

**UNIVERSIDAD DE HUANUCO**  
**FACULTAD DE INGENIERIA**  
**PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS E**  
**INFORMÁTICA**



**TESIS**

---

**“DISEÑO Y PROPUESTA DE UN SISTEMA DE CONTROL DE LA  
ILUMINACIÓN DEL LABORATORIO DE CÓMPUTO DE LA  
UNIVERSIDAD NACIONAL HERMILIO VALDIZÁN BAJO LA  
ARQUITECTURA DE COMPUTADOR RASPBERRY PI Y  
SOFTWARE LIBRE OPENHAB - HUÁNUCO 2020”**

---

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS E INFORMÁTICA**

**AUTOR: Cordero Díaz, Armando Isaac**

**ASESOR: López De La Cruz, Edgardo Cristiam Ivan**

**HUÁNUCO – PERÚ**

**2021**

# U

### TIPO DEL TRABAJO DE INVESTIGACIÓN:

- Tesis ( X )
- Trabajo de Suficiencia Profesional ( )
- Trabajo de Investigación ( )
- Trabajo Académico ( )

**LÍNEAS DE INVESTIGACIÓN:** Administración de redes y soporte de tecnologías de la información.

**AÑO DE LA LÍNEA DE INVESTIGACIÓN** ( 2018-2019 )

### CAMPO DE CONOCIMIENTO OCDE:

**Área:** Ingeniería, Tecnología

**Sub área:** Ingeniería eléctrica, Ingeniería electrónica

**Disciplina:** Ingeniería de sistemas y comunicaciones

### DATOS DEL PROGRAMA:

Nombre del Grado/Título a recibir: Título Profesional de Ingeniero de sistemas e informática

Código del Programa: P06

Tipo de Financiamiento:

- Propio ( X )
- UDH ( )
- Fondos Concursables ( )

### DATOS DEL AUTOR:

Documento Nacional de Identidad (DNI): 09449819

### DATOS DEL ASESOR:

Documento Nacional de Identidad (DNI): 40394603

Grado/Título: Magister en ciencias de la educación

Código ORCID: 0000-0001-9815-7708

### DATOS DE LOS JURADOS:

N°	APELLIDOS Y NOMBRES	GRADO	DNI	Código ORCID
1	Marin Sevillano, Richard	Doctor en ciencias de la educación	44280832	0000-0002-7604-5200
2	Huapaya Condori, Freddy	Doctor en ingeniería informática y de automatización	22506586	0000-0003-4783-3803
3	Baldeon Canchaya, Walter Teofilo	Maestro en ingeniería de sistemas e informática con mención en: gerencia de sistemas y tecnologías de información	22512084	0000-0002-4270-073X

# D

# H



# UNIVERSIDAD DE HUÁNUCO

## Facultad de Ingeniería

P. A. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

---

### ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO(A) DE SISTEMAS E INFORMÁTICA

En la ciudad de Huánuco, siendo las 18:00 horas del día 24 del mes de noviembre del año 2021, mediante la plataforma Google Meet, en cumplimiento de lo señalado en el Reglamento de Grados y Títulos de la Universidad de Huánuco, se reunieron los **Jurados Calificadores** integrado por los Docentes:

- Dr. Richard Michel Marín Sevillano      PRESIDENTE
- Dr. Freddy Ronald Huapaya Condori      SECRETARIO
- Mg. Walter Teófilo Baldeón Canchaya      VOCAL

Nombrados mediante la Resolución N° 1546 -2021-D-FI-UDH, para evaluar la Tesis intitulada: **"DISEÑO Y PROPUESTA DE UN SISTEMA DE CONTROL DE LA ILUMINACIÓN DEL LABORATORIO DE CÓMPUTO DE LA UNIVERSIDAD NACIONAL HERMILIO VALDIZAN BAJO LA ARQUITECTURA DE COMPUTADOR RASPBERRY PI Y SOFTWARE LIBRE OPENHAB - HUÁNUCO 2020"**, presentado por el (la) **Bach. CORDERO DIAZ, ARMANDO ISAAC**, para optar el Título Profesional de Ingeniero(a) de Sistemas e Informática.

Dicho acto de sustentación se desarrolló en dos etapas: exposición y absolución de preguntas: procediéndose luego a la evaluación por parte de los miembros del Jurado.

Habiendo absuelto las objeciones que le fueron formuladas por los miembros del Jurado y de conformidad con las respectivas disposiciones reglamentarias, procedieron a deliberar y calificar, declarándolo(a) **APROBADO** por **UNANIMIDAD** con el calificativo cuantitativo de **18** y cualitativo de **MUY BUENO** (Art. 47).

Siendo las 19:23 horas del día 24 del mes de noviembre del año 2021, los miembros del Jurado Calificador firman la presente Acta en señal de conformidad.

  
\_\_\_\_\_  
Presidente

  
\_\_\_\_\_  
Secretario

  
\_\_\_\_\_  
Vocal

## DEDICATORIA

“A mis padres Olga y Héctor por haberme forjado por la persona que soy hoy en día; muchos de las cosas que he logrado se las debo a ellos, en donde incluyo este. Me formaron con reglas, pero al mismo tiempo con ciertas libertades y siempre me motivaron para que pudiera alcanzar mis sueños.

Gracias mamá Olga y papá Héctor”.

## **AGRADECIMIENTO**

Agradezco a Dios por bendecirnos la vida, por guiarnos a lo largo de nuestra existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad.

Gracias a mis padres: Héctor y Olga, a mis hermanos Yanet, Christian y Abraham, a mi señora Santiago y a mi hijo Ángel por ser los principales promotores de nuestros sueños, por confiar y creer en nuestras expectativas, por los consejos, valores y principios que nos han inculcado.

# ÍNDICE

DEDICATORIA .....	II
AGRADECIMIENTO .....	III
ÍNDICE.....	IV
ÍNDICE DE TABLAS .....	VI
ÍNDICE DE FIGURAS.....	VII
RESUMEN .....	IX
ABSTRACT.....	X
INTRODUCCIÓN .....	XI
CAPÍTULO I.....	12
LÍNEA DE INVESTIGACIÓN .....	12
1.1. Línea de investigación:.....	12
1.2. Descripción del problema: .....	12
1.3. Formulación del problema: .....	13
1.4. Justificación del estudio:.....	13
1.4.1. Justificación Práctica .....	13
1.4.2. Justificación Teórica .....	14
1.5. Propuesta de solución y alcance: .....	14
1.6. Objetivos de la investigación: .....	15
1.6.1. Objetivo principal: .....	15
1.6.2. Objetivos secundarios:.....	15
CAPÍTULO II.....	16
FUNDAMENTACIÓN TEÓRICA.....	16
2.1. Antecedentes de la investigación: .....	16
2.1.1. A nivel internacional:.....	16
2.1.2. A nivel nacional:.....	19

2.1.3. A nivel local:.....	21
2.2. Marco teórico:.....	23
2.2.1. Sistema de control: .....	23
2.2.2. Internet de las cosas:.....	24
2.2.3. Raspberry Pi 3 B+:.....	38
2.3. Glosario de términos: .....	40
CAPÍTULO III.....	42
METODOLOGÍA.....	42
3.1. Metodología de desarrollo: .....	42
3.1.1. Fase 1: Analizar Requerimientos .....	42
3.1.2. Fase 2: Desarrollar Diseño Lógico.....	42
3.1.3. Fase 3: Desarrollar Diseño Físico.....	43
3.1.4. Fase 4: Probar, optimizar y documentar diseño.....	43
3.2. Herramientas de desarrollo: .....	43
CAPÍTULO IV.....	45
DESARROLLO E IMPLEMENTACIÓN .....	45
4.1. Desarrollo e implementación: .....	45
4.1.1. Fase 1: Análisis de requerimientos: .....	45
4.1.2. Fase 2: Desarrollar el diseño lógico:.....	45
4.1.3. Fase 3: Desarrollar el diseño físico:.....	53
4.1.4. Fase 4: Probar, optimizar y documentar el diseño:.....	62
4.2. Resultados: .....	88
REFERENCIAS .....	94
CONCLUSIONES .....	95
ANEXOS.....	96

## ÍNDICE DE TABLAS

Tabla 1: Herramientas de desarrollo Hardware .....	44
Tabla 2: Herramientas de desarrollo Software.....	44
Tabla 3: Requerimientos funcionales.....	45
Tabla 4: Configuración Router0 (Fuente: Elaboración propia).....	51
Tabla 5: Asignando Protocolo Estático al Router0.....	52
Tabla 6: Asignando Protocolo Estático al Router1 .....	52
Tabla 7: Programación en Phyton.....	68
Tabla 8: Código en Java script del Caso I .....	71
Tabla 9: Código en Java script del Caso II .....	71
Tabla 10: Código en Java script del caso III .....	72
Tabla 11: Código en Java script del caso IV .....	72
Tabla 12: Código en Java script del caso V .....	73
Tabla 13: Código en Java script del caso VI.....	74
Tabla 14: Código en Java para el Caso Ejemplo.....	82
Tabla 15: Código en Java para el Caso I.....	84
Tabla 16: Código en Java para el Caso II.....	85
Tabla 17: Código en Java para el Caso III.....	86
Tabla 18: Código en Java para el Caso Integro.....	87



## ÍNDICE DE FIGURAS

Figura 1: Sistema de Control (Fuente: Elaboración propia) .....	24
Figura 2: Niveles de inteligencia de los dispositivos. ....	30
Figura 3: Capas de funcionamiento del Cloud y Edge Computing .....	38
Figura 4: Raspberry Pi 3B+ (Fuente: Elaboración propia) .....	39
Figura 5: Modelo OSI de Redes (Fuente: Elaboración propia) .....	42
Figura 6: Red Lan (Fuente: Elaboración propia) .....	47
Figura 7: Consola para probar conectividad (Fuente: Elaboración propia) ...	48
Figura 8: Consola Clic Router (Fuente: Elaboración propia).....	50
Figura 9: Red WAN Huánuco - Lima (Fuente: Elaboración propia) .....	53
Figura 10: Red Alámbrica Básica.....	54
Figura 11: Dispositivos IoT .....	55
Figura 12: Red Inalámbrica Básica .....	56
Figura 13: Escritorio de Pc - Navegador .....	57
Figura 14: Escritorio Pc - Monitor IoT .....	57
Figura 15: Red Inalámbrica Básica con Móvil .....	58
Figura 16: Red Inalámbrica con Autenticación.....	59
Figura 17: Controles para iniciar sesión.....	61
Figura 18: Escritorio del Servidor - Monitor IoT.....	61
Figura 19: Dispositivo Lámpara con Autenticación .....	64
Figura 20: Dispositivo WebCam con Autenticación .....	65
Figura 21: Escritorio de la WebCam - Monitor IoT .....	65
Figura 22: Red con el Dispositivo Raspberry .....	67
Figura 23: Consola de programación del Dispositivo Raspberry. ....	67
Figura 24: Ejecución directa del dispositivo con Alt + Clic .....	68
Figura 25: Dispositivos Adicionales. ....	69

Figura 26: Etiquetas activadas de las conexiones entre los dispositivos. ....	70
Figura 27: Red perimetral con Dispositivos IoT .....	75
Figura 28: Dispositivos de red sin conexión.....	76
Figura 29:Dispositivos de red con conexión .....	77
Figura 30: Escritorio de MiPC y el navegador con la página de Cisco.....	78
Figura 31: Escritorio del Servidor y la Aplicación Monitor IoT .....	79
Figura 32: Interfaz para crear Usuario y Contraseña.....	79
Figura 33: Ventana de los dispositivos IoT en la Aplicación Monitor IoT .....	80
Figura 34: Dispositivos de red con sus respectivas conexiones. ....	80
Figura 35: Disposición de los dispositivos IoT según ubicación. ....	81
Figura 36: Consola de programación con Java .....	82
Figura 37: Consola con el Código en Java para el Caso Ejemplo .....	83
Figura 38: Panel de control de openhab2 en localhost.....	88
Figura 39: Panel control para agregar el foco .....	88
Figura 40: Agregar el binding de Xiaomi.....	89
Figura 41: Encontró el binding de Xiaomi .....	89
Figura 42: Canales, en Modo de Switch .....	89
Figura 43: Propiedades del foco inteligente.....	90
Figura 44: Canales del Foco.....	90
Figura 45: Menú de Items .....	90
Figura 46: Id del Foco Yeelight. ....	91
Figura 47: Item asociado a un canal del Foco .....	91
Figura 48: Menu de Enlace del Item .....	91
Figura 49: Panel principal del Item.....	92
Figura 50: Modelos soportados por el Binding de Xiaomi.....	92
Figura 51: Panel finalizado del Foco de Xiaomi .....	92
Figura 52: Foco encendido mediante el Servidor OpenHab .....	93

## RESUMEN

Esta investigación de tipo tecnológica se orientó en diseñar y proponer un Sistema de Control de Iluminación para los Laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizán bajo la arquitectura de computador Raspberry Pi y el software libre OpenHab. Ante esta pandemia, y las limitaciones de poder aplicar la investigación in situ se optó por emular todo el proceso y proponer una propuesta de solución para el control de la iluminación de los laboratorios bajo software libre, empleando pequeñas arquitecturas de servidores como la computadora Raspberry Pi, y bajo la tecnología del Internet de las cosas, así mismo para los implementos a nivel de hardware se utilizaron bombillas inteligentes de la marca Xiaomi y Phillips para las pruebas correspondientes, si bien es cierto fue un trabajo a escala, podríamos afirmar que el procedimiento se podría aplicar a cualquier escala, a cualquier habitación, laboratorio o entorno.

La metodología empleada para el diseño fue la del Top-Down basada en las siguientes fases: Análisis de Requerimientos, diseño Lógico, diseño Físico, pruebas, documentar diseño, implementar y probar la red y monitorear y Optimizar la Red. Estas fases permitieron identificar los requerimientos de la red a implantar, también realidad el diseño lógico y físico mediante el uso del software Packet tracer, y la computadora Raspberry Pi conjuntamente con las bombillas inteligentes para probar a escala el sistema de encendido automático.

Finalmente se logró la conexión y el funcionamiento a escala de red bajo un servidor OpenHab para el control del encendido de las bombillas inteligentes bajo la tecnología del Internet de las cosas.

Palabras clave: IOT, OpenHab, Raspberry Pi, Raspbian, Smart bulb, Domótica.

## ABSTRACT

This technological research was oriented to design and propose a Lighting Control System for the Computer Laboratories of the Universidad Nacional Hermilio Valdizán under the Raspberry Pi computer architecture and the free software OpenHab. Given this pandemic, and the limitations of being able to apply the research in situ, it was decided to emulate the whole process and propose a proposed solution for the control of the lighting of the laboratories under free software, using small server architectures such as the Raspberry Pi computer, and under the technology of the Internet of things, likewise for the hardware level implements were used smart bulbs of the Xiaomi and Phillips brand for the corresponding tests, although it is true that it was a scale work, we could say that the procedure could be applied to any scale, to any room, laboratory or environment.

The methodology used for the design was Top-Down based on the following phases: Requirements Analysis, Logical design, Physical design, testing, document design, implement and test the network and monitor and Optimize the Network. These phases allowed to identify the requirements of the network to be implemented, also the logical and physical design through the use of Packet tracer software, and the Raspberry Pi computer together with the smart bulbs to test at scale the automatic ignition system.

Finally, the connection and operation at network scale was achieved under an OpenHab server to control the lighting of smart bulbs under the Internet of Things technology.

Keywords: IOT, OpenHab, Raspberry Pi, Raspbian, Smart Bulb, Domótica.

# INTRODUCCIÓN

La investigación inicia con el proceso de análisis y el diagnóstico situacional en la cual amerita la redacción del siguiente problema: ¿De qué manera se diseña y se propone un Sistema de Control de Iluminación para los Laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizán bajo la arquitectura de computador Raspberry Pi y el software libre OpenHab?, siendo este el problema fundamentado en el Capítulo I, en la cual también se da a conocer el objetivo principal de Diseñar y proponer un Sistema de Control de Iluminación para los Laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizán bajo la arquitectura de computador Raspberry Pi y el software libre OpenHab. En este capítulo también se da a conocer la propuesta de crear una red a escala y emular el funcionamiento de un sistema de encendido de luces para los laboratorios de Cómputo de la Universidad Hermilio Valdizán de Huánuco. En el Capítulo II se da conocer los antecedentes y el marco teórico que comprende los temas Internet de las Cosas, Servidores Domótica con software libre, sistemas operativos Raspbian, Hardware de computadoras con Raspberry Pi, manejo de dispositivos inteligentes como bombillas y el empleo de Edge computing o computación en el borde. En el Capítulo III se da a conocer la metodología empleada para el diseño y propuesta de la red siendo la metodología Top Down propuesta por Cisco para el desarrollo de esta investigación, en el mismo capítulo se dan a conocer las fases empleadas y cuáles de ellas se utilizaron en la presente investigación. Finalmente, en el Capítulo IV se dan a conocer los resultados de la implementación a escala de la red, mostrando los diagramas y resultado de la ejecución de la emulación de la red empleando el software Packet tracer, además fotografías con el resultado de la red implementada a escala realizando la conexión del servidor con OpenHab y el encendido y control de la bombilla inteligente, en el mismo capítulo se anexa todo el procedimiento, pasos, secuencias para realizar la instalación desde cero y poder aplicar posteriormente en cualquier red física y real.

# CAPÍTULO I

## LÍNEA DE INVESTIGACIÓN

### 1.1. Línea de investigación:

La investigación se basa en Línea: Tecnologías de la información y comunicación, cuyo objetivo es: Aprovechar las herramientas y tecnologías de las computadoras y comunicaciones, así como las nuevas tecnologías emergentes que apuntan hacia la eficiencia, automatización y seguridad de los procesos en las variadas áreas del desarrollo humano generando así impactos productivos en las organizaciones.

La investigación comprende los temas Internet de las Cosas (Domótica, manejo de dispositivos inteligentes como bombillas), Redes (Servidores, switch, accesspoint), Software libre (S.O. Raspbian, OpenHab, Packet tracer), Hardware de computadoras con Raspberry Pi, y el empleo de Edge computing o computación perimetral.

### 1.2. Descripción del problema:

Las sistematización y optimización de los procesos en relación a calidad y precio viene siendo un tema importante en el desarrollo e investigación en el área de las ciencias informáticas, conjuntamente el empleo de software libre y tecnologías emergentes que permiten soluciones prácticas y efectivas.

Cada vez las instituciones emplean la tecnología para la sistematización y optimización de sus procesos, tareas, actividades, reduciendo costos, mejorando la eficacia y control de los mismos, para ello se necesita el conocimiento teórico y técnico y las herramientas adecuadas. La Universidad Nacional Hermilio Valdizán ubicada en la Av. Universitaria N° 601-607, Pillco Marca 10003, brinda las carreras profesionales en sus diferentes especialidades y con la infraestructura adecuada para el desarrollo de las laborales académicas y administrativas; la universidad cuenta con un laboratorio de cómputo ubicado en el primer piso del primer pabellón del campus universitario; en cuanto a las dimensiones y detalles ver el plano adjunto en la sección anexos.

El laboratorio de Cómputo presenta los siguientes inconvenientes: escasa iluminación en las separaciones de los ambientes, la falta de control en el tema de consumo de energía, en algunos casos las luces se quedan prendidas hasta el día siguiente, algunas bombillas están quemadas y no son reemplazadas, se ha pretendido reemplazarlas por las bombillas con sensores, de activarse al detectar movimiento, pero ha fracasado ya que en este laboratorio hay una afluencia masiva de personas en diferentes horarios del día; estos problemas han conllevado a que los usuarios del laboratorio de Cómputo se sientan incómodos por la falta de una buena iluminación, y por parte de la administración del laboratorio la pérdida de tiempo en recurrir a cada Switch para el prendido y apagado de cada bombilla, y en algunos casos agregando más puntos de energía eléctrica o tendiendo cableado adicional para la iluminación de más áreas, conllevando al desorden y a la falta de estética en la instalación eléctrica del laboratorio.

Ineficacia en los encendidos y apagados de los fluorescentes debido a que se realizan manualmente.

### **1.3. Formulación del problema:**

¿De qué manera se diseña y se propone un Sistema de Control de Iluminación para los Laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizán bajo la arquitectura de computador Raspberry Pi y el software libre OpenHab?

### **1.4. Justificación del estudio:**

#### **1.4.1. Justificación Práctica**

Con el estudio se da a conocer una propuesta tecnológica a escala de cómo controlar dispositivos inteligentes mediante una red edge computing usando software libre sin la necesidad de recurrir a soluciones costosas, la solución reside en el diseño de la automatización del prendido y apago de las luces de los laboratorios de la Universidad Nacional Hermilio Valdizán.

#### **1.4.2. Justificación Teórica**

El aporte teórico se plasma en el diseño y el manual de instalación del sistema de control de iluminación, para que posteriormente en una futura investigación o la implementación de la misma por la universidad sea factible y eficiente siguiendo las instrucciones dadas en los productos obtenidos por la presente investigación.

#### **1.5. Propuesta de solución y alcance:**

La solución está planteada en base al uso de las tecnologías de la información y comunicación, específicamente en el área de software libre, redes de computadoras, e internet de las cosas; la idea central de la solución consiste en el diseño de una red perimetral (edge computing) bajo el enfoque del Internet de las cosas para establecer la comunicación entre dispositivos inteligentes: en este caso bombillas, y conectarlas a un servidor, para la propuesta se usa la computadora Raspberry pi, que permitirá control de la red y automatizando el prendido y apagado de las bombillas. El internet de las cosas propicia la solución de este problema, de poder conectar dispositivos inteligentes (bombillas, enchufes, sensores etc.) y permitir su conexión al borde de la red en otras palabras estableciendo una comunicación interna, local, sin hacer uso de la nube o el Internet, claro está que este modelo incrementa la velocidad de transmisión de data y la seguridad de los mismos.

La solución se inicia el diseño y en la aplicación a escala de una red de datos que controle la iluminación del laboratorio de cómputo bajo un servidor con software libre en este caso con el sistema operativo Linux y la solución OpenHab.

El diseño consistirá primero en la elaboración de un mapa de red usando el software Packet tracer detallando las conexiones y los componentes a usar; posteriormente la propuesta a escala recreará la red perimetral o la red de iluminación inteligente usando el computador Raspberry, una bombilla inteligente; más adelante ante la entrega de esta investigación a las autoridades de la institución, queda a disposición la implementación



a gran escala con la compra del servidor y bombillas inteligentes para todos los laboratorios de la universidad.

## **1.6. Objetivos de la investigación:**

### **1.6.1. Objetivo principal:**

Diseñar y proponer un Sistema de Control de Iluminación para los Laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizán bajo la arquitectura de computador Raspberry Pi y el software libre OpenHab.

### **1.6.2. Objetivos secundarios:**

- \* Diseñar la propuesta empleando el software Packet tracer, considerando los elementos de la red y las conexiones inteligentes.
- \* Elaborar la propuesta a escala, creando una red local, conectando el computador Raspberry Pi, y la Bombilla Inteligente.
- \* Instalar y configurar la distribución OpenHab para realizar la conexión lógica de la bombilla con el servidor y así poder controlar el encendido y apagado.
- \* Documentar los procedimientos de instalación configuración y control.

## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

#### 2.1. Antecedentes de la investigación:

##### 2.1.1. A nivel internacional:

Pajares, (2016) realizó la siguiente investigación: Estudio de la implantación de Internet de las Cosas, en las redes Logísticas de la Cadena de Suministro, Valencia. La investigación llegó a las siguientes principales conclusiones:

El sistema de gestión del tráfico utiliza sensores para conectarse con los puentes móviles del puerto y las carreteras cercanas, de tal forma que cuando llega un barco de considerable altura, el tráfico se desvía hacia otra ruta y el puente se abre. Para gestionar eficientemente todos los flujos de tráfico, los sensores del puente y la carretera están conectados a través de una red IPv4. Los datos capturados por los sensores se transmiten a la central "Port Road Center" (Centro de camino de puerto), el cual controla el tráfico rodante en el puerto, además, realiza un análisis para determinar la congestión y la velocidad del tráfico, y cualquier otro problema o incidente importante que surja. Este sistema de gestión del tráfico es importante para monitorizar, gestionar y controlar el tráfico en los puentes móviles y en las carreteras. Este sistema, como agregado también calcula el peso de los vehículos con el objetivo de obtener el volumen de tráfico en los 140 puentes que tiene el puerto. Este sistema de gestión de tráfico también permite una mejor integración con los usuarios, simplificando contactos directos e intercambio de información, haciendo más rápida y eficiente la toma de decisiones, de tal forma que los usuarios pueden decidir y elegir rápidamente la forma de transporte que más les conviene para el traslado de sus mercaderías, gracias a la información obtenida del puerto de Hamburgo. Con todo, este sistema permite mejorar y simplificar los procesos y ofrecer mejores servicios a las

compañías navieras. Por este motivo, el primer paso es la planeación y desarrollo de una infraestructura de red IPv4 interna.

Cuzme, (2015) realizó la investigación: El Internet de las Cosas y las consideraciones de Seguridad, Quito. La investigación llegó a las siguientes importantes conclusiones:

En este estudio, se identifica claramente que los desarrolladores e implementadores de esta tecnología tienen que tomar en cuenta la seguridad desde la concepción y el desarrollo de las herramientas IoT dependiendo del entorno en donde se van a instalar. Caso contrario, existe la posibilidad del riesgo en que los futuros sistemas sean susceptibles y vulnerables a ataques cibernéticos como los que existen en el presente, debido a que mayormente las herramientas no fueron creadas para pertenecer en un ambiente conectado al Internet. Se identificó una estructura de seguridad para entornos IoT, en donde pertenecen herramientas como sensores/ dispositivos, Gateway (puertas de enlaces), medios de transmisión, plataformas de captura de información y programas que permiten visualizar la información de forma estructurada. La siguiente estructura se lo presenta con una visión más específica orientada al dispositivo u objeto; Este esquema, involucra herramientas como circuitos integrados programables (microcontroladores), firmware, actuadores, sensores, conectividad, fuente de alimentación, sistema operativo y aplicaciones; cada una de estas herramientas en ambas estructuras desempeñan un papel muy relevante en la seguridad del IoT, resaltados en los respectivos niveles de seguridad que se deben tener en cuenta para cada etapa o nivel, de esta forma se puede alcanzar una seguridad integrada tanto en un entorno IoT como en un objeto. El Open Source Software/Hardware es una comunidad en línea de trabajo en equipo donde se encuentran involucrados estudiantes, profesionales, usuarios, clientes y miembros de

empresas de tecnologías, siendo el punto de encuentro adecuado para compartir diversos y variados conocimientos en diferentes áreas de tecnología permitiendo así integrar de forma variada conocimientos y experiencias acerca de soluciones y respuestas a problemas diversos; y presentar temas de discusión de los cuales obtener beneficios, todo ello construye una base de conocimiento al que todos pueden acceder para desarrollar varias y mejores alternativas de solución a ciertas vulnerabilidades detectadas en objetos, herramientas, dispositivos o sistemas.

Ramírez, (2018) realizó la investigación: Metodología para la implementación de IoT, Bogota. La investigación llego a las siguientes principales conclusiones:

Hoy en día, se pueden ejecutar implementaciones de IoT debido a que la mayoría de los avances tecnológicos necesarios para realizarlo ya se han hecho, y algunas compañías, fabricantes y las organizaciones han iniciado la implementación de una versión a pequeña escala. En la implementación del IoT no sólo se tomará en cuenta la instalación de tecnología en un entorno específico, es imprescindible halla las necesidades para desarrollar los servicios a partir de las mismas necesidades, teniendo en cuenta la concepción de los usuarios o clientes con el objetivo de encontrar qué población requiere IoT y para que lo necesita. Hay que tener en cuenta lo valioso que es la información para las instituciones y empresas. Es decir, para el tipo de institución o empresa se debe evaluar detalladamente que tipo de plataforma y seguridad será utilizada para la implementación del IoT evitando que datos importantes puedan ser expuestos. Los dispositivos y plataformas que se han de utilizar para la implementación de un IoT deben de garantizar la interoperabilidad e interconectividad entre ellas facilitando así su eficiente y correcto funcionamiento. Hoy en día, diferentes fabricantes ofrecen en su línea de ventas estas cualidades las

que contribuyen al reducido costo para la implementación. Haciendo uso del IoT podemos contribuir a la disminución de la contaminación por CO<sub>2</sub> o huella de carbono. Mayormente se obtiene una reducción de costos cuando se implementa la arquitectura y la tecnología de forma simultánea en comparación a cuando se implementa la tecnología a una arquitectura ya establecida. El IoT es una opción de negocio y a su vez una herramienta para aumentar la productividad y comodidad de los empleados incrementando así las ganancias significativamente. Es importante la participación de los funcionarios e involucrados en la identificación de las necesidades para obtener resultados reales y satisfactorios.

#### **2.1.2. A nivel nacional:**

Ttacca, (2017) realizó la investigación: Diseño de una Red Fog basado en el Internet de las Cosas para monitorear la contaminación en la Bahía del Lago Titicaca, Puno. La investigación llego a las principales siguientes conclusiones: El procedimiento para diseñar la red FOG computing con IoT se realizó simulando en programas diseñados para desarrollar una red de comunicaciones que apuntan a los protocolos de comunicación del Internet de las cosas. Para el desarrollo de la red en una red FOG computing basado en el Internet de las cosas se llegó a utilizar un programa de simulación para elaborar una red con características que apuntan al IoT la cual tiene características específicas, como conexiones wifi, asignación de direcciones IPv4, con resultados satisfactorios. Se desarrolló un prototipo básico con capacidad de transmisión de datos hacia un servidor e independencia energética, para monitorear las condiciones ambientales y del entorno de las aguas del Lago Titicaca que utilizan los protocolos de comunicación IoT, y costos muy reducidos para realizar el remplazo en caso de averías, un fácil mantenimiento, además, se adaptan a las difíciles

condiciones ambientales del Lago Titicaca. Al implementar motas en la Laguna del Titicaca se monitorea y procesa en tiempo real los datos captados por los sensores de temperatura y el Ph del agua, esta información es enviada hacia el servidor que comparte la información de la cual se puede acceder a los resultados ingresando al sitio web “[www.iottiticaca.com](http://www.iottiticaca.com)” en el cual es posible visualizar los resultados de las estadísticas como los parámetros ambientales con actualizaciones programadas en un cierto intervalo de tiempo, además se puede visualizar el historial de todas las incidencias y sucesos. Los valores de estos datos representan cantidades referenciales de las diferentes condiciones ambientales puesto que captan los datos partiendo de una muestra en un área muy pequeña las Motas de sensores.

Palacios y Rodríguez, (2017) realizó la investigación: La Contribución de la Tecnología frente a la Inseguridad ciudadana integrando seguridad con los aportes del Internet de las Cosas, Lima. La investigación llegó a las siguientes principales conclusiones:

El objetivo es introducir es una solución tecnológica que apunta a la disminución de la inseguridad ciudadana a través de la ejecución del paradigma del IoT, la cual hoy en día es la tecnología que puede nos ofrece mejores resultados frente a la inseguridad ciudadana; Ya es conocido los beneficios que dicha tecnología hasta ahora ha venido ofreciendo con la integración a los objetos que están en nuestro entorno a través del Internet. En ese sentido, se especifican los siguientes requerimientos para el encaminado del CHIP GPS para el monitoreo y control de los objetos.

Leyva, (2017) realizó la investigación: Solución Domótica utilizando IoT y Arduino para mejorar el control de accesos de seguridad residencial en la Urb. Covicorti, Trujillo. La investigación llegó a las siguientes principales conclusiones:

Se alcanzó acortar el tiempo de activación de los periféricos de accesos de seguridad en la residencial en un 99.20%. Se alcanzó acortar el número de vulnerabilidad en los accesos de seguridad de la residencial en un 43.75%. Se alcanzó acortar el tiempo de comprobación de los accesos de seguridad en la residencial en un 98.38%.

Con respecto a la disminución de costos como proveedor, se alcanzó los siguientes resultados: VAN = S/ 45891, C/B = S/ 1.36, TIR = 104%. Se pronostica que el capital se recuperará en 7 meses y 9 días. Con la solución domótica utilizando el Arduino y el Internet de las cosas se alcanzó una mejora significativa en el control de accesos de seguridad en la residencial de la Urb. Covicorti, Trujillo.

### **2.1.3. A nivel local:**

Jesús, (2018) realizó la investigación: Modelo conceptual del Internet de las cosas para la Educación Superior en la UNHEVAL, Huánuco. La investigación llegó a las siguientes principales conclusiones:

En relación al primer objetivo se identificó las diversas aplicaciones de la internet de las cosas que podrían incorporarse dentro de la UNHEVAL, en el análisis se pudo evidenciar que el avance y desarrollo del internet de las cosas genera nuevas formas, estilos, tipos y procesos de educación, siendo estas las que revolucionan el proceso enseñanza-aprendizaje, de las cuales podemos identificar las plataformas a medida, desarrolladas por diversas instituciones de educación superior, empresas, entre otras, que utilizan el internet de las cosas para conectar mediante red sus distintos dispositivos; así como también las plataformas de código abierto que se utilizan la mayoría de las universidades ya que es de mayor accesibilidad para los usuarios, pudiendo de esta manera hacer cambios según las necesidades de los usuarios, entre las cuales se

encuentran las plataformas DOKEOS, CLAROLINE y MOODLE. En relación al segundo objetivo como se puede visualizar en el análisis de las plataformas propuestas DOKEOS, CLAROLINE y MOODLE, para la evaluación de la adaptabilidad del internet de las cosas en la enseñanza a los futuros profesionales de la UNHEVAL se demuestra que al aplicar herramientas mediante el internet de las cosas se fomenta la comunicación entre docente y alumnos, fomenta el debate y la discusión, y la accesibilidad del alumno por los temas y contenidos desarrollados en las clases. Así pues al aplicar el juicio de expertos para la evaluación de la métrica para cada una de las plataformas propuestas podemos apreciar que la de mayor aceptación es la plataforma MOODLE, la cual obtuvo mayor puntuación siendo esta 2.47 aplicando el modelo propuesto por (Lovellette, 1999), quien califica como la plataforma más eficiente para el proceso de enseñanza-aprendizaje entre el docente y el alumno, así como al realizar el análisis de medias para las tres plataformas se obtuvo que la plataforma MOODLE tiene un promedio de 7.40, y la plataforma CLAROLINE presenta un promedio de 6.70, y la plataforma DOKEOS presenta un promedio de 7.01; de dichos resultados se observa que existe una predominancia hacia la plataforma MOODLE para el proceso de aprendizaje-enseñanza. Siendo así que en relación al tercer objetivo se propuso el modelo conceptual de una plataforma virtual, tomando el enfoque del internet de las cosas, el cual mejorará la enseñanza superior universitaria de la UNHEVAL, ya que al evaluar las características académicas, tecnológicas y administrativas de la plataforma MOODLE, el objetivo predominante es fortalecer el proceso de enseñanza aprendizaje entre docente alumno, como la interacción constante y monitoreo permanente de los conocimientos adquiridos durante el desarrollo de los cursos que se imparten a través de las plataformas virtuales. Se concluye así que la adaptación del internet de las cosas mediante la implementación



de una plataforma virtual es efectiva ya que permitirá al alumno poder utilizar todas herramientas tecnológicas con las que cuenta actualmente la Universidad Nacional Hermilio Valdizán.

## **2.2. Marco teórico:**

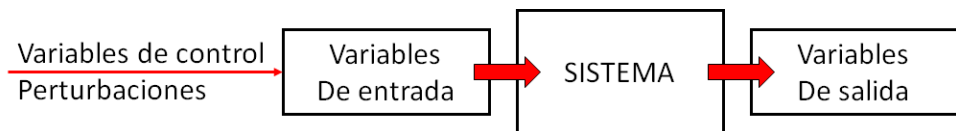
### **2.2.1. Sistema de control:**

Un sistema de control es aquel sistema que se define conceptualmente como una entidad que recibe unas interacciones del exterior o variables de entrada, y cuya respuesta a estas interacciones provenientes del exterior son las llamadas variables de salida. Las interacciones externas al sistema se clasifican en dos tipos que son, las perturbaciones que no se pueden manipular es decir que no es posible algún tipo de control y las variables de control, que se pueden manipular.

Entre los tipos de sistemas, tenemos el concepto de sistema de control. Un sistema de control es el que presenta una serie de elementos que influyen en el funcionamiento del sistema. El objetivo de un sistema de control es alcanzar, mediante el procesamiento de las variables de control, un dominio sobre las variables de salida, de tal modo que estas consigan unos valores anticipados (consigna). Un sistema de control ideal puede alcanzar su objetivo cumpliendo los siguientes requisitos:

- \* Garantizar el equilibrio y, específicamente, ser robusto frente a errores y perturbaciones en los modelos.
- \* Ser tan eficiente como sea posible, según ciertos criterios preestablecidos. Generalmente estos criterios consisten en que la realización del control sobre las variables de entrada sea aplicable, evitando comportamientos irreales y bruscos.
- \* Ser fácilmente de implementar y cómodo de manipular en tiempo real con ayuda de un sistema informático (Computadora, celular, Tablet, etc.).

- \* Un sistema de control está compuesto por los siguientes elementos básicos los cuales permiten su manipulación:
  - \* Sensores. Son los que captan las interacciones exteriores para luego conocer los valores de las variables de entrada.
  - \* Controlador. Es aquel que, conociendo los valores captados por los sensores y la directiva impuesta, calcula la acción que debe desarrollarse para modificar las variables de control según cierta estrategia.
  - \* Actuador. Es aquel mecanismo que aplica la acción impuesta por el controlador y que modificara las variables de control. (Upcommons, s.f)



*Figura 1: Sistema de Control (Fuente: Elaboración propia)*

### **2.2.2. Internet de las cosas:**

Para Domodesk (2019) define el Internet de las cosas como la consolidación a través de la red de redes de una "red" que aloja una gran cantidad de objetos o dispositivos, es decir, poder interconectar a esta todas las cosas de este mundo como pueden ser electrodomésticos, vehículos, dispositivos mecánicos, o simplemente objetos tales como muebles, calzados, maletas, biosensores, dispositivos de medición, o cualquier objeto que podemos imaginar.

Para Gracia (2019) La definición de IoT puede ser la colección e interconexión entre dispositivos y objetos a través de una red (del tipo intranet o Internet), de dónde todos ellos pueden ser monitoreados e interactuar. Respecto al tipo de dispositivos u objetos, pueden ser cualquiera, desde dispositivos mecánicos y sensores hasta objetos de uso cotidiano como pueden ser la refrigeradora, el calzado, correa, reloj o la ropa. Cualquier objeto podría ser conectado al internet e interactuar sin la necesidad de la manipulación humana, el alcance por tanto es una relación de

interacción de máquina a máquina, o llamado también como una interacción de dispositivos M2M que significa (machine to machine).

El término "Internet de las cosas", describe el creciente y amplio conjunto de elementos digitales, los cuales, hoy en día son miles de millones y que funcionan mediante las redes de escala potencialmente global. A diferencia del Internet normal (de las personas), los elementos IoT están compuestos solo de sensores inteligentes y otros dispositivos. Entre sus alcances son principalmente la captación de datos obtenidos de sensores remotos como, por ejemplo, en las plataformas petroleras, la recolección de datos meteorológicos y el control de termostatos inteligentes. (Packard, 2019).

Mayormente en los avances tecnológicos el objetivo que se pretende alcanzar es hacer más cómodas nuestras vidas en nuestro entorno y así como llegar a una mayor seguridad en diferentes áreas. Como por ejemplo: Supongamos una refrigeradora que es capaz de indicarnos cuando baja la temperatura o avisarnos que algún alimento se está pasando de tiempo o ha caducado, o un escritorio que nos deje constancia en donde se ha dejado cada objeto de la oficina, o que pudiésemos saber dónde se ubica cada cosa, que esté en nuestro alcance, en cada momento, o para controlar todo nuestro hogar u oficina desde un móvil inteligente, tablet o PC (desde el inicio o puerta de entrada, hasta la cadena del inodoro), o que estemos avisados de la concentración de bacterias y virus en el aire mediante una pulserita para alérgicos con sensores, o miles de entornos por descubrir. Estos son algunos escenarios de todo lo que podríamos alcanzar con el Internet de las cosas; Utilizando la domótica y control para dar a lugar a unas nuevas formas de relación M2M, a la inmótica, a la domótica y al más puro control.

Según Evans (2017) el Internet de las cosas (IdC), llamado también en algunas ocasiones "Internet de los objetos", lo modifica todo, incluyéndonos a nosotros mismos. Si bien tiene analogía a un

enunciado arriesgado, hay que darle la merecida importancia al impacto que el Internet ha tenido sobre varias áreas de nuestra vida como son la comunicación, educación, la empresa, la ciencia, la humanidad y el gobierno. Sin lugar a dudas el Internet es una de las tecnologías más poderosas e importantes en toda nuestra historia. Hoy en día debemos tener en cuenta que IoT representa la siguiente evolución del Internet, que será un gran salto en la capacidad para distribuir, analizar y reunir datos que podemos convertirlos en información útil, conocimiento y en el mejor de los casos en sabiduría. En este punto de vista, IoT se convierte inmensamente necesario e importante. Ya se están iniciando proyectos de IoT que prometen disminuir distancias entre pobres y ricos, mejorar justamente la planeación y distribución de los recursos en el mundo en el que vivimos para los más necesitados y ayudarnos a entender el planeta para que alcancemos ser menos destructivos y mucho más proactivos. Aun así, siempre existirán obstáculos que amenacen con retrasar el desarrollo del IoT, como la migración de IPv4 a IPv6, la aplicación de un conjunto de normativas en común y el desarrollo de fuentes alternativas de energía para miles e incluso de millones de pequeños sensores. Sin embargo, mientras que los gobiernos, las empresas, las áreas académicas y los organismos normativos trabajen conjuntamente para solucionar estos problemas, IoT continúa su camino. Por lo tanto, el objetivo de esta documentación es explicar, en términos simples y claros, de qué se trata IoT de forma tal que se pueda entender su alto alcance para modificar todo lo que hoy en día forma parte de nuestra realidad.

### **¿Es posible realmente conectar todo a la red?**

La respuesta correcta es que sí, no en este momento, pero si a corto plazo y más pronto de lo que pensamos. El principal obstáculo que está retrasando este alcance es la limitada cantidad de direccionamiento que nos brinda el actual protocolo IPv4, debido a la cantidad de dispositivos y objetos que se pretende interconectar.

Por tal razón es ya inminente la migración al protocolo IPv6, este nuevo protocolo de direccionamiento permitirá la evolución e implementación del Internet de las cosas ya que se podrán interconectar una cantidad ilimitada de cualquier objeto.

Una vez superado dicho inconveniente, y con la progresiva disminución de costos de esta tecnología a través del tiempo, no solamente será posible, si no imprescindible debido a sus múltiples utilidades y la gran cantidad de aplicaciones que se podrán ejecutar con el Internet de las cosas. Como, por ejemplo, ropa inteligente, aparatos implantados en nuestro organismo cuidando por nuestro bienestar, en las tiendas: bolsas de snacks advirtiéndonos su compra por contener altos valores en calorías, grasas, y azúcar cuidando nuestra salud.

### **¿De qué forma repercute el Internet de las cosas respecto a temas de seguridad o privacidad, en nuestras vidas?**

Una vez más nos topamos con la eterna dicotomía, si tenemos en cuenta los beneficios en cuanto a seguridad física o de bienes, sin lugar a dudas vamos hacia un entorno más seguro, si por el contrario nos damos cuenta que todo ese intercambio de información es muy particular, es decir basada en gustos, modas, tendencias, anhelos, deseos, pasiones...datos íntimos, aficiones, sentimientos, entre otros,... pues la seguridad se convierte en un "control en manos ajenas" (eufemísticamente LA NUBE) que nos hace cada vez más vulnerables (eufemísticamente BIG DATA).

Por lo tanto, apuntando hacia la seguridad de bienes y física en nuestra vida o de cualquier persona, ya sea sobre su persona, sobre su hogar, negocio, etc., desempeñara un papel muy importante, aumentando la seguridad en vehículos, viviendas, y todo tipo de bienes o simplemente sobre las personas.

En uno de los ejemplos anteriores, nosotros podemos saber en todo momento donde se ubican los objetos si todos y cada uno de dichos objetos contienen conexión a Internet y entonces disminuir prácticamente a cero de probabilidades la posibilidad de perderlo.

En otro caso, se vería aumentada de forma exponencial la seguridad en viviendas, edificios y fábricas, ya que con un ilimitado de objetos que contienen sensores y monitorizados no habrá incidencias que se desborden, fugas de agua, avisos por invasión, incendios, bases de datos que contiene con detalles todas las incidencias que han ocurrido en una vivienda, son algunas de las varias funciones que seremos capaces programar y ejecutar. Respecto a la seguridad personal se verá aumentada con dispositivos u objetos como relojes que pudiesen capturar el pulso y en consecuencia enviarnos mensajes de precaución y alerta, por ejemplo que en caso de un derrame cerebral o paro cardíaco automáticamente realizar una llamada a emergencias, dispositivos que nos alerten antes de sufrir una crisis por asma por ejemplo, todo ello y mucho más es lo que se podemos alcanzar con las nuevas tecnologías en informática y comunicaciones, como se le quiera llamar inmótica, domótica, Aml, Ambiance Intelligence, Smartcities, Objetos Inteligentes, Smart Grid, IoT o simplemente “un mundo feliz”.

Como una consecuencia de la superación de ser humano respecto al uso de herramientas para solucionar problemas, los avances tecnológicos son imparables, los beneficios sin excusas y el resto de la sociedad tendrá que poner cierto límite a factores que tienen que ver con los derechos, la privacidad y deberes, y teniendo en cuenta todo lo vinculado al reconocimiento de la dignidad de cada persona.

### **¿De qué capacidades está dotado el IoT?**

Comunicación y cooperación: los dispositivos y objetos mediante una interfaz de red serán capaces de conectarse al Internet y los servicios que ofrece y también entre ellos mismos, siendo así capaces de intercambiar, modificar y actualizar los datos que entre ellos comparten y establecer comunicaciones con los demás elementos de una red como los servidores.

Capacidad de direccionamiento: Estos dispositivos u objetos pueden ser configurados y localizables desde cualquier lugar de la red debido a que poseen sensores e interfaces de red.

Identificación: Estos dispositivos u objetos pueden ser identificados mediante tecnologías inalámbricas tales como NFC (Near Field Communication), RFID (Radio Frequency Identification), códigos de barra para la lectura mediante un lector óptico, u otras formas de identificar a un dispositivo u objeto en una red.

Localización: Al contener estos dispositivos sensores, interfaces de red y estando interconectados, se tiene en tiempo real conocimiento sobre su ubicación y estado físico, pudiendo así, conocer en qué lugar se encuentra en todo momento.

Actuación: Estos dispositivos u objetos al poseer sensores e instrucciones previamente configuradas serán capaces de reaccionar y según ello podrían manipular su entorno.

### **¿Cuáles son los niveles de inteligencia de estos dispositivos?**

Según los expertos en el tema pueden existir como mínimo 4 niveles de inteligencia:

Nivel 1: Identidad. Debido a que el objeto posee una dirección IP, será capaz de identificarse de manera única.

Nivel 2: Ubicación. Debido a que el objeto forma parte de una red, se podrá conocer dónde se encuentra dicho objeto o por dónde ha estado.

Nivel 3: Estado. Debido a que el objeto esta interconectado a una red y posee sensores que captan datos de su entorno, será capaz de comunicar el estado en que se encuentra, así como sus condiciones.

Nivel 4: Contexto. Debido a que el objeto posee sensores e instrucciones de reacción, el dispositivo será capaz de percibir el entorno del ambiente en que se encuentra.

Si lo revisamos hasta aquí nos puede parecer esta clasificación muy simple ya que está basada bajo la premisa de dispositivos u objetos un poco "tontitos", sin embargo, lo que hoy en día nos

ofrece el mercado hace más variada la clasificación, y podríamos adicionar niveles tales como:

Nivel 5: Criterio. El dispositivo u objeto se comunica, se identifica, se analiza su entorno, se ubica, toma decisiones y se ejecuta en función de su criterio.

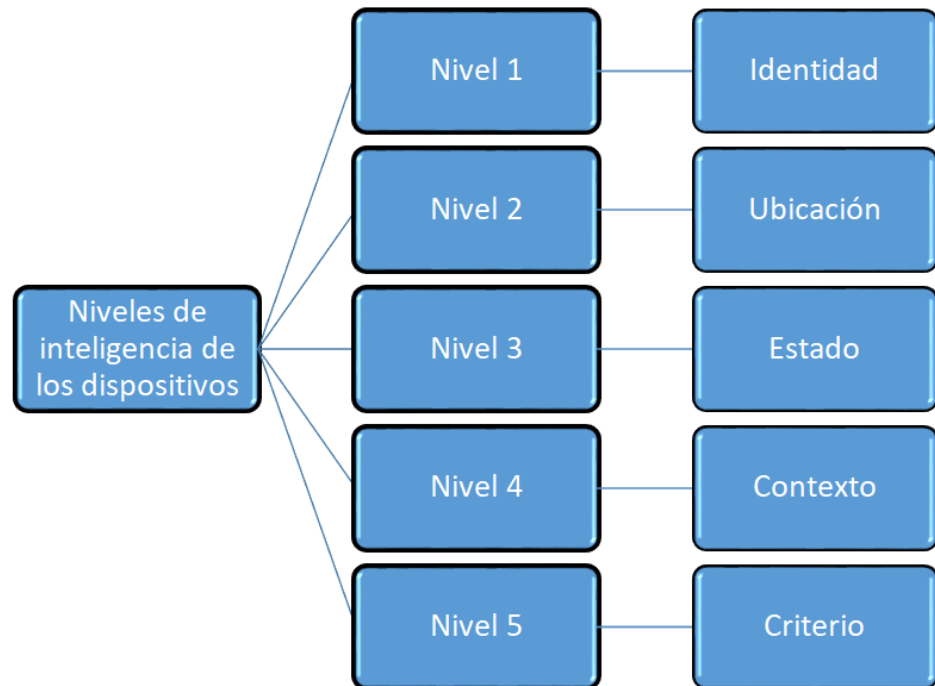


Figura 2: Niveles de inteligencia de los dispositivos.

### ¿Qué protocolos de comunicación utiliza el IoT?

Existe el dilema, cuando nos referimos de domótica, M2M o de automatización industrial, varias opciones de iniciativas se dan para intentar combinar un protocolo que sirva como estándar y de alguna forma tenga alcance la idiosincrasia de los sistemas que deseamos meter en el entorno.

Como por ejemplo, tenemos Bluetooth LTE, Zigbee (IEEE 802.15.4), 6LowPan, GSM (incluyendo sus correspondientes Gs), WiFi, u otras como SigFox y desde luego una expectativa a un mejor mundo de protocolo TCP/IP que conocemos que tiene capacidad de distribuir redes a nivel mundial, pero que no está planeado para introducir sus antiguas arquitecturas en sensores que deberían funcionar, como mínimo algunos, durante varios años, es decir capaces por si solos en el tema de carga (Viento,



solar, etc.) o auto capaces en el tema de consumo, esto sin tener en cuenta de ciento y un mil juicios propietarios lidiando el protagonismo para el gran negocio que se dará en tiempos cercanos.

### **¿Qué es y de que partes se compone un dispositivo M2M?**

Se ha visto con anterioridad que los dispositivos M2M son una tecnología adecuada en de momento de desarrollar el internet de las cosas realizaremos una breve descripción para entender mejor en qué consisten.

Los dispositivos u objetos M2M (machine to machine) como su propio nombre indica consiste en dispositivos capaces de obtener una comunicación tanto con otros dispositivos M2M como con el servidor.

Las partes que conforman un dispositivo M2M son las siguientes:

Dispositivos de gestión: Este grupo está conformado por todas aquellos dispositivos o máquinas cuyo trabajo principal consiste en gestionar datos (paneles de información, puntos de venta, alarmas en hogares, contadores, etc.)

Dispositivos M2M: Este grupo está conformado por aquellos dispositivos (módulos) que están interconectados de forma remota a cualquier sistema informático que capture datos y mantenga una comunicación con el servidor. En ciertos momentos también tendrán la capacidad para procesar información individualmente.

Servidor: Será aquel sistema que capture y envíe información de las máquinas y a la vez ejecute una administración eficiente de la misma información.

Red de comunicación: Consiste en medio físico por el cual viaja los datos e información pudiendo de ser de dos tipos: inalámbrico y cableado, siendo el inalámbrico el método recomendado debido a que por principio no tiene lógica implementar el Internet de las cosas mediante cableado a menos que sea algún dispositivo específico que necesite conexión alámbrica. (Domodesk, 2019).

### **¿Por qué es IoT tan importante?**

Actualmente, el internet de las cosas se ha transformado en una de las tecnologías más importantes de nuestro siglo. Ahora es posible establecer una comunicación más fluida entre las personas, procesos y objetos gracias a que podemos conectar objetos de la vida diaria como vehículo, aparatos de cocina, termostatos, monitores de bebés al Internet mediante los dispositivos integrados.

Debido a la informática de costos reducidos, el big data, la nube, la analítica de datos y las tecnologías móviles, los dispositivos físicos pueden captar datos y compartir datos con un mínimo de intervención humana. Hoy en día y en nuestro mundo hiperconectado, los sistemas digitales pueden supervisar, grabar, modificar y ajustar cada variación entre los objetos conectados. Permitiendo una cooperación entre este mundo físico que se combina con el mundo digital.

### **¿Qué tecnologías hacen posible IoT?**

Recientemente, una serie de avances en diversas tecnologías de la información y comunicación han hecho realidad el IoT. No obstante, la idea de IoT existe desde hace mucho tiempo:

El acceso a tecnología de sensores de bajo coste y baja potencia. La tecnología del IoT es más factible para ser fabricado, debido a los sensores asequibles y fiables.

Conectividad. Hoy en día es más fácil conectar uno o varios sensores a la nube y a otros objetos para conseguir una transmisión de datos mucho más eficiente.

Plataformas de Cloud Computing. Tanto los consumidores como las empresas pueden acceder a una única infraestructura que necesitan para ampliar la capacidad sin tener que gestionarlo todo en diferentes infraestructuras gracias al aumento de la disponibilidad de plataformas en la nube.

Analítica de datos y Machine learning. Con los avances alcanzados gracias a las nuevas tecnologías en analítica de datos y machine learning, conjuntamente con el acceso a grandes cantidades de datos e información almacenados en la nube, las empresas pueden captar y almacenar información de forma más rápida, segura y más sencilla. Los avances de estas tecnologías interconectadas siguen ampliando los límites del IoT; Estas tecnologías están constantemente retroalimentadas por los datos generados por IoT. Inteligencia artificial (IA) conversacional. Los nuevos alcances en redes neuronales han generado el avance de las lenguas naturales (NLP) en los objetos y dispositivos IoT. Como por ejemplo los asistentes artificiales Cortana, Alexa y Siri, y los han convertido en dispositivos eficientes para el uso doméstico. (Oracle, 2019)

### **Edge Computing (Computación Perimetral):**

La computación perimetral es un tipo de computación que ocurre en un entorno físico del usuario, de la fuente de datos, o próxima a ellas. Al aplicar servicios de computación cerca de estos entornos, los usuarios obtienen servicios más confiables y rápidos, y las empresas e instituciones aprovechan la flexibilidad de la computación en la nube híbrida. Con la computación perimetral, una institución o empresa puede utilizar y distribuir una agrupación común de recursos en una gran variedad de entornos o ubicaciones.

### **¿Por qué es importante el edge computing?**

Podemos comprobar un ejemplo real de Computación perimetral en la aplicación de la realidad virtual y aumentada. Comúnmente, la realidad virtual y aumentada disminuye su capacidad debido al ancho de banda insuficiente y una alta latencia. Quienes trabajan con dichas tecnologías padecen los problemas frecuentes con la baja potencia en latencia y baja velocidad, de tal modo que no permite la capacidad de alcanzar una experiencia envolvente. Una solución sería, que la computación perimetral permita que se suban

a la nube aquellas partes del proceso que gasten o consuman muchos recursos informáticos. Si analizamos otro ejemplo más como en los Autos inteligentes es que la aplicación y ejecución de esos mismos procesos en el Internet de las cosas (IoT). Imaginemos si hubiese latencia mientras el vehículo procesa la información sobre las velocidades, direcciones, estacionamiento, etc., podría suceder algún tipo de accidente. Al igual que los demás dispositivos u objetos del Internet de las cosas (IoT), el auto inteligente debe de captar, almacenar, procesar los datos y tomar decisiones lo más rápido posible en tiempo real.

Algunas otras utilidades de la Computación perimetral incluyen la capacidad para captar, almacenar, procesar y analizar datos en grandes cantidades en el mismo entorno, lo que nos permite tomar decisiones en tiempo real. La computación perimetral disminuye aún más el riesgo de perder o exponer nuestros datos e información confidencial, debido a que mantiene toda la información en un entorno contiguo. De tal forma que las instituciones o empresas puedan monitorizar mejor la distribución de sus datos e información, como por ejemplo que las entidades cumplan con las normativas, proteger los secretos comerciales del área, o que cumplan con las políticas estandarizadas siguiendo el Reglamento General de Protección de Datos (GDPR).

Por último, gracias a la computación perimetral, los usuarios y clientes empresariales acceden y disfrutan de la disminución de costos y flexibilidad relacionados con la Computación perimetral. Debido a la alta potencia y velocidad de la gestión de la información cerca de su entorno, los entornos adyacentes pueden seguir trabajando independientemente del entorno central, incluso si el entorno central deja de funcionar. Además, la computación perimetral permite disminuir considerablemente los costos que se deberían de pagar por el ancho de banda para trasladar la información entre los entornos adyacentes y centrales.

### **¿Cuáles son las ventajas del edge computing?**

La computación perimetral se ha enfocado en centralizar los servicios en un grupo de entornos que gestionan grandes cantidades de datos. Este enfoque permite que se pueda compartir y ampliar los recursos que conforman la computación perimetral alcanzando una mayor eficiencia, sin desatender en la gestión de los dispositivos el control y la seguridad de las empresas.

La computación perimetral tiene como objetivo solucionar aquellos casos prácticos que la centralización de la computación en las nubes no puede solucionar de forma eficiente, mayormente debido a los estándares y restricciones de la red. La computación perimetral se enfoca en varios pequeños entornos informáticos que padecen las restricciones del ancho de banda, que disminuyen los costos de la implementación de los elementos de la red, disminuyen las tardanzas en la transmisión de datos, limitan los errores, fallas del servicio y gestiona mejor la transmisión de los datos confidenciales. Los tiempos de subida de los datos se reducen en milisegundos, y los servicios en línea aplicados en los objetos y dispositivos más cercanamente de los usuarios y clientes mejoran las capacidades de procesamiento y almacenamiento en memoria caché estática y dinámica.

Para los proveedores de servicios y las empresas estas ventajas se traducen en aplicaciones de alta disponibilidad y baja latencia con monitoreo en tiempo real. Y para los usuarios y clientes finales, esto traduce en una experiencia del cliente más eficiente.

### **¿Cuáles son los desafíos del edge computing?**

El principal problema de computación perimetral no solamente es la distribución sino también es la escalabilidad:

La distribución horizontal enfocados a varios entornos pequeños puede ser más difícil que adicionar la capacidad equivalente a un entorno central de datos. Los clientes y las empresas pequeñas pueden alcanzar dificultades en el momento de administrar el aumento de los costos principales de las ubicaciones físicas de los entornos.

Mayormente, los entornos de la computación perimetral se ubican en lugares alejados y no cuentan con el alcance técnico suficiente. Si se sucede alguna falla en el sitio, se debería de contar con una infraestructura al alcance de tal forma que los empleados en el sitio que no tienen experiencia técnica puedan reparar fácilmente y que un grupo pequeño de especialistas que se encuentran en otro lugar puedan administrar y gestionar de forma remota y centralizada.

Se debe enfocar en todos los lugares de la computación perimetral que las operaciones de administración y control se puedan aplicar en todos los demás entornos para simplificar el control y la gestión, Agilizar la resolución de incidentes y problemas para evitar que el desempeño de la aplicación presente pequeñas diferencias entre los diferentes entornos, lo que se denomina comúnmente “Des configuración”

Aunque la computación perimetral ofrece mejores y mayor control de gestión sobre los flujos de datos e información al limitar los datos según su geografía, la seguridad física del entorno suele ser mucho menor. Lo cual puede provocar un mayor riesgo de situaciones accidentales o maliciosas, como por ejemplo la ruptura o desconexión de algún cable por error. (Hat, 2019)

Para Zaroni (2019) el origen del término Computación en la nube no está claramente definido, pero fue en los '90s en el que las instituciones o empresas de telecomunicaciones, que anteriormente ofrecían conexiones de datos punto a punto, luego comenzaron a ofrecer nuevas prestaciones de conexión como la VPN red privada virtual a un mínimo costo. Y así surgió la nube. En el mes de agosto del año 2006 Amazon Web Services (AWS) introdujo su Elastic Compute Cloud siendo este el primer servicio de Computación en la nube disponible para el público. Microsoft Azure se dio a conocer como “Azure” en el mes de octubre del año 2008 y fue lanzado el 1 de febrero del 2010 como Windows Azure, antes de ser rebautizado como Microsoft Azure el 25 de marzo de 2014.

Con una fuerte incursión hace poco tiempo de los dispositivos conectados (Internet de las cosas), debido a las interfaces de conexión y sensores, cantidades mayores de datos digitales comenzaron a aglomerarse en los servidores de las empresas que ofrecían estos servicios (iniciando Samsung a Amazon, siguiendo por LG, Telefónica, LG, Wearables, SmartThings, Fitbit, Nokia, etc.). Los comercios aumentan sin pausa: Existirá según la empresa de telecomunicaciones Cisco unos 50 000 millones de dispositivos IoT en el año 2020. El enfoque está basado en los datos e información que dichos sensores producen a cada segundo. Cada vez más datos para captar, controlar, gestionar, almacenar, predecir, analizar, etc.

Estas cantidades de datos e información resultan pasivos si carecen de interconexión. Podemos crear un electrodoméstico muy inteligente, como, por ejemplo, una refrigeradora, pero si esta congeladora no se conecta con los demás electrodomésticos de cualquier otra marca o fabricante, todo su potencial de inteligencia quedara disminuida. Y peor aun cuando se trata de grandes entornos empresariales e industriales como aeropuertos, fábricas, oficinas o edificios, como, por ejemplo, los autos autónomos.

¿Pero qué sucede si toda esta data en lugar de transferirse hasta la nube, antes se analizan y procesan? ¿Es decir, en algún punto antes de llegar al servidor central, esta data se analiza y procesa directamente en el propio dispositivo? De tal forma que, cada dispositivo o elemento conectado se podría hacer el papel de un nodo de conectividad, procesamiento y almacenamiento. A esto se le llama La computación perimetral.

Entonces, la computación perimetral permite que los datos e información producidos por los dispositivos u objetos que conforman el IoT se procesen más cercanamente en donde se crearon en lugar de transmitirlos a través de largos recorridos para que puedan llegar a la nube.

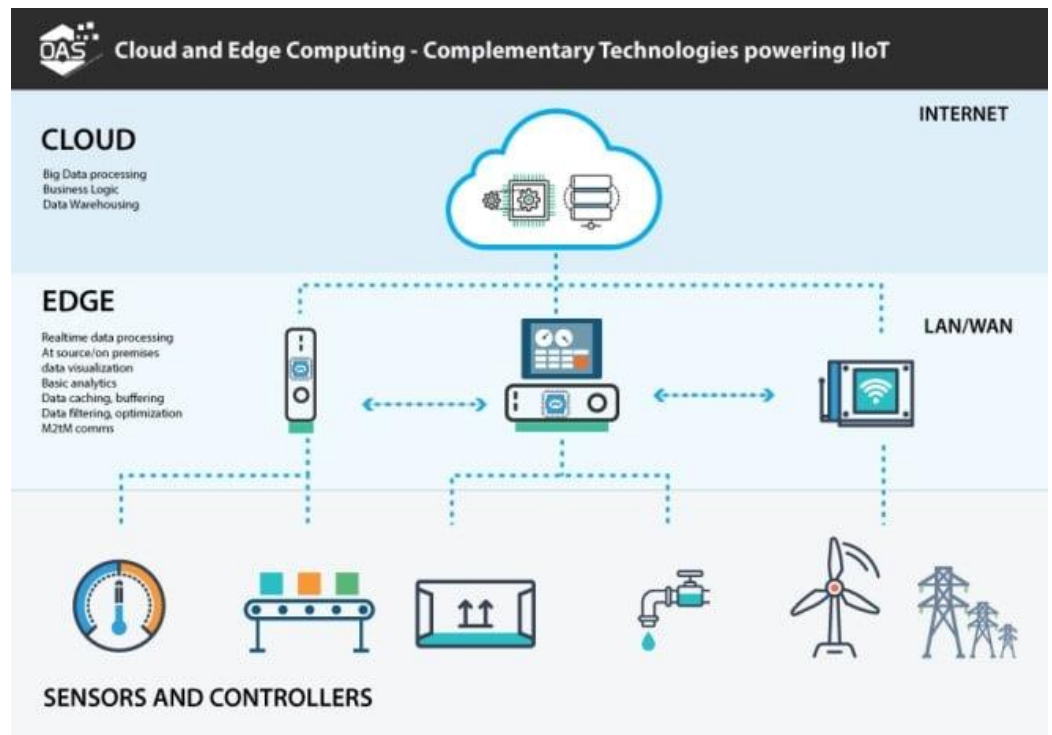


Figura 3: Capas de funcionamiento del Cloud y Edge Computing

### 2.2.3. Raspberry Pi 3 B+:

Es una minicomputadora provista de una placa electrónica en la que se puede cargar un sistema operativo por medio de una memoria Micro Sd (Raspbian, Windows 10, Risc OS Pi, etc.).

Para realizar todo tipo de proyectos de Robótica, Electrónica, Mecatrónica, etc.

Se puede realizar también como un servidor, Centro multimedia, Centro de información, fines recreativos, etc.

Procesador: CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.

RAM: 1GB LPDDR2 SDRAM

Conexión Wifi: 2.4GHz y 5GHz IEEE 802.11. b/g/n/ac, Bluetooth 4.2, BLE.

Puerto Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)

Conector GPIO de 40 pines.

Puerto HDMI, Puerto CSI para cámara, Puerto DSI para pantalla táctil.



Incorpora 4 puertos Usb 2.0

Salida de audio estéreo y vídeo compuesto (Jack 3.5mm)

Ranura para Micro SD

Power over ethernet (PoE) 5V/ 2.5A

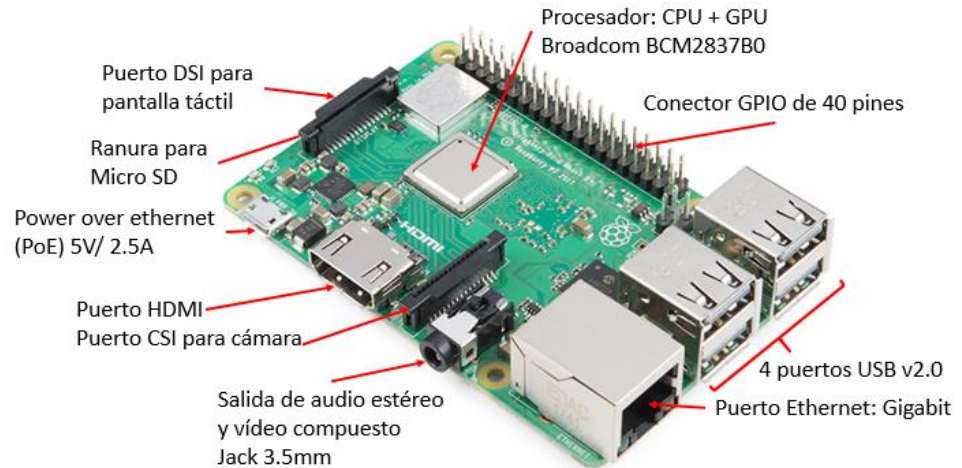


Figura 4: Raspberry Pi 3B+ (Fuente: Elaboración propia)

### **Formatear la memoria MicroSD y descargar los programas:**

Formatear la memoria MicroSD

Entrar a la dirección web: <https://www.raspberrypi.org/>.

Seleccionar “Downloads”, Noobs,

Extraer la carpeta Noobs y copiar el contenido de la carpeta Noobs en la memoria MicroSD.

### **Instalando Raspbian:**

Raspbian es el sistema operativo oficial de la Raspberry, es un Linux y está basado en la distribución Debian; Es libre, es decir, todo el código fuente está abierto (Se puede acceder, añadir mejoras, corregir errores, etc.).

Conectar la memoria MicroSD en el Raspberry.

En el Raspberry, conectar Ratón, teclado y Hdmi y la Energía.

En la pantalla de la computadora y en el programa Noobs, seleccionar “Raspbian”.

Paralelamente, aparecieron las denominadas HATS (sombreros) que expandían la potencia de la Raspberry a niveles más amplios que las famosas shields propias del mundo Arduino.

### **2.3. Glosario de términos:**

#### **Big data:**

Es la agrupación de datos o combinaciones de agrupaciones de datos cuyas características en complejidad (variabilidad), tamaño (volumen), y transmisión de datos (velocidad) complican su recepción, administración, control, procesamiento, almacenamiento y análisis mediante herramientas tecnológicas estándares, tales como la estadística convencional y bases de datos relacionales, en un intervalo de tiempo necesario para que sean provechosos.

#### **Biosensor:**

Un biosensor es aquel dispositivo que está conformado por dos elementos principales: Un receptor biológico como por ejemplo el ADN, las proteínas, las células, etc., el cual está capacitado para detectar una sustancia específica aprovechando las variaciones biomoleculares y un Transductor o sensor, que está capacitado para interpretar la reacción de detectar la biología que elabora el receptor y "traducirla" en una señal que se puede contabilizar.

#### **Inmótica:**

La inmótica es el conjunto de conocimientos tecnológicos aplicados a la gestión, automatización y control de edificios inteligentes no orientados a la vivienda, como centros comerciales, hoteles, escuelas, hospitales, universidades, etc., posibilitando una administración y control eficiente del empleo de la energía, contribuir con la seguridad, comodidad, y comunicación entre el sistema y el usuario.

#### **M2M:**

Se entiende como "machine to machine" al intercambio de información, es decir, es la comunicación en forma de datos entre dos máquinas que se encuentran en distintos lugares, pero conectadas remotamente.

**NFC:**

Se entiende como “Near Field Communication” es cual es una tecnología inalámbrica de reducido alcance que tiene la facultad de conectar dos dispositivos o elementos al enviar una señal, y que al mismo tiempo puede también captar una señal. Aceptando, por lo tanto, una actualización escritura-lectura en ambos sentidos.

**NLP:**

Se entiende como Procesamiento de lenguajes naturales al sentido de comunicación, psicoterapia y desarrollo personal, inventado por Richard Bandler y John Grinder en los Estados Unidos (California), al inicio de la década de los 70. Sus inventores sustentan que se da una conexión entre el lenguaje (lingüística) y los procesos neurológicos (neuro); En el lenguaje lingüística, y los patrones de comportamiento aprendidos durante la experiencia (programación), aceptando que estos rasgos se pueden cambiar para mejorar, lograr y alcanzar objetivos específicos en la vida.

**ZIGBEE:**

Está conformado por un grupo de protocolos de alto nivel de comunicación. Los cuales son utilizados para los servicios de emisión de señales digitales de datos enfocado a ahorrar lo mayormente posible en energía. Es una tecnología basada en el estándar de la IEEE, el IEEE 802.15.4.

## CAPÍTULO III

### METODOLOGÍA

#### 3.1. Metodología de desarrollo:

La metodología empleada para el diseño y la propuesta del sistema de control está basada en la metodología Top-Down que consiste primero en analizar los requerimientos precisos para que en base a estos requerimientos, desarrollar la topología y el protocolo de red a emplear, luego seleccionar las tecnologías y dispositivos para dar inicio a las fases de documentación e implementación de la propuesta para llegar a la aplicación, monitoreo y optimización de la red propuesta en un ciclo que no tiene final (Saavedra, 2017).

A continuación, se muestran las fases de la metodología:

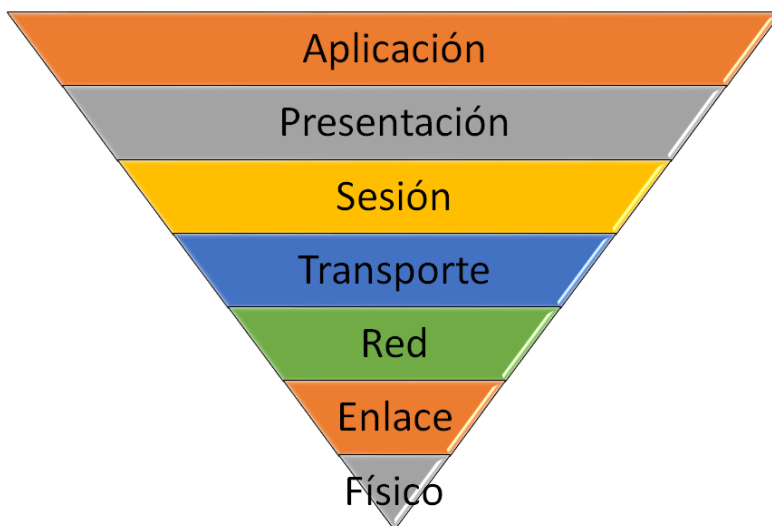


Figura 5: Modelo OSI de Redes (Fuente: Elaboración propia)

A continuación, se muestran las fases de la metodología:

##### 3.1.1. Fase 1: Analizar Requerimientos

- \* Analizar metas del negocio
- \* Analizar metas técnicas

##### 3.1.2. Fase 2: Desarrollar Diseño Lógico

- \* Diseñar topología de red

- \* Desarrollar estrategias de administración de red

### **3.1.3. Fase 3: Desarrollar Diseño Físico**

- \* Seleccionar tecnologías y dispositivos para la red

### **3.1.4. Fase 4: Probar, optimizar y documentar diseño**

- \* Probar el diseño de red
- \* Optimizar el diseño de red
- \* Documentar el diseño

Fase 5: Implementar y probar la red

- \* Realizar cronograma de implementación
- \* Implementación del diseño de red (final)
- \* Realizar pila de pruebas.

Fase 6: Monitorear y Optimizar la Red

- \* Operación de la red en producción
- \* Monitoreo de la red
- \* Optimización de la red

De las cuales se emplearán las cuatro primeras fases de la metodología ya que el estudio solo se enfoca en el diseño y propuesta del sistema.

## **3.2. Herramientas de desarrollo:**

HARDWARE	Bombilla Led inteligente marca Xiaomi Esta Bombilla led inteligente se conectara a la red WIFI para ser controlada por la centralita bajo el servidor OpenHab
	Lámpara Donde se conectara la bombilla, esta debe estar conectada a la red eléctrica.
	Computador/ Servidor (RaspBerry Pi) Computadora que actuara de centralita para controlar el sistema de iluminación.
	Router Switch Access Point Dispositivo de enlace para la conexión de la bombilla con la centralita.
	Teléfono móvil Dispositivo que permitirá controlar desde una interfaz dotada por la centralidad en red el control del sistema de iluminación.

Tabla 1: Herramientas de desarrollo Hardware

SOFTWARE	Sistema Operativo Ubuntu Server Distribución Ubuntu Server 14 LTS, donde se instalara la solución y la configuración de la solución
	OpenHab Solution En este paquete se hará la programación y se programara mediante scripts el binding o unión de los dispositivos inteligentes (bombilla) con la centralidad conectada a la red.
	Aplicación Cliente para el teléfono Móvil La aplicación por defecto será el navegador con la dirección IP de la centralita.

Tabla 2: Herramientas de desarrollo Software

## CAPÍTULO IV

### DESARROLLO E IMPLEMENTACIÓN

#### 4.1. Desarrollo e implementación:

##### 4.1.1. Fase 1: Análisis de requerimientos:

Requerimientos físicos:

- \* Cable de red RJ45
- \* Switch fastEthernet
- \* Bombilla Inteligente
- \* Servidor con OpenHab
- \* Conector RJ45
- \* Crimper.

Requerimientos funcionales:

CÓDIGO	DESCRIPCIÓN	TIPO
REQ01	La red debe tener topología estrella y al menos emplear la tecnología Fast ethernet (1000Mbps)	Hardware
REQ02	En la red debe existir un switch al menos para poder conectar el servidor con las bombillas inteligentes.	Hardware
REQ03	El sistema OpenHab debe instalarse en un sistema operativo libre como Linux, en una distribución como Ubuntu server en la versión 14.0 como mínimo.	Software
REQ04	Las bombillas inteligentes deben tener el protocolo ZigBee y/o la tecnología wifi.	Software

*Tabla 3: Requerimientos funcionales*

##### 4.1.2. Fase 2: Desarrollar el diseño lógico:

A continuación, se muestra el diseño lógico y detallado de la red:

#### RED LAN:

##### Insertando los dispositivos:

- \* **Switch0:** Network devices > Switches > 2950-24
- \* **Switch1:** Network devices > Switches > 2950-24
- \* **Pc0:** End devices > Pc
- \* **Pc1:** End devices > Pc

\* **Pc2:** End devices > Pc

\* **Pc3:** End devices > Pc

\* **Pc4:** End devices > Pc

\* **Pc5:** End devices > Pc

### **Conectando los dispositivos:**

\* Conexión de dispositivos iguales: Cable indirecto (Copper cross)

\* Conexión de dispositivos diferentes: Cable directo (Copper straight)

\* **Coopper straight:** Connections > Coopper straight-Through

\* Connections > Directo: Switch0 (FastEthernet0/1) – Pc0  
(FastEthernet0)

\* Connections > Directo: Switch0 (FastEthernet0/2) – Pc1  
(FastEthernet0)

\* Connections > Directo: Switch0 (FastEthernet0/3) – Pc2  
(FastEthernet0)

\* Connections > Directo: Switch1 (FastEthernet0/1) – Pc3  
(FastEthernet0)

\* Connections > Directo: Switch1 (FastEthernet0/2) – Pc4  
(FastEthernet0)

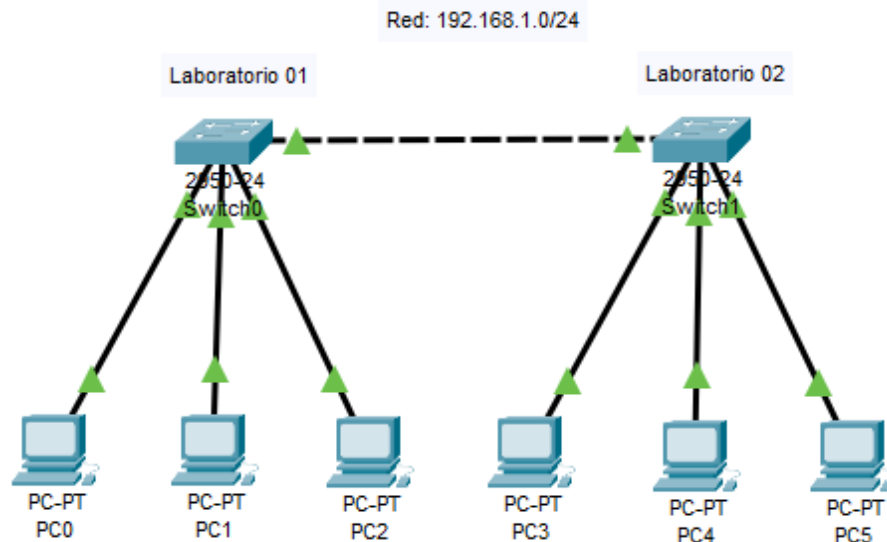
\* Connections > Directo: Switch1 (FastEthernet0/3) – Pc5  
(FastEthernet0)

\* Connections > Cruzado: Switch0 (FastEthernet0/4) – Switch1  
(FastEthernet0/4)

### **Insertando Notas:**

\* Seleccionar “Place note”





\*

Figura 6: Red Lan (Fuente: Elaboración propia)

### Configurando las Pc:

- \* Seleccionar Pc0 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.2, Mask: 255.255.255.0
- \* Seleccionar Pc1 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.3 Mask: 255.255.255.0
- \* Seleccionar Pc2 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.4 Mask: 255.255.255.0
- \* Seleccionar Pc3 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.5 Mask: 255.255.255.0
- \* Seleccionar Pc4 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.6 Mask: 255.255.255.0
- \* Seleccionar Pc5 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.7 Mask: 255.255.255.0

### Comprobando la conexión mediante el comando ping:

- \* Seleccionar la Pc0,
- \* Pestaña Desktop > Command prompt
- \* Digitar ping 192.168.1.7 [Enter]

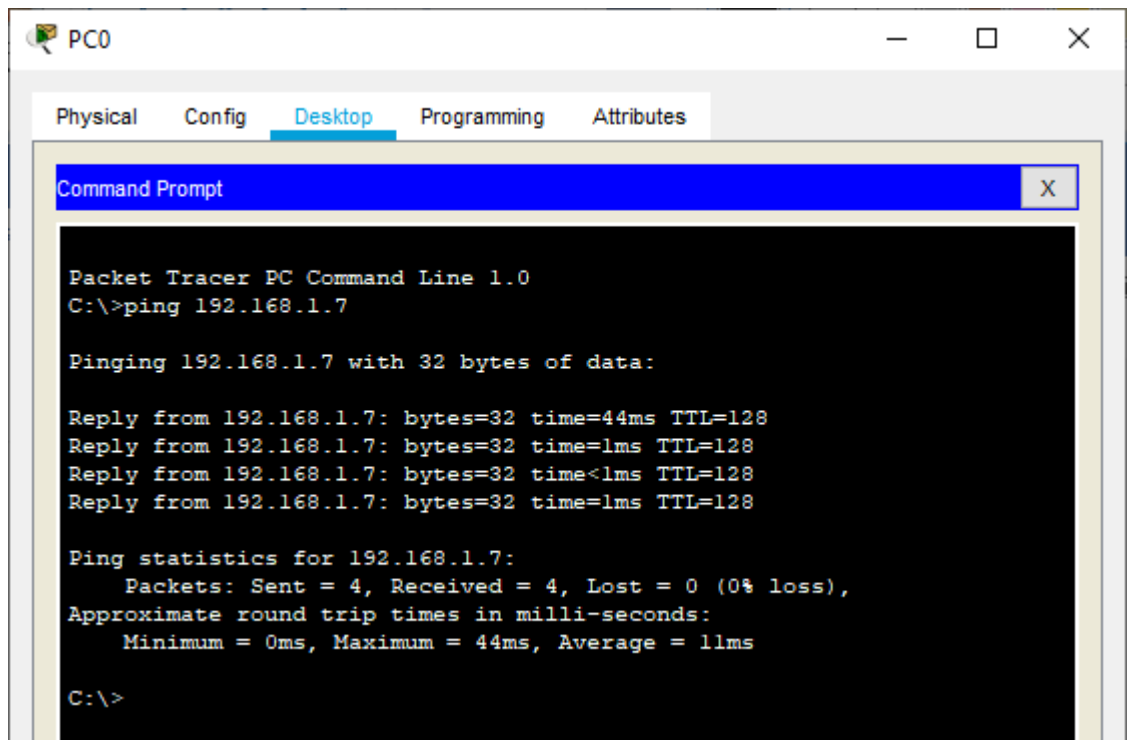


Figura 7: Consola para probar conectividad (Fuente: Elaboración propia)

## RED WAN:

### Insertando los dispositivos:

- \* **Switch0:** Network devices > Switches > 2950-24
- \* **Switch1:** Network devices > Switches > 2950-24
- \* **Pc0:** End devices > Pc
- \* **Pc1:** End devices > Pc
- \* **Pc2:** End devices > Pc
- \* **Pc3:** End devices > Pc
- \* **Pc4:** End devices > Pc
- \* **Pc5:** End devices > Pc
- \* **Router0:** Network devices > Routers > 2911
- \* **Router1:** Network devices > Routers > 2911

### Conectando los dispositivos:

- \* **Conexión de dispositivos iguales:** Cable indirecto (Copper cross)
- \* **Conexión de dispositivos diferentes:** Cable directo (Copper straight)
- \* Connections > Directo: Switch0 (FastEthernet0/1) – Pc0 (FastEthernet0)

- \* Connections > Directo: Switch0 (FastEthernet0/2) – Pc1 (FastEthernet0)
- \* Connections > Directo: Switch0 (FastEthernet0/3) – Pc2 (FastEthernet0)
- \* Connections > Directo: Switch1 (FastEthernet0/1) – Pc3 (FastEthernet0)
- \* Connections > Directo: Switch1 (FastEthernet0/2) – Pc4 (FastEthernet0)
- \* Connections > Directo: Switch1 (FastEthernet0/3) – Pc5 (FastEthernet0)
- \* Connections > Directo: Router0 (FastEthernet0/0) – Switch0 (FastEthernet0/4)
- \* Connections > Directo: Router1 (FastEthernet0/0) – Switch1 (FastEthernet0/4)

### **Configurando los dispositivos:**

#### **\* Para la red de Huánuco:**

- \* Seleccionar Pc0 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.2, Mask: 255.255.255.0
- \* Seleccionar Pc1 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.3 Mask: 255.255.255.0
- \* Seleccionar Pc2 > Pestaña Desktop > Ip configuration > Ip address: 192.168.1.4 Mask: 255.255.255.0

#### **\* Para la red de Lima:**

- \* Seleccionar Pc3 > Pestaña Desktop > Ip configuration > Ip address: 192.168.2.5 Mask: 255.255.255.0
- \* Seleccionar Pc4 > Pestaña Desktop > Ip configuration > Ip address: 192.168.2.6 Mask: 255.255.255.0
- \* Seleccionar Pc5 > Pestaña Desktop > Ip configuration > Ip address: 192.168.2.7 Mask: 255.255.255.0

### **Conectando los routers:**

- \* Agregando módulos al slot 0:
- \* Seleccionar Router0 > Pestaña Physical > Modules: WIC -1T
- \* Seleccionar el Router1 > Pestaña Physical > Modules: WIC -1T

## Conectando los Routers:

- \* Connections > Serial DCE: Router0 (Serial0/0/0) – Router1(Serial0/0/0)

## ¿Quién tiene el reloj? – Visualizando las etiquetas de los puertos:

- \* Menú Options > Preferences > Activar: Always show port labels in logical WorkSpace.
- \* El router que contiene el reloj se le llama Master.

## Configurando mediante programación el Router0:

- \* Seleccionar el Router0.
- \* En la Pestaña Cli, presionar [Enter]

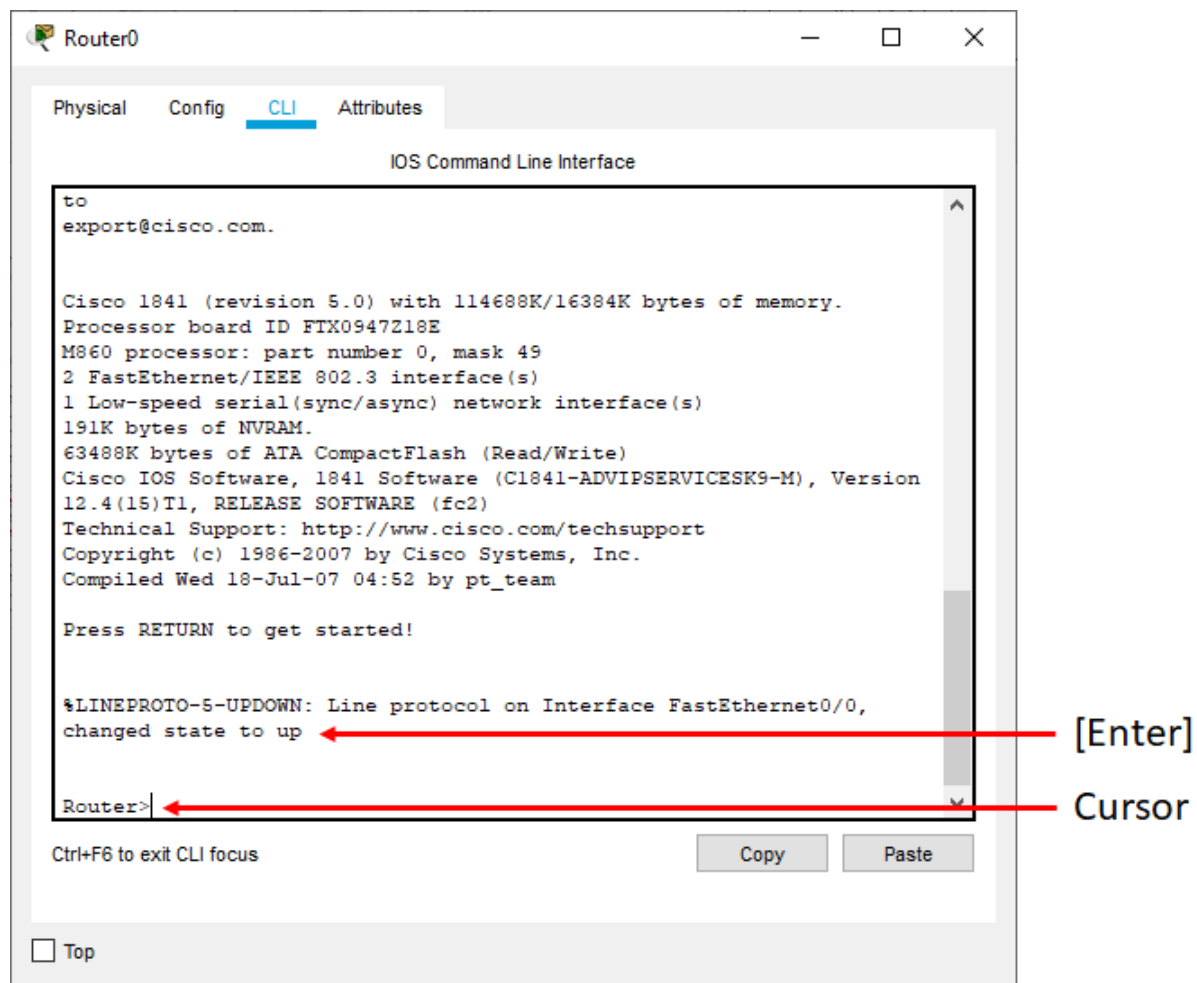


Figura 8: Consola Clic Router (Fuente: Elaboración propia)

Digitar:

```

--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]: no
-- Digitamos No para que no reconozca el sistema operativo.
Router> enable
-- Modo administrativo:
Router# configure terminal
-- Modo configuración:
Router (Config)# interface gigabitEthernet 0/0
Router (Config-if)# ip address 192.168.1.1 255.255.255.0
Router (Config-if)# no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)# exit
Router(config)# interface serial 0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.252
Router(config-if)# clock rate 128000
Router(config-if)# no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
Router(config-if)# [Ctrl]+[Z]
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router# write
Building configuration...
[OK]
Router#

```

\*

Tabla 4: Configuración Router0 (Fuente: Elaboración propia)

### Configurando mediante el asistente el Router1:

- \* Seleccionar el Router1
- \* Pestaña Config > FastEthernet0/0 > Ip address: 192.168.2.1 255.255.255.0, Port status: activar ON
- \* Pestaña Config > Serial0/0/0> Ip address: 10.0.0.2 255.255.255.252, Port status: activar ON
- \* Se puede enviar paquetes entre Routers pero no entre Pcs.

### Asignando protocolo estático:

- \* Seleccionar el Router0
- \* Pestaña CLI

```

Router> enable
Router# configure terminal
Router(config)# ip route 192.168.2.0 255.255.255.0 10.0.0.0
Router(config)# [Ctrl]+[Z]
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router# write
Building configuration...
[OK]
Router#

```

*Tabla 5: Asignando Protocolo Estático al Router0*

- \* Seleccionar el Router1
- \* Pestaña CLI

```

Router(config-if)# exit
Router(config)# ip route 192.168.1.0 255.255.255.0 10.0.0.0
Router(config)# [Ctrl]+[Z]
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router# write
Building configuration...
[OK]
Router#

```

*Tabla 6: Asignando Protocolo Estático al Router1*

### **Configurando las Puertas de entradas de las Pcs:**

- \* Pc0, Pc1, Pc2 > Pestaña Desktop > Ip configuration > Default gateway:  
192.168.1.1
- \* Pc3, Pc4, Pc5 > Pestaña Desktop > Ip configuration > Default gateway:  
192.168.2.1
- \* Se puede enviar paquetes entre routers y pcs.

### **Probando:**

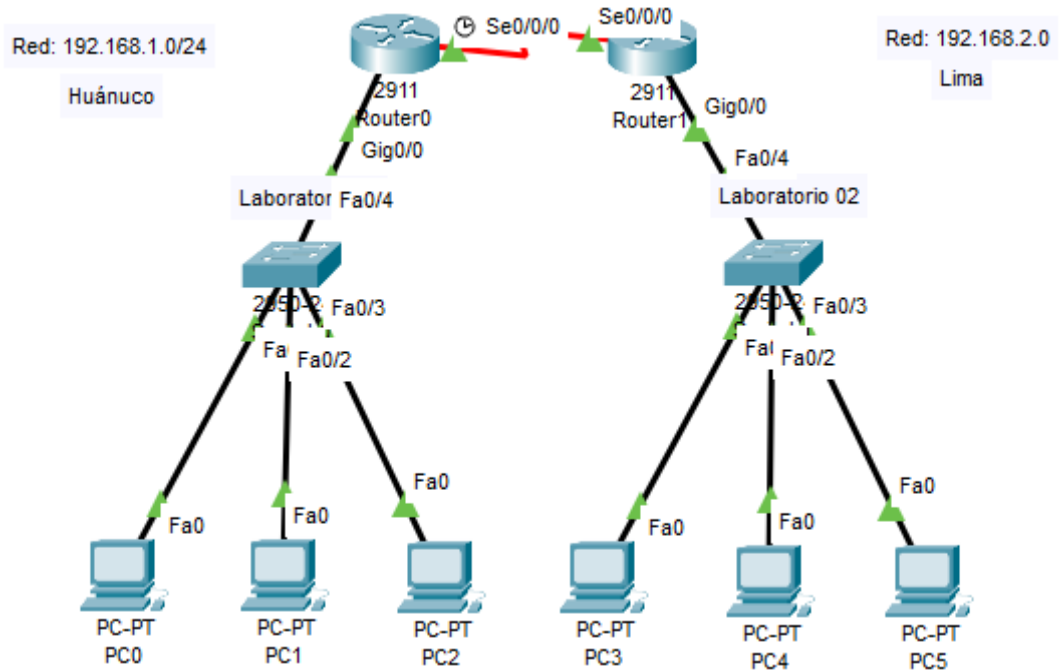


Figura 9: Red WAN Huánuco - Lima (Fuente: Elaboración propia)

#### 4.1.3. Fase 3: Desarrollar el diseño físico:

Para el desarrollo físico, dada las circunstancias de la pandemia, se optó por emular el proceso físico, recurriendo al software Packet tracer y empleando la tecnología IoT.

##### INTERNET DE LAS COSAS (IOT):

\* **Red alámbrica básica:**

\* **Insertando los dispositivos:**

\* **Insertando el Foco inteligente:**

\* End devices > Home > Light

\* **Insertando el Switch:**

\* Network devices > Switches > 2960

\* **Insertando el dispositivo enlazador:**

\* Network devices > Wireless devices > Home gateway

\* **Insertando la Pc:**

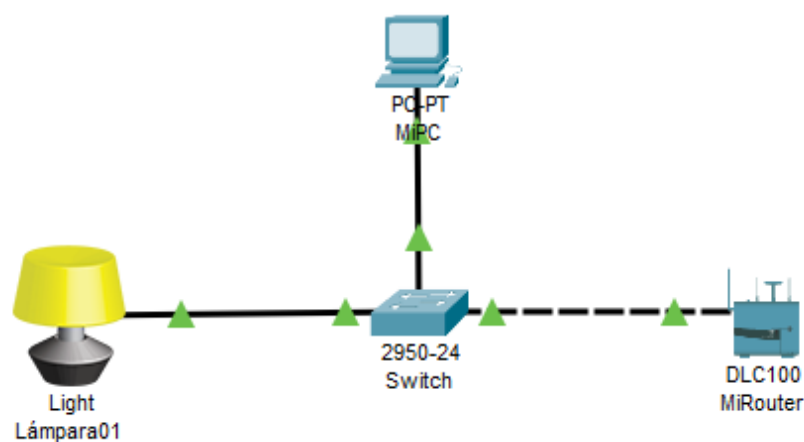
\* End devices > PC

\* **Configurando los nombres:**

\* IoT0: Pestaña Config > Settings > Display name: Lámpara01

\* Switch0: Pestaña Config > Settings > Display name: Switch

- \* Home gateway0: Pestaña Config > Settings > Display name: MiRouter
- \* Pc0: Pestaña Config > Settings > Display name: MiPC
- \* **Conectando los dispositivos:**
- \* Connections > Cruzado: Switch (Fast ethernet0/1) – MiRouter (Ethernet 1)
- \* Conecctions > Directo: Switch (FastEtherne0/2) – MiPC (FastEthernet0)
- \* Conecctions > Directo: Switch (FastEtherne0/3) – Lámpara01 (FastEthernet0)



*Figura 10: Red Alámbrica Básica.*

### **Configurando los dispositivos:**

- \* **Configuración por defecto de MiRouter:**
- \* Pestaña Config > Internet > Activado: DHCP
- \* Pestaña Config > LAN > Ip address: 192.168.25.1
- \* Pestaña Config > LAN > Subnet mask: 255.255.255.0
- \* **Configurando MiPC:**
- \* Pestaña Desktop > Web browser > 192.168.25.1: "Request timeout"
- \* Pestaña Config > FastEthernet0 > Ip configuration > Activar: DHCP
- \* Pestaña Desktop > Web browser > 192.168.25.1: (Home gateway login)
- \* Otra forma:
- \* Pestaña Desktop > IoT monitor > Login > "Request timeout"
- \* Pestaña Config > FastEthernet0 > Ip configuration > Activar: DHCP.
- \* Pestaña Desktop > IoT monitor > Login > (IoT monitor)
- \* **Configurando Lámpara01:**



- \* Pestaña Config > Settings > IoT server > Activar: Home gateway.
- \* Pestaña Config > FastEthernet0 > Ip configuration > Activar: DHCP.

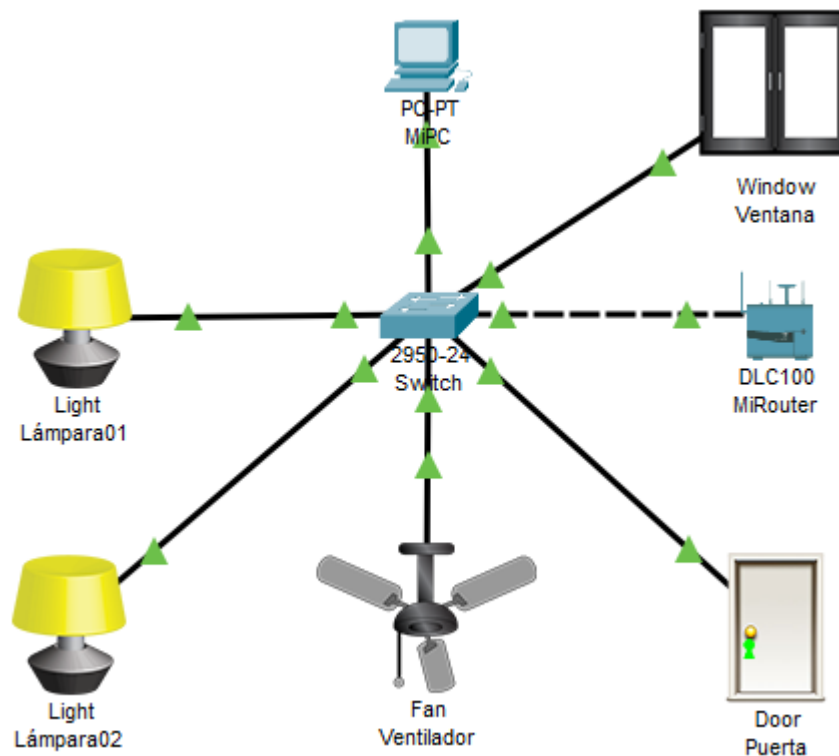
**Controlando el dispositivo desde MiPC con el navegador:**

- \* Seleccionar MiPC
- \* Pestaña Desktop > Web browser > Url: 192.168.25.1; Username: admin; Password: admin

**Controlando el dispositivo desde MiPC con la aplicación “IoT monitor”:**

- \* Pestaña Desktop > IoT Monitor > Login

**Insertando otros dispositivos:**



*Figura 11: Dispositivos IoT*

**Red inalámbrica básica:**

- \* Del ejercicio anterior, eliminar el cableado de los dispositivos inteligentes.

**Configuración por defecto de MiRouter:**

- \* Pestaña Config > Wireless > SSID: HomeGateway

**Cambiando por el adaptador inalámbrico a los dispositivos inteligentes:**

- \* Podemos cambiar el adaptador inalámbrico o agregar el adaptador inalámbrico a los dispositivos.
- \* Seleccionar la Lámpara01
- \* Pestaña Config > Botón: Advanced > Pestaña I/O Config > Network adapter: PT-IOT-NM-1CFE x PT-IOT-NM-1W
- \* Repetir para los demás dispositivos.

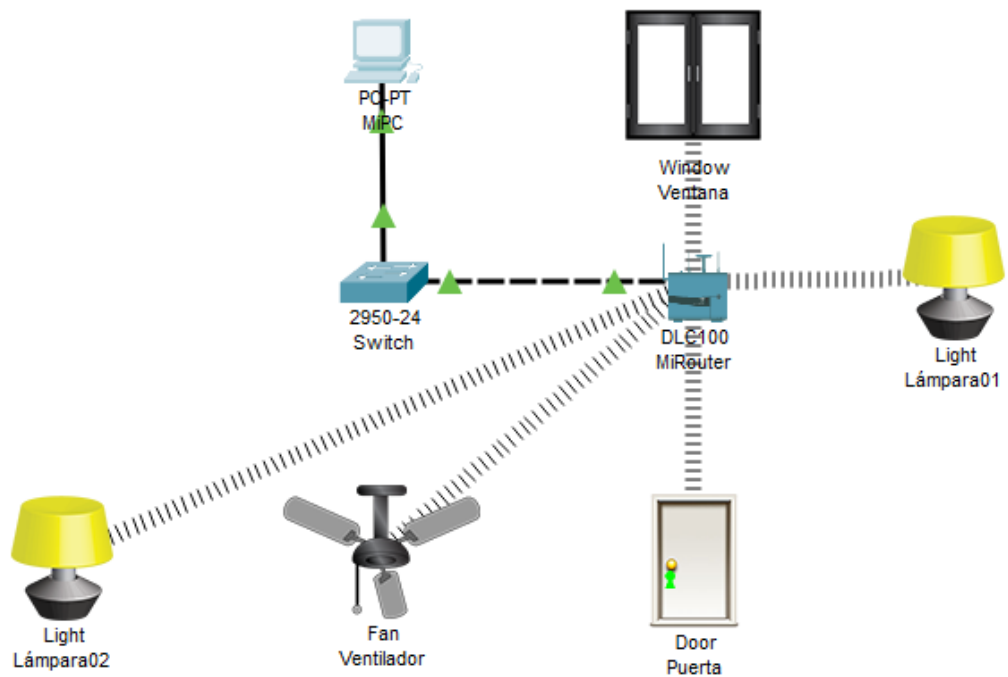


Figura 12: Red Inalámbrica Básica

- \* **Reconfigurando el dispositivo:**
- \* Seleccionar la Lámpara.
- \* Pestaña Config > Settings > IoT server > Activar: Home gateway.
- \* **Controlando los dispositivos desde MiPC con el navegador:**
- \* Seleccionar MiPC
- \* Pestaña Desktop > Web browser > Url: 192.168.25.1; Username: admin; Password: admin

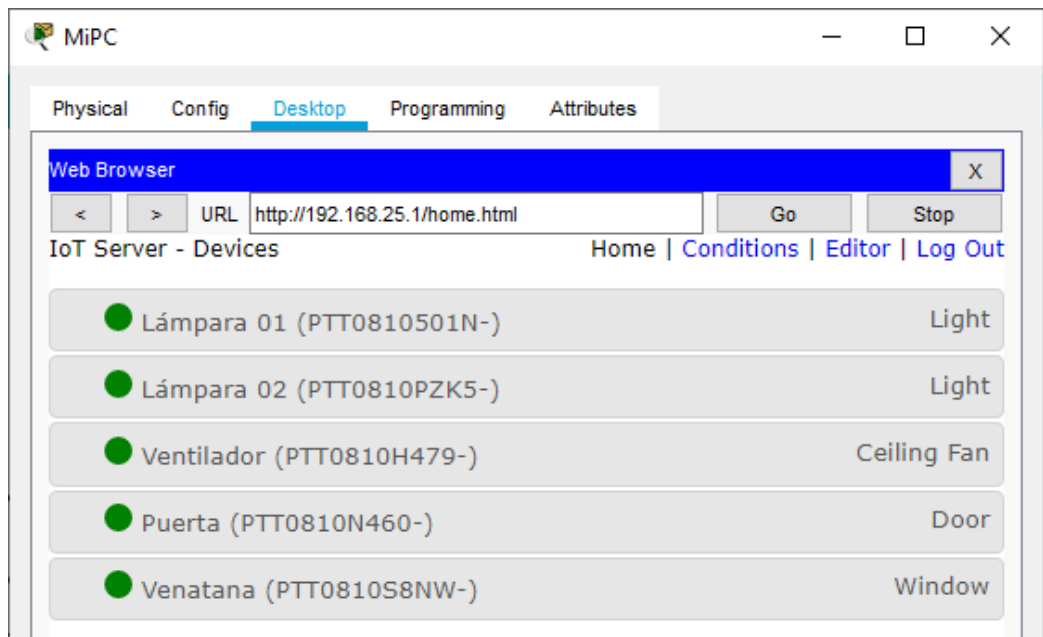


Figura 13: Escritorio de Pc - Navegador

## Controlando los dispositivos desde MiPC con la aplicación “IoT monitor”:

- \* Pestaña Desktop > IoT Monitor > Login

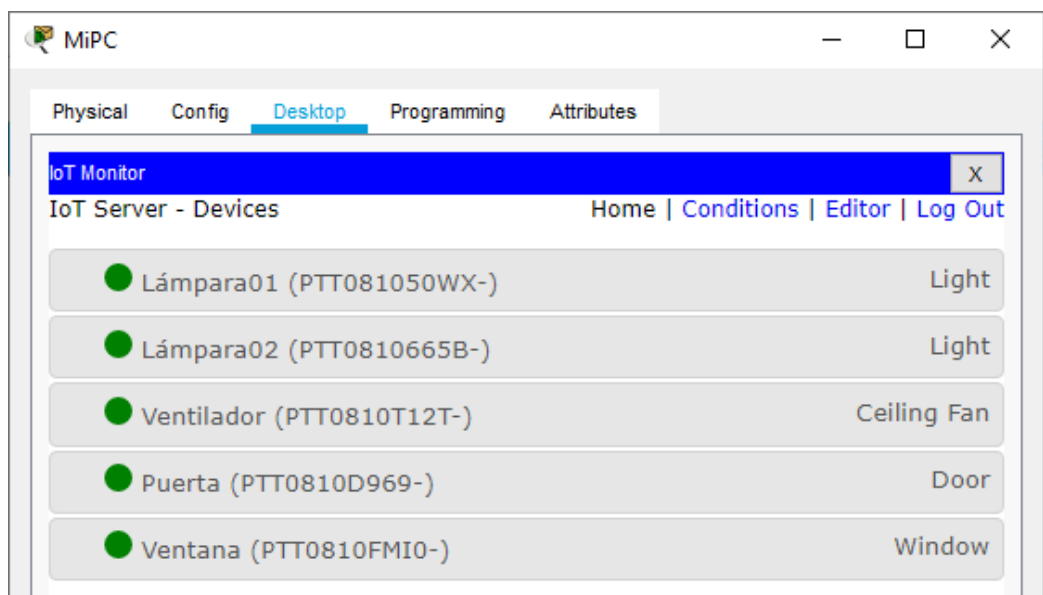


Figura 14: Escritorio Pc - Monitor IoT

- \* **Insertando el Móvil:**
- \* End devices > Smart device
- \* **Configurando el Móvil:**
- \* Seleccionar el Móvil
- \* Pestaña Config > Settings > Display name: MiMóvil

\* Pestaña Config > Wireless0 > SSID: HomeGateway

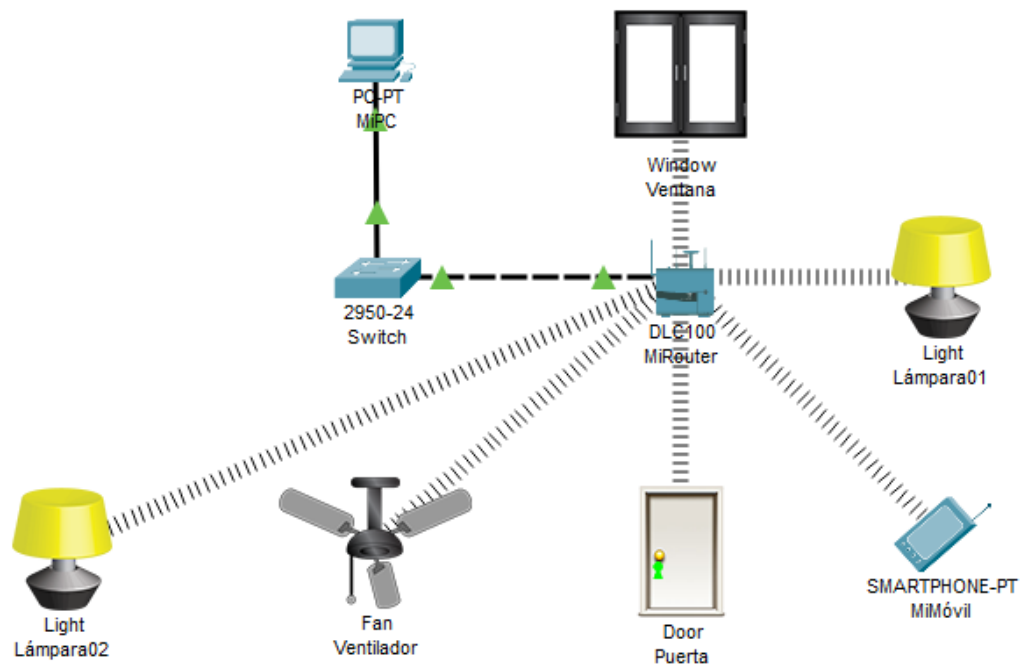


Figura 15: Red Inalámbrica Básica con Móvil

\* **Controlando los dispositivos desde el Móvil con el navegador:**

\* Seleccionar MiMóvil

\* Pestaña Desktop > Web browser > 192.168.25.1, admin, admin.

\* **Controlando los dispositivos desde el Móvil con la aplicación “IoT monitor”:**

\* Seleccionar MiMóvil

\* Pestaña Desktop > IoT Monitor > Login

\* **Red inalámbrica con autenticación:**

\* **Configurando MiRouter:**

\* Seleccionar MiRouter.

\* Pestaña Config > Wireless > SSID: MiCasa

\* Pestaña Config > Wireless > Autenticación > Activar: WPA2-PSK:  
123456789

\* Se verifica que desde MiPC se puede acceder al servidor, pero ya no a los dispositivos. Los dispositivos no pueden conectarse a una red con contraseña.

\* **Configurando el Móvil:**

\* Seleccionar el Móvil.

- \* Pestaña Config > Wireless0 > SSDI: MiCasa
- \* Pestaña Config > Wireless0 > Autenticación: WPA2-PSK: 123456789
- \* **Configurando los dispositivos IoT:**
- \* Seleccionar la Lámpara 01.
- \* Pestaña Config > Wireless0 > SSDI: MiCasa
- \* Pestaña Config > Wireless0 > Autenticación: WPA2-PSK: 123456789
- \* Pestaña Config > Settings > IoT server > Activar: Remote server
- \* Pestaña Config > Settings > IoT server > Server address: 192.168.25.1
- \* Pestaña Config > Settings > IoT server > User name: admin
- \* Pestaña Config > Settings > IoT server > Password: admin
- \* Pestaña Config > Settings > IoT server > Botón: Connect/
- \* Pestaña Config > Settings > Gateway/ DNS Ipv4 > Activar: DHCP.
- \* Pestaña Config > Settings > IoT server > Botón: Connect/ Refresh

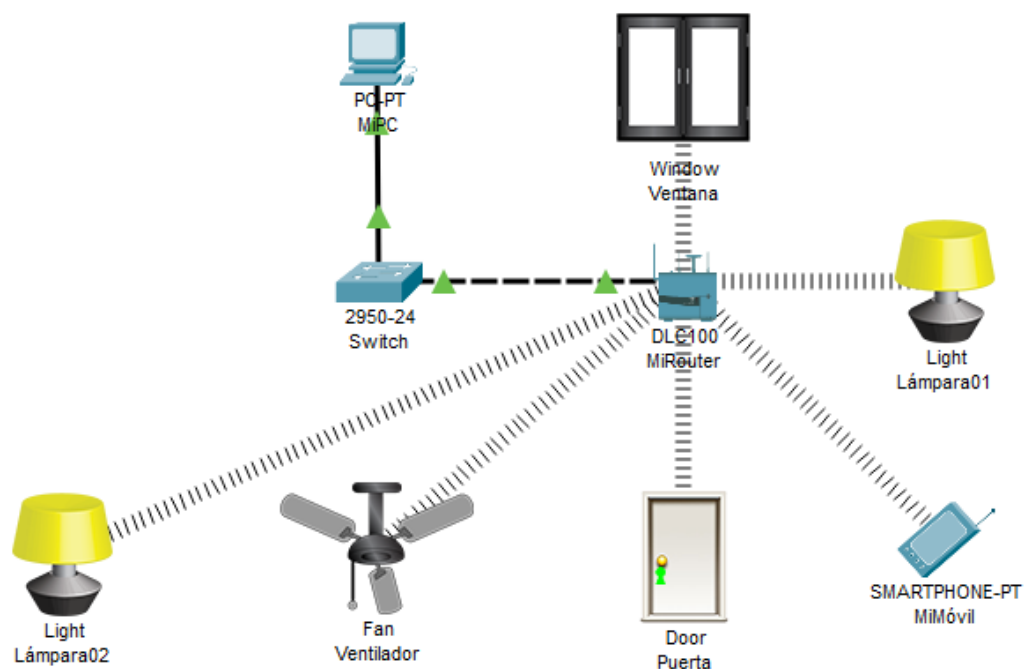


Figura 16: Red Inalámbrica con Autenticación.

- \* **Controlando los dispositivos desde MiPC con el navegador:**
- \* Seleccionar MiPC
- \* Pestaña Desktop > Web browser > Url: 192.168.25.1; User name: admin; Password: admin
- \* **Controlando los dispositivos desde MiMóvil con el navegador:**
- \* Seleccionar MiMóvil.

- \* Pestaña Desktop > Web browser > Url: 192.168.25.1; User name: admin; Password: admin.
- \* **Red con Servidor IoT:**
- \* **Insertando los dispositivos:**
- \* **Lámpara:** End devices > Home > Light
- \* **Switch:** Network devices > Switches > 2960
- \* **Pc:** End devices > Pc
- \* **Servidor:** End devices > Server PT
- \* **Conectando los dispositivos:**
- \* Connections > Directo: Server – Switch
- \* Connections > Directo: Switch - Pc
- \* Connections > Directo: Switch - Light
- \* **Configurando los dispositivos:**
- \* **Configurando el Servidor:**
- \* Seleccionar Server-PT
- \* Pestaña Config > Settings > Display name: Servidor
- \* Pestaña Config > Settings > Gateway (No necesario): 192.168.0.1
- \* Pestaña Config > FastEthernet > Ip address: 192.168.0.2
- \* Pestaña Config > FastEthernet > SubNet Mask: 255.255.255.0
- \* **Configurando el servicio DHCP (Asignación automática de Ips):**
- \* Pestaña Services > DHCP > Default gateway (No necesario): 192.168.0.1
- \* Pestaña Services > DHCP > Start Ip address: 192.168.0.10
- \* Pestaña Services > DHCP > Maximum number of users: 50
- \* Pestaña Services > DHCP > Interface: Service: On
- \* Pestaña Services > DHCP > Seleccionar Save
- \* **Configurando el Computador:**
- \* Seleccionar PC-PT
- \* Pestaña Config > Settings > Display name: MiPC
- \* Pestaña Config > FastEthernet > activar DHCP (Se actualizan los valores automáticamente)
- \* **Configurando el Servidor para IoT:**

- \* Seleccionar el Servidor.
- \* Pestaña Services > IoT > activar ON
- \* Pestaña Desktop > IoT monitor
- \* IoT Server Address: 192.168.0.2, User name: admin, Password: admin, Login
- \* Wrong username or password
- \* Sign up now
- \* User name: admin
- \* Password: admin
- \* Create

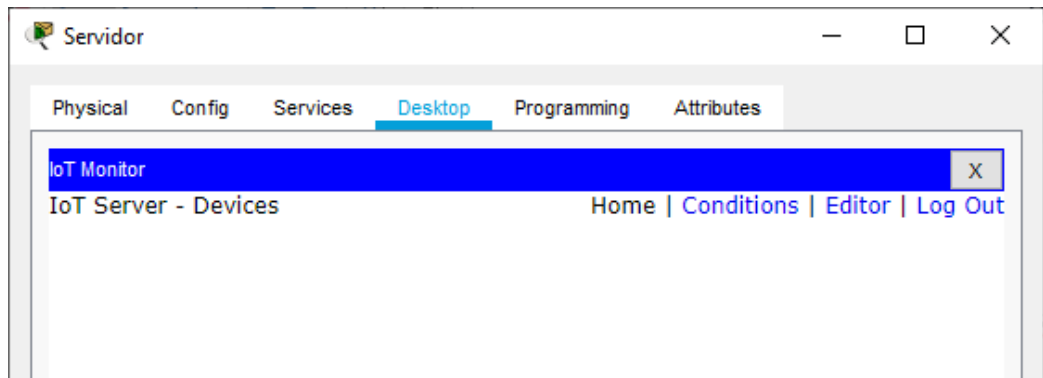
**Registration Server Account Creation**

Username:

Password:

*Figura 17: Controles para iniciar sesión*

- \* (El servidor para IoT ya está configurado, pero aún no aparecen los dispositivos)



*Figura 18: Escritorio del Servidor - Monitor IoT*

- \* **Configurando el dispositivo IoT:**
- \* Seleccionar Light
- \* Pestaña Config > Settings > Display name: Lámpara 01
- \* Pestaña Config > FastEthernet > activar DHCP
- \* Pestaña Config > Settings > activar Remote server, Server address: 192.168.0.2, User name: admin, Password: admin, Connect

#### **4.1.4. Fase 4: Probar, optimizar y documentar el diseño:**

##### **Controlando el dispositivo IoT desde el servidor con el navegador:**

- \* Seleccionar el Servidor.
- \* Pestaña Desktop > Web browser
- \* Url: 192.168.0.2, Go
- \* Username: admin
- \* Password: admin

##### **Controlando el dispositivo IoT desde el Servidor con la aplicación “IoT monitor”:**

- \* Seleccionar el Servidor.
- \* Pestaña Desktop > IoT monitor > Username: admin, Password: admin, Sign in

##### **Controlando el dispositivo IoT desde la Pc con el navegador:**

- \* Seleccionar la Pc
- \* Pestaña Desktop > Web browser
- \* Url: 192.168.0.2, Go
- \* Username: admin
- \* Password: admin

##### **Controlando el dispositivo IoT desde la Pc con la aplicación “IoT monitor”:**

- \* Seleccionar la Pc
- \* Pestaña Desktop > IoT monitor
- \* IoT server address: 192.168.0.2
- \* Username: admin
- \* Password: admin

##### **Controlando el dispositivo IoT desde el Móvil con el navegador:**

##### **Insertando los dispositivos:**

##### **\* Insertando el Móvil:**

- \* End devices > Smart device

##### **\* Insertando el AccessPoint:**

- \* Network devices > Wireless devices > Ap-Pt



### **Conectando los dispositivos:**

#### \* **Conectando el Switch con el AccessPoint**

\* Connections > CruzadoDirecto: Switch - AccessPoint (Port 0)

### **Configurando el AccessPoint:**

\* Seleccionar el AccessPoint

\* Pestaña Config > Settings > Display name: AccessPoint

\* Pestaña Config > Port 1 > SSID: MiCasa

\* Con autenticación:

\* Pestaña Config > Port 1 > Authentication: activar WPA2-PSK, Pass: 123456789

### **Configurando el Móvil:**

\* Seleccionar el Móvil

\* Pestaña Config > Settings > Display name: MiMóvil

\* Pestaña Config > Wireless > SSID: MiCasa

\* Con autenticación:

\* Pestaña Config > Wireless > Authentication: activar WPA2-PSK, Pass: 123456789

### **Controlando el dispositivo IoT desde el Móvil con el navegador/ Aplicación "IoT monitor":**

\* Seleccionar el móvil

\* Pestaña Desktop > Web browser > URL: 192.168.0.2, Username: admin, Password: admin, Sign in.

### **Insertando el dispositivo IoT inalámbrico Light:**

\* End devices > Home > Light

#### \* **Conectando y configurando el dispositivo IoT inalámbrico Light:**

\* Seleccionar el Light

\* Pestaña Config > Settings > Display name: Lámpara 02

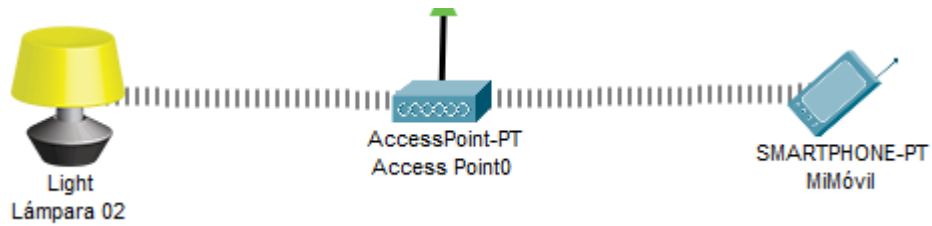
\* Haga clic en Advanced

\* Pestaña I/O config > Network adapter: PT-IOE-NM-1W

\* Pestaña Physical > Verificar la conexión

\* Pestaña Config > Wireless > SSID: MiCasa

- \* Con autenticación:
- \* Pestaña Config > Wireless > Authentication: activar WPA2-PSK, Pass: 123456789, Ipconfiguration: activar DHCP



*Figura 19: Dispositivo Lámpara con Autenticación*

- \* **Configurando para que se conecte al Servidor:**
- \* Seleccionar la Lámpara 02
- \* Pestaña Config > Settings > activar: Remote server, Server address: 192.168.0.2, User name: admin, Password: admin, Connect/Refresh
- \* **Controlando el dispositivo Lámpara 02 desde el Móvil:**
- \* Seleccionar MiMóvil
- \* Pestaña Desktop > Web browser > URL: 192.168.0.2, Username: admin, Password: admin, Sign in.
- \* **Insertando el dispositivo inalámbrico Webcam:**
- \* End devices > Home > Webcam
- \* **Conectando y Configurando el dispositivo inalámbrico Webcam:**
- \* Seleccionar el Webcam
- \* Pestaña Config > Settings > Display name: Webcam
- \* Haga clic en Advanced
- \* Pestaña I/O config > Network adapter: PT-IOE-NM-1W
- \* Pestaña Physical > Verificar la conexión
- \* Pestaña Config > Wireless > SSID: MiCasa
- \* Con autenticación:
- \* Pestaña Config > Wireless > Authentication: activar WPA2-PSK, Pass: 123456789, Ipconfiguration: activar DHCP
- \*

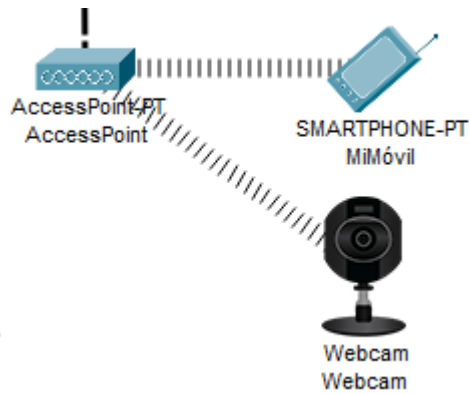


Figura 20: Dispositivo WebCam con Autenticación

- \* **Configurando para que se conecte al Servidor:**
- \* Seleccionar el Webcam
- \* Pestaña Config > Settings > activar: Remote server, Server address: 192.168.0.2, User name: admin, Password: admin, Connect
- \* **Controlando el dispositivo Webcam desde el Móvil:**
- \* Seleccionar el Móvil
- \* Pestaña Desktop > Web browser > URL: 192.168.0.2, Username: admin, Password: admin, Sign in.

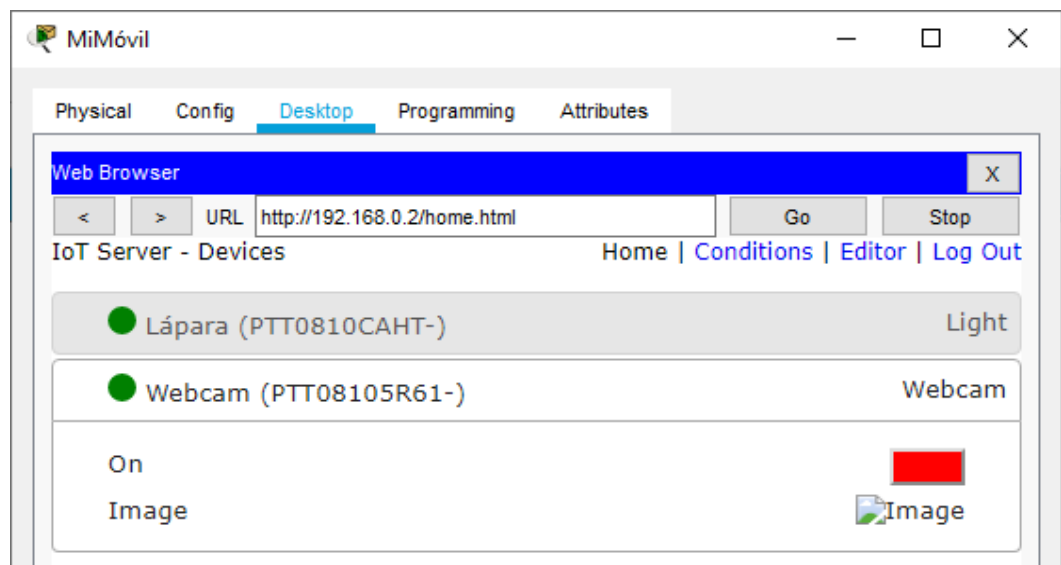


Figura 21: Escritorio de la WebCam - Monitor IoT

### Controlando el dispositivo IoT programando la placa SBC (RaspBerry Pi) y activando con el Push button:

- \* La tarjeta SBC es básicamente una simple placa de computadora que puede ser programada usando un lenguaje de programación. Como sabemos los Leds (considerados como componentes IoT) no tienen una interfaz Ethernet, por tanto, se conectarán por medio de un cable

personalizado de IoT (IoT custom cable) a los pines de entrada/salida de nuestra tarjeta SBC.

#### **Insertando los dispositivos:**

- \* **Placa SBC0:** Components > SBC-Board
- \* **Botón:** Components > Sensors > Toggle Push Button

#### **Configurando los dispositivos:**

- \* Seleccionar SBC-Board
- \* Pestaña Config > Settings > Display name: RaspBerry
- \* Pestaña Config > Wireless > SSID: MiCasa
- \* Con authentication:
- \* Pestaña Config > Wireless > Authentication: activar WPA2-PSK, Pass: 123456789, Ipconfiguration: activar DHCP

#### \* **Configurando para que se conecte al servidor:**

- \* Seleccionar el RaspBerry
- \* Pestaña Config > Settings > activar: Remote server, Server address: 192.168.0.2, User name: admin, Password: admin, Connect/Refresh

#### \* **Controlando el dispositivo Raspberry desde el Móvil:**

- \* Seleccionar MiMóvil
- \* Pestaña Desktop > Web browser > URL: 192.168.0.2, Username: admin, Password: admin, Sign in.

#### **Conectando los dispositivos al Raspberry:**

- \* Connections > IoT custom cable: SBC0 (D0) – Lámpara 01 (D0)
- \* Connections > IoT custom cable: SBC0 (D1) – Push button (D0)
- \* Connections > IoT custom cable: SBC0 (D2) – Lámpara 02 (D0)

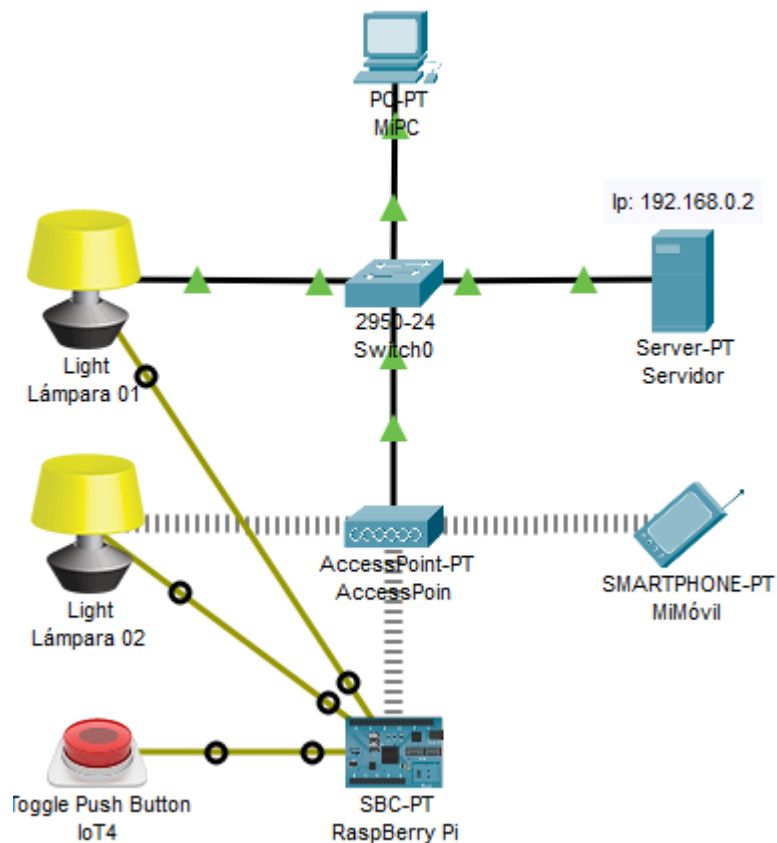


Figura 22: Red con el Dispositivo Raspberry .

### Programando la tarjeta SBC0 con Python:

- \* (Para que se encienda la Lámpara al presionar el Interruptor).
- \* Seleccionar SBC0.
- \* Pestaña Programming > New > Name: main, Type: Empty – Python, Create.
- \* Seleccionar (doble clic) main.py

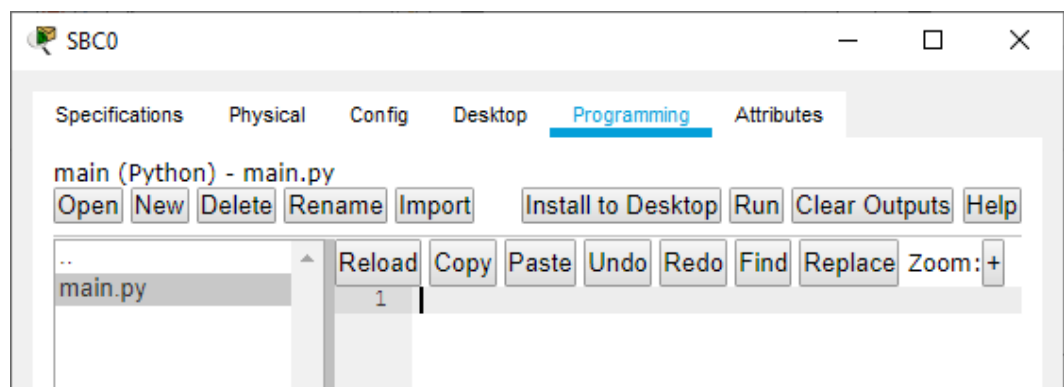


Figura 23: Consola de programación del Dispositivo Raspberry.

- \* **Código en Python:**

```

from gpio import *
from time import *

def main():
    while True:
        value = digitalRead(1)
        if value > 0 :
            customWrite(0,1)
        else :
            customWrite(0,0)
        delay(500)

if __name__ == "__main__":
    main()

```

Tabla 7: Programación en Phyton.

- \* Haga clic en Run
- \* Ejecutando con Alt + Clic

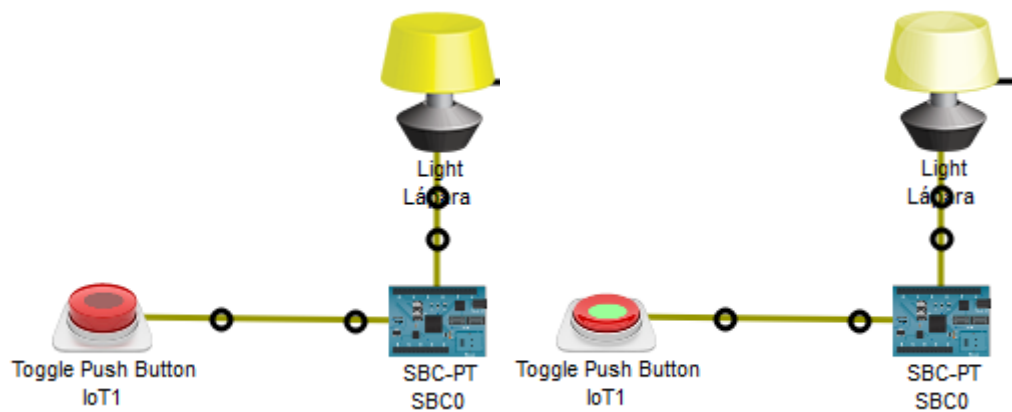


Figura 24: Ejecución directa del dispositivo con Alt + Clic

### Agregando más dispositivos IoT:

- \* End devices > Home > Window
- \* End devices > Home > Door

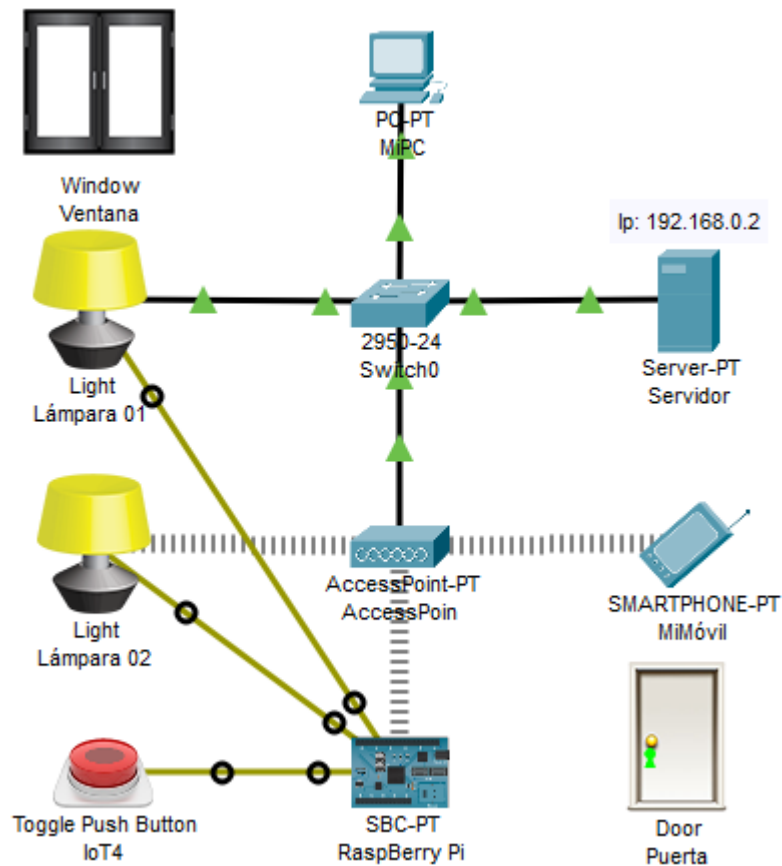


Figura 25: Dispositivos Adicionales.

\* **Cofigurando los nombres:**

\* Seleccionar Window

\* Pestaña Config > Settings > Display name: Ventana

\* Seleccionar Door

\* Pestaña Config > Settings > Display name: Puerta

\* **Conectando los dispositivos IoT:**

\* Connections > IoT custom cable: SBC0 ( D3)-Windows (D0)

\* Connections > IoT custom cable: SBC0 (D4)-Door (D0)

\* Visualizando los Slots digitales:

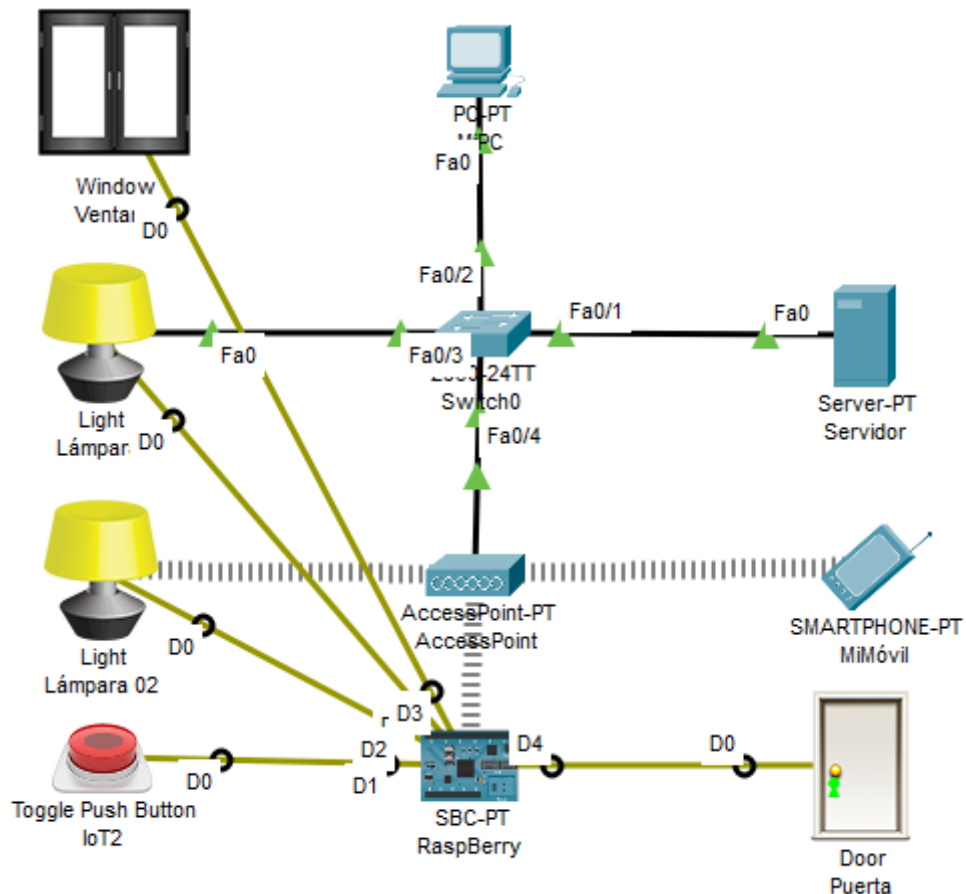


Figura 26: Etiquetas activadas de las conexiones entre los dispositivos.

\* **Se puede agregar un conector más a un dispositivo (Puerta):**

- \* Seleccionar la Puerta
- \* Pestaña Physical
- \* Botón Advanced
- \* Pestaña I/O config > Digital slots: 2

**Programando la placa SBC0 con JavaScript:**

- \* Seleccionar SBC0
- \* Pestaña Programming > New > Name: main2, Type: Empty – JavaScript, Create
- \* Seleccionar (Doble clic) main2.js

**Código en JavaScript:**

\* **Caso 1:**

- \* Sabiendo que la lámpara 01 está conectada por el Slot digital: 0 y sus estados son 0: apagado, 1: encendido medio y 2: encendido máximo.



- \* Se pide, que la lámpara se encienda (Encendido máximo) desde las 18:00 a 6:00 horas.
- \* Caso contrario, que la lámpara se apague.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoL=customRead(0);

  h=18;//Para mdificar la hora.

  delay(2000);
  if((h >= 18) || (h <= 6)){
    customWrite(0,[2,'Máximo',s]);/* Slot_Lampara,[estado,'Comentario',s] */
  }
  else customWrite(0,[0,'Apagado',s]);
}

```

Tabla 8: Código en Java script del Caso I

- \* **Caso 2:**
- \* Sabiendo que la Ventana está conectada por el Slot digital: 3 y sus estados son 0: cerrado y 1: abierto.
- \* Se pide, que la ventana se cierre desde las 19:00 a 8:00 horas.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoV=customRead(0);

  delay(2000);
  if((h >= 19) || (h <= 8)){
    customWrite(3,[0,'Cerrado',s]);/* Slot_Ventana,[estado,'Comentario',s] */
  }
}

```

Tabla 9: Código en Java script del Caso II

- \* **Caso 3:**
- \* Sabiendo que la Puerta está conectada por el Slot digital: 4 y sus estados son 0: cerrado y 1: abierto.

- \* Se pide, que la puerta se cierre desde las 21:00 a 7:00 horas.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoP=customRead(0);

  delay(2000);
  if((h >= 21) || (h <= 7)){
    customWrite(4,[0,'Cerrado',s]);/* Slot_Puerta,[estado,'Comentario',s] */
  }
}

```

Tabla 10: Código en Java script del caso III

- \* **Caso 4:**

- \* Sabiendo que:

- \* La Ventana está conectada por el Slot digital: 3 y sus estados son 0: cerrado y 1: abierto.

- \* La Lámpara está conectada por el Slot digital: 0 y sus estados son 0: apagado, 1: encendido medio y 2: encendido máximo.

- \* Si la Ventana está abierta y la lámpara está apagada, se pide, que la Lámpara se encienda (Encendido medio) desde las 18:00 a 6:00 horas.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoV=customRead(3);
  var estadoL=customRead(0);

  delay(2000);
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6)))
  {
    customWrite(0,[1,'Encendido medio',s]);
  }
}

```

Tabla 11: Código en Java script del caso IV

- \*

- \* **Caso 5:**

- \* Sabiendo que:
- \* La Ventana está conectada por el Slot digital: 3 y sus estados son 0: cerrado y 1: abierto.
- \* La Lámpara está conectada por el Slot digital: 0 y sus estados son 0: apagado, 1: encendido medio y 2: encendido máximo.
- \* Si la Ventana está abierta y la lámpara está apagada, se pide, que la Lámpara se encienda (Encendido medio) y la ventana se cierre desde las 18:00 a 6:00 horas.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoV=customRead(3);
  var estadoL=customRead(0);

  delay(2000);
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6))){
    customWrite(0,[1,'Encendido medio',s]);
  }
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6))){
    customWrite(3,[0,'Cerrado',s]);
  }
}

```

*Tabla 12: Código en Java script del caso V*

- \* **Caso 6:**
- \* Sabiendo que:
- \* La Ventana está conectada por el Slot digital: 3 y sus estados son 0: cerrado y 1: abierto.
- \* La Lámpara está conectada por el Slot digital: 0 y sus estados son 0: apagado, 1: encendido medio y 2: encendido máximo.
- \* La Puerta está conectada por el Slot digital: 4 y sus estados son 0: cerrado y 1: abierto.
- \* Si la Ventana está abierta y la lámpara está apagada desde las 18:00 a 6:00 horas, se pide:
- \* Que la Lámpara se encienda (Encendido medio),
- \* Que la ventana se cierre y
- \* Que la puerta se abra.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoV=customRead(3);
  var estadoL=customRead(0);
  var estadoP=digitalRead(4);

  delay(2000);
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6))){
    customWrite(0,[1,'Encendido medio',s]);
  }
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6))){
    customWrite(3,[0,'Cerrado',s]);
  }
  if(estadoV==1 && estadoL==0 && ((h >= 18) || (h <= 6))){
    customWrite(4,[1,'Abierto',s]);
  }
}
}

```

*Tabla 13: Código en Java script del caso VI*

- \* Haga clic en Run, Alt + Clic o haga clic en Push button.

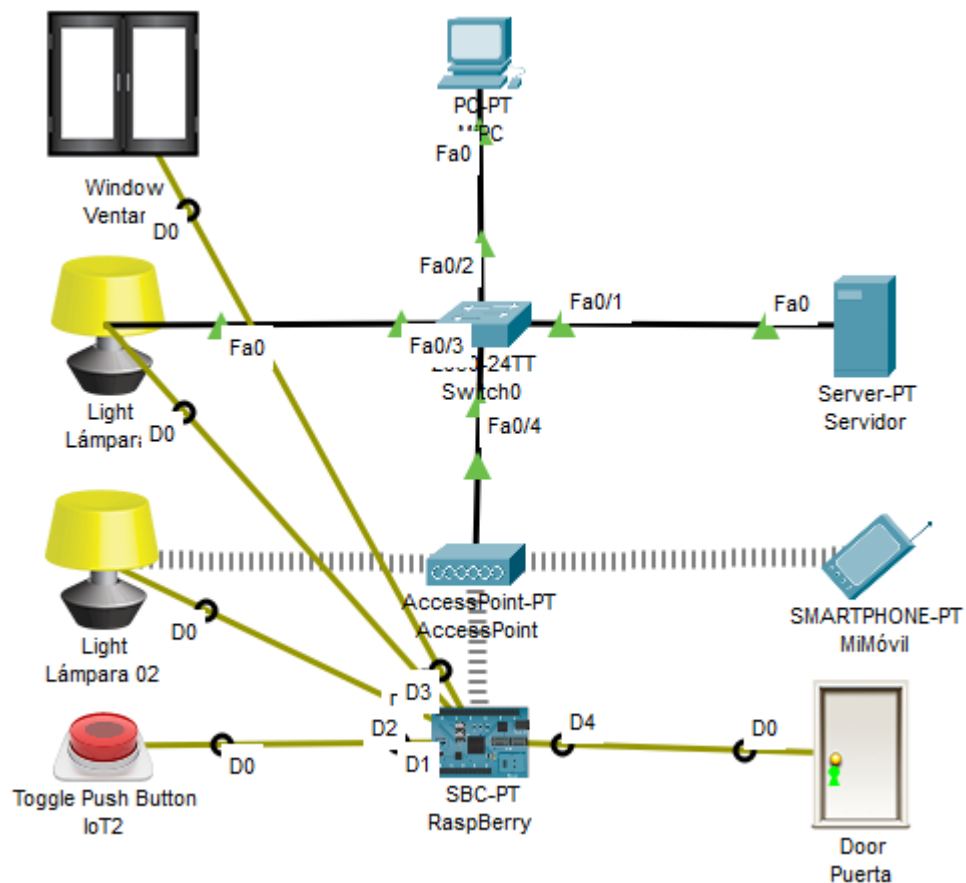


Figura 27: Red perimetral con Dispositivos IoT

## SISTEMA DE CONTROL DE LA ILUMINACIÓN DE LOS LABORATORIOS DE CÓMPUTO DE LA UNIVERSIDAD NACIONAL HERMILIO VALDIZAN BAJO LA ARQUITECTURA DEL COMPUTADOR RASPBERRY PI

- \* **Insertando los dispositivos:**
- \* **Servidor:** End devices > Server
- \* **Switch:** Network devices > Switches > 2950-24
- \* **Pc:** End devices > Pc
- \* **Raspberry:** Componets > SBC board
- \* **Accesspoint:** Wireless devices > Ap-Pt
- \* **Móvil:** End devices > Smart device
- \* **5 Lámparas:** End devices > Home > Light
- \* **Configurando los nombres:**

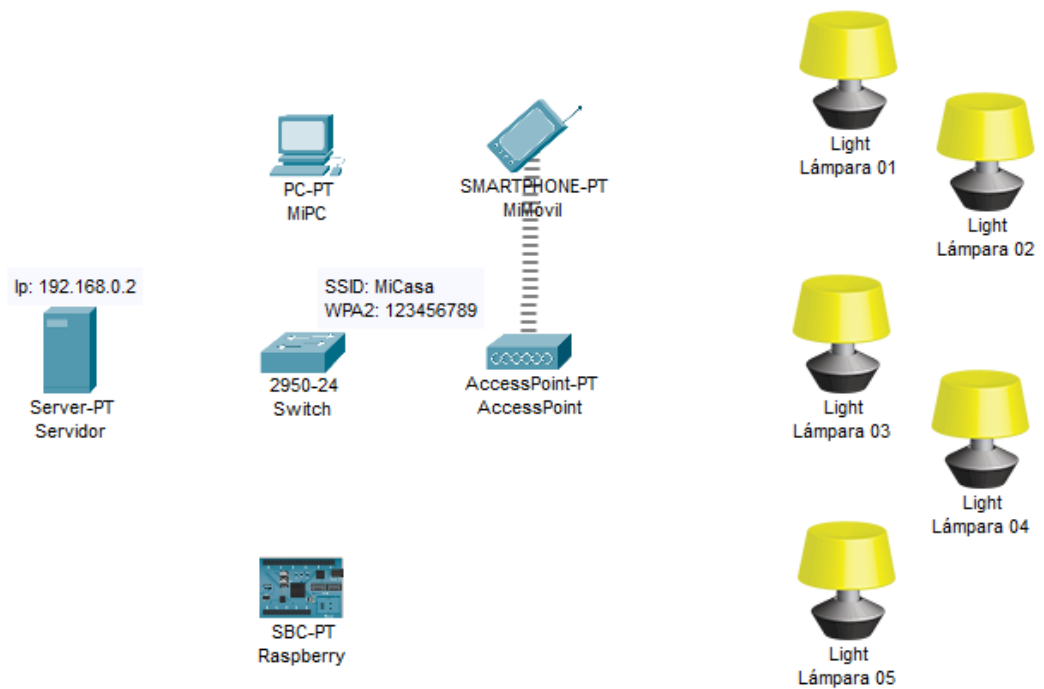


Figura 28: Dispositivos de red sin conexión

- \* **Conectando los dispositivos:**
- \* Connections > Directo: Swith (FastEthernet) – Servidor (FastEthernet)
- \* Connections > Directo: Swith (FastEthernet) – MiPC (FastEthernet)
- \* Connections > Directo Swith (FastEthernet) – AccessPoint (Port 0)
- \* Agregando el módulo Fast Ethernet al Raspberry Pi)
- \* Connections > Directo: Swith (FastEthernet) – Raspberry pi (FastEthernet)
- \* Connections > IoT custom cable: Raspberry pi (D0) – Lámpara 01 (D0)
- \* Connections > IoT custom cable: Raspberry pi (D1) – Lámpara 02 (D0)
- \* Connections > IoT custom cable: Raspberry pi (D2) – Lámpara 03 (D0)
- \* Connections > IoT custom cable: Raspberry pi (D3) – Lámpara 04 (D0)
- \* Connections > IoT custom cable: Raspberry pi (D4) – Lámpara 05 (D0)

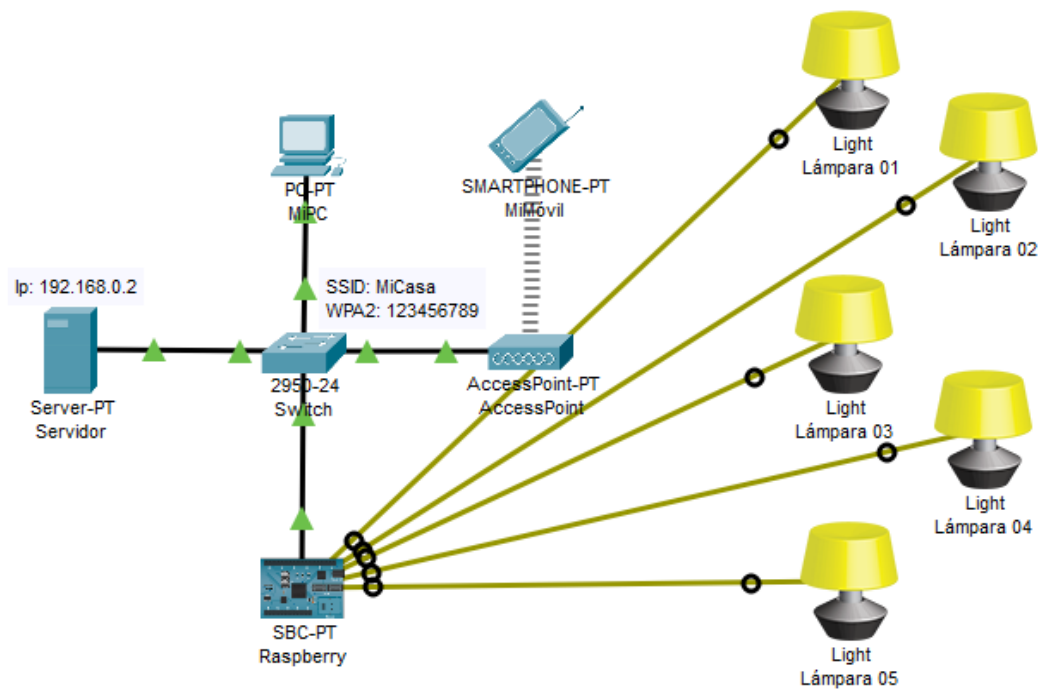


Figura 29:Dispositivos de red con conexión

\* **Configurando los dispositivos:**

\* **Servidor:**

\* Seleccionar el Servidor

\* FastEthernet > Ip configuration > Activar: Static, IpAddress: 192.168.0.2

\* Pestaña Config > Settings > Gateway: 192.168.0.1 (No necesario)

\* Asignación automática de Ips:

\* Pestaña Services > DHCP > Activar: ON

\* Pestaña Services > DHCP > Default gateway: 192.168.0.1 (No necesario)

\* Pestaña Services > DHCP > Start Ip address: 192.168.0.10, Maximum number of users: 50, Save

\* **MiPC:**

\* Seleccionar MiPC

\* Pestaña Config > FastEthernet > Ip configuration > Activar: DHCP

\* **Raspberry Pi:**

\* Seleccionar Raspberry

\* Pestaña Config > FastEthernet > Ip configuration > Activar: DHCP

\* **AccessPoint:**

- \* Seleccionar el AccessPoint
- \* Pestaña Config > Port 1 > SSID: MiCasa, WPA2: 123456789
- \* **MiMóvil:**
- \* Seleccionar MiMóvil
- \* Pestaña Config > Wireless > SSID: MiCasa, WPA2: 123456789, Activar: DHCP
- \* Desde el Servidor, MiPC, MiMóvil y Raspberry se puede acceder al servidor (192.168.0.2) solamente con el navegador web, pero a la página genérica.

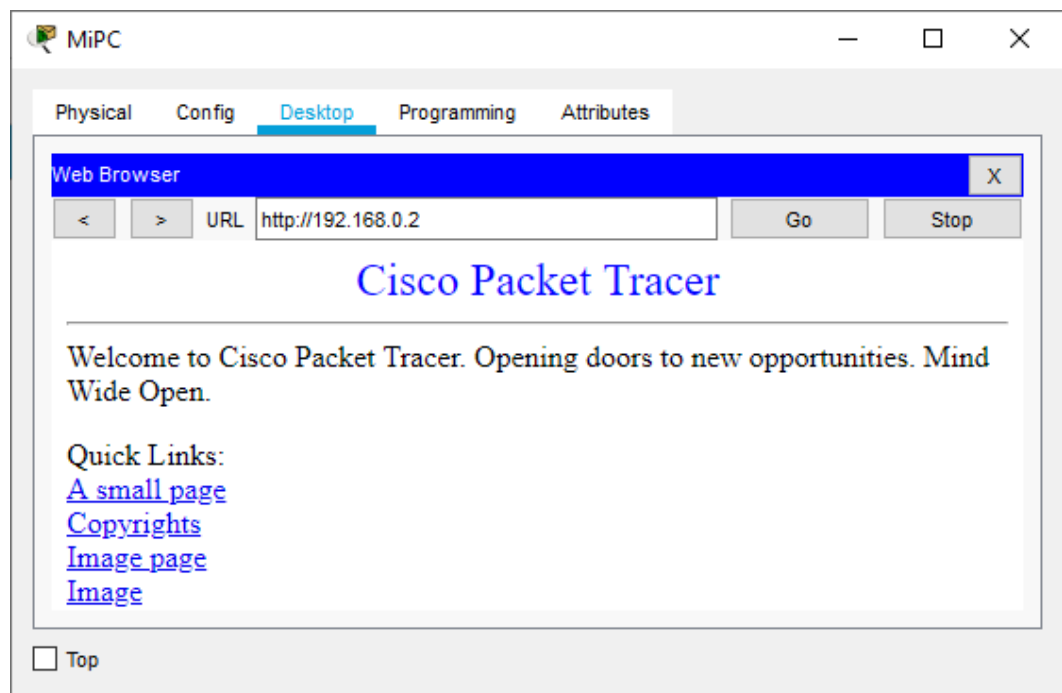
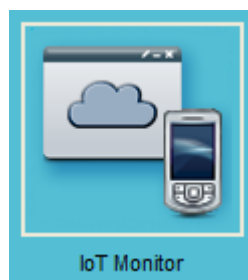


Figura 302: Escritorio de MiPC y el navegador con la página de Cisco

- \* **Configurando el Servidor para IoT**
- \* Seleccionar el Servidor
- \* Pestaña Services > IoT > Activar: ON
- \* Pestaña Desktop > IoT monitor:





- \* Desde el Servidor, MiPC, MiMóvil y Raspberry se puede acceder al servidor (192.168.0.2) tanto con el Navegador web como con la Aplicación IoT monitor, pero hay que crear un Usuario y Contraseña (Username: admin y Password: admin).

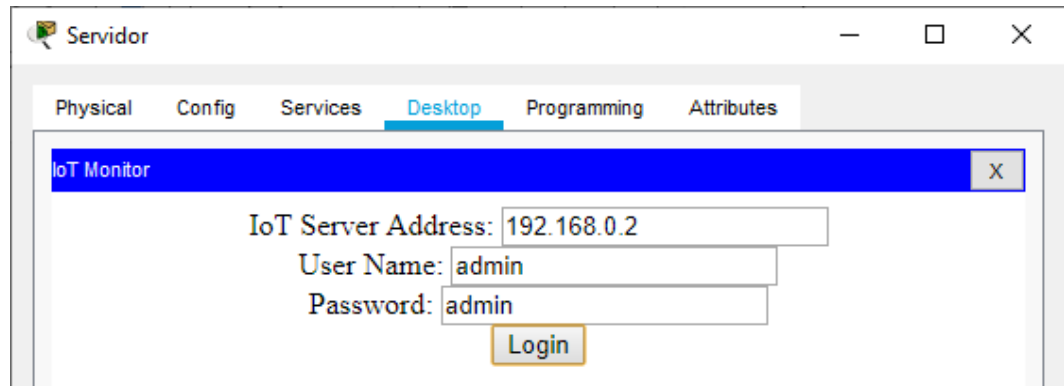


Figura 31: Escritorio del Servidor y la Aplicación Monitor IoT

- \* Wrong username or password
- \* Sign up now
- \* Create

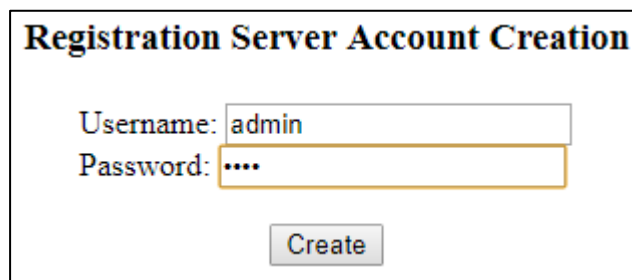


Figura 32: Interfaz para crear Usuario y Contraseña.

- \* El servidor para IoT ya está configurado. Se puede acceder a la ventana de los dispositivos IoT (Lámparas) pero están vacías.
- \* **Configurando los dispositivos IoT (Lámparas):**
- \* Seleccionar cada dispositivo IoT
- \* Agregando el módulo PT-IoT-NM-1W a los dispositivos IoT
- \* Pestaña Config > Wireless > SSID: MiCasa, WPA2: 123456789, Activar DHCP
- \* Configurando para que se conecten al servidor:
- \* Pestaña Config > Settings > Activar: Remote server, 192.168.0.2, admin, admin. Connect/ Refresh
- \* **Controlando los dispositivos IoT desde el Servidor, MiPC, MiMóvil y Raspberry:**

- \* Desde el Servidor, MiPC, MiMóvil y Raspberry se puede acceder al servidor (192.168.0.2) y a la ventana de los dispositivos IoT (Lámparas) y se pueden visualizar todos los dispositivos IoT. También pueden ser controlados.

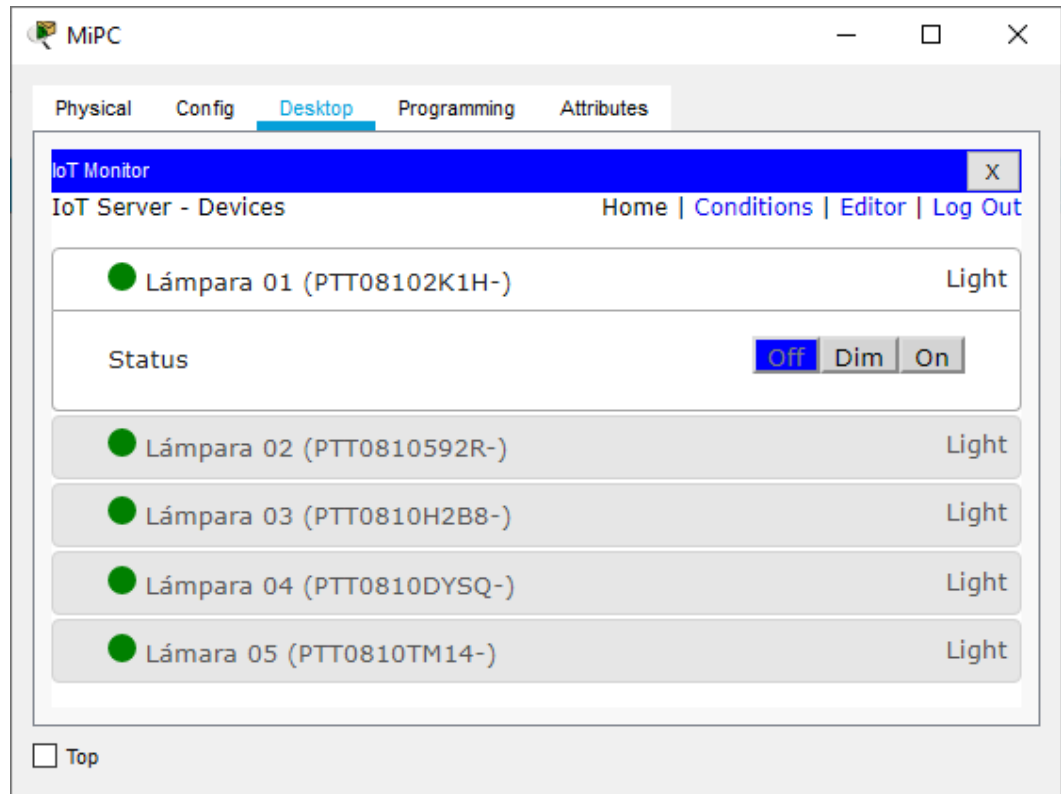


Figura 33: Ventana de los dispositivos IoT en la Aplicación Monitor IoT

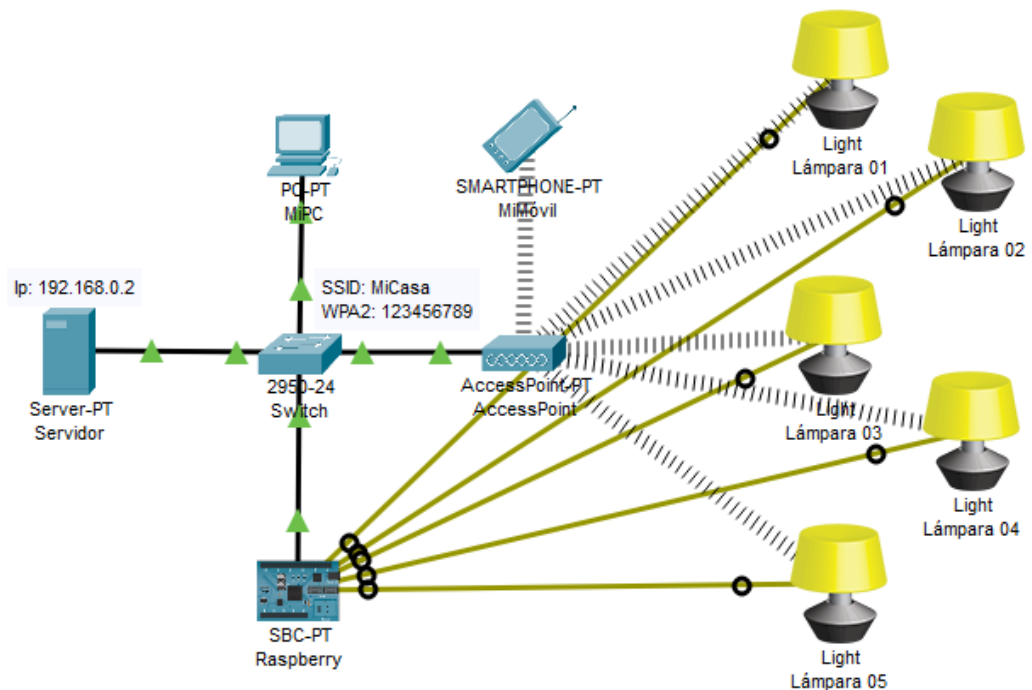


Figura 34: Dispositivos de red con sus respectivas conexiones.

- \* ¿Cómo acceder a los dispositivos desde el celular al Raspberry?
- \* **Insertando alguna imagen de fondo:**
- \* Seleccionar Set background image.
- \* Seleccionar Browse y seleccionar la imagen: “Plano CEI - Nuevo - Sin fluorescentes”
- \* Activar Use original image
- \* Seleccionar Apply.
- \* **Posicionando las lámparas según el ambiente:**
- \* Lámpara 01 -> Laboratorio 01
- \* Lámpara 02 -> Laboratorio 02
- \* Lámpara 03 -> Laboratorio 03
- \* Lámpara 04 -> Pasadizo
- \* Lámpara 05 -> Sala de espera

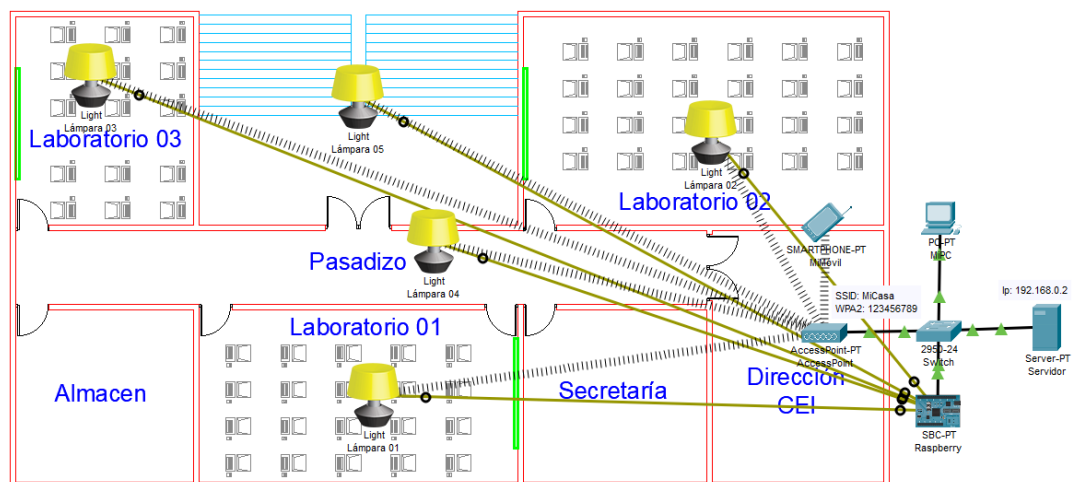


Figura 35: Disposición de los dispositivos IoT según ubicación.

- \* **Automatizando el encendido y apagado de los dispositivos IoT (Lámparas) programando el Raspberry Pi:**
- \* Apagar el AccessPoint.
- \* Seleccionar Raspberry
- \* Pestaña Programming > New > Name: main, Type: Empty – JavaScript
- \* Seleccionar main.js

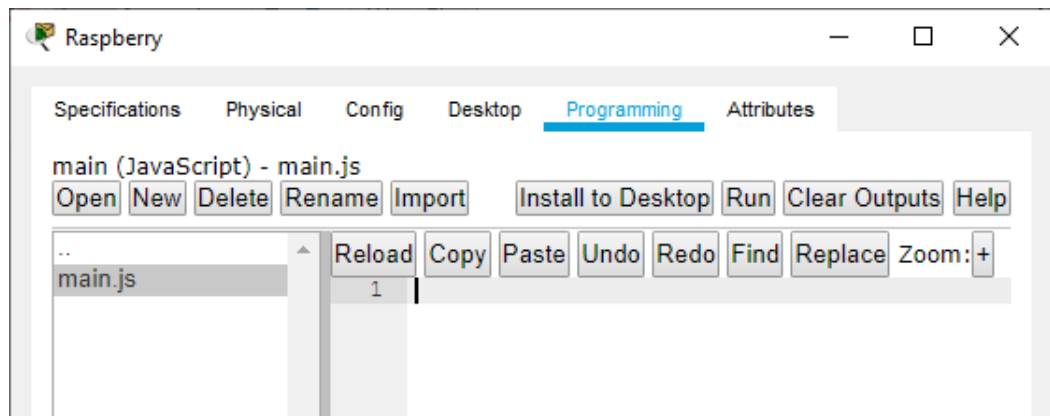


Figura 36: Consola de programación con Java

- \* **Caso Ejemplo:**
- \* Sabiendo que la lámpara 01 está conectada por el Slot digital: 0 y sus estados son 0: apagado, 1: encendido medio y 2: encendido máximo.
- \* Se pide, que la lámpara se encienda (Encendido máximo) desde las 18:00 a 6:00 horas.
- \* Caso contrario, que la lámpara se apague.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  var estadoL=customRead(0);

  h=18;//Para mdificar la hora.

  delay(2000);
  if((h >= 18) || (h <= 6)){
    customWrite(0,[2,'Máximo',s]);/* Slot_Lampara,[estado,'Comentario',s] */
  }
  else customWrite(0,[0,'Apagado',s]);
}

```

Tabla 14: Código en Java para el Caso Ejemplo

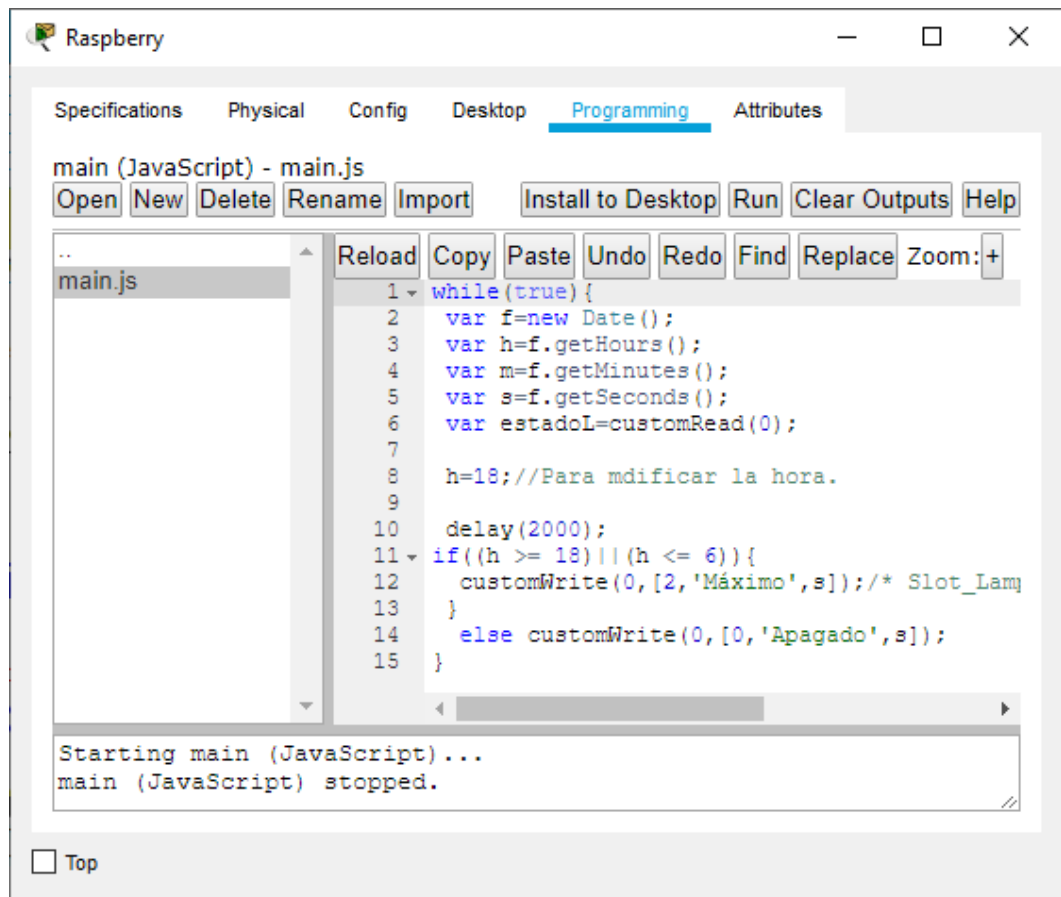


Figura 37: Consola con el Código en Java para el Caso Ejemplo

\*

\* **Objetivos específicos:**

\* **Caso 1:**

\* La falta de control en el tema de consumo de energía, debido a que en algunas ocasiones las luces están encendidas en ambientes vacíos. Generando pérdida de tiempo en el personal administrativo al recurrir a cada interruptor para el encendido o apagado de cada fluorescente. Y otras veces las luces se quedan encendidas hasta el día siguiente.

\* Programar las Lámparas de los laboratorios 01, 02 y 03 de tal forma que se enciendan (Encendido medio) de 07:00 a 21:00. Caso contrario que estén apagadas.

```

while(true){
    var f=new Date();
    var h=f.getHours();
    var m=f.getMinutes();
    var s=f.getSeconds();
    //var estadoV=customRead(0);
    var estadoL1=customRead(0);
    var estadoL2=customRead(1);
    var estadoL3=customRead(2);
    var estadoL4=customRead(3);
    var estadoL5=customRead(4);
    //var estadoP=digitalRead(6);

    h=6;

    delay(2000);
    if(((h >= 7) || (h <= 21))){
        customWrite(0,[1,'Encendido medio',s]);
    }
    if(((h >= 7) || (h <= 21))){
        customWrite(1,[1,'Encendido medio',s]);
    }
    if(((h >= 7) || (h <= 21))){
        customWrite(2,[1,'Encendido medio',s]);
    }
    //Caso contrario
    if(((h >= 22) || (h <= 6))){
        customWrite(0,[0,'Apagado',s]);
    }
    if(((h >= 22) || (h <= 6))){
        customWrite(1,[0,'Apagado',s]);
    }
    if(((h >= 22) || (h <= 6))){
        customWrite(2,[0,'Apagado',s]);
    }
}

```

Tabla 15: Código en Java para el Caso I

- \* **Caso 2:**
- \* Escasa iluminación en las separaciones de los ambientes (Pasadizo).
- \* Programar la Lámpara del pasadizo (Lámpara 04) de tal forma que se encienda (Encendido máximo) de 07:00 a 13:00 y de 15:00 a 21:00. Caso contrario que se apaguen.

```

while(true){
var f=new Date();
var h=f.getHours();
var m=f.getMinutes();
var s=f.getSeconds();
//var estadoV=customRead(0);
var estadoL1=customRead(0);
var estadoL2=customRead(1);
var estadoL3=customRead(2);
var estadoL4=customRead(3);
var estadoL5=customRead(4);
//var estadoP=digitalRead(6);

h=14;

delay(2000);
if(((h >= 7) || (h <= 13)) && ((h >= 15) || (h <= 21))){
  customWrite(3,[2,'Encendido máximo',s]);
}
//Caso contrario
if((h == 14) || ((h >= 22) || (h <= 6))){
  customWrite(3,[0,'Apagado',s]);
}
}
}

```

*Tabla 16: Código en Java para el Caso II*

- \* **Caso 3:**
- \* Escasa iluminación en el área de espera (Escaleras).
- \* Programar la Lámpara de las escaleras (Lámpara 05) de tal forma que se encienda (Encendido medio) de 07:00 a 21:00.

```

while(true){
  var f=new Date();
  var h=f.getHours();
  var m=f.getMinutes();
  var s=f.getSeconds();
  //var estadoV=customRead(0);
  var estadoL1=customRead(0);
  var estadoL2=customRead(1);
  var estadoL3=customRead(2);
  var estadoL4=customRead(3);
  var estadoL5=customRead(4);
  //var estadoP=digitalRead(6);

  h=14;

  delay(2000);
  if(((h >= 7) || (h <= 21))){
    customWrite(4,[1,'Encendido medio',s]);
  }
  //Caso contrario
  if(((h >= 22) || (h <= 6))){
    customWrite(4,[0,'Apagado',s]);
  }
}

```

\*

*Tabla 17: Código en Java para el Caso III*

\* **Integrando todos los casos:**



```

while(true){
    var f=new Date();
    var h=f.getHours();
    var m=f.getMinutes();
    var s=f.getSeconds();
    //var estadoV=customRead(0);
    var estadoL1=customRead(0);
    var estadoL2=customRead(1);
    var estadoL3=customRead(2);
    var estadoL4=customRead(3);
    var estadoL5=customRead(4);
    //var estadoP=digitalRead(6);
    h=14;
    delay(2000);
    if(((h >= 7) || (h <= 21))){
        customWrite(0,[1,'Encendido medio',s]);
    }
    if(((h >= 7) || (h <= 21))){
        customWrite(1,[1,'Encendido medio',s]);
    }
    if(((h >= 7) || (h <= 21))){
        customWrite(2,[1,'Encendido medio',s]);
    }
    if(((h >= 7) || (h <= 13))&&((h >= 15) || (h <= 21))){
        customWrite(3,[2,'Encendido máximo',s]);
    }
    if(((h >= 7) || (h <= 21))){
        customWrite(4,[1,'Encendido medio',s]);
    }
    //Caso contrario
    if(((h >= 22) || (h <= 6))){
        customWrite(0,[0,'Apagado',s]);
    }
    if(((h >= 22) || (h <= 6))){
        customWrite(1,[0,'Apagado',s]);
    }
    if(((h >= 22) || (h <= 6))){
        customWrite(2,[0,'Apagado',s]);
    }
    if((h == 14) || ((h >= 22) || (h <= 6))){
        customWrite(3,[0,'Apagado',s]);
    }
    if(((h >= 22) || (h <= 6))){
        customWrite(4,[0,'Apagado',s]);
    }
}
}

```

Tabla 18: Código en Java para el Caso Integro

## 4.2. Resultados:

- \* Panel de control de openhab2 en localhost

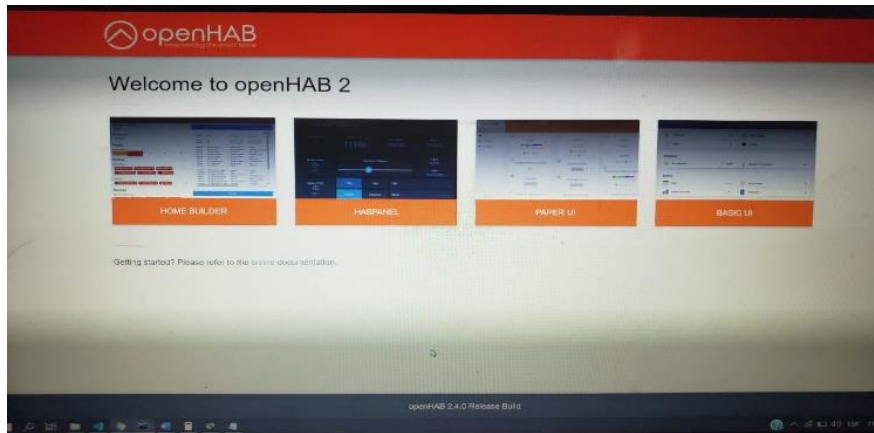


Figura 38: Panel de control de openhab2 en localhost

- \* Panel control para agregar el foco

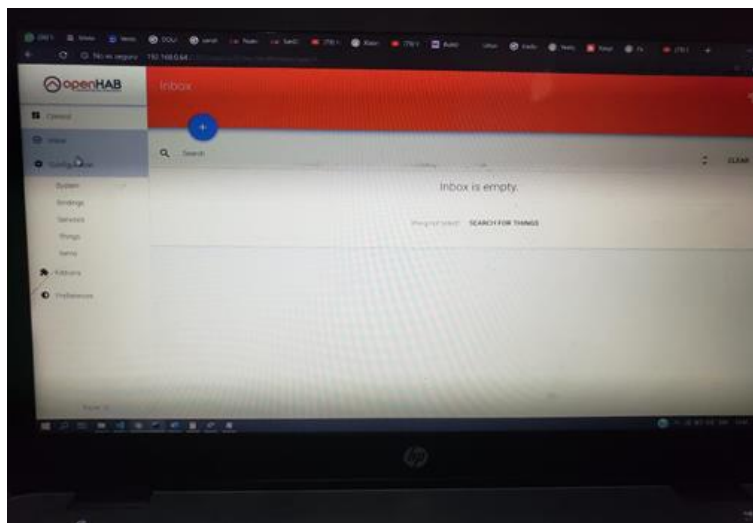


Figura 39: Panel control para agregar el foco

- \* Agregar el binding de Xiaomi

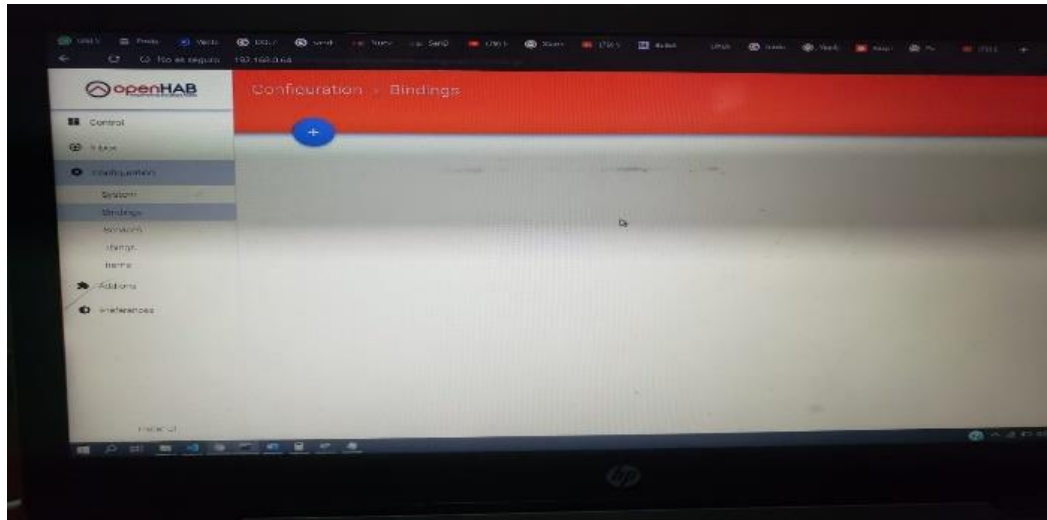


Figura 40: Agregar el binding de Xiaomi

\* Encontró el binding de Xiaomi

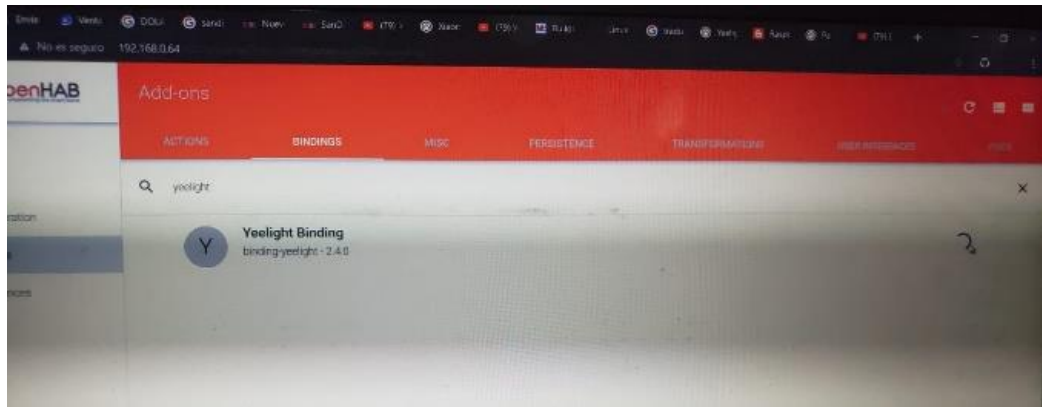


Figura 41: Encontró el binding de Xiaomi

\* Prueba de primer panel con los debidos canales, en modo de switch

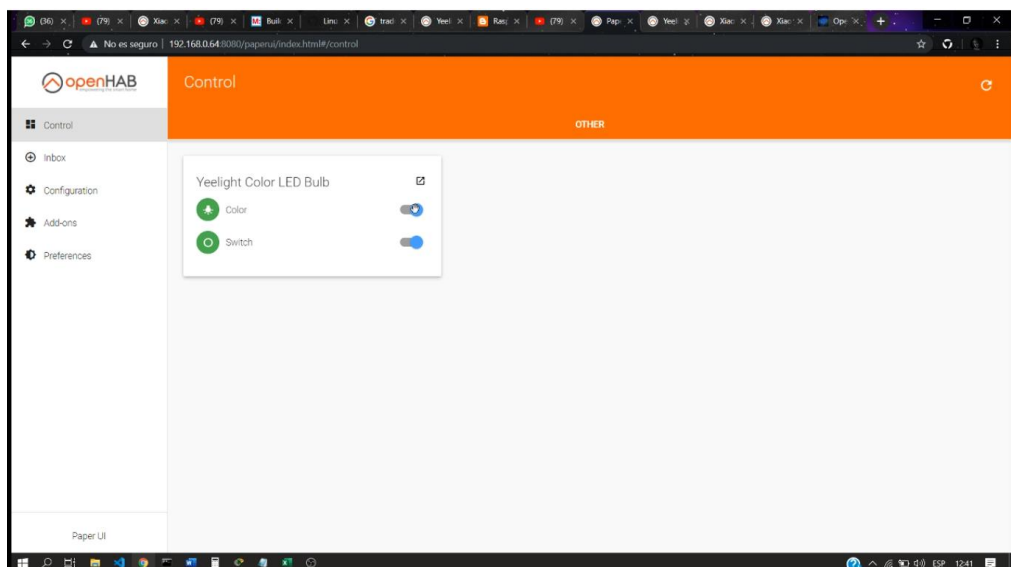


Figura 42: Canales, en Modo de Switch

\* Vista de propiedades del foco yeelight

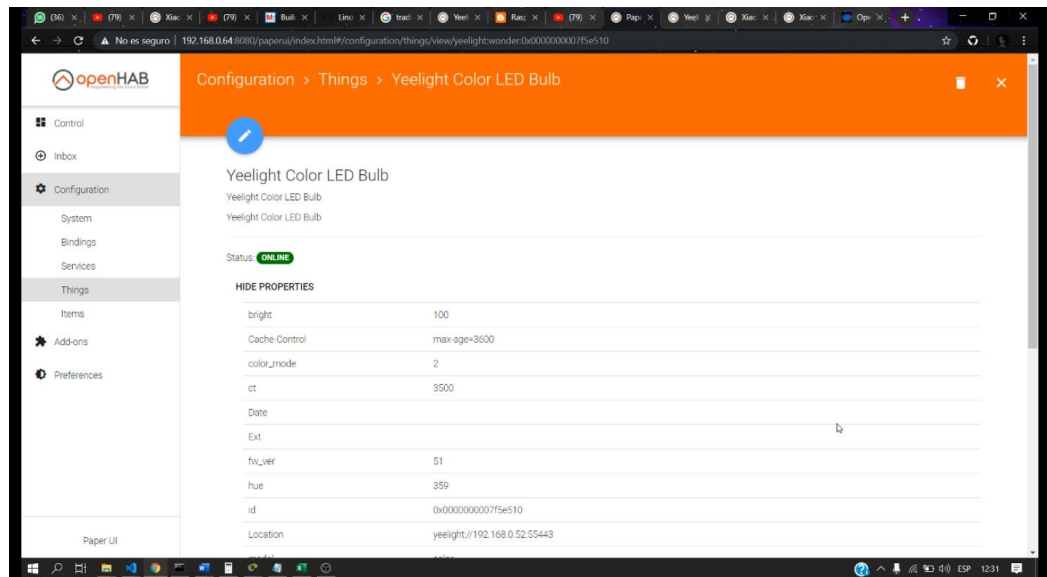


Figura 43: Propiedades del foco inteligente

\* Visualización de los canales del foco

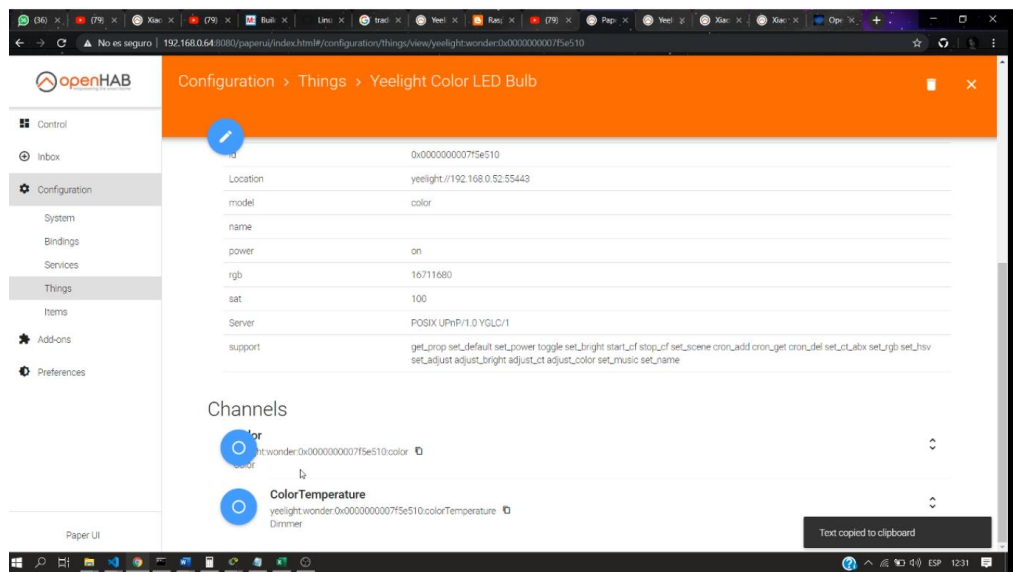


Figura 44: Canales del Foco.

\* Menú de creación de ítems (necesario para la ubicación en el panel)

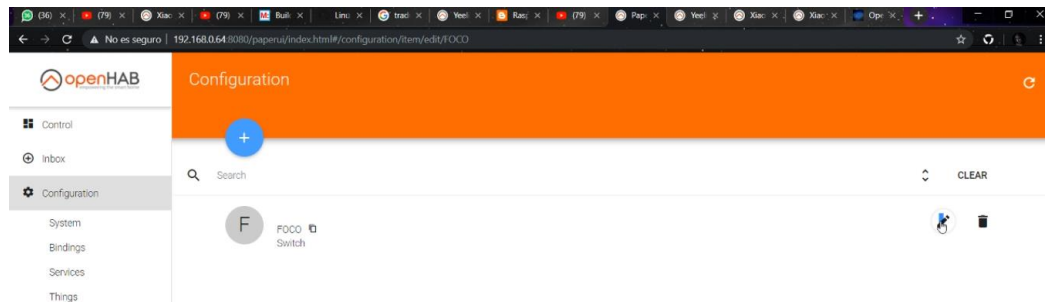


Figura 45: Menú de Items

\* Visualización del id del foco yeelight

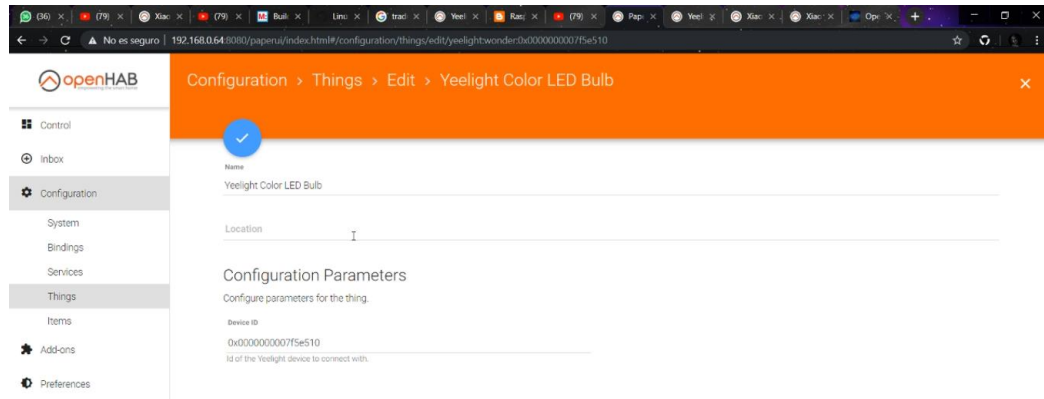


Figura 46: Id del Foco Yeelight.

\* Creación de un ítem asociado a un channel del foco

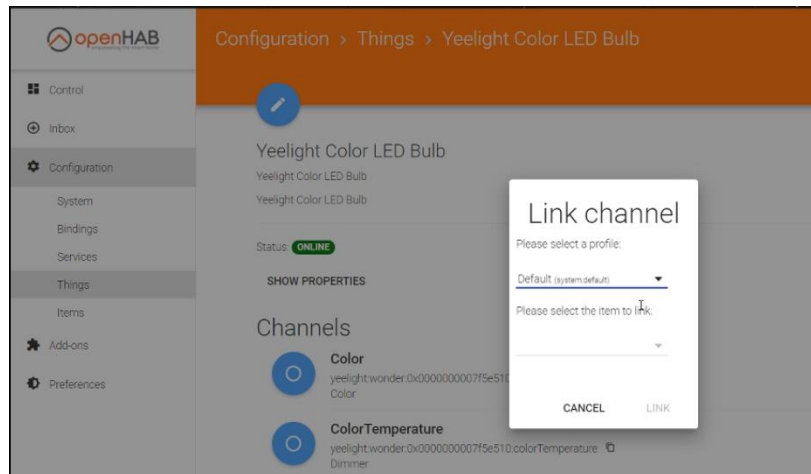


Figura 47: Item asociado a un canal del Foco

\* Menú de enlace del ítem a un channel

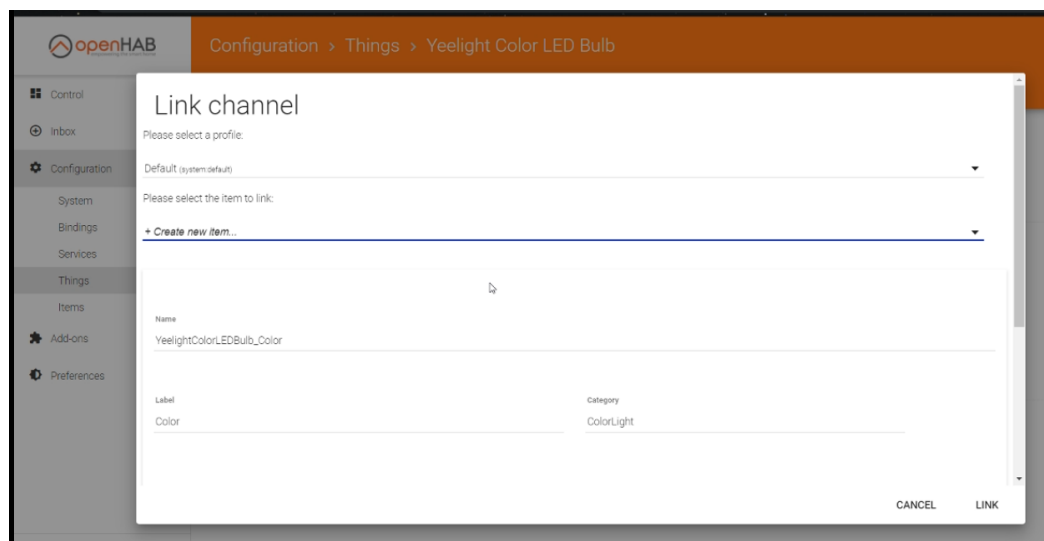


Figura 48: Menu de Enlace del Item

- \* Visualización del ítem asociado al channel en el panel principal

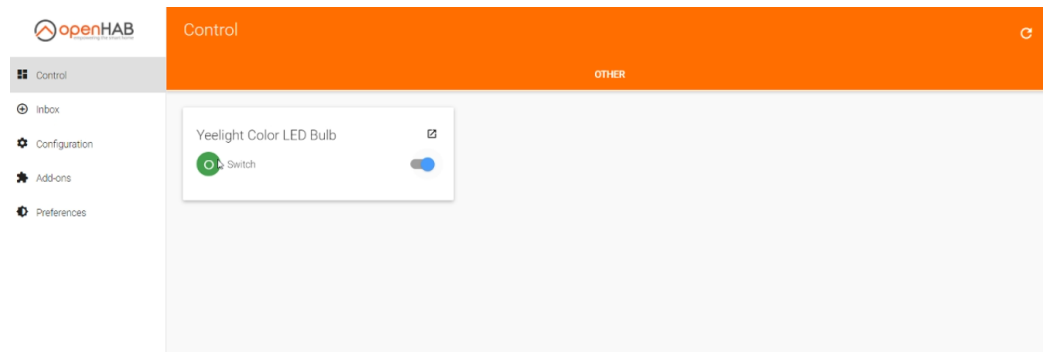


Figura 49: Panel principal del Item

- \* Visualización de los modelos soportados por binding de xiaomi

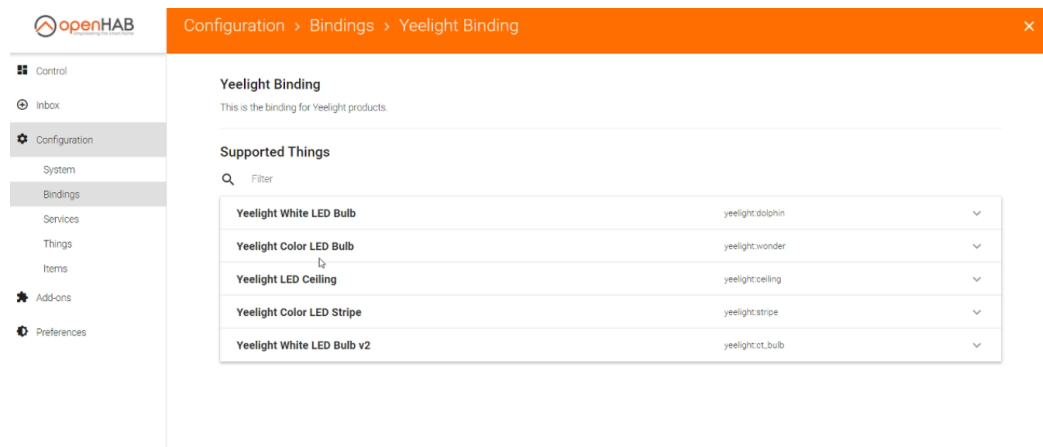


Figura 50: Modelos soportados por el Binding de Xiaomi.

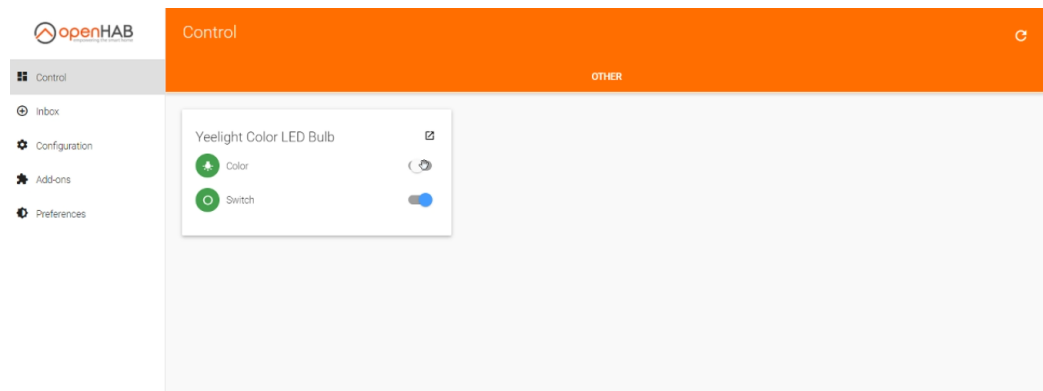


Figura 51: Panel finalizado del Foco de Xiaomi

- \* Finalmente se logró el control del encendido de las luces mediante el servidor OpenHab, este modelo a escala se puede implementar fácilmente para todo un ambiente bajo una red de área local.



*Figura 52: Foco encendido mediante el Servidor OpenHab*

## REFERENCIAS

- Domodesk. (2019). *Domodesk*. Obtenido de <https://www.domodesk.com/221-a-fondo-que-es-iot-el-internet-de-las-cosas.html>
- Evans, D. (Abril de 2017). *https://www.cisco.com*. Obtenido de Cisco Internet Business Solutions Group (IBSG) Cisco IBSG © 2011 Cisco y/o sus filiales. Todos los derechos reservados.04/11 Informe técnico Internet de las cosas Cómo la próxima evolución de Internet lo cambia todo: [https://www.cisco.com/c/dam/global/es\\_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf](https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf)
- Gracia, M. (2019). *Deloitte*. Obtenido de <https://www2.deloitte.com/es/es/pages/technology/articles/loT-internet-of-things.html>
- Hat, R. (2019). *Edge computing*. Obtenido de <https://www.redhat.com/es/topics/edge-computing/what-is-edge-computing>
- Miranda, C. V. (2020). *Sistemas informáticos y redes locales*. Madrid, España: Paraninfo.
- Oracle. (2019). *Oracle*. Obtenido de <https://www.oracle.com/co/internet-of-things/what-is-iot.html>
- Packard, H. (2019). *Hewlett Packard*. Obtenido de <https://www.hpe.com/lamerica/es/what-is/internet-of-things.html>
- Ramírez Madrid, D. A. (2018). *Metodología para la implementación de IoT*. Bogota.
- Saavedra, J. C. (18 de 06 de 2017). *Metodología Top-Down para el Diseño de Redes*. Obtenido de <http://juancarlossaavedra.me/2017/06/infografia-metodologia-top-down-para-el-diseno-de-redes/>
- Upcommons. (s.f). *Portal de acceso abierto al conocimiento de la UPC*. Obtenido de <https://upcommons.upc.edu/bitstream/handle/2099.1/3330/34059-5.pdf?sequence=5&isAllowed=y>
- Zanoni, L. (17 de enero de 2019). *Telefonica Business Solutions*. Obtenido de <https://www.business-solutions.telefonica.com/es/cloud-hub/knowledge-center/what-is-edge-computing/>



## CONCLUSIONES

Se elaboró la propuesta de la red sistematizando el control del encendido de las luces mediante una red de área local empleando la tecnología Internet de las cosas y el servidor Raspberry Pi conjuntamente con la solución OpenHab para el control de las luces inteligentes.

Se elaboró el diseño y también se realizó la emulación de la red empleando el software Packet tracer que permitió la construcción del diseño y aplicar la emulación del funcionamiento de la red y ver el funcionamiento de cada componente usando el Internet de las cosas, en la emulación se controló el encendido de las luces de la red y otros dispositivos similares.

Se Instaló y se configuro la distribución OpenHab para realizar la conexión de la bombilla con el servidor y así poder controlar el encendido y apagado. La instalación fue realizada en el computador Raspberry Pi para luego conectarlo a una red local y probar el control del encendido de las luces inteligentes.

Finalmente se podría concluir el estudio proyectando los resultados del mismo a un entorno físico real y a gran escala en la cual desde el mismo servidor de puedan controlar otros dispositivos como cortinas, sensores, puertas y demás.

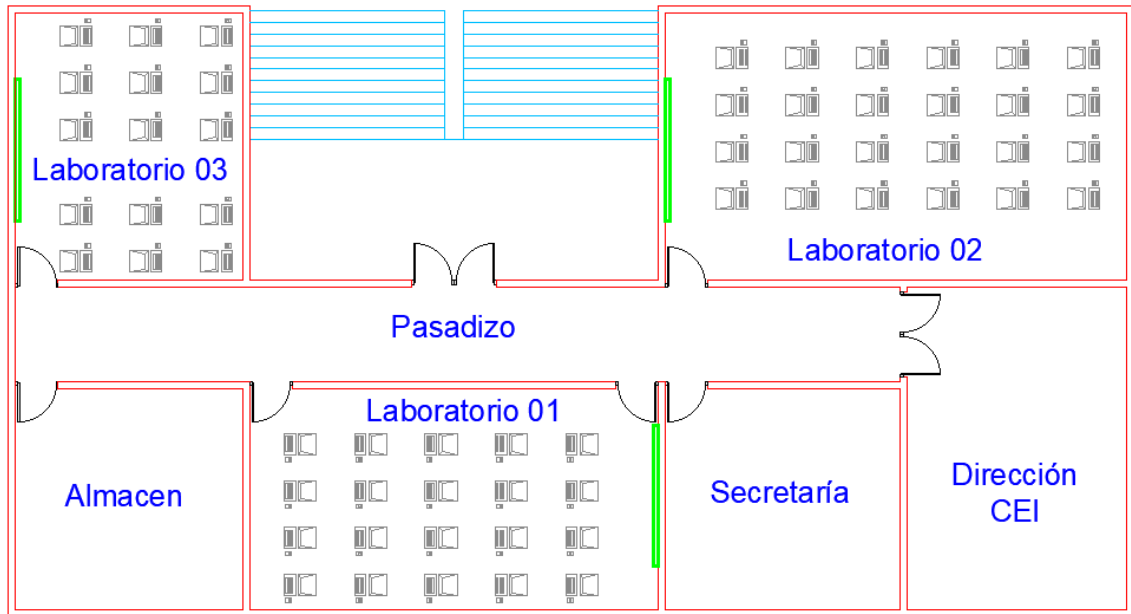
## ANEXOS

### Fotografías:

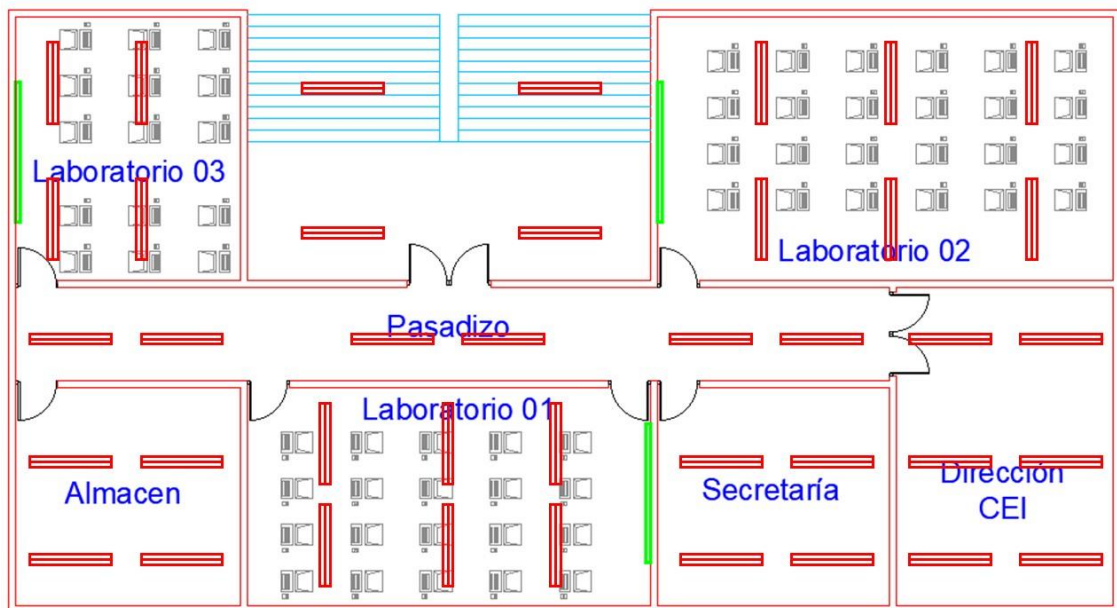


**Imágenes:**

Plano de los laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizan – CEI:



Plano de los laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizan – CEI con sus respectivos fluorescentes:



Plano de los laboratorios de Cómputo de la Universidad Nacional Hermilio Valdizan – CEI con las bombillas inteligentes:

