



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS**

**Tema:**

---

“APLICACIÓN MÓVIL HÍBRIDA, PARA LA CONSULTA DE PLANILLAS ELÉCTRICAS Y GEOLOCALIZACIÓN DE LAS AGENCIAS DE LA EMPRESA ELÉCTRICA AMBATO”

---

**Trabajo de titulación Modalidad:** Proyecto de investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Brayan Patricio Cujano Soto

**TUTOR:** Ing. Mg. Carlos Israel Núñez Miranda

**Ambato – Ecuador**

**Marzo 2020**

## **APROBACIÓN DEL TUTOR**

En calidad de Tutor del Trabajo de Investigación con el tema: “APLICACIÓN MÓVIL HÍBRIDA, PARA LA CONSULTA DE PLANILLAS ELÉCTRICAS Y GEOLOCALIZACIÓN DE LAS AGENCIAS DE LA EMPRESA ELÉCTRICA AMBATO” desarrollado bajo la modalidad Proyecto de Investigación por el señor Brayan Patricio Cujano Soto, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, agosto 2020



Firmado electrónicamente por:  
**CARLOS ISRAEL  
NUNEZ MIRANDA**

---

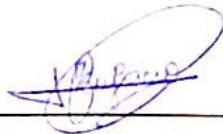
Ing. Mg. Carlos Israel Núñez Miranda

TUTOR

## **AUTORÍA DEL TRABAJO DE TITULACIÓN**

El presente trabajo de investigación titulado: “APLICACIÓN MÓVIL HÍBRIDA, PARA LA CONSULTA DE PLANILLAS ELÉCTRICAS Y GEOLOCALIZACIÓN DE LAS AGENCIAS DE LA EMPRESA ELÉCTRICA AMBATO”. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2020



---

Brayan Patricio Cujano Soto

CC: 1804728903

AUTOR

## APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Brayan Patricio Cujano Soto, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado “APLICACIÓN MÓVIL HÍBRIDA, PARA LA CONSULTA DE PLANILLAS ELÉCTRICAS Y GEOLOCALIZACIÓN DE LAS AGENCIAS DE LA EMPRESA ELÉCTRICA AMBATO”, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, agosto 2020



Firmado electrónicamente por:  
**ELSA PILAR  
URRUTIA**

---

Ing. Pilar Urrutia, Mg.  
PRESIDENTA DEL TRIBUNAL



Firmado electrónicamente por:  
**DAVID OMAR  
GUEVARA  
AULESTIA**

---

Ing. David Guevara  
PROFESOR CALIFICADOR



Firmado electrónicamente por:  
**FELIX OSCAR  
FERNANDEZ  
PENA**

---

Ing. Félix Fernandez  
PROFESOR CALIFICADOR

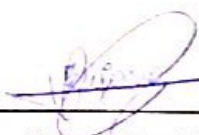


## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, agosto 2020



Brayan Patricio Cujano Soto

CC: 1804728903

AUTOR

## **DEDICATORIA**

Este proyecto va dedicado primordialmente a Dios por permitir continuar con mi vida y mis estudios, al apoyo de mis padres que a pesar de todas las cosas siempre me brindaron su respaldo para poder culminar mis carreras y llegar a ser un profesional, gracias infinitas a ellos, ya que sin su apoyo yo no estuviera culminando esta etapa importante para mí y mi familia, a mis hermanos, cuñados que siempre se preocuparon por mí y me ayudaron de todas las formas posibles.

Va por ustedes porque admiro y valoro todo su esfuerzo y sacrificio que realizaron para poder brindarme todo lo que yo necesitaba y hacer de mí una persona de bien.

Brayan Patricio Cujano Soto

## **AGRADECIMIENTO**

A todo el personal que conforma mi querida FISEI le doy las gracias por formarme como un profesional y darme consejos en ciertas etapas de mi carrera.

A mis compañeros y amigos que compartimos la mayoría de la carrera juntos, gracias por guiarme y ayudarme en mi formación académica, gracias por convertir mi carrera universitaria en los mejores momentos de mi vida.

Agradezco a Diego Cadme por confiar en mí para ser parte de su equipo de desarrollo en la Empresa Eléctrica Ambato, por compartir sus conocimientos y sobre todo por brindarme su amistad incondicional durante todo este proceso.

Mis sinceros agradecimientos al Ingeniero Carlos Nuñez por guiarme en la elaboración de este proyecto y por compartir sus conocimientos.

Brayan Patricio Cujano Soto

## ÍNDICE GENERAL DE CONTENIDOS

CAPITULO I.- MARCO TEÓRICO .....	1
1.1. Antecedentes investigativos .....	1
1.2. Objetivos.....	3
1.2.1 Objetivo general:.....	3
1.2.2 Objetivos específicos: .....	3
CAPÍTULO II.- METODOLOGÍA .....	4
2.1. Materiales .....	4
2.1.1. Humanos .....	4
2.1.2. Institucionales .....	4
2.1.3. Otros.....	4
2.2. Métodos .....	5
2.2.1. Modalidad de la investigación .....	5
2.2.2. Población y muestra.....	5
2.3. Fundamentación Teórica .....	6
2.3.1. Aplicaciones móviles.....	6
2.3.2. Tipos de aplicaciones para dispositivos móviles .....	6
2.3.2.1. Aplicaciones web .....	6
2.3.2.2. Aplicaciones nativas.....	7
2.3.2.3. Aplicaciones híbridas .....	7
2.3.3. Frameworks de desarrollo para aplicaciones móviles híbridas .....	7
2.3.3.1. React Native .....	7
2.3.3.2. Xamarin.....	9
2.3.3.3. Ionic.....	11
2.3.4. Angular .....	14
2.3.5. JavaScript.....	15
2.3.6. TypesCript .....	16
2.3.7. Metodologías Ágiles.....	16
2.3.8. Metodología Scrum.....	17
2.3.9. Servicios web.....	18
2.3.10. Tipo de servicios web .....	18
2.3.11. Servidores de aplicaciones.....	20

2.3.12. Tipos de servidores de aplicaciones.....	20
2.3.13. Sistemas de Gestión de Base de Datos .....	21
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....	24
3.1. Análisis y discusión de los resultados .....	24
3.2. Procesamiento de la información .....	24
3.3. Análisis e interpretación de resultados .....	25
3.4. Desarrollo del Proyecto .....	36
3.4.1. Tecnología a utilizar para el desarrollo de la aplicación web administrativa .....	37
3.4.2. Tecnología a utilizar para el desarrollo y alojamiento de los servicios web .....	37
3.4.3. Determinar el Framework para el desarrollo de la aplicación móvil multiplataforma.....	37
3.4.4. Explicación del esquema de las historias de usuario .....	41
3.4.5. Historias de usuario .....	42
3.4.6. Desarrollo del Product Backlog.....	49
3.4.7. Asignación del Product Backlog a los Sprints.....	50
3.4.8. <i>Estimación de las fechas de entrega para cada Sprint</i> .....	51
3.4.9. Desarrollo del Sprint 1 .....	51
3.4.10. Desarrollo del Sprint 2 .....	110
3.4.11. Desarrollo del Sprint 3 .....	116
3.4.12. Desarrollo del Sprint 4.....	124
3.4.13. Desarrollo del Sprint 5 .....	129
CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES .....	132
Referencias Bibliográficas .....	134
Anexos.....	136

## ÍNDICE DE TABLAS Y FIGURAS

Tabla 1. Materiales Económicos (Presupuesto) .....	4
Tabla 2. Población.....	5
Tabla 3. Plan de recolección de información. ....	24
Tabla 4. Resultados Pregunta 1 .....	25
Tabla 5. Resultados Pregunta 2.....	27
Tabla 6. Resultados Pregunta 3.....	28
Tabla 7. Resultados Pregunta 4.....	29
Tabla 8. Resultados Pregunta 5.....	31
Tabla 9. Resultados Pregunta 6.....	32
Tabla 10. Resultados Pregunta 7.....	33
Tabla 11. Resultados Pregunta 8.....	35
Tabla 12. Tabla comparativa. Frameworks de desarrollo de aplicaciones híbridas..	38
Tabla 13. Definición del equipo Scrum. ....	40
Tabla 14. Esquema historias de usuario.....	41
Tabla 15. Historia de usuario 1 .....	42
Tabla 16. Historia de usuario 2 .....	43
Tabla 17. Historia de usuario 3 .....	43
Tabla 18. Historia de usuario 4 .....	44
Tabla 19. Historia de usuario 5. ....	45
Tabla 20. Historia de usuario 6 .....	45
Tabla 21. Historia de usuario 7 .....	46
Tabla 22. Historia de usuario 8 .....	47
Tabla 23. Historia de usuario 9 .....	47
Tabla 24. Historia de usuario 10 .....	48
Tabla 25. Product Backlog.....	49
Tabla 26. Product Backlog según su Sprint. ....	50
Tabla 27. Estimación de las fechas de entrega para cada Sprint.....	51
Tabla 28. Desarrollo del Sprint 1. ....	51
Tabla 29. Desarrollo del Sprint 2. ....	110
Tabla 30. URIs utilizadas para intercambio de información.....	110
Tabla 31. Desarrollo del Sprint 3. ....	116
Tabla 32. Desarrollo del Sprint 4. ....	124
Tabla 33. Desarrollo del Sprint 5. ....	129
Figura 1. Arquitectura de React Native.....	8
Figura 2. Arquitectura de Xamarin .....	10
Figura 3. Ionic Creator .....	12
Figura 4. Arquitectura de Ionic .....	13
Figura 5. Servidor de aplicaciones WildFly.....	20
Figura 6.- Tabla de precios por el uso de Apis de Google. ....	23
Figura 7. Resultados Pregunta 1.....	26
Figura 8. Resultados Pregunta 2.....	27
Figura 9. Resultados de la Pregunta 3 .....	28
Figura 10. Resultados Pregunta 4.....	30

Figura 11. Resultados Pregunta 5.....	31
Figura 12. Resultados Pregunta 6.....	32
Figura 13.- Resultados Pregunta 6 .....	34
Figura 14. Resultados Pregunta 8.....	35
Figura 15. Frameworks de desarrollo de aplicaciones híbridas .....	38
Figura 16. Modelo de la Base de Datos. ....	52
Figura 17. Instalar nuevo software.....	53
Figura 18. Añadir Repositorio.....	53
Figura 19. Iconos de GWT.....	54
Figura 20. Creación de un nuevo proyecto. ....	54
Figura 21. Datos para crear proyecto nuevo. ....	55
Figura 22. Configuración Project Facets.....	56
Figura 23. Configuración Java Build Path. ....	56
Figura 24. Estructura del proyecto. ....	57
Figura 25. Ejecutar un proyecto. ....	58
Figura 26. Instalar plugin de GWT para Firefox.....	58
Figura 27. Creación de un servicio remoto. ....	83
Figura 28. Solución error al crear un servicio remoto. ....	84
Figura 29. Diseño de la ventana WinEeasa_ubi_geo_ageList.....	91
Figura 30. Diseño de la ventana WinEeasa_ubi_geo_age. ....	99
Figura 31. Crear un proyecto web dinámico.....	100
Figura 32. Información para crear un proyecto web dinámico. ....	101
Figura 33. Convertir en proyecto Maven. ....	102
Figura 34. Código del archivo module.xml .....	106
Figura 35. Código del archivo standalone.xml .....	106
Figura 36. Escoger el driver configurado.....	107
Figura 37. Configurar parámetros de conexión.....	108
Figura 38. Correr servicio web en WildFly. ....	109
Figura 39. Consumir servicio web en SoapUI. ....	109
Figura 40. Creación de las variables del sistema. ....	113
Figura 41. Edición variable del sistema Path. ....	114
Figura 42. Edición variable del sistema Path. ....	114
Figura 43. Creación de una aplicación en Ionic. ....	115
Figura 44. Creación de un proyecto en Google Cloud.....	117
Figura 45. Habilitar API de GoogleMaps. ....	118
Figura 46. Agregar plugin de GoogleMaps.....	118
Figura 47. Vistas de la pantalla de geolocalización en iOS. ....	121
Figura 48. Vistas de la pantalla de geolocalización en Android .....	122
Figura 49. Vista de la pantalla principal en iOS y Android. ....	124
Figura 50. Vista del Segmento uno .....	126
Figura 51. Vista del Segmento dos. ....	127
Figura 52. Vista del segmento tres.....	128
Figura 53. Versiones de la aplicación .....	129
Figura 54. Encuesta clasificación de contenido .....	130
Figura 55.- Aplicación publicada en la Play Store.....	131

## RESUMEN EJECUTIVO

El desarrollo de aplicaciones híbridas ha sido un tema que está tomando mucho interés en los desarrolladores que buscan flexibilidad y reutilizar el código, mismo que será 99 por ciento compartido con los diferentes sistemas operativos móviles.

Los smartphones se han convertido en una herramienta indispensable para las empresas y para personas en general, en los países avanzados la tecnología móvil ha generado más ingresos económicos que cualquier otra tecnología, debido a la creciente demanda los desarrolladores se han visto en la necesidad de crear aplicaciones que mejoren la experiencia de los usuarios a la hora de usar los smartphones.

Las aplicaciones móviles son necesarias para los smartphones, pero crear aplicaciones para la variedad de sistemas operativos móviles que existen conlleva a incrementar costos de desarrollo al tener que reescribir el código para cada plataforma, por tal motivo las aplicaciones híbridas se están convirtiendo en una tecnología muy explorada ya que cumple con todos los requisitos buscados como optimización de tiempo, dinero y aprendizaje de nuevos lenguajes de programación, cosas que hoy son muy importantes para desarrollar aplicaciones. A lo largo de la investigación se indagará sobre los frameworks más comunes para desarrollar este tipo de aplicaciones y finalmente decidirse por el mejor para llevar a cabo el desarrollo de la aplicación.

Por lo necesario que es para una empresa tener una aplicación móvil, en la Empresa Eléctrica Ambato se ha desarrollado una, bajo el nombre de EEASA móvil. Dicha aplicación facilitará a los usuarios consultar el valor de la planilla eléctrica desde cualquier lugar, además de conocer los puntos de pago más cercanos a la ubicación actual.

**Palabras Clave:** Celulares, aplicaciones híbridas, framework, servidor de aplicaciones.



## ABSTRACT

The development of hybrid applications has been an issue that is taking a lot of interest in developers that need flexibility and reusing code in your applications, this will be 99 percent shared with different mobile operating systems.

The smartphones have been converted in one necessary tools for businesses and people, in improve countries the mobile technology has generated more economic income that any other technology, due to the growing demand, the developers has been in need to create application for improve the experiences of user in the moment of using smartphones. Throughout the research, the most common frameworks for developing this type of applications will be investigated and finally deciding on the best one to carry out the development of the application.

The mobile applications are necessary for smartphones, but develop mobile applications for variety of operating systems leads to increase development cost because developers have to rewrite the source code for each platform, for this reason the hybrid applications are becoming in a highly explored technology, because meet all the finding requirements as optimization time, money and learning new programming languages, things that are very importants to develop applications.

Because it is necessary for a company to have a mobile application, one has been developed at Empresa Eléctrica Ambato, under the name of EEASA móvil. This application will facilitate users to consult the value of the electrical bill from anywhere, in addition to knowing the payment points closest to the current location.

**Key words:** Smartphones, hybrid applications, frameworks, application servers

## CAPITULO I.- MARCO TEÓRICO

### 1.1. Antecedentes investigativos

De todos los trabajos que tuvieron cierta similitud con la investigación, se puede mencionar los siguientes:

Luis Gonzalo Yaguapaz Madera en el año 2018, en su trabajo de grado “Estudio del framework Ionic para el desarrollo de aplicaciones móviles híbridas” concluyó que las nuevas tecnologías, nuevos lenguajes de programación y las tendencias de desarrollo como lo es Ionic ayudan a los programadores y dueños del producto a ahorrar costos y tiempo significativamente en este caso generar aplicaciones para diferentes plataformas partiendo de un solo proyecto de desarrollo que se basa en un único lenguaje de programación[1]. Además, que el estudio de nuevas herramientas se la debe hacer de manera minuciosa, para poder aprovechar al máximo las bondades de las herramientas investigadas.

Esteban Cano Insa en el año 2015, en el informe final de su tesis de investigación “BiciPark: Web App con Ionic y acceso a servicios Rest” recomienda usar Ionic porque aporta un ecosistema de herramientas que en la actualidad son completamente gratuitas y que van a facilitar en gran medida las labores de desarrollo de aplicaciones híbridas[2].

Ms. Johnny Alexander Salazar Cardona, Ing. David Alberto Angarita, Ing. Juan David Montoya en el año 2016, en su trabajo de investigación “Tendencias en desarrollo móvil bajo las tecnologías Android e IOS” concluyen que a pesar de que existen diferentes enfoques de desarrollo móvil, para seleccionar cual es la mejor opción de trabajo se debe evaluar diferentes factores según cada desarrollo, con el fin de determinar cuál es la mejor opción[3].

En la Universidad Shanghai Lixin University of Commerce de China, Hongjun Li investigó sobre “RESTful Web Service Frameworks in Java”, el cual habla del uso de la arquitectura REST para crear servicios web, porque es una manera más flexible de transportar datos por el protocolo HTTP y permite usar los métodos propios de dicho protocolo[4].

Roxana Madalina Gogonea en el año 2016, en su trabajo final de grado “Cordova Generator: Generación automática de plugins Apache Cordova para Aplicaciones Híbridas” dice que, para el desarrollo de este software, ha sido necesario definir una serie de objetivos que marcasen el avance del proyecto. De esta manera, el primer paso ha consistido en analizar las diferentes tecnologías presentes en el mercado para el desarrollo de aplicaciones híbridas, en el que se ha determinado, que la mejor tecnología para el desarrollo de las aplicaciones híbridas es Ionic. La razón por la que se ha escogido esta tecnología es porque hace uso de AngularJS, que permite crear aplicaciones rápidas y robustas. Y que, además, cuenta con elementos visuales inspiradas en las diferentes plataformas móviles, permitiendo de esta manera, crear aplicaciones que se asemejan visualmente a una aplicación nativa[5].

## **1.2. Objetivos**

### **1.2.1 Objetivo general:**

Desarrollar una aplicación móvil híbrida para la consulta de planillas eléctricas y localización geográfica de las agencias de la Empresa Eléctrica Ambato.

### **1.2.2 Objetivos específicos:**

- 1) Identificar los procesos y requerimientos funcionales para el desarrollo de la aplicación.
- 2) Determinar el framework de trabajo para realizar aplicaciones móviles híbridas.
- 3) Implementar la aplicación móvil para la consulta de planillas eléctricas y localización geográfica de las agencias de la Empresa Eléctrica Ambato.

## CAPÍTULO II.- METODOLOGÍA

### 2.1. Materiales

#### 2.1.1. Humanos

- Investigador.
- Docente tutor de la Universidad Técnica de Ambato.
- Tutor Empresarial de la EEASA.

#### 2.1.2. Institucionales

- Bibliotecas y repositorios virtuales de la Universidad Técnica de Ambato.

#### 2.1.3. Otros

- Computadora.
- Escritorio de trabajo.
- Internet.
- Dispositivo móvil.
- Framework de desarrollo.

#### 2.1.4. Económicos (Presupuesto)

*Tabla 1. Materiales Económicos (Presupuesto)*

No	Detalle	Unidad	Cantidad	Valor unitario	Valor total
1	Copias/impresiones	c/u	500	\$0.05	\$25
2	Computadora	c/u	1	\$1150	\$1150
3	Dispositivo móvil	c/u	1	\$300	\$300
4	Servicio de Internet	c/mes	5	\$18	\$90
6	Licencia de desarrollador de Google	c/u	1	\$25	\$25

7	Licencia de desarrollador de Apple	c/año	1	\$99	\$99
				Sub Total	\$1689
				Imprevistos (10%)	\$168
				Total	\$1857

*Elaborado por: El investigador*

La Tabla 1 hace referencia a los gastos realizados durante el desarrollo de la investigación propuesta.

## **2.2. Métodos**

### **2.2.1. Modalidad de la investigación**

La modalidad de investigación del proyecto es bibliográfica, aplicada y de campo.

Bibliográfica: porque se recolecta información de fuentes como libros, revistas, proyectos de investigación, etc. Para el desarrollo de la investigación.

Aplicada: Por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

De campo: porque el proyecto se lo realiza a nivel empresarial y la realización de la investigación es de gran aporte para los usuarios de la EEASA.

### **2.2.2. Población y muestra**

*Tabla 2. Población*

<b>No</b>	<b>Tipo de población</b>	<b>Cantidad</b>
1	Presidencia Ejecutiva	1
2	Departamento Comercial	2
3	Dirección de Recursos Humanos	1

4	Cientes de la EEASA	46
Total		50

*Elaborado por: El investigador*

Como se muestra en la Tabla 2. Para la población se establece un total de 50 personas, las cuales son administrativos interesados en el desarrollo de este proyecto y clientes que antes de realizar su pago de la planilla, usan las pantallas informativas de la de la EEASA para consultar el valor a pagar.

Debido al tamaño de la población que es menor o igual a 100, no es necesario realizar una muestra y se trabaja con toda la población.

## **2.3. Fundamentación Teórica**

### **2.3.1. Aplicaciones móviles**

Son programas diseñados o pensados para ejecutarse en dispositivos inteligentes como: tablets, teléfonos, reproductores mp3, etc. Tienen como finalidad realizar actividades, ya sea de ocio, empresariales, educativas o informativas[6].

### **2.3.2. Tipos de aplicaciones para dispositivos móviles**

En Ecuador desde la llegada de tecnologías como el internet, Wifi, etc. La tecnología móvil, de manera especial los teléfonos inteligentes han avanzado notablemente, además de las tecnologías mencionadas, las aplicaciones móviles son aquellas que dan el 100% de funcionalidad e interés al teléfono. A continuación, se muestra los tipos de aplicaciones móviles.

#### **2.3.2.1. Aplicaciones web**

Las aplicaciones web son aquellas que se ejecutan desde un navegador sin la necesidad de instalar en el dispositivo, la principal ventaja de este tipo de aplicaciones, es que no depende de ningún sistema operativo para su ejecución y su desventaja es, que no se puede utilizar los elementos de hardware como: bluetooth, cámara, GPS, etc.

Para este tipo de aplicaciones es necesario que el dispositivo móvil posea de una conexión a internet[7].

#### **2.3.2.2. Aplicaciones nativas**

Las aplicaciones nativas son aquellas que se desarrollan pensando en el sistema operativo móvil en el que se va a utilizar, este tipo de aplicaciones a diferencia de las aplicaciones web se necesita instalar la aplicación además de otorgar permisos para su ejecución, una de sus ventajas es que, se puede sacar el máximo provecho a su hardware, es decir se puede realizar aplicaciones que requieran de los servicios de ubicación, bluetooth, cámara, acelerómetro, etc[7]. Una desventaja de este tipo de aplicación es que, para tener funcionando en las diferentes plataformas móviles se debe desarrollar una aplicación por plataforma, esto conlleva a mayores gastos.

#### **2.3.2.3. Aplicaciones híbridas**

Las aplicaciones móviles híbridas son aplicaciones que combinan lo mejor de los tipos de aplicaciones anteriores, ya que se utiliza tecnologías como HTML, CSS, JavaScript en el caso de aplicaciones web y se puede utilizar el hardware de los dispositivos en el caso de aplicaciones nativas[7]. Este tipo de aplicaciones se disfraza de aplicaciones nativas, Porque se instala como una aplicación nativa, pero se ejecuta dentro de un contenedor web. Una ventaja, es que se puede distribuir a las diferentes plataformas con el mismo código, reduciendo los gastos de su desarrollo, la desventaja de este tipo de aplicaciones es que, la aplicación será más lenta al momento de ejecutar a diferencia de las aplicaciones nativas.

### **2.3.3. Frameworks de desarrollo para aplicaciones móviles híbridas**

#### **2.3.3.1. React Native**

React Native es un framework de desarrollo de aplicaciones híbridas desarrollada por Facebook, basada en JavaScript. Cuando se lanzó por primera vez este framework, los desarrolladores solo daban soporte para iOS, pero a medida de la aceptación que tuvo el soporte para Android se ha agregado y aún se está expandiendo[8].



El principal propósito de React Native es que el desarrollador no debe requerir de mucho tiempo para desarrollar una aplicación móvil, Debido que cada sistema operativo tiene diferentes aspectos, capacidades y especificaciones, React Native permite crear un solo código fuente, pero hace que los gráficos se procesen de diferente manera como si se tratara de una aplicación nativa dependiendo del sistema operativo objetivo.

### Arquitectura de React Native



Figura 1. Arquitectura de React Native

Elaborado por: El investigador

A continuacion, se explicará la Aquitectura de React Native mostrada en la Figura 1.

### Front End

Debido a que React Native fue creado en base a React, el front end lo podemos diseñar en código JavaScript, utilizando los motores de JavaScript V8 para el caso de Android y JavaScriptCore para iOS, estos motores se encargan de interpretar el código en JavaScript, dando como resultado la funcionalidad programada en el script.

### **Puente React Native**

El puente nativo de React Native actúa como un traductor, comunicándose con el motor de JavaScript y ejecutando código nativo, Java para Android y C++ para iOS, es por eso que React Native no se ejecuta sobre un WebView, y da al usuario una apariencia idéntica a la de una aplicación nativa.

### **Sistema Operativo Móvil**

El sistema operativo móvil se encarga de mostrar al usuario la aplicación nativa construida en React Native.

### **Ventajas de React Native**

A continuación se presenta las ventajas tomadas de la referencia [8].

- React Native no es un Web View, es decir todo lo que se ejecuta es nativo.
- React Native permite tener acceso a las componentes propias de los dispositivos móviles, siempre y cuando se tengas conocimiento para la implementación de los mismo.

### **Desventajas de React Native**

A continuación se presenta las desventajas tomadas de la referencia [8].

- Usa un peso adicional a la que pesaría cualquier otra aplicación.
- Es una framework que aún no se encuentra maduro para grandes proyectos.

#### **2.3.3.2. Xamarin**

Es un framework de desarrollo de aplicaciones móviles para generar aplicaciones nativas para Android, iOS y Windows Phone, a partir de una base de código C#.NET[9].

Xamarin fue lanzado en el 2011, por los mismos ingenieros que crearon el proyecto Mono. Pero en el 2016 fue comprado por la empresa Microsoft y paso a ser una plataforma gratuita, además que se puede integrar con Microsoft Visual Studio.

Xamarin propone un enfoque de desarrollo multiplataforma, el cual permite compartir 100% el código de la lógica de negocio, pero las interfaces deben ser programadas de manera individual para cada sistema operativo móvil.

### Arquitectura de Xamarin



*Figura 2. Arquitectura de Xamarin*

*Elaborado por: El investigador*

A continuacion, se explicará la Aquitectura de Xamarin mostrada en la Figura 2.

### Interfaz Gráfica

La interfaz gráfica para la aplicación se tiene que diseñar una para cada sistema operativo móvil, es por eso al crear un proyecto en Xamarin me genera tres clases que tiene la siguiente nomenclatura nombreApp.iOS, nombreApp.Android, nombreApp.Uwp (para iOS, Android y Windows Phone respectivamente)[9].

## **Mono**

Mono es un puente que utiliza Xamarin para comunicarse con el sistema operativo objetivo, se encarga de traducir el código C# a código nativo, es por eso que Xamarin no funciona sobre un Web View, más bien es una aplicación nativa[9].

## **Sistema Operativo Móvil**

El sistema operativo móvil se encarga de mostrar al usuario la aplicación nativa construida en Xamarin.

## **Ventajas de Xamarin**

A continuación se presenta las ventajas tomadas de la referencia [10].

- Xamarin ejecuta las aplicaciones de manera nativa, sin importar del sistema operativo móvil en el cual se está ejecutando.
- Permite llamar código existente, es decir si tienes una porción de código en Java, se puede utilizar en Xamarin.

## **Desventajas de Xamarin**

A continuación se presenta las desventajas tomadas de la referencia [11].

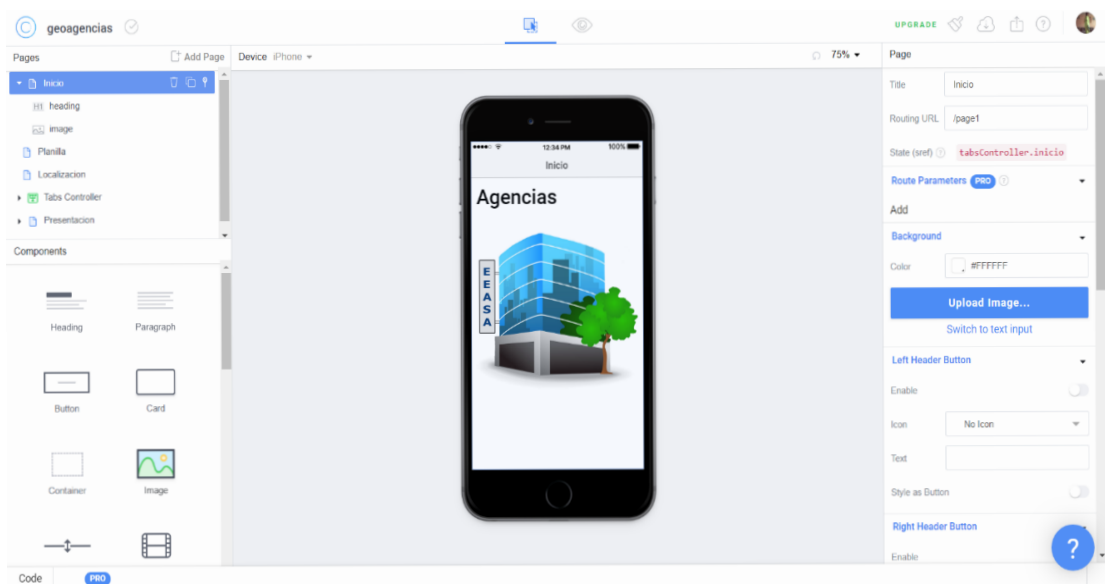
- Se debe diseñar la interfaz gráfica por separado para cada sistema operativo móvil.
- No permite actualización instantánea, es decir cada vez que prueben la aplicación, hay que ejecutar y compilar la misma.

### **2.3.3.3. Ionic**

Es un framework que me permite crear aplicaciones híbridas con tecnologías web como HTML, CSS y JavaScript. Utiliza Apache Cordova para empaquetar las aplicaciones para plataformas como Android e IOS[12].

La principal ventaja es que Ionic es un framework 100% libre y de código abierto, bajo la licencia del Instituto Tecnológico de Massachusets (MIT). Ionic combina varias tecnologías que trabajan en conjunto para crear aplicaciones móviles híbridas de una forma rápida y sencilla.

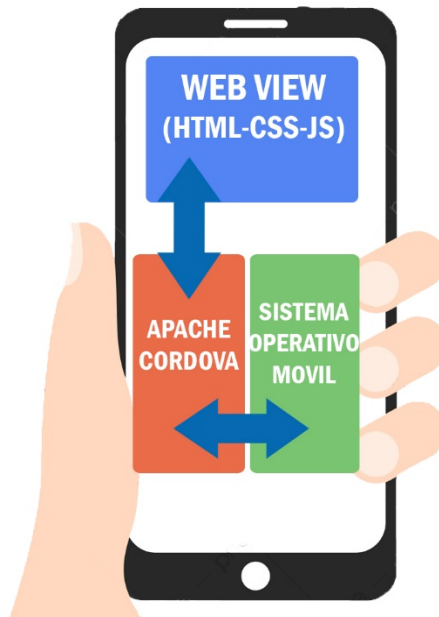
Ionic permite a los desarrolladores de aplicaciones móviles construir una base de código única y usar en todas las plataformas principales como Android, iOS, Windows Phone. Ionic es bien conocido para el desarrollo de aplicaciones móviles de alto rendimiento, porque se puede utilizar **Ionic Creator**, el cual sirve para desarrollar la interfaz gráfica de la aplicación sin que se escriba una sola línea de código. Ionic Creator utiliza el concepto de arrastrar y soltar, es decir, seleccionar los componentes que se utilizarán en la pantalla, arrastrar hacia el lugar preferido donde se situará el componente y soltar, La Figura 3 muestra la interfaz de Ionic Creator.



*Figura 3. Ionic Creator*

*Elaborado por: El investigador*

## Arquitectura de Ionic



*Figura 4. Arquitectura de Ionic*

*Elaborado por: El investigador*

A continuación, se explicará la Aquitectura de Ionic mostrada en la Figura 4.

### **Web View**

Es un navegador web que ejecuta código HTML, CSS y JavaScript, motivo por el cual podemos realizar aplicaciones intuitivas y dinámicas. El Web View puede tener desde una única interfaz de usuario ya sea para consulta de datos mediante un servidor remoto, hasta una interfaz que utilice funcionalidades propias del celular como el GPS para localización geográfica, esto es posible gracias a Apache Cordova[12].

### **Apache Cordova**

Es el software que se encarga de convertir la aplicación web en aplicación móvil nativa para diferentes sistemas operativos, además provee de APIs que interactúan con las características nativas del dispositivo, como acceder a la cámara, emails, agenda, GPS, etc... mediante JavaScript[13].

Una vez compilada la aplicación web podemos instalar en el sistema operativo para el cual fue compilado.

### **Sistema operativo móvil**

Apache Cordova funciona como un puente entre el sistema operativo y el Web View, ya que se encarga de comunicar con la Web View en lenguaje JavaScript y con el sistema operativo en lenguaje nativo[13].

