DISEÑO E IMPLEMENTACIÓN DE UN MONITOR NO INTRUSIVO DE POTENCIA DE ELECTRODOMÉSTICOS

ANDRÉS FRANCISCO SALGADO PEÑA
SERGIO ANDRÉS AVILA PELÁEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍAS
ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2021

DISEÑO E IMPLEMENTACIÓN DE UN MONITOR NO INTRUSIVO DE POTENCIA DE ELECTRODOMÉSTICOS

ANDRÉS FRANCISCO SALGADO PEÑA
SERGIO ANDRÉS AVILA PELÁEZ

Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director
Yulieth Jiménez Manjarrés
Dra. Ingeniería Electrónica

CO-DIRECTOR
Carlos A. Angulo Julio
Magíster en Ingeniería Electrónica

CO-DIRECTOR
César A. Duarte Gualdrón
PhD. Electrical and Computer Engineering

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍAS
ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2021

DEDICATORIA

A Dios porque sin él nada de esto sería realidad.

A mi madre por su amor y apoyo incondicional,
por cada una de las veces en qué se convirtió
en mi fuerza, inspiración y mi motor.

A mi padre por su voz de aliento en momentos
difíciles, por ver en mí, sus sueños cumplidos
y luchar a diario por darme lo mejor.

A mis hermanos por su apoyo y afecto.

A doña Amanda por hacerme sentir miembro mas
de su familia en los años que viví en Bucaramanga.

A doña Sandra y doña Margarita, por que me
acogieron en mis años de inicio de carrera en Barbosa.

A mi novia por su apoyo incondicional en cada
momento de dificultad, por esa ayuda y creer en mi
para cumplir mis metas.
— Andrés Salgado

Este proyecto de grado está dedicado a mi madre,
sin su apoyo, comprensión, amor y ejemplo
no habría llegado hasta este punto en mi vida,
gracias por mantenerme enfocado y por recordarme siempre
que lo más importante e incondicional será la familia.

A mi padre por ser una gran compañía
A mi hermana por su paciencia y comprensión durante todo este proceso.

A toda mi familia, por el apoyo que me han brindado durante toda la vida.

A mis amigos y profesores, por todas sus enseñanzas y consejos.
— Sergio Avila

AGRADECIMIENTOS

A **los profesores Yulieth Jiménez, César Gualdron y Carlos Angulo**, por su acompañamiento, colaboración y asesoria durante todo el proceso.

A **nuestros familiares**, por su apoyo incondicional, por su paciencia y persistencia en mantener claros nuestros objetivos.

A **nuestros compañeros** y amigos con quienes compartimos grandes momentos a lo largo de esta carrera.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

GLOSARIO

Clasificador es un modelo encargado de predecir la clase de un grupo de datos, se puede resumir como la tarea de aproximar una función de mapeo F con variables de entradas X a una salida variable discreta Y.

Colecciones Son la primera estructura de datos en las bases de datos de Firebase, se podrian comparar con los esquemas en SQL.

CSS es un lenguaje de estilos usado para definir el color, tamaño y espaciado de un contenido de una página web, también sirve para realizar animaciones y crear la estructura responsiva.

DataFrame es un tipo de dato definido por la librería PANDAS de Python cuya estructura son arreglos de dos dimensiones. Se caracteriza por su facilidad para manejar grandes volúmenes de datos con las herramientas de la librería.

HTML es un lenguaje con el que se estructuran las páginas web, está compuesto por etiquetas que marcan el inicio y fin de cada elemento de la página.

JavaScript es un lenguaje de programación orientado a objetos que permite crear contenido dinámico, éste funciona del lado de usuario e interpreta todas las peticiones realizadas por los navegadores.

SQL es un lenguaje de programación que permite realizar consultas con el fin de recuperar información de las bases de datos con estructura relacional.

TypeScript es un lenguaje de programación de código abierto basado en JavaScript, que añade tipado estático y objetos basados en clases.

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UN MONITOR NO INTRUSIVO DE POTENCIA DE ELECTRODOMÉSTICOS [*]

AUTORES: ANDRÉS FRANCISCO SALGADO PEÑA, SERGIO ANDRÉS ÁVILA PELÁEZ [**]

PALABRAS CLAVE: DESAGREGACIÓN DE CARGA, ELECTRODOMÉSTICO, IDENTIFICACIÓN DE CARGA, MONITORIZACIÓN DE POTENCIA, NO INTRUSIVO.

DESCRIPCIÓN:

La monitorización no intrusiva se encarga de identificar los cambios de voltaje y corriente generados por los electrodomésticos en la línea principal de una casa, de manera que se determine cuáles son los dispositivos conectados y cuál es su consumo energético sin necesidad de medir cada uno de los elementos de forma individual. Previamente en una tesis doctoral se habían diseñado algoritmos aplicados a monitorización intrusiva. En este trabajo de grado estos fueron traducidos al lenguaje de programación Python y documentados utilizando Notebooks de Jupyter, para luego ser migrados y probados en una Raspberry Pi 3 B +. Entre las librerías principales se resalta el uso de Numpy, Pandas y Tensorflow, las cuales permitieron el procesamiento de datos de forma eficiente. Para ello, se emularon señales de entrada asociadas a mediciones reales de electrodomésticos, se calcularon sus características y se ingresaron estos resultados a un clasificador, encargado de predecir cuál dispositivo fue encendido. Finalmente, los datos del dispositivo detectado son transmitidos de forma inalámbrica utilizando Wifi a una base de datos desarrollada en Firebase. Se utilizó la herramienta Firestore Cloud, esto con el fin de visualizar los resultados en una página web desarrollada en Angular y alojada en el hosting de Firebase que se diseñó y adecuó a la aplicación.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Yulieth Jiménez Manjarrés, Dra. Ingeniería Electrónica, Co-Director: César Antonio Duarte, PhD. Electrical and Computer Engineering, Co-Director: Carlos Andrés Angulo Julio, Magíster en Ingeniería Electrónica.

ABSTRACT

TITLE: DESIGN AND IMPLEMENTATION OF A HOME APPLIANCE NON-INTRUSIVE LOAD MONITORING [*]

AUTHORS: ANDRÉS FRANCISCO SALGADO PEÑA, SERGIO ANDRÉS ÁVILA PELÁEZ [**]

KEYWORDS: CHARGE DISAGGREGATION, HOME APPLIANCE, CHARGE IDENTIFICATION, POWER MONITORING, NON INTRUSIVE.

DESCRIPTION:

Non-intrusive load monitoring devotes to identify and analyze the current and voltage changes generated by home appliances on the main home line, in order to determine the devices connected and their power consumption without individual measurements. Previously, non-intrusive load monitoring algorithms were designed in a doctoral dissertation. These algorithms were translated to Python language and documented using Jupyter Notebooks in this undergrad work, and furthermore they were migrated and tested in a Raspberry Pi 3 B +. Main libraries are Numpy, Pandas and Tensorflow, which allowed an efficiently data processing. For this purpose, input signals associated to real measurements of home appliances were emulated. Moreover, their features were calculated and used as inputs to a classifier to deliver the prediction of the input device. Finally, data of the detected device were transmitted wireless using WiFi toward a database developed on Firebase. The Firestore Cloud tool was used to visualize the results in a website developed in Angular and stored in the Firebase hosting that was designed for the application.

---

[*] Bachelor Thesis

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Yulieth Jiménez Manjarrés, Dra. Ingeniería Electrónica, Co-Director: César Antonio Duarte, PhD. Electrical and Computer Engineering, Co-Director: Carlos Andrés Angulo Julio, Magíster en Ingeniería Electrónica.

INTRODUCCIÓN

La alta demanda de energía eléctrica a nivel mundial es una situación preocupante ya que se prevé que alcance los 33300 TWh para el 2030, donde en el año 2017 alcanzó un incremento anual del 1.9 %, el más grande desde el 2010 [1]. En América Latina y el Caribe la demanda energética ha ido creciendo a una tasa del 3.2 % de 2000 a 2014 y se estima que la demanda de electricidad aumente de 2.3 a 2.7 veces hasta 2060 [1]. Por tal razón, es indispensable buscar soluciones para poder solventar esta gran demanda energética, ya que podría generarse una crisis como la ocurrida en los años 2015 y 2016 en Colombia donde hubo una escasez de energía provocada por la disminución en la capacidad para suministrar la demanda debido al fenómeno de El Niño [2].

Se pensaría que una solución a estos problemas sería incrementar el suministro de energía. Sin embargo, esto tendría fuertes impactos en el medio ambiente y en la economía por la necesidad de suplir dicho aumento de energía, lo cual lleva a generar altas inversiones de dinero. La eficiencia energética se presenta como una alternativa a esta problemática. Estudios han demostrado que mejoras en este campo hechas desde el año 2000 evitaron el uso de un 12 % de energía adicional, así como la generación de un 12 % más de gases de efecto invernadero y 20 % más de importaciones de combustibles fósiles [3]. Una de las formas de promover la eficien-

---

1 World-Energy-Council. “Escenarios Energéticos Mundiales 2017”. En: (2017), pág. 3.

2 Mateus Valencia. “Crisis Energética en Colombia”. En: *Revistas UD (Universidad distrital Francisco José de Caldas)* 4 (2016).

3 IEA. “Market Report Series energy efficiency 2017”. En: *IEA Publ* (2017), págs. 1-143.

cia consiste en brindar información detallada del consumo, lo cual ha demostrado que los usuarios podrían disminuir su gasto energético entre un 5 % y 15 % [4]. Sin embargo, estos datos que realimentan al usuario usualmente se presentan de forma general.

Un enfoque diferente, conocido como desagregación de carga, consiste en entregar cifras del consumo de cada electrodoméstico, permitiéndole al usuario conocer cuál es el elemento que está consumiendo más energía en la vivienda. Esto ha logrado un efecto positivo, ya que, según estudios realizados, el 81 % de los hogares analizados encontraron útil la información y un 38 % demostró haber aprendido algo nuevo de ello. El resultado de estos cambios fue un mayor ahorro de energía en las viviendas [4].

En el grupo GISEL de la Universidad Industrial de Santander se vienen desarrollando investigaciones referentes a la monitorización no intrusiva mediante el estudio de desagregación de carga. Dichas investigaciones arrojaron como resultado el desarrollo de algoritmos diseñados en una tesis doctoral [5] para poder identificar el electrodoméstico y la energía que éste consume en una vivienda. Para ello, se requiere implementar el sistema en un dispositivo portable y visualizar resultados en un aplicativo web.

El presente trabajo de grado se desarrolló teniendo como propósito poder implementar algoritmos de desagregación de carga en una tarjeta de desarrollo, con el

---

[4] Sarah Darby. “The effectiveness of feedback on energy consumption”. En: *A Review for DEFRA of the Literature on Metering, Billing and direct Displays486.2006* (2006), pág. 26.

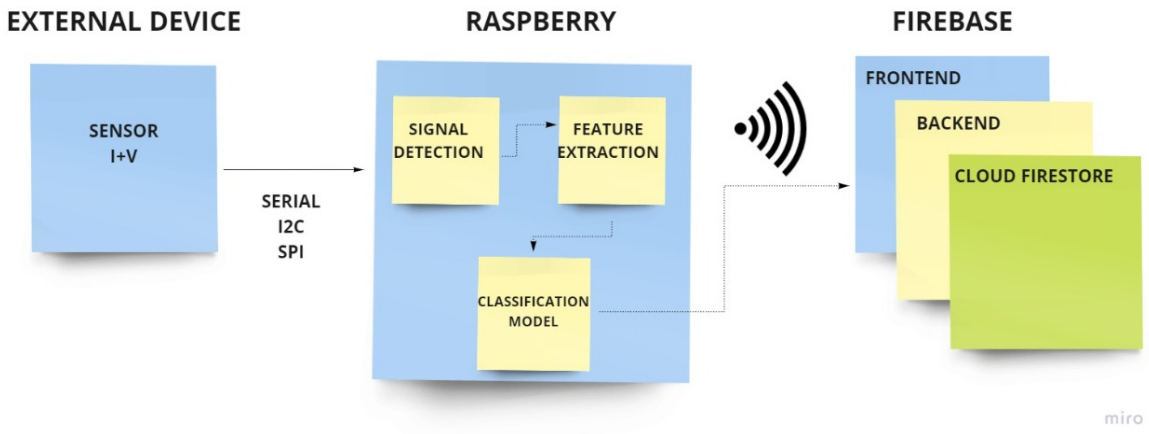
[5] Y. Jimenez. “Automatic Disaggregation of Residential Electrical Consumption with Non-Intrusive Methods.” Tesis. Colombia: Universidad Industrial de Santander, 2018.

fin de determinar la energía que consumen los diferentes electrodomésticos dentro de una vivienda, de forma que se pueda enviar la información a una base de datos y poder ilustrar los resultados en una página web.

Dicho proceso se realizó primero traduciendo a Python los algoritmos diseñados en Matlab, para luego ser implementados en la Raspberry PI 3 B+. Se requería una señal totalmente limpia antes de realizar un procesamiento en la tarjeta de desarrollo para que los códigos puedan identificar de manera eficiente el electrodoméstico que ha sido conectado. Los datos obtenidos por medio de la tarjeta reprogramable son enviados de manera inalámbrica a un aplicativo web que tiene como objetivo mostrar que dispositivo se conectó de acuerdo a cada una de las señales digitalizadas de electrodomésticos que fueron probadas.

La Figura 1 presenta un esquema que resume el dispositivo propuesto en el presente proyecto. Se aclara que el dispositivo externo es simulado al igual que su comunicación con la Raspberry PI 3 B +.

Figura 1. Esquema completo del monitor



Este trabajo está dividido en 6 capítulos, en los cuales se presenta el desarrollo del proyecto y las herramientas usadas. De igual forma, se describe el desarrollo funcional sobre la implementación de los algoritmos, que fueron traducidos para ser migrados y probados en la Raspberry Pi 3 B+, así como la estructura y herramientas empleadas para el almacenamiento en base de datos y visualización de estos en la página web. Para finalizar, se presentan los resultados obtenidos con las pruebas realizadas, conclusiones y recomendaciones para posibles trabajos futuros.

1. OBJETIVOS

Objetivo general

- Desarrollar un dispositivo para estimar el consumo de potencia de los electrodomésticos de una vivienda en forma no intrusiva, implementando algoritmos de desagregación de carga en una Raspberry Pi 3 B +.

Objetivos específicos

1. Implementar algoritmos de desagregación de carga en un dispositivo reprogramable.
2. Estimar el consumo de potencia realizando pruebas con distintas señales digitalizadas de electrodomésticos de una vivienda.
3. Crear aplicativo web para visualizar los datos obtenidos por el monitor de potencia.
4. Transmitir de forma inalámbrica los datos obtenidos por el monitor de potencia para ser visualizados por el usuario.

2. MARCO TEÓRICO

En este capítulo se describe el software implementado para la ejecución de los algoritmos de desagregación de carga diseñados previamente en una tesis doctoral[5]. Además, se describe la base de datos implementada para la visualización de la información obtenida tras la lectura de las señales digitalizadas y procesadas por la Rapsberry Pi 3 B+.

2.1. Monitorización no intrusiva de carga

Es un proceso encargado de identificar cada una de las firmas de carga generadas por electrodomésticos que estén conectados dentro de una casa, sin necesidad de interrumpir el flujo de carga que circula en un hogar [6]. El proceso de identificación funciona analizando los cambios de corriente y tensión que se presentan a la hora de conectar un dispositivo electrónico, reconociendo qué electrodoméstico es y su consumo energético. Este tipo de medición permite realizar una instalación muy sencilla, ya que permite generar un mantenimiento óptimo en comparación con las técnicas tradicionales de monitoreo de carga intrusiva que requieren sub-medición y cableado interior [6].

2.2. Raspberry Pi 3 B+

Entre las diferentes alternativas de tarjetas de desarrollo para la ejecución de procesos de análisis se encuentra la Raspberry Pi 3 B+. Esta posee grandes características como se visualiza en la tabla 1, pensadas para su implementación en procesos

---

[6] G.W. Hart. “Nonintrusive appliance load monitoring”. En: *Proceedings of the IEEE* (1992).

de análisis o IoT. Por ejemplo, posee la fila de pines, alrededor de 40, entre ellos 26 pines GPIO como se puede observar en las figuras 2 y 3, los cuales pueden designarse como puertos de entrada o salida [7] y utilizarse para una amplia gama de propósitos.

Tabla 1. Características Raspberry PI 3 B+.

Características	**Raspberry PI 3 B+**
Procesador	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
Frecuencia de reloj	1,4 GHz
Memoria	1GB LPDDR2 SDRAM
Conectividad inalámbrica	2.4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE
Conectividad de red	Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)
Puertos	GPIO 40 pines HDMI4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Toma auriculares / vídeo compuesto Micro SD Micro USB (alimentación)Power-over-Ethernet (PoE)
Alimentación	5 V a 3A microUSB 5 V a 2.5A microUSB

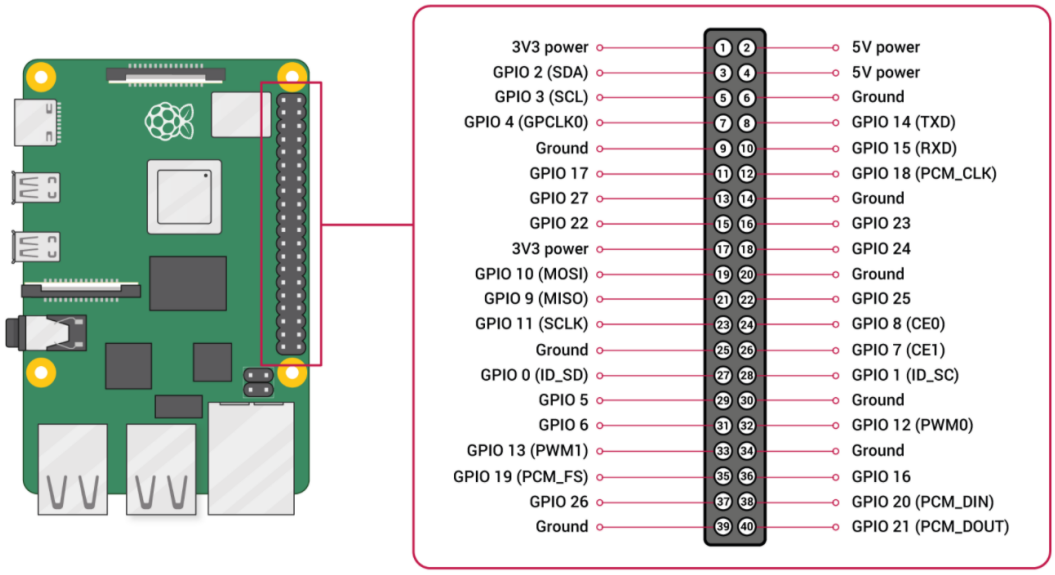
Fuente: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf.

Referente al sistema operativo usado para programar la Raspberry Pi 3 B+, se presentan diversas distribuciones de Linux oficiales, entre esas la seleccionada para el presente proyecto es Raspbian 10 Buster, la cual cuenta con las ultimas actualizaciones de seguridad y compatibilidad con perifericos.

---

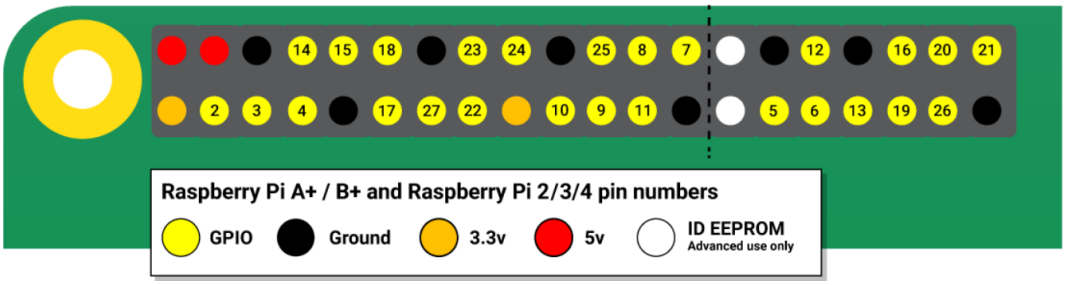
[7] Raspberry-Pi-3-Model-B+. *Documentation*. http://sindominio.net/ash. Consulta: 30 de Enero de 2021.

Figura 2. Pines Raspberry PI 3 B+.



Fuente: https://www.raspberrypi.org/documentation/usage/gpio/README.md

Figura 3. Descripción de pines Raspberry PI 3 B+.



Fuente: https://www.raspberrypi.org/documentation/usage/gpio/README.md

2.3. Lenguajes de Programación en Raspberry Pi

Se pueden utilizar diferentes lenguajes de programación sobre la Raspberry, entre los cuales se encuentran Ruby, C, Python, entre otros. Estos permiten crear programas para diferentes proyectos referentes a IoT, inteligencia artificial, lectura de sensores y procesamiento de datos, y se explican a continuación.

2.3.1. Ruby Es un lenguaje de programación de código abierto, ágil, dinámico y fácil de usar. Está enfocado a la simplicidad debido a la sintaxis que maneja, es un leguaje que va en crecimiento gracias a su framework de desarrollo web Ruby on Rails [8].

2.3.2. C Es un lenguaje de programación de medio nivel desarrollado entre 1969 y 1972, orientado a objetos de propósito general asociado al sistema operativo UNIX. Este ha sido la base para muchos otros lenguajes de programación ya que tiene una gran facilidad para escribir código compacto y sencillo a su misma vez [9]. A pesar de su antigüedad, se sigue implementando en la mayoria de los sistemas operativos desarrollados.

2.3.3. Python Es el lenguaje de programación más utilizado actualmente para el desarrollo y creación de proyectos relacionados a inteligencia artificial, Big Data, desarrollo web, entre otros. Pues permite que los usuarios tengan una experiencia y aprendizaje más dinámico, de manera que sea más fácil de utilizar. Este le permite al usuario trabajar rápidamente e integrar sistemas de manera más efectiva a la hora

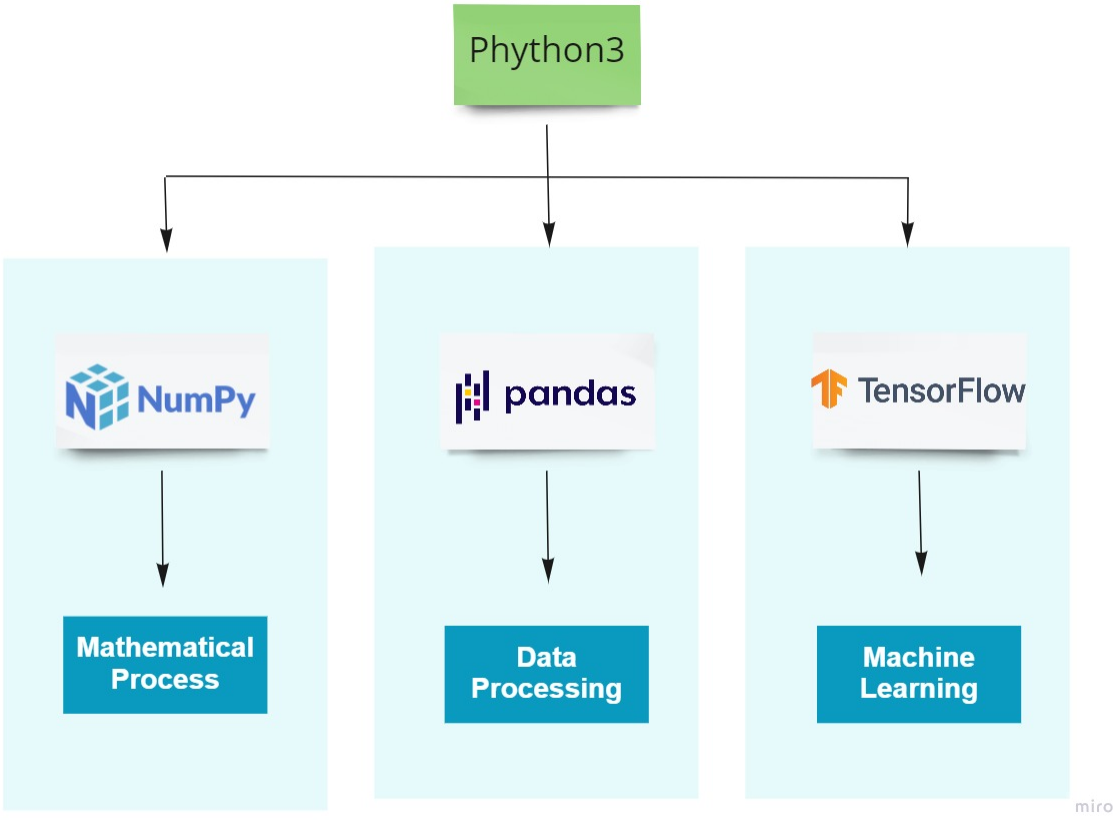
---

[8] Lenguaje.de.Programación.Ruby. *Documentación*. https://www.ruby-lang.org/es/. Consulta: 30 de Enero de 2021.

[9] Brian Kernighan. *The C programming language*. Englewood Cliffs, N.J: Prentice Hall, 1988.

de utilizar la Raspberry Pi 3 B+. Entre sus ventajas se tiene que cuenta con diferentes librerías creadas para procesamiento preciso y rápido de funciones matemáticas avanzadas, como la transformada S utilizada en este proyecto. En la Figura 4, se observan los módulos principales de Python utilizados en el proyecto, los cuales fueron esenciales para el desarrollo del mismo y se explican seguidamente.

Figura 4. Módulos de Python de mayor utilidad en este proyecto



Numpy Es la libreria de computación matemática de Python. A diferencia de la librería nativa de Python para operaciones matemáticas, este módulo garantiza el mejor rendimiento de cada operación utilizando la potencia de los códigos de C y Fortran en un lenguaje de más fácil interpretación y uso. Esto permite que los investigadores futuros puedan entender el código con mayor facilidad, adicionalmente

Numpy cuenta con documentación y comunidad activa, lo cual la hace más robusta frente a bugs.[10]

Pandas Es una librería de procesamiento de datos que utiliza su propio tipo de tablas llamado DataFrame, con el fin de optimizar el procesamiento, búsqueda y modificación de los datos. Gracias a sus librerías se pueden procesar los datos más rápido a diferencia de las listas o arreglos nativos del sistema.[11]

TensorFlow Es una librería de aprendizaje de máquinas, desarrollado por Google y con una comunidad activa en constante desarrollo. Es utilizado y reconocido por diferentes industrias como parte de la implementación de Machine Learning y AI en la actualidad. Es fácilmente escalable y portable gracias a estar desarrollado sobre Python.[12]

2.4. Machine Learning

Machine Learning o traducido al español como *aprendizaje de máquinas*, es un subcampo de la Inteligencia Artificial que permite desarrollar algoritmos de auto aprendizaje. Los cuales utilizando información de datos entregados por el usuario puedan hacer predicciones. Esto presenta un avance significativo en el procesamiento de grandes cantidades de datos, debido a que requerían reglas o modelos derivados de humanos, en cambio, Machine Learning ofrece una alternativa eficiente para capturar la información y mejorar gradualmente el rendimiento de los modelos pre-

---

[10] Numpy. *Numpy Project*. https://numpy.org/about/. Consulta: 30 de Enero de 2021.

[11] The pandas-development team. *pandas-dev/pandas: Pandas*. Ver. latest. Feb. de 2020. DOI: 10.5281/zenodo.3509134.

