



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA: SISTEMAS INFORMÁTICOS

**TEMA: DESARROLLO DE UN TRADUCTOR DE PALABRAS DE ESPAÑOL A
QUICHUA (IMBABURA) PARA SISTEMAS ANDROID.**

AUTOR: LEONARDO ENRIQUE LÓPEZ BEJARANO

TUTOR: ING. JUAN CARLOS MORENO

AÑO 2015

ÍNDICE GENERAL

| | |
|--|----|
| ÍNDICE GENERAL | 1 |
| ÍNDICE DE GRÁFICOS..... | 3 |
| ÍNDICE DE TABLAS..... | 3 |
| INTRODUCCIÓN..... | 4 |
| ANTECEDENTES..... | 5 |
| ALCANCE | 7 |
| OBJETIVOS DEL TRABAJO DE TITULACIÓN | 8 |
| Objetivo General | 8 |
| Objetivos Específicos..... | 8 |
| CAPÍTULO I..... | 9 |
| 1 FUNDAMENTACIÓN TEÓRICA..... | 9 |
| 1.1 Lingüística del idioma Quichua | 9 |
| 1.2 Requerimientos mínimos del Sistema Android para el desarrollo de la aplicación..... | 10 |
| 1.3 Android 2.0/2.1 (Eclair), 2.2 (Frozen Yogurt) y 2.3 (Gingerbread) | 14 |
| 1.4 Android 3.0 (Honeycomb) | 16 |
| 1.5 Android 4.0 (Ice Cream Sandwich), 4.1/4.2 (Jelly Bean) y 4.3 (Key Lime Pie) | 16 |
| 1.6 Requerimientos mínimos de hardware para que se pueda ejecutar esta nueva versión de Android..... | 18 |
| 1.7 Diferencias entre versiones de sistema Android | 19 |
| 1.8 Java..... | 20 |
| 1.8.1 Características de Java | 20 |
| 1.9 SQLite | 22 |
| 1.10 Justificación metodológica | 25 |
| 1.11 METODOLOGÍA 3P | 28 |
| Principios que sustentan el modelo | 28 |
| CAPÍTULO II..... | 30 |
| 2 DIAGNÓSTICO | 30 |
| 2.1 Descripción del problema..... | 30 |
| 2.2 Identificación de los procesos unitarios. | 30 |
| 2.3 Modelo base | 31 |
| 2.4 DISEÑO DE LA PROPUESTA..... | 32 |
| 2.5 Desarrollo del prototipo. | 33 |
| 2.6 Presentación final de pantallas | 35 |

| | | |
|--------------------|--|----|
| 2.6.1 | Pantalla de traducción de palabras | 35 |
| 2.6.2 | Pantalla de ingreso de nuevas palabras | 35 |
| 2.6.3 | Pruebas de la aplicación..... | 35 |
| 2.6.4 | Dispositivos de prueba | 36 |
| 2.7 | Resultados obtenidos | 38 |
| 2.8 | Conclusiones | 38 |
| 2.9 | Recomendaciones..... | 39 |
| BIBLIOGRAFÍA | | 41 |
| ANEXO 1..... | | 44 |
| | Código Fuente de la aplicación. | 44 |
| ANEXO 2..... | | 62 |
| | Manual de instalación | 62 |
| GLOSARIO | | 65 |

ÍNDICE DE GRÁFICOS

| | |
|--|----|
| Gráfico 1-1 Estadísticas uso de Android | 10 |
| Gráfico 1-2 Versiones Android | 13 |
| Gráfico 1-3 Pantalla Android 2.3 | 14 |
| Gráfico 1-4 Pantalla Android 2.3 | 15 |
| Gráfico 1-5 Pantalla Android 2.3 | 15 |
| Gráfico 1-6 Pantalla Android 3.0 | 16 |
| Gráfico 1-7 Pantalla Android 4.0 | 17 |
| Gráfico 1-8 Pantalla Android 4.2 | 17 |
| Gráfico 2-1 Modelo Base | 30 |
| Gráfico 2-2 Pantalla de traducción de palabras | 32 |
| Gráfico 2-3 Pantalla de ingreso de palabras | 33 |
| Gráfico 2-4 Diagrama de flujo de procesos | 34 |
| Gráfico 2-5 Pantalla final de traducción de palabras | 35 |
| Gráfico 2-6 Pantalla final de ingreso de palabras | 35 |
| Gráfico 2-7 Cuadro estadístico de pruebas realizadas | 38 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1-1 Características de las versiones Android (Cont.) | 11 |
| Tabla 1-2 Características de Java (Cont.) | 20 |
| Tabla 1-3 Metodología de desarrollo de software | 26 |
| Tabla 1-4 Ponderación de las metodologías | 27 |
| Tabla 1-5 Ventajas y desventajas metodología 3P | 28 |
| Tabla 2-1 Modelo Base | 31 |
| Tabla 2-2 Formato para pruebas de funcionamiento | 36 |
| Tabla 2-3 Resultado de pruebas realizadas | 37 |

UNIVERSIDAD TECNOLÓGICA “ISRAEL”

INFORME FINAL DE RESULTADOS DEL TT

| | |
|--|--|
| Carrera: | INGENIERÍA EN SISTEMAS INFORMÁTICOS |
| Autor del TT: | Leonardo López Bejarano |
| Tema del TT: | Desarrollo de un traductor de palabras de Español a Quichua (Imbabura) para sistemas Android. |
| Articulación con la línea de Investigación Institucional: | Tecnología aplicada a la producción y sociedad |
| Sublínea de Investigación Institucional: | Simulación, desarrollo y automatización de procesos industriales, empresariales y de la sociedad |
| Fecha de presentación: | 1 de Junio del 2013 |

INTRODUCCIÓN

El Objetivo del presente proyecto es una aplicación que sea fácil de utilizar para la traducción de palabras del español al quichua que sea instalable en dispositivos móviles que dispongan del sistema operativo Android; se ha seleccionado este sistema operativo debido a ser uno de los sistemas operativos más comerciales y el cual brinda múltiples ventajas para el desarrollo de la aplicación.

El desarrollo de la aplicación trabaja desde la versión 2.3 Gingerbread que es una de las versiones más estables y comerciales del sistema Android hasta la versión 4.4 Kitkat, que actualmente está en los últimos dispositivos móviles.

Para el desarrollo de la aplicación se consideró el lenguaje de programación Java debido a ser un lenguaje multiplataforma e independiente del sistema operativo, el cual nos brinda las características necesarias para el desarrollo de la aplicación. La programación de Java esta realizado a través del IDE Eclipse que nos da el interfaz de programación.

La base de datos que se utilizó en el desarrollo de la aplicación es SQLite, un motor de base de datos que no necesita ser instalada y que puede ser transportable ya que es una librería del IDE Eclipse. Una de las características principales de la base de datos es que no necesita servidor para funcionar, por lo que se ajusta a las necesidades de aplicaciones para dispositivos móviles

Se seleccionó la traducción del idioma quichua debido a la difusión que se está realizando dentro de la provincia de Imbabura; esta difusión se la realiza a través de medios impresos radiales y digitales.

La difusión del idioma se la está realizando a través de TIC's para fomentarla con los jóvenes de la provincia, las nuevas generaciones están perdiendo el aprendizaje del idioma, lo que se pretende es dar el interés correspondiente para que el idioma no se pierda

En base a esta iniciativa y la necesidad de una aplicación donde se pueda utilizar un traductor del idioma español al idioma quichua (Imbabura) de una manera en la cual no necesariamente se deba utilizar una conexión de internet o la utilización de un navegador para realizar traducciones básicas entre los idiomas español y quichua.

Para la programación del aplicativo se utilizó la metodología 3P ya que se consideró la más apropiada por la comunicación que existe entre el programador y el usuario evitando la pérdida de tiempo en el desarrollo y cumpliendo las necesidades

Se constató a través del Google Play que existen aplicaciones instalables para sistemas Android de varios autores pero estas aplicaciones están generalizadas para el idioma Quichua sin definir una dialéctica o región donde la utilicen y la única aplicación específica es para quichua hablantes de la provincia de Chimborazo en vista de este análisis es que nos centramos en realizar el desarrollo de la herramienta de apoyo para la provincia de Imbabura.

ANTECEDENTES

En referencia al artículo publicado en el blog Ecuador del sur turismo y cultura, El idioma Quichua es originario de la época del Imperio Inca y es parte de una familia de lenguas así como el español, italiano, portugués; se extiende desde el sur de Colombia hasta el norte de Chile y Argentina, la mayor variedad de dialectos del

idioma quichua se encuentran en Ecuador y Perú. (Ecuador del sur turismo y cultura, 2014)

En la presentación realizada por (Ecuador LLaktapak, 2013, págs. 15-18) menciona lo siguiente: “El dialecto del Norte. Característica del habla kichwa de las provincias de Imbabura, y Pichincha...El dialecto Central. Hablado en Cotopaxi, Tungurahua, Chimborazo y Bolívar... El dialecto del Sur. Hablado en Azuay, Cañar y Loja...El dialecto Amazónico. Hablado en las provincias de la Amazonía ecuatoriana por los pueblos kichwas.”. El idioma quichua dentro de nuestro país se divide en 4 dialectos, cada uno de estos dialectos tiene diferencias y semejanzas entre sí, las regiones que hablan el idioma quichua son norte, centro, sur y Amazonía.

En el trabajo de titulación de las autoras “Dueñas y Peláez” se menciona lo siguiente: “El pueblo indígena de Otavalo está conformado por 157 comunas Kichwa-Otavalo, éstas se hallan esparcidas en las parroquias urbanas y rurales del cantón...Kichwa: idioma que adoptaron con la conquista inca, hoy ha perdido vigencia y ha sido declarado por la UNESCO lengua en peligro de extinción. El castellano es segunda lengua.”. El idioma quichua de a poco está perdiéndose es así el caso que una de las principales organizaciones a nivel mundial la ha considerado como lengua en peligro de extinción y el castellano que está considerada como segunda lengua para las comunas de esta provincia está ganando adeptos. (Dueñas Aguilar & Peláez Ponce, 2013, págs. 58, 59)

“La Asamblea Constituyente de Ecuador incluyó hoy el kichwa y el shuar como idiomas oficiales del país, en relación de interculturalidad con el español... El texto dice: "El castellano es el idioma oficial del Ecuador; el castellano, el kichwa (quechua) y shuar (idioma de indígenas de la Amazonía) son idiomas oficiales de relación intercultural". Una vez que los idiomas quichua y shuar se los incluyo como idiomas oficiales de nuestro país se propone la difusión del idioma para evitar que las nuevas generaciones pierdan el uso de la lengua. Desde hace unos años atrás se estaba viviendo un proceso de supresión idiomática porque los distintos gobiernos nunca han aceptado el carácter de este país como multilingüe (La hora, 2008)

“Unos ocho millones de personas en América del Sur se comunican en quechua, considerada la lengua indígena con más hablantes en toda América, con la mayor parte de hablantes distribuidos en Perú (de 4 a 5 millones), Ecuador (2,2 millones) y Bolivia (1,7 millones)”. Unas dos millones de personas en Ecuador se comunican en

quichua, considerada la lengua indígena con más hablantes en toda América pero está siendo considerado como una lengua que representa la pobreza es así que las nuevas generaciones no están aprendiendo el idioma quichua de parte de sus predecesores. (La Hora, AFP, 2011)

En las diferentes fuentes de aplicaciones para dispositivos Android se localizó dos tipos de aplicaciones las que necesitan de un navegador o browser y una conexión de internet para poder funcionar y las aplicaciones instalables o aplicaciones que fueron generados en archivos APK ,esta última no necesita de internet para funcionar, pero requieren ser instalados en el dispositivo móvil. De las aplicaciones que son instalables se encontró dos aplicaciones representativas en la traducción del idioma español a quichua, estas son “SimiDic” y “Habla Quechua” las ventajas de estas aplicaciones es la traducción de varios idiomas originarios de Sudamérica como son quichua, guaraní, aymara, en lo que corresponde al idioma quichua las traducciones corresponden a otras regiones como son de Bolivia y Perú respectivamente, ninguna de estas toma relación con el idioma quichua de nuestro país. En vista de esto y que en nuestro país se manejan ambas lenguas español y quichua, y las herramientas de apoyo que existen en el mercado, ninguna realiza traducciones básicas del idioma quichua de Imbabura, es por esto que se ha considerado la realización de esta herramienta de apoyo para facilitar la traducción del idioma quichua tomando como referencia a la provincia de Imbabura.