### **Ventajas del framework Ionic**

A continuación se presenta las ventajas tomadas de la referencia [12].

- Con un solo código fuente podemos compilar aplicaciones para Android, IOS, Windows Phone.
- Se basa en la utilización de HTML junto con CSS y JavaScript.

### **Desventajas del framework Ionic**

A continuación se presenta las desventajas tomadas de la referencia [12].

- El rendimiento de una aplicación en Ionic puede notarse menor al rendimiento que tiene una aplicación desarrollada de forma nativa.
- Ionic es un framework muy joven por el cual esta propenso a muchas actualizaciones constantemente.

Ionic es un framework muy eficaz, gracias a su integración con Angular, la programación Angular en se hace usando TypeScript.

### **2.3.4. Angular**

Angular es uno de los frameworks más populares y preferidos por los desarrolladores JavaScript, Angular es bastante amigable con otros frameworks, por eso

se lo utiliza en frameworks que permiten el desarrollo de aplicaciones móviles híbridas como: Ionic, NativeScript[14].

Este framework se convirtió en una plataforma de desarrollo de aplicaciones móviles híbridas, porque con Angular las aplicaciones son reactivas y no recargan el navegador, todo es muy dinámico y asíncrono.

### **Principales características de Angular**

- **Velocidad y rendimiento**
  - **Generación de código:** Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript[12].
  - **División del código:** Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes[12].
  
- **Productividad tomadas de [12]**
  - **Plantillas:** Permite crear rápidamente vistas de interfaz de usuario con una sintaxis de plantilla simple y potente.
  - **Angular CLI:** Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como previsualizar de forma instantánea la aplicación.
  - **IDEs:** Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs.

### **2.3.5. JavaScript**

JavaScript es un lenguaje interpretado usado para múltiples propósitos, una de las innovaciones que ayudo a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código[15].

Para aprovechar esta plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en relación con portabilidad e integración. A la vez,



interfaces de programación de aplicaciones(APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales[15].

### **2.3.6. TypesCript**

Es un lenguaje de programación moderno que permite crear aplicaciones web robustas en JavaScript. TypeScript no requiere de ningún tipo de plugin, puesto que lo que hace es generar código JavaScript que se ejecuta en cualquier navegador, plataforma o sistema operativo[15].

TypeScript convierte su código en Javascript común. Es llamado también Superset de Javascript, lo que significa que, si el navegador está basado en Javascript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el Javascript como lenguaje original.

¿Qué es un superset?

Se trata de un lenguaje escrito sobre otro lenguaje. En este caso Typescript es eso, un lenguaje basado en el original, ofreciéndonos grandes beneficios. Existen otros superset de JavaScript pero estos se alejan del código original, Typescript, por el contrario, es muy similar a Javascript [15].

### **2.3.7. Metodologías Ágiles**

Las metodologías ágiles buscan desarrollar software funcional en los tiempos establecidos por el grupo de trabajo. Estas metodologías permiten adaptar la forma de trabajo a las condiciones del proyecto, es decir si los requerimientos del software cambia el equipo debe adaptar el producto a estos cambios.

#### **Ventajas de las Metodologías ágiles**

A continuación se detalla las ventajas de las metodologías ágiles tomadas de la referencia [16].

- Trabajar con mayor velocidad y eficiencia.
- Mejora la Calidad del Producto.
- Ahorra tiempo y costos.
- El equipo de desarrollo interactúa más entre sí.

### **2.3.8. Metodología Scrum**

Es un marco de trabajo que se utiliza para desarrollo de productos y software. Se utiliza en proyectos de complejidad media ya que está orientada a normalizar los errores que se pueden producir en desarrollos largos, debido a sus reuniones diarias los integrantes del equipo aseguran el cumplimiento de los objetivos establecidos para el proyecto[17].

Las reuniones son el pilar fundamental de la metodología en especial la que se hace después de entregar un sprint para analizar mejoras en el proyecto. Scrum está compuesto de los siguientes roles, mismos que fueron tomados de la referencia [18]:

- Dueño del producto (Product Owner). - es aquel que tiene una visión de lo que desea construir, puede ser una persona interna o externa a la empresa que desarrolla el producto. Este rol asume una persona que tenga disponibilidad, entendida en el tema de lo que se desea construir y que tenga facilidad de comunicación con los integrantes del equipo.
- Facilitador (Scrum Master). – como principal función está la de conseguir un equipo de alto rendimiento, tanto como equipo de desarrollo y Dueño del producto, además se encarga de eliminar los obstáculos que llegasen a ser un impedimento para que el equipo Scrum alcance el objetivo del Sprint.
- Equipo de Desarrollo (Development Team). – el equipo de desarrollo integra los profesionales que de manera conjunta desarrollan el producto.

### **Eventos Scrum**

- **Planificación de iteración (Sprint planning)**

Sprint Planning reúne a todo el equipo Scrum. El Equipo de Desarrollo pronostica lo que puede desarrollarse de manera realista durante el Sprint. El propietario del producto analiza el objetivo que el equipo Scrum espera lograr al final del Sprint[16].

- **Reunión diaria de sincronización (Daily Scrum)**

Cada día, el Equipo de Desarrollo dedica 15 minutos para sincronizar las actividades y desarrollar un plan para las próximas 24 horas. El objetivo de esta reunión es aumentar la comunicación con los miembros del equipo, cada uno de los miembros del equipo revisan el trabajo del resto y verifican si los objetivos a entregar al final de la iteración se están cumpliendo[18].

- **Demostración de requisitos completados (Sprint review)**

Una vez que se ha completado el Sprint, se lleva a cabo una revisión del Sprint, en la revisión del Sprint se verifica los logros obtenidos durante la iteración, problemas que acontecieron y como dar solución a dichos problemas[16].

- **Retropectiva (Sprint retrospective)**

Se lleva a cabo una vez que se completa la Revisión de Sprint y antes de que comience la nueva Planificación de Sprint. El equipo analiza cómo ha sido la forma de trabajar durante la iteración, además analiza si se está o no consiguiendo los objetivos planteados al inicio de la iteración [16].

### **2.3.9. Servicios web**

Es un conjunto de métodos, los cuales se puede acceder desde internet o intranet, las invocaciones de estos métodos son independientes del sistema operativo, lenguaje de programación en el cual fueron desarrollados los servicios.

### **2.3.10. Tipo de servicios web**

Los servicios web mas utilizados en la actualidad son SOAP Y REST.

## **SOAP**

SOAP (Simple Object Access Protocol) es un protocolo utilizado para el intercambio de información basados en XML sobre la red, esta información por lo general viaja sobre HTTP/HTTPS[19].

SOAP envía un mensaje desde la aplicación cliente hasta la aplicación servidor, solicitando que se ejecute el conjunto de métodos que el servicio web posee, el servidor responde los resultados del método realizado con un mensaje en formato XML[20].

## **REST**

REST (Representational State Transfer) es una tecnología más blanda que transporta información por medio del protocolo HTTP/HTTPS, a diferencia de SOAP, REST puede transmitir cualquier tipo de datos como XML, JSON, Binarios, etc. El formato más utilizado para transmitir información en servicios web tipo REST es Json, porque es interpretado de manera natural por JavaScript [21].

Una ventaja muy significativa de REST es que la información en formato Json son mucho más livianos y más rápidos para procesar la información, esto ayuda a la hora de utilizar en aplicaciones móviles puesto que, al ser mas livianos, necesitara menos recursos y menos ancho de banda para que la información del servicio web sea procesada.

REST realiza las operaciones más utilizadas para el manejo de datos POST(crear), GET(consultar), PUT(editar), DELETE(eliminar). La aplicación desarrollada envía un mensaje con la URI del recurso que necesita ser consumido, estas peticiones se realizan mediante la operación GET, y la información que el servidor devuelve es en formato Json[22].

Las URIs son direcciones que usará los servicios web para comunicarse con el cliente y devolver el método requerido. Si el método requiere de parámetros, este se lo envía en la misma URI.

### 2.3.11. Servidores de aplicaciones

Es un servidor en la red que se encarga de gestionar la lógica de negocio y acceso a datos de la aplicación. Los principales beneficios de usar un servidor de aplicaciones es la alta disponibilidad y escalabilidad[23]. La disponibilidad se refiere que la aplicación debe estar funcionando 24/7 y la escalabilidad se refiere a la capacidad que tiene el servidor de aplicaciones para hacer frente al aumento de peticiones de los usuarios.

### 2.3.12. Tipos de servidores de aplicaciones

#### Glassfish

Es un servidor de aplicaciones desarrollada por Sun MicroSystem, implementa tecnologías de java, por lo cual es un servidor que ejecuta aplicaciones construidas con estas especificaciones[23].

#### WildFly

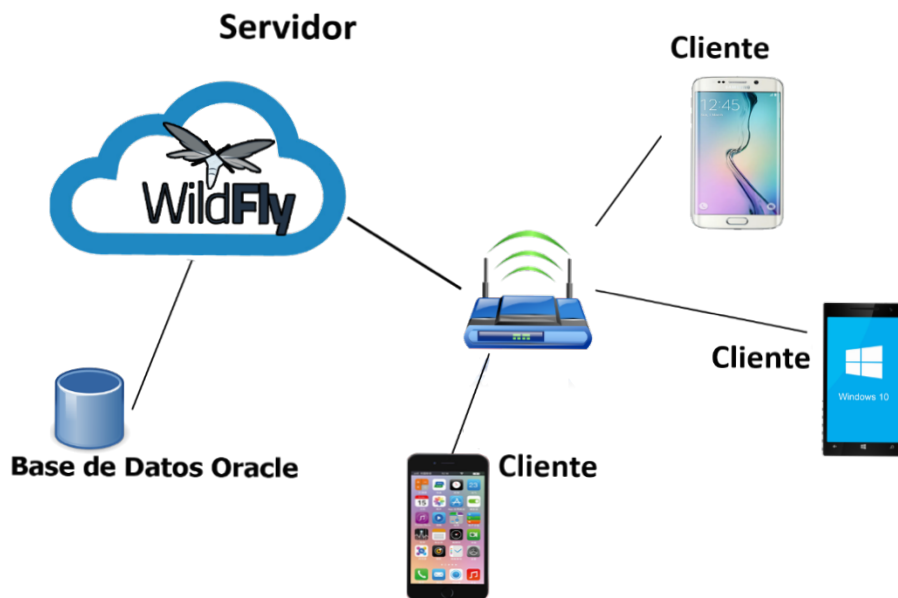


Figura 5. Servidor de aplicaciones WildFly.  
Elaborado por: El investigador

Es un servidor de aplicaciones liviano y flexible, está desarrollada en Java, es de código libre y se puede utilizar en cualquier plataforma que soporte la maquina virtual de java[24].

La Figura 5 muestra la trazabilidad que WildFly realiza para dar respuesta a las peticiones de los clientes.

Wildfly es un servidor de aplicaciones libre y totalmente compatible con Eclipse, sus principales características son:

- Despliegue rápido.
- Peso ligero.
- Pool de conexiones con balanceo de carga.

### **2.3.13. Sistemas de Gestión de Base de Datos**

Un Sistema de Gestión de Base de Datos (SGBD) permite la creación, gestión y administración de bases de datos. Es conocido también como un gestor de datos, y a través de él, se maneja todo acceso a la información en una base de datos. Un sistema de gestión de base de datos puede ser utilizado por cualquier persona con o sin conocimientos amplios de informática dentro de una empresa[18].

Los Sistemas de Gestión de Base de Datos se pueden clasificar por la forma en que administran los datos: Relacionales (SQL) y No Relacionales (NoSQL).

#### **Sistemas Gestores de Bases de Datos Relaciones**

Existen muchos SGBD actualmente, pero entro los más populares se encuentran:

- MySQL
- MariaDB
- Oracle DataBase

- PostgreSQL
- Microsoft SQL Server

### **Oracle DataBase**

Es un sistema de gestión de base de datos, producido y comercializado por Oracle Corporation. Su primera versión fue lanzada en el año de 1979, ahora con 41 años de experiencia se ha convertido en un lenguaje muy utilizado por los administradores de base de datos de las empresas.

Sus principales características son:

- Herramienta de administración gráfica intuitiva y cómoda de utilizar.
- Lenguaje de diseño de bases de datos muy completo (PL/SQL).
- Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.

### **API de Google Maps**

Con el SDK de Maps para Android e iOS, se puede mostrar los mapas de Google en aplicaciones móviles, páginas web, etc. La API gestiona automáticamente el acceso a los servidores de Google Maps, la descarga de datos, la visualización del mapa y la respuesta a los gestos del mapa. También puede usar para agregar marcadores, polígonos y superposiciones a un mapa básico. La API le permite agregar los siguientes iconos a un mapa:

- Iconos anclados a posiciones específicas en el mapa (Marcadores).
- Conjuntos de segmentos de línea (Polilíneas).
- Segmentos cerrados (polígonos).
- Gráficos de mapa de bits anclados a posiciones específicas en el mapa (superposiciones de suelo).
- Conjuntos de imágenes que se muestran en la parte superior de los mosaicos del mapa base (superposiciones de mosaicos).

Si bien es cierto a partir del 16 de julio del 2018 Google Maps pasó a ser de pago, sin embargo, Google deja libre el uso de Mapas Nativos para dispositivos móviles, si se desea utilizar una API diferente a los que están disponible gratuitamente, Google indica que se debe introducir los datos de facturación, con el fin de cobrar las tarifas en caso de que sea necesario, es decir no obligatoriamente se deberá pagar, solo se debe pagar si se superan un número de visitas al mes. En la Figura 6 indica los precios que a pagar y el número de visitas que se puede usar gratis por cada API que Google ofrece.

SKU	\$200 de crédito mensual Uso gratuito equivalente	Rango de volumen mensual (Precio por miles)	
		De 0 a 100,000	De 100,001 a 500,000
Mobile Native Static Maps	Cargas ilimitadas	\$0.00	\$0.00
Mobile Native Dynamic Maps	Cargas ilimitadas	\$0.00	\$0.00
Embed	Cargas ilimitadas	\$0.00	\$0.00
Embed Advanced	Hasta 14,000 cargas	\$14.00	\$11.20
Static Maps	Hasta 100,000 cargas	\$2.00	\$1.60
Dynamic Maps	Hasta 28,000 cargas	\$7.00	\$5.60
Static Street View	Hasta 28,000 panorámicas	\$7.00	\$5.60
Dynamic Street View	Hasta 14,000 panorámicas	\$14.00	\$11.20

*Figura 6.- Tabla de precios por el uso de Apis de Google.*

*Fuente: <https://cloud.google.com/maps-platform/pricing/sheet?hl=es-419>*



## CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

### 3.1. Análisis y discusión de los resultados

Esta investigación va dirigida a los administrativos y clientes de la Empresa Eléctrica Ambato Regional Centro Norte S.A. Con el fin de recolectar información, La Tabla 3 muestra el plan que se aplicará en el desarrollo del proyecto.

*Tabla 3. Plan de recolección de información.*

<b>Preguntas básicas</b>	<b>Explicación</b>
¿Para qué?	Para determinar si es necesario la implementación de una aplicación móvil híbrida, para la consulta de planillas eléctricas y geolocalización de las agencias de la EEASA.
¿De qué personas u objetos?	Personal administrativo y clientes de la EEASA
¿Quién?	Investigador
¿Cuándo?	marzo – agosto 2019
¿Dónde?	Empresa Eléctrica Ambato Regional Centro Norte S.A.
¿Cuántas veces?	1 vez
¿Qué técnicas de recolección?	Encuesta
¿Con qué?	Cuestionario

*Elaborado por: El investigador*

### 3.2. Procesamiento de la información

- 1) Elaboración del material para las encuestas.
- 2) Aplicar la encuesta en la EEASA.
- 3) Tabulación de la información obtenida.

- 4) Estudio estadístico de datos para presentación de resultados.

### 3.3. Análisis e interpretación de resultados

- 1) Análisis crítico de la información recolectada.
- 2) Interpretación de resultados.
- 3) Realizar conclusiones y recomendaciones.

A continuación, se presenta los resultados obtenidos de la encuesta realizada a la población establecida para el proyecto.

Para el análisis e interpretación de los resultados se muestra una tabla con la pregunta y opciones que tuvieron los encuestados, además de gráficos estadísticos los cuales serán útiles para la toma de decisiones y formular conclusiones y recomendaciones para la investigación.

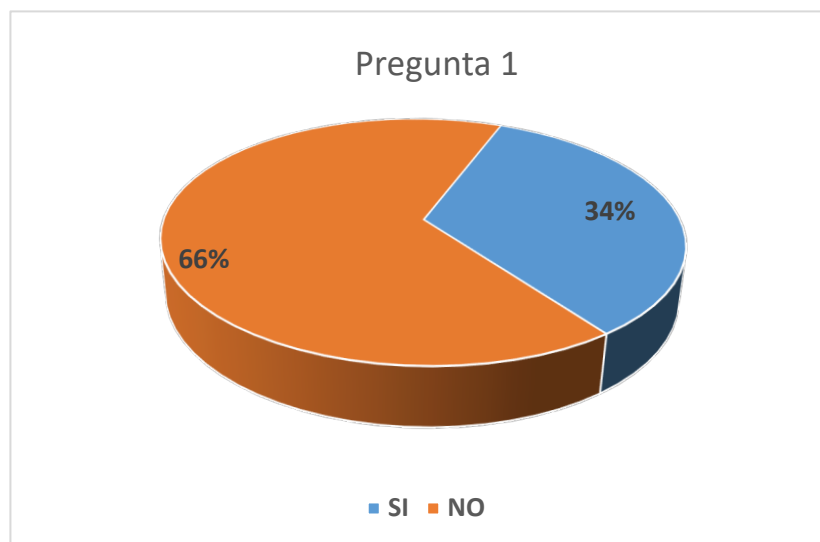
**Pregunta 1.** ¿Conoce todas las agencias de la empresa eléctrica donde puede realizar los pagos de sus planillas eléctricas?

*Tabla 4. Resultados Pregunta 1*

<b>Alternativas</b>	<b>Respuestas Obtenidas</b>	<b>Porcentajes %</b>
Si	17	34%
No	33	66%
<b>TOTAL</b>	50	100%

*Elaborado por: El investigador*

### Figura de los resultados de la Pregunta 1 mostrados en la Tabla 4



*Figura 7. Resultados Pregunta 1*

*Elaborado por: El investigador*

### **Análisis**

Las 50 personas encuestadas representan el 100%, como se muestra en el Figura 7 el 66% de las personas, es decir 33 encuestados no conocen las agencias que la EEASA ofrece para el pago de sus planillas eléctricas, y 17 usuarios que sería un 34% si conoce todas las agencias de la EEASA.

### **Interpretación**

Se determina que las personas que realizaron las encuestas en su mayoría no conocen todas las agencias que la EEASA pone a disposición de los usuarios para poder consultar y realizar los pagos de las planillas eléctricas.

**Pregunta 2.** ¿Cuántas veces usted acude a los puntos de información para consultar el valor de su planilla eléctrica?

Tabla 5. Resultados Pregunta 2

Alternativas	Respuestas Obtenidas	Porcentajes %
1	9	18 %
2	3	6 %
3	5	10 %
4	1	2 %
+ 4	32	64 %
<b>TOTAL</b>	50	100 %

Elaborado por: El investigador

Figura de los resultados de la Pregunta 2 mostrados en la Tabla 5

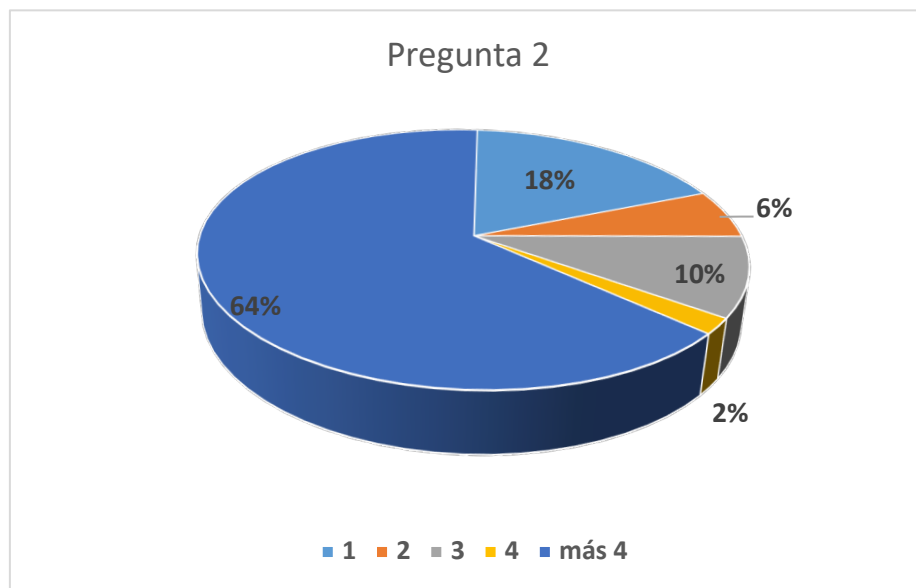


Figura 8. Resultados Pregunta 2

Elaborado por: El investigador

### Análisis

Las 50 personas encuestadas representan el 100%, la Figura 8 muestra que un 18% que equivale a 9 encuestados, ha visitado una sola vez los puntos de información de la EEASA, el 6%, es decir 3 encuestados visita dos veces los puntos de información de

la EEASA, el 10%, equivalente a 5 encuestados visita 3 veces los puntos de información de la EEASA, el 2% (1 encuestado) visita 4 veces los puntos de información de la EEASA y el 64% (32 encuestados) visitan más de 4 veces los puntos de información de la EEASA.

### Interpretación

Con los resultados obtenidos mediante la encuesta, la mayoría de personas encuestadas visitan más de 4 veces los puntos de información con el fin de consultar su planilla eléctrica y poder cancelarla a tiempo y evitar ser cortados el servicio, los encuestados que optaron por las otras opciones dicen que consultan sus planillas desde la página de la EEASA.

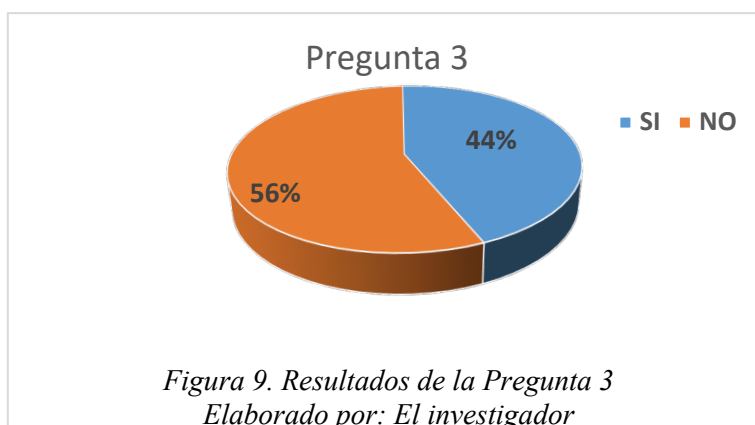
**Pregunta 3.** ¿Ha tenido problemas para consultar su planilla eléctrica en los puntos de información de la EEASA?

*Tabla 6. Resultados Pregunta 3*

Alternativas	Respuestas Obtenidas	Porcentajes %
Si	22	44 %
No	28	56 %
<b>TOTAL</b>	50	100 %

*Elaborado por: El investigador*

**Figura de los resultados de la Pregunta 3 mostrados en la Tabla 6**



## **Análisis**

Las 50 personas encuestadas representan el 100%, la Figura 9 indica que el 44% (22 encuestados) han tenido problemas con la pantalla informativa al momento de consultar las planillas eléctricas, mientras que un 54% (28 encuestados) dice que no se le ha presentado ningún problema a la hora de realizar las consultas.

## **Interpretación**

Según los resultados de la tabulación la diferencia de las respuestas de los encuestados no es muy grande, puesto que los puntos de información que dispone la EEASA también genera problemas como daños de la pantalla táctil, lo cual les dificulta a los usuarios obtener información de su planilla eléctrica.

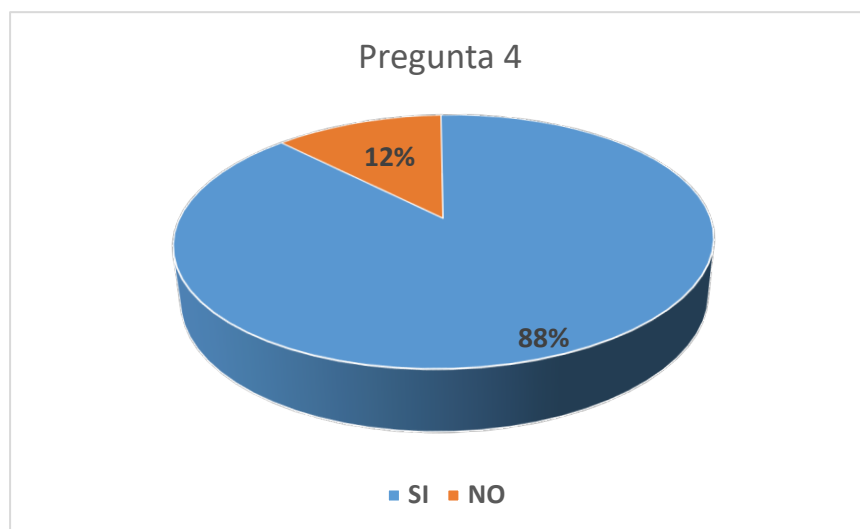
**Pregunta 4.** ¿Le gustaría saber el valor de su planilla eléctrica y donde realizar los pagos sin necesidad de acercarse a los puntos de información de la EEASA?

*Tabla 7. Resultados Pregunta 4*

<b>Alternativas</b>	<b>Respuestas Obtenidas</b>	<b>Porcentajes %</b>
Si	44	88 %
No	6	12 %
<b>TOTAL</b>	50	100 %

*Elaborado por: El investigador*

### Figura de los resultados de la Pregunta 4 mostrados en la Tabla 7



*Figura 10. Resultados Pregunta 4*

*Elaborado por: El investigador*

### **Análisis**

Las 50 personas encuestadas representan el 100%, la Figura 10 indica que el 88% (44 encuestados) requieren de nuevas maneras de consultar su planilla eléctrica y apenas el 12% (6 encuestados) manifiesta que no les gustaría tener otras formas de consultar las planillas eléctricas.

### **Interpretación**

Mediante la tabulación de las encuestas realizadas en la pregunta 4 la mayoría de usuarios quieren consultar las planillas eléctricas sin tener que acudir a los puntos de información que la EEASA tiene, ya que este procedimiento les evita hacer largas filas en las pantallas informativas.

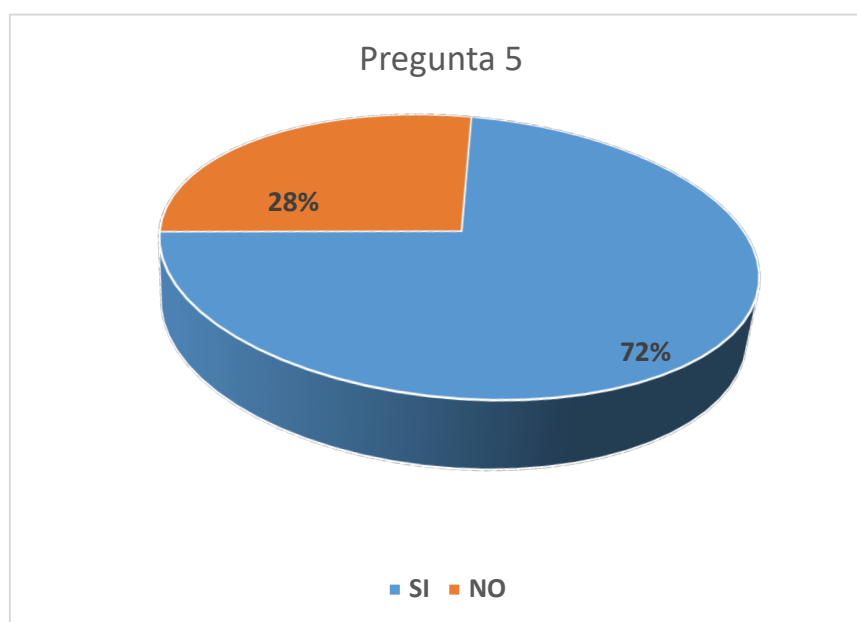
**Pregunta 5.** ¿Posee de un Smartphone en la actualidad?

*Tabla 8. Resultados Pregunta 5*

<b>Alternativas</b>	<b>Respuestas Obtenidas</b>	<b>Porcentajes %</b>
Si	36	72 %
No	14	28 %
<b>TOTAL</b>	50	100 %

*Elaborado por: El investigador*

**Figura de los resultados de la Pregunta 5 mostrados en la Tabla 8**



*Figura 11. Resultados Pregunta 5*

*Elaborado por: El investigador*

**Análisis**

Las 50 personas encuestadas representan el 100%, como se muestra en la Figura 11 el 88% (36 encuestados) tienen un teléfono inteligente indistintamente de su sistema operativo y el 28% (14 encuestados) no tienen teléfonos inteligentes.



## Interpretación

Se determina que la mayoría de encuestados si posee un teléfono inteligente el cual es indispensable para el desarrollo de esta investigación.

### Pregunta 6. ¿Qué Sistema Operativo tiene su Smartphone?

Tabla 9. Resultados Pregunta 6

Alternativas	Respuestas Obtenidas	Porcentajes %
IOS	15	30 %
Android	21	42 %
Windows Phone	0	0 %
Otros	14	28 %
<b>TOTAL</b>	<b>50</b>	<b>100 %</b>

Elaborado por: El investigador

### Figura de los resultados de la Pregunta 6 mostrados en la Tabla 9

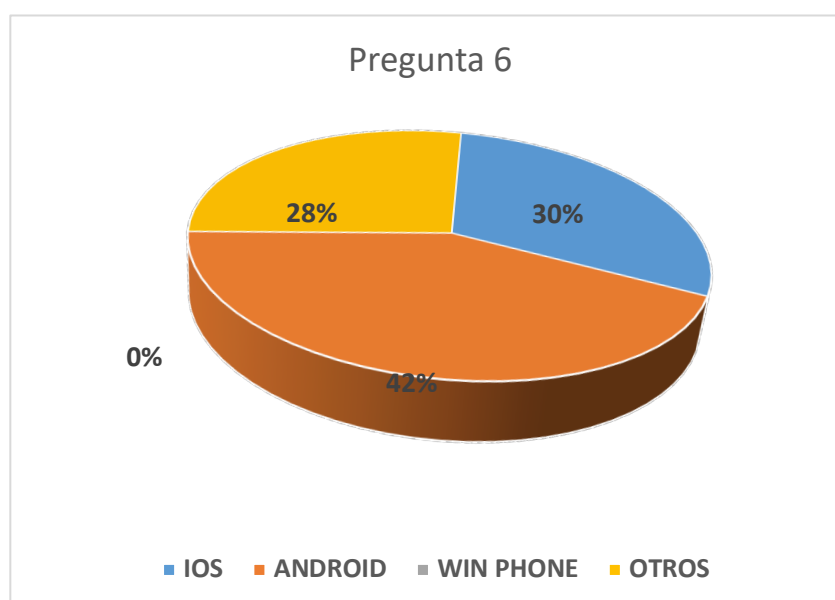


Figura 12. Resultados Pregunta 6

Elaborado por: El investigador

## **Análisis**

Las 50 personas encuestadas representan el 100%, la Figura 12 muestra que el 42% (21 encuestados) poseen un smartphone con sistema operativo Android, el 30% (15 encuestados) tienen un smartphone con sistema operativo IOS, ningún encuestado tiene teléfono inteligente con Windows Phone y el 28% (14 encuestados) contestaron otros porque son personas que no tiene smartphone.

## **Interpretación**

Se determina que los clientes de la EEASA, en su mayoría tienen teléfonos inteligentes tanto Android e IOS.

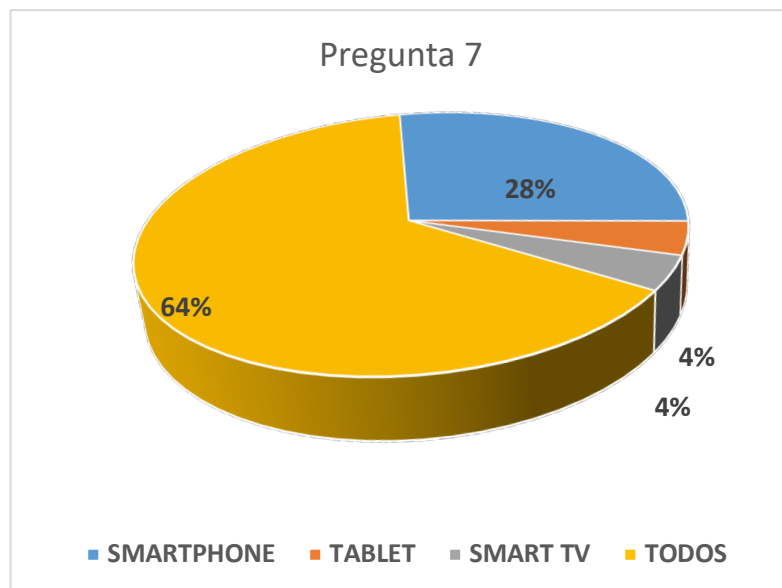
**Pregunta 7.** De la siguiente lista de dispositivos. ¿Cuál le gustaría usar para consultar su planilla eléctrica?

