

+Convertirlo en un MCP

Scrapeando el Catastro para resolver disputas por los olivos

Sólo hay dos cosas ciertas en la vida: La muerte y **los impuestos**

- Benjamin Franklin

Programar **puede ser un superpoder**

- Juan Luis Cano

Todo poder conlleva una **gran responsabilidad**

- El tío Ben / La tía May (según la Peli de Spiderman que te hayas visto)





\$ whoami

De **(North)** Granada.

Graduado en Ing. Informática.

Desarrollador Backend en **una**
consultora.

Aparte de programar: Toco el clarinete,
cuido olivos, y hasta hago teatro.

En general, **hago cosas.**



¿Me estás buscando en redes?

@X @IvanInVR(@mastodon.la)

@IvanitiX

/ivan-vr

¡Atención!

La siguiente charla contiene refes a Spiderman y Dr. Strange, memes , olivos fuera de Jaén, mapas, paquetes de Python e IAs de código abierto y no tan abierto.
Viewer discretion is advised.



Iván vs El Catastro

01

Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

Replantando el proyecto como paquete

- Mi viaje a Vigo
- ¿Podemos sacar más información?
- La API del Catastro y otros menesteres

03

Conteniendo la API y aplicándolo hasta en la sopa

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.
- Una API para simplificar otra API con FastAPI
- MCPs con FastMCP

Iván vs El Catastro

01

Scrapeando la página

- ¿A santo de qué?
- La página del Catastro
- Mi primer script

02

Replantando el proyecto como paquete

- Mi viaje a Vigo
- ¿Podemos sacar más información?
- La API del Catastro y otros menesteres

03

Conteniendo la API y aplicándolo hasta en la sopa

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.
- Una API para simplificar otra API con FastAPI
- MCPs con FastMCP

¿A santo de qué?



Pague este recibo directamente, capturando el código QR con su móvil.

AYUNTAMIENTO DE MI PUEBLO

IBI RÚSTICA ANUAL 2025

Periodo de ingreso sin recargo 01/09/2025 hasta el 20/11/2025

REF. CATASTRAL
18030A014007650000RK

DENOMINACIÓN
MAZERUCA HAFAS

PRECIO
XX,XX€

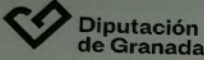
TITULAR SEGÚN EL PADRÓN:

NIF:

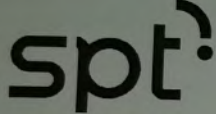
TIPO PAGO	PERIODO PAGO	EMISORA	MODO	REFERENCIA	IDENTIFICACION	IMPORTE A INGRESAR
VOLUNTARIA	01/09/2025 - 20/11/2025					
EJECUTIVA RECARGO 5%	21/11/2025 - 20/12/2025					



C.P.R.: Ejemplar para el contribuyente



Diputación de Granada



servicio provincial tributario

ESPAÑA



FRANQUEO PAGADO

CARTAS

Destino 2 G1

IBI RÚSTICA ANUAL 2025

AYUNTAMIENTO DE MI PUEBLO
Nº ORDEN:



Mi momento Dev

¿Alguien ha intentado esto antes?

← → ↺

https://github.com/search?q=catastro&type=repositories

🔍

☆

🔒

👤

🔔

📧

📁

☰

☰

🔍 catastro

🔧

+

🕒

🔗

📧

👤

Filter by

<> Code

881k

📁 Repositories

1.3k

🔔 Issues

395

🔗 Pull requests

121

🗨 Discussions

5

👤 Users

418

More

Languages

JavaScript

Python

HTML

Jupyter Notebook

Java

C#

PHP

TypeScript

CSS

R

More languages...

Advanced

Owner

Size

Number of followers

Number of forks

1.3k results (233 ms)

Sort by: Best match

🔖 Save

⋮

👤

ariseff/overcoming-catastrophic

Implementation of "Overcoming catastrophic forgetting in neural networks" in Tensorflow

Jupyter Notebook · 302 · Updated on 23 may 2020

👤

oscarfonts/catastro2postgis

Python · 28 · Updated on 3 sept 2013

👤

shivamsaboo17/Overcoming-Catastrophic-forgetting-in-Neural-Networks

Elastic weight consolidation technique for incremental learning.

neural-network · deepmind · incremental-learning · elastic-weight-consolidation

Jupyter Notebook · 139 · Updated on 23 dic 2020

📁

gisce/pycatastro

Python library to access Spanish catastro

python · api · hacktoberfest · catastro

Python · 35 · Updated on 16 ago 2024

📁

geomatico/cidownloader

Catastro Inspire Downloader

Python · 18 · Updated on 17 mar

📁

rOpenSpain/CatastRo

R package to query Sede Electrónica del Catastro (Spain)

cran · r · maps · gis · spatial

R · 24 · Updated 13 days ago

👤

Sponsor open source projects you depend on

Contributors are working behind the scenes to make open source better for everyone—give them the help and recognition they deserve.

Explore sponsorable projects →

💡 ProTip!

Press the `/` key to activate the search input again and adjust your query.

¿Coordenadas?

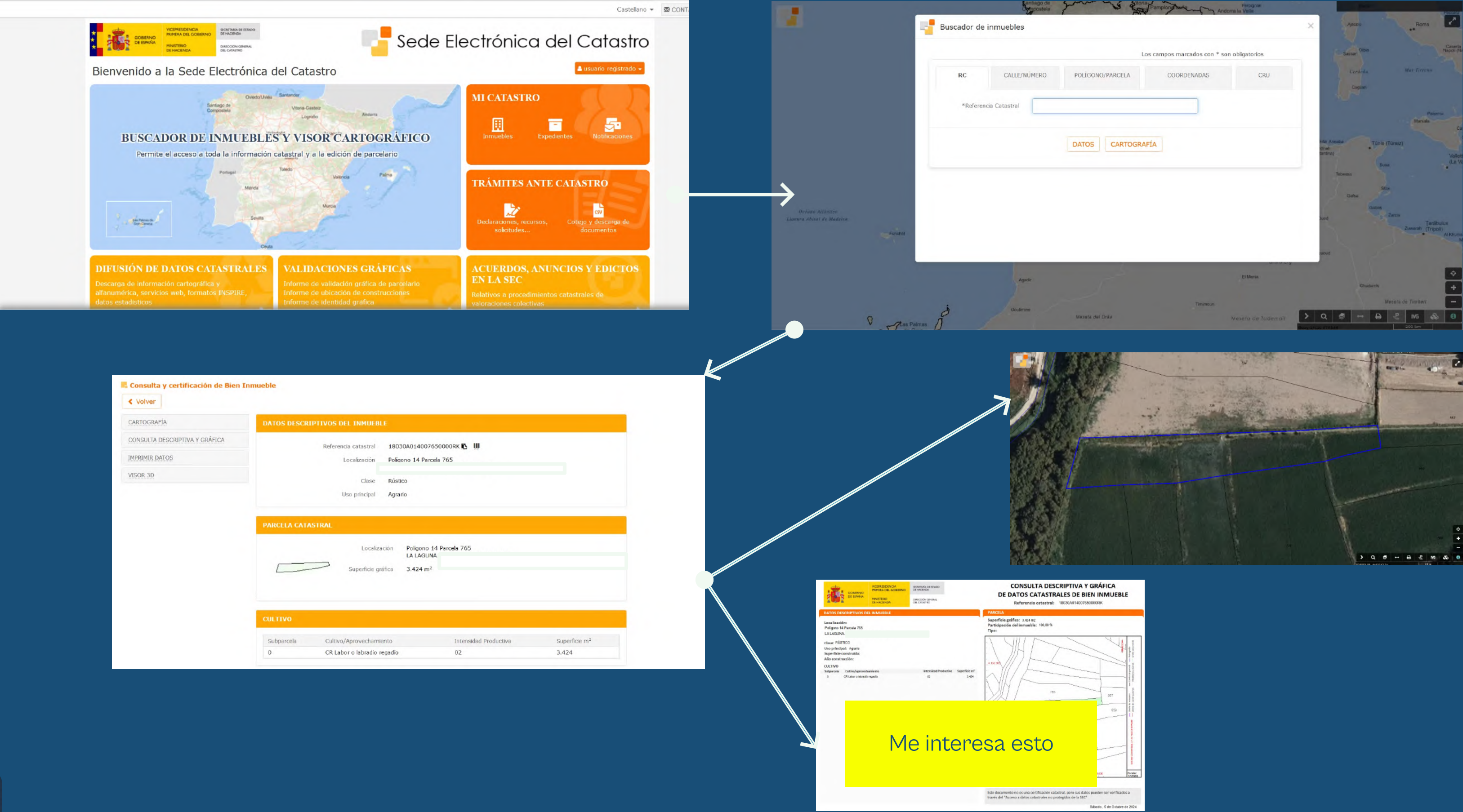
¿Mantenimiento?

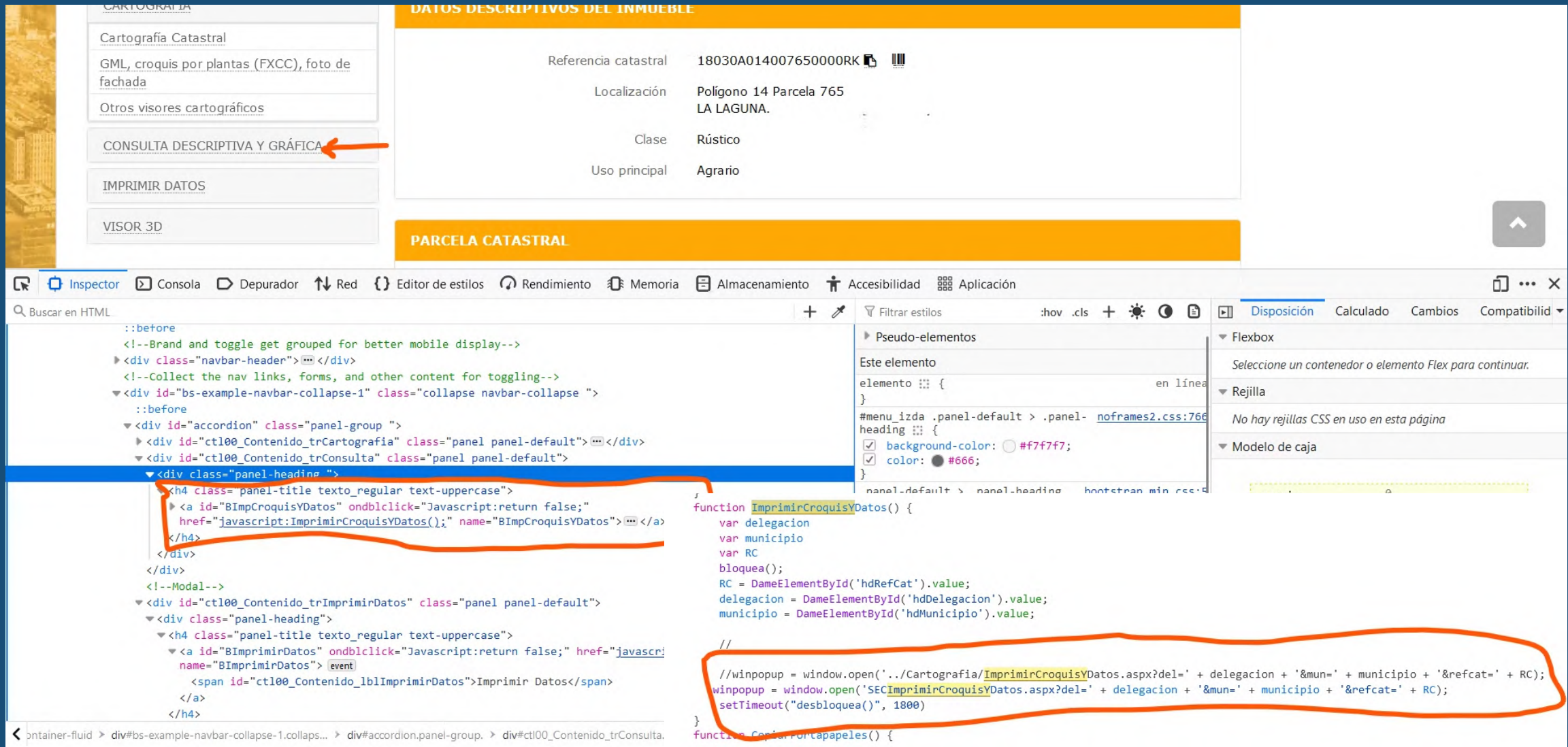
Geopackages

¿Resto info?

Try Pitch

El flujo de la Web





Delegación Municipio

18|03|0A014007650000RK

Primera versión del scraper



```
import requests
from pypdf import PdfMerger
from time import sleep
from sys import argv
from datetime import datetime

def get_data_from_catastro(reference: str):
    ref = reference
    province = ref[0:2]
    town = ref[3:5]

    print(ref, province, town)

    request = requests.get(f'https://www1.sedecatastro.gob.es/CYCBienInmueble/SECImprimirCroquisYDatos.aspx?del={province}&mun={town}&refcat={ref}', verify=False)

    if request.status_code == 200:
        with open(f'{ref}_data.pdf', 'wb') as f:
            f.write(request.content)
            f.close()

        return f'{ref}_data.pdf'

    return None
```

Iván vs El Catastro

01

Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

Replantando el proyecto como paquete

- Mi viaje a Vigo
- ¿Podemos sacar más información?
- La API del Catastro y otros menesteres

03

Conteniendo la API y aplicándolo hasta en la sopa

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.
- Una API para simplificar otra API con FastAPI
- MCPs con FastMCP

Sip, estuve en Vigo :)



📖 README

📄 Code of conduct

📄 BSD-2-Clause license

✎

desalkila

La vivienda es un derecho, no un negocio

Resumen

⚠ Warning

Este proyecto está en pausa hasta aclarar los siguientes pasos. Eres libre de hacer con este código lo que consideres dentro de los términos de la licencia (ver `LICENSE`). Si tienes cualquier duda, por favor abre un issue.

🏠 callejero

Próximos pasos

- Documentar el código actual más allá de este escueto README
- Crear las tablas `"gold"` con agregaciones que faciliten el trabajo analítico
- Crear un notebook que explique cómo trabajar con estos datos
- Tratar de replicar este esquema para otras ciudades españolas
- Incorporar el catastro, el CONEX y otras fuentes para ubicar todos los Airbnb que se puedan
- ?



- Incorporar el catastro,





Oye sobrino, tú que eres informático...

Y encima los pueblos miden de forma distinta...

1

Fanega

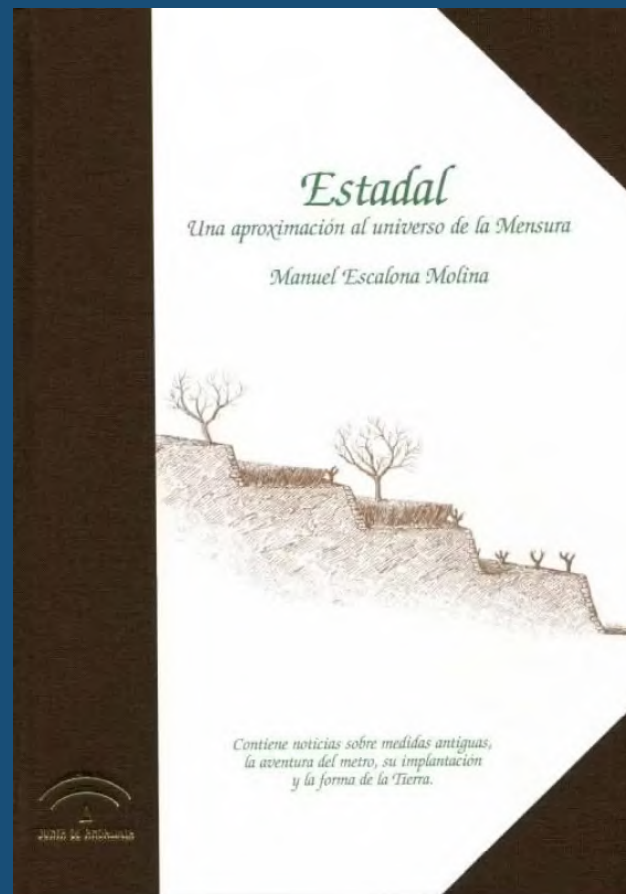
=

12

Celemines

Fanega

Con diversas denominaciones aparece esta medida agraria: fanega, fanegada, hanega, hanegada y anega⁹⁹, que se proclama, de manera destacada y a gran distancia de todas las demás, como la medida superficial tradicional más importante del país. Al menos en 38 provincias españolas se podrá encontrar utilizada esta medida superficial agraria, con cabidas muy dispares; desde la de 280 m², en Guriezo (Santander), la fanega de linar de 838 m² en León, la fanega de huerta de 1397 m² en Baza (Granada), hasta la fanegada de puño de 15.744 m² en Canarias. Casos excepcionales, por su extensión, aparecen en las Respuestas Generales del Catastro de Ensenada en la provincia de Granada, de este modo, en el municipio de Orce la fanega de secano de 3^a equivale a 111.389 m². En la publicación Pesas, Medidas y Monedas, de la Dirección General de Agricultura (hacia 1925) no consta esta fanega, ni otras similares en la provincia de Granada.



- Jaén: Guarromán: fanega de 7493 m². Benatae: fanega de 6451 m². Santistebán del Puerto y otros 21 municipios : 6439,56 m²*¹⁰³. Carboneros: 6400 m². Torres de Albánchez: 6282 m². Jaén y otros 15 municipios: fanega de 8963 varas castellanas², 6263 m²¹⁰⁴. Carchelejo: 6247 m². Carchel: 6242 m². Villanueva de la Reina: 5759 m². Andújar y otros 9 municipios: 5707 m²¹⁰⁵. Bailén, Martos, Jamilena y Torredonjimeno: 570² m².¹⁰⁶ Valdepeñas de Jaén: 5600 m². Pozo Alcón: 5050 m². Baeza: 5000 m². Baños de la Encina: 4807 m². Úbeda y otros 20 municipios: 4697 m²¹⁰⁷. Torreperogil: 4656 m². Bélmez de la Moraleda: 4600 m². Alcaudete: 4509 m². Iznatoraf: 4392 m². Jabalquinto: 4254 m². Rus y Mármol: 4087 m². Linares: 4024 m². Alcalá la Real, Castillo de Locubín y Frailes: 3566 m². Beas de Segura y Segura de la Sierra: 644 m². Fuensanta de Martos: 570 m². La Iruela: 557 m².

- Granada: Baza: fanega de huerta, 1397 m²; de vega, 1643 m²; de orillas, 2376 m²; de secano, 4996 m². Huéscar: fanega riego 1^a, 1747 m²; riego 2^a, 2037 m²; riego 3^a, 3494 m²; secano, 5531 m². Guadix: fanega de riego, 2348 m². Loja: fanega de secano, 5637 m². Guadix, Iználor y Montefrío: fanega de 4695 m². Granada, Motril y Santa Fe: fanega de secano, 4697 m². Alhama: fanega de riego y secano, 6115 m². Albuñol, de 6439,56 m²*. Castril, Castillejar (Huéscar): fanega de riego de 1^a y 2^a, 1747 m²; riego de 3^a, 2306 m². Galera (Huéscar): de riego de 1^a y 2^a, 1747 m²; de riego de 3^a, 2096 m²; de secano, 6987 m². Puebla de Don Fadrique (Huéscar): fanega de riego, 2348 m²; de secano 6439,56 m²*. Benamaurel (Baza): de riego 2147 m²; de secano, 6439,56 m²*. Caniles (Baza): fanega de riego 1^a, 1526 m²; riego 2^a, 1615 m²; de 3^a, 1712 m²; de 4^a, 1810 m²; de secano, 3144 m². Cortes (Baza): fanega de riego, 1643 m². Cúllar (Baza): fanega de riego, 1747 m²; de secano, 6439,56 m²*. Zújar (Baza): fanega de riego, 1611 m²; de secano, 3494 m². Freila (Baza): fanega de riego, 1747 m²; de secano, 3285 m². Alcázar (Albuñol): fanega de 4697 m². Orce (Huéscar): fanega riego 1^a, 1415 m²; de 2^a, 1747 m²; de 3^a, 2096 m²; secano 1^a, 3144 m²; de 2^a, 4472 m². Zafarroya (Alhama): fanega, 6261 m².

Iván vs El Catastro

01

Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

Replantando el proyecto como paquete

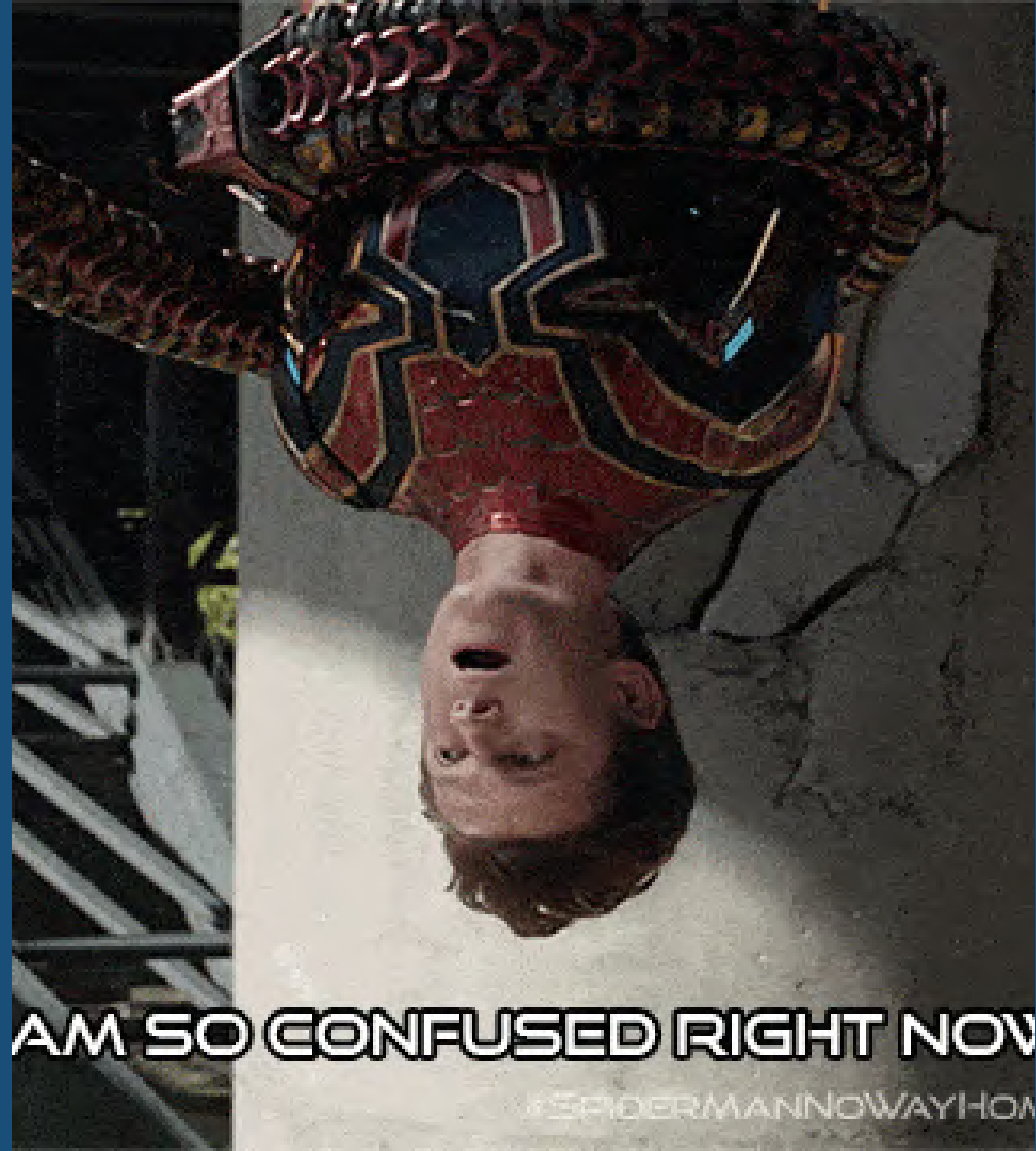
- Mi viaje a Vigo
- ¿Podemos sacar más información?
- La API del Catastro y otros menesteres

03

Conteniendo la API y aplicándolo hasta en la sopa

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.
- Una API para simplificar otra API con FastAPI
- MCPs con FastMCP

¿Y cómo saco la info?







Investigando las APIs

Google

api catastro

X🔊📷🔍

🌐👤

Todo

Videos

Imágenes

Noticias

Videos cortos

Web

Libros

Más

Herramientas



Portal de la Dirección General del Catastro
https://www.catastro.hacienda.gob.es › Webservic... PDF

Servicios web libres de la Sede Electrónica del Catastro
4 dic 2023 — Proporciona funcionalidades para acceder al callejero catastral (provincias, municipios, calles, números) y a los datos no protegidos de los ...
23 páginas



Sede Electrónica del Catastro
https://www.sedecatastro.gob.es › SECAccDescargaDatos

Difusión de datos catastrales
En primer lugar, puede descargar de forma interactiva toda la información catastral de los inmuebles (salvo titularidad y valor catastral). Esta información se ...

Tabla de contenido

1

Introducción.....

4

2

Servicios que proporcionan datos catastrales no protegidos

4

2.1

Servicio de callejero y datos catastrales no protegidos.

4

2.1.1

Listado de PROVINCIAS.....

4

2.1.2

Listado de MUNICIPIOS de una provincia.....

5

2.1.3

Listado de VÍAS de un municipio.....

6

2.1.4

Listado de los NÚMEROS de una vía.....

7

2.1.5

Consulta de DATOS CATASTRALES NO PROTEGIDOS de un inmueble identificado por su localización.

8

2.1.6

Consulta de DATOS CATASTRALES NO PROTEGIDOS de un inmueble identificado por su Referencia Catastral.

13

2.1.7

Consulta de DATOS CATASTRALES NO PROTEGIDOS de un inmueble identificado por polígono-parcela.....

14

2.2

Conversor de coordenadas.....

14

2.2.1

Conversión X,Y a RC.....

14

2.2.2

Conversión RC a X,Y.....

15

2.2.3

Localización de referencias por proximidad a coordenadas X,Y.....

16

2.3

Invocación de los servicios.....

17

2.3.1

Introducción.....

17

2.3.2

Formato de salida JSON.....

17

2.3.3

Servicios de callejero y datos catastrales no protegidos.

18

2.3.4

Conversor de coordenadas.

19

Anexo I

Errores en servicios web.

20

Anexo II

Tipos de vía.....

22

Anexo III

Ejemplos.....

23

2.3 Invocación de los servicios.

2.3.1 Introducción

La especificación de los servicios está disponible en la URL del WSDL:

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejero.svc?singleWsdl>

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejeroCodigos.svc?singleWsdl>

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejeroCodigos.svc?singleWsdl>

El endpoint para las peticiones SOAP son:

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejero.svc?singleWsdl>

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejeroCodigos.svc?singleWsdl>

<http://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejeroCodigos.svc?singleWsdl>

MY ANSWER IS

NO



2.3.2 Formato de salida JSON

Todos los servicios devuelven los datos de salida en formato XML, si se desea obtener la salida en formato JSON los servicios REST están disponibles en los siguientes enlaces.

Causísticas. Tengo...

1

Tengo una Referencia Catastral

2

Tengo una tierra: Tengo su Parcela y Polígono

3

Tengo una casa / piso: Tengo la calle y el número (?)

Causísticas. Quiero...

1

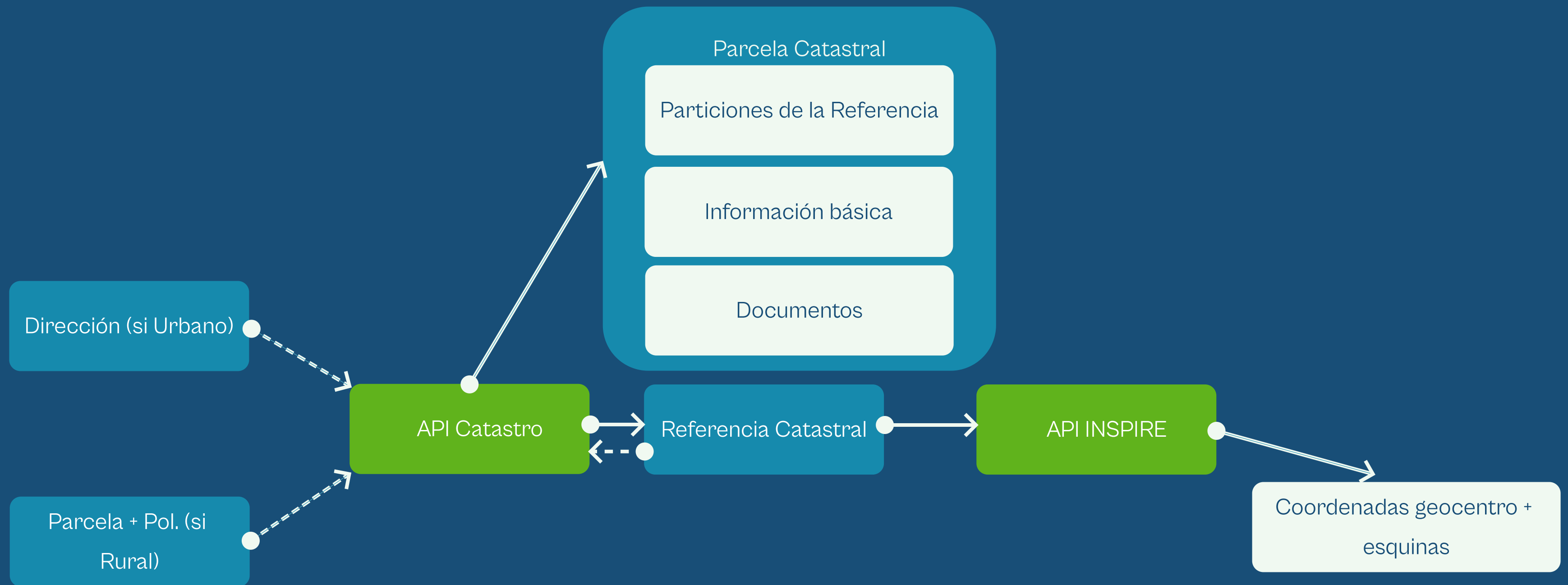
Quiero información básica: RC, Superficie, Dirección...

2

Quiero las coordenadas para dibujarlo en un mapa

3

Quiero descargar documentación relacionada creada por el Catastro



El flujo de información

Sede Electrónica del Catastro

Castellano

CONTÁCTENOS

CONSULTA Y CERTIFICACIÓN DE BIEN INMUEBLE

CARTOGRAFÍA

CONSULTA DESCRIPTIVA Y GRÁFICA

IMPRIMIR DATOS

VISOR 3D

DATOS DESCRIPTIVOS DEL INMUEBLE

Referencia catastral1528401VG3812N0001JQ

LocalizaciónCR MADRID 12
23008 JAEN (JAÉN)

ClaseUrbano

Uso principalCultural

Superficie construida143.228 m²

Año construcción2004

PARCELA CATASTRAL

Parcela construida sin división horizontal

LocalizaciónCR MADRID 12
JAEN (JAÉN)

Superficie gráfica181.741 m²

CONSTRUCCIÓN

Uso principal	Escalera	Planta	Puerta	Superficie m²	Tipo Reforma
ENSEÑANZA	1	00	A2	7.628	
ENSEÑANZA	1	00	A3	19.595	
ENSEÑANZA	1	00	A4	11.533	
ENSEÑANZA	1	00	B1	4.482	
ENSEÑANZA	1	00	B2	11.037	
ENSEÑANZA	1	00	B3	12.175	
ENSEÑANZA	1	00	B4	12.113	
ENSEÑANZA	1	00	B5	4.330	
ENSEÑANZA	1	00	C1	2.228	
ENSEÑANZA	1	00	C2	3.215	
ENSEÑANZA	1	00	C3	4.842	
ENSEÑANZA	1	00	D1	2.633	
ENSEÑANZA	1	00	D2	5.856	
ENSEÑANZA	1	00	D3	7.035	
APARCAMIENTO	1	00	01	30.346	
APARCAMIENTO	D4	-1	01	927	
ALMACEN	D4	-1	02	394	
ENSEÑANZA	D4	00	01	674	
ENSEÑANZA	D4	01	01	620	
ENSEÑANZA	D4	02	01	762	
ENSEÑANZA	D4	03	01	755	
ALMACEN	D4	04	02	48	

Un ejemplo: El Campus Las Lagunillas

Try Pitch

https://ovc.catastro.meh.es/OVCServWeb/OVCWcfCallejero/COVCCallejero.svc/json/Consulta_DNPLOC?

Provincia=Jaen&Municipio=Jaen&Sigla=CR&Calle=Madrid&Numero=12

▼ control:		▼ loint:	
cudnp:	1	es:	"1"
cucons:	22	pt:	"00"
▼ bico:		pu:	"A2"
▼ bi:		▼ dfcons:	
▼ idbi:		stl:	"7628"
cn:	"UR"	▼ dvcons:	
▼ rc:		dtip:	"ENSEÑANZA"
pc1:	"1528401"	▼ 1:	
pc2:	"VG3812N"	lcd:	"ENSEÑANZA"
car:	"0001"	▼ dt:	
cc1:	"J"	▼ lourb:	
cc2:	"Q"	▼ loint:	
▼ dt:		es:	"1"
▼ loine:		pt:	"00"
cp:	"23"	pu:	"A3"
cm:	"50"	▼ dfcons:	
cmc:	"900"	stl:	"19595"
np:	"JAÉN"	▼ dvcons:	
nm:	"JAEN"	dtip:	"ENSEÑANZA"
▼ locs:		▼ 2:	
▼ lous:		lcd:	"ENSEÑANZA"
▼ lourb:		▼ dt:	
▶ dir:	{ cv: "575", tv: "CR", nv: "MADRID", ... }	▼ lourb:	
▶ loint:	{ es: "T", pt: "OD", pu: "OS" }	▼ loint:	
dp:	"23008"	es:	"1"
dm:	"4"	pt:	"00"
ldt:	"CR MADRID 12 23008 JAEN (JAÉN)"	pu:	"A4"
▼ debi:			
luso:	"Cultural"		
sfc:	"143228"		
cpt:	"100,000000"		

```
<consulta_dnp>
<control>
<cudnp>NÚMERO DE INMUEBLES DE LOS QUE SE PROPORCIONAN DATOS</cudnp>
<cucons>NÚMERO DE UNIDADES CONSTRUCTIVAS (INCLUYENDO ELEMENTOS COMUNES)</cucons>
<cucul>NÚMERODE SUBPARCELAS (CULTIVOS)</cucul>
</control>
<bico>
<bi>
<idbi>
<cn>TIPO DE BIEN INMUEBLE</cn>
<rc>REFERENCIA CATASTRAL
<pc1>POSICIONES 1-7 DE LA REFERENCIA CATASTRAL (RC) DEL INMUEBLE</pc1>
<pc2>POSICIONES 8-14 DE LA RC DEL INMUEBLE</pc2>
<car>POSICIONES 15-19 DE LA RC (CARGO)</car>
<cc1>PRIMER DÍGITO DE CONTROL DE LA RC</cc1>
<cc2>SEGUNDO DÍGITO DE CONTROL DE LA RC </cc2>
</rc>
</idbi>
<dt>DOMICILIO TRIBUTARIO DEL INMUEBLE (SI EXISTE)
<loine>LOCALIZACIÓN DEL INE
<cp>CÓDIGO DE PROVINCIA INE</cp>
<cm>CÓDIGO DE MUNICIPIO INE</cm>
</loine>
<cmc>CÓDIGO DE MUNICIPIO DGC</cmc>
<np>NOMBRE DE PROVINCIA</np>
<nm>NOMBRE DE MUNICIPIO</nm>
<nem>NOMBRE ENTIDAD MENOR (SI EXISTE)</nem>
<locs>
<lous>LOCALIZACIÓN MUNICIPAL DEL BIEN URBANO (SI EXISTE)
<lourb>LOCALIZACIÓN URBANA
<dir>DIRECCIÓN
<cv>CÓDIGO DE LA VÍA</cv>
<tv>TIPO DE VÍA</tv>
<nv>NOMBRE DE VÍA</nv>
<pnp>PRIMER NÚMERO DE POLICÍA (SI EXISTE)</pnp>
<plp>PRIMERA LETRA ASOCIADA AL NÚMERO DE POLICÍA (SI EXISTE)</plp>
<snp>SEGUNDO NÚMERO DE POLICÍA (SI EXISTE)</snp>
<slp>SEGUNDA LETRA ASOCIADA AL NÚMERO DE POLICÍA (SI EXISTE)</slp>
<km>KILÓMETRO (SI EXISTE)</km>
<td>DIRECCIÓN NO ESTRUCTURADA (SI EXISTE)</td>
</dir>
<loint>LOCALIZACIÓN INTERNA (SI EXISTE)
```



Un ejemplo: El Campus Las Lagunillas

Houston, tenemos problemas.

1

El Callejero es un lío

2

Los Números a veces no coinciden

3

Tengo que estar todo el rato mirando montones de palabros para llegar a algo.

4

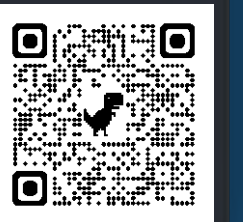
Tengo que al menos llevármelo a Python para trabajar con él

¿Cómo podemos ver los mapas?

Puntos rápidos

- Folium es una librería basada en Leaflet (una librería de JS), que a su vez usa mapas de OpenStreetMap.
- Permite incorporar Tilesets personalizados.
- Se puede guardar como HTML
- Se pueden añadir items como Puntos con iconos, Líneas, Figuras...
- En nuestro caso, tenemos la información geográfica → Pequeñas conversiones de datos y... ¡Presto!

```
1 import folium
2
3 map_geo = folium.Map(location=geocentro, zoom_start=18)
4 coordinates = [] # Formato [[lat, lon], [lat, lon], ...]
5 folium.raster_layers.WmsTileLayer(url = 'https://www.ign.es/wms-inspire/pnoa-ma?SERVICE=WMS&',
6                                   layers = 'OI.OrthoimageCoverage',
7                                   transparent = False,
8                                   control = True,
9                                   name = 'Ortofoto PNOA',
10                                  overlay=True,
11                                  show = True,
12                                  zoom_control=False,
13                                  scrollWheelZoom=False,
14                                  dragging=False
15                                  ).add_to(map_geo)
16 folium.PolyLine(
17     locations=coordinates,
18     color="#FF0000",
19     weight=3,
20 ).add_to(map_geo)
21
22 folium.LayerControl().add_to(map_geo)
23
24 map_geo.save('mapa.html')
```





Ok, usemos requests

Puntos rápidos

- Requests es una librería para gestionar peticiones HTTP.
- Para recoger la información que va por parámetros GET (es decir, los que van después de la ruta con el símbolo ? y separados por &), usaremos el método requests.get y el argumento params.
- Podemos saber si la respuesta es un OK mirando el status_code del request.
- La información que nos importa está en el content. Quizás conviene hacer un json.loads a la info para tratarla como un diccionario.

```
1 req = requests.get(f'{URL_BASE_CALLEJERO}/Consulta_DNPLOC',
2                     params={
3                         'Provincia': info_calle.municipio.provincia,
4                         'Municipio': info_calle.municipio.municipio,
5                         'Sigla': info_calle.tipo_via,
6                         'Calle': info_calle.calle,
7                         'Numero': numero
8                     })
9
10 if req.status_code == 200 and comprobarErrores(req.json()):
11     info_cadastrale = json.loads(req.content)
12     cudnp = info_cadastrale.get("consulta_dnplocResult", {})
```

```
1 self.municipio = info_cadastrale.get('consulta_dnplocResult')
2                     .get('bico') # No especifica
3                     .get('bi') # No especifica
4                     .get('dt') # Domicilio tributario del inmueble (si existe)
5                     .get('nm') # Nombre del municipio
```

El caso de INSPIRE

Puntos rápidos

- INSPIRE, a diferencia de Catastro, SOLO usa XML.
- "A veces, un proveedor de API te odia y solamente ofrece respuestas en XML"- Joel Grus, Ciencia de Datos desde 0.
- ¿Y si se pudiese convertir el XML en un JSON o Dict de Python? Aquí entra en juego XML2Dict.

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
<!--Parcela Catastral de La D.G. del Catastro.-->
-<!--
  La precisión es la que corresponde nominalmente a la escala de captura de la cartografía
-->
-<FeatureCollection xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd http://inspire.ec.europa.eu/schemas/cp/4.0 http://inspire.ec.europa.eu/schemas/cp/4.0/CadastralParcels.xsd"
timeStamp="2025-05-24T19:49:30" numberMatched="1" numberReturned="1">
  -<member>
    -<cp:CadastralParcel gml:id="ES.SDGC.CP.0915103UF7601N">
      -<cp:areaValue uom="m2">57376</cp:areaValue>
      -<cp:beginLifespanVersion>2007-12-20T00:00:00</cp:beginLifespanVersion>
      -<cp:endLifespanVersion xsi:nil="true" nilReason="http://inspire.ec.europa.eu/codelist/VoidReasonValue/Unpopulated"/>
      -<cp:geometry>
        -<gml:MultiSurface gml:id="MultiSurface_ES.SDGC.CP.0915103UF7601N" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
          -<gml:surfaceMember>
            -<gml:Surface gml:id="Surface_ES.SDGC.CP.0915103UF7601N.1" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
              -<gml:patches>
                -<gml:PolygonPatch>
                  -<gml:exterior>
                    -<gml:LinearRing>
                      -<gml:posList srsDimension="2" count="100">
36.688058 -4.445289 36.688081 -4.445357 36.688158 -4.445591 36.688235 -4.445826 36.688369 -4.446234 36.688387 -4.446289 36.688455 -4.446497 36.688517 -4.446683 36.688536 -4.446739
36.688586 -4.44689 36.688595 -4.446915 36.688601 -4.446934 36.6886 -4.446934 36.688601 -4.446937 36.688601 -4.446936 36.688609 -4.446963 36.68861 -4.446966 36.688611 -4.446966
36.68862 -4.446992 36.688619 -4.446993 36.68862 -4.446995 36.688621 -4.446995 36.688625 -4.447007 36.688772 -4.446936 36.688766 -4.446916 36.688781 -4.446908 36.688779 -4.446903
36.688797 -4.446895 36.688824 -4.446882 36.688835 -4.446915 36.688849 -4.446908 36.68886 -4.446945 36.688986 -4.446884 36.689096 -4.446927 36.689099 -4.447038 36.689185 -4.447209
36.689182 -4.447212 36.689223 -4.447292 36.689328 -4.447573 36.689351 -4.447561 36.689357 -4.447578 36.689359 -4.44758 36.689363 -4.447582 36.689367 -4.447584 36.689371 -4.447584
36.689375 -4.447583 36.689524 -4.4475 36.689795 -4.447349 36.68973 -4.447167 36.689684 -4.447035 36.689662 -4.446973 36.689679 -4.446964 36.689708 -4.44695 36.689708 -4.446953
36.689936 -4.446841 36.689891 -4.446701 36.689883 -4.446674 36.689886 -4.446672 36.689938 -4.446646 36.689915 -4.446572 36.689901 -4.446528 36.689835 -4.446318 36.689831 -4.446305
36.68984 -4.446301 36.690319 -4.446049 36.690195 -4.445662 36.690049 -4.445206 36.690045 -4.445193 36.690035 -4.445164 36.690019 -4.445112 36.690018 -4.445111 36.690014 -4.445097
36.689935 -4.444853 36.689566 -4.443705 36.68954 -4.443623 36.689533 -4.443614 36.68952 -4.443603 36.689511 -4.443596 36.689504 -4.443593 36.689464 -4.443584 36.689442 -4.44358
36.68942 -4.443573 36.689398 -4.443565 36.689387 -4.443554 36.689376 -4.443542 36.688659 -4.443968 36.688549 -4.444033 36.688549 -4.444033 36.68826 -4.444206 36.688026 -4.444347
36.687823 -4.44447 36.687801 -4.444507 36.687807 -4.444524 36.687814 -4.444544 36.687846 -4.444642 36.687882 -4.444751 36.687981 -4.445055 36.688014 -4.445155 36.688019 -4.445168
36.688058 -4.445289
                      -</gml:posList>
                    -</gml:LinearRing>
                  -</gml:exterior>
                -</gml:PolygonPatch>
              -</gml:patches>
            -</gml:Surface>
          -</gml:surfaceMember>
        -</gml:MultiSurface>
      -</cp:geometry>
    -</cp:CadastralParcel>
  -</member>
-</FeatureCollection>
```

```
1 import xmltodict
2
3 geometry_request = requests.get(f'{URL_BASE_GEOGRAFIA}',
4                                 params={
5                                     'service': 'wfs',
6                                     'version': '2',
7                                     'request': 'getfeature',
8                                     'STOREDQUERY_ID': 'GetParcel',
9                                     'refcat': self.rc,
10                                    'srsname': projection
11                                 })
12
13 geometry = xmltodict.parse(geometry_request.content)
14 geoposition = geometry.get('FeatureCollection')
15     .get('member')
16     .get('cp:CadastralParcel')
17     .get('cp:referencePoint')
18     .get('gml:Point')
19     .get('gml:pos')
20     .split(' ')
```

"Pero la info de los números y calles..."

Bueno...

Pillemos el Callejero de Catastro

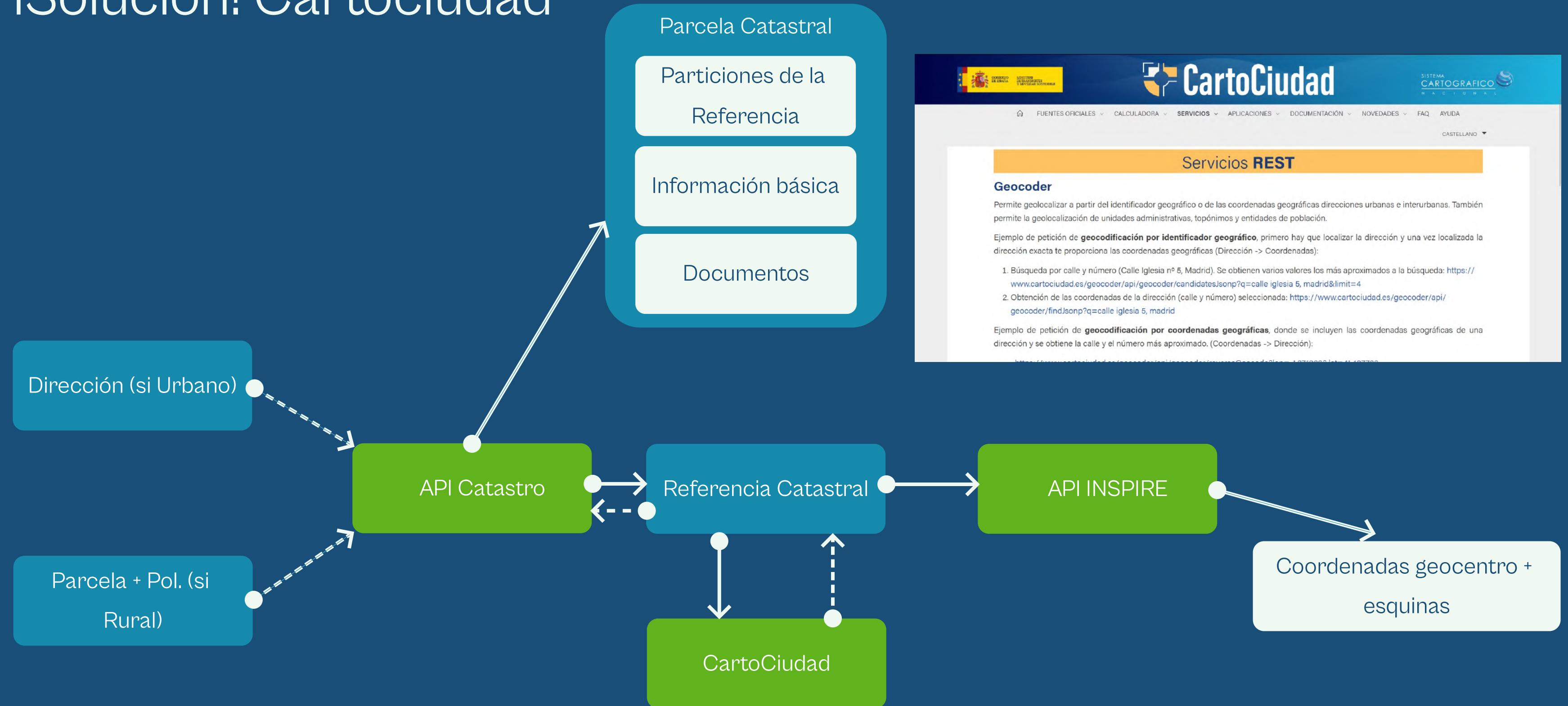
```
1 def listar_tipos_via():
2     """
3     Retorna una lista de los tipos de vía disponibles.
4     Returns:
5     list: Una lista de los tipos de vía disponibles.
6     """
7
8     return TIPOS_VIA
9
10 def listar_calles(provincia: str, municipio: str):
11     """
12     Devuelve una lista de calles para una provincia y municipio dados.
13     Args:
14     provincia (str): El nombre de la provincia.
15     municipio (str): El nombre del municipio.
16     Returns:
17     list: Una lista de calles en formato "tipo de vía nombre de vía".
18     """
19
20     provincia_final = MAPEOS_PROVINCIAS.get(provincia.capitalize())
21     if provincia.capitalize() in MAPEOS_PROVINCIAS.keys():
22         else provincia
23     if provincia_final.upper() in listar_provincias()
24     and municipio.upper() in listar_municipios(provincia=provincia_final):
25         response = requests.get(f'{URL_BASE_CALLEJERO}/ObtenerCallejero',
26                                 params={
27                                     'Provincia': provincia_final,
28                                     'Municipio': municipio
29                                 })
30     if response.status_code == 200 and comprobar_erroses(response.json()):
31         calles_dict_raw = json.loads(response.content)
32         return [{"calle.get('dir').get('tv')}: {calle.get('dir').get('nv')}"]
33         for calle in calles_dict_raw.get('consulta_callejeroResult')
34             .get('callejero')
35             .get('calle')]
36     else:
37         return []
38     else: return []
```

No tan bueno...

Hay veces que los números de Catastro no coinciden con los Postales



¡Solución! Cartociudad





¿Y cómo puedo usar este código en cualquier proyecto sin tener que estar haciendo copypaste?

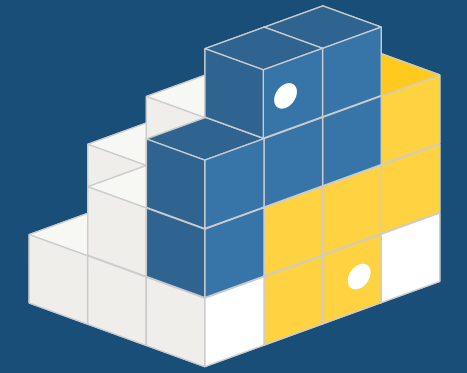
hatch como creador de paquetes

- Hatch es un gestor de paquetes fácil de usar pero con mucha expansibilidad.
- CLI fácil y muy rápida
- Facilita la descarga y gestión de versiones de Python para testear
- Permite la ejecución de scripts de instalación y uso de uv.
- Simplifica el versionado
- Simplifica la construcción y publicación a PyPI.

```
1 pip install hatch
```



Hablemos de paquetes



En programación, un paquete es una colección de código organizado en módulos que provee una funcionalidad. Esto nos ayuda a organizar, reutilizar y compartir código eficientemente.

01

ES UNA COLECCIÓN DE MÓDULOS

Contiene toda la funcionalidad relacionada en unos pocos módulos.



02

GESTIONA ESPACIOS DE NOMBRES

Evita conflictos entre funciones y clases ya que cada paquete tiene su espacio de nombres.



03

FACILITA LA REUSABILIDAD

Con ello, se puede compartir y utilizar ese código para montones de proyectos.



04

AYUDA A GESTIONAR DEPENDENCIAS

Simplifica la instalación, gestión y actualización de paquetes.



Entonces...

01

Iniciar el paquete y copiar el código



```
1 hatch new <nombre_paquete>
```

02

Hacer tests



```
1 hatch test
```

03

Construir y subir



```
1 hatch build
```

Entonces...

01

Iniciar el paquete y copiar el código

```
ivan@DESKTOP-26I4IUM: /mnt X + v
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos$ hatch new estoessunpaquete
estoesunpaquete
├── src
│   └── estoessunpaquete
│       ├── __about__.py
│       └── __init__.py
├── tests
│   └── __init__.py
├── LICENSE.txt
├── README.md
└── pyproject.toml
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos$ |
```

Pon aquí tu código



```
1 hatch new <nombre_paquete>
```



```
1 hatch new --init
```

Fun fact: Usé en una primera versión setuptools, pero me salían unos avisos raros de PyPI al subir y tuve que acabar usando Hatch. Como tenía un archivo setup.py, me lo convirtió a un pyproject.toml

Entonces...

02

Hacer tests



1 hatch test

```
[[tool.hatch.envs.test.matrix]]
python = ["3.8", "3.9", "3.10", "3.11", "3.12", "3.13"]
```

```
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos/escatastrolib$ hatch test --all
===== hatch-test.py3.13 =====
test session starts
platform linux -- Python 3.13.1, pytest-8.3.5, pluggy-1.6.0
rootdir: /mnt/e/Proyectos/escatastrolib
configfile: pyproject.toml
plugins: rerunfailures-14.0, xdist-3.6.1, mock-3.14.0
collected 16 items

tests/test_calle.py ... [ 18%]
tests/test_info_catastral.py ..... [ 56%]
tests/test_municipio.py ... [ 75%]
tests/test_utils.py .... [100%]

===== 16 passed in 29.40s =====
===== hatch-test.py3.12 =====
test session starts
platform linux -- Python 3.12.8, pytest-8.3.5, pluggy-1.6.0
rootdir: /mnt/e/Proyectos/escatastrolib
configfile: pyproject.toml
plugins: rerunfailures-14.0, xdist-3.6.1, mock-3.14.0
collected 16 items

tests/test_calle.py ... [ 18%]
tests/test_info_catastral.py ..... [ 56%]
tests/test_municipio.py ... [ 75%]
tests/test_utils.py .... [100%]
```

Entonces...

03

Construir y subir

```
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos/escatastrolib$ hatch build
DEPRECATION: --no-python-version-warning is deprecated. pip 25.1 will enforce this behaviour change. A possible replacement is to remove the flag as it's a no-op. Discussion can be found at https://
ithub.com/pypa/pip/issues/13154
dist/escatastrolib-0.0.1rc2.tar.gz
dist/escatastrolib-0.0.1rc2-py2.py3-none-any.whl
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos/escatastrolib$ hatch publish -r test
2025/05/25 14:48:06.115312 cmd_run.go:1285: WARNING: cannot start document portal: dial unix /run/user/1000/bus: connect: no such file or directory
2025/05/25 14:48:07.019655 cmd_run.go:1285: WARNING: cannot start document portal: dial unix /run/user/1000/bus: connect: no such file or directory
? Master password: [hidden]
2025/05/25 14:48:15.080117 cmd_run.go:1285: WARNING: cannot start document portal: dial unix /run/user/1000/bus: connect: no such file or directory
2025/05/25 14:48:17.087177 cmd_run.go:1285: WARNING: cannot start document portal: dial unix /run/user/1000/bus: connect: no such file or directory
Password / Token:
dist/escatastrolib-0.0.1rc2-py2.py3-none-any.whl ... already exists
dist/escatastrolib-0.0.1rc2.tar.gz ... already exists
```

bitwarden-keyring 0.3.1

pip install bitwarden-keyring

Buscar proyectos

AyudaPatrocinadoresAccederRegistrarse

keyring 25.6.0

pip install keyring

✓ Versión más reciente

Publicación: 25 dic 2024

Store and access your passwords safely.

Navegación

Descripción de proyecto

Histórico de versiones

Archivos de descarga

Verified details

Responsables

Metainformación

Descripción de proyecto

pypi v25.6.0python 3tests failingRuffdocs passano skeleton 2024lifted!chaton gitter

The Python keyring library provides an easy way to access the system keyring service from python. It can be used in any application that needs safe password storage.

These recommended keyring backends are supported:

- macOS [Keychain](#)
- Freedesktop [Secret Service](#) supports many DE including GNOME (requires [secretstorage](#))
- KDE4 & KDE5 [KWallet](#) (requires [dbus](#))
- [Windows Credential Locker](#)

Other keyring implementations are available through [Third-Party Backends](#).

Installation - Linux

On Linux, the KWallet backend relies on [dbus-python](#), which does not always install correctly when using pip



Buscar proyectos

AyudaPatrocinadoresAccederRegistrarse

ESCatastroLib 0.0.1rc0

pip install ESCatastroLib

✓ Versión más reciente

Publicación: 9 feb 2025

Description

Navegación

Descripción de proyecto

Histórico de versiones

Archivos de descarga

Verified details

These details have been [verified by PyPI](#)

Responsables



IvanInVR

Unverified details

These details have **not** been verified by PyPI

Enlaces del proyecto

Homepage

Descripción de proyecto

ESCatastroLib

La librería ESCatastroLib proporciona una interfaz sencilla para consultar el Catastro de España y obtener información geográfica precisa. Con esta librería, puedes acceder a datos como la ubicación de parcelas, características de construcción y valores geográficos.

Clases

Clase Municipio

Representa un municipio (ciudad o pueblo).

Atributos/Inputs del constructor

`provincia (str)`: El nombre de la provincia donde se encuentra el municipio. Un valor por defecto se establece y se puede proporcionar un valor diferente si es necesario.

`municipio (str, opcional)`: El nombre del municipio. Si no se proporciona al crear una instancia, se lanzará una excepción. Si se especifica, debe coincidir al menos parcialmente con el nombre de un municipio válido encontrado

<https://pypi.org/project/ESCatastroLib/>

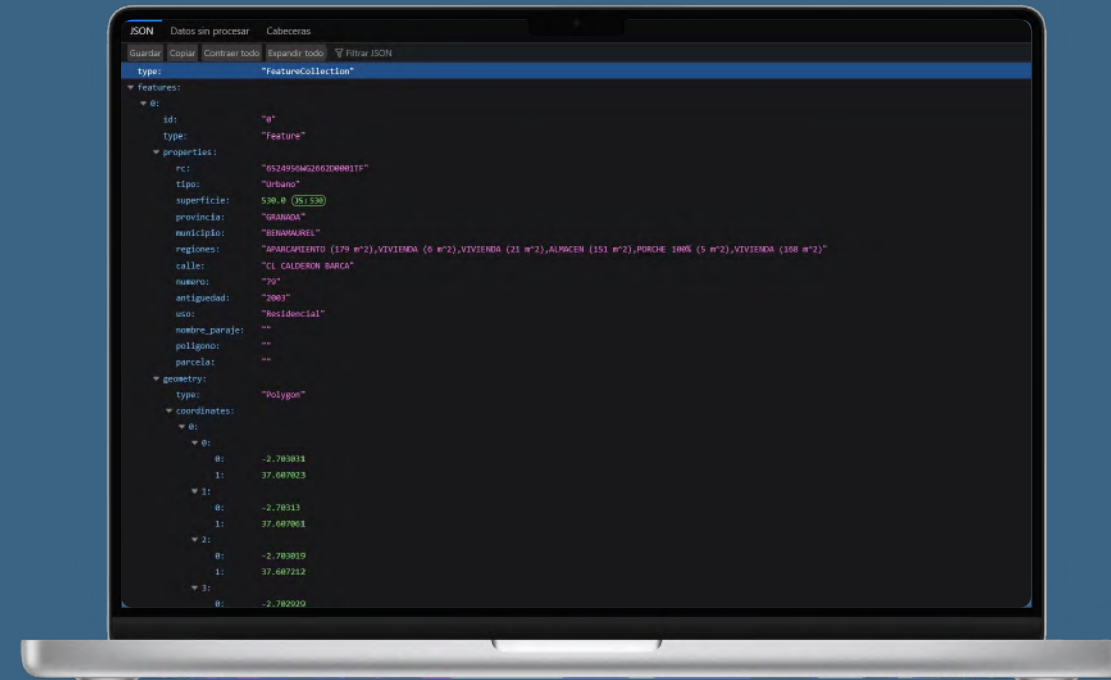
El resultado



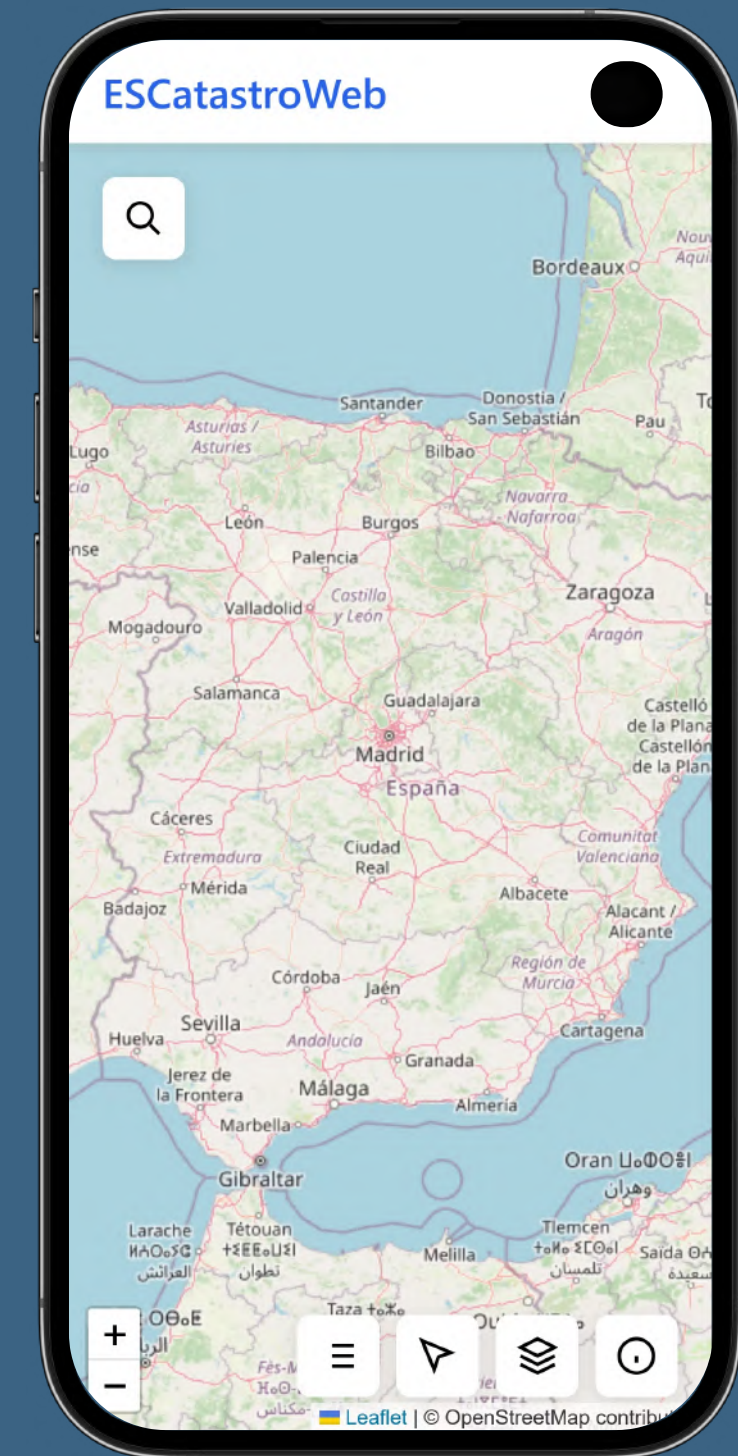
Y con el paquete...
¿Qué puedo hacer?

Una API Proxy

O sea, una API que te redirige a una API



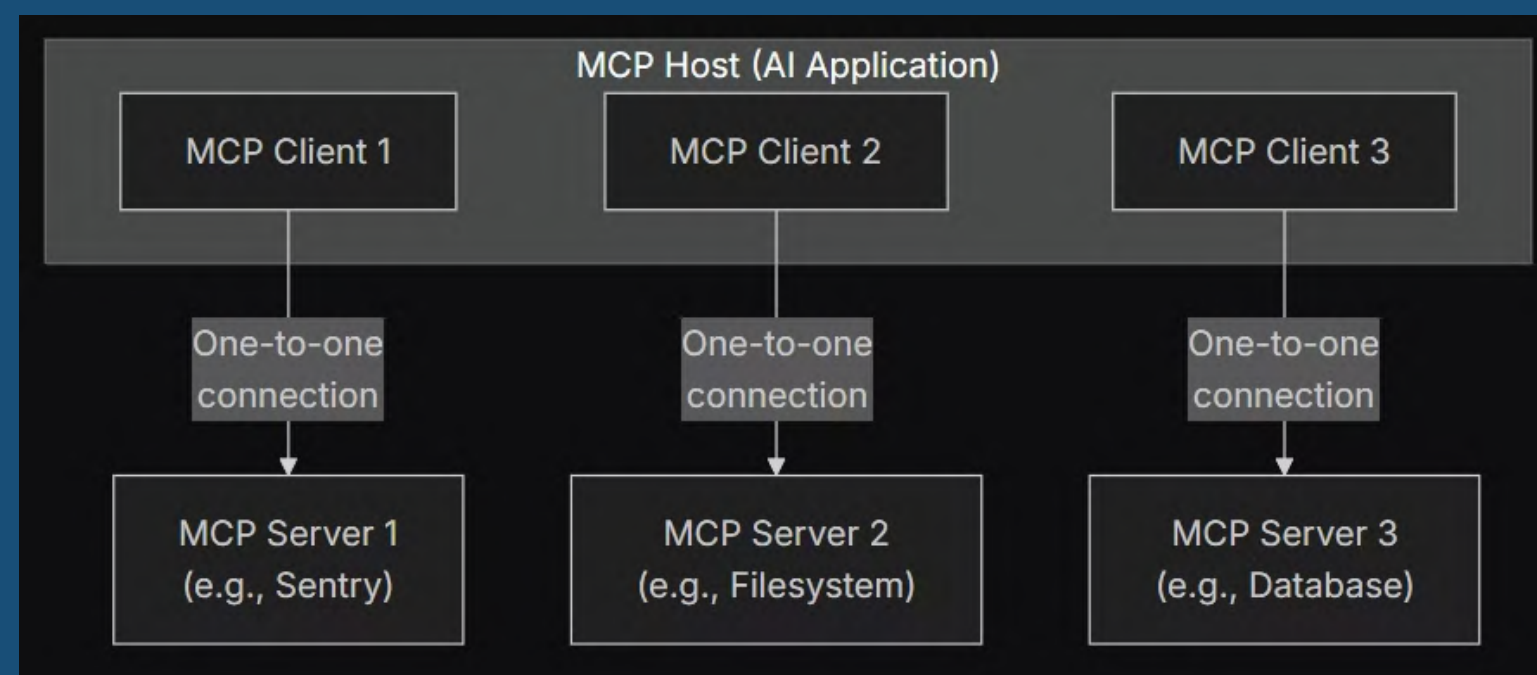
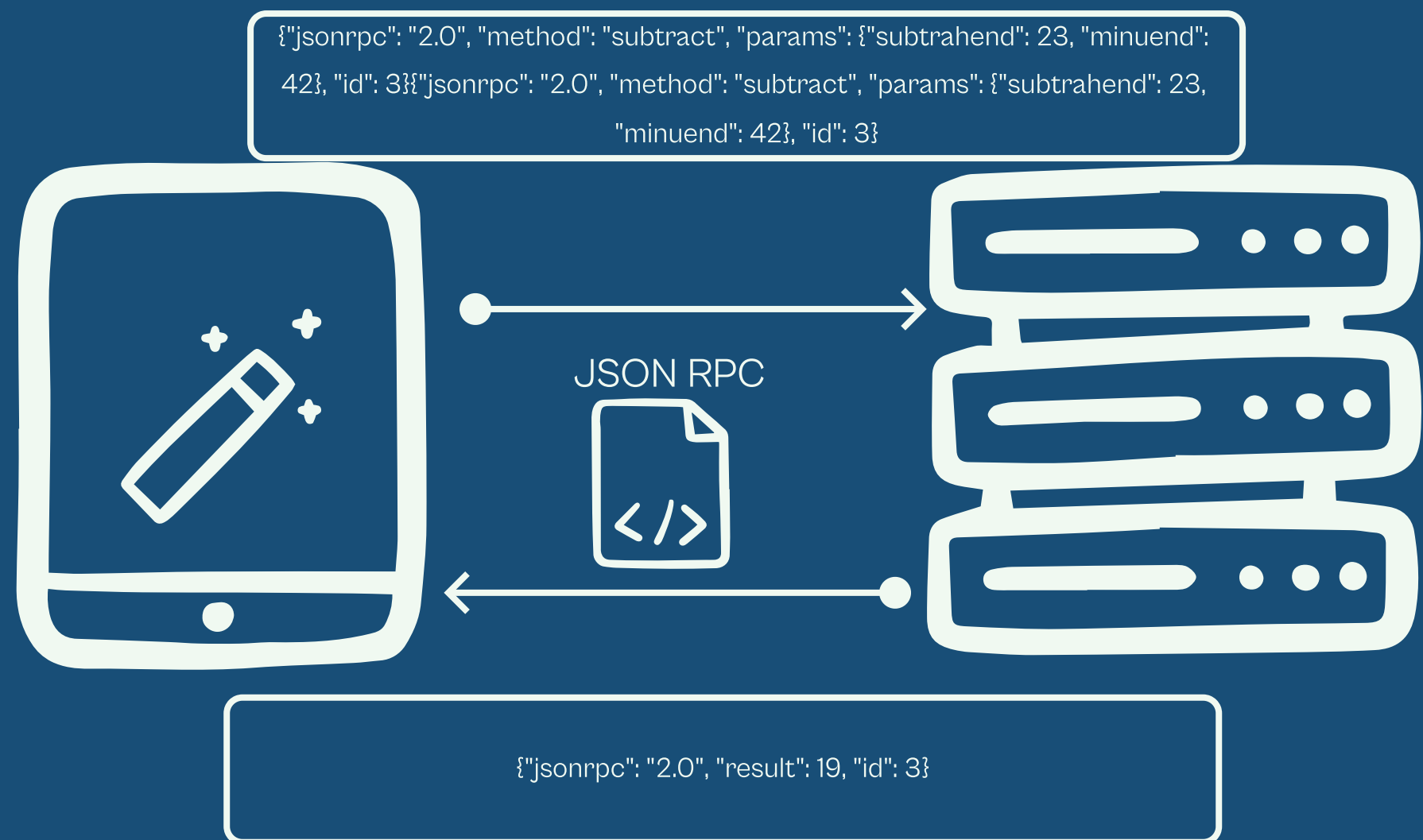
Y con eso una Web



O un Servidor MCP



¿Qué es eso de los MCP?



La librería FastMCP

Welcome to FastMCP 2.0!

Copy page

The fast, Pythonic way to build MCP servers and clients.



FastMCP is the standard framework for building MCP applications. The **Model Context Protocol** (MCP) provides a standardized way to connect LLMs to tools and data, and FastMCP makes it production-ready with clean, Pythonic code:

```
1 from fastmcp import FastMCP
2
3 mcp = FastMCP("My MCP Server")
4
5 @mcp.tool
6 def greet(name: str) -> str:
7     return f"Hello, {name}!"
8
9 if __name__ == "__main__":
10     mcp.run(transport="http", port=8000)
```

La librería FastMCP



TOOLS/HERRAMIENTAS

Aquellas herramientas que permiten ejecutar tareas remotas.

Estas pueden alterar el sistema al que se llama.

Parecido al POST/PUT/DELETE en REST

```
@mcp.tool
def add(a: int, b: int) -> int:
    """Adds two integer numbers together."""
    return a + b
```



RESOURCES/RECURSOS

Aquellas peticiones que SOLO permiten leer información.

Parecido al GET en REST

```
# Resource returning JSON data (dict is auto-serialized)
@mcp.resource("data://config")
def get_config() -> dict:
    """Provides application configuration as JSON."""
    return {
        "theme": "dark",
        "version": "1.2.0",
        "features": ["tools", "resources"],
    }
```



PROMPTS/PLANTILLAS DE PREGUNTA

Son plantillas que la propia IA te autocompleta para que puedas

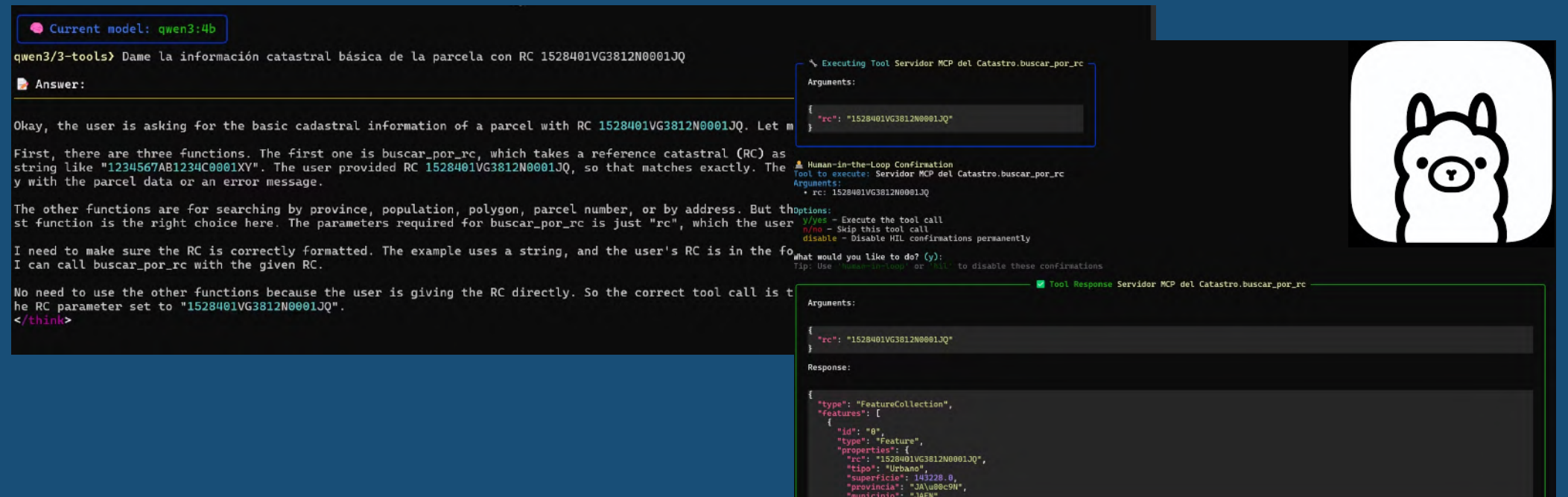
Parecido a usar Swagger en una API.

```
@mcp.prompt
def ask_about_topic(topic: str) -> str:
    """Generates a user message asking for an explanation of a topic."""
    return f"Can you please explain the concept of '{topic}'?"
```

Probando nuestro server

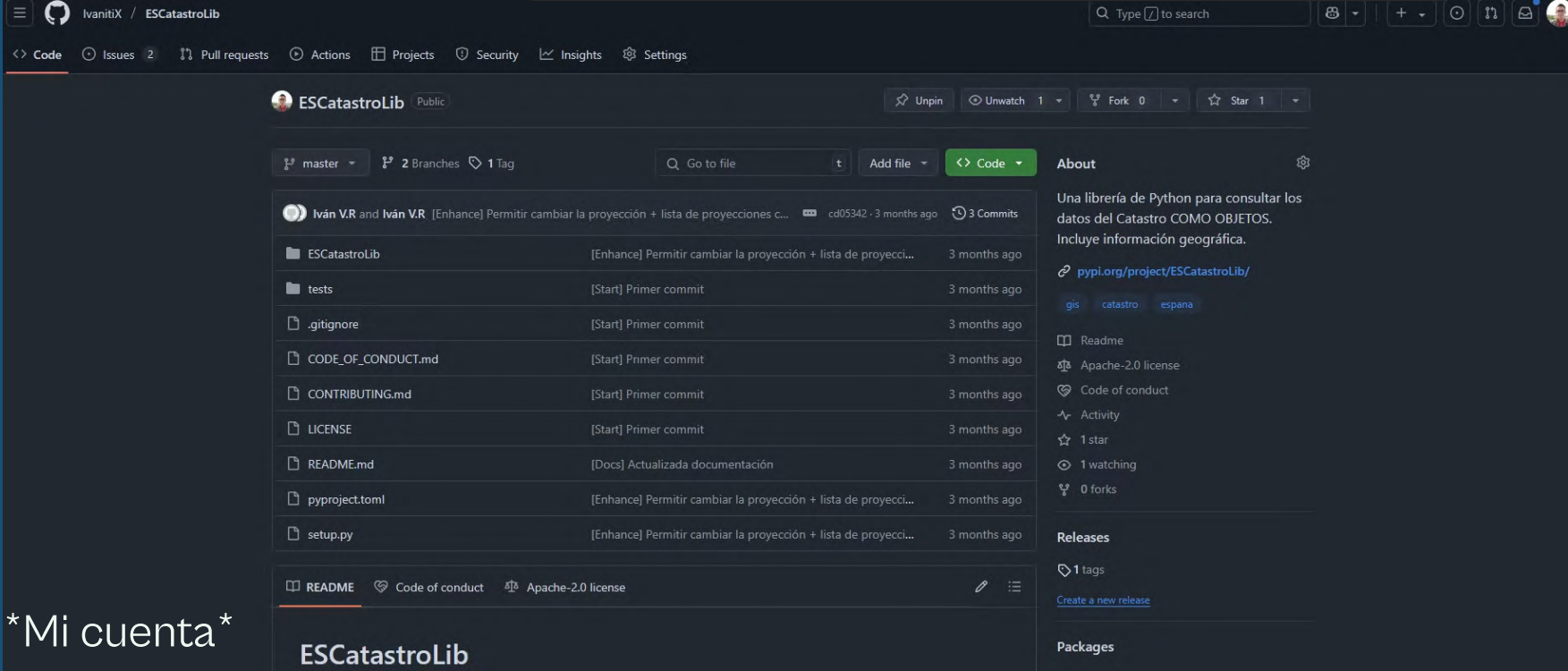
```
1 {
2   "mcpServers": {
3     "Servidor MCP del Catastro": {
4       "command": "uv",
5       "args": [
6         "run",
7         "--with",
8         "fastmcp",
9         "fastmcp",
10        "run",
11        "/mnt/e/Proyectos/catastromcp/main.py:catastro_mcp"
12      ]
13    }
14  }
15 }
```

```
1 uv run fastmcp install gemini-cli <archivo>:<mcp>
```



El endgame

<https://github.com/IvanitiX/ESCatastroLib>



Público, ¿qué habéis aprendido?

Puedes darme tu feedback
aquí. ¡Ojo que sorteo algo!



¡Muchas gracias!

¿PATO?

¿Me estás buscando en redes?

Try Pitch  @IvanInVR(@mastodon.la)

 @IvanitiX

 /ivan-vr

