

MONITOREO DE FUENTE DE ENERGIA SOLAR (SOLAR 12 HORAS)

Francisco Vázquez Guzmán

Tecnológico Nacional de México/Instituto Tecnológico de Tehuacán
fvazquez@ittehuacan.edu.mx

Antonio Vázquez Aguilar

Tecnológico Nacional de México/Instituto Tecnológico de Tehuacán
avaguilar@hotmail.com

Liliana Elena Olguín Gil

Tecnológico Nacional de México/Instituto Tecnológico de Tehuacán
lolguing@gmail.com

Eduardo Vázquez Zayas

Tecnológico Nacional de México/Instituto Tecnológico de Tehuacán
zayas_91@hotmail.com

Félix Román Joaquín

Alumno del Instituto Tecnológico de Tehuacán
felixromanyuka@gmail.com

Julio Cesar López Hernández

Tecnológico Nacional de México/Instituto Tecnológico de Tehuacán
jc_lopezhernandez@outlook.com

Resumen

La tendencia en el uso de los recursos naturales para generar energía nos conduce a adoptar nuevos medios que nos den la funcionalidad requerida sustituyendo los generadores de energía convencionales con generadores sustentables que obtengan la energía de recursos como el aire o la luz solar. El objetivo que se trazó fue desarrollar una aplicación móvil de monitoreo, utilizando un módulo wifi que envíe datos al móvil a través de una conexión a

internet para la automatización de un panel solar en el que se obtengan temperatura y humedad del medio que lo rodea, captación solar y detección de fallas.

Los resultados parciales fueron: Construcción de un prototipo de panel solar con desplazamiento seguidor del sol. Diseño e implementación de hardware y software seguidor del sol. Diseño e implementación de aplicación móvil de control y monitoreo del módulo seguidor.

Palabra(s) Clave(s): *Panel, seguidor y monitoreo.*

Abstract

The trend in the use of natural resources to generate energy leads us to adopt new means that give us the functionality required by replacing conventional generators with sustainable generators that obtain energy from resources such as air or sunlight.

The objective was to develop a mobile monitoring application using a wifi module that sends data to the mobile through an internet connection for the automation of a solar panel in which the temperature and humidity of the surrounding environment are obtained, solar pickup and fault detection.

The partial results were: Construction of a prototype of solar panel with follower the sun. Design and implementation of sun tracking hardware and software. Design and implementation of mobile monitoring and monitoring application of the follower module.

Keywords: *Panel, follower and monitoring.*

1. Introducción

En los últimos tiempos se ha comenzado a dar una mayor importancia a las energías limpias, siendo éstas obtenidas mediante la utilización de recursos renovables, generalmente provenientes de la naturaleza, para poder transformarlas en energía eléctrica que puede ser utilizada en el hogar o en cualquier instalación de la red eléctrica de un edificio, institución o entorno en particular.

Una de las fuentes de energías renovables es la energía solar, debido a que la tecnología necesaria para poder ser utilizada es la instalación de paneles solares fotovoltaicos [Díaz & Carmona, 2010], láminas de variado espesor que se encargan de capturar la radiación lumínica solar y transformarla en un recurso utilizable. Para poder utilizar esta fuente energética es necesario contar con un espacio para poder instalar los paneles solares, siendo generalmente utilizados los techos o terrazas, aunque también se da el emplazamiento de los denominados parques solares, que consisten en instalaciones a gran escala y utilizados para brindarles energía eléctrica a una gran cantidad de hogares como también abastecer a ciudades enteras. A todo esto, se le suma el software de monitoreo que es el encargado de automatizar el panel solar, registrar datos estadísticos y demás. En nuestro país, existen empresas distribuidoras de este tipo de sistemas, las cuales ofrecen dispositivos similares.

El desarrollo del software de monitoreo para el presente proyecto está basada en un sistema operativo móvil de código abierto. El sistema de monitoreo es relevante ya que se puede obtener información para observar el presupuesto de consumo, la captación solar y ayuda con la detección de fallas.

2. Métodos

Durante esta etapa se contemplaron las actividades de construcción del prototipo de panel solar, pruebas con panel fijo y móvil, análisis de requerimientos para módulo de monitoreo, diseño e implementación de software de monitoreo y pruebas.

