

A Spider-Man figure is shown from the waist up, sitting and reading a book titled "The Decision Book" by Michael Mauboussin. The book is open, showing text and diagrams. The background is a bright, hazy sky with stone walls on either side. The text "No es magia, es Python: Tejiendo una red para atrapar las APIs del Catastro" is overlaid in white on a dark blue background.

No es magia, es Python: Tejiendo una red para atrapar las APIs del Catastro



Iván Valero Rodríguez

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

La saga de Iván vs El Catastro

01

Homecoming: Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

Iván, Far From Home: Replantando el proyecto como paquete

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

03

Conteniendo la API en el Multiverso de La Locura

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.

La saga de Iván vs El Catastro

01

Homecoming: Scrapeando la página

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

02

Iván, Far From Home: Replantando el proyecto como paquete

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

03

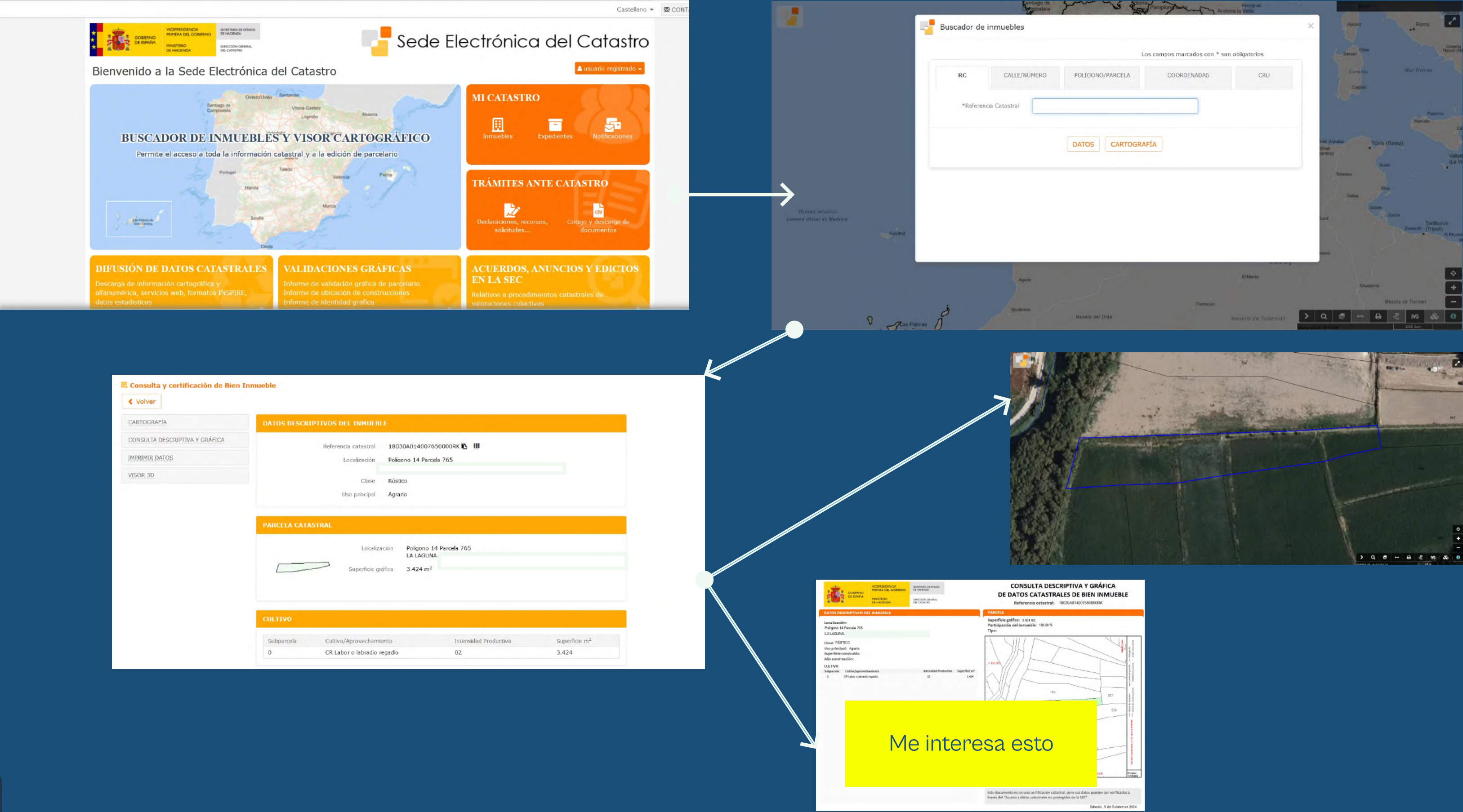
Conteniendo la API en el Multiverso de La Locura

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.

¿A santo de qué?

