**1*. Investigación de las plataformas de desarrollo:***

***• Documentar las características principales de Arduino***

***(PlatformIO) y MicroPython (RT-Thread), destacando sus***

***diferencias y aplicaciones en IoT.***

**Características Principales de Arduino (PlatformIO)**

Arduino es una plataforma de código abierto basada en hardware y software fácil de usar. Se utiliza principalmente para el desarrollo de proyectos electrónicos y de IoT (Internet de las Cosas).

**Hardware**

- Placas: Arduino ofrece una variedad de placas (como Arduino Uno, Mega, Nano, etc.) que permiten conectar sensores, actuadores, y otros módulos electrónicos.

- Microcontroladores: La mayoría de las placas Arduino utilizan microcontroladores de la familia AVR (como el ATmega328) o de la familia ARM (como el SAMD21).

- Entradas/Salidas: Las placas tienen pines de entrada/salida digitales y analógicas que se pueden usar para conectar una variedad de dispositivos.

- Conectividad: Algunas placas incluyen conectividad WiFi, Bluetooth o Ethernet, lo que facilita su uso en aplicaciones IoT.

**Software**

- Arduino IDE: El entorno de desarrollo integrado (IDE) de Arduino es fácil de usar y está diseñado para principiantes. Permite escribir código en C/C++ y cargarlo en las placas.

- PlatformIO: Es una plataforma de desarrollo más avanzada que soporta múltiples entornos de desarrollo embebido, incluyendo Arduino. Se integra en IDEs como VS Code y proporciona características como gestión de bibliotecas, compatibilidad con múltiples plataformas, y herramientas de depuración.

- Bibliotecas: Arduino tiene una extensa colección de bibliotecas que simplifican la interacción con sensores, motores, módulos de comunicación, etc.

**Aplicaciones en IoT**

- Prototipos Rápidos: Arduino es ideal para crear prototipos de dispositivos IoT debido a su simplicidad y la vasta cantidad de recursos disponibles; como, por ejemplo:

- Automatización del Hogar: Se usa para desarrollar soluciones de automatización en el hogar, como sistemas de control de iluminación, temperatura, y seguridad.

- Proyectos Educativos: Su facilidad de uso lo hace popular en el ámbito educativo, donde se utiliza para enseñar conceptos básicos de electrónica y programación.

**Características Principales de MicroPython (RT-Thread)**

MicroPython es una implementación de Python 3 optimizada para correr en microcontroladores. Es útil para proyectos de IoT y aplicaciones donde se requieren recursos limitados, como memoria y potencia de procesamiento

RT-Thread es un sistema operativo en tiempo real (RTOS) que se puede usar junto con MicroPython para proyectos más complejos y de IoT que requieran un control preciso del tiempo y los recursos, sistemas industriales, automotrices, etc.

**Hardware**

- Compatibilidad: MicroPython es compatible con una amplia gama de microcontroladores, incluyendo los populares ESP32, STM32, y otros.

- RT-Thread: Este RTOS es compatible con una variedad de microcontroladores y ofrece una capa de abstracción de hardware que facilita la gestión de recursos y la implementación de sistemas multitarea.

**Software**

- Lenguaje de Programación: MicroPython permite escribir código en Python, lo que simplifica el desarrollo para aquellos familiarizados con este lenguaje. Es particularmente útil para tareas rápidas de scripting y pruebas.

- Interactividad: Ofrece un REPL (Read-Eval-Print Loop) que permite ejecutar comandos en tiempo real en el microcontrolador, facilitando la depuración y el ajuste del código.

- RT-Thread: Proporciona un entorno multitarea robusto y eficiente. Permite a los desarrolladores crear aplicaciones en tiempo real donde múltiples tareas pueden ejecutarse simultáneamente, lo cual es crucial para aplicaciones IoT complejas.

- Drivers y Componentes: RT-Thread incluye una gran cantidad de drivers y componentes para manejar dispositivos periféricos, protocolos de comunicación, y sistemas de archivos, haciendo más sencillo el desarrollo de aplicaciones embebidas complejas.

**Aplicaciones en IoT**

- Dispositivos Conectados: MicroPython es ideal para dispositivos IoT que requieren programación rápida y fácil, como sensores conectados o dispositivos simples de control.

- Desarrollo Ágil: Es excelente para desarrolladores que desean una solución ágil y flexible, donde el tiempo de desarrollo es crítico.

- Sistemas en Tiempo Real: Con RT-Thread, es posible desarrollar aplicaciones IoT que requieren una gestión precisa del tiempo y de los recursos, como sistemas industriales, automotrices, o de telecomunicaciones.

**Diferencias y Aplicaciones en IoT**

- Facilidad de Uso: Arduino es más accesible para principiantes gracias a su sencilla interfaz y comunidad extensa. MicroPython, aunque también es fácil de usar, requiere un poco más de conocimiento previo en programación.

- Lenguaje de Programación: Arduino usa C/C++, mientras que MicroPython usa Python, lo que puede ser un factor decisivo dependiendo de la preferencia o experiencia del desarrollador.

- Flexibilidad y Complejidad: PlatformIO ofrece un entorno avanzado para desarrolladores que buscan más control y flexibilidad en sus proyectos, mientras que MicroPython con RT-Thread es ideal para desarrollos rápidos que requieren capacidades en tiempo real.

- Aplicaciones IoT: Arduino es más adecuado para proyectos sencillos o prototipos, mientras que MicroPython con RT-Thread es más potente para proyectos IoT más complejos que requieren multitarea y un sistema operativo en tiempo real.

Cada plataforma tiene sus fortalezas y la elección entre ellas depende del tipo de proyecto IoT, el nivel de experiencia del desarrollador, y los requisitos específicos del sistema.

***• Investigar sobre diferentes módulos shield disponibles para***

***ESP32 y su aplicación en controladores IoT.***

El ESP32 es un microcontrolador versátil con capacidades avanzadas de WiFi y Bluetooth, lo que lo convierte en una excelente opción para proyectos de IoT. Existen varios módulos y shields disponibles que amplían sus capacidades, permitiendo a los desarrolladores implementar soluciones más complejas.  
Los Shields son una placa de circuito impreso que se coloca sobre la placa Arduino y se conecta a ella mediante el acoplamiento de sus pines sin necesidad de alguna otra conexión externa.

Algunos de los modelos más comunes de shields y sus aplicaciones en controladores IoT son:

**1. ESP32 CAM**

* **Descripción**: Un módulo ESP32 con una cámara incorporada, como la OV2640.
* **Aplicación en IoT**:
  + **Vigilancia Remota**: Utilizado para construir cámaras de seguridad conectadas a internet, permitiendo la transmisión en vivo o la captura de imágenes cuando se detecta movimiento.
  + **Reconocimiento de Imágenes**: Puede usarse en sistemas de reconocimiento facial o de objetos, donde se necesita analizar imágenes en tiempo real.

**2. ESP32 LoRa Shield**

* **Descripción**: Un shield que añade capacidades de comunicación LoRa (Long Range) al ESP32, ideal para comunicaciones de largo alcance con bajo consumo de energía.
* **Aplicación en IoT**:
  + **Redes de Sensores**: Ideal para desplegar redes de sensores distribuidos en áreas amplias, como monitoreo ambiental en zonas rurales.
  + **Agricultura Inteligente**: Utilizado en sistemas de riego y monitoreo de cultivos donde la conectividad celular o WiFi no está disponible.

**3. ESP32 OLED Display Shield**

* **Descripción**: Un shield que integra una pantalla OLED con el ESP32, normalmente de 0.96 o 1.3 pulgadas.
* **Aplicación en IoT**:
  + **Interfaces de Usuario**: Usado para mostrar datos en tiempo real, como lecturas de sensores, estado del sistema, o menús de configuración en dispositivos IoT.
  + **Dispositivos Portátiles**: Ideal para wearables o dispositivos IoT portátiles donde es necesario visualizar información directamente en el dispositivo.

**4. ESP32 Relay Shield**

* **Descripción**: Un shield que incluye relés controlados por el ESP32, lo que permite controlar dispositivos de mayor potencia.
* **Aplicación en IoT**:
  + **Automatización del Hogar**: Control de luces, electrodomésticos, o sistemas HVAC desde una aplicación móvil o un sistema automatizado.
  + **Sistemas de Control Industrial**: Permite automatizar procesos en sistemas industriales, como el encendido y apagado de motores o bombas.

**5. ESP32 Ethernet Shield**

* **Descripción**: Un shield que añade conectividad Ethernet al ESP32, permitiendo conexiones cableadas a redes locales.
* **Aplicación en IoT**:
  + **Controladores de Red Fijos**: Utilizado en entornos donde la conectividad WiFi no es estable o segura, como en sistemas de automatización industrial.
  + **Gateways IoT**: Sirve como punto de acceso para recopilar datos de varios dispositivos IoT y enviarlos a la nube a través de una conexión por cable.

**6. ESP32 Battery Shield**

* **Descripción**: Un shield que permite alimentar el ESP32 con una batería, incluyendo circuitos para la carga y gestión de energía.
* **Aplicación en IoT**:
  + **Dispositivos Portátiles**: Facilita la creación de dispositivos IoT autónomos que requieren movilidad, como sensores portátiles o dispositivos de monitoreo de salud.
  + **Sistemas Autónomos**: Ideal para proyectos IoT que necesitan operar sin conexión constante a una fuente de energía, como estaciones meteorológicas remotas.

**7. ESP32 Sensor Shields**

* **Descripción**: Shields que integran múltiples sensores (como temperatura, humedad, presión, acelerómetros, etc.) en una sola placa.
* **Aplicación en IoT**:
  + **Monitoreo Ambiental**: Permite crear estaciones de monitoreo ambiental que recopilan y envían datos sobre temperatura, humedad, presión atmosférica, etc.
  + **Sistemas de Alarma**: Usado en sistemas de seguridad para detectar cambios en el entorno, como la detección de humo, gas, o movimientos sospechosos.

**8. ESP32 GSM/GPRS Shield**

* **Descripción**: Un shield que agrega conectividad GSM/GPRS al ESP32, permitiendo la comunicación a través de redes móviles.
* **Aplicación en IoT**:
  + **Rastreo GPS**: Implementación en sistemas de rastreo GPS donde se necesita enviar la ubicación a través de redes móviles.
  + **Conectividad en Zonas Remotas**: Usado en áreas donde no hay conectividad WiFi, permitiendo que los dispositivos IoT se comuniquen a través de la red celular.

**9. ESP32 CAN Bus Shield**

* **Descripción**: Un shield que añade un controlador CAN Bus al ESP32, permitiendo la comunicación en redes de automóviles o sistemas industriales.
* **Aplicación en IoT**:
  + **Automoción**: Utilizado en vehículos para conectar el ESP32 a la red CAN Bus y monitorizar o controlar diferentes subsistemas del automóvil.
  + **Sistemas Industriales**: Implementado en maquinaria industrial que utiliza CAN Bus para la comunicación entre diferentes controladores y sensores.

**10. ESP32 GPS Shield**

* **Descripción**: Un shield que incorpora un módulo GPS para la localización geográfica.
* **Aplicación en IoT**:
  + **Rastreo de Vehículos**: Usado para desarrollar sistemas de rastreo de flotas o vehículos individuales.
  + **Proyectos de Navegación**: Implementado en dispositivos de navegación portátiles o sistemas de geolocalización para actividades al aire libre.