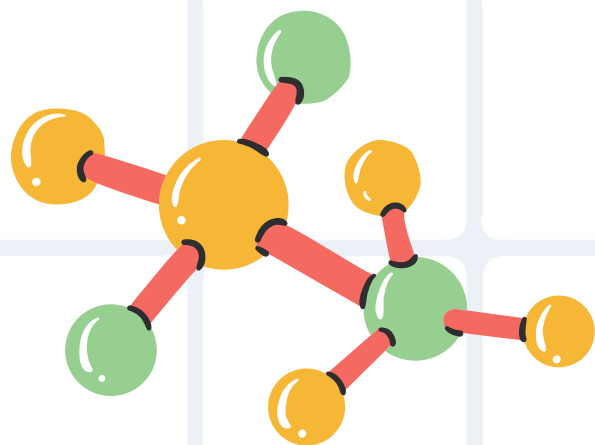
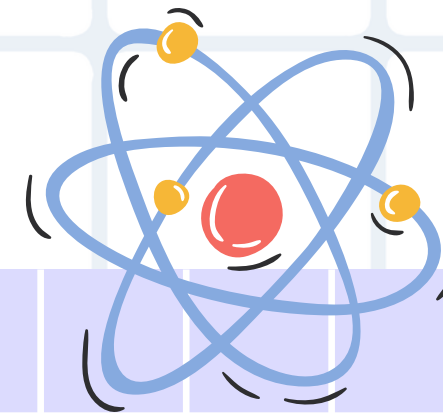
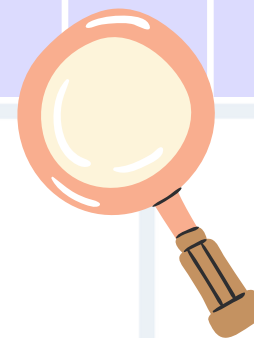


# BICIPARKING



ÁNGEL BARRUZ



# PRODUCT DESCRIPTION



## Base de datos Bici\_Mad - Parking\_Mad

- Parking\_Mad: DataFrame obtenido a partir de las siguientes tablas parkings de la comunidad de Madrid:
  - 1- Parking Municipales para Residentes.
  - 2- Parking Municipales.
  - 3- Parking Disuasorios.
- Bici\_Mad: DataFrame, a partir de una base de datos en MySQL, con las estaciones de bici en la Comunidad de Madrid.

## Dataframe final: Bici\_Mad - Parking\_Mad

El resultado , es la unión de Bici\_Mad y Parking\_Mad, para obtener la estación Bici\_Mad más cercana, respecto al punto de origen (Parking\_Mad).

## Resultado

El usuario, puede obtener el resultado de dos formas:

- 1 - Indicando la opción "**todos**". Se mostrará la tabla completa con todos los registros de las relaciones entre Parking\_Mad (Origen) y Bici\_Mad (Destino).
- 2- Indicando la opción "**uno**". Se introduce el punto de origen deseado para que devuelva la tabla con la relación de un punto específico.

# PRODUCT DESCRIPTION



## OPCIÓN "todos":

	Origen	Destino	Tipo	Distancia
0	Aparcamiento disuasorio Aviación Española	224 - Caramuel	Disuasorio	6779.868246
1	Aparcamiento disuasorio Aviación Española	224 - Caramuel	Publicos Municipales	6779.868246
2	Aparcamiento disuasorio Estadio Metropolitano ...	215 - Gutierre de Cetina	Disuasorio	4728.940915
3	Aparcamiento disuasorio Estadio Metropolitano ...	215 - Gutierre de Cetina	Publicos Municipales	4728.940915
4	Aparcamiento disuasorio Fuente de la Mora	249 - Manuel Caldeiro	Disuasorio	2389.581535

## OPCIÓN "uno":

Elija su parking: Aparcamiento para residentes. Codorniz

	Origen	Destino	Tipo	Distancia
648	Aparcamiento para residentes. Codorniz	224 - Caramuel	Residentes	3960.759407

