

TRABAJO PRACTICO N°4



MATERIA: Sensores y Actuadores

PROFESORES: Ing. Jorge Morales

Tec.Sup. Gonzalo Vera

ALUMNA: Maria Carolina Nis

1) Que es Amazon web service IoT

AWS IoT Core permite conectar dispositivos a servicios de **AWS** y a otros dispositivos, proteger datos e interacciones, procesar y actuar sobre datos de dispositivos y habilitar las aplicaciones para que interactúen con dispositivos aunque no estén conectados.

Más que ver las soluciones de IoT como una pirámide, donde los datos en el campo se generan para ir subiendo hasta llegar a la analítica, AWS IoT propone un círculo en el cual hay tres elementos principales:

- Data Services
- Device Software
- Control Services

Como se muestra a continuación:



2) Que es Arduino IoT cloud

Si juntamos una placa Arduino e IoT, tendremos la solución Arduino IOT, pero el plus o novedad que ofrece Arduino es que todo esto lo tenemos en el cloud.

Arduino nos ofrece con esta solución hardware, firmware, servicios en la nube y conocimiento.

La plataforma Arduino IoT Cloud puede ser programada con bocetos elaborados de forma más sencilla y rápida por el sistema. Generando **esquemas automáticos** cuando se configure algo nuevo, lo que ayudará a un desarrollador a pasar de desempaquetar un tablero de control o dashboard a un dispositivo operativo dentro de cinco minutos.

Arduino IoT Cloud también proporciona otros métodos de interacción, incluyendo API REST HTTP, MQTT, herramientas de línea de comandos, Javascript y Websockets, por mencionar algunos.

También para esta nueva plataforma, se deberán usar las placas de la familia MKR, que suministra una forma simplificada de crear nodos de IoT y dispositivos de vanguardia. Estos utilizan una gama

de opciones de conectividad y compatibilidad con hardware de terceros, pasarelas y sistemas en la nube.

3) Como crearía un tablero de instrumentos de medición en cada una de estas plataformas

AWS

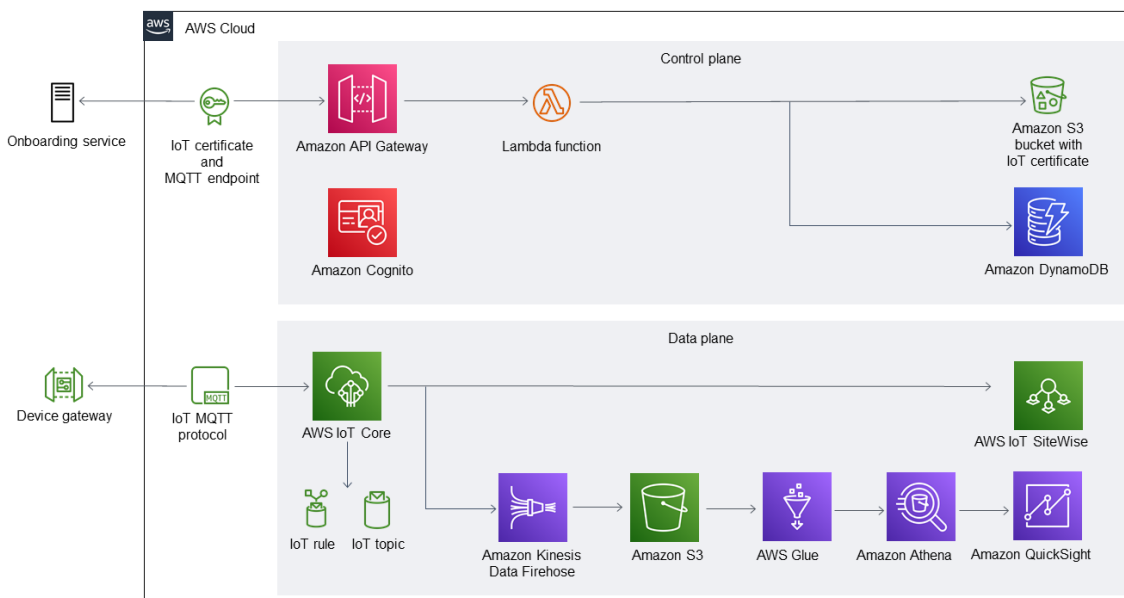
Para el plano de control:

- Un microservicio sin servidor para la incorporación de dispositivos de IoT mediante llamadas a la API REST.
- Certificados de seguridad de IoT puntos de enlace de MQ Telemetry Transport (MQTT) para el microservicio de incorporación para su uso al enviar tráfico de dispositivos.
- Amazon API Gateway para exponer tres puntos de enlace REST a fin de crear, recuperar y eliminar dispositivos incorporados.
- Amazon Cognito para proteger el servicio de incorporación. El Quick Start genera un token de actualización para que los usuarios pueden obtener un token de sesión de Amazon Cognito.
- Una función de Lambda con el objetivo de alojar un servicio para crear, recuperar y eliminar dispositivos incorporados.
- Un bucket de Amazon S3 para almacenar certificados de IoT.
- Amazon DynamoDB para almacenar metadatos de servicios incorporados.

Para el plano de datos:

- Una puerta de enlace que permite que la nube de AWS capture el tráfico de los dispositivos de IoT.
- Un protocolo MQTT de IoT para que se puedan enviar los datos al punto de enlace de MQTT desde el microservicio de incorporación.
- AWS IoT Core, que incluye un agente de MQTT que recibe tráfico de forma segura procedente de dispositivos autorizados, filtrados por el tema de IoT. AWS IoT Core usa los siguientes recursos:
 - Una regla de IoT para enviar el tráfico a AWS IoT Sitewise Monitor y al lago de datos de IoT. Puede añadir más reglas para poder usar el Quick Start en otros casos de uso.

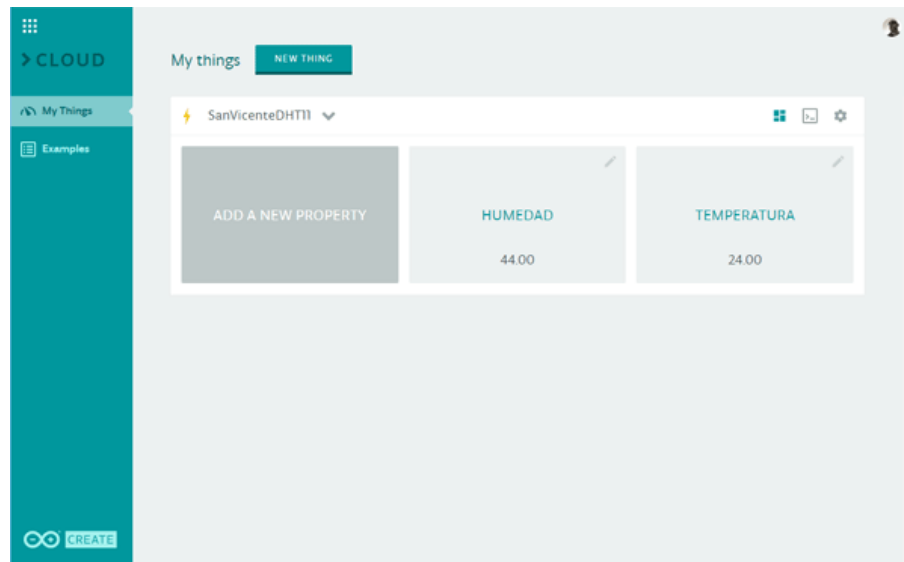
- Un tema de IoT para filtrar mensajes. El tema se configura dentro de la plantilla de AWS CloudFormation.
- AWS IoT SiteWise para monitorear los datos en tiempo real.
- Un lago de datos de IoT a fin de capturar datos inactivos para su monitoreo posterior. De forma predeterminada, el lago de datos se configura para mostrar datos de las últimas 24 horas.
 - Amazon Kinesis Data Firehose para transferir los datos a Amazon S3.
 - Amazon S3 para almacenar datos sin procesar capturados de sensores junto con datos depurados procesados por el script de extracción, transformación y carga (ETL).
 - AWS Glue para alojar el catálogo de datos, los rastreadores y los trabajos de ETL sin servidor para el lago de datos de IoT.
 - Amazon Athena con el objetivo de consultar los datos de sensores para verlos en paneles.
 - Amazon QuickSight para mostrar los datos de sensores en un panel preconfigurado. Puede configurar más paneles si quiere.



