

Conectividad en redes inalámbricas de área personal utilizando Arduino y componentes electrónicos

***Connectivity in Personal Area Wireless Networks Using Arduino and Electronic
Components***

***Conectividade em redes sem fio de área pessoal usando Arduino e componentes
eletrônicos***

César Manuel Hernández Mendoza

Instituto Tecnológico Superior de Irapuato, México

cesar.hernandez@itesi.edu.mx

<https://orcid.org/0000-0003-4667-9816>

Juan Pablo Serrano Rubio

Instituto Tecnológico Superior de Irapuato, México

juserrano@itesi.edu.mx

<https://orcid.org/0000-0003-3705-5112>

Fernando De los Reyes Quiroz

Instituto Tecnológico Superior de Irapuato, México

fedelosreyes@itesi.edu.mx

<https://orcid.org/0000-0002-0284-9476>

Resumen

En la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Irapuato existen materias como Principios Eléctricos, Programación de Dispositivos Móviles y Fundamentos de Telecomunicaciones en las que la tecnología *bluetooth* tiene un papel trascendental, pues gracias a esta un profesionalista del área puede solventar diversos problemas de conectividad inalámbrica local a un precio accesible y de manera eficaz. En algunas ocasiones, sin embargo, el estudiante podía verse obligado a limitar sus habilidades cuando el temario avanzaba

y tenía que desarrollar prácticas de conectividad inalámbrica de mayor complejidad, por lo que recurría continuamente a fuentes de información en Internet que no garantizaban confiabilidad en la información expuesta, o bien evitaba investigar y desarrollar sus propios programas al descargar herramientas o aplicaciones genéricas de terceros en la Play Store, los cuales no eran los más adecuados o no se adaptaban a las necesidades técnicas de sus componentes.

En el contexto planteado, los aprendices no lograban dominar el fundamento técnico necesario al utilizar código y programación de páginas de Internet que eran aplicados a sensores y actuadores con los que no contaban, debido a la gran variedad que existe en el mercado, lo que repercutía en su rendimiento académico. Por lo que estudiantes y profesores desarrollaron una guía técnica para ayudar al alumno a realizar prototipos con transmisión de datos y reforzar aquellos conocimientos donde la tecnología *bluetooth* es un común denominador y así pueda aplicar las competencias específicas de las materias mencionadas.

Como resultado se ha logrado que los estudiantes trabajen en equipo fomentando el liderazgo, se han creado hábitos de estudio técnico-prácticos y elaborado prototipos sustentando la solución del algún problema citado. Además, se han generado *apps* de código abierto y fomentado la constante participación en eventos académicos, lo que ha traído la obtención de primeros lugares y algunas patentes, todo lo cual ha contribuido a mejorar su rendimiento académico.

Palabras clave: Android, Arduino, *bluetooth*, código abierto, dispositivo móvil, electrónica, HC05, programación, red inalámbrica local, rendimiento académico.

Abstract

In the Computer Systems Engineering career at the Higher Technological Institute of Irapuato there are subjects such as Electric Principles, Mobile Device Programming and Telecommunications Fundamentals, in which Bluetooth technology is of importance, since it is one of the technologies and electronic modules with which a professional in the area can solve various local wireless connectivity problems at an accessible and efficient price. Occasionally students could be forced to limit their skills when the scholar program progress and they had to develop more complex wireless connectivity practices, so they continually resorted to sources of information on the Internet that did not guarantee reliability in the exposed information as well as

they avoided researching and develop their own programs by downloading generic tools or applications from third parties in the Play Store, which were not the most appropriate or did not adapt to the technical needs of their components.

With the problems raised, the student could not master the technical foundation necessary to use code and programming of Internet pages that were applied to sensors and actuators that did not have, due to the wide variety that exists in the market, impacting on their academic performance. For this reason students and teachers have developed a technical guide in order to help them to make prototypes with data transmission to avoid this problem and reinforce the student's knowledge through the use of Bluetooth technology as a common denominator and means by which the student can apply the specific competences of the mentioned subjects.

As a result, students have been managed to work as a team, fostering leadership, technical-practical study habits have been created, prototypes have been developed, supporting the solution of aforementioned problem, the generation of open source mobile applications (apps), the constant participation in academic events with obtaining first places and some patents that have contributed to improve their academic performance.

Keywords: Android, Arduino, bluetooth, open source, mobile device, electronics, HC05, programming, local wireless network, academic performance.

Resumo

Na corrida de Sistemas de Computação Engenharia pelo Instituto Tecnológico Superior de Irapuato existem temas como Princípios Elétricos, Programação de dispositivos móveis e Fundamentos de Telecomunicações em tecnologia Bluetooth tem um papel vital, porque graças a este uma área profissional pode resolver vários Problemas de conectividade sem fio local a um preço acessível e de forma eficaz. Às vezes, porém, o aluno poderia ser forçado a limitar suas habilidades quando a agenda foi em movimento e teve que desenvolver a conectividade sem fio prática mais complexo, por isso continuamente recorreram a fontes de informação na Internet que não garantia informações confiabilidade expostos ou impedido de pesquisa e desenvolver seus próprios programas para download ferramentas ou aplicativos de terceiros genéricos na Play Store, que não foram adaptados ou não adaptados às necessidades técnicas dos seus componentes.

No presente contexto, os formandos foram capazes não dominar a base técnica necessária a utilização de código e programação de páginas da Internet que foram aplicadas para sensores e atuadores com que não contavam por causa da variedade que existe no mercado, que reverberou em seu desempenho acadêmico. Para que os alunos e professores desenvolveram um guia técnico para ajudar os alunos a fazer a transmissão de dados protótipos e reforçar essas habilidades onde a tecnologia Bluetooth é um denominador comum e, portanto, pode aplicar os poderes específicos dos materiais acima mencionados.

Como resultado, os alunos puderam trabalhar em equipe promovendo a liderança, criaram hábitos de estudo técnico-prático e desenvolveram protótipos que apóiam a solução de qualquer problema mencionado. Além disso, têm gerado aplicativos de código aberto e encorajou continuou participação em eventos acadêmicos, que trouxe a obtenção primeiros lugares e algumas patentes, todos os quais tem ajudado a melhorar seu desempenho acadêmico.

Palavras-chave: Android, Arduino, bluetooth, código aberto, dispositivo móvel, eletrônica, HC05, programação, rede sem fio local, desempenho acadêmico.

Fecha Recepción: Julio 2018

Fecha Aceptación: Diciembre 2018

Introducción

Algunas de las áreas involucradas en el proyecto para la generación de una guía técnica son la electrónica digital básica, la programación, los dispositivos móviles y las telecomunicaciones, las cuales forman parte de proyectos integradores con los que estudiantes y profesores han logrado colaborar y generar productividad. Asimismo, al fomentar el trabajo colaborativo se aplican competencias de investigación, comprensión y aprendizaje, pues “por ser una estrategia de investigación formativa no se orienta a la generación de un conocimiento objetivamente nuevo sino a la apropiación comprensiva de conocimientos ya elaborados pero nuevos para el estudiante” (Hewitt, 2007, p 236).

Específicamente la programación se llevó a cabo con Arduino:

Una plataforma de electrónica abierta, es decir que utiliza *hardware* libre para la creación de prototipos basada en *software* y *hardware* flexibles y fáciles de usar, su funcionamiento se basa en una placa con un micro controlador y un entorno de desarrollo (Llanga y Cano, 2014, p79).

Lo anterior aunado a que se trata de una de las herramientas más comunes con la que los estudiantes de nivel medio y nivel superior interactúan. Adquirir una de estas placas es sencillo y a un bajo costo: cualidades que contribuyen a que los estudiantes opten por adquirir una de ellas con el fin no solo de realizar actividades académicas, sino por iniciativa propia y para poder investigar e indagar más sobre algún área o tema que les interese. Sin duda esta herramienta incentiva y permite al estudiante obtener un perfil investigador de la ciencia y tecnología.

La programación es una de las razones de ser de un ingeniero en sistemas computacionales. Al respecto, es sabido que hay diferentes lenguajes, técnicas, certificaciones y especializaciones, entre ellas la programación de dispositivos móviles, la cual conlleva el uso de la programación orientada a objetos (POO) y de herramientas de código libre como Android, Arduino y otros entornos de acceso libre, por ejemplo el MIT App Inventor, por lo que, al estar implementando este tipo de actividades, el estudiante logra trabajar con diversas plataformas y paradigmas. Para el caso de la materia ligada a la electrónica (Principios Eléctricos y Aplicaciones Digitales), la competencia alcanzada implica resolver problemas del ámbito computacional con ayuda de sistemas digitales. Y en este proyecto se ha logrado cumplir con esta competencia al involucrar la tecnología *bluetooth* y diversos microcontroladores, auxiliándose de aplicaciones que se generan en la materia de Programación de Dispositivos Móviles, además de aplicar los estándares y protocolos de comunicación vistos en la materia de Fundamentos de Telecomunicaciones. Todo esto se documentó en una guía cuyo propósito es identificar los aspectos teórico-técnicos de las áreas involucradas (electrónica, programación, redes) en los primeros semestres del estudiante. De esta manera, se adquieren las bases para emprender nuevos proyectos y prototipos con mayor impacto en la carrera de Ingeniería en Sistemas Computacionales (ISC) del Instituto Tecnológico Superior de Irapuato (ITESI).

El que un estudiante logre analizar, integrar y aplicar diversas áreas de su carrera puede reforzar sus habilidades adquiriendo nuevos conocimientos y haciendo uso de las competencias señaladas en la tabla 1.

Tabla 1. Materias y competencias ligadas al proyecto

| Materia | Competencia alcanzada |
|---|---|
| Fundamentos de Programación | Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno. |
| Programación Orientada a Objetos (POO) | Diseñar e implementar objetos de programación que permitan resolver situaciones reales y de ingeniería. |
| Principios Eléctricos y Aplicaciones Digitales | Comprende y aplica las herramientas básicas de análisis de los sistemas analógicos y digitales para resolver problemas del ámbito computacional. |
| Fundamentos de Telecomunicaciones | Analiza los componentes y la funcionalidad de diferentes sistemas de comunicación para evaluar las tecnologías utilizadas actualmente como parte de la solución de un proyecto de conectividad. |

Fuente: Elaboración propia con información obtenida en los temarios de la retícula actual de la carrera de ISC del Itesi

El objetivo de este proyecto fue lograr como producto una guía técnica en la cual los estudiantes puedan consultar la información, requisitos y requerimientos adecuados de las materias relacionadas para la generación de prototipos de conectividad inalámbrica local, y con ello resolver la problemática planteada. Asimismo, las ventajas y beneficios que los estudiantes han recibido a partir del uso de la guía técnica son la implementación de talleres o cursos de divulgación impartidos a estudiantes y profesores de nivel básico y medio superior, la generación de productos de calidad, prototipos controlados vía inalámbrica, como brazos robóticos automatizados, evasores de obstáculos, seguidores de línea y proyectos con impacto positivo en el medio ambiente. Por mencionar un ejemplo, en el año 2016 estudiantes de la carrera obtuvieron un tercer lugar a nivel posgrado en un concurso de innovación y desarrollo tecnológico de la Asociación Nacional de Universidades e Instituciones de Educación Superior (Anuies), también se obtuvo el primer lugar de la fase local del Evento Nacional Estudiantil de Innovación Tecnológica (ENEIT) celebrado en el Itesi, y finalmente el primer lugar en la categoría de desarrollo tecnológico en el Encuentro de Jóvenes Investigadores. Otro aspecto para resaltar es que fruto de lo anterior se han registrado tres productos en el Instituto Nacional de Derechos de Autor (Indautor).

Metodología

Para llevar a cabo la guía técnica se han analizado tres áreas en tres etapas fundamentales de la carrera de ISC llevadas a cabo según el orden cronológico e impartición en su retícula. A continuación, se describen concretamente las referencias teóricas/técnicas de cada una de ellas, a partir del hecho de que algunos conceptos o términos son vistos y expuestos por el profesor durante las primeras clases, por lo que solo se describe la información que el estudiante requiere para la integración de estas áreas.

Etapa 1: Electrónica Digital

- Materia: Principios Eléctricos y Aplicaciones Digitales.
- Semestre: 4.º.
- Temas involucrados:
 - Tema 1. Fundamentos de circuitos eléctricos.
 - Tema 4. Convertidores.
- Competencia alcanzada: Comprende y aplica las herramientas básicas de análisis de los sistemas analógicos y digitales para resolver problemas del ámbito computacional.

Aspectos técnicos

Dos temas son analizados en esta etapa: 1) Arduino como plataforma base en la que el estudiante logra impulsar sus proyectos de automatización, robótica o domótica, y 2) el módulo *bluetooth* con el que interactuarán con su dispositivo móvil.

Conocimiento de Arduino

El estudiante debe saber que Arduino es declarado como *hardware* y *software* libre. Comúnmente al *hardware* se le conoce como una placa con un microcontrolador programable y del cual existe una gran variedad de tipos (Uno, Mega, Nano, etc.) y al *software* con el que interactúa suele llamarse *Arduino IDE* en referencia a la interfaz o ambiente gráfico en el que se programa la placa.

¿Por qué Arduino?

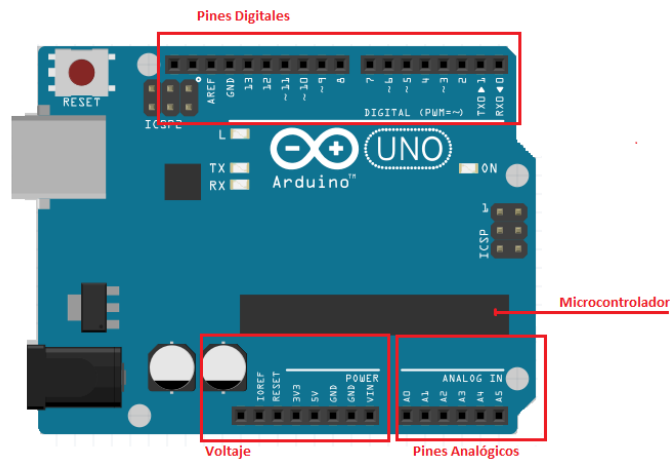
“Arduino simplifica el proceso de trabajar con electrónica digital, implica bajos costos de adquisición, funciona con diferentes sistemas operativos por lo que es multiplataforma y su software es de código abierto” (Vega, Santamaria y Rivas, 2014, p 26). Arduino Uno y Arduino Mega son los dos tipos de placas que se trabajan en esta guía al ser los más comunes, económicos, accesibles en el mercado local y debido a que son compatibles con un vasto universo de componentes electrónicos. Igualmente:

La plataforma permite la enseñanza de conceptos incluidos dentro del campo de la automática como el control de motores, el control de posición y velocidad, control de trayectorias, refuerzo de conocimientos sobre electrónica, manejo de comunicaciones inalámbricas, integración con sistemas Android de una manera aplicada y sin requerir conocimientos avanzados de programación (Soriano *et al.*, 2013).

¿Cómo funciona Arduino?

Arduino utiliza un microcontrolador programable cuyo circuito integrado posee las tres características básicas de un procesador común de cualquier computadora: una unidad central de procesamiento, la cual es la encargada de ejecutar el bloque de instrucciones o programa que se le “cargue”; una memoria limitada pero suficiente para almacenar un bloque de instrucciones para realizar infinitamente la misma operación hasta desgastar el suministro eléctrico o voltaje, y un medio de comunicación con los periféricos de entrada y salida cuyo propósito es interactuar con sensores y actuadores.

Figura 1. Segmentación de pines en Arduino UNO



Fuente: Elaboración propia

El microcontrolador del Arduino (véase figura 2) es el ATMEGA328P-PU de 8 bits, el cual posee una pequeña memoria RAM, un microprocesador, un convertidor analógico digital, un oscilador para la transmisión de datos, entradas digitales, entradas analógicas y buses de comunicación. Algunas de sus características principales son las siguientes: 23 pines de entrada/salida, memoria flash 32 kB programable, SRAM con capacidad para 2 kB y una memoria EEPROM de 1 kB.

Figura 2. Microcontrolador ATMEGA328P-PU



Fuente: <https://www.microchip.com/wwwproducts/en/ATmega328P>

Señales de E/S y compatibilidad de componentes

“Las señales representan información, a menudo las señales no llevan directamente la información necesaria y es posible que no estén libres de perturbaciones, el procesamiento de señales forma la base para resaltar, extraer, almacenar o transmitir información útil” (Ambardar,

2002). Por tal motivo, el estudiante debe comprender cuáles de los componentes de electrónica que está utilizando son capaces de enviar o recibir este tipo de señales digitales o analógicas y realizar posteriormente algún tipo de procesamiento con los datos extraídos.

Existe una lista de materiales y componentes que deben ser identificados, tal es el caso de los sensores, los cuales son todos los componentes y dispositivos capaces de detectar magnitudes físicas para convertirlas en magnitudes eléctricas medibles; en cambio, un actuador es todo aquel componente y dispositivo capaz de transformar algún tipo de energía en otro, según el objetivo de cada uno de ellos, comunicándose por medio de pines y un puerto USB con el que se almacena el código programado en Arduino IDE. A continuación, en la tabla 2, se enlistan los sensores y actuadores que se utilizan en la materia de Principios Eléctricos y se asigna el tipo de componente que es cada uno de ellos.

Tabla 2. Sensores y actuadores utilizados por estudiantes de ISC

| Sensores | Actuadores | Shields |
|--------------------------|-----------------------------|---------------------------|
| Sensor de golpe | Servomotor Sg90 | Motor drive Shield L2939d |
| Sensor de temperatura | Motor reductor de doble eje | Ethernet Shield W5100 |
| Sensor ultrasónico | Mini bomba de agua | LCD Keypad Shield 6x2 |
| Sensor óptico reflectivo | | Sensor Shield v5 |
| Sensor de humedad | | Proto Shield |
| Sensor RFID | | Shields puente H |

Fuente: Elaboración propia

Pines del Arduino

Toda placa Arduino cuenta con tres áreas para tres tipos de pines. Existen los pines de alimentación y suministro de voltaje que manejan niveles de 3 V y 5 V, así como de tierra (GND); un área con pines analógicos, los cuales van del pin A0 al pin A5 y se caracterizan por leer distintos niveles de tensión que van de 0 a 5 voltios, y cuenta con otra área para pines digitales que van del pin 1 al 13 y su característica principal es que solo puede leer dos niveles de señal: LOW para valores cercanos al 0 V y HIGH para valores cerca de los 5 V. Los pines Serial 0(RX) y 1(TX), por su parte, tienen el objetivo de enviar y recibir datos, mientras que el pin AREF es el encargado de

asignar un valor específico a los pines analógicos a través de la función “analogReference”. En la figura 1 puede observarse la ubicación de estas áreas en el Arduino UNO.

Shields

“Un Shield no es más que una placa de circuito impreso que se coloca en la parte superior de una placa Arduino y se conecta a ella mediante el acoplamiento de sus pines son necesidad de cable” (Torrente, 2013, p. 117). Para el desarrollo de algunas prácticas más complejas son necesarios algunos componentes extra compatibles con las placas de Arduino para permitir expandir sus capacidades y trabajar con los mismos niveles de voltaje que la placa. A este tipo de componentes se les conoce como *shields* y existe una gran variedad de ellos. En la figura 3 se muestran algunos ejemplos.

Figura 3. De base Arduino UNO, encima Shield Wifi, encima Shield Ethernet.



Fuente: <https://aprendiendoarduino.wordpress.com/2016/03/28/shields-arduino/>

Las *shields* también logran facilitar la codificación de los dispositivos al programador y reducir las líneas de código necesarias para hacerlos funcionar, además de poder incluir librerías accesibles y de fácil instalación.

Arduino IDE, uso general

Para el uso del entorno de desarrollo o *software* que interactúa con la placa programable, debe tomarse en cuenta los siguientes puntos:

1) Para compilar un programa primero es necesario guardarlo, posteriormente puede dar clic sobre el icono azul superior izquierdo con una palomita, en la parte inferior se notificará si existe algún problema con la sintaxis.

2) Si el proyecto que se está desarrollando requiere de algún componente o *shield* con librería, esta deberá descargarse como archivo rar, instalarse y declararse al inicio del programa.

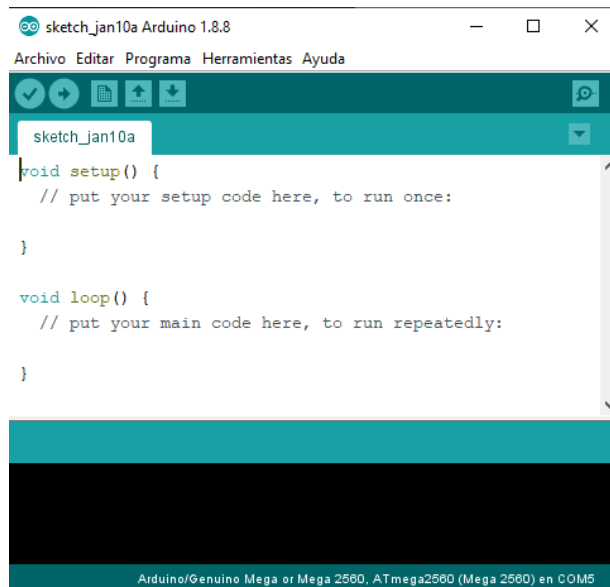
3) Deberán declararse las variables al principio del programa con los componentes y pines que estarán trabajando como entradas y salidas según su uso (INPUT/OUTPUT).

4) El código principal suele denominarse *sketch*, el cual se segmenta en dos partes: *void setup* y *void loop*. En el *void setup* se declaran y asignan valores iniciales a las variables declaradas, así como entradas y salidas en los puertos que estarán interactuando con los componentes del prototipo. En el *void loop*, se codifica la acción o acciones que los sensores o actuadores estarán realizando infinitamente a modo de ciclo, según las condiciones e indicaciones que se les programen.

5) Una vez compilado el programa sin errores, este quedará disponible para “subirse” o “cargarse” en la placa Arduino. Para ello deberá seleccionarse en el menú “Herramientas” la placa y puerto indicada para cada práctica.

En la figura 4 pueden observarse las características principales de la interfaz IDE de Arduino, como el nombre del programa, la ubicación de variables y el código principal que se estará ejecutando.

Figura 4. Pantalla principal de Arduino IDE

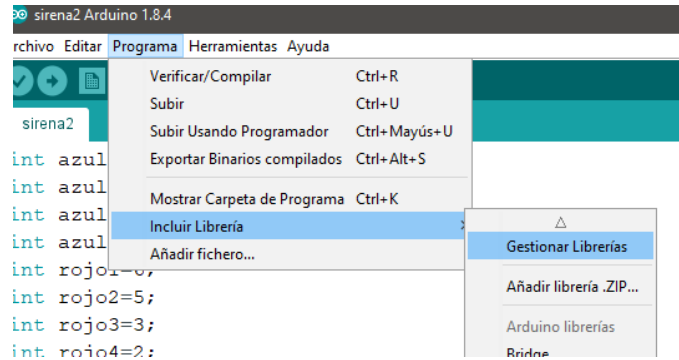


Fuente: Elaboración propia

Instalación de librerías

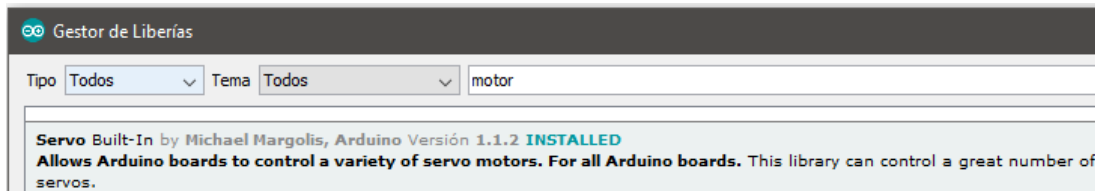
Una librería puede considerarse como un complemento o extensión que permite, para el caso de Arduino, que dos o más componentes electrónicos interactúen, o bien logren transmitir datos entre ellos. En el caso de Arduino, las librerías son generados por terceros y, al ser libres, cualquier usuario puede hacer modificaciones o adaptaciones. Arduino IDE cuenta por defecto con un grupo de librerías listas para ser utilizadas, basta con que el programador declare al inicio del *sketch* alguna de estas para hacer uso de ella. Al respecto, existen dos maneras para instalar una librería al Arduino IDE. La primera es mediante el propio asistente del *software* (ver figuras 5 y 6), siguiendo el menú “Programa”, “Incluir Librería” y después en “Gestionar Librerías”. A continuación aparecerá una ventana con una caja de texto para realizar búsquedas específicas con una descripción del componente o *shield* y una opción para descargarla.

Figura 5. Menú para acceder al gestor de librerías



Fuente: Elaboración propia

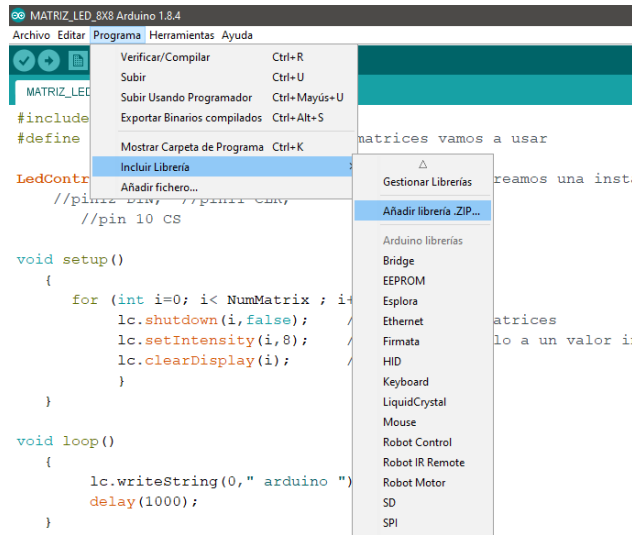
Figura 6. Ventana de búsqueda para librerías en Arduino IDE



Fuente: Elaboración propia

La segunda opción permite instalar manualmente la librería previamente descargada de alguna página oficial o del sitio web del fabricante, según el componente o *shield* a utilizar. Para ello se debe dar clic sobre el menú “Programa”, “Incluir Librería” y posteriormente en “Añadir Librería .ZIP” (véase figura 7). Se abrirá el cuadro de diálogo para buscar la librería en alguna ubicación y se da clic en aceptar. Arduino IDE notificará que la librería se añadió en la parte inferior de notificaciones de compilación y podrá ser visible en la lista de librerías del menú “Programa”.

Figura 7. Menú para agregar una librería en formato zip

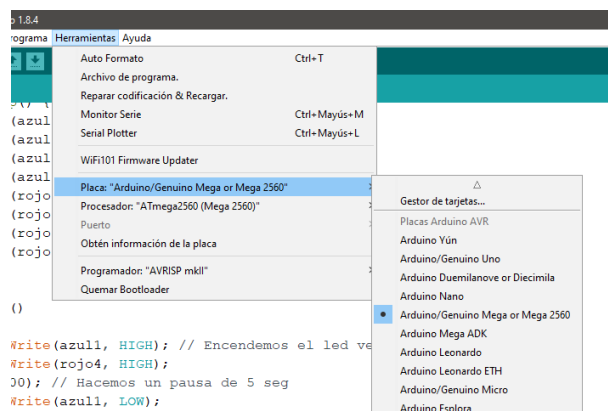


Fuente: Elaboración propia

Cargar programa en placa Arduino

Antes de cargar el *sketch* en alguna placa Arduino, deberá tener bien identificado el tipo de placa con la que se cuenta e indicar esta información al *software*. Puede hacerse en el menú “Herramientas”, después en “Placa” y se da clic sobre el nombre indicado (véase figura 8). De no hacerlo, el *sketch* no se cargará y se genera una notificación con el error.

Figura 8. Lista de placas Arduino



Fuente: Elaboración propia

Además, se debe indicar el puerto USB del equipo al que está conectado la placa, siguiendo el menú “Herramientas”, “Puerto” y se selecciona el puerto habilitado. Cabe recalcar que el equipo que detecta al Arduino le asigna un puerto de comunicación (COM), el cual se habilitará solo

cuando se detecte la placa y una vez que se descarguen por única vez los *drivers* necesarios para su funcionamiento.

Etapa 2: Telecomunicaciones y tecnología bluetooth

- Materia: Fundamentos de Telecomunicaciones.
- Semestre: 5.º.
- Temas involucrados:
 - Tema 1: Sistema de comunicación.
 - Tema 2: Medios de transmisión.
 - Tema 5: Modelos y dispositivos de comunicación.
- Competencia alcanzada: Analiza los componentes y la funcionalidad de diferentes sistemas de comunicación para evaluar las tecnologías utilizadas actualmente como parte de la solución a un proyecto de conectividad.

Bluetooth

Es uno de los medios de transmisión inalámbrica más comunes que existen al día de hoy. Muy utilizada en equipos de cómputo portátiles y dispositivos móviles, se le considera como “una tecnología de corto alcance diseñada para conexiones de periféricos a ordenador o dispositivos, está optimizada para los transceptores de radio de bajo consumo ideales para dispositivos personales” (Raya, Raya y Martínez, 2008, p. 101).

La tecnología *bluetooth* es un estándar que conjunta diversos protocolos de comunicación, aplicados en dispositivos de bajo costo que utilizan la transmisión Full-Duplex con una conectividad por radiofrecuencia para transmitir voz y datos en una red inalámbrica de área personal (WPAN), operando a una velocidad de transferencia de 720 kb/segundo en un rango óptimo de 10 metros y a una frecuencia que puede variar de los 2.4 a 2.48 GHz. Analizando esta tecnología desde el punto de vista electrónico, el módulo *bluetooth* HC-05 es actualmente uno de los más eficientes para la conexión inalámbrica local al poseer mayores características y prestaciones que el HC-06, ya que, al hacer uso de comandos AT (código preestablecido cuyo propósito es interactuar por medio de una terminal entre el ser humano y el componente), este

puede ser operado de manera esclavo o maestro: su principal diferencia cuando los estudiantes desarrollan proyectos y aplicaciones que requieren de este componente.

El módulo de *bluetooth* HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo maestro-esclavo, esto es, que además de recibir conexiones desde una computadora o tablet, también es capaz de generar conexiones hacia otros dispositivos *bluetooth*. Lo que permite conectar dos módulos de *bluetooth* y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos. A continuación, se describe la función para cada uno de los pines en el módulo HC-05 (dado que cuenta con dos pines más que el HC-06, se deduce que este último tendrá las características mencionadas en los cuatro pines con los que cuenta):

- 1) Pin State: con este pin se determina el estado actual en el que fue configurado/programado el módulo (esclavo/maestro).
- 2) Pin RX: este pin recibe los datos que son procesados en la placa Arduino por medio del sketch con el que fue programado.
- 3) Pin TX: este pin transmite datos leídos del módulo *bluetooth* y los regresa a la placa Arduino para esperar alguna nueva indicación del sketch.
- 4) Pin GND: pin de tierra, necesario para el funcionamiento del módulo.
- 5) Pin VCC: pin de suministro de voltaje que puede ir al pin de 3.3 V o 5 V, según las características de cada práctica.
- 6) Pin KEY: este pin es el que se requiere para indicar al módulo que estará trabajando como maestro por medio de comandos AT.

Estados del módulo HC-05

El módulo *bluetooth* puede presentarse en cuatro posibles estados: desconectado, conectado, modo AT1 y modo AT2. En la tabla 3 se describen algunas características.

Tabla 3. Estados del módulo *bluetooth* HC-05

| Estado | Características |
|--------------|---|
| Desconectado | <ul style="list-style-type: none"> - El módulo no establece comunicación con ningún otro dispositivo. - No puede interpretar ningún comando AT. - Su led, enciende y apaga rápidamente. |
| Conectado | <ul style="list-style-type: none"> - El módulo ha establecido comunicación con otro dispositivo. - Es capaz de enviar y recibir información por lo pines RX y TX. - Su led parpadea dos veces continuas y hace una espera. |
| Modo AT1 | <ul style="list-style-type: none"> - El módulo es capaz de recibir comandos AT a la velocidad con la que fue programado. - Para entrar a este modo, se requiere presionar el botón del módulo. |
| Modo AT2 | <ul style="list-style-type: none"> - El módulo puede recibir comandos AT a una velocidad preestablecida de 38 400 baudios. - Para entrar a este modo, se requiere dejar presionado el botón durante unos tres segundos. |

Fuente: Elaboración propia

Programa básico con Arduino y módulo HC-05

Una de las primeras prácticas que se puede realizar por medio del componente *bluetooth* es encender un led por medio de una aplicación. Para ello se requiere de una *protoboard*, un led, resistencia, fuente de alimentación y un dispositivo (teléfono celular) en el que se instalará la aplicación. En la figura 9 puede observarse este componente.

Figura 9. Módulo *bluetooth* maestro-esclavo



Fuente: <https://www.pccomponentes.com/modulo-hc-05-bluetooth-compatible-con-arduino>

A continuación, en la figura 10, se puede observar el código fuente para la creación de una práctica básica con transmisión *bluetooth*, en la que el objetivo es encender y apagar un led.

Figura 10. Programa para encender un led por conexión *bluetooth*

```

led_blue $
1 #include <SoftwareSerial.h>
2 SoftwareSerial BT(5,6); // TX= 5, RX=6
3 char opcion;
4
5
6 void setup()
7 {
8   pinMode(13, OUTPUT); //pin 13 al led
9   // Velocidad serial del Modulo BT
10  BT.begin(9600);
11 }
12
13
14 void loop() |
15 {
16   if (BT.available())
17   {
18     opcion=(BT.read());
19     if (opcion=='e')
20     {
21       digitalWrite(13, HIGH);
22       BT.println("LED on");
23     }
24     if (opcion=='a')
25     {
26       digitalWrite(13, LOW);
27     }
28   }
29 }

```

Fuente: Elaboración propia

- Línea 1: se incluye la librería requerida para el uso del módulo *bluetooth*.
- Línea 2: el pin 5 se conecta al puerto TX del módulo HC-05, el pin 6 se conecta al puerto RX del módulo HC-05.
- Línea 3: se declara la variable “opción”, la cual recibirá la instrucción de encender o apagar.
- Línea 8: se declara el pin 13 como puerto de conexión del Arduino al led.
- Línea 10: velocidad estándar del módulo *bluetooth*.
- Línea 16: si el dispositivo externo (teléfono celular) ya está conectado con el HC-05.
- Línea 19 y 24: condiciones para encender o apagar el led.

Finalmente, el módulo *bluetooth* HC-05 implica la aplicación de los estándares oficiales de comunicación y velocidad vistas en la materia de Fundamentos de Telecomunicaciones, como un componente de transmisión inalámbrica Full-Duplex.

Etapa 3: Programación

- Materia: Fundamentos de Programación (FP) / Programación Orientada a Objetos (POO).
- Semestre: 1.º / 2.º.
- Temas involucrados:
 - Tema 1: Diseño algorítmico (FP).
 - Tema 2: Introducción a la programación (FP).
 - Tema 3: Control de flujo (FP).
 - Tema 2: Clases y Objetos (POO).
 - Tema 4: Polimorfismo (POO).
- Competencia alcanzada: Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno (FP).
- Competencia alcanzada: Diseñar e implementar objetos de programación que permitan resolver situaciones reales y de ingeniería (POO).

Como ya se mencionó, una de las áreas más básicas e indispensables que debe dominar un ingeniero en sistemas computacionales es la programación. Para ello, el proyecto pretende que el estudiante domine y se adentre a nuevos desafíos en los que pueda adquirir nuevos conocimientos y paradigmas de programación. “Se pueden utilizar muchos lenguajes para programar una computadora, el más básico es el lenguaje máquina, aunque lo más frecuente es utilizar lenguajes de alto nivel” (Gottfried, 2005). En este sentido, los estudiantes de la carrera se involucran a la programación desde el primer semestre, se cubren las competencias de la materia en cuestión y se vinculan con algunas materias de semestres más avanzados.

Aspectos técnicos

A partir de la materia de Fundamentos de Programación, el estudiante adquiere los conocimientos necesarios para realizar programas en línea de comandos, o bien en una interfaz con algún lenguaje de alto nivel como .NET, por lo que el desarrollo de competencias ligadas a la programación se aplica directamente al estar configurando y programando diversos módulos y sensores de electrónica en el Arduino IDE, según las condiciones e instrucciones determinadas por el objetivo a cumplir. Lo que implica el análisis de algoritmos, diagramas de flujo y pseudocódigo que posteriormente deberá ser compilado y “subido” a la placa programable del Arduino.

En el caso de programación de *apps* con la tecnología *bluetooth*, el estudiante puede optar por una herramienta de libre acceso como el App Inventor, y con ello crear nuevos proyectos y vincular diversas áreas y conocimientos del ingeniero en sistemas computacionales.

En la actualidad existe diversos entornos que siguen invitando a “construir” productos derivados de lenguajes o entornos un tanto simplificados, pero no menos productivos que son capaces de producir importantes aportaciones, no sólo personales; sino el inicio de futuros desarrollos, algunos de estos lenguajes son: Scratch, Alice, Kodu, RoboMind, RoboLab y AppInventor (Díaz, 2015).

El estudiante de sistemas computacionales se ve obligado a realizar diagramas de flujo y otras actividades propias de la carrera para plantear una posible solución al problema al que se está enfrentando en términos informáticos, por lo cual es indispensable el desarrollo de uno o varios algoritmos. “Para entender bien un algoritmo, hay que realizarlo y ejecutarlo (...). El término *algoritmo* se utiliza para describir un método de resolución de un problema que es adecuado para su implementación como programa de computadora” (Sedgewick, 1995, p. 4).

App Inventor es un entorno de desarrollo creado por Google y el MIT que utiliza el sistema operativo Android como base y requiere de algoritmos bien sustentados para la creación de aplicaciones móviles. Entre sus principales ventajas se encuentran las listadas a continuación:

- El desarrollo de programación por bloques, lo que facilita en gran medida al estudiante que su análisis algorítmico o pseudocódigos puedan implementarse mediante bloques preprogramados y listos para cambiar parámetros o estructurados secuencialmente a modo de algoritmo para la ejecución de un determinado objetivo.

- Accesible y seguro: las aplicaciones pueden desarrollarse y almacenarse en línea, se requiere de una cuenta de correo en la nube y contraseña para acceder, el único requisito es contar con el servicio de internet.
- Instalador compatible: significa que, una vez terminada la aplicación, App Inventor da la opción de descargar la *app* con el formato .apk y listo para ser instalado en dispositivos Android, inclusive poder montar una imagen o fotografía como icono.
- Compatible con una gran variedad de sensores y componentes como wifi, *bluetooth*, GPS y mapas, entre otros.

En la figura 11 se muestra la pantalla principal del sitio web que aloja App Inventor, cuya liga es <http://appinventor.mit.edu>.

Figura 11. Ventana de acceso a la plataforma App Inventor

You are now logged out of MIT App Inventor

You are logged out. You can either close this window or use the button below to login to MIT App Inventor again.

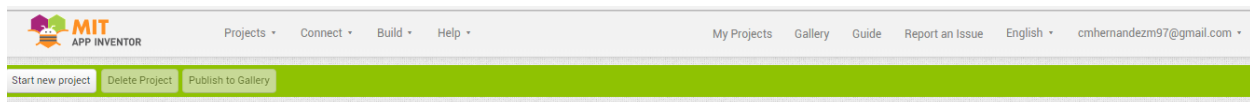
Login Again to MIT App Inventor



Fuente: <http://appinventor.mit.edu>

El uso de App Inventor requiere mínimos conocimientos de programación, por lo que para un estudiante del área es sumamente fácil de usar como un primer paso en creación de *apps* en el dispositivo, así como emular en el equipo de cómputo.

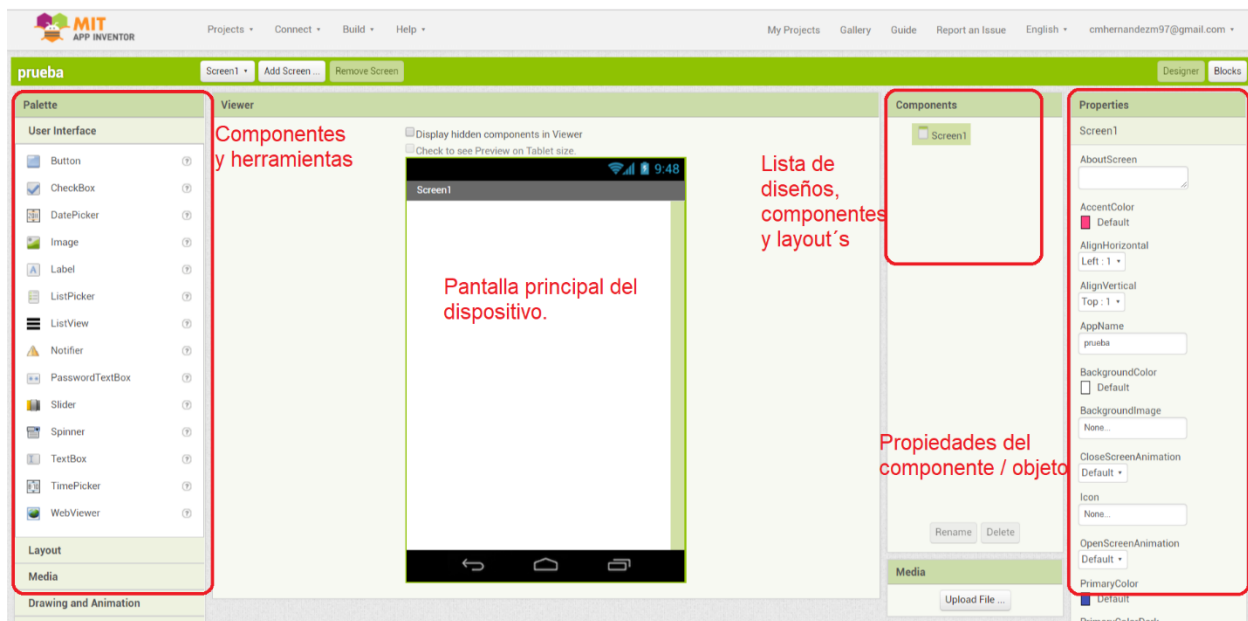
Figura 12. Menú principal de App Inventor



Fuente: <http://appinventor.mit.edu>

Una vez que el usuario ingresa con su datos de registro al sistema, puede inmediatamente crear su primera aplicación en línea dando clic al botón “Start new project” (véase figura 12). En la figura 13 se observan los módulos del nuevo proyecto con los que el estudiante podrá interactuar, los cuales son fáciles de identificar y utilizar.

Figura 13. Pantalla principal de un proyecto nuevo en App Inventor



Fuente: Elaboración propia

Componentes y herramientas

Al igual que un lenguaje de programación de alto nivel, App Inventor cuenta con una lista de objetos y componentes que se pueden agregar a la interfaz, tales como etiquetas, cajas de texto, botones, *layouts*, paneles, entre otros. Basta con seleccionar el componente y arrastrarlo a la pantalla principal.

Lista de componentes

En esta sección, aparecerán todos los componentes y objetos involucrados en la aplicación, a modo de cambiar su nombre, renombrarlo o eliminarlo. Es útil en el caso de seleccionar componentes sobreposicionados.

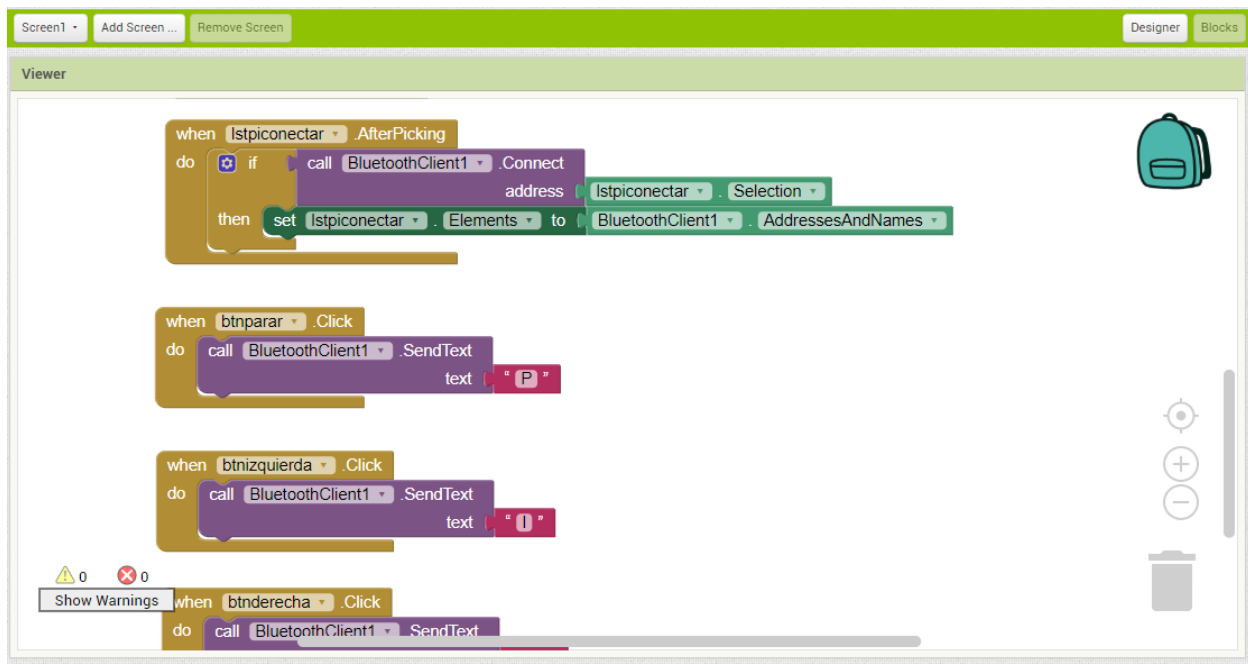
Propiedades del objeto

Como se puede anticipar, en esta sección el usuario puede modificar el efecto visual de los componentes utilizados, así como el tamaño, estilo o alineación del mismo.

Bloques de instrucciones

En la figura 14 se muestran los bloques de instrucciones que utiliza App Inventor, los cuales podrían pasar como “código fuente” de la aplicación al ser más un conjunto de instrucciones preprogramadas para la configuración y adaptación de parámetros (también puede considerarse pseudocódigo), sin embargo se requiere del análisis algorítmico y de la secuencia de instrucciones que deben ejecutarse en relación con el *sketch* del Arduino IDE. Es sumamente importante que el código fuente del *sketch* esté ligado a las opciones programadas en App Inventor.

Figura 14. Código fuente de la sección bloques en App Inventor



Fuente: Elaboración propia

Al finalizar la aplicación, el estudiante puede hacer una primera prueba del resultado obtenido en su dispositivo, pues es común encontrar algunas fallas técnicas en las dimensiones de algunos componentes. Para crear el instalador de la aplicación, se da clic sobre el botón “Build”

y enseguida elegir la opción de su preferencia, pues otra de las cualidades de esta plataforma es que permite descargar la aplicación con algún lector de sellos QR o por medio de un código, además del tradicional “guardar en el dispositivo”.

Finalmente, las tres áreas mencionadas en el proyecto culminan con el desarrollo de algún prototipo de transmisión inalámbrica, y se obtiene como resultado una guía técnica suficiente para que los estudiantes puedan emprender el desarrollo y emprendimiento de proyectos más complejos.

Conclusión

El desarrollo de actividades y prácticas relacionados con este proyecto fomentan en el estudiante el interés por emprender nuevos prototipos en los que pueda aplicar y aumentar sus conocimientos. En el área de las telecomunicaciones llegan a comprender los estándares y protocolos aplicados en la transmisión inalámbrica por *bluetooth*, inclusive programando estos módulos en sus dos versiones (maestro-esclavo), además de poder retransmitir y vincular hasta más de dos dispositivos bluetooth. En el área de la programación el estudiante tiene acceso a plataformas de *software* libre capaces de crear aplicaciones móviles de manera sencilla y eficaz, por lo que ya es común que estos estudiantes presenten algunos semestres de ventaja sobre sus compañeros, lo cual garantiza un mejor rendimiento y comprensión en los semestres dedicados a esta plataforma. En el área de la electrónica, los estudiantes pueden ver reflejados los términos y conceptos vistos en sus temarios al momento de realizar sus prácticas.

El resultado de obtener una guía técnica para estudiantes de los primeros semestres es el rápido comprendimiento de las tres áreas de la carrera involucradas, la integración con equipos de estudiantes más avanzados con los que tienen que colaborar y ejercer una práctica de enseñanza entre compañeros, el desarrollo de diversas habilidades y competencias implícitas en los temarios de la carrera, el liderazgo al ser un miembro activo de un grupo de trabajo y delegar algunas actividades, la iniciativa para generar nuevas ideas en proyectos integradores, la participación continua en concursos y eventos de la índole y la generación de nuevos conocimientos.

Referencias

- Ambardar, A. (2002). Procesamiento de señales analógicas y digitales (2.a ed.). México: Ediciones Paraninfo.
- Díaz, J. (2015). Open hardware y software, herramientas para el desarrollo de competencias educativas. *Revista Iberoamericana de Producción Académica y Gestión Educativa*, 2(3), 4-18.
- Gottfried, B. (2005). Programación en C (2.a ed.). Madrid, España: McGraw-Hill.
- Hewitt, N. (2007). El proyecto integrador: una estrategia pedagógica para lograr la integración y la socialización del conocimiento. *Psychologia. Avances de la disciplina*, 1(1), 236-240.
- Llanga, D. J. y Cano, D. A. (2014). Implementación del prototipo para el control de acceso docente mediante la interoperabilidad de WSN y RFID. (tesis de grado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. p.79 Recuperado de <http://dspace.esoch.edu.ec/bitstream/123456789/3789/1/98T00059.pdf>.
- Raya, J. L., Raya, L. y Martínez, M. A. (2008), *Redes locales, instalación y configuración básicas* (1.a ed.). México: Alfaomega.
- Sedgewick, R. (1995). *Algoritmos en c++* (1.a ed.). Estados Unidos: Prentice Hall.
- Soriano, A., Marín, L., Casalilla, J. I., Valera, A., Valles, M. y Albertos, P. (2013). Plataforma robótica de bajo coste y recursos limitados basada en Arduino y dispositivos móviles. Ponencia presentada en las XXXIV Jornadas de Automática. Terrassa, 4 al 6 de septiembre de 2013.
- Torrente, O. (2013). *Arduino. Curso práctico de formación* (1.a ed.). México: Alfaomega.
- Vega, A. M., Santamaria, F. y Rivas, E. (2014). Internet de los objetos empleando Arduino para la gestión eléctrica domiciliaria. *Red EAN*, (77), 22-41. Recuperado de http://www.scielo.org.co/scielo.php?script=sci_abstract&pid=S012081602014000200001

| Rol de Contribución | Definición (solo poner nombre del autor) |
|--|--|
| Conceptualización | César Manuel Hernández Mendoza, Fernando De los Reyes Quiroz, Juan Pablo Serrano Rubio (igual). |
| Metodología | César Manuel Hernández Mendoza, Fernando De los Reyes Quiroz, Juan Pablo Serrano Rubio (igual). |
| Software | César Manuel Hernández Mendoza |
| Validación | Fernando De los Reyes Quiroz |
| Análisis Formal | Juan Pablo Serrano Rubio |
| Investigación | Juan Pablo Serrano Rubio |
| Recursos | César Manuel Hernández Mendoza, Juan Pablo Serrano Rubio (igual) |
| Curación de datos | Juan Pablo Serrano Rubio, Fernando De los Reyes Quiroz (igual) |
| Escritura - Preparación del borrador original | César Manuel Hernández Mendoza |
| Escritura - Revisión y edición | César Manuel Hernández Mendoza, Fernando De los Reyes Quiroz (igual) |
| Visualización | César Manuel Hernández Mendoza |
| Supervisión | César Manuel Hernández Mendoza, Fernando De los Reyes Quiroz, Juan Pablo Serrano Rubio (igual). |
| Administración de Proyectos | Juan Pablo Serrano Rubio |
| Adquisición de fondos | César Manuel Hernández Mendoza, Juan Pablo Serrano Rubio (igual) |