



FUENTES DE DATOS - HACKATIÑO 2017

En este documento se describen las diferentes **fuentes de datos** que el Ayuntamiento de A Coruña pone a disposición de la organización del **Hackatiño 2017**

BICICORUÑA

CSV anonimizado con datos históricos sobre el uso de BiciCoruña

<https://app.box.com/s/ujyrxo7eq4px4uk0x3wgte4yt99kzrii>

API - JSON - Listado de estaciones de BiciCoruña con sus ubicaciones (tiempo real).

<https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets>

API - JSON - Estado de las diferentes estaciones de BiciCoruña (tiempo real)

<https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getMeasures>

ESTADO PARKINGS SUBTERRÁNEOS EN TIEMPO REAL

API - JSON - Listado de Parkings subterráneos de Coruña con sus ubicaciones (tiempo real).

<https://smart.coruna.es/sib-api/api/v1/openDataParkingSubterraneo/getAssets>

API - JSON - Obtener toda la Información de un parking subterráneo consultando por ID (tiempo real)

[https://smart.coruna.es/sibapi/api/v1/openDataParkingSubterraneo/getAssetInfo?
\\$id=CONCHINAS](https://smart.coruna.es/sibapi/api/v1/openDataParkingSubterraneo/getAssetInfo?&id=CONCHINAS)

API - JSON - Obtener los estados de ocupación de todos los parkings subterráneos de Coruña (tiempo real)

<https://smart.coruna.es/sib-api/api/v1/openDataParkingSubterraneo/getMeasures>

ESTADO PLAZAS DE PARKING DE SUPERFICIE (CARGA/DESCARGA + MOVILIDAD REDUCIDA)



API - JSON - Listado de plazas de parking de superficie sensorizadas con sus ubicaciones (tiempo real).

<https://smart.coruna.es/sib-api/api/v1/openDataParkingSuperficie/getAssets>

API - JSON - Obtener toda la información de una plaza parking de superficie sensorizada consultando por su ID (tiempo real).

[https://smart.coruna.es/sib-api/api/v1/openDataParkingSuperficie/getAssetInfo?\\$id=27E5](https://smart.coruna.es/sib-api/api/v1/openDataParkingSuperficie/getAssetInfo?$id=27E5)

API - JSON - Obtener los estados de ocupación de las plazas de parking de superficie sensorizadas (tiempo real)

<https://smart.coruna.es/sib-api/api/v1/openDataParkingSuperficie/getMeasures>

ESTADO PARADAS DE AUTOBUS EN TIEMPO REAL

API - JSON - Listado de paradas de autobuses urbano con sus ubicaciones (tiempo real).

<https://smart.coruna.es/sib-api/api/v1/openDataParadaAutobus/getAssets>

API - JSON - Obtener toda la info de una parada de autobus urbano (tiempo real).

[https://smart.coruna.es/sib-api/api/v1/openDataParadaAutobus/getAssetInfo?\\$id=1](https://smart.coruna.es/sib-api/api/v1/openDataParadaAutobus/getAssetInfo?$id=1)

UBICACIÓN DE AUTOBUSES EN TIEMPO REAL

API - JSON - Listado de autobuses urbanos con sus ubicaciones (tiempo real).

<https://smart.coruna.es/sib-api/api/v1/openDataAutobus/getAssets>



MANUAL DE USO DE LAS APIS PARA OBTENER LOS DATOS EN TIEMPO REAL

Las urls de las APIs facilitadas devuelven sus resultados en formato JSON.

Vamos a ver un ejemplo:

Obtener el listado de estaciones de bicioruña con sus ubicaciones en tiempo real

<https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets>

Hay que usar una petición GET a la URL anterior, enviando en el header de la petición un token de acceso que se denomina **x-sofia2-apikey** con el siguiente valor (igual para todas las URLs y fuentes de datos): **14b0fdf8d58541a087c49a97f61903eb**

A continuación se muestran diferentes ejemplos de acceso, en función del lenguaje de programación empleado por el desarrollador. Es fácilmente entendible para que pueda adaptarse a otros lenguajes.

Línea de comandos

```
curl --request GET \  
--url https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets \  
--header 'content-type: application/json;charset=utf-8' \  
--header 'x-sofia2-apikey: 14b0fdf8d58541a087c49a97f61903eb'
```

Petición HTTP - GET

```
GET /sib-api/api/v1/openDataBiciCoruna/getAssets HTTP/1.1  
Host: smart.coruna.es  
X-SOFIA2-APIKey: 14b0fdf8d58541a087c49a97f61903eb  
Content-Type: application/json;charset=utf-8
```

Ruby

```
require 'uri'  
require 'net/http'  
  
url = URI("https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets")  
  
http = Net::HTTP.new(url.host, url.port)  
http.use_ssl = true  
http.verify_mode = OpenSSL::SSL::VERIFY_NONE  
  
request = Net::HTTP::Get.new(url)
```



```
request["x-sofia2-apikey"] = '14b0fdf8d58541a087c49a97f61903eb'  
request["content-type"] = 'application/json;charset=utf-8'
```

```
response = http.request(request)  
puts response.read_body
```

Python

```
import requests  
  
url = "https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets"  
  
headers = {  
    'x-sofia2-apikey': "14b0fdf8d58541a087c49a97f61903eb",  
    'content-type': "application/json;charset=utf-8",  
}  
  
response = requests.request("GET", url, headers=headers)  
  
print(response.text)
```

