

## Pract\_1: Obtención de Datos en la Nube con Interfaces API

Alberto Zafra Navarro

Cabe destacar que todos los archivos comentados durante la memoria se encuentran en la carpeta **pract\_1** del repositorio **comm\_pract** del usuario de GitHub denominado **albertozafra7**. El enlace a la carpeta es:

[https://github.com/albertozafra7/comm\\_pract/tree/master/pract\\_1](https://github.com/albertozafra7/comm_pract/tree/master/pract_1).

### Tarea 1 - Capturar datos de API de Openweather

En esta tarea se utilizarán los servicios web de Openweather para obtener los datos del tiempo de cualquier lugar. Para ello ha sido preciso registrarse en la web de Openweather y obtener una api key que nos permita como usuarios obtener los datos deseados del tiempo en un formato JSON, XML o HTML, la API key obtenida y utilizada durante toda la práctica ha sido "69019a20ac0d9cf59895844430362f0b".

Obtenida dicha API key se ha procedido a probar distintos ejemplos de uso de la API de Openweather, comenzando con el ejemplo básico que se encuentra en la **documentación**, el cual es la visualización en tiempo real de las condiciones atmosféricas de un lugar en concreto. Esta prueba se ha realizado accediendo a la api mediante un enlace de la siguiente forma:

```
1 api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

Donde **city name** es el nombre de la ciudad/pueblo/localidad de la que se requiere obtener la información y la API key es la key obtenida anteriormente (69019a20ac0d9cf59895844430362f0b). Dicho esto el enlace utilizado para las pruebas es **el siguiente**:

```
1 api.openweathermap.org/data/2.5/weather?q=Albatera&appid=69019a20ac0d9cf59895844430362f0b
```

Tras introducir este enlace en el navegador se obtiene un resultado similar al de la Figura 1 como resultado. También es posible descargar dichos datos en un archivo de tipo JSON. El cual se mostraría de la siguiente manera:

```
1 {"coord":{"lon":-0.8706,"lat":38.179},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"base":"stations","main":{"temp":286.88,"feels_like":285.64,"temp_min":284.12,"temp_max":291,"pressure":1022,"humidity":51,"sea_level":1022,"grnd_level":1019},"visibility":10000,"wind":{"speed":1.07,"deg":321,"gust":1.42},"clouds":{"all":6},"dt":1645693151,"sys":{"type":2,"id":2001857,"country":"ES","sunrise":1645684965,"sunset":1645725062},"timezone":3600,"id":2522228,"name":"Albatera","cod":200}
```

Como se puede visualizar en la Figura 1 el uso de esta API nos proporciona información sobre las coordenadas de la localidad, cómo se encuentra el cielo (despejado, nublado,...), información sobre temperatura en grados Kelvin, la presión y humedad ambiente, su altura al nivel del mar, la velocidad del viento y su orientación, datos sobre las nubes y datos sobre la ciudad/pueblo del que se están obteniendo los datos atmosféricos.

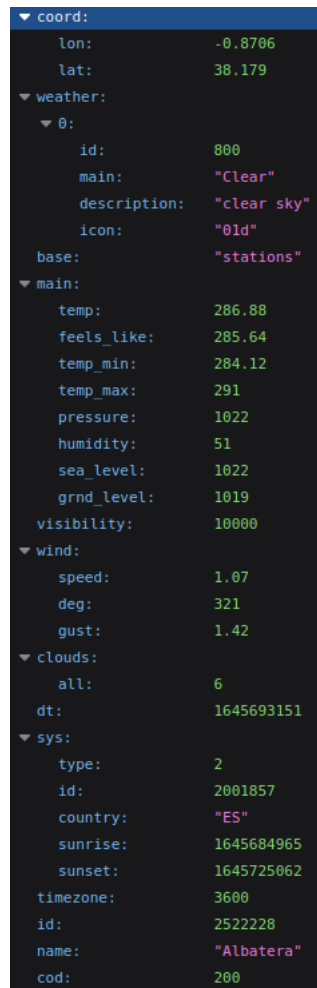


Figura 1: Datos del tiempo en Albatera.

## Tarea 2 - Capturar datos de API de Openweather con NodeRed

Para esta tarea fue necesario instalar Node-Red en el dispositivo utilizado para realizar las pruebas de configuración y conexión con Openweather. Para ello se siguió [el siguiente tutorial](#).

Una vez instalado completamente Node-Red fue necesario introducir los nodos de Openweather para obtener la información de la API sin necesidad de hacer una request http de la información. Sin embargo, al tener una versión desactualizada de nodeJS hubo muchos problemas al actualizar Node-Red para obtener la opción de "Manage palets" necesaria para instalar los nodos de Openweather y de UI para poder realizar una interfaz gráfica en la que se pueda interactuar con los datos obtenidos de dicha API.

Tras completar la instalación correctamente se prosiguió a realizar las pruebas de uso con Node-Red. Estas pruebas se basaron en la obtención de los datos de forma similar a los obtenidos en la **Tarea 1**. La prueba inicial se basó en la obtención de los datos del tiempo de la localidad de Albatera, para ello se insertó un nodo de tipo inyección, el nodo de Openweather map que obtiene los datos de la localidad que le indiques desde Node-Red junto a API-Key introducida, finalmente se debe incluir un nodo que obtenga la salida de dichos datos, el resultado se muestra en la Figura 2.

Cabe destacar que como Node-Red ha sido instalado localmente, por lo que para acceder al entorno de desarrollo de Node-Red es preciso lanzar desde el terminal el comando `node-red` y acceder desde el navegador a la dirección <http://localhost:1880/>.

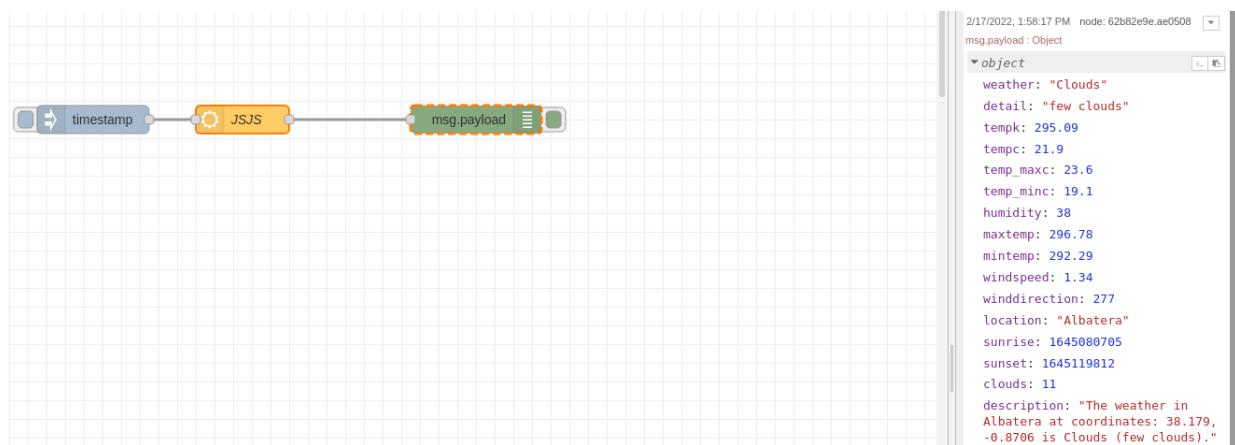


Figura 2: Estructura y resultado de la integración de OpenWeather con Node-Red.

Seguidamente, se modificó la estructura añadiendo un nodo de tipo función el cual obtenía directamente la temperatura en grados centígrados de la localidad de Albatera, para ello dentro del nodo función se introdujo el siguiente código:

```

1 msg.payload=msg.payload.tempc
2 return msg;
```

Mediante dicha implementación se obtuvo el diagrama y el resultado mostrado en la Figura 3.

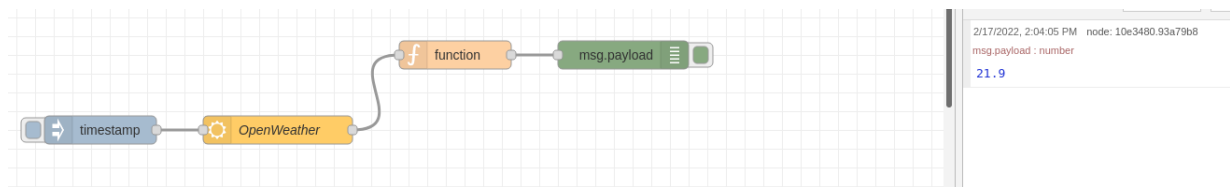


Figura 3: Estructura y resultado de la integración de OpenWeather con funciones en Node-Red.

Finalmente, como ampliación a esta tarea se realizaron diversas pruebas de integración de OpenWeather con Node-Red. Como tarea adicional inicial se realizó una integración de la librería UI con OpenWeather utilizando Node-Red como base, el ejercicio de testeo inicial se basó en ampliar la estructura previamente mostrada en la Figura 3 introduciendo un nodo de tipo gauge, para que fuese más sencillo visualizar la temperatura actual de Albatera, además se modificó el nodo de tipo inyección para que se realizase este proceso constantemente en un intervalo de 0.5s, de esta forma el elemento del dashboard se encontraría en constante actualización. La estructura resultante en Node-Red se muestra en la Figura 4, mientras que el resultado obtenido es el mostrado en la Figura 5.

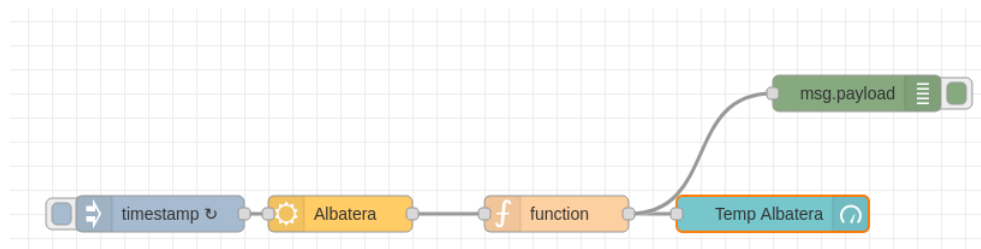


Figura 4: Estructura de la integración de OpenWeather junto a un Gauge.

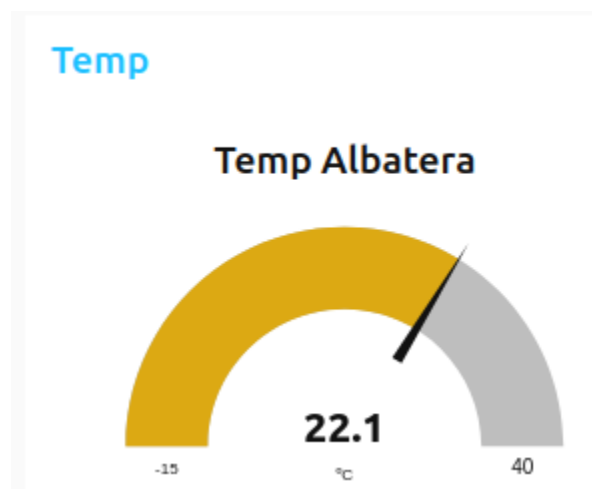


Figura 5: Estructura de la integración de OpenWeather junto a un Gauge.

Como se ha indicado anteriormente, Node-Red ha sido instalado localmente por lo que para acceder a dicho dashboard es preciso acceder desde el navegador a <http://localhost:1880/ui/>.

Por otro lado se expandió esto mediante la creación de una gráfica que muestra la humedad de la localidad por medio de una gráfica que se generaba con los datos de las predicciones de los 5 próximos días. Para ello se modificó el nodo de OpenWeather para recibir los datos de la predicción de los próximos 5 días. Como resultado se obtuvo un array de 40 posiciones el cual mostraba todos los posibles datos atmosféricos. No obstante, para poder realizar una gráfica partiendo del filtrado de los datos de humedad se prosiguió a realizar un bucle for que inyectara de forma periódica los datos individuales de humedad dentro del nodo de tipo chart. Como resultado se obtuvo un flujo similar al de la Figura 6, el cual mostraba los datos obtenidos de una forma similar a la mostrada en la Figura 7. Si se desea introducir el flujo de trabajo se encuentra en GitHub con el nombre de `Humidity_chart.json`.