ARDUINO CLOUD

Quizás sea la más **sencilla para utilizar en proyectos IoT con Arduino**. No hace falta decir que es totalmente compatible con cualquier placa de Arduino que tenga conectividad.

Para **configurar** un dispositivo es muy sencillo, solo tienes que **seguir 4 simples pasos**. Dentro del repositorio oficial, encontramos la **librería *ArduinoCloud.h*** que nos permite mandar la información a la infraestructura que tiene Arduino en la nube.



El propio **configurador** te da hasta **el código que tienes que cargar en tu placa**, dependiendo de la que hayas elegido. Utiliza el protocolo MQTT para transmitir los datos. Por debajo trabaja con la **infraestructura en la nube de Amazon**.

Aunque es muy sencillo conectar un dispositivo, echo en falta la documentación necesaria para entender cómo funciona por debajo esta plataforma.

Las **limitaciones** son evidentes, **no** tiene una **API** para consultar los datos desde otras aplicaciones, **no** tiene **histórico** y **no** representa la **información en tiempo real**. Según Arduino, muy **pronto irán incorporando estas funcionalidades**.

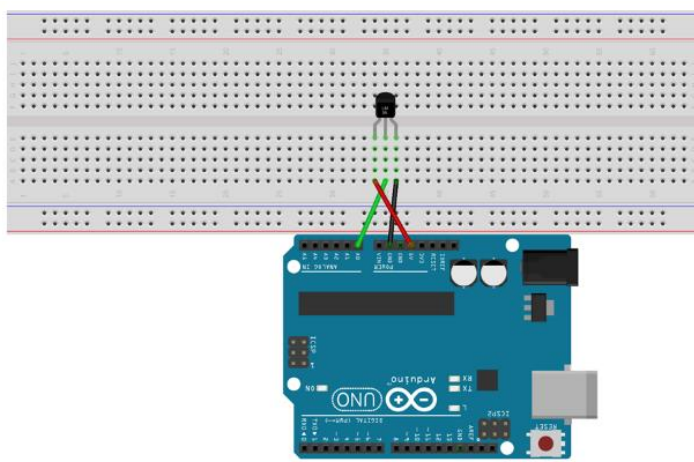
De momento se puede **usar de forma gratuita** esta plataforma. Las limitaciones las impondrán las tasas de lecturas de los sensores y los propios dispositivos.

4) como implementaría un sensor inteligente de temperatura

Cómo debemos **leer el sensor de temperatura LM35 en Arduino**. En cualquier proyecto que se precie es muy recomendable tener un sensor de temperatura para monitorear. En este caso hemos optado por un sensor bastante normal que detecta temperaturas desde -55°C a 150°C, 1°C equivale a 10mV y soporta voltajes de entre 4V y 30V. Todo esta información se ha obtenido de la **ficha técnica del LM35**. Cuando leemos un sensor analógico con Arduino lo hacemos a través de la función `analogRead` que nos da un valor entre 0 y 1023, 1024 valores posibles. Si tenemos 0V a la entrada nos devolverá 0 y si tenemos 5V nos devolverá 1023.

A partir de esta información podemos obtener una fórmula matemática que nos calcule la temperatura en función del voltaje que nos facilita el LM35.

$$\text{Temperatura} = \text{Valor} * 5 * 100 / 1024$$



Codificación

```
1  /*
2  Creado: Luis del Valle (ldelvalle@programafacil.com)
3  https://programafacil.com
4  */
5
6  // Declaración de variables globales
7  float tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)
8  int pinLM35 = 0; // Variable del pin de entrada del sensor (A0)
9
10 void setup() {
11   // Configuramos el puerto serial a 9600 bps
12   Serial.begin(9600);
13 }
14
15 void loop() {
16   // Con analogRead leemos el sensor, recuerda que es un valor de 0 a 1023
17   tempC = analogRead(pinLM35);
18
19   // Calculamos la temperatura con la fórmula
20   tempC = (5.0 * tempC * 100.0) / 1024.0;
21
22   // Envía el dato al puerto serial
23   Serial.print(tempC);
24   // Salto de línea
25   Serial.print("\n");
26
27   // Esperamos un tiempo para repetir el loop
28   delay(1000);
29 }
30 }
```

