



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Trabajo de fin de Carrera titulado:

Implementación de Redes Neuronales Artificiales en el Control Domótico de
Iluminación en una Vivienda

Realizado por:

Allen Efrén Wilchez Ramón

Director del proyecto:

MsC. Diego Fernando Bustamante Villagómez

Como requisito para la obtención del título de:

MAGISTER EN DISEÑO INDUSTRIAL Y DE PROCESOS

QUITO, OCTUBRE del 2022

DECLARACIÓN JURAMENTADA

Yo, ALLEN EFRÉN WILCHEZ RAMÓN, ecuatoriano, con Cédula de ciudadanía N° 1721305454, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido presentado anteriormente para ningún grado o calificación profesional, y se basa en las referencias bibliográficas descritas en este documento.

A través de esta declaración, cedo los derechos de propiedad intelectual a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido en la Ley de Propiedad Intelectual, reglamento y normativa institucional vigente.

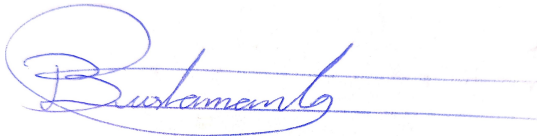


Allen Efrén Wilchez Ramón

C.I.: 1721305454

DECLARACIÓN DEL DIRECTOR DE TESIS

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

A handwritten signature in blue ink, appearing to read 'Bustamante', is written over a horizontal line. The signature is fluid and cursive.

MsC. Diego Fernando Bustamante Villagómez

LOS PROFESORES INFORMANTES:

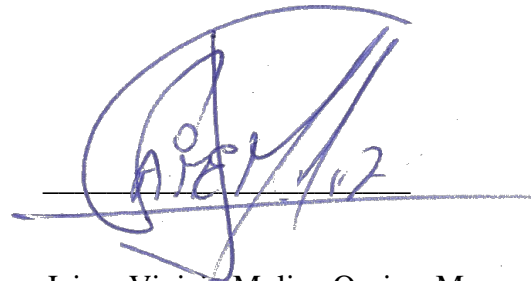
PhD. (c) Diana Belén Peralta Zurita Msc.

Jaime Vinicio Molina Osejos, Msc

Después de revisar el trabajo presentado lo han calificado como apto para su
defensa oral ante el tribunal examinador.

A handwritten signature in blue ink, appearing to be 'Diana Belén Peralta Zurita', written over a horizontal line.

PhD. (c) Diana Belén Peralta Zurita Msc.

A handwritten signature in blue ink, appearing to be 'Jaime Vinicio Molina Osejos', written over a horizontal line.

Jaime Vinicio Molina Osejos, Msc

Quito, 18 de octubre de 2022

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

A handwritten signature in blue ink, appearing to be 'Allen Efrén Wilchez Ramón', written on a light-colored background.

Allen Efrén Wilchez Ramón

C.I.: 1721305454

Índice

Glosario.....	12
Resumen	13
Abstract	14
Introducción.....	15
<i>Antecedentes.....</i>	<i>15</i>
Estado del arte	20
<i>Domótica.....</i>	<i>20</i>
Aporte de la domótica.....	21
<i>Transmisión de datos</i>	<i>22</i>
<i>Inteligencia Artificial.....</i>	<i>22</i>
<i>Machine Learning.....</i>	<i>22</i>
<i>Redes neuronales artificiales.....</i>	<i>23</i>
Funciones de una neurona artificial	24
Capas de la estructura de una neurona artificial	25
Método.....	26
<i>Topología.....</i>	<i>30</i>
<i>Equipo por utilizar</i>	<i>31</i>
<i>Conexiones</i>	<i>32</i>

<i>Flujo de envío y recepción de datos</i>	33
<i>Experimentos</i>	35
Experimento 1	35
Experimento 2	35
Experimento 3	35
Experimento 4	36
Resultados	41
<i>Experimento 1</i>	41
<i>Experimento 2</i>	42
<i>Experimento 3</i>	43
<i>Experimento 4</i>	46
Análisis de resultados, conclusiones y recomendaciones	51
Análisis de resultados	51
Conclusiones	54
Recomendaciones	55
Anexos	59
<i>Anexo A Programación del sistema completo</i>	59
<i>Anexo B Programación de la red neuronal</i>	71
<i>Anexo C Almacenamiento de datos</i>	75
<i>Anexo D Planilla</i>	78

Lista de Figuras

Figura 1 Modelo de la neurona artificial	23
Figura 2 Funciones de una neurona artificial	25
Figura 3 Estructura de capas.....	26
Figura 4 Plano de vivienda general	27
Figura 5 División de la vivienda por secciones	28
Figura 6 Cableado de iluminarias.....	29
Figura 7 Distribución cableado sensores, relés e interruptores	29
Figura 8 Topología de RED.....	30
Figura 9 Diagrama de flujo del control domótico	34
Figura 10 Fotografía instalación de sensores y relés sección 3- sala, sección 4 - biblioteca, sección 5-comedor	38
Figura 11 Fotografía sensor y relés seccion1-dormitorio1	39
Figura 12 Fotografía sensor y relé sección 6 cocina	39
Figura 13 Fotografía sensor y relé sección 2 dormitorio2.....	40
Figura 14 Fotografía conexión sensores, relés e interruptores a tarjeta Raspberry	40
Figura 15 Programa experimento 1	41
Figura 16 Programa experimento 2	42
Figura 17 Historia de consumo energético de la vivienda.....	43
Figura 18 Programa experimento 3	44
Figura 19 Documentos de Google Sheet para el almacenamiento de datos	45
Figura 20 Programación de red neuronal opción 1 entrenamiento.....	47
Figura 21 Coste de la función en el primer entrenamiento.....	47
Figura 22 Programación de red neuronal opción 2 entrenamiento.....	48
Figura 23 Coste de la función en el segundo entrenamiento.....	49

Figura 24 Programación de red neuronal opción 3 entrenamiento.....	49
Figura 25 Coste de la función en el segundo entrenamiento	50
Figura 26 Datos obtenidos con red neuronal ejemplo 1	51
Figura 27 Datos obtenidos con red neuronal ejemplo2	52
Figura 28 Datos obtenidos con red neuronal ejemplo 3	53
Figura 29 Ejemplo toma de datos Sección 1 Dormitorio1	75
Figura 30 Ejemplo toma de datos Sección 2 Dormitorio2	75
Figura 31 Ejemplo toma de datos Sección 3 Sala	76
Figura 32 Ejemplo toma de datos Sección 4 Biblioteca.....	76
Figura 33 Ejemplo toma de datos Sección 5 Comedor	77
Figura 34 Ejemplo toma de datos Sección 6 Cocina	77
Figura 35 Planilla de luz de la vivienda	78

Lista de Tablas

Tabla 1 Dispositivos y distribución	28
Tabla 2 Equipos por utilizar	31
Tabla 3 Disposición de cables	33
Tabla 4 Datos de entrada y salida esperada.....	37
Tabla 5 Datos obtenidos	45
Tabla 6 Datos tomados de la sección 5 Comedor.....	54

Lista de Ecuaciones

<i>Ecuacion</i> (1)	24
<i>Ecuacion</i> (2).....	24
<i>Ecuación</i> (3).....	25

Dedicatoria

A mis padres, Alfredo y María Victoria, que siempre me han apoyado incondicionalmente para que yo pueda superar cada reto que me he propuesto, a mis hermanos, Byron, Samantha, Juan Francisco y Cesar, ya que siempre han estado ahí dándome aliento, a mis sobrinos Doménica, Amelia, Romina y Samuel ya que con sus ocurrencias y compañía han sabido darme fuerzas.

Allen Wilchez

Agradecimiento

A Dios por permitirme esta oportunidad de crecer profesionalmente y tener el apoyo de mis familiares y amigos, a mis padres por siempre apoyarme en cada proyecto que deseo comenzar, a mis hermanos que con su apoyo siempre son el impulso necesario.

A la Universidad Internacional SEK junto con sus profesores por las enseñanzas impartidas tanto en lo académico como en lo humano.

A mis amigos Andrea, Karen y Jairo que dentro de la maestría siempre supieron dar ese empujón de ayuda incondicional.

A Mavi por su apoyo incondicional y alentarme a seguir adelante.

Glosario

IoT – Internet of Thing (Internet de las cosas)

RNA – Redes Neuronales Artificiales

IA – Inteligencia Artificial

Resumen

La implementación de una red neuronal artificial en el control domótico de iluminación en una vivienda plurifamiliar conocida como piso se lleva a cabo a partir de la instalación de sensores de presencia así como también relés que permitan el control a través de una Raspberry el encendido y apagado de las luces de manera manual y automática en la parte de domótica, una vez teniendo el control de la iluminación de una vivienda se realiza la definición de una red neuronal, en este caso se aplica una red neuronal con una función de activación sigmoide con un entrenamiento de backpropagation y aprendizaje supervisado, definiendo a la red multicapa con 3 neuronas en la capa de entrada, 15 neuronas en la capa oculta y 1 sola neurona en la capa de salida. Con dicha red neuronal la cual se encuentra implementada y la domótica instalada físicamente en la vivienda se tienen valores aproximados a los estimado siendo estos 0.98 y -0.00013 para los valores esperados de 1 y 0 respectivamente y no presentando fallas en el encendido y apagado.

Palabras clave: machine learning, Redes neuronales artificiales, domótica, inteligencia artificial, control, iluminación.

Abstract

The implementation of an artificial neural network in the home automation control of lighting in a multi-family dwelling known as an apartment is carried out from the installation of presence sensors as well as relays that allow control through a Raspberry on and off the lights manually and automatically in the home automation part, Once having the control of the lighting of a house, the definition of a neural network is performed, in this case a neural network is applied with a sigmoid activation function with a backpropagation training and supervised learning, defining the multilayer network with 3 neurons in the input layer, 15 neurons in the hidden layer and 1 single neuron in the output layer. With this neural network, which is implemented and the home automation system physically installed in the house, the estimated values are 0.98 and -0.00013 for the expected values of 1 and 0 respectively, and there are no failures in the switching on and off.

Key words: machine learning, artificial neural networks, home automation, artificial intelligence, control, lighting.

Introducción

Antecedentes

En el Ecuador la domótica se considera un término relativamente nuevo y no muy común su implementación en los hogares, dando un pensamiento a las personas que no podría estar al alcance de todas por desconocimiento según lo expresa Gómez (2021), las empresas en el Ecuador que están desarrollando sus actividades en esta área lo hacen implementando los sistemas creados en el exterior y Ecuador se vuelve solo consumidor y lo que se espera es que se puedan desarrollar equipos para domótica en el país indica Morejón (2016).

La eficiencia energética está siendo desplazada a un segundo plano en entornos como las viviendas, si bien es cierto que se busca la reducción del consumo energético mediante la aplicación de domótica en vivienda, como lo indica el Ministerio de electricidad y energía Renovable (2021) el consumo energético de las viviendas está distribuidos en el 46% en equipos incluidos electrodomésticos, el 49% en iluminación y el restante en equipos más pequeños.

Tomando en cuenta lo expuesto anteriormente surge la necesidad de proponer la implementación de un control inteligente en este caso mediante redes neuronales artificiales para el control domótico de la iluminación que representa casi la mitad del consumo energético de una vivienda promedio.

La domótica parte del punto de la automatización de una casa; es decir, dotar de la capacidad de realizar tareas simples como el controlar el encendido y apagado de luces, control de temperatura en ambientes, seguridad, bienestar, confort, así como también tener referencia del consumo energético de la casa y así poder administrar mejor los recursos de esta, permitiendo así ser capaz de entrelazar varias redes y dispositivos de manera local y remota (Carrillo & Vite, 2016).

Basados en la estructura de un sistema domótico estos cuentan con tres partes o elementos principales entre los cuales podemos mencionar a los sensores, quienes serán los que ayuden en la adquisición de datos del entorno, el controlador, el cual receptorá estos datos los procesara y enviar una señal ya controlada y la parte de los actuadores los cuales ejecutarán las órdenes emitidas por el controlador (Camargo, Coronel, & Calderón, 2015).

En la implementación de un sistema domótico en la actualidad se puede relacionar con disciplinas y tecnologías como inteligencia artificial (IA) y entre esta rama la aplicación de redes neuronales artificiales (RNA), también se presenta el internet de la cosas (IoT), sistemas de aprendizaje automático (machine learning), teniendo así también el aprendizaje profundo o conocido también como deep learning, complementando así también con computación en la nube y el uso de plataformas para facilitar las telecomunicaciones y la conexión inalámbrica, presentando así los beneficios en la implementación de domótica como el mejorar la calidad de vida del usuario ya que esto permite comunicación integral de los espacios y dispositivos, permitiendo un control automático y control a distancia de las condiciones de una vivienda, y una de la principales razones que sería la gestión eficientes de los recursos instalados y ahorro energético de la vivienda (Rebato, 2020).

Dentro de la inteligencia artificial se encuentra una rama conocida como redes neuronales las cuales se basan en la estructura del cerebro, donde sus resultados son básicamente no lineales las cuales poseen una entrada de información mayormente conocida como entrada neta, la procesan y generan información, simulan el comportamiento de neuronas biológicas; estas pueden organizarse de forma que se pueden diferenciar capas en un RNA, teniendo así una capa de entrada, la cual es la que recibe la información, un capa oculta que se encargan de recibir dicha información,

procesarla y entregarla a las neuronas de salida, y la capa de salida la cual recibe la información procesada y la remite a la red (Mahecha, 2018).

En trabajos relacionados con domótica y redes neuronales podemos destacar a Mahecha (2018), su topología se basa en un controlador, sensores y actuadores los cuales se enlazan a su base de datos mediante una aplicación. Su red neuronal está sustentada en una red multicapa e implementado el algoritmo backpropagation teniendo así un 5% como error en comparación de la salida deseada. Dividiendo así la red multicapas, en la primera capa encontramos cada una de las entradas como día, hora, sensor de presencia, y nivel de luz, teniendo en la capa intermedia un número aleatorio de neuronas, y en la capa de salida se puede observar los valores para cada uno de los actuadores en este caso la intensidad de cada led. Teniendo en cuenta que el algoritmo se basa en un aprendizaje supervisado ya que se tenía información que la salida deseada a través de encuestas realizadas a usuarios. Como complemento al sistema involucra asistente de voz que le permiten encender o apagar las luces, así como también el cambio de tono y la intensidad, dentro de su apartado de trabajos futuros sugiere se complemente con control de seguridad y de temperatura, en cuanto a eficiencia energética ampliar la gama de dispositivos que se puedan acoplar al sistema.

En otro punto de vista tenemos el desarrollo y lo que menciona Camargo, Coronel, & Calderón (2015) en su publicación basándose en el control de luminosidad en la vivienda ocupando una red neuronal con topología feedforward-backpropagation la cual permite identificar el comando de voz utilizado por el usuario respaldándose en una base de datos de posibles palabras para las diferentes acciones que se pueda realizar y desarrollando una aplicación Android en el cual muestre el estado de los puntos de control y permitiendo también el control a distancia de las luces de la vivienda.

En el trabajo realizado por Espino, Pérez, Cossio (2020) buscan integrar en un sistema mejoras para uso y control de los recursos de una vivienda usando RNA teniendo acceso a una base de datos online y poder tener acceso mediante internet desde cualquier lugar basándose en un algoritmo supervisado. En revisión del trabajo desarrollado por Martínez & Gudiño (2019) hacen referencia al trabajo de un control domótico para un edificio de iluminación, calefacción donde se involucra la humedad y temperatura y lo relaciona con un sistema para el reconocimiento facial aplicando redes neuronales la cual aplica un algoritmo supervisado ya que se posee una base de datos para el reconocimiento facial y basándose en el algoritmo backpropagation.

Por parte de Huamán (2017) en su trabajo hace referencia a la implementación de controles adaptativos usando redes neuronales y presentando también la alternativa del método de árbol de decisión para el control inteligente de la parte de iluminación en edificios ya sean estos considerados casas u oficinas teniendo en cuenta que las condiciones de los ambientes se ven afectados por factores externos como el color de las paredes y techos.

El internet de las cosas (IoT) es un desarrollo de tecnología que permite que los dispositivos estén entrelazados entre sí mediante conexión a internet, como lo indica Escobar & Villazón (2018) en su trabajo en el cual ocupan esta tecnología para el desarrollo de una plataforma de monitorización del consumo energético y permite el control y automatización de la iluminación, calefacción solar y también la extracción de aire.

Y tomando en consideración el trabajo realizado por Henríquez & Palma (2011) con el tema Control automático de condiciones ambientales en domótica usando redes neuronales artificiales podemos destacar el uso de redes neuronales artificiales para el

control de la iluminación en una oficina así como también puede monitorear la calefacción y el uso de las puertas y ventanas la cual identifica los patrones que tiene el usuario durante los días en la oficina, así como también el uso de una aplicación para el monitoreo de estados y control que permite al usuario tener conocimiento de lo que está aconteciendo en su lugar de trabajo. Dicho trabajo habla de el algoritmo de retro propagación en una RNA multicapa, el entrenamiento se basa en un promedio de 200 iteraciones de cada red, tardando en el entrenamiento 1 hora aproximadamente, donde se puede entrenar con los datos de 60 días, y esta frecuencia se considera suficiente para tener el sistema actualizado.

El presente trabajo de implementación tiene como enfoque el área de ahorro y eficiencia energética, comodidad y confort con cierta incidencia en la parte de personalización del espacio. Teniendo así un control de la iluminación con la implementación de redes neuronales, adicional el desarrollo de una aplicación de visualización de estados de los equipos instalados de domótica.

La implementación de este proyecto no busca sólo el confort y comodidad de los habitantes de la vivienda si no que una vez relacionados todos los puntos a implementar y como consecuencia del funcionamiento del control domótico a través de redes neuronales artificiales es aprovechar de mejor manera los recursos para poder optar por ahorro y eficiencia energética, que en la actualidad se vive en una sociedad que busca suplir este punto crítico.

El objetivo principal del presente trabajo es Implementar una red neuronal artificial en el control domótico de iluminación en una vivienda plurifamiliar conocida como piso, mediante la captación y procesamiento de datos de presencia para obtener una respuesta y predicción del comportamiento del sistema de iluminación.

- Implementar un sistema domótico mediante la aplicación de sensores de presencia para el control de iluminación en una vivienda.
- Implementar redes neuronales en el control domótico de iluminación de una vivienda mediante una red con topología backpropagation y aprendizaje supervisado para poder obtener una respuesta y predicción.
- Implementar el almacenamiento de datos del estado de los dispositivos, mediante el envío de información a una plataforma externa distinta a la unidad de control, para que se pueda ocupar para el procesamiento de información posterior por el usuario.

La implementación de una red neuronal a un sistema domótico del control de iluminación de una vivienda instalando sensores de presencia y relés para el control de la iluminación permitirá tener un mejor control y optimización del encendido y apagado de las luces en la vivienda.

Estado del arte

Domótica

La domótica se considera una herramienta de la era moderna la cual hace uso de las tecnologías para poder aplicar control y automatización a viviendas, centrada en facilitar la vida de las personas brindando confort y que estos puedan tener un mejor diseño de las viviendas, hogares más acogedores, y personalizados. El avance tecnológico ha impulsado la domótica lo cual ha favorecido ya que los costos de instalación e implementación han disminuido, y ampliando los productos y servicios a los cuales se podría acceder. La domótica siempre se basa en sensores y actuadores que le permiten recibir los datos de las variables físicas, las procesa y devuelve una acción como respuesta mediante los actuadores. La implementación de domótica en las viviendas va

de la mano de las normativas de vigentes de la construcción. Lo importante de la domótica aparte del control y automatización de los dispositivos de las viviendas es que el usuario puede interactuar entre el usuario y el sistema (Asociacion Española de Domotica e Inmotica, 2021).

Aporte de la domótica. La implementación de Domótica dentro de una vivienda permite crear un escenario de confort para las personas que lo habiten, también permitiendo que tengan acceso a toda la información de manera presencial o remota, la domótica da impulso a los siguientes puntos:

- **Ahorro energético:** debido a la información obtenida por los sensores puede dar gestión en cuanto a iluminación, climatización entre otros parámetros que se necesite controlar, al poder tener el control y automatización puede mostrar al usuario la información necesaria para las respectivas correcciones para disminuir el consumo energético y por ende aumentar la eficiencia.
- **Accesibilidad:** Facilita el manejo de información y por ende el control de los elementos que podamos encontrar en el hogar de manera presencial o de forma remota.
- **Seguridad:** La domótica también entra en el área de seguridad en cuanto a vigilancia de personas, animales y bienes. En cuanto a cámaras, alarmas, detección de incendios, apertura de puertas.
- **Comunicaciones:** Una vez implementado el control y automatización se puede acceder a una supervisión remota de todos los parámetros considerados, también permite la transmisión de datos dándole acceso a internet.

Transmisión de datos

En el ambiente domótico como se mencionó anteriormente el uso de datos es recurrente tanto la recepción y la emisión por lo cual podemos encontrar:

- **Cable propio:** Es un sistema de los más comunes utilizados para la instalación los principales son el par trenzado hasta de 4 pares, cable coaxial, par apantallado y en muchas ocasiones la fibra óptica.
- **Cable compartido:** esta predominado por el uso de redes ya existentes y que permiten la trasmisión de datos ya sean estas el cableado de la red eléctrica, red telefónica o muchas veces la red de datos.
- **Inalámbrico:** Es una de las características en los últimos tiempos de la domótica, utilizar, radio frecuencia, infrarrojo, Wifi o Bluetooth (Mahecha, 2018)

Inteligencia Artificial

La inteligencia artificial busca crear sistemas y maquinas que se puedan comportar de manera inteligente, la inteligencia artificial en sus principios se basó en dar solución a teoremas, formulas entre otros campos, con el pasar del tiempo se ha ido buscando aplicaciones para el mundo real, lo que se busca en realmente que se puedan cumplir con procesos que hace el cerebro humano como son (Garcia, 2009):

- Aprendizaje
- Razonamiento
- Autocorrección

Machine Learning

Siendo una rama de la IA, esta aplica algoritmos los cuales permiten que un ordenador el cual disponen de datos masivos los analice y pueda regresar una predicción

para un cierto evento. Teniendo varias categorías donde podemos encontrar (Iberdrola, 2021):

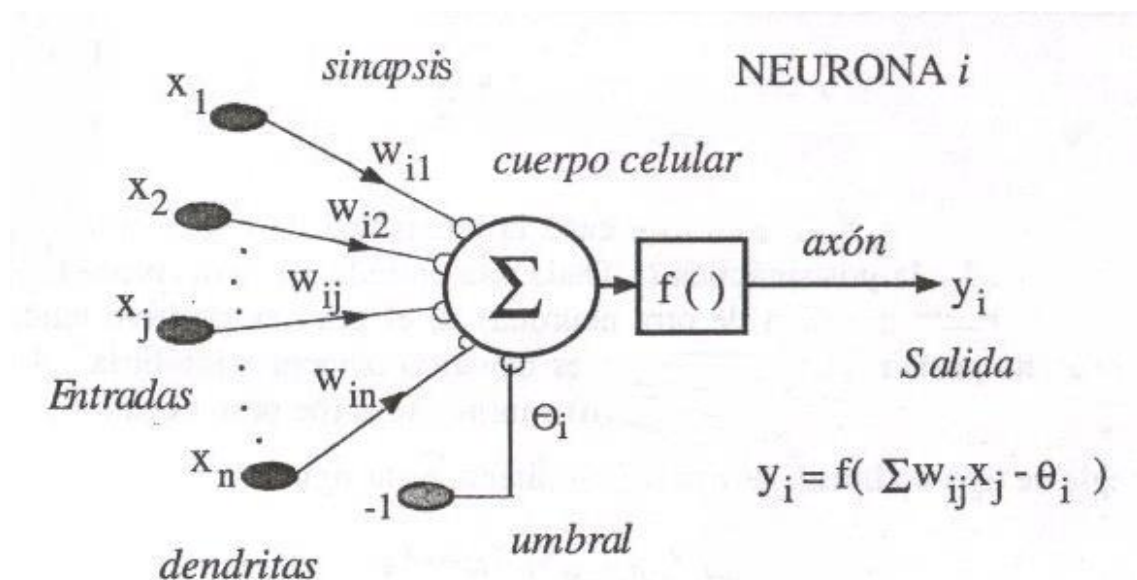
- Aprendizaje supervisado: se basa en una base de datos previa la cual le permite tomar decisiones o hacer predicciones
- Aprendizaje no supervisado: se tiene acceso a una base de datos en la cual se debe buscar algún patrón que permita la toma de decisiones o hacer predicciones
- Aprendizaje por refuerzo: Aprende a partir de su propia experiencia para tomar la mejor decisión.

Redes neuronales artificiales

Siendo una rama de las IA, estas son estructuras de hardware y software que emulan el comportamiento del cerebro humano, el modelo de una red neuronal (Figura 1).

Figura 1

Modelo de la neurona artificial



Nota. Adaptado de Conceptos básicos sobre RNA, por Ibliblio

Funciones de una neurona artificial

Con relación a una neurona biológica para poder comunicarse con una neurona vecina atraviesan por un proceso electroquímico, para poder imitar este comportamiento se disponen de varias funciones como son:

- Función de propagación (Net_j): Matemáticamente hablando es la suma ponderada de las entradas que se encuentra recibiendo (Ecuación 1) lo que quiere decir, es que es la suma de las entradas multiplicadas por una ponderación correspondientes.


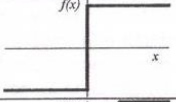
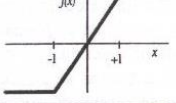
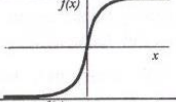
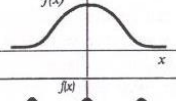
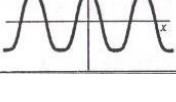
$$Net_j = (w_{ij} y_i(t)) \quad (1)$$

- Función de activación (F): Combina el resultado obtenido en la función de propagación con el estado actual de dicha neurona (Ecuación 2) para conseguir el nivel de activación (Figura 2).

$$A_j(t + 1) = F(a_j(t), Net_j) \quad (2)$$

Figura 2

Funciones de una neurona artificial

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(ax + \varphi)$	$[-1, +1]$	

Nota. Adaptado de *Conceptos básicos sobre RNA*, por Ibiblio

- Función de salida: esta función se encarga de convertir el estado de activación de la neurona en una salida de esta.

$$F_j(a_j(t)) = a_j(t) \quad (3)$$

Capas de la estructura de una neurona artificial

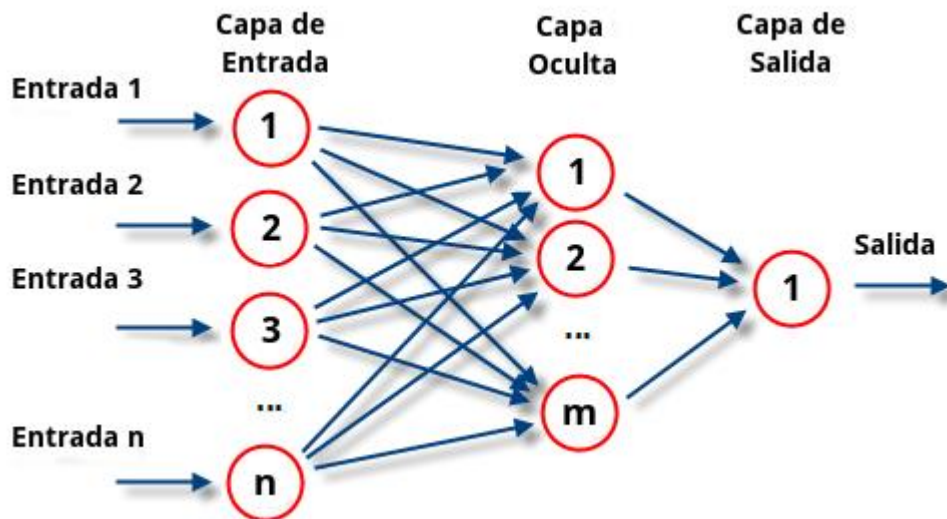
Las neuronas según su función se organizan en capas (Figura 3) ya que estas reciben información del exterior, la cual procesan y transmiten a la siguiente, es así como tenemos tres capas (Utrilla, 2014):

- Capa de entrada: Estas son las neuronas que se encuentran en el inicio y reciben la información para ser procesada.
- Capa oculta: el número de neuronas en esta capa depende de la utilización de esta, esta capa recibe la información la procesa y la pasa a la neurona de salida.

- Capa de salida: las neuronas de esta capa reciben la información de las neuronas de la capa oculta y las envía a la red.

Figura 3

Estructura de capas

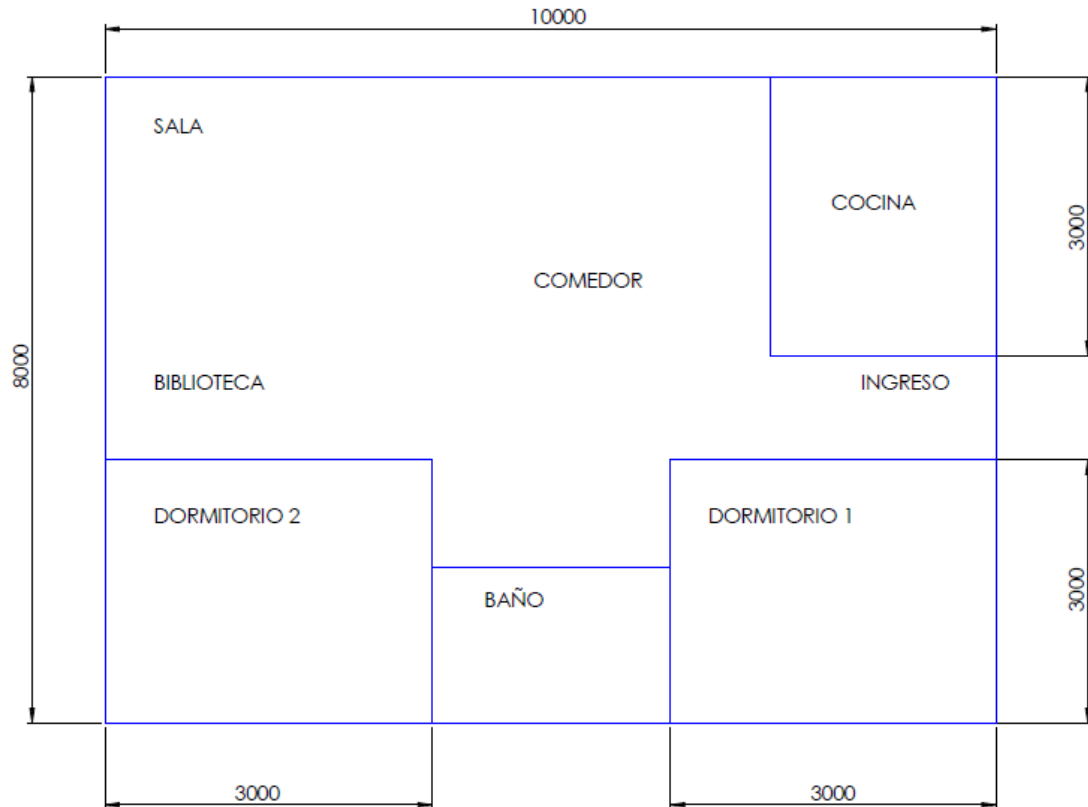


Nota. Adaptado de Conceptos básicos sobre RNA, por Ibiblio

Método

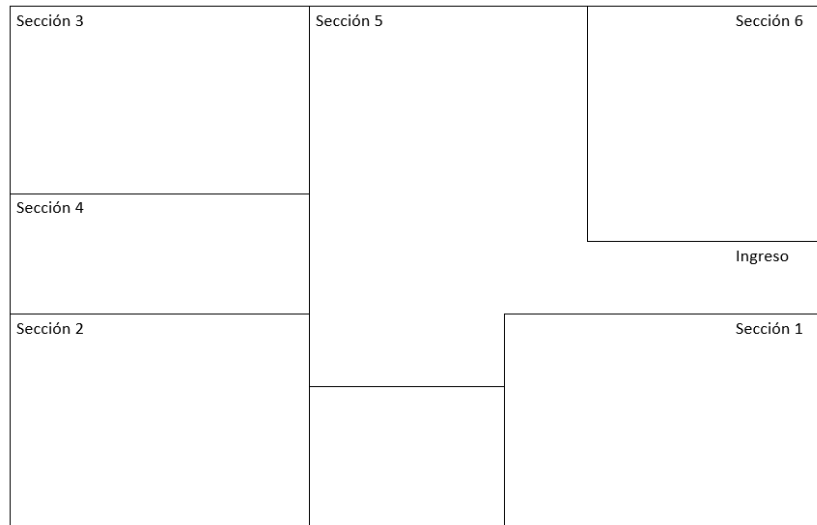
La metodología por ocupar en el presente trabajo se basa en el método experimental, ya que se busca la implementación de redes neuronales para el control domótico de iluminación en una vivienda, la cual cuenta aproximadamente con 80 m² dividida en varias áreas, dos dormitorios, sala, biblioteca, cocina, comedor (Figura 4).

Figura 4

Plano de vivienda general

En la cual se implementará sensores de presencia en cada sección, cada sección divide un área de la casa (Figura 5), los sensores deben ser capaces de abarcar la mayor área posible y no crear puntos ciegos dentro de la instalación.

Figura 5

División de la vivienda por secciones

Estimando así 6 sensores de presencia, distribuidos a la par con los puntos de emisión de luz, es decir, uno por cada punto de iluminación. Respecto a los actuadores se busca tener control a cada uno de los puntos de iluminación de la vivienda, es decir cuenta con 6 puntos de iluminación donde se busca poder tener un control ON-OFF de los focos o lámparas LED. Dicha distribución de los sensores (Tabla 1).

Tabla 1

Dispositivos y distribución

Ubicación	Sección	Sensor de presencia	Dispositivo	Sensor Presencia	Conexión
Dormitorio 1	1	x	Foco LED	PIR	Cableado
Dormitorio 2	2	x	Foco LED	PIR	Cableado
Sala	3	x	Lampara LED	PIR	Cableado
Biblioteca	4	x	Lampara LED	PIR	Cableado
Comedor	5	x	Lampara LED	PIR	Cableado
Cocina	6	x	Foco LED	PIR	Cableado

Se utiliza el mismo cableado estructural de la vivienda para de la alimentación de las iluminarias (Figura 6) y se adicional un cableado para obtener los datos de interruptores, así como también para sensores y relés (Figura 7).

Figura 6

Cableado de iluminarias

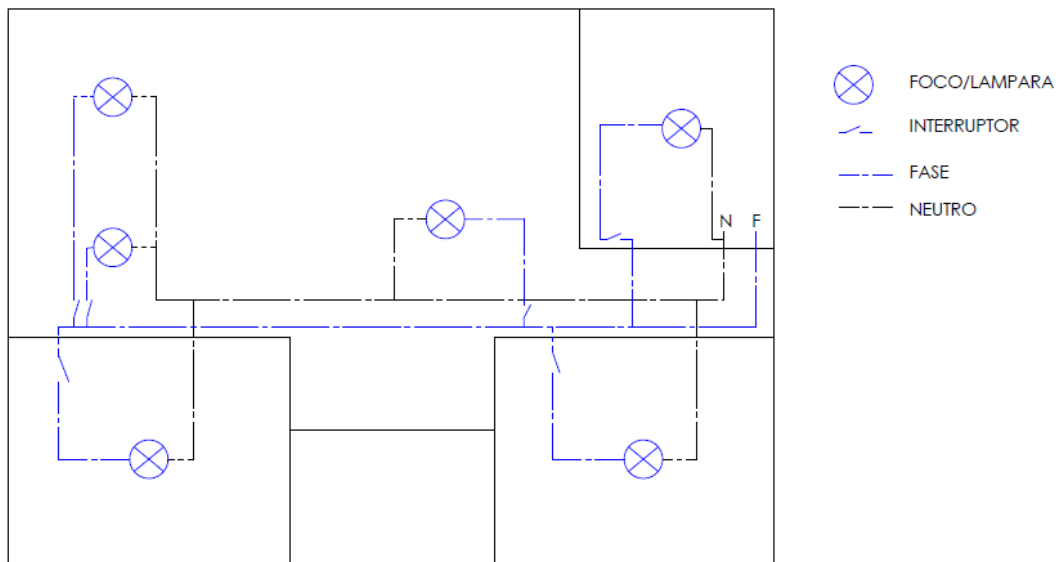
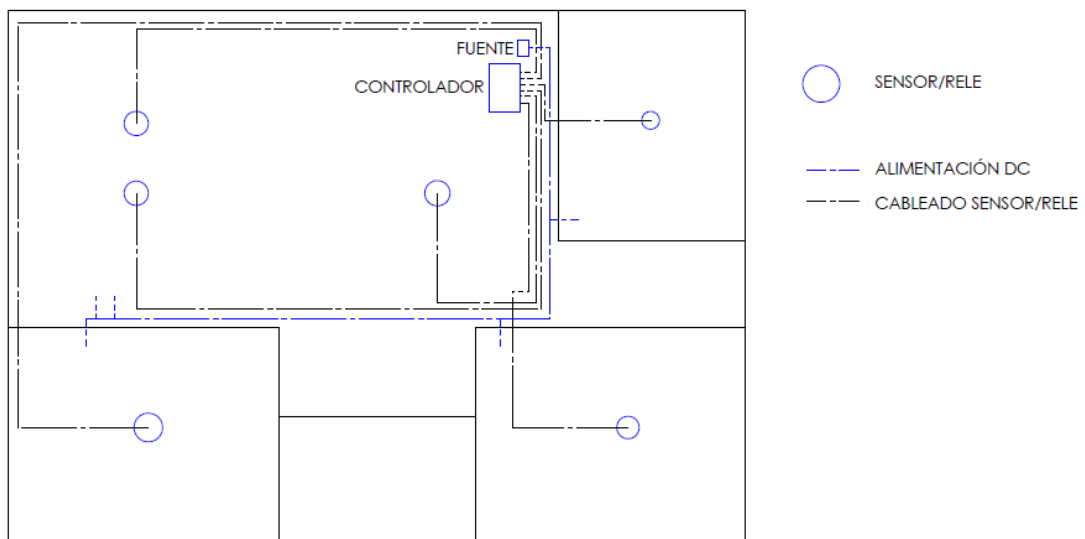


Figura 7

Distribución cableado sensores, relés e interruptores

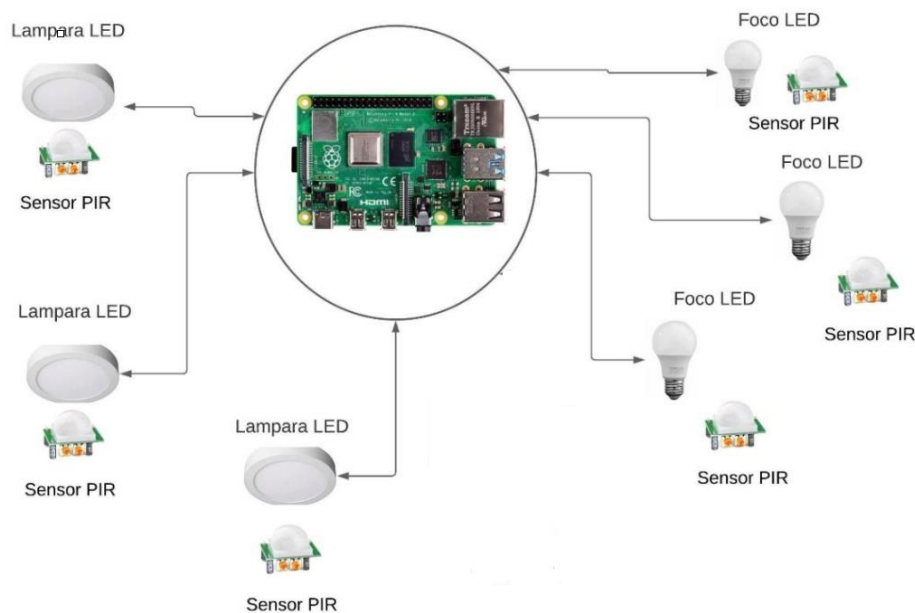


Topología

La topología de red se basará en una red cableada la cual interconectará los dispositivos, cada una de las secciones como se mencionó anteriormente tendrá un actuador y un sensor de presencia, todos los dispositivos estarán emitiendo y recibiendo datos de una tarjeta de control, en este caso una Raspberry Pi 4. La cual recibirá los datos emitidos por los sensores, los procesará y emitirá la orden a los relés para proceder con el encendido o apagados de las luces, el sistema aparte del encendido y apagado de las luces mediante el sensor podrá optar por el encendido y apagado mediante un interruptor (Figura 8).

Figura 8

Topología de RED



La tarjeta Raspberry Pi4 también almacenará los datos de presencia en cada una de las secciones de la vivienda y está también tendrá la capacidad de desarrollar el entrenamiento e implementación la red neuronal.

Equipo por utilizar

Dentro del desarrollo del proyecto se implementará el sistema con los siguientes equipos presentando las siguientes características (Tabla 2):

Tabla 2

Equipos por utilizar

Componente	Cantidad	Características
Raspberry Pi 4 Computer	1	Modelo B 4Gb Ram Porcesador 64-bit quad-core Cortex-A72 2micro puertos HDMI 2 puertos USB 3.0 2 puertos USB 2.0 Puerto Gigabit Ethernet Wireless 802.11b/g/n/ac Bluetooth 5.0 Fuente de poder 5V/ 3A
Sensor PIR	6	HC-SR501 Dimensiones aproximadas: 3.2cm x 2.4cm x 1.8 cm Circuitos de control incluidos en el módulo Tiempo de activación y sensibilidad ajustable Voltaje de operación: 4.5 – 20 V Corriente de operación: <60uA Salida: Pulso lógico 3.3V Tiempo de retardo: 5 – 200 Segundos (ajustable) Distancia de detección: 3 – 7 Metros (ajustable)
Modulo Relé	6	Voltaje de Operación: 5V DC. Señal de Control: TTL (3.3V o 5V). Modelo Relay: SRD-05VDC-SL-C. Capacidad máx: 10A/250VAC, 10A/30VDC.

		Corriente máx: 10A (NO), 5A (NC). Tiempo de acción: 10 ms / 5 ms. Entradas Optoacopladas. Indicadores LED de activación.
Fuente de alimentación	1	Marca: Anera Switching Power Supply Modelo: AE-PSU3080
Cable UTP		Categoría 5E 24 AWG
Cable flexible		Paralelo XT 2x20 AWG
Foco LED	3	100-240V 50/60 Hz 6W
Lámparas LED	3	11-240 V 50/60 Hz 24 W
Interruptores	4	Interruptor de pared Manejo de señal DC 5V
Interruptores conmutadores	4	Interruptor de pared Manejo de señal DC 5V

Conexiones

Teniendo así la conexión para cada dispositivo de cada sección, la conexión entre los sensores, actuadores y controlador se lo realiza mediante cable, utilizando cable UTP categoría 5E par trenzado de 8 hilos, los cuales se utilizan 6, tres para el sensor de presencia y 3 para el relé. Teniendo así en los dos casos 1 hilo para alineación 5 V, otro para GND y el tercero para señal. Así también se tiene la conexión para la señal del interruptor se trabaja con una conexión pull down y se trabaja con 5 V y GND cuya señal entra en una dos los pines de la Raspberry. Las alimentaciones para los sensores de presencia, los relés y la utilizada en los pulsadores provienen de una fuente de alimentación externa, cuyo GND es puenteado con el GND de la Raspberry (Tabla 3).

Tabla 3

Disposición de cables

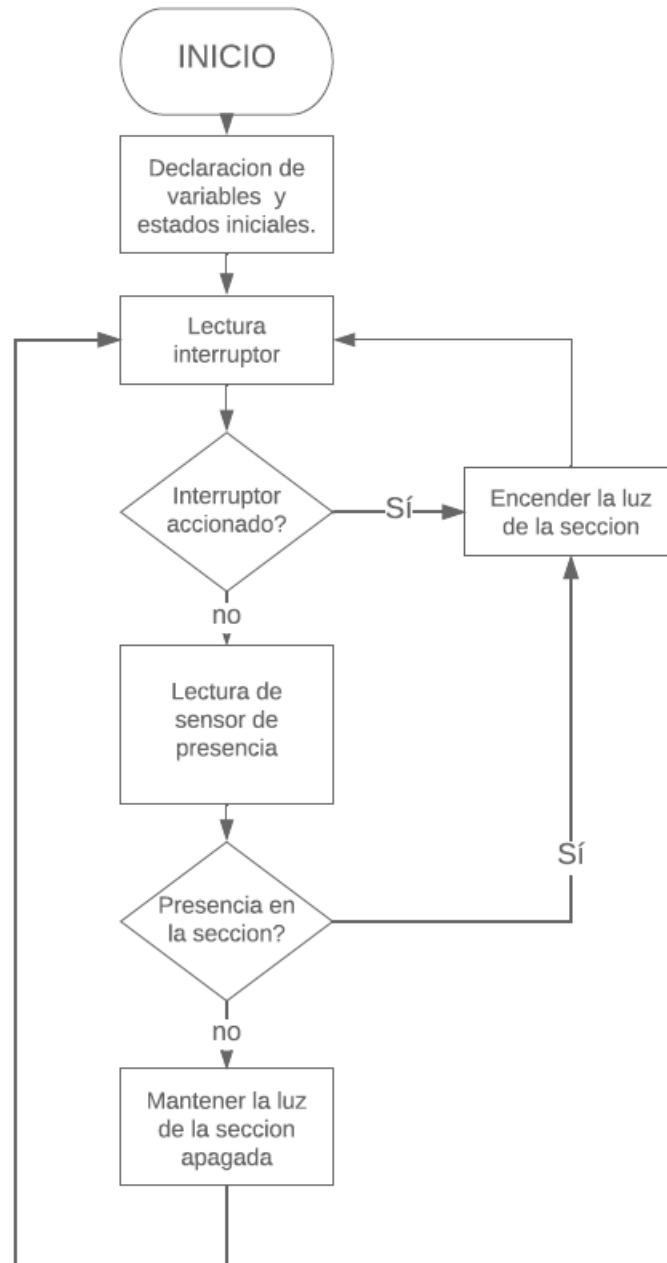
Dispositivo	Cable	Descripción
PIR	Verde	5 v
	Blanco/Café	GND
	Café	Señal
Relé	Blanco/Naranja	5 v
	Blanco/Azul	GND
	Blanco/Verde	Señal

En los equipos a utilizar se está basando en equipos que permitan la conectividad entre las diferentes plataformas o sistemas operativos y así como también buscar no dependencia de marcas, teniendo en cuenta la facilidad de obtención y la comercialización de estos. Teniendo así principalmente en hardware: controlador, sensores (presencia) y actuadores y a nivel de software: programador de la tarjeta electrónica y de la base de datos.

Flujo de envío y recepción de datos

En el flujo de envío de datos, la topología de red se basa en el envío de datos a través de cable. Teniendo así el dos posibles entradas para el encendido de las luces, por interruptor o por sensor de presencia, ambas señales manejan pulsos de 5 V, la tarjeta Raspberry verifica la señal y envía la orden al relé para cambiar de estado y permita el encendido y apagado de las luces (Figura 9).

Figura 9

Diagrama de flujo del control domótico

Teniendo esto se plantea 4 pasos o experimentos secuenciales los cuales ayudaran al desarrollo del proyecto en óptimas condiciones.

Experimentos

Experimento 1

Implementación de los circuitos de control, así como la implementación de circuitos de potencia.

- Variable principal: Lectura de datos de sensores

En este proceso se tendrá que adecuar las instalaciones de la casa para poder instalar lo que son los sensores y el respectivo circuito de potencia el cual me permitirá el control de las luces de cada una de las secciones especificadas de la vivienda.

Una vez implementados se procederán a realizar pruebas de conectividad, es decir estar recibiendo y/o enviado datos entre el sensor, la tarjeta y el relé. Una vez implementado esto, se empezará el proceso de calibración de sensores de cada uno de los puntos de cada sección de la vivienda.

Experimento 2

Crear una red de prueba para el control de los actuadores en base a los datos enviados por el sensor de presencia.

- Variable Principal: accionamiento de actuador

En este proceso se verificará en la instalación del acoplamiento de los circuitos de control con los circuitos de potencia para el accionamiento de los actuadores. Se programa la tarjeta para la recepción de datos de los sensores y envíe la señal al relé para el encendido y apagado de las luces.

Experimento 3

Almacenar los datos de fecha y hora y de los estados de los componentes.

- Variables principales:

- Datos sensor presencia
- Datos para actuador

Una vez garantizada la comunicación y la calibración de los sensores de presencia se procederá con el enlace a la plataforma exterior para el almacenamiento de datos de los sensores, para que así el usuario tenga una referencia del comportamiento del sistema.

Experimento 4

Se define la RNA de manera que se tiene una red neuronal artificial multicapa con un patrón de entrenamiento de backpropagation y una función de activación sigmoide.

- Variable Principal:
 - Accionamiento de actuador
 - Datos sensor de presencia
 - Datos de interruptor
 - Datos de hora

En este proceso se validará la estructura de la RNA y la interacción con los datos, y ver los resultados sean los esperados, en este paso es en el cual se determina la tasa de aprendizaje o learning rate que es la que afecta la velocidad y calidad de la red así como también la cantidad de iteraciones o “epochs” para obtener las predicciones más cercanas a 1 de ser el caso de encendido o de 0 en el caso de ser que se necesita apagada la lámpara o foco del departamento.

Una vez teniendo en cuenta que el control domótico y el almacenamiento de datos en la plataforma externa están definidos, hay que definir la red neuronal que sea aplicable para este caso. Como se definió en un principio se utilizará una red neuronal backpropagation en una red multicapa, en la cual se definen que vamos a tener 3 capas,

una de entrada con tres neuronas las cuales responden a las tres variables consideradas para el control que son, la lectura del sensor de presencia, la lectura del interruptor y la hora del día, una capa oculta y una capa de salida con una neurona que es la salida correspondiente a la lámpara o foco led que cambiara su estado de encendido o apagado según lo determine la red neuronal.

La función definida para la implementación en este caso es la función sigmoide ya que sus propiedades matemáticas permiten el realizar de mejor manera el método de backpropagation ya que trabaja con sus derivadas, este método trabaja de derecha a izquierda para así mejorar la precisión y las predicciones. En este caso conocemos las entradas y las salidas esperadas teniendo así la tabla 4.

Tabla 4

Datos de entrada y salida esperada

Hora	Sensor presencia	Interruptor	Salida/ Lámpara
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Donde tenemos, 3 entradas, la primera donde se reflejan la hora, la cual es 0 cuando está fuera del horario establecido de encendido de luces (desde las 8 am -18 pm) y es 1 cuando está dentro del horario establecido (18 pm - 8 am), la segunda donde se refleja el estado del sensor de presencia, 0 cuando no hay detección y 1 cuando detectó presencia, y la tercera entrada corresponde al interruptor, 0 cuando no ha sido accionado y 1 cuando el usuario lo accionó, dichas combinaciones determinan el estado de la lámpara o foco del departamento representando así el encendido con un 1 y el estado apagado con el 0.

Una vez teniendo así implementado físicamente los sensores, relés y la red neuronal funcionando en la vivienda (Figura10, Figura 11, Figura 12, Figura 13, Figura 14)

Figura 10

Fotografía instalación de sensores y relés sección 3- sala, sección 4 -biblioteca, sección 5-comedor



Figura 11

Fotografía sensor y relés seccion1-dormitorio1



Figura 12

Fotografía sensor y relé sección 6 cocina

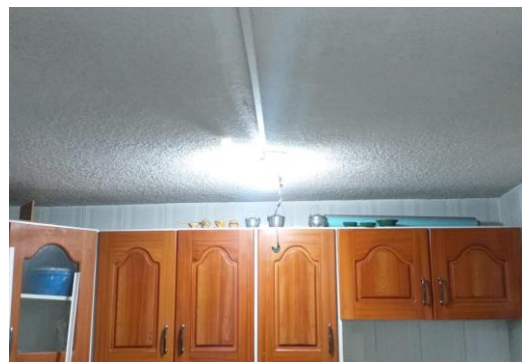


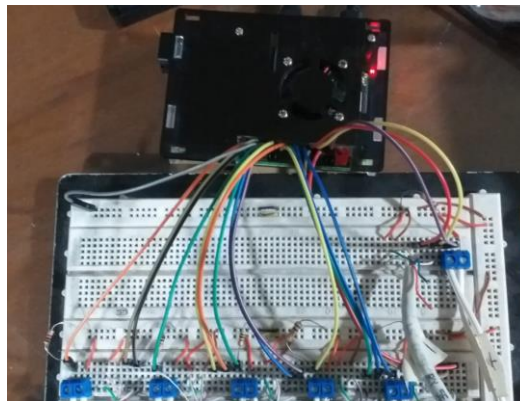
Figura 13

Fotografía sensor y relé sección 2 dormitorio2



Figura 14

Fotografía conexión sensores, relés e interruptores a tarjeta Raspberry



Resultados

Basándose en las experimentaciones planteadas se obtienen los siguientes resultados.

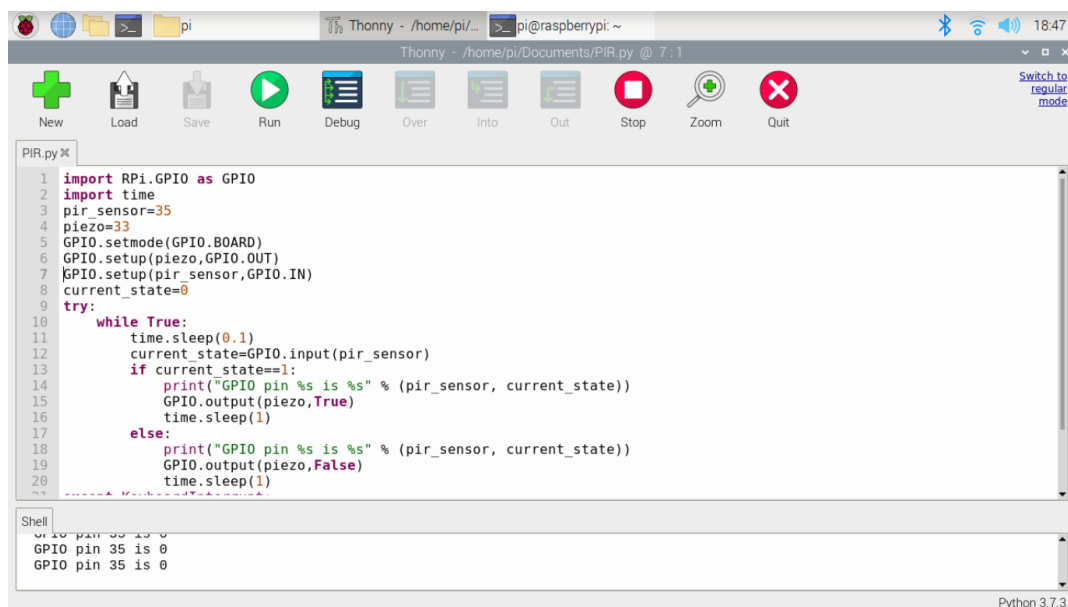
Experimento 1

Implementación de los circuitos de control, así como la implementación de circuitos de potencia.

Se implemento en la vivienda con las respectivas adecuaciones el cableado expuesto con las respectivas canaletas desde los puntos de la toma de datos hasta la tarjeta de adquisición de estos. Se implemento un código (Figura 15) para que la tarjeta pueda leer los datos emitidos por el sensor de presencia y que esta pueda tener una respuesta de confirmación.

Figura 15

Programa experimento 1



```
1 import RPi.GPIO as GPIO
2 import time
3 pir_sensor=35
4 piezo=33
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(piezo,GPIO.OUT)
7 GPIO.setup(pir_sensor,GPIO.IN)
8 current_state=0
9 try:
10     while True:
11         time.sleep(0.1)
12         current_state=GPIO.input(pir_sensor)
13         if current_state=1:
14             print("GPIO pin %s is %s" % (pir_sensor, current_state))
15             GPIO.output(piezo,True)
16             time.sleep(1)
17         else:
18             print("GPIO pin %s is %s" % (pir_sensor, current_state))
19             GPIO.output(piezo,False)
20             time.sleep(1)
```

Shell

```
GPIO pin 35 is 0
GPIO pin 35 is 0
GPIO pin 35 is 0
```

Python 3.7.3

Obteniendo así una respuesta acorde a lo esperado, sensor de presencia funciona correctamente y el programa permite leer y obtener una respuesta según sea la señal captada.

Experimento 2

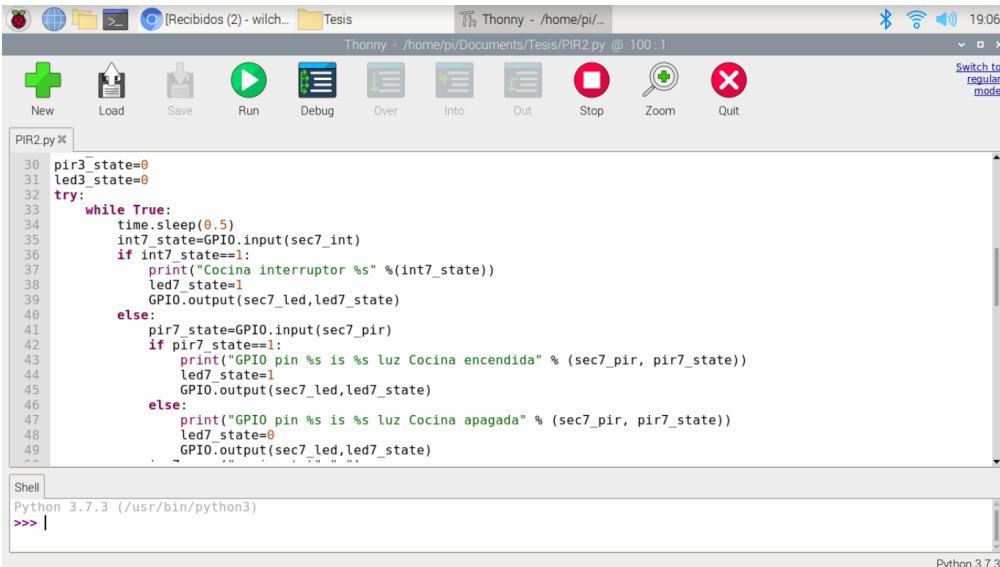
Crear una red de prueba para el control de los actuadores en base a los datos enviados por el sensor de presencia.

- Variable Principal: accionamiento de actuador

En base al programa para la captura de dato de presencia se instalan los dispositivos respectivamente en tres secciones para las respectivas pruebas las cuales cuentan con la señal de interruptor, señal de sensor de presencia y con el respectivo relé que receptorá la señal para el encendido y apagado de luces (Figura 16).

Figura 16

Programa experimento 2



```
PIR2.py x
30 pir3_state=0
31 led3_state=0
32 try:
33     while True:
34         time.sleep(0.5)
35         int7_state=GPIO.input(sec7_int)
36         if int7_state==1:
37             print("Cocina interruptor %s" %(int7_state))
38             led7_state=1
39             GPIO.output(sec7_led,led7_state)
40         else:
41             pir7_state=GPIO.input(sec7_pir)
42             if pir7_state==1:
43                 print("GPIO pin %s is %s luz Cocina encendida" % (sec7_pir, pir7_state))
44                 led7_state=1
45                 GPIO.output(sec7_led,led7_state)
46             else:
47                 print("GPIO pin %s is %s luz Cocina apagada" % (sec7_pir, pir7_state))
48                 led7_state=0
49                 GPIO.output(sec7_led,led7_state)
50
```

Shell
Python 3.7.3 (/usr/bin/python3)
>>> |

Python 3.7.3

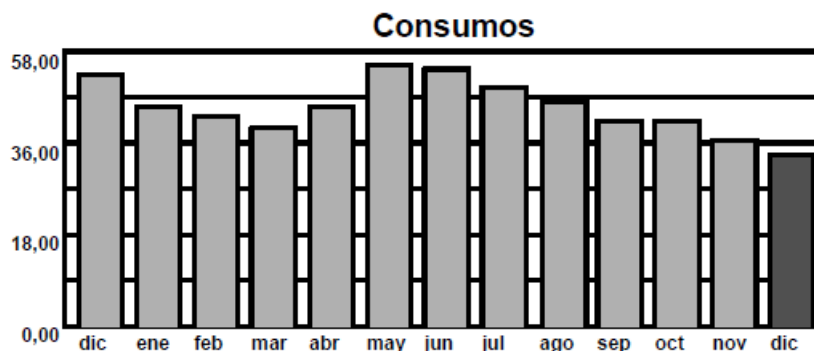
Una vez instalados los sensores y ver el comportamiento de estos ya en el sitio, es necesario una calibración en la sensibilidad y en el tiempo de respuesta de los sensores, es decir hay que variar los potenciómetros que se disponen en los módulos PIR según sea la necesidad, se calibran de tal manera que cubran el rango adecuado de su sección y no muestre una presencia cuando una persona este en otra sección y no haya ese cruce

de áreas de monitoreo. Teniendo así un tiempo de respuesta de alrededor de 1.5 segundos para la activación y para la desactivación cuando una persona abandona la sección de alrededor de 20 segundos.

Al finalizar el experimento 2 lo que es la implementación de la domótica en la vivienda se pudo observar una de las ventajas que viene a ser el ahorro energético, al estar solo actuando solo la iluminación es difícil separar el consumo solo de este parámetro, pero se tiene una referencia en la planilla de que se ve el decremento de consumo desde el mes de mayo 2021 que se realizó progresivamente la instalación de sensores, relés y el respectivo control (Figura 17), la planilla en anexo

Figura 17

Historia de consumo energético de la vivienda



Experimento 3

Almacenar los datos de fecha y hora y de los estados de los componentes.

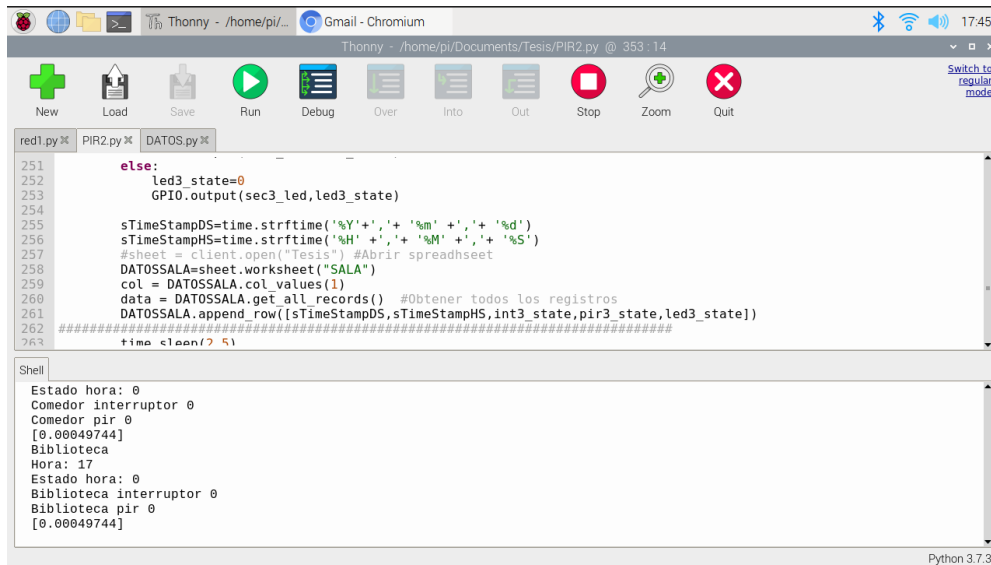
- Variables principales:
 - Datos sensor presencia y datos actuador

Una vez garantizada la comunicación y la calibración de los sensores de presencia se procede a modificar el programa para poder almacenar los datos captados en una

plataforma externa a la Raspberry, en este caso se procede al almacenamiento de los datos en Google Sheets (Figura 18).

Figura 18

Programa experimento 3



The screenshot shows the Thonny Python IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The main editor window displays a Python script with the following code:

```
251     else:
252         led3_state=0
253         GPIO.output(sec3_led,led3_state)
254
255         sTimeStampDS=time.strftime('%Y'+','+'%m'+','+'%d')
256         sTimeStampHS=time.strftime('%H'+','+'%M'+','+'%S')
257         #sheet = client.open("Tesis") #Abrir spreadhseet
258         DATOSSALA=sheet.worksheet("SALA")
259         col = DATOSSALA.col_values(1)
260         data = DATOSSALA.get_all_records() #0btener todos los registros
261         DATOSSALA.append_row([sTimeStampDS,sTimeStampHS,int3_state,pir3_state,led3_state])
262         #####
263         time.sleep(2.5)
```

Below the code editor is a Shell window showing the output of the script:

```
Estado hora: 0
Comedor interruptor 0
Comedor pir 0
[0.00049744]
Biblioteca
Hora: 17
Estado hora: 0
Biblioteca interruptor 0
Biblioteca pir 0
[0.00049744]
```

The bottom right corner of the IDE window indicates the Python version: Python 3.7.3.

Cada una de las secciones maneja una hoja independiente del documento de Google Sheet en el cual se podrá almacenar todos los registros (Figura 19).

Figura 19

Documentos de Google Sheet para el almacenamiento de datos

FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA
2022,07,23	23,11,09	0	0	0
2022,07,23	23,11,32	0	1	1
2022,07,23	23,11,53	0	1	1
2022,07,23	23,12,14	0	1	1
2022,07,23	23,12,36	0	0	0
2022,07,23	23,12,57	0	0	0
2022,07,23	23,13,19	0	0	0
2022,07,23	23,13,40	0	0	0
2022,07,23	23,14,01	0	0	0
2022,07,23	23,14,23	0	1	1
2022,07,23	23,14,44	0	1	1
2022,07,23	23,15,05	0	1	1
2022,07,23	23,15,27	0	0	0
2022,07,23	23,15,48	0	0	0
2022,07,23	23,16,10	0	0	0
2022,07,23	23,16,31	0	0	0
2022,07,23	23,16,52	0	0	0
2022,07,23	23,17,14	0	1	1
2022,07,23	23,17,36	0	1	1
2022,07,23	23,17,57	0	1	1
2022,07,23	23,18,18	0	0	0
2022,07,23	23,18,40	0	0	0
2022,07,23	23,19,01	0	0	0
2022,07,23	23,19,22	0	0	0
2022,07,23	23,19,44	0	0	0

En tabla 5 se puede observar los datos almacenados presentando así la fecha, hora de la toma de dato, el estado de interruptor, el estado del sensor de presencia, y el estado de la lámpara.

Tabla 5

Datos obtenidos

FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA
2022,07,23	23,11,09	0	0	0
2022,07,23	23,11,32	0	1	1
2022,07,23	23,11,53	0	1	1
2022,07,23	23,12,14	0	1	1
2022,07,23	23,12,36	0	0	0
2022,07,23	23,12,57	0	0	0
2022,07,23	23,13,19	0	0	0
2022,07,23	23,13,40	0	0	0
2022,07,23	23,14,01	0	0	0
2022,07,23	23,14,23	0	1	1
2022,07,23	23,14,44	0	1	1
2022,07,23	23,15,05	0	1	1
2022,07,23	23,15,27	0	0	0
2022,07,23	23,15,48	0	0	0
2022,07,23	23,16,10	0	0	0

Experimento 4

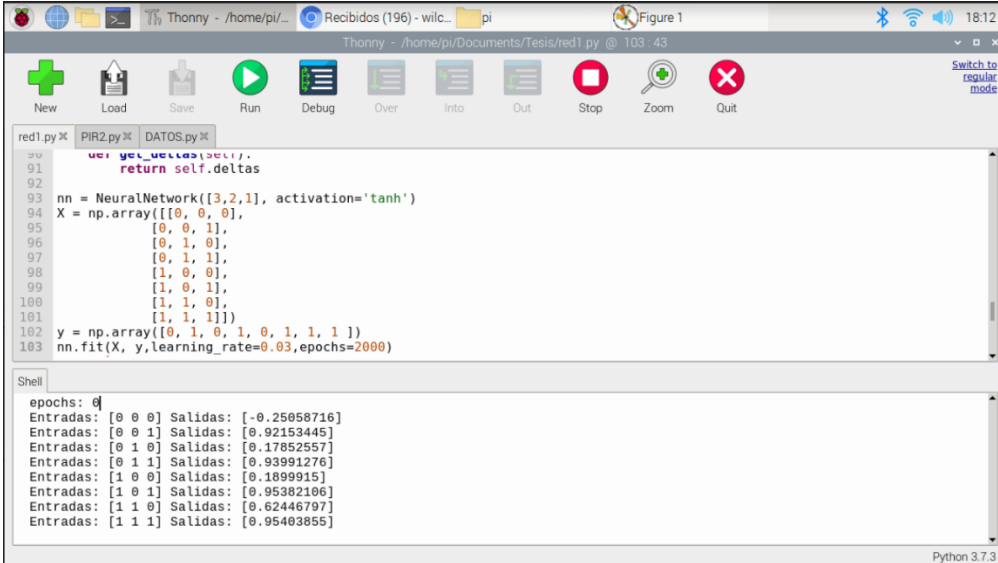
Se define la RNA de manera que se tiene una red neuronal artificial multicapa con un patrón de entrenamiento de backpropagation y una función de activación sigmoide.

- Variable Principal:
 - Accionamiento de actuador
 - Datos sensor de presencia
 - Datos de interruptor
 - Datos de hora

Se realizan varias iteraciones modificando los parámetros de número de neuronas de la capa oculta, valor de la tasa de aprendizaje y el número de iteraciones (epochs). En la primera opción entrenamiento se puede observar una capa oculta con 2 neuronas, una tasa de aprendizaje de 0.03 y epochs de 2000, se observa una respuesta bastante acorde a lo deseado y los valores que deben ser 1 se acercan por arriba del 0.9 excepto la séptima opción de entrada que refleja un valor de 0.62 (Figura 20).

Figura 20

Programación de red neuronal opción 1 entrenamiento



```

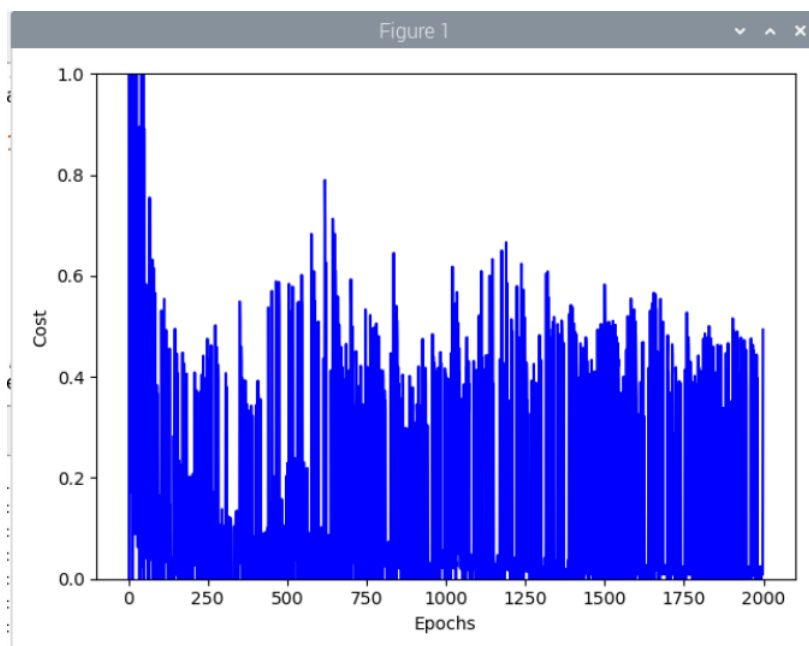
red1.py ✕ | PIR2.py ✕ | DATOS.py ✕
90
91     def get_deltas(self):
92         return self.deltas
93
94 nn = NeuralNetwork([3,2,1], activation='tanh')
95 X = np.array([[0, 0, 0],
96              [0, 0, 1],
97              [0, 1, 0],
98              [0, 1, 1],
99              [1, 0, 0],
100             [1, 0, 1],
101             [1, 1, 0],
102             [1, 1, 1]])
103 y = np.array([0, 1, 0, 1, 0, 1, 0, 1, 1, 1])
104 nn.fit(X, y, learning_rate=0.03, epochs=2000)
Shell
epochs: 0
Entradas: [0 0 0] Salidas: [-0.25058716]
Entradas: [0 0 1] Salidas: [0.92153445]
Entradas: [0 1 0] Salidas: [0.17852557]
Entradas: [0 1 1] Salidas: [0.93991276]
Entradas: [1 0 0] Salidas: [0.1899915]
Entradas: [1 0 1] Salidas: [0.95302166]
Entradas: [1 1 0] Salidas: [0.62446797]
Entradas: [1 1 1] Salidas: [0.95403855]
Python 3.7.3

```

Así se tiene como resultado que el coste de la función en este caso no tiende a 0 como se espera (Figura 21).

Figura 21

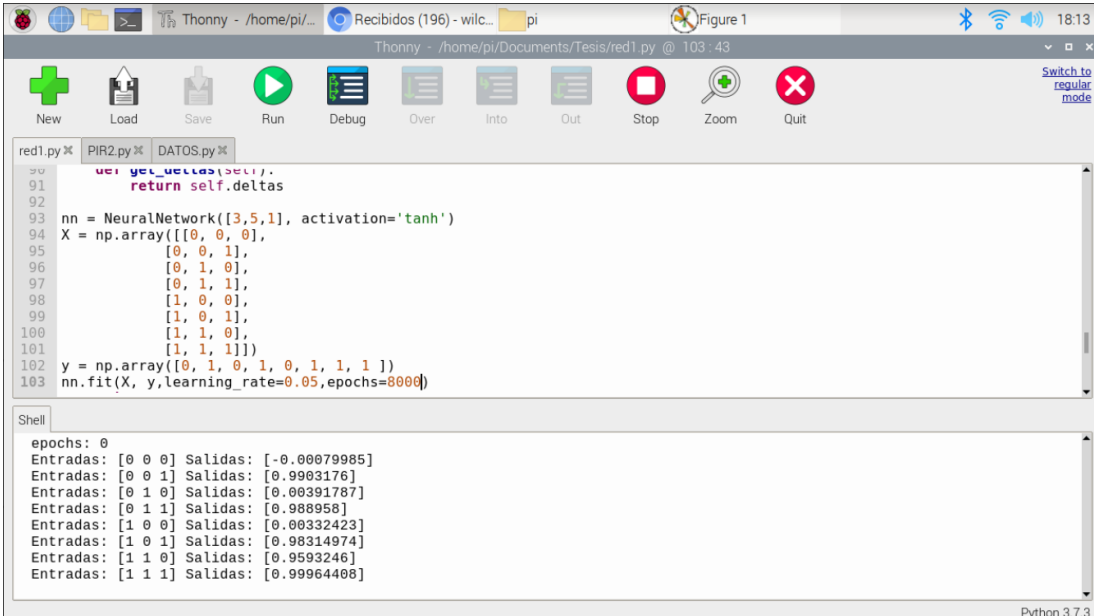
Coste de la función en el primer entrenamiento



Se procede a un segundo entrenamiento modificando el número de neuronas en la capa culta a 5, una tasa de aprendizaje de 0.05 y un epochs de 8000, con estos parámetros la red neuronal se comporta de mejor manera, mejorando la aproximación de los valores de salida a 1, con valores sobre el 0.95 (Figura 22).

Figura 22

Programación de red neuronal opción 2 entrenamiento



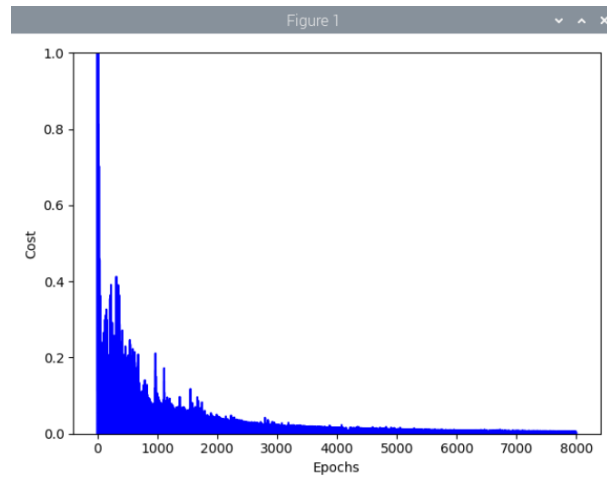
```
red1.py x PIR2.py x DATOS.py x
90 def get_deltas(self):
91     return self.deltas
92
93 nn = NeuralNetwork([3,5,1], activation='tanh')
94 X = np.array([[0, 0, 0],
95              [0, 0, 1],
96              [0, 1, 0],
97              [0, 1, 1],
98              [1, 0, 0],
99              [1, 0, 1],
100             [1, 1, 0],
101             [1, 1, 1]])
102 y = np.array([0, 1, 0, 1, 0, 1, 1, 1])
103 nn.fit(X, y, learning_rate=0.05, epochs=8000)

Shell
epochs: 0
Entradas: [0 0 0] Salidas: [-0.00079985]
Entradas: [0 0 1] Salidas: [0.9903176]
Entradas: [0 1 0] Salidas: [0.00391787]
Entradas: [0 1 1] Salidas: [0.988958]
Entradas: [1 0 0] Salidas: [0.00332423]
Entradas: [1 0 1] Salidas: [0.98314974]
Entradas: [1 1 0] Salidas: [0.9593246]
Entradas: [1 1 1] Salidas: [0.99964408]
```

También se obtiene la gráfica del coste de la función para esta configuración de parámetros en la opción de entrenamiento, en este caso en comparación del anterior se ve mejoría en los valores arrojados ya que esta opción si permite tender al valor de 0 como se espera (Figura 23).

Figura 23

Coste de la función en el segundo entrenamiento



Con la opción de mejorar los resultados para la salida esperada se realiza un tercer entrenamiento teniendo en esta ocasión 15 neuronas en la capa oculta, una tasa de aprendizaje de 0.08 y epochs de 15000, donde el entrenamiento devuelve valores por encima de 0.98 para los valores de 1 esperados como se observa en la figura 24.

Figura 24

Programación de red neuronal opción 3 entrenamiento

```

red1.py x PIR2.py x DATOS.py x
90     def get_deltas(self):
91         return self.deltas
92
93 nn = NeuralNetwork([3,15,1], activation='tanh')
94 X = np.array([[0, 0, 0],
95              [0, 0, 1],
96              [0, 1, 0],
97              [0, 1, 1],
98              [1, 0, 0],
99              [1, 0, 1],
100             [1, 1, 0],
101             [1, 1, 1]])
102 y = np.array([0, 1, 0, 1, 0, 1, 1, 1])
103 nn.fit(X, y, learning_rate=0.08, epochs=15000)

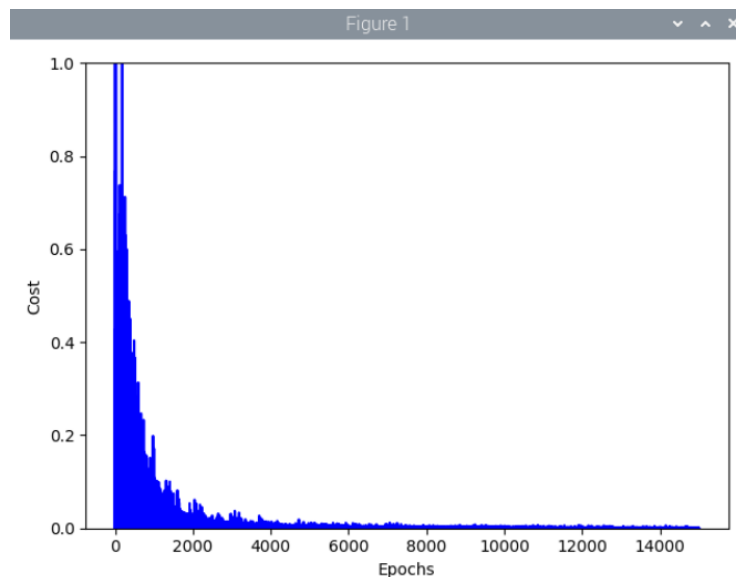
Shell
epochs: 10000
Entradas: [0 0 0] Salidas: [-0.00112569]
Entradas: [0 0 1] Salidas: [0.99632621]
Entradas: [0 1 0] Salidas: [4.53889666e-06]
Entradas: [0 1 1] Salidas: [0.98899425]
Entradas: [1 0 0] Salidas: [-0.00013829]
Entradas: [1 0 1] Salidas: [0.99966129]
Entradas: [1 1 0] Salidas: [0.98065166]
Entradas: [1 1 1] Salidas: [0.99962193]
Python 3.7.3

```

Obteniendo en este caso una mejor respuesta esperada para la salida para el encendido de las lámparas y focos del departamento, el costo de la función como se observa en la figura 25 tiende a 0 como es lo esperado.

Figura 25

Coste de la función en el segundo entrenamiento



Una vez obtenidos los resultados se aplican las condiciones del tercer entrenamiento a cada una de las secciones considerando las entradas de cada una de las mismas, la hora, el sensor de presencia de cada sección, y cada interruptor.

Ya implementado en las instalaciones del departamento con la programación adecuada se replica la red neuronal a cada una de las secciones, considerando el rango de hora determinado y las lecturas del sensor de presencia e interruptor de cada sección el tiempo de respuesta para la activación es de 1.5 a 2 segundos, y para el apagado si el sensor deja de emitir señal un tiempo de respuesta de 20 segundos, tomando en cuenta que con el envío de los datos para almacenaje en Google Sheet estos tiempo de respuesta aumenta 5 segundos para la activación y 30-40 segundos para la desactivación.

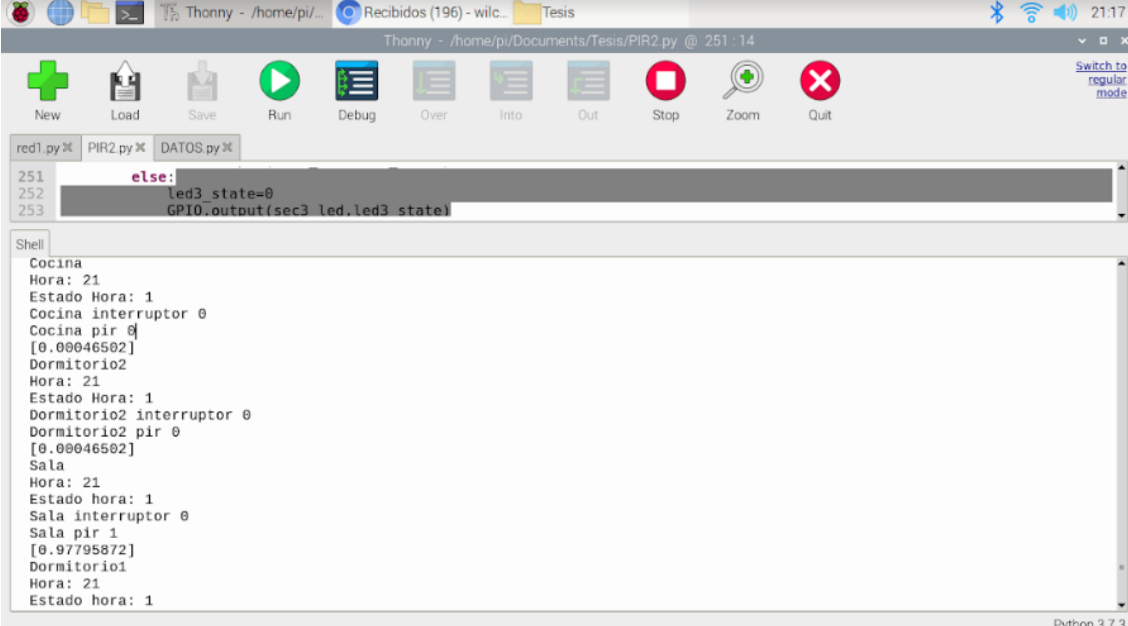
Análisis de resultados, conclusiones y recomendaciones

Análisis de resultados

El resultado de la red neuronal aplicada es la adecuada por el resultado esperado, ya que al brindar una respuesta próxima a la esperada en el caso de encendido de 1 y de apagado 0 nos da como predicción para el encendido de las luces de al menos 0.98 que se puede considerar bastante precisa y para el apagado con valores cercanos a 0 de -0.00013. En este caso se puede considerar un aproximado del resultado esperado del 100% ya que no se tiene resultados fuera de los esperados con las condiciones establecidas de hora, estado de sensor y estado de interruptor, se pueden observar varios datos en las siguientes figuras donde se observa que se está obteniendo la predicción esperada para cada circunstancia.

Figura 26

Datos obtenidos con red neuronal ejemplo 1



The screenshot shows the Thonny Python IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The editor window displays the following code:

```
251 else:
252     led3_state=0
253     GPIO.output(sec3_led,led3_state)
```

The Shell window shows the following output:

```
Cocina
Hora: 21
Estado Hora: 1
Cocina interruptor 0
Cocina pir 0
[0.00046502]
Dormitorio2
Hora: 21
Estado Hora: 1
Dormitorio2 interruptor 0
Dormitorio2 pir 0
[0.00046502]
Sala
Hora: 21
Estado hora: 1
Sala interruptor 0
Sala pir 1
[0.9795872]
Dormitorio1
Hora: 21
Estado hora: 1
```

En la Figura 26 se puede observar el caso de la Sección 6 Cocina donde la hora: 21 que está dentro del rango de hora para encendido, el estado del interruptor 0 y el estado

del sensor PIR en 0 por lo que la matriz de entrada para este caso es $[1,0,0]$ la respuesta esperada en base a la información de la tabla 4 es $[0]$, la predicción de la red neuronal nos da resultado $[0.00046502]$.

Figura 27

Datos obtenidos con red neuronal ejemplo2

```

251
252     else:
253         led3_state=0
           GPIO.output(sec3_led.led3_state)

```

```

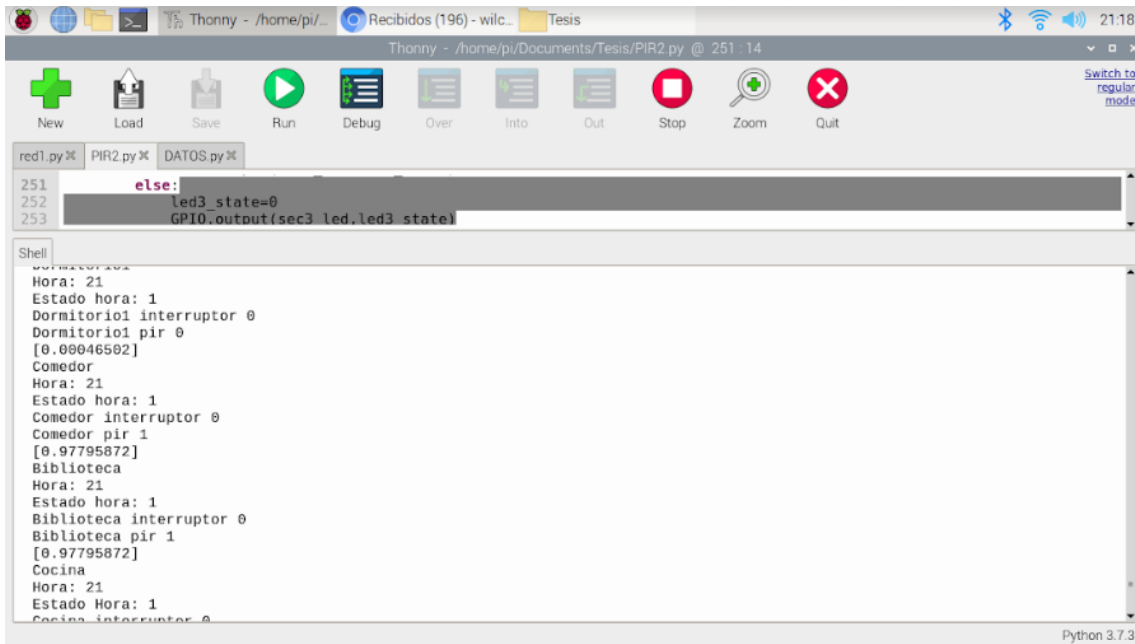
Shell
Dormitorio2
Hora: 21
Estado Hora: 1
Dormitorio2 interruptor 0
Dormitorio2 pir 0
[0.00046502]
Sala
Hora: 21
Estado hora: 1
Sala interruptor 0
Sala pir 1
[0.97795872]
Dormitorio1
Hora: 21
Estado hora: 1
Dormitorio1 interruptor 1
Dormitorio1 pir 1
[0.99967832]
Comedor
Hora: 21
Estado hora: 1

```

En la Figura 27 se puede observar el caso de la Sección 1 Domitorio1 donde la hora: 21 que está dentro del rango de hora para encendido, el estado del interruptor 1 y el estado del sensor PIR en 0 por lo que la matriz de entrada para este caso es $[1,1,0]$ la respuesta esperada en base a la información de la tabla 4 es $[1]$, la predicción de la red neuronal nos da resultado $[0.99967832]$.

Figura 28

Datos obtenidos con red neuronal ejemplo 3



The screenshot shows the Thonny Python IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Info, Out, Stop, Zoom, and Quit. The editor window displays a Python script with the following code:

```

251     else:
252         led3_state=0
253         GPIO.output(sec3_led.led3_state)

```

The Shell window below shows the output of the script, which is a series of sensor readings for different rooms:

```

Dormitorio1
Hora: 21
Estado hora: 1
Dormitorio1 interruptor 0
Dormitorio1 pir 0
[0.08046502]
Comedor
Hora: 21
Estado hora: 1
Comedor interruptor 0
Comedor pir 1
[0.97795872]
Biblioteca
Hora: 21
Estado hora: 1
Biblioteca interruptor 0
Biblioteca pir 1
[0.97795872]
Cocina
Hora: 21
Estado Hora: 1
Cocina interruptor 0

```

The bottom right corner of the IDE window indicates the Python version is 3.7.3.

En la Figura 28 se puede observar el caso de la Sección 4 Biblioteca donde la hora: 21 que está dentro del rango de hora para encendido, el estado del interruptor 0 y el estado del sensor PIR en 1 por lo que la matriz de entrada para este caso es [1,0,1] la respuesta esperada en base a la información de la tabla 4 es [1], la predicción de la red neuronal nos da resultado [0.97795872].

En la tabla 6 se puede observar un fragmento de los datos obtenidos ya aplicando la red neuronal donde se muestra el estado de las condiciones que son hora, estado de interruptor y sensor de presencia, haciendo la comparativa con la tabla 4 de datos de entrada y salida esperada, se obtiene que la red neuronal cumple con lo esperado.

Tabla 6

Datos tomados de la sección 5 Comedor

FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA
2022,07,23	23,11,13	0	1	1
2022,07,23	23,11,35	0	1	1
2022,07,23	23,11,57	0	1	1
2022,07,23	23,12,18	0	1	1
2022,07,23	23,12,40	0	1	1
2022,07,23	23,13,01	0	0	0
2022,07,23	23,13,22	0	0	0
2022,07,23	23,13,43	0	0	0
2022,07,23	23,14,05	0	1	1
2022,07,23	23,14,26	0	1	1
2022,07,23	23,14,48	0	1	1
2022,08,11	21,08,19	0	0	0
2022,08,11	21,08,52	1	0	1
2022,08,11	21,09,24	0	1	1
2022,08,11	21,09,56	0	1	1
2022,08,11	21,10,42	0	1	1
2022,08,11	21,11,14	0	0	0
2022,08,11	21,11,48	1	0	1
2022,08,11	21,12,25	0	0	0
2022,08,11	21,12,57	1	0	1
2022,08,11	21,13,38	0	1	1
2022,08,11	21,14,20	0	1	1
2022,08,11	21,14,51	0	1	1
2022,08,11	21,15,23	0	1	1

Conclusiones

La implementación de un sistema domótico físico en la vivienda se basó en la instalación de sensores de presencia en cada una de las secciones determinadas como áreas independientes, así como también se realizó la toma de datos para el accionamiento de los interruptores, así como también se instaló los respectivos relés para el control del encendido y apagado de las lámparas y focos led. Todos estos

dispositivos se instalaron de forma cableada la cual llega a una unidad de control en este caso una Raspberry pi 4 la cual toma los datos los procesa y envía una respuesta a los actuadores, y así se obtuvo el encendido manual y automático de las luces de una vivienda.

La implementación una red neuronal en el control domótico de iluminación de una vivienda mediante una red con topología backpropagation y aprendizaje supervisado para poder obtener una respuesta y predicción dio resultados óptimos ya que al tener unas entradas definidas y unas salidas ya conocidas el entrenamiento por backpropagation dio resultados cercanos a los esperados para el encendido se espera el valor de 1 y la predicción da como resultados 0.98 y para el valor esperado de 0 el valor más cercano que da la predicción de la red neuronal es de -0.00013, dándonos así una respuesta del 100% sin errores a la salida de malos encendidos de las luces, esta respuesta del 100% se da haciendo un análisis de los datos almacenados en el archivo de Google sheet considerando los datos preliminares de la tabla 4, así como también en sitio validando que se prendan y se apaguen las iluminarias en el momento que lo amerite.

La implementación del almacenamiento de los datos de estado de los dispositivos entienda sensor de presencia, interruptor y estado del encendido de lámpara o foco led de cada una de las secciones, permitirá el posterior análisis del consumo energético de la vivienda en cuanto a iluminación se refiere, dándole así una idea de cómo podría realizar un ahorro de consumo eléctrico.

Recomendaciones

Para futuros trabajos se puede realizar la implementación de más sensores, uno de ellos podría ser un sensor de luz y que entre como un nuevo parámetro de entrada y así

no depender de la hora como condicionante. Es decir, se entraría el campo del confort del ser humano en cuanto a la cantidad luminosa necesaria para estar en un ambiente apropiado, teniendo ya esta variable se podría realizar una regulación de las iluminarias dependiendo de la cantidad luminosa necesaria.

Mejorar el sistema de almacenaje de datos ya que por ser una plataforma exterior al del controlador estamos limitados por la velocidad de recepción de datos lo que hace que la respuesta del encendido y apagado se alargue y haya un desfase de tiempo. Se podría implementar dentro de la misma plataforma Raspberry con la opción de poder mostrar los datos almacenados de una manera mas interactiva mediante un HMI (Interfaz Humano Máquina).

Mejorar el sistema de topología de red, se podría realizar el cambio de sensores por cableado a sensores inalámbricos que permitan respetar la estética del departamento y no haya problemas por cables mal conectados y ruptura de cables. Esto nos permitiría una conexión de mayor cantidad de sensores sin la necesidad de seguir agregando cableado para la adquisición de datos y alimentación de los sensores.

Referencias

- Aprende Machine Learning*. (26 de julio de 2018). Obtenido de Crear una red neuronal en Python desde cero: <https://www.aprendemachinelearning.com/crear-una-red-neuronal-en-python-desde-cero/?fbclid=IwAR1V7PBo2SMi2qyxuAA8HcCmM-ONabtPUZ1JH1Dm3NsneAwxwaV8ejW5BUQ>
- Asociacion Española de Domotica e Inmotica. (2021). *Cedom*. Obtenido de Que es domótica: <http://www.cedom.es/sobre-domotica/que-es-domotica>
- Barrera, M., Londoño, N., Carvajal, J., & Alejandro, F. (2012). *Análisis de diseño de un prototipo de sistema domótico de bajo costo*. Medellín: Universidad de Antioquia.
- Camargo, E., Coronel, C., & Calderon, M. (2015). Hogar inteligente por control de voz usando redes neuronales. *Revista Colombiana de Tecnologías de Avanzada*, 16-17.

- Carrillo, M., & Vite, C. (2016). *Aplicacion de una red neuronal para el control domótico de encendido de luces en una vivienda de la urbanizacion Santa Angela-Chiclayo 2015*. Lambayeque: Universidad Nacional Pedro Ruiz Gallo.
- Cedeño, F. (2018). *Desarrollo de un sistema domótico y aplicación para dispositivos móviles Android para el control de luces*. Zumpango: Universidad Autónoma del estado de México.
- Computerworld. (marzo de 2021). *Computerworld powered by Ekos*. Obtenido de La domótica e inmótica en Ecuador: <http://computerworld.com.ec/actualidad/tendencias/128-domoticaeinmoticaecuador.html>
- Custodio, A. (5 de agosto de 2008). *Gestiopolis*. Obtenido de Métodos y técnicas de investigación científica: <https://www.gestiopolis.com/metodos-y-tecnicas-de-investigacion-cientifica/>
- Escobar, E., & Villazón, A. (2018). *Sistema de monitoreo energético y control domótico basado en tecnología internet de las cosas*. Cochabamba: Universidad Privada Boliviana.
- Espino, L., Perez, Y., & Cossio, E. (2020). *Propuesta de un sistema de control, monitoreo y asistencia para optimización de recursos energéticos en el hogar*. México: Computing Science.
- Garcia, E. (2009). *Vida e inteligencia artificial*. La Habana: ACIMED- SciELO.
- Gómez, L. (2021). La vida de hogar ahora es touch. *Lideres*.
- Henriquez, M., & Palma, P. (2011). Control automático de condiciones ambientales en domotica usando redes neuronales artificiales. *Scielo*, 125-139.
- Huamán, J. (2017). *Control inteligente de sistemas de iluminacion en edificios*. Piura: Universidad de Piura.
- Iberdrola. (2021). *Innovacion*. Obtenido de Que es machine learning: <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>
- Ibiblio. (2021). *Herramientas en GNU/Linux para estudiantes universitarios*. Obtenido de Conceptos básicos sobre RNA: https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html
- Innovation, A. (22 de octubre de 2019). *Atria Innovation*. Obtenido de Qué son las redes neuronales y sus funciones: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>
- Jara, A. (s.f.). *Introduccion a la domótica viviendas inteligentes*. Murcia: Universidad de Murcia.

- Mahecha, J. (2018). *Diseño e implementación de una aplicación domótica para iluminación usando inteligencia artificial*. Bogotá: Universidad de La Salle-Ciencia Unisalle.
- Martinez, A., & Gudiño, F. (2019). *Diseño e implementación de reconocimiento facial en un sistema domótico*. México: Universidad Nacional Autónoma de México.
- Morejon, L. (24 de Agosto de 2016). *Dialoguemos La academia en la comunidad*. Obtenido de La domótica en Ecuador esta en sus primeros pasos y la universidad debe dinamizar su uso: <https://dialoguemos.ec/2016/08/la-domotica-en-ecuador-esta-en-sus-primeros-pasos-y-la-universidad-debe-dinamizar-su-uso/>
- Parada, E., Illera, M., Sepúlveda, S., Guevara, D., & Medina, B. (2016). *Sistema de control domótico de bajo costo: un respaldo a la generación ecológica de energía eléctrica en Colombia*. Bogotá: Universidad distrital Francisco José de Caldas.
- Rebato, C. (14 de Febrero de 2020). *Think Big/Empresas*. Obtenido de Domótica e Inteligencia Artificial: del hogar inteligente a las Smart cities: <https://empresas.blogthinkbig.com/domotica/>
- Renovable, Ministerio de Electricidad y Energía. (2021). *Ministerio de Electricidad y Energía Renovable*. Recuperado el 18 de mayo de 2021, de Eficiencia Energética Sector Residencial: <http://historico.energia.gob.ec/eficiencia-energetica-sector-residencial/>
- Román, R. (2011). *Diseño de un sistema domótico para el control de iluminación y monitoreo de consumo eléctrico*. Bucaramanga: Universidad industrial de Santander.
- Smart, J. (marzo de 2021). *JEdi smart*. Obtenido de Cómo la domótica en Ecuador puede renovar tu hogar: <https://jedi.com.ec/domotica-ecuador/>
- Smart, J. (03 de 2021). *JEdi smart*. Obtenido de Beneficios de tener una smart Home en Ecuador: <https://jedi.com.ec/smart-home-en-ecuador/>
- Utrilla, D. (2014). *Sistema de arquitectura multisensorial para supervisión y seguridad industrial aplicando tecnología de inteligencia artificial*. Lima: Universidad Nacional Mayor de San Marcos.

Anexos

Anexo A Programación del sistema completo

```
import RPi.GPIO as GPIO
import time
import numpy as np
import gspread
from oauth2client.service_account import ServiceAccountCredentials
from datetime import datetime

sec5_int=37
sec5_pir=35
sec5_led=33
sec1_int=31
sec1_pir=29
sec1_led=23
sec2_int=40
sec2_pir=38
sec2_led=32
sec6_int=26
sec6_pir=24
sec6_led=22
sec3_int=18
sec3_pir=16
sec3_led=12
sec4_int=7
sec4_pir=15
sec4_led=13

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sec6_led,GPIO.OUT)
GPIO.setup(sec6_pir,GPIO.IN)
```

```
GPIO.setup(sec6_int,GPIO.IN)
GPIO.setup(sec2_led,GPIO.OUT)
GPIO.setup(sec2_pir,GPIO.IN)
GPIO.setup(sec2_int,GPIO.IN)
GPIO.setup(sec3_led,GPIO.OUT)
GPIO.setup(sec3_pir,GPIO.IN)
GPIO.setup(sec3_int,GPIO.IN)
GPIO.setup(sec1_led,GPIO.OUT)
GPIO.setup(sec1_pir,GPIO.IN)
GPIO.setup(sec1_int,GPIO.IN)
GPIO.setup(sec5_led,GPIO.OUT)
GPIO.setup(sec5_pir,GPIO.IN)
GPIO.setup(sec5_int,GPIO.IN)
GPIO.setup(sec4_led,GPIO.OUT)
GPIO.setup(sec4_pir,GPIO.IN)
GPIO.setup(sec4_int,GPIO.IN)
int6_state=0
pir6_state=0
led6_state=0
int2_state=0
pir2_state=0
led2_state=0
int3_state=0
pir3_state=0
led3_state=0
int1_state=0
pir1_state=0
led1_state=0
int5_state=0
pir5_state=0
led5_state=0
```

```
int4_state=0
pir4_state=0
led4_state=0
def sigmoid(x):
    return 1.0/(1.0 + np.exp(-x))

def sigmoid_derivada(x):
    return sigmoid(x)*(1.0-sigmoid(x))

def tanh(x):
    return np.tanh(x)

def tanh_derivada(x):
    return 1.0 - x**2

class NeuralNetwork:

    def __init__(self, layers, activation='tanh'):
        if activation == 'sigmoid':
            self.activation = sigmoid
            self.activation_prime = sigmoid_derivada
        elif activation == 'tanh':
            self.activation = tanh
            self.activation_prime = tanh_derivada

        # inicializo los pesos
        self.weights = []
        self.deltas = []
        # capas = [2,3,2]
        # rando de pesos varia entre (-1,1)
```

```

# asigno valores aleatorios a capa de entrada y capa oculta
for i in range(1, len(layers) - 1):
    r = 2*np.random.random((layers[i-1] + 1, layers[i] + 1)) - 1
    self.weights.append(r)

# asigno aleatorios a capa de salida
r = 2*np.random.random( (layers[i] + 1, layers[i+1])) - 1
self.weights.append(r)

def fit(self, X, y, learning_rate=0.2, epochs=100000):
    # Agrego columna de unos a las entradas X
    # Con esto agregamos la unidad de Bias a la capa de entrada
    ones = np.atleast_2d(np.ones(X.shape[0]))
    X = np.concatenate((ones.T, X), axis=1)

    for k in range(epochs):
        i = np.random.randint(X.shape[0])
        a = [X[i]]

        for l in range(len(self.weights)):
            dot_value = np.dot(a[l], self.weights[l])
            activation = self.activation(dot_value)
            a.append(activation)

        # Calculo la diferencia en la capa de salida y el valor obtenido
        error = y[i] - a[-1]
        deltas = [error * self.activation_prime(a[-1])]

        # Empezamos en el segundo layer hasta el ultimo
        # (Una capa anterior a la de salida)
        for l in range(len(a) - 2, 0, -1):
            deltas.append(deltas[-1].dot(self.weights[l].T)*self.activation_prime(a[l]))

        self.deltas.append(deltas)

```

```

# invertir
# [level3(output)->level2(hidden)] => [level2(hidden)->level3(output)]
deltas.reverse()

# backpropagation
# 1. Multiplicar los delta de salida con las activaciones de entrada
# para obtener el gradiente del peso.
# 2. actualizo el peso restandole un porcentaje del gradiente
for i in range(len(self.weights)):
    layer = np.atleast_2d(a[i])
    delta = np.atleast_2d(deltas[i])
    self.weights[i] += learning_rate * layer.T.dot(delta)

if k % 10000 == 0: print('epochs:', k)

def predict(self, x):
    ones = np.atleast_2d(np.ones(x.shape[0]))
    a = np.concatenate((np.ones(1).T, np.array(x)), axis=0)
    for l in range(0, len(self.weights)):
        a = self.activation(np.dot(a, self.weights[l]))
    return a

def print_weights(self):
    print("LISTADO PESOS DE CONEXIONES")
    for i in range(len(self.weights)):
        print(self.weights[i])

def get_deltas(self):
    return self.deltas

```

```

nn = NeuralNetwork([3,15,1], activation='tanh')
X = np.array([[0, 0, 0],
              [0, 0, 1],
              [0, 1, 0],
              [0, 1, 1],
              [1, 0, 0],
              [1, 0, 1],
              [1, 1, 0],
              [1, 1, 1]])
y = np.array([0, 1, 0, 1, 0, 1, 1, 1 ])
nn.fit(X, y, learning_rate=0.08, epochs=15000)

scope =
["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/spreadsheets",
https://www.googleapis.com/auth/drive.file", "https://www.googleapis.com/auth/drive"]

creds = ServiceAccountCredentials.from_json_keyfile_name("tesis.json", scope)
client = gspread.authorize(creds)
sheet = client.open("Tesis") #Abrir spreadhseet

try:
    while True:
        time.sleep(2.5)
        print("Cocina")
        int6_state=GPIO.input(sec6_int)
        pir6_state=GPIO.input(sec6_pir)
        HoraCocina=int(time.strftime('%H'))
        print("Hora: %s" %(HoraCocina))
        if HoraCocina>0 and HoraCocina<8 or HoraCocina>17 and HoraCocina<24:
            HC=1
        else:
            HC=0
        print("Estado Hora: %s" %(HC))
        print("Cocina interruptor %s" %(int6_state))

```



```

print("Cocina pir %s" %(pir6_state))
AC=np.array([HC, pir6_state, int6_state])
RC=nn.predict(AC)
print(RC)
if RC>0.9:
    led6_state=1
    GPIO.output(sec6_led,led6_state)
else:
    led6_state=0
    GPIO.output(sec6_led,led6_state)

sTimeStampDC=time.strftime('%Y'+','+ '%m' +','+ '%d')
sTimeStampHC=time.strftime('%H' +','+ '%M' +','+ '%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSCOCINA=sheet.worksheet("COCINA")
col = DATOSCOCINA.col_values(1)
data = DATOSCOCINA.get_all_records() #Obtener todos los registros

DATOSCOCINA.append_row([sTimeStampDC,sTimeStampHC,int6_state,pir6_state,led6_state])
#####
##

time.sleep(2.5)
print("Dormitorio2")
int2_state=GPIO.input(sec2_int)
pir2_state=GPIO.input(sec2_pir)
HoraDormitorio2=int(time.strftime('%H'))
print("Hora: %s" %(HoraDormitorio2))

if HoraDormitorio2>0 and HoraDormitorio2<8 or HoraDormitorio2>17 and
HoraDormitorio2<24:
    HD2=1
else:
    HD2=0

```

```

print("Estado Hora: %s" %(HD2))
print("Dormitorio2 interruptor %s" %(int2_state))
print("Dormitorio2 pir %s" %(pir2_state))
AD2=np.array([HD2, pir2_state, int2_state])
RD2=nn.predict(AD2)
print(RD2)
if RD2>0.9:
    led2_state=1
    GPIO.output(sec2_led,led2_state)
else:
    led2_state=0
    GPIO.output(sec2_led,led2_state)

sTimeStampDD2=time.strftime('%Y'+','+'%m'+','+'%d')
sTimeStampHD2=time.strftime('%H'+','+'%M'+','+'%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSDORMITORIO2=sheet.worksheet("DORMITORIO2")
col = DATOSDORMITORIO2.col_values(1)
data = DATOSDORMITORIO2.get_all_records() #Obtener todos los registros

DATOSDORMITORIO2.append_row([sTimeStampDD2,sTimeStampHD2,int2_state,pir2_state,le
d2_state])

#####
#####

time.sleep(2.5)
print("Sala")
int3_state=GPIO.input(sec3_int)
pir3_state=GPIO.input(sec3_pir)
HoraSala=int(time.strftime('%H'))
print("Hora: %s" %(HoraSala))
if HoraSala>0 and HoraSala<8 or HoraSala>17 and HoraSala<24:
    HS=1

```

```

else:
    HS=0
    print("Estado hora: %s" %(HS))
    print("Sala interruptor %s" %(int3_state))
    print("Sala pir %s" %(pir3_state))
    AS=np.array([HS, pir3_state, int3_state])
    RS=nn.predict(AS)
    print(RS)
    if RS>0.9:
        led3_state=1
        GPIO.output(sec3_led,led3_state)
    else:
        led3_state=0
        GPIO.output(sec3_led,led3_state)

sTimeStampDS=time.strftime('%Y'+','+'%m'+','+'%d')
sTimeStampHS=time.strftime('%H'+','+'%M'+','+'%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSSALA=sheet.worksheet("SALA")
col = DATOSSALA.col_values(1)
data = DATOSSALA.get_all_records() #Obtener todos los registros

DATOSSALA.append_row([sTimeStampDS,sTimeStampHS,int3_state,pir3_state,led3_state])
#####
##

time.sleep(2.5)
print("Dormitorio1")
int1_state=GPIO.input(sec1_int)
pir1_state=GPIO.input(sec1_pir)
HoraDormitorio1=int(time.strftime('%H'))
print("Hora: %s" %(HoraDormitorio1))

```

```

if HoraDormitorio1>0 and HoraDormitorio1<8 or HoraDormitorio1>17 and
HoraDormitorio1<24:
    HD1=1
else:
    HD1=0
print("Estado hora: %s" %(HD1))
print("Dormitorio1 interruptor %s" %(int1_state))
print("Dormitorio1 pir %s" %(pir1_state))
AD1=np.array([HD1, pir1_state, int1_state])
RD1=nn.predict(AD1)
print(RD1)
if RD1>0.9:
    led1_state=1
    GPIO.output(sec1_led,led1_state)
else:
    led1_state=0
    GPIO.output(sec1_led,led1_state)

sTimeStampDD1=time.strftime('%Y'+','+'%m'+','+'%d')
sTimeStampHD1=time.strftime('%H'+','+'%M'+','+'%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSDORMITORIO1=sheet.worksheet("DORMITORIO1")
col = DATOSDORMITORIO1.col_values(1)
data = DATOSDORMITORIO1.get_all_records() #Obtener todos los registros

DATOSDORMITORIO1.append_row([sTimeStampDD1,sTimeStampHD1,int1_state,pir1_state,le
d1_state])

#####
#####

time.sleep(2.5)
print("Comedor")
int5_state=GPIO.input(sec5_int)

```

```

pir5_state=GPIO.input(sec5_pir)
HoraComedor=int(time.strftime('%H'))
print("Hora: %s" %(HoraComedor))
if HoraComedor>0 and HoraComedor<8 or HoraComedor>17 and HoraComedor<24:
    HCM=1
else:
    HCM=0
print("Estado hora: %s" %(HCM))
print("Comedor interruptor %s" %(int5_state))
print("Comedor pir %s" %(pir5_state))
ACM=np.array([HCM, pir5_state, int5_state])
RCM=nn.predict(ACM)
print(RCM)
if RCM>0.9:
    led5_state=1
    GPIO.output(sec5_led,led5_state)
else:
    led5_state=0
    GPIO.output(sec5_led,led5_state)

sTimeStampDCO=time.strftime('%Y'+','+'%m'+','+'%d')
sTimeStampHCO=time.strftime('%H'+','+'%M'+','+'%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSCOMEDOR=sheet.worksheet("COMEDOR")
col = DATOSCOMEDOR.col_values(1)
data = DATOSCOMEDOR.get_all_records() #Obtener todos los registros

DATOSCOMEDOR.append_row([sTimeStampDCO,sTimeStampHCO,int5_state,pir5_state,led5_
state])

#####
#####

time.sleep(2.5)

```

```

print("Biblioteca")
int4_state=GPIO.input(sec4_int)
pir4_state=GPIO.input(sec4_pir)
HoraBiblioteca=int(time.strftime('%H'))
print("Hora: %s" %(HoraBiblioteca))
if HoraBiblioteca>0 and HoraBiblioteca<8 or HoraBiblioteca>17 and HoraBiblioteca<24:
    HB=1
else:
    HB=0
print("Estado hora: %s" %(HB))
print("Biblioteca interruptor %s" %(int4_state))
print("Biblioteca pir %s" %(pir4_state))
AB=np.array([HB, pir4_state, int4_state])
RB=nn.predict(AB)
print(RB)
if RB>0.9:
    led4_state=1
    GPIO.output(sec4_led,led4_state)
else:
    led4_state=0
    GPIO.output(sec4_led,led4_state)

sTimeStampDB=time.strftime('%Y'+','+'%m'+','+'%d')
sTimeStampHB=time.strftime('%H'+','+'%M'+','+'%S')
#sheet = client.open("Tesis") #Abrir spreadhseet
DATOSBIBLIOTECA=sheet.worksheet("BIBLIOTECA")
col = DATOSBIBLIOTECA.col_values(1)
data = DATOSBIBLIOTECA.get_all_records() #Obtener todos los registros

DATOSBIBLIOTECA.append_row([sTimeStampDB,sTimeStampHB,int4_state,pir4_state,led4_state])

```

```
#####
#####
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

Anexo B Programación de la red neuronal

```
import numpy as np

def sigmoid(x):
    return 1.0/(1.0 + np.exp(-x))

def sigmoid_derivada(x):
    return sigmoid(x)*(1.0-sigmoid(x))

def tanh(x):
    return np.tanh(x)

def tanh_derivada(x):
    return 1.0 - x**2

class NeuralNetwork:

    def __init__(self, layers, activation='tanh'):
        if activation == 'sigmoid':
            self.activation = sigmoid
            self.activation_prime = sigmoid_derivada
        elif activation == 'tanh':
            self.activation = tanh
            self.activation_prime = tanh_derivada
```

```

# inicializo los pesos
self.weights = []
self.deltas = []
# capas = [2,3,2]
# rando de pesos varia entre (-1,1)
# asigno valores aleatorios a capa de entrada y capa oculta
for i in range(1, len(layers) - 1):
    r = 2*np.random.random((layers[i-1] + 1, layers[i] + 1)) - 1
    self.weights.append(r)
# asigno aleatorios a capa de salida
r = 2*np.random.random( (layers[i] + 1, layers[i+1])) - 1
self.weights.append(r)

def fit(self, X, y, learning_rate=0.2, epochs=100000):
    # Agrego columna de unos a las entradas X
    # Con esto agregamos la unidad de Bias a la capa de entrada
    ones = np.atleast_2d(np.ones(X.shape[0]))
    X = np.concatenate((ones.T, X), axis=1)

    for k in range(epochs):
        i = np.random.randint(X.shape[0])
        a = [X[i]]

        for l in range(len(self.weights)):
            dot_value = np.dot(a[l], self.weights[l])
            activation = self.activation(dot_value)
            a.append(activation)

        # Calculo la diferencia en la capa de salida y el valor obtenido
        error = y[i] - a[-1]
        deltas = [error * self.activation_prime(a[-1])]

```



```

# Empezamos en el segundo layer hasta el ultimo
# (Una capa anterior a la de salida)
for l in range(len(a) - 2, 0, -1):
    deltas.append(deltas[-1].dot(self.weights[l].T)*self.activation_prime(a[l]))
self.deltas.append(deltas)

# invertir
# [level3(output)->level2(hidden)] => [level2(hidden)->level3(output)]
deltas.reverse()

# backpropagation
# 1. Multiplicar los delta de salida con las activaciones de entrada
# para obtener el gradiente del peso.
# 2. actualizo el peso restandole un porcentaje del gradiente
for i in range(len(self.weights)):
    layer = np.atleast_2d(a[i])
    delta = np.atleast_2d(deltas[i])
    self.weights[i] += learning_rate * layer.T.dot(delta)

if k % 10000 == 0: print('epochs:', k)

def predict(self, x):
    ones = np.atleast_2d(np.ones(x.shape[0]))
    a = np.concatenate((np.ones(1).T, np.array(x)), axis=0)
    for l in range(0, len(self.weights)):
        a = self.activation(np.dot(a, self.weights[l]))
    return a

def print_weights(self):
    print("LISTADO PESOS DE CONEXIONES")
    for i in range(len(self.weights)):
        print(self.weights[i])

```

```

def get_deltas(self):
    return self.deltas

nn = NeuralNetwork([3,15,1], activation='tanh')
X = np.array([[0, 0, 0],
              [0, 0, 1],
              [0, 1, 0],
              [0, 1, 1],
              [1, 0, 0],
              [1, 0, 1],
              [1, 1, 0],
              [1, 1, 1]])
y = np.array([0, 1, 0, 1, 0, 1, 1, 1 ])
nn.fit(X, y, learning_rate=0.08, epochs=15000)
for e in X:
    print("Entradas:",e,"Salidas:",nn.predict(e))

import matplotlib.pyplot as plt

deltas=nn.get_deltas()
valores=[]
index=0
for arreglo in deltas:
    valores.append(arreglo[1][0]+arreglo[1][0])
    index=index+1

plt.plot(range(len(valores)),valores, color='b')
plt.ylim([0,1])
plt.ylabel('Cost')
plt.xlabel('Epochs')
plt.tight_layout()

```


Figura 31

Ejemplo toma de datos Sección 3 Sala

FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA
2022.07.23	23,11,05	0	0	0
2022.07.23	23,11,28	0	1	1
2022.07.23	23,11,49	0	1	1
2022.07.23	23,12,11	0	0	0
2022.07.23	23,12,33	0	0	0
2022.07.23	23,12,54	0	0	0
2022.07.23	23,13,15	0	0	0
2022.07.23	23,13,36	0	0	0
2022.07.23	23,13,58	0	1	1
2022.07.23	23,14,19	0	1	1
2022.07.23	23,14,40	0	1	1
2022.07.23	23,15,02	0	0	0
2022.07.23	23,15,23	0	0	0
2022.07.23	23,15,45	0	0	0
2022.07.23	23,16,06	0	0	0
2022.07.23	23,16,28	0	0	0
2022.07.23	23,16,49	0	0	0
2022.07.23	23,17,11	0	0	0
2022.07.23	23,17,32	0	0	0
2022.07.23	23,17,53	0	1	1
2022.07.23	23,18,15	0	1	1
2022.07.23	23,18,36	0	1	1
2022.07.23	23,18,57	0	0	0
2022.07.23	23,19,18	0	0	0
2022.07.23	23,19,40	0	0	0

Figura 32

Ejemplo toma de datos Sección 4 Biblioteca

FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA
2022.07.23	23,11,17	0	0	0
2022.07.23	23,11,39	0	0	0
2022.07.23	23,12,00	0	0	0
2022.07.23	23,12,21	0	0	0
2022.07.23	23,12,43	0	0	0
2022.07.23	23,13,04	0	0	0
2022.07.23	23,13,26	0	0	0
2022.07.23	23,13,47	0	0	0
2022.07.23	23,14,09	0	0	0
2022.07.23	23,14,30	0	0	0
2022.07.23	23,14,51	0	0	0
2022.07.23	23,15,12	0	0	0
2022.07.23	23,15,34	0	0	0
2022.07.23	23,15,55	0	0	0
2022.07.23	23,16,17	0	0	0
2022.07.23	23,16,38	0	0	0
2022.07.23	23,17,00	0	0	0
2022.07.23	23,17,21	0	0	0
2022.07.23	23,17,43	0	0	0
2022.07.23	23,18,04	0	0	0
2022.07.23	23,18,26	0	0	0
2022.07.23	23,18,47	0	0	0
2022.07.23	23,19,08	0	0	0
2022.07.23	23,19,29	0	0	0
2022.07.23	23,19,51	0	0	0

Figura 33

Ejemplo toma de datos Sección 5 Comedor

	A	B	C	D	E	F	G	H
1	FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA			
2	2022.07.23	23,11,13	0	1	1			
3	2022.07.23	23,11,35	0	1	1			
4	2022.07.23	23,11,57	0	1	1			
5	2022.07.23	23,12,18	0	1	1			
6	2022.07.23	23,12,40	0	1	1			
7	2022.07.23	23,13,01	0	0	0			
8	2022.07.23	23,13,22	0	0	0			
9	2022.07.23	23,13,43	0	0	0			
10	2022.07.23	23,14,05	0	1	1			
11	2022.07.23	23,14,26	0	1	1			
12	2022.07.23	23,14,48	0	1	1			
13	2022.07.23	23,15,09	0	1	1			
14	2022.07.23	23,15,31	0	1	1			
15	2022.07.23	23,15,52	0	0	0			
16	2022.07.23	23,16,13	0	0	0			
17	2022.07.23	23,16,35	0	0	0			
18	2022.07.23	23,16,56	0	0	0			
19	2022.07.23	23,17,18	0	0	0			
20	2022.07.23	23,17,39	0	1	1			
21	2022.07.23	23,18,01	0	1	1			
22	2022.07.23	23,18,22	0	1	1			
23	2022.07.23	23,18,43	0	1	1			
24	2022.07.23	23,19,04	0	1	1			
25	2022.07.23	23,19,26	0	0	0			
26	2022.07.23	23,19,47	0	0	0			

Figura 34

Ejemplo toma de datos Sección 6 Cocina

	A	B	C	D	E	F	G	H
1	FECHA	HORA	ESTADO INTERRUPTOR	SENSOR PRESENCIA	ESTADO LAMPARA			
2	2022.07.23	23,10,56	0	0	0			
3	2022.07.23	23,11,20	0	0	0			
4	2022.07.23	23,11,42	0	1	1			
5	2022.07.23	23,12,04	0	0	0			
6	2022.07.23	23,12,25	0	0	0			
7	2022.07.23	23,12,47	0	0	0			
8	2022.07.23	23,13,08	0	0	0			
9	2022.07.23	23,13,29	0	0	0			
10	2022.07.23	23,13,51	0	1	1			
11	2022.07.23	23,14,12	0	1	1			
12	2022.07.23	23,14,33	0	0	0			
13	2022.07.23	23,14,55	0	0	0			
14	2022.07.23	23,15,16	0	0	0			
15	2022.07.23	23,15,38	0	0	0			
16	2022.07.23	23,15,59	0	0	0			
17	2022.07.23	23,16,20	0	0	0			
18	2022.07.23	23,16,42	0	0	0			
19	2022.07.23	23,17,04	0	0	0			
20	2022.07.23	23,17,25	0	0	0			
21	2022.07.23	23,17,46	0	0	0			
22	2022.07.23	23,18,08	0	1	1			
23	2022.07.23	23,18,29	0	0	0			
24	2022.07.23	23,18,50	0	0	0			
25	2022.07.23	23,19,11	0	0	0			
26	2022.07.23	23,19,33	0	0	0			

Anexo D Planilla

Figura 35


Planilla de luz de la vivienda



Empresa Eléctrica Quito S.A.E.E.Q.
Matriz: Bartolome de las Casas E1-24 y Av. 10 de Agosto

Ruc: 1790053881001
Contribuyente especial, resolución No. 5368
OBLIGADO A LLEVAR CONTABILIDAD

Nro. factura 001-999-061777431
Nro. doc. interno 2791115181
Fecha de emisión 23-12-2021
Fecha de Vencimiento 07-01-2022
Número de autorización 2312202101179005388100120019990617774310087812916



K200008610400

VALOR TOTAL: 5,07

Información del Consumidor

CUENTA CONTRATO 200008610400 Código Único Eléctrico 1401080320

Nombre cliente WILCHEZ ARICHABALA EFREN ALFREDO (WILCHEZ ARICHABALA EFREN ALFREDO)

Cédula 0700910029 Tipo de tarifa Arconel BTCRSD01 - BT Residencial

Celular 0991505491 Geocódigo 1420M016000697 Unidad de Lectura 1420M016

Correo Electrónico alfredowilches@hotmail.com

Dirección del servicio PASAJE JUNCOS 15 PASAJE N64B / COMITE DEL PUEBLO PB / LOT. / COMITÉ DEL PUEBLO - QUITO

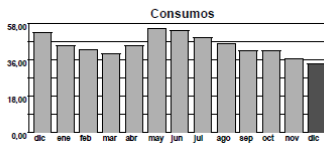
1. Información Servicio Eléctrico y Alumbrado Público

Número de medidor 1002146816
Tipo de consumo leído
Fecha desde 26-11-2021
Días facturados 28
Fecha hasta 23-12-2021

Descripción	Fecha Hasta	Lectura Actual	Lectura Anterior	Diferencia Consumo	Consumo Subtotal	Consumo interno Transformador	Consumo Total	Unidad Medida	Monto (\$)
Energía activa total	23-12-2021	476,00	440,00	0,00	36,00	0,00	36,00	kWh	2,82

2. Valores Pendientes

VALORES PENDIENTES (2)	0,00
------------------------	------



Servicio Eléctrico y Alumbrado Público

Valor Consumo	2,82
Comercialización	1,41
Subsidio Cruzado Beneficiado	1,55
Subsidio Tarifa Dignidad Comer.	0,54
Subtotal Servicio Eléctrico (SE)	2,14
Servicio Alumbrado Público General	0,25
Subtotal Alumbrado Público (APG)	0,25
Intereses por Mora	0,01
Subtotal Otros Rubros	0,01
Base I.V.A. 0%	2,93
I.V.A. 0%	0,00
Base Exento de IVA	0,53
Exento de IVA	0,00
TOTAL SE Y APG (1)	2,40

3. Planes de Financiamiento Autorizados por el Consumidor

PLANES DE FINANCIAMIENTO (3)	0,00
------------------------------	------