BIG BAND

Grupo 7

Rubén Tejero
Luis Felipe
Javier Palomares
Mario Dader
Tomás Serra
Pablo De Miguel

BIG BAUD

Índice:

1. Introducción:

1.1 Motivación:

La pulsera nace con la necesidad de monitorizar el estado del cuerpo mientras se realiza esfuerzo físico y para mejorar la calidad del entrenamiento. El diseño está pensado para distintas implementaciones, ya sean médicas o deportivas.

Es un producto versátil que se amolda a las necesidades que pueda tener el usuario. Al tratarse de un producto Open Source, el propio usuario puede crear su propia implementación, dándole un gran valor añadido sobre las otras opciones del mercado.

1.2 Explicación:

La meta es crear una pulsera de actividad que integre diferentes tipos de sensores para el análisis y la monitorización la persona que la porte. En principio, está enfocado a personas deportistas, para que realicen ejercicio con un control total de su cuerpo, incluso del ambiente (Sensor UV), durante situaciones de esfuerzo. Sin embargo también puede ser utilizada para medir la actividad cotidiana y la calidad del descanso.

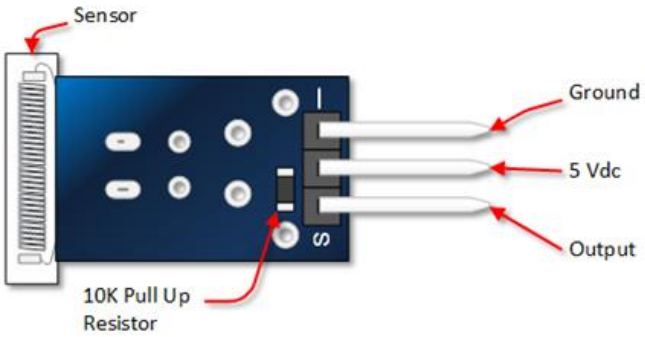
2. Producto:

2.1 Hardware:

Se van a comentar brevemente los sensores y la placa de desarrollo que se han empleado en el proyecto. La placa utilizada dispone de una pantalla OLED integrada, Wifi, Bluetooth y posibilidad de alimentación a través de batería. La mayoría de los sensores han sido elegidos por su precio asequible, su popularidad y su correcto funcionamiento. Además, existen librerías en Arduino de estos componentes que facilitan su utilización de cara a la parte software. Los sensores son los siguientes:

- <u>Sensor de impacto KY-031:</u>

Cuenta con un muelle, que al recibir un impacto, realiza contacto con uno de los extremos y mediante un montaje en pull-up (viene integrado en la PCB) genera un pulso. Es imprescindible el uso de un antirrebotes, para evitar falsos pulsos.



- Sensor de temperatura LM35:

Este detector de temperatura destaca por su linealidad en su rango de medidas (hoja de especificaciones). Cada aumento en 1 ºC supone un aumento de 10 mV de tensión. La tensión de salida va desde los -0.5 V hasta los 1.5 V (rango de temperatura desde los -50 ºC hasta los 150 ºC).

- Sensor detector UV ML8511:

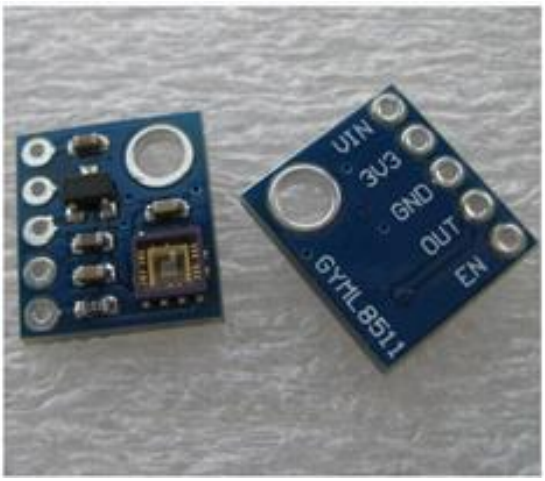
Este sensor entrega una señal analógica dependiente de la cantidad de luz ultravioleta (longitud de onda entre 280-390 nm) que recibe. Para obtener mayor precisión necesitaremos obtener también el valor de la tensión de alimentación. Por tanto, se requiere el uso de 2 pines para la lectura analógica y otros 2 para alimentación y masa.

Las lecturas analógicas nos dan un valor comprendido entre 0 y 2^n-1, donde n es el número de bits que emplea el ADC para la conversión. Por tanto, la expresión de la tensión de salida es la siguiente:

$$V_{rad} = \frac{3.3}{V_{3.3\,read}} x\ uv_{level}$$

donde 3.3 hace referencia a la tensión con que alimentamos el sensor, $V_{3.3\,read}$ al valor que proporciona el ADC al leer el pin de alimentación y uv_{level} el valor del pin de salida del sensor. Esta tensión nos va a permitir calcular la radiación en $\frac{mW}{cm^2}$. Para ello tenemos que realizar una “regla de 3” en la que especificamos tensión mínima que esperamos obtener, tensión máxima (no superior a la de alimentación del sensor), valor mínimo de radiación y valor máximo.

Por otro lado, se ha empleado una función que se encarga de promediar 8 veces la medida obtenida por el ADC. El sensor funciona correctamente en el exterior, en el interior su sensibilidad y fiabilidad decrecen hasta desaparecer. Una buena forma de calibrar el dispositivo es: salir al exterior y comprobar que los datos obtenidos se corresponden con los dados por el AEMET.



- Sensor humedad:

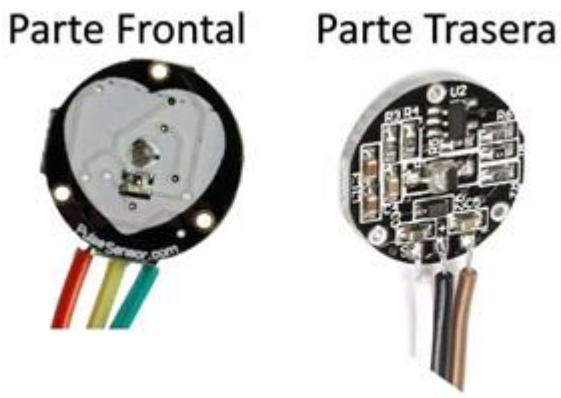
Se ha utilizado la bioimpedancia propia de la piel para implementar este sensor. Teniendo en cuenta que al sudar, la humedad hace que la bioimpedancia de la piel baje, hemos realizado un divisor de tensión con una resistencia de carga de 1 MΩ, que nos proporciona un valor de tensión correspondiente al grado de humedad en el que se encuentre la piel. Esta tensión, se obtiene mediante dos electrodos colocados en la base del dispositivo (pues es necesario un buen contacto)

- Sensor de pulso analógico

Proporciona una salida analógica que se lee mediante un ADC. El valor que se obtiene se compara con una referencia y si la supera, es que se ha producido un latido. En este caso el ADC tiene 12 bits de precisión, por lo que elegimos 2600 (generalmente se elige un valor medio como referencia). Los valores que puede arrojar el ADC van desde 0 hasta 2^n-1, siendo n el número de bits empleados por el ADC.

La medida que proporciona el dispositivo es la cantidad de pulsos en 6 segundos y multiplicar por 10.

Debido a que no hay código valido online para la implementación, se adjunta en el anexo el código empleado.



Las partes del integrado son 3: una para la salida analógica(S) otra para la alimentación(+) y finalmente otra para la masa(-). Si miramos la parte trasera su distribución es la siguiente: [S| + | -].

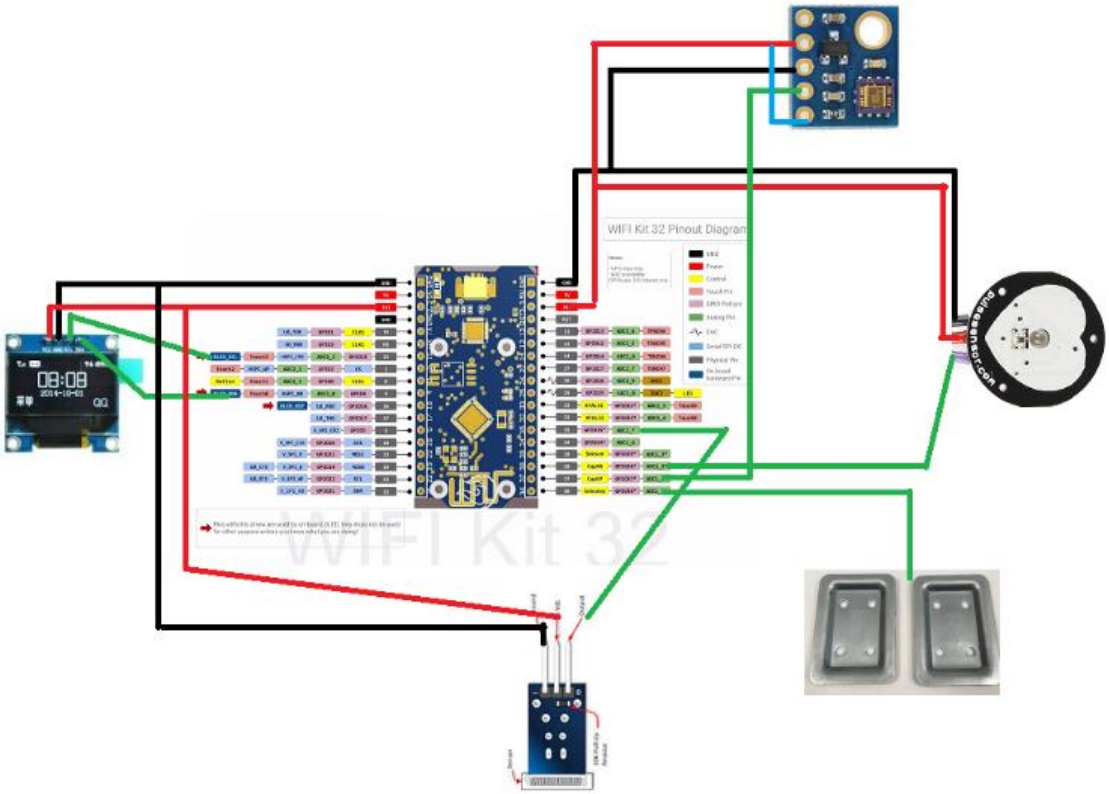
- Placa microcontrolador ESP32

Se seleccionó un ESP32 (escogimos un modelo con pantalla OLED y adaptador para batería integrado) debido a su superioridad frente a un ESP8266.Se puede programar en C y es posible utilizar FreeRTOS en caso de que el sistema requiera medidas en tiempo real. La documentación disponible es de mayor calidad y está muy extendido su uso para temas de IoT. Además es posible conectarlo a través de Wi-Fi o por Bluetooth. Esto nos permitió desarrollar una aplicación móvil que consumía los datos producidos por los sensores y que se comunicaba con el micro a través de Bluetooth.

También cuenta con un modo ahorro de energía que no hemos llegado a tocar en el proyecto pero sin lugar a dudas representaría una mejora sustancial respecto a la duración de la batería.

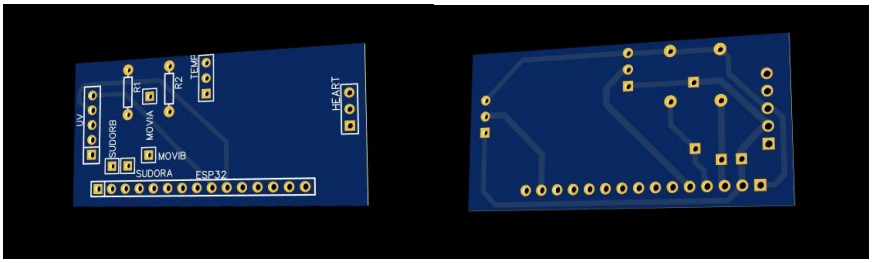
A continuación se presenta un enlace que sirve como guía de inicio: https://robotzero.one/heltec-wifi-kit-32/

Por último, se muestra una imagen del dispositivo y de las funcionalidades de sus pines y las conexiones de los sensores con la placa:



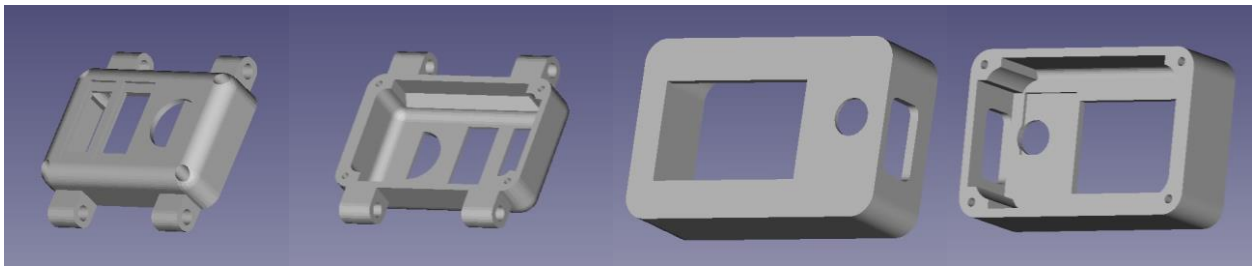
- Batería LiPo recargable 500 mAh
- PCB

Con el fin de integrar los componentes en la carcasa de la manera más discreta posible, se procedió a fabricar una placa de circuito impreso con las conexiones adecuadas, donde posteriormente se soldaron cada uno de los componentes expuestos anteriormente. A continuación se muestran los planos de la PCB y el resultado final.



- Carcasa y correas

La carcasa de la pulsera se diseñó acorde con las dimensiones de los componentes electrónicos y de su colocación según las posiciones más óptimas para obtener parámetros corporales. Para producir la carcasa mediante impresión 3D se utilizó como material PLA.



Finalmente para su ajuste adecuado a la muñeca se añadió una correa elástica con cierre de Velcro.

2.3 Funcionamiento y posibles utilidades de los sensores

Sensor de impacto: permite cambiar de pantalla al usuario. Es una forma eficiente de evitar cambios de pantalla innecesarios y por tanto de malgastar la batería.

Detector sudoración/humedad: Modo deportivo: Si el sensor detecta un grado de humedad en nuestra muñeca superior a cierto valor, se pondrá a uno y mostrará un mensaje por pantalla tal que: “Buen ejercicio/esfuerzo, ¡sigue así!”

Sensor de rayos UV: detectará el índice de radiación solar en la ubicación donde nos encontremos en un momento dado. En función de este y de la temperatura ambiente (SENSOR DE TEMPERATURA), podremos determinar un mensaje en función del riesgo que tiene un usuario de quemarse o no con la luz solar.

Pulsioxímetro: Se pueden implementar diferentes perfiles según utilidad.

- Modo médico: para pacientes que necesiten tomarse la tensión, avisar cuando el pulsómetro sobrepase un nivel determinado.
- Modo deportivo: El dispositivo pueda tomar valores en un intervalo de tiempo determinado (o cuando por ejemplo el deportista pulse el acelerómetro) y luego pueda ver en casa en qué tramos lo ha pasado mejor, en cuáles sus pulsaciones se disparan etc.
- Modo Somnus: cuando las pulsaciones bajen de un determinado umbral, significará que el paciente duerme plácidamente. Cuando vuelvan a subir, se habrá despertado. Entre medias, se podría medir el tiempo que el usuario ha estado dormido.

Sensor de Temperatura: nos mostrará por pantalla la temperatura corporal. Podremos comparar cambios en esta medida.

2.4 Posibles mejoras

Existen diversas mejoras que podrían hacer de nuestro producto una oferta aún más atractiva. Estas van a estar basadas principalmente en el ahorro de la batería y en la incorporación de nuevas funcionalidades. Las exponemos a continuación:

- Mover la muñeca para activar la pantalla
- Reducción de brillo con el sensor UV
- Diseño reducido hasta el tamaño de la pantalla
- Sensor de pulso intermitente
- Hora
- Modo bajo consumo
- Disminuir tamaño Lipo

1 Acelerómetro: La aplicación servirá para establecer la distancia recorrida, la velocidad a la que se ha recorrido la distancia y el número de calorías quemadas. Puede empezar a contar todo esto a partir de cuándo se pulse un botón o cuando el sensor de sudoración esté activo. Una idea más alejada de la actual sería que podría servir para personas mayores, o enfermos de alguna patología que implique desmayos o caídas no controladas (ej: narcolepsia), el acelerómetro activaría una alarma para avisar de dicha caída.

2 Establecer una contraseña tocando la pulsera para desactivar el teléfono.

2.5 Problemas Surgidos:

Hay mucha discusión acerca de si las medidas que recogen estos sensores son fiables a nivel médico o no. Lo que es innegable es que nos preocupamos cuando la pulsera nos dice que tenemos 120 pulsaciones en reposo. Existen muchos tipos de sensores que miden el pulso cardíaco, hay incluso integrados que aparte del pulso miden el nivel de oxigenación en sangre. Nos quedamos con este sensor porque la lectura de los pines era analógica. El resto funcionaban con I2C, pero a la hora de conectarlos a nuestra placa nos fue imposible establecer conexión con ellos.

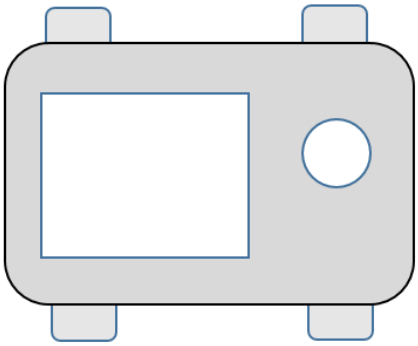
En este caso los ejemplos que encontramos por Internet no nos servían completamente, pues algunos empleaban interrupciones que estaban diseñadas para los micros de las placas Arduino, por lo que no nos servían.

También nos dio problemas el LM8511 ya que en Internet había infinidad de montajes y ejemplos para un sensor parecido pero que no servían para este. Además había que realizar un proceso de calibración previo para que los valores tratados tuviesen sentido.

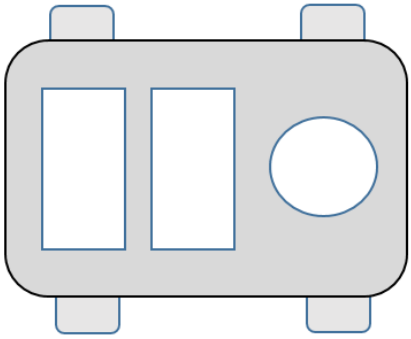
Respecto al sensor de humedad, lo que hemos tratado de hacer es medir la bioimpedancia de la piel. Nos dimos cuenta que variaba en función de la humedad superficial de la piel por lo que diseñamos un sensor que midiera la caída de tensión entre dos puntos. Esto lo teníamos que trasladar a la pulsera, que requería de dos contactos cercanos en la muñeca para realizar las medidas. Una vez solventamos estos problemas, tuvimos que asociar la medida a un estado de sudoración por medio de comparación con umbrales establecidos mediante prueba y error.

3. Manual de usuario

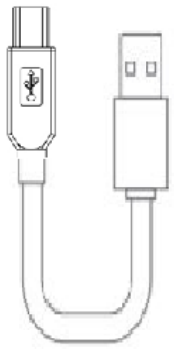
01: PARTES DE LA BIG BAND



Parte superior



Parte inferior

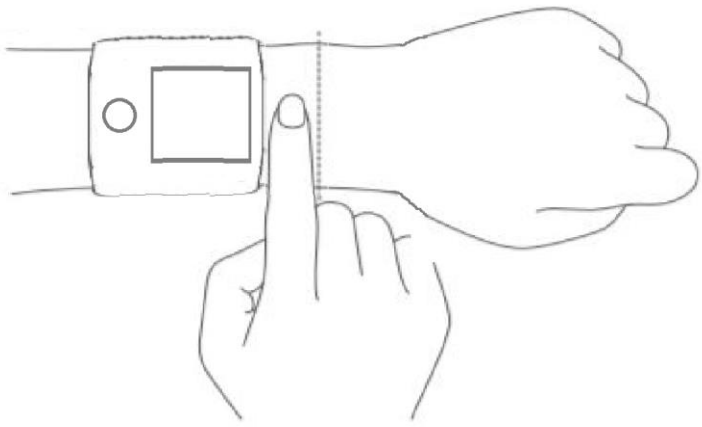


CABLE DE CARGA

Aquí se puede observar la parte superior e inferior de la pulsera
y el cable necesario para cargarla. En la parte superior se encuentra la pantalla (1) y el sensor de UV (2). En la parte inferior se encuentra el sensor de pulso (3), el sensor de temperatura (4) y el de humedad (5). El cable de carga necesario es un cable USB tipo B. Más adelante se podrá ver que se pueden comprar nuevas muñequeras para poder darle un toque más personal.

02: PRIMER USO DE LA BIG BAND

Lo ideal es cargarla completamente antes del primer uso, para ello, debemos usar el cable y conectarlo por la parte de la pulsera que tiene el sistema de conexión. Una vez colocado podremos conectarla a cualquier puerto USB, tanto de nuestro ordenador, de un hub o de un cargador.



Una vez cargada, introduzca la mano y apriete la muñequera alrededor de la muñeca hasta que esté lo suficientemente firme y en una posición cómoda. Recuerde que algunos sensores necesitan un buen contacto con su piel para el correcto funcionamiento.

03: CONEXIÓN ENTRE DISPOSITIVOS

Ahora hay que proceder a la conexión entre la pulsera y el teléfono móvil. Es compatible tanto con teléfonos de Apple y de Android. Lo primero es buscar la pulsera en las conexiones bluetooth disponibles para tu teléfono. Una vez emparejados, es necesaria la App del producto, para poder gestionarla. Busque en su tienda de archivos (Apple Store o Play Store) Health Band y descárguela. Dentro de Health Band sólo hay que buscar el nombre de la pulsera y ya quedará perfectamente conectada y lista para usar.

04: USANDO LA BIG BAND

Después de que la BIG BAND quede emparejada correctamente comenzará a rastrear y analizar sus actividades diarias y sus hábitos de sueño.

Mueva la muñeca lateralmente para visualizar en pantalla los diferentes datos de su actividad. La primera pantalla muestra la temperatura, la segunda la humedad superficial de la piel; la tercera la radiación UV que está recibiendo y la última el número de pulsaciones por minuto.

05: CÓMO CAMBIAR LA MUÑEQUERA DE LA BIG BAND

Quítese la pulsera de la muñeca y desenganche la muñequera de los cuatro puntos de enganche que tiene de manera suave, pero con firmeza. Una vez retirada, coloque la nueva muñequera y colóquela como se encontraba la anterior. Ajústela a la muñeca y lista para usar de nuevo.

06: CARGANDO LA BIG BAND

Cuando el indicador de batería esté bajo, deberás conectar el cable con la pulsera para poder cargarlo. Se puede conectar a casi cualquier puerto USB, ya que apenas necesita 1A/ 5v de potencia para poder cargarlo. Casi cualquier cargador vale, incluso el mismo del smartphone sirve.

07: CONSEJOS DE USO Y SEGURIDAD

Cuando utilice la BIG BAND para medir su frecuencia cardíaca, por favor mantenga la muñequera ajustada y la muñeca firme.

Si la BIG BAND entra en contacto con el agua, use un paño suave para secarlo lo antes posible, preferiblemente antes de usar el dispositivo de nuevo.

Durante el uso diario, por favor no use la Banda demasiado apretada alrededor de su muñeca, pues podría provocar molestias.

Es recomendable mantener limpia tanto la parte de los sensores superiores e inferiores para su correcto funcionamiento.

Si notas algún tipo de alergia a la pulsera o algún tipo de enrojecimiento o picor en la zona de contacto, deja de usar la pulsera y acude a un médico para ver el problema.

4. ANÁLISIS DE MERCADO

4.1 PREVISIÓN DE CRECIMIENTO DE VENTAS DE LAS PULSERAS INTELIGENTES

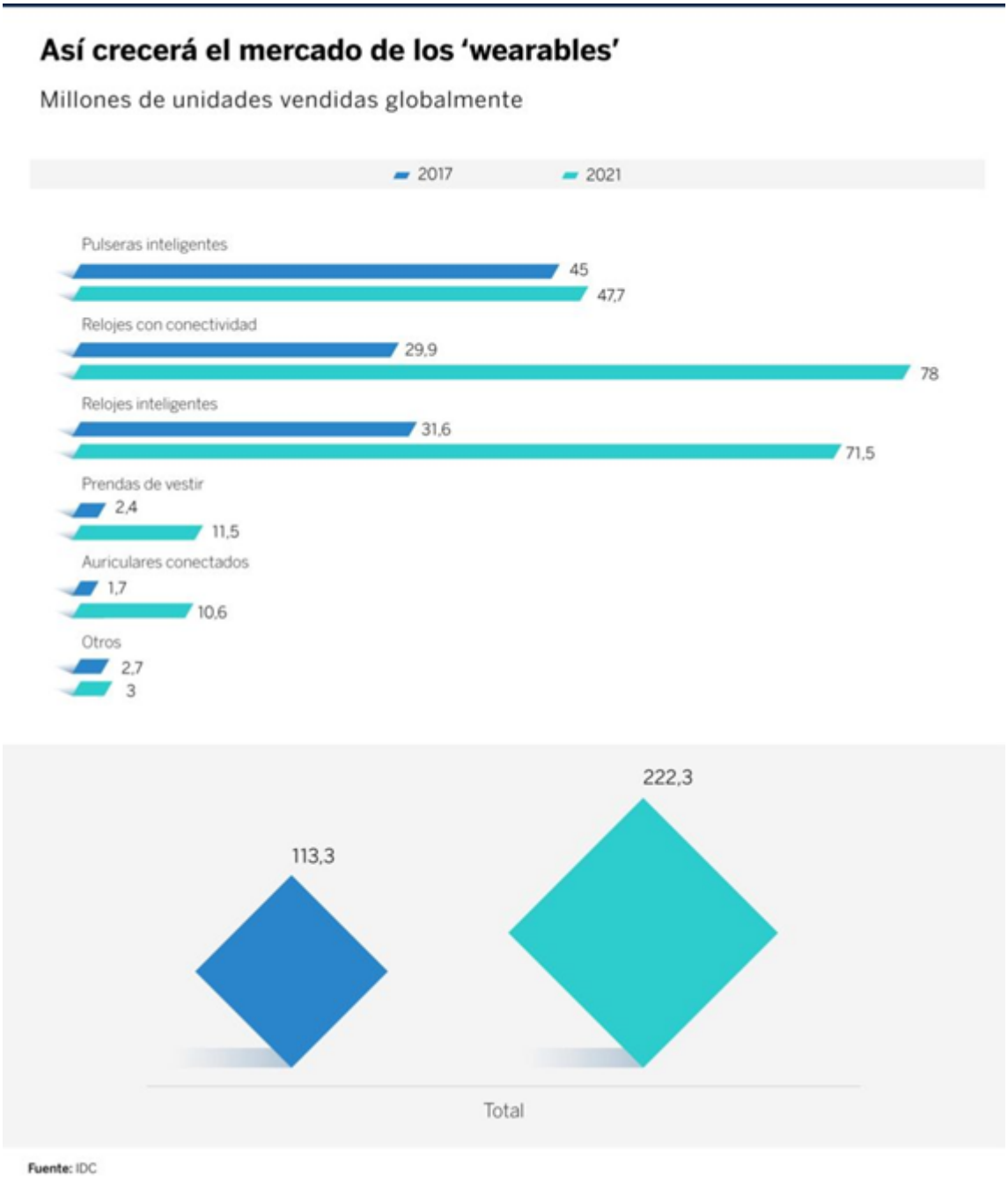
En los últimos años ha habido un enorme crecimiento en el uso de las pulseras inteligentes, que ofrecen todo tipo de aplicaciones, siendo uno de sus usos más comunes la monitorización de actividad y parámetros corporales. Estos dispositivos son la muestra de que los **'wearables'** —o tecnología ponible, es decir, tecnología que está incorporada a prendas o accesorios de uso cotidiano— ya empiezan a ser un objeto de consumo masivo.

Pero eso es solo el principio, como esperan las empresas que están invirtiendo en la industria de los 'wearables'. Las pulseras inteligentes, con sus diferentes niveles de prestaciones, son solo una de las mejores opciones dentro del gran universo de dispositivos que entran en esta categoría.

Desde el punto de vista de los usos, el reto es ampliarlos y, sin abandonar los ligados solo al 'fitness' y a la salud —donde hoy está la fortaleza de los 'wearables'—, ser también un aparato que mejore la productividad y la conectividad.

¿Cuáles son las previsiones comerciales de este mercado? Dos recientes informes tratan de responder a esa pregunta; uno, elaborado por la consultora International Data Corporation (IDC), ofrece una perspectiva global, el otro, de eMarketer, se centra en EE. UU.

Según IDC, el mercado global de los 'wearables' se va prácticamente a duplicar entre 2017 y 2021, al pasar de 113,3 millones de unidades vendidas en 2017 a 222,3 millones en 2021, con una tasa media de crecimiento anual del 18,4%.



Crecimiento del mercado de los 'weareables'.

Respecto a EE.UU., y en este caso según las proyecciones de eMarketer, las previsiones son de un crecimiento ralentizado, aunque no desdeñable. En esta consultora calculan que el número de usuarios adultos de 'wearables' en EE.UU. crecerá en 2018 un 11,9%, hasta alcanzar una tasa de penetración en la población del 19,6%, que será del 22,6% en 2021.

Vistos ambos estudios podemos concluir que invertir en un negocio de venta de pulseras inteligentes es una opción más que interesante de cara a los próximos años.

4.2 ANÁLISIS DE COMPETIDORES

Pulseras más vendidas

WILLFUL SW350	XIAOMI MiBAND 3	POLAR A370	FITBIT CHARGE 3	GARMIN VIVOSMART 4
35 g	20 g	31 - 37 g	29 g	20 - 21 g
Pantalla Táctil Color 0.96''	Pantalla táctil OLED 0.42''	Pantalla Color Táctil TFT 80x160	Pantalla Táctil OLED 100x150	Pantalla Táctil OLED 48x128
GPS (mediante smartphone)	GPS (mediante smartphone)	GPS (mediante smartphone)	GPS (mediante smartphone)	GPS (mediante smartphone)
Sensor Óptico Integrado	Sensor Óptico Integrado	Sensor Óptico Integrado	Sensor Óptico Integrado	Sensor Óptico Integrado
37 eur	27,90 eur	124,10	139 eur	128 eur

Características que reúnen las pulseras existentes.

Pulsómetro: Registro óptico de pulsaciones, normalmente lo llevan integrado en la caja de pulsera, habitualmente en la parte inferior y captan las pulsaciones a través de la piel. En algunos casos como en la pulsera Garmin Vivosmart4, se incorpora un pulsioxímetro, de forma que también mide los niveles de saturación de oxígeno en la sangre.
Resistentes al agua: algunas a salpicaduras, otras sumergibles hasta 50 metros
Control de sueño
Pantalla a color
Emparejamiento con el móvil para facilidad de disponer datos por GPS

Registro de pasos, calorías y distancias
Registro de saturación de oxígeno en la sangre
Respiraciones guiadas mediante suaves vibraciones
Modos deportivos, reconoce tipo de ejercicio
Monitor de estrés
Alerta de inactividad

Las pulseras de actividad en su mayoría están dirigidas a cualquier persona, utilizándose para monitorizar la actividad cotidiana. En muchas de estas pulseras, existe la posibilidad de configurar diferentes modos de forma que puedan adaptarse al deseo de cualquier deportista.

Las empresas ofrecen aplicaciones en diferentes interfaces con las que poder seguir la actividad o descargar planes de entrenamiento.

A su vez proporcionan una duración de batería longeva y muestran los mensajes y alertas de los smartphones.

En cuanto al diseño, se tratan de pulseras compactas, pequeñas y ligeras, disponibles en varios colores y con una pantalla de fácil lectura. En la mayoría de productos de este tipo, la pantalla se suele adoptar a los niveles de luz ambiental.

4.3 TARGET/CLIENTE OBJETIVO

En nuestro caso queremos enfocarlo principalmente a deportistas ya que nuestra pulsera dispone de un gran número de sensores para captar más parámetros: sudoración, temperatura corporal, aceleración, pulsaciones, concentración de oxígeno y radiación ultravioleta del ambiente. Por lo tanto, interesará principalmente a deportistas, tanto de élite como aficionados, que quieran llevar un estricto control sobre las condiciones y desempeño de su cuerpo. Sin embargo, es perfectamente útil para cualquier perfil de persona, incluso para el seguimiento de algún tipo de patología.

Proporcionaremos el hardware, que se comunicará con una aplicación que puede estar instalada tanto en un Smartphone como en un ordenador. En estos interfaces se mostrarán los datos de la sesión de actividad que estamos realizando al instante o un registro de una sesión previamente realizada.

4.4 COSTE

Componentes	Precio (aprox)/unidad
Temperatura: Sensor LM35	1,35 eur
Sudoración: Creación propia	-
Acelerómetro: GY-521 MPU-6050	2,79 eur
Sensor UV: ML8511	4,4 eur
Pulsioxímetro: Max30100	2 eur
Módulo Pro ESP-32 OLED V2.0 TT60 + Conector	22 eur
Baterías lipo (500 y 900 mAh)	5,50 eur
Material flexible PLA	0.5 eur
PRECIO ESTIMADO	40 EUR

Teniendo en cuenta el coste de los componentes, el trabajo de desarrollo de nuestros ingenieros, y el coste de compra y mantenimiento de maquinaria necesaria, estimamos que el precio de la pulsera es de 80 EUR.

4.5 PROVEEDORES

En cuanto a proveedores de bienes, se contactará con empresas que tengan puntos de venta o distribución en Madrid, para aligerar los costes y tiempos de entrega.

Componentes	Proveedores
Temperatura: Sensor LM35	Electrónica embajadores
Sudoración: Creación propia	-
Acelerómetro: GY-521 MPU-6050	Conectrol, S.A.
Sensor UV: ML8511	Electrónica y más
Pulsioxímetro: Max30100	Amazon
Módulo Pro ESP-32 OLED V2.0 TT60 + Conector	Banggod
Baterías lipo (500 y 900 mAh)	Rcmadrid
Material flexibleàPLA	trdimension

4.6 ESTRATEGIA DE MARKETING

Distribución

- Establecimiento de página web con tienda online
- En un futuro venta en tiendas físicas
- Previsión de establecer puntos de venta y atención al usuario propios.

Promoción

- Anuncios en páginas web deportivas y tecnológicas
- Anuncios en redes sociales: Facebook, Instagram, etc.
- Promoción en centros deportivos como gimnasios
- En un futuro contrato de publicidad con deportistas famosos.

5. Conclusión

Tras el largo proceso de elaboración de nuestro producto, hemos podido determinar unas conclusiones claras y que hacen de nuestra pulsera un artículo diferencial. Estas quedan explicadas a continuación:

-Oferta de un producto competitivo en relación a la calidad precio ofertada por otras grandes firmas como Samsung, Garmin o Xiaomi.

-Gran posibilidad de establecer cualquier tipo de mejora por parte de cualquier usuario que compre nuestra pulsera. Esto se debe a que va a poder establecer cualquier mejora por software al ser nuestro código de tipo “Open Source”.

-Considerable resistencia a impactos debidos a la robustez de nuestra carcasa, que protege el núcleo de nuestro producto de posibles impactos.

-Garantía de la viabilidad del negocio debido al auge que están teniendo los smart watch en la actualidad y la previsión de que las ventas de este tipo de dispositivos se mantengan en los próximos años.

6. ANEXO

Datasheet de los componentes:

- Sensor de temperatura LM35: http://www.ti.com/lit/ds/symlink/lm35.pdf
- Sensor detector UV ML8511: https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/ML8511_3-8-13.pdf

Tutoriales de los componentes:

- Sensor de impacto KY-031: https://electronicastore.net/producto/modulo-ky-031-sensor-de-impacto-arduino-o-micro/

- Sensor de temperatura LM35: https://programarfacil.com/tutoriales/fragmentos/leer-el-sensor-de-temperatura-lm35-en-arduino/

- Sensor detector UV ML8511: https://learn.sparkfun.com/tutorials/ml8511-uv-sensor-hookup-guide/all

Aquí se va a introducir todo el código necesario para el correcto funcionamiento del dispositivo:

```
#include <Arduino.h>
#include <U8x8lib.h>
#include <stdint.h>
#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8 HAVE HW I2C
#include <Wire.h>
#endif
      #include "BluetoothSerial.h"
      #if !defined(CONFIG_BT_ENABLED) ||
!defined(CONFIG_BLUEDROID_ENABLED)
      #error Bluetooth is not enabled! Please run `make menuconfig` to
and enable it
      #endif
      //-------------------------------------------
      //              DEFINICION DE PINES
      const int Aread = 37;
      const int UVOUT = 34;//12
      const int REF_3V3 = 39;//14
      const int BUTTON = 35;//27
      const int HUM = 36;
      const int PIN_PULSO = 38;
      //-------------------------------------------
      //              DEFINICION DE CONSTANTES
      const uint32_t t_espera = 3000;
      const uint32_t t_espera_pres = 4000;
      const uint32_t t_espera_ble = 10000;
      const uint32_t t_button = 500;
      const uint32_t t_muestreo = 333;
      const uint32_t t_medicion = 6000;
      const int Threshold = 2500;
      //-------------------------------------------
      //-------------------------------------------
      //            DEFINICION DE VARIABLES
      BluetoothSerial SerialBT; // No es una variable como tal
      int sensor;
      double tem;
      double humedad;
      int n = 0;
      int dig_hum = 0;
      int lectura = 0;
      double pulso = 0;
      float uvIntensity;
      int uvLevel;
      int refLevel;
      float outputVoltage;
      uint32_t tiempo = 0;
      uint32_t t_puls = 0;
      uint32_t t_ble = 0;
      uint32_t t = 0;
      uint32_t t_largo = 0;
      int ref = 0;
      int pulsado = LOW; // HIGH
      uint32_t anterior = 0;
      boolean bAlarm = false;
      String text;
      String hora;
      //-------------------------------------------
      //-------------------------------------------
      //               PINES PANTALLA
      U8X8 SSD1306 128X64 NONAME SW I2C u8x8(/* clock=*/ 15, /* data=*/
4, /* reset=*/ 16);
```

```
61.       //-------------------------------------------
62.       /*
63.        * Configuro los pines como entrada, Inicializo el puerto serie y
   la pantalla
64.        */
65.       void setup() {
66.         pinMode(Aread,INPUT);
67.         pinMode(UVOUT,INPUT);
68.         pinMode(REF_3V3,INPUT);
69.         pinMode(HUM,INPUT);
70.         pinMode(BUTTON,INPUT);
71.         pinMode(PIN PULSO,INPUT);
72.         u8x8.begin();
73.         Serial.begin(115200);
74.         SerialBT.begin("ESP32test"); //Bluetooth device name
75.         Serial.println("The device started, now you can pair it with
   bluetooth!");
76.       }
77.       //-------------------------------------------
78.       //              FUNCIONES AUXILIARES
79.       /*
80.        * Muestra en la parte superior de la pantalla "ELCO PROJECT"
81.        */
82.       void pre(void)
83.       {
84.         u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
85.         u8x8.clear();
86.
87.         u8x8.inverse();
88.         u8x8.print(" ELCO PROJECT ");
89.         u8x8.setFont(u8x8_font_chroma48medium8_r);
90.         u8x8.noInverse();
91.         u8x8.setCursor(0,1);
92.       }
93.       //-------------------------------------------
94.       /*
95.        * Muestra en la parte superior de la pantalla
96.        * "Temp (Celsius)"
97.        */
98.        void preT(void)
99.       {
100.        u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
101.        u8x8.clear();
102.
103.        u8x8.inverse();
104.        u8x8.print(" Temp (Celsius) ");
105.        u8x8.setFont(u8x8_font_chroma48medium8_r);
106.        u8x8.noInverse();
107.        u8x8.setCursor(0,1);
108.      }
109.      //-------------------------------------------
110.      /*
111.       * Muestra por pantalla "UV I(mW/cm^2)"
112.       */
113.       void preI(void)
114.      {
115.        u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
116.        u8x8.clear();
117.
118.        u8x8.inverse();
119.        u8x8.print("UV I(mW/cm^2)");
120.        u8x8.setFont(u8x8_font_chroma48medium8_r);
121.        u8x8.noInverse();
122.        u8x8.setCursor(0,1);
123.      }
124.      //-------------------------------------------
```

```
125.      /*
126.       * Muestra por pantalla "Sweat Level"
127.       */
128.       void preH(void)
129.      {
130.        u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
131.        u8x8.clear();
132.
133.        u8x8.inverse();
134.        u8x8.print("Sweat Level");
135.        u8x8.setFont(u8x8_font_chroma48medium8_r);
136.        u8x8.noInverse();
137.        u8x8.setCursor(0,1);
138.      }
139.      /*
140.       * Muestra por pantalla Heart Rate (bpm)
141.       */
142.       void preP(void){
143.        u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
144.        u8x8.clear();
145.
146.        u8x8.inverse();
147.        u8x8.print("Heart Rate (bpm)");
148.        u8x8.setFont(u8x8_font_chroma48medium8_r);
149.        u8x8.noInverse();
150.        u8x8.setCursor(0,1);
151.       }
152.      //----------------------------------------
153.
154.      /*
155.       * Indica el nivel de sudoracion respecto a una referencia.
156.       */
157.      void hum_level(void){
158.        if((ref == 0) && (dig_hum > 100)){
159.          ref = humedad;
160.        }
161.      }
162.      //-----------------------------------------------------------------
    ---------
163.      //Takes an average of readings on a given pin
164.      //Returns the average
165.      int averageAnalogRead(int pinToRead)
166.      {
167.        byte numberOfReadings = 8;
168.        unsigned int runningValue = 0;
169.        for(int x = 0 ; x < numberOfReadings ; x++)
170.          runningValue += analogRead(pinToRead);
171.
172.        runningValue /= numberOfReadings;
173.        return(runningValue);
174.      }
175.      //The Arduino Map function but for floats
176.      float mapfloat(float x, float in_min, float in_max, float out_min,
    float out_max)
177.      {
178.        return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
    out min;
179.      }
180.
181.      //-----------------------------------------------------------------
    ------------
182.      //-----------------------------------------------------------------
    ------------
183.      /*
184.       * Presenta la temperatura en pantalla
185.       */
```

```
186.      void temperature(void){
187.        preT();
188.        u8x8.setFont(u8x8_font_open_iconic_weather_4x4);
189.        u8x8.drawGlyph(6, 3, '@'+5);
190.        delay(500);
191.        pre();
192.        u8x8.setCursor(0,1);
193.        u8x8.setFont(u8x8_font_inb33_3x6_n);
194.        //u8x8.setFont(u8x8_font_amstrad_cpc_extended_f);
195.      //  u8x8.setCursor(0,2);
196.        //u8x8.print("temp(°C):");
197.        u8x8.setCursor(0,2);
198.        u8x8.print(tem);
199.        }
200.      //-------------------------------------------------------------------
    ----------
201.        /*
202.         * Presenta la radiacion UVA por pantalla
203.         */
204.      void radiation(void){
205.        preI();
206.        u8x8.setFont(u8x8_font_open_iconic_weather_4x4);
207.        u8x8.drawGlyph(6, 3, '@'+0);
208.        delay(500);
209.        pre();
210.        u8x8.setFont(u8x8_font_inb33_3x6_n);
211.        u8x8.setCursor(0,2);
212.        u8x8.print(uvIntensity);
213.        }
214.      //-------------------------------------------------------------------
    ------------
215.        /*
216.         * Presenta la humedad por pantalla
217.         */
218.      void humedo(void){
219.        preH();
220.        u8x8.setFont(u8x8_font_open_iconic_weather_4x4);
221.        u8x8.drawGlyph(6, 3, '@'+1);
222.        delay(500);
223.        pre();
224.        u8x8.setFont(u8x8_font_inb33_3x6_n);
225.        u8x8.setCursor(0,2);
226.        //u8x8.print(humedad);
227.        if (ref != 0){
228.          if ((humedad-ref) <= 0.5){
229.            u8x8.print(0);
230.          } else if((humedad-ref) <= 1.2) {
231.            u8x8.print(1);
232.          } else {
233.            u8x8.print(2);
234.          }
235.        }
236.        }
237.      //-------------------------------------------------------------------
    -----------
238.      /*
239.       *   Presenta el pulso por pantalla
240.       */
241.      void pulsaciones(void){
242.         preP();
243.         u8x8.setFont(u8x8_font_open_iconic_weather_4x4);
244.         u8x8.drawGlyph(6, 3, '@'+2);
245.         delay(500);
246.         pre();
247.         u8x8.setFont(u8x8_font_inb33_3x6_n);
248.         u8x8.setCursor(0,2);
```

```
249.          if(pulso == 0){
250.           pulso = -1;
251.          } else if(pulso <= 40) {
252.           pulso = 65;
253.          }
254.         u8x8.print(pulso);
255.       }
256.       /*
257.        *
    ================================================================================
    ======
258.        */
259.       void loop() {
260.         pulsado = digitalRead(BUTTON);
261.         if(pulsado){ // !pulsado
262.           t_puls = millis();
263.           if(!bAlarm){
264.             n++;
265.             n%=5; // 4
266.             bAlarm = true;
267.           }
268.
269.         } else{
270.           if((millis()-t_puls) > t_button && bAlarm){
271.             bAlarm = false;
272.           }
273.         }
274.         if((millis()-t_puls) > t_espera_pres){
275.           n = 0;
276.         }
277.         if ((millis()-tiempo >= t_espera)|| (pulsado)){ // !pulsado
278.           tiempo = millis();
279.           if (n == 0){
280.               u8x8.clear();
281.           } else if (n == 1){
282.               sensor = averageAnalogRead(Aread);
283.               tem = (sensor * 5.0/40.96);
284.               temperature();
285.           } else if(n == 2){
286.               dig_hum = averageAnalogRead(HUM);
287.               humedad = dig_hum * 3.3/4096;
288.               hum_level();
289.               humedo();
290.           }else if(n == 3) {
291.               if ((millis()-t) < t_muestreo) {
292.               }else {
293.                 lectura = averageAnalogRead(PIN_PULSO);
294.                 if (lectura > Threshold){
295.                   pulso++;
296.                 }
297.               }
298.
299.               if ((millis()-t_largo) < t_medicion){
300.                   } else {
301.                     t_largo = millis();
302.                     pulso *=6;
303.                     pulsaciones();
304.                     pulso = 0;
305.                   }
306.               }
307.           else {
308.             uvLevel = averageAnalogRead(UVOUT);
309.             refLevel = averageAnalogRead(REF_3V3);
310.             // Use the 3.3V power pin as reference to get a very accurate
    output value from sensor
311.             outputVoltage = (3.3 / refLevel) * uvLevel;
```

```
312.            uvIntensity = mapfloat(outputVoltage,0.7,2.9,0.0,15.0);
313.            if (uvIntensity <= 0){
314.              uvIntensity = 0;
315.              }
316.            radiation();
317.          }
318.        }
319.
320.
321.        if((millis()-t_ble)>= t_espera_ble){
322.          t_ble = millis();
323.          text = "{\"";
324.          text +="heartrate\":";
325.          text +=pulso;
326.          text +=",\"temperature\":";
327.          text +=tem;
328.          text +=",\"humidity\":";
329.          text += humedad;
330.          text +=",\"Radiation\":";
331.          text += uvIntensity;
332.          text +="}";
333.          SerialBT.println(text);
334.
335.      //-
336.
337.
338.        }
339.      }
340.
```