[12] Martín Abadi y col. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

dictivos. En la actualidad es posible disfrutar desde algoritmos de reconocimiento de voz hasta carros autónomos[13] gracias a estos algoritmos. Existen tres tipos de Machine Learning:

- Aprendizaje supervisado: Éste aprendizaje fue utilizado en el presente proyecto, consiste en entrenar un modelo utilizando *datos de entrenamiento* conociendo la respectiva clasificación de estos.
- Aprendizaje no supervisado: Consiste en analizar datos sin clasificación o estructura conocida para encontrar información relevante sin necesidad de un resultado previo o función de recompensa.
- Aprendizaje reforzado: Este tipo de aprendizaje consiste en desarrollar un sistema (agente) que mejora su rendimiento basado en las interacciones con el ambiente. Estas interacciones dan como resultado una recompensa. Se diferencia del aprendizaje supervisado en su realimentación durante el entrenamiento, ya que no es una referencia fija si no una función de recompensas.

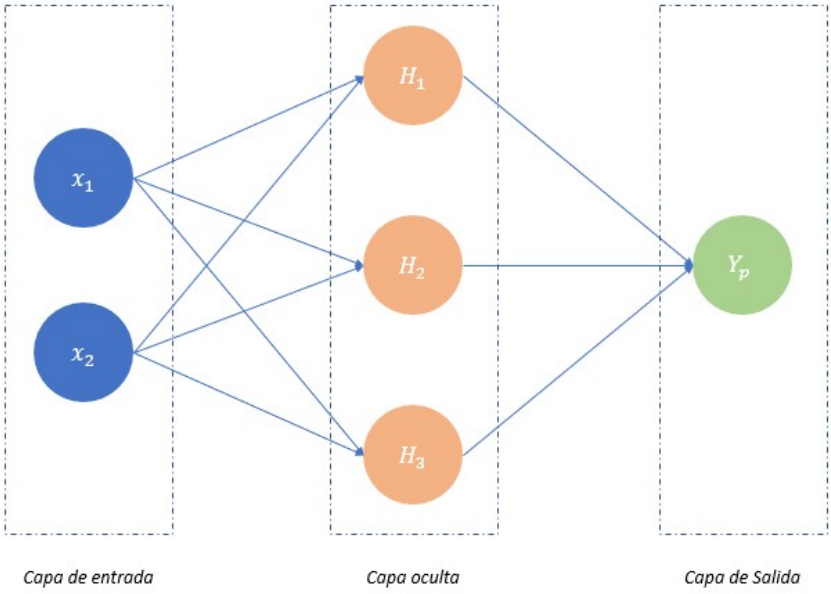
Redes Neuronales Artificiales Son un tipo de algoritmo de Machine Learning que trata de imitar el comportamiento del cerebro humano. En un nivel básico se imita el comportamiento de una neurona. La arquitectura básica de una red neuronal se compone de Capa de Entrada, Capa Oculta y Capa de Salida [14]. En este arreglo todas las entradas están conectadas con cada neurona de la capa oculta y así mismo todas estas están conectadas con la capa de salida como se observa en la Figura 5.

---

[13] Sebastian Raschka. *Python machine learning : unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. Birmingham, UK: Packt Publishing, 2015.

[14] Pramod Singh. *Learn TensorFlow 2.0 Implement Machine Learning and Deep Learning Models with Python*. Berkeley, CA: Apress Imprint Apress, 2020.

Figura 5. Arquitectura básica de una red neuronal



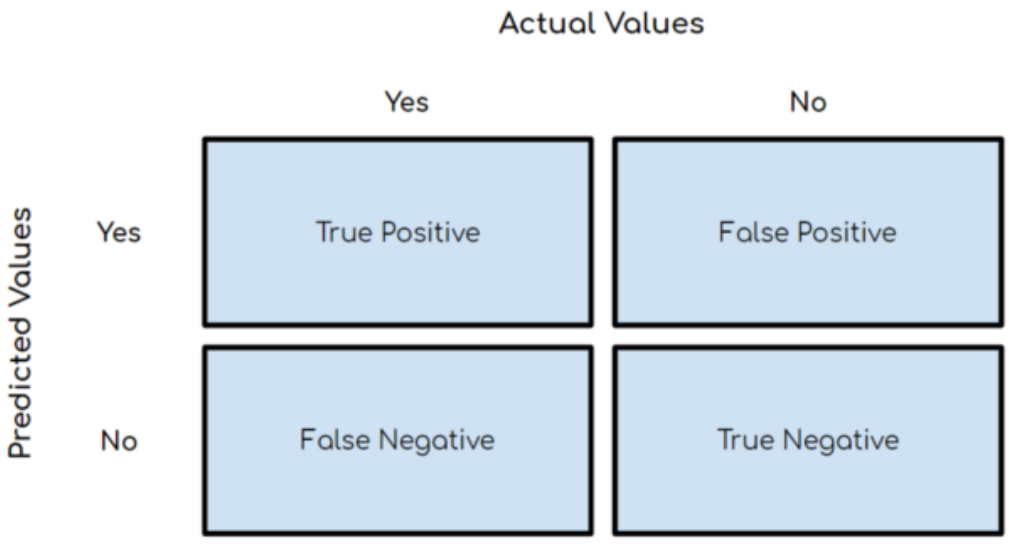
Las Deep Neural Networks (DNN) o traducido al español como *redes neuronales profundas*, son redes neuronales simples con más de una capa oculta. En los modelos de Tensor Flow como el utilizado para este proyecto, se definió el número de neuronas de cada una de las capas ocultas en los parámetros de entrada de la función. Esta elección impacta el tiempo de procesamiento del modelo.

2.4.1. Evaluación de modelos Al finalizar el entrenamiento de un modelo de Machine Learning es necesario caracterizarlo con ciertas métricas predeterminadas que permiten compararlo con otros modelos, e identificar cual es el mejor modelo para utilizar. Se presentan a continuación algunas de las métricas principales.

Matriz de confusión La primera métrica utilizada en los modelos de clasificación y presentada en el proyecto actual es la matriz de confusión, también conocida como matriz de error. Consiste en un resumen presentado en tablas donde se puede ver el número de predicciones correctas e incorrectas divididas en las diferentes

categorías [15]. Se observa un ejemplo en la Figura 6.

Figura 6. Estructura de una matriz de confusión 2x2



Fuente: https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d

Del ejemplo se pueden obtener los siguientes términos clave:

- Positive (P): La observación fue positiva (Para la figura, *Yes*)
- Negative (N): La observación fue negativa (Para la figura, *No*)
- True Positivo (TP): Positivo verdadero, es la salida cuando el modelo predice correctamente la clase positiva

[15] Terence Shin. *Understanding the Confusion Matrix and How to Implement it in Python*. https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d. Consulta: 30 de Enero de 2021.

- True Negative (TN): Negativo verdadero, es la salida cuando el modelo predice correctamente la clase negativa
- False Positive (FP): Se conoce también como error tipo 1, el falso positivo indica que el modelo predijo erróneamente una clase positiva cuando realmente era negativa
- False Negative (FN): Se conoce también como error tipo 2, el falso negativo indica que el modelo predijo erróneamente una clase negativa cuando realmente era positiva

Exactitud Se define como la cantidad de predicciones que el modelo clasifico correctamente, como se observa en la ecuación 1.

$$Exactitud = \frac{Nmero\ de\ predicciones\ correctas}{Total\ de\ predicciones} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precisión También conocido como valor predictivo positivo, es la proporción de instancias relevantes contra las instancias entregadas, quiere decir, el número de identificaciones positivas que fueron correctas[15], como se observa en la ecuación 2.

$$Precisin = \frac{TP}{TP + FP} \quad (2)$$

2.5. Jupyter

El proyecto de código fuente abierto, sin ánimo de lucro, ha evolucionado para dar soporte a la ciencia de datos de forma interactiva y a la computación científica en diferentes lenguajes de programación. Entre sus principales herramientas presenta los Jupyter Notebooks. Estos son aplicaciones web de código abierto que permiten crear y compartir documentos, los cuales contienen códigos en vivo, ecuaciones,

gráficos y texto narrativo. Entre sus usos más comunes se encuentra la limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje de máquinas entre otros.

Para el presente proyecto se implementaron en estos documentos web las traducciones a Python de todos los algoritmos de Matlab desarrollados en un trabajo previo [5], permitiendo así tener un registro detallado de los códigos, sus pruebas y explicación, así como la facilidad de compartir estos documentos creando un nuevo entorno de aprendizaje y entrega de resultados [16].

2.6. Angular CLI

Angular es un framework de código abierto desarrollado por Google, especializado para la creación de aplicaciones web de una sola página utilizando HTML y TypeScript. Además, es el más utilizado a nivel mundial por su fácil escalabilidad y optimización en el desarrollo de componentes, ya que separa completamente el front-end del back-end [17]. Esto permite una mayor facilidad en la construcción de una página web previniendo escribir código repetitivo y de mantener todo de forma más ordenada, ya que implementa un patrón de arquitectura de software llamado modelo vista controlador. De igual forma, permite la instalación de herramientas de código abierto que permiten utilizar plantillas personalizadas para la creación de tablas, gráficas, barras y menús de navegación, entre otros [18].

---

[16] Jupyter. *Jupyter Project*. https://jupyter.org/about. Consulta: 30 de Enero de 2021.

[17] Platzi. *Frontend con Angular*. https://platzi.com/desarrollo-angular/. Consulta: 30 de Enero de 2021.

[18] Angular. *Introduction to the Angular Docs*. https://angular.io/docs. Consulta: 30 de Enero de 2021.

2.7. Firebase

Es una plataforma en la nube creada por Google que permite llevar a cabo la implementación de aplicaciones web y móvil de forma más sencilla. Brinda diferentes servicios de analítica, autenticación, bases de datos en tiempo real, funciones en la nube, almacenamiento en la nube, almacenamiento de imágenes y páginas web [19].

2.7.1. Firebase Analytics (Analítica) Este servicio brinda información estadística y realiza una medición referente a la interacción que tienen los usuarios con la aplicación móvil o web [20].

2.7.2. Firebase Authentication (Autenticación) Esta herramienta brinda servicios de backend para realizar validaciones de los usuarios que se encuentren registrados en la app móvil o web. Esta función puede vincularse con diferentes plataformas de desarrollo como Android, IOS y web [21].

2.7.3. Firebase Realtime Database (Base de datos en tiempo real) Permite almacenar datos en la nube y sincronizarlos en tiempo real. Estos deben ser guardados en formato JSON, lo cual es de mucha utilidad ya que no se necesita código SQL para realizar las consultas en la base de datos [22].

---

[19] Firebase. *Guías de Firebase*. url https://firebase.google.com/docs/guides?hl=es. Consulta: 30 de Enero de 2021.

[20] Firebase.Analytics. *Google Analytics*. https://firebase.google.com/docs/analytics?hl=es. Consulta: 30 de Enero de 2021.

[21] Firebase.Authentication. *Authentication*. https://firebase.google.com/docs/auth?hl=es. Consulta: 30 de Enero de 2021.

[22] Firebase. *Firebase Realtime*. https://firebase.google.com/docs/database?hl=es. Consulta: 30 de Enero de 2021.

2.7.4. Cloud Functions (Funciones en la nube) Es un framework que permite ejecutar sin necesidad de servidores las funciones desarrolladas en TypeScript y JavaScript, almacenándolas en la nube y ejecutando de forma automática el código de backend a peticiones HTTPS. Además, protege la privacidad y seguridad de la lógica implementada [23].

2.7.5. Cloud Firestore (Almacenamiento de datos en la nube) Es una base de datos no relacional que no necesita de la implementación de código SQL para realizar las consultas de sus datos, ya que implementa el uso de colecciones para almacenarlos. Además de poder sincronizarlos y actualizarlos en tiempo real, este servicio ofrece una ininterrupción con otras herramientas de Firebase [24].

2.7.6. Cloud Storage (Almacenamiento en la nube) Es una herramienta que permite cargar o descargar imágenes, audio, video y otros tipos de contenido generado por el usuario sin importar la calidad de la red [25].

2.7.7. Firebase Hosting (Almacenamiento de páginas web) Es un servidor para alojar las aplicaciones móviles o web desarrolladas y entregar contenido dinámico y estático. Además, brinda seguridad SSL y HTTP2 sin ningún costo brinda una mayor facilidad de administración de versiones [26].

---

[23] Firebase.Cloud. *Cloud Functions*. https://firebase.google.com/docs/functions?hl=es. Consulta: 30 de Enero de 2021.

[24] Firebase.Firestore. *Cloud Firestore*. https://firebase.google.com/docs/firestore?hl=es. Consulta: 30 de Enero de 2021.

[25] Firebase.Storage. *Cloud Storage*. https://firebase.google.com/docs/storage?hl=es. Consulta: 30 de Enero de 2021.

[26] Firebase.Hosting. *Firebase Hosting*. https://firebase.google.com/docs/hosting?hl=es. Consulta: 30 de Enero de 2021.

3. DESARROLLO FUNCIONAL

Con lo descrito anteriormente se realiza el diseño web y traducción de los algoritmos de desagregación de carga para la implementación del monitor no intrusivo.