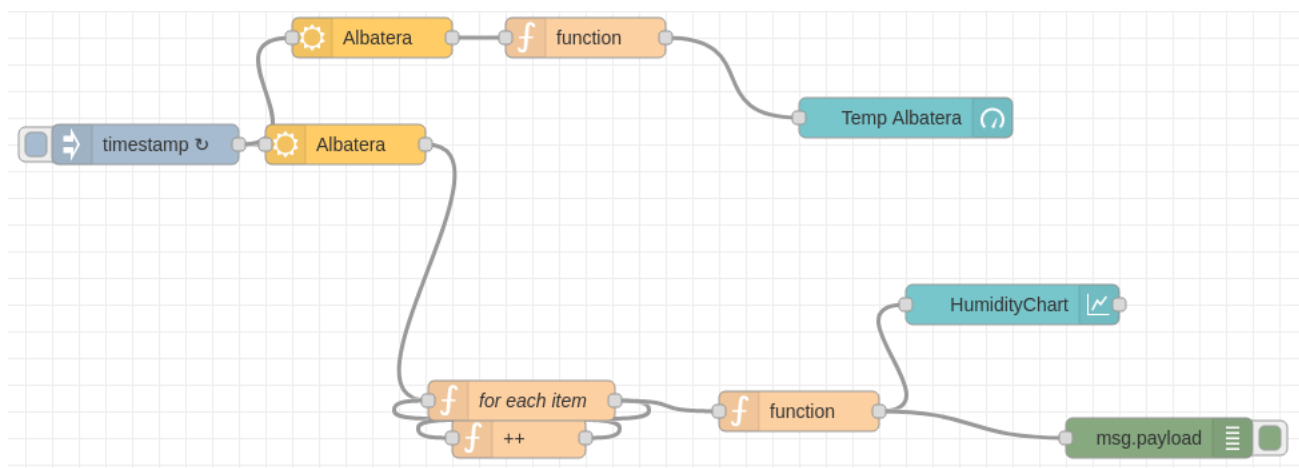


Figura 6: Estructura de la integración de lectura de la predicción de humedad de los próximos 5 días.

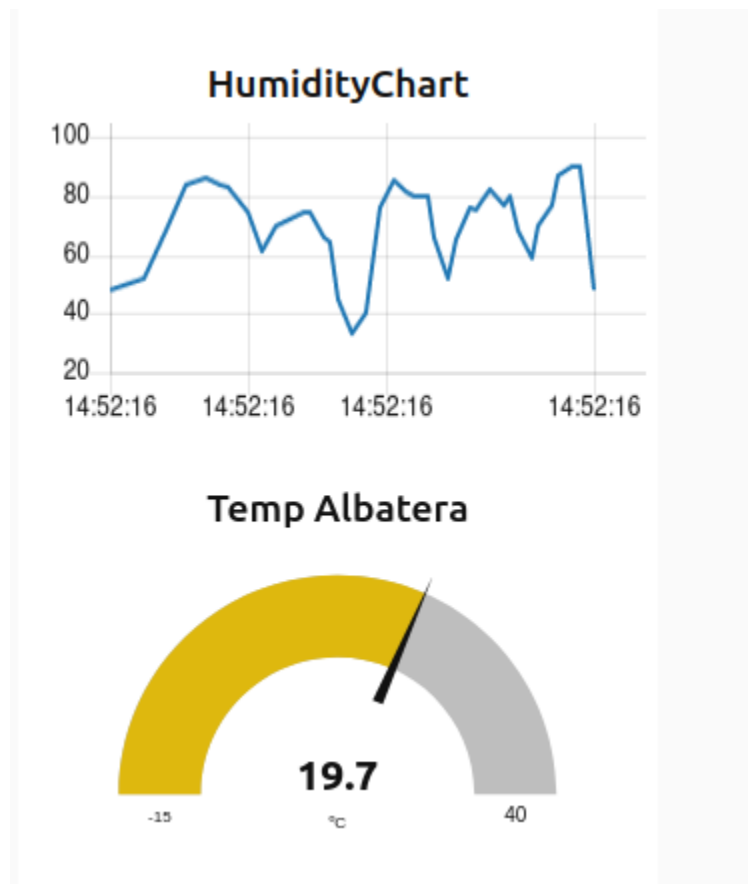


Figura 7: Resultado de la integración de lectura de la predicción de humedad de los próximos 5 días.

Por otro lado, resultó interesante generar la misma gráfica, pero en tiempo real, de forma que se pudiese visualizar la evolución de esta en vivo, cuya implementación se muestra en la Figura 8. No obstante, el resultado no fue el esperado, puesto que la predicción de temperatura en tiempo real es constante durante largo periodos de tiempo. Por lo que, en su mayoría, se visualizaría una recta totalmente horizontal a lo largo del tiempo. Sin embargo, tras estar diversas horas ejecutando el programa se pudo observar en la gráfica una variación de la humedad, esto es mostrado en la Figura 9.

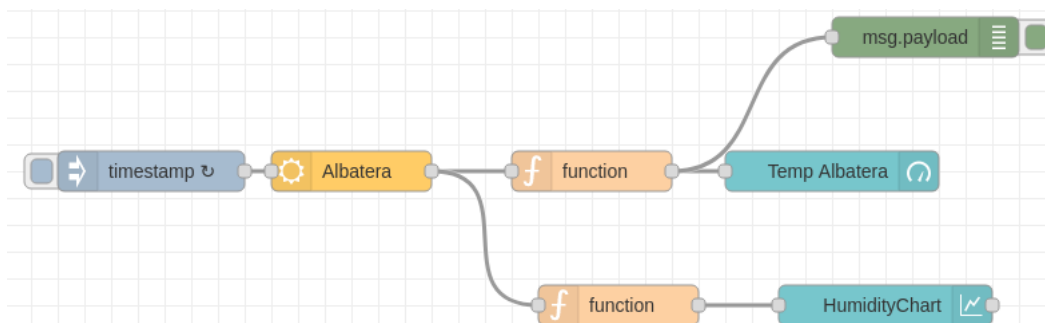


Figura 8: Estructura de la integración de lectura de la predicción de humedad en vivo.

Temp

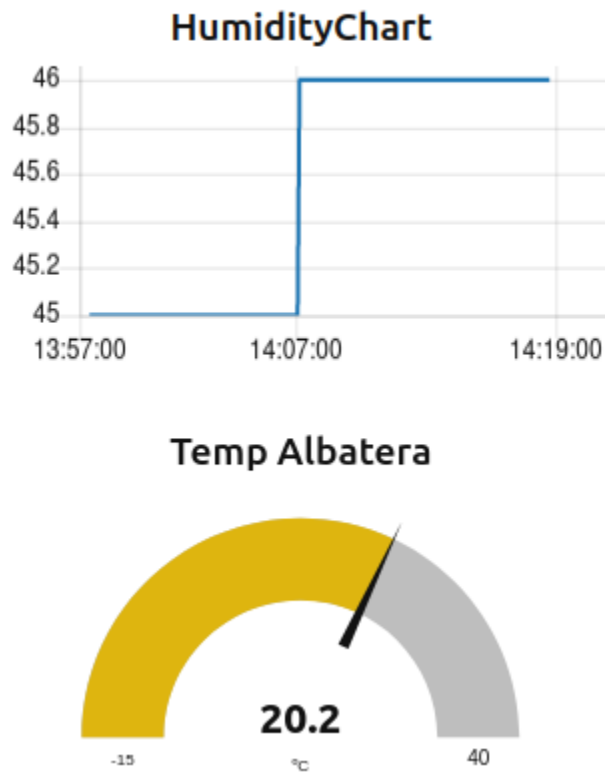


Figura 9: Resultado de la integración de lectura de la predicción de humedad en vivo.

### Tarea 3 - Capturar datos de API de la Red Eléctrica de España

En esta tarea se deseaba realizar el mismo proceso realizado anteriormente con OpenWeather. No obstante, en este caso no fue necesario obtener una API key, puesto que se puede acceder a la API con los ejemplos mostrados en la web de <https://www.ree.es/es/apidatos>. Los cuales se basan en peticiones http de tipo request para obtener los datos en un formato json.

En el primer ejemplo realizado se obtuvieron los datos en formato json sin ningún procesamiento posterior. Para ello se utilizó la estructura denominada EjemploRequest.json mostrada en la Figura 10, obteniendo el resultado mostrado en la Figura 11.

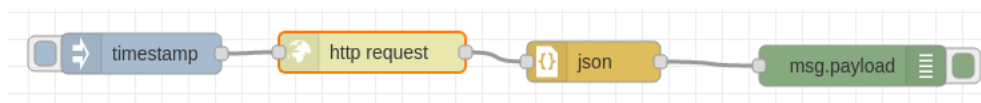


Figura 10: Estructura de la petición de datos de la Red Eléctrica de España.

