



# **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

## **DISEÑO DE UN SISTEMA DE CONTROL BMS (BUILDING MANAGEMENT SYSTEM), PARA LA GESTIÓN DEL BUS DOMÓTICO HDL BUS PRO, PARA LA GENERACIÓN DE MACRODATOS BASADO EN SOFTWARE LIBRE**

**OSCAR WLADIMIR GÓMEZ MORALES**

Trabajo de Titulación modalidad: Proyecto de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito para la obtención del grado de:

**MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN INDUSTRIAL**

**RIOBAMBA – ECUADOR**

**Julio 2018**

**ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO  
INSTITUTO DE POSTGRADO Y EDUCACION CONTINUA**

**CERTIFICACIÓN**

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, denominado “DISEÑO DE UN SISTEMA DE CONTROL BMS (BUILDING MANAGEMENT SYSTEM), PARA LA GESTIÓN DEL BUS DOMÓTICO HDL BUS PRO, PARA LA GENERACIÓN DE MACRODATOS BASADO EN SOFTWARE LIBRE”, de responsabilidad del Ingeniero Oscar Wladimir Gómez Morales, ha sido prolijamente revisado y se autoriza su presentación.

Ing. Wilson Zúñiga Vinuesa, M.Sc.

**PRESIDENTE**

-----

**FIRMA**

Ing. Carlos Fabara Davila, MSc.

**DIRECTOR**

-----

**FIRMA**

Ing. David Raimundo Rivas, MSc.

**MIEMBRO**

-----

**FIRMA**

Ing. Milton Fabricio Pérez, MSc.

**MIEMBRO**

-----

**FIRMA**

Riobamba, Julio del 2018

©2018, Oscar Wladimir Gómez Morales

Se autoriza la producción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el derecho de autor.

## DERECHOS INTELECTUALES

Yo, Oscar Wladimir Gómez Morales, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

---

Ing. Oscar Wladimir Gómez Morales.

Cc.: 0503358624

## **DEDICATORIA**

Dedico este logro principalmente a Dios, por darme la vida y sus bendiciones por permitir concluir un paso más en mi vida profesional. A mis amados padres José Gómez y Julia Morales por enseñarme a luchar por mis sueños y objetivos y ser el pilar más importante y por demostrarme siempre su amor y apoyo incondicional. A Diana quien es la madre de mis dos hijas Ailyn y Sofía quienes son mi fortaleza para seguir adelante en todos mis proyectos.

A todas y cada una de las personas que ayudaron directa e indirectamente en la culminación de este proyecto.

*Oscar.*

## AGRADECIMIENTO

Ante todo agradezco a Dios por permitirme lograr todas las metas que me he propuesto y me propondré durante el transcurso de mi vida profesional.

Enfatizo mis más sinceros agradecimientos a la Escuela Superior Politécnica de Chimborazo, en especial al Instituto de Posgrados y Educación Continua por su programa de Maestría en Sistemas de Control y Automatización Industrial.

De manera especial agradezco a los miembros de tribunal al MSc Carlos Fabara Davila en calidad de director, al MSc David Rivas y al MSc Milton Fabricio Pérez por guiarme en el transcurso de la elaboración del proyecto de titulación, sus enseñanzas y tiempo ha permitido culminar el presente proyecto de titulación.

Y a todas las personas que de una u otro forma aportaron para que el presente proyecto haya llegado a su feliz término.

*Oscar.*

## CONTENIDO

|   |          |
|---|----------|
| RESUMEN.....  | xv       |
| ABSTRACT.....   | xvi      |
| <b>CAPÍTULO I</b>   |          |
| <b>1. PLANTEAMIENTO DEL PROBLEMA .....</b>                        | <b>1</b> |
| 1.1. Formulación del problema.....                                | 2        |
| 1.2. Preguntas directrices .....                                  | 2        |
| 1.3. Justificación del problema .....                             | 3        |
| 1.4. Objetivos.....   | 5        |
| 1.4.1. <i>Objetivo general</i> .....                              | 5        |
| 1.4.2. <i>Objetivos Específicos</i> .....                         | 5        |
| 1.5. Hipótesis .....  | 6        |
| 1.5.1. <i>Hipótesis General</i> .....                             | 6        |
| <b>CAPÍTULO II</b>  |          |
| <b>2. MARCO TEÓRICO .....</b>                                     | <b>7</b> |
| 2.1. Antecedentes.....  | 7        |
| 2.2. Bases teóricas .....   | 8        |
| 2.3. Marco conceptual .....                                       | 9        |
| 2.3.1. <i>Domótica e Inmótica</i> .....                           | 9        |
| 2.3.2. <i>Sistemas de Control BMS</i> .....                       | 9        |
| 2.3.3. <i>Arquitectura del sistema de control BMS</i> .....       | 11       |
| 2.3.4. <i>Niveles de Control que trabaja el sistema BMS</i> ..... | 11       |

|   |    |
|---|----|
| <b>2.3.5. Funciones y aplicaciones de los niveles de control</b> .....  | 12 |
| <b>2.3.5.1. Nivel 1 Supervisión</b> .....                               | 12 |
| <b>2.3.5.2. Nivel 2 Control</b> .....                                   | 13 |
| <b>2.3.5.3. Nivel 3 Campo</b> .....                                     | 13 |
| <b>2.3.6. Sistema Domótico HDL Buspro</b> .....                         | 14 |
| <b>2.3.7. Ventajas del Sistema HDL Buspro</b> .....                     | 14 |
| <b>2.3.8. Sistema de Comunicación.</b> .....                            | 14 |
| <b>2.3.9. Topología del Sistema</b> .....                               | 15 |
| <b>2.3.10. Componentes del sistema HDL Buspro</b> .....                 | 17 |
| <b>2.3.11. Reglas de Enrutamiento</b> .....                             | 19 |
| <b>2.3.12. HDL Buspro Herramienta de configuración HBST</b> .....       | 19 |
| <b>2.3.13. Selección del Software libre</b> .....                       | 20 |
| <b>2.3.13.1. Principales Características del Lenguaje Python</b> .....  | 20 |
| <b>2.3.13.2. Principales Características del Lenguaje Ruby</b> .....    | 21 |
| <b>2.3.13.3. Principales Características de JAVA NetBeans IDE</b> ..... | 21 |
| <b>2.3.14. Macrodatos (Big Data)</b> .....                              | 23 |
| <b>2.3.15. Protocolo de comunicación UDP</b> .....                      | 24 |
| <b>2.3.15.1. Puertos UDP</b> .....                                      | 24 |
| <b>2.3.15.2. Formato del mensaje UDP</b> .....                          | 25 |
| <br><b>CAPÍTULO III</b>   |    |
| <b>3. DESARROLLO DEL SISTEMA DE CONTROL BMS</b> .....                   | 26 |
| <b>3.1. Instalación de los softwares</b> .....                          | 26 |
| <b>3.1.1. Oracle Java JDK 8u111 (Java SE Development Kit)</b> .....     | 26 |

|        |  |    |
|--------|--|----|
| 3.1.2. | <i>NetBeans IDE 8.2 (Entorno de Desarrollo Integrado)</i>                        | 27 |
| 3.1.3. | <i>MySQL version 5.5.53.0 (Database Management System)</i>                       | 27 |
| 3.1.4. | <i>Workbench 6.3 CE</i>  | 27 |
| 3.2.   | <b>Diseño de la interfaz del sistema bms</b>                                     | 28 |
| 3.2.1. | <i>Ventana de acceso al sistema de control BMS</i>                               | 29 |
| 3.2.2. | <i>Venta para crear un nuevo usuario</i>   | 30 |
| 3.2.3. | <i>Ventana de menú del sistema de control BMS</i>                                | 31 |
| 3.2.4. | <i>Ventana del consumo energético</i>  | 32 |
| 3.2.5. | <i>Ventana de iluminación</i>  | 33 |
| 3.2.6. | <i>Ventana de seguridad</i>  | 34 |
| 3.2.7. | <i>Ventana de menú de escenas</i>  | 35 |
| 3.2.8. | <i>Ventana de cargas inductivas</i>  | 36 |
| 3.2.9. | <i>Ventana de históricos</i>   | 37 |
| 3.3.   | <b>Implementación de la comunicación UDP</b>                                     | 38 |
| 3.3.1. | <i>Conformación del bus de HDL Buspro</i>  | 38 |
| 3.3.2. | <i>Generación de la trama UDP en NetBeans para el control de los equipos HDL</i> | 41 |
| 3.3.3. | <i>Generación de Macrodatos (Big Data) del sistema de control BMS</i>            | 45 |

## CAPÍTULO IV

|        |  |    |
|--------|--|----|
| 4.     | <b>Pruebas y resultados</b>  | 49 |
| 4.1.   | <b>Crear un nuevo usuario desde el sistema BMS y guardar en la base de datos del sistema</b> | 50 |
| 4.2.   | <b>Pruebas de envío de peticiones desde el sistema BMS hacia los equipos de HDL</b>          | 53 |
| 4.2.1. | <i>Solicitud para encender las salidas de relés</i>  | 53 |

|        |  |    |
|--------|--|----|
| 4.2.2. | <i>Petición para encender las salidas de los dimmers</i> .....                           | 54 |
| 4.2.3. | <i>Petición para encender las cargas inductivas y leer las entradas analógicas</i> ..... | 55 |
| 4.2.4. | <i>Lectura de datos de los equipos de HDL hacia el sistema de control BMS</i> .....      | 56 |
| 4.3.   | <b>Pruebas de generación de macrodatos del sistema de control BMS</b> .....              | 61 |
| 4.4.   | <b>Generación de reportes de macrodatos y alarmas</b> .....                              | 63 |
| 4.5.   | <b>Análisis del tráfico de datos en el bus HDL del sistema de control BMS</b> .....      | 64 |
| 4.5.1. | <i>Análisis de una petición enviada desde el BMS al módulo de entrenamiento</i> .....    | 65 |
| 4.6.   | <b>Test SUS (System Usability Scale)</b> .....   | 67 |
| 4.6.1. | <i>Utilización de la escala SUS</i> .....  | 67 |
| 4.7.   | <b>Validación de la Hipótesis</b> .....  | 69 |
|        | <b>CONCLUSIONES</b> .....  | 72 |
|        | <b>RECOMENDACIONES</b> .....   | 73 |
|        | <b>BIBLIOGRAFÍA</b>  |    |
|        | <b>ANEXOS</b>  |    |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| <b>Tabla 1.2:</b> Algunos componentes de la marca HDL Bus Pro. ....                               | 17 |
| <b>Tabla 2.2:</b> Puertos UDP .....   | 24 |
| <b>Tabla 1.3:</b> Descripción de funcionamiento de las ventanas del sistema BMS.....              | 29 |
| <b>Tabla 2.3:</b> Definición de secuencia de conexión de terminal cable especial HDL. ....        | 40 |
| <b>Tabla 3.3:</b> Definición de secuencia de conexión de terminal del cable UTP CAT5E.....        | 41 |
| <b>Tabla 4.3:</b> Trama UDP generada para encender un canal de relé número 3. ....                | 42 |
| <b>Tabla 5.3:</b> Datos para el cálculo de la redundancia cíclica. ....                           | 45 |
| <b>Tabla 1.4:</b> Código de operación para leer el voltaje de Power Meter. ....                   | 57 |
| <b>Tabla 2.4:</b> Respuesta de módulo Power Meter para leer el voltaje. ....                      | 58 |
| <b>Tabla 3.4:</b> Resultados del test SUS aplicado a los usuarios del sistema de control BMS..... | 68 |
| <b>Tabla 4.4:</b> Estándares de almacenamiento de macrodatos (Big Data). ....                     | 71 |

## ÍNDICE DE ILUSTRACIONES

|  |    |
|--|----|
| <b>Figura 1.1:</b> Diagrama del sistema de control BMS.....                        | 5  |
| <b>Figura 1.2:</b> Principales Gestiones de un sistema de control BMS. ....        | 10 |
| <b>Figura 2.2:</b> Arquitectura del sistema BMS. ....                              | 11 |
| <b>Figura 3.2:</b> Niveles de control del sistema BMS y aplicaciones.....          | 12 |
| <b>Figura 4.2:</b> Nivel 1 de supervisión del sistema BMS.....                     | 12 |
| <b>Figura 5.2:</b> Nivel 2 de control del sistema BMS.....                         | 13 |
| <b>Figura 6.2:</b> Nivel 3 de campo del sistema BMS. ....                          | 13 |
| <b>Figura 7.2:</b> Puerto Buspro el conector tiene D+, D-, COM y 24VDC.....        | 15 |
| <b>Figura 8.2:</b> (a) Topología tipo bus, (b) topología tipo estrella. ....       | 15 |
| <b>Figura 9.2:</b> Topología del sistema.....                                      | 16 |
| <b>Figura 10.2:</b> La Topología en anillo no es permitida.....                    | 16 |
| <b>Figura 11.2:</b> Panel principal del HBST.....                                  | 19 |
| <b>Figura 12.2:</b> Formato del mensaje UDP. ....                                  | 25 |
| <b>Figura 1.3:</b> Diagrama de flujo del sistema de control BMS a implementar..... | 26 |
| <b>Figura 2.3:</b> Entorno de trabajo NetBeans IDE 8.2. ....                       | 27 |
| <b>Figura 3.3:</b> Entorno de trabajo Workbench 6.3 CE. ....                       | 28 |
| <b>Figura 4.3:</b> Ventana de acceso al sistema BMS.....                           | 30 |
| <b>Figura 5.3:</b> Ventana para crear usuarios nuevos. ....                        | 31 |
| <b>Figura 6.3:</b> Ventana del menú del sistema BMS.....                           | 32 |
| <b>Figura 7.3:</b> Ventana de consumo eléctrico del sistema BMS. ....              | 33 |
| <b>Figura 8.3:</b> Ventana de iluminación del sistema BMS. ....                    | 34 |
| <b>Figura 9.3:</b> Ventana de seguridad del sistema BMS.....                       | 35 |
| <b>Figura 10.3:</b> Ventana de menú de escenas del sistema BMS.....                | 36 |
| <b>Figura 11.3:</b> Ventana de cargas inductivas del sistema BMS.....              | 37 |
| <b>Figura 12.3:</b> Ventana de históricos del sistema BMS.....                     | 38 |

|   |    |
|---|----|
| <b>Figura 13.3:</b> Conformación de la comunicación del sistema BMS.....  | 39 |
| <b>Figura 14.3:</b> Conexión del cable HDL Buspro/KNX.....  | 40 |
| <b>Figura 15.3:</b> Conexión del cable UPT CAT5E. ....  | 41 |
| <b>Figura 16.3:</b> Trama UDP del sistema HDL Bus Pro.....  | 42 |
| <b>Figura 17.3:</b> Diagrama de bloques de la trama UDP para el control de los equipos HDL. ....                      | 43 |
| <b>Figura 18.3:</b> Clase udp_socket y datos de cabecera de HDL.....  | 43 |
| <b>Figura 19.3:</b> Datos para el cálculo de la redundancia cíclica.....  | 44 |
| <b>Figura 20.3:</b> Esquema bms con las dos tablas de alarmas y login. ....   | 47 |
| <b>Figura 21.3:</b> Esquema macrodatos con la tabla de consumo_energetico.....  | 48 |
| <b>Figura 1.4:</b> Módulo implementado para las pruebas del sistema BMS. ....   | 50 |
| <b>Figura 2.4:</b> Generación de un nuevo usuario desde el sistema BMS.....   | 50 |
| <b>Figura 3.4:</b> Datos guardados de los usuarios generados desde el sistema BMS. ....                               | 51 |
| <b>Figura 4.4:</b> Lectura desde el BMS hacia la base de datos del nuevo usuario. ....                                | 52 |
| <b>Figura 5.4:</b> Datos incorrectos al ingresar al sistema BMS.....  | 52 |
| <b>Figura 6.4:</b> Instrucciones para accionar las salidas de relé del sistema BMS.....                               | 53 |
| <b>Figura 7.4:</b> Acción enviada desde el Sistema BMS hacia las salidas de relé. ....                                | 54 |
| <b>Figura 8.4:</b> Instrucciones para enviar el porcentaje de los dimmers hacia los equipos de<br>HDL.....            | 54 |
| <b>Figura 9.4:</b> Acción enviada desde el Sistema BMS hacia los dimmers. ....  | 55 |
| <b>Figura 10.4:</b> Instrucciones para enviar a accionar las cargas inductivas del sistema BMS. ....                  | 56 |
| <b>Figura 11.4:</b> Petición desde el BMS para encender las cargas inductivas/lectura de la entrada<br>analógica..... | 56 |
| <b>Figura 12.4:</b> Función para leer el voltaje del equipo Power Meter.....  | 58 |
| <b>Figura 13.4:</b> ventana de consumo energético, medición de parámetros de energía del canal 2.                     | 59 |
| <b>Figura 14.4:</b> Ventana de seguridad del BMS, adquisición de datos de sensores y de la<br>cámara.....             | 60 |
| <b>Figura 15.4:</b> Ventana de históricos del BMS, adquisición de datos de variables eléctricas. ....                 | 61 |
| <b>Figura 16.4:</b> Guardado de datos del sistema BMS para la generación de los macrodatos.....                       | 62 |

|   |    |
|---|----|
| <b>Figura 17.4:</b> Guardado de datos de alarmas y usuarios del sistema BMS. ....                       | 62 |
| <b>Figura 18.4:</b> Registro de macrodatos generados por el sistema BMS. ....                           | 63 |
| <b>Figura 19.4:</b> Registro de alarmas generadas por el sistema BMS.....                               | 64 |
| <b>Figura 20.4:</b> Estructura del software Wireshark, 3 paneles. ....                                  | 65 |
| <b>Figura 21.4:</b> Datos capturados de la trama enviada desde el BMS a los equipos de HLD. ....        | 66 |
| <b>Figura 22.4:</b> Tiempos de retardo de varias peticiones realizadas desde el BMS. ....               | 67 |
| <b>Figura 23.4:</b> Almacenamiento de información de los equipos conectados en el bus HDL. ....         | 69 |
| <b>Figura 24.4:</b> Almacenamiento de información de los equipos conectados en el bus HDL. ....         | 70 |
| <b>Figura 25.4:</b> Información de los macrodatos guardados en la base de datos MySQL<br>Workbench..... | 70 |

## RESUMEN

El presente trabajo de titulación tuvo como propósito el diseño de un sistema de control BMS (Building Management System) para la supervisión y el control del bus domótico HDL Buspro, el sistema genera macrodatos (Big Data) del consumo energético del BMS los macrodatos son guardados en una base de datos. El desarrollo del sistema de control BMS se realizó en software libre, la comunicación de los equipos con el BMS se implementó con el protocolo de comunicación UDP (User Datagram Protocol) el cual se basa en el intercambio de datagramas él envío de los datagramas es posible sin que se establezca una conexión proporcionando un mecanismo para distinguir múltiples aplicaciones fuente o destino en un mismo host. El BMS diseñado es un software muy flexible y fácil para controlar para los usuarios dando como resultado el incremento de la eficiencia, la productividad, confort y el ahorro energético donde se aplique este sistema. Los resultados obtenidos de los experimentos demuestran que el sistema de control BMS desarrollado cumple con los parámetros y las características de generar macrodatos del consumo energético para su posterior análisis de los mismos. Los macrodatos generados en esta investigación servirán a futuro para diseñar un plan de mantenimiento de una edificación y tomar correctivos del mismo. Se recomienda la implementación de estos sistemas de control BMS en instalaciones como hospitales, aeropuertos, universidades, centros comerciales, hoteles, puertos marítimos, etc. Que lleguen a instalar estos sistemas para obtener un alto grado de sostenibilidad energética, destacando el ahorro energético y permitiendo a las personas que habitan o trabajan en esas instalaciones darles un nivel de confort adecuado para realizar sus operaciones mejorando su productividad.

**Palabras clave:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, < BUILDING MANAGEMENT SYSTEM (BMS)>, <HDL BUSPRO>, <KNX>, <MACRODATOS>, <SOCKET UDP>, <SISTEMA DE CONTROL>, <SOFTWARE LIBRE>, <USER DATAGRAM PROTOCOL (UDP)>

## ABSTRACT

The purpose of the present titration work was to design a BMS control system (Building Management System) for the supervision and control of the HDL Buspro home automation bus, the system generates big data for the energy consumption of the BMS. Big data are saved in a database. The development of the BMS control system was carried out in free software, the communication of the equipment with the BMS was implemented with the UDP communication protocol (User Datagram Protocol) which is based on the datagram exchange it sent of the datagrams is possible without establishing a connection by providing a mechanism to distinguish multiple source or target applications on the same host. The BMS designed is a very flexible and easy software to control for the users resulting in the increase of efficiency, productivity, comfort and energy saving where this system is applied. The results obtained from the experiments show that the BMS control system developed complies with the parameters and characteristics of generating large data on energy consumption for their subsequent analysis. The big data generated in this research will serve in the future to design a maintenance plan for a building and take corrective measures. It is recommended the implementation of these BMS control systems in facilities such as hospitals, airports, universities, shopping centers, hotels, seaports, etc. That they arrive to install these systems to obtain a high degree of energetic sustainability, emphasizing the energetic saving and allowing to the people who inhabit or work in those facilities to give them an adequate level of comfort to carry out their operations improving their productivity.

**KEYWORDS:** < BUILDING MANAGEMENT SYSTEM (BMS)>, <HDL BUSPRO>, <KNX>, <MACRODATOS>, <SOCKET UDP>, <SYSTEM CONTROL>, <FREE SOFTWARE>, < USER DATAGRAM PROTOCOL (UDP)>

# CAPÍTULO I

## INTRODUCCIÓN

### 1. PLANTEAMIENTO DEL PROBLEMA

A nivel mundial los países desarrollados como son Estados Unidos, España, Alemania, Francia, Japón, China, Reino Unido entre otros; buscan construir edificaciones lo más inteligentes posibles a que se debe esto, se debe a tener una eficiencia en operatividad, gestión y control de todos los sistemas de una edificación y a crear una base de datos (macrodatos) del comportamiento de los equipos que integran la inteligencia del edificio, y de esta forma facilitar el aumento de la comodidad, protección y seguridad de sus ocupantes y un ahorro energético. El problema para implementar un sistema de gestión y control a un edificio es el costo del Hardware y software.

A nivel de Latino América las edificaciones ya están adoptando estas tecnologías poco a poco se está migrando a tener edificios supervisados y controlados servicios tales como la calefacción, iluminación, audio, accesos, ventilación y aire acondicionado, de forma que se garantiza su funcionamiento a niveles máximos de eficiencia y ahorro.

Las edificaciones en Ecuador son construidas de forma tradicional, es decir poseen una infraestructura con instalaciones no monitoreadas ni controladas, estas instalaciones por lo general son:

- Climatización (ventilación, calefacción y refrigeración).
- Alumbrado y fuerza.
- Instalación de Generación Eléctrica (gas, diésel).
- Comunicaciones (telefonía, TV, redes informáticas, Sonido, Video vigilancia, etc.).
- Instalaciones electromecánicas (Ascensores, Escaleras mecánicas, pasillos Instalación eléctrica rodantes, correo neumático, etc.).
- Instalación hidráulica (agua fría y agua caliente sanitaria).
- Instalaciones de gas (Gas LP o natural).
- Instalaciones de protección contra incendios.
- Instalación de sistema de alarma. (Sistema de seguridad), entre otras.

Lo que conlleva a un gasto Energético muy elevado en las edificaciones por ende a pagar mensualmente planillas de luz elevadas; tener un sistema de protección y seguridad obsoleto en las edificaciones por lo que estos sistemas trabajan de forma autónoma y no hay un control sobre ellos lo que puede conllevar a tener muchos riesgos y peligros.

Por tal razón es necesario establecer sistemas de control de edificios (BMS), para poder monitorear y controlar las instalaciones ya descritas anteriormente y de este modo proteger y brindar seguridad de una manera adecuada a los usuarios que utilizan diferentes edificaciones ya sean privadas o públicas, y disminuir considerablemente el gasto energético, de esa manera contribuiremos a la economía del país y al medio ambiente.

El sistema de control BMS que se pretende diseñar, permitirá la integración, gestión y el control de los diferentes dispositivos del bus Domótico HDL Buspro, lo que permitirá tener seguridad, confort, comunicaciones y eficiencia energética en una edificación.

Gracias a la utilización de un software libre donde se podrá realizar múltiples tareas y aplicaciones sin ningún límite ni restricciones, dependiendo lo que se requiera gestionar y controlar; al diseñar el sistema de control BMS con un software propietario conllevaría a pagar una elevada suma de dinero por las licencias y con limitadas tareas y funciones, si se requiere aumentar tareas y funciones al sistema BMS se debería pagar un adicional por las Tags adquiridas.

El presente tema de investigación, que se desarrolla a continuación está basado en la problemática descrita en los párrafos anteriores.

### **1.1. Formulación del problema**

¿Un confiable diseño de un sistema de control BMS (Building Management System) para la gestión del bus domótico HDL Buspro para la generación de macrodatos basado en software libre, garantizara la gestión y control correcto de una edificación?

### **1.2. Preguntas directrices**

¿Cómo funciona un sistema de control BMS?

¿Cómo envían los datos los equipos al bus domótico HDL?

¿Cómo almacenar los macrodatos adquiridos del sistema de control BMS?

¿Qué tipo de comunicación utiliza el sistema domótico HDL BUS PRO?

¿Cómo vincular y sincronizar los datos del bus domótico HDL con el sistema de control BMS?

### **1.3. Justificación del problema**

La evolución tecnológica de diferentes disciplinas, como la microelectrónica, las telecomunicaciones, la informática, la arquitectura y la automática, ha posibilitado una interacción de las mismas que ha desembocado en el concepto de edificio inteligente.(CRISTOBAL ROMERO, FRANCISCO VASQUEZ, 2009)

Al tener todas estas tecnologías integradas viene la necesidad de tener una gestión y un control sobre ellas, para aprovechar de una manera eficiente estas tecnologías y permitir una mayor calidad de vida a través de la tecnología, ofreciendo una reducción del trabajo doméstico, técnico y garantizar un mantenimiento al instante al momento que surja una falla en las instalaciones, un aumento del bienestar, confort y de la seguridad de sus habitantes, usuarios y una racionalización correcta del uso energético.

En la actualidad en nuestro país comienza a surgir paulatinamente edificaciones tanto públicas como privadas con ciertas características automatizadas e inteligentes algunas de estas son:

- Control de la climatización e iluminación en áreas comunes.
- Monitorización de ascensores y escaleras eléctricas.
- Control de riego en jardines.
- Sistemas de detección de incendios.
- Sistemas de acceso.
- Sistemas CCTV (circuito cerrado de televisión).

Lo que conlleva a crear y diseñar sistemas de gestión y control para tener un dominio sobre estas tecnologías que se van integrando de una forma muy rápida en nuestro país.

Al implementar sistemas que gestionen y controlen las edificaciones conseguimos un buen control de las condiciones de confort interior, posibilidad de control individual, el aumento de la productividad del personal, mejora la fiabilidad de la edificación y la vida, respuesta eficaz a las quejas relacionadas con la climatización, ahorro de tiempo y dinero durante el mantenimiento, el seguimiento eficaz y la orientación del consumo energético.

Tomando en cuenta esta creciente necesidad de energía, los objetivos nacionales del buen vivir recalcan en su objetivo número 11.1.q; Mantener actualizada una base de datos intersectorial de la oferta energética, los centros de transformación y los centros de consumo, para construir

balances energéticos y planificar el abastecimiento del país.(“Plan Nacional Buen Vivir 2013-2017.pdf,” n.d.)

Por otro lado el objetivo 11.1.t, también nos indica; Fortalecer la investigación científica en los ámbitos energéticos, de industrias básicas y de generación y uso sustentable de energía renovable, para la industria, los hogares, el transporte y la producción.(“Plan Nacional Buen Vivir 2013-2017.pdf,” n.d.)

### **DECRETO EJECUTIVO 1681**

21 de abril de 2009

**Art 1:** “Las entidades y organismos que conforman la Administración Pública Central e Institucional de la Función Ejecutiva realizarán el recambio a tecnologías eficientes en iluminación, datos comunicaciones, cargas de fuerza.....”(Red Iberoamericana de Gestión y Eficiencia Energética para un Desarrollo sostenible, 2011)

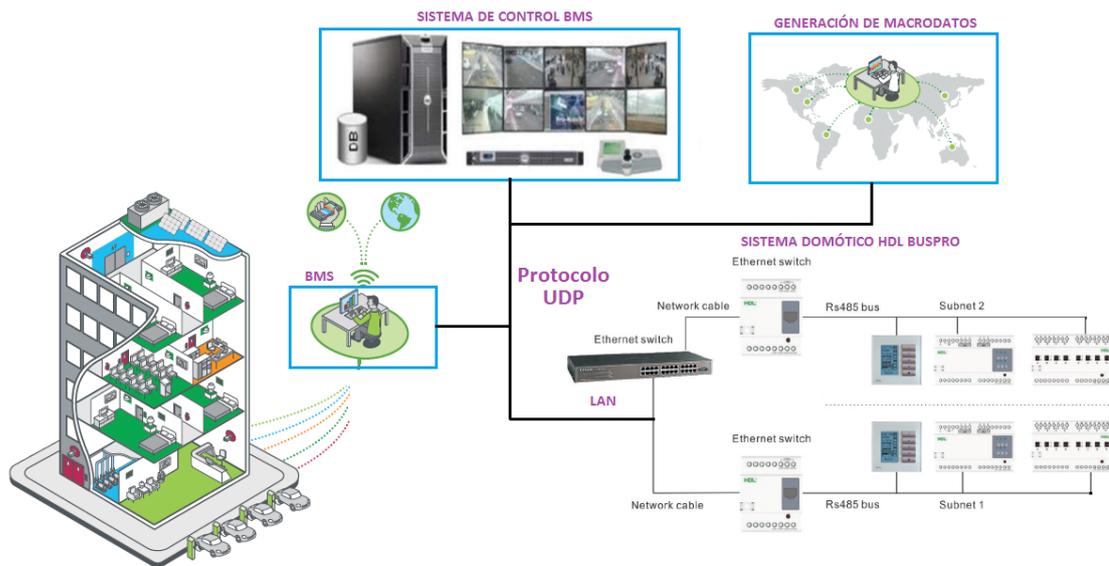
### **DECRETO EJECUTIVO 238**

28 de enero de 2010 Reforma al 1681

**Art 4:** “En caso de incumplimiento de las disposiciones (...) la autoridad competente procederá a imponer la sanción pertinente...”(Red Iberoamericana de Gestión y Eficiencia Energética para un Desarrollo sostenible, 2011)

Procurando estos objetivos, artículos y decretos ejecutivos, cualquier esfuerzo investigativo, que mejore la gestión eficiente del ahorro Energético, como el aquí propuesto, es de suma importancia para dar cumplimiento al plan nacional del buen vivir.

En la figura 1.1 se observa el diagrama del sistema de control BMS a implementar.



**Figura 1.1:** Diagrama del sistema de control BMS.  
**Realizado por:** Gómez Oscar, 2018.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Diseñar un sistema de control BMS (Building Management System) para la gestión del bus domótico HDL Buspro, para la generación de macrodatos basado en software libre.

### 1.4.2. Objetivos Específicos

- Estudiar el estado del arte para la generación de macrodatos, la comunicación y su vinculación con el bus de HDL, al sistema de control BMS.
- Investigar qué tipo de software libre resulta ser adecuado para el diseño de un sistema de control BMS.
- Diseñar un aplicativo basado en software libre para el sistema de control BMS para el bus domótico HDL Buspro.
- Garantizar la comunicación de los equipos dispuestos en el diseño del sistema de control BMS para el bus domótico HDL Buspro.

## **1.5. Hipótesis**

### ***1.5.1. Hipótesis General***

El sistema de registro permitirá almacenar la información de los equipos conectado en el bus, bajo los estándares de almacenamiento de macrodatos.

## CAPÍTULO II

### 2. MARCO TEÓRICO

#### 2.1. Antecedentes

A medida que la tecnología de la automatización, gestión y control de hogares y edificaciones avanza a pasos agigantados en otros países desarrollados no es excepción que a Latinoamérica lleguen estas nuevas tecnologías y las pongan en marcha no se diga en Ecuador, en la actualidad los hogar y edificación ya implementan sistemas de control y gestión ganando con esto confort, seguridad, accesibilidad y comunicaciones. Al tener estas tecnologías viene la necesidad de estudiar su arquitectura de instalación, funcionamiento, protocolos y normas; por lo que en las Universidades de nuestro país ya existen proyectos de investigación acerca de estas tecnologías.

En el país existe un solo trabajo de investigación que trata de integrar todas las diferentes tecnologías existentes en un edificio para su gestión, de Marcos Orbes Astudillo estudiante de la Universidad de Cuenca de la Facultad de Ingeniería (2012), quien realizo una tesis previa a la obtención del grado de Magister en Telemática titulado: “DISEÑO DE UN EDIFICIO INTELIGENTE”, el trabajo de investigación trata sobre el estudio de diferentes tipos de redes, protocolos, normas, arquitecturas de comunicaciones, sistemas de control para el diseño de edificaciones inteligentes aplicando diferentes tecnologías existentes en el mercado, para poder resolver los siguientes problemas.

- Mejores comunicaciones de voz, datos y video.
- Ahorro de la energía eléctrica.
- Mejor control de mantenimiento de las instalaciones de un edificio.
- Mayor seguridad, tanto para el edificio como para los ocupantes.
- Control de incendios.
- Control de estacionamiento y acceso vehicular.
- Control de ventilación.
- Eficiente uso de los ductos para electricidad y comunicaciones.
- Automatización y Control.

Este trabajo se relaciona con el tema de investigación planteado, ya que propone estudiar todos los sistemas de control que intervienen en un edificio inteligente para la gestión de diferentes tecnologías de automatización Domóticos e Inmóticos como son: Tecnología X10, Protocolo BIBus, Protocolo BACnet, Protocolos LonWorks.

## **2.2. Bases teóricas**

Al integrarse una variedad de equipos, mecanismos y sistemas en una edificación se crea la necesidad de poder controlar, gestionar y supervisar estas tecnologías de ahí es donde se crea el sistema de control BMS (Building Management System). Al tener estas tecnologías y sistemas instalados en edificaciones se ha realizado estudios e investigaciones del comportamiento de los sistemas de control, de la eficiencia y ahorro de la energía entre otras, por lo que se hace mención el siguiente estudio:

Cadena Salazar, Julio et al, 2015 en su trabajo de tesis para obtener el título de Máster en Ingeniería de la Energía especialidad en Gestión Energética, titulado “Eficiencia Energética en Edificaciones Empleando Sistemas de Control y Automatización basados en el Estándar Knx”, en el cual habla sobre las ventajas que se tiene al instalar e implementar tecnología de punta en las diferentes edificaciones, principalmente al automatizar los distintos sistemas de los edificios lo que conlleva al ahorro energético, debido que las edificaciones son las que consumen mayor cantidad de energía.

Para realizar el trabajo utiliza el estándar konnex (KNX) a nivel de gestión se ha escogido como herramienta el software NETx BMS Studio, que es un paquete de herramientas que incluyen NETx BMS Server y NETX Voyager. El primero es una solución que funciona como servidor OPC (OLE/COM5 for Process Control), una tecnología de protocolos de alto nivel para el desarrollo de aplicaciones de software que recoge los valores de los datapoints de diferentes sistemas (interoperabilidad), por ejemplo: KNX, BACnet, LonWorks y se los proporciona a los clientes del BMS. El segundo (Voyager) actúa como cliente del servidor OPC y se emplea para controlar y visualizar por medio de los datapoints los dispositivos de automatización y control instalados en una vivienda o edificación.(Becario, 2012)

## **2.3. Marco conceptual**

### **2.3.1. Domótica e Inmótica**

**Domótica.-** Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología. La domótica es una alta tecnología aplicable fundamentalmente a las viviendas (casa, departamentos, etc.) que permite dotar a las mismas de un sistema inteligente de gestión y control. Este tipo de instalaciones coordina distintos elementos electrónicos dentro de la vivienda para facilitar un uso central y controlado del confort del hogar. La tecnología de la domótica tiene el principal objetivo de brindar la mayor seguridad posible, con el mayor ahorro energético y calidad de confort.(Roció Basa, 2007)

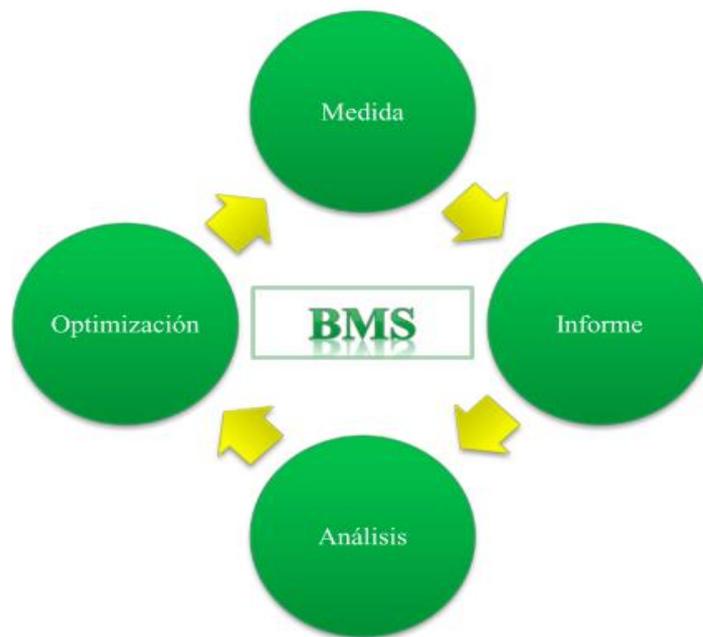
**Inmótica.-** Significa "sistema de gerenciamiento del edificio", es el que realmente controla y regula a un edificio, o sea su gestión total. Se pueden diferenciar ambos términos en que la Domótica es de aplicación para las viviendas y la Inmótica se incorpora a sistemas de gestión técnica automatizada a las instalaciones del sector de la producción y los servicios (plantas industriales, hoteles, hospitales, edificios de oficinas, aeropuertos, parques tecnológicos, bancos, universidades, etc.) con el objetivo de reducir el consumo de energía, aumentar el confort y seguridad de los mismos.(Rocio Basa, 2007)

Al estudiar el significado y las diferencias de estas dos ramas la domótica y la inmótica se puede concluir que a lo que se quiere llegar es a un fin común de aportar servicios de gestión energética, seguridad, bienestar y comunicaciones; al integrar un sistema de control BMS a una edificación se garantiza; gestión y medición de servicios, supervisión de las instalaciones, control de alarmas, copia de seguridad cuando existe fallos en el sistema, respaldo remoto ininterrumpido no se necesita técnicos en campo, supervisión remota, mantenimiento eficiente de los equipos en función de las condiciones.

### **2.3.2. Sistemas de Control BMS**

El Building Management System (BMS) es un sistema de gestión de edificaciones, supervisan y controlan servicios tales como la calefacción, ventilación y aire acondicionado, de forma que se garantiza su funcionamiento a niveles máximos de eficiencia y ahorro. Esto se consigue gracias a que se mantiene un equilibrio óptimo entre las condiciones, uso energético y requisitos operativos.(TREND, 2016)

En la figura 1.2, se muestra las principales gestiones que realiza el sistema BMS.



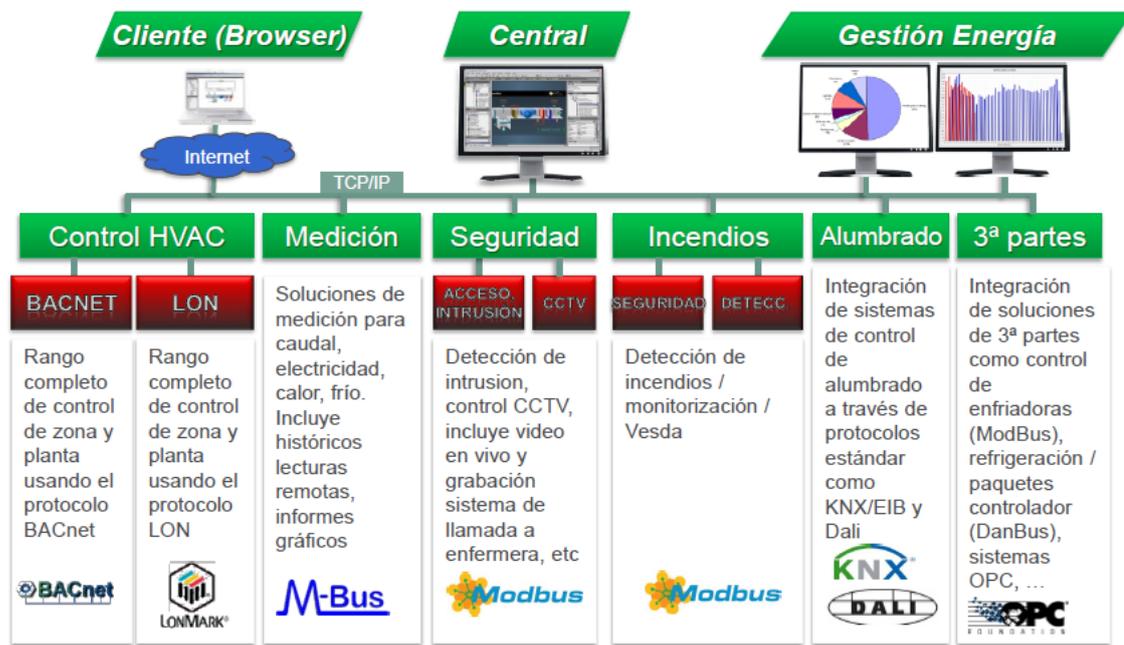
**Figura 1.2:** Principales Gestiones de un sistema de control BMS.  
**Fuente:** (Carnero, 2014)

Los sistemas que suelen integrarse a un BMS son:

- Iluminación.
- Calefacción.
- Ventilación.
- Sistemas de subestación eléctricas.
- Aire acondicionado y calefacción.
- Sistemas de seguridad.
- Sistemas contra incendio.
- Comunicación.
- Control de accesos.
- Suministros energéticos.
- Sistema hidráulico.
- sistemas especiales (sistema de riego, sistema neumático).
- Sistemas de trasporte vertical (ascensores, escaleras electromecánicas).

### 2.3.3. Arquitectura del sistema de control BMS

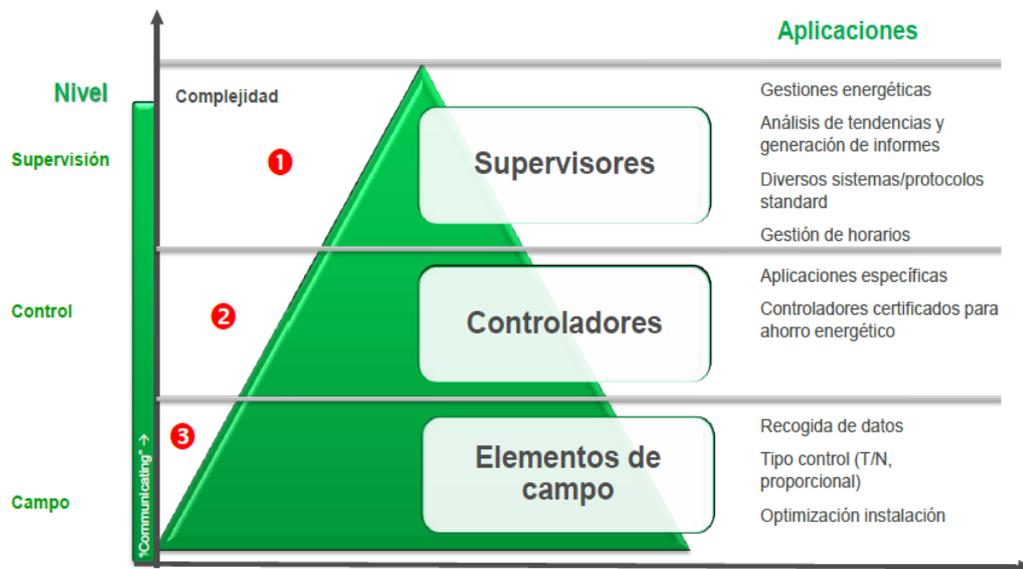
El tipo de arquitectura que utiliza los sistemas de control BMS es una arquitectura Distribuida; la inteligencia del sistema está distribuida por todos los módulos sean sensores o actuadores, suele ser típico de los sistemas de cableado en bus. En la figura 2.2, se muestra la arquitectura del sistema BMS y algunas gestiones que este realiza.



**Figura 2.2:** Arquitectura del sistema BMS.  
Fuente: (Carnero, 2014)

### 2.3.4. Niveles de Control que trabaja el sistema BMS

En los niveles de control que trabajan los sistemas BMS, son de supervisión, control y campo, en la figura 3.2, se observa cómo están constituida los niveles del control que trabaja el sistema BMS y algunas de sus aplicaciones.



**Figura 3.2:** Niveles de control del sistema BMS y aplicaciones.  
Fuente: (Carero, 2014)

### 2.3.5. Funciones y aplicaciones de los niveles de control

#### 2.3.5.1. Nivel 1 Supervisión

Las aplicaciones que se puede tener en este nivel de supervisión son las siguientes:

- Gestiones energéticas.
- Análisis de tendencias y generaciones de informes.
- Diversos sistemas, protocolos y estándar.
- Gestión de horarios.

En la figura 4.2, se observa las funcionalidades del nivel de supervisión.(Carero, 2014)



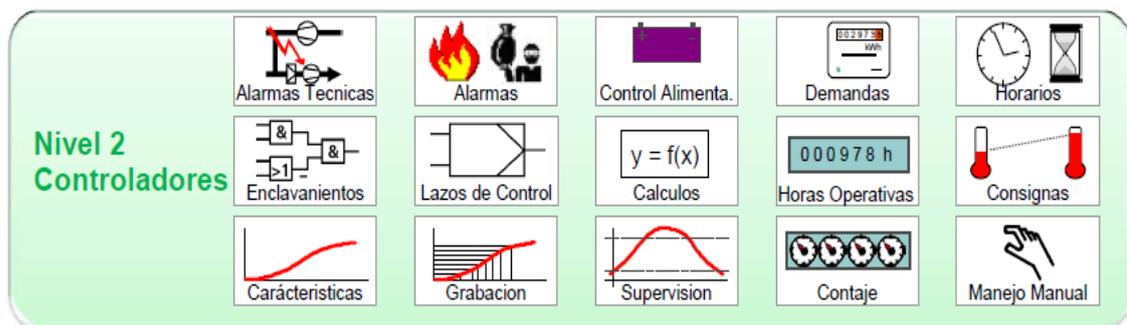
**Figura 4.2:** Nivel 1 de supervisión del sistema BMS.  
Fuente: (Carero, 2014)

### 2.3.5.2. Nivel 2 Control

Las aplicaciones que se puede tener en este nivel de control son las siguientes:

- Aplicaciones específicas (técnicas de control).
- Controladores.

En la figura 5.2, se observa las funcionalidades del nivel de control.(Carnero, 2014)



**Figura 5.2:** Nivel 2 de control del sistema BMS.

**Fuente:** (Carnero, 2014)

### 2.3.5.3. Nivel 3 Campo

Las aplicaciones que se puede tener en este nivel de control son las siguientes:

- Adquisición de datos.
- Tipo control (T/N, proporcional).
- Optimización instalación.

En la figura 6.2, se observa las funcionalidades del nivel de campo.(Carnero, 2014)



**Figura 6.2:** Nivel 3 de campo del sistema BMS.

**Fuente:** (Carnero, 2014)

### **2.3.6. Sistema Domótico HDL Buspro**

Buspro es un Sistema de automatización para casas/edificios desarrollado por HDL puede ser utilizado para control de iluminación, cortinas, HVAC, TV, música, dispositivos de teatro en casa, seguridad, riego, Medición de energía, etc.(BUSPRO, 2016)

### **2.3.7. Ventajas del Sistema HDL Buspro**

- Simplificación en los diseños de los circuitos de iluminación, fáciles de instalar y dar mantenimiento.
- Estructura de hardware y software flexible.
- Múltiples formas de control de iluminación, tales como, escenas en un área simple y escenas en áreas combinadas, interruptores temporizados, operaciones secuenciales, regulación automática de iluminación, control mediante señales IR, etc.
- Las escenas de iluminación pueden ser cambiadas sin necesidad de modificar el hardware, modificando solamente el software.
- Cada dispositivo en la red es independiente, de forma que si un dispositivo sufre un daño el resto de la red sigue funcionando con normalidad.

### **2.3.8. Sistema de Comunicación.**

El Bus domótico HDL Buspro utiliza el estándar industrial RS-485 se clasifica como red troncal basada en Ethernet y RS-485 basado en la red de bus bidireccional, en la red troncal puede adoptar topología en estrella, topología de árbol y topología de anillo, mientras que la red de bus sólo puede adoptar topología de bus. La red bus del sistema del HDL-bus adopta el modo del megabus RS-85, que hace el megabus en mejores condiciones en cualquier momento. Simultáneamente funciona con la exclusiva tecnología de control inteligente HDL CSMA / CD que se puede combinar con software / hardware para asegurar que puede obtener la velocidad máxima de transferencia independientemente de la cantidad de dispositivos de bus y la distancia de bus en el sistema. El sistema de HDL-BUS adopta el protocolo abierto y alto de la escalabilidad, que puede ligar el sistema con cualquier sistema de control sin problemas. Por ejemplo: sistema BA, sistema de control central, sistema de seguridad, sistema de lectura de contador remoto y sistema de control de iluminación de escenario, etc.(“Design Guidance Intelligent building control system SINCE 1985 HDL-BUS,” 2016)

El sistema de comunicación utiliza un puerto físico con las siguientes características:

- Cable KNX estándar o CAT5 (superior).
- (Tasa de transmisión) Baud rate: 9600 bps.
- Longitud máxima Subnet: 1200mts.
- Voltaje de Bus: 25 VDC.
- Mecanismo de detección de conflictos. CSMA/CD.

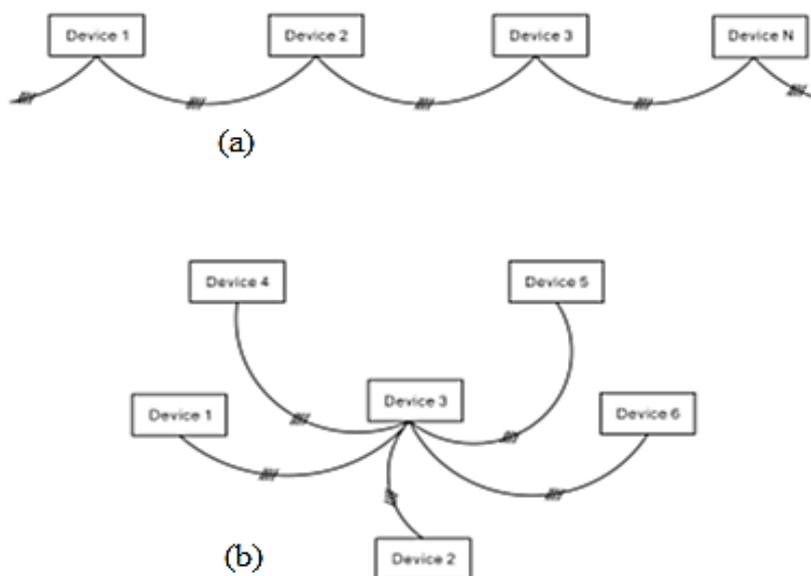
En la figura 7.2, se observa el puerto Buspro.



**Figura 7.2:** Puerto Buspro el conector tiene D+, D-, COM y 24VDC.  
Fuente: (BUSPRO, 2016)

### 2.3.9. Topología del Sistema

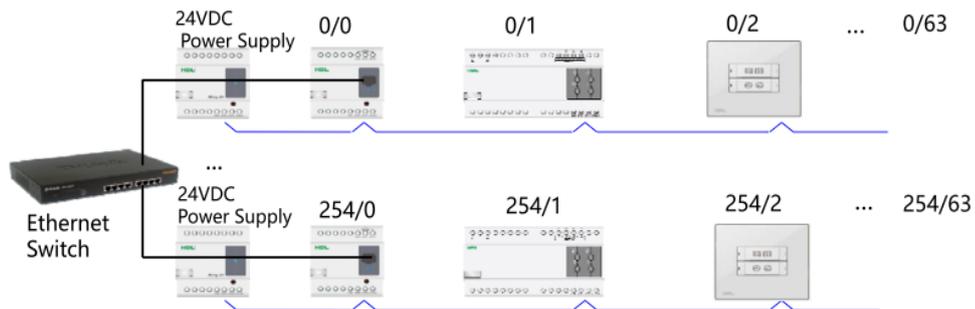
Las topologías aceptadas por el sistema HDL BUS-PRO para conectar los equipos son tipo bus y tipo estrella, aunque la topología en estrella no es muy recomendada. En la figura 8.2, se puede observar las dos topologías aceptadas por el sistema HDL BUS-PRO.



**Figura 8.2:** (a) Topología tipo bus, (b) topología tipo estrella.  
Fuente: (“Design Guidance Intelligent building control system SINCE 1985 HDL-BUS,” 2016)

Cada dispositivo tiene Subnet ID y Device ID excepto la Fuente de poder, ejemplo, 1/2 significa la subnet ID=1, Device ID=2, esta combinación hace única a cada dispositivo dentro de toda la red. La capacidad del sistema es de  $255 \times 64 = 16,320$ . (BUSPRO, 2016)

En la figura 9.2, se observa la topología del sistema.



**Figura 9.2:** Topología del sistema.  
Fuente: (BUSPRO, 2016)

La topología en anillo no es permitida, si dos Subnets son conectadas entre sí por el bus de datos como se muestra en la figura 10.2, cuando se genera un comando, este se quedara circulando en el anillo, por siempre lo cual sobrecargara el bus y dará fallo en el sistema, La topología en estrella tampoco es recomendable en grandes redes como un edificio donde un cable de la Subnet puede ser aproximarse a 1 Km, el cableado siempre produce baja señal de transmisión, resultado de una lata cantidad de ruido (SNR), el cableado recomendado es el punto a punto (Hand in hand). (BUSPRO, 2016)



**Figura 10.2:** La Topología en anillo no es permitida.  
Fuente: HDL Buspro abril 6, 2015.

### 2.3.10. Componentes del sistema HDL Buspro

La marca HDL Buspro tiene una alta gama de controladores, sensores y actuadores que son específicamente para la automatización de hogares y edificios.

En la tabla 1.2 se en listan varios componentes del sistema de HDL Buspro.

**Tabla 1.2:** Algunos componentes de la marca HDL Bus Pro.

| COMPONENTE  | CARACTERÍSTICAS   | FUNCIÓN   |
|---|---|---|
| <p><b>HDL DLP Panel</b></p>                                | <p>Interfaz de usuario con pantalla de cristal líquido, cuenta con 8 botones programables y cuatro pestañas de navegación.</p>  | <p>Control de todos los dispositivos contenidos en el bus domótico, posee una pestaña dedicada para control de audio y otra para el sistema de climatización.</p>   |
| <p><b>Módulo de control de la condición del aire</b></p>  | <p>Avanzado módulo de tecnología que se instalan cerca de la unidad en sí para ahorrar el costo de cableado largo a la ordinaria Termostato y reemplazar el termostato feo en la pared, es habilitado y controlado por el SB-DLP panel que han construido en el sensor de temperatura y aire acondicionado controlar.</p> | <p>5 canales de salida: 5A Relé Todo el Compresor y la potencia del ventilador 24VAC, 110VAC, 220VAC se pueden conectar al relé de paso 5A Velocidad del ventilador: alta, media baja Modo: Enfriamiento, calefacción, humidificador, deshumidificador Control Compresor de una o varias etapas.</p>      |
| <p><b>Fuente de poder</b></p>                            | <p>Entrada 110V, 50/60Hz, salida de 2400mA y 750mA.</p>   | <p>Provee de alimentación al bus Domótico.</p>  |
| <p><b>Phase digital Power meter</b></p>                  | <p>Es un equipo digital inteligente de tres fases medidor que mide y registra los parámetros eléctricos.</p>  | <p>Puede ser utilizado para la grabación del consumo de energía. Puede grabar y guarde los parámetros eléctricos del último año. Y la corriente, voltaje, potencia y factor de potencia también se pueden medir por el metro. La corriente, voltaje, consumo de energía, tiempo, interruptor UV, etc.</p> |

|   |   |  |
|---|---|--|
| <p><b>Módulo de Relé</b></p>             | <p>Módulo de relé de 4 canales, 10 amperios por canal.</p>  | <p>Control de iluminación, soporta escenas y secuencias.</p>   |
| <p><b>Módulo dimmer</b></p>              | <p>Módulo dimmer de 6 canales, 2 amperios por canal.</p>  | <p>Control de iluminación dimmerizable para cargas resistivas.</p>   |
| <p><b>Power interface, para DLP</b></p>  | <p>Voltaje 24VDC, Data+, Data.</p>  | <p>Provee alimentación y el bus de comunicación para el DLP.</p>   |
| <p><b>Z- Audio</b></p>                  | <p>Acepta memoria SD hasta 8Gb, reproduce desde servidor FTP, radio FM, entrada auxiliar, salida amplificada de 50W*2.</p>  | <p>Sistema centralizado de audio.</p>  |
| <p><b>Sensor 8 en 1</b></p>            | <p>Capaz de medir temperatura, luminosidad, movimiento, además posee emisores y receptores infrarrojos.</p>   | <p>Utilizado para seguridad y para controlar electrodomésticos que funcionen con control remoto.</p>   |
| <p><b>Módulo Lógico</b></p>            | <p>Este módulo cuenta con 960 bloques lógicos, y la condición de la lógica de las entradas puede establecer el estado de un canal, fecha, semana, hora, valor de una entrada externa de una escena, etc.</p>  | <p>Mediante el uso de diferentes relaciones lógicas para establecer diferentes destinos de control, cuatro relaciones lógicas están disponibles para cada bloque lógico: AND, OR, NAND, NOR.</p>                       |
| <p><b>Módulo IP</b></p>                | <p>Puerto de comunicación Ethernet y Bus Pro.</p>   | <p>Permite la interacción con el bus a través de la red LAN.</p>   |
| <p><b>Módulo compacto RCU</b></p>      | <p>Un total de 24 entradas de contacto seco pueden utilizarse como objetivos de control a través del software de gestión de hoteles. También permite que las entradas de contacto seco funcionen con las salidas de LED para administrar y controlar timbres de terceros.</p> | <p>Timbre LED indicador es compatible y controlado a través de 5 canales dedicados de la RCU, esto permite que el timbre para mostrar el número de la habitación, y la lavandería, la limpieza y el estado de DND.</p> |

Fuente: ("Design Guidance Intelligent building control system, HDL BUS," 2015)



### **2.3.13. Selección del Software libre**

Para seleccionar el software libre primero vamos a dar una pequeña definición de software libre; Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos. (HISPA LINUX, 2011)

En la actualidad existen diferentes softwares libres para realizar diferentes aplicaciones, para realizar nuestro sistema de control BMS hablaremos de 3 softwares orientados a objetos, sus principales características y escogeremos el que más se asemeja a nuestras necesidades los cuales son Python, Ruby, y Java NetBeans IDE.

#### **2.3.13.1. Principales Características del Lenguaje Python**

Python es un lenguaje de programación creado por Guido van Rossum a finales de los ochenta, y que gracias a sus características ha llegado a ser un lenguaje muy conocido en la actualidad. A continuación se listan las principales características que este software posee: (Carolina Valencia Gil, Carolina Henao Acosta, 2010)

- **Simple:**

Python es un lenguaje muy simple, por lo que es muy difícil iniciarse en este lenguaje. El pseudocódigo natural de Python es una de sus grandes fortalezas.

- **Propósito General:**

Usando el lenguaje Python se puede crear todo tipo de programas; programas de propósito general y también se pueden desarrollar páginas Web.

- **Open Source:**

Debido a la naturaleza de Python de ser Open Source; ha sido modificado para que pueda funcionar en diversas plataformas (Linux, Windows, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE y PocketPC). Al ser Open Source es gratuito.

- **Lenguaje Orientado a Objetos:**

Al ser un Lenguaje Orientado a Objetos es construido sobre objetos que combinan datos y funcionalidades.

- **Extensas Librerías:**

Python contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero, las librerías pueden ayudar a hacer varias cosas como expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web, CGI, ftp, correo electrónico, XML, XML-RPC, HTML, archivos WAV, criptografía, GUI, y también otras funciones dependientes del Sistema.(Carolina Valencia Gil, Carolina Henao Acosta, 2010)

### *2.3.13.2. Principales Características del Lenguaje Ruby*

En ruby, se combinan las capacidades de los lenguajes funcionales con las de los lenguajes imperativos orientados a objetos, de esta manera cada elemento en ruby tiene un compartimiento similar, pero personalizado para las conveniencias de cada usuario.(Ken Constantino Daniel Fernando, 2010)

Sus principales características son las siguientes:

- Tecnología útil, de fácil implementación.
- Su éxito radica en la rapidez de sus aplicaciones.
- El planteamiento es distinto al resto de tecnologías.
- Cuenta con un aprendizaje más rápido que en otros lenguajes.
- En expansión.
- No dispone de mucha documentación ya que es relativamente nuevo.
- Cuenta con código libre.
- Tiene una extensa comunidad detrás de él que la respalda.
- Favorece en el ahorro de líneas de código.

### *2.3.13.3. Principales Características de JAVA NetBeans IDE*

NetBeans IDE es un entorno de desarrollo integrado (IDE), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación.

El lanzamiento de NetBeans IDE provee varias características, tales como mejores características de edición JavaScript, soporte para usar estructuras Spring de soporte web, integración MySQL más ajustada y una mejor manera de compartir librerías entre proyectos dependientes. El aclamado soporte para Ruby/JRuby ha sido mejorado con un nuevo editor de soluciones rápidas (Quick Fix), un administrador para la plataforma Ruby, soporte para depuración rápida en JRuby y muchas otras nuevas características como son:

- Soporte JavaScript
  - Sintaxis Resaltada y
  - Completación de código y análisis de tipo
  - Soluciones Rápidas (Quick Fixes) y verificación de sintaxis
  - Refactorización
- Mejoras en el Desempeño
  - Inicio hasta 40% más rápido
  - Promociones más inteligentes, así que la competición de código es más rápida
  - Menor consumo de memoria
- Soporte MySQL en exploración de bases de datos
  - Registro de servidores MySQL
  - Ver, crear y borrar bases de datos
  - Fácil lanzamiento de administración para MySQL
- Java Mobility (Aplicaciones para móviles)
  - Emular Mpowerplayer MIDP para aplicaciones MIDP en MacOS X (disponible en el centro de extensiones)
  - Estructurador SVG (SVG composer para componentes SVG de uso frecuente)
  - Documentación y estabilidad mejorada

(“Información NetBeans IDE 6.1,” 2015)

Al estudiar algunos softwares que se asemejen a las características que necesitamos para el desarrollo del sistema de control BMS se escogió trabajar con JAVA NetBeans IDE la programación es basada en objetos, la principal característica por la cual se escogió JAVA es la facilidad que permite para crear, borrar los macrodatos que son adquiridos del sistema BMS.

#### 2.3.14. *Macrodatos (Big Data)*

Los datos que genera el sistema de control BMS se guardará en una base de datos generando un Big Data para su posterior estudio del comportamiento de las señales eléctricas, para ello se habla un poco de la definición de los macrodatos o Big Data.

Big data, macrodatos, datos masivos, inteligencia de datos o datos a gran escala es un concepto que hace referencia a un conjunto de datos tan grandes que aplicaciones informáticas tradicionales de procesamiento de datos no son suficientes para tratar con ellos y los procedimientos usados para encontrar patrones repetitivos dentro de esos datos. (macrodatos e inteligencia de datos, alternativas a big data, 2016)

Para trabajar con los macrodatos existen muchas herramientas como son:

- Cassandra
- NoSQL
- Avro
- Flume
- HBase
- MySQL
- ZooKeeper
- Jaql

Estas herramientas trabajan con los tres tipos de macrodatos:

- **Datos estructurados:** datos que tienen bien definidos su longitud y su formato, como las fechas, los números o las cadenas de caracteres. Se almacenan en tablas. Un ejemplo son las bases de datos relacionales y las hojas de cálculo.
- **Datos no estructurados:** datos en el formato tal y como fueron recolectados, carecen de un formato específico. No se pueden almacenar dentro de una tabla ya que no se puede desgranar su información a tipos básicos de datos. Algunos ejemplos son los PDF, documentos multimedia, correos electrónicos o documentos de texto.
- **Datos semiestructurados:** datos que no se limitan a campos determinados, pero que contiene marcadores para separar los diferentes elementos. Es una información poco regular como para ser gestionada de una forma estándar. Estos datos poseen sus propios metadatos semiestructurados que describen los objetos y las relaciones entre ellos, y pueden acabar siendo aceptados por convención. Un ejemplo es el HTML, el XML o el JSON. (macrodatos e inteligencia de datos, alternativas a big data, 2016)

Los datos que se guardarán del sistema de control BMS son datos estructurados porque se tienen datos bien definidos como son: las fechas, la hora, voltaje, potencia, corriente, consumo energético, factor de potencia, alarmas.

### 2.3.15. Protocolo de comunicación UDP

UDP (User Datagram Protocol), el protocolo de datagrama de usuario es un estándar TCP/IP que está definido en RFC 768, el bus domótico HDL Buspro trabaja con este protocolo la principal característica es el transporte rápido de datos.

UDP proporciona un servicio de datagramas sin conexión que ofrece entrega de mejor esfuerzo, lo que significa que UDP no garantiza la entrega ni comprueba la secuencia de los datagramas lo que hace que la garantía de que un paquete llegue a su destino sea menor que si se usara TCP.

#### 2.3.15.1. Puertos UDP

Los puertos UDP proporcionan una ubicación para enviar y recibir mensajes UDP. Un puerto UDP funciona como una única cola de mensajes que recibe todos los datagramas destinados al programa especificado mediante cada número de puerto del protocolo. Es decir, los programas basados en UDP pueden recibir varios mensajes a la vez.

El lado de servidor de cada programa que utiliza UDP atiende los mensajes que llegan a su número de puerto conocido. Todos los números de puerto de servidor UDP inferiores a 1.024 (y algunos números superiores) están reservados y registrados por la Autoridad de números asignados de Internet (IANA, Internet Assigned Numbers Authority).

Cada puerto de servidor UDP se identifica mediante un número de puerto conocido o reservado. En la tabla 2.2 se muestra una lista parcial de los números de puerto de servidor UDP conocidos que utilizan programas basados en UDP estándar. (Molina, 2014)

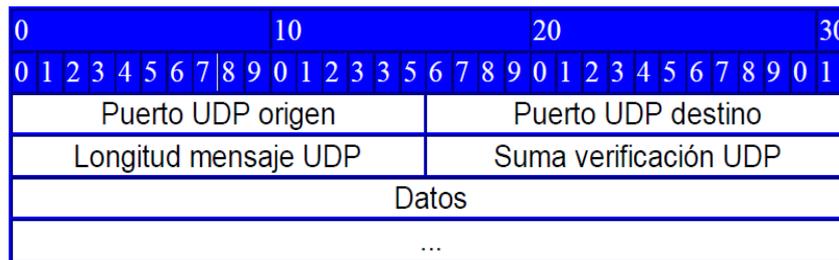
**Tabla 2.1:** Puertos UDP

| Número de puerto UDP | Descripción   |
|----------------------|---|
| 53                   | Consultas de nombres DNS  |
| 69                   | Protocolo trivial de transferencia de archivos (TFTP)                       |
| 137                  | Servicios de nombres NetBIOS  |
| 138                  | Servicios de datagramas NetBIOS   |
| 161                  | Protocolo simple de administración de redes (SNMP)                          |
| 520                  | Protocolo de información de enrutamiento (RIP-Routing Information Protocol) |

Fuente: (Molina, 2014)

### 2.3.15.2. Formato del mensaje UDP

- **Puerto UDP de origen:** (16 bits, opcional), Número de puerto de la maquina origen.
- **Puerto UDP de destino:** (16 bits), Numero de puerto de la maquina destino.
- **Longitud del mensaje UDP:** (16 bits), Especifica la longitud medida en bytes del mensaje UDP incluyendo la cabecera. La longitud mínima es de 8 bytes.
- **Suma de verificación UDP:** (16 bits, opcional), Suma de comprobación de errores del mensaje. Para su cálculo se utiliza una pseudo-cabecera que también incluye las direcciones IP origen y destino. Para conocer estos datos, el protocolo UDP debe interactuar con el protocolo IP.
- **Datos:** Aquí viajan los datos que se envían las aplicaciones. Los mismos datos que envía la aplicación origen son recibidos por la aplicación destino después de atravesar toda la red de redes, en la figura 12.2 se observa el formato del mensaje UDP.(User & Protocol, 2014)



**Figura 12.1:** Formato del mensaje UDP.

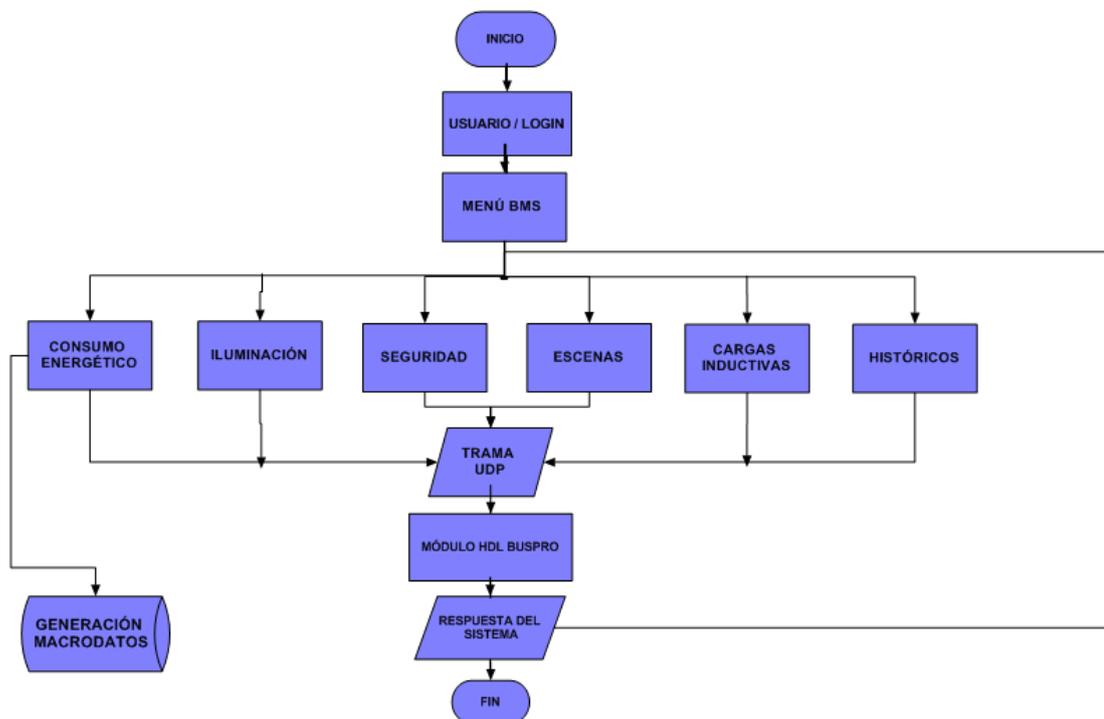
**Fuente:** (User & Protocol, 2014)

## CAPÍTULO III

### 3. DESARROLLO DEL SISTEMA DE CONTROL BMS

#### 3.1. Instalación de los softwares

Para entender de una mejor manera como va estar constituido el sistema de control BMS en la figura 1.3, se observa el diagrama de flujo del funcionamiento del sistema.



**Figura 1.3:** Diagrama de flujo del sistema de control BMS a implementar.

Realizado por: Gómez Oscar, 2018.

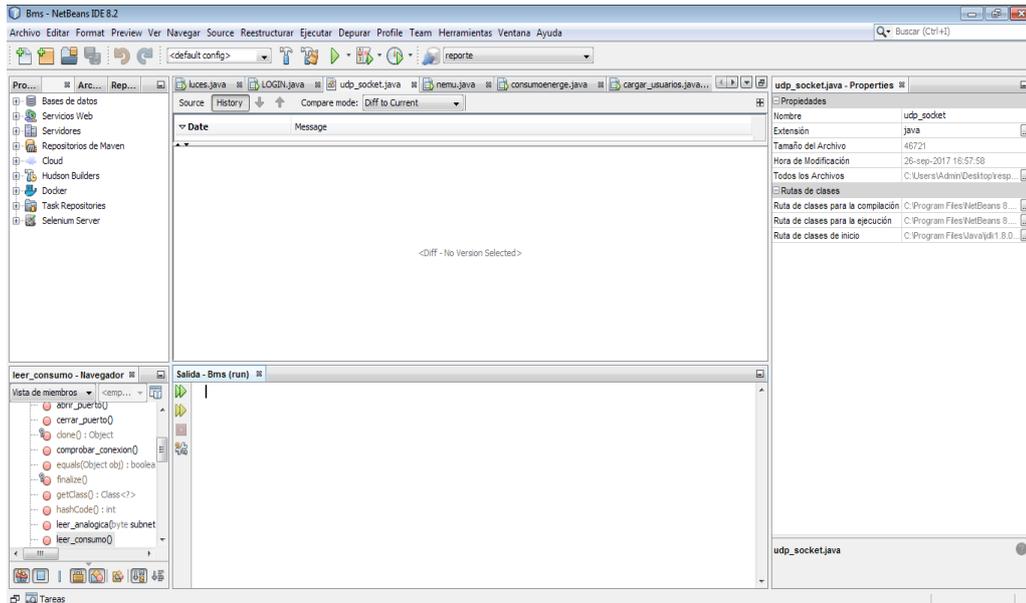
#### 3.1.1. Oracle Java JDK 8u111 (Java SE Development Kit)

Para empezar al desarrollo del BMS se necesita instalar los softwares necesarios en este caso primero se instala JAVA JDK 8u111 de 64 bits este software contiene un conjunto de programas y librerías que permiten desarrollar programas en Java.

Nos permite compilar, generar documentos, depurar, monitorear ejecutar los programas.

### 3.1.2. NetBeans IDE 8.2 (Entorno de Desarrollo Integrado)

Una vez instalado el JDK de Java se procede a instalar el entorno de trabajo NetBeans IDE la versión 8.2, este programa es de código abierto es decir software libre orientado a objetos; el entorno de trabajo de NetBeans se observa en la figura 2.3, ahí se procederá a diseñar la interfaz gráfica y la comunicación UDP del sistema de control BMS.



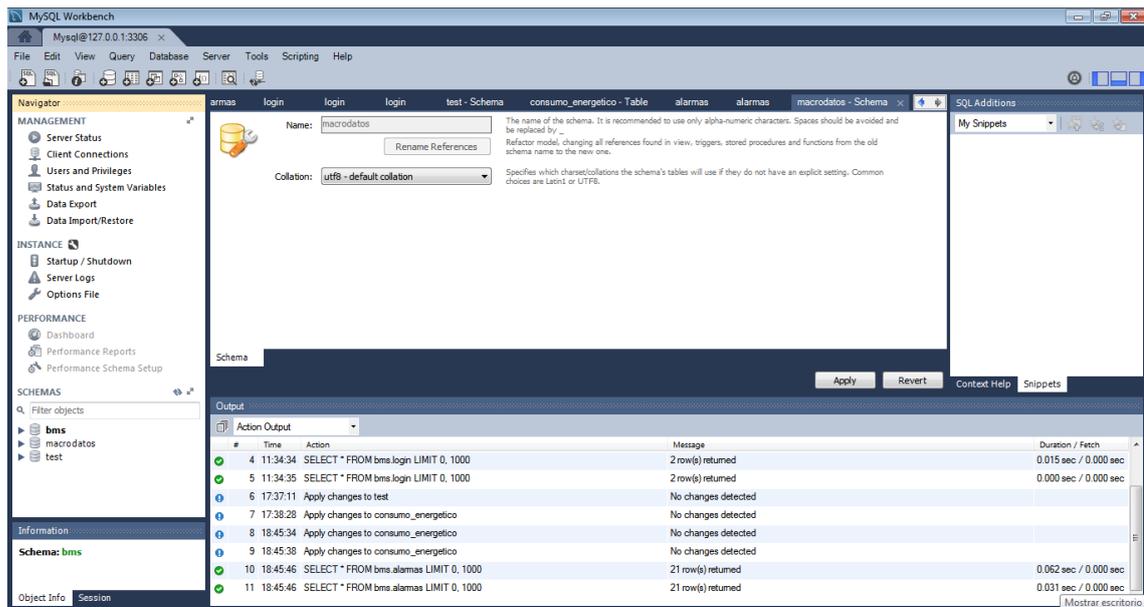
**Figura 2.3:** Entorno de trabajo NetBeans IDE 8.2.  
Realizado por: Gómez Oscar, 2018.

### 3.1.3. MySQL version 5.5.53.0 (Database Management System)

El sistema de control BMS genera un sin número de datos cada segundo estos también llamados macrodatos (Big Data), a los cuales se necesita guardarlos en una base de datos en este caso se utilizó el sistema de administración de base de datos MySQL versión 5.5.53.0 el cual permite guardar todo tipo de datos algunos de estos son: float, int, char, enum, double, decimal, real, varchar, longtext, entre otros.

### 3.1.4. Workbench 6.3 CE

Este software permite visualizar los esquemas de la base de datos con esta herramienta se crea las tablas donde se guardarán los macrodatos del BMS, el programa permite comunicarse con el entorno de trabajo de NetBeans para adquirir los datos y llenar las tablas; en la figura 3.3 se observa el entorno de trabajo de Workbench.



**Figura 3.1:** Entorno de trabajo Workbench 6.3 CE.  
**Realizado por:** Gómez Oscar, 2018.

### 3.2. Diseño de la interfaz del sistema bms

El sistema de control BMS a diseñar constará de diferentes ventanas las cuales tendrán fusiones específicas en la tabla 1.3 se describe un breve resumen de las funciones principales que hace cada una de ellas.

**Tabla 1.3:** Descripción de funcionamiento de las ventanas del sistema BMS.

|   | Nombre de las ventanas                   | Función de las ventanas  |
|---|--|--|
| 1 | Acceso al sistema de control BMS (Login) | Permite acceder al sistema BMS con un usuario.   |
| 2 | Crear nuevo usuario                      | Permite crear nuevos usuarios con jerarquías de acceso.  |
| 3 | Menú del sistema de control BMS          | Permite navegar hacia las diferentes funciones del sistema BMS.  |
| 4 | Consumo energético                       | Permite visualizar el consumo de todo el sistema que se encuentra conectado con el BMS.  |
| 5 | Iluminación                              | Permite manipular las cargas de iluminación; focos y dimmers.  |
| 6 | Seguridad                                | Permite visualizar los estados de los sensores y la cámara de seguridad.   |
| 7 | Escenas                                  | Permite cambiar de ambiente a diferentes zonas incluido audio.   |
| 8 | Cargas inductivas                        | Permite manipular las cargas inductivas (motores), entradas y salidas analógicas (1-10V y 4 a 20 mA) que está conectado al sistema BMS, acceder a las alarmas que se generan e imprimir el reporte de alarmas. |
| 9 | Históricos                               | Permite observar en tiempo real el consumo energético y guarda la gráfica característica, permite acceder a los reportes de los macrodatos que guarda en la base datos el BMS e imprimirlos.                   |

Realizado por: Gómez Oscar, 2018.

### 3.2.1. Ventana de acceso al sistema de control BMS

Esta ventana permite acceder al BMS por seguridad con un usuario y una clave previamente creada por un usuario con una jerarquía máxima, el sistema BMS en este caso tiene 4 jerarquías que se puede escoger al momento de crear un nuevo usuario las cuales son:

- Gerente
- Administrador
- Supervisor
- Técnico

La ventana para acceder al BMS está configurada, que únicamente el gerente pueda crear nuevos usuarios mientras que el administrador, supervisor y el técnico no tienen esa opción en la figura 4.3, se puede ver la configuración de la ventana.



**Figura 4.3:** Ventana de acceso al sistema BMS.  
Realizado por: Gómez Oscar, 2018.

### 3.2.2. *Venta para crear un nuevo usuario*

En todos los sistemas de control existen opciones para crear nuevos usuarios para que otras personas puedan acceder al sistema a supervisar y controlar los distintos sistemas que se encuentra integrados entre ellos, en la figura 5.3, se puede observar la venta para crear nuevos usuarios para el sistema BMS el cual solo puede acceder el gerente en este caso.



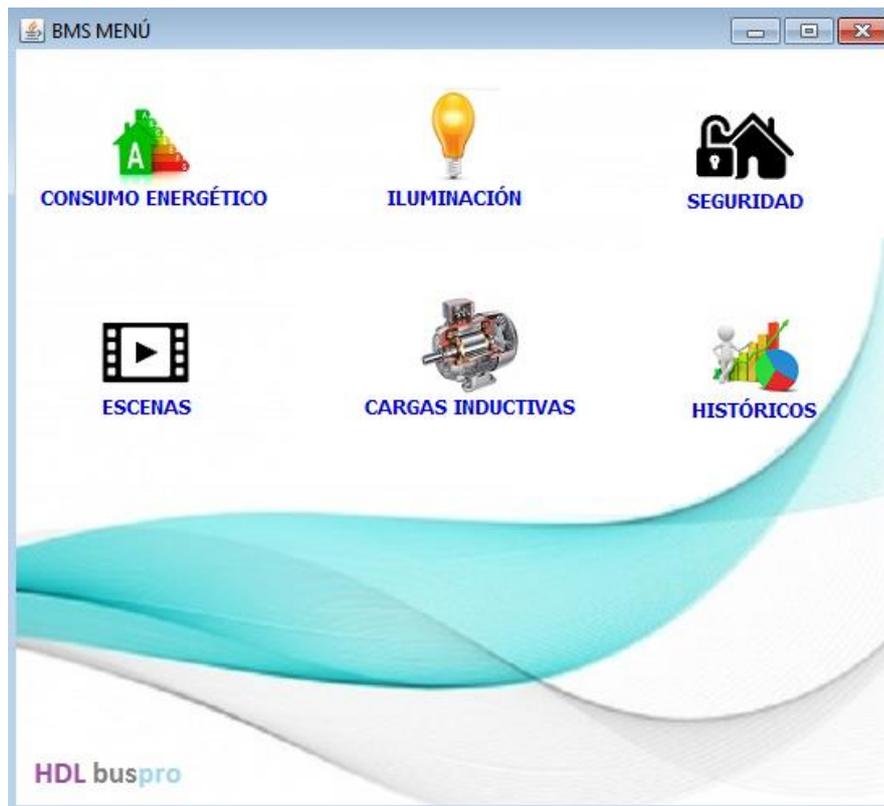
**Figura 5.3:** Ventana para crear usuarios nuevos.  
Realizado por: Gómez Oscar, 2018.

### 3.2.3. Ventana de menú del sistema de control BMS

Para el sistema BMS se realizó una ventana de menú donde los operadores pueden navegar por las diferentes funciones que tiene el sistema BMS las cuales son las siguientes:

- Consumo energético
- Iluminación
- Seguridad
- Escenas
- Cargas inductivas
- Históricos

En la figura 6.3 se observa la venta del menú del sistema BMS.



**Figura 6.3:** Ventana del menú del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

#### 3.2.4. *Ventana del consumo energético*

En esta ventana se puede observar las variables de consumo eléctrico por cada línea es decir el sistema nos permite ver el estado de las variables por fase si tenemos sistemas monofásicos bifásicos y trifásicos podemos medir sin problema, las siguientes variables eléctricas que el sistema mide son las siguientes:

- Voltaje [V]
- Corriente [A]
- Potencia activa [Kw]
- Potencia aparente [kw]
- Potencia reactiva [Kw]
- Factor de potencia

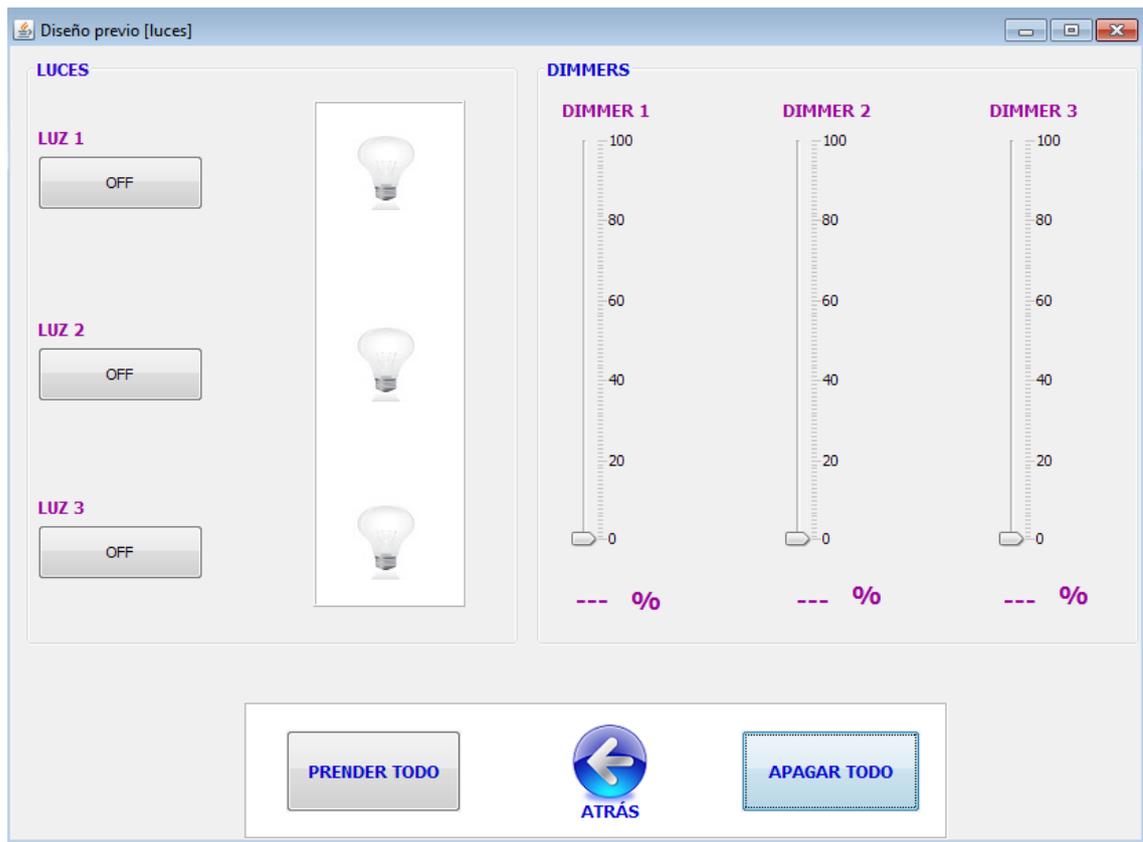
En la figura 7.3 se puede observar la ventana realizada.



**Figura 7.3:** Ventana de consumo eléctrico del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

### 3.2.5. Ventana de iluminación

Esta ventana contiene todas las cargas de iluminación que están integradas al sistema BMS desde la cual se monitorea y se controlan las mismas, en el BMS existen 2 clases de iluminación la normal que enciende luces de 120[V] AC, y las cargas de dimmers que puede variar la intensidad de luz de 0 a 100%, en la ventana también existe la opción de enviar a prender y apagar todas las luces a la vez, en la figura 8.3, se observa la ventana de iluminación.



**Figura 8.3:** Ventana de iluminación del sistema BMS.  
**Realizado por:** Gómez Oscar, 2018.

### 3.2.6. Ventana de seguridad

En esta ventana se puede monitorear a los sensores que están conectados al sistema BMS como son:

- Sensor de movimiento
- Sensor de luxes
- Sensor de temperatura
- Sensor de presencia de personas

Adicionalmente se tiene una cámara de seguridad anti vandálica tipo domo integrada al sistema BMS.

En la figura 9.3, podemos observar la ventana de seguridad del sistema BMS.



**Figura 9.3:** Ventana de seguridad del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

### 3.2.7. *Ventana de menú de escenas*

Esta ventana permite escoger la escena que de acuerdo al momento, situación y actividades que se vaya a realizar se pueda escoger un ambiente predeterminado, el sistema BMS consta de 4 escenas las cuales son:

- Escena de descanso o relax

- Escena de reunión
- Escena de lectura
- Escena de cena romántica

Las escenas del sistema están constituidas por diferentes niveles de iluminación y audio dependiendo de la escena escogida, en la figura 10.3, se observa el menú de escenas del sistema BMS.



**Figura 10.3:** Ventana de menú de escenas del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

### 3.2.8. Ventana de cargas inductivas

En esta ventana se monitorea y controla el estado de las cargas inductivas (motores), que están integradas al sistema como son: 1 bomba de agua, 1 bomba contra incendios, 1 generador y 1 extractor en la figura 11.3 se puede observar la ventana.

Desde la misma ventana se monitorea una cisterna de agua el porcentaje de llenado es de 0 a 100% al estar en 30% el sistema BMS activa una alarma en bajo cuando está en el 80% de llenado el sistema activa una alarma en alto y estos estados son guardados en la base de datos del sistema BMS, de la misma ventana se puede consultar el registro de las alarmas el cual informa la fecha

y hora en la cual se activaron las alarmas el ack (acknowledgement) de la alarma si fue aceptada o atendida la misma y el usuario que estaba dentro del sistema BMS.



**Figura 11.3:** Ventana de cargas inductivas del sistema BMS.

**Realizado por:** Gómez Oscar, 2018.

### 3.2.9. Ventana de históricos

En esta ventana se observa el estado de las variables de consumo eléctrico como son:

- Voltaje [V]
- Corriente [A]
- Factor de potencia
- Consumo energético [Kw/h]
- Potencia [W]

A la vez se observa una gráfica en tiempo real del consumo energético; desde esta ventana se puede consultar los macrodatos que están guardados en la base de datos para crear un archivo y poder imprimirlos, tiene la opción de guardar la gráfica del consumo energético para posteriormente poderle estudiarla, en la figura 12.3, se observa la ventana desarrollada.



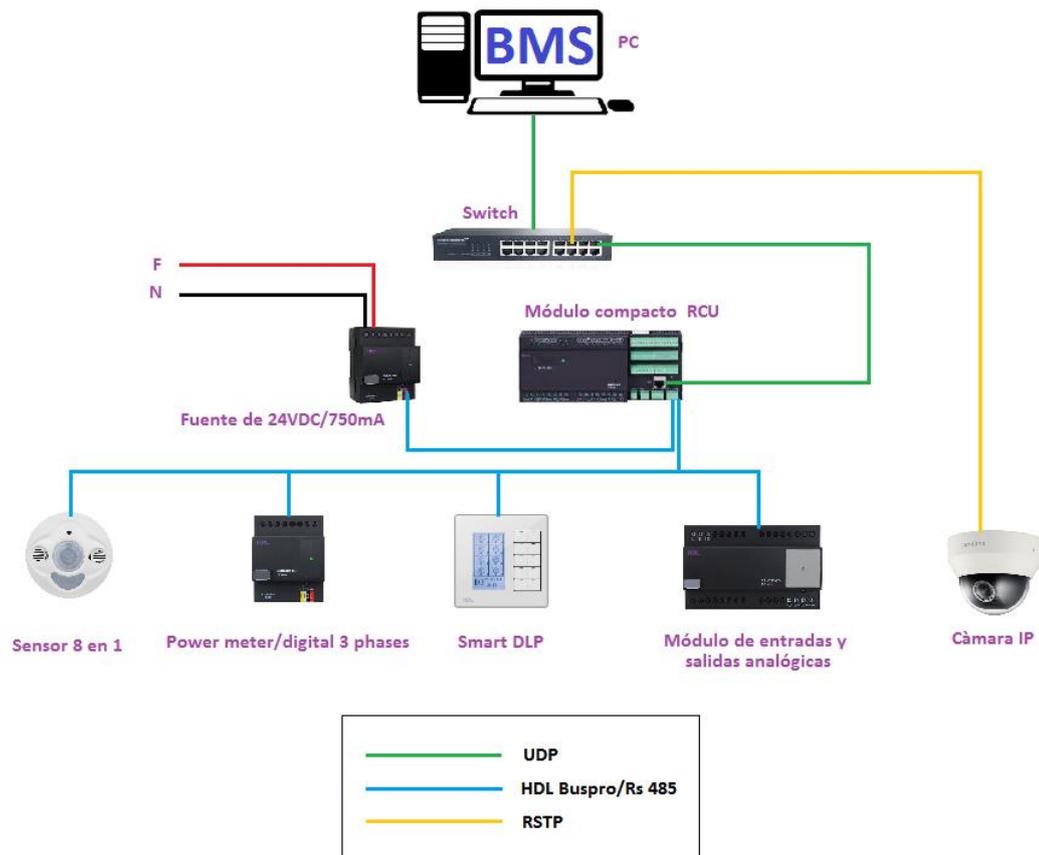
**Figura 12.3:** Ventana de históricos del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

### 3.3. Implementación de la comunicación UDP

Una vez terminado el diseño de la interfaz del sistema de control BMS se procede a realizar la comunicación entre las diferentes ventanas con los equipos de HDL Buspro, para ellos se utilizó el protocolo de comunicación UDP (User Datagram Protocol, protocolo de datagrama de usuario), es un protocolo que trabaja en el nivel 4 de la capa OSI es decir en el nivel de transporte este se basa en el intercambio de datagramas el envío de los datagramas es posible sin que se establezca una conexión.

#### 3.3.1. Conformación del bus de HDL Buspro

En la figura 13.3, se observa como está conformado la comunicación del sistema de control BMS con los componentes de HDL Buspro.



**Figura 13.2:** Conformación de la comunicación del sistema BMS.  
**Realizado por:** Gómez Oscar, 2018.

En el diagrama se observa que el sistema de control BMS diseñado está instalado en una Pc, la cual se comunica a un Switch por el protocolo de comunicación UDP el mismo que está conectado al módulo compacto HDL-RCU (Room Control Unit) con la mismo protocolo UDP; desde el modulo compacto se controla la comunicación entre la Pc y los otros equipos que están conectados al Bus de HDL, el modulo compacto cuenta con un puerto de comunicación Ethernet y un puerto Bus Pro.

Se conforma el Bus de HDL quien adopta el estándar industrial RS-485 quien utiliza el cable par trenzado para la comunicación el bus trabaja con un voltaje de 24 VDC, se puede utilizar el cable especial de HDL Buspro/KNX para realizar el bus o el cable UTP CAT5E para este caso se utilizó la segunda opción.

En el Bus se conectaron los siguientes equipos:

- Una fuente para alimentar al Bus HDL, input 110V/220V output 24V/750mA
- Un módulo compacto RCU
- Un sensor 12 en 1
- Un medidor de energía Power Meter

- Un módulo de entradas y salidas analógicas de 0-10V/4-20mA
- Un Smart DLP

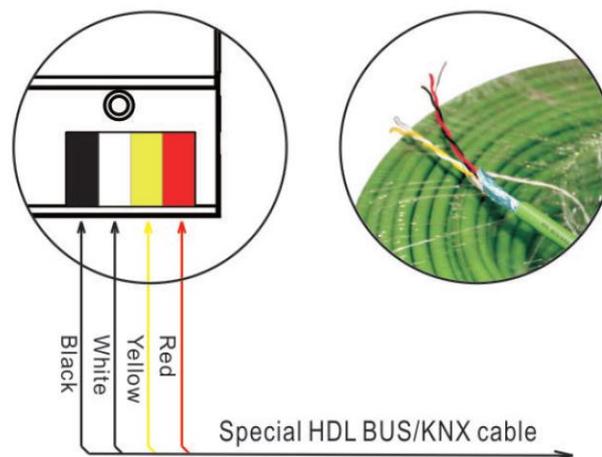
Para entender de una mejor manera como está constituido la conexión del cable del bus de HDL Buspro/KNX, en la tabla 2.3, se describe la conexión de los colores del cable.

**Tabla 2.3:** Definición de secuencia de conexión de terminal cable especial HDL.

| COM    | Common            | Black  |
|--------|-------------------|--------|
| DATA - | Signal -          | White  |
| DATA + | Signal +          | Yellow |
| DC 24V | 24 V power supply | Red    |

Fuente:(“Design Guidance Intelligent building control system, HDL BUS,” 2015)

En la figura 14.3, se observa como está conformado la conexión del cable de HDL Buspro/KNX



**Figura 14.3:** Conexión del cable HDL Buspro/KNX

Fuente:(“Design Guidance Intelligent building control system, HDL BUS,” 2015)

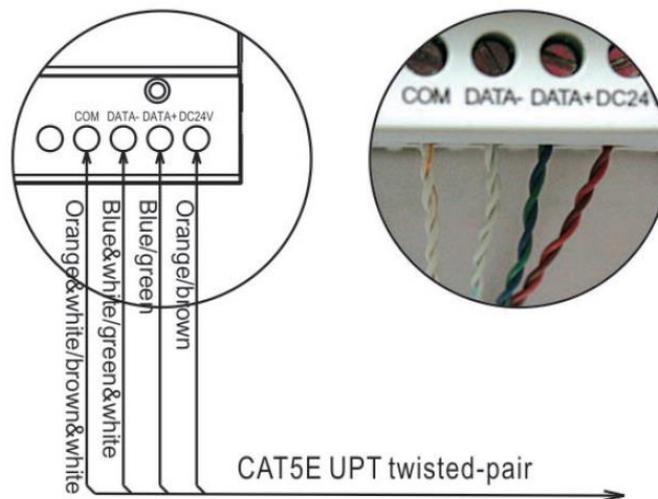
En la tabla 3.3 se describe como está constituido la conexiones del bus con el cable UPT CAT5E que se implementó para el sistema de control BMS.

**Tabla 3.1:** Definición de secuencia de conexión de terminal del cable UTP CAT5E.

| COM    | Common            | Orange & White/Brown&White |
|--------|-------------------|----------------------------|
| DATA - | Signal -          | Blue&White/Green&White     |
| DATA + | Signal +          | Blue/Green                 |
| DC 24V | 24 V power supply | Orange/Brown               |

Fuente: (“Design Guidance Intelligent building control system, HDL BUS,” 2015)

En la figura 15.3, se observa como está conformado la conexión del cable UPT CAT5E.



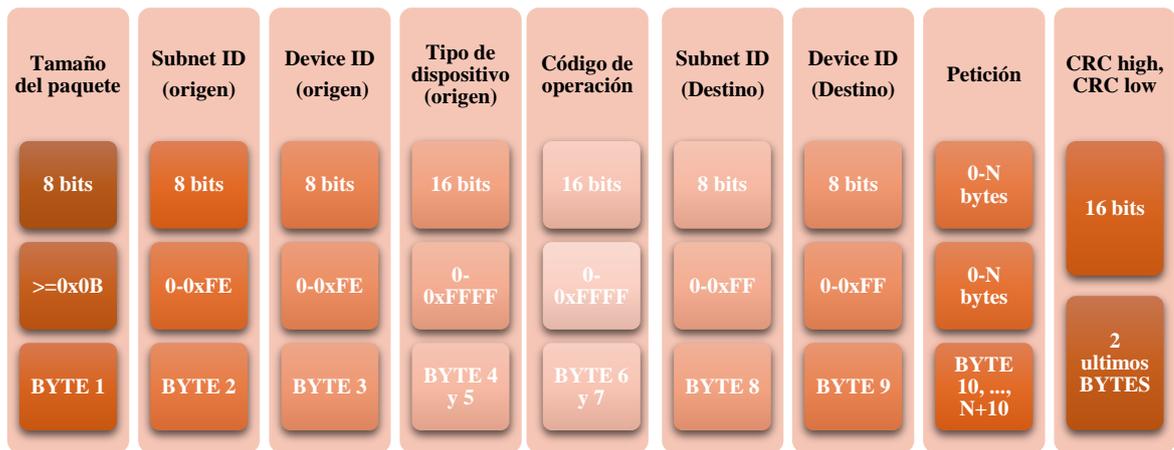
**Figura 15.3:** Conexión del cable UPT CAT5E.

Fuente: (“Design Guidance Intelligent building control system, HDL BUS,” 2015)

### 3.3.2. Generación de la trama UDP en NetBeans para el control de los equipos HDL

En la figura 16.3, se observa el modelo de la trama que utiliza el sistema Domótico HDL Bus Pro para su comunicación.

En la trama primero se especifica el tamaño del paquete que se requiere enviar, el tamaño mínimo del paquete que se envía es de 10 bytes seguido de la subnet ID y el device ID del equipo de HDL de origen, a continuación el tipo de equipo de origen, el código de operación, la subnet ID y device ID del destinatario, por último la petición y el código de redundancia cíclica.



**Figura 16.3:** Trama UDP del sistema HDL Bus Pro.

Elaborado por: Gómez Oscar, 2018.

Un ejemplo si se quisiera encender el canal 3 del módulo de relé a un valor del 100% que se encuentra en la subred 1 y cuya dirección de dispositivo es 3 utilizando la PC la trama quedaría especificada de acuerdo a la tabla 4.3.

**Tabla 4.3:** Trama UDP generada para encender un canal de relé número 3.

| Tamaño de la trama | Dispositivo de origen | Tipo de dispositivo de origen | Código de operación           | Dispositivo de destino | Petición              | Código de redundancia cíclica |
|--------------------|-----------------------|-------------------------------|-------------------------------|------------------------|-----------------------|-------------------------------|
| 0x0d               | 0x01 0x64             | 0xff 0xfe                     | 0x00 0x31                     | 0x01 0x03              | 0x03 0x64             | 0x5f 0x48                     |
| 13 bytes           | Subnet ID=1           | Dispositivo PC                | Single channel lighth control | Subnet ID=1            | Canal 3<br>Valor 100% |                               |

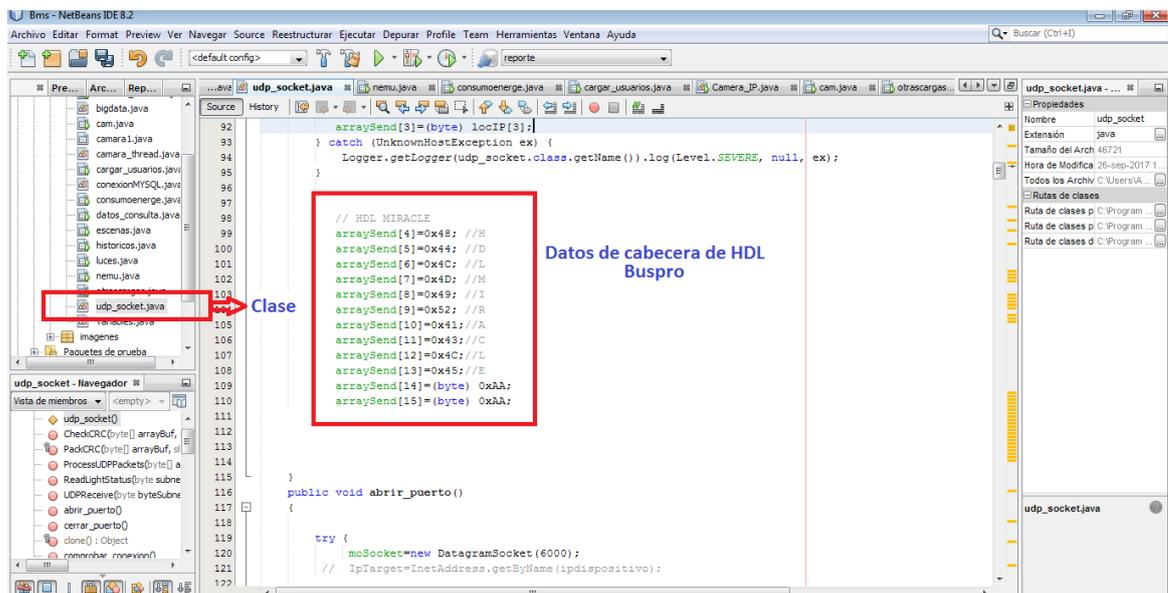
Elaborado por: Gómez Oscar, 2018.

En NetBeans se desarrolló una clase llamada `udp_socket` quien es la encargada de realizar la comunicación UDP con los equipos de HDL y la interacción con las demás ventanas del sistema BMS, en la clase `udp_socket` se envía las peticiones a los equipos y los estados de sus entradas y salidas para actualizar las ventanas del BMS, es la encargada se empaquetar los datos que corresponden a la acción que se tome de las ventanas del BMS cuando ya está empaquetados los datos procede a calcular el código de redundancia cíclica y los empaqueta con los datos de cabecera en la figura 17.3, se observa como está conformado la trama UDP que se envía así el modulo compacto para que este lo convierta en la trama de HDL Buspro y los distintos equipos que están conectados al bus puedan funcionar dependiendo de las acciones que tome el usuario que está interviniendo en el BMS.



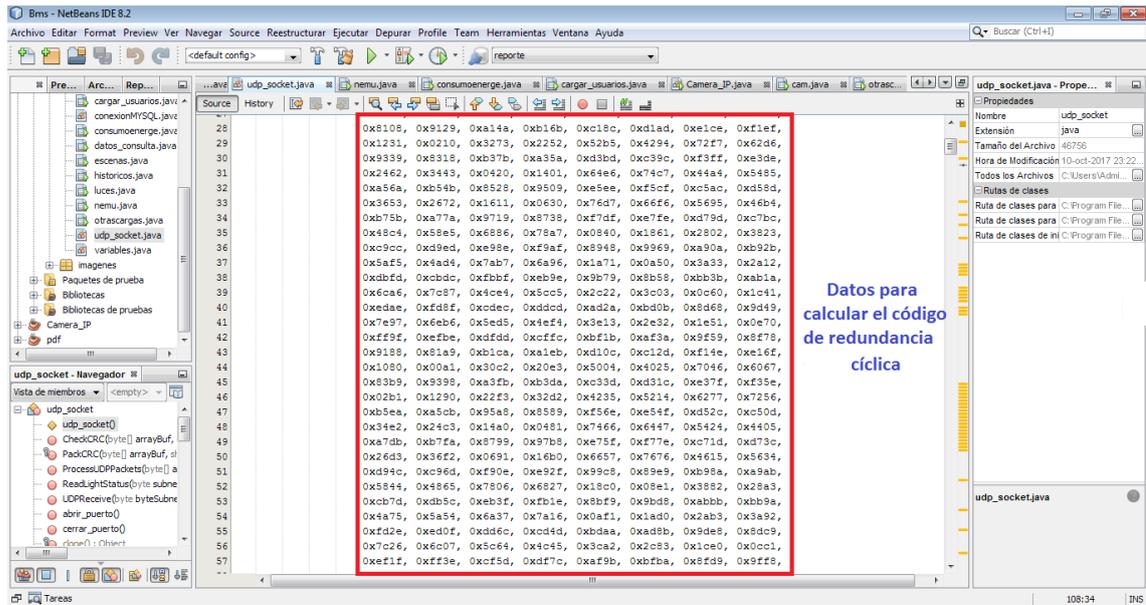
**Figura 17.3:** Diagrama de bloques de la trama UDP para el control de los equipos HDL.  
 Elaborado por: Gómez Oscar, 2018.

Dentro de esta clase `udp_socket` están incorporados los datos de la cabecera que proporciona la fábrica de HDL Buspro, estos datos sirven para todos los equipos que tiene HDL sin estos datos no sería posible la comunicación UDP en la figura 18.3, se observa creada la clase `udp_socket` en NetBeans y la cabecera de HDL.



**Figura 18.3:** Clase `udp_socket` y datos de cabecera de HDL.  
 Elaborado por: Gómez Oscar, 2018.

En la figura 19.3, se observa los datos para calcular el código de redundancia cíclica estos datos proporciona la empresa de HDL sirven para la comunicación de todos sus equipos, los datos están incorporados dentro de la clase `udp_socket`.



**Figura 19.3:** Datos para el cálculo de la redundancia cíclica.  
**Realizado por:** Gómez Oscar, 2018.

Para una mejor visualización de los datos para el cálculo de la redundancia cíclica se observa en la tabla 5.3.

**Tabla 5.3:** Datos para el cálculo de la redundancia cíclica.

|        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x0000 | 0x1021 | 0x2042 | 0x3063 | 0x4084 | 0x50a5 | 0x60c6 | 0x70e7 |
| 0x8108 | 0x9129 | 0xa14a | 0xb16b | 0xc18c | 0xd1ad | 0xe1ce | 0xf1ef |
| 0x1231 | 0x0210 | 0x3273 | 0x2252 | 0x52b5 | 0x4294 | 0x72f7 | 0x62d6 |
| 0x9339 | 0x8318 | 0xb37b | 0xa35a | 0xd3bd | 0xc39c | 0xf3ff | 0xe3de |
| 0x2462 | 0x3443 | 0x0420 | 0x1401 | 0x64e6 | 0x74c7 | 0x44a4 | 0x5485 |
| 0xa56a | 0xb54b | 0x8528 | 0x9509 | 0xe5ee | 0xf5cf | 0xc5ac | 0xd58d |
| 0x3653 | 0x2672 | 0x1611 | 0x0630 | 0x76d7 | 0x66f6 | 0x5695 | 0x46b4 |
| 0xb75b | 0xa77a | 0x9719 | 0x8738 | 0xf7df | 0xe7fe | 0xd79d | 0xc7bc |
| 0x48c4 | 0x58e5 | 0x6886 | 0x78a7 | 0x0840 | 0x1861 | 0x2802 | 0x3823 |
| 0xc9cc | 0xd9ed | 0xe98e | 0xf9af | 0x8948 | 0x9969 | 0xa90a | 0xb92b |
| 0x5af5 | 0x4ad4 | 0x7ab7 | 0x6a96 | 0x1a71 | 0x0a50 | 0x3a33 | 0x2a12 |
| 0xdbfd | 0xcdbc | 0xfbbf | 0xeb9e | 0x9b79 | 0x8b58 | 0xbb3b | 0xab1a |
| 0x6ca6 | 0x7c87 | 0x4ce4 | 0x5cc5 | 0x2c22 | 0x3c03 | 0x0c60 | 0x1c41 |
| 0xedae | 0xfd8f | 0xcdec | 0xddcd | 0xad2a | 0xbd0b | 0x8d68 | 0x9d49 |
| 0x7e97 | 0x6eb6 | 0x5ed5 | 0x4ef4 | 0x3e13 | 0x2e32 | 0x1e51 | 0x0e70 |
| 0xff9f | 0xefbe | 0xdfdd | 0xcffc | 0xbf1b | 0xaf3a | 0x9f59 | 0x8f78 |
| 0x9188 | 0x81a9 | 0xb1ca | 0xa1eb | 0xd10c | 0xc12d | 0xf14e | 0xe16f |
| 0x1080 | 0x00a1 | 0x30c2 | 0x20e3 | 0x5004 | 0x4025 | 0x7046 | 0x6067 |
| 0x83b9 | 0x9398 | 0xa3fb | 0xb3da | 0xc33d | 0xd31c | 0xe37f | 0xf35e |
| 0x02b1 | 0x1290 | 0x22f3 | 0x32d2 | 0x4235 | 0x5214 | 0x6277 | 0x7256 |
| 0xb5ea | 0xa5cb | 0x95a8 | 0x8589 | 0xf56e | 0xe54f | 0xd52c | 0xc50d |
| 0x34e2 | 0x24c3 | 0x14a0 | 0x0481 | 0x7466 | 0x6447 | 0x5424 | 0x4405 |
| 0xa7db | 0xb7fa | 0x8799 | 0x97b8 | 0xe75f | 0xf77e | 0xc71d | 0xd73c |
| 0x26d3 | 0x36f2 | 0x0691 | 0x16b0 | 0x6657 | 0x7676 | 0x4615 | 0x5634 |
| 0xd94c | 0xc96d | 0xf90e | 0xe92f | 0x99c8 | 0x89e9 | 0xb98a | 0xa9ab |
| 0x5844 | 0x4865 | 0x7806 | 0x6827 | 0x18c0 | 0x08e1 | 0x3882 | 0x28a3 |
| 0xcb7d | 0xdb5c | 0xeb3f | 0xfb1e | 0x8bf9 | 0x9bd8 | 0xabbb | 0xbb9a |
| 0x4a75 | 0x5a54 | 0x6a37 | 0x7a16 | 0x0af1 | 0x1ad0 | 0x2ab3 | 0x3a92 |
| 0xfd2e | 0xed0f | 0xdd6c | 0xcd4d | 0xbdaa | 0xad8b | 0x9de8 | 0x8dc9 |
| 0x7c26 | 0x6c07 | 0x5c64 | 0x4c45 | 0x3ca2 | 0x2c83 | 0x1ce0 | 0x0cc1 |
| 0xef1f | 0xff3e | 0xcf5d | 0xdf7c | 0xaf9b | 0xbfba | 0x8fd9 | 0x9ff8 |
| 0x6e17 | 0x7e36 | 0x4e55 | 0x5e74 | 0x2e93 | 0x3eb2 | 0x0ed1 | 0x1ef0 |

Realizado por: Gómez Oscar, 2018.

### 3.3.3. Generación de Macrodatos (Big Data) del sistema de control BMS

El sistema de control BMS cada milisegundo está generando datos los mismos que son guardados en una basa de datos que se realizó con MySQL y el entorno de trabajo Workbench.

En Workbench se realizó 2 esquemas (Schemas) de base de datos uno llamado bms y el otro macrodatos; el esquema bms contiene las tablas de las alarmas y de login.

En la tabla de alarmas se almacena los siguientes datos:

- Estado
- Fecha
- Hora
- ack
- Usuario

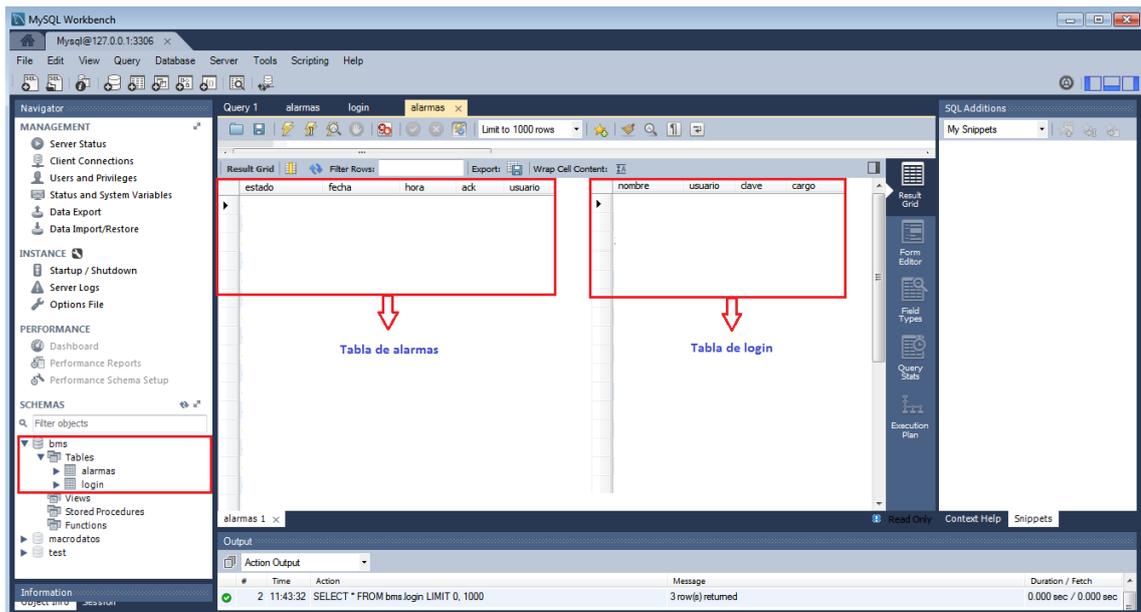
En la columna de estado se guardará el estado de alarma de la cisterna de agua en este caso alarma en LOW y alarma en HIGH, en la columna de fecha guarda la fecha en que se produjo las alarmas, en la columna de hora guarda la hora minutos y segundos en que se produjo las alarmas, en la columna de ack guarda el estado de la arma si fue atendida o no, en la columna de usuario guarda el usuario quien está gestionando al sistema BMS.

En la tabla de login se almacenan los siguientes datos:

- Nombre
- Usuario
- Clave
- Cargo

En la columna de nombre guarda los nombres y apellidos de los usuarios nuevos, en la columna de usuario guarda las siglas como el usuario se va a logiar, en la columna de clave guarda la clave de cada usuario, en la columna de cargo guarda el cargo de cada usuario en este caso del sistema BMS tenemos 4 cargos, gerente, administrador, supervisor y técnico.

En la figura 20.3, se observa el entorno de trabajo Workbench generado el esquema bms con las 2 tablas de alarmas y login.



**Figura 20.3:** Esquema bms con las dos tablas de alarmas y login.  
**Realizado por:** Gómez Oscar, 2018.

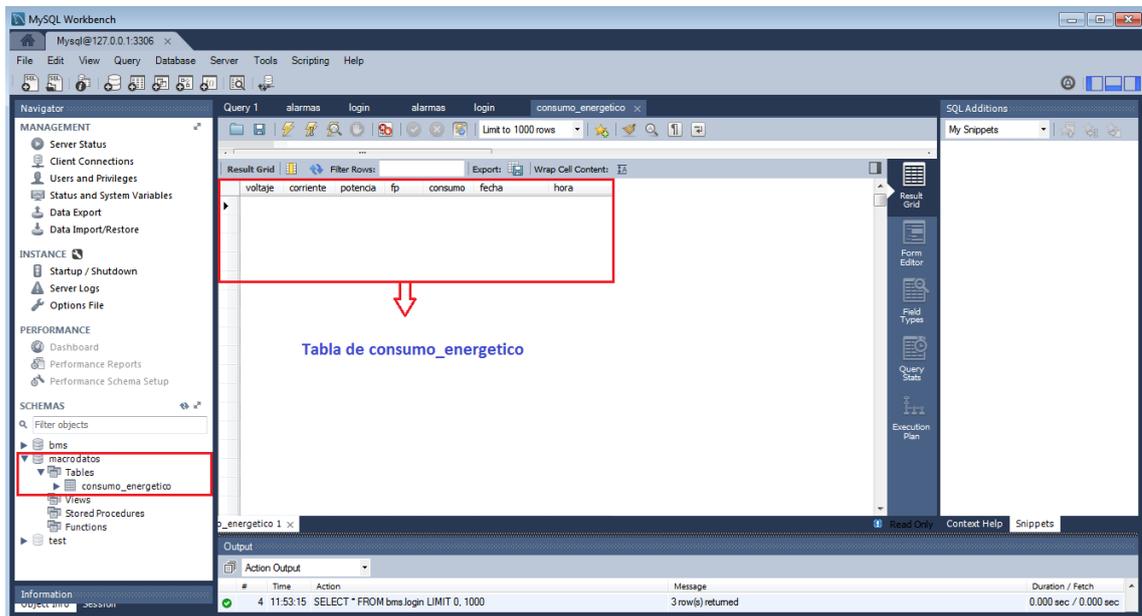
El esquema llamado macrodatos contiene una tabla llamada consumo\_energetico en la cual se almacena los siguientes datos:

- Voltaje
- Corriente
- Potencia
- Fp (factor de potencia)
- Consumo
- Fecha
- Hora

En la columna de voltaje guarda el voltaje de línea que está conectado los equipos del sistema BMS, en la columna de corriente guarda el consumo de corriente de todo el sistema, en la columna de potencia guarda la potencia total del sistema que consume, en la columna del factor de potencia se guarda el fp que generas todas las cargas conectado al BMS, en la columna de consumo guarda el consumo energético del sistema BMS, en la columna de fecha guarda todas las fechas que el sistema está trabajando y en la columna de hora guarda la hora minutos y segundos que el sistema BMS esta trabajo.

Con todos estos macrodatos guardados en la base datos del sistema de control BMS se puede validar y estudiarlos posteriormente para analizar el consumo energético en una edificación.

En la figura 21.3, se observa el entorno de trabajo Workbench generado el esquema macrodatos con la tabla de consumo\_energetico.



**Figura 21.3:** Esquema macrodatos con la tabla de consumo\_energetico.  
Realizado por: Gómez Oscar, 2018.

## CAPÍTULO IV

### 4. Pruebas y resultados

En este capítulo se realizó el análisis del comportamiento práctico del sistema de control BMS, con sus diferentes componentes que lo integran y el análisis de los macrodatos que genera el sistema.

Para las pruebas se realizó la construcción de un módulo donde se encuentran colocados los equipos de HDL Buspro el modulo cuenta con los siguientes equipos:

- Un módulo HDL-RCU (Room Control Unit), este módulo cuenta con un puerto de comunicación Ethernet para la comunicación con la pc y un puerto Bus pro, para la comunicación con los equipos Domóticos, además posee 17 salidas de relé y 4 salidas dimmerizables.
- Un módulo sensor 12 en 1, cuenta con un sensor de movimiento PIR, un sensor de temperatura, un sensor de presencia, un receptor de señales infrarrojas, un emisor de señales infrarrojas y 2 salidas de relé.
- Un módulo Smart DLP, que es la interfaz de usuario, cuenta con 4 ventanas para control de iluminación, una ventana para control de audio y una ventana para control de climatización.
- Una fuente de poder, es la encargada de alimentar al bus y a los equipos trabaja a 110 V/220V AC, 50 a 60 Hz, la salida es de 24 VDC-750mA.
- Un medidor de energía de 3 fases, este equipo registra los parámetros eléctricos y permite guardar el consumo energético, corriente, voltaje, potencia y factor de potencia.
- Un módulo analógico, este módulo contiene 6 canales que admite diferentes tipos de señal ya sea de entrada y de la salida las cuales son de 0-10 VDC y 4-20 mA, tiene 2 salidas las cuales se puede configurar de 0-10VAC y de 4-20 mA.
- Una cámara IP de marca hikvision.

En el Anexo A se encuentra el Datasheet de cada equipo.

En la figura 1.4 se observar el módulo implementado.



**Figura 1.4:** Módulo implementado para las pruebas del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

#### 4.1. Crear un nuevo usuario desde el sistema BMS y guardar en la base de datos del sistema

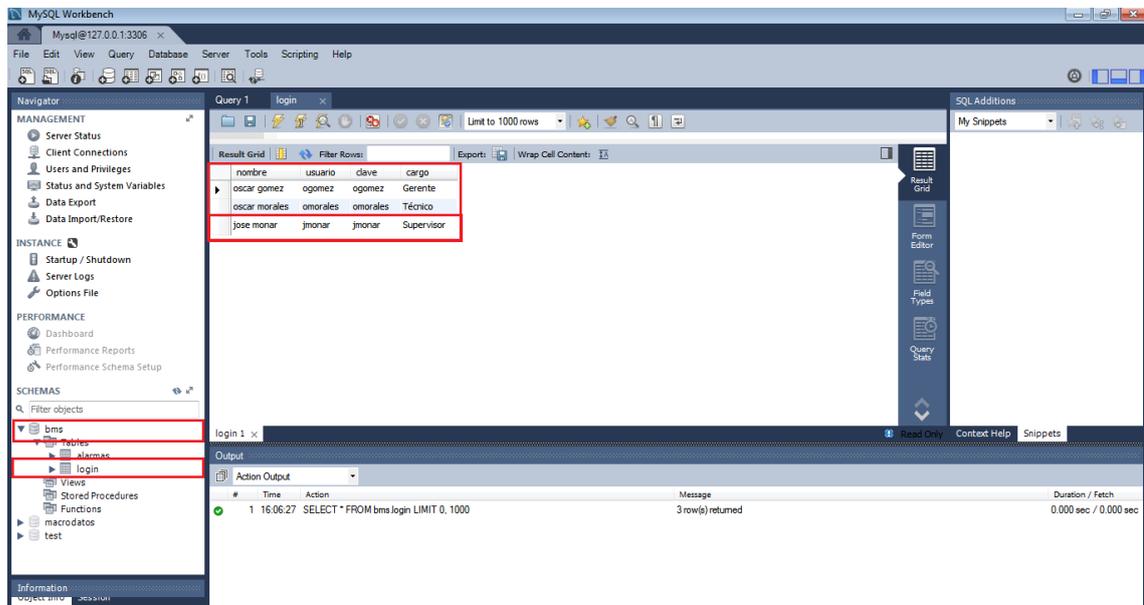
El sistema BMS tiene la opción para crear nuevos usuarios con distintas jerarquías, se realizó la prueba creando un nuevo usuario desde el sistema BMS en la figura 2.4, se observa la operación realizada, esta operación solo lo puede hacer el que tiene mayor jerarquía en este caso el gerente.



**Figura 2.4:** Generación de un nuevo usuario desde el sistema BMS.  
Realizado por: Gómez Oscar, 2018.

Se ingresa los datos del nuevo usuario en la ventana correspondiente en el sistema BMS, guardamos los datos y el sistema envía un mensaje si el guardado fue exitoso o no.

El sistema BMS se debe comunicar con la base de datos en donde guarda la información del nuevo usuario en las tablas que ya están creadas, en la figura 3.4, se comprueba que los datos se guardaron con éxito en la base de datos.



**Figura 3.4:** Datos guardados de los usuarios generados desde el sistema BMS.

**Realizado por:** Gómez Oscar, 2018.

Cuando se comprueba que el sistema está guardando de manera correcta los datos de los usuarios en las columnas que se asignaron para cada información en la base de datos se procede a ejecutar el BMS, y probar con el nuevo usuario creado para comprobar que el sistema BMS está leyendo los datos de la base de datos correctamente en la figura 4.4 se observa la prueba del nuevo usuario para ingresar al sistema BMS.



**Figura 4.1:** Lectura desde el BMS hacia la base de datos del nuevo usuario.  
Realizado por: Gómez Oscar, 2018.

La lectura desde el sistema BMS hacia la base de datos es correcta el usuario y el password, en la ventana para ingresar al BMS el sistema muestra automáticamente el tipo de usuario que es y muestra el botón para ingresar al sistema de control BMS, en la figura 5.4, se muestra cuando los datos del usuario son incorrectos.



**Figura 5.4:** Datos incorrectos al ingresar al sistema BMS.  
Realizado por: Gómez Oscar, 2018.

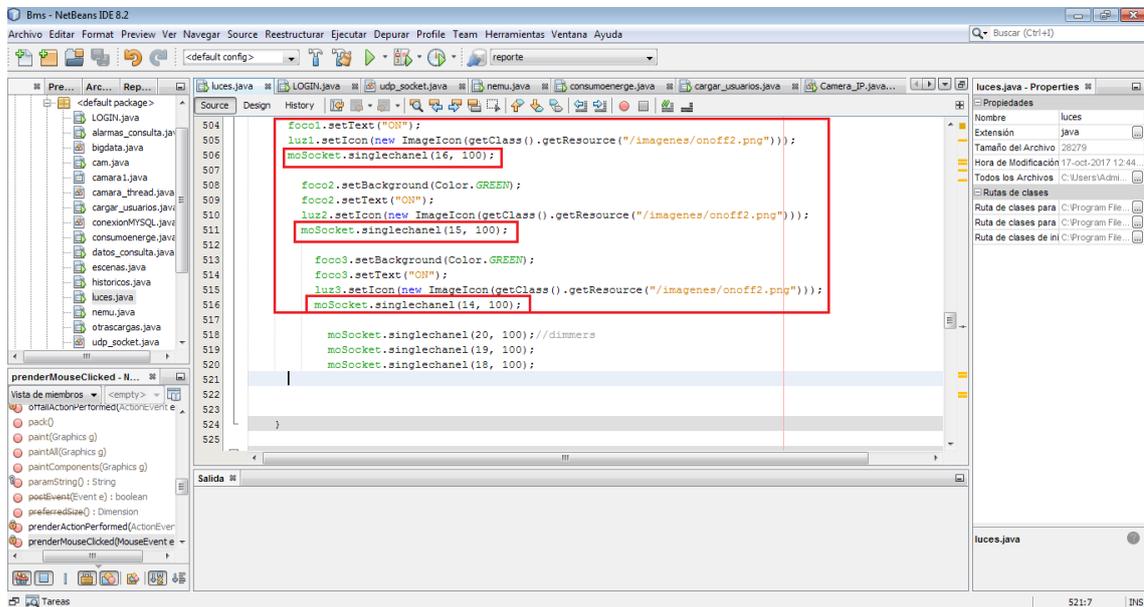
Cuando se ingresan datos incorrectos el sistema BMS lee la base de datos y al no encontrar esos datos el sistema envía un mensaje de usuario incorrecto.

## 4.2. Pruebas de envío de peticiones desde el sistema BMS hacia los equipos de HDL

Al tener todo diseñado las ventanas del BMS la comunicación UDP, se procede a enviar peticiones desde el sistema hacia los equipos de HDL para probar la comunicación UDP y las distintas salidas de accionamientos que se tiene en el sistema.

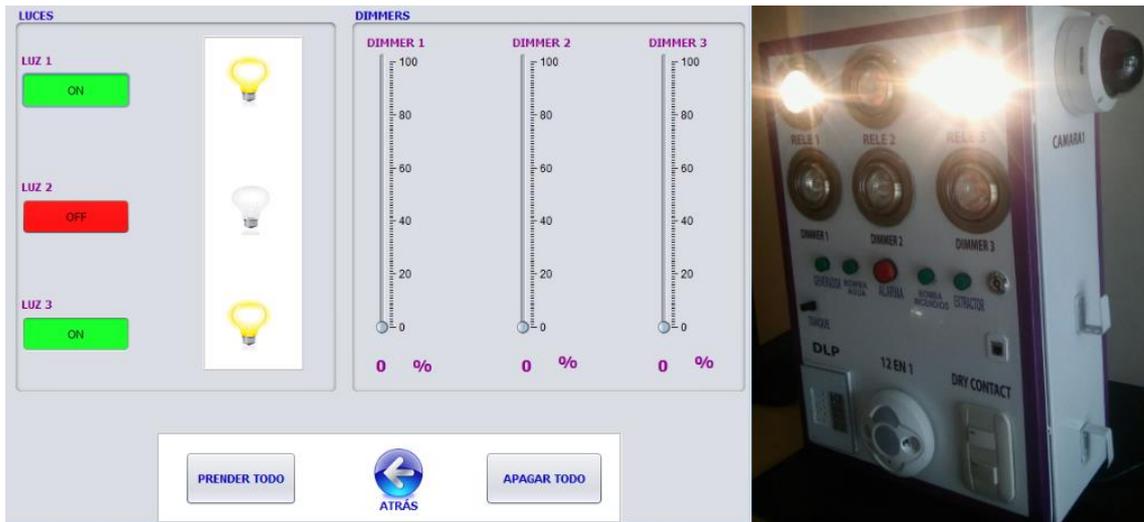
### 4.2.1. Solicitud para encender las salidas de relés

Las salidas de relé se probó individualmente canal por canal posteriormente se acciono todas las salidas a la vez (encendido y apagado), para enviar a encender desde el BMS una salida de relé se escoge el número de canal al cual se va encender y el porcentaje de encendido en este caso las salidas de relé tienen dos porcentajes de 0 y 100%, en la figura 6.4, se observa la programación para encender los canales de relés, la función moSocket es la encargada de enviar el número de canal y el porcentaje a encender cuando se acciona el botón de encendido este se coloca en un estado de ON y de color verde caso contrario se coloca de color rojo y en estado de OFF.



**Figura 6.4:** Instrucciones para accionar las salidas de relé del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

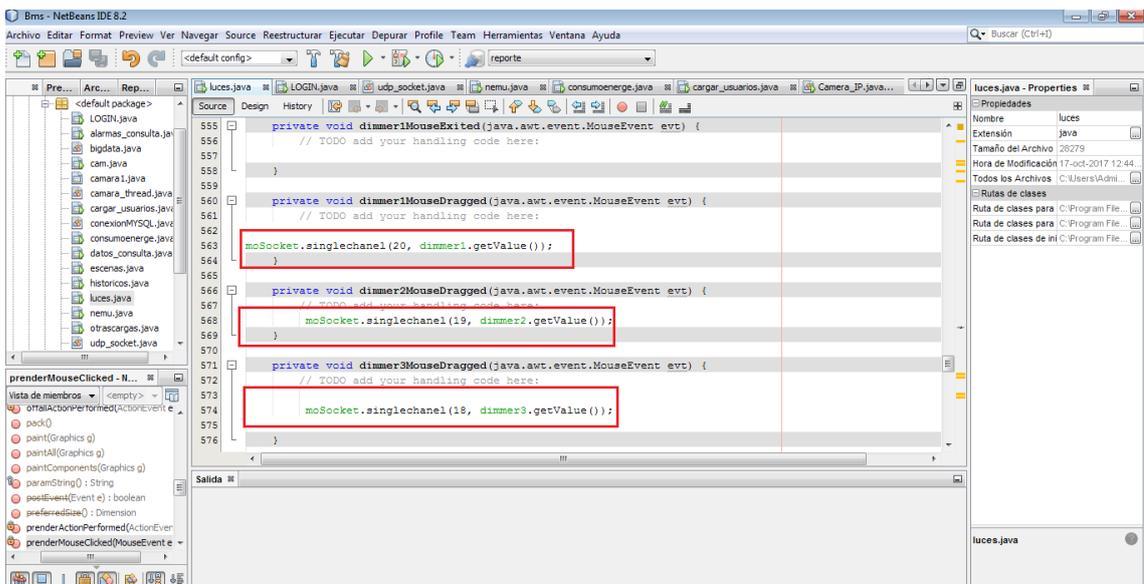
En la figura 7.4, se observa la petición enviada desde el sistema BMS para encender dos salidas de relé.



**Figura 7.4:** Acción enviada desde el Sistema BMS hacia las salidas de relé.  
Realizado por: Gómez Oscar, 2018.

#### 4.2.2. *Petición para encender las salidas de los dimmers*

Las salidas de los dimmers se probaron cada uno proporcionalmente de 0 a 100% para visualizar la proporción de luz en el módulo, en la figura 8.4, se observa el envío de los datos a través de la función `moSocket` de cada uno de los dimmers. Estos valores son guardados en una variable `getValue` para ser visualizados en la ventana de iluminación del BMS la cantidad de porcentaje que tiene cada uno de los dimmers.



**Figura 8.4:** Instrucciones para enviar el porcentaje de los dimmers hacia los equipos de HDL.  
Realizado por: Gómez Oscar, 2018.

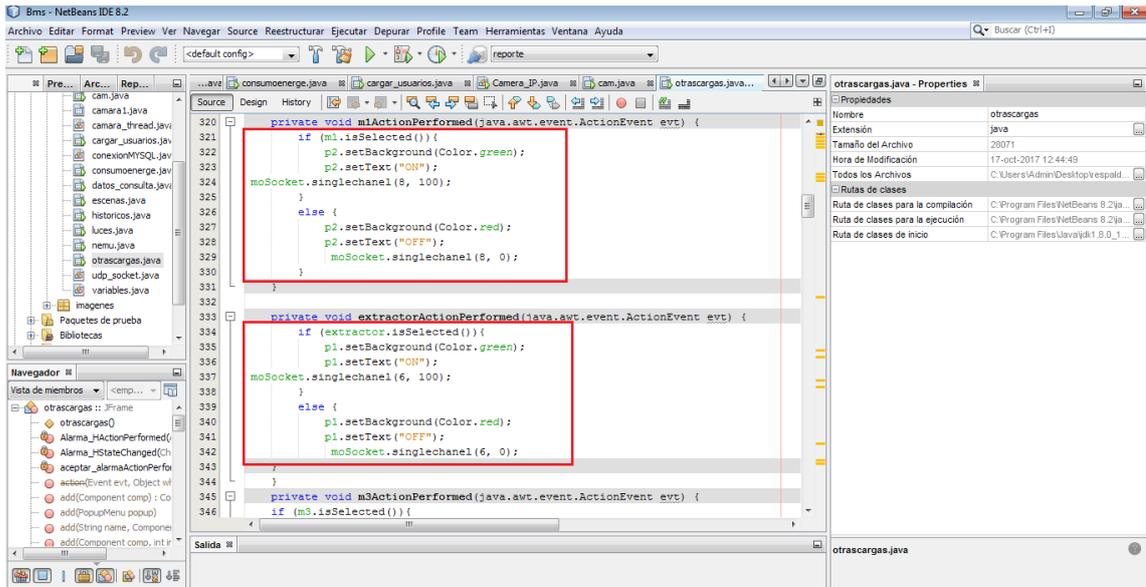
En la figura 9.4, se observa la petición enviada desde el sistema BMS para encender los dimmers proporcionalmente; el primer dimmer está al 2% el segundo al 3% y el tercero a un 10% de iluminación.



**Figura 9.4:** Acción enviada desde el Sistema BMS hacia los dimmers.  
Realizado por: Gómez Oscar, 2018.

#### 4.2.3. *Petición para encender las cargas inductivas y leer las entradas analógicas*

Las salidas de las cargas inductivas se probaron individualmente canal por canal y todas a la vez para constatar si no existen conflictos al enviar todas las salidas a encender, caso que no ocurrió. En la figura 10.4, se observa el código del BMS como se envía la petición a los equipos, se direcciona el número de canal que se va accionar, la función moSocket es la encargada de enviar el número de canal y el porcentaje a encender cuando se acciona el botón de encendido este se coloca en un estado de ON y de color verde caso contrario se coloca de color rojo y en estado de OFF.



**Figura 10.4:** Instrucciones para enviar a accionar las cargas inductivas del sistema BMS.  
**Realizado por:** Gómez Oscar, 2018.

En la figura 11.4, se observa la petición enviada desde el sistema BMS para encender las cargas inductivas del módulo, al mismo tiempo se observa la lectura de la señal analógica, del módulo hacia el BMS.



**Figura 11.4:** Petición desde el BMS para encender las cargas inductivas/lectura de la entrada analógica.  
**Realizado por:** Gómez Oscar, 2018.

#### 4.2.4. Lectura de datos de los equipos de HDL hacia el sistema de control BMS

El sistema BMS cada milisegundo lee continuamente el estados de los sensores de los equipos del módulo al mismo tiempo se puede manipular el DLP para accionar las cargas de relé, los

dimers, las cargas inductivas y las entradas analógicas. Los indicadores del sistema BMS se actualizan el estado de las cargas y los sensores.

Para entender cómo se realizó la lectura de los datos de los equipos hacia el sistema BMS se va a explicar la adquisición de una variable de un equipo de HDL.

La empresa HLD Buspro proporciona un documento confidencial el cual fue entregado para el desarrollo de este proyecto de investigación por el representante de la marca de HDL en el Ecuador, este documento contiene los códigos de operación del bus HDL y los tipos de comandos para la transmisión y recepción en el bus.

Se realizara la lectura del voltaje de línea del canal 2 del equipo Power Meter de HDL, en la tabla 1.4, se observa el código de operación para leer el voltaje.

**Tabla 1.4:** Código de operación para leer el voltaje de Power Meter.

| Command Code: 0 x D902             |                           |             |
|------------------------------------|---------------------------|-------------|
| Communication Mode: Point to Point |                           |             |
| Scope of SubNet ID: 0-254          | Scope of Device ID: 0-254 |             |
| Additional Contents                |                           |             |
| Size of Additional Contents: 0     |                           |             |
| Index                              | Remark                    | Value Scope |

Fuente:(Chapter et al., 2014)

El código de operación es el 0 x D902 para leer el voltaje la comunicación es de punto a punto se escoge la sub red y el ID del equipo en este caso el Power meter está en la sub red 1 y el ID es el 5, estos datos se programa en NetBeans en la clase udp\_socket en la figura 12.4, se observa el código de la función para la lectura del voltaje.

```

608
609
610 public String leer_voltaje(byte subnet, byte device, int canal)
611 {
612     String voltaje=" ";
613     byte[] voltajes = new byte[1024];
614     arrayPackCRC[0]=(byte)0x0B; // tamaño del paquete a enviar
615     arrayPackCRC[1]=(byte)0x0C; // original subnet ID
616     arrayPackCRC[2]=(byte)0xFB; //original device ID
617     arrayPackCRC[3]=(byte)0xFF; //original device type PC higher then 8
618     arrayPackCRC[4]=(byte)0xFE; //original device type PC lower then 8
619     arrayPackCRC[5]=(byte)0xd9; //codigo de operacion higher then 8 | single chanel control
620     arrayPackCRC[6]=(byte)0x02; //codigo de operacion lower then 8 | single chanel control
621     arrayPackCRC[7]=(byte)0x01; //subnet ID of target device
622     arrayPackCRC[8]=(byte)0x05; //device ID of target device
623
624     PackCRC(arrayPackCRC, (short) (arrayPackCRC.length-6));
625     for(short i=0; i<=arrayPackCRC.length-7;i++)
626     {
627         arraySend[i+16]=arrayPackCRC[i];
628     }
629     arraySend[25]=H_CRC;
630     arraySend[26]=L_CRC;
631     // System.out.println("Petición");
632
633     tamaño_paquete=(short) arraySend.length;
634     DatagramPacket enviar;
635     try {
636         enviar = new DatagramPacket(arraySend,31,InetAddress.getByNome(ipdispositivo), 6000);
637     }
638 }

```

**Figura 2.4:** Función para leer el voltaje del equipo Power Meter.  
**Realizado por:** Gómez Oscar, 2018.

Una vez enviada la petición para leer el voltaje el equipo envía la respuesta en la tabla 2.4 se muestra la configuración de como envía los datos el equipos.

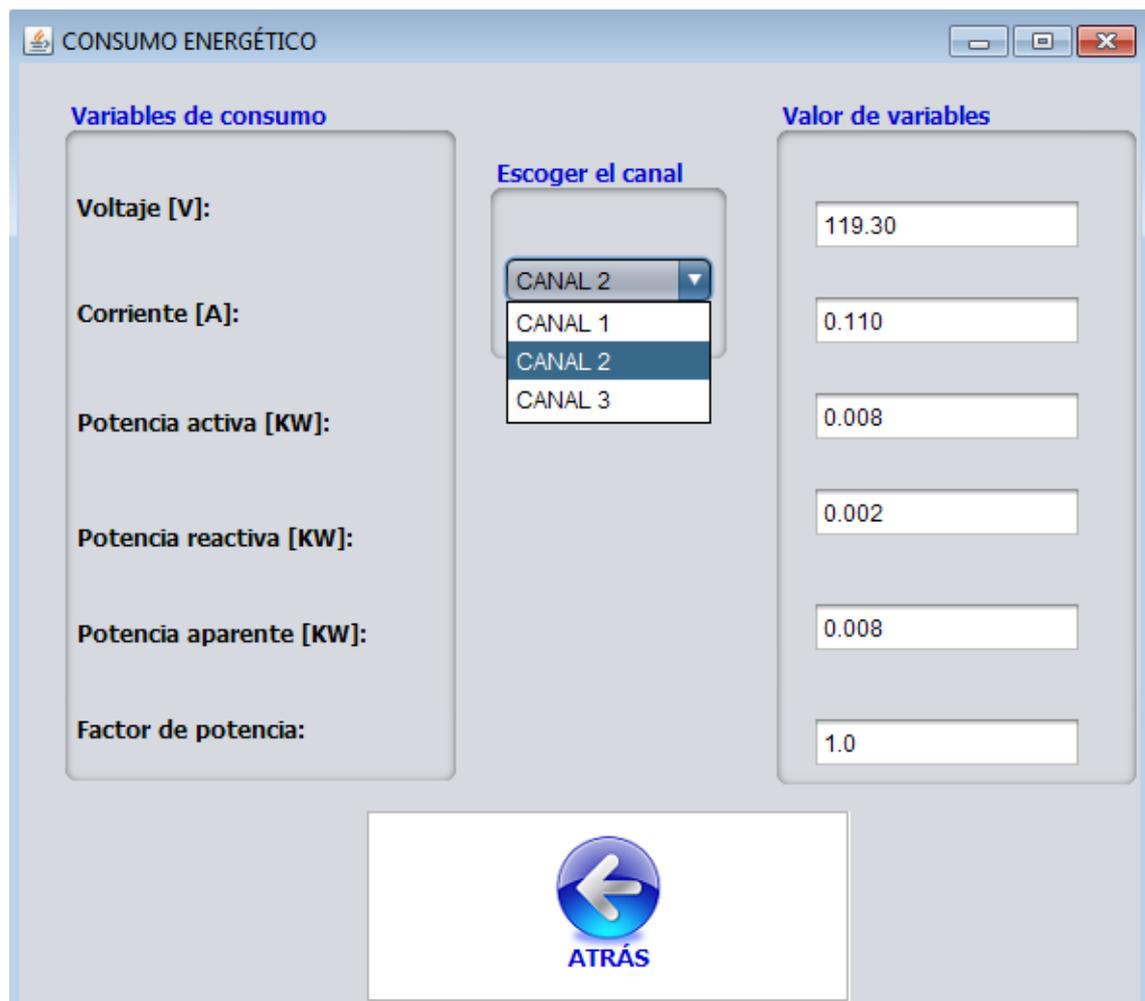
**Tabla 2.4:** Respuesta de módulo Power Meter para leer el voltaje.

| Command Code: 0 x D903             |               |                           |
|------------------------------------|---------------|---------------------------|
| Communication Mode: Point to Point |               |                           |
| Scope of SubNet ID: 0-254          |               | Scope of Device ID: 0-254 |
| Additional Contents                |               |                           |
| Size of Additional Contents: 12    |               |                           |
| Index                              | Remark        | Value Scope               |
| 1                                  | Chn 1 voltage | Value / 256 integer part  |
| 2                                  |               | Value % 256 integer part  |
| 3                                  |               | Decimal part (0-9)        |
| 4                                  |               | Decimal part (0-9)        |
| 5                                  | Chn 2 voltage | Value / 256 integer part  |
| 6                                  |               | Value % 256 integer part  |
| 7                                  |               | Decimal part (0-9)        |
| 8                                  |               | Decimal part (0-9)        |
| 9                                  | Chn 3 voltage | Value / 256 integer part  |
| 10                                 |               | Value % 256 integer part  |
| 11                                 |               | Decimal part (0-9)        |
| 12                                 |               | Decimal part (0-9)        |

**Fuente:**(Chapter et al., 2014)

El equipo envía un vector de tamaño de 12 bits, el sistema BMS está conectado al canal 2 del equipo en donde entra la línea de alimentación del sistema por lo tanto los bits que se utilizan son el 5, 6, 7 y 8; los bits 5 y 6 entregan el valor entero de la medición los bits 7 y 8 es la parte decimal del voltaje.

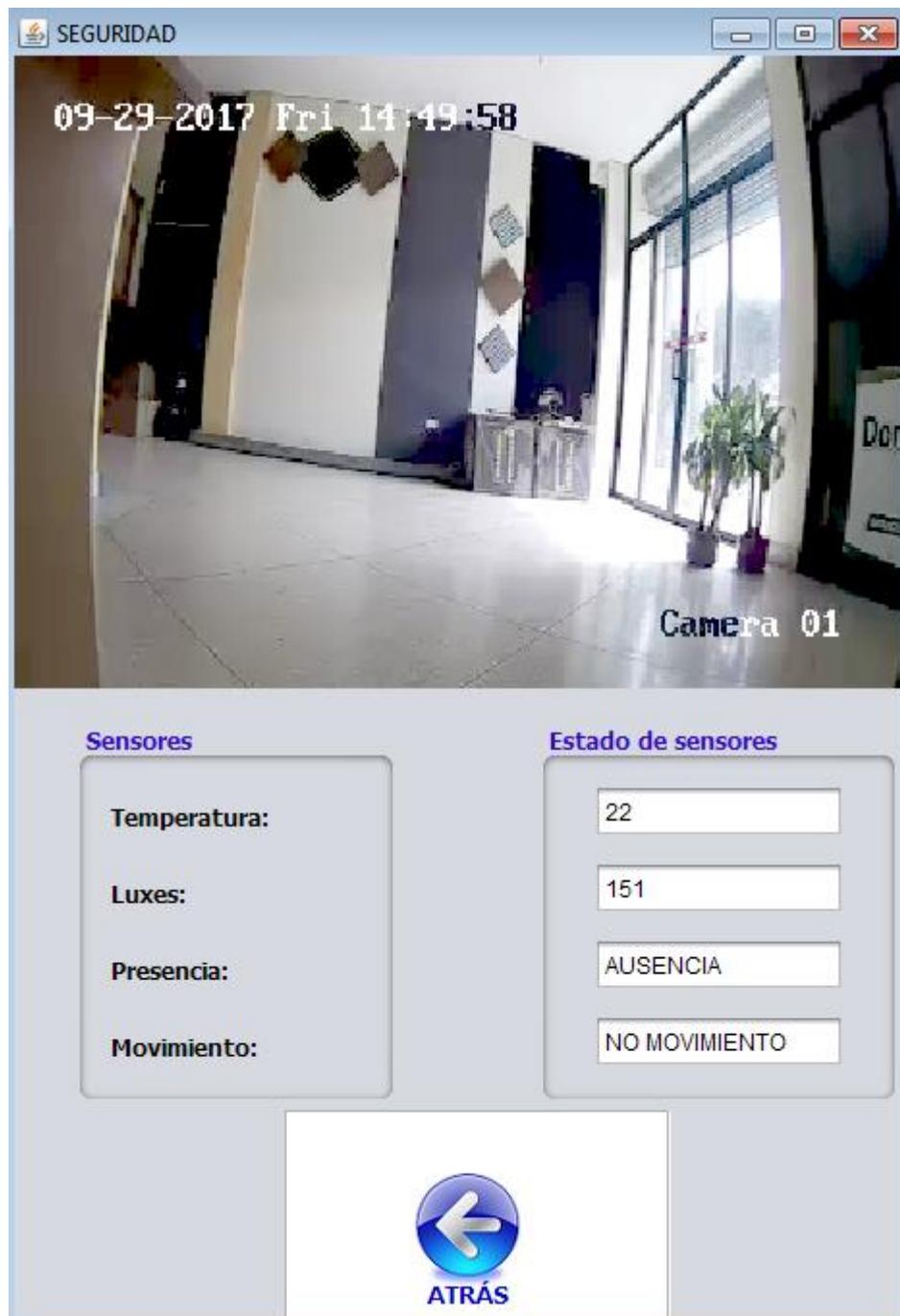
Si se quisiera medir en el canal 1 o 3 el principio es el mismo ya explicado anteriormente en la figura 13.4 se observa al sistema BMS midiendo el voltaje de línea que se encuentra conectado en el canal 2.



**Figura 13.4:** Ventana de consumo energético, medición de parámetros de energía del canal 2.  
**Realizado por:** Gómez Oscar, 2018.

En la figura 13.4 se observa que el sistema BMS está midiendo otros parámetros eléctricos para la adquisición de las otras variables eléctricas el principio es el mismo como ya se detalló anteriormente para el voltaje.

Gracias a los códigos de operación del bus HDL fue posible la lectura de los diferentes sensores que tiene el sistema BMS implementado, en la figura 14.4 se observa la ventana de seguridad que se la probó con la cámara y los sensores de temperatura, luxes, presencia y movimiento.

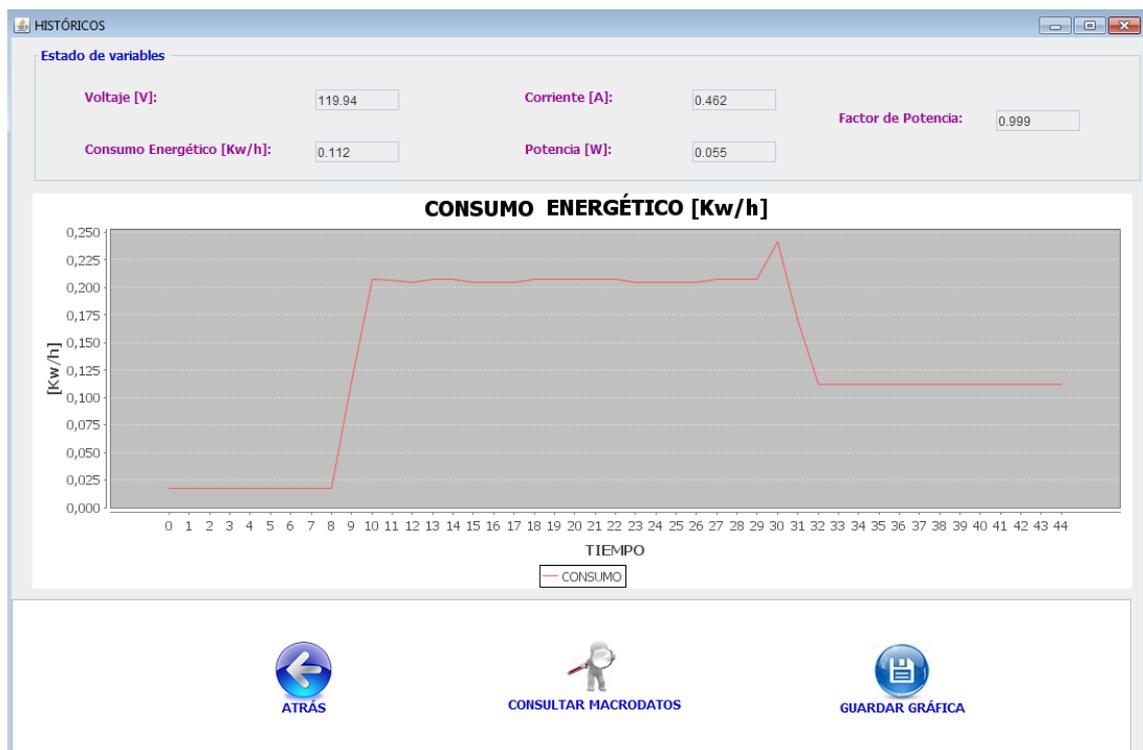


**Figura 3.4:** Ventana de seguridad del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

En la ventana de seguridad se aprecia los datos recibidos de los sensores y la cámara de seguridad, al probar la comunicación de datos de la cámara se pudo apreciar que el video es en tiempo real

no existe retardo, esto se debe a que se utilizó el protocolo de comunicación RSTP quien optimiza el envío de los datos de multimedia y hace más rápido la convergencia del video.

Desarrollado el algoritmo del código de operación de todos los equipos que intervienen en el sistema de control BMS, se adquieren todas las variables en la figura 15.4 en la ventana de históricos del sistema se comprobó la adquisición de datos adquiridos de los equipos, se observa las distintas variables eléctricas medidas y la gráfica en tiempo real del consumo energético de todo el sistema BMS.

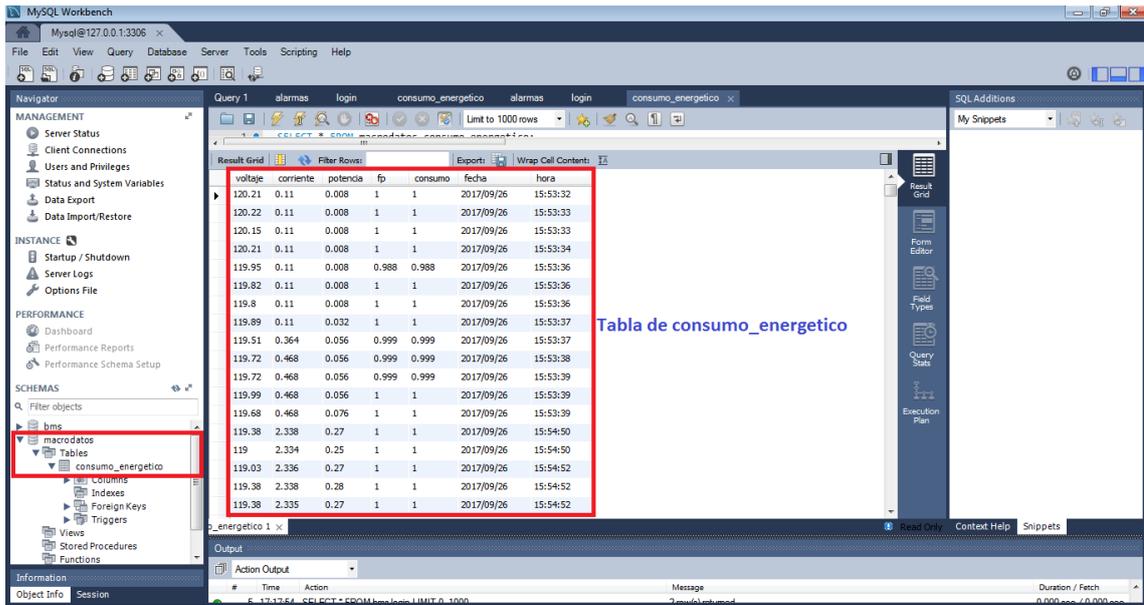


**Figura 15.4:** Ventana de históricos del BMS, adquisición de datos de variables eléctricas.  
Realizado por: Gómez Oscar, 2018.

### 4.3. Pruebas de generación de macrodatos del sistema de control BMS

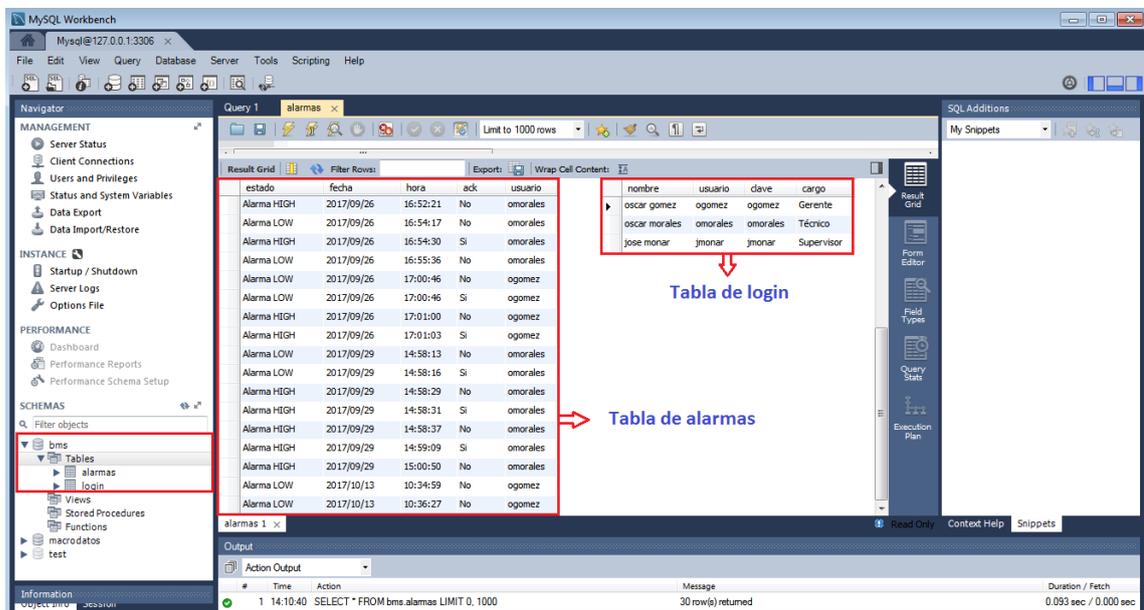
El sistema BMS guarda en una base de datos todas las variables eléctricas; generando un archivo de macrodatos, las pruebas se realizó a plena carga del módulo variando las cargas por diferentes días, para que se guarden en las tablas ya asignadas para la generación de los macrodatos en la figura 16.4 se observa el esquema bms con la tabla de consumo\_energetico con diferentes datos adquiridos desde el sistema BMS.

El guardado de los datos en la tabla de consumo\_energetico fue exitosa con todos los parámetros dispuestos para la generación de los macrodatos.



**Figura 16.4:** Guardado de datos del sistema BMS para la generación de los macrodatos.  
Realizado por: Gómez Oscar, 2018.

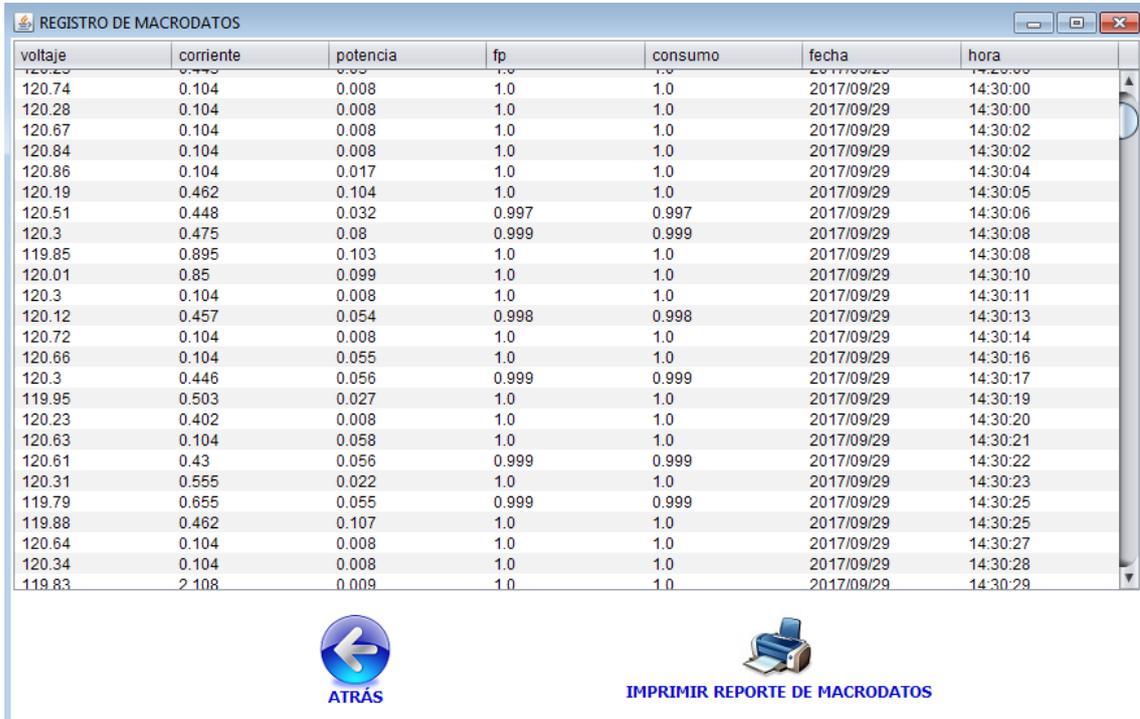
De la misma forma el esquema de la base de datos bms con las dos tablas login y alarmas fue el guardado exitoso con los parámetros de configuración de almacenamiento ya indicados en el capítulo anterior, en la figura 17.4 se observa las tablas con los datos adquiridos del sistema BMS.



**Figura 17.4:** Guardado de datos de alarmas y usuarios del sistema BMS.  
Realizado por: Gómez Oscar, 2018.

#### 4.4. Generación de reportes de macrodatos y alarmas

El sistema BMS implementado tiene la opción de generar reportes, desde el BMS se lee los datos generados por el sistema que están en las tablas de la base de datos y se genera un archivo XPS para poder imprimirlos, en la figura 18.4 se observa el registro de los macrodatos que se encuentra guardados en la base de datos del sistema.



| voltaje | corriente | potencia | fp    | consumo | fecha      | hora     |
|---------|-----------|----------|-------|---------|------------|----------|
| 120.74  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:00 |
| 120.28  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:00 |
| 120.67  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:02 |
| 120.84  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:02 |
| 120.86  | 0.104     | 0.017    | 1.0   | 1.0     | 2017/09/29 | 14:30:04 |
| 120.19  | 0.462     | 0.104    | 1.0   | 1.0     | 2017/09/29 | 14:30:05 |
| 120.51  | 0.448     | 0.032    | 0.997 | 0.997   | 2017/09/29 | 14:30:06 |
| 120.3   | 0.475     | 0.08     | 0.999 | 0.999   | 2017/09/29 | 14:30:08 |
| 119.85  | 0.895     | 0.103    | 1.0   | 1.0     | 2017/09/29 | 14:30:08 |
| 120.01  | 0.85      | 0.099    | 1.0   | 1.0     | 2017/09/29 | 14:30:10 |
| 120.3   | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:11 |
| 120.12  | 0.457     | 0.054    | 0.998 | 0.998   | 2017/09/29 | 14:30:13 |
| 120.72  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:14 |
| 120.66  | 0.104     | 0.055    | 1.0   | 1.0     | 2017/09/29 | 14:30:16 |
| 120.3   | 0.446     | 0.056    | 0.999 | 0.999   | 2017/09/29 | 14:30:17 |
| 119.95  | 0.503     | 0.027    | 1.0   | 1.0     | 2017/09/29 | 14:30:19 |
| 120.23  | 0.402     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:20 |
| 120.63  | 0.104     | 0.058    | 1.0   | 1.0     | 2017/09/29 | 14:30:21 |
| 120.61  | 0.43      | 0.056    | 0.999 | 0.999   | 2017/09/29 | 14:30:22 |
| 120.31  | 0.555     | 0.022    | 1.0   | 1.0     | 2017/09/29 | 14:30:23 |
| 119.79  | 0.655     | 0.055    | 0.999 | 0.999   | 2017/09/29 | 14:30:25 |
| 119.88  | 0.462     | 0.107    | 1.0   | 1.0     | 2017/09/29 | 14:30:25 |
| 120.64  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:27 |
| 120.34  | 0.104     | 0.008    | 1.0   | 1.0     | 2017/09/29 | 14:30:28 |
| 119.83  | 2.108     | 0.009    | 1.0   | 1.0     | 2017/09/29 | 14:30:29 |

**Figura 18.4:** Registro de macrodatos generados por el sistema BMS.

Realizado por: Gómez Oscar, 2018.

En la figura 19.4 se observa la generación del reporte de las alarmas del sistema BMS de la misma manera estos datos son leídos desde el BMS hacia las tablas de las base de datos, se puede afirmar que los datos leídos son correctos ya que se comparó las tabla de la base de datos con la generada por el sistema BMS.

Mientras el sistema BMS sigue generando alarmas y guarda en la base de datos el registro del sistema BMS se sigue actualizando constantemente.

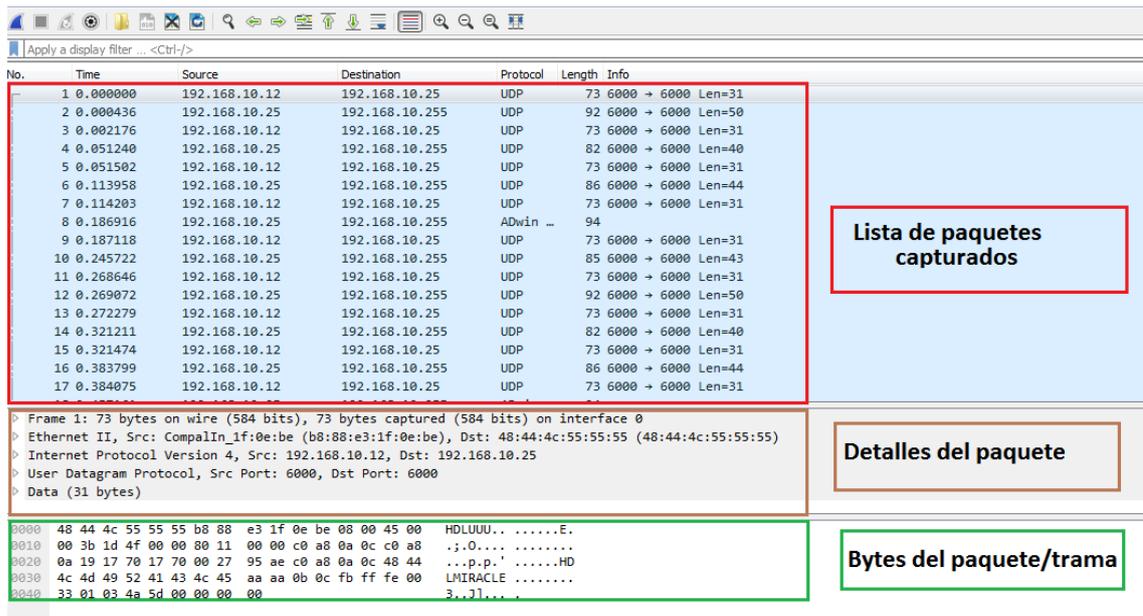
| estado      | fecha      | hora     | ack | usuario  |
|-------------|------------|----------|-----|----------|
| Alarma HIGH | 2017/09/26 | 17:01:03 | Si  | ogomez   |
| Alarma HIGH | 2017/09/26 | 17:01:00 | No  | ogomez   |
| Alarma LOW  | 2017/09/26 | 17:00:46 | Si  | ogomez   |
| Alarma LOW  | 2017/09/26 | 17:00:46 | No  | ogomez   |
| Alarma LOW  | 2017/09/26 | 16:55:36 | No  | omorales |
| Alarma HIGH | 2017/09/26 | 16:54:30 | Si  | omorales |
| Alarma LOW  | 2017/09/26 | 16:54:17 | No  | omorales |
| Alarma HIGH | 2017/09/26 | 16:52:21 | No  | omorales |
| Alarma HIGH | 2017/09/26 | 16:51:57 | Si  | omorales |
| Alarma HIGH | 2017/09/26 | 16:51:20 | No  | omorales |
| Alarma LOW  | 2017/09/26 | 16:50:55 | Si  | omorales |
| Alarma LOW  | 2017/09/26 | 16:50:48 | No  | omorales |
| Alarma LOW  | 2017/09/26 | 16:07:16 | Si  | ogomez   |
| Alarma LOW  | 2017/09/26 | 16:06:53 | No  | ogomez   |
| Alarma HIGH | 2017/09/26 | 16:06:29 | Si  | ogomez   |
| Alarma HIGH | 2017/09/26 | 16:06:23 | No  | ogomez   |
| Alarma LOW  | 2017/09/26 | 15:56:56 | No  | ogomez   |
| Alarma HIGH | 2017/09/26 | 15:56:04 | Si  | ogomez   |
| Alarma HIGH | 2017/09/26 | 15:55:56 | No  | ogomez   |
| Alarma LOW  | 2017/09/26 | 15:55:16 | Si  | ogomez   |
| Alarma LOW  | 2017/09/26 | 15:55:11 | No  | ogomez   |




**Figura 19.4:** Registro de alarmas generadas por el sistema BMS.  
**Realizado por:** Gómez Oscar, 2018.

#### 4.5. Análisis del tráfico de datos en el bus HDL del sistema de control BMS

Para analizar el tráfico de red al momento que se envía una petición desde el BMS a los equipos de HDL se utilizó el software Wireshark el cual es un analizador de protocolos open-source su principal objetivo es el análisis de tráfico de red, debido a que entiende la estructura de los protocolos, se pueden visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, este software consta con 3 paneles en la figura 20.4 se observa su interfaz.



**Figura 20.4:** Estructura del software Wireshark, 3 paneles.  
 Realizado por: Gómez Oscar, 2018.

En el panel de la lista de paquetes capturados muestra una tabla de los paquetes actuales capturados de la red la tabla contiene las siguientes columnas:

- No: la cantidad de paquetes capturados enumera en forma secuencial.
- Time: El tiempo que se capturo el paquete.
- Source: es la fuente es la dirección IP de donde se obtuvo el paquete.
- Destination: es el destino a donde se envía el paquete.
- Protocol: es el protocolo de comunicación utilizada para transmitir el paquete.
- Length: es el tamaño del paquete.
- Info: Información del paquete.

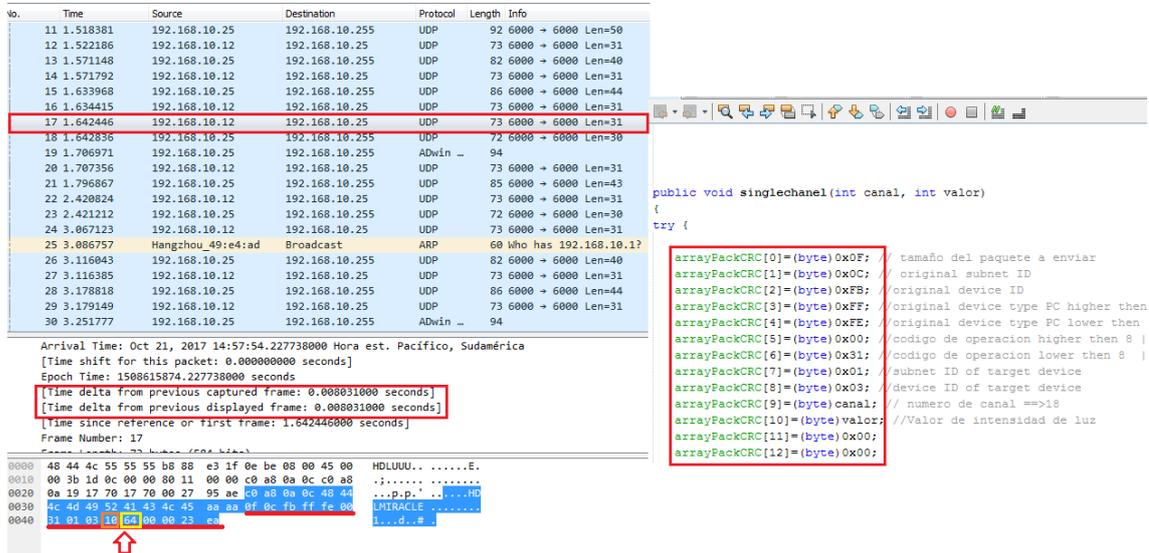
En el panel de detalle del paquete este contiene la información sobre el paquete o una trama esta opción permite verificar individualmente cada paquete.

En el panel de bytes del paquete/trama, muestra al paquete o trama a medida que viaja a través de la red.

#### 4.5.1. Análisis de una petición enviada desde el BMS al módulo de entrenamiento

Para la comprensión de la trama que se envía desde el BMS a los equipos de HDL se realizó un análisis detallado de la misma en la figura 21.4 se puede observar desde donde se envía la trama de datos en este caso de la pc que corresponde a la dirección IP 192.168.10.12 en donde se ejecuta

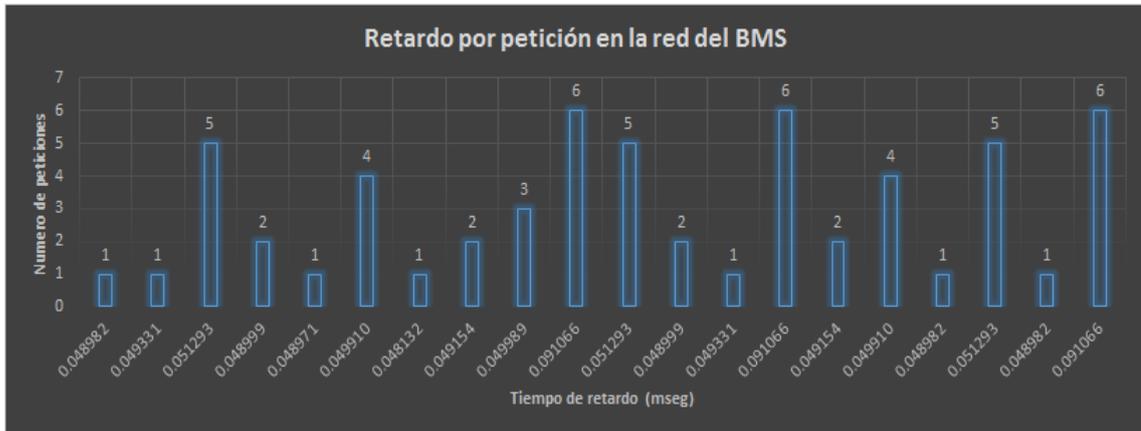
el BMS, mientras que el destino es el módulo compacto RCU que está asignado con la dirección IP 198.162.10.25, se observa el protocolo que se está utilizando que es el UDP este primeramente inicializa el puerto 6000, luego se envían los paquetes de datos, y finalmente se recibe el paquete de datos, en este caso de 31 bytes.



**Figura 21.4:** Datos capturados de la trama enviada desde el BMS a los equipos de HLD.  
Realizado por: Gómez Oscar, 2018.

El tiempo que se demoró en realizar la instrucción de encender una salida de relé fue de 0.008031000 segundos, se aprecia la trama envía desde el BMS en la parte derecha de la figura 21.4 y la trama capturada en Wireshark que definitivamente son iguales, la instrucción que se ejecuto fue encender a 100% (hexadecimal 64) la salida número 16 (hexadecimal 10) del equipo con ID número 3 de la sub red 01.

Para tener un promedio de cuando es el retardo por cada instrucción que se realiza desde el BMS hacia los equipos se adquirió 20 muestras con peticiones diferentes en la figura 22.4 se observa los datos y los tiempos de retardo de las tramas.



**Figura 22.4:** Tiempos de retardo de varias peticiones realizadas desde el BMS.  
**Realizado por:** Gómez Oscar, 2018.

Al tomar datos de la red por el número de peticiones que se realiza desde el BMS se tiene un promedio de 0,0535818 milisegundos que demora en realizar una acción el sistema BMS.

#### 4.6. Test SUS (System Usability Scale)

Para validar el sistema de control BMS con los usuarios se aplicó un test llamado SUS, Escala de Usabilidad del Sistema.

El cuestionario SUS fue desarrollado en 1986 como parte de la introducción de la ingeniería de usabilidad a los sistemas de oficina de Digital Equipment Co. Ltd. Su propósito era proporcionar un test fácil de completar (número mínimo de cuestiones), fácil de puntuar y que permitiera establecer comparaciones cruzadas entre productos. Ha sido usado extensivamente en evaluaciones de proyectos en Digital Equipment Co. Ltd (sistemas de oficina, dirección de sistemas, herramientas técnicas y sistemas de hardware), resultando como simple y fiable. (“System Usability Scale (SUS),” 2016)

##### 4.6.1. Utilización de la escala SUS

La escala SUS se utiliza generalmente después de que un usuario ha tenido la oportunidad de utilizar un sistema pero antes de que cualquier informe o discusión tenga lugar. Se solicitará a los usuarios el registro inmediato de su respuesta a cada punto, en lugar de pensar largamente en los mismos. Todos los puntos han de ser comprobados. Si el usuario no se siente capaz de responder a alguna cuestión en particular, habrá de señalar el valor central de la escala. (“System Usability Scale (SUS),” 2016)

Los resultados del test se tabularon en la tabla 3.4.

**Tabla 3.4:** Resultados del test SUS aplicado a los usuarios del sistema de control BMS.

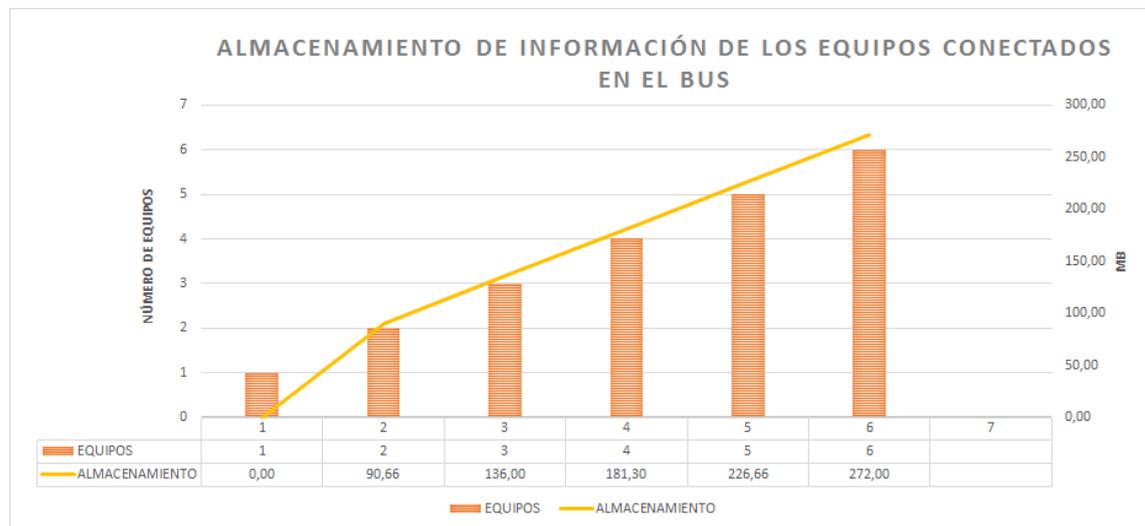
| TEST SUS<br>(System Usability Scale)                                | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 | Subject 6 | Subject 7 | Subject 8 | Subject 9 | Subject 10 | Subject 11 | Subject 12 | Subject 13 | Subject 14 | Subject 15 |
|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|------------|------------|------------|------------|------------|
| I think I would like to visit the BMS frequently.                   | 3         | 4         | 3         | 4         | 3         | 4         | 4         | 3         | 4         | 3          | 3          | 4          | 4          | 3          | 4          |
| I found the BMS unnecessarily complex.                              | 3         | 3         | 2         | 3         | 3         | 3         | 4         | 3         | 2         | 4          | 4          | 3          | 2          | 3          | 3          |
| I thought it was easy to use the BMS.                               | 4         | 4         | 4         | 4         | 3         | 3         | 4         | 4         | 4         | 3          | 4          | 4          | 4          | 4          | 4          |
| I think I would need the support of an expert to travel the BMS.    | 3         | 4         | 4         | 3         | 3         | 4         | 4         | 4         | 3         | 4          | 3          | 4          | 4          | 4          | 3          |
| I found the various possibilities of the BMS quite well integrated. | 4         | 4         | 3         | 3         | 3         | 4         | 3         | 3         | 3         | 4          | 4          | 4          | 3          | 3          | 3          |
| I thought there was too much inconsistency in the BMS.              | 3         | 3         | 4         | 3         | 3         | 4         | 4         | 4         | 2         | 3          | 4          | 3          | 4          | 3          | 4          |
| I imagine that most people would learn very quickly to use the BMS. | 3         | 4         | 4         | 4         | 4         | 4         | 4         | 4         | 3         | 4          | 3          | 3          | 4          | 4          | 4          |
| I found the BMS very large when traveling.                          | 4         | 2         | 3         | 2         | 3         | 3         | 3         | 3         | 4         | 4          | 3          | 3          | 4          | 3          | 2          |
| I felt very confident in handling the BMS.                          | 4         | 4         | 4         | 3         | 4         | 4         | 3         | 4         | 4         | 3          | 3          | 4          | 4          | 4          | 4          |
| I need to learn a lot of things before I get used to BMS.           | 4         | 4         | 4         | 4         | 4         | 3         | 3         | 3         | 4         | 3          | 4          | 3          | 3          | 4          | 4          |
| SCORE   | 87,5      | 90        | 87,5      | 82,5      | 82,5      | 90        | 90        | 87,5      | 82,5      | 87,5       | 87,5       | 87,5       | 90         | 87,5       | 87,5       |
| SUS BMS   | 87,166667 |           |           |           |           |           |           |           |           |            |            |            |            |            |            |

Realizado por: Gómez Oscar, 2018.

#### 4.7. Validación de la Hipótesis

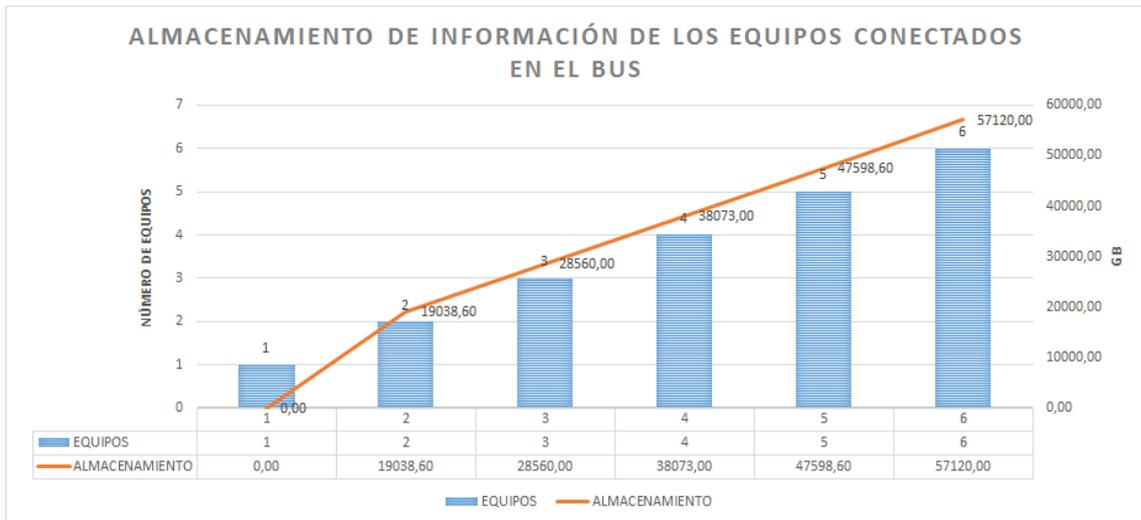
La validación de la hipótesis se la realizo de forma estadística, de acuerdo a los equipos conectados en el Bus doméstico HDL y la capacidad de memoria que consume por cada equipo; el consumo de memoria de la base de datos comienza a partir de la conexión de 2 equipos que es de 90,66 MB, 3 equipos 136 MB, 4 equipos 181,30 MB, 5 equipos 226,66MB y finalmente con 6 equipos que es el número máximo conectados en el bus es de 272 MB.

En la figura la figura 23.4 se observa el almacenamiento de la información de los equipos conectados en el bus, en el tiempo de un día.



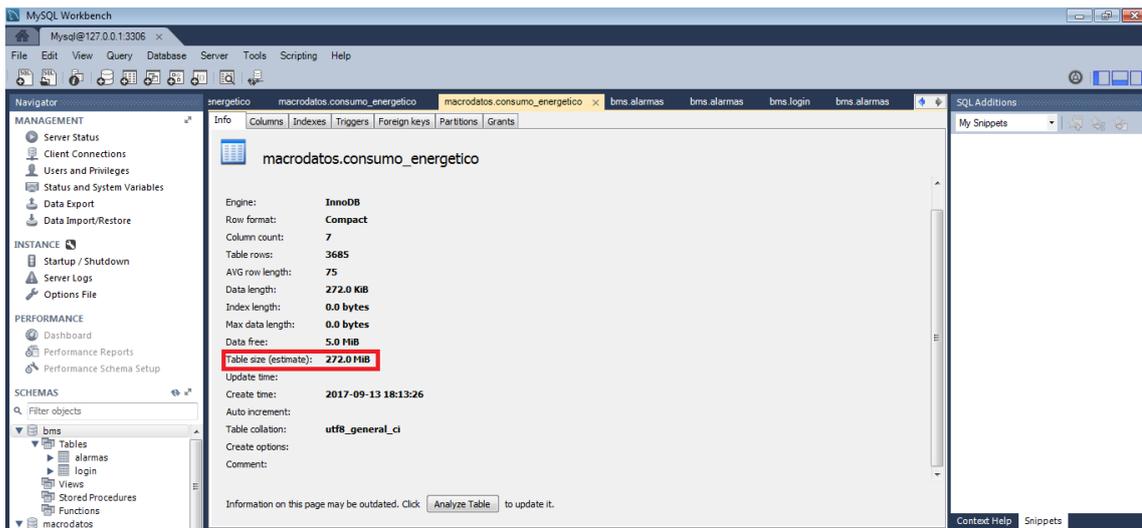
**Figura 23.4:** Almacenamiento de información de los equipos conectados en el bus HDL.  
**Realizado por:** Gómez Oscar, 2018.

En la base de datos se ha guardado 210 días de información de los equipos conectados en el bus, en la figura número 24.4 se puede observar la información guardada.



**Figura 24.4:** Almacenamiento de información de los equipos conectados en el bus HDL.  
**Realizado por:** Gómez Oscar, 2018.

Los datos consumidos de la base de datos se observa en la información de la tabla de MySQL Workbench, en la figura 25.4 se observa la información.



**Figura 25.4:** Información de los macrodatos guardados en la base MySQL Workbench.  
**Realizado por:** Gómez Oscar, 2018.

Para el almacenamiento de los macrodatos generados por el sistema BMS, se siguieron los estándares de almacenamiento de macrodatos los cuales se describen a continuación en la tabla número 4.4.

**Tabla 4.1:** Estándares de almacenamiento de macrodatos (Big Data).

| ESTÁNDAR |                     | DESCRIPCIÓN  | SISTEMA BMS   |
|----------|---------------------|--|---|
| 1        | NÚMERO DE VARIABLES | La cantidad de variables generadas pueden ser desde una hasta N variables.   | Tiene 16 variables (Voltaje, corriente, potencia, fp, consume, hora fecha, estado alarma, ack, usuario, nombre, cargo, clave) |
| 2        | VELOCIDAD           | En este contexto, la velocidad a la cual se refresca (tiempo) la adquisición de los macrodatos va desde los 5 ms hasta los 2 segundos.   | Los macrodatos se refrescan cada 1 ms, en las tablas.   |
| 3        | ESTRUCTURA          | <b>Estructurados:</b> Datos que tienen bien definido su longitud y su formato, como las fechas, los números o las cadenas de caracteres.<br><b>No estructurados:</b> Datos en el formato tal y como fueron recolectados, No se pueden almacenar dentro de una tabla.<br><b>Semiestructurados:</b> Datos que no se limitan a campos determinados, pero que contiene marcadores para separar los diferentes elementos  | Estructurado: Fechas, hora, usuarios, clave, estado de alarma, cargo.   |
| 4        | CAPTURA             | ¿De dónde provienen todos estos datos?, por personas, estaciones meteorológica, equipos computacionales, estaciones energéticas, estaciones astronómicas, sensor networks, etc.  | Equipos inteligentes que están conectado a un bus.<br>Gestionada por un sistema computacional.                                |
| 5        | ALMACENAMIENTO      | <b>Almacenamiento en grafo:</b> Donde se establece que la información guardada son en los nodos.<br><b>Almacenamiento clave-valor:</b> Los datos se almacenan de forma similar a los mapas.<br><b>Almacenamiento documental:</b> Guardan un gran parecido con las bases de datos Clave-Valor, diferenciándose en el dato que guardan.<br><b>Almacenamiento orientado a filas y columnas:</b> Se orienta a almacenar datos con tendencia a escalar horizontalmente y vertical, por lo que permite guardar diferentes atributos y objetos. | El almacenamiento es orientado a filas y columnas (tablas)  |

**Realizado por:** Gómez Oscar, 2018.

El sistema de control BMS permite guardar la información de los equipos que están conectados en el bus de HDL Buspro, se comprobó conectando de uno en uno los equipos hasta llegar al máximo número del sistema del BMS que son 6 dispositivos versus la memoria de la base de datos y observando a la par los esquemas de MySQL la generación de los macrodatos del sistema BMS que si cumple con los estándares de almacenamiento de los macrodatos, con lo cual se puede dar por verdadera la hipótesis planteada.

## CONCLUSIONES

A continuación se presenta las conclusiones obtenidas durante el desarrollo de esta Tesis, la cual expone el funcionamiento del sistema de control BMS, la generación de macrodatos y la transmisión de los datos por medio del protocolo UDP.

- Al analizar los tiempos de retardo de las peticiones enviadas y recibidas por el sistema BMS, en la red no existe problemas significativos en la transmisión de datos las velocidades de transmisión son altas y con bajas tasas de errores si al futuro se quisiera colocar nuevos equipos en el bus HDL y ampliar el sistema no habría ningún inconveniente en la red del BMS.
- Al emplear una topología de red tipo Bus, tiene la característica que cuando el tráfico es bajo, las velocidades de transmisión son altas y con bajas tasas de error; mientras que el rendimiento es bajo cuando el tráfico es intenso. Además hace posibles modificaciones o ampliaciones del sistema, ya que se pueden agregar o suprimir equipos fácilmente al bus.
- Los macrodatos generados por el sistema BMS, se pueden utilizar conjuntamente con un sistema basado en Machine Learning, para optimizar el consumo energético de la edificación.
- Los macrodatos generados en esta investigación servirán a futuro para diseñar un plan de mantenimiento de una edificación y tomar correctivos del mismo.
- La generación de macrodatos consume gran cantidad de espacio en disco por lo que se debe realizar un backup cada 6 meses para evitar saturar el dispositivo de almacenamiento.
- Al implementar la comunicación UDP en el sistema BMS, se ejecutó un broadcast en el sistema para el accionamiento de los equipos conectados en el bus; el envío y recepción de los datagramas fue fiable sin pérdidas de datos en el sistema.

## RECOMENDACIONES

A continuación se presenta las recomendaciones más importantes durante el desarrollo del trabajo de titulación.

- Al conectar el sistema de control BMS con el modulo donde se encuentra los equipos conectados al bus HDL, verificar que el puerto 6000 este abierto caso contrario se tendrá problemas de comunicación.
- Tener cuidado en ejecutar dos veces al mismo tiempo el sistema BMS por lo que la primera vez el sistema abre el puerto 6000 y al ejecutar otra vez al sistemas el puerto está ocupado por lo tanto se genera conflictos al tratar de abrir un puerto que ya está abierto.
- Al enviar a ejecutar una acción del sistema BMS hacia al módulo donde se encuentra los equipos conectados al bus HDL, y no responder a la acción solicitada, se deberá mandar nuevamente la acción esto se debe a que el protocolo UDP no garantiza que los datagramas lleguen a su destino o pueden llegar los datagramas duplicados descompuestos o perdidos sin aviso.
- Al ejecutar la ventana de seguridad esperar unos 10 segundos y no realizar ninguna otra acción por lo que al iniciar la ventana de seguridad el sistema manda abrir el puerto 6000 y al realizar otra acción va a dar conflictos de comunicación en la función moSocket, el retardo se debe a que la cámara se está conectando al sistema BMS por el protocolo de RSTP este protocolo lo primero que hace es enlazar los datos y verificar el estado de todas las trayectorias de la comunicación por ese motivo es el tiempo de demora en la ejecución de la ventana.
- Se recomienda la implementación de estos sistemas de control BMS en instalaciones como hospitales, aeropuertos, universidades, centros comerciales, hoteles, puertos marítimos, etc. Que lleguen a instalar estos sistemas para obtener un alto grado de sostenibilidad energética, destacando el ahorro energético y permitiendo a las personas que habitan o trabajan en esas instalaciones darles un nivel de confort adecuado para realizar sus operaciones mejorando su productividad.

## BIBLIOGRAFÍA

- Becario, N. D. E. L.** (2012). *Màster en Enginyeria de l ' Energia*.
- BUSPRO, H.** (2016). Buspro Introducción al Sistema Contenidos.
- Carnero, V.** (2014). Sistemas de Automatización y Control de Instalaciones de Hospitales.
- Carolina Valencia Gil, Carolina Henao Acosta, juan P. O. V.** (2010). Desarrollo de Software Orienta a Objetos. Retrieved December 19, 2016, from <http://es.slideshare.net/juanpabloov18/desarrollo-de-software-orienta-a-objetos>
- Chapter, T., Bus, H. D. L., Code, O., Chapter, I. T., Codes, O., & Pro, H. D. L. B.** (2014). System Specifications Networking Operation Code This is confidential documents , it is forbidden to System specifications.
- CRISTOBAL ROMERO, FRANCISCO VASQUEZ, C. D. C.** (2009). *DOMÓTICA E INMÓTICA- VIVIENDAS Y EDIFICIOS INTELIGENTES*. (RA-MA., Ed.) (2DA ed.).
- Design Guidance Intelligent building control system, HDL BUS.** (2015).
- Design Guidance Intelligent building control system SINCE 1985 HDL-BUS.** (2016).
- HISPA LINUX.** (2011). ¿Qué es el Software Libre? | Hispalinux. Retrieved December 19, 2016, from <http://hispalinux.es/SoftwareLibre>
- Información NetBeans IDE 6.1.** (2012). Retrieved September 10, 2017, from [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html)
- Ken Constantino Daniel Fernando, P. hu J. E.** (2008). Características, ventajas y desventajas de las tecnologías de desarrollo Ruby on Rails y PHP – Blog para las publicaciones de nuestras investigaciones ITIC92. Retrieved December 19, 2016, from <https://jditic92.wordpress.com/2015/05/11/caracteristicas-ventajas-y-desventajas-de-las-tecnologias-de-desarrollo-ruby-on-rails-y-php/>
- macrodatos e inteligencia de datos, alternativas a big data.* (2016). Retrieved from <http://www.fundeu.es/recomendacion/macrodatosalternativa-abig-data-1582/>
- Molina, C. E.** (2014). Protocolo de datagramas de usuario (UDP).
- Plan Nacional Buen Vivir 2013-2017.pdf.** (n.d.).
- Red Iberoamericana de Gestión y Eficiencia Energética para un Desarrollo sostenible.**

(2011). Eficiencia Energética en edificios, 257.

**Rocio Basa.** (2007). ¿Que es la Domótica?: Domotica e Inmotica. Retrieved February 12, 2017, from <http://rociobasa7.blogspot.com/2007/09/domotica-e-inmotica.html>

**System Usability Scale (SUS).** (2016). Retrieved November 23, 2017, from <https://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/CuestCon.htm>

**TREND.** (2016). **Sistema BMS.** Retrieved December 15, 2016, from <https://www.trendcontrols.com/es-ES/bmssystem/Paginas/default.aspx>

**User, U. D. P., & Protocol, D.** (2014). Protocolo UDP, 2014.

## ANEXOS

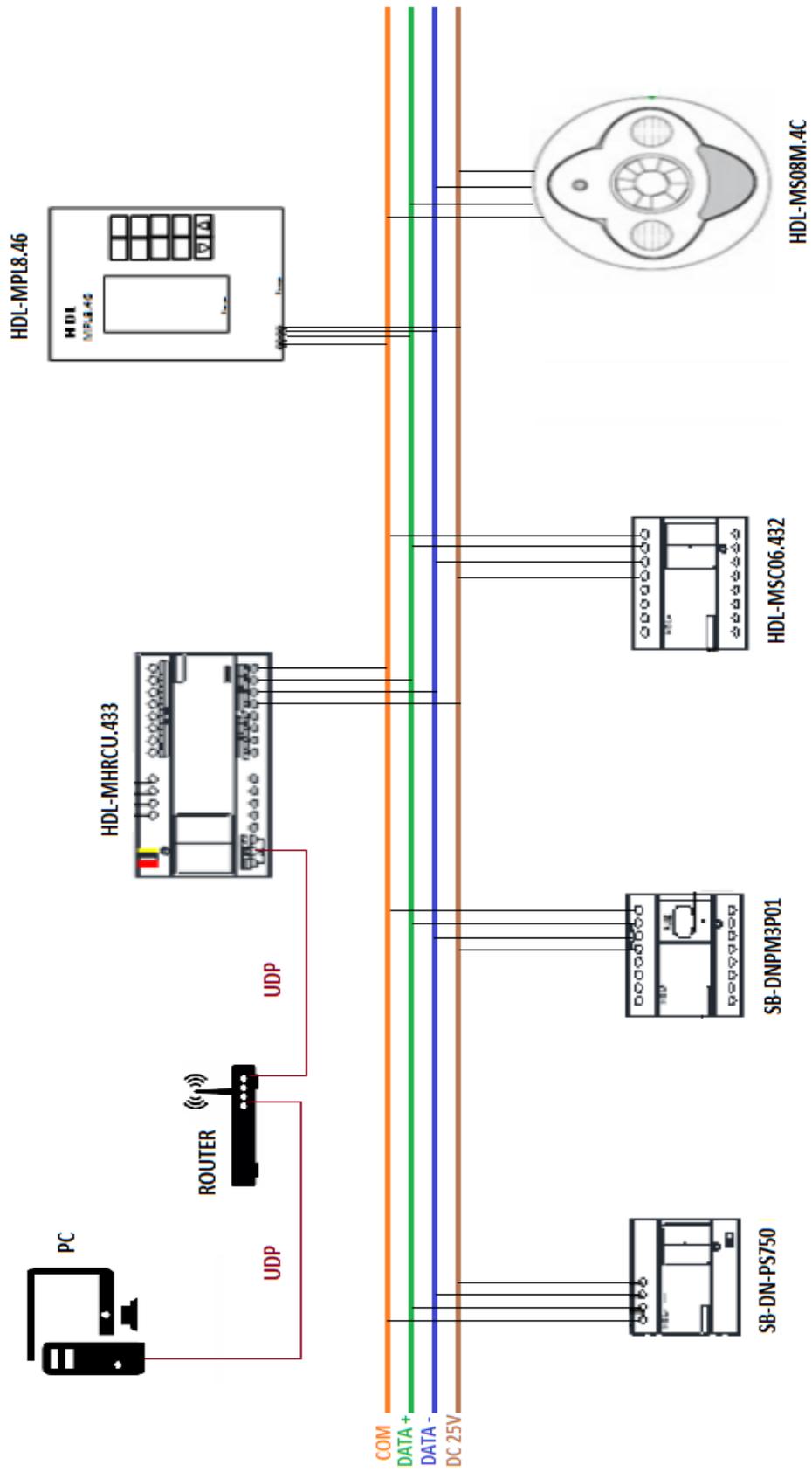
### Anexo A: Test SUS aplicado a los usuarios del sistema

#### ***System Usability Scale***

© Digital Equipment Corporation, 1986.

|  | Strongly disagree        |                          |                          |                          | Strongly agree           |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 1. I think that I would like to use this system frequently                                   | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 2. I found the system unnecessarily complex  | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 3. I thought the system was easy to use  | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 4. I think that I would need the support of a technical person to be able to use this system | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 5. I found the various functions in this system were well integrated                         | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 6. I thought there was too much inconsistency in this system                                 | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 7. I would imagine that most people would learn to use this system very quickly              | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 8. I found the system very cumbersome to use   | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 9. I felt very confident using the system  | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |
| 10. I needed to learn a lot of things before I could get going with this system              | <input type="checkbox"/> |
|  | 1                        | 2                        | 3                        | 4                        | 5                        |

## Anexo B: Diagrama de Conexión.



## Parameters

| Electric Parameters:      |                                 |
|---------------------------|---------------------------------|
| Working voltage           | DC20 - 30V                      |
| Static power consumption  | 60mW/DC24V                      |
| Dynamic power consumption | 300mW/DC24V                     |
| Signal interface          | HDL Buspro, RJ45 , INNER Buspro |
| RJ45 interface            | UDP/IP network Interface        |
| TRIAC                     | 16A TRIAC, minimum load is 30w  |
| Buspro terminal           | 0.75-0.85mm single-core cable   |
| Relay output              | 5A/10A                          |
| Environmental Conditions: |                                 |
| Working temperature       | -10°C~45°C                      |
| Working relative humidity | Up to 90%                       |
| Storage temperature       | -20°C~+60°C                     |
| Storage relative humidity | Up to 93%                       |
| Approved:                 |                                 |
| CE                        |                                 |
| RoHS                      |                                 |
| Production Information:   |                                 |
| Dimension                 | 216*90*66 ( mm )                |
| Weight                    | 725g                            |
| Housing material          | Nylon                           |
| Installation              | 35mm DIN rail installation      |
| Protection degree         | IP20                            |

## HDL Buspro Definition for Cable

| HDL Buspro | HDL Buspro/KNX |
|------------|----------------|
| DC24V      | Red            |
| COM        | Black          |
| DATA-      | White          |
| DATA+      | Yellow         |

## Important Notes

- Installation - Distribution box (DB)
- Buspro cable - CAT6E or HDL Buspro/KNX cable, 0.8mm single-core copper cable.
- Buspro connection - Series connection (hand-in-hand).
- RJ45 cable - CAT6E.
- Make sure the IP address and MAC address are unique in IP network.
- Fuse must be aR type, and current should not exceed 10A.
- When you replace the fuse, you should turn off the power.

## Overview



HDL-MHRCU.433 is the hotel room controller which is the core of the hotel system, it has mix controller, doorbell status display, 24-channel dry contacts and LED outputs. Hotel controller can support 48 channels. From the hotel management software (IHMS), it can control and monitor doorbell panel, curtain, lights, fan and AC.

## Functions

### The host control functions

- It supports 48 channels, each dimmer channel has parameters for lower limit, the higher limit, the maximum limit.
- Total is 99 scenes, each scene's running time is 0-250s.
- Each relay channel has protection delay time (0-60mins).
- Each relay channel can delay turn ON time (0-250s).
- Can run the scene according to the status of card master.
- Hotel host has RJ45 port for connection with reception or management center. With hotel management software, this module can integrate with third-party management software.

### Mix control functions

- Channel 1 to 13 are relay outputs, each channel is 5A. The load types are incandescent lamp, halogen lamp, low voltage halogen lamp.
- Channel 14 to 17 are relay outputs, each channel is 10A. The load types are incandescent lamp, halogen lamp, low voltage halogen lamp.
- Channel 18 to 21 are dimming outputs, each channel is 1A. It has short circuit protection, and the fuse is easy to replace. If the channel is short circuit or the fuse is wrong, the fuse indicator will flash.
- Channel 22 is relay for doorbell, it can control the doorbell status.

### Dry contract

- 24-channel dry contract inputs. (The control target should be set from IHMS)

### Door bell function

- 5-channel LED outputs. (it can connect to the third-party door bell panel.)

## Parameters

| Electrical Parameters (from HDL-MPPI.46): |                          |
|---|--------------------------|
| Power supply                              | DC12-30V                 |
| Power consumption (from HDL-MPPI.46)      | 30mADC24V                |
| Environmental Conditions:                 |                          |
| Working temperature                       | -5°C~45°C                |
| Working relative humidity                 | Up to 90%                |
| Storage temperature                       | -20°C~+60°C              |
| Storage relative humidity                 | Up to 93%                |
| Approved                                  |                          |
| CE  |                          |
| RoHS                                      |                          |
| Product Information:                      |                          |
| LCD resolution                            | 160x80                   |
| Dimensions                                | 86×116.5×10.5 (mm)       |
| Weight                                    | 130.5(g)                 |
| Housing material                          | Glass/Aluminum , ABS, PC |
| Installation                              | US wall box              |
| Protection degree                         | IP20                     |

## Important Notes

- It must work in conjunction with HDL-MPPI.46 (be installed in wall-box).
- Buspro cable — HDL Bus/KNX cable, 0.8mm single-core copper cable.
- Buspro Connection — Series connection (hand-in-hand).
- Installation — US wall box.

## Installation Step

- Connect bus cables. Make sure the color of wire complies with the definition.
- Make sure the Bus cable type is correct and has no short circuit.
- Mount the HDL-MPPI.46 in the wall-box.
- Put this device into HDL-MPPI.46.

## HDL Buspro Definition for Cable

| CATS/CAT5E                | HDL Buspro | HDLBuspro/KNX |
|---------------------------|------------|---------------|
| Brown/ Orange             | DC24V      | Red           |
| Brown white/ Orange white | COM        | Black         |
| Blue white/Green white    | DATA-      | White         |
| Blue/ Green               | DATA+      | Yellow        |

## Overview



**HDL-MPL8.46** 8-button multifunction LCD switch is a new generation of wall-mounted switch with new design. It has 8 control buttons and the left & right side of the button can be programmed separately. 2 buttons are used for the page switching. Each button can be displayed with an icon in the LCD screen, which is customized for final user.

## Functions

- Adjustable LCD backlight and LED indicator.
- Specified page for Floor Heating, HVAC, Music.
- Icon for Key Buttons can be customized by users.
- Button combination and Double Button combination function available.
- Button mutual exclusion available.
- Upgrading online from HDL Buspro is available.
- Built-in IR receiver, need to replace a plate with an IR receiving hole to enable the IR remote control function.
- Multi Key Mode: Invalid, Single on-off, single on, single off, Combination on-off, combination on, combination off, double click/single on-off, double click/combination on-off, momentary, clock, short/long press, short press/long momentary press, hyperlink.
- Multi Key control Mode: Scene, Sequence, timer switch, universal switch, single channel on-off, broadcast scene, broadcast channel, curtain, GPRS Control, Panel Control, alarm control, music play, General control.
- Functions including the following: option IR Receiver, Lock, Turn on/off AC, Cooling & heating for AC, Auto temperature setting, Fan Speed regulating, AC Mode setting, AC Temperature regulating, backlight adjusting, AC Lock, Backlight and status indicator light setting, floor heating temperature and mode setting etc.
- Communication : HDL Buspro.

## Parameters

| Electrical Parameters:          |                            |
|---------------------------------|----------------------------|
| Buspro working voltage          | DC12~30V                   |
| Buspro power consumption        | 60mA/DC24V                 |
| Measuring voltage               | AC100~250V                 |
| Maximum current in each channel | 16A                        |
| Environmental Conditions:       |                            |
| Working temperature             | -5°C~45°C                  |
| Working relative humidity       | Up to 90%                  |
| Storage temperature             | -20°C~+60°C                |
| Storage relative humidity       | Up to 93%                  |
| Approved:                       |                            |
| CE                              |                            |
| RoHS                            |                            |
| Product Information:            |                            |
| Dimensions                      | 72×90×66 (mm)              |
| Weight                          | 242g                       |
| Housing material                | Nylon, PC                  |
| Installation                    | 35mm DIN rail installation |
| Protection degree               | IP20                       |
| Installation Position           | Distribution Box (DB)      |

## HDL Buspro Cable Guide

| CAT6/CAT6E                | HDL Buspro | HDL Buspro/KNX |
|---------------------------|------------|----------------|
| Brown/ Orange             | DC24V      | Red            |
| Brown white/ Orange white | COM        | Black          |
| Blue white/Green white    | DATA-      | White          |
| Blue/ Green               | DATA+      | Yellow         |

## Installation Step

- Mount the device on the DIN rail inside the DB box.
- Mark up the load cable and power supply cable.
- Make sure the connection of load and HDL Buspro is correct.
- Check the HDL Buspro connection.

## Safety precautions



- The screw down strength is less than 0.4Nm
- Do not make wrong connection on Buspro interface, it will damage the Buspro interface of this module
- Never let liquids get into the module, it will damage this device
- Do not get AC power into Buspro wire, it will damage all devices in the system
- Avoid contact with liquids and aggressive gas

## Overview



SB-DN-PM1P03 is a single-phase intelligent digital power meter that measures and records electric parameters. It can be used for power consumption recording. It can record and save the electrical parameters for the last year. And the current, voltage, power and power factor can also be measured by the meter. The current, voltage, power consumption, time, UV Switch, etc. It can be programmed for the logic input. It has 3 separate channels, the maximum current for each channel is 16A. The extra current transformer can be added in high power detection.

## Functions

- The voltage, current, power, active power, reactive power, apparent power, power factor, etc.
- 24 Logic Blocks: The Phase voltage, current of each channel, time, channel electric degree in specified time interval, Universal Switch can be used as the logic inputs to trigger the control target. The logic relations are AND, OR, they can be set to trigger the delay time.
- Recorded data can be stored for one year. The record can be checked from the application software by month, day, hour or minute.

## Important Notes

- Buspro cable – CAT6E or HDL Buspro/KNX cable
- Buspro Connection – Series connection (hand-in-hand)
- Maximum current in each channel is 16A, when it is not connected to the current transformer



## Parameters

| Electrical Parameters:        |                        |
|-------------------------------|------------------------|
| Buspro power supply           | DC12-30V               |
| Static power consumption      | 40mADC24V              |
| Dynamic power consumption     | 90mADC24V              |
| Range of temperature sensor   | -20°C to 60°C          |
| IR transmit frequency         | 38KHz                  |
| IR emission distance          | 4m                     |
| Illumination detection range  | 0-5000Lux              |
| PIR sensing range in diameter | 6m (install height-3m) |
| Ultrasonic sensor in diameter | 8m                     |
| Environmental Conditions:     |                        |
| Working temperature           | -5°C~45°C              |
| Working relative humidity     | Up to 90%              |
| Storage temperature           | -20°C~+60°C            |
| Storage relative humidity     | Up to 93%              |
| Approved                      |                        |
| CE                            |                        |
| RoHS                          |                        |
| Product Information:          |                        |
| Dimensions                    | 110(Diameter)×33(mm)   |
| Weight                        | 206.7(g)               |
| Housing material              | Lens, ABS              |
| Installation                  | Ceiling mount          |
| Protection degree             | IP20                   |

## Important Notes

- Installation Location — If use PIR function indoor, please keep it away from AC outlets and heat source.
- Environment temperature — if use PIR function, please make sure the temperature is below 36°C.
- Detect target — PIR only can detect moving objects with heat radiation like human, animal, ultrasonic can detect all moving objects.
- Buspro cable — CAT5E or HDL Buspro/KNX cable.
- Buspro connection — Series connection (hand-in-hand).
- Installation — Indoor and ceiling-mount, keep it away from AC outlet.

## Safety Precautions



- The screw down strength should not exceed 0.1Nm.
- Do not make wrong connection on Buspro interface, it will damage the Buspro interface of this module.
- Do not get AC power into Buspro wire, it will damage all devices in the system.
- Avoid contact with liquids or corrosive gases.

## Overview



**SB-CMS-12in1** 12 in 1 sensor includes temperature sensor, PIR motion sensor, LUX sensor, ultrasonic sensor, dry contacts, IR received, IR emitter, 2-channel 5A relay output and logic block. The logic block can combine all sensors for different applications. It supports HDL security mode and command.

## Functions

- Temperature, LUX, IR motion, ultrasonic, 2 dry contact inputs, 2 external inputs, logic, totally 9 inputs.
- 2 logic Relations: OR, AND.
- 24 logic blocks function, maximum 9 logic inputs, up to 20 control targets in each logic block.
- 2-channel 5A Relay output.
- Build PID for constant LUX control.
- Up to 40 IR receiving control target.
- Up to 240 IR sending control target.
- Maximum 24 programmable logic blocks.
- Security function, to be used with the security module.
- Supports upgrade from HDL Buspro.

## HDL Buspro Definition for cable

| CAT5/CAT5E                | HDL Buspro | HDL Buspro/KNX |
|---------------------------|------------|----------------|
| Brown/ Orange             | DC24V      | Red            |
| Brown white/ Orange white | COM        | Black          |
| Blue white/Green white    | DATA-      | White          |
| Blue/ Green               | DATA+      | Yellow         |

## Parameters

| Electric Parameters:          |                            |
|-------------------------------|----------------------------|
| Working power                 | DC16-30V                   |
| Static BUS power consumption  | 20mA/DC24V                 |
| Dynamic BUS power consumption | 200mA/DC24V                |
| Relay output channel          | 10A                        |
| TRIAC output channel          | 2A                         |
| Environmental Conditions:     |                            |
| Working temperature           | -5°C~45°C                  |
| Working relative humidity     | Up to 90%                  |
| Storage temperature           | -20°C~+60°C                |
| Storage relative humidity     | Up to 93%                  |
| Approved:                     |                            |
| CE                            |                            |
| RoHS                          |                            |
| Production Information:       |                            |
| Dimensions                    | 144×90×66 (mm)             |
| Weight                        | 417.5g                     |
| Housing material              | Nylon, PC                  |
| Installation                  | 35mm DIN rail installation |
| Protection degree             | IP20                       |

## Installation Step

- Mount the device on the DIN rail inside the DB box
- Mark up each channel.
- Connect the load and HDL Buspro, check if there is any short circuit in output connection cable.
- Check the HDL Buspro connection, avoid any mistakes.
- Mark up each output connection cable, separate high power and low power cable.

## HDL Buspro Definition for Cable

| HDL Buspro | HDL Buspro/KNX | CAT5/CAT6E               |
|------------|----------------|--------------------------|
| COM        | Black          | Brown White/Orange White |
| DATA-      | White          | Blue White/Green White   |
| DATA+      | Yellow         | Blue/Green               |
| DC24V      | Red            | Brown/Orange             |

## Overview



SB-DN-HMX12 is a mix controller. It has the output of relay, dimming and 0-10V analog functions. It can be used for dimming control, on/off control of any devices, fluorescent lamp control, curtain control, etc.

## Functions

- Channel No.10 and channel No. 11 control the lamps for controllable silicon dimming, current does not exceed 2A output.
- Channel 1 to channel 8 are relay output channels, each channel is 10A.
- Channel No.12 is DC0~10V output. It controls the fluorescent lamp together with the relay output channel.
- Channel No.9 is curtain control.
- It has 12 areas, each area has 48 scenes.

## Important Notes

- Bus cable - HDL Buspro/KNX cable, 0.8mm single-core copper cable.
- Bus Connection - Series connection (hand-in-hand)
- Connection checking - Check all connection after installation.
- Output Channel - Maximum current of each channel is 2A.
- Type of Load - incandescent lamp, halogen lamp, LED light. Leading Edge dimming technology is available.
- Make sure the working temperature does not exceed 60°C.
- Channel No.10 and No.11 are TRIAC channels, they need to connect with a 4A aR fuse for protection.
- Connect with breakers or fuses for relay module.

## Parameters

| Electrical Parameters:    |                            |
|---------------------------|----------------------------|
| Input voltage             | AC240V±10% 50~60Hz         |
| Output current            | 750mA                      |
| Output voltage            | DC24V                      |
| Output ripple wave        | Less than 150mV            |
| Environmental Conditions: |                            |
| Working temperature       | -5°C~45°C                  |
| Working relative humidity | Up to 90%                  |
| Storage temperature       | -20°C~+60°C                |
| Storage relative humidity | UP to 93%                  |
| Approved:                 |                            |
| CE                        |                            |
| RoHS                      |                            |
| Product Information:      |                            |
| Dimensions                | 72×90×66 (mm)              |
| Weight                    | 188g                       |
| Housing material          | Nylon, PC                  |
| Installation              | 35mm DIN rail installation |
| Protection degree         | IP20                       |
| Installation Position     | Distribution Box (DB)      |

## HDL Buspro Definition for Cable

| CAT5/CAT5E                | HDL Buspro | HDL Buspro/<br>KNX |
|---------------------------|------------|--------------------|
| Brown/ Orange             | DC24V      | Red                |
| Brown white/ Orange white | COM        | Black              |
| Blue white/Green white    | DATA-      | White              |
| Blue/ Green               | DATA+      | Yellow             |

## Important Notes

- Buspro cable – CAT5E or HDL Buspro/KNX cable
- BUSPRO Connection - Series connection (hand-in-hand)
- When system needs more power supplies, please pay attention to the correct connection, see the connection diagram in right side.
- 110V AC power supply is not supported

## Overview



SB-DN-PS750 is a power supply module. The module can supply DC24V voltage , 750mA current.

## Functions

- Protection: Short Circuit and Over Current protection
- Voltage Supply: DC24V
- Ripple Wave: Less than 150mV

## Installation Step

- Mount the device on the DIN rail inside the DB Box.
- Connect HDL BUSPRO cable.
- Check the connection, avoid any mistakes.