

A. Modelo: Estructura IoT de 7 Capas

La arquitectura de siete capas del Internet de las Cosas (IoT) es un marco conceptual que divide la estructura de un sistema IoT en siete niveles distintos: Percepción, Red, Preprocesamiento, Almacenamiento, Procesamiento, Análisis y Presentación. Cada una de estas capas tiene una función específica. A continuación, se detalla cada una de estas capas:

1. Capa de Percepción

Esta es la capa más baja de la arquitectura IoT y se encarga de la recolección de datos del mundo físico. Sus componentes principales son:

- **Sensores y Dispositivos de Recolección de Datos:** Incluyen una amplia gama de sensores (temperatura, movimiento, humedad, calidad del aire, etc.) y dispositivos como cámaras y lectores RFID, que capturan datos del entorno y los convierten en información digital.
- **Actuadores:** Dispositivos que ejecutan acciones en el mundo físico según los comandos recibidos, como abrir una válvula o encender una luz.
- **Procesamiento de Bajo Nivel:** Algunos sistemas realizan un preprocesamiento de datos en esta capa, como la selección preliminar o análisis, antes de su transmisión.

2. Capa de Conectividad

Funciona como puente entre la capa de percepción y las capas superiores, transmitiendo datos y comandos. Incluye:

- **Conectividad y Protocolos de Comunicación:** Emplea tecnologías como Wi-Fi, Bluetooth, Zigbee, LoRaWAN, 5G y protocolos como MQTT, HTTP, CoAP para la transmisión de datos.
- **Seguridad de la Red:** Implementación de medidas de seguridad para proteger los datos durante su transmisión.

3. Capa de Procesamiento de Datos / Edge Computing

Se enfoca en el procesamiento inicial o de borde de los datos recopilados para reducir la latencia y mejorar la eficiencia. Incluye:

- **Preprocesamiento de Datos:** Filtración, agregación y análisis preliminar de datos en dispositivos de borde cercanos a los sensores.

4. Capa de Almacenamiento

Encargada de almacenar datos recopilados y procesados, asegurando su disponibilidad para análisis y aplicaciones futuras. Incluye:

- **Bases de Datos y Almacenamiento en la Nube:** Soluciones para guardar datos de manera segura y accesible.

5. Capa de Procesamiento en la Nube

Realiza análisis avanzados y procesamiento de grandes volúmenes de datos. Incluye:

- **Análisis de Datos y Aprendizaje Automático:** Uso de algoritmos y modelos de IA para interpretar datos y extraer insights.

6. Capa de Aplicación (análisis)

Desarrolla servicios y aplicaciones para usuarios finales, utilizando los datos procesados. Incluye:

- **Aplicaciones y Servicios de Usuario:** Aplicaciones móviles, web, dashboards para visualizar datos, recibir alertas y controlar dispositivos.
- **APIs para Integración:** Interfaces de programación que permiten la interacción y el acceso a los datos por otras aplicaciones.

7. Capa de Presentación

Diseña interfaces de usuario intuitivas y accesibles para la interacción con el sistema IoT. Incluye:

- **Diseño de Interfaz de Usuario (UI):** Creación de dashboards y aplicaciones amigables que permiten a los usuarios visualizar datos y manejar dispositivos de manera efectiva.
- **Experiencia de Usuario (UX):** Mejora la interacción del usuario con el sistema, asegurando que la presentación de datos y controles sea comprensible y útil.

Esta estructura detallada facilita la comprensión de cómo cada componente y capa contribuye al funcionamiento general de un sistema IoT, desde la captura de datos hasta la interacción del usuario con esos datos, promoviendo un diseño y desarrollo de sistemas más efectivos y centrados en el usuario.

Hoja de Ruta en el desarrollo de proyecto IoT

A continuación, la explicación de los avances de información y conocimientos que debieran incorporar los estudiantes en el desarrollo de los proyectos ABP que pronto presentaremos:

1. Capa de Percepción

Esta capa se enfoca en el inicio del proyecto modular IoT y abarca las primeras prácticas de programación. Aquí se desarrolla el sistema de sensores modulares, fundamentales para la recopilación de datos del entorno. Los estudiantes aprenden a personalizar y configurar estos sensores para recoger diversos tipos de información, dependiendo de las necesidades específicas de su proyecto.

2. Capa de Red

Esta capa incluye el aprendizaje y la implementación de una red IoT configurable, donde los datos recogidos por los sensores son transmitidos. Los estudiantes adquieren conocimientos sobre cómo configurar y gestionar esta red para asegurar una comunicación efectiva y segura entre los dispositivos del sistema IoT.

3. Capa de Preprocesamiento

En la capa de preprocesamiento, se introduce a los estudiantes al tratamiento inicial de los datos recogidos, utilizando conceptos matemáticos fundamentales como derivadas e integrales. Esta etapa es crucial para preparar los datos para análisis posteriores, mejorando la calidad de la información y facilitando su interpretación.

4. Capa de Almacenamiento

La capa de almacenamiento se centra en la programación web y el almacenamiento de datos. Los estudiantes aprenden sobre la creación de bases de datos y el manejo de datos a través de tecnologías web como HTML, CSS, SASS, Bootstrap, y JavaScript. Esta capa es esencial para la gestión eficiente de los datos recogidos y su accesibilidad para análisis y visualización.

5. Capa de Procesamiento

En esta capa, los datos son analizados y procesados en una plataforma en la nube. Los estudiantes utilizan lenguajes de programación de alto nivel como Python y Node.js para desarrollar algoritmos que permitan interpretar los datos, extraer conclusiones y tomar decisiones basadas en la información procesada.

6. Capa de (Análisis) aplicacion

Esta capa profundiza en el análisis de datos y la visualización, donde los estudiantes aplican técnicas avanzadas para interpretar los datos recogidos y presentarlos de manera comprensible. Se fomenta el desarrollo de aplicaciones y dashboards para una visualización efectiva.

7. Capa de Presentación

Finalmente, la capa de presentación se centra en el diseño de interfaces de usuario intuitivas y atractivas para sistemas IoT, utilizando principios de diseño y usabilidad. Los estudiantes aprenden a crear interfaces que

permitan a los usuarios interactuar eficientemente con el sistema IoT, mejorando la experiencia de usuario.

Conclusión

Cada una de estas capas representa una fase crítica en el desarrollo de un sistema IoT modular, permitiendo adquirir una comprensión profunda de cada aspecto involucrado en la creación de soluciones IoT integradas y funcionales.

