

SEMANA 5

Lectura Comprensiva: Introducción al Internet Industrial de las Cosas (IIoT) y Protocolos en Soluciones IoT

Tema 1: Introducción al Internet de las Cosas (IoT) e Internet Industrial de las Cosas (IIoT)

¿Qué es el Internet de las Cosas (IoT)?

El **Internet de las Cosas (IoT)** es un conjunto de dispositivos físicos interconectados que pueden recopilar, procesar y transmitir datos a través de redes como Internet. Estos dispositivos suelen incluir sensores, actuadores, microcontroladores, módulos de comunicación inalámbrica y sistemas embebidos.

¿Y qué es el Internet Industrial de las Cosas (IIoT)?

El **Internet Industrial de las Cosas (IIoT)** es una aplicación especializada del IoT en entornos industriales. Se enfoca en mejorar la eficiencia operativa, reducir costos y aumentar la productividad mediante la automatización, monitoreo remoto y análisis avanzado de datos en sectores como la manufactura, energía, transporte y logística.

TEMA 2: Capa de Software en IIoT

En el IIoT, la **capa de software** juega un papel fundamental. Es donde se programan, gestionan y controlan los dispositivos conectados. Algunas herramientas y plataformas clave son:

SDKs (Software Development Kits)

Los **SDKs** son conjuntos de herramientas que permiten desarrollar aplicaciones específicas para ciertos dispositivos o plataformas. Por ejemplo:

- ESP-IDF para programar ESP32

- Arduino IDE para trabajar con placas como Arduino Uno, Nano, etc.
- SDKs de AWS IoT o Azure IoT para conectar dispositivos a la nube

Raspberry Pi

Es una mini computadora que puede ejecutar sistemas operativos completos (como Linux) y programas complejos. Ideal para prototipado, gateways IoT o servidores locales.

FreeRTOS

Es un sistema operativo en tiempo real de código abierto diseñado para microcontroladores. Permite manejar múltiples tareas simultáneas en dispositivos con pocos recursos, ideal para IIoT.

TEMA 3: Capa de Aplicaciones en IoT

La **capa de aplicaciones** permite interactuar con los dispositivos IoT desde dispositivos móviles o web. Dos herramientas populares y accesibles son:

MIT App Inventor

Una plataforma visual para crear aplicaciones móviles Android sin necesidad de escribir código. Usa bloques lógicos para construir interfaces y funcionalidades.

Kodular

Similar a App Inventor, pero basado en la comunidad y con más opciones de integración. También permite crear apps móviles de forma visual.

Ambas herramientas son ideales para estudiantes o personas sin experiencia en programación móvil, ya que facilitan la creación de interfaces de control para dispositivos IoT.

TEMA 4: Creación de un Entorno de Comunicación Inalámbrica y IIoT

Para construir un entorno funcional de IIoT, se requiere:

Componentes básicos:

- **Sensores:** capturan información del ambiente (temperatura, humedad, movimiento, etc.)
- **Microcontrolador:** procesa los datos (Arduino, ESP8266, ESP32)
- **Módulo de comunicación inalámbrica:** transmite los datos (Wi-Fi, Bluetooth, LoRa, Zigbee)
- **Red local o cloud:** lugar donde se almacenan y analizan los datos

Flujo básico de funcionamiento:

1. El sensor recoge datos
 2. El microcontrolador procesa esos datos
 3. Se envían por Wi-Fi u otra red a un servidor local o a la nube
 4. Una aplicación o dashboard muestra los resultados o toma decisiones automáticas
-

TEMA 5: Concepto del Protocolo HTTP en Soluciones IoT

HTTP (**H**ypertext **T**ransfer **P**rotocol) es el protocolo utilizado en la web para transferir información entre clientes y servidores.

¿Cómo funciona en IoT?

Un dispositivo IoT puede actuar como **cliente HTTP**, enviando solicitudes (GET, POST, PUT, DELETE) a un servidor API RESTful. Por ejemplo:

- Un sensor manda datos vía POST a un servidor
 - Una app móvil hace una petición GET para obtener valores actualizados
-

TEMA 6: Optimización de HTTP para IoT

Dado que los dispositivos IoT tienen recursos limitados (batería, memoria), es importante optimizar el uso de HTTP.

Técnicas comunes:

- **GZIP:** comprime payloads de texto (como JSON) para reducir tamaño de datos y consumo energético

- **CBOR (Concise Binary Object Representation)**: formato binario compacto alternativo a JSON, ideal para transmisiones IoT

Estas técnicas mejoran el rendimiento y disminuyen el tráfico de red.

TEMA 7: Comunicación entre Sistemas Embebidos y Servidores API RESTful

¿Qué es una API RESTful?

Una API RESTful es una interfaz que permite la comunicación entre dos sistemas usando métodos HTTP estándar (GET, POST, PUT, DELETE). En IoT, se usan para conectar dispositivos con plataformas en la nube.

Ejemplo práctico:

Un ESP32 con sensor de temperatura puede hacer un **POST HTTP** a una URL como `https://api.ejemplo.com/medidas` para enviar los datos recolectados.

TEMA 8: Configuración de un Sistema Embebido como Soft Access Point

Un dispositivo IoT puede configurarse como un **punto de acceso Wi-Fi (Soft AP)** para facilitar su conexión inicial cuando no hay red disponible.

¿Para qué sirve?

- Permite configurar credenciales de red Wi-Fi desde una app móvil o navegador
- Útil en entornos donde no existe cobertura Wi-Fi previa

Pasos típicos:

1. Activar modo Soft AP en el dispositivo (ESP8266 / ESP32)
 2. Conectar desde un teléfono o PC al SSID generado
 3. Abrir un servidor web local
 4. Enviar credenciales de red o parámetros de configuración
-

TEMA 9: CoAP como Alternativa a HTTP en IoT

CoAP (**C**onstrained **A**pplication **P**rotocol) es un protocolo ligero diseñado específicamente para dispositivos con recursos limitados.

Características:

- Funciona sobre UDP (menos overhead que TCP)
- Soporta multicast (enviar mensaje a varios dispositivos a la vez)
- Bajo consumo de energía
- Ideal para redes de sensores remotos

Comparación rápida:

Característica	HTTP	CoAP
Transporte	TCP	UDP
Uso típico	Web general	Dispositivos IoT
Consumo	Alto	Bajo

TEMA 10: HTTP y Gateways en IoT

¿Qué es un Gateway en IoT?

Un **gateway** es un dispositivo intermediario que conecta diferentes redes o protocolos. En IoT, permite:

- Traducir protocolos locales (Modbus, MQTT, CAN) a protocolos IP (HTTP, HTTPS)
- Procesar datos antes de enviarlos a la nube (edge computing)
- Actuar como firewall o protección de red

Tipos de gateways:

- **Gateways de borde (Edge Gateways):** procesan datos localmente
 - **Gateways en la nube:** sincronizan directamente con plataformas IoT remotas
-



TEMA 11: Ejemplo Práctico de Implementación IoT

Proyecto sugerido:

Construir un sistema que mida temperatura y humedad con un sensor DHT11/22 y lo envíe a una plataforma en la nube usando HTTP.

Componentes necesarios:

- Placa ESP8266 o ESP32
- Sensor DHT11/DHT22
- Fuente de alimentación
- Conexión Wi-Fi

Flujo de trabajo:

1. Leer datos del sensor
2. Conectar a la red Wi-Fi
3. Hacer una petición POST HTTP a una plataforma como ThingSpeak o Blynk
4. Visualizar los datos en tiempo real



Recursos Adicionales

Aquí tienes algunos enlaces útiles para profundizar en estos temas:

- [Documentación oficial de HTTP – W3C](#)
- [MDN Web Docs – HTTP](#)
- [Sitio oficial de CoAP](#)
- [FreeRTOS – Documentación oficial](#)
- [MIT App Inventor](#)
- [Kodular – Plataforma de desarrollo móvil](#)
- [ThingSpeak – Plataforma IoT para visualización de datos](#)