



La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**AHORRO ENERGÉTICO EN ALUMBRADO PÚBLICO CON EL
DESARROLLO DE UN PROTOTIPO DE SISTEMA DE
TELEGESTIÓN REMOTO PARA LÁMPARAS TIPO LED DE LA
EMPRESA ELÉCTRICA EMELNORTE S.A.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELÉCTRICO**

FLORES LORA RICHARD ORLANDO

DIRECTOR: DR. ING. GABRIEL BENJAMÍN SALAZAR YÉPEZ

Quito, mayo 2018

AVAL

Certifico que el presente trabajo fue desarrollado por Richard Orlando Flores Lora, bajo mi supervisión.

Dr. Ing. Gabriel Salazar Y.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Richard Orlando Flores Lora, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Richard Orlando Flores Lora

DEDICATORIA

Este trabajo lo dedico a mis padres Fabián y Raquel, quienes fueron los impulsores de este nuevo logro, que con sacrificio y confianza hicieron de mí el hombre que soy ahora, infundiéndome en mí todos los valores que me llevaron a este día.

A mis hermanos que me apoyaron y exigieron ser el mejor, y demostrármelos que, si se puede, de quienes espero sigan mis pasos y logren más que yo.

A mi tía Marty y mi abuelita Enriquetita que me apoyaron incondicionalmente, y que gracias a ellas mi vida en Quito fue la mejor etapa que he vivido.

Al amor de mi vida Ery, que estuvo conmigo durante todos los años de vida estudiantil, apoyándome y ayudándome a superar cada batalla.

AGRADECIMIENTO

La elaboración de este proyecto de titulación se la debo en primer lugar al Ing. Edison Eche, jefe del Centro de Control SCADA y al Ing. Felipe Aguirre, jefe del Departamento de Alumbrado Público de EMELNORTE S.A., quienes fueron los mentores y auspiciantes del tema, gracias a ellos se logró obtener la aceptación del proyecto dentro de la empresa, la gestión administrativa, las luminarias de prueba y toda la información necesaria para su desarrollo.

En segundo lugar, agradezco al Dr. Gabriel Salazar, quien infundió en mí un mayor entusiasmo por la eficiencia energética, que sin su aprobación, generosidad y total apoyo no hubiera sido posible emprender este último y más grande reto universitario.

A la Escuela Politécnica Nacional y a todos los profesores que pasaron por mi vida estudiantil, quienes impartieron sus valiosos conocimientos en pro de mi superación personal y académica, muchos de ellos se ganaron un lugar especial en mis buenos recuerdos, cada uno con su propia forma de enseñar han logrado llegar a sus estudiantes.

El mayor agradecimiento es para mis padres, quienes sacrificaron mucho para poder permitirme ser parte de la mejor universidad del Ecuador, que sin ellos jamás hubiera sido posible cumplir este sueño, se los debo todo a ellos, y no olvidar a mis hermanos quienes infundían en mí un motivo para triunfar y demostrarles que todo es posible cuando tienes apoyo y ganas de hacerlo.

A mi amor y compañera de vida estudiantil, quien me apoyó hasta en los momentos más desesperantes, quien me brindó palabras de aliento y ayuda incondicional durante muchos años, por ella jamás me rendí, se lo agradezco enormemente.

A todos mis amigos, que durante la vida universitaria no dejaron que el estrés y el aburrimiento se apodere de mí, sin bullying no hay amistad, y por experiencia tampoco hay diversión.

Y, por último y no menos importantes, un agradecimiento especial a Larry Page, Sergey Brin y Jawed Karim, Steve Chen, Chad Hurley, quienes facilitaron el proceso de aprendizaje implementando las herramientas más usadas en todo el mundo, para bien o para mal, han sido un uso fundamental en la investigación de este proyecto.

ÍNDICE DE CONTENIDO

| | |
|---|------|
| AVAL..... | I |
| DECLARACIÓN DE AUTORÍA | II |
| DEDICATORIA | III |
| AGRADECIMIENTO | IV |
| ÍNDICE DE CONTENIDO..... | V |
| ÍNDICE DE FIGURAS | IX |
| ÍNDICE DE TABLAS | XI |
| ÍNDICE DE ECUACIONES | XII |
| RESUMEN | XIII |
| ABSTRACT..... | XIV |
| CAPÍTULO 1..... | 1 |
| 1. INTRODUCCIÓN..... | 1 |
| 1.1 Situación actual del alumbrado público de EMELNORTE | 1 |
| 1.2 Justificación | 1 |
| 1.3 Problemática | 2 |
| 1.4 Objetivos..... | 4 |
| 1.4.1 Objetivo General | 4 |
| 1.4.2 Objetivos Específicos | 4 |
| 1.5 Alcance | 4 |
| 1.6 Contenido del Documento..... | 5 |
| CAPÍTULO 2..... | 6 |
| 2. MARCO TEÓRICO | 6 |
| 2.1 Generalidades de la iluminación LED..... | 6 |
| 2.1.1 El LED como fuente de luz | 6 |
| 2.1.2 Aplicaciones para la tecnología LED | 7 |
| 2.1.3 Disposición de LED en la lámpara | 10 |
| 2.1.4 Regulación de intensidad luminosa del LED | 11 |
| 2.2 Alumbrado con luminarias LED | 12 |
| 2.2.1 Sistemas de alumbrado LED | 12 |
| 2.2.2 Tipos de luminarias LED | 13 |
| 2.2.3 Alimentación y regulación de alumbrado LED | 15 |
| 2.3 Métodos de regulación de alumbrado público | 17 |

| | | |
|--|---|----|
| 2.3.1 | Método de control de fase | 17 |
| 2.3.2 | Protocolo de regulación DMX-512 | 18 |
| 2.3.3 | Protocolo de regulación 1 a 10V | 18 |
| 2.3.4 | Protocolo de regulación DALI | 19 |
| 2.4 | Iluminación vial | 19 |
| 2.4.1 | Reducción de niveles de iluminación | 20 |
| 2.4.2 | Niveles mínimos de iluminación admisibles | 20 |
| 2.5 | Gestión de alumbrado exterior | 21 |
| 2.5.1 | Fundamentos básicos | 21 |
| 2.5.2 | Sensores | 22 |
| 2.5.3 | Sistemas de control | 23 |
| CAPÍTULO 3..... | | 26 |
| 3. DISEÑO DEL PROTOTIPO DE SISTEMA DE TELEGESTIÓN REMOTO DE ALUMBRADO PÚBLICO..... | | 26 |
| 3.1 | Descripción General del Sistema de Telegestión Remoto de Alumbrado Público..... | 26 |
| 3.2 | Diseño de la Etapa de Alimentación Eléctrica | 28 |
| 3.3 | Diseño de la Etapa de Control | 30 |
| 3.3.1 | Preparación de la placa Arduino | 31 |
| 3.3.2 | Sistema de Comunicación Local | 38 |
| 3.3.3 | Sistema de Comunicación Global | 47 |
| 3.3.4 | Etapa de Monitoreo | 49 |
| 3.3.5 | Lógica de funcionamiento del prototipo controlador de nodo | 50 |
| 3.3.6 | Prototipos de experimentación del dispositivo controlador de nodo ... | 51 |
| 3.4 | Propuesta de control de iluminación | 55 |
| 3.4.1 | Propuesta 1 – Centro de la ciudad de Ibarra | 56 |
| 3.4.2 | Propuesta 2 – Av. Camilo Ponce y Galo Plaza (Parque Ciudad Blanca) | |
| | 59 | |
| CAPÍTULO 4..... | | 63 |
| 4. EFICIENCIA ENERGÉTICA CON EL IMPLEMENTO DEL SISTEMA DE TELEGESTIÓN PROPUESTO..... | | 63 |
| 4.1 | Análisis de consumo energético de alumbrado público con la situación actual | 63 |
| 4.1.1 | Cálculo de las horas de operación | 63 |
| 4.1.2 | Cálculo de la carga instalada | 64 |
| 4.1.3 | Cálculo de consumo de energía | 65 |

| | | |
|------------|---|----|
| 4.1.4 | Medición del consumo de energía de las luminarias de prueba sin la aplicación del prototipo | 66 |
| 4.2 | Análisis de consumo energético de alumbrado público con la implementación del prototipo de sistema de telegestión | 67 |
| 4.2.1 | Cálculo de la eficiencia energética para la propuesta 1 | 71 |
| 4.2.2 | Medición del consumo de energía de la luminaria de prueba con la aplicación del prototipo | 72 |
| 4.2.3 | Cálculo de la eficiencia energética para la propuesta 2 | 73 |
| CAPÍTULO 5 | | 75 |
| 5. | EVALUACIÓN TÉCNICO-ECONÓMICA DEL SISTEMA DE TELEGESTIÓN DESARROLLADO | 75 |
| 5.1 | Mantenimiento preventivo y correctivo de luminarias de alumbrado público 75 | |
| 5.1.1 | Mantenimiento preventivo | 75 |
| 5.1.2 | Mantenimiento correctivo | 76 |
| 5.2 | Análisis económico de la implementación de sistemas de telegestión comerciales | 77 |
| 5.2.1 | Propuesta de Telegestión Owlet Nightshift del Grupo Schröder | 77 |
| 5.2.2 | Propuesta de Telegestión de Philips | 79 |
| 5.3 | Análisis económico de la implementación del prototipo desarrollado | 81 |
| 5.3.1 | Costos de materiales del equipo controlador de nodo | 81 |
| 5.3.2 | Costos de materiales para el dispositivo controlador de Nodo del prototipo desarrollado | 81 |
| 5.3.3 | Costos de materiales del sistema de telegestión para la propuesta 1 con el prototipo desarrollado | 83 |
| 5.3.4 | Costos de materiales del sistema de telegestión para la propuesta 2 con el prototipo desarrollado | 84 |
| 5.4 | Análisis económico de energía..... | 85 |
| 5.4.1 | Costos de energía con el sistema actual de alumbrado público | 85 |
| 5.4.2 | Costos de energía con la propuesta 1 | 86 |
| 5.4.3 | Costos de energía con la propuesta 2 | 87 |
| 5.5 | Retorno de inversión de la implementación de sistemas de telegestiones 88 | |
| 5.6 | Cálculo de VAN y TIR | 90 |
| CAPÍTULO 6 | | 94 |
| 6. | ANÁLISIS DE RESULTADOS | 94 |
| CAPÍTULO 7 | | 98 |

| | |
|---|-----|
| 7. CONCLUSIONES Y RECOMENDACIONES | 98 |
| 8. REFERENCIAS BIBLIOGRÁFICAS..... | 101 |
| 9. ANEXOS | 103 |
| ANEXO I | 103 |
| ANEXO II | 108 |
| ANEXO III | 113 |
| ANEXO IV | 121 |
| ANEXO V | 187 |
| ANEXO VI | 196 |
| ANEXO VII | 199 |
| ANEXO VIII | 200 |
| ANEXO IX | 203 |
| ANEXO X | 206 |
| ANEXO XI | 208 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 2.1. Iluminación ornamental del parque Ciudad Blanca – Ibarra [4]..... | 8 |
| Figura 2.2. Iluminación arquitectónica del Santuario Señor del Amor – Caranqui [Autoría Propia]... | 8 |
| Figura 2.3. Iluminación de Vitriñas [5] | 9 |
| Figura 2.4. Iluminación Vial Av. Camilo Ponce – Ibarra [Autoría Propia]..... | 9 |
| Figura 2.5. Módulo LED COB [3]..... | 10 |
| Figura 2.6. Módulo LED SMD [3]..... | 10 |
| Figura 2.7 Regulación analógica para varios niveles de amplitud de corriente [Autoría Propia] | 11 |
| Figura 2.8 Modulación por Ancho de Pulso (PWM) [Autoría Propia]..... | 12 |
| Figura 2.9. Luminaria LED de iluminación interior de GE [7]..... | 13 |
| Figura 2.10. Luminaria LED de alumbrado exterior Voltana [8]..... | 13 |
| Figura 2.11. Luminaria LED re-lamping a partir de una luminaria de Vapor de Sodio [9]..... | 14 |
| Figura 2.12. Luminaria LED retrofit tipo farola [10]..... | 15 |
| Figura 2.13 Vista externa de la luminaria Philips para experimentación [Autoría Propia] | 16 |
| Figura 2.14. Vista interna de la luminaria Philips para experimentación [Autoría Propia] | 16 |
| Figura 2.15. Driver de luminaria LED (Xitanium de Philips) [12]..... | 17 |
| Figura 3.1. Descripción general del Sistema de Telegestión de Alumbrado Público [Autoría Propia] | 26 |
| Figura 3.2. Mini Panel Solar [Autoría Propia] | 28 |
| Figura 3.3. Módulo de carga de batería Step-Down DC-DC MPPT [Autoría Propia]..... | 28 |
| Figura 3.4. Batería recargable 12V [Autoría Propia]..... | 29 |
| Figura 3.5. Módulo de carga de batería tipo lipo [Autoría Propia] | 29 |
| Figura 3.6. Batería recargable tipo lipo 11.1V [Autoría Propia] | 29 |
| Figura 3.7. Módulo regulador de voltaje Step-Down DC-DC [Autoría Propia] | 30 |
| Figura 3.8. Arduino Uno R3 [Autoría Propia]..... | 31 |
| Figura 3.9. Software de programación de Arduino [Autoría Propia] | 32 |
| Figura 3.10. Sensor de corriente [Autoría Propia] | 33 |
| Figura 3.11. Módulo Mosfet [Autoría Propia]..... | 34 |
| Figura 3.12. Módulo de control de potencia (Mosfet IRF840) [Autoría Propia]..... | 35 |
| Figura 3.13. Simulación en Proteus del circuito de control y potencia del Mosfet IRF840 [Autoría Propia]..... | 35 |
| Figura 3.14. Esquema de conexiones del circuito diseñado [Autoría Propia]..... | 36 |
| Figura 3.15. Circuito para impresión en placa PCB [Autoría Propia]..... | 36 |
| Figura 3.16. Módulo Relé [Autoría Propia] | 37 |
| Figura 3.17. Componentes de una red ZigBee local tipo Mesh [16]..... | 39 |
| Figura 3.18. Módulo Xbee serie 2 [Autoría Propia]..... | 40 |
| Figura 3.19. Comparación de Tecnologías Inalámbricas [17] | 40 |
| Figura 3.20. Módulo Xbee Explorer mini USB [Autoría Propia] | 41 |
| Figura 3.21. Software de programación de Xbee [Autoría Propia] | 42 |
| Figura 3.22. Shield Xbee [Autoría Propia] | 47 |
| Figura 3.23. Módulo GSM/GPRS SIM900 para Arduino [Autoría Propia]..... | 48 |
| Figura 3.24. Interfaz de monitoreo del prototipo desarrollada en Visual Studio 2017. | 50 |
| Figura 3.25. Vista real del primer prototipo controlador de nodo para la experimentación [Autoría Propia]..... | 51 |
| Figura 3.26. Vista real del segundo prototipo controlador de nodo [Autoría Propia] | 52 |
| Figura 3.27. Vista real del tercer prototipo controlador de nodo (Versión Final) [Autoría Propia].... | 53 |
| Figura 3.28. Diagrama de bloques del funcionamiento del dispositivo controlador de nodo [Autoría Propia]..... | 53 |
| Figura 3.29. Esquema de conexiones del prototipo controlador de nodo [Autoría Propia] | 54 |
| Figura 3.30. Pruebas de laboratorio de las luminarias provistas por la empresa para el desarrollo del proyecto en el Centro de Control [Autoría Propia] | 54 |
| Figura 3.31. Ubicación de instalación de luminarias tipo LED en el centro de Ibarra [Autoría Propia] | 56 |
| Figura 3.32. Calle José Joaquín Olmedo (Centro de Ibarra) a) 19:30 y b) 00:30 [Autoría Propia] 57 | |

| | |
|---|----|
| Figura 3.33. Curva de intensidad horaria promedio de usuarios viales del centro de Ibarra [Autoría Propia]..... | 58 |
| Figura 3.34. Perfil de niveles de iluminación propuesto para el centro de Ibarra (medido en porcentaje) [Autoría Propia]..... | 58 |
| Figura 3.35. Ubicación de instalación de luminarias tipo LED en las calles adyacentes al parque Ciudad Blanca [Autoría Propia] | 59 |
| Figura 3.36. Av. Camilo Ponce (Parque Ciudad Blanca) 01:15 [Autoría Propia] | 60 |
| Figura 3.37. Av. Camilo Ponce (Parque Ciudad Blanca) a) 19:45 y b) 00:45 [Autoría Propia] | 61 |
| Figura 3.38. Curva de intensidad horaria promedio de usuarios viales de la Av. Camilo Ponce y Galo Plaza Lasso [Autoría Propia] | 62 |
| Figura 3.39. Perfil de niveles de iluminación propuesto para la Av. Camilo Ponce y Galo Plaza Lasso (medido en porcentaje) [Autoría Propia] | 62 |
| Figura 4.1. Driver LG Innotek LLP 110W 1A 68-110V - Luminaria Voltana 4 [Autoría Propia]..... | 65 |
| Figura 4.2 Medidas obtenidas Antes y Después con el sistema actual [Autoría Propia] | 66 |
| Figura 4.3. Gráfica de Potencia vs. Iluminación (Driver XITANIUM) [Autoría Propia] | 69 |
| Figura 4.4. Gráfica de Potencia vs. Iluminación (Driver LG Innotek) [Autoría Propia] | 70 |
| Figura 4.5 Medidas obtenidas Antes y Después con la aplicación del prototipo [Autoría Propia] ... | 73 |
| Figura 5.1. Esquema del servicio Owllet Nightshift de Schröder [20]..... | 77 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 2.1. Parámetros fotométricos de evaluación según el tipo de vía [14] | 20 |
| Tabla 3.1. Diferencias funcionales entre ZigBee y otras tecnologías [17] | 41 |
| Tabla 3.2. Parámetros de configuración generales. | 45 |
| Tabla 3.3. Parámetros de configuración del módulo Coordinador..... | 45 |
| Tabla 3.4. Parámetros de configuración de los módulos Ruteadores. | 46 |
| Tabla 3.5. Intensidad horaria promedio de usuarios viales del centro de Ibarra | 57 |
| Tabla 3.6. Intensidad horaria promedio de usuarios viales de los alrededores del parque Ciudad Blanca | 60 |
| Tabla 4.1. Potencia de salida del driver XITANIUM para cada porcentaje de iluminación | 69 |
| Tabla 4.2. Potencia de salida del driver LG Innotek para cada porcentaje de iluminación..... | 70 |
| Tabla 4.3. Datos de cada nivel para la Calle Sánchez y Cifuentes | 71 |
| Tabla 4.4. Datos de cada nivel para la Avenida Camilo Ponce | 73 |
| Tabla 4.5. Resumen de Cálculos Eléctricos del Sistema de Telegestión de Alumbrado Público | 74 |
| Tabla 5.1. Comparación de vida útil entre lámparas de Vapor de Sodio y LED [19]..... | 76 |
| Tabla 5.2. Listado de equipos de la propuesta de Schröder para 72 luminarias | 78 |
| Tabla 5.3. Precios de materiales e instalación para la propuesta de Schröder para 72 luminarias | 78 |
| Tabla 5.4. Listado de equipos de la propuesta de Schröder para 230 luminarias | 78 |
| Tabla 5.5. Precios de materiales e instalación para la propuesta de Schröder para 230 luminarias | 79 |
| Tabla 5.6. Listado de equipos de la propuesta de Philips para 72 luminarias | 79 |
| Tabla 5.7. Precios de materiales e instalación para la propuesta de Philips para 72 luminarias | 80 |
| Tabla 5.8. Listado de equipos de la propuesta de Philips para 230 luminarias | 80 |
| Tabla 5.9. Precios de materiales e instalación para la propuesta de Philips para 230 luminarias .. | 80 |
| Tabla 5.10. Precios unitarios de los materiales en Ecuador..... | 82 |
| Tabla 5.11. Precios unitarios de los materiales importados desde China | 82 |
| Tabla 5.12. Listado de equipos para la propuesta 1 con el prototipo de este proyecto | 83 |
| Tabla 5.13. Precios de materiales e instalación para la propuesta 1 con el prototipo de este proyecto | 84 |
| Tabla 5.14. Listado de equipos para la propuesta 1 con el prototipo de este proyecto | 84 |
| Tabla 5.15. Precios de materiales e instalación para la propuesta 1 con el prototipo de este proyecto | 85 |
| Tabla 5.16. Resumen del Análisis Económico del Sistema de Telegestión de Alumbrado Público | 87 |
| Tabla 5.17. Costo de inversión de cada alternativa..... | 88 |
| Tabla 5.18. Costo de Mantenimiento y Operación de alumbrado público mensual..... | 88 |
| Tabla 5.19. Ahorro energético mensual con sistemas de telegestión | 88 |
| Tabla 5.20. Recuperación de la inversión de cada alternativa para la propuesta 1 | 89 |
| Tabla 5.21. Recuperación de la inversión de cada alternativa para la propuesta 2 | 90 |
| Tabla 5.22. Flujo de Fondos para el cálculo de TIR y VAN para decisión de inversión de la Propuesta 1..... | 92 |
| Tabla 5.23. Cálculo de VAN y TIR para la Propuesta 1 | 92 |
| Tabla 5.24. Flujo de Fondos para el cálculo de TIR y VAN para decisión de inversión de la Propuesta 2..... | 93 |
| Tabla 5.25. Cálculo de VAN y TIR para la Propuesta 2 | 93 |

ÍNDICE DE ECUACIONES

| | |
|---|----|
| Ecuación 3.1. Cálculo de intensidad de área promedio [18] | 55 |
| Ecuación 4.1. Cálculo de horas de operación mensual..... | 63 |
| Ecuación 4.2. Horas de operación mensual..... | 63 |
| Ecuación 4.3. Cálculo de potencia instalada..... | 64 |
| Ecuación 4.4. Potencia instalada para 72 luminarias LED PHILIPS | 64 |
| Ecuación 4.5. Potencia instalada para 230 luminarias LED VOLTANA 4 | 64 |
| Ecuación 4.6. Cálculo de energía consumida mensualmente con el sistema actual..... | 65 |
| Ecuación 4.7. Energía consumida mensualmente con el sistema actual para la propuesta 1 | 65 |
| Ecuación 4.8. Energía consumida mensualmente con el sistema actual para la propuesta 2 | 66 |
| Ecuación 4.9. Cálculo de pérdidas de potencia del LED driver | 67 |
| Ecuación 4.10. Potencia constante del LED driver XITANIUM..... | 68 |
| Ecuación 4.11. Expresión que relaciona la potencia de salida del driver XITANIUM con el porcentaje de intensidad luminosa | 68 |
| Ecuación 4.12. Expresión para calcular la ecuación de una recta conociendo dos puntos..... | 68 |
| Ecuación 4.13. Potencia constante del LED driver XITANIUM..... | 69 |
| Ecuación 4.14. Expresión que relaciona la potencia de salida del driver LG Innotek con el porcentaje de intensidad luminosa | 70 |
| Ecuación 4.15. Cálculo de energía consumida mensualmente con regulación de iluminación..... | 71 |
| Ecuación 4.16. Cálculo del ahorro energético con la propuesta 1..... | 72 |
| Ecuación 4.17. Porcentaje de ahorro energético efectivo con la propuesta 1 | 72 |
| Ecuación 4.18. Cálculo del ahorro energético con la propuesta 2..... | 74 |
| Ecuación 4.19. Porcentaje de ahorro energético efectivo con la propuesta 2 | 74 |
| Ecuación 5.1. Cálculo del ahorro económico importando de China | 83 |
| Ecuación 5.2. Porcentaje de ahorro económico importando de China..... | 83 |
| Ecuación 5.3. Costo de la energía en iluminación del centro de la ciudad bajo la situación actual | 85 |
| Ecuación 5.4. Costo de la energía en iluminación de la Av. Camilo Ponce y Galo Plaza bajo la situación actual..... | 86 |
| Ecuación 5.5. Costo de la energía para la propuesta 1..... | 86 |
| Ecuación 5.6. Cálculo del ahorro económico para la propuesta 1..... | 86 |
| Ecuación 5.7. Porcentaje de ahorro económico con la propuesta 1..... | 86 |
| Ecuación 5.8. Costo de la energía para la propuesta 2..... | 87 |
| Ecuación 5.9. Cálculo del ahorro económico para la propuesta 2..... | 87 |
| Ecuación 5.10. Porcentaje de ahorro económico con la propuesta 2..... | 87 |
| Ecuación 5.11. Expresión para calcular el VAN | 91 |
| Ecuación 5.12. Expresión para calcular el TIR..... | 91 |

RESUMEN

El presente proyecto de titulación está dirigido a mejorar la situación actual de iluminación LED en alumbrado público de la ciudad de Ibarra aprovechando las características de esta tecnología en las luminarias de EMELNORTE S.A. para maximizar la eficiencia energética y obtener ahorros económicos en el consumo de energía de alumbrado público y mantenimiento del mismo, para lograr esto, se ha visto como mejor alternativa el desarrollo de un prototipo de sistema de telegestión de alumbrado público, además proponer alternativas de instalación en las zonas donde la Empresa Eléctrica Regional del Norte tiene instaladas luminarias de este tipo, para luego poder determinar mediante un análisis técnico y económico la viabilidad del prototipo desarrollado frente a sistemas de telegestión de alumbrado público disponibles en el mercado.

PALABRAS CLAVE: prototipo, telegestión, eficiencia, alumbrado, EMELNORTE, LED.

ABSTRACT

The present titling project is aimed at improving the current lighting situation LED lighting in the city of Ibarra taking advantage of the characteristics of this technology in the luminaires of EMELNORTE S.A. to maximize energy efficiency and get economic savings in the consumption of public lighting energy and maintenance, to achieve this, it development has been seen as the best alternative of a prototype telemanagement system of public lighting, also propose installation alternatives in the areas where the Northern Regional Electric Company has installed luminaires of this type, to later be able to determine by means of an analysis technical and economic feasibility of the developed prototype versus telemanagement systems of public lighting available in the market.

KEYWORDS: prototype, remote management, efficiency, lighting, EMELNORTE, LED.

CAPÍTULO 1.

1. INTRODUCCIÓN

1.1 Situación actual del alumbrado público de EMELNORTE

La empresa eléctrica EMELNORTE brinda servicio de energía eléctrica para el norte del país, con pequeñas fuentes de generación, transmisión y distribución, además del servicio de alumbrado público dentro de sus competencias en vías urbanas y rurales, espacios destinados al tránsito peatonal, plazas, parques, complejos deportivos, entre otros; por lo que debe brindar una iluminación acorde a las necesidades de la ciudadanía, y en función de la tecnología actual disponible, de acuerdo con las normas vigentes del país.

En la mayoría de las vías de tránsito vehicular, las luminarias de alumbrado público de una fase y neutro disponen de un conductor piloto para su alimentación, mientras que las luminarias de doble fase no cuentan con una infraestructura de uso exclusivo para iluminación, sino que se encuentran siendo alimentadas desde el conductor piloto y la red de distribución eléctrica destinadas al consumo eléctrico de la ciudadanía.

El centro de control dispone del sistema ADMS funcional para generación, transmisión y distribución, dejando desatendido este sector.

1.2 Justificación

Durante los últimos años, las lámparas de alumbrado público clásicas que fueron instaladas muchos años atrás dentro de las competencias de la empresa eléctrica EMELNORTE S.A. se encuentran siendo reemplazadas por lámparas tipo LED para obtener un considerable ahorro de energía eléctrica en iluminación pública, estas nuevas lámparas de marcas como Philips y Schröder disponen de balastos inteligentes con características dimerizables para controlar su intensidad luminosa, un avance tecnológico que las grandes empresas de iluminación en el mercado han venido perfeccionando en la última década, con el fin de comercializar no solo productos físicos, sino también ofrecer a su clientela servicios energéticos, sin embargo esta característica en específico aún no está siendo aprovechada por las empresas de distribución de energía en el país.

Por esta razón, en este proyecto se plantea desarrollar un prototipo de sistema de telegestión de alumbrado público que controlará y monitoreará el uso de estas nuevas luminarias con tecnología LED, para incrementar el ahorro energético de la empresa y mejorar la calidad de servicio de iluminación ofrecido a la ciudadanía, su seguridad y bienestar, cabe resaltar que el proyecto en mención contribuirá a la reducción del CO2 emitido en el uso de combustibles fósiles para generación de energía, viéndose un mejor resultado si a futuro se planea iluminar el país con tecnología LED con sus respectivos y adecuados sistemas de telegestión.

Otro beneficio del desarrollo de este sistema de gestión es conseguir un prototipo de acuerdo con las especificaciones y necesidades de una empresa nacional. Una solución de terceros podría tener mayor costo y aun así no alinearse correctamente a los requerimientos de EMELNORTE S.A. Para este propósito, las aplicaciones propias del sistema ADMS pueden ser utilizadas para automatizar esta área.

En el país, los sistemas de comunicación son escasos cuando hablamos de distribución de energía. El sistema de comunicaciones que se pretende desarrollar puede tomarse a futuro como una base para que la telegestión de sistemas eléctricos del país prolifere, sin la necesidad de requerir los servicios millonarios de empresas extranjeras.

Al no adquirir la tecnología de telegestión existentes en el mercado se reducen costos de inversión en el proyecto, que incluyen el software de control, cuyo valor de uso anual es un gasto exagerado e innecesario al disponer de software útil para dicho propósito en el centro de control de EMELNORTE. A futuro, el sistema ADMS puede ser adaptado para administrar sistemas de telegestión.

1.3 Problemática

El alumbrado público supone el mayor concurrente en el dispendio de recursos públicos de una ciudad, entre los años 2000 y 2012 el consumo de energía en alumbrado público fue de aproximadamente el 4,70% de la energía bruta total consumida en el país [1].

A fin de obtener un ahorro energético significativo, para finales del año 2017 EMELNORTE S.A planea continuar con la instalación de luminarias tipo LED, en sustitución de las luminarias clásicas de vapor de sodio. En complemento a esta estrategia de ahorro, su consumo se puede reducir significativamente controlando la intensidad de luminosidad de estas nuevas lámparas tipo LED en horarios adecuados, y sería muy práctico tener la

capacidad para hacerlo de forma remota llevando información como consumo energético, tiempo de encendido, energía ahorrada y reporte de daños de las luminarias desde el Centro de Control, que actualmente administra los sistemas de protección, medición y activación de subestaciones, alimentadores y centrales de generación, además de llevar la atención con el cliente vía telefónica, identificar fallos, organizar y distribuir los equipos de campo para la solución de desperfectos en el suministro eléctrico.

Adicionalmente, las luminarias dañadas representan riesgos de accidentalidad vehicular e inseguridad peatonal causado por los puntos ciegos producidos por falla de las luminarias y por el tiempo de duración de las mismas, durante el año 2016 se produjeron 142 siniestros viales provocados por esta causa, equivalentes al 0,47% de total [2], debido a que el sistema actual de detección de fallas de luminarias es mediante llamadas telefónicas realizadas por los usuarios, que únicamente las reportan cuando los afectan directamente, viéndose muy dificultosa la labor de mantenimiento de las luminarias en mal estado que no son reportadas dentro de la urbe, así como las que se encuentran en zonas rurales o caminos aislados.

De persistir esta situación, los sistemas de alumbrado público seguirán siendo ineficientes y no cumplirán con lo exigido por los entes reguladores para este servicio causando llamados de atención y molestias en los usuarios de la empresa distribuidora, así como inseguridad para los usuarios viales.

En vista de que la problemática en el alumbrado público se debe a la falta de control y monitoreo del mismo, es necesario implementar un sistema de telegestión de alumbrado público a fin de obtener un ahorro energético importante mediante el control de la intensidad luminosa, además el mismo que permitirá el monitoreo del funcionamiento adecuado de las luminarias, para esto, la solución ideal es el diseño e implementación de un prototipo que controle la capacidad de iluminación y proporcione la información necesaria para realizar el mantenimiento preventivo y correctivo de las luminarias tipo LED administrado desde el Centro de Control de Emelnorte.

Por su aplicación, el sistema de telegestión del alumbrado público deberá considerar un sistema de comunicaciones que permita enlazar todos los dispositivos de la red, una red para llevar órdenes desde el sistema de control de la empresa hasta los sistemas de iluminación. La red por implementar tiene varios desafíos técnicos y económicos; tales como: el software requerido, protocolos necesarios, y una adecuada selección de la capa física, etc. El desarrollo de este sistema de comunicaciones, los mecanismos de variación de iluminación y la interconexión de ambos son el motivo de este proyecto de titulación.

1.4 Objetivos

1.4.1 Objetivo General

El objetivo general de este Estudio Técnico es: Desarrollar e implementar un prototipo de sistema de telegestión (software y hardware) para luminarias tipo LED de alumbrado público de la Empresa Eléctrica Regional Norte S.A. para reducir el consumo energético en iluminación.

1.4.2 Objetivos Específicos

Los objetivos específicos de este Estudio Técnico son:

- Investigar la manera de controlar las lamparas tipo LED para propósitos de ahorro energético en alumbrado público.
- Diseñar e instalar el sistema de telegestión de alumbrado público en las lamparas elegidas para la puesta en marcha del prototipo.
- Establecer comunicación entre el sistema de alumbrado público y el centro de control de EMELNORTE con el sistema desarrollado.
- Comparar el ahorro energético que el prototipo ofrece frente al sistema convencional utilizado actualmente.

1.5 Alcance

En base a la información proporcionada por la empresa eléctrica EMELNORTE S.A. sobre las características de funcionamiento y ubicación de las luminarias tipo LED instaladas o a instalar en la ciudad de Ibarra se procederá a diseñar e instalar el dispositivo que se encargará del control y monitoreo en una pequeña muestra de lámparas de alumbrado público, que junto con un equipo de medición y un sistema de comunicaciones adecuado serán parte del prototipo de telegestión de este proyecto.

Con el fin de establecer la efectividad del prototipo de telegestión diseñado se instalará un equipo de medición en otra muestra de igual tamaño sin la adición del dispositivo para poder realizar un estudio comparativo de ahorro energético.

Con los datos obtenidos durante un periodo adecuado de prueba y los precios reales de los equipos utilizados en el desarrollo del prototipo para este proyecto se procederá a realizar un análisis costo – beneficio para proyectar el ahorro energético en base al porcentaje de ahorro obtenido en este proyecto.

1.6 Contenido del Documento

El capítulo 1 contiene los objetivos y alcance del proyecto, una breve introducción la situación actual del alumbrado público y los problemas que éste mantiene.

En el capítulo 2 se profundiza en la teoría de iluminación LED, alumbrado público vial, sistemas de regulación y gestión de alumbrado exterior.

En el capítulo 3 se detalla la elaboración del prototipo de sistema de telegestión de alumbrado público, cada una de sus etapas, su funcionamiento y se propone niveles de iluminación.

En el capítulo 4 se realiza el análisis teórico y experimental del consumo energético de alumbrado público en la situación actual y con el implemento del sistema de telegestión.

En el capítulo 5 se realiza la evaluación técnico – económica del sistema de telegestión desarrollado.

En el capítulo 6 se realiza el análisis de resultados del proyecto de titulación desarrollado.

En el capítulo 7 se muestran las conclusiones y recomendaciones de este proyecto.

CAPÍTULO 2.

2. MARCO TEÓRICO

2.1 Generalidades de la iluminación LED

2.1.1 El LED como fuente de luz

El LED se inventó en 1956, mientras que las primeras aplicaciones en la industria llegaron en 1970, siendo escasas por su baja eficacia luminosa, usándose en la electrónica de entretenimiento, electrodomésticos y automóviles. Pero luego de años de investigaciones se logró incrementar su eficacia.

Un LED, cuyas siglas en inglés significan *Light Emitting Diode* se considera como un dispositivo con ánodo y cátodo que al ser polarizado de forma directa se comporta como semiconductor causando un fenómeno llamado electroluminiscencia, el cual transforma la energía eléctrica en radiación visible (luz monocromática), lo que las hace fuentes de luz de estado sólido, sin filamento o gas inerte como en las tecnologías tradicionales [3].

Según [3]: algunas de las características ventajosas de la tecnología LED comparadas con las fuentes de luz convencionales son:

- Alta eficacia luminosa que mejora a lo largo del tiempo con el avance de esta tecnología.
- Alta eficiencia en colores, ya que al ser monocromáticos se puede obtener una gama muy alta de colores sin la necesidad de filtros de color.
- Funcionamiento altamente fiable, ya que su vida útil se encuentra entre 50.000 y 100.000 horas, que para el uso en alumbrado público representa un promedio de 17 años.
- Permiten ahorro en gastos de mantenimiento, debido a su larga vida útil.
- Debido a que emiten prácticamente luz visible no causan perjuicios de radiación.
- Alto índice de Reproducción Cromática.

- Por su tamaño reducido se pueden utilizar en incontables aplicaciones.
- Alta resistencia a golpes y vibraciones, ideales para trabajar en condiciones mecánicas adversas gracias a sus compactos componentes.
- Bajo consumo de energía gracias a la poca potencia instalada.
- Funcionamiento confiable frente a bajas temperaturas, hasta aproximadamente -30°C.
- Mayor rentabilidad económica con respecto a su bajo consumo y su larga vida útil
- Ofrecen facilidad de diseño para soluciones de iluminación innovadoras, por su pequeño tamaño y su variedad de colores.
- Luminarias fácilmente regulables con métodos de control de fase, 1-10V, DALI, protocolos sin cable, etc.
- Su proceso de encendido es instantáneo al 100% de flujo.

2.1.2 Aplicaciones para la tecnología LED

El mercado de iluminación en la última década ha ido evolucionando, las grandes empresas de esta área ya no se dedican a vender únicamente bienes, sino que ahora también venden servicios de iluminación con sistemas de control específicamente elaborados y adaptados para cada necesidad, por ejemplo: Iluminación arquitectónica de fachadas, iluminación marítima, iluminación de centros de recreación, iluminación industrial, iluminación vial, iluminación exterior, iluminación de vitrinas, iluminación de señalización ornamental, señalización de tránsito, hostelería, anuncios publicitarios, etc. [3].

La iluminación ornamental LED del parque Ciudad Blanca de Ibarra, antes llamado Parque Céntrica Bulevar, que se observa en la **Figura 2.1**, lo hizo acreedor al segundo lugar en el concurso mundial **“Auroralia 2013”**, donde se calificó la reducción del impacto medioambiental y la originalidad.



Figura 2.1. Iluminación ornamental del parque Ciudad Blanca – Ibarra [4]

Para diseñar efectos decorativos con varios colores en fachadas de lugares turísticos se utiliza luminarias LED de alta potencia con niveles altos de flexibilidad, flujo lumínico, resistencia mecánica y durabilidad [3], un ejemplo de ello es la fachada iluminada de la iglesia de Caranqui, que se observa en la **Figura 2.2.**



Figura 2.2. Iluminación arquitectónica del Santuario Señor del Amor – Caranqui [Autoría Propia]

La iluminación LED mantiene la calidad de los productos que se exhiben en vitrinas, como se observa en la **Figura 2.3**.



Figura 2.3. Iluminación de Vitrinas [5]

En estos casos se necesita de pequeños puntos de luz para iluminación de orientación, haciéndose necesario el uso de pequeñas lámparas empotrables, algo que con tecnología LED es extremadamente fácil.

En iluminación vial, como en la **Figura 2.4**, se cumple con los estándares de iluminación vial adecuados, con una fuente de luz óptima y alta eficacia óptica [3].



Figura 2.4. Iluminación Vial Av. Camilo Ponce – Ibarra [Autoría Propia]

2.1.3 Disposición de LED en la lámpara

Se refiere a los métodos de montaje de los módulos LED en la superficie del circuito impreso (PCB), de la placa de la luminaria, siendo estas de dos tipos, según [3]:

- **Tecnología Chip on Board (COB):** Los tres componentes del LED se montan directamente sobre la placa de circuito impreso, en la **Figura 2.5** se observa el montaje de los 3 componentes.



Figura 2.5. Módulo LED COB [3]

- **Tecnología de Montaje en Superficie:** Los tres componentes del LED son pre-montados en cápsulas, para que luego, el paquete obtenido sea soldado a la placa de circuito impreso, como se observa en la **Figura 2.6**.



Figura 2.6. Módulo LED SMD [3]

2.1.4 Regulación de intensidad luminosa del LED

Para poder tener una regulación en los LED se puede fácilmente reducir la corriente eléctrica que atraviesa el LED, para esto existen dos técnicas diferentes.

- **Regulación analógica.**

Para ejecutar la técnica de regulación analógica hay que reducir la amplitud de la intensidad de corriente (I) que fluye hacia el LED, como se observa en la **Figura 2.7**.

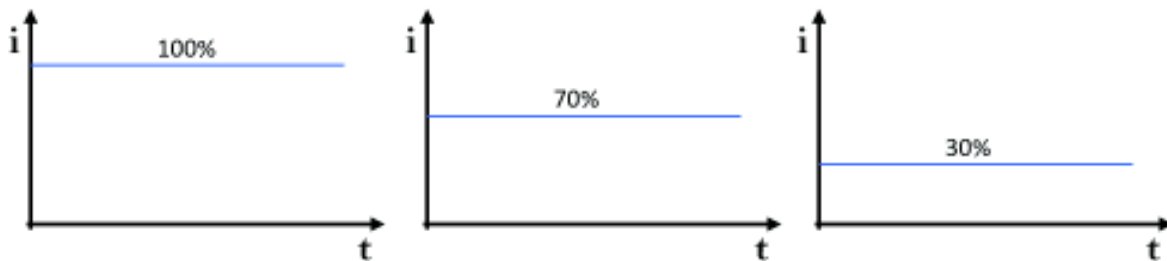


Figura 2.7 Regulación analógica para varios niveles de amplitud de corriente [Autoría Propia]

- **Regulación por PWM (Pulse Width Modulation o en español Modulación por ancho de pulso).**

El objetivo de aplicar esta técnica es variar a conveniencia en una señal periódica lo que se conoce como ciclo de trabajo, que para nuestro caso vendría a ser de tipo DC, convirtiéndola en una señal cuadrada a una determinada frecuencia, para controlar la cantidad de energía que se envía al LED, como se observa en la **Figura 2.8**. Cuanto mayor este ciclo, mayor será la intensidad media efectiva del LED, por lo que se incrementará la iluminación que se percibirá. Gracias a que el ojo humano no percibe el pequeño parpadeo generado por la modulación de la señal, la iluminación parecerá continua.

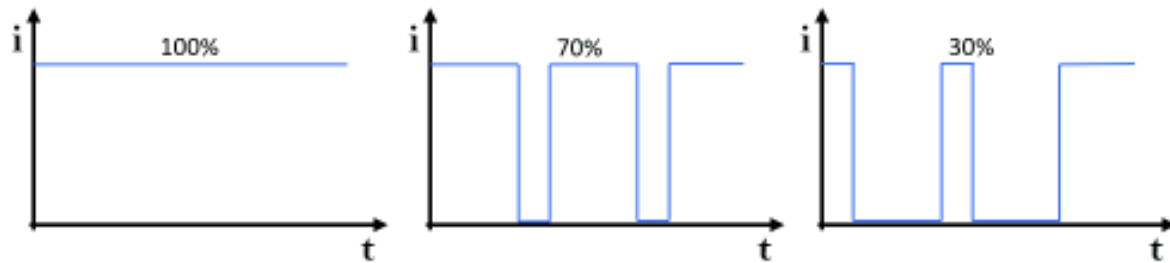


Figura 2.8 Modulación por Ancho de Pulso (PWM) [Autoría Propia]

La modulación por ancho de pulso no tiene limitaciones frente a regulaciones de intensidad bajas, como las tiene la regulación analógica, que podría causar diferencias en la emisión de luz del LED o desviaciones de la cromaticidad. Para evitar estos problemas en intensidades bajas se debe combinar la técnica de regulación analógica con la modulación de ancho de pulso [3].

2.2 Alumbrado con luminarias LED

2.2.1 Sistemas de alumbrado LED

Se define al sistema de alumbrado LED como el conjunto que incorpora la tecnología LED necesaria para obtener una luminaria lista para su uso, que permita la integración de la fuente de luz LED con todos los complementos necesarios para el correcto funcionamiento y protección de los componentes delicados, además de disponer de todos los circuitos auxiliares y una adecuada conexión a la red eléctrica [6].

Este sistema se compone de varios dispositivos:

- Luminarias (Armazón)
- Tiras o módulos LED
- Drivers
- Circuitos Eléctricos y electrónicos

Estos elementos, funcionando armónicamente son los que hacen posible la ventaja frente al resto de tecnologías convencionales de iluminación.

2.2.2 Tipos de luminarias LED

La luminaria LED es el centro de los sistemas de alumbrado, entre ellas es necesario diferenciar los tipos de luminarias, entre las usadas en iluminación de interiores, observada en la **Figura 2.9** y las que se instalan para alumbrado exterior, como en la **Figura 2.10**.



Figura 2.9. Luminaria LED de iluminación interior de GE [7]



Figura 2.10. Luminaria LED de alumbrado exterior Voltana [8]

Según [6]: entre las luminarias de alumbrado exterior se presentan tres posibilidades de instalación:

- **Luminaria de nueva instalación:** donde todos sus componentes son nuevos, especialmente diseñados para ese tipo de luminaria, es decir, que ninguna de sus partes fue utilizada anteriormente en otra luminaria.

Véase **Figura 2.10**.

- **Luminaria modificad (re-lamping):** donde se reemplaza las convencionales lámparas de vapor de sodio, halogenuros metálicos, etc., por lámparas LED de conexión directa, conservando el armazón de la anterior lámpara, como la luminaria de la **Figura 2.11**, es decir, se quita los componentes internos para adaptar la carcasa con componentes de tecnología LED.



Figura 2.11. Luminaria LED re-lamping a partir de una luminaria de Vapor de Sodio [9]

- **Luminaria modificada (retrofit):** tal como se observa en la **Figura 2.12**, en estas luminarias se modifica su diseño original, sean estas modificaciones mecánicas y/o eléctricas obteniendo un cambio en menor o mayor grado de las condiciones iniciales de la luminaria.



Figura 2.12. Luminaria LED retrofit tipo farola [10]

2.2.3 Alimentación y regulación de alumbrado LED

Los LED no se pueden conectar directamente a la red eléctrica, ya que su alimentación necesita ser adaptada adecuadamente a cada tipo de luminaria, para esto se utiliza un controlador, conocido como *driver*, el cual se encarga de reducir y rectificar el voltaje recibido desde la red eléctrica de la manera más eficiente y segura posible, para adecuarlo a las características de funcionamiento de los LED.

El voltaje de la salida del driver se establece en base a la cantidad de leds que serán conectados a él, mientras que la corriente se mantendrá constante para todos los módulos LED, para que la luminosidad del conjunto sea la misma.

La energía que se envía a la salida del driver es inferior a la que ingresa al mismo, debido a las pérdidas que siempre tienen este tipo de dispositivos electrónicos, además de la parte de energía que se disipa en forma de calor. Los fabricantes de drivers se enfocan en reducir estas pérdidas lo más posible, para estar próximos al 100% de eficiencia [11].

El driver se encarga de mantener la disipación de calor para no desperdiciar energía. De no ser por este componente, los LED no funcionarían, es más, se quemarían, ya que para operar correctamente necesitan solo miliamperios de corriente [11].

La parte externa e interna de la luminaria a utilizar en este proyecto se observan en la **Figura 2.13** y en la **Figura 2.14** respectivamente.



Figura 2.13 Vista externa de la luminaria Philips para experimentación [Autoría Propia]

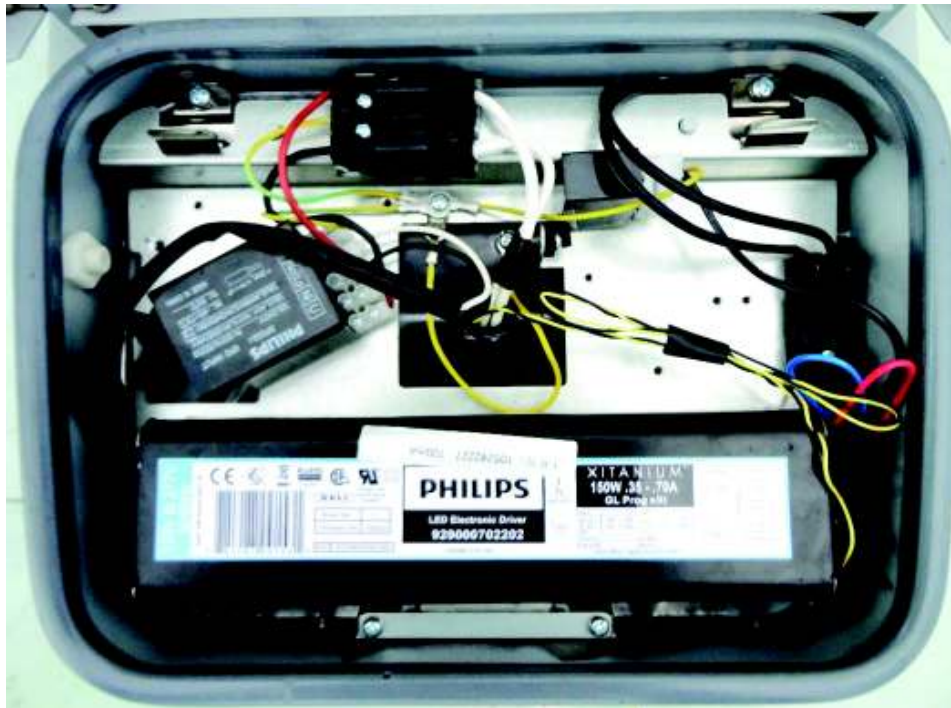


Figura 2.14. Vista interna de la luminaria Philips para experimentación [Autoría Propia]

Las luminarias Philips para experimentación llevan el driver XITANIUM 150 W 150W 0.35 - 0.70A GL Prog sXt, que se observa en la **Figura 2.15**.



Figura 2.15. Driver de luminaria LED (Xitanium de Philips) [12]

2.3 Métodos de regulación de alumbrado público

Para desarrollar un uso más adecuado e inteligente de la cantidad de luz se necesita integrar regulación en la luminosidad de las lámparas obteniendo un ahorro energético considerable y a la vez creando ambientes de alto confort, los sistemas de regulación de luminosidad deben ser adecuados a cada tipo de instalación con sus respectivas necesidades particulares. Estos protocolos o sistemas de regulación indirectamente afectan en el aspecto ecológico de manera positiva, proporcionando un uso de recursos naturales y reduciendo el CO₂ emitido hacia la atmósfera en la generación de la energía [11].

2.3.1 Método de control de fase

En la regulación de iluminación se utiliza el método de control de fase, con un equipo regulador instalando directamente entre la línea de alimentación y la lámpara, cuya función

es controlar entre el 1% y el 100% de la iluminación recortando la señal de voltaje sinusoidal de la red [11].

Para el caso de la tecnología LED han nacido otros métodos o protocolos de control como: DMX 512, 1 - 10V, regulación DALI (Digital Addressable Lighting Interface), etc. [11].

2.3.2 Protocolo de regulación DMX-512

Permite el control de hasta 512 dispositivos con el mismo BUS a velocidades igual o mayores a 25Hz. Este método es utilizado en lámparas de incandescencia, luminarias LED y algunas lámparas de fluorescencia generalmente usadas para espectáculos y conciertos. Por lo que se encuentra estandarizado.

2.3.3 Protocolo de regulación 1 a 10V

Es un método de control analógico que necesita de un cable con dos hilos polarizados para regular por medio de la variación de voltaje. En la misma línea de control se conectan varias luminarias que funcionarán con la misma intensidad luminosa. Este método es utilizado en lámparas de incandescencia, luminarias LED y lámparas de fluorescencia.

Por defecto, los drivers con esta función vienen configurados para alcanzar el nivel de luz máximo permitido por la luminaria, dejando la entrada de este control (conductores +1-10V y -1-10V) en circuito abierto, con un voltaje de 10V, y para conseguir el nivel de luz mínimo permitido se debe poner 1V en la entrada de control o a la vez cortocircuitando la misma.

El rango de regulación de este sistema se encuentra entre el 1% y el 100%, con una curva de regulación casi lineal [11].

Con la regulación del voltaje se obtiene únicamente la regulación de luminosidad, para el encendido y apagado se debe colocar un interruptor en la línea de alimentación.

Para los dispositivos de control con este tipo de regulación, se requiere suministrar una corriente comprendida entre 10uA para un voltaje de 10V y 2mA para un voltaje de 1V [11].

2.3.4 Protocolo de regulación DALI

DALI es la versión mejorada de DSI. Es un protocolo de regulación con un cable de control de dos hilos sin polaridad. Permite el control de hasta 64 luminarias, y al ser bidireccional es posible obtener el estado de la luminaria. Este método es utilizado en luminarias LED y lámparas de fluorescencia.

La interfaz de este sistema es una estructura tipo maestro – esclavo, quiere decir que un dispositivo en la cabecera envía la información hacia los demás equipos que solo ejecutan comandos o responden a solicitudes de información recibidas desde el equipo principal [11].

El rango de regulación de este sistema se encuentra entre el 0,1% y el 100%, con una curva de regulación logarítmica [11].

En este caso no se necesita del uso de interruptores para el encendido y apagado, ya que esto se realiza mediante comandos enviados a través de la línea de control.

2.4 Iluminación vial

En las vías de tránsito vehicular se debe procurar que la iluminación mantenga una circulación segura, para ello el sistema de iluminación debe asegurar una visibilidad completa del trazado de la calzada, sus límites, posibles intersecciones y otras zonas viales importantes.

La disposición y regulación de la iluminación vial depende del uso que se dé a la vía, por ejemplo, para carreteras rurales no es necesario tener una iluminación constante en todos los puntos de la misma, por lo que se acostumbra a tener una mayor distancia de separación entre luminarias que en calles dentro de una ciudad, siempre y cuando se garantice divisar zonas luminosas que provoquen la atención de los usuarios viales [13].

Por esto se recomienda tomar en consideración algunos factores:

- Tipo de vía y zonas especiales.
- Intensidad, velocidad y composición de tráfico.
- Tiempo de utilización de luces de cruce.

2.4.1 Reducción de niveles de iluminación

Se puede optar por la regulación de los niveles de luminosidad o incluso por el apagado de tramos de luminarias a determinados horarios como medidas de ahorro energético, considerando que aproximadamente el 4,70% del consumo de energía del país es en alumbrado público vial [1].

Para obtener una iluminación vial adecuada se debe precisar la intensidad luminosa en función de las necesidades temporales o permanentes de cada vía, lo que significa que dependerá de la hora del día y de las actividades que se desarrollan en la misma.

2.4.2 Niveles mínimos de iluminación admisibles

Tabla 2.1. Parámetros fotométricos de evaluación según el tipo de vía [14]

| Características de la Vía | | Parámetros fotométricos de evaluación (Regulación No. CONELEC 008/11) |
|---------------------------|---|---|
| Tipo de Vía | Condiciones de la Vía | |
| M1 | Una calzada vehicular por lado, con ancho de 12m (3 carriles de 4m cada una), 2 aceras de 1,5m cada una y un parterre central de 2m. Pavimento clase R3, con un coeficiente de luminancia medio Q_0 de 0,07. Factor de mantenimiento recomendado de 0,85 que depende del valor de vida útil especificado. | Luminancia media $L_m \geq 2Cd/m^2$ Uniformidad general $U_0 \geq 40\%$ Uniformidad longitudinal $U_l \geq 70\%$ Incremento de umbral $T_i \leq 10$ Relación de alrededor $SR \geq 0,5$ |
| M2 | Una calzada vehicular por lado, con ancho de 8m (2 carriles de 4m cada una), 2 aceras de 1,5m cada una y un parterre central de 1m. Pavimento clase R3, con un coeficiente de luminancia medio Q_0 de 0,07. Factor de mantenimiento recomendado de 0,85 que depende del valor de vida útil especificado. | Luminancia media $L_m \geq 1,5Cd/m^2$ Uniformidad general $U_0 \geq 40\%$ Uniformidad longitudinal $U_l \geq 60\%$ Incremento de umbral $T_i \leq 10$ Relación de alrededor $SR \geq 0,5$ |
| M3 | Una calzada vehicular, con ancho de 8m (2 carriles de 4m cada una), 2 aceras de 1,5m cada una. Pavimento clase R3, con un coeficiente de luminancia medio Q_0 de 0,07. Factor de mantenimiento recomendado de 0,85 que depende del valor de vida útil especificado. | Luminancia media $L_m \geq 1Cd/m^2$ Uniformidad general $U_0 \geq 40\%$ Uniformidad longitudinal $U_l \geq 60\%$ Incremento de umbral $T_i \leq 15$ Relación de alrededor $SR \geq 0,5$ |
| M4 | Una calzada vehicular, con ancho de 7m (2 carriles de 3,5m cada una), 2 aceras de 1m cada una. Pavimento clase R3, con un coeficiente de luminancia medio Q_0 de 0,07. Factor de mantenimiento recomendado de 0,85 que depende del valor de vida útil especificado. | Luminancia media $L_m \geq 0,75Cd/m^2$ Uniformidad general $U_0 \geq 40\%$ Incremento de umbral $T_i \leq 15$ Relación de alrededor $SR \geq 0,5$ |
| M5 y M6 | Una calzada vehicular, con ancho de 4m, 2 aceras de 1m cada una. Pavimento clase R3, con un coeficiente de luminancia medio Q_0 de 0,07. Factor de mantenimiento recomendado de 0,85 que depende del valor de vida útil especificado. | Luminancia media $L_m \geq 0,5Cd/m^2$ Uniformidad general $U_0 \geq 35\%$ Incremento de umbral $T_i \leq 15$ Relación de alrededor $SR \geq 0,5$ |

Cada tipo de vía es diferente de otra, por lo que para obtener una máxima eficiencia energética se deben realizar mediciones de luminancias e iluminancias medias de cada calzada y la uniformidad. Para después realizar la reducción de los niveles de luminosidad

manteniendo la uniformidad, por lo mismo, se debe reducir el porcentaje en todas las lámparas de la vía por igual.

Estos parámetros han sido normalizados por el CONELEC, ahora ARCONEL en la Regulación N° CONELEC 008/11, para ello se lo ejecuta considerando los parámetros mínimos establecidos, que se muestran resumidos en la **Tabla 2.1** obtenida de [14].

Según [15]: estos parámetros mínimos se pueden modificar de acuerdo con la densidad de tráfico durante los periodos con menor tráfico, con el objetivo de conseguir ahorro de energía, estos cambios deben ser adecuados a la clase de densidad de tráfico a la que se cambie, cumpliéndose con todos los requisitos de uniformidad y deslumbramiento.

2.5 Gestión de alumbrado exterior

Con el avance de la tecnología LED en luminarias de exterior y el uso masivo que éstas han venido teniendo alrededor del mundo en la última década, en los últimos años se han desarrollado funciones de alumbrado inteligente aplicables a las necesidades de los usuarios, consiguiendo instalaciones muy eficientes.

Estas tecnologías conformadas por sensores y sistemas de telegestión funcionando independientemente o en conjunto pueden llegar a conseguir ahorros de energía eléctrica de hasta el 85% [13].

Los sistemas con control inteligente en instalaciones de alumbrado público se pueden conectar a las redes de Smart Cities, y gracias a la flexibilidad de los sistemas se pueden gestionar desde cualquier dispositivo con conectividad inalámbrica, ofreciendo al usuario de la ciudad bienestar, seguridad y sostenibilidad, poniendo los recursos de una ciudad a disposición de la ciudadanía [13].

2.5.1 Fundamentos básicos

Según [13]: para las ciudades que desean convertirse en Smart es necesario adecuar su alumbrado público por estos motivos:

- **Mejorar la gestión de las instalaciones de alumbrado público:** Los problemas que conlleva los tiempos de mantenimiento de las luminarias dañadas representan

pérdidas económicas y de tiempo, molestias para la ciudadanía pueden ser evitados con esta mejora. El disponer de la información adecuada para gestionar de manera inteligente la red de alumbrado permitirán supervisar y valorar las instalaciones de iluminación reduciendo el costo de mano de obra y la frecuencia de interrupción vehicular.

- **Producir un ahorro sustancial en el consumo de energía:** Combinar los sistemas de telegestión con las instalaciones de luminaria LED existentes pueden llevar a evitar el exceso de luz y el consumo innecesario de energía.
- **Facilitar la luz bajo demanda:** Para casos donde la vía no sea muy transitada en horarios específicos, el mantener funcionando las luminarias a todo su potencial es un derroche innecesario de energía, para evitar eso es ideal tener un escenario de regulación a demanda o petición de acuerdo con el lugar y horarios de necesidad. También se pueden programar para escenarios habituales y para ocasiones especiales.
- **Confirmar la fiabilidad de las instalaciones:** Los sistemas de gestión de alumbrado pueden administrar cada una de las luminarias, su operación optimizada y prevención de fallos, para asegurar la eficacia de las instalaciones de iluminación.

2.5.2 Sensores

A parte de gestionar la intensidad luminosa por horarios y cantidad de tránsito, se puede efectuar un sistema más completo haciendo uso de sensores o detectores de localización, movimiento, velocidad, presencia, volumen de tráfico, etc.

Según [13]: para el caso específico de iluminación de alumbrado público se utilizan los siguientes sensores:

Sensores de luz solar

El sistema inteligente de alumbrado público puede ser gestionado automáticamente cuando la luz solar llegue a ser inferior a un límite programado de insuficiencia de luz, para anocheceres, días nublado, etc.

Sensores de detección de movimiento

En lugares de poca actividad vehicular o peatonal, el nivel de las luminarias puede ser reducido a un nivel muy bajo y ser elevado cuando se lo necesite, disminuyendo el consumo de energía en un mayor porcentaje, aunque para considerar el uso de estos sensores se debe tomar en consideración aspectos de seguridad vehicular y peatonal.

Sensores de velocidad y dirección

Estos sensores, en comparación con los sensores de movimiento garantizan la activación adecuada de las luminarias en una extensión mayor para garantizar la seguridad y visualización de la carretera en función de la velocidad de cada vehículo.

2.5.3 Sistemas de control

Según [13]: existen tres tipos de sistemas de control principales:

Stand Alone o Independiente

En este sistema, cada luminaria se comporta de manera independiente ya que disponen de su propia unidad de control. Disponen de celdas fotoeléctricas integradas en la unidad, permitiendo una instalación inmediata, haciendo fácil el reinstalarla en otra luminaria existente. Utiliza sensores de movimiento para detectar la presencia de vehículos y peatones, cada sensor se ajusta para evitar activaciones o intensidades innecesarias.

Presenta estas ventajas:

- Baja inversión y amortización rápida.
- Ahorro energético de hasta el 30% (Estudios realizados por Schröder).
- Fácil instalación.
- Reducción de costos de energía.

Red Autónoma

Este sistema proporciona una gestión de la instalación de alumbrado público de manera autónoma gracias al sistema de comunicación que las interconecta entre sí. Se puede reforzar el sistema de gestión con el uso de sensores de movimiento para proporcionar seguridad y confort a los usuarios viales.

Las ventajas que presenta este sistema son:

- Gestión total de la instalación.
- Rápida amortización.
- Optimización del esquema de iluminación.
- Flexibilidad.
- Ahorro de energía de hasta el 50% (Estudios realizados por Schröder).
- Fácil instalación.
- Fácil programación.
- Reducción de costos de energía.
- Fácil ampliación de la red.
- Fiabilidad de la red.
- Red inalámbrica.

Red Interoperable

Es una combinación de tecnologías orientadas al futuro con una interfaz web amigable al usuario gracias al internet de las cosas. Se la gestiona por medio de internet, de tal manera que sea capaz de supervisar el estado operativo, consumo de energía y posibles fallos desde cualquier lugar del mundo.

Las ventajas de este sistema son:

- Gestión total de la instalación.
- Rápida amortización.

- Flexibilidad y compatibilidad.
- Ahorro de energía incluso de hasta el 85%.
- Tecnología de punta (radiofrecuencia, geolocalización, servidor web, etc.)
- Instalación rápida al ser inalámbrica.
- Costos de energía reducidas.
- Gestión de información, datos y alarmas.
- Fácil integración a terceros.
- Fácil y sencilla ampliación de la red.
- Tecnologías de comunicación inalámbrica como GPRS, wifi, radiofrecuencia, etc.

CAPÍTULO 3.

3. DISEÑO DEL PROTOTIPO DE SISTEMA DE TELEGESTIÓN REMOTO DE ALUMBRADO PÚBLICO

3.1 Descripción General del Sistema de Telegestión Remoto de Alumbrado Público

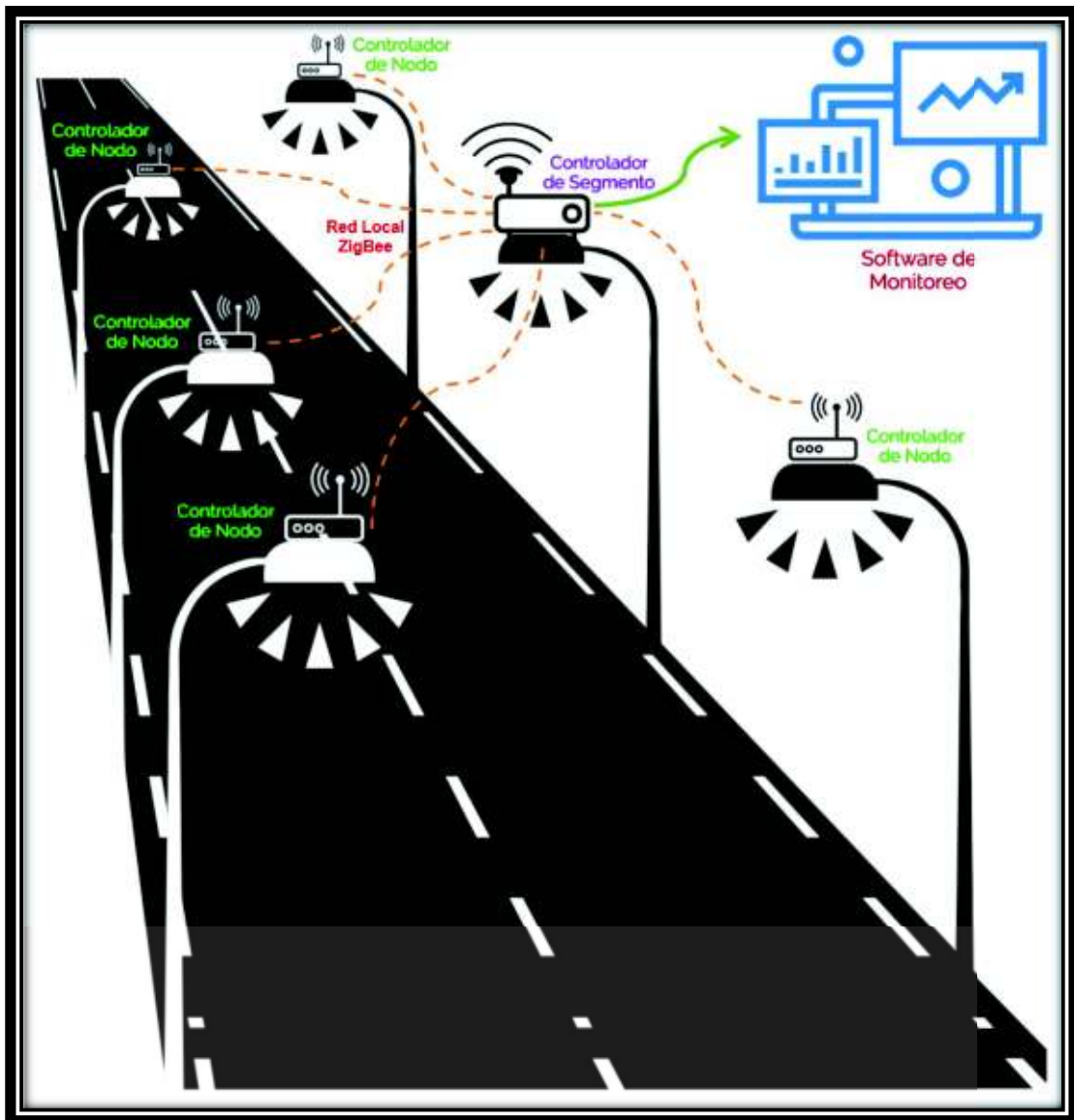


Figura 3.1. Descripción general del Sistema de Telegestión de Alumbrado Público
[Autoría Propia]

Para el sistema de telegestión de alumbrado público, se seleccionarán los dispositivos necesarios y adecuados que cumplan con los objetivos de este proyecto. En la **Figura 3.1** se muestra una idea clara del objetivo del proyecto, donde se puede observar los dispositivos controladores de los nodos terminales, que se comunican exclusivamente con el nodo coordinador a través de comunicación con tecnología ZigBee Mesh, la comunicación de los datos de todos los nodos desde el coordinador a la unidad de monitoreo ubicada en el Centro de Control de EMELNORTE se puede realizar a través de una Red Celular, o de manera más económica con la misma red ZigBee, para su respectivo análisis y acciones necesarias y pertinentes.

Para el sistema se requiere de un dispositivo de control autónomo, una y la más adecuada opción es el uso de un microcontrolador que son el corazón del dispositivo de cada nodo, sensores y actuadores para el control de la luminaria. Viéndose adecuado el uso de la placa Arduino que, al ser una plataforma electrónica de código abierto flexible al diseño de prototipos electrónicos, por su fácil uso de software libre y hardware integrado para un rápido y correcto uso, evitando fallos innecesarios en el conexionado de elementos periféricos.

Para la implementación de la red ZigBee con topología mallada, los dispositivos Xbee de la empresa *Digi International* tienen las características adecuadas con respecto al aprovechamiento mínimo de energía, distancias de transmisión, etc. Esta red estará conectando todos los nodos terminales con el nodo coordinador, los primeros enviando datos del sensor de corriente y recibiendo órdenes de actuación desde este último.

La comunicación entre el nodo coordinador y la unidad de monitoreo el dispositivo a usar es a través de la misma red ZigBee, adaptando y permitiendo mantener una red inalámbrica funcional y activa lista para el acoplamiento de futuros proyectos.

Para el funcionamiento de estos dispositivos se necesita de un sistema de alimentación con la autonomía suficiente para un uso continuo de aproximadamente 12 horas.

Y finalmente, para la unidad de monitoreo remota es adecuado y recomendable el manejo del programa de monitoreo desde el Centro de Control SCADA de la empresa, mediante un programa desarrollado en Visual Studio 2017.

Para el diseño del prototipo se debe considerar algunas etapas, locales y globales, que funcionando en armonía forman el prototipo de sistema de telegestión.

3.2 Diseño de la Etapa de Alimentación Eléctrica

Una propuesta energética eficiente y de bajo costo para la energización de dispositivos electrónicos de poco consumo es el uso de energía solar, de manera que, para el correcto y total funcionamiento del dispositivo de cada nodo se necesita la autonomía eléctrica suficiente para un uso intenso en horario nocturno, por lo que es necesario el uso de una batería recargable con su respectivo sistema de carga.

En este caso, como alimentación primaria utilizaremos un mini panel solar, conectado a una batería recargable a través de un controlador de carga Step-Down MPPT.

Para ello se puede optar por el uso de paneles solares de 1,5W a 12V y de 3W a 6V, con sus respectivas baterías y módulos de carga.

El panel solar de 1,5W a 12V observado en la **Figura 3.2** alimentará la batería de la **Figura 3.4** a través del controlador de carga que se observa en la **Figura 3.3**.



Figura 3.2. Mini Panel Solar [Autoría Propia]

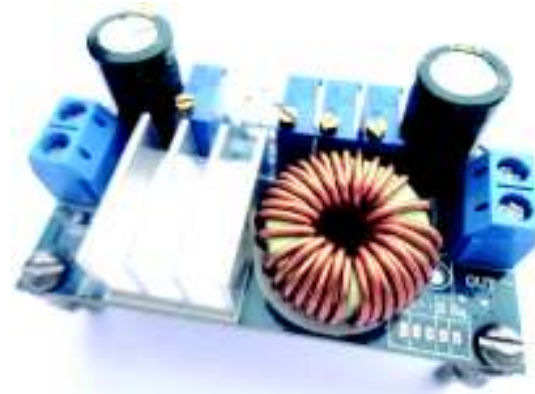


Figura 3.3. Módulo de carga de batería Step-Down DC-DC MPPT [Autoría Propia]



Figura 3.4. Batería recargable 12V [Autoría Propia]

Para el panel solar de 3W a 6V que alimentará la batería de la **Figura 3.6** se necesita el módulo de carga que se observa en la **Figura 3.5**.



Figura 3.5. Módulo de carga de batería tipo lipo [Autoría Propia]



Figura 3.6. Batería recargable tipo lipo 11.1V [Autoría Propia]

Para la alimentación de Arduino se establecerá un voltaje de 10V, que a la vez servirá de alimentación al módulo mosfet variador de voltaje 1 – 10V para el driver de la lámpara, para lo cual se tiene que usar un regulador de voltaje, como el que se muestra en la **Figura 3.7**.



Figura 3.7. Módulo regulador de voltaje Step-Down DC-DC [Autoría Propia]

Los detalles técnicos de los equipos y dispositivos utilizados para la etapa de alimentación del prototipo se encuentran en el **ANEXO VIII**.

3.3 Diseño de la Etapa de Control

El sistema de Arduino está diseñado para la entrada de señales provenientes de sensores que captan cambios en el medio, integrándolos con salidas digitales y analógicas para el control de actuadores de iluminación, motores, etc.

Para los dispositivos controladores de nodo, el microcontrolador ATmega328 de la placa de Arduino observado en la **Figura 3.8** será programado a través del software IDE propio de la marca para funcionar con el requerimiento de la empresa eléctrica, este es el encargado de registrar la corriente que llega a la lámpara para comprobar el funcionamiento de esta y transmitir esta información al nodo coordinador mediante comunicación inalámbrica ZigBee en una topología de malla.

El dispositivo controlador de nodo es alimentado por una de las baterías recargables que se muestran en la **Figura 3.4** y **Figura 3.6**.

Este dispositivo debe ser de bajo costo, fácil configuración y poco consumo energético. El uso de esta tecnología de bajo costo tiene como finalidad el reemplazo fácil y rápido del dispositivo controlador en caso de daño, destrucción o robo.

Para evitar daños naturales y desgaste prematuro, todo el conjunto de elementos que forman el dispositivo será encapsulado en una caja para su protección, por la altura a la que se encontrarán instalados y el difícil acceso a los mismos, el hurto de los dispositivos de nodo es casi imposible.



Figura 3.8. Arduino Uno R3 [Autoría Propia]

3.3.1 Preparación de la placa Arduino

La placa de Arduino tiene la facilidad de ser alimentada de dos maneras, la primera es a través de la conexión con el puerto USB de la computadora, la cual proporciona una alimentación de 5V, para el modo de programación esta será la alimentación utilizada, mientras que para el uso de pruebas del dispositivo será alimentado desde la batería recargable a través de su plug para alimentación externa entre 7 y 12 V.

El entorno de trabajo del IDE de Arduino descargado gratuitamente de la página web www.arduino.cc al momento del desarrollo de este proyecto disponía de la *versión 1.8.5*, que se muestra en la **Figura 3.9**.

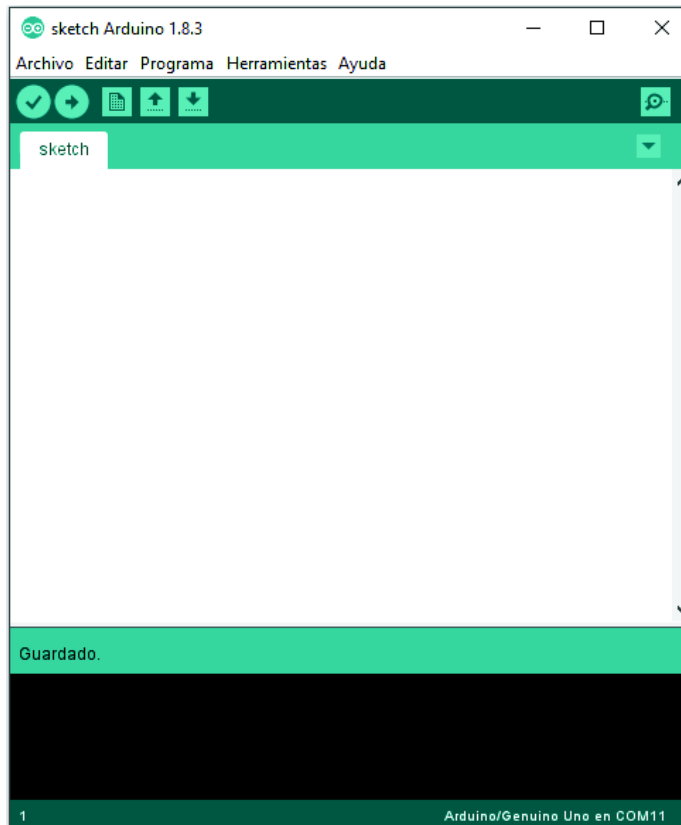


Figura 3.9. Software de programación de Arduino [Autoría Propia]

Descripción del código de programación

La programación del microcontrolador ATmega328 se realizará a través del software propio de Arduino en código de programación C++, el cual se encuentra en el **ANEXO I**.

El código programado se divide en varias secciones, la declaración de constantes y variables, la sección de configuraciones iniciales (*void setup*), el lazo de repetición infinita (*void loop*) y la sección de funciones.

- **Declaración de Constantes y Variables**

Donde se declara las constantes y las variables necesarias que se usarán para la ejecución y desarrollo de las operaciones del programa.

- **Void Setup**

Donde se configura y activa las librerías necesarias para el uso funciones de la placa, las características de sus pines de entrada y salida y las ejecuciones iniciales del programa.

- **Void Loop**

Donde se describe las acciones a tomarse y el orden de ejecución de las funciones del programa, que se repetirá de forma infinita.

- **Funciones**

Donde se declara y programa las funciones que serán llamadas dentro del bucle infinito (*void loop*), el orden donde se encuentren están funciones no altera el resultado de la compilación del programa.

El funcionamiento de los algoritmos de las funciones necesarias para la ejecución del programa se encuentra detallado en los diagramas de bloques del **ANEXO II**.

Sensores y Actuadores

Sensor de Corriente de Efecto Hall

Como periférico de entrada tenemos un sensor de corriente analógico de efecto hall, como el de la **Figura 3.10**, para detectar fallos de encendido de la luminaria, lo que hace el sensor para fines del proyecto, es medir la corriente que atraviesa por el hacia la lámpara, si el valor enviado desde el sensor hacia el Arduino supera un valor umbral significa que los LED de la lámpara se encuentran encendidos, caso contrario se asume que por una u otra razón la lámpara no se encuentra funcionando.

El envío de este dato únicamente se realizará cada vez que se produzca un cambio en la condición del sensor.

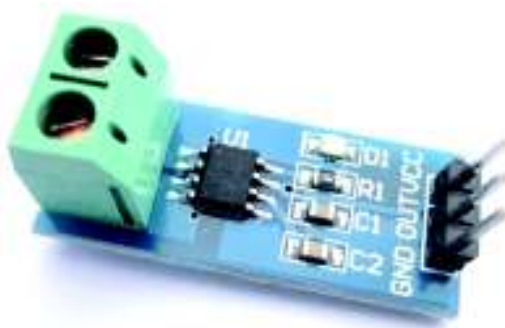


Figura 3.10. Sensor de corriente [Autoría Propia]

La precisión de este sensor no es la adecuada para realizar cálculos, por lo que será utilizada únicamente para determinar como un nivel lógico el encendido y apagado de la luminaria.

Como periféricos de salida tenemos un módulo mosfet y un módulo relé:

Módulo Mosfet

El mosfet, de la **Figura 3.11** será el encargado de conmutar el voltaje de 10V de acuerdo con la señal PWM enviada desde el Arduino para obtener la variación de voltaje entre 0 y 10V adecuada y necesaria para el control del parámetro de intensidad luminosa del driver de la lámpara.

El cambio de este valor dentro del código del programa se realizará únicamente cuando el dato de control de la señal pwm enviado desde la unidad de monitoreo sea diferente al dato actual de dicha señal en la programación de la placa.



Figura 3.11. Módulo Mosfet [Autoría Propia]

Después de realizar las pruebas necesarias en las luminarias proporcionadas para el desarrollo del proyecto no fue posible realizar el control por voltaje de 1-10V, por lo que se procedió a diseñar un circuito de control y potencia aislado para utilizar la salida del driver como fuente controlable para alimentar los LEDS, obteniendo una variación en la intensidad de corriente precisa, el cual se muestra en la **Figura 3.12**.



Figura 3.12. Módulo de control de potencia (Mosfet IRF840) [Autoría Propia]

El circuito posee una parte de control y otra de potencia, los transistores PIC47 aíslan la parte de control de la parte de potencia, que junto con la resistencia R1 de 10ohm proporcionan al Gate del mosfet IRF840 la corriente suficiente para polarizarlo, la resistencia de potencia R3 controla la corriente que se transmite a los LED de la luminaria. El circuito simulado en Proteus se muestra en la **Figura 3.13**, el esquema de conexión de la placa en la **Figura 3.14** y el circuito impreso del PCB en la **Figura 3.15**.

La señal de control es la misma que la utilizada en el caso 1-10V.

