Logotipo

Descripción generada automáticamente

PROYECTO FINAL DE CICLO

**TÍTULO**

**CICLO FORMATIVO**

TÉCNICO EN XXXX

**NOMBRE DEL ALUMNO**

**CURSO 20XX-20XX**

**INFORME EVALUACIÓN TRABAJOS FIN DE CICLO**

|  |
| --- |
| ALUMNO: |
| TITULO DEL PROYECTO: |
| CENTRO DE PRACTICAS |

**VALORACIÓN**

|  |
| --- |
|  |
|  |
|  |
|  |

Originalidad / Innovación\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Viabilidad/ Realista\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Forma/Presentación\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Defensa Oral \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Observaciones y Comentarios

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Murcia, días xx y xx de junio de 202x

Tribunal constituido a xxx de xxx de 202x para la defensa de los trabajos:

Fdo. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fdo. \_\_\_\_\_\_\_\_\_\_\_\_

**DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD PROYECTO FINAL DE CICLO**

Yo, **\_\_\_\_\_\_\_\_\_\_\_\_**, con documento de identificación **\_\_\_\_\_\_\_\_\_** y estudiante del CICLO DE GRADO SUPERIOR \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ en CESUR MURCIA, declaro que asumo la autoría y originalidad de la memoria de prácticas entregada.

Murcia, a xx de junio 202x

Fdo.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**AGRADECIMIENTOS**

**RESUMEN**

**ABSTRACT**

**ÍNDICE**

[Introducción 14](#_Toc125622681)

[Marco teórico 14](#_Toc125622682)

[Objetivos 14](#_Toc125622683)

[Metodología 14](#_Toc125622684)

[Resultados 14](#_Toc125622685)

[Conclusiones 14](#_Toc125622686)

[Bibliografía 15](#_Toc125622687)

Introducción

The growing interest in **Internet of Things (IoT)** technologies has transformed the way people interact with their homes, providing smarter and more efficient solutions for everyday tasks. **Home automation and security** are two significant areas where IoT has become highly impactful, enabling homeowners to monitor and control various aspects of their environment remotely. This project focuses on utilizing affordable and cheap IoT components to create a connected system that enhances **security**, **energy efficiency**, and **comfort** for the user in their home.

## System Overview

The setup is mainly built around a **Microcontroller**, capable of being onboarded with custom software, which can be connected to a breadboard for ease of build, low cost, and connection to a Wi-Fi network. The microcontroller of choice for this project is the **ESP8266**. By integrating several sensors with the ESP8266, the system is designed to monitor environmental factors such as temperature, humidity, motion, and the state of essential home parts such as windows or doors.

A connected ESP8266 guarantees real-time connection to the sensors, allowing homeowners to access their home status in real time. Additionally, it enables automated control over appliances, utilizes sophisticated algorithms, and employs **machine learning** to save energy consumption massively. This provides a user-friendly way to control the home from the palm of their hand.

## Affordability and Functionality

The focus of this project is **affordability** and **functionality**. Many home automation systems come with a high price tag, which is why IoT systems aren't widely supported, despite the level of technological advancement in this field. This approach demonstrates that it is possible to build an **effective and reliable system** using **low-cost components** like the ESP8266 and readily available sensors for very little cost.

Despite its affordability, this system doesn't compromise on important features, such as **real-time monitoring**, **remote control**, and **automated responses** based on sensor data.

## Automation and Energy Efficiency

The system can automatically control appliances based on sensor inputs. For example:

* **Lights** can be switched off when no motion is detected.
* **Fans**, **air conditioning units**, **heaters**, and **humidifiers** can be regulated based on temperature readings.

This ensures that appliances are only active when needed, which significantly lowers energy consumption, especially when no one is present in the home.

## Machine Learning for Smart Automation

Beyond simple functionality, this system incorporates **Machine Learning** to elevate automation to a whole new level. Unlike rule-based systems that rely on predefined, hardcoded values and thresholds, machine learning allows the system to **adapt** and **improve its responses over time**.

One key application is **temperature management**. By analyzing temperature data patterns, the system can predict potential temperature spikes and adjust the air conditioning in advance, ensuring a comfortable indoor environment while saving energy.

## Anomaly Detection for Security

Another critical role of machine learning is **anomaly detection**, particularly for **security purposes**. By feeding the system with sensor data, it learns typical behaviors, such as regular motion patterns or usual door and window usage, and flags anomalies that may indicate potential risks.

For instance, if a door sensor detects unexpected activity late at night, the system can alert the user immediately, adding a proactive layer of security.

## Seamless Communication and Scalability

The system emphasizes **seamless communication** and **scalability**. Using the ESP8266's Wi-Fi capabilities, sensor data is transmitted to a server in real time, enabling continuous monitoring and control from anywhere. The data is processed and stored in an SQL database, where it can be further analyzed to improve system behavior.

This approach allows for easy expansion. More sensors or devices can be added to the system, ensuring scalability while securely storing data for long-term analysis and machine learning training.

## User Interface and Implementation

Users have access to both a **web app** and a **mobile app** that utilize the system's **cloud-based services**. This design facilitates real-time communication between the hardware, server, and user interfaces. Users can receive updates or control connected appliances from any location. The system can be implemented in any home setup without requiring complex or expensive infrastructure.

Marco teórico

The project is built around the **ESP8266**, a low-cost, Wi-Fi enabled microcontroller widely used in IoT systems. Its primary advantage lies in its ability to easily connect to Wi-Fi networks and communicate with other devices via standard network protocols, making it perfect for **real-time data transmission**, which is crucial for smart home automation.

### **ESP8266 Specifications**

The ESP8266 is a **development kit** that allows programming of Wireless LAN-controlled projects with open-source firmware. It has the ability to store 96 KB of data, 64 KB of RAM, uses a **2.4 GHz frequency** with a **802.11 b/g/n** wireless standard. It utilizes several key features important to this project, such as:

* **Processor cores**
* **GPIO (General Purpose Input/Output)**
* **PSRAM (Pseudo-Static RAM)**
* **WIFI, Timers, RTC**
* **GDB debugging** capabilities

### **Sensor Integration with ESP8266**

The ESP8266 interacts with a variety of sensors, each designed to monitor specific aspects of the home environment:

* **Temperature sensor (e.g., DHT22)**: This sensor monitors ambient temperature and relays that data to the ESP8266. Accurate temperature readings are essential for controlling **HVAC systems** (heating, ventilation, and air conditioning), helping to maintain comfort and optimize energy use.
* **Humidity sensor**: Often included along with the temperature sensor (e.g., DHT22), this measures moisture levels in the air, allowing for automated responses such as turning on a dehumidifier or ventilation system when humidity exceeds a set threshold.

The **DHT22** offers a measuring range of **-40°C to 80°C**, with a very reliable accuracy of ±0.5°C. It operates with voltages between **3.3V to 5.5V**, which makes it perfect for our setup, ensuring flexibility with different power supplies.

* **Motion sensor (e.g., PIR sensor)**: This sensor detects movement in a specified area and is essential for security applications (e.g., activating lights or sounding alarms when motion is detected). It also contributes to energy savings (e.g., turning off lights when no motion is detected).

The **PIR sensor** measures **infrared (IR) radiation** emitted by objects, detecting movement based on temperature changes. It operates at **3.3V to 5V**, making it compatible with the ESP8266. Detection range is around **7 to 10 meters** with a viewing angle of **90 to 110 degrees**.

* **Door/Window sensor (e.g., magnetic switch)**: This sensor detects whether a door or window is open or closed, providing crucial security data to the system.

These sensors feed real-time data into the ESP8266, which processes and transmits it over Wi-Fi to a back-end server. This allows real-time control of appliances, improving both **security** and **energy efficiency**.

### **Machine Learning Integration**

A key differentiator of this project is the integration of **machine learning (ML)**. While traditional IoT systems operate based on static rules (e.g., turning on air conditioning if the temperature exceeds a threshold), machine learning allows the system to **adapt**, **learn from data**, and make predictions.

* **Anomaly Detection**: This is one of the primary machine learning concepts applied in this project. By analyzing patterns over time, the system can identify behaviors that fall outside the norm and alert the user to potential issues. Examples of anomalies include:
  + **Security anomalies**: Detecting unexpected movement, which could indicate a potential security breach.
  + **Temperature anomalies**: Identifying sudden spikes or drops in temperature, which could trigger actions like adjusting the HVAC system or alerting the user.

### **Research in IoT and Machine Learning**

This project builds on established research in both **IoT** and **machine learning** for home automation. Numerous studies have demonstrated the effectiveness of IoT in **real-time monitoring** and **remote control** of household appliances, but recent advancements in machine learning have pushed these systems to the next level by enabling **predictive** and **adaptive behavior**.

A 2020 study published in the **Journal of Internet of Things and Machine Learning** explored how integrating machine learning with IoT systems can lead to **more efficient energy management** in smart homes. By analyzing sensor data patterns, machine learning algorithms can predict user behaviors and automatically adjust settings like temperature and lighting, optimizing energy use. This research forms the foundation for the predictive elements in our project.

In terms of security, a study from **IEEE’s Internet of Things Magazine** emphasized the importance of anomaly detection in home security systems. The authors highlighted that machine learning models trained on motion and door sensor data can detect **unusual patterns** and alert homeowners in real time. This concept of anomaly detection for security is applied directly in our project to improve the security features of the smart home.

### **Technical Aspects and Coding**

This project combines both hardware and software components to achieve its goals:

* **Hardware**: The ESP8266 microcontroller is responsible for interfacing with the sensors, collecting data, and transmitting it to the server.
* **Firmware**: The ESP8266 runs a **C++ based firmware**, which handles the logic for sensor data collection and transmission.
* **Server-side**: The server is built using **Python**, a popular programming language for IoT applications, which receives sensor data, stores it in a SQL database, and runs machine learning models to analyze the data.

#### ****Key Libraries****:

* **Flask/Django**: Python web frameworks that manage the API through which the ESP8266 sends data to the server. These frameworks also serve the web and mobile applications, allowing users to interact with the system.
* **TensorFlow**: A powerful machine learning library used to build and train **anomaly detection models**. TensorFlow processes the sensor data and determines when the system should trigger alerts or take corrective actions.
* **SQL**: A relational database used to store historical sensor data, enabling long-term views of sensor activity for machine learning and anomaly detection.

### **Front-End Development: React and React Native**

The front-end is built with **React** and **React Native**:

* **React**: Used for the web interface, allowing users to monitor sensor readings, control appliances, and receive alerts via a web browser.
* **React Native**: Powers the mobile application, enabling users to control the system from their smartphones or tablets. **React Native** allows much of the React codebase to be reused, ensuring the mobile app mirrors the functionality of the web app.

Objetivos

The primary objectives of this project revolve around building a **cost-effective**, **energy-efficient**, and **intelligent** home automation system using **ESP8266**, a range of **sensors**, and **machine learning**. These objectives serve as the foundation for the project, helping to organize the work into manageable, measurable tasks and ensuring that the system delivers value in terms of both **automation** and **intelligence**. Below are the detailed objectives:

#### ****Clarifying the Features of the IoT Project****

This project focuses on automating a variety of home appliances and monitoring security aspects using sensors and ESP8266. The key features include:

* **Security Monitoring**: Real-time monitoring using door, window, and motion sensors to alert the user in case of security anomalies.
* **Environmental Control**: Monitoring indoor conditions such as temperature and humidity, and automating responses like adjusting the air conditioning or dehumidifier based on sensor readings.
* **Real-Time Data Logging and Control**: The system ensures that sensor data is continuously logged and analyzed for real-time decision-making, enabling intelligent control over connected devices like fans, lights, or alarms.

#### ****Specific Tasks and Goals****

To ensure the project's success, the following tasks and goals have been set:

* Implement a **low-cost IoT system** for home automation using the **ESP8266** microcontroller, incorporating both **security** and **environmental monitoring**.
* Ensure that the system can **log real-time sensor data** and control **at least 3 types of appliances** (e.g., lights, fans, air conditioning).
* Develop both a **web** and **mobile interface** to allow users to monitor sensor data, control devices, and receive alerts, all in real-time.

#### ****SMART Framework for Objectives****

To ensure that the objectives are clear and achievable, the **SMART criteria** have been used: **Specific**, **Measurable**, **Achievable**, **Relevant**, and **Time-bound**.

* **Specific**: Design and implement a **low-cost, energy-efficient IoT system** for home automation using **ESP8266**, focusing on **security** and **environmental monitoring**. This includes developing a fully integrated system with multiple sensors (temperature, motion, door/window sensors) that can automate appliance control and alert users in real-time.
* **Measurable**: The system will be considered successful if it logs data from **at least 5 sensors** and controls **at least 3 types of appliances** based on sensor data. This includes turning appliances on or off, adjusting settings based on temperature or humidity, and detecting and responding to security threats.
* **Achievable**: To ensure that the goals are within reach, the **Wokwi simulation environment** will be used. This allows for real-time communication between the **ESP8266** and the back-end server, simulating the interactions between sensors, devices, and the server without needing physical hardware during the initial development phase.
* **Relevant**: Leveraging **machine learning** is crucial for achieving **intelligent automation**. By incorporating models that can detect **anomalies** in sensor behavior (e.g., sudden temperature spikes or unexpected motion), the system can take proactive actions, making the project not only reactive but also predictive and preventive.
* **Time-bound**: The entire system—including sensors, server communication, machine learning integration, and app development—will be completed within a **defined timeline**, ensuring that development stays on track.

#### ****Developing a Full System for Monitoring and Control****

One of the key goals is to develop an integrated system that includes the following components:

* A **web application** and **mobile application** built using **React** and **React Native**, allowing users to view sensor data and control devices from anywhere.
* A **back-end server** (built using Python and Flask/Django) to handle data storage, machine learning processing, and real-time communication with the ESP8266 and sensors.
* **Real-time data processing and automation**, where the server processes incoming sensor data and triggers actions such as turning on a fan, sending security alerts, or adjusting the air conditioning automatically based on pre-trained machine learning models.

### **Summary of Objectives**

In summary, this project aims to build an IoT system that is **affordable**, **energy-efficient**, and **intelligent**, using ESP8266 to monitor security and environmental conditions. Through the use of machine learning and real-time automation, the system will be capable of improving the quality of life at home by automating common tasks and ensuring that appliances are only active when necessary, all while providing enhanced security through anomaly detection.

Metodología

This section outlines the step-by-step process undertaken to achieve the objectives of this IoT home automation project. The focus is on the practical implementation of the system, which integrates various sensors, machine learning, and the ESP8266 microcontroller to enhance security and energy efficiency.

#### ****1. Participantes (Participants)****

The project is being developed by a single student (your name), under the supervision of a tutor. The student is responsible for designing the system, integrating hardware and software components, implementing the machine learning models, and testing the system. The tutor provides guidance to ensure the project meets the objectives and aligns with the course requirements.

#### ****2. Búsqueda de Información (Research Process)****

To develop the project, research was conducted on:

* **ESP8266 and sensor documentation** to understand the hardware requirements and integration processes.
* **Machine learning resources**, focusing on anomaly detection and regression models applicable to IoT systems.
* **OpenWeatherAPI** documentation for retrieving real-time weather data to enhance the prediction models.
* **Academic papers and online resources** related to IoT in home automation and the use of machine learning for predictive control and security.

#### ****3. Descripción del Proceso (Process Description)****

#### ****a) System Design****

#### The system architecture revolves around the ESP8266 microcontroller, which connects to various sensors (temperature, humidity, motion, and door/window sensors). These sensors collect data from the environment and transmit it to a server, where the data is stored and processed. The server, built in Python, uses this data to control appliances like lights and air conditioners through a web and mobile application.

#### ****b) Sensor Integration and Data Collection****

#### The system uses multiple sensors to monitor environmental and security conditions in the home:

****Temperature and humidity sensors (DHT22)**:** These sensors monitor the ambient conditions and send data to the ESP8266, which transmits it to the server. This data is used to automate devices such as air conditioners and dehumidifiers.

**Motion sensors (PIR)**: The PIR sensor detects movement and triggers actions, such as turning on lights or sending a security alert when motion is detected in specific areas

**Door/window sensors**: These sensors detect the opening and closing of doors or windows and send real-time security alerts to the system.

The ESP8266 handles the communication between the sensors and the server using its built-in Wi-Fi capabilities.

**c) Machine Learning Model Implementation**

###### A machine learning model is used to predict indoor temperature based on the data collected from the sensors and external weather data from OpenWeatherAPI. The goal is to automate air conditioning to ensure a comfortable environment while optimizing energy use. The model, initially a linear regression, is trained using synthetic data to simulate the effect of environmental changes and AC operation. Depending on performance, more complex models like Random Forest may be explored.

##### **d) Data Communication and Real-time Processing**

##### The ESP8266 sends sensor data to the server via HTTP requests. The server processes this data and stores it in an SQL database. Based on predefined conditions (e.g., if the temperature rises above a certain threshold), the system automatically adjusts appliances. Additionally, the machine learning model predicts future indoor temperatures, allowing the system to proactively control devices like air conditioners or fans.

##### **e) Web and Mobile Application Development**

##### The user interface is built using React for the web application and React Native for the mobile application. These applications allow users to monitor sensor readings and control appliances remotely. Users can also receive real-time alerts for events such as motion detection or changes in temperature.

#### ****4. Cronología (Timeline)****

#### The project was divided into several phases to ensure completion within the given timeframe:

#### ****5. Evaluación y Seguimiento del Programa (Evaluation and Monitoring)****

#### To ensure the system’s reliability and effectiveness, regular testing was performed. This included:

* **Sensor accuracy checks**: Ensuring that all sensors were functioning correctly and transmitting accurate data.
* **Machine learning model evaluation**: The model's performance was evaluated using metrics such as Mean Squared Error (MSE) and R² score to ensure that predictions were accurate enough for real-time control​
* **System responsiveness**: Testing the system's ability to control appliances in real-time based on sensor data and machine learning predictions. Feedback from the system was monitored, and adjustments were made as necessary.

#### ****6. Secuencia de Actividades (Sequence of Activities)****

#### The project was carried out through the following sequence of activities:

#### ****System design**:** The architecture was drafted, and the components were selected based on the project’s goals.

#### ****Hardware setup**:** Sensors were connected to the ESP8266, and initial tests were performed to ensure correct functionality.

#### ****Data transmission setup**:** The ESP8266 was programmed to send data to the server via HTTP requests.

#### ****Machine learning model implementation**:** The machine learning model was developed and trained using synthetic data.

#### ****Front-end development**:** The web and mobile apps were developed to allow user interaction with the system.

#### ****Testing and refinement**:** The system was tested in various scenarios to ensure that it met the performance and reliability requirements.

#### ****7. Justificación de la Intervención (Justification of the Approach)****

#### This project takes an affordable approach to home automation by utilizing the low-cost ESP8266 microcontroller and readily available sensors. The use of machine learning for predictive control allows the system to proactively manage home appliances, ensuring comfort and security while optimizing energy efficiency.

Desarrollo

Supone el núcleo del trabajo. La estructura se debe adaptar al tipo de proyecto que se lleva a cabo. Aquí se deben poner de manifiesto los conocimientos adquiridos por el alumno. Pueden aparecer sub-apartados como:

1. Secuencia de actividades
2. Justificación de la intervención
3. Revisión de literatura sobre una temática
4. Evaluación y seguimiento del programa

Conclusiones

Breve reflexión final que integre y sintetice todo el proyecto. Aquí deben aparecer:

1. Principales hallazgos y/o aprendizajes a raíz del proyecto
2. Valoración del proyecto con respecto a los objetivos propuestos

Bibliografía

En la bibliografía se deben introducir todas las fuentes de información de donde se han obtenido las ideas para la elaboración del trabajo. Es decir, se deben introducir las referencias, ya sean de libros, artículos, páginas web, blogs, etc. de las que se haya encontrado la información. ¡Ojo! No significa que se pueda copiar y pegar información de alguna de esas referencias, sino que la idea se ha sacado de esa referencia, y posteriormente, desde vuestras palabras, habéis transcrito al presente trabajo. A continuación, se exponen ejemplos de cómo se introducen las referencias bibliográficas. Nos basaremos en las normas APA 6ª Edición, las cuales podéis encontrar en Google.

Ejemplo: Alonso, D. y Campo, J. (2001). Iniciación al atletismo en primaria. Barcelona: Inde.