24 proyectos ABP del internet de las cosas

Área 1: Monitoreo y Control

1. **Sistema de Monitoreo Ambiental:** Desarrollo de un sistema IoT para monitorear la calidad del aire y la temperatura en diferentes zonas de una ciudad.
2. **Control Inteligente de Iluminación:** Implementación de un sistema para controlar la iluminación de un edificio basado en la ocupación y la luz natural.
3. **Monitoreo de Consumo Energético:** Diseño de un dispositivo IoT para monitorear y reportar el consumo de energía en tiempo real.

Área 2: Salud y Bienestar

4. **Pulsera de Monitoreo de Salud:** Creación de una pulsera inteligente que monitorea signos vitales y envía alertas médicas.
5. **Sistema de Asistencia para Personas Mayores:** Desarrollo de un sistema que utiliza sensores para ayudar en el cuidado de personas mayores, detectando caídas o comportamientos inusuales.
6. **Monitoreo Remoto de Pacientes:** Implementación de un sistema para el seguimiento de pacientes con enfermedades crónicas desde su hogar.

Área 3: Agricultura y Medio Ambiente

7. **Sistema de Riego Inteligente:** Diseño de un sistema IoT para optimizar el riego en la agricultura basado en la humedad del suelo y las condiciones climáticas.
8. **Monitoreo de Fauna Silvestre:** Creación de un sistema para rastrear y monitorear animales en peligro de extinción usando sensores IoT.
9. **Gestión de Residuos Inteligente:** Desarrollo de contenedores de basura inteligentes que notifican cuando están llenos para optimizar la recogida.

Área 4: Industria y Manufactura

10. **Mantenimiento Predictivo en la Industria:** Implementación de un sistema IoT para predecir fallas en la maquinaria y planificar mantenimiento preventivo.
11. **Logística Inteligente:** Desarrollo de una solución para rastrear en tiempo real los envíos y optimizar las rutas de entrega.
12. **Seguridad Industrial Mejorada:** Creación de un sistema de sensores IoT para detectar y alertar sobre condiciones inseguras en entornos industriales.

Área 5: Hogar y Domótica

13. **Asistente Virtual para el Hogar:** Desarrollo de un sistema de asistencia basado en IoT que controla dispositivos domésticos mediante comandos de voz.
14. **Sistema de Seguridad Integrado:** Implementación de un sistema de seguridad doméstica con cámaras inteligentes, sensores de movimiento y alertas automáticas.
15. **Gestión de Energía Doméstica:** Creación de un sistema para monitorear y controlar el uso de energía en el hogar, optimizando el consumo.

16. **Automatización de Cortinas y Climatización:** Diseño de un sistema que ajusta automáticamente cortinas y climatización según la hora del día y las condiciones climáticas.

Área 6: Electrodomésticos Smart

17. **Refrigerador Inteligente:** Desarrollo de un refrigerador que monitorea su contenido, sugiere recetas y realiza pedidos automáticos de alimentos.
18. **Lavadora Inteligente con Autodiagnóstico:** Implementación de una lavadora que detecta problemas y realiza un autodiagnóstico, enviando informes al servicio técnico.
19. **Horno Inteligente con Control Remoto:** Creación de un horno que se puede controlar remotamente y ajustar según el tipo de alimento y receta.
20. **Cafetera Programable IoT:** Diseño de una cafetera que se puede programar y controlar a través de una aplicación móvil.

Área 7: Inmótica y Ciudad Inteligente

21. **Sistema de Iluminación Pública Inteligente:** Desarrollo de un sistema para controlar la iluminación en las calles, ajustándose a las condiciones de tráfico y horarios.
22. **Gestión Inteligente del Tráfico:** Implementación de un sistema para optimizar el flujo de tráfico en tiempo real utilizando sensores y datos de IoT.
23. **Sistemas de Alerta de Calidad del Aire:** Creación de una red de sensores para monitorear la calidad del aire y proporcionar alertas públicas.
24. **Red de Estaciones de Carga Inteligentes para Vehículos Eléctricos:** Diseño de una red de estaciones de carga para vehículos eléctricos con gestión inteligente y reservas a través de una aplicación.

B. Implementación General

Para plantear la implementación de un proyecto ABP (Aprendizaje Basado en Proyectos) utilizando la estructura de 7 capas del IoT de manera general y aplicable a cualquier proyecto, incluidos los 24 propuestos, y organizando las tecnologías utilizadas de forma coherente y pedagógica, se estructura la propuesta de la siguiente manera:

Título: Proyecto ABP #[Número]: [Nombre del proyecto]

¿Qué es?

[Descripción breve del proyecto, su propósito y enfoque.]

Consiste en

[Explicación detallada de lo que implica el proyecto, incluyendo los componentes, tecnologías y metodologías que se utilizarán.]

Importancia

[Descripción de la relevancia del proyecto, enfocándose en su impacto educativo, social, tecnológico, y cualquier otro ámbito relevante.]

Desarrollo etapa por etapa en estructura IoT de 7 capas:

1. Capa de Percepción:

Utilizaremos módulos de desarrollo como Arduino, ESP8266, ESP32, STM32 y Raspberry Pi para recolectar datos a través de diversos sensores. Estos dispositivos serán la base para la adquisición de datos ambientales, de usuario o de proceso, según lo requiera el proyecto.

2. Capa de Conectividad:

Para la transmisión de datos, emplearemos tecnologías inalámbricas integradas en los módulos mencionados, como Wi-Fi (ESP8266, ESP32, Raspberry Pi) y Bluetooth. Los protocolos de comunicación serán seleccionados según la necesidad de eficiencia, alcance y consumo energético del proyecto.

3. **Capa de Preprocesamiento / Edge:**

Utilizaremos Raspberry Pi y ESP32 por su capacidad de realizar procesamiento de datos cerca de la fuente. Esto incluye filtrado de datos, agregación y decisiones lógicas simples para reducir la latencia y el volumen de datos enviados a la nube.

4. **Capa de Almacenamiento:**

Para el almacenamiento de datos, se empleará MongoDB como solución de base de datos en la nube, apoyada por Docker para facilitar la implementación y escalabilidad del almacenamiento de datos recopilados.

5. **Capa de Procesamiento en la Nube:**

El procesamiento avanzado de datos se realizará en la nube, utilizando Node.js y frameworks como Express para el manejo de solicitudes y la lógica de negocio, permitiendo análisis complejos y la generación de insights a partir de los datos recogidos.

6. **Capa de Aplicación:**

Desarrollaremos aplicaciones web y móviles utilizando HTML, CSS, JavaScript y frameworks como React o Angular para la interfaz de usuario, y Node.js + Express para el backend, ofreciendo a los usuarios finales una forma intuitiva de interactuar con el sistema, visualizar datos y gestionar dispositivos.

