



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA
INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE
SOFTWARE LIBRE ARDUINO**

Daniel Estuardo González López

Asesorado por el Inga. Ingrid Salomé Rodríguez García de Loukota

Guatemala, mayo de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA
INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE
SOFTWARE LIBRE ARDUINO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

DANIEL ESTUARDO GONZÁLEZ LÓPEZ

ASESORADO POR LA INGA. INGRID SALOMÉ RODRÍGUEZ GARCÍA DE
LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, MAYO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Christian Moisés De La Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Walter Jacobo Galicia García
EXAMINADOR	Ing. Carlos Alberto Navarro Fuentes
EXAMINADOR	Ing. Cristian Antonio Orellana López
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE SOFTWARE LIBRE ARDUINO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 21 de septiembre de 2020.

Daniel Estuardo González López

Guatemala 1 de marzo 2021

Ingeniero
Julio César Solares Peñate
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Apreciable Ingeniero Solares,

Me permito dar aprobación al trabajo de graduación titulado "**Diseño de un sistema domótico para el control de vivienda inteligente de clase media basado en la plataforma de software libre Arduino**", del señor **Daniel Estuardo González López**, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones de este.

Sin otro particular, me es grato saludarle.

Atentamente,



Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiado 5356

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

Guatemala, 11 de marzo de 2021

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE SOFTWARE LIBRE ARDUINO**, desarrollado por el estudiante **Daniel Estuardo González López**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

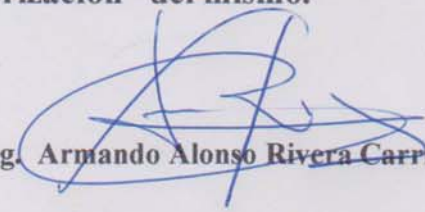
ID Y ENSEÑAD A TODOS

Ing. Julio César Solares Peñate
Coordinador de Electrónica



REF. EIME 88. 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; DANIEL ESTUARDO GONZÁLEZ LÓPEZ titulado; DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE SOFTWARE LIBRE ARDUINO, procede a la autorización del mismo.


Ing. Armando Alonso Rivera Carrillo



GUATEMALA, 29 DE ABRIL 2,021.

DTG. 220.2021.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE UN SISTEMA DOMÓTICO PARA EL CONTROL DE VIVIENDA INTELIGENTE DE CLASE MEDIA BASADO EN LA PLATAFORMA DE SOFTWARE LIBRE ARDUINO**, presentado por el estudiante universitario: **Daniel Estuardo González López**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana



Guatemala, mayo de 2021.

AACE/asga

ACTO QUE DEDICO A:

Mis padres

Miriam Audeth López Galindo de González y Maynor Roderico González Álvarez, como reconocimiento por ser la fuente de cada éxito obtenido a lo largo de mi vida, por ser mi guía y enseñarme las cosas más importantes de la vida.

Mis hermanos

Andrea, Gabriela y Rafael González, por ser grandes modelos a seguir, por el apoyo incondicional y el acompañamiento en cada paso, tanto profesional como personal.

Mis amigos

Por hacer memorable y divertida cada etapa de mi carrera universitaria

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por la formación, el conocimiento y las herramientas profesionales brindadas durante toda mi carrera.
Facultad de Ingeniería	Por ser una importante influencia en mi carrera.
Mis asesores de tesis	Ingeniera Ingrid de Loukota e ingeniero Carlos Pérez por toda la ayuda brindada.
Mi primo	Víctor, por estar presente en cada etapa de mi vida, por el apoyo y la motivación que siempre me ha dado.
Mis amigos de la Facultad	César Díaz y Abdel Chojolán por el compañerismo y los buenos momentos vividos durante todos estos años

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IXX
RESUMEN	XII
OBJETIVOS.....	XIII
INTRODUCCIÓN	XVV
1. HISTORIA DE ARDUINO Y DOMÓTICA	1
1.1. Estado actual de los diferentes sistemas de una vivienda.....	1
1.2. Arduino	3
1.2.1. Historia	4
1.2.2. Modelos de Arduino	6
1.3. Historia de la domótica	8
2. ACTUALIDAD DE ARDUINO	11
2.1. Software y hardware libre	11
2.2. Arduino	11
2.2.1. Hardware.....	11
2.2.1.1. Arduino UNO	12
2.2.1.2. Seeeduino	17
2.2.1.3. Sistema Grove.....	18
2.2.2. Software.....	19
2.3. Periféricos de Arduino	22
3. PROTEUS PROFESSIONAL 8.....	23

3.1.	Proteus professional.....	23
4.	ELEMENTOS DE UN SISTEMA DOMÓTICO	25
4.1.	Sensores	25
4.2.	Sensores analógicos	26
4.3.	Actuadores	28
4.3.1.	Comunicadores.....	29
4.3.2.	Arduino Shield Ethernet	29
4.3.3.	Módulo Wifi para Arduino.....	30
4.3.4.	Módulo XBee	30
4.3.5.	Módulo Bluetooth	30
4.3.6.	Módulos infrarrojos.....	31
4.3.7.	Puerto serie.....	31
4.4.	Servidor.....	32
5.	ARQUITECTURA DE SISTEMAS DOMÓTICOS	33
5.1.	Arquitectura de los sistemas domóticos	33
5.1.1.	Distribuida	33
5.1.2.	Centralizada.....	34
6.	DISEÑO DE UN SISTEMA DOMÓTICO	337
6.1.	Metodología.....	37
6.2.	Diseño de prototipo de sistema domótico.....	37
6.2.1.	Estructura.....	37
6.3.	Funcionamiento.....	40

CONCLUSIONES	45
RECOMENDACIONES	47
BIBLIOGRAFÍA.....	49
APÉNDICES	51

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Distintos sistemas de una casa independientes entre sí.....	2
2.	Unificación de todos los sistemas gracias a la domótica.....	3
3.	Industruino PROTO D21G	8
4.	Vista frontal de la placa Arduino UNO.....	13
5.	Elementos de la placa Arduino UNO.....	14
6.	Vista frontal de la placa Seeeduino	17
7.	IDE de Arduino.....	19
8.	Monitor serial de Arduino	21
9.	Proceso de creación de proyecto utilizando el software Proteus	24
10.	Curva característica de los termistores TTC3A.....	27
11.	Arduino como servidor web.....	32
12.	Arquitectura distribuida.....	34
13.	Arquitectura centralizada.....	35
14.	Estructura del sistema domótico centralizado utilizando Arduino.....	38
15.	Arduino Ethernet Shield acplado y Arduino UNO conectado a router	38
16.	Estructura completa del sistema domótico.....	39
17.	Simulación del sensor DHT11	40
18.	Comunicación serial mediante software Arduino	42
19.	Comunicación con el sistema utilizando protocolo HTTP.....	43

LISTA DE SÍMBOLOS

Símbolo	Significado
\$	Dólar estadounidense
kB	Kilobytes
MHz	Megahertz
mA	Miliamperios
V	Voltios

GLOSARIO

ARM	<i>Advanced RISC Machine.</i>
E/S	Entrada o salida.
IDE	Entorno de Desarrollo Integrado.
IDII	Interaction Desing Institute Ivrea.
PCB	<i>Printed Circuit Board</i> , placa de circuito impreso.
PIC	<i>Programmable Integrated Circuited</i> , circuito integrado programable.
PWM	Modulación por ancho de pulsos, siglas en inglés de <i>pulse-width modulation</i> .
RISC	<i>Reduced Instruction Set Computer.</i>
SCE	Sistema de Cableado Estructurado.
TIC	La tecnología informática y comunicaciones.
VCC	Voltaje en corriente continua.

Vin

Voltaje de entrada.

RESUMEN

La domótica hace referencia a todos los mecanismos que se han automatizado dentro de los sistemas domésticos, como la iluminación, cerraduras, climatización y otros. Además, da respuesta al cambio social, dando lugar a formar parte de las nuevas tendencias para facilitar los hogares.

Si bien es cierto que la tecnología está avanzando rápido, y sumado a ello el hecho que el ser humano busca economizar procesos que le permitan la comodidad y rapidez de la situación. Por lo que resulta una opción interesante, incluir tecnología en procesos del hogar y cotidianos para satisfacer necesidades diversas que las personas y familias van presentando.

El diseño de un sistema domótico para el control de vivienda inteligente de clase media basado en la plataforma de software libre Arduino consta de varias partes para fundamentarse. La primera consiste en presentar los antecedentes generales y sustentar el objetivo a cumplir; de igual manera, se presenta la historia de lo que es el Arduino y sus modelos, así como la historia de la domótica.

En los primeros capítulos de este trabajo de investigación, se definirán términos básicos como software y hardware libre. Así mismo, se explicará lo que es Arduino, Proteus Professional, la arquitectura de los sistemas domóticos y la manera en que se distribuyen. En cuanto al marco metodológico se mostrará el diseño del prototipo del sistema que se está proponiendo y, se mostrará la estructura y el funcionamiento de este. Por último, se exponen las conclusiones y recomendación, como la bibliografía utilizada para fundamentar el diseño del prototipo.

OBJETIVOS

General

Diseñar un sistema domótico para el control de vivienda inteligente basado en la plataforma de software libre Arduino.

Específicos

1. Exponer los conocimientos requeridos para comprender los sistemas domóticos.
2. Detallar el procedimiento para diseñar un sistema domótico simple, estable y de bajo presupuesto.
3. Determinar los dispositivos adaptables a las placas Arduino que conformarán el sistema.

INTRODUCCIÓN

En la actualidad, la tecnología forma gran parte de la vida del ser humano y cada vez existe una mayor dependencia de la sociedad hacia todo tipo de dispositivos móviles como teléfonos inteligentes, tabletas, laptops, entre otros. De esta necesidad nace la idea de la domótica, la idea de simplificar procesos cotidianos a partir de la inclusión de la tecnología en los hogares.

En palabras sencillas, la domótica se define como el conjunto de servicios proporcionados por sistemas tecnológicos para satisfacer las necesidades básicas de seguridad, comunicación, gestión energética y confort.

Al usar este término se hace referencia a los mecanismos de automatización y control de los sistemas domésticos comunes como el de la iluminación, climatización, cortinas, puertas y ventanas, cerraduras, dispositivos de riego, electrodomésticos, suministro de agua, suministro de gas, suministro de electricidad, entre otros.

La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles. Pero la evolución, además de medirse por referencias tecnológicas, también se mide por consumidores y sectores del mundo y, si bien, América Latina aún no está arraigada a las propiedades que ofrece la domótica, es inevitable que muy pronto así será.

Actualmente, los sistemas domóticos tienen un precio de instalación muy elevado. Estos suelen utilizar buses de transmisión de información que posibilitan una domótica robusta como son el EIB, X10, CEBus, LonWorks/LongTalk y ZigBee. Una alternativa más barata y casera consiste en la utilización de placas Arduino.

En este trabajo se realizará un diseño básico de un sistema domótico haciendo uso del microcontrolador Arduino debido a que es de software libre, con el fin de que las licencias y las librerías para el uso de todos los sensores, actuadores, módulos y demás complementos que son de uso libre y gratuito se logre crear un sistema estable con un presupuesto muy inferior al de las viviendas de alta categoría.

1. HISTORIA DE ARDUINO Y DOMÓTICA

1.1. Estado actual de los diferentes sistemas de una vivienda

Se tomará de base una casa que no posee un sistema domótico, todas las redes son independientes entre sí. Es decir, la red de telefonía no tiene relación con la red de iluminación de la casa o el sistema de multimedia y entretenimiento no está conectado con el sistema de calefacción. La independencia de estas diferentes redes hace que la accesibilidad, la gestión y el control de todos estos elementos se dificulten y se requiera de mucho equipo diferente e incompatible entre sí para su manejo.

Debido a que sin un sistema domótico no es posible monitorear ni gestionar inteligentemente la iluminación, la climatización, el riego, el agua caliente sanitaria, los electrodomésticos, entre otros, resulta difícil el tema de ahorro energético y, por tanto, ahorro económico. Sin un sistema que permita el control eficiente de todos estos recursos, se tiende al desperdicio de los recursos.

Los sistemas de seguridad en una casa son de suma importancia, es uno de los espacios más sensibles de todos donde los que interactúa el ser humano. En el hogar se fraguan los vínculos más íntimos y personales.

Cuando se habla de seguridad en el hogar se piensa en sistemas contra intrusión, en mecanismos que prevengan de sufrir actos vandálicos, robos y otros percances procedentes principalmente del exterior, algo comprensible pero que responde a una visión incompleta y sesgada si no se amplía la perspectiva. Es decir, al tratar la seguridad en los hogares y viviendas se debe tener en cuenta

estos asuntos, por supuesto, pero además también se debe considerar las amenazas que provienen del interior y que pueden poner en riesgo tanto a personas como a infraestructuras, bienes y componentes de cualquier vivienda.

Sin un sistema domótico instalado en el hogar, no se puede tener control ni saber qué está sucediendo en la casa en cualquier momento, no se cuenta con un cierre automático de todas las puertas y ventanas por lo que se puede olvidar cerrar alguna de estas dejándola expuesta. Tampoco se cuenta con un sistema de cámaras y vigilancia, alarmas personales o alarmas técnicas que permiten detectar incendios, fugas de gas, fallos del suministro eléctrico entre otros incidentes.

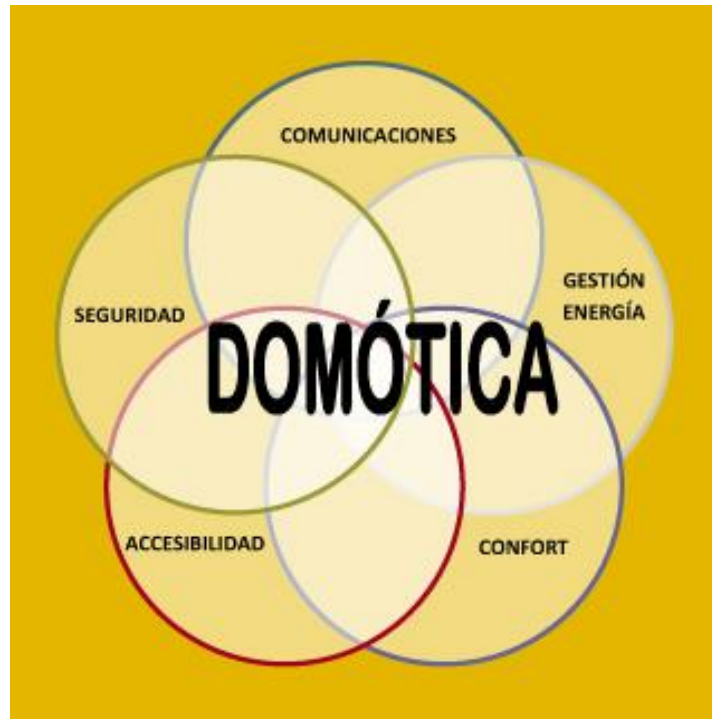
La falta de un sistema domótico instalado en casa, afecta a otros aspectos como la comunicación, el confort, las compras e incluso la salud al no tener asistentes virtuales sanitarios, alarmas de salud, medicina monitorizada.

Figura 1. **Distintos sistemas de una casa independientes entre sí**



Fuente: elaboración propia, empleando Paint.

Figura 2. **Unificación de todos los sistemas gracias a la domótica**



Fuente: ELKAN, Michael. *¿Qué aporta la domótica?*

<https://www.monografias.com/trabajos107/domotica-mejor-calidad-vida/image001.jpg>. Consulta:
3 de agosto de 2020.

1.2. **Arduino**

Se conoce como Arduino a la plataforma para la creación de electrónica de código abierto. Permite el establecimiento de distintos tipos de microordenadores para diferentes usos. Se caracteriza por ser libre, flexible y sencilla para quienes la utilizan. Parte de la libertad que brinda, es que los usuarios pueden modificar como crear las placas, partiendo de la misma base y dándoles la utilidad que deseen. Esta placa se basa en un microcontrolador, es un circuito para guardar instrucciones. Asimismo, cuenta con distintos puertos de entrada y de salida para que pueda procesar y recibir los datos que se le envían, tanto del entorno de programación como a otros dispositivos.

1.2.1. Historia

De forma estricta, el proyecto "Arduino" se inició en el año 2005 como un proyecto enfocado a estudiantes en el IDII. En aquellos años, los estudiantes usaban el microcontrolador BASIC Stamp, cuyo costo era de \$100, un costo considerablemente alto para un estudiante promedio. Antes del año 2005, específicamente durante el año 2003, Hernando Barragán había creado la plataforma de desarrollo Wiring como resultado de su proyecto de tesis en la maestría en el IDII, bajo la supervisión de Massimo Banzi y Casey Reas, quienes eran conocidos por haber trabajado en el lenguaje Processing y daban clases en el IDII. El objetivo del proyecto era crear herramientas simples y de bajo costo para la creación de proyectos digitales por parte de personas sin altos conocimientos técnicos o sin un perfil de ingeniería. El proyecto Wiring era una placa de desarrollo de hardware que constaba de una PCB con un microcontrolador ATmega168, un IDE basado en funciones de procesamiento y una biblioteca de funciones para programar fácilmente el microcontrolador. Regresando al año 2005, Massimo Banzi junto con David Mellis, otro estudiante del IDII, y David Cuartielles, agregaron soporte a Wiring para el microcontrolador ATmega8, más económico que el inicial, Atmega168. Pero en lugar de continuar el desarrollo en Wiring, se separaron del proyecto y lo renombraron Arduino.

El nombre Arduino viene de un bar en Ivrea, Italia; en donde algunos de los fundadores del proyecto Arduino solían reunirse. El bar tiene el nombre de "Bar di Re Arduino", y fue nombrado en honor a Arduino de Ivrea, quien fue el margrave de la Marcha de Ivrea y Rey de Italia desde el año 1002 hasta el año 1014.

El equipo inicial de Arduino estaba conformado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis. Hernando Barragán no fue invitado a participar. Posteriormente, el proyecto Wiring siguió adelante con

placas menos densas y costosas que se distribuyeron a través de la comunidad de código abierto.

Adafruit Industries, un proveedor de componentes electrónicos y fabricante de placas de circuito impreso, entre ellas placas Arduino, ubicado en la ciudad de Nueva York, estimó a mediados del año 2011 que se habían producido comercialmente más de 300 000 placas Arduino oficiales. En el año 2013, estimó que se encontraban en manos de usuarios 700,000 placas Arduino oficiales.

En octubre del año 2016, Federico Musto, actualmente ex CEO de Arduino, adquirió el 50 % de la compañía tras haber adquirido las acciones de uno de los miembros fundadores del equipo. En abril del año 2017, la revista Wired informó que Musto había fabricado su propio expediente académico, habiéndolo publicado en el sitio web de Arduino, cuenta personal de LinkedIn, e incluso en documentos comerciales oficiales italianos. Musto afirmaba tener un PhD en ciencias de la computación por el Instituto Tecnológico de Massachusetts, y un MBA de la Universidad de Nueva York. La revista Wired reportó que ninguna de las universidades donde él afirmaba haber estudiado tenía registro alguno de la asistencia de Musto. Afirmó más tarde, en una entrevista a Wired, que realmente nunca había obtenido los grados académicos.

En el año 2017, Massimo Banzi anunció la creación de la Fundación Arduino, declarando que sería un nuevo comienzo para Arduino. Dicha fundación, según palabras del mismo Banzi, permitirá defender los valores fundamentales de la Comunidad Arduino dentro del ecosistema de código abierto y hacer que nuestro compromiso, haciendo referencia a la empresa Arduino, con el código abierto sea más sólido que nunca. Sin embargo, ha existido cierta incertidumbre del desarrollo actual de dicha iniciativa.

La controversia en torno a Federico Musto continuó en julio del año 2017, según los informes, por haber retirado licencias de código abierto, esquemas y códigos del sitio web de Arduino, lo que provocó protesta por parte de la comunidad de creadores.

En octubre del año 2017, Arduino anunció su asociación con la multinacional ARM Holdings. El anuncio decía, en parte, que ARM reconoce la independencia como un valor central de Arduino sin ningún acuerdo de uso exclusivo con la arquitectura ARM. Arduino tiene la intención de seguir trabajando con todos los proveedores y arquitecturas de tecnología.

Para la producción en serie de la primera versión se tomó en cuenta que el coste no fuera mayor de 30 euros, que fuera ensamblado en una placa de color azul, debía ser Plug and Play y que trabajara con todas las plataformas informáticas tales como MacOSX, Windows y GNU/Linux. Las primeras 300 unidades de Arduino se las dieron a los alumnos del IDII, con el fin de que las probaran y empezaran a diseñar sus primeros prototipos.

Tras el enorme éxito del proyecto, aparecieron clones y compatibles y sistemas similares, basados en otros microcontroladores, como Pingüino, basado el PIC 18F. Incluso el propio fabricante de los PIC, microchip, lanzó chipKIT, con PIC32 compatible con el hardware y el software de Arduino.

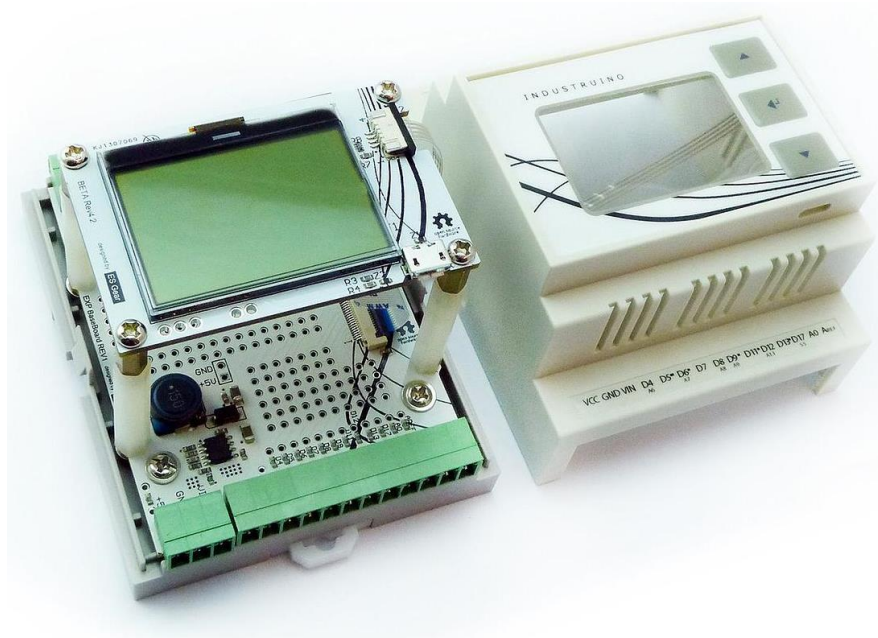
1.2.2. Modelos de Arduino

- Arduino Uno
- Arduino Leonardo
- Arduino Due
- Arduino Yún
- Arduino Tre, en desarrollo

- Arduino Zero, en venta en la tienda de EEUU
- Arduino Micro
- Arduino Esplora
- Arduino Mega ADK
- Arduino Ethernet
- Arduino Mega 2560
- Arduino Robot
- Arduino Mini
- Arduino Nano
- LilyPad Arduino Simple
- LilyPad Arduino SimpleSnap
- LilyPad Arduino
- LilyPad Arduino USB
- Arduino Pro Mini
- Arduino Fio
- Arduino Pro

Basado en Arduino Leonardo, Atmega 32U4, nace la plataforma INDUSTRIINO, que pretende llevar a la facilidad de programación de Arduino, al ambiente industrial y del control más profesional.

Figura 3. **Industruino PROTO D21G**



Fuente: Industruino. *Industruino PROTO D21G*.

<https://industruino.com/shop/product/industruino-proto-d21g-8>. Consulta: 25 de noviembre de 2020.

1.3. **Historia de la domótica**

El primer indicio de domótica, como se conoce en la actualidad, se dio a inicios de la década de los 70, cuando aparecieron los primeros dispositivos de automatización en edificios de países de vanguardia como Estados Unidos, Alemania y Japón, pero como una prueba piloto. Fue en la década del 80 cuando realmente los sistemas integrados se utilizaron a nivel comercial, para luego desarrollarse en el aspecto doméstico de las casas urbanas. Allí es cuando la domótica consigue integrar dos sistemas, el eléctrico y el electrónico, en pos de la comunicación integral de los dispositivos del hogar.

Acorde a los cambios, el auge de la informática hogareña permite incorporar en los edificios el SCE que facilita la conexión de terminales y redes. Así, estos edificios reciben el nombre de inteligentes, por su automatismo al servicio del propietario. El gran crecimiento de los rascacielos de oficinas comerciales fue de gran ayuda al movimiento de la domótica ya que ésta permitía lograr una eficiencia inédita para el servicio, desempeño y control de dichas oficinas.

El primer programa que utilizó la domótica fue el X-10. Creado en Estados Unidos en 1984, permite lograr eficiencia y bajo consumo de energía en los sistemas de control de edificios inteligentes. Estas instalaciones regían bajo el sistema X-10, protocolo de comunicaciones que opera a través del accionar de un control remoto. Dicho protocolo fue desarrollado en Escocia en 1976 por Pico Electronics y sigue siendo la tecnología más utilizada dentro de la domótica. Al transmitir datos por líneas de baja tensión, la relación costo-beneficio sigue siendo la mejor opción en el rubro.

Implantada desde hace más de treinta años, la domótica ha progresado a gran escala desde que se desarrollaron las redes informáticas de comunicación, ya sea por sistema cableado o Wi-Fi. El avance tecnológico vino a suplir las falencias de los comienzos, ya que permite integrar de manera eficiente todos los dispositivos tecnológicos de una casa. Con el fin de la década de los 80, las tecnológicas destinadas a fines comerciales, comienzan a llegar a los hogares.

Irrumpe la era de la TIC, que posibilita entender una forma más realista de comprender la instalación domótica en casa. En la actualidad, hay una oferta consolidada en torno a los servicios de domótica. Nuevos protocolos permiten un desarrollo que en un principio era impensado. Sistemas de desarrollo 2,0 como el ZigBee permiten conformar un protocolo inalámbrico de comunicación domótica. Al requerir una baja tasa de envío de datos, ahora es uno de los

protocolos más requeridos para las casas inteligentes, ya sea en sensores de movimiento, detectores de humo y otras funciones de seguridad en el hogar.

Con la domótica aplicada a la automatización hogareña se mejora en seguridad, confort y ahorro energético, aspectos muy observados por los poseedores de estos sistemas. La llegada de Internet a gran velocidad provocó un giro favorable para su desarrollo.

En los últimos años, el mercado de ofertas se ha extendido, permitiendo encontrar diversas variantes de equipos domésticos de integración domótica, como es el caso del EIB, que es un conductor eléctrico que ayuda a optimizar los distintos sistemas de seguridad y funcionalidad que componen una casa.

2. ACTUALIDAD DE ARDUINO

2.1. Software y hardware libre

El software libre es definido como:

El software que respeta la libertad de los usuarios y la comunidad. En grandes líneas, significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el Software Libre es una cuestión de libertad, no de precio.¹

El hardware libre no es tan sencillo de definir como el software libre, sin embargo, la definición actualmente aceptada es la de “Hardware libre es aquel hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita.”²

2.2. Arduino

Arduino es una plataforma de código abierto en la que se pueden realizar cualquier tipo de creaciones electrónicas. Dicha plataforma está basada en software y hardware libre y sencillo de utilizar para quien desee hacerlo.

2.2.1. Hardware

Arduino es una plataforma de hardware libre, es decir, tanto su diseño como su distribución pueden utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin la necesidad de adquirir licencia. Por esta razón, existen

¹ El Diario de Turing. *¿Qué es el software libre?* p.1.

² El Diario de Turing. *¿Qué es el hardware libre?* p.1.

diferentes tipos de placas, ya sean oficiales, que son creadas por la comunidad Arduino y las no oficiales creadas por terceros, pero con características similares que cumplen con los requerimientos dados por Arduino.

Arduino es una plataforma de hardware abierto que facilita la programación de un microcontrolador. Los microcontroladores usan los sensores para percibir el mundo físico y los actuadores para interactuar con él. Las placas Arduino, como todos los microcontroladores, leen sobre los sensores y escriben sobre los actuadores.

2.2.1.1. Arduino UNO

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador. A continuación, se resumen sus características técnicas:

- Microcontrolador: ATmega 328
- Voltaje: 5 V
- Voltaje de entrada recomendado: 7-12 V
- Voltajes límites de entrada: 6-20 V
- Pines Digitales E/S: 14, de los cuales 6 son salida PWM
- Pines de entrada analógica: 6
- Corriente continua para pines E/S: 40 mA
- Corriente continua para pines de 3,3 V: 50 mA

- Memoria flash: 32 kB, de los cuales 0.5 kB son utilizados para el arranque
- SRAM: 2 kB
- EEPROM: 1 kB
- Velocidad de reloj: 16 MHz

Figura 4. **Vista frontal de la placa Arduino UNO**

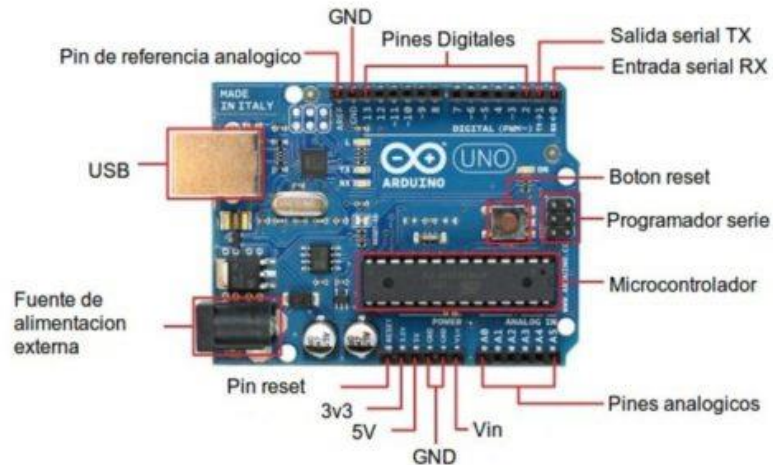


Fuente: Prometec. *Placa Arduino UNO*. <https://prometec.mx/producto/arduino-uno/>.

Consulta: 3 de agosto de 2020.

En la figura 5 se muestra una placa Arduino UNO señalando la posición y el nombre de sus elementos principales y una breve descripción de cada uno.

Figura 5. Elementos de la placa Arduino UNO



Fuente: Meca Fénix, Ingeniería. *¿Qué es Arduino?*

<https://www.ingmecafenix.com/electronica/arduino/>. Consulta: 4 de agosto de 2020.

- Referencia para pines analógicos, AREF: Tensión de referencia para entradas analógicas. Se utiliza con la función `analogReference()`.
- Pines de tierra, GND: Masa del circuito para pines, es decir es la tensión de referencia de 0 V.
- Pines digitales de entrada y salida: En los pines se conectará la patilla de dato del sensor/actuador. Desde estos se podrá leer la información del sensor o activar el actuador.

Hay 14 pines digitales que pueden utilizarse como entrada o salida con las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Operan a 5 voltios. Cada pin proporciona o recibe como máximo 40 mA y disponen de una resistencia pull-up (desconectada por defecto) de 20-50 kOhmios. Ciertos pines son reservados para determinados usos:

- Serie: 0(RX) y 1(TX). Utilizados para recibir (RX) y transmitir (TX) datos serie. Están directamente conectados a los pines serie del microcontrolador. Utilizando estos pines se podrá conectar con otras placas.
 - Interrupciones externas: 2 y 3. Estos pines pueden ser configurados para activar interrupciones.
 - PWM: 3, 5, 6, 9, 10 y 11. Proporcionan una salida de 8 bits en modo PWM.
 - SPI: 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI.
 - Led: 13. Este pin está conectado con un led de la placa. Cuando se le asigne un valor HIGH se encenderá, en cambio, si se deja en LOW estará apagado.
- Conector USB: Existen varios tipos de conectores USB, en concreto esta placa utiliza el tipo B hembra. Con lo cual se necesitará un cable tipo B macho – tipo A macho (aunque se pueden utilizar otros este es el más extendido) que deberá conectarse a un conector tipo A hembra (por ejemplo, a un ordenador o al cargador de un móvil). La placa se puede alimentar con la tensión de 5 V que le proporciona el bus serie USB. Cuando se cargue un programa a la placa desde el software de Arduino se inyectará el código del ordenador por este bus.
 - Botón Reset: Utilizando este botón se podrá reiniciar la ejecución del código del microcontrolador.
 - ICSP, In Circuit Serial Programming: Es un conector utilizado en los dispositivos PIC para programarlos sin necesidad de tener que retirar el chip del circuito del que forma parte.

- Microcontrolador ATmega328: El microcontrolador es el elemento más importante de la placa. Es donde se instalará y ejecutará el código que se haya diseñado. Ha sido creado por la compañía Atmel, tiene un voltaje operativo de 5 V, aunque se recomienda como entrada de 7-12 V con un límite de 20 V. Contiene 14 pines digitales de entrada y salida, 6 pines analógicos que están conectados directamente a los pines de la placa Arduino comentados anteriormente. Dispone de 32 kB de memoria flash, de los cuales 512 bytes son utilizados por el bootloader.

En la memoria flash se instalará el programa a ejecutar. El bootloader será el encargado de preparar el microcontrolador para que pueda ejecutar el programa. También tiene una memoria EEPROM de 1 kB que puede ser leída o escrita con la librería EEPROM. En la parte de procesamiento dispone de un reloj de 16 Mhz y 2 kB de memoria RAM.

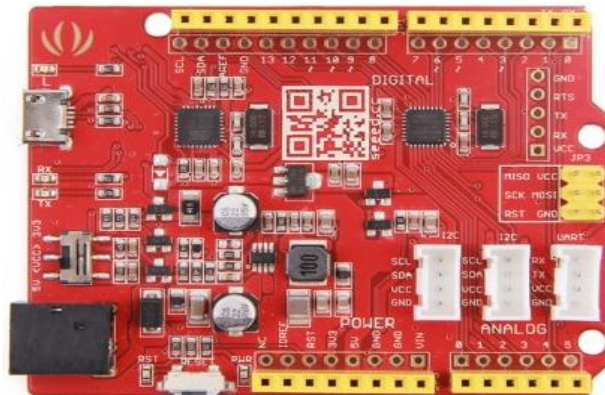
- Fuente de alimentación externa: la placa puede ser alimentada también mediante corriente continua suministrada por el conector jack de 3,5 mm que podrá recibir entre 7 y 12 V.
- Pin de Reset: Se puede imitar el funcionamiento del botón Reset suministrando un valor LOW(0V) para reiniciar el microcontrolador.
- Pin de 3,3 V: Desde aquí se podrá suministrar 3,3 V a los dispositivos que lo necesiten con una corriente máxima de 50 mA. Es generada gracias al chip FTDI integrado en la placa.
- Pin de 5 V: Este pin saca una tensión de 5 V del regulador de la placa. El regulador es necesario puesto que puede ser alimentada con distintos voltajes.

- Pin de Vin: Es el voltaje de entrada cuando se usa una fuente de alimentación externa (no tiene en cuenta la conexión USB). Se puede proporcionar voltaje a la placa a través de este pin, o en caso de que se esté utilizando una fuente de alimentación externa tomar el valor que está siendo suministrado.
- Pines analógicos: Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales pero su uso requiere de una lógica mayor.

2.2.1.2. Seeeduino

Seeeduino V 4,2 es una placa compatible con Arduino que se basa en ATmega328P MCU. En el mundo hay muchas placas compatibles con Arduino, y generalmente se cree que Seeeduino V 4,2 es uno de los mejores Arduinos. Seeeduino V 4,2 es mucho más estable, fácil de usar e incluso atractivo.

Figura 6. Vista frontal de la placa Seeeduino



Fuente: Antratek. *SeeeduinoV4.2*. <https://www.antratek.com/seeeduino>. Consulta: 10 de agosto de 2020.

Seeeduino V 4,2 está basado en el gestor de arranque Arduino UNO y con un ATMEGA16U2 como convertidor de UART a USB, lo que significa que la placa puede funcionar básicamente como un chip FTDI. Se puede programar la placa a través de un cable micro-USB. También se puede alimentar la placa a través de una entrada DC Jack, de 7 a 15 V es aceptable. Hay un interruptor para elegir el voltaje de suministro del sistema, 3,3 V o 5 V, que es muy útil si se desea configurar el sistema a 3,3 V para ahorrar energía. Finalmente, la interfaz de Grove integrada en tres puede hacer que la placa se conecte fácilmente a Groves.

- Características técnicas:
 - Microcontrolador ATmega328P
 - Arduino cargador de arranque UNO
 - 14 pines de E / S digitales, 6 salidas PWM
 - 6 entradas analógicas
 - Encabezado de ISP
 - Arduino UNO-R3 Shield Compatible
 - Programación Micro USB y fuente de alimentación
 - conectores Grove incorporados
 - Interruptor de alimentación de operación del sistema de 3,3 / 5V

2.2.1.3. Sistema Grove

- Grove es:

Un sistema de prototipos de conectores estandarizados y modulares. Grove adopta un enfoque de bloques de construcción para ensamblar productos electrónicos. En comparación con el sistema basado en puentes o soldaduras. Es más fácil conectar, experimentar y construir, y simplifica el sistema de aprendizaje, pero no hasta el punto en que se queda sin sentido. El sistema Grove consta de una unidad base, vástago, y varios módulos, ramas, con conectores estandarizados. Las

personas que originaron el sistema Grove, Seeedstudio, han tratado de usar tallos y ramas como parte del léxico Grove. No se necesita una unidad base para conectarse a los módulos Grove. Se puede usar un cable para pasar de los pines de la Raspberry Pi o Arduino a los conectores Grove.³

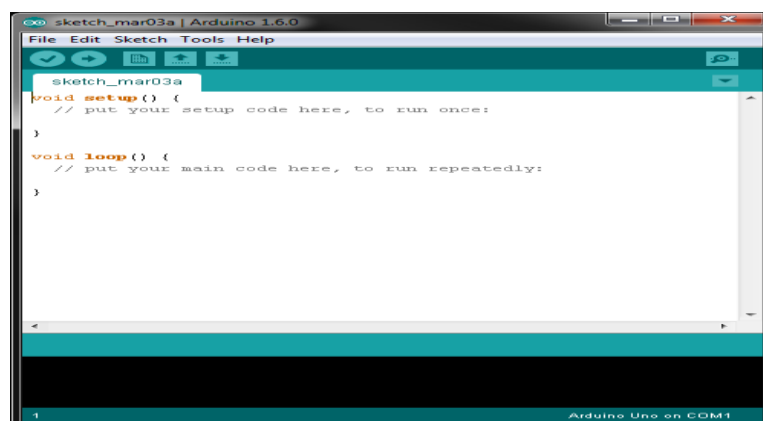
2.2.2. Software

Arduino no solo es una placa azul popular con la que hacer proyectos pequeños para encender y apagar leds. Arduino va mucho más allá simplificando el trabajo con microcontroladores y ofreciendo las siguientes ventajas: barato, multiplataforma, software libre y extensible mediante librerías en C++ y hardware libre y extensible.

- El software de Arduino es:

Un programa informático compuesto por un conjunto de herramientas de programación, un IDE. Dicho IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación constando de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica o GUI. Además, incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware.⁴

Figura 7. IDE de Arduino



Fuente: *Aprendiendo Arduino*. <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>. Consulta: 15 de agosto de 2020.

³ Seeed Technology. *Sistema Grove*. <https://wiki.seeedstudio.com/Seeeduino/>.

⁴ Aprendiendo Arduino. *Software Arduino*. <https://aprendiendoarduino.wordpress.com/>.

La plataforma Arduino tiene un lenguaje propio que está basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino como Java, Processing, Python, Mathematica, Matlab, Perl, Visual Basic, entre otros. Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes mencionados soportan. El entorno de desarrollo de Arduino es sencillo e intuitivo además puede descargarse gratuitamente desde su página oficial para distintos sistemas operativos.

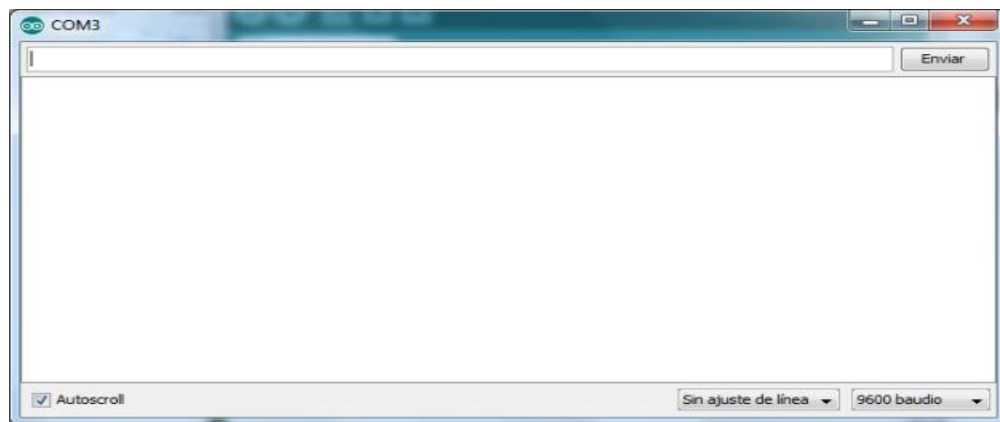
- Menú: Para hacer la conexión del IDE con la placa se va a Herramientas. Desde aquí se puede hacer la configuración para establecer la conexión con la placa Arduino. Haciendo click en Tarjeta, aparecerá una lista con los tipos de placa Arduino que el programa comprende.

En el campo Puerto Serial se selecciona el que corresponda a la placa que se conectará mediante USB. Si utilizamos Windows el puerto tendrá un nombre del estilo COMx pero en Linux será /dev/ttyUSBx donde x es un número. En caso de que aparezcan varios puertos serie y no se sepa cuál es el de nuestra placa se procede a desconectarla, anotando los puertos que aparecen. Reconectar la placa y volver a mirar la lista de puertos. El nuevo puerto que haya aparecido será el de nuestra placa.

- Botones de funciones comunes: Estos botones son accesos rápidos a ciertas acciones que también están disponibles mediante el menú. Los botones son los siguientes:
 - Verificar: comprueba y compila el código.
 - Cargar: además de compilar el código lo inyecta en la placa.

- Nuevo: crea un nuevo sketch.
- Abrir: abre un sketch previamente guardado.
- Guardar: almacena en disco los cambios realizados en el sketch.
- Monitor Serial: abre una nueva ventana desde la que se comunicará bidireccionalmente vía serie con la placa, es decir, se leerá la información que envía o proporcionarla. La ilustración 5 muestra esta ventana.

Figura 8. **Monitor serial de Arduino**



Fuente: elaboración propia, empleando Paint.

- Editor de texto: Un programa estándar consta de 3 partes. La primera consiste en la inclusión de librerías y la declaración de constantes o variables globales que se podrán utilizar en cualquier función del programa. La segunda es el método `setup()` que es el encargado de inicializar los dispositivos conectados a la placa y será ejecutado solo al iniciar el sistema. La tercera parte consiste en el método `loop()` que ejecutará su código continuamente, es decir, en modo bucle. Aquí es donde se escribirá la lógica de la placa Arduino. Como el lenguaje es muy similar a C es posible crear otros métodos para separar bloques funcionales y dejar ordenado el programa.

- Área de mensajes: Muestra la situación del programa al haber utilizado uno de los botones comunes.
- Consola de texto: Aquí aparecerán con mayor detalle los eventos del área de mensajes.

2.3. Periféricos de Arduino

Para lograr que un sistema domótico esté completo, es necesario, además del órgano central que controla el sistema, disponer una gama de sensores que puedan recoger datos y dar información real sobre la situación actual de la habitación de la vivienda que se está controlando. Utilizando esta información, el sistema domótico se comunicará con los actuadores para realizar la acción necesaria en cada caso. También es necesaria una interfaz para que el usuario pueda comunicarse con el sistema.

Los sensores y actuadores estarán conectados por medio de alambre, alambrado, a la placa de Arduino, algunos de ellos necesitarán de librerías extra para ser controlados y configurados, para ello es necesario buscar la librería pertinente e instalarla en la carpeta de librerías del entorno de desarrollo de Arduino.

Además, será necesario llamar a dicha librería al inicio del código. Si los sensores y actuadores son de índole digital, hay que tener en cuenta que solamente son necesarios dos valores para su control: HIGH y LOW. En cambio, si los elementos son analógicos, su uso es levemente más complejo ya que existen muchos valores posibles de lectura y escritura.

3. PROTEUS PROFESSIONAL 8

3.1. Proteus professional

- Proteus:

Es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción.⁵

Sin la utilización de la suite Proteus, el proceso para construir un equipo electrónico basado en un microprocesador se compone de cinco etapas (Diseño del circuito electrónico, diseño del circuito impreso, construcción del prototipo físico, desarrollo del software, pruebas de funcionamiento). Solo al final del proceso se es capaz de detectar los errores y cualquier problema exige volver a ejecutar el ciclo completo.

El depurado de errores puede convertirse en una labor difícil o tediosa con relación al tiempo y recursos, lo que conlleva un alto coste económico. Sin embargo, con la herramienta Proteus el proceso se acorta mucho al diseñar el circuito electrónico dentro del programa y ahí mismo simular su funcionamiento detectando los errores al instante.

⁵ Hubor-Proteus. *¿Qué es Proteus?* p.1.

Figura 9. **Proceso de creación de proyecto utilizando el software Proteus**



Fuente: Chile, CEC. *MICROPROCESADORES*.

<http://ingenieriadesitemasmicroprocesadores.blogspot.com/2016/10/>. Consulta: 12 de septiembre de 2020.

4. ELEMENTOS DE UN SISTEMA DOMÓTICO

4.1. Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

- Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, entre otros.
- Una magnitud eléctrica puede ser una resistencia eléctrica, como en una RTD, una capacidad eléctrica, como en un sensor de humedad o un sensor capacitivo, una tensión eléctrica, como en un termopar, una corriente eléctrica, como en un fototransistor, entre otros.

Un sensor es capaz de medir cualquier variable ambiental de su entorno con el fin de realizar acciones en función de los cambios en el entorno. Entre los sensores más utilizados para el control de una vivienda inteligente se encuentran:

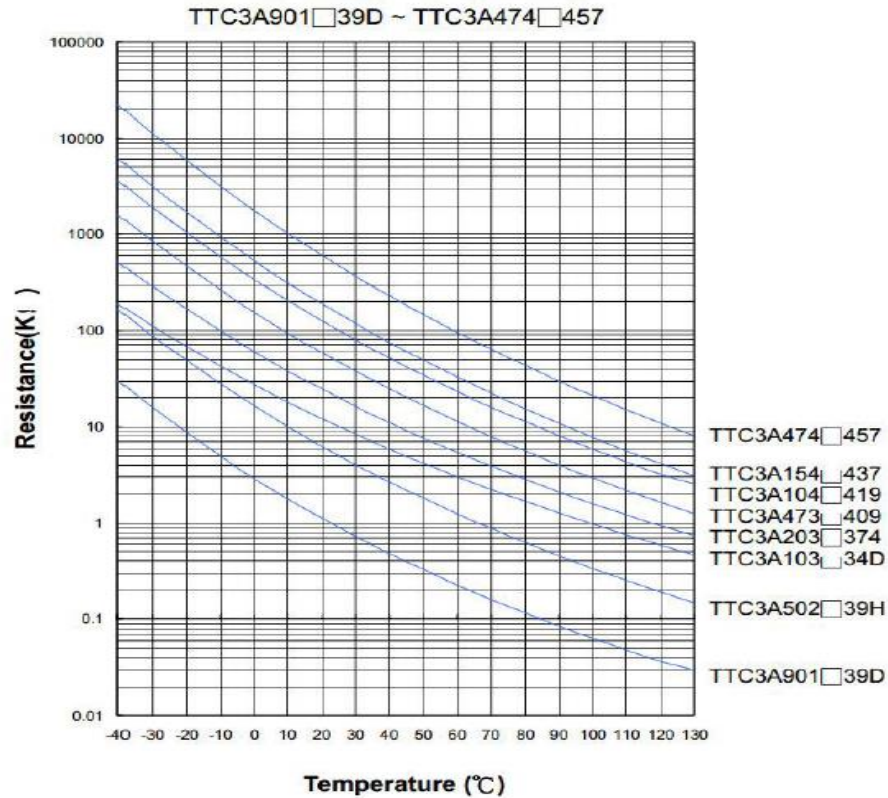
- Módulo de gas: el detector de gas hace que la vivienda gane en seguridad al detectar niveles altos de gas avisando al usuario para que se tomen las medidas correspondientes. Luego el sistema realizará alguna acción de seguridad como abrir ventanas de manera automática o desconectar la mayor parte de red eléctrica de la vivienda para evitar incendios y explosiones.

- Módulo PIR: otro elemento que interviene en la seguridad cuando no hay nadie en casa es un detector de movimiento. En caso de detectar suficiente movimiento se leerá un nivel alto y se avisará al usuario del movimiento. También se puede utilizar para confort al encender las luces o la calefacción de forma automática cuando hay movimiento en un área específica de la vivienda.
- Módulo de luz: dispositivo es capaz de detectar el nivel de intensidad de luz que hay en la habitación de forma analógica. El sistema leerá el voltaje y en caso de detectar un nivel bajo de luz podría encender las luces de la habitación siempre y cuando se detecte movimiento.
- Módulo de humedad y temperatura: algunos dispositivos son capaces de obtener varias mediciones en el mismo módulo. El módulo DHT11 es capaz de representar digitalmente la humedad ambiental medida en porcentaje, además de la temperatura en °C. Tiene una precisión decimal y dispone de su propia librería que contiene los métodos para recoger sus mediciones. Este módulo puede colocarse tanto dentro de la vivienda como en la zona externa de la casa, por ejemplo, un balcón o el jardín.

4.2. Sensores analógicos

Para explicar la programación de sensores analógicos se tomará como ejemplo un sensor analógico de temperatura o un termistor. En primer lugar, se debe conocer la curva característica de la resistencia del sensor, con ella se sabe el valor de resistencia que fuerza un valor predeterminado de temperatura.

Figura 10. **Curva característica de los termistores TTC3A**



Fuente: Yaxun. *Sensores NTC*. <http://fixedinductor.portuguese.sell.everychina.com/p-106533552-yellow-film-sealed-ntc-thermistor-with-enameled-wire-50k-ohm-100k-ohm-r.html>.

Consulta: 13 de septiembre de 2020.

Una vez que se conoce la resistencia del termistor se resuelve la ecuación de Steinhart-Hart que permite relacionar la resistencia eléctrica de un semiconductor en diferentes temperaturas. Seguidamente, se desarrolla un pequeño programa que permita resolver la ecuación y obtener la temperatura a partir de la resistencia eléctrica del termistor.

Los parámetros del termistor normalmente se determinan por el fabricante del dispositivo o bien pueden calcularse mediante 3 mediciones precisas de temperatura y resistencia del dispositivo en cuestión.

4.3. Actuadores

Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre elemento externo. Este recibe la orden de un regulador, controlador o de un Arduino, que es nuestro caso y en función a ella genera la orden para activar un elemento final de control como, por ejemplo, una válvula.

Existen varios tipos de actuadores, por ejemplo:

- Electrónicos
- Hidráulicos
- Neumáticos
- Eléctricos
- Motores
- Bombas

En determinadas ocasiones, es necesario un “driver” o manejador para mandar órdenes desde Arduino. Es importante mencionar que los pines de Arduino solo pueden manejar un máximo de 40 mA y recomendable usar 20 mA de forma continua. Así mismo, Arduino solo puede manejar un total de 200 mA de salida. Es decir que la corriente máxima que admite Vcc y GND son 200 mA.

4.3.1 Comunicadores

La correcta comunicación entre todos los elementos que conforman el sistema domótico es fundamental para el buen funcionamiento de este. El medio por el que circula la información puede ser alámbrico o inalámbrico. La alternativa de hacer la comunicación por aire, inalámbrica, es más barata debido a que no se tiene que gastar en cableado lo que permite un resultado más estético también. En cambio, las transmisiones son menos seguras y pueden ser alteradas por ruido.

Por otra parte, los sistemas que utilizan cables físicos para enviar los datos tienen un costo de instalación más elevado, pero ofrecen una transmisión más segura y robusta.

4.3.2 Arduino Shield Ethernet

El módulo Arduino Shield Ethernet conecta la placa Arduino a Internet rápidamente. Sólo hay que conectar este módulo en la placa Arduino, conectarlo a su red con un cable RJ45 y seguir algunas instrucciones para controlar el mundo Arduino a través de internet. El módulo Arduino Shield Ethernet permite a una placa Arduino conectarse a internet. Se basa en el chip de ethernet Wiznet W5100. El chip Wiznet W5100 ofrece una red (IP) capaz de usar TCP y UDP. Soporta hasta cuatro conexiones de socket simultáneas y posee su propia biblioteca.

Otra alternativa para conectar la placa a internet es un módulo Ethernet, como el ENC28J60 por ejemplo, que ofrece la misma funcionalidad y libera espacio en la placa.

4.3.3. Módulo Wifi para Arduino

Si se desea evitar los cables al hacer la conexión de la placa a internet, se debe optar por una conexión Wifi haciendo uso de un módulo Wifi. El módulo ESP8266 es el módulo Wifi para Arduino por excelencia.

El ESP8266 es un microprocesador de bajo coste con Wifi integrado fabricado por Espressif que se puede utilizar en los proyectos de electrónica con Arduino. Antes del ESP8266, las opciones disponibles para conectar un Arduino a Wifi (como el Wifi Shield) eran prohibitivamente caras.

4.3.4. Módulo XBee

Este elemento se comunica de forma inalámbrica utilizando el conjunto de protocolos de alto nivel de comunicación inalámbrica ZigBee. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y que necesiten un bajo consumo. Utiliza unas frecuencias comprendidas entre 865 MHz y 2,4GHz.

4.3.5. Módulo Bluetooth

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo. Opera mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda.

Bluetooth tiene la ventaja de simplificar el descubrimiento y configuración de los dispositivos, ya que éstos pueden indicar a otros los servicios que ofrecen,

lo que redundaría en la accesibilidad de estos sin un control explícito de direcciones de red, permisos y otros aspectos típicos de redes tradicionales. El WiFi utiliza el mismo espectro de frecuencia que Bluetooth con una potencia de salida mayor que lleva a conexiones más sólidas.

4.3.6. Módulos infrarrojos

Otra forma de comunicación sin medio físico es la utilización de rayos infrarrojos. Este tipo de comunicación consigue tener menos interferencias debido a la mayor frecuencia del espectro electromagnético en la que trabaja. Normalmente no es un método de comunicación entre placas Arduino, sino que es utilizado para contactar con los electrodomésticos del hogar.

4.3.7. Puerto serie

Un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadores y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

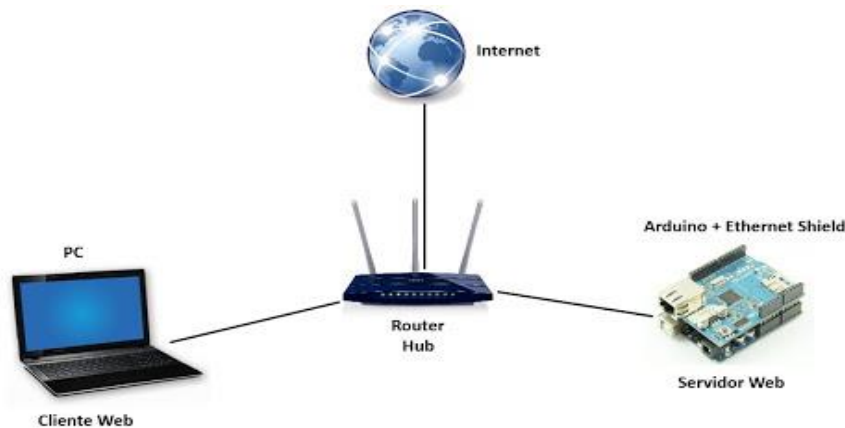
Es otro método de comunicación cableada que se usa para la comunicación entre un PC y una placa, o entre placas. El puerto serie consiste en un canal de recepción y otro de transmisión haciendo que el punto de recepción de un dispositivo esté conectado con el de transmisión del otro dispositivo. Todas las placas Arduino disponen de al menos un puerto serie compuesto por los pines digitales 0(rx) y 1(tx). Al conectar un cable de USB de la placa al ordenador estará aprovechando este puerto serie en el que se deba configurar la velocidad de datos en bits por segundo, baudios, para la transmisión de datos.

4.4. Servidor

Es el principal componente del sistema ya que es el responsable del control de todos los elementos de y desde este, el usuario podrá controlar todos los dispositivos conectados. Con la ayuda del servidor es posible controlar el sistema incluso fuera de casa, por ejemplo, desde el teléfono celular mediante una aplicación, enviando un mensaje de texto o desde el navegador de internet.

Existen varias alternativas para fungir como servidor. Utilizar una computadora portátil pero no es recomendable debido a que ocupará mucho espacio y se requiere que el sistema sea lo más compacto posible. Una mejor alternativa sería el ordenador de placa reducida de bajo costo Raspberry Pi desarrollada en el Reino Unido por la Raspberry Pi Foundation. Otra opción es utilizar la placa Arduino la cual no es tan potente como la Raspberry, pero permitirá armar un sistema robusto y más barato.

Figura 11. **Arduino como servidor web**



Fuente: Diymakers. *Crear servidor web con Arduino*. <http://diymakers.es/crear-servidor-web-con-arduino/>. Consulta: 17 de septiembre de 2020.

5. ARQUITECTURA DE SISTEMAS DOMÓTICOS

5.1. Arquitectura de los sistemas domóticos

La arquitectura de los sistemas domóticos se enfoca en las estructuras de su red, es decir, en la forma en la que los diferentes elementos se van a ubicar. Existen diferentes estilos de estructuras o arquitecturas de sistema, pero son las llamadas centralizadas y distribuidas las que son más utilizadas en la actualidad. Existen también otras arquitecturas como la descentralizada y la mixta que tienen cierta presencia en los hogares. Todos estos aspectos deben ser definidos por el experto en la instalación de domótica en viviendas y la decisión de qué arquitectura es mejor depende de varios aspectos como la distribución de las habitaciones de la vivienda o el tamaño y el alcance del sistema.

El servidor en todos los casos va a ser el responsable de comunicar al resto de elementos eventos globales como puede ser el apagado de todas las luces de la vivienda. En caso de que un dispositivo tenga que ser programado en un protocolo de comunicación distinto, éste ha de ser capaz de hacer la traducción necesaria.

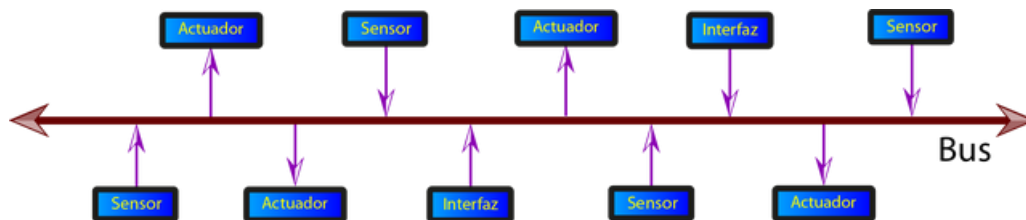
5.1.1. Distribuida

Según la página web de Hogartec, expertos en domótica, la arquitectura distribuida se da cuando cada uno de los actuadores y sensores funciona como un controlador que tiene la capacidad de actuar y enviar información al sistema según lo que se recibe de otros dispositivos, lo que significa que cada uno de los

dispositivos dentro del sistema cuenta con inteligencia propia y se puede controlar mediante diferentes actividades.

Las ventajas de un dispositivo distribuido son su seguridad de funcionamiento, permiten un profundo rediseño de la red y ampliaciones, sus productos son muy fiables, su coste y cableado no es tan grande como en un nivel 1. El inconveniente en este caso es que requiere bastante programación.

Figura 12. **Arquitectura distribuida**



Fuente: DomoFutur. *Arquitectura distribuida*.

<https://domoticaudem.wordpress.com/arquitectura-de-los-sistemas/>. Consulta: 23 de septiembre de 2020.

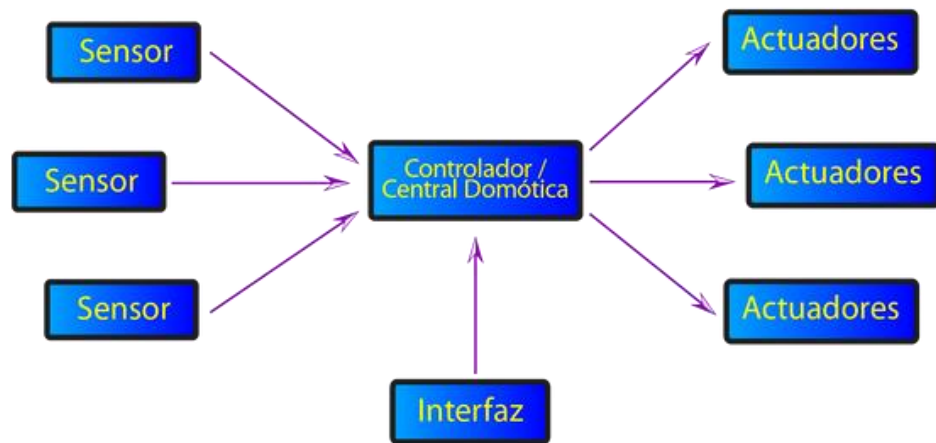
5.1.2. Centralizada

Los expertos en domótica de Hogartec describen la arquitectura centralizada como un controlador centralizado que es el que se encarga de enviar la información a los actuadores e interfaces, tal como han registrado los sensores o los propios usuarios con lo que se realizará una actividad específica. Si en este caso falta el controlador principal, el sistema deja de funcionar.

Entre sus ventajas destaca el hecho de que, tanto los sensores como los actuadores son de tipo universal, su operatividad e instalación son sencillas y tienen un coste bastante reducido en comparación con otros sistemas. Sus

inconvenientes es que conllevan un notable cableado, todos los sistemas de la central, no se puede ampliar demasiado su capacidad y requieren una interfaz de usuario.

Figura 13. **Arquitectura centralizada**



Fuente: DomoFutur. *Arquitectura centralizada*.

<https://domoticaudem.wordpress.com/arquitectura-de-los-sistemas/>. Consulta: 23 de septiembre de 2020.

6. DISEÑO DE UN SISTEMA DOMÓTICO

6.1. Metodología

Diseño de la investigación:

- Empleo de libros relacionados con el tema y consultas en internet.
- Bibliografía conceptual.
- Uso de placas y software Arduino para la elaboración y prueba de los circuitos necesarios.
- Uso del software de simulación Proteus 8 Professional para mostrar la simulación de sensores.

6.2. Diseño de prototipo de sistema domótico

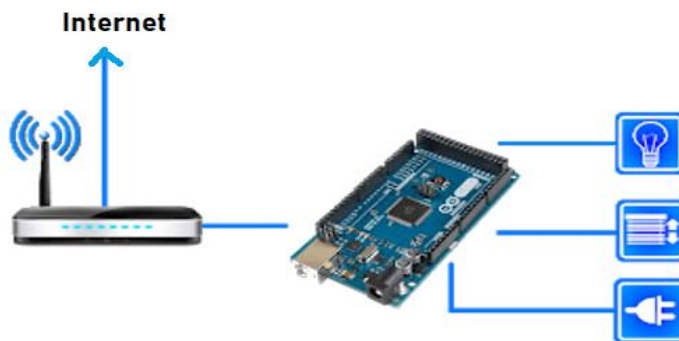
En este capítulo se realizará el diseño de un sistema domótico básico propio haciendo uso de todos los elementos ya vistos en esta investigación: la placa de Arduino UNO, placa Seeeduno, sensores, actuadores, un router para conectar el sistema a internet y un dispositivo móvil para el control remoto del sistema.

6.2.1. Estructura

Para el prototipo del sistema domótico que se propondrá en este punto, se ha decidido utilizar una arquitectura centralizada, siendo el servidor o controlador, una placa Arduino UNO. Esta placa actuará como órgano central del sistema en

lugar de una computadora ya que ésta, por sí misma, tiene las características necesarias para garantizar el funcionamiento del sistema.

Figura 14. **Estructura del sistema domótico centralizado utilizando Arduino**



Fuente: elaboración propia, empleando Paint.

Para poder realizar la conexión del Arduino a internet, se hará uso del Arduino Ethernet Shield. Este estará conectado al router mediante un cable RJ45, esto permitirá la comunicación con el sistema desde cualquier dispositivo conectado a la red local de dicho router.

Figura 15. **Arduino Ethernet Shield acplado y Arduino UNO conectado a router**



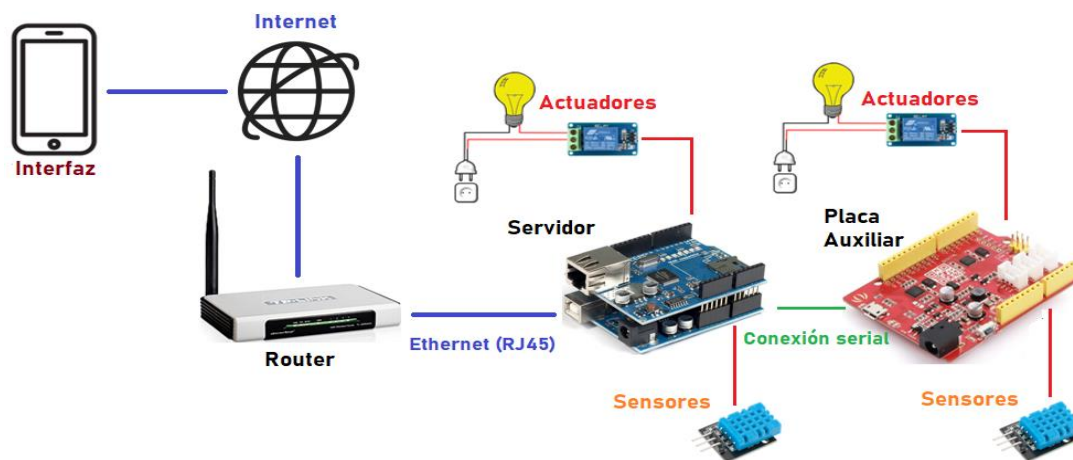
Fuente: Engineers Garage. *Arduino Ethernet*.

<https://www.engineersgarage.com/arduino/arduino-ethernet-shield-controlled-switch/>. Consulta:

27 de octubre de 2020.

Debido a que en un sistema domótico es necesario el uso de muchos sensores y actuadores, una sola placa de Arduino no alcanza para la conexión y manipulación de todos estos. Esto depende también del tamaño del sistema, pero en la mayoría de las ocasiones es necesario utilizar más de una placa. Para el complemento de la placa Arduino UNO se ha utilizado la placa Seeeduno debido a que, aunque es un microcontrolador de baja potencia, tiene buen rendimiento de procesamiento y facilita la conexión y gestión de sensores y actuadores. La comunicación entre la placa de Arduino UNO y la placa de Seeeduno es de tipo serial. Se toma en cuenta que, debido a que la conexión es en su mayoría alámbrica, se necesitará mucho cable para las conexiones ya que cada placa y cada sensor podrían colocarse en diferentes habitaciones. En este caso se simulará conectar la placa Arduino en la sala o recibidor y la placa Seeeduno en el comedor.

Figura 16. **Estructura completa del sistema domótico**

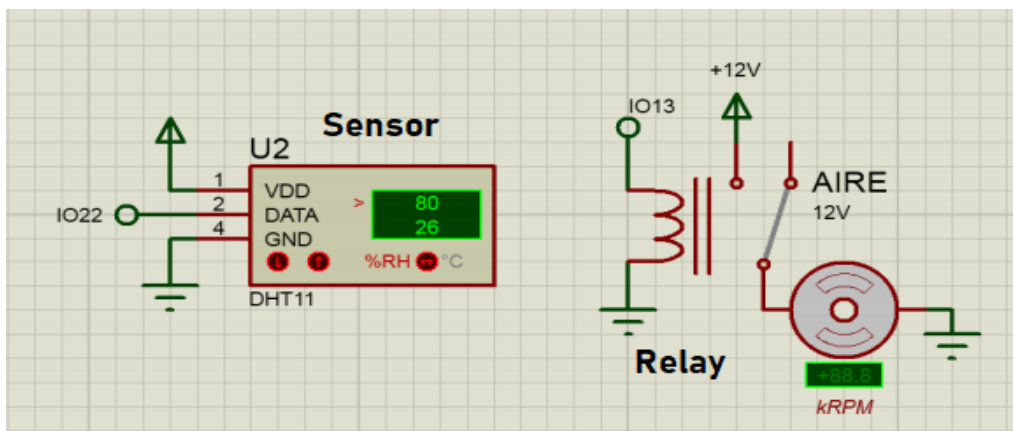


Fuente: elaboración propia, empleando Paint.

6.3. Funcionamiento

El software Proteus no cuenta con todos los modelos para la simulación de los dispositivos que componen el sistema domótico propuesto, por esta razón, no se puede comprobar el funcionamiento de todos los sensores que pueden ser utilizados en el sistema. A modo de ejemplo, a continuación, se observa la simulación del sensor de temperatura DHT11 presente en la biblioteca de transductores el software Proteus. Como se puede observar, el componente muestra la humedad relativa, así como la temperatura del área en dónde se encuentre el sensor. En este caso, el sensor se utiliza para obtener información sobre la temperatura de una habitación (el comedor, por ejemplo) y poder controlarla con ayuda de un actuador sobre el equipo de climatización. El sensor, al ser un elemento electrónico, se conecta directamente en una terminal del microcontrolador, pero para activar el actuador, como aire acondicionado por ejemplo, es necesario el uso de un relay debido a que se necesita controlar un voltaje más grande.

Figura 17. Simulación del sensor DHT11



Fuente: elaboración propia, empleando Proteus Professional 8.

Como la comunicación en el sistema se basa en una arquitectura centralizada, se debe de saber que las órdenes van a pasar siempre por el servidor, aunque las mismas no vayan dirigidas específicamente a él. La placa colocada en el receptor será capaz de controlar elementos de la habitación, como luces y temperatura y también actuará como servidor. Como ya se ha mencionado, los sensores son los encargados de captar las magnitudes físicas y con esa información se toma la decisión de modificarlas o no con ayuda de los actuadores.

Parte de la función del servidor es presentar la información al usuario ya sea por el mismo software de Arduino o por un navegador de internet utilizando el protocolo HTTP. Si existe alguna orden para el área del receptor, la placa actuará en consecuencia, si la orden es para el comedor, es decir, para otra placa, el servidor enviará dicha orden a dicha placa mediante el puerto serie reservado para la comunicación entre placas. Por otro lado, la placa colocada en el comedor será la responsable de controlar los elementos en esa habitación.

Se tienen dos opciones para comunicarnos con el sistema. La primera opción de comunicación es una comunicación serial, mediante el puerto serie. Los puertos serie son la forma principal de comunicar una placa Arduino con un ordenador. Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión). No obstante, pueden existir otros conductores para referencia de tensión, sincronismo de reloj, entre otros.

Figura 18. **Comunicación serial mediante software Arduino**



Fuente: elaboración propia, empleando el software de Arduino.

La otra opción de comunicación entre usuario y sistema es mediante el protocolo HTTP utilizado desde un navegador de internet. Esta opción permite controlar el sistema desde cualquier dispositivo móvil que soporte este protocolo y esté conectado a la misma red que el sistema. Para poder acceder al sistema, debemos colocar en la barra de búsqueda la dirección IP del servidor y el puerto que por defecto es el puerto 80.

Figura 19. **Comunicación con el sistema utilizando protocolo HTTP**



Fuente: elaboración propia, empleando Paint.

CONCLUSIONES

1. El tema de la domótica, hoy en día, es poco conocido en Guatemala por las personas, por lo que el mercado que abarca es, aún, muy pequeño.
2. La domótica ofrece muchas ventajas, como el confort, automatización de distintos sistemas como la iluminación, subida y bajada de persianas, gestión multimedia. También seguridad en el hogar, con una gran diversidad de detectores para controlar que no hay ninguna fuga o problema en la vivienda.
3. La plataforma de desarrollo de Arduino es suficiente y capaz, por sí misma, de dar atención y realizar el procesamiento de todas las acciones previstas. Debido a que las librerías de Arduino son libres y el hardware tiene un precio bastante económico, se puede concluir también que el sistema propuesto tiene un coste muy inferior al de los sistemas utilizados en viviendas de clase alta.
4. Las diferentes placas y sensores que ofrece la plataforma de desarrollo de Arduino permiten realizar las tareas de diseño de forma simples, así como la construcción del sistema físico. Debido a la interconexión entre varias placas es posible agregar más elementos al sistema en el futuro, esto quiere decir que el sistema propuesto es escalable.
5. Crear un sistema domótico propio es más económico que comprar uno ya hecho, pero hay que tener en cuenta que se necesitan conocimientos no solo de domótica sino también de diseño y elaboración de placas,

programación en diferentes lenguajes, protocolos de comunicación, entre otros.

RECOMENDACIONES

1. Se considera importante investigar sobre otros aspectos relacionados con los sistemas domóticos.
2. Se pide que se realicen mantenimientos preventivos para evitar cualquier desorden en las funciones programadas del sistema.
3. Es necesario asegurarse de que el sistema que se instale sea ampliable y escalable. En la mayoría de los proyectos los usuarios necesitarán añadir más dispositivos o nuevas funciones.
4. Es importante la elección de una tecnología que perdure en el tiempo y tenga la posibilidad de ampliarse fácilmente.
5. Si se desea acceso remoto a la vivienda, manipular elementos como la luz desde cualquier otro lugar, es importante asegurarse de tener acceso a internet ininterrumpido tanto en la casa controlada como en el lugar en donde se encuentre quien quiera comandar aspectos de forma remota.
6. Una vez instalado el sistema, es muy importante entender el funcionamiento del mismo para no darle un mal uso. Hay que conocer cuáles son sus límites y cuáles son sus posibilidades.

BIBLIOGRAFÍA

1. ANDRADES, Fran. *¿Qué es el Hardware Libre?* Para El Diario Turing, 2013. [en línea]. <https://www.eldiario.es/turing/Hardware-Libre_0_139986451.html>. [Consulta:13 de septiembre 2020].
2. Domótica Integrada. Instalaciones domóticas. *Tipos y elementos que debes conocer*. [en línea]. <<https://domoticaintegrada.com/instalaciones-domoticas/#El-elementosdeasinstalacionesdomotica>>. [Consulta: 03 de agosto 2020].
3. IES Camp de Morvarde. *JADIAZ. PLACA ARDUINO UNO*. [en línea]. <<http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>>. [Consulta:25 de noviembre 2020].
4. MÉNDEZ, Astrid. *Automatización y control de residencias, utilizando tecnologías de información y sistemas expertos*. Guatemala: 2008. 188 p.
5. Seeed. *The IoT Hardware Enabler*. Seeeduino V4.2. [en línea]. <<https://www.seeedstudio.com/Seeeduino-V4-2-p-2517.html>>. [Consulta:04 de agosto 2020].
6. Aprendiendo Arduino. *Hardware Arduino*. [en línea]. <<https://aprendiendoarduino.wordpress.com/category/perifericos/>>. [Consulta:12 de octubre 2020].

APÉNDICES

Apéndice 1. **Programación del sensor DHT en lenguaje Arduino y comunicación serial con el servidor.**

```
#include "DHT.h"  
#define tipoDHT DHT11  
  
char buffer[51];  
char habitacion[15];  
char dispositivo[15];  
char valor[10];  
  
#define pinLED 13  
#define pinDHT 8  
DHT dht(pinDHT, tipoDHT);  
  
void setup() {  
  pinMode(pinLED, OUTPUT);  
  pinMode(pinDHT, OUTPUT);  
  dht.begin();  
  Serial.begin(9600);  
}  
  
void loop() {
```

Continuación apéndice 1.

```
int hayMensaje;
```

```
hayMensaje = escucharMensaje();
```

```
if(hayMensaje)
```

```
actuar();
```

```
}
```

```
int escucharMensaje() {
```

```
if(Serial.available()) {
```

```
leerPuertoSerie();
```

```
return 1;
```

```
}
```

```
return 0;
```

```
}
```

```
//lectura pines
```

```
void leerPuertoSerie() {
```

```
char caracter;
```

```
int i=0;
```

```
memset(&buffer, 0, sizeof(buffer)); //borramos el contenido anterior de
```

```
buffer
```

```
while(Serial.available()) { //recogemos en buffer todo el mensaje
```

```
caracter = Serial.read();
```

```
delay(2);
```

```
buffer[i] = caracter;
```

```
i++;
```

Continuación apéndice 1.

}

if (strcmp(buffer, "") != 0) {

 descomponerMensaje();

}

}

void descomponerMensaje() {

 char* barra1;

 char* barra2;

 char* barra3;

 habitacion[0] = 0;

 dispositivo[0] = 0;

 valor[0] = 0;

 barra1 = strstr(buffer, "/") + 1;

 barra2 = strstr(barra1, "/") + 1;

 barra3 = strstr(barra2, "/") + 1;

 strncat(habitacion, buffer, barra1-buffer-1);

 strncat(dispositivo, barra1, barra2-barra1-1);

 strncat(valor, barra2, barra3-barra2-1);

}

////////////////////fin bloque escuchar

Continuación apéndice 1.

```
////////////////////inicio bloque actuar  
void actuar() {  
  if(strcmp(buffer,"salirDeCasa") == 0 ) //si introducimos el evento salir de  
  casa apagaremos todas las luces(leds)  
    salirDeCasa();  
  else if(strcmp(habitacion,"comedor") == 0 ) //si el mensaje va dirigido a  
  esta habitacion (comedor)  
    actuarEnComedor();  
  }  
  
void salirDeCasa() {  
  
  digitalWrite(pinLED, 0);  
  }  
  
void actuarEnComedor() {  
  ////////////////////activar led  
  if(strcmp(dispositivo,"led") == 0 ) {  
    digitalWrite(pinLED, atoi(valor));  
    Serial.print("comedor/led/ok"); //enviar respuesta al recibidor  
  }  
  
  ////////////////////leer temperatura  
  else if(strcmp(dispositivo,"temperatura") == 0 )  
    leerTemperatura();  
  
  ////////////////////leer humedad
```


Continuación apéndice 1.

```
else if(strcmp(dispositivo,"humedad") == 0 )  
  leerHumedad();  
}
```

```
void leerTemperatura() {
```

```
  int temperatura;
```

```
  String msg = "comedor/temperatura/";
```

```
  temperatura = dht.readTemperature();
```

```
  delay(25); //la lectura es un poco lenta
```

```
  if (isnan(temperatura)) {//si la lectura no da como resultado un numero
```

enviamos un mensaje de fallo

```
    msg.concat("fallo"); Serial.print(msg);} 
```

```
  else {
```

```
    msg.concat(temperatura); Serial.print(msg);} 
```

```
}
```

```
void leerHumedad() {
```

```
  int humedad;
```

```
  String msg = "comedor/humedad/";
```

```
  humedad = dht.readHumidity();
```

```
  delay(25); //la lectura es un poco lenta
```

```
  if (isnan(humedad)) {//si la lectura no da como resultado un numero
```

enviamos un mensaje de fallo

```
    msg.concat("fallo"); Serial.print(msg);
```

```
}
```

Continuación apéndice 1.

```
_else {  
  msg.concat(humedad); Serial.print(msg);  
  }  
}
```

Fuente: elaboración propia, empleando el software de Arduino.

Apéndice 2. Código del Ethernet Shield en lenguaje Arduino y comunicación HTTP con servidor

```
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
SoftwareSerial puertoSerieAComedor = SoftwareSerial(rxPin, txPin);

//Ethernet
#include "etherShield.h"
#include "ETHER_28J60.h"
static uint8_t mac[6] = {0x54, 0x55, 0x58, 0x10, 0x00, 0x24};
static uint8_t ip[4] = {192, 168, 1, 12};
static uint16_t port = 80;
ETHER_28J60 e;

char buffer[51];
char *bufferEther;
char habitacion[15];
char dispositivo[15];
char valor[10];
int hayMensaje;
boolean esperarMensajeComedor = false;

#define pinLED 7

void setup() {
    pinMode(pinLED, OUTPUT);
```

Continuación apéndice 2.

```
Serial.begin(9600);
Serial.println("Inicio test!");
Serial.println("Formato de los eventos: habitacion/dispositivo/valor/");
Serial.println();
puertoSerieAComedor.begin(9600);
e.setup(mac, ip, port);
}
```

```
void loop() {
  hayMensaje = escucharMensaje();
  if(hayMensaje){
    actuar();
    Serial.println();
    Serial.println("-----");
    Serial.println("-----");
    Serial.println();
  }
}
```

//////////////////////////////////////inicia bloque escuchar

```
int escucharMensaje() {
  if(Serial.available()) {
    leerPuertoSerie();
    return 1;
  }
}
```

Continuación apéndice 2.

```
else if(puertoSerieAComedor.available()) {
    leerPuertoSerieAComedor();
    return 2;
}
else if(bufferEther = e.serviceRequest()) { //por ethernet
    if (strcmp(bufferEther,"") != 0 && strcmp(bufferEther,"favicon.ico") != 0 &&
    strcmp(bufferEther,"1.0") != 0) {
        leerHTTP();
        return 3;
    }
}
return 0;
}
```

```
void leerPuertoSerie() {
    char caracter;
    int i=0;

    memset(&buffer, 0, sizeof(buffer)); //borramos el contenido anterior de
buffer
    while(Serial.available()) { //recogemos en buffer todo el mensaje
        caracter = Serial.read();
        delay(2);
        buffer[i] = caracter;
        i++;
    }
}
```

Continuación apéndice 2.

```
if (strcmp(buffer, "") != 0 ) {
    Serial.println("Descomponiendo mensaje del puertoSerie....");
    descomponerMensaje(false);
}
}

void leerPuertoSerieAComedor() {
    char caracter;
    int i=0;

    memset(&buffer, 0, sizeof(buffer)); //borramos el contenido anterior de
buffer
    while(puertoSerieAComedor.available()) { //recogemos en buffer todo el
mensaje
        caracter = puertoSerieAComedor.read();
        delay(2);
        buffer[i] = caracter;
        i++;
    }

    if (strcmp(buffer, "") != 0 ) {
        Serial.println("Descomponiendo mensaje del puertoSerieAComedor....");
        descomponerMensaje(false);
    }
}
```

Continuación apéndice 2.

```
void leerHTTP() {  
    Serial.println("Descomponiendo mensaje HTTP....");  
    descomponerMensaje(true);  
    esperarMensajeComedor = true;  
}
```

```
void descomponerMensaje(boolean ether) {  
    char* barra1;  
    char* barra2;  
    char* barra3;  
  
    habitacion[0] = 0;  
    dispositivo[0] = 0;  
    valor[0] = 0;  
  
    if(ether)  
        barra1 = strstr(bufferEther, "/") + 1;  
    else  
        barra1 = strstr(buffer, "/") + 1;  
    barra2 = strstr(barra1, "/") + 1;  
    barra3 = strstr(barra2, "/") + 1;  
  
    if(ether)  
        strncat(habitacion, bufferEther, barra1-bufferEther-1);  
    else
```

Continuación apéndice 2.

```
    strncat(habitacion, buffer, barra1-buffer-1);
    strncat(dispositivo, barra1, barra2-barra1-1);
    strncat(valor, barra2, barra3-barra2-1);

    if(ether) {
        Serial.print("BufferEther: "); Serial.println(bufferEther);
    }
    else {
        Serial.print("Buffer: "); Serial.println(buffer);
    }
    Serial.print("Habitacion: "); Serial.println(habitacion);
    Serial.print("Dispositivo: "); Serial.println(dispositivo);
    Serial.print("Valor: "); Serial.println(valor);
}
////////////////////////////////////fin bloque escuchar

////////////////////////////////////inicia bloque actuar
void actuar() {
    if(strcmp(buffer,"salirDeCasa") == 0 || strcmp(bufferEther,"salirDeCasa")
== 0) //si introducimos el evento salir de casa apagaremos todas las luces(leds)
    salirDeCasa();

    else if(strcmp(habitacion,"recibidor") == 0 ) //si el mensaje va dirigido a esta
habitacion (recibidor) ejecutamos la orden
    actuarEnRecibidor();
```


Continuación apéndice 2.

```
else if(strcmp(habitacion,"comedor") == 0 && hayMensaje != 2) //si el
mensaje va dirigido al comedor se lo enviamos
    comunicacionAComedor();
```

```
else if(strcmp(habitacion,"comedor") == 0 && esperarMensajeComedor ==
false) //si el mensaje lo recibimos del comedor (y NO era una orden http anterior)
    presentarInformacionComedor(); //presentamos los valores obtenidos
por el puerto serie
```

```
else if(strcmp(habitacion,"comedor") == 0) //si el mensaje lo recibimos del
comedor (y SI era una orden http anterior)
    imprimirHTMLcomedor(); //presentamos los valores obtenidos por el
puerto serie y por html
}
```

```
void salirDeCasa() {
    Serial.println("Saliendo de casa.... Apagar luces");
    digitalWrite(pinLED, 0);

    if(hayMensaje != 3)
        puertoSerieAComedor.print(buffer);

    else{
        puertoSerieAComedor.print(bufferEther);
        imprimirHTML();
    }
}
```

Continuación apéndice 2.

```
        e.print("<p>"); e.print("Saliendo de casa... Apagar luces");
e.print("</p>");
    e.respond();
}
}
```

```
void actuarEnRecibidor() {
    if(strcmp(dispositivo,"led") == 0 && strcmp(dispositivo,"led") == 0)
        digitalWrite(pinLED, atoi(valor));
    if(hayMensaje == 3) {
        imprimirHTML();
        e.respond();
    }
}
```

```
void comunicacionAComedor() {
    if(hayMensaje != 3) //si el mensaje no viene desde http lo enviamos por
el puerto serie del comedor
        puertoSerieAComedor.print(buffer);
    else { //en caso de que la orden sea desde http enviamos el mensaje,
esperamos la respuesta del comedor, y la imprimimos
        puertoSerieAComedor.print(bufferEther);
        if(strcmp(dispositivo,"temperatura") == 0 ||
strcmp(dispositivo,"humedad") == 0 ); //dejamos que el comedor lea el sensor
DHT
```

Continuación apéndice 2.

```
        delay(50);
    }
}
```

```
void imprimirHTML() {
    e.print("<h1>Servidor Web Arduino</h1>");
    e.print("<p>Formato de los eventos: /habitacion/dispositivo/valor/</p>");
    e.print("<p>-----</p><p></p>");

    e.print("<p>"); e.print("BufferEther: "); e.print(bufferEther); e.print("</p>");
    e.print("<p>"); e.print("Habitacion: "); e.print(habitacion); e.print("</p>");
    e.print("<p>"); e.print("Dispositivo: "); e.print(dispositivo); e.print("</p>");
    e.print("<p>"); e.print("Valor: "); e.print(valor); e.print("</p>");
    e.print("<p>-----</p><p></p>");
}
```

```
void presentarInformacionComedor() {
    if(strcmp(dispositivo,"led") == 0 && strcmp(valor,"ok") == 0) {
        Serial.println("Accion sobre el led del comedor efectuada");
    }
    else if(strcmp(dispositivo,"temperatura") == 0 ) {
        Serial.print("Temperatura del comedor: "); Serial.print(valor);
Serial.println("gradosC");
    }
    else if(strcmp(dispositivo,"humedad") == 0 ) {
```

Continuación apéndice 2.

```
        Serial.print("Humedad del comedor: "); Serial.print(valor);  
Serial.println("%");  
    }  
}
```

```
void imprimirHTMLcomedor() {  
    esperarMensajeComedor = false;  
    presentarInformacionComedor();  
    imprimirHTML();  
  
    if(strcmp(dispositivo,"led") == 0 && strcmp(valor,"ok") == 0) {  
        e.print("<p>"); e.print("Accion sobre el led del comedor efectuada");  
    }  
    else if(strcmp(dispositivo,"temperatura") == 0 ) {  
        e.print("<p>"); e.print("Temperatura: "); e.print(valor);  
e.print("gradosC</p>");  
    }  
    else if(strcmp(dispositivo,"humedad") == 0 ) {  
        e.print("<p>"); e.print("Humedad: "); e.print(valor); e.print("%</p>");  
    }  
  
    e.respond();  
}
```

Fuente: elaboración propia, empleando el software de Arduino.