Para el diseño del software se contemplaron dos alternativas de software, la programación en el microcontrolador Arduino [Arduino, 2016] y la programación de la aplicación móvil de Android analizando los requisitos mínimos necesarios para poder ejecutar cada una de las herramientas de software. Los requisitos a considerar fueron los siguientes:

- Voltaje necesario con el que operan.
- Listado de comandos para configurar.
- Conexión de los pines.

- Un sensor DHT11 que mide la temperatura y humedad del medio que rodea. Funciona con su propia librería para el correcto funcionamiento en el sistema.
- Un módulo wifi ESP8266 que se configura por comandos AT, que son instrucciones codificadas que conforman un lenguaje de comunicación.

Se analizó a detalle el software servidor donde se almacenan los datos del sistema. Este servidor es de código abierto teniendo así la opción de descargar el código fuente y poder instalarlo en un servidor local.

Una vez estudiado los componentes que conforman al sistema de monitoreo, se estructuró la forma en que operan conjuntamente para lograr obtener datos en tiempo real.

Posteriormente se diseñó el diagrama de conexión entre el sensor DHT11, módulo wifi ESP8266 y el microcontrolador Arduino UNO, ver figura 1.

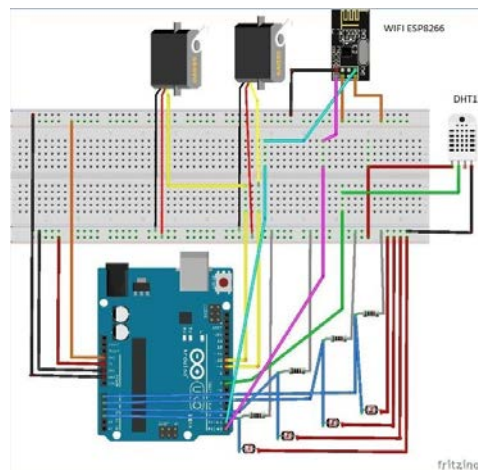


Figura 1 Diagrama de conexión del sistema.

El diseño de este diagrama actúa de la siguiente manera:

- EL sensor DHT11 está conectado al pin que alimenta con un voltaje de 5 V.
- El módulo wifi ESP8266 está conectado al pin que alimenta con un voltaje de 3.3 V.
- El diseño en que opera el sistema de monitoreo dispone del funcionamiento de tres conexiones; el módulo wifi con el arduino y el DHT11, el servidor y el

módulo wifi y por último la aplicación móvil conectada al servidor. En la figura 2, se observa que el módulo wifi es el encargado de enviar los datos en tiempo real al servidor, este servidor es el que provee los datos que solicita la aplicación instalada en el móvil.

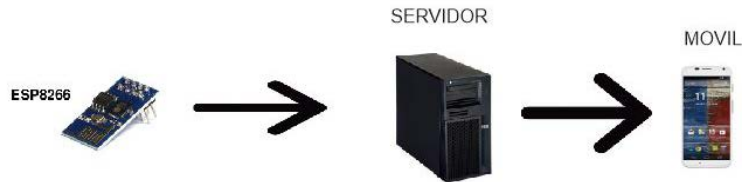


Figura 2 Diagrama de funcionamiento de sistema de monitoreo.

Para los datos de humedad y temperatura del medio ambiente se utilizó un sensor denominado DHT11 el cual se conectó al microcontrolador Arduino UNO para recibir los datos en tiempo real. El módulo Wifi se configuró a través del microcontrolador Arduino, existiendo así una conexión entre el módulo Wifi, el sensor DHT11 y el servidor. Para el diseño y desarrollo de la aplicación móvil, el primer paso fue la creación del splash screen el cual contiene el logo del sistema elaborado en la IDE de Android Studio, ver figura 3.



Figura 3 Visualización del splash screen.

Para la creación de este splash screen se insertó un nuevo layout y se configuró el ImageView con la imagen del logo, ver figura 4.

```
splash_activity.xml x
<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context=".SplashScreenActivity" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:contentDescription="SOLAR 12hrs"
        android:scaleType="fitXY"
        android:src="@drawable/logo" />

</RelativeLayout>
```

Figura 4 Código fuente de splash screen.

Para la configuración en el servidor externo, el primer paso fue configurar los servicios mediante el registro de usuario correspondiente, ver figura 5.