Node.js

```
var http = require("https");  
  
var options = {  
    "method": "GET",  
    "hostname": "smart.coruna.es",  
    "port": null,  
    "path": "/sib-api/api/v1/openDataBiciCoruna/getAssets",  
    "headers": {  
        "x-sofia2-apikey": "14b0fdf8d58541a087c49a97f61903eb",  
        "content-type": "application/json;charset=utf-8",  
    }  
};  
  
var req = http.request(options, function (res) {  
    var chunks = [];  
  
    res.on("data", function (chunk) {  
        chunks.push(chunk);  
    });  
});
```



```
res.on("end", function () {  
  var body = Buffer.concat(chunks);  
  console.log(body.toString());  
});  
});  
  
req.end();
```



EJEMPLOS DE INVOCACIÓN Y RESULTADOS ESPERADOS:

Listado de estaciones de bicioruña con sus ubicaciones

<https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getAssets>

```
{
  "data": "[{"Feed": {"assetId": "6", "assetName": "Plz Pontevedra", "geometry": {"coordinates": [-8.406864166259766, 43.367919921875]}}, {"Feed": {"assetId": "13", "assetName": "Praza de Vigo", "geometry": {"coordinates": [-8.406954765319824, 43.362613677978516]}}, {"Feed": {"assetId": "24", "assetName": "Puerta Real", "geometry": {"coordinates": [-8.395782470703125, 43.37001419067383]}}, {"Feed": {"assetId": "11", "assetName": "Castros", "geometry": {"coordinates": [-8.39080810546875, 43.35383224487305]}}, {"Feed": {"assetId": "15", "assetName": "Matogrande", "geometry": {"coordinates": [-8.40505599975586, 43.34028625488281]}}, {"Feed": {"assetId": "12", "assetName": "Puerto", "geometry": {"coordinates": [-8.3984375, 43.36930465698242]}}, {"Feed": {"assetId": "19", "assetName": "So", "geometry": {"coordinates": [-8.404376983642578, 43.37088394165039]}}, {"Feed": {"assetId": "9", "assetName": "Casa del Agua", "geometry": {"coordinates": [-8.418445587158203, 43.36918258666992]}}, {"Feed": {"assetId": "1", "assetName": "Estación de Tren", "geometry": {"coordinates": [-8.4093017578125, 43.35293197631836]}}, {"Feed": {"assetId": "3", "assetName": "Torre de Hércules", "geometry": {"coordinates": [-8.402005195617676, 43.38335037231445]}}, {"Feed": {"assetId": "22", "assetName": "Paseo Puentes", "geometry": {"coordinates": [-8.415596961975098, 43.36427307128906]}}, {"Feed": {"assetId": "23", "assetName": "Monelos", "geometry": {"coordinates": [-8.404940605163574, 43.34931945800781]}}, {"Feed": {"assetId": "16", "assetName": "Los Rosales", "geometry": {"coordinates": [-8.43204402923584, 43.37434387207031]}}, {"Feed": {"assetId": "8", "assetName": "Millenium", "geometry": {"coordinates": [-8.423005104064941, 43.3769645690918]}}, {"Feed": {"assetId": "21", "assetName": "Sagrada Familia", "geometry": {"coordinates": [-8.415870666503906, 43.357887268066406]}}, {"Feed": {"assetId": "20", "assetName": "Plz. de España", "geometry": {"coordinates": [-8.396331787109375, 43.372802734375]}}, {"Feed": {"assetId": "14", "assetName": "Conchiñas", "geometry": {"coordinates": [-8.419282913208008, 43.361183166503906]}}, {"Feed": {"assetId": "7", "assetName": "Obelisco", "geometry": {"coordinates": [-8.402587890625, 43.36846923828125]}}, {"Feed": {"assetId": "18", "assetName": "Tabacos", "geometry": {"coordinates": [-8.401966094970703, 43.35763931274414]}}, {"Feed": {"assetId": "5", "assetName": "Aquarium", "geometry": {"coordinates": [-8.41003131866455, 43.38148880004883]}}, {"Feed": {"assetId": "10", "assetName": "Estación Autobuses", "geometry": {"coordinates": [-8.406013488769531, 43.353599548339844]}}, {"Feed": {"assetId": "17", "assetName": "Campo de Marte", "geometry": {"coordinates": [-8.398441314697266, 43.37588882446289]}}}],
  "error": null,
  "errorCode": null,
  "ok": true
}
```

Estado de las diferentes estaciones de BiciCoruña

<https://smart.coruna.es/sib-api/api/v1/openDataBiciCoruna/getMeasures>

```
{
  "data": [{"Feed": {"assetId": "9", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "16", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "5", "method": "REAL"}]}, {"Feed": {"assetId": "11", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "12", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "0", "method": "REAL"}]}, {"Feed": {"assetId": "15", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "12", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "6", "method": "REAL"}]}, {"Feed": {"assetId": "8", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "16", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "4", "method": "REAL"}]}, {"Feed": {"assetId": "6", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "24", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "13", "method": "REAL"}]}, {"Feed": {"assetId": "13", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "16", "method": "REAL"}, {"name": "BicisDisponibles", "desc": "Bicicletas disponibles", "unit": "u", "measure": "9", "method": "REAL"}]}, {"Feed": {"assetId": "19", "measures": [{"name": "PuestosActivos", "desc": "Total de Puestos", "unit": "u", "measure": "16", "method": "REAL"}]}]}]
```

