



# POSGRADOS

## MAESTRÍA EN

---

# ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE  
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

DESARROLLO DE UN SISTEMA DE GESTIÓN PARA EL CONTROL DE  
ILUMINACIÓN Y CLIMATIZACIÓN EN EDIFICACIONES, APLICADO  
AL EDIFICIO "PADRE JUAN BOTTASSO" DE LA UNIVERSIDAD  
POLITÉCNICA SALESIANA, SEDE CUENCA

AUTOR:

LUIS ALBERTO RIOFRÍO PEÑA

DIRECTOR:

JUAN DIEGO JARA SALTOS

CUENCA - ECUADOR  
2021

***Autor:***



***Luis Alberto Riofrío Peña***

Ingeniero en Electrónica y Telecomunicaciones.  
Candidato a Magíster en Electrónica y Automatización,  
Mención en Informática Industrial por la Universidad  
Politécnica Salesiana - Sede Cuenca.  
luis\_riofrio4@hotmail.com

***Dirigido por:***



***Juan Diego Jara Saltos***

Ingeniero Electrónico.  
Métodos matemáticos y simulación en ingeniería.  
Telemática  
jjaras@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

**DERECHOS RESERVADOS**

©2021 Universidad Politécnica Salesiana.  
CUENCA – ECUADOR – SUDAMÉRICA  
RIOFRÍO PEÑA LUIS ALBERTO

***DESARROLLO DE UN SISTEMA DE GESTIÓN PARA EL CONTROL DE ILUMINACIÓN Y CLIMATIZACIÓN EN EDIFICACIONES, APLICADO AL EDIFICIO “PADRE JUAN BOTTASSO” DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE CUENCA***

# Índice general

<b>Índice de Figuras</b>	<b>V</b>
<b>Índice de Tablas</b>	<b>VII</b>
<b>Resumen</b>	<b>VIII</b>
<b>Abstract</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción general del problema.....	4
1.2. Objetivos.....	9
1.2.1. Objetivo general.....	9
1.2.2. Objetivos específicos.....	9
1.3. Contribuciones.....	10
1.4. Organización del manuscrito.....	10
<b>2. 2. Sistema de gestión de edificios</b>	<b>11</b>
2.1. Estado del Arte.....	12
2.1.1. Arquitectura.....	12
2.1.2. Redes de comunicaciones.....	13
2.1.3. Sensores y actuadores.....	13
2.1.4. Modelo de dispositivos.....	14
2.1.5. Comandos, retroalimentación y etiquetas.....	15
2.1.6. Funcionalidad del sistema de gestión.....	15
2.2. Tecnologías para automatización de edificios.....	19
2.2.1. BACnet.....	19
2.2.2. KNX.....	20
2.2.3. LonWorks.....	20
2.2.4. Dali.....	20
2.2.5. Modbus.....	21

2.2.6. Zigbee .....	21
2.2.7. Zwave.....	22
2.2.8. LoRaWan .....	22
2.2.9. Buspro.....	22

**3. Desarrollo de un sistema de automatización de edificios (BAS)**

**28**

3.1. Desarrollo del BAS .....	29
3.1.1. Servicios de un edificio con un BAS .....	30
3.1.2. Integración de sistemas .....	32
3.1.3. Automatización y control.....	33
3.1.4. Arquitectura de un sistema BAS.....	33
3.2. Desarrollo .....	35
3.2.1. Hardware.....	42
3.2.2. Software .....	48
3.2.3. Interfaz de usuario y programación.....	51
3.2.4. iRidium Studio .....	52
3.2.5. iRidium Server.....	66
3.3. Resultados y Discusión.....	89
3.4. Conclusiones.....	94

**A. Anexo: Planos de AutoCAD para iluminación y climatización**

**96**

**B. Anexo II: Presupuesto**

**99**

# Índice de Figuras

2.1. Arquitectura de un BAS. ....	12
3.1. Aspectos funcionales de un BAS. Desarrollo: autor .....	30
3.2. Arquitectura general de un BAS. Desarrollo: autor .....	34
3.3. Vista frontal del edificio “Padre Juan Bottasso”. ....	38
3.4. Vista posterior del edificio “Padre Juan Bottasso”. ....	39
3.5. Diseño en programa sketch up pro. ....	40
3.6. Vista 3D en programa sketch up pro. ....	41
3.7. Fotografía de la maqueta impresa. ....	41
3.8. Fotografía de la maqueta, servidor y equipos utilizados. ....	42
3.9. RCU Room Control Unit. ....	45
3.10. 750mA Power Supply Module. ....	46
3.11. DLP Touch. ....	46
3.12. Ceiling Mount 8in1 sensor. ....	47
3.13. Compatibilidad iRidium. Fuente: autor .....	51
3.14. Diagrama de flujo del programa iRidium. Fuente: autor .....	54
3.15. Componentes iRidium Panel Project. Fuente: autor .....	55
3.16. Interfaz de usuario. Fuente: autor .....	56
3.17. Creación iRidium Panel Project. Fuente: autor.....	57
3.18. Software iRidium Studio. Fuente: autor.....	58
3.19. Simulador en aplicación i3 pro. Fuente: autor .....	61
3.20. Aplicación i3 pro. Fuente: autor .....	61
3.21. Script para niveles de desplazamiento. Fuente: autor.....	63
3.22. Script pop-up para interfaz i3 pro. Fuente: autor.....	64
3.23. Script botón back para interfaz i3 pro. Fuente: autor .....	65
3.24. Control de comandos en interfaz i3 pro. Fuente: autor .....	65
3.25. Comunicación entre servidor, equipos y paneles. Fuente: autor	67
3.26. Crear proyecto servidor. Fuente: autor.....	69
3.27. Tokens del driver. Fuente: autor.....	70

3.28. Comandos de los equipos. Fuente: autor.....	71
3.29. Tags y retroalimentación. Fuente: autor .....	71
3.30. Horarios proyecto. Fuente: autor .....	72
3.31. Eventos. Fuente: autor .....	73
3.32. Rutinas del proyecto. Fuente: autor .....	74
3.33. Modificación de datos en el servidor. Fuente: autor.....	75
3.34. Intercambio de datos. Fuente: autor.....	79
3.35. Escenas. Fuente: autor .....	81
3.36. Activar escenas. Fuente: autor .....	82
3.37. Ubicación base de datos. Fuente: autor.....	83
3.38. Gráficas desde el servidor web. Fuente: autor.....	84
3.39. Sincronización servidor web e interfaz de usuario. Fuente: autor	84
3.40. iRidium Transfer. Fuente: autor .....	85
3.41. Scripts sensores y aire acondicionado. Fuente: autor .....	86
3.42. Lectura de temperatura y nivel de luxes en servidor. Fuente: autor .....	87
3.43. Notificaciones push. Fuente: autor .....	88
3.44. Notificaciones push en el script del proyecto. Fuente: autor.....	88
3.45. Consumo energético aproximado de iluminación. Fuente: autor	93
 B.1. Presupuesto sistema BAS edificio “Padre Juan Bottasso”.....	 100

# Índice de Tablas

2.1. Funcionalidades de un sistema BAS.....	16
2.2. Comparativa de tecnologías cableadas .....	24
2.3. Comparativa de tecnologías cableadas .....	25
2.4. Comparativa de tecnologías inalámbricas .....	26
3.1. Servicios en edificios .....	31
3.2. Arquitectura .....	50
3.3. Arquitectura .....	52
3.4. Arquitectura .....	77
3.5. Arquitectura .....	78

# Resumen

El presente trabajo describe la investigación aplicada en campo para desarrollar un sistema de gestión de edificios que permita el control de iluminación y climatización, con el objetivo de mejorar el desempeño energético y calidad de aire en el edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana sede Cuenca. Para ello se identifican los elementos que inciden en la iluminación y climatización determinando su consumo energético y corrientes de aire, buscando mantener niveles aceptables de confort y seguridad en entornos críticos. Se realiza la planimetría técnica para su implementación y se desarrolla el software de gestión, además se incluye la creación de un servidor con una base de datos en SQL. Se obtiene el análisis de costos de su implementación y se simula el comportamiento del edificio para obtener las mejores opciones de implementación con el único propósito de que la estructura propuesta pueda convertirse en una edificación inteligente, disponiendo de un sistema de gestión. Un sistema convencional como actualmente dispone el edificio “Padre Juan Bottasso”, desperdicia una gran cantidad de energía eléctrica por mantener iluminación encendida en espacios innecesarios, esto agregando que si existiera un sistema de climatización estos valores aumentarían aún más. Al implementar un sistema de gestión de edificios, se trabaja y controla el comportamiento del edificio de acuerdo con las necesidades de ocupación obteniendo así un ahorro energético considerable según el análisis de las simulaciones, logrando en la práctica un posible aumento de confort de los ocupantes del edificio.

# Abstract

This work describes the research applied in the field to develop a building management system that allows the control of lighting and air conditioning, with the objective of improving energy performance and air quality in the building "Padre Juan Bottasso" of the Salesian Polytechnic University Cuenca. For this purpose, the elements that affect lighting and air conditioning are identified, determining their energy consumption and air currents, seeking to maintain acceptable levels of comfort and safety in critical environments. The technical planimetry for its implementation is carried out and the management software is developed, including the creation of a server with a SQL database. The cost analysis of its implementation is obtained and the behavior of the building is simulated to obtain the best implementation options with the sole purpose that the proposed structure can become an intelligent building, having a management system. A conventional system, as the "Padre Juan Bottasso" building currently has, wastes a large amount of electrical energy by keeping lighting on in unnecessary spaces, adding that if there were an air conditioning system, these values would increase even more. By implementing a building automation system, the system works and controls the behavior of the building according to the occupancy needs, thus obtaining considerable energy savings according to the analysis of the simulations, achieving in practice a possible increase in comfort of the building occupants.

# Capítulo 1

## Introducción

Tradicionalmente, los edificios se construyen con sistemas independientes, esto implica el uso de varios softwares para tener el monitoreo y control de cada sistema que se encuentra en un edificio. Los sistemas que normalmente se encuentran en una edificación son: ventilación, iluminación, climatización, control de potencia, sistemas de detección y contra incendios, seguridad, medición inteligente, y otros más. Al operar de forma independiente dificulta el despliegue de un sistema global que dé respuestas inmediatas a cualquier tipo de eventos, así como reducir el consumo de energía, controlar el sistema de iluminación, monitorear la calidad del aire, control de temperatura, y otros.

La continua innovación y los avances tecnológicos conducen a nuevas formas de vida de la población a nivel mundial. Con el surgimiento de la industria 4.0 la automatización viene a ocupar un espacio importante en la gestión eficaz de una edificación. La economía y los mercados globales evolucionan de manera vertiginosa, por lo que es imperante que los países estén preparados para innovar sus procesos, sino se verán rezagados respecto a los países desarrollados. Por ello y ante el interés que representan estas soluciones, en el sector de la construcción aparecen tecnologías que impactan directamente en el funcionamiento de una edificación y que permiten gestionar, controlar y administrar de mejor forma una edificación. A esto se suma la capacidad de poder integrar sistemas con las nuevas infraestructuras.

Los costos de energía seguirán aumentando, además, sabemos el impacto ambiental que el planeta padece en la actualidad, agregando una pandemia

que ha cambiado el diario vivir de las personas. Por ello existe una alta demanda en dar soluciones que permitan obtener eficiencia energética, conservación de energía, calidad de aire. Las soluciones de edificaciones inteligentes no solo permiten tener eficiencia energética y ser sostenibles, sino que también se les puede automatizar para poder controlarlas y gestionarlas. [Wood Laura, 2020]

Es así como, para suplir estas necesidades las empresas desarrollan softwares de gestión de edificios que por definición es una red electrónica que se utiliza para supervisar y controlar los servicios eléctricos, electrónicos y mecánicos de un edificio [Australian Government, 2021]. En la actualidad el fácil y rápido acceso a equipamiento tecnológico, la conciencia climática que han incorporado tanto las empresas constructoras como los organismos de control y el poder adquisitivo de los consumidores principalmente en los países con economías emergentes alrededor del mundo presuponen que las empresas que desarrollan sistemas de gestión tendrán un importante crecimiento en los próximos años.

Este aumento de demanda se ve reflejado de acuerdo con la tasa compuesta de crecimiento anual CAGR de más del 12 % para el 2021. Además se espera un registro CAGR de más del 23 % en los próximos 5 años y se proyecta que la próxima gama de soluciones para edificios inteligentes crecerá a una tasa compuesta de crecimiento anual CAGR del 32 % en los próximos ocho años por servicios y software. [Valzania Gilberto, 2018]

De esta forma para cumplir con la demanda actual que una edificación requiere para su automatización, se lo puede representar en tres pasos, el primero es la interacción con el entorno y se lo define como nivel de campo, es el equipamiento físico que permite la extracción de los datos. Como segundo paso se define la automatización de los componentes que están ligados al nivel de campo. El tercer y último paso es el uso de sistemas informáticos para integrar en una sola plataforma los sistemas que se encuentran en un edificio. Esto se conoce como sistemas de gestión y por sus siglas en inglés Building Management System (BMS) ó también conocido como Building Automation System (BAS).

Los sistemas de gestión de edificios en la actualidad son importantes para cualquier infraestructura que se instala con el objetivo de tener edificios seguros, sostenibles y confiables sin perder niveles de confort. El sistema permite la gestión centralizada de los equipos de infraestructura a través

de aplicaciones informáticas integradas y garantiza que las operaciones se ejecuten de manera eficiente y segura. Los sistemas abiertos en la actualidad tienen la capacidad de soportar múltiples protocolos de comunicación que permiten obtener interoperabilidad en sistemas de control, resultando en el monitoreo y control de sistemas eléctricos, electrónicos, mecánicos y electromecánicos desde una sola plataforma de gestión, convirtiéndose en la opción más viable para las empresas que desean crear edificios inteligentes. Las empresas que ofrecen soluciones de sistemas de gestión aprovechan la interoperabilidad y su capacidad para comunicarse entre ellos permitiendo ofrecer sistemas conectados a la red con características mejoradas.

Este proyecto plantea el desarrollo de un sistema de gestión de edificios para el control de iluminación y climatización, que responda a las necesidades de eficiencia energética y flujo de corrientes de aire dentro del edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana Sede-Cuenca. Además, se identifican los elementos y/o dispositivos necesarios para la implementación de Sistemas de Control de Energía y Climatización, así como determinar las zonas de mayor incidencia de consumo de energía y corrientes de aire dentro del edificio.

## 1.1. Descripción general del problema

El repunte de las emisiones globales de carbono hacia fines del año 2019 es una severa advertencia de que no se está haciendo lo suficiente para acelerar las transiciones de energía en todo el mundo. Las tendencias de 2020 subrayan el desafío de frenar las emisiones al tiempo que garantizan el crecimiento económico y la seguridad energética. [Shaikh et al., 2014]

De acuerdo con las emisiones de  $CO_2$  relacionadas con la energía en edificaciones, estas han sufrido un aumento en los últimos años. Para el año 2019 los edificios aumentaron sus emisiones ya sea directa o indirectamente de electricidad y el calor comercial a  $10GtCO_2$ . Como es de conocimiento general la electricidad se genera en parte por la combustión de recursos no renovables, como lo son los combustibles fósiles. Este uso de combustibles fósiles para la generación de electricidad implica la emisión de contaminantes de aire y de gases de efecto invernadero (GEI). De esto se relaciona que el paso para el control de emisión de  $CO_2$  y el consumo eléctrico a nivel nacional se realiza cuantificando el factor de emisión eléctrico, que corresponde a la cantidad de  $CO_2$  que se emite en la atmósfera, por cada unidad de electricidad generada. [Posso Alex; Melo Paul; Flores Santiago; Haro Lenin, 2019]. De esta forma el consumo está superando la demanda y se presentan diferentes factores que han logrado registrar los niveles más altos jamás registrados. Estos factores incurren en el aumento de demanda de energía para calefacción, refrigeración, cambios climáticos extremos, falta de políticas de eficiencia energética, inversión insuficiente en edificios sostenibles.

Considerando que en Ecuador las regulaciones existentes son aún permisivas en la prevención de emisión dióxido carbono en las construcciones es por lo tanto evidente que en la Universidad Politécnica Salesiana (UPS) tampoco cuenta con una planificación para el control de la emisión de dióxido carbono en sus construcciones. Tomando los consumos energéticos del año 2018 como referencia por los datos obtenidos en anteriores trabajos de investigación se produce un consumo anual de 1'039.364 KWh [David et al., 2019].

Además, de acuerdo con el informe del año 2019 denominado Factor de emisión de  $CO_2$  del sistema nacional interconectado de Ecuador se concluye que: el factor de emisión de  $CO_2$  es de  $0,4509 tCO_2/MWh$ . Dado estos valores se requiere transformar las unidades de consumo energético de la

UPS a MWh. El resultado queda expresado de la siguiente manera: 1.039,36 MWh. La conversión a toneladas  $CO_2$  para definir la emisión de dióxido de carbono es la multiplicación del consumo en MWh y el factor de emisión de acuerdo con el informe del año 2019, quedando una emisión de carbono de: 468,64 toneladas de dióxido de carbono. [Davila Fernando; Varela Daniel, 2014]

Los sistemas que influyen en el aumento de emisiones en edificaciones son los componentes eléctricos y mecánicos debido al incremento de refrigeración de espacios, electrodomésticos y accesorios eléctricos. Si bien la electricidad representó un tercio del uso de energía de los edificios en el 2019, el uso de combustibles fósiles también aumentó a una tasa de crecimiento promedio anual marginal del 0.7 % desde el 2010. Es así como las emisiones de edificios aumentaron un poco más de  $3GtCO_2$  en 2019 que representa un incremento del 5 % desde 2010. Cuando se consideran las emisiones indirectas de la generación de energía, los edificios fueron responsables del 28 % de las emisiones globales de  $CO_2$  relacionadas con la energía en 2019. En términos absolutos, las emisiones de  $CO_2$  relacionadas con los edificios aumentaron, alcanzando un máximo histórico de  $10GtCO_2$  en 2019.

La demanda de servicios de energía a partir del año 2016 en especial en el área de la electricidad para refrigeración, electrodomésticos y dispositivos conectados superó el consumo respecto a la eficiencia energética resultando en un aumento de emisiones relacionadas con la energía en edificios. La intensidad energética (uso de energía final por  $m^2$ ) se ha visto disminuido desde 2010 entre un valor de 0.5 % y 1 %, pero esta tasa es mucho menor que el crecimiento anual de superficie construida que se ha mantenido en el 2.5 % desde 2010.

Con esto vemos que existe una enorme brecha de eficiencia energética aún por cubrir, ya que la evolución energética necesaria en los edificios es menor a la expansión de superficie de construcción especialmente para economías emergentes. Para obtener un escenario de desarrollo sostenible, la intensidad energética debe llegar a valores de al menos 2.5 %, y esto podría alcanzarse con normativas de energía más estrictas, optimización de recursos y rendimientos de electricidad y otras medidas de eficiencia energética. [Shaikh et al., 2014], [Abergel Thibaut and Delmastro Chiara, 2020]

Además, la edificación Juan Bottasso no contempla en su etapa

constructiva un sistema de ventilación para circulación de corrientes de aire, esto justificado porque en su etapa de planificación no se esperaba que llegue una pandemia de tal magnitud como la que estamos viviendo. Sin embargo, es recomendable según la Organización Mundial de la Salud mantener los espacios con ventilación adecuada porque permite introducir a la edificación aire limpio del exterior eliminando el aire viciado del interior. Con ello se mantiene una mejor calidad de aire en el edificio. La ventilación se puede lograr con diferentes sistemas, que regularmente son sistemas mecánicos, y así evitar la propagación de la COVID-19. [[Organización Mundial de la Salud, 2021](#)]

Para fines de nuestro proyecto, este estudio se centra en la aplicación de tecnologías para obtener eficiencia energética, y de acuerdo con esta alternativa tenemos edificios comerciales de diferentes tipos: oficinas, hospitales, escuelas, hoteles, centros comerciales, estaciones de la policía, etcétera. Los diferentes tipos de construcción tienen necesidades energéticas únicas y la energía utilizada en edificios residenciales y comerciales, representó el 20 % del consumo mundial de energía entregada en 2018. En referencia a la Perspectiva Energética Internacional 2019 (IEO2019), la Administración de Información Energética (EIA) de Estados Unidos proyecta que la energía global respecto al consumo en edificaciones crecerá en un 1.3 % por año en promedio de 2018 a 2050. En los países que no forman parte de la Organización para la Cooperación y el Desarrollo Económico (países no pertenecientes a la OCDE), EIA proyecta que la energía consumida en edificios crecerá en más de 2 % por año, o alrededor de cinco veces la tasa de los países de la OCDE. De igual forma se espera que a inicios del año 2020, el uso total de electricidad en edificios de países no pertenecientes a la OCDE supere el uso de electricidad en los países de la OCDE. Para el año 2050 se estima que los edificios en países no pertenecientes a la OCDE utilizarán colectivamente aproximadamente el doble de electricidad que los edificios en los países de la OCDE. [[Hojjati Behjat, 2019](#)]

La electricidad es la principal fuente de energía para la iluminación, climatización, equipos y electrodomésticos, componentes esenciales para el consumo energético en una edificación, y es esta fuente de energía la que presenta el mayor crecimiento en edificios residenciales y comerciales. El consumo mundial de electricidad continúa aumentando más rápido que la población mundial, lo representa en un aumento en la cantidad promedio de electricidad consumida por persona (consumo de electricidad per cápita),

según las estadísticas internacionales de energía de la Administración de Información de Energía de EE. UU. (EIA). Casi todo el aumento es atribuible al creciente consumo de electricidad en los países en desarrollo fuera de la Organización para la Cooperación y el Desarrollo Económico (OECD). [Kahan Ari, 2020]

El aumento de energía se debe normalmente a la demanda de servicios que actualmente existen y el estilo de vida considerando el confort. Además, considerando que actualmente el ecosistema se ve afectado por las actividades humanas se ve la importancia de construir edificios inteligentes. Según el Departamento de Energía de los Estados Unidos. (DOE), y enfocándose en el proyecto de desarrollo planteado, la iluminación y climatización consumen aproximadamente el 50 % del uso de energía en un edificio comercial promedio. Al implementar sistemas inteligentes de gestión de edificios, la iluminación y climatización pueden disminuir los costos de energía entre 30 % y 50 % [Kahan Ari, 2020].

Considerando que, en la ciudad de Cuenca, no existen edificaciones que dentro de su planificación constructiva involucren la opción de un edificio inteligente, debido a que:

- Al ser un software que controla y monitorea los diferentes sistemas que involucran el comportamiento del edificio se tiene que trabajar juntamente con otras ingenierías desde que empieza la construcción y esto produce un escepticismo por cuanto sus sistemas serán controlados por un software independiente que no utilizan en sus ingenierías y por el temor a que pueda verse afectado el comportamiento de sus equipos.
- A esto se suma que no existe un verdadero interés en una inversión de este tipo por cuanto no es una real necesidad para la operación de un edificio ya que normalmente son edificaciones que se construyen para la venta y el gasto final lo realiza el cliente. En el caso de la edificación que se realiza en esta tesis el grupo inversor es la propia universidad en este caso en particular se entiende que la falta de proveedores en ofrecer sistemas de gestión y el poco conocimiento por no decir nulo de las empresas constructoras produce que no se pueda tomar en cuenta estos sistemas en la planificación constructiva del edificio.
- Los costos de inversión para la implementación de un sistema de gestión son altos puestos que se requieren de equipamiento de hardware y software para su funcionamiento. Esto produce que las personas

responsables de la ejecución de la obra no tomen en cuenta agregar más soluciones a sus edificaciones debido a los costos que estos representan.

- El medio en el que convivimos no conoce de los costos/beneficios que implica la utilización de un sistema de gestión para una correcta administración de la edificación.
- En el país no rigen normativas que obliguen a tener edificios inteligentes para mejorar la eficiencia ambiental, y esto además de que no existe una verdadera conciencia al momento de construir para reducir la emisión de gases y la huella de carbono, esto repercute en que las edificaciones sean construidas de forma convencional.
- El mantenimiento de un sistema de gestión y de todos los sistemas que lo conforman es importante para su correcto funcionamiento es por ello por lo que este costo de mantenimiento incurre en un valor importante en la administración del edificio para mantener todos los dispositivos, sistemas y programaciones operativas. Una falla en el sistema implica en un fallo o alarma en el software por lo que si no se aplica los correctivos necesarios el sistema de gestión no cumple con su trabajo para el que fue creado.
- Al implementar un sistema de gestión se produce una optimización de recursos en todo el componente de gestión. Esto genera malestar por la implementación de estos sistemas debido a que la mano de obra de persona se reduce porque el software cumple con obligaciones que normalmente la ejecuta un ser humano.
- Otra de las situaciones por las que no se optan la utilización de estos sistemas es por su producción de fallos. Al ser componentes eléctricos y electrónicos están susceptibles a fallos, y que pueden producir daños importantes como fallas eléctricas o errores en administración de energía, sensores de energía, almacenamiento de información, confidencialidad de los datos que se manejan dentro del edificio. Al ser tecnologías nuevas pueden aparecer errores no intencionales que produzcan diferentes tipos de errores y esta razón es por la que el medio no acepta el inteligenciar los edificios en su totalidad.

Es importante mencionar, que la tesis plantea la solución de un sistema de gestión de edificios mediante el control de iluminación y climatización. Sin embargo, en esta región del país no se toma en cuenta comúnmente como un proceso constructivo en edificios la instalación de sistemas de

ventilación o climatización para permitir el flujo constante de corrientes de aire, y tomando en cuenta lo que está causando la pandemia actual este componente toma un papel fundamental para disminuir el contagio masivo de la COVID-19.

Es así que se propone la realización de este trabajo con el fin de poder mostrar la importancia de un sistema de gestión en una edificación además de evaluar el comportamiento del edificio con un software de gestión y de los subsistemas que se puedan implementar dentro de un edificio ya construido. Para ello se propone desarrollar un sistema de gestión de edificios para el control de iluminación y climatización, que responda a las necesidades de eficiencia energética y flujo de corrientes de aire dentro del edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana Sede-Cuenca.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Desarrollar un sistema de gestión de edificios para el control de iluminación y climatización, que responda a las necesidades de eficiencia energética y flujo de corrientes de aire dentro del edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana Sede-Cuenca.

### **1.2.2. Objetivos específicos**

- Desarrollar una propuesta para crear un sistema de gestión de edificios para el control de iluminación, medición de energía y climatización, que mejore el desempeño energético y calidad del aire en el edificio “Padre Juan Botasso” de la Universidad Politécnica Salesiana Sede-Cuenca.
- Identificar los elementos y/o dispositivos necesarios para la Implementación de Sistemas de Control de Energía y Climatización, así como determinar las zonas de mayor incidencia de consumo de energía y corrientes de aire dentro del edificio.
- Realizar un levantamiento de los equipos disponibles en el mercado y analizar los protocolos de comunicación para la implementación en edificios BMS.

### **1.3. Contribuciones**

El presente trabajo propone la investigación aplicada en campo para desarrollar un sistema de gestión de edificios para el control de iluminación y climatización, que responda a las necesidades de eficiencia energética y flujo de corrientes de aire dentro del edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana Sede-Cuenca, y que permita generar niveles aceptables de confortabilidad y seguridad en entornos críticos.

### **1.4. Organización del manuscrito**

El proyecto de desarrollo que se encuentra en esta tesis contiene los siguientes lineamientos en cada capítulo:

- Capítulo 1: Introducción y determinación del problema.
- Capítulo 2: Sistema de gestión de edificios que identifica los componentes necesarios para considerarse un sistema de automatización de edificios.
- Capítulo 3: Desarrollo de un sistema de gestión (BAS) que define los criterios para cumplir con los componentes que integran un BAS.
- Referencias bibliográficas.
- Anexos.

## Capítulo 2

# 2. Sistema de gestión de edificios

Desde hace décadas las empresas se han dedicado a la investigación y desarrollo de sistemas que permitan obtener servicios confiables y energéticamente eficientes sin sacrificar niveles de confort. Con ello los desarrolladores de estos productos ofrecen sistemas inteligentes y siguen innovando para mejorar y convertir sus productos en sistemas que se adapten a nuevas tecnologías. Los esfuerzos se dedican en obtener sistemas inteligentes de múltiples empresas y que sean integrables entre ellos para el monitoreo y control desde un software de gestión. [[Jia et al., 2019](#)] y [[Liu et al., 2019](#)]

A lo largo de los años se han dado algunas definiciones a los sistemas de gestión de edificios, sin embargo, nos enfocaremos en las referencias bibliográficas escritas en la ISO 16484-3 y en la EN 15232 [[Martirano and Mitolo, 2020](#)], que basan sus conceptos en el alcance de la funcionalidad esperada de los sistemas de gestión de edificios, que lo nombraremos de ahora en adelante como BAS. [[Domingues et al., 2016](#)]

## 2.1. Estado del Arte

Un BAS es un sistema de gestión orientado al control y monitoreo de servicios de construcción, y se lo denomina sistema de control y automatización de edificios [Domingues et al., 2016] y [Ehrlich, 2014].

### 2.1.1. Arquitectura

En el nivel de arquitectura se define de acuerdo con la Figura 2.1, en el cual un BAS se constituye de tres niveles: [Fernbach et al., 2011]

- Nivel de campo donde se interactúa con los dispositivos (sensores, actuadores) y se realizan los procesos técnicos. Los datos ambientales se recopilan (medición, conteo) y se transforman para ser transmitidos y procesados. Los parámetros del ambiente actúan en respuesta a los comandos recibidos del sistema.
- Nivel de automatización donde se procesan las mediciones y se ejecutan las funciones es la etapa que proporciona la funcionalidad de control. En este nivel se asigna todo tipo de secuencias y funciones preparadas para el nivel de campo, estableciendo conexiones lógicas y bucles de control.
- Nivel de gestión donde se llevan a cabo las actividades y se presentan los datos del sistema mediante una interfaz que entrega funcionalidades para aplicaciones empresariales y de gestión. Una interfaz única presenta al operador toda la información del edificio.

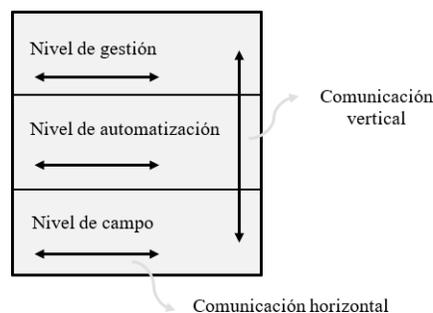


Figura 2.1: Arquitectura de un BAS.

Fuente: autor

### 2.1.2. Redes de comunicaciones

Las redes de comunicación son las responsables de garantizar el traspaso de información entre los niveles de la arquitectura de un sistema BAS, es así como, un nivel de automatización hay un paso de información al nivel de gestión y normalmente esta comunicación es por una red LAN o basada en IP. La comunicación de las redes puede ser tanto alámbrica como inalámbrica y con ello permite la comunicación entre dispositivos de control, computadoras, servidores y aplicaciones. Los buses de campo permiten la interacción entre dispositivos de campo, como controladores, sensores y actuadores. La instalación general de redes de comunicación consta de buses de campo y de una sola red troncal basada en IP.

Los dispositivos de red y los de buses de campo están conectados entre sí bajo topologías predefinidas y específicas. Por ello se manejan protocolos de control de acceso a los medios cuyo objetivo es garantizar la comunicación entre dispositivos evitando que las señales transmitidas se superpongan cuando más de un dispositivo transmite al mismo tiempo.

En la actualidad las empresas han desarrollado diferentes tecnologías de redes de control que varían según su operación. De esta manera tenemos diferentes estándares como: BACnet [Park and Hong, 2009], KNX [Lee and Hong, 2008], LonWorks [Shahnasser and Wang] y [Snoonian, 2003], Modbus [You and Ge, 2019], Opc [Fernbach et al., 2011], y en sistemas inalámbricos tenemos: Zigbee [Lee and Hong, 2008], LoRaWAN [Monfort, 2017], y otras.

### 2.1.3. Sensores y actuadores

Un BAS contiene sensores, actuadores y más componentes de hardware. Los sensores permiten las mediciones y los actuadores responden a una señal. Los sensores y actuadores se conectan a las entradas/salidas de los módulos de hardware quienes tienen la lógica de control. Para el funcionamiento en esta etapa de campo los edificios utilizan el software de gestión que controla los actuadores físicos para activar luces, motores, aires acondicionados, persianas y otros. Dependiendo de la retroalimentación de las entradas monitoreadas por los sensores estos actuadores responden a estas entradas ya comandos del sistema programado.

#### 2.1.4. Modelo de dispositivos

Cada dispositivo o hardware contiene su configuración que es previamente parametrizada de acuerdo con los requerimientos del sistema de gestión. La parametrización de cada dispositivo refleja las características de flujo de los datos enviados a través de la red, por ejemplo, la resolución de cámaras de seguridad, la sensibilidad de los sensores de ocupación o de detectores de humo, los intervalos de medición de temperatura de los sensores, etcétera. Normalmente los dispositivos de hardware no contienen la inteligencia por ello se rigen a los comandos dados por el sistema de gestión [[Vandenbergh et al., 2006](#)].

Enfocado en un sistema BAS los dispositivos de control tienen dos interfaces, la interfaz eléctrica/electrónica la cual define la conexión de los dispositivos con el sistema, y la interfaz de aplicación que permite la interacción de otros dispositivos y aplicaciones de software con los dispositivos de control. De acuerdo con estos requerimientos se usan los controladores y cada controlador responde a un tipo particular de dispositivo, de esta forma un sistema BAS debe tener la capacidad de soportar cientos de dispositivos que provienen de diferentes marcas, protocolos y estándares. La inmensa variedad de dispositivos de diferentes marcas confirma que obtener un controlador general compatible para todos los dispositivos es aún muy difícil de alcanzar, sin embargo, si diferentes dispositivos comparten similitud de protocolos, estándares y el sistema de gestión soporta los mismos, es muy probable que se los pueda monitorear y controlar con el sistema de gestión.

Para poner en contexto, por ejemplo, un sistema de cámaras de seguridad que manejen una interfaz estandarizada ONVIF cuyo objetivo es la interoperabilidad de productos de seguridad física basados en IP, permitiría colocar cámaras de seguridad de diferentes fabricantes y monitorearlas desde un solo sistema de gestión que soporte el estándar ONVIF [[Senst et al., 2011](#)]. En un sistema inalámbrico con protocolo de comunicación Zigbee que es un protocolo de baja potencia dirigido a aplicaciones de automatización como por ejemplo sensores de ocupación y que permiten tener conectividad de equipos para encendido/apagado de luminarias por ejemplo, de la misma forma que el estándar ONVIF si un controlador soporta este protocolo y el sistema de gestión lo reconoce a este controlador entonces podemos integrar diferentes fabricantes de dispositivos Zigbee en una sola plataforma de gestión [[Lee and Hong, 2008](#)].

Es importante definir la plataforma de gestión para considerar las marcas y equipos que se integran a esta plataforma para la posterior integración y automatización del edificio.

### **2.1.5. Comandos, retroalimentación y etiquetas**

El sistema de gestión se refiere al uso de comandos, retroalimentación (feedback) y etiquetas (tags). Los comandos permiten la ejecución de operaciones en los dispositivos. Cuando se activa un comando los dispositivos cambian su estado a la programación de ese comando.

Por ejemplo, encender/apagar una luminaria es el resultado de ejecutar un comando. La retroalimentación permite conocer el estado del comando activado de tal forma que el software de gestión tiene pleno conocimiento de lo que cada botón, escena, rutina se está ejecutando, visualmente para el operador del sistema de gestión la retroalimentación le permite estar enterado del funcionamiento global del sistema de automatización. Las etiquetas son usadas para el servidor del sistema de gestión, y son variables que permiten recibir retroalimentación del estado de los comandos en el servidor.

### **2.1.6. Funcionalidad del sistema de gestión**

En esta sección se define las funcionales del sistema de gestión. Para ello nos basamos en la siguiente Tabla 2.1 la cual considera la funcionalidad básica que debe tener cualquier sistema de gestión moderno. Para ello se referencia a los dos estándares internacionales y que define las siguientes características: agrupación y zonas, eventos y alarmas, horarios y rutinas, escenarios y histórico de base de datos.

Tabla 2.1: Funcionalidades de un sistema BAS  
 [Domingues et al., 2016]

<b>Funcionalidad</b>	<b>ISO – 16484 – 3</b>	<b>EN 15232</b>
Agrupación/zonas	El control de zona individual es una función requerida.	El concepto de zonas define el control de actividades específicas para la habitación o zona y las preferencias de ajuste.
Eventos/alarmas	El manejo y notificación de eventos y alarmas es una función del operador y proporcionada por el HMI.	Se refiere a las notificaciones de los cambios de estado del sistema para varios propósitos. Las notificaciones de alarmas son importantes para detectar el mal funcionamiento o situación ocurrida.
Horarios/rutinas	Proporciona una sección que describe las funciones de programación para horarios.	Especifica los horarios y rutinas para el control de actividades como el flujo de aire en una habitación.
Escenarios	No hay una terminología directa pero el servicio de gestión de energía como requisito, que según EN 15232 se beneficia de este concepto.	Varios modos de funcionamiento del edificio que se adaptan a las necesidades de los ocupantes.
Base de datos	Este estándar presenta la base de datos y los medios para recuperar esos datos como funcionalidades de gestión de un BAS.	Recopilación y registro de datos que ofrecen los sistemas de gestión de edificios.

### **Agrupación y zonas**

La agrupación de dispositivos y zonas permite simplificar la programación al mismo tiempo que organiza y estructura de mejor forma el comportamiento del edificio considerando las necesidades de los ocupantes en cuanto a su confort y eficiencia energética. Por ejemplo, las luminarias y sensores de ocupación de los pasillos se agrupan en una sola zona simplificando el comportamiento de los dispositivos en ese lugar. También se puede agrupar por sistemas es decir el sistema de aire acondicionado se pueda activar con el encendido de luminarias cuando se requiera ingresar a una habitación para cualquier actividad. De igual formase realizan agrupaciones para alarmar un edificio por la activación de un detector de humo en este caso la agrupación se realiza de la siguiente forma, dependiendo del piso donde se active la alarma la agrupación se realizaentre el piso superior e inferior donde se produjo la activación de la alarma. Las agrupaciones se pueden realizar de diferentes formas y cada aplicaciones diferente.

La automatización de edificios está ligada a la idea de controlar y monitorear espacios, y de estos espacios los dividimos en subespacios que los llamamos zonas [Domingues et al., 2016]. Las zonas se pueden interpretar de acuerdo con el requerimiento de lo que se desea realizar, pueden ser por pisos, por habitaciones, por espacios dentro de habitaciones, etcétera. Mediante esta agrupación de zonas lo que se intenta es facilitar al operador del sistema de gestión su interacción con el sistema y también la rápida atención a cualquier evento que se presente.

### **Eventos y alarmas**

Los eventos se producen cuando se modifica el estado de un sistema estos pueden ser el encendido de un aire acondicionado, el apagado de una lámpara, la activación de una alarma, etcétera. Normalmente en la programación de un sistema de gestión se delimita las notificaciones de eventos y estos están caracterizados por situaciones que implican el cambio de comportamiento del edificio, por ejemplo, notificar cuando el consumo eléctrico está por superar la media mensual de consumo, o el cambio de temperatura de una habitación, la apertura de puertas que normalmente son para acceder a áreas críticas como un cuarto de telecomunicaciones. Los eventos nos permiten estar atentos a cualquier cambio que perjudique el normal comportamiento del edificio.

Por otro lado, las alarmas son situaciones excepcionales que se originan en un dispositivo en específico y que alteran el correcto funcionamiento del edificio y del sistema de gestión. Esta información de alarmas es importante para tener una rápida respuesta a cualquier suceso y controlar el evento que causó esta activación. Además, también puede haber alarmas por malfuncionamiento de uno o varios dispositivos que de igual forma nos permitiría reaccionar de forma inmediata para mantener el sistema en buen funcionamiento.

### **Horarios y rutinas**

Para un sistema de gestión es importante manejar horarios y rutinas que se ejecutan diaria o esporádicamente esto simplifica la operación del edificio por usuarios. Para ello se realizan programaciones para que se ejecuten tareas a horarios predeterminados y que se vuelvan rutinas para actividades diarias. Los horarios se programan de acuerdo con las fechas, días, horas, minutos y segundos. A medida que las rutinas avanzan los comandos programados se activan. Por ejemplo, puede haber horarios para los días laborables, horarios para feriados, horarios para fines de semana, etcétera.

### **Escenas**

Las escenas en el sistema de gestión BAS, se describe como el estado deseado en un dispositivo o un grupo de dispositivos [Domingues et al., 2016]. Las escenas pueden ser estáticas o dinámicas, las escenas estáticas no cambian de estado una vez programada esa escena, estas pueden ser por ejemplo el encendido de luces de la fachada de un edificio a un horario específico. A diferencia de las escenas dinámicas que, si cambian de acuerdo con parámetros de programación, es decir, basándonos en el mismo ejemplo del encendido de luces de la fachada esta escena puede estar programada para que se encienda cuando haya alcanzado cierto nivel de luxes el sensor. De esta forma esta escena encenderá las luminarias a diferentes horas cada día.

Las escenas se pueden realizar para todo tipo de dispositivo. Puede haber escenas con nombre “presentación” la cual al momento de activarse puede encender el proyector, atenuar las luminarias a un 15 %, bajar las persianas y encender una cinta LED de luz guía para caminar sobre un pasillo.

Se puede ejecutar escenas para mantener niveles de confort. En una habitación configurar para encender un aire acondicionado cuando la temperatura esté por encima de 27 grados Celsius y apagar cuando se tenga una temperatura por debajo de los 21 grados Celsius.

### **Base de datos**

Los registros para base de datos son importantes para comprender y conocer el comportamiento diario del sistema de gestión BAS, principalmente cuando el operador del sistema tiene poca interacción con el BAS. El registro de datos se encarga de registrar los comandos, transiciones de los dispositivos, eventos, alarmas, de tal forma que se comprenda de mejor forma la actividad del BAS con fines de identificar su comportamiento.

## **2.2. Tecnologías para automatización de edificios**

Una edificación inteligente se puede denominar a la estructura que facilita los procesos mediante la automatización para todo tipo de operaciones del edificio, como lo pueden ser: iluminación, climatización, seguridad, ventilación, calefacción, etc. Para este tipo de edificios se utilizan sensores, actuadores, tecnologías de la información y la comunicación para administrar y controlar sistemas. El enfoque para obtener un sistema de gestión debe ser la capacidad de integrarse de manera efectiva con los diferentes fabricantes de dispositivos. Un edificio puede manejar múltiples marcas y muchos dispositivos que entre todos ellos permiten la operación de un edificio. Para ello tenemos tecnologías, estándares y protocolos de automatización que actualmente se encuentran en el mercado y que es importante identificarlos [McGibney et al., 2016]. Entre algunos de ellos tenemos:

### **2.2.1. BACnet**

BACnet es un protocolo de comunicación estándar internacional ISO aprobado para sistemas de control y automatización de edificio [23]. Desarrollado por un comité de proyecto establecido por la Sociedad Americana de Ingenieros de Calefacción, Refrigeración y Aire acondicionado (ASHRAE). BACnet define un modelo de información que organiza los dispositivos del sistema utilizando una colección estándar de objetos [Fernbach et al., 2011]. En BACnet un dispositivo consta de un objeto de dispositivo que define las propiedades del dispositivo, modelo, y lista de objetos asociados al dispositivo. El modelo BACnet también admite tipos

de objetos patentados que fabricantes registren funcionalidades que están dentro del estándar [Domingues et al., 2016].

### 2.2.2. KNX

KNX es una tecnología europea creada por BUS (EIB), European home System (EHS) y Batibus que dieron lugar al establecimiento de la Asociación Konnex en 1999. En el año 2004 se estandarizó como norma EN50090 y en el año 2006 fue reconocido como el estándar internacional ISO/IEC14543-3 [Domingues et al., 2016]. La funcionalidad de aplicaciones con KNX se extiende en bloques funcionales, cada bloque funcional es implementado por un solo dispositivo y cada dispositivo puede albergar múltiples bloques funcionales. Un bloque funcional tiene un comportamiento bien definido y consta de uno o más puntos de datos. Un solo punto de datos representa un solo dato de la aplicación. Dependiendo del tipo, puede ser una entrada o una salida, así como un parámetro que influye en el comportamiento del bloque funcional. Para la interoperabilidad de dispositivo cada punto de datos tiene un punto de datos definido el cual establece el formato, la codificación, rango y unidad del punto de datos [Granzer and Kastner, 2012].

### 2.2.3. LonWorks

LonWorks o Local Operating Network (LON) es un protocolo de red de control abierta y aplicación desarrollada por Echelon Corporation, también ha sido estandarizado como ISO/IEC-14908 y como estándar europeo EN14908. La pila de protocolos de red es una red enrutada simplificada con hasta 255 subredes, cada una compuesta por un máximo de 127 nodos. Cada nodo puede ser miembro de uno de los 256 grupos de multidifusión disponibles para obtener una mayor eficiencia en la comunicación: los paquetes de multidifusión se enrutan en un dominio de colisión si hay al menos un miembro que pertenece al grupo correcto [Antonini et al., 2014].

### 2.2.4. Dali

Dali cuyo significado es Digital Addressable Lighting Interfaces es un protocolo dedicado para el control de iluminación digital que proporciona funcionalidad a nivel de campo. Dali ha sido desarrollado para permitir un control digital de balastos fluorescentes reemplazando el control analógico. Mediante Dali cada dispositivo se le puede asignar una dirección separada lo que permite el control digital de los dispositivos individuales. Además, también se pueden programar para operar en grupos proporcionando una

excelente flexibilidad ya que no es necesario cambiar el cableado para una reconfiguración [Granzer and Kastner, 2012].

### 2.2.5. Modbus

Modbus es un protocolo de capa de aplicación que proporciona un esquema de comunicación entre cliente-servidor. Las comunicaciones de Modbus requieren un acuse de recibo/acuse de recibo negativo a cada mensaje y generalmente se realizan sobre Ethernet, con el uso de TCP/IP para proporcionar garantías de transporte confiables, aunque al emplear EIA RS-232 y EIA RS-485 también es posible. Como la mayoría de las implementaciones de Modbus comunes se ejecutan a través de Ethernet, admite comunicaciones de unidifusión y transmisión en un medio compartido con una estrategia de CSMA/CD [Antonini et al., 2014] y [You and Ge, 2019].

### 2.2.6. Zigbee

ZigBee es una tecnología de comunicación inalámbrica, que se aplica en corto alcance, baja tasa de transmisión entre los distintos equipos electrónicos. ZigBee como estándar emergente de tecnología de transmisión inalámbrica tienen sus características técnicas únicas: bajo consumo de potencia, fiabilidad, expansión.

Las aplicaciones de ZigBee se implementan por los objetos de aplicación que se distribuyen a través de los dispositivos Zigbee. Para habilitar la interoperabilidad entre los dispositivos ZigBee, se denominan perfiles de solicitud. Actualmente, los perfiles de aplicaciones están disponibles para un dominio de aplicación específico. Contienen un conjunto de especificaciones de dispositivos lógicos que describen una funcionalidad dedicada. La especificación de cada dispositivo define los llamados grupos de zigbee que deben implementarse (grupos obligatorios). Además, hay grupos opcionales también incluidos dentro de estas especificaciones. Un objeto de aplicación es, por lo tanto, una implementación de una especificación de dispositivo lógico (o al menos de todos sus grupos obligatorios) dentro de un dispositivo físico ZigBee. Un ejemplo de dicha especificación de dispositivo es el encendido/apagado la luz que representa la funcionalidad de una luminaria simple.

Un grupo de zigbee es una colección de atributos y comandos. Mientras que un solo atributo de un clúster representa un solo dato del proceso

a controlarse (es decir, el estado de una luz), los comandos se utilizan para manipular estos atributos, así como para iniciar acciones dentro del dispositivo [Granzer and Kastner, 2012] y [Li et al., 2010].

### 2.2.7. Zwave

El protocolo Z-WAVE desarrollado por Sigma Designs, Inc. Es un protocolo inalámbrico que proporciona cifrado de paquetes, protección contra la integridad y servicios de autenticación de dispositivos. Transmite las frecuencias de 868.42 MHz (Europa) y 908.42 MHz (Estados Unidos) diseñados para comunicaciones de datos de ancho de banda baja en dispositivos integrados, como sensores de seguridad, alarmas y paneles de control de automatización [Fouladi and Ghanoun, 2013].

### 2.2.8. LoRaWan

El protocolo de comunicación inalámbrico LoRaWan se utiliza cada vez más en el concepto del internet de las cosas. La capa física del protocolo de comunicación LORAWAN utiliza la modulación LORA, que está patentada por SEMTECH. La comunicación de LoRaWan representa una solución para el concepto IoT. El protocolo utiliza una topología de red de tipo estrella, en la cual cada nodo LORA puede comunicarse directamente con el módulo de pasarela. Una de las principales ventajas de LoRaWan es que utiliza la frecuencia de ISM (industrial, científica y médica) sin licencia, por lo que no se necesita ninguna licencia. La única limitación se realiza mediante el parámetro del ciclo de trabajo que se regula dependiendo del área geográfica diferente. Los dispositivos LORA están alimentados con baterías. LoRa Alliance define tres clases de dispositivos: Clase A, Clase B y Clase C [Lavric and Petrariu, 2018].

### 2.2.9. Buspro

Buspro es un sistema de control inteligente desarrollado independientemente por HDL. Es un protocolo de bus de comunicación distribuido basado en el estándar RS-485, que proporciona el control de todos los dispositivos del edificio y la integración con software y hardware de terceros. Permite que las tecnologías se fusionen y se puedan expandir de una manera simplista. Las funciones del sistema de Buspro cubren iluminación, dimerización, aire acondicionado, calefacción por suelo radiante, entretenimiento en el hogar, seguridad y protección, etc. Adecuado para edificios residenciales, de oficinas, complejos urbanos, grandes espacios,

## *2.2. TECNOLOGÍAS PARA AUTOMATIZACIÓN DE EDIFICIOS* 23

hoteles y otros edificios comerciales. Puede determinar las funciones y la escala del sistema de acuerdo con sus propias necesidades.

Cada tecnología, protocolo y estándar de comunicación tiene sus beneficios y se utilizan para sistemas específicos. Para ello hacemos Tablas comparativas [2.2](#) y [2.3](#) de las tecnologías cableadas en sistemas de gestión de edificios y una Tabla comparativa [2.4](#) para tecnologías inalámbricas [[HDL Automation, 2021e](#)].

Tabla 2.2: Comparativa de tecnologías cableadas

	<b>Bacnet</b>	<b>Modbus</b>
Uso	Comunicación a través de dispositivos.	Conexión entre dispositivos.
	Entornos industriales, transporte,	HVAC, iluminación, seguridad,
Mercado	gestión de energía, automatización de edificios, salud y seguridad.	controles de acceso, transporte y mantenimiento, gestión de energía.
Ventajas	Escalabilidad entre costo, rendimiento y tamaño del sistema. Aprobación y adopción por parte de casi todos los principales proveedores. Internetworking robusto que incluye múltiples tipos de LAN y acceso telefónico. Crecimiento sin restricciones y la capacidad de agregar nuevas innovaciones y funciones en cualquier momento.	Adecuado para volúmenes de datos pequeños y medianos (menor igual 255 bytes). Transferencia de datos diseñada para aplicaciones industriales. Publicado abiertamente y libre de regalías. Fácil de implementar y mantener.
Desventajas	Limitado el número de dispositivos de campo que se pueden conectar a una estación maestra excepto Ethernet TCP/IP Longitud del cable MT/TP Infraestructura Ethernet. El nuevo estándar de seguridad no está implementado en todos los dispositivos.	Limitar el número de tipos de datos; No se admiten objetos binarios grandes. No existe un método estándar para que un nodo encuentre la descripción de un objeto de datos, es decir, encontrar un valor de registro representa una temperatura entre 30° y 175°. Sin seguridad contra comandos no autorizados o interceptación de datos. Las transmisiones deben ser contiguas, lo que limita los tipos de dispositivos de comunicaciones remotas a aquellos que pueden almacenar datos en búfer para evitar brechas en la transmisión.

Tabla 2.3: Comparativa de tecnologías cableadas

	<b><i>LonWorks</i></b>	<b><i>KNX</i></b>
Uso	Dispositivos de red a través de líneas eléctricas, fibra óptica y otros medios.	Conexión entre dispositivos.
Mercado	Redes de control de domótica, industriales, de transporte y de servicios públicos.	HVAC, control de acceso, sensores, seguridad e iluminación.
Ventajas	Herramienta basada en web; ahorra tiempo y dinero. Numerosos desarrolladores de productos Lonworks en el mercado. Menos arquitectura a nivel de dispositivo.	Protocolo abierto, fácil de interactuar. Funciona en diferentes tipos de transmisión. Topología descentralizada; Bajo consumo de energía.
Desventajas	Los dispositivos y variables controlados están conectados a un dispositivo de control separado. (No recomendado debido a interrupciones en la red que producen fallas en el sistema). Las extensiones solo se permiten a través del Consorcio LonMark. Requiere el chip Neuron para el movimiento de red del protocolo. Cerca de la capacidad de plug/play, pero aún lejos de lograr la interconectividad con Microsoft Windows.	KNX no está ampliamente adoptado fuera de Europa. Eso significa que no existe la misma comunidad que existe para otros protocolos adoptados más ampliamente.

Tabla 2.4: Comparativa de tecnologías inalámbricas

	<b>Zigbee</b>	<b>LoraWan</b>
Uso	Consiste en coordinador, enrutadores y dispositivos finales.	Consiste en LoRa Gateway, servidores y dispositivos finales.
Ventajas	Configurar la red es muy simple y fácil. No tiene controlador central y las cargas se distribuyen uniformemente en la red. Es fácil monitorear y controlar los electrodomésticos desde el control remoto. Tomará el lugar de los dispositivos basados en tecnología infrarroja existentes. Esto ahorrará costos de reemplazo de la batería, ya que zigbee usa una batería de litio que dura mucho tiempo. La red es escalable y es fácil agregar un dispositivo final zigbee remoto a la red.	Tiene un rango de cobertura muy amplio alrededor de 5 km en áreas urbanas y 15 km en áreas suburbanas. Consume menos energía y, por lo tanto, la batería dura más. Un solo dispositivo LoRa Gateway está diseñado para cuidar miles de dispositivos finales o nodos. Es fácil de implementar debido a su arquitectura simple. Utiliza la técnica Adaptive Data Rate para variar la velocidad de datos de salida / salida Rf de los dispositivos finales. Esto ayuda a maximizar la vida útil de la batería, así como la capacidad general de la red LoRaWAN.
Desventajas	Se requiere conocimiento del sistema para que el propietario opere dispositivos compatibles con zigbee. No es seguro como el sistema basado en wifi. El costo de reemplazo será alto cuando ocurra algún problema en los electrodomésticos compatibles con Zigbee. Al igual que otros sistemas inalámbricos, la comunicación basada en zigbee es propensa a ser atacada por personas no autorizadas. La cobertura es limitada y, por lo tanto, no se puede utilizar como sistema de comunicación inalámbrica para exteriores. Se puede utilizar en aplicaciones inalámbricas de interior.	Se puede utilizar para aplicaciones que requieren una velocidad de datos baja, es decir, hasta aproximadamente 27 Kbps. El tamaño de la red LoRaWAN está limitado según el parámetro llamado ciclo de trabajo. Se define como el porcentaje de tiempo durante el cual el canal puede estar ocupado. Este parámetro surge de la regulación como factor limitante clave para el tráfico servido en la red LoRaWAN. No es un candidato ideal para su uso en aplicaciones en tiempo real que requieren menor latencia y requisitos de fluctuación limitada.

## *2.2. TECNOLOGÍAS PARA AUTOMATIZACIÓN DE EDIFICIOS 27*

Para el proyecto se ha optado por el uso del protocolo de bus de comunicación distribuido, basado en el estándar RS-485. Una de las razones por la que se escoge este protocolo es por la capacidad que tiene para soportar una gran cantidad de dispositivos en el mismo bus de comunicación. Normalmente estos dispositivos que utilizan puertos RS-485 emplean protocolo Modbus.

Además, al definir los equipos de hardware para el proyecto, el protocolo con el que trabaja estos equipos es en un protocolo Buspro que se basa en el estándar RS-485, es por ello por lo que se define esta tecnología como el sistema a emplearse para el proyecto de desarrollo. También añadir que la inmunidad al ruido que ofrece el estándar RS-485 hace que la interfaz sea muy versátil. Otra de las ventajas, es su distancia de cableado que teóricamente llega a los 1200 metros. En resumen, la capacidad de este estándar para transmitir a altas velocidades, a una gran distancia, con la inmunidad al ruido que soporta y con la gran cantidad de dispositivos que soporta en el mismo bus, lo convierte en una excelente opción para la aplicación que se desarrolla en este proyecto.

## **Capítulo 3**

# **Desarrollo de un sistema de automatización de edificios (BAS)**

Para obtener un buen desarrollo de un BAS se tiene que encontrar un balance donde se obtenga una mayor comodidad a un costo de operación reducido. Para ello un BAS se enfoca en el control optimizado de sistemas que comprenden un edificio como lo son: HVAC, ventilación, calefacción, iluminación y seguridad. Además, si se considera que todo los sistemas y servicios están unificados en una interfaz única en un centro de monitoreo y control esto facilita el trabajo del operador. Los tiempos de respuesta son inmediatos y precisos con un mínimo de esfuerzo personal donde se esperan beneficios indirectos debido a la productividad de la fuerza laboral. Y el registro de datos en el nivel de gestión facilita la adquisición de estos para tareas como instalaciones, contabilidad, asignación de costos, mantenimientos predictivos y correctivos.

### 3.1. Desarrollo del BAS

Si bien la inversión de un BAS resulta un mayor costo a la construcción de un edificio este se ve beneficiado por el costo operacional que representa el edificio durante su vida útil, que según estudios representa ser siete veces la inversión inicial para la construcción [30]. De esta manera se debe enfocar en el análisis donde un concepto de construcción garantice un costo óptimo de ciclo vida y no un costo mínimo de inversión. De esta forma se requiere juntar las experiencias de diferentes campos y las ingenierías se ven obligadas a trabajar en conjunto y colaborar entre sí junto al ingeniero de diseño del edificio formando un solo equipo.

Los sistemas de automatización integrales de hoy generalmente abarcan todo de un BAS, como se observa en la Figura 3.1 donde se muestra las diferentes dimensiones. La norma ISO 16484-2: 2004 especifica los requisitos para que el hardware realice las tareas dentro de un sistema de control y automatización de edificios (BACS). Proporciona los términos, definiciones y abreviaturas para la comprensión de ISO 16484-2 e ISO16484-3. ISO 16484-2: 2004 se refiere solo a elementos/dispositivos físicos, es decir, dispositivos para funciones de gestión, estaciones de operador y otros dispositivos de interfaz del sistema humano; controladores, estaciones de automatización y controladores específicos de aplicación; dispositivos de campo y sus interfaces; cableado e interconexión de dispositivos; herramientas de ingeniería y puesta en servicio [Martirano and Mitolo, 2020].

Los edificios deben proporcionar un entorno productivo y atractivo para los usuarios a la vez que mantiene la eficiencia de costos para maximizarlos ingresos sobre todo el ciclo de vida. Esto incluye específicamente los problemas de gestión. A medida que la administración de instalaciones tiene que ser más eficientes, los servicios de BAS deben estar bien integrados en la automatización de oficinas y flujo de trabajo.

Como ejemplo, considere las salas de conferencias las cuales activan automáticamente el aire acondicionado cuando se hayan reservado. Además, los sistemas de administración de hoteles pueden ajustar automáticamente la operación HVAC, dependiendo de si una habitación se alquila. La asignación de costos para el control del clima y la iluminación con los datos de medición en vivo de BAS y la programación óptima del mantenimiento preventivo basado en el monitoreo automático de equipos y la medición de la hora

de servicio también. Otras dimensiones de los edificios inteligentes son infraestructuras avanzadas para la comunicación de datos y el intercambio de información para promover la productividad, pero también un diseño estructural avanzado.

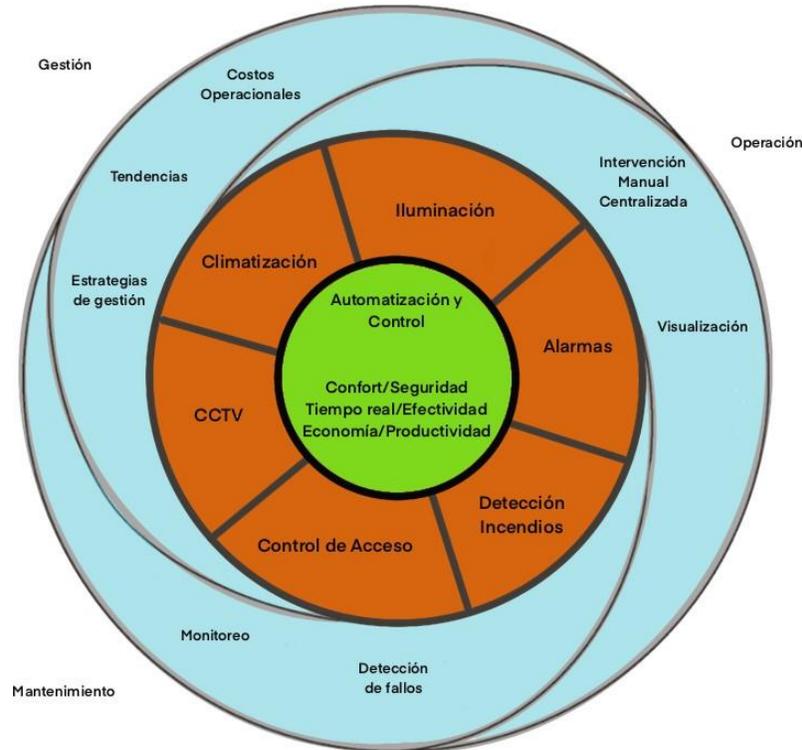


Figura 3.1: Aspectos funcionales de un BAS. Desarrollo: autor [Kastner et al., 2005]

### 3.1.1. Servicios de un edificio con un BAS

Como lo comentamos en el desarrollo de un BAS lo que se requiere es proporcionar una solución donde se puede optimizar los espacios y mantener niveles de confort para los ocupantes. Esto quiere decir que en cualquier tipo de construcción de edificios que se ejecute debe ofrecer un ambiente seguro. Los edificios satisfacen estas demandas mediante el diseño adecuado

de la estructura del edificio y la infraestructura técnica, estos últimos se conocen como servicios de construcción.

Por ejemplo, la ventilación se puede lograr mediante la apertura de ventanas mediante un diseño estructural o por ventilación forzada mediante servicio mecánico de construcción, el suministro de agua puede incluir bombas de presurización, la distribución de energía puede extenderse con monitoreo del factor de potencia o cogeneración in situ, el confort térmico depende de la temperatura del aire, de la humedad del aire, del flujo de aire. La calidad del aire se puede controlar con sensores, con sistemas de ventilación forzada se suministra aire por ductos para proporcionar calefacción y refrigeración. Hay muchos servicios de construcción y cada uno de ellos se parametriza de acuerdo con diferentes factores que se toman en cuenta para lograr un ambiente que se considere apropiado para los ocupantes.

Los diferentes tipos de construcciones de edificios tendrán requisitos distintos según la aplicación para el que fue diseñado. La Tabla 3.1 muestra sistemas que destacan para la utilización en un BAS.

Tabla 3.1: Servicios en edificios  
[Kastner et al., 2005]

<b>Sistemas</b>	<b>Servicios</b>
Control de clima	HVAC (calefacción, ventilación, aire acondicionado), humedad, calidad de aire.
Confort visual	Iluminación, Iluminación artificial, iluminación natural (persianas motorizadas).
Seguridad	Detección de incendios, alarma de gases, detección de fuga de agua, audio evacuación, iluminación de emergencia, CCTV, alarma de intrusión, control de accesos.
Transporte Audio	Elevadores, gradas eléctricas, cinta transportadora. Sistema de distribución de audio.
Suministro y desechos	Distribución de energía, manejo de desperdicios, agua potable, agua caliente, desperdicio de agua.
Comunicación	Networking, PBX, WAN.
Otros	Video Walls, equipamiento presentaciones, sistemas flexibles.

### 3.1.2. Integración de sistemas

Normalmente las construcciones se ejecutan con diferentes contratistas, razón por la que los servicios de construcción se automatizan completamente por separado. Esto se debe a que las empresas contratistas no cubren todas las ingenierías. Sin embargo, todo cambia cuando los fabricantes de las diferentes ingenierías permiten el intercambio de información en sus equipos. De esta forma si tomamos en cuenta que las persianas en una edificación juegan un rol importante en el control de HVAC ya que la radiación solar que ingresa al edificio provoca un aumento en la temperatura del aire, entonces cerrar las persianas automáticamente donde la radiación solar incide puede reducir significativamente el consumo energético. Otro ejemplo es los detectores de movimiento donde pueden proporcionar detección de intrusos durante la noche y control automático de iluminación en horas de trabajo.

Así mismo un BAS hoy debe tener la capacidad de aceptar información con casi todos los servicios de construcción, es así como las tecnologías de la información y de intercambio de datos juegan un papel fundamental en esta área. Por ejemplo, los sistemas de control de acceso no solo se limitan a una apertura/cierre de puertas sino también en un BAS se los utiliza para autenticación biométrica de personal, para iluminación de pasillos al pasar la tarjeta por un acceso, control de ascensores al piso donde la tarjeta del ocupante tiene acceso, restricciones en áreas que el ocupante no puede ingresar, es decir una solución de control de acceso con tarjetas de proximidad nos permite contemplar su uso en otras ingenierías o sistemas. Así mismo los sensores de calidad de aire nos pueden servir también para detección de presencia, medir temperatura, comandos de infrarrojos, etcétera. Contemplamos el uso de un sensor para obtener beneficios adicionales en otros componentes del edificio y optimizamos costos. Los sistemas de CCTV nos permiten estar informados del aforo de un edificio y así mismo se lo puede ocupar para detección de presencia, recuento de personas para un mejor control del sistema de aire acondicionado.

De esta forma es importante tomar en cuenta que todos los beneficios que se pueden obtener con la integración de sistemas se limitan a examinar todos los servicios de construcción de forma conjunta basándose en la funcionalidad y confort de los ocupantes obteniendo eficiencia energética.

### 3.1.3. Automatización y control

Para la automatización de edificios y su control se deben conocer a detalle cada proceso de los sistemas que engloba el correcto funcionamiento del edificio. De esta forma un proceso consta de numerosos subprocesos. Los sistemas y procesos de mayor preocupación al momento de trabajar en un BAS es claramente el HVAC y es porque involucran grandes capacidades térmicas, los parámetros del sistema son graduales. Es así como se debe tener un conocimiento profundo de cada sistema y proceso que compete a cada ingeniería con el fin de lograr la confiabilidad requerida de funcionamiento. La confiabilidad de los sistemas debe garantizar protección, seguridad y confort. Un desafío importante para un BAS es garantizar esta confiabilidad, además, los sistemas deben ser de larga vida útil, deben adaptarse a los avances continuos de tecnología, adoptar nuevos desarrollos, aceptar la mayoría de los estándares internacionales, esto alarga el ciclo de desarrollo de un BAS debido a todos los trámites y procedimientos para conseguir esta estandarización, debe lograrse facilidad de uso y funcionamiento para los operadores e inclusive de los mismos ocupantes del edificio.

### 3.1.4. Arquitectura de un sistema BAS

En la Figura 3.2 se describe una arquitectura general para un BAS. Como lo definimos en un capítulo anterior un BAS tiene tres niveles. A nivel de campo lo definimos como la interacción con el mundo físico, los datos ambientales son recopilados y se transforma en la representación necesaria para su transmisión y procesamiento. También los parámetros del entorno son controlados de manera física en respuesta a los comandos recibidos del sistema. El control automático se encuentra en el siguiente escalón y es el nivel de automatización. Este nivel incluye todo tipo de secuencias que se ejecutan de forma autónoma. Opera sobre los datos recibidos del nivel de campo estableciendo conexiones lógicas y lazos de control. Se puede tener un intercambio de datos entre diferentes entidades de procesamiento para comunicarse entre ellos tomando valores o mediciones de alguna entidad para utilizarla en otra, a esto se le conoce como comunicación horizontal. El nivel de automatización prepara toda la información y valores para enviar mediante la comunicación vertical al nivel de gestión.

El nivel de gestión recibe toda la información de los niveles inferiores y toda la información del sistema es accesible. En este nivel se presenta un interfaz amigable y funcional para el operador del edificio con una

intervención manual del sistema. Se proporciona acceso vertical para manejar los valores del nivel de automatización, modificar parámetros, por ejemplo, se puede modificar rutinas, horarios, activaciones/desactivaciones de eventos, etcétera. Las alertas se muestran en el interfaz creado y aparecen por malfuncionamiento de algún sistema, fallas técnicas o en condiciones críticas. Se mantiene un registro de datos para almacenamiento de los datos históricos con la posibilidad de generar informes y estadísticas.

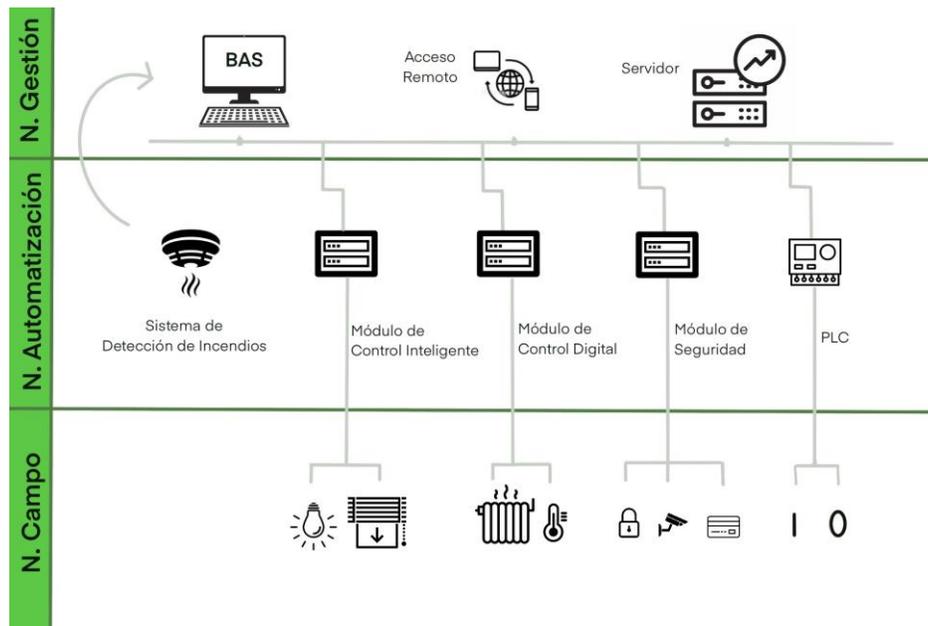


Figura 3.2: Arquitectura general de un BAS. Desarrollo: autor [NETxAutomation, 2019]

Por lo general, la automatización se diseña en forma distribuida, con múltiples unidades de procesamiento responsables de subprocesos contenidos localmente o por separado. Los beneficios de este tipo de diseño distribuido son reducir las latencias en los lazos de control, evitar puntos únicos de falla, reducir el riesgo de cuellos de botella en el desempeño y permitir que los subsistemas estén fuera de servicio debido a fallas o mantenimiento programado sin afectar otras partes. Ciertamente, los sistemas distribuidos son más difíciles de diseñar y manejar que los centralizados. [Kastner et al., 2005]

### 3.2. Desarrollo

Actualmente el edificio está terminado y listo para sus operaciones. Antes de continuar con las necesidades de equipamiento para el desarrollo del sistema BAS, se identifican los componentes que afectan directamente al consumo energético parámetro esencial para el correcto funcionamiento del sistema BAS. Uno de los componentes principales es la iluminación la cual el edificio utiliza diferentes tipos de lámparas LED y tiene un funcionamiento de encendido/apagado es decir no contemplan el uso de luminarias dimerizables, por ende el equipamiento para el control de las luminarias debe contemplar el uso de relés con la capacidad de soportar mínimo 5A por canal de relé y que sean diseñados para control de lámparas tipo incandescente, LED, halógenas. El edificio no contempla el uso de sistemas de climatización o ventilación en el edificio sin embargo, se ha considerado en esta tesis puesto que es un componente importante en el consumo energético para edificios que si contemplan su uso y además, debido a la pandemia que vivimos la recirculación de aire permite reducir las exposiciones del SARS-CoV-2, el virus que causa el COVID-19 y se vuelve un componente importante a considerar en futuras construcciones en zonas geográficas donde el sistema de ventilación o climatización no es necesario. Para este componente es necesario definir un equipo que pueda controlar sistemas centralizados de ventilación mediante activación/desactivación de los equipos por lo que se ocupará relés de igual forma y contactos secos para control. Se requiere controlar equipos de detección y activación de alarmas para prevenir sucesos que pueden ocurrir dentro del edificio como hurto, incendios, etc. Para ello se requiere utilizar contactos secos que permitan el control de estos sistemas.

Para la propuesta de desarrollo de un BAS se plantea la utilización de software y hardware especializado en la gestión de edificios que tengan la capacidad de controlar y monitorear dispositivos. Para ello se propone que el software cumpla con ciertos requisitos como:

- La capacidad de desarrollar interfaces que permitan crear entornos que faciliten al usuario la operación del edificio y que su desarrollo cumplan con la exigencia y necesidad de cada proyecto.
- Que, mediante el uso de un servidor se pueda operar localmente sin necesidad de acceso a internet.

- Que el software tenga la virtud de trabajar con equipos de otras marcas. Es común utilizar equipos que operan bajo diferentes protocolos en el caso de grandes proyectos de automatización. Por ejemplo: KNX, Buspro, Zigbee para controlar iluminación, persianas, Modbus para sistemas de ingeniería sea bombas, calderos, TCP/IP para equipamiento de audio/video.
- Que el software permita generar drivers propios (programación personalizada) de acuerdo con la complejidad lógica de trabajo de los equipos e integración de los mismos al sistema. Para ello se utiliza JavaScript.
- Que se pueda operar mediante un servidor para que los datos estén protegidos. El servidor debe tener la capacidad de funcionar las 24 horas del día y puede interactuar con las interfaces de usuario a través de un protocolo propio.
- Que permita intercambiar datos e información de nuestro sistema desarrollado a MQTT y enviarlos a servicios en la nube y plataformas IoT, logrando integrar diferentes objetos de automatización en un único sistema global.
- La mayoría de controladores para sistemas profesionales están escritos en lenguaje C++ ya que su comportamiento es rápido y estable. El software tiene la capacidad de no necesitar pasarelas (gateways) para cada sistema sino que se acople sin el uso de ellas para controlar diferentes sistemas, optimizando el uso de hardware.
- Generar gráficas y tendencias del comportamiento de los equipos y sus consumos.

Es así que en nuestra propuesta de desarrollo incluye la evaluación de estos requisitos para definir el software. Partimos del software ya que es la pieza fundamental al momento de definir un sistema de gestión. Es así que se propone el uso de una plataforma de gestión llamada iRidium mobile, cuya empresa se encuentra ubicada en Rusia. Este software que hemos escogido cumple con los requisitos que necesitamos para la gestión de una edificación.

Además, que permite la creación de nuevos drivers que permiten adaptarse a protocolos que no se integran actualmente. Esto presenta una gran ventaja sobre otros sistemas ya que normalmente los sistemas de gestión trabajan

sobre un protocolo propio lo que obliga a trabajar con el equipamiento de la misma marca o en el mejor de los casos adquirir pasarelas (gateways) para poder integrarse al software de gestión incurriendo en gastos adicionales y esa integración no es nativa propiamente dicha sino que permite una limitación en el uso del equipamiento. iRidium por otro lado al ser una empresa que no comercializa equipamiento de hardware sino se enfoca en el desarrollo de su software le interesa integrarse a todos los sistemas de automatización posibles. Para ello gracias a su lenguaje de JavaScript y la API de iRidium se pueden crear visualizaciones avanzadas para integrarse a equipos de marcas terceras logrando integrarse prácticamente a cualquier protocolo de comunicación.

El equipamiento de hardware debe cumplir con los siguientes requisitos para el correcto funcionamiento de nuestro sistema BAS:

- Canales de relés y dimerización para control de iluminación.
- Entradas de contactos secos para sensores y alarmas.
- Canales de relés para sistemas de aire acondicionado.
- Control IP para conexión de equipamiento a la red y nube.
- Sensores de movimiento, temperatura y luxes.
- Fuente de alimentación para red de comunicación.
- Panel de pared para control y monitoreo.

Hemos analizado diferentes marcas para el equipamiento de hardware. Es de esta forma que hemos escogido un fabricante que primero se integre al software de gestión ya que creemos es un parámetro importante para trabajar nativamente con los equipos. Se define la marca HDL la cual es una empresa de procedencia China que permite una integración total de sus equipos con el software de gestión. La gama de productos que oferta facilita la gestión de todos los componentes que engloban un edificio. Por último los costos de los productos son económicos en comparación a otros proveedores como Schneider Electric, Johnson Controls, Bosch, etcétera.

Para el obtener el resultado del desarrollo del BAS en el edificio se plantean los siguientes lineamientos. El primer lineamiento es el emplazamiento del edificio en el programa AutoCAD para identificar los sistemas a controlar

en el caso puntual del proyecto se controla la iluminación y climatización. Como siguiente lineamiento se procede con la impresión de una planta del edificio con tecnología 3D para instalar los sistemas de iluminación y climatización. El tercer linemiento es la selección de equipos para el control y monitoreo. Y como cuarto y último lineamiento el desarrollo del software de gestión para la operación del usuario.

Para la realización del proyecto se emplaza la planimetría del edificio en el programa AutoCAD y en el se identifican los espacios que controlaremos y monitorearemos en el software de desarrollo. En el caso del edificio “Padre Juan Bottasso” las plantas tienen el mismo emplazamiento en los cuatro pisos del edificio. El edificio cuenta con un área de 1200 metros cuadrados de construcción, en el se haya el subsuelo el cual cumple la función de espacios para laboratorios. Después tenemos la planta baja que está concebida para áreas de coworking y startups. Finalmente, desde la primera planta alta hasta la tercera planta alta el edificio se compone de 6 aulas por cada piso, oficinas y un espacio de sala de múltiples servicios. Estos planos se los puede apreciar en la sección de Anexos. A continuación se muestra fotografías de la edificación en la actualidad para ello referirse a las Figuras 3.3 y 3.4.



Figura 3.3: Vista frontal del edificio “Padre Juan Bottasso”.



Figura 3.4: Vista posterior del edificio “Padre Juan Bottasso”.

La planimetría para el componente de iluminación se visualiza en el apéndice de Anexos. Para este componente se presenta una sola planta debido a que las demás plantas son réplicas de esta planta. En este plano desarrollado en AutoCAD se emplaza las luminarias utilizadas tanto en espacios de trabajo como en pasillos.

El edificio no cuenta actualmente con un sistema de climatización o ventilación en el edificio. Esto normalmente no se planifica debido a las condiciones climáticas que se tiene en la ciudad de Cuenca durante todo el año que promedian los 13 grados centígrados y llegan a 20 grados centígrados en máxima temperatura promedio. Es por ello que hasta antes de la pandemia no se veía la importancia de instalar sistemas de climatización en los edificios. Sin embargo, en la actualidad de acuerdo a recomendaciones de la OMS se observa la importancia de contar con sistemas que permitan el flujo de corrientes de aire constantemente. La planimetría de igual forma se encuentra en Anexos y en este diseño muestra la ductería y rejillas para el sistema de climatización. Se presenta

una sola planta porque los demás pisos del edificio es una réplica de este piso.

Una vez realizada la planimetría se empieza con la fabricación de una maqueta la cual es una réplica del edificio en una sola planta. Para el desarrollo del proyecto se replicó un piso en una maqueta impresa con tecnología 3D del edificio “Padre Juan Bottasso” de la Universidad Politécnica Salesiana, ubicada en la ciudad de Cuenca. En las Figuras 3.5 y 3.6 se muestra el plano en 2D y 3D del edificio y en las Figuras 3.7 y 3.8 una fotografía de la maqueta.

Para la impresión 3D primero se realizó un levantamiento del piso del edificio en el programa sketchup pro, para conseguir simetría en la escala de la impresión. Después se exporta el archivo generado en STL para que se pueda abrir en el software de la impresora 3D el cual se llama Prusa Slicer. En este software se parametriza los componentes como, por ejemplo: materiales, calidad de impresión, patrones de relleno, soportes, etcétera. Con ello realizado se procede con la impresión. La edificación cuenta con 4 pisos emplazados con el mismo diseño en cada piso. Mediante esta replica obtenida se desarrolló un software de gestión para edificios buscando la gestión de los siguientes sistemas: control de iluminación, control de aire acondicionado y calidad del aire. Se desarrolla una maqueta impresa para interactuar entre el desarrollo del software y el hardware, de tal forma que se pueda visualizar el funcionamiento de la edificación con el software de desarrollo. De esta manera se busca comprobar que la gestión de la edificación sea centralizada en el software.

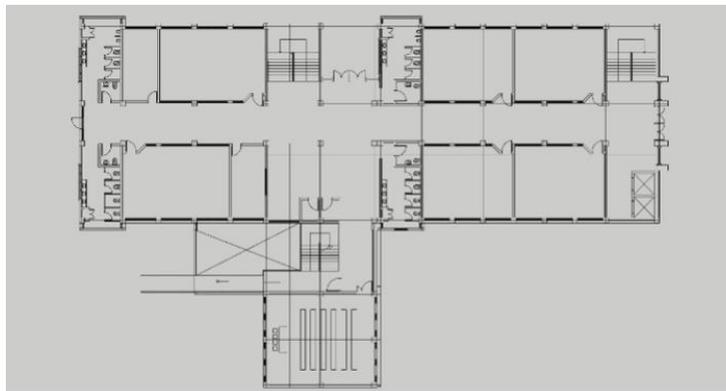


Figura 3.5: Diseño en programa sketch up pro.



Figura 3.6: Vista 3D en programa sketch up pro.

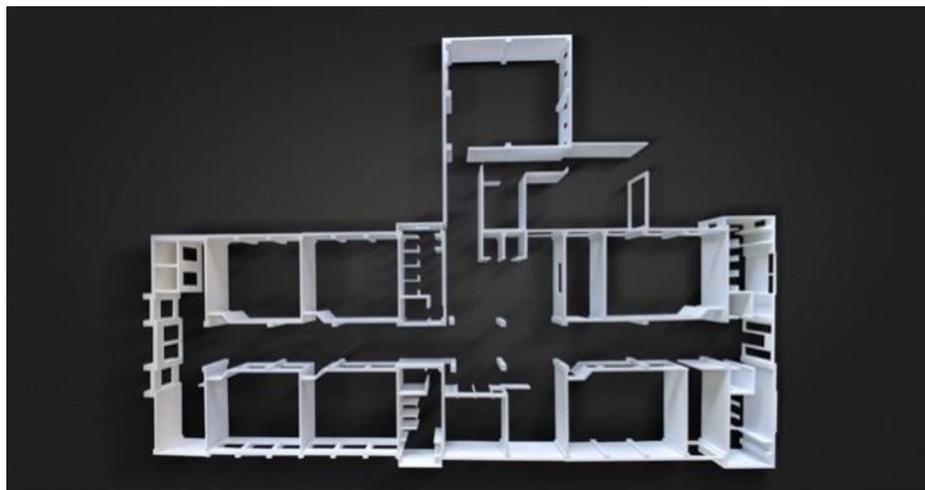


Figura 3.7: Fotografía de la maqueta impresa.



Figura 3.8: Fotografía de la maqueta, servidor y equipos utilizados.

En el sistema de gestión se ocupan equipos y un software especializado en crear entornos de automatización personalizados de acuerdo a la solución requerida. Dentro del marco de equipamiento físico se utilizan equipos de la marca HDL Buspro y para el software se utiliza la plataforma de automatización Iridium. El software comercial es licenciado, sin embargo, se puede desarrollar un proyecto de forma gratuita incluyendo pruebas de funcionamiento con los equipos.

### 3.2.1. Hardware

Para optar por un fabricante de hardware se definen tres parámetros, el primero, la integración de sus equipos con software de terceros que permita la comunicación con otros fabricantes, segundo, la apertura del fabricante en permitir manipular su software de aplicación por parte del desarrollador del sistema de gestión y tercero, los beneficios tanto de productos como de costos al momento de su adquisición.

Cuando se habla de la construcción de un edificio este puede tener diferentes fines, tenemos edificios comerciales, edificios residenciales, edificios para hoteles, edificios para hospitales, y edificios de otras funcionalidades. Es así como, al buscar fabricantes que brinden estos productos se debe tomar en cuenta mínimos detalles de acuerdo con la función que va a tener el edificio.

Dentro de los fabricantes más conocidos en los sistemas BAS, tenemos: Schneider Electric, Johnson Controls, Honeywell, Bosch, y más. Estas empresas presentan soluciones tanto de hardware como de software. Sin embargo, el mayor inconveniente al optar por estos fabricantes es su nivel de integración, ya que son sistemas muy cerrados que buscan trabajar con sus propios productos, si bien tienen opciones para integración con marcas terceras los precios en equipos aumentan. Es así como al tomar en cuenta el medio en el que vivimos donde se busca optimizar costos de implementación además de buscar alternativas que permitan cumplir con los parámetros esenciales previamente definidos, se ha determinado que hay soluciones en el mercado con las que se puede alcanzar sistemas de gestión para edificaciones.

De acuerdo con lo analizado se ha optado por escoger al fabricante de productos de automatización HDL Automation debido a sus prestaciones en equipamiento la cual permite una comunicación nativa con empresas de terceros. Además, su variedad de productos permite que sea una gran alternativa para utilizarlo en soluciones residenciales, comerciales, hoteleras, hospitalarias y más. Así, esta empresa ofrece un sistema de automatización integral considerando el tipo de edificio a construir. Otro detalle interesante es que los costos de productos son asequibles en comparación a los grandes fabricantes existentes en el mercado.

HDL Automation es una compañía que brinda productos de automatización y que se encuentra situada en Guangzhou, China. Está enfocada en dar soluciones para hogares, edificios y hoteles inteligentes. Dentro de sus proyectos más importantes han ejecutado soluciones para el aeropuerto internacional de Dubai, el edificio 41X del Instituto Australiano de Arquitectura, el apartamento MOMA contemporáneo de Beijing, el hotel Macau Morpheus [HDL Automation, 2021e].

El sistema de control inteligente de HDL utiliza tecnología alámbrica e inalámbrica. El sistema alámbrico utiliza un protocolo propietario de los sistemas HDL, llamado BusPro, que tiene la facilidad de integrarse con

otros sistemas de control. BusPro se conforma de una red de malla en toda su aplicación. Esta red es el sistema central de la comunicación para todos los sistemas internos que se utilizan a través de HDL. Para la comunicación se requiere un solo tendido de cable en toda la edificación para crear una red BusPro. El sistema de control inteligente le permite controlar, monitorear y gestionar los equipos en el edificio a través de los paneles de pared, teléfonos inteligentes, tabletas o computadoras. Además, las funciones de sensores, rutinas, horarios, lógica en general se pueden combinar para que el sistema de control inteligente ejecute automáticamente las tareas relacionadas [[HDL Automation, 2021f](#)]

El sistema Buspro tiene posibilidad de integrarse con otros sistemas de control a través de un ajuste de protocolo. Dentro de las opciones que Buspro puede manejar son: iluminación, dimerización, aire acondicionado, calefacción por suelo radiante, entretenimiento en el hogar, seguridad y protección, etc. Adecuado para edificios de oficinas, residenciales, complejos, hoteles y otros edificios comerciales.

Una solución de automatización de edificios comerciales como es el caso de nuestro proyecto se basa en una solución de sistemas inteligentes. El cual permite controlar y gestionar la iluminación, audio, alarmas, cortinas, climatización, sistema de seguridad y más funciones del edificio. Durante muchos años, las soluciones de automatización de edificios comerciales con marca HDL sea han diseñado con la premisa de ofrecer comodidad, confort y eficiencia energética. Es de esta forma que para el desarrollo del proyecto se utiliza equipamiento de la marca HDL bajo el protocolo de comunicación BusPro. Los equipos que cumplen las características necesarias se detallan a continuación:

- RCU Room Control Unit: la Figura 3.9 representa el módulo RCU el cual es una unidad que permite integrar en un solo dispositivo diferentes funciones. Es así que la capacidad de este equipo permite controlar hasta 48 canales de relés, contactos secos y dimerización. Cuenta con 24 contactos secos. Tiene estados de alarmas y salidas LED, comunicación por RJ45 para conexión de red. Su conexión a la red permite que pueda ser integrado con softwares de otros fabricantes para intercambiar información.

Este dispositivo también permite monitorear y controlar equipos

de aire acondicionado, ventiladores, cortinas, iluminación, se puede visualizar el estado de los aires acondicionados y otros estados enviando al software de monitoreo del servidor en tiempo real.

El RCU soporta hasta 4 aires acondicionados, cada aire acondicionado puede ser controlado por hasta 3 paneles (termostatos). Si el aire acondicionado está centralizado, puede asignar un canal del RCU o del módulo afiliado a la FCU (unidad de fan coil). Si el aire acondicionado está centralizado, pero no desea asignar canales a un módulo asociado, o desea utilizar el control IR, no es necesario asignar los canales. Introduzca sólo el ID del panel de panel de CA (termostato).

Configuración: El aire acondicionado puede configurarse para que durante las diferentes estaciones del año se genere la temperatura más adecuada.



Figura 3.9: RCU Room Control Unit.  
[HDL Automation, 2021d]

- 750mA Power Supply Module: la Figura 3.10 representa el módulo de alimentación es de de 750 mA y proporciona voltaje de 24 V CC. Todo sistema de control tiene una fuente de alimentación para su bus de comunicación la cual puede ser fuentes comunes en el mercado que cumplan con el voltaje de alimentación requerido por el sistema de comunicación. En el caso del proyecto se utiliza una fuente del mismo fabricante.



Figura 3.10: 750mA Power Supply Module.  
[HDL Automation, 2021c]

- DLP Touch: la Figura 3.11 representa el panel de pared que permite un control táctil. Mediante este equipo se puede controlar todo el sistema de automatización de una forma rápida y sencilla. Se pueden editar sus íconos para identificar rápidamente que se va a controlar.



Figura 3.11: DLP Touch.  
[HDL Automation, 2021b]

- Ceiling Mount 8-in-1: la Figura 3.12 representa el sensor el cual cuenta con 8 funciones en 1 solo dispositivo. Su montaje es en techo. Dentro de sus capacidades incluye el sensor de movimiento PIR, sensor de

temperatura, sensor LUX, contacto seco, bloques lógicos, control por IR, etc. El sensor admite el modo de seguridad HDL y puede descargar el código IR de la herramienta de configuración HDL Buspro y emitir códigos IR para controlar equipos. Normalmente los sensores de diferentes fabricantes como: Bosch, DSC, Paradox, Honeywell, y más, son usados para aplicaciones específicas como: sensor de movimiento o sensor de temperatura. En el caso del sensor 8 en 1 presenta una ventaja sobre sus competidores y son sus 8 funciones en un solo equipo para lo cual el sensor permite además contiene dos canales de relés.



Figura 3.12: Ceiling Mount 8in1 sensor.  
[HDL Automation, 2021a]

Se ha definido la utilización de estos dispositivos para el correcto funcionamiento del sistema BAS. El equipo RCU Room Control Unit es un dispositivo completo para las diferentes necesidades del edificio como el control de iluminación con sus canales de relés y dimerización, se pueden controlar hasta cuatro aires acondicionados por cada módulo utilizando los primeros 4 canales de relés del módulo. Las entradas de contacto seco nos permiten integrar con equipos de marcas terceras, conectar sensores para detección, activación/desactivación de alarmas. El módulo IP integrado nos permite gestionar y monitorear todo el proyecto en la nube. El equipo 750mA power supply permite mantener el todos los dispositivos alimentados y en red. Es importante que el operador además de la interfaz de usuario del software a desarrollar tenga la opción de controlar desde un panel en pared para ello se utiliza un DLP Touch el cual permite el manejo del sistema desde un panel de pared. El sensor 8 en 1 representa en nuestro proyecto el equipo que permite sensar el movimiento de las personas dentro del edificio además nos facilita en la medición de los luxes para generar escenas y ambientes de acuerdo a la necesidad del espacio medido.

### 3.2.2. Software

Para definir el software en un BAS es crucial coordinar conjuntamente con el hardware a utilizar. El software de desarrollo debe permitir la integración con los equipos de la edificación ya sean de un fabricante de diferentes fabricantes, además es importante que permita tener la habilidad de manipular y controlar cualquier equipo que abra su software de aplicaciones. Para ello se ha optado por un software que de facilidades para el programador en la integración con la mayor cantidades de estándares y protocolos de comunicación posible.

De la misma forma que permita controlar casi cualquier equipo que de apertura a la manipulación de las API (Interfaz de programación de aplicaciones) de los productos. Con base en esto se opta por el uso del software de control de la empresa iRidium que inicia operaciones en el año 2006 en Rusia. iRidium es una plataforma de integración que permite controlar y monitorear sistemas de automatización de diversos fabricantes y protocolos de comunicación además de dispositivos IoT. iRidium maneja diferentes plataformas para soluciones residenciales la solución iRidium lite y para soluciones comerciales iRidium pro. Nos enfocamos en iRidium pro para el desarrollo del software de gestión, aclarando que es un software licenciado.

iRidium pro es una plataforma IoT que consiste en:

- Dispositivos IoT.
- Base de datos.
- Multimedia.

Kit de herramientas para:

- Creación de GUI.
- Creación de módulos de controlador.
- Automatización de edificios/procesos.

Para ello iRidium pro trabaja en conjunto con componentes que son indispensables para obtener la solución global de un BAS:

- iRidium studio: es el entorno de programación para desarrollar interfaces y proyectos en el servidor.
- i3 pro: es la interfaz de usuario para controlar la edificación.
- iRidium server: un controlador para sistemas de automatización e IoT, destinado a recopilar y analizar datos, exportar e importar bases de datos, crear rutinas y horarios, intercambiar datos entre diferentes protocolos, etc.
- Drivers: una base de controladores para diferentes sistemas (AMX, Crestron, Bacnet, Modbus, Helvar, etc.).
- iRidium cloud: un servicio en la nube para almacenar proyectos de visualización, proyectos de servidor y licencias.
- iRidium JavaScript API: un conjunto de funciones y eventos API iRidium y JavaScript 1.5 destinados a controlar la visualización y los controladores en proyectos iRidium.

La arquitectura de iRidium se basa en el nivel de campo, nivel de conexiones, nivel de controladores locales y nivel de aplicación como se muestra en la Tabla 3.2.

Tabla 3.2: Arquitectura  
[iRidium Studio, 2021a]

<b>Arquitectura</b>	<b>Servicios</b>
Nivel de campo	iRidium puede ser usado con dispositivos IoT. El firmware esta diseñado para trabajar con diferentes chips: Atmega328, STN32, ESP8266, etc. La configuración se realiza en el software iRidium studio.
Nivel de conexiones	Para la comunicación de entre dispositivos IoT, iRidium soporta múltiples protocolos y estándares de comunicación como por ejemplo: CAN, ethernet, RS485, modbus, Bacnet, etc.
Nivel de controladores	iRidium implementa el enfoque de computación Edge/Fog. iRidium Server como controlador de automatización universal en el objeto y también iRidium Server como puerta de enlace universal para sistemas multiprotocolo.
Nivel de aplicación	iRidium proporciona un conjunto de herramientas para la creación de aplicaciones de IoT. Interfaces GUI para los principales sistemas operativos. Generación de tendencias/gráficos.

La compatibilidad de iRidium pro es con cualquier API abierto. Tiene una potencial compatibilidad con todos los estándares de IoT industrial por ejemplo la Figura 3.13, de tal forma que sus funciones de soluciones son casi ilimitadas.

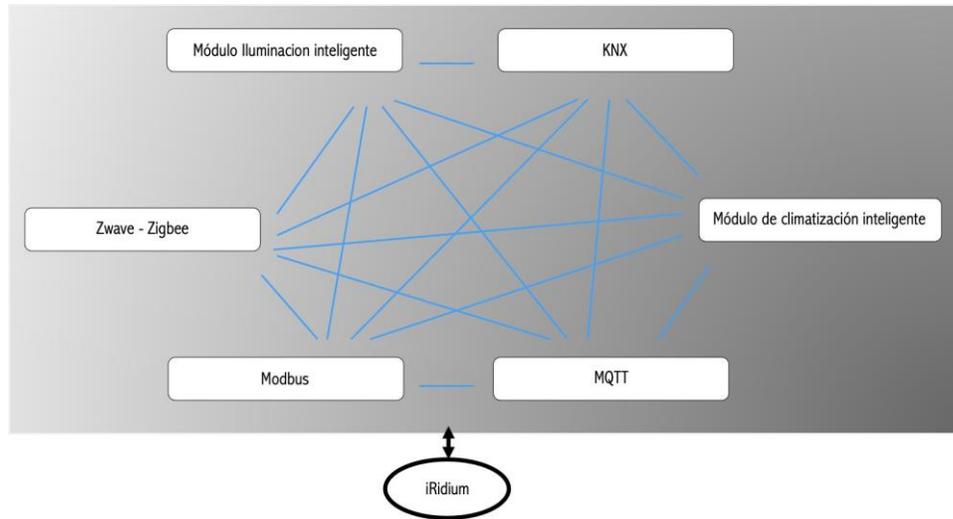


Figura 3.13: Compatibilidad iRidium. Fuente: autor

### 3.2.3. Interfaz de usuario y programación

Antes de iniciar con la programación se deben cumplir con requerimientos que solicita iRidium y estos son:

- Registro en el sitio web: registrarse en el sitio web de iRidium. El registro en el sitio web es necesario solo para integradores y desarrolladores. Se envía a un usuario final una invitación por correo electrónico para utilizar un proyecto.
- Instalación de software: para usar iRidium se debe descargar el paquete completo de iRidium para Windows. En el incluye:
  - iRidium studio: entorno para crear una interfaz gráfica (GUI) y configurar el servidor iRidium.
  - i3 pro: una aplicación para Windows para lanzar proyectos creados en iRidium Studio.
  - iRidium Transfer: para cargar interfaces en paneles de control para realizar pruebas.
  - iRidium Server: para almacenar una base de datos y lógicas y para una conexión centralizada con paneles de control.

- Obtener licencia: para que un panel de control funcione con equipos, se requiere una licencia que mantiene un costo de acuerdo a las variables a controlar como se observa en la Tabla 3.3. Después del registro en el sitio web, tiene el modo de prueba, el cual funciona por un tiempo limitado y que permite la interacción del software con los equipos. Hay tres tipos de licencias: visualization only license que funciona solo con la interfaz de usuario i3 pro sin considerar el iridium server. Tenemos server only license si se requiere solo lógicas y trabaja con datos y no requiere control de interfaz de usuario. El tercer tipo de licencia es visualization + server license el cual contiene los beneficios de las dos licencias anteriores. El costo según el tipo de licencia que se escoga varía de acuerdo con la cantidad de paneles y tags a ocupar en el proyecto. Para paneles empieza desde 1, 5, 10, 25 e infinita cantidad de paneles. Para las tags empieza 50, 250, 1000, 2500 y 10000 tags. Además, el costo también varía con el estatus de cuenta de dealer que se mantenga. Para ello hay estatus de Gold, Silver y Bronze. El descuento se describe de la siguiente manera:

Tabla 3.3: Arquitectura  
[iRidium Studio, 2021a]

<b>Estado</b>	<b>Bronce</b>	<b>Plata</b>	<b>Oro</b>
Descuento	20 %	33 %	40 %
Modo prueba	5 minutos	20 minutos	20 minutos
Condiciones	Ninguna	1.000,00 USD/año	3.500,00 USD/año

En el caso de nuestro proyecto luego de analizar los costos y necesidades de nuestro software de desarrollo para comprobar su funcionamiento, se utiliza el estatus Bronze con una compra de licencia con 20 % de descuento y con modo de prueba del sistema.

### 3.2.4. iRidium Studio

En este apartado se profundiza el funcionamiento del software. Es así como empezamos con los componentes de visualización del proyecto, previo a esta revisión de los componentes se indica los pasos a seguir de un proyecto global para su correcto funcionamiento y en la Figura 3.14 se observa el diagrama de flujo del software iRidium:

- Registro de credenciales en iridiummobile.net e iRidium cloud.

- Descarga e instalación de los componentes para configuración del programa iridium.
- Crear de los componentes de visualización y parametrizar la conexión a equipos (iRidium Studio).
- Pruebas de funcionamiento con el emulador y iRidium transfer.
- Subir el proyecto a la nube y compartir con los usuarios para pruebas de funcionamiento.
- Adquirir licencia y compartir proyecto con los usuarios.

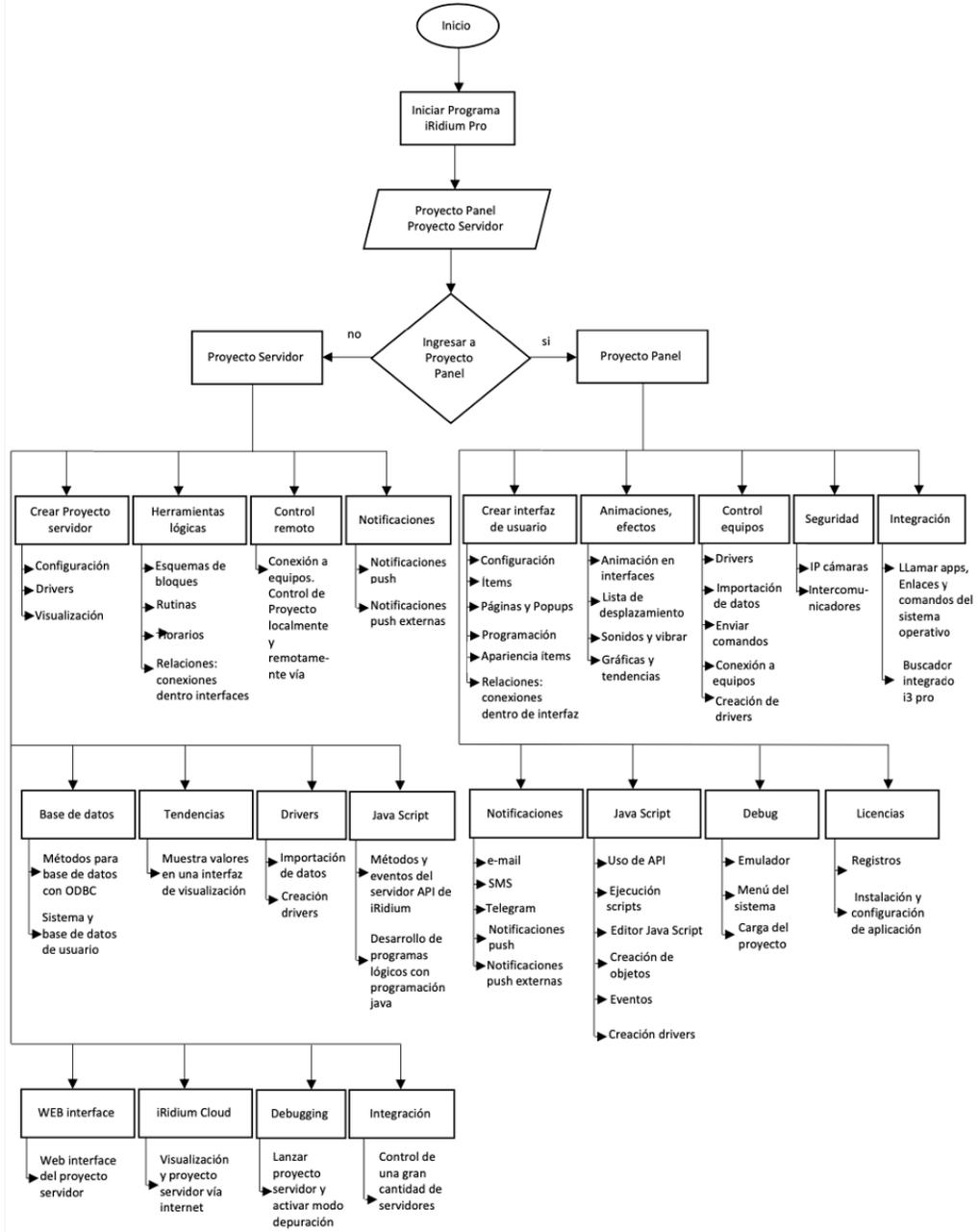


Figura 3.14: Diagrama de flujo del programa iRidium. Fuente: autor

### Componentes de visualización

Para la interfaz de usuario se empieza creando un proyecto en iRidium Panel Project el cual muestra una interfaz gráfica que facilita el control por el operador. Además, se completa el proyecto con la creación del proyecto servidor llamado Server Project el cual cumple la función de ser un servidor controlado desde una interfaz web.

Ambos proyectos se crean desde el software principal iRidium Studio el cual engloba la programación tanto para el Panel Project como para el Server Project. Mediante esta introducción se empieza por definir en la Figura 3.15. los componentes del iRidium Panel Project:

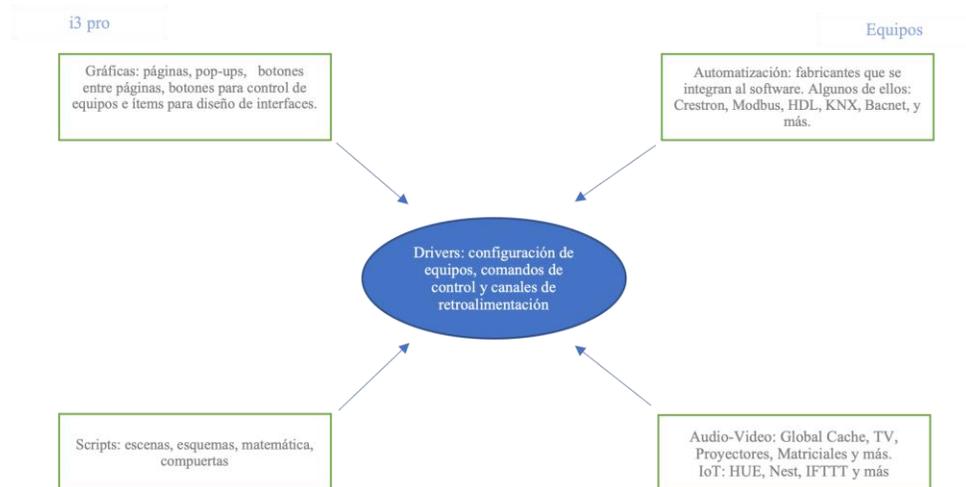


Figura 3.15: Componentes iRidium Panel Project. Fuente: autor

En el iRidium Panel se crea la interfaz de usuario de acuerdo con los requerimientos que se tenga en el edificio. En el caso del proyecto desarrollado se usa la siguiente interfaz de usuario y que se observa en la Figura 3.16

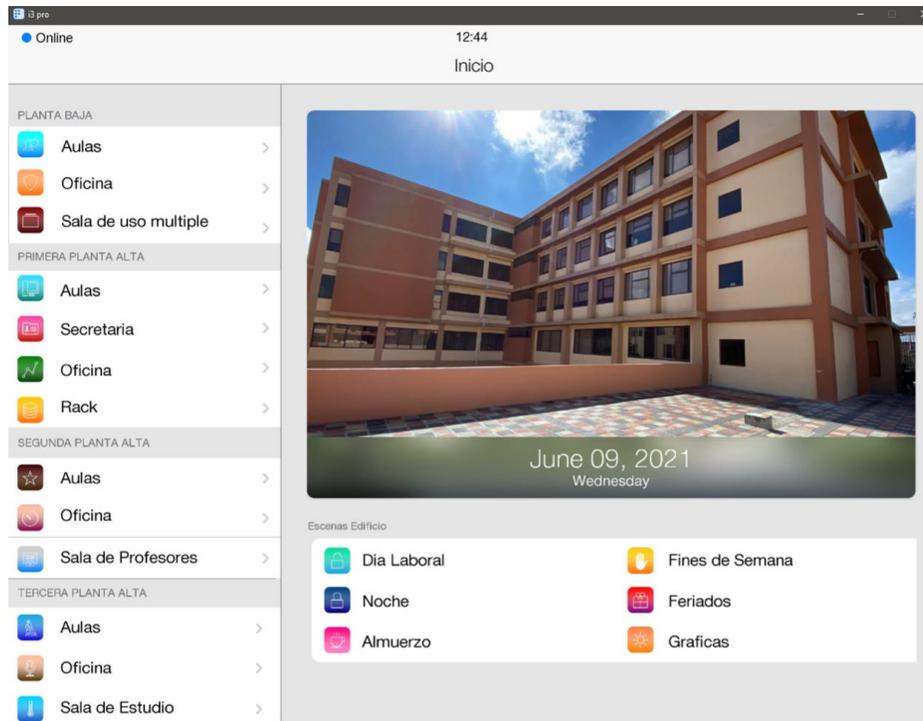


Figura 3.16: Interfaz de usuario. Fuente: autor

### Interfaz de iRidium Studio

Para la interfaz de iRidium Studio abrimos la carpeta previamente descargada de la página web de iRidium Pro Components y abrimos el programa iRidium Studio.

En el interfaz se crea un nuevo iRidium Panel Project y este se refleja en la parte superior izquierda en la columna Project, y esta columna es la responsable de las gráficas, íconos, páginas y pop-ups. Las propiedades de cada ítem se encuentran en la misma columna Project en la parte inferior. En la columna derecha del programa en la parte superior se muestra las propiedades de los drivers y está directamente relacionada con los equipos y fabricantes de productos de automatización, controles de comandos y canales de retroalimentación. Estas son las herramientas para trabajar en el espacio de trabajo que se encuentra en el centro del software. En la parte superior se observa el menú para el diseño y parametrización de la interfaz a crear.

### Creación del iRidium Panel Project

Para la creación de un iRidium Panel Project y un iRidium Server Project se lo realiza dirigiéndose a File, New y luego elegir el proyecto a crear, como se muestra en la Figura 3.17. Automáticamente aparece un pop-up con parámetros que se deben completar siguiendo las instrucciones y depende del dispositivo final en el que se va a operar. Después es importante revisar la configuración del proyecto y esto se lo realiza presionando el botón de configuración. En el se puede configurar la resolución del proyecto, protector de pantalla, contraseñas de ingreso, en que página inicia el programa y más opciones.

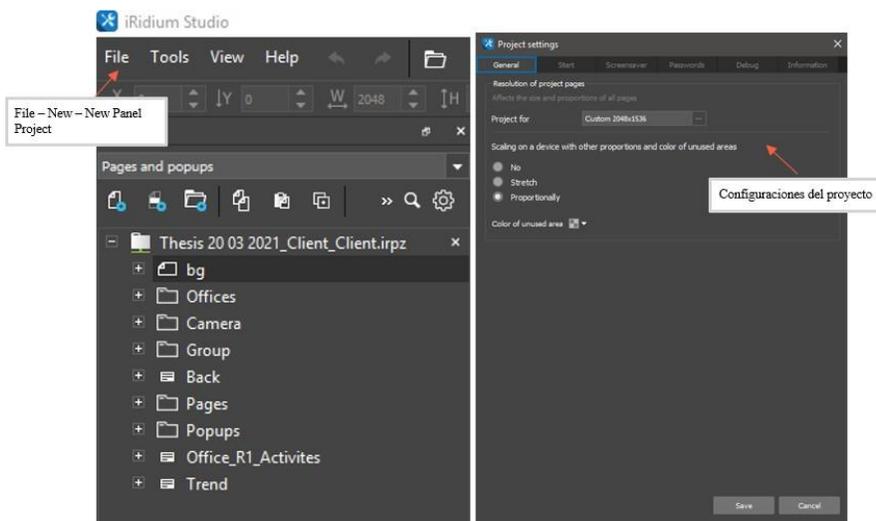


Figura 3.17: Creación iRidium Panel Project. Fuente: autor

En esta sección se empieza a crear las páginas, pop-ups, ítems e íconos de acuerdo con la necesidad de cada proyecto. En el caso del proyecto en ejecución se obtuvo una interfaz que se observa en la Figura 3.18.

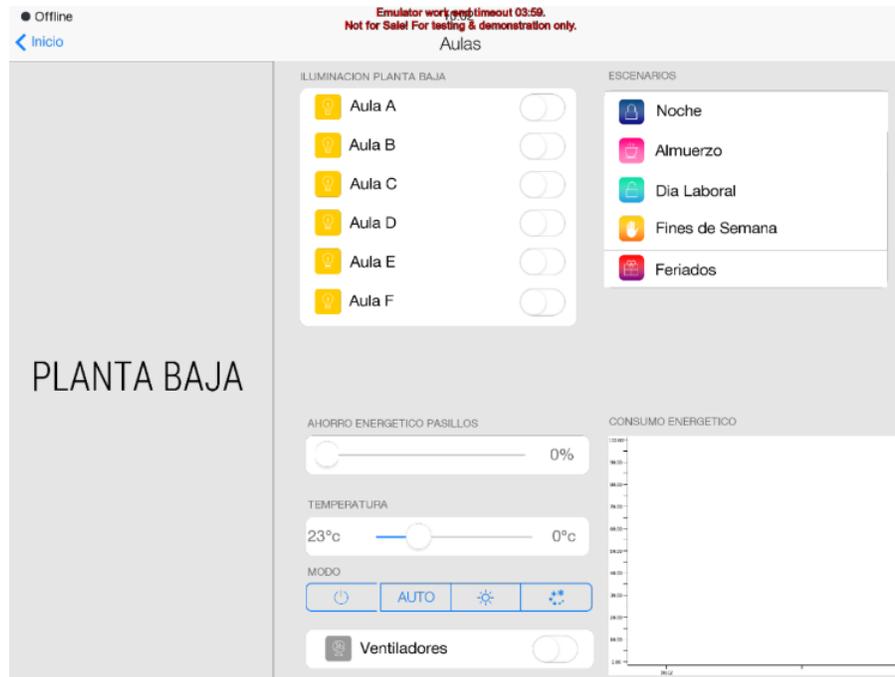


Figura 3.18: Software iRidium Studio. Fuente: autor

### Tipos de gráficas y propiedades

Cada ítem se crea al presionar el botón de “Draw Ítem” y este se parametriza de acuerdo con el requerimiento de ese botón, y esto se realiza en las propiedades del objeto. En las propiedades del objeto se puede configurar las coordenadas de los ítems, después de eso se encuentran parámetros específicos como lo es el tipo de botón, el de canal de retroalimentación que significa como va a reaccionar un botón al ser presionado o como reaccionar cuando recibe un comando de un equipo, también el tipo de presión o queno ejecute nada al presionar se puede configurar en este espacio. Después se tiene la programación del botón que es la pestaña donde se asignan comandos para cumplir con una acción específica. Por último, se tiene la pestaña de estados, que permite personalizar al botón programado.

Dentro de las propiedades de los objetos tenemos:

- Botón: el boton se utiliza para presionar. Tiene 2 formas: "sin presionar"(falso) y "presionado"(verdadero). Cuando se presiona un botón, se puede enviar un comando. El valor que se envía se selecciona

en el comando y puede ser un número o una cadena. Pueden haber diferentes configuraciones como que un estado “presionado” de un botón significa que un usuario presionó el botón y este no vuelve al estado no presionado cuando se suelta el botón. Otras configuraciones significan que un botón cambia su estado a “presionado” si obtiene del equipo cualquier valor que no sea cero. El cambio de estado se puede desactivar y un botón se puede utilizar como etiqueta [iRidium Studio, 2021c].

- Etiqueta: La etiqueta es un elemento que no se presiona. Tiene solo 1 forma de apariencia. Una etiqueta se usa para mostrar datos, un bloque de texto o como elemento decorativo [iRidium Studio, 2021c].
- Botón Trigger: El botón trigger es un elemento que se puede presionar y que se fija en uno de los dos estados: “apagado” (valor 1) y “encendido” (valor 2) en cada pulsación y envía estos valores al equipo. Cuando se reciben datos, un disparador cambia su estado a apagado o encendido si el valor recibido está de acuerdo con el Valor 1 o Valor 2 [iRidium Studio, 2021c].
- Arriba/Abajo: es un elemento que se puede presionar que tiene 2 estados: “sin presionar” y “presionado”; en él se comporta casi como un botón, pero envía valores al equipo que aumentan o disminuyen en comparación con el valor actual. Cuando se presiona el botón Arriba/Abajo toma el valor actual de las variables, lo cambia a un valor establecido y lo envía al equipo. Normalmente se lo usa para: control de temperatura +/- 1 grado, dimerización y tareas similares donde la variable debe cambiarse con un cierto paso [iRidium Studio, 2021c].
- Botón de estados múltiples: se diferencia del botón en tener numerosas formas respecto a su apariencia. Cuando se presiona, cambia un estado “no presionado” (primero) a “presionado” (último), y al hacerlo muestra todos sus estados internos. El proceso de moverse entre el primer y el último estado es la animación de cuadros. La visualización de estados no se puede detener en el medio (se usa el nivel Multiestado para ello). La configuración permite hacer una animación de bucle y se repite siempre que se envíe un valor distinto de cero al botón o mientras la interfaz funcione) [iRidium Studio, 2021c].
- Nivel: es un elemento que permite un control deslizante en movimiento. Tiene dos tipos de formas: “sin relleno” y “relleno”. Un control deslizante

cambia un estado por otro moviéndose a lo largo de la barra de un extremo a otro. Los valores máximos y mínimos entre los que se mueve un control deslizante se define para un nivel. La posición de un control deslizante determina un valor que se envía al equipo (cuando se mueve un control deslizante con el cursor o el dedo) o un valor que debe mostrarse cuando proviene del equipo. El movimiento de un control deslizante es lineal, para un movimiento circular hay un elemento de nivel circular [iRidium Studio, 2021c].

- Cuadro de edición: es una ventana donde se puede ingresar y editar texto desde el teclado. Un cuadro de edición tiene dos estados “no seleccionado” y “seleccionado” que se activan cuando un cursor se encuentra en el cuadro de edición. El texto ingresado se puede enviar presionando enter o mediante un botón separado que toma un valor del cuadro de edición y lo envía donde se requiere [iRidium Studio, 2021c].
- Joystick: es un nivel de dos coordenadas que genera ValueX y ValueY cuando un cursor se mueve en un área limitada por rangos de ejes X e Y. La característica peculiar importante de un joystick es la capacidad de obtener el valor actual del color debajo del cursor. Si se coloca una imagen con una paleta en un joystick, cuando se controla, devuelve el color debajo del cursor que se puede dividir en componentes RGB y enviar al atenuador RGB [iRidium Studio, 2021c].
- Lista estática: es un elemento para navegar en el proyecto que permite colocar numerosas ventanas emergentes en una página y moverse entre ellas con la ayuda de gestos. Una lista estática parece un elemento gráfico en una página o una ventana emergente. La configuración de una lista estática incluye una lista de ventanas emergentes utilizadas y su orden. Si una ventana emergente está en una lista estática, no se puede usar de forma independiente o en otra lista (límite técnico) [iRidium Studio, 2021c].
- Tendencia lineal: es una tendencia donde los puntos, los valores de la variable, están conectados con una línea. Una tendencia lineal permite mostrar el historial de cambios de varios objetos de valor en el rango de tiempo y valor establecido. Una tendencia lineal obtiene datos de la base de datos del servidor iRidium o de una base de datos SQL externa (en el caso de una base de datos externa, se requiere el siguiente script). i3 pro no puede almacenar su propia base de datos, por eso las tendencias solo están disponibles si se utiliza un servidor o una base

de datos externa. Algunos parámetros de una tendencia lineal solo se pueden editar mediante un script [iRidium Studio, 2021c].

### Pruebas y visualización del proyecto

En esta sección se muestra cómo realizar las pruebas de funcionamiento, como obtener las licencias para realizar estas pruebas y la configuración de la aplicación. Para ello se presiona el botón “Emulador” del software iRidium Studio y este permite lanzar nuestra interfaz creada en la aplicación i3 pro tal como se observa en la Figura 3.19 y Figura 3.20.



Figura 3.19: Simulador en aplicación i3 pro. Fuente: autor

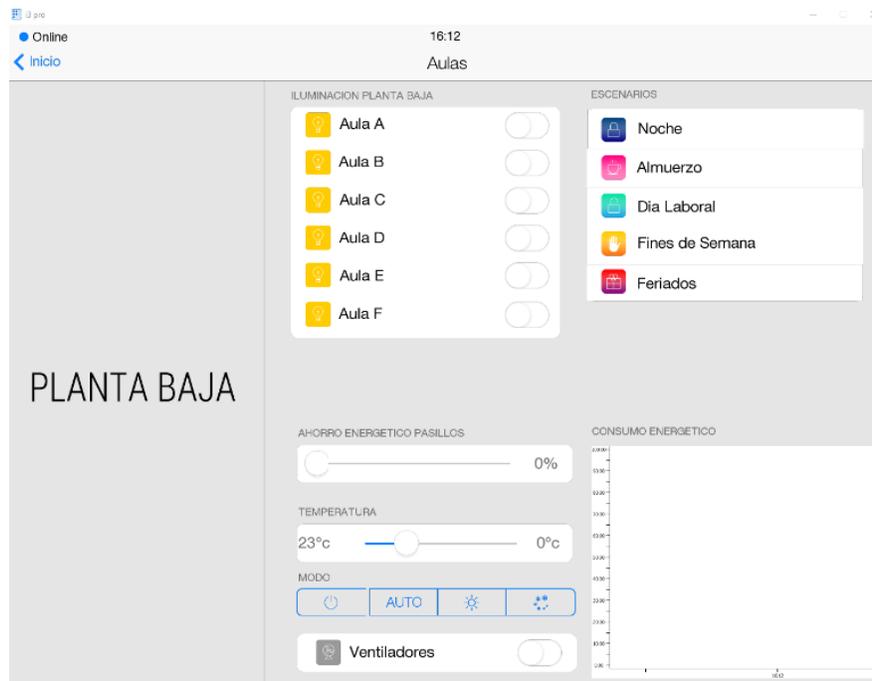


Figura 3.20: Aplicación i3 pro. Fuente: autor

Al empezar las pruebas vamos a recibir un mensaje en la aplicación

de que no se encuentra la licencia. Sin embargo, una vez ingresadas nuestras credenciales en la aplicación la licencia de prueba se actualizará automáticamente a los minutos de prueba de acuerdo con el programa de dealer que se mantenga con iRidium. Por ejemplo, si se mantiene una cuenta de estatus “Bronze” las pruebas están limitadas por tiempo, después de este tiempo la aplicación ya no controla ningún equipo y se tiene que volver a lanzar. La cuenta de estatus “Silver” y “Gold” está limitada por tiempo pero permite interactuar por mayor tiempo con los equipos. La licencia es responsable de permitir la interacción entre la aplicación y los equipos físicos. En nuestro proyecto trabajamos con la licencia Bronze [iRidium Studio, 2021c].

### **Desarrollo de la interfaz de usuario para aplicación i3 pro**

El proceso de desarrollo para la interfaz de usuario se desarrolla en scripts bajo lenguaje de programación javascript. En la figura 3.21 se presenta el programa para realizar el desplazamiento de una barra para atenuar la iluminación y para desplazar una barra y parametrizar un nivel de temperatura.

Así mismo se muestra el script con la programación en javascript. Este script permite

1. Parametrizar en el eje X y en el eje Y de la interfaz de usuario. Se lo usa para colocar mediante el desplazamiento una temperatura específica y así mismo para atenuar la iluminación a un porcentaje deseado. Este script se lo utiliza dentro de cada piso del edificio. Como programación se define las coordenadas de desplazamiento en el eje X y Y mediante el uso de fórmulas.
2. Se utiliza una función que permite activar eventos en el sistema ya sea al recibir datos de un equipo o al presionar un elemento, para ello se utilizan métodos en este caso usamos el método IR.AddListener. Dentro de este método se configuran tres parámetros que son: evento: el identificador del evento monitoreado por el oyente(listener), después tenemos entrada: el parámetro de entrada necesario para crear el oyente. Si no existe tal parámetro indicar 0, y finalmente acción: la función sin nombre o el enlace a la función nombrada en la que se realizan las acciones.

- Luego se accede a los elementos gráficos en las páginas/popups mediante método IR.GetItem.

```

25 Slider
26 Open_popups
27 Button_Back
28 ColorPicker
29 YahooWeatherWL...

```

```

function UserSlider(Level, Slider) {
    function Move() {
        Slider.X = Level.X + Level.Value * Level.Width / 100 - Slider.Width * Level.Value / 100;
    }
    IR.AddListener(IR.EVENT_START, 0, Move);
    IR.AddListener(IR.EVENT_ITEM_PRESS, Level, Move);
    IR.AddListener(IR.EVENT_MOUSE_MOVE, Level, Move);
    IR.AddListener(IR.EVENT_TOUCH_MOVE, Level, Move);
    IR.SetInterval(1000, Move);
}

function UserSlider_vert(Level, Slider) {
    function Move() {
        Slider.Y = Level.Y - Level.Height - Level.Value * Level.Height / 100 - (Slider.Height - Slider.Height * Level.Value / 100);
    }
    IR.AddListener(IR.EVENT_START, 0, Move);
    IR.AddListener(IR.EVENT_ITEM_PRESS, Level, Move);
    IR.AddListener(IR.EVENT_MOUSE_MOVE, Level, Move);
    IR.AddListener(IR.EVENT_TOUCH_MOVE, Level, Move);
    IR.SetInterval(1000, Move);
}

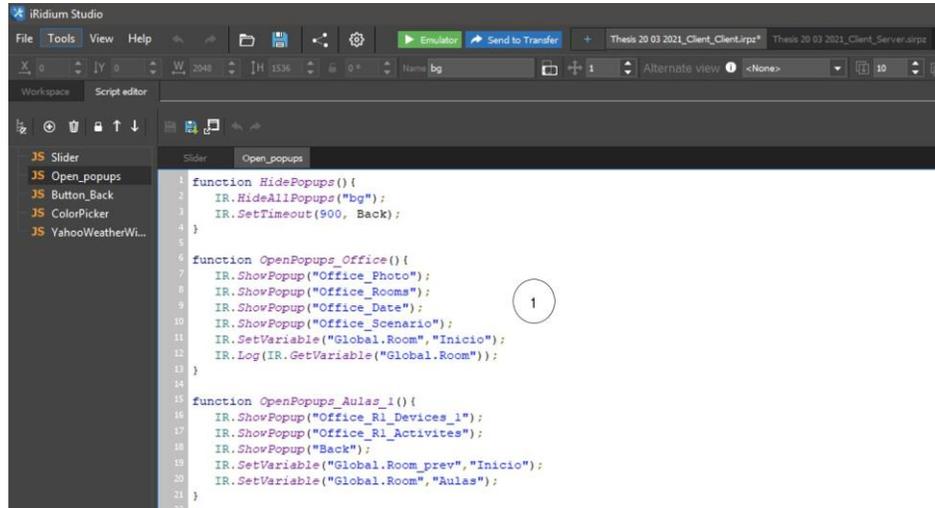
var UserSlider_1 = new UserSlider(IR.GetItem("Office_R1_Activites").GetItem("Slider1"), IR.GetItem("Office_R1_Activites").GetItem("Slider Dim..."));
var UserSlider_2 = new UserSlider(IR.GetItem("Office_R1_Activites").GetItem("Slider2"), IR.GetItem("Office_R1_Activites").GetItem("Slider Tr..."));
var UserSlider_3 = new UserSlider(IR.GetItem("Office_R2_Activites").GetItem("Slider1"), IR.GetItem("Office_R2_Activites").GetItem("Slider 1A..."));
var UserSlider_4 = new UserSlider(IR.GetItem("Office_R2_Activites").GetItem("Slider2"), IR.GetItem("Office_R2_Activites").GetItem("Slider 2A..."));
var UserSlider_5 = new UserSlider(IR.GetItem("Office_R3_Activites").GetItem("Slider1"), IR.GetItem("Office_R3_Activites").GetItem("Slider Temperature"));

```

Figura 3.21: Script para niveles de desplazamiento. Fuente: autor

Para trasladarse por la aplicación y abrir popups, realizar cambios de páginas se utiliza el siguiente script y la API iRidium. Para ello se genera el siguiente script de la Figura 3.22. Para este script se utilizan funciones, métodos y propiedades de los objetos utilizados para controlar ítems, páginas y popups en en el proyecto.

- Para ello utilizamos los métodos que se representan mediante los siguientes parámetros: objeto IR: el cual es un objeto global que contienen funciones y constantes de la API de iRidium. Los métodos del objeto IR permiten hacer referencia a páginas y popups de interfaces de visualización. De igual forma se puede ejecutar comandos relacionados con el mismo trabajo del sistema operativo. La cadena de comandos aplicada al objeto IR proporciona acceso a cualquier objeto de interfaz de usuario y permite hacer referencia a propiedades. La jerarquía de objetos: IR>Página/Popup>Ítem>Estado.



The screenshot shows the iRidium Studio interface with a script editor open. The script editor contains the following JavaScript code:

```
1 function HidePopups() {
2   IR.HideAllPopups("bg");
3   IR.SetTimeout(900, Back);
4 }
5
6 function OpenPopups_Office() {
7   IR.ShowPopup("Office_Photo");
8   IR.ShowPopup("Office_Rooms");
9   IR.ShowPopup("Office_Date");
10  IR.ShowPopup("Office_Scenario");
11  IR.SetVariable("Global.Room", "Inicio");
12  IR.Log(IR.GetVariable("Global.Room"));
13 }
14
15 function OpenPopups_Aulas_1() {
16  IR.ShowPopup("Office_Rl_Devices_1");
17  IR.ShowPopup("Office_Rl_Activites");
18  IR.ShowPopup("Back");
19  IR.SetVariable("Global.Room_prev", "Inicio");
20  IR.SetVariable("Global.Room", "Aulas");
21 }
22 }
```

A circled number '1' is placed next to line 11, highlighting the `IR.SetVariable` call.

Figura 3.22: Script pop-up para interfaz i3 pro. Fuente: autor

Para regresar a la página principal de la interfaz de usuario se utiliza el script de la Figura 3.23. Para esta programación se utilizó la el API de iRidium y se basa en el control de tags y se usan los métodos `paraleer` y `escribir` etiquetas de diferentes objetos en el proyecto. Ellos pueden ser información del sistema, información del driver, retroalimentación de los drivers, propiedades de la interfaz de usuario.

1. Con la ayuda del método `IR.GetVariable`, se hace referencia a las propiedades de la interfaz de usuario para tener un ítem de regresar cuando estemos dentro de una página. Es necesario que para referenciar una tag es necesario incluir el nombre completo en la solicitud

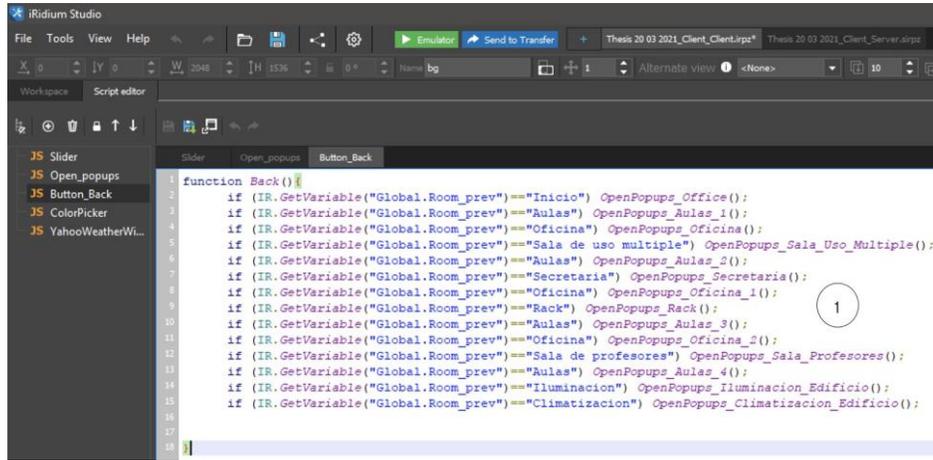


Figura 3.23: Script botón back para interfaz i3 pro. Fuente: autor

En esta sección se presenta la programación que utilizamos para la interacción del usuario con la interfaz. Sin embargo el software también permite asignar comandos a los botones, y se lo realiza mediante arrastrar/soltar con el uso del mouse, de acuerdo con la Figura 3.24. De esta manera se usan los drivers de la columna derecha y se asigna el comando a controlar y se requiere arrastrar/soltar al botón que se desea controlar.



Figura 3.24: Control de comandos en interfaz i3 pro. Fuente: autor

### 3.2.5. iRidium Server

iRidium Server es un software y hardware para sistemas de automatización e IoT, que funciona en un servidor y proporciona la conexión con equipos de automatización. Se usa para las siguientes tareas:

- Crear rutinas y horarios.
- Establecer la lógica de la interacción del equipo.
- Exportación e importación de bases de datos para utilizar esta información para identificar comportamientos del edificio.
- Intercambio de datos entre diferentes protocolos.
- Notificaciones push que permite enviar desde el servidor notificaciones a la interfaz de usuario informando de algún evento inesperado que se suscita.
- Gráficas que permiten visualizaciones de una forma más amigable con el usuario.
- Herramientas para crear drivers para cualquier equipo.
- Javascript para programación.
- Interfaz WEB.
- iRidium Cloud para almacenar interfaz de proyectos, proyectos con servidor, licencias.

Las ventajas de ocupar un iRidium Server:

- El servidor une los paneles de control.
- El servidor proporciona una carga de trabajo mínima del equipo.
- El servidor lanza escenas en cualquier momento.
- El servidor proporciona un intercambio de datos seguro con paneles de control.
- El servidor se puede utilizar como controlador programable.
- El servidor se puede utilizar como puerta de enlace para el intercambio de datos.

- El servidor funciona localmente sin acceso a Internet.
- El servidor admite la mayoría de los protocolos existentes.
- El servidor ayuda a convertir un objeto de automatización en parte de la estructura global de IoT [iRidium Studio, 2021b].

### Componentes y comunicación de un proyecto servidor

Se presenta los componentes del servidor, control del servidor, entradas/salidas comunicación de equipos. La Figura 3.25 muestra la comunicación entre sistemas y se observa que el servidor iRidium es el puente entre los equipos de automatización y la interfaz de usuario.

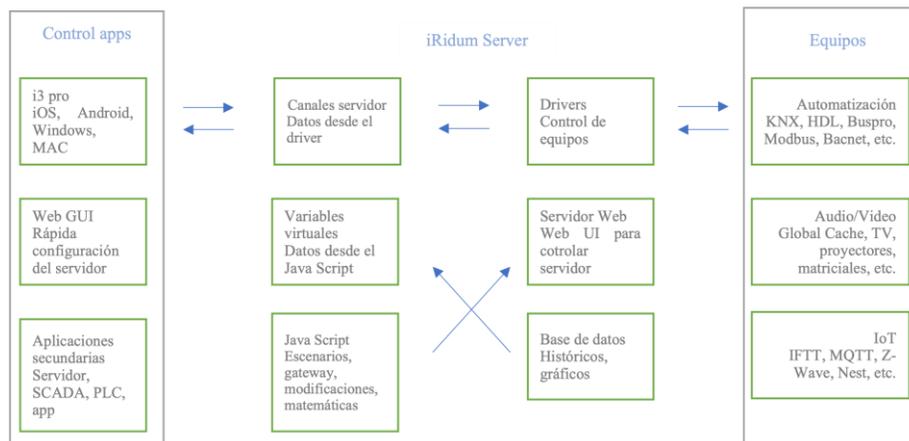


Figura 3.25: Comunicación entre servidor, equipos y paneles. Fuente: autor

### Componentes de un servidor

- Drivers: están conectados al equipo y almacenan la lista de variables que se pueden controlar. Los drivers están relacionados con los canales del servidor.
- Canales del servidor: las variables intermedias entre los paneles de control y los canales de los drivers. Los canales del servidor unen variables de todos los drivers en un espacio.

- Variables virtuales: almacenan datos del servidor y resultados del trabajo de JavaScript que no están relacionados con las variables reales del driver.
- Base de datos: mantiene un registro de los cambios de las variables del servidor.
- Javascript: describe escenas, lógica y horarios (características inteligentes del servidor). Permite la modificación de datos cuando pasan por el servidor.
- Servidor web: permite la conexión del servidor iRidium a través de la interfaz web para ver el estado y la configuración. Permite el trabajo de la API REST con la ayuda de las cuales aplicaciones y sistemas de terceros pueden controlar las funciones del servidor a través de http [iRidium Studio, 2021b].

### Control del servidor

- i3 pro: la interfaz de usuario iRidium que inicia la visualización del control del servidor. Incluye:
  - El driver del servidor iRidium en i3 pro: este driver establece la conexión con el servidor a través de TCP.
  - JavaScript en i3 pro: permite la creación de escenas que funcionan en paneles de control.
  - La interfaz gráfica en i3 pro: la interfaz visual para controlar equipos de automatización.
- Interfaz web: es la interfaz para controlar la configuración del servidor que se puede abrir en el navegador. No muestra la visualización del usuario.
- Aplicaciones de terceros: las aplicaciones y dispositivos que pueden controlar el servidor mediante el envío de comandos http configurados de acuerdo con la API REST del servidor. [iRidium Studio, 2021b]

### Entradas/Salidas. Comunicación con equipamiento

Para el uso del servidor es necesaria la conexión al equipo desde paneles de control. En el caso simple de uso del servidor, es necesario conectarse al equipo desde varios paneles de control. Se configura la transferencia de

datos por medio del servidor el cual permite el funcionamiento de varios paneles de control.

De esta forma se realizan las siguientes acciones en un proyecto servidor [iRidium Studio, 2021b]:

- Se crea un proyecto de servidor.
- Se agrega la lista de drivers que desea controlar en ella.
- Se configura comandos y canales de retroalimentación para los drivers.
- Se crea un proyecto para i3 pro sobre la base del proyecto del servidor.

### Crear proyecto servidor en iRidium Studio

Se crea un proyecto servidor al presionar en File, New y posteriormente New server project para ello ver la Figura 3.26. Para cargar los drivers se pueden realizar de tres formas, la primera es desde el panel de dispositivos presionar agregar driver y escoger el driver correspondiente. La segunda es mediante la importación de archivos que permite iRidium y la tercera es mediante un escaneo de dispositivos en la red local la cual es la opción que se trabaja en este proyecto.

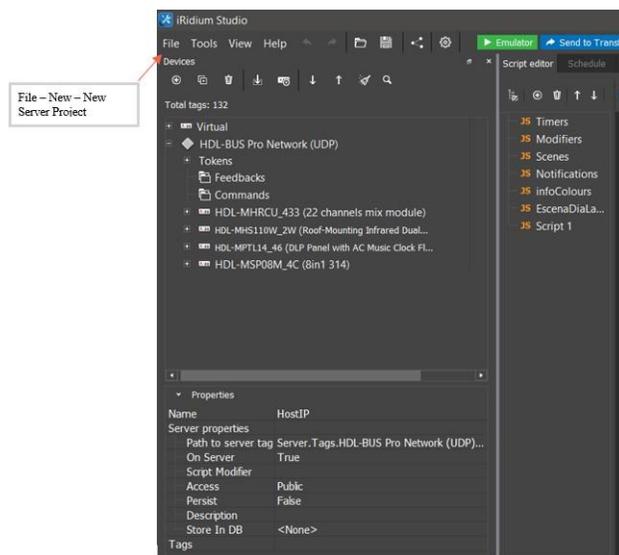


Figura 3.26: Crear proyecto servidor. Fuente: autor

### Drivers y variables del servidor

La información de entradas/salidas del proyecto servidor se dividen en dos partes: drivers y servidor iRidium.

- Drivers: en los drivers se encuentra las configuraciones para la conexión de los equipos de automatización.
- Servidor Iridium: se encuentra las relaciones entre las variables de los equipos de automatización y las variables del servidor. De esta forma los paneles de control se refieren con las variables del servidor para después enviarse la información a los equipos.[iRidium Studio, 2021b]

La proyección del driver en las variables de iRidium Server consta de las siguientes pestañas:

Tokens del driver: las variables del driver. Muestran el estado de conexión al equipo. Esto se puede ver en la Figura 3.27.

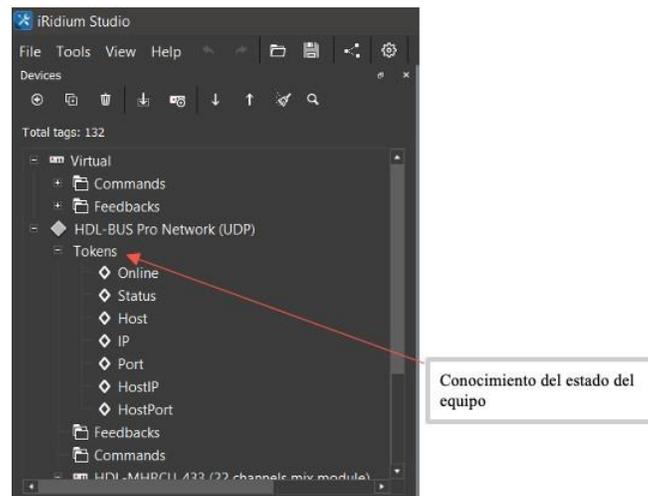


Figura 3.27: Tokens del driver. Fuente: autor

Canales: las variables para el control. Esto se observa en la Figura 3.28.

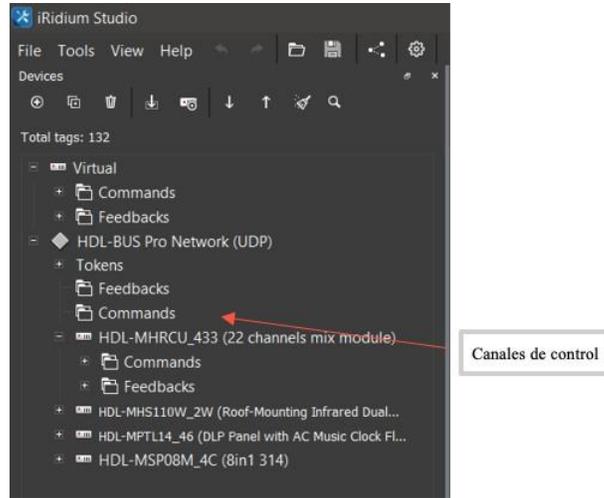


Figura 3.28: Comandos de los equipos. Fuente: autor

Tags: las variables de las que se puede recibir retroalimentación. Para ello ver la Figura 3.29

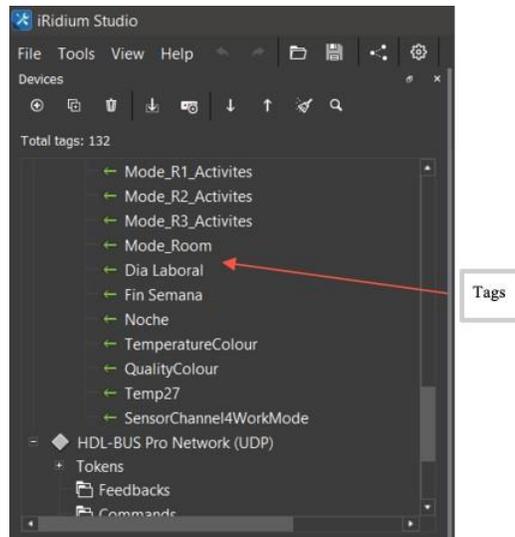


Figura 3.29: Tags y retroalimentación. Fuente: autor

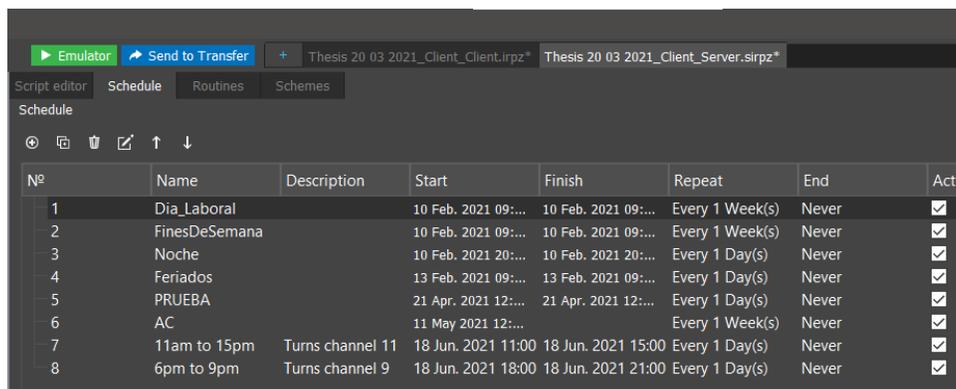
Al escribir un valor en el canal del servidor, activamos el comando al equipo. Lo mismo funciona para la retroalimentación: se enviarán desde la retroalimentación del driver al tag del servidor correspondiente.

Las variables del servidor se pueden escribir, modificar o leer con la ayuda del editor javascript. Esta modificación significa que hay una transformación de los datos en el servidor antes de enviarse los comandos al equipos o antes de enviar retroalimentación a los paneles de control.[iRidium Studio, 2021b]

## Horarios

Para programar los horarios para la creación de ambientes se lo realiza en el proyecto servidor en la pestaña "Schedule". Para ello el proyecto servidor incluye un programador que trabaja como un listado de eventos. Los horarios se pueden definir para una sola aplicación o se puede hacer en intervalos específicos.

Esta sección permite trabajar conjuntamente con las rutinas y se puede repetir en intervalos desde un día a años. En la Figura 3.30 se observa los horarios creados para el proyecto y la Figura 3.31 es la ventana que permite parametrizar el horario.



Nº	Name	Description	Start	Finish	Repeat	End	Acti
1	Dia_Laboral		10 Feb. 2021 09:...	10 Feb. 2021 09:...	Every 1 Week(s)	Never	<input checked="" type="checkbox"/>
2	FinesDeSemana		10 Feb. 2021 09:...	10 Feb. 2021 09:...	Every 1 Week(s)	Never	<input checked="" type="checkbox"/>
3	Noche		10 Feb. 2021 20:...	10 Feb. 2021 20:...	Every 1 Day(s)	Never	<input checked="" type="checkbox"/>
4	Ferriados		13 Feb. 2021 09:...	13 Feb. 2021 09:...	Every 1 Day(s)	Never	<input checked="" type="checkbox"/>
5	PRUEBA		21 Apr. 2021 12:...	21 Apr. 2021 12:...	Every 1 Day(s)	Never	<input checked="" type="checkbox"/>
6	AC		11 May 2021 12:...		Every 1 Week(s)	Never	<input checked="" type="checkbox"/>
7	11am to 15pm	Turns channel 11	18 Jun. 2021 11:00	18 Jun. 2021 15:00	Every 1 Day(s)	Never	<input checked="" type="checkbox"/>
8	6pm to 9pm	Turns channel 9	18 Jun. 2021 18:00	18 Jun. 2021 21:00	Every 1 Day(s)	Never	<input checked="" type="checkbox"/>

Figura 3.30: Horarios proyecto. Fuente: autor

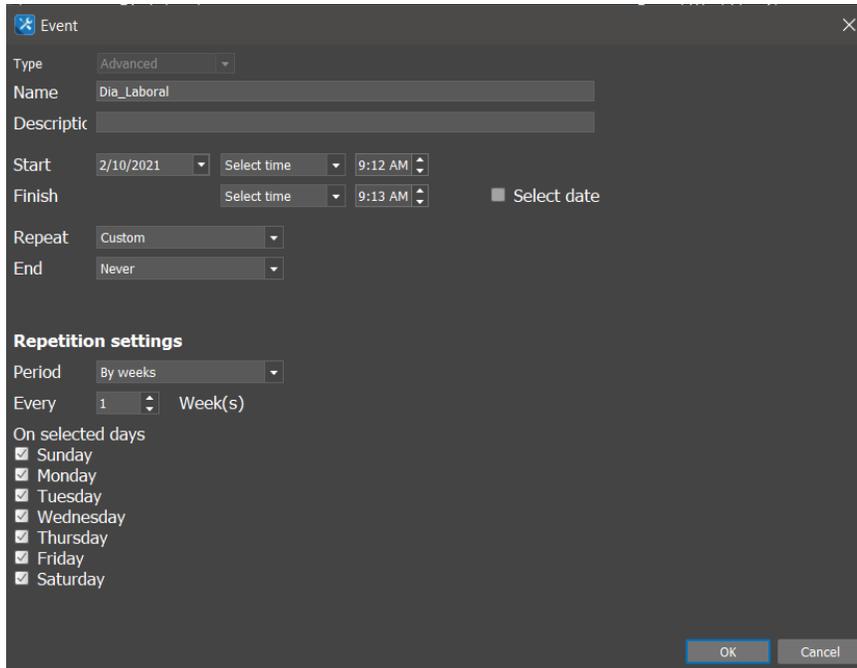


Figura 3.31: Eventos. Fuente: autor

El software de iRidium tiene un temporizador astronómico que está destinado para crear horarios y rutinas, y que están ligadas a iniciosde horarios como amanecer y atardecer. De esta forma permite encender luminarias al atardecer o cerrar cortinas al amanecer.

### Rutinas

Para crear rutinas se requiere tener previamente la programación de los horarios. La rutina es una escena que se ejecuta automáticamente cuando ocurre un evento específico. El evento puede ser la recepción de datos del equipo o el evento del horario del servidor. Para esta sección se presiona en la pestaña "Routines". En esta parte de la programación se define que, cuando se inicie un evento entonces se ejecuta una acción. En la Figura 3.32 se observa la programación de una rutina.

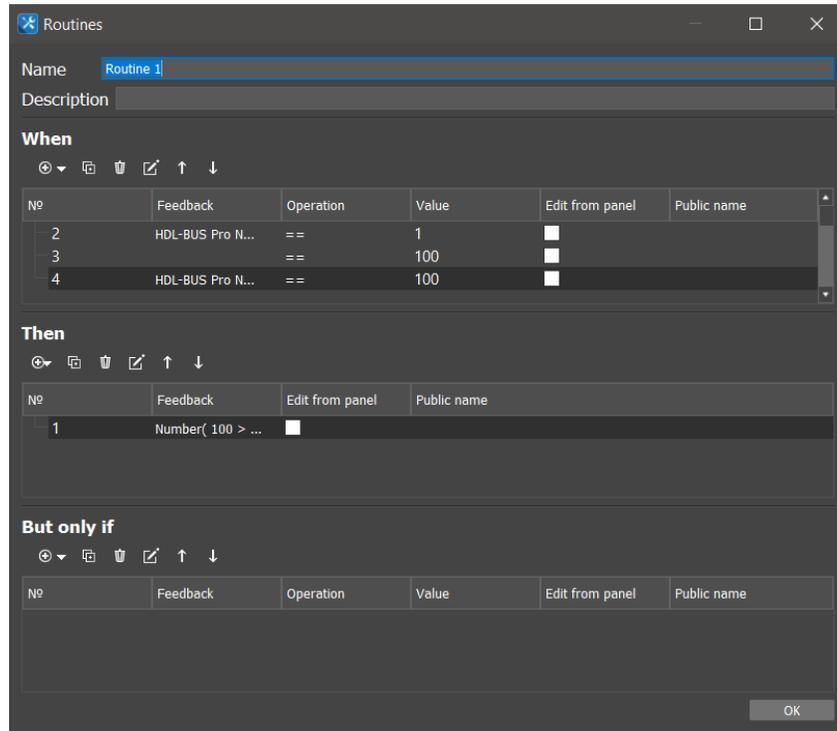


Figura 3.32: Rutinas del proyecto. Fuente: autor

### Modificación de datos

Para la modificación de datos recibidos antes de enviarlos según lo previsto ya sea en equipos de automatización o en paneles de control. Esta modificación se realiza con la ayuda de la función de modificación del script, la cual se configura en las propiedades de una canal o una tag del servidor (las variables virtuales también se modifican sobre la misma base). Si dicha función modifica por ejemplo un canal del servidor entonces el valor recibido desde un panel de control se analiza en el script y luego se envía al driver. O si la función modifica una tag del servidor entonces el valor recibido por la tag de un driver se analiza en el script y luego se envía a los paneles de control. Esto se puede ver en la Figura 3.33.

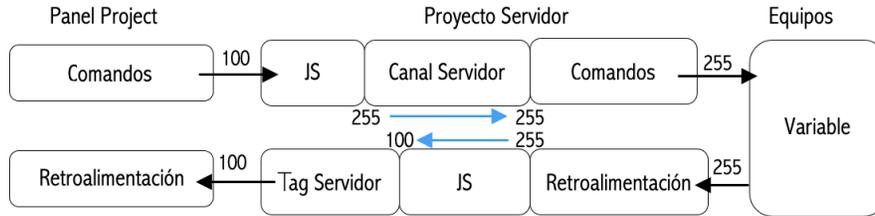


Figura 3.33: Modificación de datos en el servidor. Fuente: autor

```

1. function modify_send (in_Type, in_Name, in_Value)
2. {
3.   in_Value *= 255/100;
4.   return in_Value;
5. }
6. function modify_receive (in_Type, in_Name, in_Value)
7. {
8.   in_Value *= 100/255;
9.   return in_Value;
10. }

```

El canal o tag que activa la función le envía los argumentos:

- in\_Type: el tipo de variable, 0 - el canal, 1 - la etiqueta de retroalimentación
- in\_Name: el nombre completo del comando o tag
- in\_Value: el valor del comando o tag

De esta forma realizamos el tratamiento de los datos cuando pasan por el servidor en dos direcciones (desde los paneles de control hasta los equipos y viceversa). El valor modificado (in\_Value) se devuelve a la variable con la ayuda del comando `return in_Value;` no es necesario devolver el valor modificado; la acción solo se puede realizar una vez. Entonces, el valor

inicial recibido por la función no será recibido por el destino pero conducirá a la ejecución de la función.

Los valores modificados de los tags y los canales se muestran en la interfaz web del servidor, es decir: los valores de las tags aparecen en la interfaz web del servidor como resultado del trabajo de la función de modificación de secuencia de comandos, los valores, establecidos en la interfaz web mediante el comando "Set", se envían a la entrada de la función de modificación de secuencia de comandos antes de enviarse a un canal de un dispositivo físico [iRidium Studio, 2021b].

Otro aspecto importante es poder modificar los canales y tags de los drivers. Los canales y tags del driver están siempre relacionados con las variables del servidor y se pueden modificar. Al ejecutar un comando, tiene que enviar un valor (a veces el valor no es necesario, luego se envía una cadena vacía). Este valor se modifica antes de enviarlo al equipo:

Comandos o canales del driver:

1. function rangeModify (in\_Type, in\_Name, in\_Value)
2. {
3. in\_Value \*= 255/100;
4. return in\_Value.toFixed(0);
5. }

Tags del driver:

1. function rangeModifyBack (in\_Type, in\_Name, in\_Value)
2. {
3. in\_Value \*= 100/255;
4. return in\_Value.toFixed(0);
5. }

### Lectura y escritura de variables

Las variables de los servidores tales como: driver token, canales, tags, canales virtuales y tags virtuales. Sus estados pueden cambiar y solicitar tanto desde los paneles de control como desde los scripts en el servidor.

Es así que la lectura y escritura se realizan con los siguientes comandos en javascript:

`IR.GetVariable(name)`, `IR.SetVariable(name,value)`

Para paneles de control las variables del servidor se parecen a los comandos del driver y canales.

- Escritura de valores en variables:
  1. Copiar todo el nombre de la variable desde la pestaña del servidor de iRidium.
  2. Usar el comando `IR.SetVariable` para escribir el valor en la variable.

la forma queda representada de la siguiente manera: `IR.SetVariable("Server.Trayecto.Name", value)`, como lo muestra la Tabla 3.4.

Tabla 3.4: Arquitectura

<b>Entrada</b>	<b>Valor</b>	<b>Descripción</b>
Server.Trayecto.Name	Server.Channels.Virtual 1	Nombre de la variable del servidor
Valor	100	El valor de escritura de la variable del servidor
<b>Salida</b>		
Correcto	verdadero	Si la escritura es verdadera

1. `IR.AddListener(IR.EVENT_START, 0, function()`
2. {

3. IR.SetVariable("Server.Channels.Virtual 1", 100);
4. });

- Lectura de valores en variables:

1. Copiar todo el nombre de la variable desde la pestaña del servidor de iRidium.
2. Usar el comando IR.GetVariable para escribir el valor en la variable.

La forma queda representada de la siguiente manera: IR.GetVariable("Server.Trayecto.Name"), como lo muestra la Tabla 3.5.

Tabla 3.5: Arquitectura

<b>Entrada</b>	<b>Valor</b>	<b>Descripción</b>
Server.Trayecto.Name	Server.Channels.Virtual 1	Nombre de la variable del servidor
<b>Salida</b>		
Valor	100	El valor en la variable del servidor

1. IR.AddListener(IR.EVENT\_START, 0, function()
2. {
3. IR.GetVariable("Server.Channels.Virtual 1", 100);
4. });

El método puede funcionar en los canales y tags del servidor.

### **Función de puerta de enlace para intercambio de datos entre canal virtual y tag**

El comando virtual Virtual Channel (Server.Channels.Virtual 1) no se encuentra asociado con la tag de retroalimentación virtual (Server.Tags.Virtual 1), por lo que esta cambia el valor de la variable virtual en el servidor.

Para devolver el valor enviado desde i3 pro al servidor y a i3 pro, se necesita crear una función script que modifique la función de secuencia de comandos. Esta función escribirá el valor enviado desde i3 pro en la tag virtual de retroalimentación. Ese es el cambio que se verá en el panel de control y esto se refleja en la Figura 3.34.

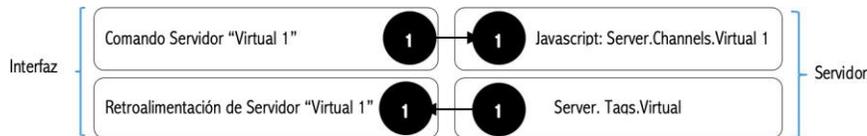


Figura 3.34: Intercambio de datos. Fuente: autor

Para configurar la transferencia de datos del comando virtual a la tag:

1. Crear un canal virtual (comando) y una tag (canal de retroalimentación) con el mismo nombre. Por ejemplo, "Virtual1".
2. Abra el editor de secuencia de comandos la pestaña de JS en la barra de herramientas y se genera un nuevo script. Agregar la siguiente función en él:
  - a) `function virtual_t_virtual (in_Type, in_Name, in_Value)`
  - b) `{`
  - c) `var namepart = in_Name.split(".");`
  - d) `) IR.SetVariable ("Server.Tags.- In_Name, in_Value);`
  - e) `return in_Value;`
  - f) `}`
3. Regrese al comando virtual "Virtual 1"(Server.Channels.Virtual 1), seleccione la función `virtual_to_virtual` en la lista desplegable del parámetro Script Modifier.
4. Cuando se escribe un valor en `Server.Channels.Virtual 1`, este valor se envía automáticamente a `Server.Tags.Virtual 1` y se puede mostrar como un canal de retroalimentación en el panel de control.

### **Función de puerta de enlace para intercambio de datos entre drivers**

Una ventaja del software que se utiliza en este proyecto es la capacidad de este en transferir datos entre equipos o sistemas de automatización. Las funciones de modificación en el script permiten esta ejecución.

Es decir, por ejemplo, al cambiar un registro de un driver bajo comunicación Modbus, enviar un nuevo valor a una dirección en KNX:

```

1. function gate_to_modbus (in_Type, in_Name, in_Value) 2.
{
3. IR.SetVariable("Server.Channels.Modbus TCP 2.Channel 12",
in_Value*100);
4. return in_Value;
5. }

```

La función monitorea los cambios del canal de retroalimentación en el driver Modbus. Selecciona este en el script de la tag del servidor Modbus. Si se da el mismo nombre a las variables que queremos conectar, entonces es suficiente para crear una función que modifica y enlaza con todas las retroalimentaciones que se pueden transferir a otro driver:

```

1. function gate_to_modbus (in_Type, in_Name, in_Value) 2.
{
3. var namepart = in_Name.split(".");
4. IR.SetVariable("Server.Channels.Modbus TCP 2.- namepart[1],
in_Value);
5. return in_Value;
6. }

```

### **Escenas**

La mejor forma para obtener un sistema de automatización es al crear ambientes en horarios pre establecidos. Y es fundamental ejecutarle con scripts. Es así que una escena se activa cuando se envía un valor desde el

panel de control al canal virtual del servidor.

Se muestra en la figura 3.35 y figura 3.36 la programación que se realiza en el proyecto de desarrollo en cuanto a crear ambientes en diferentes horarios y espacios de acuerdo con la necesidad del edificio. Para este script se trabaja sobre la API de iRidium en el sistema.

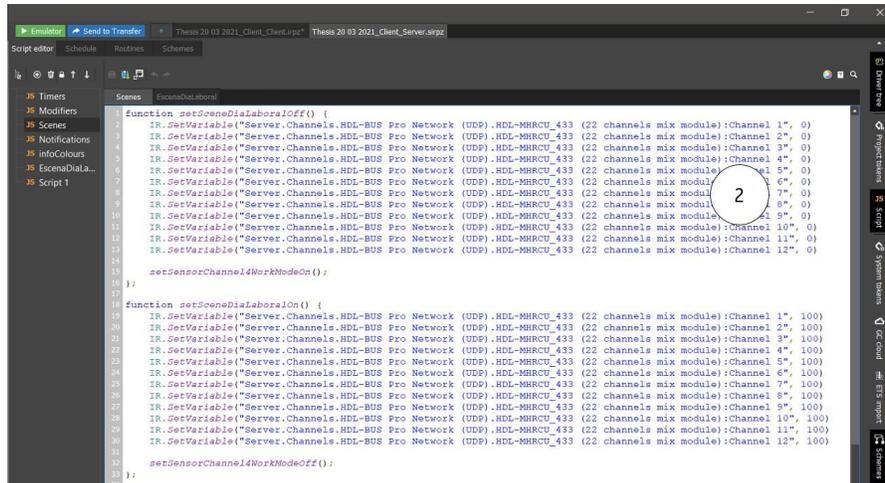
1. Se requiere nuevamente el uso de métodos y para ello se utiliza el método `IR.log(data)` el cual su entrada se la puede describir con los siguientes tipos: objeto, string, número, matriz de datos para la salida en el registro, y su salida es de tipo string. Es así que los datos enviados a la consola se convierten en string. Mediante este método se obtienen los datos de salida en la consola de la interfaz de usuario.
2. Y para la activación de comandos se utiliza la ayuda del método `IR.SetVariable` que se enfoca en la misma definición que dimos para el método `IR.GetVariable`. En este script este método nos ayuda a identificar el estado de canal de retroalimentación del driver utilizado. El método puede ser usado para leer y escribir, sin embargo, el uso en este script es para escribir el valor de 0 o 100 y depende de la escena que cumpla con la condición.

```

1 IR.AddListener(IR.EVENT_START, 0, function() {
2   //Launch a listener if the main is global tag work.
3   IR.SetGlobalListener(IR.EVENT_GLOBAL_TAG_CHANGE, function(name, value) {
4     IR.Log("Active global: " + name + " | Value: " + value);
5     if(name == "Server.Channels.Dia Laboral" && value == 0) {
6       IR.Log("Dia Laboral OFF")
7       setSceneDiaLaboralOff();
8     } else if(name == "Server.Channels.Dia Laboral" && value == 100) {
9       IR.Log("Dia Laboral ON")
10      setSceneDiaLaboralOn();
11     } else if(name == "Server.Channels.Noche" && value == 0) {
12       IR.Log("Noche ON")
13       setSceneNocheOff();
14     } else if(name == "Server.Channels.Noche" && value == 100) {
15       IR.Log("Noche OFF")
16       setSceneNocheOn();
17     } else if(name == "Server.Channels.Fin Semana" && value == 0) {
18       IR.Log("Light OFF")
19       setSceneFinSemanaOff();
20     } else if(name == "Server.Channels.Fin Semana" && value == 100) {
21       IR.Log("Light ON")
22       setSceneFinSemanaOn();
23     } else if(name == "Drivers.HDL-BUS Pro Network (UDP).HDL-MPROM_4C (6in1 314) Input 1 Temperature") {
24       IR.Log("Changed temperature on HDL-MPROM_4C to: " + value);
25       //this feedback can be used on panel as feedback "note" (when feedback dragged on item) to setup on any colour
26       infoPanelTemperature("Server.Tags.TemperatureColour", value);
27       if(value > 26)
28         turnOnAirConditioner();
29       else if(value < 23)
30         turnOffAirConditioner();
31     } else if(name == "Drivers.HDL-BUS Pro Network (UDP).HDL-MH110W_2W (Roof-Mounting Infrared Dual-Technology Motion
32     IR.Log("Changed MH110W_2W movement sensor value to: " + value);
33     if(value > 0)
34       setSensorTimeout();
35     } else if(name == "Drivers.HDL-BUS Pro Network (UDP).HDL-MPROM_4C (6in1 314) Input 3 PIR") {
36       IR.Log("Changed MPROM_4C movement sensor value to: " + value);
37       if(sensorWorkingNow() {
38         if(value == 0) {
39           turnOnLights();
40         } else {
41           turnOffLights();
42         }
43       }
44     }
45   }
46 }

```

Figura 3.35: Escenas. Fuente: autor



```

Script editor | Schedule | Routines | Schemes
Thesis 20 03 2021_Client_Client.jspz | Thesis 20 03 2021_Client_Server.stpz

Scenes | EscenaDialLaboral

35 Timers
35 Modifiers
35 Scenes
35 Notifications
35 InfoColours
35 EscenaDial...
35 Script 1

function setSceneDialLaboralOff() {
  1 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 1", 0)
  2 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 2", 0)
  3 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 3", 0)
  4 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 4", 0)
  5 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 5", 0)
  6 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 6", 0)
  7 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 7", 0)
  8 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 8", 0)
  9 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 9", 0)
  10 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 10", 0)
  11 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 11", 0)
  12 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 12", 0)
  13
  14 setSensorChannel4WorkModeOn();
  15
  16 }
  17
  18 function setSceneDialLaboralOn() {
  19 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 1", 100)
  20 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 2", 100)
  21 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 3", 100)
  22 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 4", 100)
  23 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 5", 100)
  24 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 6", 100)
  25 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 7", 100)
  26 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 8", 100)
  27 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 9", 100)
  28 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 10", 100)
  29 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 11", 100)
  30 IR.SetVariable("Server.Channels.HDL-BUS Pro Network (UDP).HDL-MHRCU_433 (22 channels mix module):Channel 12", 100)
  31
  32 setSensorChannel4WorkModeOff();
  33
  34 }
  35
  36 }
  37
  38 }
  39
  40 }
  41
  42 }
  43
  44 }
  45
  46 }
  47
  48 }
  49
  50 }
  51
  52 }
  53
  54 }
  55
  56 }
  57
  58 }
  59
  60 }
  61
  62 }
  63
  64 }
  65
  66 }
  67
  68 }
  69
  70 }
  71
  72 }
  73
  74 }
  75
  76 }
  77
  78 }
  79
  80 }
  81
  82 }
  83
  84 }
  85
  86 }
  87
  88 }
  89
  90 }
  91
  92 }
  93
  94 }
  95
  96 }
  97
  98 }
  99
  100 }
  101
  102 }
  103
  104 }
  105
  106 }
  107
  108 }
  109
  110 }
  111
  112 }
  113
  114 }
  115
  116 }
  117
  118 }
  119
  120 }
  121
  122 }
  123
  124 }
  125
  126 }
  127
  128 }
  129
  130 }
  131
  132 }
  133
  134 }
  135
  136 }
  137
  138 }
  139
  140 }
  141
  142 }
  143
  144 }
  145
  146 }
  147
  148 }
  149
  150 }
  151
  152 }
  153
  154 }
  155
  156 }
  157
  158 }
  159
  160 }
  161
  162 }
  163
  164 }
  165
  166 }
  167
  168 }
  169
  170 }
  171
  172 }
  173
  174 }
  175
  176 }
  177
  178 }
  179
  180 }
  181
  182 }
  183
  184 }
  185
  186 }
  187
  188 }
  189
  190 }
  191
  192 }
  193
  194 }
  195
  196 }
  197
  198 }
  199
  200 }
  201
  202 }
  203
  204 }
  205
  206 }
  207
  208 }
  209
  210 }
  211
  212 }
  213
  214 }
  215
  216 }
  217
  218 }
  219
  220 }
  221
  222 }
  223
  224 }
  225
  226 }
  227
  228 }
  229
  230 }
  231
  232 }
  233
  234 }
  235
  236 }
  237
  238 }
  239
  240 }
  241
  242 }
  243
  244 }
  245
  246 }
  247
  248 }
  249
  250 }
  251
  252 }
  253
  254 }
  255
  256 }
  257
  258 }
  259
  260 }
  261
  262 }
  263
  264 }
  265
  266 }
  267
  268 }
  269
  270 }
  271
  272 }
  273
  274 }
  275
  276 }
  277
  278 }
  279
  280 }
  281
  282 }
  283
  284 }
  285
  286 }
  287
  288 }
  289
  290 }
  291
  292 }
  293
  294 }
  295
  296 }
  297
  298 }
  299
  300 }
  301
  302 }
  303
  304 }
  305
  306 }
  307
  308 }
  309
  310 }
  311
  312 }
  313
  314 }
  315
  316 }
  317
  318 }
  319
  320 }
  321
  322 }
  323
  324 }
  325
  326 }
  327
  328 }
  329
  330 }
  331
  332 }
  333
  334 }
  335
  336 }
  337
  338 }
  339
  340 }
  341
  342 }
  343
  344 }
  345
  346 }
  347
  348 }
  349
  350 }
  351
  352 }
  353
  354 }
  355
  356 }
  357
  358 }
  359
  360 }
  361
  362 }
  363
  364 }
  365
  366 }
  367
  368 }
  369
  370 }
  371
  372 }
  373
  374 }
  375
  376 }
  377
  378 }
  379
  380 }
  381
  382 }
  383
  384 }
  385
  386 }
  387
  388 }
  389
  390 }
  391
  392 }
  393
  394 }
  395
  396 }
  397
  398 }
  399
  400 }
  401
  402 }
  403
  404 }
  405
  406 }
  407
  408 }
  409
  410 }
  411
  412 }
  413
  414 }
  415
  416 }
  417
  418 }
  419
  420 }
  421
  422 }
  423
  424 }
  425
  426 }
  427
  428 }
  429
  430 }
  431
  432 }
  433
  434 }
  435
  436 }
  437
  438 }
  439
  440 }
  441
  442 }
  443
  444 }
  445
  446 }
  447
  448 }
  449
  450 }
  451
  452 }
  453
  454 }
  455
  456 }
  457
  458 }
  459
  460 }
  461
  462 }
  463
  464 }
  465
  466 }
  467
  468 }
  469
  470 }
  471
  472 }
  473
  474 }
  475
  476 }
  477
  478 }
  479
  480 }
  481
  482 }
  483
  484 }
  485
  486 }
  487
  488 }
  489
  490 }
  491
  492 }
  493
  494 }
  495
  496 }
  497
  498 }
  499
  500 }
  501
  502 }
  503
  504 }
  505
  506 }
  507
  508 }
  509
  510 }
  511
  512 }
  513
  514 }
  515
  516 }
  517
  518 }
  519
  520 }
  521
  522 }
  523
  524 }
  525
  526 }
  527
  528 }
  529
  530 }
  531
  532 }
  533
  534 }
  535
  536 }
  537
  538 }
  539
  540 }
  541
  542 }
  543
  544 }
  545
  546 }
  547
  548 }
  549
  550 }
  551
  552 }
  553
  554 }
  555
  556 }
  557
  558 }
  559
  560 }
  561
  562 }
  563
  564 }
  565
  566 }
  567
  568 }
  569
  570 }
  571
  572 }
  573
  574 }
  575
  576 }
  577
  578 }
  579
  580 }
  581
  582 }
  583
  584 }
  585
  586 }
  587
  588 }
  589
  590 }
  591
  592 }
  593
  594 }
  595
  596 }
  597
  598 }
  599
  600 }
  601
  602 }
  603
  604 }
  605
  606 }
  607
  608 }
  609
  610 }
  611
  612 }
  613
  614 }
  615
  616 }
  617
  618 }
  619
  620 }
  621
  622 }
  623
  624 }
  625
  626 }
  627
  628 }
  629
  630 }
  631
  632 }
  633
  634 }
  635
  636 }
  637
  638 }
  639
  640 }
  641
  642 }
  643
  644 }
  645
  646 }
  647
  648 }
  649
  650 }
  651
  652 }
  653
  654 }
  655
  656 }
  657
  658 }
  659
  660 }
  661
  662 }
  663
  664 }
  665
  666 }
  667
  668 }
  669
  670 }
  671
  672 }
  673
  674 }
  675
  676 }
  677
  678 }
  679
  680 }
  681
  682 }
  683
  684 }
  685
  686 }
  687
  688 }
  689
  690 }
  691
  692 }
  693
  694 }
  695
  696 }
  697
  698 }
  699
  700 }
  701
  702 }
  703
  704 }
  705
  706 }
  707
  708 }
  709
  710 }
  711
  712 }
  713
  714 }
  715
  716 }
  717
  718 }
  719
  720 }
  721
  722 }
  723
  724 }
  725
  726 }
  727
  728 }
  729
  730 }
  731
  732 }
  733
  734 }
  735
  736 }
  737
  738 }
  739
  740 }
  741
  742 }
  743
  744 }
  745
  746 }
  747
  748 }
  749
  750 }
  751
  752 }
  753
  754 }
  755
  756 }
  757
  758 }
  759
  760 }
  761
  762 }
  763
  764 }
  765
  766 }
  767
  768 }
  769
  770 }
  771
  772 }
  773
  774 }
  775
  776 }
  777
  778 }
  779
  780 }
  781
  782 }
  783
  784 }
  785
  786 }
  787
  788 }
  789
  790 }
  791
  792 }
  793
  794 }
  795
  796 }
  797
  798 }
  799
  800 }
  801
  802 }
  803
  804 }
  805
  806 }
  807
  808 }
  809
  810 }
  811
  812 }
  813
  814 }
  815
  816 }
  817
  818 }
  819
  820 }
  821
  822 }
  823
  824 }
  825
  826 }
  827
  828 }
  829
  830 }
  831
  832 }
  833
  834 }
  835
  836 }
  837
  838 }
  839
  840 }
  841
  842 }
  843
  844 }
  845
  846 }
  847
  848 }
  849
  850 }
  851
  852 }
  853
  854 }
  855
  856 }
  857
  858 }
  859
  860 }
  861
  862 }
  863
  864 }
  865
  866 }
  867
  868 }
  869
  870 }
  871
  872 }
  873
  874 }
  875
  876 }
  877
  878 }
  879
  880 }
  881
  882 }
  883
  884 }
  885
  886 }
  887
  888 }
  889
  890 }
  891
  892 }
  893
  894 }
  895
  896 }
  897
  898 }
  899
  900 }
  901
  902 }
  903
  904 }
  905
  906 }
  907
  908 }
  909
  910 }
  911
  912 }
  913
  914 }
  915
  916 }
  917
  918 }
  919
  920 }
  921
  922 }
  923
  924 }
  925
  926 }
  927
  928 }
  929
  930 }
  931
  932 }
  933
  934 }
  935
  936 }
  937
  938 }
  939
  940 }
  941
  942 }
  943
  944 }
  945
  946 }
  947
  948 }
  949
  950 }
  951
  952 }
  953
  954 }
  955
  956 }
  957
  958 }
  959
  960 }
  961
  962 }
  963
  964 }
  965
  966 }
  967
  968 }
  969
  970 }
  971
  972 }
  973
  974 }
  975
  976 }
  977
  978 }
  979
  980 }
  981
  982 }
  983
  984 }
  985
  986 }
  987
  988 }
  989
  990 }
  991
  992 }
  993
  994 }
  995
  996 }
  997
  998 }
  999
  1000 }
  1001
  1002 }
  1003
  1004 }
  1005
  1006 }
  1007
  1008 }
  1009
  1010 }
  1011
  1012 }
  1013
  1014 }
  1015
  1016 }
  1017
  1018 }
  1019
  1020 }
  1021
  1022 }
  1023
  1024 }
  1025
  1026 }
  1027
  1028 }
  1029
  1030 }
  1031
  1032 }
  1033
  1034 }
  1035
  1036 }
  1037
  1038 }
  1039
  1040 }
  1041
  1042 }
  1043
  1044 }
  1045
  1046 }
  1047
  1048 }
  1049
  1050 }
  1051
  1052 }
  1053
  1054 }
  1055
  1056 }
  1057
  1058 }
  1059
  1060 }
  1061
  1062 }
  1063
  1064 }
  1065
  1066 }
  1067
  1068 }
  1069
  1070 }
  1071
  1072 }
  1073
  1074 }
  1075
  1076 }
  1077
  1078 }
  1079
  1080 }
  1081
  1082 }
  1083
  1084 }
  1085
  1086 }
  1087
  1088 }
  1089
  1090 }
  1091
  1092 }
  1093
  1094 }
  1095
  1096 }
  1097
  1098 }
  1099
  1100 }
  1101
  1102 }
  1103
  1104 }
  1105
  1106 }
  1107
  1108 }
  1109
  1110 }
  1111
  1112 }
  1113
  1114 }
  1115
  1116 }
  1117
  1118 }
  1119
  1120 }
  1121
  1122 }
  1123
  1124 }
  1125
  1126 }
  1127
  1128 }
  1129
  1130 }
  1131
  1132 }
  1133
  1134 }
  1135
  1136 }
  1137
  1138 }
  1139
  1140 }
  1141
  1142 }
  1143
  1144 }
  1145
  1146 }
  1147
  1148 }
  1149
  1150 }
  1151
  1152 }
  1153
  1154 }
  1155
  1156 }
  1157
  1158 }
  1159
  1160 }
  1161
  1162 }
  1163
  1164 }
  1165
  1166 }
  1167
  1168 }
  1169
  1170 }
  1171
  1172 }
  1173
  1174 }
  1175
  1176 }
  1177
  1178 }
  1179
  1180 }
  1181
  1182 }
  1183
  1184 }
  1185
  1186 }
  1187
  1188 }
  1189
  1190 }
  1191
  1192 }
  1193
  1194 }
  1195
  1196 }
  1197
  1198 }
  1199
  1200 }
  1201
  1202 }
  1203
  1204 }
  1205
  1206 }
  1207
  1208 }
  1209
  1210 }
  1211
  1212 }
  1213
  1214 }
  1215
  1216 }
  1217
  1218 }
  1219
  1220 }
  1221
  1222 }
  1223
  1224 }
  1225
  1226 }
  1227
  1228 }
  1229
  1230 }
  1231
  1232 }
  1233
  1234 }
  1235
  1236 }
  1237
  1238 }
  1239
  1240 }
  1241
  1242 }
  1243
  1244 }
  1245
  1246 }
  1247
  1248 }
  1249
  1250 }
  1251
  1252 }
  1253
  1254 }
  1255
  1256 }
  1257
  1258 }
  1259
  1260 }
  1261
  1262 }
  1263
  1264 }
  1265
  1266 }
  1267
  1268 }
  1269
  1270 }
  1271
  1272 }
  1273
  1274 }
  1275
  1276 }
  1277
  1278 }
  1279
  1280 }
  1281
  1282 }
  1283
  1284 }
  1285
  1286 }
  1287
  1288 }
  1289
  1290 }
  1291
  1292 }
  1293
  1294 }
  1295
  1296 }
  1297
  1298 }
  1299
  1300 }
  1301
  1302 }
  1303
  1304 }
  1305
  1306 }
  1307
  1308 }
  1309
  1310 }
  1311
  1312 }
  1313
  1314 }
  1315
  1316 }
  1317
  1318 }
  1319
  1320 }
  1321
  1322 }
  1323
  1324 }
  1325
  1326 }
  1327
  1328 }
  1329
  1330 }
  1331
  1332 }
  1333
  1334 }
  1335
  1336 }
  1337
  1338 }
  1339
  1340 }
  1341
  1342 }
  1343
  1344 }
  1345
  1346 }
  1347
  1348 }
  1349
  1350 }
  1351
  1352 }
  1353
  1354 }
  1355
  1356 }
  1357
  1358 }
  1359
  1360 }
  1361
  1362 }
  1363
  1364 }
  1365
  1366 }
  1367
  1368 }
  1369
  1370 }
  1371
  1372 }
  1373
  1374 }
  1375
  1376 }
  1377
  1378 }
  1379
  1380 }
  1381
  1382 }
  1383
  1384 }
  1385
  1386 }
  1387
  1388 }
  1389
  1390 }
  1391
  1392 }
  1393
  1394 }
  1395
  1396 }
  1397
  1398 }
  1399
  1400 }
  1401
  1402 }
  1403
  1404 }
  1405
  1406 }
  1407
  1408 }
  1409
  1410 }
  1411
  1412 }
  1413
  1414 }
  1415
  1416 }
  1417
  1418 }
  1419
  1420 }
  1421
  1422 }
  1423
  1424 }
  1425
  1426 }
  1427
  1428 }
  1429
  1430 }
  1431
  1432 }
  1433
  1434 }
  1435
  1436 }
  1437
  1438 }
  1439
  1440 }
  1441
  1442 }
  1443
  1444 }
  1445
  1446 }
  1447
  1448 }
  1449
  1450 }
  1451
  1452 }
  1453
  1454 }
  1455
  1456 }
  1457
  1458 }
  1459
  1460 }
  1461
  1462 }
  1463
  1464 }
  1465
  1466 }
  1467
  1468 }
  1469
  1470 }
  1471
  1472 }
  1473
  1474 }
  1475
  1476 }
  1477
  1478 }
  1479
  1480 }
  1481
  1482 }
  1483
  1484 }
  1485
  1486 }
  1487
  1488 }
  1489
  1490 }
  1491
  1492 }
  1493
  1494 }
  1495
  1496 }
  1497
  1498 }
  1499
  1499 }
  1500
  1500 }
}

```

Figura 3.36: Activar escenas. Fuente: autor

## Solicitud de base de datos

La base de datos se crea en el servidor de iRidium al crear un proyecto servidor. Si en su momento se trabaja con varios proyectos entonces se crea una base de datos por cada proyecto. También se puede personalizar una base de datos:

- Base de datos del sistema
  1. La escritura en base de datos es hecha en la configuración de las variables del servidor en el iRidium Studio.
  2. Se puede leer con la ayuda de solicitudes especiales de SQL.
  3. No se puede editar ni borrar.
- Base de datos del usuario
  1. La escritura en la base de datos se realiza en en JavaScript.
  2. Se puede leer con la ayuda de solicitudes especiales de SQL.
  3. Se puede editar y borrar con la ayuda de JavaScript.

Se puede encontrar el archivo de la base de datos en la interfaz web del servidor tal como se muestra en la Figura 3.37.

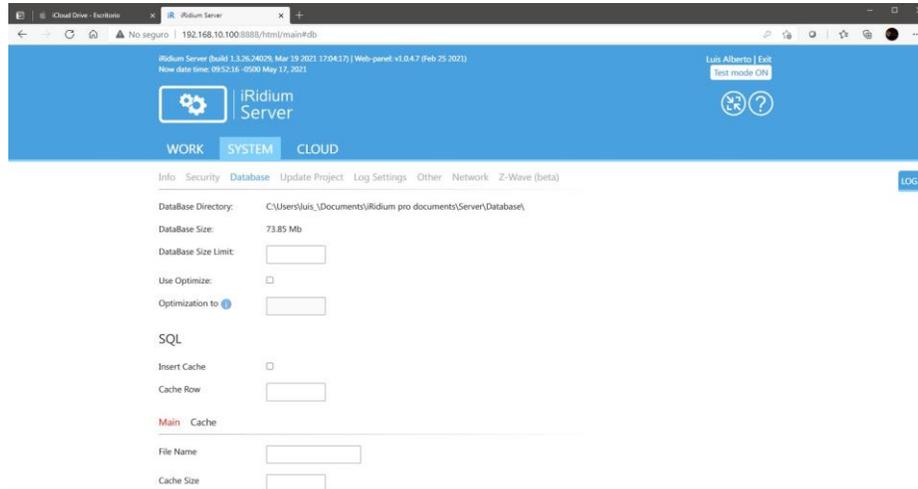


Figura 3.37: Ubicación base de datos. Fuente: autor

## Gráficas

Para mostrar gráficas cuando cambian las variables, iRidium guarda los valores de las variables en la base de datos SQL que se encuentra en la interfaz web del servidor.

Cuando se requiere crear un gráfico de variables cuando estas cambian, la interfaz de usuario realiza una solicitud a la base de datos para el período de tiempo en cuestión y se muestra el resultado de la gráfica. Además, se configura el guardado en la base de datos y se puede mostrar la variable en la interfaz de usuario como un gráfico. En la Figura 3.38 se observa la gráfica del sensor que cuando se establece en 27 grados Celsius, se produce la activación del sistema de climatización.

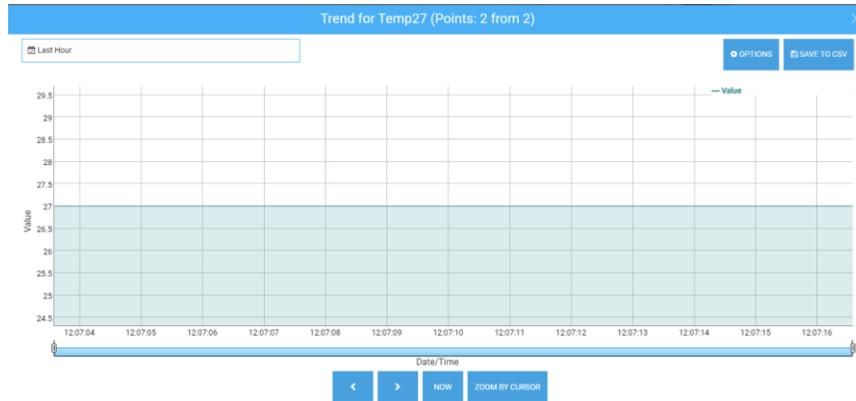


Figura 3.38: Gráficas desde el servidor web. Fuente: autor

### Sincronización con interfaz de usuario y servidor

En el proyecto se usa una interfaz de usuario y además un proyectoservidor, esto es lo más recomendable para ello se configura la conexión con el equipo de automatización, se crean tags virtuales y scripts para ellos en el lado del servidor. Después se realiza la importación del proyecto servidor permite que se pueda crear el conjunto de variables en el proyecto de visualización. Y de esta forma se asigna las variables importadas a los elementos gráficos. Para la sincronización de estos dos componentes se presiona en la pestaña de la Figura 3.39. [iRidium Studio, 2021c].

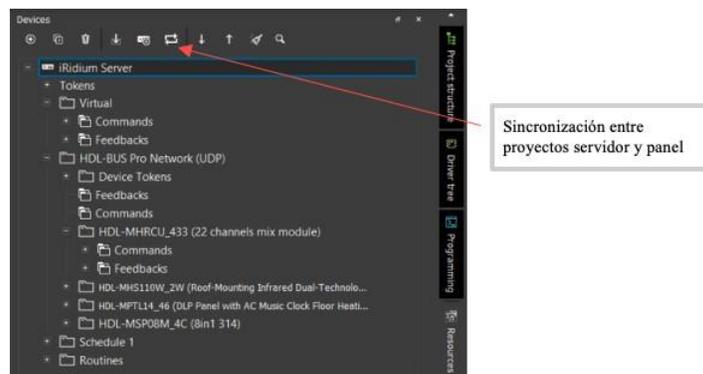


Figura 3.39: Sincronización servidor web e interfaz de usuario. Fuente: autor

### Cargar proyectos en el servidor

La carga de proyectos del servidor (\*.sirpz) en el servidor se realiza de manera similar a la carga de proyectos en i3 pro, es decir, con la ayuda de la aplicación iRidium "Transfer" en su PC, para ello ver la Figura 3.40. El aplicativo "Transfer" busca todos los servidores que se ejecutan en una subred IP y los muestra en su interfaz. Envía su proyecto y licencia a iRidium Server con la ayuda de Transfer.

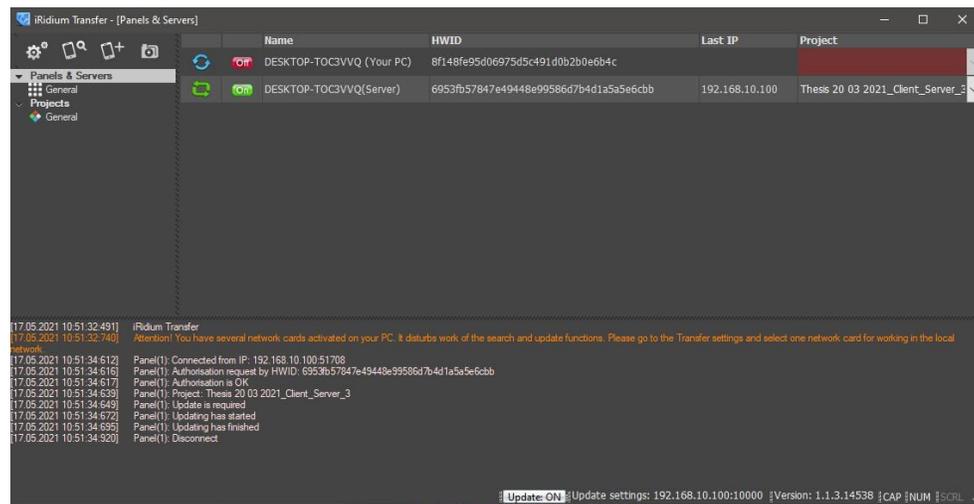


Figura 3.40: iRidium Transfer. Fuente: autor

En la gráfica tenemos un texto color naranja de Attention, pero se trata netamente de la conexión de nuestra computadora que se encuentra conectada a la red WiFi mientras que el proyecto se controla con un cable ethernet por LAN con una red diferente a la conexión WiFi de nuestro computador.

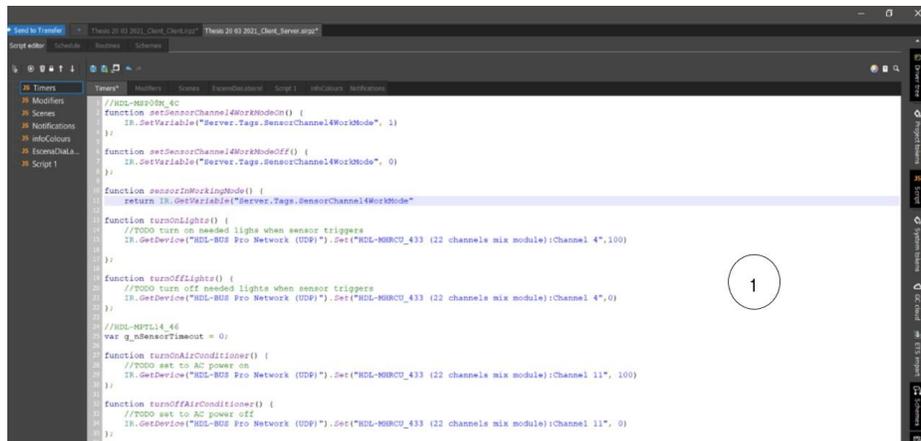
### Scripts proyecto servidor

En esta sección se coloca scripts desarrollados para el proyecto y que permiten trabajar con sensores, temperatura, aire acondicionado, notificaciones.

Para los sensores 8 en 1 se realiza un script el cual enciende luminarias del pasillo de la edificación, además dicho sensor permite el monitoreo

de temperatura en grados centigrados y el nivel de luxes obtener el nivel de iluminación en dicha área. En la Figura 3.41 se observa el script de desarrollo. En la misma Figura 3.41 también se programa el componente de aire acondicionado. En la Figura 3.42 se observa el servidor Web y los valores obtenidos por el sensor 8 en 1 en el servidor para componentes de niveles de luxes, temperatura, movimiento.

1. En este script se observa una programación similar a la trabajada en los componentes anteriores y se define por el uso de la API de Iridium con los métodos que me permiten manipular la programación de acuerdo con la necesidad de la edificación del proyecto. Bajo este criterio aparece el uso del método `IR.GetDevice` el cual es una función que permite usarse para acceder a los drivers. En este caso se accede al driver del módulo `HDLMHRCU_433`.



```

//HDL-MHRCU_433
function setSensorChannel4WorkModeOn() {
  IR.SetVariable("Server.Tags.SensorChannel4WorkMode", 1)
}

function setSensorChannel4WorkModeOff() {
  IR.SetVariable("Server.Tags.SensorChannel4WorkMode", 0)
}

function sensorIsWorkingMode() {
  return IR.GetVariable("Server.Tags.SensorChannel4WorkMode")
}

function turnOnLights() {
  //TODO turn on needed lights when sensor triggers
  IR.GetDevice("HDL-BUS Pro Network (UDP)").Set("HDL-MHRCU_433 (22 channels mix module):Channel 4", 100)
}

function turnOffLights() {
  //TODO turn off needed lights when sensor triggers
  IR.GetDevice("HDL-BUS Pro Network (UDP)").Set("HDL-MHRCU_433 (22 channels mix module):Channel 4", 0)
}

//HDL-MPEL14_46
var g_sensorTimeout = 0;

function turnOnAirConditioner() {
  //TODO set to AC power on
  IR.GetDevice("HDL-BUS Pro Network (UDP)").Set("HDL-MHRCU_433 (22 channels mix module):Channel 11", 100)
}

function turnOffAirConditioner() {
  //TODO set to AC power off
  IR.GetDevice("HDL-BUS Pro Network (UDP)").Set("HDL-MHRCU_433 (22 channels mix module):Channel 11", 0)
}

```

Figura 3.41: Scripts sensores y aire acondicionado. Fuente: autor

Name	Value	Set	Status
HDL-BUS Pro Network (UDP).HDL-MSP08M_4C (8in1 314):Input 5 Dry Contact 1	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MSP08M_4C (8in1 314):Input 4 Sonex	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MSP08M_4C (8in1 314):Input 3 PIR	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MSP08M_4C (8in1 314):Input 2 LUX	584.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MSP08M_4C (8in1 314):Input 1 Temperature	26.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Read Temperature New	21.500000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Read Temperature	22.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor Temperature Mode	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor Sampling Tempera...	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor Mode Select	1.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor Heating Switch	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor "Night" mode temp	20.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor "Day" mode temp	20.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor "Common" mode t...	20.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):Floor "Away" mode temp	20.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):AC Power	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):AC Mode	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):AC Lock	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):AC Heating Temp	0.000000	Set	✓
HDL-BUS Pro Network (UDP).HDL-MPTL14_46 (DLP Panel with AC Music Clock Floor Heating):AC Fan Speed	0.000000	Set	✓

First Page / Prev Page / 2 / 3 / 4 / 5 / 6 / Next Page / Last Page

Figura 3.42: Lectura de temperatura y nivel de luxes en servidor. Fuente: autor

El servidor iRidium puede notificar al usuario sobre cualquier evento en el sistema de automatización que reciba. Una de las formas de notificación es enviar notificaciones Push a la interfaz de usuario. El servidor de iRidium envía el comando "para notificar los paneles de control" al sitio web móvil de iRidium. El sitio web móvil de iRidium redirige el comando a los servicios de Apple o Google (según el sistema operativo del panel de control). Para las notificaciones de alarmas se utiliza el siguiente script y este notifica tanto en el iRidium server como en el interfaz de usuario. Para ello ver la Figura 3.43.

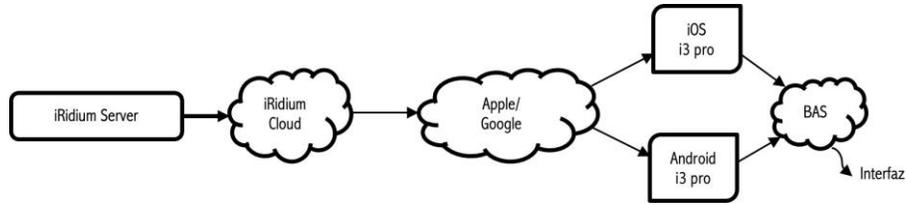


Figura 3.43: Notificaciones push. Fuente: autor

No es posible enviar notificaciones push sin tener un proyecto servidor iRidium. Si una interfaz de usuario no está conectada a internet no se puede recibir notificaciones push. En el caso del script que se muestra en la Figura 3.44, el método usado con la ayuda del API de iRidium permite enviar notificaciones a dispositivos iOS y Android.

1. En este script se observa una programación similar a la trabajada en los componentes anteriores y se define por el uso de la API de Iridium con los métodos que me permiten manipular la programación de acuerdo con la necesidad de la edificación del proyecto. Bajo este criterio aparece el uso del método `IR.GetDevice` el cual es una función que permite usarse para acceder a los drivers. En este caso se accede al driver del módulo `HDLMHRCU_433`.

El método `IR.SendPush` envía notificaciones al panel de control y al servidor tal como se muestra en la Figura 3.44.

```

//how to setup notifications - https://dev.iridi.com/Push_notifications/en
//how to setup handler in project if needed - https://dev.iridi.com/Push_API/en#Function_for_Processing_of_the_Push_Status

//these functions can be used in routines or in tag change listener to send push when value changes

function makeTemperatureNotification(in_sValue) {
  var l_sValue = in_sValue || 27;
  IR.SendPush('Temperature above ' + l_sValue + '°C', 'temperature warning', 1);
}

function makeAlarmNotification() {
  IR.SendPush('Alarm was triggered', 'alarm warning', 1);
}

function makeEnergyConsumptionNotification() {
  IR.SendPush('Energy consumption was exceeded', 'energy warning', 1);
}
  
```

Figura 3.44: Notificaciones push en el script del proyecto. Fuente: autor

### **3.3. Resultados y Discusión**

Actualmente el edificio no cuenta con un sistema de automatización en su implementación, es decir su funcionamiento base es el encendido/apagado de luminarias manualmente y el flujo de corriente de aire no está contemplado. No hay medición de temperatura, ni tampoco se encuentra contemplado medir el nivel de luminosidad de los espacios interiores.

De esta forma en este proyecto se propuso el desarrollo de un software de gestión para mejorar el comportamiento del edificio “Padre Juan Bottasso” basado en el control de iluminación y climatización. Para ello se utiliza una maqueta de tal forma que se pueda emular el comportamiento del software en el edificio.

La aplicación esta diseñada para una operación confortable en todo el sistema de iluminación y climatización. El sistema de gestión funciona desde una interfaz de usuario que se visualiza en una tablet y desde un proyecto servidor generado en un computador con sistema operativo Windows. Se usa una interfaz de usuario para que el operador del edificio se le facilite la interacción con el software y se agrega un proyecto servidor para generar una automatización avanzada en la programación del sistema que ya ha sido explicada en la sección anterior, además de poder guardar en una base de datos el comportamiento del edificio.

Para el sistema de iluminación se lo dividió en tres áreas: aulas (6 circuitos de iluminación), pasillos (1 circuito de iluminación) y oficinas (2 circuitos de iluminación). Todas estas áreas funcionan en un rango de 0 a 100. Para el control y optimización de iluminación se plantea en el proyecto el uso de escenas, horarios y rutinas. De esta forma las escenas encienden/apagan luminarias y aire acondicionado de acuerdo con las rutinas diarias del edificio. Para ejemplificar esta optimización lo que se hace es definir un horario de operación y este puede ser limitado a ciertos días de operación y por intervalos de tiempo específico, una vez definido un horario se procede con las rutinas es decir que si la condición de horario se cumple en la rutina se ejecuta una acción de programación. Las escenas en el control de iluminación nos permiten crear ambientes que se repiten todo el tiempo y facilita al momento de programar reduciendo el tiempo de programación.

Para el control de iluminación en pasillos se utiliza un sensor 8 en 1 de

la marca HDL que permite el encendido/apagado de luminarias en el lugar. Además de cumplir como sensor de movimiento ofrece otras mediciones como nivel de luminosidad, temperatura, comandos IR, contactos secos, y más. Este sensor facilita en gran cantidad la integración de más equipos para obtener diferentes mediciones. Mediante el uso de este sensor se podrían realizar un sin número de combinaciones de programación para obtener un comportamiento adecuado en el espacio que se está trabajando. Es así como por ejemplo se puede programar que si en un espacio hay movimiento y el nivel de luminosidad está por debajo de 250 luxes se programa el encendido de las luminarias caso contrario no lo harían. O también se puede realizar una programación combinada la cual, si el sensor obtiene una medición de temperatura que esté por encima de 27 grados centígrados, pero no hay movimiento no se encendería el aire acondicionado con el fin de optimizar el recurso energético.

Para finalizar en el área de la iluminación de acuerdo con el valor medido por el sensor en el nivel de luminosidad dando la medida en luxes, si el valor está por debajo de 250 luxes en el espacio medido se procede con el encendido de luminarias y si el valor sobrepasa los 400 luxes se procede con el apagado de luminarias en dicho espacio.

Para la climatización se procede a realizar una programación de activación/desactivación que depende netamente de la temperatura del espacio medido. Con ello se logra mantener un confort en el edificio y una recirculación de aire. En este caso se utiliza un panel físico de la marca HDL para la medición de temperatura y cuando el valor de temperatura sobrepasa los 27 grados centígrados se activa el canal que controla el aire acondicionado. Cuando la temperatura vuelve a estar por debajo de 24 grados centígrados el aire acondicionado se apaga.

Para notificar sucesos que cambian el comportamiento del edificio se ha colocado alarmas. Para ello cuando la temperatura sobrepasa los 27 grados centígrados recibimos una alarma en el interfaz de usuario para que se alerte de este suceso. También se tiene una notificación cuando se activa una alarma, es decir, cuando un detector óptico de un sistema de detección de incendios se activa se notifica al interfaz de usuario para que el operador se percate de esta situación y se realice una atención inmediata al suceso. Una vez definido el funcionamiento base del edificio y el software desarrollado, el desarrollo y programación resultante demuestra el cálculo de consumo de energía y niveles de confort en diferentes escenarios.

La comparación siempre es entre el funcionamiento base que imita el comportamiento de un edificio sin automatización y el BAS. El escenario planteado es el siguiente:

- Sistema iluminación operando ocho horas.
- Las luces se mantienen encendidas mientras hay alguien dentro del espacio, se apagan con un 50 % de probabilidad cuando los ocupantes dejan el espacio ocupado y se apagan al final de la jornada laboral.

El número de ocupantes del edificio se ha definido de acuerdo con la ocupación promedio en los diferentes espacios: 6 aulas por piso con una cantidad de 150 alumnos. La oficina, pasillos y laboratorios con una cantidad de 50 personas. La simulación es considerando un día de 24 horas de operación. La edificación cuenta con 1200 metros cuadrados de construcción.

Los resultados obtenidos se evalúan teóricamente ya que el edificio no ha empezado a operar en un 100 % desde su creación ya que fue inaugurado el 31 de enero del 2020 y mientras se definía su operación llegó en el mes de marzo del 2020 la pandemia del coronavirus y desde ahí ha estado cerrado la mayoría del tiempo y su operación desde su creación ha sido intermitente con uso de menos del 5 % de su capacidad, esto sucede hasta la fecha en la que se termina esta tesis quedando los análisis de resultados para una siguiente investigación.

El consumo energético de las lámparas de acuerdo con el tipo de luminaria que tiene el edificio es el siguiente:

$$\begin{aligned}
 P(\text{Watts}) &= \#luminarias * consumo(\text{Watts}) \\
 P(\text{Watts}) &= (107 * 60) + (36 * 40) + (82 * 24) + \\
 &\quad (27 * 60) + 3 * ((8 * 24) + (67 * 60)) \\
 P(\text{Watts}) &= 24084 \text{Watts}
 \end{aligned}$$

Si aproximamos que el uso de energía diario del edificio en horas es de 8 y la concurrencia al mismo es de 22 días se realiza la siguiente fórmula:

$$\text{Consumo} = P (\text{Watts}) * \text{horas} * \text{días}$$

$$\text{Consumo} = 24084 * 8 * 22$$

$$\text{Consumo} = 4'238,78\text{Wh/mes}$$

$$\text{Consumo} = 4238,78\text{kWh/mes}$$

El consumo energético de acuerdo con las cantidades de luminarias que se encuentran en el edificio es de 4.238,78 kWh/mes. De esta forma considerando que el costo mensual de consumo energético por kWh en Cuenca es de 0,095 centavos de dólar según [David et al., 2019], y cuando el edificio se encuentra 100 % operativo durante 8 horas de uso entonces se realiza lo siguiente:

$$\text{Total} = \text{Consumo} * \text{Costo}$$

$$\text{Total} = 4238,78\text{kWh/mes} * \$0,095\text{USD}$$

$$\text{Total} = \$402,68\text{USD}$$

Es decir el valor total a cancelar por cada mes es de \$ 402,68 USD aproximadamente.

En comparación entre el sistema convencional donde el consumo energético es de 4238,78 KWh como se observa en la Figura 3.45, si esto lo representamos con el uso del software de gestión desarrollado estos valores se ven alterados en el componente de iluminación que es el dato teórico medible en el proyecto ya que, con la creación de horarios de uso, rutinas programadas, escenas de acuerdo a la necesidad del espacio y sumado a los sensores de movimiento del edificio es lógico pensar que el consumo energético en el componente de iluminación se vería disminuido. Según el artículo [Sembroiz et al., 2019], se observa la comparativa tanto en consumo energético como el confort del edificio entre un edificio convencional y un edificio con sistema BAS.

Mediante el análisis en la más alta predicción del sistema el consumo energético se reduce en un total del 31 % con un sistema BAS quedando en 2967,14 KWh el consumo energético del sistema de iluminación. Esto debido a los componentes de programación previamente desarrollados en el software. Esto representa un costo anual de 3382,54 USD presentando un ahorro económico de 1449,66 USD.

*Consumo energético aproximado de iluminación actualmente*

	Luminaria 40W	Luminaria ojo de buey 24W	Luminaria 60W	Potencia (Watts)	Horas uso	Días uso	Consumo por piso
Subsuelo			107	6.420	8	22	1'129.920
Planta baja	36	82	27	5.028	8	22	884.928
Primera planta alta		8	67	4.212	8	22	741.312
Segunda planta alta		8	67	4.212	8	22	741.312
Tercera planta alta		8	67	4.212	8	22	741.312
Consumo mensual aproximado (Watts)							4'238.784
Consumo mensual aproximado (Kwh)							4.238,78
Demanda de energía							4.238,78
Costos de energía (Kwh)							0,095
Total mensual							\$ 402,68
Total anual							\$ 4.832,21

Figura 3.45: Consumo energético aproximado de iluminación. Fuente: autor

Un sistema convencional como actualmente lo es el edificio “Padre Juan Bottasso” desperdicia una gran cantidad de energía eléctrica por mantener iluminación encendida en espacios innecesarios, esto agregando que si existiera un sistema de climatización estos valores aumentarían aún más. Al contar con un sistema BAS incorporado el sistema trabaja y controla el comportamiento del edificio de acuerdo a las necesidades de ocupación.

El ahorro energético y confort se refleja cuando al acceder a un aula o un espacio de coworking el espacio se encuentra debidamente adaptado a las necesidades de los ocupantes, por ejemplo, las luminarias se encuentran encendidas en un porcentaje de acuerdo a los niveles de luxes medidos en el lugar y el sistema de aire acondicionado se encuentra en la temperatura adecuada. Otro ejemplo de ahorro energético se puede medir cuando el ocupante accede al edificio y las luces se encienden por la comparación que hace el software entre el sensor de movimiento y el nivel de luxes que tiene el espacio en el lugar, si el nivel de luxes es mayor a un umbral programado entonces estas luces no se encienden puesto que el ocupante puede desplazarse sin dificultad. Se pueden optar por un universo de combinaciones programables dependiendo de la necesidad de cada edificio.

El costo de un sistema de gestión BAS se define por el uso de componentes de hardware y software. Para ello presentamos una cotización que se encuentra en anexos la cual nos muestra que el costo del software es de \$ 4800+iva USD para un ilimitado número de paneles y 10000 tags. El costo de equipamiento para el edificio Juan Bottasso es de \$ 21178.80+iva USD. El valor restante es mano de obra. Con ello si solo tomamos en cuenta la amortización por costo de energía esta inversión se recuperaría en 20 años aproximadamente. Sin embargo, hay que tener en cuenta que en este análisis solo se está tomando en cuenta la iluminación y no otros sistemas que influyen en el consumo energético y que también se gestionan. Además, el sistema de gestión no solo busca optimizar el consumo energético sino también tener edificios donde su gestión es unificada en un solo software, obtener información en tiempo real para actuar de forma inmediata a eventos que se producen en la edificación consiguiendo seguridad en nuestros sistemas instalados.

En la discusión final de este estudio se evidencia que un sistema BAS es capaz de reducir el consumo de energía entre un 19 % a 31 % manteniendo los niveles de confort de los ocupantes del edificio, situación similar que se ha planteado en nuestro proyecto en el edificio “Padre Juan Botasso”, el cual tendrá que ser probado cuando se permita regresar a una nueva normalidad en donde el edificio tenga su 100 % de operatividad, y con esto utilizar la aplicación desarrollada, donde se podrá obtener valores reales de eficiencia energética gracias al software desarrollado y comparar los niveles de confort y eficiencia energética.

### **3.4. Conclusiones**

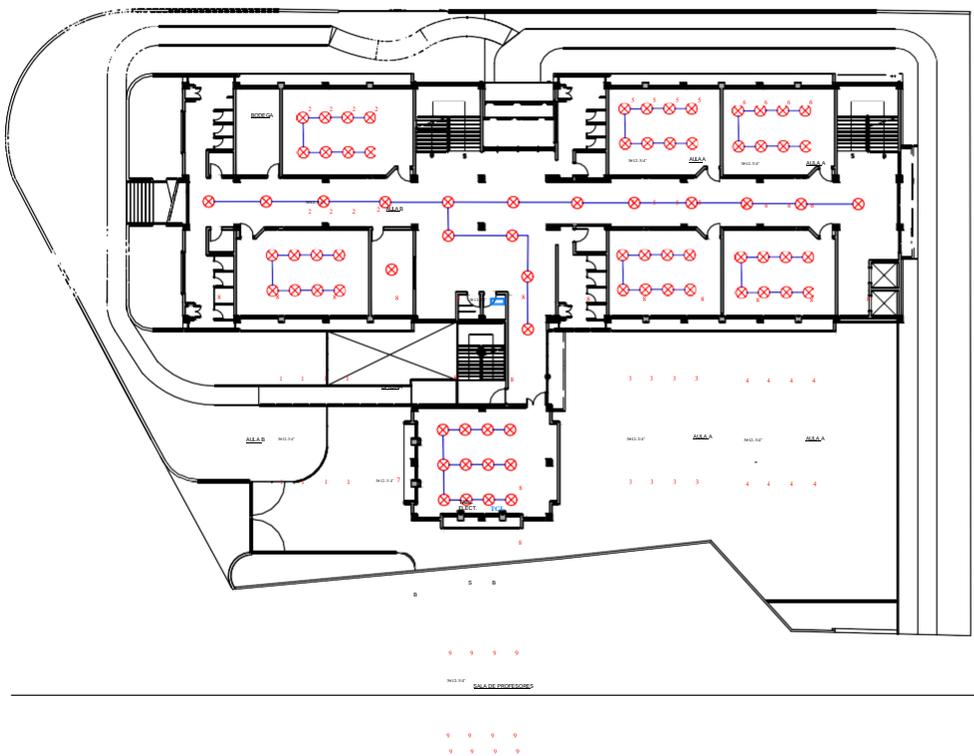
1. El sistema de gestión implementado controla el sistema de iluminación y climatización en una maqueta que representa el interior del edificio “Padre Juan Bottasso” ubicado en la ciudad de Cuenca. Se realizó un levantamiento de la planimetría en el programa AutoCAD para diseñar los planos de iluminación y climatización. Estos planos también sirvieron para la impresión de la maqueta en formato 3D. Los resultados de simulación del comportamiento del software fueron satisfactorios para el control de estos sistemas y para obtener eficiencia energética.
2. La programación del software para la creación de ambientes permite obtener eficiencia energética manteniendo niveles de confort en la

edificación; evidenciando teóricamente que con el uso de un software de gestión se obtiene un ahorro energético en un rango del 19 % al 31 % según el artículo publicado *Planning and Operational Energy Optimization Solutions for Smart Buildings* en la revista *Information Sciences* respecto a un sistema constructivo convencional, artículo que hemos referenciado para obtener los resultados.

3. Se utilizó equipamiento de hardware de la marca HDL que usa una tecnología propietaria de comunicación que se basa en el estándar RS485. Para el software se utilizó la plataforma iRidium que permite el control e integración de sistemas de automatización de diferentes fabricantes.
4. Los equipos para la implementación de sistemas de gestión y de comunicación están disponibles en el mercado. La tecnología basada en el estándar RS485 es la más conveniente para la ejecución del proyecto porque permite una gran longitud de línea de comunicación y tiene una alta velocidad de transmisión. Además, es compatible con la marca que se ha seleccionado para el equipamiento del sistema de gestión de este proyecto.
5. El costo de implementación del sistema de gestión con equipamiento de hardware es de 21,178.80+IVA. Si tomamos en cuenta solo el componente de consumo energético, esta inversión se recuperará en aproximadamente 20 años. Sin embargo, no se puede determinar de esta forma la amortización de la inversión puesto que hay sistemas como la climatización, ventilación, motores, equipos, que mantienen un consumo energético alto y que no se toman en cuenta en este análisis debido a dos razones; la primera: el edificio no contempla un sistema de climatización y ventilación, la segunda: el edificio no se encuentra operando completamente por motivo de pandemia. Por esta razón, se sugiere que el estudio de amortización y de consumo energético global se lo pueda realizar en una nueva investigación.

## **Apéndice A**

# **Anexo: Planos de AutoCAD para iluminación y climatización**



**ESPECIFICACIONES TECNICAS**

Las instalaciones eléctricas debe cumplir con las especificaciones técnicas de las normas IEC.

Los puntos que se entregan son sólo indicativos en lo que se refiere a los recorridos de tubería. La localización exacta lo deberá hacer el Contratista.

Los retornos marcados en este plano deberán ir al tablero T.C.L.

Se instalará punto de voz y datos categoría 6A en tablero T.C.L. proveniente de rack de telecomunicaciones.

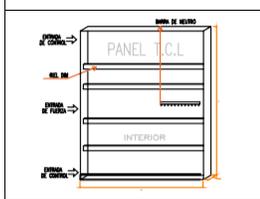
Todo el cableado eléctrico va en diferente configuración del cableado electrónico.

Ubicación del sensor web se define en obra y se instalará punto de voz y datos categoría 6A.

Se instalará un punto de voz y datos categoría 6 desde el tablero T.C.L. al equipo ELP.

Los cables del edificio, muestra de los retornos y otros se cablearán desde el tablero eléctrico al tablero T.C.L.

Llevar a los sensores de movimiento cable de voz y datos categoría 6



**SIMBOLOGIA:**

- Tablero de Automatización
- Punto de Suministro
- Tubería DMF F

  
**UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR**  
 MAESTRIA EN ELECTRONICA Y AUTOMATIZACION  
 ING. LUIS ALBERTO ROFERO PERAL

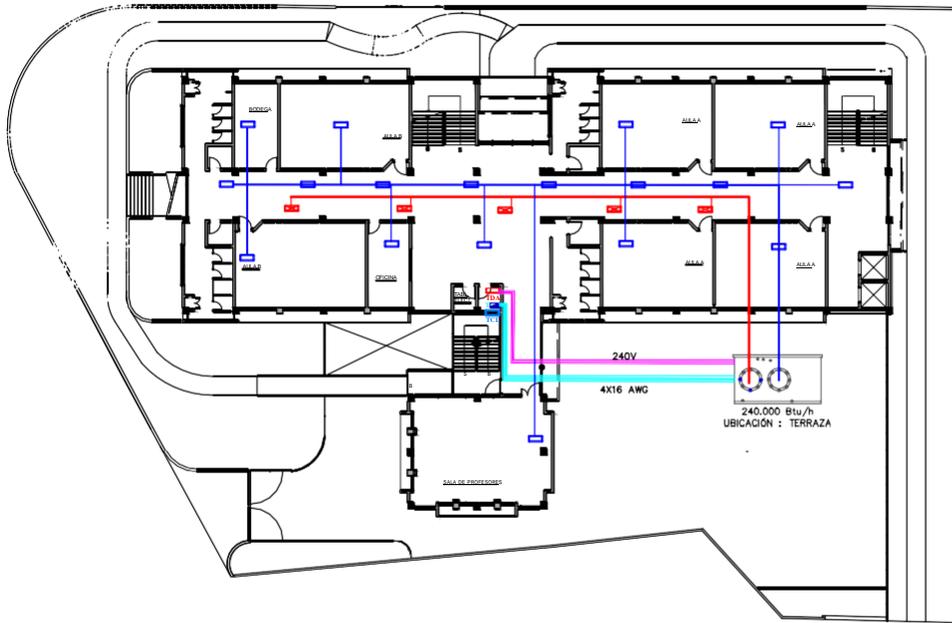
**NOMBRE DEL PROYECTO:**

DESARROLLO DE UN SISTEMA DE GESTIÓN PARA EL CONTROL DE ILUMINACIÓN Y CLIMATIZACIÓN EN EDIFICACIONES, APLICADO AL EDIFICIO "PADRE JUAN BOTTASO" DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE CUENCA

**CONTENIDO:**

DISEÑO DEL PLANO DE CONTROL DE ILUMINACIÓN PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN

ESCALA: LAS INDICADAS  
 FECHA: NOVIEMBRE 2021  
 1/1



PLANTA TIPO

Plano de Climatización

1:150

ESPECIFICACIONES TECNICAS

Las instalaciones de aire acondicionado deberá cumplir con las especificaciones Mínimas de las normas ASHRAE:

Las juntas que se entregan son aún indicadas en lo que se refiere a los requisitos de ductería y tubería. La localización exacta la deberá hacer el Contratista.

Se instalará un punto de control de 220V entre la unidad de paquete y el T.D.A.

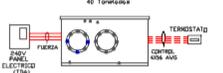
Se instalará un punto de control 4x8 AWG entre la unidad de paquete y el Tablero de Control T.C.A.

Ubicación de la unidad de paquete en la terraza.

La ductería de suministro tendrá las medidas que se encuentran en el anexo 1

La ductería de retorno tendrá las medidas que se encuentran en el anexo 1

DIAGRAMA DE ALIMENTACION UNIDAD PAQUETE 14



SIMBOLOGIA:

- Tablero Electrico
- Termostato
- Tablero de Automatización
- Ductería de suministro de 24" x 12"
- Ductería de retorno de 24" x 12"



MAESTRIA EN ELECTRONICA Y AUTOMATIZACION

ING. LUIS ALBERTO ROFERO PERAL

NOMBRE DEL PROYECTO:

DESARROLLO DE UN SISTEMA DE GESTIÓN PARA EL CONTROL DE ILUMINACIÓN Y CLIMATIZACIÓN EN EDIFICACIONES, APLICADO AL EDIFICIO PADRE JUAN BOTTASO DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE CUENCA

CONTENIDO:

DISEÑO DEL PLANO DE CONTROL DE CLIMATIZACIÓN PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN

ESCALA: LAS INDICADAS  
FECHA: NOVIEMBRE 2021

1/1

## **Apéndice B**

### **Anexo II: Presupuesto**

**COTIZACIÓN**  
MR-001-643

info@mrautomatizacion.com  
www.mrautomatizacion.com  
móvil. +593 939688461

Cliente: Luis Alberto Riofrío  
Ciudad: Cuenca  
Proyecto: Automatización edificio "Padre Juan Bottasso"



Código	Concepto	Unidad	Cantidad	Precio Unitario	Precio Total
<b>101</b>	<b>Equipos de automatización</b>				
101.001	Módulo compacto. HDL Buspro, RJ45, Interfaz de red UDP/IP. Tiene un controlador de mezcla, 22 canales de salida, 24 canales de contactos secos y 12 canales de salidas LED	u	4,00	756,60	3.026,40
101.002	Sensor 8in1 (montaje en techo), incluye sensor de movimiento PIR, sensor de temperatura, sensor LUX, contacto seco, etc. Puede cargar el código IR desde la herramienta de configuración de HDL Buspro y enviar el código IR para controlar el dispositivo	u	28,00	163,80	4.586,40
101.003	Sensor PIR & LUX (montaje en techo), incluye sensor de movimiento PIR, sensor LUX, bloque lógico. El bloque lógico puede combinar todos los sensores para diferentes aplicaciones	u	8,00	150,80	1.206,40
101.004	Fuente de voltaje 750 mA, entrada de 110V/220V AC 50-60Hz, salida DC24/750mA, (protección contra sobrecalentamiento, sobrecarga, sobretensión).	u	4,00	117,00	468,00
101.005	Botonera pantalla táctil a color Enviro (pantalla de 4,3 pulgadas). El panel se puede usar para controlar iluminación, escenas, música y calefacción de piso. Tiene sensor de temperatura incorporado y receptor IR.	u	4,00	566,80	2.267,20
101.006	Botonera táctil 4 botones US, con retroiluminación LED (selección de color RGB en software). Interruptor, atenuación y control de escena. No admite instalación empotrada.	u	28	176,80	4.950,40
101.007	Interfaz de botoneras US. Proporciona energía y se comunica con la señal del interruptor del panel. Admite diferentes botoneras inteligentes tipo US.	u	28	45,50	1.274,00
101.008	Accesorios (no incluye cableado)	u	1,00	600,00	600,00
101.009	Armarios de automatización	u	4,00	250,00	1.000,00
<b>202</b>	<b>Software de Gestión</b>				
202.001	Computador servidor	u	1,00	1800,00	1.800,00
202.002	Licencia de gestión ENTERPRISE+ (10,000 tags)	u	1,00	4800,00	4.800,00
<b>303</b>	<b>Mano de obra</b>				
303.001	Instalación, Supervisión, programación y puesta en marcha de todo el sistema de audio y automatización. Parte/A. Dirección y seguimiento de obra, planos y esquemas con personal técnico. Parte/B. Instalación y conexión de elementos de automatización. Parte/C. Configuración completa del sistema, documentación, puesta en marcha, entrega	u	1,00	2000,00	2.000,00

Subtotal	\$ 27.978,80
I.V.A	\$ 3.357,46
<b>TOTAL</b>	<b>\$ 31.336,26</b>

Figura B.1: Presupuesto sistema BAS edificio "Padre Juan Bottasso".

# Bibliografía

- Abergel Thibaut and Delmastro Chiara. Tracking Buildings 2020 – Analysis - IEA, 6 2020. URL <https://www.iea.org/reports/tracking-buildings-2020#>.
- A. Antonini, A. Barengi, G. Pelosi, and S. Zonouz. Security challenges in building automation and SCADA. *Proceedings - International Carnahan Conference on Security Technology*, 2014-Octob(October), 2014. ISSN 10716572. doi: 10.1109/CCST.2014.6986996.
- Australian Government. Building management systems , 2021. URL <https://www.energy.gov.au/business/equipment-and-technology-guides/building-management-systems#toc-anchor-more-information>.
- I. David, J. Sánchez, M. Patricio, S. Taday, I. J. Luis, and R. Espinoza. ANÁLISIS DE CONSUMO DE ENERGÍA EN EL EDIFICIO CORNELIO MERCHÁN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, APLICACIÓN DEL PROTOCOLO IPMVP Y EVALUACIÓN DE MEDIDAS DE COSTO CERO. Technical report, Cuenca, 2019.
- Davila Fernando; Varela Daniel. DETERMINACION DE LA HUELLA DE CARBONO EN LA UNIVERSIDAD POLITECNICA SALESIANA, SEDE QUITO, CAMPUS SUR. Technical report, Quito, 2014. URL <http://dspace.ups.edu.ec/handle/123456789/7248>.
- P. Domingues, P. Carreira, R. Vieira, and W. Kastner. Building automation systems: Concepts and technology review. *Computer Standards and Interfaces*, 45:1–12, 2016. ISSN 09205489. doi: 10.1016/j.csi.2015.11.005. URL <http://dx.doi.org/10.1016/j.csi.2015.11.005>.
- P. Ehrlich. *Building automation*, volume 31. 2014. ISBN 9783540754114. doi: 10.1515/9783035612912-008.

- A. Fernbach, W. Granzer, and W. Kastner. Interoperability at the management level of building automation systems: A case study for BACnet and OPC UA. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2011. doi: 10.1109/ETFA.2011.6059106.
- B.Fouladi and S. Ghanoun. Security Evaluation of the Z-Wave Wireless Protocol. *Black hat*, page 6, 2013.
- W. Granzer and W. Kastner. Information modeling in heterogeneous Building Automation Systems. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, pages 291–300, 2012. doi: 10.1109/WFCS.2012.6242577.
- HDL Automation. Products-HDL Automation-Residential Hospitality Commercial, 2021a. URL <https://www.hdlautomation.com/product10000000163401.html>.
- HDL Automation. Products-HDL Automation-Residential Hospitality Commercial, 2021b. URL <https://www.hdlautomation.com/productex-100000109027698.html>.
- HDL Automation. Products-HDL Automation-Residential Hospitality Commercial, 2021c. URL <https://www.hdlautomation.com/product100000054462724.html>.
- HDL Automation. Products-HDL Automation-Residential Hospitality Commercial, 2021d. URL <https://www.hdlautomation.com/product100000206366553.html>.
- HDL Automation. Smart Home Automation Products Manufacturer - HDL Automation, 2021e. URL <https://www.hdlautomation.com/>.
- HDL Automation. Whole Smart Home Automation Company - HDL Smart Home, 2021f. URL <https://hdlautomation.com/about.html>.
- Hojjati Behjat. Global energy consumption driven by more electricity in residential, commercial buildings - Today in Energy - U.S. Energy Information Administration (EIA), 10 2019. URL <https://www.eia.gov/todayinenergy/detail.php?id=41753>.
- iRidium Studio. iRidium controlling of Smart home and IoT devices, 2021a. URL <https://iridi.com/>.

- iRidium Studio. iRidium server, 2021b. URL [https://dev.iridi.com/Products\\_server/en](https://dev.iridi.com/Products_server/en).
- iRidium Studio. iRidium graphic interface - creating of Smart Home visualization, 2021c. URL [https://dev.iridi.com/Editor\\_Tools/en](https://dev.iridi.com/Editor_Tools/en).
- M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan. Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101(July 2018):111–126, 2019. ISSN 09265805. doi: 10.1016/j.autcon.2019.01.023. URL <https://doi.org/10.1016/j.autcon.2019.01.023>.
- Kahan Ari. Global electricity consumption continues to rise faster than population - Today in Energy - U.S. Energy Information Administration (EIA), 6 2020. URL <https://www.eia.gov/todayinenergy/detail.php?id=44095>.
- W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman. Communication systems for building automation and control. *Proceedings of the IEEE*, 93(6):1178–1203, 2005. ISSN 00189219. doi: 10.1109/JPROC.2005.849726.
- A. Lavric and A. I. Petrariu. LoRaWAN communication protocol: The new era of IoT. *2018 14th International Conference on Development and Application Systems, DAS 2018 - Proceedings*, pages 74–77, 2018. doi: 10.1109/DAAS.2018.8396074.
- W. S. Lee and S. H. Hong. KNX - zigbee gateway for home automation. *4th IEEE Conference on Automation Science and Engineering, CASE 2008*, pages 750–755, 2008. doi: 10.1109/COASE.2008.4626433.
- P. Li, J. Li, L. Nie, and B. Wang. Research and application of ZigBee protocol stack. *2010 International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2010*, 2:1031–1034, 2010. doi: 10.1109/ICMTMA.2010.275.
- H. Liu, H. Ning, Q. Mu, Y. Zheng, J. Zeng, L. T. Yang, R. Huang, and J. Ma. A review of the smart world. *Future Generation Computer Systems*, 96 (2017):678–691, 2019. ISSN 0167739X. doi: 10.1016/j.future.2017.09.010. URL <http://dx.doi.org/10.1016/j.future.2017.09.010>.
- L. Martirano and M. Mitolo. Building Automation and Control Systems (BACS): A Review. *Proceedings - 2020 IEEE International Conference on*

*Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe, IEEEIC / I and CPS Europe 2020*, (May 2013), 2020. doi: 10.1109/IEEEIC/ICPSEurope49358.2020.9160662.

- A. McGibney, S. Rea, and J. Ploennigs. Open BMS - IoT driven architecture for the internet of buildings. *IECON Proceedings (Industrial Electronics Conference)*, pages 7071–7076, 2016. doi: 10.1109/IECON.2016.7793635.
- I. O. Monfort. Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa. 6 2017. URL <http://hdl.handle.net/10609/64365><http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64365/6/iordonezTFM0617memòria.pdf>.
- NETxAutomation. (1) NETxAutomation en Twitter: "Here is the all-in-one #BMS #solution <https://t.co/OPeFpcjt0l> #OPC #KNX #BACnet #LaMPS #ShutterControl #SmartBuilding #SmartCity <https://t.co/D8EfUHJv9U>/ Twitter, 6 2019. URL <https://twitter.com/netxautomation/status/1139184225146540033>.
- Organización Mundial de la Salud. Preguntas y respuestas sobre la ventilación y el aire acondicionado y la COVID-19, 3 2021. URL <https://www.who.int/es/news-room/q-a-detail/coronavirus-disease-covid-19-ventilation-and-air-conditioning>.
- T. J. Park and S. H. Hong. Experimental case study of a BACnet-based lighting control system. *IEEE Transactions on Automation Science and Engineering*, 6(2):322–333, 2009. ISSN 15455955. doi: 10.1109/TASE.2008.2008148.
- Posso Alex; Melo Paul; Flores Santiago; Haro Lenin. Factor de emisión de CO2 del sistema Nacional Interconectado del Ecuador, 2019. ISSN 1098-6596. URL [www.ambiente.gob.ec](http://www.ambiente.gob.ec).
- D. Sembroiz, D. Careglio, S. Ricciardi, and U. Fiore. Planning and operational energy optimization solutions for smart buildings. *Information Sciences*, 476:439–452, 2019. ISSN 00200255. doi: 10.1016/j.ins.2018.06.003. URL <https://doi.org/10.1016/j.ins.2018.06.003>.
- T. Senst, M. Pätzold, R. H. Evangelio, V. Eiselein, I. Keller, and T. Sikora. On building decentralized wide-area surveillance networks based on ONVIF. *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2011*, pages 420–423, 2011. doi: 10.1109/AVSS.2011.6027365.

- H. Shahnasser and Q. Wang. 1309 Controlling Industrial Devices Over Tcp/Ip. pages 1309–1314.
- P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34:409–429, 2014. ISSN 13640321. doi: 10.1016/j.rser.2014.03.027. URL <http://dx.doi.org/10.1016/j.rser.2014.03.027>.
- D. Snoonian. Can building automation systems overcome interoperability problems to assert control over our offices, hotels, and airports? By Deborah Snoonian. *IEEE Spectrum*, August:18–23, 2003.
- Valzania Gilberto. How Smart Buildings Will Grow in the Next 5 Years | WRLD3D, 12 2018. URL <https://www.world3d.com/blog/how-smart-buildings-will-grow-in-the-next-5-years/>.
- W. Vandenberghe, B. Latré, F. De Greve, P. De Mil, S. den Berghe, K. Lamont, I. Moerman, M. Mertens, J. Avonts, C. Blondia, and G. Impens. A system architecture for wireless building automation, 2006. URL <http://hdl.handle.net/1854/LU-353627>.
- Wood Laura. Smart Building Market: Growth, Trends & Forecasts to 2025, 1 2020. URL <https://www.globenewswire.com/news-release/2020/01/21/1973111/0/en/Smart-Building-Market-Growth-Trends-Forecasts-to-2025.html>.
- W. You and H. Ge. Design and Implementation of Modbus Protocol for Intelligent Building Security. *International Conference on Communication Technology Proceedings, ICCT*, pages 420–423, 2019. doi: 10.1109/ICCT46805.2019.8946996.