



UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO SIMÓN BOLÍVAR
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN

INFORME FINAL DE TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE
INGENIERO EN ELECTRÓNICA

“DISEÑO DE PROTOTIPO DE SISTEMA EMBEBIDO CON LÓGICA
SECUENCIAL PARA LA GESTIÓN DEL USO DE PUESTOS DE TRABAJO EN
LABORATORIOS”

AUTORES:

Freddy Josué Jirón Pérez

Ian Adonay Orozco García

TUTOR:

Msc. Ing. María Virginia Moncada

Lunes, 12 de octubre del 2015

Resumen / Abstract

El presente documento muestra los pasos seguidos y resultados obtenidos en el diseño de un prototipo basado en un sistema embebido para la gestión eficiente de laboratorios que favorezcan el aprendizaje, así como brindar al usuario una herramienta que controle y monitoree las conexiones eléctricas, intensidad de iluminación en puestos de trabajo del laboratorio, generar registros digitales de las actividades llevadas a cabo en el laboratorio y observar los registros obtenidos en un sitio web en red local.

Contenido

<i>Resumen / Abstract</i>	i
Contenido	ii
<i>Lista de figuras</i>	iv
<i>Lista de tablas</i>	iv
1 Introducción.....	1
2 Antecedentes	2
3 Justificación.....	3
4 Objetivos.....	4
4.1 General	4
4.2 Específico.....	4
5 Marco teórico.....	5
5.1 Teoría de Control.....	5
5.1.1 Sistema Embebido	6
5.2 Automatización de actividades.....	7
5.3 Microcontrolador	7
5.3.1 Estructura interna	8
5.3.2 El procesador o UCP.....	9
5.3.3 Memoria de programa	9
5.3.4 Memoria de datos.....	11
5.3.5 Configuración de los puertos en un microcontrolador	13
5.3.6 Estructura de código.....	14
5.3.7 Lógica secuencial	15
5.4 Protocolo I2C (Inter-Integrated Circuit)	15
5.4.1 Definición	15
5.4.2 Funcionamiento	16
5.5 USB.....	17
5.5.1 Evolución	17
5.5.2 USB en los microcontroladores.....	18
5.6 Actuadores	19
5.6.1 Diferencias entre contactor y relé	19
5.6.2 Transistores de potencia	21

5.7	Modelo cliente-servidor.....	22
5.7.1	¿Qué es un cliente?	22
5.7.2	¿Qué es un servidor web?	22
5.7.3	Características del modelo.....	22
5.7.4	Ventajas.....	23
5.7.5	Desventajas	23
5.8	Servidor Apache	23
5.8.1	¿Qué es?	23
5.8.2	Funcionamiento	24
6	Análisis y presentación de resultados	25
6.1	Análisis.....	25
6.2	Diseño del prototipo.....	26
6.2.1	Módulo Unidad Central.....	28
6.2.2	Módulo Control de Puesto.....	34
6.2.3	Aplicación	42
6.2.4	Módulo WEB.....	47
6.3	Prueba del diseño propuesto.....	49
6.4	Análisis y corrección de errores	50
7	Conclusión.....	51
8	Recomendaciones	52
9	Referencias	53
	Anexos.....	55

Lista de figuras

Figura 1. Diagrama de bloques sistema de control de lazo abierto.....	5
Figura 2. Diagrama de bloques sistema de control de lazo cerrado.....	6
Figura 3. Diagrama de arquitectura Von Neumann.....	8
Figura 4. Diagrama de arquitectura Harvard.....	9
Figura 5. Diagrama de memoria de programa.....	10
Figura 6. Organización interna de una RAM 64X4.....	12
Figura 7. Estructura de la RAM en un microcontrolador.....	13
Figura 8. Modelo simple de entrada/salida.....	15
Figura 9. Relé Zhejiang Dongya Electronic de 5Volts DC.....	20
Figura 10. Contactador Siemens Sirius 3 KW/400V.....	21
Figura 11. Funcionamiento del servidor Apache.....	24
Figura 12. Diagrama general del sistema.....	26
Figura 13. Esquema general del sistema.....	27
Figura 14. Circuito del módulo Unidad Central.....	29
Figura 15. Esquema del módulo unidad central.....	29
Figura 16. Vista frontal de panel de control microcontrolador maestro.....	30
Figura 17. Diagrama de flujo propuesto para la unidad de procesamiento.....	32
Figura 18. Diagrama de flujo propuesto para la interrupción de la unidad de procesamiento.....	33
Figura 19. Diagrama en bloques del módulo de control de puesto.....	35
Figura 20. Vista frontal de módulo control de puesto.....	36
Figura 21. Circuito del módulo de control de puesto.....	37
Figura 22. Diagrama de flujo propuesto para subsistema de procesamiento y comunicación I.....	38
Figura 23. Diagrama de flujo propuesto para subsistema de procesamiento y comunicación II.....	39
Figura 24. Diagrama de flujo propuesto para la interrupción del subsistema de procesamiento y comunicación.....	39
Figura 25. Circuito de driver para corriente alterna.....	40
Figura 26. Circuito del driver LED.....	41
Figura 27. Interfaz gráfica de la aplicación.....	44
Figura 28. Captura de pantalla de página WEB.....	48
Figura 29. Diagrama de flujo propuesto para el módulo WEB.....	49

Lista de tablas

Tabla 3. Especificaciones técnicas de modelos.....	41
Tabla 4. Descripción de interfaz gráfica.....	45
Tabla 5. Mensajes USB.....	46

1 Introducción

La Facultad de Electrotecnia y Computación (FEC) de la Universidad Nacional de Ingeniería (UNI) cuenta con laboratorios destinados a la realización de prácticas académicas. Cada laboratorio requiere la asignación de un responsable, el cual se encarga de llevar un registro de las actividades realizadas, préstamos de puestos y dispositivos presentes en el mismo.

Actualmente, estas tareas son llevadas a cabo mediante la elaboración de un manuscrito, generando una dependencia en la disponibilidad de los documentos al momento de proporcionar esta información a las diversas autoridades pertinentes que lo requieren, lo que consume tiempo y recurso humano, haciendo de esta una gestión deficiente.

Un laboratorio de ingeniería Electrónica tiene puestos de trabajo que disponen de equipos delicados y de alto costo económico, por tanto, es recomendable que estos equipos se desconecten al finalizar la jornada académica o práctica.

Por tanto, es necesario actualizar el sistema de registro de actividades en los laboratorios y al mismo tiempo diseñar un sistema capaz de manipular la alimentación de los dispositivos presentes en los puestos de trabajo.

Bajo este contexto se propuso diseñar un sistema de control que permite encender/apagar los equipos a utilizar en los puestos de trabajo en un laboratorio, así como generar registros digitales del uso de este para dar solución a los problemas existentes en la actualidad.

2 Antecedentes

Desde su invención, el Microcontrolador ha permitido solucionar problemas que se presentan a diario, entre ellos el control de iluminación.

Actualmente según (Latin America Journals Online, 2014), (Sistema regional de información en línea para revistas científicas de América Latina, el Caribe, España y Portugal, 2014) y (Portal de publicaciones científicas y técnicas, 2014) no hay algún trabajo que se asemeje a éste, sin embargo, existen avances a nivel Latinoamericano, como por ejemplo (Londoño Ospina, Barrera Durango, Carvajal, & Fonseca, 2012), el cual es un diseño de un sistema domótico de bajo coste para aplicaciones didácticas, donde emplean el concepto de tecnología ambiental en busca de la forma más óptima para el ahorro de energía, controlando la iluminación del sitio que va en dependencia de la presencia de personas en el lugar. Al final concluyeron que la interfaz visual carece de atractivo, registro y alarma.

En el mercado existen dispositivos para apagar luces de forma remota o automática, ya que existe Dimmer¹ con mando, sencillos y complejos de 2, 4 u 8 líneas, temporizadores, basados en componentes semi-conductores analógicos para restablecimiento de red, entre otros; estos carecen de registro y la mayoría son orientados para uso en el hogar.

Estos dispositivos aparte de tener una sola función, son costosos o difíciles de conseguir, porque para introducirlos a Nicaragua hay muchos requerimientos por parte de aduana (envío, trámites, impuestos sobre el artículo, etc), lo que hace poco rentable e inseguro, su comercialización implica gastos en varios dispositivos y a veces el producto no llega a su destino.

¹ Regulador de energía

3 Justificación

En la actualidad, los laboratorios de la carrera de Ingeniería Electrónica destinados a prácticas académicas en la Universidad Nacional de Ingeniería no cuentan con un registro digital para las actividades realizadas, tampoco con un sistema capaz de manipular la alimentación de los dispositivos presentes en los puestos de trabajos.

Por tanto, es necesario diseñar un sistema que permita realizar registros digitales que facilite la elaboración de informes y que sea capaz de manipular la conexión en los puestos de trabajo al suministro eléctrico, y así se evitaría que los equipos queden encendidos por error humano y tengan mejor vida útil con una adecuada administración.

Con la acreditación constante de la carrera, se requiere evidencias claras del uso de laboratorios para docencia e investigación y el uso de este sistema daría un plus al contar con un sistema novedoso en los laboratorios.

Con el deseo de contribuir al desarrollo de la universidad y basándonos en los conocimientos adquiridos durante la formación como estudiantes de la carrera de ingeniería electrónica, surgió la idea de realizar este trabajo monográfico, con el fin de permitir un mayor control de las actividades experimentales de aprendizaje que se llevan a cabo en los laboratorios, obteniendo datos más precisos y fiables.

4 Objetivos

4.1 General

- Diseñar un sistema de control y registro para la gestión eficiente en el uso de puestos de trabajo, mediante microcontroladores en laboratorios de Electrónica.

4.2 Específico

- Diseñar un sub-sistema que permita al usuario ajustar los niveles de iluminación y habilitar/deshabilitar la alimentación comercial en los puestos de trabajo (mediante controladores y actuadores de potencia) por medio de un controlador central y actuadores.
- Diseñar un sistema de registro (hora de uso, grupo, asignatura y encargado) accesible por medio de un explorador web mediante el modelo cliente-servidor y texto plano.
- Diseñar una interfaz gráfica amigable con el usuario para el monitoreo y control de los puestos de trabajo mediante el lenguaje de programación C#.
- Integrar los diversos módulos del sistema mediante protocolos de comunicación I2C y USB 2.0.

5 Marco teórico

5.1 Teoría de Control

Los sistemas de control son el resultado de una necesidad existente para resolver problemas cotidianos, de industrias, o bien, facilitar la ejecución de diversas actividades, en la cual el operario ejerce la función de control, decide cuando y como manipular las variables de tal modo que se obtengan los resultados que se desean.

Los elementos básicos de un sistema de control son:

- Objetivos
- Sistema de control
- Resultados

Los objetivos son las entradas o señales actuantes de interés y los resultados son las salidas o variables que se desea controlar. El objetivo de un sistema de control es controlar las salidas en alguna forma prescrita mediante las entradas a través de los elementos del sistema de control (Kuo, 1996).

Hay dos tipos de sistemas de control:

- Sistema de control en lazo abierto (sistemas no realimentados).

La figura 1 presenta la estructura general de un sistema de control de lazo abierto.



Figura 1. Diagrama de bloques sistema de control de lazo abierto

Este tipo de sistema se puede dividir en dos partes: el controlador y el proceso controlado. Siendo la señal de entrada la que se aplica al controlador, cuya salida actúa como señal actuante y esta luego controla el proceso para lograr que la variable controlada se desempeñe de acuerdo a los estándares preestablecidos. El controlador puede ser un amplificador, unión mecánica, filtro, u otro elemento de

control para casos simples. En casos complejos, puede ser una computadora tal como un microprocesador o un microcontrolador. Gracias a su simplicidad y economía se encuentra en muchas aplicaciones no críticas, e.g., lavadoras.

- Sistemas de control en lazo cerrado (sistema realimentado).

La figura 2 presenta la estructura general de un sistema de control de lazo cerrado.



Figura 2. Diagrama de bloques sistema de control de lazo cerrado

Este tipo de sistemas es el más exacto debido a que hay una conexión de salida hacia la entrada para obtener un mejor control, ya que, la señal de salida se compara con la entrada de referencia y así obtener la diferencia entre ambas (error) y corregirlo (compensación).

5.1.1 Sistema Embebido

El término “embebido” (también es conocido como “incrustado”) se utiliza para referirse a dispositivos que son parte integral del sistema, controlar equipos, operación de maquinarias o plantas industriales completas, está caracterizado por tener circuitos integrados que son parte fundamental del sistema en que se encuentran. Este sistema puede quedar tan oculto al ojo del usuario que la presencia de los CI² no está a simple vista.

² Circuito Integrado

5.2 Automatización de actividades

Para disminuir el error humano, es recomendable automatizar aquellas actividades que suelen ser repetitivas y así cumplirlas con precisión al menos en un 90% y garantizando confiabilidad.

Los controladores hoy en día permiten realizar ese tipo de actividades, siguiendo una lógica basada en las actividades que realiza el usuario. Algunos ejemplos de estas actividades sería, el cierre seguro de la puerta principal, el apagado de luces, aire acondicionado y alimentación total del local.

5.3 Microcontrolador

El primer Microcontrolador fue creado en 1971 por la empresa Texas Instruments después de experimentar con los famosos microprocesadores de Intel 4004 y 8008, saliendo el Microcontrolador TMS 1000 en 1974. Desde entonces hasta la fecha, han ayudado a la automatización de tareas. Algunos de sus beneficios más notables son: su bajo coste, capacidad de cómputo y la versatilidad para adaptarlo a cualquier problemática a la que se quiera dar solución en comparación al PLC³.

Entre sus características principales a manera general entre todas las familias están las siguientes:

- Consumo de cómputo menor al 1mA.
- RAM, CPU, EEPROM y módulos de comunicación serial⁴ integrados en un solo dispositivo.
- Conversores Analógico/Digital, Módulo CCP⁵, Entradas/Salidas dinámicas.
- Generador de impulsos de reloj⁶ que sincronizan el funcionamiento de todo el sistema.
- Bajo costo y compacto.

³ Controlador Lógico Programable

⁴ USART, SPI, I2C, entre otros

⁵ Comparador, Captura y PWM

⁶ Puede ser interno o externo

Con estas características, dichos controladores se han hecho paso poco a poco en la industria, ya que tienen un consumo casi nulo (en cuanto a corriente) y son programables en lenguaje de alto nivel si uno lo desea.

Estos integrados necesitan periféricos externos para poder controlar cargas grandes (motores, cisternas, etcétera), por lo que se auxilian de actuadores como relés y contactores.

Cabe destacar, que una de sus ventajas es la versatilidad al momento de programarlo; es decir, poder ensamblar un sin número de instrucciones en un solo dispositivo sin necesidad de quitarlo en donde esté montado, ya que por medio de sus terminales ICSP⁷, se puede programar directamente aplicando altos voltajes de programaciones en los pines correspondientes y con una secuencia síncrona.

Una de las desventajas de los microcontroladores es la susceptibilidad al ruido, por lo que el aislamiento y las protecciones deben ser las mejores para que el integrado funcione de la mejor manera posible.

5.3.1 Estructura interna

A diferencia de las computadoras, los microcontroladores utilizan una arquitectura interna distinta.

Arquitectura Von Neumann: Dispone de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. Se accede a ésta a través de un sistema de buses únicos (direcciones, datos y control), ver figura 3.

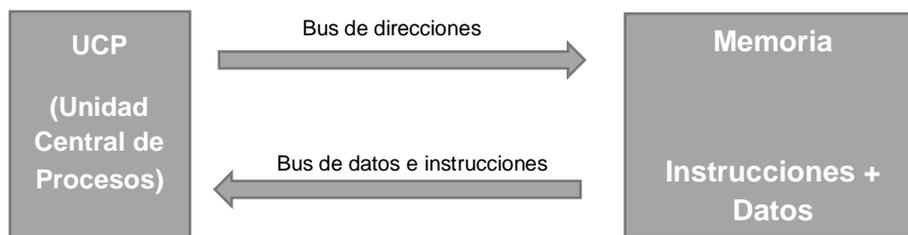


Figura 3. Diagrama de arquitectura Von Neumann

⁷ Programación Serial en Circuito o In-Circuit Serial Programming

Arquitectura Harvard: La principal diferencia con la anterior es que tiene la memoria de datos dividida en dos: una que contiene instrucciones y otra que contiene datos. Cada una con su bus de acceso independiente y con posibilidad a realizar operaciones de acceso R/W⁸ simultáneamente en ambas memorias. Los microcontroladores utilizan esta estructura, ver figura 4.

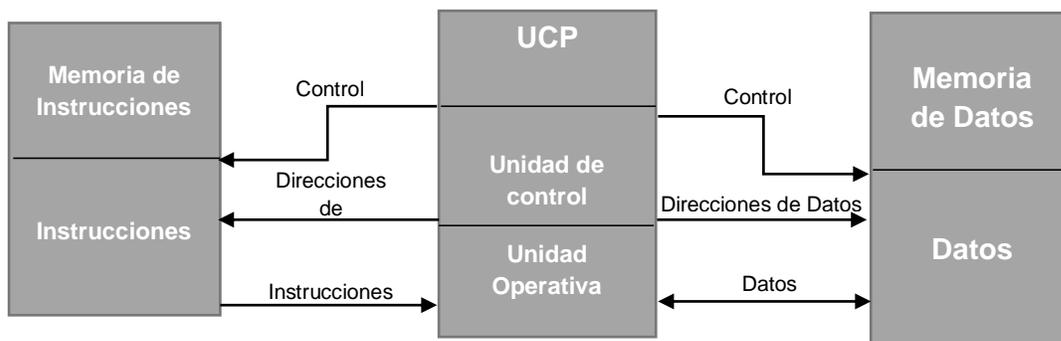


Figura 4. Diagrama de arquitectura Harvard

5.3.2 El procesador o UCP

Es la parte de más importancia en el microcontrolador. Es el encargado de direccionar la memoria de instrucciones, recibir el código de operación de la instrucción en curso, decodificarlo, ejecutarlo y almacenar el resultado.

5.3.3 Memoria de programa

Esta es la memoria de instrucciones, aquí es donde se almacena el programa o código que el microcontrolador debe ejecutar. Hasta el momento no hay posibilidad de utilizar memorias externas de ampliación para lo que compete a instrucciones. Existen 5 tipos de memoria, pero las usadas para este dispositivo son estas dos:

- **Memoria EEPROM** se graba a través de un circuito grabador que es controlado mediante un PC. Se puede grabar y borrar una memoria EEPROM aproximadamente 1000 veces y es una memoria lenta.

⁸ Read/Write o Lectura/Escritura

- **Memoria FLASH** se podría decir que es una versión mejorada de la EEPROM ya que tiene las mismas características pero más optimizada, y consume menos energía, tiene mayor capacidad de almacenamiento, fijo sustituto de la EEPROM desde hace años atrás.

La memoria de programa se divide en páginas de 2048 posiciones. En dependencia del modelo de microcontrolador que se utilice, la implementación en el número de posiciones en la memoria de programas puede variar. Los microcontroladores de gama baja tienen por lo general 1024 posiciones implementadas, es decir, de 0000h a 03FFh; el resto no está implementado, véase el área gris en la figura 5.

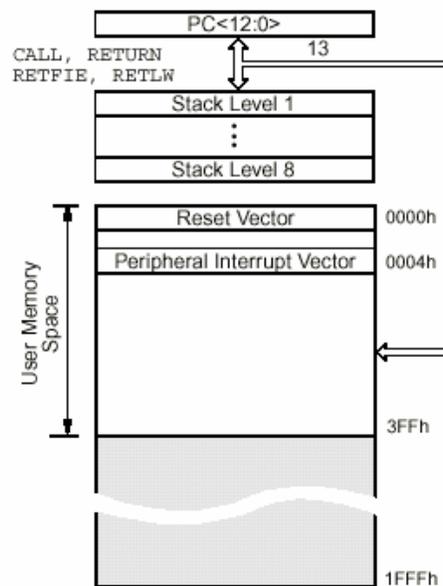


Figura 5. Diagrama de memoria de programa

Cuando se ejecuta un RESET, el contador de programa (PC) apunta a la primera dirección, en este caso 0000h y el microcontrolador se inicia nuevamente. Éste concepto es muy importante debido a que en las primeras direcciones del programa se deben escribir todo lo relacionado a la iniciación del mismo, es decir, configuración de entradas/salidas, etc.

En el caso de interrupciones, el contador de programa apunta a la dirección 0004h, ahí es donde se aloja la programación para atender a la solicitud.

La pila o stack, consta de 8 posiciones (niveles), es como que estén apilados 8 bloques de cemento, el último en ponerse es el primero en retirar. Con esta analogía, cada bloque contiene la dirección y los datos de la instrucción que se está ejecutando. Cuando se efectúa una llamada (usualmente se utiliza el término *CALL*) o una interrupción, el ordenador sabe el lugar de retorno (utilizando la instrucción *RETURN* o el equivalente según el caso) para continuar con la ejecución del programa.

5.3.4 Memoria de datos

Este tipo de memorias son en base a la memoria RAM para lo que compete a transferencia Lectura/Escritura de datos, ya que cualquier ubicación de la dirección de memoria es fácilmente accesible como cualquier otra.

La memoria RAM normalmente viene con capacidades de palabras de 1K, 4K, 8K, 16K, 64K, 128K y 1024K, y con tamaños de palabra de uno, cuatro u ocho bits. Claro está que se puede aumentar la capacidad de la palabra y su tamaño mediante la combinación de integrados de memoria.

La figura 6 muestra la arquitectura simplificada de una RAM que almacena 64 palabras de cuatro bits cada una. Estas palabras tienen direcciones que van de 0 a 63_{10} .

Con el fin de seleccionar una de las 64 ubicaciones para lectura o escritura.

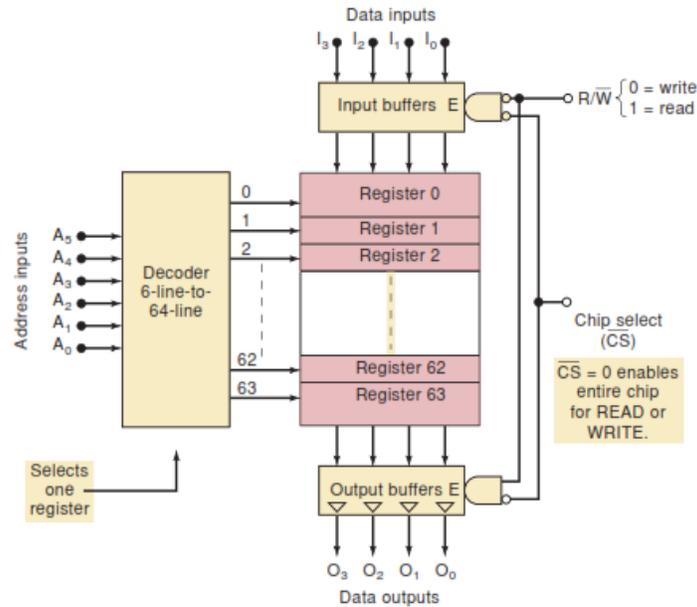


Figura 6. Organización interna de una RAM 64X4

Esta parte de la memoria se divide en dos:

- **SRAM:** La RAM estática es donde están los registros específicos (SFR) con 24 posiciones de tamaño byte, dos de ellas no son operativas. Los Registros de Propósito General (GPR) con 68 posiciones. En la gama media de los microcontroladores de la familia PIC, la RAM está dividida en dos bancos (banco 0 y banco 1), el tamaño de cada uno va en dependencia de la capacidad del microcontrolador, en la siguiente figura se muestra que cada banco es de 128 bytes (7Fh), véase figura 7.

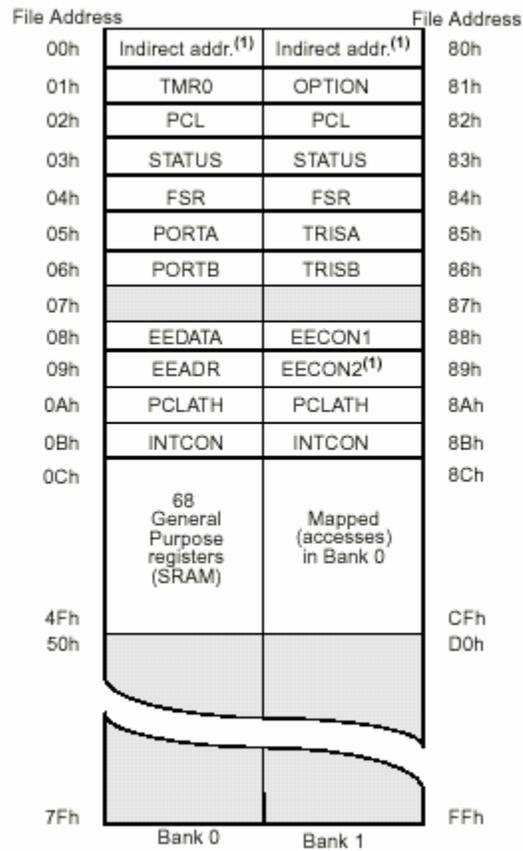


Figura 7. Estructura de la RAM en un microcontrolador

- **EEPROM:** Aquí se pueden almacenar datos que no se pierden al desconectar la alimentación.

5.3.5 Configuración de los puertos en un microcontrolador

La configuración de los puertos en un microcontrolador hace uso de la tabla de registro de la memoria de datos, que está dividida en el **BANCO 0** y **BANCO 1**.

En dependencia del microcontrolador, la dirección de los registros de puertos puede variar, por lo general en gama media-baja es de la siguiente manera:

STATUS	dirección 0x3
PORTA	dirección 0x5
PORTB	dirección 0x6
TRISA	dirección 0x5
TRISB	dirección 0x6

En algunos microcontroladores vienen asignados puertos como entrada de datos por defecto, para tal caso que se quiera configurar cualquier puerto o pin se puede asignar un valor **CERO (0)**, por ejemplo, la analogía **O** puede utilizarse para reflejar **Output**, que indicaría salida, en caso contrario se asigna un **UNO (1)**, como **I** de **Input**. Esa asignación se hace en **TRISA**, **TRISB**. Si **TRISA** es igual a 00000000 quiere decir que todos sus pines están configurados como salida.

Al energizar el microcontrolador, este inicia utilizando el BANCO 0, en dependencia del modelo, familia y fabricante. Habrá que verificar en la hoja de datos el BANCO en el cual se encuentran los registros TRIS para poder realizar la configuración de los mismos, esto se puede hacer utilizando el registro **STATUS**.

STATUS es un registro de 8 bits, donde el quinto bit (RP0) define la posición del banco que se está utilizando, es decir, si RP0 está en CERO (0), está en el BANCO 0 y 1 para caso contrario.

5.3.6 Estructura de código

Los microcontroladores son muy versátiles, ya que se pueden utilizar para solucionar cualquier problemática, tanto ha sido su éxito que es programable en varios lenguajes, ensamblador, C, C++, entre otros, pero hay algo en común en todos esos lenguajes y es la estructura del código en el microcontrolador, es decir, el orden a seguir antes de escribir las instrucciones del código principal, a como se mencionó anteriormente, la configuración de los registros, variables de entradas y salida, entre otras cosas, todas ellas deben estar previamente configuradas para que el código principal funcione de forma correcta.

- Cabecera
- Directivas del preprocesador
- Prototipos de funciones adicionales
- Función principal (Main)
- Funciones adicionales

5.3.7 Lógica secuencial

Flip-flops, contadores y registros son los principales componentes en un sistema de lógica secuencial, ya que tiene las características de que sus salidas y secuencia de operaciones dependen tanto de las entradas (inputs) presentes y de las entradas que se produjeron anteriormente (en caso que hubiese). Pese que los sistemas lógicos secuenciales son generalmente más complejos que los sistemas lógicos combinatorios, los procedimientos esenciales para la solución de problemas se aplican de igual forma para ambos tipos de sistemas. Ambos son susceptibles a circuitos abiertos, cortocircuitos, fallas internas en los integrados y situaciones similares, véase la figura 8 para mayor referencia.

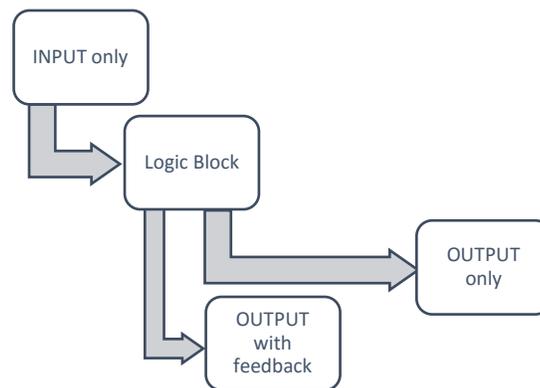


Figura 8. Modelo simple de entrada/salida

Este tipo de sistemas utilizan el flanco de reloj. Los circuitos al tener salidas que alimentan a la entrada implica el uso de variables de salida en la parte condicional de la descripción.

5.4 Protocolo I2C (Inter-Integrated Circuit)

5.4.1 Definición

El protocolo I2C, también llamado, I cuadrado C, desarrollado por Phillips Semiconductor en 1992⁹, es un protocolo serial de dos líneas en el cual se puede comunicar con múltiples dispositivos que tienen una dirección única.

⁹ I2C fue patentado ese año

Según (García Breijo, 2008) puede configurarse como comunicación de un maestro y varios esclavos o una configuración *Multimaestro*. En ambas configuraciones, el dispositivo maestro es el que tiene la iniciativa en la transferencia, decide con quién se realiza, el sentido de la misma (envío o recepción desde el punto de vista del maestro) y cuándo finaliza. Cuando el maestro inicia una comunicación, primero transmite la dirección del dispositivo con el cual se quiere comunicar y los esclavos comprueba si la dirección concuerda con la suya. La transmisión puede ser de lectura o escritura, el último bit de la dirección lo indica; así el maestro estará en transmisión y el esclavo en recepción o viceversa. En cualquier caso la señal de reloj la genera el maestro.

Los dos hilos del bus I2C SDA¹⁰ y SCL¹¹ son dos líneas de colector abierto, es decir, ambas líneas deben utilizar resistencias externas en configuración *pull-up* para asegurar un nivel alto cuando no hay dispositivos conectados al bus.

El número de dispositivos conectados y la longitud de conexión están limitados por la capacidad de direccionamiento (de 7 a 10 bits) y por la máxima carga del bus (400pF). La velocidad máxima estándar es de hasta 100 Kbps, la rápida hasta 400 Kbps y la Alta hasta 3.4 Mbps.

5.4.2 Funcionamiento

La transmisión comienza con un bit de inicio, generalmente en las funciones de programación es llamado START y termina con el bit de finalización o STOP. START se establece con una transición de alto a bajo en la línea SDA (normalmente a nivel alto) cuando la línea SCL está a nivel alto. STOP se establece cuando se produce una transición de bajo a alto en la línea SDA cuando SCL está a nivel alto; de esta forma los datos de la línea SDA solo cambian en el estado bajo de la línea SCL.

¹⁰ Data Line

¹¹ Clock Line

Ya con la transmisión iniciada, el controlador maestro envía la dirección del esclavo con el que se desea establecer comunicación. En este caso, la dirección es de 7 bits con formato de byte. Tras la dirección se adjunta un bit de lectura/escritura.

Una vez el master envía la dirección (o datos), el esclavo genera un bit de reconocimiento (ACK), si el maestro no recibe ese bit, la comunicación se interrumpe, generando la señal de STOP. El maestro también puede recibir datos, en este caso es él quién genera la señal de reconocimiento para cada byte recibido, menos para el último, en este caso el esclavo libera la línea SDA y el master genera un STOP.

Puede ser que el maestro luego de una transmisión/recepción, no deje de utilizar el bus y siga en comunicación con el esclavo, en esta ocasión solo repite la condición de START que va luego de un pulso de reconocimiento.

En los microcontroladores de gama media existen dos módulos que permiten trabajar con I2C, el BSSP (Basic Synchronous Serial Port) y el MMSP (Master Synchronous Serial Port), se diferencian en la manera en la cual el maestro trabaja. El MSSP permite detectar condiciones de START y STOP por interrupción.

Los registros asociados a este módulo son seis: **SSPCON**, **SSPCON2**, **SSPADD**, **SSPBUF**, **SSPSTAT** y el **SSPSR** (Microchip, 2001).

5.5 USB

5.5.1 Evolución

En su investigación (Axelson, 2009) encontró que la principal razón por la cual las nuevas interfaces no aparecen muy a menudo es debido a que las interfaces existentes tienen la ventaja de todos los periféricos que utilizan los usuarios. Pero a medida que los ordenadores se hicieron más poderosos y el número de periféricos aumentaron, las interfaces más antiguas tuvieron problemas de “cuello de botella”, es decir, comunicaciones lentas y con opciones limitadas para la expansión. Esta es la razón por la que el protocolo USB fue impulsado al desarrollo.

El *Universal Serial Bus* en su versión 1.0 fue lanzado en Enero de 1996 disponible para los PC con el lanzamiento de Windows 95. La compatibilidad de USB en estas versiones fue limitado y con errores, pese que tampoco habían muchos periféricos que utilizaran USB en esos tiempos, algo que mejoró cuando se lanzó Windows 98, pues ya existían más periféricos USB y lo ayudó a afianzarse como una interfaz popular.

En abril del 2000 llegó la revisión 2.0 de USB que agregó velocidades de 480Mbps. Esta alta velocidad hizo que el protocolo fuera más atractivo para periféricos como impresoras, unidades de disco y cámaras de vídeo, manteniendo la compatibilidad con interfaces antiguas y de menor velocidad.

USB 3.0 revisión 1.0 fue lanzado en Noviembre de 2008 que fue impulsado con Windows 7. Esta versión define una nueva arquitectura de doble bus con dos buses físicos que operan en paralelo, el nuevo bus llega a tener una súper velocidad de 5Gbps y este mismo aumentó diez veces la velocidad de USB 2.0, USB 3.0 al tener doble bus puede transferir datos en ambas direcciones al mismo tiempo. En la actualidad esta interfaz se encuentra en el desarrollo de tecnología *On The Go* (OTG) la cual permite a los periféricos USB acceder a otros dispositivos USB, como por ejemplo cuando se quiere conectar una cámara a una impresora directamente.

5.5.2 USB en los microcontroladores

En la actualidad, los microcontroladores siendo programados en lenguaje de alto nivel permiten amplio número de instrucciones ejecutadas en cuestión de segundos. Esto ha dado paso a que los microcontroladores con hardware USB puedan ser utilizados en un sinnúmero de aplicaciones.

Existen muchos tipos de comunicación USB, pero en microcontroladores de gama alta (hasta en esta gama se integró este módulo). Se utilizan cuatro clases hasta el momento las cuales son:

- **CDC** (*Communications Device Class*): se basa en emular un puerto serial para enlazar la comunicación, en microcontroladores se debe utilizar un driver específico

para poder hacer funcionar el dispositivo que se está desarrollando, permite una velocidad de 1.2 Mbps aproximadamente.

- **Mass Storage:** Utilizada para almacenamiento masivo como en: Pendrives, discos rígidos entre otros dispositivos, aunque puede ser utilizado para guardar pequeños datos en la memoria EEPROM de los microcontroladores, tiene una alta tasa de transferencia.
- **Bulk Transfer:** Es la que permite la mayor tasa de transferencia entre transmisión y recepción. Pueden enviarse largas cadenas de datos sin obstruir el bus ya que esta clase “espera” a que el bus esté disponible. Es mayormente utilizado para transferencias masivas y rápidas, es el que utilizan las impresoras, tiene una transferencia estándar entre 400Mbps a 1Gbps en dependencia de la aplicación. Esta clase al igual que CDC usa un driver para su funcionamiento, pero es un poco más rápido y mejor. Es una de las razones por la cual permite una alta tasa de transferencia.
- **HID (*Human Interface Device*):** Es la clase más lenta que puede ser usado en los microcontroladores, ya que permite una tasa de transferencia estándar de 60 Kbps, pero pese a eso, es la más idónea, debido a que no todas las aplicaciones desarrolladas en microcontroladores requieren una alta tasa de transferencia de datos, no requiere driver ya que usa la tecnología *Plug and Play* y es automáticamente reconocido. Las funciones son sencillas y no se necesita muchos conocimientos de programación para hacer uso de ellas. Es ideal como hardware libre siempre y cuando se respete los marcos legales de las licencias que se estén utilizando.

5.6 Actuadores

5.6.1 Diferencias entre contactor y relé

El electromagnetismo ha generado un buen avance en la manipulación de energía eléctrica para circuitos electrónicos. Un **relé** es un interruptor accionado por

electromagnetismo. El circuito controlado y el circuito controlador están separados galvánicamente entre sí. Está compuesto por una bobina con núcleo de hierro, un inducido como elemento de accionamiento mecánico, un muelle de recuperación y los contactos de conmutación.

El paso de tensión en los terminales de la bobina del electroimán produce un campo electromagnético, atrayendo de esta manera al inducido móvil por el núcleo de la bobina y posteriormente el inducido actúa sobre los contactos del relé (se pueden abrir o cerrar en dependencia del modelo de relé). Si se deja de aplicar corriente a través de la bobina, el inducido recupera su posición inicial mediante la fuerza de un muelle.

La figura 9 muestra un relé de 5 voltios del distribuidor Zhejiang Dongya Electronic.



Figura 9. Relé Zhejiang Dongya Electronic de 5Volts DC

Los contactores se utilizan para solucionar tareas de regulación y mando. Principalmente para control directo de motores, válvulas, acoplamientos, entre otros. Debido a su elevado grado de seguridad y sencillez para diseño, montaje, mantenimiento y puesta en servicio, son los principalmente utilizados en el control industrial. Su funcionamiento es semejante al de un relé, siendo la diferencia más notoria entre ellos la corriente y tensión que pueden controlar sus contactos.

La figura 10 muestra un contactor marca Siemens de 400V.



Figura 10. Contactor Siemens Sirius 3 KW/400V

En dependencia del problema que se deba solucionar, se puede llegar a utilizar un contactor o relé, lo que definirá esto será el tipo de carga que se deba manejar.

5.6.2 Transistores de potencia

En dependencia a las necesidades de diseño se puede optar por el transistor más adecuado para éste.

5.6.2.1 Bipolar

Pionero en el ramo, pese al tiempo que tiene en el mercado desde que se desarrolló. Algunos diseñadores aún lo toman en cuenta, aunque no es el mejor en cuanto a frecuencia de trabajo (10-80Khz) que va en dependencia del fabricante, pero por su gran robustez, capacidad de carga en corriente y coste de fabricación con respecto a otros transistores.

5.6.2.2 MOS

Alta frecuencia de trabajo (100Khz-500Khz) dependiendo del fabricante, alta temperatura de operación (200°C aproximadamente), Alta impedancia de entrada (1K aproximadamente), son las características de este tipo de transistores que varían dependiendo del fabricante, en sí, el transistor MOS tiene ventajas en sus entradas.

5.6.2.3 IGBT

Combinación de ambos anteriormente mencionado, son lo mejor de lo mejor, trabaja con tensión, tiempos de conmutación bajos, disipación mucho mayor (no se calientan tanto como los MOS) y coste altísimo de fabricación.

5.7 Modelo cliente-servidor

Es un proceso basado en la interacción de una computadora local (el cliente) y la remota (el servidor).

5.7.1 ¿Qué es un cliente?

Es el que hace peticiones al servidor, el cual procesa y retorna los resultados al cliente apropiado. Usualmente, los clientes y los servidores se comunican entre sí mediante una red, pero también pueden estar en un mismo sistema (mismo hardware).

El modelo cliente-servidor es usado por los servicios de e-mail, acceso a webs, acceso a base de datos y muchos protocolos de internet (HTTP, SMTP, DNS), etc. (ALEGSA, s.f.)

5.7.2 ¿Qué es un servidor web?

Es un software instalado en un equipo el cual debe tener todas las condiciones necesarias para servir o entregar páginas web que sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización. (Norfi Carrodegas, s.f.)

5.7.3 Características del modelo

La máquina que sirve como servidor (host) es la que tiene en ejecución programas de servidor que contestan las solicitudes de los clientes. Habitualmente los clientes inician la sesión de comunicación y los servidores esperan la llegada de solicitudes.

Pueden actuar como una sola entidad o como entidades separadas, es decir, realizando actividades o tareas independientes.

El servidor puede dar servicio a múltiples clientes de forma concurrente.

La escalabilidad del modelo es transparente para el usuario final, es decir, si hay una actualización de software o actualización/reemplazo de hardware en cualquiera de las dos plataformas.

Las funciones cliente-servidor pueden ser dinámicas, porque un servidor puede convertirse en cliente cuando realiza una solicitud de servicio a otras plataformas dentro de la red.

5.7.4 Ventajas

- Fácil mantenimiento y actualización desde el lado del servidor.
- Toda la información es almacenada en el servidor, ya que por lo general tiene más seguridad que los clientes.
- Favorece el uso de interfaces gráficas más intuitivas con el usuario.

5.7.5 Desventajas

- Si el número de clientes simultáneos es elevado, el servidor puede saturarse.
- Cuando el servidor falla, el servicio queda congelado para los clientes.
- La seguridad en este modelo es importante, por lo que se deben hacer verificaciones en el cliente y servidor.

5.8 Servidor Apache

5.8.1 ¿Qué es?

Es un servidor WEB desarrollado en 1995 de código abierto basado en NCSA httpd, el desarrollo de este fue llevado a cabo por Apache Group y apollado Apache Software Foundation (The Apache Software Foundation, s.f.).

Existen varios tipos de servidores web, entre ellos AppServ, es un software gratuito¹², libre de uso, totalmente configurable y de sencilla instalación en Windows, Linux/Unix y Macintosh, cuenta con los siguientes paquetes:

¹² Comunidad AppServ Open Project <http://www.appservnetwork.com/index.php>

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

- Apache WEB Server
- Lenguaje PHP
- MySQL Data Base
- Manejador de base de datos phpMyAdmin

Al instalarlo en nuestro ordenador dispondremos de un servidor completo, con todos los requisitos para ejecutarlo de forma local.

5.8.2 Funcionamiento

En la figura 11 se puede observar el esquema de funcionamiento de Apache. Consiste en un ordenador corriendo el programa servidor (AppServ), este gestiona las carpetas y recursos que se mostrarán en la página web restringiendo el acceso a ficheros que no se quieran exponer, el cliente hace la petición por medio de un explorador web utilizando la dirección o dominio de la página web de la cual se quiere la información, esta petición es recibida y atendida por el servidor, el cual se encargará de enviar los archivos del sitio web solicitado al cliente.

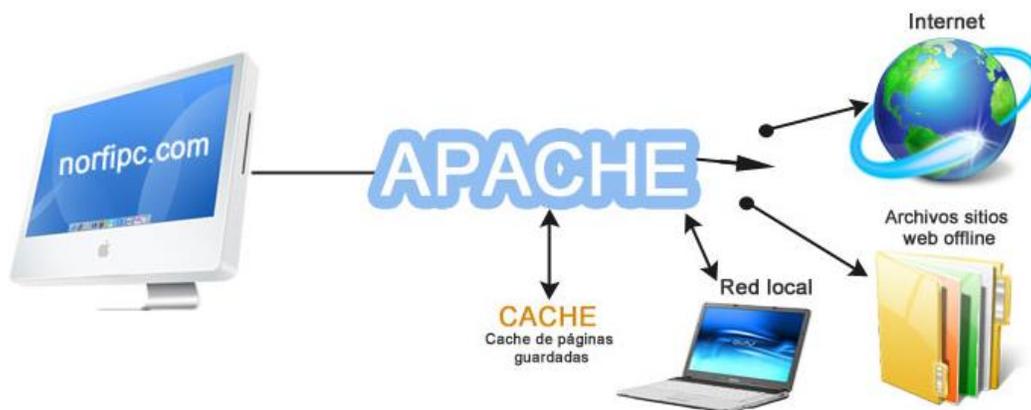


Figura 11. Funcionamiento del servidor Apache

6 Análisis y presentación de resultados

Para la elaboración de los sistemas de control de cargas, iluminación autoajutable y la intercomunicación de los mismos, se usó el método experimental basado en análisis, diseño, prueba y corrección llevando a cabo las siguientes etapas:

6.1 Análisis

En esta etapa se analizó, según observación y experiencia de los autores, la gestión de los laboratorios para identificar las necesidades más significativas, encontrando los siguientes problemas:

- La elaboración del registro de actividades, préstamos de puestos y uso de dispositivos es realizado mediante un manuscrito, lo cual genera en el aspecto administrativo una dependencia en la disponibilidad física de esa información al momento que las autoridades pertinentes lo requieran.
- Carencia de un sistema de protección en caso de interrupciones bruscas del suministro eléctrico debido a equivocaciones al momento de una práctica académica.
- Modo de iluminación poco adaptativo a las necesidades de los usuarios. Los horarios de uso son desde tempranas horas del día hasta la noche. La iluminación empleada es por medio de lámparas fluorescentes el cual no puede ser considerado adecuado para todas las actividades que se requiere.
- El laboratorio se encuentra dividido en múltiples puestos de trabajos independientes, todos ellos se encuentran conectados al suministro eléctrico en todo momento sin importar la cantidad de puestos en uso ya que la única forma para gestionar esta acción es mediante la desconexión manual del toma corriente por parte del usuario, lo cual no es una práctica habitual.

Dada la problemática se establecieron los siguientes requerimientos para el sistema:

- Poder generar registros digitales de las actividades realizadas en el laboratorio.
- Disponer de un método fácil de distribución de los registros.

- Facilitar el uso racional de la energía, al brindar una herramienta para manipular las conexiones eléctricas de los puestos de trabajo, desde el puesto del responsable del laboratorio.

6.2 Diseño del prototipo

El sistema consiste en una herramienta para el control y monitoreo de conexiones eléctricas e intensidad de iluminación en los puestos de trabajo, así mismo generar registros digitales de las actividades llevadas a cabo en el laboratorio y visualizar el contenido de éstos mediante un sitio web en una red de área local.

Tomando en consideración los requerimientos que debe tener el sistema, se optó por un diseño modular basado en cuatro módulos independientes e interconectados.

En la figura 12 se muestra el diagrama en bloques del sistema en general.

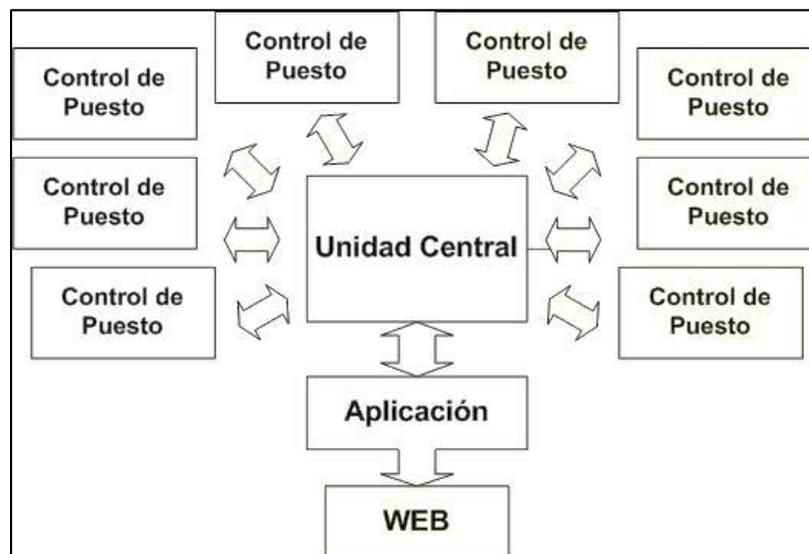


Figura 12. Diagrama general del sistema

En la figura 13 se puede observar un esquema general del sistema.

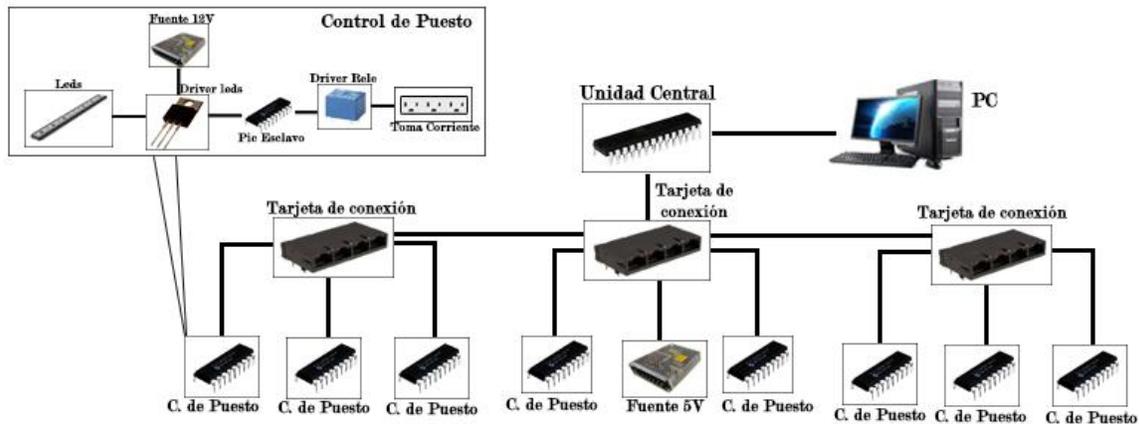


Figura 13. Esquema general del sistema

El módulo unidad central y los módulos control de puesto tienen una relación maestro-esclavo mediante el protocolo I2C, donde cada puesto dispone de una réplica del módulo control de puesto que hace la función de unidad remota que puede ser controlada desde la unidad central, brindando así escalabilidad ya que facilita el poder añadir más módulos, que un diseño unificado no permitiría.

La aplicación (APP) de escritorio alojada en un ordenador ofrece una interfaz gráfica del sistema, la cual se comunica con la unidad central mediante el protocolo USB, con la finalidad de enviar o recibir datos de las unidades remotas. A su vez, permite generar registros digitales del estado de los puestos y las actividades realizadas en el laboratorio, los cuales pasan a formar parte del módulo web. Este último módulo almacena los registros en un servidor local para que puedan ser visualizados desde un sitio web, al que se puede acceder mediante un explorador desde un dispositivo conectado en la misma red local, brindando una alternativa a la forma de distribución de la información basada en manuscritos.

El sistema tiene la característica de ser robusto ante errores o fallos ocurridos en algún módulo o interconexión, permitiendo continuar con la funcionalidad del resto del sistema siempre y cuando estas funciones no estén relacionadas o hagan uso del recurso que presenta el problema.

6.2.1 Módulo Unidad Central

Su objetivo principal es servir como centro de comunicación e intérprete (puerta de enlace) entre los módulos de control de puesto y la aplicación de escritorio, a su vez, da la posibilidad al usuario de enviar orden de apagar o encender las líneas de alimentación a las unidades remotas.

El módulo se diseñó usando un Microcontrolador PIC18F2550 de Microchip, se escogió este Microcontrolador, dentro de la amplia variedad, por las siguientes características:

- Fácil de conseguir en el mercado del país
- Diseño compacto de hardware para su implementación en diseños PCB¹³
- Módulo integrado para comunicación USB e I2C
- Cantidad de salidas/entradas suficientes
- Soporta ICSP
- Memoria ROM y RAM suficiente para la ejecución de las rutinas necesarias

Se integraron las conexiones de alimentación del sistema con la comunicación del protocolo I2C mediante un conector RJ45 para optimizar espacio.

¹³ Printed Circuit Board o Placa de Circuito Impreso

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

En la figura 14 se muestra el circuito propuesto.

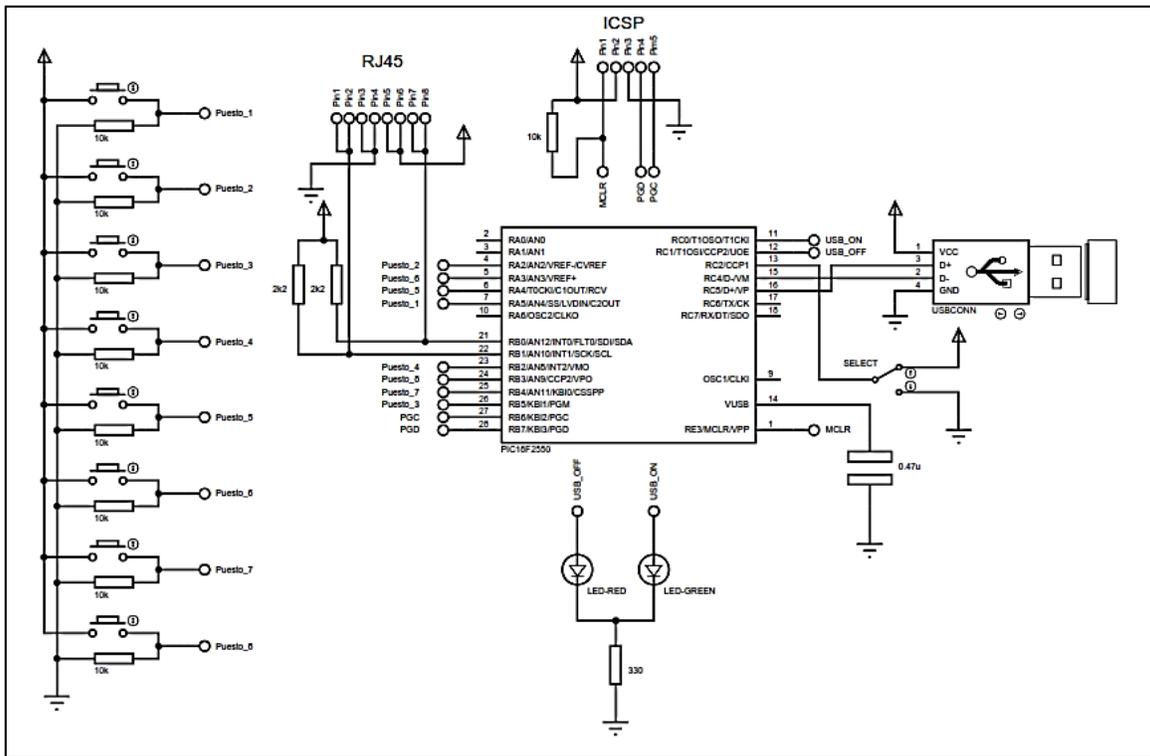


Figura 14. Circuito del módulo Unidad Central

En la figura 15 se muestra el diagrama en bloques del sistema el cual se encuentra en cuatro subsistemas.

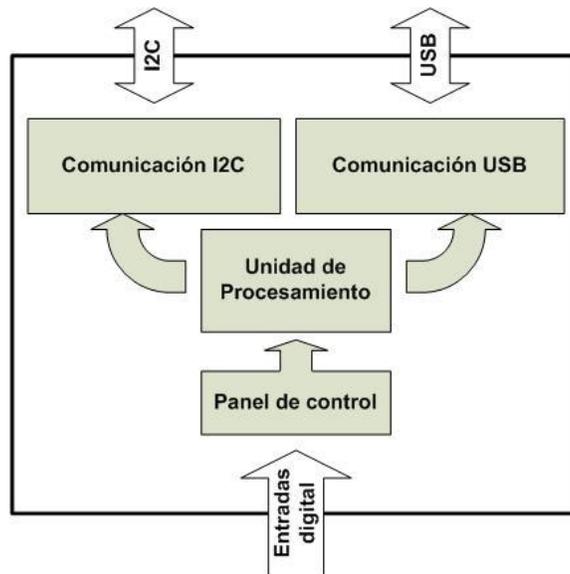


Figura 15. Esquema del módulo unidad central

6.2.1.1 Panel de Control

Proporciona una alternativa independiente al ordenador para el control de las conexiones eléctricas desde unidad central mediante un conjunto de pulsadores normalmente abiertos (N/A) para los puestos de trabajo. También dispone de un selector (Select) para escoger si el módulo usará la comunicación USB o no, esto evita la total dependencia de la aplicación de escritorio para controlar las unidades remotas. Posee 2 indicadores LED los cuales informan al usuario el estado de la alimentación del módulo y USB, éste último presenta dos estados, rojo para comunicación no establecida y verde caso contrario, véase figura 16.

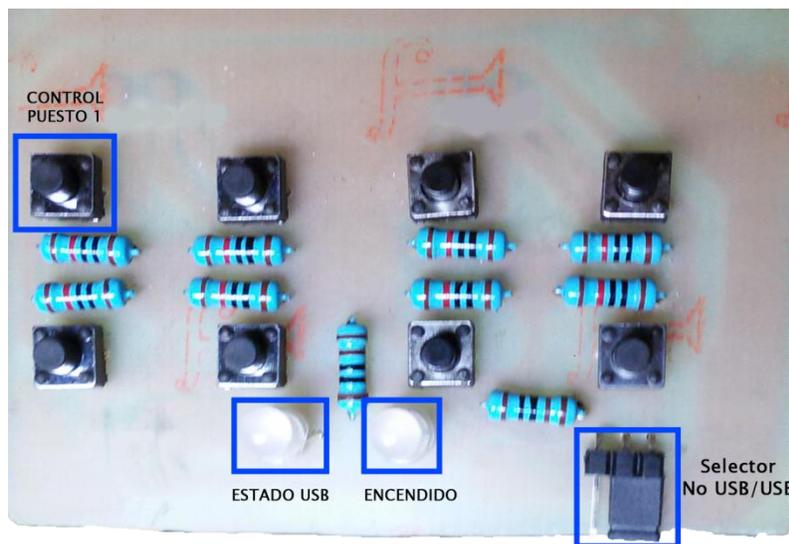


Figura 16. Vista frontal de panel de control microcontrolador maestro

6.2.1.2 Unidad de procesamiento

Inicialmente, la lógica del sistema lee el estado del selector, basándose en eso se decide si se va a inicializar o no la comunicación USB, en el estado en que no, el sistema solo queda ejecutando el funcionamiento de control para las líneas AC¹⁴ mediante el panel de control, en caso en que sí, inmediatamente inicializa la comunicación USB entre el ordenador y el dispositivo, espera la enumeración y ya cuando está lista, establece la comunicación, luego, pregunta si el ordenador ha

¹⁴ Alternating Current o Corriente Alterna

enviado datos, en caso que sí, lee e interpreta esos datos (dirección del esclavo y acción a ejecutar sobre ese mismo) y la envía al destinatario por medio del bus I2C.

Las acciones que se interpretan en esta unidad son:

- Refrescar
- Apagado total
- Encender/apagar línea AC en cualquiera de los puestos asociados
- Cambiar los valores de PWM¹⁵ en cualquiera de los puestos asociados

Las tareas que desempeña este subsistema a través del Microcontrolador PIC18F2550 son:

- Recibir, interpretar y reenviar los datos provenientes de los módulos a sus respectivos destinatarios mediante los protocolos pertinentes.
- Esperar y ejecutar la orden de control por parte del usuario en caso que se desee manipular la línea AC de alguna unidad remota específica.
- Inicializar la comunicación USB e I2C.
- Monitorear si hay datos provenientes del buffer USB por parte del ordenador.

¹⁵ Pulse-Width Modulation o Modulación por Ancho de Pulso

Los autores del proyecto proponen el diagrama de flujo de la figura 17 para la unidad de procesamiento.

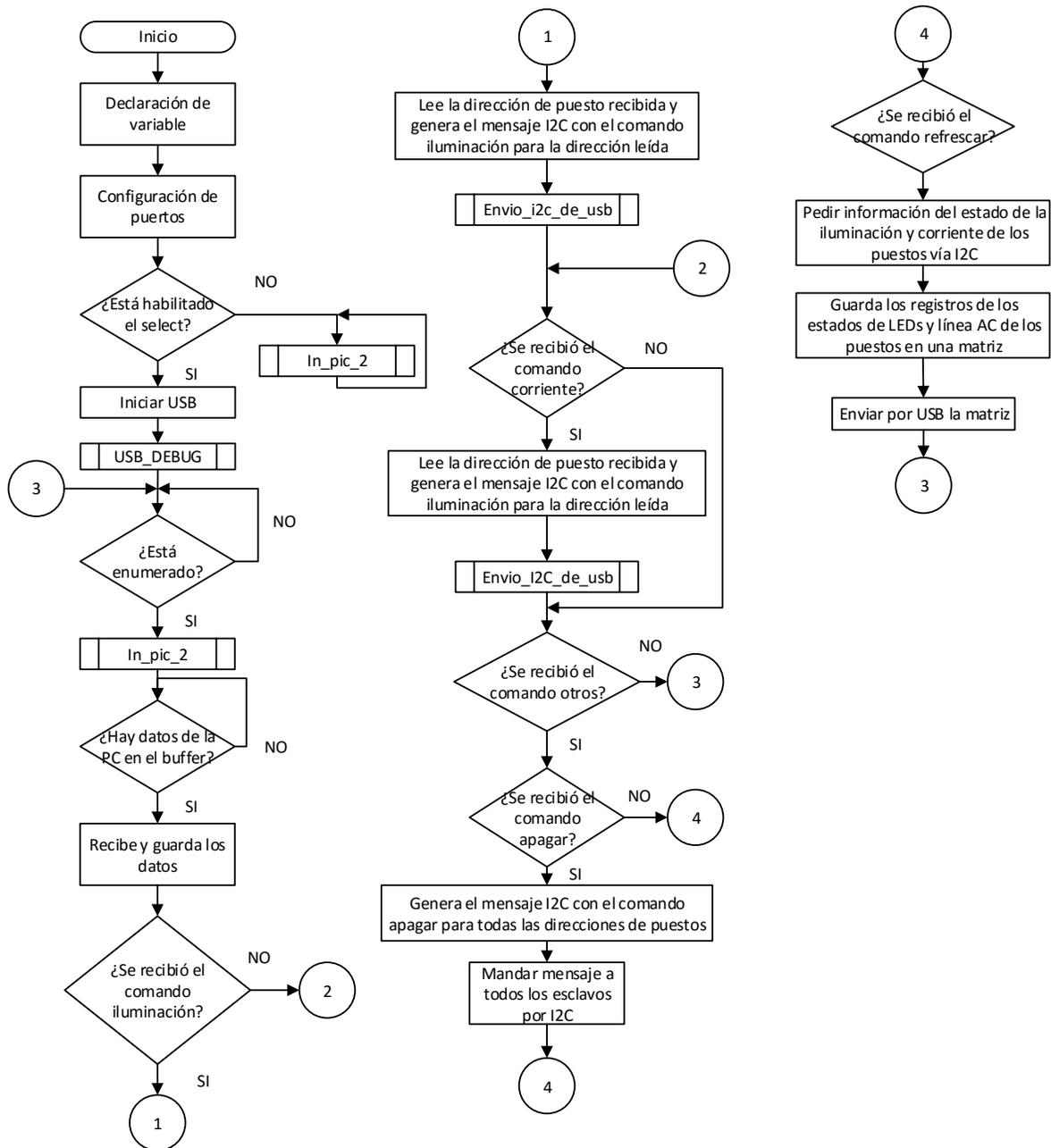


Figura 17. Diagrama de flujo propuesto para la unidad de procesamiento

Los autores del proyecto proponen el diagrama de flujo de la figura 18 para la interrupción de la unidad de procesamiento.

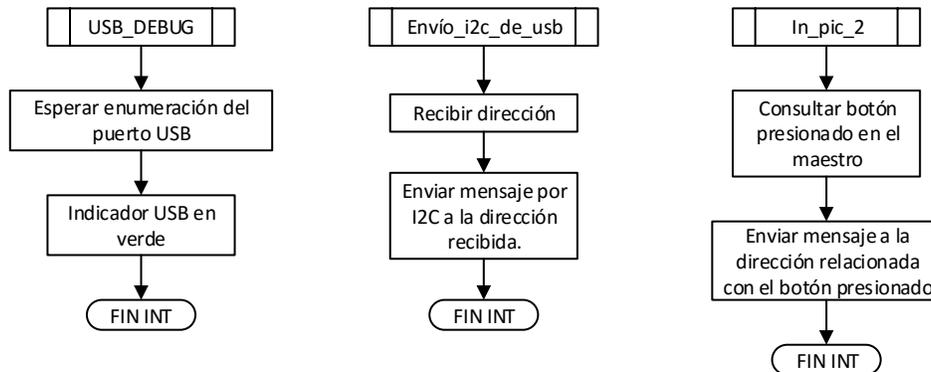


Figura 18. Diagrama de flujo propuesto para la interrupción de la unidad de procesamiento

6.2.1.3 Comunicación I2C

Es el encargado de la transducción entre las señales digitales enviadas desde la unidad de procesamiento a los módulos de unidades remotas.

Se utilizó este protocolo debido a que la aplicación no requiere el uso de una alta velocidad de transferencia, la cantidad máxima de esclavos que se pueden obtener de forma simultáneas (128) y permite el modo Multimaestro en caso que se requiera anexar uno o más maestros.

La integración de las dos líneas que usa el protocolo con la conexión de alimentación mediante un conector RJ45 también fue un factor importante ya que permite energizar y comunicar las unidades remotas usando la misma estructura de cable, permitiendo así construir líneas de cableado estructurado para tener una robusta distribución.

La dirección de los esclavos y los códigos de mensaje I2C pueden ser vistos en Anexo D.

6.2.1.4 Comunicación USB

Es la encargada de establecer comunicación con el módulo aplicación mediante el uso del protocolo USB 2.0.

Se optó por este protocolo debido a que es de 2 líneas, con altas tasas de transferencia, de conector compacto en comparación a otros protocolos (serial y paralelo), en mejora constante y actualmente la mayoría de los ordenadores cuentan con al menos un puerto para esta comunicación.

Se escogió la clase HID¹⁶ sobre las demás clases existentes por la principal razón que no requiere un driver específico que deba estar instalado en el ordenador, ya que utiliza la tecnología “plug and play”.

Para el manejo del driver USB se estudió y utilizó la librería mCHID.dll proporcionada por la empresa MecaniqueUK¹⁷ para el control de dispositivos USB mediante protocolo HID.

Con fines de diseño se empleó un conector USB hembra tipo B.

6.2.2 Módulo Control de Puesto

También llamado unidad remota, éste módulo se diseñó para ubicarse en cada puesto de trabajo permitiendo controlar los niveles de iluminación, conexión al suministro eléctrico y mantener un registro sobre el estado actual de éstos (PWM & Relé).

El control de los periféricos se puede hacer de dos (2) maneras, por software, mediante una orden desde la aplicación de escritorio recibida a partir de la unidad central mediante el bus I2C, o por hardware, usando dos botones de pulsación, uno para cada periférico.

¹⁶ Human Interface Device

¹⁷ Empresa desarrolladora de la librería para gestión del puerto USB usando clase HID, para mayor información visitar: <https://www.mecanique.co.uk/shop/>

El módulo se diseñó usando un Microcontrolador PIC16F88 de microchip, se escogió este Microcontrolador, dentro de la amplia variedad, por las siguientes características:

- Módulo PWM de 8 bits integrado
- Diseño compacto de hardware para su implementación en diseños PCB
- Módulo integrado para comunicación I2C (por software o hardware)
- Cantidad de salidas/entradas suficientes en comparación a la familia 12F
- Soporta ICSP
- Memoria ROM y RAM suficiente para la ejecución de las rutinas necesarias

En la figura 19 se muestra el diagrama en bloques de este módulo el cual está conformado por tres sub-sistemas:

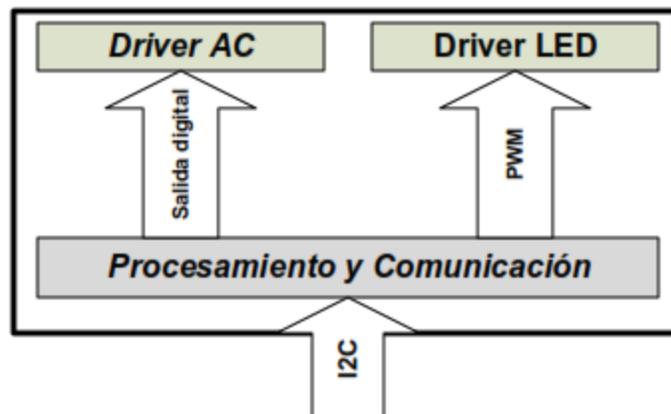


Figura 19. Diagrama en bloques del módulo de control de puesto

En la figura 20 se muestra una vista detallada del módulo control de puesto con sus indicadores y botones.

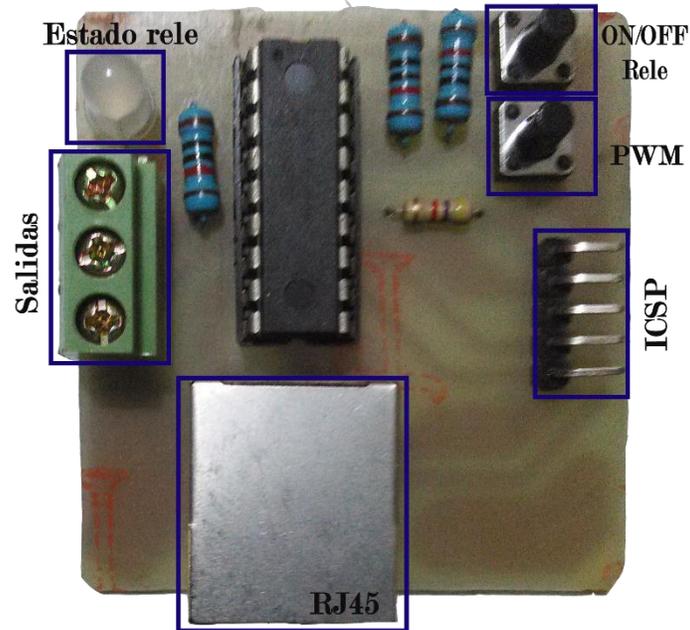


Figura 20. Vista frontal de módulo control de puesto

6.2.2.1 Procesamiento y Comunicación

Interpreta y responde las órdenes I2C provenientes de la Unidad Central, subsistemas Driver LED y AC.

Las tareas que desempeña este subsistema a través del Microcontrolador PIC16F88 son:

- Recibir y distribuir las órdenes provenientes del módulo unidad central.
- Recibir y ejecutar la orden de control por parte del usuario en caso que se desee manipular la línea AC o intensidad de iluminación del espacio de trabajo.
- Almacenar el estado de sus salidas.
- Indicar al usuario por medio de un LED el estado de la conexión a la línea AC.

En la figura 21 se muestra el circuito para este subsistema.

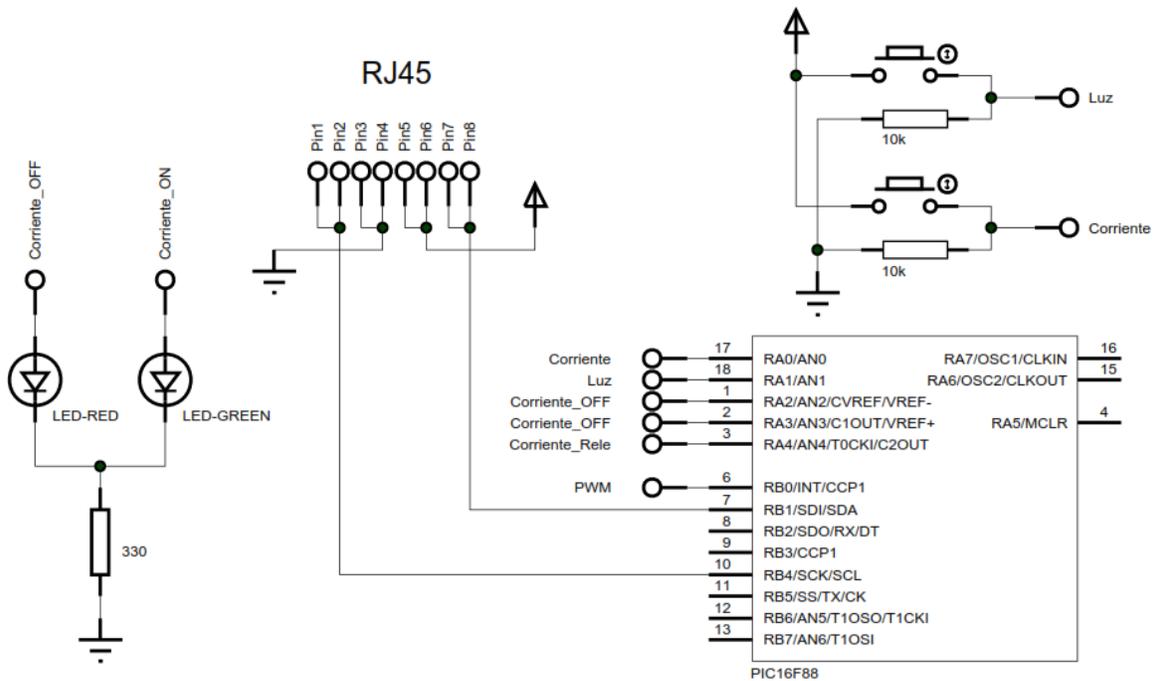


Figura 21. Circuito del módulo de control de puesto

Inicialmente la unidad espera la interacción del usuario en los botones, si se presiona el botón 1 por un tiempo programado, habilita la línea AC, si se mantiene presionado por más tiempo deshabilita la línea AC y envía a 0% la salida PWM. Cada vez que se presiona el botón 2, se modifica el ciclo de trabajo de la señal PWM con un ciclo de trabajo del 20%, 50%, 100% y 0 cambiando de uno a otro una vez por clic, en cada uno de estos casos, almacena el estado de la línea AC e intensidad de iluminación.

Posteriormente, si ninguna de las condiciones anteriores se ejecutó, el módulo entra en un estado de “espera” para que la unidad central envíe una orden vía I2C, en dependencia del código recibido, ejecuta la acción correspondiente.

Los autores del proyecto proponen los diagramas de flujo de las figuras 22 y 23 para el subsistema de procesamiento y comunicación.

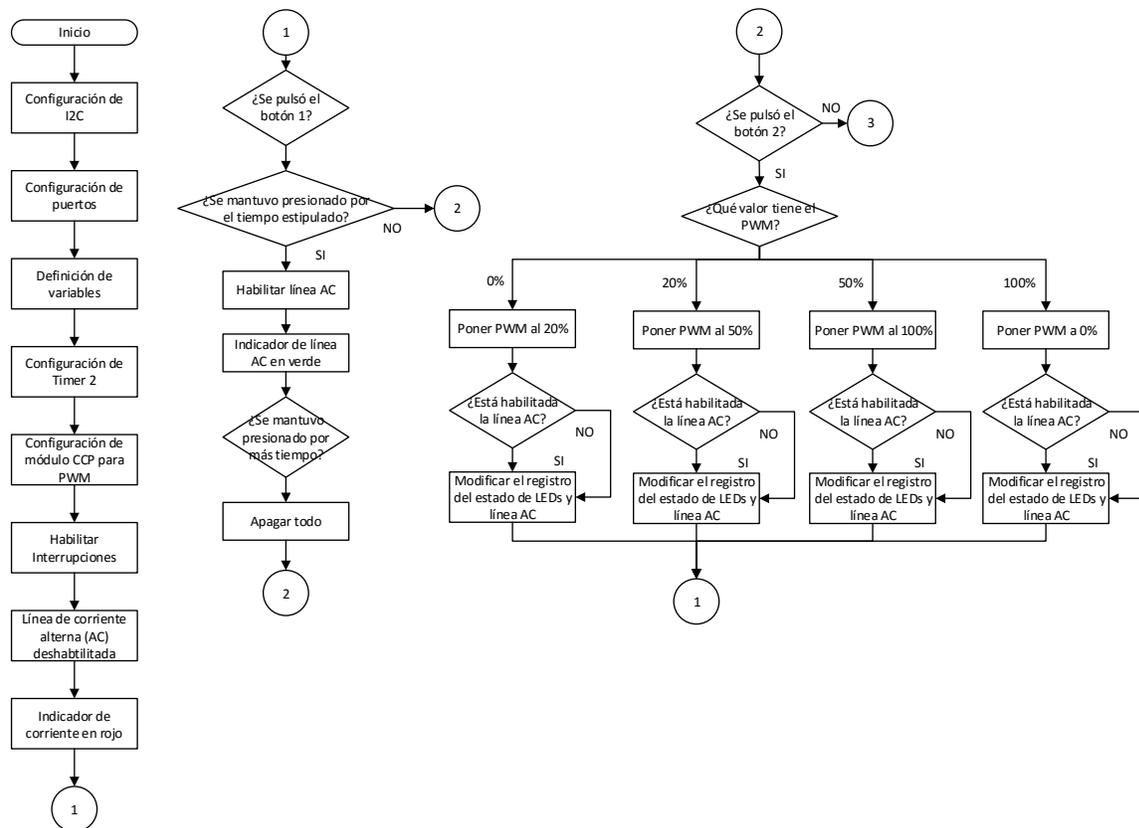


Figura 22. Diagrama de flujo propuesto para subsistema de procesamiento y comunicación I

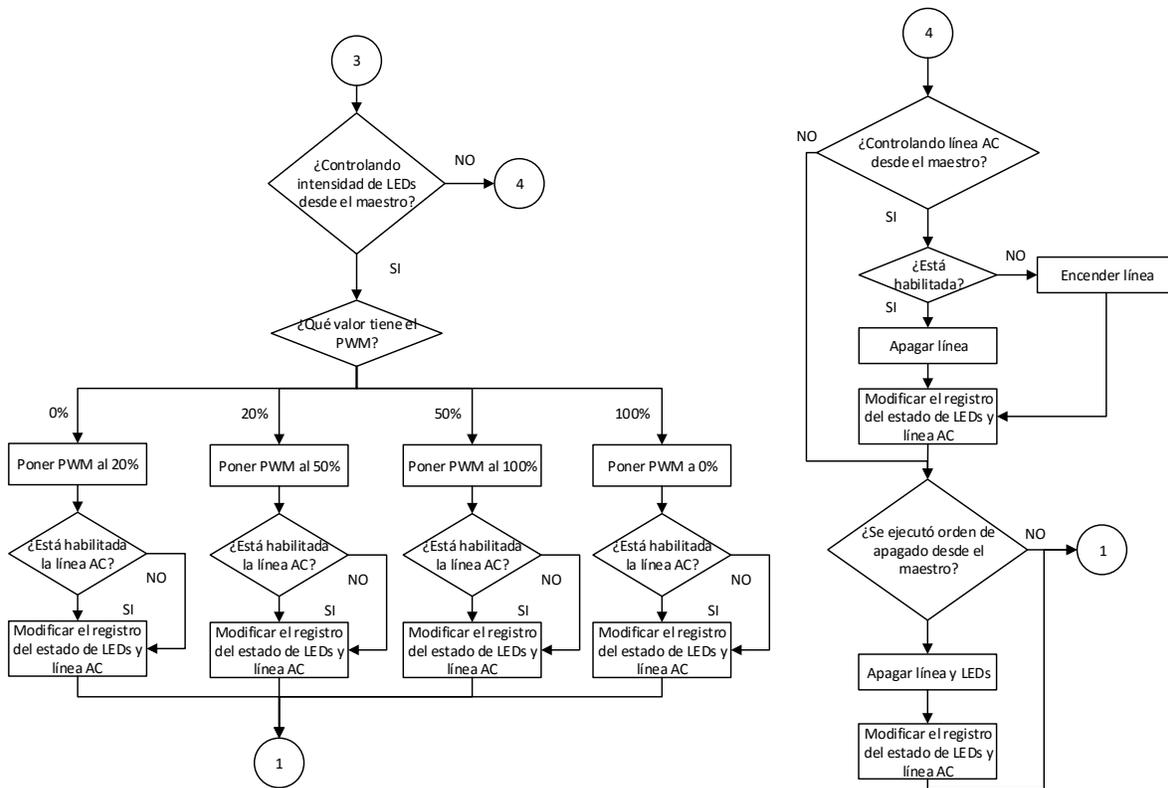


Figura 23. Diagrama de flujo propuesto para subsistema de procesamiento y comunicación II

Los autores del proyecto proponen el diagrama de flujo de la figura 24 para la interrupción del subsistema de la unidad de procesamiento y comunicación.

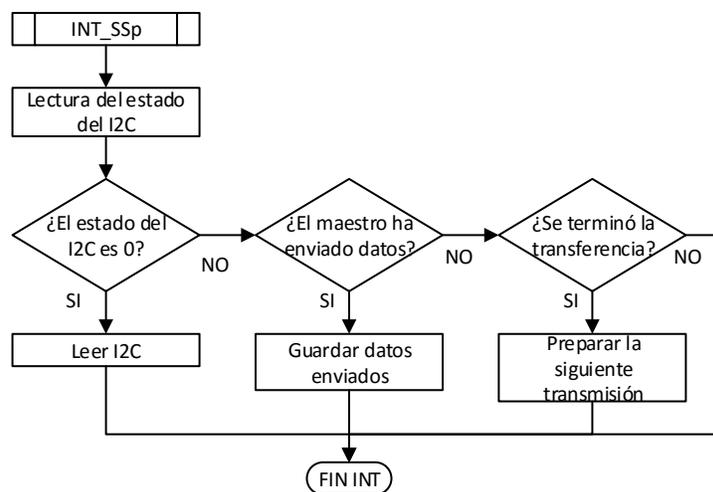


Figura 24. Diagrama de flujo propuesto para la interrupción del subsistema de procesamiento y comunicación

6.2.2.2 Driver AC

Tiene como objetivo manipular mediante un pulso de entrada, la conexión al suministro eléctrico utilizando un actuador de potencia.

Para la conexión de la corriente alterna en los espacios de trabajo se diseñó este driver basado en un relé accionado mediante un transistor bipolar de propósito general (2N3904), este último controlado desde su base mediante la salida del Microcontrolador. Al tener una salida de un uno lógico (5V) la corriente de base en el transistor hace que entre en saturación excitando la bobina del relé, el cual cambiará su estado en la salida y así poder controlar la carga.

Para este diseño, se utilizó un relé con capacidad de 10 A y 125V AC en los contactos de fuerza, la implementación de este permite usar otro dispositivo terminal como un contactor, en caso que se requiera manipular una carga de mayor potencia. En la figura 25 se muestra el circuito del driver para corriente alterna.

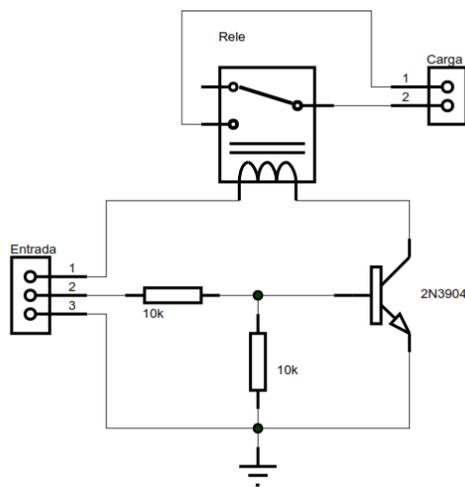


Figura 25. Circuito de driver para corriente alterna

6.2.2.3 Driver LED

Su propósito es permitir la variación de la intensidad en la iluminación de los puestos recibiendo una entrada digital proveniente del microcontrolador 16F88 para manipular la alimentación de un conjunto LED de alta luminosidad. Para esto se utilizó el esquema mostrado en la figura 26, que dispone de un MOSFET y un transistor bipolar el cual regula la corriente que pasa por los LEDs conectados de

manera serial. Cuando el circuito está funcionando, el transistor trabaja y la puerta (Gate) del MOSFET recibe una tensión positiva a través de la resistencia de 4,7K que lo hace conducir, encendiendo los LEDs.

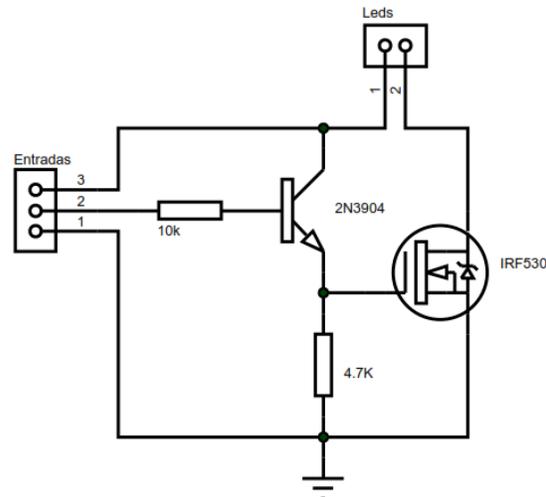


Figura 26. Circuito del driver LED

Se usó tiras LEDs SMD¹⁸ modelo 5050 debido a la versatilidad de poder manipular la intensidad de luz, alta potencia, buenas prestaciones en comparación a otros modelos (SMD 3528 o 3014), bajo costo y consumo energético.

En la tabla 1 podemos observar las comparativas en los diferentes modelos:

Modelo	Led/Metro	W/Metro	Lm/Metro	Tamaño
SMD3528	60 Led/m	4,8W/m	4Lm/m	9x6mm
SMD3014	120 Led/m	12W/m	8Lm/m	10x6,5mm
SMD5050	60 Led/m	14,4W/m	12Lm/m	10x6,5mm
SMD5050 High Power	60 Led/m	14,4W/m	20Lm/m	14x7mm

Tabla 1. Especificaciones técnicas de modelos

¹⁸ Surface Mounted Device o Dispositivo de Montaje Superficial

6.2.3 Aplicación

Es una aplicación de escritorio desarrollada en C# utilizando el IDE¹⁹ Visual Studio de Microsoft²⁰ instalada en un ordenador la cual permite un mayor control sobre los puntos de energía eléctrica habilitados generando un impacto positivo en la reducción de tiempo que se demora el usuario en preparar una práctica o desconectar todos los equipos de los puestos de trabajo al terminar la jornada del día.

Las tareas que desempeña son:

- Brindar una interfaz gráfica para el control y monitoreo de las conexiones eléctricas e iluminación de los puestos de trabajo.
- Permitir al usuario generar registros de las actividades que se llevan a cabo en el local.
- Guardar en un directorio estipulado los registros generados por el usuario.
- Leer, interpretar y mostrar los datos recibidos por la unidad central.
- Gestionar los recursos necesarios para enlazar la comunicación USB con la unidad central.

Utiliza la clase EasyHID.cs para definir las funciones de comunicación que utiliza el dispositivo USB y que proveen al ordenador de las mismas, las funciones más importantes en la aplicación son las siguientes:

Conecta la aplicación al controlador

```
public static extern bool Connect(IntPtr pHostWin);
```

Desconecta la aplicación al controlador

```
public static extern bool Disconnect();
```

Lee el buffer de entrada para monitorear si hay datos provenientes del dispositivo, el tamaño del buffer depende de la constante BUFFER_IN_SIZE, en este caso es de 32 bytes pero en la aplicación solo se usó 8 bytes para la transferencia.

```
public static extern bool Read(UInt32 pHandle, IntPtr pData);
```

Escribe en el buffer de salida con los datos almacenados en la variable pData, que es un puntero a un buffer no administrado. El tamaño del buffer depende de la

¹⁹ Integrated Development Environment o Ambiente de Desarrollo Integrado

²⁰ Gratis para estudiantes en su sitio web: <https://www.dreamspark.com/>

constante `BUFFER_OUT_SIZE`, en este caso es de 32 bytes, la aplicación solo utiliza 3 bytes de éste, la función retorna 1 si se envió el reporte y 0 en caso contrario.

```
private static extern bool Write(UInt32 pHandle, IntPtr pData);
```

Con éstas funciones se obtiene el `vendorID` & `productID`

```
public static extern UInt32 GetVendorID(UInt32 pHandle);
```

```
public static extern UInt32 GetProductID(UInt32 pHandle);
```

Activamos las notificaciones para recibir mensajes cada vez que se produce un evento de lectura. El servicio se activa una vez que el dispositivo ya se ha conectado al controlador.

```
public static extern void SetReadNotify(UInt32 pHandle, bool pValue);
```

Obtenemos el estado de conexión del dispositivo mediante el `vendorID` y `productID`

```
public static extern bool IsAvailable(UInt32 pVendorId, UInt32 pProductId);
```

Esta es la función lee el buffer de entrada para ver si hay datos disponibles. En primera instancia asigna memoria con un tamaño igual al `BUFFER_IN_SIZE` a un puntero:

```
public static bool Read(UInt32 pHandle, out byte[] pData)
{
    IntPtr unmanagedBuffer = Marshal.AllocHGlobal(BUFFER_IN_SIZE);
    bool result = Read(pHandle, unmanagedBuffer);

    try { pData = new byte[BUFFER_IN_SIZE]; Marshal.Copy(unmanagedBuffer,
pData, 0, BUFFER_IN_SIZE); }
    finally { Marshal.FreeHGlobal(unmanagedBuffer); }

    return result;
}
```

Esta función escribe un dato en el buffer de salida

```
public static bool Write(UInt32 pHandle, byte[] pData)
{
    IntPtr unmanagedBuffer = Marshal.AllocHGlobal(BUFFER_OUT_SIZE);
    bool result;

    try { Marshal.Copy(pData, 0, unmanagedBuffer, BUFFER_OUT_SIZE); result =
Write(pHandle, unmanagedBuffer); }
    finally { Marshal.FreeHGlobal(unmanagedBuffer); }

    return result;
}
```

Básicamente la aplicación intercepta los mensajes de Windows a la espera de algún evento, haciendo la intercepción de un mensaje y realizando una operación en dependencia del valor recibido teniendo tres estados:

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

- Dispositivo_conectado
- Dispositivo_desconectado
- Lee_dispositivo

Lo cual se programó una función para cada uno de esos eventos, así se podrá realizar diferentes acciones en consecuencia a la notificación interceptada.

```
private void Dispositivo_Conectado(UInt32 handle){}
private void Dispositivo_desconectado(UInt32 handle) {}
private void Lee_datos(UInt32 In_handle) {}
```

Para interconectar la aplicación al controlador se utiliza la función:

```
EasyHID.Connect(Handle);
```

Y para corroborar que realmente el dispositivo se conectó se utiliza la función:

```
if (EasyHID.IsAvailable(EasyHID.VENDOR_ID, EasyHID.PRODUCT_ID) == true)
```

Ya que se detecta cualquier dispositivo conectado que coincide con el VID²¹ & PID²² declarado en la aplicación.

En la figura 27 se muestra la interfaz de la aplicación.

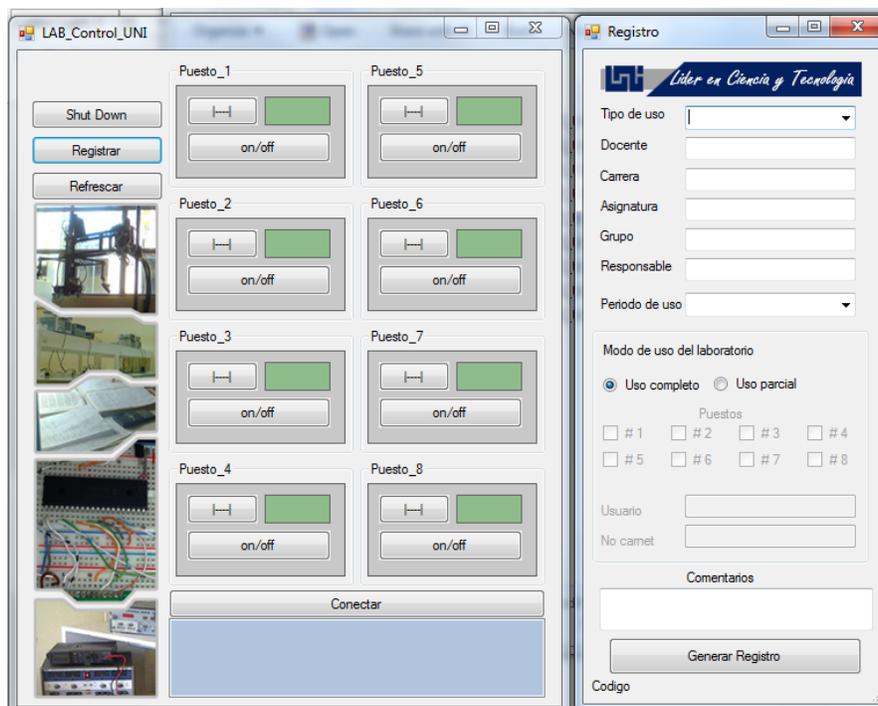


Figura 27. Interfaz gráfica de la aplicación

²¹ Vendor ID
²² Product ID

Está conformada por dos ventanas, ventana control y ventana registro. En la ventana de control se puede interactuar con las unidades remotas mediante el uso de botones e indicadores visuales, véase tabla 2 y anexo A para mayor referencia.

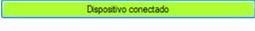
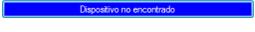
Nombre	Imagen de referencia	Descripción
Conectar		Inicializa la comunicación entre el microcontrolador maestro con el ordenador, y a su vez refleja el estado de la conexión USB.
		Indica la correcta comunicación entre la aplicación y el microcontrolador maestro.
		Indica que no se ha logrado establecer conexión con el microcontrolador maestro.
		Indica la pérdida de comunicación de la aplicación con el microcontrolador maestro.
Apagar		Ordena al microcontrolador maestro enviar un mensaje con la orden "apagar" a todos los esclavos
Registrar		Abre la ventana del formulario de registro.
Refrescar		Consulta el estado de las salidas en los esclavos y actualiza la interfaz.
Interfaz de puesto		Indica la correcta comunicación con el esclavo relacionado a ese puesto, permitiendo la interacción con los botones.
		Indica la falta de comunicación con el esclavo relacionado a ese puesto, deshabilitando la interacción con los botones.
Botón Iluminación		Cambia el valor del PWM en la salida del microcontrolador esclavo con el que esté relacionado.
Indicador de Iluminación		Indica un valor de PWM a 0% en la salida del microcontrolador esclavo con el que este relacionado.
		Indica un valor de PWM a 20% en la salida del microcontrolador esclavo con el que este relacionado.
		Indica un valor de PWM a 50% en la salida del microcontrolador esclavo con el que este relacionado.
		Indica un valor de PWM a 100% en la salida del microcontrolador esclavo con el que este relacionado.
Boton alimentación		Habilita o deshabilitar la salida al relé del microcontrolador esclavo con el que esté relacionado .
Indicador de alimentación		Indica un "1" lógico en la salida al relé del esclavo con el que esté relacionado.
		Indica un "0" lógico en la salida al relé del esclavo con el que esté relacionado .
Cuadro de acciones		Refleja un registro temporal de acciones realizadas.

Tabla 2. Descripción de interfaz gráfica

Los botones se encargan de enviar las órdenes para la manipulación de las unidades remotas y los indicadores permiten visualizar el estado de estas mediante cambios de colores, para ello se hace uso de valores hexadecimales para asociar los puestos y las acciones con la interfaz.

La tabla 3 muestra los tipos de mensajes ocupados en la comunicación USB.

Tipo de mensaje	Trama	Descripción
Dirección del puesto	0x01	Dirección asociada al puesto 1.
	0x02	Dirección asociada al puesto 2.
	0x03	Dirección asociada al puesto 3.
	0x04	Dirección asociada al puesto 4.
	0x05	Dirección asociada al puesto 5.
	0x06	Dirección asociada al puesto 6.
	0x07	Dirección asociada al puesto 7.
	0x08	Dirección asociada al puesto 8.
Comando de control	0x0A	Comando de acción asociado a la salida del relé.
	0x0B	Comando de acción asociado a la variación del PWM.
	0x0C	Comando de acción asociado a acciones varias.
Sub-comando de control	0x0F	Sub comando de acción asociado a la consulta de los estados de las unidades remotas.
	0xF0	Sub comando de acción asociado al apagado general de las unidades remotas.

Tabla 3. Mensajes USB

La ventana registro está conformada por un formulario para la adquisición de los principales datos relacionados al uso y actividades realizadas en el laboratorio, teniendo los siguientes campos de llenado:

- Tipo de uso: Dedicado para especificar el uso del laboratorio, consta de las opciones predeterminadas: Práctica de laboratorio, reposición de laboratorio, desarrollo de proyecto, práctica independiente y extracurricular, así mismo brinda la opción de ingresar un título diferente a estas opciones con la finalidad de poder adecuarse a cualquier actividad realizada.
- Docente: En este campo se especifica el nombre del docente encargado u organizador de la actividad a realizar.
- Carrera: Permite especificar la carrera con la que esté relacionada la actividad
- Asignatura: permite indicar la relación del evento llevado a cabo en el local con alguna materia académica.
- Grupo: relaciona las actividades con un grupo de clase.
- Periodo de uso: brinda un aproximado del tiempo requerido o solicitado para la elaboración de la actividad.
- Modo de uso del laboratorio: esta sección permite la realización de registro de actividades en la cual se indica si se hace uso total o parcial del laboratorio.

- Comentarios: Este campo es una sección abierta donde se puede hacer registro de cualquier acontecimiento o eventualidad que requiera formar parte del registro, así mismo puede hacerse uso de este campo para anotar los préstamos de equipos que serán utilizados en actividades fuera del laboratorio.

En esta ventana se generan archivos digitales de texto plano (*.txt) en los que se almacena la información introducida en el formulario bajo nombres relacionados con hora, fecha y el texto introducido o seleccionado en el campo “tipo de uso” para disponer de un código único para cada registro, el cual puede ser visualizado en la parte inferior de la ventana al momento de generar el registro.

6.2.4 Módulo WEB

Este módulo tiene como objetivo brindar un forma de visualizar y distribuir los registro generados por la aplicación de escritorio, para ello está conformado por un servidor local donde se almacenan estos, y un sitio WEB que es alojado en él.

Existen variarías aplicaciones que permiten montar un servidor sobre un ordenador, entre las que se encuentra “AppServ 2.10.3”, el cual es un software gratuito²³, libre de uso, totalmente configurable y de sencilla instalación en Windows, Linux/Unix y Macintosh, contando con los siguientes paquetes:

- Apache WEB Server 2.2.8
- Lenguaje PHP 6.0.0
- MySQL Data Base 6.0.4-alpha
- Manejador de base de datos phpMyAdmin 2.10.3

Se realizó uso del AppServ para disponer de un servidor ejecutado en un ordenador de sistema operativo Windows 7, siendo este sistema operativo el más implementado en las computadoras del laboratorio.

El sitio web permite mostrar el contenido de los registros almacenados en el servidor, donde la interacción con el sitio es mediante los navegadores web de dispositivos en la misma red local. La página web dispone de hipervínculos

²³ Comunidad AppServ Open Project <http://www.appservnetwork.com/index.php>

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

relacionados a los meses del año mediante los cuales se puede acceder a las listas de registros, permitiendo abrirlos desde el mismo navegador a como se puede observar en la figura 28. Para el desarrollo de este sitio se hizo uso de los lenguajes HTML5, CSS y PHP, los que nos permitieron darle organización estética y funcionalidad al sitio WEB.

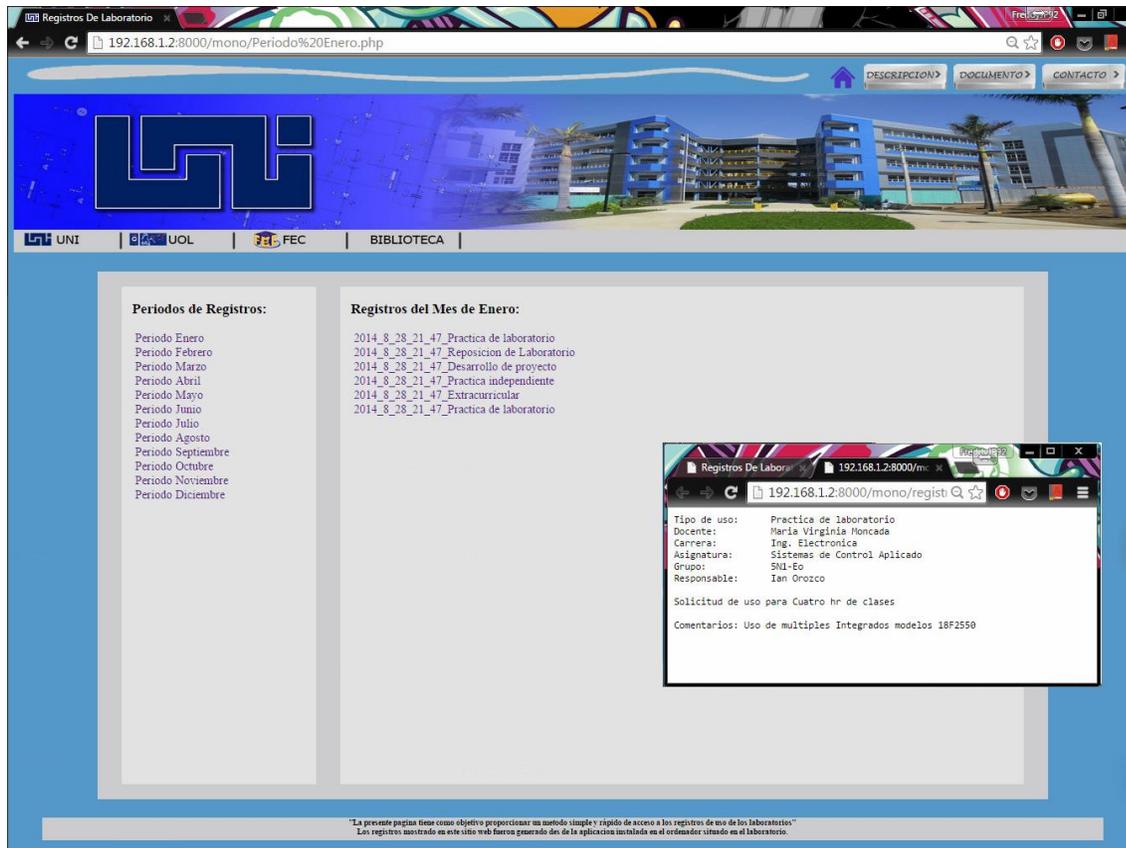


Figura 28 Captura de pantalla de página WEB

La lógica usada en los segmentos de código PHP permite relacionar los textos presentes en la página y los títulos, con los documentos digitales donde se encuentran redactados los registros, esto se puede observar en la figura 29.

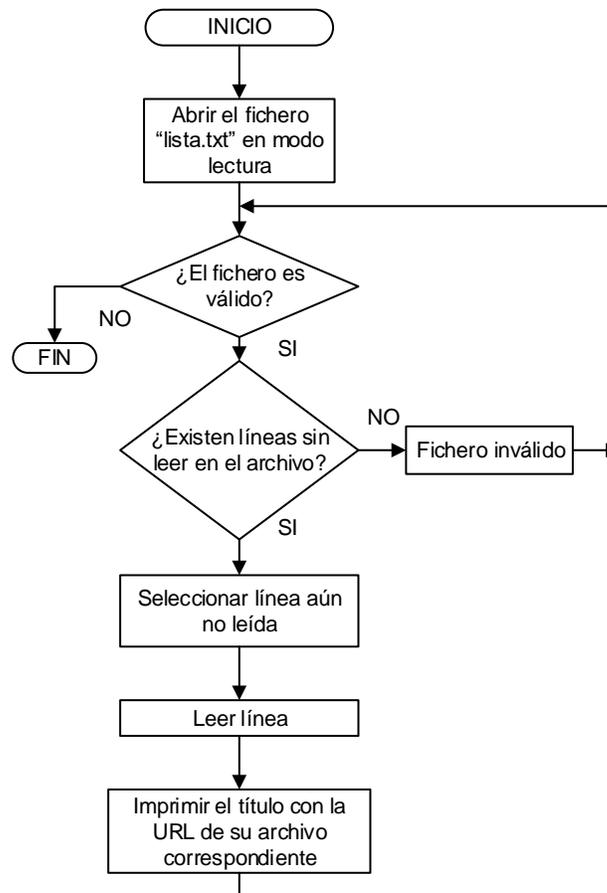


Figura 29. Diagrama de flujo propuesto para el módulo WEB

6.3 Prueba del diseño propuesto

Una vez los diferentes módulos cumplieron los requerimientos para lo que fueron diseñados. Se procedió a realizar pruebas del sistema en general, evaluando todas las características de los módulos así como la perfecta comunicación entre ellos.

En el montaje del hardware de los módulos control de puesto y unidad central. Se desarrolló en tabla de nodo con seis ejemplares del módulo control de puesto y un módulo unidad central. También se realizaron pruebas de distancias con cables de 11.75 metros y no se obtuvo ningún resultado negativo.

Para las pruebas del software y página web se hizo uso de una laptop con sistema operativo Windows 7 64 bits, la aplicación no presentó ningún inconveniente en su funcionamiento probándose en 3 ordenadores distintos.

El driver LED funcionó a su perfección sin ningún calentamiento luego de conectarle una tira de 5 metros de LEDs de alta luminosidad lo cual brinda seguridad en su implementación.

Posteriormente se realizó el montaje de los módulos en PCB, obteniendo resultados satisfactorios utilizando un módulo de unidad central, dos unidades remotas, una fuente de alimentación de 12V, una fuente de alimentación de 5V, dos drivers LEDs y dos drivers relé.

Se verificó el control desde la aplicación de los ocho puesto de trabajo, la buena creación de los registros digitales y el funcionamiento de la página web al momento de mostrar estos registros obteniendo resultados satisfactorios en relación a tiempos de respuestas en base a una petición del usuario.

6.4 Análisis y corrección de errores

Se hicieron ligeras modificaciones a los diseños PCB para mejorar la presentación del prototipo y agregándole los pines para utilizar ICSP, se anexó una placa más para poder interconectar las unidades remotas y las fuentes de alimentación para la parte de control.

Se hizo una depuración a la aplicación ya que presentaba muchos problemas al ejecutar la acción “refrescar”, mejorando los métodos de refrescamiento.

Se mejoró la estética de la aplicación y página web.

Los códigos y diseños de circuitos impresos con los que se obtuvieron los resultados esperados posterior a la corrección de errores pueden ser vistos en Anexo B.

Los precios para la fabricación de las unidades implementadas para el proyecto monográfico pueden ser vistos en Anexo C.

7 Conclusión

A lo largo del desarrollo del trabajo monográfico se llegaron a las siguientes conclusiones:

- Desde la perspectiva de los autores, el prototipo responde a la necesidad planteada para la gestión de un laboratorio de electrónica en la Universidad Nacional de Ingeniería ya que recrea de forma real varias funcionalidades realizadas por el responsable de laboratorios, planteados al inicio del trabajo monográfico.
- El desarrollo de una interfaz gráfica para el control de iluminación y toma corrientes que disponen los puestos de trabajo permite disminuir el tiempo que demora el encargado en preparar una práctica o desconectar todos los equipos de las áreas de trabajo al terminar la jornada del día.
- El sistema da una alternativa para la creación y visualización de los registros de actividades realizadas en el laboratorio.
- La realización de este trabajo monográfico, permitió desarrollar nuevas competencias de aprendizaje aplicadas en el proyecto, como es la programación orientada a objetos, desarrollo web, entre otras.

8 Recomendaciones

A lo largo del desarrollo del trabajo monográfico se identificaron las siguientes recomendaciones:

- La implementación de una comunicación inalámbrica quitaría la dependencia del cableado en las unidades remotas y permitiría la movilidad libre de estas.
- El uso de software para base de datos permitiría agregar funcionalidades al sistema.
- Un mayor desarrollo de la aplicación mediante Visual Studio permitiría agregar funcionalidades para la implementación de una aplicación web.
- Desarrollar una aplicación móvil con las funcionalidades de la aplicación ejecutada en el ordenador permitiría movilidad al usuario.
- Anexar una función para determinar la carpeta en la cual se guardarán los registros.
- Generar un ejecutable (.exe) para la instalación de la aplicación y sus complementos.
- Asociar la aplicación directamente con el software servidor.
- Agregar seguridad a la página web.
- Cambiar el uso de texto plano por hojas de cálculo o semejantes.
- Implementar un método de resumen para presentar informes de los registros generados.

9 Referencias

- ALEGSA. (s.f.). *ALEGSA*. Obtenido de <http://www.alegsa.com.ar/Dic/cliente%20servidor.php>
- AppServNetwork. (s.f.). *App Serv Open Project*. Obtenido de <http://www.appservnetwork.com/index.php?newlang=spanish>
- Axelson, J. (2009). *USB Complete The Developer's Guide* (Fourth ed.). Madison, Wisconsin: Lakeview Research LLC.
- García Breijo, E. (2008). *Compilador C CCS y simulador Proteus para microcontroladores PIC*. Barcelona, España: Alfaomega.
- Hogan, B. P. (2013). *HTML 5 and CSS3* (Second ed.). (S. D. Pfalzer, Ed.) Dallas, Texas: The Pragmatic Bookshelf.
- I2C communication between PICs*. (13 de Mayo de 2013). Obtenido de Neuroscience and Robotics Lab: http://hades.mech.northwestern.edu/index.php/I2C_communication_between_PICs
- Latin America Journals Online*. (2014). Obtenido de JOL: <http://www.lamjol.info/index.php/index/search/advancedResults>
- Londoño Ospina, N., Barrera Durango, M., Carvajal, J., & Fonseca, A. (29 de Mayo de 2012). *Scielo*. Obtenido de Scientific Electronic Library Online: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-62302012000200011&lng=es&nrm=.pf&tlng=es
- Microchip. (2001). *I2C Master Mode*. Recuperado el 13 de mayo de 2014, de <http://www.microchip.com/>: <http://ww1.microchip.com/downloads/en/DeviceDoc/i2c.pdf>
- Norfi Carrodegua. (s.f.). *Norfipc*. Obtenido de <http://norfipc.com/internet/instalar-servidor-apache.html>
- NORMA Oficial Mexicana NOM-025-STPS-2008, Condiciones de iluminación en los centros de trabajo*. (23 de Febrero de 2013). Obtenido de Diario oficial de la federación: http://dof.gob.mx/nota_detalle.php?codigo=5076393&fecha=30/12/2008
- Portal de publicaciones científicas y técnicas*. (2014). Obtenido de PPCT: <http://ppct.caicyt.gov.ar/index.php/index/search/advancedResults>
- Sistema regional de información en línea para revistas científicas de América Latina, el Caribe, España y Portugal*. (2014). Obtenido de Latindex: <http://www.latindex.unam.mx/busAva/busAva.html>

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

The Apache Software Foundation. (s.f.). *Apache HTTP Server Project*. Obtenido de http://httpd.apache.org/ABOUT_APACHE.html

The World Wide Web Consortium. (1995-2003). Obtenido de <http://www.w3.org/MarkUp/historical>

Verle, M. (13 de Mayo de 2008). *PIC Microcontrollers*. Obtenido de mikroElektronika: <http://www.mikroe.com/chapters/view/6/>

Anexos

- **Anexo A:** Manual de la aplicación
- **Anexo B:** Circuitos impresos
- **Anexo C:** Costo del prototipo
- **Anexo D:** Tablas

Anexo A: Manual de la aplicación

A.1. Manual de aplicación y página web

1. Encender la computadora cliente-servidor
2. Buscar la aplicación llamada "APP_Mono_Control_I2C_USB_WEB" en el escritorio y dar doble clic:



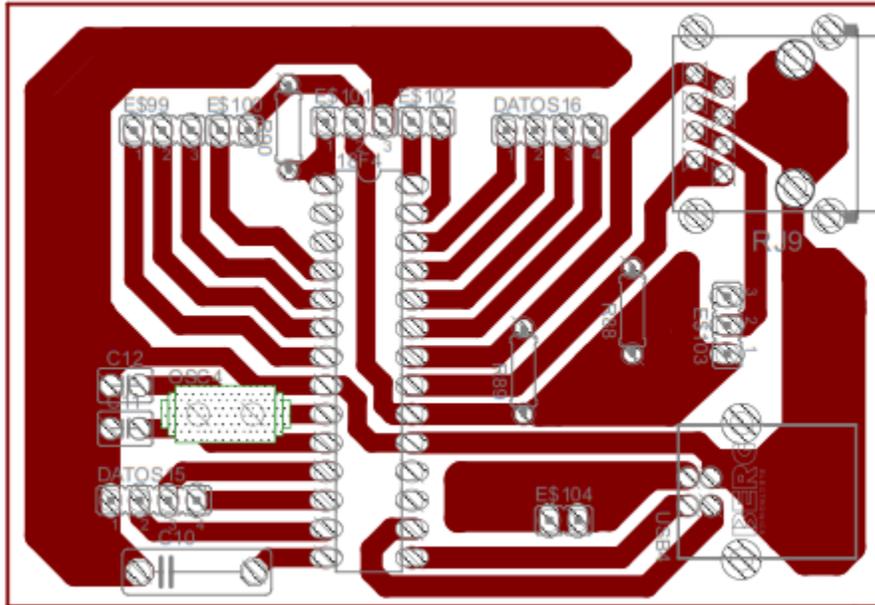
3. Apretar el botón **conectar** dentro de la aplicación
4. Presionar el botón **registrar** dentro de la aplicación
5. Llenar el formulario, seleccionar los puestos a utilizar y dar clic al botón **Generar registro**
6. Configurar los puestos de trabajo según la actividad que se va a realizar, tendrá 10 minutos para ello
7. Abrir el navegador de su preferencia (Chrome, Firefox, Microsoft Edge, entre otros), y en la barra de búsqueda escriba, 192.168.1.2:8000/mono
8. Seleccione el mes que quiere visualizar el registro, de clic al registro deseado
9. Se abrirá una nueva ventana con los datos del registro seleccionado
10. Al finalizar la jornada laboral, cierre la aplicación y el navegador
11. Apague el ordenador

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

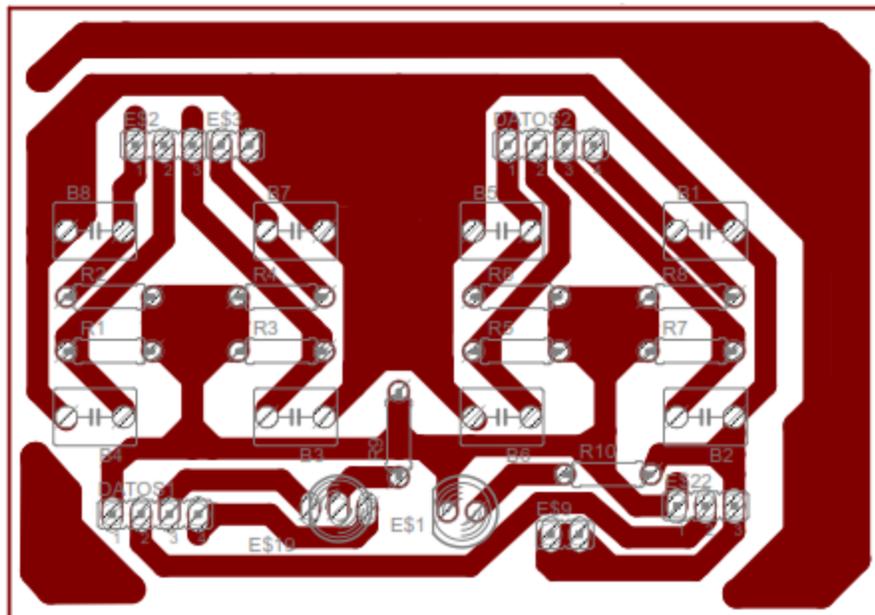
Anexo B: Circuitos impresos

B.1. Circuito impreso de la unidad central.

Parte baja:

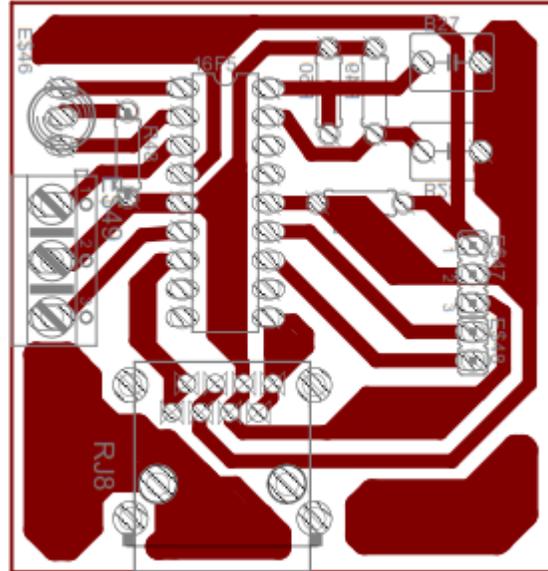


Parte alta:

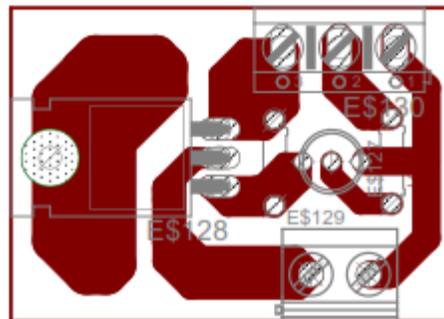


Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

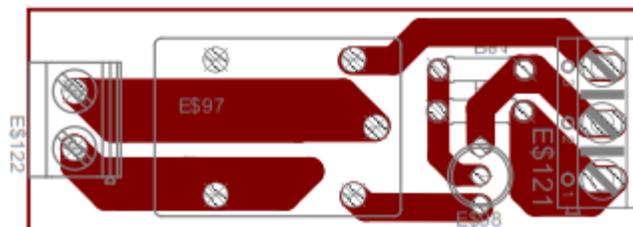
B.2. Circuito impreso de las unidades remotas.



B.3. Circuito impreso del driver LED.

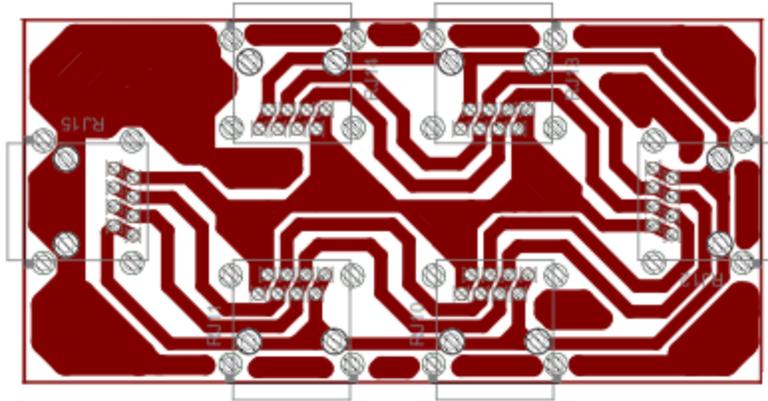


B.4. Circuito impreso del driver Relé.



Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

B.5. Circuito impreso placa conexión.



Anexo C: Costo del prototipo

C.1. Detalle del costo de las unidades control de puesto y unidad central del prototipo.

Las tablas 1, 2, 3, 4, 5 y 6 detallan los componentes utilizados con su costo, tanto, para la elaboración del prototipo, no se incluyen otros factores como tiempo, bibliografía, herramientas de diseño, costo de envío, entre otros, que influyen en el verdadero costo de desarrollo.

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
Push button	8	5.50	44
Resistencia 10k Ω	8	2	16
Resistencia 1k Ω	2	2	4
Resistencia 2.2k Ω	2	2	4
Resistencia 4.7k Ω	1	2	2
Led 3mm verde	1	5	5
Led 3mm RGB	1	15	15
Selector	2	15	30
RJ 45 hembra	1	20	20
USB tipo B hembra	1	20	20
Pin header 40 pines	1	10	10
Capacitor 0.47 μ F	1	10	10
Capacitor 22pF	2	10	20
Cristal 4 Mhz	1	40	40
PIC18F2550	1	143.5	143.5
Base DIP 28 pines	1	17.5	17.5
Tarjeta de cobre 5x7 cm	1	27.8	27.8
TOTAL			428.8

Tabla 1. Costo de los componentes para unidad central

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
Push button	2	5.50	11
Resistencia 10kΩ	2	2	4
Resistencia 1kΩ	1	2	2
Resistencia 4.7kΩ	1	2	2
Led 3mm RGB	1	15	15
RJ 45 hembra	1	20	20
Pin header 40 pines	1	10	10
Bloque conector 3 pines	1	10	10
PIC16F88	1	85.5	85.5
Base DIP 18 pines	1	10	10
Tarjeta de cobre 5x5 cm	1	27.8	27.8
TOTAL			197.3

Tabla 2. Costo de los componentes para unidad control de puesto

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
Resistencia 1kΩ	1	2	2
Led 3mm verde	1	5	5
RJ 45 hembra	6	20	120
Tarjeta de cobre 10x5 cm	1	40	40
TOTAL			167

Tabla 3. Costo de los componentes para tarjeta de conexión

Diseño de prototipo de sistema embebido con lógica secuencial para la gestión del uso de puestos de trabajo en laboratorios

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
IRF530	1	25	25
Resistencia 10kΩ	1	2	2
Resistencia 4.7kΩ	1	2	2
Bloque conector 2 pines	1	8	8
Bloque conector 3 pines	1	10	10
Disipador TO-220	1	11	11
TOTAL			58

Tabla 4. Costo de los componentes para driver LED

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
Relé 5V DC	1	35	35
Resistencia 10kΩ	2	2	4
Transistor 2N3904	1	5	5
Bloque conector 2 pines	1	8	8
Bloque conector 3 pines	1	10	10
TOTAL			62

Tabla 5. Costo de los componentes para driver relé

Descripción	Cantidad	Costo unitario (C\$)	Costo (C\$)
Unidad central	1	428.8	428.8
Unidad control de puesto	8	197.3	1578.4
Tarjeta de conexión	2	167	334
Driver LED	8	58	464
Driver relé	8	62	496
TOTAL			3301.2

Tabla 6. Costo total del prototipo

Anexo D: Tablas

D.1. Tipos de mensajes enviados por protocolo I2C

Las tablas 7 y 8 detallan los mensajes que pueden ser enviados o recibidos por los microcontroladores por el protocolo I2C.

Tipo de mensaje	Trama del mensaje	Descripción
Esclavo a Maestro	0b00000001	PWM en 20% con salida al relé en 0
	0b00000010	PWM en 50% con salida al relé en 0
	0b00000011	PWM en 100% con salida al relé en 0
	0b00000100	PWM en 0% con salida al relé en 0
	0b01000001	PWM en 20% con salida al relé en 1
	0b01000010	PWM en 50% con salida al relé en 1
	0b01000011	PWM en 100% con salida al relé en 1
	0b01000100	PWM en 0% con salida al relé en 1
Maestro a Esclavo	0b00000010	Cambia el estado de la salida del relé
	0b00000110	Cambia el valor del PWM
	0b00001110	Establecer PWM a 0% y salida de relé a 0

Tabla 7. Código de mensaje I2C

Dirección	Descripción
0xA0	Dirección de escritura del puesto 1
0xA1	Dirección de lectura del puesto 1
0xA2	Dirección de escritura del puesto 2
0xA3	Dirección de lectura del puesto 2
0xA4	Dirección de escritura del puesto 3
0xA5	Dirección de lectura del puesto 3
0xA6	Dirección de escritura del puesto 4
0xA7	Dirección de lectura del puesto 4
0xB0	Dirección de escritura del puesto 5
0xB1	Dirección de lectura del puesto 5
0xB2	Dirección de escritura del puesto 6
0xB3	Dirección de lectura del puesto 6
0xB4	Dirección de escritura del puesto 7
0xB5	Dirección de lectura del puesto 7
0xB6	Dirección de escritura del puesto 8
0xB7	Dirección de lectura del puesto 8

Tabla 8. Direcciones de los esclavos I2C