# PRODUCT DESCRIPTION

## INPUT

```
1  from unittest import result
2  import pandas as pd
3  import argparse
4
5  df_final = pd.read_csv('../project_m1_new/df_final.csv')
6  print('Data loaded...')
7
8  # filter = ['Aparcamiento disuasorio Aviación Española']
9
10 def argument_parser():
11     parser = argparse.ArgumentParser(description= 'Application for arithmetic calculations' )
12     help_message = 'You have two options. Option 1:Registro completo. Option 2:Unico Registro'
13     parser.add_argument('-f', '--function', help=help_message, type=str)
14     args = parser.parse_args()
15     return args
16
17
18
19 if __name__ == '__main__':
20     print(type(argument_parser()))
21     if argument_parser().function == 'todos':
22         result = df_final
23     elif argument_parser().function == 'uno':
24         parking = input('Elija su parking: ')
25         result = df_final[df_final['Origen'] == parking]
26     else:
27         result = 'FATAL ERROR...you need to select the correct method'
28
29 print(f'The result is => {result}')
```

# OUTPUT "todos"

```
(p_01) C:\ironhack\project_m1_new\modules>Python main.py -f todos
Data loaded...
<class 'argparse.Namespace'>
The result is =>      Unnamed: 0
0      0      Aparcamiento disuasorio Aviación Española      224 - Caramuel      Disuasorio      6779.868246
1      1      Aparcamiento disuasorio Aviación Española      224 - Caramuel      Publicos Municipales      6779.868246
2      2      Aparcamiento disuasorio Estadio Metropolitano ...      215 - Gutierre de Cetina      Disuasorio      4728.940915
3      3      Aparcamiento disuasorio Estadio Metropolitano ...      215 - Gutierre de Cetina      Publicos Municipales      4728.940915
4      4      Aparcamiento disuasorio Fuente de la Mora      249 - Manuel Caldeiro      Disuasorio      2389.581535
..      ...      ...      ...      ...      ...
646      646      Aparcamiento para residentes. Vital Aza      215 - Gutierre de Cetina      Publicos Municipales      415.949952
647      647      Aparcamiento para residentes. Vital Aza      215 - Gutierre de Cetina      Residentes      415.949952
648      648      Aparcamiento para residentes. Codorniz      224 - Caramuel      Residentes      3960.759407
649      649      Aparcamiento para residentes. Conde de Casal /...      77 - Conde de Casal      Residentes      103.216465
650      650      Aparcamiento para residentes. Corazón de María I      241 - Camilo José Cela      Residentes      450.181476

[651 rows x 5 columns]
```

# OUTPUT "uno"

```
(p_01) C:\ironhack\project_m1_new\modules>Python main.py -f uno
Data loaded...
<class 'argparse.Namespace'>
Elija su parking: Aparcamiento disuasorio Aviación Española
The result is =>      Unnamed: 0
0      0      Aparcamiento disuasorio Aviación Española      224 - Caramuel      Disuasorio      6779.868246
1      1      Aparcamiento disuasorio Aviación Española      224 - Caramuel      Publicos Municipales      6779.868246
```

# TECHNOLOGY STACK



## Software diseñado:

- Módulo Adcquisition: se guarda la Base de Datos MySQL Bici\_Mad y los json convertidos de la API de los Parkings de la Comunidad de Madrid (\*\* IMPORTANTE: Se sacan las tablas a partir de un CSV porque se tiene que limpiar los datos al dar error en formato json en la línea 322 de la tabla Parking Municipal)
- Módulo Wrangling\_2: el código elaborado en Jupyter Notebook, se pasa a funciones.
- Main: cargamos la tabla final de Parking\_Mad y Bici\_Mad para ejecutar argparse y que el usuario pueda interactuar.
- El lenguaje de programación utilizado es Python, versión 3.9
- Las librerías utilizadas han sido: pandas, requests, json, csv, numpy, sqlalchemy, shapely.geometry, geopandas , argparse, unittest.

# PAIN POINTS

- Conexión a base de datos MySQL -> Estaciones BiciMad

```
1 connection_string = 'mysql://ironhack_user:%Vq=c>G5@173.201.189.217/BiciMAD'  
2 engine = create_engine(connection_string)
```

- Concat de los Parking CM : Municipales, Municipales Residentes y Disuasorios.

	NOMBRE	LATITUD	LONGITUD	Tipo	key
0	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Disuasorio	0
1	Aparcamiento disuasorio Estadio Metropolitano ...	40.434154	-3.598831	Disuasorio	0
2	Aparcamiento disuasorio Fuente de la Mora	40.484443	-3.664503	Disuasorio	0

# PAIN POINTS

- Unión de DataFrames

		NOMBRE_x	LATITUD_x	LONGITUD_x	Tipo	key	NOMBRE_y	LATITUD_y	LONGITUD_y
0	Aparcamiento disuasorio	Aviación Española	40.383353	-3.783593	Disuasorio	0	1a - Puerta del Sol A	40.417214	-3.701834
1	Aparcamiento disuasorio	Aviación Española	40.383353	-3.783593	Disuasorio	0	1b - Puerta del Sol B	40.417313	-3.701603
2	Aparcamiento disuasorio	Aviación Española	40.383353	-3.783593	Disuasorio	0	2 - Miguel Moya	40.420589	-3.705842

- Diferencia de Distancia, calculada en metros.

Unnamed: 0		NOMBRE_x	LATITUD_x	LONGITUD_x	Tipo	key	NOMBRE_y	LATITUD_y	LONGITUD_y	Diferencia
0	0	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Disuasorio	0	1a - Puerta del Sol A	40.417214	-3.701834	9869.005208
1	1	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Disuasorio	0	1b - Puerta del Sol B	40.417313	-3.701603	9897.046533
2	2	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Disuasorio	0	2 - Miguel Moya	40.420589	-3.705842	9613.315289



# PAIN POINTS

- Estación Bici\_Mad más cercana respecto a Parking\_Mad

	Unnamed: 0	NOMBRE_x	LATITUD_x	LONGITUD_x	Tipo	key	NOMBRE_y	LATITUD_y	LONGITUD_y	Diferencia
226	226	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Disuasorio	0	224 - Caramuel	40.409136	-3.728534	6779.868246
490	87082	Aparcamiento disuasorio Aviación Española	40.383353	-3.783593	Publicos Municipales	0	224 - Caramuel	40.409136	-3.728534	6779.868246
745	481	Aparcamiento disuasorio Estadio Metropolitano ...	40.434154	-3.598831	Disuasorio	0	215 - Gutierre de Cetina	40.429706	-3.640994	4728.940915

- Tabla final renombrada

```
1 df_final.rename(columns = {'NOMBRE_x': 'Origen', 'NOMBRE_y': 'Destino', 'Diferencia': 'Distancia'}, inplace=True)
```

# DEMO



# CODE SAMPLE

- Cálculo de Distancias

```
: 1 # Calculamos el minimo de distancia
2 df_final = pm_bm.groupby('NOMBRE_x')['Diferencia'].min()
3 df_final.head(3)
```

```
: 1 # Hacemos Merge para incluir el minimo
2 df_final = pm_bm.merge(df_final, on='NOMBRE_x', suffixes=('', '_'))
3 df_final.head(3)
```

```
: 1 # Creamos una nueva columna, renombrando la del mínimo y eliminando la del cálculo de distancias.
2 df_final = df_final[df_final['Diferencia'] == df_final['Diferencia_']].drop('Diferencia_', axis=1)
3 df_final.head(3)
```

```
: 1 df_final = df_final[['NOMBRE_x', 'NOMBRE_y', 'Tipo', 'Diferencia']]
2 df_final.shape
```

Activar Window:



**GRACIAS**