The Raspberry Pi logo, featuring a stylized raspberry fruit with two green leaves, is centered in the background. The fruit is composed of several red, oval-shaped segments, and the leaves are a vibrant green with black outlines.

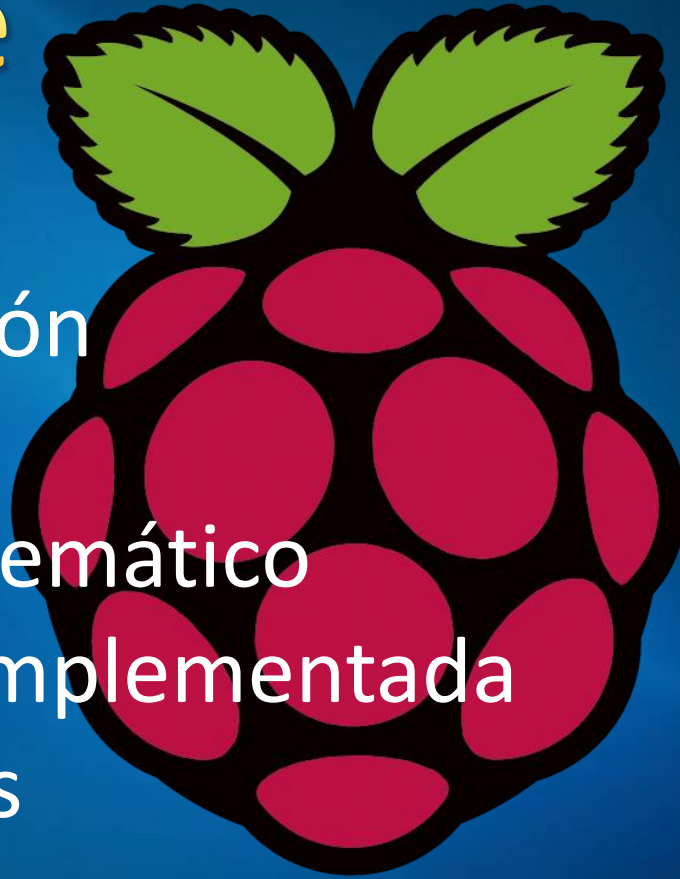
# Proyecto Raspberry Pi: Estación Meteorológica

David Fernández Torres

Master de Ingeniería Industrial

# Índice

- 1.Introducción
- 2.Objetivos
- 3.Flujo esquemático
- 4.Solución implementada
- 5.Resultados
- 6.Conclusiones



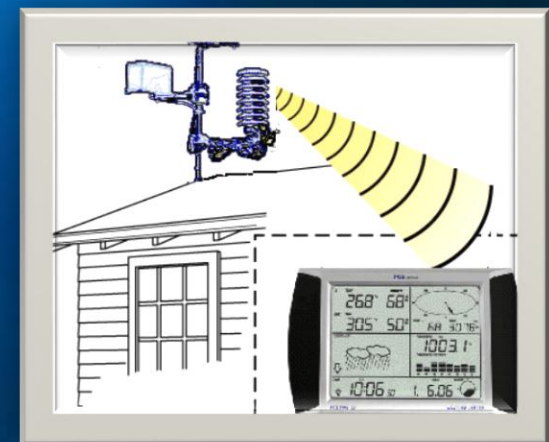
# Introducción

- Una estación meteorológica es un aparato electrónico que es capaz de medir diferentes variables relacionadas con la climatología.

Temperatura,  
Humedad



Viento, Presión,  
Predicción del  
tiempo



# Objetivos

- Conseguir realizar una estación con los siguientes requisitos básicos:
  - Variables a medir:

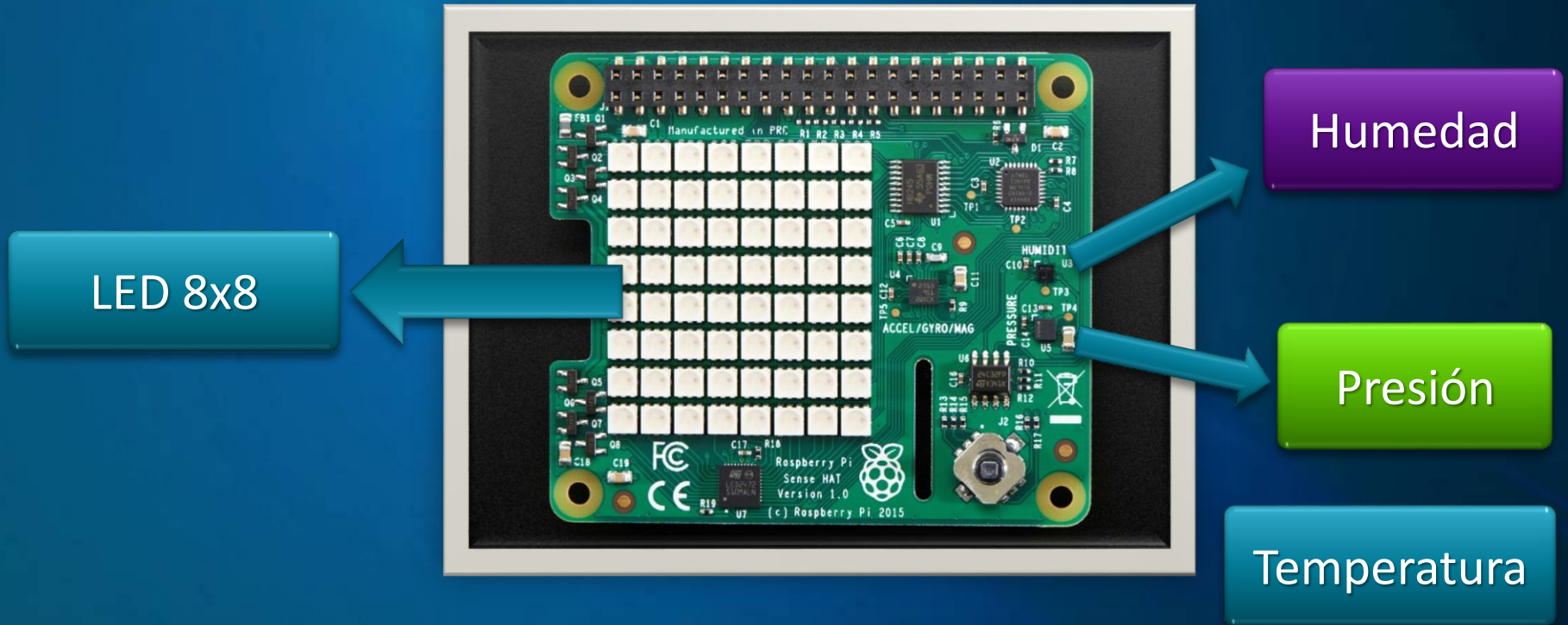
Temperatura

Presión

Humedad

- Mostrar los datos por pantalla en la Raspberry.
- Esperar cada cierto tiempo para tomar muestras
- Almacenar los datos en formato tabla Excel.

# Sense Hat



# Flujo esquemático

Iniciamos  
variables

Realizamos login  
con Google Drive

Leemos valores  
de los sensores

Establecemos un  
tiempo de  
muestreo

Subimos los valores  
a Google Drive

Mostramos por  
ambas pantallas los  
valores

Escribimos en la tabla  
nuestras mediciones y  
la hora





# Solución implementada. Paquetes

- Si queremos que todo funcione necesitamos instalar los siguientes paquetes:

- Pycrypto
- Python\_openssl
- pyopenssl
- Oauth2client (versión 1.5.2)
- gspread



# Solución implementada. Medidas

- Para medir las magnitudes usaremos las funciones vistas en las prácticas:
  - Temperatura: `sense.get_temperatura()`
  - Presión: `sense.get_pressure()`
  - Humedad: `sense.get_humidity()`
  - Usaremos la función `round()` para redondear estos valores y no tener tantos decimales.





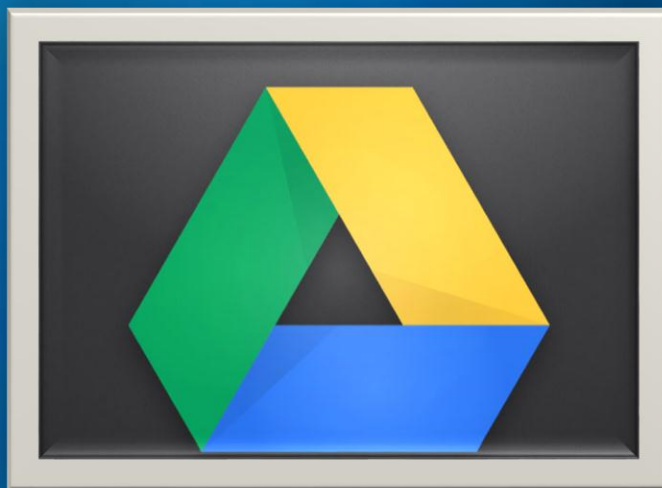
# Solución implementada. Pantalla



- Para la pantalla usaremos la función `sense.show_message`. Cada variable la pondremos de un color para reconocerla mejor.
- También mostraremos en el Shell del programa los valores que obtenemos con la función `Print`.
- La espera entre mediciones la ejecutaremos con `time.sleep()`

# Solución implementada. Datos


- Almacenaremos los datos en forma de tabla de Excel. Para ello usaremos Google Drive.



- Crear un protocolo de conexión entre la Raspberry y G. Drive, esto lo hacemos con la herramienta G. Developers.

# Resultados

- Hemos conseguido lo que buscábamos:
  - Somos capaces de medir Temperatura, Presión y Humedad.
  - Mostramos los valores por pantalla de manera clara.
  - Controlamos el tiempo de muestreo.



```
La Temperatura ambiente (C) es de: 31.2
La Humedad es de: 41.6
La Presión es de: 1020.1

Se introdujo una nueva fila con los datos en el archivo raspberry
La Temperatura ambiente (C) es de: 31.2
La Humedad es de: 40.9
La Presión es de: 1020.0

Se introdujo una nueva fila con los datos en el archivo raspberry
```

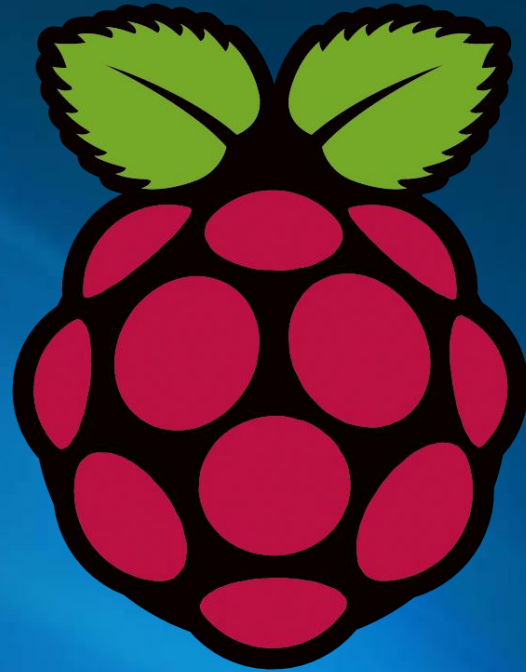
# Resultados

- Guardamos la información en Internet y lo podemos consultar estemos dónde estemos.

	A	B	C	D	E
1	Hora	Temperatura	Humedad	Presión	
2		31.00	35.30	1023.8	Máximo
3		29.00	32.40	1023.5	Minimo
4		30.07	33.24	1023.6	Media
5	2017-01-25 21:20:40.591555	29.00	35.00	1023.7	
6	2017-01-25 21:21:44.677746	30	35.00	1023.8	
7	2017-01-25 21:22:48.610268	29.20	35.30	1023.7	
8	2017-01-25 22:25:25.369479	31.00	32.60	1023.7	
9	2017-01-25 22:35:54.438830	30.00	33.10	1023.6	
10	2017-01-25 22:37:59.151104	30.00	32.40	1023.6	
11	2017-01-25 22:39:01.580094	30.40	32.70	1023.6	
12	2017-01-25 22:42:08.186793	30.30	33.20	1023.5	

# Conclusiones

- Se ha podido crear una aplicación de utilidad real en la Raspberry Pi.
- Posibles campos de utilización: asegurarse de que un lugar esté siempre en unas condiciones concretas como neveras industriales, o en procesos donde las condiciones ambientales no deben cambiar.
- Puntos a mejorar: Utilización de sensores aún mas precisos. Comprar otros tipos de sensores como de velocidad y dirección del viento, luminosidad...



¡GRACIAS POR SU  
ATENCIÓN!