*Tabla 10. Resultados Pregunta 7*

<b>Alternativas</b>	<b>Respuestas Obtenidas</b>	<b>Porcentajes %</b>
Smartphone	14	28 %
Tablet	2	4 %
Smart TV	2	4 %
Todos los anteriores	32	64 %
<b>TOTAL</b>	50	100 %

*Elaborado por: El investigador*

**Figura de los resultados de la Pregunta 7 mostrados en la Tabla 10**



*Figura 13.- Resultados Pregunta 6*

*Elaborado por: El investigador*

### **Análisis**

Las 50 personas encuestadas representan el 100%, la Figura 13 indica que el 64% (32 encuestados) necesitan obtener información de la planilla eléctrica desde cualquier dispositivo inteligente, el 28% (14 encuestados) quieren consultar su planilla desde un smartphone, el 4% (2 encuestados) desde una tablet y el 4% (2 encuestados) desde un Smart TV.

### **Interpretación**

De los resultados obtenidos de la pregunta 7, todas las personas encuestadas están interesados en consultar su planilla eléctrica en dispositivos inteligentes, estos pueden ser smartphones, tablet, Smart TV entre otros.

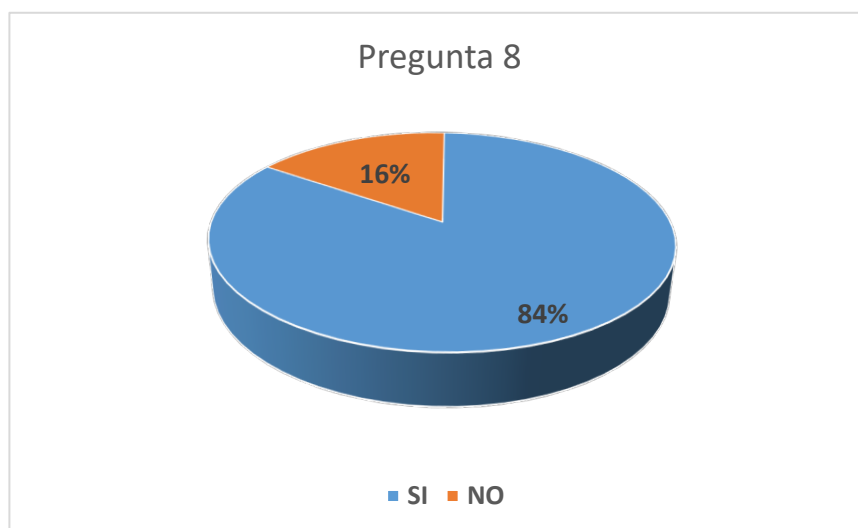
**Pregunta 8.** ¿Posee algún plan de internet, ya sea plan de datos móviles o para su hogar?

*Tabla 11. Resultados Pregunta 8*

<b>Alternativas</b>	<b>Respuestas Obtenidas</b>	<b>Porcentajes %</b>
Si	42	84 %
No	8	16 %
<b>TOTAL</b>	50	100 %

*Elaborado por: El investigador*

**Figura de los resultados de la Pregunta 8 mostrados en la Tabla 11**



*Figura 14. Resultados Pregunta 8*

*Elaborado por: El investigador*

### **Análisis**

Las 50 personas encuestadas representan el 100%, la Figura 14 indica que el 84% (42 encuestados) tienen acceso a internet y el 16% (8 encuestados) no tienen acceso a internet.

## **Interpretación**

Se determina que la mayoría de usuarios encuestados tienen la facilidad de acceder a internet, ya que poseen de un plan de internet este sea móvil o fijo, mismo que facilitará el desarrollo de la presente investigación.

Después de analizar e interpretar los resultados se concluye que es necesario implementar una aplicación móvil híbrida, para la consulta de planillas eléctricas y geolocalización de las agencias de la EEASA. Esta aplicación es de mucha importancia, motivo por el cual se presenta a continuación el desarrollo de la aplicación móvil, misma que beneficiará a todos los clientes y trabajadores de la empresa puesto que facilitara los procesos de consulta del valor de la planilla eléctrica y, además, conocer cuales son las agencias donde se puede realizar el pago respectivo.

### **3.4. Desarrollo del Proyecto**

La Empresa Eléctrica Ambato Regional Centro Norte S. A. desea desarrollar e implementar una aplicación móvil multiplataforma para la consulta de planillas y geolocalización de las agencias, tomando en cuenta la ficha de especificación de requerimientos establecidas en un comité informático previo a la inicialización del desarrollo del presente proyecto.

Las fichas de requerimientos están adjuntadas en la sección anexos, mismas que requieren:

- Aplicación Web para administrar los datos de las agencias.
- Servicios Web.
- Aplicación móvil multiplataforma.

### **3.4.1. Tecnología a utilizar para el desarrollo de la aplicación web administrativa**

Para el desarrollo de cualquier tipo de aplicación web en la Empresa Eléctrica Ambato se utiliza como lenguaje de desarrollo java, con el IDE de desarrollo Eclipse en cualquiera de sus versiones. Además de la tecnología de Google Web Toolkit y GXT el cual permite usar de componentes para construir interfaces gráficas. El gestor de base de datos utilizado es Oracle.

Para la realización de la aplicación web en el proyecto se utilizará las mismas tecnologías, ya que la EEASA cuenta con licencias, en el caso de Oracle, se utilizará el IDE Eclipse para la aplicación administrativa, con el fin que el personal a cargo del sistema pueda seguir dando soporte cuando el proyecto de investigación finalice.

### **3.4.2. Tecnología a utilizar para el desarrollo y alojamiento de los servicios web**

La EEASA cuenta con servidores cuyo servidor de aplicaciones configurado es WildFly, por tal razón los servicios web que se desarrollen para la realizar la aplicación móvil deben ser desarrollados bajo el lenguaje java, y alojados en WildFly.

### **3.4.3. Determinar el Framework para el desarrollo de la aplicación móvil multiplataforma**

Para determinar el framework que se va a utilizar para desarrollar la aplicación móvil, en el Capítulo II, sección 2.3 Fundamentación Teórica se investigó los frameworks más utilizados en la actualidad para el desarrollo de aplicaciones móviles híbridas.

En la Figura 15, se muestra los frameworks que fueron previamente investigados, y los mismos que se compararán en la Tabla 12 para tomar decisiones sobre el framework a utilizar.

## Ionic, React Native o Xamarin



Figura 15. Frameworks de desarrollo de aplicaciones híbridas

Elaborado por: El investigador

Para determinar el framework se ha puesto a consideración diversos puntos que son importantes para el desarrollador de la aplicación, los cuales se detallan a continuación en la siguiente Tabla.

Tabla 12. Tabla comparativa. Frameworks de desarrollo de aplicaciones híbridas

	<b>Ionic</b>	<b>React Native</b>	<b>Xamarin</b>
Lenguaje	<ul style="list-style-type: none"><li>- HTML</li><li>- CSS</li><li>- JavaScript</li></ul>	<ul style="list-style-type: none"><li>- JavaScript</li><li>- Java</li><li>- Objective C</li></ul>	<ul style="list-style-type: none"><li>- C#</li></ul>
Rendimiento	<ul style="list-style-type: none"><li>- Apariencia nativa</li></ul>	<ul style="list-style-type: none"><li>- Casi nativa</li></ul>	<ul style="list-style-type: none"><li>- Nativa</li></ul>
Interfaz	<ul style="list-style-type: none"><li>- Componentes propios</li></ul>	<ul style="list-style-type: none"><li>- Componentes nativos</li></ul>	<ul style="list-style-type: none"><li>- Componentes nativos</li></ul>
Comunidad	<ul style="list-style-type: none"><li>- Grande y activa</li></ul>	<ul style="list-style-type: none"><li>- Relativamente popular</li></ul>	<ul style="list-style-type: none"><li>- Poco popular</li></ul>

Compatibilidad de código	- 99%	- 90%	- 90%
Actualización Instantánea	- Si	- Si	- No

*Elaborado por: El investigador*

*Fuente: <https://openwebinars.net/blog/comparativa-react-native-y-diferentes-frameworks/>*

Aplicaciones como Facebook e Instagram fueron desarrolladas en React Native porque es un framework que genera aplicaciones lo más cercanas a nativas, motivo por el cual se necesita tener conocimientos de los lenguajes nativos como Java para Android y Objective C para iOS, al igual que Xamarin.

Xamarin, es una tecnología que utiliza recursos nativos, donde lo único que es híbrido es el código programado en la capa de acceso a datos, y para poder desarrollar en este framework se requiere aprender un nivel básico de programación en cada uno de los lenguajes nativos de cada plataforma, Porque al momento de utilizar componentes como la cámara, GPS y demás componentes propios de cada sistema operativo móvil se realiza mediante código nativo, es por esto que la interfaz se desarrolla por separado para cada plataforma.

Como framework de desarrollo para el proyecto se utilizará Ionic, ya que la línea de aprendizaje es fácil de aprender porque maneja lenguajes ya conocidos en el mundo de los desarrollos web como HTML, CSS, JavaScript, a diferencia de Xamarin que necesitamos aprender C# para poder realizar aplicaciones móviles híbridas.

Además, Ionic permite crear aplicaciones con una apariencia y fluidez más cercanas a las aplicaciones nativas. Gracias a que su interfaz gráfica se puede realizar en HTML podemos diseñar una interfaz amigable al usuario utilizando hojas de estilo



como complemento para un excelente diseño, el punto a favor que tiene Ionic es que permite actualización instantánea, es decir que se puede verificar los cambios que se le haga a las aplicaciones rápidamente ya que se puede visualizar sobre un navegador, a diferencia de Xamarin que se necesita compilar la aplicación cada vez que se realice cambios. Otra de las razones del uso de Ionic es que su código es 99% reutilizado en todas las plataformas móviles, React Native y Xamarin en este punto se quedan atrás porque manejan componentes nativos, haciendo de su código menos compatible con todos los sistemas operativos móviles.

Como metodología ágil para el desarrollo de este proyecto se utilizará Scrum, ya que es una metodología apta para trabajar en grupos de trabajo y se basa en la entrega incremental del producto.

Los miembros y roles del equipo SCRUM están conformados de la manera en que la Tabla 13 indica.

*Tabla 13. Definición del equipo Scrum.*

<b>Rol</b>	<b>Persona</b>	<b>Descripción</b>
Product Owner	Ing. Oscar Armas	Perteneciente al Departamento Comercial
Scrum Master	Ing. Diego Cadme	Perteneciente al Área Informática DRI
Development Team	Ing. Henry Cordova	Desarrollador
	Ing. Eduardo Carillo	Perteneciente al Departamento de Comunicación
	Sr. Brayan Cujano	Desarrollador

*Elaborado por: El investigador*

Para llevar a cabo este proyecto con la metodología SCRUM, necesitamos construir historias de usuario que se desarrollarán en los Sprint del proyecto. Para redactar las historias de usuario se usará formato expuesto en la Tabla 14:

*Tabla 14. Esquema historias de usuario.*

<b>Título de la historia de usuario</b>	
<b>Id</b>	
<b>Descripción</b>	
<b>Criterios de aceptación:</b>	<b>Prioridad:</b>
	<b>Estimación:</b>
	<b>Encargado:</b>

*Elaborado por: El investigador*

#### 3.4.4. Explicación del esquema de las historias de usuario

**Título de la historia de usuario:** es reemplazado por el nombre de la historia de usuario.

**Id:** Por lo general se usan números enteros empezando desde el 1, mismo que identifica a la historia de usuario.

**Descripción:** en este apartado se espera la explicación de la historia de usuario determinando el COMO, QUIERO Y PARA.

COMO: se escribe el rol (quién).

QUIERO: se escribe la tarea que se desea desarrollar (qué).

PARA: se escribe las funcionalidades de la historia de usuario (objetivo).

**Criterios de Aceptación:** son las condiciones que la historia de usuario debe cumplir para definir como terminada.

**Prioridad:** se define la importancia que tiene la historia de usuario en referencia a las otras.

**Estimación:** se estima el tiempo que durará la realización de la historia de usuario,

**Encargado:** persona del equipo a quien se le asigna la tarea de la historia de usuario.

### 3.4.5. Historias de usuario

La Tabla 15 expone la historia de usuario 1, correspondiente al diseño de la base de datos.

*Tabla 15. Historia de usuario 1*

<b>Diseño de la base de datos</b>	
<b>Id</b>	1
<b>Descripción</b>	YO como Product Owner QUIERO que se diseñe el modelo de la base de datos necesaria para el desarrollo de la aplicación PARA almacenar y administrar la información necesario que requerirá la aplicación.
<b>Criterios de aceptación:</b>  - <b>La base de datos está diseñada según los requerimientos dados.</b>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 2 días
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

La Tabla 16 expone la historia de usuario 2, correspondiente al desarrollo de la aplicación web para geolocalización de las agencias y pantalla de inicio.

Tabla 16. Historia de usuario 2

<b>Desarrollo de la aplicación web para geolocalización de las agencias y pantalla de inicio</b>	
<b>Id</b>	2
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle la aplicación web cliente PARA manipular la información de la base de datos creada.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- La aplicación web compila y ejecuta correctamente.</li> <li>- La aplicación web realiza las tareas esperadas</li> </ul>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 2 días
	<b>Encargado:</b> Brayan Cujano

Elaborado por: El investigador

La Tabla 17 expone la historia de usuario 3, correspondiente al desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio.

Tabla 17. Historia de usuario 3

<b>Desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio</b>	
<b>Id</b>	3
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle los servicios web PARA obtener información de la base de datos.
<b>Criterios de aceptación:</b>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 7 días

<ul style="list-style-type: none"> <li>- <b>Los servicios web devuelven la información correcta</b></li> <li>- <b>Los servicios web fueron realizados con el protocolo REST y devuelve los datos en formato JSON</b></li> </ul>	<b>Encargado:</b> Brayan Cujano
---	---------------------------------

*Elaborado por: El investigador*

La Tabla 18 expone la historia de usuario 4, correspondiente al desarrollo de servicios web para consulta de planillas eléctricas.

*Tabla 18. Historia de usuario 4*

<b>Desarrollo de servicios web para consulta de planillas eléctricas</b>	
<b>Id</b>	4
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle los servicios web PARA obtener información de la base de datos.
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- <b>Los servicios web devuelven la información correcta</b></li> <li>- <b>Los servicios web fueron realizados con el protocolo REST y devuelve los datos en formato JSON</b></li> </ul>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Ing. Henry Cordova

*Elaborado por: El investigador*

La Tabla 19 expone la historia de usuario 5, correspondiente a la preparación del entorno de desarrollo y creación de la aplicación móvil.

*Tabla 19. Historia de usuario 5.*

<b>Preparación del entorno de desarrollo y creación de la aplicación móvil</b>	
<b>Id</b>	5
<b>Descripción</b>	YO como Product Owner QUIERO que se prepare el entorno de desarrollo y además crear la aplicación móvil PARA empezar con el desarrollo de la aplicación móvil híbrida, para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato
<b>Criterios de aceptación:</b>  - <b>El entorno de desarrollo y la aplicación estén listos para trabajar</b>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 2 día
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

La Tabla 20 expone la historia de usuario 6, correspondiente a la edición de imágenes

*Tabla 20. Historia de usuario 6*

<b>Edición de imágenes</b>	
<b>Id</b>	6
<b>Descripción</b>	YO como Product Owner QUIERO que editen las imágenes de las agencias e imágenes informativas PARA utilizar en la aplicación móvil.

<b>Criterios de aceptación:</b> - Las imágenes coinciden con lo solicitado.	<b>Prioridad:</b> Media
	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Ing. Eduardo Carrillo

*Elaborado por: El investigador*

La Tabla 21 expone la historia de usuario 7, correspondiente al desarrollo de la pantalla de geolocalización de las agencias.

*Tabla 21. Historia de usuario 7*

<b>Desarrollo de la pantalla de geolocalización de las agencias</b>	
<b>Id</b>	7
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle la pantalla de geolocalización de las agencias PARA que los usuarios estén informados de los lugares, horarios y días en los que puede efectuar los pagos
<b>Criterios de aceptación:</b> - La pantalla obtiene las coordenadas geográficas de las agencias y muestra con markers la ubicación en el mapa. - La pantalla contenga un botón que muestre el listado de todas las agencias con su respectivo nombre, dirección y horario de atención.	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

La Tabla 22 expone la historia de usuario 8, correspondiente al desarrollo de la pantalla de inicio.

*Tabla 22. Historia de usuario 8*

<b>Desarrollo de la pantalla de inicio</b>	
<b>Id</b>	8
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle la pantalla de inicio PARA mantener a los clientes informados sobre la EEASA
<b>Criterios de aceptación:</b>  - <b>La pantalla de inicio concuerda con el bosquejo dado en la Historia de usuario N° 5</b>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

La Tabla 23 expone la historia de usuario 9, correspondiente al desarrollo del módulo para obtener datos del cliente.

*Tabla 23. Historia de usuario 9*

<b>Desarrollo del módulo para obtener datos del cliente</b>	
<b>Id</b>	9
<b>Descripción</b>	YO como Product Owner QUIERO que se desarrolle el módulo de consulta de planillas eléctricas PARA que el cliente visualice el valor de la planilla eléctrica del mes correspondiente.
<b>Criterios de aceptación:</b>	<b>Prioridad:</b> Alta



- <b>El módulo brinde la información exacta al cliente</b>	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

La Tabla 24 expone la historia de usuario 10, correspondiente a Implementar la aplicación móvil en la Empresa Eléctrica Ambato.

*Tabla 24. Historia de usuario 10*

<b>Desarrollo del módulo para consulta de planillas</b>	
<b>Id</b>	10
<b>Descripción</b>	YO como Product Owner QUIERO implementar la aplicación móvil en la EEASA PARA que los usuarios puedan hacer uso de la misma y mantenerse informado de donde y cuanto pagar por el consumo eléctrico.
<b>Criterios de aceptación:</b>  - <b>El módulo brinde la información exacta al cliente</b>	<b>Prioridad:</b> Alta
	<b>Estimación:</b> 15 días
	<b>Encargado:</b> Brayan Cujano

*Elaborado por: El investigador*

Las historias de usuario generan un Product Backlog, el cual contiene un listado de las tareas que se intenta desarrollar mientras se ejecuta el proyecto.

### 3.4.6. Desarrollo del Product Backlog

Como lo indica la metodología SCRUM se debe desarrollar el Product Backlog con las funcionalidades, prioridad y estimaciones en tiempo que se necesite para la ejecución de los mismos. La Tabla 25 muestra la estimación de tiempo en semanas para desarrollar cada historia de usuario.

*Tabla 25. Product Backlog.*

<b>Id</b>	<b>Título</b>	<b>Prioridad</b>	<b>Estimación (semanas)</b>
1	Diseño de la base de datos	10	0.33 semanas
2	Desarrollo de la aplicación web para geolocalización de las agencias y pantalla de inicio	9	0.33 semana
3	Desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio	8	1 semanas
4	Desarrollo de servicios web para consulta de planillas eléctricas	7	2 semanas
5	Edición de imágenes	6	2 semanas
6	Preparación del entorno de desarrollo y creación de la aplicación móvil.	5	0.33 semana
7	Desarrollo de la pantalla de geolocalización de las agencias	4	2 semanas
8	Desarrollo de la pantalla de inicio	3	2 semanas
9	Desarrollo del módulo para consultas de la planilla eléctrica.	2	3 semanas
10	Implementar la aplicación móvil en la EEASA	1	2 semanas

*Elaborado por: El investigador*

### 3.4.7. Asignación del Product Backlog a los Sprints

La Tabla 26 muestra la estimación de tiempos para la ejecución de cada Sprint, tal y como dice la teoría, cada iteración tendrá un tiempo de duración máximo de un mes.

*Tabla 26. Product Backlog según su Sprint.*

<b>Nombre Sprint</b>	<b>Id</b>	<b>Título</b>	<b>Estimación (semanas)</b>
Sprint 1	1	Diseño de la base de datos.	2 semanas
	2	Desarrollo de la aplicación web para geolocalización de las agencias y pantalla de inicio.	
	3	Desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio.	
Sprint 2	4	Desarrollo de servicios web para consulta de planillas eléctricas.	4 semanas
	5	Edición de imágenes.	
	6	Preparación del entorno de desarrollo y creación de la aplicación móvil.	
Sprint3	7	Desarrollo de la pantalla de geolocalización de las agencias.	4 semanas
	8	Desarrollo de la pantalla de inicio.	
Sprint4	9	Desarrollo del módulo para consultas de la planilla eléctrica.	3 semanas
Sprint5	10	Implementar la aplicación móvil en la EEASA.	2 semanas

*Elaborado por: El investigador*

### 3.4.8. Estimación de las fechas de entrega para cada Sprint

Los tiempos expuestos en la Tabla 27 solo será un estimado para la entrega de cada iteración de la aplicación móvil.

Tabla 27. Estimación de las fechas de entrega para cada Sprint.

Nombre Sprint	Fecha inicio	Fecha fin	Fecha de entrega
Sprint 1	1 de agosto del 2019	15 de agosto del 2019	26 de agosto del 2019
Sprint 2	17 de agosto del 2019	16 de septiembre del 2019	26 de septiembre del 2019
Sprint 3	13 de octubre del 2019	12 de noviembre del 2019	13 de noviembre del 2019
Sprint4	15 de diciembre del 2019	12 de diciembre del 2019	13 de diciembre del 2019
Sprint5	6 de enero del 2020	20 de enero del 2019	21 de enero del 2019

Elaborado por: El investigador

### 3.4.9. Desarrollo del Sprint 1

Nombre Sprint	Id	Título
Sprint 1	1	Diseño de la base de datos
	2	Desarrollo de la aplicación web
	3	Desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio

Tabla 28. Desarrollo del Sprint 1.

Elaborado por: El investigador

## Desarrollo del Sprint 1 mostrado en la Tabla 28

### Id 1. Diseño de la Base de Datos

Para el diseño de la base de datos se ha utilizado PowerDesigner, porque es una herramienta que facilita la construcción de una base de datos sólida, desde la más pequeña hasta la más grande. Además, permite exportar scripts del modelo de base de datos creada para poder ejecutarlos en un servidor de base de datos, este puede ser cualquiera, para el desarrollo de este proyecto utilizaremos Oracle 12c.

La Figura 16 muestra el diseño de la base de datos usada para la administración de datos que serán manipulados mediante la aplicación móvil que se desarrollará.

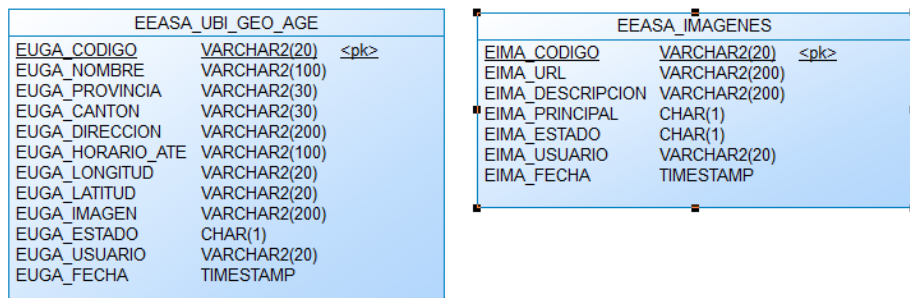


Figura 16. Modelo de la Base de Datos.

Elaborado por: El investigador

### Id 2. Desarrollo de la aplicación web

#### Software Requerido

- Eclipse Kepler versión 4.3.
- Google Web Toolkit (GWT) versión gwt-2.5.1.
- GXT versión gxt-2.2.5.
- Java jdk versión 7u7.
- Mozilla Firefox versión 24.0.

## Instalar el plugin GWT para Eclipse Kepler

1. En la barra de herramientas dar clic en el menú **Help** y otro clic en el submenú **Install New Software**. Ver Figura 17.

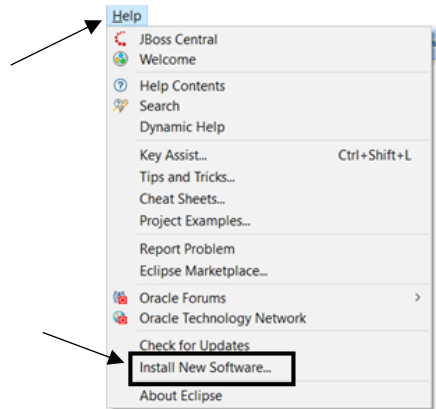


Figura 17. Instalar nuevo software.

Elaborado por: El investigador

2. En la ventana siguiente dar clic en Add, después aparece otra ventana y llenamos los datos como **Name** y **Location** y dar clic en **OK**. Ver Figura 18.

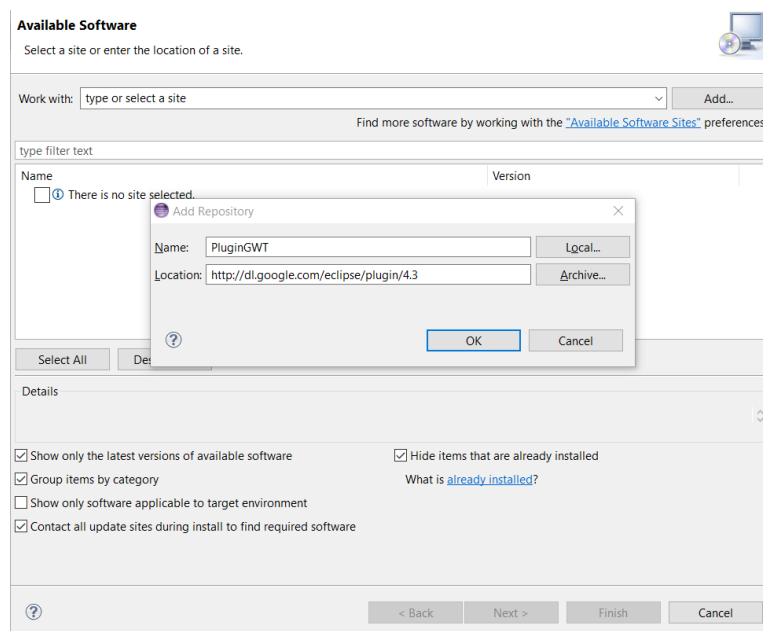


Figura 18. Añadir Repositorio.

Elaborado por: El investigador

3. Luego seleccionar solo las opciones del plugin de GWT y dar clic en **Next**, aceptar los términos y dar clic en **Finish**. Seguidamente reiniciar el IDE y podremos visualizar los iconos de GWT como se muestra en la Figura 19.

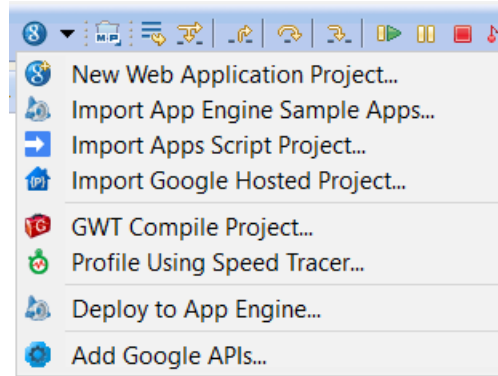


Figura 19. Iconos de GWT.

Elaborado por: El investigador

### Creación de un nuevo proyecto

- Para crear un nuevo proyecto dar clic en el menú **File, New y Web Application Project**, como la Figura 20 indica.

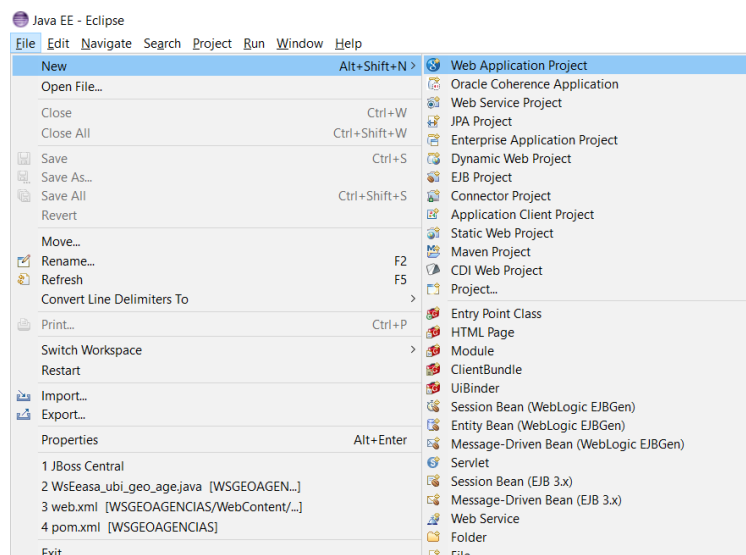


Figura 20. Creación de un nuevo proyecto.

Elaborado por: El investigador

- Después otorgar un nombre para proyecto, el package del proyecto y desactivar la opción de **Use Google App Engine** y dar clic en **Finish**.

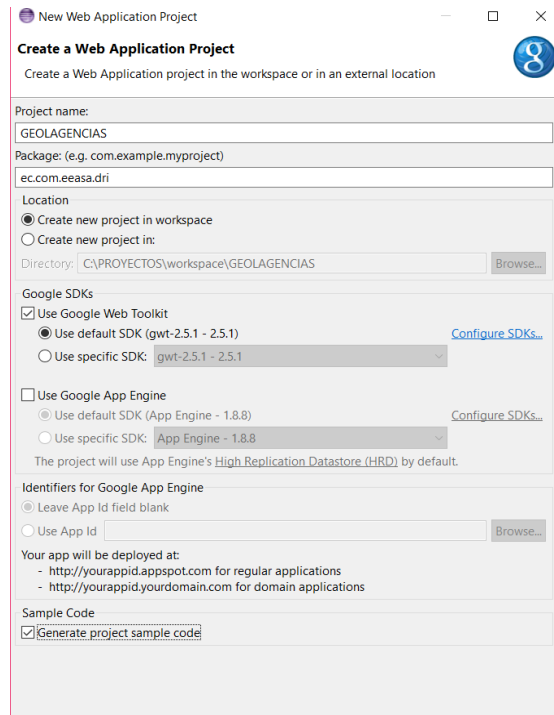


Figura 21. Datos para crear proyecto nuevo.

Elaborado por: El investigador

## Configuración del proyecto creado

- Para configurar el proyecto dar clic derecho sobre el proyecto y buscar la opción de **Properties** y dar clic.
- En la opción de **Project Facets** seleccionar los checkboxes, de tal manera que quede tal y cual muestra la figura 22.
- Presionar **OK** listo.



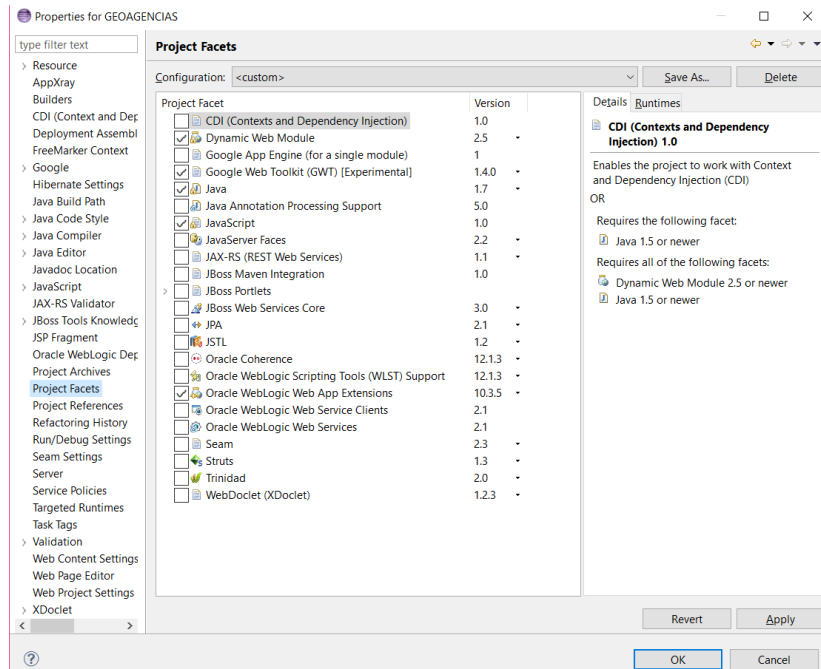


Figura 22. Configuración Project Facets.

Elaborado por: El investigador

- Ahora en la opción **Build Path** importar todas las librerías que vayamos a ocupar en el proyecto, incluido el ojdbc7 que será necesario para comunicarse con la base de datos Oracle.

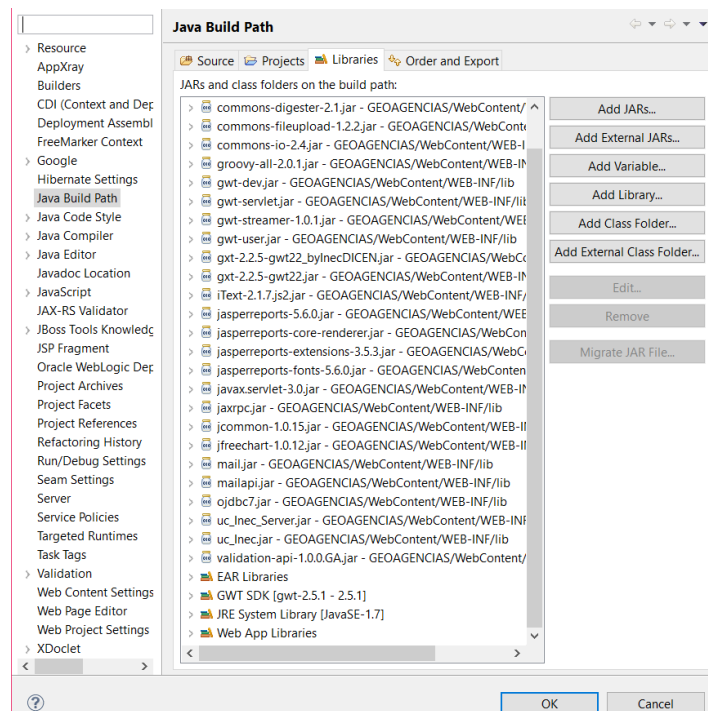


Figura 23. Configuración Java Build Path.