ALCANCE

Desarrollar una aplicación para dispositivos móviles como tabletas y Smartphone que utilicen el sistema operativo Android, la aplicación se instalará en el dispositivo móvil el cual dará la facilidad de traducir palabras del español al quichua (Imbabura) de forma textual de manera fácil y rápida para instituciones educativas primarias en la provincia de Imbabura, estos por ser la provincia que más empeño y recursos está empleando en la difusión del idioma quichua, o también como herramienta de apoyo para usuarios que estén interesados en el aprendizaje del idioma.

La aplicación tendrá la facilidad de ingresar nuevas palabras como sean necesarias por parte del usuario, quien será el responsable que estas sean ingresadas en el idioma quichua, ya que no es posible controlar directamente la aplicación en el dispositivo móvil ya que trabaja sin conexión a Internet.

OBJETIVOS DEL TRABAJO DE TITULACIÓN

Objetivo General

- Desarrollar una aplicación que traduzca palabras del español al idioma Quichua (Imbabura) para dispositivos móviles (tabletas y Smartphones) con sistema operativo Android como herramienta de apoyo básico para personas interesadas en el idioma.

Objetivos Específicos

- Determinar las características básicas de la lingüística Quichua que se necesitará para el desarrollo de la aplicación.
- Determinar los requerimientos mínimos del Sistema Operativo (Android) para el desarrollo de la aplicación.
- Identificar las características mínimas del hardware necesario para la instalación de la aplicación.
- Aplicar el lenguaje de programación JAVA y la metodología de construcción de la aplicación.
- Diseñar la aplicación que traduzca palabras de español a quichua para sistemas operativos Android.

CAPÍTULO I

1 FUNDAMENTACIÓN TEÓRICA.

1.1 Lingüística del idioma Quichua

En referencia al documento publicado por Lipski, desde la formación de las llamadas interlenguas que se originaron en el siglo XVI y que se expandieron desde Ecuador hasta Argentina. El sistema vocálico del idioma quichua esta conformado por tres vocales que son a – i – u. Los quichua hablantes que no dominan el español suelen mezclar los idiomas “(mesa de madera > misa di madira, usted > ostí)”. En la provincia de Imbabura todavía se encuentran muchos hablantes de quichua que han aprendido el castellano después de la adolescencia, que hablan más quichua que castellano en su vida cotidiana y cuyos contactos con la lengua española se limitan a las comunidades rurales donde viven. (Lipski, 2012, págs. 1, 7)

La mayoría de pobladores de esta región mantienen el quichua como primera lengua pero los más pequeños ya no continúan con el aprendizaje del idioma sino que adoptan el español como primera lengua (Lipski, 2012, pág. 8)

De acuerdo al estudio realizado por Lipski se tiene:

Los participantes que reunían estos criterios no fueron escogidos de acuerdo a su pronunciación sino por el empleo de configuraciones morfosintácticas propias del hablante bilingüe que no alcanza un dominio completo del castellano:

- Orden de palabras OBJETO-VERBO en frases no topicalizadas: maicito tengo; dos hermana tenía.
- Empleo de la partícula topicalizadora –ca del quechua: yo-ca dos hijos tengo; nosotrosca no podemos guardar.
- Uso de –ta(n) < también como marcador de discurso: mi papá-tan joven murió; mi mamita mi papá-tan no tenía.
- Uso del gerundio en vez de un verbo conjugado en cláusulas subordinadas: papa-ca gusano hinchando mismo ya acabó; enojando no sabía comer.
- Eliminación de artículos definidos: lobo come borrego; en páramo anda mi cuñada.

- Errores de concordancia sujeto-verbo y adjetivo-sustantivo: mis huahua[s] ya no borda[n]; ella-tan solito [solita] quedando.

1.2 Requerimientos mínimos del Sistema Android para el desarrollo de la aplicación.

En referencia a la publicación realizada por la página web unocero en 2013; la distribución del sistema operativo Android Gingerbread (2.3) es la que mayor presencia comercial tiene en dispositivos móviles como son Smartphone y tablets. Esta versión es una de las más populares ya que se mantiene en segundo lugar con el 33% de uso entre estos dispositivos, muchos de ellos se actualizaron a versiones más recientes a pesar de esto hay una gran cantidad de dispositivos que no pudieron actualizar por limitaciones técnicas. (unocero, 2013)

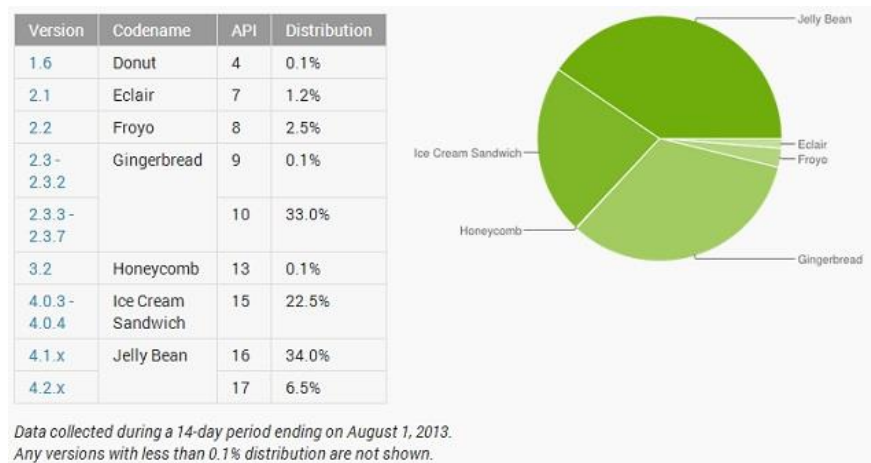


Gráfico 1-1 Estadísticas uso de Android
Fuente: (AndroidExperto, 2013)

De esta manera tomamos como base la versión de Android Gingerbread 2.3 por ser una de las primeras versiones que aún se encuentran en el mercado en dispositivos móviles (Smartphone y tablets). A continuación se describe las principales características de las diferentes versiones de Android a partir de la versión 2.3 Gingerbread hasta la versión 4.4 kitkat que son las versiones que utilizaremos para el desarrollo y funcionamiento de nuestra aplicación.

Características de las diferentes versiones de Android

| Versión Android | Características |
|-----------------------------------|---|
| 2.3 Gingerbread | <p>Fue Lanzado el 6 de Diciembre del 2010 y contenía:</p> <ul style="list-style-type: none"> ▪ Soporte para dispositivos móviles. ▪ Actualización del diseño de la interfaz del usuario. ▪ Soporte para pantallas extra grandes y resoluciones mayores. ▪ Soporte mejorado para desarrollo de código nativo. ▪ Recolección de elementos concurrentes para un mayor rendimiento. ▪ Un administrador de descargas: para descargar archivos grandes. ▪ Soporte nativo para múltiples cámaras. <p>Hubo mejoras al sistema y arreglos en fallas del 2.2.3 al 2.3.7.</p> |
| 3.0 / 3.1 / 3.2 Android HoneyComb | <p>Fue Lanzado el 22 de Febrero del 2011 Sus características son:</p> <ul style="list-style-type: none"> ▪ Mejor Soporte para Tablets. ▪ Escritorio 3D con Widgets rediseñados. ▪ Mejoras en el navegador Web. ▪ Sistema Multitarea mejorado. ▪ Soporte para videochat mediante Google Talk. ▪ Añade soporte para una gran variedad de periféricos y accesorios con conexión USB, teclados, ratones, hubs, dispositivos de juego y cámaras digitales. |
| 4.0 Android Ice Cream Sandwich | <p>Lanzado el 19 de Octubre del 2011. Es una versión que unifica el uso de cualquier dispositivo tanto en teléfonos, tablets, televisores, netbooks, etc. Incluye:</p> <ul style="list-style-type: none"> ▪ Interfaz limpia y moderna. ▪ Multitarea mejorada. ▪ Gestor de tráfico de datos de internet. ▪ Corrector de texto rediseñado y mejorado. ▪ Captura de pantalla en dos botones. ▪ Reconocimiento de voz del usuario. |

Tabla 1-1 Características de las versiones Android (Cont.)

Fuente: (Nuñez, 2013)

| Versión Android | Características |
|----------------------------------|---|
| 4.1, 4.2, 4.3 Android Jelly Bean | <p>Lanzado el 9 de Julio del 2012, basado en el núcleo de Linux 3.0.31, Fue una actualización incremental con el enfoque primario de mejorar la funcionalidad y rendimiento de la interfaz del usuario.</p> <ul style="list-style-type: none"> ▪ Mejora la fluidez y de la estabilidad. ▪ Ajuste automático de Widgets cuando se añaden al escritorio. ▪ Dictado por voz mejorado. ▪ Notificaciones mejoradas, con acceso más rápido a más información. ▪ Google Chrome se convierte en el navegador por defecto de Android. ▪ Getual Mode para personas discapacitadas visualmente. ▪ Cambios en la interfaz como la nueva barra de búsquedas. |
| 4.4 Android KitKat | <ul style="list-style-type: none"> ▪ Corrección de la frecuencia de las conexiones de datos. ▪ Optimización del proceso mm-camera-daemon, que en la versión 4.4.2 provoca un uso de la CPU excesivo en algunas apps. ▪ Enfoque de la cámara mejorado en el modo convencional y con HDR. ▪ Varios parches que mejoran el funcionamiento del enlace Bluetooth. ▪ Corrección de los reinicios aleatorios. ▪ Reparación de los accesos directos eliminados después de la actualización de algunas apps. ▪ Depuración de la seguridad de las conexiones USB. ▪ Parche para el proceso de conexión automática a las redes WiFi. ▪ Mejora de varios parámetros de funcionamiento de la cámara. ▪ Parches para el calendario, la mensajería MMS, la gestión de contactos. ▪ Corrección de los bloqueos de la pantalla de activación. ▪ Parche para la gráfica de monitorización del uso de datos. |

Tabla 1-1 Características de las versiones de Android
Fuente: (Nuñez, 2013)

Evolución de las versiones del sistema operativo Android



Gráfico 1-2 Versiones Android

Fuente: (Estévez, 2015)

“Que Android se haya convertido en el sistema operativo preferido de usuarios y fabricantes, no es ninguna novedad. Esto es gracias a su flexibilidad y performance, pero además influye en ello la gran cantidad de aplicaciones que se ofrecen en las distintas tiendas de aplicaciones. Sin embargo, el SO de Google es afectado por un problema bastante importante, que es el de la fragmentación, es decir la tendencia a ramificarse en muchas versiones y modificaciones diferentes. Esto es herencia del software libre, y en el ecosistema Android se vuelve mucho más importante ya que al existir en el mercado tantos Smartphone y tablets con resoluciones de pantalla, procesadores y características disímiles, desarrollar aplicaciones que se adapten y ejecuten en cada modificación de este SO para móviles es una tarea difícil y que rara vez se cumple”. Android se ha convertido en el sistema operativo de mayor aceptación en el uso de dispositivos móviles, esto se debe a la gran cantidad de aplicaciones que se ofrecen en las diferentes tiendas sin embargo este sistema operativo presenta un problema importante el cual es la fragmentación es decir la ramificación del sistema que es un problema de los sistemas abiertos. (Androidexperto.com, 2013)