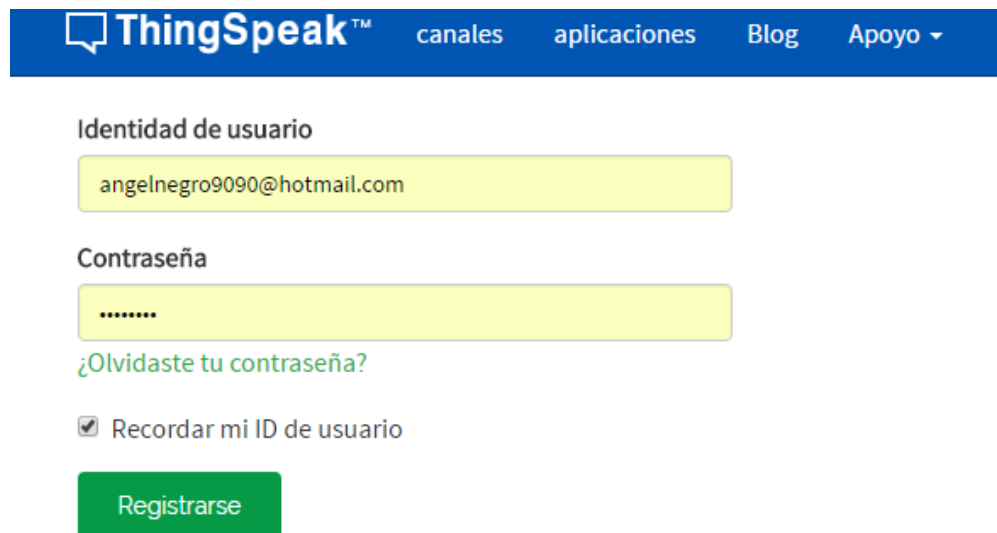


Figura 5 Registro de usuario en servidor.

Una vez registrado se creó un nuevo canal en el servidor, ver figura 6. En este canal se crearon dos campos, Field 1 y Field 2 los cuales sirvieron para el almacenamiento de los datos de humedad y temperatura, ver figura 7.

The screenshot shows the ThingSpeak interface for a channel named 'Datosdth11'. At the top, there is a navigation bar with 'Channels', 'Apps', 'Blog', and 'Support'. Below the channel name, the following information is displayed: Channel ID: 117216, Author: FELIXROMAN, and Access: Public. There are tabs for 'Private View', 'Public View', 'Channel Settings', 'API Keys', and 'Data Import /'. The 'Channel Settings' tab is active, showing a 'Percentage complete' of 50%. The 'Channel ID' is 117216. The 'Name' field contains 'Datosdth11'. The 'Description' field contains 'lectura de temperatura y humedad'. There are two 'Field' entries: 'Field 1' with 'Field Label 1' and a checked checkbox, and 'Field 2' with 'Field Label 2' and a checked checkbox.

Figura 6 Creación de un canal en el servidor.

This is a close-up view of the 'Channel Settings' form. It shows the 'Description' field with the text 'lectura de temperatura y humedad'. Below it are two 'Field' entries: 'Field 1' with 'Field Label 1' and a checked checkbox, and 'Field 2' with 'Field Label 2' and a checked checkbox.

Figura 7 Campos de un canal.

Uno de los pasos más importantes es obtener una clave para la conexión entre el módulo wifi y el servidor. En la figura 8 se muestra la clave proporcionada por el servidor.

The screenshot shows the 'Write API Key' page in ThingSpeak. At the top, there are tabs for 'Private View', 'Public View', 'Channel Settings', 'API Keys', and 'Data Import /'. The 'API Keys' tab is active. The page title is 'Write API Key'. Below the title, there is a 'Key' field containing the alphanumeric string 'D612UOCNL26IXKJS'. At the bottom, there is an orange button labeled 'Generate New Write API Key'.

Figura 8 Clave API KEY del servidor.

Al realizar la configuración del servidor y obtener la clave de conexión se realizó la codificación en el IDE Arduino para el funcionamiento de envío de datos en tiempo real al servidor.

Se definió la variable GET Y FIELD2, la primera contiene la clave obtenida por el servidor para poder realizar la conexión entre el módulo wifi y el servidor, la segunda hace referencia a los datos que son de humedad y temperatura enviadas al servidor, ver figura 9.

```
String GET = "GET /update?key=D612UOCNL26IXKJS&field1=";  
String FIELD2 = "&field2=";
```

Figura 9 Configuración del servidor en IDE Arduino.

En el método updateTemp se crea la comunicación con el servidor a través del protocolo de comunicación TCP y envía los datos de temperatura y humedad, ver figura 10.

```
bool updateTemp(String tenmpF, String humid){  
  
    String cmd = "AT+CIPSTART=\"TCP\", \"\"";  
    //agrega la direcc IP y el puerto  
    cmd += IP;  
    cmd += "\",80";  
    //connect  
    Serial.println(cmd);  
    delay(2000);  
    if(Serial.find("Error")){  
        return false;  
    }  
    cmd = GET;  
    cmd += tenmpF;  
    cmd += FIELD2;  
    cmd += humid;  
    cmd += "\r\n";  
    Serial.print("AT+CIPSEND=");  
    Serial.println(cmd.length());  
    if(Serial.find(">")){  
        //send through command to update values  
        Serial.print(cmd);  
    }else{  
        Serial.println("AT+CIPCLOSE");  
    }  
}
```

Figura 10 Código fuente del método updateTemp.

Este método recibe dos parámetros los cuales son los datos de temperatura y humedad (figura 11). Una vez recibidos se crea la conexión con el servidor y se envían en tiempo real estos parámetros.

```
bool updateTemp(String tempF, String humid){
```

Figura 11 Parámetros recibidos del sensor DHT11.

Al tener configurado el servidor y en funcionamiento el módulo wifi, se codificó el software de monitoreo el cual consistió en el desarrollo e implementación de la aplicación móvil. Se añade una nueva clase de java la cual se ocupa de la gestión del splash screen, ver figura 12.

```
public class SplashScreenActivity extends Activity{  
  
    // Duracion del splash screen  
    private static final long SPLASH_SCREEN_DELAY = 3000;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        //orientacion  
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);  
        // Ocultar barra de titulo  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
  
        setContentView(R.layout.splash_activity);  
  
        TimerTask task = () -> {  
  
            // inicia la sig actividad  
            Intent mainIntent = new Intent().setClass(  
                SplashScreenActivity.this, MainActivity.class);  
            startActivity(mainIntent);  
  
            // Cerrar la actividad por lo que el usuario no será capaz de volver este  
            // Actividad pulsando el botón Atrás  
            finish();  
        };  
  
        // simular un proceso de tiempo al iniciar la app  
        Timer timer = new Timer();  
        timer.schedule(task, SPLASH_SCREEN_DELAY);  
    }  
}
```

Figura 12 Código fuente gestión de splash screen.

En el Layout principal se inserta un webview el cual cargará el canal de ThingSpeak.

3. Resultados

Para la prueba con panel fijo, se registraron datos de voltaje generados por el panel solar sin la implementación del módulo seguidor. Para ello, el panel fue colocado sobre una casa para poder hacer mediciones durante el día domingo 29 de mayo de 2016, de 7:30 am a 7:50 pm. Fue un día despejado y un poco nublado durante las últimas horas de la tarde, ver figura 13.



Figura 13 Prueba de panel fijo a la intemperie.

Para la comunicación con el software servidor, se practicó con un teléfono celular con sistema operativo Android 5.1 con procesador de 4 núcleos y 2 GB de RAM, obteniendo así una mejor fluidez y rapidez de la aplicación para ejecutarla.

Se programó el software de monitoreo para la automatización del panel solar en el IDE de Arduino, se hicieron pruebas para comprobar la conexión del sistema con el servidor que realmente estuviera enviando datos en tiempo real a través de una conexión wifi.

Se construyeron los soportes de los actuadores, sensores y del circuito principal, haciendo cortes y uniones de piezas de aluminio en las que se colocarían los servomotores [D. Vallejo, 24]. El movimiento del panel está supeditado por la acción de los servomotores en el plano vertical y horizontal; el peso soportado en

el plano horizontal fue manejado con facilidad a diferencia del plano vertical donde el peso del panel impedía un movimiento fluido, por lo que se optó por el uso de un servomotor más potente en ese plano, ver figura 14.

