

Proyecto final:

Sensor de movimiento (podómetro inteligente)

25 de mayo de 2024

**María Adames, Miguel Andres, Jorge Perdomo**

### **¿Qué es un podómetro inteligente?**

Es un dispositivo que utiliza sensores para contar y rastrear los pasos dados por una persona. A diferencia de los podómetros tradicionales, los podómetros inteligentes están equipados con tecnología avanzada que les permite realizar un seguimiento más preciso de la actividad física y proporcionar una variedad de funciones adicionales. Estos dispositivos suelen estar diseñados para llevarlos en el cuerpo, como en el bolsillo, la muñeca o el cinturón.

Las características comunes de un podómetro inteligente incluyen:

1. Conteo de pasos
2. Distancia recorrida
3. Calorías quemadas
4. Conectividad
5. Notificaciones
6. Monitorización de la frecuencia cardíaca
7. GPS integrado

Los podómetros inteligentes son populares entre las personas que desean llevar un registro de su actividad física diaria, establecer objetivos de ejercicio y realizar un seguimiento de su progreso en la mejora de la salud y la forma física. También son utilizados por deportistas y atletas para realizar un seguimiento de su rendimiento y establecer registros personales. Estos dispositivos están diseñados para proporcionar una visión más completa de la actividad física y la salud en comparación con los podómetros tradicionales.

### **Elementos de construcción más comunes utilizados para fabricar estos dispositivos:**

1. **Arduino UNO R4:** Es la evolución del conocido Arduino Uno, este modelo de microcontrolador mantiene en la placa de manera integrada una amplia conectividad ya que tiene un módulo ESP32 para conectividad Wi-Fi y Bluetooth. Su distintiva matriz de LED de 12x8 hace posible prototipar visuales directamente en la placa, y con un conector Qwiic, puedes crear proyectos de estilo plug-and-play [1].
2. **ESP-32:** Altamente versátil y potente basado en el microcontrolador ESP32. Incluye dos núcleos de procesador, lo que permite la ejecución simultánea de tareas en paralelo. El módulo ESP32 ofrece soporte para Wi-Fi 802.11 b/g/n y Bluetooth 4.2, lo que permite la conexión a redes inalámbricas y la comunicación con otros dispositivos Bluetooth, lo que lo hace ideal para aplicaciones IoT (Internet de las cosas). Además de Wi-Fi y Bluetooth, el ESP32 admite una variedad de protocolos de comunicación, como SPI, I2C, UART, y más, lo que facilita la integración con otros dispositivos y sensores. Los módulos de desarrollo ESP32 suelen tener un tamaño compacto y están diseñados para ser fácilmente integrados en proyectos electrónicos (51 x 23 x 8 mm).

3. **Arduino RP2040 connect:** Lleva el microcontrolador Raspberry Pi RP2040 al factor de forma Nano. Aprovecha al máximo el núcleo dual de 32 bits Arm® Cortex®-M0+ para realizar proyectos de Internet de las Cosas con conectividad Bluetooth y WiFi gracias al módulo U-blox Nina W102. Es programable en múltiples lenguajes, incluyendo Micro Python y C/C++ [2].

#### **Sensores más utilizados para construcción de un podómetro:**

1. Módulo Acelerómetro y giroscopio MPU6050.
2. Módulo Acelerómetro ADXL335.
3. Acelerómetro integrado en el microcontrolador RP2040 LSM6DSOXTR [3].

Hemos tratado ya con todos los dispositivos enumerados en la lista anterior, y por temas de simplicidad y costo, hemos decidido utilizar como sensor principal el que viene integrado en la placa del RP2040, en caso de haya dificultades con el elemento se procederá con acelerómetro ADXL335.

#### **Módulos bluetooth más comunes:**

1. Módulo bluetooth UNO R4 (integrado).
2. Módulo bluetooth ESP32 (integrado).
3. Módulo bluetooth RP2040 (integrado) [4].

Ya que todos los microcontroladores vienen con su módulo de conectividad bluetooth integrada no tenemos que preocuparnos en escoger uno para el proyecto, en caso de que se requiera un módulo a modo de “plan B”, investigando hemos encontrado los siguientes:

1. Módulo bluetooth HC-05.
2. Módulo bluetooth HC-06.

Los cuales ofrecen las características necesarias para completar el proyecto.

#### **Costo aproximado de todos los componentes:**

ELEMENTO	PRECIO
Arduino UNO R4	B./ 25.00
Arduino RP2040 Connect	B./ 25.50
ESP32 Wroom-32 (38 pins)	B./ 5.00
MPU6050	B./ 7.00
ADXL335	B./ 5.00
HC-05	B./ 10.00
HC-06	B./ 10.00

#### **Planeación del proyecto:**

En primer lugar, vamos a emplear una aplicación desarrollada en Thunkable [5], una plataforma que facilita la creación de aplicaciones para Android de manera eficiente y simplificada. Esta aplicación se programará en un lenguaje basado en bloques, lo que permitirá establecer una conexión entre el dispositivo móvil, el Arduino y el sensor de movimiento IMU. Una vez desarrollada la aplicación, estableceremos una conexión directa con la nube, específicamente con Firebase, donde alojaremos los datos de inicio de sesión. Además, utilizaremos una base de datos alojada en Google Drive, lo que nos permitirá acceder a la información directamente desde la nube, ya sea mediante un código de Python o cualquier otra aplicación, y visualizar los datos.

Este proceso será posible gracias a que la aplicación facilita una conexión rápida entre la nube y el dispositivo móvil. Sin embargo, es importante mencionar que la única limitación es que el dispositivo móvil necesita tener acceso a una red Wifi o a Internet.

Cabe destacar que en versiones anteriores del proyecto se utilizó APP Inventor [6], pero tras una investigación exhaustiva, hemos concluido que Thunkable es una opción más actualizada y eficiente para llevar a cabo este proyecto.

### Diagrama de flujo del proyecto:

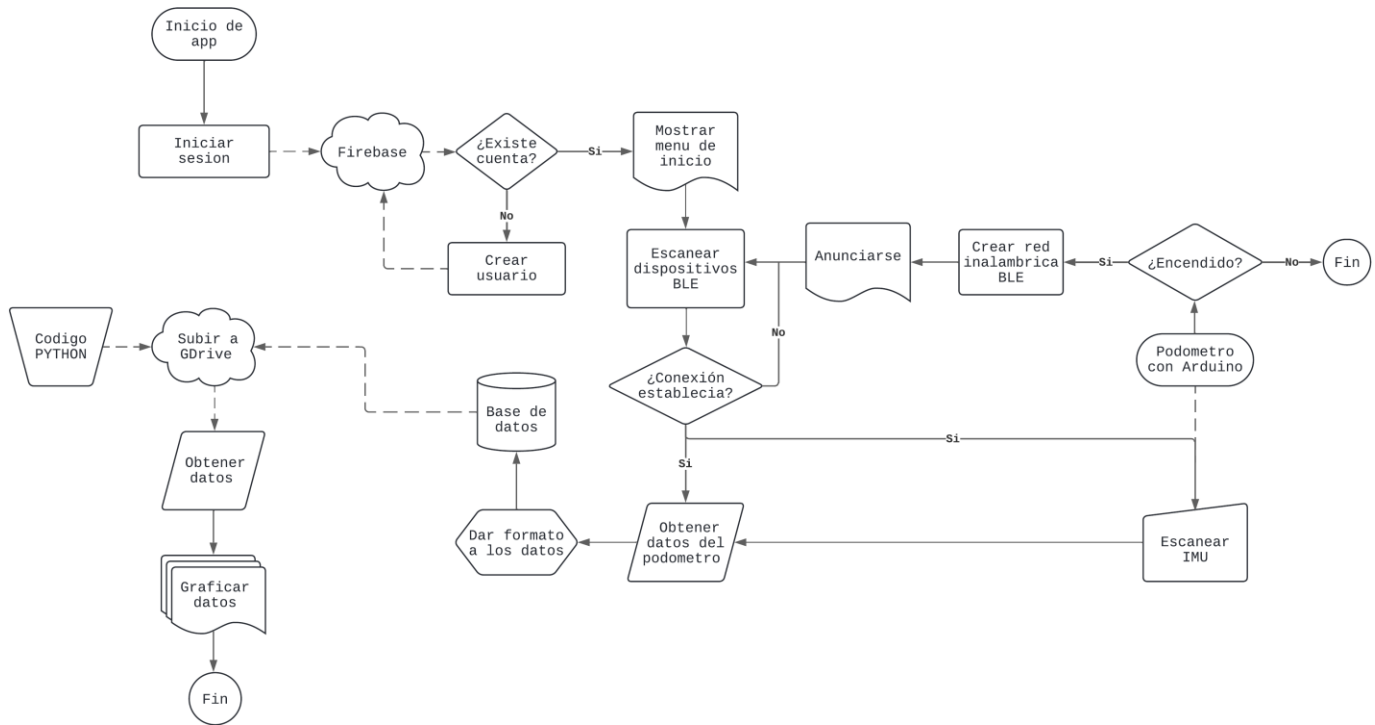
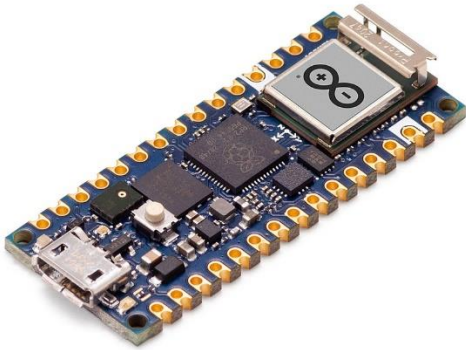


Ilustración 1: Diagrama de flujo del proyecto podómetro inteligente.

Aquí podemos observar la dirección que mantendrá el dispositivo, y también los puntos donde realizará la conexión a la nube y su comportamiento general.

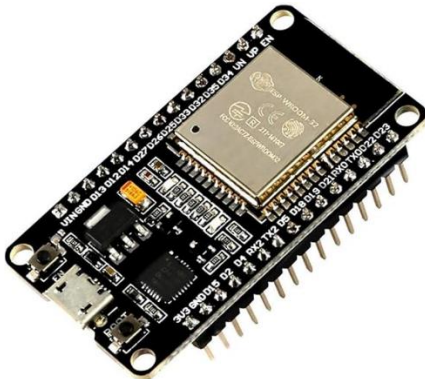
## Anexos:



*Ilustración 2: Arduino RP2040 Connect.*



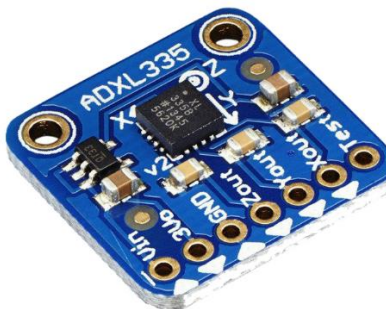
*Ilustración 3: Arduino UNO R4 Wifi.*



*Ilustración 4: ESP-32 Wroom 32.*



*Ilustración 5: Modulo IMU MPU-6050.*



*Ilustración 6: Modulo IMU ADXL335.*

## **Bibliografia:**

- [1] Arduino, “UNO R4 WiFi,” *Arduino Docs*. <https://docs.arduino.cc/hardware/uno-r4-wifi/> (accessed May 25, 2024).
- [2] Arduino, “Nano RP2040 Connect,” *Arduino Docs*. <https://docs.arduino.cc/hardware/nano-rp2040-connect/> (accessed May 25, 2024).
- [3] Arduino, “Accessing IMU Data on Nano RP2040 Connect,” *Arduino Docs*. <https://docs.arduino.cc/tutorials/nano-rp2040-connect/rp2040-imu-basics/#:~:text=The,LSM6DSOXTR,-sensor>. (accessed May 25, 2024).
- [4] Arduino, “BLE Device to Device with Nano RP2040 Connect,” *Arduino Docs*. <https://docs.arduino.cc/tutorials/nano-rp2040-connect/rp2040-ble-device-to-device/> (accessed May 25, 2024).
- [5] “Why Thinkable,” *thinkable.com*. <https://thinkable.com/why-thinkable> (accessed May 25, 2024).
- [6] Massachusetts Institute of Technology, “MIT App Inventor | Explore MIT App Inventor,” *Mit.edu*, 2019. <https://appinventor.mit.edu/>