Elaborado por: El investigador

## Estructura del proyecto web creado

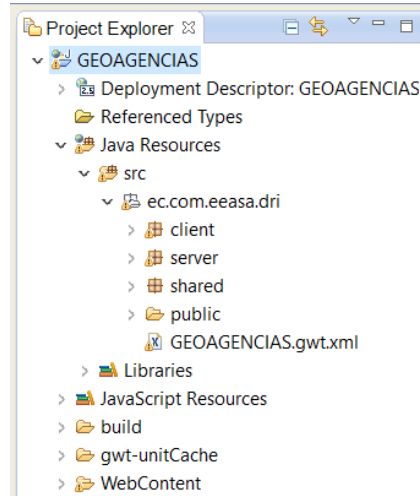


Figura 24. Estructura del proyecto.

Elaborado por: El investigador

Explicación de la estructura del proyecto web que indica la Figura 24. Donde:

**GEOAGENCIAS:** es la carpeta contenedora de todas las clases cuyo nombre lo especificamos cuando creamos el proyecto ver Figura 21.

**src:** contiene los paquetes necesarios para la aplicación.

**ec.com.eeasa.dri.client:** este paquete contiene la clase principal de la aplicación, el cual se ejecutara en primera instancia cuando el proyecto se ponga en ejecución. Además, contiene la interfaz asíncrona la cual se encarga de comunicarse con ec.com.eeasa.dri.server.

**ec.com.eeasa.dri.server:** contiene la clase Implement la cual se encarga de comunicarse con ec.com.eeasa.dri.client.

**ec.com.eeasa.dri.shared:** en este paquete contiene los objetos de la base de datos con todas sus propiedades y métodos set y get.

**WebContent:** contiene el archivo de configuración.xml, y una página inicial esta puede ser: .jsp, .html, .xhtml, htm, php, etc... Además, contiene las clases .js que se generan automáticamente cuando se ejecuta el proyecto, las mismas que se encargan de transformar el código de las interfaces echas en java a código JavaScript para que sea entendido por un navegador web.

## Ejecutar un proyecto GWT

1. Clic derecho en el proyecto, en la Opción de **Run As** escoger **Web Application**. Ver Figura 25.

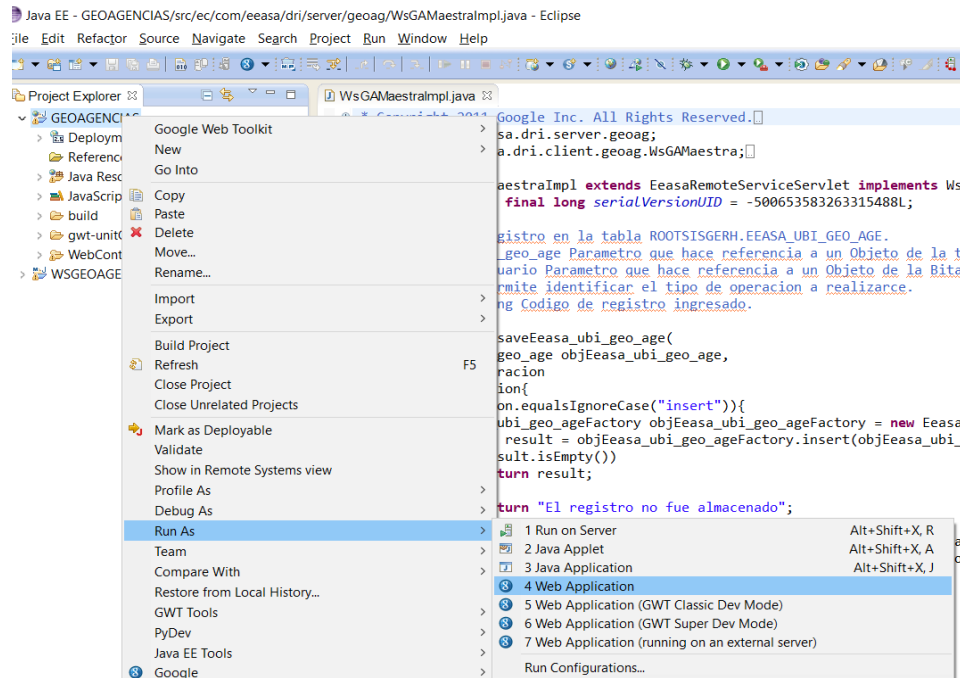


Figura 25. Ejecutar un proyecto.

Elaborado por: El investigador

Para ejecutar el proyecto necesitamos Firefox versión 24.0, ya que incluye características útiles para la construcción de proyectos web. Cuando ejecutemos el proyecto por primera vez Firefox pedirá instalar el plugin de GWT, para la cual el navegador nos provee de un botón para realizar la instalación, ver Figura 26, instalamos el plugin, reiniciamos el navegador y listo, ahora podemos observar la aplicación creada.

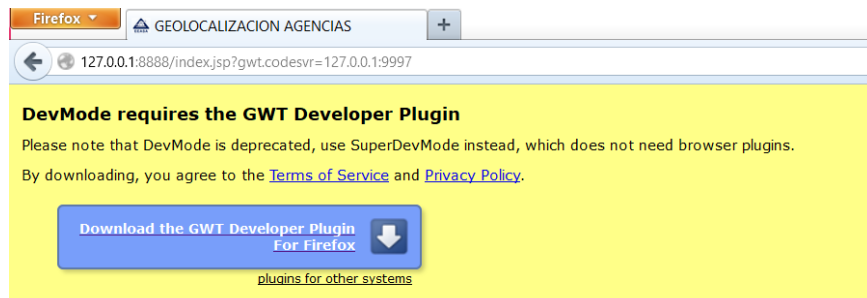


Figura 26. Instalar plugin de GWT para Firefox.

Elaborado por: El investigador

Una vez listo nuestro IDE de programación se procede a crear las clases y métodos para el correcto funcionamiento de la aplicación web y así poder pasar los criterios de aceptación definidos en la Historia de Usuario N° 3.

Para el desarrollo de la aplicación web, es necesario crear la cadena de conexión, la cual se encargará de dar respuesta a las peticiones que la aplicación web realice. La cadena de conexión se la realizará con la base de datos Oracle, para eso se necesita una clase llamada **Coneccion** y configurar el archivo **web.xml**.

### Clase Coneccion

```
public Coneccion(String _ip, String _bd, String _usr, String _pass,
String _puerto) throws Exception {

    this.ip=_ip;
    this.bd=_bd;
    this.usr= _usr;
    this.pass= _pass;
    this.puerto = _puerto;
    this.LookupName="";
    flagWeLogic=false;
}

public Coneccion(String inLookupName) throws Exception {

    this.LookupName=inLookupName;
    flagWeLogic=true;
}

public Connection getConexion() throws Exception {
    if(myConnection==null || myConnection.isClosed())
    {
        try {
            if(LookupName.isEmpty() && flagWeLogic==false)
            {
                Properties prop = new Properties();
                prop.put("user", usr);
                prop.put("password", pass);

                Class.forName("oracle.jdbc.OracleDriver");
```

```

        myConnection =
(OracleConnection)DriverManager.getConnection(getJdbcUrlOracle(ip, this.bd,
this.puerto), prop);
    }
    else
    {
        if(LookupName.isEmpty())
            throw new Exception( "No se ha agregado
el parametro de LookupName en web.xml");
        InitialContext myInitialContext = new
InitialContext();
        DataSource myDataSource = (DataSource)
myInitialContext.lookup(LookupName);
        myConnection = myDataSource.getConnection();
    }
}
catch (SQLException er) {
    String mgs= er.getMessage();
    if(mgs.isEmpty()&&flagWeLogic)
        throw new Exception( "No se ha agregado el
parametro de LookupName en web.xml");
    else
        throw new Exception( mgs+"\n"+"No existe
conexión con el servidor de Base de Datos" );
}
}
return myConnection;
}
private String getJdbcUrlOracle(String ip, String bd, String prto) {
    if(ip.length() == 0) ip = "localhost";
    String databaseName = bd;
    String url = (new
StringBuilder("jdbc:oracle:thin:@")).append(ip).append(":").append(prto).append(":")
).toString();
    if(databaseName.length() > 0)
        url = (new
StringBuilder(String.valueOf(url))).append(databaseName).toString();
    else
        url = (new
StringBuilder(String.valueOf(url))).append("MYDATABASE").toString(); return url;
}
}
}

```

## Web.xml

```
<web-app id="WebApp_ID" version="2.5"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name>GeoAgencias</display-name>
  <context-param>
    <param-name>LookupName</param-name>
    <param-value>GeoAgenciasDS</param-value>
  </context-param>
  <context-param>
    <param-name>db_Base</param-name>
    <param-value>orcl</param-value>
  </context-param>
  <context-param>
    <param-name>db_Clave</param-name>
    <param-value>GEOAGENCIAS2019</param-value>
  </context-param>
  <context-param>
    <param-name>db_Host</param-name>
    <param-value>localhost</param-value>
  </context-param>
  <context-param>
    <param-name>db_Usuario</param-name>
    <param-value>GEOAGENCIAS</param-value>
  </context-param>
  <context-param>
    <param-name>db_puerto</param-name>
    <param-value>1521</param-value>
  </context-param>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

## Parámetros del archivo de configuración web.xml

LookupName: nombre del servicio de configuración para utilizar en el servidor WebLogic.

db\_Base: nombre de la base de datos global, este nombre se especifica en instalar el servidor de base de Datos.

db\_Usuario: el nombre del usuario de base de datos que hayamos creado previamente.

db\_Clave: la clave del usuario de la base de datos.

db\_Host: se especifica la dirección IP del computador donde está instalado el servidor de base de datos.

db\_puerto: el puerto en el que se instaló el servidor de base de datos.

Para el desarrollo de este Sprint se documentará la programación de las clases necesarias para manipular la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE, ya que la lógica de programación de la clase GEOAGENCIAS.EEASA\_IMAGENES es la misma.

Programación de la clase que contiene los métodos setters, getters y propiedades para administrar la información de la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE.

```
public class Eeasa_ubi_geo_age implements Serializable{
    private static final long serialVersionUID = 3375509497120637013L;
    private String EUGA_CODIGO;
    private String EUGA_NOMBRE;
    private String EUGA_PROVINCIA;
    private String EUGA_CANTON;
    private String EUGA_DIRECCION;
    private String EUGA_HORARIO_ATE;
    private String EUGA_LONGITUD;
    private String EUGA_LATITUD;
    private String EUGA_IMAGEN;
    private String objEeasa_ubi_geo_age;
    /**
     * Constructor que permite la creacion del objeto vacio de
Eeasa_ubi_geo_age.
     */
    public Eeasa_ubi_geo_age(){
    }
    public Eeasa_ubi_geo_age(
        String inEuga_codigo,
        String inEuga_nombre,
```

```

        String inEuga_provincia,
        String inEuga_canton,
        String inEuga_direccion,
        String inEuga_horario_ate,
        String inEuga_longitud,
        String inEuga_latitud,
        String inEuga_imagen){
            this.setEuga_codigo(inEuga_codigo);
            this.setEuga_nombre(inEuga_nombre);
            this.setEuga_provincia(inEuga_provincia);
            this.setEuga_canton(inEuga_canton);
            this.setEuga_direccion(inEuga_direccion);
            this.setEuga_horario_ate(inEuga_horario_ate);
            this.setEuga_longitud(inEuga_longitud);
            this.setEuga_latitud(inEuga_latitud);
            this.setEuga_imagen(inEuga_imagen);
        }
    /**
     * @return (String) EUGA_CODIGO
     */
    public String getEuga_codigo(){
        return EUGA_CODIGO;
    }
    public String getEuga_nombre(){
        return EUGA_NOMBRE;
    }
    /**
     * @return (String) EUGA_PROVINCIA
     */
    public String getEuga_provincia(){
        return EUGA_PROVINCIA;
    }
    /**
     * @return (String) EUGA_CANTON
     */
    public String getEuga_canton(){
        return EUGA_CANTON;
    }
    /**
     * @return (String) EUGA_DIRECCION
     */
    public String getEuga_direccion(){
        return EUGA_DIRECCION;
    }

```



```

}
/**
 * @return (String) EUGA_HORARIO_ATE
 */
public String getEuga_horario_ate(){
    return EUGA_HORARIO_ATE;
}
/**
 * @return (String) EUGA_LONGITUD
 */
public String getEuga_longitud(){
    return EUGA_LONGITUD;
}
/**
 * @return (String) EUGA_LATITUD
 */
public String getEuga_latitud(){
    return EUGA_LATITUD;
}
/**
 * @return (String) EUGA_IMAGEN
 */
public String getEuga_imagen(){
    return EUGA_IMAGEN;
}
/**
 * @param (String) inEuga_codigo
 */
public void setEuga_codigo(String inEuga_codigo ){
    this.EUGA_CODIGO = inEuga_codigo;
}
/**
 * @param (String) inEuga_nombre
 */
public void setEuga_nombre(String inEuga_nombre ){
    this.EUGA_NOMBRE = inEuga_nombre;
}
/**
 * @param (String) inEuga_provincia
 */
public void setEuga_provincia(String inEuga_provincia ){
    this.EUGA_PROVINCIA = inEuga_provincia;
}
}

```

```

/**
 * @param (String) inEuga_canton
 */
public void setEuga_canton(String inEuga_canton ){
    this.EUGA_CANTON = inEuga_canton;
}
/**
 * @param (String) inEuga_direccion
 */
public void setEuga_direccion(String inEuga_direccion ){
    this.EUGA_DIRECCION = inEuga_direccion;
}
/**
 * @param (String) inEuga_horario_ate
 */
public void setEuga_horario_ate(String inEuga_horario_ate ){
    this.EUGA_HORARIO_ATE = inEuga_horario_ate;
}
/**
 * @param (String) inEuga_longitud
 */
public void setEuga_longitud(String inEuga_longitud ){
    this.EUGA_LONGITUD = inEuga_longitud;
}
/**
 * @param (String) inEuga_latitud
 */
public void setEuga_latitud(String inEuga_latitud ){
    this.EUGA_LATITUD = inEuga_latitud;
}
/**
 * @param (String) inEuga_imagen
 */
public void setEuga_imagen(String inEuga_imagen ){
    this.EUGA_IMAGEN = inEuga_imagen;
}
/**
 * Asigna una objeto Eeasa\_ubi\_geo\_age con los campos de la tabla
ROOTSISGERH.EEASA_UBI_GEO_AGE.
 * @param String
 */
public void setEeasa_ubi_geo_age(String inEeasa_ubi_geo_age){
    this.objEeasa_ubi_geo_age = inEeasa_ubi_geo_age;
}

```

```

    }
    /**
     * Retorna un objeto Eeasa_ubi_geo_age con los campos de la tabla
    ROOTSISGERH.EEASA_UBI_GEO_AGE.
     * @return String
     */
    public String getEeasa_ubi_geo_age(){
        return this.objEeasa_ubi_geo_age;
    }
}

```

Creación de la Clase **WsGAMaestraImpl**, la cual se utiliza para programar toda la lógica de negocio de la aplicación.

```

public class WsGAMaestraImpl extends EeasaRemoteServiceServlet implements
WsGAMaestra {
    private static final long serialVersionUID = -500653583263315488L;
    /**
     * Guarda un registro en la tabla ROOTSISGERH.EEASA_UBI_GEO_AGE.
     */
    public String saveEeasa_ubi_geo_age(
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age,
        String operacion
    )throws Exception{
        if(operacion.equalsIgnoreCase("insert")){
            Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
            String result =
objEeasa_ubi_geo_ageFactory.insert(objEeasa_ubi_geo_age);
            if(!result.isEmpty())
                return result;
            else
                return "El registro no fue almacenado";
        }else{
            Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
            int result =
objEeasa_ubi_geo_ageFactory.update(objEeasa_ubi_geo_age);
            if(result==0)
                return "El registro no fue modificado";
            else
                return "El registro fue modificado correctamente";
        }
    }
}

```

```

        }
    }
    /**
     * Elimina un registro en la tabla ROOTSISGERH.EEASA\_UBI\_GEO\_AGE.
     */
    public String deleteEeasa_ubi_geo_age(
        String inDeleteType,
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
        int result =
objEeasa_ubi_geo_ageFactory.delete(inDeleteType,objEeasa_ubi_geo_age);
        if(result==0)
            return "El registro no fue eliminado";
        else
            return "El registro fue eliminado correctamente";
    }
    /**
     * Obtiene un registro en la tabla ROOTSISGERH.EEASA\_UBI\_GEO\_AGE.
     */
    public ArrayList<HashMap<String, Object>> selectEeasa_ubi_geo_age(
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
        return objEeasa_ubi_geo_ageFactory.select(objEeasa_ubi_geo_age);
    }
    /**
     * Obtiene todos los registros de la tabla
     */
    public ArrayList<HashMap<String, Object>>
selectAllEeasa_ubi_geo_age()throws Exception{
        Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
        return objEeasa_ubi_geo_ageFactory.selectAll();
    }
    /**
     * Obtiene un vector dinamico paginado de los objetos equivalentes a los
registros de la tabla ROOTSISGERH.EEASA\_UBI\_GEO\_AGE.
     */
    public ArrayList<HashMap<String, Object>> selectPagedEeasa_ubi_geo_age(
        Integer inOffset,

```

```

        Integer inLimit,
        String inOrderBy,
        String inDirection,
        String inFilterText
    )throws Exception{
        Eeasa_ubi_geo_ageFactory objEeasa_ubi_geo_ageFactory = new
Eeasa_ubi_geo_ageFactory(getDB());
        return objEeasa_ubi_geo_ageFactory.selectPaged(
            inOffset,
            inLimit,
            inOrderBy,
            inDirection,
            inFilterText
        );
    }
}

```

Esta clase es dependiente de los métodos que tiene la clase **Eeasa\_ubi\_geo\_ageFactory**, misma que posee los métodos de llamada a los paquetes creados en la base de datos.

### Clase Eeasa\_ubi\_geo\_ageFactory

```

public class Eeasa_ubi_geo_ageFactory {
    /**
     *Definicion de la variable de conexion.
     */
    DataBaseObject dbo;
    public Eeasa_ubi_geo_ageFactory (DataBaseObject inDbo)
    {
        dbo=inDbo;
    }
    /**
     * Inserta un registro en la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE.
     */
    public String insert(
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        String sql_fun =
"GEOAGENCIAS.PKG\_EEASA\_UBI\_GEO\_AGE.F\_INSERT\(?,?,?,?,?,?,?,?\)";
        ArrayList<Object> parametros = new ArrayList<Object>();

```

```

        parametros.add(objEeasa_ubi_geo_age.getEuga_codigo());
        parametros.add(objEeasa_ubi_geo_age.getEuga_nombre());
        parametros.add(objEeasa_ubi_geo_age.getEuga_provincia());
        parametros.add(objEeasa_ubi_geo_age.getEuga_canton());
        parametros.add(objEeasa_ubi_geo_age.getEuga_direccion());
        parametros.add(objEeasa_ubi_geo_age.getEuga_horario_ate());
        parametros.add(objEeasa_ubi_geo_age.getEuga_longitud());
        parametros.add(objEeasa_ubi_geo_age.getEuga_latitud());
        parametros.add(objEeasa_ubi_geo_age.getEuga_imagen());
        Object msg = dbo.SQL_CALL_FUNCTION(sql_fun,parametros);
        return msg.toString();
    }
    /**
     * Modifica un registro en la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE.
     */
    public int update(
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        String sql_fun =
"GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE.F_UPDATE(?,?,?,?,?,?,?,?)";
        ArrayList<Object> parametros = new ArrayList<Object>();
        parametros.add(objEeasa_ubi_geo_age.getEuga_codigo());
        parametros.add(objEeasa_ubi_geo_age.getEuga_nombre());
        parametros.add(objEeasa_ubi_geo_age.getEuga_provincia());
        parametros.add(objEeasa_ubi_geo_age.getEuga_canton());
        parametros.add(objEeasa_ubi_geo_age.getEuga_direccion());
        parametros.add(objEeasa_ubi_geo_age.getEuga_horario_ate());
        parametros.add(objEeasa_ubi_geo_age.getEuga_longitud());
        parametros.add(objEeasa_ubi_geo_age.getEuga_latitud());
        parametros.add(objEeasa_ubi_geo_age.getEuga_imagen());
        Object filasAfectadas = dbo.SQL_CALL_FUNCTION(sql_fun,parametros);
        return Integer.valueOf(filasAfectadas.toString());
    }
    /**
     * Elimina un registro en la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE.
     */
    public int delete(
        String inDeleteType,
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        String sql_fun = "GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE.F_DELETE(?,?)";
        ArrayList<Object> parametros = new ArrayList<Object>();
        parametros.add(inDeleteType);

```

```

        parametros.add(objEeasa_ubi_geo_age.getEuga_codigo());
        Object filasAfectadas = dbo.SQL_CALL_FUNCTION(sql_fun,parametros);
        return Integer.valueOf(filasAfectadas.toString());
    }
    /**
     * Obtiene un registro en la tabla GEOAGENCIAS.EEASA_UBI_GEO_AGE.
     */
    public ArrayList<HashMap<String, Object>> select(
        Eeasa_ubi_geo_age objEeasa_ubi_geo_age
    )throws Exception{
        ArrayList<HashMap<String, Object>> listEeasa_ubi_geo_age = new
ArrayList<HashMap<String, Object>>();
        String sql_fun = "GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE.SP_SELECT(?,?)";
        ArrayList<Object> parametros = new ArrayList<Object>();
        parametros.add(objEeasa_ubi_geo_age.getEuga_codigo());
        listEeasa_ubi_geo_age = dbo.SQL_CALL_PROCEDURE(sql_fun,parametros);
        return listEeasa_ubi_geo_age;
    }
    /**
     * Obtiene todos los registros de la tabla GEOAGENCIAS.EEASA_UBI_GEO_AGE.
     */
    public ArrayList<HashMap<String, Object>> selectAll()throws Exception{
        ArrayList<HashMap<String, Object>> listEeasa_ubi_geo_age = new
ArrayList<HashMap<String, Object>>();
        String sql_fun =
"GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE.SP_SELECT_ALL(?)";
        listEeasa_ubi_geo_age = dbo.SQL_CALL_PROCEDURE(sql_fun);
        return listEeasa_ubi_geo_age;
    }
    /**
     * Obtiene un vector dinamico paginado de los objetos equivalentes a los
registros de la tabla GEOAGENCIAS.EEASA_UBI_GEO_AGE.
     */
    public ArrayList<HashMap<String, Object>> selectPaged(
        Integer inOffset,
        Integer inLimit,
        String inOrderBy,
        String inDirection,
        String inFilterText
    )throws Exception{
        ArrayList<HashMap<String, Object>> listEeasa_ubi_geo_age = new
ArrayList<HashMap<String, Object>>();

```

```

        String sql_fun =
"GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE.SP_SELECT_PAGINATED(?, ?, ?, ?, ?, ?, ?)";
        ArrayList<Object> parametros = new ArrayList<Object>();
        parametros.add(inOffset);
        parametros.add(inLimit);
        parametros.add(inOrderBy);
        parametros.add(inDirection);
        parametros.add(inFilterText);
        listEeasa_ubi_geo_age = dbo.SQL_CALL_PROCEDURE(sql_fun,parametros);
        return listEeasa_ubi_geo_age;
    }
}

```

Para que la aplicación funcione correctamente es necesario tener los paquetes creados en la base de datos, estos paquetes son utilizados en el código de la clase antes creada.

Para crear paquetes en Oracle es necesario tener dos partes del paquete, es decir debemos tener la primera parte que es el encabezado en cual consta las funciones y procedimientos utilizados con sus respectivos parámetros, además de la segunda parte que es el cuerpo del paquete, cuyo cuerpo debe constar además de las funciones y procedimientos del encabezado la programación de cada método.

### **Encabezado del paquete GEOAGENCIAS.PKG\_EEASA\_UBI\_GEO\_AGE**

```

CREATE OR REPLACE PACKAGE GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE AUTHID CURRENT_USER AS

TYPE T_CURSOR IS REF CURSOR;

/**
* FUNCION F_INSERT
* INGRESA UN REGISTRO EN LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
* @return VARCHAR CODIGO DEL REGISTRO INGRESADO
*/

FUNCTION F_INSERT(

IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE,

IN_EUGA_NOMBRE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_NOMBRE%TYPE,

```



```

IN_EUGA_PROVINCIA IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_PROVINCIA%TYPE,
IN_EUGA_CANTON IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CANTON%TYPE,
IN_EUGA_DIRECCION IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_DIRECCION%TYPE,
IN_EUGA_HORARIO_ATE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_HORARIO_ATE%TYPE,
IN_EUGA_LONGITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LONGITUD%TYPE,
IN_EUGA_LATITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LATITUD%TYPE,
IN_EUGA_IMAGEN IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_IMAGEN%TYPE
)RETURN VARCHAR2;

/**
* FUNCION F_UPDATE
* ACTUALIZA UN REGISTRO DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
* @return NUMBER NUMERO DE REGISTROS ACTUALIZADOS
*/

FUNCTION F_UPDATE(
IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE,
IN_EUGA_NOMBRE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_NOMBRE%TYPE,
IN_EUGA_PROVINCIA IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_PROVINCIA%TYPE,
IN_EUGA_CANTON IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CANTON%TYPE,
IN_EUGA_DIRECCION IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_DIRECCION%TYPE,
IN_EUGA_HORARIO_ATE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_HORARIO_ATE%TYPE,
IN_EUGA_LONGITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LONGITUD%TYPE,
IN_EUGA_LATITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LATITUD%TYPE,
IN_EUGA_IMAGEN IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_IMAGEN%TYPE
)RETURN NUMBER;

/**
* PROCEDIMIENTO SP_SELECT
* @return OUT_CURSOR REGISTRO DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
*/

PROCEDURE SP_SELECT(

```

```

OUT_CURSOR OUT T_CURSOR,

IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE

);

/**

* PROCEDIMIENTO SP_SELECT_ALL

* @return OUT_CURSOR COLECCION DE REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

*/

PROCEDURE SP_SELECT_ALL(

OUT_CURSOR OUT T_CURSOR

);

/**

* PROCEDIMIENTO SP_SELECT_PAGINATED

* SELECCIONA N REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

* @return OUT_CURSOR COLECCION DE REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

*/

PROCEDURE SP_SELECT_PAGINATED(

OUT_CURSOR OUT T_CURSOR,

IN_START_INDEX IN NUMBER,

IN_LIMIT IN NUMBER,

IN_ORDER_BY IN VARCHAR2,

IN_DIRECTION IN VARCHAR2,

IN_FILTER_TEXT IN CLOB

);

/**

* FUNCION F_DELETE

* ELIMINA UN REGISTRO EN LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

* @return NUMBER NUMERO DE REGISTROS ELIMINADOS

*/

FUNCTION F_DELETE(

```

```

IN_DELETE_TYPE IN VARCHAR,

IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE

)RETURN NUMBER;

END;

/

```

### **Cuerpo del paquete GEOAGENCIAS.PKG\_EEASA\_UBI\_GEO\_AGE**

```

CREATE OR REPLACE PACKAGE BODY GEOAGENCIAS.PKG_EEASA_UBI_GEO_AGE AS

/**

* FUNCION F_INSERT

* INGRESA UN REGISTRO EN LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

* @return VARCHAR CODIGO DEL REGISTRO INGRESADO

*/

FUNCTION F_INSERT(

IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE,

IN_EUGA_NOMBRE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_NOMBRE%TYPE,

IN_EUGA_PROVINCIA IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_PROVINCIA%TYPE,

IN_EUGA_CANTON IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CANTON%TYPE,

IN_EUGA_DIRECCION IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_DIRECCION%TYPE,

IN_EUGA_HORARIO_ATE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_HORARIO_ATE%TYPE,

IN_EUGA_LONGITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LONGITUD%TYPE,

IN_EUGA_LATITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LATITUD%TYPE,

IN_EUGA_IMAGEN IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_IMAGEN%TYPE

)RETURN VARCHAR2 IS

OUT_EUGA_CODIGO VARCHAR2(20);

NUM_ROWS NUMBER;

BEGIN

    IF (IN_EUGA_CODIGO IS NULL) THEN

```

```

SELECT
GEOAGENCIAS.PKG_SEG_SE_UTILS.F_GEN_CODIGO_VARCHAR('EEASA_UBI_GEO_AGE', 'EUGA_CODIGO'
, 'EUGA') INTO OUT_EUGA_CODIGO FROM DUAL;

ELSE

    OUT_EUGA_CODIGO := IN_EUGA_CODIGO;

END IF;

INSERT INTO GEOAGENCIAS.EEASA_UBI_GEO_AGE(

    EUGA_CODIGO,

    EUGA_NOMBRE,

    EUGA_PROVINCIA,

    EUGA_CANTON,

    EUGA_DIRECCION,

    EUGA_HORARIO_ATE,

    EUGA_LONGITUD,

    EUGA_LATITUD,

    EUGA_IMAGEN,

    EUGA_ESTADO,

    EUGA_USUARIO,

    EUGA_FECHA

) VALUES (

    OUT_EUGA_CODIGO,

    IN_EUGA_NOMBRE,

    IN_EUGA_PROVINCIA,

    IN_EUGA_CANTON,

    IN_EUGA_DIRECCION,

    IN_EUGA_HORARIO_ATE,

    IN_EUGA_LONGITUD,

    IN_EUGA_LATITUD,

    IN_EUGA_IMAGEN,

    '1',

```

```

        'Desconocido',

        CURRENT_TIMESTAMP

    );

    NUM_ROWS := SQL%RowCount;

    RETURN OUT_EUGA_CODIGO;

END;

/**
 * FUNCION F_UPDATE
 * ACTUALIZA UN REGISTRO DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
 * @return NUMBER NUMERO DE REGISTROS ACTUALIZADOS
 */

FUNCTION F_UPDATE(

    IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE,

    IN_EUGA_NOMBRE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_NOMBRE%TYPE,

    IN_EUGA_PROVINCIA IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_PROVINCIA%TYPE,

    IN_EUGA_CANTON IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CANTON%TYPE,

    IN_EUGA_DIRECCION IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_DIRECCION%TYPE,

    IN_EUGA_HORARIO_ATE IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_HORARIO_ATE%TYPE,

    IN_EUGA_LONGITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LONGITUD%TYPE,

    IN_EUGA_LATITUD IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_LATITUD%TYPE,

    IN_EUGA_IMAGEN IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_IMAGEN%TYPE

)RETURN NUMBER IS

    NUM_ROWS NUMBER;

BEGIN

    UPDATE GEOAGENCIAS.EEASA_UBI_GEO_AGE SET

        EUGA_NOMBRE = IN_EUGA_NOMBRE,

        EUGA_PROVINCIA = IN_EUGA_PROVINCIA,

        EUGA_CANTON = IN_EUGA_CANTON,

```

```

EUGA_DIRECCION = IN_EUGA_DIRECCION,

EUGA_HORARIO_ATE = IN_EUGA_HORARIO_ATE,

EUGA_LONGITUD = IN_EUGA_LONGITUD,

EUGA_LATITUD = IN_EUGA_LATITUD,

EUGA_IMAGEN = IN_EUGA_IMAGEN,

EUGA_USUARIO = 'Desconocido',

EUGA_FECHA = CURRENT_TIMESTAMP

WHERE

EUGA_CODIGO = IN_EUGA_CODIGO;

NUM_ROWS := SQL%RowCount;

RETURN NUM_ROWS;

END;

/**
* PROCEDIMIENTO SP_SELECT
* SELECCIONA UN REGISTRO DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
* @return OUT_CURSOR REGISTRO DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
*/

PROCEDURE SP_SELECT(
OUT_CURSOR OUT T_CURSOR,
IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE
)IS
BEGIN

OPEN OUT_CURSOR FOR

SELECT

EUGA_CODIGO,

EUGA_NOMBRE,

EUGA_PROVINCIA,

EUGA_CANTON,

EUGA_DIRECCION,

```

```

        EUGA_HORARIO_ATE,

        EUGA_LONGITUD,

        EUGA_LATITUD,

        EUGA_IMAGEN,

        EUGA_ESTADO,

        EUGA_USUARIO,

        EUGA_FECHA

    FROM GEOAGENCIAS.EEASA_UBI_GEO_AGE

    WHERE EUGA_ESTADO = '1' AND

           EUGA_CODIGO = IN_EUGA_CODIGO;

END;

/**

* PROCEDIMIENTO SP_SELECT_ALL

* SELECCIONA TODOS LOS REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

* @return OUT_CURSOR COLECCION DE REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

*/

PROCEDURE SP_SELECT_ALL(

OUT_CURSOR OUT T_CURSOR

)IS

BEGIN

    OPEN OUT_CURSOR FOR

    SELECT

        EUGA_CODIGO,

        EUGA_NOMBRE,

        EUGA_PROVINCIA,

        EUGA_CANTON,

        EUGA_DIRECCION,

        EUGA_HORARIO_ATE,

        EUGA_LONGITUD,

```

```

        EUGA_LATITUD,

        EUGA_IMAGEN,

        EUGA_ESTADO,

        EUGA_USUARIO,

        EUGA_FECHA

    FROM GEOAGENCIAS.EEASA_UBI_GEO_AGE

    WHERE EUGA_ESTADO = '1';

END;

/**

* PROCEDIMIENTO SP_SELECT_PAGINATED

* SELECCIONA N REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

* @return OUT_CURSOR COLECCION DE REGISTROS DE LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE

*/

PROCEDURE SP_SELECT_PAGINATED(

OUT_CURSOR OUT T_CURSOR,

IN_START_INDEX IN NUMBER,

IN_LIMIT IN NUMBER,

IN_ORDER_BY IN VARCHAR2,

IN_DIRECTION IN VARCHAR2,

IN_FILTER_TEXT IN CLOB

)IS

TMP_FILTER_TEXT VARCHAR2(300);

BEGIN

    IF (IN_FILTER_TEXT IS NULL) THEN TMP_FILTER_TEXT := ''; ELSE TMP_FILTER_TEXT :=

IN_FILTER_TEXT; END IF;

    OPEN OUT_CURSOR FOR

        'SELECT

            NUM_REGISTROS,

            EUGA_CODIGO,
```



```
EUGA_NOMBRE,  
EUGA_PROVINCIA,  
EUGA_CANTON,  
EUGA_DIRECCION,  
EUGA_HORARIO_ATE,  
EUGA_LONGITUD,  
EUGA_LATITUD,  
EUGA_IMAGEN,  
EUGA_ESTADO,  
EUGA_USUARIO,  
EUGA_FECHA
```

```
FROM (
```

```
SELECT
```

```
ROWNUM RNUM,
```

```
NUM_REGISTROS,
```

```
EUGA_CODIGO,
```

```
EUGA_NOMBRE,
```

```
EUGA_PROVINCIA,
```

```
EUGA_CANTON,
```

```
EUGA_DIRECCION,
```

```
EUGA_HORARIO_ATE,
```

```
EUGA_LONGITUD,
```

```
EUGA_LATITUD,
```

```
EUGA_IMAGEN,
```

```
EUGA_ESTADO,
```

```
EUGA_USUARIO,
```

```
EUGA_FECHA
```

```
FROM (
```

```
SELECT
```

```

COUNT(*) OVER ( ) NUM_REGISTROS,

EUGA_CODIGO,

EUGA_NOMBRE,

EUGA_PROVINCIA,

EUGA_CANTON,

EUGA_DIRECCION,

EUGA_HORARIO_ATE,

EUGA_LONGITUD,

EUGA_LATITUD,

EUGA_IMAGEN,

EUGA_ESTADO,

EUGA_USUARIO,

EUGA_FECHA

FROM GEOAGENCIAS.EEASA_UBI_GEO_AGE

WHERE EUGA_ESTADO = '1' AND(LENGTH(''||TMP_FILTER_TEXT||') = 0

OR GEOAGENCIAS.PKG_SEG_SE_UTILS.F_STRING_IN(''||TMP_FILTER_TEXT||'),

COALESCE(EUGA_CODIGO, '')||' '||

COALESCE(EUGA_NOMBRE, '')||' '||

COALESCE(EUGA_PROVINCIA, '')||' '||

COALESCE(EUGA_CANTON, '')||' '||

COALESCE(EUGA_DIRECCION, '')||' '||

COALESCE(EUGA_HORARIO_ATE, '')||' '||

COALESCE(EUGA_LONGITUD, '')||' '||

COALESCE(EUGA_LATITUD, '')||' '||

COALESCE(EUGA_IMAGEN, '')||' '||

COALESCE(EUGA_ESTADO, '')||' '||

COALESCE(EUGA_USUARIO, '')||' '||

EUGA_FECHA) = 1)

ORDER BY '||IN_ORDER_BY||' '||IN_DIRECTION||')

```

```

        WHERE ROWNUM <= '||(IN_START_INDEX+IN_LIMIT)||'

    WHERE RNUM > '||IN_START_INDEX;

END;

/**
 * FUNCION F_DELETE
 * ELIMINA UN REGISTRO EN LA TABLA GEOAGENCIAS.EEASA_UBI_GEO_AGE
 * @return NUMBER NUMERO DE REGISTROS ELIMINADOS
 */

FUNCTION F_DELETE(

IN_DELETE_TYPE IN VARCHAR,

IN_EUGA_CODIGO IN GEOAGENCIAS.EEASA_UBI_GEO_AGE.EUGA_CODIGO%TYPE

)RETURN NUMBER IS

NUM_ROWS NUMBER;

BEGIN

    IF(IN_DELETE_TYPE = '1') THEN

        DELETE FROM GEOAGENCIAS.EEASA_UBI_GEO_AGE

        WHERE

            EUGA_CODIGO = IN_EUGA_CODIGO;

    ELSE

        UPDATE GEOAGENCIAS.EEASA_UBI_GEO_AGE SET

            EUGA_ESTADO = '0',

            EUGA_FECHA = CURRENT_TIMESTAMP

        WHERE

            EUGA_CODIGO = IN_EUGA_CODIGO;

    END IF;

    NUM_ROWS := SQL%RowCount;

    RETURN NUM_ROWS;

END;

END;
```

Como se puede observar la clase **WsGAMaestraImpl** implementa métodos del servicio remoto **WsGAMaestra**, dichos métodos deben estar definidos en el servicio remoto antes mencionado.

Para crear el servicio remoto, es necesario crear en la parte del cliente, entonces dirigirse al paquete **ec.com.eeasa.dri.client** y dar clic derecho, luego clic en la opción de **Google Web Toolkit** y finalmente clic en la opción **GWT remote service**, ver Figura 27. En la ventana siguiente especificar el nombre del servicio remoto y dar clic en el botón **Finish**.

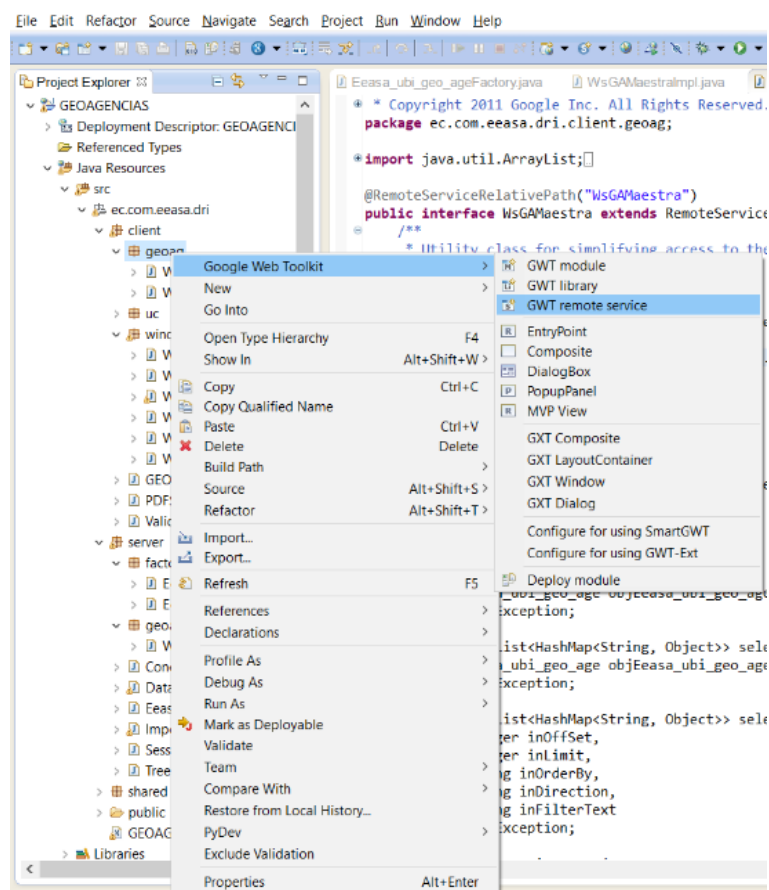


Figura 27. Creación de un servicio remoto.

Elaborado por: El investigador

Al crear el servicio remoto, se creará con un error el cual dice que debemos crear una interfaz asincrónica, para dar solución a este error dar clic derecho donde muestra el error y seleccionar la opción **Create interfaz WsGAMaestraAsync**, tal como muestra la Figura 28.

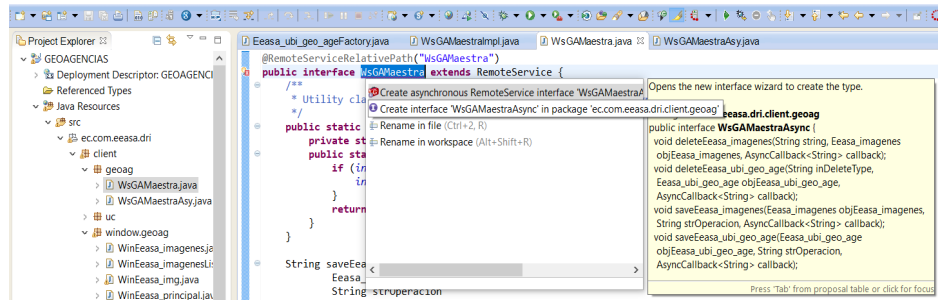


Figura 28. Solución error al crear un servicio remoto.

Elaborado por: El investigador

Lo siguiente será crear las ventanas del proyecto en el lado del cliente, las cuales permiten visualizar y manipular de manera gráfica la base de datos creada.

Para crear las vistas para este proyecto se realizará de manera manual, es decir con código puro sin utilizar el **WindowsBuilder Editor** que trae Eclipse. Para la manipulación de los datos la aplicación posee dos ventanas por tabla, la primera ventana muestra un listado de todos los datos en un gridView y además de botones para editar, eliminar y agregar un nuevo dato. La segunda ventana es la que permite editar o ingresar un nuevo dato ya que contiene los componentes como Textfields, checkboxes, etc... los cuales permiten ingreso de caracteres para almacenar información en la base de datos.

### Codificación de la ventana WinEeasa\_ubi\_geo\_ageList

```
public class WinEeasa_ubi_geo_ageList extends Window implements ListLoadData{
    /**
     * Definicion de los controles de la ventana WinEeasa_ubi_geo_ageList.
     */
    private WinEeasa_ubi_geo_age objWinEeasa_ubi_geo_age;
    private uc_Grid grdEeasa_ubi_geo_age;
```

```

private uc_PagingToolBar<WinEeasa_ubi_geo_ageList> ptbEeasa_ubi_geo_age;
/**
 * Definicion de variables de la ventana WinEeasa_ubi_geo_ageList.
 */
protected Integer TOTAL_RECORDS=0;
protected Integer PAGE_OFFSET=0;
protected Integer PAGE_LIMIT=30;
protected String PAGE_ORDER_BY="EUGA_CODIGO";
protected String PAGE_DIRECTION="ASC";
protected String FILTER_TEXT="";
private Eeasa_ubi_geo_age objEeasa_ubi_geo_age;
/**
 * {Constructor} Definicion del constructor de la clase
WinEeasa_ubi_geo_ageList.
 */
public WinEeasa_ubi_geo_ageList() {
/**
 * Definicion de las propiedades de la ventana
WinEeasa_ubi_geo_ageList.
 */
    setHeading("Listado de las Agencias de la EEASA");
    setActive(false);
    setIconStyle("window");
    setMinWidth(800);
    setMinHeight(500);
    setInitialWidth(800);
    setHeight(500);
    setMaximizable(false);
    setMinimizable(false);
    setClosable(false);
    setLayout(new BorderLayout());
/**
 * Definicion del contenedor del encabezado de la ventana
WinEeasa_ubi_geo_ageList.
 */
    HtmlContainer hcEncabezadoEeasa_ubi_geo_ageList = new
HtmlContainer();
    hcEncabezadoEeasa_ubi_geo_ageList.setStyleName("titleBackground");
    hcEncabezadoEeasa_ubi_geo_ageList.setHtml("El formulario muestra el
listado de las Agencias de la EEASA");
/**
 * Definicion de los botones de accion de la ventana
WinEeasa_ubi_geo_ageList.*

```

```

        final uc_ButtonToolBar btbEeasa_ubi_geo_ageList = new
uc_ButtonToolBar("btnNuevo", "btnEditar", "btnEliminar");
        /**
        * Boton que permite crear un registro de la tabla Eeasa\_ubi\_geo\_age.
        */
        btbEeasa_ubi_geo_ageList.addListener("btnNuevo",Events.Select, new
Listener<ButtonEvent>() {
            @Override
                public void handleEvent(ButtonEvent e) {
                    objWinEeasa_ubi_geo_age = new
WinEeasa_ubi_geo_age(null);
                    objWinEeasa_ubi_geo_age.show();
                    objWinEeasa_ubi_geo_age.addListener(Events.Hide, new
Listener<ComponentEvent>() {
                        public void handleEvent(ComponentEvent be) {
                            loadData();
                        }
                    });
                });
            });
        /**
        * Boton que permite editar un registro de la tabla
Eeasa\_ubi\_geo\_age.
        */
        btbEeasa_ubi_geo_ageList.addListener("btnEditar",Events.Select, new
Listener<ButtonEvent>() {
            @Override
                public void handleEvent(ButtonEvent e) {
                    if
                    (grdEeasa_ubi_geo_age.getSelectionModel().getSelectedItem() == null) {
                        uc_Mensajes.error("Por favor, seleccione el
registro que desea modificar.");
                        return;
                    }else{
                        HashMap<String, Object> rEeasa_ubi_geo_age=
grdEeasa_ubi_geo_age.getSelectedRow();
                        objWinEeasa_ubi_geo_age = new
WinEeasa_ubi_geo_age(rEeasa_ubi_geo_age);
                        objWinEeasa_ubi_geo_age.show();

                        objWinEeasa_ubi_geo_age.addListener(Events.Hide, new
Listener<ComponentEvent>() {

```

```

        public void handleEvent(ComponentEvent
be) {
                                loadData();
                                }
        });
    }
}
});
/**
 * Boton que permite eliminar un registro de la tabla
Eeasa_ubi_geo_age.
 */
    btbEeasa_ubi_geo_ageList.addListener("btnEliminar", Events.Select, new
Listener<ButtonEvent>(){
        @Override
            public void handleEvent(ButtonEvent e) {
                if
(grdEeasa_ubi_geo_age.getSelectionModel().getSelectedItem() == null) {
                    uc_Mensajes.error("Por favor, seleccione el
registro que desea eliminar.");
                    return;
                }else{
                    BaseModel rEeasa_ubi_geo_age=
grdEeasa_ubi_geo_age.getSelectionModel().getSelectedItem();

                    deleteEeasa_ubi_geo_ageList(rEeasa_ubi_geo_age.get("EUGA_CODIGO").toString(
));
                }
            }
        });

/**
 * Definicion del grid que contiene la informacion de la tabla
Eeasa_ubi_geo_age.
 */
    List<ColumnConfig> configs = new ArrayList<ColumnConfig>();
    configs.add(new ColumnConfig("EUGA_CODIGO", "C&oacutedigo", 50));
    configs.add(new ColumnConfig("EUGA_NOMBRE", "Nombre Agencia", 200));
    configs.add(new ColumnConfig("EUGA_PROVINCIA", "Provincia", 100));
    configs.add(new ColumnConfig("EUGA_CANTON", "Cant&oacuten", 100));
    configs.add(new ColumnConfig("EUGA_DIRECCION", "Direcci&oacuten",
300));

```



```

        configs.add(new ColumnConfig("EUGA_HORARIO_ATE", "Horario
Atenci&oacuten", 300));
        configs.add(new ColumnConfig("EUGA_LONGITUD", "Longitud", 100));
        configs.add(new ColumnConfig("EUGA_LATITUD", "Latitud", 100));
        configs.add(new ColumnConfig("EUGA_IMAGEN", "Ubi imagen", 170));
        configs.add(new ColumnConfig("EUGA_ESTADO", "Estado", 50));
        grdEeasa_ubi_geo_age = new uc_Grid(new ColumnModel(configs),false);
        grdEeasa_ubi_geo_age.addListener(Events.CellDoubleClick, new
Listener<BaseEvent>(){
            public void handleEvent(BaseEvent be) {
                btnEeasa_ubi_geo_ageList.FireEvent("btnEditar",
Events.Select);
            }
        });

        grdEeasa_ubi_geo_age.getColumnModel().setHidden(0, true);
        grdEeasa_ubi_geo_age.getColumnModel().setHidden(8, true);
        grdEeasa_ubi_geo_age.getColumnModel().setHidden(9, true);
        /**
         * Definicion de la barra de navegacion para la ventana
WinEeasa_ubi_geo_ageList.
         */
        ptbEeasa_ubi_geo_age = new
uc_PagingToolBar<WinEeasa_ubi_geo_ageList>(this);
        ptbEeasa_ubi_geo_age.setPageSize(PAGE_LIMIT);
        /**
         * Definicion del panel que contiene el los componentes de la ventana
WinEeasa_ubi_geo_ageList.
         */
        FormPanel frmEeasa_ubi_geo_ageList = new FormPanel();
        frmEeasa_ubi_geo_ageList.setBodyBorder(false);
        frmEeasa_ubi_geo_ageList.setPadding(0);
        frmEeasa_ubi_geo_ageList.setHeaderVisible(false);
        frmEeasa_ubi_geo_ageList.setFrame(false);
        frmEeasa_ubi_geo_ageList.setLayout(new FitLayout());
        frmEeasa_ubi_geo_ageList.add(grdEeasa_ubi_geo_age);
        frmEeasa_ubi_geo_ageList.setTopComponent(new
uc_FindToolBar<WinEeasa_ubi_geo_ageList>(this));
        frmEeasa_ubi_geo_ageList.setBottomComponent(ptbEeasa_ubi_geo_age);
        /**
         * Definicion de la barra de botones de la ventana
WinEeasa_ubi_geo_ageList.
         */

```

```

        BorderLayoutData bld_bbEeasa_ubi_geo_ageList = new
BorderLayoutData(LayoutRegion.SOUTH, 32.0f);
        bld_bbEeasa_ubi_geo_ageList.setMargins(new Margins(5, 3, 0, 0));
        /**
         * Aniade los controles a la ventana WinEeasa_ubi_geo_ageList.
         */
        add(hcEncabezadoEeasa_ubi_geo_ageList, new
BorderLayoutData(LayoutRegion.NORTH, 40.0f));
        add(frmEeasa_ubi_geo_ageList, new
BorderLayoutData(LayoutRegion.CENTER));
        add(btbEeasa_ubi_geo_ageList, bld_bbEeasa_ubi_geo_ageList);
        /**
         * Carga la informacion al evento Show la ventana
WinEeasa_ubi_geo_ageList.
         */
        addListener(Events.Show, new WindowListener() {
            public void handleEvent(WindowEvent be) {
                loadData();
            }
        });
    }
    /**
     * Permite cargar los registros de la tabla Eeasa_ubi_geo_age.
     */
    public void loadData(){
        grdEeasa_ubi_geo_age.mask("Espere un momento, por favor.");
        WsGAMaestra.Util.getInstance().selectPagedEeasa_ubi_geo_age(
            getPageOffset(),
            getPageLimit(),
            PAGE_ORDER_BY,
            PAGE_DIRECTION,
            getFilterText(),
            new AsyncCallback<ArrayList<HashMap<String, Object>>>(){
                @Override
                public void onSuccess(ArrayList<HashMap<String, Object>>
result){
                    grdEeasa_ubi_geo_age.unmask();
                    grdEeasa_ubi_geo_age.setDatos(result);

                    setTotalRecords((result.size()>0)?Integer.valueOf(result.get(0).get("isInt_
NUM_REGISTROS")).toString():0);

                    ptbEeasa_ubi_geo_age.resetPaging(getTotalRecords(), getPageOffset());

```

```

        grdEeasa_ubi_geo_age.getSelectionModel().select(0,
false);
    }
    @Override
    public void onFailure(Throwable caught) {
        grdEeasa_ubi_geo_age.unmask();
        uc_Mensajes.error(caught);
    }
});
}
/**
 * Permite eliminar un registro de la tabla Eeasa_ubi_geo_age.
 */
public void deleteEeasa_ubi_geo_ageList(final String inRecordDelete) {
    objEeasa_ubi_geo_age = new Eeasa_ubi_geo_age();
    objEeasa_ubi_geo_age.setEuga_codigo(inRecordDelete);
    uc_Mensajes.confirm("Confirmaci&oacuten", "Desea eliminar el registro
seleccionado",
        new Listener<MessageBoxEvent>(){
            @Override
            public void handleEvent(MessageBoxEvent be){
                if
(Dialog.YES.equals(be.getButtonClicked().getItemId())) {
                    final MessageBox msg = uc_Mensajes.msgEliminar();

WsGAMaestra.Util.getInstance().deleteEeasa_ubi_geo_age("1",objEeasa_ubi_geo
_age,
                    new AsyncCallback<String>(){
                        @Override
                        public void onSuccess(String result){
                            msg.close();
                        }
                    });
                if(grdEeasa_ubi_geo_age.getStore().getCount()==1){
                    setPageOffset((getPageOffset())>0)?getPageOffset()-getPageLimit():0);
                }
                loadData();
            }
            @Override
            public void onFailure(Throwable caught){
                uc_Mensajes.error(caught);
                msg.close();
            }
        }
    }
}

```

```

    });
    }
}
});
}
@Override public void setPageOffset(Integer inPageOffset) {this.PAGE_OFFSET
= inPageOffset;}
@Override public Integer getPageOffset() {return this.PAGE_OFFSET;}
@Override public void setPageLimit(Integer inPageLimit) {this.PAGE_LIMIT =
inPageLimit;}
@Override public Integer getPageLimit() {return this.PAGE_LIMIT;}
@Override public void setFilterText(String inPageFilter) {this.FILTER_TEXT
= inPageFilter;}
@Override public String getFilterText() {return this.FILTER_TEXT;}
@Override public void setTotalRecords(Integer inTotalRecords)
{this.TOTAL_RECORDS = inTotalRecords;}
@Override public Integer getTotalRecords() {return this.TOTAL_RECORDS;}
}

```

Esta sección de código permite al usuario ver el listado de las agencias de la Empresa Eléctrica Ambato. Ver la Figura 29, la cual muestra la ventana finalizada.

### Diseño de la ventana WinEeasa\_ubi\_geo\_ageList

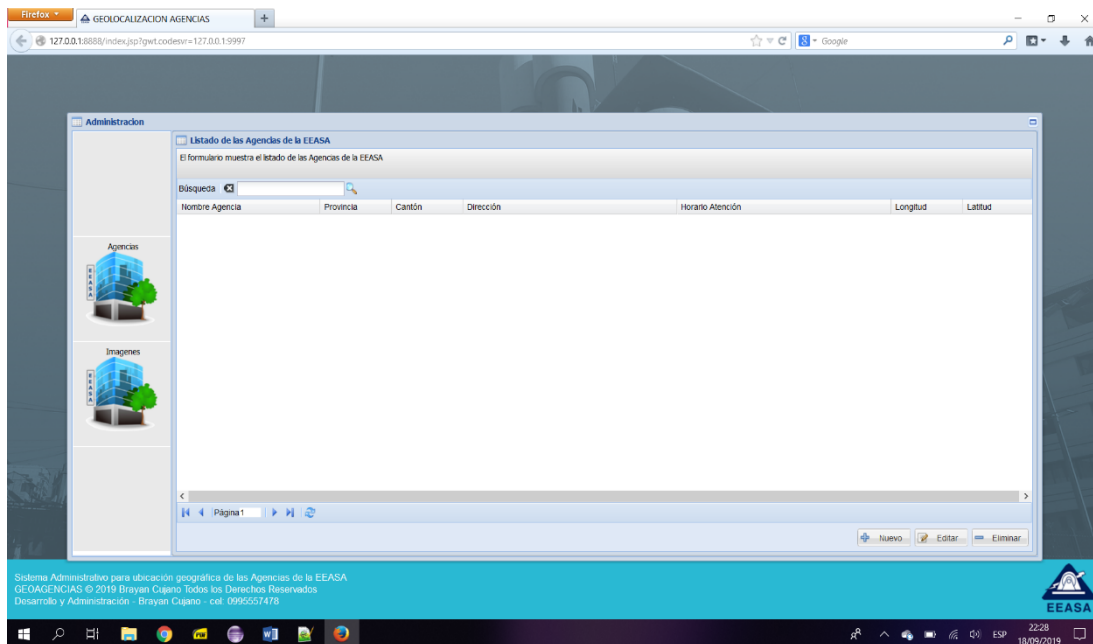


Figura 29. Diseño de la ventana WinEeasa\_ubi\_geo\_ageList.

Elaborado por: El investigador

## Codificación de la ventana WinEeasa\_ubi\_geo\_age

```
public class WinEeasa_ubi_geo_age extends Window {
    /**
     * Definicion de variables de la ventana WinEeasa_ubi_geo_age.
     */
    private HashMap<String, Object> recordEeasa_ubi_geo_age;
    private Eeasa_ubi_geo_age objEeasa_ubi_geo_age;
    private String strOperacion;
    private Image imagenAgencia;

    /**
     * Definicion de los controles de la ventana WinEeasa_ubi_geo_age.
     */
    private uc_TextField txtEuga_codigo;
    private uc_TextField txtEuga_nombre;
    private uc_TextField txtEuga_provincia;
    private uc_TextField txtEuga_canton;
    private uc_TextField txtEuga_direccion;
    private uc_TextField txtEuga_horario_ate;
    private uc_TextField txtEuga_longitud;
    private uc_TextField txtEuga_latitud;
    private uc_TextField txtEuga_imagen;
    /**
     * {Constructor} Definicion del constructor de la clase WinEeasa_ubi_geo_age.
     */
    public WinEeasa_ubi_geo_age(HashMap<String, Object>
inRecordEeasa_ubi_geo_age) {
        /**
         * Asignacion de la fila seleccionada al objeto
recordEeasa_ubi_geo_age.
         */
        recordEeasa_ubi_geo_age = inRecordEeasa_ubi_geo_age;

        /**
         * Definicion de las propiedades de la ventana WinEeasa_ubi_geo_age.
         */
        setHeading("Agencias EEASA");
        setIconStyle("window");
        setModal(true);
        setMinWidth(775);
        setMinHeight(375);
        setInitialWidth(775);
    }
}
```

```

        setHeight(375);
        setActive(false);
        setLayout(new BorderLayout());
        /**
         * Definicion del contenedor del encabezado de la ventana
WinEeasa_ubi_geo_age.
        */
        HtmlContainer hcEncabezadoEeasa_ubi_geo_age = new HtmlContainer();
        hcEncabezadoEeasa_ubi_geo_age.setStyleName("titleBackground");
        hcEncabezadoEeasa_ubi_geo_age.setHtml("El formulario permite editar
de las Agencias de la EEASA");
        /**
         * Definicion de los botones de accion de la ventana
WinEeasa_ubi_geo_age.
        */
        final uc_ButtonToolBar btbEeasa_ubi_geo_age = new
uc_ButtonToolBar("btnGuardar","btnSalir");
        /**
         * {uc_TextField} Campo de texto que contiene .
        */
        txtEuga_codigo = new uc_TextField();
        txtEuga_codigo.setFieldLabel("Euga_codigo");
        txtEuga_codigo.setMaxLength(20);
        /**
         * {uc_TextField} Campo de texto que contiene .
        */
        txtEuga_nombre = new uc_TextField(1);
        txtEuga_nombre.setFieldLabel("Nombre Agencia");
        txtEuga_nombre.setMaxLength(100);
        txtEuga_nombre.setSoloMayus(true);
        /**
         * {uc_TextField} Campo de texto que contiene .
        */
        txtEuga_provincia = new uc_TextField();
        txtEuga_provincia.setFieldLabel("Provincia");
        txtEuga_provincia.setMaxLength(30);
        txtEuga_provincia.setSoloMayus(true);
        /**
         * {uc_TextField} Campo de texto que contiene .
        */
        txtEuga_canton = new uc_TextField();
        txtEuga_canton.setFieldLabel("Cant&ocuten");
        txtEuga_canton.setMaxLength(30);

```

```

txtEuga_canton.setSoloMayus(true);
/**
 * {uc_TextField} Campo de texto que contiene .
 */
txtEuga_direccion = new uc_TextField(1);
txtEuga_direccion.setFieldLabel("Direcci&oacuten");
txtEuga_direccion.setMaxLength(200);
txtEuga_direccion.setSoloMayus(true);
/**
 * {uc_TextField} Campo de texto que contiene .
 */
txtEuga_horario_ate = new uc_TextField(1);
txtEuga_horario_ate.setFieldLabel("Horario Atenci&oacuten");
txtEuga_horario_ate.setMaxLength(100);
txtEuga_horario_ate.setSoloMayus(true);
/**
 * {uc_TextField} Campo de texto que contiene .
 */
txtEuga_longitud = new uc_TextField(1);
txtEuga_longitud.setFieldLabel("Longitud");
txtEuga_longitud.setMaxLength(20);
/**
 * {uc_TextField} Campo de texto que contiene .
 */
txtEuga_latitud = new uc_TextField(1);
txtEuga_latitud.setFieldLabel("Latitud");
txtEuga_latitud.setMaxLength(20);
/**
 * {uc_TextField} Campo de texto que contiene .
 */
txtEuga_imagen = new uc_TextField(1);
txtEuga_imagen.setFieldLabel("Ubi imagen");
txtEuga_imagen.setMaxLength(200);
txtEuga_imagen.setReadOnly(false);

imagenAgencia = new Image();
imagenAgencia.setUrl("imagenes/agencias/agencias.png");
imagenAgencia.setSize("100px", "180px");
/**
 * Definicion del contenedor de los componentes de la ventana
WinEeasa_ubi_geo_age.
 */
LayoutContainer lcEncabezadoCol0 = new LayoutContainer();

```

```

lcEncabezadoCol0.setId("lcEncabezadoCol0");
lcEncabezadoCol0.setLayout(new FormLayout());
lcEncabezadoCol0.add(txtEuga_nombre, new FormData("90%"));
lcEncabezadoCol0.add(txtEuga_provincia, new FormData("90%"));
lcEncabezadoCol0.add(txtEuga_canton, new FormData("90%"));
lcEncabezadoCol0.add(txtEuga_direccion, new FormData("90%"));

LayoutContainer lcEncabezadoCol1 = new LayoutContainer();
lcEncabezadoCol1.setId("lcEncabezadoCol1");
lcEncabezadoCol1.setLayout(new FormLayout());
lcEncabezadoCol1.add(txtEuga_horario_ate, new FormData("90%"));

lcEncabezadoCol1.add(txtEuga_longitud, new FormData("90%"));
lcEncabezadoCol1.add(txtEuga_latitud, new FormData("90%"));
lcEncabezadoCol1.add(txtEuga_imagen, new FormData("90%"));

LayoutContainer lcEncabezado1 = new LayoutContainer();
lcEncabezado1.setId("lcEncabezado1");
lcEncabezado1.setStyleAttribute("padding", "0px 0px 10px 0px");
lcEncabezado1.setLayout(new ColumnLayout());
lcEncabezado1.add(lcEncabezadoCol0, new ColumnData(0.5));
lcEncabezado1.add(lcEncabezadoCol1, new ColumnData(0.5));

final ContentPanel cntFotoPrincipal = new ContentPanel();
cntFotoPrincipal.setBodyBorder(false);
cntFotoPrincipal.setHeaderVisible(false);
cntFotoPrincipal.setIconStyle("window");
cntFotoPrincipal.setFrame(true);
cntFotoPrincipal.setLayout(new FormLayout());
cntFotoPrincipal.add(imagenAgencia, new FormData("100%"));

LayoutContainer lcEncabezado = new LayoutContainer();
lcEncabezado.setId("lcEncabezado");
lcEncabezado.setLayout(new ColumnLayout());
lcEncabezado.add(lcEncabezado1, new ColumnData(0.70));
lcEncabezado.add(cntFotoPrincipal, new ColumnData(0.30));

final ContentPanel cntEncabezado = new ContentPanel();
cntEncabezado.setBodyBorder(false);
cntEncabezado.setHeaderVisible(false);
cntEncabezado.setIconStyle("window");
cntEncabezado.setFrame(true);

```



```

cntEncabezado.setLayout(new FormLayout());
cntEncabezado.add(lcEncabezado, new FormData("100%"));

final FormPanel frmEeasa_ubi_geo_age = new FormPanel();
frmEeasa_ubi_geo_age.setBodyBorder(false);
frmEeasa_ubi_geo_age.setPadding(0);
frmEeasa_ubi_geo_age.setHeaderVisible(false);
frmEeasa_ubi_geo_age.setIconStyle("window");
frmEeasa_ubi_geo_age setFrame(true);
frmEeasa_ubi_geo_age.setLayout(new FormLayout());
frmEeasa_ubi_geo_age.add(cntEncabezado, new FormData("100%"));
/**
 * Boton que permite guardar la informacion de la tabla
Eeasa_ubi_geo_age.
 */
btbEeasa_ubi_geo_age.addListener("btnGuardar",Events.Select, new
Listener<ButtonEvent>() {
    @Override
    public void handleEvent(ButtonEvent e) {
        if(frmEeasa_ubi_geo_age.isValid())
            saveEeasa_ubi_geo_age();
    }
});
/**
 * Boton que permite salir de la ventana WinEeasa_ubi_geo_age.
 */
btbEeasa_ubi_geo_age.addListener("btnSalir",Events.Select, new
Listener<ButtonEvent>(){
    public void handleEvent(ButtonEvent e) {
        hide();
    }
});
/**
 * Definicion de la barra de botones de la ventana
WinEeasa_ubi_geo_age.
 */
BorderLayoutData bld_bbEeasa_ubi_geo_age = new
BorderLayoutData(LayoutRegion.SOUTH, 32.0f);
bld_bbEeasa_ubi_geo_age.setMargins(new Margins(5, 3, 0, 0));
/**
 * Ania de los controles a la ventana WinEeasa_ubi_geo_age.
 */

```

