

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01- 27

# **INTEGRACIÓN DE SISTEMAS CON HARDWARE Y SOFTWARE LIBRE PARA EL DISEÑO DE UNA CASA INTELIGENTE PARA PERSONAS CON DISCAPACIDAD**

Andrés Sánchez Espinosa

Jorge Machado Cortés

Tecnología en Electrónica

Director(es) del trabajo de grado

Andrés Felipe Tobón

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**Fecha**

2016

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RESUMEN

---

Este trabajo se realizó para mejorar la calidad de vida de las personas con discapacidad, en especial aquellas que permanecen mucho tiempo solas en casa y se les dificulta realizar ciertas actividades como encender y apagar las luces de diferentes lugares de la casa; abrir la puerta del garaje; la puerta principal, encender el aire acondicionado. Dicho trabajo se materializó mediante la creación de una aplicación android instalada en un Smartphone que se comunica por medio de un módulo bluetooth a una placa Arduino uno®. Esta placa es la encargada de activar-desactivar los elementos eléctricos. También se logra un ahorro energético al poder apagar oportunamente los diferentes dispositivos cuando no es necesario tenerlos encendidos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01- 22

## RECONOCIMIENTOS

---

Expresamos los sinceros agradecimientos a:

A nuestras familias que pese a grandes obstáculos que se presentaron durante la carrera siempre estuvieron ahí para apoyarnos.

Al instituto tecnológico metropolitano por su acompañamiento en este proceso de aprendizaje.

A los profesores, por haber compartido todo ese conocimiento fundamental para la formación académica necesaria para la realización de este proyecto; por la motivación y colaboración.

Al asesor Andrés Tobón por su gran ayuda y amabilidad en el momento de resolver dudas para lograr la materialización del proyecto.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## ACRÓNIMOS

---

MIT: Massachussets Institute of Technology.

App: Aplicaciones.

PWM: Pulse-width modulation (modulación por ancho de pulso).

MAC: Media Access control (control de acceso al medio).

API: Interfaz de Programación de Aplicaciones

WIFI: Wireless Fidelity (Fidelidad inalámbrica).

TCP/IP: Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet).

KNX: Konnex(es un protocolo de comunicaciones destinado a la domótica).

LED: Light Emitting Diode (diodo emisor de luz)

OSI: Open System Interconnection(es un modelo de referencia para los protocolos de la red de arquitectura en capas).

HTML: HyperTextMarkupLanguage (lenguaje de marcas de hipertexto) hace referencia al lenguaje de marcado para la elaboración de páginas web.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## TABLA DE CONTENIDO

1.	INTRODUCCIÓN.....
2.	MARCO TEÓRICO.....
3.	METODOLOGÍA .....
4.	RESULTADOS Y DISCUSIÓN .....
5.	CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO .....
6.	REFERENCIAS .....
7.	APÉNDICE.....

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# 1. INTRODUCCIÓN

---

Este proyecto hizo un aporte a las personas con discapacidad para mejorar su calidad de vida de una manera muy eficiente y segura ya que puede activar-desactivar: las luminarias, puerta principal, puerta de garaje y ventilación desde cualquier lugar de la casa sin recibir ayuda de otra persona evitando así movimientos y esfuerzos innecesarios que pueden resultar peligrosos o fatigantes para la persona con discapacidad. Estas personas van a poder estar más cómodas e independientes, alcanzado el objetivo general de integración de sistemas con hardware y software libre para el diseño de una casa inteligente para personas con discapacidad.

El mejoramiento de la calidad de vida de las personas discapacitadas es algo en los que todas las personas deben aportar y por medio de este proyecto de un sistema domótico, se hizo la tarea de implementar una maqueta como ejemplo de una casa domótica para que pueda ser materializado en cualquier momento por cualquier persona que esté interesado en mejorar la calidad de vida.

Esta maqueta consta del control de iluminarias, ventilación, puerta del garaje y la puerta principal, las cuales serán controladas mediante un Smartphone que tenga bluetooth, esto se realizó por medio de la integración de software y hardware libre: App inventor2® y Arduino uno®.

El proyecto deja un inicio para que en un futuro sea cada vez más amplio y eficaz ya que tiene la posibilidad de crecer según las necesidades del usuario. En caso de que se cuente con una casa más grande se puedan hacer las modificaciones necesarias para un mayor número de actuadores.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2. MARCO TEÓRICO

---

El proyecto consta de una maqueta (ver Imagen 35). Esta representa una casa con sus divisiones que permite mostrar el control automático de la iluminación de: sala, cocina, baño, dos alcobas, corredor, un garaje, apertura de la puerta principal, apertura y cierre del garaje, activación de un ventilador que simula un aire acondicionado y un sensor (Im35) que mide la temperatura que luego se visualiza en la aplicación android, según (Javale D, &Mohsin M, &Nandanwar S, &Shingate M, 2013,p.383)Android es un software para dispositivos móviles que incluye un sistema operativo basado en Linux. Esta maqueta recibe en total 14 señales digitales (salidas) y una entrada analógica para recibir la señal del sensor de temperatura Im 35. Estas señales de control provienen de los relés activados por la placa Arduino uno® que consta de 20 pines que se pueden configurar como entrada o salidas digitales, los cuales 6 son entradas analógicas, 6 salidas PWM, 2 de comunicación serial. La alimentación de la placa es de 9 a 12 Vdc, su entorno de programación basado en lenguaje C. El control de la temperatura se realizó con el circuito integrado Im35 según la hoja de datos editado por Texas instruments (2015) este es un sensor de temperatura con una precisión calibrada de 1 °C. Su rango de medición abarca desde -55 °C hasta 150 °C. La salida es lineal y cada grado Celsius equivale a 10 mV, por lo tanto:

$$150 \text{ }^{\circ}\text{C} = 1500 \text{ mV}$$

$$-55 \text{ }^{\circ}\text{C} = -550 \text{ mV}$$

Para visualizar la temperatura en el celular es necesario escalizar la señal de salida del Im 35 con la siguiente fórmula:

$$x = \frac{\frac{A_o}{C} \times v}{R}$$

Dónde:

**X** es la señal escalizada.

**Ao** es la lectura del pin analógico del microcontrolador.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

**C** es la resolución de conversor analógico digital en este caso 1023 bytes contando desde el cero como primera posición.

**V** la tensión de trabajo del microcontrolador en este caso 5 voltios.

**R** la resolución del sensor Im35 que es 10mV/°C.

Quedando definida en el código del microcontrolador o Arduino uno:

`temp= (lectura_Im*100*5)/1023.0`

Se usaron siete señales digitales de salida para la iluminación, cuatro para la puerta del garaje (giro izquierda y giro derecha de un motor paso a paso), dos para la solenoide de la puerta principal y otra para el ventilador que representa el aire acondicionado, adicionalmente se utiliza una entrada analógica (Ao) que mide la temperatura de la casa por medio del sensor Im35 y es visualizada en la aplicación android.

El control se realizó por medio del software libre (appinventor2®) y hardware libre (placa Arduino uno®).

El appinventor2® es una plataforma de licencia gratuita elaborada por google, y sus librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License). Según (Hak. J. Kim y Jonathan Modell, 2012, p.1) Permite a los usuarios desarrollar aplicaciones para los teléfonos Android utilizando un navegador web. En esta plataforma se realizó una aplicación android, la cual puede controlar a la placa Arduino uno® por medio de un módulo bluetooth que facilita la comunicación serial RS-232 (protocolo de comunicación) entre estos dos dispositivos, la aplicación se instaló en un Smartphone el cual consta del sistema operativo android y tiene bluetooth.)

El App inventor2® se programa usando bloques de programación. Estos bloques están hechos con elementos comunes a la mayoría de los lenguajes de programación existentes. Se colocan bloques para construir bucles, condiciones, variables, etc. que permiten pensar lógicamente y solucionar los problemas de forma metódica. La creación de aplicaciones se realiza a través de una plataforma en línea sin necesidad de instalar un programa en el computador, y podemos ir probando su comportamiento en el Smartphone utilizando otra aplicación creada por App inventor2® llamada MIT AI2 Companion®. Una vez creada la aplicación podemos descargarla a nuestro Smartphone o tableta.

El módulo bluetooth HC-05 V2.0 (ver imagen 1) posee las siguientes características: Tensión de alimentación 5V, nivel lógico de comunicación serial es 3.3v, Temperatura

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

de trabajo: -20 °C a +75 °C, Frecuencia: 2.4GHz banda ISM con un alcance aproximado de 10 metros. Por medio de los comandos AT se puede cambiar la configuración, por ejemplo cambiar la contraseña, el nombre, la velocidad de transmisión, trabajar como esclavo o como maestro etc. El módulo bluetooth HC-05 consta de 34 pines de los cuales sólo 6 pines son de fácil acceso para uso en protoboard estos son: VCC, GND, TX, RX, STATE. Además posee un regulador interno que permite su alimentación de 3.6 a 6V (ver imagen 2).El módulo bluetooth entrega los datos a la placa Arduino uno® a través del protocolo serial RS-232.

### Características:

Compatible con el protocolo Bluetooth V2.0.

Voltaje de alimentación: 3.3VDC – 6VDC.

Voltaje de operación: 3.3VDC

Baud rate ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

Baud rate por defecto: 9600

Tamaño: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)

Corriente de operación: < 40 mA

Corriente modo sleep: < 1mA

Imagen 1. Modulo bluetooth HC-05 para protoboard

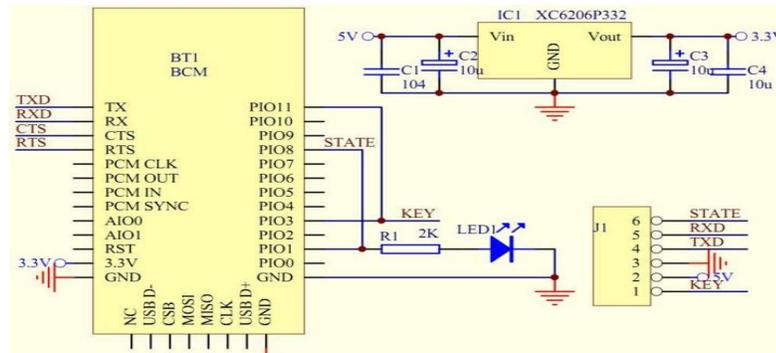


Itead, Studio (2010). Ilustración módulo HC-05. Recuperado de <https://tinyul.com/pz3jrja>

En la Imagen 1, podemos observar la apariencia real del módulo bluetooth HC-05 mostrando los pines de conexión para hacer la comunicación con la placa Arduino uno®.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Imagen 2. Diagrama del módulo bluetooth HC-05



Itead, Studio (2010).ilustración módulo HC-05. Recuperado de <https://tinyul.com/pz3jrja>

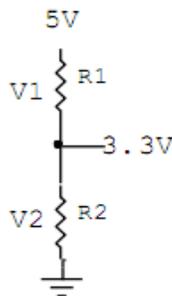
El modulo bluetooth trabaja a 3.3v en sus señales de transmisión de datos (TX) recepción de datos (Rx), la tarjeta del Arduino uno® trabaja a 5v por lo que es necesario hacer un divisor de voltaje para garantizar que sean 3.3v y no 5v los que lleguen al módulo en su pin de (RX) evitando que se dañe. En su pin de salida (Tx) no es necesario ya que este se conecta al de entrada del Arduino uno® (Rx) y reconoce los 3.3v hasta 5v como señal lógica. según (itead studio, 2010).

Hay varias formas de hacer esta modificación pero se optó por un divisor resistivo ya que es fácil y económico de elaborar con elementos que están a la mano, además dió buenos resultados sin problemas desde el inicio. Para poder utilizar un divisor resistivo con señales lo primero que tenemos que tener en cuenta es a quien vamos a conectar la salida del divisor, ya que si tiene una impedancia en el mismo orden de magnitud (o inferior) que la resistencia de salida el paralelo de ambas hará que los cálculos que hayamos realizado estén mal.

Se requiere realizar un divisor de tensión que tenga una entrada de 5V y una salida de 3.3V, con una impedancia de salida que no sea muy alta (ver imagen 3).

Imagen 3. Diagrama del divisor de tensión.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



$$V2 = \frac{R2}{R1+R2} * V \quad V2 = 3.3 \quad R2=10k\Omega$$

$$3.3V = \frac{10k\Omega}{R1 + 10k\Omega} * 5$$

$$0.66 (R1+10k\Omega) = 10k\Omega$$

$$R1 = \frac{10k\Omega}{0.66} - 10k\Omega$$

$$R1=5.15K\Omega$$

Llevándolo a resistencias comerciales queda R1 de 5kΩ y R2 de 10kΩ dando un voltaje de salida de 3.3v.

Este arreglo se debe de conectar de la salida TX del Arduino uno® hacia la entrada RX del módulo bluetooth.

La placa Arduino uno® es una placa con un micro controlador de la marca Atmel y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB conectado a un módulo adaptador USB-Serie que permite programar el micro controlador desde cualquier PC. Dispone de 20 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V. También dispone de entradas y salidas analógicas. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM. Tiene un lenguaje de programación propio basado en lenguaje C. (Arduino, 2015)

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de un quemador de microcontrolador.

También cuenta con su propio software (entorno de programación) que se puede descargar de forma gratuita en la página oficial <https://www.arduino.cc> donde incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

El entorno de Arduino se puede ampliar mediante el uso de las librerías, al igual que la mayoría de las plataformas de programación. Las librerías proporcionan funcionalidad adicional para uso en bocetos, por ejemplo, trabajar con el hardware o la manipulación de los datos. Para utilizar una librería en un boceto, selecciónelo de Sketch>Import Library. Un número de librerías viene instalado con el IDE, pero también se puede descargar o crear uno propio.

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and ComputationGroup del MIT Media Lab dirigido por John Maeda.

Wiring fue creado para facilitar la creación de objetos y espacios interactivos, en vista de lo frustrante y difícil que podía ser producir este tipo de aplicaciones. Es un proyecto abierto, iniciado por el diseñador y artista Hernando Barragán, en el Interaction Design Institute I'verea, en Italia, y ahora desarrollado en la Facultad de Arquitectura y Diseño de la Universidad de los Andes, en Colombia. Según (Hernando Barragán, 2011, p.156) Wiring permite escribir software para manipular dispositivos conectados a la plataforma electrónica para crear objetos y espacios interactivos que pueden tener la capacidad de sentir y modificar el ambiente.

**Motor paso a paso:**

Se trabajó con un motor paso a paso 28BYJ-48 (ver imagen 4.) para abrir y cerrar la puerta del garaje para tener una mayor precisión.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Un motor paso a paso es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control. El motor paso a paso se comporta de la misma manera que un conversor digital-analógico (D/A) y puede ser gobernado por impulsos procedentes de sistemas lógicos.

Este motor presenta las ventajas de tener precisión y repetitividad en cuanto al posicionamiento dentro del funcionamiento de la maqueta. Entre otras prestaciones que funcionan como motor de frecuencia variable; motor de corriente continua sin escobillas; servomotores y motores controlados digitalmente.

Existen varios tipos de motores pasa a paso pero los más comunes son los unipolares y los bipolares.

**Bipolares:** Estos tienen generalmente 4 cables de salida. Necesitan ciertos trucos para ser controlados debido a que requieren del cambio de dirección de flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento.

**Unipolares:** estos motores suelen tener 5 o 6 cables de salida dependiendo de su conexión interna. Este tipo se caracteriza por ser más simple de controlar, estos utilizan un cable común a la fuente de alimentación y posteriormente se van colocando las otras líneas a tierra en un orden específico para generar cada paso, si tienen 6 cables es porque cada par de bobinas tienen un común separado, si tiene 5 cables es porque las cuatro bobinas tienen un polo común; un motor unipolar de 6 cables puede ser usado como un motor bipolar si se deja las líneas del común al aire.

El motor paso a paso 28BYJ-48 según (Forum sur Argentina, 2014) es un motor que viene con un driver ULN2003 (ver Imagen 5.) Para Arduino Tipo de Conexión 5 cables (al controlador del motor)

### **Ficha técnica**

Tensión: 5 a 12 VDC

Frecuencia (combinación): 100Hz

Modo de paso: Half-step (medio paso)/secuencia de señal de control es de 8 (1 ciclo).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Ángulo de paso: Modo de medio paso: 8 secuencias de señal de control (recomendada). Representa 5.625 grados por cada paso / 64 pasos por una revolución completa.

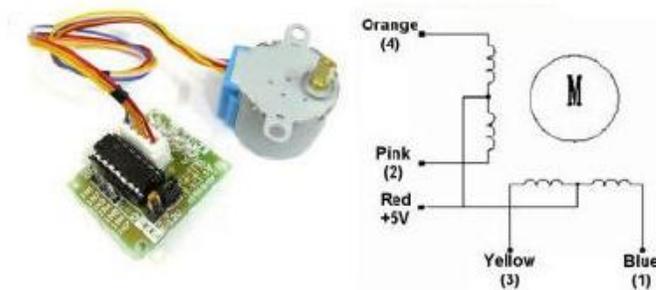
Paso completo: secuencia señal de control de 4 pasos. Representa 11,25 grados por paso / 32 pasos por una revolución del eje del motor completa.

Relación de engranaje

Fabricante especifica 64:1. 64 pasos por rotación del motor x 63,684 relación de transmisión = 4076 pasos por revolución completa (aproximadamente).

Cableado del controlador ULN2003: A- (Blue), B- (Pink), C- (Yellow), D- (Orange), E- (Red, Mid-Point)

Imagen 4. Motor paso a paso 28BYJ-48



#### Half-Step Switching Sequence

Lead Wire Color	--> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 Orange	-	-						-
3 Yellow		-	-	-				
2 Pink				-	-	-		
1 Blue						-	-	-

Forumsur Argentina, (2014). Esquema del motor paso a paso. Recuperado de <https://es.scrib.com>

Imagen 5. Módulo de control para el motor paso a paso con el circuito integrado ULN 2003

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Forumsur Argentina, (2014). Esquema del motor paso a paso. Recuperado de <https://es.scrib.com>

El circuito integrado ULN 2003 (Hangzhou Silan Microelectronics, 2015, p.1) el cual es un arreglo de transistores Darlington, es alimentado de 5V o a 12v DC. ES capaz de proporcionar una gran ganancia de corriente para proteger la placa Arduino ya que la placa proporciona poca corriente (alrededor de 40mA) y por sí sola no es capaz de dar la corriente necesaria para mover el motor paso a paso 28BYJ-48 (aproximadamente 55mA).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3. METODOLOGÍA

---

Para hacer la aplicación Android se encontró gran variedad de plataformas de programación entre las cuales están: AppMakr®, AndroidCreator®, Mobincube®, appsgeyser®, App inventor2®, entre muchas otras pero se decidió por App inventor2® por su lenguaje de programación el cual es muy sencillo, intuitivo y de gran facilidad de comunicación con la tarjeta Arduino uno®.

Seguidamente se programó la salida digital 2 de la placa Arduino uno®, por medio de una aplicación básica Android para activar y desactivar un LED, esta aplicación Android consta de dos botones uno para seleccionar y conectar el bluetooth, el otro para activar la salida de la tarjeta Arduino uno®. (Ver Imagen 18.) y apéndice C. Esto con el objetivo de ensayar la comunicación entre la aplicación Android y la tarjeta Arduino uno®.

#### **Aplicación android básica para hacer pruebas de comunicación con la tarjeta Arduino:**

Esta aplicación básica consta de un selector de bluetooth para poderse comunicar con el módulo bluetooth HC-05 que está conectado a la tarjeta Arduino uno®, también lleva un botón que cuando se presiona envía el caracter “a” a la tarjeta Arduino uno® para que active la salida digital 2 de la tarjeta, y si el botón se mantiene presionado por un momento, envíe el caracter” para que desactive la salida digital 2 que ya se había activado.

En el Apéndice C. Se muestra paso a paso cómo activar y desactivar sólo una salida de la tarjeta Arduino uno® por medio del Smartphone. Para eso se realizó una aplicación básica en android, la programación del módulo HC-05, y la programación del Arduino uno®.

#### **Programación del módulo bluetooth HC-05.**

Para realizar la comunicación entre la placa y el celular se programó el módulo bluetooth con los comandos AT.

Para que el HC-05 entre en modo comandos AT, fue necesario realizar una conexión entre la tarjeta Arduino uno® y el modulo bluetooth, (Ver imagen 24.) ayudado por

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

medio de la librería software Serial.h y el monitor serial (ver Imagen 26) del entorno Arduino. En este se pueden ingresar los comandos necesarios para que el módulo trabaje adecuadamente. (Ver apéndice B, Imagen 25).

#### **Programación de la tarjeta Arduino uno®.**

Para la programación de la placa se utilizaron los caracteres ("a" para activar y "b" para desactivar) que son enviados por la aplicación android mediante el puerto serial. (Ver Imagen 27).

#### **Diseño de la Aplicación android final**

Ya comprobado el funcionamiento de la aplicación con la comunicación bluetooth y la placa Arduino uno® se procede a elaborar la aplicación final que controlará la maqueta.

La aplicación android final (Ver Imagen 6) consta de trece botones. Cada botón activa o desactiva cada una de las salidas del Arduino uno®. Tiene un selector de lista para seleccionar el módulo bluetooth y poderlo conectar, y un botón para cerrar la aplicación. También lleva un campo de texto donde se visualiza la temperatura.

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Imagen 6. Aplicación final terminada mostrando la temperatura real.

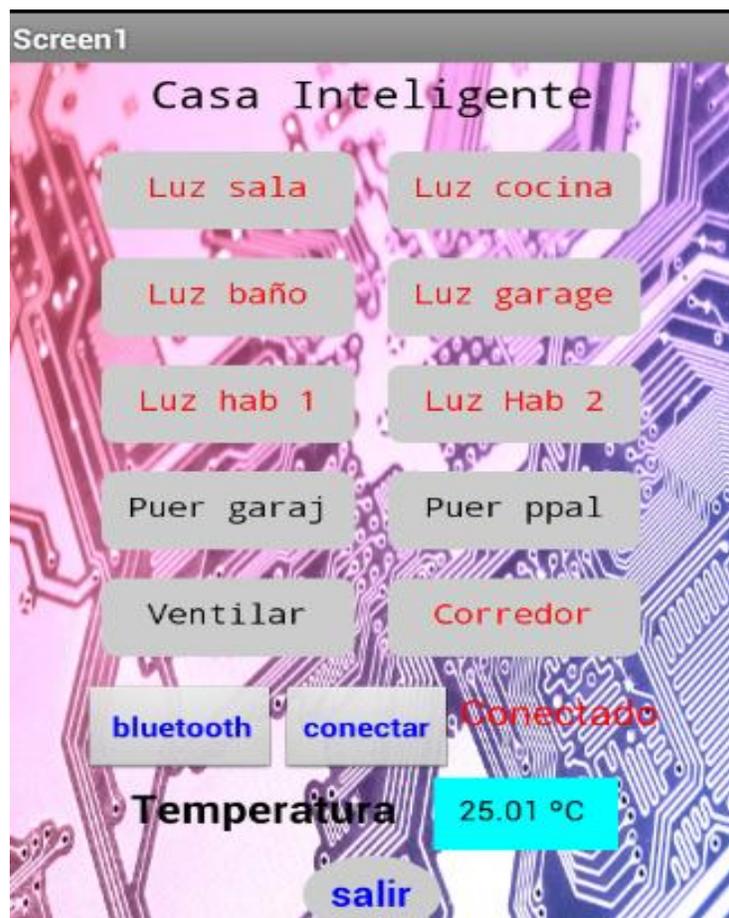


Imagen tomada de la aplicación que se realizó para el smartphone.

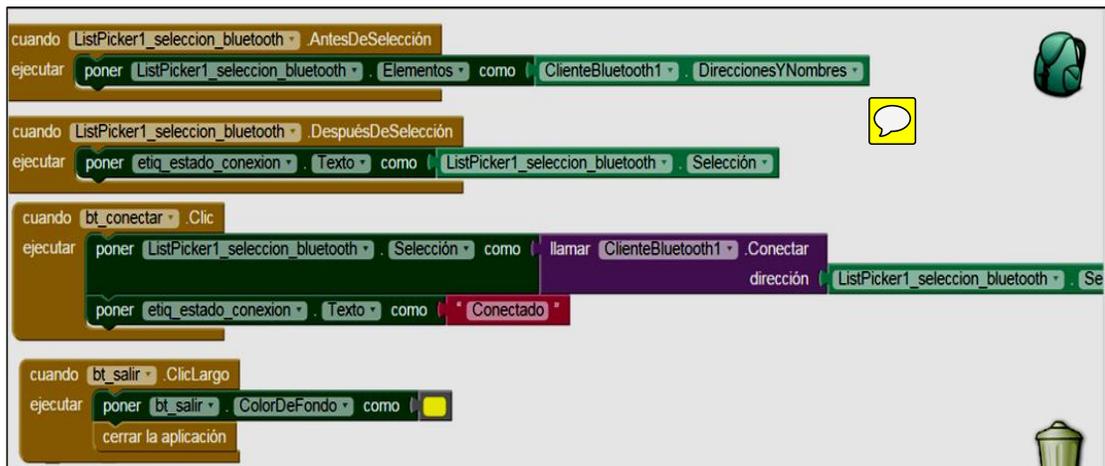
Esta aplicación consta de un *label* o *etiqueta* para mostrar el texto que se visualiza en pantalla y lleva el nombre de casa inteligente. En esta etiqueta es posible modificar la apariencia y la ubicación del texto.

Por otra parte lleva ocho *horizontal Arrangement*, cada uno corresponde a una fila. En el primero tiene dos botones llamados: luz sala, luz cocina. En el segundo se encuentran los botones: luz baño, luz garage. En el tercero están los botones: luz hab

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1, luz hab 2. En el cuarto: puerta garaje, puerta principal. En el quinto: ventilador, corredor. En el sexto se encuentra un *ListPicker* que se llama bluetooth para seleccionar el módulo bluetooth de la lista; un botón que se llama **conectar** para hacer la conexión con el bluetooth seleccionado y un *label* que muestre sí el bluetooth está conectado o desconectado. En el séptimo está otro *label* que dice temperatura y hay un *TextBox* o sea campo de texto que es el encargado de mostrar la temperatura que está censando el circuito integrado lm 35. En el octavo *horizontal Arrangement* hay un botón llamado **salir** que sí se mantiene presionado nos permite cerrar la aplicación en el Smartphone. (Ver Imagen 6)

Imagen 7. Programación de los bloques de la aplicación.



La Imagen 7 muestra cuatro bloques: en el primero está el *ListPicker* “antes de selección”. Sí se presiona este se muestra una lista de las direcciones bluetooth disponibles.

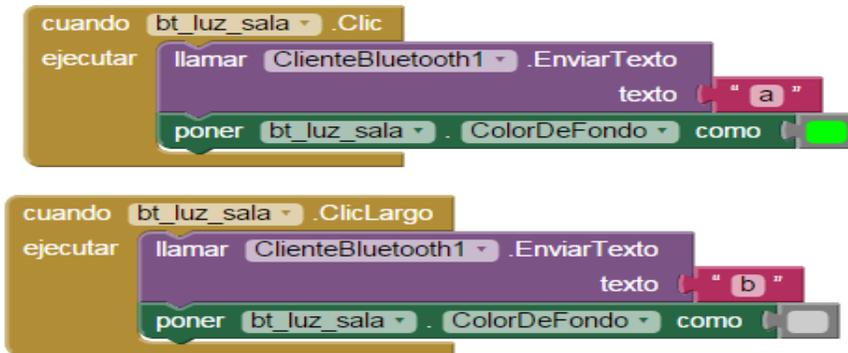
En el segundo bloque está el *ListPicker* “después de selección”. Este es para seleccionar un bluetooth de la lista mostrada en el bloque anterior.

En el tercer bloque está el botón **conectar**, al presionar este se conecta con el bluetooth seleccionado anteriormente y a la vez pone un texto de conectado o no conectado en el *label* o etiqueta que se encuentra en el sexto *horizontal Arrangement*.

En el cuarto bloque está el botón salir. Sí se mantiene presionado un momento se pone el fondo del botón en amarillo y a la vez cierra la aplicación.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

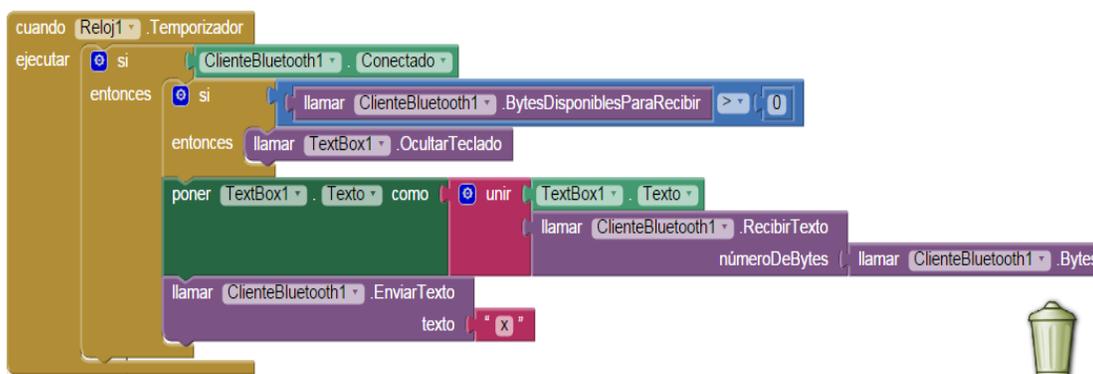
Imagen 8. Programación de los botones.



Después de hacer la conexión con el módulo bluetooth ya se pueden activar los botones que encenderán los actuadores. Cada botón tiene dos funciones clic (pulsación) y clic largo (pulsación sostenida) sí pulsamos, este llama al bluetooth y le envía una letra 'a' que será el caracter que activará la salida digital 2 del Arduino uno® y a la vez pondrá el fondo del botón en verde. El cambio de color es con el objeto de percibir el cambio de estado de cada botón.

Con clic largo, es decir pulsación sostenida, este llama al bluetooth y envía un carácter 'b' que desactivará la salida digital 2 de la placa y a la vez pondrá el fondo del botón en gris. Así sucesivamente es con el resto de los botones (ver apéndice D). Por cada botón se deben utilizar caracteres diferentes y es muy importante tener la precaución de poner el mismo formato (si es minúscula o mayúscula) tanto en la aplicación como en el código del Arduino.

Imagen 9. Bloque del reloj.



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

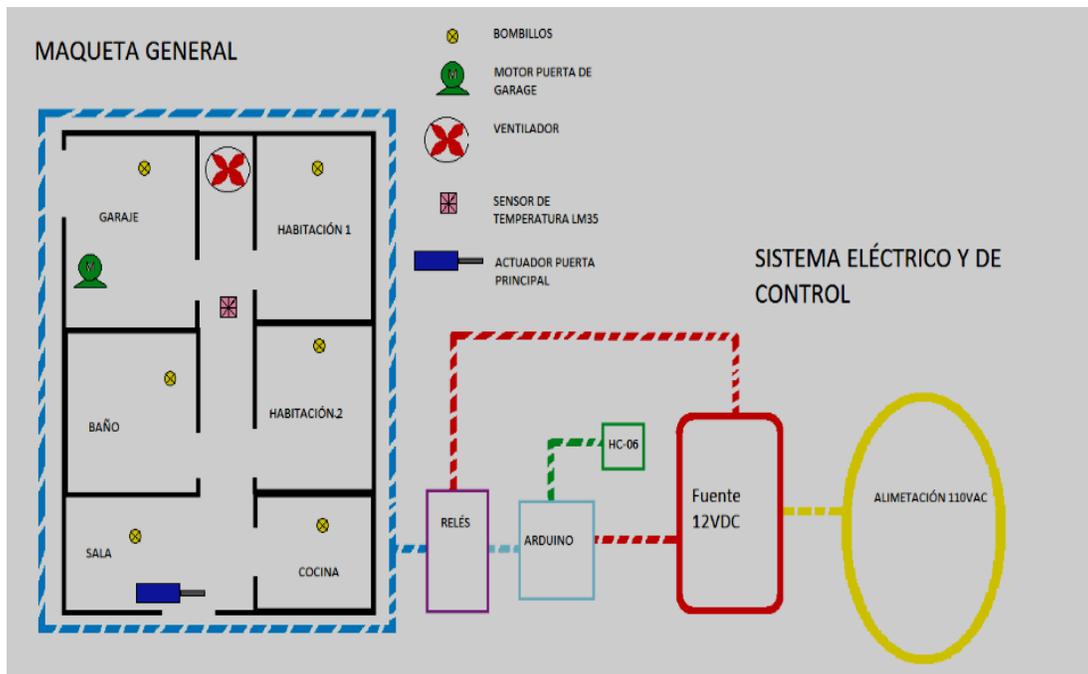
En la Imagen 9 está el bloque del reloj como temporizador. Si el bluetooth está conectado y tiene bytes disponibles para recibir en el puerto serial mayores de cero (si hay datos) entonces llama al textbox1 y al bluetooth para recibir los datos de la temperatura y mostrarlos en el textbox1 que en la aplicación es el espacio donde muestra la temperatura. Esta secuencia la realiza cada 1000ms ya que es el tiempo requerido para que el módulo y el Arduino uno® puedan hacer la comunicación adecuadamente y evitar que se sobre pongan los datos.

Imagen 10. Bloque del reloj para limpiar el puerto.



El reloj2 de la Imagen 10, tiene la tarea de limpiar el textbox1 cada 3 segundos, para que el dato de la temperatura muestre sólo el dato actual y borre el anterior, si no es así se acumularía en pantalla y se perdería la lectura de datos. Para borrar un texto, se pone un texto vacío en el textbox1.

Imagen 15. Diagrama de la maqueta general.



 Institución Universitaria	<b>INFORME FINAL DE  TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la Imagen 15 se ve la maqueta general con sus divisiones, sistemas de control y la ubicación de cada elemento. Se aprecia el orden de los bloques partiendo de la red doméstica 110VAC, que alimenta una fuente regulada a 12VDC, esta a su vez energiza la placa Arduino uno® y el módulo de relés en la parte de potencia.

Imagen 16. Maqueta terminada.

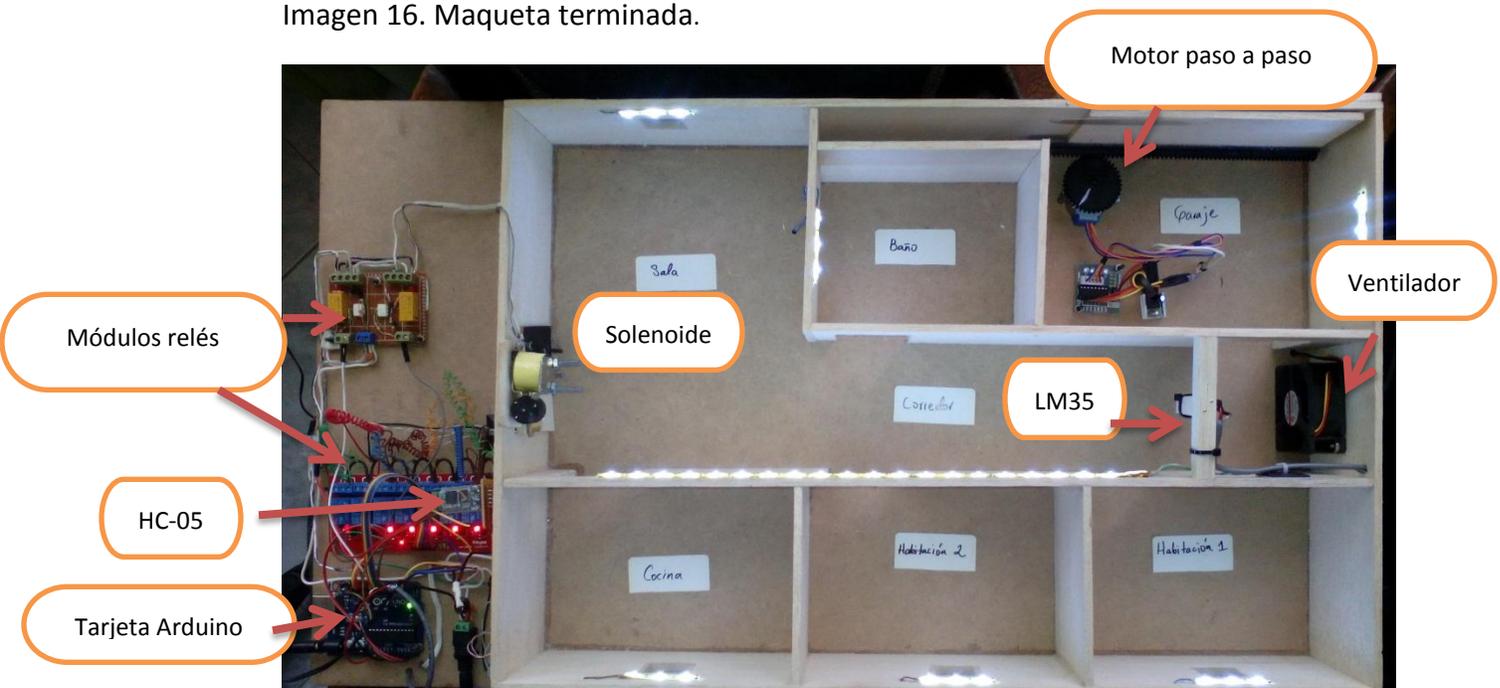
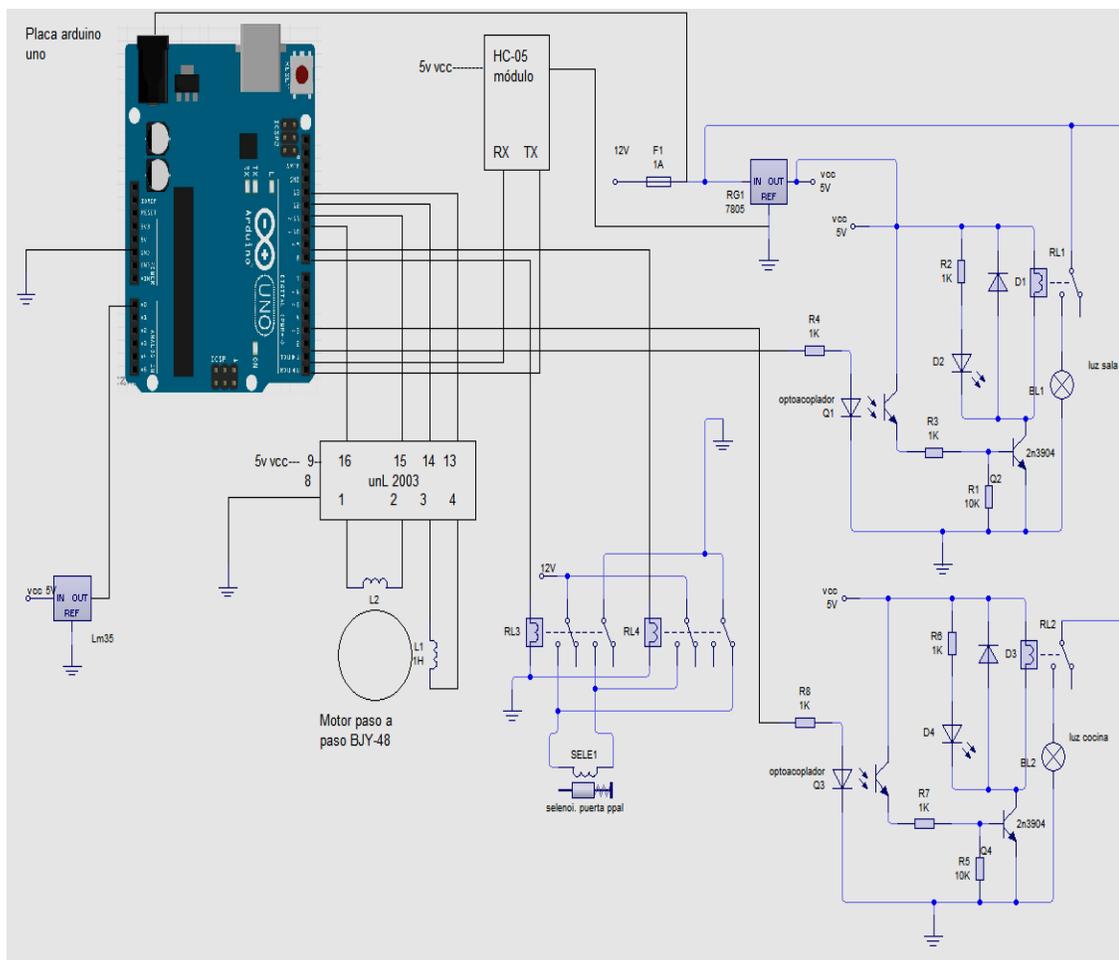


Imagen tomada de la maqueta terminada.

En la Imagen 16 se puede apreciar la maqueta terminada, consta de: una sala, dos habitaciones, una cocina, un baño, un garaje, y un corredor, todas con su respectiva iluminación. El garaje tiene una puerta corrediza la cual se abrirá por medio de un motor paso a paso, la puerta principal abrirá y cerrará por medio de un solenoide, el ventilador está situado en el final del corredor para que pueda ventilar toda el área de la casa y el sensor lm35 está situado en el corredor.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	<b>Código</b>	FDE 089
		<b>Versión</b>	03
		<b>Fecha</b>	2015-01-22

Imagen 17. Diagrama de conexión y esquema electrónico.



Arduino (2015).imagen de placa arduino uno.Recuperado de <https://arduino.cc>

Los módulos de relés son los encargados de acoplar la parte de control con la de potencia para evitar que se dañe el circuito de control. Este bloque está alimentado por la salida del regulador de voltaje (lm7805) a 5 VDC. Dicho bloque compuesto por un optoacoplador 4N25 (el cual es activado por la señal de 5VDC Arduino uno®) que sirve como aislamiento de etapa de potencia y de control para minimizar el efecto del ruido eléctrico producido por las bobinas de los relés en el momento de ser

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

accionados. El transistor NPN con referencia 2n3904 sirve para accionar el relé que maneja las cargas de 12 VCD. Seguidamente se tiene en paralelo a la bobina del relé un diodo N1401 polarizado en inversa para evitar que se dañe el transistor cuando la bobina del relé genere fuerza contra electromotriz luego de ser desconectada. También paralelo a la bobina del relé se encuentra un diodo LED con su resistor limitador de corriente de 1k, para visualizar el accionamiento del relé.

En la tarjeta Arduino uno® van conectados los módulos de relés, el sensor analógico lm35, el motor paso a paso y el módulo bluetooth HC-05.

El módulo bluetooth HC-05 se alimenta a 5VDC, y la señal serial RS232 se conecta cruzada con la de la placa Arduino, es decir, RX del HC-05 con Tx del Arduino uno®; TX del HC-05 con Rx del Arduino uno®.

El motor paso a paso se conecta al Arduino a través de un drive de potencia, El circuito integrado ULN 2003, el cual es un arreglo de transistores Darlington para obtener una ganancia de corriente, es alimentado a 5VDC. Este sirve para proteger la placa Arduino, semejante a la función de los relés, ya que la placa Arduino uno® proporciona poca corriente (40ma) y el motor paso a paso consume aproximadamente 55mA.

El solenoide de la puerta principal está gobernado por dos bloques de relés cada uno con doble contacto, ya que se debe invertir la polaridad para abrir o cerrar la puerta. Es de anotar que en ningún momento estos dos relés se pueden accionar al mismo tiempo ya que se generaría un cortocircuito, por tal motivo se manipula el código del microcontrolador para que esto no suceda. (Ver Imagen 17)

### **PINADO DE SALIDAS DE LA PLACA ARDUINO**

2= (Pin digital) luz sala

3= (Pin digital) luz cocina

4= (Pin digital) luz baño

5= (Pin digital) luz garaje

6= (Pin digital) luz habitación uno

7= (Pin digital) luz habitación dos

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

8 y 9= (Pines digitales) puerta principal

10, 11, 12, 13= (Pines digitales) puerta garaje

A0= (Pin analógico) sensor de temperatura Lm35

A4= (Pin analógico configurado como digital) luz corredor

A5= (Pin analógico configurado como digital) ventilador

El cuarto bloque es un adaptador de 12vdc a 3A que sirve de fuente de voltaje para energizar la maqueta, este será conectado a la red doméstica de 110VAC.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. RESULTADOS Y DISCUSIÓN

---

Se diseñó la maqueta de una casa para hacer un sistema domótico controlado inalámbricamente por medio de un Smartphone o Tablet. El sistema consistió en automatizar las luces, la puerta del garaje, la puerta principal, la ventilación y monitorear la temperatura. La gran ventaja de este proyecto es que se puede modificar y mejorar a largo plazo ya que la gran cantidad de elementos compatibles (sensores actuadores, etc.) que se vienen desarrollando en el mercado es bastante amplio y cada vez es más asequible para el usuario común.

Se hizo la comunicación de 13 señales digitales todas como salidas, 1 análoga como entrada y 2 del puerto serial, por medio de un Smartphone y un módulo bluetooth para controlar la tarjeta Arduino uno®.

Se realizó la comunicación de una señal de temperatura por medio de una tarjeta Arduino uno® que se visualiza en la aplicación del Smartphone (ver imagen 6) por medio de un módulo bluetooth HC-05.

Se diseñó una aplicación para sistema operativo android con el software libre App inventor2® Para interactuar con los periféricos de la maqueta.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

1. El software y hardware libre son una gran combinación que permite implementar sistemas domóticos para personas con discapacidad mejorando la calidad de vida, sin tener que depender de otras personas para realizar ciertas actividades como: encender las luces, abrir o cerrar las puertas y encender o apagar la ventilación.

2. Una de las limitaciones que tiene el proyecto “INTEGRACIÓN DE SISTEMAS CON HARDWARE Y SOFTWARE LIBRE PARA EL DISEÑO DE UNA CASA INTELIGENTE PARA PERSONAS CON DISCAPACIDAD” es el poco alcance o el poco radio de cobertura que posee el módulo bluetooth HC-05, de 10 metros aproximadamente (suficiente para la maqueta), ya que si fuera a instalarse en una casa grande sería necesario cambiar el modulo bluetooth por uno de más capacidad como por ejemplo el RN41 que tiene una distancia de transmisión hasta 100 metros pero que es un poco más costoso.

3. El Appinventor2® es un software fácil de usar ya que permite diseñar aplicaciones sin necesidad de tener mucho conocimiento de programación.

### **Recomendaciones:**

Al programar la tarjeta Arduino uno® para recibir las señales transmitidas por medio del módulo bluetooth desde el Smartphone es necesario utilizar los mismos tipos de datos (caracteres) para garantizar la comunicación entre el Smartphone y la placa.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Configurar la tarjeta Arduino uno, para la transmisión de datos por medio del bluetooth es sólo una de las tantas opciones que hay para realizar esta comunicación ya que también es posible hacerla por medio de infrarrojo, radio frecuencia, Ethernet, wifi, etc.

Si la casa es bastante amplia se recomienda utilizar un módulo bluetooth con más alcance ya que el actual sólo tiene un alcance de aproximadamente 10 metros pero en el mercado se pueden encontrar hasta de 100 metros con la diferencia en el costo, se utilizó el módulo bluetooth HC-05 que sólo tiene un alcance de 10 metros porque es suficiente para el diseño de la maqueta y mostrar su funcionamiento cerca de ella.

**Trabajo futuro:**

Se pueden ampliar las entradas y las salidas cambiando el hardware, o la tarjeta Arduino uno®, que sólo cuenta con 20 pines por la tarjeta Arduino mega® que cuenta con 54 pines, teniendo la posibilidad de agregar más sensores y actuadores, de acuerdo a la necesidad y tipo de infraestructura, y para mejorar su comunicación se puede cambiar el módulo bluetooth por una red wifi que permita manejar los elementos desde una distancia mayor.

Se puede agregar un sistema de cámaras conectadas a la red wifi que se pueda visualizar desde el mismo Smartphone para tener un monitoreo más completo y eficaz.

Un sistema de alarma sería muy importante en este proyecto para garantizar la seguridad de las personas que vivan en la casa.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## REFERENCIAS

- 
- Texas instruments. (2015). LM35 PrecisionCentigradeTemperatureSensors.Agosto 8, 2015, de Texas Instruments Incorporated.Sitio web: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
  - Hak. J. Kim,Jonathan Modell. (2012, Julio 3). Mobile App Design Tool for Smart phones: A Tutorial. International Journal of Soft Computing and Engineering (IJSCE), Volume-2, pp.237-242.2015.agosto 8, De Dialnet Base de datos.
  - Itead Studio. (2010). HC-05 -Bluetooth to Serial Port Module .agosto 2015, de .robotshop.com Sitio web: <http://tinyurl.com/pz3jrja>
  - Arduino. (2015). Technicalspecs.agosto 8, 2015, de ArduinoSitio web: <https://www.arduino.cc/en/Main/arduinoBoardUno>
  - Hernando Barragán. (2011). ¿Qué es Wiring?. *Revista de Arquitectura de la Universidad de los Andes*, volume8, pp.156-158.
  - Hangzhou Silan . (2015). ULN2003. octubre 2015, de Microelectrónica Sitio web: [http://www.datasheetcatalog.com/datasheets\\_pdf/U/L/N/2/ULN2003.shtml](http://www.datasheetcatalog.com/datasheets_pdf/U/L/N/2/ULN2003.shtml)
  - Forumsur Argentina. (2014). Motor Paso a Paso 28BYJ-48 y Driver ULN2003 para Arduino. Julio 09, 2014, de Forumsur Argentina Sitio web: <https://es.scribd.com/doc/233141193/Motor-Paso-a-Paso-28BYJ-48-y-Driver-ULN2003-para-Arduino>
  - Javale D, &Mohsin M, &Nandanwar S, &Shingate M.. (2013 marzo). Home Automation and Security System Using Android ADK. *International Journal of Electronics Communication and Computer Technology*, volume 3, p.383.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## APÉNDICE

### Apéndice A

Imagen 18. Diseño terminado de la aplicación básica de prueba.



#### Aplicación básica de prueba:

En esta imagen se muestra el diseño básico de la aplicación la cual consta de:

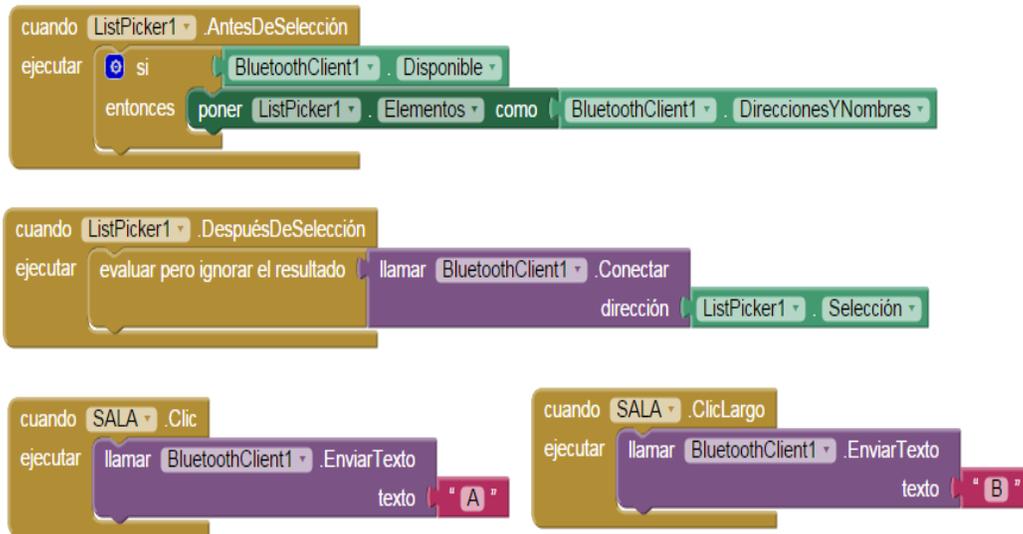
*screen1*. Este *screen1* es la pantalla donde se realiza el diseño, sobre esta quedarán plasmados los botones y demás elementos. Se puede cambiar de apariencia agregándole una imagen de fondo, poner animación, cambiar el color etc.

Para que los botones quedaran más organizados se agregó dos *Horizontal Arrangement* que permite colocarlos en el área deseada, en uno de los *Horizontal Arrangement* se puso un *listPicker* y se cambió el nombre por *CONECTAR*, este es para que muestre los dispositivos bluetooth disponibles, en el otro *horizontal Arrangement* se agregó un botón el cual tiene el nombre de *SALA*, un sólo botón es suficiente para activar y desactivar la salida del Arduino uno®, también es necesario agregar un cliente bluetooth para poder hacer la comunicación con el módulo bluetooth HC-05 que está conectado a la tarjeta del Arduino uno®, este cliente bluetooth queda oculto en la aplicación.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

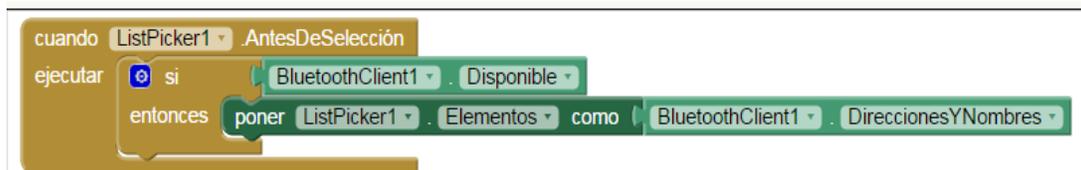
Ya con el diseño hecho se pasa a la ventana de bloques donde se realiza la programación.

Imagen 19. Programación de los bloques de la aplicación básica de prueba.



En la Imagen 19 se observa la programación necesaria para para que la aplicación pueda enviar una señal a la tarjeta Arduino uno® para encender y apagar un LED.

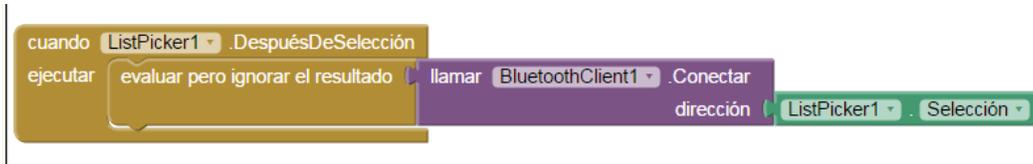
Imagen 20. ListPicker antes de selección.



Sí se presiona *listPicker* y sí está disponible el bluetooth *client* pone la lista de dispositivos bluetooth para que se visualice en este *listPicker* como se ve en la Imagen 20.

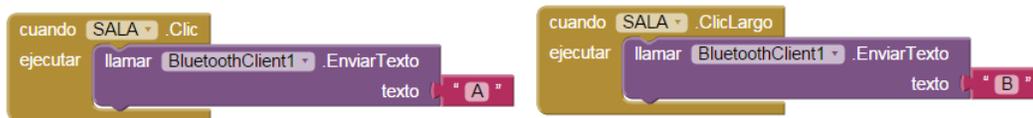
	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Imagen 21. ListPicker después de selección.



Después de llamar la lista de los bluetooth disponibles se selecciona el HC-05 para hacer la comunicación correspondiente con la tarjeta Arduino uno®. (Ver Imagen 21.)

Imagen 22. Botón 1 o botón sala



Después de hacer la selección del bluetooth se procede a programar el botón1(SALA), al seleccionar el botón aparecerán diferentes bloques de programación, debido a que el proyecto consta de varios botones se decide utilizar el bloque de clic y clic largo para ahorrar espacio en la aplicación y darle más practicidad.

Al pulsar el botón1 (SALA) este llama al bluetoothclient y envía un carácter (letra A) a la tarjeta Arduino uno®.

Sí se mantiene pulsado el mismo botón1 (SALA) este llama al bluetoothclient y envía un carácter (letra B) a la tarjeta Arduino uno®. (Ver Imagen 22.)

Luego de terminar la programación de la aplicación se usa la opción de generar código QR para el archivo .apk (Ver Imagen 23)

Imagen 23. Enlace para descargar la aplicación al Smartphone.



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Este código se puede leer por medio de una aplicación que pueda leer códigos QR, puede ser la aplicación llamada Barcode Scanner que está disponible en playstore de forma gratuita, este código lo lleva a un enlace para hacer la descarga directa de la aplicación y poderla instalar en el Smartphone.

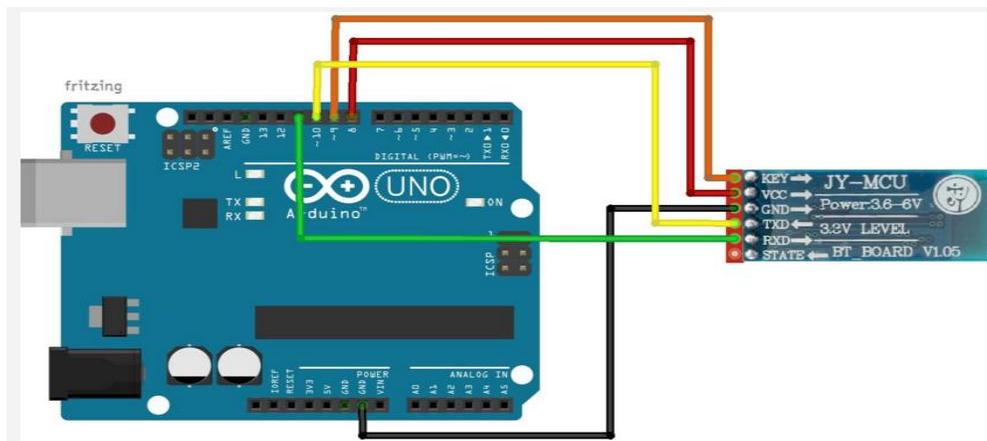
Luego se procede a programar el módulo HC-05 para lograr una adecuada comunicación entre el Smartphone y la tarjeta Arduino uno®.

El HC-05 tiene un modo de comandos AT que debe activarse mediante un estado alto en el PIN34 mientras se enciende (o se resetea) el módulo. En las versiones para protoboard este pin viene marcado como “KEY o EN” (Ver Imagen 1). Una vez que se entra en el modo de comandos AT, se puede configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, contraseña, modo maestro/esclavo, etc.

## Apéndice B.

### Como programar el módulo HC-05 con los comandos AT

Imagen 24.Plano en Arduino uno® para entrar en comandos AT y configurar el modulo bluetooth.



Imagenes tomadas de la pagina web arduino y de iteam studio.

La Imagen24.muestra la forma correcta de conectar el módulo bluetooth para configurar los comandos AT.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para que el HC-05 entre en modo comandos AT, es necesario que el pin KEY se encuentre en HIGH (alto) antes de encender el módulo. Por eso se conecta la tensión Vcc del módulo Bluetooth al pin 8 del Arduino.

El consumo del módulo es menor de 40mA y el Arduino uno® es capaz de alimentarlo sin problemas, por eso el módulo se encenderá cuando pongamos HIGH en el pin 9. Esto permitirá poner en HIGH el pin digital 8, al iniciar nuestro programa y después levantar el pin 8, de este modo cuando encienda entrará en el modo de comandos AT.

Txd y Rxd se deben conectar cruzados con los pines de comunicación de Arduino uno®, que se usará mediante la librería software Serial. (Ver Imagen 25)

Imagen 25. Programación para el Arduino uno® para entrar en comandos AT del bluetooth.

```
comandos_at$
#include <SoftwareSerial.h>

SoftwareSerial BT1(10, 11); // RX | TX
void setup()
{
  pinMode(8, OUTPUT); // Al poner en HIGH forzaremos el modo AT
  pinMode(9, OUTPUT); // cuando se alimente de aqui
  digitalWrite(9, HIGH);
  delay (500) ; // Espera antes de encender el modulo
  Serial.begin(9600);
  Serial.println("Levantando el modulo HC-05");
  digitalWrite (8, HIGH); //Enciende el modulo
  Serial.println("Esperando comandos AT:");
  BT1.begin(4600);
}

void loop()
{
  if (BT1.available())
    Serial.write(BT1.read());
  if (Serial.available())
    BT1.write(Serial.read());
}
```

En la imagen 25 se ve la programación necesaria del de Arduino uno®, para poder entrar en comandos AT del módulo bluetooth.

Para ver los comandos AT y modificarlos se debe abrir el monitor serie del Arduino uno® que se encuentra en la parte superior derecha en forma de lupa. (Ver Imagen 26)

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Imagen 26. Aérea donde se modifican los comandos AT.



Estos son algunos de los comandos básicos (que se utilizaron) para configurar el módulo bluetooth HC-05.

AT: para saber si está comunicado deberá responder con un ok.

AT+VERSION: para saber cuál es la versión

AT+ADDR: para saber la Mac del dispositivo

AT+ROLE: para saber si está como esclavo o maestro

AT+ROLE= 0: para configurarlo como esclavo

AT+ROLE= 1: para configurarlo como maestro

AT+NAME: para saber el nombre

AT+NAME=XXX: para cambiar el nombre

AT+UART=9600: para ponerle una velocidad de 9600 baudios

AT+ORGL: para reset y queda de fábrica

AT+PSWD: para saber la clave que tiene

AT+PSWD=XXX: para asignar una clave diferente

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Apéndice C.

### Programación de la tarjeta Arduino uno® para encender y apagar un LED.

Imagen 27. Programación básica para prender un LED por medio de la aplicación android.

```

void setup() {
  pinMode(2,OUTPUT); //configura el pin 2 como salida

  Serial.begin(9600); //Indica al programa que vamos
                      //a iniciar la comunicación serial a 9600 baudios
}

void loop() {
  char dato=Serial.read(); //lee los datos que están en el puerto serial
                          //y los almacena en la variable dato.

  if(dato=='A')digitalWrite(2,HIGH); //activa el pin 2
  if(dato=='B')digitalWrite(2,LOW); //desactiva el pin 2
}

```

El setup es la primera función en ejecutarse dentro de un programa en Arduino uno®. Es básicamente, donde se “setean” las funciones que llevará a cabo el microcontrolador. Aquí es donde se establece algunos criterios que requieren una ejecución única. Por ejemplo, si el programa va a usar comunicación serial, en el setup se establece el comando Serial.begin para indicarle al programa que se va a iniciar la comunicación serial a 9600 baudios (la velocidad de transmisión de datos).

Si se va a utilizar un pin determinado como salida de voltaje, se usa el *pinMode* para indicarle al Arduino uno® que determinado pin funcionará como salida, usando el parámetro OUTPUT.

La función loop en Arduino es la que se ejecuta un número infinito de veces. Al encenderse el Arduino se ejecuta el código del setup y luego se entra al loop, el cual se repite de forma indefinida hasta que se apague o se reinicie el microcontrolador.

Char; es un tipo de variable que ocupa un byte de memoria y almacena un valor de carácter.

```
char dato=Serial.read();
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La variable dato que es tipo char lee los datos que están en el puerto serial y los almacena en la variable dato.

```
if(dato=='A')digitalWrite(2,HIGH);
```

```
if(dato=='B')digitalWrite(2,LOW);
```

La función digitalWrite se utiliza para activar o desactivar las salidas digitales.

Entonces sí la variable dato es igual al caracter A pone en alto la salida digital 2 y sí dato es igual al caracter B pone la salida digital 2 en bajo.

El código se carga a la tarjeta Arduino uno® por medio del cable USB y se compila para que muestre sí tiene errores.

### **Ensayos de comunicación.**

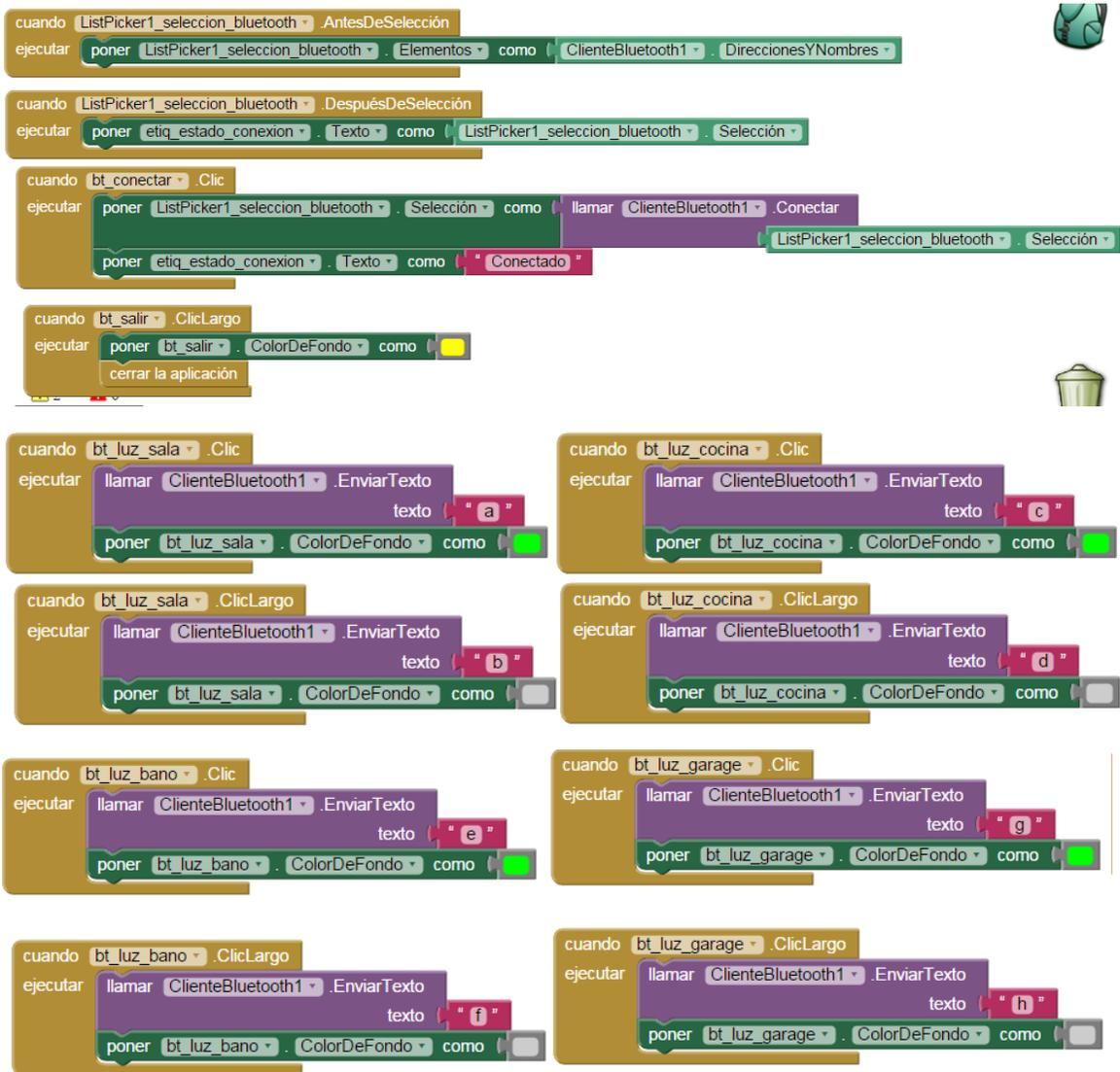
Primero se activa el bluetooth del Smartphone, se abre la aplicación y se selecciona el módulo bluetooth HC-05 se presiona conectar, sí el módulo HC-05 se enciende de forma intermitente es porque hay comunicación, se coloca un LED en la salida digital 2 del Arduino uno® hacia el pin GND (tierra) y presionamos el botón de la aplicación para activar esta salida digital 2 y sí se mantiene presionado el mismo botón se desactiva la salida digital 2.

Ya con esta prueba básica funcionando se puede comenzar a ensayar una a una las salidas digitales del Arduino uno® mediante la pulsación de teclas en el móvil.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Apéndice D.

### Programación en bloques de la aplicación en App inventor.



The image displays the following code blocks in App Inventor:

- Bluetooth Selection:**
  - When `ListPicker1_seleccion_bluetooth` is selected (AntesDeSelección), execute: `poner ListPicker1_seleccion_bluetooth . Elementos como ClienteBluetooth1 . DireccionesYNombres`.
  - When `ListPicker1_seleccion_bluetooth` is selected (DespuésDeSelección), execute: `poner etiq_estado_conexion . Texto como ListPicker1_seleccion_bluetooth . Selección`.
- Connection Control:**
  - When `bt_conectar` is clicked (.Clic), execute:
    - `poner ListPicker1_seleccion_bluetooth . Selección como llamar ClienteBluetooth1 . Conectar`
    - `poner etiq_estado_conexion . Texto como "Conectado"`
  - When `bt_salir` is clicked (.ClicLargo), execute:
    - `poner bt_salir . ColorDeFondo como [Yellow]`
    - `cerrar la aplicación`
- Light Indicators (Click Events):**
  - `bt_luz_sala` (.Clic): `llamar ClienteBluetooth1 . EnviarTexto texto "a"`, `poner bt_luz_sala . ColorDeFondo como [Green]`
  - `bt_luz_cocina` (.Clic): `llamar ClienteBluetooth1 . EnviarTexto texto "c"`, `poner bt_luz_cocina . ColorDeFondo como [Green]`
  - `bt_luz_bano` (.Clic): `llamar ClienteBluetooth1 . EnviarTexto texto "e"`, `poner bt_luz_bano . ColorDeFondo como [Green]`
  - `bt_luz_garage` (.Clic): `llamar ClienteBluetooth1 . EnviarTexto texto "g"`, `poner bt_luz_garage . ColorDeFondo como [Green]`
- Light Indicators (Long-Click Events):**
  - `bt_luz_sala` (.ClicLargo): `llamar ClienteBluetooth1 . EnviarTexto texto "b"`, `poner bt_luz_sala . ColorDeFondo como [Grey]`
  - `bt_luz_cocina` (.ClicLargo): `llamar ClienteBluetooth1 . EnviarTexto texto "d"`, `poner bt_luz_cocina . ColorDeFondo como [Grey]`
  - `bt_luz_bano` (.ClicLargo): `llamar ClienteBluetooth1 . EnviarTexto texto "f"`, `poner bt_luz_bano . ColorDeFondo como [Grey]`
  - `bt_luz_garage` (.ClicLargo): `llamar ClienteBluetooth1 . EnviarTexto texto "h"`, `poner bt_luz_garage . ColorDeFondo como [Grey]`

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

cuando **bt\_luz\_hab1** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "i"  
 poner **bt\_luz\_hab1** .ColorDeFondo como

cuando **bt\_luz\_hab2** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "k"  
 poner **bt\_luz\_hab2** .ColorDeFondo como

cuando **bt\_luz\_hab1** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "j"  
 poner **bt\_luz\_hab1** .ColorDeFondo como

cuando **bt\_luz\_hab2** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "l"  
 poner **bt\_luz\_hab2** .ColorDeFondo como

cuando **bt\_puerta\_garage** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "m"  
 poner **bt\_puerta\_garage** .ColorDeFondo como

cuando **bt\_puerta\_ppal** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "o"  
 poner **bt\_puerta\_ppal** .ColorDeFondo como

cuando **bt\_puerta\_garage** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "n"  
 poner **bt\_puerta\_garage** .ColorDeFondo como

cuando **bt\_puerta\_ppal** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "p"  
 poner **bt\_puerta\_ppal** .ColorDeFondo como

cuando **bt\_ventilacion** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "q"  
 poner **bt\_ventilacion** .ColorDeFondo como

cuando **bt\_corredor** .Clic  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "s"  
 poner **bt\_corredor** .ColorDeFondo como

cuando **bt\_ventilacion** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "r"  
 poner **bt\_ventilacion** .ColorDeFondo como

cuando **bt\_corredor** .ClicLargo  
 ejecutar  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "t"  
 poner **bt\_corredor** .ColorDeFondo como

cuando **Reloj1** .Temporizador  
 ejecutar  
 si **ClienteBluetooth1** .Conectado  
 entonces  
 si **llamar ClienteBluetooth1 .BytesDisponiblesParaRecibir** > 0  
 entonces  
 llamar **TextBox1** .OcultarTeclado  
 poner **TextBox1** .Texto como unir **TextBox1** .Texto  
 llamar **ClienteBluetooth1** .RecibirTexto  
 llamar **ClienteBluetooth1** .BytesDisponiblesParaRecibir  
 llamar **ClienteBluetooth1** .EnviarTexto  
 texto "x"

cuando **Reloj2** .Temporizador  
 ejecutar  
 poner **TextBox1** .Texto como "0"

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## Apéndice E.

### Código de la programación en Arduino uno®.

```
#include <SoftwareSerial.h>
```

```
#include <Stepper.h>
```

```
//la secuencia debe ser azul, amarillo, rosa, naranja
```

```
//una vuelta entera corresponde a 2048 pasos en el BLY-48
```

```
Stepper(2048,10,12,11,13); //se asigna pines para controlar motor paso a paso
```

```
#define TxD 1 //se define constante para la comunicación serial rs232
```

```
#define RxD 0 //se define constante para la comunicación serial rs232
```

```
SoftwareSerialblue(TxD,RxD); //asignación de pines para comunicación serial ficticia por software
```

```
void setup(){// Inicio setup
```

```
motor.setSpeed(10);
```

```
Stepper(2048,10,12,11,13); //se asigna pines para controlar motor paso a paso
```

```
pinMode(2,OUTPUT); //Luz sala
```

```
pinMode(3,OUTPUT); //Luz cocina
```

```
pinMode(4,OUTPUT); //Luz baño
```

```
pinMode(5,OUTPUT); //luz garaje
```

```
pinMode(6,OUTPUT); //Luz habitación 1
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

pinMode(7,OUTPUT); //Luz habitación 2
pinMode(8,OUTPUT); //activar solenoide puerta principal
pinMode(9,OUTPUT); //descativar solenoide puerta principal
pinMode(10,OUTPUT); //motor
pinMode(11,OUTPUT); //motor
pinMode(12,OUTPUT); //motor
pinMode(A5,OUTPUT); //Ventilador
pinMode(A4,OUTPUT); //Luz corredor

digitalWrite(8,LOW); //aseguro que las salidas que activan las puerta principal estén
//apagadas
digitalWrite(9,LOW); //aseguro que las salidas que activan las puerta ppal. Estén apagadas

blue.begin(9600); //tasa de transmisión en baudios/s de la comunicación serial ficticia por
software
Serial.begin(9600); //tasa de transmisión en baudios/s de la comunicación serial física

} //fin setup

voidloop(){ // inicio loop

char dato=Serial.read();
char comando = blue.read();
floatlectura_lm=analogRead(A0);
floattemp=(lectura_lm*100*5)/1023.0;
intaux=0;

Serial.println(dato); //imprimir en serial lo que envía el celular+

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

Serial.print(temp);
Serial.println(" °C ");
delay(100);

if(comando=='x'){
blue.print(temp);
blue.println(" °C ");
    //delay(1000);
    }

switch(dato){ //inicio case

case 'a':digitalWrite(2,HIGH); //Luz Sala on
break;
case 'b':digitalWrite(2,LOW); //Luz Sala off
break;

case 'c': digitalWrite(3,HIGH); //Luz cocina on
break;
case 'd': digitalWrite(3,LOW); //Luz cocina off
break;

case 'e': digitalWrite(4,HIGH); //Luz baño on
break;
case 'f': digitalWrite(4,LOW); //Luz baño off
break;

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

case 'g': digitalWrite(5,HIGH); //Luz garaje on
break;
case 'h': digitalWrite(5,LOW); //Luz garaje off
break;

```

```

case 'i': digitalWrite(6,HIGH); //Luz habitación 1 on
break;
case 'j': digitalWrite(6,LOW); // Luz habitación 1 off
break;

```

```

case 'k': digitalWrite(7,HIGH); // Luz habitación 2 on
break;
case 'l': digitalWrite(7,LOW); // Luz habitación 2 off
break;

```

```

case 'm':   for(int mot=0; mot<2048; mot++){
motor.step(-1);
          }

```

```

motor.step(-2048); // abre puerta indicando al motor -360º
break;
delay(200);
case 'n':
motor.step(2048); // cierra puerta indicando al motor 360º

```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

break;

case 'o': digitalWrite(8,LOW);

delay(500);

digitalWrite(9,HIGH); // abre puerta principal.

delay(500);

digitalWrite(9,LOW);

break;

case 'p': digitalWrite(9,LOW);

delay(500);

digitalWrite(8,HIGH); // cierra puerta principal

delay(500);

digitalWrite(8 ,LOW);

break;

case 'q': digitalWrite(A5,HIGH); //ventilador on

break;

case 'r': digitalWrite(A5,LOW); //ventilador off

break;

case 's': digitalWrite(A4,HIGH); //Luz corredor on

break;

case 't': digitalWrite(A4,LOW); // Luz corredor off

break;

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
} //fin case
}
```

## Apéndice F.

### Explicación de código de Arduino uno.

Imagen 11. Configuración de las librerías y puerto serial.

```
codigo_casa_i
#include <SoftwareSerial.h>

#include <Stepper.h>
//LA SECUENCIA DEBE SER AZUL,AMARILLO,ROSA, NARANJA
//UNA VUELTA ENTERA CORRESPONDE A 2048 PASOS EN EL
//BJY-48
Stepper motor(2048,10,12,11,13);//se asigna pines para controlar motor paso a paso

#define TxD 1 //se define constante para la comunicacion serial rs232
#define RxD 0 //se define constante para la comunicacion serial rs232

SoftwareSerial blue(TxD,RxD); //asignacion de pines para comunicacion serial ficticia por software
```

En la Imagen 11, se muestra la librería del puerto serial y la librería para el motor paso a paso, se definen los pines para la comunicación serial.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Imagen 12. Configuración de pines como salida.

```

void setup(){//inicio setup

  motor.setSpeed(10);
  Stepper motor(2048,10,12,11,13);//se asigna pines para controlar motor paso a paso

  pinMode(2,OUTPUT); //luz sala
  pinMode(3,OUTPUT); //luz cocina
  pinMode(4,OUTPUT); //luz baño
  pinMode(5,OUTPUT); //luz garaje
  pinMode(6,OUTPUT); //Luz habitacion 1
  pinMode(7,OUTPUT); //Luz habitacion 2
  pinMode(8,OUTPUT); //activar selenoide puerta principal
  pinMode(9,OUTPUT); //descativar selenoide puerta principal
  pinMode(10,OUTPUT); //motor
  pinMode(11,OUTPUT); //motor
  pinMode(12,OUTPUT); //motor
  pinMode(A5,OUTPUT); //Ventilador
  pinMode(A4,OUTPUT); //luz corredor

  digitalWrite(8,LOW);//aseguro que las salidas que activan las puerta ppal esten apagadas
  digitalWrite(9,LOW);//aseguro que las salidas que activan las puerta ppal esten apagadas

  blue.begin(9600);//tasa de transmision en baudios/s de la comunicacion serial ficticia por software
  Serial.begin(9600);//tasa de transmision en baudios/s de la comunicacion serial fisica

```

En la Imagen 12, la función void setup se ejecuta una sola vez. Dentro de ella se asignan los pines para el motor paso a paso y su velocidad (10rpm) se configuran los otros pines como salidas, se configura la velocidad de transferencia del puerto serial en 9600 baudios y los pines analógicos A5 y A4 como pines de salida digital.

Imagen 13. Configuración del void loop

```

codigo_casa_i$
void loop() {//// inicio loop

  char dato=Serial.read();
  char comando = blue.read();
  float lectura_lm=analogRead(A0);
  float temp=(lectura_lm*100*5)/1023.0;
  int aux=0;

  Serial.print(temp);//Serial.println(dato);///imprimir en serial lo que envia el celular+
  Serial.println(" °C ");
  delay(100);

  if(comando=='x'){
    blue.print(temp);
    blue.println(" °C ");
    delay(1000);
  }
}

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La función loop (Imagen 13) se ejecuta un número infinito de veces. Lo que hace que las variables del puerto serial estén constantemente censando por medio del sensor de temperatura Im35, si la variable comando es igual a **x** envía los datos recibidos de temperatura a la aplicación android instalad en el Smartphone con la función de printy el println es para poner el símbolo de grados centígrados °C.

Imagen 14. Comandos switch case.

```

codigo_casa_i$
  delay(1000);
}

switch(dato){///inicio case

  case 'a':digitalWrite(2,HIGH); //luz sala on
    break;
  case 'b':digitalWrite(2,LOW); //luz sala off
    break;

  case 'c': digitalWrite(3,HIGH); //luz cocina on
    break;
  case 'd': digitalWrite(3,LOW); //luz cocina off
    break;
}

```

Una sentencia **switch** compara el valor de una variable a los valores especificados en las declaraciones de casos. Cuando se encuentra una sentencia **case** cuyo valor coincide con la de la variable, se ejecuta el código en esa sentencia case.

La palabra clave **Break** sale de la sentencia switch, y se suele utilizar al final de cada caso. Sin una sentencia break, la sentencia switch continuará ejecutando las siguientes expresiones.

En la Imagen 14, si la variable dato es igual al case **'a'** entonces se pone en alto la salida 2 del Arduino uno®. Y así sucesivamente con todos los casos necesarios para activar todas las salidas. Hay que tener presente que la variable "dato" recibe los caracteres enviados desde el celular por tal motivo los caracteres deben ser exactamente iguales para que los pueda reconocer.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES Andres Sanchez E.

Jorge Machado C.

FIRMA ASESOR Audli A. A.

FECHA ENTREGA: 12-01-2016\_\_\_\_\_

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO\_\_\_ ACEPTADO\_\_\_ ACEPTADO CON MODIFICACIONES\_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_