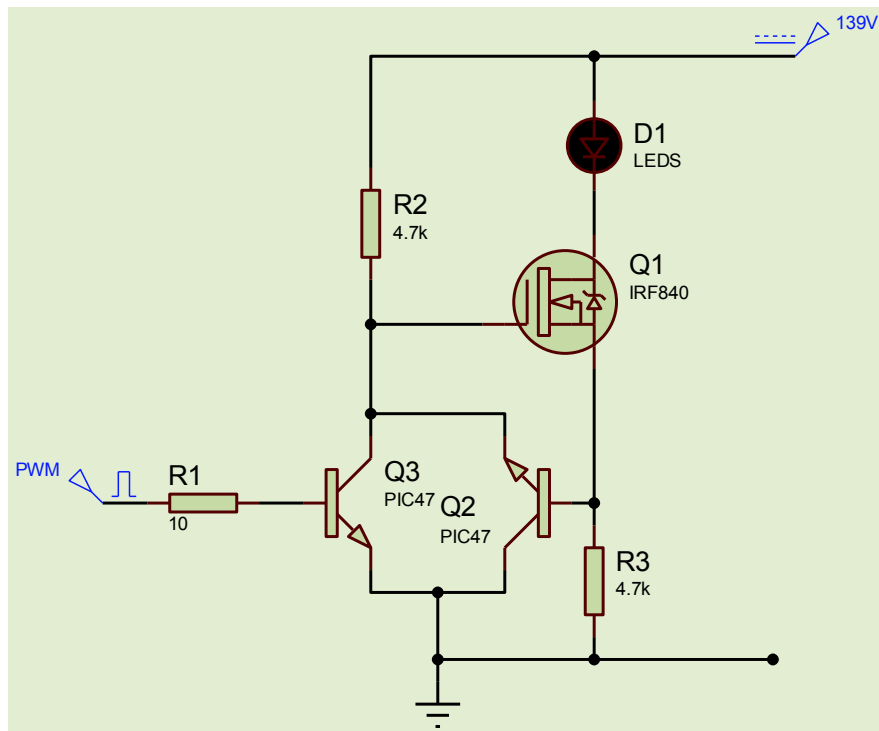


Figura 3.13. Simulación en Proteus del circuito de control y potencia del Mosfet IRF840 [Autoría Propia]

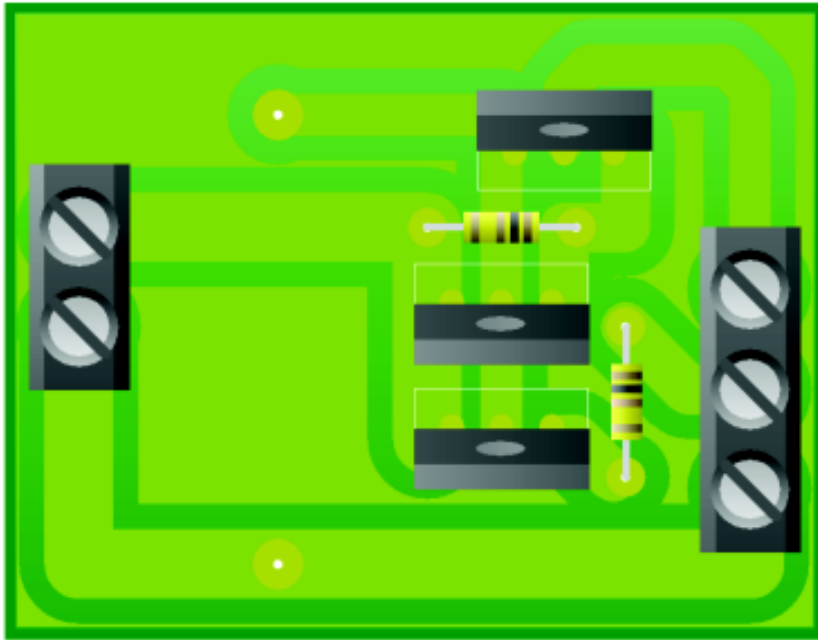


Figura 3.14. Esquema de conexiones del circuito diseñado *[Autoría Propia]*

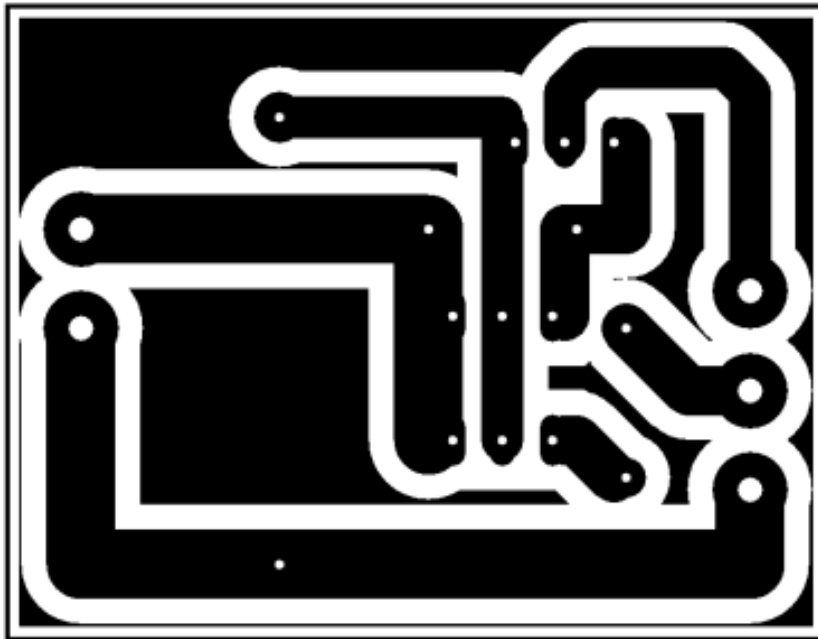


Figura 3.15. Circuito para impresión en placa PCB *[Autoría Propia]*

Para mantener una temperatura adecuada en el circuito de potencia diseñado se utilizará un ventilador pequeño de 12V, además del viento mismo al que se expone el prototipo al estar colocado al aire libre.

Módulo Relé

Debido a que el driver no tiene la característica de apagar la lámpara completamente con la variación del voltaje de control de 0 – 10V, se ve necesario el uso de un interruptor que actúe en la línea de alimentación, para lo que se instalará un módulo relé, como el de la **Figura 3.16** en condición normalmente cerrado.

De igual manera, el cambio en la posición del relé se ejecutará únicamente cuando la posición actual del pin correspondiente sea diferente al valor enviado desde la unidad de monitoreo.

El apagado de la luminaria con el uso de la placa mosfet elaborada es válido para el método de control de fase con el driver como fuente, reemplazando la función establecida del relé, esto conllevaría al funcionamiento permanente del mosfet, teniendo éste que soportar los 139V DC a la salida del driver durante las 24 horas del día, por tal motivo se mantendrá el uso del relé como se estableció al principio.

Todas estas consideraciones de actuación de cada elemento son implementadas para evitar consumos de corriente innecesarios en el dispositivo.



Figura 3.16. Módulo Relé [Autoría Propia]

Los detalles técnicos de los equipos y dispositivos utilizados para la etapa de control del prototipo se encuentran en el **ANEXO IX**.

El sistema de comunicación de este proyecto se divide en dos, el sistema de comunicación local y el sistema de comunicación global.

3.3.2 Sistema de Comunicación Local

La comunicación entre nodos se llevará a cabo con la implementación de una red inalámbrica ZigBee, con el protocolo IEEE 801.15.4; para nuestro caso, esta red estará configurada como una red tipo Mesh. Los dispositivos con esta tecnología son diseñados para consumir poca energía, además, gracias al protocolo estandarizado, se pueden comunicar entre ellos, independientemente del fabricante.

Cuando se configura una red inalámbrica con tipología mallada se consigue auto detectar un nodo o ruta perdida y reconfigurar automáticamente la red, reparando cualquier daño.

Tipos de Nodo

Dentro de esta red mesh (malla) se localizan 3 tipos de nodos, como se muestra en la **Figura 3.17**. Debido a que en la red pueden existir varios ruteadores se crean muchos caminos de comunicación entre los nodos más lejanos, para esto el coordinador y los ruteadores son los que eligen automáticamente el camino más corto y eficiente para transmitir la información, si una ruta falla buscan la mejor opción para reemplazarla.

- **Nodo coordinador:** es el encargado de formar la red y comunicarse con todos sus nodos, también cumple con las funciones de manejar las direcciones, la seguridad y la auto regeneración, entre otras funciones, en cada red existe un solo coordinador.
- **Nodos ruteadores:** se unen a una red formada, reciben y envían sus propios datos, y también puede rutear o enrutar el tráfico desde los nodos más distantes, es decir que se encargan de expandir las distancias de alcance de la red sirviendo como intermediario para permitir la comunicación entre el nodo coordinador y los nodos de dispositivos terminales fuera del alcance del coordinador, si la red no es tan extensa, el uso de ruteadores es opcional, aunque para garantizar un mejor desempeño se recomienda utilizarlos.
- **Nodos de dispositivos terminales:** se unen de igual manera a una red formada previamente por el coordinador, recibe y envía sus propios datos, se comunican

únicamente con los ruteadores y con el coordinador, en ningún caso se comunican entre ellos, por esta razón estos dispositivos pueden consumir menos energía y además pueden trabajar de manera intermitente permaneciendo la mayor parte del tiempo inactivos (dormidos) y solo despertándose por cortos periodos de tiempo para verificar datos de entrada o salida, ejecutar sus funciones y volverse a su estado de inactividad.

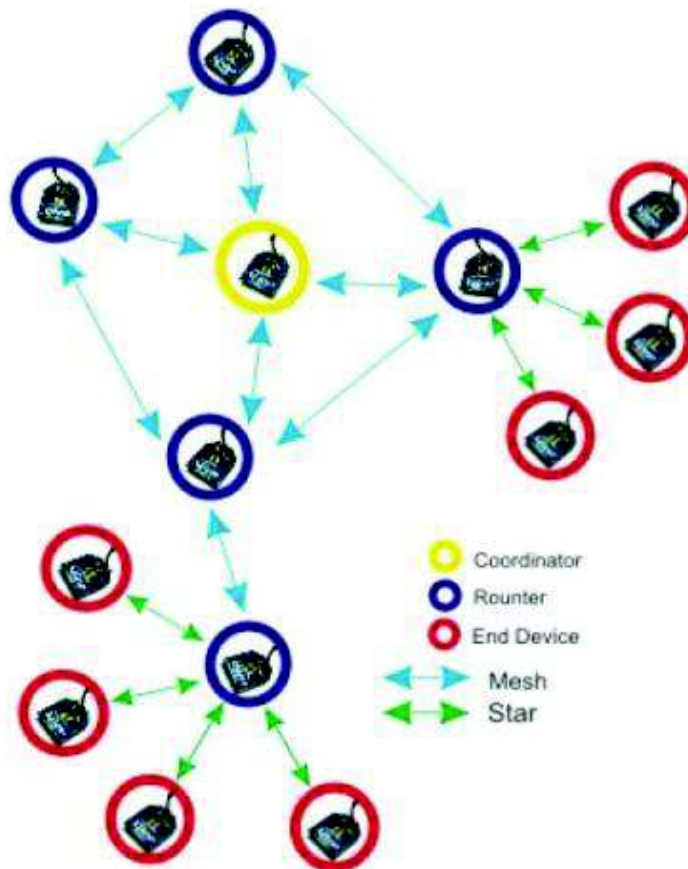


Figura 3.17. Componentes de una red ZigBee local tipo Mesh [16]

Los dispositivos para usar en este proyecto en la formación de la red local serán los módulos Xbee de *Digi Internacional*, que se observa en la **Figura 3.18**, cuyas especificaciones técnicas se detallan en el **ANEXO X**, junto con los demás dispositivos de la etapa de comunicación.



Figura 3.18. Módulo Xbee serie 2 [Autoría Propia]

Las comunicaciones inalámbricas xbee se encuentran en un rango de alcance y velocidad adecuado e ideal para la implementación de redes estáticas, escalables y con muchos dispositivos, con pocos requisitos de ancho de banda, en las que se necesite una duración de batería muy larga, que es nuestro caso. La falta de soporte TCP/IP la hace insuficiente para la comunicación global de este proyecto, por lo que para esa etapa se necesita de otro tipo de sistema de comunicaciones. La comparación entre tecnologías se observa en la **Figura 3.19**.

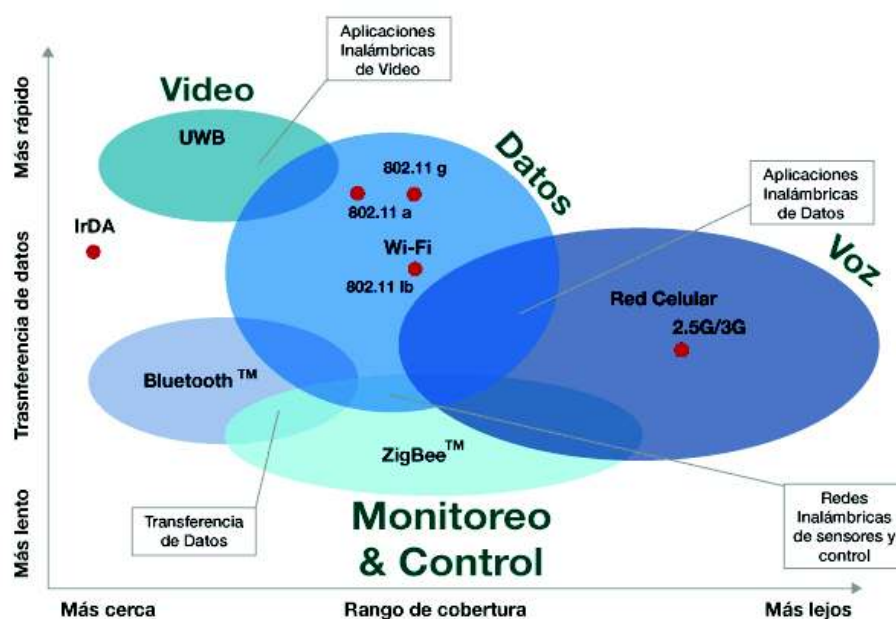


Figura 3.19. Comparación de Tecnologías Inalámbricas [17]

El protocolo ZigBee funciona en microprocesadores de 8, 16 y 32 bits, destinado al uso de Control Inalámbrico mediante el uso de Redes Inalámbricas de Sensores y Control, ofreciendo un bajo consumo de energía a bajo costo, alta confiabilidad y seguridad, con un estándar abierto y mundial, pero con bajas velocidades de transmisión de datos, las diferencias entre ZigBee y otras tecnologías como RFID, Wibree (BLE – Bluetooth Low Energy) y WiMAX se observan en la **Tabla 3.1**. [17].

Tabla 3.1. Diferencias funcionales entre ZigBee y otras tecnologías [17]

| ZigBee | Otras Tecnologías |
|--|--|
| Tiempo de vida: Décadas | Tiempo de vida: 2 años |
| Duración de Batería: Años | Duración de Batería: Días |
| Hardware hasta de 8 bits | Hardware complejo |
| Control simple | Información de alta definición |
| Velocidad de transmisión de datos baja | Velocidad de transmisión de datos alta |

Para poder configurar y programar estos dispositivos de comunicación se requiere de la placa programadora xbee explorer mini USB de la **Figura 3.20**, que sirve de enlace de comunicación entre el computador y el módulo xbee con el uso del programa XCTU, cuya interfaz se observa en la **Figura 3.21**.



Figura 3.20. Módulo Xbee Explorer mini USB [Autoría Propia]

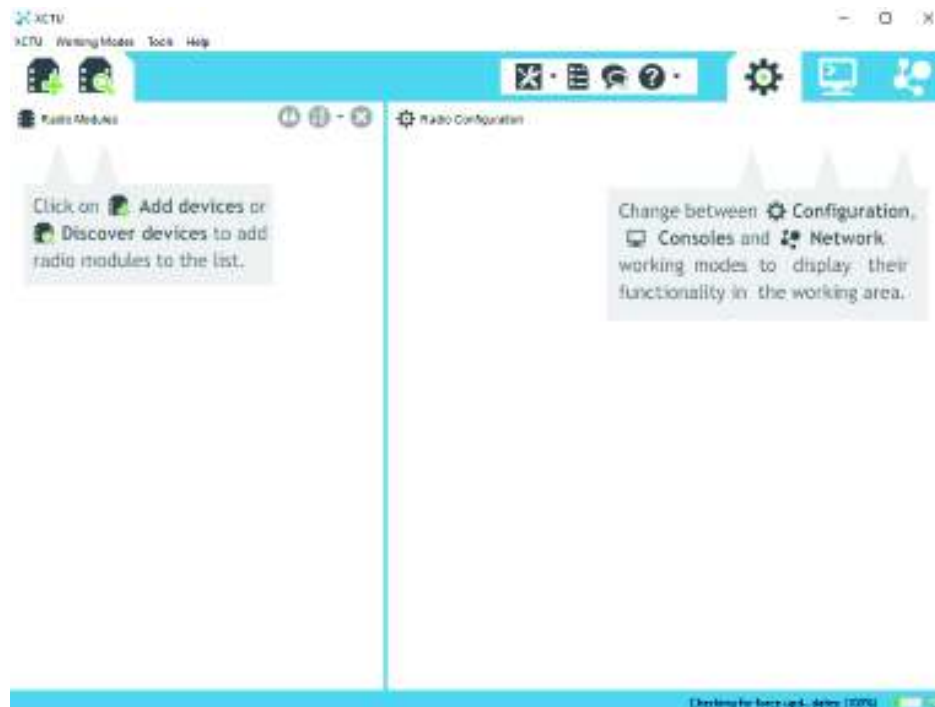


Figura 3.21. Software de programación de Xbee [Autoría Propia]

Como se mencionó antes, los módulos Xbee se pueden configurar en modo Coordinador, Router y como Dispositivo Terminal, para garantizar la confiabilidad de la red local, en nuestra red existirá un único Coordinador y el resto de los nodos serán configurados como Ruteadores, de esta manera, si un nodo falla, la transmisión de datos desde los nodos restantes cambiará su camino saltando el nodo fallado para mantener la comunicación del resto de nodos funcional. Para la implementación del prototipo a gran escala, lo ideal sería tener un ruteador en cada esquina con vista directa de ser posible, para disminuir los problemas de transmisión que producen los edificios.

Configuración de los Módulos Xbee

Para mantener una comunicación inalámbrica permanente entre todos los nodos controladores se requiere configurar los parámetros de los módulos xbee de cierta forma que se garantiza una comunicación broadcast entre todos los equipos de la red local. Para lograr este propósito, los parámetros a configurar son los siguientes:

Direccionamiento

a. Dirección PAN (Personal Area Network)

Este código representa la dirección de la Red y es única por cada red, y al ser éste de 16 bits permite coexistir hasta 65536 redes en la misma zona geográfica.

b. Dirección de cada módulo

Todos los nodos de la red requieren de una dirección para identificarlos, y para eso existen 3 posibilidades:

- **Número de Serie:** Cada módulo xbee posee un código de 64 bits que lo identifica y lo hace único en todo el mundo. (Ejemplo: 0013A40040D7D11D)
- **Dirección:** Cada módulo xbee posee un código de 16 bits asignado de forma dinámica por el Coordinador que lo identifica y lo hace único en la red. (Ejemplo: 0x21d5)
- **Cadena de texto:** Cada módulo xbee puede ser nombrado manualmente por el programador con caracteres de fácil entendimiento para el ser humano que lo identifica y lo hace único en la red. (Ejemplo: "NODO LUMINARIA 4")

c. Canales

El protocolo ZigBee opera a 2.4 GHz, con 16 canales alrededor de esta frecuencia, la selección del canal la realiza el coordinador en el momento de formación de la red, escaneando todos los canales para verificar los canales ocupados y disponibles, eligiendo para su red un canal libre o menos saturado.

Software

De acuerdo con la función que tenga cada módulo en la red, se programa el Firmware adecuado, por lo que existen 3 clases:

- Firmware para Coordinador
- Firmware para Ruteador
- Firmware para Dispositivo Terminal

Modos de Funcionamiento

Existen 3 modos de funcionamiento, según el tipo de intercambio de datos.

- **Modo Transparente:** Forma una conexión punto a punto entre dos nodos, permitiendo que los dos reciban y envíen datos por su puerto serial, es decir que reenvían al aire todo lo que reciben por el puerto serial, físicamente representaría un cable serial.
- **Modo Comandos AT:** Este es el modo de configuración y control del módulo de manera local, en donde la información que ingresa por el puerto serial son comandos y no datos, los cuales no son retransmitidos al aire, sino que es procesada y ejecutada por el propio módulo.
- **Modo Comandos API:** En este modo la información se recibe y envía en forma de paquetes de datos, que contienen metadatos (datos adicionales de protocolo, necesarios para su funcionamiento). Estos paquetes pueden contener datos o comandos AT, los datos se retransmiten a los módulos remotos y los comandos AT pueden ser locales (para el propio módulo) y remotos (destinados a otros módulos de la red), lo que brinda la posibilidad de reprogramar o reconfigurar los módulos sin conectarlos directamente cada uno, sino desde el centro de operación.

A continuación, se detallan los parámetros más importantes a configurar en los módulos Xbee, el detalle de todos los parámetros grabados en el módulo Coordinador y en los módulos Ruteadores se muestran en el **ANEXO III**.

Los parámetros generales de comunicación configurados en todos los módulos Xbee son los que se encuentran en la **Tabla 3.2**.

Tabla 3.2. Parámetros de configuración generales.

| Set port Parameters | |
|----------------------------|------|
| Baud Rate | 9600 |
| Data Bits | 8 |
| Parity | None |
| Stop Bits | 1 |
| Flow Control | None |

Los parámetros configurados en el módulo Xbee correspondiente al nodo Coordinador se muestran en la **Tabla 3.3**.

Tabla 3.3. Parámetros de configuración del módulo Coordinador.

| Cod | Parámetro | Valor ingresado |
|---------------------------|--------------------------|------------------------|
| Networking | | |
| ID | Pan ID | 1704 |
| CE | Coordinador Enable | Enable [1] |
| Addressing | | |
| SH | Serial Number High | 13A200 |
| SL | Serial Number Low | 40D7D11D |
| DH | Destination Address High | 0 |
| DL | Destination Address Low | FFFF |
| NI | Node Identifier | Coordinador |
| Serial Interfacing | | |
| AP | API Enable | Transparent mode [0] |
| Sleep Modes | | |
| SM | Sleep Mode | No Sleep (Router) [0] |

Para que los módulos Xbee de la Serie 2 envíen la información hacia todos los módulos de la red (*modo Broadcast*) se debe configurar el parámetro **DL**: Destination Address Low con el valor FFFF.

Los parámetros configurados en los módulos Xbee correspondientes a cada nodo Ruteador se muestran en la **Tabla 3.4**.

Tabla 3.4. Parámetros de configuración de los módulos Ruteadores.

| Cod | Parámetro | Valor ingresado |
|--------------------|--------------------------|-----------------------|
| Networking | | |
| ID | Pan ID | 1704 |
| CE | Coordinador Enable | Disable [0] |
| JV | Channel Verification | Enable [1] |
| Addressing | | |
| SH | Serial Number High | 13A200 |
| SL | Serial Number Low | 40DCA2E3 |
| DH | Destination Address High | 0 |
| DL | Destination Address Low | FFFF |
| NI | Node Identifier | Router_Luminaria1 |
| Sleep Modes | | |
| SM | Sleep Mode | No Sleep (Router) [0] |

El valor de **SL** que se muestra en la **Tabla 3.4** con color azul cambia debido a que es un valor único en cada módulo xbee existente en el mundo, mientras que el valor **SH** es igual siempre que el fabricante de los módulos Xbee sea el mismo, en este caso **Digi International**.

El valor o nombre del parámetro **NI** de cada ruteador es ingresado a criterio del programador, en este caso se podría ingresarlo de acuerdo con la ubicación del nodo.

Para poder utilizar el módulo Xbee junto con la placa de Arduino Uno se requiere de otra placa de adaptación conocida como Xbee Shield, como la de la **Figura 3.22**, que funciona como conector del módulo Xbee con la placa Arduino, suministrando el voltaje adecuado para el funcionamiento del módulo Xbee, basta con conectarlas entre sí a través de los pines macho y hembra respectivos.



Figura 3.22. Shield Xbee [Autoría Propia]

3.3.3 Sistema de Comunicación Global

Para la comunicación global se puede optar por varios caminos:

1. Comunicación Xbee

Siendo el propósito de la empresa crear y disponer a futuro de una red de comunicación inalámbrica autoadministrable y económica se plantea mantener una comunicación Xbee completa, desde el centro de control hasta los dispositivos más lejanos del prototipo, lo que implica instalar nodos ruteadores exclusivos (hasta darles un uso extra en el futuro) para que retransmitan la información hasta llegar al lugar deseado, lo recomendable es instalar un dispositivo en cada cuadra para garantizar los 120 metros de alcance en línea de vista de estos dispositivos.

2. Comunicación GSM

Este sistema únicamente constaría de dos equipos de comunicación GSM, uno en el nodo controlador de segmento del prototipo de telegestión y otro en el centro de control, los mismos que se comunicarán entre ellos mediante protocolos de comunicación GSM, para hacer esto posible se necesita de la contratación de un plan de mensajería ilimitada a una operadora de telefonía celular para cada equipo.

3. Comunicación GPRS

Este último es el sistema más complejo y completo que se pueda utilizar, se necesitaría únicamente de un dispositivo con comunicación GPRS para enviar los datos desde el nodo controlador de segmento hacia la nube, almacenarlos en una base de datos, procesarlos y

utilizarlos según las necesidades de la empresa, para este equipo se necesita contratar un plan de datos a una operadora de telefonía celular. Por su complejidad, este se puede considerar como un tema de tesis para proyectos futuros, de hecho, el centro de control mantiene abierto este tema para cualquier estudiante que quiera aceptar el desafío, con propósitos que abarcarían el sistema de telegestión y muchos proyectos planificados por el Área de Control.

Para la comunicación global del proyecto con las opciones 2 y 3 se puede utilizar el módulo GSM/GPRS SIM900, que como se puede observar en la **Figura 3.23**, que dispone de una más alta velocidad de transmisión de datos a un mayor alcance.



Figura 3.23. Módulo GSM/GPRS SIM900 para Arduino [Autoría Propia]

El módulo GSM/GPRS requiere de una SIM Card de cualquier operadora (activada) que se coloca en la parte trasera de la placa, sus especificaciones técnicas se encuentran en el **ANEXO X**.

Las opciones 2 y 3 son las más costosas debido a la contratación de servicios adicionales de facturación mensual, para este prototipo se realizará la ejecución del sistema de comunicación global con la primera opción, para lo cual se necesita desarrollar una interfaz gráfica y conectarla directamente desde la computadora desde donde se efectuará el monitoreo al primer nodo ruteador, que en este caso se encontrará instalado en el centro de control. Se puede optar por hacer de este nodo el coordinador de toda la red.

Las configuraciones de los módulos que servirán de ruteadores cada 100 metros aproximadamente son las mismas que en la red local xbee que se muestran en el **ANEXO III**, en la sección de configuración del módulo ruteador. De igual forma el anexo mencionado muestra la configuración del módulo coordinador.

3.3.4 Etapa de Monitoreo

Para la Unidad de Monitoreo del proyecto se desarrolló un programa en Visual Studio para fines de pruebas y operación del prototipo, cuya interfaz gráfica se muestra en la **Figura 3.24**. La ventaja más importante de tener un programa propio es el costo de su desarrollo frente a los programas comerciales encontrados en el mercado.

La interfaz programada permite al usuario realizar las siguientes actividades:

- Establecer la comunicación hacia el equipo descubriendo los puertos disponibles de la computadora y permitiendo seleccionar el correcto si hubiese más de uno.
- Seleccionar la velocidad de transmisión (siempre debe ser la misma a la que fueron configurados los demás equipos de comunicación).
- Enviar cualquier tipo de dato, pueden ser para realizar pruebas o enviar comandos de configuración de los módulos xbee para reconfigurar la red de manera remota, sin tener que ir al lugar de instalación y conectarse a cada módulo.
- Mostrar los datos recibidos sin interpretación y adaptación del software.
- Permite encender y apagar las luminarias de todo el sistema de telegestión.
- Elegir el nivel de intensidad luminosa, entre el 10% y el 100%.
- Observar el estado de cada luminaria del sistema (Encendido: "Luminaria 1 | Encendida" en color verde y Apagado: "Luminaria 1 | Apagada" en color rojo).
- Se observa la recepción de cada evento y se almacena en una tabla que lleva el registro de eventos, en una base de datos propia del programa.
- En otra tabla se almacena el tiempo de cada nivel de iluminación, la energía consumida, el costo y el ahorro energético y económico de cada luminaria.
- Estos datos se pueden exportar a Excel para análisis, cálculos y estudios.

Como proyecto futuro, se propone implementar el monitoreo desde el sistema ADMS del Centro de Control, manteniendo entre los dos programas una redundancia de control y monitoreo, garantizando así confiabilidad de la unidad de monitoreo en caso de error de algún software.



Figura 3.24. Interfaz de monitoreo del prototipo desarrollada en Visual Studio 2017 [Autoría Propia]

Todos los ajustes, configuraciones necesarias y programación de la interfaz de monitoreo se encuentran en el **ANEXO IV**.

En el **ANEXO XI** se encuentra el manual de uso del software elaborado para este proyecto.

3.3.5 Lógica de funcionamiento del prototipo controlador de nodo

El prototipo de control de nodo funciona en la siguiente secuencia de instrucciones:

1. Recibe información desde la unidad de monitoreo (Software).
2. Si recibe un cambio en el estado de la luminaria, ejecuta la acción con el relé.

3. Si recibe un cambio en el estado de flujo luminoso, envía la señal pwm adecuada al mosfet de potencia para permitir la variación del voltaje necesario para realizar el cambio en la intensidad luminosa.
4. Lee la entrada del sensor de corriente y envía la información solo si existe un cambio en el estado de la luminaria.

3.3.6 Prototipos de experimentación del dispositivo controlador de nodo

Primer Prototipo

El primer prototipo implementado acopla los elementos antes mencionados para tratar de controlar el driver de la luminaria con el método 1-10V, obteniendo como resultado la imposibilidad de aplicar este método en las luminarias de prueba, por lo que para la segunda versión se probará el método de control de fase.

En la **Figura 3.25** se puede observar una vista real de la primera versión del prototipo de experimentación.

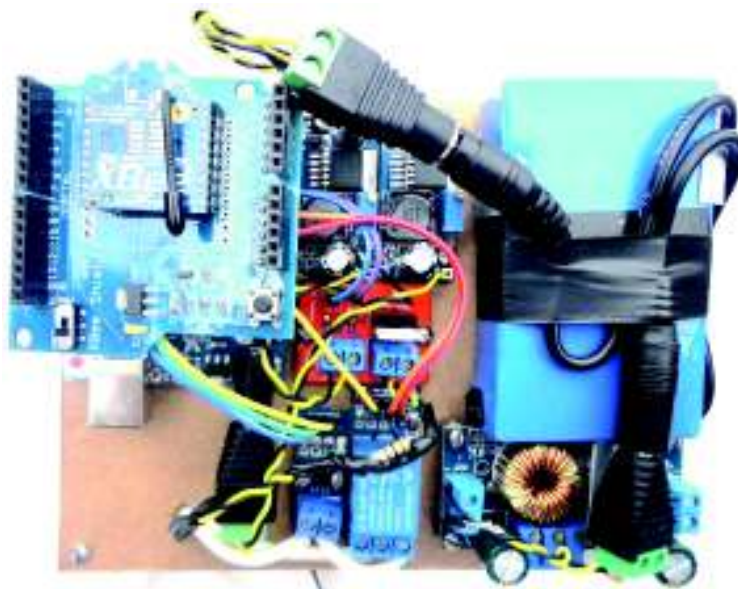


Figura 3.25. Vista real del primer prototipo controlador de nodo para la experimentación
[Autoría Propia]

Segundo Prototipo

Para la segunda versión del prototipo controlador de nodo se eliminó un regulador de voltaje innecesario, manteniendo la misma programación del microcontrolador y de las comunicaciones, cambiando el método de control 1-10V por el método de control de fase aplicado entre el driver y los leds. Con este método se logró tener el cambio en la intensidad luminosa deseado.

Se cambió la batería de la **Figura 3.4** por la batería tipo LIPO de la **Figura 3.6**, junto con sus respectivos controladores de carga y paneles solares. En las pruebas se concluyó que este tipo de batería no soporta las doce horas continuas de uso, lo que para el tercer prototipo utilizará nuevamente la batería de la **Figura 3.4** junto con su respectivo controlador de carga y panel solar.

En la **Figura 3.26** se observa la vista real de la segunda versión del prototipo.



Figura 3.26. Vista real del segundo prototipo controlador de nodo [Autoría Propia]

Tercer Prototipo (Diseño Final)

Para la tercera versión del prototipo controlador de nodo solo se reemplazó la etapa de alimentación, la capacidad de la batería fue suficiente para soportar las 12 horas de operación nocturna y recargarse durante el día.

En la **Figura 3.27** se observa la vista real de la tercera versión del prototipo.



Figura 3.27. Vista real del tercer prototipo controlador de nodo (Versión Final) [Autoría Propia]

El diagrama de bloques de la **Figura 3.28** muestra las etapas de funcionamiento general del dispositivo controlador de nodo.

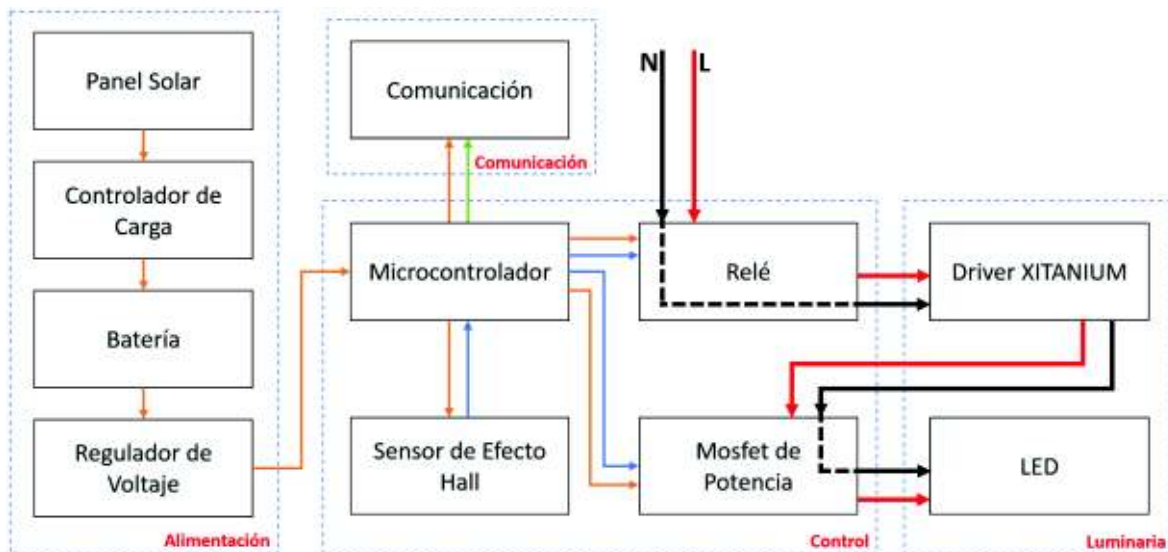


Figura 3.28. Diagrama de bloques del funcionamiento del dispositivo controlador de nodo [Autoría Propia]

La **Figura 3.29** muestra las conexiones de los prototipos desarrollados, señalados en un recuadro azul se especifican la diferencia entre el método 1-10V y el método de control de fase, y en la **Figura 3.30** se observan las luminarias de prueba en el Centro de Control:

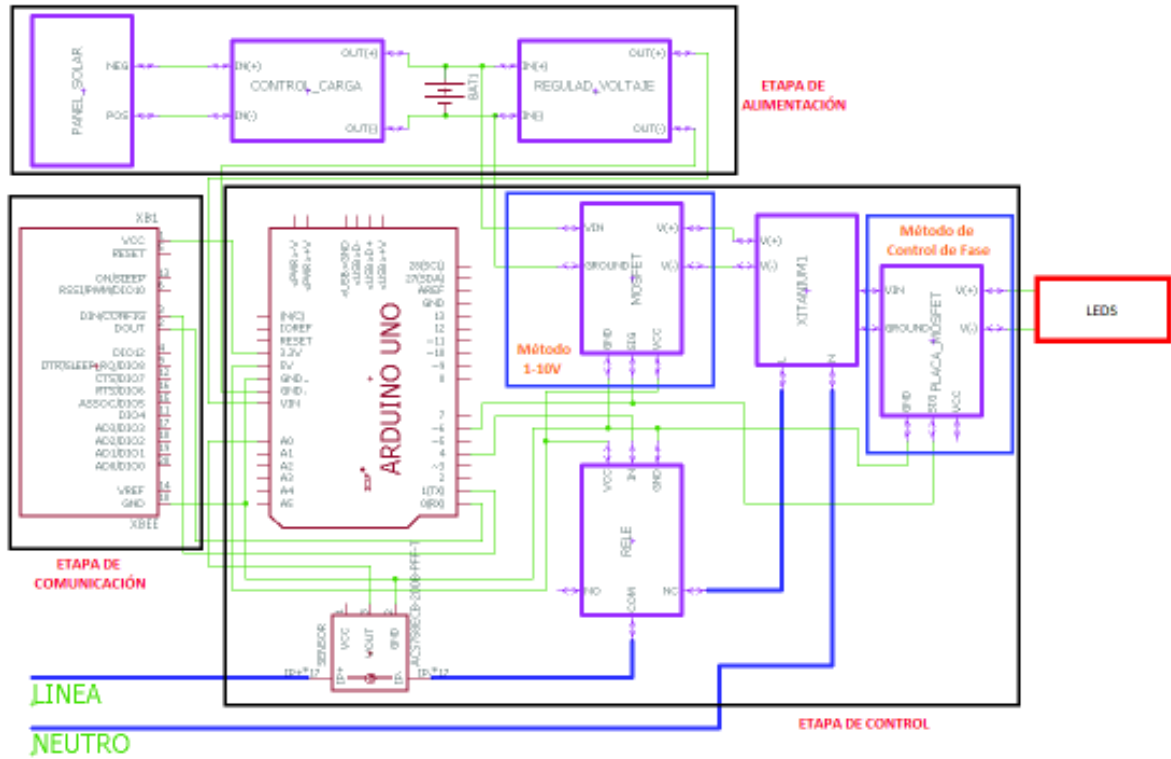


Figura 3.29. Esquema de conexiones del prototipo controlador de nodo [Autoría Propia]

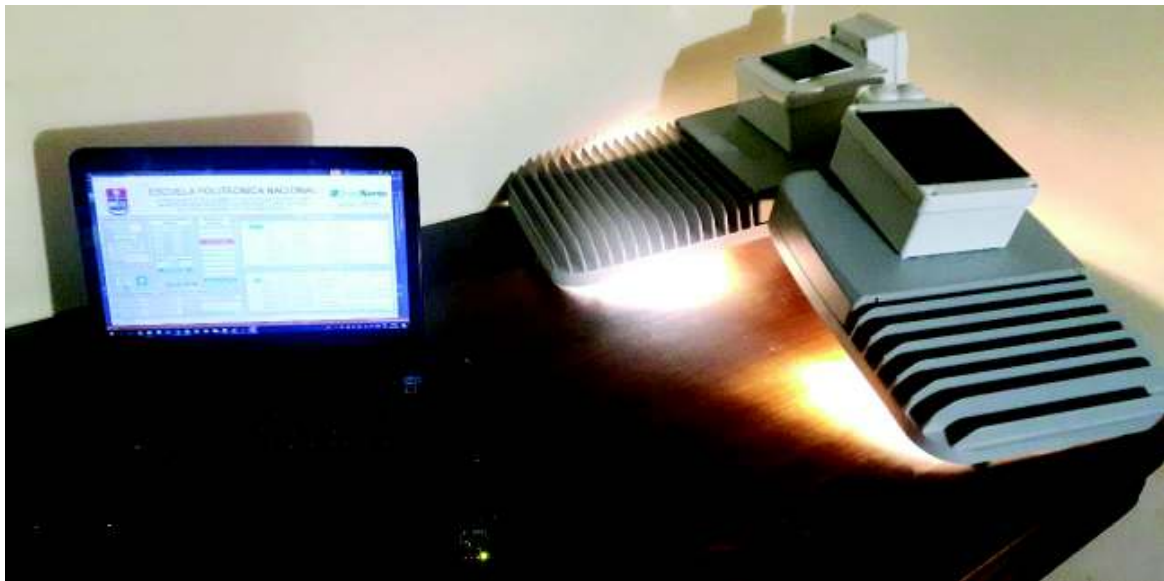


Figura 3.30. Pruebas de laboratorio de las luminarias provistas por la empresa para el desarrollo del proyecto en el Centro de Control [Autoría Propia]

3.4 Propuesta de control de iluminación

Para esta propuesta se desarrollará un análisis de circulación vehicular en las vías donde EMELNORTE tiene instalado las nuevas luminarias Philips tipo LED del proyecto de cambio de luminarias de vapor de sodio por luminarias LED en el centro de la ciudad iniciado en el año 2015, ubicación que se puede observar en la **Figura 3.31**, el cual continuará su ejecución a partir del cuarto trimestre del año 2018; luminarias que se propone a la empresa en este trabajo instalar este sistema de telegestión de alumbrado público.

