



**UNIVERSIDAD  
NACIONAL  
DE LOJA**

PFC-CIS



*Área de la Energía, las Industrias y los Recursos Naturales No Renovables*

---

**Carrera de Ingeniería en Sistemas**

# **Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinoza” de la Universidad Nacional de Loja.**

*TESIS PREVIA A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS*

***Autor:***

Freddy Alexander Chamorro Encalada

***Director:***

Ing. Ana Lucía Colala Troya, Mg. Sc.

Loja- Ecuador

2015

## **Certificado del director**

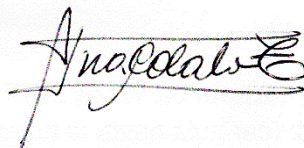
Ing. Ana Lucía Colala Troya, Mg. Sc.

**DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS**

CERTIFICA:

Haber dirigido, revisado y corregido en todas sus partes el desarrollo del Proyecto de Fin de Carrera de Ingeniería en Sistemas titulado: **“Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinosa” de la Universidad Nacional de Loja”**, con autoría del egresado Freddy Alexander Chamorro Encalada. En razón de que la misma reúne a satisfacción los requisitos de fondo y forma, exigidos para la investigación de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal designado para el efecto.

Loja, 10 de julio del 2015.



.....  
Ing. Ana Lucía Colala Troya, Mg. Sc.

**Director de Tesis**

## **Autoría**

Yo **FREDDY ALEXANDER CHAMORRO ENCALADA** declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

**Firma:** .....  


**Cédula:** 1105050304

**Fecha:** Loja, 22 de julio del 2015.

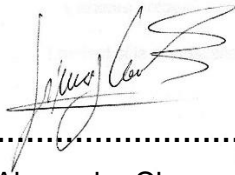
**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL  
AUTOR, PARA LA CONSULTA, REPRODUCCIÓN  
PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL  
TEXTO COMPLETO.**

Yo **FREDDY ALEXANDER CHAMORRO ENCALADA**, declaro ser autor de la tesis titulada: **GUÍA DE CAMPO MÓVIL PARA LA IDENTIFICACIÓN DE ESPECIES VEGETALES FORESTALES NATIVAS DEL ECUADOR TOMANDO COMO PILOTO EL JARDÍN BOTÁNICO “REINALDO ESPINOSA” DE LA UNIVERSIDAD NACIONAL DE LOJA**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veinte y dos días del mes de julio del dos mil quince.

Firma:.....

**Autor:** Freddy Alexander Chamorro Encalada.

**Cédula:** 1105050304

**Dirección:** Malacatos barrio Rumishitana.

**Correo electrónico:** fachamorroe@unl.edu.ec

**Teléfono:** 3106231 **Celular:** 0982825576

**DATOS COMPLEMENTARIOS**

**Director de Tesis:** Ing. Ana Lucía Colala Troya.

**Tribunal de Grado:** Ing. Walter Rodrigo Tene Ríos, Mg. Sc.

Ing. Carlos Miguel Jaramillo Castro, Mg. Sc.

Ing. Jorge Iván Tocto, Mg. Sc.

## **Agradecimiento**

Agradezco a Dios y a la Santísima Virgen María, por permitirme terminar esta etapa de estudios. A mi madre por su apoyo, consejos, comprensión, amor, y ayuda incondicional durante toda mi vida.

A mi esposa por su paciencia, comprensión, y apoyo para terminar esta etapa de estudios universitarios.

A mi directora de tesis la Ingeniera Ana Lucía Colala por tenerme paciencia y por guiarme en el desarrollo de este proyecto.

A todos mis maestros que me impartieron sus conocimientos y experiencias en el transcurso de mi vida estudiantil y que me ayudaron de una u otra forma para hacer posible la realización de mi profesión.

Para finalizar quiero agradecer a todos mis familiares, amigos y a todas las personas que me incentivaron y me motivaron para seguir adelante con los objetivos de este propósito.

## **Dedicatoria**

El presente trabajo de titulación lo dedico a Dios y a la Santísima Virgen María, por guiarme por el sendero correcto cada día, por iluminarme, protegerme y darme fuerzas para superar las adversidades a lo largo de toda mi vida.

A mi madre, que ha sido el pilar fundamental de mi vida, con su demostración de una madre ejemplar me ha enseñado a no desfallecer ni rendirme ante nada, y a perseverar a través de sus sabios consejos.

A mi esposa, hija, hermanos, tíos y demás familiares que han depositado su confianza en mí y me dan sus palabras de aliento cuando más las necesito.

A mis amigos y a todas las personas que me incentivaron y me motivaron para seguir adelante con los objetivos de este propósito.

Para finalizar quiero dedicar este trabajo a todos mis maestros que me guiaron y compartieron sus conocimientos y experiencias durante toda mi vida estudiantil y que contribuyeron a mi desarrollo personal, intelectual y espiritual.

## **A. Título**

Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinosa” de la Universidad Nacional de Loja.

## **B. Resumen**

En el presente Proyecto de Fin de Carrera (PFC), se brindó una solución a los problemas con los que cuentan los estudiantes de la carrera de Ingeniería Forestal del Área Agropecuaria y de los Recursos Naturales Renovables (AARNR) de la Universidad Nacional de Loja (UNL), referentes a la identificación de especies vegetales forestales nativas, para lo cual se realizó el análisis e implementación de una guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinosa” de la UNL, con la finalidad de brindar una solución eficaz y eficiente a la comunidad universitaria involucrada y a la sociedad en general, acorde a las tendencias tecnológicas actuales aprovechando las ventajas que proporciona la tecnología móvil, como portabilidad, facilidad de acceso a la información entre otras.

En el presente PFC se utilizó la metodología de desarrollo ágil SCRUM adaptada a aplicaciones móviles, la cual permitió implementar y documentar el proyecto de manera ordenada y entendible. Debido a la variedad de sistemas operativos para dispositivos móviles existentes en la actualidad se consideró la necesidad de realizar la aplicación para las plataformas que predominan en la comunidad universitaria involucrada, las cuales son Android e iOS.

Para el desarrollo de la guía de campo se utilizó el framework Apache Cordova que permitió la reutilización del código al momento de implementar la aplicación en varias plataformas, reduciendo así el tiempo, dinero y espacio.

El uso de las técnicas y herramientas expuestas en el presente PFC, han permitido la construcción exitosa de la aplicación móvil, y realizar diferentes pruebas con usuarios finales y herramientas tecnológicas determinando el buen funcionamiento de la misma. La guía de campo móvil ha sido publicada en las diferentes tiendas oficiales de aplicaciones y puede ser buscada con el nombre de EcuadorNatural, está disponible de forma gratuita para la comunidad universitaria y sociedad en general.



## **Summary**

In this final year thesis project a solution is provided to the problems that students face from the forestry degree program within the Agricultural and Natural Renewable Resources Area of the National University of Loja, concerning the identification of forest plant species native to Ecuador, for which the analysis and implementation of a mobile field guide for the identification of native forest plant species of Ecuador. The Botanical Garden Reinaldo Espinosa of the UNL was used as a test, in order to provide an effective and efficient solution for the involved university community and society in general, according to current technological trends and taking advantage of mobile technology provides such as portability and ease of access to information amongst others.

In this end of year thesis project SCRUM agile development methodology was used, which was adapted to the development of mobile applications, which allowed implementation and documentation of the project in an orderly and understandable manner. Due to the variety of operating systems for mobile devices existing today the need to deploy the application for platforms that dominate the university community involved, namely Android and iOS was also considered.

For the development of the mobile application Apache Cordova framework was used which allowed the reuse of code when implementing the application on various platforms, thus reducing time, money and space.

Using the tools and techniques outlined in this end of year thesis project, a successful construction of the mobile application was possible, as well as carrying out different tests with a variety of end users and technological tools in order to verify it functioned correctly. The mobile application has been published in the various official mobile application stores and can be found under the name EcuadorNatural. It is available free of charge to the university community and society in general.

# Índice de Contenidos

## Índice General

C. Introducción .....	1
D. Revisión de literatura .....	2
1 . Aplicaciones móviles para el reconocimiento de especies vegetales .....	2
1.1 Leafsnap.....	3
1.2 Identificación de especies vegetales utilizando dispositivos móviles.....	3
2. Tecnologías y herramientas utilizadas en el desarrollo de la aplicación móvil .....	4
2.1 Apache Cordova (Phonegap).....	4
2.2 HTML5.....	5
2.3 Javascript .....	5
2.4 CSS .....	6
2.5 JQuery Mobile.....	6
2.6 Servidor Web .....	7
2.6.1 Framework de desarrollo .....	7
2.6.2 Servidores web con framework tornado .....	7
2.7 Base de datos SQL.....	8
2.7.1 Postgresql.....	8
2.8 OPEN CV .....	8
2.9 Eclipse y Android SDK.....	9
2.10 Xcode.....	9
3. Especies vegetales forestales.....	9
3.1 Definición de especies vegetales forestales.....	9
3.2 Especies vegetales forestales del Ecuador. ....	9
3.3 Reconocimiento de especies vegetales .....	11
3.4 Segmentación de la imagen .....	12
3.5 Detección de bordes.....	12
3.6 Descriptores elípticos de Fourier.....	12
4. Jardín botánico “Reinaldo Espinosa” .....	13
E. Materiales y Métodos .....	15
1. Métodos de investigación .....	15
1.1 Estudio de casos .....	15
1.1 Método inductivo.....	15

2. Técnicas de recolección de información .....	15
2.1 Encuesta.....	15
2.2 Observación directa .....	15
2.3 Investigación Bibliográfica.....	16
3. Metodología para el desarrollo de aplicaciones móviles. ....	16
3.1 Fase 1: Determinación de la situación y formulación problemática .....	16
3.2 Fase 2: Fase Exploratoria .....	16
3.3 Fase 3: Fase Experimental .....	16
3.4 Fase 4: Pruebas de validación y funcionamiento .....	18
F. Resultados .....	19
1. Fase 1: Determinación de la situación y formulación problemática .....	19
2. Fase 2: Fase Exploratoria.....	19
2.1 Análisis de Frameworks para el desarrollo de aplicaciones móviles Multiplataforma .....	19
2.2 Análisis de Frameworks y librerías para el reconocimiento de imágenes.....	22
2.3 Análisis de proveedores para el alojamiento de servicios web en la nube .....	22
2.4 Análisis de Frameworks para la implementación de un servidor web.....	24
2.5 Análisis de herramientas para la implementación de bases de datos .....	25
2.6 Selección de la Metodología para el desarrollo de la aplicación móvil. ....	26
3. Fase 3: Fase Experimental .....	27
3.1 Iteración 0: Análisis y diseño de la aplicación .....	28
3.1.1 Especificación de requerimientos.....	28
3.1.1.1 Introducción .....	28
3.1.1.2 Propósito. ....	28
3.1.1.3 Ámbito del Sistema. ....	28
3.1.1.4 Personal Involucrado .....	28
3.1.1.5 Definiciones y Abreviaturas.....	29
3.1.1.6 Descripción General del Producto.....	30
3.1.1.7 Requisitos Específicos .....	31
a. Requisitos comunes de las interfaces.....	31
b. Requerimientos Funcionales.....	32
c. Requerimientos No Funcionales .....	38
3.1.2 Diagrama de Casos de uso.....	40
3.1.3 Descripción de Casos de Uso.....	41

3.1.3.1 CU001: Iniciar Sesión .....	42
3.1.3.2 CU002: Registro Usuario .....	43
3.1.3.3 CU003: Buscar Especie Vegetal .....	44
3.1.3.4 CU004: Identificar Especie Vegetal.....	45
3.1.3.5 CU005: Registrar Especie Vegetal.....	46
3.1.3.6 CU006: Verificar usuario en el servidor web .....	48
3.1.3.7 CU007: Registrar usuarios en el servidor web .....	49
3.1.3.8 CU008: Consultar especie vegetal en el servidor web .....	49
3.1.3.9 CU009: Identificar especie vegetal en el servidor web .....	50
3.1.3.10 CU0010: Registrar especie vegetal en el servidor web .....	50
3.1.4 Arquitectura de la aplicación .....	51
3.1.5 Interfaz gráfica de usuario.....	52
3.1.5.1 Iniciar Sesión .....	52
3.1.5.2 Registrar Usuario .....	53
3.1.5.3 Buscar Especie Vegetal .....	53
3.1.5.4 Identificar Especies Vegetales .....	54
3.1.5.5 Registro de Especies Vegetales .....	54
3.1.6 Modelo de datos .....	55
3.1.7 Análisis del servidor web.....	55
3.2 Planificación del proyecto .....	56
3.3 Iteración 1: Autenticación en la aplicación móvil .....	57
3.3.1 Creación de un proyecto con Apache Cordova .....	57
3.3.2 Implementar el servidor local y los métodos para la Autenticación en la aplicación móvil.....	59
3.4 Iteración 2: Registrar especies vegetales por medio de una hoja.....	62
3.5 Iteración 3: Identificación de especies vegetales por medio de la hoja .....	63
3.6 Iteración 4: Consulta de todas las especies vegetales. ....	63
3.7 Iteración 5: Implementación de un servidor web online para respuesta a peticiones de aplicación móvil. ....	64
3.8 Iteración 6: compilar la aplicación para el sistema operativo Android e iOS. ....	67
3.8.1 Compilar la aplicación para iOS .....	67
3.9 Iteración 7: Publicar la aplicación en las tiendas de Android, e iOS. ....	68
3.9.1 Publicar la aplicación en Google Play .....	69
3.9.1.1. Exportar y firmar la aplicación. ....	69

3.9.1.2 Acceder a la consola de Google play para desarrolladores. ....	70
3.9.1.3 Publicar la aplicación .....	72
3.9.2 Publicar la aplicación en App Store.....	74
4. Fase 4: Pruebas de validación y funcionamiento .....	76
4.1 Pruebas de carga .....	76
4.1.1 Autenticación en la aplicación móvil.....	77
4.1.2 Registrar especies vegetales por medio de una hoja .....	79
4.1.3 Identificar especies vegetales por medio de una hoja .....	81
4.1.4 Buscar todas las especies vegetales .....	83
4.2 Pruebas funcionales .....	85
4.2.1 Caso de Prueba.....	85
4.2.2 Ejecución de pruebas .....	85
4.3 Pruebas de rendimiento.....	93
4.4 Pruebas de usabilidad .....	94
G. Discusión .....	95
1. Desarrollo de la propuesta alternativa.....	95
2. Valoración técnica económica ambiental .....	99
H. Conclusiones .....	102
I. Recomendaciones .....	103
J. Bibliografía.....	104
K. Anexos .....	107
Anexo 1: Implementación de las funciones correspondientes a la iteración 1. ....	107
Anexo 2: Implementación de las funciones correspondientes a la iteración 2. ....	111
Anexo 3: Implementación de las funciones correspondientes a la iteración 3. ....	120
Anexo 4: Implementación de las funciones correspondientes a la iteración 4. ....	124
Anexo 5: Pruebas funcionales de la Aplicación Móvil .....	130
Anexo 6: Encuesta.....	143
Anexo 7: Resultados de la Encuesta .....	145

## Índice de Tablas

<b>TABLA I. PRINCIPALES ESPECIES FORESTALES DEL ECUADOR [14]</b> .....	9
<b>TABLA II. COMPARATIVA ENTRE FRAMEWORKS APACHE CORDOVA Y TITANIUM APPCELERATOR</b> .....	20
<b>TABLA III. COMPARATIVA FRAMEWORKS FRONT-END JQUERY MOBILE Y SENCHACH TOUCH</b> .....	21
<b>TABLA IV. LIBRERÍAS PARA EL PROCESAMIENTO DE IMÁGENES</b> .....	22
<b>TABLA V. LISTADO DE PROVEEDORES CON ALOJAMIENTO GRATUITO</b> .....	22
<b>TABLA VI. COMPARATIVA DE FRAMEWORKS WEB ESCRITOS EN PYTHON.</b> ...	24
<b>TABLA VII. COMPARATIVA DE HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE BASE DE DATOS</b> .....	25
<b>TABLA VIII. COMPARATIVA ENTRE METODOLOGÍAS AGILES Y TRADICIONALES</b> .....	26
<b>TABLA IX. COMPARATIVA DE LAS METODOLOGÍAS DE DESARROLLO ÁGIL SCRUM Y XP [28]</b> .....	27
<b>TABLA X. EQUIPO DE TRABAJO</b> .....	27
<b>TABLA XI. DATOS DEL ANALISTA/DESARROLLADOR DEL PROYECTO</b> .....	28
<b>TABLA XII. DATOS DEL SUPERVISOR DEL PROYECTO</b> .....	29
<b>TABLA XIII. TIPO DE USUARIO COLABORADOR</b> .....	30
<b>TABLA XIV. TIPO DE USUARIO CONSULTOR</b> .....	31
<b>TABLA XV. VERSIONES DE CADA SISTEMA OPERATIVO SOPORTADO POR LA APLICACIÓN MÓVIL</b> .....	31
<b>TABLA XVI. MÉTODOS A IMPLEMENTAR EN EL SERVIDOR WEB PARA EL RECONOCIMIENTO DE ESPECIES VEGETALES</b> .....	55
<b>TABLA XVII. ITERACIONES (SPRINTS) DEL PROYECTO</b> .....	56
<b>TABLA XVIII. MÉTODOS IMPLEMENTADOS (ITERACIÓN 2)</b> .....	62
<b>TABLA XIX. MÉTODOS IMPLEMENTADOS (ITERACIÓN 3)</b> .....	63
<b>TABLA XX. MÉTODOS IMPLEMENTADOS (ITERACIÓN 4)</b> .....	63
<b>TABLA XXI. RENDIMIENTO DE LA APLICACIÓN EN VARIOS DISPOSITIVOS MÓVILES</b> .....	93
<b>TABLA XXII. RECURSOS HUMANOS</b> .....	99
<b>TABLA XXIII. RECURSOS MATERIALES</b> .....	100
<b>TABLA XXIV. RECURSOS TÉCNICOS Y TECNOLÓGICOS</b> .....	100
<b>TABLA XXV. COSTO TOTAL DEL PROYECTO FIN DE CARRERA</b> .....	101

## Índice de Figuras

<b>Figura 1.</b> Apache Cordova, estructura de interacción con el dispositivo [5] .....	5
<b>Figura 2:</b> ThemeRoller [6] .....	6
<b>Figura 3:</b> Comunicación entre cliente-servidor con objetos JSON [8] .....	7
<b>Figura 4:</b> Ejemplos de reconstrucción con EFD, primera columna imagen original, segunda columna imagen segmentada, tercera columna contorno reconstruido con EFD [3] .....	13
<b>Figura 5:</b> Croquis del jardín botánico “Reinaldo Espinosa” [19] .....	14
<b>Figura 6:</b> Sendero de la flora nativa en el Jardín botánico “Reinaldo Espinosa” [20] .	14
<b>Figura 7:</b> Diagrama de casos de uso .....	41
<b>Figura 8:</b> Arquitectura de despliegue de la aplicación .....	52
<b>Figura 9:</b> Interfaz para ingresar a la aplicación .....	52
<b>Figura 10:</b> Interfaz de registro de usuario en la aplicación .....	53
<b>Figura 11:</b> Interfaz de búsqueda de especies vegetales .....	53
<b>Figura 12:</b> Interfaz de identificación de especies vegetales .....	54
<b>Figura 13:</b> Interfaz de Registro de Especies Vegetales.....	54
<b>Figura 14:</b> Diagrama ER del modelo de datos .....	55
<b>Figura 15.</b> Importar proyecto Android desde directorio Apache Cordova. ....	58
<b>Figura 16.</b> Directorio www, sitio de trabajo de la aplicación móvil. ....	59
<b>Figura 17:</b> Depurar la aplicación móvil en un dispositivo android .....	59
<b>Figura 18:</b> Directorio clonado desde repositorio Git (lugar de trabajo) .....	64
<b>Figura 19:</b> Cuenta creada en openshift con repositorio ejemplo creado desde línea de comandos.....	65
<b>Figura 20:</b> Repositorio clonado implementado en servidor local .....	65
<b>Figura 21:</b> Proceso de subida de actualizaciones del servidor web implementado ....	66
<b>Figura 22:</b> Listado de especies vegetales registradas en la aplicación móvil .....	67
<b>Figura 23:</b> Directorio de la aplicación móvil para la plataforma iOS .....	68
<b>Figura 24:</b> Depurar la aplicación iOS en un dispositivo real con Xcode.....	68
<b>Figura 25:</b> Exportar la aplicación android.....	69
<b>Figura 26:</b> Firmar la aplicación con clave.....	69
<b>Figura 27:</b> Datos para crear clave para android.....	70
<b>Figura 28:</b> Seleccionar la ruta para el archivo apk firmado.....	70
<b>Figura 29:</b> Registrar una cuenta de desarrollador en Google Play .....	71
<b>Figura 30:</b> Consola de Google Play .....	71

<b>Figura 31:</b> Añadir nueva aplicación Android.....	72
<b>Figura 32:</b> Entorno de aplicación móvil a subir.....	72
<b>Figura 33:</b> Subir APK de producción.....	73
<b>Figura 34:</b> Ficha de la aplicación Android en Google Play .....	73
<b>Figura 35:</b> Aplicación publicada en Google Play .....	73
<b>Figura 36:</b> Panel de iTunes. ....	74
<b>Figura 37:</b> Finalizar el proceso de validación. ....	74
<b>Figura 38:</b> Datos de aplicación a publicar en App Store .....	75
<b>Figura 39:</b> Aplicación iOS enviada para revisión.....	75
<b>Figura 40:</b> Resultados de prueba de carga de autenticación de usuarios (208 usuarios) .....	77
<b>Figura 41:</b> Resultados de prueba de carga de autenticación de usuarios (100 usuarios) .....	78
<b>Figura 42:</b> Resultados de prueba de carga de autenticación de usuarios (300 usuarios) .....	78
<b>Figura 43:</b> Resultados de prueba de carga de registro de especies vegetales (208 usuarios) .....	79
<b>Figura 44:</b> Resultados de prueba de carga de registro de especies vegetales (100 usuarios) .....	80
<b>Figura 45:</b> Resultados de prueba de carga de registro de especies vegetales (300 usuarios) .....	80
<b>Figura 46:</b> Resultados de prueba de carga de identificación de especies vegetales (208 usuarios) .....	81
<b>Figura 47:</b> Resultados de prueba de carga de identificación de especies vegetales (100 usuarios) .....	82
<b>Figura 48:</b> Resultados de prueba de carga de identificación de especies vegetales (300 usuarios) .....	82
<b>Figura 49:</b> Resultados de prueba de carga de la búsqueda de todas las especies vegetales (208 usuarios) .....	83
<b>Figura 50:</b> Resultados de prueba de carga de la búsqueda de todas las especies vegetales (100 usuarios) .....	84
<b>Figura 51:</b> Resultados de prueba de carga de la búsqueda de todas las especies vegetales (300 usuarios) .....	84



## **C. Introducción**

El presente Proyecto de Fin de Carrera (PFC), surge del análisis a la problemática encontrada en la comunidad universitaria de la carrera de Ingeniería Forestal del AARNR de la UNL, concerniente a la identificación de especies vegetales forestales nativas del Ecuador, donde se pudo evidenciar que los estudiantes de la carrera de Ingeniería Forestal, no cuentan con una guía de campo portable de especies vegetales forestales nativas, además tienen dificultad para determinar los beneficios de las mismas, falta de precisión al identificarlas y definir las, entre otras dificultades. Sobre la base del análisis realizado a los inconvenientes encontrados, se ha considerado pertinente implementar una aplicación móvil para la identificación de especies vegetales forestales nativas, para los sistemas operativos Android e iOS, brindando una solución a la comunidad universitaria involucrada.

Para el desarrollo de la aplicación móvil se procedió a establecer una solución factible a través un análisis comparativo entre distintas posibilidades de implementación, tomando en cuenta algoritmos, arquitectura de aplicación, modelo de datos y metodología de desarrollo a utilizar. Para continuar se diseñó la aplicación móvil multiplataforma para dispositivos móviles smartphone con mayor demanda en la carrera de Ingeniería Forestal del AARNR de la UNL, además se implementó un servidor web que es el encargado de proveer una respuesta a la aplicación móvil que realiza la petición de autenticación y registro de usuarios, la identificación y registro de especies vegetales y la consulta de las mismas, posteriormente se procedió a publicar la aplicación móvil en las tiendas oficiales de aplicaciones para las plataformas Android e iOS, y a ubicar el servidor web en la nube. Para finalizar se realizaron varios tipos de pruebas a la aplicación móvil y al servidor web con la finalidad de determinar el correcto funcionamiento y el grado de aceptación de los mismos.

En el presente documento consta todo el proceso realizado para dar cumplimiento al PFC, el mismo que empieza con una revisión de la literatura que permitió adquirir los conocimientos necesarios para el desarrollo de la aplicación móvil y del servidor web, seguidamente se encuentran los materiales y métodos empleados para la implementación de una solución adecuada al problema planteado, mismos que se encuentran plasmados en la sección de resultados y posteriormente en la discusión se describe los pasos realizados para dar cumplimiento a cada objetivo planteado. Finalmente se encuentran las conclusiones y recomendaciones emitidas luego de haber culminado el desarrollo del presente PFC.

## **D. Revisión de literatura**

### **1. Aplicaciones móviles para el reconocimiento de especies vegetales.**

Las aplicaciones móviles han adquirido gran acogida por parte de los usuarios, gracias al avance que han tenido los dispositivos móviles para las conexiones de red y demás servicios que ofertan, por cuanto surge la necesidad de implementar aplicaciones móviles las cuales ofrecen la ventaja de acceder a la información o servicios desde cualquier lugar y en cualquier momento, sin necesidad de contar con un computador [1]. Por otro lado, los seres humanos estamos rodeados de especies vegetales, a las que se les da diversos usos, ornamentales, producción de alimentos, medicinales, industriales entre otros. Pero lamentablemente la mayoría de las personas sabe muy poco acerca de las plantas que le rodean y que están a su disposición, en algunos casos ni siquiera se conoce el nombre de los árboles que se encuentra en algún parque por el que se camina todos los días.

Las guías suelen ser demasiado específicas orientadas a algún sector geográfico (en especial las impresas), por lo que es muy costoso utilizarlas para un usuario común sin mayores conocimientos en botánica. Sin embargo, el conocimiento acerca de las plantas está disponible en la web, basta con ingresar el nombre de una especie en cualquier navegador para encontrarse con descripciones, usos y fotografías de la planta en cuestión.

Aprovechando esta disponibilidad de información han surgido sendas iniciativas para generar una guía de campo en que la búsqueda se hace utilizando una imagen tomada con uno de estos dispositivos móviles smartphone, la que luego se envía a un servidor en el que se procesa la imagen para identificar las especies en cuestión.

Al no ser un escenario muy explotado pero si muy importante existen algunas aplicaciones móviles a nivel internacional, que permiten el acceso a esta información acorde a su ubicación geográfica, entre ellas se puede destacar:

## **1.1 Leafsnap**

Leafsnap es la primera aplicación para móviles que aborda el problema de identificación de plantas utilizando sólo una fotografía tomada desde un dispositivo móvil. Su versión actual incluye especies presentes en el noreste de Norteamérica y está disponible para el sistema operativo iOS.

La aplicación primero clasifica la imagen como válida, para ello detecta si lo que se le envía corresponde a una hoja sobre un fondo sólido de color claro, lo que en caso de ser negativo se le indica al usuario tal falla y además se dan las instrucciones para tomar correctamente la fotografía; por el contrario si la imagen corresponde a una hoja se prosigue a segmentar la foto separando el fondo de la hoja, luego se extrae la característica de la forma de la hoja, se contrasta con la base de datos y por un criterio de cercanía por distancia euclidiana se determinan las posibles respuestas, y la identificación final queda a cargo del usuario [2].

## **1.2 Identificación de especies vegetales utilizando dispositivos móviles**

La finalidad del proyecto para la identificación de especies vegetales utilizando dispositivos móviles de la Universidad de Chile es desarrollar una aplicación móvil para Android que sirva como guía para la identificación de especies vegetales de una base de datos de 15 especies vegetales, permitiendo al usuario la identificación de especies vegetales a través de una muestra fotográfica tomada por el mismo.

Además permite generar una base de datos con especies locales que eventualmente podría crecer para abarcar especies de todo Chile, por lo que el trabajo se centra en desarrollar un sistema similar a leafsnap pero enfocándose en el sistema operativo Android, con una aplicación nativa [3].

Estos casos de éxito fueron de gran utilidad en el desarrollo del presente trabajo, puesto que permitieron determinar la importancia que tienen las aplicaciones móviles en nuevos escenarios de investigación aun no explotados, para así implementar una aplicación móvil para identificación de especies vegetales en el Ecuador. La idea es brindar una solución de acceso gratuito a usuarios del sistema operativo Android, e iOS cubriendo el área geográfica de nuestra localidad, y establecer un trabajo de acceso público, con un código abierto para futuros trabajos.

## **2. Tecnologías y herramientas utilizadas en el desarrollo de la aplicación móvil.**

En su mayoría la comunidad universitaria del AARNR de la UNL que se orienta por el estudio de especies vegetales, posee un smartphone que utiliza el Sistema Operativo Android o iOS, para los cuales no existe una aplicación disponible en las tiendas oficiales, que facilite la identificación de especies vegetales forestales nativas del Ecuador. Razón por la cual se ha estimado conveniente desarrollar una aplicación móvil multiplataforma Android e iOS, en el presente escenario de estudio.

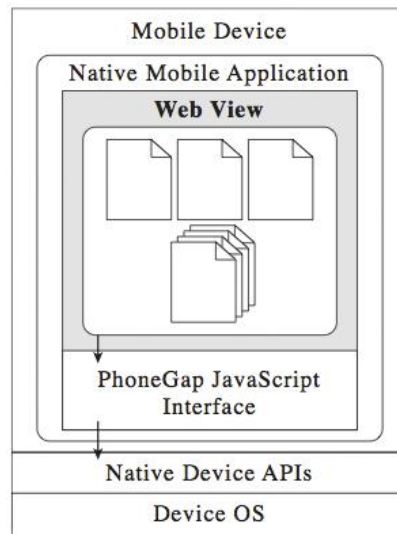
Para la implementación de la aplicación se ha utilizado tecnologías como: Apache Cordova, jQuery Mobile, Servidor Web, entre otros. También se ha utilizado herramientas como: Eclipse y Xcode.

A continuación se explicará detalladamente cada una de las tecnologías y herramientas utilizadas.

### **2.1 Apache Córdoba (Phonegap)**

Apache Cordova ha adquirido gran prestigio puesto que permite construir aplicaciones móviles multiplataforma, esto implica un menor tiempo de desarrollo y por ende disminuyen los costes de implementación de las aplicaciones [4].

Actualmente no existe ninguna diferencia entre Phonegap y Apache Cordova, a excepción del nombre y desde dónde se descargan. Ambos son gratuitos, Open Source y se utilizan del mismo modo. Phonegap no es más que una distribución de Apache Cordova. Cuando un desarrollador implementa una función en una aplicación que utiliza uno de los API de Apache Cordova, la aplicación llama a la API mediante JavaScript, y luego, una capa especial dentro de la aplicación traduce la llamada al API de Apache Cordova en un API nativo apropiado para la función en particular. Como un ejemplo, la forma en que se accede a la cámara de un iPhone es diferente a la de Android, por lo que esta capa común del API permite al desarrollador implementar una única interfaz que se traduce dentro de la aplicación en un API nativo apropiado para cada plataforma móvil compatible [5].



**Figura 1:** Apache Cordova, estructura de interacción con el dispositivo [5].

Apache Cordova trabaja con tecnologías como HTML5, JavaScript y CSS. A continuación se explicará el rol que cumple cada una de ellas dentro de un proyecto y algunos aspectos que se deben considerar para su uso.

## 2.2 HTML5

Esta tecnología es utilizada para el diseño de la interfaz de la aplicación móvil la cual permite que el diseño sea escalable y dinámico, con un lenguaje robusto en documentación.

## 2.3 Javascript

JavaScript permite gestionar la lógica de la aplicación, por ejemplo si la aplicación requiere hacer uso de información de otras aplicaciones por medio de JavaScript se puede consumir los recursos de un Servicio Web, esto lo hace mediante peticiones Ajax al servidor donde se encuentran los datos que se requieren consumir.

En un ejemplo de peticiones Ajax sería el siguiente:

```
$.ajax({
  // En data puedes utilizar un objeto JSON, un array o un query string
  data: {"parametro1" : "valor1", "parametro2" : "valor2"},
  //Cambiar a type: POST si necesario
  type: "GET",
  // Formato de datos que se espera en la respuesta
  dataType: "json",
  // URL a la que se enviará la solicitud Ajax
  url: "script.php", })
```

```

:success (function( data, textStatus, jqXHR ) {
    if ( console && console.log ) {
        console.log ("La solicitud se ha completado correctamente."); } })
:error (function( jqXHR, textStatus, errorThrown ) {
    if ( console && console.log ) {
        console.log( "La solicitud a fallado: " + textStatus);
    }
});

```

## 2.4 CSS

CSS es un lenguaje que sirve para organizar el aspecto de los componentes en una página HTML. Las hojas de estilo CSS son de suma importancia dentro de la aplicación puesto que permiten modificar los estilos del diseño de la misma (colores, letra, transiciones, entre otros), proporcionando así una interfaz agradable para el usuario.

## 2.5 JQuery Mobile

JQuery Mobile es un Framework Front-End (interfaz, final frontal o frontal) que se puede integrar fácilmente a Apache Cordova, permitiendo crear aplicaciones móviles con características de una aplicación web empaquetadas en una aplicación nativa. Además permiten realizar aplicaciones responsivas que se adapten a la resolución de pantalla de los diferentes dispositivos móviles. Una característica principal de jQuery Mobile es que se puede personalizar los temas para la interfaz de la aplicación, para ello existe la herramienta en línea como ThemeRoller, en donde se diseña y visualiza en tiempo real cómo van a quedar cada uno de los componentes, en la Figura 2 se muestra un ejemplo.

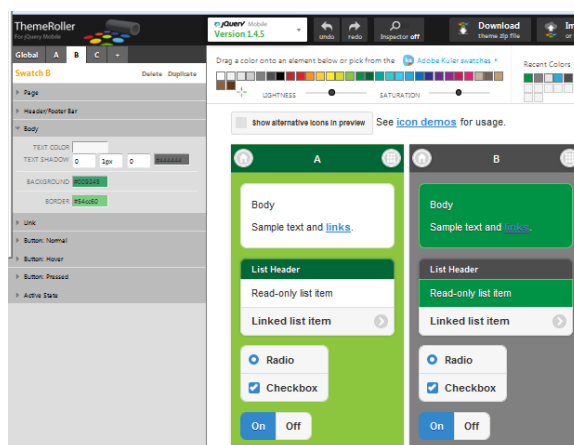


Figura 2: ThemeRoller [6].

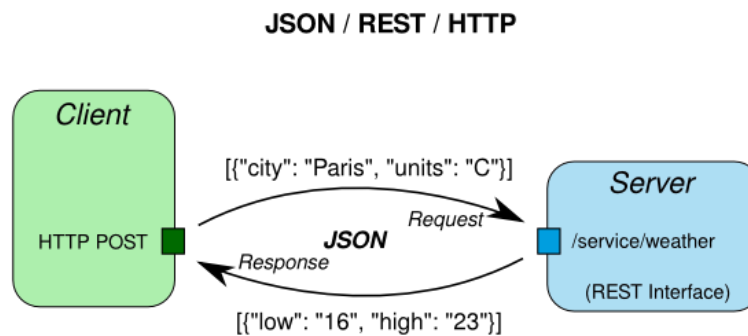
Luego de personalizar el tema se lo puede descargar y añadir los archivos CSS al proyecto de Apache Cordova.

## 2.6 Servidor Web

Los Servicios Web (WS) permiten el intercambio de datos entre aplicaciones haciendo uso de protocolos y estándares. En las aplicaciones móviles que necesitan comunicarse con otros sistemas es necesario hacer uso de un Servicio web, los cuales utilizan protocolos seguros para realizar el intercambio de datos por esta razón son muy utilizados en el desarrollo de aplicaciones [7].

Los datos se pueden intercambiar utilizando una estructura XML, JSON, entre otros, donde los mensajes JSON se pueden representar como Object y como Array.

Una petición o respuesta en formato JSON es un conjunto de pares nombre/valor. La figura 3 muestra un ejemplo de comunicación entre cliente – servidor con un objeto JSON.



**Figura 3:** Comunicación entre cliente-servidor con objetos JSON [8].

### 2.6.1 Framework de desarrollo

Un framework de desarrollo es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en el cual otro proyecto puede ser organizado y desarrollado [9].

### 2.6.2 Servidores web con framework tornado

Tornado está compuesto de un servidor web de código abierto y un conjunto de módulos escritos en Python.

Es un servidor web escalable no-bloqueante y extremadamente rápido. Al ser no-bloqueante, puede manejar miles de conexiones simultáneamente, lo que hacen que sea adecuado para aplicaciones web de tiempo real, objetivo por otra parte, para el que fue creado inicialmente [10].

## **2.7 Base de datos SQL**

El lenguaje de consulta estructurada o SQL, es un lenguaje de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra relacional permitiendo efectuar consultas con el fin de recuperar de una forma sencilla información de interés de la base de datos, así como también hacer cambios en ellas [9].

### **2.7.1 Postgresql**

Postgresql es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, vistas, entre otros [11].

## **2.8 OPEN CV**

OpenCV (Open Computer Vision) es una librería de software libre de visión por computador que incluye cuatro componentes diferentes para la visión por computador en tiempo real. Ha sido creada por un equipo de desarrolladores de todo el mundo. El primer componente es CxCore, el cual nos proporciona los tipos genéricos que serán usados por las demás librerías como pueden ser árboles, listas, colas, secuencias, imágenes y las funciones que operan sobre estos. El segundo componente es CvReference que incluye todas las funciones de análisis y procesamiento. El tercer componente es la librería CvAux que contiene componentes experimentales y obsoletos del proyecto OpenCV. El último componente es la librería HighGUI que nos proporciona los elementos necesarios para la captura, presentación y almacenamiento de imágenes. El componente HighGUI viene como una librería adicional. Esta librería nos proporciona una extensa lista de interfaces de captura. No provee herramientas para crear diferentes hilos ni flujos de datos, la adquisición, procesamiento y presentación de la imagen debe ser explícita.

OpenCV es una fuente fantástica de implementaciones estandarizadas de las funciones de visión por computador. Es posible que un quinto componente específico para sistemas de visión por computador sea añadido a la colección de OpenCV.

Esta librería está disponible para los lenguajes de programación C++ y Python y para los sistemas operativos Windows, Linux y Mac OS [12].



## 2.9 Eclipse y Android SDK

Para desarrollar aplicaciones móviles Android existen algunas herramientas, una de ellas es Eclipse misma que puede integrarse con Android SDK (Software Developer Kit). Actualmente Google ha lanzado su propia versión de Eclipse para Android la cual viene con el SDK y demás herramientas de desarrollo previamente instaladas. Esta herramienta se la puede descargar de forma gratuita desde la página oficial de desarrolladores Android, disponible según el sistema operativo en el cual se trabaje (Windows, Linux, Mac OS X).

## 2.10 Xcode

Xcode es un entorno de desarrollo de Apple el cual se distribuye junto con Mac OS X. Esta herramienta permite emular aplicaciones para sistema Operativo iOS, también permite generar el archivo binario de la aplicación para publicarla en el App Store.

## 3. Especies vegetales forestales

### 3.1 Definición de especies vegetales forestales

Una especie forestal es aquella que pasa de los 10cm., de diámetro a la altura del pecho, y tiene más de 5m., de altura. Es importante saber que en el Ecuador existen aproximadamente 3200 especies forestales, lamentablemente los estudios de muchas de estas especies han culminado en su descripción [13].