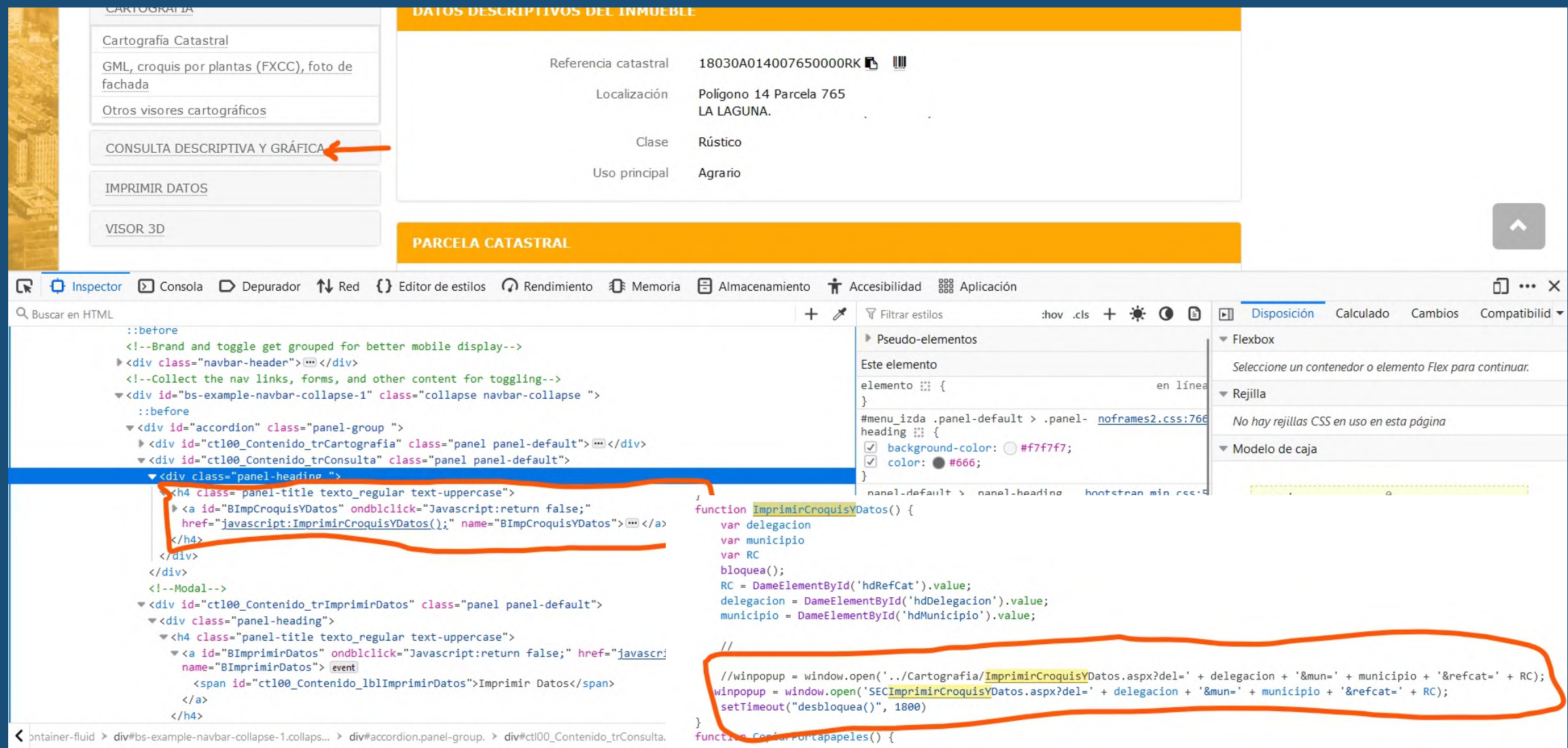
AYUNTAMIENTO DE MI PUEBLO		IBI RÚSTICO		20093118116BA02R000000 N° fijo: BA116000000000	A PAGAR
Periodo de ingreso sin recargo		hasta el			
REF. CATASTRAL		DENOMINACIÓN		PRECIO	
18030A014007650000RK		AS MARALIÑAS		XX,XX€	
...					
...					
...					
APELLIDO APELLIDO2 NOMBRE		00000000T			
CL DIRECCION ,0,00		GRANADA		9052118000000000000000000000000096000	
Emisora: 180002		Referencia: 0000000000		Identificación: 100000000	