De igual manera se tendrá como segunda opción el implemento del prototipo en las luminarias Voltana instaladas de la Avenida Camilo Ponce, junto al parque Ciudad Blanca.

Para determinar la intensidad luminosa mínima necesaria para brindar un buen servicio de iluminación en estas vías se realizaron inspecciones nocturnas con la finalidad de analizar la intensidad horaria promedio de usuarios viales.

$$I = \frac{n(x)}{t}$$

Ecuación 3.1. Cálculo de intensidad de área promedio [18]

donde:

I = intensidad horaria promedio

n(x) = número de vehículos que atraviesan la sección fija (x)

t = intervalo temporal de estudio

cuyas unidades son:

$$[\text{Vehículos/Hora}] \text{ o } [\text{Peatones/Hora}]$$

Para este estudio se realizó inspecciones nocturnas tomando el conteo de usuarios viales durante 15 minutos para cada hora de análisis, y asumiendo que el flujo vehicular y peatonal será constante se obtienen resultados aproximados de cada hora para proponer varias alternativas.

3.4.1 Propuesta 1 – Centro de la ciudad de Ibarra

Instalar el prototipo con las nuevas luminarias XITANIUM de PHILIPS que sirven iluminación al centro de Ibarra, alrededor de 15 manzanas, específicamente en las siguientes calles:

Dirección Norte – Sur:

- Sánchez y Cifuentes
- José Joaquín Olmedo
- Simón Bolívar
- Antonio José de Sucre

Dirección Este – Oeste:

- Cristóbal Colón
- Juan de Velasco
- Pedro Moncayo
- Miguel Oviedo

La siguiente figura muestra las calles exactas donde se encuentran instaladas las luminarias tipo LED:

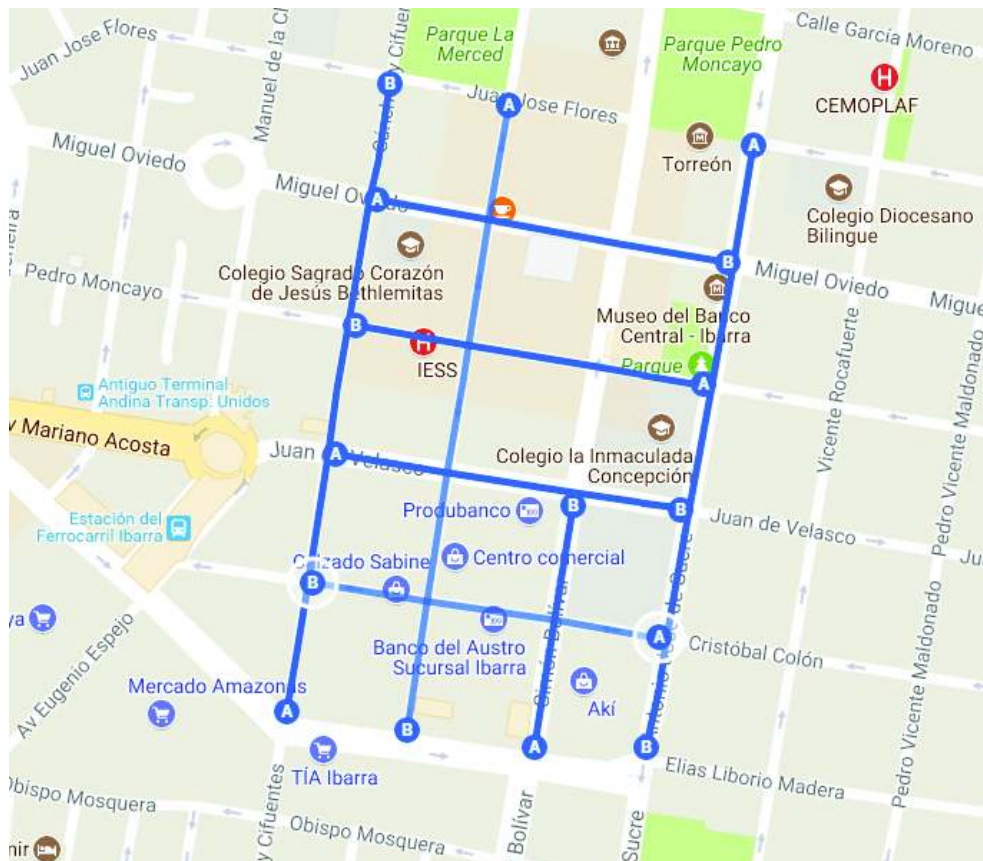


Figura 3.31. Ubicación de instalación de luminarias tipo LED en el centro de Ibarra
[Autoría Propia]

En total, se encuentran instaladas 72 luminarias LED en esta zona, durante las inspecciones diurnas y nocturnas se pudo constatar que dos luminarias se encuentran funcionando las 24 horas del día.

Los resultados de las inspecciones nocturnas para este sector fueron los que se encuentran en la **Tabla 3.5**, junto con las fotografías del lugar de la **Figura 3.32**:

Tabla 3.5. Intensidad horaria promedio de usuarios viales del centro de Ibarra

| Hora | Int. Vehicular | Int. Peatonal |
|---------------|-----------------------|----------------------|
| 18:30 a 19:30 | 588 | 204 |
| 19:30 a 20:30 | 428 | 128 |
| 20:30 a 21:30 | 228 | 108 |
| 21:30 a 22:30 | 152 | 96 |
| 22:30 a 23:30 | 72 | 76 |
| 23:30 a 00:30 | 68 | 36 |
| 00:30 a 01:30 | 48 | 0 |
| 01:30 a 02:30 | 52 | 4 |
| 02:30 a 03:30 | 44 | 12 |
| 03:30 a 04:30 | 36 | 8 |
| 04:30 a 05:30 | 108 | 12 |
| 05:30 a 06:30 | 388 | 236 |



a)

b)

Figura 3.32. Calle José Joaquín Olmedo (Centro de Ibarra) a) 19:30 y b) 00:30
[Autoría Propia]

Con estos resultados se puede obtener la curva de intensidad horaria promedio de usuarios viales, mostrada en la **Figura 3.33**.

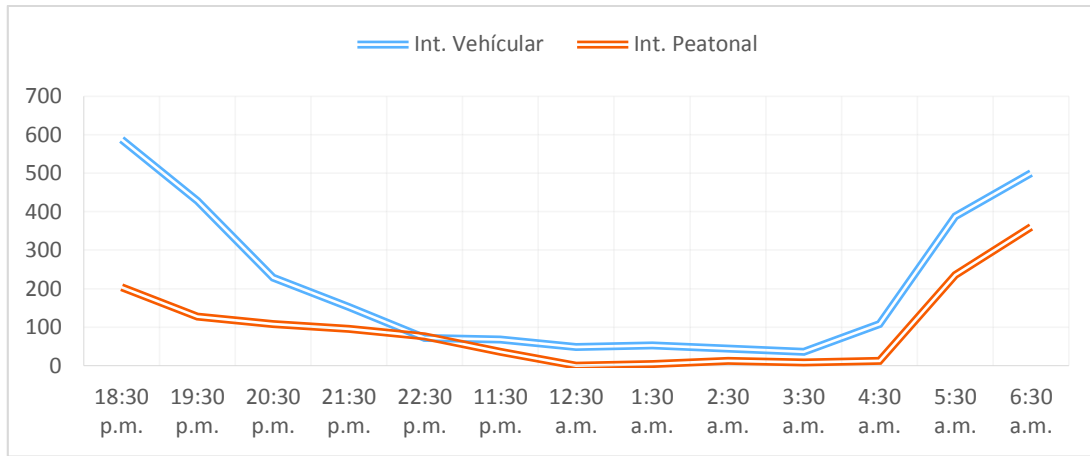


Figura 3.33. Curva de intensidad horaria promedio de usuarios viales del centro de Ibarra [Autoría Propia]

De acuerdo con la curva de la **Figura 3.33** se propone los niveles mínimos a los cuales se recomienda mantener el nivel de iluminación en estas calle, desde las 18:30 (tiempo en que se encienden las lámparas de toda la ciudad) hasta las 06:30 (tiempo en que se apagan las lámparas de toda la ciudad), que se muestran en la **Figura 3.34**. Esta es una recomendación a criterio del autor de este trabajo.

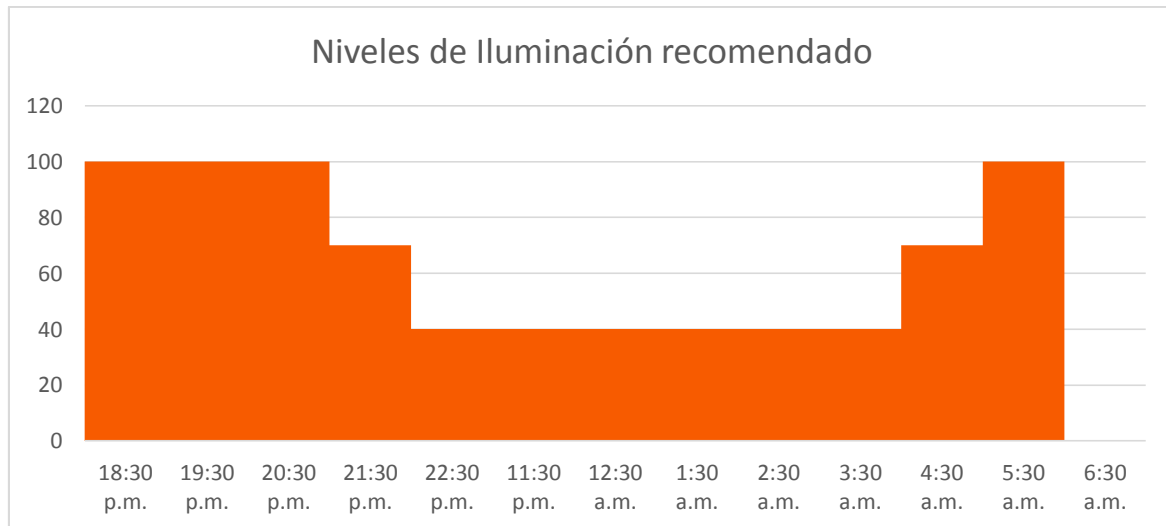


Figura 3.34. Perfil de niveles de iluminación propuesto para el centro de Ibarra (medido en porcentaje) [Autoría Propia]

3.4.2 Propuesta 2 – Av. Camilo Ponce y Galo Plaza (Parque Ciudad Blanca)

Instalar el prototipo en las luminarias VOLTANA 4 de SCHRÉDER instaladas en la Av. Camilo Ponce Enríquez y Galo Plaza Lasso, alrededor del Parque Ciudad Blanca, emblema de iluminación de la ciudad, que disponen de las mismas características dimerizables que las luminarias XITANIUM de PHILIPS.

La cantidad total de 230 luminarias LED Voltana 4 están instaladas en las dos vías paralelas que rodean el parque, ubicadas en la disposición que se observa en la **Figura 3.35**.

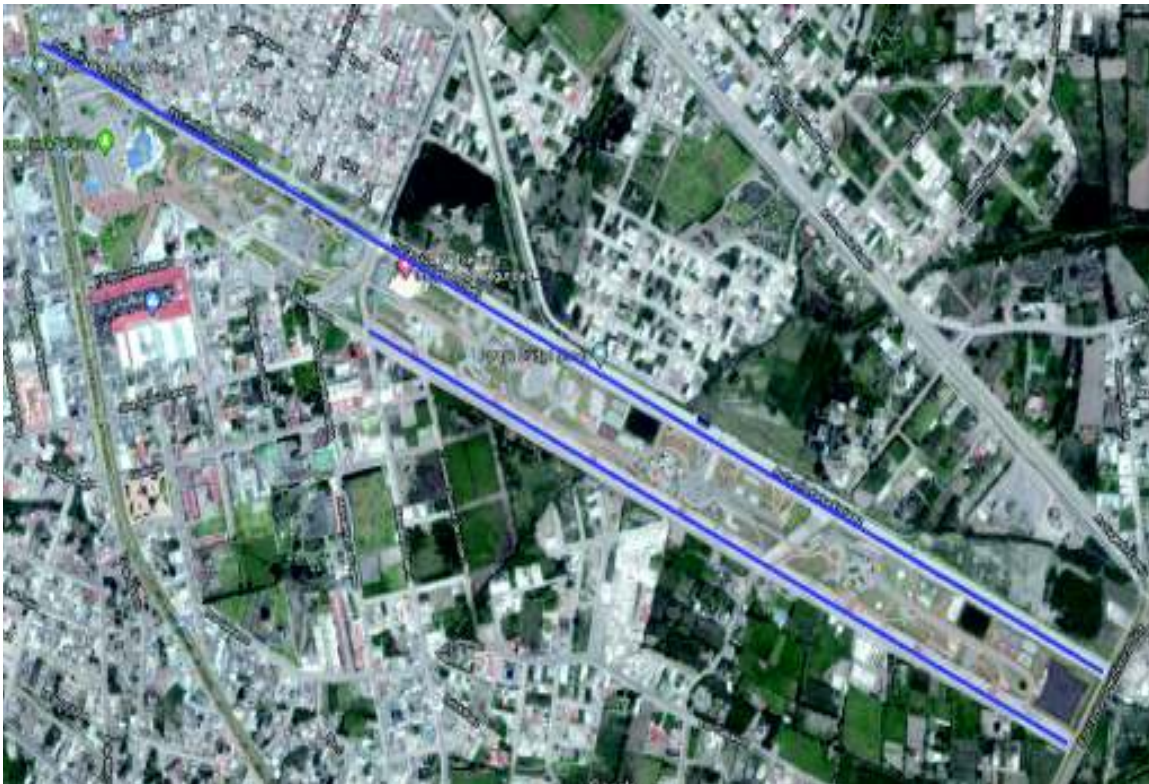


Figura 3.35. Ubicación de instalación de luminarias tipo LED en las calles adyacentes al parque Ciudad Blanca [Autoría Propia]

Los resultados de las inspecciones nocturnas para este sector fueron los de la **Tabla 3.6**, y en la **Figura 3.36** y **Figura 3.37** se puede observar las fotografías del lugar.

Tabla 3.6. Intensidad horaria promedio de usuarios viales de los alrededores del parque Ciudad Blanca

| Hora | Int. Vehicular | Int. Peatonal |
|---------------|----------------|---------------|
| 18:30 a 19:30 | 204 | 32 |
| 19:30 a 20:30 | 112 | 12 |
| 20:30 a 21:30 | 96 | 12 |
| 21:30 a 22:30 | 92 | 4 |
| 22:30 a 23:30 | 36 | 0 |
| 23:30 a 00:30 | 8 | 0 |
| 00:30 a 01:30 | 8 | 0 |
| 01:30 a 02:30 | 4 | 0 |
| 02:30 a 03:30 | 0 | 0 |
| 03:30 a 04:30 | 4 | 0 |
| 04:30 a 05:30 | 20 | 0 |
| 05:30 a 06:30 | 132 | 40 |



Figura 3.36. Av. Camilo Ponce (Parque Ciudad Blanca) 01:15 [Autoría Propia]



a)



b)

Figura 3.37. Av. Camilo Ponce (Parque Ciudad Blanca) a) 19:45 y b) 00:45 [*Autoría Propia*]

Con estos resultados se puede obtener la curva de intensidad horaria promedio de usuarios viales, mostrada en la **Figura 3.38**.

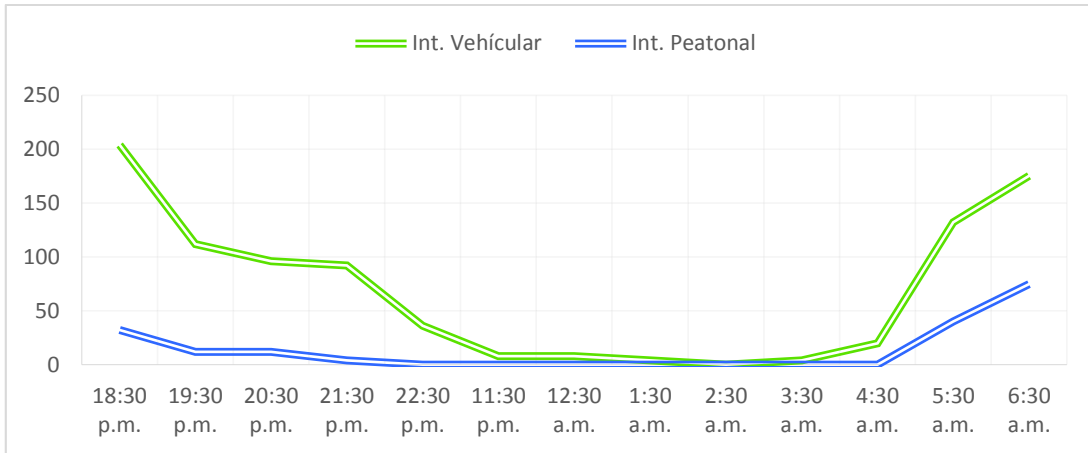


Figura 3.38. Curva de intensidad horaria promedio de usuarios viales de la Av. Camilo Ponce y Galo Plaza Lasso [Autoría Propia]

De igual manera, basándose en la curva de la **Figura 3.38** se propone los niveles mínimos a los cuales se recomienda mantener el nivel de iluminación de estas vías, desde las 18:30 hasta las 06:30, que se muestran en la **Figura 3.39**. La diferencia en este caso radica en que las vías mencionadas se encuentran alejadas de zonas residenciales, además de la excesiva iluminación interna del parque Ciudad Blanca que ayuda a la maximización de ahorro energético con la regulación de estas luminarias.

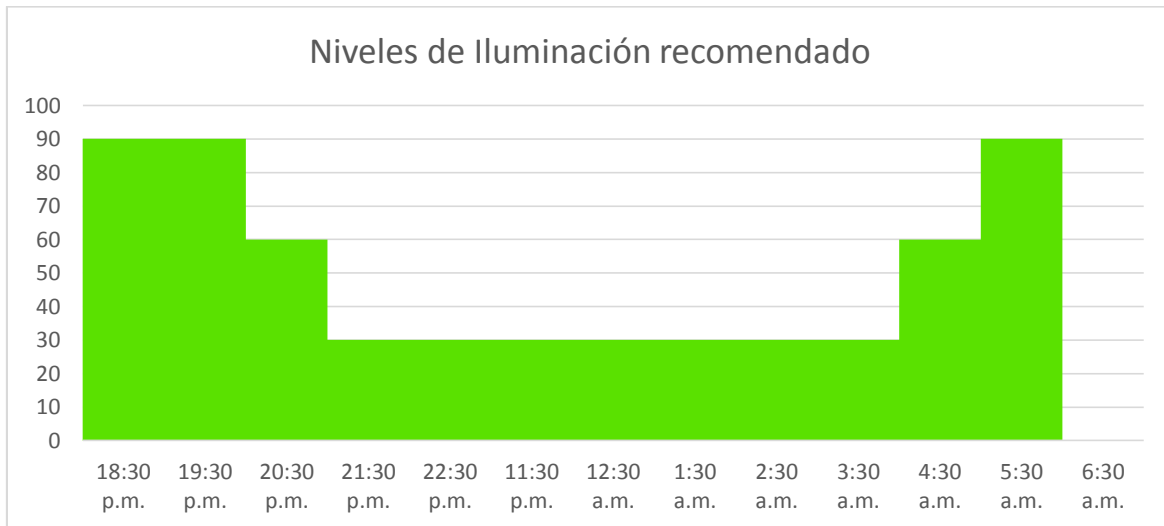


Figura 3.39. Perfil de niveles de iluminación propuesto para la Av. Camilo Ponce y Galo Plaza Lasso (medido en porcentaje) [Autoría Propia]

CAPÍTULO 4.

4. EFICIENCIA ENERGÉTICA CON EL IMPLEMENTO DEL SISTEMA DE TELEGESTIÓN PROPUESTO

4.1 Análisis de consumo energético de alumbrado público con la situación actual

Para este análisis trabajaremos en los cálculos con los datos reales de las dos propuestas, es decir, cantidad exacta de luminarias instaladas en cada sector y potencias nominales de entrada y salida del driver correspondiente a cada luminaria.

4.1.1 Cálculo de las horas de operación

Para alumbrado público se considerará 12 horas diarias de uso de iluminación vial durante un promedio de 30.4 días por mes, obtenidos de los 365 días al año durante 12 meses, entonces, el número total de horas de operación es equivalente a multiplicar estos dos parámetros:

$$H_m = h_d \times d$$

Ecuación 4.1. Cálculo de horas de operación mensual

donde:

H_m = horas de operación mensual

h_d = horas de operación diarias

d = número de días mensuales

$$H_m = 12 \text{ h} \times 30.4 \text{ días} = 364.8 \text{ h/mes}$$

Ecuación 4.2. Horas de operación mensual

4.1.2 Cálculo de la carga instalada

La carga instalada es el resultado de la suma de las potencias de todas las lámparas, o a la vez la multiplicación del número de lámparas con la potencia individual:

$$P_i = P \times n$$

Ecuación 4.3. Cálculo de potencia instalada

donde:

P_i = potencia instalada

P = potencia nominal de una luminaria

n = número de luminaria

Propuesta 1 - Centro de Ibarra

Para este proyecto se usaron lamparas *Philips XITANIUM 150W 0.35 - 0.70A GL Prog sXt*, que tiene una potencia de entrada nominal de 165W, dato obtenido del datasheet del driver de la lámpara que se encuentra en el **ANEXO V**.

Al realizar el cálculo para 72 luminarias de 165W tenemos que:

$$P_i = 165 \text{ W} \times 72 = 11880 \text{ W} = 11.88 \text{ kW}$$

Ecuación 4.4. Potencia instalada para 72 luminarias LED PHILIPS

Propuesta 2 – Av. Camilo Ponce y Galo Plaza (Parque Ciudad Blanca)

Para este proyecto se usaron lamparas *Voltana 4 – LG Innotek LLP 110W 1A 68-110V*, que tiene una potencia de entrada nominal de 130W y potencia de salida máxima hacia el panel de LED de 110W, dato obtenido del driver de la lámpara que se encuentra en la **Figura 4.1**.

Al realizar el cálculo para 230 luminarias de 130W tenemos que:

$$P_i = 130 \text{ W} \times 230 = 37950 \text{ W} = 29.9 \text{ kW}$$

Ecuación 4.5. Potencia instalada para 230 luminarias LED VOLTANA 4



Figura 4.1. Driver LG Innotek LLP 110W 1A 68-110V - Luminaria Voltana 4 [Autoría Propia]

4.1.3 Cálculo de consumo de energía

El consumo de energía eléctrica es equivalente a multiplicar la potencia instalada con el número de horas de funcionamiento:

$$E = P_i \times H_m$$

Ecuación 4.6. Cálculo de energía consumida mensualmente con el sistema actual
donde:

E = energía consumida mensualmente

P_i = potencia instalada

H_m = horas de operación mensual

Propuesta 1 - Centro de Ibarra

$$E = 11.88 \text{ kW} \times 364.8 \text{ h/mes} = 4333.82 \text{ kWh/mes}$$

Ecuación 4.7. Energía consumida mensualmente con el sistema actual para la propuesta

Propuesta 2 – Av. Camilo Ponce y Galo Plaza (Parque Ciudad Blanca)

$$E = 29.9 \text{ kW} \times 364.8 \text{ h/mes} = 10907.52 \text{ kWh/mes}$$

Ecuación 4.8. Energía consumida mensualmente con el sistema actual para la propuesta 2

4.1.4 Medición del consumo de energía de las luminarias de prueba sin la aplicación del prototipo

Para corroborar los valores calculados, en el centro de control de la empresa se realizaron mediciones experimentales del consumo de energía de dos lámparas LED Philips de prueba facilitadas por el departamento de alumbrado público durante 6 días, 6 horas al día con un medidor electrónico proporcionado por la empresa, los resultados obtenidos se muestran en las fotografías de la **Figura 4.2**.



Figura 4.2 Medidas obtenidas Antes y Después con el sistema actual [Autoría Propia]

En la **Ecuación 4.7** se muestra el consumo calculado para 72 luminarias, durante 30.4 días, 12 horas al día, su equivalente al caso de medición real es $E_c = 11.88 \text{ kWh}/72 \text{ h}$.

Los valores medidos que se observan en la **Figura 4.2** no son exactos debido a que la precisión del medidor utilizado no presenta valores decimales, el valor medido es

aproximadamente de $E_M = 12 \text{ kWh}/72 \text{ h}$, con un error aproximado de $\pm 1 \text{ kWh}$. Pese a ese inconveniente, esta prueba de medición nos ofrece un valor semejante al calculado.

Las lámparas Philips para experimentación fueron extraídas de su ubicación en campo únicamente por 2 semanas, tiempo proporcionado por la empresa para cualquier experimentación en beneficio de este proyecto.

Las luminarias Voltana no se pudo conseguir debido a que el departamento de alumbrado público no pudo facilitar la autorización para retirar ninguna lámpara de la Av. Camilo Ponce para fines de experimentación, tampoco disponen de lámparas de esta marca en bodega, todas las lamparas compradas de esta marca fueron instaladas en la Av. Camilo Ponce y Galo Plaza Lasso.

4.2 Análisis de consumo energético de alumbrado público con la implementación del prototipo de sistema de telegestión

En esta sección se analizará la energía consumida con la implementación del prototipo para las dos propuestas presentadas, partiendo de los perfiles de niveles de iluminación propuestos para los 2 casos.

Para relacionar con los cálculos de energía consumida de la situación actual, se tomará como base la misma cantidad de luminarias en cada sector.

El dato de potencia de salida P_o nos brinda las pérdidas de potencia consumida por el LED driver:

$$P_d = P - P_o$$

Ecuación 4.9. Cálculo de pérdidas de potencia del LED driver

donde:

P_d = *pérdidas de potencia del LED driver*

P = *potencia nominal de salida de la luminaria*

P_o = *potencia de salida controlable máxima de la luminaria*

Propuesta 1 - Centro de Ibarra

La potencia de salida controlable de cada luminaria está dentro del rango 7.5W a 150W, valores para 1% y 100% de iluminación respectivamente, dato obtenido del datasheet de la luminaria del **ANEXO V**.

$$P_d = 165 W - 150 W = 15 W$$

Ecuación 4.10. Potencia constante del LED driver XITANIUM

La variación del voltaje de la fuente ocasiona una variación lineal en cuanto a voltaje y corriente, y por lo tanto también de la potencia de salida, cuyo comportamiento con respecto al porcentaje de iluminación viene dado por la expresión:

$$95x - 66y + 400 = 0$$

Ecuación 4.11. Expresión que relaciona la potencia de salida del driver XITANIUM con el porcentaje de intensidad luminosa

Que se obtiene del cálculo de la ecuación de la recta conociendo dos puntos:

$$A(1\%, 7.5 W) \text{ y } B(100\%, 150 W)$$

Con la expresión:

$$\frac{x - x_1}{y - y_1} = \frac{x_2 - x_1}{y_2 - y_1}$$

Ecuación 4.12. Expresión para calcular la ecuación de una recta conociendo dos puntos

Resolución:

$$\rightarrow \frac{x-1}{y-7.5} = \frac{100-1}{150-7.5}$$

$$\rightarrow \frac{x-1}{y-7.5} = \frac{66}{95}$$

$$\rightarrow 95(x - 1) = 66(y - 7.5)$$

$$\rightarrow 95x - 95 = 66y - 495$$

$$\rightarrow 95x - 66y + 400 = 0$$

Entonces, con esta ecuación se calcula los valores de potencia de salida para cada intervalo de 10%, los cuales se muestran en la **Tabla 4.1**, y en la **Figura 4.3** se observa la gráfica de potencia en función de iluminación.

Tabla 4.1. Potencia de salida del driver XITANIUM para cada porcentaje de iluminación

| Iluminación [%] | Potencia [W] | Iluminación [%] | Potencia [W] |
|-----------------|--------------|-----------------|--------------|
| 100 | 150 | 50 | 78,0303 |
| 90 | 135,6061 | 40 | 63,63636 |
| 80 | 121,2121 | 30 | 49,24242 |
| 70 | 106,8182 | 20 | 34,84848 |
| 60 | 92,42424 | 10 | 20,45455 |

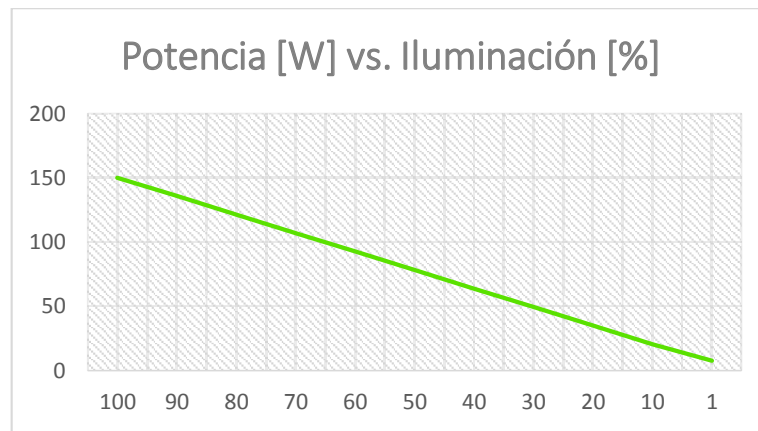


Figura 4.3. Gráfica de Potencia vs. Iluminación (Driver XITANIUM) [Autoría Propia]

Propuesta 2 - Av. Camilo Ponce y Galo Plaza (Parque Ciudad Blanca)

Al igual que en la luminaria PHILIPS, la potencia de salida controlable de cada luminaria VOLTANA 4 está dentro del rango 20 W a 110 W, dato obtenido de su driver.

$$P_d = 130 W - 110 W = 20 W$$

Ecuación 4.13. Potencia constante del LED driver LG Innotek

El comportamiento de la potencia de salida con respecto al porcentaje de iluminación viene dado por la expresión:

$$10x - 11y + 210 = 0$$

Ecuación 4.14. Expresión que relaciona la potencia de salida del driver LG Innotek con el porcentaje de intensidad luminosa

Que se obtiene del cálculo de la ecuación de la recta conociendo dos puntos:

$$A(1\%, 20 W) \text{ y } B(100\%, 110 W)$$

Entonces, con esta ecuación se calcula los valores de potencia de salida para cada intervalo de 10%, los cuales se muestran en la **Tabla 4.2**, y en la **Figura 4.4** se observa la gráfica de potencia en función de iluminación.

Tabla 4.2. Potencia de salida del driver LG Innotek para cada porcentaje de iluminación

| Iluminación [%] | Potencia [W] | Iluminación [%] | Potencia [W] |
|-----------------|--------------|-----------------|--------------|
| 100 | 110 | 50 | 64,54545455 |
| 90 | 100,9090909 | 40 | 55,45454545 |
| 80 | 91,81818182 | 30 | 46,36363636 |
| 70 | 82,72727273 | 20 | 37,27272727 |
| 60 | 73,63636364 | 10 | 28,18181818 |

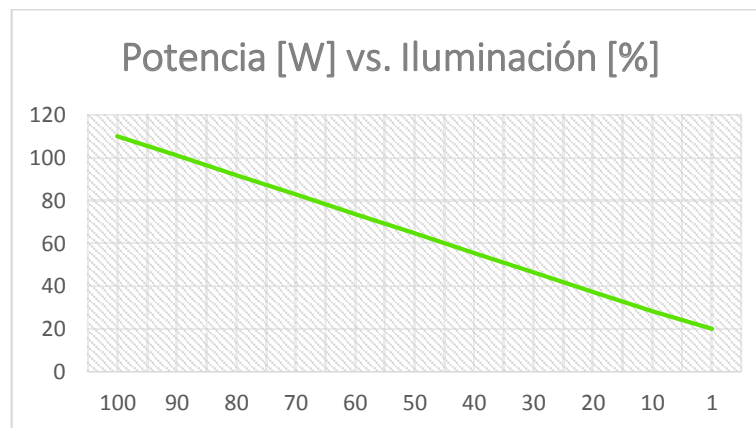


Figura 4.4. Gráfica de Potencia vs. Iluminación (Driver LG Innotek) [Autoría Propia]

Para calcular la energía de N luminarias con la influencia de la regulación de intensidad luminosa se utiliza esta expresión:

$$E = N \times \left((P_d \times 12) + \sum_{j=1}^m (P_j \times H_j) \right) \times 30.4 \text{ días}$$

Ecuación 4.15. Cálculo de energía consumida mensualmente con regulación de iluminación

donde:

N = número de luminarias

E = energía consumida mensualmente

P_d = potencia constane del LED driver

P_j = potencia de la luminaria con nivel (j)

H_j = tiempo que permanece funcionando la iluminaria con nivel (j)

4.2.1 Cálculo de la eficiencia energética para la propuesta 1

De la **Figura 3.34**, que muestra los niveles de iluminación propuestos para el centro de la ciudad de Ibarra se tiene 3 niveles durante las 12 horas de uso de las luminarias, entonces se tendrá consumos en cada nivel y la suma de estos será la energía total consumida, los datos necesarios para este caso se encuentran en la **Tabla 4.3**.

Tabla 4.3. Datos de cada nivel para la Calle Sánchez y Cifuentes

| Nivel [%] | # Horas [h] | Potencia [W] |
|-----------|-------------|--------------|
| 100 | 4 | 150 |
| 70 | 2 | 106,8182 |
| 40 | 6 | 63,63636 |

→ $E_1 = 72 \times ((15 \times 12) + (150 \times 4) + (106,8182 \times 2) + (63,63636 \times 6)) \times 30.4 \text{ días}$

→ $E_1 = 3010.59 \text{ kWh/mes}$

Ahorro energético con la Propuesta 1

$$E_{a1} = \left(1 - \frac{E_1}{E}\right) \times 100$$

Ecuación 4.16. Cálculo del ahorro energético con la propuesta 1

donde:

E_{a1} = porcentaje de energía ahorrada mensualmente

E = energía consumida mensualmente con la situación actual

E_1 = energía consumida mensualmente con la propuesta 1

$$E_{a1} = \left(1 - \frac{3010.59}{4333.82}\right) \times 100 = 30.53\%$$

Ecuación 4.17. Porcentaje de ahorro energético efectivo con la propuesta 1

4.2.2 Medición del consumo de energía de la luminaria de prueba con la aplicación del prototipo

De igual manera, los valores calculados deben ser contrastados con valores reales medidos del consumo de energía de la lámpara de prueba con la aplicación del prototipo durante 3 días, 12 horas al día con el mismo medidor.

El valor calculado para la propuesta 1 es $E_1 = 3010.59 \text{ kWh/mes}$, y su equivalente al caso de medición real es $E_c = 8.25 \text{ kWh}/72h$.

Los resultados obtenidos de las mediciones se muestran en la fotografía de la **Figura 4.5**.



Figura 4.5 Medidas obtenidas Antes y Después con la aplicación del prototipo [Autoría Propia]

4.2.3 Cálculo de la eficiencia energética para la propuesta 2

De igual manera, de la **Figura 3.39**, que muestra los niveles de iluminación propuestos para la Avenida Camilo Ponce se tiene 3 niveles para las 12 horas de uso de las luminarias, los datos necesarios para este caso se encuentran en la **Tabla 4.4**.

Tabla 4.4. Datos de cada nivel para la Avenida Camilo Ponce

| Nivel [%] | # Horas [h] | Potencia [W] |
|-----------|-------------|--------------|
| 90 | 3 | 100,90909 |
| 60 | 2 | 73,63636 |
| 30 | 7 | 46,36363 |

$$\rightarrow E_2 = 230 \times ((20 \times 12) + (100,90909 \times 3) + (73,63636 \times 2) + (46,36363 \times 7)) \times 30,4 \text{ días}$$

$$\rightarrow E_2 = 7093.70 \text{ kWh/mes}$$

Ahorro energético con la Propuesta 2

$$E_{a2} = \left(1 - \frac{E_2}{E}\right) \times 100$$

Ecuación 4.18. Cálculo del ahorro energético con la propuesta 2

donde:

E_{a2} = porcentaje de energía ahorrada mensualmente

E = energía consumida mensualmente con la situación actual

E_2 = energía consumida mensualmente con la propuesta 2

$$E_{a1} = \left(1 - \frac{7093.70}{10907.52}\right) \times 100 = 34.97\%$$

Ecuación 4.19. Porcentaje de ahorro energético efectivo con la propuesta 2

En la **Tabla 4.5** se muestra el resumen de cálculos energéticos para las dos propuestas.

Tabla 4.5. Resumen de Cálculos Eléctricos del Sistema de Telegestión de Alumbrado Público

| Análisis Eléctrico del Sistema de Telegestión de Alumbrado Público | | |
|---|--------------------|--------------------|
| | Propuesta 1 | Propuesta 2 |
| Potencia instalada | 11,88 kW | 29,9 kW |
| Energía al 100% | 4333,82 kWh/mes | 10907,52 kWh/mes |
| Energía con telegestión | 3010,59 kWh/mes | 7093,70 kWh/mes |
| Ahorro Energético | 30,53 % | 34,97 % |

CAPÍTULO 5.

5. EVALUACIÓN TÉCNICO-ECONÓMICA DEL SISTEMA DE TELEGESTIÓN DESARROLLADO

5.1 Mantenimiento preventivo y correctivo de luminarias de alumbrado público

Los trabajos de mantenimiento de alumbrado público son un gasto que puede reducirse con el implemento de sistemas de control y telegestión de alumbrado público.

5.1.1 Mantenimiento preventivo

La empresa eléctrica EMELNORTE S.A. no realiza mantenimientos preventivos constantes en cada uno de los elementos de las instalaciones de alumbrado público, tampoco se lleva un adecuado cronograma de trabajo para el reemplazo de las luminarias por terminación de su vida útil. Los gastos por mantenimiento se producen cuando existen reportes de averías por parte de los usuarios o se encuentra fallos durante las contadas inspecciones anuales del personal de alumbrado público.

El sistema de telegestión de alumbrado público puede proporcionar a la empresa un considerable ahorro monetario al llevar un registro de tiempo de uso y parámetros de funcionamiento normal de luminarias de reciente instalación, así se puede realizar mantenimiento únicamente de las luminarias que lo necesiten, o un recambio de las que han llegado a su límite de vida útil, planificando y reduciendo la cantidad de ejecuciones de todas estas acciones.

La vida útil de las luminarias es diferente de acuerdo con el tipo de lámpara y su fabricante, siendo para las lámparas tipo LED mucho mayor a las lámparas de Vapor de Sodio, una comparación aproximada se puede ver en la **Tabla 5.1**.

Tabla 5.1. Comparación de vida útil entre lámparas de Vapor de Sodio y LED [19]

| Tipo de Lámpara | Vida útil [h] |
|------------------------|----------------------|
| Vapor de Sodio | 10.000 – 28.000 |
| LED | >50.000 |

Los valores mostrados en la tabla anterior equivalen al caso donde la luminosidad de las lámparas es el 100% de su capacidad, por otro lado, la vida útil de las lámparas aumenta con el uso de un sistema de regulación.

5.1.2 Mantenimiento correctivo

El mantenimiento correctivo se realiza para reparar las averías en todas las luminarias, procediendo a la sustitución o reparación de lámparas, sistemas de alimentación o control después de que han sido reportadas por los usuarios viales.

Las consecuencias ocasionadas por el tiempo que permanecen estas lámparas dañadas se pueden evitar gracias al sistema de telegestión, obteniendo ahorros económicos y beneficios sociales con la ciudadanía.

Los costos por mantenimiento anual por luminarias de alumbrado público en EMELNORTE S.A. es \$70.00 aproximadamente, dato obtenido gracias al Ing. Felipe Aguirre, representante del departamento de alumbrado público de la empresa.

La empresa puede ahorrar de 20 a 35% en gastos de mantenimiento de alumbrado público con el uso de sistemas de telegestión, gracias al registro en tiempo real del funcionamiento de cada luminaria.

5.2 Análisis económico de la implementación de sistemas de telegestión comerciales

Las empresas que ofrecen servicios de iluminación disponen en el mercado de hardware y software especializado para la gestión de iluminación, al ser esta una tecnología en auge de desarrollo el coste de esta es extremadamente alto, lo que es una ventaja significativa para el prototipo desarrollado en este proyecto de titulación.

5.2.1 Propuesta de Telegestión Owlet Nightshift del Grupo Schröder

El grupo Schröder Bolivia organizó un curso de iluminación LED en Cochabamba, en el mes de Julio de 2017, donde se proporcionó información sobre la iluminación LED, sistemas de alumbrado, sistemas de telegestión y productos relacionados de la marca, del que se obtuvo información técnica y costos referenciales de primera mano de sus organizadores, como resumen de su servicio se muestra el esquema de la **Figura 5.1**.

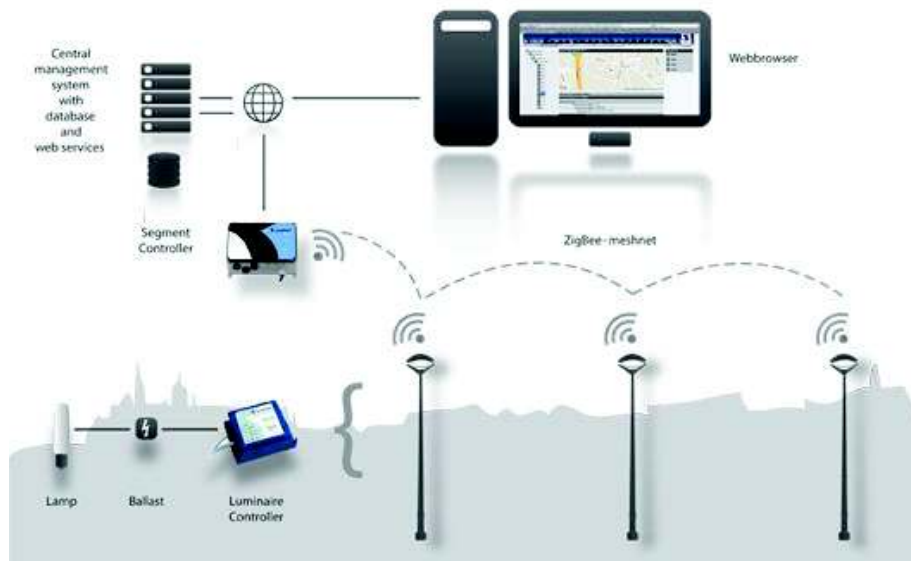


Figura 5.1. Esquema del servicio Owlet Nightshift de Schröder [20]

Cada Controlador de Segmento de Schröder puede administrar hasta 150 Controladores de Luminaria.

Costos para la Propuesta 1 – Centro de Ibarra

Tabla 5.2. Listado de equipos de la propuesta de Schröder para 72 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|---------------------|
| 1 | Luminaire Controller (LuCo) | \$ 97,00 | 72 | \$ 6.984,00 |
| 2 | Segment Controller (SeCo) | \$ 11.550,00 | 1 | \$ 11.550,00 |
| 3 | Central Management System with Database and Web Services | \$ 0,00 | 1 | \$ 0,00 |
| 4 | Software - WebBrowser | \$ 45.000,00 | 1 | \$ 45.000,00 |
| Total de costos RED INTERGESTIONABLE OWLET NIGHTSHIFT | | | | \$ 69.534,00 |

Tabla 5.3. Precios de materiales e instalación para la propuesta de Schröder para 72 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|---------------------|
| 1 | Caja para tableros de control | \$ 500,00 | 1 | \$ 500,00 |
| 2 | Materiales para instalación | \$ 350,00 | 1 | \$ 350,00 |
| 3 | Asesoría y servicios para implementación completa. (4 meses aproximadamente) | \$ 16.700,00 | 1 | \$ 16.700,00 |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 72 | \$ 1.512,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 2 | \$ 42,00 |
| Total de costos de materiales e instalación | | | | \$ 19.104,00 |

El costo total de esta propuesta es US\$ 82.638,00.

Costos para la Propuesta 2 – Av. Camilo Ponce y Galo Plaza

Tabla 5.4. Listado de equipos de la propuesta de Schröder para 230 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|---------------------|
| 1 | Luminaire Controller (LuCo) | \$ 97,00 | 230 | \$ 22.310,00 |
| 2 | Segment Controller (SeCo) | \$ 11.550,00 | 2 | \$ 23.100,00 |
| 3 | Central Management System with Database and Web Services | \$ 0,00 | 1 | \$ - |
| 4 | Software - WebBrowser | \$ 45.000,00 | 1 | \$ 45.000,00 |
| Total de costos RED INTERGESTIONABLE OWLET NIGHTSHIFT | | | | \$ 90.410,00 |

Tabla 5.5. Precios de materiales e instalación para la propuesta de Schröder para 230 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|---------------------|
| 1 | Caja para tableros de control | \$ 500,00 | 2 | \$ 1.000,00 |
| 2 | Materiales para instalación | \$ 350,00 | 1 | \$ 350,00 |
| 3 | Asesoría y servicios para implementación completa. (4 meses aproximadamente) | \$ 16.700,00 | 1 | \$ 16.700,00 |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 230 | \$ 4.830,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 3 | \$ 63,00 |
| Total de costos de materiales e instalación | | | | \$ 19.692,00 |

El costo total de esta propuesta es US\$ 113.353,00.

5.2.2 Propuesta de Telegestión de Philips

La empresa Master Light, con el Ingeniero Rodrigo Polo representante de Philips en Ecuador facilitaron información técnica y costos de los equipos y productos de la marca, siendo ellos los vendedores de las lámparas LED que la empresa eléctrica ha adquirido para el proyecto de sustitución vigente.

Cada Controlador de Segmento de Philips puede administras hasta 150 Controladores de Luminaria, al igual que en el equipo de Schröder.

Costos para la Propuesta 1 – Centro de Ibarra

Tabla 5.6. Listado de equipos de la propuesta de Philips para 72 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|------------------------------------|-----------------|----------|---------------------|
| 1 | Starsense Wireless LLC7300/00 | \$ 103,00 | 72 | \$ 7.416,00 |
| 2 | Controlador de Segmento (AmpLight) | \$ 12.500,00 | 1 | \$ 12.500,00 |
| 3 | System Management with Database | \$ 0,00 | 1 | \$ - |
| 4 | Starsense Supervisor Software | \$ 50.000,00 | 1 | \$ 50.000,00 |
| Total de costos SISTEMA STARSENSE | | | | \$ 69.916,00 |

Tabla 5.7. Precios de materiales e instalación para la propuesta de Philips para 72 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|--------------------|
| 1 | Caja para tablero de control | \$ 500,00 | 1 | \$ 500,00 |
| 2 | Materiales para instalación | \$ 350,00 | 1 | \$ 350,00 |
| 3 | Asesoría y servicios para implementación completa. | \$ 0,00 | 1 | \$ 0,00 |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 72 | \$ 1.512,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 2 | \$ 42,00 |
| Total de costos de materiales e instalación | | | | \$ 2.404,00 |

El costo total de esta propuesta es US\$ 72.320,00.

Costos para la Propuesta 2 – Av. Camilo Ponce y Galo Plaza

Tabla 5.8. Listado de equipos de la propuesta de Philips para 230 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|------------------------------------|-----------------|----------|---------------------|
| 1 | Starsense Wireless LLC7300/00 | \$ 103,00 | 230 | \$ 7.416,00 |
| 2 | Controlador de Segmento (AmpLight) | \$ 12.500,00 | 2 | \$ 12.500,00 |
| 3 | System Management with Database | \$ 0,00 | 1 | \$ - |
| 4 | Starsense Supervisor Software | \$ 50.000,00 | 1 | \$ 50.000,00 |
| Total de costos SISTEMA STARSENSE | | | | \$ 69.916,00 |

Tabla 5.9. Precios de materiales e instalación para la propuesta de Philips para 230 luminarias

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|--------------------|
| 1 | Caja para tablero de control | \$ 500,00 | 2 | \$ 1.000,00 |
| 2 | Materiales para instalación | \$ 350,00 | 1 | \$ 350,00 |
| 3 | Asesoría y servicios para implementación completa. | \$ 0,00 | 1 | \$ - |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 230 | \$ 4.830,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 3 | \$ 63,00 |
| Total de costos de materiales e instalación | | | | \$ 6.243,00 |

El costo total de esta propuesta es US\$ 104.933,00.

5.3 Análisis económico de la implementación del prototipo desarrollado

La diferencia fundamental del prototipo desarrollado frente a los productos anteriormente detallados es el costo, siendo extremadamente bajo, lo que beneficia a la empresa y permite implementarlo para un mayor número de luminarias de alumbrado público.

5.3.1 Costos de materiales del equipo controlador de nodo

Al ser un prototipo en desarrollo, se tiene costos de cada elemento que forma parte de este, todos los elementos utilizados pueden encontrarse en el comercio electrónico de Ecuador, aunque de ser posible se recomienda importar los elementos desde China para tener un ahorro de más del 40%.

A continuación, se muestra la diferencia de precios entre los elementos comprados en Ecuador y en China, hay que recalcar que los tramites de importación son aproximadamente del 30%, dato obtenido del taller de importaciones dirigido por Cristian Muñoz en la empresa Wosi Solutions el sábado 9 de septiembre de 2017.

5.3.2 Costos de materiales para el dispositivo controlador de Nodo del prototipo desarrollado

Los costos de los materiales con precios del mercado electrónico ecuatoriano se muestran en la **Tabla 5.10** y la proforma con los materiales de la Empresa ecuatoriana WOLF ELECTRONICS se encuentran en el **ANEXO VI**.

Tabla 5.10. Precios unitarios de los materiales en Ecuador

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|------------------------|---|-----------------|----------|------------------|
| 1 | Arduino Uno R3 | \$ 10,08 | 1 | \$ 10,08 |
| 2 | Módulo XBee Serie 2 | \$ 39,20 | 1 | \$ 39,20 |
| 3 | Shield Xbee | \$ 7,84 | 1 | \$ 7,84 |
| 4 | Módulo MOSFET (Comercial o Diseñado) | \$ 3,92 | 1 | \$ 3,92 |
| 5 | Sensor de corriente (ACS712ELCTR-05B-T) | \$ 5,04 | 1 | \$ 5,04 |
| 6 | Módulo Relé 5Vdc - 125Vac (1 canal) | \$ 5,04 | 1 | \$ 5,04 |
| 7 | Mini Panel Solar 12V 1,5W | \$ 7,28 | 1 | \$ 7,28 |
| 8 | Batería recargable 12V - 1800mAh | \$ 16,24 | 1 | \$ 16,24 |
| 9 | Módulo de carga de batería Step-Down DC-DC MPPT 12V | \$ 7,28 | 1 | \$ 7,28 |
| 10 | Módulo Regulador de voltaje Step-down DC-DC (LM2596) 12V-5V | \$ 6,72 | 1 | \$ 6,72 |
| Total de costos | | | | \$ 108,64 |

Los costos de los materiales con precios por unidad al realizar la importación desde China se encuentran en la **Tabla 5.11**.

Tabla 5.11. Precios unitarios de los materiales importados desde China

| Item | Descripción | Precio Unitario | Precio + 30% Importación | Cantidad | Precio Total USD |
|------------------------|---|-----------------|--------------------------|----------|------------------|
| 1 | Arduino Uno R3 | \$ 2,64 | \$ 3,43 | 1 | \$ 3,43 |
| 2 | Módulo XBee Serie 2 | \$ 21,99 | \$ 28,59 | 1 | \$ 28,59 |
| 3 | Shield Xbee | \$ 1,99 | \$ 2,59 | 1 | \$ 2,59 |
| 4 | Módulo MOSFET | \$ 0,50 | \$ 0,65 | 1 | \$ 0,65 |
| 5 | Sensor de corriente (ACS712ELCTR-05B-T) | \$ 1,31 | \$ 1,70 | 1 | \$ 1,70 |
| 6 | Módulo Relé 5Vdc - 125Vac (1 canal) | \$ 0,67 | \$ 0,87 | 1 | \$ 0,87 |
| 7 | Mini Panel Solar 12V 1,5W | \$ 2,72 | \$ 3,54 | 1 | \$ 3,54 |
| 8 | Batería recargable 12V - 1800mAh | \$ 7,19 | \$ 9,35 | 1 | \$ 9,35 |
| 9 | Módulo de carga de batería Step-Down DC-DC MPPT 12V | \$ 5,98 | \$ 7,77 | 1 | \$ 7,77 |
| 10 | Módulo Regulador de voltaje Step-down DC-DC (LM2596) 12V-5V | \$ 0,71 | \$ 0,92 | 1 | \$ 0,92 |
| Total de costos | | | | | \$ 59,41 |

Ahorro económico entre elementos comprados en Ecuador y elementos importados desde China

$$A_c = \left(1 - \frac{C_{CH}}{C_{ECU}}\right) \times 100$$

Ecuación 5.1. Cálculo del ahorro económico importando de China

donde:

A_c = porcentaje de ahorro económico

C_{CH} = costo con productos importados

C_{ECU} = costo con productos comprados en Ecuador

$$A_c = \left(1 - \frac{59.41}{108.64}\right) \times 100 = 45.31\%$$

Ecuación 5.2. Porcentaje de ahorro económico importando de China

5.3.3 Costos de materiales del sistema de telegestión para la propuesta 1 con el prototipo desarrollado

Los costos del sistema de telegestión para la propuesta 1 con precios del mercado electrónico ecuatoriano son:

Tabla 5.12. Listado de equipos para la propuesta 1 con el prototipo de este proyecto

| Item | Descripción | Precio Unitario | Cant. | Precio Total * USD | Precio Total ** USD |
|---|----------------------------|-----------------|-------|--------------------|---------------------|
| 1 | Equipo controlador de Nodo | \$ 118,64 | 71 | \$ 8.423,44 | \$ 4.606,78 |
| 2 | Controlador de segmento | \$ 141,64 | 1 | \$ 141,64 | \$ 77,46 |
| 4 | Software - WebBrowser | \$ 100,00 | 1 | \$ 100,00 | \$ 100,00 |
| Total de costos del PROTOTIPO DESARROLLADO | | | | \$ 8.665,08 | \$ 4.784,24 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Tabla 5.13. Precios de materiales e instalación para la propuesta 1 con el prototipo de este proyecto

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|--------------------|
| 1 | Caja para tablero de control | \$ 10,00 | 1 | \$ 10,00 |
| 2 | Materiales para instalación | \$ 50,00 | 1 | \$ 50,00 |
| 3 | Asesoría y servicios para implementación completa. | \$ 0,00 | 1 | \$ 0,00 |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 71 | \$ 1.491,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 2 | \$ 42,00 |
| Total de costos de materiales e instalación | | | | \$ 1.593,00 |

El costo total de esta propuesta con productos comprados en Ecuador es US\$ 10.258,08.

El costo total de esta propuesta con productos importados desde China es US\$ 6.377,24.

5.3.4 Costos de materiales del sistema de telegestión para la propuesta 2 con el prototipo desarrollado

Los costos del sistema de telegestión para la propuesta 1 con precios del mercado electrónico ecuatoriano son:

Tabla 5.14. Listado de equipos para la propuesta 1 con el prototipo de este proyecto

| Item | Descripción | Precio Unitario | Cant. | Precio Total * USD | Precio Total ** USD |
|---|----------------------------|-----------------|-------|---------------------|---------------------|
| 1 | Equipo controlador de Nodo | \$ 118,64 | 229 | \$ 27.168,56 | \$ 14.858,49 |
| 2 | Controlador de segmento | \$ 141,64 | 1 | \$ 141,64 | \$ 77,46 |
| 4 | Software - WebBrowser | \$ 100,00 | 1 | \$ 100,00 | \$ 100,00 |
| Total de costos del PROTOTIPO DESARROLLADO | | | | \$ 27.410,20 | \$ 15.035,95 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Tabla 5.15. Precios de materiales e instalación para la propuesta 1 con el prototipo de este proyecto

| Item | Descripción | Precio Unitario | Cantidad | Precio Total USD |
|--|--|-----------------|----------|--------------------|
| 1 | Caja para tablero de control | \$ 10,00 | 1 | \$ 10,00 |
| 2 | Materiales para instalación | \$ 50,00 | 1 | \$ 50,00 |
| 3 | Asesoría y servicios para implementación completa. | \$ 0,00 | 1 | \$ 0,00 |
| 4 | Montaje del control de la luminaria | \$ 21,00 | 229 | \$ 4.809,00 |
| 5 | Montaje de tablero principal | \$ 21,00 | 2 | \$ 42,00 |
| Total de costos de materiales e instalación | | | | \$ 4.911,00 |

El costo total de esta propuesta con productos comprados en Ecuador es US\$ 32.321,20.

El costo total de esta propuesta con productos importados desde China es US\$ 19.946,95.

5.4 Análisis económico de energía

Según la Agencia de Regulación y Control de Electricidad, con datos actualizados hasta 2015, el costo medio de la energía por kWh de la Empresa Eléctrica Regional Norte es de 9,51 ¢USD en el año 2015 (Dato que se encuentra en el **ANEXO VII**).

Este precio será el parámetro de relación para calcular los consumos de energía de los casos analizados en el Capítulo 4.

5.4.1 Costos de energía con el sistema actual de alumbrado público

Centro de Ibarra – Situación Actual

El consumo de la energía bajo la modalidad de uso actual del sistema de alumbrado público es: $E = 4333.82 \text{ kWh/mes}$

Con este dato, el costo de la energía mensual para las 72 luminarias es:

$$C_E = E \times 0.0951 = 4333.82 \times 0.0951 = 412.15 \text{ \$/mes}$$

Ecuación 5.3. Costo de la energía en iluminación del centro de la ciudad bajo la situación actual

Av. Camilo Ponce y Galo Plaza – Situación Actual

El consumo de la energía bajo la modalidad de uso actual del sistema de alumbrado público es: $E = 10907.52 \text{ kWh/mes}$

Con este dato, el costo de la energía mensual para 230 luminarias es:

$$C_E = E \times 0.0951 = 10907.52 \times 0.0951 = 1037.31 \text{ \$/mes}$$

Ecuación 5.4. Costo de la energía en iluminación de la Av. Camilo Ponce y Galo Plaza bajo la situación actual

5.4.2 Costos de energía con la propuesta 1

El consumo de la energía en alumbrado público para la propuesta 1 con la aplicación de un sistema de telegestión es: $E_1 = 3010.59 \text{ kWh/mes}$

Con este dato, el costo de la energía mensual para las 72 luminarias es:

$$C_{E_1} = E_1 \times 0.0951 = 3010.59 \times 0.0951 = 286.21 \text{ \$/mes}$$

Ecuación 5.5. Costo de la energía para la propuesta 1

$$A_{C_{E_1}} = \left(1 - \frac{C_{E_1}}{C_E}\right) \times 100$$

Ecuación 5.6. Cálculo del ahorro económico para la propuesta 1

donde:

$A_{C_{E_1}}$ = porcentaje de ahorro económico con la propuesta 1

C_{E_1} = costo de la energía para la propuesta 1

C_E = costo de la energía con la situación actual

$$A_{C_{E_1}} = \left(1 - \frac{286.31}{412.15}\right) \times 100 = 30.53\%$$

Ecuación 5.7. Porcentaje de ahorro económico con la propuesta 1

5.4.3 Costos de energía con la propuesta 2

El consumo de la energía en alumbrado público para la propuesta 2 con la aplicación de un sistema de telegestión es: $E_2 = 7093.70 \text{ kWh/mes}$

Con este dato, el costo de la energía mensual para 230 luminarias es:

$$C_{E_2} = E_2 \times 0.0951 = 7093.70 \times 0.0951 = 674.61 \text{ \$/mes}$$

Ecuación 5.8. Costo de la energía para la propuesta 2

$$A_{C_{E_2}} = \left(1 - \frac{C_{E_2}}{C_E}\right) \times 100$$

Ecuación 5.9. Cálculo del ahorro económico para la propuesta 2

donde:

$A_{C_{E_2}}$ = porcentaje de ahorro económico con la propuesta 2

C_{E_2} = costo de la energía para la propuesta 2

C_E = costo de la energía con la situación actual

$$A_{C_{E_2}} = \left(1 - \frac{674.61}{1037.31}\right) \times 100 = 34.97\%$$

Ecuación 5.10. Porcentaje de ahorro económico con la propuesta 2

En la **Tabla 5.16** se muestra el resumen de cálculos económicos para las dos propuestas.

Tabla 5.16. Resumen del Análisis Económico del Sistema de Telegestión de Alumbrado Público

| Análisis Económico del Sistema de Telegestión de Alumbrado Público | | |
|--|-----------------|------------------|
| | Propuesta 1 | Propuesta 2 |
| Costo de Energía al 100% | 412,15 [\$/mes] | 1037,31 [\$/mes] |
| Costo de Energía con Telegestión | 286,21 [\$/mes] | 674,61 [\$/mes] |
| Ahorro Económico | 30,53 [%] | 34,97 [%] |

5.5 Retorno de inversión de la implementación de sistemas de telegestiones

El pliego tarifario del servicio de alumbrado público es entregado a las empresas distribuidoras por la Agencia de Regulación y Control de Electricidad (ARCONEL), para el cálculo del retorno de la inversión de cada alternativa de telegestión se usará el último valor proporcionado en informes por el ARCONEL.

La **Tabla 5.17** se muestra el resumen de los costos de inversión para las dos propuestas, la **Tabla 5.18** muestra los costos por mantenimiento y operación y la **Tabla 5.19** indica el dinero ahorrado mensualmente en US\$.

Tabla 5.17. Costo de inversión de cada alternativa

| Inversión | | |
|---------------------|--------------|---------------|
| | Propuesta 1 | Propuesta 2 |
| Schröder | \$ 82.638,00 | \$ 113.353,00 |
| Philips | \$ 72.320,00 | \$ 104.933,00 |
| * Prototipo | \$ 10.258,08 | \$ 32.321,20 |
| ** Prototipo | \$ 6.377,24 | \$ 19.946,95 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Se asume un ahorro en mantenimiento y operación del 30% para fines de cálculo de retorno de inversión.

Tabla 5.18. Costo de Mantenimiento y Operación de alumbrado público mensual

| Costo M&O anual | | |
|--------------------|-------------|--------------|
| | Propuesta 1 | Propuesta 2 |
| Actual | \$ 5.040,00 | \$ 16.100,00 |
| Telegestión | \$ 3.528,00 | \$ 11.270,00 |

Tabla 5.19. Ahorro energético mensual con sistemas de telegestión

| Ahorro Energético | |
|--------------------|-----------|
| Propuesta 1 | \$ 125,84 |
| Propuesta 2 | \$ 362,69 |

En la **Tabla 5.20** y **Tabla 5.21** se realiza el análisis de la recuperación de la inversión de cada alternativa analizada anteriormente para las propuestas 1 y 2 respectivamente.

Tabla 5.20. Recuperación de la inversión de cada alternativa para la propuesta 1

| Recuperación de Inversión para la Propuesta 1 | | | | |
|---|---------------|---------------|--------------|--------------|
| Año | Schröder | Philips | * Prototipo | ** Prototipo |
| 1 | \$ -79.615,94 | \$ -69.297,94 | \$ -7.236,02 | \$ -3.355,18 |
| 2 | \$ -76.593,87 | \$ -66.275,87 | \$ -4.213,95 | \$ -333,11 |
| 3 | \$ -73.571,81 | \$ -63.253,81 | \$ -1.191,89 | \$ 2.688,95 |
| 4 | \$ -70.549,74 | \$ -60.231,74 | \$ 1.830,18 | \$ 5.711,02 |
| 5 | \$ -67.527,68 | \$ -57.209,68 | \$ 4.852,24 | \$ 8.733,08 |
| 6 | \$ -64.505,61 | \$ -54.187,61 | \$ 7.874,31 | \$ 11.755,14 |
| 7 | \$ -61.483,55 | \$ -51.165,55 | \$ 10.896,37 | \$ 14.777,21 |
| 8 | \$ -58.461,48 | \$ -48.143,48 | \$ 13.918,44 | \$ 17.799,27 |
| 9 | \$ -55.439,42 | \$ -45.121,42 | \$ 16.940,50 | \$ 20.821,34 |
| 10 | \$ -52.417,36 | \$ -42.099,36 | \$ 19.962,56 | \$ 23.843,40 |
| 11 | \$ -49.395,29 | \$ -39.077,29 | \$ 22.984,63 | \$ 26.865,47 |
| 12 | \$ -46.373,23 | \$ -36.055,23 | \$ 26.006,69 | \$ 29.887,53 |
| 13 | \$ -43.351,16 | \$ -33.033,16 | \$ 29.028,76 | \$ 32.909,60 |
| 14 | \$ -40.329,10 | \$ -30.011,10 | \$ 32.050,82 | \$ 35.931,66 |
| 15 | \$ -37.307,03 | \$ -26.989,03 | \$ 35.072,89 | \$ 38.953,72 |
| 16 | \$ -34.284,97 | \$ -23.966,97 | \$ 38.094,95 | \$ 41.975,79 |
| 17 | \$ -31.262,90 | \$ -20.944,90 | \$ 41.117,02 | \$ 44.997,85 |
| 18 | \$ -28.240,84 | \$ -17.922,84 | \$ 44.139,08 | \$ 48.019,92 |
| 19 | \$ -25.218,78 | \$ -14.900,78 | \$ 47.161,14 | \$ 51.041,98 |
| 20 | \$ -22.196,71 | \$ -11.878,71 | \$ 50.183,21 | \$ 54.064,05 |
| 21 | \$ -19.174,65 | \$ -8.856,65 | \$ 53.205,27 | \$ 57.086,11 |
| 22 | \$ -16.152,58 | \$ -5.834,58 | \$ 56.227,34 | \$ 60.108,18 |
| 23 | \$ -13.130,52 | \$ -2.812,52 | \$ 59.249,40 | \$ 63.130,24 |
| 24 | \$ -10.108,45 | \$ 209,55 | \$ 62.271,47 | \$ 66.152,30 |
| 25 | \$ -7.086,39 | \$ 3.231,61 | \$ 65.293,53 | \$ 69.174,37 |
| 26 | \$ -4.064,32 | \$ 6.253,68 | \$ 68.315,60 | \$ 72.196,43 |
| 27 | \$ -1.042,26 | \$ 9.275,74 | \$ 71.337,66 | \$ 75.218,50 |
| 28 | \$ 1.979,80 | \$ 12.297,80 | \$ 74.359,72 | \$ 78.240,56 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Tabla 5.21. Recuperación de la inversión de cada alternativa para la propuesta 2

| Recuperación de Inversión para la Propuesta 2 | | | | |
|---|----------------|---------------|---------------|---------------|
| Año | Schröder | Philips | * Prototipo | ** Prototipo |
| 1 | \$ -107.488,67 | \$ -99.068,67 | \$ -26.456,87 | \$ -14.082,62 |
| 2 | \$ -101.624,34 | \$ -93.204,34 | \$ -20.592,54 | \$ -8.218,29 |
| 3 | \$ -95.760,01 | \$ -87.340,01 | \$ -14.728,21 | \$ -2.353,96 |
| 4 | \$ -89.895,68 | \$ -81.475,68 | \$ -8.863,88 | \$ 3.510,37 |
| 5 | \$ -84.031,35 | \$ -75.611,35 | \$ -2.999,55 | \$ 9.374,70 |
| 6 | \$ -78.167,02 | \$ -69.747,02 | \$ 2.864,78 | \$ 15.239,03 |
| 7 | \$ -72.302,69 | \$ -63.882,69 | \$ 8.729,11 | \$ 21.103,36 |
| 8 | \$ -66.438,36 | \$ -58.018,36 | \$ 14.593,44 | \$ 26.967,69 |
| 9 | \$ -60.574,03 | \$ -52.154,03 | \$ 20.457,77 | \$ 32.832,02 |
| 10 | \$ -54.709,70 | \$ -46.289,70 | \$ 26.322,10 | \$ 38.696,35 |
| 11 | \$ -48.845,37 | \$ -40.425,37 | \$ 32.186,43 | \$ 44.560,68 |
| 12 | \$ -42.981,04 | \$ -34.561,04 | \$ 38.050,76 | \$ 50.425,01 |
| 13 | \$ -37.116,71 | \$ -28.696,71 | \$ 43.915,09 | \$ 56.289,34 |
| 14 | \$ -31.252,38 | \$ -22.832,38 | \$ 49.779,42 | \$ 62.153,67 |
| 15 | \$ -25.388,05 | \$ -16.968,05 | \$ 55.643,75 | \$ 68.018,00 |
| 16 | \$ -19.523,72 | \$ -11.103,72 | \$ 61.508,08 | \$ 73.882,33 |
| 17 | \$ -13.659,39 | \$ -5.239,39 | \$ 67.372,41 | \$ 79.746,66 |
| 18 | \$ -7.795,06 | \$ 624,94 | \$ 73.236,74 | \$ 85.610,99 |
| 19 | \$ -1.930,73 | \$ 6.489,27 | \$ 79.101,07 | \$ 91.475,32 |
| 20 | \$ 3.933,59 | \$ 12.353,59 | \$ 84.965,39 | \$ 97.339,65 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

5.6 Cálculo de VAN y TIR

En esta sección se analizará la rentabilidad del proyecto y su viabilidad. La rentabilidad del proyecto debe ser mayor o al menos igual a realizar una inversión de poco riesgo con entidades financieras solventes, caso contrario es mejor invertir en las posibilidades de bajo riesgo en lugar de desarrollar el proyecto. *“Lo que mueve este mundo siempre será el dinero”.*

Para el cálculo de VAN y TIR se considerará las dos propuestas con las cuatro posibilidades de inversión, utilizando la tasa de interés para Inversión Pública del mes de febrero de 2018 establecida por el Banco Central del Ecuador [21].

Para realizar el cálculo de manera automática se utilizará una hoja de cálculo en Excel, que permite calcular el VAN y TIR con funciones preestablecidas.

Datos para su cálculo

Los costos de inversión de las alternativas para cada propuesta se encuentran en la **Tabla 5.17**.

El tiempo de vida de un driver electrónico en general oscila entre las 50000 y 100000 horas de uso, para nuestro análisis consideraremos como tiempo máximo 50000 horas, equivalente a 11.41 años para el caso de alumbrado público. Para este análisis se asumirá un tiempo de vida útil de los dispositivos controladores de nodo de 10 años para cada alternativa.

Tiempo de análisis = N = 10 años

Flujo Neto al año n = f_n = \$ 3.022,06 (Valor por concepto de ahorro en mantenimiento y consumo energético)

Tasa de interés = i = 8,02%

El valor del VAN se puede calcular con la siguiente ecuación:

$$VAN = -I_0 + \sum_{n=1}^N \frac{f_n}{(1+i)^n}$$

Ecuación 5.11. Expresión para calcular el VAN

El valor de TIR, se calcula cuando el VAN = 0:

$$-I_0 + \sum_{n=1}^N \frac{f_n}{(1+TIR)^n} = 0$$

Ecuación 5.12. Expresión para calcular el TIR

Tabla 5.22. Flujo de Fondos para el cálculo de TIR y VAN para decisión de inversión de la Propuesta 1

| Proyecto para Evaluar | Schröder | Philips | * Prototipo | ** Prototipo |
|-----------------------|------------------------|---------------------|---------------------|--------------------|
| Tasa de descuento | 8,02% | 8,02% | 8,02% | 8,02% |
| <i>Período</i> | <i>Flujo de Fondos</i> | | | |
| 0 | -\$82.638,00 | -\$72.320,00 | -\$10.258,08 | -\$6.377,24 |
| 1 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 2 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 3 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 4 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 5 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 6 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 7 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 8 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 9 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |
| 10 | \$3.022,06 | \$3.022,06 | \$3.022,06 | \$3.022,06 |

Tabla 5.23. Cálculo de VAN y TIR para la Propuesta 1

| Cálculo de VAN y TIR | | | | | | | |
|----------------------|----------------------|---------|----------------------|-------------|--------------|--------------|--------------|
| Schröder | | Philips | | * Prototipo | | ** Prototipo | |
| TIR | -15,07% | TIR | -13,37% | TIR | 26,70% | TIR | 46,34% |
| VAN | \$ -62.378,01 | VAN | \$ -52.060,01 | VAN | \$ 10.001,91 | VAN | \$ 13.882,75 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Decisiones entre proyectos para la propuesta 1

- No conviene hacer las opciones de Schröder y Philips dado que ofrece un retorno menor al del mercado, comparado con estas opciones es mejor optar por una posibilidad de bajo riesgo.
- Conviene realizar las opciones de * Prototipo y ** Prototipo dado que ofrece un retorno mayor al mercado.
- Entre las dos opciones posibles, la de mayor productividad es la opción ** Prototipo, por lo que se recomienda optar por importar los dispositivos desde China.

Tabla 5.24. Flujo de Fondos para el cálculo de TIR y VAN para decisión de inversión de la Propuesta 2

| Proyecto para Evaluar | Schröder | Philips | * Prototipo | ** Prototipo |
|-----------------------|------------------------|---------------|--------------|--------------|
| Tasa de descuento | 8,02% | 8,02% | 8,02% | 8,02% |
| <i>Período</i> | <i>Flujo de Fondos</i> | | | |
| 0 | -\$113.353,00 | -\$104.933,00 | -\$32.321,20 | -\$19.946,95 |
| 1 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 2 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 3 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 4 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 5 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 6 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 7 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 8 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 9 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |
| 10 | \$5.864,33 | \$5.864,33 | \$5.864,33 | \$5.864,33 |

Tabla 5.25. Cálculo de VAN y TIR para la Propuesta 2

| Cálculo de VAN y TIR | | | | | | | |
|----------------------|---------------|---------|---------------|-------------|-------------|--------------|--------------|
| Schröder | | Philips | | * Prototipo | | ** Prototipo | |
| TIR | -10,48% | TIR | -9,39% | TIR | 12,61% | TIR | 26,63% |
| VAN | \$ -74.038,34 | VAN | \$ -65.618,34 | VAN | \$ 6.993,46 | VAN | \$ 19.367,71 |

* Precios con productos comprados en Ecuador

** Precios con productos importados desde China

Decisiones entre proyectos para la propuesta 2

- No conviene hacer las opciones de Schröder, Philips y * Prototipo dado que ofrece un retorno menor al del mercado, comparado con estas opciones es mejor optar por una posibilidad de bajo riesgo.
- Como en el caso anterior, conviene realizar las opciones de * Prototipo y ** Prototipo dado que ofrece un retorno mayor al mercado.
- Entre las dos opciones posibles, la de mayor productividad, como era de esperarse sigue siendo la opción ** Prototipo, por lo que se recomienda optar por importar los dispositivos desde China.

CAPÍTULO 6.

6. ANÁLISIS DE RESULTADOS

Este prototipo de sistema de alumbrado público se presenta como un equipo que brinda la posibilidad de obtener ahorro energético y una herramienta de trabajo que facilita la operación y monitoreo de las luminarias de alumbrado público para los responsables del centro de control y alumbrado público de EMELNORTE S.A.

Con el uso de este tipo de tecnología se conseguirá:

- Mejorar la gestión de iluminación en instalaciones de alumbrado público.
- Incrementar el ahorro energético en iluminación vial.
- Mejorar el proceso de mantenimiento de las luminarias de alumbrado, disminuyendo el trabajo de campo y minimizando el uso de los recursos humanos necesarios para esta labor.
- Brindar una mejor calidad de servicio de iluminación al ciudadano, evitando falta de iluminación por fallos en luminarias y accidentes de tránsito.
- Proporcionar las bases de una comunicación urbana inalámbrica propia para futuros avances tecnológicos tales como instalación de cámaras de seguridad, sensores de flujo de tráfico, radares de velocidad, detectores de campo magnético en alimentadores, actuadores de re-conectores, etc.
- Maximizar el uso de las instalaciones de iluminación al tener el control manual o automático de todos los equipos de alumbrado público.

El sistema de telegestión brindará funciones al alumbrado público tales como:

- Ahorro energético en el consumo de iluminación, adaptándose a los perfiles de iluminación de diversas estrategias con el fin de minimizar los recursos sin sacrificar la calidad del servicio.

- Control y gestión de alumbrado, ordenando el encendido y apagado de cada zona de acuerdo con las necesidades, programándolas de manera automática o manual, según lo requiera cada vía.
- Supervisión y manipulación en tiempo real de las luminarias de alumbrado público.
- Llevar un registro del tiempo de uso de cada luminaria para garantizar un trabajo adecuado de mantenimiento y reemplazo cuando lleguen al final de su vida útil.

Los niveles de iluminación propuestos se analizaron en base al flujo vial de cada caso, dichos parámetros quedan abiertos a cambios futuros que considere adecuados el Área de Alumbrado Público de la Empresa Eléctrica EMELNORTE.

Para la parte experimental de este proyecto no se pudo conseguir un medidor de energía con precisión decimal, pese a ello, las medidas tomadas en las lámparas de prueba se aproximan a los valores calculados, es por esto por lo que el análisis técnico – económico se realizó con dichos cálculos teóricos.

En la determinación de los niveles de iluminación para las dos propuestas, la razón fundamental de decisión es la carencia de vida nocturna en la ciudad, al ser una ciudad pequeña y considerada como peligrosa debido al incremento de la delincuencia, las personas prefieren no salir a las calles pasadas las 21:00, como se logró comprobar durante las inspecciones nocturnas, aunque este aspecto desfavorece al comercio de la ciudad, es beneficioso para el propósito de un sistema de telegestión.

En la propuesta 1, la decisión de que los niveles de iluminación sean 100, 70 y 40% fue influenciada por la ubicación central de esta zona, pese al poco flujo de usuarios viales hay que considerar que al ser una zona comercial y residencial necesita de buena iluminación.

Mientras que en la propuesta 2, debido a la lejanía del parque y la falta de transporte hasta el mismo es un punto relevante en la toma de decisión, en las inspecciones nocturnas se pudo percatar que aproximadamente solo el 35% del parque es mayormente visitado, desde el redondel de la madre hasta el edificio del ECU911, en el resto del parque se puede observar que mientras transcurre el inicio de la noche la cantidad de visitantes disminuye. Además, el parque dispone de iluminación LED excesiva, que por sí sola llega a iluminar

las vías mencionadas en esta propuesta, es por esto por lo que los niveles de iluminación de las luminarias de las vías pueden ser reducidas hasta un 30%.

Para la propuesta 1 se obtendrá un 30.53% de ahorro energético, mientras que en la propuesta 2 se llegará a obtener el 34.97%, estos valores pueden aumentar considerando que durante los últimos meses de 2017 y lo que va del 2018, las 34 luminarias del tercer bloque al fondo del parque se mantuvieron apagadas algunos días en la madrugada.

Se puede observar una gran diferencia entre los costos de inversión en la instalación de productos de Schröder, Philips y el prototipo desarrollado con precios ecuatorianos y de importación, como se puede observar en el resumen de costos de la **Tabla 5.17**.

Como se puede notar, la diferencia de precios entre los sistemas comerciales y el prototipo desarrollado es bastante clara, aunque es necesario recalcar que no se puede comparar un sistema en versión prototipo con un sistema comercial perfeccionado durante varios años, es importante mencionar que el desarrollo y experimentación de un sistema propio contribuye al avance tecnológico, tanto de EMELNORTE S.A. como del país. Se espera que a futuro se continúe con el desarrollo de este prototipo para llegar a tener un sistema de telegestión 100% funcional y a prueba de errores.

En la **Tabla 5.20** y **Tabla 5.21** podemos observar que para comenzar a ver resultados positivos con la implementación de un sistema de telegestión comercial tendrán que pasar más de 23 años para la propuesta 1 y más de 17 años para la propuesta 2, mientras que con el prototipo desarrollado se podrá ver ahorros a partir del segundo y tercer año respectivamente. Gracias a que las funciones de ahorro energético que aplica el equipo de control en las luminarias no dependen de la tecnología utilizada en los equipos, sino que mientras se garantice el control de los niveles de iluminación el ahorro será el mismo. Además de que, el mayor costo en los sistemas comerciales es el software de monitoreo a utilizarse, que en el caso del prototipo se puede elaborar con un poco de estudio de programación y comunicaciones, reduciendo considerablemente el costo total del sistema, a esto añadámosle el costo de instalación que en el caso de hacerlo por trabajadores de la empresa para el prototipo ayudaría a la disminución del costo total.

El cálculo del VAN y TIR demuestran la rentabilidad del prototipo desarrollado frente a sistemas comerciales, además de ayudar con el avance tecnológico del país. Cabe mencionar que en ambos casos es mejor importar los elementos electrónicos desde China, para obtener mejor rentabilidad.

CAPÍTULO 7.

7. CONCLUSIONES Y RECOMENDACIONES

- La tecnología LED es más costosa que las tecnologías usadas en el pasado, aunque a largo plazo representan un ahorro frente a las otras, además de que brindan una gestión de regulación de los niveles de iluminación completa adaptable a cada caso de estudio, según las necesidades del lugar.
- La tecnología usada en este proyecto es adaptable a cualquier luminaria LED que tenga características de dimerización por control de voltaje analizadas en este trabajo, ventaja frente a sistemas comerciales que requieren de sus propios equipos para funcionar correctamente. Además, el controlador desarrollado puede ser adaptable a otro tipo de lámparas, no solo LED, con la aplicación de periféricos adecuados a cada tecnología.
- Para el ahorro energético se puede llegar a considerar un ahorro de incluso más del 50% si se llegara a considerar apagar cierta cantidad de lámparas cada determinada distancia en línea recta y en lugares donde no se requiere iluminación directa para momentos donde el flujo de usuarios viales sea nulo.
- El ahorro en mantenimiento de luminarias se debe al incremento de la vida útil de las luminarias con el control de estas, además de tener un monitoreo en tiempo real evitando inspecciones innecesarias por parte del personal de alumbrado público.
- Desde el centro de control de la empresa se podrá recibir datos del funcionamiento en tiempo real de cada luminaria, proveyendo la información adecuada al funcionamiento irracional de las luminarias, como por ejemplo la activación de luminarias en horario diurno. Además de que el software es completamente variable a conveniencia y necesidades futuras de la empresa.
- Los dispositivos controladores de nodo no requieren de un gran tamaño de almacenamiento de datos, ya que todos estos son enviados inalámbricamente al centro de monitoreo.
- La fase de experimentación no tuvo la satisfacción deseada debido a la falta de luminarias existentes y previstas por la empresa para este propósito, los trámites administrativos fueron un inconveniente a la hora de ejecutar el desarrollo de este

proyecto, pese a eso se logró realizar gran parte de las actividades planificadas en el inicio de la tesis.