3.1. Implementación de Algoritmos en Raspberry Pi 3 B+

La base funcional de este proyecto se encuentra en los algoritmos de desagregación de carga desarrollados en un trabajo previo. Era necesario traducirlos revisando cada variable y cada tipo de dato para asegurar que el procesamiento de los datos se realizara como en el algoritmo original. Para realizar esta tarea se utilizó el lenguaje de programación Python debido a que éste ofrecía portabilidad, escalabilidad y flexibilidad en el código y en la implementación. Estas utilidades del lenguaje resultaron muy útiles para poder trasladar el código entre diferentes ambientes de prueba y cambiar rápidamente de procesar un grupo pequeño de datos a un set completo. De igual forma, el desarrollo del código cumple con las recomendaciones principales de programación sobre comentarios y documentación, buscando garantizar que el lector y los investigadores futuros puedan entender con facilidad los códigos y modificarlos a su gusto. Esta tarea se desarrolló de la siguiente forma:

3.1.1. Diseño de algoritmos en ambiente de pruebas Para diseñar los algoritmos y no depender inicialmente del ambiente de la Raspberry Pi 3 B+, el cual presentaba dificultades de visualización debido a que no es su propósito, se utilizó un ambiente remoto de pruebas, el cual se configuró con las librerías nombradas previamente y considerando que sería implementado posteriormente en una tarjeta de desarrollo. Esto permitió trabajar y simular el algoritmo con una mejor visualización. Las ventajas de Python se vieron en todos los pasos de desarrollo del algoritmo y

también se implementaron pruebas sobre Jupyter Notebook. Esta herramienta permitió que el código se desarrollara de una forma más sencilla de entender gracias a los notebooks.

3.1.2. Selección de datos de prueba Para realizar las pruebas con información similar a la esperada en la entrada del dispositivo se propuso utilizar un dataset público. Entre estos se encontró el presentado por PLAID [27], el cual presenta los datos a utilizar en diversos archivos separados por comas (CSV). Los datos consisten en muestras de electrodomésticos en el momento de conexión, donde usualmente se registra y observa el transitorio y estado estable de la señal.

3.1.3. Ejecución en ambiente de pruebas Se validó el correcto funcionamiento de los algoritmos comparando los resultados obtenidos en Matlab con el algoritmo original contra los resultados vistos en Jupyter con el nuevo algoritmo. Adicionalmente, se eligieron los electrodomésticos del set de pruebas que contenían el mayor número de datos, estos fueron Ventilador, Secador de Cabello y Portátil. Esto permitió que el entrenamiento del Clasificador se realizara con el mayor número de datos posibles obteniendo mejores resultados en su precisión. Cabe aclarar que, al tener un clasificador entrenado con tres dispositivos, cualquier dispositivo no registrado que sea procesado se clasificara con un porcentaje de precisión en uno de los dispositivos ya registrados.

3.1.4. Migración y pruebas en Raspberry Pi 3 B+ Al finalizar con el algoritmo y comprobar que sus resultados coincidían con lo esperado, el siguiente paso consis-

---

[27] Jingkun Gao y col. “PLAID: A Public Dataset of High-Resoultion Electrical Appliance Measurements for Load Identification Research: Demo Abstract”. En: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. BuildSys ’14. Memphis, Tennessee: Association for Computing Machinery, 2014, 198–199. DOI: 10.1145/2674061.2675032.

tió en compilar esta información en el sistema destino, la Raspberry Pi 3 B+. Para esto, se instaló el sistema operativo Raspbian 10 en su compilación más reciente, la cual cuenta con una versión de Kernel 5.4.83 y contiene todos los programas básicos de un sistema operativo basado en linux.
Los pasos de la migración de códigos y ejecución fueron:

1. Verificación e importación de las librerías a ser utilizadas, las cuales garantizan que el código pueda ser leído y compilado. Para esto, se instalaron todas las librerías necesarias que faltaban en el sistema, lo cual en linux se realiza con facilidad desde la consola de comandos.

2. Pruebas funcionales de detección de eventos y generación de características de las señales, comparación con el ambiente de prueba, ubicación de los archivos y regularización del ambiente de las Raspberry PI 3 B+.

3. Después de garantizar las características como se esperaban, consistió en compilar los algoritmos de clasificación, para garantizar que el funcionamiento de estos fuera como en el ambiente de pruebas.

4. Luego de tener el clasificador funcional, el siguiente paso fue probar la conectividad con la base de datos, ésta se realiza con una librería de Firebase para Python, la cual carga la información a la base de datos directamente.

5. El último paso se baso en tomar una señal de prueba y agregarla al flujo presentado anteriormente y comprobar que sea procesada, identificada y enviada a la página web.

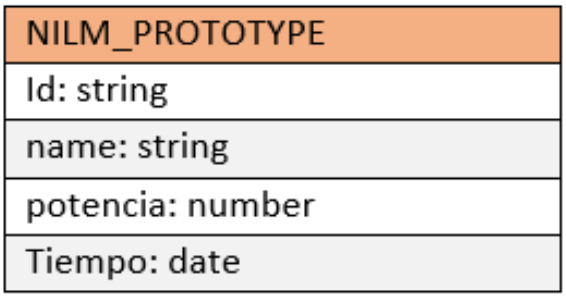
3.2. Almacenamiento en base de datos

Uno de los aspectos más importantes del presente proyecto es el acceso a la información obtenida por la Raspberry PI 3 B+, de manera que cualquier usuario con

las credenciales especiales pueda consultar los datos obtenidos por la tarjeta de desarrollo. Para esto, se utilizó Cloud Firestore para crear la base de datos, ya que esta plataforma brinda la creación de una base de datos no relacional en forma de colecciones, lo cual indica que no es necesario emplear código SQL facilitando el desarrollo de servicios. Además de que se adapta perfectamente para el desarrollo de muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento [28].

Como primera parte, se diseñó la estructura de la base de datos, determinando qué información es enviada desde la Rapsberry PI 3 B+ y se realizó un modelo para identificar la estructura que deben llevar las tablas que contemplan la base de datos. En la Figura 7 se ilustra el modelo de base de datos:

Figura 7. Modelo de base de datos.

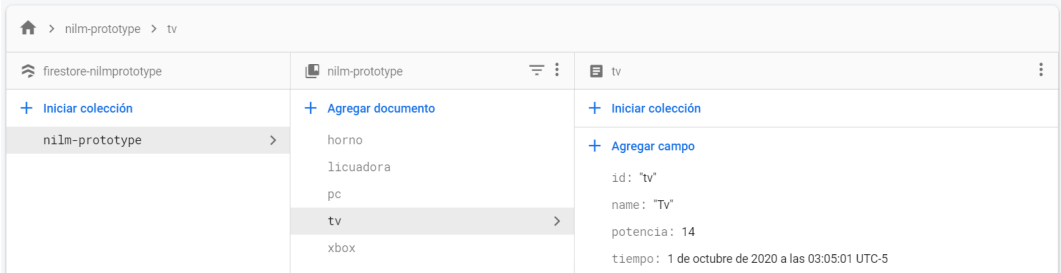


Determinado el modelo de base de datos, se ingresa a la página de Firebase y se inicia sesión con una cuenta de Google, y se selecciona el ítem Cloud Firestore. Allí, se determina el nombre de la base de datos y se crea el objeto asociado a un electrodoméstico, cabe aclarar que no es necesario tener registrados en la base datos los electrodomésticos a los cuales se les desea identificar su consumo, ya que una vez son identificados por la Raspberry PI 3 B+, este se crea automáticamente

[28] Amazon. *¿Qué es NoSQL?* https://aws.amazon.com/es/nosql/. Consulta: 30 de Enero de 2021.

cuando se conecta la Raspberry PI 3 B+ como se observa en la Figura 8.

Figura 8. Estructura en colecciones (Base de datos).



3.2.1. Backend Para implementar las peticiones GET realizadas a la base de datos y visualizar la información que se encuentra allí registrada, se crea una sección aparte de la Web llamada Backend. En este se realiza toda la lógica referente a las consultas en base de datos. Los códigos utilizados para realizar dichas peticiones se pueden visualizar en la Figura 9, estas peticiones pueden ser guardadas en Firebase con la herramienta " Cloud Functions", la cual permite almacenar el código de backend para realizar las peticiones HTTPS.

3.2.2. Página Web Para el desarrollo de la página web se decidió implementar el framework Angular ya que permite tener más organizada la construcción de la aplicación web, separando la estructura CSS, HTML y de JavaScript para cada componente y permitir ser escalable para futuras modificaciones.

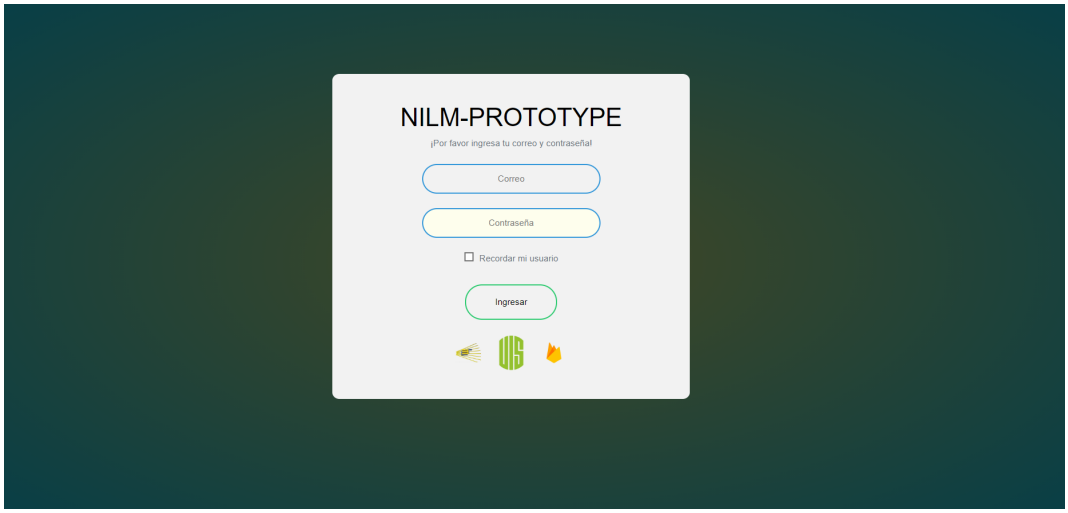
La página desarrollada en este proyecto consta de una ventana de Login, que se puede observar en la figura 10, en la cual se podrá ingresar con un usuario y contraseña ya previamente registrados. Para ser más amigable con el usuario, se agregaron algunas alertas de notificaciones referentes a si la contraseña está mal escrita o

Figura 9. Códigos peticiones get (Backend).

```
const app = express();
app.use( cors ({ origin: true }) );

app.get('/nilm-prototype',async (req,res) => {

  const nimlRef = db.collection('nilm-prototype');
  const docsSnap = await nimlRef.get();
  const elementos = docsSnap.docs.map( doc => doc.data() );

  res.json( elementos );

});
```

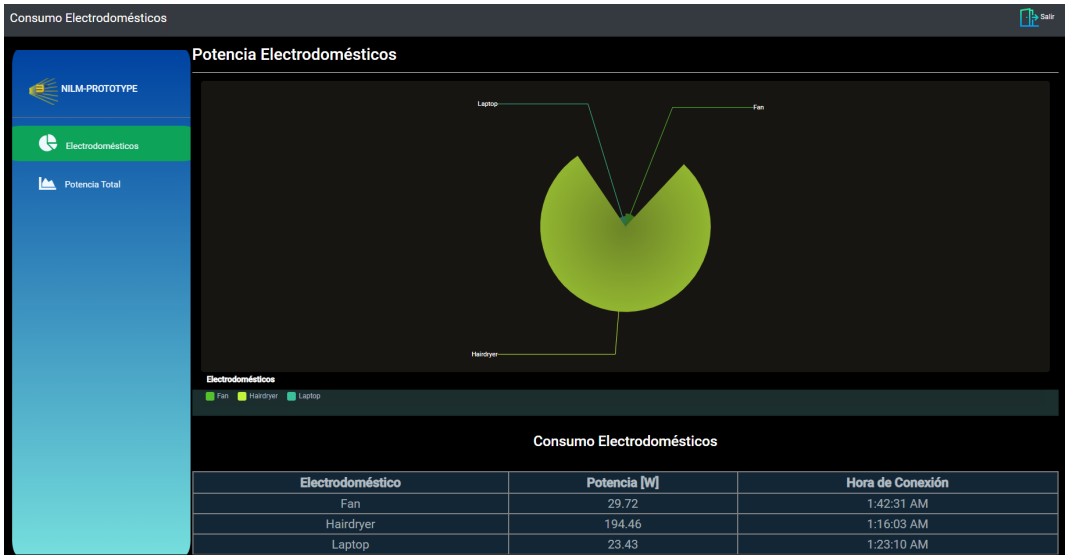
si el correo que se ha ingresado no se encuentra registrado para poder dar acceso a la aplicación, así como de poder ir directamente a las páginas de la Universidad Industrial de Santander, Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones y de Firebase. Además, se incorporó un checkbox que permite recordar el correo que se ha ingresado con anterioridad para facilitar el ingreso al usuario.

Figura 10. Login.



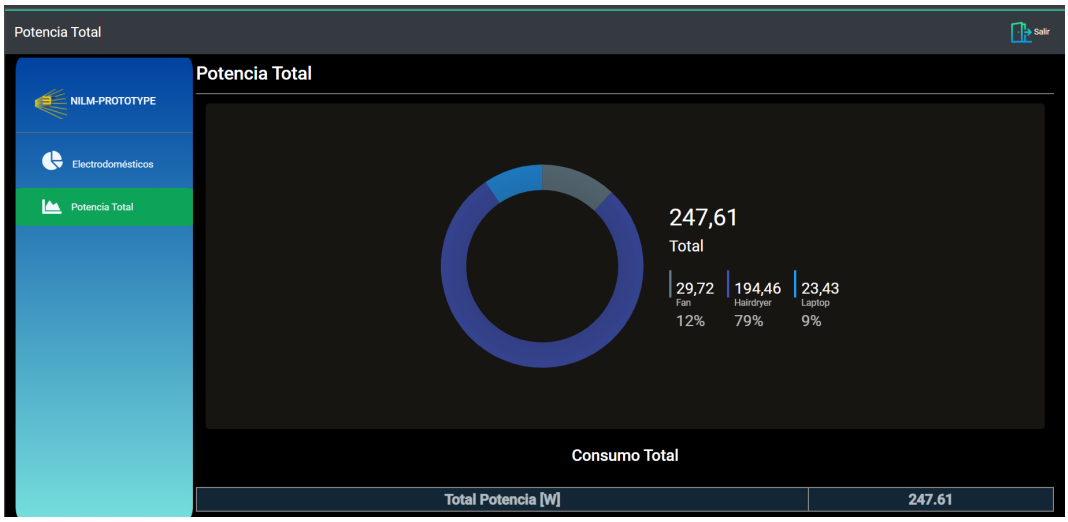
Una vez se ha ingresado, el usuario tiene dos opciones para visualizar la información registrada en base de datos Electrodomésticos y Total Potencia. El ítem Electrodomésticos, como se observa en la figura 11, contiene una gráfica de pastel indicando la información del consumo de cada uno de los electrodomésticos que se encuentran registrados en la base de datos, así como una tabla que indica el consumo y hora de conexión de cada uno de ellos. La sección Total Potencia, como se ilustra en la figura 12, muestra una gráfica con la representación en porcentaje del consumo de cada electrodoméstico, así como el total.