En base a la publicación realizada por Androidexperto en el año 2013 se indica lo siguiente: Los desarrolladores de Android han modificado la licencia del kit de desarrollo para poder escribir aplicaciones, estas deberán cumplir una serie de pautas para evitar la fragmentación. (Androidexperto.com, 2013)

“La fragmentación del producto se puede ver fácilmente cuando nos interesamos por adquirir un dispositivo con Android instalado. A pesar de que la última versión del SO ya está disponible hace mucho tiempo en el mercado, todavía se siguen fabricando Smartphone y tablets que incorporan versiones más antiguas de Android”. En las

diferentes versiones del sistema operativo Android se puede apreciar la fragmentación del que se dispone. En el mercado aún se puede encontrar dispositivos móviles como Smartphone o Tablet que se comercializan con versiones anteriores de este sistema operativo. (Androidexperto.com, 2013)

A continuación se detallará las características de cada de una de las diferentes versiones para conocer el historial de este sistema operativo

1.3 Android 2.0/2.1 (Eclair), 2.2 (Frozen Yogurt) y 2.3 (Gingerbread)

“Si bien esta versión ya ha sido superada en cuanto a capacidad, características y performance, todavía se puede encontrar ampliamente instalada en muchos Smartphone y tablets, aún modelos de alta gama y con precios elevadísimos”. Las primeras versiones que salieron al mercado fueron Eclair, Frozen yogurt, Gingerbread esta última es una de las versiones que mayor aceptación tuvo en el mercado y actualmente aún se encuentra en varios modelos de dispositivos móviles (Androidexperto.com, 2013)

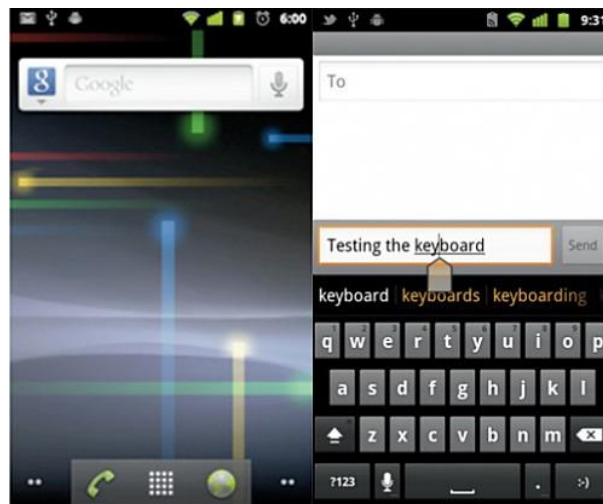


Gráfico 1-3 Pantalla Android 2.3
Fuente: (AndroidExperto, 2013)

“La versión 2.3 es una de las más extendidas entre los dispositivos de la actualidad, y representó un salto muy importante con respecto a las anteriores versiones, Android 2, 2.1 y 2.2”. La versión Gingerbread 2.3 en relación a las versiones anteriores marcó la diferencia en relación a las versiones anteriores y es una de las mejores por lo que aún en la actualidad se la sigue utilizando en varios modelos de dispositivo móviles. (Androidexperto.com, 2013)



Gráfico 1-4 Pantalla Android 2.3
Fuente: (AndroidExperto, 2013)

“Esta versión de Android incorpora importantes ventajas como el soporte para Adobe Flash, función de acceso Wi-Fi, soporte nativo para VoIP, nuevo gestor de descargas y muchas otras funcionalidades con las cuales competir con los principales proveedores de sistemas operativos móviles como Apple y Symbian”. Una de las principales características de la versión Gingerbread 2.3 es la incorporación de varios soportes dentro del sistema operativo como las mencionadas en el artículo de referencia, esto ha hecho que esta versión le permita competir con otros sistemas operativos de alto nivel. (Androidexperto.com, 2013)



Gráfico 1-5 Pantalla Android 2.3
Fuente: (AndroidExperto, 2013)

1.4 Android 3.0 (Honeycomb)

“Más conocida por su nombre código, Honeycomb, esta es la versión 3 de Android, la cual tiene una particularidad y es la de no poseer una versión para Smartphone, ya que sólo se encuentra disponible para tablets. Si estamos buscando una Tablet, entonces la mejor opción disponible es que la misma cuente con Android 3 Honeycomb”. La versión Honeycomb 3.0 se la utilizó solo para dispositivos móviles específicamente en tablets. (Androidexperto.com, 2013)



Gráfico 1-6 Pantalla Android 3.0
Fuente: (AndroidExperto, 2013)

1.5 Android 4.0 (Ice Cream Sandwich), 4.1/4.2 (Jelly Bean) y 4.3 (Key Lime Pie)

“Esta es en la actualidad la versión más nueva de Android, donde gracias a las profundas modificaciones realizadas a la interfaz, se obtienen múltiples ventajas relacionadas con la usabilidad del dispositivo que lo contiene. Algunos de los cambios más importantes son la nueva tipografía llamada Roboto, la cual es mucho más legible y aprovecha mejor la alta resolución de las pantallas modernas. También han sufrido cambios la barra de sistema y el sistema de notificaciones”. La versión 4.xx de Android es una de las más recientes liberadas para usuarios, las modificaciones en la interfaz como es la tipografía, esta característica aprovecha la resolución de la pantalla, además de mejorar varios aspectos de notificaciones. (Androidexperto.com, 2013)



Gráfico 1-7 Pantalla Android 4.0
Fuente: (AndroidExperto, 2013)

“Otro de los cambios más significativos de esta versión es la posibilidad de añadir más de un perfil de usuario, lo que significa que podrán personalizar y gestionar documentos y configuraciones múltiples usuarios”. Uno de los nuevos aspectos que presenta esta versión de Android es la de agregar y configurar varios usuarios personalizados, esta característica en versiones anteriores no era posible realizarlo. (Androidexperto.com, 2013)

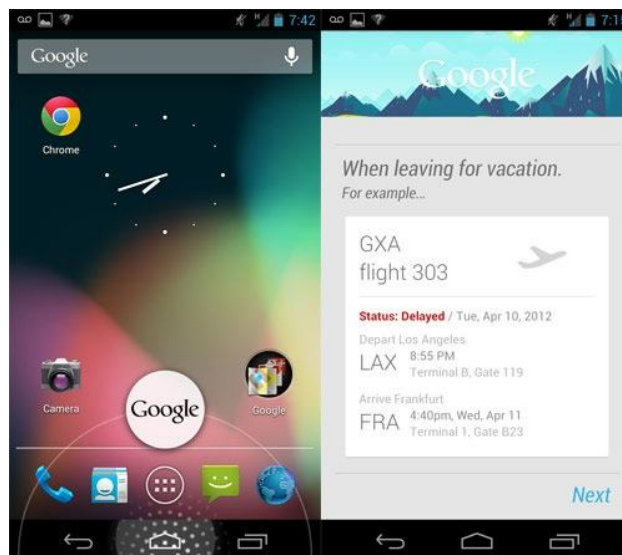


Gráfico 1-8 Pantalla Android 4.2
Fuente: (AndroidExperto, 2013)

En base a esta información la aplicación a desarrollar será funcional para las diferentes versiones del sistema operativo Android desde la 2.3 (GingerBread), por ser

la versión que más dispositivos móviles la utilizan, hasta la versión 4.4 (Kitkat) última versión con la cual se realizaron las pruebas de la aplicación y elaboración del presente documento.

1.6 Requerimientos mínimos de hardware para que se pueda ejecutar esta nueva versión de Android

En referencia al artículo publicado por ENTER.CO en 2011, el sistema operativo Android no tiene requerimientos mínimos por lo que el sistema puede ser instalado en dispositivos móviles de cualquier característica. Tal es así que los equipos que utilizan las versiones Gingerbread 2.3 y Froyo 2.2 pueden ser actualizados a la versión más reciente como Ice Cream Sandwich esto se debe a que el hardware de las versiones anteriores pueden soportar sin problema las actualizaciones para versiones recientes de sistema operativo Android, las actualizaciones de versión más dependen del fabricante del dispositivo ya que son ellos los que deciden hasta que versión de sistema operativo soporta cada uno de los modelos fabricados (ELNUEVODIA.COM, 2011)

“Si un usuario tiene un dispositivo viejo, ¿qué tan fácil será la actualización, y qué tan pronto podemos esperarla?: Hugo Barra Director de Producto Android: De nuevo, es algo que depende más de cada fabricante. Ice Cream Sandwich (versión 4.0) está diseñado para minimizar la fragmentación entre dispositivos, y por eso funciona tanto para tablets como Smartphone. Pero sí hay compatibilidad con dispositivos viejos”. Según el director de producto Android las versiones más recientes están diseñadas para minimizar la fragmentación entre dispositivos es por esta razón que puede funcionar tanto en Smartphone como en tablets (ELNUEVODIA.COM, 2011)

“Los dispositivos en los que Google controla el software, como el Nexus S o el Motorola Xoom, tendrán la actualización automática en pocas semanas. Otros, como el Galaxy S II, son productos de Samsung y otros fabricantes, y ellos controlan el software, entonces la velocidad de actualización depende más de ellos”. Las actualizaciones de versión del sistema operativo Android dependen de cada fabricante ya que son ellos los que controlan la liberación del software para cada modelo, con esto limitan a que un equipo con características anteriores pueda actualizarse a las últimas versiones del sistema operativo. (ELNUEVODIA.COM, 2011)

De los dispositivos móviles existentes en el mercado se realizó la búsqueda del equipo con las características más básicas y que esté funcionando el sistema operativo Android 2.3, el equipo que se encontró presenta las siguientes características.

Procesador: Qualcomm 600 mHz

RAM: 256 MB

Pantalla: 3.5 pulgadas 320X480

Memoria interna: 512 MB

Tarjeta de memoria: micro SD hasta 32 GB

Con esta información consideramos que el hardware necesario para nuestra aplicación no tiene requerimientos mínimos para su funcionamiento, y que se lo puede utilizar en cualquier dispositivo móvil Android desde la versión Gingerbread 2.3 en adelante.

1.7 Diferencias entre versiones de sistema Android

De acuerdo a varias fuentes consultadas se logró recopilar las siguientes diferencias entre las varias versiones de sistema operativo:

Según el artículo publicado en la página ELNUEVODIA.COM en el año 2011. Una de las virtudes del sistema operativo es que todo es más fácil al momento de utilizarlo ya que su operatividad permite ejecutar más procesos con menos pasos (ELNUEVODIA.COM, 2011)

Haciendo referencia al hardware utilizado por las versiones Android Kitkat 4.4 y la versión Gingerbread 2.3 es que ambas pueden correr en dispositivos móviles de las mismas características "Entre las características se destaca un procesador de 1GHz, es genial que esté presente pero eso no es un requerimiento que se debe cumplir en los fabricantes". En varios fabricantes dan como característica mínima que el hardware del dispositivo móvil debe tener un procesador de 1Ghz (Taringa.net, 2011)

En referencia a un artículo realizado por Taringa en 2011: En la que indica que las marcas más populares del mercado son las que liberan el software para actualizar las versiones de Android a una superior en cada uno de los modelos, en esto cada fabricante es el que considera las características del hardware necesario sin embargo los representantes de Android Open Source pronunciaron que las diferentes versiones del sistema operativo no requieren de requisitos mínimos de hardware y que cualquier

dispositivo móvil puede soportar las diferentes versiones del sistema operativo. (Taringa.net, 2011)

1.8 Java

“Android Development Tools (ADT) es un plugin para el IDE Eclipse que está diseñado para darle un ambiente potente, integrado en la construcción de aplicaciones para Android”. El IDE Eclipse en conjunto con ADT es la aplicación que nos facilitará con el desarrollo de la aplicación para el sistema operativo Android. (developer.android.com, 2013).

ADT amplía las capacidades de Eclipse que permiten configurar rápidamente nuevos proyectos para Android, crea una interfaz de usuario de aplicación, agrega los paquetes basados en la API Framework para Android, depura sus aplicaciones utilizando las herramientas del SDK de Android, e incluso exporta el certificado (o sin certificado) en archivos .apk con el fin de distribuir la aplicación. (developer.android.com, 2013).

1.8.1 Características de Java

Las características principales que nos ofrece Java son:

| Características | Descripción |
|--------------------|--|
| Lenguaje Simple | Se lo conoce como lenguaje simple porque viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de java por eso utiliza determinadas características de C++ y se han eliminado otras. |
| Orientado a Objeto | Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación. |
| Distribuido | Permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red. |
| Robusto | Es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. |

Tabla 1-2 Características de Java (Cont.)

Fuente: (sheyla88, 2009)

| Características | Descripción |
|-----------------------------------|--|
| Seguro | La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real. |
| Indiferente a la arquitectura | Java es compatible con los más variados entornos de red, cualesquiera sean estos desde Windows 95, Unix a Windows NT y Mac, para poder trabajar con diferentes sistemas operativos. Java es muy versátil ya que utiliza byte-codes que es un formato intermedio que sirve para transportar el código eficientemente o de diferentes plataformas (Hardware - Software). |
| Portable | Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java. |
| Interpretado y compilado a la vez | Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina. |
| Multihebra o Multihilos | Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que el programa tenga se ejecutarán en tiempo real muchas funciones al mismo tiempo. |
| Dinámico | El lenguaje java es muy dinámico en la fase de enlazado, sus clases solamente actuarán en medida en que sean requeridas o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red. |
| Produce Applets | En java se pueden crear aplicaciones independientes y applets. Independientes porque se pueden comportar como cualquier programa escrito en cualquier lenguaje. |
| Alto rendimiento | Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código. |

Tabla 1-2 Características de Java
Fuente: (sheyla88, 2009)

Se ha seleccionado la base de datos SQLite ya que está construido dentro del sistema Android, esto no significa que sea la única opción para el almacenamiento de información, se puede utilizar otros sistemas de base de datos como JavaDB o MongoDB pero se debe tomar en cuenta que se necesitará empaquetar las librerías necesarias con la aplicación y que no se puede contar con soporte por parte de Android para el desarrollo.

1.9 SQLite

SQLite es una base de datos de código abierto que no necesita ser instalado ya que solo se implementa una biblioteca en el software de programación. Su pequeño tamaño y simplicidad en el configuración es ideal para aplicaciones móviles (SQLite, 2013)

SQLite no necesita ser instalado sino que se agrega la referencia de la base de datos en el API. No existe un procedimiento para la configuración de la base de datos No tiene un servicio que se necesite iniciar, detener o configurar. No posee un administrador para la creación de nuevas instancias en la base de datos, o asignación de permisos de acceso a los usuarios. No se debe realizar ningún procedimiento para ejecutar el sistema de base de datos SQLite. No es necesario realizar algún procedimiento para recuperar el sistema después de una caída o fallo de alimentación. “No hay nada que solucionar”. (SQLite, 2013)

“La mayoría de los motores de base de datos SQL se implementan como un proceso de servidor independiente. Los programas que quieran acceder a la base de datos se comunican con el servidor utilizando algún tipo de comunicación entre procesos normalmente TCP / IP para enviar peticiones al servidor y recibir de vuelta los resultados”. Con SQLite, el proceso que desea tener acceso a la base de datos lee y escribe directamente al archivo. Es decir no existe un servidor al cual deba solicitar peticiones, evitando así los procesos de comunicación. (SQLite, 2013)

“La principal ventaja es que no hay un proceso server independiente para instalar, configurar, inicializar, administrar y solucionar problemas”. Esta es una razón por la que SQLite es un motor de base de datos de configuración cero. Las aplicaciones que utilizan SQLite no necesitan del apoyo de una consola de administración para la creación de una nueva base de datos. Cualquier aplicación que pueda acceder al disco es capaz de utilizar una base de datos SQLite. (SQLite, 2013)

“Por otra parte, un motor de base de datos que utiliza un servidor puede proporcionar una mejor protección contra errores en la aplicación cliente, el mismo que no puede alterar la memoria en el servidor”. Esto se debe a que en un servidor es un solo proceso persistente, que es capaz de acceder a la base de datos de control con mayor precisión. (SQLite, 2013)

De los motores de base de datos SQL que son sin servidor. SQLite es el único que se sabe que permite que múltiples aplicaciones accedan a la misma base de datos al mismo tiempo. (SQLite, 2013)

“Una base de datos SQLite es un solo archivo de disco normal que puede ser ubicado en cualquier lugar de la jerarquía de directorios”. SQLite puede leer el archivo de disco, entonces puede leer cualquier cosa en la base de datos. SQLite puede cambiar cualquier cosa en la base de datos. Los archivos de la base de datos pueden ser copiados fácilmente en una memoria USB o ser compartido por correo electrónico. (SQLite, 2013)

“Otros motores de base de datos SQL tienden a almacenar datos como una gran colección de archivos. A menudo, estos archivos se encuentran en una ubicación estándar que sólo el motor de base de datos en sí puede acceder”. Esto hace que los datos estén más seguros, pero también hace que sea más difícil el acceso (SQLite, 2013)

“El formato de archivo de SQLite es multiplataforma. Un archivo de base de datos escrito en una máquina puede ser copiado y utilizado en una máquina diferente, con una arquitectura diferente. Big-endian o little-endian de 32-bit o 64-bit eso no importa”. La arquitectura del computador que utilice SQLite no es importante ya que el formato de archivo es multiplataforma y corre prácticamente en todas las máquinas ya que utiliza el mismo formato de archivo. Mantener el formato de archivo estable y compatible con versiones anteriores es uno de los propósitos de los desarrolladores de SQLite, por lo que las nuevas versiones deben leer y escribir archivos de bases de datos de versiones anteriores. (SQLite, 2013)

“La mayoría de los otros motores de bases de datos SQL requieren respaldar y restaurar la base de datos cuando se pasa de una plataforma a otra y muchas veces cuando se actualiza a una nueva versión del software”. Los motores de bases de datos conocidos comercialmente y de licencia gratuita tienen el proceso para respaldo y

restauración de la información este procedimiento se lo ejecuta ya sea para el traslado entre plataformas o en cambios de versión. (SQLite, 2013)

“Cuando se optimiza el tamaño, toda la biblioteca de SQLite con todo lo permitido es menor de 500KiB en tamaño (medido en una ix86 con la utilidad "tamaño" de la suite de compiladores de GNU) características innecesarias se pueden deshabilitar en tiempo de compilación para reducir aún más el tamaño de la biblioteca a bajo de los 300KiB si se desea.” El tamaño que genera el archivo de base de datos es muy pequeño lo cual le hace muy apreciable para el desarrollo de aplicaciones móviles (SQLite, 2013)

"La mayoría de los otros motores de bases de datos SQL asignan una cantidad fija de espacio en disco para cada fila en la mayoría de las tablas". El motor de base de datos asignará la cantidad de espacio en el disco duro, independientemente de la cantidad de información que se almacene. (SQLite, 2013)

“SQLite, por el contrario, utiliza sólo la cantidad de espacio en disco que realmente se necesita para almacenar la información en una fila”. Si la información que se almacena consta de un solo carácter en una columna VARCHAR (100), sólo se utilizará un byte de espacio en el disco duro. (SQLite, 2013)

“El uso de registros de longitud variable por SQLite tiene una serie de ventajas. Es el resultado de los archivos de base de datos más pequeños, obviamente. También hace que el funcionamiento de bases de datos sea más rápido”. SQLite tiene una ventaja ya que al tener un archivo de base de datos más pequeña la consulta de información es más rápida. (SQLite, 2013)

“Cada motor de base de datos SQL compila cada sentencia SQL en una especie de estructura de datos interna que luego se utiliza para llevar a cabo la obra de la declaración. Pero en la mayoría de los motores de SQL que la estructura de datos interna es una compleja red de estructuras y objetos relacionados entre sí”. En SQLite la compilación de sentencias SQL es un procedimiento en lenguaje de máquina que los usuarios pueden visualizar el lenguaje a través de la palabra clave EXPLAIN al inicio de la consulta. (SQLite, 2013)

“El uso de una máquina virtual en SQLite ha sido un gran beneficio para el desarrollo de la biblioteca. La máquina virtual proporciona un cruce nítido y bien definido entre el

front-end de SQLite (la parte que analiza las sentencias SQL y genera código de máquina virtual) y el back-end (la parte que ejecuta el código de la máquina virtual y calcula un resultado.)”. La máquina virtual permite a los desarrolladores ver con claridad y de forma legible lo que SQLite está realizando con cada declaración que compila, lo cual es una gran ayuda en la depuración. SQLite también tiene la capacidad de rastrear la ejecución de la máquina virtual, y la impresión de cada resultado conforme se ejecuta. (SQLite, 2013)

“El código fuente para SQLite es de dominio público. Ninguna reclamación de derechos de autor se hace en el código fuente del núcleo. (La documentación y el código de prueba es una cuestión diferente algunas secciones de la documentación y la lógica de la prueba se rigen por las licencias de código abierto.)”. Los colaboradores del software de SQLite han firmado declaraciones juradas. Esto significa que cualquier persona es capaz de hacer legalmente lo que quieran con el código fuente de SQLite. (SQLite, 2013)

"Hay otros motores de base de datos SQL con licencias liberales que permiten que el código sea más amplio y libremente utilizado. Pero esos otros motores todavía se rigen por la ley de derechos de autor”. En el mercado hay motores de bases de datos que aún se rigen por la ley de derechos de autor en ese aspecto SQLite es lo contrario, ya los derechos de autor, simplemente no se aplica. (SQLite, 2013)

“SQLite ofrece una serie de mejoras en el lenguaje SQL que no se encuentra normalmente en otros motores de bases de datos. La palabra clave EXPLAIN y escribiendo manifest...SQLite también proporciona declaraciones como REPLACE y ON CONFLICT, cláusula que permite un mayor control sobre la resolución de conflictos de restricción. SQLite soporta ATTACH y DETACH comandos que permiten a múltiples bases de datos independientes que deben utilizarse juntos en la misma consulta”. SQLite tiene características que otras bases de datos no disponen lo que ayuda a visualizar de mejor forma los resultados esto se define en las API lo que permite al usuario añadir nuevas funciones de SQL y secuencias de clasificación. (SQLite, 2013)

1.10 Justificación metodológica

En la búsqueda de una metodología que se adapte a las necesidades para el desarrollo de la aplicación, y en base a los puntos claves a los cuales hace referencia

la autora Lizbeth García en su blog lisvant.blogspot.com en 2013, se enumeran a continuación:

Metodologías de desarrollo de software

| Metodología | Características |
|-------------|--|
| RUP | <ul style="list-style-type: none"> • Pesado • Dividido en cuatro fases, que se dividen en iteraciones • Los artefactos son el objetivo de cada actividad • Se basa en roles • UML • Muy organizativo • Mucha documentación |
| XP | <ul style="list-style-type: none"> • Ligero • Cercano al desarrollo • Se basa en UserStories • Fuerte comunicación con el cliente • El código fuente pertenece a todos • Programación por parejas • Solo el mínimo de organización • Pobre en cuanto a documentación |
| 3P | <ul style="list-style-type: none"> • Ágil • Cercano al desarrollo, pero sin olvidar el diseño • Se basa en 3 principios: Personal, Problema, Proceso • Gran interacción con el cliente • Logra alcanzar un control y organización del proceso • Logra un equilibrio en cuanto a la generación de documentación |

Tabla 1-3 Metodología de desarrollo de software
Fuente: (García, 2013)

Ponderación de Metodologías

| | RUP | Calif | XP | Calif | 3P | Calif |
|----------------------|---|--------------|--|--------------|---|--------------|
| Procesamiento | Pesado | 3 | Ligero | 4 | Ágil | 5 |
| Desarrollo | Dividido en cuatro fases, que se dividen en iteraciones | 4 | Cercano al desarrollo | 4 | Cercano al desarrollo, pero sin olvidar el diseño. | 5 |
| Base | Se basa en roles | 3 | Se basa el UserStories | 4 | Se basa en tres principios: Personal, Problema, Proceso | 5 |
| Modelado | UML | 5 | Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación. | 4 | Modelo que describe el proceso de aprendizaje el funcionamiento de un aula en tres fases: presagio, proceso y producto. | 5 |
| Organización | Muy Organizativo | 5 | Solo el mínimo de organización | 3 | Logra alcanzar un control y organización del proceso | 4 |
| Programación | Desarrollo basado en componentes | 3 | Programación por parejas | 5 | Especificación Funcional: Se determinan las características de la solución y que es lo que hay que construir. | 5 |
| Documentación | Mucha Documentación | 5 | Pobre en cuanto a documentación | 4 | Logra un equilibrio en cuanto a la generación de documentación | 4 |
| Total | | 28 | | 28 | | 33 |

Tabla 1-4 Ponderación de las metodologías
Fuente: Investigador

Se ha seleccionado la metodología 3P en virtud de que con esta se utiliza de mejor manera el tiempo y la comunicación entre el programador y el usuario, reduciendo la elaboración de la documentación, agilizando el desarrollo de la aplicación.

1.11 METODOLOGÍA 3P

“Paradigma 3P es un método de desarrollo de software nacido al calor de la experiencia acumulada del grupo de investigación y desarrollo Atis debido a la insuficiente capacidad de respuesta a los clientes utilizando las metodologías tradicionales”. (Garcia, 2013)

Principios que sustentan el modelo

1. Los individuos y sus interacciones son más importantes que los procesos y las herramientas: El PERSONAL.
2. La comunicación con el cliente evita construir una elegante solución para un problema equivocado: El PROBLEMA.
3. El software que funciona es más importante que la documentación exhaustiva. El PROCESO. (Garcia, 2013)

| Ventajas | Desventajas |
|---|--|
| Se trabaja directa y constantemente con el cliente para evitar errores en las necesidades del sistema. | Si una fase falla hay que regresar al punto del principio. |
| Presenta mecanismos suficientes para enfrentar cualquier cambio que se presente durante el periodo de desarrollo. | Genera mucho tiempo en el desarrollo. |
| Los procesos son mucho más controlados a través de la gestión de proyecto. | Requiere de mucha experiencia para la identificación de riesgos. |
| El modelo de gestión Paradigma 3P contribuye a materializar un proceso de enseñanza aprendizaje con la optimización de recursos humanos y no humanos. | |

Tabla 1-5 Ventajas y desventajas metodología 3P
Fuente: (Delgado, 2008)

Fases

- Definición del problema.
- Identificación de los procesos unitarios.
- Diseño del prototipo.
- Desarrollo del prototipo.
- Prueba del prototipo.
- Si <no prueba de Prototipo>ir a Paso 3.
- Si <Prototipo difiere Sistema Deseado>ir a Paso 2.
- Si <no Necesidades satisfechas>ir a Paso 2.
- Implantación.
- Mantenimiento

Actividades básicas

- Codificar.
- Probar.
- Comunicar
- Idear
- Escuchar.
- Diseñar.

La metodología seleccionada para el desarrollo de la aplicación “Traductor de palabras de Español a Quichua (Imbabura) para sistemas Android.”, es conveniente ya que habrá el contacto directo entre el usuario y el programador, ya que es una aplicación para uso general, es decir que cualquier persona pueda utilizarlo, se lo desarrollará con un diseño amigable y fácil de entender.

En base a esta metodología se realizará el desarrollo de la aplicación.

CAPÍTULO II

2 DIAGNÓSTICO

2.1 Descripción del problema.

En vista de que en nuestro país se manejan ambas lenguas como son el Español y el Quichua, y las aplicaciones de ayuda se manejan a través de navegadores (browser) para realizar traducciones básicas hacia el idioma Quichua (Imbabura), estas aplicaciones se pueden utilizar solo con conexiones a internet lo que al perder cobertura de datos imposibilitan su funcionamiento Y las aplicaciones existentes no cubren la traducción del idioma Quichua de la provincia de Imbabura , considerando que cada región incluso de un mismo país difiere al de otro, dejando al usuario sin la herramienta de apoyo.

2.2 Identificación de los procesos unitarios.

Para el desarrollo del “Traductor de palabras de Español a Quichua (Imbabura) para sistemas Android.” Se han identificado los siguientes procesos unitarios.

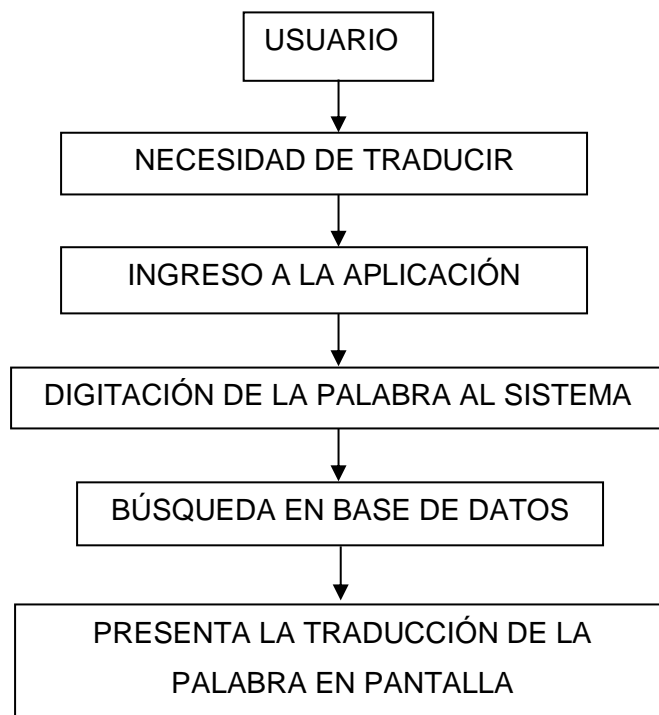


Gráfico 2-1 Modelo Base
Fuente: Investigador

USUARIO: Persona que tendrá instalada la aplicación en su dispositivo móvil sea este Smartphone o Tablet con sistema operativo Android mínimo versión 2.3. Gingerbread.

NECESIDAD DE TRADUCIR: Se presenta palabras en español que necesitan ser traducidas al quichua (Imbabura) para el entendimiento del mismo.

INGRESO A LA APLICACIÓN: El usuario ingresará a la aplicación seleccionando el ícono correspondiente a la aplicación previa su instalación en el dispositivo móvil.

DIGITACIÓN DE LA PALABRA AL SISTEMA: el usuario será el encargado de digitar las palabras que necesite al sistema para la traducción del español al quichua (Imbabura).

BÚSQUEDA EN BASE DE DATOS: Después de ingresada la palabra por el usuario el sistema realizará la búsqueda de la traducción en la base de datos

PRESENTA LA TRADUCCIÓN DE LA PALABRA EN PANTALLA: Realizada la búsqueda el sistema devolverá el resultado correspondiente a la búsqueda.

2.3 Modelo base

La aplicación requiere de un motor de base de datos para el desarrollo de la aplicación para esto se utilizará la base de datos SQLite para el almacenamiento de las palabras y donde se ingresarán las nuevas palabras que no constan en el diccionario inicial.

La estructura de la tabla a utilizar para la aplicación es la siguiente:

Modelo base de datos

| Descripción | Valor |
|---------------------|----------|
| Nombre de la tabla: | Palabra |
| Campo: | espaniol |
| Tipo de dato: | String |
| Campo: | quichua |
| Tipo de dato: | String |

Tabla 2-1 Modelo Base
Fuente: Investigador

La estructura de la tabla no requiere mayor complejidad ya que realizará la búsqueda de la palabra deseada. Y los campos al ser textos serán con tipo de dato String.

2.4 DISEÑO DE LA PROPUESTA.

Para el diseño de nuestra aplicación se tiene prevista de dos pantallas la primera pantalla de traducción con los siguientes campos:

Palabra a traducir: palabra que ingresara el usuario para la traducción respectiva CAMPO DE TEXTO AUTOCOMPLETE.

Palabra traducida: Resultado que devolverá la base datos después de la búsqueda ETIQUETA TEXTVIEW.



Gráfico 2-2 Pantalla de traducción de palabras
Fuente: Investigador

La segunda pantalla está prevista para el ingreso de nuevas palabras con su debida traducción.

Palabras en español: En este campo se ingresara el texto de la palabra en idioma español.

Palabras en quichua: En este campo se ingresara el texto de la palabra en idioma quichua correspondiente a la traducción de la palabra en español.

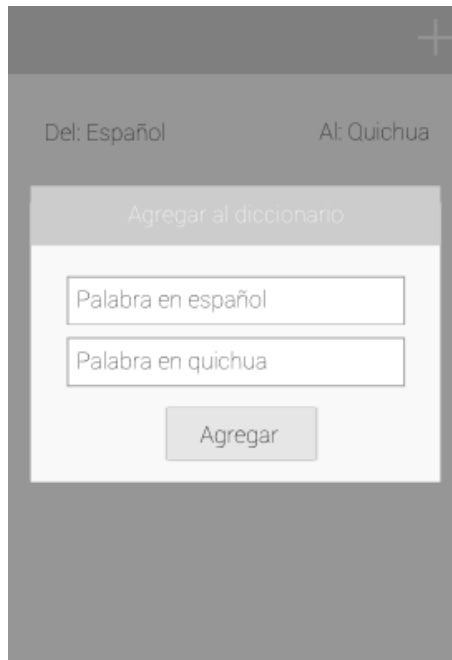


Gráfico 2-3 Pantalla de ingreso de palabras
Fuente: Investigador

2.5 Desarrollo del prototipo.

Analizado el problema que las aplicaciones existentes en el mercado las cuales necesitan de internet y deben ser ejecutadas a través de un navegador para poder ser utilizadas en el sistema operativo Android, ninguna hace referencia o apunta hacia el idioma Quichua de la provincia de Imbabura, la aplicación que desarrollaremos deberá ser autónoma para en caso de no tener conexión a internet se la pueda seguir utilizando para esto la aplicación deberá ser instalada en el dispositivo móvil.

El desarrollo de la aplicación se lo realizara en el lenguaje de programación Java por su independencia a los sistemas operativos.

Para la compatibilidad de JAVA con el sistema operativo Android se utilizara las herramientas ADT (Android Development Tools) con IDE Eclipse y base de datos SQLite.

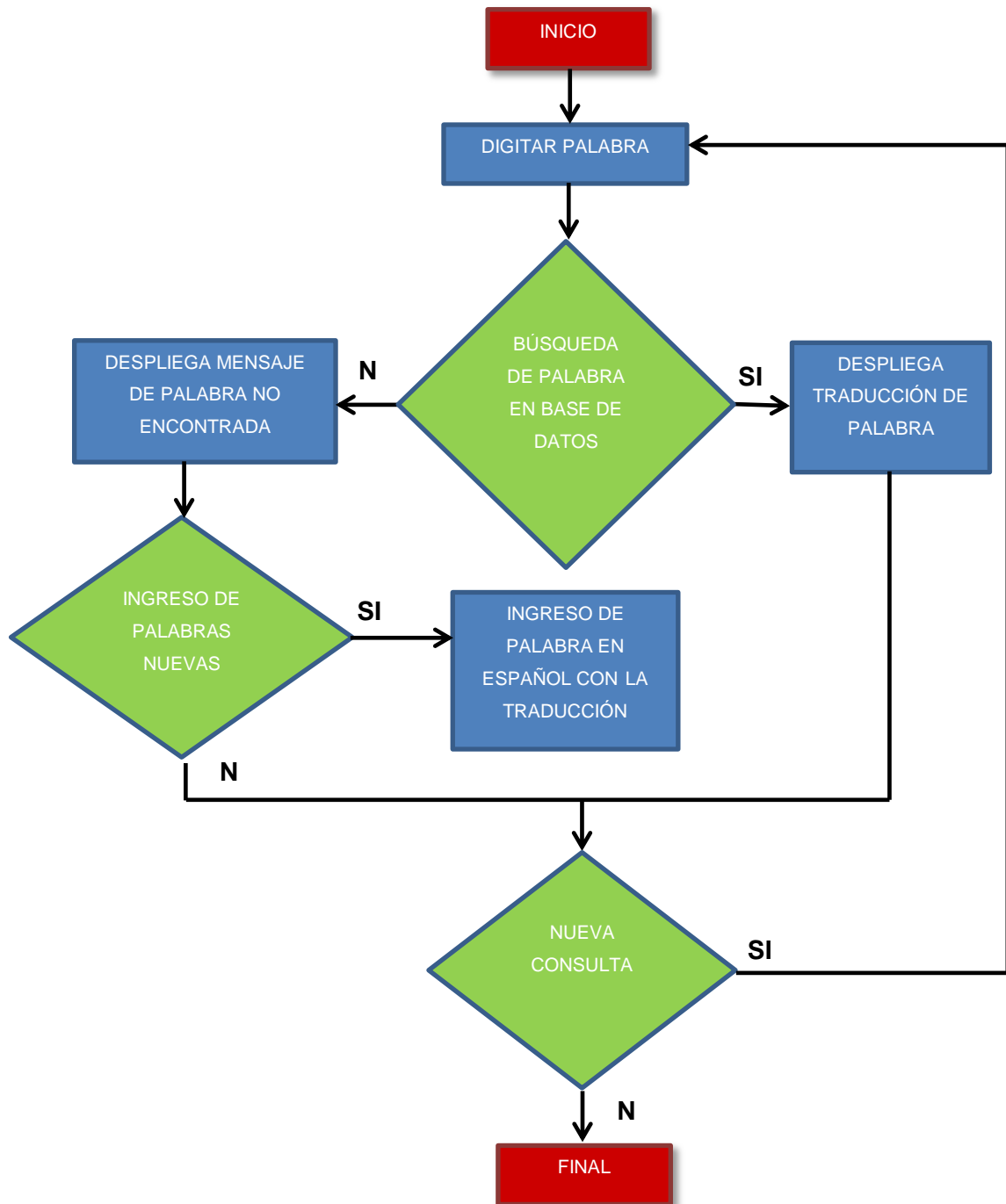


Gráfico 2-4 Diagrama de flujo de procesos
Fuente: Investigador

2.6 Presentación final de pantallas

2.6.1 Pantalla de traducción de palabras

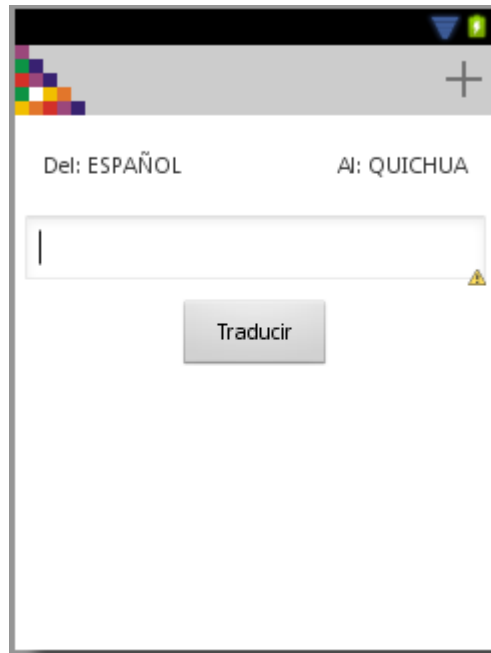


Gráfico 2-5 Pantalla final de traducción de palabras
Fuente: Investigador

2.6.2 Pantalla de ingreso de nuevas palabras

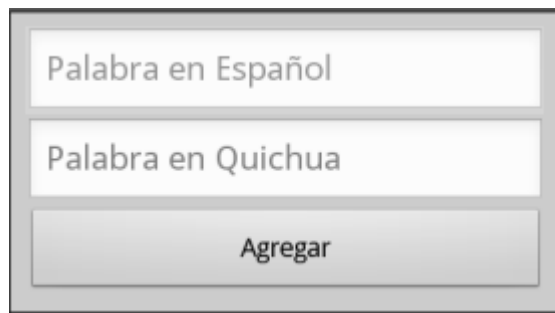


Gráfico 2-6 Pantalla final de ingreso de palabras
Fuente: Investigador

2.6.3 Pruebas de la aplicación

Para asegurar la estabilidad y la confianza del sistema así como realizar mejoras a las interfaces, se procedió a realizar pruebas que permitieran tener noción del desempeño del sistema ante usuarios normales (no por el desarrollador) en situaciones reales.

Las actividades que los individuos involucrados, en las pruebas realizadas fueron las siguientes:

- Probar el sistema
- Ingresar al sistema una vez desde cualquier dispositivo móvil y con cualquier versión igual o superior a la 2.3 Gingerbread
- Llenar un formato de prueba:

Formato de pruebas

| Fecha | Marca y modelo del Dispositivo móvil | Versión sistema Android | Comentario |
|-------|--------------------------------------|-------------------------|------------|
| | | | |

Tabla 2-2 Formato para pruebas de funcionamiento
Fuente: Investigador

Se le indicó al usuario la finalidad del llenado del formato y a lo que se refería cada campo a llenar, esto para evitar ambigüedades a la hora de llenar el formato. Con cada indicador en el formato arriba presentado se describe a continuación:

- **Fecha:** Fecha en la que se realiza las pruebas en diferentes días y horas
- **Dispositivo móvil:** Con este indicador nos permite conocer el dispositivo móvil que se utilizó para determinar las características del hardware para referencia
- **Versión sistema Android:** en este campo nos interesa conocer la versión del sistema operativo para referencia sobre la versión en la que se utilizó.
- **Comentarios:** Los comentarios sobre el sistema para mejoras en el mismo

2.6.4 Dispositivos de prueba

Para las pruebas del aplicativo se ha considerado las siguientes clases de equipos:

- Emuladores bajo sistema operativo Windows
- Smartphone
- Tablet

Resultado de pruebas realizadas

| Fecha | Marca y modelo del Dispositivo móvil | Versión sistema Android | Comentario |
|------------|--|-------------------------|--|
| 14/11/2013 | Emulador BlueStacks | 2.3 | Las pruebas realizadas son satisfactorias y se ejecuta sin ningún problema. |
| 01/12/2013 | Smartphone Samsung S5360 | 2.3.3 | Las pruebas realizadas son satisfactorias se ejecutaron sin ningún problema. |
| 01/12/2013 | Smartphone LG P705 | 4.0.3 | Las pruebas realizadas son satisfactorias y se ejecuta sin ningún problema. |
| 02/12/2013 | Tablet Samsung Galaxy 2 7" Smartphone Samsung GT-I9500 | 4.2.2 | Las pruebas realizadas son satisfactorias se ejecutaron sin ningún problema. |
| 29/04/2015 | Smartphone Samsung GT-I9500 | 4.4.2 | Las pruebas realizadas son satisfactorias y se ejecutaron sin ningún problema. |

Tabla 2-3 Resultado de pruebas realizadas
Fuente: Investigador

Cuadro estadístico de pruebas realizadas

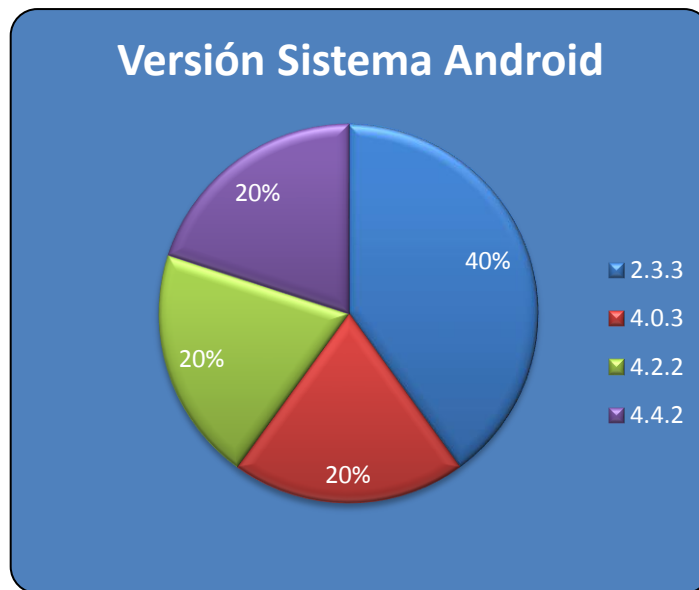


Gráfico 2-7 Cuadro estadístico de pruebas realizadas
Fuente: Investigador

2.7 Resultados obtenidos

Una vez realizadas las pruebas con varios dispositivos móviles de diferentes marcas, características y versiones del sistema operativo se pudo comprobar la compatibilidad en los mismos, por lo que la versión mínima del sistema operativo Android para ejecutar el aplicativo es la 2.3 y la versión más alta para ejecutar es la versión 4.4.2 existente en el mercado hasta el momento de las pruebas.

2.8 Conclusiones

- La lingüística del idioma quichua es muy diversa, dependiendo de la región o país donde se lo utilice, su dialéctica ha variado con el pasar de los años y la influencia con otros idiomas a pesar de tener el mismo origen. La provincia de Imbabura es la segunda en cantidad de quichua hablantes, esta provincia es una de las más visitadas por turistas tanto por sus paisajes como por las artesanías por lo que se ha considerado al dialecto de esta provincia como referente para nuestro desarrollo.
- El sistema operativo Android al tener una de las mayores distribuciones para dispositivos móviles en Smartphone y tablets en el mercado, y al considerarla como base para el desarrollo de nuestra aplicación, se determinó que la versión mínima para el desarrollo y ejecución de nuestra aplicación es la

versión 2.3 y como versión máxima la 4.4.2 de las cuales se comprobó su compatibilidad.

- El hardware utilizado en las pruebas con la versión mínima de Android 2.3 en las cuales se ejecutó nuestro aplicativo se ajusta con el modelo propuesto en este documento. Las pruebas realizadas sobre este dispositivo resultaron satisfactorias.
- El lenguaje de programación JAVA se ajustó a los requerimientos necesarios para el desarrollo de la aplicación ya que al ser un lenguaje de programación independiente, es compatible con cualquier sistema operativo. Siendo así que la aplicación es compatible con el sistema operativo Android en sus diferentes versiones.
- La metodología 3P (Personal, Problema, Proceso), dio la guía para que la aplicación se logre ejecutar adecuadamente esto debido a la directa comunicación entre el usuario y el programador reduciendo el tiempo de desarrollo de nuestra aplicación, entendiéndose de mejor manera el problema y así lograr cubrir las necesidades.
- Al finalizar el proyecto se obtiene una aplicación que realiza las traducciones básicas de palabras del idioma español al idioma quichua de la provincia de Imbabura satisfactoriamente para equipos móviles como Smartphone y tablets que tienen como sistema operativo Android desde la versión 2.3 hasta la versión 4.4.2.

2.9 Recomendaciones

- La realización de nuevos módulos o actualizaciones que mejoren las utilidades de la aplicación como la traducción inversa es decir del idioma quichua (Imbabura) al idioma español, considerando que hoy en día los dispositivos móviles están en uso de todo usuario.
- La difusión del aplicativo a través de los sitios web que están promoviendo la difusión del idioma quichua en la provincia de Imbabura para llegar a la mayor cantidad de usuarios que necesiten realizar traducciones.
- La realización de un Web Service de bloqueo para el ingreso de nuevas palabras, estas deben ser solo en idioma Español y Quichua para evitar que sean ingresadas en idiomas diferentes.

- El análisis de uso de la aplicación para ampliar la difusión a otros dispositivos móviles que utilicen sistemas operativos diferentes al sistema Android como son IOS de Mac, Windows Mobile y BlackBerry.
- La factibilidad de diseñar nuevas aplicaciones que sean basadas en los diferentes dialectos del idioma quichua, que contribuyan con la difusión del idioma de otras regiones.

BIBLIOGRAFÍA

4ndroid.com. (2010). Obtenido de <http://4ndroid.com/moviles-android/>

TechJava. (2010). Obtenido de Journal on Java Technology:
<http://www.techjava.de/topics/2010/02/launching-rcp-via-jws/>

Taringa.net. (31 de agosto de 2011). Obtenido de
<http://www.taringa.net/comunidades/soandroid/3639858/Noticia-Android-2-3-no-tiene-requerimientos-minimos.html>

AndroidExperto. (03 de 02 de 2013). Obtenido de
<http://www.androidexperto.com/aprender-android/versiones-android/>

Ecuador LLaktapak. (08 de 2013). Recuperado el 12 de mayo de 2015, de Kichwa Basico: <http://kichwa.net/wp-content/uploads/2013/08/dia-1-kichwa-basico-A1-Asamblea2.pdf>

Ecuador del sur turismo y cultura. (08 de mayo de 2014). Recuperado el 12 de 05 de 2015, de <http://ecuadordelsur.blogspot.com/2014/05/quechua-o-quichua-idioma-del-imperio.html>

Play Store de google. (2015). Recuperado el 12 de 05 de 2015, de <https://play.google.com/store/search?q=quichua&c=apps&docType=1&sp=CAFiCQoHcXVpY2h1YXoCGACKAQIIAQ%3D%3D%3AS%3AANO1ljLTdgc&hl=es>

7maccur7. (08 de diciembre de 2010). *Hipertextual*. Obtenido de
<http://www.celularis.com/software/android-2-3-requerimientos-procesadore/>

AFP. (2011). *La Hora*. Recuperado el 12 de 05 de 2015, de Noticias:
<http://www.lahora.com.ec/index.php/movil/noticia/1101448010>

Comunicación alternativa. (2012). *Globedia El diario colaborativo*. Recuperado el 17 de octubre de 2013, de <http://ec.globedia.com/promueven-revitalizar-kichwa-lengua-pueblo-otavalo>

D. Richard Hipp. (2015). Obtenido de <http://www.sqlite.org/index.html>

- Delgado, E. (2008). *Empresa de Proyectos de Arquitectura e Ingeniería*. Recuperado el 15 de noviembre de 2013, de <http://www.redalyc.org/pdf/1939/193915935003.pdf>
- developer.android.com. (2013). Recuperado el 2013, de (<http://developer.android.com/intl/es/tools/sdk/eclipse-adt.html>)
- (s.f.). *El quechua y sus hablantes: En la Pontificia Universidad Católica del Perú*.
- ENTER.CO. (2011). *Android es compatible con Smartphone viejos*. Obtenido de <http://www.elnuevodia.com/android4llegaraloscelularesmasviejos-1101119.html>
- Estévez, J. (2015). *El Grupo Informático*. Obtenido de <http://www.elgrupoinformatico.com/samsung-galaxy-comparativa-con-sus-antecesoros-t22500.html>
- Girones, J. T. (2012). *El gran libro de Android*. Barcelona-España: Alfaomega Grupo Editor.
- GÓMEZ, O. S. (2011). *Curso programación Android*. España: Creative Commons.
- Graciela. (2013). *Androidexperto*. Recuperado el 03 de noviembre de 2013, de <http://www.androidexperto.com/aprender-android/versiones-android/>
- kichwa.net. (2013). *Lingüística Quichua de Imbabura*. Recuperado el 30 de septiembre de 2013, de www.kichwa.net
- La hora. (24 de julio de 2008). ASAMBLEA NACIONAL. *El Kichwa y el Shuar son idiomas oficiales de relación intercultural*. Obtenido de http://www.lahora.com.ec/index.php/noticias/show/751124/-1/El_Kichwa_y_el_Shuar_son_idiomas_oficiales_de_relaci%C3%B3n_intercultural.html#.VVNHt9KG-So
- Milena, D. A., & Odalys, P. P. (2013). Pueblo Otavalo. En *ESTUDIO DE LOS CONOCIMIENTOS Y SABERES ANCESTRALES DE LOS PUEBLOS KICHWAS Y AFRODESCENDIENTES DE LA PROVINCIA DE IMBABURA* (págs. 58-59). Ibarra, Imbabura, Ecuador.
- MONTIEL, D. P. (2008). *INTRODUCCIÓN A SQLITE*. Creative Commons.

Revista Rikcharishun. (18 de julio de 2008). *Ecuadorunari*. Recuperado el 12 de 05 de 2015, de La Madre Tierra y sus hijos seguiremos hablando kichwa:
http://www.ecuarunari.org/es/noticias/no_20080718.htm

sheyla88. (septiembre de 2009). *Blogdiario.com*. Recuperado el 15 de noviembre de 2013, de <http://sheyla88.blogspot.es/1252973711/caracter-sticas-de-java/>

SQLite. (2013). *Características distintivas de SQLite*. Recuperado el 20 de noviembre de 2013, de <http://www.sqlite.org/different.html>

SQLite. (2013). *SQLite*. Recuperado el 9 de noviembre de 2014, de <http://www.sqlite.org/index.html>

unocero. (02 de agosto de 2013). *UNOCERO*. Recuperado el 31 de octubre de 2013, de <http://www.unocero.com/2013/08/02/android-jelly-bean-ahora-esta-en-40-de-dispositivos/>

Yamberla, M. V., Otavalo, J., Salazar, H., & Cajas, G. D. (2014). *WIÑAY KAWSAY*. Obtenido de http://www.radioiluman.ec/radio/?page_id=1193

ANEXO 1

Código Fuente de la aplicación.

MainActivity.java

```
package com.nel.diccionario;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.StringWriter;
import java.io.Writer;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager.NameNotFoundException;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import com.nel.diccionario.view.FromAutoComplete;
import com.nel.sql.DataBaseHelper;
import com.nel.sql.DataBaseHelperFactory;
import com.nel.sql.PalabraDAO;
import com.nel.sql.PalabraDAOFactory;
import com.nel.sql.PalabraDAOImp;

public class MainActivity extends Activity implements OnClickListener {

    private static final String FROM = "From";
    private static final String TRANSLATE = "Translate";
```

```

private boolean fromEsToQu = true;

private TextView fromTextView;
private TextView toTextView;
// private EditText palabraFromEditText;
private TextView palabraToTextView;
private Button translateButton;
private ImageButton addButton;
private ImageButton cambiarIdiomaButton;

private FromAutoComplete fromAutoCompleteTextView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    try {
        DataBaseHelperFactory.create(this, "DiccionarioDB");
        PalabraDAOFactory.getInstance().create();

        if (PalabraDAOFactory.getInstance().isEmpty()) {

            InputStream is;
            try {
                is = getResources().getAssets().open("query.sql");
                String sql = convertStreamToString(is);

                String[] queries = sql.split(";");
                for (String query : queries) {

                    DataBaseHelperFactory.getInstance().getDataBase().execSQL(query);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}

//      palabraFromEditText = (EditText)
findViewById(R.id.main_fromWord_editText);
        palabraToTextView = (TextView)
findViewById(R.id.main_toWord_TextView);
        translateButton = (Button) findViewById(R.id.main_translate_button);
        addButton = (ImageButton)
findViewById(R.id.toolbar_add_ImageButton);
        cambiarIdiomaButton = (ImageButton)
findViewById(R.id.main_change_button);
        fromTextView = (TextView) findViewById(R.id.fromTextView);
        toTextView = (TextView) findViewById(R.id.toTextView);

        fromAutoCompleteTextView =
(FromAutoComplete)findViewById(R.id.main_from_autoCompleteTextView);
        fromAutoCompleteTextView.setThreshold(3);
        fromAutoCompleteTextView.setFrom(fromEsToQu ?
PalabraDAOImp.ES : PalabraDAOImp.QU);

        translateButton.setOnClickListener(this);
        addButton.setOnClickListener(this);
        cambiarIdiomaButton.setOnClickListener(this);
    }

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putBoolean(FROM, fromEsToQu);
}

```



```

        outState.putString(TRANSLATE,
palabraToTextView.getText().toString());
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        this.fromEsToQu = savedInstanceState.getBoolean(FROM);

        this.palabraToTextView.setText(savedInstanceState.getString(TRANSLATE));
        this.setTranslateLanguajes();
    }

    @Override
    public void onClick(View v) {

        switch (v.getId()) {
        case R.id.toolbar_add_ImageButton:
            startActivity(new Intent(this, AddWordActivity.class));
            break;
        case R.id.main_translate_button:

//            String palabra = palabraFromEditText.getText().toString();
//            String palabra = fromAutoCompleteTextView.getText().toString();
//            String traduccion = "";
            try {
                if (fromEsToQu) {
                    traduccion =
PalabraDAOFactory.getInstance().find(palabra, PalabraDAOImp.ES,
PalabraDAOImp.QU);
                } else {
                    traduccion =
PalabraDAOFactory.getInstance().find(palabra, PalabraDAOImp.QU,
PalabraDAOImp.ES);
                }
            } catch (Exception e) {

```

```

        e.printStackTrace();
    }
    if (traduccion.equals("")) {
        traduccion = getString(R.string.noTraslate);
    }
    this.palabraToTextView.setText(traduccion);

    break;
case R.id.main_change_button:
    this.setTranslateLanguajes();
    break;
}
}

```

```

public static String convertStreamToString(InputStream is) throws
IOException {
    Writer writer = new StringWriter();
    char[] buffer = new char[2048];
    try {
        Reader reader = new BufferedReader(new
InputStreamReader(is, "UTF-8"));
        int n;
        while ((n = reader.read(buffer)) != -1) {
            writer.write(buffer, 0, n);
        }
    } finally {
        is.close();
    }
    String text = writer.toString();
    return text;
}

```

```

private void setTranslateLanguajes() {
    if (fromEsToQu) {
        this.fromEsToQu = false;
    }
}

```

```
        this.fromTextView.setText(getString(R.string.fromQuichua));
        this.toTextView.setText(getString(R.string.toEspaniol));
    } else {
        this.fromEsToQu = true;
        this.fromTextView.setText(getString(R.string.fromEspaniol));
        this.toTextView.setText(getString(R.string.toQuichua));
    }

    fromAutoCompleteTextView.setFrom(fromEsToQu ?
PalabraDAOImp.ES : PalabraDAOImp.QU);
}

}
```

AddWordActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.os.Message;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.nel.sql.PalabraDAO;
import com.nel.sql.PalabraDAOFactory;

public class AddWordActivity extends Activity implements OnClickListener{

    private EditText espaniolEditText;
    private EditText quichuaEditText;
    private Button agregarButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_word);

        espaniolEditText = (EditText)findViewById(R.id.add_espaniol);
        quichuaEditText = (EditText)findViewById(R.id.add_quichua);
        agregarButton = (Button)findViewById(R.id.add_button);
        agregarButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        String espaniol = espaniolEditText.getText().toString().replace(" ", "");
        String quichua = quichuaEditText.getText().toString().replace(" ", "");

        PalabraDAO palabraDAO = PalabraDAOFactory.getInstance();
```

```

    try {
        if (palabraDAO.remove(espaniol, quichua)){
            if (palabraDAO.save(espaniol, quichua)){
                Toast.makeText(this, getString(R.string.addOk),
Toast.LENGTH_SHORT).show();
                espaniolEditText.setText("");
                quichuaEditText.setText("");
            }
        }
        else{
            Toast.makeText(this, getString(R.string.addError),
Toast.LENGTH_SHORT).show();
        }
    } catch (Exception e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
}

```

FromAutoComplete.java

```
package com.nel.diccionario.view;
```

```
import java.util.ArrayList;
```

```
import android.content.Context;
```

```
import android.util.AttributeSet;
```

```
import android.widget.AdapterView;
```

```
import android.widget.AutoCompleteTextView;
```

```
import com.nel.sql.PalabraDAO;
```

```
import com.nel.sql.PalabraDAOFactory;
```

```
public class FromAutoComplete extends AutoCompleteTextView {
```

```
    private String from;
```

```
    private Context context;
```

```
    public String getFrom() {
```

```
        return from;
```

```
    }
```

```
    public void setFrom(String from) {
```

```
        this.from = from;
```

```
    }
```

```
    public FromAutoComplete(Context context) {
```

```
        super(context);
```

```
        this.context = context;
```

```
    }
```

```
    public FromAutoComplete(Context context, AttributeSet attrs) {
```

```
        super(context, attrs);
```

```
        this.context = context;
```

```
    }
```

```
    public FromAutoComplete(Context context, AttributeSet attrs, int defStyle) {
```

```
        super(context, attrs, defStyle);
```

```
        this.context = context;
```

```

    }

    @Override
    public boolean enoughToFilter() {
        if (getText().length() < 3)
            return false;
        else{
            PalabraDAO palabraDAO = PalabraDAOFactory.getInstance();
            try {
                ArrayList<String> result =
palabraDAO.findMatch(getText().toString(), from);
                ArrayAdapter<String> adapter = new
ArrayAdapter<String>(context, android.R.layout.simple_dropdown_item_1line, result);
                setAdapter(adapter);
            } catch (Exception e) {
                e.printStackTrace();
            }

            return true;
        }
    }
}
}
}

```

PalabraDAO.java

```
package com.nel.sql;
```

```
import java.util.ArrayList;
```

```
public interface PalabraDAO {
```

```
    /**
```

```
     * Find a configuration object on device.
```

```
     *
```

```
     * @return A configuration object.
```

```
     * @throws DataAccessException
```

```
    */
```

```
public String find(String palabra, String from, String to) throws Exception;
```

```
    /**
```

```
     * Save a configuration object on device.
```

```
     *
```

```
     * @param configuration
```

```
     *     A configuration object.
```

```
    */
```

```
public boolean save(String espaniol, String quichua) throws Exception ;
```

```
    /**
```

```
     * Remove a configuration object on device.
```

```
     * @return
```

```
     * @throws Exception
```

```
    */
```

```
public boolean remove(String espaniol, String quichua) throws Exception ;
```

```
    /**boolean
```

```
     * Eliminate the persistece of configuration object.
```

```
     * @throws Exception
```

```
    */
```

```
public void drop() throws Exception ;
```



```
public void create() throws Exception ;
```

```
public boolean isEmpty() throws Exception;
```

```
public ArrayList<String> findMatch(String startWith, String from) throws  
Exception;
```

```
}
```

PalabraDAOImp.java

```
package com.nel.sql;
```

```
import java.util.ArrayList;
```

```
import android.content.ContentValues;
```

```
import android.database.Cursor;
```

```
public class PalabraDAOImp implements PalabraDAO {
```

```
    private final static String TABLE_NAME = "Palabra";
```

```
    public final static String ES = "Español";
```

```
    public final static String QU = "Quichua";
```

```
    /**
```

```
     * Find a configuration object on device.
```

```
     *
```

```
     * @return A configuration object.
```

```
     * @throws Exception
```

```
     *      @
```

```
     */
```

```
    public String find(String palabra, String from, String to) throws Exception {
```

```
        String result = "";
```

```
        DataBaseHelper databaseHelper =
```

```
DataBaseHelperFactory.getInstance();
```

```
        Cursor cursor = null;
```

```
        try {
```

```
            cursor = databaseHelper.getDataBase().rawQuery("SELECT " +  
to + " FROM " + TABLE_NAME + " WHERE " + from + " = " + palabra.toUpperCase() +  
"", null);
```

```
            if (cursor.moveToFirst()) {
```

```
                result = cursor.getString(0);
```

```
            }
```

```

    } catch (Exception exception) {
        throw new Exception(exception.toString());
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return result;
}

/**
 * Save a configuration object on device.
 *
 * @param configuration
 *      A configuration object.
 * @throws Exception
 */
public boolean save(String espaniol, String quichua) throws Exception {

    boolean result = false;

    DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();
    Cursor cursor = null;
    try {
        ContentValues values = new ContentValues();
        values.put(ES, espaniol.toUpperCase().replace(" ", ""));
        values.put(QU, quichua.toUpperCase().replace(" ", ""));

        databaseHelper.getDataBase().insert(TABLE_NAME, null,
values);

        result = true;
    } catch (Exception e) {
        throw new Exception(e.toString());
    } finally {
        if (cursor != null)
            cursor.close();
    }
}

```

```

    }

    return result;
}

/**
 * Remove a configuration object on device.
 *
 * @throws Exception
 */
public boolean remove(String espaniol, String quichua) throws Exception {
    boolean r;
    DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();
    try {
        String SQL;
        SQL="DELETE FROM " + TABLE_NAME + " WHERE espaniol =
" + espaniol.toUpperCase() + """;
        databaseHelper.getDataBase().delete(TABLE_NAME, "espaniol
= " + espaniol.toUpperCase() + """, null);
        r=true;
    } catch (Exception exception) {
        r=false;
        throw new Exception(exception.toString());
    }
    return r;
}

/**
 * Eliminate the persistece of configuration object.
 *
 * @throws Exception
 */
public void drop() throws Exception {
    DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();

```

```

        try {
            databaseHelper.getDataBase().execSQL("DROP TABLE " +
TABLE_NAME);
        } catch (Exception exception) {
            throw new Exception(exception.toString());
        }
    }

    public void create() throws Exception {
        DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();
        try {
            databaseHelper.getDataBase().execSQL("CREATE TABLE IF
NOT EXISTS " + TABLE_NAME + " (" + ES + " TEXT," + QU + " TEXT);");
        } catch (Exception exception) {
            throw new Exception(exception.toString());
        }
    }

    @Override
    public boolean isEmpty() throws Exception {
        boolean result = true;
        DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();
        Cursor cursor = null;
        try {
            cursor = databaseHelper.getDataBase().rawQuery("SELECT *
FROM " + TABLE_NAME, null);

            if (cursor.moveToFirst()) {
                result = false;
            }
        } catch (Exception exception) {
            throw new Exception(exception.toString());
        } finally {
            if (cursor != null)

```

```

        cursor.close();
    }

    return result;
}

@Override
public ArrayList<String> findMatch(String startWith, String from) throws
Exception {

    ArrayList<String> result = new ArrayList<String>();

    DataBaseHelper databaseHelper =
DataBaseHelperFactory.getInstance();
    Cursor cursor = null;
    try {
        cursor = databaseHelper.getDataBase().rawQuery("SELECT " +
from + " FROM " + TABLE_NAME + " WHERE " + from + " LIKE " +
startWith.toUpperCase() + "%", null);

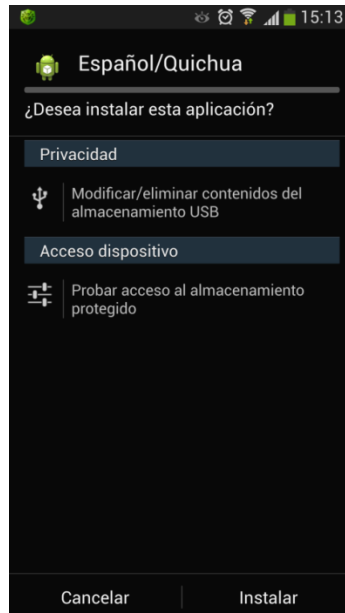
        while (cursor.moveToNext()) {
            result.add(cursor.getString(0));
        }
    } catch (Exception exception) {
        throw new Exception(exception.toString());
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return result;
}
}

```

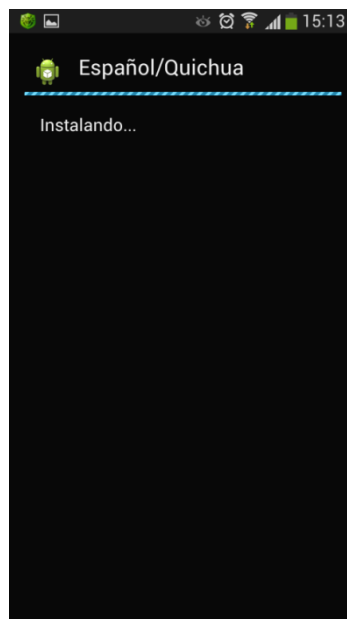
Manual de instalación

Para la instalación de la aplicación del “Traductor básico de palabras de español a quichua (Imbabura)” se procederá a realizar los siguientes pasos:

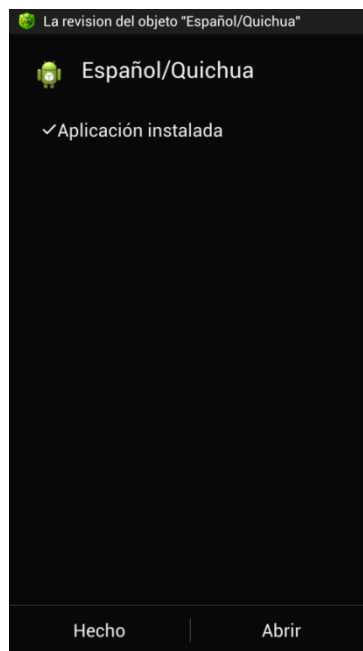
1. Descargar el archivo **DICCIONARIO.APK**
2. Mostrará un aviso de instalación de la aplicación



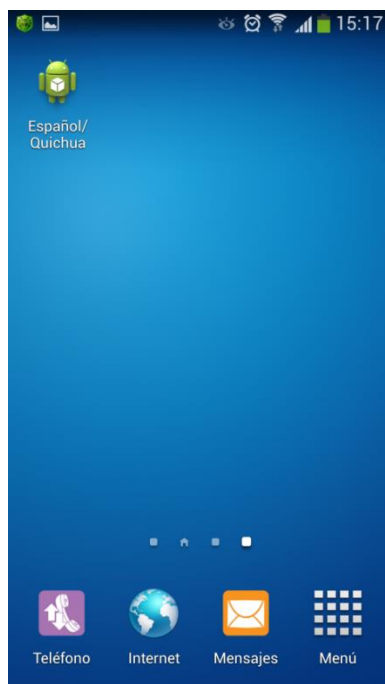
3. Iniciará el proceso de instalación



4. Al finalizar la aplicación mostrara el siguientes aviso



5. En la pantalla del escritorio se mostrara el siguiente icono, la aplicación esta lista para utilizarse.



GLOSARIO

Posts

Vocablo inglés traducido al español como mensaje.

Twitts

Expresión de un momento o idea. Puede contener texto, fotos y videos

Android

Sistema operativo pensado para teléfonos móviles basado en Linux.

Quichua / Kichwa

Familia de lenguas originaria de los Andes centrales.

Smartphone

Teléfono móvil construido sobre una plataforma informática móvil.

Tabletas / tablets

Tipo de computadora portátil de mayor tamaño que un Smartphone.

ATIS.

Grupo de investigación y desarrollo.

TIC

Tecnologías de la información y la comunicación

UNESCO

Organización de las Naciones Unidas para la Educación, Ciencia, y la Cultura.