7. **Capa de Presentación:**

La interfaz de usuario será diseñada con foco en la usabilidad y accesibilidad, empleando Bootstrap para asegurar un diseño responsivo y atractivo, facilitando la experiencia del usuario en diferentes dispositivos.

Tecnologías Involucradas:

- **Módulos de desarrollo:** Arduino, ESP8266, ESP32, STM32, Raspberry Pi

- **IDE y Frameworks:** VsCode para desarrollo general, PlatformIO para integración de dispositivos IoT, y el ecosistema Arduino para prototipado rápido.
- **Lenguajes de Programación:** Python, C++, JavaScript para el desarrollo backend y frontend.
- **Frameworks Frontend y Backend:** Bootstrap, React, Angular para el frontend; Node.js, Express para el backend.
- **Bases de Datos y Contenedores:** MongoDB para almacenamiento de datos, Docker para contenerización y despliegue de aplicaciones.

Esta estructura ofrece una guía flexible y pedagógica para el desarrollo de proyectos IoT, permitiendo adaptaciones según las necesidades específicas de cada proyecto, al tiempo que incorpora prácticas actuales de desarrollo de software y tecnologías IoT.

Detalle de Tecnologías posibles en cada capa

Para elaborar un enfoque más completo y detallado que incluya una amplia gama de tecnologías aplicables a cada una de las 7 capas del desarrollo de proyectos IoT, se presenta el siguiente desglose:

1. Capa de Percepción:

- **Sensores y Actuadores:** Variedad de sensores (temperatura, humedad, movimiento, calidad del aire, sonido, luz, etc.), actuadores (motores, LED, relés).
- **Módulos de Desarrollo:** Arduino (Uno, Mega, Nano), ESP8266, ESP32, STM32, Raspberry Pi, BeagleBone, micro:bit.
- **Cámaras:** Módulos de cámara para Raspberry Pi, cámaras IP con capacidad IoT.

2. Capa de Conectividad:

- **Protocolos de Comunicación:** Wi-Fi, Bluetooth (y BLE), Zigbee, LoRa, NB-IoT, 5G, Ethernet.
- **Módulos de Comunicación:** Módulos específicos para Arduino y Raspberry Pi para cada uno de los protocolos mencionados.

3. Capa de Preprocesamiento / Edge:

- **Plataformas de Edge Computing:** Raspberry Pi (con Docker para contenedores), NVIDIA Jetson para aplicaciones de IA, Intel NUC.
- **Sistemas Operativos:** Raspbian, Ubuntu Core, Windows 10 IoT Core, FreeRTOS para microcontroladores.
- **Frameworks de Procesamiento de Datos:** Node-RED para flujos de datos IoT, TensorFlow Lite para aplicaciones de IA en el edge.

4. Capa de Almacenamiento:

- **Bases de Datos:** MongoDB, InfluxDB (especializada en series temporales, ideal para datos IoT), MySQL, PostgreSQL.
- **Servicios de Almacenamiento en la Nube:** Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage.

5. Capa de Procesamiento en la Nube:

- **Plataformas de Nube:** AWS IoT, Google Cloud IoT, Microsoft Azure IoT, IBM Watson IoT.
- **Herramientas de Análisis de Datos:** Apache Kafka para procesamiento de flujos de datos, Apache Spark para análisis en tiempo real, Hadoop para análisis de grandes volúmenes de datos.

6. Capa de Aplicación:

- **Frameworks de Desarrollo Web:** React.js, Angular, Vue.js para el frontend; Node.js con Express, Django, Flask para el backend.

- **Plataformas de Desarrollo Móvil:** React Native, Flutter para aplicaciones móviles multiplataforma, Swift para iOS, Kotlin para Android.
- **Middleware IoT:** MQTT Broker como Mosquitto, RabbitMQ para mensajería entre dispositivos y aplicaciones.

7. Capa de Presentación:

- **Librerías de UI:** Material-UI, Bootstrap, Foundation para diseño responsivo y componentes de interfaz de usuario.
- **Herramientas de Diseño y Prototipado:** Adobe XD, Sketch, Figma para el diseño de UI/UX.
- **Frameworks CSS:** SASS, LESS para un desarrollo de estilos más eficiente.

Esta amplia selección de tecnologías ofrece a desarrolladores y docentes una rica paleta de herramientas para abordar casi cualquier desafío en el desarrollo de proyectos IoT, desde simples aplicaciones domésticas hasta complejas soluciones industriales. La elección de tecnologías específicas dependerá de los requisitos del proyecto, la experiencia del equipo de desarrollo, y los objetivos de aprendizaje en el contexto educativo.

C. Desarrollo del Proyecto (28 semanas)

Para ajustar el plan de desarrollo del proyecto al calendario académico y a la disponibilidad de las materias dictadas durante el año, consideraremos los siguientes aspectos:

- **Anuales:** Programación, Electrónica Microcontrolada, Inglés, Proyecto Integrador I.
- **1er Cuatrimestre:** Base de Datos, Ética y Deontología.
- **2do Cuatrimestre:** Sensores y Actuadores, Redes.

Plan de saberes y desarrollo del Proyecto IoT

Semana 1-4: **Capa 1 – Capa de Percepción**

- **Materias Involucradas:** Programación, Electrónica Microcontrolada.
- **Actividades Especiales:** Taller introductorio sobre Sensores y Actuadores.
- **Feedback Continuo:** Sesión de feedback al final del sprint.

Semana 5-8: **Capa 4 – Capa de Almacenamiento**

- **Materias Involucradas:** Bases de Datos.
- **Entregables en Inglés:** Documentación técnica del sprint.
- **Revisión de Conceptos Éticos:** Integración de aspectos de Ética y Deontología en el diseño de la base de datos, considerando la privacidad y seguridad de los datos.

Semana 9-12: **Capa 6 – Capa de Aplicación**

- **Materias Involucradas:** Programación.
- **Prototipado de UI/UX:** Uso de herramientas como Figma para el diseño de interfaces, fomentando la colaboración interdisciplinaria.

- **Feedback Continuo y Ajustes:** Revisión del diseño y funcionalidad de la aplicación con el grupo.

Receso Académico (Junio – Julio): Revisión y Refuerzo

- **Actividades:** Talleres de refuerzo en Programación y Electrónica Microcontrolada. Sesiones de planificación para el segundo cuatrimestre.

Semana 13-16: **Capa 2 – Capa de Conectividad** (Inicio del 2do Cuatrimestre)

- **Materias Involucradas:** Redes, Sensores y Actuadores
- **Enfoque en Colaboración:** Discusión grupal sobre la implementación de protocolos de red y seguridad.

Semana 17-20: **Capa 3 – Capa de Preprocesamiento** / Edge

- **Materias Involucradas:** Electrónica Microcontrolada, Sensores y Actuadores.
- **Presentación de Sprint en Inglés:** Exposición de los avances y desafíos encontrados, promoviendo el uso del inglés en contextos técnicos.

Semana 21-24: **Capa 5 – Capa de Procesamiento** en la Nube

- **Materias Involucradas:**
- **Integración de Conocimientos:** Aplicación de conceptos de Bases de Datos y Programación para el procesamiento de datos en la nube.

Semana 25-28: **Capa 7 – Capa de Presentación**

- **Materias Involucradas:**
- **Feedback Final y Presentación:** Presentación final del proyecto, incluyendo una demostración de la aplicación y la visualización de

datos. Sesión de feedback integral que refleje sobre el aprendizaje y los logros del equipo.

Mejoras Clave Integradas

- **Talleres Introductorios y de Refuerzo:** Facilitan la comprensión y aplicación de conceptos técnicos complejos.
- **Entregables en Inglés y Presentaciones:** Refuerzan el uso del inglés en contextos profesionales y técnicos.
- **Feedback Continuo:** Asegura la adaptabilidad del proyecto a las necesidades de aprendizaje de los estudiantes.
- **Énfasis en la Ética y Colaboración Interdisciplinaria:** Promueve la reflexión sobre el impacto social y ético de las tecnologías IoT.

Objetivos: Fomentar una experiencia de aprendizaje multifacética, preparando a los estudiantes no solo en las competencias técnicas necesarias para el desarrollo de proyectos IoT, sino también en habilidades blandas cruciales como la colaboración, comunicación y pensamiento crítico.

Detalle de los sprint

Capa 1: Capa de Percepción (sprint 1)

Objetivo de la Capa

La Capa de Percepción es el punto de entrada de los datos en cualquier sistema IoT. Utiliza sensores para recoger información del entorno y actuadores para interactuar con él. Esta capa convierte los estímulos físicos en señales digitales que pueden ser procesados por el sistema.

Tecnologías y Dispositivos Utilizados

1. Microcontroladores y Plataformas de Desarrollo:

- **Arduino (Uno, Nano, Mega):** Ampliamente usado para prototipos y proyectos educativos.

- **ESP8266 y ESP32:** Ofrecen funcionalidades de Wi-Fi integradas para proyectos IoT conectados.
- **Raspberry Pi:** Utilizado para tareas que requieren mayor capacidad de procesamiento, como procesamiento de imágenes o ejecución de algoritmos complejos.

2. **Sensores:**

- **Ambientales:** Temperatura, humedad, calidad del aire.
- **Físicos:** Movimiento (PIR), distancia (ultrasónico), luz.
- **Otros:** Sensores de peso, sensores de agua, sensores biométricos.

3. **Actuadores:**

- **Motores:** Servomotores, motores paso a paso.
- **Relés:** Para control de dispositivos de mayor potencia.
- **LEDs:** Para señalización y feedback visual.

Materias del Primer Año Involucradas

1. **Electrónica Microcontrolada:**

- Introduce los principios básicos de funcionamiento de microcontroladores, esenciales para recoger y enviar datos desde y hacia los sensores y actuadores.

2. **Programación:**

- Fundamental para desarrollar el software que opera en los microcontroladores, permitiendo la recolección de datos y el control de actuadores.

3. **Sensores y Actuadores:**

- Materia específica que abarca el uso, funcionamiento y aplicación de diversos tipos de sensores y actuadores en

proyectos prácticos, directamente relacionada con la implementación efectiva de la Capa de Percepción en proyectos IoT.

Implementación en Proyectos

- **Selección y Configuración de Sensores/Actuadores:** Basado en los requisitos del proyecto, los estudiantes eligen los dispositivos adecuados para recoger datos del entorno o interactuar con él.
- **Integración con Microcontroladores:** Conectar sensores y actuadores a microcontroladores como Arduino o ESP32, y programar la lógica necesaria para su operación.
- **Pruebas y Calibración:** Realizar pruebas para asegurar la correcta recolección de datos y la operación efectiva de los actuadores, ajustando parámetros según sea necesario.

Conclusión

La Capa de Percepción es la base sobre la cual se construyen los proyectos IoT, transformando el mundo físico en datos digitales que pueden ser utilizados por el sistema. Las materias de "Electrónica Microcontrolada", "Programación", y "Sensores y Actuadores" proporcionan los conocimientos y habilidades fundamentales para diseñar e implementar esta capa crítica, preparando a los estudiantes para enfrentar desafíos reales en el desarrollo de soluciones IoT.

Capa 4: Capa de Almacenamiento (sprint 2)

Objetivo de la Capa

La Capa de Almacenamiento es crucial para el almacenamiento seguro y eficiente de los datos recolectados y procesados por los dispositivos IoT. Esta capa garantiza que los datos estén disponibles para su posterior análisis y visualización, permitiendo tanto el almacenamiento a corto como a largo plazo en soluciones locales o en la nube.

Tecnologías y Soluciones de Almacenamiento

1. Bases de Datos NoSQL:

- **MongoDB:** Popular para almacenar grandes volúmenes de datos no estructurados o semi-estructurados de dispositivos IoT.
- **InfluxDB:** Especializada en el almacenamiento de series temporales, ideal para datos de sensores.

2. Bases de Datos SQL:

- **MySQL, PostgreSQL:** Para aplicaciones que requieren transacciones y relaciones complejas entre los datos.

3. Almacenamiento en la Nube:

- **AWS S3, Google Cloud Storage, Azure Blob Storage:** Soluciones escalables para almacenamiento de datos en la nube con alta disponibilidad y durabilidad.

4. Herramientas de Contenedores:

- **Docker:** Para desplegar y gestionar bases de datos en contenedores, facilitando la portabilidad y escalabilidad.

Materias del Primer Año Involucradas

1. Bases de Datos:

- Fundamental para entender los principios del almacenamiento de datos, diseño de esquemas de bases de datos, y la selección entre soluciones SQL y NoSQL según los requisitos del proyecto.

2. Programación:

- Proporciona conocimientos sobre cómo interactuar programáticamente con bases de datos para realizar

operaciones CRUD (Crear, Leer, Actualizar, Borrar), y cómo integrar estas operaciones en la lógica de aplicaciones IoT.

Implementación en Proyectos

- **Selección de la Base de Datos:** Los estudiantes eligen entre SQL o NoSQL basándose en la naturaleza de los datos IoT a almacenar (estructurados vs. no estructurados, necesidad de consultas complejas, etc.).
- **Configuración de Almacenamiento en la Nube:** Para proyectos que requieren escalabilidad y acceso desde cualquier lugar, se configuran soluciones de almacenamiento en la nube, aprendiendo a manejar la seguridad y el acceso a los datos.
- **Implementación de Docker para Bases de Datos:** Uso de Docker para desplegar bases de datos, lo que introduce a los estudiantes en la contenerización, un componente crítico en el desarrollo de aplicaciones modernas.
- **Integración con Aplicaciones IoT:** Desarrollan código para enviar datos desde dispositivos y sensores a las bases de datos seleccionadas, aplicando prácticas de programación segura y eficiente.

Conclusión

La Capa de Almacenamiento es vital para el éxito de cualquier proyecto IoT, ya que proporciona la infraestructura necesaria para mantener los datos accesibles y seguros a lo largo del tiempo. Mediante la materia de "Bases de Datos", los estudiantes aprenden a diseñar e implementar soluciones de almacenamiento adecuadas, mientras que en "Programación" adquieren las habilidades para integrar estas soluciones con el resto del sistema IoT, asegurando así una base sólida para el análisis y la acción basados en datos.

Capa 6: Capa de Aplicación (sprint 3)

Objetivo de la Capa

La Capa de Aplicación es donde se desarrollan las interfaces y aplicaciones que permiten a los usuarios finales interactuar con el sistema IoT. Aquí se crean soluciones que transforman los datos en información útil y accionable, ofreciendo visualizaciones, controles y análisis accesibles a través de interfaces web y móviles.

Tecnologías y Herramientas

1. Desarrollo Frontend:

- **HTML, CSS, JavaScript:** Fundamentos para crear interfaces de usuario en la web.
- **Frameworks como React, Angular, y Vue.js:** Para desarrollar aplicaciones web dinámicas y reactivas.

2. Desarrollo Backend:

- **Node.js con Express:** Popular para crear servidores web y APIs que conectan la aplicación frontend con bases de datos y servicios de procesamiento en la nube.
- **Python con Django o Flask:** Otra opción robusta para el backend, especialmente útil cuando se integran análisis de datos o inteligencia artificial.

3. Desarrollo de Aplicaciones Móviles:

- **React Native y Flutter:** Para el desarrollo de aplicaciones móviles multiplataforma que pueden compartir lógica de negocio con aplicaciones web.

4. Gestión de API:

- **Swagger o OpenAPI:** Para diseñar, documentar y consumir APIs RESTful de manera eficiente.

Materias del Primer Año Involucradas

1. **Bases de Datos:**

- Esencial para el diseño de esquemas de datos y la integración con el backend de las aplicaciones, permitiendo el manejo eficiente de los datos recopilados y procesados.

2. **Programación:**

- Proporciona los conocimientos fundamentales en lenguajes de programación y desarrollo de software necesarios para el diseño y la implementación de aplicaciones frontend y backend.

Implementación en Proyectos

- **Diseño de Interfaces de Usuario (UI):** Los estudiantes aplican principios de diseño UI/UX para crear interfaces claras y accesibles, utilizando herramientas de prototipado como Figma o Adobe XD.
- **Desarrollo Frontend:** Implementan la lógica de la interfaz de usuario utilizando HTML, CSS y JavaScript junto con frameworks avanzados para crear experiencias de usuario ricas y reactivas.
- **Construcción del Backend:** Desarrollan la lógica del servidor, manejo de API y conexiones a bases de datos utilizando Node.js/Express o Python con Django/Flask, asegurando que la aplicación pueda procesar solicitudes y entregar datos a la interfaz de usuario.
- **Integración y Pruebas:** Realizan la integración completa del sistema, conectando el frontend con el backend y servicios en la nube, y llevan a cabo pruebas exhaustivas para asegurar la funcionalidad y usabilidad de la aplicación.

Conclusión

La Capa de Aplicación es el punto de contacto entre el sistema IoT y los usuarios, transformando los datos en información valiosa y facilitando la

interacción directa con el sistema. La materia de "Programación", combinada con los fundamentos de "Bases de Datos", equipa a los estudiantes con las habilidades necesarias para desarrollar aplicaciones robustas y centradas en el usuario, fundamentales para el éxito de cualquier proyecto IoT.

Capa 2: Capa de Conectividad (sprint 4)

Objetivo de la Capa

La Capa de Conectividad es responsable de establecer la comunicación entre los dispositivos IoT y el resto del sistema, permitiendo la transmisión de datos recolectados a plataformas de almacenamiento o procesamiento, tanto local como en la nube.

Tecnologías y Protocolos de Comunicación

1. **Wi-Fi:** Utilizado para conectar dispositivos IoT a redes locales e internet, fundamental en proyectos que requieren alta velocidad de transmisión de datos.
2. **Bluetooth y BLE (Bluetooth Low Energy):** Ideal para comunicaciones a corta distancia y aplicaciones que requieren bajo consumo de energía.
3. **Zigbee y LoRa:** Protocolos enfocados en la conectividad de dispositivos IoT en redes de área amplia (WAN), útiles para aplicaciones que operan sobre grandes distancias con requerimientos de baja potencia.
4. **NB-IoT y LTE-M:** Tecnologías celulares que ofrecen conectividad a internet en áreas amplias, adecuadas para aplicaciones móviles o remotas.

Materias del Primer Año Involucradas

1. **Electrónica Microcontrolada:**

- Enseña el uso de módulos de comunicación específicos para microcontroladores, como Wi-Fi (ESP8266/ESP32) y Bluetooth, que son esenciales para la conectividad en proyectos IoT.

2. **Redes:**

- Cubre los fundamentos de las redes de computadoras, incluyendo el entendimiento de protocolos de comunicación, configuración de redes y principios de seguridad, todos cruciales para la conectividad IoT.

3. **Sensores y Actuadores:**

- Aunque esta materia se enfoca principalmente en la interacción con el entorno físico, también aborda cómo los sensores pueden comunicarse sus datos a través de diversos protocolos, integrándose en la red de dispositivos IoT.

4. **Programación:**

- Proporciona conocimientos sobre la programación de la lógica necesaria para establecer y gestionar la comunicación entre dispositivos.

Implementación en Proyectos

- **Selección de Protocolo de Comunicación:** Basándose en las necesidades del proyecto, como distancia, velocidad de transmisión de datos y consumo de energía, se elige el protocolo de comunicación más adecuado.
- **Integración de Módulos de Comunicación:** Los estudiantes implementan módulos de comunicación en sus dispositivos, asegurándose de que sean compatibles y estén correctamente configurados para transmitir datos.
- **Configuración de la Red:** Establecen la conexión a la red, aplicando conocimientos de seguridad para proteger la transmisión de datos.

- **Pruebas de Conectividad:** Realizan pruebas para asegurar una comunicación fiable entre los dispositivos IoT y la infraestructura de red o plataformas de procesamiento.

Conclusión

La Capa de Conectividad es crucial para la funcionalidad de cualquier proyecto IoT, ya que permite que los dispositivos recolectados se comuniquen eficientemente. A través de las materias de "Electrónica Microcontrolada", "Programación", "Redes", y "Sensores y Actuadores", los estudiantes adquieren una base sólida para entender y aplicar las tecnologías de comunicación necesarias en sus proyectos, garantizando así la interacción efectiva entre los componentes del sistema IoT.

Capa 3: Capa de Preprocesamiento / Edge (sprint 5)

Objetivo de la Capa

La Capa de Preprocesamiento / Edge se centra en el tratamiento inicial de los datos recogidos por los sensores antes de ser enviados a la nube o a otros sistemas para su análisis detallado. Este procesamiento preliminar incluye tareas como filtrado de datos, agregación, y análisis básico, lo que reduce la latencia y minimiza la cantidad de datos que necesitan ser transmitidos, optimizando así el uso del ancho de banda y el consumo energético.

Tecnologías y Herramientas

1. **Microcontroladores Avanzados y SBCs (Single Board Computers):**
 - **Raspberry Pi:** Capaz de ejecutar sistemas operativos completos y aplicaciones de preprocesamiento complejas.
 - **ESP32:** Ofrece capacidades de procesamiento decentes para tareas simples de preprocesamiento directamente en el dispositivo.

2. Frameworks y Herramientas de Desarrollo:

- **Node-RED:** Una herramienta de programación visual basada en flujos para conectar dispositivos IoT, APIs y servicios en línea para el preprocesamiento de datos.
- **Python con bibliotecas como NumPy y Pandas:** Utilizado en Raspberry Pi y otros dispositivos similares para realizar análisis y preprocesamiento de datos.

Materias del Primer Año Involucradas

1. Electrónica Microcontrolada:

- Introduce a los estudiantes en la programación y uso de microcontroladores avanzados y SBCs, como Raspberry Pi y ESP32, que son cruciales para las operaciones de preprocesamiento en el edge.

2. Sensores y Actuadores:

- Aunque se enfoca en la interacción con el entorno, esta materia también toca cómo los datos de sensores pueden ser procesados y utilizados localmente, lo cual es un componente importante del preprocesamiento.

3. Programación:

- Proporciona la base en lógica de programación y el uso de lenguajes y herramientas (Python, Node-RED) necesarios para implementar el preprocesamiento de datos en el dispositivo.

Implementación en Proyectos

- **Selección de Plataformas de Hardware:** Los estudiantes eligen el dispositivo más adecuado para su proyecto, considerando la complejidad del preprocesamiento requerido.

- **Desarrollo de Algoritmos de Preprocesamiento:** Implementan lógicas para filtrar, agregar y realizar análisis básicos en los datos recolectados, utilizando Python o Node-RED.
- **Optimización del Flujo de Datos:** Configuran los dispositivos para que solo los datos relevantes o resumidos sean enviados a la siguiente etapa, reduciendo el volumen de transmisión.
- **Pruebas y Ajustes:** Realizan pruebas para asegurar que el preprocesamiento sea efectivo y no descarte información crucial, ajustando los algoritmos según sea necesario.

Conclusión

La Capa de Preprocesamiento es esencial para hacer los sistemas IoT más eficientes y autónomos. A través de "Programación" y "Electrónica Microcontrolada", los estudiantes adquieren las competencias necesarias para desarrollar e implementar soluciones de preprocesamiento efectivas. Esta capa, además, facilita una introducción práctica al manejo de datos en tiempo real, preparando a los estudiantes para desafíos más complejos en el análisis y gestión de datos IoT.

Capa 5: Capa de Procesamiento en la Nube (sprint 6)

Objetivo de la Capa

La Capa de Procesamiento en la Nube centraliza el análisis avanzado y el procesamiento intensivo de datos recogidos de los dispositivos IoT. Esta capa permite realizar operaciones complejas que requieren gran capacidad de cómputo, como el análisis de grandes volúmenes de datos, aprendizaje automático, y la generación de insights valiosos y acciones automatizadas basadas en los datos.

Tecnologías y Plataformas

1. **Plataformas de Computación en la Nube:**

- **AWS (Amazon Web Services):** AWS IoT Core, Lambda para funciones sin servidor, Amazon S3 para almacenamiento.
- **Google Cloud Platform:** Google Cloud IoT Core, Google Cloud Functions, BigQuery para análisis de datos.
- **Microsoft Azure:** Azure IoT Hub, Azure Functions, Azure Cosmos DB.

2. Herramientas y Lenguajes para Análisis de Datos:

- **Node.js y Python:** Para el desarrollo de aplicaciones de backend que procesan datos. Python es particularmente popular para el análisis de datos y aprendizaje automático gracias a bibliotecas como Pandas, NumPy, y scikit-learn.
- **Apache Kafka:** Para manejar flujos de datos en tiempo real.
- **Docker y Kubernetes:** Para la contenerización y orquestación de aplicaciones, facilitando el despliegue y la escalabilidad de servicios de procesamiento de datos.

Materias del Primer Año Involucradas

1. Programación:

- Esencial para el desarrollo de la lógica de procesamiento de datos en la nube, utilizando lenguajes como Python y JavaScript (Node.js). Los estudiantes aprenden a crear aplicaciones de backend capaces de analizar y procesar datos a gran escala.

2. Bases de Datos:

- Proporciona los fundamentos necesarios para trabajar con grandes volúmenes de datos, incluyendo cómo organizar, almacenar, y acceder a los datos de manera eficiente para su análisis en la nube.

Implementación en Proyectos

- **Configuración de la Infraestructura en la Nube:** Los estudiantes configuran servicios en plataformas de nube para recibir, almacenar y procesar datos de los dispositivos IoT.
- **Desarrollo de Funciones de Análisis de Datos:** Implementan lógicas de procesamiento utilizando servicios sin servidor (como AWS Lambda o Google Cloud Functions) o contenedores (con Docker/Kubernetes), aplicando análisis de datos y posiblemente algoritmos de aprendizaje automático.
- **Integración con Bases de Datos en la Nube:** Conectan los servicios de procesamiento de datos con bases de datos en la nube para almacenar resultados de análisis y hacerlos accesibles para la capa de aplicación.
- **Pruebas y Optimización:** Realizan pruebas para asegurar la precisión del análisis de datos y optimizan el rendimiento de las aplicaciones en la nube, considerando costos, velocidad y escalabilidad.

Conclusión

La Capa de Procesamiento en la Nube es fundamental en arquitecturas IoT modernas para habilitar el análisis avanzado de datos y la toma de decisiones inteligentes a gran escala. A través de las materias de "Programación" y "Bases de Datos", los estudiantes adquieren una sólida comprensión de cómo desarrollar y desplegar soluciones de procesamiento en la nube, preparándolos para enfrentar desafíos reales en el mundo del IoT y la analítica de datos.

Capa 7: Capa de Presentación (sprint 7)

Objetivo de la Capa

La Capa de Presentación se enfoca en la visualización de datos y la interacción del usuario con el sistema IoT a través de una interfaz gráfica.

Esta capa es crucial para asegurar que la información procesada sea accesible y comprensible para los usuarios, permitiendo la toma de decisiones basada en datos y una gestión efectiva del sistema IoT.

Tecnologías y Herramientas

1. **Frameworks y Librerías de UI/UX:**

- **Bootstrap, Material-UI, Tailwind CSS:** Proporcionan componentes de interfaz de usuario prediseñados y responsivos para el desarrollo web rápido y coherente.
- **D3.js, Chart.js:** Permiten la creación de visualizaciones de datos interactivas y gráficos en aplicaciones web.

2. **Herramientas de Prototipado y Diseño:**

- **Figma, Adobe XD, Sketch:** Utilizadas para el diseño y prototipado de interfaces de usuario, facilitando la colaboración y la iteración rápida de diseños.

3. **Accesibilidad y Diseño Responsivo:**

- **Aria (Accessible Rich Internet Applications):** Asegura que las aplicaciones web sean accesibles para personas con discapacidades.
- **Pruebas de compatibilidad entre navegadores y dispositivos:** Herramientas como BrowserStack para garantizar que las aplicaciones funcionen consistentemente en diferentes entornos.

Materias del Primer Año Involucradas

Aunque el currículo presentado no especifica materias directamente relacionadas con el diseño UI/UX o desarrollo frontend avanzado, la integración de estos conceptos en el aprendizaje puede abordarse a través de:

1. **Programación:**

- Aplicación de conocimientos básicos en HTML, CSS, y JavaScript para la implementación de interfaces de usuario. La introducción a frameworks de diseño puede complementar esta formación.
- Proyectos transversales que requieren la presentación de datos y la interacción del usuario, ofreciendo una oportunidad para aplicar y expandir conocimientos de UI/UX en un contexto de proyecto real.

Implementación en Proyectos

- **Diseño de Interfaz de Usuario:** Utilizando herramientas como Figma o Adobe XD, los estudiantes conceptualizan y diseñan la interfaz de usuario, enfocándose en la usabilidad y la experiencia del usuario.
- **Implementación de la UI:** Aplican HTML, CSS y JavaScript, junto con librerías y frameworks, para transformar los diseños en interfaces web funcionales y responsivas.
- **Visualización de Datos:** Integran librerías de gráficos para presentar los datos recolectados y procesados de manera que los usuarios puedan comprender fácilmente patrones, tendencias y estadísticas.
- **Pruebas de Usabilidad y Accesibilidad:** Realizan pruebas para asegurar que la aplicación sea accesible y ofrezca una experiencia de usuario positiva en diversos dispositivos y navegadores.

Conclusión

La Capa de Presentación transforma datos complejos en información comprensible y accionable. Aunque el plan de estudios inicial no se centre explícitamente en UI/UX, la naturaleza transversal del "Proyecto Integrador I", debe proporcionar una base sólida para que los estudiantes exploren y desarrollen habilidades en diseño de interfaz y visualización de datos, fundamentales para la creación de soluciones IoT completas y centradas en el usuario.

D. Áreas de Saberes

Para asegurar una integración efectiva de los conocimientos en el desarrollo del proyecto IoT a través de la metodología de Aprendizaje Basado en Proyectos (ABP), se define claramente los objetivos de aprendizaje y los conocimientos involucrados. Comenzando con las materias anuales, aquí detallamos los conocimientos involucrados:

Proyecto Integrador I

Objetivos de Aprendizaje:

1. **Gestión de Proyectos:** Principios de gestión de proyectos, incluyendo planificación, ejecución, monitoreo y cierre de proyectos.
2. **Trabajo en Equipo y Colaboración:** Desarrollo de habilidades de trabajo en equipo, comunicación efectiva y resolución de conflictos dentro de un entorno colaborativo.
3. **Pensamiento Crítico y Resolución de Problemas:** Capacidad para analizar problemas complejos, idear soluciones innovadoras y tomar decisiones informadas.
4. **Aplicación Práctica de Conocimientos:** Integración y aplicación de conocimientos adquiridos en otras materias en el desarrollo y ejecución de un proyecto IoT completo.
5. **Presentación y Comunicación de Proyectos:** Habilidades para presentar el proyecto de manera efectiva a una audiencia técnica y no técnica, incluyendo la preparación de demos, presentaciones y documentación.
6. **Complementación:** Ampliar y/o darles foco a los saberes aprendidos en las otras materias orientados a completar los proyectos de telecomunicaciones. Cumpliendo su función de transversalidad solicitada.

Inglés

Objetivos de Aprendizaje:

1. **Comprensión Lectora Técnica:** Capacidad para entender documentación técnica, artículos científicos y manuales relacionados con el IoT y las tecnologías emergentes.
2. **Comunicación Escrita:** Habilidad para redactar documentación técnica del proyecto, incluyendo especificaciones, reportes de progreso y análisis de datos en inglés.
3. **Comunicación Oral:** Desarrollo de habilidades para presentar oralmente los avances del proyecto, explicar conceptos técnicos y participar en discusiones técnicas en inglés.
4. **Vocabulario Técnico:** Ampliación del vocabulario específico relacionado con el IoT, programación, electrónica, y campos afines, facilitando la comunicación efectiva y precisa en contextos técnicos.

Programación

Objetivos de Aprendizaje:

1. **Fundamentos de Programación:** Entendimiento profundo de conceptos básicos como variables, bucles, estructuras de control, y funciones.
2. **Lenguajes de Programación:** Competencia en lenguajes relevantes para el IoT, especialmente Python para análisis de datos, JavaScript para el desarrollo de aplicaciones web y C++ para la programación de microcontroladores.
3. **Desarrollo de Software:** Habilidades para diseñar, implementar, y depurar software que interactúe con hardware IoT, procese datos y soporte la lógica de las aplicaciones de usuario.

4. **Integración de Sistemas:** Conocimiento sobre cómo integrar diferentes componentes de software y hardware en un sistema cohesivo y funcional.
5. **Control de Versiones:** Uso eficiente de herramientas de control de versiones como Git, esencial para la colaboración en proyectos de software.

