



Consellería de Educación

**Proxecto de Innovación premiado na resolución do 9 de maio de 2017 da Dirección Xeral de Educación, Formación Profesional e Innovación Educativa** pola que se resolven os premios para o desenvolvemento de proxectos de innovación tecnolóxica ou científica e proxectos de innovación didáctica no ámbito da Formación Profesional en centros públicos dependentes da Consellería de Cultura, Educación e Ordenación Universitaria, onde se imparten ensinanzas de Formación Profesional, convocados na resolución 22 de decembro de 2016.

## Sumario

- 1 Introducción
  - ◆ 1.1 Termofluxómetro
    - ◇ 1.1.1 Transmitancia térmica - Que é?
    - ◇ 1.1.2 Medidor Sensoralia de transmitancia térmica
    - ◇ 1.1.3 Uso do medidor de transmitancia térmica
    - ◇ 1.1.4 Configurar a tablet para recibir datos de medición
      - 1.1.4.1 Descargar a app MQTT Dash.
      - 1.1.4.2 Conectar a tablet á rede da Raspberry Pi.
      - 1.1.4.3 Canais xerados pola Raspberry do termofluxómetro.
        - 1.1.4.3.1 Configurar o canal principal
        - 1.1.4.3.2 Configurar botóns de ordes e paneis de información.
        - 1.1.4.3.3 Configuración do botón on/off
        - 1.1.4.3.4 Configuración de recepción de temperaturas
        - 1.1.4.3.5 Configuración de recepción de U-valor minutil e dezminutil
        - 1.1.4.3.6 Configuración de recepción de valor medio global
  - ◆ 1.2 Medidor da calidade do aire interior
    - ◇ 1.2.1 Evolución do deseño do *hardware*
    - ◇ 1.2.2 Instalación dunha estación Raspberry
    - ◇ 1.2.3 Videos de instalación da Raspberry en Modo Autónomo
    - ◇ 1.2.4 Instalación do servidor web central Sensoralia
      - 1.2.4.1 Instalación dun servidor LAMP nunha máquina con Debian 9
      - 1.2.4.2 Creación dun usuario e base de datos en MariaDB
      - 1.2.4.3 Configuración do servidor web e instalación da aplicación Sensoralia dende as fontes.
    - ◇ 1.2.5 Videos de instalación e configuración de Sensoralia nun servidor web central
    - ◇ 1.2.6 Videos de instalación de Raspberry como cliente dun servidor web central
    - ◇ 1.2.7 Instalación "dende cero" do sistema ---Por se é de interese ---
      - 1.2.7.1 Descargar e instalar Raspbian
      - 1.2.7.2 Configuración desde cero con imagen Raspbian
- 2 Licenza

## Introdución

Aspectos xerais do Proxecto de Medición para a Eficiencia Enerxética en Hardware Libre (emeehl)

Co proxecto EMEEHL trátase de deseñar e construír dous equipos didácticos de medición necesarios para a avaliación da eficiencia enerxética en edificios e industrias. Estes equipos construíranse empregando hardware e software libre e porase á disposición da comunidade as especificacións e diagramas de funcionamento. O público destinatario destes equipos é os profesores e alumnos das familias profesionais de 'Enerxía e Auga' e 'Instalación e Mantemento?', dos Ciclos de Formación Profesional.

Os equipos comerciais existentes no mercado teñen un alto prezo que dificulta a compra e uso dos mesmos por parte de centros educativos con poucos recursos. Neste proxecto preténdese poñer á disposición da comunidade educativa instrumentos de medición totalmente funcionais que se podan construír de xeito sinxelo e con baixo custe.

Os equipos que se desenvolverán na fase inicial son:

Equipo de medición portátil de transmitancia térmica  
Equipo de medición continua de parámetros de confort e eficiencia enerxética das aulas (temperatura, humidade, calidade do aire, luminosidade, ruído, etc).

Aquí podemos ver o video de presentación do proxecto:

## Termofluxómetro

O termofluxómetro Sensoralia é un instrumento deseñado para a medición da transmitancia térmica en cerramentos. Está integramente elaborado empregando Hardware (HW) e Software (SW) libres, en particular RaspberryPi e derivados de Arduino, programados en bash, Python e C.

O equipo mide varias temperaturas dentro e fora da edificación a fin de calcular de xeito indirecto o valor da transmitancia térmica.

### Transmitancia térmica - Que é?

#### Transferencia de calor.

Cando dous corpos se atopan a diferentes temperaturas prodúcese un fluxo de calor do corpo que está a máis temperatura cara ao que está a menor temperatura. Este fluxo de calor mantense no tempo até que se igual ambas temperaturas, acadándose o equilibrio térmico.

O intercambio de calor ten lugar por medio de tres procesos distintos:

- **Radiación.** Calquera corpo, polo feito de estar a unha determinada temperatura, emite radiación electromagnética (EM). A intensidade da radiación EM depende da temperatura, así como a súa lonxitude de onda. En xeral, para temperaturas relativamente baixas podemos considerar que a maior parte da intensidade de radiación está no rango do infravermello (IR polas súas siglas en inglés). Para temperaturas elevadas pode chegar a emitir intensidades apreciables no rango da luz visible.
- **Convección.** Esta forma de transferencia de calor é característica dos fluídos (gases e líquidos) e consiste en correntes dentro dos mesmos, debidas ás diferencias de densidade causadas polo gradiente de temperaturas. Estes fluxos de fluído son as correntes convectivas e conforme desprazan o líquido ou gas intercambian calor, reducindo o gradiente de temperaturas até acadar o equilibrio.
- **Conducción.** É característica dos sólidos e consiste en que ao por en contacto dous sólidos a distinta temperatura, establécese unha transferencia de enerxía térmica cara as rexións máis cercanas e que depende das características físicas e estruturais de ambos sólidos. Os corpos que mellor deixan pasar a enerxía son bos condutores térmicos e caracterízanse por unha baixa conductividade térmica.

#### Transmitancia térmica.

É a medida do calor transmitido por unidade de tempo a través dun sistema constructivo, como unha parede ou fiestra, formado por unha ou máis capas de material de caras plano paralelas. Defínese a transmitancia térmica para un fluxo estacionario por unidade de superficie e diferenza de temperatura. Chámasele tamén valor-U e mídese en  $W/m^2 \cdot K$  no sistema internacional. Emprégase para expresar a capacidade illante dun elemento constructivo formado por unha ou máis capas de materiais e é a inversa da resistencia térmica. Inclúe os efectos de película en ambos lados, debidos aos movementos convectivos do aire no exterior e no interior. Por tanto a transmitancia térmica ten en conta a transmisión de calor tanto por conducción como por convección. É moi importante coñecer a transmitancia térmica dun cerramento, para saber cales son as perdas de calor (xeralmente no inverno) ou as ganancias (no verán) e conseguintemente poder calcular as necesidades de calefacción ou climatización. Acontece o mesmo para o cálculo das cargas dunha instalación frigorífica e por tanto dos elementos da mesma.

#### Cálculo e medición da transmitancia térmica

A transmitancia térmica pódese (a) calcular no caso de proxectos constructivos ou (b) medir, no caso de edificacións xa construídas. No primeiro caso obtense a transmitancia como inversa da resistencia global do elemento constructivo. Por súa vez a resistencia global é a suma das resistencias térmicas individuais das capas que conforman o elemento, xunto coas resistencias superficiais exterior e interior (inversa do coeficiente de

convección).

$$U = \frac{1}{R_t} = \frac{1}{R_{si} + R_1 + R_2 + \dots + R_n + R_{se}}$$

Para o caso de estimar a transmitancia térmica en elementos xa construídos, pódense empregar métodos directos como (a) a medición do fluxo de calor ou indirectos como (b) os métodos termográficos ou (c) o método de temperaturas múltiples. Neste último método é necesario medir as diferencias de temperatura no exterior e no interior durante un tempo suficientemente longo como para que se estableza un rexime estacionario de transferencia de calor. Existen no mercado diversos aparatos de medición da transmitancia térmica, nomeadamente os da casa Testo.

## Medidor Sensoralia de transmitancia térmica

### Consideracions teóricas.

A aparición de normativas reguladoras como o CTE, a Certificación Enerxética de Edificios, o RITE, etc, que obrigan ao cumprimento duns mínimos, fai que os técnicos encargados da realización de proxectos deban ter unha maior preparación teórica e práctica en relación coas medicións en obra. A transmitancia térmica é un dos valores máis problemáticos de medir se non se quere estimar mediante valores obtidos de táboas.

En edificacións xa construídas e particularmente nas construídas nos anos 70 e 80 o único xeito de ter unha estimación da transmitancia é medila por algún dos métodos directos ou indirectos. O noso termofluxómetro emprega o método indirecto de temperaturas múltiples, que consiste en estimar o fluxo de calor a partir da temperatura ambiente exterior e as temperaturas interiores de superficie e ambiental.

Se consideramos o caso simple dunha parede exterior como a da figura, existe un fluxo de calor por unidade de área dende o interior ao exterior. No rexime estacionario, o fluxo que chega á superficie interior é igual ao que atravesa a parede. Para calcular o fluxo que chega á superficie interior é preciso ter a diferenza de temperaturas interiores e o coeficiente de convección interior que facilita o CTE:

$$\dot{Q}/A = (T_i - T_{si}) \cdot h_{ci}$$

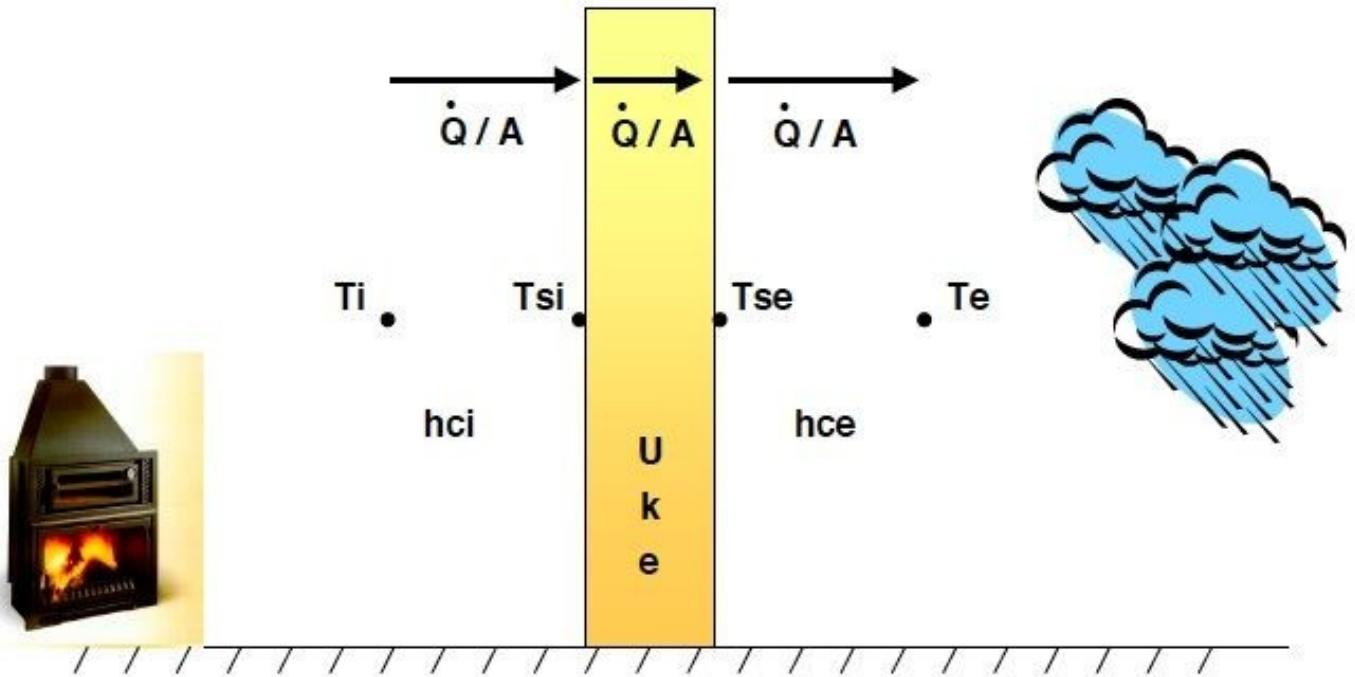
O fluxo que atravesa a parede depende do coeficiente de transmisión térmica e a diferenza de temperaturas entre o interior e o exterior:

$$\dot{Q}/A = (T_i - T_e) \cdot U$$

Idealmente, no rexime estacionario, os fluxos de calor que chegan á cara interior da parede e que atravesan á mesma, son iguais. Polo tanto, ao igualar ambas expresións e despexar, podemos obter o U-valor:

$$U = \frac{T_i - T_{si}}{T_i - T_e} \cdot h_{ci}$$

Para asegurarnos que a medición é correcta e estamos o máis cerca posible dunha situación estacionaria, debemos tomar só valores cando a diferenza entre as temperaturas exterior e interior (ambiente) sexa maior de 15 °C e ao longo dun número suficiente de horas. Os valores tomados cando da diferenza de temperaturas é inferior aos 15 °C, débense desbotar.



$Q/A$  : Flujo de calor por unidad de área en  $[W/m^2]$   
 $e$  : espesor de la pared en  $[m]$   
 $k$  : conductividad térmica en  $[W/mK]$   
 $h_{ci}$  : coef. convección interior en  $[W/m^2K]$   
 $h_{ce}$  : coef. convección exterior en  $[W/m^2K]$

$T_i$  : temperatura ambiente interior en  $[^{\circ}C]$   
 $T_{si}$  : temperatura superficie interior en  $[^{\circ}C]$   
 $T_e$  : temperatura ambiente exterior en  $[^{\circ}C]$   
 $T_i$  : temperatura ambiente interior en  $[^{\circ}C]$   
 $U$  : Coef. global de transmisión en  $[W/m^2K]$

#### Termofluxómetro sensorialia.

O noso medidor de transmitancia térmica está integramente realizado con HW e SW libre. A escolla feita para desenvolve-lo foi

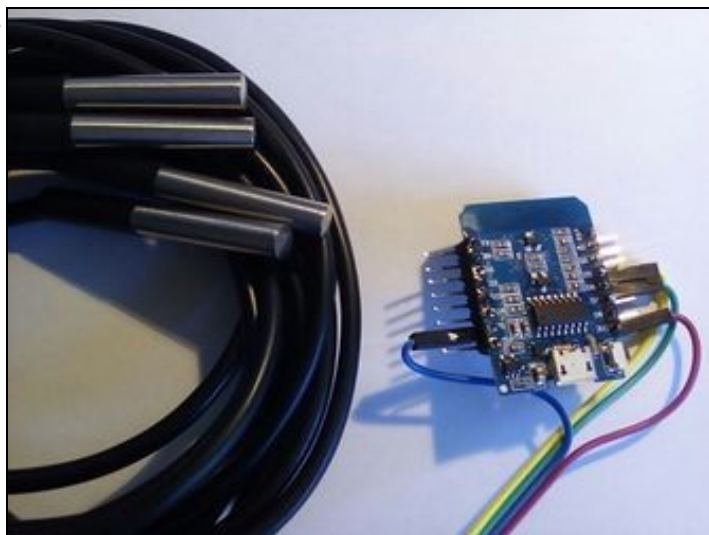
- (a) un ordenador RaspberryPi que actúa como receptor de datos e que en base aos mesmos, calcula o valor- $U$ ,
- (b) varios sensores de temperatura DS18B20 e
- (c) polo menos dous módulos ESP8266 que recollen os valores de temperatura e os envían por WiFi á RaspberryPi.

Na fotografía seguinte obsérvanse os sensores de temperatura DS18B20 conectados a un dos módulos ESP8266. Este módulo é un microchip de baixo custo que implementa toda a pila do protocolo TCP/IP e que pode ser programado en varias linguaxes de programación, entre elas o C de Arduino. Este microchip está sendo moi empregado en dispositivos de Internet das Cousas (IdT ou IoT polas súas siglas en inglés) polo seu pequeno tamaño, facilidade de programación e baixo prezo. Existen diversos encapsulados do microchip para flexibilizar o seu uso en todo tipo de aplicacións.

Cada un dos sensores (exterior e interior) está constituído por un módulo ESP8266 conectado a un ou varios sensores de temperatura e programado de xeito que se conecte á rede WiFi proporcionada pola RaspberryPi para a transmisión das temperaturas. Os scripts do módulo ESP8266 están dispoñibles neste [repositorio de GitHub](#).

A seguinte fotografía mostra a Raspberry que xestiona a conexión WiFi así como os datos proporcionados polos sensores interior e exterior. A RaspberryPi está preparada para

- (a) conectarse á rede mediante cable ethernet, en caso de existir,
- (b) xerar unha rede WiFi para por a disposición dos sensores exteriores e
- (c) xestionar os datos de temperatura. O script que xestiona a recepción e emisión de datos está dispoñible neste [repositorio GitHub](#).



Detalle do sensor de temperatura exterior



RaspberriPi montada co alimentador lipo externo

A RaspberryPi envía periodicamente (cada 10 minutos por defecto) os valores-U calculados a [esta canle de ThingSpeak](#) a fin de observar como evoluciona o valor-U ao longo do tempo e valorar a variación da transmitancia. Do mesmo xeito, a RaspberryPi ten instalado un servidor MQTT que permite enviar mensaxes a outros dispositivos como táblets ou teléfonos. Pódese observar na seguinte imaxe un exemplo de aplicación que recibe valores-U cada minuto e cada dez minutos. Este exemplo permite tamén interactuar coa Raspberry, a fin de empezar a medir ou parar cando o operador do dispositivo desexe.

### Uso do medidor de transmitancia térmica

#### Conectar a Raspberry Pi.

Para conectar a Raspberry Pi é necesario introducir en primeiro lugar a tarxeta SD previamente cargada coa [distribución proporcionada por Sensoralia aquí](#). A tarxeta SD contén o sistema operativo (SO) necesario para iniciar á Raspberry, así como o software necesario para recoller as temperaturas e procesalas. Aliméntase a Raspberry e déixase iniciar o sistema uns segundos. Para alimentar a Raspberry pódese conectar á rede eléctrica ou mediante unha batería externa, como é o noso caso. Na fotografía inferior móstrase a Raspberry montada sobre un shield que controla á batería externa.

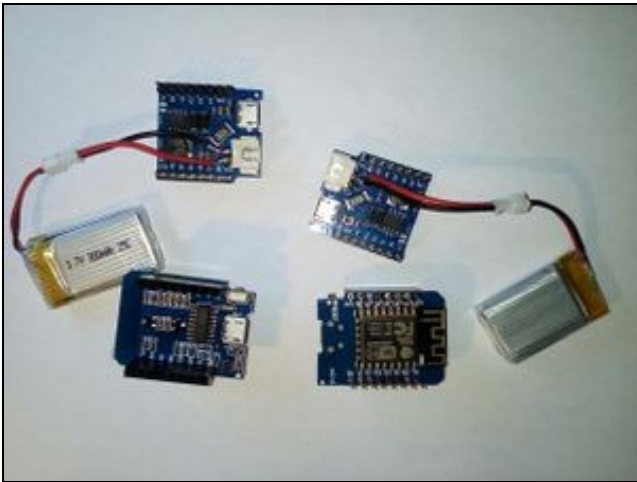


Raspberry Pi montada co alimentador lipo externo

O software que acompaña ao sistema operativo inicia automaticamente un script que se encarga de recibir as temperaturas enviadas polos sensores, así como xestionar os resultados e comunicarse con outros dispositivos externos. Non é preciso facer nada máis na Raspberry que asegurarse de que ten a SD e encender. Se fose preciso interactuar con ela, pódese facer tanto por SSH como cun cliente VNC no enderezo 192.168.3.1 da rede que xenera, que por defecto ten o SSID RaspiWiFi.

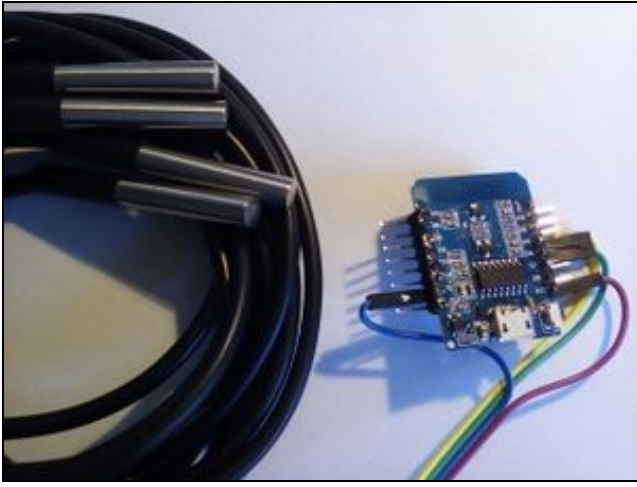
### **Conectar os sensores de temperatura.**

Cada sensor de temperatura está formado por un microchip ESP8266 conectado a unha batería lipo mediante un regulador de carga. Na seguinte imaxe móstranse dous destes sensores desmontados (por un lado o microchip e por outro o regulador xunto coa batería).



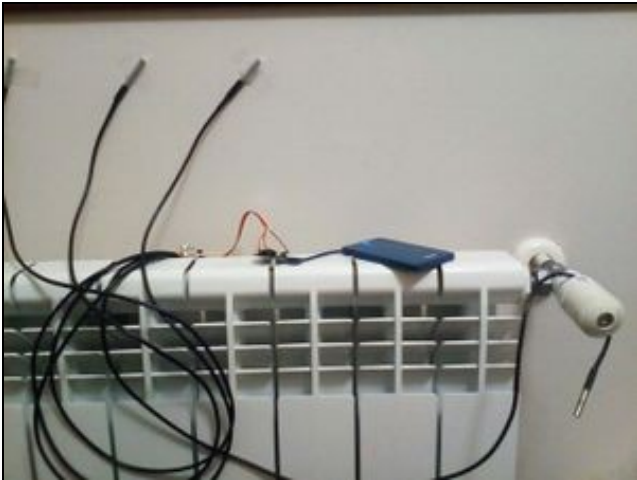
Dispositivo ESP8266 (vista anverso e reverso) e reguladores de carga

O módulo ESP8266 leva cargado un script que se encarga de conectarse automaticamente á rede creada pola Raspberry e enviar periodicamente os valores de temperatura que vai lendo nos sensores. O script é lixeiramente diferente se vai ir cargado no sensor exterior ou no interior. Por iso hai que ter coidado de non confundir un co outro. Os sensores de temperatura son de tipo DS18B20, encapsulados nunha carcasa metálica e unidos a unha manguera que facilita a conexión ao microchip ESP8266. Na imaxe inferior móstrase un detalle do sensor interior, na que se ven por un lado os DS18B20 encapsulados e polo outro o ESP8266 conectado ás mangueras.



#### Sensor de temperatura exterior

O sensor exterior pode ter un ou varios DS18B20 conectados (vanse tratar como varias medidas dunha mesma temperatura), sen embargo o sensor interior debe ter polo menos catro DS18B20. A razón disto é que polo menos tres dos DS18B20 deben estar en contacto coa cara interior do cerramento e o outro debe medir a temperatura ambiente interior. Na imaxe inferior móstrase un exemplo de como se deben dispor os sensores de temperatura do sensor interior. As distancias entre os sensores en contacto coa parede deben ser maiores e a do sensor de temperatura ambiente debe ser a maior posible en relación coa cara interior do cerramento.



#### Sensor de temperatura interior

Por outra banda, é recomendable asegurar o contacto térmico entre os sensores e a parede, empregando algún tipo de masilla adhesiva que asegure que o sensor permanecerá pegado todo o tempo necesario e que mellore o contacto térmico. Pódese empregar para isto masillas estilo Blu-tack.

Na seguinte imaxe pódense ver os elementos que constitúen o termofluxómetro:

- (a) sensor exterior (dereita)
- (b) unidade de proceso (Raspberry Pi, centro)
- (c) sensor interior (esquerda)
- (d) dispositivo Android (abaixo)





Elementos do termofluxometro

#### Habilitar a recepción de temperaturas.

Para que tanto o sensor interior como o exterior funcionen, só é preciso que a batería estea conectada e teña algo de carga. O script cárgase inmediatamente e empeza a enviar temperaturas. Por outra banda, aínda que se acabe de encender a Raspberry, non vai xestionar as temperaturas até que se lle indique dende un dispositivo externo como unha tablet ou un teléfono móbil. O dispositivo externo debe habilitarse na rede da Raspberry Pi para poder controlala. Na imaxe seguinte móstrase un exemplo de app android que habilita o script e recibe periodicamente os valores-U.





Control do termofluxómetro usando dispositivo Android

Neste exemplo vese o botón de on/off, que habilita a recepción e procesamento dos datos de temperatura. Mentres o botón estea en off, a Raspberry non poderá acceder ao canal no que se comunican as temperaturas. Unha vez habilitada mediante o estado de on, a Raspberry calculará periodicamente os valores da transmitancia, enviará os valores 10-minutais ao canal de ThingSpeak e tamén ao canal MQTT que escoita a app. No exemplo da figura anterior vese o último valor 10-minutal dunha proba, así como último valor minutal. Ningún dos valores é dunha proba real.

Pode haber un número indeterminado de dispositivos conectados ao canal MQTT da Raspberry. Só é necesario descargar a app android MQTT Dash e definir os canais aos que se vai conectar. Noutra páxina desta Wiki indícase como conectarse aos canais de ordes e de recepción de valores-U minutais e 10-minutais.

## Configurar a tablet para recibir datos de medición

### Descargar a app MQTT Dash.

Na Play Store de Android hai dispoñíbles varias apps que nos permiten acceder aos canais MQTT que proporciona o software instalado na Raspberry Pi. Entre as máis salientables están MQTT Dash, Linear MQTT Dashboard e IoT MQTT Panel. Nós imos instalar e configurar a primeira delas.

Por tanto, antes de seguir é preciso instalar na tablet ou móbil a aplicación MQTT Dash. Unha vez feito isto xa podemos empezar a configurar a mesma.

### Conectar a tablet á rede da Raspberry Pi.

O elemento central do termofluxómetro é a Raspberry Pi, que se encarga de recibir os datos dos sensores de temperatura, procesalos e emitir os resultados de U-valor. Para xestionar o fluxo de información, a Raspi xenera unha rede wifi propia á que se debe conectar calquera dispositivo que queira emitir ou recibir información. No caso da tablet é necesario seleccionar entre as wifis dispoñíbles, a que ten como SSID 'RaspiWiFi' e loguearse coa chave 'P4ssw0rd' (ambas sen espas).

Unha vez logueados na rede da Raspi e con todos os elementos encendidos, debemos configurar a app. Isto só é preciso facelo unha vez.

### Canais xerados pola Raspberry do termofluxómetro.

O servidor MQTT instalado na Raspi xestiona unha serie de canais para enviar ordes e recibir información. O principal canal é 'emeehl/transmitancia', que á súa vez dispón dos seguintes canais de recepción de información:

- Temperatura ambiente exterior: 'emeehl/transmitancia/SensorExterior/Ambiente'
- Temperatura parede exterior: 'emeehl/transmitancia/SensorExterior/Pared'
- Temperatura ambiente interior: 'emeehl/transmitancia/SensorInterior/Ambiente'
- Temperatura parede interior: 'emeehl/transmitancia/SensorInterior/Pared'

destes catro canais somente son necesarios o primeiro e os dous últimos. Aparte dos canais de recepción de información, existe outro canal para enviar os resultados, que á súa vez se dividen en:

- Resultados relativos ao último minuto: 'emeehl/transmitancia/resultados/minutal'
- Resultados relativos aos últimos dez minutos: 'emeehl/transmitancia/resultados/dezminutal'
- Resultados instantáneos: 'emeehl/transmitancia/resultados/instantaneo'
- Resultados do conxunto de datos até o momento: 'emeehl/transmitancia/resultados/resume'

Para interaccionar coa unidade de control e recibir datos nestes catro canais, é necesario enviar ordes ao canal disposto para iso:

- Ordes: 'emeehl/transmitancia/ordes'

as posibles ordes son: 'recibir', 'pausar', 'enviar', 'enviar1min', 'enviar10min', 'enviarResume', 'enviarInst'. Veremos enseguida como configurar a aplicación MQTT Dash.

### Configurar o canal principal



## MQTT Dash



Default (automatically connect on start up).

Note: this option is useful if you have just one connection configured.

- ☐ If you have more than one connection, you can create home screen shortcut for every connection.

To create shortcut long press on any connection in connections list.

- ☒ Keep screen on when connected to this broker

Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics. This serves as protection from unintentional metrics changing.

Name

emeehl/Transmitancia

Address

192.168.3.1

Port

1883

Enable connection encryption (SSL/TLS).

Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise

- ☐ connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.

- ☐ This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.

User name

User password

Client ID (must be unique)

mqttdash-1e2b5555

Tile size

- ☐ Small

#### Configuración de acceso ao canal MQTT do termofluxómetro

Ao abrir MQTT Dash por primeira vez aparécenos unha pantalla en negro con dous botóns na esquina superior dereita: botón de información da aplicación e botón de configuración de novas canles (tres puntos en vertical). Pulsamos no segundo e seleccionamos 'add'. A continuación apareceranos unha nova pantalla onde imos configurar os parámetros do servidor MQTT ao que nos queremos suscribir:

- Name: emeehl/transmitancia
- Address: 192.168.3.1
- Port: 1883 (deixamos o que está por defecto)

O resto das opcións podémolas deixar por defecto. Podemos decidir neste momento o tamaño dos pulsadores que imos ter na nosa aplicación, ou ben deixalo para máis adiante. En calquera caso é seleccionar entre 'small', 'medium' e 'large' entre as tres opcións que nos aparecen na parte de abaixo da pantalla. A nosa recomendación en canto a tamaño é 'medium' para dispositivos móbiles ou tablets de pequeno tamaño e 'medium' para tablets en xeral. Podemos tamén escoller a disposición en filas e columnas ou ben deixalas por defecto en auto.

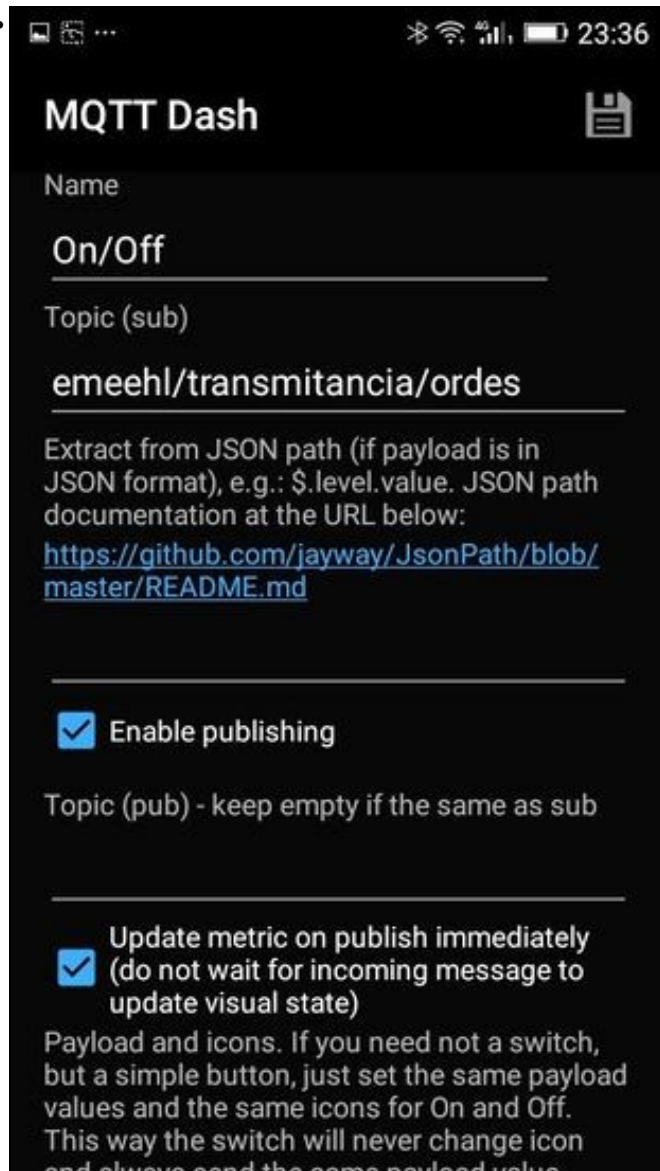
Finalmente pulsamos no icono de gardar configuración (imaxe dun disquete) e xa podemos empezar a configurar os botóns dentro do noso canal.

#### Configurar botóns de ordes e paneis de información.

Imos configurar MQTT Dash para recibir temperaturas e valores de transmitancia, así como botóns para ordes. Un exemplo do aspecto final é o da imaxe a seguir. Para chegar a facer isto temos que premer primeiro no canal que acabamos de crear e a continuación engadir novos botóns premendo na esquina superior dereita (icono de tres puntos verticais) e seleccionando 'add'.



Este vai ser un botón que se activará ou desactivará ao pulsar. Seleccionaremos por tanto 'switch/botton' no tipo de elemento. A continuación accederemos a un novo menú, no que configuraremos tanto o canal ao que accede, como as accións que realiza o botón. As dúas imaxes seguintes son autoexplicativas. Non debemos esquecer gardar os cambios ao rematar.



The screenshot shows the MQTT Dash configuration interface on a mobile device. At the top, the status bar shows the time 23:36 and various icons. The app title 'MQTT Dash' is at the top left, and a save icon is at the top right. The configuration fields are as follows:

- Name:** On/Off
- Topic (sub):** emeehl/transmitancia/orde
- JSON path:** Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:  
<https://github.com/jayway/JsonPath/blob/master/README.md>
- Enable publishing:** ☒ Enable publishing
- Topic (pub) - keep empty if the same as sub:**
- Update metric on publish immediately (do not wait for incoming message to update visual state):** ☒ Update metric on publish immediately (do not wait for incoming message to update visual state)
- Payload and icons:** If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

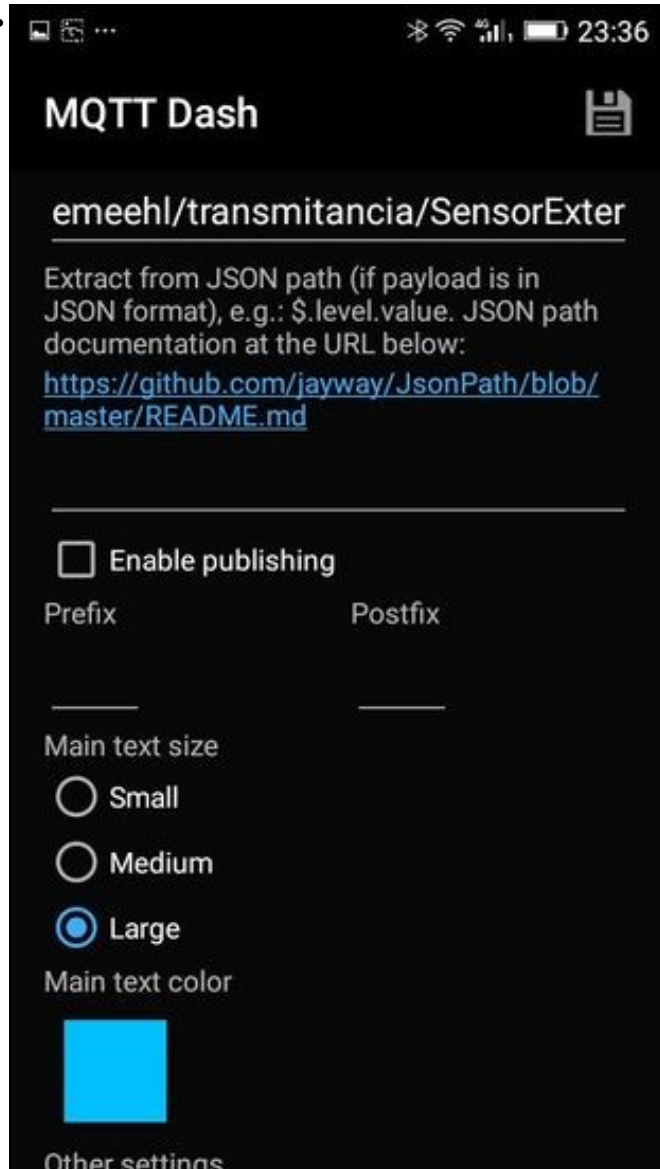




## Configuración do aspecto e ordes do botón On/Off

### Configuración de recepción de temperaturas

Imos configurar agora os dous paneis superiores da pantalla principal. Os que na imaxe xeral aparecen na parte superior e en cores azul e laranxa. Estes dous paneis mostran os valores de temperatura ambiente (exterior e interior) conforme se van recibindo dos sensores. Para crealos, en ambos casos engadimos un novo elemento á aplicación (esquina superior dereita, icono dos tres puntos verticais e seleccionar 'add'), nesta ocasión será do tipo 'text', xa que non disporá de interactividade. Só mostra información.

Para configurar a recepción de temperaturas ambiente tanto exterior como interior, debemos poñer a escoitar cada panel nos canais 'emeehl/transmitancia/SensorExterior/Ambiente' e 'emeehl/transmitancia/SensorInterior/Ambiente', tal como se mostra nas seguintes imaxes. Seleccionamos a cor na que queremos mostrar o texto e gardamos.





Topic (sub)

emeehl/transmitancia/SensorInteri

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:  
<https://github.com/jayway/JsonPath/blob/master/README.md>

☐ Enable publishing

Prefix

Postfix

Main text size

☐ Small

☐ Medium

☒ Large

Main text color

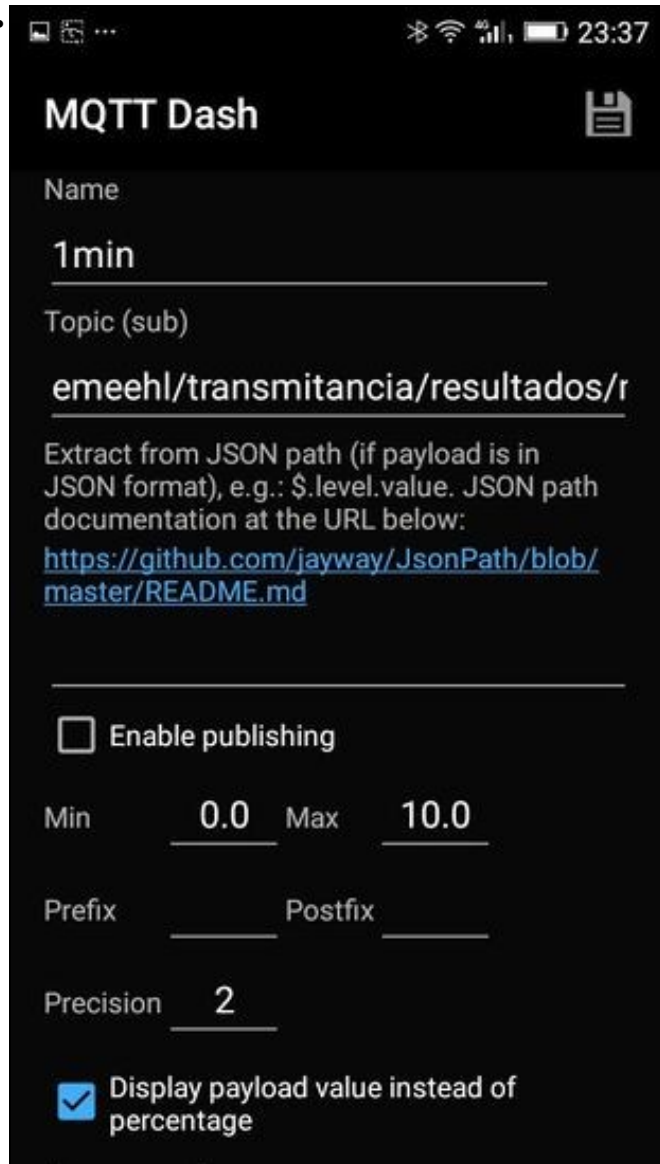


## Configuración do panel para temperatura ambiente interior

### Configuración de recepción de U-valor minutal e dezminutal

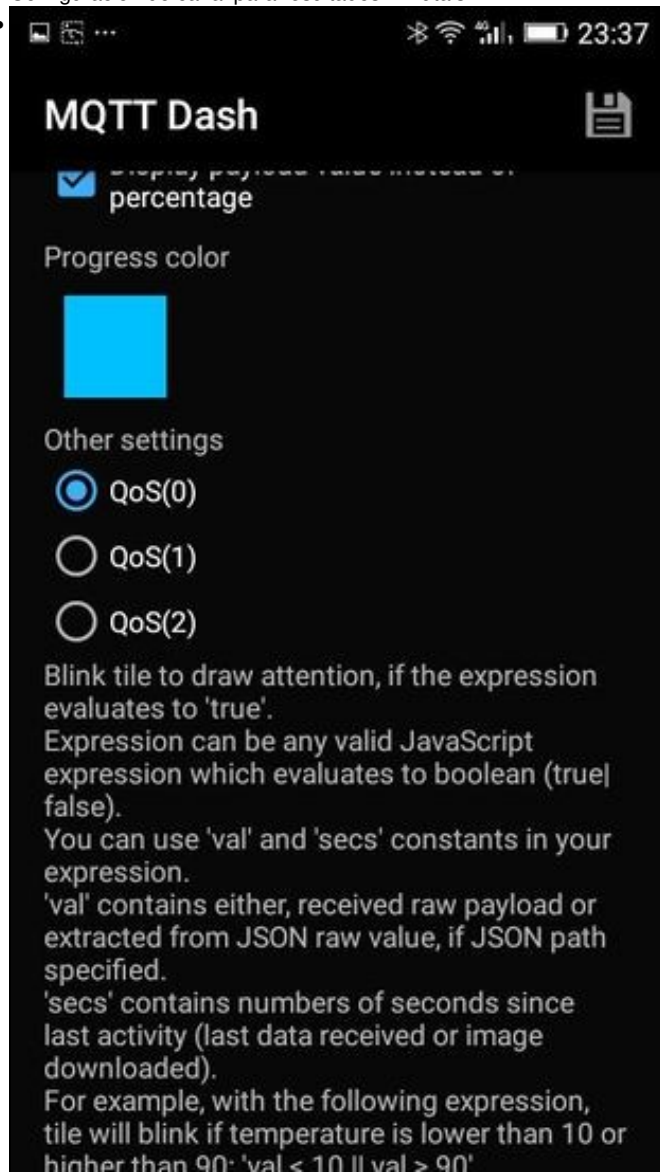
Igual que no caso anterior, seleccionaremos engadir paneis de texto, aínda que nesta ocasión lles engadiremos un arco semicircular para indicar cal é a porcentaxe do máximo valor que se mostra en cada momento.

Para configurar o panel que mostra os valores medios en cada minuto pódese seguir as seguintes imaxes. O canal no que escoita é 'emeehl/transmitancia/resultados/minutal'



The screenshot shows the MQTT Dash app interface on a mobile device. The status bar at the top indicates the time is 23:37. The app title is "MQTT Dash". The configuration fields are as follows:

- Name:** 1min
- Topic (sub):** emeehl/transmitancia/resultados/r
- Extract from JSON path:** (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:  
<https://github.com/jayway/JsonPath/blob/master/README.md>
- Enable publishing:** ☐
- Min:** 0.0 **Max:** 10.0
- Prefix:**  **Postfix:**
- Precision:** 2
- Display payload value instead of percentage:** ☒



Para o caso dos valores correspondentes a 10 minutos, faise o mesmo coa diferenza de que o canal é 'emeehl/transmitancia/resultados/10minuta'.

23:38

# MQTT Dash

displaying and changing (e.g. thermostat control temperature displaying). Payload is expected to be a string.

Name

10min

Topic (sub)

emeehl/transmitancia/resultados/10minuta

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:  
<https://github.com/jayway/JsonPath/blob/master/README.md>

☐ Enable publishing

Min

0.0

Max

10.0

Prefix

Postfix


|

Precision

2

23:39

# MQTT Dash



☐ Enable publishing

Min

0.0

Max

10.0

Prefix

Postfix

Precision

2

☒ Display payload value instead of percentage

Progress color

Other settings

☒ QoS(0)

☐ QoS(1)

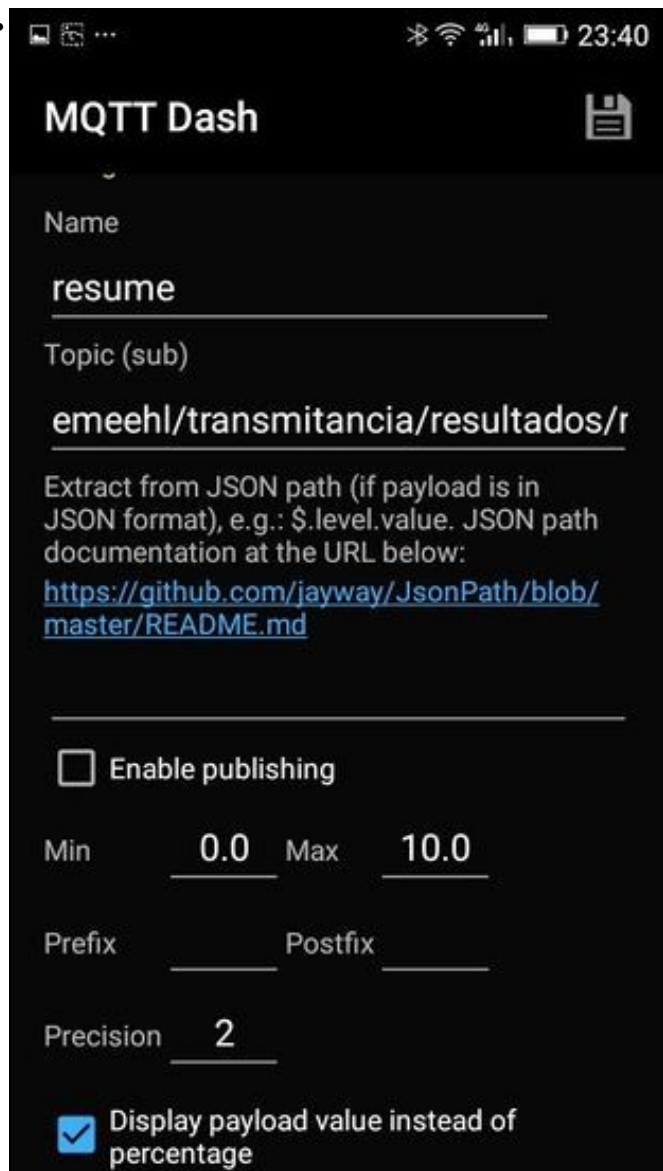
☐ QoS(2)

Blink tile to draw attention, if the expression evaluates to 'true'

## Configuración do panel para resultados 10 minutos

### Configuración de recepción de valor medio global

Para obter o valor medio calculado até o momento, temos que dotar de interactividade o novo panel. Para empezar creamos un panel como os dous anteriores. As seguintes imaxes ilustran a configuración. O canal que escoita é 'emeehl/transmitancia/resultados/resume'.



The screenshot shows the MQTT Dash mobile application interface. At the top, the status bar displays icons for Bluetooth, Wi-Fi, cellular signal, and battery, along with the time 23:40. The app title 'MQTT Dash' is in the top left, and a save icon is in the top right. The configuration fields are as follows:

- Name:** A text input field containing the word 'resume'.
- Topic (sub):** A text input field containing 'emeehl/transmitancia/resultados/r'.
- JSON Path:** A text area containing the instruction 'Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:' followed by a blue hyperlink: <https://github.com/jayway/JsonPath/blob/master/README.md>.
- Enable publishing:** A checkbox that is currently unchecked.
- Min/Max:** Two input fields. 'Min' contains '0.0' and 'Max' contains '10.0'.
- Prefix/Postfix:** Two empty input fields.
- Precision:** An input field containing the number '2'.
- Display payload value instead of percentage:** A checkbox that is checked.

23:41

# MQTT Dash

Min

0.0

Max

10.0

Prefix

Postfix

Precision

2

☒

Display payload value instead of percentage

Progress color

Other settings

☒ QoS(0)

☐ QoS(1)

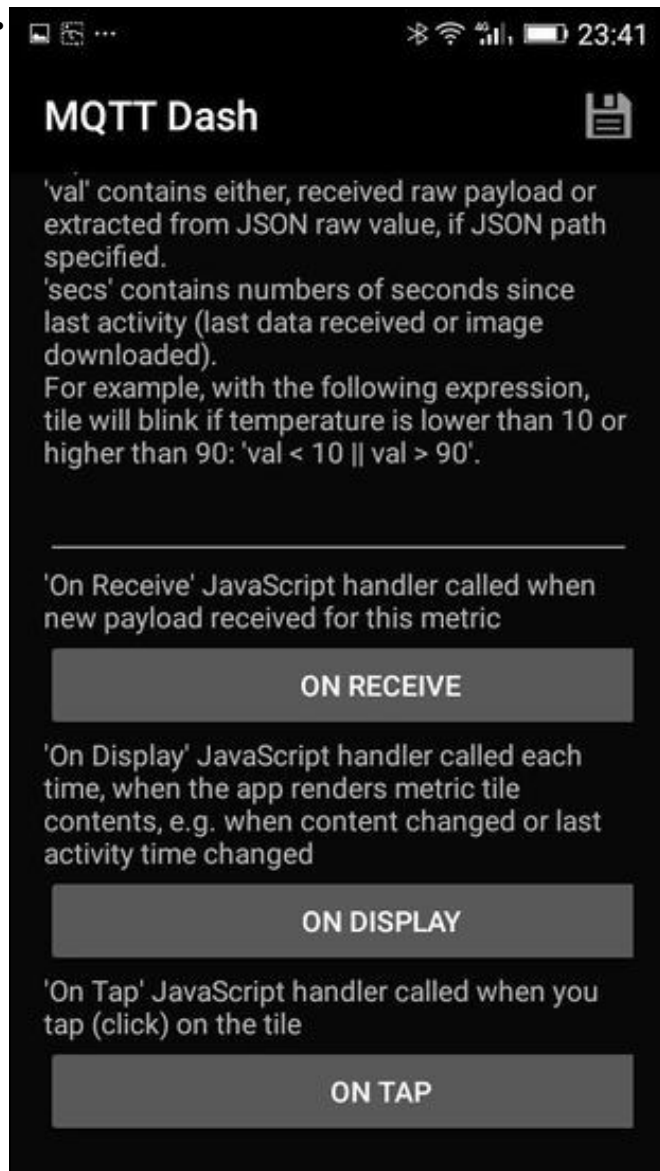
☐ QoS(2)

Blink tile to draw attention, if the expression evaluates to 'true'.

Expression can be any valid JavaScript expression which evaluates to boolean (true|false).

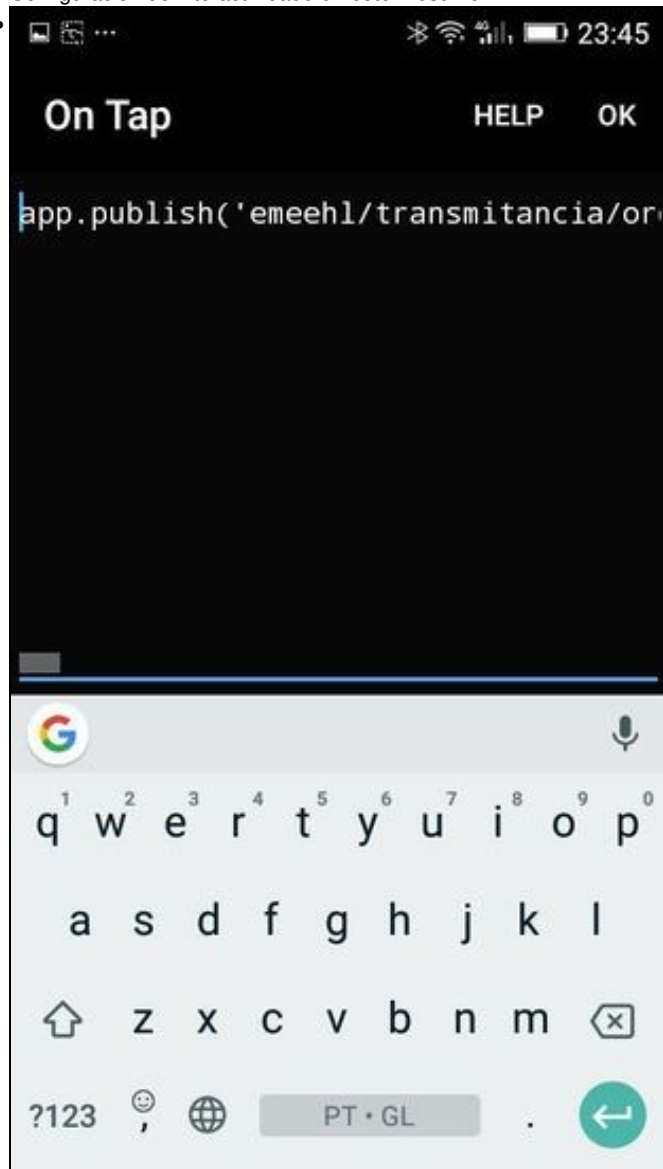
You can use 'val' and 'secs' constants in your expression

A continuación, para dotar ao panel de interactividade, debemos engadir unha orde javascript. Para iso imos abaixo da pantalla e prememos en 'On Tap'. Aí nos aparecerá unha nova pantalla na que escribiremos o texto 'app.publish('emeehl/transmitancia/ordes', 'enviarResume', false, 0)'. Ilústrase nas imaxes inferiores.





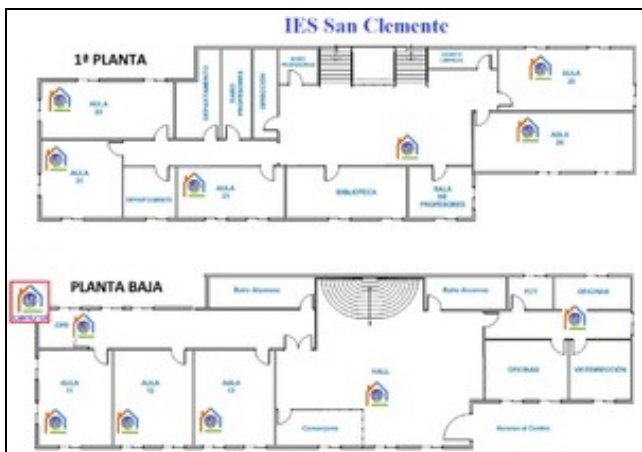
Configuración de interactividad en botón resume



---

A idea inicial do proxecto era a realización dun **dispositivo de medición**

- ◊ Intensidade sonora
- ◊ Outros gases, como: Monóxido de carbono, Benceno, Ozono, etc.



Configuración aproximada do sistema de medición de parámetros de ambiente no interior do IES San Clemente.

O dispositivo será fácilmente ampliable e, nun futuro próximo, temos pensado que as estacións sexan compatibles con máis sensores que permitan medir: Radón, Vibracións do edificio, etc. E, colocando sensores no exterior: Pluviómetro, velocidade do vento, dirección do vento, medición da luz ultravioleta, ... e, en xeral, calquera tipo de parámetro.

O sistema permite instalar unha única estación independente, que recolla e mostre vía web os datos dos sensores conectados directamente a ela, ou configurar un conxunto de estacións conectadas a unha rede e centralizar todos os datos recollidos nunha estación central. En calquera dos casos, poderemos configurar vía web todos os sensores, estacións e visualización dos datos, medias, alertas, etc.

**Este sistema de sensores ofrece unha serie de beneficios a nivel de centro como:**

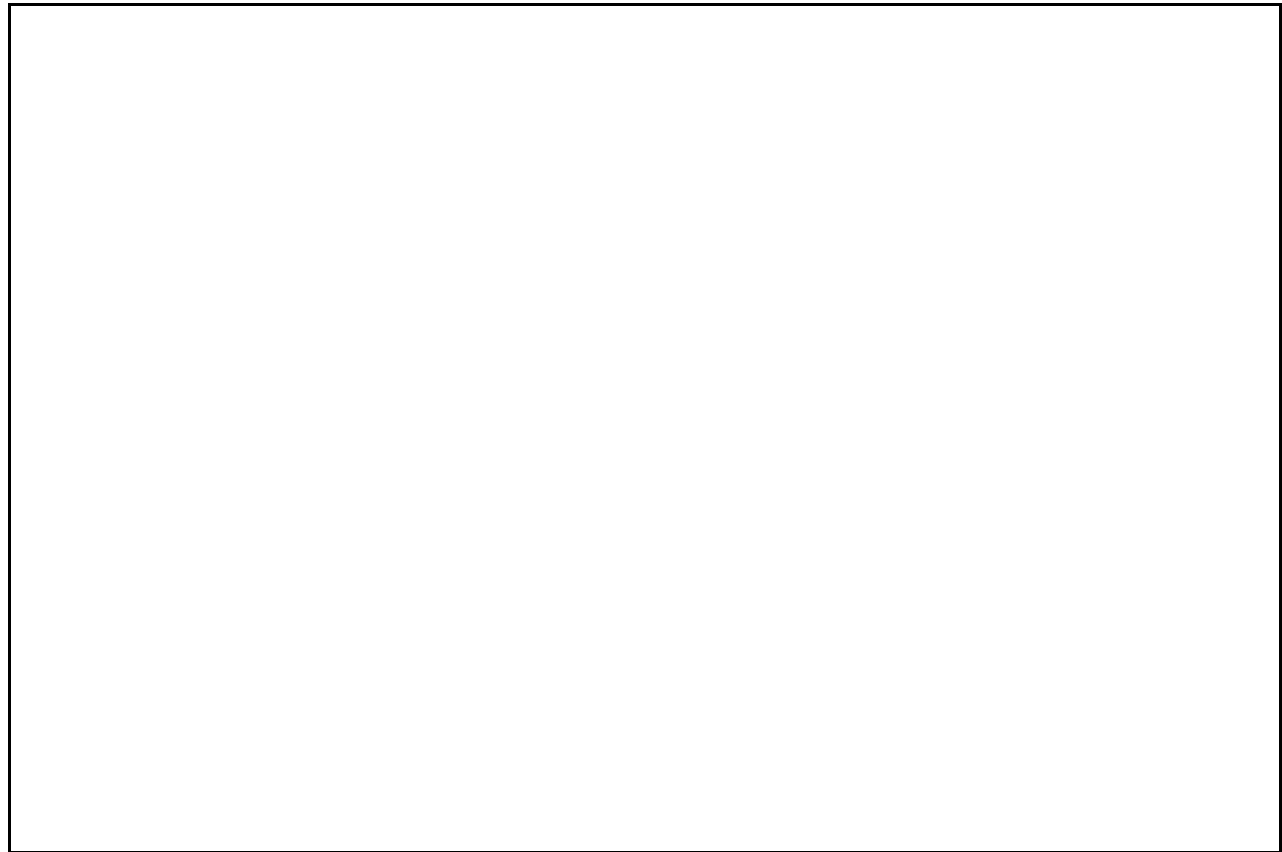
- Monitorización de parámetros da calidade do ambiente interior.
- Optimización dos tempos de acceso da calefacción.
- Recomendación da renovación do aire en aulas.
- Axuste da luminosidade en aulas.
- Alertas por altos índices de gases, intensidade sonora, temperatura e humidade fora dun rango, etc.

**Con respecto á aula, este proxecto tamén aporta múltiples beneficios:**

- O profesorado implicado adquire un alto coñecemento en: parámetros de calidade do ambiente interior, sistemas baseados en Raspberry Pi, administración de sistemas operativos Linux, programación en bash, Python, JavaScript, PHP,..., administración de bases de datos MySQL, deseño e creación de placas de circuito impreso e deseño e impresión de obxectos 3D.
- O alumnado realizará distintas prácticas baseadas nos coñecementos adquiridos polos profesores nos módulos impartidos por estes de programación web en entorno cliente e servidor e administración de sistemas operativos.
- Calquera persoa poderá empregar todo o realizado neste proxecto a nivel de *hardware* e *software*, pois trátase dun proxecto totalmente libre e público.
- Calquera membro da comunidade educativa poderá empregar a base de datos de parámetros reais que se irán xerando e almacenando na estación central Sensorialia do IES San Clemente. E así poder utilízalos para programar aplicacións web ou sacar conclusións da evolución de ditos parámetros nun entorno real.

**Evolución do deseño do *hardware***

O núcleo central das estacións é unha Raspberry Pi 3. A elección da Raspberry fronte ao Arduino foi pensando na facilidade de ampliación das estacións, a súa potencia de procesamento cando traballan as estacións de forma autónoma e a facilidade da administración e programación do sistema operativo [Raspbian](#).



Con respecto aos sensores, probouse con distintos modelos conectados por diferentes portos, como: 1-Wire, Serie, I2C, Analóxicos con conversor AD, etc. Por comodidade, e bo funcionamento para os parámetros que decidimos medir, seleccionáronse sensores conectados polo porto I2C. Estes nos ofrecen un xeito sinxelo de detectar cales están conectados, e a posterior lectura dos datos con librerías Python. Tamén se decide conectar un conversor AD I2C para permitir a conexión de ata catro sensores analóxicos para a medición de gases e son.

- Así, o *hardware* utilizado nesta primeira versión da Estación Sensoralia será o seguinte:



Raspberry Pi 3 Model B.



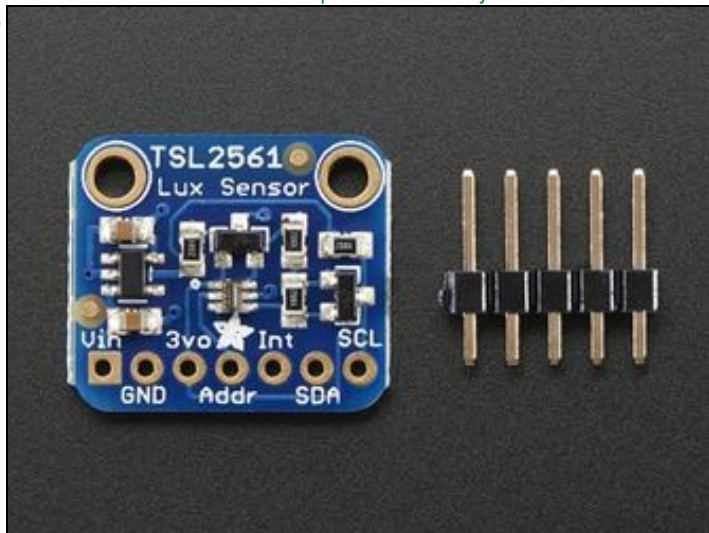
Micro SD Card (exemplo).



Fonte de Alimentación de, como mínimo 2,5A (exemplo).



Adafruit BME280 I2C or SPI Temperature Humidity Pressure Sensor.



Adafruit TSL2561 Digital Luminosity/Lux/Light Sensor Breakout.



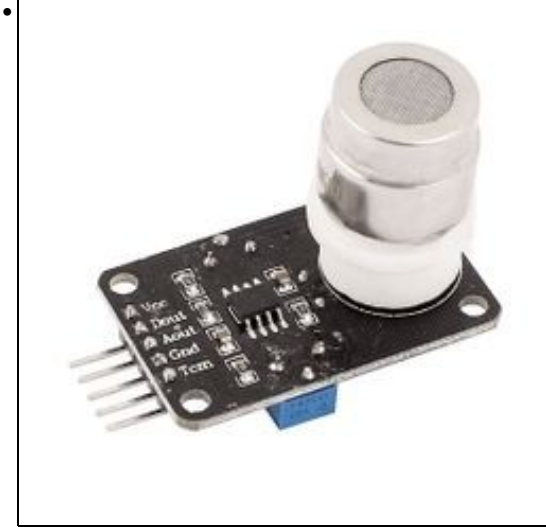
Adafruit CCS811 Air Quality Sensor Breakout - VOC and eCO<sub>2</sub>.



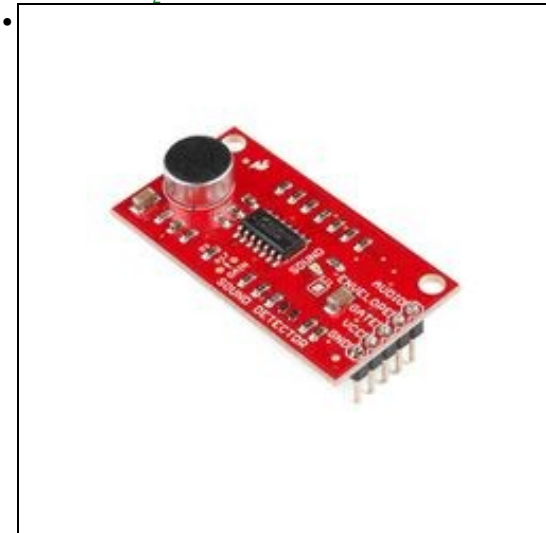
ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier.



#### Sensores de Gas MQ.



#### Sensor de CO<sub>2</sub> MG811 e Calibración.

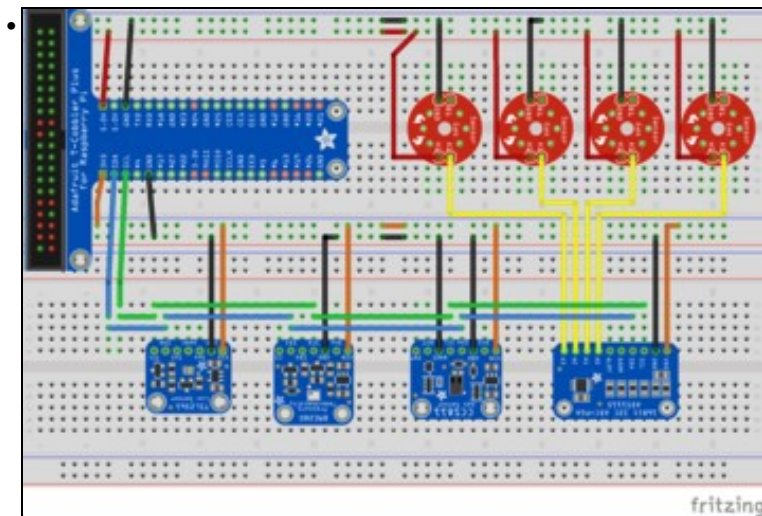


#### SparkFun Sound Detector.

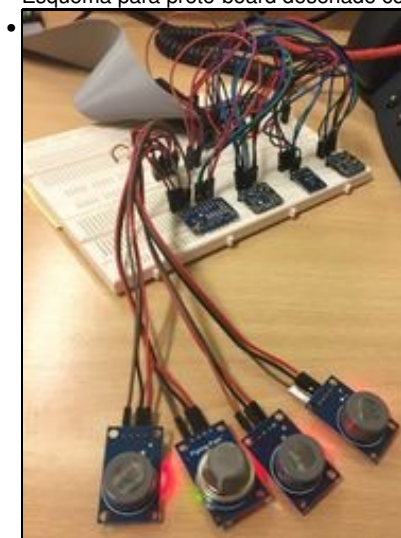
Hardware que nun futuro engadiremos:

- [Sensor CO<sub>2</sub> I2c](#) para substituir o sensor analóxico I2C CCS811.
- [Sparkfun Estación meteorolóxica](#)
- [Adafruit HUZZAH ESP8266](#) ou [Adafruit HUZZAH32 ESP32](#) para evitar o uso da Raspberry Pi en moitos sensores, e así mellorar en integración e gasto enerxético. Este dispositivo permite conectar ao seu GPIO múltiples sensores analóxicos, dixitais, conetados ao porto serie, porto I2C, etc, e enviar os datos recollidos vía Wifi a estación máis cercana.
- Os pasos dados para chegar ao montaxe final pódese resumir así:





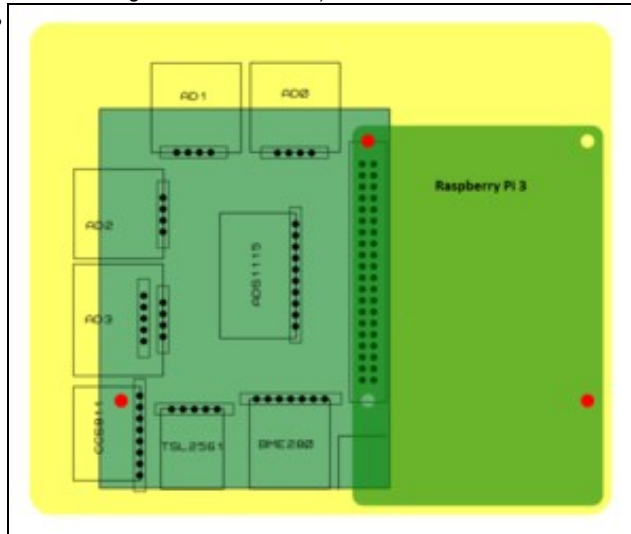
Esquema para proto-board deseñado có software Fritzing (Sensoralia-Proto-Board.fzz)



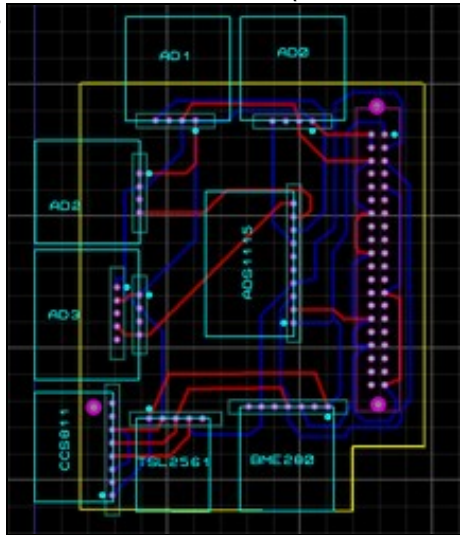
Montaxe dos sensores na proto-board.



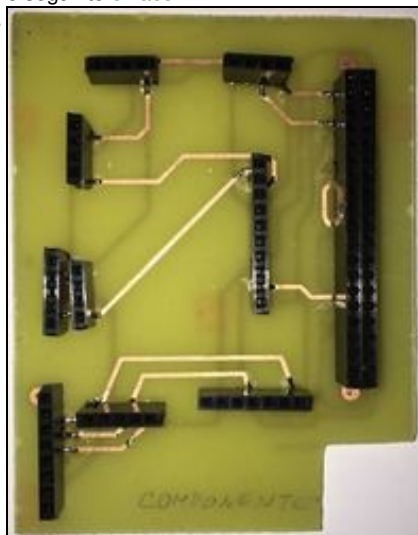
Montaxe dos sensores en placa de prototipos empregando unha **HAT** (*Hardware Attached on Top*). Comprobamos que **NON** é boa solución pois a calor desprendida pola Raspberry altera as medicións significativamente (a temperatura sube uns 10°C, a humidade baixa e a medición de gases vese alterada).



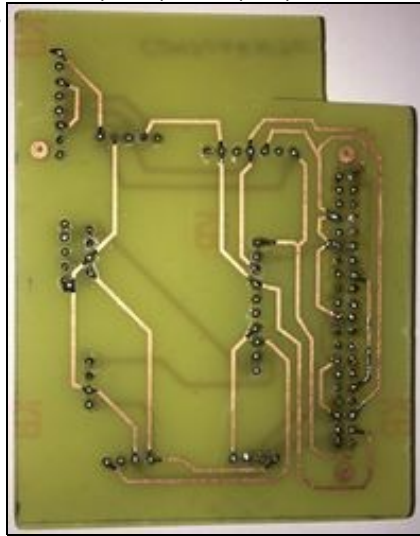
Decídese deseñar unha **HAS** (*Hardware Attached on Side*), evitando así a alteración das medidas pola calor desprendida na Raspberry.



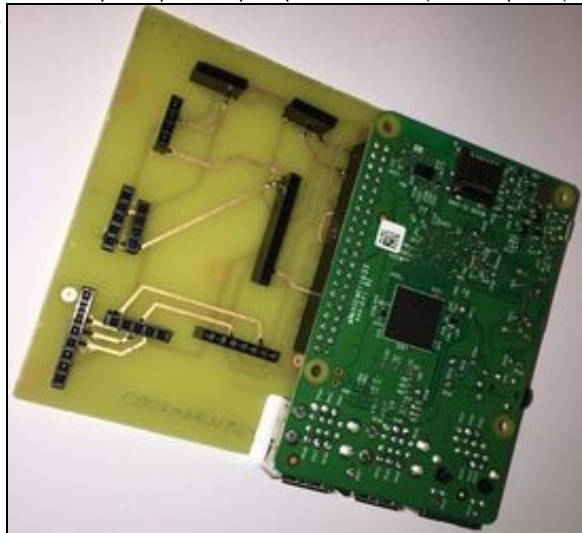
Deseño da placa HAS coa axuda do profesor *Paco Lado Vara* do CIFP Politécnico de Santiago. Pódese baixar o deseño **Proteus Ares** dende o seguinte [enlace](#).



Imaxe do prototipo HAS pola parte onde irán "pinchados" os sensores (cara de compoñentes).



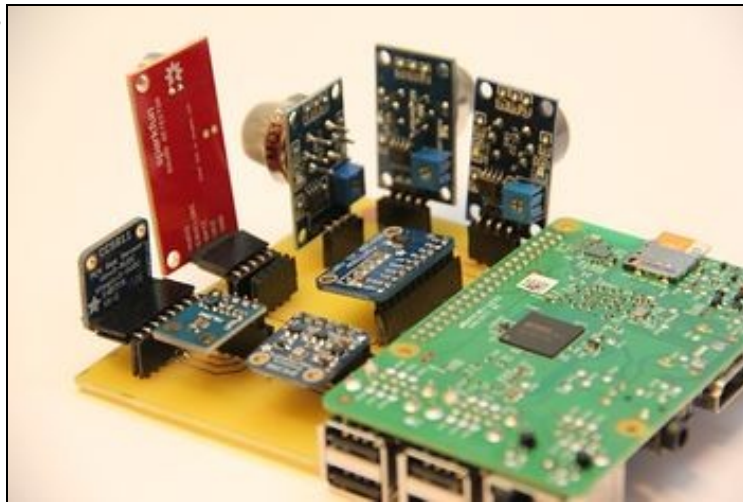
Imaxe do prototipo HAS pola parte de atrás (cara de pistas)



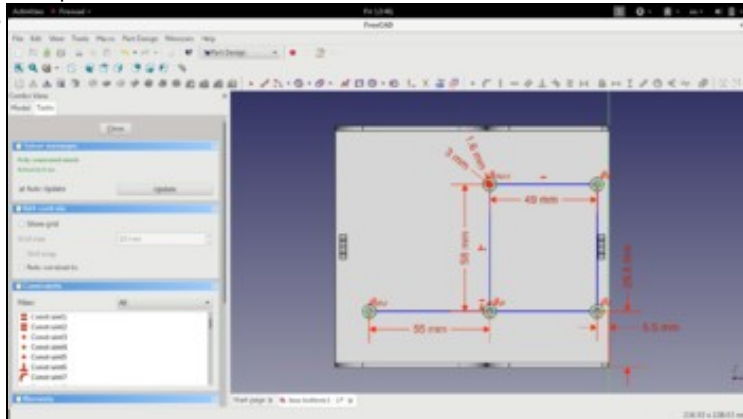
Imaxe da HAS xa conectada á Raspberry. Os sensores quedan no outro lado para conseguir que a estación teña a menor altura posible (gañamos a altura do conector GPIO).



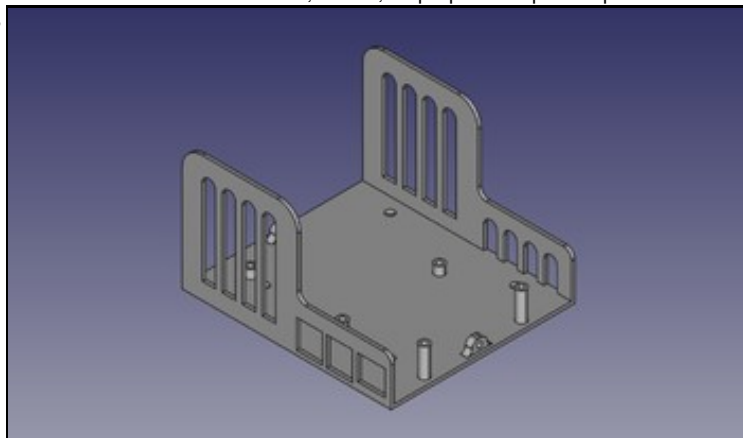
Imaxe dos sensores conectados na HAS.



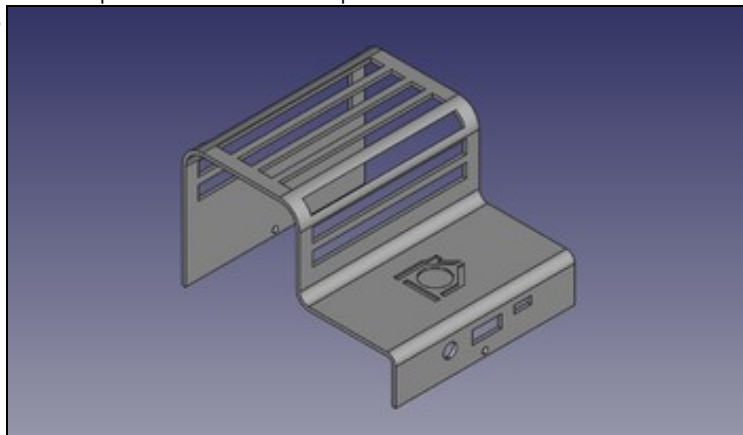
Vista posterior dos sensores conectados na HAS.



Coa axuda dunha empresa externa, deséñase unha caixa a medida para a estación Sensorialia. Pódense descargar os arquivos de deseño da caixa en formato [FreeCAD](#) e, tamén, xa preparados para imprimir en formato [STL](#).

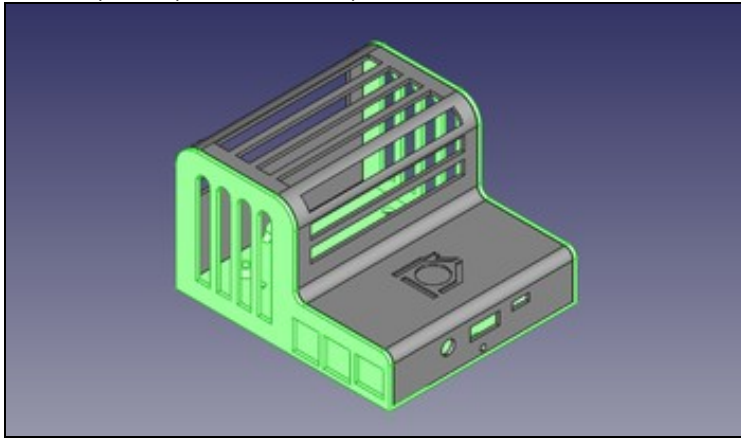


Imaxe da parte inferior da caixa no proceso do deseño.





Imaxe da parte superior da caixa no proceso do deseño.



Imaxe completa da caixa no proceso do deseño.



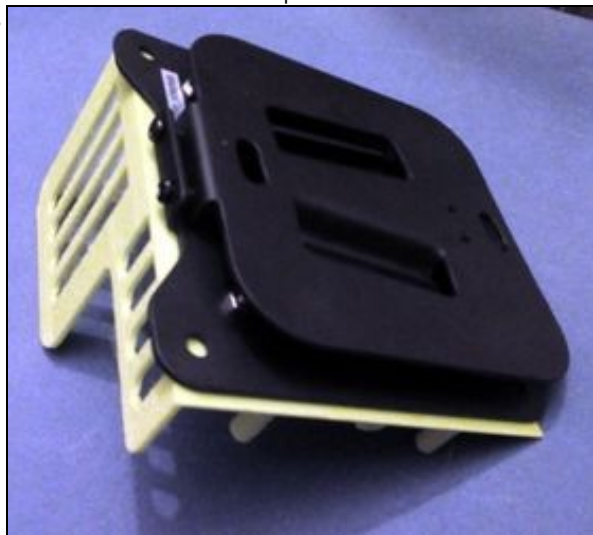
Imaxe da parte inferior da caixa recién impresa e coa Raspberry xa anclada nela (esta peza tarda unhas 3 horas en imprimirse).



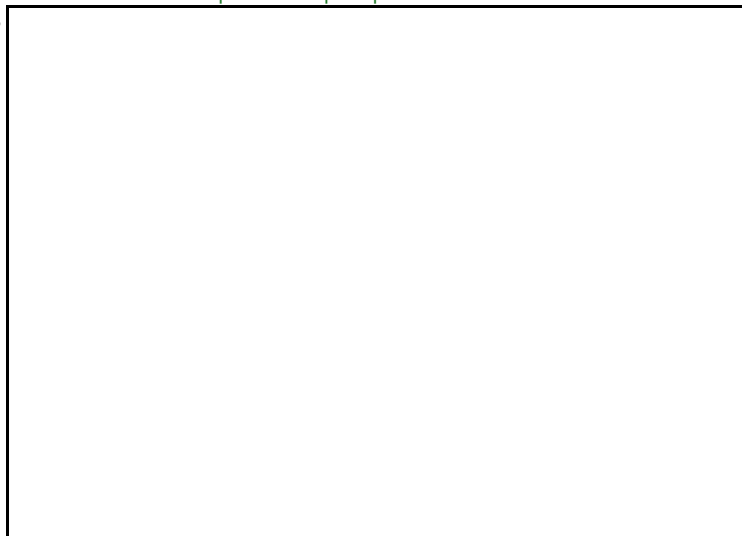
Imaxe da parte superior da caixa recién impresa (esta peza tarda unhas 4 horas en imprimirse).



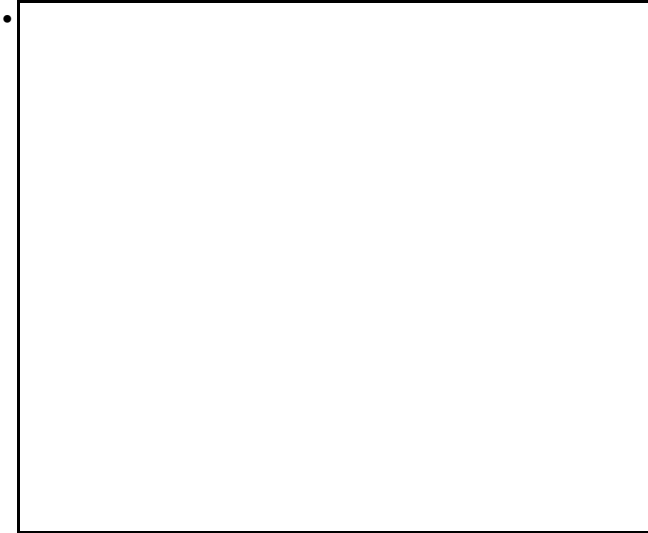
Imaxe da caixa totalmente impresa.



Imaxe da caixa cun soporte vesa para pared.



Aquí vemos como queda o interior da estación sensorialia unha vez anclada á caixa. Para anclar a Raspberry e a tarxeta de sensores á caixa empréganse parafusos de 6mm con rosca M3 ([Referencia Rational Stock: FP45721](#)).



Imaxe da estación sensorialia case rematada de montar.

### Instalación dunha estación Raspberry

En cada zona que queramos monitorizar debemos instalar unha estación cós sensores necesarios para poder medilos.

Para montar cada unha destas estacións, seguiremos estes pasos:

#### 1.- Facernos có *hardware* necesario:

- Unha Raspberry Pi 3 B.
- Unha SD Card para o sistema operativo.
- Unha fonte de alimentación para Raspberry Pi 3 de, como mínimo, 2,5A e, a poder ser, tamén con botón de aceso.
- A tarxeta Sensorialia de conexión dos sensores ao GPIO.
- Os sensores necesarios para medir os parámetros que nos interesen.
- A caixa Sensorialia (opcional).

2.- Baixar o documento [sensorialia.zip](#) do [servidor de arquivos](#), descomprimila para obter a imaxe do sistema sensorialia.img e, logo, restaurar a imaxe na SD Card empregando algún *software* para tal efecto como o [Win32DiskImager](#).

3.- Montar todo o sistema, empregando a Raspberry Pi 3, os sensores escollidos, a tarxeta HAS e a caixa a medida realizada para a estación Sensorialia (opcional).

4.- Iniciar a estación xa montada e acceder a ela por rede, utilizando un PC configurándolle unha IP axeitada para acceder á Raspberry Pi 3 dende un navegador, sabendo que os parámetros de inicio configurados na Estación son:

```
#Configuracion da eth0
IP/MS : 192.168.200.200/24
PE : 192.168.200.1
DNS : 8.8.8.8
```

Hai que acceder por Ethernet, pois a estación NON se conecta automaticamente a ningunha Wifi existente. A conexión por Wifi pode configurarse máis adiante có asistente web.

Web de acceso ao asistente de configuración: <http://192.168.200.200>

#### 5.- Unha vez que accedemos, poderemos configurar os seguintes parámetros:

- Configuración inicial :
  - ◊ Nome da estación, dominio e localización.
  - ◊ Rede Wifi á que nos interesa conectala (se é o caso).
  - ◊ Configuración IP da rede Ethernet e Wifi.



- Se a estación traballa ela soa ou nun sistema de múltiples estacións.
- Sensores conectados á estación e a súa configuración.
- Correo que emprega a estación para enviar notificacións e tamén o *mail* (ou *mails*) aos que enviará ditas notificacións.

6.-Cando xa a estación está funcionando:

- Podemos acceder vía web á configuración da nosa estación para ver as medidas tomadas e tamén para acceder á súa configuración. O nome de usuario e contrasinal empregado para administrar a estación vía web, por defecto, é : **admin/Sensores.**, a cal teremos que cambiar no primeiro *login*.
- Sen necesidade de iniciar sesión, calquera usuario poderá configurar unha serie de parámetros de visualización dos valores e gráficas que se van lendo.

## Videos de instalación da Raspberry en Modo Autónomo

### INSTALACIÓN DA RASPBERRY EN MODO AUTÓNOMO:

### XESTIÓN DA RASPBERRY EN MODO AUTÓNOMO:

#### Instalación do servidor web central Sensorialia

Xa vimos nos apartados anteriores cómo configurar unha estación Raspberry coa aplicación web funcionando. Agora imos ver cómo podemos instalar a aplicación web nun servidor web con Apache en MySQL. Neste caso farémolo cunha distribución Linux Debian 9.

#### Instalación dun servidor LAMP nunha máquina con Debian 9

#### Cómo instalar Apache e MariaDB en Debian

#### Creación dun usuario e base de datos en MariaDB

- Dende o phpMyadmin podemos crear un usuario dende unha ventana de SQL ou dende a propia interfaz de phpMyAdmin. Máis información [aquí](#).
- Si o queremos facer con comandos SQL podería ser algo así:

```
CREATE USER 'sensorialia'@'%' IDENTIFIED BY 'lacontraseña';
GRANT USAGE ON *.* TO 'sensorialia'@'%' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX
CREATE DATABASE IF NOT EXISTS `sensorialia`;
GRANT ALL PRIVILEGES ON `sensorialia`.* TO 'sensorialia'@'%';
```

#### Configuración do servidor web e instalación da aplicación Sensorialia dende as fontes.

Cando xa temos instalado o servidor web, o MySQL e creado o usuario de MySQL ou MariaDB, imos a facer a instalación do código da web Sensorialia.

```
# Unha vez conectados ó sistema como root faremos as seguintes tarefas:

# Imos crear un dominio www.misensorialia.local, có que temos que facer resolución DNS local para poder acceder.
# Editaremos o arquivo /etc/hosts
nano /etc/hosts

# E engadiremos unha liña como a seguinte:(sustituir 192.168.1.20 pola IP do servidor)
192.168.1.20    www.misensorialia.local    misensorialia.local

# En Windows o arquivo hosts atópase en: C:\windows\system32\drivers\etc\hosts

# Probaremos a facer un ping á dirección www.misensorialia.local ou misensorialia.local e comprobaremos que resolve correctamente:
ping www.misensorialia.local

# Unha vez feito o paso previo da resolución DNS local, xa podemos continuar có proceso de configuración do servidor web Apache.

#Agora imos crear o cartafol en /var/www/misensorlia.local ónde descargar o contido de sensorialia.zip
```

```

mkdir /var/www/misensoralia.local

# Accedemos a ese cartafol, descargamos sensoralia.zip e o descomprimos o .zip
cd /var/www/misensoralia.local
wget url_descarga_sensoralia.zip

# Descomprimos o arquivo sensoralia.zip:
unzip sensoralia.zip

# Accedemos á carpeta cgi
cd cgi

# Damos permisos ó arquivo ajustarpermisos.sh
chmod 755 ajustarpermisos.sh

# Executamos ese arquivo:
./ajustarpermisos.sh

# Agora configuraremos no servidor web Apache o novo directorio e o dominio virtual:

# Crearemos un arquivo chamado sensoralia.conf en /etc/apache2/sites-available/
nano /etc/apache2/sites-available/sensoralia.conf

# Contenido do arquivo /etc/apache2/sites-available/sensoralia.conf
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName www.misensoralia.local
    ServerAlias misensoralia.local

    DocumentRoot /var/www/misensoralia.local/public

    <Directory />
        Options FollowSymLinks
        AllowOverride All
    </Directory>
    <Directory /var/www/misensoralia.local/public>
        Options FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error-misensoralia.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access-misensoralia.log combined
</VirtualHost>

# Gravamos o arquivo

# Habilitamos o novo servidor web.
a2ensite sensoralia.conf

# Reiniciamos o servidor web, ou recargamos a configuración:
service apache2 reload / service apache2 restart

# A partir desde intre xa nos podemos conectar cá url http://www.misensoralia.local e seguir os pasos de configuración.

```

## Videos de instalación e configuración de Sensoralia nun servidor web central

### INSTALACIÓN DE SENSORALIA NUN SERVIDOR WEB CENTRAL:

### CONFIGURACIÓN DE SENSORALIA NUN SERVIDOR WEB CENTRAL:

## Videos de instalación de Raspberry como cliente dun servidor web central

### INSTALACIÓN dunha estación Raspberry como cliente dun servidor web central:

### Instalación "dende cero" do sistema ---Por se é de interese ---

## Descargar e instalar Raspbian

- Descargamos e instalamos la versión RASPBIAN STRETCH LITE. Para la instalación desde Windows utilizamos el *software* **Etcher**.

**Nota:** Cuando se encuentre todo configurado crearemos una imagen, para clonar en el resto de los sensores, utilizando el [siguiente manual](#). La versión de "instalación rápida" se realizó de este modo.

## Configuración desde cero con imagen Raspbian

Configurar los siguientes parámetros:

- Usuarios + password :

El password por defecto de pi es "raspberry" Tanto al usuario **pi** como a **root** les configuramos el password **abc123..**

```
#Le cambiamos el password a pi
$ passwd
#Nos pasamos a root
$ sudo su
#Le cambiamos el password a root
$ passwd

# Como usuario root configurar
$ raspi-config
# 4 localization options
## I1 change locale : es_ES.UTF-8 ... Deixar só esta marcada
# 4 localization options
## I2 change time zone : Europe Madrid
# 4 localization options
## I3 change keyboard layout : PC Genérico 105 teclas - Español Teclas Windows
# 5 Interfacing Options
## P2 SSH
# 5 Interfacing Options
## P5 i2C

# Bajamos velocidad del i2C
$ sudo nano /boot/config.txt
dtparam=i2c_baudrate=10000

# Reiniciamos
$ reboot
```

- Para la configuración inicial de la red copiar el contenido del archivo **/etc/network/interfaces** siguiente:

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

# Interfaz eth0
auto eth0
#allow-hotplug eth0
iface eth0 inet static
    address 10.0.0.16
    netmask 255.255.0.0
    gateway 10.0.0.254
    dns-nameservers 8.8.8.8
```

- Actualizamos el equipo

```
# Actualizamos paquetes y distribución
apt update
apt upgrade
apt dist-upgrade
```

```
# Instalamos rpi-update para actualizar firmware de raspberry
apt install rpi-update
rpi-update

# Hacemos un reboot al terminar
reboot
```

- Hacemos copias de seguridad de algunos archivos que luego se van a cambiar:

```
$ cp /etc/hosts /etc/hosts.bak
$ cp /etc/hostname /etc/hostname.bak
$ cp /etc/network/interfaces /etc/network/interfaces.bak
$ cp /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/wpa_supplicant.conf.bak
```

- Instalamos pip y pip3 para la instalación posterior de módulos Python:

```
# Instalamos herramientas pip, whois(mkpasswd) para cifrar contraseñas e zip
$ apt install python-pip python3-pip whois zip
# Para los sensores:
$ apt install python-smbus i2c-tools build-essential python-dev git -y
```

- Instalación de módulos Python que vamos a necesitar:

```
# Herramientas netifaces para descubrir configuración de red y requests para realizar envíos al servidor por PHP
$ pip3 install netifaces requests smb2
# Librerías para los sensores:
$ pip install Adafruit_CCS811 RPi.bme280 tsl2561 Adafruit_ADS1x15
##
#MySQLdb para trabajar con las bases de datos desde Python
#$ apt install python3-mysqldb
```

- Instalación de apache2 php7 mariadb y phpmyadmin:

```
# Instalar como root

apt update
apt upgrade
apt install apache2 mariadb-server php7.0 libapache2-mod-php7.0 php7.0-mysql php7.0-gd php7.0-opcache phpmyadmin

mysql_secure_installation

....

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n]
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

- Modificar la configuración de Apache habilitando mod rewrite y AllowOverride All:

```
a2enmod rewrite

nano /etc/apache2/sites-enabled/000-default.conf

# Poner la siguiente configuración:
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/public

    <Directory "/var/www/public">
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all Granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

service apache2 restart
```

- Subir por FTP el contenido del archivo fuentesSensoralia.zip a la carpeta /var/www:

- Poner IPs por defecto:

Poner IPs por defecto: 192.168.200.200/24 eth0 192.168.200.201/24 wifi

AL COPIAR LOS FICHEROS HACER ESTOS CAMBIOS. chmod 755 /var/www/cgi/resetearInstalacionRaspberry.sh

Ejecutar como root para reiniciar la instalación: /var/www/cgi/resetearInstalacionRaspberry.sh

Hacer enlace simbólico desde el home de pi ln -s /var/www/www </source>

- Contenido de archivo clearlogs.sh (permisos 755):

```
#!/bin/bash
echo "----- Borrando Logs -----"
for i in /var/log/*; do cat /dev/null > $i; done
```

```

for i in /var/log/apache2/*; do cat /dev/null > $i; done
for i in /var/log/mysql/*; do cat /dev/null > $i; done

echo "----- Borrando .gz-----"
find /var/log/. -type f -regex ".*\.gz$" -delete
find /var/log/. -type f -regex ".*\.1$" -delete
find /var/log/. -type f -regex ".*\.2$" -delete
echo "----- Logs Borrados -----"

# Permisos de ejecución
chmod 755 clearlogs.sh

```

- Contenido de archivo updates.sh (permisos 755):

```

#!/bin/bash
clear
apt-get update
apt-get upgrade
apt-get dist-upgrade
apt-get autoremove
rpi-update

# Permisos de ejecución
chmod 755 updates.sh

# Creamos una tarea de cron como root:
crontab -e

# Añadimos la última línea. Comprobar que tenemos esas 2 líneas en el cron de la Raspberry:
@reboot sleep 90; /usr/bin/python3 /var/www/cgi/inicio.py &
0 2 */15 * * /root/clearlogs.sh > /dev/null 2>&1 # Borrado de logs cada 15 días

# You may uncomment the following lines if you want `ls` to be colorized:
export LS_OPTIONS='--color=auto'
eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

```

- Usuarios e contrasinais por defecto na Raspberry:

- mySQL - **root:sensoralia-**,
- mySQL - **sensoralia:sensordb-**,
- SSH - **pi:abc123..**
- SSH - **root:abc123..** (acceso remoto inhabilitado por defecto)
- Web - **admin:Sensores.**

## Licenza

A Consellería de Cultura, Educación e Ordenación Universitaria de Galicia ten os dereitos de edición e difusión deste proxecto de Innovación.