- Las comunicaciones inalámbricas de este proyecto fueron un verdadero pero agradable y satisfactorio reto, desarrollar esta tesis fue un proceso de aprendizaje de nuevos conocimientos que servirán para desarrollar tecnología propia en pro de emprender un negocio de eficiencia energética y automatización de sistemas de iluminación, razón principal por la que en un principio fue aceptada la ejecución de este gran proyecto.
- El aplicar conocimientos propios en el desarrollo de tecnología que ayuda a la eficiencia energética y automatización de procesos, especialmente en iluminación es para mí totalmente satisfactorio, viendo que este es un tema que me ha apasionado durante toda mi carrera universitaria.
- Para la implementación del sistema de telegestión con el prototipo desarrollado incluyendo comunicación GSM/GPRS se recomienda mantener el control del plan de datos contratado, ya que el volumen del tráfico que tendrá que manejar el controlador de segmento aumenta con el incremento de la cantidad de luminarias. Realizar pruebas de volumen de tráfico antes de contratar un plan de datos a una operadora telefónica.
- Una vez instalados los sistemas de comunicación se puede mejorar el sistema implementando sensores de velocidad, detectores de movimiento y colaborar en conjunto con el ECU911 colocando cámaras de seguridad y equipos de seguridad para mejorar el servicio hacia los usuarios viales. La tecnología actual en seguridad es demasiado amplia, y solo necesita de un método de comunicación para funcionar, y que mejor que utilizar el mismo sistema de comunicación para varios fines.
- Se recomienda no dejar este proyecto en teoría, la mejora de este contribuirá a la expansión y condecoración del nivel de desarrollo tecnológico de la empresa y del país, dejando de ser únicamente consumidores y pasando a ser desarrolladores de tecnología.
- Para la implementación a gran escala se recomienda utilizar elementos de importación, de esta manera se obtiene la maximización total en los costos de los equipos, que como se pudo ver en la tabla de recuperación de la inversión es la alternativa con mayores rentabilidades entre las opciones analizadas.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] P. Carvajal y A. Orbe, Balance Energético Nacional 2013, Ministerio Coordinador de Sectores Estratégicos, 2013.
- [2] ANT, Estadísticas de Transporte Terrestre y Seguridad Vial, Siniestros diciembre 2016, Quito: Agencia Nacional de Tránsito, 2016.
- [3] Schröder, Curso de Iluminación LED - CAPITULO 1: Introducción a la Iluminación LED, Cochabamba, 2017.
- [4] D. Gallegos, «El Ciudadano,» [En línea]. Available: <http://www.elciudadano.gob.ec/ibarra-se-empodera-del-espacio-publico-con-el-parque-centrica-bulevar/>.
- [5] «LEDS International,» [En línea]. Available: <http://www.ledsinternational.com/es/iluminacion-para-vitrinas/>.
- [6] Schröder, Curso de Iluminación LED - CAPITULO 2: Sistemas de Alumbrado LED, Cochabamba, 2017.
- [7] G. E. Company, «Current powered by GE,» [En línea]. Available: <http://solutions.currentbyge.com/LightingWeb/la/north/productos/iluminacion-interior/luminarias-empotradas/luminacion-serie-et.jsp>.
- [8] S. Halleux, CATÁLOGO VOLTANA - ILUMINACIÓN LED PARA CUALQUIER APLICACIÓN, Schröder.
- [9] AIN, «Contrainjerencia,» 30 08 2014. [En línea]. Available: <http://www.contrainjerencia.com/?p=92545>.
- [10] «Solo Stocks,» [En línea]. Available: <http://www.solostocks.com/venta-productos/iluminacion-led/faros-led/modulo-led-40w-para-adaptacion-de-farol-villa-fernandino-9511406>.
- [11] Schröder, Curso de Iluminación LED - CAPITULO 3: Drivers y Control de Alumbrado, Cochabamba, 2017.
- [12] P. L. H. B.V., Datasheet Xitanium 150W 0.35-0.70A GL Prog sXt, 2017.
- [13] Schröder, Curso de Iluminación LED - CAPITULO 4: Alumbrado Exterior, Cochabamba, 2017.
- [14] MEER, Especificaciones Técnicas de Materiales y Equipos del Sistema de Distribución - Luminarias Tipo LED, Primera ed., 2014.
- [15] CONELEC, Regulación N° CONELEC 008/11, 2011.

- [16 Wikispaces, «Xbee,» [En línea]. Available: <https://xbee.wikispaces.com>.
]
- [17 A. Raúl, Redes Inalámbricas ZigBee de Sensores con Módulos Xbee - Teoría Introductoria,
] Cochabamba: TecBolivia, 2013.
- [18 Ingeniería del Tráfico I. Parámetros Fundamentales, Madrid: Departamento de Ingeniería
] Mecánica - Universidad Carlos III.
- [19 L. Chabla y D. Córdova, EFICIENCIA ENERGÉTICA EN EL ALUMBRADO PÚBLICO DEL CENTRO
] HISTÓRICO DE CUENCA: TELEGESTIÓN Y SUSTITUCIÓN DE LUMINARIAS, Cuenca, 2015.
- [20 Schröder, Curso de Iluminación LED - CAPITULO 5: Telegestión, Cochabamba, 2017.
]
- [21 B. C. d. Ecuador, «Tasas de Interés - Banco Central del Ecuador,» [En línea]. Available:
] <https://contenido.bce.fin.ec/docs.php?path=/documentos/Estadisticas/SectorMonFin/TasasInteres/Indexe.htm>.

9. ANEXOS

ANEXO I

El código de programación del Arduino de cada nodo es el siguiente:

```
// Declaracion de constantes y variables (Variables Globales)
int sensorPin = A0;           // Entrada del sensor de corriente en pin 5
int switchPin = 4;           // Salida digital para Rele en pin 4
float pwmPin = 6;            // Salida PWM (0-5V) para control de
intensidad luminosa

char sensorLED = 0;          // Valor de estado del sensor para enviar via
serial (digital)
float sensibilidad = 0.168; // Sensibilidad en V/A calibrada al sensor de
efecto hall
float offset = 0.1;          // Amplitud del valor del ruido (error)
float sensorValor;          // Valor del sensor analogo (0-1023)
float voltaje;              // Amplitud de voltaje (0V-5V) obtenido desde
el sensor
float corriente = 0;         // Valor de la corriente inicializada en cero
float Imax = 0;             // Valor de la corriente maxima inicializada
en cero
float Imin = 0;             // Valor de la corriente minima inicializada
en cero
float Irms = 0;             // Valor de la corriente rms
long tiempoActual = 0;      // Inicializacion de la variable de tiempo
char ledOn = 'q';           // Valor ascii a enviar correspondiente al
estado de encendido de la lampara
char ledOff = 'w';          // Valor ascii a enviar correspondiente al
estado de apagado de la lampara

char switchOn = 'e';        // Valor ascii a recibir correspondiente al
estado de encendido del rele normalmente cerrado (APAGADO DE LAMPARA)
char switchOff = 'r';       // Valor ascii a recibir correspondiente al
estado de apagado del rele normalmente cerrado (ENCENDIDO DE LAMPARA)
int a = 0;                  // Variable bandera de rele (1 -> rele
abierto (lampara apagada))
int b = 10;                 // Variable bandera de pwm (10=100%, 9=90%,
8=80%, 7=70%, 6=60%, 5=50%, 4=40%, 3=30%, 2=20%, 1=10%)
int c = 0;                  // Variable bandera que verifica si se
produce un cambio en la pwm generada
int d = 1;                  // Variable bandera del sensor (1 -> sensor
activado (lampara apagada))
int e = 2;                  // Variable bandera que permite el envio del
dato del sensor solo cuando este cambie de posicion

char datoRecibido = 0;      // Dato recibido por el puerto serial

// Porcentajes de ancho de pulso PWM (10% a 100%)
char pwm100 = 'a';         // PWM al 100%
char pwm90 = 's';          // PWM al 90%
char pwm80 = 'd';          // PWM al 80%
char pwm70 = 'f';          // PWM al 70%
```

```

char pwm60 = 'g';           // PWM al 60%
char pwm50 = 'h';           // PWM al 50%
char pwm40 = 'j';           // PWM al 40%
char pwm30 = 'k';           // PWM al 30%
char pwm20 = 'l';           // PWM al 60%
char pwm10 = 'm';           // PWM al 10%

char pwmEnvio = 'a';        // Valor de confirmacion de cambio de
intensidad luminosa
float pwmOut = 0;           // Valor de PWM por defecto para sacar senal
de voltaje por el puerto PWM6

float tiempo = 0;           // Inicializacion de la variable de tiempo
para funcion envioSensor
float tiempoEnvio = 10000; // Intervalo de tiempo de envio de datos por
el puerto serial

void setup() {
  Serial.begin(9600);        // Abrir puerto serial a 9600 bps

  // Declaracion de pines de entrada y salida
  pinMode(sensorPin, INPUT); // Sensor de corriente
  pinMode(switchPin, OUTPUT); // Activador del Rele en fase de
alimentacion
  pinMode(pwmPin, OUTPUT);   // Salida PWM para control de mosfet

  // Ejecuciones iniciales
  digitalWrite(switchPin, HIGH); // Iniciar con la lampara encendida
(Rele NC)
  analogWrite(pwmPin, pwmOut);   // Despliega el valor de pwm
correspondiente por default al inicio del programa
  tiempo = tiempoEnvio;
}

void loop() {
  datoRecibido = recibirDatos(); // Llama a la funcion recibirDatos()
para verificar el estado del puerto serial
  switchRele(datoRecibido);       // Llama a la funcion switchRele(char
dato) para ejecutar el Rele (encendido y apagado de la lampara)
  pwmControl(datoRecibido);       // Llama a la funcion pwmControl(char
dato) para ejecutar el control de pwm
  sensorLED = lecturaSensor();    // Llama a la funcion lecturaSensor()
para verificar el estado del pin del sensor
  envioSensor(sensorLED);         // Llama a la funcion
envioSensor(char LED) para enviar via serial el valor de sensorLED
}

char recibirDatos()             // Recepcion de datos por el puerto
serial
{
  if(Serial.available() > 0){    // Verifica si el puerto serial
contiene algun dato
    datoRecibido = Serial.read(); // Variable que almacena
momentaneamente el dato del puerto serial
  }
  return datoRecibido;           // Devuelve fuera de la funcion el
valor de datoSerial
  //delay(500);
}

```



```

}

void switchRele(char dato)          // Operacion del Rele
{
    if(dato == switchOn && a == 0){ // APAGADO DE LAMPARA (rele
normalmente cerrado)
        digitalWrite(switchPin,LOW);
        a=1;
    }
    if(dato == switchOff && a == 1){ // ENCENDIDO DE LAMPARA (rele
normalmente cerrado)
        digitalWrite(switchPin,HIGH);
        a=0;
    }
    //delay(100);
}

void pwmControl(char dato)          // Generacion de PWM
{
    c = b;
    // Generacion de PWM (0-5V)
    if(dato == pwm100 && b != 10){
        pwmOut = 255 * 0.0;          // PWM con valor de 255,0 equivalente
a 5,0V (100%)
        b = 10;
        pwmEnvio = 'a';
    }
    if(dato == pwm90 && b != 9){
        pwmOut = 255 * 0.1;          // PWM con valor de 229,5 equivalente
a 4,5V (90%)
        b = 9;
        pwmEnvio = 's';
    }
    if(dato == pwm80 && b != 8){
        pwmOut = 255 * 0.2;          // PWM con valor de 204,0 equivalente
a 4,0V (80%)
        b = 8;
        pwmEnvio = 'd';
    }
    if(dato == pwm70 && b != 7){
        pwmOut = 255 * 0.3;          // PWM con valor de 178,5 equivalente
a 3,5V (70%)
        b = 7;
        pwmEnvio = 'f';
    }
    if(dato == pwm60 && b != 6){
        pwmOut = 255 * 0.4;          // PWM con valor de 153,0 equivalente
a 3,0V (60%)
        b = 6;
        pwmEnvio = 'g';
    }
    if(dato == pwm50 && b != 5){
        pwmOut = 255 * 0.5;          // PWM con valor de 127,5 equivalente
a 2,5V (50%)
        b = 5;
        pwmEnvio = 'h';
    }
    if(dato == pwm40 && b != 4){

```

```

        pwmOut = 255 * 0.6;           // PWM con valor de 102,0 equivalente
a 2,0V (40%)
        b = 4;
        pwmEnvio = 'j';
    }
    if(dato == pwm30 && b != 3){
        pwmOut = 255 * 0.7;           // PWM con valor de 76,5 equivalente
a 1,5V (30%)
        b = 3;
        pwmEnvio = 'k';
    }
    if(dato == pwm20 && b != 2){
        pwmOut = 255 * 0.8;           // PWM con valor de 51,0 equivalente
a 1,0V (20%)
        b = 2;
        pwmEnvio = 'l';
    }
    if(dato == pwm10 && b != 1){
        pwmOut = 255 * 0.9;           // PWM con valor de 25,5 equivalente
a 0,5V (10%)
        b = 1;
        pwmEnvio = 'm';
    }
}
if(c != b){                           // Aplica cambios a la pwm solo si hay
la orden de hacerlo
    analogWrite(pwmPin,pwmOut);       // Despliega el valor de pwm
correspondiente
    Serial.print(pwmEnvio);
}
//delay(100);
}

char lecturaSensor()
{
    corriente = 0;                       // Iniciar funcion con corriente
en cero
    Imax = 0;                             // Iniciar funcion con corriente
maxima en cero
    Imin = 0;                             // Iniciar funcion con corriente
minima en cero
    Irms = 0;                             // Iniciar funcion con corriente
rms en cero
    tiempoActual = millis();              // Tiempo hasta este instante
    while(millis() - tiempoActual < 500){ // Recoleccion de mediciones
durante 0.5s
        sensorValor = analogRead(A0);
        voltaje = sensorValor * (5.0 / 1023.0); // Lectura del sensor
convertida a voltaje (0V-5V)
        corriente = 0.9 * corriente + 0.1 * ((voltaje - 2.486)/sensibilidad);
// Ecuación para obtener la corriente pico
        if(corriente > Imax){
            Imax = corriente;
        }
        if(corriente < Imin){
            Imin = corriente;
        }
    }
}
Irms = (((Imax - Imin) / 2) - offset) * 0.707;

```

```

if(Irms < 0.5){
    sensorLED = ledOff;      // Si la lampara esta apagada
}
if(Irms >= 0.5){
    sensorLED = ledOn;      // Si la lampara esta encendida
}
return sensorLED;
//delay(1000);
}

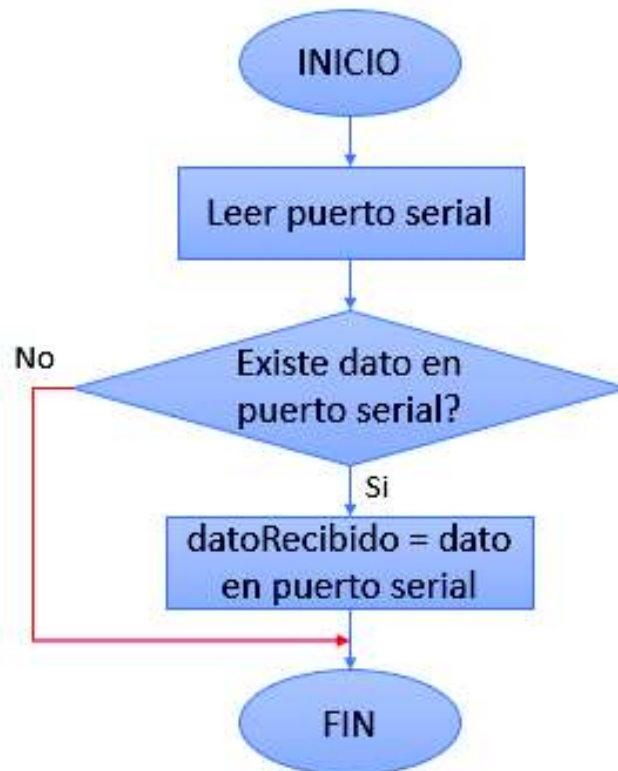
void envioSensor(char LED)
{
    if(e != sensorLED){
        if(tiempo <= millis()){
            tiempo = millis() + tiempoEnvio;
        }
        Serial.print(LED);
        e = sensorLED;
    }
    //delay(1000);
}

```

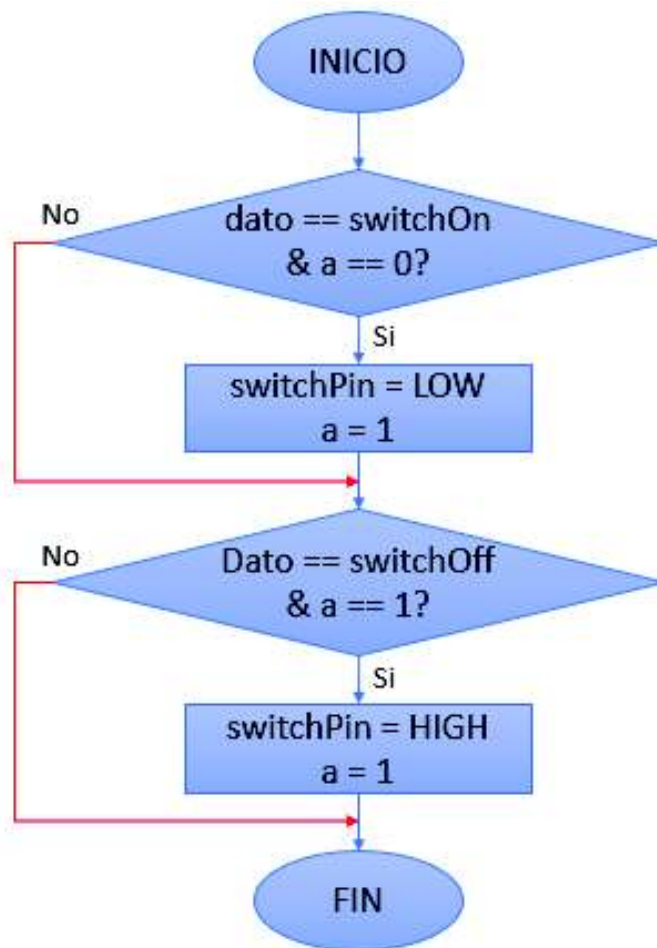
ANEXO II

Los diagramas de bloques de las funciones del programa son los siguientes:

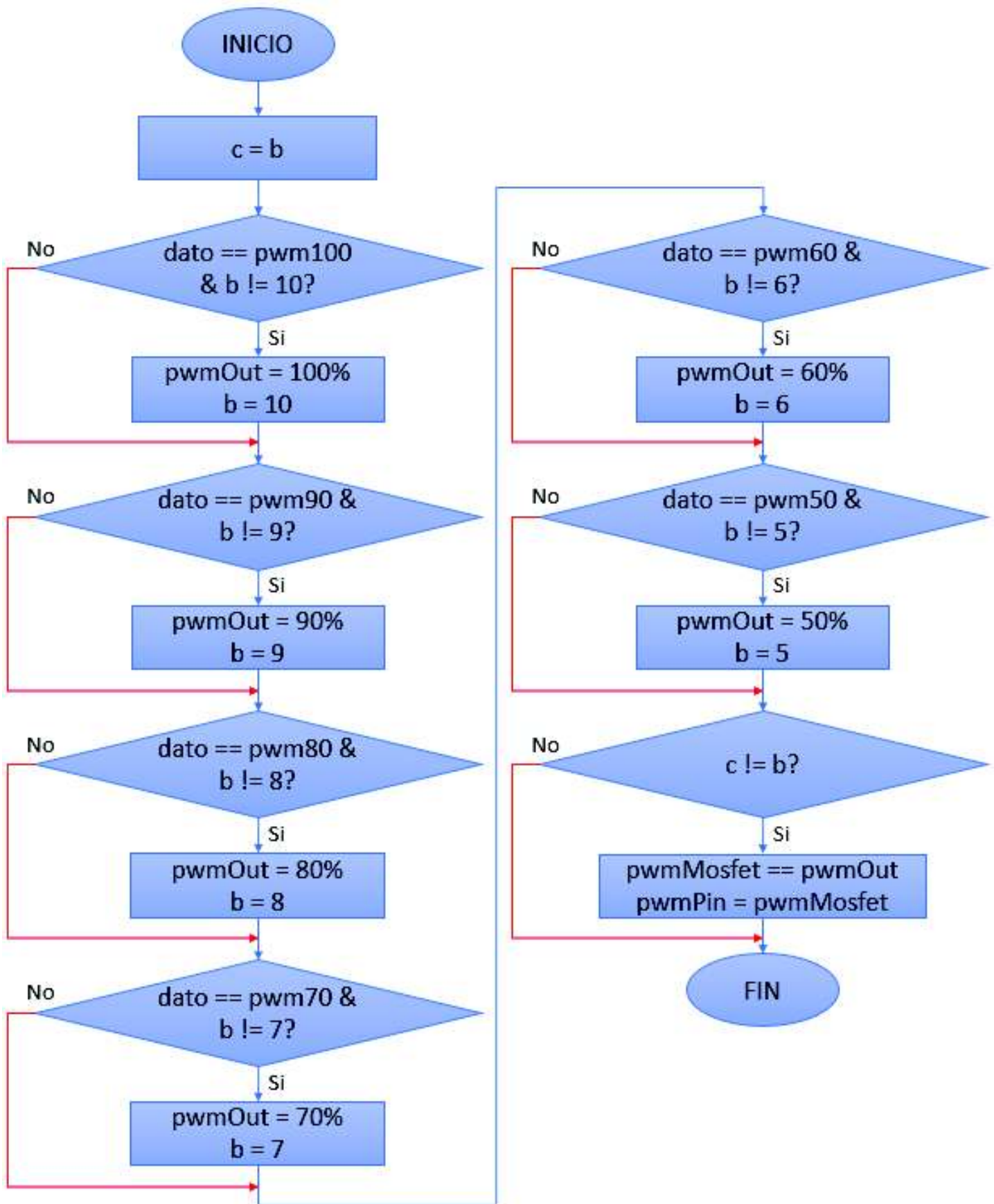
recibirDatos()



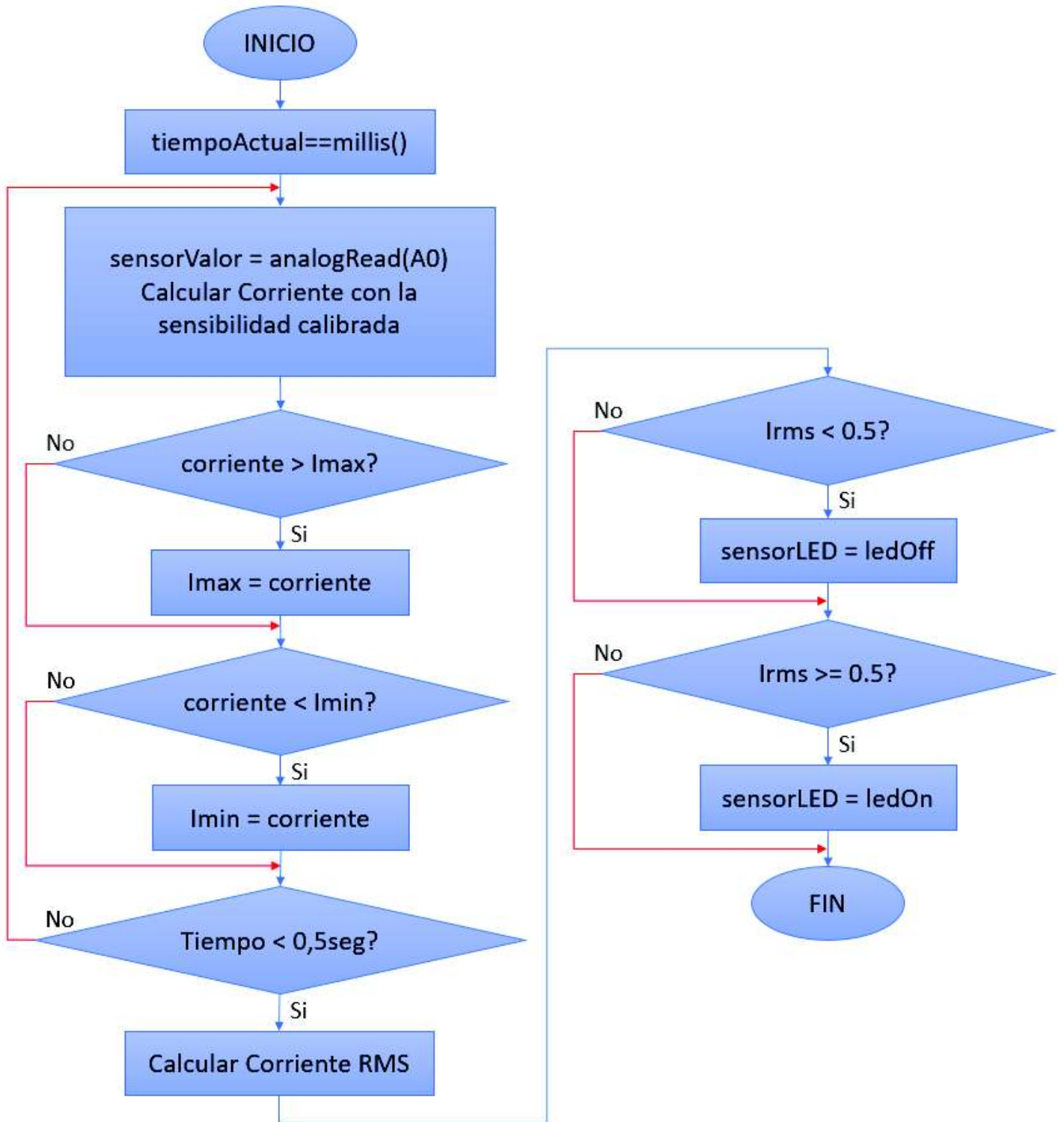
switchRele()



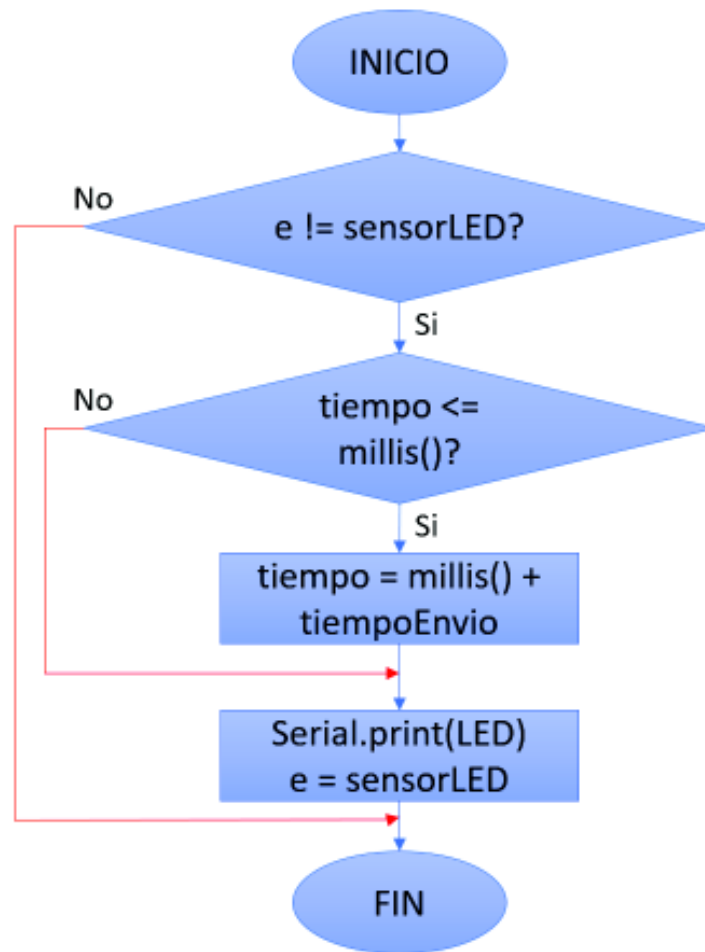
pwmControl()



lecturaSensor()



envioSensor()



ANEXO III

Los parámetros configurados en el módulo Xbee correspondiente al nodo Coordinador son los siguientes:



Product family: XB24C **Function set:** ZIGBEE TH Reg **Firmware version:** 4060

▼ **Networking**
Change networking settings

| | | | | |
|---|---------------------------------|-----------------------------------|------------|--|
| i | ID PAN ID | <input type="text" value="1704"/> | | |
| i | SC Scan Channels | <input type="text" value="7FFF"/> | Bitfield | |
| i | SD Scan Duration | <input type="text" value="3"/> | exponent | |
| i | ZS ZigBee Stack Profile | <input type="text" value="0"/> | | |
| i | NJ Node Join Time | <input type="text" value="FF"/> | x 1 sec | |
| i | NW Network Watchdog Timeout | <input type="text" value="0"/> | x 1 minute | |
| i | JV Channel Verification | Disabled [0] ▼ | | |
| i | JN Join Notification | Enabled [1] ▼ | | |
| i | OP Operating PAN ID | 1704 | | |
| i | OI Operating 16-bit PAN ID | 1360 | | |
| i | CH Operating Channel | 13 | | |
| i | NC Number of Remaining Children | 14 | | |
| i | CE Coordinator Enable | Enabled [1] ▼ | | |
| i | DO Device Options | <input type="text" value="0"/> | Bitfield | |
| i | DC Device Controls | <input type="text" value="0"/> | Bitfield | |


▼ Addressing

Change addressing settings

| | | |
|---|---|--|
| i | SH Serial Number High | 13A200 |
| i | SL Serial Number Low | 40D7D11D |
| i | MY 16-bit Network Address | 0 |
| i | MP 16-bit Parent Address | FFFE |
| i | DH Destination Address High | <input type="text" value="0"/> |
| i | DL Destination Address Low | <input type="text" value="FFFF"/> |
| i | NI Node Identifier | <input type="text" value="Coordinator"/> |
| i | NH Maximum Hops | <input type="text" value="1E"/> |
| i | BH Broadcast Radius | <input type="text" value="0"/> |
| i | AR Many-to-One Route Broadcast Time | <input type="text" value="FF"/> x 10 sec  |
| i | DD Device Type Identifier | <input type="text" value="A0000"/> |
| i | NT Node Discovery Backoff | <input type="text" value="3C"/> x 100 ms  |
| i | NO Node Discovery Options | <input type="text" value="0"/> |
| i | NP Maximum Number of Transmission Bytes | 54 |
| i | CR PAN Conflict Threshold | <input type="text" value="3"/> |

▼ ZigBee Addressing

Change ZigBee protocol addressing settings

| | | |
|---|--------------------------------|---|
| i | SE ZigBee Source Endpoint | <input type="text" value="E8"/> |
| i | DE ZigBee Destination Endpoint | <input type="text" value="E8"/> |
| i | CI ZigBee Cluster ID | <input type="text" value="11"/> |
| i | TO Transmit Options | <input type="text" value="0"/> Bitfield  |


▼ RF Interfacing

Change RF interface options

| | | |
|---|-------------------|---|
| i | PL TX Power Level | <input type="text" value="Highest [4]"/> ▼ |
| i | PM Power Mode | <input type="text" value="Boost Mode Enabled [1]"/> ▼ |
| i | PP Power at PL4 | 8 |

▼ Security

Change security parameters

| | | |
|----------------------------------|----------------------|--|
| EE Encryption Enable | Disabled [0] | ▼ |
| EO Encryption Options | 0 | Bitfield  |
| KY Encryption Key | <input type="text"/> | |
| NK Network Encryption Key | <input type="text"/> | |



▼ Serial Interfacing

Change modem interfacing options

| | | |
|--|-----------------------|-------------------|
| BD Baud Rate | 9600 [3] | ▼ |
| NB Parity | No Parity [0] | ▼ |
| SB Stop Bits | One stop bit [0] | ▼ |
| RO Packetization Timeout | 3 | x character times |
| D6 Pin 16 - DIO6/nRTS Configuration | Disable [0] | ▼ |
| D7 Pin 12 - DIO7/nCTS Configuration | nCTS flow control [1] | ▼ |
| AP API Enable | Transparent mode [0] | ▼ |
| AO API Output Mode | Native [0] | ▼ |






▼ AT Command Options

Change AT command mode behavior

| | | |
|--------------------------------------|-----|---|
| CT AT Command Mode Timeout | 64 | x 100ms  |
| GT Guard Times | 3E8 | x 1ms  |
| CC Command Sequence Character | 2B | Recommend...7F (ASCII) |



▼ Sleep Modes

Configure low power options to support end device children

| | | |
|--|-----------------------|--|
| SP Cyclic Sleep Period | 20 | x 10 ms  |
| SN Number of Cyclic Sleep Periods | <input type="text"/> | |
| SM Sleep Mode | No Sleep (Router) [0] | ▼ |
| ST Time before Sleep | 1388 | x 1 ms  |
| SO Sleep Options | 0 | bitfield  |
| WH Wake Host | 0 | x 1 ms  |
| PO Poll Rate | 0 | x 100 ms  |


▼ I/O Settings

Modify DIO and ADC options

| | | |
|---|----------------------------|---|
| D0 Pin 20 - DIO0/AD0/CB Configuration | Commissioning Button [1] | ▼ |
| D1 Pin 19 - DIO1/AD1/n..._ATTN Configuration | Disabled [0] | ▼ |
| D2 Pin 18 - DIO2/AD2/SPI_SCLK Configuration | Disabled [0] | ▼ |
| D3 Pin 17 - DIO3/AD3/nSPI_SSEL Configuration | Disabled [0] | ▼ |
| D4 Pin 11 - DIO4/SPI_MOSI Configuration | Disabled [0] | ▼ |
| D5 Pin 15 - DIO5/Associated Configuration | Associated indicator [1] | ▼ |
| D8 Pin 9 - DIO8/nDTR/Sleep_Rq Configuration | Sleep_Rq [1] | ▼ |
| D9 Pin 13 - DIO9/nOn_Sleep Configuration | Awake/Asleep indicator [1] | ▼ |
| P0 Pin 6 - DIO10/RSSI .../PWM0 Configuration | RSSI PWM Output [1] | ▼ |
| P1 Pin 7 - DIO11/PWM1 Configuration | Disabled [0] | ▼ |
| P2 Pin 4 - DIO12/SPI_MISO Configuration | Disabled [0] | ▼ |
| P3 Pin 2 - DIO13/DOOUT Configuration | DOOUT [1] | ▼ |
| P4 Pin 3 - DIO14/DIN/nConfig Configuration | DIN [1] | ▼ |
| PR Pull-up Resistor Enable | 1F8F | |
| PD Pull-up/down Direction | 1FFF | |
| LT Associate LED Blink Time | 0 x10 ms |  |
| RP RSSI PWM Timer | 28 x 100 ms |  |

▼ I/O Sampling

Configure IO sampling parameters

| | | |
|---|----------|---|
| IR IO Sampling Rate | 0 x 1 ms |  |
| IC Digital IO Change Detection | 0 | |
| V+ Supply Voltage High Threshold | 0 | |

▼ Diagnostic Commands

Access diagnostic parameters




| | |
|----------------------------------|------|
| VR Firmware Version | 4060 |
| HV Hardware Version | 2E46 |
| AI Association Indication | 0 |
| DB RSSI of Last Packet | 0 |
| %V Supply Voltage | CEA |

Los parámetros configurados en los módulos Xbee correspondientes a cada nodo Ruteador son los siguientes:

Product family: XB24-ZB **Function set:** ZigBee Router AT **Firmware version:** 22A7



▼ **Networking**

Change networking settings

| | | |
|--|---|--|
| ID PAN ID | <input type="text" value="1704"/> | |
| SC Scan Channels | <input type="text" value="FFFF"/> | Bitfield  |
| SD Scan Duration | <input type="text" value="3"/> | exponent |
| ZS ZigBee Stack Profile | <input type="text" value="0"/> | |
| NJ Node Join Time | <input type="text" value="FF"/> | x 1 sec  |
| NW Network Watchdog Timeout | <input type="text" value="0"/> | x 1 minute  |
| JV Channel Verification | <input type="text" value="Enabled [1]"/> | ▼ |
| JN Join Notification | <input type="text" value="Disabled [0]"/> | ▼ |
| OP Operating PAN ID | 1704 | |
| OI Operating 16-bit PAN ID | 1360 | |
| CH Operating Channel | 13 | |
| NC Number of Remaining Children | C | |

▼ Addressing

Change addressing settings

| | | |
|---|---|--|
| i | SH Serial Number High | 13A200 |
| i | SL Serial Number Low | 40DCA2E3 |
| i | MY 16-bit Network Address | 2D3E |
| i | DH Destination Address High | <input type="text" value="0"/> |
| i | DL Destination Address Low | <input type="text" value="FFFF"/> |
| i | NI Node Identifier | <input type="text" value="Router1"/> |
| i | NH Maximum Hops | <input type="text" value="1E"/> |
| i | BH Broadcast Radius | <input type="text" value="0"/> |
| i | AR Many-to-One Route Broadcast Time | <input type="text" value="FF"/> x 10 sec  |
| i | DD Device Type Identifier | <input type="text" value="30000"/> |
| i | NT Node Discovery Backoff | <input type="text" value="3C"/> x 100 ms  |
| i | NO Node Discovery Options | <input type="text" value="0"/> |
| i | NP Maximum Number of Transmission Bytes | 54 |
| i | CR PAN Conflict Threshold | <input type="text" value="3"/> |

▼ ZigBee Addressing

Change ZigBee protocol addressing settings

| | | |
|---|--------------------------------|---------------------------------|
| i | SE ZigBee Source Endpoint | <input type="text" value="E8"/> |
| i | DE ZigBee Destination Endpoint | <input type="text" value="E8"/> |
| i | CI ZigBee Cluster ID | <input type="text" value="11"/> |


▼ RF Interfacing

Change RF interface options

| | | |
|---|-----------------|---|
| i | PL Power Level | <input type="text" value="Highest [4]"/> ▼ |
| i | PM Power Mode | <input type="text" value="Boost Mode Enabled [1]"/> ▼ |
| i | PP Power at PL4 | 3 |

▼ Security

Change security parameters

| | | |
|---|-----------------------|---|
| i | EE Encryption Enable | <input type="text" value="Disabled [0]"/> ▼ |
| i | EO Encryption Options | <input type="text" value="0"/> Bitfield  |
| i | KY Encryption Key | <input type="text"/> |



Serial Interfacing

Change modem interfacing options

| | | |
|---------------------------------|----------------------|-------------------|
| BD Baud Rate | 9600 [3] | ▼ |
| NB Parity | No Parity [0] | ▼ |
| SB Stop Bits | One stop bit [0] | ▼ |
| RO Packetization Timeout | 3 | x character times |
| D7 DIO7 Configuration | CTS flow control [1] | ▼ |
| D6 DIO6 Configuration | Disable [0] | ▼ |



AT Command Options

Change AT command mode behavior

| | | | |
|--------------------------------------|-----|------------------------|---|
| CT AT Command Mode Timeout | 64 | x 100ms |  |
| GT Guard Times | 3E8 | x 1ms |  |
| CC Command Sequence Character | 2B | Recommend...7F (ASCII) | |

Sleep Modes

Configure low power options to support end device children

| | | | |
|--|-----------------------|---------|---|
| SM Sleep Mode | No Sleep (Router) [0] | ▼ | |
| SN Number of Cyclic Sleep Periods | 1 | | |
| SO Sleep Options | 0 | | |
| SP Cyclic Sleep Period | 20 | x 10 ms |  |
| ST Time before Sleep | 1388 | x 1 ms |  |
| PO Poll Rate | 0 | | |