Electrónica Microcontrolada

Objetivos de Aprendizaje:

1. **Fundamentos de Electrónica:** Conocimientos básicos sobre componentes electrónicos, circuitos y principios de electrónica digital y analógica.
2. **Microcontroladores:** Uso y programación de microcontroladores (como Arduino, ESP32), incluyendo la lectura de datos de sensores y control de actuadores.
3. **Diseño de Circuitos:** Habilidades para diseñar y construir circuitos electrónicos que sirvan como base para sistemas IoT.
4. **Solución de Problemas:** Capacidad para diagnosticar y solucionar problemas de hardware, una habilidad crítica en el desarrollo de proyectos IoT.
5. **Seguridad y Eficiencia Energética:** Conciencia sobre las prácticas de diseño para asegurar la seguridad de los dispositivos y la eficiencia en el uso de la energía.

Estos objetivos de aprendizaje y conocimientos involucrados aseguran una base sólida no solo en los aspectos técnicos necesarios para el desarrollo de proyectos IoT, sino también en habilidades blandas cruciales para el éxito profesional en el ámbito de la tecnología y la ingeniería.

Continuando con la articulación de los conocimientos y objetivos de aprendizaje para las materias del primer cuatrimestre, aquí detallamos lo correspondiente a **Base de Datos** y **Ética y Deontología**:

Base de Datos

Objetivos de Aprendizaje:

1. **Fundamentos de Base de Datos:** Entender los conceptos básicos de bases de datos, incluyendo modelos de datos, esquemas y normalización.
2. **SQL y NoSQL:** Desarrollar habilidades en el manejo de bases de datos SQL (como MySQL, PostgreSQL) para almacenar y manipular datos estructurados, y NoSQL (como MongoDB) para datos no estructurados o semiestructurados.
3. **Diseño de Esquemas de Datos:** Aprender a diseñar esquemas eficientes y escalables que soporten las necesidades de almacenamiento de datos de aplicaciones IoT.
4. **Seguridad en Base de Datos:** Conocimiento sobre prácticas de seguridad para proteger los datos almacenados, incluyendo encriptación, gestión de permisos y prevención de inyecciones SQL.
5. **Integración de Datos:** Competencias para integrar bases de datos con aplicaciones de backend, utilizando APIs para el intercambio de datos entre la aplicación y la base de datos.

Ética y Deontología

Objetivos de Aprendizaje:

1. **Principios Éticos y Deontológicos:** Comprender los fundamentos éticos y deontológicos aplicables al campo de la tecnología e ingeniería, incluyendo responsabilidad profesional, confidencialidad y respeto por la privacidad.
2. **Implicaciones Éticas de la Tecnología:** Reflexionar sobre las implicaciones éticas del desarrollo y uso de tecnologías

emergentes, particularmente en lo relacionado con el IoT, la inteligencia artificial y el big data.

3. **Normativa y Legislación:** Conocimiento sobre la legislación relevante que regula el uso de tecnologías, protección de datos personales (como GDPR en Europa) y propiedad intelectual.
4. **Análisis de Casos:** Habilidad para analizar casos de estudio que presenten dilemas éticos en el contexto tecnológico, fomentando el pensamiento crítico y la toma de decisiones éticas.
5. **Desarrollo Sostenible y Tecnología:** Promover la conciencia sobre la importancia de la sostenibilidad y el impacto social de las decisiones tecnológicas, buscando soluciones que contribuyan positivamente a la sociedad y al medio ambiente.

Al integrar estos conocimientos en el desarrollo de proyectos IoT, los estudiantes no solo adquieren habilidades técnicas indispensables para el diseño y manejo de bases de datos, sino que también desarrollan una comprensión profunda de las responsabilidades éticas y legales que conlleva la creación y gestión de tecnología: “la responsabilidad social universitaria o Técnica superior”. Esta combinación de conocimientos técnicos y éticos es fundamental para formar profesionales capaces de navegar los desafíos del mundo tecnológico actual de manera responsable e informada.

Finalizamos con la integración de conocimientos y objetivos de aprendizaje para las materias dictadas en el segundo cuatrimestre, **Redes y Sensores y Actuadores**, fundamentales para el desarrollo completo del proyecto IoT:

Redes

Objetivos de Aprendizaje:

1. **Fundamentos de Redes:** Comprensión de los principios básicos de las redes de computadoras, incluyendo topologías de red, modelos OSI y TCP/IP.

2. **Protocolos de Comunicación:** Profundizar en el conocimiento de protocolos esenciales para IoT como HTTP, MQTT, CoAP, y entender sus aplicaciones, ventajas y limitaciones.
3. **Configuración de Redes:** Desarrollo de habilidades para configurar redes seguras, incluyendo la configuración de Wi-Fi y redes cableadas, VPNs, y la implementación de medidas de seguridad como firewalls y protocolos de encriptación.
4. **Análisis y Gestión de Redes:** Capacidad para diagnosticar y resolver problemas de red, así como gestionar el rendimiento de la red utilizando herramientas de monitoreo y análisis.
5. **Redes Inalámbricas y Tecnologías IoT:** Exploración de tecnologías específicas para IoT como Zigbee, LoRa, NB-IoT, enfocándose en sus casos de uso y cómo se integran en proyectos IoT.

Sensores y Actuadores

Objetivos de Aprendizaje:

1. **Tipos y Funciones de Sensores:** Identificar y comprender el funcionamiento de diversos tipos de sensores (temperatura, humedad, movimiento, luz, etc.), y su aplicación en sistemas IoT.
2. **Implementación de Actuadores:** Aprender sobre diferentes actuadores (motores, relés, LEDs) y cómo se controlan mediante señales electrónicas para realizar acciones físicas.
3. **Interfaz Sensor-Actuador con Microcontroladores:** Desarrollo de habilidades prácticas para conectar sensores y actuadores a microcontroladores (como Arduino, ESP32), incluyendo la programación necesaria para leer datos de los sensores y controlar actuadores.
4. **Procesamiento de Señales:** Entender los principios básicos detrás del procesamiento de señales recogidas por sensores, incluyendo

filtrado, conversión analógica a digital y técnicas de reducción de ruido.

5. **Aplicaciones Prácticas en Proyectos IoT:** Aplicar conocimientos sobre sensores y actuadores en la creación de prototipos funcionales que interactúen con el entorno, recogiendo datos o realizando acciones específicas basadas en esos datos.

La inclusión de estos conocimientos en el proyecto permite a los estudiantes abordar de manera integral el diseño y desarrollo de sistemas IoT, desde la comprensión y aplicación de la comunicación y la red hasta la interacción directa con el entorno físico mediante sensores y actuadores.