

INTERNET DE LAS COSAS (IoT) CON PYTHON Y RASPBERRY PI

Prof. César Rodríguez

Índice general

1. INTRODUCCIÓN	2
1.1. Estructura del Proyecto por Etapas	3
1.2. Resultados del Proyecto	3
1.3. GUIA PRÁCTICA	4
1.4. PLATAFORMA IoT EN LA NUBE	4
1.5. ENFOQUE BASADO EN LA RESOLUCIÓN DE PROBLEMAS	5
1.6. TECNOLOGIAS CLAVES	6
1.6.1. PYTHON	6
1.6.2. RASPBERRY PI	7
1.6.3. FLASK	7
1.6.4. AWS	8
1.6.5. PUBNUB	9

Capítulo 1

INTRODUCCIÓN

En las últimas décadas, el desarrollo tecnológico ha impulsado transformaciones profundas en la forma en que las personas interactúan con su entorno. Uno de los avances más significativos en este proceso es el Internet de las Cosas (IoT, por sus siglas en inglés), concepto que hace referencia a la interconexión de dispositivos físicos a través de redes digitales con el fin de recopilar, procesar y compartir información en tiempo real. Esta tendencia ha abierto un amplio abanico de oportunidades en diversos sectores como la salud, la industria, la educación, el transporte y la gestión de recursos, generando cambios en la eficiencia de los procesos y en la calidad de vida de la sociedad.

El IoT se fundamenta en la integración de sensores, actuadores, sistemas de comunicación y plataformas de análisis de datos, permitiendo que objetos cotidianos se conviertan en elementos inteligentes capaces de interactuar entre sí y con los usuarios. Su implementación ofrece beneficios como la automatización de tareas, optimización en la toma de decisiones y reducción de costos operativos, además de posibilitar soluciones innovadoras para problemáticas actuales como la sostenibilidad ambiental, la movilidad urbana o la atención médica remota.

No obstante, el crecimiento del IoT también plantea retos importantes relacionados con la seguridad de la información, la interoperabilidad de los dispositivos y la gestión de grandes volúmenes de datos. Estos desafíos hacen necesario el desarrollo de propuestas que garanticen no solo la viabilidad técnica de los sistemas, sino también su confiabilidad y aceptación social.

En este contexto, el presente proyecto de tesis (**IoT con Python y Raspberry Pi**) busca orientar en el estudio y aplicación del Internet de las Cosas, analizando sus fundamentos teóricos, sus potenciales aplicaciones y las limitaciones que enfrenta en su implementación. De esta manera, se pretende aportar conocimiento que contribuya al diseño de soluciones tecnológicas innovadoras y sostenibles, alineadas con las demandas actuales de la sociedad y con las perspectivas de un futuro cada vez más digitalizado.

En un contexto más amplio de este proyecto, en esta introducción se destacan los siguientes aspectos claves:

- Naturaleza y Enfoque del Proyecto:
 - El proyecto puede tomarse como una guía práctica y completa dirigida tanto a usuarios **principiantes** o **intermedios** interesados en el Internet de las Cosas (IoT).
 - Busca un **equilibrio completo** entre el **desarrollo de dispositivos en Raspberry Pi** y el **desarrollo de servidores locales y remotos utilizando Python**.
- Objetivo Principal y Producto Final:
 - El objetivo central es construir una **plataforma en la nube en la que múltiples usuarios pueden iniciar sesión de forma segura y controlar y monitorear sus dispositivos autorizados en tiempo real**.
 - El producto final será un **'dashboard'** que muestre gráficos visuales con datos de sensores y actuadores en tiempo real, con una **infraestructura a nivel de la nube**, que permitirá una solución IoT completa desde el dispositivo hasta la nube.
- Tecnologías Claves:
 - Las **tecnologías principales** en las que se trabajará son **Python, Raspberry Pi, Flask, AWS y PubNub**.

1.1. Estructura del Proyecto por Etapas

- **Etap 1: Introducción a IoT:** Se enfoca en **entender qué es el Internet de las Cosas y sus principales componentes**, como dispositivos inteligentes, sensores y actuadores. También se tendrá como objetivo entender **cuatro importantes modelos de comunicación del Internet de las Cosas**.
- **Etap 2: Primer Proyecto IoT:** Se **ejecutarán labores prácticas** desarrollando un proyecto IoT simple con sensores mostrando datos en una aplicación web, utilizando la técnica **AJAX** para la comunicación. Aquí se construirá un **dispositivo IoT de defensa contra robos** que detecta movimiento, activa una alarma y envía alertas, permitiendo también la desactivación de la alarma.
- **Etap 3: Seguridad y Protocolos de Comunicación:** Después de analizar ventajas y desventajas del proyecto de la Sección 2, se estudiará en profundidad los **protocolos de comunicación en tiempo real y ligeros** para IoT, como **MQTT, WebSockets**, y se hará una **demonstración práctica con PubNub**. También se cubrirá **seguridad en Internet y criptografía**, incluyendo SSL/TLS y protocolos HTTP.
- **Etap 4: Reconstrucción del Proyecto con PubNub y AWS:** Se **reconstruirá el proyecto anterior utilizando PubNub como el principal protocolo de comunicación** en lugar de **AJAX**. Además, se aprenderá a **desplegar el servidor IoT en la nube de AWS**.
- **Etap 5: Implementación de Seguridad:** Se centrará en el desarrollo de las terminologías de seguridad estudiadas en la Sección 3. Esto incluye la adquisición de un **nombre de dominio personalizado y su aseguramiento con certificados SSL/TLS**, así como la implementación de una **funcionalidad de inicio de sesión de usuario segura** y el almacenamiento de detalles de usuario en una base de datos integrada. Se gestionarán las **reglas de seguridad de entrada para HTTPS en el servidor remoto de AWS**.
- **Etap 6: Conexión Segura de Usuarios y Dispositivos:** Se implementará una forma segura para que usuarios y dispositivos IoT se conecten al servidor. Se utilizará la **funcionalidad de administrador de acceso de PubNub** para que los usuarios administradores puedan **otorgar acceso de lectura y escritura en tiempo real** a usuarios no administradores y dispositivos.
- **Etap 7: Proyecto Final: Sistema de Monitoreo Remoto:** Se añadirán más sensores y actuadores para construir un **sistema de monitoreo salud/batería**, familiarizándose con convertidores digitales, interfaz periférica serial y más.

1.2. Resultados del Proyecto

- Al finalizar el proyecto, se **entenderá lo que se necesita para construir una solución IoT propia e integral**, desde la simplicidad a nivel de dispositivo hasta la complejidad de la infraestructura a nivel de la nube. Podrán expandir el proyecto añadiendo más dispositivos y funcionalidades según surjan necesidades.

En esencia, en este proyecto se estudian desde los fundamentos teóricos y el desarrollo local de dispositivos, hasta la implementación de una plataforma IoT segura, escalable y en la nube (servidor remoto), capacitandonos para abordar problemas del mundo real con un conjunto robusto de **tecnologías y conocimientos**.

1.3. GUIA PRÁCTICA

Se pretende con este proyecto (**IoT con Python y Raspberry Pi**) construir una **guía práctica y completa** para usuarios principiantes e intermedios en el campo del Internet de las Cosas (IoT), con el objetivo de que se **entiendan y resuelvan problemas del mundo real de IoT**.

En un contexto más amplio de esta **introducción**, esta guía práctica se manifiesta a través de varias secciones y proyectos claves:

- **Desarrollo de proyectos IoT:**

- La etapa 2 se centra en la **puesta en práctica** con el **desarrollo de un proyecto simple de IoT utilizando sensores que muestran datos en una aplicación web**, empleando la técnica AJAX para la comunicación cliente-servidor.
- Un ejemplo detallado de este enfoque es el **proyecto de detector de movimiento antirrobo**, donde se conecta un sensor PIR y un zumbador a una Raspberry Pi, y se escribe código Python para detectar movimiento y controlar el zumbador, además de configurar un servidor web HTTP básico.
- La etapa 4 reconstruye el proyecto anterior para utilizar PubNub como protocolo de comunicación principal y enseña **cómo desplegar un servidor IoT en la nube de AWS**.
- La etapa 7 culmina con un **último proyecto llamado sistema de monitoreo salud/batería**. En este proyecto, se añadan los sensores y actuadores necesarios para monitoria tales variable, y los participantes se familiarizan con convertidores digitales, interfaz periférica serial y más. El producto final es un **tablero de control que muestra gráficos visuales con datos de sensores y actuadores en tiempo real**.

- **Desarrollo de servidor y seguridad:**

- El proyecto proporciona un equilibrio entre el **desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos usando Python**.
- La etapa 5 se enfoca en el **desarrollo de conceptos de seguridad**, incluyendo la **obtención y protección de un dominio personalizado con certificados SSL/TLS de Let's Encrypt**. Esto implica instalar software de terceros como **Certbot** y configurar reglas de seguridad de entrada para HTTPS.
- También se implementará una **funcionalidad segura de inicio de sesión de usuario** y el almacenamiento de detalles de usuario en una base de datos integrada.
- La etapa 6 implementa una **forma segura para que los usuarios y dispositivos IoT se conecten al servidor IoT**. También utilizara la funcionalidad de **administrador de acceso de PubNub** para que los usuarios administradores puedan **otorgar acceso de lectura y escritura en tiempo real a usuarios no administradores y dispositivos**. Esto se visualizara en un **tablero de control para administradores** que listara los usuarios en línea y ofrecera botones para conceder permisos.

1.4. PLATAFORMA IoT EN LA NUBE

Se describe la **Plataforma IoT en la Nube** como un componente esencial para construir una **solución integral de IoT**, que abarca desde la simplicidad a nivel de dispositivo hasta la complejidad de la infraestructura a nivel de nube. En el contexto del **intrucción del proyecto**, se aprenderá a desarrollar una **plataforma en la nube segura (secure cloud platform)** donde múltiples usuarios pueden iniciar sesión, controlar y monitorear sus dispositivos autorizados en tiempo real. El proyecto se equilibra entre el **desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos utilizando Python**. La Plataforma IoT en la Nube se construira y se utilizará a lo largo de varias etapas:

- **Despliegue en la nube de AWS:**

- La etapa 4 se centra en la **reconstrucción del proyecto** inicial para utilizar PubNub como protocolo de comunicación principal y se enseñara **cómo desplegar un servidor IoT en la nube de AWS (deploy IoT server into AWS cloud)**.

- La configuración de seguridad para este servidor en la nube es crucial. Por ejemplo, al asegurar un dominio personalizado con certificados SSL/TLS, se destaca la necesidad de **asignar reglas de seguridad de entrada para HTTPS en el servidor remoto de AWS** para permitir la conexión en el puerto 443.

■ **Protocolos de comunicación y seguridad en la nube:**

- Las etapas 3 y 4 abordan en profundidad los **protocolos de comunicación en tiempo real y ligeros** como MQTT y WebSockets, y su implementación con **PubNub**. PubNub es descrito como un servicio de entrega sobre WebSockets, lo que sugiere su rol como una capa de comunicación gestionada en la nube para IoT.
- La etapa 5 se dedica al **desarrollo de conceptos de seguridad** en la plataforma en la nube, incluyendo:
 - La obtención y **protección de un dominio personalizado con certificados SSL/TLS de Let's Encrypt**. Esto implica la instalación de software de terceros como Certbot y la configuración de las reglas de seguridad de entrada para HTTPS en el servidor remoto.
 - La implementación de una **funcionalidad segura de inicio de sesión de usuario** y el almacenamiento de los detalles del usuario en una base de datos integrada.
- La etapa 6 implementa una **forma segura para que los usuarios y dispositivos IoT se conecten al servidor IoT**. También utiliza la funcionalidad de **administrador de acceso de PubNub (PubNub access manager functionality)** para que los usuarios administradores puedan otorgar acceso de lectura y escritura en tiempo real a usuarios no administradores y dispositivos. Esto se visualiza en un **tablero de control para administradores** que lista a los usuarios en línea y permitiera conceder permisos.

El objetivo final es que, al finalizar el proyecto, los participantes sepan **lo que se necesita para construir su propia solución IoT integral**, abarcando desde la simplicidad a nivel de dispositivo hasta la complejidad de la infraestructura a nivel de nube.

1.5. ENFOQUE BASADO EN LA RESOLUCIÓN DE PROBLEMAS

Se indica que en el proyecto 'IoT con Python y Raspberry Pi' se adopta un **enfoque basado en la resolución de problemas**, particularmente centrándose en **problemas de IoT del mundo real**. Este enfoque se enmarca en una **guía práctica completa** destinada tanto a principiantes como a usuarios intermedios. En el contexto más amplio del **Resumen del Curso**, este enfoque se desarrolla de la siguiente manera:

- **Comprensión y aplicación:** El proyecto sigue un **enfoque basado en la reconstrucción completa**. Esto significa que ayuda a los usuarios del sistema a **entender el por qué antes del cómo y qué**. Esta metodología sugiere que se busca una comprensión profunda de los desafíos antes de abordar su implementación técnica.
- **Proyectos prácticos para problemas del mundo real:**
 - La etapa 2 del proyecto se enfoca en **ensuciarse las manos** con el desarrollo de un **proyecto simple de IoT utilizando sensores que muestran datos en una aplicación web**.
 - Un ejemplo concreto de esto es el **proyecto de detector de movimiento antirrobo**, que busca crear un dispositivo de defensa contra robos que detecta movimiento, activa una alarma y envía alertas, además de permitir la desactivación de la alarma por parte del usuario.
 - La etapa 7 presenta un **último proyecto llamado sistema de monitoreo**. En este proyecto, se añaden más sensores y actuadores acordes con lo que se quiera monitorear para **construir algo significativo para un caso de uso del mundo real**. El producto final es un tablero de control que muestra gráficos visuales con datos de sensores y actuadores en tiempo real.
- **Desarrollo de soluciones integrales:** El proyecto busca un equilibrio entre el desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos usando Python. El objetivo final de este enfoque basado en la resolución de problemas es que, al finalizar el proyecto,

los participantes sepan **lo que se necesita para construir su propia solución IoT integral**, abarcando desde la simplicidad a nivel de dispositivo hasta la complejidad de la infraestructura a nivel de nube. Esto incluye el desarrollo de una plataforma en la nube segura donde múltiples usuarios pueden iniciar sesión, controlar y monitorear sus dispositivos autorizados en tiempo real.

En resumen, el 'Enfoque Basado en Resolución de Problemas' es central en el proyecto, guiando a los desarrolladores del sistema a través de la comprensión de los desafíos de IoT y proporcionándoles las herramientas y la experiencia práctica para construir soluciones funcionales y seguras para escenarios del mundo real.

1.6. TECNOLOGIAS CLAVES

1.6.1. PYTHON

Python [1] es una de las **tecnologías principales** en el proyecto 'Internet de las Cosas con Python y Raspberry Pi'. Su rol es fundamental y se equilibra entre el desarrollo a nivel de dispositivo y el desarrollo de servidores. En el contexto más amplio de las **Tecnologías Clave** del proyecto, Python se utiliza para:

- **Programación de Dispositivos IoT con Raspberry Pi:**

- Los desarrolladores aprenderán a escribir **código Python básico** para detectar señales de sensores (como el sensor PIR para movimiento) y controlar actuadores (como el zumbador). Este código permite la detección de movimiento y la activación de alarmas.
- También se utilizara Python para implementar la lógica de comunicación bidireccional, permitiendo a los usuarios interactuar con el dispositivo, como desactivar una alarma.

- **Desarrollo de Servidores Locales y Remotos:**

- Python se empleara para crear un **servidor web HTTP básico con Flask** en la Raspberry Pi, que se ejecutara en la red Wi-Fi local.
- El proyecto mantiene un **equilibrio completo entre el desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos usando Python**. Esto incluye el despliegue de un servidor IoT en la nube de AWS.

- **Funcionalidades del Servidor IoT (con Flask):**

- Dentro de la aplicación Flask, Python se utilizara para añadir funcionalidades esenciales como la provisión de detalles adicionales como el ID de usuario y la lista de usuarios en línea a la página web.
- Se implementara lógica en Python para **poblar variables con registros de usuarios en línea**, incluyendo nombres, IDs de usuario y estados de acceso de lectura y escritura.
- Python es fundamental para **crear endpoints** en la aplicación Flask para recibir solicitudes, como las de concesión de permisos.
- También se usara para manejar la **lógica de permisos de usuario**, almacenando permisos de lectura y escritura en la base de datos y llamando al servidor de PubNub para conceder acceso específico a los usuarios.
- Se asegura que los paneles de control de acceso solo sean visibles para los usuarios administradores, utilizando la ID de usuario para añadir sentencias condicionales en el código HTML a través de las plantillas Jinja, gestionadas por Python en el servidor.

En resumen, Python es una tecnología central que une tanto la programación del hardware (Raspberry Pi) como la creación de la infraestructura de software (servidores, lógica de negocio, seguridad y gestión de usuarios) que forman la solución integral de IoT del proyecto.

1.6.2. RASPBERRY PI

Se establece claramente que **Raspberry Pi** [2] es una de las **tecnologías principales** abordadas en el proyecto 'Internet de las Cosas con Python y Raspberry Pi '. Su papel es fundamental para el **desarrollo de dispositivos IoT** y para albergar servidores locales. En el contexto más amplio de las **Tecnologías Clave**, la Raspberry Pi se utilizara para:

- **Plataforma de Hardware para Dispositivos IoT:**

- El proyecto mantiene un **equilibrio completo entre el desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos usando Python**. Esto subraya su importancia como la plataforma física sobre la que se construyen las soluciones IoT.
- En la etapa 2, los desarrolladores del proyecto 'se ensuciarán las manos' con un **proyecto simple de IoT** utilizando sensores y mostrando datos en una aplicación web, con la Raspberry Pi como el cerebro del dispositivo.

- **Implementación de Proyectos Prácticos:**

- Para el **proyecto de detector de movimiento antirrobo**, la Raspberry Pi es el dispositivo central. Se conecta con el **sensor PIR (Passive Infrared)** para detectar movimiento y un **zumbador (buzzer)** como actuador para activar una alarma.
- Los desarrolladores escribirán código Python para detectar señales y controlar el zumbador, así como para añadir funcionalidad de comunicación bidireccional, permitiendo a los usuarios desactivar la alarma.
- En la etapa 7, la Raspberry Pi continuará siendo la plataforma para el **sistema de monitoreo atmosférico**, un proyecto que implica la adición de más sensores y actuadores para construir algo significativo para un caso de uso del mundo real.

- **Alojamiento de Servidores Locales:**

- La Raspberry Pi se utilizara para ejecutar un **servidor web HTTP básico con Flask** en la red Wi-Fi local. Esto permitira que los usuarios, conectados a la misma red, accedan a una página web desde sus navegadores y reciban actualizaciones en vivo del sensor.
- También se añadirá un botón en la página web para que los usuarios puedan controlar los actuadores, como desactivar el zumbador.

- **Configuración y Preparación del Entorno:**

- Para empezar, se sugiere a los desarrolladores configurar sus Raspberry Pi siguiendo las instrucciones oficiales, lo que incluye el uso de una tarjeta SD de al menos 8GB para el sistema operativo.
- Se menciona la opción de utilizar un monitor HDMI o realizar una **conexión de escritorio remoto** a la Raspberry Pi.
- Para la Raspberry Pi 3, se enfatiza la necesidad de **habilitar SSH** a través de la terminal usando `sudo raspi_config`.

En resumen, la Raspberry Pi es la **piedra angular del hardware** en este proyecto, permitiendo a los estudiantes interactuar directamente con el mundo físico a través de sensores y actuadores, y también sirve como una plataforma de bajo costo para desarrollar y probar servidores IoT a nivel local, antes de pasar a despliegues en la nube.

1.6.3. FLASK

Se indica que **Flask** [3] es una de las **tecnologías principales** utilizadas en el proyecto 'Internet de las Cosas con Python y Raspberry Pi'. Su rol es crucial para el **desarrollo de servidores IoT**, tanto a nivel local como remoto. En un contexto más amplio de las **Tecnologías Clave** del proyecto, Flask se empleara para:

- **Desarrollo de Servidores Web HTTP Locales:**

- Los desarrolladores aprenderán a escribir un **servidor web HTTP básico en Python Flask en Raspberry Pi**. Este servidor se ejecutará en la red Wi-Fi local, permitiendo que los usuarios de la misma red accedan a una página web y reciban actualizaciones en vivo de los sensores.
 - También se añadirá un botón en la página web para que los usuarios puedan controlar los actuadores, como desactivar un zumbador, comunicándose con el servidor Flask.
- **Desarrollo de Servidores Remotos y en la Nube:**
 - El proyecto mantiene un **equilibrio completo entre el desarrollo de dispositivos en Raspberry Pi y el desarrollo de servidores locales y remotos usando Python**, donde Flask juega un papel clave en la parte del servidor. Esto incluye el despliegue de servidores IoT en la nube de AWS.
 - **Implementación de la Lógica del Servidor IoT:**
 - **Provisión de Detalles al Cliente:** Flask se utilizara para pasar detalles adicionales como el ID de usuario y la lista de usuarios en línea a la página web del cliente.
 - **Población de Datos de Usuarios en Línea:** Se implementa lógica en Flask (usando Python) para crear un mapa `online_user_records` que contenga el nombre del usuario, el ID de usuario y los estados de permisos de lectura y escritura (representados como 'checked' o 'unchecked' para el HTML). Este mapa es luego enviado a las plantillas Jinja en `index.html` para poblar dinámicamente la lista de usuarios en línea.
 - **Gestión de Permisos de Usuario:**
 - Flask recibirá solicitudes de 'concesión' de permisos (grant) desde el código JavaScript del cliente a través de *endpoints* específicos.
 - La aplicación Flask verifica si la solicitud proviene de un usuario administrador antes de procesar la solicitud.
 - Se utilizará Flask para **almacenar los permisos de lectura y escritura del usuario en la base de datos y luego llamar al servidor de PubNub para conceder acceso de lectura y escritura específico a este usuario**.
 - **Control de Visibilidad del Panel de Acceso:** La aplicación Flask, al enviar el ID de usuario al cliente, permite que las plantillas Jinja utilicen sentencias condicionales (if) para que el panel de control de acceso solo sea visible para los usuarios administradores.

En síntesis, Flask es una **pieza fundamental en la arquitectura del servidor IoT** presentada en el proyecto, facilitando la comunicación entre los dispositivos, los usuarios y la infraestructura de la nube. Permite la creación de la interfaz web, la gestión de datos de usuarios, la implementación de la lógica de permisos y la interacción bidireccional, contribuyendo a la construcción de una **solución IoT segura y escalable**.

1.6.4. AWS

AWS (Amazon Web Services) [4] es reconocida como una de las **tecnologías principales** que se abordarán en el proyecto 'Internet de las Cosas con Python y Raspberry Pi'. Su rol es fundamental en el contexto más amplio de las Tecnologías Clave del proyecto, particularmente en el **desarrollo y despliegue de la infraestructura IoT en la nube**. En este contexto, AWS se utilizará para:

- **Plataforma de Despliegue en la Nube:**
 - El proyecto enseña a **desplegar el servidor IoT en la nube de AWS**. Esto complementa el desarrollo de dispositivos en Raspberry Pi y servidores locales, logrando un **'equilibrio completo'** entre ambos.
 - El objetivo es construir una **plataforma en la nube *serverless*** en la que múltiples usuarios puedan iniciar sesión de forma segura y controlar y monitorear sus dispositivos autorizados en tiempo real.
 - El producto final del proyecto incluirá una **infraestructura a nivel de la nube**.
- **Alojamiento del Servidor Remoto:**

- AWS es la plataforma donde reside el **servidor remoto** para la aplicación IoT. Se menciona específicamente una **instancia EC2**, lo que implica el uso de servicios de cómputo virtual de AWS para alojar el servidor.

■ Configuración de Seguridad en la Nube:

- Para asegurar el servidor alojado, se gestionan las **reglas de seguridad de entrada (inbound security rules)** en AWS. Específicamente, se deben **asignar reglas de seguridad de entrada para HTTPS** en la configuración de la instancia EC2 para permitir el tráfico seguro a través del puerto 443. Esto es crucial para la implementación de SSL/TLS con certificados Let's Encrypt.

En resumen, AWS es la **columna vertebral de la infraestructura en la nube** para la solución IoT del proyecto, permitiendo el despliegue de servidores remotos escalables y seguros, y facilitando la gestión de la conectividad y la seguridad a nivel de la nube para los dispositivos y usuarios.

1.6.5. PUBNUB

PubNub [5] es explícitamente una de las **tecnologías principales** que se abordarán en el proyecto 'Internet de las Cosas con Python y Raspberry Pi '. Su papel es fundamental en el contexto más amplio de las Tecnologías Clave, especialmente en la **gestión de la comunicación en tiempo real y la seguridad de acceso** en entornos IoT. En el contexto de las Tecnologías Clave, PubNub se utiliza para:

■ Protocolo de Comunicación en Tiempo Real y Ligero:

- En el proyecto se estudiará en profundidad los protocolos de comunicación en tiempo real y ligeros para Internet de las Cosas, incluyendo MQTT, WebSockets y, finalmente, realizará una **demonstración práctica con PubNub**.
- Es seleccionado como el **principal protocolo de comunicación** para un proyecto reconstruido en la etapa 4, reemplazando a la técnica de *long polling* con AJAX. Esto subraya su importancia para una comunicación eficiente y reactiva en la aplicación IoT.

■ Gestión de Acceso y Permisos de Usuario:

- PubNub se utilizara para implementar una funcionalidad clave: el **administrador de acceso de PubNub**. Esta característica permitira a los usuarios administradores **otorgar acceso de lectura y escritura en tiempo real** a todos los usuarios no administradores y a los dispositivos.
- La lógica de la aplicación Flask interactúa directamente con PubNub. Después de almacenar los permisos de lectura y escritura en la base de datos, la aplicación **llama al servidor de PubNub para conceder acceso de lectura y escritura específico a este usuario**.
- La concesión de permisos de lectura y escritura a través de PubNub es el segundo paso de un proceso, siendo el primero la generación de una clave de autorización para el usuario específico y su almacenamiento en la base de datos.

■ Seguridad y Control en el Ecosistema IoT:

- Al permitir el otorgamiento de permisos de acceso en tiempo real, PubNub contribuye significativamente a un **ecosistema IoT fuerte, seguro, en tiempo real y escalable**. Esto es vital para asegurar que solo los usuarios y dispositivos autorizados puedan controlar y monitorear otros dispositivos.

En síntesis, PubNub es una **tecnología crucial** que no solo facilita la comunicación en tiempo real de baja latencia en el sistema IoT, sino que también es instrumental en la implementación de un **modelo de seguridad robusto** mediante la gestión dinámica de permisos de acceso, lo que es esencial para construir una **plataforma IoT multiusuario y segura**.

Bibliografía

- [1] Python Software Foundation, “Python Language Reference.” <https://www.python.org>, 2023. Accessed: 2023-10-27.
- [2] Raspberry Pi Foundation, “Raspberry Pi Documentation.” <https://www.raspberrypi.org/documentation/>, 2023. Accessed: 2023-10-27.
- [3] Pocoo, “Flask - Web Development, One Drop at a Time.” <https://flask.palletsprojects.com/>, 2023. Accessed: 2023-10-27.
- [4] Amazon Web Services, Inc., “What is AWS?.” <https://aws.amazon.com/what-is-aws/>, 2023. Accessed: 2023-10-27.
- [5] PubNub Inc., “Real-time Communication APIs and Platform.” <https://www.pubnub.com/>, 2023. Accessed: 2023-10-27.