5) como implementaría un sensor inteligente de altitud

BMP280 SENSOR DE PRESIÓN DIGITAL

El BMP280 se basa en la probada tecnología de sensor de presión Piezoresistiva de Bosch, alta precisión y linealidad y estabilidad a largo plazo. Es un sensor de presión barométrica absoluto especialmente diseñado para aplicaciones móviles. Sus pequeñas dimensiones y su bajo consumo de energía de 2.7 μ A a 1Hz permiten la implementación en dispositivos con batería como teléfonos móviles, módulos de GPS o relojes. El BMP280 funciona con un ruido más bajo, admite nuevos modos de filtro y una interfaz SPI. El módulo proporciona una medición precisa de presión barométrica y temperatura en el medio ambiente; debido a su alta precisión en la medición de la presión, y los cambios de presión con la altitud, podemos calcular la altitud con precisión que hace que sea un altímetro exacto. Con este sensor también podemos medir la humedad relativa.

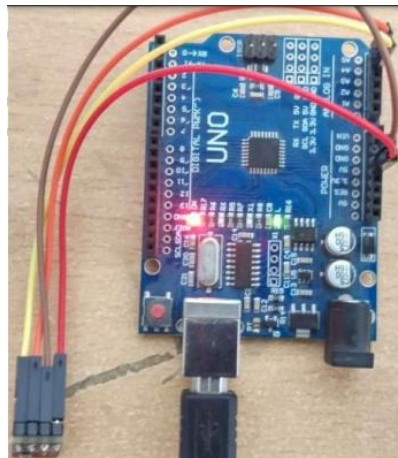
CABLEADO BMP280 Y ARDUINO UNO

Interfaz digital: I²C

SDA-A4

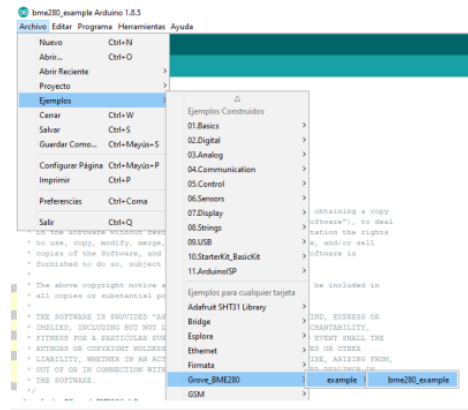
SCL-A5

VIN-5V



CÓDIGO DE PRUEBA

Para probar estos sensores se descargan las librerías Grove_BME280 y SFE_BMP180-master cambiar el nombre de la carpeta no comprimida a SFE_BMP180 y compruebe que las carpetas tengan los archivos .cpp y .h; coloque ambas carpetas en la carpeta de librerías de arduino. Reinicie el IDE y cargamos demo.

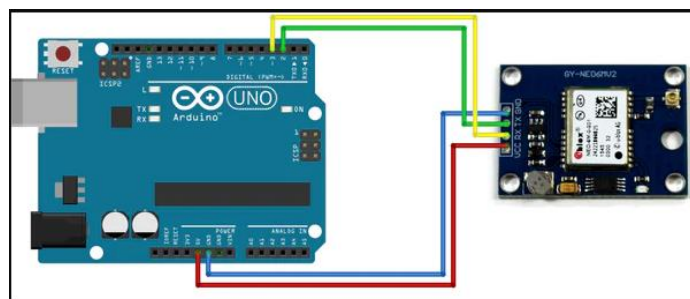


6) como implementaría un sensor inteligente de posicionamiento global

Material

- [Arduino Uno](#)
- [Cables dupond M-H](#)
- [Módulo GPS Neo GPS6MV2](#)

Diagrama de conexión



El modulo utiliza el protocolo UART por lo que para conectar el modulo con el micro solo hace falta simular un puerto serial con la librería **SoftwareSerial.h** que viene instalada en arduino por defecto.

Para ello usamos incluimos la librería y declaramos el puerto serial gps y cada que haya datos de entrada del gps lo mandamos al monitor serial para poder visualizarlo.

