



**Manual de consultas  
API REST – Estaciones meteorológicas en  
carreteras**



**Fecha: 2 de octubre de 2023**

# Tabla de contenidos

Tabla de contenidos .....	2
1    Introducción.....	3
2    Estaciones meteorológicas .....	3
2.1    Parámetros.....	4
2.2    Posibles errores en la llamada.....	4
2.3    Llamada de ejemplo .....	4
2.3.1    Descripción de la salida .....	4
2.3.2    Ejemplo.....	5

## 1 Introducción

A través del proyecto de Territorio Rural Inteligente de Castilla y León se recogen en tiempo real datos sobre las condiciones meteorológicas que se están produciendo en determinados puntos sensorizados de la red de carreteras de titularidad autonómica. Estas estaciones recogen valores de temperatura, humedad relativa, punto de rocío, presión atmosférica, velocidad y dirección del viento, radiación solar, radiación UV y pluviometría.

Existen dos tipos de ubicaciones de las estaciones meteorológicas para la medición de las condiciones ambientales. Unas se encuentran situadas en los silos de sal ubicados por la red de carreteras autonómica o en los centros de conservación de carreteras. Y otras se sitúan a pie de carretera en puntos especialmente sensibles para la detección de heladas. Hay estaciones meteorológicas repartidas por las nueve provincias de Castilla y León.

Para obtener información de las estaciones meteorológicas se ha creado una CDA para que cualquier desarrollador pueda acceder a la información recogida en tiempo real por los sensores. Para ello se han diseñado dos servicios:

- **meteoSilos** (SaltSiloRaw): Servicio que devuelve información sobre las condiciones ambientales que se están midiendo en las estaciones meteorológicas situadas en los silos de sal y en centros de conservación de carreteras.
- **meteoCarreteras** (RoadfrostsensorRaw): Servicio que devuelve información sobre las condiciones ambientales que se están midiendo en las estaciones meteorológicas situadas a pie de carretera.

El objeto del presente documento es detallar la especificación de dichos servicios.

## 2 Estaciones meteorológicas

Para obtener las condiciones meteorológicas se ha definido la siguiente query:

- URL:
  - Estaciones meteorológicas de tipo **SaltSiloRaw**  
[https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc\\_jcyl/verticals/sql/carreteras.cda&dataAccessId=SaltSiloRaw&paramstart=&paramfinish=&paramentityid=&paramprovince=](https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc_jcyl/verticals/sql/carreteras.cda&dataAccessId=SaltSiloRaw&paramstart=&paramfinish=&paramentityid=&paramprovince=)
  - Estaciones meteorológicas de tipo **RoadfrostsensorRaw**  
[https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc\\_jcyl/verticals/sql/carreteras.cda&dataAccessId=RoadfrostsensorRaw&paramstart=&paramfinish=&paramentityid=&paramprovince=](https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc_jcyl/verticals/sql/carreteras.cda&dataAccessId=RoadfrostsensorRaw&paramstart=&paramfinish=&paramentityid=&paramprovince=)
- Las credenciales de acceso para la consulta son:
  - Usuario: sc\_jcyl\_publico
  - Contraseña: sc\_jcyl\_publico
- Los formatos de salida que puede devolver el servicio son JSON, XML y CSV. Por defecto el formato será XML. Si se desea descargar un fichero csv será necesario indicar al final de la url: &output=csv.

## 2.1 Parámetros

Los parámetros que se pasan a la función son los siguientes:

1. **dataAccessId**: tipo (String), indica el tipo de entidad que estamos consultando SaltsiloRaw o RoadfrostsensorRaw.
2. **paramentityid**: tipo (String), indica el ID de la estación meteorológica en concreto que se desea consultar.
3. **paramprovince**: tipo (String), indica la provincia de la cuál deseamos consultar los datos de las estaciones meteorológicas, escrito en minúsculas y sin tilde (avila, burgos, leon, palencia, salamanca, segovia, soria, valladolid, zamora).
4. **paramstart**: fecha en texto (formato **YYYY-MM-DD%20hh%3Amm%3Ass**, siendo YYYY el año, MM el mes, DD el día, hh la hora, mm los minutos y ss los segundos y utilizando el sistema horario de 24 horas). Se seleccionarán las medidas recogidas por la estación meteorológica cuya fecha de inicio sea posterior a esa fecha y hora.
5. **paramfinish** fecha en texto (formato **YYYY-MM-DD%20hh%3Amm%3Ass**, siendo YYYY el año, MM el mes, DD el día, hh la hora, mm los minutos y ss los segundos y utilizando el sistema horario de 24 horas). Se seleccionarán las medidas recogidas por la estación meteorológica hasta la fecha de fin.

En esta llamada no es obligatorio ningún parámetro, ya que en caso de que no se indique ninguno, la consulta devolverá todos los valores almacenados en la base de datos para todas las estaciones meteorológicas situadas en silos de sal y centros de conservación de carreteras definidos en el Anexo I desde su instalación hasta el momento de la consulta.

Se recomienda realizar consultas acotadas en tiempo y sobre estaciones concretas o por provincia para no ralentizar la obtención de datos de la consulta.

## 2.2 Posibles errores en la llamada

1. **Si se incluye un id de la entidad** que no está definido en el Anexo I devolverá valores nulos.
2. **Si se incluye una provincia** que no está definida en el Anexo I devolverá valores nulos. Se recuerda que paramprovince debe estar escrito en minúsculas y sin tilde (avila, burgos, leon, palencia, salamanca, segovia, soria, valladolid, zamora)

## 2.3 Llamada de ejemplo

[https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc\\_jcyl/verticals/sql/carreteras.cda&dataAccessId=SaltsiloRaw&paramstart=2023-03-01%2000%3A00%3A00&paramfinish=2023-03-31%2023%3A59%3A59&paramentityid=ZA\\_CIOVI\\_CAR\\_0001](https://bi.territoriointeligente.jcyl.es/pentaho/plugin/cda/api/doQuery?path=/public/sc_jcyl/verticals/sql/carreteras.cda&dataAccessId=SaltsiloRaw&paramstart=2023-03-01%2000%3A00%3A00&paramfinish=2023-03-31%2023%3A59%3A59&paramentityid=ZA_CIOVI_CAR_0001)

### 2.3.1 Descripción de la salida

La salida devuelve un fichero en xml o json con todas las medidas recogidas por la estación meteorológica de tipo SaltSilo de **ZA\_CIOVI\_CAR\_0001** situada en el término municipal de Otero de Bodas en la provincia de Zamora en el **mes de marzo de 2023**.

Se especifica en la consulta:

- dataAccessId=SaltsiloRaw
- paramentityid=ZA\_CIOVI\_CAR\_0001
- paramstart=2023-03-01%2000%3A00%3A00
- paramfinish=2023-03-31%2023%3A59%3A59

Al ejecutar la consulta se devuelve la lista de parámetros recogidos por la estación indicada en cada uno de los periodos de tiempo indicados. A continuación, se indican los valores devueltos por la consulta y en el orden en que aparecen:

0. **entityid**: Devuelve un String indicando la estación de la que se consultan los datos.
1. **entitytype**: Devuelve un String indicando el tipo de estación de la que se consultan los datos, en este caso SaltSilo.
2. **timeinstant**: Devuelve un String indicando el instante de tiempo (fecha y hora) en la que se han medido los datos.
3. **temperature**: Devuelve un Numeric con la temperatura media en grados centígrados (°C) que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos
4. **relativehumidity**: Devuelve un Numeric con la humedad relativa media en tanto por 100 (%) que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos.
5. **windspeed**: Devuelve un Integer con la velocidad del viento media en Km/h que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos
6. **winddirection**: Devuelve un Numeric con la dirección media del viento en grados que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos
7. **dewpoint**: Devuelve un Numeric con el punto de rocío medio que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos.
8. **windgust**: Devuelve un Numeric con la racha de viento en Km/h, siendo esta la velocidad máxima del intervalo de medida de 10 minutos.
9. **solarradiation**: Devuelve un Numeric con el nivel medio de radiación solar medido en W/m<sup>2</sup> que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos.
10. **uvradiation**: Devuelve un Numeric con el nivel medio de radiación UV medido en W/m<sup>2</sup> que ha medido la estación meteorológica durante el intervalo de medida de 10 minutos.
11. **rainlevel**: Devuelve un Numeric con el nivel de precipitación acumulado medido en mm durante el intervalo de medida de 10 minutos.
12. **municipality**: Devuelve un String con el municipio en el que se ubica la estación meteorológica.
13. **province**: Devuelve un String con la provincia en la que se ubica la estación meteorológica.

## 2.3.2 Ejemplo

```
{
  "metadata": [
    { "colName": "entityid", "colType": "String", "colIndex": 0 },
    { "colName": "entitytype", "colType": "String", "colIndex": 1 },
    { "colName": "timeinstant", "colType": "String", "colIndex": 2 },
    { "colName": "temperature", "colType": "Numeric", "colIndex": 3 },
    { "colName": "relativehumidity", "colType": "Numeric", "colIndex": 4 },
    { "colName": "windspeed", "colType": "Integer", "colIndex": 5 },
    { "colName": "winddirection", "colType": "Numeric", "colIndex": 6 },
    { "colName": "dewpoint", "colType": "Numeric", "colIndex": 7 },
    { "colName": "windgust", "colType": "Numeric", "colIndex": 8 },
    { "colName": "solarradiation", "colType": "Numeric", "colIndex": 9 },
    { "colName": "uvradiation", "colType": "Numeric", "colIndex": 10 },
    { "colName": "rainlevel", "colType": "Numeric", "colIndex": 11 },
    { "colName": "municipality", "colType": "String", "colIndex": 12 },
    { "colName": "province", "colType": "String", "colIndex": 13 }
  ],
  "resultset": [
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:06:26.613Z", 0, 65, 1, 60, -6, 1, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:06:26.798Z", 0, 65, 1, 60, -6, 1, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:06:26.987Z", 0, 67, 4, 47, -6, 11, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:16:11.089Z", 0, 67, 4, 47, -6, 11, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:16:11.265Z", 0, 67, 4, 47, -6, 11, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:16:11.441Z", 0, 66, 2, 38, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:26:10.058Z", 0, 66, 2, 38, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:26:10.249Z", 0, 66, 2, 38, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:26:10.426Z", -1, 66, 2, 39, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:36:24.855Z", -1, 66, 2, 39, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:36:25.057Z", -1, 66, 2, 39, -6, 3, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:36:25.230Z", -1, 66, 0, 41, -6, 0, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:45:49.910Z", -1, 66, 0, 41, -6, 0, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:45:50.106Z", -1, 66, 0, 41, -6, 0, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:45:50.289Z", -1, 68, 1, 33, -6, 1, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:56:09.356Z", -1, 68, 1, 33, -6, 1, 0, 0, 0, "otero de bodas", "zamora"
    ],
    [
      "ZA_CIOVI_CAR_0001", "SaltSilo", "2023-03-01T00:56:09.536Z", -1, 68, 1, 33, -6, 1, 0, 0, 0, "otero de bodas", "zamora"
    ]
  ]
}
```