```

2/24/2022, 1:39:14 PM node: 1556d2ef.b37e45
msg.payload: Object
▼ object
  ▼ data: object
    type: "Precios mercado peninsular en tiempo real"
    id: "mer13"
    ▼ attributes: object
      title: "Precios mercado peninsular en tiempo real"
      last-update: "2022-02-23T20:17:46.000+01:00"
      description: null
    ▶ meta: object
  ▼ included: array[2]
    ▼ 0: object
      type: "PVPC (€/MWh)"
      id: "1001"
      groupId: null
      ▼ attributes: object
        title: "PVPC (€/MWh)"
        description: null
        color: "#ffcf09"
        type: null
        magnitude: "price"
        composite: false
        last-update: "2022-02-23T20:17:46.000+01:00"
        ▶ values: array[24]
      ▼ 1: object
        type: "Precio mercado spot (€/MWh)"
        id: "600"
        groupId: null
        ▶ attributes: object

```

Figura 11: Resultado de la petición de datos de la Red Eléctrica de España.

Seguidamente se prosiguió a la obtención de los datos del precio de la luz en una franja concreta de tiempo en el pasado, para ello se utilizó una de las estructuras exportadas en el archivo `electric_flow`, puesto que pese a que ambas son similares a la mostrada en la Figura 12, el contenido de cada uno de los nodos varía, como por ejemplo en el nodo de la función el contenido es el siguiente para la obtención de los datos dentro de una franja de tiempo:

```

1 msg.payload = msg.payload.included[0].attributes.values[13]
2 return msg;

```

Y el siguiente para la obtención del porcentaje de energía renovable generada en esa misma franja de tiempo:

```

1 msg.payload = msg.payload.included[0].attributes.values[13].percentage
2 return msg;

```

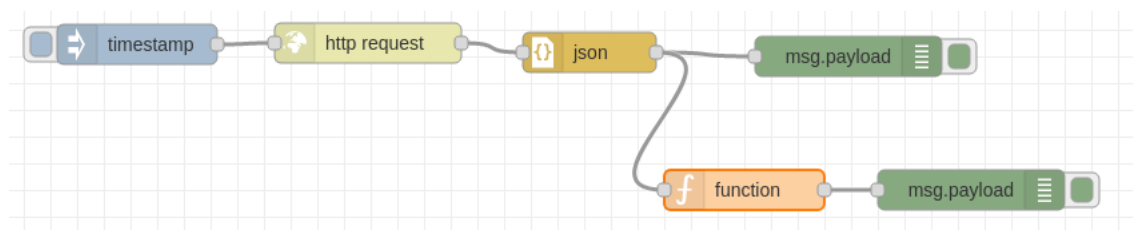


Figura 12: Estructura de la obtención de datos de la Red Eléctrica de España a las 1PM.

El resultado obtenido de dichos datos fue el mostrado en la Figura 13.

```
▼ 13: object
  value: 354.61
  percentage:
    0.6301601123096334
  datetime:
    "2022-02-24T13:00:00.
    000+01:00"
```

Figura 13: Resultado de la obtención de datos de la Red Eléctrica de España a las 1PM.

Finalmente, como ampliación a esta tarea se realizó la obtención de los datos del porcentaje de energía renovable en ese determinado instante, dando como resultado uno similar al de la Figura 14.

```
2/24/2022, 1:54:32 PM node: e9daeb791f0020d0
msg.payload : number
0.4554276783457168
```

Figura 14: Resultado de la obtención de datos de la Red Eléctrica de España sobre el porcentaje de energía renovable.

#### Tarea 4 - Tarea adicional: Uso de la Instagram APIs.

Como tarea adicional se ha planteado la investigación y pruebas de integración de diversas APIs de Instagram junto a Node-Red.

Inicialmente se instalaron e intentaron utilizar los nodos de Instagram contenidos en la paleta de Node-Red, dichos nodos se muestran en la Figura 15. Sin embargo, tras investigar al respecto sobre esta conexión directa de Node-Red se descubrió que en 2020 se deshabilitó el uso de APIs de terceros que interactuasen con la red social de Instagram, tal y como se indica en el [siguiente enlace](#). Puesto que el uso de la API anteriormente mencionada quedó completamente descartada se decidió indagar sobre la [Instagram Basic Display API](#) y la [Instagram Graph API](#). La diferencia entre ambas se basa en que mientras que la Instagram Basic Display API permite interactuar de forma directa con los perfiles de un usuario, mientras que el Instagram Graph API se basa en obtener las analíticas de una cuenta profesional de Instagram.



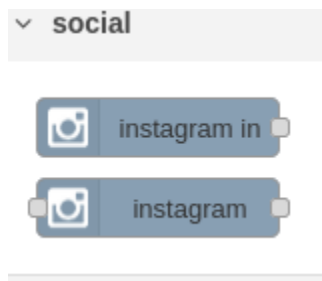


Figura 15: Nodos de Instagram instalados desde la paleta de Node-Red.

Tras realizar distintas pruebas e investigar en profundidad no se consiguió ningún resultado que verificase la correcta conexión entre estas plataformas, puesto que la mayoría de ejemplos se encuentra relacionados con la interacción de Instagram con otro tipo de apps móviles. Por lo que se decidió dejar como incompleto este apartado por el momento.