

# **Arduino y Android una Pareja para Aplicaciones de Ubicuidad**

**Ferley Medina Rojas**

Universidad Cooperativa de Colombia Sede Neiva, Neiva, Huila, Colombia,  
ferley.medina@campusucc.edu.co

**Yhon Jerson Robles Puentes**

Universidad Cooperativa de Colombia Sede Neiva, Neiva, Huila, Colombia,  
yhon.jerson@campusucc.edu.co

## **ABSTRACT**

Operating system Android for mobile devices, tablets, phones, PDAs, and more. Using open source philosophy using the Linux kernel and hardware abstraction layer, ie, the kernel is used for the components of Android, the Dalvik machine, libraries and applications to run on a standardized environment not dependent on the hardware, and its graphical interface defined as an activity through which realizes the communication between the application and this.

With the use of programming in Android, Arduino board, his sketch, the mobile device, and means of connectivity, integrated are achieved lead to a solution that enables on-off control of the bulbs in a house conditions minimum, contributes to environmental sustainability and economic conditions of those who live by saving energy.

**Keywords.** Android, Arduino, minimum conditions of house, environment.

## **RESUMEN**

Android un sistema operativo para dispositivos móviles tablets, celulares, PDAs, entre otros. Con el uso de la filosofía de código abierto usando el kernel de Linux como capa de abstracción del hardware, es decir, el kernel se usa para que los componentes de Android, la máquina Dalvik, las librerías y las aplicaciones se ejecuten sobre un entorno estandarizado que no dependa del hardware, y su interfaz gráfica definida como una actividad mediante la cual realiza la comunicación entre la aplicación y este.

Con el uso de la programación en Android, la tarjeta Arduino, su sketch, el dispositivo móvil, y los medios de conectividad, integrados se logran dar origen a una solución que permite el control del encendido y apagado de los bombillos de una casa de condiciones mínimas, contribuye a la sostenibilidad del medio ambiente y a las condiciones económicas de quienes la habitan por el ahorro de energía.

**Palabras clave.** Android, Arduino, casa de condiciones mínimas, medio ambiente.

## **1. INTRODUCCIÓN**

El poder prender y apagar las luces de una casa de condiciones mínimas desde un dispositivo móvil le permite a la gente mejorar las condiciones del nivel de vida debido al ahorro en plata que puede tener cada usuario que adquiera este servicio por que puede tener el controles de las bombillas que están prendidas sin razón de estar, contribuyendo a mejorar el medio ambiente, por el uso racional de las bombillas utilizadas.

Con la tarjeta Arduino la cual tiene integrada los servicios de Bluetooth, micro controlador ATmega 328, programable en el software Arduino y los diez y seis pines de entrada y salida para controlar los dispositivos se logra construir el prototipo.

Con la programación en Android, el cual es un sistema de uso libre se puede tener una solución asequible a todos los usuarios de dispositivos móviles. Con el entorno de desarrollo Eclipse se permite la integración de todos los dispositivos físicos como lógicos combinados con la tecnología inalámbrica, para contribuir con nuevos desarrollos al uso de los dispositivos móviles.

## **2. MARCO TEÓRICO**

Se definen los siguientes conceptos:

### **2.1. COMPUTACIÓN UBIQUA**

También conocida como computación dominante, fue nombrada y popularizada por primera vez por Mark Weiser, quien pensando en el usuario final creó ambientes en donde están comprometidos los cálculos, y la comunicación de forma transparente, integrados con la inteligencia, la computación basada en las nubes, y la red de sensores para en el sitio donde se encuentra la persona interactúe con el medio que lo rodea. (Infosys , 2012)

### **2.2. ANDROID**

Un sistema operativo para dispositivos móviles tablets, celulares, PDAs, entre otros; desarrollado principalmente por Google con el uso de la filosofía de código abierto, por lo tanto, cualquier persona puede descargar el código fuente, modificarlo dependiendo las necesidades y compartir los cambios con la comunidad.

En el año de 2008 se iniciaron las primeras versiones llegando en Abril de 2009 la versión de Android 1.5(CupCake) empezándose a popularizar, teniendo para Mayo de 2012 la versión 4.2(Ice Cream Sandwich) en Jelly Bean, la cual permite tomar fotos en ángulos de 360 (Photo Sphere), escribir deslizando el dedo entre las teclas en la pantalla y que se complementa con un predictor de palabras, y con la retransmisión de contenidos de nuestro equipo a otros dispositivos como la televisión (Gesture Typing) entre otras. (Gómez, 2012)

#### **2.2.1. CARACTERÍSTICAS DE ANDROID**

Usa el kernel de Linux como capa de abstracción del hardware, es decir, el kernel se usa para que los componentes de Android, la máquina Dalvik, las librerías y las aplicaciones se ejecuten sobre un entorno estandarizado que no dependa del hardware (Felix, 2012), con framework de aplicaciones que permite la reutilización y remplazo de componentes, navegador Web integrado basado en el motor de renderizado Webkit,, gráficos optimizados a través de la biblioteca de gráficos 2D y 3D de la OpenGL ES 1.1., almacenamiento de datos en BBDD SQLite, soporta formatos de archivos, MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG y GIF, conectividad, telefonía GSM, Bluetooth, EDGE, 3G y WIFI, soporta dispositivos adicionales, cámara de video, pantalla táctil, GPS, entorno de desarrollo, emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE. (Rodríguez, 2012)

#### **2.2.2. ARQUITECTURA DE ANDROID**

Android tiene la siguiente estructura:

Aplicaciones, como correo electrónico, programas SMS, calendario, mapas, navegador y contactos entre otros.

Marco de trabajo de aplicaciones, permite la reutilización de componentes, debido a que cualquiera aplicación puede publicar sus capacidades y cualquiera otra aplicación puede luego hacer uso de esas capacidades, encontrando gestor de actividades, de ventanas, de paquetes, de telefonía, de recursos, de localización, proveedor de contenidos, de notificaciones, vista del sistema y servicios de XMPP.

Librerías, están conformadas por un conjunto de bibliotecas de C/C++, administrador de superficies, medios de framework, SQLite, Open GLjES, FreeType, Webkit, SGL, SSL, LBC.

Runtime, conformado por el núcleo de librerías basadas en Java y la máquina virtual, Dalvik, permitiendo que cada aplicación Android corra su propio proceso con su propia instancia.

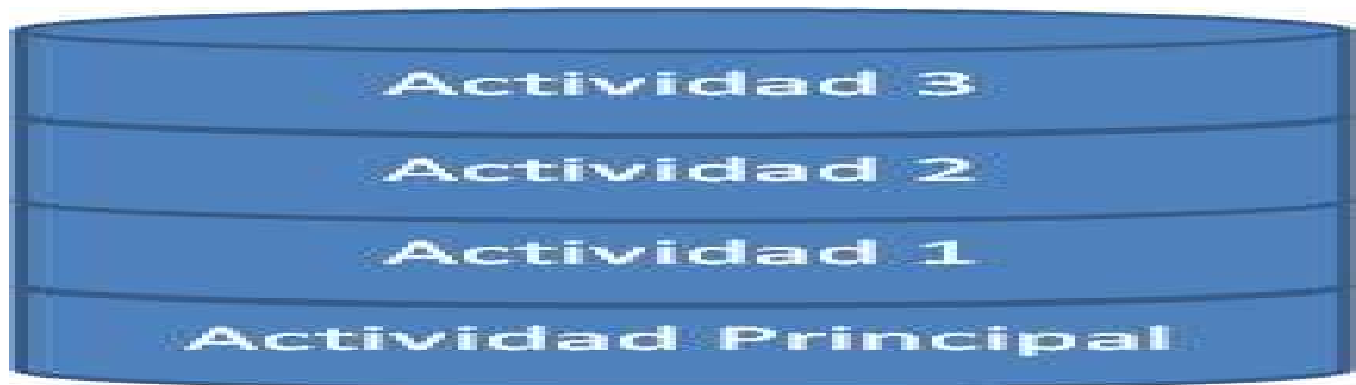
Kernel de Linux, suministra los servicios básicos de controladores de, pantalla, cámara, Bluetooth, memoria flash, carpetas, USBm, teclados, tarjetas inalámbricas, audio y de encendido. (Tómas Gironés, 2010)

### 2.2.3. DALVIK

Es la máquina virtual utilizada por Android, diseñada y escrita por Dan Bornstein con apoyo de ingenieros de Google, la cual corre por encima del kernel de Linux 2.6, delegando las tareas relacionadas con la gestión de hilos y memoria de bajo nivel, usada para que haya múltiples instancias funcionando con un impacto muy bajo en el rendimiento de memoria del dispositivo por su ejecución mapeable. Debido a que su funcionamiento está basado en registros puede ejecutar las clases generadas por el compilador Java las cuales son posteriormente convertidas en formato nativo mediante el uso de la herramienta dx, haciéndolo de forma automática al ejecutar un servicio o aplicación. (AndroidDevMX, 2011).

### 2.2.4. ACTIVIDAD

Se define como una ventana mediante la cual se hace la interfaz gráfica al usuario, permitiendo la comunicación entre la aplicación y este. (Báez, 2007) Los elementos se guardan en ficheros XML, que contiene los elementos en la pantalla (layout), los botones, los textos, los checkbox y demás elementos, almacenándose en ./res/layout, para ser tratados en la clase NameActivity.class.



**Figura 1: Pila de Actividades Android**

La Fig. 1, muestra la pila de actividades conformada de acuerdo al orden de su invocación, las cuales se van terminando cuando la de encima ha finalizado o cuando el sistema las destruye por necesidad de memoria, para ello el sistema evalúa la actividad de mayor tiempo parada, en caso de mayor requerimiento de memoria, y si no está en el foco la elimina de completo, a excepción de la actividad principal.

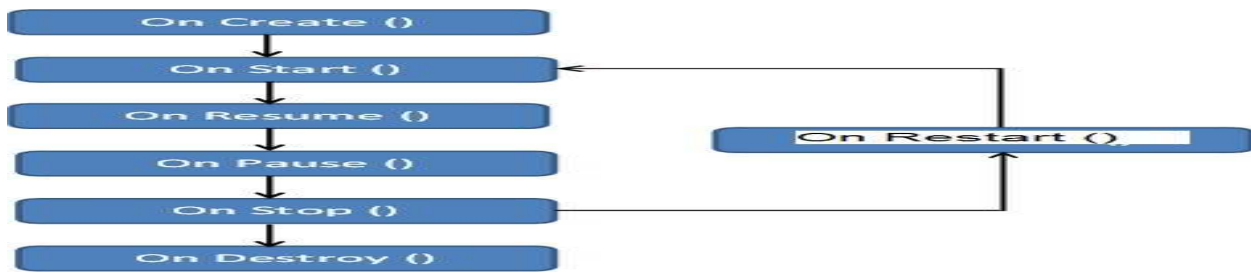
### CICLO DE VIDA DE UNA ACTIVIDAD

El ciclo de vida de una actividad está compuesto de tres estados:

Activo, la actividad se está ejecutando y es la principal.

Pausado, la actividad se está ejecutando, es visible, no es la principal, se guarda la información para prevenir una posible pérdida de los datos en caso de que el sistema decida prescindir de ella para liberar memoria.

Parado, la actividad no está visible al usuario, está detenida y el sistema la puede liberar para tener mayor memoria disponible. En caso de volverla a necesitar se reiniciará desde el comienzo.



**Figura 2: Ciclo de vida de una actividad**

En la fig. 2, se muestra los diferentes métodos por los cuales puede pasar una actividad.

On create (), crea la actividad, y recibe el parámetro del estado de la actividad anterior con el objetivo de preservar la información.

On Star (), se ejecuta después de On create () o On Restart () según corresponda.

On Resume (), se ejecuta cuando la actividad está en primer plano y establece el inicio de la interactividad entre el usuario y la aplicación.

On Pause (), se ejecuta para que la actividad no este en primer plano, dando paso a otra actividad.

On Stop (), la actividad pasa un largo periodo de inactividad.

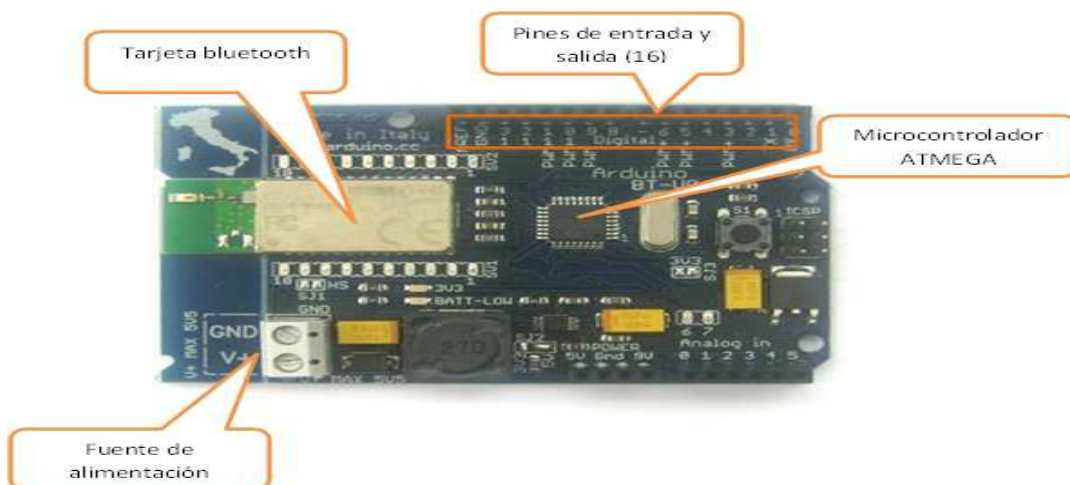
On Destroy (), se utiliza para finiquitar una actividad, se debe hacer cuando la actividad ya no es necesaria.

On Restart (), cuando la actividad se encuentra en la pila de actividades la reinicia, partiendo de cero.

### 2.3. TARJETA DE CONEXIÓN DE NODOS

La tarjeta Arruino BT.V06, es una tarjeta micro controladora basada en Tamegua 328 Italiana de 1KB, memoria flash de 16KB, velocidad de reloj de 16MHz., alimentador de energía de 5 a 20 voltios. Comunicación serial con pines digitales o (RX), (TX), USB y controladores FTDI, con 16 pines de entrada y salida, uso de tecnología Bluetooth.

La programación se realiza con el software Arruino, él cual esta basado en Java, tiene un entorno con editor de texto para escribir código, una área de mensajes, consola de texto, barra con botones de funciones y una series de menees. Los sketch, son los archivos con los programas en código fuente que se copian al micro controlador para realizar la tarea planeada en la tarjeta; el conjunto de los sketch forman el sketchbook. (Arduino, 2012).



### **Figura 3: Tarjeta Arduino**

En la Fig. 3, al lado izquierdo en la parte inferior se encuentra la fuente de alimentación, en la parte superior la tarjeta bluetooth, hacia la derecha los pines digitales para control de entradas y salidas y en el centro el micro controlador Atmega 328.

#### **2.4. CASA DE CONDICIONES MÍNIMAS**

Una casa de condiciones mínimas está dotada de los servicios públicos básicos, agua, energía eléctrica y gas; con dos habitaciones, sala comedor, cocina y una batería sanitaria con lo cual se estaría hablando de aproximadamente seis bombillo. Su precio no debe pasar de los ciento treinta y cinco salarios mínimos legal vigente, debe tener fácil accesibilidad, adecuación cultural y una habitabilidad digna. (Fedevivienda, 2009)

#### **2.5. DISPOSITIVO MÓVIL**

Se utiliza para ejecutar las aplicaciones realizadas en Andnroid y él envió de las señales a la tarjeta Arduino para el encendido y apagado de los bombillos. El Samsung Galaxy Ace con banda GSM y Edge 850 de 900/1800/1900Mhz, con sistema operativo Android 2.2, conectividad Bluetooth, USB 2, WIFI, GPS, almacenamiento masivo (Samsung, 2012)

#### **2.6. APP INVENTOR**

Software de programación orientado a los eventos, de forma visual, permitiéndole al usuario elaborar aplicaciones para Android mediante el entrelazado de bloques, hecho en Java, diseñada por Google, disponible en la Web de entorno on line de forma gratuita. (Massachusetts, 2012) Requiere para su funcionamiento la instalación del JDK versión 7 y del entorno app inventor.

### **3. CONSTRUCCIÓN DEL PROTOTIPO**

Para la construcción de prototipo fue necesario tener en cuentas las siguientes etapas:

#### **3.1. MODELAMIENTO DE LA APLICACIÓN**

El modelamiento de la aplicación se elaboró con herramientas UML.

##### **3.1.1. DIAGRAMA DE CASO DE USO**

En la Fig. 4, el diagrama de caso de uso, con un actor llamado usuario, quien es él que utiliza la aplicación de encendido y apagado de los bombillos de la casa de condiciones mínima e interactúa con los siguientes procesos:

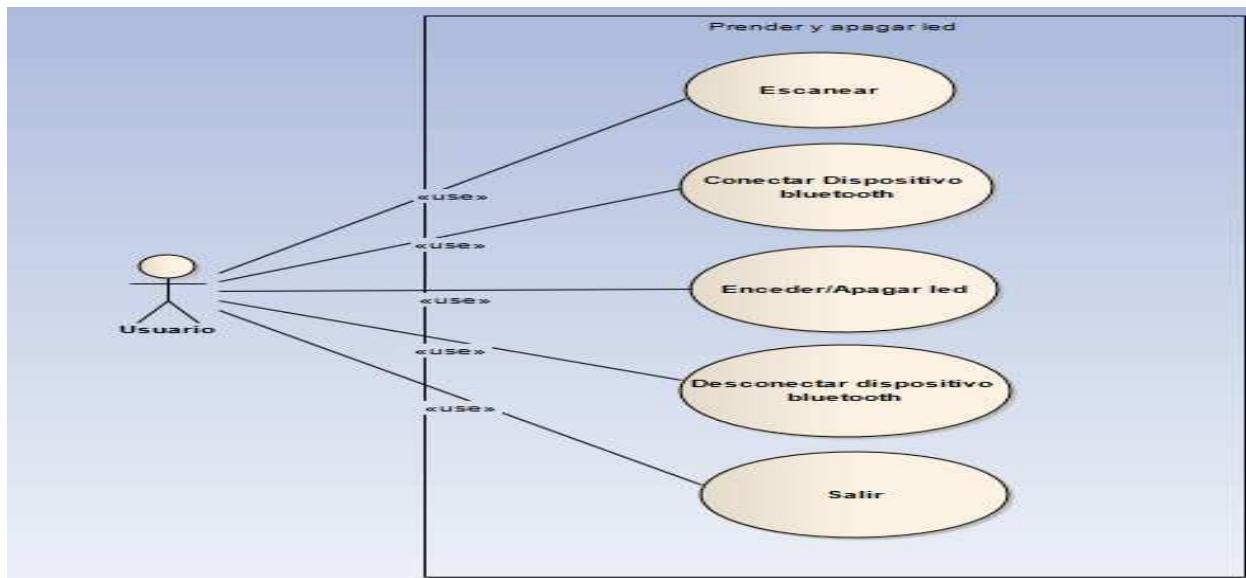
Escanear, busca los dispositivos con bluetooth disponible para lograr una conectividad.

Conectar dispositivos bluetooth, como producto del escaneo se encuentra la tarjeta Arduino que tiene la MAC 00:06:66:44:F7:31 lográndose la conexión entre los dos dispositivos para poder interactuar con el prototipo.

Encender/Apagar, conectada la tarjeta Arduino mediante el dispositivo bluetooth se puede encender y apagar el Led requerido, el cual simula a un bombillo, del uno al seis que son el máximo de bombillos que tiene una casa de condiciones mínimas una vez se ha realizado su entrega.

Desconectar dispositivos bluetooth, terminar la conexión con la tarjeta Arduino, quedando el prototipo desconectado, evitando el consumo de energía.

Salir, sale de la aplicación, revisando que la conexión con la tarjeta Arduino esté terminada, evitando la interacción con el prototipo.



**Figura 4: Diagrama de caso de uso**

### 3.1.2. DIAGRAMA DE CLASES

La Fig. 5. describe el diagrama de clases de la aplicación mostrando las siguientes:

Button, elaborar cada uno de los seis botones utilizando el método SetEnable que están asociados a los seis Led que simulan los bombillos del prototipo.

BluetoothCliente, permite el encendido y apagado de cada uno de los seis Led utilizando los métodos Conect y Disconnect.

Label, determina si la etiqueta esta visible u oculta, para permitir la visualización del botón de ayuda.

Activity, es la actividad principal de la aplicación donde se muestran los seis botones que van a prender y apagar los Led con los métodos de:

Apagar, para terminar la interacción con él sistema.

Conectar, al estar presionado por más de cinco segundos el botón el cambia su etiqueta pasándola a desconectar.

Desconectar, al estar presionado por más de cinco segundos el botón el cambia su etiqueta pasándola a desconectar.

Encender, cheque el sistema, si este está encendido lo deja quieto en caso contrario procede a realizar el encendido del prototipo verificando que previamente exista la conexión entre el dispositivo móvil y la tarjeta Arduino.

Salir, termina la conexión con la tarjeta Arduino en caso de estar conectada, y sale del aplicativo.

Mostrar ayuda, despliega un mensaje indicando la funcionalidad del prototipo e interactúa con los métodos de apagar, conectar, desconectar, encender y salir.

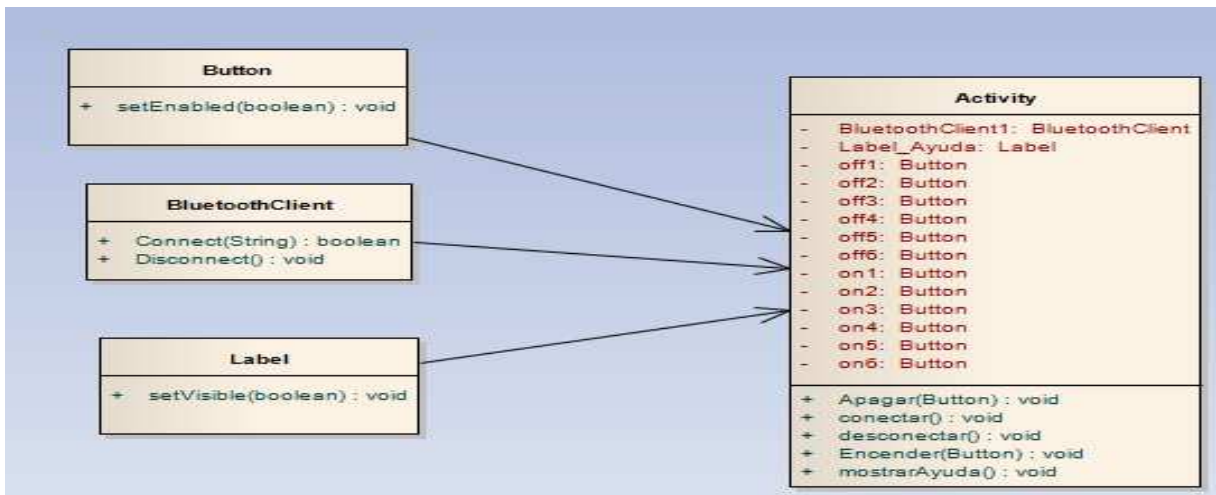


Figura 5: Diagrama de Clases

### 3.2. REQUERIMIENTOS TÉCNICOS

Para la elaboración del prototipo se requiere el uso de una tarjeta Arduino Bt.V6 ATmega 328, la cual tiene un sistema de conexión bluetooth para permitir la conectividad entre el dispositivo móvil y la tarjeta. Una serie de pines (diez y seis) mediante el cual se hace el control de las señales que entran y salen del prototipo que permite el encendido y apagado de los Led (bombillos). Un micro controlador, que permite instalar el sketch (programa de control de los pines).

El sketch es el programa (elaborado con el software Arduino versión 1.01. que se instala en el micro controlador para la activación de los seis pines digitales desde el número, 8 al 13, los cuales va a recibir las señales de encendido o de apagado. Utilizando los caracteres de la “a” la “f” para encendido y de la “g” a la “l” para el apagado, relacionando el carácter a para el pin 8 de encendido y la g para el apagado, luego el carácter b para el pin 9 de encendido y el carácter h para el apagado, y así sucesivamente hasta terminar con las letras y los pines mencionados. El sketch se debe copiar al micro controlador de la tarjeta Arduino mediante el uso de la interfaz serial (com3).

Un dispositivo móvil con sistema operativo Android, para instalar el programa elaborado en Adroid, el cual hará la interfaz entre el usuario y el prototipo.

La programación en Android se realizó con el software app inventor para lo cual se requiere tener en la máquina local (computador) el JDK versión 7 el app inventor, se trabaja on line y está disponible en la dirección <http://appinventor.mit.edu/>, en la cual se desarrollaron e implementaron las siguientes actividades:

#### ACTIVIDAD PRINCIPAL



## **Figura 6: Actividad principal**

En la Fig. 6, de la actividad principal se muestran los botones de conectar, salir, ayuda, led1, led2, led3, led4, led5 y led6.

### **ACTIVIDAD DE CONECTAR**

Al pulsar el botón de conectar que esta en la Fig. 6, el sistema determina los dispositivos bluetooth disponible y se conecta con la tarjeta Arduino utilizando la MAC como proceso de validación, cambiando la etiqueta a desconectar.

### **ACTIVIDAD DESCONECTAR**

Al tener oprimido durante cinco segundo el botón descrito en la actividad de conectar, este desconecta la tarjeta Arduino del sistema, pasando de nuevo la etiqueta conectar.

### **ACTIVIDAD SALIR**

Botón que se muestra en la Fig. 6, al presionarlo permite salir de la aplicación. En caso de estar conectada la tarjeta realiza primero la desconexión y luego si la terminación del aplicativo.

### **ACTIVIDAD AYUDA**

Al presionar el botón de ayuda que se muestra en la Fig. 6, despliega una etiqueta indicando el fin de la construcción de la aplicación.

### **ACTIVIDAD DE BOTONES**

Con esta actividad se apagan y se prenden los botones (Led) mostrados en la Fig. 6.

Una protoboard, para instalar la fuente de poder, los seis Led y los seis cables, los cuales son usados para conectar los Led con los pines de la tarjeta Arduino.

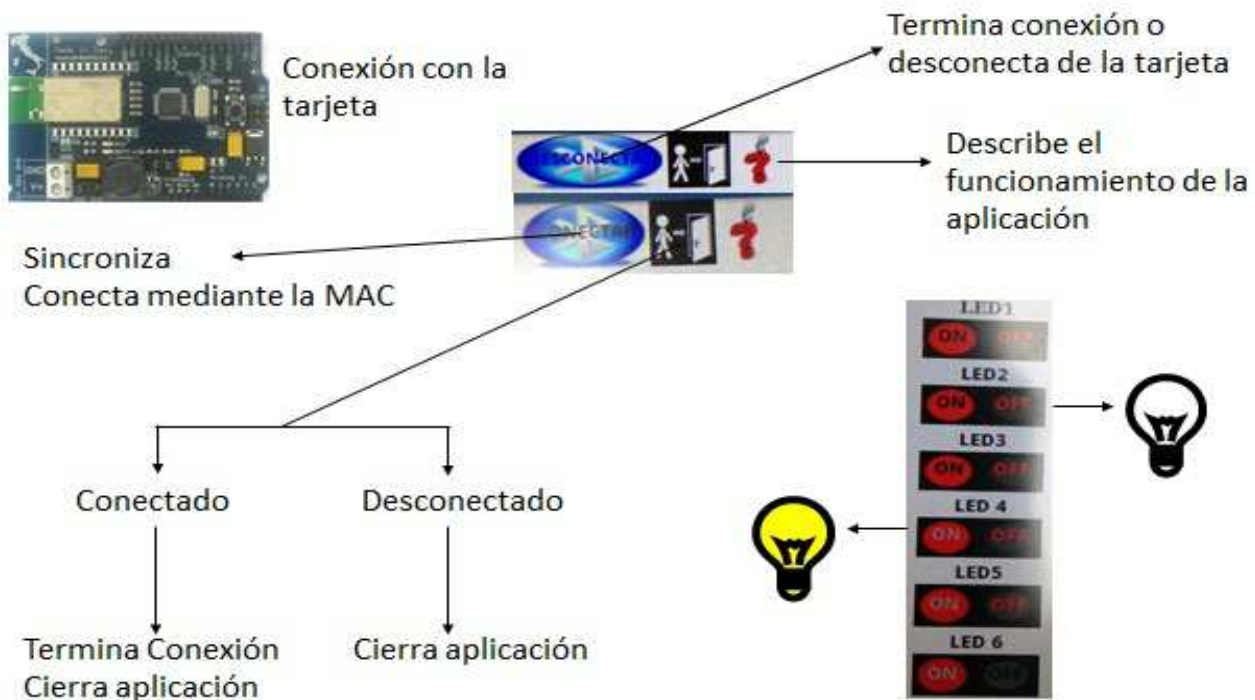
## **4. IMPLEMENTACIÓN DEL PROTOTIPO**

La implementación del prototipo se hace instalando en la protoboard, la fuente de poder, los seis Led, los seis cables que conectan a los Led con los pines de la tarjeta Arduino, la transferencia del programa de Android al dispositivo móvil.

Observando la figura 7, se encuentra el botón conectar el cual al ser presionado hace la sincronización del dispositivo móvil con la tarjeta Arduino dando origen a la conexión. El botón desconectar al ser presionado termina la conexión del dispositivo móvil con la tarjeta Arduino. Al presionarse el icono de salir este verifique cual es el estado de conexión del dispositivo móvil con la tarjeta Arduino, si está conectado lo desconecta y sale del programa, en caso contrario sale del programa. El icono de ayuda muestra un mensaje indicando cual es la funcionalidad del programa.

Luego se encuentra otro icono el cual tiene los Led del uno al seis con las opciones de on, para encender el Led y el de off, para apagar el Led. Por ello el bombillo de encendido a la izquierda al lado del on y el bombillo apagado a la derecha al lado del off.





**Figura 7: Diagrama de la aplicación**

El software desarrollado para la implementación del prototipo descrito en la Fig. 7, esta bajo la licencia Copy Left de los autores.



**Figura 8: Implementación del prototipo**

En la Fig. 8, como se describe el funcionamiento del prototipo, observándose la protoboard donde se encuentran los led, la tarjeta Arduino (control de los leds, mediante el sketch) y el dispositivo móvil (programación Android) de donde se interactúa con el encendido y apagado de los leds.

Con la construcción e implementación de este prototipo se le pueden agregar los controles de potencia para hacer posible el prender y apagar las bombillas, abrir y cerrar las puertas, encender y apagar electrodomésticos de una casa de condiciones mínimas, además de diferentes equipos y motores de una industria determinada, contribuyendo al ahorro de energía y a una solución de ubicuidad que contribuye al mejoramiento de la calidad de vida de los seres humano.

## 5. CONCLUSIONES

A la construcción e implementación de este prototipo se le pueden agregar los controles de potencia para hacer posible el prender y apagar las bombillas, abrir y cerrar las puertas, encender y apagar electrodomésticos de una casa de condiciones mínimas, además de diferentes equipos y motores de una industria determinada.

Los sketch, son programas que se construyen en Java en el software Arduino para la administración de los diferentes accesorios que tiene la tarjeta Arduino, permitiendo el control de los dispositivos deseados conectados.

Con la programación de este prototipo en Android implementada en dispositivos móviles, se contribuye al desarrollo de aplicaciones de computación ubicua, facilitando el que hacer de las tareas diarias de los usuarios involucrados.

## REFERENCIAS

- AndroidDevMX. (15 de 11 de 2011). *Java y la máquina virtual Dalvik*. Recuperado el 01 de 11 de 2012, de <http://www.androiddevmx.net/java-y-la-maquina-virtual-dalvik>
- Arduino. (2012). *A reference to past and present Arduino boards, shields and accessories*. Recuperado el 20 de 10 de 2012, de <http://arduino.cc/en/Main/Boards>
- Báez, M. Á. (01 de 10 de 2007). *Introducción a Android*. Obtenido de <http://www.ucm.es/info/tecnomovil/documentos/android.pdf>
- Fedevivienda. (29 de 01 de 2009). *Mesa-VIS Enero 29 de 2009*. Recuperado el 01 de 10 de 2012, de <https://mesavis.uniandes.edu.co/presentaciones%202009/Present%20FEDEVIVIENDA%20enero%202009.pdf>
- Felix, J. F. (03 de 2012). *Plataforma domótica basada en la integración de*. Recuperado el 17 de 11 de 2012, de [http://www.iiis.org/CDs2012/CD2012SCI/CISCI\\_2012/PapersPdf/CA326YQ.pdf](http://www.iiis.org/CDs2012/CD2012SCI/CISCI_2012/PapersPdf/CA326YQ.pdf)
- Gómez, J. (30 de 10 de 2012). *Características de la nueva versión Android 4.2*. Recuperado el 27 de 11 de 2012, de <http://www.adslzone.net/article9822-caracteristicas-de-la-nueva-version-android-42.html>
- Infosys . (01 de 02 de 2012). *Construyendo la Empresa de l Mañana*. Recuperado el 15 de 11 de 2012, de [www.infosys.com/mexico/spanish/building-tomorrows-entreprise/pages/pervasive-computing.aspx](http://www.infosys.com/mexico/spanish/building-tomorrows-entreprise/pages/pervasive-computing.aspx)
- Massachusetts. (02 de 09 de 2012). *Mit App Inventor Teach*. Recuperado el 01 de 12 de 2012, de <http://appinventor.mit.edu/teach/>
- Rodriguez, D. B. (09 de 01 de 2012). *Desarrollo de un videojuego 3D para Android*. Recuperado el 15 de 11 de 2012, de <http://upcommons.upc.edu/pfc/bitstream/2099.1/14100/1/77795.pdf>
- Samsung. (12 de 01 de 2012). *Galaxy Ace Negro*. Recuperado el 01 de 11 de 2012, de <http://www.samsung.com/co/consumer/mobile-phones/mobile-phones/smart-phone/GT-S5830OKLCOM-spec>
- Tómas Gironés, J. (2010). *El gran libro de Android*. Barcelona: Marcombo S.A.

## AUTORIZACIÓN Y RENUNCIA

Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito