

# ANEXO C

## Fuentes de datos pre procesadas, diagnostico y análisis

### Datos sobre delitos de hurto en Bogotá

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [66]: df_upz = pd.read_csv("Datasets/DelitosBogotaUPZ.csv")
df_fip_homi = pd.read_csv("Datasets/FIP_homicidios_sept_2018_Bogota.csv", delimiter=";")
df_siedco_bog = pd.read_csv("Datasets/SIEDCO_BOG_2019_2021.csv", delimiter=";")
df_siedco_bog_from_2016 = pd.read_csv("Datasets/SIEDCO_delitos_2016_2020.csv", delimiter=";", low_memory=False)
df_sca = pd.read_csv("Datasets/Sectores_catastrales_suba.csv")
df_barrios_sca_suba = pd.read_csv("Datasets/Barrio_sca_suba.csv")
df_ind_noc_upz = pd.read_csv("Datasets/Ind_Noc_UPZ.csv")
df_upz_sca = pd.read_csv("Datasets/UPZ_SCA.csv", delimiter=";")
```

## 1. Exploración de datos

### 1.1 Indicadores y datos estadísticos por UPZ

```
In [3]: df_upz.head()
```

	ID_UPZ	Codigo_UPZ	Nombre_UPZ	Mes	Homicidios_2018	Homicidios_2019	Homicidios_2020	Variacion_Homicidios_Porc	Homicidios_Bogota_2019	Lesiones_Personales_2019	...	Lesiones_Personales_Bogota_2019
0	20	UPZ68	El Tesoro	Ene-Feb	2	3	9	-66.67	150	31	...	2733
1	46	UPZ57	Gran Yomasa	Ene-Feb	8	2	3	0.00	150	87	...	2733
2	98	UPZ78	Tintal Norte	Ene-Feb	0	0	0	100.00	150	9	...	2733
3	115	UPZ115	Capellania	Ene-Feb	0	0	0	0.00	150	12	...	2733
4	88	UPZ116	Alamos	Ene-Feb	0	0	0	0.00	150	12	...	2733

5 rows x 23 columns

### 1.2 Homicidios en Bogotá 2010 - 2018

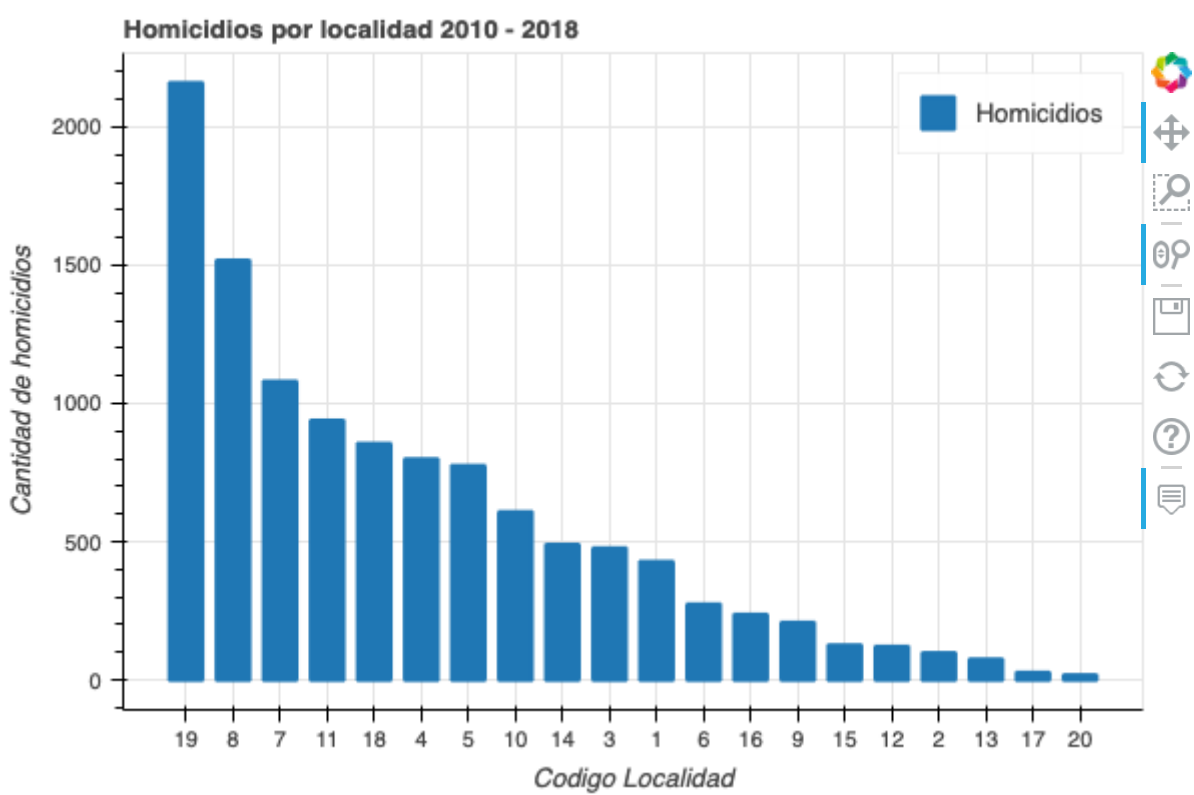
```
In [4]: df_fip_homi['CodigoLocalidad'] = df_fip_homi['barrio'].str.split('E-').str[1]
df_fip_homi.loc[df_fip_homi['CodigoLocalidad'].isnull(), 'CodigoLocalidad'] = df_fip_homi.loc[df_fip_homi['CodigoLocalidad'].isnull(), 'barrio'].str.split(' ').str[-1]
df_fip_homi = df_fip_homi[~df_fip_homi['CodigoLocalidad'].isin(['TERMINAL'])]
df_fip_homi.head()
```

	year	id_mpio	municipio	id_dpto	departamento	capital	ruralidad	iica	zomac	pdet	...	movil_agresor	movil_victima	edad	sexo	clase_empleado	escolaridad	profesion	estado_civil	pais_nar
0	2010	11001	BOGOTÁ D.C.	11	CUNDINAMARCA	1	Ciudades y aglomeraciones	Bajo	0	0	...	A PIE	A PIE	30	MASCULINO	EMPLEADO PARTICULAR	SECUNDARIA	NO REPORTADO	SOLTERO	COLOMB
1	2010	11001	BOGOTÁ D.C.	11	CUNDINAMARCA	1	Ciudades y aglomeraciones	Bajo	0	0	...	A PIE	A PIE	51	MASCULINO	EMPLEADO PARTICULAR	SECUNDARIA	NO REPORTADO	SOLTERO	COLOMB
2	2010	11001	BOGOTÁ D.C.	11	CUNDINAMARCA	1	Ciudades y aglomeraciones	Bajo	0	0	...	A PIE	A PIE	41	MASCULINO	EMPLEADO PARTICULAR	SECUNDARIA	NO REPORTADO	SOLTERO	ECUADC
3	2010	11001	BOGOTÁ D.C.	11	CUNDINAMARCA	1	Ciudades y aglomeraciones	Bajo	0	0	...	A PIE	A PIE	16	MASCULINO	ESTUDIANTE	SECUNDARIA	NO REPORTADO	SOLTERO	COLOMB
4	2010	11001	BOGOTÁ D.C.	11	CUNDINAMARCA	1	Ciudades y aglomeraciones	Bajo	0	0	...	A PIE	A PIE	24	FEMENINO	INDEPENDIENTE	SECUNDARIA	NO REPORTADO	SOLTERO	COLOMB

5 rows x 35 columns

```
In [5]: import pandas_bokeh
pandas_bokeh.output_notebook()
df_fip_homi['CodigoLocalidad'].value_counts()._set_name('Homicidios').plot_bokeh(
    kind='bar',
    xlabel='Codigo Localidad',
    ylabel='Cantidad de homicidios',
    title='Homicidios por localidad 2010 - 2018'
)
```

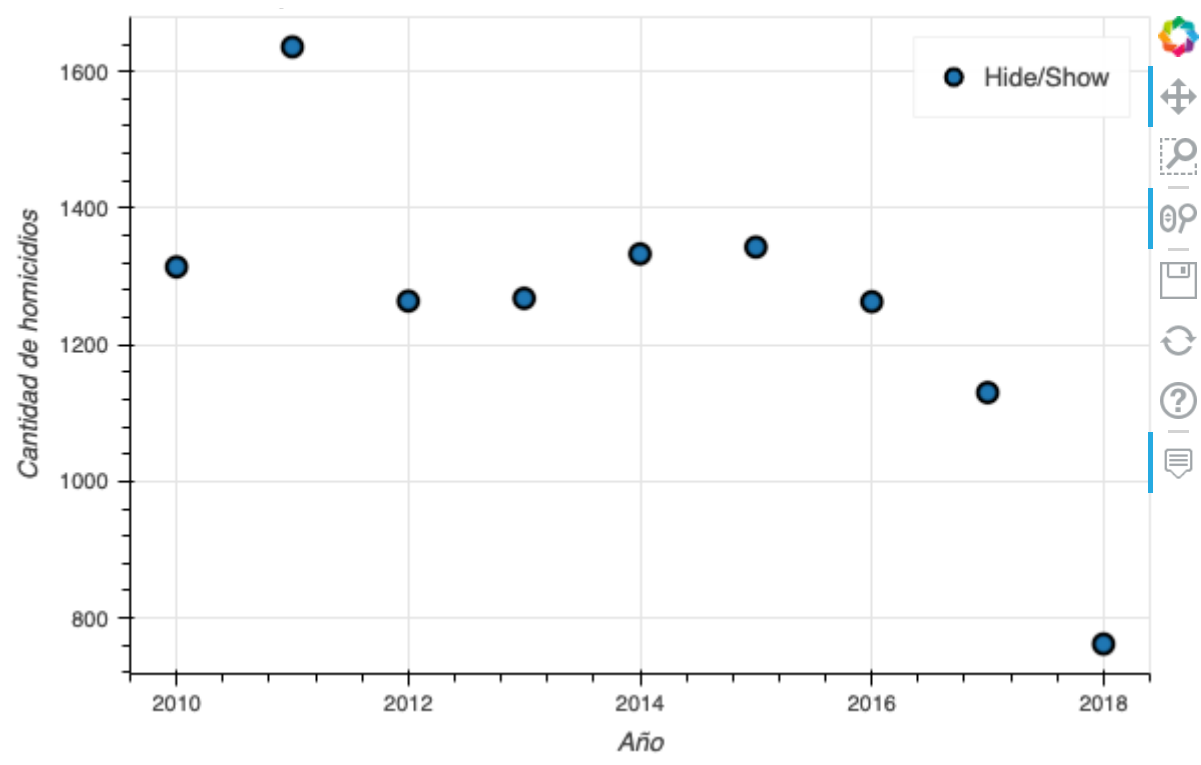
Loading BokehJS ...



Out[5]: Figure(id = '1039', ...)

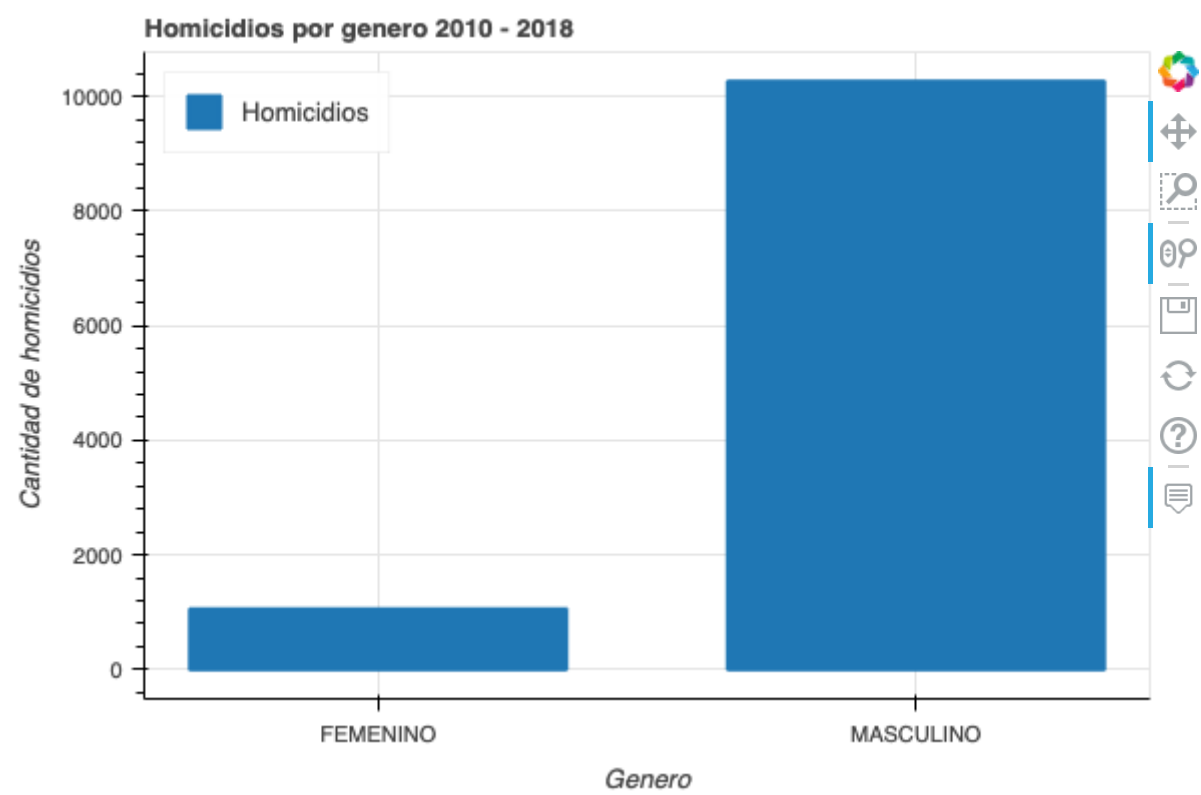
```
In [6]: df_fip_homi['year'].value_counts()._set_name('Homicidios').plot_bokeh(
    kind='scatter',
    xlabel='Año',
    ylabel='Cantidad de homicidios',
    title='Homicidios por año 2010 - 2018'
)
```

Homicidios por año 2010 - 2018



Out[6]: Figure(id = '1146', ...)

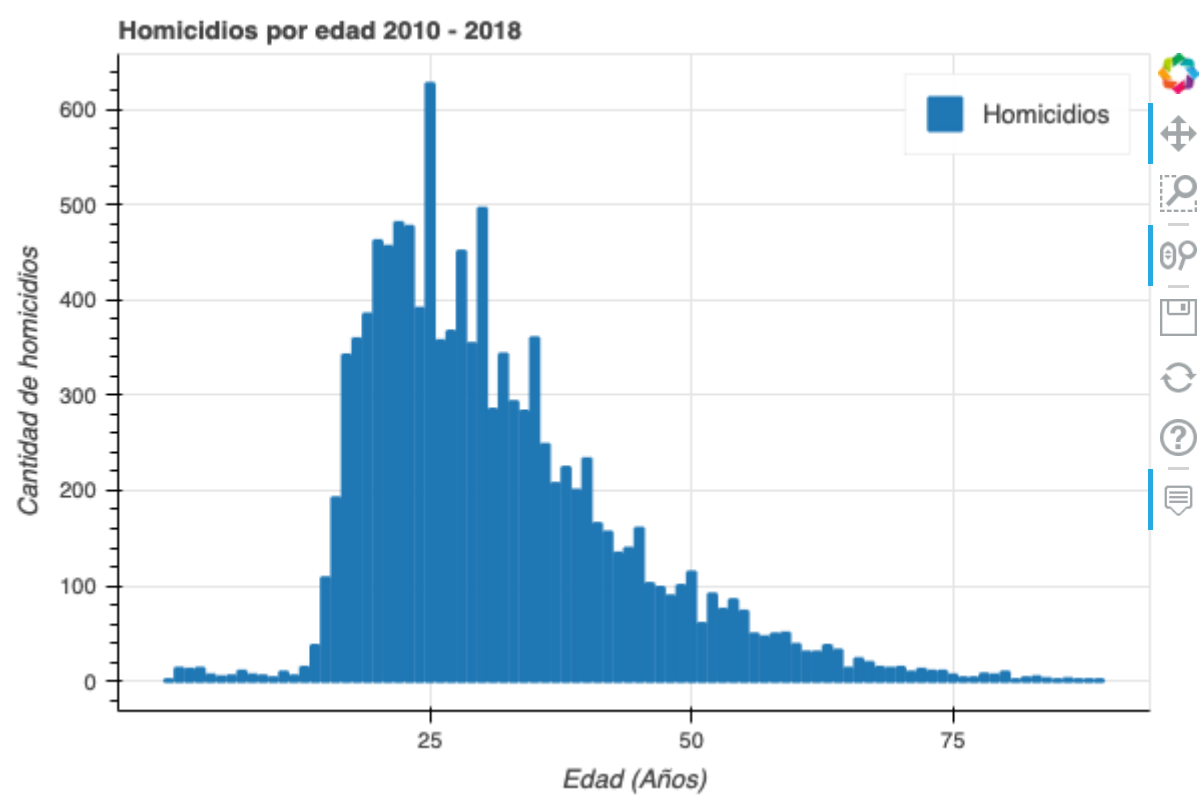
```
In [7]: df_fip_homi['sexo'].value_counts().sort_index()._set_name('Homicidios').plot_bokeh(
    kind='bar',
    xlabel='Genero',
    ylabel='Cantidad de homicidios',
    title='Homicidios por genero 2010 - 2018',
    legend='top_left'
)
```



Out[7]: Figure(id = '1306', ...)

```
In [8]: pandas_bokeh.output_notebook()
df_fip_homi['edad'].astype('int').value_counts().sort_index()._set_name('Homicidios').plot_bokeh(
    kind='bar',
    xlabel='Edad (Años)',
    xticks=[25, 50, 75, 100],
    ylabel='Cantidad de homicidios',
    title='Homicidios por edad 2010 - 2018'
)
```

Loading BokehJS ...

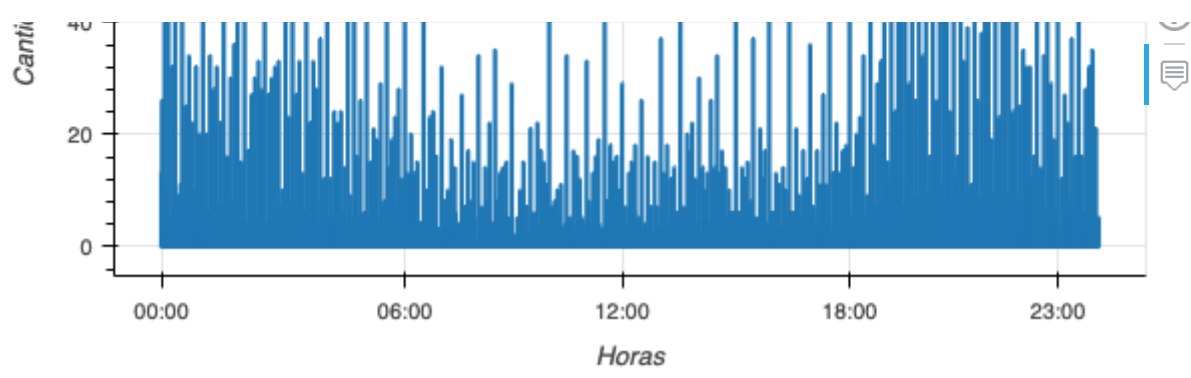


Out[8]: Figure(id = '1468', ...)

```
In [9]: from datetime import datetime
pandas_bokeh.output_notebook()
pd.to_datetime(df_fip_homi['hora'], format='%H:%M:%S').dt.time.apply(lambda x: x.strftime('%H:%M')).value_counts().sort_index()._set_name('Homicidios').plot_bokeh(
    kind='bar',
    xticks=[0, 351, 664, 992, 1292],
    xlabel='Horas',
    ylabel='Cantidad de homicidios',
    title='Homicidios por horas 2010 - 2018',
    legend='top_center'
)
```

BokehJS 2.3.2 successfully loaded.





Out[9]: Figure(id = '1612', ...)

### 1.3 Delitos en Bogota de 2016 - 2020

In [10]: `df_siedco_bog_from_2016.head()`

	AÑO	FECHA	DEPARTAMENTO	MUNICIPIO	BARRIO	DELITO	Cantidad
0	2016	19/12/16	CUNDINAMARCA	BOGOTÁ D.C. (CT)	ABRAHAM LINCON E-6	ARTÍCULO 103. HOMICIDIO	1
1	2016	5/01/16	CUNDINAMARCA	BOGOTÁ D.C. (CT)	ABRAHAM LINCON E-6	ARTÍCULO 109. HOMICIDIO CULPOSO ( EN ACCIDENTE...	1
2	2016	31/05/16	CUNDINAMARCA	BOGOTÁ D.C. (CT)	ABRAHAM LINCON E-6	ARTÍCULO 109. HOMICIDIO CULPOSO ( EN ACCIDENTE...	1
3	2016	8/08/16	CUNDINAMARCA	BOGOTÁ D.C. (CT)	ABRAHAM LINCON E-6	ARTÍCULO 109. HOMICIDIO CULPOSO ( EN ACCIDENTE...	1
4	2016	4/09/16	CUNDINAMARCA	BOGOTÁ D.C. (CT)	ABRAHAM LINCON E-6	ARTÍCULO 109. HOMICIDIO CULPOSO ( EN ACCIDENTE...	1

Delitos de todo tipo por localidad desde el 2016, para identificar las localidades con más delitos

In [11]: `df_siedco_bog_from_2016['CodigoLocalidad'] = df_siedco_bog_from_2016['BARRIO'].str.split('E-').str[1]  
df_siedco_bog_from_2016.loc[df_siedco_bog_from_2016['DELITO'].str.contains('HURTO') , 'CodigoLocalidad'].value_counts()`

Out[11]:

11	53852
8	51992
10	46513
1	37535
2	27249
9	25502
7	23733
13	23233
16	22257
19	19027
3	18997
12	18730
18	17614
4	15170
14	14727
5	11146
6	10114
15	9847
17	4492
20	7
1 BORRAR	3
CHESELINE	1

Name: CodigoLocalidad, dtype: int64

### Dato importante

La localidad 11 de Suba es la localidad con más delitos de hurto desde el año 2016 superando a kenedy, que son las 2 localidades con más delitos de todo tipo

In [12]: `df_plt_siedco = df_siedco_bog_from_2016.loc[(df_siedco_bog_from_2016['DELITO'].str.contains('HURTO')) &  
(~df_siedco_bog_from_2016['CodigoLocalidad'].isin(['1 BORRAR', 'CHESELINE ']))]  
df_plt_siedco['CodigoLocalidad'].value_counts()._set_name('Delitos de hurto')`

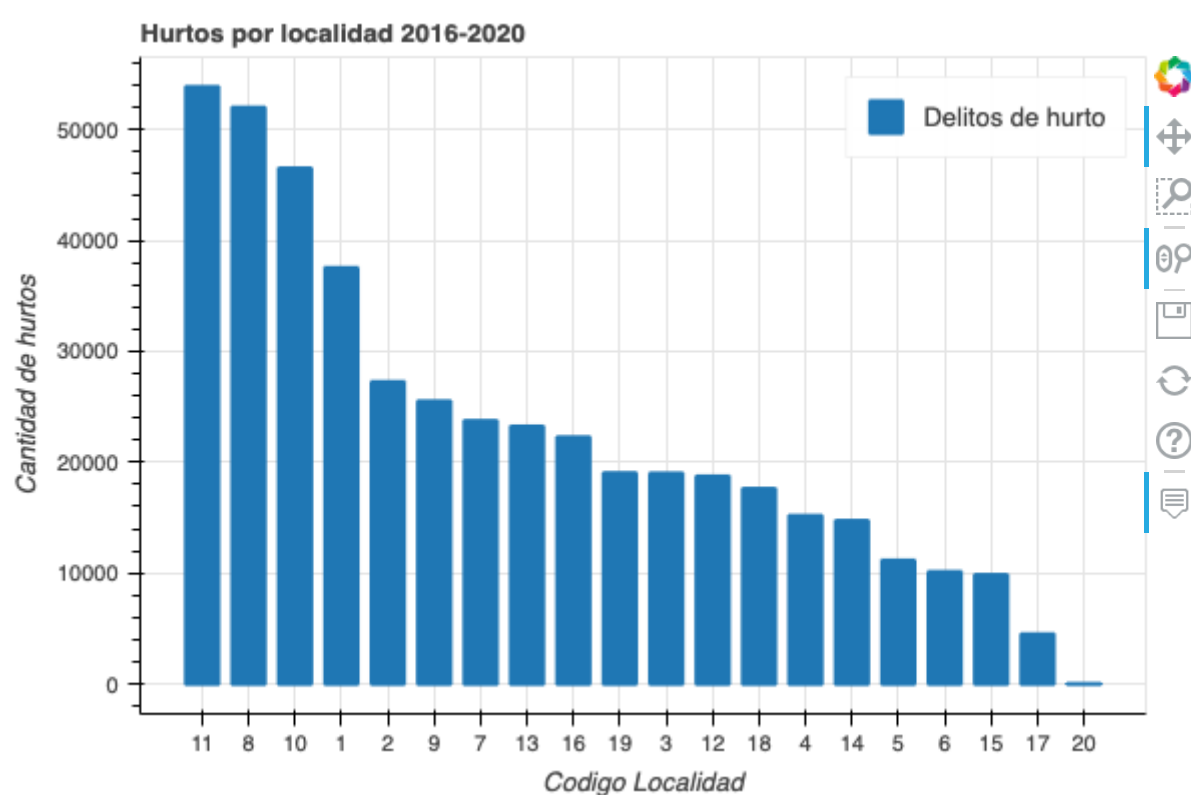
Out[12]:

11	53852
8	51992
10	46513
1	37535
2	27249
9	25502
7	23733
13	23233
16	22257
19	19027
3	18997
12	18730
18	17614
4	15170
14	14727
5	11146
6	10114
15	9847
17	4492
20	7

Name: Delitos de hurto, dtype: int64

In [13]: `import pandas_bokeh  
pandas_bokeh.output_notebook()  
df_plt_siedco['CodigoLocalidad'].value_counts()._set_name('Delitos de hurto').plot_bokeh(  
kind='bar',  
xlabel='Codigo Localidad',  
ylabel='Cantidad de hurtos',  
title='Hurtos por localidad 2016-2020'  
)`

BokehJS 2.3.2 successfully loaded.



Out[13]: Figure(id = '1756', ...)

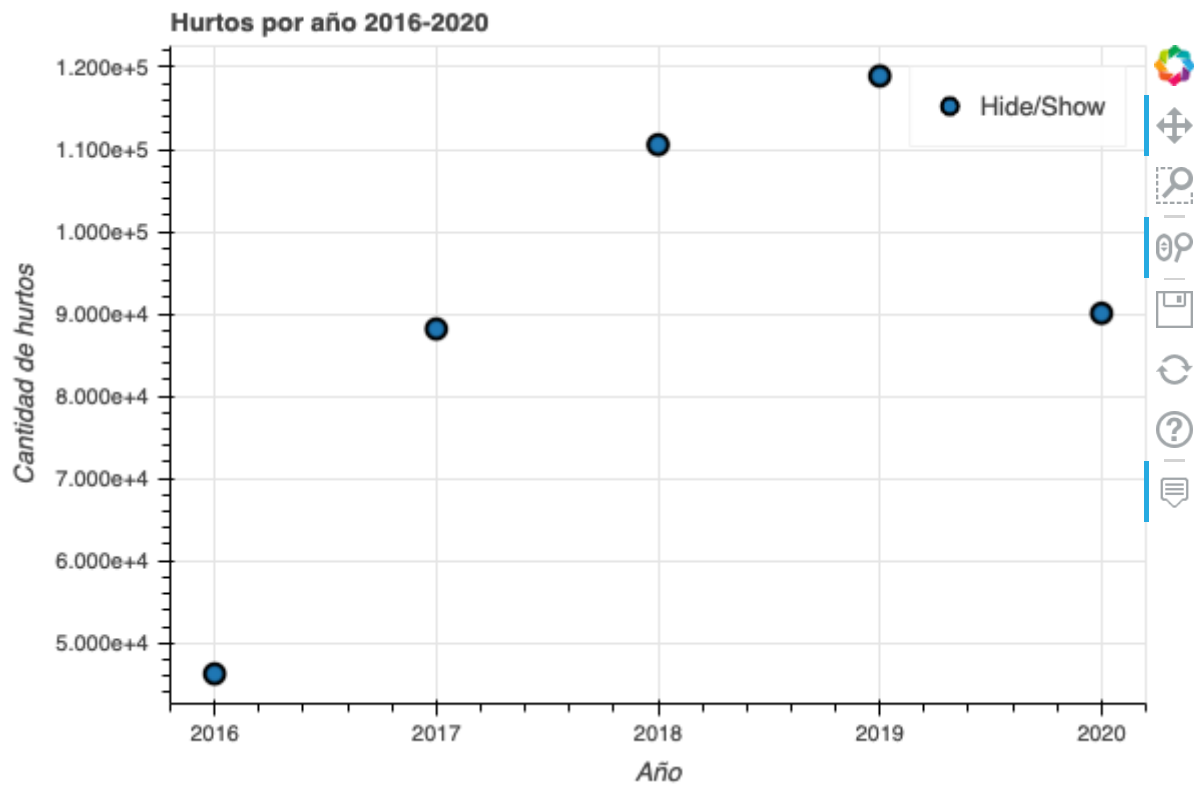
In [14]: `import pandas_bokeh`

```

pandas_bokeh.output_notebook()
df_plt_siedco['AÑO'].value_counts()._set_name('Delitos de hurto').plot_bokeh(
    kind='scatter',
    xlabel='Año',
    ylabel='Cantidad de hurtos',
    title='Hurtos por año 2016-2020'
)

```

BokehJS 2.3.2 successfully loaded.



Out[14]: Figure(id = '1864', ...)

## 1.4 Delitos de hurto en Bogotá 2019 - Actualidad (Marzo 2021)

In [70]: df\_siedco\_bog.dtypes

```

Out[70]: FECHA      object
        HORA      object
        MUNICIPIO object
        BARRIOS   object
        ZONA      object
        CLASE SITIO object
        ARMAS MEDIOS object
        GENERO    object
        DELITOS   object
        CANTIDAD  int64
        dtype: object

```

```

In [73]: df_siedco_bog['CODIGO_LOCALIDAD'] = df_siedco_bog['BARRIOS'].str.split('E-').str[1]
df_siedco_bog.loc[:, 'FECHA'] = pd.to_datetime(df_siedco_bog['FECHA'], format='%d/%m/%y')
df_siedco_bog['AÑO'] = df_siedco_bog['FECHA'].dt.year
df_siedco_bog['MES'] = df_siedco_bog['FECHA'].dt.month
df_siedco_bog['DIA_SEMANA'] = df_siedco_bog['FECHA'].dt.dayofweek
df_siedco_bog['DIA_MES'] = df_siedco_bog['FECHA'].dt.day
df_siedco_bog.loc[:, 'HORA'] = pd.to_datetime(df_siedco_bog['HORA'], format='%H:%M').dt.hour
df_siedco_bog.head()

```

	FECHA	HORA	MUNICIPIO	BARRIOS	ZONA	CLASE SITIO	ARMAS MEDIOS	GENERO	DELITOS	CANTIDAD	CODIGO_LOCALIDAD	AÑO	MES	DIA_SEMANA	DIA_MES
0	2021-03-20	0	BOGOTÁ D.C. (CT)	BELLA SUIZA E-1	URBANA	VIAS PUBLICAS	SIN EMPLEO DE ARMAS	FEMENINO	DELITOS SEXUALES	1	1	2021	3	5	20
1	2021-03-20	0	BOGOTÁ D.C. (CT)	BELLA SUIZA E-1	URBANA	VIAS PUBLICAS	SIN EMPLEO DE ARMAS	FEMENINO	VIOLENCIA INTRAFAMILIAR	2	1	2021	3	5	20
2	2021-03-20	0	BOGOTÁ D.C. (CT)	BELLA SUIZA E-1	URBANA	VIAS PUBLICAS	SIN EMPLEO DE ARMAS	MASCULINO	DELITOS SEXUALES	1	1	2021	3	5	20
3	2021-03-20	0	BOGOTÁ D.C. (CT)	PENSILVANIA E-16	URBANA	RODANDO SOBRE VIA - VIA PUBLICA	ARMA BLANCA / CORTOPUNZANTE	MASCULINO	HURTO A PERSONAS	1	16	2021	3	5	20
4	2021-03-20	0	BOGOTÁ D.C. (CT)	PENSILVANIA E-16	URBANA	RODANDO SOBRE VIA - VIA PUBLICA	ARMA BLANCA / CORTOPUNZANTE	MASCULINO	HURTO A CELULARES (CANTIDAD)	1	16	2021	3	5	20

Función para obtener el intervalo asociado a la hora

```

In [17]: def get_hour_interval(hour):
        if hour < 12:
            return "MAÑANA"
        elif hour > 12 and hour < 13:
            return "MEDIO DIA"
        elif hour > 13 and hour < 18:
            return "TARDE"
        else:
            return "NOCHE"

```

Función para obtener el intervalo asociado al día de la semana

```

In [18]: def get_day_of_week_interval(day):
        if day > 3:
            return "FIN DE SEMANA"
        else:
            return "ENTRE SEMANA"

```

Función para obtener el intervalo asociado al día del mes

```

In [19]: def get_day_of_month_interval(day):
        if (day > 13 and day < 19) or (day > 27 and day < 3):
            return "QUINCENA"
        else:
            return "NO QUINCENA"

```

Función para obtener el intervalo asociado al mes del año

```

In [20]: def get_month_interval(month):
        if month > 10:
            return "FIN DE AÑO"
        elif month < 3:
            return "INICIO DE AÑO"
        elif month > 5 and month < 8:
            return "PRIMA DE MITAD DE AÑO"
        else:
            return "MES COMUN"

```

```

In [21]: df_siedco_bog['HORA_INT'] = df_siedco_bog['HORA'].apply(lambda x: get_hour_interval(x))
df_siedco_bog['DIA_SEMANA_INT'] = df_siedco_bog['DIA_SEMANA'].apply(lambda x: get_day_of_week_interval(x))
df_siedco_bog['DIA_MES_INT'] = df_siedco_bog['DIA_MES'].apply(lambda x: get_day_of_month_interval(x))
df_siedco_bog['MES_INT'] = df_siedco_bog['MES'].apply(lambda x: get_month_interval(x))

```

### 1.4.1 Filtro por hurtos en la localidad de suba en el año 2020

```
In [38]: df_siedco_suba_2020 = df_siedco_bog[(df_siedco_bog['AÑO']==2020) & (df_siedco_bog['CODIGO_LOCALIDAD']=='11') & (df_siedco_bog['DELITOS'].str.contains('HURTO'))]
df_siedco_suba_2020 = df_siedco_suba_2020.loc[-df_siedco_suba_2020['DELITOS'].isin(['HURTO A BICICLETAS (CANTIDAD)', 'HURTO A CELULARES (CANTIDAD)']),:]
df_siedco_suba_2020.head()
```

```
Out[38]:
```

	FECHA	HORA	MUNICIPIO	BARRIOS	ZONA	CLASE SITIO	ARMAS MEDIOS	GENERO	DELITOS	CANTIDAD	CODIGO_LOCALIDAD	AÑO	MES	DIA_SEMANA	DIA_MES	HORA_INT	DIA_SEMANA_INT	DIA_MES_INT
57370	2020-12-31	0	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	GARAJE	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	11	2020	12	3	31	MAÑANA	ENTRE SEMANA	DIA COMPLETO
57392	2020-12-31	1	BOGOTÁ D.C. (CT)	LA BRITALIA E-11	URBANA	CASAS DE HABITACION	SIN EMPLEO DE ARMAS	FEMENINO	HURTO A RESIDENCIAS	1	11	2020	12	3	31	MAÑANA	ENTRE SEMANA	DIA COMPLETO
57402	2020-12-31	2	BOGOTÁ D.C. (CT)	BOSQUE DE LA COLINA E-11	URBANA	CALLEJÓN	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	11	2020	12	3	31	MAÑANA	ENTRE SEMANA	DIA COMPLETO
57404	2020-12-31	2	BOGOTÁ D.C. (CT)	BOSQUE DE LA COLINA E-11	URBANA	CALLEJÓN	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A CELULARES (CASOS)	1	11	2020	12	3	31	MAÑANA	ENTRE SEMANA	DIA COMPLETO
57433	2020-12-31	3	BOGOTÁ D.C. (CT)	SAN ANDRES AFIDRO E-11	URBANA	VEHICULO TAXI	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	11	2020	12	3	31	MAÑANA	ENTRE SEMANA	DIA COMPLETO

```
In [61]: len(df_siedco_suba_2020)
```

Out[61]: 18123

```
In [65]: df_siedco_suba_2020.dtypes
```

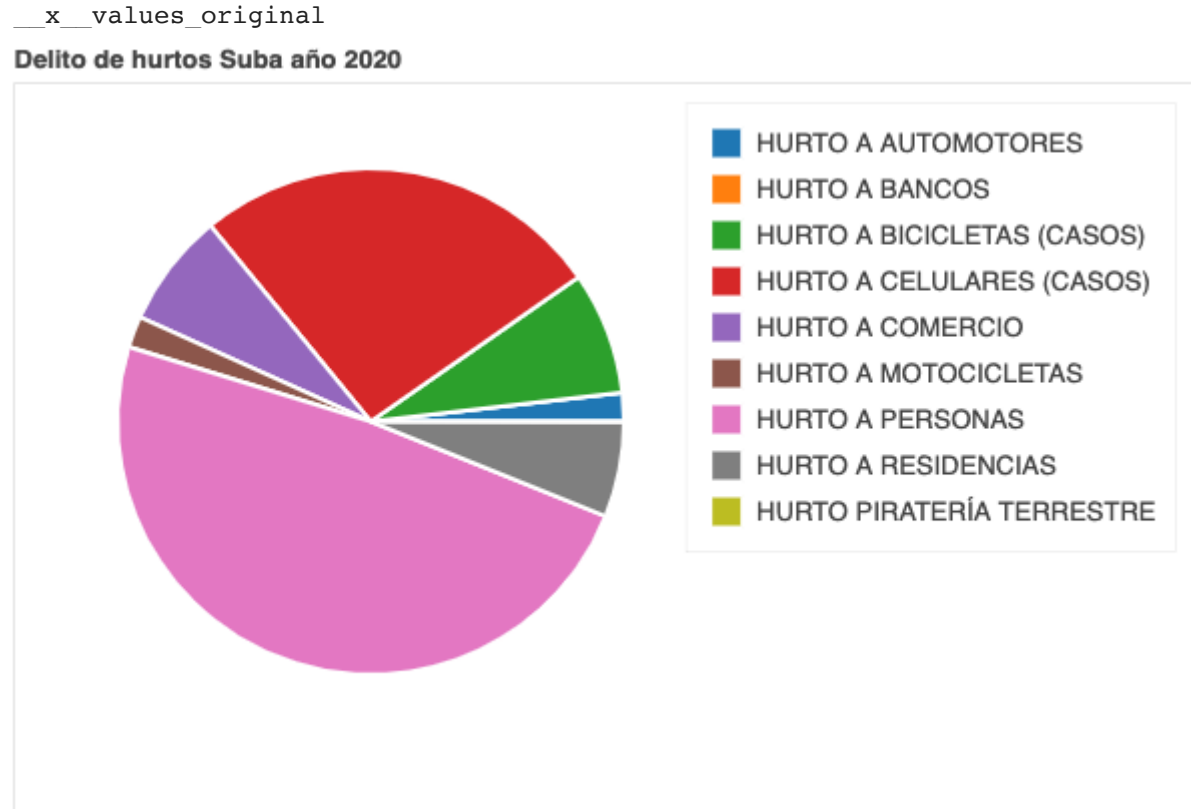
```
Out[65]: FECHA          datetime64[ns]
HORA            int64
MUNICIPIO       object
BARRIOS         object
ZONA            object
CLASE SITIO     object
ARMAS MEDIOS    object
GENERO          object
DELITOS         object
CANTIDAD        int64
CODIGO_LOCALIDAD object
AÑO             int64
MES             int64
DIA_SEMANA     int64
DIA_MES        int64
HORA_INT       object
DIA_SEMANA_INT object
DIA_MES_INT    object
MES_INT        object
dtype: object
```

```
In [71]: df_siedco_suba_2020['DELITOS'].value_counts()
```

```
Out[71]: HURTO A PERSONAS          8819
HURTO A CELULARES (CASOS)   4757
HURTO A BICICLETAS (CASOS)  1420
HURTO A COMERCIO            1327
HURTO A RESIDENCIAS         1103
HURTO A MOTOCICLETAS        361
HURTO A AUTOMOTORES         330
HURTO PIRATERÍA TERRESTRE    4
HURTO A BANCOS              2
Name: DELITOS, dtype: int64
```

```
In [41]: import pandas_bokeh
pandas_bokeh.output_notebook()
df_siedco_suba_2020['DELITOS'].value_counts().sort_index()._set_name('Delitos de hurto').plot_bokeh(
    kind='pie',
    xlabel='Meses',
    ylabel='Cantidad de hurtos',
    title='Delito de hurtos Suba año 2020'
)
```

BokehJS 2.3.2 successfully loaded.

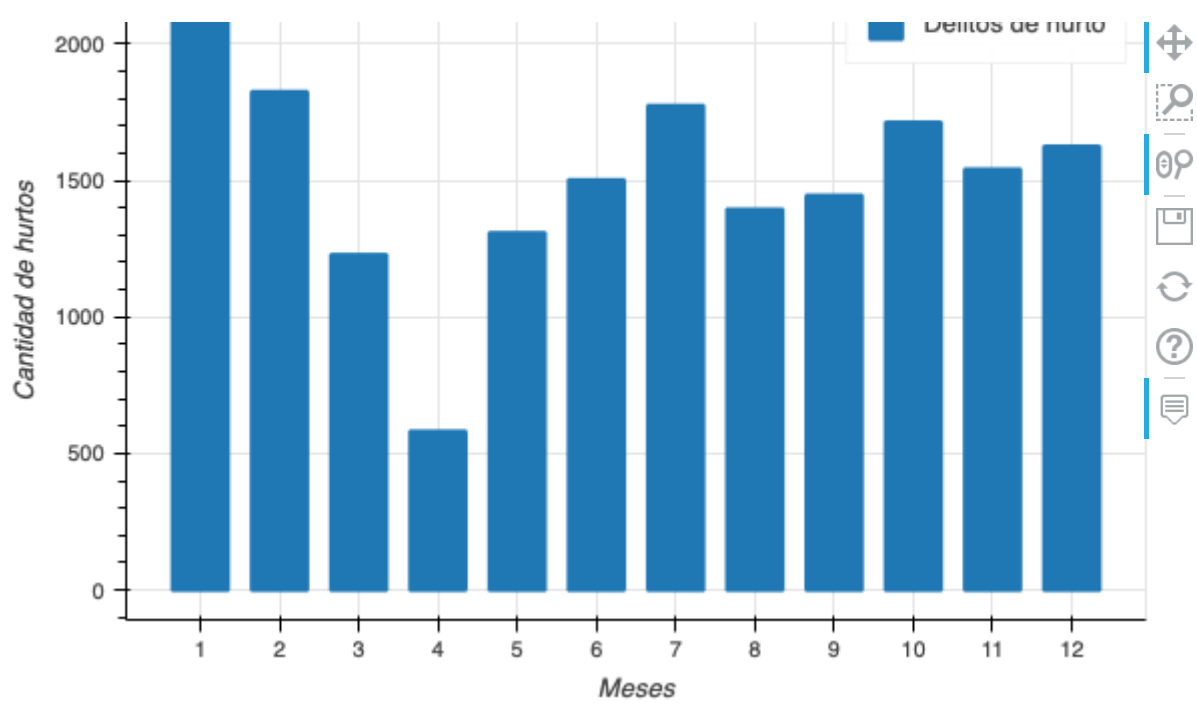


Out[41]: Figure(id = '3926', ...)

```
In [42]: import pandas_bokeh
pandas_bokeh.output_notebook()
df_siedco_suba_2020['MES'].value_counts().sort_index()._set_name('Delitos de hurto').plot_bokeh(
    kind='bar',
    xlabel='Meses',
    ylabel='Cantidad de hurtos',
    title='Delito de hurtos Suba año 2020'
)
```

BokehJS 2.3.2 successfully loaded.

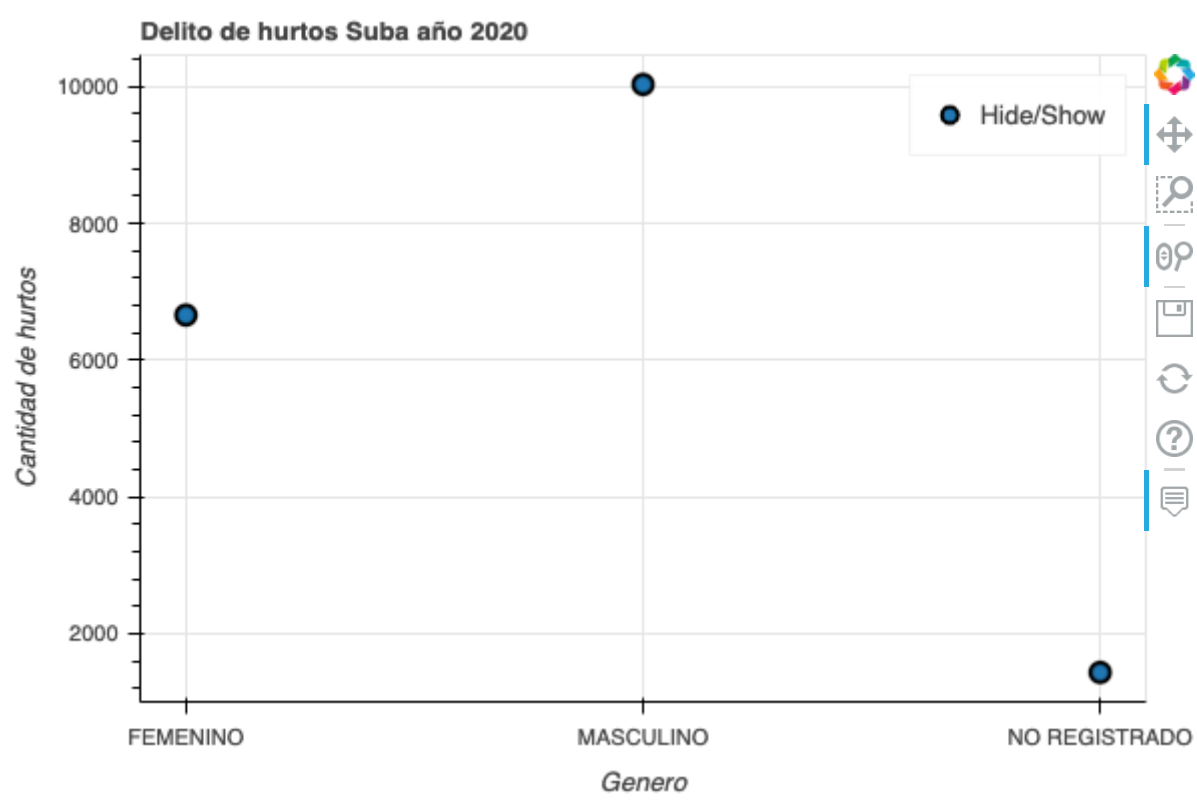




Out[42]: Figure(id = '4077', ...)

```
In [60]: import pandas_bokeh
pandas_bokeh.output_notebook()
df_siedco_suba_2020['GENERO'].value_counts().sort_index().set_name('Delitos de hurto').plot_bokeh(
    kind='scatter',
    xlabel='Genero',
    ylabel='Cantidad de hurtos',
    title='Delito de hurtos Suba año 2020'
)
```

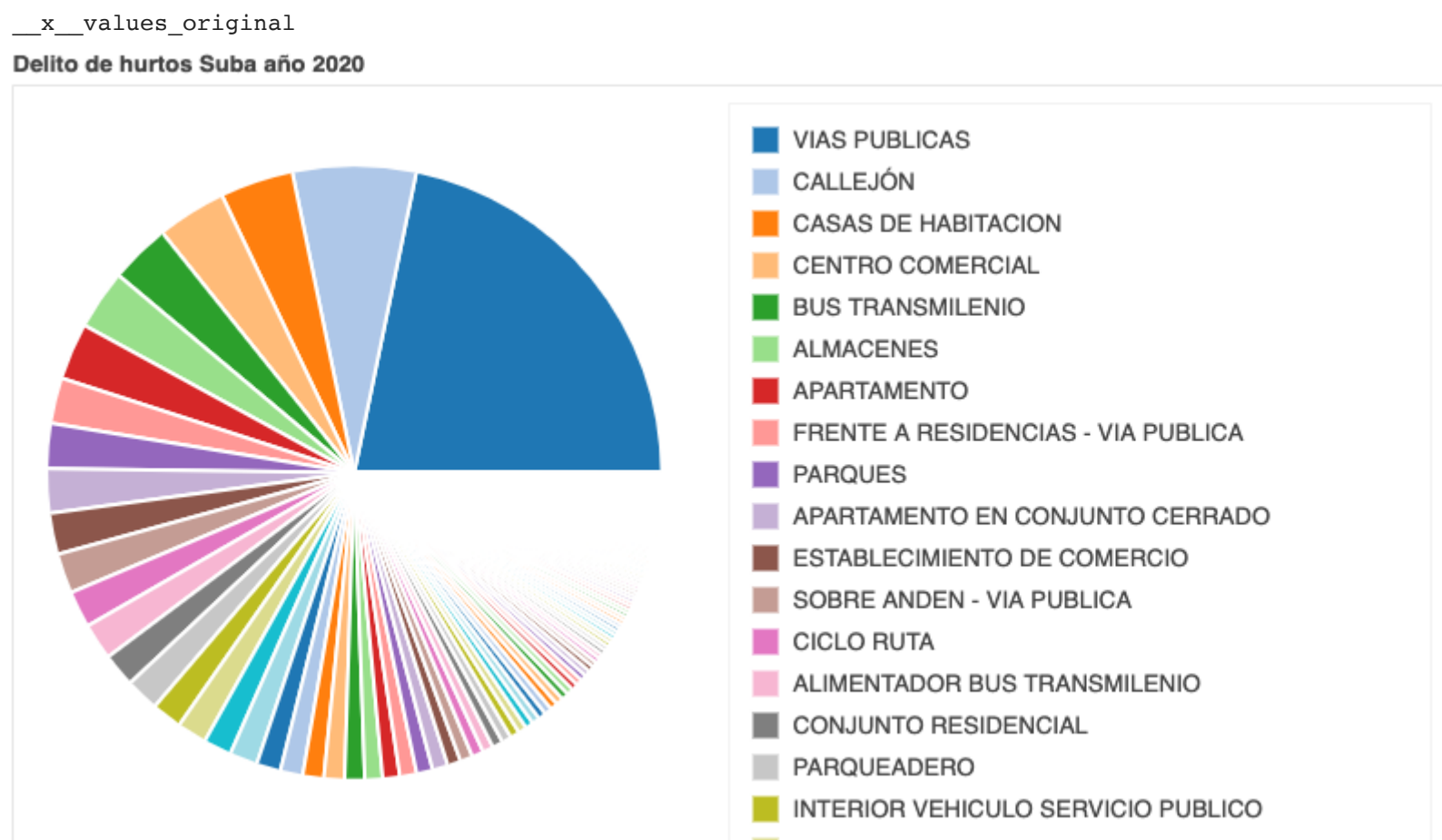
BokehJS 2.3.2 successfully loaded.



Out[60]: Figure(id = '5733', ...)

```
In [53]: import pandas_bokeh
pandas_bokeh.output_notebook()
df_siedco_suba_2020['CLASE SITIO'].value_counts().set_name('Delitos de hurto').plot_bokeh(
    figsize=(800, 450),
    kind='pie',
    xlabel='Clase Sitio',
    ylabel='Cantidad de hurtos',
    title='Delito de hurtos Suba año 2020'
)
```

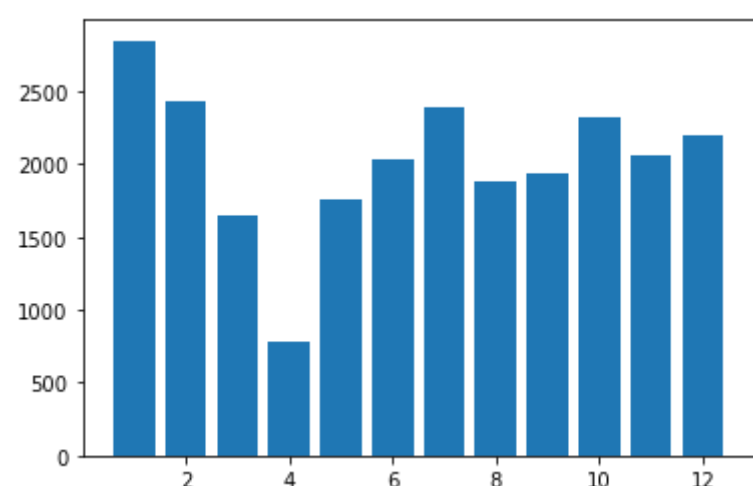
BokehJS 2.3.2 successfully loaded.



Out[53]: Figure(id = '5038', ...)

```
In [14]: plt.bar(df_siedco_suba_2020['MES'].value_counts().index, df_siedco_suba_2020['MES'].value_counts())
```

Out[14]: <BarContainer object of 12 artists>



```
In [15]: df_siedco_suba_2020['CLASE SITIO'].value_counts()
```

```
Out[15]: VIAS PUBLICAS          5251
CALLEJÓN                 1657
CENTRO COMERCIAL         908
CASAS DE HABITACION      863
BUS TRANSMILENIO         825
...
COMPRAVENTA              1
HELADERIA                1
INSTALACIONES DE LA POLICIA 1
ESTACION TRANSMICABLE MANITAS 1
ESTACION TRANSMILENIO PATIO BONITO 1
Name: CLASE SITIO, Length: 209, dtype: int64
```

```
In [16]: df_siedco_suba_2020['DELITOS'].value_counts()
```

```
Out[16]: HURTO A PERSONAS          8819
HURTO A CELULARES (CANTIDAD) 4757
HURTO A CELULARES (CASOS)    4757
HURTO A BICICLETAS (CANTIDAD) 1420
HURTO A BICICLETAS (CASOS)   1420
HURTO A COMERCIO             1327
HURTO A RESIDENCIAS          1103
HURTO A MOTOCICLETAS         361
HURTO A AUTOMOTORES          330
HURTO PIRATERÍA TERRESTRE    4
HURTO A BANCOS                2
Name: DELITOS, dtype: int64
```

## 1.5 Sectores Catastrales en suba

Espacio geográfico en el que se divide el Distrito Capital, de acuerdo a las características de la clase de suelo. Este comprende los barrios en el área urbana, las veredas en el área rural, y mixtos en las áreas rurales con características urbanas. Su extensión geográfica es el área urbana y rural del Distrito Capital.

```
In [17]: df_sca.head()
```

SCACODIGO	SCATIPO	SCANOMBRE	OBJECTID	GLOBALID	geometry
0	9234	0 SABANA DE TIBABUYES NORTE	11	{3A52C763-921B-442D-AAC9-2B37C05FC61F}	[[[-74.104288155, 4.76105034699998], [-74.1040...
1	9243	0 LAS MERCEDES I	49	{5F071F17-F8ED-407F-A705-242E981328BD}	[[[-74.08921468599999, 4.757596653000007], [-7...
2	9112	0 NIZA SUBA	77	{FAE550DE-C4AF-426D-9507-E1F3EF0F6DF8}	[[[-74.074925931, 4.732137048000027], [-74.074...
3	9238	0 SANTA TERESA DE SUBA	93	{25BE4416-29BF-47B5-B69D-0F897D90271C}	[[[-74.09722899600001, 4.720904598000004], [-7...
4	9258	0 CASA BLANCA SUBA I	95	{70ABB853-7A95-4E80-9498-34B2D49C28D5}	[[[-74.07400745799998, 4.736242091999998], [-7...

### Función para obtener el area del poligono de un sector catastral

```
In [18]: def get_area(coordinates):
from shapely.geometry import shape
import json
geo_json = '{"type": "Polygon", "coordinates": '+coordinates+'}'
box = json.loads(geo_json)
return shape(box).area
```

### Función para obtener el centroide del poligono de un sector catastral

```
In [19]: def get_centroid(coordinates):
from geojson_utils import centroid
import json
geo_json = '{"type": "Polygon", "coordinates": '+coordinates+'}'
box = json.loads(geo_json)
return centroid(box)
```

### Función para obtener la mediana del radio del poligono de un sector catastral

```
In [20]: import geopy.distance
import statistics
import math

def get_median_radius(row):
centroid = row['CENTROID']
distances = []
dict_radius = {}
for coordinate in row['geometry'][0]:
coords_1 = (centroid['coordinates'][1], centroid['coordinates'][0])
coords_2 = (coordinate[1], coordinate[0])
distances.append(geopy.distance.geodesic(coords_1, coords_2).m)
dict_radius[geopy.distance.geodesic(coords_1, coords_2).m] = coords_1
return math.floor(statistics.median(distances))
```

### Obtener el centroide y area para cada poligono de los sectores catastrales

```
In [21]: df_sca['CENTROID'] = df_sca['geometry'].apply(lambda x: get_centroid(x))
df_sca['AREA'] = df_sca['geometry'].apply(lambda x: get_area(x))
```

### Casteo del poligono de cada sector catastral a json

```
In [22]: from ast import literal_eval
df_sca['geometry'] = df_sca['geometry'].apply(literal_eval)
```

### Obtener la mediana del radio para cada poligono de los sectores catastrales

```
In [23]: %timeit df_sca['RADIUS_MEDIAN'] = df_sca.apply(lambda x: get_median_radius(x), axis=1)
6.59 s ± 106 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
In [24]: df_sca.head()
```

SCACODIGO	SCATIPO	SCANOMBRE	OBJECTID	GLOBALID	geometry	CENTROID	AREA	RADIUS_MEDIAN
0	9234	0 SABANA DE TIBABUYES NORTE	11	{3A52C763-921B-442D-AAC9-2B37C05FC61F}	[[[-74.104288155, 4.76105034699998], [-74.1040...	{'type': 'Point', 'coordinates': [-74.10699246...	0.000042	478
1	9243	0 LAS MERCEDES I	49	{5F071F17-F8ED-407F-A705-242E981328BD}	[[[-74.08921468599999, 4.757596653000007], [-7...	{'type': 'Point', 'coordinates': [-74.09139869...	0.000005	169
2	9112	0 NIZA SUBA	77	{FAE550DE-C4AF-426D-9507-E1F3EF0F6DF8}	[[[-74.074925931, 4.732137048000027], [-74.074...	{'type': 'Point', 'coordinates': [-74.07829843...	0.000080	799
3	9238	0 SANTA TERESA DE SUBA	93	{25BE4416-29BF-47B5-B69D-0F897D90271C}	[[[-74.09722899600001, 4.720904598000004], [-7...	{'type': 'Point', 'coordinates': [-74.09896395...	0.000045	433
4	9258	0 CASA BLANCA SUBA I	95	{70ABB853-7A95-4E80-9498-34B2D49C28D5}	[[[-74.07400745799998, 4.736242091999998], [-7...	{'type': 'Point', 'coordinates': [-74.07499557...	0.000007	196

## 1.5.1 Barrios de sectores catastrales en Suba

```
In [25]: df_barrios_sca_suba
```

```
Out[25]:
```

	BARRIO	SCA_ID
0	EL BATAN E-11	233.0
1	LA BRITALIA E-11	973.0
2	BOSQUE DE LA COLINA E-11	875.0
3	SAN ANDRES AFIDRO E-11	137.0
4	URB. NUEVA ZELANDIA E-11	921.0
...	...	...
491	SAN FRANCISCO NORTE E-11	250.0
492	CASA EN EL PRADO SAN PEDRO E-11	770.0
493	EL REFUGIO DE LA COLINA II SECTOR E-11	875.0
494	MONTICELLO E-11	929.0
495	HACIENDA HATO CHICO E-11	11.0

496 rows x 2 columns

## 1.6 Indices nocturnos por UPZ en Bogota

In [26]:

```
df_ind_noc_upz.head()
```

Out[26]:

	ALTO	BAJO	FECHA	LOCCODIGO	MEDIO	MUY_ALTO	MUY_BAJO	OBJECTID	P_ILUMINAC	P_PERSONAS	...	T_PERSONAS	T_PRESENCI	T_PRESEN_1	T_PUNTOS	T_QUE_VEO	T QUIENES	T_SEN
0	70.693512	0.671141	1568073600000	13	16.554810	12.080537	0.000000	1	1.134320	1.125634	...	366.0	69.0	81.0	447.0	1107.0	1024.0	
1	65.573770	0.000000	1568073600000	8	14.754098	19.672131	0.000000	2	0.149571	0.295248	...	96.0	5.0	32.0	61.0	136.0	132.0	
2	62.318841	2.028986	1568073600000	13	23.188406	12.173913	0.289855	3	0.823938	0.645856	...	210.0	128.0	83.0	345.0	956.0	760.0	
3	51.612903	2.822581	1568073600000	13	38.104839	6.653226	0.806452	4	1.085039	0.612025	...	199.0	89.0	69.0	496.0	1257.0	1009.0	
4	44.978166	3.930131	1568073600000	3	36.681223	4.803493	9.606987	5	0.561973	0.322928	...	105.0	92.0	23.0	229.0	486.0	315.0	

5 rows x 30 columns

### Filtro por las indices de las UPZ de la localidad de Suba

In [27]:

```
df_ind_noc_suba = df_ind_noc_upz[df_ind_noc_upz['LOCCODIGO']==11]
df_ind_noc_suba.iloc[-1,df_ind_noc_suba.columns.get_loc('UPLCODIGO')] = 'UPZ1'
df_ind_noc_suba['UPZ_CODIGO'] = df_ind_noc_suba['UPLCODIGO'].str.split('UPZ').str[1].astype(int)
df_ind_noc_suba.iloc[-1,df_ind_noc_suba.columns.get_loc('UPLCODIGO')] = '1'
df_ind_noc_suba.head()
```

/opt/anaconda3/envs/tesis/lib/python3.9/site-packages/pandas/core/indexing.py:1720: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
self.setitem\_single\_column(loc, value, pi)  
<ipython-input-27-0becb4c110f9>:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
df\_ind\_noc\_suba['UPZ\_CODIGO'] = df\_ind\_noc\_suba['UPLCODIGO'].str.split('UPZ').str[1].astype(int)

Out[27]:

	ALTO	BAJO	FECHA	LOCCODIGO	MEDIO	MUY_ALTO	MUY_BAJO	OBJECTID	P_ILUMINAC	P_PERSONAS	...	T_PRESENCI	T_PRESEN_1	T_PUNTOS	T_QUE_VEO	T QUIENES	T_SEN	T_TR
9	70.789474	1.052632	1568073600000	11	21.052632	7.105263	0.000000	33	0.949301	0.962633	...	37.0	50.0	380.0	940.0	938.0	1139.0	
10	47.804878	6.991870	1568073600000	11	42.113821	3.089431	0.000000	34	1.529430	0.959557	...	31.0	56.0	615.0	1395.0	1100.0	1824.0	
11	30.409357	18.128655	1568073600000	11	47.368421	3.508772	0.584795	35	0.431422	0.209134	...	17.0	34.0	171.0	375.0	173.0	496.0	
21	56.221198	2.995392	1568073600000	11	33.410138	7.142857	0.230415	14	1.102331	1.107181	...	38.0	96.0	434.0	982.0	841.0	1275.0	
39	0.000000	45.555556	1568073600000	11	15.555556	0.000000	38.888889	53	0.204039	0.027680	...	2.0	1.0	90.0	191.0	7.0	85.0	

5 rows x 31 columns

### Poligonos de las UPZ de la localidad de suba

In [28]:

```
import json

with open('MapsGeoJson/Suba_upz.geojson') as f:
    data = json.load(f)
```

### Función para calcular el area de cada upz mediante jsonpath

In [29]:

```
def get_area_upz(upl_codigo):
    from jsonpath import parse
    from shapely.geometry import shape
    import json
    coordinates = parse("$.features[ @.properties.UPLCODIGO='"+upl_codigo+"'].geometry").find(data)[0]
    return shape(coordinates).area
```

### Función para obtener el poligono de cada upz mediante jsonpath

In [30]:

```
def get_polygon_upz(upl_codigo):
    from jsonpath import parse
    from shapely.geometry import shape
    import json
    coordinates = parse("$.features[ @.properties.UPLCODIGO='"+upl_codigo+"'].geometry").find(data)[0]
    return coordinates
```

### Creación de area y poligono de cada upz

In [31]:

```
df_ind_noc_suba.loc[:, 'geometry'] = df_ind_noc_suba.loc[:, 'UPLCODIGO'].apply(lambda x: get_polygon_upz(x))
df_ind_noc_suba.loc[:, 'area'] = df_ind_noc_suba.loc[:, 'UPLCODIGO'].apply(lambda x: get_area_upz(x))
df_ind_noc_suba.head()
```

/opt/anaconda3/envs/tesis/lib/python3.9/site-packages/pandas/core/indexing.py:1597: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
self.obj[key] = value  
/opt/anaconda3/envs/tesis/lib/python3.9/site-packages/pandas/core/indexing.py:1676: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
self.setitem\_single\_column(ilocs[0], value, pi)

Out[31]:

	ALTO	BAJO	FECHA	LOCCODIGO	MEDIO	MUY_ALTO	MUY_BAJO	OBJECTID	P_ILUMINAC	P_PERSONAS	...	T_PUNTOS	T_QUE_VEO	T QUIENES	T_SEN	T_TRANSPOR	UPLCODIGO	UPI
--	------	------	-------	-----------	-------	----------	----------	----------	------------	------------	-----	----------	-----------	-----------	-------	------------	-----------	-----



	ALTO	BAJO	FECHA	LOCCODIGO	MEDIO	MUY_ALTO	MUY_BAJO	OBJECTID	P_ILUMINAC	P_PERSONAS	...	T_PUNTOS	T_QUE_VEO	T QUIENES	T_SENDERO	T_TRANSPOR	UPLCODIGO	UPI
9	70.789474	1.052632	1568073600000	11	21.052632	7.105263	0.000000	33	0.949301	0.962633	...	380.0	940.0	938.0	1139.0	744.0	UPZ19	
10	47.804878	6.991870	1568073600000	11	42.113821	3.089431	0.000000	34	1.529430	0.959557	...	615.0	1395.0	1100.0	1824.0	1241.0	UPZ24	
11	30.409357	18.128655	1568073600000	11	47.368421	3.508772	0.584795	35	0.431422	0.209134	...	171.0	375.0	173.0	496.0	334.0	UPZ23	
21	56.221198	2.995392	1568073600000	11	33.410138	7.142857	0.230415	14	1.102331	1.107181	...	434.0	982.0	841.0	1275.0	907.0	UPZ27	
39	0.000000	45.555556	1568073600000	11	15.555556	0.000000	38.888889	53	0.204039	0.027680	...	90.0	191.0	7.0	85.0	134.0	UPZ2	

5 rows x 33 columns

## 1.8 Base de Sectores catastrales por UPZ y asociación de Índices Nocturnos por UPZ

Teniendo en cuenta los sectores catastrales que pertenecen a cada UPZ se realiza una mezcla de las diferentes fuentes de datos, teniendo en cuenta que los índices nocturnos están dados por UPZ, mediante el área de cada UPZ y de cada sector catastral se puede sacar el porcentaje o proporción de cada uno de los indicadores para cada sector catastral.

```
In [32]: df_base_sca_upz = df_sca.merge(df_upz_sca, left_on='OBJECTID', right_on='SCA_ID', how='inner')
df_base_sca_upz = df_base_sca_upz[df_base_sca_upz['UPZ_ID'].notnull()]
df_base_sca_upz.head()
```

Out[32]:	SCACODIGO	SCATIPO	SCANOMBRE	OBJECTID	GLOBALID	geometry	CENTROID	AREA	RADIUS_MEDIAN	UPZ_ID	SCA_ID
0	9234	0	SABANA DE TIBABUYES NORTE	11	{3A52C763-921B-442D-AAC9-2B37C05FC61F}	{[-74.104288155, 4.76105034699998], [-74.1040...]}	{'type': 'Point', 'coordinates': [-74.10699246...]}	0.000042	478	71	11
1	9243	0	LAS MERCEDES I	49	{5F071F17-F8ED-407F-A705-242E981328BD}	{[-74.08921468599999, 4.757596653000007], [-7...]}	{'type': 'Point', 'coordinates': [-74.09139869...]}	0.000005	169	27	49
2	9112	0	NIZA SUBA	77	{FAE550DE-C4AF-426D-9507-E1F3EF0F6DF8}	{[-74.074925931, 4.732137048000027], [-74.074...]}	{'type': 'Point', 'coordinates': [-74.07829843...]}	0.000080	799	24	77
3	9238	0	SANTA TERESA DE SUBA	93	{25BE4416-29BF-47B5-B69D-0F897D90271C}	{[-74.09722899600001, 4.720904598000004], [-7...]}	{'type': 'Point', 'coordinates': [-74.09896395...]}	0.000045	433	28	93
4	9258	0	CASA BLANCA SUBA I	95	{70ABB853-7A95-4E80-9498-34B2D49C28D5}	{[-74.07400745799998, 4.736242091999998], [-7...]}	{'type': 'Point', 'coordinates': [-74.07499557...]}	0.000007	196	23	95

### Función para consultar los lugares más cercanos a un sector catastral teniendo en cuenta su radio

Se utiliza el api de google maps para buscar los lugares más cercanos, los siguientes son los parametros que se utilizan:

- **location:** Latitud y Longitud del punto sobre el cual se va a realizar la búsqueda, en este caso el centroide del polígono del sector catastral.
- **radius:** Radio sobre el cual se va a realizar la búsqueda en metros, en este caso la mediana del radio del polígono del sector catastral.
- **type:** Tipo del lugar sobre el que se va a hacer la búsqueda, para más información ver el [listado de tipos soportados](#)
- **key:** API Key generada en la cuenta de google cloud, fue seteada como variable de entorno para protegerla.

Para más información se puede consultar la [documentación](#)

```
In [33]: def get_nearby_places(row, place_type):
import requests
import json
import os
api_key = os.environ['GOOGLE_API_KEY']
latitude = row['CENTROID']['coordinates'][1]
longitude = row['CENTROID']['coordinates'][0]
radius = row['RADIUS_MEDIAN']
url = f"https://maps.googleapis.com/maps/api/place/nearbysearch/json?location={latitude},{longitude}&radius={radius}&type={place_type}&key={api_key}"
r = requests.get(url)
data = json.loads(r.text)
return data
```

### Consulta de bares por sector catastral

```
In [34]: df_base_sca_upz['BAR_JSON_RES'] = df_base_sca_upz.apply(lambda x: get_nearby_places(x, 'bar'), axis=1)
```

### Consulta de bancos por sector catastral

```
In [35]: df_base_sca_upz['BANK_JSON_RES'] = df_base_sca_upz.apply(lambda x: get_nearby_places(x, 'bank'), axis=1)
```

### Consulta de estaciones de policia por sector catastral

```
In [36]: df_base_sca_upz['STORE_JSON_RES'] = df_base_sca_upz.apply(lambda x: get_nearby_places(x, 'store'), axis=1)
```

### Consulta de cajeros por sector catastral

```
In [37]: df_base_sca_upz['ATM_JSON_RES'] = df_base_sca_upz.apply(lambda x: get_nearby_places(x, 'atm'), axis=1)
```

### Consulta de parques por sector catastral

```
In [38]: df_base_sca_upz['PARK_JSON_RES'] = df_base_sca_upz.apply(lambda x: get_nearby_places(x, 'park'), axis=1)
```

### Creación de densidades basada en los resultados del api de google maps

```
In [39]: df_base_sca_upz['DENSIDAD_BANCARIA'] = df_base_sca_upz['BANK_JSON_RES'].apply(lambda x: len(x['results']))
df_base_sca_upz['DENSIDAD_NOCTURNA'] = df_base_sca_upz['BAR_JSON_RES'].apply(lambda x: len(x['results']))
df_base_sca_upz['DENSIDAD_TIENDAS'] = df_base_sca_upz['STORE_JSON_RES'].apply(lambda x: len(x['results']))
df_base_sca_upz['DENSIDAD_CAJEROS'] = df_base_sca_upz['ATM_JSON_RES'].apply(lambda x: len(x['results']))
df_base_sca_upz['DENSIDAD_PARQUES'] = df_base_sca_upz['PARK_JSON_RES'].apply(lambda x: len(x['results']))
```

### Casteo de identificadores de UPZ y SCA para poder realizar el merge con la base de índices nocturnos por UPZ

```
In [40]: df_base_sca_upz['UPZ_ID'] = df_base_sca_upz['UPZ_ID'].astype(int)
df_base_sca_upz['SCA_ID'] = df_base_sca_upz['SCA_ID'].astype(int)
```

### Merge con la base de índices nocturnos por UPZ y renombramiento de algunas columnas

```
In [42]: df_base_sca_upz = df_base_sca_upz.merge(df_ind_noc_suba, left_on='UPZ_ID', right_on='UPZ_CODIGO', how='left')
df_base_sca_upz.rename(columns={'geometry_x': 'SCA_GEOMETRY'}, inplace=True)
df_base_sca_upz.rename(columns={'OBJECTID_x': 'SCA_OBJECTID'}, inplace=True)
df_base_sca_upz.rename(columns={'AREA': 'SCA_AREA'}, inplace=True)
df_base_sca_upz.rename(columns={'geometry_y': 'UPZ_GEOMETRY'}, inplace=True)
df_base_sca_upz.rename(columns={'area': 'UPZ_AREA'}, inplace=True)
df_base_sca_upz.head()
```

Out[42]:	SCACODIGO	SCATIPO	SCANOMBRE	SCA_OBJECTID	GLOBALID	SCA_GEOMETRY	CENTROID	SCA_AREA	RADIUS_MEDIAN	UPZ_ID	T_PUNTOS	T_QUE_VEO	T QUIENES	T_SENDERO	T_TRANSF
0	9234	0	SABANA DE TIBABUYES NORTE	11	{3A52C763-921B-442D-AAC9-2B37C05FC61F}	[[[-74.104288155, 4.76105034699998], [-74.1040... [-74.10699246...]]	{'type': 'Point', 'coordinates': [-74.10699246...]]	0.000042	478	71	658.0	1715.0	1730.0	1901.0	130
1	9243	0	LAS MERCEDES I	49	{5F071F17-F8ED-407F-A705-242E981328BD}	[[[-74.08921468599999, 4.757596653000007], [-7... [-74.09139869...]]	{'type': 'Point', 'coordinates': [-74.09139869...]]	0.000005	169	27	434.0	982.0	841.0	1275.0	90
2	9112	0	NIZA SUBA	77	{FAE550DE-C4AF-426D-9507-E1F3EF0F6DF8}	[[[-74.074925931, 4.732137048000027], [-74.074... [-74.07829843...]]	{'type': 'Point', 'coordinates': [-74.07829843...]]	0.000080	799	24	615.0	1395.0	1100.0	1824.0	124
3	9238	0	SANTA TERESA DE SUBA	93	{25BE4416-29BF-47B5-B69D-0F897D90271C}	[[[-74.09722899600001, 4.720904598000004], [-7... [-74.09896395...]]	{'type': 'Point', 'coordinates': [-74.09896395...]]	0.000045	433	28	679.0	1580.0	1502.0	2018.0	135
4	9258	0	CASA BLANCA SUBA I	95	{70ABB853-7A95-4E80-9498-34B2D49C28D5}	[[[-74.07400745799998, 4.736242091999998], [-7... [-74.07499557...]]	{'type': 'Point', 'coordinates': [-74.07499557...]]	0.000007	196	23	171.0	375.0	173.0	496.0	35

5 rows x 54 columns

### Proporción de cada sector catastral en relación a la UPZ

```
In [44]: df_base_sca_upz['PROP'] = (df_base_sca_upz['SCA_AREA']*1000)/(df_base_sca_upz['UPZ_AREA']*1000)
```

```
In [45]: df_base_sca_upz.to_csv('Datasets/Bases/base_sca_upz.csv', sep=',', encoding='utf-8', index=False)
```

### 1.9 Base final

#### Merge con la base de índices nocturnos por UPZ y SCA con la base de delitos del 2020 en suba

```
In [47]: df_base_sca_upz.rename(columns={'BARRIOS':'BARRIO'}, inplace=True)
df_base_barrios_sca = df_siedco_suba_2020.merge(df_barrios_sca_suba, how='inner', on='BARRIO')
df_base_barrios_sca['SCA_ID'] = df_base_barrios_sca['SCA_ID'].astype(int)
df_base_delitos_barrios = df_base_barrios_sca.merge(df_base_sca_upz, left_on='SCA_ID', right_on='SCA_OBJECTID', how='inner')
```

```
In [48]: df_base_delitos_barrios.head()
```

Out[48]:	FECHA_x	HORA	MUNICIPIO	BARRIOS	ZONA	CLASE SITIO	ARMAS MEDIOS	GENERO	DELITOS	CANTIDAD	T_QUE_VEO	T QUIENES	T_SENDERO	T_TRANSPOR	UPLCODIGO	UPLNOMBRE	UPZ_CODIGO
0	2020-12-31	0	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	GARAJE	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	705.0	598.0	923.0	562.0	UPZ20	LA ALHAMBRA	20
1	2020-12-30	16	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	PARQUES	ARMA DE FUEGO	MASCULINO	HURTO A PERSONAS	1	705.0	598.0	923.0	562.0	UPZ20	LA ALHAMBRA	20
2	2020-12-29	3	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	CONSTRUCCION	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	705.0	598.0	923.0	562.0	UPZ20	LA ALHAMBRA	20
3	2020-12-28	5	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	HUMEALES	ARMA DE FUEGO	MASCULINO	HURTO A PERSONAS	1	705.0	598.0	923.0	562.0	UPZ20	LA ALHAMBRA	20
4	2020-12-28	5	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	HUMEALES	ARMA DE FUEGO	MASCULINO	HURTO A CELULARES (CANTIDAD)	1	705.0	598.0	923.0	562.0	UPZ20	LA ALHAMBRA	20

5 rows x 76 columns

#### Creación de features por SCA teniendo en cuenta la proporción de los índices por UPZ

```
In [50]: df_base_delitos_barrios.loc[:, 'ILUMINACION'] = df_base_delitos_barrios['T_ILUMINAC']*df_base_delitos_barrios['PROP']
df_base_delitos_barrios.loc[:, 'PERSONAS'] = df_base_delitos_barrios['T_PERSONAS']*df_base_delitos_barrios['PROP']
df_base_delitos_barrios.loc[:, 'SEGURIDAD'] = df_base_delitos_barrios['T_PRESENCI']*df_base_delitos_barrios['PROP']
df_base_delitos_barrios.loc[:, 'SENDERO'] = df_base_delitos_barrios['T_SENDERO']*df_base_delitos_barrios['PROP']
df_base_delitos_barrios.loc[:, 'TRANSPORTE'] = df_base_delitos_barrios['T_TRANSPOR']*df_base_delitos_barrios['PROP']
```

#### Renombramiento de algunas columnas

```
In [51]: df_base_delitos_barrios.rename(columns={'CLASE SITIO':'CLASE_SITIO'}, inplace=True)
df_base_delitos_barrios.rename(columns={'ARMAS MEDIOS':'ARMAS_MEDIOS'}, inplace=True)
df_base_delitos_barrios.rename(columns={'FECHA_x':'FECHA_DELITO'}, inplace=True)
df_base_delitos_barrios.rename(columns={'FECHA_y':'FECHA_INDICES'}, inplace=True)
df_base_delitos_barrios.rename(columns={'geometry':'UPZ_GEOMETRY'}, inplace=True)
```

#### Eliminación de outliers y eliminación de features repetidos

```
In [52]: delitos_to_clean = ['HURTO A CELULARES (CANTIDAD)', 'HURTO A BICICLETAS (CANTIDAD)', 'HURTO PIRATERÍA TERRESTRE', 'HURTO A BANCOS']
df_base_delitos_barrios = df_base_delitos_barrios[~df_base_delitos_barrios['DELITOS'].isin(delitos_to_clean)]
df_base_delitos_barrios.drop('SCA_ID_y', axis=1, inplace=True)
df_base_delitos_barrios.drop('OBJECTID_y', axis=1, inplace=True)
```

```
In [53]: df_base_delitos_barrios.head()
```

Out[53]:	FECHA_DELITO	HORA	MUNICIPIO	BARRIOS	ZONA	CLASE_SITIO	ARMAS_MEDIOS	GENERO	DELITOS	CANTIDAD	UPLNOMBRE	UPZ_CODIGO	UPZ_GEOMETRY	UPZ_AREA	PROP	ILUMINACION
0	2020-12-31	0	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	GARAJE	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	LA ALHAMBRA	20	{'type': 'Polygon', 'coordinates': [[[-74.0539...]]}	0.000232	0.358233	322.409633
1	2020-12-30	16	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	PARQUES	ARMA DE FUEGO	MASCULINO	HURTO A PERSONAS	1	LA ALHAMBRA	20	{'type': 'Polygon', 'coordinates': [[[-74.0539...]]}	0.000232	0.358233	322.409633
2	2020-12-29	3	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	CONSTRUCCION	SIN EMPLEO DE ARMAS	MASCULINO	HURTO A PERSONAS	1	LA ALHAMBRA	20	{'type': 'Polygon', 'coordinates': [[[-74.0539...]]}	0.000232	0.358233	322.409633
3	2020-12-28	5	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	HUMEALES	ARMA DE FUEGO	MASCULINO	HURTO A PERSONAS	1	LA ALHAMBRA	20	{'type': 'Polygon', 'coordinates': [[[-74.0539...]]}	0.000232	0.358233	322.409633
5	2020-12-28	5	BOGOTÁ D.C. (CT)	EL BATAN E-11	URBANA	HUMEALES	ARMA DE FUEGO	MASCULINO	HURTO A CELULARES (CASOS)	1	LA ALHAMBRA	20	{'type': 'Polygon', 'coordinates': [[[-74.0539...]]}	0.000232	0.358233	322.409633

5 rows x 79 columns

```
In [54]: df_base_delitos_barrios.columns
```

```
Out[54]: Index(['FECHA_DELITO', 'HORA', 'MUNICIPIO', 'BARRIOS', 'ZONA', 'CLASE_SITIO',
```

```
'ARMAS_MEDIOS', 'GENERO', 'DELITOS', 'CANTIDAD', 'CODIGO_LOCALIDAD',
'AÑO', 'MES', 'DIA_SEMANA', 'DIA_MES', 'HORA_INT', 'DIA_SEMANA_INT',
'DIA_MES_INT', 'MES_INT', 'BARRIO', 'SCA_ID_x', 'SCACODIGO', 'SCATIPO',
'SCANOMBRE', 'SCA_OBJECTID', 'GLOBALID', 'SCA_GEOMETRY', 'CENTROID',
'SCA_AREA', 'RADIUS_MEDIAN', 'UPZ_ID', 'BAR_JSON_RES', 'BANK_JSON_RES',
'STORE_JSON_RES', 'ATM_JSON_RES', 'PARK_JSON_RES', 'DENSIDAD_BANCARIA',
'DENSIDAD_NOCTURNA', 'DENSIDAD_TIENDAS', 'DENSIDAD_CAJEROS',
'DENSIDAD_PARQUES', 'ALTO', 'BAJO', 'FECHA_INDICES', 'LOCCODIGO',
'MEDIO', 'MUY_ALTO', 'MUY_BAJO', 'P_ILUMINAC', 'P_PERSONAS',
'P_PRESENCI', 'P_PRESEN_1', 'P_QUE_VEO', 'P QUIENES', 'P_SENDERO',
'P_TRANSPOR', 'P_UPL', 'SHAPE.AREA', 'SHAPE.LEN', 'T_ILUMINAC',
'T_PERSONAS', 'T_PRESENCI', 'T_PRESEN_1', 'T_PUNTOS', 'T_QUE_VEO',
'T_QUIENES', 'T_SENDERO', 'T_TRANSPOR', 'UPLCODIGO', 'UPLNOMBRE',
'UPZ_CODIGO', 'UPZ_GEOMETRY', 'UPZ_AREA', 'PROP', 'ILUMINACION',
'PERSONAS', 'SEGURIDAD', 'SENDERO', 'TRANSPORTE'],
dtype='object')
```

```
In [55]: df_base_delitos_barrrios.to_csv('Datasets/Bases/base_delitos_barrrios_2020.csv', sep=',', encoding='utf-8', index=False)
```