```

        add(hcEncabezadoEeasa_ubi_geo_age, new
BorderLayoutData(LayoutRegion.NORTH, 40.0f));
        add(frmEeasa_ubi_geo_age, new
BorderLayoutData(LayoutRegion.CENTER));
        add(btbtEeasa_ubi_geo_age,bld_bbEeasa_ubi_geo_age);
        loadEeasa_ubi_geo_age();
    }
    /**
     * Permite cargar los controles del registro selccionado de la tabla
Eeasa_ubi_geo_age.
     */
    public void loadEeasa_ubi_geo_age(){
        strOperacion = "insert";
        if(recordEeasa_ubi_geo_age!=null){
            strOperacion = "update";
            if(recordEeasa_ubi_geo_age.get("EUGA_CODIGO")!=
null)txtEuga_codigo.setValue(recordEeasa_ubi_geo_age.get("EUGA_CODIGO").toString())
;
                if(recordEeasa_ubi_geo_age.get("EUGA_NOMBRE")!=
null)txtEuga_nombre.setValue(recordEeasa_ubi_geo_age.get("EUGA_NOMBRE").toString())
;

            if(recordEeasa_ubi_geo_age.get("EUGA_PROVINCIA")!=null)txtEuga_provincia.se
tValue(recordEeasa_ubi_geo_age.get("EUGA_PROVINCIA").toString());
                if(recordEeasa_ubi_geo_age.get("EUGA_CANTON")!=
null)txtEuga_canton.setValue(recordEeasa_ubi_geo_age.get("EUGA_CANTON").toString())
;

            if(recordEeasa_ubi_geo_age.get("EUGA_DIRECCION")!=null)txtEuga_direccion.se
tValue(recordEeasa_ubi_geo_age.get("EUGA_DIRECCION").toString());
                if(recordEeasa_ubi_geo_age.get("EUGA_HORARIO_ATE")!=
null)txtEuga_horario_ate.setValue(recordEeasa_ubi_geo_age.get("EUGA_HORARIO_ATE").t
oString());

            if(recordEeasa_ubi_geo_age.get("EUGA_LONGITUD")!=null)txtEuga_longitud.setV
alue(recordEeasa_ubi_geo_age.get("EUGA_LONGITUD").toString());
                if(recordEeasa_ubi_geo_age.get("EUGA_LATITUD")!=
null)txtEuga_latitud.setValue(recordEeasa_ubi_geo_age.get("EUGA_LATITUD").toString(
));

            if(recordEeasa_ubi_geo_age.get("EUGA_IMAGEN")!=null)txtEuga_imagen.setValue
(recordEeasa_ubi_geo_age.get("EUGA_IMAGEN").toString());

```

```

        if(recordEeasa_ubi_geo_age.get("EUGA_IMAGEN")!=null)imagenAgencia.setUrl(re
cordEeasa_ubi_geo_age.get("EUGA_IMAGEN").toString());
    }
}
/**
 * Permite almacenar la informacion de la tabla Eeasa_ubi_geo_age.
 */
public void saveEeasa_ubi_geo_age(){
    objEeasa_ubi_geo_age = new Eeasa_ubi_geo_age(
        txtEuga_codigo.getValue(),
        txtEuga_nombre.getValue(),
        txtEuga_provincia.getValue(),
        txtEuga_canton.getValue(),
        txtEuga_direccion.getValue(),
        txtEuga_horario_ate.getValue(),
        txtEuga_longitud.getValue(),
        txtEuga_latitud.getValue(),
        txtEuga_imagen.getValue());

    objEeasa_ubi_geo_age.setEeasa_ubi_geo_age((recordEeasa_ubi_geo_age!=null)?r
ecordEeasa_ubi_geo_age.toString(): "");

    WSGAMaestra.Util.getInstance().saveEeasa_ubi_geo_age(
        objEeasa_ubi_geo_age,
        strOperacion,
        new AsyncCallback<String>(){
            @Override
            public void onSuccess(String result) {
                uc_Mensajes.info("Informacion", "El registro se ha
guardado correctamente.");
                hide();
            }
            @Override
            public void onFailure(Throwable caught) {
                uc_Mensajes.error(caught);
            }
        });
}
}
}

```

La sección de código mostrada en la sección anterior permite manipular los datos de las tablas creadas en la historia de usuario 1, es decir se puede ingresar, modificar un campo, ver la Figura 30, para ver el diseño final de la ventana.

### Diseño de la ventana WinEeasa\_ubi\_geo\_age

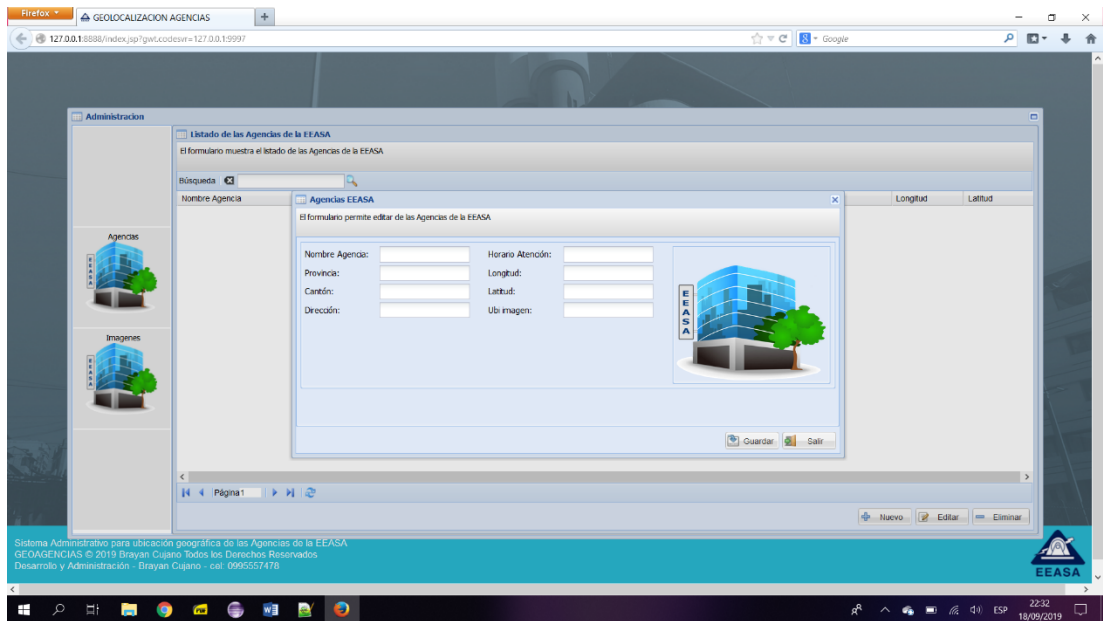


Figura 30. Diseño de la ventana WinEeasa\_ubi\_geo\_age.

Elaborado por: El investigador

### Id 3. Desarrollo de servicios web para geolocalización de las agencias y pantalla de inicio

Para desarrollar los servicios web para geolocalización de las agencias y pantalla de inicio, se documentará la programación de las clases necesarias para manipular la tabla GEOAGENCIAS.EEASA\_UBI\_GEO\_AGE, ya que la lógica de programación de la clase GEOAGENCIAS.EEASA\_IMAGENES es la misma.

Para crear servicios web el IDE Eclipse ofrece diferentes formas de realizar, estas son:

- Crear un proyecto Web Dinámico.
- Crear un proyecto Web Maven.
- Crear directamente un proyecto RESTful Web Services.

En esta investigación se creará una fusión de las dos primeras opciones, es decir crear un proyecto web dinámico y después transformarlo a tipo Maven.

Para esto dar clic en **File**, posar el mouse en la opción **New** y escoger **Dynamic Web Project**, en el caso que esta opción no aparezca, buscar la opción **Other** y buscar el proyecto Web Dinámico y dar clic sobre él. Ver Figura 31.

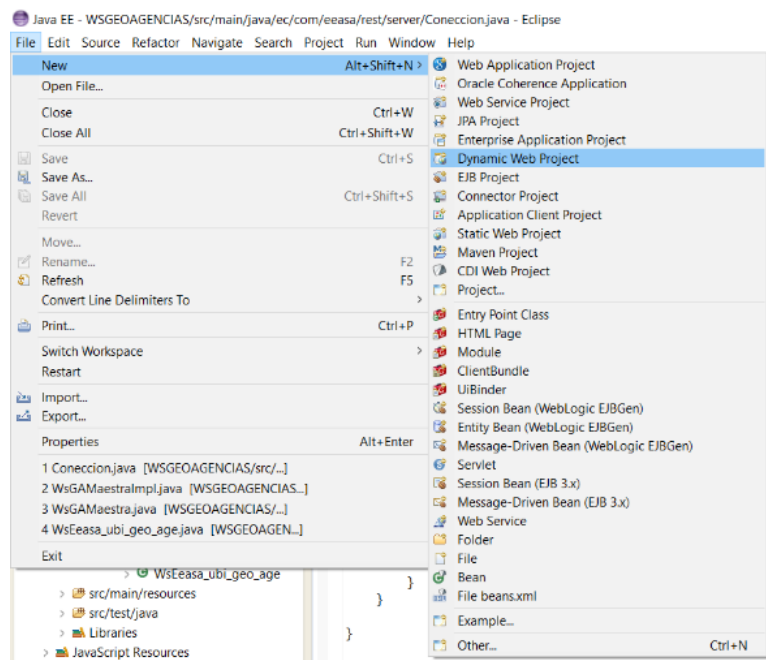
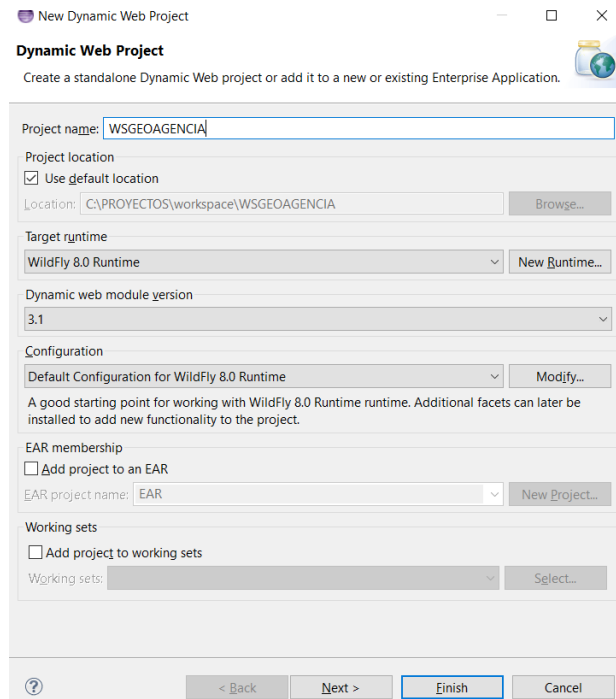


Figura 31. Crear un proyecto web dinámico.

Elaborado por: El investigador

En la ventana siguiente especificar la información relacionada con la creación de proyecto como es: nombre del proyecto, localización del proyecto, implementación para un servidor de aplicaciones. Una vez facilitado esta información dar clic en el botón **Finish** y esperar a que se cree el proyecto.

La Figura 32 muestra los parámetros iniciales para crear un proyecto web dinámico.



*Figura 32. Información para crear un proyecto web dinámico.*

*Elaborado por: El investigador*

Para convertir al proyecto web dinámico creado anteriormente, dar clic derecho en la carpeta inicial del proyecto, posar el mouse en la opción **Configure** y finalmente dar clic en la opción **Convert Maven Project**.

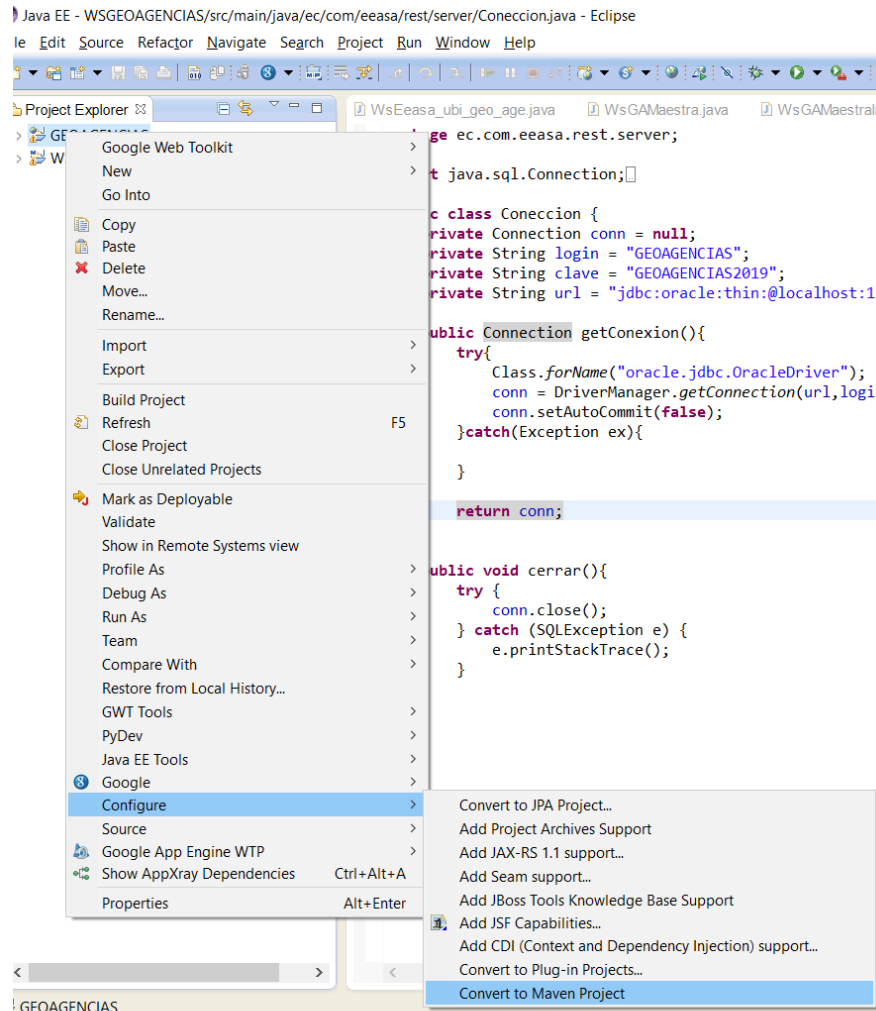


Figura 33. Convertir en proyecto Maven.

Elaborado por: El investigador

Al convertir a proyecto Maven se creará automáticamente un archivo **pom.xml** en el cual se especifica las dependencias que se va a utilizar para el desarrollo de servicios web. A continuación, se presenta la configuración del archivo **pom.xml**.

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

```

```

<groupId>WSGEOAGENCIAS</groupId>
<artifactId>WSGEOAGENCIAS</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>com.sun.jersey</groupId>
    <artifactId>jersey-core</artifactId>
    <version>1.19</version>
  </dependency>
  <dependency>
    <groupId>com.sun.jersey</groupId>
    <artifactId>jersey-json</artifactId>
    <version>1.19</version>
  </dependency>
  <dependency>
    <groupId>com.sun.jersey</groupId>
    <artifactId>jersey-servlet</artifactId>
    <version>1.19</version>
  </dependency>
</dependencies>
<build>
<finalName>WSGEOAGENCIAS</finalName>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.3</version>
      <configuration>
        <warSourceDirectory>WebContent</warSourceDirectory>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.1</version>

```



```

        <configuration>
            <source>1.7</source>
            <target>1.7</target>
        </configuration>
    </plugin>
</plugins>
</build>
</project>

```

Se necesita una clase nueva para almacenar los métodos de los servicios web, la diferencia de la aplicación web administrativa y el proyecto donde se desarrolla los servicios web, es que esta última mencionada no posee ventanas de manipulación de datos, en lugar de estas ventanas se desarrolla los métodos web, para este caso se utilizará el método GET.

### Programación de la Clase WsEeasa\_ubi\_geo\_age

```

@Path("/eeasa")
public class WsEeasa_ubi_geo_age extends Application {

    @GET
    @Path("/obtenerAgencias")
    @Consumes({MediaType.APPLICATION_JSON})
    @Produces({MediaType.APPLICATION_JSON})
    public ArrayList<HashMap<String, Object>> obtenerAgencias() throws
Exception{
        WSGAMaestraImpl maestraImp = new WSGAMaestraImpl();
        ArrayList<HashMap<String, Object>> lista =
maestraImp.selectAllEeasa_ubi_geo_age();
        return lista;
    }

    @GET
    @Path("/obtenerImagenes")
    @Consumes({MediaType.APPLICATION_JSON})
    @Produces({MediaType.APPLICATION_JSON})
    public ArrayList<HashMap<String, Object>> obtenerImagenes() throws
Exception{
        WSGAMaestraImpl maestraImpl = new WSGAMaestraImpl();

```

```

        ArrayList<HashMap<String, Object>> lista =
maestraImpl.selectAllEeasa_imagenes();
        return lista;
    }
}

```

La clase **WsEeasa\_ubi\_geo\_age** es la contenedora de los servicios web, Donde:

**@Path**: identifica la ruta en la cual estará disponible el servicio web. En este caso la URI del servicio web se forma de la siguiente manera, la URI base del servidor en el que esta implementado el servicio web, el Path padre que es aquel que tenemos configurado en el archivo **web.xml**, y los **@Path** que se especifican en la clase **WsEeasa\_ubi\_geo\_age**.

**@GET**: esta anotación se usa solo en el cuando se necesite obtener información de la base de datos.

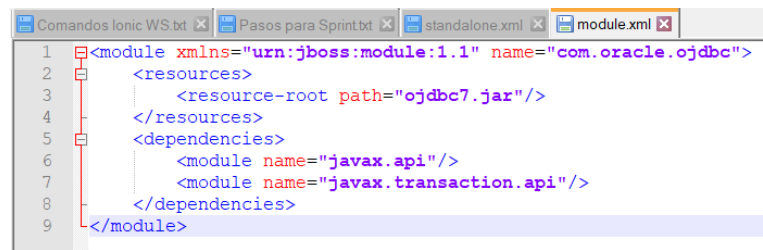
**@Consumes**({MediaType.APPLICATION\_JSON}): esta anotación indica que a la información devuelta por el servicio web se debe acceder en el formato especificado, es este caso JSON.

**@Produces**({MediaType.APPLICATION\_JSON}): esta anotación indica que la información devuelta por el servicio web es en formato JSON.

### **Publicación del servicio web en el servidor de aplicaciones WildFly**

1. Como primer paso será configurar el driver de base de datos en el servidor. Descargar el servidor de aplicaciones WildFly de la siguiente dirección <https://wildfly.org/downloads/>.
2. Luego descomprimir y copiar la carpeta raíz de preferencia al disco C.

3. Crear una carpeta llamada oracle en la siguiente dirección [WildFly\_HOME]\modules\system\layers\base\com\ y dentro de la carpeta oracle crear otra carpeta con el nombre de main.
4. Descargar el ojdbc7 y pegar dentro de la carpeta main. Link de descarga del ojdbc: <https://www.oracle.com/database/technologies/jdbc-drivers-12c-downloads.html>
5. Crear un archivo de configuración con el nombre module.xml dentro de la carpeta mencionada en el paso 4. El archivo module.xml debe tener las líneas de código que muestra la Figura 34.



```

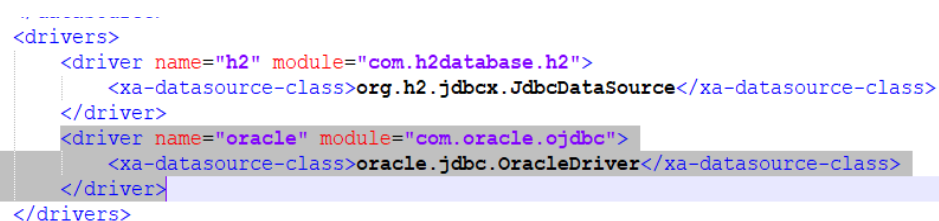
1 <module xmlns="urn:jboss:module:1.1" name="com.oracle.ojdbc">
2   <resources>
3     <resource-root path="ojdbc7.jar"/>
4   </resources>
5   <dependencies>
6     <module name="javax.api"/>
7     <module name="javax.transaction.api"/>
8   </dependencies>
9 </module>

```

Figura 34. Código del archivo module.xml

Elaborado por: El investigador

6. En el archivo de configuración standalone.xml ubicada en la siguiente dirección [WildFly\_HOME]\modules\standalone\configuration. Dentro de la etiqueta <drivers> escribir el código subrayado de la Figura 35.



```

<drivers>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
  </driver>
  <driver name="oracle" module="com.oracle.ojdbc">
    <xa-datasource-class>oracle.jdbc.OracleDriver</xa-datasource-class>
  </driver>
</drivers>

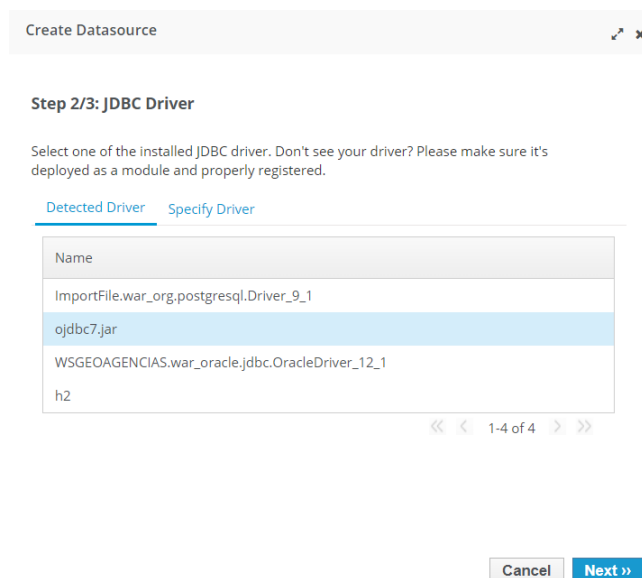
```

Figura 35. Código del archivo standalone.xml

Elaborado por: El investigador

7. Luego de realizar la configuración del driver para la base de datos Oracle, ejecutar el servidor WildFly, para eso dar doble clic en el archivo standalone.bat ubicado en [WildFly\_HOME]\bin.
8. En un navegador web poner la dirección URI de la consola de administración del servidor WildFly <http://localhost:9992/console/App.html> iniciar sesión con la cuenta de administrador.

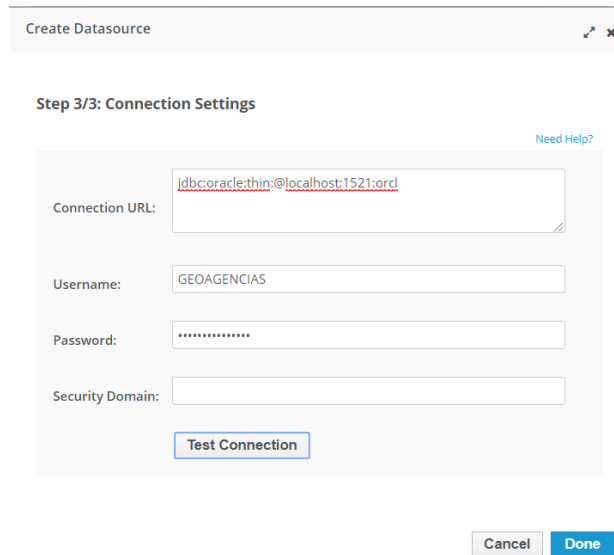
9. Dentro de la consola de administrador de WildFly dar clic en el menú **Configuration** y en el submenú **Connector** seleccionar **Datasources**.
10. Dentro de la ventana de configuración de un nuevo datasource dar clic en el botón **Add** y especificar el nombre de la conexión y el JNDI el cual debe empezar con la palabra java: seguir el ejemplo del ExampleDS que el servidor WildFly da como guía. Y dar clic en el botón **Next**.
11. En la siguiente ventana que muestra la Figura 36 escoger el driver que se configuró en el paso 3 al 6 y dar clic en el botón **Next**, en este caso el driver se llama ojdbc7.jar



*Figura 36. Escoger el driver configurado.*

*Elaborado por: El investigador*

12. En la ventana siguiente llenar los parámetros de conexión, mismos que se muestran en la Figura 37. Configurar parámetros de conexión.



The screenshot shows a dialog box titled "Create Datasource" with a close button (x) in the top right corner. Below the title bar, it says "Step 3/3: Connection Settings" and a "Need Help?" link. The main area contains four input fields: "Connection URL:" with the value "jdbc:oracle:thin:@localhost:1521:orcl", "Username:" with the value "GEOAGENCIAS", "Password:" with a masked password "\*\*\*\*\*", and "Security Domain:" which is empty. A "Test Connection" button is located below the fields. At the bottom of the dialog, there are "Cancel" and "Done" buttons.

Figura 37. Configurar parámetros de conexión.

*Elaborado por: El investigador*

13. Para finalizar la configuración del datasource dar clic en el botón **Test Connection**, y si le configuraron de manera correcta saldrá un mensaje indicando que la conexión fue satisfactoria, caso contrario repetir estos 13 pasos.
14. Para correr el servicio web, abrir Eclipse y dar clic derecho sobre el proyecto Maven, posar el mouse sobre la opción **Run As** y luego dar clic en la opción **Run on Server**.

15. Después escoger el servidor que muestra la Figura 38 y dar clic en el botón **Finish**.

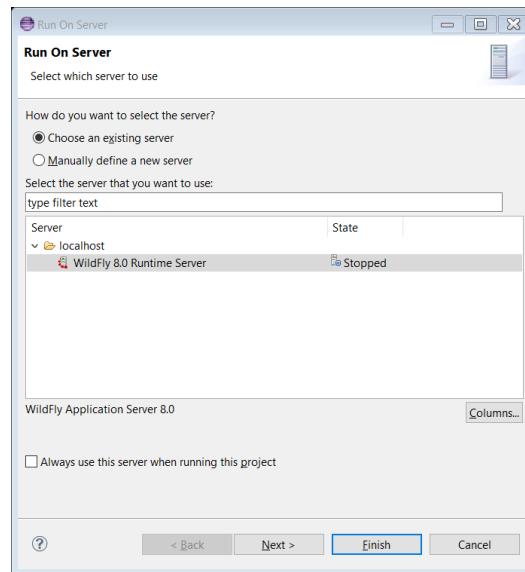


Figura 38. Correr servicio web en WildFly.

Elaborado por: El investigador

16. Para visualizar el servicio web usaremos SoapUI, esta es una aplicación cliente para servicios web. La Figura 39 muestra el resultado en formato Json que devuelve al realizar la petición del servicio web mediante el método @GET.

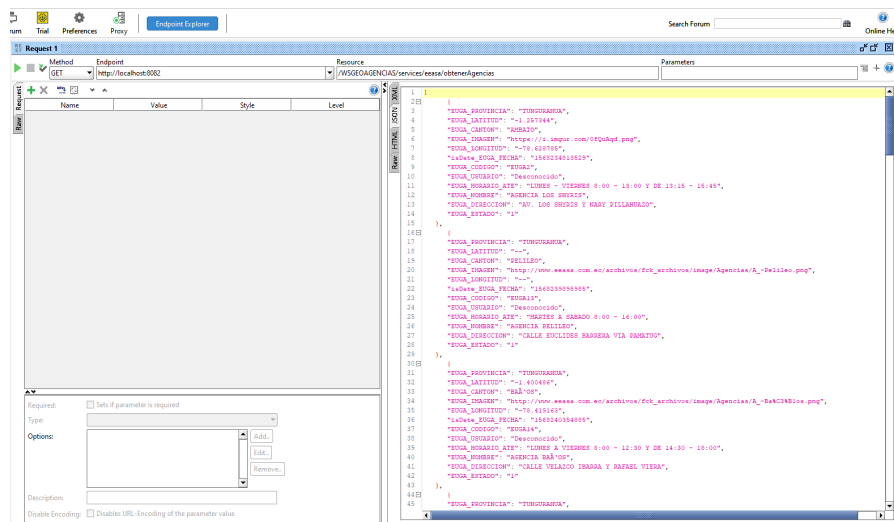


Figura 39. Consumir servicio web en SoapUI.

Elaborado por: El investigador

#### 1.4.10. Desarrollo del Sprint 2

Tabla 29. Desarrollo del Sprint 2.

Nombre Sprint	Id	Título
Sprint 2	1	Desarrollo de servicios web para consulta de planillas eléctricas
	2	Edición de imágenes
	3	Preparación del entorno de desarrollo y creación de la aplicación móvil.

Elaborado por: El investigador

#### Desarrollo del Sprint 2 mostrado en la Tabla 29.

##### Id 1. Desarrollo de servicios web para consulta de planillas eléctricas.

Para desarrollar el servicio web, se toma a consideración el procedimiento del **Id 3** del Sprint número 1.

A continuación, la Tabla 30 muestra las URIs asignadas para los servicios web creados.

Tabla 30. URIs utilizadas para intercambio de información

Método	URIs
GET	<a href="http://xxx.xxx.xxx.xxx/appeasa/rest/eeasa/obtenerAgencias">http://xxx.xxx.xxx.xxx/appeasa/rest/eeasa/obtenerAgencias</a>
	<a href="http://xxx.xxx.xxx.xxx:xxxx/appeasa/rest/eeasa/obtenerImagenes">http://xxx.xxx.xxx.xxx:xxxx/appeasa/rest/eeasa/obtenerImagenes</a>
	<a href="http://xxx.xxx.xxx.xxx/WSConsultaMobil/consultaeasa/consulta/?cuenta=xxxxxx">http://xxx.xxx.xxx.xxx/WSConsultaMobil/consultaeasa/consulta/?cuenta=xxxxxx</a>
	<a href="http://xxx.xxx.xxx.xxx/WSConsultaMobil/consultaeasa/FacturasPendientes/?cuenta=xxxxxx">http://xxx.xxx.xxx.xxx/WSConsultaMobil/consultaeasa/FacturasPendientes/?cuenta=xxxxxx</a>
	<a href="http://xxx.xxx.xxx.xxx:xxxx/WSConsultaMobil/consultaeasa/FacturasConsumo/?cuenta=xxxxxx">http://xxx.xxx.xxx.xxx:xxxx/WSConsultaMobil/consultaeasa/FacturasConsumo/?cuenta=xxxxxx</a>

Elaborado por: El investigador

Las URIs que se describen en la Tabla 4 son aquellas que darán funcionalidad a los servicios web, porque mediante estos podemos acceder a los métodos que la aplicación móvil requiere para su correcto funcionamiento, a continuación, se explicará la funcionalidad que tiene cada URI.

**Obtener Agencias:** devuelve un array tipo JSON con la información de las agencias de la Empresa Eléctrica Ambato. Esta información es: código, nombre, provincia, cantón, dirección, horario de atención, longitud, latitud, url imagen.

**Obtener Imágenes:** devuelve un array tipo JSON con las direcciones de las imágenes que se visualizarán en la página principal de la aplicación móvil. Los datos devueltos son los siguientes: código, url, descripción, principal.

**Consulta:** este servicio web devuelve los datos del cliente de la EEASA, el parámetro que se necesita es el número de cuenta. Los datos devueltos son: agencia, cedula, codigoResultado, cuenta, dirección, mensajeResultado, nombreTitular, numFacturas, numeroMedidor, tarifa, valorDeuda.

**Facturas Pendientes:** este servicio web devuelve los datos de las facturas que están pendientes de cancelar por el consumo de energía en la EEASA el parámetro que se necesita es el número de cuenta. Los datos devueltos son: autorización, consumo, fechaAutorizacion, fechaEmision, identificador, lecturaActural, lecturaAnterior, mesConsumo, numeroFac, totalDeuda.

**Facturas Consumo:** este servicio web devuelve los datos de los últimos doce meses de consumo de energía en la EEASA, para poder mostrar un reporte gráfico de la cantidad de consumo que se está realizando cada mes, el parámetro que se necesita es el número de cuenta. Los datos devueltos son: autorización, consumo,



fechaAutorizacion, fechaEmision, identificador, lecturaActural, lecturaAnterior, mesConsumo, numeroFac, totalDeuda.

## **Id 2. Edición de imágenes.**

Software Requerido

- Photoshop versión CC 2019.
- Cámara Fotográfica.

La edición de imágenes se realiza en Photoshop, cuyo procedimiento depende del diseño de la imagen que se necesita realizar.

## **Id 3. Preparación del entorno de desarrollo y creación de la aplicación móvil.**

El framework de desarrollo que se utilizará es Ionic, este framework trabaja con AngularJs, Apache Cordova, Node.js, motivo por la cual se instalara primero dichos componentes para finalizar con la instalación de Ionic. A continuación, se mostrará los pasos de instalación.

