I+IoT RLab: Diseño y desarrollo de un laboratorio remoto de internet industrial de las cosas

Sebastián Tobar[1], Mariano Zapata[1,2],
Ana Laura Diedrichs[1], Cristian Pérez Monte[1],
Gustavo Mercado[1] y Carlos Taffernaberry[1]

[1] gridTICs, Ingeniería Electrónica, Facultad Regional Mendoza, UTN
{gmercado, cartaffe, ana.diedrichs, mstobar,cfperez}@frm.utn.edu.ar
1

[2] Ingeniería Electrónica, Facultad Regional Mendoza, UTN
mariano.zapata@alumnos.frm.utn.edu.ar

Abstract. Las redes de Internet de las Cosas (IoT) e Internet de las Cosas Industriales (IIoT) han tenido un gran desarrollo en los últimos años debido a su amplio campo de uso, su gran variedad y su simplicidad en su desarrollo. En el presente trabajo se propone el desarrollo de un laboratorio remoto para enseñar, estudiar y desarrollar las técnicas, arquitecturas y protocolos de sistemas de IIoT de extremo a extremo (end to end), es decir, se podrán desarrollar sistemas que comienzan en los sensores y siguen toda la cadena de IoT hasta la computación en la nube y el big data. En toda la cadena de IoT mencionada, los elementos podrán ser configurados, programados, visualizados y probados por los desarrolladores, lo que permitirá implementar un sistema integral de IIoT. En el desarrollo, se usará infraestructura virtualizada para administrar reservas y recursos de forma dinámica.

Keywords: IoT · IIoT · Raspberry Pi · 6LoWPAN · Contiki · OpenMote · Infraestructura · Proxmox

1 Introducción

Este trabajo presenta el desarrollo de un laboratorio remoto IoT cuyo objetivo es proporcionar a usuarios, universidades y pequeñas empresas interesadas en dispositivos IoT (Internet de las Cosas) e IIoT (Internet Industrial de las Cosas) un entorno integral para el estudio, la creación de prototipos y la enseñanza de técnicas avanzadas, arquitecturas innovadoras y protocolos específicos de estos dispositivos.
Este entorno educativo y de investigación está diseñado para satisfacer la necesidad de prototipado y evaluación de aplicaciones IoT, para colaborar en la formación básica en conceptos fundamentales y facilitar la aplicación práctica en proyectos complejos. Para estudiantes, ofrece herramientas y recursos que facilitan la comprensión y la experimentación con las últimas tecnologías IoT/IIoT. En el caso de las universidades, el entorno proporciona un marco estructurado para la enseñanza y la investigación, permitiendo a estudiantes y docentes explorar y desarrollar nuevas soluciones en un campo de rápida evolución.
En la sección 2 se presenta la arquitectura del I-IoT Lab donde detallamos sus

componentes principales. En la sección 3 se muestra la configuración y virtualización en el servidor. En la sección 4 se presentan las pruebas de funcionamiento del laboratorio. Finalmente en la sección 5 se listan propuestas de mejora y en la sección 6 las conclusiones.

2 Arquitectura del I-IoT Lab

En la Fig.2 se presenta la arquitectura completa del laboratorio virtual para I-IoT. En la esquina inferior derecha de la figura, se pueden ver tres placas OpenMote cc2538 [1] conformando un solo grupo. Estas están configuradas como nodos sensores (encargados de recopilar información mediante sensores y los envían a través de protocolos inalámbricos al nodo coordinador). Cada una de estas placas está conectada a un puerto serial a través del conector USB ofrecido por el mismo módulo openmote , como se muestra en la Figura 1, lo que permite acceder a ellas a través del puerto serie. Esto facilita la programación individual de cada nodo, proporcionando flexibilidad y control en los procesos de programación, compilación y carga de firmware. Los nodos están conectados a sensores y son programados utilizando el sistema operativo Contiki-NG [2], conocido por ofrecer una pila completa de protocolos TCP/IP y soporte para 6LoWPAN [3].

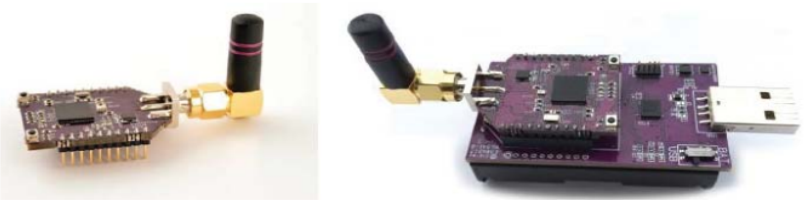

Fig. 1: A la izquierda se observa el módulo OpenMote-cc2538 utilizado, a la derecha se encuentra montado sobre la placa OpenUSB.

Además, se implementó un nodo coordinador utilizando una placa OpenMote conectada a una Raspberry Pi mediante USB, corriendo el sistema operativo Raspbian-OS [4] (el nodo coordinador es el encargado de recibir la información desde los nodos sensores y transformarla para enviarla fuera de la red WSN). Al igual que los nodos sensores, el nodo coordinador también está programado usando Contiki-NG. La comunicación entre los nodos sensores y el nodo coordinador se realiza de forma inalámbrica, empleando el estándar IEEE 802.15.4 con TSCH (Time Slotted Channel Hopping) [5] para las capas física y de enlace de datos, e IPv6 para la capa de red. El objetivo de TSCH es minimizar la imprevisibilidad del entorno inalámbrico, permitiendo así redes inalámbricas de baja potencia y alta fiabilidad, características esenciales para aplicaciones de IoT industrial. La interoperabilidad entre IEEE 802.15.4 e IPv6 se logra mediante el uso de 6LoWPAN, que facilita la conversión entre ambos protocolos.

Los demás componentes y sistemas se implementan utilizando el entorno de virtualización Proxmox [7]. Este entorno permite ejecutar máquinas virtuales, contenedores y otros recursos, y facilita la administración y el uso de los nodos según las reservas realizadas por los usuarios. Entre los componentes configurados se incluyen la gestión centralizada de usuarios a través de Zentyal, la reserva dinámica de recursos mediante Apache VCL, la conexión remota mediante un emulador de terminal como Apache Guacamole, un firewall (opnsense) y un proxy como NGINX.

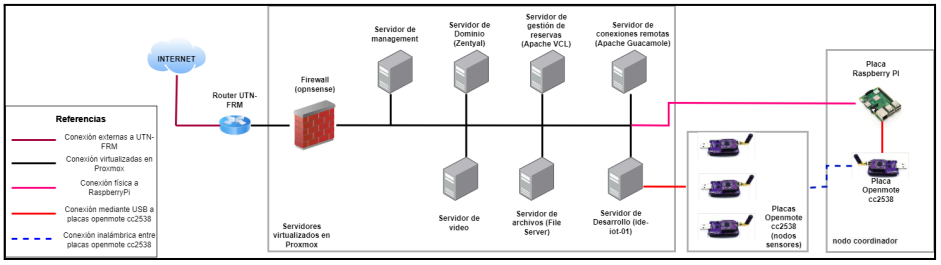

Fig. 2 Despliegue de la arquitectura creada y configurada del laboratorio remoto. Se observan los servidores virtualizados, la placa RaspberryPi, las placas IIoT openmote-cc2538 y las diferentes conexiones involucradas.

3 Implementación y configuración del servidor

A continuación se enumeran y describen los componentes utilizados para constituir el sistema completo del Laboratorio Remoto de IoT.

3.1 Entorno de programación o desarrollo

Los entornos de programación para las placas OpenMote-cc2538 y Raspberry Pi se establecieron utilizando contenedores de Linux configurados en Proxmox.

Inicialmente, los contenedores tenían restricciones de seguridad que impedían la interacción con periféricos y puertos externos. Para superar esta limitación, se modificó el archivo de configuración del contenedor, lo que permitió el acceso a los puertos seriales y de red. A través del puerto serial, los contenedores se comunican con tres placas OpenMote-cc2538 conectadas directamente, mientras que el puerto de red facilita la comunicación con una placa Raspberry Pi.

En los contenedores se instalaron el sistema operativo Contiki-NG y el compilador ARM, necesarios para la compilación y carga de firmware en las placas conectadas por USB. Para programar la placa OpenMote-cc2538 conectada a la Raspberry Pi, se utiliza una conexión SSH a la Raspberry Pi empleando las credenciales provistas por Apache VCL [8]. Esta conexión permite el acceso a una terminal de comandos, lo que facilita la programación de la placa OpenMote-cc2538 a través del puerto serie de la Raspberry Pi.

3.2 Reserva de recursos

Para lograr la reserva bajo demanda de los recursos, se utilizó la herramienta Apache VCL (Virtual Computing Lab). Esta herramienta gestiona los entornos de desarrollo ejecutando el demonio vclcliend en cada máquina administrada. Al establecerse una reserva, el nodo de gestión de VCL se conecta a través de SSH a la máquina correspondiente con el usuario «vclstaff» y almacena la información de las reservas dentro del directorio de inicio de «vclstaff». El demonio vclclientd toma esta información y ejecuta una instancia de ssh-server con la opción AllowUsers, lo que permite que solo el usuario dueño de la reserva pueda iniciar sesión.

El usuario puede elegir el tiempo de reserva, que puede ser desde 30 minutos hasta 4 horas. Si el usuario no reclama la reserva en un período de 15 minutos, esta se pierde.

3.3 Conexión remota

Para la conexión remota a las interfaces de desarrollo, se utilizó el software Apache Guacamole [9], el cual funciona como un emulador de terminal a través de un navegador web. Los usuarios disponen de un límite de hasta 8 terminales para usar de forma simultánea, permitiendo la programación en paralelo así como impresión del puerto serial, carga de firmware, captura de paquetes de red (sniffeo), entre otras propiedades.

3.4 Gestión centralizada de usuarios

Para gestionar el acceso a todas las plataformas de forma ordenada y centralizada, se utilizó Zentyal [10], una solución de software integral que ofrece funciones como controlador de dominio, firewall, servidor DNS y NTP, entre otros.

Se anexaron los contenedores de Proxmox al dominio de Zentyal, permitiendo a los usuarios utilizar sus credenciales para acceder a los entornos de desarrollo. Además, se realizaron configuraciones en Apache VCL y Apache Guacamole, modificando la autenticación de usuarios mediante LDAP para VCL y utilizando un conector para Guacamole.

De este modo, cada nuevo usuario solo necesita credenciales creadas en Zentyal para acceder a todo el sistema de laboratorio.

3.5 Captura de paquetes

Dado que el sistema realiza una comunicación inalámbrica entre las placas IIoT mediante protocolos industriales, resulta interesante, para el desarrollo del proyecto, observar el proceso de intercambio de información.

Para lograr lo anterior, el sistema operativo Contiki-NG cuenta con un sniffer de red adaptado para capturar los protocolos utilizados por las placas openmote-cc2538. Para utilizarlo, se debe compilar y cargar el firmware en una placa y, posteriormente, ejecutar un script de Python (Fig. 3). Una vez hecho esto, se observa en la terminal la captura de paquetes. Esto generará un archivo con extensión .pcap, ampliamente reconocido por herramientas de análisis de tráfico como Wireshark o NetworkMiner. En la Fig. 4 se observa el resultado de la captura de paquete de la red IIoT. Es importante destacar que en el laboratorio I-IoT no se mide el rendimiento de las

comunicaciones, sino la conectividad de las redes LP-PAN y sus protocolos.

```
Frame Stats:
            Captured: 54
          Non-Frame: 0
              Piped: 0
          Not Piped: 54
      Dumped to PCAP: 54
mariano.zapata@ide-iot-01:~/contiki-ng/tools/sensniff$
```

Fig. 3: Captura de paquetes a través de cualquier placa OpenMote mediante la ejecución del script sensniff.py

4 Pruebas de funcionamiento

Las pruebas del sistema se realizaron progresivamente durante su desarrollo, comenzando por las herramientas de reserva (Apache VCL) y el emulador de terminal (Apache Guacamole), y finalizando con la programación de las placas IIoT y la comunicación inalámbrica entre ellas.

El primer paso fue validar que un usuario creado en Zentyal pudiera iniciar sesión tanto en Apache VCL como en Apache Guacamole utilizando las credenciales creadas en Zentyal. El usuario, perteneciente al grupo «lab-iot», debe seleccionar dicho grupo en la opción «Please select an authentication method to use» e ingresar sus credenciales, como se observa en la Fig. 5.

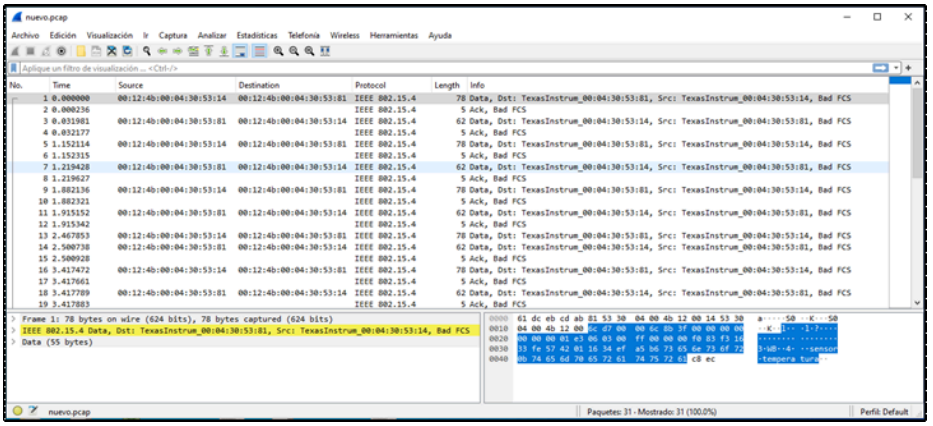

Fig. 4: Análisis de la captura de tráfico IIoT a través de Wireshark. En esta figura se verifica la conectividad entre mote mediante paquetes IEEE 802.15.4

Fig. 5: Acceso de usuario a la plataforma Virtual Computing Lab.

El siguiente punto consistió en realizar una reserva dinámica de un entorno de programación mediante Apache VCL (Fig. 6) y conectarse posteriormente a través del emulador de terminal Guacamole (Fig. 7).

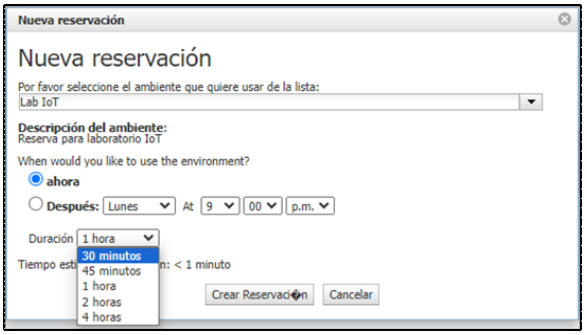

Fig. 6: Captura del alta de una nueva reserva del entorno de desarrollo (ide-iot-01) en el sistema Apache VCL.

Fig. 7: Captura de la interfaz de Apache Guacamole en la que se observa los dos entornos de desarrollo a través del emulador de terminal, en este caso el usuario solamente podrá hacer uso del entorno que reservó en el paso anterior..

Por último, se evaluó la factibilidad y la facilidad de programar las placas IIoT utilizando el emulador de terminal, en la Fig. 8 se observa el formato inicial de la terminal de desarrollo.

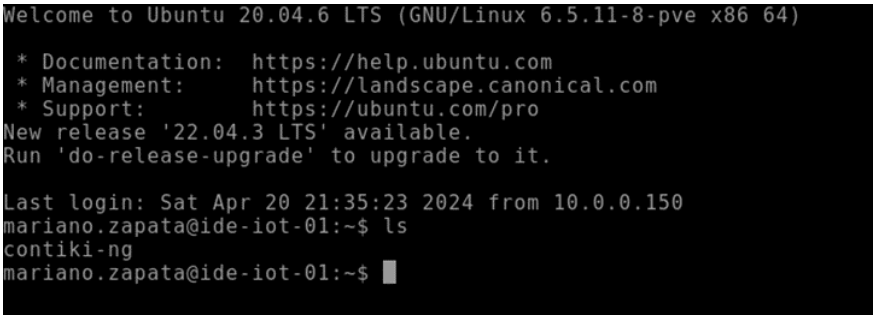

Fig. 8: Captura de la interfaz de desarrollo a través del emulador de terminal (Apache Guacamole), se observa que el usuario corresponde al nombre de quién realizó la reserva y el host que corresponde al nombre del entorno de desarrollo.

Para evaluar la programación de las placas IIoT (Fig. 9), se empezó con la compilación y ejecución del clásico ejemplo «Hola Mundo» [10] proporcionado por Contiki-NG. Utilizando editores de texto como vim o nano, se modificó el bucle de tiempo que define la demora en la impresión del mensaje «Hola Mundo» a través del puerto serie. Aunque este programa no implica la comunicación entre las placas IIoT, permitió validar que los permisos otorgados al contenedor, así como el montaje del directorio de los usuarios y los permisos de compilación, eran correctos.

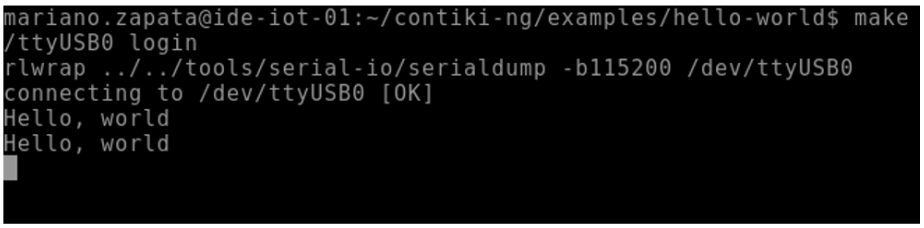

Fig. 9: Captura del resultado de la compilación y ejecución del programa “Hola Mundo” en una placa OpenMote-cc2538.

En la siguiente etapa, se desarrolló un programa para utilizar el sensor de temperatura y humedad incorporado en las placas OpenMote. Para ello, se utilizó la plantilla disponible en la carpeta de Contiki-NG, como se observa en la Fig. 10.

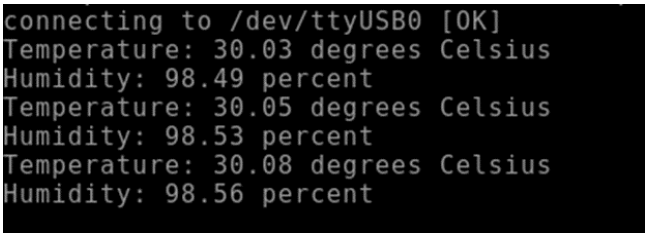

Fig. 10: Impresión de temperatura y humedad de sensor embebido en OpenMote

Posteriormente, se probó la comunicación inalámbrica entre las placas utilizando los protocolos MQTT y CoAP. Para CoAP, se configuró la placa IoT conectada a la interfaz USB de la Raspberry Pi como enrutador de borde y concentrador. Esta configuración permite comunicar dos redes aisladas y obtener información de las placas IoT utilizadas como sensores. Los datos son finalmente obtenidos a través de un script de python construido con ese objetivo, el resultado de dicho script se observa en la figura 11. Los datos obtenidos en la Raspberry Pi pueden enviarse a cualquier servicio para su posterior análisis.

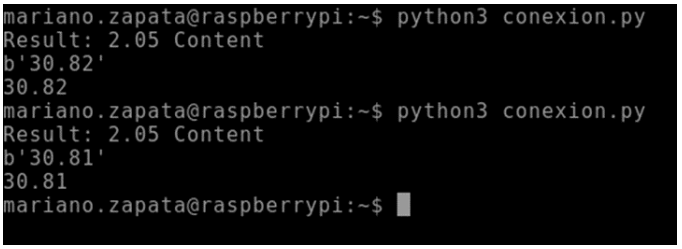

Fig. 11: Impresión de temperatura de forma remota (utilizando CoAP) desde el coordinador, en este caso representado por la placa Raspberry Pi.

Para las pruebas que utilizan el protocolo MQTT, se configuró el bróker Mosquitto en la placa Raspberry Pi, utilizando la placa IoT conectada al USB como enrutador de borde. Esta configuración permitió que las placas OpenMote-cc2538 pudieran publicar y suscribirse a temas creados en el bróker. Debido a la configuración y la ubicación del bróker en la arquitectura de red, fue necesario agregar una entrada en la tabla de enrutamiento de la Raspberry Pi. Esto garantizó que toda la información entrante con direccionamiento IPv6 desde el router de borde fuera correctamente enrutada hacia la dirección de red de búsqueda de la misma Raspberry Pi. El resultado de la comunicación puede observarse en la Fig. 12.

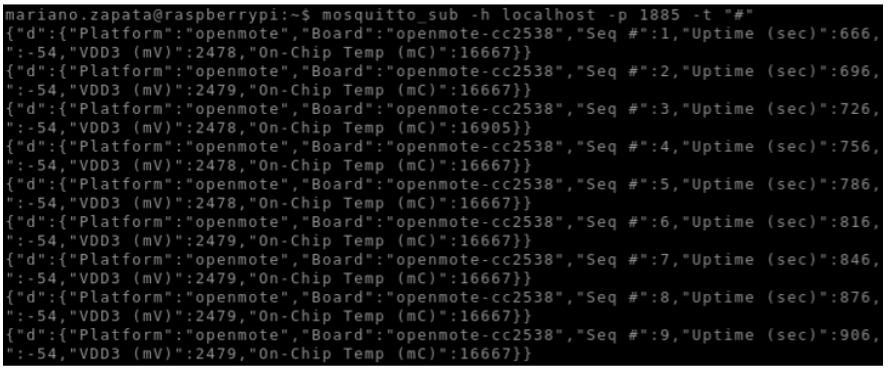

Fig. 12: Impresión de estado de placa OpenMote remota (utilizando MQTT) desde el nodo coordinador, en este caso se utilizó como broker la herramienta Mosquitto.

5 Propuestas de mejora

Aunque el laboratorio ha avanzado significativamente en la disponibilidad y el acceso a las tecnologías IIoT, así como en su programación y protocolos, se identifican algunas áreas de mejora.

Diversificación de Protocolos IoT: Actualmente, el laboratorio se centra en un conjunto específico de protocolos IoT del tipo Thread. Ampliar el espectro para incluir tecnologías como WiFi, LoRa, BLE y Zigbee permitiría a los usuarios explorar y comparar diferentes herramientas y determinar cuál se adapta mejor a sus necesidades.

Integración de Tecnologías: Además de conocer las tecnologías IIoT, el laboratorio podría beneficiarse de integrar soluciones industriales adicionales, tales como redes industriales, sistemas de control, ciberseguridad IT/OT e inteligencia artificial. Esto proporcionaría una visión más completa y práctica del ámbito industrial.

Mejora del Entorno de Desarrollo: Actualmente, el entorno de desarrollo se basa en una terminal de comandos emulada a través de un navegador web. Aunque funcional, esta configuración puede complicar la programación de las placas OpenMote-cc2538. La implementación de un entorno de desarrollo integrado (IDE) estandarizado, como Visual Studio Code o Eclipse, podría mejorar la eficiencia y facilitar la programación y compilación del firmware.

6 Conclusiones

El laboratorio ofrece a estudiantes de sistemas de IIoT, a pequeñas y medianas empresas (pymes) y al público en general la oportunidad de explorar las arquitecturas, los protocolos y la programación de dispositivos IIoT, específicamente con la tecnología de red PAN IEEE 802.15.4 y el protocolo 6lowPAN.
Permite acceder a tecnologías avanzadas con una inversión mínima, requiriendo únicamente un ordenador, una conexión a internet estable y un navegador. Además, permite interactuar con placas utilizadas en el ámbito industrial sin incurrir en el alto

coste de adquisición.
Desde una perspectiva de investigación, se ha logrado integrar múltiples tecnologías para crear un sistema coherente y funcional. Cabe destacar el desarrollo de una configuración innovadora que combina el protocolo MQTT de Contiki-NG con el bróker Mosquitto instalado en el router de borde, una configuración que no cuenta con referencias académicas previas.
En el futuro, se plantea actualizar el laboratorio a otras tecnologías de red, protocolos de comunicaciones y nuevas aplicaciones de IIoT.

7 Contexto y agradecimientos

El trabajo está incluido en el Proyecto Final de Grado «Laboratorio Remoto de IoT e IIoT, para la enseñanza, la asistencia y la consultoría a empresas», acreditado por la Cátedra Proyecto Final de la Carrera de Ingeniería de Electrónica de la UTN.
El trabajo final ha sido realizado por alumnos tesistas. Por su parte, el grupo gridTICs, que cumple funciones de *stakeholder*, se encarga de las tareas de dirección técnica, asesoramiento, financiación y provisión de laboratorios.
Se agradece la colaboración de CYTED, Grant No. 520RT0011.

8 Referencias

[1] OpenMote-cc2538. Disponible en https://doc.riot-os.org/group boards openmotecc2538.html. Julio 2023.
[2] Contiki: The Open Source OS for the Internet of Things. Disponible en https://www.contiki-os.org. Mayo 2019.
[3] Kushalnagar, N., Montenegro, G., Schumacher, C.: IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals (2007).
[4] RaspberryPi. Disponible en https://www.raspberrypi.com/software/operating-systems/. Julio 2024.
[5] Vilajosana, Xavier, et al. "OpenMote: Open-source prototyping platform for the industrial IoT." *Ad Hoc Networks: 7th International Conference, AdHocHets 2015, San Remo, Italy, September 1-2, 2015. Proceedings 7*. Springer International Publishing, 2015
[6] Molisch, A.F., Balakrishnan, K., Chong, C.C., Emami, S., Fort, A., Karedal, J., Kunisch, J.,Schantz, H., Schuster, U. and Siwiak, K.,: IEEE 802.15. 4a channel model-final report. IEEE P802, 15(04), p.0662 (2004).
[7] Proxmox. Disponible en https://www.proxmox.com/en/. Julio 2023.
[8] Apache VCL. Disponible en https://vcl.apache.org/ Julio 2023.
[9] Apache Guacamole. Disponible en https://guacamole.apache.org/releases/1.5.2/ Julio 2023.
[10] Repositorio del proyecto Industrial IoT Lab - Grupo de Investigación GridTICs. URL: https://github.com/GridTICs/iot-lab/