### 3.2 Especies vegetales forestales del Ecuador.

Las ventajas comparativas de nuestro País permite que en sus bosques crezcan alrededor de 5.000 especies de arbóreas, puesto que en algunas zonas disponen de 12 horas de luz al día durante todo el año, lo que incide en una mayor velocidad de crecimiento de especies forestales valiosas, tanto nativas como exóticas [14]. En la tabla I, se enumeran las principales especies vegetales forestales del Ecuador, su nombre común, científico y su desarrollo fisiológico, como información básica de las mismas.

**TABLA I. PRINCIPALES ESPECIES FORESTALES DEL ECUADOR [14]**

<b>PRINCIPALES ESPECIES FORESTALES Y SU CICLO VEGETATIVO EN ECUADOR</b>		
<b>Nombre común</b>	<b>Nombre científico</b>	<b>Desarrollo fisiológico (años)</b>
Arabisco, garza	Jacaranda mimosifolia	18
Faique	Acacia macracantha	20
Guaylo	Delostoma integrifolium	25

Higuerón rojo	<i>Ficus cuatrecasana</i>	15
Amarillo	<i>Centrolobium ochroxylum</i>	25
Balsilla	<i>Heliocarpus americanus</i>	5
Igodeoso	<i>Ficus insipida</i>	8
Pumamaqui	<i>Oreopanax rosei</i>	25
Sangre de drago	<i>Croton pavonis</i>	25
Capulí	<i>Prunus capulí</i>	20
	<i>Prunus cerolina</i>	20
Fresno	<i>Fraxinus chinensis</i>	25
Cedro blanco	<i>Cedrela montana</i>	30
Changue	<i>Caesalpinia spinosa</i>	30
Cedro de monte	<i>Cedrela rosei</i>	30
Roble andino	<i>Roupala obovata</i>	20
Huilco	<i>Anadenanthera colubrina</i>	25
Cascarilla	<i>Cinchona macrocalyx</i>	25
Cascarilla hoja de luma	<i>Cinchona lucumaefolia</i>	25
Cascarilla roja	<i>Cinchona pubescens</i>	25
Cascarilla fina	<i>Cinchona officinalis</i>	25
Guararo	<i>Lafoensia acuminata</i>	25
Luma	<i>Pouteria lúcuma</i>	25
Caimito	<i>Chrysophyllum aurantum</i>	25
Copal	<i>Dacryodes peruviana</i>	25
Chaviande	<i>Virola dixonii</i>	30
Ebano	<i>Zizyphus thysiflora</i>	25
Eucalipto	<i>Eucalyptus globulus</i>	15
	<i>Eucalyptus saligna</i>	15
	<i>Eucalyptus robusta</i>	15
	<i>Eucalyptus grandis</i>	15
	<i>Eucalyptus deglupta</i>	15
Romerillo fino	<i>Prumnopitys harmsiana</i>	50
Fernan Sánchez	<i>Triplaris cumingiana</i>	25
Romerillo mollón	<i>Retrophyllum rospigliosii</i>	25

Guayacán	<i>Tabebuia chrysantha</i>	40
Fresno	<i>Tecoma stans</i>	25
Romerillo azuceno	<i>Podocarpus oleifolius</i>	100
Arrayán	<i>Myrcianthes hallii</i>	25
Arrayan aromático	<i>Myrcianthes fragrans</i>	20
Jigua	<i>Nectandra reticulata</i>	30
Leucaena	<i>Leucaena leucocephala</i>	15
Duco colorado	<i>Clusia latipes</i>	18
Laurel de monte	<i>Myrica pubescens</i>	18
Mambla-bombón	<i>Erythrina poeppingiana</i>	25
Moral bobo	<i>Claricia racemosa</i>	20
Moral fino	<i>Clorophora tinctoria</i>	25
Mascarey	<i>Hyeronima chocoensis</i>	25
Moltilón	<i>Hyeronima alchornoidea</i>	25
Macharé	<i>Symphonia globifera</i>	25
Nogal	<i>Juglans neotropica</i>	30
Pachaco	<i>Zchizolobium parahybum</i>	10
Palo de vaca	<i>Alseis eggersii</i>	25
Pino	<i>Pinus radiata</i>	20
Roble de Guayaquil	<i>Tabebuia pentaphyla</i>	20
Roble	<i>Terminalia superba</i>	20
Roble de Esmeraldas	<i>Terminalia amazónica</i>	20
Samán, caucho blanco	<i>Samanea saman</i>	20
Sande	<i>Brosimum utile</i>	20
Sangre de gallina	<i>Vismia vacifera</i>	25
Teca	<i>Tectona grandis</i>	25
Tamarindo	<i>Tamarindus indica</i>	25

### 3.3 Reconocimiento de especies vegetales

El reconocimiento de una especie vegetal se puede hacer tomando en cuenta una variedad de criterios. Basándose en un criterio de usabilidad se puede descartar utilizar las flores, los frutos y las semillas, debido a que suelen encontrarse por períodos muy limitados durante el año.

La hoja de una planta puede diferenciarse por la nervadura, la textura o la forma de la hoja. La nervadura y la textura de las hojas pueden tener variaciones pequeñísimas, apenas perceptibles, entre distintas especies, por lo que una mejor opción es utilizar la forma de la hoja. Otra característica que se descarta usar es el color, pues la hoja puede variar mucho su color dependiendo de la época del año o de las condiciones de crecimiento de la planta (disponibilidad de agua, nutrientes, luz, dióxido de carbono, etc.) [3].

### **3.4 Segmentación de la imagen**

En cualquier proceso de reconocimiento de imágenes, el paso fundamental es diferenciar claramente todos los elementos que componen la imagen. Esto es segmentar los diferentes objetos del fondo. Para ello se pueden aplicar multitud de procesos y técnicas distintas, dependiendo del tipo de imagen y el resultado deseado. Normalmente la segmentación de una imagen monocromática se basa en las características de las tonalidades de gris, tales como la discontinuidad y la similitud. La discontinuidad busca líneas, bordes o puntos en función a cambios abruptos en la tonalidad de gris, mientras que la similitud establece regiones basándose en las relaciones espaciales o cromáticas que puede haber entre los píxeles que la forman [15].

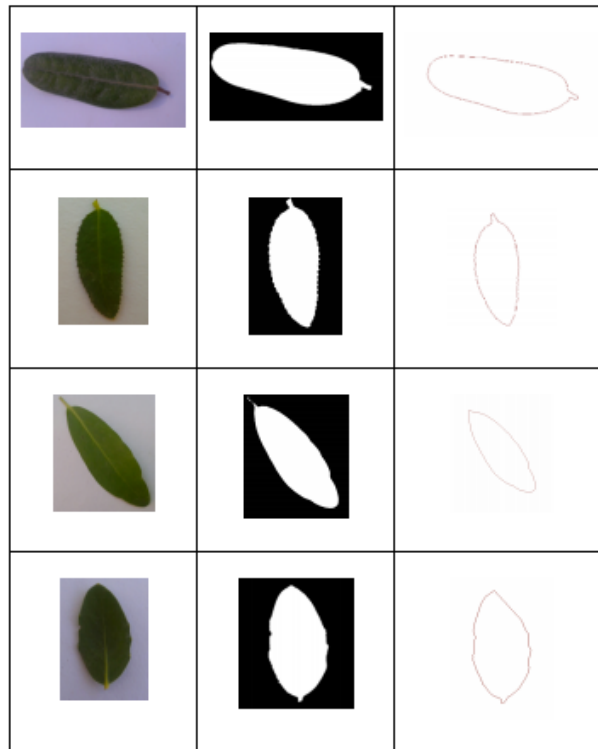
### **3.5 Detección de bordes**

Para obtener la forma de la imagen es necesario obtener los bordes o contorno de la imagen. Los bordes de una imagen se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Estos nos facilitan una valiosa información sobre las fronteras de los objetos que nos interesa identificar del resto de la imagen [16].

### **3.6 Descriptores elípticos de Fourier**

La descripción de la forma de la hoja se puede hacer con una función que describa el borde como una curva o una que describa el área de la hoja. Para describir la curva, uno de los algoritmos más populares es el de descriptores elípticos de Fourier (EFD). Para este proyecto se optará por los (EFD) en primera instancia por la facilidad de comprensión e implementación, además de que permiten una representación suficientemente buena para que la búsqueda entregase resultados coherentes. Por otra parte explorar e implementar otras funciones como los momentos de Zernik [17] (descripción del área) podría tomar demasiado tiempo. Los EFD tienen la ventaja de ser invariantes a la escala, rotación y posición de la curva, por lo que se ajusta muy bien a

las necesidades de la aplicación. Así sólo es importante tomar la foto completamente de frente a la hoja. En la Figura 4 se pueden ver ejemplos de la reconstrucción, a partir de los descriptores, del borde de algunas hojas.



**Figura 4:** Ejemplos de reconstrucción con EFD, primera columna imagen original, segunda columna imagen segmentada, tercera columna contorno reconstruido con EFD [3].

#### 4. Jardín botánico “Reinaldo Espinosa”

El Jardín Botánico, fundado en 1949 por el ilustre Botánico Reinaldo Espinosa, cuenta con un área aproximada de 7 ha., localizado a 5km de la ciudad de Loja en la vía a Vilcabamba; a 214 m.s.n.m., con una precipitación anual de 900 mm y temperatura media de 15,3<sup>0</sup>c., es el más antiguo del Ecuador y el único ubicado en el nudo de convergencia de las corrientes bioclimáticas cálidas húmedas de la Amazonia y cálidas secas de la vertiente del Pacífico, situación que da origen a una diversidad florística única de la hoya y provincia de Loja [18].

En sus predios se han inventariado 1385 especies vegetales entre nativas y exóticas y una considerable fauna asociada a este sistema. Se halla inscrito en la Organización Internacional para la Conservación de Jardines Botánicos (BGCI). La ubicación de las plantas en el jardín no obedece a una distribución sistemática ordenada, pero contribuye

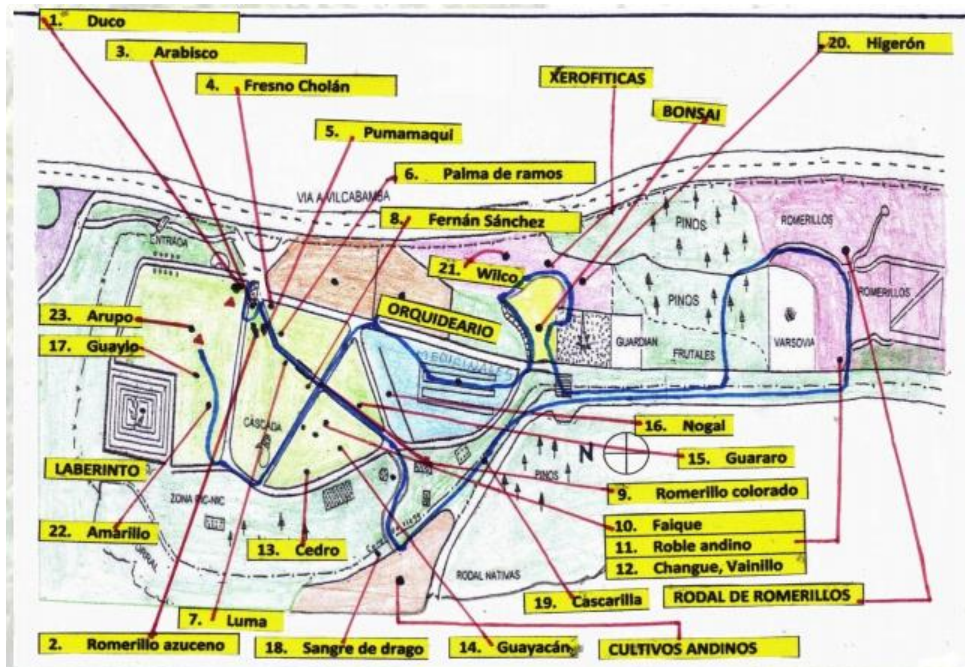
un laboratorio biológico inigualable en el país. Para el desarrollo del presente PFC se pretende tomar una muestra de las especies vegetales forestales nativas.

En la figura 5, se puede apreciar el jardín botánico “Reinaldo Espinosa”, clasificado por sectores según la ubicación de especies vegetales en el mismo.



**Figura 5:** Croquis del jardín botánico “Reinaldo Espinosa” [19].

En la figura 6, se puede apreciar los lugares en los que se encuentran las diferentes especies vegetales, sus senderos y acceso a ellas.



**Figura 6:** Sendero de la flora nativa en el Jardín botánico “Reinaldo Espinosa” [20].

## **E. Materiales y Métodos**

Para el desarrollo del presente trabajo de titulación se utilizó algunos métodos de investigación, los cuales permitieron realizar un análisis profundo de la situación actual en el campo de estudio y conjuntamente con el empleo de técnicas de recolección de información, se logró determinar los requerimientos funcionales y no funcionales necesarios para diseñar e implementar una solución factible al problema planteado.

Para el desarrollo de aplicaciones móviles se utilizó una metodología de desarrollo ágil, la cual permitió documentar la solución al problema de manera eficiente y cumpliendo los estándares requeridos en el desarrollo de aplicaciones móviles.

### **1. Métodos de investigación**

**1.1 Estudio de casos.-** Este método fue utilizado con la finalidad de realizar una exploración y análisis de casos de estudio que impliquen el reconocimiento de especies vegetales por medio de dispositivos móviles (ver sección Revisión de Literatura, subsección 1).

**1.2 Método inductivo.-** Este método se lo aplicó en el momento de la recopilación de la información que sirvió para esclarecer y profundizar algunos conceptos del caso de estudio, además permitió delimitar el escenario de estudio para resolver el problema de investigación.

### **2. Técnicas de recolección de información**

**2.1 Encuesta.-** Esta técnica fue utilizada para determinar la problemática existente sobre el campo de estudio, y establecer una solución para los sistemas operativos móviles con mayor afluencia en la carrera de Ingeniería Forestal del AARNR de la UNL. También se utilizó para evaluar el funcionamiento de la guía de campo móvil implementada, mediante la aplicación de una encuesta estructurada a una muestra de la comunidad universitaria involucrada de la UNL (ver Anexo 7).

**2.2 Observación directa.-** Esta técnica de investigación fue empleada para comprender la situación actual del problema abordado en el presente PFC, para lo cual se utilizó fichas de observación directa, y así poder elaborar un buen análisis, e implementar una solución a las necesidades de la comunidad universitaria involucrada.

**2.3 Investigación Bibliográfica.-** Haciendo uso de esta técnica se sustenta la base teórica de la investigación mediante consultas a: fuentes bibliográficas confiables, textos, revistas indexadas, artículos científicos, caso de éxito, apuntes, documentos varios, así como también fuentes informáticas fidedignas.

### **3. Metodología para el desarrollo de aplicaciones móviles.**

Para desarrollar aplicaciones móviles al igual que en un proyecto de software tradicional es necesario hacer uso de una metodología adecuada, la cual constituye una guía para planificar, organizar el proceso de desarrollo y documentar el mismo. En el PFC se ha empleado la metodología Scrum, la cual proporciona un conjunto de pasos a seguir en el desarrollo de una aplicación móvil de manera ágil.

A continuación se explicará más detalladamente como se desarrolló cada fase de dicha metodología.

#### **3.1 Fase 1: Determinación de la situación y formulación problemática**

En esta fase se determina la problemática del presente PFC, estableciendo como se genera el problema, logrando tener una visión general de la situación actual con un previo análisis y determinando ¿Qué pasa?, ¿A quién les pasa eso?, ¿Dónde ocurre ese fenómeno?, ¿Cuándo sucede?, con esto se puede determinar la magnitud del problema y con esto dar una solución óptima [21].

#### **3.2 Fase 2: Fase Exploratoria**

Esta fase se basó en una revisión de literatura a profundidad con la finalidad de determinar las herramientas necesarias para la elaboración de aplicaciones móviles con reconocimiento de objetos, y analizar las posibles alternativas que nos permitirán brindar solución al problema planteado.

Teniendo claro el problema a solucionar y las causas que lo originan, se selecciona en base a una lista de posibles soluciones y el estudio de las mismas, la mejor solución por funcionalidad, factibilidad y accesibilidad. Logrando determinar las herramientas viables para el desarrollo del proyecto [21].

#### **3.3 Fase 3: Fase Experimental**

##### **3.3.1 Participantes en el proyecto**

En esta fase se ha establecido el equipo que participó en el desarrollo del presente proyecto asignándoles a cada uno un rol de acuerdo a los especificados en la metodología Scrum los cuales son: Product Owner (Propietario del producto), Scrum Master (Facilitador), Scrum Team (Equipo) y Usuarios.



### **3.3.2 Iteración 0: Análisis y diseño de la aplicación.**

En este sprint (iteración) se realizaron algunas tareas que corresponden al análisis y diseño de la aplicación móvil, dentro de estas tareas se tiene: Análisis de requerimientos, modelado de casos de uso, modelo de datos, interfaz gráfica de usuario y la arquitectura de la aplicación.

### **3.3.3 Planificación del proyecto**

Luego de realizar la especificación de requerimientos, se procedió a elaborar la planificación del proyecto, para ello se tomó como base los requerimientos encontrados dividiéndolos en iteraciones, asignando un tiempo estimado para cada iteración. A continuación se describirán las iteraciones en las que fue dividido el proyecto:

### **3.3.4 Iteración 1: Autenticación en la aplicación móvil**

En esta iteración se procedió a implementar los requerimientos asignados en la **Iteración 0** del proyecto, para ello se consideró a la iteración como un subproyecto en el que se realizaron las siguientes fases:

#### **3.3.4.1 Análisis y diseño (Pre-juego)**

Tomando como base el análisis y diseño de la aplicación realizado en la **iteración 0**, en esta fase se realizó nuevamente un análisis en especial de la interfaz gráfica y del modelo de datos, para determinar si lo antes realizado esta acorde con especificado en la fase de análisis de requerimientos.

#### **3.3.4.2 Implementación (Juego)**

En esta fase se realizó la codificación de las funciones necesarias para dar cumplimiento a los requerimientos asignados.

#### **3.3.4.3 Pruebas (Post-juego)**

En esta fase se realizó las pruebas funcionales de la aplicación para determinar si se cumplen los requerimientos establecidos por el usuario y asignados a la iteración, para lo cual las pruebas del desarrollo de la aplicación móvil se documentarán en la fase 4.

3.3.5 Iteración 2: Registrar especies vegetales por medio de una hoja

3.3.6 Iteración 3: Identificación de especies vegetales por medio de la hoja

3.3.7 Iteración 4.- Consulta de todas las especies vegetales

3.3.8 Iteración 5: Implementación de un servidor web online para respuesta a peticiones de aplicación móvil.

3.3.9 Iteración 6: compilar la aplicación para el sistema operativo Android e iOS.

3.3.10 Iteración 7: Publicar la aplicación en las tiendas de Android, e iOS.

### **3.4 Fase 4: Pruebas de validación y funcionamiento**

En la presente fase se realiza algunas pruebas a la aplicación móvil, dentro de las cuales se encuentran las pruebas funcionales en donde se valida los criterios de aceptación de cada requerimiento funcional en las plataformas Android, e iOS.

Además se desarrollarán pruebas de usabilidad y pruebas de interacción persona-dispositivo para lograr determinar si la aplicación desarrollada cumple con los objetivos iniciales y resuelve los problemas detectados de manera eficiente.

## **F. Resultados**

### **1. Fase 1: Determinación de la situación y formulación problemática**

En esta fase se determinó la problemática de presente PFC, que se especificó en el anteproyecto, la cual permitió tener una visión general de la situación actual, donde se pudo evidenciar que la mayoría de estudiantes de la carrera de Ingeniería Forestal, del Área de Agropecuaria y los Recursos Naturales Renovables (AARNR) de la UNL, tienen dificultad para identificar especies vegetales forestales de la localidad, debido a que no se cuenta con una base de datos móvil, situación que permitió plantear el problema de investigación el cual fue ¿La implementación de una guía de campo móvil facilitará la identificación de especies vegetales forestales nativas del Ecuador a la comunidad universitaria involucrada?, que fue el punto de partida para poder realizar una solución óptima al mismo.

### **2. Fase 2: Fase Exploratoria**

Esta fase se basó en una revisión de literatura a profundidad, con la finalidad de establecer las herramientas necesarias para la elaboración de aplicaciones móviles con reconocimiento de objetos, y las posibles alternativas que nos permitirán brindar solución al problema planteado.

Teniendo claro el problema a solucionar y las causas que lo originan, se selecciona en base a una lista de posibles soluciones y el estudio de las mismas la mejor solución por funcionalidad, factibilidad y accesibilidad. Logrando determinar las herramientas viables para el desarrollo del proyecto.

Para dar cumplimiento a esta fase se desarrollaron algunas tareas que se describirán a continuación:

#### **2.1 Análisis de Frameworks para el desarrollo de aplicaciones móviles Multiplataforma**

En lo que respecta este proyecto, se realiza un desarrollo híbrido. Se trata de un tipo de aplicación cuyo núcleo está escrito en *HTML5*, *CSS* y *Javascript*, encapsulado en un contenedor nativo.

Este tipo de aplicaciones híbridas permiten escribir parte de la aplicación en lenguaje nativo y parte de la aplicación en lenguaje *web*, aprovechando las ventajas de cada modo de programación. De este modo se tiene acceso total al dispositivo en aquellos puntos en los que la aplicación lo requiera, como por ejemplo determinadas propiedades

actualmente exclusivas de las aplicaciones nativas, como son el acceso a la cámara, geolocalización o el procesamiento de audio y vídeo [22].

Para dar solución a estas necesidades se ha desarrollado Frameworks como Apache Cordova, Titanium Appcelerator, Ionic, entre otros, los cuales permite desarrollar aplicaciones híbridas multiplataforma.

Cabe recalcar que cuando se hace uso de estos Frameworks siempre se tiene que realizar pequeños cambios en el código para adaptarlos a las diferentes plataformas correctamente, sin embargo estos cambios requieren menor esfuerzo que tener que desarrollar toda la aplicación desde cero para cada plataforma requerida [23].

Con la finalidad de determinar la herramienta idónea para desarrollar la aplicación móvil en el presente escenario de estudio, se realizó el análisis de dos Frameworks para de desarrollo de aplicación móviles multiplataforma, los cuales son Apache Cordova y Titanium Appcelerator. La Tabla II presenta el análisis comparativo realizado.

**TABLA II. COMPARATIVA ENTRE FRAMEWORKS APACHE CORDOVA Y TITANIUM APPCELERATOR**

<b>Apache Cordova</b>	<b>Titanium Appcelerator</b>
Utiliza tecnología Lenguaje <i>web</i> ( <i>HTML5</i> , <i>CSS3</i> , <i>Javascript</i> ) muy extendido, conocido, documentado, de fácil uso y aprendizaje, y único para todas las plataformas.	Interfaz de usuario utilizando JavaScript.
Empaqueta la aplicación utilizando el propio Framework.	Compila la aplicación a código nativo.
Permite acceder al hardware del dispositivo.	Permite acceder al hardware del dispositivo.
Permite compilar la aplicación en la nube mediante la herramienta Apache Cordova Build para la plataforma requerida.	Para compilar se debe utilizar un IDE propio para la plataforma requerida.
Se puede acceder a los controles nativos mediante JavaScript.	Necesita librerías para acceder a controles nativos.
Proceso de desarrollo sencillo por uso de tecnología web.	Su uso requiere una curva de aprendizaje bastante alargada
Permite desarrollar aplicaciones para los sistemas operativos Apple iOS, Google	Permite desarrollar aplicaciones para los sistemas operativos Apple iOS, Google

Android, Blackberry OS, Blackberry 10 WebOS, Windows Mobile y Symbian.	Android, BlackberryOS.
Software Libre (GPL).	Tiene una licencia Apache 2.0.

Luego de realizar el análisis comparativo entre los Frameworks antes mencionados se puede concluir que tanto Apache Cordova como Titanium permite desarrollar la aplicación móvil multiplataforma para la identificación de especies vegetales.

Sin embargo, analizando las características de cada uno de estos Frameworks se puede evidenciar que para el desarrollo de la aplicación móvil antes mencionada resulta factible utilizar Apache Cordova, puesto que se trata de una aplicación de consulta de datos en un servidor web en donde es primordial el diseño de la interfaz, misma que se la puede realizar eficientemente mediante el uso de tecnologías que nos brinda Apache Cordova como son HTML5, CSS3, Javascript. Además permite la integración con Frameworks Front-End como jQuery Mobile y Sencha Touch, permitiendo obtener una interfaz amigable e intuitiva. En la tabla III, se muestra una comparativa de las 2 tecnologías.

**TABLA III. COMPARATIVA FRAMEWORKS FRONT-END JQUERY MOBILE Y SENCHA TOUCH**

<b>Jquery Mobile</b>	<b>Sencha Touch</b>
Puede integrarse con Apache Cordova.	Puede integrarse con Apache Cordova.
Utiliza sintaxis fácil de entender.	Utiliza sintaxis que requiere una curva de aprendizaje alargada.
Cuenta con soporte para Ajax.	Cuenta con soporte para Ajax.
Trabaja con html5.	Trabaja con html5.
Se puede personalizar los temas en la herramienta ThemeRoller.	No existe una herramienta de personalización de temas similar.

Luego de realizar la comparativa se ha considerado pertinente utilizar jQuery Mobile, debido a la simplicidad de sintaxis en desarrollo de una interface amigable con el usuario.

## 2.2 Análisis de Frameworks y librerías para el reconocimiento de imágenes

Existen diferentes librerías en el mercado para el procesamiento de imágenes y las más conocidas se describen a continuación en la tabla IV, de las cuales se pretende tomar la más fácil de implementar y soportada por el servidor web a implementar.

**TABLA IV. LIBRERÍAS PARA EL PROCESAMIENTO DE IMÁGENES.**


OpenCV	MatLab	JAVAVIS
Disponible para los lenguajes de programación C++ y Python.	Disponible para los lenguajes de programación C++	Disponible para los lenguajes de programación java
Buena documentación para la implementación de reconocimiento de imágenes, acorde al caso de estudio.	Poca documentación para la implementación de reconocimiento de imágenes, acorde al caso de estudio.	Poca documentación para la implementación de reconocimiento de imágenes, acorde al caso de estudio.
Para su instalación necesita un espacio pequeño en el disco.	Para su instalación necesita un espacio amplio en el disco.	Para su instalación necesita un espacio pequeño en el disco.
Curva de aprendizaje dedicada.	Fácil instalación.	Fácil instalación.
Acceso gratuito.	Acceso restringido	Acceso gratuito

Luego de haber analizado las librerías disponibles para la implementación, se ha optado por la librería OpenCV por su amplia documentación y fácil acceso en la nube.



## 2.3 Análisis de proveedores para el alojamiento de servicios web en la nube

En la tabla V, se realiza una tabla comparativa de los diferentes proveedores que ofrecen alojamiento gratuito, para aplicaciones o implementación de servicios web.

**TABLA V. LISTADO DE PROVEEDORES CON ALOJAMIENTO GRATUITO**

	Parse un lugar donde alojar nuestras aplicaciones en la nube sin necesidad de montar ningún servidor, parse trabaja con las plataformas IOS, OSX, Android, JavaScript, Windows Phone 8, Windows 8, .NET, Xamarin, Unity y REST API. Además nos ofrecen un servicio de monitorización en tiempo real de nuestras
---	---

	<p>aplicaciones o incluso ejecutar nuestros propios programas en la nube, con Parse podemos crear nuestras redes sociales en cuestión de minutos. Un servicio bastante interesante para comenzar a desarrollar debido a que su servicio gratuito está bastante bien.</p>
	<p>Nitrous.IO nos ofrecen de forma gratuita desplegar una aplicación, con 384Mb de RAM y 750Mb de espacio en disco. Podemos alojar proyectos en: Ruby/Rails, Node.js, Python/Django, Go. Dispone de un Web IDE bastante completo el cual se puede usar para entorno colaborativos, permite conexión con GitHub. Podemos usar diferentes bases de datos como MongoDB o Mysql o trabajar con memcached entre otras opciones. En lugar a tener en cuenta por su gran diversidad de servicios que ofrecen.</p>
	<p>Openshift es un proyecto de Red Hat de computación en la nube. Podemos trabajar con diferentes lenguajes como Java, Ruby, PHP, Node.js, Python y Perl, o usar frameworks CakePHP, Django, Ruby/Rails o Tornado entre otros. En cuanto a base de datos podemos usar MongoDB, Mysql o PostgreSQL. Dispone conexión con GitHub y acceso SSH. Es un lugar bastante robusto y con una gran comunidad.</p>
	<p>Appfog nos ofrece 2Gb de Ram y la posibilidad de instalar 8 servicios de forma gratuita. Nos permite desplegar aplicaciones en Java, Java rails, Java Spring Mysql, Node, Node Express, PHP, Drupal, Django, Python, Ruby/Rails entre otros. Como base de datos podemos usar Mysql, MongoDB, PostgreSQL, RabbitMQ o Redis. Además nos permite instalar diferentes Addons como por ejemplo SendGrid, MongoHQ, ElephantSQL, Logentries entre un gran listado. También puedes elegir la región donde desplegar tus sitios. Para mi es el lugar más completo que conozco.</p>

	<p>En Heroku puedes programar en Ruby, Node.js, Clojure, Java, Python y Scala, podemos trabajar con Postgres, Memcache, Mongo, Redis o Hadoop entre otros servicios, nos ofrecen la posibilidad de instalar multitud de addons, como Cluster Mysql, MongoHQ, Treasure Data Hadoop, MongoLab, Pusher, Spreedplyentre otros. Usa Git para los despliegues y ofrece un servicio para clonar rápidamente nuestras aplicaciones. Otra alternativa a considerar gracias a la cantidad de Addons que nos ofrecen.</p>
	<p>Cloudno.de es otra de las alternativas que nos permiten tener un plan gratuito, está pensada para el desarrollo en Node.js y es de las pocas que ofrecen CouchDB, además de Redis, nos permite usarWebSockets y usa para desplegar Git. Podemos usar nuestros propios dominios o un subdominio que elijamos cuando creemos nuestra aplicación.</p>

Luego de analizar los proveedores disponibles se ha optado por openshift, que ofrece una extensa documentación en la nube, un sinnúmero de lenguajes de programación y bases de datos conocidas.

Openshift permite el acceso remoto a su aplicación a través del protocolo Secure Shell (SSH). Utiliza la línea de comandos para gestionar directamente el servidor, permite comprobar los registros y probar los cambios rápidos [24]. Además admite la configuración e instalación de herramientas en el servidor acorde a la necesidad.

## 2.4 Análisis de Frameworks para la implementación de un servidor web

Al haber elegido la librería para reconocimiento de imágenes OpenCV, que puede trabajar con los lenguajes de programación C++ y python, se ha elegido trabajar con el lenguaje de programación Python, a continuación se en la tabla VI, realiza una comparación de los frameworks más conocidos y de fácil implementación en python:

**TABLA VI. COMPARATIVA DE FRAMEWORKS WEB ESCRITOS EN PYTHON.**

Bottle	Tornado	Django
Proceso de desarrollo sencillo	Proceso de desarrollo sencillo	Su uso requiere una curva de aprendizaje bastante alargada



Un solo hilo request-response	Configuración de múltiples hilos request-response	Múltiples hilos request-response
Compatible con Python 2.7 o 3.4	Compatible con Python 2.7 hasta 3.4	Compatible con Python 2.7 o 3.4
Framework de desarrollo web de código abierto	Framework de desarrollo web de código abierto	Framework de desarrollo web de código abierto
Buena documentación.	Buena documentación.	Buena documentación.
Con bloqueo de servidor web	Sin bloqueo del servidor web	Bloqueo configurable del servidor web
BSD License	Licencia Apache 2.0	BSD License
Última versión 2013	Estado de desarrollo activo	Estado de desarrollo activo

Luego de haber analizado los frameworks más conocidos se ha optado por utilizar el servidor web “Tornado” por su fácil implementación.

## 2.5 Análisis de herramientas para la implementación de bases de datos

Al haber elegido un servidor web escrito en python se debe implementar una base de datos compatible con dicho lenguaje de programación, a continuación en la tabla VII, se analiza las herramientas más comunes en el mercado para la implementación de una base de datos en el servidor web a implementar.

**TABLA VII.** COMPARATIVA DE HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE BASE DE DATOS

PostgreSQL	Mysql
Sistema operativo multiplataforma	Sistema operativo multiplataforma
Librería disponible para la comunicación con python.	Librería disponible para la comunicación con python.
Extensa documentación.	Extensa documentación.
Licencia BSD	GPL o Uso comercial
Amplia variedad de tipos de datos nativos.	Variedad de tipos de datos nativos
Alta concurrencia.	Multihilo y multiusuario.

Al poseer conocimientos de Postgres se decidió utilizar esta herramienta, además por su amplia documentación, libre acceso y amplia variedad de tipos de datos nativos.

## 2.6 Selección de la Metodología para el desarrollo de la aplicación móvil.

Una metodología es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de software [25]. Existen dos tipos de metodologías para el desarrollo de software las ágiles y las tradicionales.

Las metodologías tradicionales se basan principalmente en dividir el proceso de desarrollo en diferentes etapas. Este tipo de metodologías imponen una disciplina rígida y se centran en el control del proceso mediante la definición de roles, actividades, herramientas, notaciones para modelado y documentación detallada. Las metodologías tradicionales no se adaptan fácilmente a los cambios, por esta razón no son recomendables utilizarlas donde los requisitos del sistema puedan cambiar. Entre las metodologías tradicionales se encuentran RUP, Iconix, entre otros.

Las metodologías ágiles se centran en la creación de iteraciones durante el ciclo de vida del proyecto. Cada iteración contiene planificación, análisis, diseño, desarrollo, pruebas y documentación. Estas metodologías son adaptables a diferentes tipos de proyectos.

Los valores de las metodologías ágiles son:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

Entre las metodologías ágiles se encuentra XP (Extreme Programming), Scrum, ASD (Adaptive Software Development), entre otros [26].

La tabla VIII presenta un análisis comparativo entre las principales características de las metodologías ágiles y las metodologías tradicionales.

**TABLA VIII.** COMPARATIVA ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES

Metodologías ágiles	Metodologías tradicionales
El cliente es parte del equipo.	El cliente participa en reuniones.
Poca documentación.	Documentación exhaustiva.
No existe un contrato fijado.	Existe un contrato fijado.
Mayor flexibilidad ante los cambios.	Resistencia ante cambios.
Proceso de desarrollo flexible.	Proceso de desarrollo más controlado.
Grupos de desarrollo grandes.	Grupos de desarrollo mediano o pequeño.

Analizando el cuadro comparativo presentado anteriormente se puede notar los beneficios que proporcionan las metodologías ágiles para el desarrollo de software. Para la implementación de aplicaciones móviles es conveniente utilizar este tipo de metodologías sobre todo por la flexibilidad que estas proporcionan. Para determinar la metodología a utilizar en el presente proyecto se realizó un análisis de dos metodologías ágiles más utilizadas para el desarrollo de aplicaciones móviles como son: Scrum y XP (Extreme Programming) [27]. La tabla IX presenta el análisis comparativo realizado.

**TABLA IX. COMPARATIVA DE LAS METODOLOGÍAS DE DESARROLLO ÁGIL SCRUM Y XP [28].**

Scrum	XP(Extreme Programming)
Utiliza herramientas actuales y productivas.	Ligero pero eficiente.
Los equipos son altamente productivos con prioridades y responsabilidades definidas.	Se centra en la productividad y la calidad, mismas que son evaluadas al finalizar el sistema.
Se adapta a cualquier tipo de proyectos.	Se utiliza para proyectos pequeños.
Equipo de trabajo multidisciplinario.	Programación en parejas.

Luego de analizar las características de estas metodologías se ha decidido utilizar Scrum para el desarrollo del presente proyecto debido a su adaptabilidad para proyectos de desarrollo de aplicaciones móviles a corto plazo.

### 3. Fase 3: Fase Experimental

Teniendo claro las herramientas a utilizar en el desarrollo del presente PFC, es necesario conocer el equipo de trabajo con el que se cuenta para el desarrollo de la aplicación móvil, a continuación en la tabla X, se muestra los participantes del proyecto.

**TABLA X. EQUIPO DE TRABAJO.**

No.	Nombre	Rol
1	Product Owner (Propietario del producto)	Universidad Nacional de Loja
2	Scrum Master (Facilitador)	Freddy Chamorro
3	Scrum Team (Equipo)	Freddy Chamorro Ana Lucía Colala Troya
4	Usuarios	Comunidad universitaria de la UNL

Posterior a la conformación del equipo de trabajo, se realizaron las tareas (sprint) definidas según la metodología SCRUM para el desarrollo de la guía de campo móvil para la identificación de especies vegetales forestales tomando como piloto el jardín botánico “Reinaldo Espinosa” de la UNL.

### **3.1 Iteración 0: Análisis y diseño de la aplicación**

#### **3.1.1 Especificación de requerimientos**

##### **3.1.1.1 Introducción**

La especificación de requerimientos permite recopilar la información necesaria que servirá de guía para los desarrolladores del software, puesto que aquí se especifica cada una de las necesidades del cliente a fin de que al finalizar el proyecto se pueda verificar el cumplimiento de cada uno de los requerimientos establecidos.

En esta fase se describe el personal involucrado, funciones del software, requisitos funcionales, requisitos no funcionales, entre otros.

##### **3.1.1.2 Propósito.**

Determinar las funcionalidades que tendrá la guía de campo móvil para la identificación de especies vegetales forestales tomando como piloto el jardín botánico “Reinaldo Espinosa” de la UNL.

##### **3.1.1.3 Ámbito del Sistema.**

La aplicación móvil permitirá a la comunidad universitaria involucrada realizar consultas de todas las especies vegetales registradas en el inventario, identificar las especies vegetales por medio de una hoja de la especie vegetal, además registrar aquellas que no constan en el inventario y registrar muestras de especies ya existentes.

##### **3.1.1.4 Personal Involucrado**

En las tablas XI, XII, se detallan los datos del personal involucrado en el desarrollo del presente proyecto.

**TABLA XI. DATOS DEL ANALISTA/DESARROLLADOR DEL PROYECTO**

<b>Nombres</b>	Freddy Alexander
<b>Apellidos</b>	Chamorro Encalada
<b>Dirección</b>	Malacatos-Loja
<b>Teléfono</b>	0982825576
<b>Rol</b>	Analista/desarrollador
<b>Cargo</b>	Analista/desarrollador
<b>Categoría Profesional</b>	Superior
<b>Responsabilidades</b>	Realizar el análisis de requerimientos.

	Implementar la guía de campo móvil para la identificación de especies vegetales forestales tomando como piloto el jardín botánico “Reinaldo Espinosa” de la UNL.
--	--

**TABLA XII. DATOS DEL SUPERVISOR DEL PROYECTO**

<b>Nombres</b>	Ana Lucía
<b>Apellidos</b>	Colala Troya
<b>Dirección</b>	Loja
<b>Teléfono</b>	0997214629
<b>Rol</b>	Analista/desarrollador
<b>Cargo</b>	Supervisor del proyecto
<b>Categoría Profesional</b>	Superior
<b>Responsabilidades</b>	Realizar las revisiones del proyecto.

### 3.1.1.5 Definiciones y Abreviaturas

Las definiciones especificadas permiten clarificar los términos utilizadas y las abreviaciones reducir las expresiones que se utilizan con frecuencia en este proyecto, con la finalidad de ocupar menos espacio dentro de los cuadros de requerimientos, casos de uso, entre otros.

#### a. Definiciones

Aplicación móvil: Una aplicación móvil es una aplicación informática que puede ejecutarse en teléfonos inteligentes, tabletas y otros dispositivos móviles.

Interfaz: Permite la interacción del usuario con la aplicación.

Sistema operativo: Es un programa que gestiona los recursos de un computador.

#### b. Abreviaturas

UNL: Universidad Nacional de Loja.

AARNR: Área Agropecuaria y de los Recursos Naturales Renovables.

RF: Requerimiento Funcional.

RNF: Requerimiento no Funcional.

CU: Caso de uso

### 3.1.1.6 Descripción General del Producto

Se realizará una aplicación móvil como guía de campo móvil para la identificación de especies vegetales forestales tomando como piloto el jardín botánico “Reinaldo Espinosa” de la UNL, misma que pretende brindar a la comunidad universitaria involucrada (estudiantes, docentes del AARNR y visitantes de la UNL), que concurren al jardín botánico “Reinaldo Espinosa”, una alternativa innovadora y eficaz para que puedan consultar, identificar las especies vegetales y registrar las que no estén inventariadas.

La aplicación móvil utilizará un servidor web para realizar consultas, identificación y registro de especies vegetales, por lo cual se implementará un servidor que proporcione estos servicios.

#### a. Funciones del Producto

En la aplicación móvil permitirá realizar las siguientes acciones:

- Autenticación de usuario.
- Consulta en línea de especies vegetales forestales nativas.
- Identificación en línea de especies vegetales forestales nativas.
- Registro en línea de especies vegetales y recolección de muestras de especies ya inventariadas.

#### b. Características de los usuarios

En las tablas XIII, XIV, se muestran los tipos de usuarios que tendrán contacto directo con la aplicación móvil.

**TABLA XIII. TIPO DE USUARIO COLABORADOR**

<b>Tipo de usuario</b>	Colaborador
<b>Formación</b>	Estudiante de pregrado, docente.
<b>Habilidades</b>	Conocimientos medios y altos en el manejo de tecnología móvil. Conocimientos medios y altos en el reconocimiento de especies vegetales,
<b>Actividades</b>	Este tipo de usuarios podrá consultar, identificar por las hojas las especies vegetales inventariadas, y registrar las especies vegetales no inventariadas.

**TABLA XIV. TIPO DE USUARIO CONSULTOR**

<b>Tipo de usuario</b>	Consultor
<b>Formación</b>	General
<b>Habilidades</b>	Conocimientos bajos y medios en el manejo de tecnología móvil.
<b>Actividades</b>	Este tipo de usuarios podrá consultar, e identificar por las hojas las especies vegetales inventariadas.

### **c. Restricciones**

La aplicación móvil debe ser desarrollada con las siguientes restricciones:

- Se hará uso de la metodología Scrum.
- Para el desarrollo de la Aplicación móvil se utilizará el Framework Apache Cordova.
- Se utilizará tecnologías como JavaScript, HTML5 y CSS3.
- Para la implementación del servidor web se utilizará el framework tornado de python, una base de datos Postgresql, y la librería para el reconocimiento de imágenes OpenCV.

### **d. Suposiciones y Dependencias**

El dispositivo móvil en el que funcionará la aplicación debe contar con un sistema Android o iOS.

#### **3.1.1.7 Requisitos Específicos**

##### **a. Requisitos comunes de las interfaces**

###### **i. Interfaz de usuario.**

- La interfaz gráfica de la aplicación móvil será diseñada de tal manera que sea fácil e intuitiva de utilizar.
- Se elaborará un manual de usuario, mismo que servirá como guía a los usuarios de la misma.

###### **ii. Interfaz de Hardware**

La aplicación se ejecutará en un dispositivo móvil Android, iOS, las versiones soportadas por cada sistema operativo se muestran en la tabla XI.

**TABLA XV. VERSIONES DE CADA SISTEMA OPERATIVO SOPORTADO POR LA APLICACIÓN MÓVIL.**

<b>Sistema Operativo</b>	<b>Versiones</b>
Android	2.3.X – 4.4.X
iOS	7 – 8.3

### iii. Interfaz de Software

Se requiere que el dispositivo cuente con Sistema Operativo Android, o iOS.

### iv. Interfaz de Comunicación.

Se requiere que el dispositivo tenga conexión a Internet cuando se vaya a usar de la aplicación.

#### b. Requerimientos Funcionales

<b>Referencia</b>	RF-001	
<b>Nombre</b>	Ingresar a la aplicación	
<b>Descripción</b>	La aplicación móvil permitirá al usuario ingresar a la aplicación mediante el correo electrónico y clave, previamente registrados.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
Criterios de Aceptación		Estado
Verificar que al ejecutar la aplicación móvil se muestre en la pantalla <b>[Inicio de sesión]</b> , la opción para la identificación del usuario mediante clave y correo electrónico personal.		Por verificar
Verificar que al seleccionar la opción <b>[Recordar Usuario]</b> de la pantalla de Inicio de sesión, se guarde los datos de identificación del usuario.		Por verificar
Verificar si al presionar el botón <b>[Entrar]</b> con uno o los dos campos vacíos, la aplicación presenta un mensaje indicativo.		Por verificar
Verificar si al presionar el botón <b>[Entrar]</b> sin una conexión a internet, la aplicación presenta un mensaje indicativo.		Por verificar
Verificar si cuando se ingresa en correo electrónico o la clave incorrecta, la aplicación presenta un mensaje indicando la razón por la que no se puede ingresar a la pantalla <b>[Principal]</b> .		Por verificar
Verificar si al ingresar los datos de la cuenta correctamente, la aplicación muestra la pantalla <b>[Principal]</b> con el perfil correspondiente al usuario.		Por verificar

<b>Referencia</b>	RF-002
<b>Nombre</b>	Registrar en la aplicación
<b>Descripción</b>	La aplicación móvil permitirá al Usuario registrarse en la aplicación con sus datos personales (nombres, apellidos, correo electrónico, clave y perfil de usuario).



<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al presionar el botón <b>[Registrar]</b> , con los nombre o apellidos que contienen caracteres no válidos, la aplicación presenta un mensaje indicativo.		Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> , con el correo electrónico no válido, la aplicación presenta un mensaje indicativo.		Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> , con uno o varios campos vacíos, la aplicación presenta un mensaje indicativo.		Por verificar
Permitir al usuario registrar en la aplicación con dos tipos de perfiles, " <b>Consultor</b> ", para realizar identificación de especies vegetales en la aplicación, o " <b>Colaborador</b> ", para aportar en el registro de especies vegetales a la aplicación.		Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> con los datos correctos, la aplicación presenta un mensaje indicativo y regresa a la pantalla de Inicio de sesión.		Por verificar

<b>Referencia</b>	RF-003	
<b>Nombre</b>	Verificar perfil de usuario	
<b>Descripción</b>	La aplicación móvil permitirá al usuario visualizar las opciones disponibles en la aplicación de acuerdo a su perfil de usuario.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al ingresar en la aplicación, en la pantalla principal se muestran las opciones disponibles de acuerdo al perfil de usuario elegido en el registro.		Por verificar
Verificar si al ingresar en la aplicación, se muestra los datos de usuario, nombres y apellidos, que corroboren el inicio de sesión del usuario.		Por verificar

<b>Referencia</b>	RF-004
<b>Nombre</b>	Consultas de especies vegetales forestales nativas

<b>Descripción</b>	La aplicación móvil permitirá al usuario consultar todas las especies vegetales registradas en la aplicación.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al presionar el botón <b>[Consultar]</b> , se muestra la pantalla de búsqueda especies vegetales.		Por verificar
Verificar si al mostrar la pantalla de Búsqueda de especies vegetales, se muestran todas las especies vegetales registradas en la aplicación móvil.		Por verificar
Verificar si al filtrar la búsqueda de especies vegetales, se filtra el listado por el nombre común o científico de la especie vegetal.		Por verificar
Verificar si al seleccionar de la lista, la especie vegetal de interés, se muestra la pantalla Información detallada de la especie vegetal.		Por verificar

<b>Referencia</b>	RF-005	
<b>Nombre</b>	Identificación de especies vegetales forestales nativas	
<b>Descripción</b>	La aplicación móvil permitirá al usuario identificar las especies vegetales por medio de una hoja de la especie vegetal en referencia a las especies vegetales registradas en la aplicación.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al presionar el botón <b>[Identificar]</b> , se muestran la cámara del dispositivo móvil, para la captura de la hoja de la especie vegetal.		Por verificar
Verificar si al capturar la imagen de la hoja de la especie vegetal, se muestra la pantalla de Identificación de la especie vegetal.		Por verificar
Verificar si en la pantalla de Identificación de especie vegetal, se muestra la captura de la imagen de la especie vegetal.		Por verificar
Verificar si al seleccionar el botón <b>[Recapturar Hoja]</b> , se muestra la cámara del dispositivo para recapturar la especie vegetal de acuerdo a las especificaciones.		Por verificar

Verificar si al presionar el botón <b>[Buscar Especie]</b> , sin una conexión a internet, la aplicación muestra un mensaje indicativo.	Por verificar
Verificar si al presionar el botón <b>[Identificar Especie]</b> , la aplicación muestra la pantalla <b>[Búsqueda de especies vegetales]</b> , con las cinco especies vegetales más cercanas a la búsqueda.	Por verificar

<b>Referencia</b>	RF-006
<b>Nombre</b>	Registro de especies vegetales forestales nativas
<b>Descripción</b>	La aplicación móvil permitirá al usuario con perfil " <b>Colaborador</b> ", registrar las especies vegetales por medio de una hoja de la especie vegetal, con nombre común y científico, además de las coordenadas geográficas del registro de la especie vegetal.
<b>Prioridad</b>	Alta
<b>Versión</b>	1
<b>Criterios de Aceptación</b>	
	<b>Estado</b>
Verificar si al presionar el botón <b>[Registrar]</b> , se muestra la pantalla Registro de especie vegetal, con los datos de geolocalización actual (latitud, longitud).	Por verificar
Verificar si al presionar el botón <b>[Capturar]</b> , se muestra la cámara del dispositivo para capturar la hoja de la especie vegetal.	Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> , con uno o los dos campos vacíos, la aplicación muestra un mensaje indicativo.	Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> , caracteres ingresados no válidos, la aplicación muestra un mensaje indicativo.	Por verificar
Verificar si al presionar el botón <b>[Registrar]</b> , sin capturar una imagen, la aplicación muestre un mensaje indicativo.	Por verificar

<b>Referencia</b>	RF-007
<b>Nombre</b>	Implementar el servidor web para respuesta de peticiones de la aplicación móvil.
<b>Descripción</b>	Servidor web para responder peticiones de aplicación móvil para el reconocimiento de especies vegetales forestales nativas.
<b>Prioridad</b>	Alta

<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor web, recibe peticiones de la aplicación móvil y responde las peticiones en formato JSON.		Por verificar
Verificar que el servidor web responda de manera rápida a las solicitudes de la aplicación móvil.		Por verificar
Verificar que el servidor web, sea de acceso público.		Por verificar

<b>Referencia</b>	RF-008	
<b>Nombre</b>	Verificar usuario en el Servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de verificación de usuario emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de verificación de usuario con los datos emitidos desde la aplicación móvil.		Por verificar
Verificar que el servidor emita una respuesta en formato JSON.		Por verificar

<b>Referencia</b>	RF-009	
<b>Nombre</b>	Registrar usuario en la Base de Datos.	
<b>Descripción</b>	El servidor web recibe la petición de registro de usuario desde la aplicación móvil con los datos personales, para el registro en la Base de datos.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de registro de usuario con los datos emitidos desde la aplicación móvil.		Por verificar
Verificar que el servidor valida los datos del usuario con los existentes en la Base de Datos.		Por verificar
Verificar que el servidor registra el nuevo usuario en la base de datos.		Por verificar
Verificar que el servidor emita una respuesta en formato JSON.		Por verificar

<b>Referencia</b>	RF-0010	
<b>Nombre</b>	Consultar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de consulta de todas las especies vegetales registradas emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de consultar todas las especies vegetales emitida desde la aplicación móvil.		Por verificar
Verificar que el servidor consulte en la Base de Datos, todas las especies vegetales registradas.		Por verificar
Verificar que el servidor emita una respuesta en formato JSON.		Por verificar

<b>Referencia</b>	RF-0011	
<b>Nombre</b>	Identificar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de identificación de la especie vegetal emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de identificación de especies vegetales con la imagen de la hoja de la especie vegetal emitida desde la aplicación móvil.		Por verificar
Verificar que el servidor identifique la especie vegetal de acuerdo a las muestras registradas en la Base de Datos.		Por verificar
Verificar que el servidor emita una respuesta en formato JSON.		Por verificar

<b>Referencia</b>	RF-0012	
<b>Nombre</b>	Registrar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de registro de la especie vegetal, y guarda la especie vegetal emitida por la aplicación móvil en la Base de Datos.	

<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de registro de especies vegetales con los datos de la especie vegetal emitida desde la aplicación móvil.		Por verificar
Verificar que el servidor valide la especie vegetal de acuerdo a las muestras registradas en la Base de Datos.		Por verificar
Verificar que el servidor registra la nueva especie vegetal o muestra de una especie vegetal existente en la base de datos.		Por verificar
Verificar que el servidor emita una respuesta en formato JSON.		Por verificar

### c. Requerimientos no Funcionales.

#### i. Rendimiento

<b>Referencia</b>	RNF-001
<b>Nombre</b>	Tiempos de consulta
<b>Descripción</b>	La aplicación móvil permitirá al usuario identificar especies vegetales en tiempo real.
	El tiempo de respuesta al usuario será el mínimo posible, permitiendo a los usuarios realizar la identificación de manera fácil y rápida.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

#### ii. Seguridad

<b>Referencia</b>	RNF-002
<b>Nombre</b>	Ingreso a la aplicación
<b>Descripción</b>	Para acceder a la aplicación los usuarios deben registrarse.
	Los usuarios deben autenticarse mediante correo electrónico y contraseña para acceder a la aplicación.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

### iii. Disponibilidad

<b>Referencia</b>	RNF-003
<b>Nombre</b>	Acceso a la aplicación
<b>Descripción</b>	La aplicación estará disponible al 100%, puesto que podrá ser instalada en el dispositivo móvil permitiendo el acceso en todo momento y desde cualquier lugar, siempre y cuando el dispositivo tenga una conexión a internet.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

### iv. Mantenibilidad

<b>Referencia</b>	RNF-004
<b>Nombre</b>	Actualización de la aplicación
<b>Descripción</b>	La aplicación se desarrollará de tal manera que en el futuro se puedan agregar nuevas funcionalidades. Se documentará todo el desarrollo de la aplicación con la finalidad de reutilizar el proyecto para futuros trabajos de desarrollo de aplicaciones móviles.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

### v. Portabilidad

<b>Referencia</b>	RNF-005
<b>Nombre</b>	Sistema operativo
<b>Descripción</b>	La aplicación será desarrollada para los sistemas operativos, Android e iOS, la misma que se encontrará disponible de forma gratuita en las tiendas online de cada plataforma.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

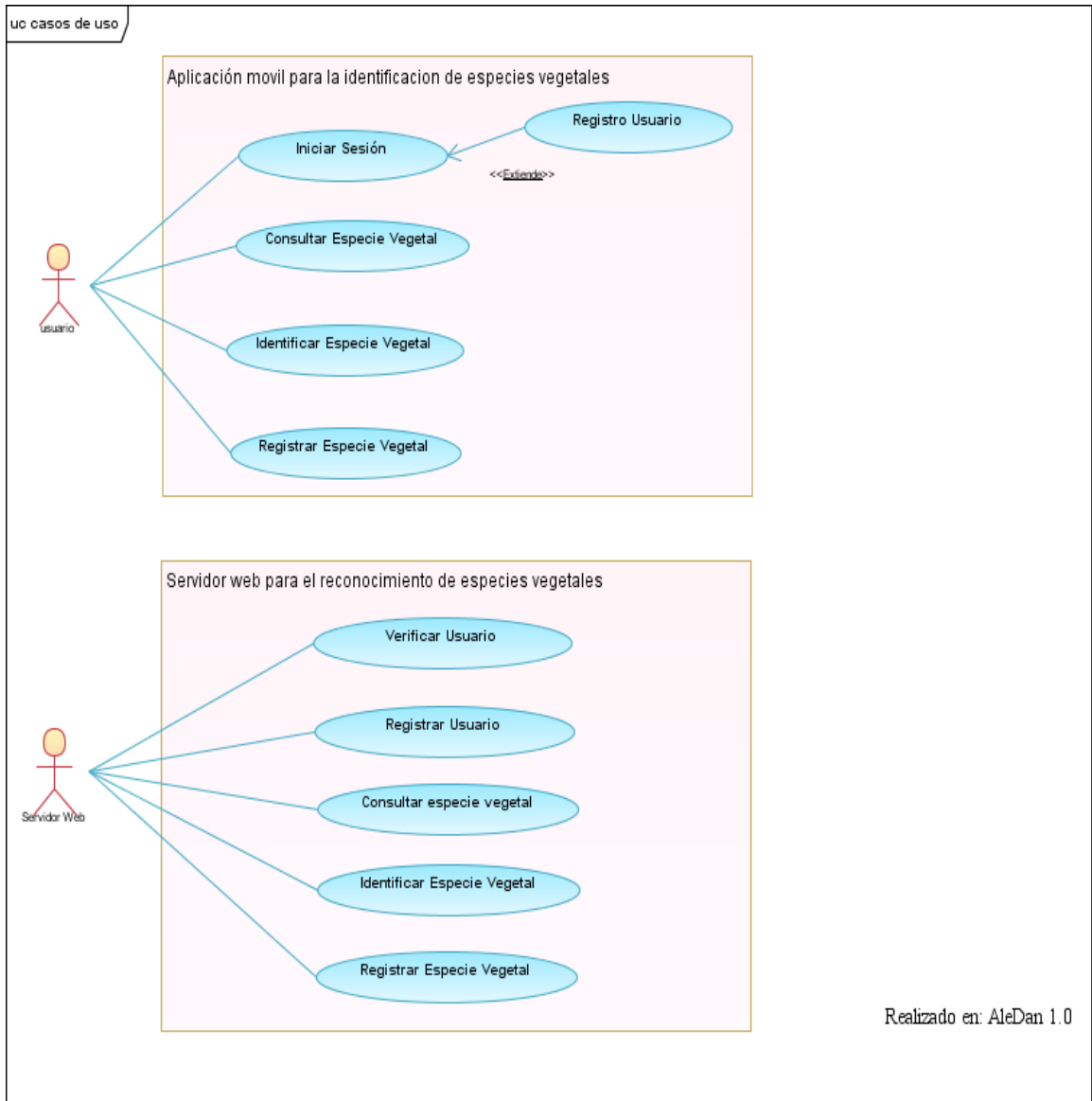
## vi. Usabilidad

<b>Referencia</b>	RNF-006
<b>Nombre</b>	Uso de la aplicación
<b>Descripción</b>	La aplicación tendrá una interfaz agradable e intuitiva, permitiendo a los usuarios realizar consultas y registros de especies vegetales eficientemente.
	Se mostrará la información necesaria de las especies vegetales registradas, evitando que los usuarios realicen acciones innecesarias.
<b>Prioridad</b>	Alta
<b>Versión</b>	1

### 3.1.2 Diagrama de Casos de uso

Utilizando los requerimientos funcionales obtenidos del usuario, se construye el diagrama de casos de uso que permite comprender el funcionamiento deseado de la aplicación móvil y determinar las características del mismo. En el presente caso de estudio el usuario interactúa con la aplicación móvil para registrarse, iniciar sesión, consultar las especies vegetales registradas, identificar una especie vegetal por medio de una hoja de la misma y registrar las no inventariadas. Una vez realizada cualquiera de las acciones anteriores la aplicación móvil realiza la petición al servidor web, el que se encarga de autenticar o registrar el usuario, y consultar, identificar o registrar las especies vegetales en la base de datos. En la figura 7 se presenta el diagrama de casos de uso para la implementación de la aplicación móvil y el servidor web.





**Figura 7:** Diagrama de casos de uso

### 3.1.3 Descripción de Casos de Uso

Una vez realizado el diagrama de casos de uso se efectúa una descripción detallada de cada, uno con la finalidad de determinar las características que tendrá la aplicación móvil, para lo cual se utilizó una plantilla donde se incluyen: nombre del caso de uso, código, tipo, actor, pantallas de referencia, requerimientos de referencia, resumen, objetivos, pre-condición, post-condición, curso normal de eventos, y curso alterno.

### 3.1.3.1 CU001: Iniciar Sesión

<b>Caso de Uso:</b>	Iniciar Sesión	<b>Actor:</b>	Usuario
<b>Código:</b>	CU001	<b>Tipo:</b>	Primario-Sistema
<b>Pantallas:</b>	P001-P003		
<b>Ref. Req.</b>	RF001 - RF003		
<b>Resumen:</b>	El usuario ingresa su correo electrónico personal y clave para su autenticación antes de ingresar a la aplicación móvil.		
<b>Objetivos:</b>	Autenticar el usuario antes de ingresar a la aplicación.		
<b>Precondición</b>	Que el usuario se haya registrado en la aplicación en la aplicación móvil.		
<b>Post condición</b>	Usuario autenticado correctamente.		
<b>Curso normal de eventos</b>			
1. La aplicación presenta la pantalla <b>[Inicio de sesión]</b> .			
2. El usuario ingresa el correo electrónico personal y su clave en la pantalla <b>[Iniciar sesión]</b> .			
3. El usuario presiona el botón <b>[Ingresar]</b> , en la pantalla <b>[Iniciar Sesión]</b> .			
4. La aplicación valida que los campos obligatorios de la pantalla <b>[Iniciar Sesión]</b> , no estén vacíos.			
5. La aplicación valida que el correo electrónico y clave de usuarios sean correctos.			
6. La aplicación envía y valida la información el servidor web, y la aplicación móvil muestra la pantalla <b>[Principal]</b> , con las opciones acorde al perfil de usuario.			
7. EL caso de uso finaliza.			
<b>Cursos alternos de eventos</b>			
<b>A. Campos obligatorios vacíos en Ingresar a la aplicación</b>			
A.1 La aplicación muestra el mensaje "Campos obligatorios vacíos", en la pantalla <b>[Iniciar Sesión]</b> .			
<b>B. Usuario y/o clave, o correo electrónico incorrectos.</b>			
B.1 La aplicación muestra el mensaje "Usuario no válido", en la pantalla <b>[Iniciar Sesión]</b> .			
B.2 La aplicación muestra el mensaje "La dirección de correo electrónico es incorrecta", en la pantalla <b>[Iniciar Sesión]</b> .			

<p><b>C. Servicio no disponible</b></p> <p>C.1 En el caso que no existe una conexión a internet la aplicación presentará el mensaje "Conexión a internet no disponible", en la pantalla <b>[Iniciar Sesión]</b>.</p> <p>C.2 En el caso que el servidor web no esté disponible la aplicación presentará el mensaje "Servidor no disponible en estos momentos", en la pantalla <b>[Iniciar Sesión]</b>.</p>
<p><b>D. Recordar Usuario</b></p> <p>D.1 El usuario puede seleccionar la opción <b>[Recordar Usuario]</b>, de la pantalla <b>[Iniciar Sesión]</b>, para mantener sus datos en la aplicación.</p>

### 3.1.3.2 CU002: Registro Usuario

<b>Caso de Uso:</b>	Registro de Usuario	<b>Actor:</b>	Usuario
<b>Código:</b>	CU002	<b>Tipo:</b>	Primario-Sistema
		<b>Pantallas:</b>	P001 -P002
<b>Ref. Req.</b>	RF001 - RF002		
<b>Resumen:</b>	El usuario registra sus datos personales en la aplicación móvil para obtener acceso a la aplicación móvil.		
<b>Objetivos:</b>	Registrar el usuario para obtener acceso a la aplicación móvil.		
<b>Precondición</b>	Que el usuario no se haya registrado en la aplicación en la aplicación móvil.		
<b>Post condición</b>	EL usuario puede autenticarse correctamente.		
<b>Curso normal de eventos</b>			
1 .La aplicación presenta la pantalla <b>[Inicio de sesión]</b> .			
2. El usuario presiona el botón <b>[Registrar]</b> en la pantalla [Iniciar Sesión].			
3. La aplicación presenta la pantalla <b>[Registro de Usuario]</b> .			
4. El usuario ingresa sus datos personales: nombres, apellidos, correo electrónico y clave.			
5. El usuario selecciona su perfil de usuario: " <b>Consultor</b> " o " <b>Colaborador</b> ", acorde a sus necesidades.			
6. La aplicación valida que los campos obligatorios de la pantalla <b>[Registro de Usuario]</b> , no estén vacíos.			
7. La aplicación valida que los nombres, apellidos, clave y correo electrónico del usuario sean correctos.			
8. La aplicación móvil envía y valida la información en el servidor web, y la aplicación muestra la pantalla <b>[Iniciar sesión]</b> , con el mensaje "Usuario registrado correctamente".			
9. EL caso de uso finaliza.			

Cursos alternos de eventos	
<b>A. Campos obligatorios vacíos en registrar usuario en la aplicación.</b>	
A.1 La aplicación muestra el mensaje "Campos obligatorios vacíos", en la pantalla <b>[Registro de Usuario]</b> .	
<b>B. Datos de usuario incorrectos.</b>	
B.1 La aplicación muestra el mensaje "Nombres y/o Apellidos no validos", en la pantalla <b>[Registro de Usuario]</b> .	
B.2 La aplicación muestra en mensaje "La dirección de correo electrónico es incorrecta", en la pantalla <b>[Registro de Usuario]</b> .	
B.3 La aplicación muestra el mensaje "El correo electrónico del usuario a registrar ya existe", en la pantalla <b>[Registro de Usuario]</b> .	
<b>C. Servicio no disponible</b>	
C.1 En el caso que no existe una conexión a internet la aplicación presentará el mensaje "Conexión a internet no disponible", en la pantalla <b>[Iniciar Sesión]</b> .	
C.2 En el caso que el servidor web no esté disponible la aplicación presentará el mensaje "Servidor no disponible en estos momentos", en la pantalla <b>[Iniciar Sesión]</b> .	

### 3.1.3.3 CU003: Consultar Especie Vegetal

<b>Caso de Uso:</b>	Consultar Especie Vegetal	<b>Actor:</b>	Usuario
<b>Código:</b>	CU003	<b>Tipo:</b>	Primario-Sistema
<b>Ref. Req.</b>	RF004		
<b>Resumen:</b>	El usuario consulta todas las especies vegetales registradas en la aplicación, para filtrarlas acorde a su criterio.		
<b>Objetivos:</b>	Consultar todas las especies vegetales registradas en la aplicación móvil.		
<b>Precondición</b>	Que se haya realizado el caso de uso <b>Iniciar Sesión</b> .		
<b>Post condición</b>	EL usuario podrá observar un listado de todas las especies vegetales registradas en la aplicación.		
Curso normal de eventos			
1. La aplicación presenta la pantalla <b>[Principal]</b> .			
2. El usuario presiona el botón <b>[Consultar]</b> en la pantalla <b>[Principal]</b> .			
3. La aplicación presenta la pantalla <b>[Búsqueda de Especies Vegetales]</b> .			

4. La aplicación busca la información en el servidor web y muestra una lista de las especies vegetales registradas en la aplicación, para que el usuario elija la que desea consultar.
5. Para ver los detalles de la especie vegetal el usuario selecciona de la lista el ítem que desea consultar.
6. La aplicación muestra la pantalla <b>[Información detallada de Especie Vegetal]</b> .
7. El caso de uso finaliza.
<b>Cursos alternos de eventos</b>
<b>A. Criterio de búsqueda de especies vegetales.</b>
A.1 La aplicación filtra la búsqueda acorde al nombre común o científico de la especie vegetal, en la pantalla <b>[Búsqueda de Especies Vegetales]</b> .
<b>B. Servicio no disponible</b>
B.1 En el caso que no existe una conexión a internet la aplicación presentará el mensaje "Conexión a internet no disponible", en la pantalla <b>[Búsqueda de Especies Vegetales]</b> .
B.2 En el caso que el servidor web no esté disponible la aplicación presentará el mensaje "Servidor no disponible en estos momentos", en la pantalla <b>[Búsqueda de Especies Vegetales]</b> .

### 3.1.3.4 CU004: Identificar Especie Vegetal

<b>Caso de Uso:</b>	Identificar Especie Vegetal	<b>Actor:</b>	Usuario
<b>Código:</b>	CU004	<b>Tipo:</b>	Primario-Sistema
		<b>Pantallas:</b>	P003-P004-P005-P006
<b>Ref. Req.</b>	RF005		
<b>Resumen:</b>	El usuario captura una fotografía de la hoja de la especie vegetal a identificar con un fondo blanco, para su reconocimiento.		
<b>Objetivos:</b>	Identificar una especie vegetal por medio de una hoja de la misma.		
<b>Precondición</b>	Que se haya realizado el caso de uso <b>Iniciar Sesión</b> .		
<b>Post condición</b>	La aplicación muestra como resultado un listado de 5 especies vegetales, las más acertadas de la búsqueda.		
<b>Curso normal de eventos</b>			
1. La aplicación presenta la pantalla <b>[Principal]</b> .			
2. El usuario presiona el botón <b>[Identificar]</b> en la pantalla <b>[Principal]</b> .			
3. La aplicación muestra la cámara del dispositivo móvil.			

4. EL usuario captura la hoja de la especie vegetal a identificar con un fondo blanco.
5. La aplicación muestra la pantalla <b>[Identificación de Especie Vegetal]</b> con la imagen de la hoja capturada.
6. El usuario selecciona el tipo de hoja simple o compuesta de la caja de opciones <b>[Tipo de Hoja]</b> .
7. El usuario presiona el botón <b>[Buscar Especie]</b> de la pantalla <b>[Identificación de Especie Vegetal]</b> .
8. La aplicación busca la información en el servidor web y muestra la pantalla <b>[Búsqueda de Especies Vegetales]</b> con una lista de 5 especies vegetales registradas en la aplicación, más cercanas a la búsqueda.
9. Para ver los detalles de la especie vegetal el usuario selecciona de la lista el ítem que desea consultar.
10. La aplicación muestra la pantalla <b>[Información detallada de Especie Vegetal]</b> .
11. El caso de uso finaliza.
<b>Cursos alternos de eventos</b>
<b>A. Hoja de especie vegetal no válida.</b>
A.1 El usuario presiona el botón <b>[Recapturar Hoja]</b> , por no cumplir con los requerimientos necesarios.
<b>B. Servicio no disponible</b>
B.1 En el caso que no existe una conexión a internet la aplicación presentará el mensaje "Conexión a internet no disponible", en la pantalla <b>[Identificación de Especie Vegetal]</b> .
B.2 En el caso que el servidor web no esté disponible la aplicación presentará el mensaje "Servidor no disponible en estos momentos", en la pantalla <b>[Identificación de Especie Vegetal]</b> .

### 3.1.3.5 CU005: Registrar Especie Vegetal

<b>Caso de Uso:</b>	Registrar Especie Vegetal	<b>Actor:</b>	Usuario
<b>Código:</b>	CU005	<b>Tipo:</b>	Primario-Sistema
		<b>Pantallas:</b>	P003-P005-P007
<b>Ref. Req.</b>	RF006		
<b>Resumen:</b>	El usuario captura una fotografía de la hoja de la especie vegetal a registrar, ingresa el nombre común y científico, y registra la especie vegetal en el inventario.		
<b>Objetivos:</b>	Registrar una especie vegetal por medio de una hoja de la misma.		

<b>Precondición</b>	Que se haya cumplido el caso de uso <b>Registrar usuario</b> , con perfil de usuario " <b>Colaborador</b> ".
	Que se haya realizado el caso de uso <b>Iniciar Sesión</b> .
<b>Post condición</b>	La aplicación muestra la especie vegetal registrada.
<b>Curso normal de eventos</b>	
1. La aplicación presenta la pantalla <b>[Principal]</b> .	
2. El usuario presiona el botón <b>[Nueva Especie]</b> en la pantalla <b>[Principal]</b> .	
3. La aplicación móvil muestra la pantalla <b>[Registro de Especie Vegetal]</b> , con los datos de geolocalización actual (latitud y longitud).	
4. EL usuario presiona el botón <b>[Capturar Hoja]</b> , de la pantalla <b>[Registro de Especie Vegetal]</b> .	
5. EL usuario captura la hoja de la especie vegetal a registrar con un fondo blanco.	
6. La aplicación muestra la pantalla <b>[Registro de Especie Vegetal]</b> con la imagen de la hoja capturada.	
7. El usuario selecciona el tipo de hoja simple o compuesta de la caja de opciones <b>[Tipo de Hoja]</b> .	
8. El usuario ingresa el nombre común y científico de la especie vegetal a registrar.	
9. El usuario presiona el botón <b>[Registrar]</b> de la pantalla <b>[Registro de Especie Vegetal]</b> .	
10. La aplicación valida y registra la información en el servidor web y muestra la pantalla <b>[Información detallada de la Especie Vegetal]</b> con la especie vegetal registrada en la aplicación móvil.	
11. Para ver los detalles de la especie vegetal el usuario selecciona de la lista el ítem que desea consultar.	
12. La aplicación muestra la pantalla <b>[Información detallada de Especie Vegetal]</b> .	
13. El caso de uso finaliza.	
<b>Cursos alternos de eventos</b>	
<b>A. Hoja de especie vegetal no válida.</b>	
A.1 El usuario presiona el botón <b>[Capturar Hoja]</b> , por no cumplir con los requerimientos necesarios.	
A.2 Al no capturar una imagen con la cámara del dispositivo, la aplicación muestra el mensaje "Imagen a registrar no válida", en la pantalla <b>[Registro de Especie Vegetal]</b> .	

<p><b>B. Servicio no disponible</b></p> <p>B.1 En el caso que no existe una conexión a internet al presionar el botón <b>[Registrar]</b> de la pantalla <b>[Registro de Especie Vegetal]</b>, la aplicación presentará el mensaje "Conexión a internet no disponible", en la pantalla <b>[Búsqueda de Especies Vegetales]</b>.</p> <p>B.2 En el caso que el servidor web no esté disponible, al presionar el botón <b>[Registrar]</b> de la pantalla <b>[Registro de Especie Vegetal]</b>, la aplicación presentará el mensaje "Servidor no disponible".</p>
<p><b>C. Campos obligatorios vacíos en registrar especie vegetal en la aplicación.</b></p> <p>C.1 La aplicación muestra el mensaje "Campos obligatorios vacíos", en la pantalla <b>[Registro de Especie Vegetal]</b>.</p>
<p><b>D. Datos de especie vegetal incorrectos.</b></p> <p>D.1 La aplicación muestra el mensaje "Nombre común y/o científico no válidos", en la pantalla <b>[Registro de Especie Vegetal]</b>.</p>

### 3.1.3.6 CU006: Verificar usuario en el servidor web

<b>Caso de Uso:</b>	Verificar usuario en el Servidor web.	<b>Actor:</b>	Servidor web
<b>Código:</b>	CU006	<b>Tipo:</b>	Primario-Sistema
		<b>Pantallas:</b>	X
<b>Ref. Req.</b>	RF007-RF008		
<b>Resumen:</b>	EL servidor responde a la solicitud de verificación de usuario emitida por la aplicación móvil.		
<b>Objetivos:</b>	Responder a la solicitud de verificación de usuario emitida por la aplicación móvil.		
<b>Precondición</b>	El servidor web recibe una solicitud de verificación de usuario.		
<b>Post condición</b>	Emisión de respuesta del servidor web.		
<b>Curso normal de eventos</b>			
1. El servidor web comprueba los datos del usuario, emitidos por la aplicación móvil.			
2. El servidor web emite una respuesta en formato JSON.			
3. El caso de uso finaliza.			



### 3.1.3.7 CU007: Registrar usuarios en el servidor web

<b>Caso de Uso:</b>	Registrar usuario en el Servidor web.	<b>Actor:</b>	Servidor web
<b>Código:</b>	CU007	<b>Tipo:</b>	Primario-Sistema
<b>Pantallas:</b>	X		
<b>Ref. Req.</b>	RF007-RF009		
<b>Resumen:</b>	EL servidor responde a la solicitud de registro de usuario emitida por la aplicación móvil.		
<b>Objetivos:</b>	Registrar usuario en la base de datos del Servidor web.		
<b>Precondición</b>	El servidor web recibe una solicitud de registro de usuario.		
<b>Post condición</b>	Emisión de respuesta del servidor web.		
<b>Curso normal de eventos</b>			
1. El servidor web comprueba los datos del usuario, emitidos por la aplicación móvil.			
2. El servidor almacena en la base de datos, los datos del usuario registrado.			
3. El servidor web emite una respuesta en formato JSON.			
4. El caso de uso finaliza.			
<b>Cursos alternos de eventos</b>			
<b>A. El servidor no puede registrar usuario.</b>			
A.1 El servidor web emite una respuesta en formato JSON.			

### 3.1.3.8 CU008: Consultar especie vegetal en el servidor web

<b>Caso de Uso:</b>	Consultar especie vegetal en el servidor web.	<b>Actor:</b>	Servidor web
<b>Código:</b>	CU008	<b>Tipo:</b>	Primario-Sistema
<b>Pantallas:</b>	X		
<b>Ref. Req.</b>	RF007-RF0010		
<b>Resumen:</b>	EL servidor responde a la solicitud de consulta de especies vegetales emitida por la aplicación móvil.		
<b>Objetivos:</b>	Consultar especies vegetales en la base de datos del Servidor web.		
<b>Precondición</b>	El servidor web recibe una solicitud de consulta de especies vegetales.		
<b>Post condición</b>	Emisión de respuesta del servidor web.		
<b>Curso normal de eventos</b>			
1. El servidor web comprueba la solicitud de consulta emitida por la aplicación móvil.			

2. El servidor consulta en la base de datos los datos todas las especies vegetales registradas.
3. El servidor web emite una respuesta en formato JSON.
4. El caso de uso finaliza.

### 3.1.3.9 CU009: Identificar especie vegetal en el servidor web

<b>Caso de Uso:</b>	Identificar especie vegetal en el servidor web.	<b>Actor:</b>	Servidor web
<b>Código:</b>	CU009	<b>Tipo:</b>	Primario-Sistema
<b>Pantallas:</b>	X		
<b>Ref. Req.</b>	RF007-RF0011		
<b>Resumen:</b>	EL servidor responde a la solicitud de identificación de especie vegetal emitida por la aplicación móvil.		
<b>Objetivos:</b>	Identificar especies vegetales de acuerdo a las registradas en la base de datos del Servidor web.		
<b>Precondición</b>	El servidor web recibe una solicitud de identificación de especie vegetal.		
<b>Post condición</b>	Emisión de respuesta del servidor web.		
<b>Curso normal de eventos</b>			
1. El servidor web comprueba los datos emitidos por la aplicación móvil.			
2. El servidor identifica de acuerdo a las especies vegetales registradas en base de datos, las 5 especies vegetales más cercanas a la búsqueda.			
3. El servidor web emite una respuesta en formato JSON.			
4. El caso de uso finaliza.			
<b>Cursos alternos de eventos</b>			
<b>A. El servidor web no se encuentra disponible.</b>			
A.1 El servidor web emite una respuesta en formato JSON.			

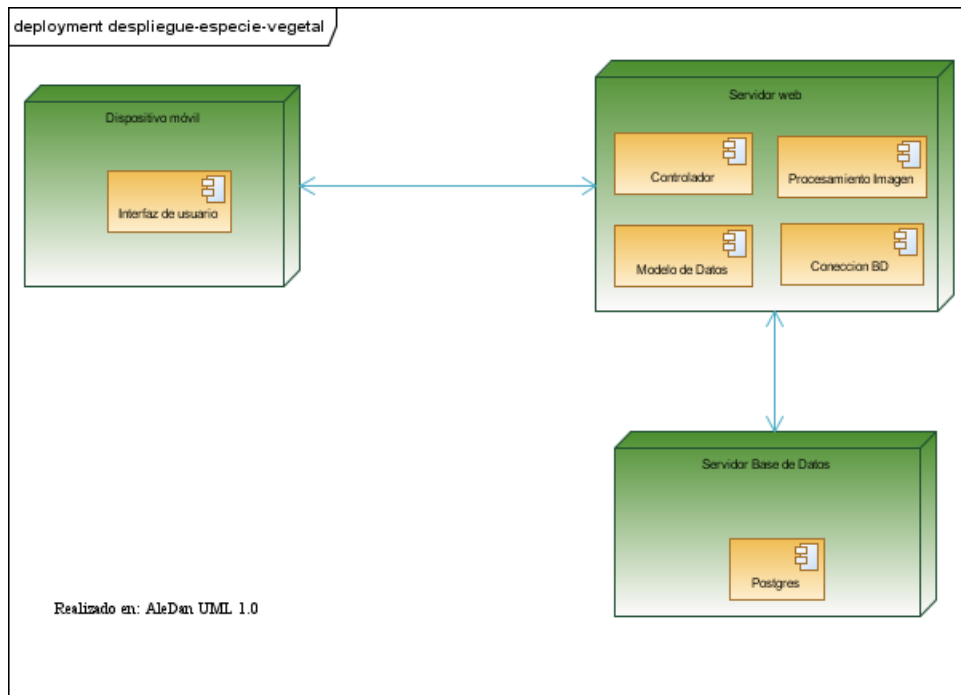
### 3.1.3.10 CU0010: Registrar especie vegetal en el servidor web

<b>Caso de Uso:</b>	Registrar especie vegetal en el servidor web.	<b>Actor:</b>	Servidor web
<b>Código:</b>	CU0010	<b>Tipo:</b>	Primario-Sistema
<b>Pantallas:</b>	X		
<b>Ref. Req.</b>	RF007-RF0012		
<b>Resumen:</b>	EL servidor responde a la solicitud de registro de una especie vegetal emitida por la aplicación móvil.		

<b>Objetivos:</b>	Registrar especies vegetales en la base de datos del Servidor web.
<b>Precondición</b>	El servidor web recibe una solicitud de registro de especie vegetal, emitida por la aplicación móvil.
<b>Post condición</b>	Emisión de respuesta del servidor web.
<b>Curso normal de eventos</b>	
1. El servidor web comprueba los datos emitidos por la aplicación móvil.	
2. El servidor registra la especie vegetal en la base de datos del servidor web.	
3. El servidor web emite una respuesta en formato JSON.	
4. El caso de uso finaliza.	
<b>Cursos alternos de eventos</b>	
<b>A. El servidor web no se encuentra disponible.</b>	
A.1 El servidor web emite una respuesta en formato JSON.	
<b>B. Registro de ejemplar de especie existente</b>	
B.1 El servidor web almacena como ejemplar de una especie vegetal ya registrada en la base de datos, y web emite una respuesta en formato JSON.	

### 3.1.5 Arquitectura de la aplicación

Luego de haber analizado las arquitecturas posibles para el desarrollo del presente PFC, se ha optado por la arquitectura en la cual el dispositivo móvil se utiliza para obtener las imágenes y desplegar los resultados, la autenticación de usuarios, la identificación y registro de especies vegetales se realiza en el servidor y el almacenamiento de la información de usuarios y especies vegetales se ejecuta en la base de datos como se muestra en la figura 8, debido a que se pretende que la solución al presente problema sea compatible con el mayor número de dispositivos móviles utilizados en el presente campo de estudio. Debido a que no todos los dispositivos móviles disponibles tienen la misma capacidad y velocidad de procesamiento de la información para desarrollar el reconocimiento localmente, se lo realizó implementando un servidor web que realice la identificación de la imagen y devuelve el resultado en el menor tiempo posible.



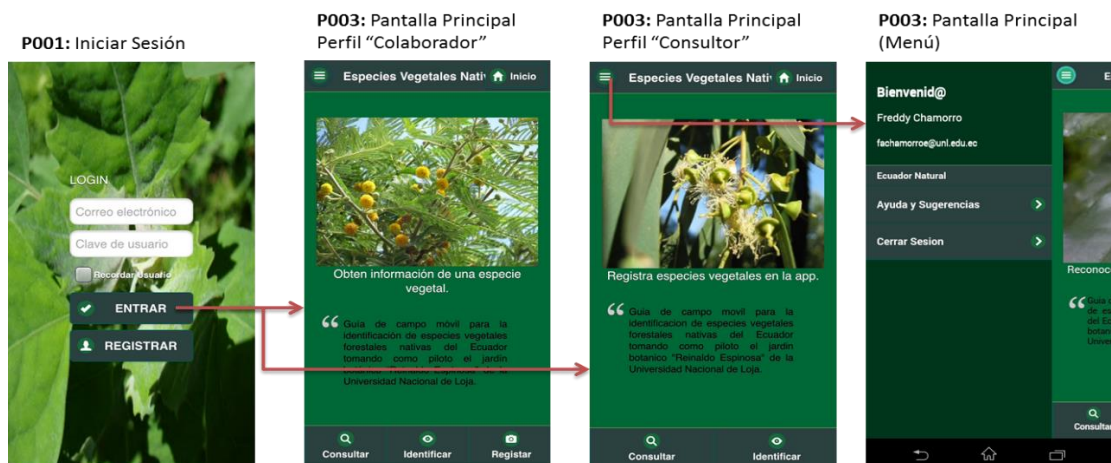
**Figura 8:** Arquitectura de despliegue de la aplicación

### 3.1.5 Interfaz gráfica de usuario

En este apartado se muestra las interfaces de la aplicación móvil desarrollada que servirá a los usuarios como guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinosa” de la Universidad Nacional de Loja.

#### 3.1.5.1 Iniciar Sesión

La pantalla P001 muestra la ejecución de la aplicación y presenta las opciones [Entrar] y [Registrar]. El usuario ingresa con sus respectivas credenciales y visualice la pantalla 002 con las opciones disponibles acorde a su perfil de usuario (Ver figura 9).



**Figura 9:** Interfaz para ingresar a la aplicación.

### 3.1.5.2 Registrar Usuario

La interfaz que se presenta en la figura 10 permite el registro de usuarios en la aplicación móvil.



Figura 10: Interfaz de registro de usuario en la aplicación.

### 3.1.5.3 Buscar Especie Vegetal

La interfaz que se presenta en la figura 11, permite buscar todas las especies vegetales registradas en la aplicación móvil.

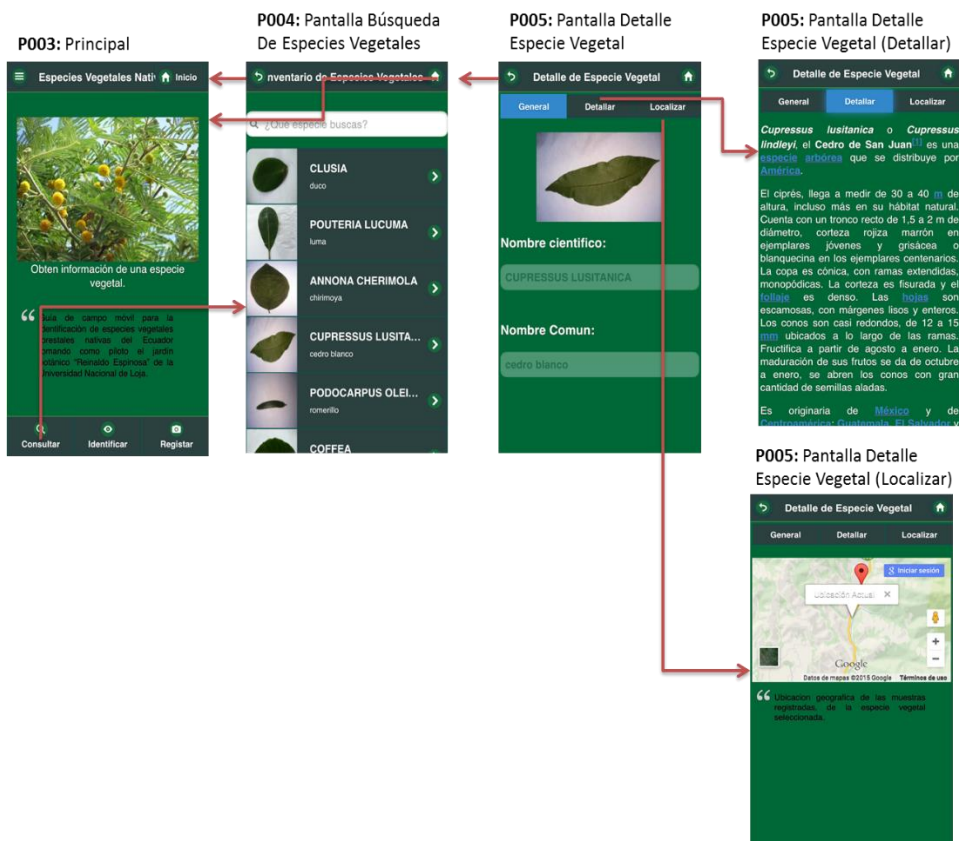


Figura 11: Interfaz de búsqueda de especies vegetales

### 3.1.5.4 Identificar Especies Vegetales

La interfaz que se presenta en la figura 12, hace uso de la cámara del dispositivo móvil para capturar la hoja de la especie vegetal con un fondo blanco, para luego presionando el botón [Buscar Especie] mostrar un listado de las 5 especies más cercadas a la búsqueda obteniendo la información detallada de la misma y la localización en el mapa.



Figura 12: Interfaz de identificación de especies vegetales

### 3.1.5.5 Registro de Especies Vegetales

La interfaz que se presenta en la figura 13, está habilitada solo para usuarios registrados con un perfil "Colaborador", que tienen los conocimientos necesarios para registrar las especies vegetales, para el registro de las mismas el dispositivo móvil hace uso de la geolocalización y la cámara para la captura de la hoja de la especie vegetal con un fondo blanco. El usuario debe conocer el nombre común, científico, y tipo de hoja de la especie para poder registrarla.

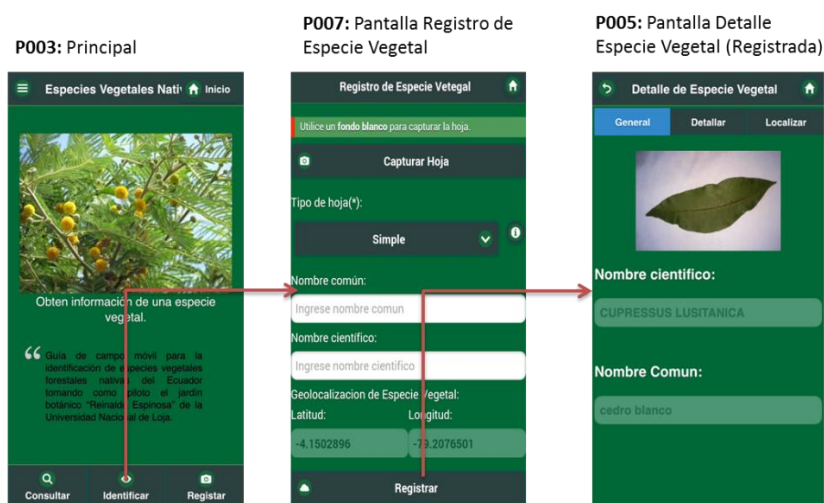
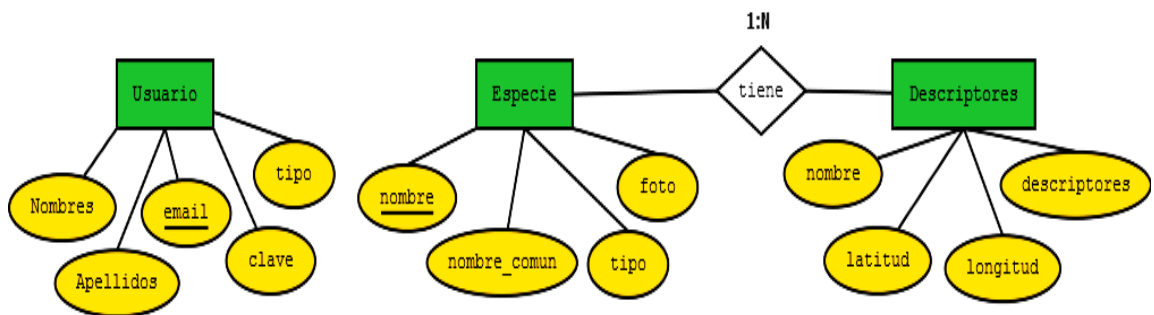


Figura 13: Interfaz de Registro de Especies Vegetales

### 3.1.6 Modelo de datos

El modelo de datos elaborado para guardar la información de las especies vegetales permitirá registrar por cada planta: nombre, nombre común, imagen de una hoja de la planta, la latitud y longitud, donde se registra una especie, los descriptores que identifican el borde de la hoja. Todos estos datos son únicos a cada planta, excepto por los descriptores, que se toman desde varias muestras. La entidad usuario, que permite el registro y autenticación de los usuarios, con la finalidad de restringir las opciones de acceso, de acuerdo al perfil. El modelo entidad relación se muestra en la Figura 14.



**Figura 14:** Diagrama ER del modelo de datos

### 3.1.7 Análisis del servidor web

El servidor web a implementar debe contar con los métodos descritos en la tabla XVI, que permitirán dar respuesta en formato JSON, a las peticiones del usuario por medio de la aplicación móvil.

**TABLA XVI. MÉTODOS A IMPLEMENTAR EN EL SERVIDOR WEB PARA EL RECONOCIMIENTO DE ESPECIES VEGETALES**

Nombre del Método	Parámetros que recibe	Datos que retorna
verificarUsuario	<ul style="list-style-type: none"> <li>Correo</li> <li>Clave</li> </ul>	Retorna los siguientes datos del usuario en formato JSON: <ul style="list-style-type: none"> <li>usuarioV [nombres, apellidos, perfil]</li> </ul>
registrarUsuario	<ul style="list-style-type: none"> <li>Nombres</li> <li>Apellidos</li> <li>Perfil</li> <li>Clave</li> <li>Email</li> </ul>	Retorna los siguientes datos en formato JSON: <ul style="list-style-type: none"> <li>usuarioR[estadou]</li> </ul>

buscarTodos		Retorna los siguientes datos en formato JSON: <ul style="list-style-type: none"> <li>plantas [nombre, nombre_comun, foto]</li> </ul>
buscarPuntosInteres	<ul style="list-style-type: none"> <li>especievegetal</li> </ul>	Retorna los siguientes datos en formato JSON: <ul style="list-style-type: none"> <li>planta [nombre, latitud, longitud ]</li> </ul>
buscarEspecieFoto	<ul style="list-style-type: none"> <li>especie, tipohoja</li> </ul>	Retorna los siguientes datos en formato JSON: <ul style="list-style-type: none"> <li>planta[nombre, comun, photo]</li> </ul>
registrarEspecie	<ul style="list-style-type: none"> <li>comun</li> <li>científico</li> <li>latitud</li> <li>longitud</li> <li>imagen</li> <li>tipohoja</li> </ul>	Retorna los siguientes datos en formato JSON: <ul style="list-style-type: none"> <li>planta[nombre, comun, photo]</li> </ul>

### 3.2 Planificación del proyecto

Luego de haber analizado los requerimientos de la aplicación a implementar en el presente PFC, se procedió a establecer las iteraciones y tiempos, en que se desarrollarán, en la tabla XVII se establece los tiempos para cada iteración.

**TABLA XVII. ITERACIONES (SPRINTS) DEL PROYECTO**

Id.	Fecha Inicio	Fecha Fin	Días	Meta	Requerimiento
1	19-01-2015	06-02-2015	15	Autenticación en la aplicación móvil	R001-RF002-RF003-RF007-RF008-RF009
2	09-02-2015	20-03-2015	30	Registrar especies vegetales por medio de una hoja	RF006-RF0012



3	23-03-2015	24-04-2015	25	Identificación de especies vegetales por medio de la hoja	FR005-RF0011
4	27-04-2015	08-05-2015	10	Consulta de todas las especies vegetales.	RF004-RF0010
5	11-05-2015	29-05-2015	15	Implementación de un servidor web online para respuesta a peticiones de aplicación móvil.	RF008-RF009- RF0010- RF0011- RF0012
6	01-06-2015	03-06-2015	3	Compilar la aplicación para el sistema operativo Android e iOS.	
7	04-06-2015	08-06-2015	3	Publicar la aplicación en las tiendas de Android, e iOS.	

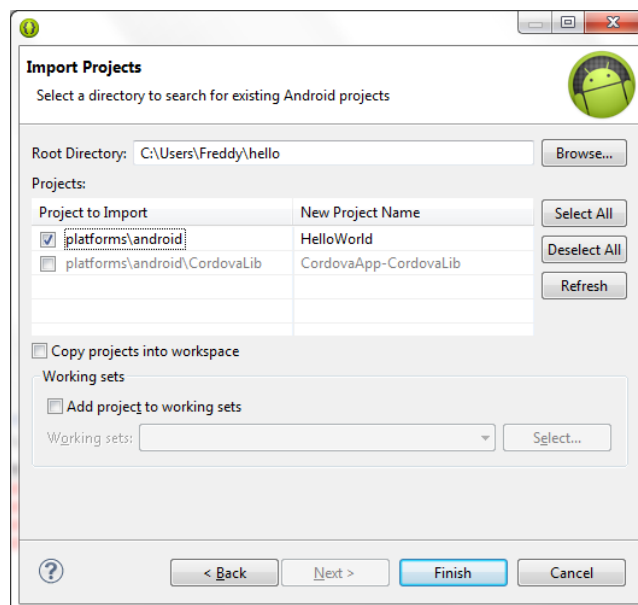
### 3.3 Iteración 1: Autenticación en la aplicación móvil

En esta iteración se realizaron algunas actividades para cumplir con la meta establecida, primero se creó la estructura base de la aplicación móvil, posteriormente se realizó la implementación del servidor web con la base de datos de manera local para el registro de usuarios, se desarrolló las funciones necesarias para la conexión del servidor web con la aplicación móvil, para continuar se implementó los requerimientos establecidos en la planificación, y finalmente se realizó las pruebas funcionales de la aplicación móvil.

#### 3.3.1 Creación de un proyecto con Apache Cordova

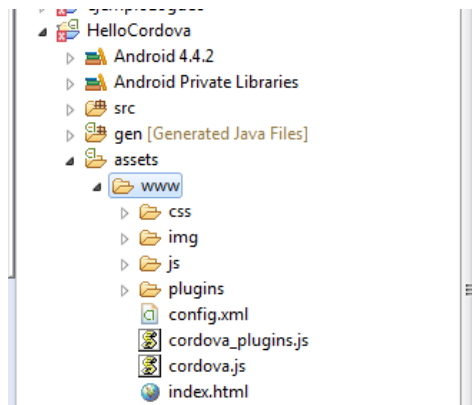
1. Descargar e instalar **Node.js**.
2. Descargar e instalar un cliente de **git**, si usted no tiene uno.
3. Instale el **cordova** desde la línea de comando (cmd o terminal). Ejecute: `sudo npm install -g cordova`

4. Crear una aplicación en un directorio específico con la línea de comandos: `$ cordova create hello com.example.hello HelloWorld`
5. Ingrese a la carpeta de la aplicación creada de forma automática, desde la línea de comandos: `$ cd hello`
6. Añadir las plataformas de desarrollo a utilizar:
  - Android:** `$ cordova platform add ios`
  - iOS:** `$ cordova platform add android`
7. Añadir los plugins a utilizar, de acuerdo a las funciones que se quiere utilizar del dispositivo móvil en la aplicación:
  - Cámara del dispositivo:** `$ cordova plugin add org.apache.cordova.camera`
  - Geolocalización:** `$ cordova plugin add org.apache.cordova.geolocation`
8. Para modificar y depurar la aplicación Android se ha utilizado la herramienta eclipse, importar el directorio android de la aplicación creada por línea de comandos (ver Figura 15).



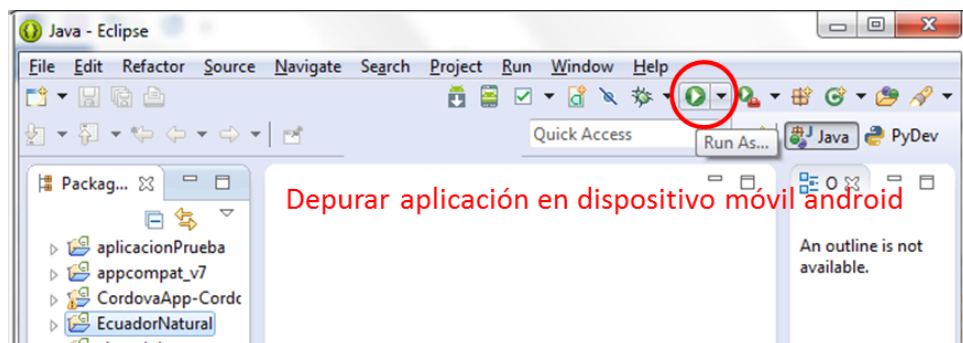
**Figura 15:** Importar proyecto Android desde directorio Apache Cordova.

9. En el directorio **assets** del proyecto verificar la existencia del directorio **www** (ver figura 16), en el cual se implementará la aplicación móvil.



**Figura 16:** Directorio www, sitio de trabajo de la aplicación móvil.

10. Añadir en el directorio **www** los archivos .js y .css de los Frameworks Front- End en los respectivos subdirectorios creados.
11. Después de haber configurado el directorio de trabajo, se procedió a implementar la aplicación móvil en el directorio **www**.
12. Para depurar la aplicación móvil en un dispositivo real o emulado, se debe hacer clic en el botón que se indica en la figura 17.



**Figura 17:** Depurar la aplicación móvil en un dispositivo android

### 3.3.2 Implementar el servidor local y los métodos para la Autenticación en la aplicación móvil

Como primer paso se ha descargado e implementado el framework tornado de python, se ha creado el archivo main.py, y se ha codificado el código siguiente que provee de los métodos para la comunicación del servidor web con la aplicación móvil. A continuación se muestra el código para la implementación del servidor web de manera local, con los métodos de registro y verificación de usuario. Es necesario mencionar que la respuesta que emite el servidor web a la aplicación móvil es en formato JSON.

```
from tornado.ioloop import IOLoop
from tornado.web import RequestHandler, Application, url, asynchronous
from tornado import ioloop,web
```

```

from controlador.ControladorPlantas import *
from controlador.ControladorUsuario import *
class verificaUsuario(RequestHandler):
    def post(self):
        correo=self.get_argument("correo","")
        clave=self.get_argument("clave","")
        usuarioV=buscarUsuario(correo,clave)
        self.write({'usuarioV':usuarioV})
class registraUsuario(RequestHandler):
    def post(self):
        nombre=self.get_argument("nombre","")
        apellido=self.get_argument("apellido","")
        perfil=self.get_argument("perfil","")
        clave=self.get_argument("clave","")
        correo=self.get_argument("email","")
        usuarioR=registrarUsuario(nombre,apellido,perfil,correo,clave)
        self.write({'usuarioR':usuarioR})
def make_app():
    return Application([
        url(r"/verificaUsuario", verificaUsuario),
        url(r"/registraUsuario", registraUsuario),  ])
def main():
    app = make_app()
    app.listen(8081,address='X.X.X.X')
    IOLoop.instance().start()
if __name__ == "__main__":
    main()

```

Los controladores, conexión con la base de datos y clase utilizada se encuentran implementados en el Anexo 1, sección 1: Servidor web.

Del lado del cliente para la comunicación de la aplicación móvil con el servidor web se ha creado el archivo usuarios.js, en el que se encuentran implementados los métodos de registro y verificación de usuarios, en el siguiente método se muestra un ejemplo de la comunicación de la aplicación móvil con el servidor para el registro de usuarios:

```

function registrarUsuarioBD(rnombre,rapellido,rclave,rcorreo,rperfil){
$.mobile.loading( 'show', {
    text: 'Registrando..',
    textVisible: true,
    theme: 'a',
    html: "" });
    archivoValidacion="http://servidorpy-
reconocimientoev.rhcloud.com/registraUsuario";
$.ajax({
    type    : "POST",
    url     : archivoValidacion,
    crossDomain: true,
    data    : {'nombre' : rnombre, 'apellido':rapellido,'email': rcorreo, 'clave':rclave,
'perfil':rperfil},
    dataType : 'json',
    success  : function(response) {
        var obj = $.parseJSON(response.usuarioR);
        $.mobile.loading( "hide" );
        $.each(obj, function(index, element){
            if (element.registrado==0) {
                toast(element.estadou);
                return false;
            }else{
                toast1(element.estadou);
                liberarCamposRegistro();
                $.mobile.changePage("#entrar");
            } });
        return false; },
    error    : function(data) {
        toast('Servidor no disponible en estos momentos');
        $.mobile.loading( "hide" );
        return false;
    } });
    return false; }

```

Para observar el método de verificación de usuario ver el anexo 1, sección 3: Aplicación móvil. Explicando la implementación del método expuesto se tiene:

El método predefinido **\$.mobile.loading** permite mostrar al usuario un mensaje de espera mientras se procesa la peritación, la variable **archivoValidacion**, contiene la dirección HTTP, para el envío de parámetros y la petición al servidor web.

Dentro de la función \$.ajax(), se utiliza algunos parámetros como:

**type:** Se utiliza el método POST, para enviar los datos al servidor.

**url:** Dirección http para el envío de datos.

**crossDomain:** Valor booleano true, por solicitud a dominio externo.

**data:** Los datos enviados en formato json al servidor.

**dataType:** El tipo de datos enviado al servidor, valor 'json',

**success:** Función que recibe la respuesta del servidor.

**error:** Función que se ejecuta en caso de haber error en envío de datos al servidor o cuando el servidor web no se encuentra disponible.

### 3.4 Iteración 2: Registrar especies vegetales por medio de una hoja

Para implementar los requisitos relacionados al registro de especies vegetales tanto en la aplicación móvil como en el servidor web, se ha creado un archivo especievegetal.js, camara.js, y geolocation.js, que contiene los métodos de comunicación de la aplicación con el servidor y se ha añadido los métodos correspondientes, para el procesamiento de la solicitud y la respuesta al usuario.

**TABLA XVIII. MÉTODOS IMPLEMENTADOS (ITERACIÓN 2)**

Método implementado aplicación móvil	Método implementado servidor web
subirImagenRegistrarImgur()	X
subirEspecieServidorRegistro(sfoto)	registraEspecie()
capturarImagenRegistrar()	X
iniciarLocalizacion()	X

En el anexo 2, se encuentra la implementación de los métodos más importantes de los especificados en la tabla XVIII, además se muestra el método para obtener los descriptores de una hoja de la especie vegetal y la estructura de las tablas para el almacenamiento de las mismas.

Una vez establecidos los métodos para el registro de especies vegetales se procedió a realizar la población de la base de datos con las especies vegetales forestales nativas más significativas localizadas en el jardín botánico “Reinaldo Espinosa” de la UNL.

### 3.5 Iteración 3: Identificación de especies vegetales por medio de la hoja

Para implementar los requerimientos relacionados a la identificación de especies vegetales se ha desarrollado los respectivos métodos, tanto en la aplicación móvil como en el servidor web, que le permiten visualizar una respuesta al usuario. Para lo cual se ha implementado los métodos dentro de los archivos camara.js y especievegetal.js. Además se ha implementado en el servidor web los métodos necesarios para la identificación de especies vegetales. En la tabla XIX se listan los métodos implementados:

**TABLA XIX. MÉTODOS IMPLEMENTADOS (ITERACIÓN 3)**

Método implementado aplicación móvil	Método implementado servidor web
capturarImagenIdentificar()	X
subirEspecieServidor(sfoto)	buscarEspecieFot()
subirImagenIdentificarImgur()	X

En el anexo 3, se muestra la implementación de los métodos de la tabla XX, para la identificación de especies vegetales en la aplicación móvil y en el servidor web.

### 3.6 Iteración 4: Consulta de todas las especies vegetales.

Para la implementación de los requerimientos relacionados a la consulta de especies vegetales se ha implementado algunos métodos en la aplicación y en servidor web, con la finalidad de mostrar al usuario un listado de todas las especies vegetales y la información detallada de cada una, así también la localización en el mapa de cada muestra registrada según la especie. A continuación en la tabla XX, se muestra los métodos implementados:

**TABLA XX. MÉTODOS IMPLEMENTADOS (ITERACIÓN 4)**

Método implementado aplicación móvil	Método implementado servidor web
buscarTodas()	buscarTodos()
cargarMapa()	buscarPuntosInteres()
cargarResultados()	X
cargarDatosWikipedia()	X

En el anexo 4, se muestra de manera detallada la implementación de cada método del servidor anexo 4, sección 1, y de la aplicación móvil en el anexo 4, sección 2.

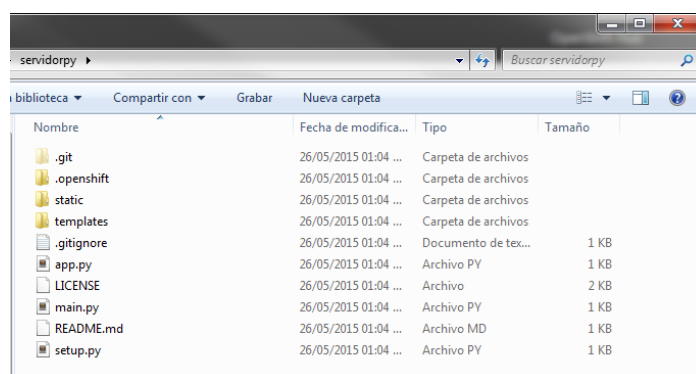
### 3.7 Iteración 5: Implementación de un servidor web online para respuesta a peticiones de aplicación móvil.

Una vez implementada todas las funcionalidades del servidor web en una red local, para la comunicación con la aplicación móvil, se ha procedido a la implementación del servidor web en el proveedor de alojamiento Openshift con una cuenta gratuita. Para subir el servidor web a la plataforma openshift, se lo hizo mediante acceso ssh, y se siguió los pasos descritos a continuación:

1. Se debe instalar la herramienta Ruby en caso de no tenerla.
2. Instalar un cliente git, en caso de no tenerlo.
3. Instalar la gema rhc de ruby para permitir al acceso a la plataforma openshift.
4. Configurar el cliente ssh, con los datos registrados en la página oficial de openshift para obtener un token de autorización de acceso.
5. Una vez configurado el entorno, se debe crear un servicio acorde a las necesidades, para el presente proyecto se ha hecho uso de un repositorio GIT, que tiene la base del servidor web en tornado a implementar, la línea de comandos para la creación del sitio de alojamiento sería:

```
rhc create-app servidorpy python-2.7 from-code  
https://github.com/avinassh/openshift-tornado-starter.git
```

6. Una vez creado el sitio de trabajo para el alojamiento del servidor web, en este caso con el nombre **servidorpy**, se crea una copia del entorno de trabajo, en el directorio actual de la línea de comando como se muestra en la figura 18:



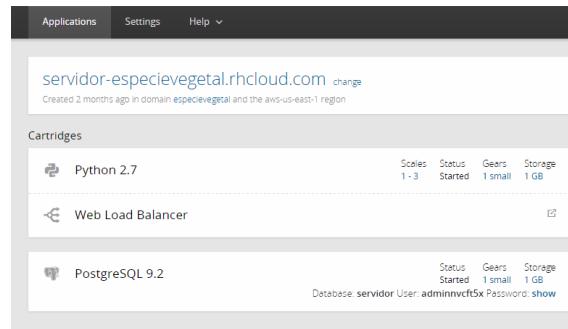
**Figura 18:** Directorio clonado desde repositorio Git (lugar de trabajo)

7. Se añadió la base de datos en Postgres mediante línea de comandos:

```
rhc cartridge add postgresql-9.2 -a servidorpy
```



8. Para observar que se han creado el repositorio de nuestro servidor correctamente se puede ingresar en la página oficial de openshift ([www.openshift.redhat.com](http://www.openshift.redhat.com)), con nuestro usuario y contraseña registrada, y observar el repositorio creado, como se muestra en la figura 19:

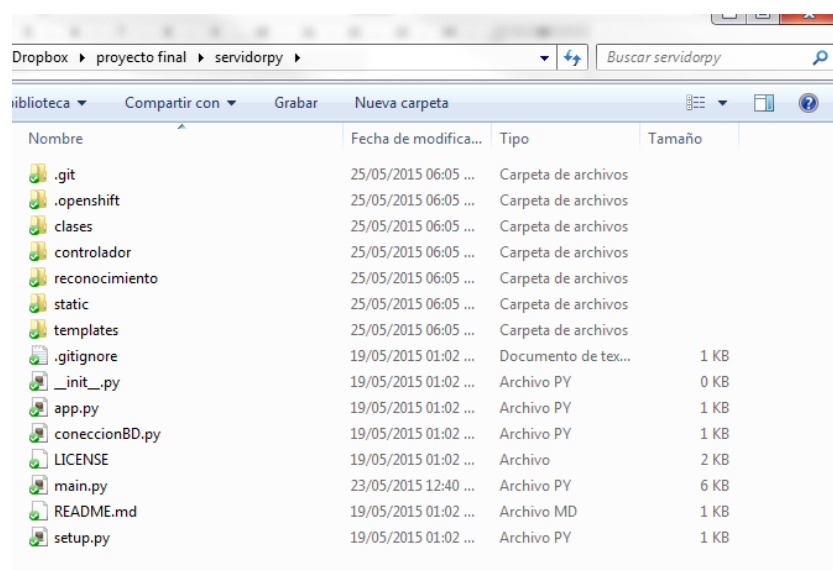


**Figura 19:** Cuenta creada en openshift con repositorio ejemplo creado desde línea de comandos

9. Una vez implementadas las herramientas se accede mediante línea de comandos al sitio de trabajo para la configuración de la base de datos, haciendo uso de la siguiente línea de comandos:

```
rhc ssh servidorpy
```

10. Posterior a la configuración de la base de datos de acuerdo a nuestras necesidades con las tablas y funciones requeridas, se procedió a la adaptación del código del servidor local, en el repositorio clonado en nuestro equipo, del paso 6, de lo cual obtuvimos el directorio modificado que se observa en la figura 20:

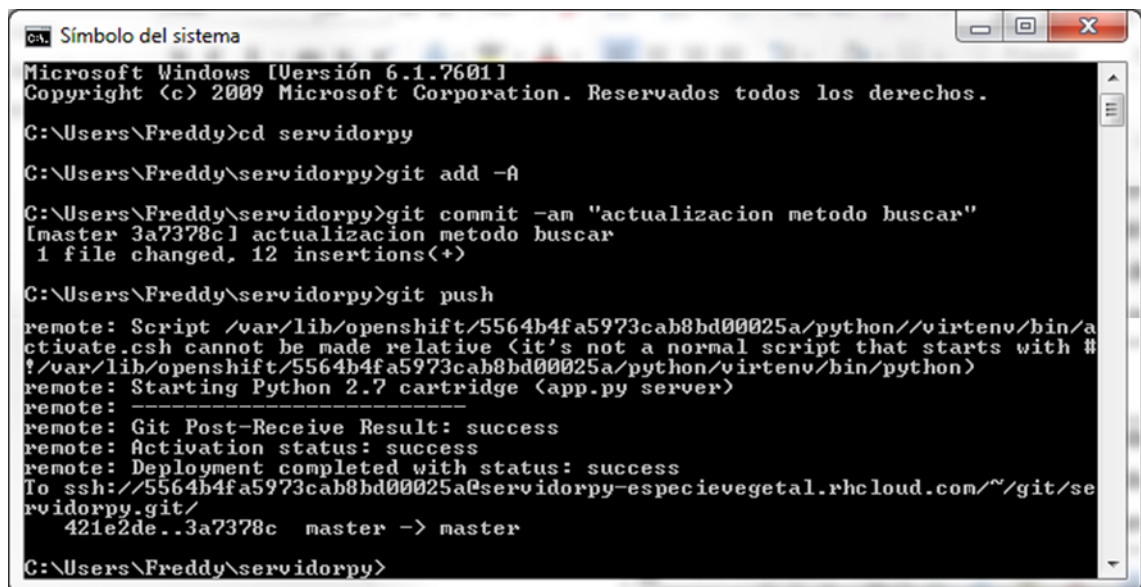


**Figura 20:** Repositorio clonado implementado en servidor local

11. Luego de haber implementado el código del servidor web lo subimos a la plataforma openshift haciendo uso de la línea de comandos, nos ubicamos en el directorio clonado y utilizamos los siguientes comandos:

```
git add -A
git commit -am "msmje"
git push
```

En la figura 21, se muestra el proceso de subida a openshift de las modificaciones realizadas al servidor por defecto. Los comandos antes mencionados se pueden utilizar cada vez que se realice una modificación al servidor web y se necesite actualizarlo.



```
ca Símbolo del sistema
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Freddy>cd servidorpy

C:\Users\Freddy\servidorpy>git add -A

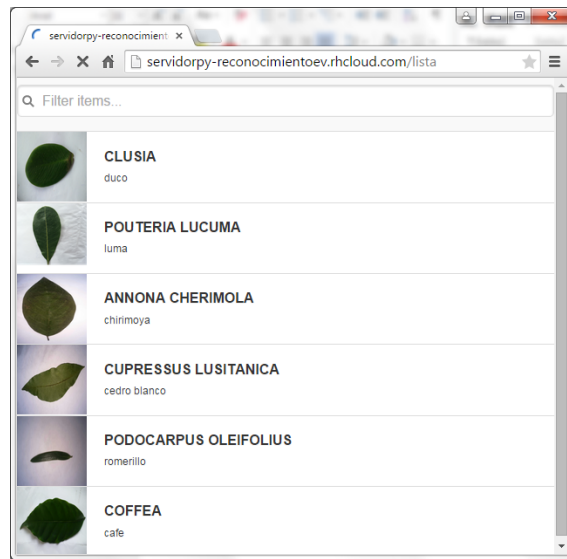
C:\Users\Freddy\servidorpy>git commit -am "actualizacion metodo buscar"
[master 3a7378c] actualizacion metodo buscar
1 file changed, 12 insertions(+)
```

```
C:\Users\Freddy\servidorpy>git push
remote: Script /var/lib/openshift/5564b4fa5973cab8bd00025a/python/virtenv/bin/activate.csh cannot be made relative (it's not a normal script that starts with #!/var/lib/openshift/5564b4fa5973cab8bd00025a/python/virtenv/bin/python)
remote: Starting Python 2.7 cartridge (app.py server)
remote: -----
remote: Git Post-Receive Result: success
remote: Activation status: success
remote: Deployment completed with status: success
To ssh://5564b4fa5973cab8bd00025a@servidorpy-especievegetal.rhcloud.com:/git/servidorpy.git/
421e2de..3a7378c master -> master

C:\Users\Freddy\servidorpy>
```

**Figura 21:** Proceso de subida de actualizaciones del servidor web implementado

El servidor web para el presente PFC, se encuentra ubicado en la dirección <http://servidorpy-reconocimientoev.rhcloud.com/>, dirección a la cual se realizan las peticiones desde la aplicación móvil. Para observar todas las especies vegetales registradas en la base de datos se ha establecido un método y dirección accesible desde cualquier navegador web, en la figura 22, se puede observar la respuesta del servidor a la petición del navegador web:



**Figura 22:** Listado de especies vegetales registradas en la aplicación móvil

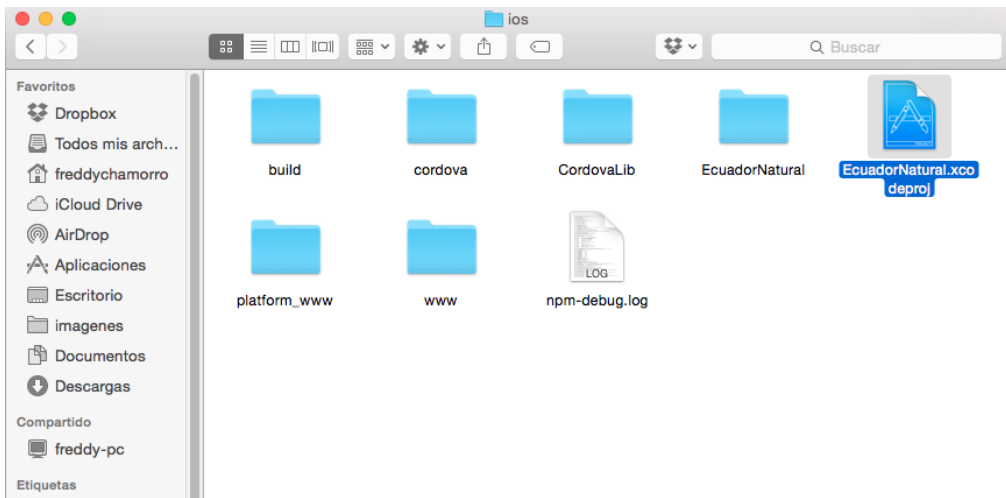
### **3.8 Iteración 6: Compilar la aplicación para el sistema operativo Android e iOS.**

La compilación de la aplicación móvil se la ha realizado para las plataformas Android e iOS, que son los sistemas operativos móviles con mayor demanda en el campo de estudio. Es necesario mencionar que la implementación de la aplicación móvil se la realizó en la plataforma Android por lo cual se en este apartado se procede a documentar el procedimiento solo para la compilación en iOS.

#### **3.8.1 Compilar la aplicación para iOS**

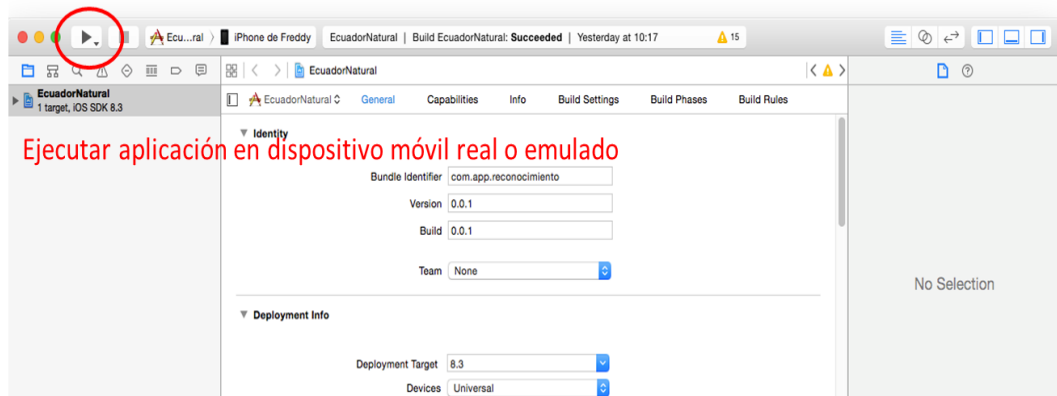
Para compilar la aplicación para iOS es necesario hacer uso de una computadora Mac e instalar las herramientas Xcode y Apache cordova. Luego de ello ejecutar los siguientes pasos:

1. Ingresar al directorio raíz de la aplicación creada en la iteración 1, sección 1 creación de un proyecto con apache cordova, ingresar al subdirectorío platform, se observará las plataformas de desarrollo añadidas desde la línea de comandos, en este caso se observará las plataformas Android e iOS, a continuación elegir ios y abrir el archivo con extensión .xcodeproj, con la herramienta Xcode, en la figura 23 se muestra el archivo que compete a la aplicación en desarrollo.



**Figura 23:** Directorio de la aplicación móvil para la plataforma ios.

2. Copiar los archivos implementados del subdirectorio www, de la carpeta assets de la plataforma Android con la cual se desarrolló la aplicación móvil. Pegar los archivos en el directorio www, de la plataforma ios.
3. Compilar la aplicación ios en la herramienta Xcode, en la pestaña Product utilizar la opción build.
4. Se debe depurar la aplicación en un dispositivo iPhone real, debido a que la aplicación móvil hace uso de recursos del dispositivo móvil, como la cámara, geolocalización, entre otros. Para depurar la aplicación en un dispositivo real se debe hacer clic en el icono indicado en la figura 24.



**Figura 24:** Depurar la aplicación iOS en un dispositivo real con Xcode

### 3.9 Iteración 7: Publicar la aplicación en las tiendas de Android, e iOS.

En la presente iteración se ha realizado la publicación de la Aplicación móvil en las respectivas tiendas dependiendo del sistema operativo móvil. La aplicación para Android ha sido publicada en Google Play, y para ios la aplicación se la ha publicado en App store.

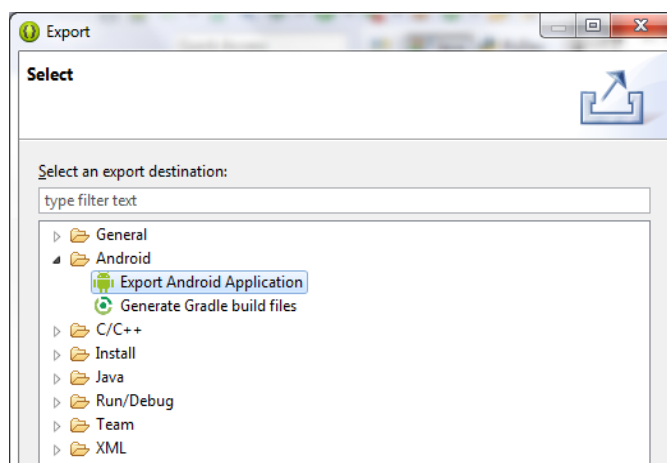
### 3.9.1 Publicar la aplicación en Google Play

Para publicar una aplicación en Google Play se ha realizado algunos pasos que se describirán a continuación:

#### 3.9.1.1. Exportar y firmar la aplicación.

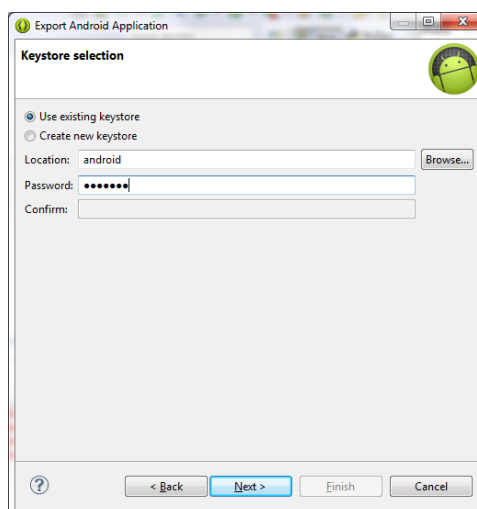
Luego de haber culminado el proceso de implementación se ha procedido a firmar la aplicación para poder publicarla en Google Play. Para firmar la aplicación se ha utilizado la herramienta Eclipse en la cual fue construida la aplicación.

Primeramente se ha realizado clic derecho sobre la aplicación y se ha seleccionado la opción exportar/Export Android Application, como se indica en la figura 25.



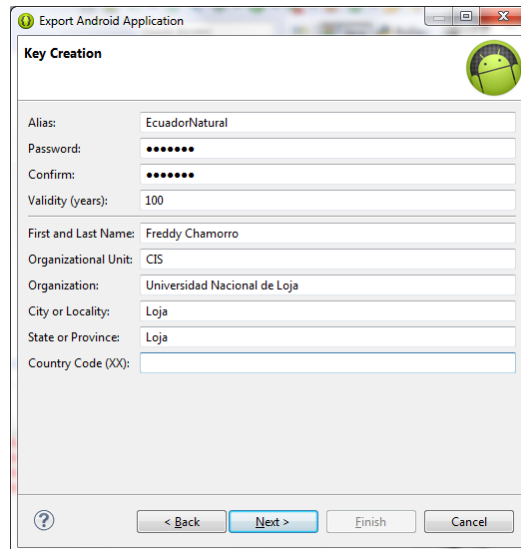
**Figura 25:** Exportar la aplicación android

Luego presionar el botón [Next] de la pantalla indicada en la figura 25, en la pantalla se indica la aplicación que se va a firmar, nuevamente se presiona el botón [Next], se ingresa los valores por defecto la localización android y la clave android, y se presiona el botón [Next], en la figura 26 se muestra lo expuesto:



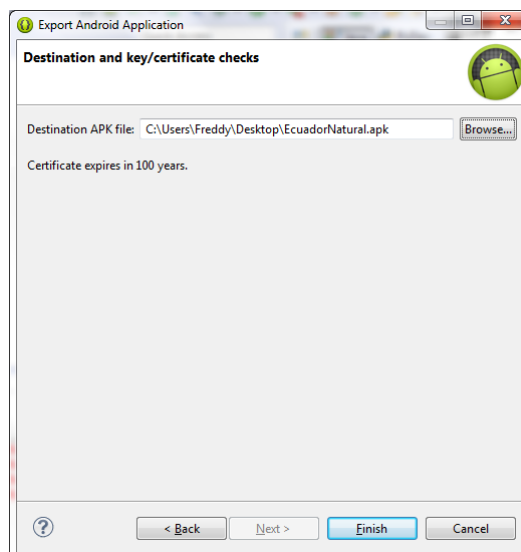
**Figura 26:** Firmar la aplicación con clave

A continuación se debe llenar algunos datos para completar el proceso de creación de la clave (Ver figura 27).



**Figura 27:** Datos para crear clave para android

Finalmente se debe seleccionar la dirección donde se ubicará el archivo **apk** firmada y presionar el botón [Finish] (Ver figura 28).



**Figura 28:** Seleccionar la ruta para el archivo apk firmado.

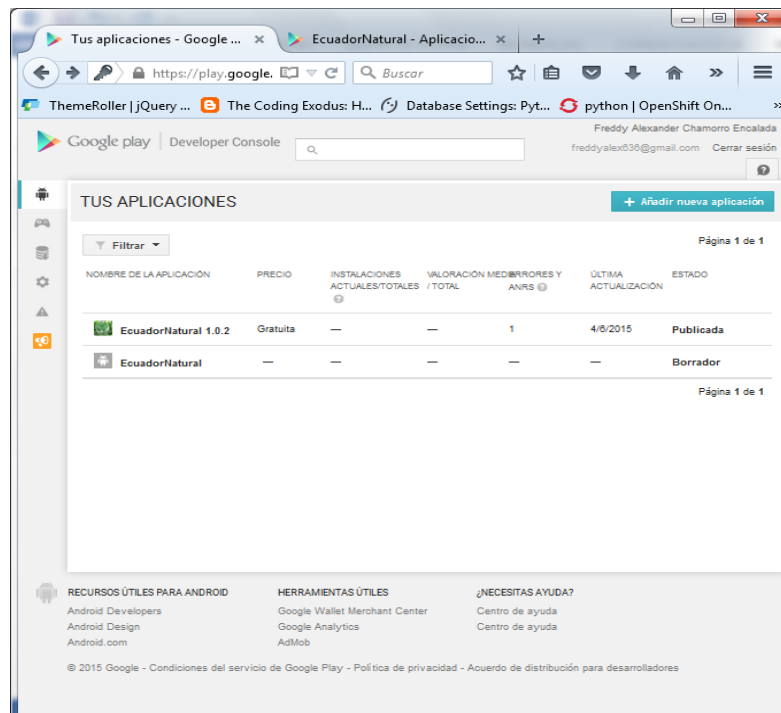
### 3.9.1.2 Acceder a la consola de Google play para desarrolladores.

Para acceder a la consola de Google Play se requiere tener una cuenta en gmail y acceder a la siguiente dirección <https://play.google.com/apps/publish/>, al ingresar se debe iniciar sesión con la cuenta de gmail, y se mostrará una pantalla como se indica en la figura 29.



**Figura 29:** Registrar una cuenta de desarrollador en Google Play.

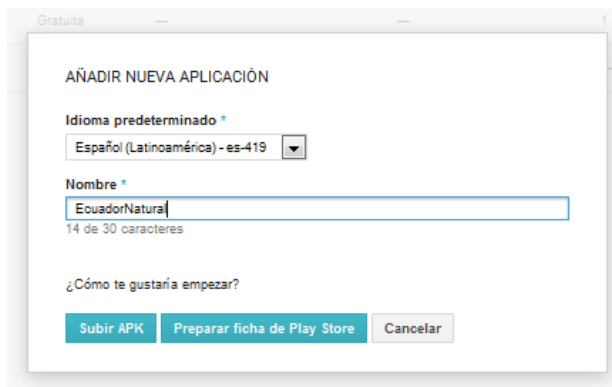
Posteriormente se debe aceptar las condiciones y proceder a llenar los datos de la tarjeta de crédito para realizar el pago correspondiente, luego se debe llenar algunos datos de la cuenta para completar el registro. Finalmente se mostrará una pantalla como se indica en la figura 30, en donde ya se puede proceder a publicar la aplicación.



**Figura 30:** Consola de Google Play

### 3.9.1.3 Publicar la aplicación

Para publicar la aplicación presionamos el botón [Añadir nueva aplicación] ubicado en la pantalla mostrada en la figura 30. A continuación se presentará una pantalla para ubicar el nombre de la aplicación (ver figura 31), y presionar el botón [Subir APK].



**Figura 31:** Añadir nueva aplicación Android

En la figura 32, se muestra la pantalla de la aplicación creada en donde procedemos a subir toda la información de nuestra aplicación.

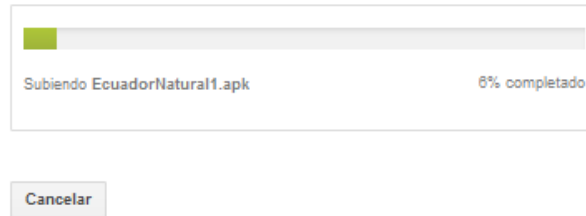


**Figura 32:** Entorno de aplicación móvil a subir

Se procede a presionar el botón [Subir tu primer archivo APK], y a continuación buscamos la aplicación a subir en nuestro ordenador y procedemos a subirla (ver Figura 33).

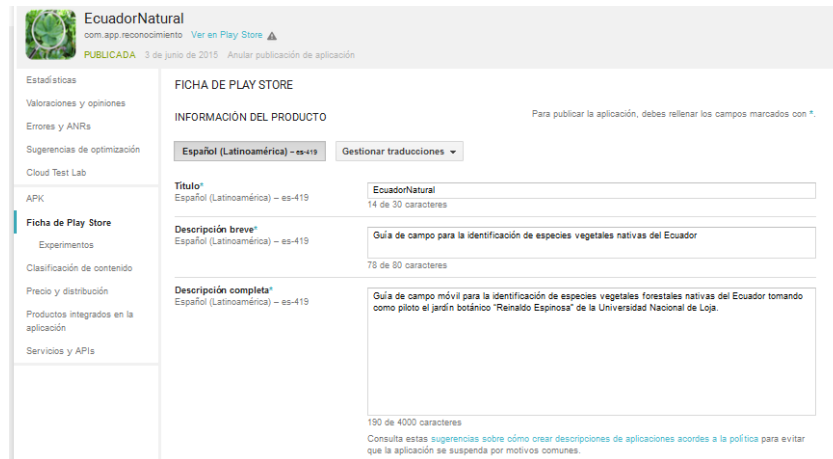


### SUBIR NUEVO APK DE PRODUCCIÓN



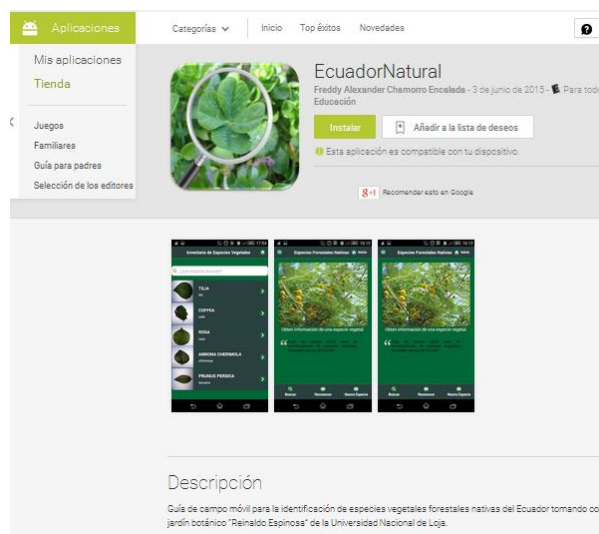
**Figura 33:** Subir APK de producción

A continuación llenamos la ficha de la aplicación con toda la información descriptiva (ver Figura 34).



**Figura 34:** Ficha de la aplicación Android en Google Play

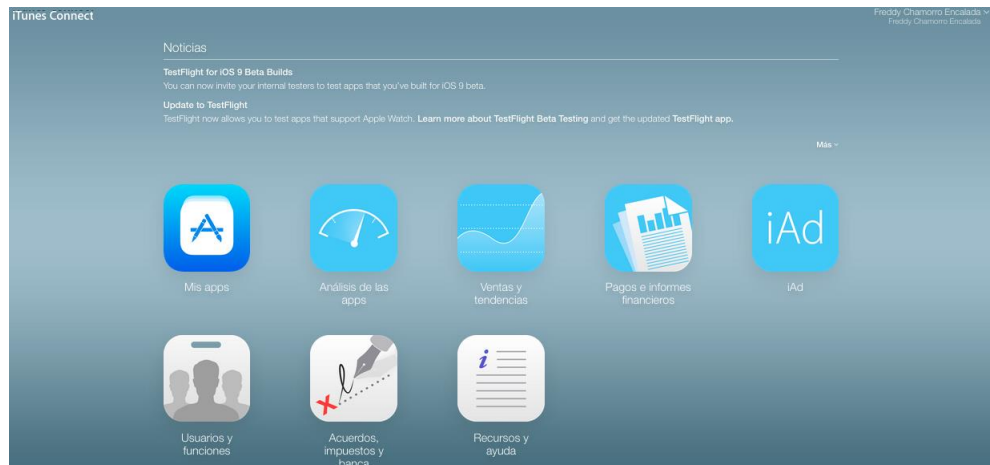
Una vez finalizada la ficha de la aplicación se procede a publicarla y en el transcurso de una o varias horas la aplicación aparecerá en la tienda de Android (Google Play ver Figura 35).



**Figura 35:** Aplicación publicada en Google Play

### 3.9.2 Publicar la aplicación en App Store

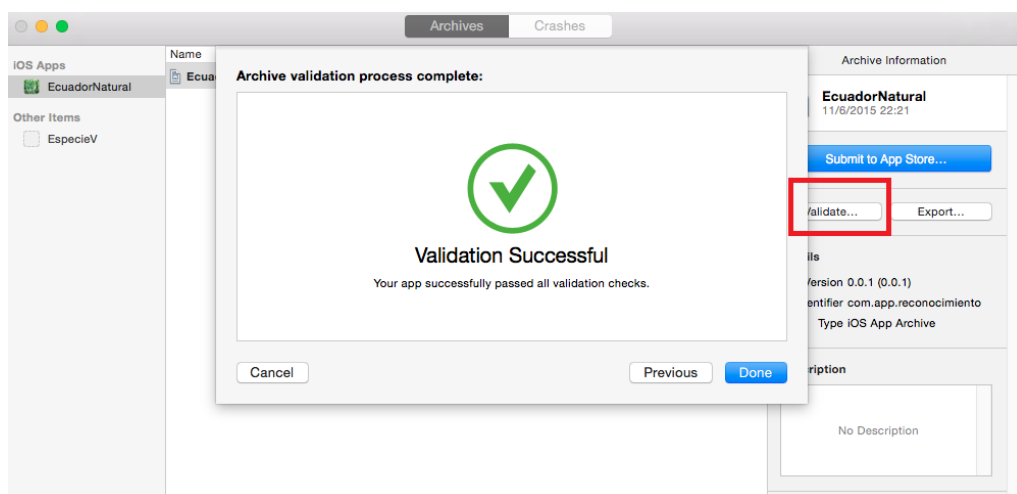
Para publicar la aplicación en App Store primeramente se debe registrar como desarrollador de Apple ingresando a la siguiente dirección <https://developer.apple.com/programs/ios/>, luego de haber registrado los datos y efectuado el pago correspondiente ya se puede publicar la aplicación, para ello se debe ingresar a iTunes en donde se muestra la pantalla de la figura 36.



**Figura 36:** Panel de iTunes.

Ingresamos a Manage Your Apps y después se elige la opción Add New App y se procede a llenar los datos de la aplicación como nombre, descripción, logo, pantallas y finalmente el archivo que contiene la aplicación. También es importante enviar los datos de una cuenta de demostración para que los revisores de Apple puedan verificar las funcionalidades de la aplicación. .

Luego de validar la aplicación en XCode antes de ser enviada a App Store se muestra la pantalla mostrada en la figura 37.



**Figura 37:** Finalizar el proceso de validación.

Una vez validada la aplicación móvil se procede a llenar la información de la misma en la iTunes Connect (Ver figura 38).

The screenshot shows the iTunes Connect interface for submitting an app. The app is titled "Ecuador Natural" with version "Versión 1.0.9". The "Información general de la app" section includes fields for the app icon, ID de Apple (1005293002), and version (1.0.9). The category is set to "Educación" and "Utilidades". The "Información de contacto del representante comercial" section is filled out with "Freddy Chamorro Encalada" as the name, "Loja" as the address, "barrio rumishitana" as the neighborhood, "+593" as the phone prefix, "Ecuador" as the country, and "962825576" as the phone number. The email is "freddyalex636@gmail.com". There is a "Copyright" field with "unl" and a "Selecionar archivo" button for the app's coverage file. The "Compilación" section shows a build for "1.0.9 (1.0.9)".

**Figura 38:** Datos de aplicación a publicar en App Store.

Finalmente luego de haber enviado la aplicación para que sea revisada por Apple se muestra la aplicación como se indica en la figura 39.



**Figura 39:** Aplicación iOS enviada para revisión.

## 4. Fase 4: Pruebas de validación y funcionamiento

En la presente fase se realizó algunas pruebas a la aplicación móvil, dentro de las pruebas ejecutadas se encuentran las funcionales en las cuales se validó los criterios de aceptación de cada requerimiento funcional en las plataformas Android, e iOS. Además se aplicó una encuesta estructurada a una muestra de 30 estudiantes del AARNR de la UNL para validar los requerimientos (ver anexo). Esta encuesta también permitió realizar las pruebas de rendimiento de la aplicación en las plataformas antes mencionadas. Además permitió ejecutar las pruebas de usabilidad a la aplicación móvil. Para finalizar se realizaron pruebas de carga haciendo uso de la herramienta Apache Jmeter.

### 4.1 Pruebas de carga

Para realizar las pruebas de carga se utilizó la herramienta Apache Jmeter, misma que permite determinar el rendimiento del Servidor Web mediante el acceso de un determinado número de usuarios, así como la cantidad de peticiones que se pueden realizar a la aplicación de manera simultánea.

Para determinar la muestra de estudiantes utilizados en la ejecución de las pruebas de carga se empleó la fórmula estadística que se presenta a continuación, en la cual se tomó como base el número de estudiantes y docentes del AARNR de la Universidad Nacional de Loja, cuya cantidad de estudiantes es 693 y de docentes es de 215, dando un total de 908 individuos [29].

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$

n = Tamaño de la muestra.

N = Tamaño de la población (908 estudiantes y docentes)

$\delta$  = Desviación estándar (valor constante de 0,5).

Z= 1,96 (tabla de distribución normal para el 95% de confiabilidad y 5% error).

e = Límite aceptable de error (valor constante 0,05).

#### Aplicación de la fórmula

$$n = \frac{908 * 0.5^2 * 1.96^2}{(908 - 1)0.05^2 + 1.96^2 * 0.5}$$

$$n = \frac{872.04}{4.188}$$

$$n = 208$$

Después de aplicar la fórmula se determinó que la muestra es de 208 individuos.

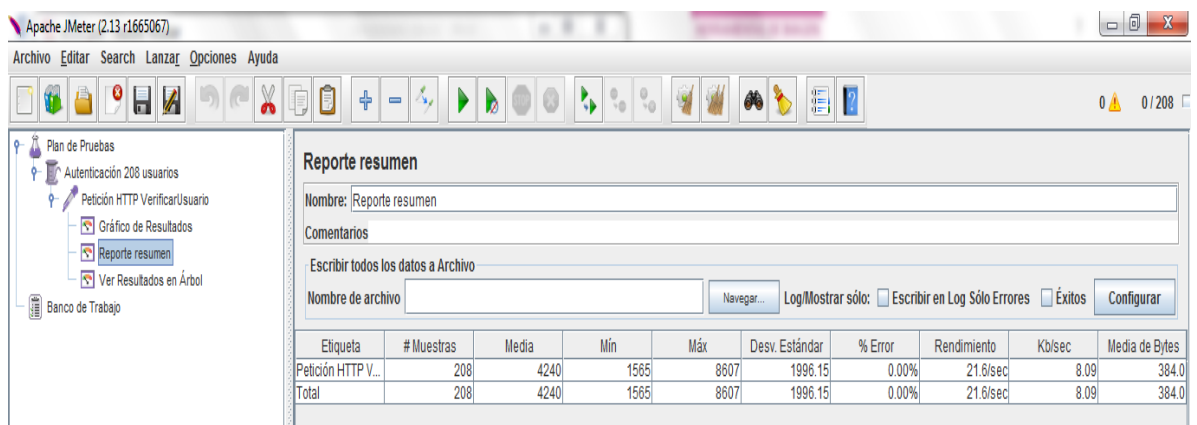
Posteriormente se procedió a ejecutar las pruebas para registro, identificación y consulta de especies vegetales, y autenticación de usuarios con el número de individuos determinados en la muestra. También se realizó las pruebas con una cantidad aleatoria de usuarios (100 y 300), con la finalidad de determinar el rendimiento de la aplicación en cada uno de los escenarios presentados.

#### 4.1.1 Autenticación en la aplicación móvil

##### 1. Caso de prueba

Número de usuarios: 208 (Muestra determinada con la formula estadística explicada anteriormente).

##### Resultados



**Figura 40:** Resultados de prueba de carga de autenticación de usuarios (208 usuarios)

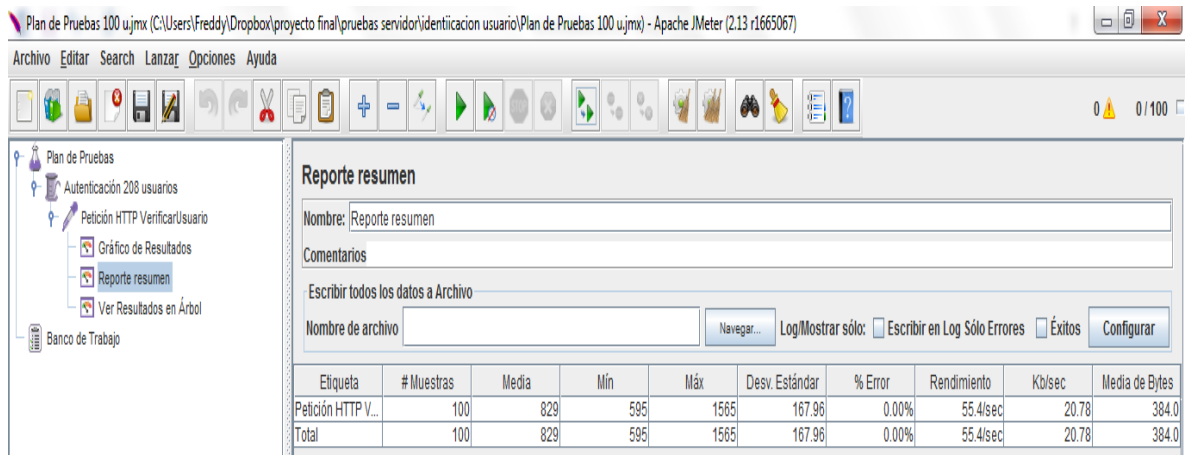
##### Análisis de resultados

Según los resultados obtenidos para una cantidad de 208 usuarios que se autentican en la aplicación de manera simultánea, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es 4240 para las peticiones de los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1565 milisegundos, y el tiempo máximo de respuesta es de 8607 milisegundos. El tamaño medio de las respuestas es de 384 bytes. El rendimiento del servidor es de 21.6 peticiones por segundo.

##### 2. Caso de prueba

Número de usuarios: 100

## Resultados



**Figura 41:** Resultados de prueba de carga de autenticación de usuarios (100 usuarios)

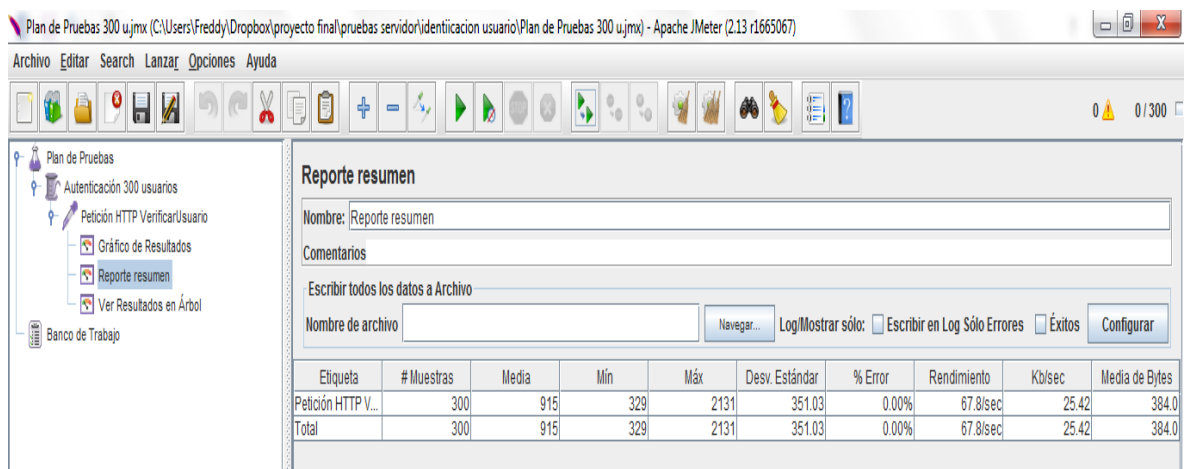
### Análisis de resultados

Según los resultados obtenidos para una cantidad de 100 usuarios que se autentican en la aplicación de manera simultánea, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 829, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 595 milisegundos, y el tiempo máximo de respuesta es de 1565 milisegundos. El tamaño medio de las respuestas es de 384 bytes. El rendimiento del servidor es de 55.4 peticiones por segundo.

### 3. Caso de prueba

Número de usuarios: 300

### Resultados



**Figura 42:** Resultados de prueba de carga de usuarios (300 usuarios)

## Análisis de resultados

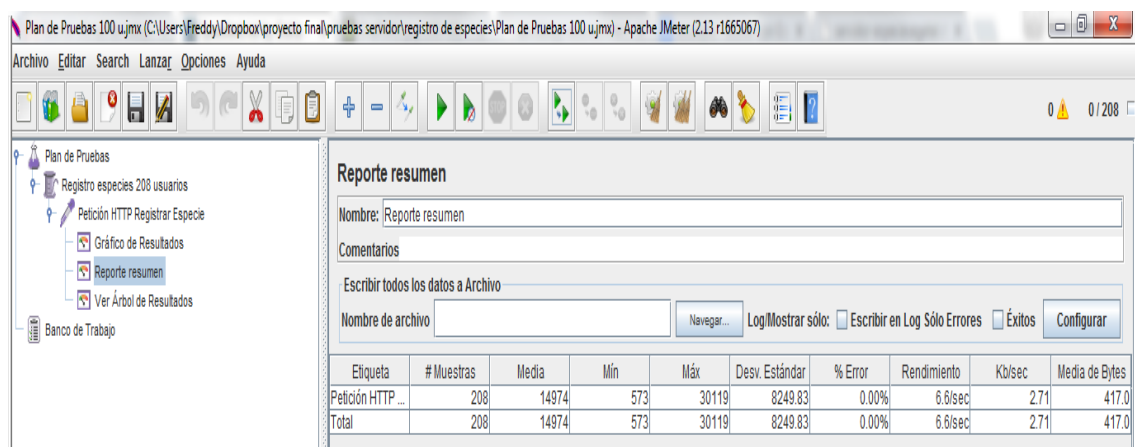
Según los resultados obtenidos para una cantidad de 300 usuarios que se autentican en la aplicación de manera simultánea, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 915, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 329 milisegundos, y el tiempo máximo de respuesta es de 2131 milisegundos. El tamaño medio de las respuestas es de 384 bytes. El rendimiento del servidor es de 67.8 peticiones por segundo.

### 4.1.2 Registrar especies vegetales por medio de una hoja

#### 1. Caso de prueba 1

Número de usuarios: 208 (Muestra determinada con la formula estadística explicada).

#### Resultados



**Figura 43:** Resultados de prueba de carga de registro de especies vegetales (208 usuarios)

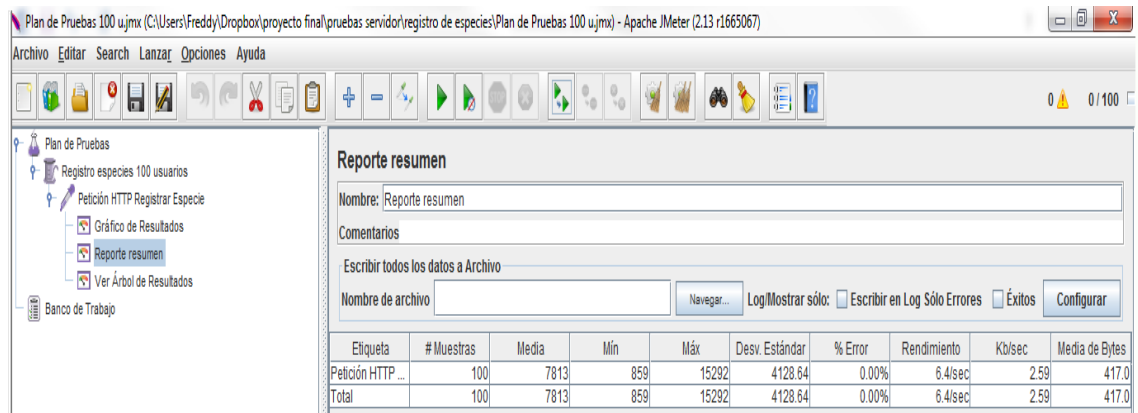
## Análisis de resultados

Según los resultados obtenidos para una cantidad de 208 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 14974, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 573 milisegundos, y el tiempo máximo de respuesta es de 30119 milisegundos. El tamaño medio de las respuestas es de 417 bytes. El rendimiento del servidor es de 6.6 peticiones por segundo.

#### 2. Caso de prueba 2

Número de usuarios: 100

## Resultados



**Figura 44:** Resultados de prueba de carga de registro de especies vegetales (100 usuarios)

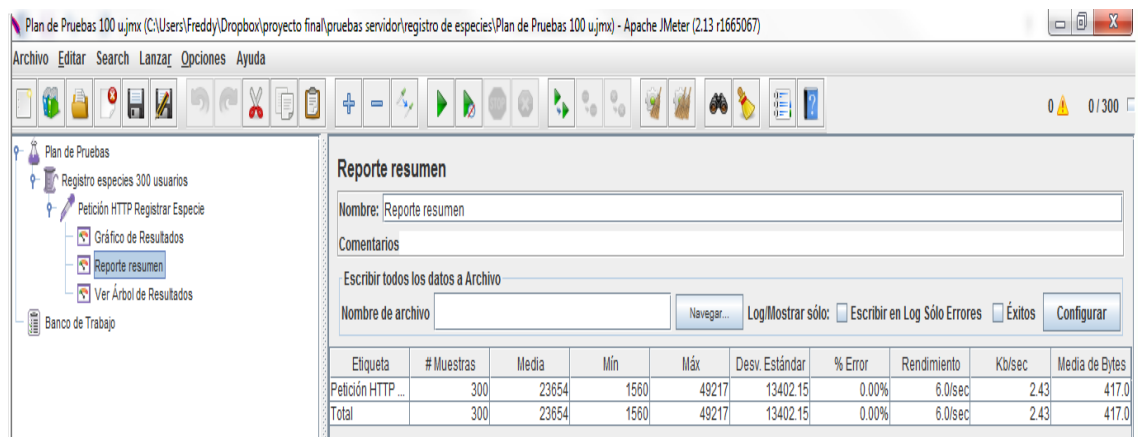
### Análisis de resultados

Según los resultados obtenidos para una cantidad de 100 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, en un periodo de tiempo de subida de 1 segundo se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 7813, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 859 milisegundos, y el tiempo máximo de respuesta es de 15292 milisegundos. El tamaño medio de las respuestas es de 417 bytes. El rendimiento del servidor es de 6.4 peticiones por segundo.

### 3. Caso de prueba 3

Número de usuarios: 300

## Resultados



**Figura 45:** Resultados de prueba de carga de registro de especies vegetales (300 usuarios)



## Análisis de resultados

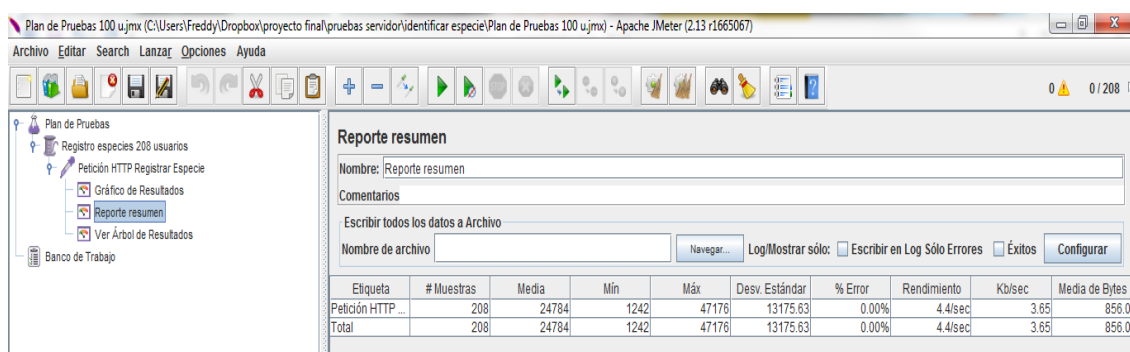
Según los resultados obtenidos para una cantidad de 300 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 23654, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1560 milisegundos, y el tiempo máximo de respuesta es de 49217 milisegundos. El tamaño medio de las respuestas es de 417 bytes. El rendimiento del servidor es de 6 peticiones por segundo.

### 4.1.3 Identificar especies vegetales por medio de una hoja

#### 1. Caso de prueba 1

Número de usuarios: 208 (Muestra determinada con la formula estadística explicada anteriormente).

#### Resultados



**Figura 46:** Resultados de prueba de carga de identificación de especies vegetales (208 usuarios)

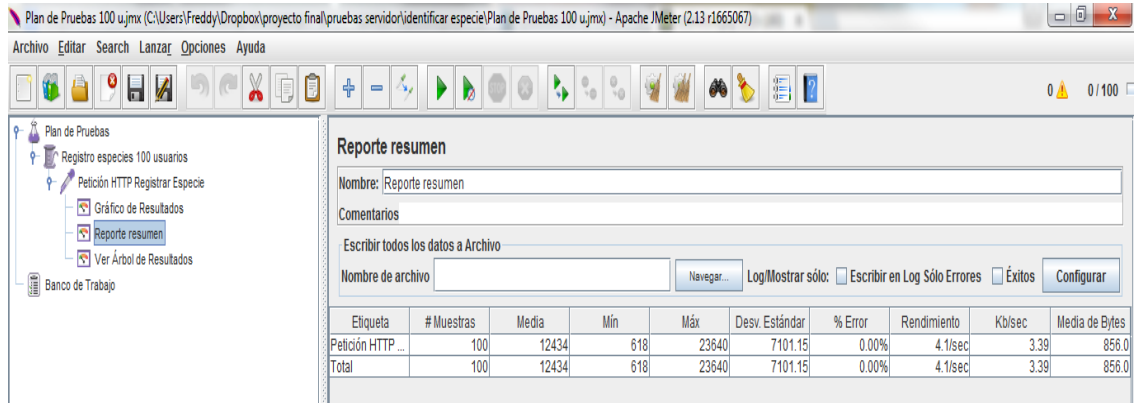
## Análisis de resultados

Según los resultados obtenidos para una cantidad de 208 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 24784, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1242 milisegundos, y el tiempo máximo de respuesta es de 47176 milisegundos. El tamaño medio de las respuestas es de 856 bytes. El rendimiento del servidor es de 4.4 peticiones por segundo.

## 2. Caso de prueba 2

Número de usuarios: 100

### Resultados



**Figura 47:** Resultados de prueba de carga de identificación de especies vegetales (100 usuarios)

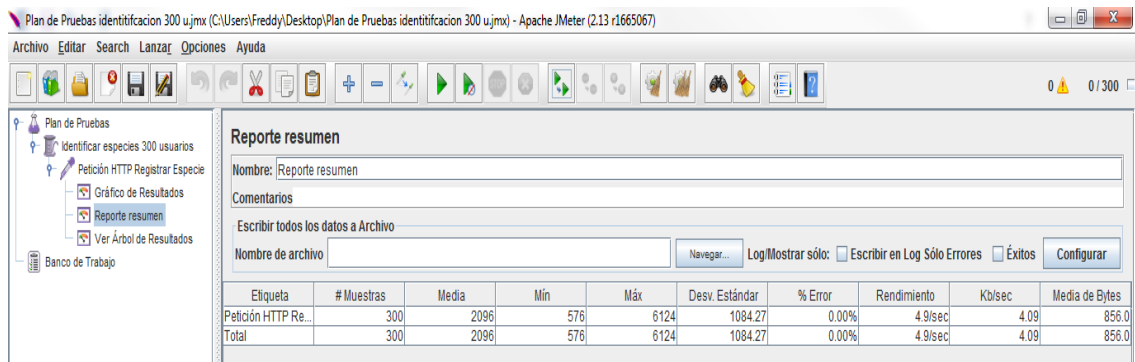
### Análisis de resultados

Según los resultados obtenidos para una cantidad de 100 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 12434, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 618 milisegundos, y el tiempo máximo de respuesta es de 23640 milisegundos. El tamaño medio de las respuestas es de 856 bytes. El rendimiento del servidor es de 4.1 peticiones por segundo.

## 3. Caso de prueba 3

Número de usuarios: 300

### Resultados



**Figura 48:** Resultados de prueba de carga de identificación de especies vegetales (300 usuarios)

## Análisis de resultados

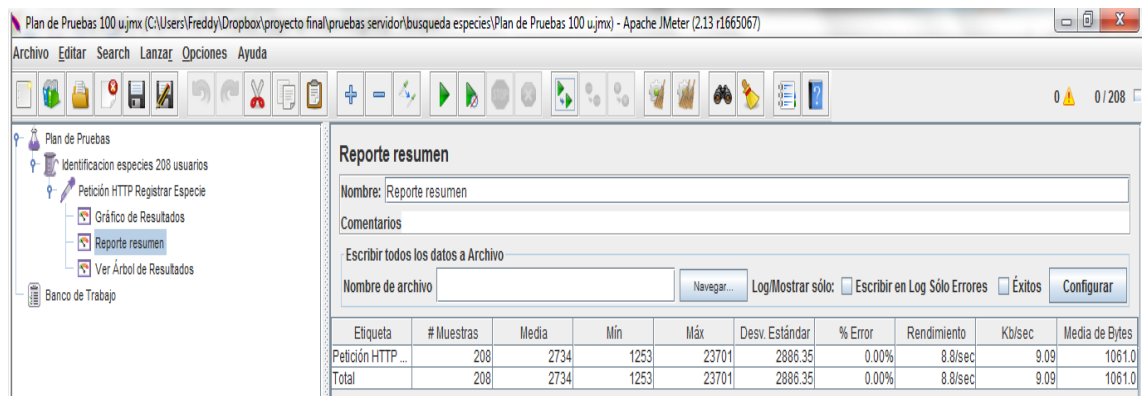
Según los resultados obtenidos para una cantidad de 300 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 60 segundos, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 2096, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 576 milisegundos, y el tiempo máximo de respuesta es de 6124 milisegundos. El tamaño medio de las respuestas es de 856 bytes. El rendimiento del servidor es de 4.9 peticiones por segundo.

### 4.1.4 Buscar todas las especies vegetales

#### 1. Caso de prueba 1

Número de usuarios: 208 (Muestra determinada con la formula estadística explicada anteriormente).

#### Resultados



**Figura 49:** Resultados de prueba de carga de la búsqueda de todas las especies vegetales (208 usuarios)

## Análisis de resultados

Según los resultados obtenidos para una cantidad de 208 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 2734, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1253 milisegundos, y el tiempo máximo de respuesta es de 23701 milisegundos. El tamaño medio de las respuestas es de 1061 bytes (variable al número de especies). El rendimiento del servidor es de 8.8 peticiones por segundo.

## 2. Caso de prueba 2

Número de usuarios: 100

### Resultados



**Figura 50:** Resultados de prueba de carga de la búsqueda de todas las especies vegetales (100 usuarios)

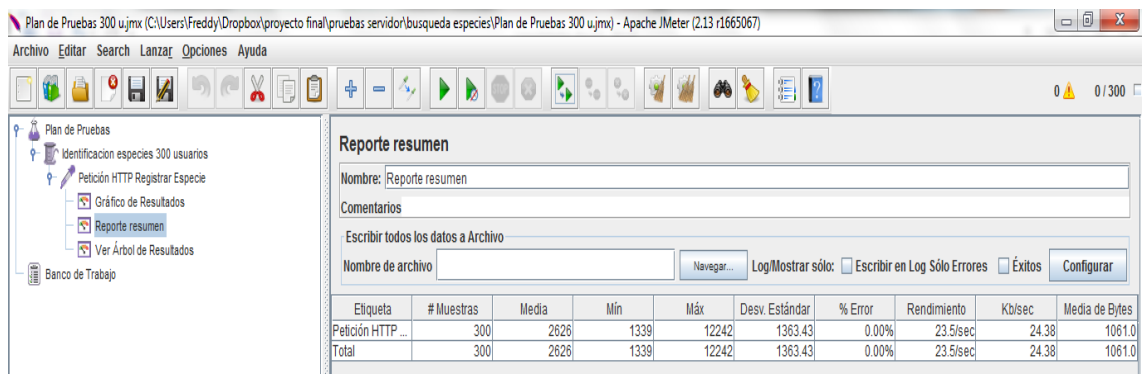
### Análisis de resultados

Según los resultados obtenidos para una cantidad de 100 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 2418, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1257 milisegundos, y el tiempo máximo de respuesta es de 7136 milisegundos. El tamaño medio de las respuestas es de 1061 bytes (variable al número de especies). El rendimiento del servidor es de 13.0 peticiones por segundo.

## 3. Caso de prueba 3

Número de usuarios: 300

### Resultados



**Figura 51:** Resultados de prueba de carga de la búsqueda de todas las especies vegetales (300 usuarios)

## Análisis de resultados

Según los resultados obtenidos para una cantidad de 300 usuarios que registran un ejemplar de una especie vegetal en la aplicación móvil de manera simultánea, con un periodo de tiempo de subida de 1 segundo, se puede determinar que la prueba se ha realizado sin errores. El tiempo promedio en milisegundos es de 2626, para la respuesta los usuarios. El tiempo mínimo de respuesta del servidor al usuario es de 1339 milisegundos, y el tiempo máximo de respuesta es de 12242 milisegundos. El tamaño medio de las respuestas es de 1061 bytes (variable al número de especies). El rendimiento del servidor es de 23.5 peticiones por segundo.

### 4.2 Pruebas funcionales

Las presentes pruebas se las ha realizado validando los criterios de aceptación planteados en la especificación de requerimientos funcionales en las 2 plataformas seleccionadas (Android, iOS). También se ha argumentado la validación mediante una encuesta aplicada a los usuarios reales de la aplicación (ver Anexo 6: Encuesta).

#### 4.2.1 Caso de Prueba

Las pruebas funcionales de la aplicación se las ha realizado tomando como ejemplo un usuario registrado con datos del desarrollador del presente PFC, con esta cuenta se ha validado cada uno de los requerimientos funcionales especificados (Ver sección Resultados Fase 3, subsección iteración 0). A continuación se presenta la validación de cada requerimiento.

#### 4.2.2 Ejecución de pruebas

##### a. Requerimiento 1

<b>Referencia</b>	RF-001	
<b>Nombre</b>	Ingresar a la aplicación	
<b>Descripción</b>	La aplicación móvil permitirá al Usuario ingresar a la aplicación mediante el correo electrónico y clave, previamente registrados.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
Criterios de Aceptación		Estado
Verificar que al ejecutar la aplicación móvil se muestre en la pantalla [Inicio de sesión], la opción para la identificación del usuario mediante clave y correo electrónico personal.		Se cumple el criterio.
Verificar que al seleccionar la opción <b>[Recordar Usuario]</b> de la pantalla de Inicio de sesión, se guarde los datos de identificación del usuario.		Se cumple el criterio.

Verificar si al presionar el botón <b>[Entrar]</b> con uno o los dos campos vacíos, la aplicación presenta un mensaje indicativo.	Se cumple el criterio.
Verificar si al presionar el botón <b>[Entrar]</b> sin una conexión a internet, la aplicación presenta un mensaje indicativo.	Se cumple el criterio.
Verificar si cuando se ingresa en correo electrónico o la clave incorrecta, la aplicación presenta un mensaje indicando la razón por la que no se puede ingresar a la pantalla <b>[Principal]</b> .	Se cumple el criterio.
Verificar si al ingresar los datos de la cuenta correctamente, la aplicación muestra la pantalla <b>[Principal]</b> con el perfil correspondiente al usuario.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 1.

#### b. Requerimiento 2

<b>Referencia</b>	RF-002	
<b>Nombre</b>	Registrar en la aplicación	
<b>Descripción</b>	La aplicación móvil permitirá al Usuario registrarse en la aplicación con sus datos personales (nombres, apellidos, correo electrónico, clave y perfil de usuario).	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al presionar el botón <b>[Registrar]</b> , con los nombre o apellidos que contienen caracteres no válidos, la aplicación presenta un mensaje indicativo.		Se cumple el criterio.
Verificar si al presionar el botón <b>[Registrar]</b> , con el correo electrónico no valido, la aplicación presenta un mensaje indicativo.		Se cumple el criterio.
Verificar si al presionar el botón <b>[Registrar]</b> , con uno o varios campos vacíos, la aplicación presenta un mensaje indicativo.		Se cumple el criterio.
Permitir al usuario registrar en la aplicación con dos tipos de perfiles, " <b>Consultor</b> ", para realizar identificación de especies vegetales en la aplicación, o " <b>Colaborador</b> ", para aportar en el registro de especies vegetales a la aplicación.		Se cumple el criterio.

Verificar si al presionar el botón <b>[Registrar]</b> con los datos correctos, la aplicación presenta un mensaje indicativo y regresa a la pantalla de Inicio de sesión.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 2.

### c. Requerimiento 3

<b>Referencia</b>	RF-003
<b>Nombre</b>	Verificar perfil de usuario
<b>Descripción</b>	La aplicación móvil permitirá al usuario visualizar las opciones disponibles en la aplicación de acuerdo a su perfil de usuario.
<b>Prioridad</b>	Alta
<b>Versión</b>	1
<b>Criterios de Aceptación</b>	
Verificar si al ingresar en la aplicación, en la pantalla principal se muestran las opciones disponibles de acuerdo al perfil de usuario elegido en el registro.	Se cumple el criterio.
Verificar si al ingresar en la aplicación, se muestra los datos de usuario, nombres y apellidos, que corroboren el inicio de sesión del usuario.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 3.

### d. Requerimiento 4

<b>Referencia</b>	RF-004
<b>Nombre</b>	Consultas de especies vegetales forestales nativas
<b>Descripción</b>	La aplicación móvil permitirá al usuario consultar todas las especies vegetales registradas en la aplicación.
<b>Prioridad</b>	Alta
<b>Versión</b>	1
<b>Criterios de Aceptación</b>	
Verificar si al presionar el botón <b>[Consultar]</b> , se muestra la pantalla de búsqueda especies vegetales.	Se cumple el criterio.

Verificar si al mostrar la pantalla de Búsqueda de especies vegetales, se muestran todas las especies vegetales registradas en la aplicación móvil.	Se cumple el criterio.
Verificar si al filtrar la búsqueda de especies vegetales, se filtra el listado por el nombre común o científico de la especie vegetal.	Se cumple el criterio.
Verificar si al seleccionar de la lista, la especie vegetal de interés, se muestra la pantalla Información detallada de la especie vegetal.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 4.

#### e. Requerimiento 5

<b>Referencia</b>	RF-005	
<b>Nombre</b>	Identificación de especies vegetales forestales nativas	
<b>Descripción</b>	La aplicación móvil permitirá al usuario identificar las especies vegetales por medio de una hoja de la especie vegetal en referencia a las especies vegetales registradas en la aplicación.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar si al presionar el botón <b>[Identificar]</b> , se muestran la cámara del dispositivo móvil, para la captura de la hoja de la especie vegetal.		Se cumple el criterio.
Verificar si al capturar la imagen de la hoja de la especie vegetal, se muestra la pantalla de Identificación de la especie vegetal.		Se cumple el criterio.
Verificar si en la pantalla de Identificación de especie vegetal, se muestra la captura de la imagen de la especie vegetal.		Se cumple el criterio.
Verificar si al seleccionar el botón <b>[Recapturar Hoja]</b> , se muestra la cámara del dispositivo para recapturar la especie vegetal de acuerdo a las especificaciones.		Se cumple el criterio.
Verificar si al presionar el botón <b>[Buscar Especie]</b> , sin una conexión a internet, la aplicación muestra un mensaje indicativo.		Se cumple el criterio.



Verificar si al presionar el botón [Identificar Especie], la aplicación muestra la pantalla <b>[Búsqueda de especies vegetales]</b> , con las especies vegetales más cercanas a la búsqueda.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 5.

#### f. Requerimiento 6

<b>Referencia</b>	RF-006	
<b>Nombre</b>	Registro de especies vegetales forestales nativas	
<b>Descripción</b>	La aplicación móvil permitirá al usuario con perfil " <b>Colaborador</b> ", registrar las especies vegetales por medio de una hoja de la especie vegetal, con nombre común y científico, además de las coordenadas geográficas del registro de la especie vegetal.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
	<b>Criterios de Aceptación</b>	<b>Estado</b>
	Verificar si al presionar el botón <b>[Registrar]</b> , se muestra la pantalla Registro de especie vegetal, con los datos de geolocalización actual (latitud, longitud).	Se cumple el criterio.
	Verificar si al presionar el botón <b>[Capturar]</b> , se muestra la cámara del dispositivo para capturar la hoja de la especie vegetal.	Se cumple el criterio.
	Verificar si al presionar el botón <b>[Registrar]</b> , con uno o los dos campos vacíos, la aplicación muestra un mensaje indicativo.	Se cumple el criterio.
	Verificar si al presionar el botón <b>[Registrar]</b> , caracteres ingresados no válidos, la aplicación muestra un mensaje indicativo.	Se cumple el criterio.
	Verificar si al presionar el botón <b>[Registrar]</b> , sin capturar una imagen, la aplicación muestre un mensaje indicativo.	Se cumple el criterio.
<b>Observaciones</b>	Las pantallas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 6.	

#### g. Requerimiento 7

<b>Referencia</b>	RF-007	
<b>Nombre</b>	Implementar el servidor web para respuesta de peticiones de la aplicación móvil.	
<b>Descripción</b>	Servidor web para responder peticiones de aplicación móvil para el reconocimiento de especies vegetales forestales nativas.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
Criterios de Aceptación		Estado
Verificar que el servidor web, recibe peticiones de la aplicación móvil y responde las peticiones en formato JSON.		Se cumple el criterio.
Verificar que el servidor web responda de manera rápida a las solicitudes de la aplicación móvil.		Se cumple el criterio.
Verificar que el servidor web, sea de acceso público.		Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 7, 8, 9, 10, 11.	

#### h. Requerimiento 8

<b>Referencia</b>	RF-008	
<b>Nombre</b>	Verificar usuario en el Servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de verificación de usuario emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
Criterios de Aceptación		Estado
Verificar que el servidor acepta la solicitud de verificación de usuario con los datos emitidos desde la aplicación móvil.		Se cumple el criterio.
Verificar que el servidor emita una respuesta en formato JSON.		Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 7.	

### i. Requerimiento 9

<b>Referencia</b>	RF-009	
<b>Nombre</b>	Registrar usuario en la Base de Datos.	
<b>Descripción</b>	El servidor web recibe la petición de registro de usuario desde la aplicación móvil con los datos personales, para el registro en la Base de datos.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de registro de usuario con los datos emitidos desde la aplicación móvil.		Se cumple el criterio.
Verificar que el servidor valida los datos del usuario con los existentes en la Base de Datos.		Se cumple el criterio.
Verificar que el servidor registra el nuevo usuario en la base de datos.		Se cumple el criterio.
Verificar que el servidor emita una respuesta en formato JSON.		Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 8. Los criterios de aceptación 2 y 3, se validan en el anexo 5 sección 7.	

### j. Requerimiento 10

<b>Referencia</b>	RF-0010	
<b>Nombre</b>	Consultar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de consulta de todas las especies vegetales registradas emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
<b>Criterios de Aceptación</b>		<b>Estado</b>
Verificar que el servidor acepta la solicitud de consultar todas las especies vegetales emitida desde la aplicación móvil.		Se cumple el criterio.
Verificar que el servidor consulte en la Base de Datos, todas las especies vegetales registradas.		Se cumple el criterio.

Verificar que el servidor emita una respuesta en formato JSON.	Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 9.

#### k. Requerimiento 11

<b>Referencia</b>	RF-0011	
<b>Nombre</b>	Identificar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de identificación de la especie vegetal emitida por la aplicación móvil.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
	<b>Criterios de Aceptación</b>	<b>Estado</b>
	Verificar que el servidor acepta la solicitud de identificación de especies vegetales con la imagen de la hoja de la especie vegetal emitida desde la aplicación móvil.	Se cumple el criterio.
	Verificar que el servidor identifique la especie vegetal de acuerdo a las muestras registradas en la Base de Datos.	Se cumple el criterio.
	Verificar que el servidor emita una respuesta en formato JSON.	Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 10.	

#### l. Requerimiento 12

<b>Referencia</b>	RF-0012	
<b>Nombre</b>	Registrar especie vegetal en el servidor web.	
<b>Descripción</b>	El servidor responde a la solicitud de registro de la especie vegetal, y guarda la especie vegetal emitida por la aplicación móvil en la Base de Datos.	
<b>Prioridad</b>	Alta	
<b>Versión</b>	1	
	<b>Criterios de Aceptación</b>	<b>Estado</b>

Verificar que el servidor acepta la solicitud de registro de especies vegetales con los datos de la especie vegetal emitida desde la aplicación móvil.	Se cumple el criterio.
Verificar que el servidor valide la especie vegetal de acuerdo a las muestras registradas en la Base de Datos.	Se cumple el criterio.
Verificar que el servidor registra la nueva especie vegetal o muestra de una especie vegetal existente en la base de datos.	Se cumple el criterio.
Verificar que el servidor emita una respuesta en formato JSON.	Se cumple el criterio.
<b>Observaciones</b>	Las pruebas de ejecución correspondientes a la validación del presente requerimiento se encuentran en el Anexo 5 sección 11.

### 4.3 Pruebas de rendimiento

Las pruebas de rendimiento se las realizó mediante la ejecución de la aplicación en diferentes versiones de cada sistema operativo, para ello se realizó una encuesta (Ver anexo 6), a usuarios reales que manipularon la aplicación.

De acuerdo a las encuestas realizadas se determinó que el rendimiento de la aplicación varía de acuerdo a la versión de sistema operativo Android. En la Tabla XXI se indica el rendimiento por cada versión.

**TABLA XXI. RENDIMIENTO DE LA APLICACIÓN EN VARIOS DISPOSITIVOS MÓVILES**

Versión	Rendimiento
Android 4.4	Excelente
Android 4.3	Excelente
Android 4.2	Excelente
Android 3.0.0 – 3.X.X	Bueno
Android 2.3.3 – 2.3.7	Regular

Se pudo determinar que la aplicación móvil funciona bien en el sistema operativo android a partir de la versión 2.3.3 y posteriores. Además se pudo corroborar que la versión de sistema operativo android utilizada es la 3.X.X.

En el sistema operativo iOS se determinó que el rendimiento de la aplicación es excelente en las versiones 7.X.X y posteriores. Los resultados antes presentados se los

puede verificar en el Anexo 7 en donde se encuentra más detalladamente los resultados obtenidos en la encuesta.

#### **4.4 Pruebas de usabilidad**

La usabilidad dentro de un software implica algunos aspectos, entre los más importantes se tiene la eficiencia y eficacia de una aplicación, por esta razón en este apartado se evaluó la aplicación considerando los dos parámetros antes mencionados. Para ello se utilizó la encuesta antes mencionada de la cual se pudo obtener los siguientes resultados:

Se pudo evidenciar que la aplicación móvil tiene buenos tiempos de respuesta en la identificación de especies con un 65%, y 20% de los encuestados piensan que la aplicación tiene excelente tiempo de respuesta y el 15% cree que el tiempo de respuesta es regular, cabe mencionar que esto se debe a la conexión de internet utilizada.

Además se pudo determinar que la aplicación móvil permite en un 100% identificar especies vegetales, y un 95% de los encuestados obtiene los resultados esperados, el 5% que no obtiene los resultados esperados es porque la especie vegetal no se encuentra registrada correctamente. Al mismo tiempo se pudo determinar que la aplicación permite registrar usuarios, asignar un perfil, buscar especies vegetales, identificar y registrar especies vegetales en un 100%. Para finalizar los encuestados consideran que la aplicación móvil es una excelente herramienta que facilita la identificación de especies vegetales obteniendo información oportuna (ver Anexo 7: Resultados de la Encuesta).

## **G. Discusión**

### **1. Desarrollo de la propuesta alternativa**

El desarrollo del proyecto de titulación consta de 4 fases de las cuales, la primera trata sobre la determinación de la situación actual y formulación del problema que se establecieron en el anteproyecto, en la segunda fase se realizó el análisis y se determinó las herramientas a utilizar, en la tercera fase se ejecutó el diseño e implementación de la aplicación, y para finalizar en la cuarta fase se ejecutaron las respectivas pruebas para validar el correcto funcionamiento de la misma, en la identificación de especies vegetales forestales nativas. Para alcanzar los objetivos inicialmente planteados a continuación se describe cada uno de ellos y las actividades realizadas:

#### **1.1 Establecer una solución factible a través de un análisis comparativo entre distintas posibilidades de implementación, tomando en cuenta algoritmos, arquitectura de hardware y software y modelo de datos.**

Para alcanzar este objetivo se realizó el análisis de los frameworks más utilizados para la implementación de una aplicación multiplataforma (ver sección Resultados, fase 2, subsección 2.1), esto con la finalidad de determinar cuál de ellos es el idóneo para el desarrollo, para continuar se procedió a realizar un análisis para definir la librería adecuada para el reconocimiento de imágenes para su implementación en el servidor web, se eligió OpenCV por su amplia documentación y fácil conexión con Python (ver sección Resultados, fase 2, subsección 2.2), además se estableció un proveedor de alojamiento gratuito para el servidor web tomando en cuenta que soporte la librería de reconocimiento de imágenes OpenCV, el framework para el servidor web y la base de datos a utilizar (ver sección Resultados, fase 2, subsección 2.3, 2.4, 2.5). Luego se procedió a investigar y realizar comparativas entre las metodologías de desarrollo de software permitiendo definir la que se adapte a la construcción de aplicaciones móviles (Ver sección de Resultados, fase 2, subsección 2.6).

Para continuar se determinó el equipo de trabajo (ver sección de Resultados, fase 3), personal involucrado, características de los usuarios, los requerimientos funcionales y no funcionales de la aplicación, el diagrama de casos de uso con la finalidad de comprender la interacción del usuario con la aplicación móvil y esta a su vez con el servidor web, se realizó la descripción de cada caso de uso, se analizó las diferentes arquitecturas posibles de implementación y se optó por una que favorezca al mayor

número de dispositivos móviles utilizados en el presente campo de estudio (ver sección de Resultados, fase 3, subsección 3.1).

Posteriormente se estudió el modelo de datos con la finalidad de decidir que estructura de base de datos a utilizar para almacenar la información de los usuarios y de las especies vegetales forestales nativas registradas para finalizar se estableció los tiempos de cada iteración para la construcción de la aplicación móvil.

### **1.2 Implementar un algoritmo de reconocimiento de imágenes que permita encontrar eficazmente la imagen más parecida a la de búsqueda en una base de datos de imágenes accesible vía web.**

La identificación de una especie vegetal se pudo realizar por la forma de las hojas, debido a que tienen una mayor presencia que flores, frutos y semillas, además que se las encuentra en la mayoría de las especies vegetales. Primero se analizó el algoritmo para la segmentación de la hoja, para lo cual se pide que la fotografía sea tomada con un fondo blanco, y así poder aplicar un criterio de sustracción de fondo, por el brillo de los píxeles, considerando que un píxel que es cercano al blanco es parte del fondo y el resto es parte de la hoja. De esto se reemplaza cada píxel del fondo con un píxel negro y cada píxel de la hoja con uno blanco (ver Anexo 2, sección 1).

Para obtener la forma de la imagen es necesario obtener los bordes o contorno de la imagen. De la investigación bibliográfica y de los casos de éxitos analizados, se pudo determinar que esta funcionalidad se encuentra implementada en la librería OpenCV, que entrega una muy buena aproximación del borde.

Luego de obtener el borde de la hoja fue necesario obtener los descriptores elípticos de la forma de la hoja que se puede lograr con una función que describa el borde como una curva o una que describa el área de la hoja, las funciones implementadas en el servidor web para la obtención de los descriptores de la hoja de la especie vegetal, se encuentran detalladas en el Anexo 2, sección 1.

### **1.3 Diseñar una guía de campo móvil multiplataforma para dispositivos móviles smartphone con mayor demanda en la carrera de Ingeniería Forestal del AARNR de la UNL.**

En el diseño e implementación de la aplicación móvil, primero se creó una aplicación móvil híbrida haciendo uso del framework Phonegap, que es una distribución libre de Apache Cordova el cual presentó dificultades al momento de utilizar la geolocalización en dispositivos iOS, por lo cual se optó por el framework Apache Cordova en su versión más reciente que otorgó los resultados esperados (ver sección Resultado, fase 3,



subsección 3.3), posteriormente se agregó las plataformas de desarrollo y los complementos dependiendo de los componentes internos del dispositivo móvil a utilizar (cámara, geolocalización, información de la red, transferencia de datos, entre otros), y se diseñó la interfaz de usuario (ver sección de Resultados, fase 3, subsección 3.1), para lo cual se utilizó el Framework Front-End JQuery Mobile, que permitió el desarrollo ágil de la misma.

Para continuar se codificaron las funciones necesarias para el registro y autenticación del usuario en la aplicación móvil, con la finalidad de determinar el perfil del usuario, para acorde a este proporcionar las funcionalidades de la aplicación; a continuación se implementó el servidor web en una red local con el framework bottle en el cual se pudo determinar que al procesar una solicitud este se bloquea restringiendo las peticiones de los demás usuarios, por cuanto se optó por migrar el código al framework tornado, con la base de datos, conexión, funciones y módulos necesarios para la comunicación con la aplicación móvil para la autenticación y registro de usuarios, obteniendo mejores resultados (ver sección Resultados, fase 3, subsección 3.3), es necesario mencionar que el servidor web envía una respuesta en formato JSON y la aplicación móvil se encarga de procesarla.

Una vez lograda la autenticación de usuarios, se codificó las funciones y módulos correspondientes para el registro de especies vegetales con una petición enviada en formato JSON desde la aplicación móvil al servidor web, para el registro de la especie vegetal la imagen se almacena en un servidor gratuito de imágenes externo, con la finalidad de ahorrar espacio local además fue necesario reducir el tamaño de la imagen para obtener resultados en menos tiempo, una vez enviada la petición de registro de la especie vegetal desde la aplicación móvil, el servidor se encarga de buscar la existencia de la especie en la base de datos, en caso de existir se agrega como un ejemplar de la especie vegetal caso contrario se crea la especie vegetal como nueva; esta funcionalidad está disponible solo para los usuarios que tienen perfil de Colaborador, además se utilizaron las funciones para la obtención de descriptores de la imagen y almacenarlos para la identificación de especies vegetales (ver sección Resultados, fase 3, subsección 3.4),

Luego de haber establecido los métodos para el registro de especies vegetales se procedió a realizar la población de la base de datos con las especies vegetales forestales nativas más significativas localizadas en el jardín botánico "Reinaldo Espinosa" de la UNL, para lo cual se clasificó las especies vegetales por hojas, simples y

compuestas, con la finalidad de precisar los resultados en la identificación de especies vegetales.

Una vez concluida la iteración de registro de especies vegetales se codificaron las funciones correspondientes en la aplicación móvil y en el servidor web para la identificación de especies vegetales por medio de una hoja de la especie, además se hizo uso de un servidor gratuito de imágenes para alojar la imagen de la especie vegetal y posteriormente se envía la dirección pública de la imagen para el procesamiento en el servidor web, en donde se obtienen los descriptores de la imagen y se procede a comparar con los descriptores de las imágenes almacenadas tomando las 5 especies más cercanas, la respuesta que otorga el servidor es un conjunto de 5 especies vegetales más cercanas a la búsqueda (ver sección Resultados, fase 3, subsección 3.5), Luego se añadió la funcionalidad de consulta de todas las especies vegetales, en donde el usuario puede observar la lista de todas la especies vegetales registradas, para lo cual se desarrollaron las funciones correspondientes en el servidor web y en la aplicación móvil (ver sección Resultados, fase 3, subsección 3.6),

Posterior a la implementación de la aplicación móvil y el servidor web de manera local se implantó el servidor web en la nube en el proveedor de alojamiento gratuito openshift, que mediante acceso por consola permitió la instalación de las librerías y complementos para el reconocimiento de imágenes con OpenCV, además otorgó la facilidad para establecer la estructura de la base de datos y lograr el funcionamiento del servidor web obtenido en la red local (ver sección Resultados, fase 3, subsección 3.7).

Para continuar se realizó la compilación de la aplicación móvil para el Sistema Operativo iOS, es necesario mencionar que la aplicación de codificó en un inicio para la plataforma android y luego se lo implementó para iOS con la ayuda de un computador con Sistema Operativo OS y la herramienta Xcode (ver sección Resultados, fase 3, subsección 3.8). Para finalizar el desarrollo de la guía de campo móvil se procedió a la publicación en las diferentes tiendas oficiales de aplicaciones móviles para android e iOS con el nombre de EcuadorNatural, accesible de manera gratuita, para lo cual se tuvo que adquirir una licencia de desarrollador para cada plataforma y esperar la revisión y publicación por parte de Google y Apple en las respectivas tiendas (ver sección Resultados, fase 3, subsección 3.9).

#### **1.4 Determinar el correcto funcionamiento de la aplicación en diversos dispositivos móviles, sistemas operativos y distintos escenarios mediante pruebas piloto.**

Para el cumplimiento de este objetivo se realizaron pruebas a la aplicación móvil y al servidor web, dentro de las cuales están las pruebas de carga, que sirvieron para determinar la cantidad de usuarios que pueden ejecutar peticiones de registro y autenticación de usuario, identificación, búsqueda y registro de especies vegetales al servidor de manera simultánea (ver sección Resultados, fase 4, subsección 4.1).

Además se efectuó pruebas funcionales, las cuales permitieron verificar si la aplicación móvil realizada cumple los criterios de aceptación planteados en la especificación de requerimientos. Esta verificación se ejecutó en las 2 plataformas seleccionadas Android e iOS, además permitió verificar si el servidor web implementado acepta las peticiones requeridas y entrega las respuestas esperadas según los requerimientos (ver sección Resultados, fase 4, subsección 4.2).

Para finalizar se ejecutaron pruebas de rendimiento y de usabilidad en las plataformas Android e iOS (Ver sección de Resultados, fase 4, subsecciones 4.3-4.4), para ello se realizó una encuesta estructurada a los usuarios reales que manipularon la aplicación móvil, con la finalidad de determinar la utilidad, el grado de aceptación y el rendimiento de la misma.

## **2. Valoración técnica económica ambiental**

El presente trabajo de titulación se lo realizó con la ayuda de recursos humanos, materiales, técnicos y tecnológicos. A continuación se detalla los recursos empleados y su costo respectivo.

La tabla XXII, hace referencia a los recursos humanos empleados en el desarrollo del PFC, donde consta el asesoramiento continuo del director del proyecto permitiendo una correcta ejecución del proyecto.

**TABLA XXII. RECURSOS HUMANOS**

<b>Recursos</b>	<b>Horas</b>	<b>Costo Hora</b>	<b>Costo Total</b>
Freddy Chamorro (Investigador)	880	\$15.00	\$13200.00
Ing. Ana Lucía Colala (Asesor)	32	\$20.00	\$640
<b>Subtotal:</b>			<b>\$13840.00</b>

En la tabla XXIII, se presentan los recursos materiales, mismos que fueron utilizados durante el desarrollo del PFC.

**TABLA XXIII. RECURSOS MATERIALES**

Recursos	Cantidad	Valor Unitario	Total
Cartuchos de tinta negro/color	4	\$28.00	\$112.00
Resmas de papel bond A4	2	\$4.00	\$8.00
Copias	250	\$0.02	\$5.00
Anillados	5	\$1.00	\$5.00
Suministros de oficina (lápiz, borradores, perfiles, entre otros).	-	\$20.00	\$20.00
<b>Subtotal:</b>			<b>\$150.00</b>

En la tabla XXIV, se detalla los recursos técnicos y tecnológicos empleados para el cumplimiento de los objetivos planteados en el PFC. Estos recursos son de vital importancia puesto que permitieron desarrollar todo el proceso desde el análisis, la implementación, las pruebas de la aplicación y finalmente la publicación de la misma en sus respectivas tiendas para lo cual fue necesario adquirir las licencias de desarrollador para iOS y Android.

**TABLA XXIV. RECURSOS TÉCNICOS Y TECNOLÓGICOS**

Recursos	Cantidad	Valor Unitario	Total
<b>Recursos Software</b>			
Licencia iOS para desarrolladores	1	\$100.00	\$100.00
Licencia para desarrolladores de Google Play	1	\$25.00	\$25.00
<b>Recursos Hardware</b>			
Computador portátil Mac	1	\$1200.00	\$1200.00
Móvil con sistema operativo Android	1	\$250.00	\$250.00

Móvil con sistema operativo iOS	1	\$300.00	\$350.00
Impresora	40 horas	\$0.50	\$20.00
<b>Recursos Telecomunicaciones</b>			
Internet	800 horas	\$0.50	\$400.00
<b>Subtotal:</b>			<b>\$2345.00</b>

Finalmente en la tabla XXV, se presenta un resumen de los costes por cada uno de los recursos detallados anteriormente, así como el costo total del PFC.

**TABLA XXV. COSTO TOTAL DEL PROYECTO FIN DE CARRERA**

Rubro	Valor
Recursos Humanos	\$13840.00
Recursos Materiales	\$150.00
Recursos Técnicos y Tecnológicos	\$2345.00
<b>Subtotal</b>	<b>\$16335.00</b>
Imprevistos (10% del subtotal)	\$1633.50
<b>Total</b>	<b>\$17968.5</b>

## **H. Conclusiones**

Al culminar el proyecto de titulación se puede emitir las siguientes conclusiones:

- La aplicación móvil implementada, beneficia directamente a la comunidad universitaria involucrada y sociedad en general, ya que facilita la identificación de especies vegetales forestales nativas desde cualquier lugar donde se disponga de una conexión de datos, obteniendo la descripción, localización y fotografías de las mismas.
- Las pruebas de rendimiento y usabilidad realizadas a la aplicación móvil, permitieron demostrar que ésta es accesible y fácil de utilizar para los usuarios con conocimientos bajos, medios y altos en el manejo de tecnología móvil.
- Al permitir registrar especies vegetales no inventariadas en la aplicación móvil, los estudiantes de la carrera de Ingeniería Forestal y sociedad en general, tienen la facilidad de enriquecer la base de datos con especies vegetales forestales de la localidad y del país en general.
- El presente trabajo de titulación, aporta a la sociedad una herramienta tecnológica móvil, que permite difundir la información acerca de las especies vegetales forestales de la localidad con el mundo.
- Se consiguió solventar satisfactoriamente las dificultades que tienen los estudiantes de la carrera de Ingeniería Forestal de la UNL, concernientes a la identificación de especies vegetales forestales nativas.
- En la carrera de Ingeniería Forestal de la UNL, la aplicación móvil implementada tuvo mucha aceptación, debido a que no existe en el mercado actual una aplicación multiplataforma que permita realizar la identificación de especies vegetales forestales nativas por medio de una hoja de la misma.

## **I. Recomendaciones**

Luego de haber realizado el presente PFC, se puede emitir las siguientes recomendaciones:

- Se debe inventariar el mayor número de especies vegetales forestales nativas en la aplicación móvil, para obtener una base de datos robusta con mayor aceptación por los usuarios y la sociedad en general.
- Difundir la aplicación móvil a los estudiantes de la carrera de Ingeniería Forestal de la UNL y sociedad en general, con la finalidad de que éstos complementen sus conocimientos sobre las especies vegetales forestales nativas.
- Al registrar especies vegetales en la aplicación móvil, se recomienda que lo hagan solo usuarios con conocimientos medios y altos sobre especies vegetales forestales nativas con la finalidad de almacenar información confiable.
- Se recomienda seguir investigando el campo de estudio abordado, con la finalidad de implementar nuevas funcionalidades a la aplicación móvil que permitan una mayor aceptación de la misma.

## J. Bibliografía

- [1] H., Bicen and S. Kocakoyun, “**The Evaluation of The Most used Mobile Devices Applications by students**”, procedia - soc. Behav. sci., vol. 89, pp. 756-760, oct. 2013.
- [2] COLUMBIA UNIVERSITY, “**Portal de Leafsnap**”, [En línea], que se describe en detalle en: [www.leafsnap.com](http://www.leafsnap.com), accedido el 10 de octubre 2014.
- [3] VALENZUELA Pérez, S., “**Identificación de especies vegetales utilizando dispositivos móviles**”, Chile 2013, [En línea], que se describe en detalle en: [http://www.tesis.uchile.cl/bitstream/handle/2250/113941/cf-valenzuela\\_sp.pdf](http://www.tesis.uchile.cl/bitstream/handle/2250/113941/cf-valenzuela_sp.pdf), accedido el 10 de octubre del 2014.
- [4] L. Corral, A. Sillitti, and G. Succi, “**Mobile Multiplatform Development: An Experiment for Performance Anarysis**”, Procedia Compuf. Sci., vol. 10, pp.736-143, Jan. 2012.
- [5] Arkaitz Garro, “**Apache Cordova**”, 2014, [En línea], que se describe en detalle en: <http://www.arkaitzgarro.com/Apache Cordova/capitulo-2.html>, accedido el 15 de octubre del 2014.
- [6] **ThemeRoller**, 2015, [En línea], que se describe en detalle en: <http://themeroller.jquerymobile.com/>, accedido el 10 de enero del 2015.
- [7] J. Bertram and c. Kleiner, “**Secure web service clients on Mobile Devices**”, Procedia Comput. Sci., vol. 10, pp. 696-704, Jan.2012.
- [8] “**Data Modeling for APIs. Part 2: REST and JSON**”, 2012 Linked Data Orchestration Ltd., [En línea], que se describe en detalle en: <http://linkeddataorchestration.com/2014/01/28/data-modeling-for-apis-part-2-rest-and-json>, accedido el 15 de enero del 2015.
- [9] Aguirre, M., Bolaños, I., Dela Cruz, T., “**Proyecto de implementación de servidores web con uso de tecnologías móviles Tornado y MongoDB**”, 2011, Instituto Continental, Perú, [En línea], que se describe en detalle en: [http://www.instituto.continental.edu.pe/biblioteca/images/documentos/proyectos/proyecto\\_mongo\\_db.pdf](http://www.instituto.continental.edu.pe/biblioteca/images/documentos/proyectos/proyecto_mongo_db.pdf), accedido el 15 de enero del 2015.
- [10] Arévalo, C., “**Learning Analytics, Escalabilidad, Mejora de la Experiencia de Usuario e Internacionalizacion de la Herramienta de E-Learning FLEQ**”, 2014, Universidad Rey Juan Carlos, España, [En línea], que se describe en detalle en: <http://gsyc.urjc.es/~grex/pfcs/2014-cristina-lavado/CristinaLavado-LearningAnalyticsEscalabilidadMejoraExperienciaUsuarioFLEQ.pdf>, accedido el 16 de enero del 2015.



- [11] Cazorla, Justo. “**Desarrollo de una aplicación distribuida de control y gestión logística**”, 2008, Universidad Politécnica de Cartagena, España, [En línea], que se describe en detalle en: <http://repositorio.bib.upct.es/dspace/bitstream/10317/119/1/pfc2514.pdf>, accedido el 16 de enero del 2015.
- [12] Ortiz Laguna, J., “**Reconocimiento de objetos en una cocina con una webcam**”, Universidad Carlos III de Madrid 2009. [En línea], disponible en: [http://e-archivo.uc3m.es/bitstream/handle/10016/10007/PFC\\_FranciscoJavier\\_Garcia\\_Fernandez.pdf](http://e-archivo.uc3m.es/bitstream/handle/10016/10007/PFC_FranciscoJavier_Garcia_Fernandez.pdf), accedido el 10 de octubre del 2014.
- [13] Aguirre Mendoza, Z., “**Especies forestales Bosques secos del Ecuador**”, Quito 2012, [En línea], que se describe en detalle en: <http://www.ambiente.gob.ec/wp-content/uploads/downloads/2012/10/Bosques-Secos4.pdf>, accedido el 12 de octubre del 2014.
- [14] ECUADORFORESTAL, “**Planificación estratégica bosques nativos en el Ecuador**”, 2012, [En línea], que se describe en detalle en: [http://ecuadorforestal.org/wp-content/uploads/2013/03/PE\\_BN.pdf](http://ecuadorforestal.org/wp-content/uploads/2013/03/PE_BN.pdf), accedido el 16 de octubre del 2014.
- [15] Sobrado Malpartida, E. “**Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot**”, Perú 2013. [En línea], que se describe en detalle en: [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/68/SOBRADO\\_EDDIE\\_VISION\\_ARTIFICIAL\\_BRAZO\\_ROBOT.pdf](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/68/SOBRADO_EDDIE_VISION_ARTIFICIAL_BRAZO_ROBOT.pdf), accedido el 15 de diciembre del 2014.
- [16] Kuncheva L., “**Combining pattern classifiers**”, Wiley, 2004.
- [17] M. Nixon, A. S. Aguado: “**Feature extraction and image processing**”, 2<sup>o</sup> edición, Londres, 2008. [En línea] que se describe en detalle en: [http://www.scian.cl/portal/globals\\_file.php?CS=2960&ID=1345148829.7874&D=ON](http://www.scian.cl/portal/globals_file.php?CS=2960&ID=1345148829.7874&D=ON), accedido el 15 de noviembre del 2014.
- [18] Portal del Área Agropecuaria, “**Jardín botánico “Reinaldo Espinosa”**”, 2014. [En línea], que se describe en detalle en: <http://www.unl.edu.ec/agropecuaria/jardin-botanico/>, accedido el 10 de diciembre del 2014.
- [19] Aguirre Mendoza, Z., Gutiérrez Riofrio M., “**Jardín botánico “Reinaldo espinosa”**”, 2012, [en línea] que se describe en detalle en: <http://jbreinaldoespinosa.files.wordpress.com/2012/11/floranativa.pdf>, accedido el 10 de diciembre del 2014.

- [20] Aguirre Mendoza, Z., Gutiérrez Riofrio M., "Tríptico jardín botánico "Reinaldo Espinosa", 2013, que se describe en detalle en: <http://jbreinaldoespinosa.files.wordpress.com/2013/10/triptico.pdf> , accedido el 16 de diciembre del 2014.
- [21] RODRÍGUEZ Txema, "Métodos aplicables para el desarrollo de aplicaciones móviles" [en línea], 2012 disponible en: <http://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollode-aplicaciones-moviles> , fecha consulta: 05/07/2014.
- [22] Inmaculada Mercader S., "Desarrollo de Apps Multiplataforma: Un caso comparativo", 2014, UNIVERSIDAD POLITÉCNICA DE CARTAGENA. [En línea], que se describe en detalle en: <http://repositorio.bib.upct.es/dspace/bitstream/10317/4122/1/pfc5875.pdf> accedido el 5 de febrero del 2015.
- [23] K. Mohamed and D. Wijesekera, "Performance Analysis of Web Services on Mobile Devices", Procedia Comput. Sci., vol. 10, pp. 744-TSi, Jan. 2012.
- [24] RedHat, "OpenShift Online", 2015. [En línea], que se describe en detalle en: <https://www.openshift.com/products/online>, accedido el 10 de marzo del 2015.
- [25] J. Bertram and C. Kleiner, "SecureWeb Service Clients on Mobile Devices", Procedia Comput. Sci., vol. 10, pp. 696-704, Jan.2012.
- [26] Zeuxa, "Consultoría", 2015. [En línea], que se describe en detalle en: <http://www.zeuxa.com/actuamos/consultor%C3%AD>, accedido el 10 de enero del 2015.
- [27] "Consulta de notas UPNA", 2012, [En línea], que se describe en detalle en: <https://play.google.com/store/apps/details?id=com.sigma.mobile.target.upna>, accedido el 20 de enero del 2015.
- [28] Herrera, Janina, "Aplicación móvil para consultas académicas en la universidad Nacional de Loja", 2014, Universidad Nacional de Loja, pp. 18-19.
- [29] UNL, "Rendición de cuentas", 2014, Universidad Nacional de Loja, [En línea], que se describe en detalle en: [http://unl.edu.ec/sites/default/files/contenido/transparencia/informe\\_rector\\_2014\\_web.pdf](http://unl.edu.ec/sites/default/files/contenido/transparencia/informe_rector_2014_web.pdf), accedido el 10 de abril del 2015.

## K. Anexos

### Anexo 1: implementación de las funciones correspondientes a la iteración 1.

#### 1. Servidor web

En el directorio controlador de la ubicación raíz del servidor se ha creado el archivo ControladorUsuario.py, en el cual se implementaron los controladores respectivos para la identificación de usuario y registro del mismo.

A continuación se muestra la implementación de estos métodos:

```
import json
#hace referencia a la clase usuario nivel de comunicación con la BD
from clases.usuario import Usuario
#Coneccion con la base de datos
import coneccionBD
#metodo para la identificación de usuario
def buscarUsuario(correo,clave):
    usuario = Usuario.buscarUsuario(correo,clave,coneccionBD.coneccion())
    #response.headers['Content-Type'] = 'application/json'
    return json.dumps(usuario)
#metodo para el registro de usuario
def registrarUsuario(nombre,apellido,perfil,correo,clave):
    usuario = Usuario.buscarUsuario2(correo,coneccionBD.coneccion())
    #response.headers['Content-Type'] = 'application/json'
    if usuario:
        return json.dumps({'estadou':'El correo electronico del usuario a registrar ya
existe','registrado':0})
    else:
        Usuario({'nombre': nombre,'apellido':apellido, 'perfil':perfil,
'correo':correo,'clave':clave}).save(coneccionBD.coneccion())
        return json.dumps({'estadou':'Usuario Registrado Correctamente
;'),'registrado':1})
```

La clase usuario se ha implementado en el archivo usuario.py del directorio clases dentro de la ubicación raíz del servidor con los siguientes métodos para la búsqueda del usuario y registro del mismo. A continuación se muestra el código implementado:

```
class Usuario(object):
    def __init__(self, usuar=None):
        if usuar:
            self.nombre = usuar.get('nombre')
            self.apellido = usuar.get('apellido')
            self.perfil = usuar.get('perfil')
            self.correo = usuar.get('correo')
            self.clave = usuar.get('clave')
        def save(self, db):
            conn=db.cursor()
            conn.execute("INSERT INTO usuario(nombres,apellidos,email,clave,perfil)
values (%(nombre)s,%(apellido)s,%(correo)s,%(clave)s,%(perfil)s);",
            {'nombre':self.nombre,'apellido':self.apellido,
'correo':self.correo,'clave':self.clave,'perfil':self.perfil})
            db.commit()
            conn.close()
        @staticmethod
        def buscarUsuario(correo,clave,db):
            conn=db.cursor()
            conn.execute("SELECT nombres,apellidos,perfil FROM usuario WHERE
email=%s AND clave=%s",(correo,clave))
            usuarios = []
            for usuario in conn.fetchall():
                usuarios.append(Usuario.db_to_usuario(usuario))
            conn.close()
            return usuarios
        @staticmethod
        def buscarUsuario2(correo,db):
            conn=db.cursor()
            conn.execute("SELECT nombres,apellidos FROM usuario WHERE email=%s
",(correo,))
```

```

    usuarios = []
    for usuario in conn.fetchall():
        usuarios.append(Usuario.db_to_usuario(usuario))
    conn.close()
    return usuarios

@staticmethod
def db_to_usuario(usuario_result):
    if usuario_result:
        usuario =
    {'nombres':usuario_result[0],'apellidos':usuario_result[1],'perfil':usuario_result[2]}
        return usuario
    return None

```

## 2. Base de Datos

Para guardar los datos de los usuarios se creó una base de datos Postgres y una tabla llamada **usuario**, a continuación se muestra la estructura de la tabla creada:

```

CREATE TABLE usuario
(
    nombres character varying(15),
    apellidos character varying(15),
    email character varying(30) NOT NULL,
    clave character varying(30),
    perfil character varying(12),
    CONSTRAINT pk_correo PRIMARY KEY (email )
)
WITH (
    OIDS=FALSE
);

```

## 3. Aplicación móvil

Para la comunicación del servidor web con la aplicación móvil, para la autenticación de usuarios y el registro de los mismos se ha implementado las siguientes funciones del lado del cliente:

## Verificación de Usuario

```
function verificarUsuarioServidor(rcorreo,rclave){
    $('#footer_tabs').empty();
    archivoValidacion="http://servidorpy-
reconocimientoev.rhcloud.com/verificaUsuario";
    datos={"correo ":"+ rcorreo+" , "clave":"+rclave+"};
    alert(imgur_client_id);
    peticionxxx(archivoValidacion,datos);
    var obj = $.parseJSON(resp.usuarioV);
    if (obj.length == 0){
        toast("Usuario no válido (:");
        $.mobile.changePage("#entrar");
    }else{
        $.each(obj, function(index, element){
            var output = ['<div data-id="mainTab" data-role="navbar"><ul
id="footer_tabs" >'];
            output.push('<li><a href="" data-transition="slideup" data-
icon="search" onclick="buscarTodas();">Consultar</a></li>');
            output.push('<li><a href="" data-icon="eye" data-
transition="slideup" onclick="capturarImagenIdentificar();">Identificar</a></li>');
            if(element.perfil=="Colaborador"){
                output.push('<li><a href="" data-icon="camera"
onclick="iniciarLocalizacion();">Registrar</a></li>');
            }
            output.push('</ul></div>');
            $('[data-role="footer"]').html(output.join("")).trigger('create');
            $.mobile.changePage("#page-home");
            window.localStorage.usuario
=element.nombres[0].toUpperCase()+element.nombres.slice(1)+"
"+element.apellidos[0].toUpperCase() + element.apellidos.slice(1);
            cargarNombreUsuario(rcorreo);
        });
    }
    recordarUsuario(); }
```

## Anexo 2: implementación de las funciones correspondientes a la iteración 2.

### 1. Servidor web

Para el registro de especies vegetales en el servidor web, primero se envía desde la aplicación móvil la solicitud con los datos de la especie vegetal a registrar, posteriormente se verifica si la especie vegetal ya se encuentra registrada en la base de datos comparando el nombre científico de la especie, si es el caso se registra como un ejemplar de la especie ya registrada, en caso de que no se encuentre inventariada se registra la nueva especie con la información correspondiente.

A continuación se muestra los métodos implementados para la comunicación del servidor web con la aplicación móvil para el registro de especies vegetales:

```
class registraEspecie(RequestHandler):
    def set_default_headers(self):
        self.set_header("Access-Control-Allow-Origin", "")
        self.set_header("Content-Type", "application/json")
    def post(self):
        comun=self.get_argument('comun','')
        cientifico=self.get_argument('cientifico','')
        latitud=self.get_argument('latitud','')
        longitud=self.get_argument('longitud','')
        direccion=self.get_argument('direccion','')
        a=nuevaespecie(comun,cientifico,direccion,latitud,longitud)
        self.write(json.dumps({'planta':a}))
```

La respuesta emitida del servidor web, a la aplicación móvil se la realiza en formato JSON. Luego de recibir los datos desde la aplicación móvil, en el servidor se ha creado el archivo ControladorPlantas.py para verificar la información recibida. A continuación se muestra la implementación:

```
def nuevaespecie(comun,cientifico,dirurl,latitud,longitud):
    try:
        #recuperar especie por nombre
        plant = Especie.retrieve(coneccionBD.coneccion(), cientifico)
        if plant:
            if dirurl is not None:
```

```

        conn=coneccionBD.coneccion().cursor()
        Especie.add_sample(conn,cientifico,dirurl,latitud,longitud)
        coneccionBD.coneccion().commit()
        return json.dumps(plant)
    return json.dumps(plant)
else:
    return json.dumps(Especie({'name': científico,'common_name':comun, 'photo':
dirurl, 'latitud':latitud,'longitud':longitud}).save(coneccionBD.coneccion()))
except:
    return json.dumps({"estado":"0"})

```

Luego se registra el ejemplar de la especie o se registra la nueva especie dependiendo el caso para lo cual se creó el módulo plant.py en el directorio clases de la ubicación del servidor web, en la cual se implementó el registro de la especie o del ejemplar en la base de datos. A continuación se muestra el código implementado:

```

class Especie(object):
    def __init__(self, plant=None):
        if plant:
            self.nombre = plant.get('name')
            self.common_name = plant.get('common_name')
            self.photo = plant.get('photo')
            #self.distance = plant.get('distance',0)
            self.latitud=plant.get('latitud')
            self.longitud=plant.get('longitud')
            self.directorio=plant.get('photo')
        def save(self, db):
            conn=db.cursor()
            conn.execute("INSERT INTO especiev(nombre,nombre_comun,foto) values
            (%(name)s,%(common_name)s,%(photo)s);",
            {'name':self.nombre,'common_name':self.common_name, 'photo':self.photo})
            Especie.add_sample(conn, self.nombre, self.directorio,self.latitud,self.longitud)
            db.commit()
            conn.close()
            return Especie.retrieve(db,self.nombre)
        @staticmethod

```



```

def add_sample(conn, name, directorio,latitud,longitud):
    descriptors,ax,bx,ay,by=EFD(Threshold(directorio).process(), 50,
100).fourier_coefficients()
    conn.execute("INSERT INTO descriptores(name, descriptors,latitud,longitud)
values (%(name)s,%(descriptors)s,%(latitud)s,%(longitud)s);",
                {'name':name,'descriptors':descriptors,'latitud':latitud,'longitud':longitud})
@staticmethod
def retrieve(db, name):
    conn=db.cursor()
    conn.execute("SELECT nombre,nombre_comun,foto FROM especiev WHERE
nombre=%s LIMIT 1",(name,))
    plants = []
    for plant in conn.fetchall():
        plants.append(Especie._db_to_plant(plant))
    conn.close()
return plants

```

Además al guardar la especie vegetal se realiza la obtención de los descriptores de la misma con la finalidad de facilitar la posterior identificación de la misma. Primero se realiza la segmentación de la imagen de la hoja de la especie vegetal, para lo cual se ha creado el directorio reconocimiento dentro del directorio raíz del servidor web, con el módulo threshold.py, a continuación se muestra la implementación del código:

```

def process(self):
    pixels = self.image.load()
    beta = (0.5-0.16,0.19-0.7)
    beta0 = -0.5 * (0.25 + (0.19*0.19)-(0.16*0.16))-0.49)
    scale = 1000.0
    sat_threshold = 300
    val_threshold = 300
    hist_sat = [0]*1001
    hist_val = [0]*1001
    for x in range(self.image.size[0]):
        for y in range(self.image.size[1]):
            v = float(max(pixels[x,y]))
            saturation = (float(max(pixels[x,y])-min(pixels[x,y]))/v,v/255.0)

```

```

        hist_sat[int(saturation[0]*int(scale))] +=1
        hist_val[int(saturation[1]*int(scale))] +=1
    mu1 =
(float(hist_sat[:sat_threshold].index(max(hist_sat[:sat_threshold])))/scale,
    float(hist_val[val_threshold:].index(max(hist_val[val_threshold:])) +
val_threshold)/scale)
    mu2 =
(float(hist_sat[sat_threshold:].index(max(hist_sat[sat_threshold:]))+sat_threshold)/sc
ale,
    float(hist_val[:val_threshold].index(max(hist_val[:val_threshold])))/scale)
    beta = (mu2[0] - mu1[0],mu2[1]-mu1[1])
    beta0 = -0.5 * (pow(mu2[0],2) + pow(mu2[1],2) - pow(mu1[0],2)-pow(mu1[1],2))
    for x in range(self.image.size[0]):
        for y in range(self.image.size[1]):
            v = float(max(pixels[x,y]))
            saturation = (float(max(pixels[x,y])-min(pixels[x,y]))/v,v/255.0)
            pz1 = 1/(1+exp(beta0 + beta[0]*saturation[0] + beta[1]*saturation[1]))
            if pz1 > 0.5:
                pixels[x,y] = (0,0,0)
            else:
                pixels[x,y] = (255,255,255)
    #self.image.show()
    return self.image

```

Una vez segmentada la imagen se realiza la detección de bordes y se obtiene los descriptores de la hoja de la especie vegetal, para lo cual se ha implementado el siguiente código [17] migrado desde matlab que se encuentra en el módulo EFD.py, dentro del directorio reconocimiento, a continuación se muestra el código implementado:

```

class EFD(object):
    def __init__(self, image, ndescriptors, scale):
        self.image = CreateImageHeader(image.size, IPL_DEPTH_8U, 3)
        SetData(self.image, image.tostring())
        self.ndescriptors = ndescriptors
        self.scale = scale
        self.m = 2*ndescriptors

```

```

def get_curve(self):
    im = asarray(GetMat(self.image))
    im = cvtColor(im, COLOR_BGR2GRAY)
    # ret, im = threshold(im, 127,255, 0)
    contours, hierarchy = findContours(im, RETR_TREE,
CV_CHAIN_APPROX_NONE)
    max_cont = 0
    for i in range(len(contours))[1:]:
        if len(contours[i]) > len(contours[max_cont]):
            max_cont = i
    xy=[]
    # for xys in contours:
    for pos in contours[max_cont]:
        xy.append(tuple([int(coord) for coord in pos[0]]))
    xy_ = []
    step = len(xy)/self.m
    for i in range(self.m):
        xy_.append(xy[i*step])
    return xy_
def fourier_coefficients(self):
    xy = self.get_curve()
    x = [c[0] for c in xy]
    y = [c[1] for c in xy]
    t = 2*pi/self.m
    two_over_m = 2.0/self.m
    ax = [two_over_m*sum([x[i]*cos((k+1)*t*i) for i in range(self.m)])\
        for k in range(self.ndescriptors)]
    bx = [two_over_m*sum([x[i]*sin((k+1)*t*i) for i in range(self.m)])\
        for k in range(self.ndescriptors)]
    ay = [two_over_m*sum([y[i]*cos((k+1)*t*i) for i in range(self.m)])\
        for k in range(self.ndescriptors)]
    by = [two_over_m*sum([y[i]*sin((k+1)*t*i) for i in range(self.m)])\
        for k in range(self.ndescriptors)]
    coeffs = [sqrt((pow(ax[k],2) + pow(ay[k],2))/(pow(ax[0],2)+pow(ay[0],2)))]\

```

```

+ sqrt((pow(bx[k],2) + pow(by[k],2))/(pow(bx[0],2)+pow(by[0],2)))\
    for k in range(self.ndescriptors):-1]]
self.coeffs = coeoffs
return coeoffs, ax, ay, bx, by

```

## 2. Base de datos

Para la iteración 2, se ha implementado dos tablas (especiev, descriptores) en la base de datos como se muestra en el modelo de datos en la iteración 0, análisis y diseño. A continuación se muestra la estructura de las tablas creadas.

<pre> CREATE TABLE especiev ( nombre character varying(25) NOT NULL, nombre_comun character varying(25), foto text, CONSTRAINT especiev_pkey PRIMARY KEY (nombre )) WITH (OIDS=FALSE); </pre>	<pre> CREATE TABLE descriptores ( descriptors double precision[], latitud double precision, longitud double precision, name character varying(25) ) WITH (OIDS=FALSE); </pre>
---	---

## 3. Aplicación móvil

Para realizar el registro de una especie vegetal en la aplicación móvil primero se obtiene mediante la utilización del dispositivo móvil la geolocalización (latitud y longitud) de la especie vegetal a registrar, a continuación se muestra el método implementado:

```

function iniciarLocalizacion(){
    liberarCamposEspecie();
    $.mobile.changePage('#registrarEspecieV');
    $.mobile.loading( 'show', {
        text: 'Cargando..',
        textVisible: true,
        theme: 'a',
        html: "" });
    navigator.geolocation.getCurrentPosition(dispatch,geoError);
}
function dispatch(pos) {
    $('#latitud').attr("value",pos.coords.latitude);
    $('#longitud').attr("value",pos.coords.longitude);
    $.mobile.loading( "hide" );
}

```

A continuación se realiza la captura de la hoja de la especie vegetal haciendo uso de la cámara del dispositivo móvil, para lo cual se implementó el siguiente método:

```
function capturarImagenRegistrar() {
    navigator.camera.getPicture(exito, onFail, {
        quality: 100,
        targetWidth: 250,
        destinationType: Camera.DestinationType.FILE_URI
    }); }
function exito(imageURI) {
    var capturalImage = document.getElementById('capturalImage');
    capturalImage.style.display = 'block';
    capturalImage.src = imageURI;
}
function onFail(message) {
    toast("No capturó ninguna imagen con la cámara del dispositivo");
}
```

Para el registro de la especie vegetal se carga la imagen de la especie en el servidor gratuito IMGUR, y se obtiene la dirección Http, para su posterior envío al servidor para el registro. A continuación se muestra el código para subir la especie al servidor imgur y la petición de registro enviada al servidor web.

```
//id de autenticación para registro de especie en imgur
var imgur_client_id = "4f67a82e3a157c9";
function subirImagenRegistrarImgur() {
    if (conexionInternet()){
        imageURI=document.getElementById('capturalImage').src;
        if (imageURI.length == 0||$('#cientifico').val()=='0'||$('#comun').val()=='0') {
            toast('Campos obligatorios vacios');
            return false;
        }else{
            if (nombre_valido($('#cientifico').val())&&nombre_valido($('#comun').val())){
                $.mobile.loading( 'show', {
                    text: 'Subiendo Especie..',
                    textVisible: true,
                    theme: 'a',
```

```

        html: "" });
        _photo_uri=imageURI;
        var opts = new FileUploadOptions();
        opts.fileKey = "image";
        opts.fileName = _photo_uri.substr(_photo_uri.lastIndexOf("/")+1);
        opts.mimeType = "image/jpeg";
        opts.params = {"type":"file"};
        opts.headers={'Authorization':'Client-ID ' + imgur_client_id};
        var ft = new FileTransfer();
        ft.upload(imageURI, "https://api.imgur.com/3/upload.json", win4, fail3,
opts);
    }else{
        toast("Nombre común y/o científico no válidos");
    }
}
}
function win4(resp) {
    var link = jQuery.parseJSON(resp.response).data.link;
    subirEspecieServidorRegistro(link);
    $.mobile.loading( 'show', {
        text: 'Cargando Resultados..',
        textVisible: true,
        theme: 'b',
        html: ""
    }); }
function subirEspecieServidorRegistro(sfoto){
    $.mobile.loading( 'show', {
        text: 'Cargando..',
        textVisible: true,
        theme: 'a',
        html: ""
    });
    archivoValidacion
reconocimientoev.rhcloud.com/registrarEspecie";
}

```

```

var data=[];
scomun=$('#comun').val().toLowerCase();
cientifico=$('#cientifico').val().toUpperCase();
slongitud=$('#longitud').val();
slatitud=$('#latitud').val();
$.ajax({
  type    : "POST",
  url     : archivoValidacion,
  crossDomain: true,
  data    :
  {'comun':scomun,'cientifico':cientifico,'latitud':slatitud,'longitud':slongitud,'imagen':sfoto},
  dataType : 'json',
  success  : function(response) {
    var obj = $.parseJSON(response.planta);
    $.each(obj, function(index, element){
      if (element.estado==0) {
        $.mobile.loading( "hide" );
        toast("Imagen a identificar no valida");
        return false;
      }else
      {
        cargarResultados(element.nombre,element.comun,element.photo);
      } });
    $.mobile.loading( "hide" );
    //$mobile.showPageLoadingMsg(false);
    //$mobile.loading('hide');
  },
  error    : function(data) {
    toast('Servidor no disponible en estos momentos');
    $.mobile.loading( "hide" );
  }
}); }

```

## Anexo 3: implementación de las funciones correspondientes a la iteración 3.

### 1. Servidor web

Para la identificación de una especie vegetal el servidor recibe como datos la dirección HTTP pública de la imagen subida en el servidor imgur, a continuación se muestra el método de comunicación de la aplicación web con la aplicación móvil:

```
class buscarEspecieFot(RequestHandler):
    def set_default_headers(self):
        self.set_header("Access-Control-Allow-Origin", "**")
        self.set_header("Content-Type", "application/json")
    def post(self):
        filename=self.get_argument('especie',"")
        a=buscarEspecieF(filename)
        self.write(json.dumps({'planta':a}))
```

Posteriormente se envía la foto al controlador para su verificación y respuesta, a continuación se muestra el método implementado en el módulo ControladorPlanta.py:

```
def buscarEspecieF(directorioImagen):
    try:
        planta = Especie.buscar(directorioImagen,conexcionBD.conexcion())
        return json.dumps(planta)
    except:
        return json.dumps({"estado":"0"})
```

Para finalizar se busca las coincidencias más cercanas, para lo cual se ha implementado en el módulo plants.py del directorio clases, los métodos descritos a continuación:

```
@staticmethod
def buscar(directorioI,db):
    descriptors,ax,bx,ay,by=EFD(Threshold(directorioI).process(), 50,
100).fourier_coefficients()
    conn=db.cursor()
    conn.execute("""SELECT distinct especiev.nombre as
name,especiev.nombre_comun as comun,foto
```



```

        FROM especiev, (SELECT name, vdistance(descriptors ,cast(%s as
double precision[])) as distance
        FROM descriptores
        ORDER BY distance ASC
        ) as descr
        WHERE especiev.nombre = descr.name LIMIT 5;"; (descriptors,))
plants = []
for plant in conn.fetchall():
    plants.append(Especie._db_to_plant(plant))
conn.close()
return plants

```

## 2. Base de datos

Para la comparación de los descriptores de la especie a buscar con los registrados en la base de datos se ha implementado la función **vdistance**, en la base de datos, la misma que se describe a continuación:

```

CREATE OR REPLACE FUNCTION vdistance(double precision[], double precision[])
  RETURNS double precision AS
  $BODY$
  DECLARE
    ret_sum double precision;
    sum1 double precision;
  BEGIN
    ret_sum := 0;
    for i in 1..array_length($1,1) LOOP
      sum1:= $1[i]-$2[i];
      ret_sum=ret_sum+pow(sum1,2);
    END LOOP;
    return sqrt(ret_sum);
  END
  $BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100;

```

### 3. Aplicación móvil

En la aplicación móvil para la comunicación con el servidor móvil primero se ha implementado una función para la captura de imagen por medio de la cámara del dispositivo, similar a la del anexo 2, sección 3. A continuación se realizó el método para subir la imagen al servidor de imágenes imgur, similar al método del anexo 2, sección 3, y para finalizar se realizó el método para comunicación con el servidor y procesamiento de respuesta, se encuentra implementada en el archivo especievegetal.js, que se describe a continuación:

```
function subirEspecieServidor(sfoto){
    $("#lsbuscar").empty();
    archivoValidacion                ="http://servidorpy-
reconocimientoev.rhcloud.com/buscarEspecieFoto";
    var data=[];
    $.ajax({
        type    : "POST",
        url     : archivoValidacion,
        crossDomain: true,
        data    : {'especie':sfoto},
        dataType : 'json',
        success  : function(response) {
            var obj = $.parseJSON(response.planta);
            bandera=true;
            $.each(obj, function(index, element){
                if (element.estado==0) {
                    $.mobile.loading( "hide" );
                    toast("Imagen no valida");
                    bandera=false;
                }else{
                    var $a  ='<a id="'+element.nombre+'"><h3>'+element.nombre+'</h3><p>'+element.comun+'</p></a>';
                    var $li = $('<li/>', {
                        click: function () {
                            cargarResultados(element.nombre,element.comun,element.photo);
                        }
                    });
                }
            });
        }
    });
}
```

```
    }
    }).append($a);
    $('#lsbuscar').append($li);
  }
});
$.mobile.loading( "hide" );
if (bandera) {
  $.mobile.changePage('#buscar');
  $("#lsbuscar").listview('refresh');
}else{
  return false;
}
},
error : function(data) {
  toast('Servidor no disponible en estos momentos');
  $.mobile.loading( "hide" );
}
});
}
```

## Anexo 4: implementación de las funciones correspondientes a la iteración 4.

### 1. Servidor web

En la iteración 4 se ha implementado 2 metodos para las consults de especies vegetales desde la aplicación móvil, el primer método para la búsqueda y recuperación desde la base de datos de todas las especies vegetales registradas, y el segundo método para consultar la localización de las muestras registradas en la aplicación móvil, la respuesta se obtiene en formto JSON para ambos métodos. A continuación se muestra los métodos implementados:

```
class buscarTodos(RequestHandler):
    def set_default_headers(self):
        self.set_header("Access-Control-Allow-Origin", "")
        self.set_header("Content-Type", "application/json")
    def post(self):
        a=search()
        self.write(json.dumps({'plantas':a}))

class buscarPuntosInteres(RequestHandler):
    def post(self):
        nombre=self.get_argument("planta","")
        a=puntosInteres(nombre)
        self.write({'planta':a})
```

En el controlador del servidor referente a especies vegetales ControladorPlantas.py se ha implementado los siguientes métodos para la verificación de la solicitud enviada desde la aplicación móvil. A continuación se muestra la implementación:

```
def search():
    plants = Especie.buscar2(coneccionBD.coneccion())
    #response.headers['Content-Type'] = 'application/json'
    return json.dumps(plants)
def puntosInteres(nombre):
    plants = Especie.buscarPuntosInteres(nombre,coneccionBD.coneccion())
    #response.headers['Content-Type'] = 'application/json'
    return json.dumps(plants)
```

En la clase Especie ubicada en el módulo plant.py del directorio clases en el servidor web, se ha implementado los siguientes métodos para la recuperación de la información solicitada desde la aplicación móvil, y formateo en JSON:

```
@staticmethod
def buscar2(db):
    conn=db.cursor()
    conn.execute("SELECT nombre,nombre_comun,foto FROM especiev")
    #conn.close()
    plants = []
    for plant in conn.fetchall():
        plants.append(Especie._db_to_plant(plant))
    conn.close()
    return plants

@staticmethod
def buscarPuntosInteres(nombre,db):
    conn=db.cursor()
    conn.execute("SELECT name,latitud,longitud FROM descriptores WHERE
name=%s",(nombre,))
    #conn.close()
    plants = []
    for plant in conn.fetchall():
        plants.append(Especie.busqueda_ubicacion_plant_bd(plant))
    conn.close()
    return plants

@staticmethod
def busqueda_ubicacion_plant_bd(plant_result):
    if plant_result:
        plant = {'nombre':plant_result[0],'latitud':plant_result[1],
'longitud':plant_result[2]}
        if len(plant_result)>5:
            plant['distance'] = plant_result[5]
        return plant
    return None
```

## 2. Aplicación móvil

En la aplicación móvil para realizar la solicitud de consulta de especies vegetales al servidor y el procesamiento de la respuesta se han implementado en siguientes métodos, ubicados en el archivo especievegetal.js de la aplicación móvil:

```
function buscarTodas(){
    $("#lsbuscar").empty();
    $.mobile.changePage('#buscar');
    if (conexionInternet()){
        $.mobile.loading( 'show', {
            text: 'Cargando..',
            textVisible: true,
            theme: 'a',
            html: ""
        });
        archivoValidacion = "http://servidorpy-
reconocimientoev.rhcloud.com/buscarTodos";
        var data=[];
        $.ajax({
            type : "POST",
            url : archivoValidacion,
            crossDomain: true,
            data : {'username' : 'subin'},
            dataType : 'json',
            success : function(response) {
                var obj = $.parseJSON(response.plantas);
                //console.log(response.plantas);
                $.each(obj, function(index, element){
                    var $a ='<a id="'+element.nombre+'"><h3>'+element.nombre+'</h3><p>'+element.comun+'</p></a>';

                    var $li = $('<li/>', {
                        click: function () {
                            cargarResultados(element.nombre,element.comun,element.photo);
```

```

        }
        }).append($a);
        $('#lsbuscar').append($li);
    });
    $("#lsbuscar").listview('refresh');
    $.mobile.loading( "hide" );
},
error : function(data) {
    $.mobile.loading( "hide" );
    toast('Servidor no disponible en estos momentos');
}
});
}
}
function cargarResultados (nombre,comun,foto) {
    var resultadoImage = document.getElementById('resultadoImage');
    resultadoImage.style.display = 'block';
    resultadoImage.src = foto;
    $('#rcientifico').attr('value',nombre);
    $('#rcomun').attr('value',comun);
    $.mobile.changePage('#resultado1');
    cargarMapa();
}

```

En lo que respecta a la información detallada de cada especie vegetal, se ha realizado la petición al servidor web para obtener la latitud y longitud de la ubicación de cada muestra vegetal en el mapa según la especie seleccionada, además de una búsqueda interna de la especie vegetal en la Wikipedia para proveer al usuario de información detallada de la misma, los métodos implementados se encuentran en el archivo especievegetal.js, geolocation.js, de la aplicación móvil, a continuación se muestra la implementación:

```

function cargarDatosWikipedia(){
    if (conexionInternet()){
        plant=$('#rcientifico').val().toLowerCase();
        $.ajax({

```

```

    type: "GET",
    url:
"http://es.wikipedia.org/w/api.php?action=parse&format=json&prop=text&page="+plant+"&callback=?",
    contentType: "application/json; charset=utf-8",
    async: false,
    dataType: "json",
    success: function (data, textStatus, jqXHR) {
    var markup = data.parse.text["*"];
    re = /img.*?src="(.*?)"/g
    while( match = re.exec(markup)) {
    if(match[1].indexOf("svg")==-1){
    var $a = '</div>').html(markup);
    $('#article').html($(blurb).find('p'));
    },
    error: function (errorMessage) {
    toast("Servidor no disponible intente más tarde");
    }
    });
}}

function cargarMapa(){
    if (conexionInternet()){
        //$.mobile.changePage("#mapa");
        $.mobile.loading( 'show', {
            text: 'Cargando..',
            textVisible: true,

```



```

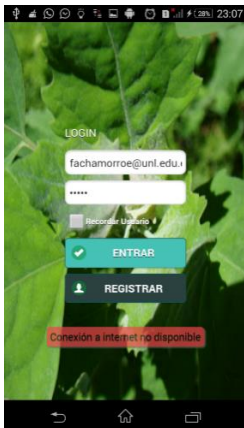

    theme: 'a',
    html: ""
  });
  archivoValidacion                                ="http://servidorpy-
reconocimientoev.rhcloud.com/buscarPuntosInteres";
  $.ajax({
    type      : "POST",
    url       : archivoValidacion,
    crossDomain: true,
    data      : {'planta' : $('#rcientifico').val()},
    dataType  : 'json',
    success   : function(response) {
      GoogleMap();
      var obj = $.parseJSON(response.planta);
      $.each(obj, function(index, element){
        cargarPuntosInteres(element.latitud,element.longitud,element.nombre);
      });
      $.mobile.loading( "hide" );
    },
    error     : function(data) {
      toast('Servidor no disponible en estos momentos');
      $.mobile.loading( "hide" );
    }
  });
}
}

```

## Anexo 5: Pruebas funcionales de la Aplicación Móvil.

1. **Requerimiento 001:** La aplicación móvil permitirá al Usuario ingresar a la aplicación mediante el correo electrónico y clave, previamente registrados.

### 1.1 Ejecución en Android.

<p><b>1. Criterio de aceptación 1 y 2</b></p> 	<p><b>2. Criterio de aceptación 3</b></p> 
<p><b>3. Criterio de aceptación 4</b></p> 	<p><b>4. Criterio de aceptación 5</b></p> 
<p><b>5. Criterio de aceptación 6</b></p> 	

## 1.2 Ejecución en iOS.

<h3>1. Criterio de aceptación 1 y 2</h3>  <p>A screenshot of the login screen. The background is a close-up of green leaves. The text 'LOGIN' is at the top. Below it is a text input field containing 'fachamorroe@unl.ec', followed by a password input field with six dots. A checked checkbox labeled 'Recordar usuario' is below the password field. Two buttons are at the bottom: a green 'ENTRAR' button with a checkmark icon and a dark grey 'REGISTRAR' button with a person icon.</p>	<h3>2. Criterio de aceptación 3</h3>  <p>A screenshot of the login screen. The background is a close-up of green leaves. The text 'LOGIN' is at the top. Below it is a text input field containing 'fachamorroe@unl.ec', followed by a text input field labeled 'Clave de usuario'. A checked checkbox labeled 'Recordar usuario' is below the password field. Two buttons are at the bottom: a green 'ENTRAR' button with a checkmark icon and a dark grey 'REGISTRAR' button with a person icon. A red error message 'Campos obligatorios vacíos' is at the bottom.</p>
<h3>3. Criterio de aceptación 4</h3>  <p>A screenshot of the login screen. The background is a close-up of green leaves. The text 'LOGIN' is at the top. Below it is a text input field containing 'fachamorroe@unl.ec', followed by a password input field with six dots. A checked checkbox labeled 'Recordar usuario' is below the password field. Two buttons are at the bottom: a green 'ENTRAR' button with a checkmark icon and a dark grey 'REGISTRAR' button with a person icon. A red error message 'Conexión a internet no disponible' is at the bottom.</p>	<h3>4. Criterio de aceptación 5</h3>  <p>A screenshot of the login screen. The background is a close-up of green leaves. The text 'LOGIN' is at the top. Below it is a text input field containing 'fachamorroe@unl.ec', followed by a password input field with six dots. A checked checkbox labeled 'Recordar usuario' is below the password field. Two buttons are at the bottom: a green 'ENTRAR' button with a checkmark icon and a dark grey 'REGISTRAR' button with a person icon. A red error message 'Usuario no válido :(' is at the bottom.</p>
<h3>5. Criterio de aceptación 6</h3>  <p>A screenshot of the app's main screen. The background is a close-up of green leaves. At the top, there is a header with a menu icon, the text 'Especies Vegetales Nati', and a home icon with the text 'Inicio'. Below the header is a large image of a plant with white flowers. Underneath the image is the text 'Reconoce especies vegetales por sus hojas.' and a quote: '“ Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja.' At the bottom, there is a navigation bar with three icons and labels: a magnifying glass icon labeled 'Consultar', a leaf icon labeled 'Identificar', and a camera icon labeled 'Registrar'.</p>	

**2. Requerimiento 002:** La aplicación móvil permitirá al Usuario registrarse en la aplicación con sus datos personales (nombres, apellidos, correo electrónico, clave y perfil de usuario).

**1.1 Ejecución en Android.**

<p><b>1. Criterio de aceptación 1</b></p> 	<p><b>2. Criterio de aceptación 2</b></p> 
<p><b>3. Criterio de aceptación 3</b></p> 	<p><b>4. Criterio de aceptación 4</b></p> 
<p><b>5. Criterio de aceptación 5</b></p> 	

## 2.2 Ejecución en iOS.

### 1. Criterio de aceptación 1

Registro de Usuario

Prueba

Apellidos(\*):  
usuario122

Clave(\*):  
.....

Correo electronico(\*):  
pruebaunl@gmail.com

Perfil Usuario(\*):  
Consultor

Nombres y/o Apellidos no validos :(

Registrar

### 2. Criterio de aceptación 2

Registro de Usuario

Nombres(\*):  
Prueba

Apellidos(\*):  
usuario

Clave(\*):  
.....

Correo electronico(\*):  
pruebaunl@gmail.com e

Perfil Usuario(\*):  
Consultor

La dirección de correo pruebaunl@gmail.com o es incorrecta.

Registrar

### 3. Criterio de aceptación 3

Registro de Usuario

Nombres(\*):  
Ingrese su nombre

Apellidos(\*):  
Ingrese su apellido

Clave(\*):  
Ingrese su clave

Correo electronico(\*):  
Ingrese su correo

Perfil Usuario(\*):  
Consultor

Campos obligatorios vacios :(

Registrar

### 4. Criterio de aceptación 4

Registro de Usuario

Correo electronico(\*):  
pruebaunl@gmail.com

Perfil Usuario(\*):  
Consultor

Registrar

Consultor  
Colaborador

### 5. Criterio de aceptación 5

LOGIN

Correo electronico

Clave de usuario

Recordar Usuario



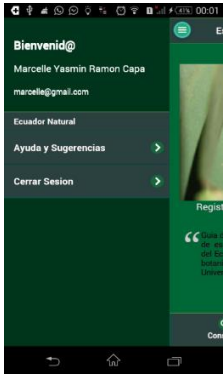
ENTRAR

REGISTRAR

Usuario Registrado Correctamente :)

**3. Requerimiento 003:** La aplicación móvil permitirá al usuario visualizar las opciones disponibles en la aplicación de acuerdo a su perfil de usuario.

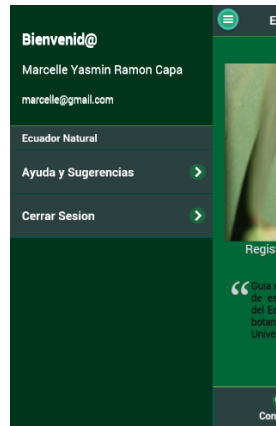
**3.1 Ejecución en Android.**

<p><b>1. Criterio de aceptación 1 (Perfil Colaborador)</b></p> 	<p><b>2. Criterio de aceptación 1 (Perfil Consultor)</b></p> 
<p><b>3. Criterio de aceptación 2</b></p> 	

**3.2 Ejecución en iOS.**

<p><b>1. Criterio de aceptación 1 (Perfil 'Colaborador')</b></p> 	<p><b>2. Criterio de aceptación 1 ('Perfil Consultor')</b></p> 
--	--

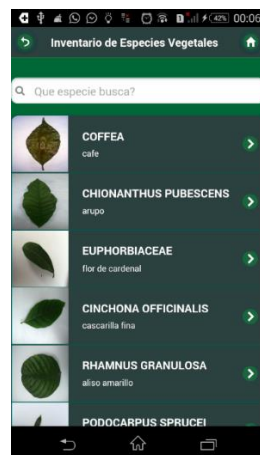
### 3. Criterio de aceptación 3



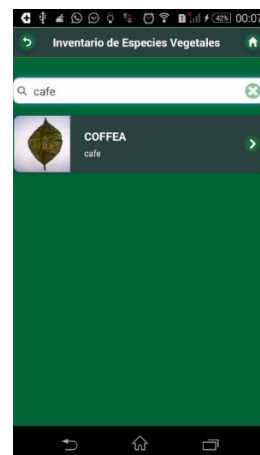
4. **Requerimiento 004:** La aplicación móvil permitirá al usuario consultar todas las especies vegetales registradas en la aplicación.

#### 4.1 Ejecución en Android.

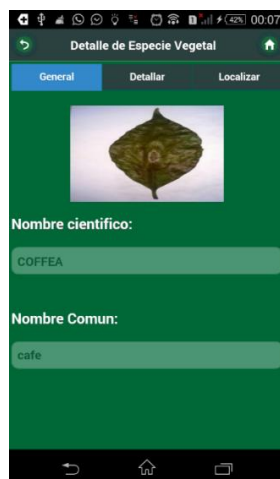
### 1. Criterio de aceptación 1 y 2



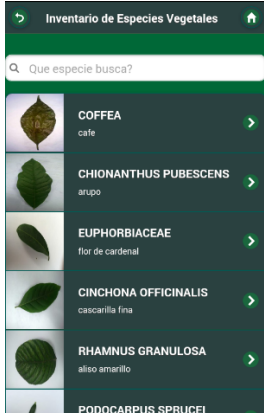
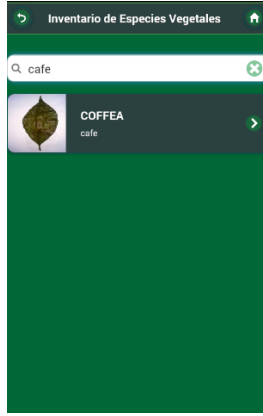

### 2. Criterio de aceptación 2



### 3. Criterio de aceptación 3





## 4.2 Ejecución en iOS.

<p><b>1. Criterio de aceptación 1 y 2</b></p> 	<p><b>2. Criterio de aceptación 2</b></p> 
<p><b>3. Criterio de aceptación 3</b></p> 	

**5. Requerimiento 005:** La aplicación móvil permitirá al usuario identificar las especies vegetales por medio de una hoja de la especie vegetal en referencia a las especies vegetales registradas en la aplicación.

## 5.1 Ejecución en Android.

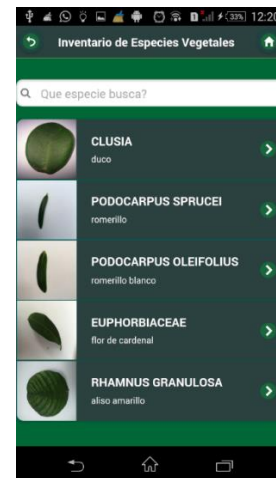
<p><b>1. Criterio de aceptación 1</b></p> 	<p><b>2. Criterio de aceptación 2,3 y 4</b></p> 
---	---



### 3. Criterio de aceptación 5

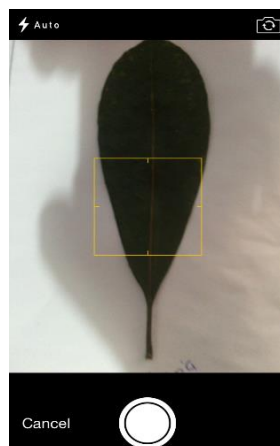


### 4. Criterio de aceptación 6



## 5.2 Ejecución en iOS.

### 1. Criterio de aceptación 1



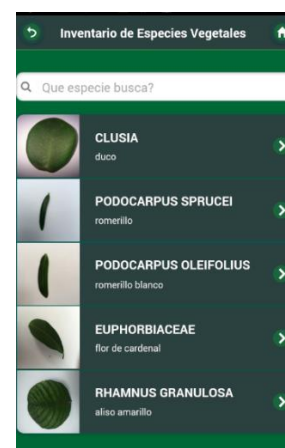
### 2. Criterio de aceptación 2,3 y 4



### 3. Criterio de aceptación 5

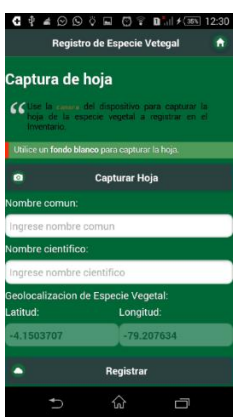



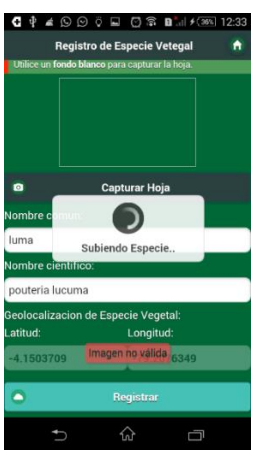


### 4. Criterio de aceptación 6



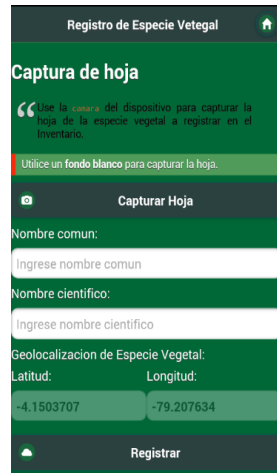
**6. Requerimiento 006:** La aplicación móvil permitirá al usuario con perfil "Colaborador", registrar las especies vegetales por medio de una hoja de la especie vegetal, con nombre común y científico, además de las coordenadas geográficas del registro de la especie vegetal.

**6.1 Ejecución en Android.**

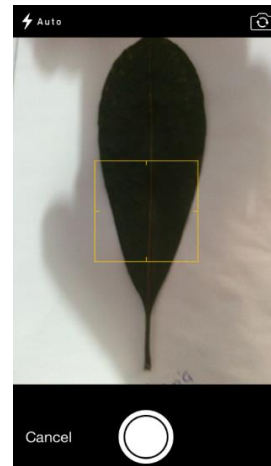
<p><b>1. Criterio de aceptación 1</b></p> 	<p><b>2. Criterio de aceptación 2</b></p> 
<p><b>3. Criterio de aceptación 3</b></p> 	<p><b>4. Criterio de aceptación 4</b></p> 
<p><b>5. Criterio de aceptación 5</b></p> 	

## 6.2 Ejecución en iOS.

### 1. Criterio de aceptación 1



### 2. Criterio de aceptación 2



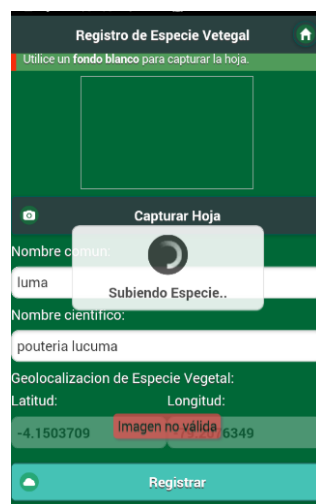
### 3. Criterio de aceptación 3



### 4. Criterio de aceptación 4

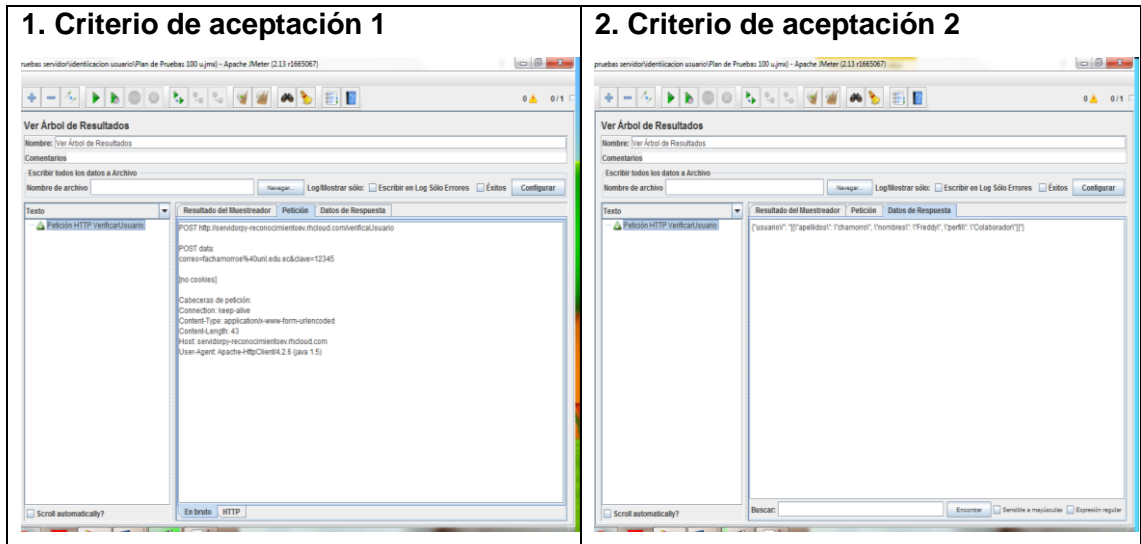


### 5. Criterio de aceptación 5



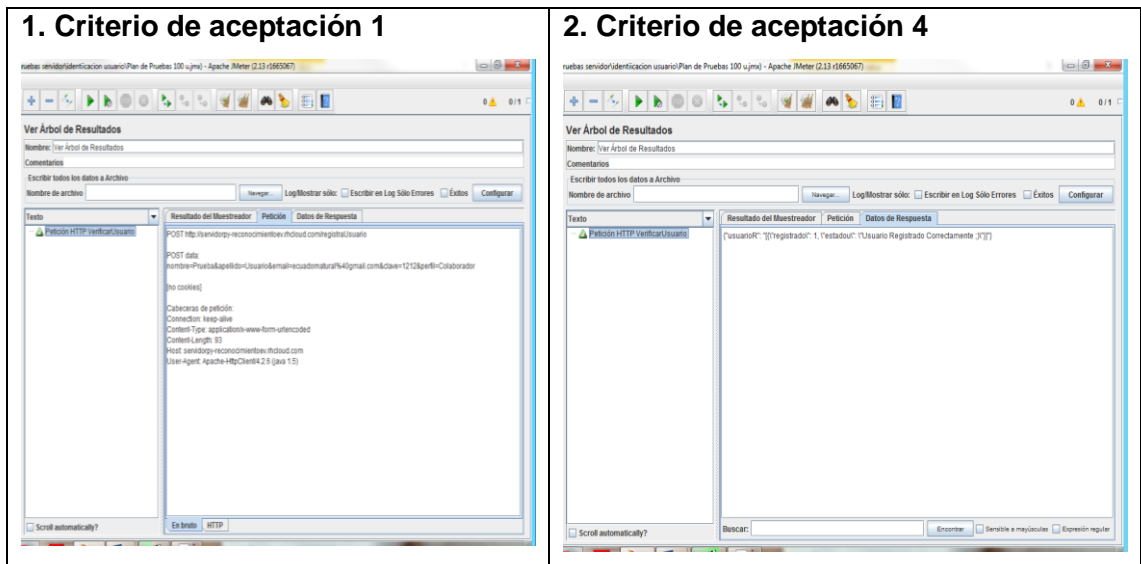
**7. Requerimiento 008:** EL servidor responde a la solicitud de verificación de usuario emitida por la aplicación móvil.

**7.1 Ejecución de prueba con Jmeter.**



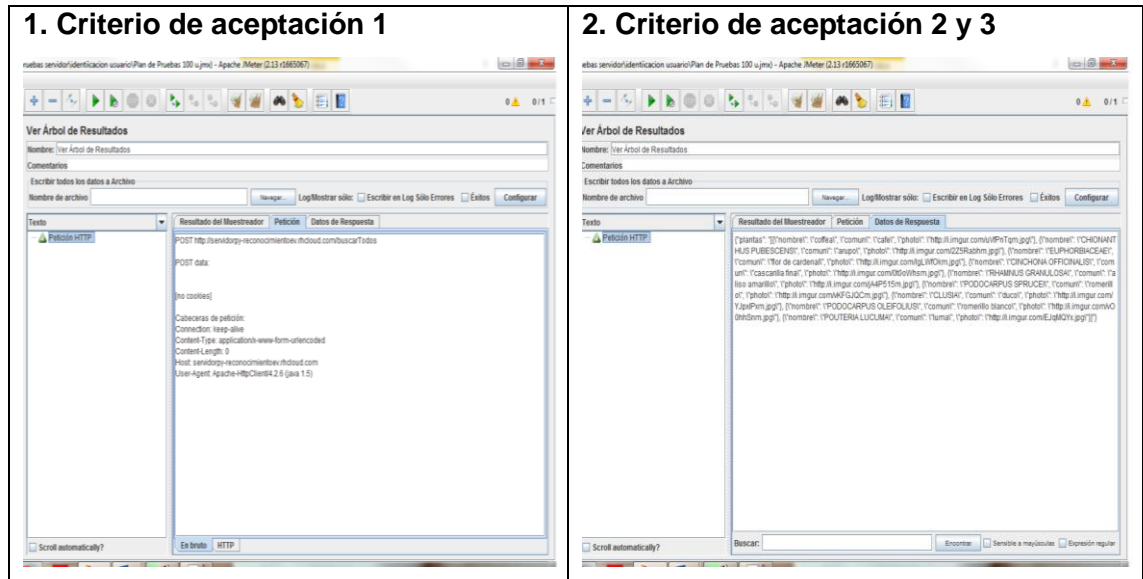
**8. Requerimiento 009:** El servidor web recibe la petición de registro de usuario desde la aplicación móvil con los datos personales, para el registro en la Base de datos.

**8.1 Ejecución de prueba con Jmeter.**



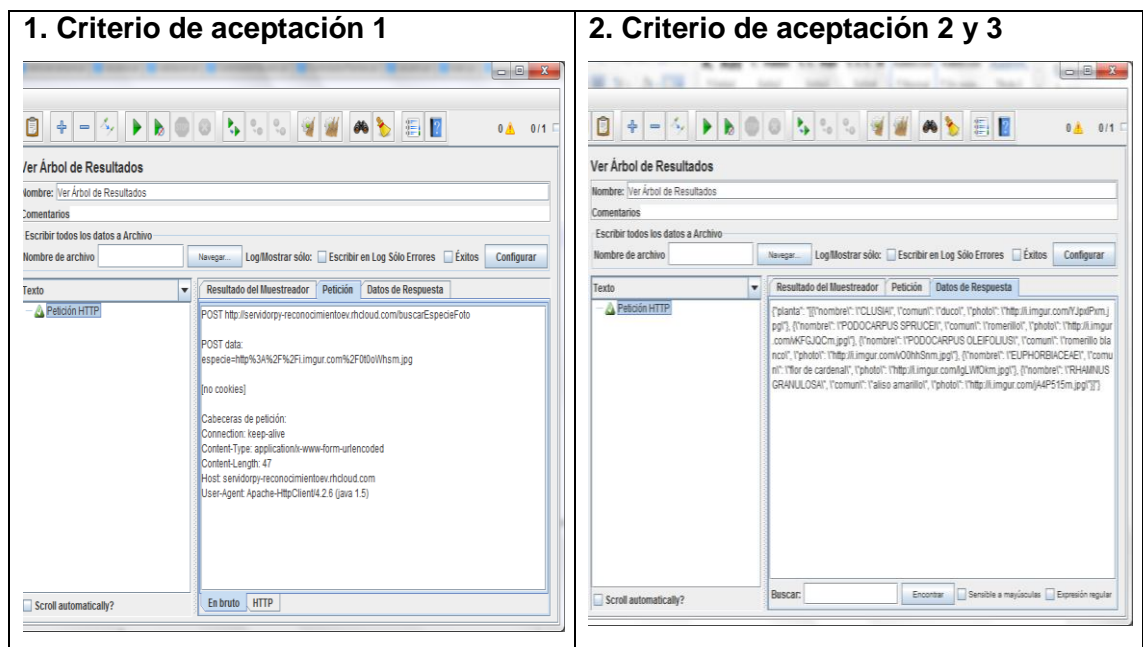
**9. Requerimiento 0010:** EL servidor responde a la solicitud de consulta de todas las especies vegetales registradas emitida por la aplicación móvil.

**9.1 Ejecución de prueba con Jmeter.**



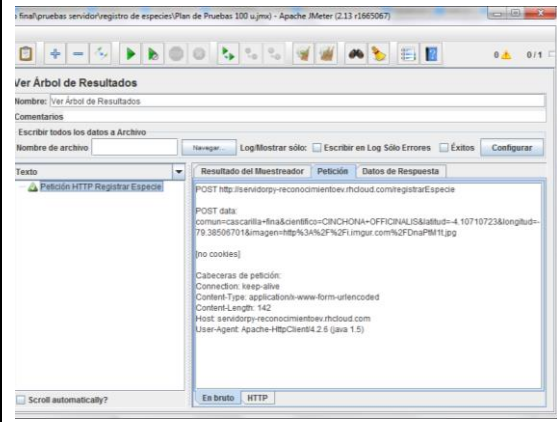
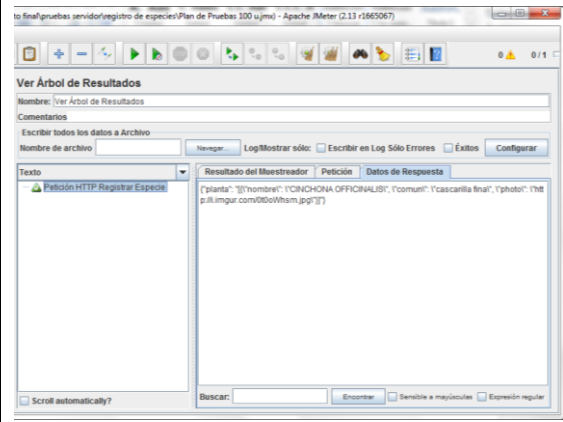
**10. Requerimiento 0011:** El servidor responde a la solicitud de identificación de la especie vegetal emitida por la aplicación móvil.

**10.1 Ejecución de prueba con Jmeter.**



**11. Requerimiento 0012:** El servidor responde a la solicitud de registro de la especie vegetal, y guarda la especie vegetal emitida por la aplicación móvil en la Base de Datos.

### 11.1 Ejecución de prueba con Jmeter.

1. Criterio de aceptación 1	2. Criterio de aceptación 2, 3 y 4
 <p>Ver Árbol de Resultados</p> <p>Nombre: Ver Árbol de Resultados</p> <p>Comentarios</p> <p>Escribir todos los datos a Archivo</p> <p>Nombre de archivo <input type="text"/> Navegar Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar</p> <p>Texto</p> <p>Petición HTTP Registrar Especie</p> <p>Resultado del Maestreador Petición Datos de Respuesta</p> <pre>POST http://sereidorp-reconocimientoev.rhcloud.com/registroEspecie Content-Type: application/x-www-form-urlencoded Content-Length: 142 Host: sereidorp-reconocimientoev.rhcloud.com User-Agent: Apache-HttpClient/4.2.5 (java 1.5)</pre> <p>Scroll automatically? <input type="checkbox"/> En bruto HTTP</p>	 <p>Ver Árbol de Resultados</p> <p>Nombre: Ver Árbol de Resultados</p> <p>Comentarios</p> <p>Escribir todos los datos a Archivo</p> <p>Nombre de archivo <input type="text"/> Navegar Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar</p> <p>Texto</p> <p>Petición HTTP Registrar Especie</p> <p>Resultado del Maestreador Petición Datos de Respuesta</p> <pre>["planta", {"nombre": "CINCHONA OFFICINALIS", "comun": "Cascarilla fina", "photo": "http://imgur.com/0BoWhsm.jpg"}]</pre> <p>Scroll automatically? <input type="checkbox"/> Buscar: <input type="text"/> Encontrar <input type="button"/> Sensible a mayúsculas <input type="checkbox"/> Expresión regular</p>

## Anexo 6: Encuesta



### UNIVERSIDAD NACIONAL DE LOJA

#### Área de la Energía, las Industrias y los Recursos Naturales No Renovables

#### *Carrera de Ingeniería en Sistemas*

#### *Encuesta para la carrera de Ingeniería Forestal*

La presente encuesta tiene como finalidad obtener información relevante sobre el funcionamiento de la aplicación móvil EcuadorNatural, misma que servirá para la sustentación del Trabajo de Titulación denominado " Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja". Por ello se solicita contestar las siguientes preguntas.

**1. ¿En qué sistema operativo instaló la aplicación móvil EcuadorNatural?**

Android ( )    iOS ( )

**2. ¿Qué versión de sistema operativo tiene su teléfono?**

Respuesta: .....

**3. ¿Qué marca y modelo tiene su teléfono?**

Respuesta: .....

**4. ¿El tiempo de respuesta al realizar la identificación de especie vegetales fue?**

Excelente ( )            Bueno ( )            Regular ( )            Malo ( )

**5. La aplicación móvil permite la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja"**

Si ( )    No ( )

**6. ¿Al realizar la identificación de especies vegetales forestales, obtiene los resultados esperados?**

Si ( )    No ( )

**7. ¿La aplicación móvil permite el registro y la autenticación de usuarios?**

Si ( )    No ( )

**8. ¿La aplicación móvil permite consultar todas las especies vegetales registradas?**

Si ( )    No ( )

**9. ¿La aplicación móvil permite registrar especies vegetales para usuarios con perfil “Colaborador”?**

Si ( )

No ( )

**10. ¿Considera que la aplicación móvil facilita la identificación de especies vegetales forestales nativas?**

Si ( )

No ( )

**GRACIAS POR SU COLABORACIÓN**



# Anexo 7: Resultados de la Encuesta

La encuesta fue aplicada a una muestra de 30 estudiantes del AARNR de la Universidad Nacional de Loja. A continuación se presentan los resultados obtenidos:

9. ¿La aplicación móvil permite registrar especies vegetales para usuarios con perfil "Colaborador"?
- Si (  ) No ( )
10. ¿Considera que la aplicación móvil facilita la identificación de especies vegetales forestales nativas?
- Si (  ) No ( )

GRACIAS POR SU COLABORACIÓN



UNIVERSIDAD NACIONAL DE LOJA  
 Área de la Energía, las Industrias y los Recursos Naturales No Renovables  
 Carrera de Ingeniería en Sistemas  
 Encuesta para la carrera de Ingeniería Forestal

La presente encuesta tiene como finalidad obtener información relevante sobre el funcionamiento de la aplicación móvil EcuadorNatural, misma que servirá para la sustentación del Trabajo de Titulación denominado " Guía de campo móvil para la identificación de especies vegetales forestales natvas del Ecuador tomando como piloto el jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja". Por ello se solicita contestar las siguientes preguntas.

1. ¿En qué sistema operativo instaló la aplicación móvil EcuadorNatural?
- Android ( ) IOS (  )
2. ¿Qué versión de sistema operativo tiene su teléfono?
- Respuesta: 8.3
3. ¿Qué marca y modelo tiene su teléfono?
- Respuesta: iPhone 5s
4. ¿El tiempo de respuesta al realizar la identificación de especie vegetales fue?
- Excelente ( ) Bueno (  ) Regular ( ) Malo ( )
5. La aplicación móvil permite la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el Jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja"
- Si (  ) No ( )
6. ¿Al realizar la identificación de especies vegetales forestales, obtiene los resultados esperados?
- Si (  ) No ( )
7. ¿La aplicación móvil permite el registro y la autenticación de usuarios?
- Si (  ) No ( )
8. ¿La aplicación móvil permite consultar todas las especies vegetales registradas, e identificar especies vegetales, por medio de una hoja de la misma?
- Si (  ) No ( )



UNIVERSIDAD NACIONAL DE LOJA  
 Área de la Energía, las Industrias y los Recursos Naturales No Renovables  
 Carrera de Ingeniería en Sistemas  
 Encuesta para la carrera de Ingeniería Forestal

La presente encuesta tiene como finalidad obtener información relevante sobre el funcionamiento de la aplicación móvil EcuadorNatural, misma que servirá para la sustentación del Trabajo de Titulación denominado " Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja". Por ello se solicita contestar las siguientes preguntas:

1. ¿En qué sistema operativo instaló la aplicación móvil EcuadorNatural?  
 Android       iOS ( )
2. ¿Qué versión de sistema operativo tiene su teléfono?  
 Respuesta: 4.2
3. ¿Qué marca y modelo tiene su teléfono?  
 Respuesta: Samsung Galaxy S3
4. ¿El tiempo de respuesta al realizar la identificación de especie vegetales fue?  
 Excelente ( )      Bueno       Regular ( )      Malo ( )
5. La aplicación móvil permite la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el Jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja"  
 Si       No ( )
6. ¿Al realizar la identificación de especies vegetales forestales, obtiene los resultados esperados?  
 Si       No ( )
7. ¿La aplicación móvil permite el registro y la autenticación de usuarios?  
 Si       No ( )
8. ¿La aplicación móvil permite consultar todas las especies vegetales registradas, e identificar especies vegetales, por medio de una hoja de la misma?  
 Si       No ( )

9. ¿La aplicación móvil permite registrar especies vegetales para usuarios con perfil "Colaborador"?  
 Si       No ( )

10. ¿Considera que la aplicación móvil facilita la identificación de especies vegetales forestales nativas?  
 Si       No ( )

GRACIAS POR SU COLABORACIÓN



UNIVERSIDAD NACIONAL DE LOJA  
 Área de la Energía, las Industrias y los Recursos Naturales No Renovables  
 Carrera de Ingeniería en Sistemas  
 Encuesta para la carrera de Ingeniería Forestal

La presente encuesta tiene como finalidad obtener información relevante sobre el funcionamiento de la aplicación móvil EcuadorNatural, misma que servirá para la sustentación del Trabajo de Titulación denominado " Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el Jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja". Por ello se solicita contestar las siguientes preguntas.

1. ¿En qué sistema operativo instaló la aplicación móvil EcuadorNatural?  
 Android (X)      IOS ( )
2. ¿Qué versión de sistema operativo tiene su teléfono?  
 Respuesta: 2.3.3
3. ¿Qué marca y modelo tiene su teléfono?  
 Respuesta: Samsung Galaxy P10.4
4. ¿El tiempo de respuesta al realizar la identificación de especie vegetales fue?  
 Excelente ( )      Bueno (X)      Regular ( )      Malo ( )
5. La aplicación móvil permite la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el Jardín botánico "Reinaldo Espinosa" de la Universidad Nacional de Loja"  
 Si (X)      No ( )
6. ¿Al realizar la identificación de especies vegetales forestales, obtiene los resultados esperados?  
 Si (X)      No ( )
7. ¿La aplicación móvil permite el registro y la autenticación de usuarios?  
 Si (X)      No ( )
8. ¿La aplicación móvil permite consultar todas las especies vegetales registradas, e identificar especies vegetales, por medio de una hoja de la misma?  
 Si (X)      No ( )

9. ¿La aplicación móvil permite registrar especies vegetales para usuarios con perfil "Colaborador"?  
 Si (X)      No ( )

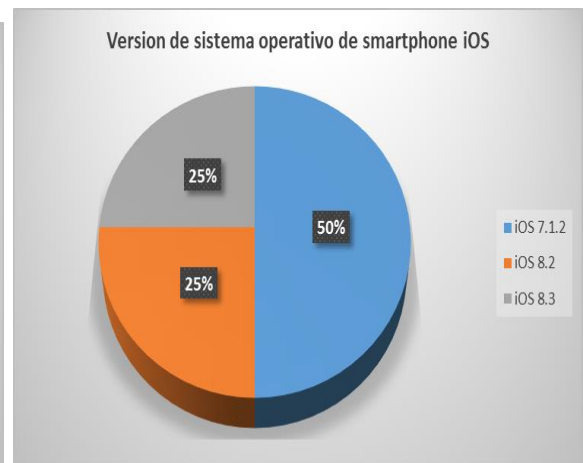
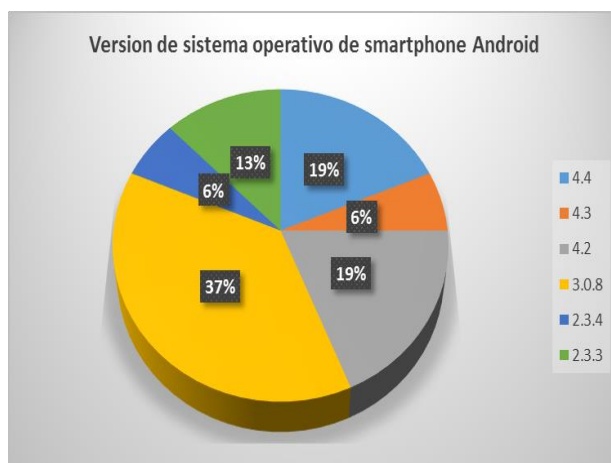
10. ¿Considera que la aplicación móvil facilita la identificación de especies vegetales forestales nativas?  
 Si (X)      No ( )

GRACIAS POR SU COLABORACIÓN

## Pregunta 1



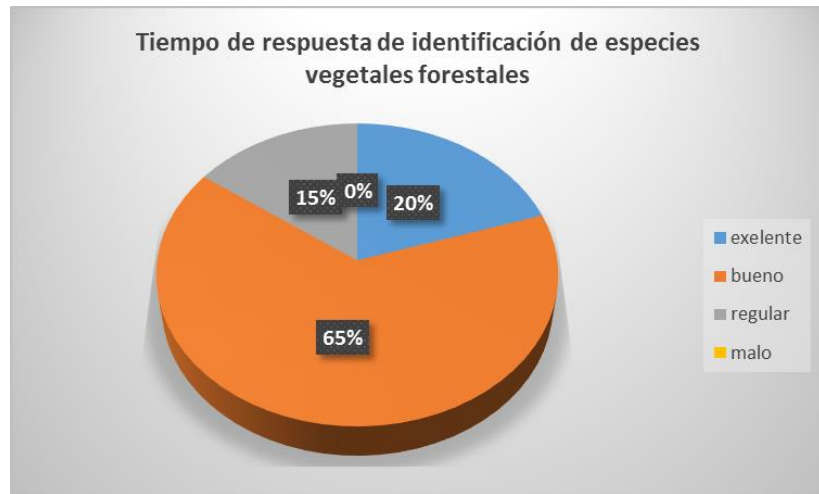
## Pregunta 2



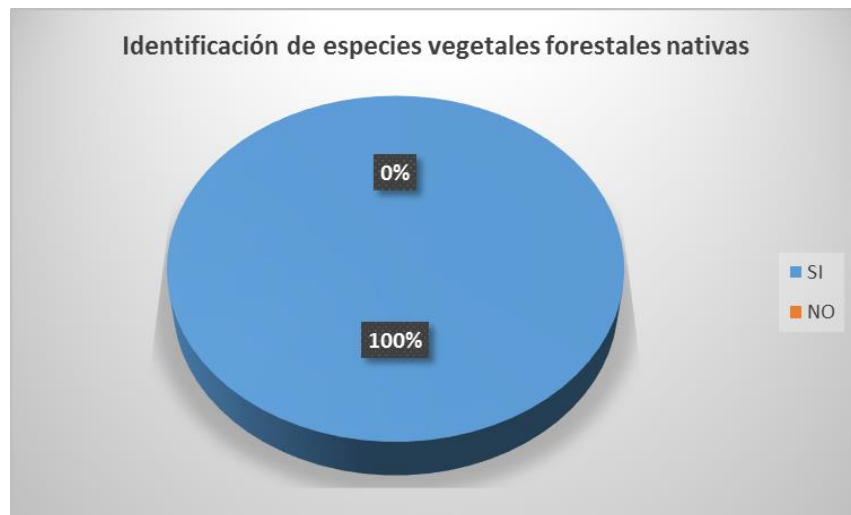
## Pregunta 3



### Pregunta 4



### Pregunta 5



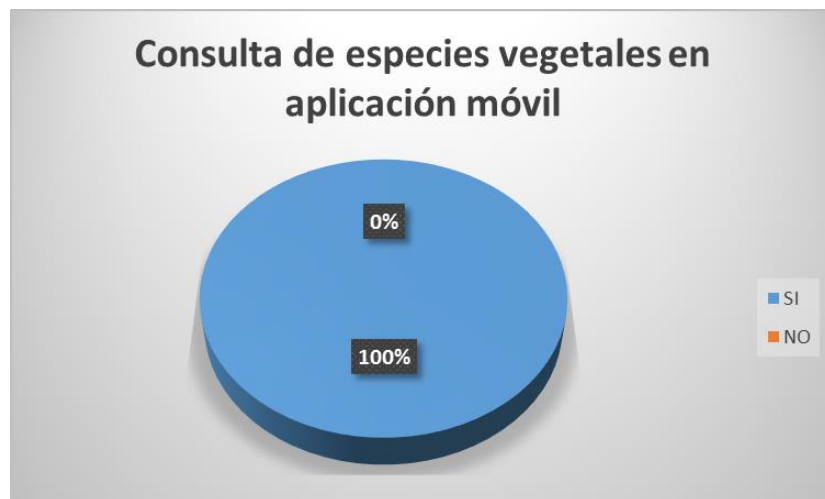
### Pregunta 6



### Pregunta 7



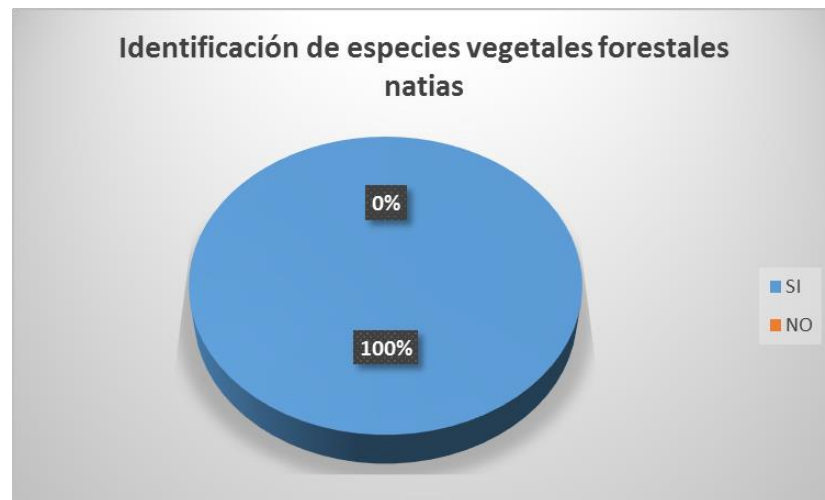
### Pregunta 8



### Pregunta 9



## Pregunta 10



### Discusión de resultados

Luego de haber realizado las pruebas de la aplicación móvil con los estudiantes del AARNR, se pudo determinar que un 80% de estudiantes cuenta con dispositivos móviles smartphone con sistema operativo android y un 20% con sistema operativo iOS para smartphone Apple. Además se pudo determinar que la aplicación móvil funciona bien en el sistema operativo android a partir de la versión 2.3.3 mientras que para iOS se pudo probar que funciona perfectamente a partir de la versión 7.0.

Además se pudo evidenciar que la aplicación móvil tiene buenos tiempos de respuesta en la identificación de especies con un 65%, y 20% de los encuestados piensan que la aplicación tiene excelente tiempo de respuesta y el 15% cree que el tiempo de respuesta es regular, cabe mencionar que esto se debe a la conexión de internet utilizada.

Además se pudo determinar que la aplicación móvil permite en un 100% identificar especies vegetales, y un 95% de los encuestados obtiene los resultados esperados, el 5% que no obtiene los resultados esperados es porque la especie vegetal no se encuentra registrada correctamente. Al mismo tiempo se pudo determinar que la aplicación permite registrar usuarios, asignar un perfil, buscar especies vegetales, identificar y registrar especies vegetales en un 100%. Para finalizar los encuestados consideran que la aplicación móvil es una excelente herramienta que facilita la identificación de especies vegetales obteniendo información oportuna.

## LICENCIA



Guía de campo móvil para la identificación de especies vegetales forestales nativas del Ecuador tomando como piloto el jardín botánico “Reinaldo Espinosa” de la Universidad Nacional de Loja por Freddy Chamorro se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



# NEW YORK ENGLISH LEARNING CENTER

THE NEW YORK ENGLISH LEARNING CENTER  
AV. PIO JARAMILLO ALVARADO  
LOJA  
CIUDAD.



23/06/2015

## CERTIFICADO

Yo certifico que se ha realizado una traducción del resumen de la tesis denominada " GUIA DE CAMPO MÓVIL PARA LA IDENTIFICACIÓN DE ESPECIES VEGETALES FORESTALES NATIVAS DEL ECUADOR TOMANDO COMO PILOTO EL JARDÍN BOTÁNICO "REINALDO ESPINOSA" DE LA UNIVERSIDAD NACIONAL DE LOJA" por Freddy Alexander Chamorro Encalada, portador de cedula número 1105050304.

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado hace uso del presente en lo que el creyere conveniente.

Atentamente

  
  
Gloria Benstead  
Gerente General  
New York English Learning Center

Ref: BK64HC University of Cambridge

Av. Pio Jaramillo Alvarado entre Chile y Cuba, Loja. Tel (07) 2583686. newyorkenglishlearningcenter@gmail.com