1. Instalar Node.js, primero descargamos de la página web <https://nodejs.org/es/>, para esta investigación se utilizo la versión 8.12.0, una vez descargado dar doble clic sobre el instalador y seguir las instrucciones de instalación.
2. Node.js integra el gestor de paquetes npm, el cual será útil para la instalación de los siguientes componentes. Ahora instalar Apache cordova, para esto abrir la consola de comandos y escribir **npm install -g cordova**, esperar que se descarge todos los paquetes, dependencias necesarias y listo.
3. Lo siguiente a instalar es un editor de código fuente, para este proyecto se utilizará Visual Studio Code, su instalador esta disponible en el siguiente enlace: <https://code.visualstudio.com/Download>.
4. Descargar e instalar Visual Studio Code, además para continuar la instalación siguiente se recomienda reiniciar su computadora.

5. Una vez reiniciada la computadora, instalar Ionic con el comando **npm install -g ionic@3.20.0**, y el framework de desarrollo esta listo, en el siguiente paso se va a instalar y configurar las librerías y componentes para poder ejecutar y construir aplicaciones móviles híbridas con Ionic.
6. Instalar el jdk de java la versión 8u201, cuyo instalador está disponible en: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>, una vez descargado instalar.
7. Instalar Andorid Studio o a su vez se puede usar los sdk de Android Studio, para el desarrollo de esta investigación se instalará Android Studio, se puede descargar de la siguiente página <https://developer.android.com/studio/index.html?hl=es-419>, cuando finalice la instalación, descargar la api 26 de Android y crear un emulador para poder ejecutar las aplicaciones creadas en Ionic.
8. Ahora descargar, descomprimir apache-ant 1.10.5 y copiar de preferencia al disco C, link de descarga de apache-ant [http://ant.apache.org/bindownload.cgi?S\\_TACT=C34409NW](http://ant.apache.org/bindownload.cgi?S_TACT=C34409NW).
9. Lo mismo hacer con gradle 4.7, link de descarga de gradle: <https://gradle.org/releases/>.
10. El último paso será editar las variables de entorno del sistema, Dar clic en el menú inicio y buscar **Variables de entorno**, en la ventana que se abrirá dar clic en el botón **Variables de entorno...** en el apartado de variables del sistema crear las variables HOME del jdk de java, del gradle y de Android, tal como indica la Figura 40.

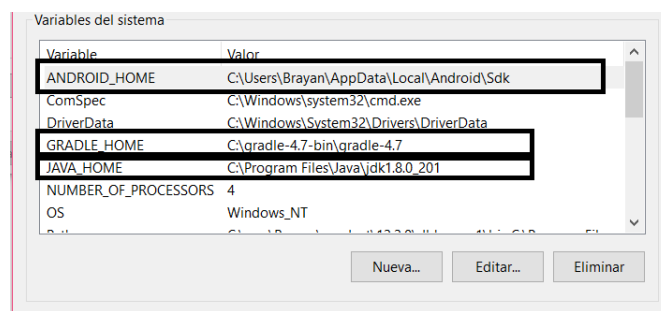
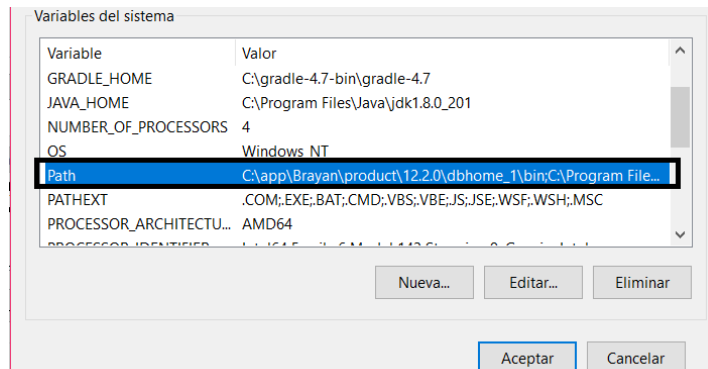


Figura 40. Creación de las variables del sistema.

Elaborado por: El investigador

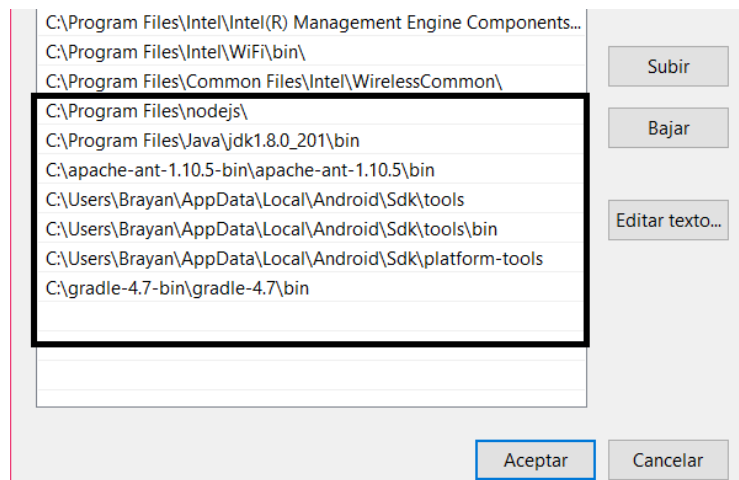
11. Ahora editar la variable de sistema llamada Path, para esto seleccionar la variable Path seleccionada en la Figura 41 y dar clic en el botón **Editar...**



*Figura 41. Edición variable del sistema Path.*

*Elaborado por: El investigador*

12. Copiar las rutas de las de las componentes mencionadas en los pasos: 1, 6, 7, 8, 9. Las variables del sistema deben quedar en el orden que muestra la Figura 42, si les hace falta alguna variable volver a realizar estos 12 pasos.

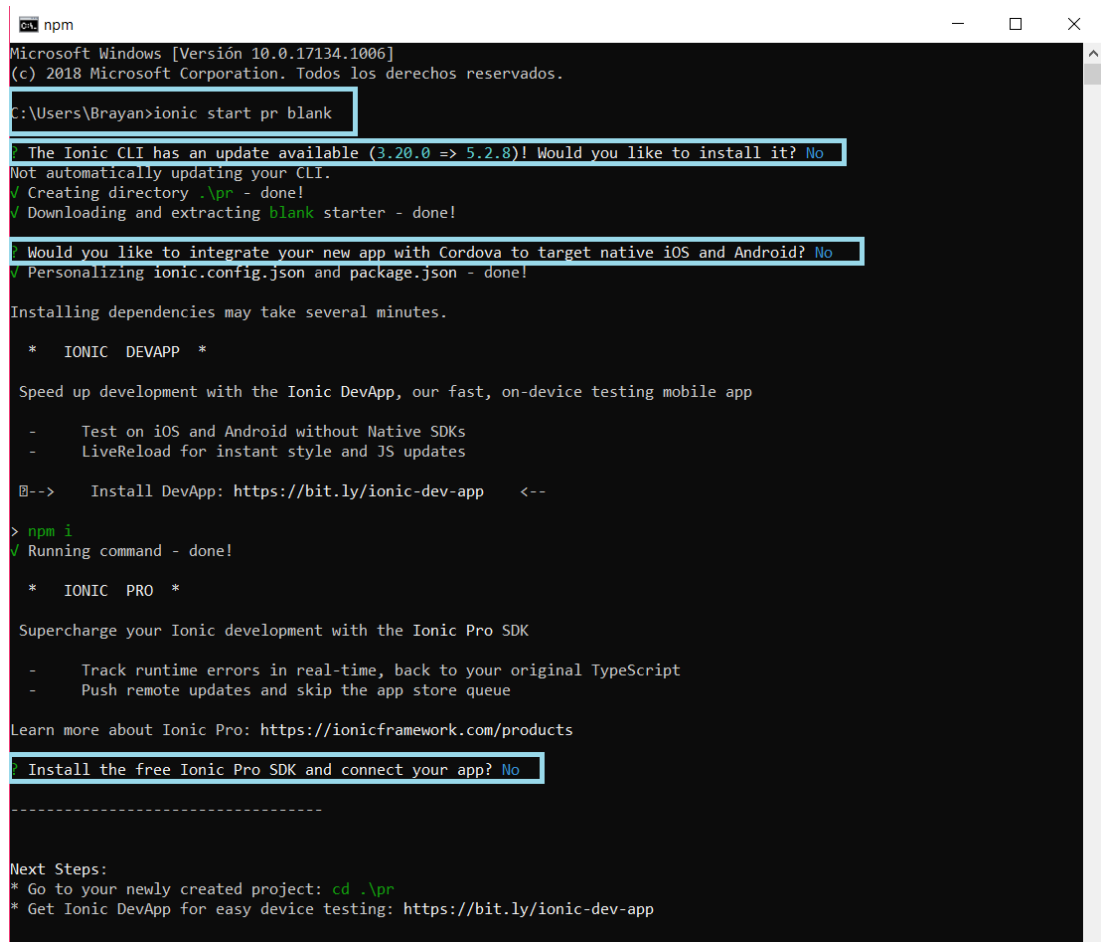


*Figura 42. Edición variable del sistema Path.*

*Elaborado por: El investigador*

Para crear la aplicación móvil híbrida abrir la consola de comandos, ingresar a la carpeta establecida para los proyectos de Ionic y escribir **ionic start geoagencias tabs**.

Después la consola me realizará preguntas a las que respondemos tal cual indica la Figura 43, esperar a que se cree la aplicación y ya podemos empezar a personalizarlo con la ayuda de Visual Studio Code.



```
npm
Microsoft Windows [Versión 10.0.17134.1006]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Brayan>ionic start pr blank

! The Ionic CLI has an update available (3.20.0 => 5.2.8)! Would you like to install it? No
Not automatically updating your CLI.
√ Creating directory .\pr - done!
√ Downloading and extracting blank starter - done!

! Would you like to integrate your new app with Cordova to target native iOS and Android? No
√ Personalizing ionic.config.json and package.json - done!

Installing dependencies may take several minutes.

* IONIC DEVAPP *

Speed up development with the Ionic DevApp, our fast, on-device testing mobile app

- Test on iOS and Android without Native SDKs
- LiveReload for instant style and JS updates

@--> Install DevApp: https://bit.ly/ionic-dev-app <--

> npm i
√ Running command - done!

* IONIC PRO *

Supercharge your Ionic development with the Ionic Pro SDK

- Track runtime errors in real-time, back to your original TypeScript
- Push remote updates and skip the app store queue

Learn more about Ionic Pro: https://ionicframework.com/products

! Install the free Ionic Pro SDK and connect your app? No

-----

Next Steps:
* Go to your newly created project: cd .\pr
* Get Ionic DevApp for easy device testing: https://bit.ly/ionic-dev-app
```

Figura 43. Creación de una aplicación en Ionic.

Elaborado por: El investigador

### Estructura de un proyecto Ionic

**Carpeta node modules:** Esta carpeta son las dependencias de npm que vienen definidas en el package.json e instaladas en local dentro del proyecto. Esta carpeta trabaja por sí sola, no se debe modificar nada allí.

**Carpeta platforms:** en esta carpeta se encuentra los archivos de cada plataforma en la que se ejecutará la aplicación desarrollada, ejemplo si necesitamos añadir la plataforma Android, allí se ubicará un subdirectorio “android”.

**Carpeta plugins:** en esta carpeta se sitúan todos los plugins que se necesiten para construir la aplicación, estos plugins pueden ser propios de ionic o de terceros.

**Carpeta resources:** esta carpeta contiene el icono de la aplicación y el icono del splash, los mismos que se pueden modificar cumpliendo ciertos requisitos que Ionic impone.

**Carpeta src:** Esta carpeta contiene directorios clasificados, por ejemplo, el directorio “pages”, donde se irán colocando las pantallas de la aplicación, así como la carpeta “assets” que son archivos de imágenes y otro tipo de materiales externos que vas a usar en las páginas, o el directorio app, que es la raíz de nuestra aplicación.

Encontrarás archivos sin carpetas contenedoras, como el index.html, que es el punto de arranque de la aplicación web. Encontrarás también el archivo service-worker.js, que es el service worker (de las Progressive Web Apps) que usarás si deseas publicar tu proyecto de Ionic como una app para la web. O el manifest.json, también para definir una Progressive Web App.

**Carpeta www:** En la carpeta "www" están los archivos que se producen cuando realiza toda la parte de la transpilación del TypeScript, el compilado de los archivos Sass. En ese proceso, todos los archivos de "src", se transforman y se vuelcan en la carpeta "www". Esos archivos producidos en "www" son los archivos de una aplicación web, con Angular, que se podrán visualizar correctamente en el navegador.

En esta carpeta no deberías editar nada, puesto que los cambios se machacarán la próxima vez que se sirva el proyecto para visualizarse a través del navegador, o al hacer el build.

**ionic.config.json:** Contiene información básica sobre la configuración del proyecto.

**package.json:** Contiene paquetes y dependencias de nodeJS.

**tconfig.json y tslint.json:** Son archivos que contienen información necesaria a la hora de compilar TypeScript.

### 3.4.11. Desarrollo del Sprint 3

*Tabla 31. Desarrollo del Sprint 3.*

<b>Nombre Sprint</b>	<b>Id</b>	<b>Título</b>
Sprint 3	1	Desarrollo de la pantalla de geolocalización de las agencias
	2	Desarrollo de la pantalla de inicio

*Elaborado por: El investigador*

## Desarrollo del Sprint 3 mostrado en la Tabla 31.

### Id 1. Desarrollo de la pantalla de geolocalización de las agencias.

- Para la localización de las agencias se utilizará el API “Maps SDK for Android e iOS” de Google, mismos que son gratuitos. Para el cual se necesita una cuenta de correo Gmail, en el cual se debe crear un proyecto de la manera que indica la Figura 44.

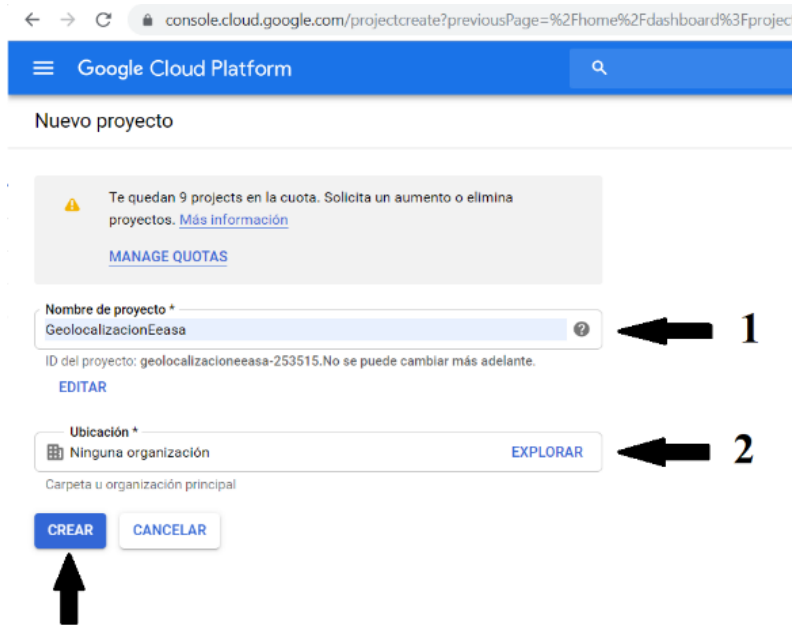


Figura 44. Creación de un proyecto en Google Cloud.

Elaborado por: El investigador

- Luego habilitar las siguientes API's de GoogleMaps: SDK for Android y Maps SDK for iOS y copiar la clave de la API, dicha clave se utilizará en el código de la aplicación. Ver Figura 45.

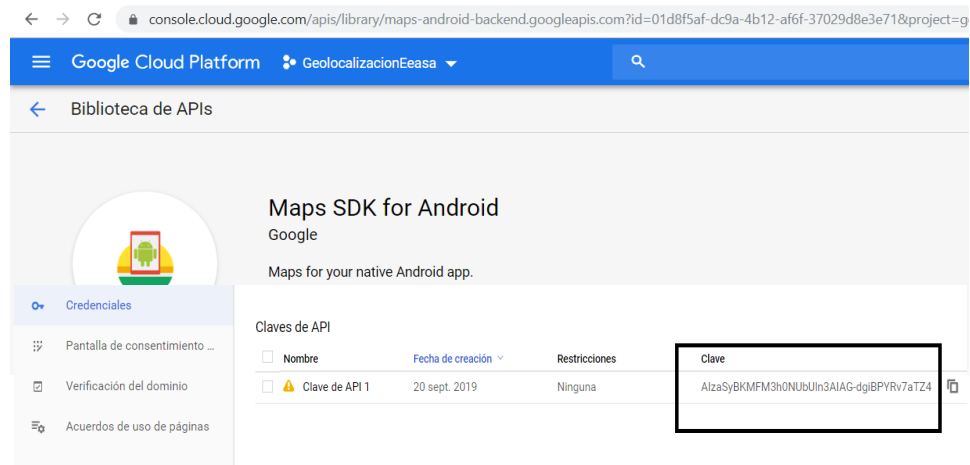
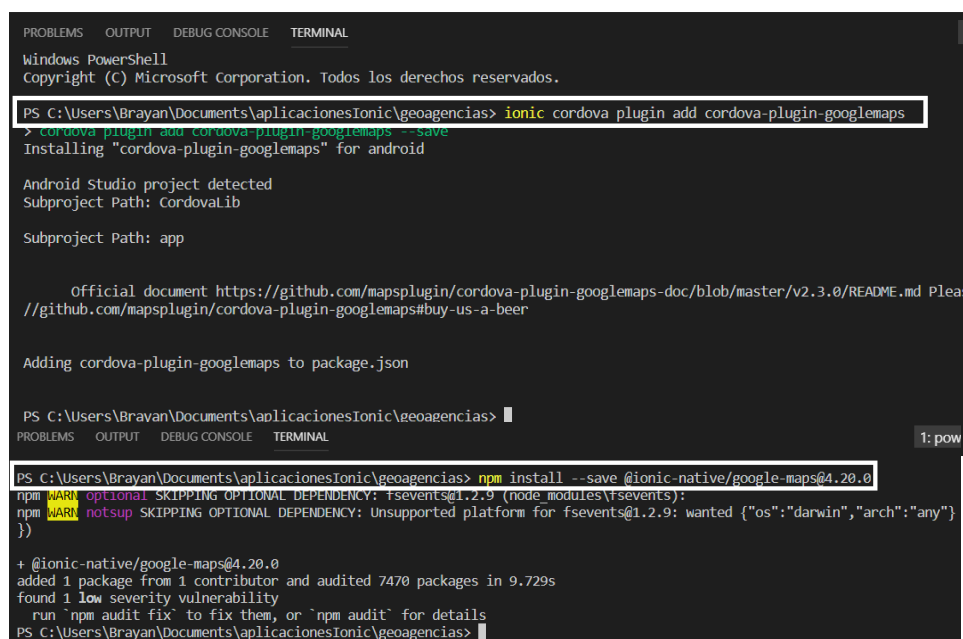


Figura 45. Habilitar API de GoogleMaps.

Elaborado por: El investigador

- Utilizando la consola de comandos, ubicarse en la carpeta raíz de la aplicación y agregar el plugin de google Maps a la aplicación creada en el id 3 del Sprint 2.
- Comando: **ionic cordova plugin add cordova-plugin-google-maps**. Luego instalar el plugin con el siguiente comando: **npm install --save @ionic-native/google-maps@4.20.0**. En la Figura 46 se muestra los resultados de cada comando ejecutado.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Brayan\Documents\aplicacionesIonic\geoagencias> ionic cordova plugin add cordova-plugin-googlemaps
> cordova plugin add cordova-plugin-googlemaps --save
Installing "cordova-plugin-googlemaps" for android

Android Studio project detected
Subproject Path: CordovaLib

Subproject Path: app

Official document https://github.com/mapsplugin/cordova-plugin-googlemaps-doc/blob/master/v2.3.0/README.md Please
//github.com/mapsplugin/cordova-plugin-googlemaps#buy-us-a-beer

Adding cordova-plugin-googlemaps to package.json

PS C:\Users\Brayan\Documents\aplicacionesIonic\geoagencias>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: pow

PS C:\Users\Brayan\Documents\aplicacionesIonic\geoagencias> npm install --save @ionic-native/google-maps@4.20.0
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"}
})
+ @ionic-native/google-maps@4.20.0
added 1 package from 1 contributor and audited 7470 packages in 9.729s
found 1 low severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
PS C:\Users\Brayan\Documents\aplicacionesIonic\geoagencias>
```

Figura 46. Agregar plugin de GoogleMaps.

Elaborado por: El investigador

- Copiar las siguientes líneas de código en el archivo de configuración del proyecto config.xml.
- 

```
<widget ...>
...
<preference name="GOOGLE_MAPS_ANDROID_API_KEY" value="(api key)" />
<preference name="GOOGLE_MAPS_IOS_API_KEY" value="(api key)" />
...
</widget>
```

Donde:

(api key). - es la clave que copiada en el segundo paso del Sprint 3.

- En Ionic cada contenedor posee 3 clases nomClase.ts, nomClase.scss, nomClase.html, En la clase nomClase.ts que se desea mostrar el mapa programar el funcionamiento.
- El siguiente código sirve para graficar en el mapa todas las agencias que dispone la Empresa Eléctrica Ambato, pero se mostrará las más cercanas a la ubicación actual.

```

/*
 * Método para cargar el mapa y sus elementos
 */
loadMap(){
  //Muestra barra de carga mientras esta listo el Mapa
  const loading = this.loadingCtrl.create();
  loading.present();
  //
  let opcionesMapa: GoogleMapOptions = {
    mapType:GoogleMapsMapTypeId.ROADMAP, //MODODO de mapa
    controls:{
      compass: true, //brujula
      myLocationButton: true, //boton de mi ubicacion
      zoom: true //aceptar zoom
    }, camera: {
      zoom: 14,
      tilt: 30
    }
  };
  this.map = GoogleMaps.create('map_canvas', opcionesMapa);
  this.map.one(GoogleMapsEvent.MAP_READY).then(
    () => {
      loading.dismiss();
      this.obtenerPosicionMarker();
      this.datos.forEach((element: any) => {

        let marker: Marker = this.map.addMarkerSync({
          title: element.EUGA_NOMBRE,
          position: {
            lat: parseFloat(element.EUGA_LATITUD),
            lng: parseFloat(element.EUGA_LONGITUD)
          },
          icon: '#DB3B3B'
        });
        marker.on(GoogleMapsEvent.INFO_CLICK).subscribe(() => {
          this.navCtrl.push(MasInformacionPage,{datosLocalizacion:element});
        });
      });
    }
  ).catch(
    error => {
      console.log('Error: ' + error);
    }
  );
}

```



```

}

/*
 * Metodo para graficar un marcador en base a la posicion actual
 */
obtenerPosicionMarker(){
  this.map.getMyLocation().then(
    respuesta => {
      this.map.moveCamera({
        zoom: 14,
        target: {lat: -1.2427,lng: -78.6265}//respuesta.latLng
      });
      this.map.addMarker({
        title: 'Mi Ubicación!',
        icon: '#3fa9f5',
        position: {lat: -1.2427,lng: -78.6265}//respuesta.latLng
      });
    }).catch(
      error => {
        console.log('Error: ' + error);
      });
}

```

- Los datos de longitud y latitud se obtienen desde el servicio web realizado en el Sprint 1. Para consumir el servicio web en Ionic, crear un proveedor con el comando **ionic g provider nomProvider**. En el mismo proveedor codificar el método para obtener una API RestFul de la siguiente manera.

```

public obtenerAgencias(){
  return new Promise(resolve => {

  this.http.get('http://190.95.194.75:7001/appeeasa/rest/eeasa/obtenerAgencias
  ').subscribe(data => {
    resolve(data);
  }, error => {
    console.log(error);
  });
});
}

```

- En la clase .html donde se mostrará el mapa, dentro de una etiqueta div crear un id el cual se dará un estilo en la página .scss. Este proceso es importante, en el caso de no hacerlo no se mostrará el mapa porque esta invisible.

## Código de la clase .html y .scss

mapa-geolocalizacion.html

```
<div id="map_canvas">
```

mapa-geolocalizacion.scss

```
#map_canvas{  
    display: block;  
    height: 100%;  
    width: 100%;  
}
```

Finalmente se muestra en la Figura 47 y 48 las capturas de pantalla tomadas de las pantallas creadas, como se puede observar en las 2 figuras antes mencionadas muestran la pantalla tanto en un dispositivo iOS como Android.

## Vistas de la pantalla de geolocalización en iOS

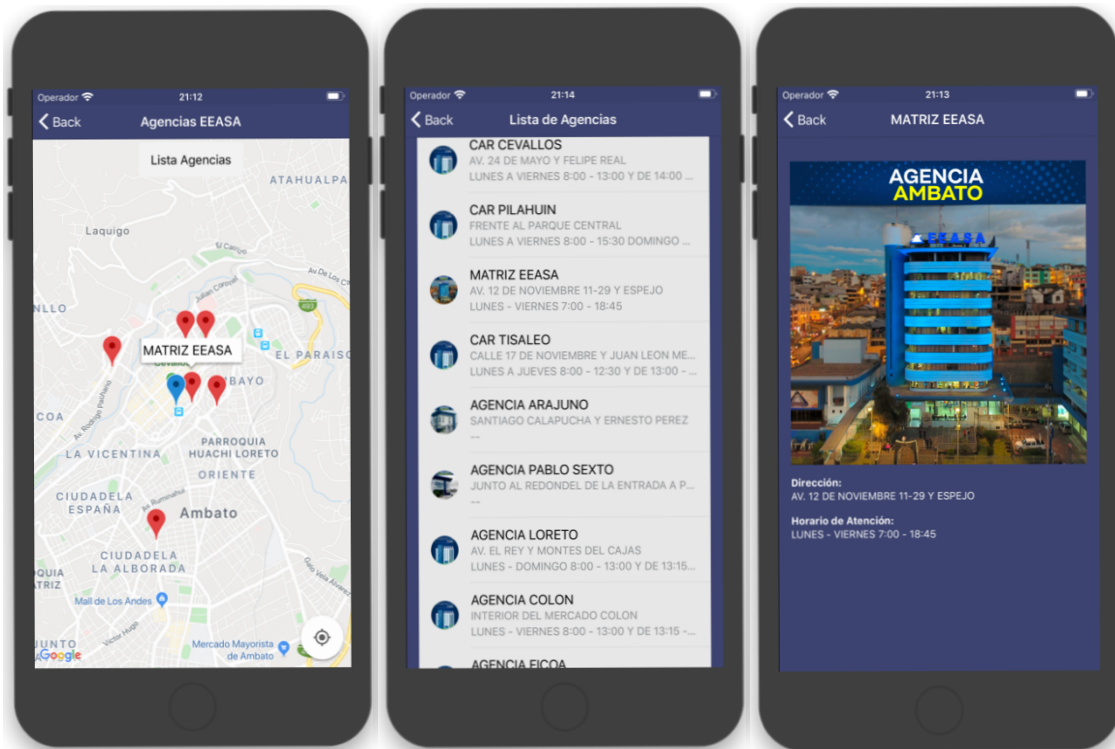


Figura 47. Vistas de la pantalla de geolocalización en iOS.

Elaborado por: El investigador

## Vistas de la pantalla de geolocalización en Android

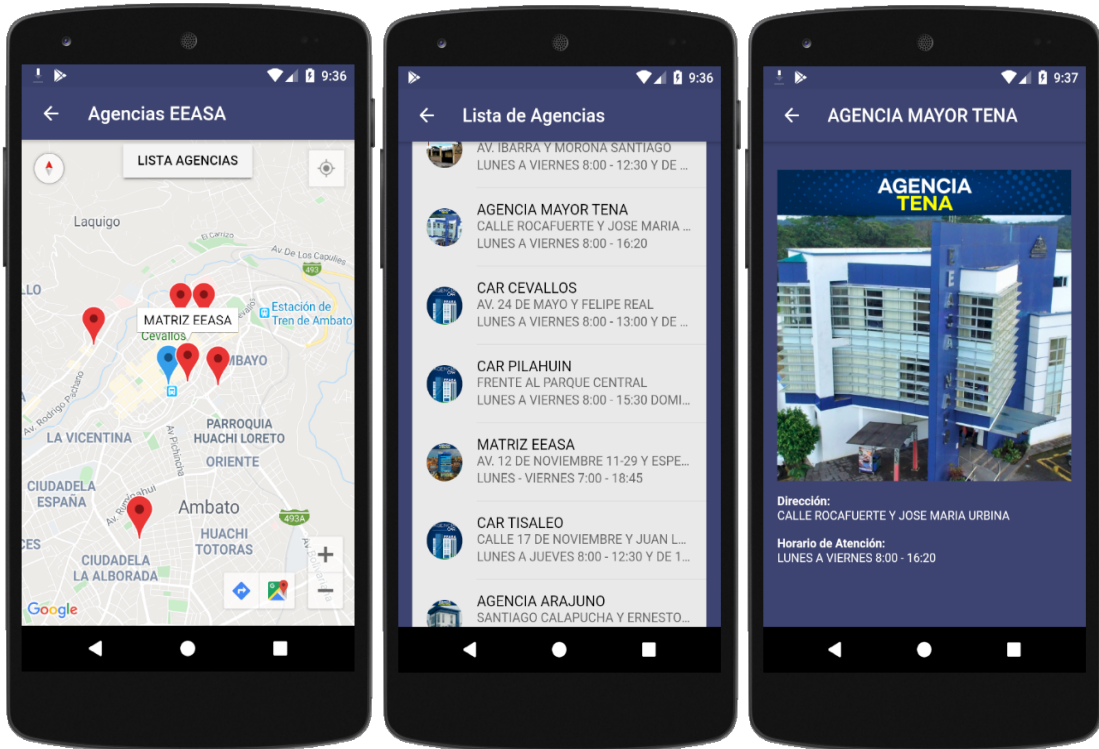


Figura 48. Vistas de la pantalla de geolocalización en Android

Elaborado por: El investigador.

### Id 2. Desarrollo de la pantalla de inicio

Para realizar el diseño de la pantalla principal, se utiliza un sistema de grid, el cual me permite agregar elementos libremente no solo de forma horizontal o vertical, como los layouts tradicionales en el caso de Android. Para realizar la pantalla mencionada ver el código siguiente.

```
<ion-header>
  <ion-toolbar color="easa">
    <ion-buttons start>
      <a href="https://www.facebook.com/EmpresaElectricaAmbato/">
        <button ion-button icon-only color="royal">
          <ion-icon name="custom-facebook"></ion-icon>
        </button>
      </a>
    </ion-buttons>
    <ion-title id="centrado">
      
    </ion-title>
  </ion-header>
```

```

        <ion-buttons end>
        <a
href="https://twitter.com/EEASA_RCN?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5
Eauthor">
        <button ion-button icon-only color="royal">
        <ion-icon name="custom-twitter"></ion-icon>
        </button>
        </a>
        </ion-buttons>
    </ion-toolbar>
</ion-header>

<ion-content class="bg">
    <div id="contenedorSlides">
        <ion-slides>
            <ion-slide *ngFor="let datos of this.datosImagenes">
                
            </ion-slide>
        </ion-slides>
    </div>
</ion-content>

<ion-footer color="footer">
    <div id="contenedorIcons">
        <ion-grid text-center>
            <ion-row>
                <ion-col id="columnIcons" (click)="abrirPlanilla()">
                    <br />
                </ion-col>
                <ion-col id="columnIcons" (click)="abrirAgencias()">
                    <br />
                </ion-col>
                <ion-col id="columnIconContacto">
                    <a
href="https://api.whatsapp.com/send?phone=593984737881&text=Es%20un%20placer%20aten
der%20su%20requerimiento,%20saludos%20cordiales">
                        <br />
                    </a>
                </ion-col>
            </ion-row>
        </ion-grid>
    </div>
</ion-footer>

```

La Figura 49 muestra una captura de la pantalla de inicio de la aplicación dsarrollada tanto para iOS y Android.

### Vista de la pantalla de inicio en iOS y Android

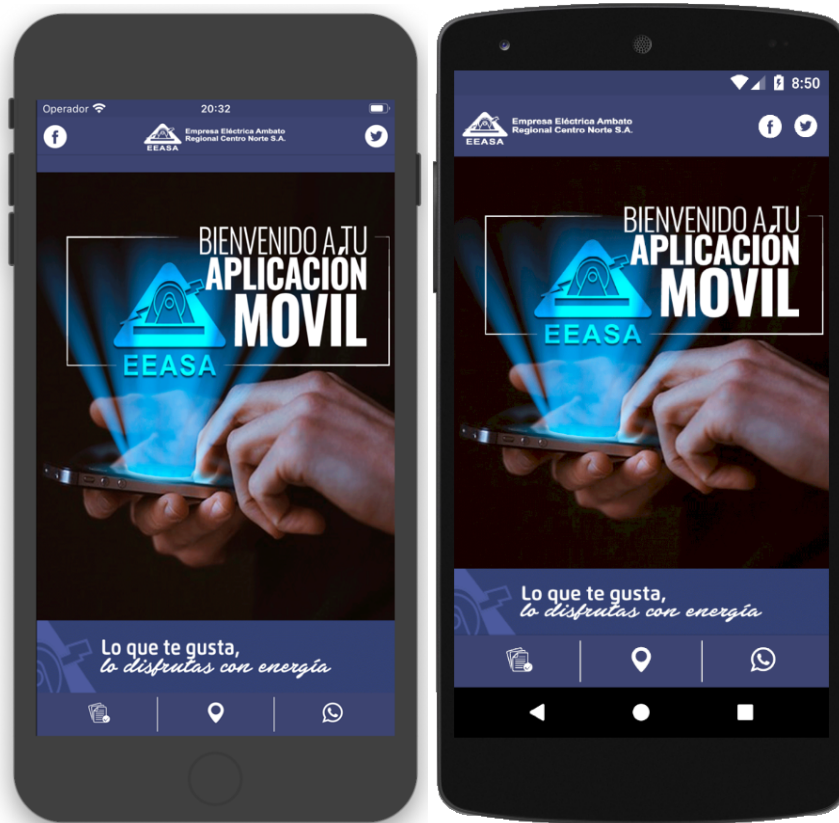


Figura 49. Vista de la pantalla principal en iOS y Android.

Elaborado por: El investigador

### 3.4.12. Desarrollo del Sprint 4

Tabla 32. Desarrollo del Sprint 4.

Nombre Sprint	Id	Título
Sprint 4	1	Desarrollo del módulo para consultas de la planilla eléctrica.

Elaborado por: El investigador

### Desarrollo del Sprint 4 que se muestra en la Tabla 32

## Id 1. Desarrollo del módulo para consultas de la planilla eléctrica.

El módulo para la consulta de planillas eléctricas se lo dividirá en 3 segmentos, donde el segmento uno visualizará la información del cliente asociado en la EEASA, el segmento dos mostrará el detalle de las facturas pendiente y el último segmento, muestra un gráfico de barras con el historial de consumo de los últimos doce meses.

### Segmento uno. - Método para obtener los datos del cliente.

```
obtenerDatosCliente() {
  this.obtenerHistConsumo();
  //Datos Cliente
  const loader = this.loadingCtrl.create({ content: "Consultando..."});
  loader.present();
  this.eeasaLoc.obtenerDatosCliente(this.txtCuenta)
  .then(data => {
    this.datosCliente[0] = data;
    loader.dismiss();
    if (this.datosCliente[0].codigoResultado == "01" || this.datosCliente[0].codigoResultado == "02") {
      this.condAviso = true;
      this.strMensaje = this.datosCliente[0].mensajeResultado;
    }if (this.datosCliente[0].numFacturas == "0") {
      this.ValorDeuda = "Cliente no registra deuda!";
      this.strAviso = " ";
      this.strAvisoFecha = "";
    }
  }else{
    this.strMensaje = "";
    this.ValorDeuda = "";
    this.strAvisoFecha = this.datosCliente[0].fechaMaxPago;
    this.strAviso = "Por favor pagar hasta: ";
  }
  },error =>{
    console.log(error);
    loader.dismiss();
  });
});
```

la Figura 50 muestra una captura de pantalla del segmento uno: obtener los datos del cliente. Si el cliente no tiene deuda le aparecerá el valor a pagar en 0, caso contrario le el cliente podrá visualizar el valor a pagar, además de la fecha máximo de pago.



Figura 50. Vista del Segmento uno

Elaborado por: El investigador

### Segmento dos. - Método para obtener información de las facturas pendientes de pago

```

this.eeasaLoc.obtenerFacturasDetalle(this.txtCuenta).subscribe(
  (data) => {
    if (data[0] == null){
      this.strAvisoDetalle = "Cliente no registra deuda!";
      this.gridDetalle = false;
    }else{
      this.gridDetalle = true;
      this.datosFacDetalle = data;
      this.strAvisoDetalle = "";
    }
  },
  (error) => {
    console.log(error);
  }
)}

```

La Figura 51 muestra una captura de pantalla del Segmento dos: obtener información de las facturas pendientes de pago. Aquí se detalla la información respecto al pago, es decir la fecha de emisión del estado de cuenta de la planilla eléctrica, su lectura anterior y la lectura actual, la cantidad de kWh consumidos durante el mes.



Figura 51. Vista del Segmento dos.

Elaborado por: El investigador

### Segmento tres. - Método para mostrar el gráfico de barras con el historial de consumo.

```

mostrarGraficoBarras(){
  this.condGrafico =true;
  let canvas = this.chartCanvas.nativeElement;
  var densityData = {
    label: 'Total kWh',
    data: this.valConsumo,
    backgroundColor: [
      '#3D4471', '#3D4471', '#3D4471', '#3D4471', '#3D4471',
      '#3D4471', '#3D4471', '#3D4471', '#3D4471', '#3D4471',
      '#3D4471', '#3D4471'
    ],
  },
  hoverBorderWidth: 0};

```



```

var chartOptions = {
  scales: {
    yAxes: [{
      barPercentage: 1
    }]},
  elements: {
    rectangle: {
      borderSkipped: 'left',
    }
  }
};

this.barChart = new Chart(canvas, {
  type: 'horizontalBar',
  data: {
    labels: this.valMesConsumo,
    datasets: [densityData],
  },
  options: chartOptions
});

```

La figura 52 muestra una captura de pantalla del segmento tres: mostrar el gráfico de barras con el historial de consumo de los últimos doce meses.



*Figura 52. Vista del segmento tres*

*Elaborado por: El Investigador*

Todo el código mostrado en el segmento 1 – 2 y 3, realiza una petición GET del servicio web, el mismo que devolverá un arreglo tipo json, para poder deserializarlo y mostrar información descifrable por el usuario.

### 3.4.12. Desarrollo del Sprint 5

Tabla 33. Desarrollo del Sprint 5.

Nombre Sprint	Id	Título
Sprint 5	1	Implementar la aplicación móvil en la EEASA.

Elaborado por: El investigador

### Desarrollo del Sprint 5 que se muestra en la Tabla 33

#### Id 1. Implementar la aplicación móvil en la EEASA.

Para implementar la aplicación móvil desarrollada, la Empresa Eléctrica Ambato facilitó la cuenta de desarrollador de Google, la cual se necesita para poder subir la aplicación a la tienda de Google Play, de tal manera que los usuarios puedan descargarse y usarla.

Para subir la aplicación primero crear el perfil del desarrollador, donde registrará los datos de la empresa o persona la cual realizó la aplicación, después se crea una versión de la aplicación, para ello subir el apk y listo.

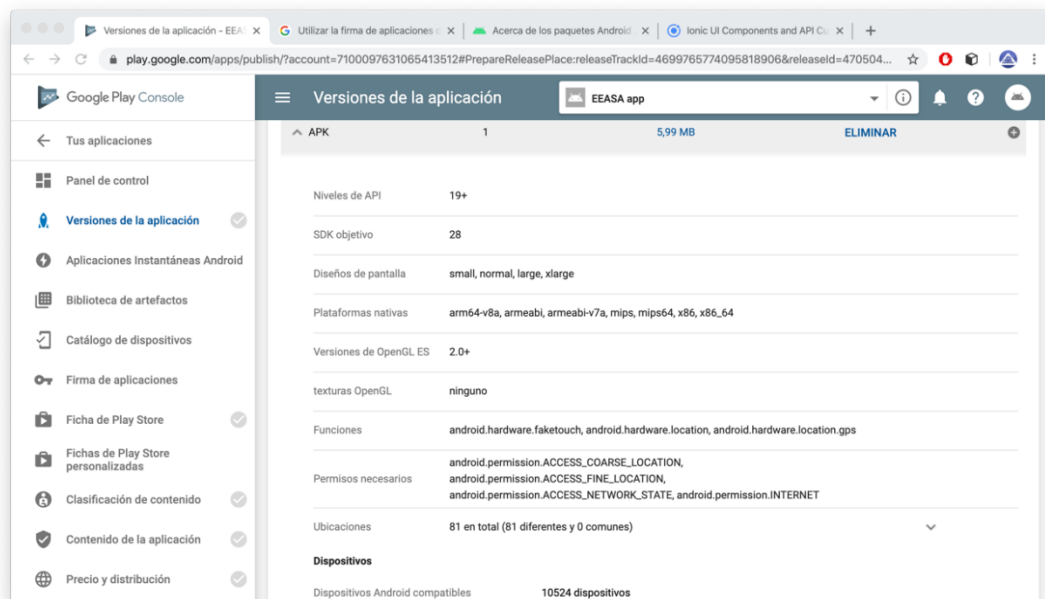


Figura 53. Versiones de la aplicación

Elaborado por: El Investigador

La Figura 53 indica la versión de la aplicación subida, además de los permisos que la aplicación necesita y los dispositivos en que la aplicación estará disponible una vez aprobada y publicada la aplicación.

Después, se debe llenar un cuestionario para la clasificación del contenido de la aplicación, como se muestra en la Figura 54 y la aplicación queda lista para ser revisada y aceptada por el personal de Google, este proceso puede tardar varios días.

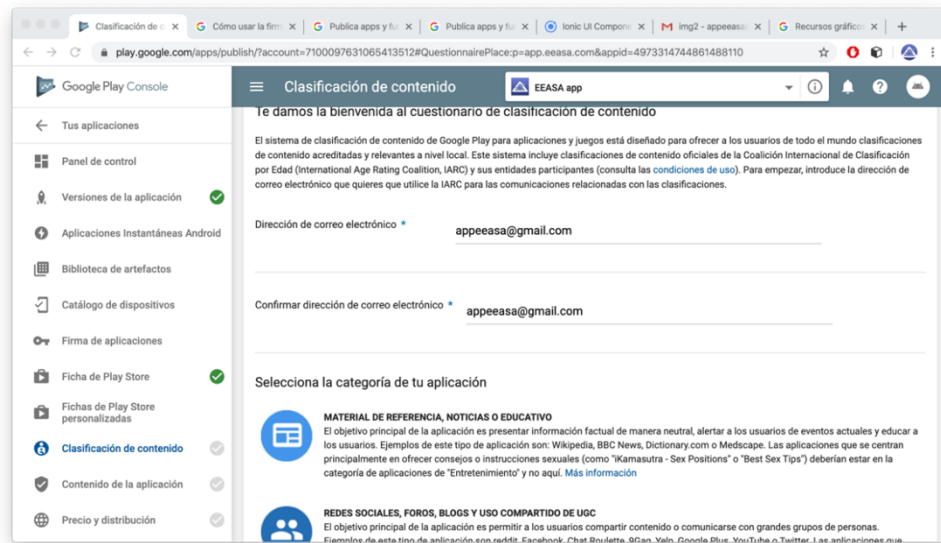


Figura 54. Encuesta clasificación de contenido

Elaborado por: El Investigador

El equipo de Google notificará si la aplicación fue aceptada o caso contrario notificará los cambios que se debe realizar para poder ser publicada en la tienda de Google Play.

El equipo de Google tardó 4 días en revisar la aplicación y finalmente publicar en Google Play Store. La figura 55 muestra que la aplicación esta disponible en la tienda de Google, la cual puede ser descargada para el libre uso de los usuarios del servicio eléctrico que la Empresa Eléctrica Ambato ofrece.

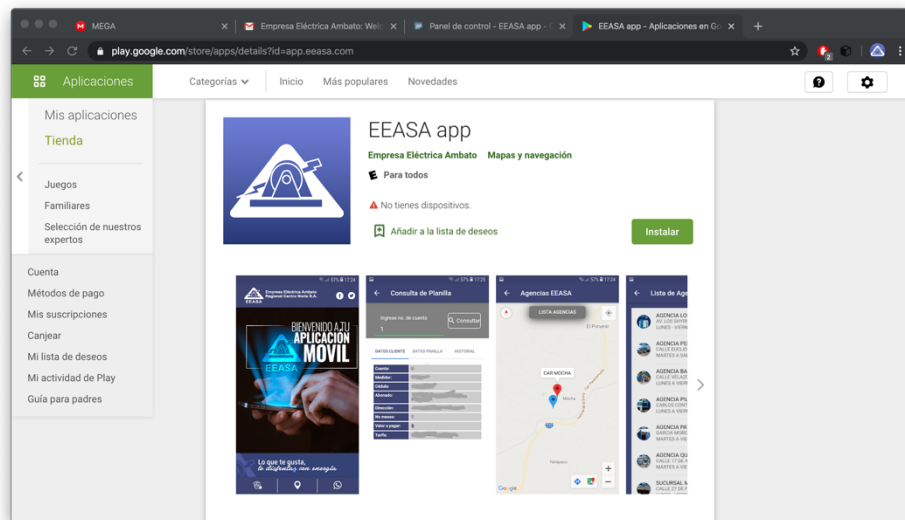


Figura 55.- Aplicación publicada en la Play Store

Realizado por: El investigador.

## **CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- La aplicación móvil desarrollada cumplió con las expectativas del presidente ejecutivo de la Empresa Eléctrica Ambato porque se realizó un levantamiento de requisitos para identificar los procesos y requerimientos funcionales de la aplicación, mismos que fueron aplicados durante el desarrollo.
- Durante la investigación de frameworks que permiten desarrollar aplicaciones móviles híbridas, se escogió el framework Ionic por ser un framework gratuito y fácil de aprender.
- Se implementó la aplicación, publicándola en la tienda de Google Play para que todos los usuarios con dispositivos móviles cuyo sistema operativo sea Android se beneficien y eviten congestiónamiento y largas filas de los puntos de información de la EEASA.

### **4.2 Recomendaciones**

- Se recomienda al Administrador de la aplicación web administrativa, mantener actualizada la información de las agencias de la Empresa Eléctrica Ambato, ya que los usuarios necesitan obtener datos reales y actualizados.
- Crear actualizaciones futuras con más funcionalidades para que los usuarios estén satisfechos con los servicios que la Empresa Eléctrica Ambato ofrece.
- La aplicación subida a la Play Store actualmente está funcionando para dispositivos móviles cuyo sistema operativo tengan Android 8 o alguna versión menor, Se recomienda para que la aplicación funcione en todas las versiones de Android, adquirir un servidor para configurar el servidor de aplicaciones y

alojar los servicios web con certificados https, dichos certificados son requeridos por las políticas de Google para aplicaciones que manejen Api's Rest y vayan a ser consumidos por dispositivos con Android 8 en adelante.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] L. Yaguapaz, “UNIVERSIDAD TÉCNICA DEL NORTE Facultad de Ingeniería en Ciencias Aplicadas Carrera de Ingeniería en Sistemas Computacionales ESTUDIO DEL FRAMEWORK IONIC 2 PARA EL DESARROLLO DE APLICACIONES MÓVILES HIBRIDAS Trabajo de grado previo a la obtención del título,” 2018.
- [2] E. Cano Insa, “Web App con Ionic y acceso a servicios Rest,” pp. 37–38, 2015.
- [3] J. A. Salazar Cardona, D. A. Angarrita, and J. D. Montoya, “Tendencias en desarrollo móvil bajo las tecnologías Android e IOS,” *Rev. Investig. la Fac. Ing. EAM*, vol. 3, pp. 28–35, 2016.
- [4] H. Li, “RESTful Web service frameworks in Java,” *2011 IEEE Int. Conf. Signal Process. Commun. Comput. ICSPCC 2011*, no. 1, 2011.
- [5] T. D. E. L. Tfg, C. Generator, A. Cordova, R. M. Gogonea, and R. Fecha, “Trabajo de final de grado,” 2016.
- [6] M. E. Joorabchi, A. Mesbah, and P. Kruchten, “Real challenges in mobile app development,” *Int. Symp. Empir. Softw. Eng. Meas.*, pp. 15–24, 2013.
- [7] D. Lisandro and P. Thomas, “Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles,” *Inst. Investig. en Informática LIDI*, pp. 766–776, 2015.
- [8] W. Danielsson, “React Native application development – A comparison between native Android and React Native,” p. 70, 2016.
- [9] A. Julián, A. Zamora, and M. S. D. Kutzias, “Estudio comparativo de frameworks multiplataforma para desarrollo de aplicaciones móviles,” 2018.
- [10] A. Lic and L. Nahuel, “Desarrollo de Aplicaciones Móviles Multiplataforma,” 2017.
- [11] E. Carrera, “DISEÑO DE UN JUEGO APLICANDO TECNOLOGÍA MÓVIL PARA APRENDIZAJE MUSICAL EN NIÑOS Y ADOLESCENTES,” 2019.
- [12] C. J. PINCAY, “IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL, BASADA EN LA UTILIZACIÓN DEL FRAMEWORK ANGULARJS PARA OPTIMIZAR EL PROCESO DE PEDIDOS EN UN TALLER ARTESANAL DE GUAYAQUIL, DEDICADO A LA COMERCIALIZACIÓN DE LUBRICANTES,” 2016.

- [13] C. Griffith, *Mobile app development with Ionic 2 : cross-platform apps with Ionic, Angular, and Cordova*. 2017.
- [14] C. Azaustre, *Desarrollo web ágil con Angular.js*. 2014.
- [15] R. FLORES, “Sistema De Georreferenciación De Restaurantes En La Ciudad De La Paz Mediante El Uso De Smartphones,” 2016.
- [16] K. Schwaber and J. Sutherland, “La Guía definitiva de Scrum: Las Reglas de Juego,” *Scrum.org*, 2014.
- [17] T. Marcelo, “APLICACIÓN MÓVIL UTILIZANDO PLATAFORMA ANDROID PARA MEJORAR LA CALIDAD DEL SERVICIO DE CONSULTA DE INFORMACIÓN DE CONSUMO ELÉCTRICO DE LA EEASA EN LA EMPRESA BESIXPLUS CÍA. LTDA,” 2013.
- [18] D. Sandoval, “SISTEMA DE LOTERÍA DE APUESTAS DEPORTIVAS EN EL FÚTBOL CON SPRING FRAMEWORK PARA LA EMPRESA ALQUIMIASOFT S.A.,” *Repos. Univ. Técnica Ambato*, vol. 53, no. 9, pp. 1689–1699, 2019.
- [19] C. A. Morales -Machuca, “Estado del Arte: Servicios Web,” *Camoralesmagooglepagescom*, 2010.
- [20] D. S. ANAYANSI, “UNA APROXIMACIÓN MDA PARA LA CONVERSIÓN ENTRE SERVICIOS WEB SOAP Y RESTFUL.,” no. 10, 2013.
- [21] L. Jaime, “Sistema de reservas web para el servicio de transporte puerta a puerta aplicando web services.,” 2018.
- [22] V. José, “Desarrollo de Servicio Web RESTful para el Acceso a Base de Datos de Entrenamiento desde Android,” 2015.
- [23] J. Torres and J. Villagomez, “ELABORAR EL MANUAL PARA LA CONFIGURACIÓN DE UN SERVIDOR GLASSFISH, UTILIZANDO EL SISTEMA OPERATIVO GNU/LINUX,” 2009.
- [24] G. Cedeño, “PLATAFORMA DE TECNOLOGÍAS INTELIGENTES PARA LA AUTOGESTIÓN DE SALUD DE PACIENTES CON DIABETES Y ASMA CON EL APOYO DE UNA APLICACIÓN MÓVIL Y UN SISTEMA DE RECOMENDACIONES, ENFOCADO EN LA CREACIÓN DE NUEVOS NODOS DE TRABAJO PARA LOS NUEVOS REQUERIMIENTOS DE,” 2018.



## ANEXOS

### Anexo N° 1. – Encuesta para determinar la factibilidad del desarrollo de la aplicación móvil para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato.

#### UNIVERSIDAD TECNICA DE AMBATO

**Indicaciones:** Marque con una X la respuesta que usted considere

1.- ¿Conoce todas las agencias de la empresa eléctrica donde puede realizar los pagos de sus planillas eléctricas?

Si

No

2.- ¿Cuántas veces usted acude a los puntos de información para consultar el valor de su planilla eléctrica?

1

2

3

4

+ 4

3.- ¿Ha tenido problemas para consultar su planilla eléctrica en los puntos de información de la EEASA?

Si

No

4.- ¿Le gustaría saber el valor de su planilla eléctrica y donde realizar los pagos sin necesidad de acercarse a los puntos de información de la EEASA

Si

No

5.- ¿Posee de un Smartphone en la actualidad?

Si

No

6.- ¿Qué Sistema Operativo tiene su Smartphone?

IOS

Android

Windows Phone

Otros

7.- De la siguiente lista de dispositivos. ¿Cuál le gustaría usar para consultar su planilla eléctrica?

**Smartphone**

**Tablet**

**Smart Tv**

**Todos los anteriores**

8.- ¿Posee algún plan de internet, ya sea plan de datos móviles o para su hogar?

**Si**

**No**

**Gracias por su colaboración**

**Anexo N° 2. – Ficha de Requerimientos Aprobado en Comité informático de la Empresa Eléctrica Ambato.**

FICHA DE ESPECIFICACIÓN DE REQUERIMIENTOS			
<b>Nombre Proyecto:</b>	Aplicación móvil, para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato	<b>Fecha:</b>	14/08/2019
<b>Tipo de Requerimiento:</b>	<input checked="" type="checkbox"/> Nuevo Requerimiento	<b>Alcance del Requerimiento</b>	
<b>Requerimiento N° 1:</b>			
Desarrollar una aplicación web para administrar las coordenadas geográficas de las agencias.			
<b>Dirección / Unidad solicitante:</b>	Departamento Comercial		
<b>Prioridad Requerimiento</b>	Alto <input checked="" type="checkbox"/>	Medio <input type="checkbox"/>	Bajo <input type="checkbox"/>
<b>Versión del Requerimiento</b>	1		
<b>Requerimientos Funcionales:</b>			
<b>Descripción</b>	Se requiere almacenar información de las coordenadas geográficas de las agencias donde se puede realizar pagos de las planillas eléctricas de la EEASA.		
<b>Observaciones</b>	El desarrollo de la aplicación web debe involucrar lo siguiente: <ul style="list-style-type: none"> <li>- Base de datos para almacenar la información.</li> <li>- Listado de todas las agencias de la EEASA.</li> <li>- Listado de imágenes que aparecerán en la pantalla principal.</li> </ul>		
<b>Listado de Anexos</b>			
<b>N°</b>	<b>Nombre</b>	<b>Descripción</b>	
1			
<b>Responsable Requerimiento</b>	<b>Responsable Modulo</b>	<b>Tutor</b>	
Ing. Oscar Armas	Brayan Cujano	Ing. Carlos Nuñez	

FICHA DE ESPECIFICACIÓN DE REQUERIMIENTOS			
<b>Nombre Proyecto:</b>	Aplicación móvil, para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato	<b>Fecha:</b>	14/08/2019
<b>Tipo de Requerimiento:</b>	<input checked="" type="checkbox"/> Nuevo Requerimiento	<b>Alcance del Requerimiento</b>	
<b>Requerimiento N° 2:</b>			
Desarrollar Servicios Web para obtención de información de la base de datos creada en el requerimiento N °1			
<b>Dirección / Unidad solicitante:</b>	Departamento Comercial		
<b>Responsable Proyecto:</b>			
<b>Prioridad Requerimiento</b>	Alto <input checked="" type="checkbox"/>	Medio <input type="checkbox"/>	Bajo <input type="checkbox"/>
<b>Versión del Requerimiento</b>	1		
<b>Requerimientos Funcionales:</b>			
<b>Descripción</b>	Se requiere obtener información mediante servicios web para ser consumidos en la aplicación móvil.		
<b>Observaciones</b>	El desarrollo de los servicios web debe involucrar lo siguiente: - Métodos GET para la obtención de información.		
<b>Listado de Anexos</b>			
<b>Nº</b>	<b>Nombre</b>	<b>Descripción</b>	
1			
<b>Responsable Requerimiento</b>	<b>Responsable Modulo</b>	<b>Tutor</b>	
Ing. Oscar Armas	Brayan Cujano	Ing. Carlos Nuñez	



FICHA DE ESPECIFICACIÓN DE REQUERIMIENTOS			
<b>Nombre Proyecto:</b>	Aplicación móvil, para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato	<b>Fecha:</b>	14/08/2019
<b>Tipo de Requerimiento:</b>	<input checked="" type="checkbox"/> Nuevo Requerimiento	<b>Alcance del Requerimiento</b>	
<b>Requerimiento N° 3:</b>			
Desarrollar una pantalla de inicio que permita visualizar información de la Empresa Eléctrica Ambato			
<b>Dirección / Unidad solicitante:</b>	Departamento Comercial		
<b>Responsable Proyecto:</b>			
<b>Prioridad Requerimiento</b>	Alto <input checked="" type="checkbox"/>	Medio <input type="checkbox"/>	Bajo <input type="checkbox"/>
<b>Versión del Requerimiento</b>	1		
<b>Requerimientos Funcionales:</b>			
<b>Descripción</b>	Se requiere brindar a la ciudadanía una pantalla informativa con los principales sucesos de la Empresa eléctrica Ambato		
<b>Observaciones</b>	El desarrollo de la pantalla de inicio debe involucrar lo siguiente: <ul style="list-style-type: none"> <li>- Accesos directos a las redes sociales</li> <li>- Imgenes informativas sobre la empresa</li> <li>- Acceso directo al chat de WhatsApp de la empresa para reportar daños en el servicio.</li> </ul>		
<b>Listado de Anexos</b>			
<b>Nº</b>	<b>Nombre</b>	<b>Descripción</b>	
1			
<b>Responsable Requerimiento</b>	<b>Responsable Modulo</b>	<b>Tutor</b>	
Ing. Oscar Armas	Brayan Cujano	Ing. Carlos Nuñez	



FICHA DE ESPECIFICACIÓN DE REQUERIMIENTOS			
<b>Nombre Proyecto:</b>	Aplicación móvil, para la consulta de planillas eléctricas y geolocalización de las agencias de la Empresa Eléctrica Ambato	<b>Fecha:</b>	14/08/2019
<b>Tipo de Requerimiento:</b>	<input checked="" type="checkbox"/> Nuevo Requerimiento	<b>Alcance del Requerimiento</b>	
<b>Requerimiento N° 4:</b>			
<p>Desarrollar una pantalla que permita acceder a la información de planillas eléctricas de los clientes de la EEASA.</p> <p><b>Nota:</b> Para este módulo la EEASA dispone de un servicio web almacenado en el servidor, el cual puede ser utilizado para la adquisición de información.</p>			
<b>Dirección / Unidad solicitante:</b>	Departamento Comercial		
<b>Responsable Proyecto:</b>			
<b>Prioridad Requerimiento</b>	Alto <input checked="" type="checkbox"/>	Medio <input type="checkbox"/>	Bajo <input type="checkbox"/>
<b>Versión del Requerimiento</b>	1		
<b>Requerimientos Funcionales:</b>			
<b>Descripción</b>	Se necesita brindar a los clientes información sobre los consumos eléctricos, para evitar saturar los puntos de información que dispone la EEASA.		
<b>Observaciones</b>	<p>El desarrollo de la pantalla debe involucrar lo siguiente:</p> <ul style="list-style-type: none"> <li>- Datos del Cliente como: <ul style="list-style-type: none"> <li>• N° de cuenta</li> <li>• N° de cedula</li> <li>• Nombres y apellidos</li> <li>• Dirección del domicilio</li> <li>• N° de medidor</li> <li>• Tipo de Tarifa</li> </ul> </li> <li>- Datos de la Planilla como: <ul style="list-style-type: none"> <li>• N° de serie</li> <li>• N° de Documento</li> <li>• Fecha Emisión</li> <li>• Lectura Anterior</li> <li>• Lectura Actual</li> <li>• Consumo</li> <li>• Total, a pagar</li> </ul> </li> </ul>		



	<ul style="list-style-type: none"> <li>- Historial de Consumo             <ul style="list-style-type: none"> <li>• Mes</li> </ul> </li> <li>- Consumo kWh</li> </ul>		
<b>Listado de Anexos</b>			
<b>Nº</b>	<b>Nombre</b>		<b>Descripción</b>
<b>1</b>			
<b>Responsable Requerimiento</b>		<b>Responsable Modulo</b>	<b>Tutor</b>
<b>Ing. Oscar Armas</b>		<b>Brayan Cujano</b>	<b>Ing. Carlos Nuñez</b>

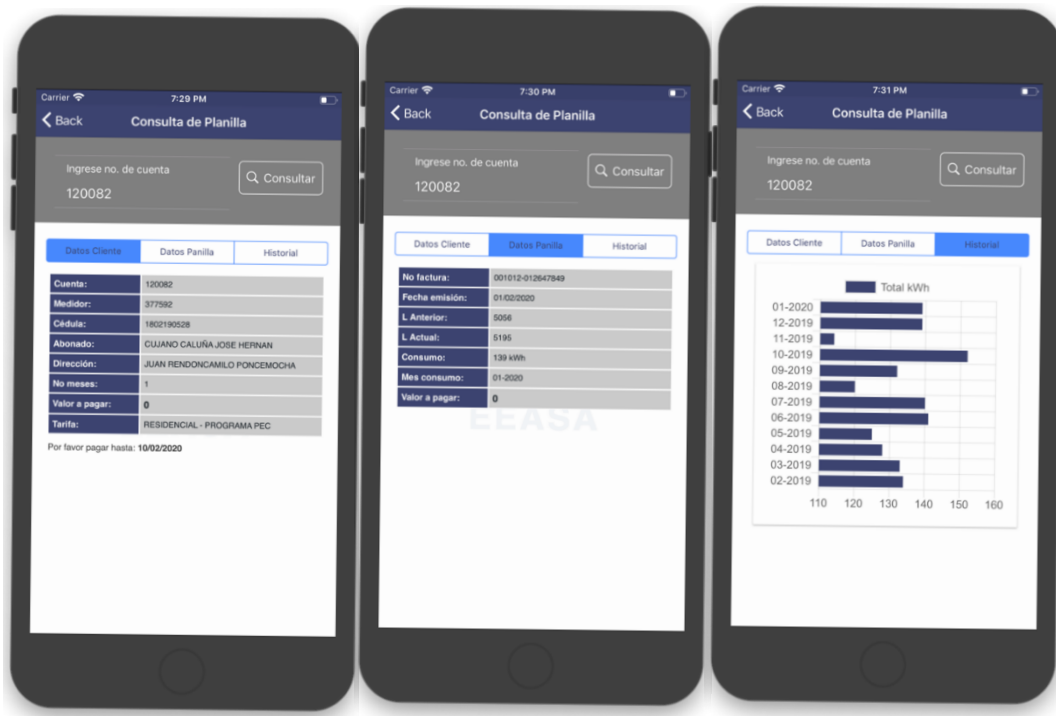
### Anexo N° 3. – Manual de usuario.

El siguiente manual tiene como finalidad mostrar el uso y funcionalidades que tiene la aplicación móvil.

- 1) **Pantalla principal**, en esta pantalla se podrá visualizar las noticias que tiene la EEASA para comunicar a los usuarios, puedes deslizar para la derecha y seguir viendo más noticias.



- 2) **Icono de consulta de planillas**, dar click en este icono y se abrirá una nueva ventana donde se ingresa el numero de cuenta, click en el botón **Consultar** y la ventana mostrará información de la planilla de consumo eléctrico.



- 3) **Icono de Geolocalización**, dar click en este icono, se abrirá una página graficando las agencias más cercanas en base a la ubicación actual, además si das click sobre el marker pintado de color rojo podras abrir directamente las indicaciones de como llegar en Google Maps. Para más información de las agencias, se puede dar click sobre la información del marker o sobre el botón **Lista Agencias**.



- 4) **Icono de WhatsApp**, dar click sobre este icono, si deseas chatear con el personal de la EEASA con el fin de reportar daños en los servicios o registrar alguna novedad.
  
- 5) - 6) **Icono de Facebook, icono de Twitter**, para seguir más de cerca o ponerse en contacto con el chat de la Empresa se puede dar click sobre estos botones y me dirigirá a las pagina oficiales de Facebook y Twitter de la EEASA.

## Anexo N° 4. – Carta de finalización del proyecto



### EMPRESA ELECTRICA AMBATO REGIONAL CENTRO NORTE S.A.

*Trabajando con energía..!*

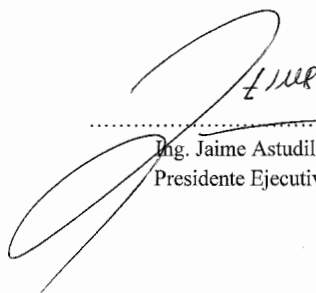
Ambato, 04 de marzo del 2020

Ingeniero Mg.  
Carlos Sánchez  
Presidente  
Unidad de Titulación  
Carrera de Sistemas Computacionales e Informática  
Facultad de Ingeniería en Sistemas, Electrónica e Industrial

En mi calidad de Presidente Ejecutivo de la Empresa Eléctrica Ambato Regional Centro Norte S.A. por medio de la presente manifiesto a usted, que el Sr: Brayan Patricio Cujano Soto, portador de la cédula de ciudadanía N° 1804728903, estudiante de Décimo nivel de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, culminó exitosamente el desarrollo e implementación del Trabajo de Titulación denominado “APLICACIÓN MÓVIL HÍBRIDA, PARA LA CONSULTA DE PLANILLAS ELÉCTRICAS Y GEOLOCALIZACIÓN DE LAS AGENCIAS DE LA EMPRESA ELÉCTRICA AMBATO”, en la empresa a la que represento.

Con estos antecedentes informo que la realización de este Trabajo de Titulación es de gran importancia para la empresa y solicito se proceda con la aprobación y trámite correspondiente.

Atentamente.



Ing. Jaime Astudillo  
Presidente Ejecutivo

Av. 12 de Noviembre 11-29 y Espejo Casilla 18-01-446  
Telf: (03) 2998600 Reparaciones: 136 Fax: (03) 2421265  
www.eeasa.com.ec e-mail: presidencia@eeasa.com.ec  
Ambato - Ecuador