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial gps(RX, TX);
```

```
#include <SoftwareSerial.h>
```

```
const byte RX = 2;
```

```
const byte TX = 3;
```

```
SoftwareSerial gps(RX, TX);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  gps.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
  if (gps.available())
```

```
  {
```

```
    char data;
```

```
    data = gps.read();
```

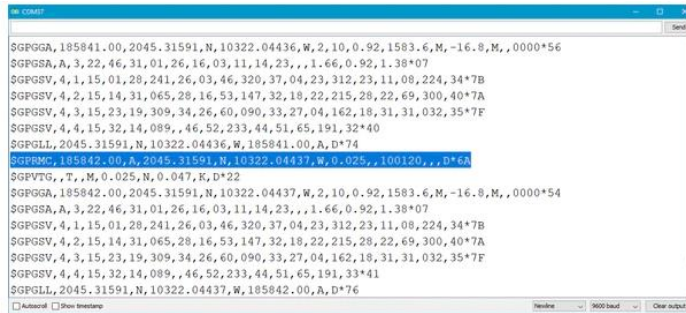
```
    Serial.print(data);
```

```
  }
```

Interpretacion de Datos

| NMEA 2.0 | | | | | |
|----------|--------|----------|----------|------|---|
| Name | Garmin | Magellan | Lowrance | SiRF | Notes: |
| GPAPB | N | Y | Y | N | Auto Pilot B |
| GPBOD | Y | N | N | N | bearing, origin to destination - earlier G-12's do not transmit this |
| GPGGA | Y | Y | Y | Y | fix data |
| GPGLL | Y | Y | Y | Y | Lat Lon data - earlier G-12's do not transmit this |
| GPGSA | Y | Y | Y | Y | overall satellite reception data, missing on some Garmin models |
| GPGSV | Y | Y | Y | Y | detailed satellite data, missing on some Garmin models |
| GPRMB | Y | Y | Y | N | minimum recommended data when following a route |
| GPRMC | Y | Y | Y | Y | minimum recommended data |
| GPRTT | Y | U | U | N | route data, only when there is an active route. (this is sometimes bidirectional) |
| GPWPL | Y | Y | U | N | waypoint data, only when there is an active route (this is sometimes bidirectional) |

Podemos ver que GPGLL nos entrega Latitud y longitud pero el mas utilizado es el GPRMC el cual nos entrega información bastante útil.



```
SGPGGA,185841.00,2045.31591,N,10322.04436,W,2,10,0.92,1583.6,M,-16.8,M,,0000*56
SGPGSA,A,3,22,46,31,01,26,16,03,11,14,23,,1.66,0.92,1.38*07
SGPGSV,4,1,15,01,28,241,26,03,46,320,37,04,23,312,23,11,08,224,34*7B
SGPGSV,4,2,15,14,31,065,28,16,53,147,32,18,22,215,28,22,69,300,40*7A
SGPGSV,4,3,15,23,19,309,34,26,60,090,33,27,04,162,18,31,31,032,35*7F
SGPGSV,4,4,15,32,14,089,,46,52,233,44,51,65,191,32*40
SGPGLL,2045.31591,N,10322.04436,W,185841.00,A,D*74
SGPRMC,185842.00,A,2045.31591,N,10322.04437,W,0.025,,100120,,,D*6A
SGPVTG,T,,M,0.025,M,0.047,K,D*22
SGPGGA,185842.00,2045.31591,N,10322.04437,W,2,10,0.92,1583.6,M,-16.8,M,,0000*54
SGPGSA,A,3,22,46,31,01,26,16,03,11,14,23,,1.66,0.92,1.38*07
SGPGSV,4,1,15,01,28,241,26,03,46,320,37,04,23,312,23,11,08,224,34*7B
SGPGSV,4,2,15,14,31,065,28,16,53,147,32,18,22,215,28,22,69,300,40*7A
SGPGSV,4,3,15,23,19,309,34,26,60,090,33,27,04,162,18,31,31,032,35*7F
SGPGSV,4,4,15,32,14,089,,46,52,233,44,51,65,191,33*41
SGPGLL,2045.31591,N,10322.04437,W,185842.00,A,D*76
```

\$GPRMC,185842.00,A,2045.31591,N,10322.04437,W,0.025,,100120,,,D*6A

Desglose del mensaje

- 185842.00 - Indica la Hora GMT 18:58:42
- A - significa que la información es correcto, de otra forma seria V
- **2045.31591** – significa la longitud 20°45.31591
- N - Norte
- **10322.04437** - significa la latitud 103°22.04347
- W - Oeste
- **0.025** – Velocidad en nudos
- 100120 -Fecha 10/enero/20