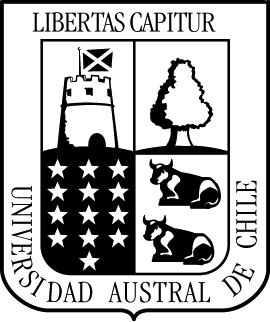


Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil en Informática



**CONJUNTO DE EXPERIENCIAS DOCENTES PARA EL DESARROLLO DE APLICACIONES INTERNAS Y EXTERNAS SOBRE *SMARTPHONES* CON BASIC4ANDROID Y TECNOLOGÍAS IOIO**

Proyecto para optar al título de

**Ingeniero Civil en Informática**

PROFESOR PATROCINANTE:

JORGE ANTONIO MORALES VILUGRÓN

INGENIERO ELECTRÓNICO

MAGISTER EN ADMINISTRACIÓN, MBA

**JORGE RODRIGO LARA HUENULEF**

VALDIVIA – CHILE

2015

# ÍNDICE

[ÍNDICE i](#_Toc427191410)

[ÍNDICE DE TABLAS ii](#_Toc427191411)

[ÍNDICE DE FIGURAS iii](#_Toc427191412)

[RESUMEN iv](#_Toc427191413)

[ABSTRACT v](#_Toc427191414)

[1. INTRODUCCIÓN. 1](#_Toc427191415)

[1.1 Objetivos 3](#_Toc427191416)

[1.1.1 Objetivo General 3](#_Toc427191417)

[1.1.2 Objetivos Específicos 3](#_Toc427191418)

[1.2 Motivación 3](#_Toc427191419)

[1.3 Impactos 4](#_Toc427191420)

[2. NIVEL ACTUAL 5](#_Toc427191421)

[2.1 Sistemas operativos 5](#_Toc427191422)

[2.2 Herramientas de desarrollo 7](#_Toc427191423)

[2.3 Estándares de desarrollo 10](#_Toc427191424)

[2.3.1 Principios de diseño 11](#_Toc427191425)

[2.3.2 Buenas prácticas 14](#_Toc427191426)

[3. TECNOLOGÍAS ASOCIADAS 16](#_Toc427191427)

[3.1 Tecnologías integradas 16](#_Toc427191428)

[3.2 Tarjeta IOIO 17](#_Toc427191429)

[3.3 SQLite 18](#_Toc427191430)

[3.3.1 ¿Qué es SQLite? 18](#_Toc427191431)

[3.3.2 Características distintivas 19](#_Toc427191432)

[3.3.3 Aplicaciones de SQLite 21](#_Toc427191433)

[3.3.4 Desventajas 21](#_Toc427191434)

[3.3.5 Ejemplo básico funcional 22](#_Toc427191435)

[4. DESCRIPCÍON DE LA METODOLOGÍA 23](#_Toc427191436)

[5. DESARROLLO 24](#_Toc427191437)

[5.1 Pruebas de conceptos 24](#_Toc427191438)

[6. CONCLUSIONES 24](#_Toc427191439)

[REFERENCIAS 25](#_Toc427191440)

[ANEXOS 29](#_Toc427191441)

[Anexo A: Código de ejemplo SQLite en Basic4Android 29](#_Toc427191442)

# ÍNDICE DE TABLAS

Tabla Página

[Tabla 1. Estudio comparativo de seis aplicaciones para programar en Android. 9](#_Toc415549425)

# ÍNDICE DE FIGURAS

Figura Página

[Figura 1. Evolución de los Abonados móviles y Penetración de Telefonía 5](#_Toc415549652)

[Figura 2. Penetración de *smartphones* en Chile el año 2014, clasificados por sistema operativo 6](#_Toc415549653)

[Figura 3. Gráfico exportado desde Excel en base al estudio realizado. 10](#_Toc415549654)

[Figura 4. Placa IOIO-OTG y sus componentes. 19](#_Toc415549655)

[Figura 5. Aplicación del ejemplo realizado utilizando SQLite en un *smartphone*. 23](#_Toc415549656)

# RESUMEN

Los dispositivos electrónicos inteligentes más usados por las personas son los smartphones por su portabilidad, y en la actualidad su avance tecnológico permite realizar múltiples tareas casi tanto como en un computador de escritorio.

Actualmente, el sistema operativo predominante en smartphones es Android, y existen varias plataformas *hardware* de desarrollo que complementan y potencian sus capacidades, siendo “IOIO” una de las placas más conocidas y utilizadas. Es por ello la importancia de estudiar las tecnologías actuales más innovadoras e ir más allá de las proporcionadas por el dispositivo, fomentando además el aprendizaje de un lenguaje de programación enfocado en Android, el que a la fecha, generalmente no es contemplado en los planes de estudios para carreras de Ingeniería o Informática, y que por su masificación presenta grandes oportunidades de negocio, tales como la publicidad que se adhieren al desarrollar una aplicación y distribuirla en el mercado de Google Play Store, o simplemente la venta del desarrollo móvil a entidades privadas y particulares de cualquier rubro.

Con el objetivo de suplir la carencia de conocimiento en el desarrollo de aplicaciones móviles, es necesario potenciar y enriquecer habilidades a través un conjunto de prototipos que sigan un modelo de desarrollo iterativo incremental y evolutivo.

En el documento se presentará un estudio previo de diseño como estándar de desarrollo móvil para realizar las experiencias sobre Android, utilizando las tecnologías internas de un *smartphone* y posteriormente incorporando múltiples tipos de sensores externos a través de la placa electrónica IOIO. Para que estas experiencias interactúen con distintos tipos de tecnologías sobre el sistema operativo Android, se escogerá una herramienta de desarrollo rápido de aplicaciones (RAD - *Rapid* *Application* *Development*) que permita generar y realzar las capacidades de la creación de nuevos productos en un corto periodo de tiempo, solucionando las falencias detectadas anteriormente. Finalizando, como producto, una completa *suite* de experiencias de desarrollo en aplicaciones para dispositivos *smartphones*, que podrían tener un futuro carácter formativo, experimental o pedagógico.

# ABSTRACT

# 1. INTRODUCCIÓN.

Con el pasar de los años, las tecnologías de la información y la comunicación (TIC) se ha transformado en uno de los pilares fundamentales de la sociedad moderna. Si bien, en un comienzo, los avances tecnológicos se presentaron como un bien escaso o de lujo en la población, hoy en día esa situación ha cambiado. Un reflejo de ello, es el alto índice de aparatos electrónicos con conexión a internet utilizados en la actualidad. La conectividad que en un principio era accesible solo para las empresas, ahora es un bien de uso masivo incorporado en los hogares, así como en el día a día de las personas, con dispositivos tales como smartphones, tablets y computadores portátiles.

Se sabe que los dispositivos móviles que más acompañan a las personas en su diario vivir son mayormente smartphones, los cuales superaron la venta de teléfonos normales el primer trimestre del año 2013 [Bus13], siendo algunos muy distintos, ya que poseen diferentes sistemas operativos, lo que implica un funcionamiento distinto para cada uno de ellos. Entre estos sistemas operativos, el más usado en la actualidad es Android [Sta14], teniendo más de 700.000 aplicaciones y juegos en el presente [Goo], que incluyen aplicaciones de utilidades y de entretención.

Los smartphones incluyen aplicaciones de telefonía básica, como son las llamadas, mensajes de texto, buzón de voz, calculadora y alarma, entre otras aplicaciones, que claramente facilitan la vida de una persona, en cuanto a comunicación de datos y tareas básicas, pero una de las grandes ventajas de estos teléfonos, es que incluyen componentes que permiten que otras aplicaciones funcionen en conjunto con las operaciones de telefonía básica de un teléfono. Estos componentes pueden ser micrófono, acelerómetro, GPS (*Global Positioning System*), pantalla táctil para entrada y salida de datos, que permiten obtener variadas aplicaciones que detonan el potencial de todos estos componentes.

Desde que apareció el primer teléfono inteligente con el sistema operativo Android, en el año 2008 [Fer08], Android Market fue la tienda virtual para albergar aplicaciones disponibles para descarga en los dispositivos con Android, con 2.300 aplicaciones aproximadamente en el año 2009 [Law09]. La tienda fue reemplazada por Google Play Store en el año 2012 [Ros12], para generar un nuevo concepto, enfocado no solo en la descarga de aplicaciones, sino también en la descarga de música, películas y libros. En la actualidad, Google Play Store, alberga más de 1 millón de aplicaciones, con más de 50.000 millones de descargas desde el 2008 [Xat13].

Con las tecnologías disponibles en los smartphones, se hace evidente la necesidad de hacer uso de estos componentes, obteniendo variados ventajas gracias a la utilización masiva de Google Play Store por parte de los usuarios. Entre los beneficios de crear aplicaciones para Google Play Store, se encuentran vender la aplicación, ofrecer publicidad, obtener un cierto grado de reconocimiento por parte de la comunidad, satisfacción personal, entre otros. Y con esto, parece lógico pensar que el número de las aplicaciones disponibles en Google Play Store irán en aumento y la cantidad de las descargas serán cada vez mayores, debido además, a que los dispositivos inteligentes son, día a día, más accesibles para el común de las personas.

Es un pensamiento muy acotado creer que las capacidades de un *smartphone* quedan o abarcan sólo dentro del mismo dispositivo. En los últimos tiempos, se ha escuchado mucho el término de “domótica”, que en palabras simples es la integración de las tecnologías de información en el hogar. Cuyo avance no está ajeno a la utilización de un smartphone para controlar todo un sistema a distancia, compuesto por dispositivos electromecánicos y sensores, que automatizan funciones dentro de una casa, tales como supervisar o gestionar la seguridad, incrementar el ahorro de energía, abrir y cerrar puertas, ventanas y portones, encender y apagar luces, alarmas y calefacción, entre otros. Esto es sólo una idea de todas las posibilidades de alcance que podría realizar un *smartphone*.

## Objetivos

### 1.1.1 Objetivo General

Diseñar e implementar un conjunto de experiencias docentes para el desarrollo de aplicaciones internas y externas sobre *smartphone* con Basic4android y tecnologías IOIO.

### 1.1.2 Objetivos Específicos

* Describir el estándar de diseño para el desarrollo de aplicaciones móviles en Android, las herramientas de desarrollo más utilizadas, las tecnologías integradas en los *smartphones* y *hardware* externo adicional.
* Definir y diseñar un conjunto de experiencias con tecnologías integradas y externas, con un nivel de dificultad creciente y conocimiento acumulativo.
* Desarrollar y documentar las experiencias definidas, con el fin de generar material replicable, que genere conocimiento con respecto a la programación en Android.
* Validar los prototipos cumpliendo los requisitos estipulados, realizar pruebas con ajustes y mejoras.

## 1.2 Motivación

* Crear un conjunto de experiencias que incluyan tecnologías integradas y externas, con el fin de facilitar el aprendizaje de las tecnologías involucradas en los *smartphone*. Impulsando la creación de nuevas aplicaciones de proyectos experimentales o de innovación.
* La escasa documentación de estándares de diseño debido al poco auge o difusión de la importancia de un buen diseño y la no obligación de tales estándares por parte de Google.
* Lentos ciclos de diseño, desarrollo y pruebas en la creación de nuevas aplicaciones móviles en los proyectos o tesis de alumnos en la universidad.
* Integrar tecnología externa al *smartphone* con el objetivo de expandir los tipos o áreas de aplicación. En particular tecnologías IOIO.
* La oferta de *smartphones* y la demanda de aplicaciones van en aumento, en conjunto a la fácil adquisición de estos dispositivos, hace de un interesante negocio desarrollar software sobre la plataforma Android.

## 1.3 Impactos

Considerando el gran número de dispositivos móviles y de la importancia que tienen éstos para las personas, en particular por aquellas aplicaciones que facilitan su diario vivir, el impacto de este proyecto está relacionado con los tres productos que se obtienen al final de su desarrollo: sistema de hardware, software y documentación, entendiéndose como sistema de hardware, la capacidad de los prototipos de leer y escribir información en dispositivos externos al *Smartphone*; entiéndase por software el que incluye la tarjeta de desarrollo del *hardware* externo y el software que se ejecuta en el teléfono inteligente; y como documentación, se entiende el registro de toda la experiencia acumulada en el desarrollo de este proyecto.

Por lo tanto, este proyecto de tesis actuará como un material de consultas o referencias para los alumnos de la carrera de Ingeniería Civil en Informática, que deseen desarrollar aplicaciones móviles para *smartphones* con Android, principalmente para alumnos tesistas que necesitan ayuda en la obtención de información estructurada con estándares de diseño claros y herramientas de desarrollo ágiles, impactando positivamente la curva de aprendizaje. Esto permitirá concentrarse en la solución del problema específico.

# 2. NIVEL ACTUAL

Con el pasar de los años, la tecnología se ha vuelto imprescindible en la sociedad chilena y mundial. Es así, como desde el 2000 al 2012, la cifra de abonados móviles casi quintuplicó su cantidad, e incluso más, al 2012, Chile tenía 24,1 millones de celulares activos, lo que corresponde a un promedio de 1,38 celulares por habitante, lo cual se ve reflejado en el gráfico de la figura 1, la que solo representa a la cantidad de celulares en general [Sub13, 16].

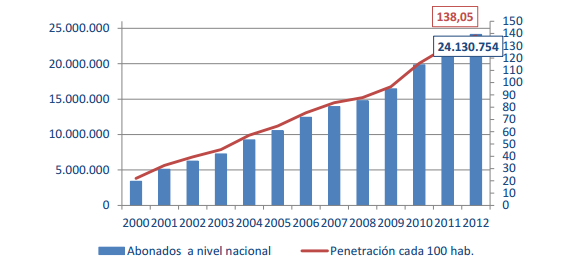


Figura 1. Evolución de los Abonados móviles y Penetración de Telefonía[[1]](#footnote-1)

## 2.1 Sistemas operativos

En cuanto a la penetración que tienen los sistemas operativos, Android va liderando en el mercado nacional, según lo que se puede observar en las figura 2. El competidor más cercano a Android, es representado por iOS, pero se puede ver una gran brecha de penetración que existe entre ambos.

Por esta razón, se ha decidido trabajar en aplicaciones móviles desarrolladas para Android, considerando además, que al día de hoy existen diferentes marcas y modelos de *smartphones* con este sistema operativo, los que cuentan con variadas tecnologías, que permiten un funcionamiento íntegro del dispositivo, ayudando en la ejecución de las tareas diarias de una persona y además, en su entretenimiento. Algunas de estas tecnologías son la pantalla táctil, micrófono, parlante, GPS, acelerómetro, sensor fotoeléctrico, cámara, led flash, sensor de proximidad, bluetooth, Wifi, conexión de datos móvil, etc.

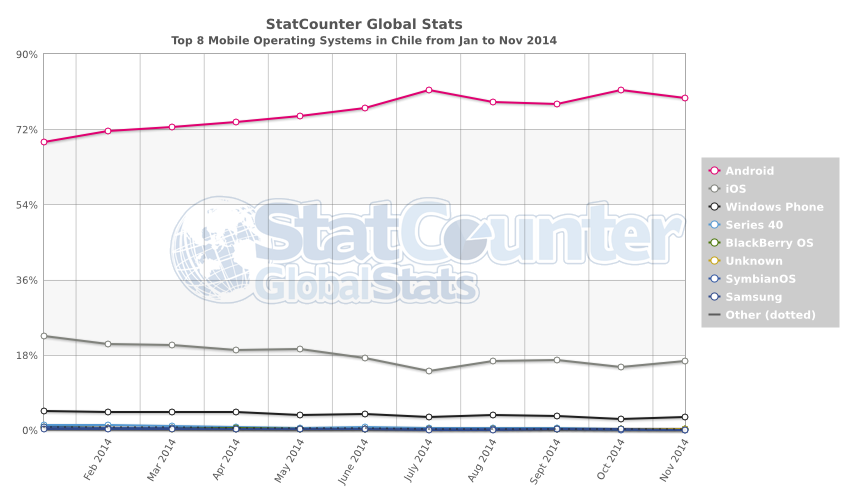


Figura 2. Penetración de *smartphones* en Chile el año 2014, clasificados por sistema operativo[[2]](#footnote-2)

Además de las tecnologías incorporadas en un teléfono inteligente, se pueden agregar nuevas funcionalidades, incorporando la tarjeta IOIO, la cual provee una conectividad robusta a cualquier dispositivo Android vía conexión USB (*Universal Serial Bus*) y se puede controlar desde la aplicación Android, usando librerías provistas para su programación [See].

Teniendo tales dispositivos y sensores de captura de información, se hace necesario tener un sistema para contener o almacenar datos en un dispositivo Android. De esta manera se presenta SQLite, ya que trabaja de forma integrada en la aplicación, y por este motivo, es recomendada para Android. SQLite nace de la idea de escribir un simple motor de base de datos SQL que permitiera leer los archivos del disco duro, y luego ser llamados en diferentes solicitudes. Entre sus características está el tamaño, rendimiento, portabilidad, estabilidad, ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), SQL, Interfaces y costo gratuito [Cab07].

## 2.2 Herramientas de desarrollo

Luego a la hora de querer comenzar un proyecto se pueden tener muy buenas ideas para crear una aplicación, pero existe un desconocimiento en cuanto al desarrollo e implementación de éstas, por lo que muchas ideas no pueden ser ejecutadas con éxito. Para ello, existen diversos programas de desarrollo como marcos de trabajo enfocadas en la programación de aplicaciones para Android. Entre las herramientas más usadas y conocidas se destacan las siguientes:

* Basic4Android: Es un simple y poderoso RAD (*Rapid Application Development*), el cual se basa en lenguaje *Visual Basic*. Facilita la programación gráfica y está enfocado en un lenguaje de programación orientado a objetos, cuenta con una fuerte y activa comunidad online con más de 69.000 desarrolladores [Bas15a]. Su precio es de 59 dólares para 2 meses, 119 dólares para 2 años, y 599 dólares para 2 años y hasta 30 desarrolladores [Bas15b].
* Xamarin: Aplicación para programación móvil basado en C# con una comunidad de 628332 desarrolladores [Xam15a]. Su precio es de 299 dólares para un año (versión *Indie*), 999 dólares por año (versión Comercial) y 1899 dólares para un año (versión Empresa) [Xam15b].
* App Inventor: Plataforma impulsada por Google. Se basa en un set de instrucciones ya hechas, las cuales son arrastradas cuando se necesiten, para formar el código [Mit15a]. El sistema es gratuito y se puede descargar fácilmente de la web [Mit15b]. Las aplicaciones fruto de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. También existe su versión online para programar desde una página *web* [Mit15c].
* LiveCode: IDE (*Integrated Development Environment*) basado en la multifuncionalidad. En este IDE, se programa una vez y el código sirve para Android, iOS, Windows, Linux y MAC. Se basa en un lenguaje orientado a eventos [Yes12]. Su precio es de 1188 dólares por año (versión Comercial) y 2388 dólares por año (versión Pro) [Liv14].
* Eclipse: IDE basado en el lenguaje de programación JAVA, que requiere la instalación de Android SDK. Consta de un simulador del dispositivo móvil, por lo que debe entregarse las características del dispositivo a simular, lo que ralentiza el proceso de depuración de las aplicaciones. A diferencia de otros IDE, la plataforma es Open Source [Ecl15].
* Android Studio: es un entorno de desarrollo integrado (IDE) para la plataforma Android. Android Studio está disponible gratuitamente para desarrolladores que quieran probarlo. Basado en IntelliJ IDEA de JetBrains, está diseñado específicamente para desarrollar para Android [And12c]. Está disponible para descargar para Windows, Mac OS X y Linux [And12d].

La tabla 1 compara todas las alternativas descritas para programar en Android, basándose en una serie de parámetros que reflejan las características de cada opción, evaluados de 1 a 7, en donde 1 es lo menos cercano y 7 lo más cercano.

Tabla 1. Estudio comparativo de seis aplicaciones para programar en Android.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Basic 4  Android | Xamarin | App  Inventor | LiveCode | Eclipse | Android Studio |
| Fácil de instalar | 5 | 5 | 7 | 5 | 3 | 5 |
| Fácil de programar | 7 | 5 | 7 | 5 | 4 | 5 |
| Precio (económico) | 5 | 3 | 7 | 2 | 7 | 7 |
| Disponibilidad de documentación | 5 | 4 | 5 | 4 | 7 | 3 |
| Vista en tiempo real del diseño | 7 | 7 | 7 | 7 | 1 | 7 |
| Interacción con dispositivo móvil | 7 | 5 | 5 | 5 | 5 | 5 |
| Integración de librerías | 6 | 6 | 1 | 6 | 7 | 6 |
| Integración de tecnologías (IOIO) | 7 | 1 | 1 | 1 | 7 | 7 |
| Estabilidad | 6 | 6 | 7 | 6 | 5 | 3 |

A partir de la tabla 1, la figura 3 representa visualmente la comparación.

Figura 3. Gráfico exportado desde Excel en base al estudio realizado.

Dentro de estas posibilidades, se decidió utilizar Basic4Android, ya que además de ser fácil de programar, es una herramienta con costo accesible, fácil de instalar e integrar con la tecnología IOIO, cómodo de utilizar, con tiempo de ejecución rápido y que además se está utilizando cada vez más en las empresas para desarrollar en Android, incluso por una empresa Valdiviana creada por un ex alumno de la carrera de Ingeniería Civil en Informática de la Universidad Austral.

## 2.3 Estándares de desarrollo

Sin embargo, no basta tener en cuenta un marco de trabajo, es necesario además llevar una cierta estructura de diseño para mantener el orden, estilo y consistencia del sistema operativo, evitando el problema de la fragmentación en Android[Gut14].

A raíz del problema de fragmentación, Google lanzó estándares de diseño a partir desde el 2012 [Hil12], de manera de asegurar la misma calidad en cualquier aplicación Android, entregando un manual de buenas prácticas y principios de diseño [And].

### 2.3.1 Principios de diseño

Estos principios de diseño fueron desarrollados por el equipo de experiencia de usuario Android, teniendo en mente los intereses de los usuarios, los cuales se deben considerar al hacer una aplicación, al aplicar tu propia creatividad y al establecer el diseño de ésta. Por tanto, se debe diseñar aplicaciones considerando tres puntos principales: sedúceme, simplifica mi vida y sorpréndeme [And12a].

**Sedúceme**

La belleza es algo que está más allá de lo superficial. Las aplicaciones de Android son elegantes y estéticamente agradables en múltiples niveles. Las transiciones son rápidas y claras; la disposición de los elementos es eficaz y las tipografías son nítidas y significativas. Los iconos de las aplicaciones son obras de arte en sí mismos. Como una herramienta bien hecha, tu aplicación debería esforzarse en combinar la belleza, simplicidad y el propósito de crear una mágica experiencia que sea fácil de usar además de poderosa.

Encántame con sorpresas

Una superficie bonita, una animación cuidadosamente colocada, o un oportuno efecto de sonido es una grata experiencia. Efectos sutiles que contribuyan a tener la sensación de realizar menos esfuerzo pero a su vez tener una fuerza poderosa en la mano.

Los objetos reales son más divertidos que botones y menús

Permite a la gente tocar y manipular directamente los objetos en tu aplicación. Eso reduce el esfuerzo cognitivo necesario para realizar una tarea además de ser emocionalmente más satisfactoria.

Permite que la adapte a mi estilo

Todos adoran sus propios toques personales porque estos ayudan a sentirse como en casa y con todo bajo control. Proporciona una personalización opcional divertida que no dificulte las tareas principales, además de proporcionar una por defecto, simple y bonita.

Consigue que me conozca

Aprende de las preferencias de la gente con el paso del tiempo. En lugar de pedirles que tomen la misma decisión continuamente, coloque las opciones que suelen seleccionar anteriormente a su alcance.

**Simplifica mi vida**

Hágalo breve

Usa frases cortas con palabras sencillas. Las personas tienden a saltarse las oraciones si son demasiado largas.

Las imágenes son más rápidas que las palabras

Considera usar imágenes para explicar ideas. Estas consiguen la atención de la gente y pueden ser mucho más eficientes que las palabras.

Decide por mí, pero déjame tener la última palabra

Seleccione su mejor respuesta y actúa antes de preguntar. Demasiadas opciones y decisiones provocan insatisfacción en las personas. Pero, por si te equivocas, permite la opción “deshacer”.

Muestra solo lo que yo necesito cuando me haga falta

Las personas se agobian cuando ven demasiada información al mismo tiempo. Separa las tareas y la información en pequeños y distinguibles fragmentos. Oculta opciones que no son esenciales en ese momento y enséñales cómo acceder a ellas.

Debería saber siempre donde estoy

Da a las personas la confianza de saber que tienen a su alrededor. Haz elementos en tu aplicación que se distingan y usa transiciones para mostrar relaciones entre las pantallas. Proporciona una retroalimentación en las tareas y los progresos.

No pierdas nunca mis cosas

Guarda lo que la gente se tomó tiempo en crear y permite que puedan acceder desde cualquier sitio a esas creaciones. Recuerda ajustes, toques personales y las creaciones a lo largo de los teléfonos, *tablets* y ordenadores. Haz que una actualización sea la cosa más simple del mundo.

Si parece igual, debe actuar del mismo modo

Ayuda a la gente a percibir las distintas funciones haciendo que tengan aspectos visuales sutilmente diferentes. Evita los modos, los cuales ocupan lugares distintos, pero actúan de diferente forma a la misma entrada.

Solo interrúmpame si es importante

Como un buen asistente personal, protege a la gente de los detalles que no son importantes. La gente quiere estar concentrada, y a menos que sea crítico y susceptible al tiempo, una interrupción puede ser agotadora y frustrante.

**Sorpréndeme**

Dame trucos que funcionen en todos los sitios

Las personas se sienten bien cuando entienden las cosas por sí mismos. Haz tu aplicación tan simple aprovechando pautas visuales y los recuerdos que tienen en la memoria de otras aplicaciones de Android. Por ejemplo, el gesto de deslizar puede ser un buen atajo para navegar.

Pequeños detalles

Divide las tareas complejas en pasos más pequeños y simples. Proporciona comentarios y sugerencias sobre las acciones del usuario, aunque sea un sutil resplandor.

No es mi fallo

Se gentil al pedir a las personas que hagan correcciones. Ellos quieren sentirse inteligentes cuando usan tu aplicación. Si algo va mal, da unas instrucciones claras de cómo solucionar los problemas separando los detalles técnicos. Si tú puedes arreglar los problemas detrás del escenario, es incluso mejor.

Haz las tareas pesadas por mí

Consigue que los principiantes se sientan como expertos habilitando opciones que ellos nunca pensaron que podrían hacer. Por ejemplo, los atajos que combinan múltiples efectos para las fotos pueden hacer a fotógrafos amateur parecer increíbles en solo unos pasos.

Haz las tareas importantes más rápido

No todas las acciones son iguales. Decide cuales son más importantes en tu aplicación y haz que esas sean fáciles de encontrar y rápidas de usar, como el botón de hacer foto en una cámara, o el botón de pausa en un reproductor de música.

### 2.3.2 Buenas prácticas

* Interacción y atracción:

Atraer y retener a usuarios mediante la implementación de los mejores patrones de interacción en Android. Por ejemplo, para ayudar a los usuarios a descubrir rápidamente el contenido en la aplicación, la aplicación debe coincidir con sus expectativas para la interacción del usuario en Android. Y para mantener a los usuarios de regreso, se debe tomar ventaja de las capacidades que revela la plataforma y abrir el contenido sin que los usuarios pasen por el lanzador de aplicaciones [And12b].

* Interfaz de usuario:

Construir una interfaz de usuario con los *Layouts* de Android para todo tipo de dispositivos. Android proporciona un marco de trabajo flexible para el diseño de interfaz de usuario, que permite visualizar distintos *Layouts* para diferentes dispositivos. Por ejemplo, como admitir varios tamaños de pantalla, admitir varias densidades de pantalla, e implementar interfaces de usuario adaptables [And12b].

* Entradas de Usuario:

Utilizar las diferentes entradas del usuario, tales como gestos de la pantalla táctil y la introducción de texto a través de métodos de entrada sobre la pantalla y teclados o botones físicos adicionales [And12b].

* Procesos en segundo plano:

Ejecutar procesos en segundo plano para mejorar el rendimiento de la aplicación y reducir al mínimo su consumo de la batería [And12b].

* Rendimiento:

Construir una aplicación que sea suave, sensible, y que utilice batería tan poco como sea posible. Por ejemplo, no realizar el trabajo que no necesitas hacer, no asignar memoria si puedes evitarlo, e inhabilitar las actualizaciones de servicios en segundo plano o disminuir la frecuencia de dichas actualizaciones cuando el nivel de batería sea bajo para optimizar la duración de ésta [And12b].

* Seguridad y privacidad:

Realizar diversas tareas manteniendo los datos de la aplicación y del usuario de forma segura. Por ejemplo, el almacenamiento de información en la memoria interna y externa o el traspaso de información en transacciones sobre internet [And12b].

* Pruebas:

Realizar y documentar pruebas como parte del ciclo de desarrollo de la aplicación Android*.* Esto podría ayudar a encontrar tempranamente *bugs* o errores y proporcionar confianza en el código [And12b].

# 3. TECNOLOGÍAS ASOCIADAS

## Tecnologías integradas

Es muy común encontrar diversas tecnologías incorporadas en un *smartphone* que facilitan y complementan las soluciones tecnológicas a través de aplicaciones desarrolladas. Algunas definiciones de estas tecnologías son:

* Acelerómetro: Un acelerómetro es un transductor electromecánico el cual produce un voltaje o carga de salida que es proporcional a la aceleración a la que se somete [Tra84, 100].
* Micrófono: Un micrófono es un elemento capaz de captar ondas sonoras convirtiendo la potencia acústica en eléctrica de similares características ondulatorias. Para ello se necesita la combinación escalonada de dos tipos de transductores. El primero de ellos consiste en una fina lámina, denominada diafragma. Su misión es transformar las variaciones de presión en vibraciones mecánicas, es por tanto un transductor mecano-acústico. El segundo transforma las vibraciones mecánicas recibidas en magnitudes eléctricas, es por tanto un transductor electromecánico. El conjunto de los dos transductores puede considerarse como uno electro-acústico [Uni03].
* GPS: Sus siglas significan *Global Position System* (Sistema de Posición Global), consistente en una compleja red mundial de satélites en órbita y que conforman un sistema de posicionamiento y ubicación geográfica de móviles en la superficie del planeta, de manera que se pueda saber con exactitud la ubicación de un objeto [Tas].
* Cámara: es un dispositivo electrónico usado para capturar y almacenar fotografías electrónicamente en un formato digital. Las cámaras digitales compactas modernas generalmente son multifuncionales y contienen algunos dispositivos capaces de grabar sonido y/o video además de fotografías [Jaz93].
* Sensor de proximidad: un sensor de proximidad es un transductor que detecta objetos o señales que se encuentran cerca del elemento sensor [Dac].
* Bluetooth: es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz [Pcm].
* Wifi: Tecnología de red inalámbrica, que permite la conexión de dispositivos móviles, a través de puntos de acceso inalámbricos, que se realizan mediante ondas de radiofrecuencia [Ibe12].
* Parlante: es un transductor electro-acústico, es decir, convierte energía eléctrica en energía acústica. Esta conversión tiene lugar en dos etapas: la señal eléctrica produce el movimiento del diafragma del altavoz y este movimiento produce a su vez ondas de presión (sonido) en el aire que rodea al altavoz [Uni03b].

## Tarjeta IOIO

El IOIO (pronunciado como yo-yo) es una tarjeta especialmente diseñada para trabajar con dispositivos Android (versión OS 1.5 o mayor). La tarjeta provee una conectividad robusta a cualquier dispositivo Android vía conexión USB y es totalmente controlable desde dentro de las aplicaciones Android usando una API de Java [Spa13].

El IOIO contiene un pequeño microcontrolador que actúa como USB Host e interpreta las peticiones de la aplicación Android. Por otro lado, el microcontrolador puede interactuar con otros dispositivos físicos conectados al mismo tales como sensores, actuadores, etc. Utilizando las señales y protocolos convencionales tales como entradas y salidas digitales, entradas analógicas, SPI, UART, entre otros [Oli13].

Para la realización de este proyecto, se utilizará la nueva versión de la placa IOIO de SparkFun Electronics, bautizada como IOIO-OTG la cual ya se encuentra disponible. Entre las nuevas características, ahora tiene un puerto mini-usb para la conexión con el dispositivo, además la nueva versión incluye un jack de conexión JST de 2 pines para la alimentación y cuenta con una nueva versión del firmware, por lo que se han actualizado las librerías y han cambiado ciertas instrucciones respecto de la versión anterior. En la Figura 4 se aprecia una tarjeta IOIO-OTG, indicándose los principales pines y partes.

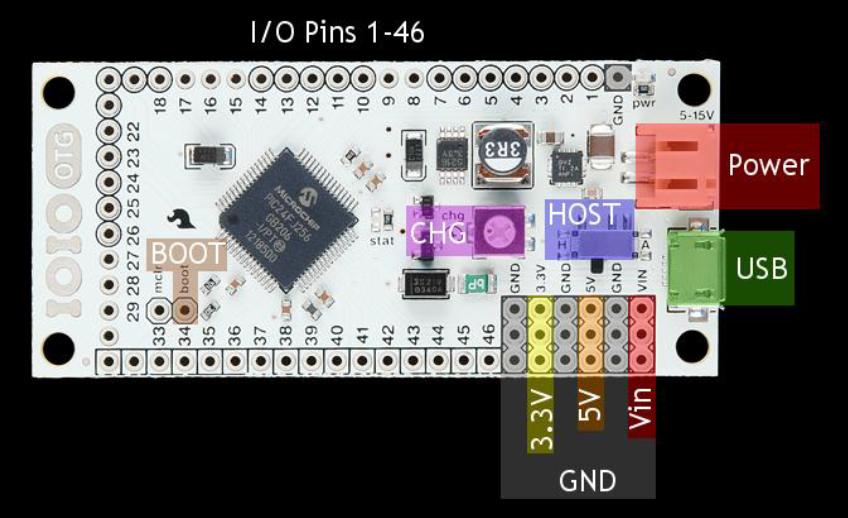


Figura 4. Placa IOIO-OTG y sus componentes[[3]](#footnote-3).

## SQLite

SQLite apareció en mayo del año 2000 de la mano de su creador D. Richard Hip, quién ha liberado las diferentes versiones de SQLite en base a la licencia GPL por lo que su código es de dominio público y puede ser modificado por cualquier persona. Gracias a esto, SQLite ha sido mejorada a lo largo de 15 años por un gran número de colaboradores y también ha sido migrada a diversas plataformas [Rom07].

### ¿Qué es SQLite?

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente [Rom07].

### 3.3.2 Características distintivas

De dominio público

Sencillamente el código fuente de SQLite es de dominio público. Por éste hecho Sqlite está libre de derechos de autor, lo que permite el libre uso de esta pequeña biblioteca de base de datos [Sql15].

Sentencias SQL Compiladas en la Máquina Virtual

Cada Motor de Base de datos SQL compila cada sentencia SQL en algún tipo de estructura de datos interna que luego se utilizará para llevar a cabo la labor de la declaración. SQLite ejecuta las sentencias y las compila a través de una máquina virtual que nos sirve mucho para poder depurar las lecturas de datos y sentencias [Sql15].

De código Legible

El código fuente de SQLite estás diseñado y codificado para ser legible y accesible hasta para un programador medio [Sql15].

Registros de Longitud Variable

A diferencia de muchos motores de bases de datos que usan capacidades estáticas en su almacenamiento, SQLite únicamente ocupa la cantidad de espacio de disco necesario para almacenar realmente la información ingresada [Sql15].

Tipado Dinámico

Permite al usuario almacenar cualquier valor de cualquier dato en cualquier columna, independientemente de la declaración del tipo de la columna [Sql15].

Compacto

Toda la librería funcional es menor de 500 KiB, de este modo es posible desactivar en tiempo de compilación numerosas características innecesarias para poder reducir aún más el tamaño de la biblioteca llegando más o menos a 300 KiB [Sql15].

Único archivo de Base de Datos

La gran ventaja que trae SQLite es poder almacenar todos los datos en un único archivo y en el directorio que desees, facilitando el copiado a un USB por ejemplo, ya que por el contrario otros tipos de gestores de BD habitualmente almacenan más de un archivo para una base de datos y en directorios de difícil acceso [Sql15].

Sin Servidor

No se necesita una comunicación (TCP/IP) o de tipo cliente/servidor para poder acceder a los datos, con SQLite puedes leer los datos directamente sin intermediarios y también con múltiples usuarios o procesos en paralelo [Sql15].

Archivo estable multiplataforma

El formato de archivo de SQLite es multiplataforma. Un archivo de base de datos por escrito en una máquina puede ser copiado y utilizado en una máquina diferente, con una arquitectura diferente. Además, los desarrolladores han prometido mantener el formato de archivo estable y compatible con versiones anteriores, por lo que las nuevas versiones de SQLite pueden leer y escribir archivos más antiguos de base de datos [Sql15].

Sin configuración

De la forma en que fue creado y diseñado SQLite, no necesita ser instalado, prendido, reiniciado o apagado, e incluso menos configurarlo. Esta cualidad permite que no haya un administrador de base de datos para crear las tablas, vistas, asignar permisos. O bien la adopción de medidas de recuperación de servidor por cada caída del sistema, ya que no hay nada que solucionar, SQLite simplemente funciona [Sql15].

### 3.3.3 Aplicaciones de SQLite

Las características y plataformas previamente mencionadas hacen de SQLite una excelente opción en diversos casos tales como:

* Cuando se requiere una base de datos integrada dentro de una aplicación. SQLite es una excelente opción por su facilidad de configuración. El inconveniente es que no escala a bases de datos demasiado grandes (en el orden de los terabytes).
* Para realizar demostración de aplicaciones que utilizan un RDBMS (¿Para qué utilizar un manejador de BD pesado que ocupa grandes recursos de sistema cuando solo se requiere hacer un demo de una aplicación?)
* Como cache local de un manejador de base de datos empresarial. Esto acelera el tiempo de respuesta y reduce la carga sobre la base de datos central.
* Para aplicaciones en dispositivos móviles que manejan una BD local que se sincroniza por *batch* con una base de datos remota.
* Almacenamiento persistente de objetos, configuraciones y preferencias de usuario. Permite fácilmente crear una estructura para almacenar configuraciones de la aplicación [Rom07].

### 3.3.4 Desventajas

* **Limitaciones en *Where***: esta limitación está dada por el soporte para clausuras anidadas [Ref14].
* **Falta de clave foránea**: se hace caso omiso de las claves foráneas; esto quiere decir, cuando se realice la creación de la tabla desde el modo consola, está permitiendo el uso de la clausura, aunque no realizara el chequeo de la misma [Ref14].
* **Falta de documentación en español**: si bien ya contamos con una comunidad latino americana de SQLite, sería importante encontrar mucha más documentación, libros, revistas, etc. como muchos otros motores de bases de datos cuentan hoy en día [Mal08].

### 3.3.5 Ejemplo básico funcional

Para realizar esta prueba, se desarrolla una pequeña aplicación en la cual se despliegan tres componentes en la pantalla principal. El primero es una entrada de texto que se utiliza para digitar cualquier tipo de información a través de un teclado virtual, el segundo componente es un botón táctil que sirve para almacenar la información digitada anteriormente, y el tercer componente tiene como fin desplegar en una lista la información almacenada en la base de datos sobre la pantalla principal.

Finalmente se logra el objetivo de implementar una aplicación que utilice conceptos básicos para el control de lectura y escritura en una base de datos SQLite sobre un *smartphone*. En la figura 5 se aprecia el dispositivo móvil ejecutando la aplicación. Un fragmento de código de esta aplicación está disponible en el Anexo A de este documento.



Figura 5. Aplicación del ejemplo realizado utilizando SQLite en un *smartphone*.

# 4. DESCRIPCÍON DE LA METODOLOGÍA

Para realizar el proyecto, se constará de un horario fijo de desarrollo, con apoyo del profesor patrocinante y con búsquedas en Internet acerca de tecnologías orientadas a *smartphones* y documentación para programar dichas tecnologías.

Para el primer objetivo específico, se condensarán conceptos obtenidos de la página oficial de desarrollo propuesta por Google para su sistema operativo Android, y se elegirá un conjunto de aquellos conceptos necesarios para las aplicaciones a desarrollar. Se utilizará múltiples páginas de Internet para buscar las tecnologías incluidas en un *smartphone*, y luego se describirán en un documento que las explique. Además, se hará una lista de las posibles opciones de marcos de trabajo para programar en Android, escogiendo la más apropiada.

Para el segundo objetivo específico, se analizarán todas las tecnologías incluidas en el documento creado en el primer objetivo específico, y se escogerán algunas para utilizar en las experiencias. Todo el conjunto de experiencias se crearán en base a un modelo de desarrollo iterativo incremental y evolutivo a través de reuniones periódicas con el patrocinante. En este punto, también se definirán todas las actividades a realizar, indicando para cada una de ellas las tecnologías que estarán asociadas.

Para el tercer objetivo específico, se creará un documento que describa paso a paso, las experiencias realizadas, incluyendo los casos en que se haya experimentado alguna dificultad técnica o consideración, para que futuros programadores puedan tener presente esos casos.

Por último, para el cuarto objetivo específico, se efectuarán pruebas y controles de las aplicaciones creadas, dentro de un grupo de estudiantes pertenecientes a un curso electivo del patrocinante, para así realizar ajustes necesarios y verificar que funcionen correctamente. En este punto, se considerará como experiencia satisfactoria, la aprobación del patrocinante finalmente.

# 5. DESARROLLO

Los elementos *hardware* y *software* que se utilizan para la mayor parte de las experiencias son los que se muestran a continuación. En algunos casos se agregaron otros dispositivos, los que están señalados en cada prueba de concepto.

*Hardware* utilizado:

* Placa IOIO-OTG
* *Notebook* Packard Bell Intel Core 2 Duo de 1.4 Ghz (para desarrollo).
* *Smartphone* Motorola Atrix 2 MB865 con Android 2.3.6 (para desarrollo).
* *Smartphone* Samsung Galaxy Camera EK-GC200 con Android 4.3 (para compatibilidad y pruebas).

*Software* utilizado en PC de desarrollo:

* Sistema operativo Windows 8 Pro.
* IDE para desarrollo Basic4Android 5.02.
* Control de versiones GIT, con respaldo web en github.com.

## 5.1 Pruebas de concepto

A continuación se muestran las experiencias de desarrollo realizadas con un enfoque pedagógico para su posterior uso formativo. Todas las pruebas de concepto tienen una relación jerárquica de menor a mayor complejidad basadas en un modelo de desarrollo iterativo incremental y evolutivo. Son nueve pruebas de concepto, cuyas cinco primeras usan sólo tecnologías internas del *smartphone* que deben ser desarrolladas bajo la temática de “reseñas” con el motivo de enlazar las distintas tecnologías involucradas en un solo objetivo, y las últimas cuatro pruebas de concepto usan tecnologías externas mediante una placa IOIO teniendo como temática la “domótica”. Para la construcción de las aplicaciones se tomaron en cuenta las siguientes “buenas prácticas”: interfaz de usuario, entradas de usuario, procesos en segundo plano, rendimiento, y pruebas.

# 6. CONCLUSIONES

# REFERENCIAS

[And12a] Android Developers (2012). Google Inc. Android Design Principles. Disponible en <http://developer.android.com/intl/es/design/get-started/principles.html>. Consultado el 30 de marzo de 2015.

[And12b] Android Developers (2012). Google Inc. Capacitación. Disponible en <http://developer.android.com/intl/es/training/index.html>. Consultado el 30 de marzo 2015.

[And12c] Android Developers (2012). Google Inc. Herramientas. Disponible en <http://developer.android.com/intl/es/tools/studio/index.html>. Consultado el 15 de marzo de 2015.

[And12d] Android Developers (2012). Google Inc. Download Android Studio. Disponible en <http://developer.android.com/intl/es/sdk/index.html>. Consultado el 15 de marzo de 2015.

[Bas15a] Basic4Android (2015). Anywhere Software. General Features. Disponible en http://www.b4x.com/android/why.html. Consultado el 20 de marzo de 2015.

[Bas15b] Basic4Android (2015). Purchase. Disponible en http://www.b4x.com/store.html. Consultado el 15 de marzo de 2015.

[Dac] Dacs (n.d.). blogspot. Sensores de proximidad. Disponible en <http://sensoresdeproximidad.blogspot.com/>. Consultado el 30 de marzo de 2015.

[Ecl15] The Eclipse Foundation (2015). Downloads. Disponible en <https://eclipse.org/downloads/>. Consultado el 15 de marzo de 2015.

[Ibe12] Iber systems. 802.11 – Wifi (2012). Disponible en <http://www.ibersystems.es/blogredesinalambricas/802-11-wifi/>. Consultado el 30 de marzo de 2015.

[Jaz93] Jazz. Wordpress (1993). Características de una cámara digital. Disponible en <https://jazz1993.wordpress.com/4-%C2%BFque-es-una-camara-digital-y-caracteristicas-de-una-camara-digital/>. Consultado el 30 de marzo de 2015.

[Liv14] LiveCode (2014). Store. Disponible en <https://livecode.com/store/>. Consultado el 15 de marzo de 2015.

[Mal08] Daniel Martin Maldonado (2008). Empresa & Economía. SQLite, el motor de base de datos ágil y robusto. Disponible en <http://www.empresayeconomia.es/aplicaciones-para-empresas/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>. Consultado el 30 de marzo de 2015.

[Mit15a] Massachusetts Institute of Technology (2015). MIT App Inventor. About Us. Disponible en <http://appinventor.mit.edu/explore/about-us.html>. Consultado el 15 de marzo de 2015.

[Mit15b] Massachusetts Institute of Technology (2015). MIT App Inventor. Installing App Inventor 2 Setup on Windows. Disponible en <http://appinventor.mit.edu/explore/ai2/windows.html>. Consultado el 15 de marzo de 2015.

[Mit15c] Massachusetts Institute of Technology (2015). MIT App Inventor. App Inventor 2 online. Disponible en <http://ai2.appinventor.mit.edu/>. Consultado el 15 de marzo de 2015.

[Oli13] Olimex Chile, (2013). IOIO para Android. Disponible en <http://www.olimex.cl/product_info.php?products_id=854>. Consultado el 20 de marzo de 2013.

[Pcm] Pc magazine (n.d.). Definition of:Bluetooth. Disponible en <http://www.pcmag.com/encyclopedia/term/38794/bluetooth>. Consultado el 30 de marzo de 2015.

[Rom07] Filein Rómmel (2007). Revista SG Buzz. SQLite: La Base de Datos Embebida. Disponible en <http://sg.com.mx/revista/17/sqlite-la-base-datos-embebida#.VRcvTvyG91Y>. Consultado el 30 de marzo de 2015.

[Spa13] SparkFun Electronics (2013). Disponible en <https://www.sparkfun.com/products/12633>. Consultado el 20 de marzo de 2015.

[Sql15] SQLite Organization (2015). Distinctive Features Of SQLite. Disponible en <http://www.sqlite.org/different.html>. Consultado el 30 de marzo de 2015.

[Tas] Tastets System (n.d.). Tecnología GPS. Disponible en <http://www.gps.cl/tecnologia/>. Consultado el 16 de marzo de 2015

[Tra84, 100] Trampe J. (1984). Mechanical Vibration and Shock Measurements. Disponible en <http://www.bksv.com/doc/bn1330.pdf>. Consultado el 30 de marzo de 2015.

[Uni03a] Universidad del País Vasco (2003). Micrófonos. Disponible en <http://www.ehu.eus/acustica/espanol/electricidad/micres/micres.html>. Consultado el 30 de marzo de 2015

[Uni03b] Universidad del País Vasco (2003). Altavoces. Disponible en <http://www.ehu.eus/acustica/espanol/electricidad/altaves/altaves.html>. Consultado el 30 de marzo de 2015.

[Xam15a] Xamarin (2015). Xamarin. Disponible en <http://xamarin.com/>. Consultado el 15 de marzo de 2015

[Xam15b] Xamarin (2015). Xamarin pricing. Disponible en <https://store.xamarin.com/>. Consultado el 15 de marzo de 2015

[Yes12] Yesid J. (2012). 5 lenguajes para programar en Android. Disponible en <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>. Consultado el 15 de marzo de 2015.

# ANEXOS

## Anexo A: Código de ejemplo SQLite en Basic4Android

Sub Process\_Globals

Dim Mi\_DB As SQL

End Sub

Sub Globals

Dim Mi\_Cursor As Cursor

Private EditText1 As EditText

Private Button1 As Button

Private ListView1 As ListView

End Sub

Sub Activity\_Create(FirstTime As Boolean)

Activity.LoadLayout("Principal")

If FirstTime Then

'Inicialización de la base de datos

Mi\_DB.Initialize(File.DirInternal,"testSQLite.db",True)

'Creación de una tabla

Mi\_DB.ExecNonQuery("CREATE TABLE IF NOT EXISTS Mi\_Tabla (id\_dato INTEGER PRIMARY KEY AUTOINCREMENT, dato TEXT)")

End If

End Sub

Sub Button1\_Click

If EditText1.Text<>"" Then

'Sentencia de Ingreso del dato del usuario a la base de datos

Mi\_DB.ExecNonQuery2("INSERT INTO Mi\_tabla (dato) VALUES (?)", Array As String(EditText1.Text))

'Cursor que contiene todos los datos de la tabla

Mi\_Cursor=Mi\_DB.ExecQuery("SELECT dato FROM Mi\_tabla ORDER BY dato ASC")

ListView1.Clear

EditText1.Text=""

If Mi\_Cursor.RowCount>0 Then

For i=0 To Mi\_Cursor.RowCount-1

Mi\_Cursor.Position=i

ListView1.AddSingleLine(Mi\_Cursor.GetString("dato"))

Next

End If

End If

End Sub

1. Subsecretaría de Telecomunicaciones (SUBTEL), en base a información proporcionada por los operadores [↑](#footnote-ref-1)
2. StatCounter GlobalStats [↑](#footnote-ref-2)
3. Figura obtenida de https://github.com/ytai/ioio/wiki/Getting-To-Know-The-IOIO-OTG-Board [↑](#footnote-ref-3)