

**Sistema de Selección Electrónica por Fase para Módulos de Carga de la
Plataforma UniGRID**

**José Luis Lemus Flórez
Miguel Ángel Ortega Cuadrado**

Asesores:

Diego Gómez, MsC.
Mauricio Pardo, PhD.

Presentado a:

Programa de Ingeniería Electrónica



**FUNDACIÓN UNIVERSIDAD DEL NORTE
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
ASIGNATURA DE PROYECTO FINAL
BARRANQUILLA
2018**

TABLA DE CONTENIDO

1. INTRODUCCIÓN	3
2. OBJETIVOS	4
2.1. OBJETIVO GENERAL	4
2.2. OBJETIVOS ESPECÍFICOS	4
3. DELIMITACIÓN	4
3.1. ALCANCES	4
3.2. LIMITACIONES	4
3.3. ENTREGABLES	5
4. ESTADO DEL ARTE	5
5. DESCRIPCIÓN DETALLADA	7
5.1. REQUERIMIENTOS DE DISEÑO	7
5.2. ESPECIFICACIONES DE DISEÑO	7
5.3. SELECCIÓN DE PLATAFORMAS TECNOLÓGICAS	8
5.3.1. Equipo de cómputo	8
5.3.2. Circuitos integrados	9
5.3.3. Selección de lenguajes de programación y framework	11
5.3.4. Base de datos	11
5.4. DESCRIPCIÓN DEL SISTEMA IMPLEMENTADO	12
5.4.1. Etapa 1: Intervención del módulo de carga	12
5.4.2. Etapa 2: Interfaz humano-máquina	15
5.4.2.1. Interfaz a nivel remoto	15
5.4.2.2. Interfaz a nivel local	17
5.4.3. Desarrollo web	19
6. PRUEBAS, RESULTADOS Y ANÁLISIS DE RESULTADOS	22
6.1. Validación del módulo de carga resistiva	22
6.2. Validación con los pulsadores	23
6.3. Validación de la interfaz de control	24
6.4. Validación de la carga de las curvas de demanda	24
6.5. Validación de la operación con carga	25
6.6. Análisis de resultados	25
7. CONCLUSIONES Y RECOMENDACIONES	26
8. BIBLIOGRAFÍA	27

1. INTRODUCCIÓN.

En la Universidad del Norte existe el nuevo laboratorio multipropósito de Ingeniería Eléctrica (UniGRID) que emula un sistema de distribución bajo el estándar IEEE-13. Este laboratorio ha abierto oportunidades importantes para instrucción e investigación. Cada oportunidad aparece en términos de necesidades/requerimientos/problemas que permiten tener soluciones más orientadas al concepto de Smart Grids. Para la siguiente propuesta se ha identificado que los módulos de laboratorio que emulan las cargas de cada nodo presentan dos inconvenientes. El primero, los selectores que permiten modificar los valores de carga no presentan un funcionamiento adecuado debido a humedad/condensación de los laboratorios del Departamento que ha creado una capa de "óxido" aislante; y segundo, si bien los selectores pueden ser modificados o intervenidos con algún solvente o sustancia química, su operación es completamente manual y no ofrecen posibilidad de operarlos remotamente o con autómatas programables, lo cual inhabilita dichos módulos para emular curvas de demanda bajo condiciones especiales, que obedezcan al desarrollo de algoritmos de reconfiguración/restauración de red ante fallas.

Es por tanto evidente que soluciones que persiguieran este problema serían más que bienvenidas para el desarrollo íntegro de las funcionalidades del laboratorio. Por tal razón, la implementación de un cerebro de módulo en los módulos de carga era más que deseable. Esto es así dado que permite ante todo la supresión de la intervención manual, permitiendo la manipulación remota de los módulos de carga. En detalle, esto es posible al utilizar una Raspberry Pi como un cerebro de cómputo. Sumado a lo anterior, la intervención al módulo de carga significó la utilización de un relé directamente conectado a los resistores. Por medio de la Raspberry Pi, se lograba la activación o desactivación de los relés y con ello la variación de los valores de resistencia en cada fase del módulo. Ahora bien, la manipulación remota se obtenía con una interfaz humano-máquina. Tal interfaz estaba apoyada en la comunicación entre la Raspberry Pi y un computador. En este último, el uso del servidor local asociado a la Raspberry Pi permitió la creación de una página web. En ella, toda una gama de funcionalidades estaban desplegadas. Tales funcionalidades consistían en la selección de un valor de resistencia en cada fase, garantizando la autonomía entre las fases. También se permitió la selección de un archivo .csv, garantizando la operación con curvas de demanda, es decir, variaciones de resistencia en un tiempo definido.

2. OBJETIVOS

2.1. OBJETIVO GENERAL.

Implementar un sistema de selección electrónica de cargas trifásicas e independiente por fase, de un banco de laboratorio que emula una carga de la plataforma UniGRID.

2.2. OBJETIVOS ESPECÍFICOS.

- Diseñar e Implementar un sistema de selección electrónica para un módulo trifásico de carga del laboratorio UniGRID.
- Dotar el módulo de carga modificado de una interfaz de control para recepción de mandos de operación por vía externa.
- Validar la operación completa del sistema desarrollado.

3. DELIMITACIÓN.

3.1. ALCANCES.

- Las fases de cada módulo de carga intervenido, serán completamente independientes.
- El sistema podrá ser controlado de manera local y remota.
- Se podrá recibir la información de la curva de demandas a partir de un formato de archivo computacional.
- El banco de cargas podrá entregar una retroalimentación sobre su estado y comportamiento. Lo cual evitaría tener que realizar mediciones previas a la inicialización del prototipo.
- El prototipo será escalable para poder ser implementado para un mayor número de módulos de carga a futuro.
- Se entregará la documentación requerida que garantice la transferencia de conocimiento que permita replicar el prototipo a los otros módulos de carga.

3.2. LIMITACIONES.

- Se trabajará con uno de los módulos netamente resistivos que se encuentran disponibles en el laboratorio. Dejando inicialmente de lado los bancos capacitivos e inductivos.
- El sistema se diseñará de tal forma que la comunicación no dependa de una conexión inalámbrica para su correcto funcionamiento.
- El formato de archivo del que se decodifica la información será .CSV, el cual puede ser editado mediante el software de hojas de cálculo *Excel* desarrollado por Microsoft.
- Para determinar el estado del banco de cargas se implementará un display 7 segmentos, no limitando esto a otra implementación más simple.
- Se trabajará bajo las condiciones controladas del laboratorio.
- El formato del instructivo tendrá formato de archivo de texto (PDF, .DOC, etc).

3.3. ENTREGABLES.

- Informe de avances.
- Informe final.
- Vídeo
- Póster.
- Ficha técnica estándar.
- Código fuente del software desarrollado.
- Manual de usuario de la interfaz de control para la recepción de mandos por vía externa.
- Guía técnica para la replicación de la intervención en un módulo de carga resistivo con fines de su automatización.
- Formato interno para solicitud de aprobación para registro de software.

4. ESTADO DEL ARTE.

La metodología del proyecto está enfocada principalmente al trabajo por secciones independientes: etapa de interfaz de control hombre-máquina (*HMI*), etapa de comunicación y etapa de módulos automatizados. Es por ello que la investigación realizada se encuentra direccionada de igual manera, sin dejar de lado los objetivos generales y específicos, y analizando las soluciones implementadas en distintas investigaciones ya realizadas anteriormente por otros autores.

Cabe resaltar que el equipo de cómputo principal para este proyecto es la *Raspberry Pi*, la cual posee diversas implementaciones y por ello las investigaciones tomadas como referencia no necesariamente tienen el mismo fin que el de este proyecto, sin embargo, aportan una solución efectiva a ciertos interrogantes específicos y, por ende, al logro del objetivo principal: Automatizar bancos de carga trifásicos.

Inicialmente el proyecto tendrá como base funcional a la placa *Raspberry Pi*, por ello los artículos tienen como marco de referencia el uso de este hardware. Por ello se debe conocer muy bien sus funcionalidades y límites para identificar cuáles pueden ser sus aportes al proyecto. Mirjana Maksimovic y compañía [1] en su artículo "*Raspberry Pi as Internet of Things hardware: Performances and Constraints*", explican por qué a pesar de existir muchos otros hardwares, la *Raspberry Pi* permanece vigente además de exponer cuáles son sus ventajas y desventajas reflejando argumentos como tamaño, costo, flexibilidad, alimentación, memoria de almacenamiento, entre otros. Asimismo, desarrollan su idea haciendo comparaciones entre otros dispositivos tales como *Arduino*, *BeagleBone*, entre otras. Y, finalmente, realizan una descripción física detallada de la placa y realizan conclusiones con base en lo descrito.

De igual manera, se debe tener en cuenta que la automatización es un área de la electrónica que actualmente genera interés en el campo científico y de investigación por lo tanto los estudios realizados son múltiples y aplicados a situaciones totalmente diversas.

Sin embargo, un proyecto realizado en India por Richu Varghese, Sai Krishnan S y Steve Thomas [2], asemeja en ciertos aspectos específicos la implementación que se va a llevar a cabo. El objetivo principal del proyecto fue automatizar una subestación de 11KV utilizando *Raspberry pi* como alternativa de solución, adicional al software de programación para PLC: *CODESYS*. La subestación cuenta con una interfaz amigable con el usuario en la cual se observa información específica de la subestación, además de tener la opción de manejar aspectos mecánicos como switches/breakers mediante la implementación de relés. Cabe resaltar que lo relevante de esta investigación está en el uso de la *Raspberry Pi* como medio entre el usuario y el equipo a automatizar, además del uso de relés para emular switches de control.

La primera fase del proyecto está enfocada en la interfaz hombre máquina; la fase de interacción y control. Las propuestas son múltiples: monitor, teclado y mouse; computador, pantalla táctil, etc. En [3] los autores implementan una pantalla táctil resistiva para ver e ingresar la información. Esta pantalla es conectada vía HDMI a la Raspberry Pi y calibrada con el software que provee el fabricante para el dispositivo. Asimismo, C. Wai Zhao y compañía en [4] implementan el uso de un ordenador para interactuar con la *Raspberry Pi* vía WiFi a través de comunicación cliente-servidor. Por otra parte, en [5] se toca el empleo de la Raspberry Pi, por medio de Bluetooth, como controlador de toda una serie de dispositivos que permiten la prestación de servicios de automatización aplicada en el hogar. Así mismo, se habla de cómo todos esos datos manejados por la placa pueden ser llevados al interior para su posterior visualización por parte del usuario y no sólo eso, sino también para lograr establecer una comunicación bidireccional entre usuario y sensores.

Por otra parte, la siguiente fase del proyecto se basa en la etapa de comunicación entre la interfaz antes señalada, habiendo adquirido la información brindada por el usuario, y el módulo de carga intervenido. Para realizar este proceso, es más que necesario el empleo de la Raspberry Pi. En [6], se describe un proceso de automatización industrial. Esto es destacable al ser de una escala mucho mayor al proyecto, pero con paralelismos significativos. Por ejemplo, en [6] se habla de utilizar la Raspberry Pi como controlador y servidor a la vez, enfoque parecido al buscado en el proyecto. Además, la programación es realizada en lenguaje Python. Sumado a ello, se destaca el empleo de HTML, JQuery, Ajax y Flash para el diseño de la página web. De esto es importante señalar el papel dual que puede desempeñar la Raspberry Pi en el proyecto. En [6], se detalla cómo se puede desempeñar su papel como servidor. Aun así, su tarea como controlador puede ser seguida en otro artículo. En [7], se tocan funcionalidades en las cuales la Raspberry Pi llega a tener un componente de monitoreo y control bastante significativo. Lo anterior resulta de interés, al buscarse en el proyecto la retroalimentación del estado y funcionamiento del módulo de carga. A pesar de que se piensa en sólo una solución a nivel hardware, en [7] se expone una mayor gama de posibilidades que se podrían obtener a partir de la Raspberry Pi. Más en concreto, una visualización del estado de los módulos de carga que no se limite al valor de carga. Sin embargo, cabe anotar que en [7] se habla de una red de sensores sustancialmente distinta a la que se podría aplicar el proyecto. Más aún, en el proyecto no se piensa una red de sensores, por lo tanto, tal parte de [7] es útil en cuanto a observar cómo opera la Raspberry Pi como controlador. Incluso, en [7] se habla cómo la placa obtiene información a partir de datos brindados por el usuario a partir de Internet. Siguiendo esta misma línea, [8] resulta ser una interesante propuesta a analizar. Específicamente, en [8] se plantea el empleo de una Raspberry Pi modelo B como

alternativa al empleo del Programmable Logic Controller (PLC). Esto es interesante dado que la utilización del PLC para automatización siempre resulta una opción a ponderar significativamente. Más aún cuando en [8] se habla de la automatización de una planta de tratamiento de agua. Es decir, una tarea que requiere una robustez amplia, normalmente asociada a los PLCs. Sin embargo, tal trabajo exhibe cómo la Raspberry Pi emerge como una alternativa funcional para lo mismo, claramente a través de una serie de acondicionamiento que diferencia su implementación a la del PLC. Siendo notable puesto que se configura en [8] una estrategia ampliamente a seguir. Es así que, ya sea en aplicaciones más robustas o en operaciones cuyos fines sean la domótica [8][9], la Raspberry Pi es una herramienta útil para ambos escenarios. Por otra parte, dadas las limitaciones que en coste puede llegar a tener el PLC, alternativas que permitan el uso de la Raspberry Pi con funcionalidades cercanas a las del PLC se han propuesto, obteniendo sistemas que asemejan tales prestaciones. En específico, que la programación de un PLC pueda darse con la Raspberry Pi en conjunto con otros dispositivos [10].

5. DESCRIPCIÓN DETALLADA.

5.1. REQUERIMIENTOS DE DISEÑO.

El sistema a desarrollar debía cumplir lo siguiente:

- Diseñar un sistema de selección electrónica que permita la no manipulación manual del módulo de carga resistivo y, además, permitiera la independencia entre las distintas fases de tal módulo.
- El sistema de selección debe poder ser manipulado de manera local y mediante la interfaz de usuario. Lo que significa que el sistema debe permitir la posibilidad de modificar los valores de resistencias desde el banco mismo o desde la interfaz de control.
- Construir un sistema de visualización a nivel local que permitiera una lectura ágil de la información por parte del usuario.
- Diseñar una interfaz humano-máquina -HMI- cuyo cometido fuera el envío de comandos de control dirigidos al módulo de carga, entre los cuales, se encontrasen curvas de demanda y requerimientos a nivel de cada fase.
- Para implementar el sistema se necesitarán 19 pines de la Raspberry Pi.

5.2. ESPECIFICACIONES DE DISEÑO.

El sistema implementado debe garantizar:

- Ser capaz de manejar curvas de demanda en formato .csv. Además, debe ser capaz de dar una retroalimentación del estado y funcionamiento del módulo de carga.
- La conexión existente entre la Raspberry Pi y la interfaz humano-máquina debe lograrse a través de comunicación inalámbrica, siendo la opción Wi-Fi la más fuerte candidata.
- Dadas las distancias dentro del laboratorio, debe garantizarse una comunicación entre la Raspberry Pi y la interfaz, así como entre la primera y demás dispositivos, que funcione a distancias de hasta 9 metros.

- La alimentación de la Raspberry Pi se da partir de las los tomacorrientes accesibles desde el laboratorio, notando que este es el cerebro de cómputo y, por tanto, su funcionamiento es fundamental.
- El sistema debe manejar corrientes de hasta 10A y las altas temperaturas ocasionadas por la operación del módulo de carga resistiva.

5.3. SELECCIÓN DE PLATAFORMAS TECNOLÓGICAS.

5.3.1. Equipo de cómputo.

El principal elemento a considerar para el diseño de este sistema es el cerebro de cómputo capaz de centralizar las diversas funcionalidades asociadas a la operación manual de los módulos de carga. Por lo tanto, el cometido de la automatización recaía en él. Los equipos que fueron en consideración fueron la Raspberry Pi y el controlador lógico programable (PLC). Como se describió en el estado del arte, ambas soluciones sirven para fines de automatización. Los criterios de evaluación siguen restricciones tanto a nivel operacional como económico. Es decir, soportan las especificaciones de diseño y conservan una relación de costo beneficio óptima.

PLC Schneider M340.

Entre las principales características a destacar de este dispositivo se encuentra su operatividad en condiciones térmicas fuertes, es decir, entornos industriales. A su vez, es un dispositivo que permite la comunicación vía USB, modbus, CANOpen y Ethernet. Este PLC, por tanto, posee una alta robustez y es adecuado para máquinas complejas [11]. Entiéndase por complejas aquellos asociados a operaciones industriales. En cuanto a su valor, el precio excede los 560 dólares.

Raspberry pi.

Se destaca la utilización de los puertos GPIO de este dispositivo que permite la excitación de elementos externos. Además, posee más de 20 pines, 4 puertos de comunicación USB, conexión a Ethernet y su manejo vía Wi-Fi es posible. Permite el manejo de banco de relés [12]. Este banco puede trabajar a tensiones de 110/220 V. A su vez, la posibilidad de empleo del lenguaje de programación Python significa la viabilidad de utilizar un lenguaje versátil y que permite el enlace con la interfaz de usuario implementada. En cuanto a su precio, este oscila a los 28 dólares.

Hardware implementado.

Se debe trabajar por la sencillez y soluciones óptimas. Se nota entonces que la subutilización, o sobreutilización, del dispositivo de cómputo no sería deseable. Aquí se nota que el PLC es un dispositivo con altas funcionalidades, que presta una amplia robustez, así como representa un costo económico considerable. Por el otro lado, se tiene a la Raspberry Pi con diversas funcionalidades de corte más “doméstico”, pero adaptable a operaciones más restrictivas. De igual manera, su costo representa una ventaja comparativa frente al PLC.

Si se considera el tipo de operación que se efectúa y el entorno dónde tiene lugar, se observa que el desarrollo se encuentre bajo condiciones de laboratorio: las condiciones térmicas, de vibración o de suciedad, no son extremas; los valores de corriente y voltaje no son excesivos. Por tanto, emplear un PLC implica una subutilización del dispositivo. En cambio, utilizar la Raspberry Pi significa hacer uso de las diversas capacidades del dispositivo sin desaprovechar ninguna de ellas.

Por último, el criterio económico es importante. La Raspberry Pi es mucho más económica que el PLC y sigue prestando funciones similares para la situación planteada por el problema. Además, este proyecto persigue ser escalable en el laboratorio UniGrid. Por lo que, en un futuro, el factor del precio es importante.

5.3.2. Circuitos integrados.

Expansor de pines.

Debido a la poca cantidad de pines I/O que posee la Raspberry pi, es necesario implementar un expansor de pines de entrada y salida. En este caso específico se elige el expansor MCP23017 el cual consiste en múltiples registros de configuración de 8-bits para entrada y salida [13]. La comunicación con el integrado se realiza mediante el protocolo I2C el cual es completamente compatible con la Raspberry pi.

Comentado [1]: Por qué es "poca"? se debe referenciar entonces a los requerimientos de nuestro proyecto. La demanda de Digital I/O no está listada en "5.1. REQUERIMIENTOS DE DISEÑO"

Este dispositivo permite expandir la cantidad de pines GPIO (*General purpose input/output*) en 16 y, debido a su forma de direccionamiento, se pueden implementar un total de 8 expansores de la misma referencia. Esto define a futuro la posibilidad de aumentar la cantidad de bancos a controlar mediante el sistema de selección diseñado.

Chip Address	Hardwired Address			i ² c Address
	A2	A1	A0	
000	GND	GND	GND	0x20
001	GND	GND	3V3	0x21
010	GND	3V3	GND	0x22
011	GND	3V3	3V3	0x23
100	3V3	GND	GND	0x24
101	3V3	GND	3V3	0x25
110	3V3	3V3	GND	0x26
111	3V3	3V3	3V3	0x27

Figura 1. Tabla de direcciones para múltiples integrados.

Para su correcto funcionamiento, el integrado posee ciertos parámetros que no se pueden pasar por alto. Los valores se muestran en la Tabla 1.

Voltaje de operación	1.8 a 5.5	V
Temperatura de	-40 a 125	°C

operación		
------------------	--	--

Tabla 1. Rangos de operación MCP23017. [13]

A continuación, se muestra la disposición del integrado.

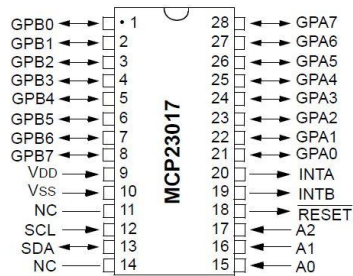


Figura 2. Diagrama de pines MCP23017 [13].

Matriz de leds y controlador.

Dado que se requiere una herramienta de visualización a nivel local, se escoge la matriz de led para ello. Se requiere, por tanto, un controlador MAX7219.

Este controlador permite utilizar una cantidad de 8 displays 7 segmentos o una matriz de 64 diodos leds mediante una librería de uso libre *Nombre de la librería* del lenguaje de programación Python.

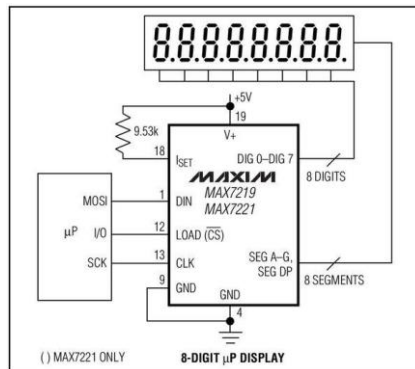


Figura 3. Disposición del controlador MAX7219. [14]

5.3.3 Selección de lenguajes de programación y framework.

Para la selección de los lenguajes de programación se debe tener en cuenta la versatilidad, tiempo de respuesta y enlace frontend-backend. Los lenguajes que cumplen estos requisitos son Javascript, PHP y Python.

Javascript, por su parte, permite la implementación de múltiples frameworks [15] lo que permitiría una migración o modificación a un framework distinto al actual. Asimismo, PHP principalmente permite la ejecución de múltiples servicios en la Raspberry Pi mediante un comando propio de este lenguaje. Finalmente, el lenguaje Python permite el control de los dispositivos conectados a la Raspberry Pi, por lo tanto, se considera fundamental para el desarrollo del sistema.

La interfaz de usuario está conformada en 2 etapas: frontend y backend.

Frontend.

Para el frontend, se utiliza el lenguaje de programación Javascript. Esta etapa consiste en diseñar la interfaz de manera funcional y amigable para el usuario. Para ella se utiliza el lenguaje de diseño gráfico CSS, más específicamente el framework Bootstrap, el cual permite mejorar la visualización y estilo de la interfaz.

Backend.

Para el backend, se utilizan ambos lenguajes PHP y Python. Mediante PHP se puede realizar el envío de la información de la sección gráfica hacia el cerebro de cómputo, el cual ejecutará la orden enviada desde la interfaz.

5.3.4. Base de datos.

Para el sistema, se aprovecha el servidor web establecido en la Raspberry Pi y se utiliza una base de datos local para leer y escribir información. En este caso se utiliza MySQL como gestor de bases de datos.

Adicionalmente el envío o lectura de la información se ve facilitado debido a la implementación de librerías que permiten el enlace entre los lenguajes de programación antes mencionados y MySQL.

5.4 DESCRIPCIÓN DEL SISTEMA IMPLEMENTADO

A continuación, en la figura se muestra el diagrama de bloques del sistema de selección diseñado.

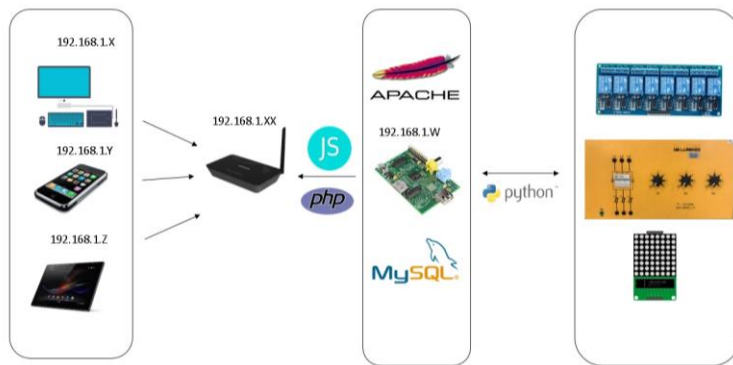


Diagrama de Bloques.
Sistema de selección

Figura 4. Diagrama de bloques del sistema.

5.4.1 Etapa 1: Intervención del módulo de carga.

El módulo de carga resistivo DL 2091 De Lorenzo es el objeto de la intervención física. Este módulo consta de 3 fases, cada uno cuenta con los valores nominales de 240 Ω , 120 Ω y 80 Ω . El módulo consta en cada una de sus fases con un selector electromecánico que permite la variación de la resistencia asignada a cada una de ellas. Los valores de resistencia nominales asociados a cada fase, según su posición son:

Posición	Valor
1	240 Ω
2	120 Ω
3	80 Ω
4	60 Ω
5	48 Ω
6	40 Ω
Off	∞

Tabla 2. Posiciones y valores de resistencias del módulo DL 2090.

El banco cuenta, además de los 6 resistores distribuidos espacialmente en paralelo (3 por cada fase), con unos fusibles en su parte inferior. Los resistores, en sus costados, estaban sujetos con unos cables que permitía su interconexión. Llegaban además a unos switches electromecánicos que estaban directamente ligados a los selectores. Es así que la

intervención debía centrarse en la necesidad de eliminar los selectores electromecánicos. Sin embargo, también se debía permitir la conservación de un componente manual para la variación de la resistencia, por lo planteado en el proyecto, y puesto que permitía una mayor versatilidad.



Figura 5. Disposición de resistencias en el banco de carga

La solución para esto se centró en un banco de relés. Los relés permiten la operación con valores de voltaje de hasta 110/220 V. Sin embargo, su energización tan sólo requiere de un valor de 5 V para estar trabajando, mientras que con 0 V ya está desenergizado. Además, al manejar con tres puertos: normalmente cerrado, normalmente abierto y común, se permite la conexión externa. Se pensó, por tanto, en el banco de relés como elemento para el manejo de los resistores y, por ende, reemplazar a los switches electromecánicos. Un banco de relés consta de 8 dispositivos. Al plantearse que por cada fase se necesitaba el control de 3 resistores, se requerían 3 relés respectivos. Por tanto, en total se requerían 9 relés. Todos estos relés iban a estar gobernados al estar conectados a los puertos de propósito general de la Raspberry Pi. Por tanto, los relés iban a estar conectados tanto a la Raspberry Pi como con los resistores.

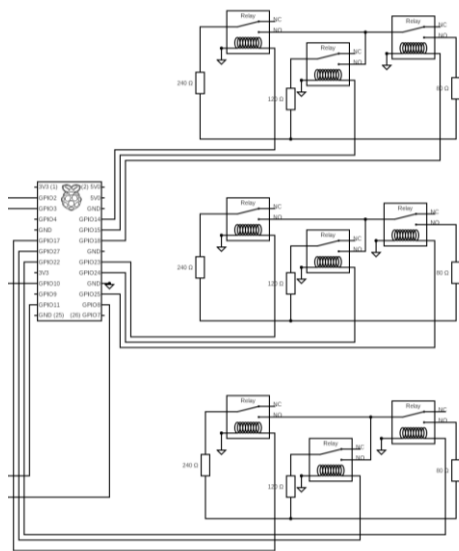


Figura 6. Esquemático de la conexión entre la Raspberry Pi, el banco de relés y los resistores

La conexión entre la Raspberry Pi y los relés garantiza que la energización de los relés, comandada desde la primera, permite que se realicen o no determinadas conexiones entre los resistores. Esto permite la obtención de los valores de resistencia nominales mostrados en la Tabla 1.

Posición	Valor	Configuración
1	240 Ω	240 Ω único resistor activado
2	120 Ω	120 Ω único resistor activado
3	80 Ω	80 Ω único resistor activado
4	60 Ω	Paralelo de 240 Ω y 80 Ω
5	48 Ω	Paralelo de 120 Ω y 80 Ω
6	40 Ω	Paralelo de 240 Ω, 120 Ω y 8 Ω
Off	∞	Circuito abierto

Tabla 3. Configuraciones necesarias, posiciones y valores en el módulo DL 2090

Tal como queda explícito en la [Tabla 2](#), unas configuraciones de resistencias, obtenidas a partir de la energización o no de los relés asignados a cada una de las fases permite la obtención de cada uno de los valores nominales de resistencia. Por tanto, una vez garantizado esto se procede también a la conexión de los fusibles. Estos también requieren ser conectados al banco de relés para así garantizar la operación, sin mayor inconveniente, de los resistores. Cabe recordar que se plantea la operación hasta con 10 A, por lo que es necesario tener estos fusibles como herramienta para prevenir daños. Efectivamente, estos proveen una protección de hasta 10 A.

5.4.2. Etapa 2: Interfaz humano-máquina.

5.4.2.1. Interfaz a nivel remoto.

Una vez las herramientas provistas para que el usuario pudiera obtener los valores de resistencia nominales ya establecidos, se requería contar con el control de esto. Para ello, se pasó a implementar una interfaz de control a nivel remoto. La Raspberry Pi debía ser la encargada de esto al manejar el banco de relés. Sin embargo, se debía idear la manera de enviarle los comandos de control. Estos debían permitir la variación de resistencia en cada una de las fases, así como la carga de curvas de demanda en documento .csv. La solución a esto está provista por la misma Raspberry Pi. Este dispositivo permite la comunicación Wi Fi. Es decir, que una vez dada podían enviarse una trama de datos desde otro dispositivo, como computador o celular, que permitiera determinadas variaciones en la Raspberry. Ahora bien, como se indicó previamente, este cerebro de cómputo permite el trabajo con PHP. Es decir, que se debía ser consecuente con esto.

La solución implementada primero consistió en garantizar la comunicación entre la Raspberry Pi y un computador portátil. Esto se logró a través de la utilización de un router que permitiera el acceso por vía Wi Fi tanto del computador como de la Raspberry Pi. Una vez conectados a ella, a través de manipulación manual en ambos dispositivos una primera vez, se da la comunicación entre ellos. El sistema implementado utiliza la red provista por el router, asignándole una IP tanto a la Raspberry Pi como al computador. Al ser los únicos dispositivos conectados, sus direcciones IP son mutuamente excluyentes. Es decir, o uno es 192.168.1.100 o el otro es 192.168.1.101.

Dada la posibilidad de utilizar PHP, la solución mediante una página web era óptima. Esta debía diseñarse esencialmente para el envío de datos que permitieran la modificación de los puertos de propósito general asociados a los relés. Esto se resume en dar 0 V o 3.3 V a tales puertos. El resultado final de la página se muestra en la siguiente imagen.

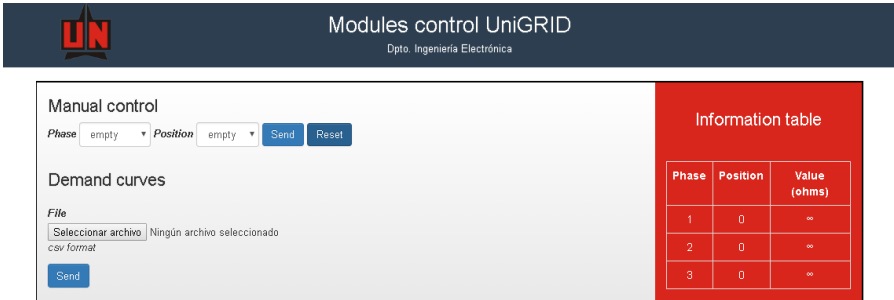


Figura 7. Interfaz de control del sistema a nivel remoto.

En la anterior figura se puede apreciar que se establecen los comandos para que se varíe la fase como la posición. El botón *Send* tiene como efecto el envío de la información utilizada por el usuario a la Raspberry Pi, así como la modificación en *Information Table* en las casillas correspondientes. El botón *Reset* tiene como finalidad llevar a todas las fases al estado Off, es decir, en circuito abierto. Por otra parte, y parte esencial de esto, es la parte de *Demand curves* donde se puede acceder a un archivo .csv. En tal archivo, se especifica los valores de resistencia en una fase en específico y los tiempos en los que se desea que se posea tal valor.

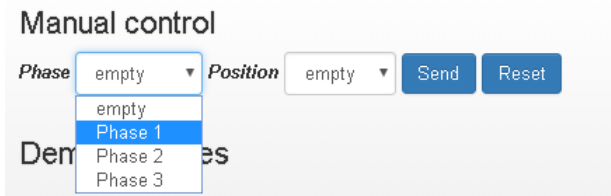


Figura 8. Selección de la fase en la interfaz de control

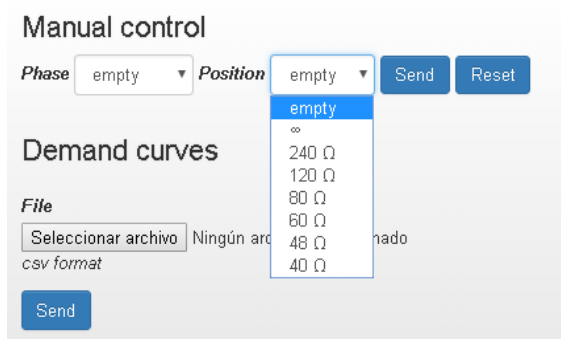


Figura 9. Selección del valor de resistencia

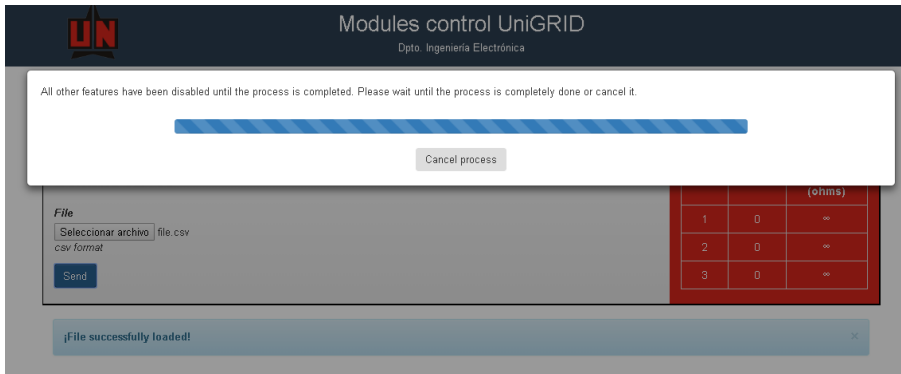


Figura 10. Carga de la curva de demanda a través de archivo .csv

Por tanto, se logran dos tipos de manejo. Un manejo que permite la variación de cada una de las fases al darle un valor en específico y otro manejo que permite la variación de la resistencia de una fase a lo largo del tiempo a través del manejo. Es decir, variaciones a nivel local.

5.4.2.2. Interfaz a nivel local.

El sistema debe garantizar que el usuario pueda detener el control a nivel remoto y operar a nivel local. Esto significa que debe bloquearse el manejo dado por la página web. Ocurrendo también debe suceder que debe haber mecanismos por los cuales se modifique la fase y la posición. Además, se pueda ver tales cambios. Para ello se requiere la serie de una serie de pulsadores con los que se logre 3 funciones: bloqueo o activación de la interfaz de control remoto, variación de la fase y variación de la resistencia. Sumado a esto, debe estar el mecanismo de visualización.

Todos estos mecanismos anteriores debían estar conectados en la Raspberry Pi. Se notó entonces que se requerían de 3 pulsadores, con dos pines cada uno, más el empleo de una matriz de leds junto con un controlador MAX7219. Esto conlleva a que se requiera de un expansor de pines MCP23017 para poder hacer una utilización de todos los dispositivos, dado que permite una mayor versatilidad al momento de utilizar la Raspberry Pi. A la solución física de esto necesitó que se pensara en una Raspberry Pi alejada de los bancos de relés, mucho más cercanos al banco de resistores. Incluso, los relés y los pulsadores iban a estar dentro del módulo de carga, mientras la Raspberry PI debía estar ligeramente alejada. Por tanto, se requirió intervenir un tramo de cables de tal manera que funcionara como una extensión entre la Raspberry Pi y una plaqueta que contase con los pines para la conexión de los dispositivos. En las siguientes figuras, se puede apreciar tanto tal plaqueta como una muestra del conjunto de dispositivos y cables utilizados.

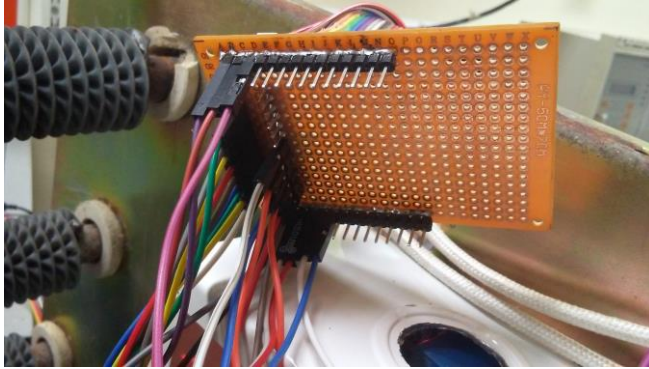


Figura 11. Plaqueta para la extensión de la conexión entre la Raspberry Pi y dispositivos esclavos.

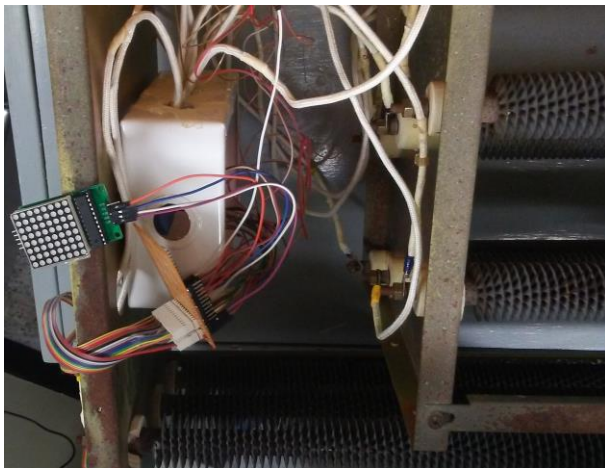


Figura 12. Conjunto de dispositivos conectados con el módulo DL 2090 abierto.

En la figura 6, se aprecia la matriz de leds conectada a la plaqueta, al igual que los demás dispositivos dentro del módulo, tales como los relés y los pulsadores. También se nota el tramo de cables que está conectada a la plaqueta y que sale del módulo para conectar con la Raspberry Pi. Se reitera otra vez que todo este conjunto de dispositivos tiene como fin permitir el control a nivel local. El pulsador inferior permite el bloqueo de la interfaz de control remota, mientras que el pulsador superior permite la variación de la fase. Por último, el pulsador intermedio permite la variación de la resistencia.



Figura 13. Vista frontal del módulo de carga DL 2090. Los selectores han sido cambiados por los pulsadores.

5.4.3. Desarrollo web.

Sección control manual.

La interfaz está diseñada de tal forma que, al seleccionar un valor de resistencias, pueda actualizar la tabla de información y la base de datos que tiene por nombre *database*. La actualización se logra de tal forma que al presionar el botón *send* se tome el valor de resistencia escogido en el selector de la interfaz y mediante una función *AJAX* se realiza la solicitud al servidor PHP, en este caso nombrado *send.php*, para enviar el valor seleccionado. En el archivo PHP se realiza la conexión a la base de datos y se envía la información además de solicitar la inicialización de un archivo Python nombrado *send.py* mediante el comando *shell_exec* y este último archivo Python será el encargado de realizar las conmutaciones en los relés para obtener el valor de resistencia seleccionado.

```
function send_command(){
    var position = document.getElementById("Posicion").value;
    var phase = document.getElementById("phase").value;
    $.ajax({
        url:'send.php',
        type:'GET',
        data:{phase:phase,position: position},
        success: function(data){
            if(data != ""){
                alert(data);
            }
        }
    });
}
```

Figura 14. Función utilizada para enviar el valor de resistencia.

```

?php
//Import functions.php file;
include 'functions.php';
//Data that will be sent to send.py file;
$position = $_GET['position'];
$phase = $_GET['phase'];

//information returned. Sent as a variable named data in file ini.js
//this will be working with the success function in ajax
if($phase == "empty" and $position == "empty"){
    echo "Fields are empty";
}elseif($phase == "empty"){
    echo "Phase's field is empty";
}elseif($position == "empty"){
    echo "Position's field is empty";
}else{
    shell_exec("python /var/www/html/send.py '$position.' '$phase.'");
}

```

Figura 15. Comando para enviar la información a la base de datos e inicializar el archivo *send.py* el cual se encarga de realizar las conmutaciones en el banco.

Adicionalmente, para realizar el reinicio de los valores de resistencia a infinito mediante el botón *reset* se realiza el mismo procedimiento anterior.

Sección curvas de demanda.

En esta sección se implementa la opción de seleccionar un archivo de tipo *.csv* el cual tendrá la información de los valores de resistencias deseados y el tiempo de ocurrencia de cada uno. Inicialmente el procedimiento realizado es discriminar la información proveniente del archivo y, similar al procedimiento anterior, enviarla a un servidor PHP, en este caso llamado *csv.php*. Este archivo será el encargado de enviar la información a la base de datos y a otro archivo Python que finalmente se encargará de realizar las respectivas conmutaciones.

En la siguiente figura se muestra la función implementada para discriminar los valores del archivo.

```

function get_csv(){
    var fileinput = document.getElementById("in_file");
    var reader = new FileReader();
    var array_spl = [];
    reader.onload = function () {
        res_str = reader.result;
        var array = res_str.split("\n");
        for(i = 1; i < array.length; i++){
            array_spl.push(array[i].split(","));
            phase1 = array_spl[0];
            phase2 = array_spl[1];
            phase3 = array_spl[2];
            time = array_spl[3];
        }
        send_csv(phase1,phase2,phase3,time,array_spl);
    };
    reader.readAsText(fileinput.files[0], 'utf8');
}

```

Figura 16. Función implementada para discriminar los valores del archivo .csv

Tabla de información.

En la tabla de información se creó una función Javascript llamada *reload_info_table* que permite la visualización actualizada de la información que se encuentra en la base de datos. Lo anterior significa que cada vez que exista una modificación en la base de datos, esta se verá reflejada en la tabla de información de la interfaz automáticamente.

Base de datos.

La base de datos MySQL se actualiza desde la interfaz o el módulo cada vez que los valores de resistencias son modificados, de tal forma que se pueda llevar un registro de la información y así, por ejemplo, tener conocimiento de qué valor de resistencia fue el seleccionado antes de apagar el sistema y cuando se encienda nuevamente el sistema se pueda observar dicho valor.

phase1	res1	phase2	res2	phase3	res3	control
0	∞	0	∞	0	∞	0

Figura 17. Información de la base de datos.

Control Manual.

Para esta sección, la lógica programable se mantiene igual, solo que en este caso se realiza la actualización de la base de datos al modificar los valores con los pulsadores implementados.

No obstante, se debe resaltar que, para poder controlar manualmente el banco de cargas, se debe activar primero dicha opción. Para ello, se utiliza un pulsador que permita realizar

esta función. Al presionar este pulsador lo que ocurrirá será enviar un valor binario “1” a la base de datos en la columna *control*. El sistema al verificar en la base de datos ese valor binario, lo que hará será deshabilitar el control por la interfaz y permitir el control manual. Esto se realiza desde un archivo Python llamado *control.py* aquí se comprueba que el botón de “inicio” fue presionado y envía el valor binario a la base de datos. En la figura se muestra el código diseñado para esta función.

```

if control == False:
    cont_con = cont_con + 1
    time.sleep(0.2)
    if cont_con > 1:
        cont_con = 0
cursor.execute("UPDATE info2 SET control =" +str(cont_con))

```

Figura 18. Código para habilitar o deshabilitar el control manual.

6. PRUEBAS, RESULTADOS Y ANÁLISIS DE RESULTADOS.

6.1. Validación del módulo de carga resistiva.

Como se indicó previamente, los valores nominales a los que los resistores están operando es de 240 Ω , 120 Ω y 80 Ω . Sin embargo, al momento de hacer una medición con el multímetro digital FLUKE 179, tales valores no fueron los obtenidos.

	Espera do (Ω)	Medid o (Ω)	Error (%)	Espera do (Ω)	Medid o (Ω)	Error (%)	Espera do (Ω)	Medid o (Ω)	Error (%)
Fase A	80	84.4	5.5	120	122.5	2.083	240	250.7	4.458
Fase B	80	83.3	4.125	120	122.7	2.25	240	257.4	7.25
Fase C	80	82.4	3	120	123.6	3	240	256.1	6.708

Tabla 4. Valores medidos y nominales de los resistores del módulo de carga, con los errores asociadas

Tales diferencias ya deben ser asociadas a variaciones debidas al desgaste de la composición de los resistores, por lo que no se puede asegurar la entrega de los valores nominales por parte del sistema. Tal garantía es provista por el módulo de carga y, como se evidencia en la Tabla 3, no se puede asegurar un completo cumplimiento. Sin embargo, la operación con ellos no arroja valores con errores muy grandes y, por tanto, no se considera una limitante para ello.

6.2. Validación con los pulsadores.

Los pulsadores deben garantizar 3 funciones, como se ha mencionado antes. La primera de ella es que se dé el bloque de la interfaz de control a nivel remoto, de tal manera que sólo se opere a nivel local. El segundo punto es que se garantice la variación de la fase. Por último, garantizar la variación de la resistencia en la fase respectiva. En efecto, lo anterior es logrado por 3 pulsadores descritos en la sección anterior.

Por tanto, se debe garantizar que una vez activado el modo local en el último pulsador, se logre la variación en los otros pulsadores. Se suma a esto la visualización que se debe lograr en la matriz de leds que dé información sobre la fase y la posición que se está obteniendo.

De manera satisfactoria, se logró que, al momento de presionar el botón inferior, la interfaz de control se bloqueaba. Que, al presionar el botón superior, se variaba la fase, escalando de 1 a 3, devolviendo luego a la fase 1. Con respecto al pulsador intermedio, se variaba la posición desde la posición off hasta la posición 6, para luego regresar a la posición off. Se logró comprobar que los valores esperados se alcanzaban al ser medido por el multímetro FLUKE 179.



Figura 14. Control a nivel local dado por los pulsadores y visualización por medio de la matriz de leds.

Como se nota en la figura, el valor mostrado en la matriz corresponde a un valor de 120 Ω para la fase 1. Sin embargo, está sujeto a variaciones propias de los resistores. En todo caso, se llega a decir que la función esperada se cumple. Se logra el bloque de la interfaz para lograr que los pulsadores trabajen junto con la matriz de leds.

6.3. Validación de la interfaz de control.

Aquí se requiere que, al momento de utilizar la interfaz, al ingresar las peticiones en cuanto a fase y resistencia, se dé los cambios necesarios en la tabla de información como en los

valores de resistencia medidos por el multímetro. Lo anterior, está sujeto a otra vez a las limitaciones dadas en las variaciones de los valores de los resistores. Sin embargo, se señala otra vez, no es limitante mayor para el desarrollo de la implementación.



Figura 15. Modificaciones en la interfaz de control y medida en el multímetro.

Lo anterior, es logrado satisfactoriamente por el sistema y la figura previa lo hace constar. Por tanto, se puede hablar del éxito de la interfaz de control a nivel remoto de control como herramienta para el control de los valores de resistencia deseado.

6.4. Validación de la carga de las curvas de demanda.

Un último requerimiento que perseguía el sistema era poder contar con la posibilidad de cargar curvas de demanda, es decir, variaciones de la carga con respecto al tiempo. En efecto, lo anterior se logró y corresponde a la sección de la interfaz llamada *Demand curves*, de la que se tuvo una aproximación antes en el informe. Ahora bien, se pasó a comprobar que los valores de resistencia se ajustaban a los dados en el archivo .csv que se cargaba y con la longitud temporal esperada.



Figura 16. Manejo de la interfaz de control por medio del archivo .csv.

Se puede apreciar un cambio tanto en el valor medido por el multímetro como en la tabla de información al cargarse con el archivo .csv. Esto indica que se ha logrado el cometido con esta funcionalidad al darse las variaciones esperadas con el comando esperado.

6.5. Validación de la operación con carga.

Al módulo de carga se le conectó una fuente que manejó una corriente de línea de 7.6 A y un voltaje de línea de 176.8 V. Además, se realizó una conexión delta en el módulo. Al momento de operar, se evidenció que los resistores tuvieron un aumento de su temperatura en un tiempo aproximado de 2 minutos. A pesar de que se operó con la carga cerca de 20 minutos, los dispositivos asociados al sistema no presentaron daños. Entre estos dispositivos se cuenta al cableado asociado, al banco de relés, a la caja que soportaba a este último

6.6. Análisis de resultados.

Los resultados arrojados por el FLUKE 179 permiten afirmar que se logró valores de carga cercanos a los nominales. Las medidas poseen los errores inherentes al cambio debido al desgaste de los resistores. Sin embargo, tal variación en todos los resistores es menor al 10%. Significando que no se manejan valores alejados de los requeridos.

Por otra parte, se garantiza que la interfaz de control en efecto manda los comandos deseados y tienen una efectiva recepción por parte de la Raspberry Pi. Esto sucede cuando se trabaja en la selección por fase y resistencia. Acarreando la configuración necesaria de relés, manejados por la Raspberry Pi, que lleve al valor esperado. También se tiene éxito al momento de cargar un documento .csv que describa una curva de demanda. Teniendo las modificaciones esperadas para ello en cuanto a los valores de resistencia.

Adicional a todo lo anterior, se logró constatar que no hay derretimiento o afectación de sus elementos por efectos térmicos al momento de utilizar una carga. Esto habla de una operación a largo tiempo estable y confiable para el usuario.

Por todo lo anterior, se puede afirmar que el sistema implementado cumple a cabalidad con lo planteado en los objetivos descritos al principio del informe. Por ende, se logra la selección electrónica trifásica del módulo de carga 2090 con una operación segura para el usuario.

7. CONCLUSIONES Y RECOMENDACIONES.

Se automatizó los módulos de carga resistiva de la microred IEEE-13 de la Universidad del Norte. Se logra la supresión completa de la operación manual. Esto a través de la intervención física ejemplificada en el uso de relés y la Raspberry Pi. La variación de resistencia es efectuada al ser los primeros energizados o no por la placa de cómputo. También se alcanza a la entrega de una interfaz humano-máquina. Soportada en la comunicación inalámbrica entre la Raspberry Pi y el computador, se logró su elaboración. Se cristaliza en ella la posibilidad de decidir en qué valor de resistencia se desea cada una de las tres fases de los bancos de resistencia. Sumado a ello, se permite la adquisición de un archivo .csv que dé las curvas de demanda que se deseen proveer para la operación del módulo.

El banco de relés es un actor principal en el sistema al acoplar la circuitería diseñada para la traducción de los comandos de control con aquella encargada de afectar directamente a

los bancos de resistores. En específico, los resistores fueron intervenidos en su cableado y, a través de éste, conectados a los relés. Estos a su vez eran controlados por la Raspberry Pi, la cual también estaba manejando una matriz de LEDs que permitiera una visualización a nivel local de la fase y la posición dada.

Se logró, por tanto, un sistema automatizado que permitiera, ante todo, la manipulación remota de los módulos de carga. Se evitó, de tal manera, la necesidad de una intervención humana. También se garantiza la posibilidad de emplear curvas de demanda. De igual manera, la correcta interacción del usuario con el sistema es dada como su visualización a nivel local como remoto.

La experiencia aplicada en el módulo de carga resistiva es transferible a los otros módulos de carga restantes: inductivo y capacitivo. Se necesita un análisis de la estructura interna de tales módulos para definir conexiones con los relés. Una vez hecho, es una secuencia procedimental establecida en el código del sistema consistente en energizar o no los relés. Además, como se espera su operación con carga, debe hacerse el recubrimiento térmico necesario para evitar problemas en tales módulos.

8. BIBLIOGRAFÍA

- [1]. M. Maksimović, V. Vujović, N. Davidović, V. Milošević, B. Perišić, "Raspberry Pi as Internet of Things hardware: Performances and Constraints", Proceedings of 1st International Conference on Electrical, Electronic and Computing Engineering IeETRAN 2014, 2014.
- [2]. R. Varghese, S. Krishnan, S. Thomas, "Automation of 11 kv Substation using Raspberry pi", en 2017 International Conference on circuits Power and Computing Technologies, 2017.
- [3]. G. Shanmuga, B. Patibandala, H. Santhanam, S. Gaddam, V. Krishna Alla, G. Krishna Alla, S. Chaitanya, S. Boppana, J. Conrad, "Bluetooth Communication using a Touchscreen Interface with the Raspberry Pi", 2013.
- [4]. C. Wai Zhao, J. Jegatheesan, S. Chee Loon, "Exploring IOT Application Using Raspberry Pi", International Journal of Computer Networks and Applications, vol. 2, no. 1, 2015, pp. 27-34.
- [5]. P Bhaskar Rao, S.K. Uma, "Raspberry Pi Home Automation with Wireless Sensors Using Smart Phone", International Journal of Computer Science and Mobile Computing, vol. 4, no. 5, 2015, pp. 797 -803.
- [6]. H. K. Merchant, D. D. Ahire, "Industrial Automation using IoT with Raspberry Pi", International Journal of Computer Applications, vol. 168, no.1, 2017, pp. 44-46.
- [7]. R. O. Nehete, A.S. Bhide, "Raspberry PI 3 Based Control and Monitor Remote Machine Automation ", International Journal Of Engineering And Computer Science, vol 6, no. 1, 2017, pp. 20151-20155.
- [8]. S. S. Lagu, S. B. Deshmukh, "Raspberry Pi for Automation of Water Treatment Plant", en International Conference on Computing Communication Control and Automation, 2015, pp. 532-534.
- [9] A. Florentin Iftimie, C. Vinte, "Upon a Home Assistant Solution Based on Raspberry Pi Platform", Informatica Economică, vol 21, no.2, 2017, pp. 5-17.
- [10] A.I Bermúdez Román, J. Gaztelumendi Arriaga, " Building and programming an autonomous robot using Raspberry Pi as a PLC", Trabajo de grado, Univ. Skovde, Västra Götaland, Suecia, 2016.
- [11] Schneider Electric. "Modicon M340 con Unity Pro. Procesadores, bastidores y módulos de fuente de alimentación. Manual de configuración" (2009). [En línea]. Disponible en: https://www.schneider-electric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/28000/FA28444/es_ES/Unity_v4.1_M340_Procesadores_Bastidores_y_Fuentes_de_Alimentacion.pdf, Consultado en: Agosto 23, 2018.
- [12] P. Kumar, U. Chandra Pati, "Arduino and Raspberry Pi based smart communication and control of home appliance system", en 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016
- [13] Microchip. "MCP23017 16-Bit I2C I/O expander with serial interface"(Device overview). [En línea]. Disponible en: <https://www.microchip.com/wwwproducts/en/MCP23017>. Consultado en: Nov 18, 2018
- [14] Maxim integrated. "MAX7219/MAX7221 Serially Interfaced, 8-Digit LED Display Drivers". [En línea]. Disponible en: <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>. Consultado en: Nov 18, 2018.
- [15] "What's New in JavaScript Frameworks — March 2018", JavaScript Report, 2018. [Online]. Disponible: <https://javascriptreport.com/whats-new-in-javascript-frameworks-march-2018/>.