Figura 11. Ventana de visualización (Electrodomésticos conectados).



3.2.3. Hosting (Alojamiento y dominio) Existen diferentes formas de visualizar el código realizado en la página web, uno de estos, es implementando un servidor local utilizando diferentes herramientas que brindan conexiones locales, pero esta forma no permite que los usuarios puedan acceder a la página desde cualquier lugar. Por tal razón es indispensable utilizar un alojamiento en la nube que brinde el acceso desde cualquier navegador. Para ello se debe buscar el más adecuado para

Figura 12. Ventana de visualización (Total consumo).



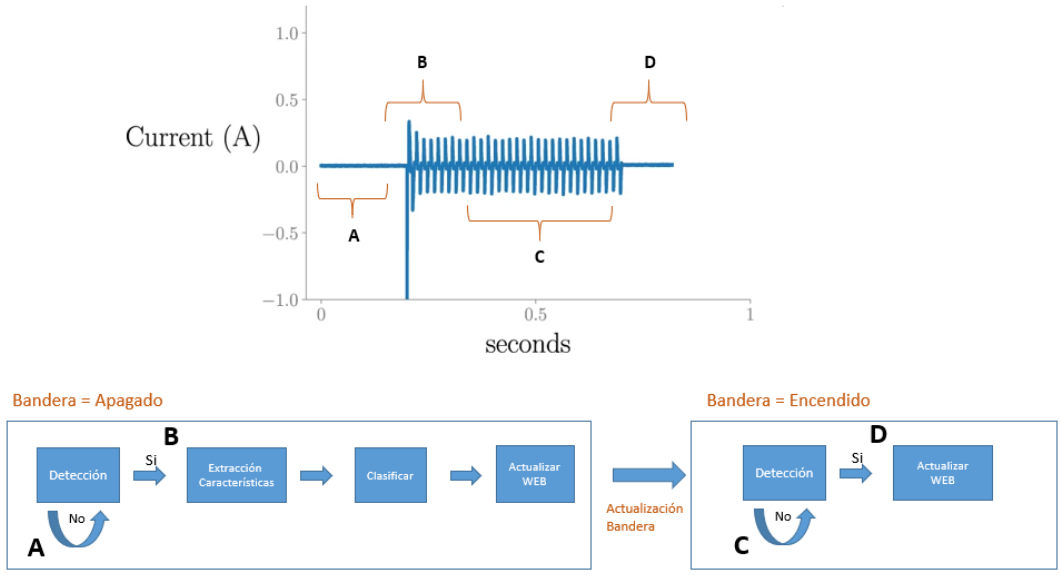
este fin. En el caso de este proyecto, fue usada la herramienta de almacenamiento gratuito proporcionada por Firebase llamada Hosting, ya que esta, nos brinda la opción de llevar de forma escalable el desarrollo de la página, además, de evitar la molesta publicidad que aparece en otros sitios de hosting gratuitos. El dominio otorgado a la página web depende del nombre que se le haya adjudicado al proyecto en Firebase, para este caso el dominio otorgado fue https://firestore-nilmprototype.web.app

4. Mejoras Funcionales

4.1. Detección de apagado

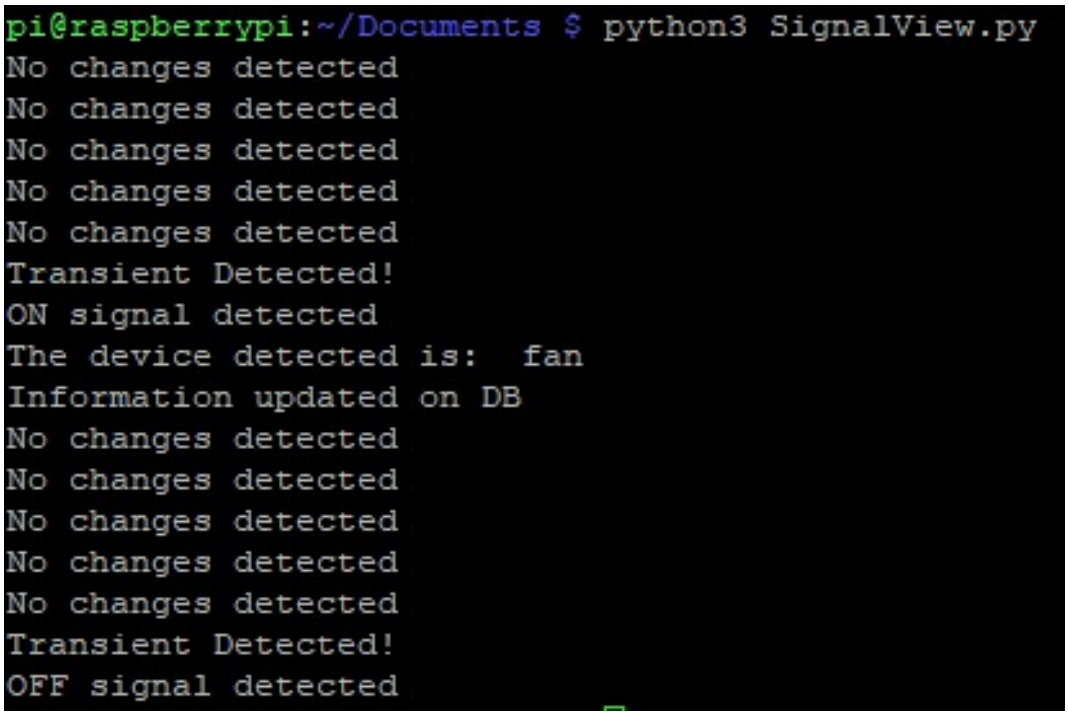
En la figura 13 se observa el comportamiento del algoritmo desarrollado como parte de las mejoras requeridas.

Figura 13. Flujo Algoritmo de apagado



El segmento A hace referencia a la señal apagada, el segmento B muestra la señal en el encendido, la cual tiene transitorio y una parte en estado estable, esta es la misma señal utilizada en las pruebas anteriores. El segmento C hace comprende la señal en estado estable mientras no se realicen cambios en la conexión, representa la duración del electrodoméstico encendido. Finalmente, el segmento D presenta la señal apagada, después de la cual se realiza el cálculo de la potencia total, actualizando la información en la base de datos.

Figura 14. Ejecución del algoritmo en la Raspberry



4.2. Base de datos

De acuerdo a lo anterior, fue necesario ajustar el modelo de base de datos para poder almacenar el tiempo de desconexión y calcular directamente en frontend el consumo que tuvo el electrodoméstico detectado, este ajuste se puede visualizar en la figura 16.

De esta forma la estructura en colecciones de la base de datos se modificó como se observa en la figura 16.

En cuanto a la parte de frontend se realiza una modificación en la pantalla Electrodomésticos agregando en la tabla de datos los campos Estado, Hora de Desconexión, Duración y Total Consumo. Para calcular el consumo del electrodoméstico es ne-

Figura 15. Modelo de base de datos.

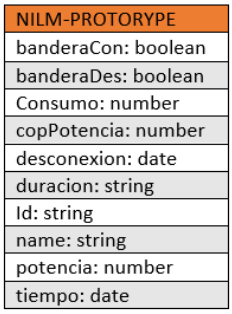
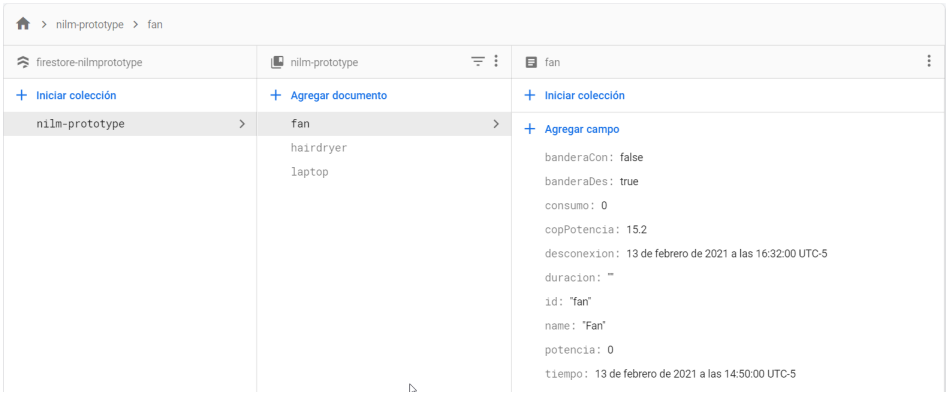


Figura 16. Estructura en colecciones (Base de datos).



cesario determinar el tiempo que este duro conectado, para ello se implementó el código que se visualiza en la figura 17.

Luego de obtenido el tiempo en que duro conectado el electrodoméstico se ejecuta la fórmula:

$$consumo = duracin * potenciaElectrodomstico. \tag{3}$$

Esta fórmula se implementa en la parte de frontend con el algoritmo que se ilustra en la figura 18.

Una vez ejecutados los algoritmos visualizados en las figuras 17 y 18, y tomando

Figura 17. Método cálculo de duración de conexión.

```
dur(item){

  for (let index = 0; index < item.length; index++) {

    var fe = new Date();
    var unixTimeCon = item[index].tiempo.seconds;
    var dateCon = new Date(unixTimeCon*1000);

    var unixTimeDes = item[index].desconexion.seconds;
    var dateDes = new Date(unixTimeDes*1000);

    var hora = dateDes.getHours() - dateCon.getHours();

    var min = dateDes.getMinutes() - dateCon.getMinutes();

    var seg = dateDes.getSeconds() - dateCon.getSeconds();

    item[index].duracion = new Date(fe.getFullYear(),fe.getMonth(),fe.getDate(),hora,min,seg);

  }
}
```

Figura 18. Método cálculo de consumo.

```
con(item){
  for (let index = 0; index < item.length; index++) {

    var fe = new Date();
    var unixTimeCon = item[index].tiempo.seconds;
    var dateCon = new Date(unixTimeCon*1000);

    var unixTimeDes = item[index].desconexion.seconds;
    var dateDes = new Date(unixTimeDes*1000);

    var hora = dateDes.getHours() - dateCon.getHours();

    var min = dateDes.getMinutes() - dateCon.getMinutes();

    var seg = dateDes.getSeconds() - dateCon.getSeconds();

    var time = new Date(fe.getFullYear(),fe.getMonth(),fe.getDate(),hora,min,seg);

    var total = time.getHours() + (time.getMinutes()/60) + (time.getSeconds()/3600);

    item[index].consumo = total*item[index].copPotencia;

  }
}
```

los campos nuevos agregados en base de datos, se puede ver en la página de Electrodomésticos y Potencia Total el dispositivo que se encuentra conectado, luego de un tiempo este se visualiza desconectado indicando el tiempo en que este duró encendido y su consumo, esto se puede evidenciar en las figuras 19, 20, 21, 22.

Figura 19. Ajustes Pagina Electrodomésticos (Electrodoméstico desconectado).

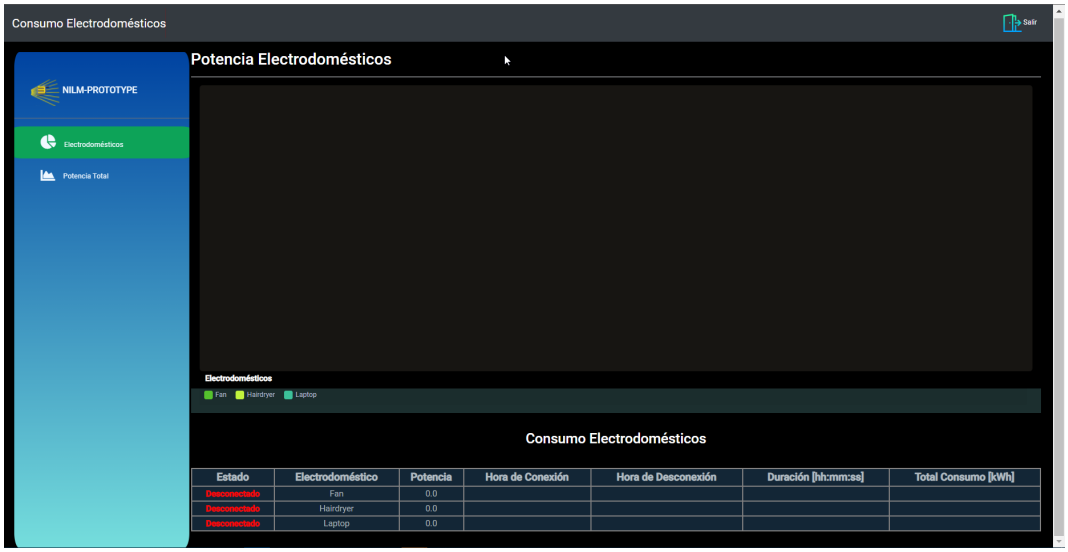


Figura 20. Ajustes Pagina Electrodomésticos (Electrodoméstico conectado).

Figura 21. Ajustes Pagina Potencia Total (Electrodoméstico conectado).

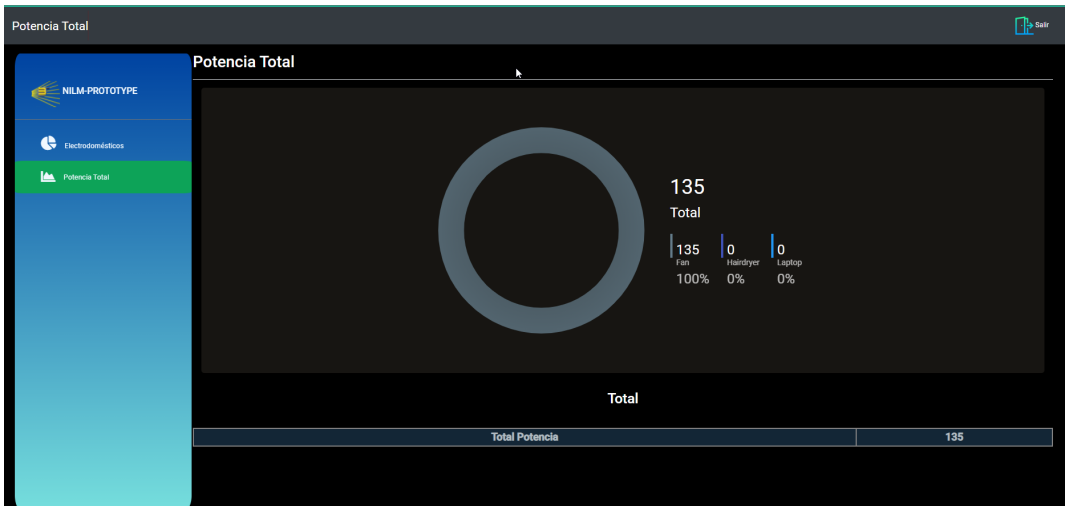
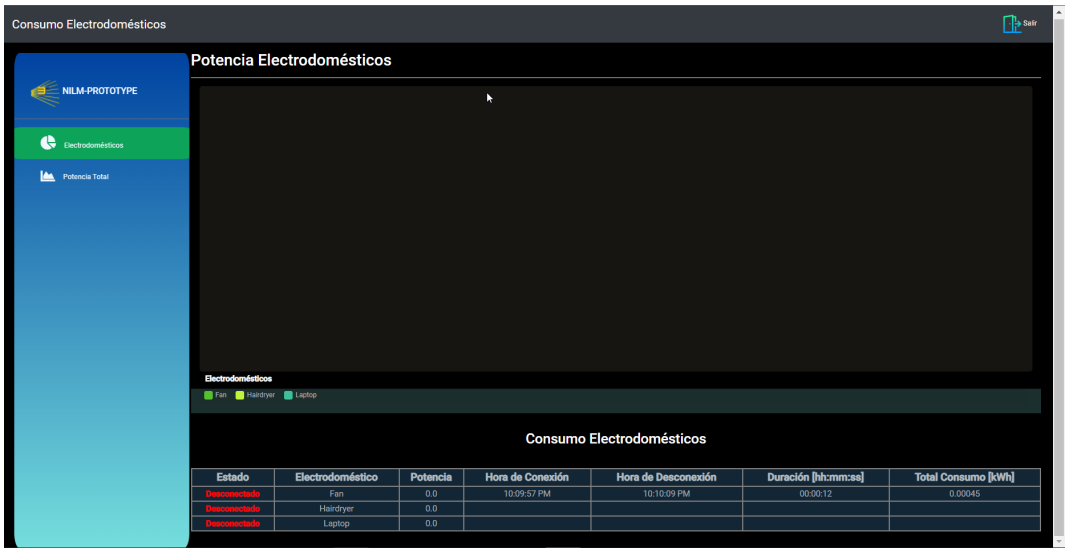


Figura 22. Ajustes Pagina Electrodomésticos (Electrodoméstico desconectado).



5. RESULTADOS

El algoritmo de desagregación de carga se pudo implementar exitosamente en la Raspberry PI 3 B+. Esto se logró gracias a la migración y adecuación de los algoritmos en la tarjeta de desarrollo obteniendo los resultados que se observan en las Figuras 23, 24 y 25 .

Figura 23. Ejecución del algoritmo en la Raspberry PI 3 B+ para el ventilador

```
pi@raspberrypi:~/Documents $ python3 TestCode.py
Time taken for loading up the data 0.1900 seconds
Start Ranges extraction
Rangs extracted
Time taken for extracting ranges 0.1930 seconds
Start features extraction
The length of transient is:  51500
/home/pi/Documents/NILMfunctions.py:299: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuple
s-or ndarrays with different lengths or shapes) is deprecated. If you meant to d
o this, you must specify 'dtype=object' when creating the ndarray
  AreaVI
/home/pi/Documents/NILMfunctions.py:211: ComplexWarning: Casting complex values
to real discards the imaginary part
  S[indexf[0],:]=np.fft.ifft(temp)
/home/pi/Documents/NILMfunctions.py:197: ComplexWarning: Casting complex values
to real discards the imaginary part
  S[0,:]=x.mean(0)    #Not used...
Features extracted
Time taken for feature extraction 155.3963 seconds
(1, 42)
(1, 280)
(1, 322)
(1, 63)
(1, 21)
(1, 84)
(1, 406)
(1, 42)
(1, 448)
448
Start Model import
Model imported Succesfully
Time taken for importing model 139.2692 seconds
Prediction start
Prediction Finish
Time taken for making the prediction 4.5110 seconds
The detected device is :  fan
Start sync to Firebase DB
Sync to DB finished
Time taken for uploading to DB 3.2426 seconds
```

Figura 24. Ejecución del algoritmo en la Raspberry PI 3 B+ para el secador de pelo

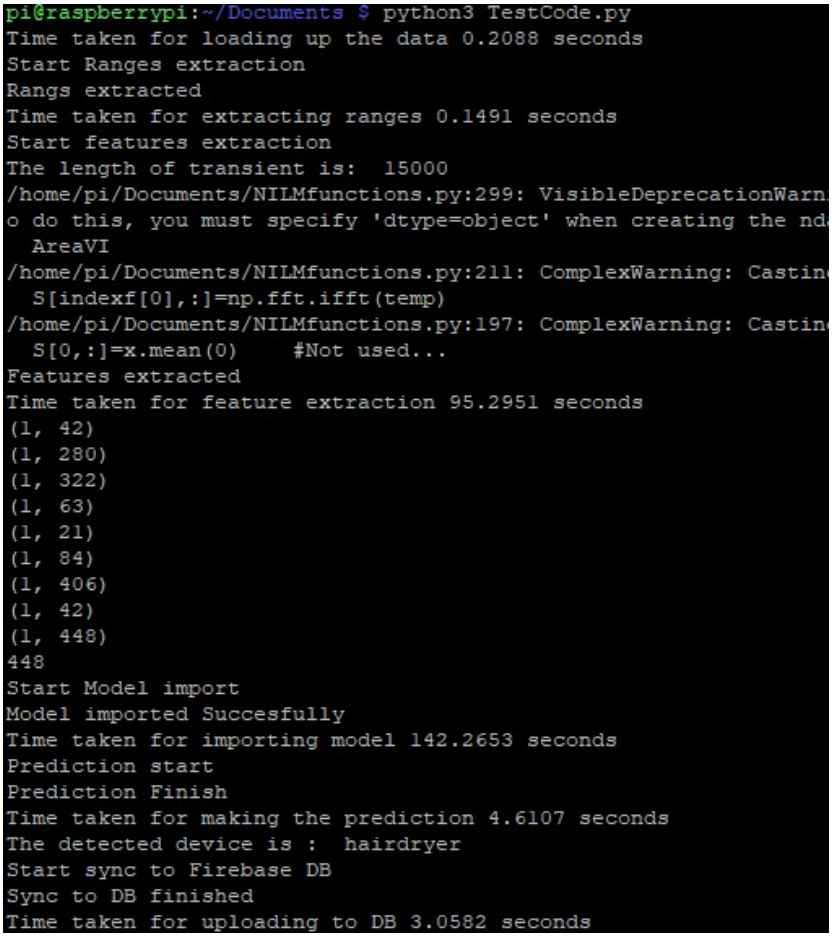
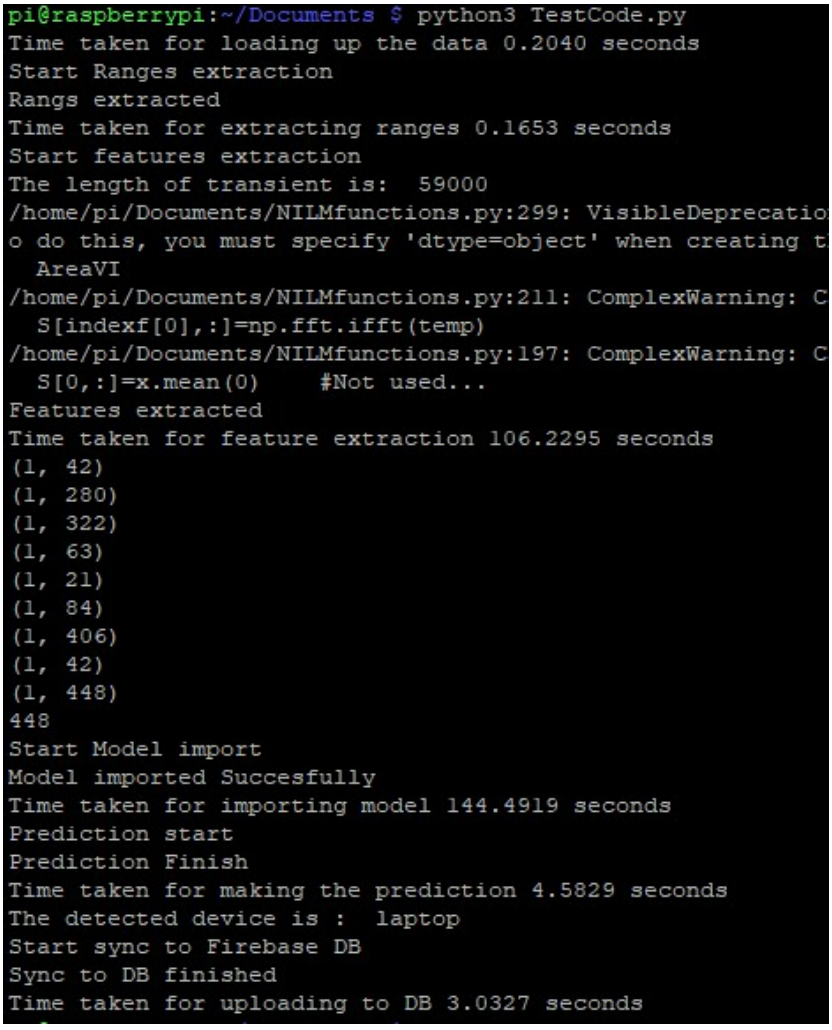


Figura 25. Ejecución del algoritmo en la Raspberry PI 3 B+ para el computador portátil



5.1. Tiempos de ejecución

En la tabla 2 se observan los tiempos de ejecución de 10 electrodomésticos, separados por tipo de algoritmo, se puede dar un tiempo promedio de ejecución de 259.39 segundos.

Tabla 2. Tiempos de ejecucíon.

Elementos	**Tiempos de ejecución de los algoritmos**					
	Rangos [s]	**Desagregación [s]**	**Importar modelo [s]**	**Predicción [s]**	**Envío de datos [s]**	**Tiempo total [s]**
Ventilador 1	0.19	155.39	139.26	4.51	3.24	302.59
Ventilador 2	0.30	108.54	138.46	4.71	3.15	255.16
Ventilador 3	0.15	108.71	138.0	4.6	3.03	254.49
Secador de cabello 1	0.15	102	139.70	4.70	3.16	249.71
Secador de cabello 2	0.33	110.88	136.23	4.64	3.03	255.11
Secador de cabello 3	0.14	95.29	142.26	4.61	3.05	245.35
Computador portátil 1	0.43	99.63	141.47	4.51	3.06	249.1
Computador portátil 2	0.15	113.28	137.59	4.56	3.10	258.68
Computador portátil 3	0.16	106.22	144.49	4.58	3.03	258.48
Computador portátil 4	0.20	117.24	140.06	4.56	3.17	265.23
Promedios	0.22	111.718	139.752	4.598	3.102	259.39

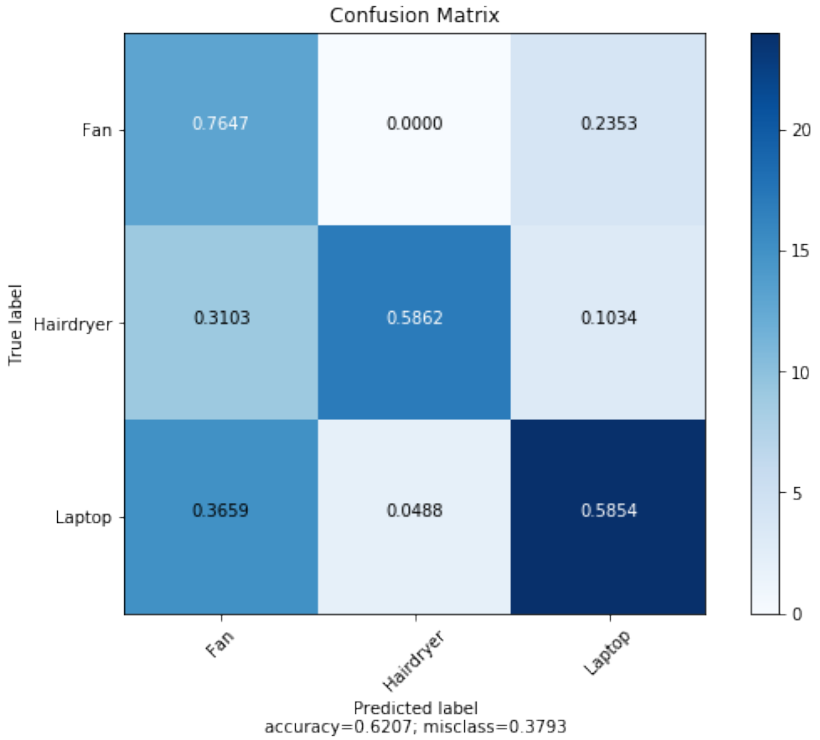
Como se observa en la tabla 2 el tiempo promedio de carga del modelo del clasificador es de: 139.75 segundos y el tiempo promedio de predicción es de 4.59 segundos.

5.2. Modelo de clasificación

A continuación se presentan resultados sobre el modelo:

- Se realizaron pruebas iniciales con un clasificador lineal obteniendo como resultado una precisión de 52.54 %
- La red neuronal profunda entrenada para este proyecto se creó con dos Capas Ocultas de 30 y 10 nodos respectivamente.
- El comportamiento del modelo se observa en la Matriz de Confusión de la Figura 26. El modelo de clasificación basado en Redes Neuronales Profundas

Figura 26. Matriz de confusión para la red neuronal profunda



tuvo un resultado de 0.62 poco satisfactorio para el numero de clases que tenía (3 clases de electrodomésticos), por lo tanto, se concluye que en su configuración actual no es el modelo más adecuado. Los resultados de exactitud nos indican que para el secador de pelo se acertó en un 88 %, esto nos indica que las señales del ventilador y el portátil pueden tener mayor ruido al momento de ser entrenadas o de probarse con el modelo. Se observa también de la matriz de confusión que el ventilador y el secador de pelo entregaron más de 30 % de resultados para ventilador, lo cual indica que se presenta una relación entre las características de estas señales.

- Las métricas principales se observan en la tabla 3.

Tabla 3. Principales resultados del modelo.

Metrica	Ventilador	Secador de Pelo	Computador Portatil
Exactitud	0.62	0.62	0.62
Precision	0.35	0.89	0.77

5.3. Visualización

Con las pruebas realizadas, se observa en las figuras 27, 28 y 30, que cada elemento probado se registra en base de datos, almacenando su hora de conexión, consumo de potencia y nombre de acuerdo a lo identificado por los algoritmos.

Figura 27. Registro en base de datos del electrodoméstico Fan (Ventilador)

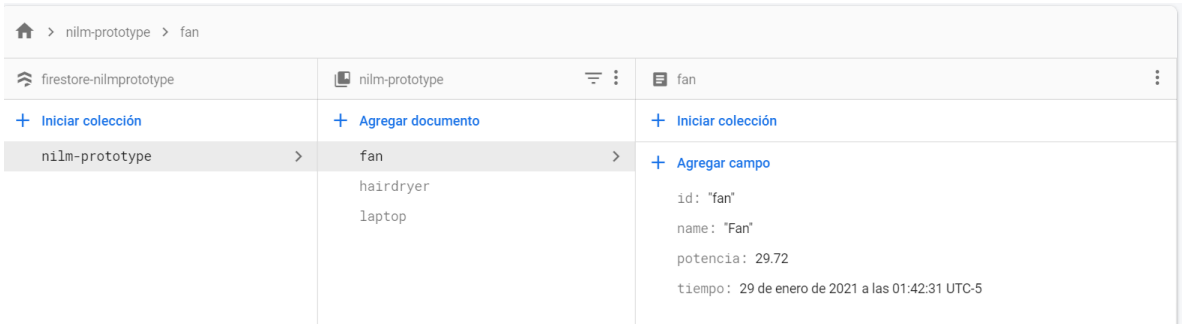
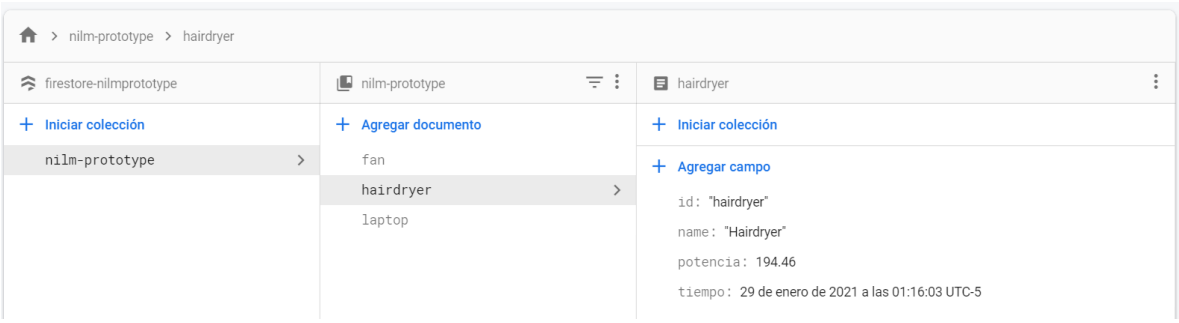
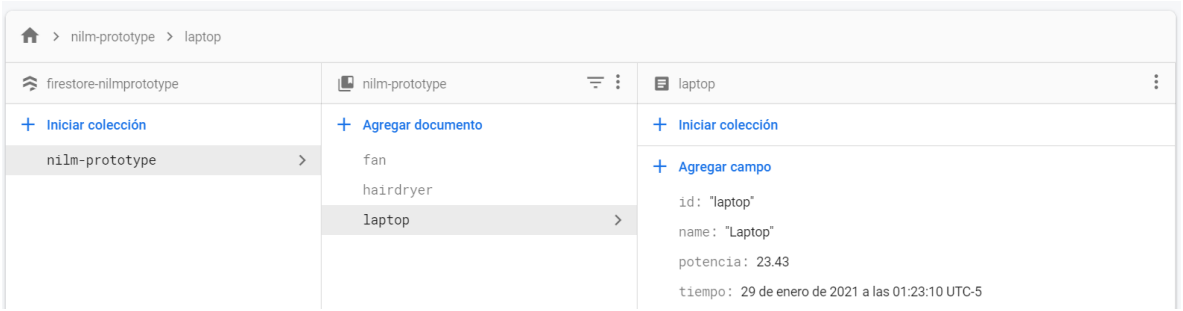


Figura 28. Registro en base de datos del electrodoméstico Hairdryer (Secador de Cabello)



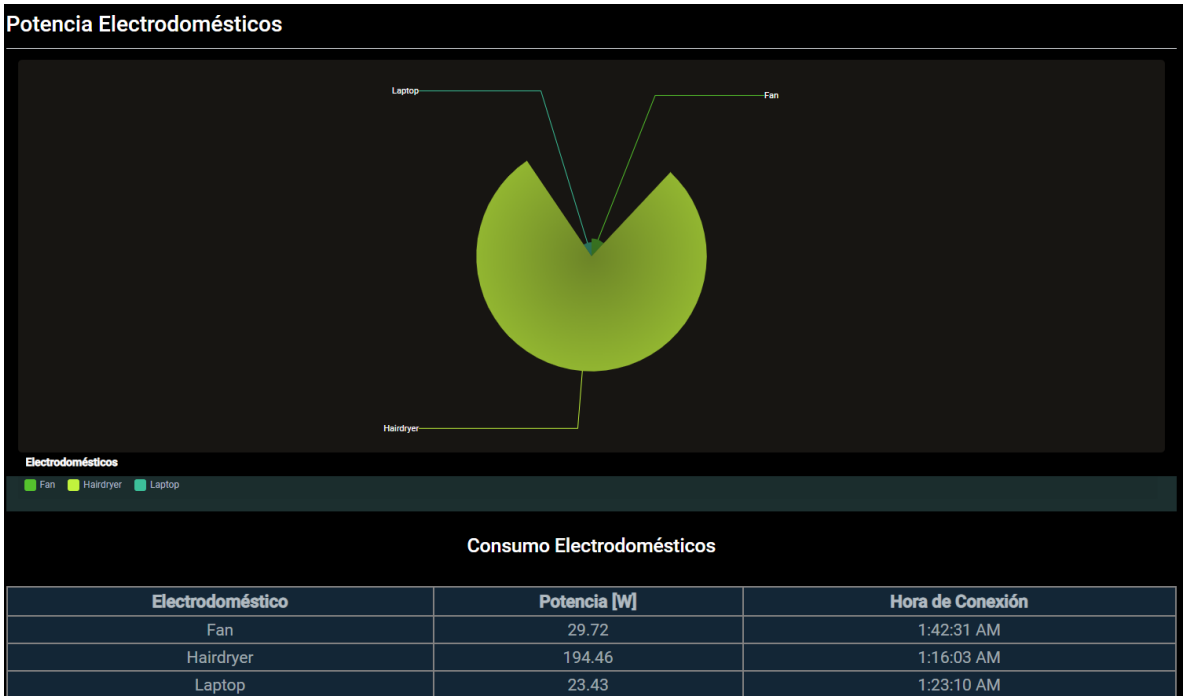
Como resultado final, se puede visualizar en la página web cada uno de los ele-

Figura 29. Registro en base de datos del electrodoméstico Laptop (Computador portátil)



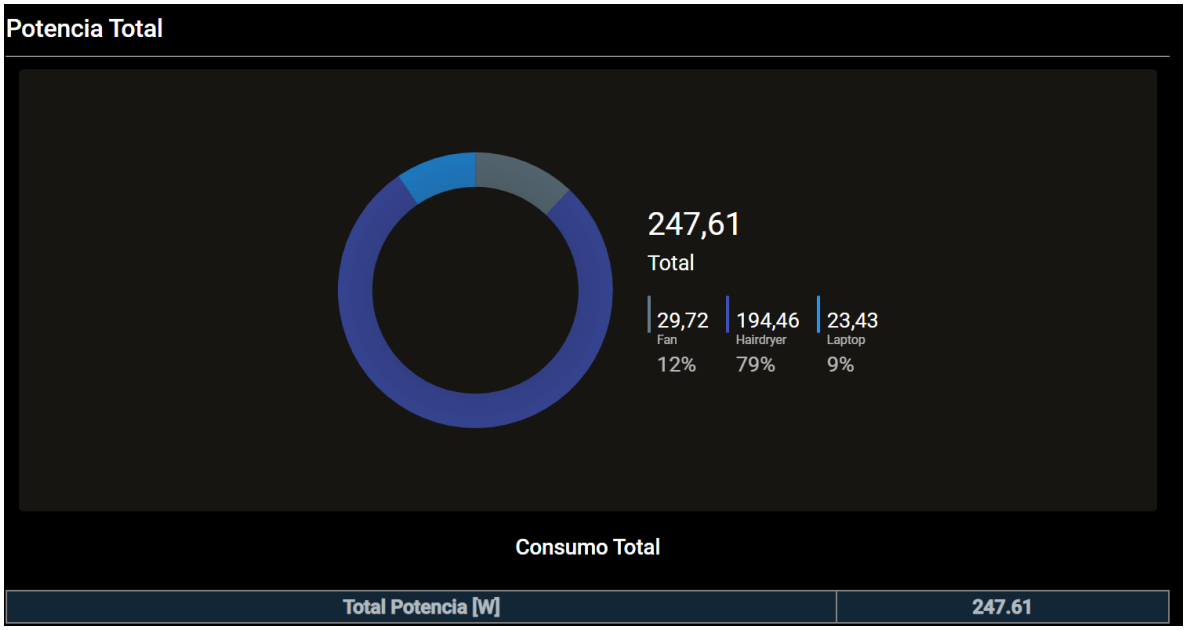
mentos que fueron identificados por la Raspberry PI 3 B+. En la figura 27, se puede observar cada uno de los electrodomésticos detectados, indicando su potencia y hora de conexión. En la página Potencia Total se pude observar el consumo energético total, de acuerdo a los elementos registrados en base de datos.

Figura 30. Visualización de datos en tiempo real página Electrodomésticos



Electrodoméstico	Potencia [W]	Hora de Conexión
Fan	29.72	1:42:31 AM
Hairdryer	194.46	1:16:03 AM
Laptop	23.43	1:23:10 AM

Figura 31. Visualización de datos en tiempo real página Potencia Total



6. CONCLUSIONES

La Raspberry Pi 3 B+ cuenta con las condiciones necesarias para procesar los datos, calcular la potencia, generar las características de la señal y clasificarlas apropiadamente haciendo uso de los algoritmos de desagregación de carga, además que permite transmitir de forma inalámbrica la información.

El modelo SVM propuesto en la tesis doctoral no fue implementado debido a disponibilidad de modelos en las librerías de Tensor Flow. Adicionalmente, el modelo de clasificación basado en Redes Neuronales Profundas tuvo un resultado de 0.62 poco satisfactorio por lo tanto, se concluye que en su configuración actual no es el más adecuado. Con estos resultados se observa que la cantidad de características ingresadas afecta el entrenamiento del algoritmo, se podrían considerar solo las más relevantes.

La transmisión de resultados de forma inalámbrica a una base de datos fue una gran ventaja, ya que permitió que la información pueda ser consultada con una mayor facilidad. La página web es accesible por cualquier usuario con conexión a internet, desde de un computador o dispositivo móvil.

El diseño de la página web facilitó la visualización en tiempo real de los datos que fueron procesados y trasmitidos por la Rasberry PI 3 B +, de manera que se puedan ver los electrodomésticos que fueron identificados y su consumo energético.

7. RECOMENDACIONES Y TRABAJO FUTURO

Al momento de enviar las señales a la Raspberry Pi 3 B+, éstas deben estar lo más limpias posibles, ya que se pueden presentar errores en el análisis si estas contienen demasiado ruido.

Es necesario que se emplee la versión 3.8 de Python o una versión anterior ya que a la fecha TensorFlow no ofrece compatibilidad para la versión 3.9 o posteriores.

Se recomienda modificar el preprocesamiento de datos y el uso de diferentes modelos de machine learning para obtener mejores métricas en el clasificador.

Es posible mejorar el tiempo de ejecución de los diferentes algoritmos ejecutados por la Raspberry se podría revisar de forma detallada el uso de memoria de cada ejecución del código. Por otro lado, se pueden gestionar los procesos de la Raspberry a nivel de Kernel y dar prioridad a la ejecución del script principal.

Además, se recomienda actualizar el backend desarrollado en Node 8 a una versión superior, ya que a partir del 15 marzo de 2021, no se admitirán ejecuciones de las funciones existentes de Node 8 en Firebase.

BIBLIOGRAFÍA

Abadi, Martín y col. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015 (vid. pág. 25).

Amazon. *¿Qué es NoSQL?* https://aws.amazon.com/es/nosql/. Consulta: 30 de Enero de 2021 (vid. pág. 36).

Angular. *Introduction to the Angular Docs*. https://angular.io/docs. Consulta: 30 de Enero de 2021 (vid. pág. 30).

Darby, Sarah. "The effectiveness of feedback on energy consumption". En: *A Review for DEFRA of the Literature on Metering, Billing and direct Displays486.2006* (2006), pág. 26 (vid. pág. 16).

Firebase. *Firebase Realtime*. https://firebase.google.com/docs/database?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 31).

Firebase. *Guías de Firebase*. url https://firebase.google.com/docs/guides?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 31).

Firebase.Analytics. *Google Analytics*. https://firebase.google.com/docs/analytics?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 31).

Firebase.Authentication. *Authentication*. https://firebase.google.com/docs/auth?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 31).

Firebase.Cloud. *Cloud Functions*. https://firebase.google.com/docs/functions?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 32).

Firebase.Firestore. *Cloud Firestore*. https://firebase.google.com/docs/firestore?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 32).

Firebase.Hosting. *Firebase Hosting*. https://firebase.google.com/docs/hosting?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 32).

Firebase.Storage. *Cloud Storage*. https://firebase.google.com/docs/storage?hl=es. Consulta: 30 de Enero de 2021 (vid. pág. 32).

Gao, Jingkun y col. "PLAID: A Public Dataset of High-Resoultion Electrical Appliance Measurements for Load Identification Research: Demo Abstract". En: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. BuildSys '14. Memphis, Tennessee: Association for Computing Machinery, 2014, 198–199. DOI: 10.1145/2674061.2675032 (vid. pág. 34).

Hart, G.W. "Nonintrusive appliance load monitoring". En: *Proceedings of the IEEE* (1992) (vid. pág. 20).

IEA. "Market Report Series energy efficiency 2017". En: *IEA Publ* (2017), págs. 1-143 (vid. pág. 15).

Jimenez, Y. "Automatic Disaggregation of Residential Electrical Consumption with Non-Intrusive Methods." Tesis. Colombia: Universidad Industrial de Santander, 2018 (vid. págs. 16, 20, 30).

Jupyter. *Jupyter Project*. https://jupyter.org/about. Consulta: 30 de Enero de 2021 (vid. pág. 30).

Kernighan, Brian. *The C programming language*. Englewood Cliffs, N.J: Prentice Hall, 1988 (vid. pág. 23).

Lenguaje.de.Programación.Ruby. *Documentación*. https://www.ruby-lang.org/es/. Consulta: 30 de Enero de 2021 (vid. pág. 23).

Numpy. *Numpy Project*. https://numpy.org/about/. Consulta: 30 de Enero de 2021 (vid. pág. 25).

Platzi. *Frontend con Angular*. https://platzi.com/desarrollo-angular/. Consulta: 30 de Enero de 2021 (vid. pág. 30).

Raschka, Sebastian. *Python machine learning : unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. Birmingham, UK: Packt Publishing, 2015 (vid. pág. 26).

Raspberry-Pi-3-Model-B+. *Documentation*. http://sindominio.net/ash. Consulta: 30 de Enero de 2021 (vid. pág. 21).

Shin, Terence. *Understanding the Confusion Matrix and How to Implement it in Python*. https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d. Consulta: 30 de Enero de 2021 (vid. págs. 28, 29).

Singh, Pramod. *Learn TensorFlow 2.0 Implement Machine Learning and Deep Learning Models with Python*. Berkeley, CA: Apress Imprint Apress, 2020 (vid. pág. 26).

team, The pandas-development. *pandas-dev/pandas: Pandas*. Ver. latest. Feb. de 2020. DOI: 10.5281/zenodo.3509134 (vid. pág. 25).

Valencia, Mateus. “Crisis Energética en Colombia”. En: *Revistas UD (Universidad distrital Francisco José de Caldas)* 4 (2016) (vid. pág. 15).

World-Energy-Council. “Escenarios Energéticos Mundiales 2017”. En: (2017), pág. 3 (vid. pág. 15).

ANEXOS

Anexo A. Manual de modificación sobre la Raspberry PI 3 B+ 3

Se debe considerar que los argumentos de entrada de la función principal que calcula las características de la señal, son dos arreglos vectoriales de igual tamaño, con la información correspondiente a la corriente y voltaje de la señal. Adicionalmente es necesario declarar la frecuencia de muestreo fs.

La ruta principal para ubicar toda la información se encuentra en /home/pi/Documents. En esta ruta se encuentran las funciones principales:

- TestCode.py: Contiene las funciones básicas del proyecto, obtención de rangos GetRanges y FinalIndivFeatureComputing5 para calcular características de la señal. Sobre este código se modifica la declaración de la señal de entrada como se observa en la figura 32 y declaración de la carpeta de entrenamiento como se observa en la figura 33.

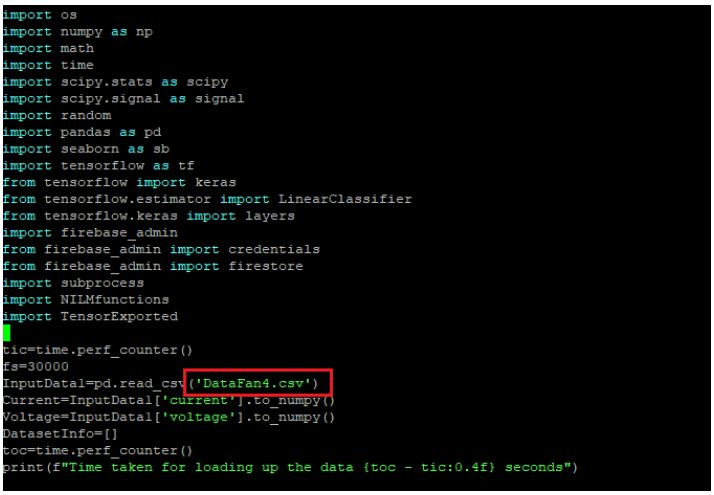
Figura 32. Nombre de la señal de entrada a modificar

```
#Prepare the pre-loaded model

print('Start Model import')
tic=time.perf_counter()
importedModel=tf.saved_model.load('/home/pi/Documents/1611865980')
toc=time.perf_counter()
print('Model imported Succesfully')
print(f"Time taken for importing model {toc - tic:0.4f} seconds")
```

- NILMfunctions.py: Contiene todas las funciones creadas para el monitor no intrusivo de potencia. Incluye la función desarrollada en una tesis doctoral previa.

- TensorExported.py: Contiene la función encargada de interpretar los resultados obtenidos por el algoritmo de caracterización de señales junto con el algoritmo de clasificación.

Figura 33. Ruta donde se ubica el algoritmo entrenado



- firestore-nilmprototype-firebase-adminsdk-78hhb-5a249053b4.json: Es la llave necesaria para acceder de forma remota a la información de la base de datos de Firebase. Fue generada de forma privada y se recomienda no compartirla para garantizar la integridad de la base de datos.

Para entrenar el modelo es necesario seguir las instrucciones entregadas en el Jupyter Notebook llamado CompleteCode, el cual contiene los comandos necesarios para exportar el modelo entrenado a una carpeta, como se indica en la sección anterior.

La demás información necesaria para entender el código y las ejecuciones realizadas durante todo el proyecto, se encuentra explicada en los Notebooks entregados.

Anexo B. Código principal en Raspberry pi

Listing 7.1. Raspberry Pi code

```
import os
import numpy as np
import math
import time
import scipy.stats as scipy
import scipy.signal as signal
import random
import pandas as pd
import seaborn as sb
import tensorflow as tf
from tensorflow import keras
from tensorflow.estimator import LinearClassifier
from tensorflow.keras import layers
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore
import subprocess
import NILMfunctions
import TensorExported

tic=time.perf_counter()
fs=30000
InputData1=pd.read_csv('DataFan4.csv')
Current=InputData1['current'].to_numpy()
Voltage=InputData1['voltage'].to_numpy()
DatasetInfo=[]
toc=time.perf_counter()
print(f"Time taken for loading up the data
{toc - tic:0.4f} seconds")
```

```
#Process the signal
print('Start Ranges extraction')
tic=time.perf_counter()
TransientRange,SteadyRange = NILMfunctions.GetRanges(
Current,100,fs,0.9)
toc=time.perf_counter()
print('Rangs extracted')
print(f"Time taken for extracting ranges {toc - tic:0.4f} seconds")

print('Start features extraction')
tic=time.perf_counter()
PowerCalc=np.multiply(Current[SteadyRange[0]:SteadyRange[1]],
Voltage[SteadyRange[0]:SteadyRange[1]]).mean()
PowerCalcRound=np.round(PowerCalc,decimals=2)
FeatureV,Feature_time,Area,Feature1,Feature2,Feature3,Feature4,
Feature7=NILMfunctions.FinalIndivFeatureComputing5(Current,Voltage,
TransientRange,SteadyRange,fs)
toc=time.perf_counter()
print('Features extracted')
print(f"Time taken for feature extraction {toc - tic:0.4f} seconds")


DatasetInfo.append([Feature1,Feature2,Feature3,Feature4,Feature7])

col_names=['FST1','FST2','FST3','FST4','FST7']
TestDF=pd.DataFrame(DatasetInfo,columns=col_names)

#Make it a big array

MasterArray=TestDF[['FST1','FST2','FST3','FST4','FST7']].to_dict('list')
Array1=np.vstack(MasterArray['FST1'])
print(Array1.shape)
Array2=np.vstack(MasterArray['FST2'])
```

```
print(Array2.shape)
Array3=np.hstack([Array1,Array2])
print(Array3.shape)
Array4=np.vstack(MasterArray['FST3'])
print(Array4.shape)
Array5=np.vstack(MasterArray['FST4'])
print(Array5.shape)
Array6=np.hstack([Array4,Array5])
print(Array6.shape)
Array7=np.hstack([Array3,Array6])
print(Array7.shape)
Array8=np.vstack(MasterArray['FST7'])
print(Array8.shape)
Array9=np.hstack([Array7,Array8])
print(Array9.shape)
FeaturesArray=Array9

Headers=[]
for i in MasterArray.keys():
    for j,k in enumerate(MasterArray[i][0]):
        Headers.append(i+'_'+str(j))
print(len(Headers))


#Finish the dataframe
FeaturesDataFrame=pd.DataFrame(FeaturesArray,columns=Headers)

#Prepare the pre-loaded model

print('Start Model import')
tic=time.perf_counter()
importedModel=tf.saved_model.load('/home/pi/Documents/1611865980')
toc=time.perf_counter()
```

```
print('Model imported Succesfully')
print(f"Time taken for importing model {toc - tic:0.4f} seconds")


print('Prediction start')
tic=time.perf_counter()
predictions = TensorExported.predict(FeaturesDataFrame, importedModel)
newPreds = []
for pred in predictions:
    newPreds.append(np.argmax(pred["probabilities"]))
toc=time.perf_counter()
print('Prediction Finish')
print(f"Time taken for making the prediction {toc - tic:0.4f} seconds")


if newPreds[0]==1:
    Result='hairdryer'
elif newPreds[0]==0:
    Result='fan'
elif newPreds[0]==2:
    Result='laptop'

print('The detected device is : ',Result)

print('Start sync to Firebase DB')

tic=time.perf_counter()
cred=credentials.Certificate(
'firestore-nilmprototype-firebase-adminsdk-78hhb-5a249053b4.json')
firebase_admin.initialize_app(cred)
db=firestore.client()
main_coll=db.collection('nilm-prototype')
UpdateDoc=main_coll.document(Result)
```

```
UpdateDoc.update({'potencia':PowerCalcRound,
'tiempo':firestore.SERVER_TIMESTAMP})
toc=time.perf_counter()
print('Sync to DB finished')
print(f"Time taken for uploading to DB {toc - tic:0.4f} seconds")
```

Anexo C. Códigos de peticiones GET a base de datos (Backend)

Listing 7.2. Raspberry Pi code

```
import * as functions from 'firebase-functions';
import * as admin from 'firebase-admin';
import * as express from 'express';
import * as cors from 'cors';

const serviceAccount = require('./serviceAccountKey.json');

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://firestore-nilmprototype.firebaseio.com"
});

const db = admin.firestore();

//Metodos de consulta en base de datos
export const getDatos = functions.https.onRequest( async(request, response) => {

  const nimlRef = db.collection('nilm-prototype');
  const docsSnap = await nimlRef.get();
  const elementos = docsSnap.docs.map( doc => doc.data() );

  response.json( elementos );

});

const app = express();
app.use( cors ({ origin: true }) );

app.get('/nilm-prototype',async (req,res) => {
```

```
  const nimlRef = db.collection('nilm-prototype');
  const docsSnap = await nimlRef.get();
  const elementos = docsSnap.docs.map( doc => doc.data() );

  res.json( elementos );

});

export const api = functions.https.onRequest(app);
```