

# Aplicación de Locales

---

Grupo 24

- Daniel García Lozano
- Marta Pérez Delgado





# ÍNDICE

---

- Objetivo
- Dataset
- Ontología
- Transformación de datos
- Aplicación

# Objetivo

---

- El objetivo es desarrollar una aplicación utilizando datos enlazados.
- En nuestro caso hemos usado datos de locales, sus ubicaciones y horarios para crear una aplicación que



# Dataset

- Nuestro dataset lo hemos sacado del portal de datos abiertos el Ayuntamiento de Madrid:

<https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=66665cde99be2410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default>

id_local	id_distrito	lo_desc_distrito	id_barrio	loc_desc_barrio	l_cod_barrio	lc_desc_seccion	coordenada_x	coordenada_y	id_tipo_acces	desc_tipo_acces	id_situacion	desc_situacion	id_vial	edifici_clase_vial	edifici_desc_vial	edifici_id
10000102	1	CENTRO	106	SOL		123	440208,59	4474240,53		Puerta Calle		Abierto	68300	CALLE	ARENAL	
10000258	1	CENTRO	106	SOL		123	440127,59	4474277,53		Puerta Calle		Abierto	68300	CALLE	ARENAL	
10000258	1	CENTRO	106	SOL		123	440127,59	4474277,53		Puerta Calle		Abierto	68300	CALLE	ARENAL	
10000275	1	CENTRO	101	PALACIO		8	0	0		Agrupado		Abierto	679700	PLAZA	SAN MIGUEL	
10000308	1	CENTRO	102	EMBAJADORES		37	440134,59	4473142,53		Puerta Calle		Abierto	774800	CALLE	VENTORRILL	
10000310	1	CENTRO	102	EMBAJADORES		47	440498,59	4473664,53		Puerta Calle		Abierto	547200	CALLE	OLMO	
10000387	1	CENTRO	106	SOL		119	440410,59	4474298,53		Puerta Calle		Abierto	614900	PLAZA	PUERTA DEL S	
10000413	1	CENTRO	103	CORTES		69	440408,59	4474047,53		Puerta Calle		Abierto	276100	CALLE	ESPOZ Y MINA	
10000422	1	CENTRO	101	PALACIO		18	439405,6	4473780,54		Puerta Calle		Abierto	801300	CALLE	YESEROS	
10000434	1	CENTRO	102	EMBAJADORES		24	439852,59	4473567,53		Puerta Calle		Abierto	743600	CALLE	TOLEDO	
10000434	1	CENTRO	102	EMBAJADORES		24	439852,59	4473567,53		Puerta Calle		Abierto	743600	CALLE	TOLEDO	
10000434	1	CENTRO	102	EMBAJADORES		24	439852,59	4473567,53		Puerta Calle		Abierto	743600	CALLE	TOLEDO	
10000442	1	CENTRO	102	EMBAJADORES		42	440471,59	4473306,53		Puerta Calle		Abierto	754400	CALLE	TRIBULETE	
10000468	1	CENTRO	105	UNIVERSIDAD		103	440066,6	4475481,53		Puerta Calle		Abierto	661700	CALLE	SAN BERNARDI	
10000473	1	CENTRO	104	JUSTICIA		73	441036,59	4475026,52		Puerta Calle		Abierto	676600	CALLE	SAN LUCAS	
10000503	1	CENTRO	106	SOL		123	0	0		Agrupado		Abierto	68300	CALLE	ARENAL	
10000533	1	CENTRO	106	SOL		124	0	0		Puerta Calle		Baja Reunifica	298000	CALLE	FLORA	
10000708	1	CENTRO	106	SOL		119	440492,59	4474153,53		Puerta Calle		Abierto	221000	CALLE	CRUZ	
10000710	1	CENTRO	102	EMBAJADORES		25	439953,59	4473651,53		Puerta Calle		Abierto	679800	CALLE	SAN MILLAN	
10000775	1	CENTRO	106	SOL		117	440341,59	4474606,53		Puerta Calle		Abierto	342950	CALLE	GRAN VIA	
10000807	1	CENTRO	103	CORTES		61	441074,59	4473525,52		Puerta Calle		Abierto	81600	CALLE	ATOCHA	
10000825	1	CENTRO	104	JUSTICIA		74	440685,59	4474665,53		Puerta Calle		Abierto	373600	CALLE	INFANTAS	
10000834	1	CENTRO	105	UNIVERSIDAD		90	440323,6	4475519,53		Puerta Calle		Abierto	659200	CALLE	SAN ANDRES	
10000868	1	CENTRO	105	UNIVERSIDAD		112	439946,6	4474995,53		Puerta Calle		Abierto	58500	CALLE	ANTONIO GRI	
10000877	1	CENTRO	102	EMBAJADORES		33	440062,59	4473350,53		Puerta Calle		Abierto	311600	CALLE	FRAY CEFERIN	



# Ontología

---

- Nuestra ontología consta de las siguientes clases: LocalBuisness, District, Address, Coordenadas.
- Y algunas de las propiedades son: Rótulo, status (si esta abierto o cerrado), número de mesas , horarios de apertura y de cierre, etc

## Transformación de datos

- Hemos añadido columnas con links a datos de Wikidata y hemos eliminado columnas que no tenían relevancia con OpenRefine.
- A partir del csv hemos generado un .yaml y lo hemos transformado en .rml, para con rmlmapper generar el archivo Ntriples con links.



# Aplicación

```
object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end - and  
mirror_obj.select=1  
mirror_obj.select=1  
context.scene.objects.active  
("selected" + str(mirror_obj.name))  
mirror_obj.select=1  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one object")
```

--- OPERATOR CLASSES ---

```
bpy.types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```