tiene lenguaje de descripción de datos ni lenguaje de manipulación de datos; y solo permite guardar y recuperar bytes [9, 10].

Se decidió el uso del SGBDM *SQLite* principalmente por ser una solución libre y su compatibilidad con las aplicaciones *Android*. Las características de este gestor se describen a continuación.

SQLite

SQLite es un ligero motor de bases de datos de código abierto y de dominio público, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla. Se usa en aplicaciones *Android*. A diferencia de otros Sistemas Gestores de BD como *MySQL*, *SQL Server* y *OracleDB*, *SQLite* tiene las siguientes ventajas:

- No requiere el soporte de un servidor: *SQLite* no ejecuta un proceso para administrar la información, si no que implementa un conjunto de librerías encargadas de la gestión.
- No necesita configuración: Libera al programador de todo tipo de configuraciones de puertos, tamaños, ubicaciones, etc.
- Usa un archivo para el esquema: Crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad, ya que los datos de las aplicaciones *Android* no pueden ser accedidos por contextos externos.
- Es de código abierto: Está disponible al dominio público de los desarrolladores al igual que sus archivos de compilación e instrucciones de escalabilidad.

Es por eso que *SQLite* es una tecnología cómoda para los dispositivos móviles. Su simplicidad, rapidez y usabilidad permiten un desarrollo muy amigable.

No obstante, *SQLite* tiene ciertas limitaciones en algunas operaciones. Por ejemplo, no se puede implementar las cláusulas *FULL OUTER JOIN* y *RIGHT OUTER JOIN*, aunque en la mayoría de casos esto no afectará [12].

Android incluye una librería de *SQLite* que permite crear bases de datos relacionales, navegar entre las tablas, ejecutar sentencias en SQL y otras funcionalidades propias del sistema *SQLite*. La base de datos resultante puede ser accedida desde el código de la aplicación como si de un objeto más se tratara, gracias a las clases contenidas en el paquete *Android.database.SQLite* [13].

1.3.4 Servicios Web

El diseño del software tiende a ser cada vez más modular. Las aplicaciones se componen de una serie de componentes (servicios) reutilizables, que pueden encontrarse distribuidos a lo largo de una serie de máquinas conectadas en red.

Los Servicios Web surgieron a principios de los años 90 para solucionar el problema de la interoperabilidad entre las aplicaciones, problema acrecentado con el desarrollo de Internet, LAN¹² y WAN¹³. La razón de que se llamasen Servicios Web es que fueron diseñados para residir en un servidor Web, y ser llamados a través de Internet, típicamente vía Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés, *Hypertext Transfer Protocol*) o Protocolo Seguro de Transferencia de Hipertexto (HTTPS, por sus siglas en inglés, *Hypertext Transfer Protocol Secure*). De esta forma se asegura que un servicio puede ser llamado por cualquier aplicación, usando cualquier lenguaje de programación, y bajo cualquier sistema operativo, siempre y cuando, por supuesto, la conexión a Internet esté activa y tenga un puerto abierto HTTP/HTTPS, lo cual es cierto para casi cualquier computador que disponga de acceso a Internet.

¹² Red de Área Local

¹³ Red de Área Amplia

Los Servicios Web permiten distribuir una aplicación, pudiendo utilizar los servicios ofrecidos por cualquier servidor. Su punto clave es la mencionada interoperabilidad entre las aplicaciones.

Un Servicio Web es un componente software distribuido al que se puede acceder mediante protocolos Web estándares, utilizando el Lenguaje de Marcado Extensible (XML, por sus siglas en inglés, *eXtensible Markup Language*) para el intercambio de información, por lo que es independiente del lenguaje de programación y de la plataforma, siendo el aspecto más interesante. Es una colección de procedimientos (métodos) a los que se puede acceder a través de la red, Internet o Intranet [14, 15].

Los Servicios Web son aplicaciones modulares autodescriptivas. Su arquitectura se describe como el envoltorio del código de aplicación. Este envoltorio proporciona medios estandarizados para la descripción de los Servicios Web y su función [16].

Un Servicio Web puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Permite exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos Web estándar. Cada Servicio Web se responsabiliza de realizar un conjunto de funciones concretas y bien definidas. Los Servicios Web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

El *World Wide Web Consortium* (W3C) define como un Servicio Web (*Web Service*), “una aplicación software identificada por un URI (*Uniform Resource Identifier*), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre agentes software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.”

El concepto clave “servicio”, se refiere a un procedimiento, un método o un objeto con una interfaz estable y pública que puede ser invocado por un cliente [17].

Las características deseables de un Servicio Web son:

- Un servicio debe poder ser accesible a través de la Web. Para ello debe utilizar protocolos de transporte estándares como HTTP, y codificar los mensajes en un lenguaje estándar que pueda conocer cualquier cliente que quiera utilizar el servicio.
- Un servicio debe contener una descripción de sí mismo. De esta forma, una aplicación podrá saber cuál es la función de un determinado Servicio Web, y cuál es su interfaz, de manera que pueda ser utilizado de forma automática por cualquier aplicación, sin la intervención del usuario.
- Debe poder ser localizado. Se debe tener algún mecanismo que permita encontrar un Servicio Web que realice una determinada función. De esta forma se tiene la posibilidad de que una aplicación localice el servicio que necesite de forma automática, sin tener que conocerlo previamente el usuario.

Los servicios Web presentan una Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) que permite crear una definición abstracta de un servicio, proporcionar una implementación concreta de dicho servicio, publicar y localizar un servicio, seleccionar un instancia de un servicio, y utilizar dicho servicio con una elevada interoperabilidad. Es posible desacoplar la implementación del servicio Web y su uso por parte de un cliente. También es posible desacoplar la implementación del servicio y del cliente. Las implementaciones concretas del servicio pueden desacoplarse a nivel de lógica y transporte. La Figura 1.4 muestra el diagrama de una arquitectura orientada a servicios.

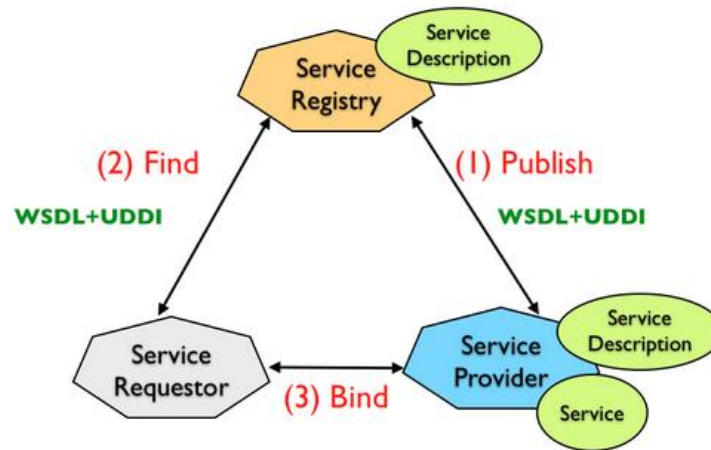


Figura 1.4: Arquitectura Orientada a Servicios [14].

El proveedor del servicio define la descripción abstracta de dicho servicio utilizando un Lenguaje de Descripción de Servicios Web (WSDL, por sus siglas en inglés, *Web Services Description Language*), que es un lenguaje basado en XML para definir las interfaces sintácticamente. A continuación se crea un Servicio concreto a partir de la descripción abstracta del servicio, produciendo así una descripción concreta del servicio en WSDL. Dicha descripción concreta puede entonces publicarse en un servicio de registro como por ejemplo UDDI (*Universal Description, Discovery and Integration*). Un cliente de un servicio puede utilizar un servicio de registro para localizar una descripción de un servicio, a partir de la cual podrá seleccionar y utilizar una implementación concreta de dicho servicio.

Los Servicios Web usan una serie de tecnologías, todas ellas basadas en XML, que son fundamentales para el desarrollo de servicios. Estas tecnologías son independientes tanto del sistema operativo como del lenguaje de programación utilizado para implementar dichos servicios. Los protocolos utilizados en los Servicios Web se organizan en cuatro capas (Figura 1.5):

- Transporte de servicios: Es la capa que se encarga de transportar los mensajes entre aplicaciones. Normalmente se utiliza el protocolo HTTP para este transporte, aunque los servicios web pueden viajar mediante otros protocolos de transferencia de hipertexto como SMTP, FTP o BEEP.

- Mensajería XML: Es la capa responsable de codificar los mensajes en XML de forma que puedan ser entendidos por cualquier aplicación. Puede implementar los protocolos XML-RPC¹⁴ o SOAP.
- Descripción de servicios: Se encarga de definir la interfaz pública de un determinado servicio. Esta definición se realiza mediante WSDL.
- Localización de servicios: Se encarga del registro centralizado de servicios, permitiendo que estos sean anunciados y localizados. Para ello se utiliza el protocolo UDDI.