▼ I/O Settings

Modify DIO and ADC options

| | | |
|------------------------------------|--|---|
| D0 AD0/DIO0 Configuration | Commissioning Button [1] | ▼ |
| D1 AD1/DIO1 Configuration | Disabled [0] | ▼ |
| D2 AD2/DIO2 Configuration | Disabled [0] | ▼ |
| D3 AD3/DIO3 Configuration | Disabled [0] | ▼ |
| D4 DIO4 Configuration | Disabled [0] | ▼ |
| D5 DIO5/Assoc Configuration | Associated indicator [1] | ▼ |
| P0 DIO10/PWM0 Configuration | RSSI PWM Output [1] | ▼ |
| P1 DIO11 Configuration | Disabled [0] | ▼ |
| P2 DIO12 Configuration | Disabled [0] | ▼ |
| PR Pull-up Resistor Enable | 1FFF | |
| LT Associate LED Blink Time | <input type="text" value="0"/> x10 ms | |
| RP RSSI PWM Timer | <input type="text" value="28"/> x 100 ms | |
| DO Device Options | <input type="text" value="1"/> Bitfield | |

▼ I/O Sampling

Configure IO sampling parameters

| | | |
|---|---------------------------------------|--|
| IR IO Sampling Rate | <input type="text" value="0"/> x 1 ms | |
| IC Digital IO Change Detection | <input type="text" value="0"/> | |
| V+ Supply Voltage High Threshold | <input type="text" value="0"/> | |

▼ Diagnostic Commands

Access diagnostic parameters

| | |
|----------------------------------|------|
| VR Firmware Version | 22A7 |
| HV Hardware Version | 194B |
| AI Association Indication | 0 |
| DB RSSI of Last Packet | 0 |
| %V Supply Voltage | AFE |

ANEXO IV

Las configuraciones necesarias y programación de la interfaz de monitoreo desarrollada en Visual Studio 2017 son los siguientes:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.IO.Ports;
using System.Data.OleDb;
using Microsoft.Office.Interop.Excel;

namespace EMELNORTE_SISTEMA_TELEGESTION
{
    public partial class Form1 : Form
    {
        private delegate void DelegadoAcceso(string accion);
        private string strBufferIn;
        private string strBufferOut;
        double operacionNumber;
        double flag;
        double flag2;
        double count;
        double flag_lum = -1;
        DateTime inicio_lum;
        DateTime final_lum;

        public Form1()
        {
            InitializeComponent();
        }

        private void AccesoForm(string accion)
        {
            strBufferIn = accion;
            //-----
            TxtDatosRecibir.Text = strBufferIn;
            if (strBufferIn == "Q")
            {
                TxtLuminarial.BackColor = Color.Green;
                TxtLuminarial.Text = "Luminaria 1 | Encendida";
                Txt_cambiorealizado.BackColor = Color.Green;
                Txt_cambiorealizado.Text = "Luminaria 1 Encendida";
                operacionNumber = operacionNumber + 1;
                TxtNumOperacion.Text = operacionNumber.ToString();

                this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
                TxtLuminarial.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);
            }
        }
    }
}
```

```

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "q")
    {
        TxtLuminaria1.BackColor = Color.Red;
        TxtLuminaria1.Text = "Luminaria 1 | Apagada";
        Txt_cambiorealizado.BackColor = Color.Red;
        Txt_cambiorealizado.Text = "Luminaria 1 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria1.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "W")
    {
        TxtLuminaria2.BackColor = Color.Green;
        TxtLuminaria2.Text = "Luminaria 2 | Encendida";
        Txt_cambiorealizado.BackColor = Color.Green;
        Txt_cambiorealizado.Text = "Luminaria 2 Encendida";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria2.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "w")
    {
        TxtLuminaria2.BackColor = Color.Red;
        TxtLuminaria2.Text = "Luminaria 2 | Apagada";
        Txt_cambiorealizado.BackColor = Color.Red;
        Txt_cambiorealizado.Text = "Luminaria 2 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria2.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "E")
    {
        TxtLuminaria3.BackColor = Color.Green;
        TxtLuminaria3.Text = "Luminaria 3 | Encendida";
        Txt_cambiorealizado.BackColor = Color.Green;
        Txt_cambiorealizado.Text = "Luminaria 3 Encendida";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria3.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "e")

```

```

        {
            TxtLuminaria3.BackColor = Color.Red;
            TxtLuminaria3.Text = "Luminaria 3 | Apagada";
            Txt_cambiorealizado.BackColor = Color.Red;
            Txt_cambiorealizado.Text = "Luminaria 3 Apagada";
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria3.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        }
        else if (strBufferIn == "R")
        {
            TxtLuminaria4.BackColor = Color.Green;
            TxtLuminaria4.Text = "Luminaria 4 | Encendida";
            Txt_cambiorealizado.BackColor = Color.Green;
            Txt_cambiorealizado.Text = "Luminaria 4 Encendida";
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria4.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        }
        else if (strBufferIn == "r")
        {
            TxtLuminaria4.BackColor = Color.Red;
            TxtLuminaria4.Text = "Luminaria 4 | Apagada";
            Txt_cambiorealizado.BackColor = Color.Red;
            Txt_cambiorealizado.Text = "Luminaria 4 Apagada";
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria4.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        }
        else if (strBufferIn == "T")
        {
            TxtLuminaria5.BackColor = Color.Green;
            TxtLuminaria5.Text = "Luminaria 5 | Encendida";
            Txt_cambiorealizado.BackColor = Color.Green;
            Txt_cambiorealizado.Text = "Luminaria 5 Encendida";
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria5.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        }
        else if (strBufferIn == "t")
        {
            TxtLuminaria5.BackColor = Color.Red;
            TxtLuminaria5.Text = "Luminaria 5 | Apagada";
            Txt_cambiorealizado.BackColor = Color.Red;

```

```

        Txt_cambiorealizado.Text = "Luminaria 5 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria5.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "Y")
    {
        TxtLuminaria6.BackColor = Color.Green;
        TxtLuminaria6.Text = "Luminaria 6 | Encendida";
        Txt_cambiorealizado.BackColor = Color.Green;
        Txt_cambiorealizado.Text = "Luminaria 6 Encendida";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria6.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "y")
    {
        TxtLuminaria6.BackColor = Color.Red;
        TxtLuminaria6.Text = "Luminaria 6 | Apagada";
        Txt_cambiorealizado.BackColor = Color.Red;
        Txt_cambiorealizado.Text = "Luminaria 6 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria6.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "U")
    {
        TxtLuminaria7.BackColor = Color.Green;
        TxtLuminaria7.Text = "Luminaria 7 | Encendida";
        Txt_cambiorealizado.BackColor = Color.Green;
        Txt_cambiorealizado.Text = "Luminaria 7 Encendida";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria7.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "u")
    {
        TxtLuminaria7.BackColor = Color.Red;
        TxtLuminaria7.Text = "Luminaria 7 | Apagada";
        Txt_cambiorealizado.BackColor = Color.Red;
        Txt_cambiorealizado.Text = "Luminaria 7 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

```

```

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
TxtLuminaria7.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
}
else if (strBufferIn == "I")
{
    TxtLuminaria8.BackColor = Color.Green;
    TxtLuminaria8.Text = "Luminaria 8 | Encendida";
    Txt_cambiorealizado.BackColor = Color.Green;
    Txt_cambiorealizado.Text = "Luminaria 8 Encendida";
    operacionNumber = operacionNumber + 1;
    TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
TxtLuminaria8.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
}
else if (strBufferIn == "i")
{
    TxtLuminaria8.BackColor = Color.Red;
    TxtLuminaria8.Text = "Luminaria 8 | Apagada";
    Txt_cambiorealizado.BackColor = Color.Red;
    Txt_cambiorealizado.Text = "Luminaria 8 Apagada";
    operacionNumber = operacionNumber + 1;
    TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
TxtLuminaria8.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
}
else if (strBufferIn == "O")
{
    TxtLuminaria9.BackColor = Color.Green;
    TxtLuminaria9.Text = "Luminaria 9 | Encendida";
    Txt_cambiorealizado.BackColor = Color.Green;
    Txt_cambiorealizado.Text = "Luminaria 9 Encendida";
    operacionNumber = operacionNumber + 1;
    TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
TxtLuminaria9.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
}
else if (strBufferIn == "o")
{
    TxtLuminaria9.BackColor = Color.Red;
    TxtLuminaria9.Text = "Luminaria 9 | Apagada";
    Txt_cambiorealizado.BackColor = Color.Red;
    Txt_cambiorealizado.Text = "Luminaria 9 Apagada";
    operacionNumber = operacionNumber + 1;
    TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
TxtLuminaria9.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

```

```

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "P")
    {
        TxtLuminaria10.BackColor = Color.Green;
        TxtLuminaria10.Text = "Luminaria 10 | Encendida";
        Txt_cambiorealizado.BackColor = Color.Green;
        Txt_cambiorealizado.Text = "Luminaria 10 Encendida";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria10.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "p")
    {
        TxtLuminaria10.BackColor = Color.Red;
        TxtLuminaria10.Text = "Luminaria 10 | Apagada";
        Txt_cambiorealizado.BackColor = Color.Red;
        Txt_cambiorealizado.Text = "Luminaria 10 Apagada";
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
TxtLuminaria10.Text, Txt_cambiorealizado.Text, lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    }
    else if (strBufferIn == "a")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 100%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if(flag_lum != 100)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
            flag_lum = 100;
        }
    }
    else if (strBufferIn == "s")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 90%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 90)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

```

```

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
    flag_lum = 90;
    }
    }
    else if (strBufferIn == "d")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 80%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 80)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
    flag_lum = 80;
    }
    }
    else if (strBufferIn == "f")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 70%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 70)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill (this.luminosidadDataSet.Operaciones);
    flag_lum = 70;
    }
    }
    else if (strBufferIn == "g")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 60%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 60)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR (TxtNumOperacion.Text,

```

```

Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    flag_lum = 60;
    }
    }
    else if (strBufferIn == "h")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 50%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 50)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    flag_lum = 50;
    }
    }
    else if (strBufferIn == "j")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 40%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 40)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    flag_lum = 40;
    }
    }
    else if (strBufferIn == "k")
    {
        TxtIntensidadAutomatica.BackColor = Color.Green;
        TxtIntensidadAutomatica.Text = "Luminosidad al 30%";
        Txt_cambiorealizado.BackColor = Color.SkyBlue;
        Txt_cambiorealizado.Text = "Luminosidad";
        if (flag_lum != 30)
        {
            operacionNumber = operacionNumber + 1;
            TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

```



```

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        flag_lum = 30;
    }
}
else if (strBufferIn == "l")
{
    TxtIntensidadAutomatica.BackColor = Color.Green;
    TxtIntensidadAutomatica.Text = "Luminosidad al 20%";
    Txt_cambiorealizado.BackColor = Color.SkyBlue;
    Txt_cambiorealizado.Text = "Luminosidad";
    if (flag_lum != 20)
    {
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        flag_lum = 20;
    }
}
else if (strBufferIn == "m")
{
    TxtIntensidadAutomatica.BackColor = Color.Green;
    TxtIntensidadAutomatica.Text = "Luminosidad al 10%";
    Txt_cambiorealizado.BackColor = Color.SkyBlue;
    Txt_cambiorealizado.Text = "Luminosidad";
    if (flag_lum != 10)
    {
        operacionNumber = operacionNumber + 1;
        TxtNumOperacion.Text = operacionNumber.ToString();

this.operacionesTableAdapter.INSERTAR(TxtNumOperacion.Text,
Txt_cambiorealizado.Text, TxtIntensidadAutomatica.Text,
lbl_date_time.Text);

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
        flag_lum = 10;
    }
}
//-----
}

private void AccesoInterrupcion(string accion)
{
    DelegadoAcceso Var_DelegadoAcceso;
    Var_DelegadoAcceso = new DelegadoAcceso(AccesoForm);
    object[] arg = { accion };
    base.Invoke(Var_DelegadoAcceso, arg);
}

private void Form1_Load(object sender, EventArgs e)
{
    // TODO: esta línea de código carga datos en la tabla
    'telegestionDataSet.Energia' Puede moverla o quitarla según sea
necesario.

```

```

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    // TODO: esta línea de código carga datos en la tabla
'luminosidadDataSet.Operaciones' Puede moverla o quitarla según sea
necesario.

this.operacionesTableAdapter.Fill(this.luminosidadDataSet.Operaciones);
    strBufferIn = "";
    strBufferOut = "";
    btn_conectar.Enabled = false;
    btn_enviar_dato.Enabled = false;
    btn_encender.Enabled = false;
    btn_apagar.Enabled = false;
    btn_100.Enabled = false;
    btn_90.Enabled = false;
    btn_80.Enabled = false;
    btn_70.Enabled = false;
    btn_60.Enabled = false;
    btn_50.Enabled = false;
    btn_40.Enabled = false;
    btn_30.Enabled = false;
    btn_20.Enabled = false;
    btn_10.Enabled = false;

}

private void btn_buscar_puertos_Click(object sender, EventArgs e)
{
    string[] PuertosDisponibles = SerialPort.GetPortNames();
    cbox_puertos.Items.Clear();
    foreach (string puerto_simple in PuertosDisponibles)
    {
        cbox_puertos.Items.Add(puerto_simple);
    }
    if (cbox_puertos.Items.Count > 0)
    {
        cbox_puertos.SelectedIndex = 0;
        MessageBox.Show("Seleccionar el Puerto de Enlace");
        btn_conectar.Enabled = true;
    }
    else
    {
        MessageBox.Show("No se ha Detectado Ningún Puerto");
        cbox_puertos.Items.Clear();
        cbox_puertos.Text = "          ";
        strBufferIn = "";
        strBufferOut = "";
        btn_conectar.Enabled = false;
        btn_enviar_dato.Enabled = false;
        btn_encender.Enabled = false;
        btn_apagar.Enabled = false;
        btn_100.Enabled = false;
        btn_90.Enabled = false;
        btn_80.Enabled = false;
        btn_70.Enabled = false;
        btn_60.Enabled = false;
        btn_50.Enabled = false;
        btn_40.Enabled = false;
        btn_30.Enabled = false;
        btn_20.Enabled = false;
    }
}

```

```

        btn_10.Enabled = false;
    }
}

private void btn_conectar_Click(object sender, EventArgs e)
{
    try
    {
        if (btn_conectar.Text=="Conectar")
        {
            SpPuertos.BaudRate =
Int32.Parse(cbox_velocidad.Text);
            SpPuertos.DataBits = 8;
            SpPuertos.Parity = Parity.None;
            SpPuertos.StopBits = StopBits.One;
            SpPuertos.Handshake = Handshake.None;
            SpPuertos.PortName = cbox_puertos.Text;
            flag = 100;
            inicio_lum = DateTime.Now;

            try
            {
                SpPuertos.Open();
                btn_conectar.Text = "Desconectar";
                btn_enviar_dato.Enabled = true;
                btn_encender.Enabled = true;
                btn_apagar.Enabled = true;
                btn_100.Enabled = true;
                btn_90.Enabled = true;
                btn_80.Enabled = true;
                btn_70.Enabled = true;
                btn_60.Enabled = true;
                btn_50.Enabled = true;
                btn_40.Enabled = true;
                btn_30.Enabled = true;
                btn_20.Enabled = true;
                btn_10.Enabled = true;
            }
            catch (Exception exc)
            {
                MessageBox.Show(exc.Message.ToString());
            }
        }
        else if (btn_conectar.Text=="Desconectar")
        {
            SpPuertos.Close();
            btn_conectar.Text = "Conectar";
            btn_enviar_dato.Enabled = false;
            btn_encender.Enabled = false;
            btn_apagar.Enabled = false;
            btn_100.Enabled = false;
            btn_90.Enabled = false;
            btn_80.Enabled = false;
            btn_70.Enabled = false;
            btn_60.Enabled = false;
            btn_50.Enabled = false;
            btn_40.Enabled = false;
            btn_30.Enabled = false;
            btn_20.Enabled = false;
            btn_10.Enabled = false;
        }
    }
}

```

```

        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message.ToString());
    }
}

private void btn_enviar_dato_Click(object sender, EventArgs e)
{
    try
    {
        SpPuertos.DiscardOutBuffer();
        strBufferOut = tbox_dato_a_enviar.Text;
        SpPuertos.Write(strBufferOut);
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message.ToString());
    }
}

private void DatoRecibido(object sender,
SerialDataReceivedEventArgs e)
{
    AccesoInterrupcion(SpPuertos.ReadExisting());
    /*string Data_in = SpPuertos.ReadExisting();
    MessageBox.Show(Data_in);
    TxtDatosRecibir.Text = Data_in;*/
}

private void btn_encender_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("v");
    pbox_on.Visible = true;
    pbox_off.Visible = false;
    btn_100.Enabled = true;
    btn_90.Enabled = true;
    btn_80.Enabled = true;
    btn_70.Enabled = true;
    btn_60.Enabled = true;
    btn_50.Enabled = true;
    btn_40.Enabled = true;
    btn_30.Enabled = true;
    btn_20.Enabled = true;
    btn_10.Enabled = true;
    if (flag2 != 1)
    {
        inicio_lum = DateTime.Now;
        flag2 = 1;
    }
}

private void btn_apagar_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("c");
    pbox_on.Visible = false;
    pbox_off.Visible = true;
    btn_100.Enabled = false;
}

```

```

        btn_90.Enabled = false;
        btn_80.Enabled = false;
        btn_70.Enabled = false;
        btn_60.Enabled = false;
        btn_50.Enabled = false;
        btn_40.Enabled = false;
        btn_30.Enabled = false;
        btn_20.Enabled = false;
        btn_10.Enabled = false;
        if (flag2 != 2)
        {
            final_lum = DateTime.Now;
            //Calculos para el 100%
            TimeSpan diferencia = new TimeSpan();
            TimeSpan consumo_lumen100 = new TimeSpan();
            TimeSpan consumo_driver = new TimeSpan();
            TimeSpan consumo_total100 = new TimeSpan();
            TimeSpan costo100 = new TimeSpan();
            diferencia = final_lum - inicio_lum;
            consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
            consumo_total100 = consumo_driver + consumo_lumen100;
            costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
            //Despliegue de datos
            lblTiempo.Text = diferencia.TotalHours.ToString();
            lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
            lblCosto100.Text = costo100.TotalHours.ToString();
            TimeSpan consumo_lumen = new TimeSpan();
            TimeSpan consumo_total = new TimeSpan();
            TimeSpan costo = new TimeSpan();
            TimeSpan ahorro_energia = new TimeSpan();
            TimeSpan ahorro_dinero = new TimeSpan();
            count = count + 1;
            lblCount.Text = count.ToString();
            if (flag == 100)
            {
                //Calculos para el % actual
                consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
                consumo_total = consumo_driver + consumo_lumen;
                costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
                //Calculos de ahorro
                ahorro_energia = consumo_total100 - consumo_total;
                ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
                lblEnergiaActual.Text = "Energia al 100% [kWh]";
                lblCostoActual.Text = "Costo al 100% [$]";
                //Despliegue de datos
                lblConsumo.Text =
consumo_total.TotalHours.ToString();
                lblCosto.Text = costo.TotalHours.ToString();
                lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
                lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();

```

```

    }
    if (flag == 90)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";

```

```

        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual

```

```

        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();

```



```

        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag2 = 2;
}
}

private void btn_100_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("a");
    Pbo100.Visible = true;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag!=100)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();

```

```

        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951 /
10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero = TimeSpan.FromTicks(ahorro_energia.Ticks *
951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
            lblCostoActual.Text = "Costo al 100% [$]";
            //Despliegue de datos
            lblConsumo.Text = consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text = ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 90)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 90% [kWh]";
            lblCostoActual.Text = "Costo al 90% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();

```

```

        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

```

```

        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }

```

```

    }
    if (flag == 30)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";

```

```

        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag = 100;
}
}

private void btn_90_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("s");
    Pbo100.Visible = false;
    Pbo90.Visible = true;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 90)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();

```

```

        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
            lblCostoActual.Text = "Costo al 100% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 90)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 90% [kWh]";
            lblCostoActual.Text = "Costo al 90% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 80)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;

```

```

        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)

```



```

        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 50% [kWh]";
            lblCostoActual.Text = "Costo al 50% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 40)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 40% [kWh]";
            lblCostoActual.Text = "Costo al 40% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 30)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 30% [kWh]";
            lblCostoActual.Text = "Costo al 30% [$]";
            //Despliegue de datos

```

```

        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
//Reiniciar contador

```

```

        inicio_lum = DateTime.Now;
        flag = 90;
    }
}

private void btn_80_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("d");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = true;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 80)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
        }
    }
}

```

```

        lblCostoActual.Text = "Costo al 100% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 90)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual

```

```

consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 70% [kWh]";
lblCostoActual.Text = "Costo al 70% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 60)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 60% [kWh]";
lblCostoActual.Text = "Costo al 60% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 50)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 50% [kWh]";
lblCostoActual.Text = "Costo al 50% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();

```

```

        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

```

```

        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag = 80;
}
}

private void btn_70_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("f");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = true;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
}

```

```

Pbo20.Visible = false;
Pbo10.Visible = false;
if (flag != 70)
{
    final_lum = DateTime.Now;
    //Calculos para el 100%
    TimeSpan diferencia = new TimeSpan();
    TimeSpan consumo_lumen100 = new TimeSpan();
    TimeSpan consumo_driver = new TimeSpan();
    TimeSpan consumo_total100 = new TimeSpan();
    TimeSpan costo100 = new TimeSpan();
    diferencia = final_lum - inicio_lum;
    consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
    consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
    consumo_total100 = consumo_driver + consumo_lumen100;
    costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
    //Despliegue de datos
    lblTiempo.Text = diferencia.TotalHours.ToString();
    lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
    lblCosto100.Text = costo100.TotalHours.ToString();
    TimeSpan consumo_lumen = new TimeSpan();
    TimeSpan consumo_total = new TimeSpan();
    TimeSpan costo = new TimeSpan();
    TimeSpan ahorro_energia = new TimeSpan();
    TimeSpan ahorro_dinero = new TimeSpan();
    count = count + 1;
    lblCount.Text = count.ToString();
    if (flag == 100)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 100% [kWh]";
        lblCostoActual.Text = "Costo al 100% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 90)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
        consumo_total = consumo_driver + consumo_lumen;

```



```

        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    }

```

```

        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;

```

```

        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)

```

```

        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);

            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 10% [kWh]";
            lblCostoActual.Text = "Costo al 10% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
        //Reiniciar contador
        inicio_lum = DateTime.Now;
        flag = 70;
    }
}

private void btn_60_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("g");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = true;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 60)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);

        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
    }
}

```

```

consumo_total100 = consumo_driver + consumo_lumen100;
costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
//Despliegue de datos
lblTiempo.Text = diferencia.TotalHours.ToString();
lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
lblCosto100.Text = costo100.TotalHours.ToString();
TimeSpan consumo_lumen = new TimeSpan();
TimeSpan consumo_total = new TimeSpan();
TimeSpan costo = new TimeSpan();
TimeSpan ahorro_energia = new TimeSpan();
TimeSpan ahorro_dinero = new TimeSpan();
count = count + 1;
lblCount.Text = count.ToString();
if (flag == 100)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 100% [kWh]";
lblCostoActual.Text = "Costo al 100% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 90)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 90% [kWh]";
lblCostoActual.Text = "Costo al 90% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}

```

```

    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";

```

```

        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual

```

```

consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 30% [kWh]";
lblCostoActual.Text = "Costo al 30% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 20)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 20% [kWh]";
lblCostoActual.Text = "Costo al 20% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();
lblCosto.Text = costo.TotalHours.ToString();
lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 10)
{
//Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
//Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
lblEnergiaActual.Text = "Energia al 10% [kWh]";
lblCostoActual.Text = "Costo al 10% [$]";
//Despliegue de datos
lblConsumo.Text =
consumo_total.TotalHours.ToString();

```



```

        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag = 60;
}
}

private void btn_50_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("h");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = true;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 50)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)

```

```

        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
            lblCostoActual.Text = "Costo al 100% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 90)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 90% [kWh]";
            lblCostoActual.Text = "Costo al 90% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 80)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 80% [kWh]";
            lblCostoActual.Text = "Costo al 80% [$]";
            //Despliegue de datos

```

```

        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;

```

```

        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    }

```

```

        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
        this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);
this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
        //Reiniciar contador
        inicio_lum = DateTime.Now;
        flag = 50;
    }
}
}

```

```

private void btn_40_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("j");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = true;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 40)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
            lblCostoActual.Text = "Costo al 100% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();

```

```

        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 90)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro

```

```

        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
}

```



```

        if (flag == 40)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);

            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 40% [kWh]";
            lblCostoActual.Text = "Costo al 40% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 30)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);

            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 30% [kWh]";
            lblCostoActual.Text = "Costo al 30% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 20)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);

            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 20% [kWh]";
            lblCostoActual.Text = "Costo al 20% [$]";

```

```

        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag = 40;
}
}

private void btn_30_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("k");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = true;
    Pbo20.Visible = false;
    Pbo10.Visible = false;
    if (flag != 30)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%

```

```

        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 100% [kWh]";
            lblCostoActual.Text = "Costo al 100% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 90)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);

```

```

        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual

```

```

        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();

```

```

        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

```

```

        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
    //Reiniciar contador
    inicio_lum = DateTime.Now;
    flag = 30;
    }
}

private void btn_20_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("1");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = true;
    Pbo10.Visible = false;
    if (flag != 20)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
    }
}

```

```

lblCosto100.Text = costo100.TotalHours.ToString();
TimeSpan consumo_lumen = new TimeSpan();
TimeSpan consumo_total = new TimeSpan();
TimeSpan costo = new TimeSpan();
TimeSpan ahorro_energia = new TimeSpan();
TimeSpan ahorro_dinero = new TimeSpan();
count = count + 1;
lblCount.Text = count.ToString();
if (flag == 100)
{
    //Calculos para el % actual
    consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
    consumo_total = consumo_driver + consumo_lumen;
    costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
    //Calculos de ahorro
    ahorro_energia = consumo_total100 - consumo_total;
    ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
    lblEnergiaActual.Text = "Energia al 100% [kWh]";
    lblCostoActual.Text = "Costo al 100% [$]";
    //Despliegue de datos
    lblConsumo.Text =
consumo_total.TotalHours.ToString();
    lblCosto.Text = costo.TotalHours.ToString();
    lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 90)
{
    //Calculos para el % actual
    consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
    consumo_total = consumo_driver + consumo_lumen;
    costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
    //Calculos de ahorro
    ahorro_energia = consumo_total100 - consumo_total;
    ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
    lblEnergiaActual.Text = "Energia al 90% [kWh]";
    lblCostoActual.Text = "Costo al 90% [$]";
    //Despliegue de datos
    lblConsumo.Text =
consumo_total.TotalHours.ToString();
    lblCosto.Text = costo.TotalHours.ToString();
    lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
}
if (flag == 80)
{
    //Calculos para el % actual
    consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
    consumo_total = consumo_driver + consumo_lumen;

```



```

        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    }

```

```

        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;

```

```

        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 20% [kWh]";
        lblCostoActual.Text = "Costo al 20% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 10)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 10% [kWh]";
        lblCostoActual.Text = "Costo al 10% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
}

```

```

        this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
        //Reiniciar contador
        inicio_lum = DateTime.Now;
        flag = 20;
    }
}

private void btn_10_Click(object sender, EventArgs e)
{
    SpPuertos.WriteLine("m");
    Pbo100.Visible = false;
    Pbo90.Visible = false;
    Pbo80.Visible = false;
    Pbo70.Visible = false;
    Pbo60.Visible = false;
    Pbo50.Visible = false;
    Pbo40.Visible = false;
    Pbo30.Visible = false;
    Pbo20.Visible = false;
    Pbo10.Visible = true;
    if (flag != 10)
    {
        final_lum = DateTime.Now;
        //Calculos para el 100%
        TimeSpan diferencia = new TimeSpan();
        TimeSpan consumo_lumen100 = new TimeSpan();
        TimeSpan consumo_driver = new TimeSpan();
        TimeSpan consumo_total100 = new TimeSpan();
        TimeSpan costo100 = new TimeSpan();
        diferencia = final_lum - inicio_lum;
        consumo_lumen100 = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
        consumo_driver = TimeSpan.FromTicks(diferencia.Ticks * 15
/ 1000);
        consumo_total100 = consumo_driver + consumo_lumen100;
        costo100 = TimeSpan.FromTicks(consumo_total100.Ticks *
951 / 10000);
        //Despliegue de datos
        lblTiempo.Text = diferencia.TotalHours.ToString();
        lblConsumo100.Text =
consumo_total100.TotalHours.ToString();
        lblCosto100.Text = costo100.TotalHours.ToString();
        TimeSpan consumo_lumen = new TimeSpan();
        TimeSpan consumo_total = new TimeSpan();
        TimeSpan costo = new TimeSpan();
        TimeSpan ahorro_energia = new TimeSpan();
        TimeSpan ahorro_dinero = new TimeSpan();
        count = count + 1;
        lblCount.Text = count.ToString();
        if (flag == 100)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
150 / 1000);
            consumo_total = consumo_driver + consumo_lumen;

```

```

        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 100% [kWh]";
        lblCostoActual.Text = "Costo al 100% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 90)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
136 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 90% [kWh]";
        lblCostoActual.Text = "Costo al 90% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 80)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
121 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 80% [kWh]";
        lblCostoActual.Text = "Costo al 80% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
    }

```

```

        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 70)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
107 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 70% [kWh]";
        lblCostoActual.Text = "Costo al 70% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 60)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
92 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;
        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 60% [kWh]";
        lblCostoActual.Text = "Costo al 60% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 50)
    {
        //Calculos para el % actual
        consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
78 / 1000);
        consumo_total = consumo_driver + consumo_lumen;
        costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
        //Calculos de ahorro
        ahorro_energia = consumo_total100 - consumo_total;

```

```

        ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 50% [kWh]";
        lblCostoActual.Text = "Costo al 50% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 40)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
64 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 40% [kWh]";
        lblCostoActual.Text = "Costo al 40% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 30)
    {
        //Calculos para el % actual
consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
49 / 1000);

        consumo_total = consumo_driver + consumo_lumen;
costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);

        //Calculos de ahorro
ahorro_energia = consumo_total100 - consumo_total;
ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
        lblEnergiaActual.Text = "Energia al 30% [kWh]";
        lblCostoActual.Text = "Costo al 30% [$]";
        //Despliegue de datos
        lblConsumo.Text =
consumo_total.TotalHours.ToString();
        lblCosto.Text = costo.TotalHours.ToString();
        lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
        lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
    }
    if (flag == 20)

```

```

        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
35 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 20% [kWh]";
            lblCostoActual.Text = "Costo al 20% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        if (flag == 10)
        {
            //Calculos para el % actual
            consumo_lumen = TimeSpan.FromTicks(diferencia.Ticks *
20 / 1000);
            consumo_total = consumo_driver + consumo_lumen;
            costo = TimeSpan.FromTicks(consumo_total.Ticks * 951
/ 10000);
            //Calculos de ahorro
            ahorro_energia = consumo_total100 - consumo_total;
            ahorro_dinero =
TimeSpan.FromTicks(ahorro_energia.Ticks * 951 / 10000);
            lblEnergiaActual.Text = "Energia al 10% [kWh]";
            lblCostoActual.Text = "Costo al 10% [$]";
            //Despliegue de datos
            lblConsumo.Text =
consumo_total.TotalHours.ToString();
            lblCosto.Text = costo.TotalHours.ToString();
            lblAhorroEnerg.Text =
ahorro_energia.TotalHours.ToString();
            lblAhorroDin.Text =
ahorro_dinero.TotalHours.ToString();
        }
        this.energiaTableAdapter.INSERTARENERGIA(lblCount.Text,
lblEnergiaActual.Text, lblTiempo.Text, lblConsumo.Text, lblCosto.Text,
lblAhorroEnerg.Text, lblAhorroDin.Text);

        this.energiaTableAdapter.Fill(this.telegestionDataSet.Energia);
        //Reiniciar contador
        inicio_lum = DateTime.Now;
        flag = 10;
    }
}

private void Timer1_Tick(object sender, EventArgs e)
{
    lbl_date_time.Text = DateTime.Now.ToString("dd/MM/yyyy
hh:mm:ss tt");
}

```



```

        //lbl_date_time.Text = DateTime.Now.ToLongTimeString();
    }

    private void operacionesBindingNavigatorSaveItem_Click(object
sender, EventArgs e)
    {
        this.Validate();
        this.operacionesBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.luminosidadDataSet);
    }

    private void operacionesBindingNavigatorSaveItem_Click_1(object
sender, EventArgs e)
    {
        this.Validate();
        this.operacionesBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.luminosidadDataSet);
    }

    private void btnExcelEventos_Click(object sender, EventArgs e)
    {
        Microsoft.Office.Interop.Excel.Application Excel = new
Microsoft.Office.Interop.Excel.Application();
        Workbook wb = Excel.Workbooks.Add(XlSheetType.xlWorksheet);
        Worksheet ws = (Worksheet)Excel.ActiveSheet;
        Excel.Visible = true;
        for (int i = 1; i < operacionesDataGridView.Columns.Count +
1; i++)
        {
            ws.Cells[1, i] = operacionesDataGridView.Columns[i -
1].HeaderText;
        }
        for (int i = 0; i < operacionesDataGridView.Rows.Count - 1;
i++)
        {
            for (int j = 0; j <
operacionesDataGridView.Columns.Count; j++)
            {
                ws.Cells[i + 2, j + 1] =
operacionesDataGridView.Rows[i].Cells[j].Value.ToString();
            }
        }
    }

    private void btnExcel_Click(object sender, EventArgs e)
    {
        Microsoft.Office.Interop.Excel.Application Excel = new
Microsoft.Office.Interop.Excel.Application();
        Workbook wb = Excel.Workbooks.Add(XlSheetType.xlWorksheet);
        Worksheet ws = (Worksheet)Excel.ActiveSheet;
        Excel.Visible = true;
        for (int i = 1; i < energiaDataGridView.Columns.Count + 1;
i++)
        {
            ws.Cells[1, i] = energiaDataGridView.Columns[i -
1].HeaderText;
        }
        for (int i = 0; i < energiaDataGridView.Rows.Count - 1; i++)

```

```
        {
            for (int j = 0; j < energiaDataGridView.Columns.Count;
j++)
                {
                    ws.Cells[i + 2, j + 1] =
energiaDataGridView.Rows[i].Cells[j].Value.ToString();
                }
            }
        }
    }
```

ANEXO V



Xitanium Prog/Prog+ LED Xtreme drivers

Xitanium 150W 0.35-0.70A GL Prog sXt

Xitanium Prog/Prog+ LED Xtreme drivers

Philips Xitanium Prog/Prog+ Programmable LED drivers are specifically designed to deliver the highest performance, protection and configurability. The portfolio offers both central and standalone dimming protocols further increasing the energy savings and CO2 reductions achieved with LED lighting. The Xtreme technology ensures maximum robustness and protection combined with a very long lifetime.

Benefits

- Ultimate robustness, offering peace of mind and lower maintenance costs
- Fully programmable LED drivers designed for the new digital and connected lighting world
- Extended diagnostics via MultiOne
- Easy to design-in, configure and install
- Energy savings through high efficiency and via multiple dimming options

Features

- High surge protection (CM/DM)
- Long lifetime and robust protection against moisture, vibration and temperature
- Configurable operating windows (AOC)
- Multiple control interfaces: DALI, 0-10V (Prog+ also AmpDim)
- Autonomous dimming via Integrated DynaDimmer
- Suitable for DC operation
- Thermal protection for driver and for module (MTP)
- Constant Light Output (CLO)
- Adjustable Start-up Time (AST)
- End-Of-Life Indicator (EOL)

Application

- Road and street lighting
- Area lighting
- Industrial lighting

Electrical input data

| Specification item | Value | Unit | Condition |
|------------------------------|-----------|-----------------|--|
| Rated input voltage range | 120...277 | V _{AC} | Performance range |
| Rated input voltage | 230 | V _{AC} | |
| Rated input frequency range | 50...60 | Hz | Performance range |
| Rated input current | 0.7 | A | @ rated output power @ rated input voltage |
| Max. input current | 1.54 | A | @ rated output power @ minimum performance input voltage |
| Rated input power | 165 | W | @ rated output power @ rated input voltage |
| Power factor | ≥ 0.97 | | @ rated output power @ rated input voltage |
| Total harmonic distortion | ≤ 12 | % | @ rated output power @ rated input voltage |
| Efficiency | ≥ 93 | % | @ rated output power @ rated input voltage |
| Rated input voltage DC range | 186...250 | V _{DC} | Performance range |
| Input voltage AC range | 108...305 | V _{AC} | Operational range |
| Input frequency AC range | 45...66 | Hz | Operational range |
| Input voltage DC range | 168...275 | V _{DC} | Operational range |
| Standby Power (TID) | 0.45 | W | |
| Isolation input to output | Basic | | |

Electrical output data

| Specification item | Value | Unit | Condition |
|---------------------------------|------------------|-----------------|-----------------------------|
| Regulation method | Constant Current | | |
| Output voltage | 125...280 | V _{DC} | |
| Output voltage max. | 300 | V | Peak voltage: at open load |
| Output current | 0.05...0.7 | A | Full output current setting |
| Output current min programmable | 200 | mA | |
| Output current min dimming | 50 | mA | |
| Output current tolerance | ± 5 | % | |
| Output current ripple LF | ≤ 15 | % | Ripple = peak / average |
| Output current ripple HF | ≤ 15 | % | |
| Output power | 7.5...150 | W | Full output |

Electrical data controls input

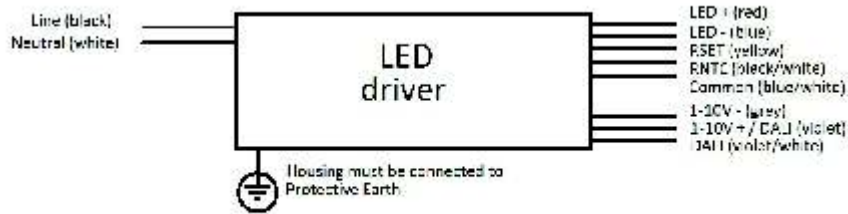
| Specification item | Value | Unit | Condition |
|--------------------|---------------------|------|--|
| Control method | 1-10V, DALI, Dimmer | | Output current amplitude dimming, DALI acc. IEC62386-102/207; 1-10V acc. IEC60929 |
| Dimming range | 10...100 | % | Default range |
| Galvanic Isolation | Basic | | |

Logistical data

| Specification item | Value |
|--------------------|--------------------------------------|
| Product name | Xitanium 150W 0.35-0.70A GL Prog s01 |
| Order code | B7279007831300 |
| Logistic code 12NC | 9290 007 02202 |
| EAN3 | B718291999782 |
| Pieces per box | 10 |

Wiring & Connections

| Specification item | Value | Unit | Condition |
|---------------------------------|--------|-----------------|--|
| Input wire cross-section | 0.82 | mm ² | solid wire, double-insulated |
| | 18 | AWG | solid wire, double-insulated |
| Input wire strip length | 8...12 | mm | |
| Output wire cross-section | 0.82 | mm ² | solid wire, double-insulated |
| | 18 | AWG | solid wire, double-insulated |
| Output wire strip length | 8...12 | mm | |
| Maximum cable length | 10000 | mm | Total length of wiring including LED module, one way |
| Maximum NTC output cable length | 0.6 | m | |

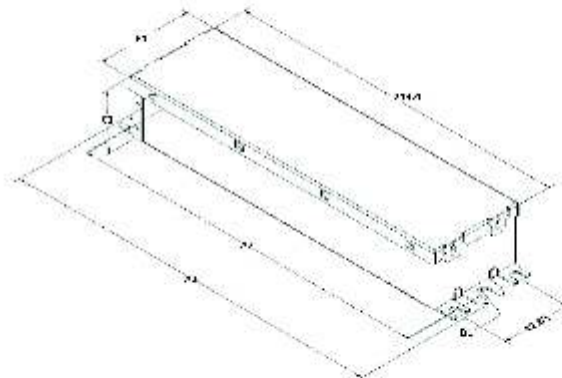


Insulation

| Insulation | Mains | LED + NTC | DALI/1-10V | Housing |
|------------|--------|-----------|------------|---------|
| Mains | | Basic | Basic | Double |
| LED + NTC | Basic | | Basic | Double |
| DALI/1-10V | Basic | Basic | | Double |
| Housing | Double | Double | Double | |

Dimensions and weight

| Specification item | Value | Unit | Condition |
|---------------------------|-------|------|-----------|
| Length (A1) | 240.5 | mm | |
| Width (B1) | 59.8 | mm | |
| Width (B2) | 42.88 | mm | |
| Height (C1) | 37.6 | mm | |
| Fixing hole diameter (D1) | 6 | mm | |
| Fixing hole distance (A2) | 226 | mm | |
| Weight | 1270 | gram | |



Operational temperatures and humidity

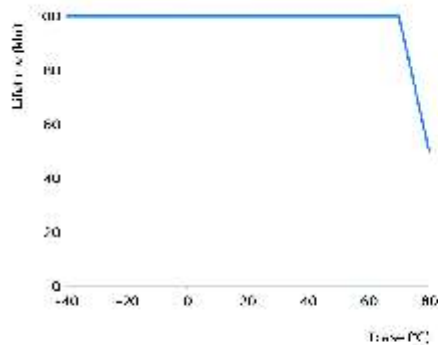
| Specification item | Value | Unit | Condition |
|------------------------------|-----------|------|---|
| Ambient temperature | -40...+55 | °C | Higher ambient temperature allowed as long as $T_{case-max}$ is not exceeded. |
| Starting Ambient temperature | -40...+55 | °C | |
| $T_{case-max}$ | 80 | °C | Maximum temperature measured at T_{case} -point |
| $T_{case-life}$ | 70 | °C | Measured at T_{case} -point |
| Maximum housing temperature | 110 | °C | In case of a failure |
| Relative humidity | 10...90 | % | Non-condensing |

Storage temperature and humidity

| Specification item | Value | Unit | Condition |
|---------------------|-----------|------|----------------|
| Ambient temperature | -40...+80 | °C | |
| Relative humidity | 5...95 | % | Non-condensing |

Lifetime

| Specification item | Value | Unit | Condition |
|--------------------|---------|-------|--|
| Driver lifetime | 100,000 | hours | Measured temperature at T_{case} -point is $T_{case-life}$. Maximum failures = 10% |



Programmable features

| Specification item | Value | Remark | Condition |
|---------------------------------------|---------------------|----------------------|---|
| Set output current (AOC) | Programmable, Rset1 | See Design-in guide. | Default output current: = 700 mA |
| LED module temperature derating (MTP) | Yes | | |
| Constant Lumen Over Lifetime (CLO) | Yes | | |
| DC emergency dimming (DCeDIM) | No | | Driver will not automatically dim when DC input voltage is applied. Set AOC value will be maintained. |
| Diagnostics | Yes | | |
| Adjustable Start-up Time AST | Yes | | |
| 1-10V minimum dim level | Yes | | |
| Integrated Dimmer | Yes | | |
| End Of Life indicator | Yes | | |

Features

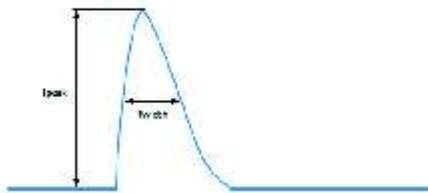
| Specification item | Value | Remark | Condition |
|---|-------|--------|----------------------|
| Open load protection | Yes | | Automatic recovering |
| Short circuit protection | Yes | | Automatic recovering |
| Over power protection | Yes | | Automatic recovering |
| Hot wiring | No | | |
| Suitable for fixtures with protection class | I | | per IEC60598 |
| Over temperature protection driver | Yes | | Automatic recovering |
| Overheating protection | Yes | | Automatic recovering |

Certificates and standards

| Specification item | Value |
|-----------------------------------|---------------------------------|
| Approval marks | CB / CCC / CE / CSA / ENEC / UL |
| Ingress Protection classification | |

Inrush current

| Specification item | Value | Unit | Condition |
|---------------------------|----------|---------|--|
| Inrush current I_{peak} | 108 | A | Input voltage 230V |
| Inrush current T_{peak} | 140 | μ s | Input voltage 230V, measured at 50% I_{peak} |
| Drivers / MCB 16A type B | ≤ 7 | pcs | |



| MCB | Rating | Relative number of LED drivers |
|-----|--------|--------------------------------|
| B | 10A | 63% |
| B | 13A | 81% |
| B | 16A | 100% (stated in datasheet) |
| B | 20A | 125% |
| B | 25A | 156% |
| C | 10A | 104% |
| C | 13A | 135% |
| C | 16A | 170% |
| C | 20A | 208% |
| C | 25A | 260% |

Driver touch current / protective conductor current

| Specification item | Value | Unit | Condition |
|---------------------------------------|-------|---------|---|
| Typical touch current (ins. Class II) | < 0.7 | mA peak | Acc. IEC61347-1, LED module contribution not included |

Surge immunity

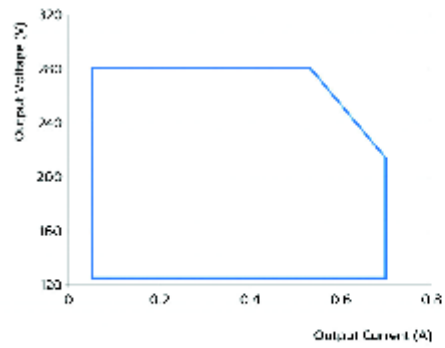
| Specification item | Value | Unit | Condition |
|-------------------------------------|-------|------|---|
| Mains surge immunity (diff. mode) | 3 | kV | Acc. IEC61000-4-5, 2 Ohm, 1.2/50us, 8/20us |
| Mains surge immunity (comm. mode) | 3 | kV | Acc. IEC61000-4-5, 12 Ohm 1.2/50us, 8/20us |
| Control surge immunity (diff. mode) | 0.5 | kV | Acc. IEC61000-4-5, 2 Ohm, 1.2/50us, 8/20us |
| Control surge immunity (comm. mode) | 3 | kV | Acc. IEC61000-4-5, 12 Ohm 1.2/50us, 8/20us |
| DAI surge immunity (comm. mode) | 3 | kV | DAI - U/N acc. IEC61000-4-5, 12 Ohm, 1.2/50us, 8/20us |

Additional information

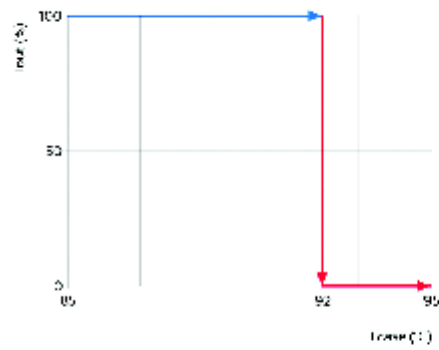
| Specification item | Default setting | Remark | Condition |
|--------------------|-----------------|--------|-----------|
| AOC | 700 | mA | |
| CLO | OFF | | |
| MTP | ON | | |
| Dynadimmer | OFF | | |
| EOL | OFF | | |
| I-10V | ON | | |

Graphs

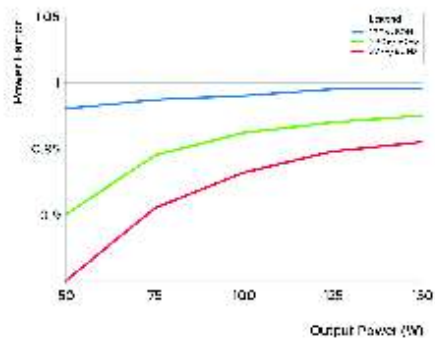
Operating window



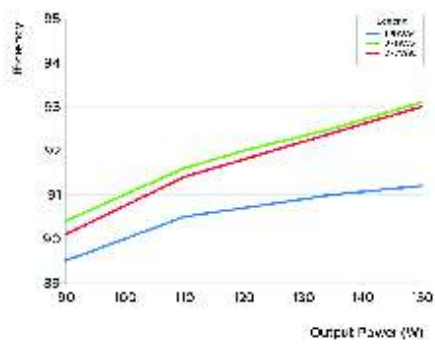
Thermal Guard



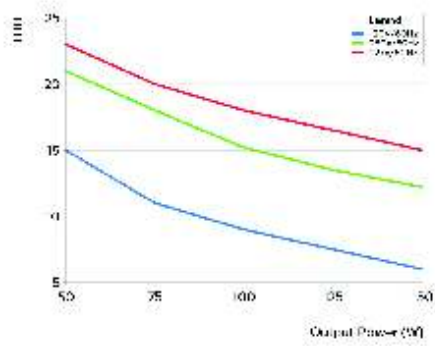
Power factor versus output power



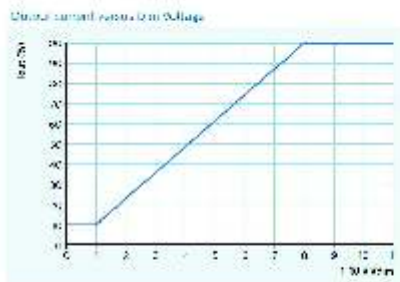
Efficiency versus output power



THD versus output power



I_{set} as function of 1-10V Interface



Notes

Installation & Application Notes:

- 1: By factory default, the 1-10V interface is enabled and DALI interface is disabled. These controls are mutually exclusive.
- 2: Integrated Dimmer cannot be overruled by DALI. These controls are mutually exclusive.
- 3: Driver is for built-in use only and must not be exposed to the elements such as snow, water and ice or to any other chemical agent which can be expected to have an adverse effect on the driver (e.g. corrosive environments). It is the responsibility of both luminaire manufacturer and installer to prevent exposure. Common sense needs to be used in order to define the proper luminaire or application IP rating.
- 4: Suitable for UL Damp & Dry locations
- 5: IEC driver housing is not allowed to be connected to accessible insulation Class II luminaire parts. Driver is suitable for insulation Class I application only.
- 6: Standard lead length on all wires: 500 +/- 30mm solid copper. Insulation rating: 105°C/600V.
- 7: Driver complies with the requirements of UL, CSA, CE, ENEC, CISPR15, FCC 47CFR15 Class A



© 2017 Philips Lighting Holding B.V. All rights reserved.

This document contains information relating to the Philips Lighting portfolio, intended for companies who may be interested in developing their product offering. Note that the information provided is subject to change. Philips Lighting does not give any representation or warranty as to the accuracy or completeness of the information included herein and shall not be liable for any action in reliance thereon. The information presented in this document is not intended as any commercial offer and does not form part of any quotation or contract.

Date of release: April 24, 2017

www.philips.com/technology

ANEXO VI

Proforma con los materiales de la Empresa CORPORACION WOLF:



WOLF ELECTRONICS

E-mail: info@corpwolf.com Celular: 0996459504

PROFORMA

FECHA: 24 de Agosto 2017.

No. 1545

Cliente: **EMMIANORTE**

RUC:

Teléfono: 062641288

Vendedor:
Corporación Wolf

| CANT | DESCRIPCION | IMG | VALOR | SUBTOTAL |
|------|---|---|-------|----------|
| 100 | ARDUINO UNO R3 + CABLE USB |  | 9,00 | 900,00 |
| 100 | MODULO XBEE SERIE 2 |  | 35,00 | 3500,00 |
| 100 | SHIELD XBEE V03 PARA ARDUINO |  | 7,00 | 700,00 |
| 100 | MODULO MOSFET (IRF520) |  | 3,50 | 350,00 |
| 100 | MODULO SENSOR DE CORRIENTE (ACS712ELCTR-05B-T) |  | 4,50 | 450,00 |
| 100 | MODULO RELE CON OPTOACOPLADOR |  | 4,50 | 450,00 |
| 3 | MODULO XBEE XPLORER MINI USB |  | 14,00 | 1400,00 |

Dirección: Quito, Av. General Rumiñahui 775 e Italo. Junto al PRODUBANCO, 2° PISO.

Ambato, Frente a la Universidad Técnica de Ambato (Campus Huachi).

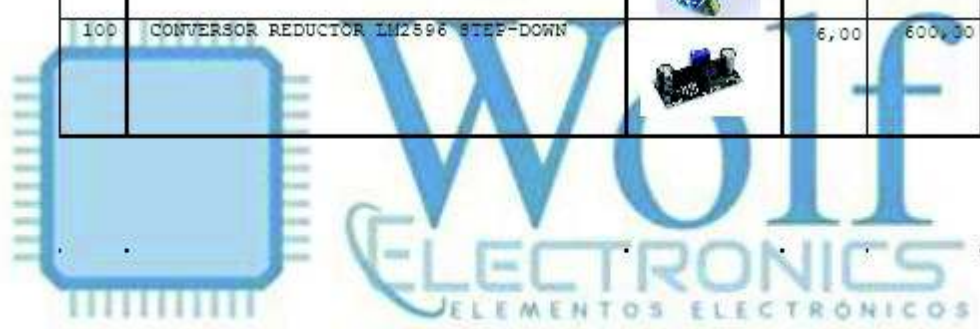
Web: www.corpwolf.com

Email: info@corpwolf.com Telf: (02)2861440

Celular: 0983558396 - 0996779364



| | | | | |
|-----|---|---|-------|---------|
| 1 | MÓDULO GSM/GPRS SIM 900 |  | 33,00 | 33,00 |
| 100 | MINI PANEL SOLAR 12V 1,5W |  | 6,50 | 650,00 |
| 100 | BATERÍA RECARGABLE 12V - 1800MAH |  | 14,50 | 1450,00 |
| 100 | MÓDULO DE CARGA DE BATERÍA STEP-DOWN DC-DC MPPT 12V |  | 6,50 | 650,00 |
| 100 | CONVERSOR REDUCTOR LM2596 STEP-DOWN |  | 6,00 | 600,00 |



VALIDEZ: 15 DIAS

TIEMPO DE ENTREGA: 21 DIAS APARTIR DE ANTICIPO

FORMA DE PAGO: 50% ANTICIPO Y 50% CONTRA ENTREGA

GARANTIA: 12 MESES POR DEFECTO DE FABRICA

SUBTOTAL

IVA 12%

TOTAL

11133,00

1335,96

12468,96

Corporación Wolf.

Firma autorizada

Dirección: Quito, Av. General Rumiñahui 775 e Ilaló. Junto al PRODUBANCO, 2º PISO.
 Ambato, Frente a la Universidad Técnica de Ambato (Campus Huachi).
 Web: www.corpwolf.com
 Email: info@corpwolf.com Telf: (02)2861440
 Celular: 0983558396 - 0996779364

ANEXO VII

| Costos del Servicio de Alumbrado Público General - 2015 | | | | | | | |
|---|---------------------------|---------------------------|------------------------|---------------------|------------------------|-----------------------------|----------------------|
| EMPRESA | DISTRIBUIDORA | COSTO MEDIO (cUSD/kWh) | COSTO ENERGÍA (USD) | COSTO AO&M (USD) | COSTO CALIDAD (USD) | COSTO EXPANSIÓN (USD) | COSTO TOTAL (USD) |
| Empresas Eléctricas | Ambato | 12,36 | 3.786.214 | 900.529 | 407.859 | 1.221.463 | 6.316.064 |
| | Azogues | 18,16 | 792.259 | 219.943 | 97.753 | 423.145 | 1.533.100 |
| | Centro Sur | 13,14 | 6.169.478 | 2.160.449 | 1.065.551 | 824.885 | 10.220.363 |
| | Cotopaxi | 10,74 | 1.907.848 | 371.220 | 186.786 | 301.412 | 2.767.267 |
| | Norte | 9,51 | 3.581.145 | 701.619 | 252.255 | 23.014 | 4.558.034 |
| | Quito | 11,23 | 14.830.566 | 4.050.873 | 2.085.335 | 2.367.820 | 23.334.594 |
| | Riobamba | 10,22 | 2.290.069 | 374.221 | 137.122 | 64.379 | 2.865.791 |
| | Sur | 10,86 | 2.208.137 | 575.291 | 183.035 | 58.527 | 3.024.990 |
| | Galápagos | 34,43 | 141.163 | 270.754 | 24.526 | 98.316 | 534.759 |
| Total Empresas Eléctricas | | 11,85 | 35.706.878 | 9.624.899 | 4.440.222 | 5.382.962 | 55.154.961 |
| Corporación Nacional de Electricidad | CNEL EP - Bolívar | 11,30 | 1.354.453 | 152.949 | 67.978 | 120.954 | 1.696.334 |
| | CNEL EP - El Oro | 9,29 | 5.262.185 | 984.080 | 449.372 | 169.928 | 6.865.565 |
| | CNEL EP - Guayas Los Ríos | 11,85 | 5.472.826 | 1.216.978 | 298.196 | 1.682.076 | 8.670.076 |
| | CNEL EP - Esmeraldas | 13,58 | 2.341.608 | 434.770 | 127.270 | 926.804 | 3.830.452 |
| | CNEL EP - Los Ríos | 13,38 | 1.315.903 | 145.519 | 64.675 | 779.636 | 2.305.733 |
| | CNEL EP - Manabí | 10,49 | 7.965.673 | 1.350.161 | 390.662 | 1.725.077 | 11.431.573 |
| | CNEL EP - Milagro | 11,55 | 2.458.219 | 595.293 | 119.103 | 554.627 | 3.727.242 |
| | CNEL EP - Santa Elena | 12,56 | 2.646.566 | 761.734 | 235.865 | 879.855 | 4.524.020 |
| | CNEL EP - Santo Domingo | 11,18 | 2.450.468 | 791.738 | 199.840 | 169.057 | 3.611.102 |
| | CNEL EP - Sucumbios | 9,79 | 1.652.786 | 215.491 | 71.317 | - | 1.939.594 |
| CNEL EP - Guayaquil | 10,78 | 9.665.766 | 3.385.153 | 129.376 | 867.851 | 14.048.146 | |
| Total CNEL EP | | 11,05 | 42.586.454 | 10.033.867 | 2.153.653 | 7.875.864 | 62.649.838 |
| Total Distribuidoras | | 11,29 | 78.293.332 | 19.658.766 | 6.593.874 | 13.258.826 | 117.804.799 |

Aprobado con Resolución Nro. 022A/15

ANEXO VIII

Detalles técnicos del mini panel solar 12V – 1,5W

| | | |
|-------------------------|---|------------------------|
| Nombre | Mini Panel Solar (12V - 1,5W o 6V – 3W) | |
| Descripción | Panel solar pequeño, ideal para proyectos electrónicos o prototipos de bajo consumo de energía. | |
| Especificaciones | Material: | Silicio monocristalino |
| | Tamaño: | 145x145mm o 115x85mm |
| | Capacidad nominal: | 6V o 12V |
| | Potencia: | 3W o 1.5W |
| | Eficiencia de salida: | Alta |
| | Tasa de conversión: | Alta |

Detalles técnicos del módulo de carga de batería Step-Down DC-DC MPPT

| | | |
|-------------------------|--|------------------|
| Nombre | Módulo de carga de batería Step-Down DC-DC MPPT 12V | |
| Descripción | Modulo controlador de carga de baterías alimentadas desde paneles solares. | |
| Especificaciones | Voltaje de alimentación: | 6 - 36V |
| | Temperatura de funcionamiento: | -40 ~ 85 C |
| | Aplicación: | Step Down Module |
| | Voltaje de salida: | 1.25V – 32V |
| | Voltaje de salida por defecto: | 5V |
| | Corriente de salida: | 0.05 - 5A |
| | Corriente de salida por defecto: | 3A |
| | Eficiencia: | >95% |
| | Protección contra cortocircuito: | Si |
| | Tamaño: | 60x31x22 mm |
| | Frecuencia de operación: | 180kHz |

Detalles técnicos de la batería recargable

| | | |
|-------------------------|---|-----------------------------|
| Nombre | Batería recargable 12V - 1800mAh | |
| Descripción | Batería de gran capacidad, largo tiempo de vida útil, con protección de seguridad contra corto circuitos. Tamaño pequeño, fácil de usar, se puede utilizar para accionar transmisores inalámbricos de gran alcance, dispositivos electrónicos, etc. | |
| Especificaciones | Capacidad: | 1800 mAh |
| | Tipo de batería: | Batería Recargable de Li-on |
| | Voltaje de entrada: | 12.6 V |
| | Tensión de salida: | 10.8 ~ 12.6 DC |
| | Corriente de salida: | 1-2.4A |
| | Tamaño: | 8x5x2 cm |

Detalles técnicos del módulo de carga de batería tipo lipo

| | | |
|-------------------------|--|---------------|
| Nombre | Módulo de carga de batería tipo lipo | |
| Descripción | Modulo controlador de carga de baterías tipo lipo alimentadas desde paneles solares. | |
| Especificaciones | Voltaje de alimentación: | 4,5V – 5,5V |
| | Temperatura de funcionamiento: | -10 ~ 85 C |
| | Voltaje de salida: | 4,2V +/- 1.5% |
| | Corriente de salida Max: | 1A |
| | Protección sobretensión: | 4,2V |
| | Protección sobredescarga: | 2,4V |
| | Protección contra sobrecorriente: | 3A |
| | Tamaño: | 60x31x22 mm |

Detalles técnicos de la batería recargable tipo lipo

| | | |
|-------------------------|---|-----------------------------|
| Nombre | Batería recargable 11,1V - 1350mAh | |
| Descripción | La batería LiPo (polímero de iones de litio) de descarga alta es una excelente manera de alimentar cualquier proyecto que requiera una batería pequeña con mucha fuerza. La velocidad de descarga es lo suficientemente alta como para alimentar componentes electrónicos y motores pequeños. | |
| Especificaciones | Capacidad: | 1350 mAh |
| | Tipo de batería: | Batería Recargable de Li-on |
| | Voltaje de entrada: | 3.7 V |
| | Tensión de salida: | 11.1 V DC |
| | Energía: | 14,99 Wh |
| | Velocidad de Descarga Continua: | 35C |

Detalles técnicos del módulo regulador de voltaje Step-Down DC-DC

| | | |
|-------------------------|--|----------------------|
| Nombre | Módulo Regulador de voltaje Step-down DC-DC (LM2596) | |
| Descripción | <p>Son módulos sencillos que permite incluir en diseños electrónicos una fuente Step Down lista para usar con una entrada IN y una salida OUT.</p> <p>Se puede regular la tensión de salida con el potenciómetro integrado. Este módulo soporta tensiones de entrada entre 4 y 40 VDC y puede dar una la salida desde 1.5V hasta unos 37 VDC, con un rendimiento excelente.</p> <p>Pueden proporcionar 2 amperios estables y llegar puntualmente hasta los 3 amperios, pero para mantenerlos se necesitará de un disipador.</p> <p>Son recomendables para proyectos en los que se quiera usar Arduino con batería, porque se puede poner la salida a 5V o 3.3V y alimentar directamente los pines Vin del Arduino, con lo que se extenderá mucho la duración de estas.</p> | |
| Especificaciones | Tipo: | Regulador de voltaje |
| | Regulador: | LM2596S DC-DC Buck |
| | Tamaño: | 43mm x 21mm x 14mm |
| | Voltaje de salida: | 1.25 V ~ 35 V |
| | Voltaje de entrada: | 3.2 V ~ 40 V |
| | Corriente de salida: | 3A |
| | Eficacia de conversión: | 92% |
| | Frecuencia de conmutación: | 65kHz |
| | Temperatura de trabajo: | -45 ~ + 85 C |

ANEXO IX

Detalles técnicos de Arduino Uno R3

| | | |
|-------------------------|---|-----------|
| Nombre | Arduino Uno R3 | |
| Descripción | <p>Este Arduino utiliza el microcontrolador ATmega328. No se necesitan drivers para sistemas operativos como Linux o Mac, mientras que para Windows es necesario descargar el driver junto con el IDE desde la página oficial de Arduino.</p> <p>La tarjeta Arduino Uno R3 añade los pines SDA y SCL cercanos al AREF. Agrega dos nuevos pines cerca del pin RESET, el IOREF que permite a los shields adaptarse al voltaje brindado desde la propia tarjeta. El otro pin se encuentra reservado para propósitos futuros, por lo que se encuentra desconectado. La tarjeta trabaja con todos los shields existentes y se podrá adaptar a los nuevos shields utilizando los pines adicionales.</p> <p>Arduino es una plataforma computacional física open-source basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje C++ en Processing/Wiring.</p> | |
| Especificaciones | Tipo de microcontrolador: | ATmega328 |
| | Voltaje de entrada: | 7-12V |
| | Nº pines digitales I/O: | 14 |
| | Nº pines analógicos IN: | 6 |
| | Nº pines analógicos O (PWM): | 6 |
| | Tamaño de memoria flash: | 32k |
| | Velocidad de reloj: | 16MHz |

Detalles técnicos del sensor de corriente

| | | |
|-------------------------|--|------------------------------------|
| Nombre | Sensor de corriente - Efecto Hall (ACS712) | |
| Descripción | <p>Este es un sensor de corriente por efecto Hall preciso para medir corrientes en AC y DC, para funcionar en ambientes comerciales o industriales. Compatible con las populares plataformas electrónicas como Arduino y Raspberry Pi.</p> <p>Funciona al transformar el campo magnético debido al paso de corriente a través de un conductor de cobre interno en el sensor en voltaje variable.</p> <p>Este sensor se lo puede encontrar en 3 modelos de acuerdo con la capacidad de amperaje a operar: ACS712ELCTR-05B-T para medir corrientes hasta 5A, ACS712ELCTR-20A-T para medir corrientes hasta 20A, ACS712ELCTR-30A-T para medir corrientes hasta 30A.</p> | |
| Especificaciones | Tipo de sensor: | efecto hall de bajo offset |
| | Tamaño: | 3,1 x 1.4 cm |
| | Tiempo de respuesta: | 5us |
| | Voltaje de salida: | Analog output 66mV/A hasta 185mV/A |
| | Voltaje de trabajo: | 4,5V – 5,5V |
| | Salida de voltaje sin corriente: | Vcc/2 |
| | Ancho de banda: | 80kHz |
| | Error total de salida: | 1,5% a T=25°C |
| | Resistencia interna: | 1,2mohm |
| | Sensitividad de salida: | 66 a 185mV/A |

Detalles técnicos del módulo Mosfet

| | | |
|------------------|---|------------------------|
| Nombre | Módulo MOSFET (IRF520) | |
| Descripción | <p>Este módulo es capaz de manejar una carga de hasta 9A. Soporta control por PWM. Puede manejar cargas como: válvulas solenoide 24V DC, motores DC, Celdas Peltier de hasta 6A, Luces LED de alta potencia, etc.</p> <p>Para activar el mosfet se debe de enviar 5V al Gate del Mosfet, esto permitirá que la corriente fluya a través de la carga y esta se active.</p> | |
| Especificaciones | Tipo de mosfet: | IRF520 |
| | Tipo: | Regulador de voltaje |
| | Tamaño: | 33 x 24mm |
| | Peso: | 10g |
| | Tipo de puertos: | Nivel Digital |
| | Voltaje de salida: | 0-24 V DC |
| | Voltaje de control: | 3,3V - 5V TTL |
| | Corriente nominal: | 9A |
| | Corriente máxima: | 6A |
| | Corriente de carga de salida: | <5A |
| | Uso de disipador: | Para cargas mayor a 6A |

Detalles técnicos del módulo Relé

| | | |
|------------------|---|---------------|
| Nombre | Módulo Relé 5Vdc - 125Vac (1 canal) | |
| Descripción | <p>Módulo de relé de 1 canal, con el aislamiento fotoeléctrico, el módulo tiene dos posiciones, puede ser abierto o cerrado por disparo, contiene leds indicadores de alimentación y acción del relé. Tiene interfaces abiertas de carga máxima: CA 250 V/10A, DC 30 V/10A.</p> | |
| Especificaciones | Voltaje de alimentación: | 5V – 12V |
| | Señal de control: | PWM (0V - 5V) |
| | Canales: | 1 |
| | Voltaje de trabajo a la salida: | 125V – 250V |

ANEXO X

Detalles técnicos del módulo Xbee serie 2

| | | |
|-------------------------|--|---|
| Nombre | Módulo XBee Serie 2 (XBee 2mW) | |
| Descripción | Son pequeños dispositivos que pueden comunicarse entre sí de manera inalámbrica. Son fabricados por Digi International, los cuales ofrecen una gran variedad de combinaciones de hardware, protocolos, antenas y potencias de transmisión. | |
| Especificaciones | Tipo de Interfaz de antena: | Interfaz UFL |
| | Potencia de transmisión: | 2 mW |
| | Distancia de transmisión al aire libre: | 120 m en interiores y 1.2 km en línea de vista libre: |
| | Velocidad: | 250 kbps |
| | Antena: | Built-in |
| | Voltaje de operación: | 2.1 – 3.6V |

Detalles técnicos del Xbee explorer mini USB

| | | |
|-------------------------|---|---------|
| Nombre | XBee Explorer mini USB | |
| Descripción | Es un adaptador USB para configuración directa de los módulos Xbee de todas las series. | |
| Especificaciones | Protocolo de comunicación: | USB 2.0 |
| | Voltaje de funcionamiento: | 3.3V |
| | Voltaje de I/O compatible: | 3.3V |

Detalles técnicos de la placa shield Xbee

| | | |
|-------------------|---|-----------|
| Nombre: | Shield Xbee V03 | |
| Descripción: | Es un adaptador Xbee USB a serial equipado con (20pin 2.0mm) sockets. Integrada FT232RL IC, puede ser utilizado para la programación o la comunicación con el tablero que básico para Arduino, pero sin interfaz USB. | |
| Especificaciones: | Voltaje de funcionamiento: | 3.3V y 5V |
| | Voltaje de I/O compatible: | 3.3V y 5V |
| | Protocolo de comunicación: | USB 2.0 |

Detalles técnicos del módulo GSM/GPRS

| | | |
|------------------|---|---------------------------------|
| Nombre | Módulo GSM/GPRS SIM900 para Arduino | |
| Descripción | The Shield GPRS se basa en el módulo SIM900 SIMCOM compatible con originales y sus clones. The Shield GPRS proporciona una forma de comunicarse mediante la red de telefonía celular GSM. El shield permite lograr SMS, MMS, GPRS y de Audio a través de UART mediante el envío de comandos AT (GSM 07.07, 07.05 y SIMCOM AT). La pantalla tiene 12 Gpio, 2 pwm y un ADC. | |
| Especificaciones | Voltaje de operación: | 3.3V - 5V |
| | Bandas de operación GSM: | Quad-Band 850/900/1800/1900 MHz |
| | Tipo GPRS: | Multi-slot clase 10/8 |
| | Clase GPRS: | Clase de estación móvil B |
| | Medio de comandos de control: | Comandos AT Estándar |
| | Tipo de servicio de mensajería: | Mensajes cortos |
| | Posee TCP/UDP: | Si |
| | Admite RTC: | Si |
| | Temperatura de operación: | -40 ° C a + 85 ° C |
| | Tamaño: | 8.5x5.7x2 cm (aprox) |
| | Conector RF: | U. FL. |
| | Antena: | 2.4G |
| | Cable: | Pigtail IPX a SMA |

ANEXO XI

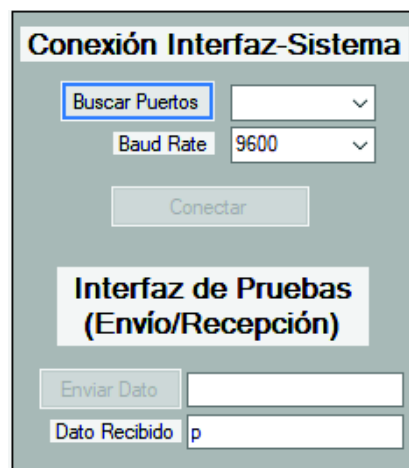
Manual de Usuario de la interfaz de monitoreo del prototipo desarrollado en Visual Studio 2017.

El software de control y monitoreo para el sistema de telegestión está compuesto de varias secciones:

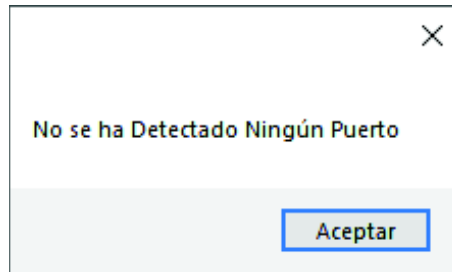


Establecer la Conexión con el Sistema de Telegestión

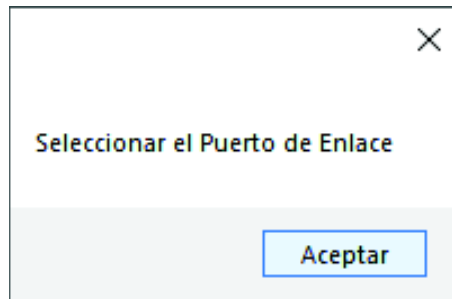
Para conectar el software con el sistema de telegestión se debe buscar los puertos disponibles de conexión en la sección “Conexión Interfaz-Sistema”, con el botón **“Buscar Puertos”**:



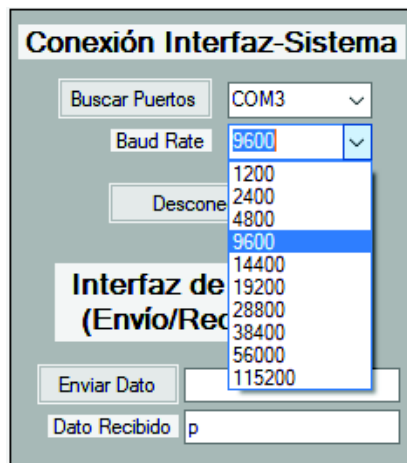
Si se muestra el siguiente mensaje, significa que ningún puerto está conectado, para solucionar esto verificar las conexiones del sistema de comunicación en los puertos de la máquina:



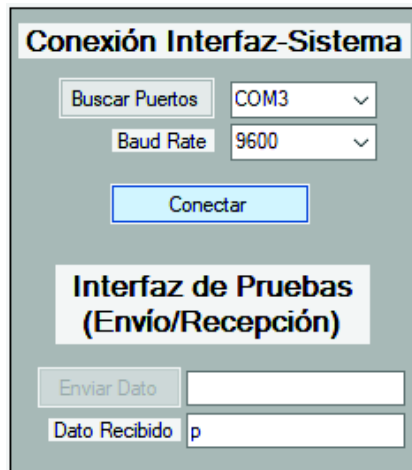
Cuando se observe el siguiente mensaje, significa que se tiene elementos conectados a uno o más puertos de la máquina:



El siguiente paso es seleccionar el puerto correcto de comunicación junto con el **“Baud Rate”** adecuado, que debe ser el mismo al que fueron programados el microcontrolador y el sistema de comunicación del prototipo:

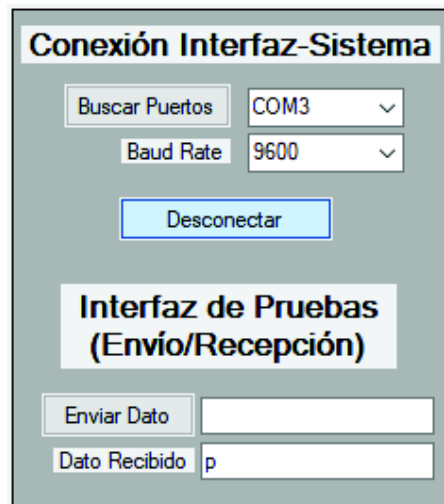


El último paso es hacer clic en el botón **“Conectar”**:



The screenshot shows a software window titled "Conexión Interfaz-Sistema". At the top, there is a "Buscar Puertos" button and a dropdown menu showing "COM3". Below that is a "Baud Rate" label and a dropdown menu showing "9600". A blue button labeled "Conectar" is centered below these settings. In the middle of the window, there is a section titled "Interfaz de Pruebas (Envío/Recepción)". At the bottom, there is an "Enviar Dato" button next to an empty text input field, and a "Dato Recibido" label next to a text input field containing the letter "p".

Para desconectar la interfaz de comunicación Máquina-Sistema hay que hacer clic en el botón **“Desconectar”**:



This screenshot is identical to the one above, showing the "Conexión Interfaz-Sistema" window. The only difference is that the blue button in the center is now labeled "Desconectar".

Encendido y Apagado de las luminarias del sistema

Para encender las luminarias basta con hacer clic en el botón **“Encender”**:

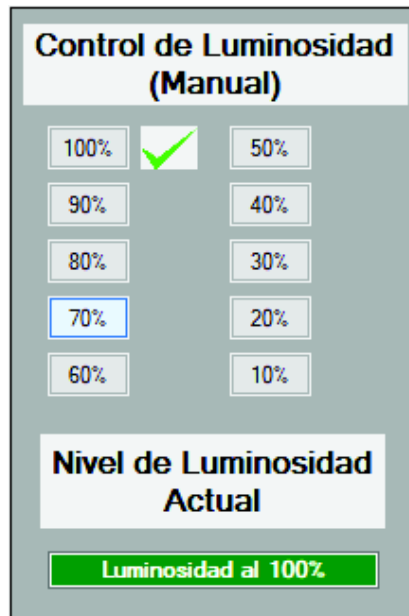


The screenshot shows a software window titled "On / Off del Sistema". It contains two buttons: a blue button labeled "Encender" and a grey button labeled "Apagar". To the right of these buttons is a red power button icon.

Y para apagar las luminarias basta con hacer clic en el botón **“Apagar”**:



La variación del nivel de iluminación se realiza con los botones de la sección “Control de Luminosidad (Manual)”, en la misma sección se muestra en palabras el nivel de luminosidad actual:



En la sección “Parámetros Fundamentales por cada Luminaria” se muestran los valores de energía y costo al máximo, energía y costo al nivel anterior, el tiempo que permaneció activo el nivel anterior y el ahorro energético y económico, todos estos datos corresponden a cada luminaria activa en el sistema:

| Parámetros Fundamentales por cada Luminaria | | | |
|---|----------------------|--------------|----------------------|
| Energía al 100% [kWh] | 0,000262758027777778 | Tiempo [h] | 0,001592473111111111 |
| Costo al 100% [\$] | 2,49882777777778E-05 | Ahorro [kWh] | 0,000183134388888889 |
| Energía al 20% [kWh] | 7,96236388888889E-05 | Ahorro [\$] | 1,74160555555556E-05 |
| Costo al 20% [\$] | 7,57219444444444E-06 | | 14 |

En la sección “Cambio Realizado” se muestra el número de evento y su respectiva descripción, con fondo rojo cuando el evento es el apagado de la luminaria y en fondo verde cuando el evento es el encendido de la luminaria, en color celeste cuando el evento es cambio de intensidad luminosa:

Cambio Realizado

66 Luminaria 10 Apagada

En la sección “Monitoreo de Luminarias” se muestra en texto el estado de cada luminaria activa en el sistema, con fondo rojo cuando la luminaria se encuentra apagada y en fondo verde cuando la luminaria se encuentra encendida:

Monitoreo de Luminarias

Luminaria 3 | Apagada

Luminaria 10 | Apagada

En la sección “Bases de Datos” se muestran el “Registro de Eventos” y el “Registro de Energía y Costos”

| Registro de Eventos | | | | |
|---------------------|---------------------------|------------------------|---------------------------|--|
| ID_OPER | ELEMENTO | DETALLE DE OPERACION | FECHA Y HORA DE OPERACION | |
| 1 | Luminaria 10 Encendi... | Luminaria 10 Encendida | 02/02/2018 08:47:08 a. m. | |
| 11 | Luminosidad | Luminosidad al 30% | 02/02/2018 08:49:21 a. m. | |
| 13 | Luminosidad | Luminosidad al 60% | 02/02/2018 08:49:26 a. m. | |
| 15 | Luminosidad | Luminosidad al 100% | 02/02/2018 08:49:29 a. m. | |
| 17 | Luminaria 3 Apagada | Luminaria 3 Apagada | 02/02/2018 08:49:38 a. m. | |
| 18 | Luminaria 10 Apagada | Luminaria 10 Apagada | 02/02/2018 08:49:38 a. m. | |
| 19 | Luminaria 3 Encendi... | Luminaria 3 Encendida | 02/02/2018 08:49:41 a. m. | |
| 2 | Luminaria 10 Apagad... | Luminaria 10 Apagada | 02/02/2018 08:47:09 a. m. | |
| 20 | Luminaria 10 Encendi... | Luminaria 10 Encendida | 02/02/2018 08:49:42 a. m. | |
| 21 | Luminosidad | Luminosidad al 70% | 02/02/2018 08:49:45 a. m. | |

| Registro de Energía y Costos | | | | | | |
|------------------------------|-------------------|--------------|--------------------------|--------------------------|--------------------------|-----------------------|
| ID_OP | EVENTO | TIEMPO [h] | ENERGIA [kWh] | COSTO [\$] | AHORRO ENERGIA [kWh] | AHORRO ECONOMICO [\$] |
| 1 | Energia al 100... | 0,0009623... | 0,00015878519444444... | 1,510044444444444E-05... | 0 | 0 |
| 10 | Energia al 40... | 0,0011233... | 8,874669444444444E-05... | 8,439805555555555E-06... | 9,661033333333333E-05... | 9,187638888888889E-06 |
| 11 | Energia al 10... | 0,0015932... | 5,576413888888889E-05... | 5,303166666666667E-06... | 0,00020712405555555... | 1,969747222222222E-05 |
| 12 | Energia al 100... | 0,0011348... | 0,00018724455555555... | 1,780694444444444E-05... | 0 | 0 |
| 2 | Energia al 100... | 0,0317131... | 0,00523267069444444... | 0,00049762697222222... | 0 | 0 |
| 3 | Energia al 100... | 0,0012562... | 0,00020727413888888... | 1,971175E-05 | 0 | 0 |
| 4 | Energia al 10... | 0,0011194... | 3,917908333333333E-05... | 3,725916666666667E-06... | 0,00014552241666666... | 1,383916666666667E-05 |
| 5 | Energia al 30... | 0,0021642... | 0,00013851147222222... | 1,317241666666667E-05... | 0,00021858847222222... | 2,078775E-05 |

Para exportar los valores a un archivo de Excel se tiene dos botones, uno por cada tabla de la base de datos:

