PLATAFORMA WEB DE INTERNET DE LAS COSAS PARA ENTORNOS EDUCATIVOS

INTERNET OF THINGS WEB PLATFORM FOR EDUCATIONAL ENVIRONMENTS

JOSÉ ALEXANDER ESCOBAR SOTO

Tesis

Asesor

Martin Alonso Tamayo Vélez

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA

MAESTRÍA EN INGENIERÍA

MEDELLÍN

2025

PLATAFORMA WEB DE INTERNET DE LAS COSAS PARA ENTORNOS EDUCATIVOS

INTERNET OF THINGS WEB PLATFORM FOR EDUCATIONAL ENVIRONMENTS

José Alexander Escobar Soto

Universidad EAFIT, Centro de laboratorios, Analista técnico, Cra. 49, 7 Sur-50, Medellín, Colombia, +57 311 6070965, jescob44@eafit.edu.co

Received: DD/MM/AA – Reviewed: DD/MM/AA -- Accepted: DD/MM/AA - DOI: https://dx.doi.org/10.6036 (To be completed by Editor)

ABSTRACT:

The Internet of Things (IoT) is a technology that enables the interconnection of devices through a network, either among themselves or via the internet. This interconnection facilitates the remote control and monitoring of devices in real-time, with applications ranging from smart homes to complex industrial process control and security systems.

In the current educational environment, higher education institutions face significant challenges in training professionals skilled in these technologies. Existing commercial IoT platforms impose limitations such as limited trial periods, restricted data capacity, and locked functionalities, requiring costly subscriptions to access all features.

to solve this issue, a completely free educational IoT platform has been developed, specifically designed for students in engineering, computing, and related fields. This platform does not have any of the aforementioned limitations, and additionally, the learning curve for its use is not as steep as in other platforms.

Resumen:

Internet de las cosas (IoT) es una tecnología que permite la interconexión de dispositivos a través de una red, ya sea entre sí o a través de internet. Esta interconexión facilita el control y monitoreo remoto de dispositivos en tiempo real, con aplicaciones que abarcan desde hogares inteligentes hasta sistemas industriales complejos de control de procesos y seguridad.

En el entorno educativo actual, las instituciones de educación superior enfrentan desafíos significativos para formar profesionales capacitados en estas tecnologías. Para hacerlo se usan las plataformas IoT comerciales existentes las cuales imponen limitaciones como periodos de prueba limitados, capacidad limitada de datos y funcionalidades bloqueadas, para acceder todas las funcionalidades se deben parar costosas suscripciones.

Para solucionar esta problemática, se desarrolla una plataforma IoT educativa completamente gratuita, diseñada específicamente para estudiantes de ingeniería, computación y áreas afines, esta plataforma no tiene ninguna de las limitaciones anteriormente mencionadas, adicionalmente la curva de aprendizaje para su uso no es tan alta como en otras plataformas.

Palabras clave: Internet de las cosas (IoT), nube, red, frontend, backend, protocolo, valores separados por coma (CSV), frameworks, microframeworks, bases de datos, flask, http, JavaScript, css, protocolo de comunicación, html, librería, servidor, sensor, actuador, entrada analógica, entrada digital, wsl

FINANCIACIÓN

[Laboratorio de Mecatrónica y mecánica experimental, Universidad EAFIT]

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

1. Introducción:

En los últimos años ha habido grandes cambios en muchas áreas relacionadas con la tecnología, esto ha obligado a los profesionales en áreas afines con estas a tener las capacidades suficientes para poderse adecuar de manera satisfactoria, teniendo en cuenta esto las instituciones educativas se ven en la necesidad de capacitar a los futuros profesionales en dichas áreas.

Una de las tecnologías que más ha evolucionado es el Internet de las Cosas (IoT) ya que tiene la capacidad de adaptarse y revolucionar distintas áreas como son industria, salud, agricultura, educación entre otras. Al tener un campo de acción tan grande se crea la necesidad de personal capacitado en el manejo de dispositivos IoT, debido a esto se ve la necesidad de instruir futuros profesionales con conocimiento en esta área que puedan crear soluciones innovadoras y eficientes. En la Figura 1 se puede apreciar una representación de dispositivos de distintos tipos conectados a una nube lo que es una red IoT.[1], [2].

Fig. 1. Representación de dispositivos conectados a una nube IoT. Creado con Copilot.

Para que las instituciones educativas se puedan adaptar a las nuevas tecnologías IoT deben tener en cuenta lo siguiente [2]:

- Las limitaciones en infraestructura para experimentación práctica.
- Alto costo en las suscripciones a plataformas comerciales de IoT.
- Rápida obsolescencia de herramientas tecnológicas.

Los problemas mencionados generan una necesidad de desarrollar soluciones que permitan el acceso a tecnologías IoT y que faciliten la experimentación práctica, para solucionar esto el presente trabajo tiene como propósito fundamental desarrollar una plataforma IoT educativa con las siguientes características:

- Permita el acceso a tecnologías IoT.
- Eliminar los costos de uso de plataformas.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

- Proporcione un entorno de aprendizaje práctico y de fácil uso.
- Gracias a su facilidad de uso incentive a los estudiantes a seguir explorando tecnologías IoT.
- Facilite la integración entre las instituciones educativas y la industria.
- Uso de dispositivos de bajo costo.

El presente trabajo tiene como propósito principal desarrollar una plataforma IoT educativa que permita a los estudiantes hacer lo siguiente:

- Disponer una plataforma IoT de forma gratuita.
- Acceder a un entorno de aprendizaje práctico y fácil de usar.
- Conectar dispositivos físicos IoT con la plataforma para enviar y/o recibir datos.
- Escalar sus proyectos de manera flexible.

Este proyecto se justifica con las siguientes razones:

- Con el uso de esta plataforma se pueden formar profesionales con conocimientos en tecnologías actuales y capaces de generar respuestas adecuadas a distintos problemas.
- Eliminar los costos de uso de plataformas.
- Quitar las restricciones que tienen otras plataformas.
- Crear una plataforma flexible y escalable que permita experimentar con tecnologías IoT.

Este artículo explora la creación de una plataforma IoT enfocada en el ámbito educativo, buscando hacerla intuitiva y accesible, sin requerir profundos conocimientos tecnológicos por parte de los usuarios.

2. Objetivos

2.1. Objetivos Generales

- Desarrollar una plataforma IoT educativa gratuita que permita a los estudiantes de ingeniería y áreas afines aprender y experimentar con tecnologías IoT de manera práctica y accesible.

2.2. Objetivos Específicos

- Facilitar la conexión y gestión de dispositivos IoT mediante una interfaz intuitiva y fácil de usar.
- Eliminar los costos asociados con las plataformas IoT comerciales, proporcionando una alternativa gratuita y sin restricciones.
- Proporcionar herramientas de visualización de datos que permitan a los usuarios monitorear y analizar la información de sensores en tiempo real.
- Implementar funcionalidades de gestión de usuarios para asegurar la independencia y privacidad de los datos de cada usuario.
- Evaluar el rendimiento y la estabilidad de la plataforma a través de pruebas de conectividad, rendimiento y transmisión de datos.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

3. Marco teórico y estado del arte:

3.1. Marco teórico:

Internet de las cosas es una tecnología que permite la interconexión de dispositivos (actuadores y sensores) a través de internet, al hacer esto se puede enviar y/o recibir información en tiempo real desde y hacia cualquier lugar [1]. Esta tecnología ha revolucionado muchos sectores como en la medicina, la agricultura, la educación, entre otros, para este caso se enfoca en la educación con el fin de preparar a los estudiantes para las nuevas tecnologías [2].

Una plataforma IoT para entornos educativos se debe de componer de varias las cuales son [3]:

- Capa de percepción: En esta capa se encuentran los sensores y actuadores, estos permiten medir las diferentes variables (sensores) y tomar acciones (actuadores).
- Capa de red: Se encarga de la transmisión de los datos entre los diferentes dispositivos y la plataforma IoT.
- Capa de aplicación: Esta es la plataforma IoT que permite ver los datos en forma numérica o grafica.

Las plataformas IoT generan múltiples veneficios en muchas áreas, entre ellas la educación [1], algunos de estos veneficios son:

- Aumentar el interés de los estudiantes en esta área y áreas afines.
- Automatización de procesos.
- Acceder a los dispositivos de forma remota.
- Almacenamiento de datos de los sensores.
- Creación de planes de control a partir de los datos recolectados.
- Mejoramiento de los procesos productivos.
- Ahorro de costos en mantenimientos.
- Optimizar recursos.

Estos son solo algunos de los veneficios del uso de plataformas IoT, pero es importante tener en cuenta que los veneficios pueden cambiar dependiendo del uso que se le de a la plataforma.

3.2. Estado del arte:

En los últimos años el internet de las cosas ha tenido grandes cambios y se ha popularizado enorme mente, debido a esto se han realizado muchas investigaciones sobre el impacto de este en muchas áreas, una de ellas es la educación por lo que a continuación se presentan algunos de los estudios más significativos:

- Impacto de IoT en la educación, cuáles son sus ventajas y sus desventajas y cuáles son sus principales retos para el futuro [4], [5].
- Tecnologías, protocolos y aplicaciones, adicionalmente las arquitecturas y sus respectivas capas [6].
- Integración de varias tecnologías en sistemas de comunicación para para la identificación y seguimiento, con sensores cableados e inalámbricos y actuadores [7].
- Una visión a las arquitecturas de las redes IoT, sus principales aplicaciones y el futuro de esta tecnología [8],
- IoT para cuidades inteligentes con sistemas de comunicación avanzados para apoyar servicios de administración de una ciudad [3].

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

4. Desarrollo:

Para desarrollar esta plataforma se deben de definir dos cosas inicialmente, la primera es los requisitos, estos dependen de lo que se quiere lograr con la plataforma y la segunda es la arquitectura por usar, esta depende de la manera en que se va a programar la plataforma, a continuación, se definen estos con más detalle.

4.1. Requisitos:

- ¿Qué dispositivos IoT se conectarán?

  Microcontroladores con conexión Wifi como lo son los ESP8266, ESP32 y MKR1000, inicialmente se trabaja con estos, pero más adelante se pueden agregar otros.

- ¿Qué datos enviarás o recibirás?

  Se envían y reciben cadenas de caracteres que contienen los datos, los datos como tal son numéricos, pero estos deben ir acompañado de caracteres para su identificación ya que al enviarlos en paquetes de datos es necesario poder diferenciarlos.

- ¿Qué funcionalidades tendrá la plataforma?

  Como la plataforma permite envió y la recepción de datos debe de tener funciones de visualización de datos tanto el valor numérico como también una gráfica que represente estos datos, adicionalmente los datos se deben de mostrar en tablas para poder ver el historial, adicionalmente debe contar con controles para enviar datos, estos controles pueden ser botones para definir estados de encendido y apagado como también controles deslizantes.

  Como se trata de una plataforma IoT debe contar con funcionalidades adicionales tales como Gestión de usuarios que permite agregar, modificar y borrar usuarios.

  Al tratarse de una plataforma IoT también es importante que tenga lo posibilidad de manipular algunos de los elementos, es decir, configurarlos para usarlos según la necesidad del estudiante.

  Otra función importante es el acceso a la plataforma con usuario y contraseña, esto es para independizar cada usuario, es decir, cada usuario tiene una sesión independiente con el fin de evitar que los datos de los usuarios se crucen o se pierdan. Adicionalmente los datos tienen que ser recolectados para luego poder descargarlos en formato CSV el cual se puede abrir en Excel.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

4.2. Arquitectura aplicación web:

Para definir la arquitectura de una aplicación web se debe tener en cuenta que deben contar con tres elementos principales, Backend, Frontend y protocolos de comunicación, estos elementos son primordiales ya que se encargan de partes diferentes de una aplicación y son necesarios para que la plataforma funcione correctamente, a continuación, se describirán estos tres elementos y se definirán que leguajes de programación usar para cada uno [9].

- **Backend**: Esta es la parte no visible de una página web, en esta se hace el procesamiento de los datos, gestión las solicitudes HTTP y conexión con las bases de datos.

  Para esta parte se selecciona Flask el cual se programa con el lenguaje Python, se selecciona este debido a que al programarse con Python es más simple que otras herramientas y su velocidad de ejecución es elevada.[10],[11], [12], [13], [14].

- **Frontend**: Es la parte que el usuario puede ver y con la que puede interactuar, para esta se utilizan tres leguajes de programación diferentes los cuales se usan muy a menudo en la creación de páginas web, los lenguajes usados son los siguientes:
  HTML: Se usa para definir la estructura de la página web.[15], [16].
  CSS: Se encarga del diseño de la página web ya que permite configurar tamaños, colores, ubicación entre otros de cada elemento. [17].
  JavaScript: Se encarga de la parte lógica, permite la interacción con los elementos. [18], [19], [20], [21], [22].

- **Protocolos para la comunicación**: Estos protocolos son el lenguaje en que se comunican dispositivos, es muy importante que todos los dispositivos tengan el mismo lenguaje para que se puedan comunicar entre sí, el protocolo usado para esta plataforma es HTTP. [23].

4.3. Implementación

Antes de iniciar la programación de la plataforma es necesario configurar el equipo en el que se va a realizar la programación y las pruebas, para el caso se usa un equipo con Windows 11 (no necesariamente tiene que ser Windows 10 u 11, también se puede usar Linux o Mac OS) y se procede a descargar e instalar los elementos necesarios los cuales son los siguientes:

- WSL (Windows supports Linux): Es un subsistema de Windows para Linux el cual permite ejecutar un entorno de Linux en Windows. [24], [25], [26].
- Oh My Posh: Es una herramienta que permite personalizar el área de comando cambiando la apariencia y las fuentes.[27].
- Miniconda: Es una versión reducida de Anaconda y solo incluye algunos de los paquetes de este. [28].
- Visual studio code(VSC): Es un editor de código abierto que permite usar en diferentes lenguajes de programación. [29].

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

Para programar la plataforma se usa VSC, la implementación de la plataforma se inicia por el backend el cual se programa en Flask, se definen e importan las librerías necesarias para una prueba básica, adicionalmente se genera un archivo para el frontend el cual necesita sus librerías específicas, una vez hecho esto se puede verificar que todos los elementos estén bien instalados, en la figura 2 se muestra una parte del código de prueba tanto del frontend como del backend.

```python
from flask import Flask, request, url_for, redirect, render_template, jsonify
from datetime import datetime

app = Flask(__name__)

#Punto 1
@app.route('/', methods=['GET'])
def hello_world():
    print("Hello, World!")
    return render_template("home.html")

#Punto 2
@app.route("/usuario", methods=['GET'])
def user1():
    print(userVal)
    return render_template("account.html", name = userVal)
```

templates > <> home.html > ...

```
{% extends "layout.html" %} {% block content %}
<p>Esto es un parrafo</p>
<font size="6">Prueba tamaño texto!</font>

<h2>Hello, World!</h2>

{% endblock %}
```

Fig. 2. Primera prueba del backend ala izquierda, frontend derecha.

A medida que se agregan más elementos a la plataforma el código se vuelve más extenso y complejo, en la figura 3 se puede ver fragmentos de código del backend y del frontend de un código con más elementos.

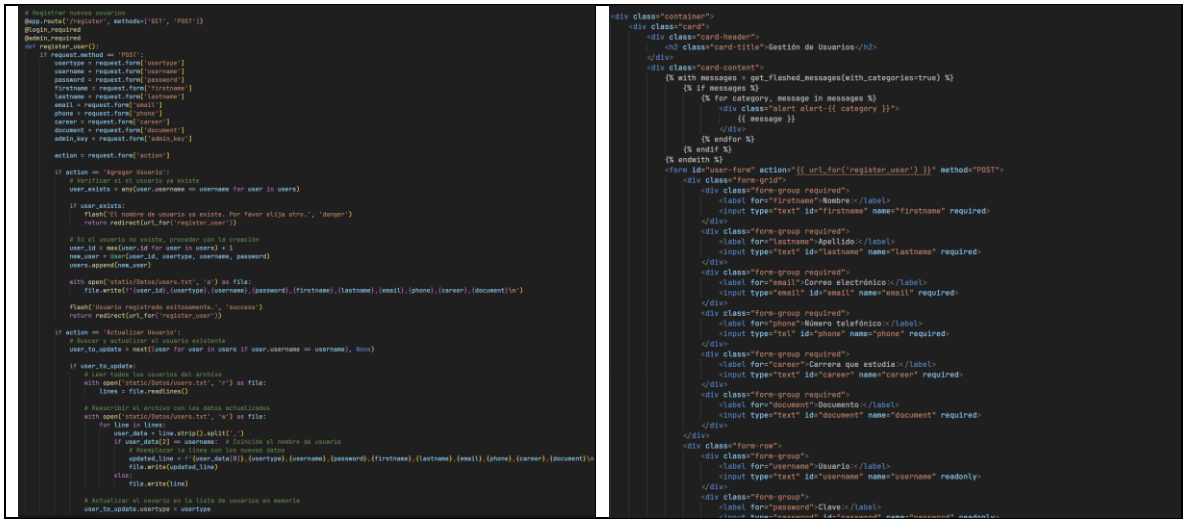

Fig. 3. Código del backend a la izquierda (flask), frontend derecha (HTML).

Como se mencionó anteriormente para la elaboración de esta plataforma se usan varios lenguajes de programación donde cada uno se encarga de una parte especifica de la plataforma, la cantidad de líneas de código es elevada por lo que solo se muestran unos fragmentos del código.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

En la figura 4 se muestran fragmentos de código del CSS que es el que maneja los estilos(apariencias) y de JavaScript que maneja la parte lógica.

```css
.btn-text-right {
    text-align: right;
}

#wrapper {
    display: flex;
    transition: margin-left 0.5s ease;
}

#main-content {
    flex-grow: 1;
    transition: margin-left 0.5s ease;
    margin-left: 270px;
    margin-right: 10px;
    margin-top: 20px;
    margin-bottom: 10px;
}

#wrapper.active #main-content {
    margin-left: 20px;
}

#wrapper.active #sidebar {
    left: -250px;
}

#sidebar {
    position: fixed;
    width: 250px;
    height: 100%;
    background: #151719;
    left: 0;
    transition: left 0.5s ease;
    overflow-y: auto;
}

#sidebar ul li {
    color: rgba(230, 230, 230, .9);
    list-style: none;
    padding: 15px 10px;
    border-bottom: 5px solid rgba(100, 100, 100, .3);
    text-align: center;
}

.logo {
    border-radius: 50%;
    display: absolute;
    margin: 0 auto;
}

#sidebar .toggle-btn {
    position: fixed;
    left: 30px;
    top: 20px;
    cursor: pointer;
}
```

```javascript
<script>
    document.addEventListener('DOMContentLoaded', function() {
        const form = document.getElementById('user-form');
        const emailInput = document.getElementById('email');
        const usernameInput = document.getElementById('username');
        const passwordInput = document.getElementById('password');
        const clearFieldsButton = document.getElementById('clearFields');
        const showUsersButton = document.getElementById('showUsers');
        const userListDiv = document.getElementById('userList');
        const addUserButton = document.getElementById('addUser');
        // const deleteUserButton = document.getElementById('deleteUser');
        const actionInput = document.getElementById('action');
        const emailContentDiv = document.getElementById('emailContent');
        const generateEmailButton = document.getElementById('generateEmail');
        const userTypeSelect = document.getElementById('usertype');

        // Función para buscar un usuario por su nombre
        async function buscarUsuario(username) {
            try {
                const response = await fetch(`/buscar_usuario/${username}`);
                if (!response.ok) {
                    throw new Error('Usuario no encontrado', username);
                }
                return await response.json();
            } catch (error) {
                console.error('Error al buscar usuario:', error);
                alert(error.message);
                return null;
            }
        }

        // Función para generar un nombre de usuario a partir del correo electrónico
        function generateUsername(email) {
            return email.split('@')[0];
        }
        // Función para generar una contraseña aleatoria de 6 dígitos
        function generatePassword() {
            return Math.floor(100000 + Math.random() * 900000).toString();
        }

        emailInput.addEventListener('input', function() {
            usernameInput.value = generateUsername(this.value);
            passwordInput.value = generatePassword();
        });

        clearFieldsButton.addEventListener('click', function() {
            form.reset();
            usernameInput.value = '';
            passwordInput.value = '';
            emailContentDiv.style.display = 'none';
            userListDiv.style.display = 'none';
        });

        addUserButton.addEventListener('click', function() {
            if (validateAddForm()) {
                if(addUserButton.textContent == 'Agregar Usuario')
                {
                    if (confirm('¿Está seguro de que desea agregar este usuario?')) {
                        actionInput.value = 'Agregar Usuario';
                        form.submit();
```

Fig. 4. Código del frontend a la izquierda (CSS), frontend derecha (Java Script).

En este punto la interfaz de la plataforma se ve como en la figura 5, en esta venta se pueden visualizar los datos recibidos y las gráficas de estos datos, también se puede ver el menú del lado izquierdo el cual sirve para navegar entre ventanas de la plataforma.

Fig. 5. Visualización de la interfaz gráfica.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

Un elemento adicional que hay que agregar es el microcontrolador, este es un dispositivo físico reprogramable al que se le pueden asignar funciones específicas, estas funciones pueden ir desde controlar otros dispositivos por medio de sus salidas o leer el estado de un sensor por medio de sus entradas, para el caso del microcontrolador a usar que es el ESP32, este cuenta con la posibilidad de conectarse con otros dispositivos por medio de Bluetooth o wifi, esta última es la que se usa para conectarse con la plataforma, en la figura 6 se muestra una imagen de un ESP32. Cabe notar que hay otros dispositivos que se pueden conectar a la plataforma. [31], [32], [33].

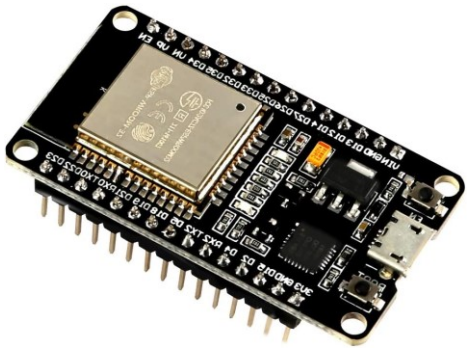

Fig. 6. ESP32. Fuente: Sigma Electrónica [30].

En la figura 7 se muestra un fragmento del código usado en el ESP32 para enviar los datos a la plataforma IoT.

```
void wifiPost() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Conexión WiFi perdida. Reconectando...");
    wifiInit();
    return;
  }

  String body = "{\"input_1\":" + String(input_1) +
                ",\"input_2\":" + String(input_2) +
                ",\"input_3\":" + String(input_3) +
                ",\"input_4\":" + String(input_4) +
                ",\"dig_1\":" + String(dig_1) +
                ",\"dig_2\":" + String(dig_2) +
                ",\"dig_3\":" + String(dig_3) +
                ",\"dig_4\":" + String(dig_4) + "}";

  http.begin(client, HTTP_HOST);
  http.addHeader("Content-Type", "application/json");
  http.addHeader("Authorization", "Basic " + encoded);

  int httpResponseCode = http.POST(body);

  if (httpResponseCode > 0) {
    Serial.printf("[HTTP] Código de respuesta: %d\n", httpResponseCode);
    String response = http.getString();
    Serial.println("[HTTP] Respuesta: " + response);
    readResponse(response);
  } else {
    Serial.printf("[HTTP] Error en la conexión, código: %d\n", httpResponseCode);
  }

  http.end();
}
```

Fig. 7. Fragmento de código para el ESP32.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

Hay que tener en cuenta que los que los fragmentos de código mostrados son solo eso, fragmentos ya que el código es muy extenso y no es posible mostrarlo todo en este artículo.

4.4. Metodología de Pruebas

Se realizan tres tipos de prueba para verificar el correcto funcionamiento de la plataforma, la primera es una prueba de conectividad, la segunda es de rendimiento y la tercera de envío y recepción de información, a continuación, se describen cada una de estas.

- **Conectividad**: Esta prueba permite verificar que los dispositivos IoT se conecten con la plataforma de forma adecuadamente. Para ello se realizan las siguientes pruebas:
    - **Acceso:** Prueba de acceso a la plataforma con diferentes usuarios.
    - **Pruebas de conexión WiFi**: Verificación de la estabilidad de la conexión de los dispositivos con la plataforma.
    - **Verificación de comunicación HTTP**: Asegurar que los dispositivos puedan enviar y recibir datos correctamente.
    - **Estabilidad de transmisión de datos**: Evaluar y verificar los datos enviados y recibidos.
- **Rendimiento**
    - **Latencia de comunicación**: Medición del tiempo de respuesta entre el envío y la recepción de datos, este puede variar dependiendo del numero de dispositivos conectados a la plataforma.
    - **Capacidad de manejo de múltiples dispositivos**: Evaluar el rendimiento de la plataforma con varios dispositivos conectados simultáneamente.
    -
- **Envió y recepción de datos**
    - **Sensores de temperatura:** Verificación de la precisión y consistencia de los datos recibidos.
    - **Actuadores digitales:** Evaluar la capacidad de la plataforma para controlar dispositivos de manera remota.
    - **Transmisión de datos en diferentes escenarios:** Pruebas en entornos con diferentes niveles de interferencia y carga de red.
- **Otras pruebas:**
  Estas pruebas consisten en navegar por las otras ventanas y verificar su funcionamiento.
    - **Mostrar datos**
    - **Dashboard**
    - **Generador de código**
    - **Gestión de usuarios**
    - **Log out**

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

Para acceder a la plataforma se deben de ingresar el usuario y la contraseña del usuario, cada usuario debe tener un usuario y una contraseña diferente con el fin de evitar los errores y la perdida de datos, en la figura 8 se puede ver la ventana de acceso.

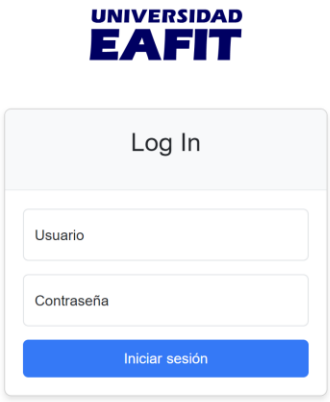

Fig. 8. Ventana de acceso.

En la figura 9 se puede ver varios computadores conectados a la plataforma al mismo tiempo, cada equipo accede a la plataforma con usuario diferente y tiene un ESP32 enviando datos a su respectivo usuario, en este momento se hacen todas las pruebas anteriormente mencionadas.

Fig. 9. Prueba de la plataforma en varios dispositivos.

Los datos se pueden ver en la figura 10, del lado izquierdo muestra los datos numéricos en una tabla, aquí se puede notar que las 6 primeras columnas corresponden a entradas analógicas, mientras que los restantes son entradas digitales, del lado derecho se muestra un gráfico por cada uno de los datos analógicos.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

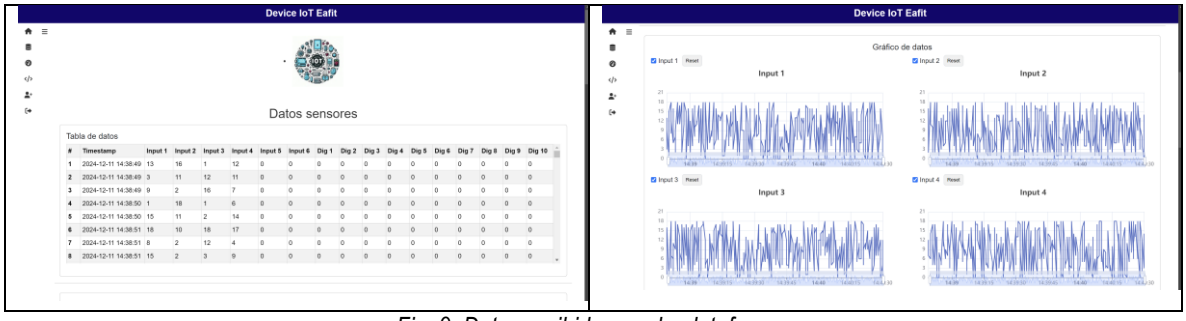

Fig. 9. Datos recibidos por la plataforma.

En la misma ventana que se muestran los datos, en la parte inferior se tienen otras herramientas muy útiles separadas por recuadros, en la figura 10 se muestran estas herramientas, la primera es Guardar y borrar datos, la opción de Guardar datos permite descargar los datos almacenados en un archivo de formato CSV el cual se puede abrir en Excel, el botón Borrar datos borra todos los datos almacenados en la plataforma, la segunda es Control de sliders, esta permite enviar tres datos con valores variables entre 0 y 100, esto permite controlar diferentes dispositivos desde la plataforma, la tercera es Matriz de elementos, en esta matriz se pueden ubicar otros elementos con los que se puede interactuar como lo son botones o indicadores.

Fig. 9. Otras herramientas.

Cuando se accede a la plataforma siempre lo hace por la ventana de inicio, pero no es la única ventana a la que se puede acceder, en la parte superior izquierda se encuentra un menú que permite ir a las otras ventanas, este menú se reducir para que solo se vean los iconos o se puede ampliar para mostrar todo el contenido, en la figura 11 se muestra el menú ampliado.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

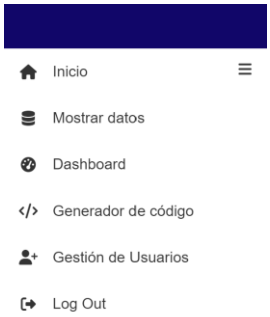

Fig. 11. Menú ampliado.

A continuación, se verifican cada una de las ventanas que tiene la plataforma:

Mostrar datos: En esta ventana se pueden cargar los archivos CSV guardados, en la figura 12 se puede ver que cuenta con dos botones uno para seleccionar el archivo desde una ubicación del computador, el otro permite ver los valores numéricos en una tabla y también en un gráfico.

Fig. 12. Mostrar datos.

Dashboard: En esta ventana se pueden arrastrar los elementos, como se puede evidenciar en la figura 13 se tiene un menú el cual cuenta con varios elementos los cuales se pueden arrastrar y soltar sobre la matriz, esto permite configurar los elementos al gusto o la necesidad del usuario, una vez configurado en la ventana de inicio se puede interactuar con estos elementos.

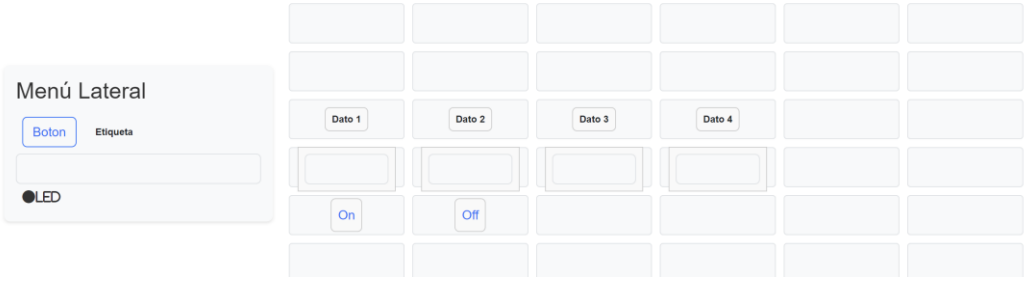

Fig. 13. Dashboard.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

Generador de código: En esta ventana se puede seleccionar uno de los ejemplos de código disponibles y ver todo el código tal como se muestra en la figura 14. Una vez abierto el código se puede presionar el botón Copiar contenido para copiar el código y poder llevarlo al software de programación.

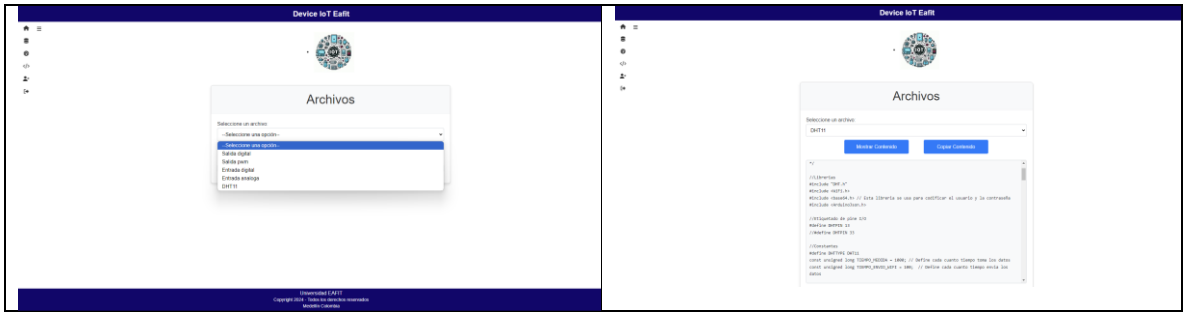

Fig. 14. Generador de código.

Gestión de usuarios: Esta ventana se utiliza para agregar, modificar o borrar usuarios, en la figura se puede apreciar las opciones que dispone.

Hay que aclarar que se pueden crear tres tipos de usuario, Estudiante, Empleado y Administrador, este último es el único que tiene los permisos necesarios para ingresar a esta venta, los otros usuarios al no tener los permisos no podrán acceder.

Fig. 15. Generador de código.

5. Conclusiones

- El desarrollo de la plataforma IoT educativa ha demostrado ser una herramienta efectiva para introducir a los estudiantes a las tecnologías IoT ya que se eliminan muchas de las barreras anteriormente enunciadas.
- La plataforma es intuitiva y fácil de usar y gracias a las pruebas realizadas se ha confirmado que los usuarios pueden aprender a utilizarla en poco tiempo.
- La plataforma es accesible para todos los usuarios ya que no tiene ningún costo, esto la hace muy útil para todos los estudiantes. Es necesario tener en cuenta que esta plataforma no tiene todas las funcionalidades de otras plataformas, pero es un buen punto de partida en el mundo del IoT.

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

- La plataforma cuenta con herramientas de visualización de los datos recibidos y de los almacenados lo que permite comprender y analizar mejor los datos de una mejor manera.
- El control de acceso mediante credenciales garantiza la privacidad y la independencia de los datos de cada usuario por lo que se tiene un entorno seguro para la experimentación.
- Las pruebas de rendimiento han demostrado la capacidad de manejar múltiples usuarios al mismo tiempo lo que demuestra la confiabilidad de la plataforma para la educación.

Futuras modificaciones podrían enfocarse en la integración de nuevas funcionalidades y la evaluación de su impacto en el aprendizaje de los estudiantes.

Referencias:

[1] Webby Lab. IoT Technology in Education [En línea]. 2024. Disponible en: https://webbylab.com/blog/impact-of-iot-technology-on-education/
[2] Phidias. Internet de las cosas: La revolución de los objetos conectados [En línea]. 2024. Disponible en: https://phidias.com/internet-de-las-cosas-iot-la-revolucion-de-los-objetos-conectados/
[3] Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. Internet of Things for Smart Cities [En línea]. 2025. Disponible en: https://doi.org/10.1109/JIOT.2014.2306328
[4] Arduino. The Impact Of IoT In Education [En línea]. 2025. Disponible en: https://www.arduino.cc/education/the-impact-of-iot-in-education
[5] Multishoring. The Impact of IoT in Education: What You Need to Know [En línea]. 2025. Disponible en: https://multishoring.com/blog/iot-in-education/
[6] Al-fuqaha A., Guizani M., Mohammadi M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications [En línea]. 2025. Disponible en: https://ieeexplore.ieee.org/document/7123563
[7] Atzori L., Iera A., Morabito G. The Internet of Things: A survey [En línea]. 2025. Disponible en: https://doi.org/10.1016/j.comnet.2010.05.010
[8] https://doi.org/10.1016/j.future.2013.01.010
[9] Elitex. Cómo Conectar Frontend y Backend [En línea]. 2025. Disponible en: https://elitex.systems/blog/how-to-connect-frontend-and-backend-all-you-need-to-know-in/
[10] Flask Documentation. Flask [En línea]. 2024. Disponible en: https://flask.palletsprojects.com/
[11] Grinberg M. Flask web development Developing web application with Python]. Sebastopol: O'Reilly Media; 2014
[12] Javatpoint. Python Flask Tutorial [En línea]. 2024. Disponible en: https://www.javatpoint.com/flask-tutorial
[13] Askpython. Flask [En línea]. 2024. Disponible en: https://www.askpython.com/python-modules/flask
[14] Javatpoint. Python Tutorial [En línea]. 2024. Disponible en: https://www.javatpoint.com/python-tutorial
[15] Javatpoint. HTML Tutorial [En línea]. 2024. Disponible en: https://www.javatpoint.com/html-tutorial
[16] Mozilla Developer Network. HTML: HyperText Markup Language [En línea]. 2024. Disponible en: https://developer.mozilla.org/en-US/docs/Web/HTML
[17] Mozilla Developer Network. CSS: Cascading Style Sheets [En línea]. 2024. Disponible en: https://developer.mozilla.org/en-US/docs/Web/CSS
[18] Mozilla Developer Network. JavaScript [En línea]. 2024. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript
[19] Javatpoint. JavaScript Tutorial [En línea]. 2024. Disponible en: https://www.javatpoint.com/javascript-tutorial
[20] Echarts. Getting Apache ECharts [En línea]. 2024. Disponible en: https://echarts.apache.org/handbook/en/get-started/
[21] Lenguajejs. Lenguaje Javascript [En línea]. 2024. Disponible en: https://lenguajejs.com/javascript/
[22] Chartjs. Chart.js: Simple yet flexible JavaScript charting for designers & developers [En línea]. 2024. Disponible en: https://www.chartjs.org/
[23] Mozilla Developer Network. Generalidades del protocolo HTTP [En línea]. 2024. Disponible en: https://developer.mozilla.org/es/docs/Web/HTTP/Overview
[24] Microsoft. Instalación de Linux en Windows con WSL [En línea]. 2024. Disponible en: https://learn.microsoft.com/es-es/windows/wsl/install
[25] Microsoft. ¿Qué es el Subsistema de Windows para Linux? [En línea]. 2024. Disponible en: https://learn.microsoft.com/es-es/windows/wsl/about
[26] Computerhoy. Cómo instalar Linux en un ordenador con Windows [En línea]. 2024. Disponible en: https://computerhoy.20minutos.es/tutoriales/como-instalar-linux-ordenador-windows-1217578
[27] Microsoft. Configuración de un símbolo del sistema personalizado para PowerShell o WSL con Oh My Posh [En línea]. 2024. https://learn.microsoft.com/es-es/windows/terminal/tutorials/custom-prompt-setup
[28] Anaconda Inc, Miniconda [En línea]. 2024. Disponible en: https://docs.anaconda.com/miniconda/
[29] Microsoft. Your code editor [En línea]. 2024. Disponible en: https://code.visualstudio.com/
[30] Sigma Electrónica. ESP-32 [En línea]. 2024. Disponible en: https://www.sigmaelectronica.net/producto/esp-32/

ISSN: 0012-7361 eISSN: 1989-1490 / DYNA Vol.XX nºX DOI: https://doi.org/10.6036/XXXX

[31] Espressif Systems. ESP32 Technical Documentation [En línea]. 2024. Disponible en: https://www.espressif.com/en/products/socs/esp32
[32] Espressif Systems. ESp32-wroom-32 [En línea]. 2024. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
[33] Electronics Hub. ESP32 Pinout [En línea]. 2024. Disponible en: https://www.electronicshub.org/esp32-pinout/

Agradecimientos:

Quiero agradecer a Hugo Alberto Murillo Hoyos por su apoyo incondicional para llevar a cabo este trabajo, también Martin Alonso Tamayo Vélez por su valiosa asesoría a lo largo del desarrollo de este proyecto y de manera especial a David Velásquez Rendon cuya ayuda y guía fueron fundamentales para la realización de este proyecto.