

INFORME FINAL DE INVESTIGACIÓN

Sistema Digital para el Control Central
del Sistema Eléctrico Utilizando
Tecnologías Arduino y Android

**DOCENTE INVESTIGADOR PRINCIPAL:
LIC. ROBERTO CARLOS GAITÁN QUINTANILLA**

**DOCENTE INVESTIGADOR ASOCIADO:
TÉC. FERMÍN OSORIO GÓMEZ**

ITCA-FEPADE CENTRO REGIONAL SAN MIGUEL

FEBRERO 2017



ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL
SANTA TECLA, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA



INFORME FINAL DE INVESTIGACIÓN

Sistema Digital para el Control Central
del Sistema Eléctrico Utilizando
Tecnologías Arduino y Android

DOCENTE INVESTIGADOR PRINCIPAL:
LIC. ROBERTO CARLOS GAITÁN QUINTANILLA

DOCENTE INVESTIGADOR ASOCIADO:
TÉC. FERMÍN OSORIO GÓMEZ

ITCA-FEPADE CENTRO REGIONAL SAN MIGUEL

FEBRERO 2017



ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL
SANTA TECLA, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA



Rectora

Licda. Elsy Escolar SantoDomingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

**Dirección de Investigación
y Proyección Social**

Ing. Mario Wilfredo Montes, Director

Ing. David Emmanuel Ágreda

Inga. Lorena Victoria Ramírez de Contreras

Sra. Edith Aracely Cardoza

Director Centro Regional San Miguel

Ing. Mario Alsides Vásquez Cruz

629.89

G144s

sv

Gaitán Quintanilla, Roberto Carlos, 1976-
Sistema digital para el control central del sistema
eléctrico utilizando tecnologías Arduino y Android /
Roberto Carlos Gaitán Quintanilla, Fermín Osorio
Gómez. -- 1ª ed. - Santa Tecla, La Libertad, El Salv. :
ITCA Editores, 2017
64 p. ; 28 cm.

ISBN : 978-99961-50-57-9

1. Control de procesos 2. Control eléctrico. 3. Control
remoto. 4. Máquinas y herramientas control numérico
I. Osorio Gómez, Fermín, 1984- coaut. II. Título.

Autor

Lic. Roberto Carlos Gaitán Quintanilla

Coautor

Téc. Fermín Osorio Gómez

Tiraje: 13 ejemplares

Año 2017

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica y el sector empresarial, como un aporte al desarrollo del país. El contenido de este informe de investigación no puede ser reproducido parcial o totalmente sin previa autorización escrita de la Escuela Especializada en Ingeniería ITCA-FEPADE. Para referirse al contenido debe citar el nombre del autor y el título del documento. El contenido de este Informe es responsabilidad de los autores.

Escuela Especializada en Ingeniería ITCA-FEPADE
Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio web: www.itca.edu.sv

TEL: (503)2132-7423

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA	4
2.1.	DEFINICIÓN DEL PROBLEMA	4
2.2.	ANTECEDENTES	5
2.3.	JUSTIFICACIÓN	5
3.	OBJETIVOS	6
3.1.	OBJETIVO GENERAL	6
3.2.	OBJETIVOS ESPECÍFICOS	6
4.	HIPÓTESIS	6
5.	MARCO TEÓRICO	7
6.	METODOLOGÍA DE LA INVESTIGACIÓN	11
6.1.	ANÁLISIS DEL SOFTWARE	11
7.	RESULTADOS	18
7.1.	IMPLEMENTACIÓN DEL SOFTWARE	24
7.2.	PRODUCTOS	52
7.3.	DISEÑO DEL CIRCUITO IMPRESO	52
7.4.	DIAGRAMA UNIFILAR	54
7.5.	CAJA DE CONTROL CENTRAL	55
7.6.	DIAGRAMA CONCEPTUAL DEL SISTEMA DIGITAL	56
7.7.	IMÁGENES DEL PROCESO DE DESARROLLO	57
7.8.	SEGMENTACIÓN DE LA RED ELÉCTRICA	59
8.	CONCLUSIONES	61
9.	RECOMENDACIONES	61
10.	GLOSARIO	61
11.	BIBLIOGRAFÍA	¡ERROR! MARCADOR NO DEFINIDO.
12.	ANEXO. INSTALACIÓN DE CLIENTE JAVA PARA LECTURA DE PETICIONES	63

1. INTRODUCCIÓN

La necesidad de encontrar soluciones al ahorro de consumo eléctrico, motivó llevar a cabo una investigación referente a métodos o técnicas de ahorro y eficiencia del uso de la electricidad. Se encontró que existen diversas formas de ahorrar energía eléctrica, una de ellas es logrando una educación eficaz del personal para el consumo racional o eficiente y cambiando los equipos existentes por otros de tecnología de menor consumo.

Sin embargo, para el uso eficiente y ahorro de la energía eléctrica con la mínima intervención humana, se pueden desarrollar sistemas de control automático. Entre éstos se investigaron los sistemas domóticos, los Controladores Lógicos Programables (PLC) y los sistemas autómatas programables como LOGO. Si bien algunos de estos sistemas y equipos podrían lograr hacer una función similar a los resultados esperados con este proyecto, tienen la limitante que el costo de adquisición es elevado en relación al costo de la tecnología Arduino y el desarrollo de un software sencillo y amigable. El sistema más económico y con la tecnología de actualidad es el sistema de control inteligente a través de Arduino. Aprovechando las ventajas de este dispositivo electrónico, es posible programar sus seis pines de salida eléctrica, a través de los pulsos eléctricos de 5V DC. En sus pines digitales y analógicos en combinación a una tarjeta de relés electrónicos de 12V DC, es posible controlar la acción de contactores que administran corrientes y voltajes en AC para control de potencia eléctrica en sistemas de iluminación, aires acondicionados y ventilación.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. DEFINICIÓN DEL PROBLEMA

La Escuela Especializada en Ingeniería ITCA-FEPADE, Regional San Miguel, para el ciclo I-2016 atiende 3 carreras: Técnico en Ingeniería en Sistemas Informáticos, Técnicos en Ingeniería Civil, Técnico en Ingeniería Eléctrica, atendiendo alrededor de 333 estudiantes de primero y segundo año de dichas carreras, repartidos en 5 edificios académicos, entre los cuales están 10 aulas académicas, 5 Laboratorios de cómputo, 1 taller de mantenimiento de computadoras, 4 talleres de eléctrica, 1 taller de civil y 4 oficinas administrativas, en las que existe un alto consumo de energía eléctrica promediando los \$3,259.21 mensuales. Esta red eléctrica se ha dividido en 8 secciones las que satisfacen las necesidades de las cargas respectivas en cada una de las aulas, laboratorios, talleres y oficinas administrativas.

Todas las aulas, centros de cómputo, talleres y oficinas cuentan con servicio de luz eléctrica. Los centros de cómputo y oficinas administrativas cuentan además con el servicio de aire acondicionado. En todas las oficinas administrativas existen computadoras, equipo de respaldo eléctrico y otros dispositivos electrónicos asignados y utilizados por el personal, quienes al finalizar su jornada laboral ya sea clase o trabajo administrativo deben apagar luces, computadoras y aires acondicionados si es el caso. Sin embargo, a menudo se presenta el problema que se encuentran computadoras, luces o aires acondicionados encendidos en las unidades administrativas o académicas mencionadas anteriormente,

ya que los usuarios olvidan apagarlos. Al no existir un mecanismo de control central y accesible al personal administrativo, hay que esperar hasta el siguiente día para ingresar a la unidad académica o administrativa y apagar la luz, aire acondicionado, computadora u otro dispositivo del que se trate, lo que ocasiona un gasto innecesario de energía eléctrica, elevando así al final de mes el monto por consumo reflejado en el recibo correspondiente.

A raíz de eso se presenta la siguiente interrogante:

¿El desarrollo de un sistema digital para el control central del sistema eléctrico utilizando tecnología Arduino permitirá la administración eficiente del flujo de energía eléctrica y mejorará el control del uso eficiente de aulas, laboratorios y talleres?

2.2. ANTECEDENTES

La Escuela Especializada en Ingeniería ITCA-FEPADE Centro Regional San Miguel, es una institución que cuenta con una reconocida trayectoria académica, que tiene como objetivo formar profesionales integrales e impulsar la capacitación y el recurso humano en la zona oriental del país. Es una institución gubernamental con administración privada por medio de la Fundación Empresarial para el Desarrollo Educativo, FEPADE.

Desde 1998 administra el antes conocido Instituto Tecnológico de San Miguel, ITESAM. En este Centro Regional se cuenta con tres carreras técnicas en ingeniería con una duración de 2 años: Eléctrica, Civil y de Sistemas Informáticos. En el Centro Regional, también se imparte cursos libres de educación continua en el área de Servicios de Desarrollo Profesional, consistentes en cursos, diplomados y capacitaciones a todo tipo de personas interesadas en desarrollarse profesionalmente. Además, el Centro Regional atiende requerimientos de formación profesional de empresas privadas y gubernamentales, así como los programas de formación promovidos por el INSAFORP.

Como toda Institución de Educación Superior (IES) realiza las actividades de investigación y proyección social.

2.3. JUSTIFICACIÓN

El alto consumo de energía eléctrica, el desgaste de luminarias, el sobre calentamiento por la subutilización de dispositivos eléctricos y electrónicos, hasta problemas laborales como multas y sanciones entre otros, son algunos de los criterios que le dan fuerza y validez al desarrollo del presente proyecto.

Al aplicar la tecnología Arduino permitirá que de forma automatizada se tenga acceso a los segmentos de la red eléctrica en ITCA-FEPADE Centro Regional San Miguel con un bajo costo de desarrollo, ya que se puede usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos de Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en un ordenador u otro dispositivo móvil, de forma local o remota dependiendo del complemento utilizado (Lenguaje de programación para la interfaz). La placa Arduino puede ser montada o comprarla

ya lista para usar, y el software para el desarrollo de aplicaciones es abierto y se puede descargar gratis de la web del desarrollador¹.

El proyecto permitió que de forma local a través de una placa Arduino pueda tenerse acceso a los segmentos de la red eléctrica y permitir o denegar así el paso de corriente a un segmento o aula, para la activación de luces o carga conectada a la red eléctrica. Así mismo, a través de una interfaz instalable en el computador o dispositivo móvil, tener acceso remoto vía internet a los mismos elementos descritos en este párrafo, lo que permitió que desde cualquier lugar del mundo se pudiera habilitar o deshabilitar el flujo de corriente eléctrica en los segmentos de la red eléctrica en ITCA-FEPADE San Miguel.

Además, el funcionamiento del proyecto dentro del Centro Regional, permitió a los estudiantes actuales contar con elementos demostrativos del uso de Arduino, siendo este un elemento que ayudó a fortalecer sus competencias como Técnicos en Ingeniería Eléctrica y Sistemas Informáticos.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Elaborar un sistema digital para el control central del sistema eléctrico utilizando tecnología Arduino en ITCA-FEPADE Centro Regional San Miguel.

3.2. OBJETIVOS ESPECÍFICOS

- Identificar y seccionar los segmentos de red de distribución eléctrica en ITCA-FEPADE Centro Regional San Miguel.
- Construir los circuitos Hardware con Arduino para el control de los segmentos de la red de energía eléctrica en ITCA-FEPADE Centro Regional San Miguel.
- Construir la interfaz gráfica de usuario para la comunicación y acceso a los segmentos de red vía web, a través de dispositivos móviles.
- Elaborar una aplicación web para el control central y acceso a los segmentos de la red eléctrica en ITCA-FEPADE Centro Regional San Miguel.

4. HIPÓTESIS

¿El desarrollo de un Sistema digital para el control central del sistema eléctrico utilizando tecnología Arduino permitirá la administración eficiente del flujo de energía eléctrica y mejorará en el control del uso eficiente de aulas, laboratorios y talleres?

Disponibile en “<https://www.Arduino.cc/en/Main/Software>”

5. MARCO TEÓRICO

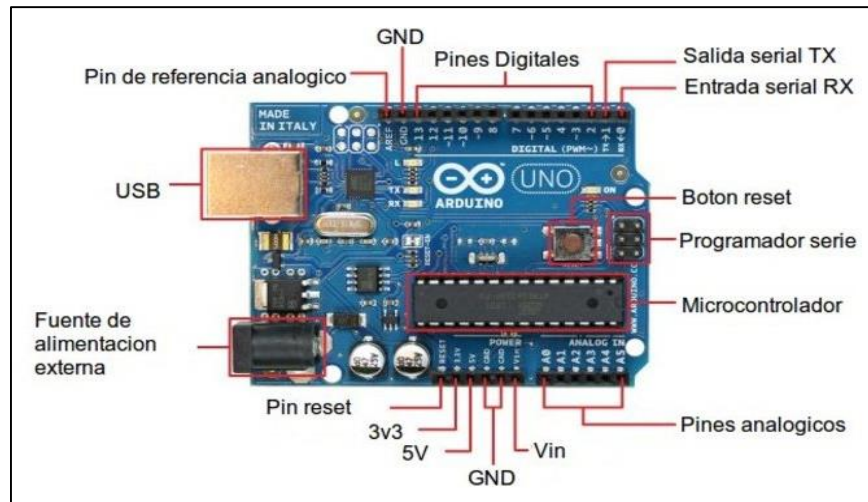


Figura1. Estructura de Arduino Uno

¿Qué es Arduino?

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital. Sus creadores son el zaragozano David Cuartielles, ingeniero electrónico y docente de la Universidad de Mälmo, Suecia y Massimo Banzi, italiano, diseñador y desarrollador Web. El proyecto fue concebido en Italia en el año 2005²

Shields para Arduino

Las Shields son placas de circuitos modulares que se montan unas encima de otras para dar funcionalidad extra a un Arduino. Esta Shields son apilables unas sobre otras.³

² Documento disponible en "<http://www.aragoninvestiga.org/La-domotica-una-nueva-tecnologia-para-una-vida-mas-comoda-en-el-hogar>"

³ Documento disponible en "<https://aprendiendoArduino.wordpress.com/2015/03/23/shields-para-Arduino>"

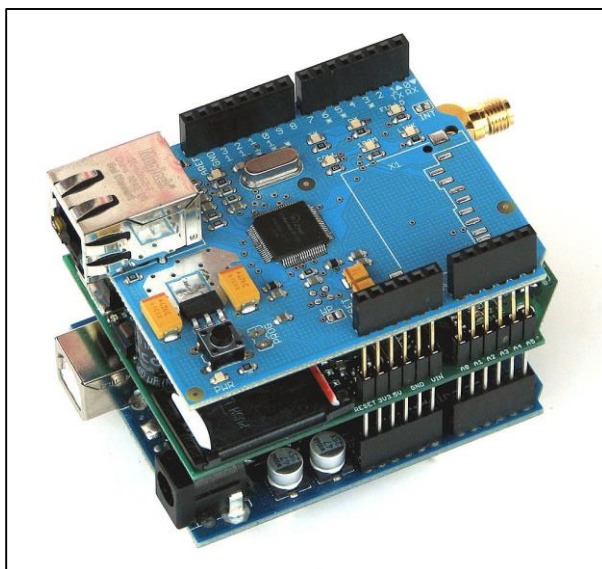


Figura 2: Placa Arduino expandida a dos Shield

Las Shields se pueden comunicar con el Arduino bien por algunos de los pines digitales o analógicos o bien por algún bus como el SPI, I2C o puerto serie, así como usar algunos pines como interrupción. Además, estas Shields se alimentan generalmente a través del Arduino mediante los pines de 5V y GND.

En muchos casos los Shields vienen sin soldar las patillas o en algunos casos sin algunos componentes soldados.

Cada Shield de Arduino debe tener el mismo factor de forma que el standard de Arduino con un espaciado de pines concreto para que solo haya una forma posible de encajarlo.

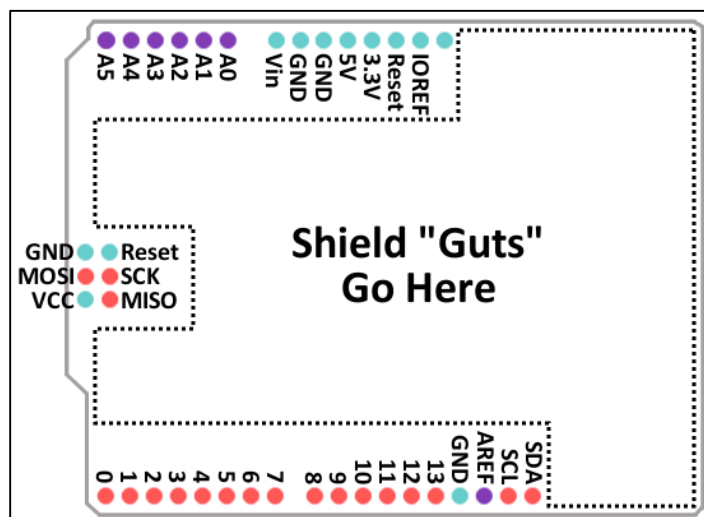


Figura 3: Pines ArduinoShield

Se debe tener en cuenta que cuando se instala un shield, habrá que leer la documentación que dirá si inhabilita algunas de las I/O si usa un bus y qué requisitos tiene para su utilización. Generalmente las Shields vienen con un ejemplo o una librería para su uso.⁴

La placa Arduino se comunica con el módulo W5100 y la micro-SD utilizando el bus SPI (mediante el conector ICSP). Esto se encuentra en los pines digitales 11, 12 y 13 en el modelo UNO y en los pines 50, 51 y 52 del modelo MEGA.

En ambas placas, el pin 10 es utilizado para seleccionar el W5100 y el pin 4 para la micro-SD. Estos pines no pueden ser utilizados para otros fines mientras la Ethernet Shield esté conectada. El en MEGA, el pin SS (53) no es utilizado, pero debe dejarse como salida para que el bus SPI funcione correctamente.

ArduinoWifiShield

Permite conectar un Arduino a Internet mediante Wifi y hace uso de la librería WiFi Library.

También dispone de un slot para una tarjeta micro SD. Este shield se comunica con Arduino a través del bus SPI mediante los pines ICSP y usa los pines 4 y 10 al igual que la Shield Ethernet. También hace uso del pin 7 que no puedes ser usado⁵



Figura 5: Arduino Shield Wifi

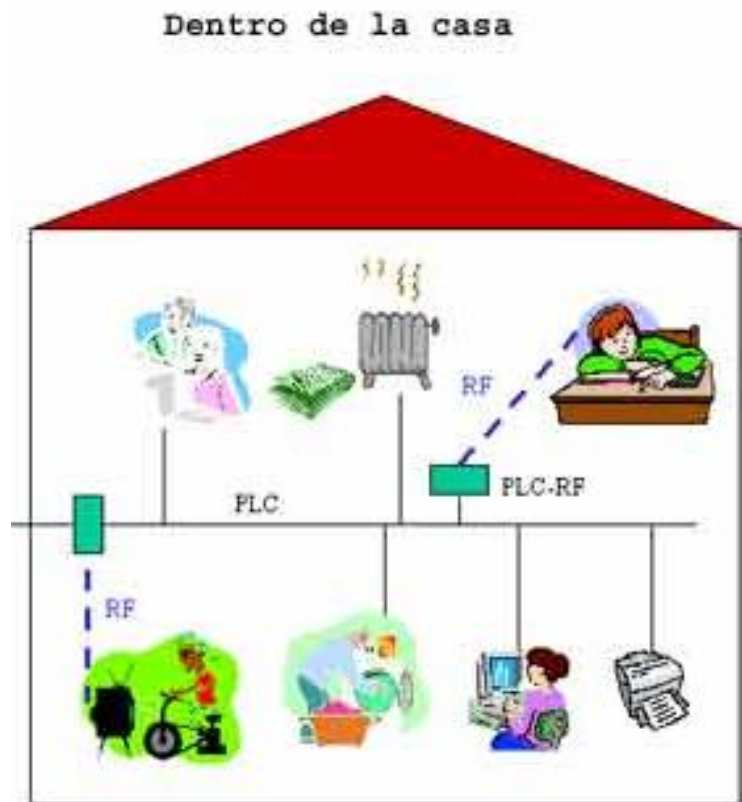
Debe tenerse en cuenta que, dado que la tarjeta SD y la wifi usan el mismo bus SPI, no pueden trabajar simultáneamente, lo que implica que se debe tenerse este aspecto en cuenta durante la programación si se usa ambos periféricos. Esta tarjeta también dispone de unos interfaces en placa para actualizaciones de firmware y diagnósticos.

⁴ Disponible en “<http://Arduino.cc/en/Main/ArduinoEthernetShield>”

⁵ Disponible en “<http://Arduino.cc/en/Main/ArduinoWiFiShield>”

La domótica: una nueva tecnología para una vida más cómoda en el hogar.

En los últimos cincuenta años las casas se han ido llenando de nuevos aparatos. Primero llegaron los electrodomésticos como el frigorífico o la lavadora que supusieron un cambio fundamental en las tareas domésticas y en la forma de vida. Hoy resulta difícil pensar cómo sería la vida sin ellos. El segundo cambio importante lo produjo la llegada de la televisión y el teléfono que modificaron los hábitos familiares y la manera de relacionarnos y conocer el mundo. Fueron el primer paso de lo que hoy se conoce como la sociedad de la información. No realizan una tarea física como los anteriores, sino que proporcionan toda una serie de servicios. La llegada de los móviles y la cada vez mayor difusión de Internet están suponiendo un nuevo paso en esta dirección. Dan la posibilidad de comunicarse en todo momento y en cualquier lugar, de recibir nuevos servicios y de participar en nuevos grupos y actividades.



Aunque este avance hacia la sociedad de la información es, en general, positivo, también ha supuesto un giro importante en las vidas de las personas y ha generado nuevos problemas y riesgos. Importantes grupos de población pueden sentirse y quedar realmente marginados de este cambio tanto por razones económicas como culturales o de edad. Se está realizando, tanto por parte de los grupos de investigación como de las administraciones que los dirigen, un importante esfuerzo para asegurar el acceso de todos a esta nueva sociedad de la información sin exclusiones y de manera equilibrada. Afortunadamente parece que la evolución de la tecnología favorece su difusión masiva tanto por la reducción de costos como por la facilidad de su utilización.

A pesar de estos grandes cambios, la estructura y los servicios que las casas proporcionan, no se han modificado demasiado. Siguen utilizando técnicas de fabricación y materiales similares sin integrar estas nuevas tecnologías y las posibilidades que ofrecen. Se busca en ellas protección, seguridad y confort, un espacio donde vivir, convivir y en ocasiones trabajar.

Por otra parte, las necesidades y la importancia que se da a estos aspectos van cambiando en la vida. Los usos que se otorga a las casas varían, por ejemplo, al formar una pareja, con la llegada de los hijos, cuando estos se hacen mayores y dejan el hogar y cuando por la edad y los problemas de salud se hace difícil vivir solos y se necesita de cuidados especiales. Sin embargo, las casas son estáticas, de hecho, se

hacen llamar inmuebles, y resulta difícil adaptarlas a estos cambios. A pesar de su alto coste, no siempre cubren plenamente los servicios que necesitamos de ellas⁶

Objetivo de la domótica: mejorar la vida en el hogar

La investigación en domótica se centra en mejorar esta situación con el uso integrado de diversas tecnologías para proporcionar nuevos servicios y mejorar los que ya se utilizan. La domótica, que tuvo su origen en la automatización de grandes edificios comerciales y de oficinas, se está extendiendo hoy a entornos domésticos o de pequeños negocios en los que se pretende combinar un diseño razonable del edificio (integrado en su entorno y utilizando técnicas arquitectónicas limpias y tradicionales) con el uso de las últimas tecnologías electrónicas para proporcionar servicios multimedia con costes razonables.

No se trata de llenar los hogares de nuevos artefactos sino más bien de que los que ya se utilizan sean más listos y sencillos de utilizar; que usen esa inteligencia y capacidad de comunicación para hacer las tareas diarias más fáciles y se adapten a las necesidades. La domótica hoy en día ha de ser amigable, fácil de usar, instalar y mantener, ubicua pero discreta y respetuosa con la privacidad.

Diferencia entre casa domótica / hogar digital

Una vivienda domótica dispone de un gran número de equipos y sistemas, principalmente autónomos, a los que hay que sumar diferentes redes, como la telefonía, las redes de datos (cableadas e inalámbricas), la televisión, electrodomésticos, equipamiento de audio y video, calefacción, aire-condicionado, seguridad, riego, iluminación, etc.

Para convertirse en un hogar digital, a una casa domótica le faltaría la convergencia de las comunicaciones, la informática y el entretenimiento gracias a las redes de banda ancha es una tendencia consolidada a nivel mundial.

6. METODOLOGÍA DE LA INVESTIGACIÓN

6.1. ANÁLISIS DEL SOFTWARE

REQUERIMIENTOS MÍNIMOS DE HARDWARE Y SOFTWARE

Los requerimientos mínimos en relación a hardware que se necesitarán para la implementación y buen funcionamiento del software serán los siguientes:

HARDWARE

- ✓ Monitor
- ✓ Teclado

⁶ Documento disponible en "<http://www.aragoninvestiga.org/La-domotica-una-nueva-tecnologia-para-una-vida-mas-comoda-en-el-hogar/>"

-
- ✓ Mouse
 - ✓ Case
 - ✓ 2 GB Disponible de Espacio en Disco duro.
 - ✓ Procesador Intel(R) Celeron CPU 900 @ 2.20 GHz o superior
 - ✓ Memoria RAM mínimo de 256 MB

SOFTWARE

- ✓ Sistema Operativo Windows XP O SUPERIOR
- ✓ XAMPP

REQUISITOS DEL SOFTWARE (Orientados a todos los requisitos que el sistema debe cumplir con el fin de resolver el problema y no convertirse en uno más)

- Control Arduino
 - Crear, modificar y desactivar usuarios.
 - Crear, modificar y desactivar pabellones.
 - Crear, modificar y desactivar aulas.
 - Crear, modificar y desactivar Arduinos.
 - Crear, modificar y desactivar pines de Arduino.
- Control eléctrico
 - Apagar y encender pines de Arduino.
 - Apagar y encender todos los pines correspondientes a un aula específica.

REQUISITOS DE SOFTWARE

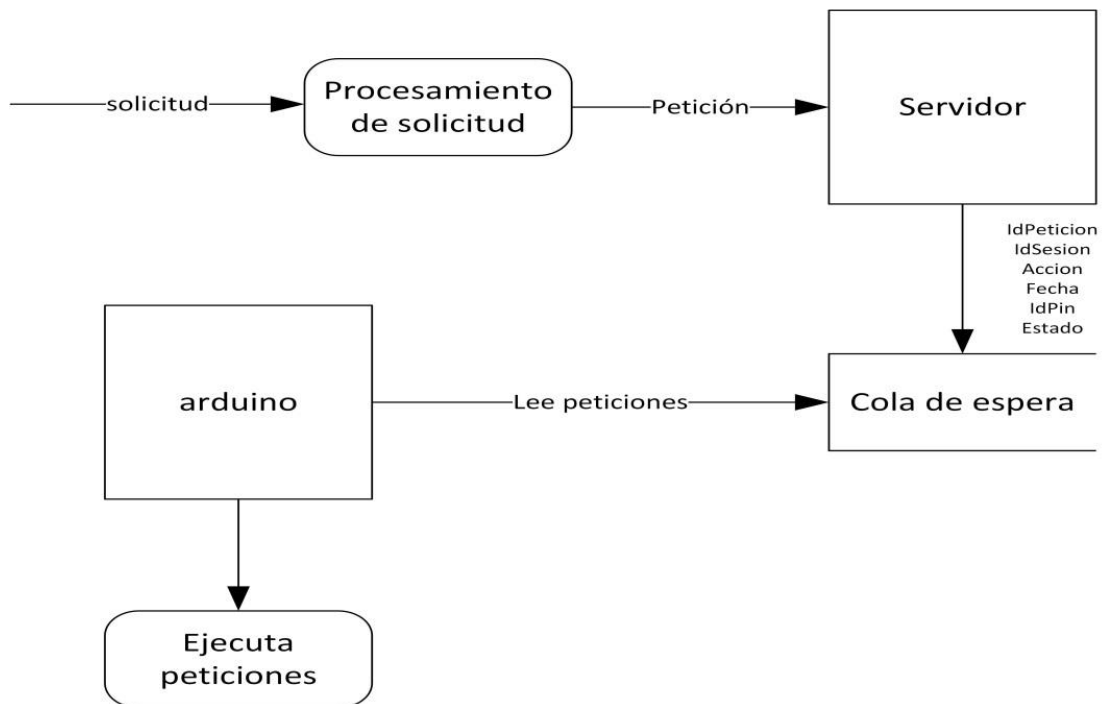
- General
 - Listado de usuarios.
 - Listado de pabellones.
 - Listado de aulas.
 - Listado de Pines de Arduino.

DIAGRAMAS DE FLUJOS DE DATOS

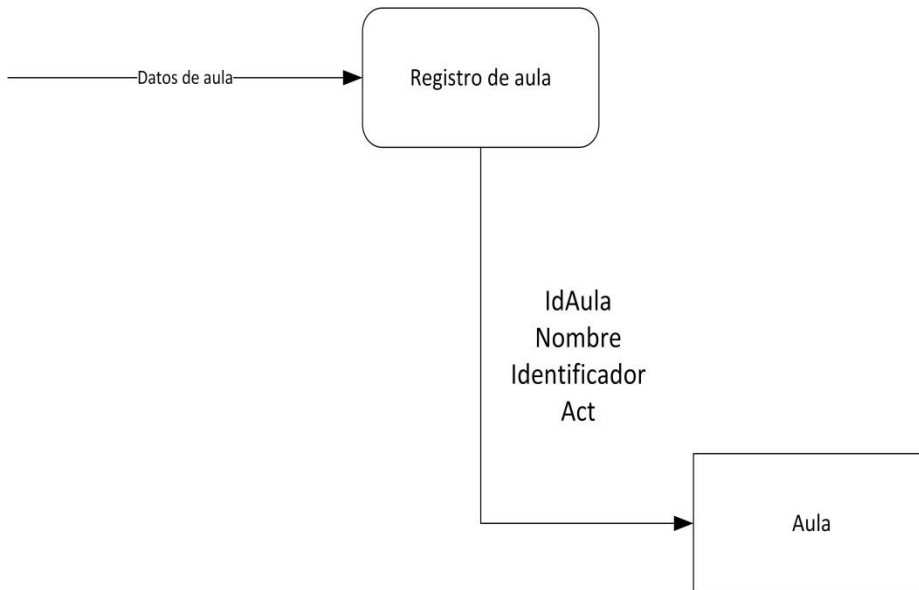
Primera fase



Segunda fase



Id	4	
proceso	Registrar aula	
Descripción	Su objetivo es guardar los datos de nuevas aulas	
Flujo de datos de origen		
Flujo	Origen	Descripción
Información del aula	Administrador	Procesa, revisa y guarda los datos de una nueva aula para agregar zonas de control de energía
Flujo de datos de destino		
Flujo	Destino	Descripción
Datos del aula	Base de datos	Almacena en la base de datos la información procesada y ordenada del aula
Resumen lógico		
El administrador ingresa la información de una nueva aula al sistema, este la revisa y analiza para posteriormente guardarla y activar las nuevas zonas de control.		

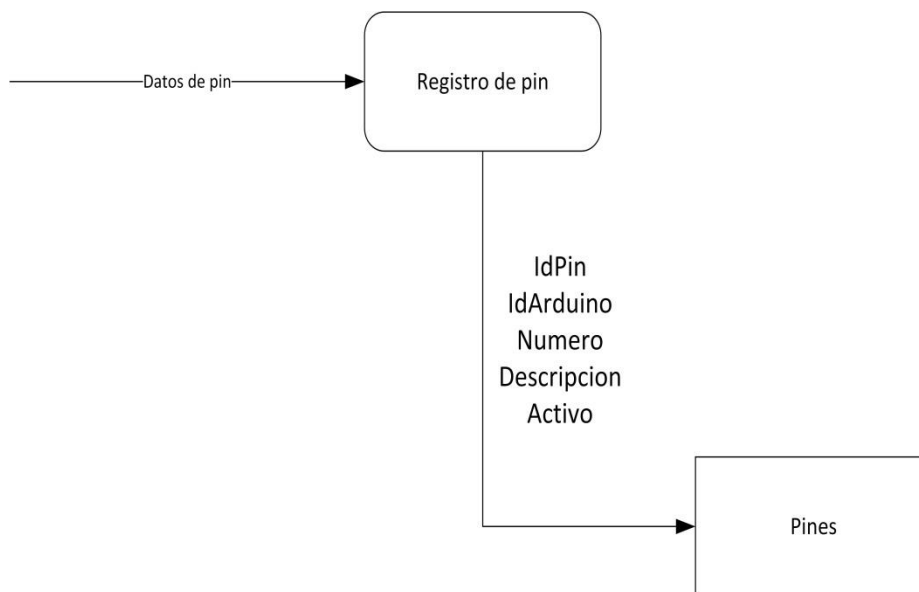


Id	6	
Proceso	Registrar pines	
Descripción	Su objetivo es realizar el registro de un nuevo pin	
Flujo de datos de origen		
Flujo	Origen	Descripción
Información del pin	Administrador	Procesa, revisa y guarda los datos de un nuevo pin

Flujo de datos de destino		
Flujo	Destino	Descripción
Datos del pin	Base de datos	Almacena en la base de datos la información procesada y ordenada de los pines
Resumen lógico		
El administrador ingresa la información de un nuevo pin al sistema, este la revisa y analiza para posteriormente guardarla y activar las nuevas zonas de control.		

Flujo de datos de registro de pines

Flujo de datos de registro de Arduino



d	5
proceso	Registrar Arduino
Descripción	Su objetivo es realizar el registro de Arduino al sistema
Flujo de datos de origen	
Flujo	Origen
Información del Arduino	Administrador
	Procesa, revisa y guarda los datos de un nuevo Arduino

Flujo de datos de destino		
Flujo	Destino	Descripción
Datos del Arduino	Base de datos	Almacena en la base de datos la información procesada y ordenada del Arduino
Resumen lógico		
El administrador ingresa la información al sistema, este la revisa y analiza para posteriormente guardarla y proceder a usar las funciones del nuevo Arduino		

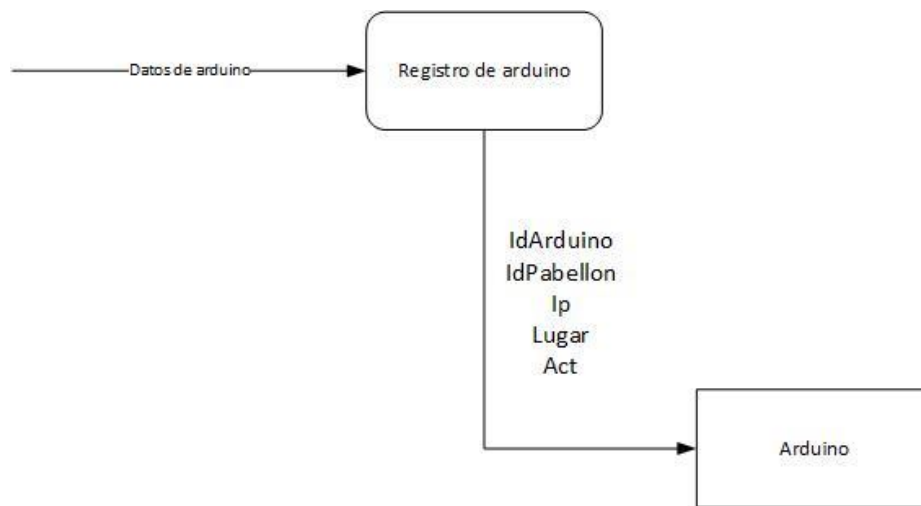
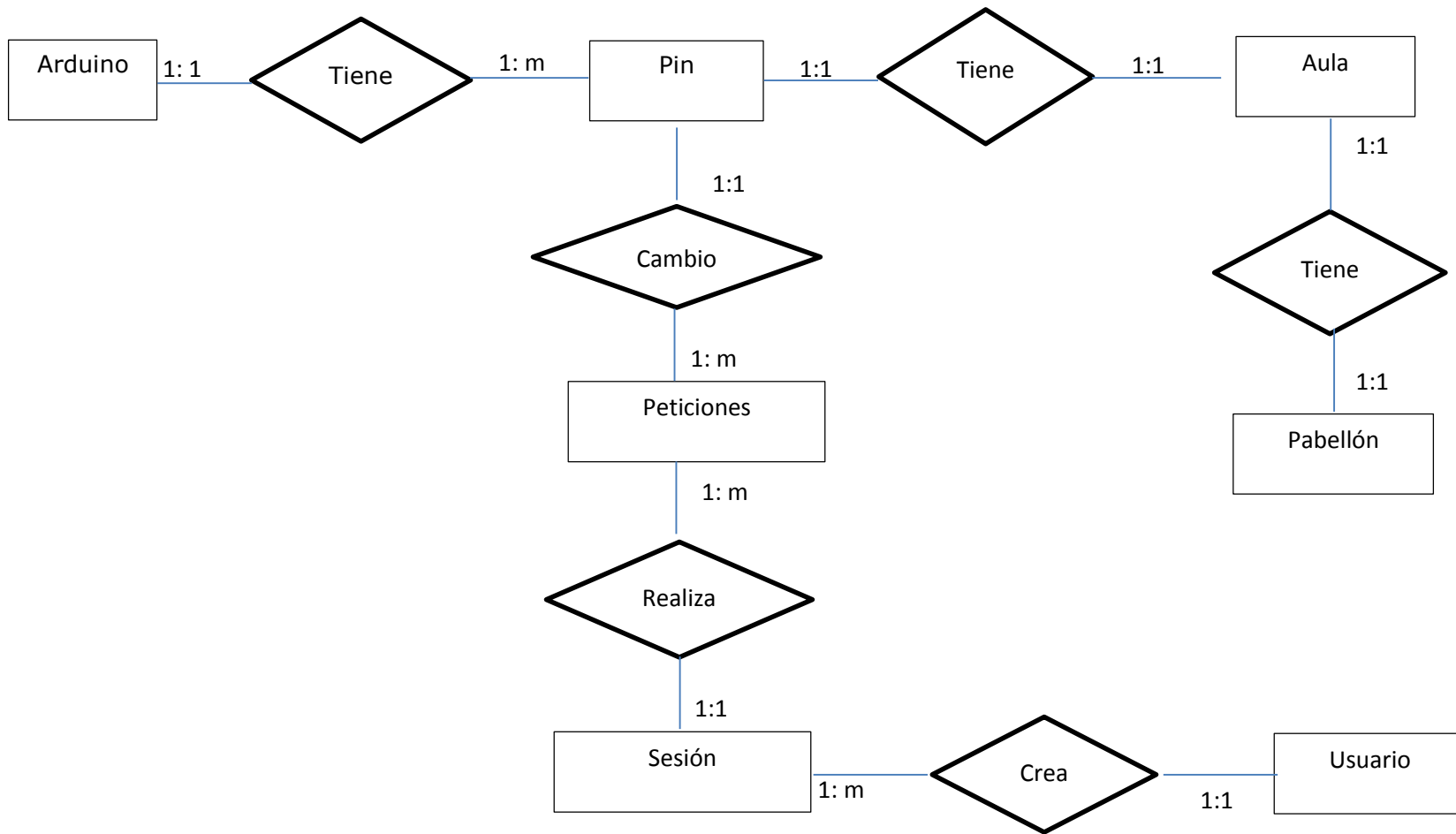


DIAGRAMA ENTIDAD RELACION



7. RESULTADOS

Diseño de la interfaz grafica



Imagen 1. Pantalla de acceso de sistema

Esta es la pantalla principal de acceso al sistema, consiste en un control de acceso con usuario y contraseña, al ingresarlos de forma correcta podrás acceder al control de sistema con los rangos de tu usuario respectivo.



Imagen 2. Pantalla principal de acceso

La pantalla principal de acceso es la encargada de mostrar los edificios a los cuales se tiene acceso de control de energía con el sistema, obteniendo los registros de la tabla “pabellones” de la base de datos.

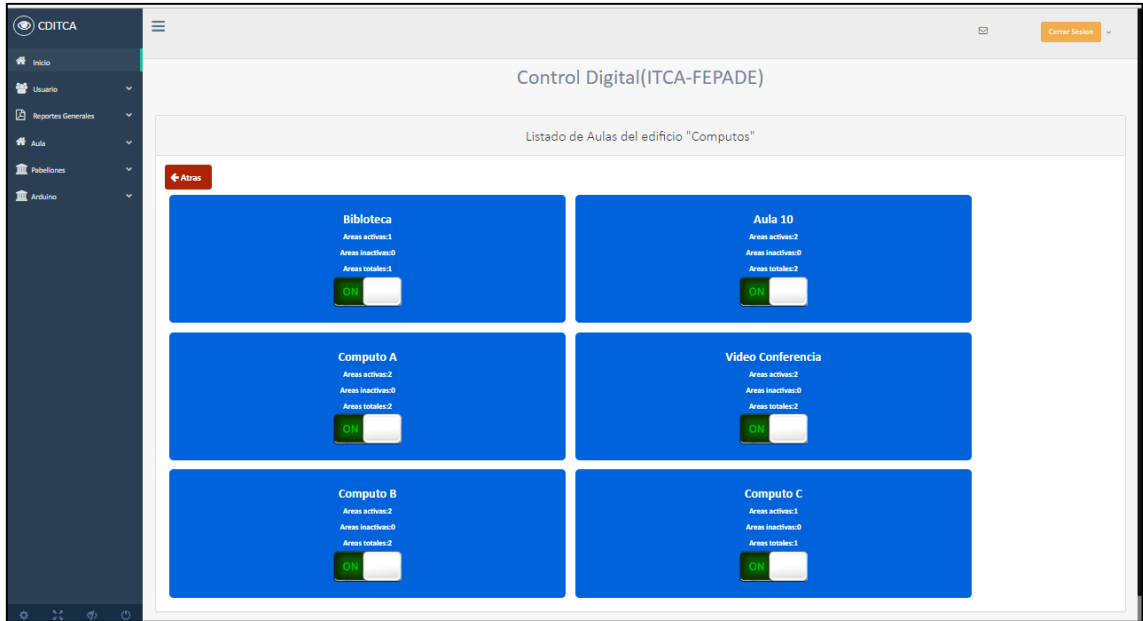


Imagen 3. Pantalla de administración general de salones

Este lugar del sistema consiste en la administración general de cada salón categorizadas por edificio académico perteneciente al ITCA de la sede regional de Sanmiguel, encendiendo y apagando los interruptores de cada salón automáticamente.

La información de los salones es obtenida de la tabla "aulas" de la base de datos



Imagen 4. Pantalla de administración individual de pines por salón

La administración de salones consiste en encender y apagar los interruptores individualmente de cada salón de clases desde este panel, se muestran todos los interruptores por salón, el estado de estos y la opción de apagarlos y encenderlos. La información de los salones se obtiene de la tabla "pines" de la base de datos.



Imagen 5. Pantalla de registro de nuevo usuario

Las funciones de forma generalizada de este sector es agregar nuevas credenciales de acceso al sistema asociadas a un usuario. Esta información es almacenada en la tabla “usuarios” de la base de datos.



Imagen 6. Pantalla de administración de usuarios

Este sector del sistema se encarga de administrar los usuarios que tienen acceso al sistema, a partir de este panel de administración podemos activar o desactivar los usuarios, de la misma forma que se pueden editar sus datos y observar el estado e información de cada uno. Estos datos son obtenidos de la tabla “usuarios”.

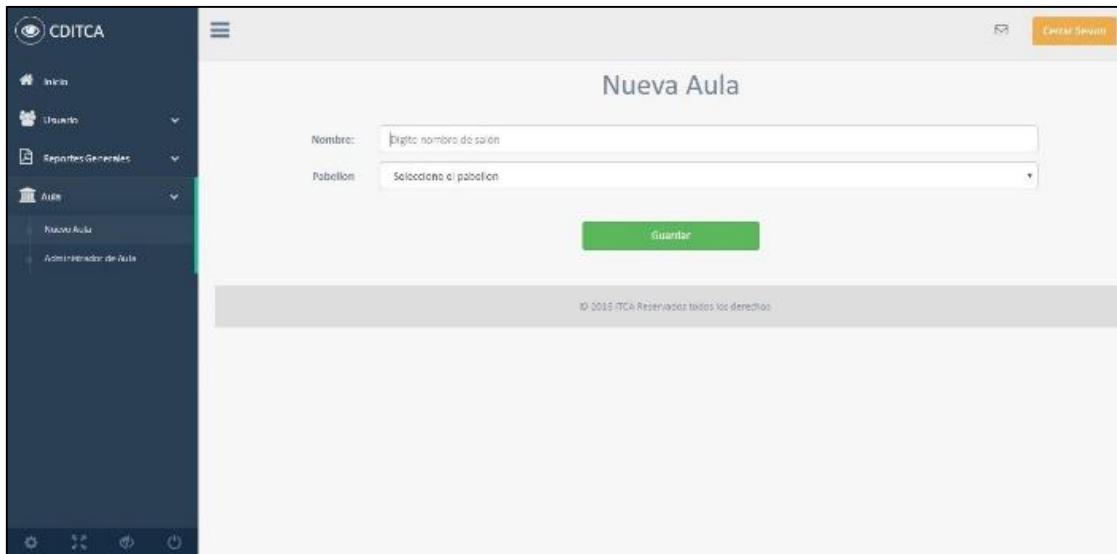


Imagen7. Pantalla para agregar nueva aula

La función generalizada de esta pantalla es ingresar nuevas aulas al sistema, la información ingresada es almacenada en la tabla “aulas” de la base de datos y estas son asociadas a un pabellón, información la cual es obtenida de la tabla “pabellones” de la base de datos.



Imagen 8. Pantalla para control de aulas

La administración de la información y disponibilidad de los salones de cada edificio es un hecho desde este panel desde administración, con las facilidades antes mencionadas anexadas la administración de cada uno de los pines de cada salón de clases, la información es obtenida de la tabla “aulas” de la base de datos.

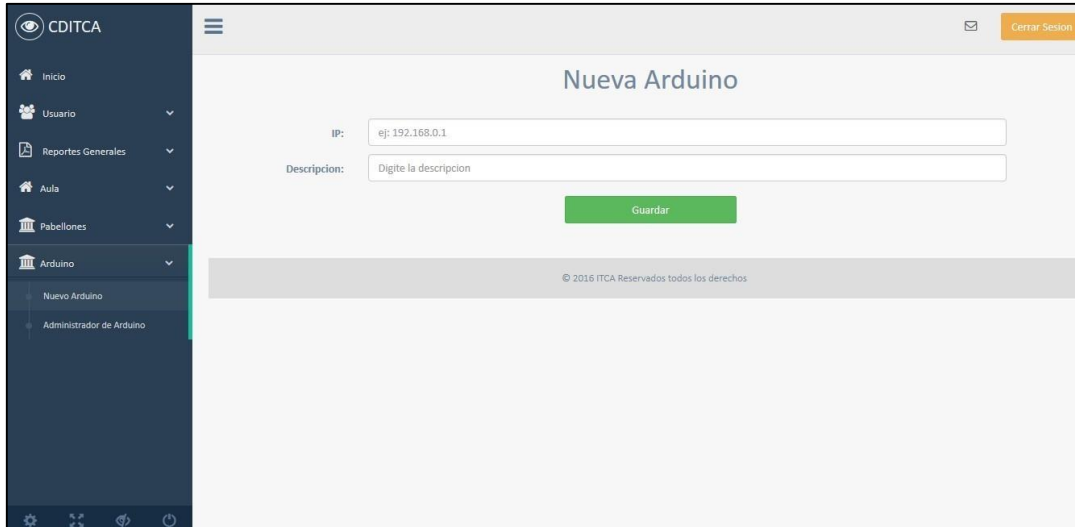


Imagen 9. Pantalla para agregar nuevo Arduino

Al implementar el proyecto de domótica a otra área con más placas de Arduino, es necesario tenerlos registrados en el sistema para hacer uso de estos, lo cual es posible desde este panel, la información ingresada se almacena en la tabla “Arduino” de la base de datos.



Imagen 10. Pantalla de administración de Arduino

La información de cada placa de Arduino se encuentra en este sector del sistema, el cual no defiriendo del resto de paneles de administración permite modificar la información y habilitar o deshabilitar las placas de Arduino para el manejo de estas en el sistema. La información de este panel de administración es obtenida de la tabla “Arduino” de la base de datos.

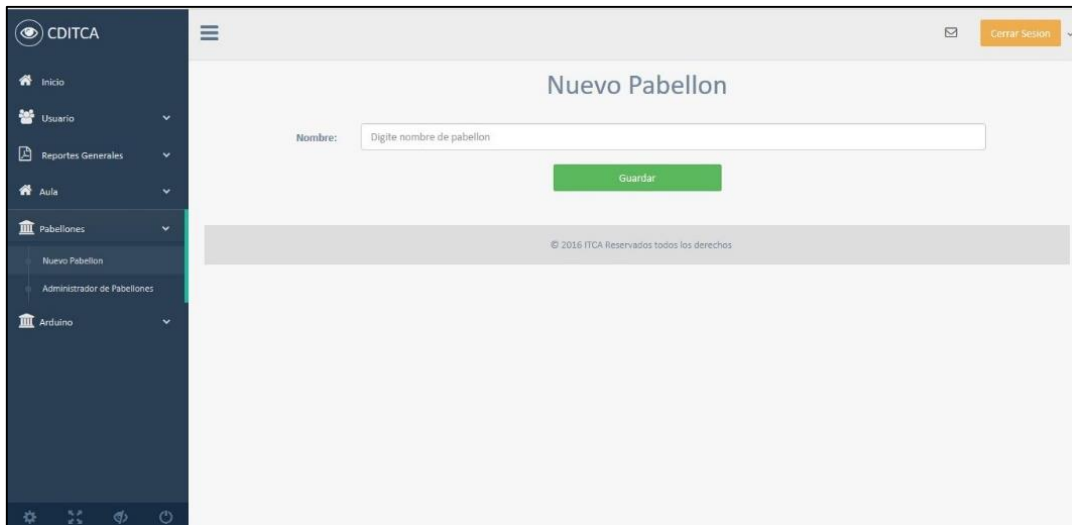


Imagen 11: Pantalla para agregar nuevo pabellón

Si deseas agregar más edificios académicos al rango de control del sistema esta es la zona idónea para el trabajo, con un simple dato puedes agregar un nuevo pabellón. La información ingresada se almacena en la tabla “pabellones” de la base de datos.



Imagen 12. Pantalla de administración de pabellones

Este es el panel de administración de los pabellones, el cual permite controlar los pabellones activos e inactivos, modificar la información de estos y mostrarla a la vez, recoge y modifica la información de la tabla “pabellones” de la base de datos.



Imagen 13. Pantalla de administración de pines

La administración de pines es una de las funciones que se puede realizar con los Arduinos instalados en el sistema, podemos intercambiar pines, agregar o eliminarlos.

7.1. IMPLEMENTACIÓN DEL SOFTWARE

Clases del Sistema Php

class/Aula.php

Descripción: Esta clase llamada “Aula” es la encargada de controlar los pines enlazados a un aula, y también puede controlarlos ya sea, encendiendo o apagando todos los pines de la misma, no importa si los pines están distribuidos en diferentes Arduinos, ella es capaz de controlarlos totalmente.

```
<?php

class Aula{

    private function getNumberOff($id){
        $link = conectar();
        $result = $link->query("select * from pin where IdAula = $id and Activo = 0") or
die($link->error);
        return $result->num_rows;
    }
    private function getNumberOn($id){
        $link = conectar();
```

```

    $result = $link->query("select * from pin where IdAula = $id and Activo = 1") or
die($link->error);
    return $result->num_rows;
}

public function getStateAll($pab = "%",$ard = "%"){
    $objeto= array();
    $link= conectar();
    $query="select IdAula as currentAula,Lugar from aula where IdAula like '$ard' and
IdPabellon like '$pab'";

    $result=$link->query($query) or die($link->error);
    if($result->num_rows>0){
        $i=0;
        while($row=$result->fetch_array(2)){
            $i++;

            $temp= new Arduino();
            $temp->id = $row[0];
            $temp->lugar = $row[1];
            $temp->ip ="";
            $temp->tPines = $this->getNumberOff($row[0])+$this->getNumberOn($row[0]);
            $temp->pinesOn = $this->getNumberOn($row[0]);
            $temp->pinesOff = $this->getNumberOff($row[0]);

            $temp2=array();
            $pinesQ="select Numero,Descripcion,Activo,IdPin from pin where IdAula =
$row[0]";
            $rPines=$link->query($pinesQ);
            if($rPines->num_rows>0){
                $pines= array();
                while($pin=$rPines->fetch_array(2)){
                    $pines["id"] = $pin[3];
                    $pines["n"] = $pin[0];
                    $pines["Desc"] = $pin[1];
                    $pines["Estado"] = $pin[2];

                    $temp2[]=$pines;
                }
            }
            $temp->pines=$temp2;
            $objeto[]=$temp;
        }
    }else{
        $temp= array();
        $temp["id"] = 0;
        $temp["lugar"] = "No hay resultados";
        $temp["ip"]="";
        $temp["tPines"] = 0;
    }
}

```

```

        $temp["pinesOn"] = 0;
        $temp["pinesOff"] = 0;
        $temp["error"]="No hay Resultados";
        $temp["pines"]=array();
        $objeto[] = $temp;
    }

    return $objeto;

}

public function setPeticionAllAula($id,$accion){
    //solicita el cambio de los pines de un Arduino completo
    $pines = array();
    $id = $id * 1;
    $accion = $accion * 1 ;
    $link = conectar();
    $result = $link->query("select IdPin from pin where IdAula = $id") or die($link->error);
    $retorno = array("hecho"=>false,"razon"=>"No completado");
    if($result->num_rows > 0){
        $i=0;
        while($row = $result->fetch_array(2)){
            $i++;

            $temp = new peticiones($link);

            $t = $temp->setNewPeticion($accion, $row[0]);

            if($t>0){

                $pines[] = $temp;
            }else{

                $retorno["razon"] = $t;
            }
        }

        $i = 0;
        $listo = false;

        while($listo === false ){
            // echo "esperando respuesta $i<br/>";
            $estados = array();
            $listo = true;
            $i++;
            // sleep(1);
            $j = 0;

```

```

foreach ($pines as $value) {
    //var_dump($value);

    if(!isset($estados[$j])){
        $estados[$j] = false;
    }
    if($estados[$j] == false){
        $t = $value->getEstate();

        if($t == "En Espera" or $t == "Procesando"){
            $estados[$j] = false;
        }
        elseif($t == "Exito"){
            $estados[$j] = true;
            $retorno["hecho"] = true;
            $retorno["razon"] = "Cambio exitoso!";
        }elseif($t == "Error"){
            $estados[$j] = true;
            $retorno["hecho"] = false;
            $retorno["razon"]="Error Interno";
        }else{
            $estados[$j] = true;
            $retorno["razon"]=$t;
        }
    }
    $j++;
}

$listo = true;
foreach ($estados as $value) {
    if($value === false){
        $listo = false;
    }
}

}
}else{
    $retorno["razon"]="Este Arduino no controla ningun pin";
    $retorno["hecho"] = false;
}

$link->close();

return $retorno;

```

```
}  
}
```

class/peticiones.php

Descripción: Quizá una de las clases más importantes del sistema PHP, esta es la que se encarga de crear instancias para una nueva petición, y monitorizar el estado para devolverle el resultado al usuario, también tiene programado su propio sistema de limpieza, por lo que, si una petición no es respondida, es eliminada y se devuelve la razón al usuario.

```
<?php  
  
class peticiones{  
    private $id;  
    public $IdPin;  
    private $creacion;  
    private $guardado;  
    private $idPin;  
    private $accion;  
    private $link;//se guarda la conexión entregada para evitar realizar demasiadas  
    conexiones durante este tiempo de peticiones  
  
    public function __construct($link = false,$accion = null, $id = null) {  
        // es usado cuando se quiere controlar una petición ya creada  
  
        $Hoy=new DateTime(null,new DateTimeZone('America/El_Salvador'));  
        $time = $Hoy->getTimestamp();  
  
        $this->accion = $accion;// que se desea realizar,  
        $this->creacion = $time;  
        $this->guardado = false;  
        $this->id = $id;//id petición  
        $this->idPin = ($id!==null)?$this->descubrirPin($id):null;  
        $this->link = $link;  
    }  
    private function conectar(){  
        $link = $this->link;  
        if($link === false or $link === null){  
            $link = conectar();//si no se mando una conexión, se crea una.  
        }  
        return $link;  
    }  
    private function descubrirPin($peticion){
```



```

    $retorno = false;
    $link = $this->conectar();
    $result = $link->query("select IdPin,Accion from peticion where IdPeticion = $peticion")
    or die($link->error);
    if($result->num_rows > 0){
        $row = $result->fetch_array(2);
        $retorno = $row[0];
        $this->accion = $row[1];
    }
    //$link->close();
    return $retorno;
}
public function setNewPeticion($accion,$idPin){

    //Se revisa si ya hay una peticion manejada en esta clase, si asi lo es, se suelta la
    anterior y se toma la nueva
    if($this->id !== null){
        $this->destruir();
    }

    if($this->validarPeticion($idPin)===false){
        return "Debe esperar al menos 15 segundos, para volver a modificar este
    dato";
    }

    $Hoy = new DateTime(null,new DateTimeZone('America/El_Salvador'));
    $time = $Hoy->getTimestamp();
    $this->creacion = $time;

    $idSesion = $_SESSION["id"];
    $link = $this->conectar();
    $link->query("insert                peticion(IdSesion,Fecha,Accion,IdPin)
    values($idSesion,$time,$accion,$idPin)") or die($link->error);
    if($link->affected_rows>0){
        $this->id = $link->insert_id;
        $this->idPin = $idPin;
        $this->accion = $accion;
        $this->IdPin = $idPin;

        return $this->id;
    }else{
        return false;
    }
}
private function validarPeticion($idPin){

    $Hoy = new DateTime(null,new DateTimeZone('America/El_Salvador'));
    $retorno = true;
    $time = $Hoy->getTimestamp();

```

```

        $link = $this->conectar();
        $query = "select * from peticion where Idpin = $idPin and ($time-Fecha) < 15";
        //echo $query."\n";
        $result = $link->query($query) or die($link->error);
        if($result->num_rows>0){
            $retorno = false;
        }
        //$link->close();
        return $retorno;
    }
    private function registrarCambio(){
        $retorno = false;
        $link = $this->conectar();
        $link->query("update pin set Activo = ".$this->accion." where IdPin = ".$this->idPin) or
die($link->error);
        if($link->affected_rows>0){
            $retorno = true;
        }

        //$link->close();
        return $retorno;
    }
    public function setEstado($estado){
        $link = $this->conectar();
        $link->query("update peticion set Estado = '$estado' where IdPeticion = ".$this->id);
        if($link->affected_rows>0){
            // $this->registrarCambio();
            return true;
        }else{
            return false;
        }
    }
    public function getEstate(){

        if($this->id === null){
            return false;
        }
        $Hoy = new DateTime(null,new DateTimeZone('America/El_Salvador'));

        $time = $Hoy->getTimestamp();

        $retorno = "No encontrado";
        $link = $this->conectar();
        $result = $link->query("select Estado from peticion where IdPeticion = ".$this-
>id) or die($link->error);
        if($result->num_rows>0){
            $row = $result->fetch_array(2);
            switch ($row[0]){
                case "N":

```

```

        $retorno = "En Espera";
        if(($time-$this->creacion)>15){
            $this->cancelar();
            $retorno = "Petición cancelada por sobrepasar Tiempo de espera";
        }
        break;
        case "P":
            $retorno = "Procesando";
            break;
        case "E":
            $retorno = "Exito";
            $this->registrarCambio();// se registra el cambio;
            break;
        case "Er":
            $retorno = "NO se puedo realizar";
            break;
        default:
            $retorno = "No definido";
            break;
        }
    }else{
        $retorno = "No encontrado";
    }

    //}$link->close();
    return $retorno;
}

public function cancelar(){

    //cancela una petición si no se proceso, SOLO si no se proceso
    $retorno = false;
    $link = $this->conectar();
    $link->query("delete from peticion where IdPeticion = ".$this->id." AND Estado
='N'") or die($link->error);
    if($link->affected_rows>0){
        $this->destruir();
        $retorno = true;
    }
    //}$link->close();
    return $retorno;
}

private function destruir(){
    //se destruyen los valores para evitar problemas de memoria
    $this->accion = null;
    $this->creacion = null;
    $this->guardado = false;
    $this->id = null;
    $this->idPin = null;
}

```

```

    $this->link = false;
}

public function getPeticiones($procesar = false){
    $peticiones = array();
    $link = $this->conectar();
    $result = $link->query("select IdPeticon as Pet,Arduino.IdArduino,Numero,Accion,Ip
from peticon join pin on peticon.IdPin = pin.IdPin join Arduino on Arduino.IdArduino =
pin.IdArduino where Estado = 'N' order by Fecha Desc") or die($link->error);
    if($result->num_rows>0){
        while($row = $result->fetch_array(2)){
            $peticiones[] =
array("peticon"=>$row[0],"Arduino"=>$row[1],"pin"=>$row[2],"accion"=>$row[3],"ip"=>$r
ow[4]);
            if($procesar === true){
                $temp = new peticiones($link, null, $row[0]);
                $temp->setEstado("E");
            }
        }
    }else{
        return false;
    }
    $link->close();
    return $peticiones;
}
}

```

Clases de la App Android

phpComunicacion.java
<p>Descripción: Esta es la principal clase de la App ya que permite la conexión al sistema php del sistema de control digital.</p>
<pre> package itca.fepade.itcacontroldigital; import android.content.Context; import android.net.ConnectivityManager; import android.net.NetworkInfo; import android.net.Uri; import java.io.BufferedReader; import java.io.BufferedWriter; import java.io.IOException; import java.io.InputStream; import java.io.InputStreamReader; import java.io.OutputStream; </pre>

```

import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class phpCommunication {
    private String url;
    public static final int CONNECTION_TIMEOUT = 10000;
    public static final int READ_TIMEOUT = 25000;
    public phpCommunication(String url){

        this.url = url;
        //Toast.makeText(getApplicationContext(),"hola",Toast.LENGTH_LONG).show();
    }
    private String solicitud(Uri.Builder Parametros) {
        HttpURLConnection conn;
        URL url = null;

        try {

            // Enter URL address where your php file resides
            url = new URL(this.url);

        } catch (MalformedURLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            return this.url;
        }
        try {
            // Setup HttpURLConnection class to send and receive data from php and mysql
            conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(READ_TIMEOUT);
            conn.setConnectTimeout(CONNECTION_TIMEOUT);
            conn.setRequestMethod("POST");

            // setDoInput and setDoOutput method depict handling of both send and receive
            conn.setDoInput(true);
            conn.setDoOutput(true);

            // Append parameters to URL
            Uri.Builder builder = Parametros;

            String query = builder.build().getEncodedQuery();

            // Open connection for sending data
            OutputStream os = conn.getOutputStream();
            BufferedWriter writer = new BufferedWriter(
                new OutputStreamWriter(os, "UTF-8"));

```

```

        writer.write(query);
        writer.flush();
        writer.close();
        os.close();
        conn.connect();

    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
        return "No se pudo contactar al servidor";
    }

    try {

        int response_code = conn.getResponseCode();

        // Check if successful connection made
        if (response_code == HttpURLConnection.HTTP_OK) {

            // Read data sent from server
            InputStream input = conn.getInputStream();
            BufferedReader reader = new BufferedReader(new InputStreamReader(input));
            StringBuilder result = new StringBuilder();
            String line;

            while ((line = reader.readLine()) != null) {
                result.append(line);
            }

            // Pass data to onPostExecute method
            return (result.toString());

        } else {

            return ("Servidor Incorrecto!");
        }

    } catch (IOException e) {
        e.printStackTrace();
        return "Servidor Incorrecto!";
    } finally {
        conn.disconnect();
    }

}

public String login(String usuario, String password) {
    Uri.Builder builder = new Uri.Builder()
        .appendQueryParameter("usuario", usuario)

```

```

        .appendQueryParam("password", password)
        .appendQueryParam("login", "true");
String resultado = solicitud(builder);
return resultado;
}

public String estadoArduinos(String idUsuario,String idPabellon) {
    Uri.Builder form = new Uri.Builder()
        .appendQueryParam("Arduinos", "true")
        .appendQueryParam("claveUser", idUsuario)
        .appendQueryParam("idPabellon", idPabellon);
String resultado = solicitud(form);
if (resultado.equals("")) {
    resultado = "[{"id\":"0\","error\":""+resultado+"","\lugar\":"ERROR DE
CONEXIÓN","ip\":"192.168.8.103\","tPines\":"0","\pinesOn\":"0\","pinesOff\":"0\","
pines\":[}]";
}
return resultado;
}

public String toggleArduino(String id,String estado,String idUsuario){
    Uri.Builder form = new Uri.Builder()
        .appendQueryParam("ArduinoToggle", estado)
        .appendQueryParam("id",id)
        .appendQueryParam("claveUser", idUsuario);
String resultado = solicitud(form);
if (resultado.equals("")) {
    // resultado = "{\"hecho\":false,\"action\":"0\","idArduino\":"2,\"claveUser\":"0\"}";
}
return resultado;
}
public String togglePin(String id,String estado,String idUsuario){
    Uri.Builder form = new Uri.Builder()
        .appendQueryParam("pinToggle", estado)
        .appendQueryParam("id",id)
        .appendQueryParam("claveUser", idUsuario);
String resultado = solicitud(form);
if (resultado.equals("")) {
    // resultado = "{\"hecho\":false,\"action\":"0\","idArduino\":"2,\"claveUser\":"0\"}";
}
return resultado;
}
public String listPabellones(String idUsuario){
    Uri.Builder form = new Uri.Builder()
        .appendQueryParam("listaPabellones", "true")

```

```

        .appendQueryParameter("claveUser",idUsuario);

String resultado = solicitud(form);
if (resultado.equals("")) {
    resultado
}
return resultado;
}
public String internetTest(){
    Uri.Builder form = new Uri.Builder()
        .appendQueryParameter("iTest", "true");
String resultado = solicitud(form);
return resultado;
}
}

```

Sketch Arduino

Arduino.ino

Descripción: este sketch es el firmware para el Arduino, cabe aclarar que debido a la forma de trabajo, el sketch es el mismo para todos los Arduinos requeridos

```

#include <Adafruit_CC3000.h>
#include <SPI.h>
#include "utility/debug.h"
#include "utility/socket.h"

#define ADAFRUIT_CC3000_IRQ 3 // pin utilizado para resetear Arduino
// Se eligen los pines establecidos para esta función, ya esta definida por la shield
#define ADAFRUIT_CC3000_VBAT 5
#define ADAFRUIT_CC3000_CS 10
// Use hardware SPI for the remaining pins
// On an UNO, SCK = 13, MISO = 12, and MOSI = 11

Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT,
SPI_CLOCK_DIVIDER); // you can change this clock speed

#define WLAN_SSID "Arduino_ITCA" // nombre de la conexión WIFI
#define WLAN_PASS "Arduino1" // contraseña de la red WIFI

// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or
WLAN_SEC_WPA2

```



```

#define WLAN_SECURITY WLAN_SEC_WPA2 //Metodo de seguridad, anteriormente
nombrados

#define LISTEN_PORT 80 // Puerto de escucha para el servidor
// El protocolo HTTP usa por defecto el puerto 80

#define MAX_ACTION 10 // Maximum length of the HTTP action that can be parsed.

#define MAX_PATH 180 // Maximum length of the HTTP request path that can
be parsed.
// There isn't much memory available so keep this short!

#define BUFFER_SIZE MAX_ACTION + MAX_PATH + 20 // Size of buffer for incoming
request data.
// Since only the first line is parsed this
// needs to be as large as the maximum action
// and path plus a little for whitespace and
// HTTP version.

#define TIMEOUT_MS 500 // Amount of time in milliseconds to wait for
// an incoming request to finish. Don't set this
// too high or your server could be slow to respond.

Adafruit_CC3000_Server httpServer(LISTEN_PORT);
uint8_t buffer[BUFFER_SIZE+1];
int bufindex = 0;
char action[MAX_ACTION+1];
char path[MAX_PATH+1];
// Estas variables son usadas para leer las peticiones recibidas en formato JSON, se ha
omitido la " debido a que al mandarlas por la URL se convierte en caracteres raros
String NUMBER = "{number:";
String STATE = ",state:";
String END = "}";

void setup(void)
{
//Se establecen como pines de salida el pin digital 8 para control de refrigeración y los
demás para los diferentes circuitos de potencia
pinMode(8,OUTPUT);
pinMode(14,OUTPUT);
pinMode(15,OUTPUT);
pinMode(16,OUTPUT);
pinMode(17,OUTPUT);
pinMode(18,OUTPUT);
pinMode(19,OUTPUT);

Serial.begin(9600);
Serial.println(F("Hello, CC3000!\n"));

```

```

Serial.print("Free RAM: "); Serial.println(getFreeRam(), DEC);

// Initialise the module
Serial.println(F("\nInitializing..."));
if (!cc3000.begin())
{
  Serial.println(F("Couldn't begin! Check your wiring?"));
  while(1);
}

Serial.print(F("\nConectandose a :")); Serial.println(WLAN_SSID);
if (!cc3000.connectToAP(WLAN_SSID, WLAN_PASS, WLAN_SECURITY)) {
  Serial.println(F("Falló!"));
  while(1);
}

Serial.println(F("Conectado!"));

Serial.println(F("Request DHCP"));
while (!cc3000.checkDHCP())
{
  delay(100); // ToDo: Insert a DHCP timeout!
}

// Display the IP address DNS, Gateway, etc.
while (!displayConnectionDetails()) {
  delay(1000);
}

// Start listening for connections
httpServer.begin();

Serial.println(F("Escuchando a los clientes HTTP"));
}

void loop(void)
{
  // Try to get a client which is connected.
  Adafruit_CC3000_ClientRef client = httpServer.available();
  if (client) {
    Serial.println(F("Client connected.));
    // Process this request until it completes or times out.
    // Note that this is explicitly limited to handling one request at a time!

    // Clear the incoming data buffer and point to the beginning of it.
    bufindex = 0;
    memset(&buffer, 0, sizeof(buffer));

    // Clear action and path strings.

```

```

memset(&action, 0, sizeof(action));
memset(&path, 0, sizeof(path));

// Set a timeout for reading all the incoming data.
unsigned long endtime = millis() + TIMEOUT_MS;

// Read all the incoming data until it can be parsed or the timeout expires.
bool parsed = false;
while (!parsed && (millis() < endtime) && (bufindex < BUFFER_SIZE)) {
  if (client.available()) {
    buffer[bufindex++] = client.read();
  }
  parsed = parseRequest(buffer, bufindex, action, path);
}

// Handle the request if it was parsed.
if (parsed) {
  Serial.println(F("Processing request"));
  Serial.print(F("Action: ")); Serial.println(action);
  Serial.print(F("Path: ")); Serial.println(path);
  // Check the action to see if it was a GET request.
  if (strcmp(action, "GET") == 0) {
    // Respond with the path that was accessed.
    // First send the success response code.
    client.fastrprintln(F("HTTP/1.1 200 OK"));
    // Then send a few headers to identify the type of data returned and that
    // the connection will not be held open.
    client.fastrprintln(F("Content-Type: text/plain"));
    client.fastrprintln(F("Connection: close"));
    client.fastrprintln(F("Server: Adafruit CC3000"));
    // Send an empty line to signal start of body.
    client.fastrprintln(F(""));
    // Now send the response data.
    //client.fastrprintln(F("Hello world!"));
    // client.fastrprint(F("You accessed path: ")); client.fastrprintln(path);
    String data = String(path);
    Serial.println(millis());
    int of = data.indexOf("[") + 1;
    int to = data.indexOf("]");

    if(of > -1 && to > -1){
      data = data.substring(of,to);
      //Serial.println("resultado de corte");
      //Serial.println(data);
      //Reiniciamos las variables
      of = -1;
      to = -1;
      for(int i = 0;i<data.length();i++){
        String letra = data.substring(i,i+1);

```

```

        ///Serial.print("Letra: ");///Serial.println(letra);
        if(letra.equals("{"){
            of = i;
        }else if(letra.equals("}"){
            to = i;
        }
        if(of>-1 && to>-1){
            process(data.substring(of,to+1));
            of = -1;
            to = -1;
        }

    }
    client.fastrprintln("done");
}else{
    ///Serial.println("No hay peticiones");
}

}
else {
    // Unsupported action, respond with an HTTP 405 method not allowed error.
    client.fastrprintln(F("HTTP/1.1 405 Method Not Allowed"));
    client.fastrprintln(F(""));
}
}

// Wait a short period to make sure the response had time to send before
// the connection is closed (the CC3000 sends data asynchronously).
delay(100);

// Close the connection when done.
Serial.println(F("Client disconnected"));
client.close();
}
}

bool parseRequest(uint8_t* buf, int bufSize, char* action, char* path) {
    // Check if the request ends with \r\n to signal end of first line.
    if (bufSize < 2)
        return false;
    if (buf[bufSize-2] == '\r' && buf[bufSize-1] == '\n') {
        parseFirstLine((char*)buf, action, path);
        return true;
    }
    return false;
}

// Parse the action and path from the first line of an HTTP request.

```

```

void parseFirstLine(char* line, char* action, char* path) {
    // Parse first word up to whitespace as action.
    char* lineaction = strtok(line, " ");
    if (lineaction != NULL)
        strncpy(action, lineaction, MAX_ACTION);
    // Parse second word up to whitespace as path.
    char* linepath = strtok(NULL, " ");
    if (linepath != NULL)
        strncpy(path, linepath, MAX_PATH);
}

// Tries to read the IP address and other connection details
bool displayConnectionDetails(void)
{
    uint32_t ipAddress, netmask, gateway, dhcpserv, dnsserv;

    if(!cc3000.getIPAddress(&ipAddress, &netmask, &gateway, &dhcpserv, &dnsserv))
    {
        Serial.println(F("Unable to retrieve the IP Address!\r\n"));
        return false;
    }
    else
    {
        Serial.print(F("\nIP Addr: ")); cc3000.printIPdotsRev(ipAddress);
        Serial.print(F("\nNetmask: ")); cc3000.printIPdotsRev(netmask);
        Serial.print(F("\nGateway: ")); cc3000.printIPdotsRev(gateway);
        Serial.print(F("\nDHCPsrv: ")); cc3000.printIPdotsRev(dhcpserv);
        Serial.print(F("\nDNSserv: ")); cc3000.printIPdotsRev(dnsserv);
        Serial.println();
        return true;
    }
}

void process(String p){
    //Serial.print("valor: ");//Serial.println(p);
    int numLen = NUMBER.length();
    int staLen = STATE.length();
    int endPos = p.indexOf(END);
    int n= p.indexOf(STATE);
    int numero = p.substring(numLen,n).toInt();
    int state = p.substring(n+staLen,endPos).toInt();

    if(numero>0){
        digitalWrite(numero,state);
        Serial.print("Numero:");Serial.print(numero);Serial.print("
Estado");Serial.println(state);
        Serial.println(millis());
    }
}

```

```
}
```

Ciente en JAVA

ServerSync.java

Descripción: Esta clase es la principal del programa, es la que se encarga de leer el estado de todos los Arduinos y de leer todas las peticiones que se cuelgan en el sistema PHP.

```
package drivers;

import util.FileHandler;
import util.Log;
import java.util.Timer;
import java.util.TimerTask;
import servers.*;
import org.json.*;

import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
//import util.PeticionHandler;

/**
 *
 * @author marvin
 */
public class ServerSync {
    private static final int DELAY = 20;
    private static final int INTERVAL = 1000;
    private static boolean readingHttp = false;

    private boolean isFirstRead = true;
    private ArrayList<ArduinoHandler> Arduinos;
    private FileHandler db = new FileHandler("db",FileHandler.JSON);
    private static final ServerSync instance = new ServerSync();
    private long times = 0;
    private boolean waitingForFirstLecture = false;
    private long reads = 0;
    private boolean fanAreRunning = false;
    private ServerSync(){

        this.Arduinos = new ArrayList<>();//se crea un array de objetos Arduino
        this.processResponse(this.db.getData());//se inicializan los Arduinos con la base de
datos interna
        this.isFirstRead = true;// se establece que se har i la primer lectura
```

```

try {
    Thread.sleep(1000);
} catch (InterruptedException ex) {
    Logger.getLogger(ServerSync.class.getName()).log(Level.SEVERE, null, ex);
}
new Handler();//arranque el bucle infinito de lecturas al servidor
}
public static ServerSync getInstance(){
    return instance;
}

private void processResponse(String response){
    //Aca se procesan las respuestas del servidor
    try{
        JSONArray data = new JSONArray(response);
        if(this.isFirstRead){
            Log.d("Realizando primera lectura",true);
            this.saveDataBase(data);//Se guarda la base de datos con toda la informaci3n
recibida
        }else{
            Log.d("Procesando peticiones [" + this.times + "]");
            this.processPeticones(data);//Solo se procesan las peticiones necesarias
        }

    }catch(Exception ex){

        this.reads++;
        if((this.reads % 50) == 0){
            this.reSendArduinos();
        }
        if(this.reads > 100000){
            this.reads = 0;
        }
    }
}
private void reSendArduinos(){
    instance.saveDatabaseAgain();
    for(int i = 0;i<this.Arduinos.size();i++){
        this.Arduinos.get(i).sendStatusToArduino();
    }
}
private void saveDataBase(JSONArray data){
    this.Arduinos = new ArrayList<>();
    db.reCreateFile(data.toString());
    for(int i = 0;i<data.length();i++){

```

```

        ArduinoHandler Arduino = new ArduinoHandler(data.getJSONObject(i));
        Arduino.sendStatusToArduino();
        if(!this.fanAreRunning){
            new ArduinoFan(Arduino.getIdArduino());
        }
        this.Arduinos.add(Arduino);
    }
    this.fanAreRunning = true;
    this.isFirstRead = false;
}
private void saveDatabaseAgain(){
    JSONArray array = new JSONArray();
    for(int i = 0;i<this.Arduinos.size();i++){
        array.put(this.Arduinos.get(i).toObject());
    }
    String data = array.toString().replace("\\", "");
    Log.d("Regerando base de datos...");
    db.reCreateFile(data);
    Log.d("La base de datos ha sido guardada exitosamente",true);
}
private void processPeticones(JSONArray data){
//    PeticionHandler.getInstance().startLisening(this.Arduinos.size());

    for(int i = 0;i<this.Arduinos.size();i++){
        this.Arduinos.get(i).processPeticones(data);
    }
}
public void changePinState(int idArduino, int numberPin, int state){
    for(int i = 0;i<this.Arduinos.size();i++){

        if(this.Arduinos.get(i).getIdArduino() == idArduino){
            this.Arduinos.get(i).changePinState(state);
            break;
        }
    }
}

private class Handler {

    public Handler(){
        this.run();
    }
    private void run() {
        Timer timer;
        timer = new Timer();
        //WebServer server = new WebServer();

```



```

TimerTask task = new TimerTask() {
    @Override
    public void run()
    {
        if(!readingHttp){
            readingHttp = true;
            String get = (isFirstRead)?"readArduinos":"readPeticones";
            processResponse(WebServer.sendRequest(Settings.GetInstance().getUrl() +
"?action=" + get));
            readingHttp = false;
        }
    }
};

timer.schedule(task, DELAY, INTERVAL);
}
}

```

ArduinoHandler.java

Descripción: esta clase trabaja de la mano con la anterior y es la encargada de controlar cada uno de los pines físicos del Arduino correspondiente esto a través de las peticiones HTTP realizadas a dichos Arduinos.

```

package drivers;

import java.util.ArrayList;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.json.JSONArray;
import org.json.JSONObject;
import servers.WebServer;
import util.Log;
//import util.PeticionHandler;

/**
 *
 * @author marvin
 */
public class ArduinoHandler {
    private int id;
    private String lugar;
    private String ip;
    private int tPines;

```

```

private int pinesOn;
private int pinesOff;
private ArrayList<Pin> pines;
//private ArduinoFan ArduinoFan;
private boolean sendingData;

public ArduinoHandler(JSONObject data) {
    this.id = data.getInt("id");
    this.lugar = data.getString("lugar");
    this.ip = data.getString("ip");
    this.tPines = data.getInt("tPines");
    this.pinesOn = data.getInt("pinesOn");
    this.pinesOff = data.getInt("pinesOff");
    this.pines = new ArrayList<>();

    JSONArray temp = data.getJSONArray("pines");
    for(int i = 0;i<temp.length();i++){
        Pin pin = new Pin(temp.getJSONObject(i));
        pines.add(pin);
    }
    JSONObject pinObject = new JSONObject();
    pinObject.put("id", 0);
    pinObject.put("n",8);
    pinObject.put("Desc", "Pin de control de ventilaci3n");
    pinObject.put("Estado",1);
    Pin fan = new Pin(pinObject);
    pines.add(fan);

}
public JSONObject toObject(){
    JSONObject object = new JSONObject();
    object.put("id", this.id);
    object.put("lugar", this.lugar);
    object.put("ip", this.ip);
    object.put("tPines", this.tPines);
    object.put("pinesOn", this.pinesOn);
    object.put("pinesOff", this.pinesOff);
    JSONArray array = new JSONArray();
    for(int i = 0;i<this.pines.size();i++){
        array.put(this.pines.get(i).toOriginalObject());
    }
    object.put("pines",array);

    return object;
}

public void changePinState(int state){
    for(int i = 0;i<this.pines.size();i++){

```

```

        if(this.pines.get(i).numero == 8){
            this.pines.get(i).estado = state;

            break;
        }
    }
    this.sendStatusToArduino();
}
public void processPeticiones(JSONArray peticiones){
    boolean iNeedSendData = false;
    for(int i = 0;i<peticiones.length();i++){
        JSONObject peticion = peticiones.getJSONObject(i);
        if(peticion.getInt("Arduino") == this.id){
            iNeedSendData = this.processPeticion(peticion);
            break;
        }
    }
    if(iNeedSendData){
        this.sendStatusToArduino();
    }else{
        Log.d("El Arduino " + this.id + " No recibí cambios");
        // PeticionHandler.getInstance().setFinished(getIdArduino());//Se notifica que ya se
        // terminé
    }

}

}
private boolean processPeticion(JSONObject peticion){
    boolean done = false;
    Log.d("Procesando petición:" + peticion.toString());
    for(int i = 0;i<this.pines.size();i++){
        if(this.pines.get(i).numero == peticion.getInt("pin")){
            this.pines.get(i).estado = peticion.getInt("accion");
            Log.d("[Arduino " + this.id + "]Se cambió estado del pin " + this.pines.get(i).numero
+ " a " + (this.pines.get(i).estado == 1?"Encendido":"Apagado"));
            done = true;
        }
    }
    return done;
}
public int getIdArduino() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

```

```

public String getLugar() {
    return lugar;
}

public void setLugar(String lugar) {
    this.lugar = lugar;
}

public String getIp() {
    return ip;
}
public void setIp(String ip){
    this.ip = ip;
}

public int gettPines() {
    return tPines;
}

public void settPines(int tPines) {
    this.tPines = tPines;
}

public int getPinesOn() {
    return pinesOn;
}

public void setPinesOn(int pinesOn) {
    this.pinesOn = pinesOn;
}

public int getPinesOff() {
    return pinesOff;
}

public void setPinesOff(int pinesOff) {
    this.pinesOff = pinesOff;
}

private JSONArray generateGet(){
    JSONArray pines = new JSONArray();
    for(int i = 0;i<this.pines.size();i++){
        pines.put(this.pines.get(i).getObject());
    }
    return pines;
}
public void sendStatusToArduino(){

```

```

// while(this.sendingData){
//     try {
//         Log.err("Esperando que Arduino " + this.id + " termine para empezar de nuevo");
//         Thread.sleep(100);
//     } catch (InterruptedException ex) {
//         Logger.getLogger(ArduinoHandler.class.getName()).log(Level.SEVERE, null, ex);
//     }
// }
new ArduinoGetSender().start();

}

public boolean isSendingData() {
    return sendingData;
}

public void setSendingData(boolean sendingData) {
    this.sendingData = sendingData;
}

private class ArduinoGetSender extends Thread{
    private void send(String response){
        if(response.contains("done")){
            Log.d("El Arduino " + getIdArduino() + " realizÃ³ el proceso exitosamente");
        }else{
            Log.err("El Arduino " + getIdArduino() + " fallÃ³ al procesar la peticiÃ³n");
            Log.err("Respuesta del Arduino:\n" +response + "\n");
        }
    }
}

@Override
public void run() {
    setSendingData(true);
    Log.d("Enviando solicitud a Arduino" + getIdArduino());
    JSONArray pines = generateGet();
    int c = 0;
    int ls = 0;
    JSONArray temp = new JSONArray();
    for(int i = 0;i<pines.length();i++){
        temp.put(pines.getJSONObject(i));
        c++;
        if((c % 3) == 0){
            ls = i;
            String token = "http://" + getIp() + "?values=" + temp.toString().replace("\n",
""");
            this.send(WebServer.sendRequest(token));
            temp = new JSONArray();

```

```

    }
    }
    temp = new JSONArray();
    for(int i = ls+1;i<pines.length();i++){
        temp.put(pines.getJSONObject(i));
    }
    if(temp.length(>0){
        String token = "http://" + getIp() + "?values=" + temp.toString().replace("\\\"", "");
        this.send(WebServer.sendRequest(token));
    }
    // PeticionHandler.getInstance().setFinished(getIdArduino());//Se notifica que ya se
termina
    try {
        Thread.sleep(1500);//Espera para no saturar el Arduino
    } catch (InterruptedException ex) {
        Logger.getLogger(ArduinoHandler.class.getName()).log(Level.SEVERE, null, ex);
    }
    setSendingData(false);
}

private class Pin{
    private int id;
    private int numero;
    private String descripcion;
    private int estado;
    public Pin(JSONObject object){
        this.id = object.getInt("id");
        this.numero = object.getInt("n");
        this.descripcion = object.getString("Desc");
        this.estado = object.getInt("Estado");
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }
}

```

```

public String getDescripcion() {
    return descripcion;
}

public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}

public int getEstado() {
    return estado;
}

public void setEstado(int estado) {
    this.estado = estado;
}

public void setOff(){
    this.estado = 0;
}
public void setOn(){
    this.estado = 1;
}

public JSONObject toOriginalObject(){
    JSONObject object = new JSONObject();
    object.put("id", this.id);
    object.put("n", this.numero);
    object.put("Desc", this.descripcion);
    object.put("Estado", this.estado);
    return object;
}
public JSONObject getObject(){
    JSONObject data = new JSONObject();
    data.put("number", this.numero);
    data.put("state", this.estado);
    return data;
}
}
}
}

```

7.2. PRODUCTOS

Al finalizar el proceso de investigación del proyecto, se obtuvieron los siguientes productos:

1. Seccionamiento y documentación de las áreas en las que se divide la red eléctrica en ITCA-FEPADE Centro Regional San Miguel.
2. Control de mando inteligente, para la administración de la red eléctrica, por sección o área.
3. Aplicación multiplataforma para el control y administración de las secciones de la red eléctrica en ITCA-FEPADE Centro Regional San Miguel.
4. Integración del control eléctrico al Sistema de Préstamo de llaves.

7.3. DISEÑO DEL CIRCUITO IMPRESO

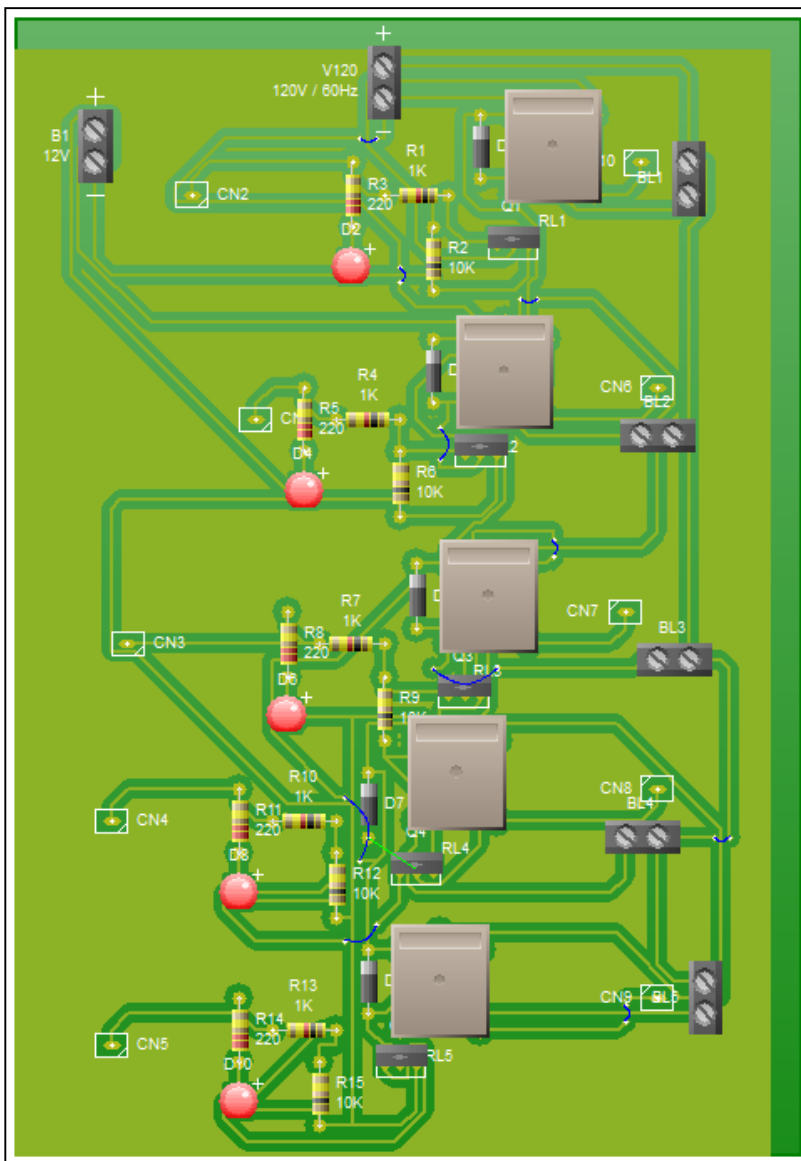


Figura 1. Placa de circuitos

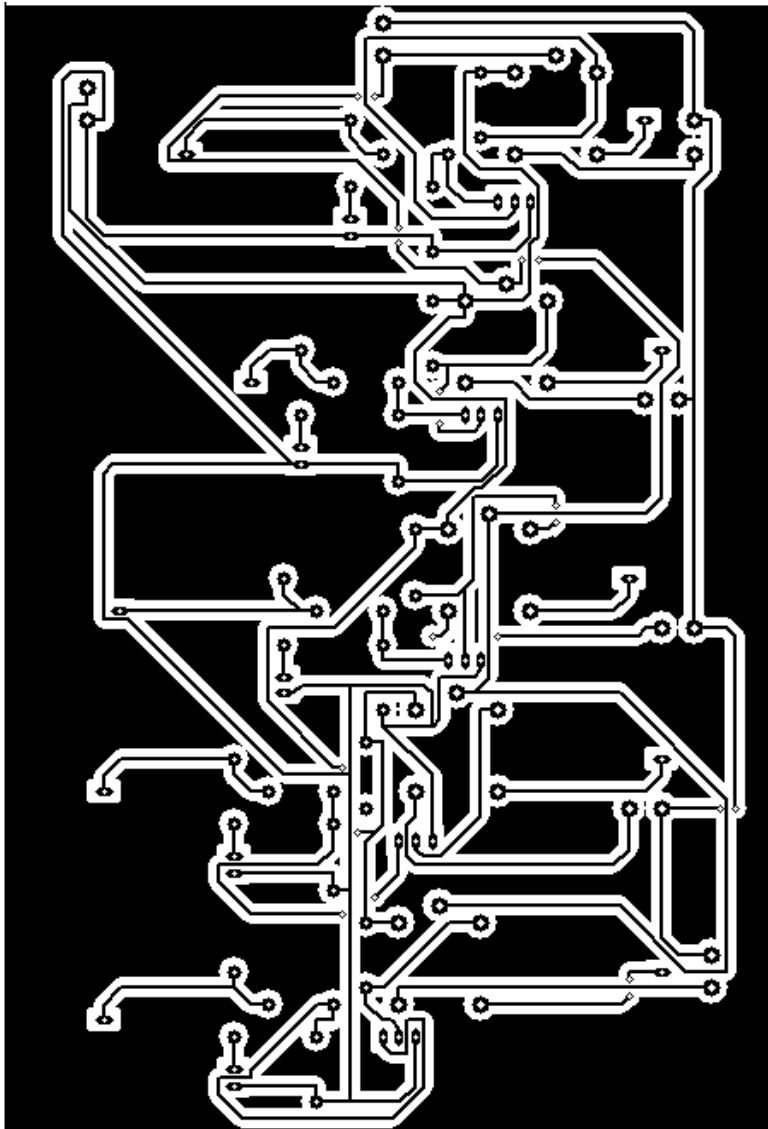
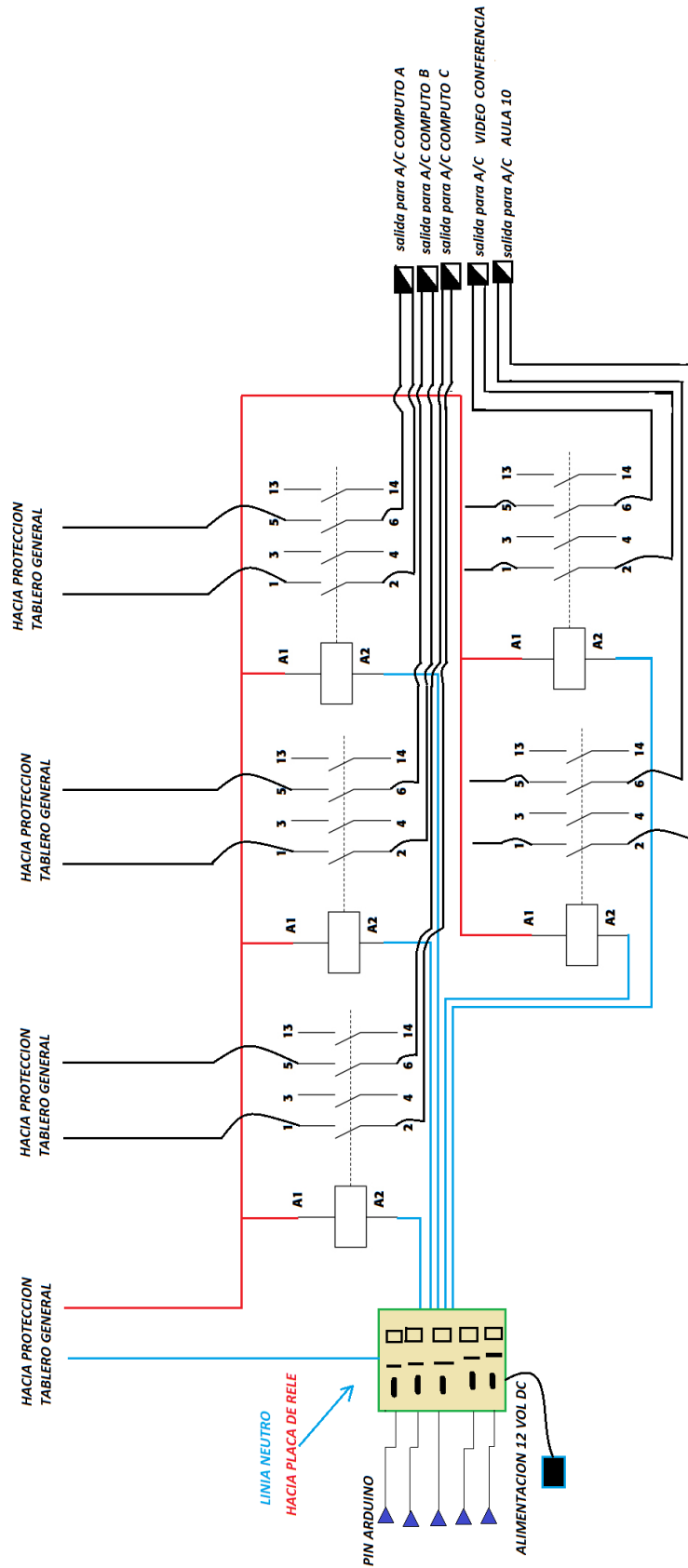


Figura 2. Placa PCB

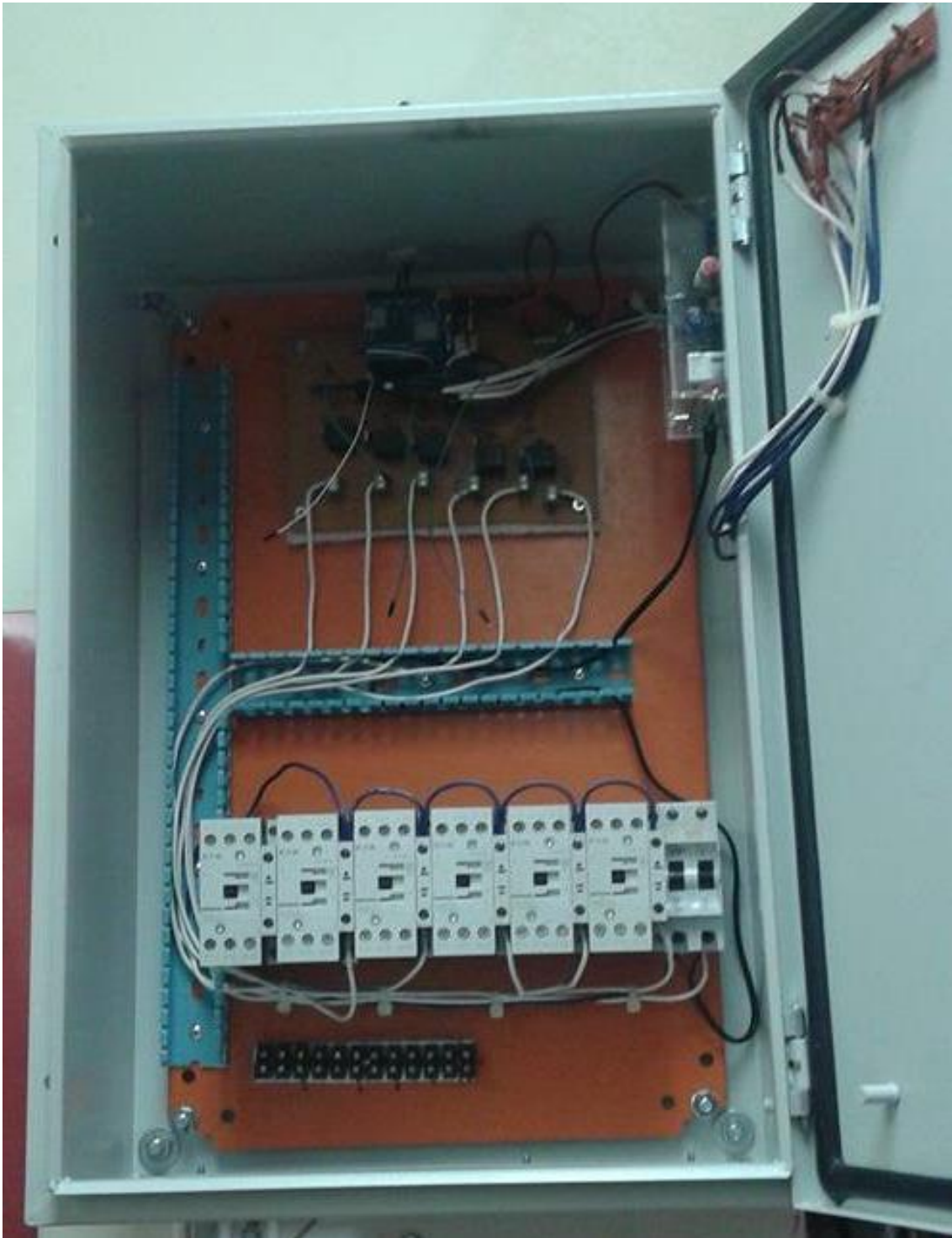
Lista de Componentes para PCB electrónica:

Nº	Nombre	Valor	Cantidad
1	Resisor	1K Ω	5
2	Resistor	220 Ω	5
3	Resistor	10K Ω	5
4	Diodo led	N Color	5
5	Relay	12 V	5
6	Bornera simple	2 Pines	7

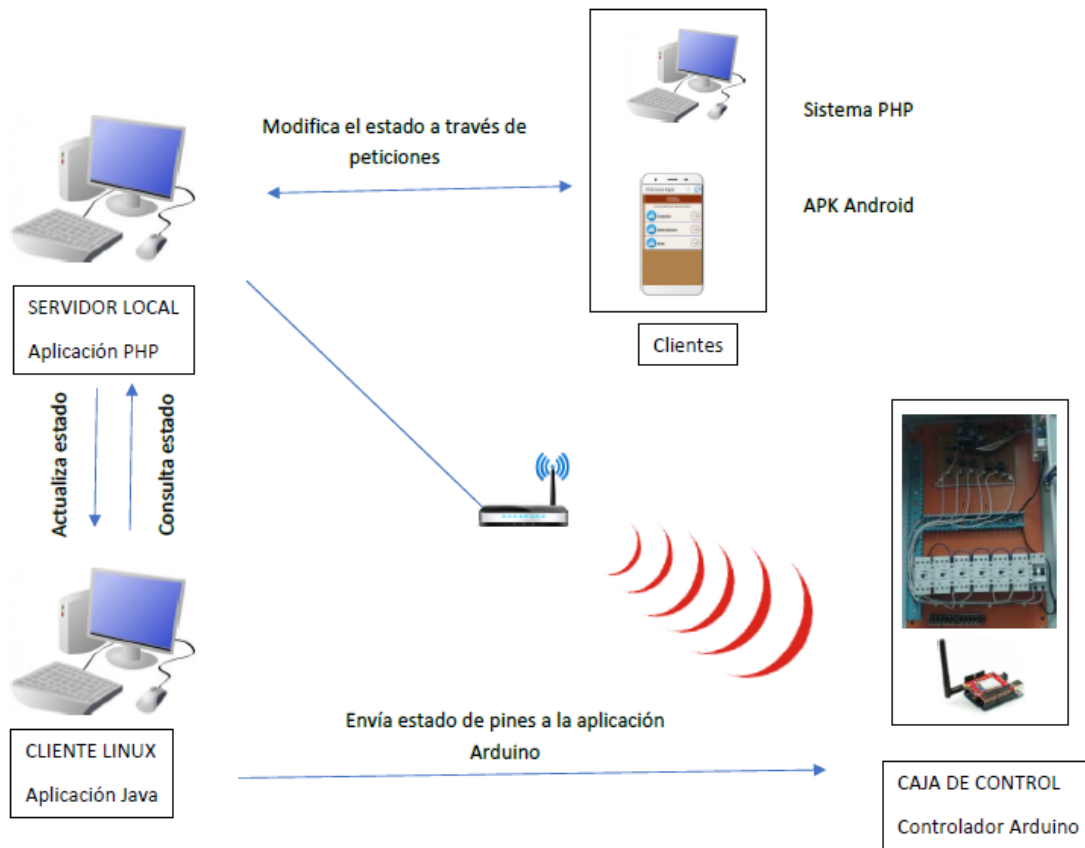
7.4. DIAGRAMA UNIFILAR



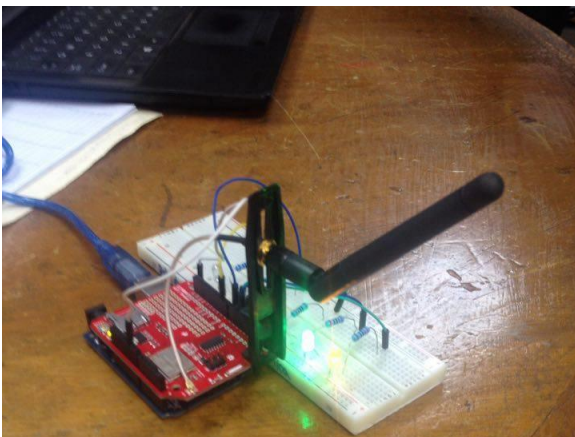
7.5. CAJA DE CONTROL CENTRAL



7.6. DIAGRAMA CONCEPTUAL DEL SISTEMA DIGITAL



7.7. IMÁGENES DEL PROCESO DE DESARROLLO





7.8.SEGMENTACIÓN DE LA RED ELÉCTRICA

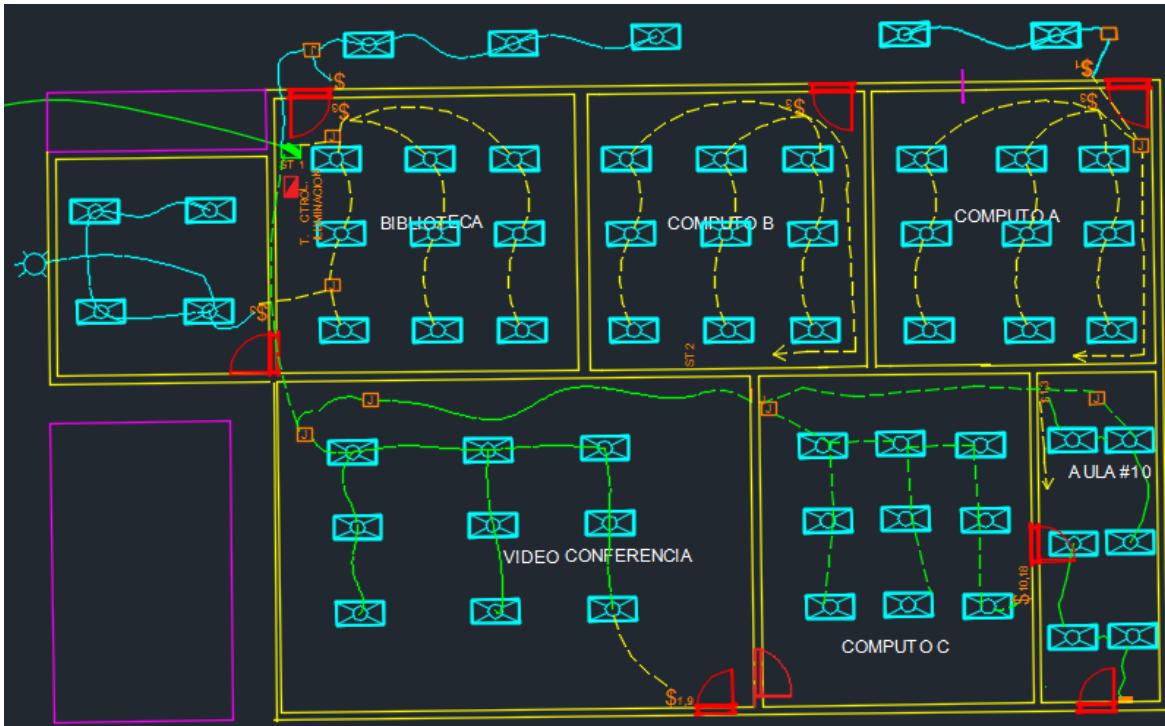


Figura 1. Diagrama unifilar de distribución de luminarias

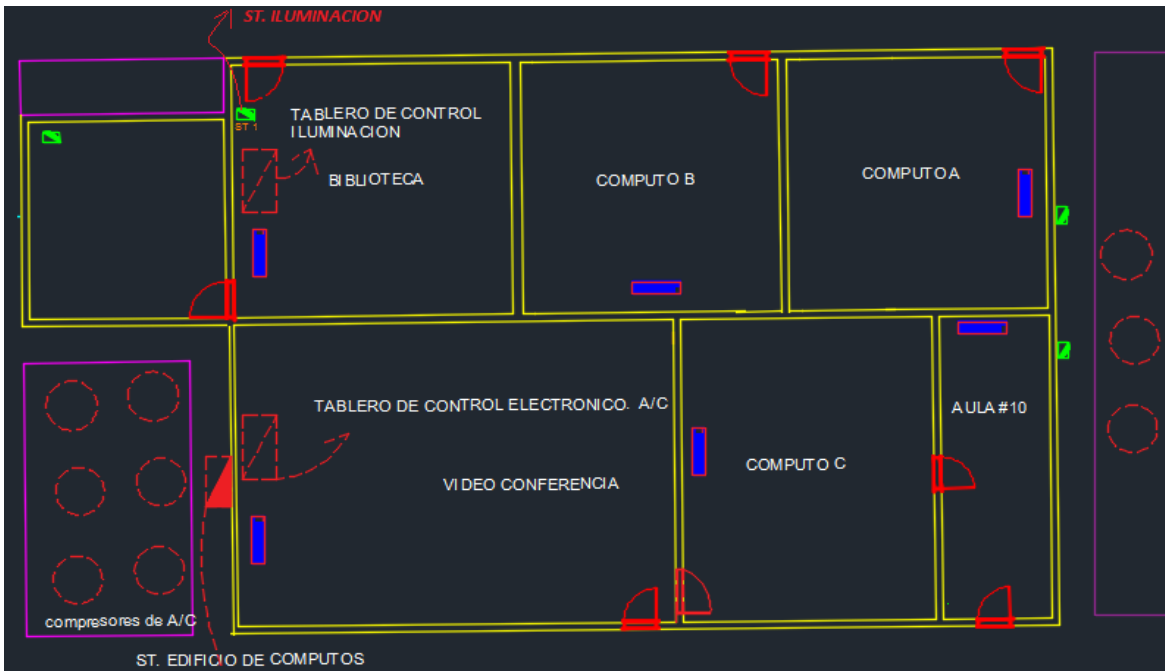


Figura 2. Diagrama esquemático de distribución A/C

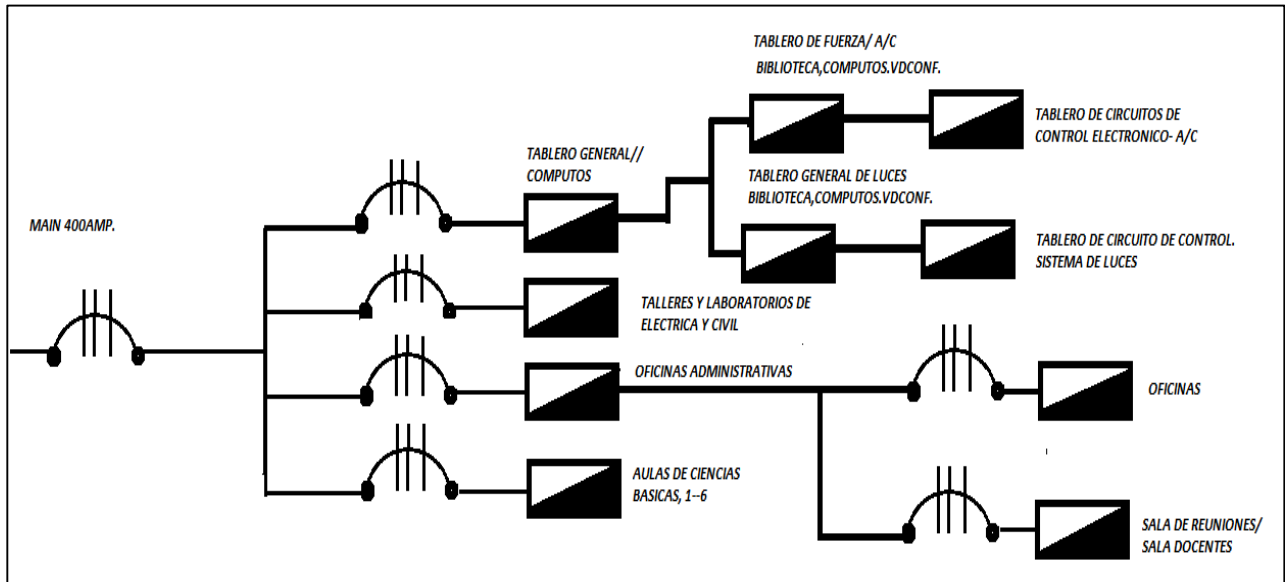


Figura 3. Diagrama Unifilar General

SIMBOLOS ELECTRICOS	
SIMBOLO	DESCRIPCION
	LUZ FLUORESCENTE, PANTALLA SUPERFICIAL 4x32w.
	ST SUB TABLERO
	CAJA DE REGISTRO
	MOTOR DE VETILACION
	LUZ INCANDESCENTE TIPO SPOT LIGHT
	TOMACORRIENTE DOBLE
	INTERRUPTOR SENCILLO
	INTERRUPTOR TRIPLE
	TABLERO DE CONTROL ELECTRÓNICO
	SUB TABLERO DE CARGA ELECTRICA.
	AIRE ACNDICIONADO TIPO SPLIT
	COMPRESOR A/C
	PROTECCION TERMOMAGNETICA, MAIN

8. CONCLUSIONES

- La utilización del Sistema Digital permite mayor control y minimiza los gastos en consumo eléctrico, así como también protege equipos encendidos accidentalmente manteniendo el ciclo de vida útil invariable.
- El uso de la Aplicación Móvil, permite el acceso y control del flujo de energía eléctrica de forma local o externa a la institución.
- El desarrollo de este proyecto permitió a estudiantes y docentes aprender y aplicar la tecnología Arduino, Android y web, mejorando su desempeño laboral y académico.
- La institución se vio beneficiada, pues ahora cuenta con un sistema digital para el control eléctrico que se integró al Sistema de préstamo de llaves lo cual optimiza el uso de los recursos en los laboratorios de cómputo y luces de Biblioteca.

9. RECOMENDACIONES

- Señalizar la caja de control donde se encuentra alojada la Placa Arduino.
- Mantener privado el acceso al Router que genera la red para el acceso al Arduino, no brindando acceso a estudiantes para no saturar la red y garantizar la eficacia del funcionamiento del sistema.
- Mantener ventilada el área de las cajas de control.
- Migrar el Servidor local a Servidor Web, para optimizar el uso de la aplicación móvil para Android.
- Establecer un programa de actualización, aumentando el número de aulas controladas con el Sistema de Control Digital.
- Desarrollar una capacitación al personal técnico para el uso y buen funcionamiento del Sistema Digital.

10. GLOSARIO

Información

Proviene del latín informare que significa noticia, información o instrucción. Este término afecta a todos los factores susceptibles de ser transmitidos o almacenados.

Informática

Proviene de la contracción de las palabras información automática.

Computación

Ciencia que agrupa al conjunto de técnicas y métodos que nos permiten estructura información haciendo uso del computador como herramienta de trabajo.

Domótica

La domótica es un conjunto de tecnologías que se aplican para la automatización y control inteligente del hogar.

Electricidad

Forma de energía que produce efectos luminosos, mecánicos, caloríficos, químicos, etc.

Arduino

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing

Shield

Un shield es una placa impresa que se pueden conectar en la parte superior de la placa Arduino para ampliar sus capacidades, pudiendo ser apilada una encima de la otra.

Processing

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

Aula

Espacio físico al que se le controla el sistema eléctrico.

Pin

Cada uno de los pines físico con los que cuenta un Arduino.

Pabellón

Sección de aulas que se encuentran en el mismo edificio.

App Android

Cliente para dispositivos Android.

Cliente JAVA

Programa escrito en JAVA que se encarga de leer las peticiones y redirigirlas a los respectivos Arduinos.

W5100

Este shield permite conectar Arduino a Internet

Ethernet Shield

Hardware que permite expandir las capacidades del Arduino.

11. REFERENCIAS

ARDUINO - ArduinoEthernetShield [En línea]. [fecha de consulta: 18 mayo 2016].

Disponible en: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

ARDUINO – WiFi. [En línea]. [fecha de consulta: 18 abril 2016].

Disponible en: <http://arduino.cc/en/Reference/WiFi>

CRESPO, José Enrique. Shields para Arduino : aprendiendo Arduino [En línea]. Madrid, 2015.

[fecha de consulta: 14 abril 2016].

Disponible en : <https://aprendiendoarduino.wordpress.com/2015/03/23/shields-para-arduino/>

DE VITO, Michael J. Principios de electricidad y electrónica. México, D. F. : Limusa, 1992. Páginas: 120 p. ISBN: 9681800273.

LAPATINE, Sol. Electrónica en sistemas de comunicación. México, D.F. : Limusa, 1993. 379 p. ISBN: 9681816412.

RIZZONI, Giorgio. Principios y aplicaciones de la ingeniería eléctrica. 3ª. ed. Bogotá: McGraw Hill, Bogotá. 2002. 976 p. ISBN: 9584103024.

SÁNCHEZ, Julio. Arduino: el corazón libre del mundo hiperconectado [En línea]. México, 2014.

Disponible en: <http://eleconomista.com.mx/industrias/2014/07/02/Arduino-corazon-libre>

12. ANEXO. INSTALACIÓN DE CLIENTE JAVA PARA LECTURA DE PETICIONES

1. Instalación física de una computadora conectada al mismo router o la misma subred de Arduino, es importante que esta máquina tenga acceso a internet si es que el sistema PHP se encuentra en internet, de no ser así, solo es necesario asegurarnos que alcanza al host en donde está ubicado en Sistema PHP.
2. Instalación del Sistema Operativo Linux Ubuntu 14.04 de 32 bits, es importante configurar dicho sistema para que no requiera contraseña para iniciar sesión, esto debido a que, si “x” razón la máquina se apagara y encendiera de nuevo, necesita abrir automáticamente el programa cliente JAVA. Además es conveniente que el equipo este configurado para encender al tener energía eléctrica para mantener siempre encendido, sin embargo, lo más recomendado es, conectar la máquina a una UPS, lo mismo para el router y los Arduinos.
3. Instalación del siguiente paquete: JDK 8 o mayor, no usar el OPEN JDK porque no es compatible.
4. Instalación del programa Cliente Arduino, esto con el Instalador provisto para dicho programa, al instalarse se cerrará la sesión actual, después de esto, iniciar nuevamente.
5. Comprobar que lee las peticiones correctamente, y así mismo que los Arduinos responden a sus solicitudes

-
6. En caso que esto no suceda, favor revisar el archivo de configuración en `/home/Usuario/ArduinoCliente/config.json`, en ella encontrará las configuraciones principales del programa, modificar a conveniencia y reiniciar la máquina.

VISIÓN

Ser una institución educativa líder en educación tecnológica a nivel nacional y regional, comprometida con la calidad, la empresarialidad y la pertinencia de nuestra oferta educativa.

MISIÓN

Formar profesionales integrales y competentes en áreas tecnológicas que tengan demanda y oportunidad en el mercado local, regional y mundial, tanto como trabajadores y como empresarios.

VALORES

EXCELENCIA: *Nuestro diario quehacer está fundamentado en hacer bien las cosas desde la primera vez.*

INTEGRIDAD: *Actuamos congruentemente con los principios de la verdad en todas las acciones que realizamos.*

ESPIRITUALIDAD: *Desarrollamos todas nuestras actividades en la filosofía de servicio, alegría, compromiso, confianza y respeto mutuo.*

COOPERACIÓN: *Actuamos basados en el buen trabajo en equipo, la buena disposición a ayudar a todas las personas.*

COMUNICACIÓN: *Respetamos las diferentes ideologías y opiniones, manteniendo y propiciando un acercamiento con todo el personal.*

La Escuela Especializada en Ingeniería ITCA - FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro centros regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.



**SEDE CENTRAL
SANTA TECLA**

Km. 11.5 Carretera a Santa Tecla, La Libertad.
Tel. (503) 2132-7400
Fax. (503) 2132-7599



**CENTRO REGIONAL
SANTA ANA**

Final 10a. Av. Sur,
Finca Procavia
Tels. (503) 2440-4348
y (503) 2440-2007
Tel./Fax. (503) 2440-3183



**CENTRO REGIONAL
MEGATEC ZACATECOLUCA**

Km. 64 1/2, desvío Hacienda El Nilo, sobre autopista a Zacatecoluca y Usulután.
Tels. (503) 2334-0763
y (503) 2334-0768



**CENTRO REGIONAL
SAN MIGUEL**

Km. 140, Carretera a Santa Rosa de Lima.
Tels. (503) 2669-2292
y (503) 2669-2298
Fax. (503) 2669-0061



**CENTRO REGIONAL
MEGATEC LA UNIÓN**

Calle Santa María, Col. Belén, atrás del Instituto Nacional de La Unión.
Tel. (503) 2668-4700