El flujo de la Web





Mi momento Dev



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
```

La saga de Iván vs El Catastro

01

Homecoming: Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

Iván, Far From Home: Replantando el proyecto como paquete

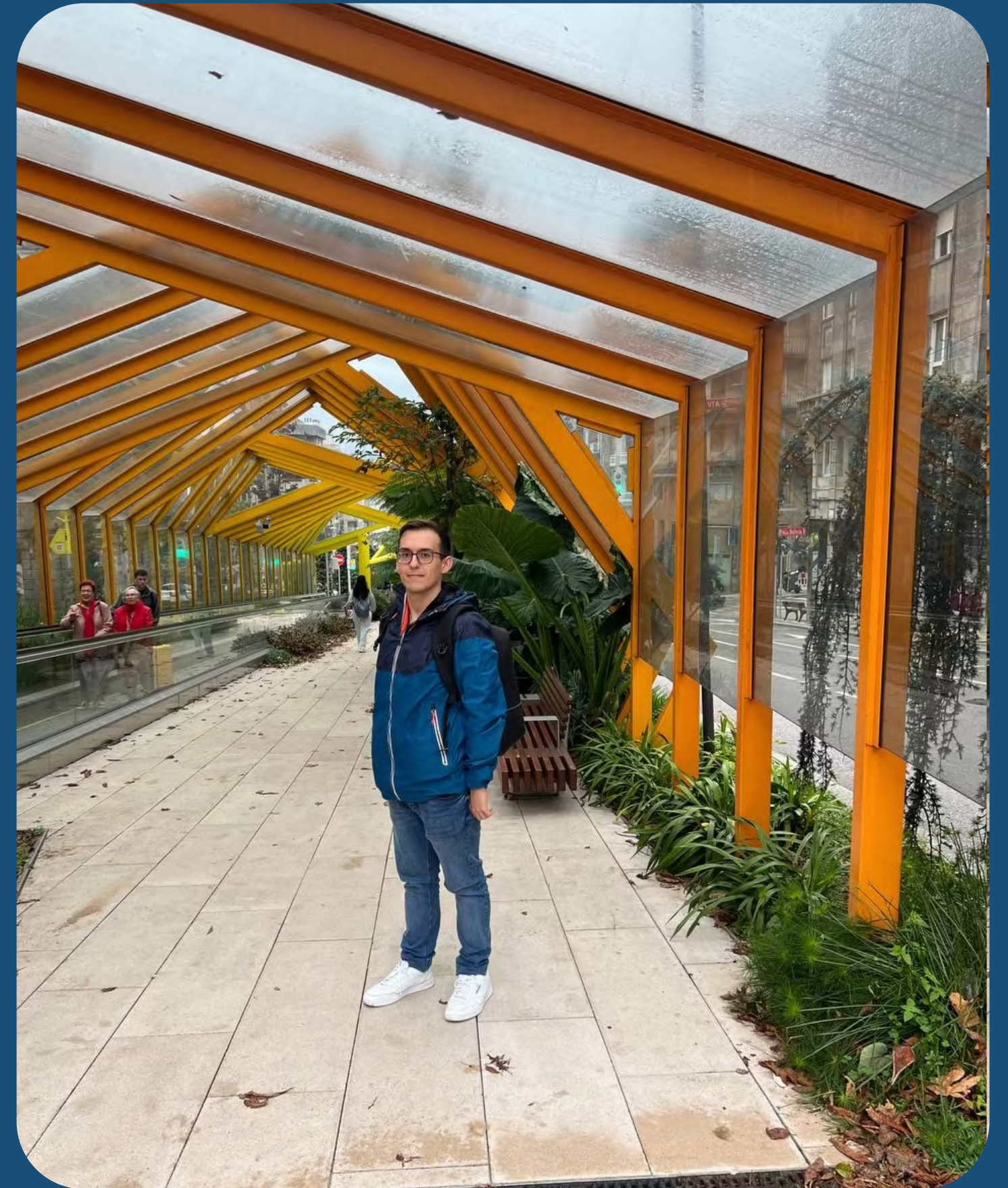
- Mi viaje a Vigo
- ¿Podemos sacar más información?
- La API del Catastro y otros menesteres

03

Conteniendo la API en el Multiverso de La Locura

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.

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...

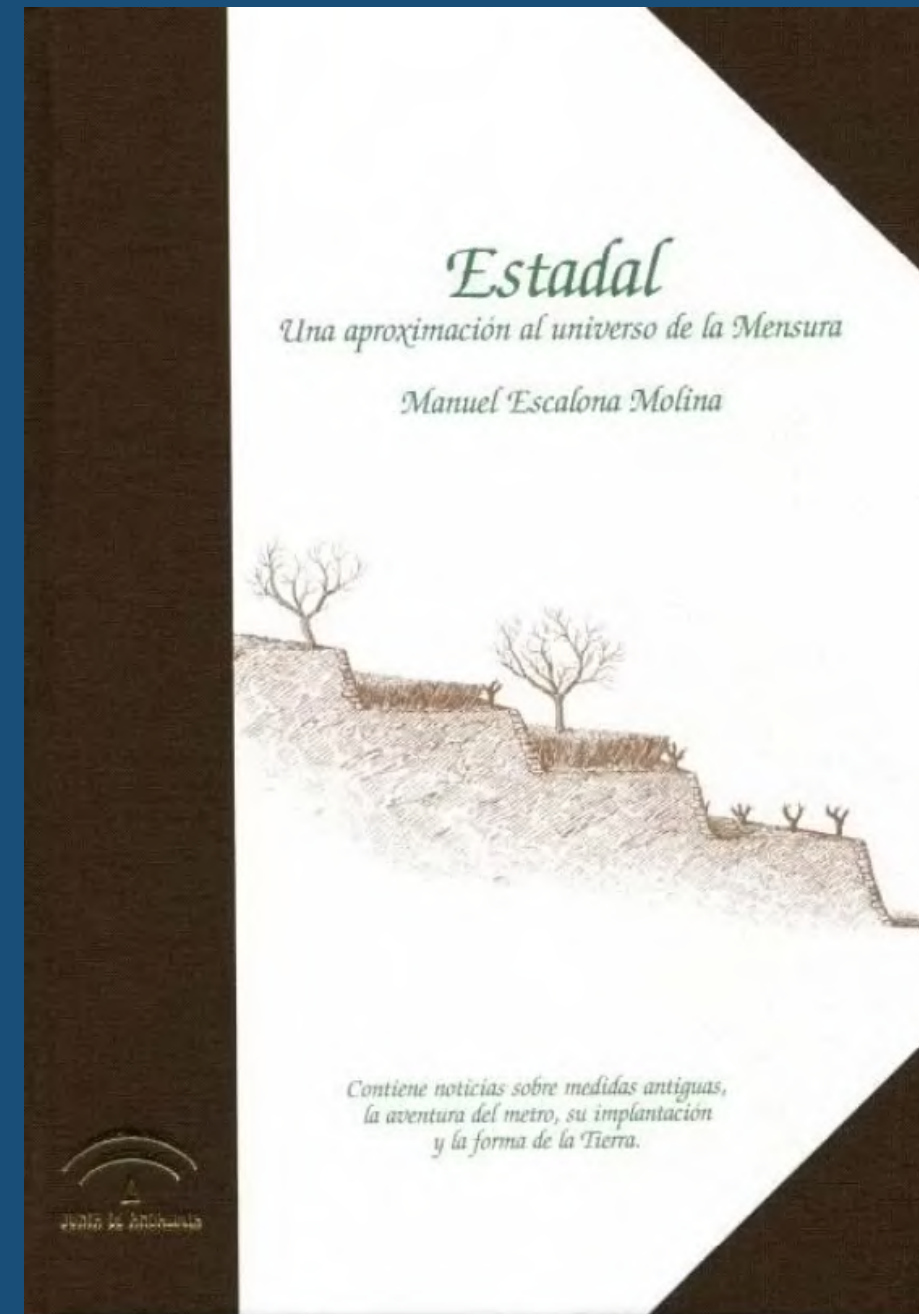
1

Fanega

=

12

Celemines



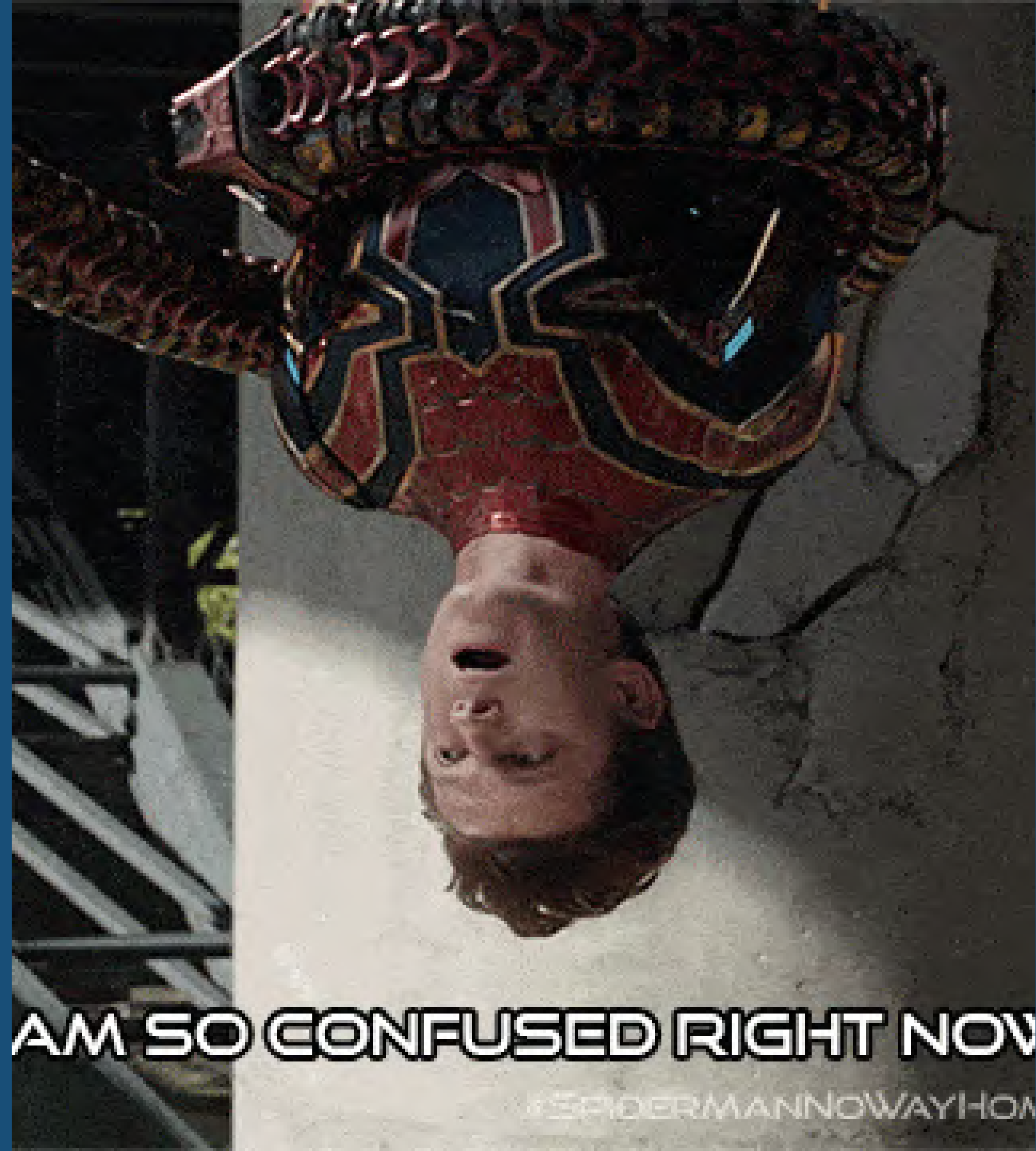
Fanega

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 3ª 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.

- Málaga: Toda la provincia: fanega de 8640 varas y 540 estadales, 6037 m². Vélez: fanega de regadío, 4697 m²; de secano, 6262 m². Antequera: de secano, 6395 m². Cañete la Real: de secano, 6440 m²*. Campillos: de secano, 6127 m².

- Granada: Baza: fanega de huerta, 1397 m²; de vega, 1643 m²; de orillas, 2376 m²; de secano, 4996 m². Huéscar: fanega riego 1ª, 1747 m²; riego 2ª, 2037 m²; riego 3ª, 3494 m²; secano, 5531 m². Guadix: fanega de riego, 2348 m². Loja: fanega de secano, 5637 m². Guadix, Iznállor 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 1ª y 2ª, 1747 m²; riego de 3ª, 2306 m². Galera (Huéscar): de riego de 1ª y 2ª, 1747 m²; de riego de 3ª, 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 1ª, 1526 m²; riego 2ª, 1615 m²; de 3ª, 1712 m²; de 4ª, 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 1ª, 1415 m²; de 2ª, 1747 m²; de 3ª, 2096 m²; secano 1ª, 3144 m²; de 2ª, 4472 m². Zafarroya (Alhama): fanega, 6261 m².

¿Y cómo saco la info?



¿Alguien ha intentado esto antes?

catastro

Filter by

<> Code881k

Repositories1.3k

Issues395

Pull requests121

Discussions5

Users418

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 matchSave

ariseff/overcoming-catastrophic

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

Jupyter Notebook · 302 · Updated on 23 may 2020

Star

oscarfonts/catastro2postgis

Python · 28 · Updated on 3 sept 2013

Star

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

Elastic weight consolidation technique for incremental learning.

neural-networkdeepmindincremental-learningelastic-weight-consolidation

Jupyter Notebook · 139 · Updated on 23 dic 2020

Star

gisce/pycatastro

Python library to access Spanish catastro

pythonapihacktoberfestcatastro

Python · 35 · Updated on 16 ago 2024

Star

geomatico/cidownloader

Catastro Inspire Downloader

Python · 18 · Updated on 17 mar

Star

rOpenSpain/CatastRo

R package to query Sede Electrónica del Catastro (Spain)

cranrmapsgispatial

R · 24 · Updated 13 days ago

Star

¿Coordenadas?

¿Mantenimiento?

Geopackages

¿Resto info?

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 →

How can we improve search? Give feedback

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

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

La saga de Iván vs El Catastro

01

Homecoming: Scrapeando la página

¿A santo de qué?

La página del Catastro

Mi primer script

02

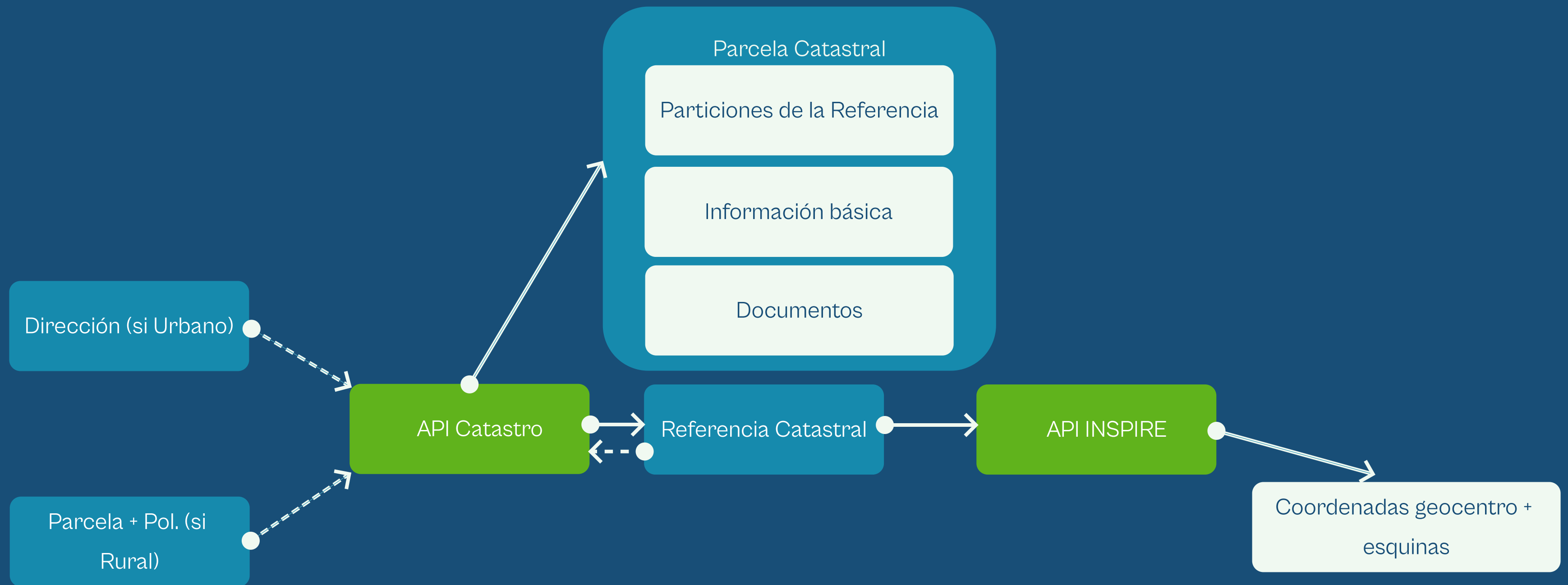
Iván, Far From Home: Replantando el proyecto como paquete

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

03

Conteniendo la API en el Multiverso de La Locura

- Requests, XML y JSON
- Viendo mis tierras con Folium
- Hatchling y cómo empaquetar mi código.



El flujo de información



Consulta y certificación de Bien Inmueble

CARTOGRAFÍA

CONSULTA DESCRIPTIVA Y GRÁFICA

IMPRIMIR DATOS

VISOR 3D

DATOS DESCRIPTIVOS DEL INMUEBLE

Referencia catastral0915103UF7601N0001BD

LocalizaciónAV GUINDOS LOS 48
29004 MALAGA (MÁLAGA)

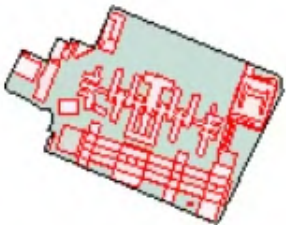
ClaseUrbano

Uso principalEdif. Singular

Superficie construida56.738 m²

Año construcción2007

PARCELA CATASTRAL



Parcela construida sin división horizontal

LocalizaciónAV GUINDOS LOS 48
MALAGA (MÁLAGA)

Superficie gráfica57.376 m²

CONSTRUCCIÓN

Uso principal	Escalera	Planta	Puerta	Superficie m²	Tipo Reforma	Fecha Reforma
ALMACEN	1	00	01	2.150		
PUBLICO	2	00	01	6.588		
ALMACEN	1	01	01	803		
PUBLICO	2	01	01	4.477		
ALMACEN	3	-1	01	2.136		
PUBLICO	3	-1	02	1.389		
APARCAMIENTO	3	-1	03	10.120		
PUBLICO	3	00	01	7.589		
SOPORT. 50%	3	00	02	575		
PUBLICO	3	02	01	8.371		
PUBLICO	3	01	01	7.456		
APARCAMIENTO	1	-1	04	1.035		
ENSEÑANZA	1	00	03	184		
DEPORTIVO	1	00	04	736		
PUBLICO	1	00	05	125		
ALMACEN	1	00	06	86		
ALMACEN	1	00	07	18		
APARCAMIENTO	1	00	04	1.035		
ENSEÑANZA	1	01	02	184		
PUBLICO	1	01	03	125		
APARCAMIENTO	1	01	04	1.035		
ENSEÑANZA	1	02	02	36		
APARCAMIENTO	1	02	03	485		

Un ejemplo: La Térmica

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

[Provincia=Malaga&Municipio=Malaga&Sigla=AV&Calle=Guindos%20Los&Numero=48](#)

▼ consulta_dnpResult:	
▼ control:	
cudnp:	1
cucons:	23
▼ bico:	
▼ bi:	
▼ idbi:	
cn:	"UR"
▼ rc:	
pc1:	"0915103"
pc2:	"UF7601N"
car:	"0001"
cc1:	"B"
cc2:	"D"
▼ dt:	
▼ loine:	
cp:	"29"
cm:	"67"
cmc:	"900"
np:	"MÁLAGA"
nm:	"MALAGA"
▼ locs:	
▼ lous:	
▼ lourb:	
▶ dir:	{ cv: "4590", tv: "AV", nv: "GUINDOS LOS", ... }
▶ loint:	{ es: "T", pt: "0D", pu: "0S" }
dp:	"29004"
dm:	"10"
ldt:	"AV GUINDOS LOS 48 29004 MALAGA (MÁLAGA)"
▼ debi:	
luso:	"RDL 1/2004 8.2a"
sfc:	"56738"
cpt:	"100,000000"
ant:	"2007"

▼ lcons:	
▼ 0:	
lcd:	"ALMACEN"
▼ dt:	
▼ lourb:	
▼ loint:	
es:	"1"
pt:	"00"
pu:	"01"
▼ dfcons:	
stl:	"2150"
▼ dvcons:	
dtip:	"INDUSTRIA NO FABRIL"
▼ 1:	
lcd:	"PUBLICO"
▼ dt:	
▼ lourb:	
▼ loint:	
es:	"2"
pt:	"00"
pu:	"01"
▼ dfcons:	
stl:	"6588"
▼ dvcons:	
dtip:	"ADMINISTRACIÓN-PÚBLICO"
▼ 2:	
lcd:	"ALMACEN"
▼ dt:	
▼ lourb:	
▼ loint:	
es:	"1"
pt:	"01"
pu:	"01"

```
<consulta_dnp>
<control>
<cudnp>NÚMERO DE INMUEBLES DE LOS QUE SE PROPORCIONAN DATOS</cudnp>
<cucons>NÚMERO DE UNIDADES CONSTRUCTIVAS (INCLUYENDO ELEMENTOS COMUNES)</cucons>
<cucul>NÚMERO DE 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: La Térmica

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



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
```

"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_errores(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



Detalle del fragmento de código anterior

```
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 comprobarErrores(response.json()):
31             calles_dict_raw = json.loads(response.content)
32             return [f"{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 []
```

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 O.
- ¿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 -4.446924 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>
```

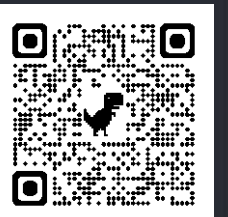
```
1 import xmltodict
2
3 geometry_request = requests.get(f'{URL_BASE_GEOGRAFIA}',
4                                 params={
5                                     'service': 'wfs',
6                                     'version': '2',
7                                     'request': 'getfeature',
8                                     '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(' ')
```

¿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!

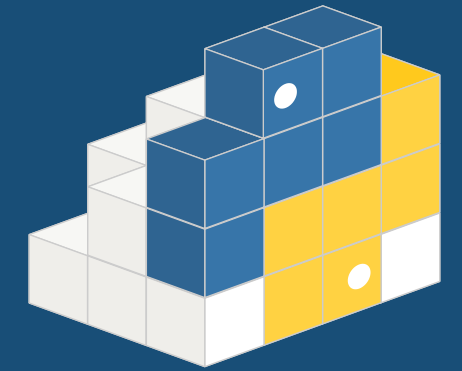
```
1 import folium
2
3 map_geo = folium.Map(location=geocentro, zoom_start=18)
4 coordinates = [] # Formato [[lat, lon], [lat, lon], ...]
5 folium.raster_layers.WmsTileLayer(url = 'https://www.ign.es/wms-inspire/pnoa-ma?SERVICE=WMS&',
6                                   layers = 'OI.OrthoimageCoverage',
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')
```





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

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.



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
```



Entonces...

01

Iniciar el paquete y copiar el código



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

02

Hacer tests



```
1 hatch test
```

03

Construir y subir

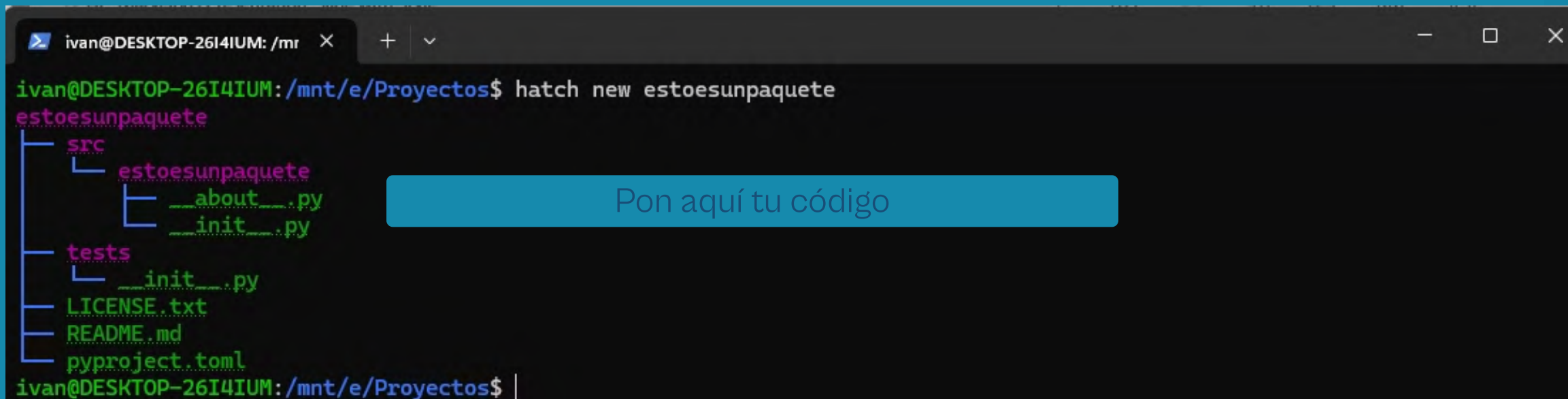


```
1 hatch build
```

Entonces...

01

Iniciar el paquete y copiar el código



```
ivan@DESKTOP-26I4IUM: /mnt X + v
ivan@DESKTOP-26I4IUM:/mnt/e/Proyectos$ hatch new estoessunpaquete
estoessunpaquete
├── 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
```

```
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
```



Ayuda

Patrocinadores

Acceder

[Registrarse](#)

keyring 25.6.0

✓ [Versión más reciente](#)

```
pip install keyring
```

Publicación: 25 dic 2024

Store and access your passwords safely.

Navegación

Descripción de proyecto

 [Histórico de versiones](#)

Archivos de descarga

Verified details

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

Responsables



jaraco

Metainformación

- Responsible: [Jason R. Coombs](#)

Descripción de proyecto

[pypi](#)
[v25.6.0](#)
[python](#)
[3](#)
[tests](#)
[failing](#)
[Ruff](#)
[docs](#)
[passing](#)
[skeleton](#)
[2024](#)
[TII](#)
[lifted!](#)
[chat](#)
[on 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

Try Pitch



[Ayuda](#)[Patrocinadores](#)[Acceder](#)[Registrarse](#)

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.

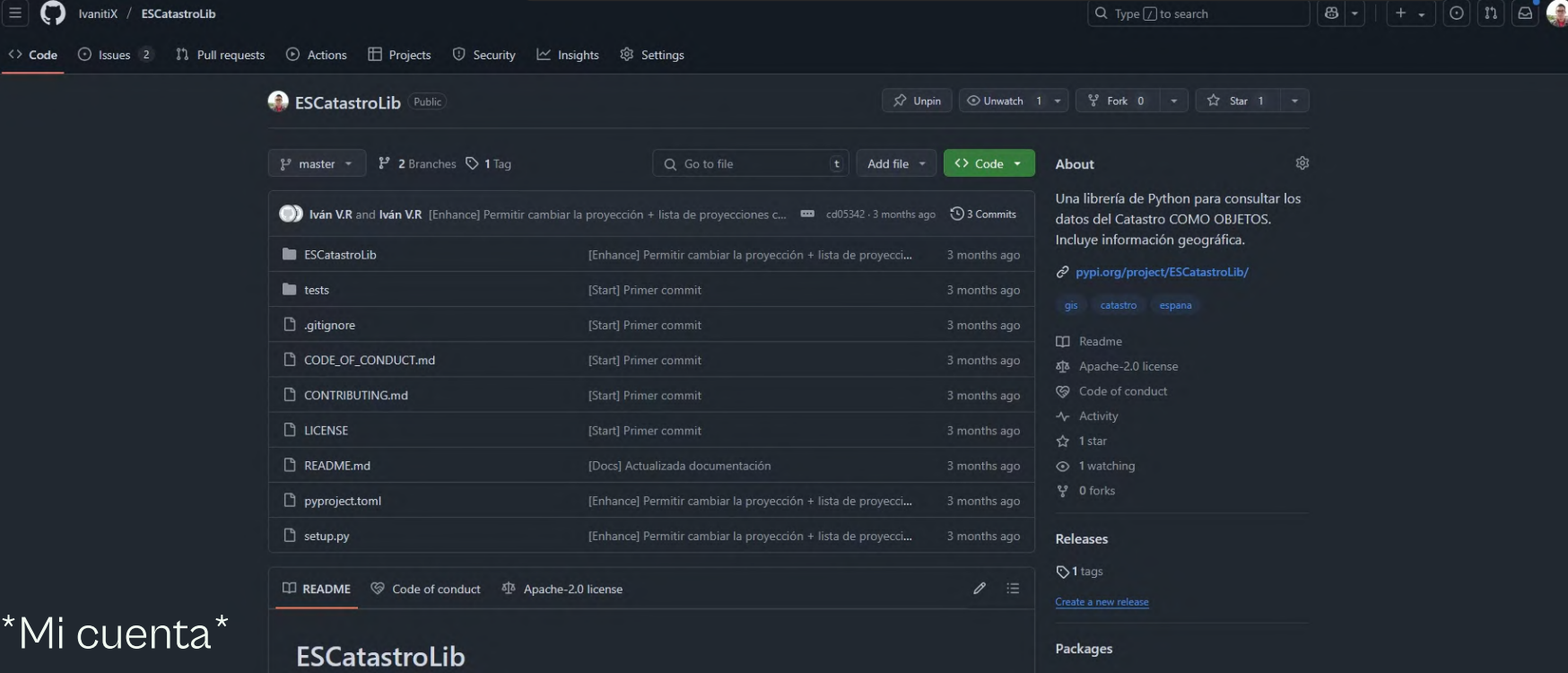
`municipio (str, opcional)`: El nombre del municipio. Si no se proporciona al crear una instancia, se lanzará una excepción. Si se especifica, debe coincidir al menos parcialmente con el nombre de un municipio válido encontrado

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

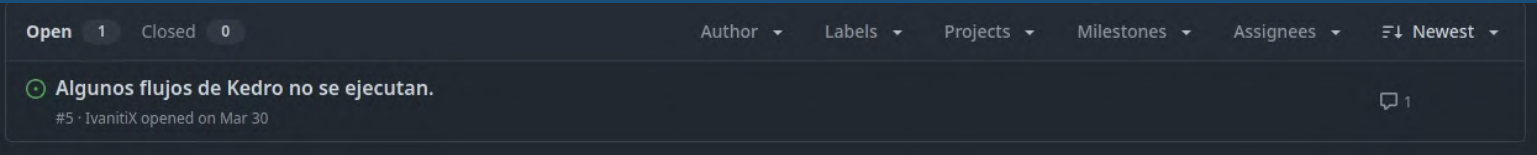
El resultado

El endgame

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



Lo de Juanlu



Público, ¿qué
habéis aprendido?



Preguntas post-charla

1. **¿Existe alguna forma de ver las coordenadas en el Eje Z?** Iván Sánchez (yo no) responde que al ser Catastro una fuente de datos cuyo objetivo es la recaudación de impuestos, no se tiene en cuenta la información de altura.
2. **¿Toda la información que ves en Catastro es pública?** No, hay una parte que solo se puede acceder por Certificado Electrónico o Cl@ve. No hay disponibles endpoints API y Selenium no parece dejarme acceder a esos datos.
3. **¿Qué datos se recogen entonces en la librería?** Ver imagen o <https://github.com/IvanitiX/ESCatastroLib?tab=readme-ov-file#clase-parcelacatastral>
4. **Fe de erratas (gracias a la corrección de Iván Sánchez):** No es geocentro o centro geográfico, ese punto se denomina centroide.

Atributos:

- `rc (str)` : La referencia catastral de la parcela.
- `provincia (str)` : El nombre de la provincia.
- `municipio (str)` : El nombre del municipio.
- `poligono (int)` : El número del polígono. Solo se usa para terrenos rurales.
- `parcela (int)` : El número de la parcela. Solo se usa para terrenos rurales.
- `tipo_via (str)` : El tipo de carretera en la dirección.
- `calle (str)` : El nombre de la calle en la dirección.
- `numero (str)` : El número de la calle en la dirección.
- `url_croquis (str)` : La URL del croquis de la parcela.
- `tipo (str)` : El tipo de terreno (urbano o rural).
- `antiguedad (str)` : La edad de la parcela (solo en lotes urbanos)
- `uso (str)` : El uso de la parcela (solo en lotes urbanos)
- `nombre_paraje (str)` : El nombre del área, solo se usa para terrenos rurales.
- `regiones (list)` : Una lista de regiones en la parcela con una descripción y superficie.
- `geocentro (dict)` : Las coordenadas del centro de la parcela. (Latitud y longitud)
- `geometria (list)` : Una lista de puntos que representan la geometría de la parcela. (Latitud y longitud)

¡Muchas gracias!

(¿SABEIS QUE TENGO PEGATINAS NUEVAS? PIDE ALGUNA)
(AH, ACORDAOS DE LA FOTO)

¿Me estás buscando en redes?

Try Pitch  @IvanInVR(@mastodon.la)

 @IvanitiX

 /ivan-vr





Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)