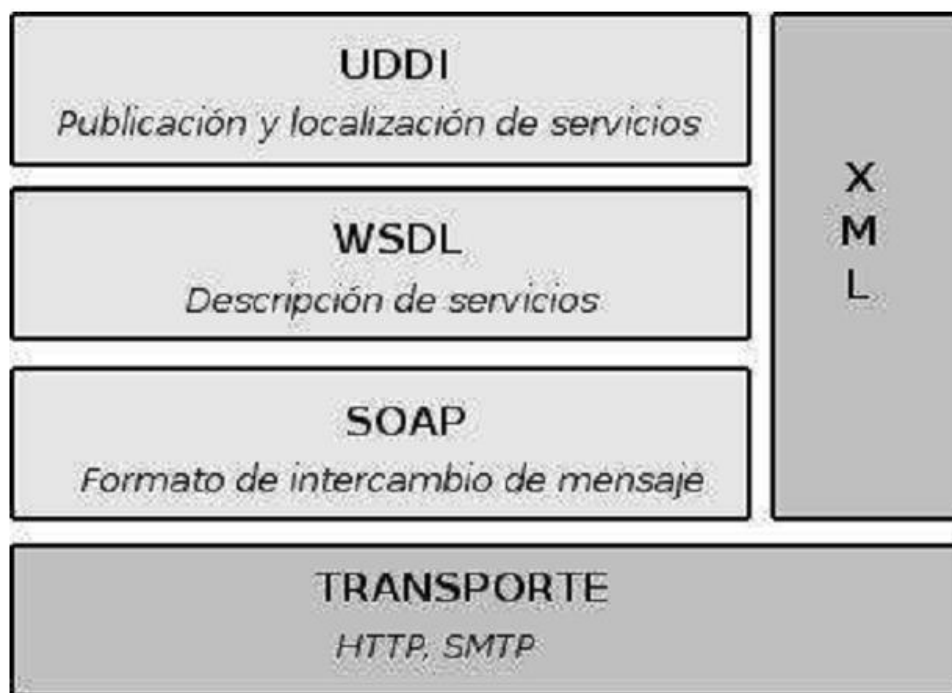


Figura 1.5: Elementos de los Servicios Web [17].

Los servicios productores y consumidores utilizan mensajes para intercambiar información de invocaciones de petición y respuesta en forma de documentos auto-contenidos que hacen muy pocas asunciones sobre las capacidades tecnológicas de cada uno de los receptores. En una Arquitectura Orientada a Servicios, cualquier interacción punto a punto implica dos *endpoints*: uno que proporciona un servicio, y otro de lo consume. Es decir, que un *endpoint* es cada

¹⁴ Remote Procedure Calls

uno de los elementos, en este caso servicios, que se sitúan en ambos extremos de la red que sirve de canal de comunicación entre ellos. Cuando se habla de servicios Web, un *endpoint* se especifica mediante una URI.

A nivel técnico, los servicios pueden implementarse de varias formas. En este sentido, se distinguen dos tipos de servicios Web: los denominados Servicios Web grandes (*big Web Services*) o Servicios Web SOAP y Servicios Web *RESTful*.

Los Servicios Web *RESTful* (*Representational State Transfer Web Services*) son adecuados para escenarios de integración básicos *ad-hoc*. Dichos servicios se suelen integrar mejor con HTTP que los basados en SOAP, ya que no requieren mensajes XML o definiciones del servicio en forma de fichero WSDL. Utilizan estándares muy conocidos como HTTP, SML, URI, MIME, y tienen una infraestructura ligera que permite que los servicios se construyan utilizando herramientas de forma mínima. Gracias a ello, el desarrollo de servicios *RESTful* es barato y tiene muy pocas barreras para su adopción [14].

La aplicación propuesta propone la implementación de un servicio Web para exportar la BD, para lo cual se empleó el estándar SOAP, debido a las características descritas a continuación, principalmente el uso del estándar XML para favorecer la interoperabilidad y su fácil integración con el Eclipse.

SOAP

SOAP (*Simple Object Access Protocol*) es un protocolo o estándar que define un lenguaje XML con la arquitectura y formato de los mensajes de intercambio en el uso de un *Web Service*. También soporta llamadas del tipo RPC. Adicionalmente, es posible mediante SOAP definir un mensaje HTTP y este punto es de especial interés puesto que el protocolo imprescindible para Internet es HTTP [18].

Los Servicios Web SOAP utilizan mensajes XML para intercomunicarse que siguen el estándar SOAP. Dichos sistemas normalmente contienen una descripción legible por la máquina de la descripción de las operaciones ofrecidas por el servicio, escrita en WSDL.

El formato de mensaje SOAP y el lenguaje de definición de interfaces WSDL se ha extendido bastante, y muchas herramientas de desarrollo, por ejemplo *Netbeans* y *Eclipse*, pueden reducir la complejidad de desarrollar aplicaciones de servicios web.

El diseño de un servicio basado en SOAP debe establecer un contrato formal para describir la interfaz que ofrece el servicio web. WSDL puede utilizarse para describir los detalles del contrato, que pueden incluir mensajes, operaciones, *bindings* y la localización del servicio web. También deben tenerse en cuenta los requerimientos no funcionales, como por ejemplo las transacciones, necesidad de mantener el estado (*addressing*), seguridad y coordinación [14].

Normalmente se utiliza SOAP para conectarse a un servicio e invocar métodos remotos, aunque puede ser utilizado de forma más genérica para enviar cualquier tipo de contenido. Se puede distinguir dos tipos de mensajes según su contenido:

- **Mensajes orientados al documento:** Contienen cualquier tipo de contenido que queramos enviar entre aplicaciones.
- **Mensajes orientados a RPC:** Este tipo de mensajes servirá para invocar procedimientos de forma remota. Puede verse como un tipo más concreto dentro del tipo anterior, ya que en este caso como contenido del mensaje se especifica el método que se quiere invocar junto a los parámetros que le pasan, y el servidor deberá devolver como respuesta un mensaje SOAP con el resultado de la invocación del el método.

SOAP puede ser utilizado sobre varios protocolos de transporte, aunque está especialmente diseñado para trabajar sobre HTTP. Dentro del mensaje SOAP se pueden distinguir los siguientes elementos (Figura 1.6):

- Un sobre (*Envelope*), que describe el mensaje, a quien va dirigido, y cómo debe ser procesado. El sobre incluye las definiciones de tipos que se usarán en el documento. Contiene una cabecera de forma opcional, y el cuerpo del mensaje.

- Una cabecera (*Header*) opcional, donde se puede incluir información sobre el mensaje. Por ejemplo, se puede especificar si el mensaje es obligatorio (debe ser entendido de forma obligatoria por el destinatario), e indicar los actores (lugares por donde ha pasado el mensaje).
- El cuerpo del mensaje (*Body*), que contiene el mensaje en sí. En el caso de los mensajes RPC se define una convención sobre cómo debe ser este contenido, en el que se especificará el método al que se invoca y los valores que se pasan como parámetros. Puede contener un error de forma opcional.
- Un error (*Fault*) en el cuerpo del mensaje de forma opcional. Servirá para indicar en una respuesta SOAP que ha habido un error en el procesamiento del mensaje de petición que se manda.
- El anexo (*Attachment*), puede contener cualquier tipo de contenido (incluido el XML). De esta forma se puede enviar cualquier tipo de contenido junto a un mensaje SOAP, como puede ser una imagen.



Figura 1.6: Elementos de un mensaje SOAP con anexos.

1.3.5 Entorno de Desarrollo Integrado

Para el desarrollo de la aplicación propuesta como solución se utilizó un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés, *Integrated Development Environment*). Un IDE no es más que un programa informático compuesto por un conjunto de herramientas de programación, que puede dedicarse en exclusiva a un sólo lenguaje de programación o bien permitir utilizar varios.

Existen varios IDEs libres para desarrollar con *Java*, siendo los más populares y robustos NetBeans y Eclipse. Se escogió el IDE Eclipse por el soporte que brinda para el desarrollo de aplicaciones *Android*.

Eclipse

Eclipse es un IDE de código abierto y multiplataforma para desarrollar lo que el proyecto llama "aplicaciones de cliente enriquecido", opuesto a las aplicaciones "cliente- liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDEs. El proyecto Eclipse define su *software* como: " una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular".

La Plataforma de Cliente Enriquecido (RCP, por sus siglas en inglés, *Rich Client Platform*) está compuesta por los siguientes componentes:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para establecer relaciones estándar
- El Standard Widget Toolkit (SWT, por sus siglas) - Una herramienta *widget*¹⁵
- portable
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes.

¹⁵ En el contexto de la programación de aplicaciones visuales, los *widgets* tienen un amplio significado como componente gráfico o control visual que el programador reutiliza, con el cual el usuario interactúa. Puede ser la ventana de una interfaz gráfica. Suelen reunirse varios *widgets* en juegos de herramientas de *widgets* que son usadas por los programadores para construir interfaces gráficas de usuario.

El IDE de Eclipse emplea módulos para proporcionar toda su funcionalidad a la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos hace a la plataforma ligera para componentes de *software* .

Android ofrece un *plugin* para Eclipse que extiende la funcionalidad de este y facilita el desarrollo de aplicaciones para *Android*. Entre las funcionalidades de este *plugin* se encuentra:

- Emulador de *Android*. Permite elegir entre distintos terminales móviles y la versión del sistema operativo.
- El acceso a herramientas de desarrollo de *Android* como tomar capturas de pantalla, la redirección de puertos, la posibilidad de depurar con puntos de parada o ver el estado de las hebras y los procesos corriendo en el sistema.)
- Asistentes para la creación rápida de aplicaciones *Android*
- Editores de código para los distintos archivos de configuración (XML) que facilitan su comprensión y desarrollo
- Interfaces gráficas que permiten el desarrollo de componentes visualmente [5].

1.4 Metodología de desarrollo de *software*

En el proceso de desarrollo de *software* intervienen innumerables variables de las más diversas naturalezas, algunas con comportamiento sumamente difuso o imprevisible, que midan constantemente el avance hacia el éxito y hacen de sus rutinas y decisiones tareas altamente riesgosas, difíciles para controlar la calidad y eficiencia y cuantificar su eficacia. Desde hace bastante tiempo existe una alternativa: el uso de una metodología, que no es más que un conjunto de procedimientos para la realización de un nuevo *software* [19], que impone un proceso disciplinado con el fin de hacerlo más predecible y eficiente [20]. Consiste en un lenguaje de modelamiento y un proceso. El lenguaje de modelamiento es la notación gráfica, que incluye diferentes tipos de diagramas. El proceso define quién debe hacer qué, cuándo y cómo alcanzar un objetivo [21].

En los últimos tiempos han cobrado auge las metodologías ágiles para el desarrollo de *software*, con el objetivo de minimizar la documentación de desarrollo, considerando el código fuente la parte más importante y el *software* que funciona como la principal medida del progreso. Estas metodologías surgieron como reacción a las tradicionales, que son más burocráticas y pesadas [22].

No obstante, el escenario tecnológico actual con el desarrollo de los dispositivos móviles y sus aplicaciones, ha llevado al surgimiento de metodologías para el desarrollo de aplicaciones móviles, que faciliten la creación de estas con alto nivel de calidad. El desarrollo de aplicaciones para proveer servicios móviles, difiere del desarrollo de *software* tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos móviles, también difieran de las del *software* clásico.

1.4.1 Metodología para el Desarrollo de Aplicaciones Móviles

Se propone el uso de la Metodología para el Desarrollo de Aplicaciones Móviles (MDAM). La misma se fundamenta en la experiencia de investigaciones previas en aplicaciones móviles, la evaluación del potencial de éxito para servicios de tercera generación denominada 6M, la ingeniería de *software* educativo con modelado orientado por objetos (ISE-OO), y principalmente en los valores de las metodologías ágiles [23].

La metodología se encuentra enmarcada en cinco fases como se muestra en la Figura 1.2, denominadas: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega.

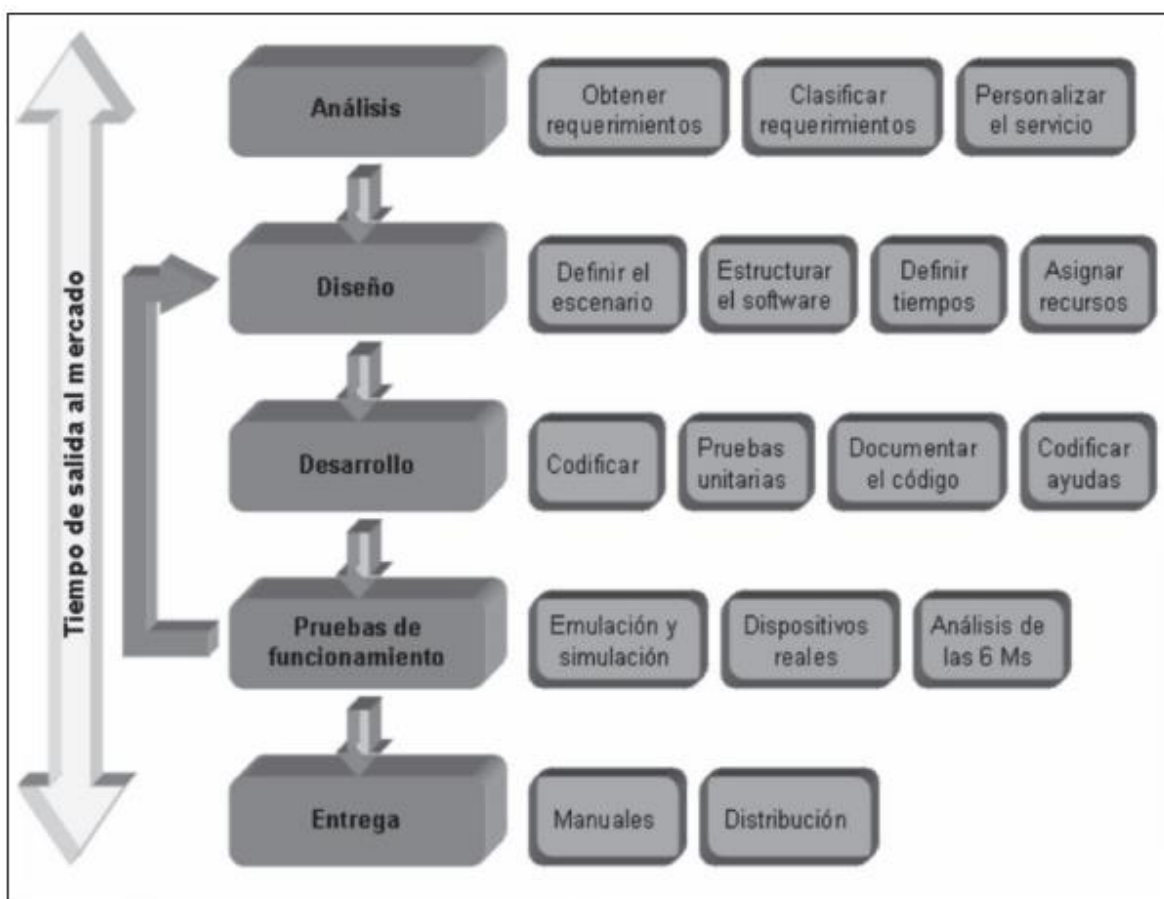


Figura 1.2: Etapas de la metodología MDAM.

Conclusiones del capítulo

En este capítulo se describió el proceso de gestión de información de las especies florísticas del Distrito Moense en el CEMA, así como el desarrollo de aplicaciones *Android* para dispositivos móviles, identificando las deficiencias que dieron origen a la investigación.

Se analizaron tecnologías y herramientas libres disponibles para el desarrollo de la aplicación propuesta, escogiendo *Android* como sistema operativo, *Java* como lenguaje de programación, *SQLite* como gestor de BD y Eclipse como entorno de desarrollo.

Se decidió el empleo de la metodología para el desarrollo de aplicaciones móviles (MDAM), para facilitar el desarrollo de la aplicación propuesta con un alto nivel de calidad.

Capítulo 2: Desarrollo de la Aplicación *Android* para la recolección de datos de especies florísticas

En este capítulo se especifican los requerimientos funcionales y no funcionales que debe cumplir la aplicación propuesta. Además, se describe el proceso de desarrollo según las etapas de la metodología de desarrollo empleada. La metodología empleada MDAM [23] cuenta con cinco etapas o fase: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega.

2.1 Análisis

En la fase Análisis se analizan las peticiones o requerimientos de las personas o entidad para la cual se desarrolla el servicio móvil “Cliente”, el propósito es definir las características del mundo o entorno de la aplicación. Se realizan las siguientes tareas: obtener requerimientos, clasificar los requerimientos y personalizar el servicio [23].

2.1.1 Requerimientos

Los requerimientos se definieron en conjunto con el cliente. Luego se clasificaron en funcionales y no funcionales, los cuales se muestran a continuación.

Requerimientos funcionales

R1: Gestionar los datos que describen una Especie florística

1.1: Insertar especie

1.2: Modificar especie

1.4: Eliminar especie

R2: Listar Especie

2.1: Filtrar por campo nombre común

2.2: Vista previa

R3: Capturar Imagen

3.1: Capturar imagen con la cámara del dispositivo

3.2: Insertar Imagen almacenada en el dispositivo

R4: Capturar Geolocalización

4.1: Tomar los datos de Latitud y Longitud del GPS del dispositivo móvil

4.2: La entrada manual de Latitud y Longitud

R5: Exportar Base de Datos

R6: Eliminar todos los datos de la Base de Datos

Requisitos no funcionales

Apariencia o Interfaz externa:

- La aplicación debe ser fácil de usar.
- El diseño debe ser agradable y atractivo.
- La aplicación debe ser interactiva, por lo que de no poderse ejecutar una acción se visualizará un mensaje de error.

Usabilidad:

- El diseño de la aplicación debe ser lo más sencillo y óptimo posible para agilizar el tiempo de respuesta del mismo.

Rendimiento:

- El tiempo de respuesta debe ser corto, por lo que el procesamiento de los datos se debe efectuar de forma rápida y eficiente.
- La aplicación debe ser diseñada específicamente para *Android* y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.

Soporte:

- La aplicación debe ser fácil de probar para facilitar las tareas en la etapa de Prueba.
- La aplicación podrá ser extendida y reutilizada, a partir de las características de modularidad del lenguaje utilizado.
- La aplicación debe ser fácil de instalar o implantar.

- La aplicación debe ser fácil de mantener para posibilitar su perfeccionamiento.

Portabilidad:

- Las herramientas y tecnologías utilizadas para el desarrollo de la aplicación deben ser *software* libre y de código abierto.

Confiabilidad:

- Ante cualquier fallo la aplicación debe mostrar un mensaje de error.

Software:

- La aplicación deberá ejecutarse en dispositivos con sistema operativo *Android* 2.3 o superior.

Hardware:

- La aplicación deberá ejecutarse en dispositivos con requerimientos mínimos de un procesador a 1.0 GB y 512 MB de RAM.

Estabilidad:

- Se debe mantener la actualización sistemática en el Herbario Digital pública para todos los consumidores.

Distribución:

- De forma gratuita al mercado.

2.1.2 Personalización del servicio

Teniendo en cuenta las preferencias, costumbres y particularidades del usuario, se definieron los siguientes requerimientos con el propósito de garantizar la aceptación del producto final:

- Mostrar una interfaz similar al Herbario Digital.
- Utilizar colores en la gama del negro, gris, verde y blanco para mayor compatibilidad con los colores del Herbario Digital y el CEMA. Las tonalidades serán suaves y relajantes, para evitar esfuerzo visual.

- Utilizar el idioma Español.
- Utilizar las opciones de la cámara como lugar de almacenamiento, calidad de la imagen, etc.

2.2 Diseño

El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa prueba de funcionamiento. Se realizan cuatro actividades en esta fase: definir el escenario, estructurar el *software*, definir tiempos y asignar recursos.

2.2.1 Definición del escenario

Las aplicaciones móviles se pueden diseñar para ejecutarse en diferentes escenarios, dependiendo del sistema de conexión y sincronización con el servidor o aplicación central; el proceso de sincronización se realiza para insertar, modificar o borrar información. Entre los diferentes escenarios se encuentran los siguientes:

1. Desconectado: Los procesos se realizan en el dispositivo móvil desconectado, después de terminar el proceso, si se requiere, puede conectarse con una aplicación central mediante el proceso de sincronización.
2. Semiconectado: Los procesos pueden ejecutarse en el dispositivo móvil desconectado, pero se requiere establecer conexión en algún momento para terminar el proceso, al sincronizar la información con el servidor o aplicación central. En los escenarios desconectado y semiconectado se recomienda utilizar los protocolos y tecnologías que se ajusten al servicio y capacidades tecnológicas del dispositivo.
3. Conectado: el dispositivo debe estar siempre conectado con la aplicación central o servidor para su correcto funcionamiento, no se almacenan datos o archivos en el móvil, la sincronización se realiza mediante la validación de formularios, usualmente se utiliza el protocolo HTTP [23].

La aplicación *Android* propuesta se encuentra en el escenario **desconectado**, ya que funciona en el dispositivo móvil sin conectarse a otro dispositivo o sistema. En el momento que se requiera se podrá conectar mediante un servicio Web a la base de Datos del Herbario Digital para sincronizar los datos.

2.2.2 Estructura del software

En este paso se deben utilizar algunos diagramas de Lenguaje de Modelado Unificado (UML, por sus siglas en inglés, *Unified Modeling Language*), según las necesidades del proyecto, modelando el sistema desde varias perspectivas (Figura 2.1).

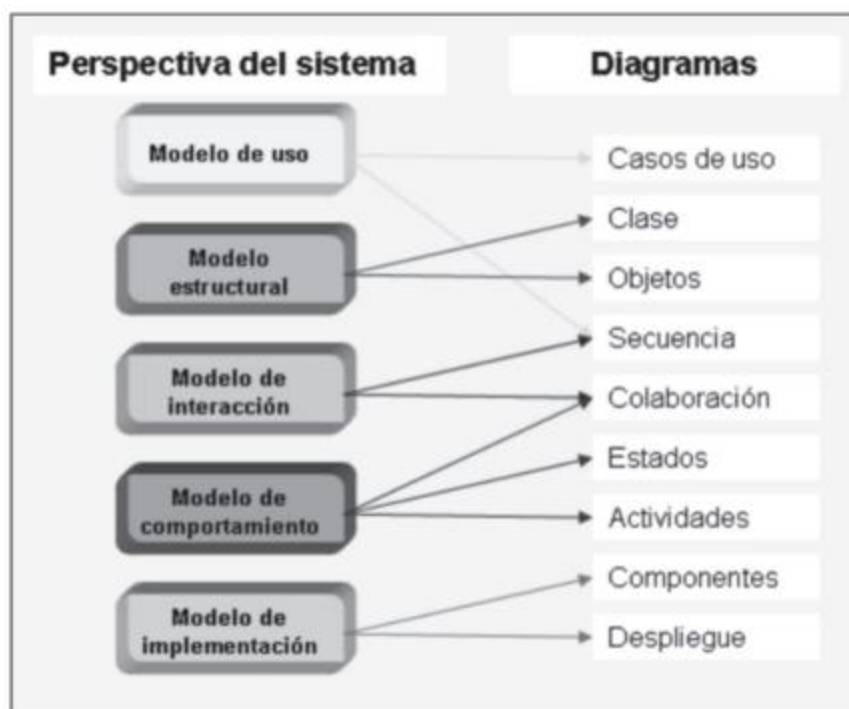


Figura 2.1: Posibles diagramas para el desarrollo de aplicaciones móviles.

Para modelar la aplicación y representar las funcionalidades de la misma se escogieron los siguientes diagramas. Del modelo de uso el diagrama de casos de uso; del modelo estructural el diagrama de clases, el diagrama de secuencia del modelo de interacción y el diagrama de despliegue del modelo de implementación.

Modelo de Casos de Uso

A partir de los requerimientos funcionales capturados se desarrolla el Modelo de Casos de Uso, que comprende la identificación de los actores y casos de uso del sistema. Los actores representan entidades externas al sistema, personas o sistemas, y son análogos a un rol del usuario [24]. Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario [25]. El diagrama de casos de uso de la aplicación se muestra en la Figura 2.2.

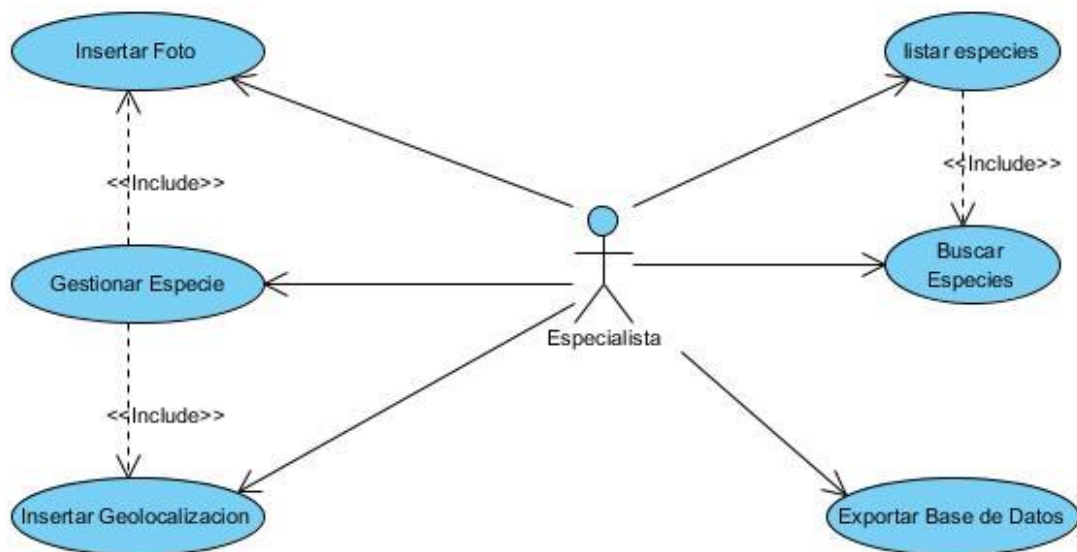


Figura 2.2: Diagrama de Casos de Uso.

La descripción de cada caso de uso facilita especificar la secuencia de acciones que el sistema debe llevar a cabo al interactuar con los actores, incluyendo

cursos alternativos dentro de la secuencia. Se realizó esta tarea utilizando voz activa y empleando una estructura de oración sustantivo-verbo-sustantivo; reflejando los cursos básicos y alternos en caso de existir, con la acción del actor y la respuesta del sistema. Las descripciones textuales de cada caso de uso se muestran en el Anexo 1.

Modelo de Interacción

Los diagramas de secuencia muestran el flujo de actividades del sistema a partir de un diseño detallado de este. Se realizan uno por cada caso de uso y en total concordancia con su descripción textual [26]. El diagrama de secuencia del caso de uso Gestionar Especie se muestra en la Figura 2.3, los demás se muestran en el Anexo 2.

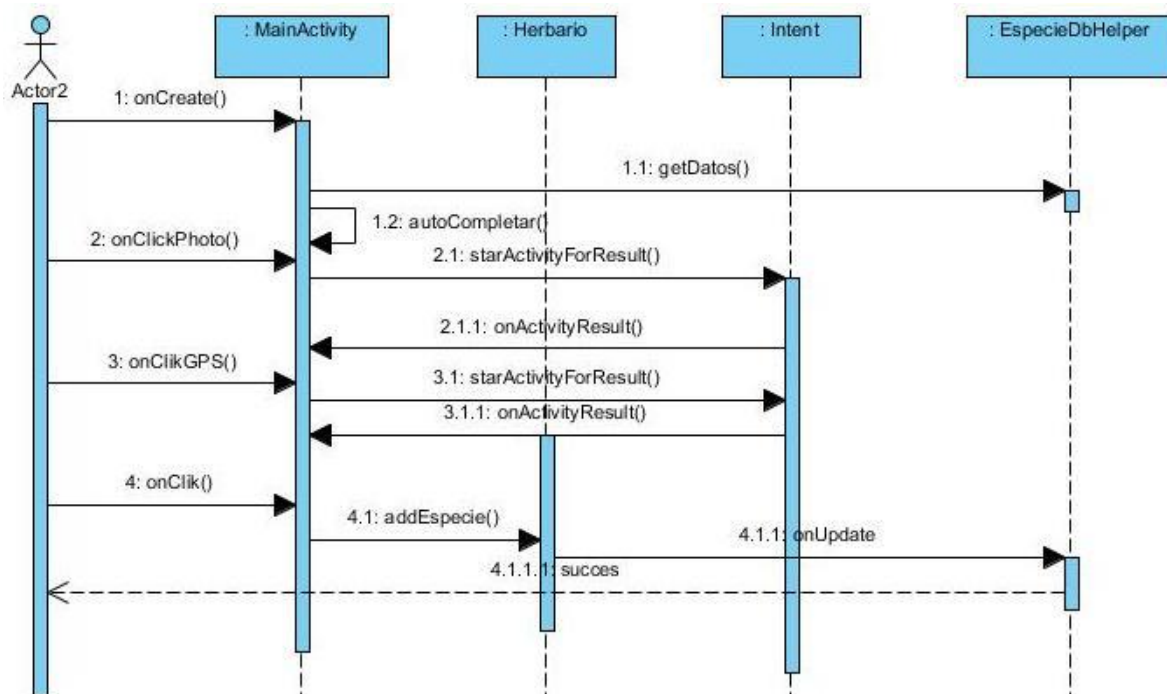


Figura 2.3: Diagrama de Secuencia del Caso de Uso Gestionar Especie.

Modelo Estructural

El diagrama de clases muestra todas las clases del *software* y las relaciones existentes entre ellas. Este diagrama (Figura 2.4) se realizó luego de los diagramas de secuencia para definir más fácilmente, a partir de las actividades modeladas previamente, los objetos y funciones a codificar.

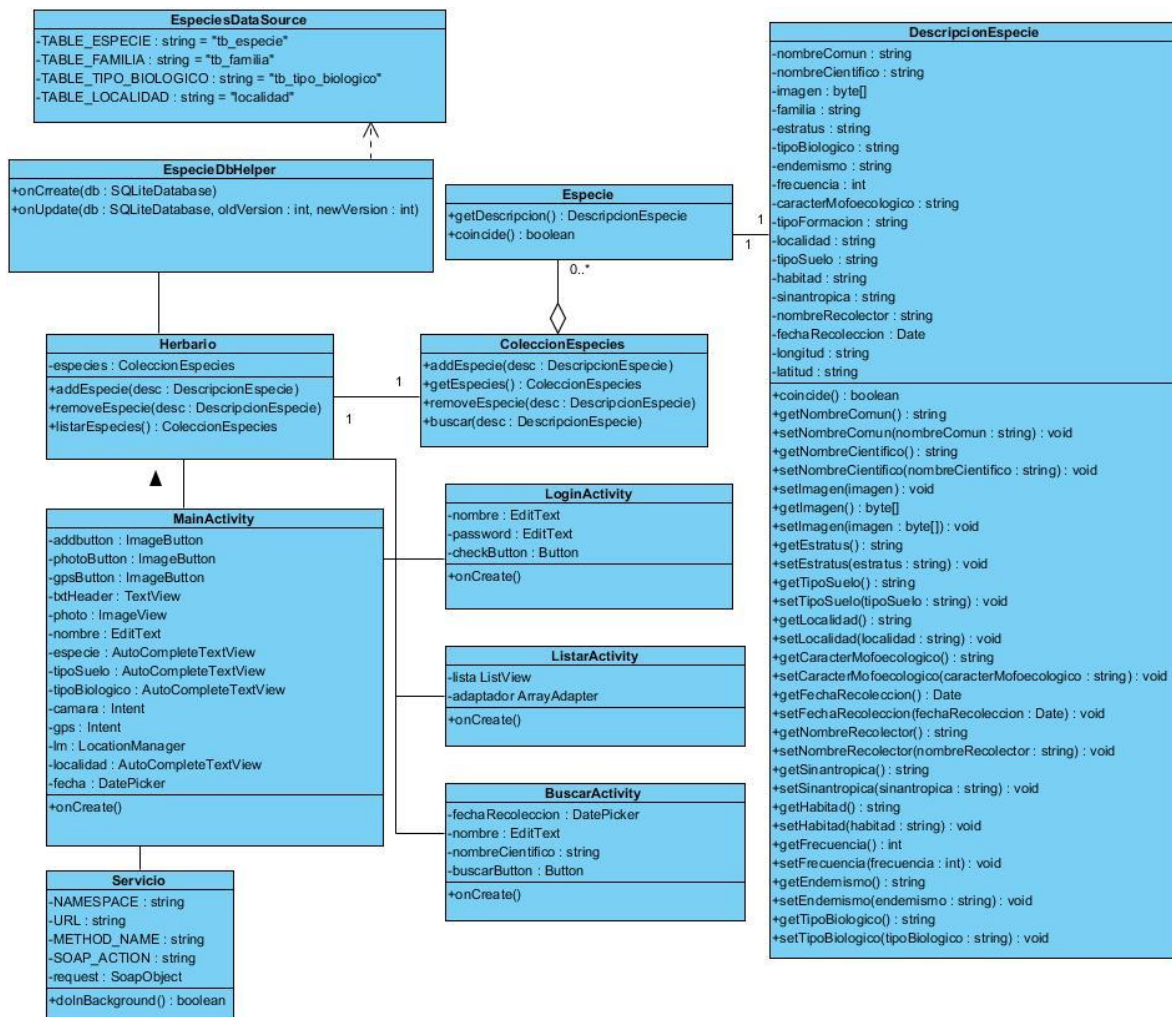


Figura 2.4: Diagrama de Clases.

Modelo de Implementación

El diagrama de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

Representa una correspondencia entre la arquitectura del *software* y la arquitectura del *hardware* [25].

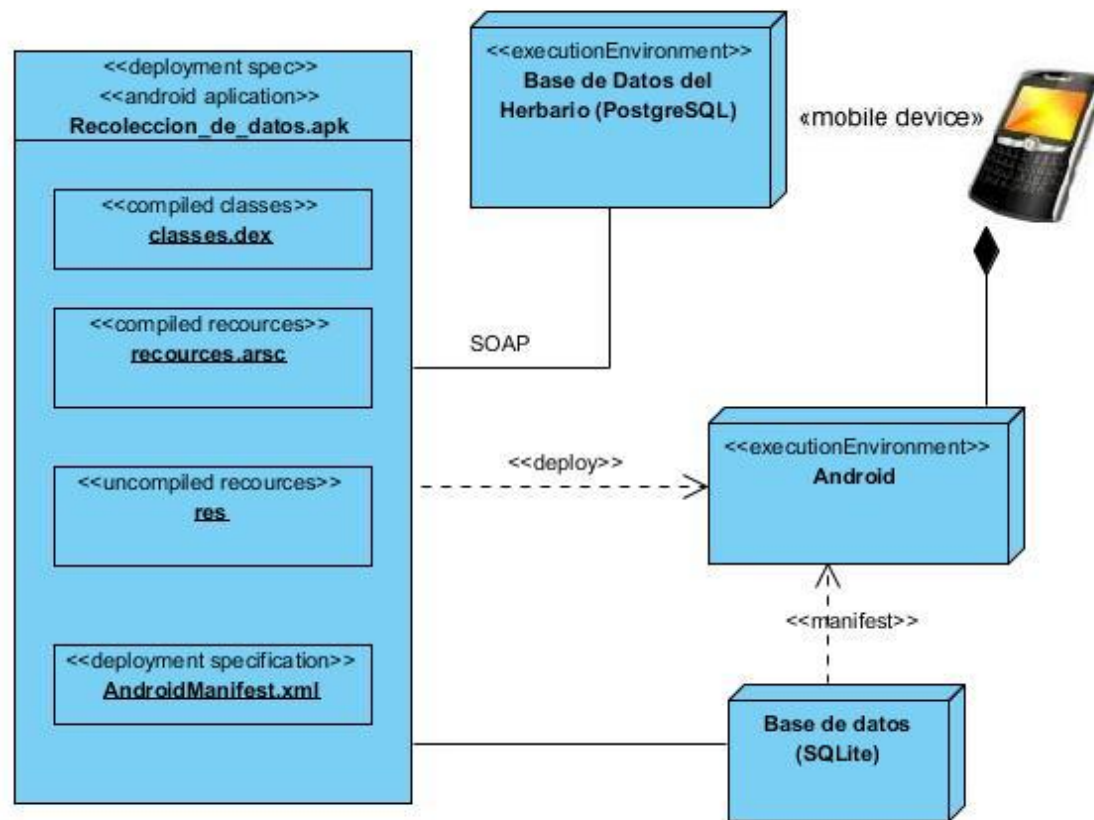


Figura 2.5: Diagrama de Despliegue.

La Figura 2.5 muestra el diagrama de despliegue del sistema informático propuesto. Se desarrollo la aplicación Recolección de Datos donde debe estar desplegada en el dispositivo móvil con *Android*, igual que la BD en *SQLite*. La aplicación Recolección de Datos actualiza los datos de las especies que se encuentran en la BD del Herbario Digital consumiendo un servicio web mediante SOAP.

2.2.3 Definición de tiempos

En este paso se establecen los plazos para cada una de las actividades restantes, con el objetivo de terminar la aplicación a tiempo para su salida al mercado. Se

tiene en cuenta el diseño computacional del *software* realizado en la tarea anterior y, las características volátiles y dinámicas de los servicios móviles [23]. Además, al inicio del proyecto se realizó una planificación, la cual fue actualizada en esta etapa y representada en un diagrama de Gantt (Figura 2.6), gráfico de tiempo para la planificación de proyectos [27], donde se muestra el tiempo en semanas planificado por cada etapa.

	Semanas																
Tareas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Análisis																	
Diseño																	
Desarrollo																	
Pruebas																	
Entrega																	

Figura 2.6: Diagrama de Gantt.

2.2.4 Asignación de recursos

En el presente paso se asignan los recursos para realizar cada actividad y alcanzar los objetivos propuestos. Además, se deben seleccionar las herramientas para el desarrollo de la aplicación móvil [23].

En esta etapa se realizó un estudio de factibilidad, expuesto en el capítulo 3, donde se analizó la disponibilidad de recursos humanos, económicos y tecnológicos para el desarrollo de la aplicación propuesta. El estudio arrojó como resultado que el desarrollo de la aplicación es factible según los aspectos técnico, económico y operativo.

El desarrollador cuenta con los recursos tecnológicos necesarios para realizar todas las actividades del proceso de desarrollo de la aplicación. Además, el CEMA cuenta con los recursos tecnológicos para el desarrollo y explotación del *software* y no se incurren en gastos económicos para esto, ya que se utilizaron tecnologías y herramientas de *software* libre y por tanto, la aplicación propuesta se distribuirá de forma libre sin costo alguno.

2.3 Desarrollo

El objetivo de esta fase es implementar el diseño en un producto de *software*. En esta etapa se realizan las siguientes actividades: codificar, pruebas unitarias, documentar el código y codificar ayudas [23].

2.3.1 Codificación

En el paso Codificación se escribió en el lenguaje de programación seleccionado, cada una de las partes definidas en los diagramas realizados en la etapa de diseño.

2.3.2 Pruebas unitarias

En el paso Pruebas unitarias se verificó el funcionamiento de la aplicación. En primer lugar, se comprobó la correcta operación de cada elemento desarrollado (objeto, clase, actividad, documento, entre otros) en forma individual. Posteriormente, se puso en funcionamiento la aplicación con todos los elementos, comprobando la interrelación entre ellos. Se ejecutaron varias pruebas y la mayoría de las veces se obtuvieron los resultados esperados. Los problemas encontrados fueron resueltos.

2.3.3 Documentación del código

A medida que se codificó y se probó cada elemento de la aplicación, se redactó la documentación sobre lo desarrollado. Entre la documentación se encuentra la generada en cada etapa de la metodología con sus artefactos y pasos, expuesta en el presente documento; el manual de ayuda elaborado y además, se hicieron comentarios en el código explicando las funciones principales para facilitar el mantenimiento del *software*.

2.3.4 Codificación de ayudas

La interfaz cuenta con un diseño intuitivo y fácil de entender. Cuenta con una serie de mensajes de ayudas que indican lo que puede o no puede hacer un usuario. Además, la aplicación cuenta con un Manual de Usuario para facilitar su instalación y explotación. Esto favorece la usabilidad de la aplicación y permite un fácil manejo por los usuarios.

2.4 Pruebas de funcionamiento

El objetivo de esta fase es verificar el funcionamiento de la aplicación en diferentes escenarios y condiciones; para esto se realizan las siguientes tareas: emulación y simulación, dispositivos reales y análisis de las 6 M [23].

2.4.1 Emulación y simulación

Se realizaron pruebas simulando el escenario y emulando el dispositivo móvil, explorando todas las utilidades y funciones de la aplicación, introduciendo diferentes datos, inclusive erróneos, para medir la funcionalidad y el nivel de robustez del *software*. Los problemas encontrados se solucionaron obteniendo resultados satisfactorios de las pruebas.

2.4.2 Dispositivos reales

Se probó la aplicación en dispositivos móviles reales para medir el desempeño y el rendimiento. Se utilizaron varios modelos de teléfonos móviles y *tablets* con sistema operativo *Android*. Las fallas encontradas durante la ejecución fueron resueltas y se verificó que el *software* cumpliera con los requerimientos especificados y el cliente estuviera satisfecho.

2.4.3 Análisis de las 6 M

Para valorar el potencial de éxito del servicio, se empleó el método de evaluación de las 6 M (*Moment, Movility, Money, Me, Machine, Multiuser*) propuesto por MDAM [23] que consiste en que un grupo de expertos evalúen 6 atributos: Momento, Movilidad, Dinero, Yo, Máquina y Multiusuario. Para esto se consultaron 10 expertos: 4 usuarios de la aplicación, 3 expertos de desarrollo de *software* y 3 expertos en el campo del desarrollo móvil; para calificar la presencia de los seis atributos en la aplicación desarrollada, empleando valores de 1 a 5 siendo 5 el de mayor presencia. La Tabla 2.1 muestra un resumen de las evaluaciones más frecuentes en cada atributo y la justificación de los valores obtenidos.

Tabla 2.1: Evaluación de las 6 M de la Aplicación *Android*.

Atributo	Definición	Calificación	Justificación
Momento	Un servicio que cuente con este atributo debe estar disponible en cualquier instante de tiempo en que el usuario desee usar dicho servicio.	5	El Especialista puede acceder a la aplicación en cualquier momento.
Movilidad	Un servicio móvil debe ser “móvil” por naturaleza, la ubicación debe ser una parte integral del servicio.	4	El usuario se puede desplazar a cualquier lugar e insertar las especies. Solo para exportar la BD debe estar conectado por <i>WiFi</i> ¹⁶ o por cable.
Dinero	Como cualquier acción comercial, un servicio móvil tiene un fin lucrativo, ya sea para el operador, para el proveedor del servicio o para el usuario.	2	La aplicación es de distribución gratuita, por lo que no genera ingresos. El CEMA reduce los costos de equipos para realizar las operaciones.
Yo	Se refiere al nivel de personalización de un servicio.	4	El servicio presenta un grado de personalización para el CEMA porque usa el idioma Español; además, permite que el usuario seleccione: las opciones de la cámara como lugar de almacenamiento, calidad de la imagen; la localización GPS; etc.; y el tipo de conexión para enviar los datos.
Máquina	La tecnología (terminal o redes) siempre es el factor que posibilita o limita; el atributo máquina busca añadir potencia a los dispositivos de última generación que cada vez tienen mayores prestaciones a nivel de <i>hardware</i> y <i>software</i> .	4	El servicio puede ser soportado por dispositivos móviles de gama baja, media o alta, que posean <i>Android</i> . El usuario puede hacer un buen uso de la aplicación sin importar la resolución de la pantalla.
Multiusuario	Busca extenderse dentro de la comunidad, que el servicio sea interactivo y que pueda utilizarse por múltiples usuarios de manera simultánea.	1	La aplicación permite la comunicación en ambas direcciones pero no es interactiva ni permite múltiples usuarios porque se usa en cada dispositivo por separado.

¹⁶ *Wireless Fidelity*, sistema para envío de datos a través redes de cómputo inalámbricas usando ondas de radio.

2.5 Entrega

Terminada la depuración de la aplicación y atendidos todos los requerimientos de última hora del cliente se da por finalizada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual de usuario de la aplicación.

El objetivo de los manuales es el entrenamiento. Una aplicación móvil debe constar de un manual donde se indique el proceso de instalación, la atención a posibles fallas en el tiempo de ejecución y, las especificaciones técnicas mínimas de *hardware* y *software* que requiere el equipo para el funcionamiento adecuado del aplicativo desarrollado [23].

2.5.1 Distribución

Luego se define la distribución: canal de comercialización de la aplicación, con el propósito de adecuar la aplicación al medio de distribución. En este caso la aplicación se distribuirá de forma gratuita cumpliendo con los requerimientos, a todos los especialistas y clientes del CEMA.

Conclusiones del capítulo

En este capítulo se documentó cada etapa de desarrollo de la aplicación según la metodología MDAM. Se mostraron los requerimientos funcionales, no funcionales y de personalización.

El diagrama de casos de uso permitió mostrar las acciones principales de la aplicación y su relación con los actores. De cada caso de uso se realizó una descripción textual y un diagrama de secuencia para representar las acciones del usuario y la aplicación. El diagrama de despliegue permitió modelar los componentes necesarios para desplegar la aplicación, así como los elementos que la componen.

Se desarrollaron dos aplicaciones: Recolección de datos y Herbario, una por cada rol identificado. Se empleó el método de evaluación de las 6 M, donde se consultaron 10 expertos para evaluar los seis atributos: Momento, Movilidad,

Dinero, Yo, Máquina y Multiusuario; obteniendo resultados favorables según los requerimientos definidos.

Capítulo 3: Estudio de factibilidad de la Aplicación *Android* para la recolección de datos de especies florísticas

En este capítulo se realiza un estudio de factibilidad de la aplicación *Android* propuesta. La factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. Para esto se apoya en 3 aspectos básicos: técnico, económico y operativo.

El éxito de un proyecto está determinado por el grado de factibilidad que se presente en cada uno de los tres aspectos anteriores. Un estudio de factibilidad sirve para recopilar datos relevantes sobre el desarrollo de un proyecto y en base a ello tomar la mejor decisión, si procede su estudio, desarrollo o implementación [28].

3.1 Factibilidad Técnica

La Factibilidad Técnica permite determinar si un proyecto es factible técnicamente. Se refiere a los recursos necesarios como herramientas, conocimientos, habilidades, experiencia, etc., que son necesarios para efectuar las actividades o procesos que requiere el proyecto. Generalmente se refiere a elementos tangibles (medibles). El proyecto debe considerar si los recursos técnicos actuales son suficientes o deben complementarse [28].

Teniendo en cuenta esto, el producto informático propuesto mejora el proceso de gestión de información de especies florísticas en el Distrito Moense, facilitando la recolección de los datos a los Especialistas del CEMA. De esta forma se cumple con el objetivo trazado en la investigación y se resuelve el problema que le dio origen.

Para el desarrollo de la aplicación se emplearon tecnologías libres y de código abierto disponibles para su uso y conocidas por el desarrollador, e igualmente el mismo contó con la infraestructura de *hardware* necesaria para su desarrollo. Por tal razón, no se incurren en gastos de desarrollo. Además, este cuenta con los conocimientos y habilidades necesarios para la implementación de la aplicación.

Por otra parte, el CEMA cuenta con la infraestructura tecnológica (*software* y *hardware*) necesaria para la implantación de la aplicación, que cumple con los requerimientos mínimos de la misma, lo cual asegura que se pueda usar sin gastos de inversión en nuevos equipos.

Teniendo en cuenta lo anteriormente planteado se puede determinar que el sistema es factible desde el aspecto técnico.

3.2 Factibilidad Económica

La Factibilidad Económica permite determinar si un proyecto es factible financieramente. Se refiere a los recursos económicos y financieros necesarios para desarrollar o llevar a cabo las actividades o procesos. Los recursos básicos que deben considerarse son el costo del tiempo, el costo de la realización y el costo de adquirir nuevos recursos [28]. En este aspecto se responde la pregunta: ¿puede realizarse el proyecto a un coste asumible por la empresa de *software* y por el cliente [27]?

Para el desarrollo de la aplicación propuesta se estimaron los valores de costo y tiempo mediante el método Estimación del Esfuerzo Basada de Casos de Uso, que fue propuesto originalmente por Gustav Karner en 1993, desarrollado mientras trabajaba en Objectory AB bajo supervisión de Ivar Jacobson (creador de los casos de uso); y posteriormente ha sido refinado por varios autores [29]. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a cierto número de factores que este propone tener en cuenta, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores [30].

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso Únicos o sin ajustar (UUCP, por sus siglas en inglés). Este valor se calcula a partir del Factor de Peso de los Actores sin ajustar (UAW, por sus siglas en inglés) y el Factor de Peso de los Casos de Uso sin ajustar (UUCW). Estos valores se calculan mediante un análisis de la cantidad de actores y casos de uso presentes en el sistema y la complejidad de cada uno de ellos respectivamente [29], como muestra la Tabla 3.1.

Tabla 3.1: Casos de Uso sin ajustar y Actores.

Paquete	Nombre	Tipo	Complejidad	Costo Estimado
Gestión	Insertar Geolocalización	UseCase	***	6400,00
Gestión	Insertar Foto	UseCase	***	6400,00
Gestión	Gestionar Especie	UseCase	***	6400,00
Gestión	Especialista	Actor	*	0,00
Visualización	Buscar Especies	UseCase	***	6400,00
Visualización	Listar Especies	UseCase	***	6400,00
Visualización	Consumidor	Actor	*	0,00

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se calculan los Puntos de Casos de Uso ajustados (UCP, por sus siglas en inglés), teniendo en cuenta los UUCP, el Factor de Complejidad Técnica (TCF, por sus siglas en inglés) y el Factor de Complejidad Ambiental (ECF) [29]. El TCF se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema, que se muestran en la Tabla 3.3. El cálculo del TCF se muestra en la Tabla 3.3.

Tabla 3.2: Factores de Complejidad Técnica.

Métricas	Descripción	Peso	Valor	TCF
TCF01	Sistema Distribuido	2,00	5,00	10,00
TCF02	Objetivos de rendimiento o tiempo de respuesta	1,00	4,00	4,00
TCF03	Eficiencia del usuario final	1,00	2,00	2,00
TCF04	Procesamiento interno complejo	1,00	4,00	4,00
TCF05	El código debe ser reutilizable	1,00	2,00	2,00
TCF06	Facilidad de instalación	0,50	5,00	2,50
TCF07	Facilidad de uso	0,50	3,00	1,50
TCF08	Portabilidad	2,00	3,00	6,00
TCF09	Facilidad de cambio	1,00	3,00	3,00
TCF10	Concurrencia	1,00	2,00	2,00
TCF11	Incluye características especiales de seguridad	1,00	2,00	2,00
TCF12	Provee acceso directo a terceras partes	1,00	5,00	5,00
TCF13	Se requieren facilidades especiales de entrenamiento a usuarios	1,00	3,00	3,00

			Total:	47,00
--	--	--	---------------	--------------

Tabla 3.3: Factor de Complejidad Técnica.

Factor	Valor
Valor desajustado de TCF (UTV, por sus siglas en inglés)	47,00
Pesos de TCF (TWF, por sus siglas en inglés)	0,01
Constante TCF (TC, por sus siglas en inglés)	0,60
TCF = TC + (UTV * TWF)	1,07

Las habilidades y el entrenamiento de los desarrolladores involucrados tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del ECF y se muestran en la Tabla 3.4. La Tabla 3.5 muestra el cálculo del ECF.

Tabla 3.4: Factores de Complejidad Ambiental.

Métricas	Descripción	Pesos	Valor	TCF
ECF01	Familiaridad con el modelo de desarrollo utilizado	1,50	4,00	6,00
ECF02	Experiencia de aplicación	0,50	3,00	1,50
ECF03	Experiencia en orientación a objetos	1,00	4,00	4,00
ECF04	Capacidad del analista líder	0,50	4,00	2,00
ECF05	Motivación	1,00	3,00	3,00
ECF06	Estabilidad de los requerimientos	2,00	4,00	8,00
ECF07	Trabajadores a tiempo compartido	-1,00	0,00	0,00
ECF08	Dificultad del lenguaje de programación	-1,00	3,00	-3,00
			Total:	21,50

Tabla 3.5: Factor de Complejidad Ambiental.

Factor	Valor
Valor desajustado de ECF (UEV, por sus siglas en inglés)	21,50
Pesos de ECF (EWF, por sus siglas en inglés)	-0,03
Constante ECF (EC, por sus siglas en inglés)	1,40
ECF = EC + (UEV * EWF)	0,75

Para determinar la cantidad de horas por caso de uso se contabilizan cuántos factores de los que afectan al ECF están por debajo del valor medio (3), para los factores ECF01 a ECF06. Luego se contabilizan cuántos factores de los que

afectan al ECF están por encima del valor medio (3), para los factores ECF07 y ECF08. Si el total es 2 o menos, se utiliza el Factor de Conversión (CF, por sus siglas en inglés) 20 horas-hombre/UCP, es decir, un Punto de Caso de Uso toma 20 horas-hombre. Si el total es 3 o 4, se utiliza el Factor de Conversión 28 horas-hombre/UCP. Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto [29].

Luego de calcular todos los factores que se tienen en cuenta para la Estimación del Esfuerzo Basada en Casos de Uso, se obtuvieron los resultados que muestra la Tabla 3.6.

Tabla 3.6: Estimación del esfuerzo y costo.

Pasos	Valor
Total de Casos de Uso	5
Puntos de Casos de Uso sin ajustar (UUCP)	52,00
Complejidad Técnica (TCF)	1,07
Complejidad Ambiental (ECF)	0,75
Puntos de Casos de Uso (UUCP * TCF * ECF) = UCP	42,00
Horas Estimadas por UUCP (CF)	20,00 horas-hombre
Esfuerzo en horas (E = CF * UCP)	840,00 horas-hombre
Valor por Hora	40 pesos
Costo Total Estimado (E * Valor por Hora)	33600,00 pesos

Analizando los resultados mostrados en la tabla anterior se puede concluir que se emplearán 20 horas-hombre para realizar un caso de uso, obteniéndose un esfuerzo de 840 horas-hombre. Tomando como valor por hora 40 pesos se obtiene un costo total de 33600 pesos.

A pesar del costo y esfuerzo estimados para el desarrollo de la aplicación, el desarrollo de la misma es gratis. Además, teniendo en cuenta que las herramientas y tecnologías son libres, el CEMA no tiene que incurrir en gastos de desarrollo o implantación. Como resultado de este análisis, se concluye que la aplicación es factible económicamente.

3.3 Factibilidad Operativa

La Factibilidad Operativa se refiere a todos aquellos recursos donde interviene algún tipo de actividad, depende de los recursos humanos que participen durante la operación del proyecto. Durante esta etapa se identifican todas aquellas actividades que son necesarias para lograr el objetivo del proyecto y se evalúa y determina todo lo necesario para llevarlas a cabo [28].

Los Especialistas del CEMA tienen experiencia con el uso de otro sistema informático para gestionar la información de especies florísticas en el Distrito Moense, el Herbario Digital, por lo que se favorece el uso y aceptación de la aplicación ya que el proceso es el mismo y la interfaz muy similar. Además, la interfaz tiene un diseño sencillo y amigable para facilitar la interacción con el usuario y su facilidad de uso y tiene en cuenta los estándares para aplicaciones *Android*.

Por tanto, los Especialistas se encuentran preparados para la utilización de la aplicación *Android* y, teniendo en cuenta las mejoras al proceso de recolección de datos, no han reportado ningún rechazo al cambio. Teniendo en cuenta esto, se puede afirmar que la aplicación es factible desde el punto de vista operativo.

Conclusiones del capítulo

Se realizó un Estudio de Factibilidad que permitió concluir que el desarrollo de la aplicación propuesta es factible desde el punto de vista técnico, económico y operativo.

La Estimación Basada en Casos de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de un sistema informático, si se emplea el modelado de casos de uso. Esto permite determinar la factibilidad económica.

Conclusiones

La aplicación *Android* desarrollada facilita la recolección de datos de especies florísticas, favoreciendo el proceso de gestión de información de estas especies en el Distrito Moense y dando cumplimiento al objetivo de la investigación.

Se emplearon tecnologías y herramientas de *software* libre para el desarrollo de la aplicación propuesta, favoreciendo las funcionalidades y la distribución de la misma.

El uso de la Metodología para el Desarrollo de Aplicaciones Móviles favoreció el desarrollo de la aplicación propuesta, maximizando la calidad del proceso de ingeniería de *software*.

El Estudio de Factibilidad realizado permitió comprobar que el desarrollo de la aplicación propuesta es factible desde el punto de vista técnico, económico y operativo.

El método de las 6 M permitió evaluar la aplicación desarrollada, obteniéndose resultados favorables que aseguran la utilización exitosa de la misma.

Recomendaciones

Se recomienda:

- Proponer la distribución de la aplicación en otros centros similares que realicen el proceso de recolección de datos de especies.

Referencias Bibliográficas

1. Fonseca, R.T., *Herbario Digital Representativo del Distrito Florístico Moaense*, 2014, Instituto Superior Minero Metalúrgico De Moa DR. ANTONIO NUÑEZ JIMENEZ: Moa.
2. Anónimo. *Proceso de inventario de los recursos florísticos en Moa*. 21 de mayo del 2015]; Available from: <http://cema.ismm.edu.cu/>.
3. Catalunya, U.O.d. *Desarrollo de aplicaciones para dispositivos móviles. Diploma de Posgrado*. 1 de abril del 2015]; Available from: [http://estudios.uoc.edu/es/masters-posgrados-especializaciones/informatica-multimedia-telecomunicacion/Posgrado_de_Desarrollo_de_aplicaciones_móviles_Universitat_Oberta_de_Catalunya_\(UOC\).htm](http://estudios.uoc.edu/es/masters-posgrados-especializaciones/informatica-multimedia-telecomunicacion/Posgrado_de_Desarrollo_de_aplicaciones_móviles_Universitat_Oberta_de_Catalunya_(UOC).htm).
4. Systems, G.B. *Aplicaciones para Dispositivos Móviles*. 2015 1 de abril del 2015]; Available from: http://www.gbts.com.mx/servicios/Aplicaciones_para_Dispositivos_Móviles_GBS_Global_BestTech_Systems.htm.
5. Blanco, P., et al., *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone*. 2009.
6. Otero, A., *Java Básico* 2003.
7. Otero, A.J.B. *Características del lenguaje Java*. . 2004 [cited 3 de junio del 2015; Available from: <http://www.iec.csic.es/cryptonomicon/java/quesjava.html>.
8. !!! INVALID CITATION !!!
9. Hernández, D.d.I.C.R., R.P. Vázquez, and J.V. Labrada, *Bases de Datos Móviles*, in *TLATEMOANI, No 14, 2013 España* 2013.
10. Anónimo. *"Móviles"*. 2013 15 de abril del 2015]; Available from: <http://basesdedatosavanzadas.wikispaces.com/Moviles>
11. Anónimo. *"Modelos de BD"*. 2000 2 de mayo del 2015]; Available from: <http://modelosbd2012t1.wordpress.com/2012/03/15/base-de-datos-moviles-3/>.
12. Revelo, J. *Tutorial De Bases De Datos SQLite En Aplicaciones Android* 2014 17 de abril del 2015]; Available from: <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos.html>.
13. Tudela, J.A., *Desarrollo de Aplicaciones para Dispositivos Móviles sobre la plataforma Android de Google* 2009, Universidad Carlos III de Madrid Escuela Politécnica Superior Madrid.
14. Anónimo *Introducción a los Servicios Web. Invocación de servicios web SOAP*. 2014.
15. Costoya, F.P. and J.M.P. Sánchez. *Sistemas Operativos Distribuidos "Introducción a los Servicios Web(Web Services)"*. 2013.

16. Cauldwell, P., R. Chawla, and V. Chopra. *Servicios web XML*. 2002 1 de abril del 2015]; Available from: <http://dialnet.unirioja.es/servlet/libro?codigo=318884>.
17. Anónimo. *Tema 4. Servicios WEB2*. in SCS – *Sistemas Cliente/Servidor 4º informática*. octubre 2008. Universidad de Vigo, España.
18. Tedeschi, N. *Web Services, un ejemplo práctico*. 2014 22 de abril del 2015]; Available from: <https://msdn.microsoft.com/es-es/library/bb972248.aspx>
19. Anónimo. *Sistemas de Información. Tema 6: Métodos, lenguajes y paradigmas*. . 2009 2 de marzo del 2015]; Available from: http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT6_Metodos_Lenguajes_Paradigmas.pdf .
20. FOWLER, M. *La Nueva Metodología*. 2005 1 de marzo del 2015]; Available from: <http://martinfowler.com/articles/newMethodology.html>.
21. DE SAN MARTIN OLIVA, P. *Uso de Metodología ICONIX*. 2010 [cited 1 de marzo del 2015; Available from: <http://www.buenastareas.com/ensayos/Uso-De-Iconix/344458.html>
22. Anónimo. *Manifiesto Ágil*. 2010 2 de marzo del 2015]; Available from: <http://agilemanifesto.org/>.
23. Mantilla, M.C.G., L.L.C. Ariza, and B.M. Delgado, *Metodología para el desarrollo de aplicaciones móviles*, in *Tecnura Vol. 18 No. 40* 2014: Colombia.
24. Quintero, Y.C., *Sistema de Clasificación Automática de Noticias a publicar en el periódico ¡ahora! Digital*, 2010, Universidad de Holguín Oscar Lucero Moya: Holguín.
25. JACOBSON, I. and G. BOOCH, *El Proceso Unificado de Desarrollo del Software* 2000: Addison-Wesley.
26. ROSENBERG, D. and M. STEPHENS, *Use Case Driven Object Modeling with UML* 2007, USA: Theory and Practice.
27. Pressman, R.S., *Ingeniería del Software, un enfoque práctico*. 5ta Edición ed2001: Mc Graw Hill.
28. Morris. *Ejemplo de Estudios de Factibilidad*. 2013 20 de mayo del 2015]; Available from: http://www.ejemplode.com/9-negocios/ejemplo_de_estudios_de-_factibilidad.htm.
29. Maidana, O.M.R. and A.M.A.L. King *Estimación Basada en Casos de Uso. UCP – Use Case Points. Estimación de Proyectos de Software*. . 2010.
30. Gracia, L.M. *Estimación por Puntos de Caso de Uso (el famoso Karner)*. 2010 20 de mayo del 2015]; Available from: <https://unpocodejava.wordpress.com/2010/11/06/estimacion-por-puntos-de-caso-de-uso-el-famoso-karner.htm>

Anexo 1. Descripción de los casos de usos.Caso de Uso Insertar foto

El especialista deja presionado el icono de la cámara, automáticamente sale un menú contextual para seleccionar si capturar la foto haciendo uso de la cámara o tomar la foto de la galería. Luego de realizar cualquiera de ambas opciones la foto saldrá sustituyendo el icono inicial y lista para ser insertada.

Precondición: La cámara debe funcionar.

Caso de Uso Insertar geolocalización

El especialista presiona el botón ON que se encuentra al lado de de los campos Latitud y Longitud. Automáticamente la aplicación se conecta con dicho servicio del dispositivo, entonces se muestran las coordenadas en los campos correspondientes. Si no se conecta con el servicio se podrá entrar las coordenadas manualmente.

Precondición: El GPS del dispositivo móvil tiene que estar encendido.

Caso de Uso Insertar especie

El especialista abre la aplicación, luego sale un formulario con los campos a llenar, en caso que en los campos (Familia, Género, Estratus, Endemismo, Localidad) exista algún dato se autocompletará este campo. El especialista presiona el botón salvar datos y automáticamente se insertan en la base de datos.

Precondición: Que la aplicación este instalada en el dispositivo móvil.

Caso de Uso Exportar Base de Datos

El especialista presiona el botón menú y escoge la acción exportar. Entonces el sistema mediante servicio web descarga los datos en la base de datos del Herbario Digital.

Caso de Uso Listar especies

El especialista presiona el botón listar y automáticamente aparece una activity con todas las especies insertadas en la base de datos. También permite tener una vista previa de los datos (Foto, Nombre Común y Nombre Científico) insertada.

Caso de Uso Buscar especies

El especialista luego de listar las especies, el sistema le permitirá hacer búsqueda mas especifica. El especialista presiona el botón buscar y automáticamente se despliega un ActionBar en la parte superior de la aplicación donde podrá buscar la especie filtrando por Nombre Común.

Anexo 2. Fotos de la aplicación

Insertar:

