

RIC

# Prototipo de un mapa virtual para la orientación de las personas en un establecimiento público empleando tecnología Arduino

## Prototype of a virtual map orientation for persons in a public establishment using Arduino technology

Edgar Espinosa<sup>1</sup>; George Arrow Yu Jiang<sup>1</sup> & Euclides Samaniego González<sup>2\*</sup>

<sup>1</sup>Licenciatura en Ingeniería de Sistemas y Computación – Facultad de Ingeniería de Sistemas Computacionales–  
Universidad Tecnológica de Panamá

<sup>2</sup>Grupo de Investigación en Inteligencia Computacional –GIICOM–  
Facultad de Ingeniería de Sistemas Computacionales – Universidad Tecnológica de Panamá

**Resumen** El proyecto consiste en el desarrollo de un prototipo de mapa virtual para orientar a las personas dentro de un lugar público utilizando tecnología Arduino, adaptándolo a través de una aplicación móvil desarrollada en el sistema operativo Android. Utiliza comunicación serial entre la plataforma Arduino y Android a través de un módulo Bluetooth en el cual el usuario de la aplicación logra ubicar su posición y obtener información del lugar en el que se encuentra.

**Palabras claves** Mapa virtual, plataforma Arduino, Google Maps, GPS.

**Abstract** The project involves the development of a prototype virtual map to guide people in a public place using Arduino technology, adapting through a mobile application developed on the Android operating system. Use serial communication between the Arduino and Android platform via a Bluetooth module in which the user application manages locate your position and information of where it is located.

**Keywords** Virtual map, Arduino platform, Google Maps, GPS.

---

\* Corresponding author: euclides.samaniego@utp.ac.pa

## 1. Introducción

Los mapas son una fuente importante de información desde la antigüedad y gran parte de la actividad cotidiana actual está relacionada con ellos. Esto ha sucedido principalmente con el desarrollo de las ciencias computacionales, la expansión de internet, el *hardware* y *software* de los equipos, y la evolución de la tecnología en dispositivos móviles.

Los mapas virtuales han tomado gran auge con aplicaciones como Waze y el servicio de Google Maps que permiten tener vistas satelitales de las calles e información de las ciudades indicando su posición a través del GPS integrado a los dispositivos, obteniendo orientación y rutas deseadas.

La plataforma Arduino, con su variedad de placas programables, diseñadas para que los entusiastas tengan la facilidad de crear proyectos en múltiples disciplinas, permite a través de sus entradas analógicas y digitales tomar información del entorno y procesarla mediante su lenguaje de programación, adaptándose a las necesidades de cada desarrollador.

El sistema operativo móvil Android como uno de los líderes del mercado tiene como característica ser de código abierto, gratuito y multiplataforma, permitiendo a cualquier persona realizar aplicaciones móviles propias.

Con las características de estas dos plataformas, está fundamentado el desarrollo de este proyecto.

El proyecto se realizó por medio de una maqueta-prototipo de un edificio, se utilizarán sensores ultrasónicos con los cuales se obtendrá la posición de un objeto e información del lugar y se enviará a través de comunicación serial y un módulo Bluetooth a una aplicación desarrollada en el sistema operativo Android.

### 1.1 Aspectos generales

El mapa electrónico es una herramienta utilizada para orientarse y guiarse en el camino. A diferencia de los mapas tradicionales hecho en algún material de hoja o textil impreso, el mapa electrónico se presenta en algún dispositivo electrónico, como *smartphones*, computadoras, *tablets*, etc., y es mostrada en una pantalla digital. Muchos de estos dispositivos tienen

GPS integrados y lo utilizan para sincronizar el mapa con la posición actual.

Google Maps es uno de los mapas electrónicos más conocido mundialmente, es utilizado para conocer caminos y los lugares de los alrededores, tiene diferentes tipos de modalidades, una de ellas visualiza con claridad las calles y el otro visualiza un mapa real a través de imágenes satelitales.

### 1.2 Caracterización del problema

Siempre que se accede a un establecimiento, por primera vez, la persona no tiene la orientación del lugar y no se tiene idea de donde se encuentra o hacia dónde dirigirse con exactitud. En el caso particular de los universitarios de primer ingreso, por ejemplo, ellos no saben dónde están los salones para asistir a sus clases, y necesitan preguntar y/o ver salón por salón y piso por piso, hasta encontrar el buscado.

Tampoco saben a qué partes del establecimiento pueden acceder y qué partes no pueden.

Este problema se refleja más en lugares grandes y lugares que tienen usuarios nuevos con frecuencia.

En lugares chicos no es muy necesaria la utilidad de un mapa pero para establecimientos o campus que tienen diferentes edificios y lugares, requieren de un mapa de guía y que pueda darle la información necesaria al usuario.

### 1.3 Alternativa de solución

El proyecto propone el desarrollo de una aplicación que crea un mapa virtual que facilite la orientación de las personas en un establecimiento o lugares públicos que un GPS no puede localizar con exactitud empleando tecnología Arduino.

### 1.4 Justificación

La creación de un mapa electrónico ayuda a los usuarios que ingresan a un establecimiento por primera vez y no saben a dónde dirigirse o dónde están parados. Esto les otorgará más seguridad de dónde se encuentran y dónde pueden acceder, tener la información necesaria del lugar en la palma de la mano.

Para los turistas que quieran visitar los *malls* o

sitios turísticos, tener a la mano un mapa virtual del lugar sería útil y evitarían perderse o no encontrar el lugar o sección que estén buscando como los restaurantes o los baños.

### 1.5 Restricciones y limitaciones

Para la implementación a gran escala se necesitará acceso completo del establecimiento para la instalación de los equipos y sensores.

Se requiere de una conexión del lugar para poder actualizar la posición del usuario.

La complejidad de la programación en la coordinación del equipo con el mapa electrónico.

Tener acceso a ciertas informaciones del establecimiento o del lugar.

### 1.6 Objetivos

#### 1.6.1 Generales

Crear un mapa virtual que facilite la orientación de las personas en un establecimiento o lugares públicos que un GPS no puede localizar con exactitud.

#### 1.6.2 Específicos

- Crear una maqueta que represente un piso del establecimiento con los diferentes lugares o secciones.
- Diseñar el mapa virtual de la maqueta y programar la sincronización del mapa con la posición del individuo.
- Instalar sensores en diferentes lugares de la maqueta que serán controlados por un Arduino.
- Crear una aplicación móvil para que el usuario tenga la información disponible.
- Darle la localización exacta al usuario dentro del establecimiento.

## 2. Diseño del mapa

### 2.1 Prueba de aplicación sencilla en Android Studio

En esta primera prueba se ejecutó una aplicación para familiarizarse con el entorno de desarrollo de Android Studio, en la cual para diferenciarse de la típica aplicación predefinida de “Hola mundo”, se probaron otras opciones

distintas como la introducción de datos, botón de acción, cambio en el diseño de la pantalla principal.

Código XML de los elementos visuales de la pantalla principal:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LytContenedor"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/LblNombre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/nombre" />
    <EditText android:id="@+id/TxtNombre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text" />
    <Button android:id="@+id/BtnAceptar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/acceptar" />
</LinearLayout>
```

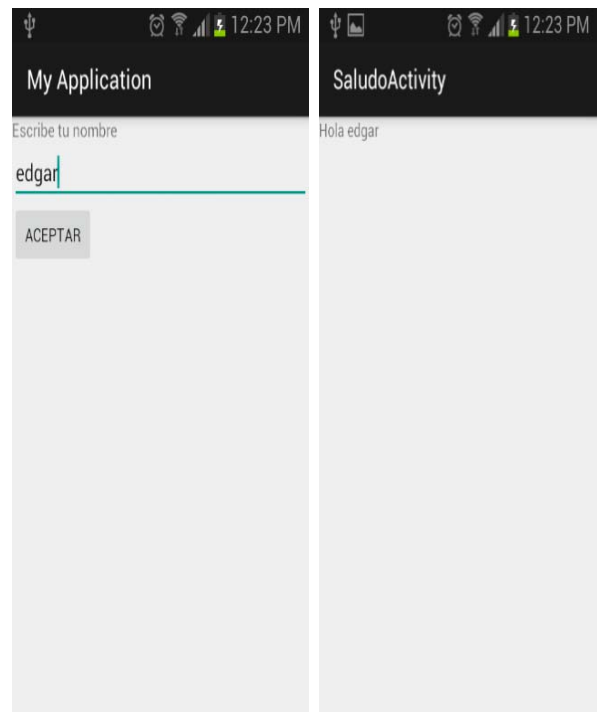


Figura 1. Prueba de aplicación con Android Studio.

## 2.2 Prueba de diseño del mapa en la aplicación

Se ejecutó una aplicación que muestra en pantalla el diseño o bosquejo preliminar del mapa virtual del edificio a través de una imagen .jpg como fondo para tener una idea de la interfaz que va a tener el usuario a la vista.

### 2.2.1 Resultado

La imagen mostrada está tomada desde la aplicación ejecutada desde un celular S3 Mini, posiblemente en dispositivo de mayor resolución de pantalla la calidad de la imagen se pierda por lo que se consideró crear una imagen escalable en SVG.

## 2.3 Prueba con imagen SVG en Android

Para probar una imagen SVG en Android se necesita de una librería especial llamada SVG Android, la cual permite desplegar imágenes en código XML, la librería no es soportada nativamente por Android.

Esto permitiría que la imagen del mapa sea escalable y en cualquier resolución de pantalla mantenga su calidad. Ver figura 2.



Figura 2. Bosquejo preliminar del mapa virtual del edificio.

Se utilizó Adobe Illustrator que es un *software* que permite crear imágenes y exportarlas en formato SVG. Ver figura 3.

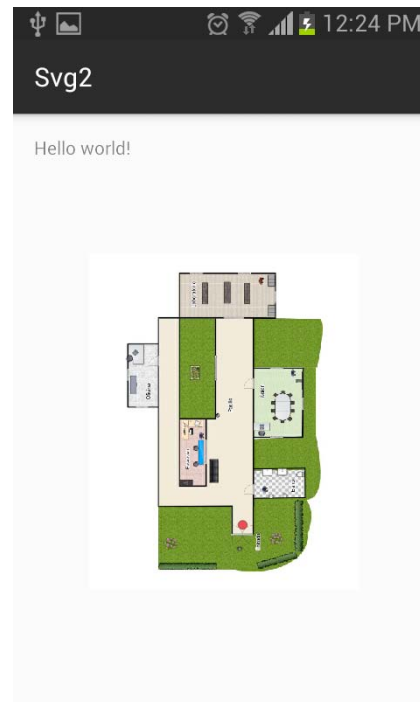


Figura 3. Prueba con imagen SVG en Android.

### 2.3.1 Resultado

En un principio se utilizó la librería SVG-android que permite mostrar una imagen sencilla SVG pero al cargar la del mapa modificado en Adobe Illustrator no logró ejecutar, debido a que era más compleja y la librería no estaba actualizada desde hacía tres años. Al utilizar una nueva librería llamada androidsvg, actualizada hace un año logró ejecutar la imagen que se había modificado del mapa con un pequeño punto rojo que mostraría la posición del usuario dentro del edificio. Aún se tiene que trabajar en mejorar el tamaño de la imagen SVG y al abrir la aplicación varias veces en ocasiones no renderiza por completo la imagen.

## 2.4 Diseño y programación de la interfaz

La aplicación está compuesta por dos pantallas, la primera pantalla es un listado de los dispositivos para la conexión Bluetooth y la segunda pantalla con el mapa y el posicionamiento del usuario. Ver figuras 4 y 5.

### 2.4.1 Interfaz del listado de los dispositivos Bluetooth

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <TextView android:id="@+id/title_paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Escoja el modulo Bluetooth:"
        android:visibility="gone"
        android:background="#666"
        android:textColor="#fff"
        android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stackFromBottom="false"
        android:layout_weight="1"
    />
    <TextView
        android:id="@+id/connecting"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView
        android:id="@+id/infoText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="If no devices are listed please pair your device in Android settings"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:visibility="invisible" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center">
    </LinearLayout>
</LinearLayout>

```

```

</LinearLayout> <?xml version="1.0"
encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <TextView android:id="@+id/title_paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Escoja el modulo Bluetooth:"
        android:visibility="gone"
        android:background="#666"
        android:textColor="#fff"
        android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stackFromBottom="false"
        android:layout_weight="1"
    />
    <TextView
        android:id="@+id/connecting"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView
        android:id="@+id/infoText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="If no devices are listed please pair your device in Android settings"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:visibility="invisible" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center">
    </LinearLayout>
</LinearLayout>

```

### 2.4.2 Interfaz del listado de los dispositivos Bluetooth

Interfaz del mapa:

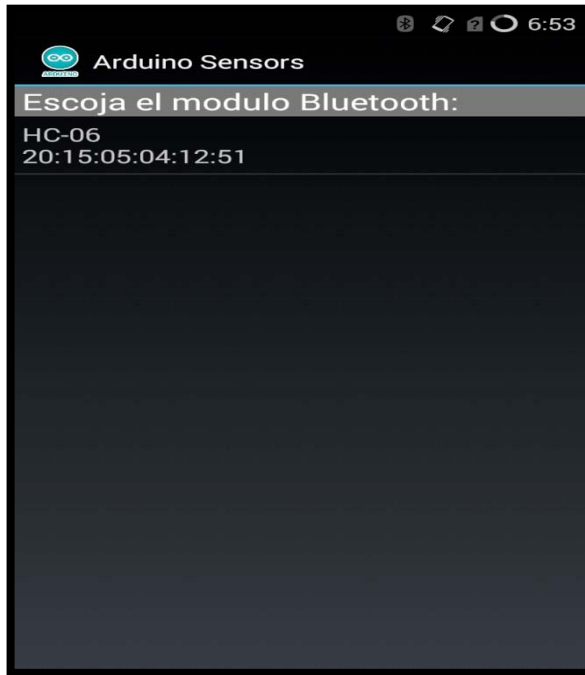


Figura 4. Pantalla del listado de los dispositivos para la conexión Bluetooth.

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/final-map"
    android:visibility="visible">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/pelotita"
        android:src="@drawable/ball"
        android:layout_marginRight="119dp"
        android:layout_alignTop="@+id/testView1"
        android:layout_centerHorizontal="true" />

```

```

<TextView
    android:id="@+id/testView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/txtString"
    android:text=""
    android:textSize="15sp" />
<TextView
    android:id="@+id/txtString"
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:layout_alignLeft="@+id/testView1"
    android:layout_alignParentBottom="true"
    android:text=""
    android:textSize="15sp" />
<Button
    android:id="@+id/buttonOn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="LED ON"
    android:visibility="invisible"
    android:enabled="false" />
<Button
    android:id="@+id/buttonOff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/buttonOn"
    android:layout_alignParentRight="true"
    android:text="LED OFF"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/sensorView0"
    android:layout_centerHorizontal="true"
    android:text="Sensor 1 Voltage =
    ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/sensorView1"

```

```

    android:layout_centerHorizontal="true"
    android:text="Sensor 2 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/sensorView2"
    android:layout_centerHorizontal="true"
    android:text="Sensor 3 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="196dp"
    android:text="Sensor 0 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/sensorView3"
    android:text="Sensor 5 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView
    android:id="@+id/sensorView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/sensorView4"
    android:text="Sensor 6 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible" />
<TextView

```

```

    android:id="@+id/sensorView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/sensorView5"
    android:text="Sensor 7 Voltage = ?????"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:visibility="invisible"
    android:enabled="false" />
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/pelotaroja"
    android:src="@drawable/redball"
    android:layout_below="@+id/sensorView1"
    android:layout_alignEnd="@+id/buttonOn"
    />
</RelativeLayout>

```

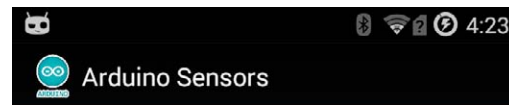


Figura 5. Pantalla con el mapa virtual y posición del usuario.

### 3. Maqueta prototipo del establecimiento

#### 3.1 Diseño del establecimiento

El prototipo consiste en un edificio en general compuesto de dos pasillos principales y otro pequeño que los conecta, en los cuales se colocarán los sensores ultrasonidos que medirán la distancia de la persona y detectarán su posición en el mapa. Estos pasillos guían a través de 5 habitaciones en donde se posicionan sensores infrarrojos y fotoceldas para detectar si las habitaciones están ocupadas o vacías.

#### 3.2 Bosquejo de la maqueta

De la figura 6 a la 8 se muestra el plano y la vista 3D de la maqueta.

#### 3.3 Selección de pasillos

Se tendrán tres (3) pasillos:

##### 3.3.1 Pasillo principal

Las personas accederán la entrada y entrarán al pasillo principal, se podrá encontrar la recepción en el pasillo y dará acceso a otras habitaciones como salones, baños y laboratorios. También dará acceso a un segundo pasillo de administrativos



Figura 6. Plano de la maqueta 1.



Figura 7. Vistas 3D de la maqueta 1.

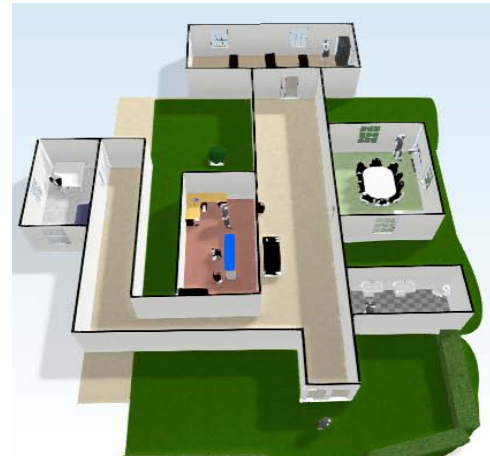


Figura 8. Vistas 3D de la maqueta 1.

##### 3.3.2 Pasillo de administrativos

Es el pasillo en donde se podrá acceder a las oficinas de los administrativos.

##### 3.3.3 Pasillo de oficinas

Son pasillos individuales que conectarán a las oficinas directamente. Para acceder a este pasillo se requiere pasar por el pasillo administrativo.

#### 3.4 Selección de habitación

Habrán 5 habitaciones, 4 a las que se pueden entrar y 1 a las que no:

Baño: Habitación para la higiene personal.

Salón de clase: Habitación para dar clases por los facilitadores y recibida por los estudiantes.

Laboratorio: Laboratorio de cómputo en donde se impartirán clases y se desarrollarán laboratorios y proyectos como programación, edición, modelado en 3D, etc.

Oficina: Oficina de un administrativo, en este caso un facilitador de la universidad. Se podrá visitar siempre y cuando el facilitador se encuentre y esté disponible.

Recepción: Toda información que se quiera saber sobre el establecimiento se podrá encontrar aquí, al igual que búsqueda de documentos y pagos. No se podrá acceder ya que es solo para administrativos y se atiende a la gente por ventanilla.

#### 3.5 Tamaño de la maqueta

La maqueta tiene una base de 40 x 32 pulgadas y un alto de 4 pulgadas. Se requiere de una maqueta grande para que los sensores de



ultrasonido puedan detectar objetos a distancia en los pasillos. El ancho de los pasillos es:

Pasillo principal: 15 cm, a este ancho el sensor de ultrasonido llega a detectar hasta 95 cm.

Pasillo de administrativos: 10 cm, a este ancho el sensor de ultrasonido llega a detectar hasta 54 cm.

Pasillo de oficinas: 8 cm, a este ancho el sensor de ultrasonido llega a detectar hasta 25 cm.

### 3.6 Diseño final de la maqueta

Ver figuras 9 y 10.



Figura 9. Vista superior de la maqueta.



Figura 10. Vista frontal de la maqueta.

### 3.7 Circuito - bosquejo del circuito

Los pines del Arduino, digital y analógicos, van conectados a un *protoboard* y del *protoboard* pasan los diferentes componentes y sensores. Los sensores de ultrasonidos estarán conectados en un mini *protoboard* para darle soporte y una base. Ver figura 11.

El módulo Bluetooth también estará conectado a un mini *protoboard*. El sensor PIR, LED y la fotocelda estarán conectados por cables con cabeza hembra, esto les facilita la conexión con las patitas.

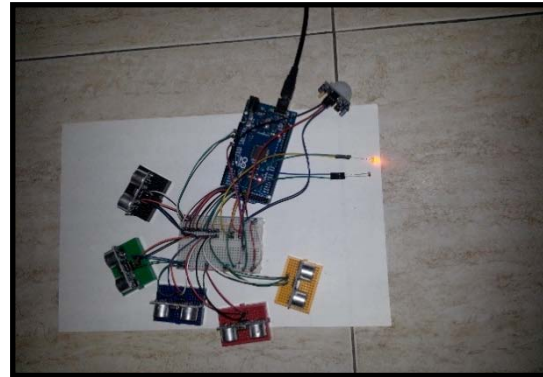


Figura 11. Bosquejo del circuito.

### 3.8 Selección de sensores y componentes

Se utilizarán 2 sensores, 1 componente electrónico y 1 módulo:

Sensor de ultrasonido: Es un sensor de proximidad que utiliza sonido y eco para detectar la distancia entre el objeto y el sensor.

Sensor PIR: Es un sensor de detección por infrarrojo, detecta si hay movimientos cerca del sensor.

Fotocelda: Es una resistencia dependiente de la luz, entre más intensa sea la luminosidad menor será la resistencia.

Módulo Bluetooth: Módulo que le permitirá al Arduino conectarse con otros dispositivos con Bluetooth. En este caso se utilizará para conectarse a un dispositivo Android.

*Protoboard*: Protoboard para conectar los pines del Arduino con los diferentes componentes y sensores.

Mini *protoboard*: Se utilizan mini *protoboards* para conectar los sensores y darles soporte.

### 3.9 Selección de cables

Se escogieron tres (3) tipos de cables:

Cable macho a macho; cable macho a hembra y cable hembra a hembra.

Estos cables, que se muestran en la figura 11, son más fáciles de manejar que los cables de cobre y dan una mejor presentación, no se requiere hacer soldaduras entre cables o componentes por la facilidad de entrada de la cabeza de los cables. Ver figura 12.



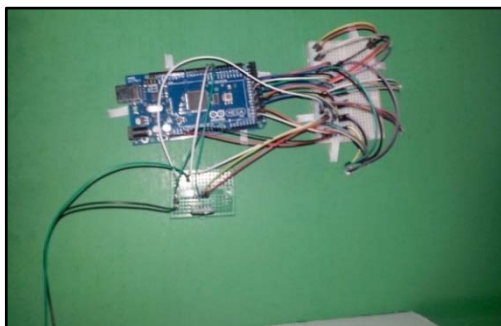
**Figura 12.** De arriba hacia abajo, macho a macho, hembra a hembra y hembra a macho.

### 3.10 Creación de la maqueta final

#### 3.10.1 Conexión del Arduino

Se utilizaron en total 12 pines digitales del Arduino, 1 pin analógico, 2 pines de comunicación serial, pin de 5V y el pin GND. 10 pines digitales se utilizaron en los sensores de ultrasonido, cada sensor utiliza 2 pines. Los otros 2 pines digitales fueron usados en los sensores PIR. El pin analógico se utilizó para la fotocelda, ya que requiere de una entrada analógica para capturar sus valores. Ver figura 13.

Los pines de comunicación serial utilizados fueron RX y TX de los pines digitales 0 y 1, estos pines se conectaron al módulo Bluetooth. El pin de 5V fue utilizado para darle voltaje a cada sensor y componente conectado al Arduino, y como siempre para todo circuito eléctrico habrá una conexión a tierra o este caso el pin GND.



**Figura 13.** Circuito conectado entre el Arduino y los sensores.

#### 3.10.2 Conexión de los sensores

Los sensores se conectaron en diferentes lugares en la maqueta, los sensores de ultrasonido se colocaron en el pasillo principal, pasillo de administrativo, el pasillo hacia las oficinas administrativas, el laboratorio y el salón de clases. Los sensores PIR se colocaron en la oficina administrativa y el baño.

La fotocelda fue colocada en la oficina administrativa, su utilidad es para señalar si el profesor está disponible o no. El módulo Bluetooth se colocó junto al Arduino ya que su función es realizar la comunicación entre el Arduino y el dispositivo móvil.

#### 3.10.3 Circuito Final

Una vez terminada la maqueta, se montó el Arduino, los sensores de ultrasonido, los sensores PIR, la fotocelda y el módulo Bluetooth. Todos estos componentes se conectaron con cables de distintas cabezas, estos cables se pasaron debajo de la maqueta y todos están conectados del Arduino a un *protoboard* principal que conectan a los sensores y otros componentes. Ver figura 14.



**Figura 14.** Circuito completo montado en la maqueta.

## 4. Conclusiones

El uso de la plataforma Arduino empezó desde el código más simple como es encender un LED y luego hacer que parpadee, pero con su constante utilización e investigación logramos adquirir experiencia y habilidades desde

desarrollar un proyecto que origina música a través de bocinas con librería de música a crear un carrito con servomotores y la creación de este proyecto.

Su uso tiene infinidad de posibilidades y diferentes proyectos desde simples hasta complejos, todo depende de la creatividad y necesidad del desarrollador y del apoyo del código abierto de la comunidad Arduino.

Adquirimos experiencia manejando el comportamiento de la placa Arduino utilizando los sensores de ultrasonidos para dar la localización de un objeto y conociendo diferentes tipos de sensores con diferentes funcionalidades pero utilizándolos con el mismo objetivo como fue el sensor PIR o infrarrojo.

También le dimos un nuevo uso a las fotoceldas, como control de posición, si se da una falta de luz esto quiere decir que hay alguien cerca.

Existen millones de componentes que se puede utilizar en el Arduino para hacer diferentes tipos de experimentos y proyectos como robots, casas inteligentes, modelos digitales a escalas, entre otros.

Nos familiarizamos con el entorno de programación del SDK Android Studio, el cual nunca habíamos probado, al igual que el manejo de librerías externas de Android, aprendimos conocimientos básicos del lenguaje XML para el diseño de las pantallas de la aplicación y la interfaz de usuario.

Utilizamos los conocimientos adquiridos de programación orientada a objeto para entender el código Java y programar las funcionalidades de la interfaz y recibir los datos por medio de comunicación serial entre el Arduino y la Aplicación en Android.

En la ejecución de las distintas aplicaciones de prueba, aprendimos sobre los distintos errores de ejecución y compilación de Android Studio, así como las distintas versiones de las APIs del Sistema Operativo Android en varios dispositivos móviles.

En este prototipo a pequeña escala se necesitó de muchas pruebas para calibrar los distintos sensores utilizados y que los datos transmitidos entre Arduino y Android fueran correctos para que la sincronización fuera lo más cercano a la precisión requerida.

## 5. Recomendaciones

Para desarrollar una aplicación Android se recomienda primero aprender Java para la programación lógica y sus funciones. Tener conocimientos en el HTML para el diseño de la interfaz de la aplicación.

App Inventor es una buena opción para desarrollar aplicaciones en Android, es más sencillo de usar y no requiere de muchas habilidades de programación ni se requiere de Java, es recomendable para principiantes.

El Arduino es un aparato fácil de conseguir y fácil de usar, se puede aprender rápido viendo y leyendo tutoriales.

Se recomienda explorar el mundo del Arduino y publicar los proyectos realizados porque se pueden enriquecer muchos proyectos con la ayuda de toda la comunidad Arduino.

Se recomienda probar todo tipo de sensores para el Arduino, algunos sensores se le pueden dar varias funcionalidades y esas funcionalidades para diferentes tipos de proyectos, desde lo simple se llegan a utilizarse en lo más complejo.

La interfaz de usuario puede ser mucho más amigable, si se tienen más conocimientos y experiencia en el lenguaje XML y se continúa agregando funcionalidades a la app o se amplía el modelo a más habitaciones dependiendo del tamaño del edificio.

Al igual que utilizando más sensores o placas de Arduino pueden obtenerse mejor precisión en la posición del usuario dentro del modelo.

## REFERENCIAS

- [1] Adafruit. (14 de septiembre de 2015). How PIRs Work. Obtenido de adafruit: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
- [2] adafruit. (14 de septiembre de 2015). Overview. Obtenido de adafruit: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview>
- [3] Android. (25 de septiembre de 2015). BluetoothAdapter. Obtenido de developer.android: <http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html>
- [4] Arduino. (25 de septiembre de 2015). Arduino / Genuino MEGA 2560. Obtenido de arduino.cc: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

*Espinosa (et al): Prototipo de un mapa virtual para la orientación de las personas en un establecimiento público empleando tecnología Arduino*

- [5] Arduino. (9 de septiembre de 2015). Ping Ultrasonic Range Finder. Obtenido de arduino.cc: <https://www.arduino.cc/en/Tutorial/Ping?from=Tutorial.UltrasonicSensor>
- [6] Arduino. (9 de septiembre de 2015). pulseIn(). Obtenido de arduino.cc: <https://www.arduino.cc/en/Reference/PulseIn>
- [7] Arduino. (4 de septiembre de 2015). shiftIn(). Obtenido de arduino.cc: <https://www.arduino.cc/en/Reference/ShiftIn>
- [8] Arduino. (4 de septiembre de 2015). shiftOut(). Obtenido de arduino.cc: <https://www.arduino.cc/en/Reference/ShiftOut>
- [9] Jeremy Blum. (17 de septiembre de 2015). Tutorial 4 for Arduino: Analog Inputs. Obtenido de jeremyblum: <http://www.jeremyblum.com/2011/01/24/arduino-tutorial-4-analog-inputs/>
- [10] NXP. (4 de septiembre de 2015). 74HC595; 74HCT595. Obtenido de nxp: [http://www.nxp.com/documents/data\\_sheet/74HC\\_HCT595.pdf](http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf)
- [11] Shuster, K. (14 de septiembre de 2015). SVG Rendering on Android. Obtenido de KioDev: <http://www.kiodev.com/svg-rendering-on-android/>
- [12] Sparkfun. (4 de septiembre de 2015). SN74HC595. Obtenido de sparkfun: <https://www.sparkfun.com/datasheets/IC/SN74HC595.pdf>
- [13] Sparkfun. (16 de septiembre de 2015). SparkFun Audio Bluetooth Breakout - RN-52. Obtenido de sparkfun: <https://www.sparkfun.com/products/12849>
- [14] Sparkfun. (16 de septiembre de 2015). SparkFun Audio Bluetooth Breakout - RN-52. Obtenido de sparkfun: <https://www.sparkfun.com/products/11924>
- [15] Sparkfun. (16 de septiembre de 2015). SparkFun BLE Mate 2. Obtenido de sparkfun: <https://www.sparkfun.com/products/13019>
- [16] Sparkfun. (16 de septiembre de 2015). SparkFun Bluetooth Mate Gold. Obtenido de sparkfun: <https://www.sparkfun.com/products/12580>
- [17] Tinkernut. (16 de septiembre de 2015). ARDUINO BLUETOOTH BASICS. Obtenido de tinkernut: <http://www.tinkernut.com/2014/03/arduino-bluetooth-basics/>
- [18] Toyo, P. (10 de septiembre de 2015). SVG Integration in Android Studio. Obtenido de android-coffe: <http://android-coffee.com/svg-integration-in-android-studio/>
- [19] WinGoodHarry. (25 de septiembre de 2015). Android Send/Receive data with Arduino using Bluetooth – Part 2. Obtenido de wingoodharry.wordpress: <https://wingoodharry.wordpress.com/2014/04/15/android-sendreceive-data-with-arduino-using-bluetooth-part-2/>