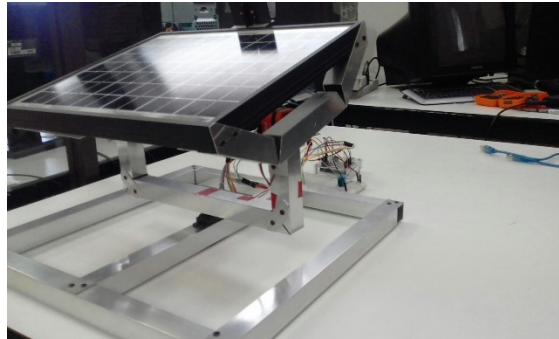


Figura 14. Prototipo del sistema.

Se diseñó la estructura de un canal en el servidor (figura 15), el cual muestra los datos enviados a través del módulo Wifi en tiempo real en modo gráfico.

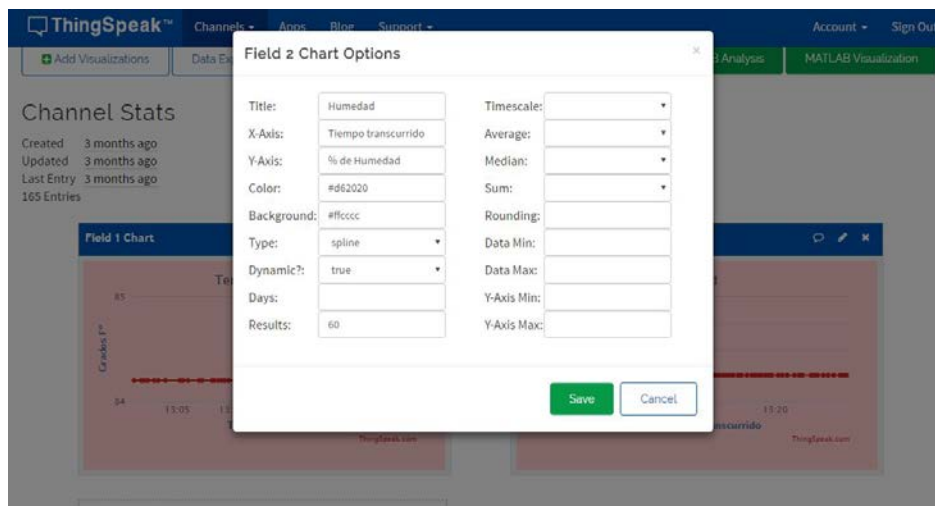


Figura 15 Diseño de un canal en el servidor ThingSpeak.

El software servidor proporciona una clave de acceso denominada “api key” la cual se utiliza para poder tener la conexión que enviará los datos al canal, tanto para el Arduino como para la aplicación móvil en Android Studio.

Posteriormente se realizó la implementación entre la conexión del módulo wifi y el sensor DHT11 conectados al microcontrolador Arduino UNO, para verificar su

correcto funcionamiento. Una vez configurado el modulo se realizó el envío de datos del sistema en tiempo real al servidor, capturando dichos datos en gráficas. Por último, se implementó la aplicación móvil la cual, al ser ejecutada, realizó correctamente la conexión con el canal del servidor, logrando la visualización de los datos requeridos del sistema en tiempo real.

La aplicación móvil se desarrolló en el IDE Android Studio mediante el lenguaje de programación java y XML para la codificación de la aplicación. Para la interfaz de la aplicación se dispuso de un diseño simple para su fácil uso, un splash-screen mostrando el logotipo del sistema e inmediatamente la información de los datos se carga en tiempo real, pudiendo así deslizar con el dedo las gráficas de temperatura y humedad (ver figura 16).



Figura 16 Diseño de interfaz de la aplicación móvil.

Se comprobó el funcionamiento del proyecto en un ambiente cerrado y al exterior, obteniendo en tiempo real los datos correctos del sistema establecidos en el objetivo general de nuestro proyecto. También se probó con registros de datos generados por el panel. Se obtuvo una correcta automatización del sistema fotovoltaico, logrando que el panel solar estuviera siempre captando la máxima energía proporcionada por los rayos solares, así como datos en tiempo real de la captación solar, temperatura y humedad, del medio que rodea al sistema.

4. Discusión

En el contexto, se debe contemplar que la Comisión Federal de Electricidad y la Comisión Regulatoria de Electricidad, establecen un marco normativo específico para la interconexión de tecnologías basadas en fuentes renovables, como los sistemas fotovoltaicos que eviten molestias o daños a otros usuarios.

Es por ello que surge la necesidad de crear sistemas que ayuden a minimizar el uso indiscriminado de los recursos naturales, además de disminuir el costo en la factura de energía eléctrica.

El aumento en los precios del consumo de energía que se han dado en la actualidad, afectan la economía de la población en general, para lo cual el generador reduciría el consumo de energía eléctrica, ayudando a economizar a las familias.

En México, los generadores de energía fotovoltaica que existen en el mercado no cuentan con un sistema de monitoreo a través de una aplicación móvil. Al no generarse registros estadísticos, se dificulta la elaboración de planes de mantenimiento del generador. También se debe considerar que los paneles solares fijos no aprovechan de manera eficiente la captación de energía solar. Al integrarse el sistema de seguimiento se consigue mayor precisión en la movilidad del dispositivo, captando mayor cantidad de energía solar.

El material que se utilizó para la estructura del prototipo fue aluminio y se optó por un perfil cuadrado. A la celda solar se le añadieron soportes en sus lados cortos, estos soportes se unen a una horquilla que lleva integrado un servomotor de 53 kg/cm para mover el panel solar en un plano vertical. La horquilla se acopla a otro servomotor de 10 kg/cm el cual se encuentra empotrado a una base fija, y esto permitió mover el panel solar en un plano horizontal, logrando así el movimiento del sistema en dos ejes. En uno de los costados largos del panel, se fijó el arreglo de cuatro sensores fotoresistivos los cuales permitieron determinar el punto en el cielo con mayor luminosidad para orientar el panel solar hacia la fuente de luz.

Por medio de una tarjeta Arduino, se capturaron las mediciones de los sensores, se procesaron y se enviaron las órdenes a los servomotores para que estos orienten al panel perpendicular a la fuente luminosa.

A la tarjeta Arduino se le agregó un módulo de comunicación Wi-Fi; esto permitió el desarrollo de una aplicación que ejecutándose en un dispositivo Android, fuera capaz de comunicarse con un servidor, el cual, registra los datos estadísticos del sistema. Dicha aplicación tiene la capacidad de recibir los datos y enviarlos al servidor web en tiempo real, es decir con tiempos menores a 6 segundos. Este aspecto fue un componente fundamental para el desarrollo del sistema de monitoreo.

A través de la aplicación móvil se visualizaron los datos que recibe el servidor Web en forma de gráfica. El hecho de tener los datos en un servidor Web brinda la posibilidad de monitorear el prototipo las 24 horas del día, en tiempo real, en cualquier lugar del mundo siempre y cuando se cuente con acceso a internet para acceder al mismo. Se guardaron las lecturas de conversión de energía durante determinado tiempo. Los datos permiten la toma de decisiones para efectuar programas de mantenimiento preventivo, además de determinar la eficiencia en el funcionamiento del dispositivo.

En un edificio con un gasto promedio de 700 kWh, se puede lograr un ahorro de hasta 135 kWh por bimestre. El usuario, tendrá un consumo de energía eléctrica amigable con el medio ambiente, a menor costo del sistema convencional, de acuerdo al programa de fomento de sistemas fotovoltaicos en México.

Su estado de desarrollo de acuerdo al Technology Readiness Level (TRL), es de nivel 6 porque se ha demostrado y modelado el prototipo en un ambiente punto a punto relevante. Es por ello que se concluye que la propuesta de un sistema monitoreo de fuente de energía solar es una propuesta real, viable y que puede impactar a la vivienda tradicional, al sector empresarial, a las escuelas y al gobierno estatal y federal.

5. Bibliografía y Referencias

- [1] Arduino. (2016). <https://www.arduino.cc/>
- [2] D. Vallejo, H. (24). MOTORES Y SERVOS PARA MINI-ROBÓTICA. *Saber Electrónica*, 61-64.

- [3] Diaz, T., & Carmona, G. (2010). Instalaciones solares fotovoltaicas. McGraw-Hill.
- [4] Electrónica Unicrom. (22 de junio de 2016). <http://unicrom.com/ldr-fotorresistencia-fotorresistor/>
- [5] Sandoval García, E. R. (Febrero de 2016). Revista Universitaria Digital de Ciencias Sociales. <http://www.cuautitlan.unam.mx/rudics/ejemplares/0102/art06.html>.