



INTELIGENCIA ARTIFICIAL I - E2

# UN VISTAZO A LOS ACCIDENTES DE TRÁNSITO EN BGA CON IA

Julián David Pérez Uribe  
Yonatan Fernando Montañez Araque  
Juan Sebastián Mantilla Serrano



# OBJETIVO

---

Nuestra finalidad al desarrollar este proyecto es **predecir la gravedad del accidente** en base al tipo de vehículo involucrado, periodo del accidente (diurno/nocturno), la comuna donde ocurrió y el mes o día del accidente.



# DATASET

---

El dataset fue obtenido Datos Abiertos Colombia a través del siguiente enlace **[https://www.datos.gov.co/Transporte/Accidentes-de-Transito-ocurridos-en-el-Municipio-d/7cci-nqqb/about\\_data](https://www.datos.gov.co/Transporte/Accidentes-de-Transito-ocurridos-en-el-Municipio-d/7cci-nqqb/about_data)**.

El dataset contiene información sobre accidentes de tránsito en la ciudad de Bucaramanga, donde cada fila representa un dato estadístico de accidente registrado y notificado, tomando datos de la fecha, año, mes, día, hora, comuna, barrio, tipo de actor vial involucrado, gravedad, entre otros.

# DATASET

Numero de registros-columnas: (39193, 24)

	ORDEN	FECHA	AÑO	MES	DÍA	GRAVEDAD	PEATON	AUTOMOVIL	CAMPERO	CAMIONETA	...	VOLQUETA	MOTO	BICICLETA	OTRO	BARRIO	HORA	ENTIDAD	COMUNA	Propietario de Vehículo	DIURNIO/NOCTURNO
0	1	01/01/2012	2012	01. Enero	07. Domingo	Con heridos	0	1	0	0	...	0	0	0	0	Mutis	12:15:00	AGENTES DTB	17. MUTIS	Particular	Diurno
1	2	01/01/2012	2012	01. Enero	07. Domingo	Solo daños	0	1	0	1	...	0	0	0	0	Regaderos Norte	14:00:00	AGENTES DTB	02. NORORIENTAL	Empresa	Diurno
2	3	01/01/2012	2012	01. Enero	07. Domingo	Solo daños	0	0	0	1	...	0	0	0	0	Cabecera Del Llano	12:00:00	AGENTES DTB	12. CABECERA DEL LLANO	Particular	Diurno
3	4	01/01/2012	2012	01. Enero	07. Domingo	Solo daños	0	1	0	1	...	0	0	0	0	Norte Bajo	18:30:00	AGENTES DTB	03. SAN FRANCISCO	Empresa	Nocturno
4	5	01/01/2012	2012	01. Enero	07. Domingo	Con heridos	1	0	0	0	...	0	1	0	0	Dangond	00:30:00	AGENTES DTB	11. SUR	Particular	Nocturno
5	6	01/01/2012	2012	01. Enero	07. Domingo	Solo daños	0	1	0	0	...	0	1	0	0	San Miguel	21:10:00	AGENTES DTB	06. LA CONCORDIA	Empresa	Nocturno
6	7	02/01/2012	2012	01. Enero	01. Lunes	Con heridos	1	0	0	0	...	0	0	0	0	La Pedregosa	13:50:00	AGENTES DTB	09. LA PEDREGOSA	Empresa	Diurno
7	8	02/01/2012	2012	01. Enero	01. Lunes	Solo daños	0	0	0	2	...	0	0	0	0	Bucaramanga	19:20:00	AGENTES DTB	08. SUR OCCIDENTE	Particular	Nocturno
8	9	02/01/2012	2012	01. Enero	01. Lunes	Solo daños	0	1	0	0	...	0	0	0	1	Regaderos Norte	12:00:00	AGENTES DTB	02. NORORIENTAL	Empresa	Diurno
9	10	02/01/2012	2012	01. Enero	01. Lunes	Con heridos	0	1	0	0	...	0	0	0	0	Tejar Norte	16:35:00	AGENTES DTB	01. NORTE	Particular	Diurno

10 rows × 24 columns

Filas  
39,2K

Columnas  
24

# DATASET

## Tratamiento de datos

```
1 #Borrar columna fecha y hora
2 df.drop(['FECHA', 'HORA', 'BARRIO'], axis=1, inplace=True)
3 df.drop(df[df['COMUNA'] == 'NO DISPONIBLE'].index, inplace=True)
4
5 # Modificar variables por valores más simples
6 df.MES.replace({
7     '01. Enero': 1,
8     '02. Febrero': 2,
9     '03. Marzo': 3,
10    '04. Abril': 4,
11    '05. Mayo': 5,
12    '06. Junio': 6,
13    '07. Julio': 7,
14    '08. Agosto': 8,
15    '09. Septiembre': 9,
16    '10. Octubre': 10,
17    '11. Noviembre': 11,
18    '12. Diciembre': 12
19 }, inplace=True)
20
21 df.DÍA.replace({
22     '01. Lunes': 1,
23     '02. Martes': 2,
24     '03. Miercoles': 3,
25     '04. Jueves': 4,
26     '05. Viernes': 5,
27     '06. Sabado': 6,
28     '07. Domingo': 7
29 }, inplace=True)
30
31 df['DIURNIO/NOCTURNO'].replace({
32     'Diurno': 1,
33     'Nocturno': 0
34 }, inplace=True)
```

```
36 df['GRAVEDAD'] = df['GRAVEDAD'].replace('Con Muertos', 'Con muertos')
37
38 df['GRAVEDAD'].replace({
39     'Solo daños': 1,
40     'Con heridos': 2,
41     'Con muertos': 3
42 }, inplace=True)
43
44 df.ENTIDAD.replace({
45     'AGENTES DTB': 1,
46     'AGENTES POLICIA NACIONAL': 2
47 }, inplace=True)
48
49 df.COMUNA.replace({
50     'CORREG. 1': 'CORREGIMIENTO',
51     'CORREG. 2': 'CORREGIMIENTO',
52     'CORREG. 3': 'CORREGIMIENTO'
53 }, inplace=True)
54
55 df.rename(columns={'Propietario de Vehículo': 'Propietario_Vehiculo'}, inplace=True)
56
57 df.Propietario_Vehiculo.replace({
58     'Particular': 1,
59     'Empresa': 2
60 }, inplace=True)
```

# DATASET

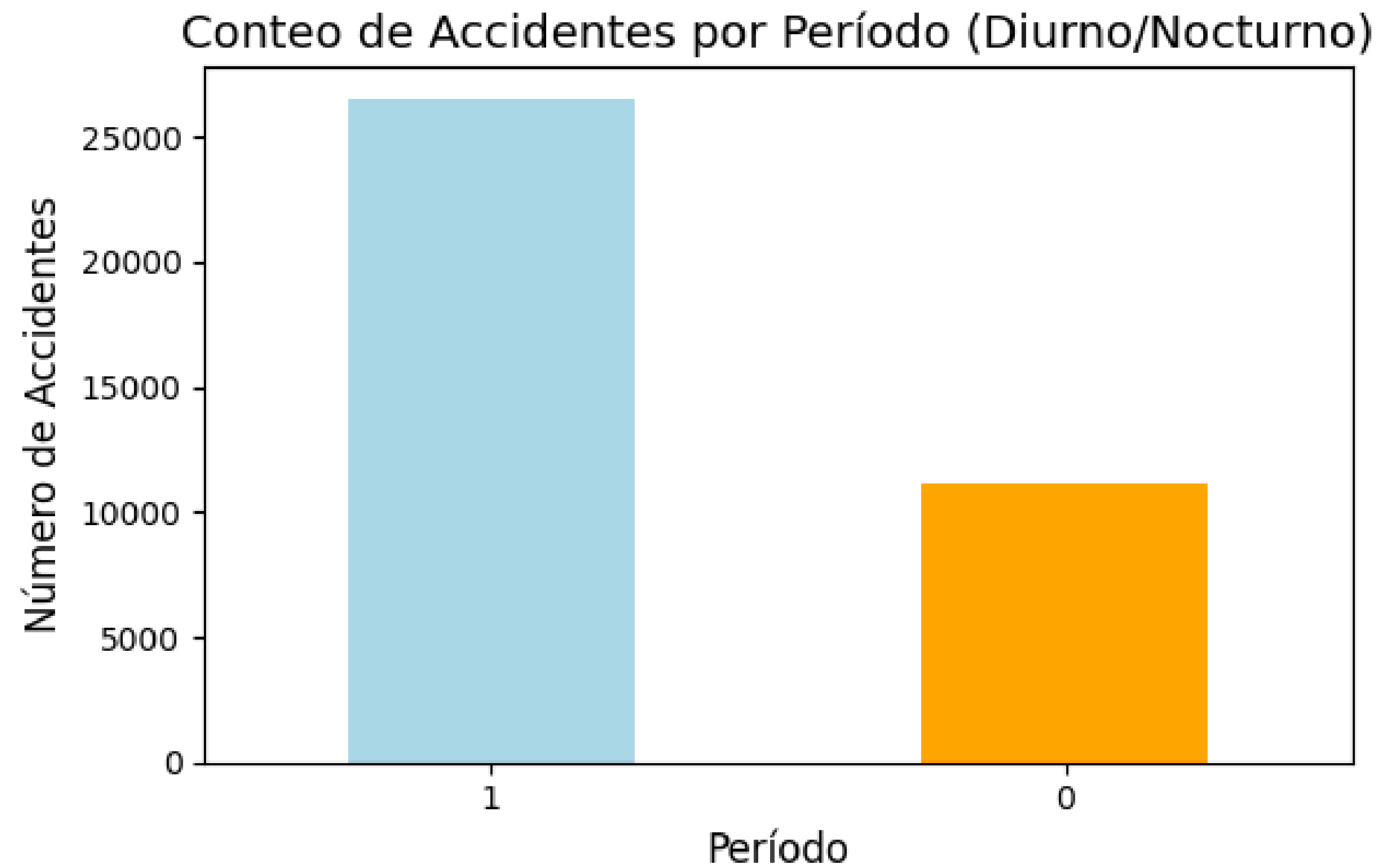
## Tipos de datos de las columnas, revisión de valores únicos y de valores nulos

```
<class 'pandas.core.frame.DataFrame'>
Index: 37640 entries, 0 to 39192
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   ORDEN                  37640 non-null  int64  
1   AÑO                    37640 non-null  int64  
2   MES                    37640 non-null  int64  
3   DÍA                    37640 non-null  int64  
4   GRAVEDAD               37640 non-null  int64  
5   PEATON                 37640 non-null  int64  
6   AUTOMOVIL             37640 non-null  int64  
7   CAMPERO                37640 non-null  int64  
8   CAMIONETA              37640 non-null  int64  
9   MICRO                 37640 non-null  int64  
10  BUSETA                 37640 non-null  int64  
11  BUS                    37640 non-null  int64  
12  CAMION                 37640 non-null  int64  
13  VOLQUETA               37640 non-null  int64  
14  MOTO                   37640 non-null  int64  
15  BICICLETA              37640 non-null  int64  
16  OTRO                   37640 non-null  int64  
17  ENTIDAD                37640 non-null  int64  
18  COMUNA                 37640 non-null  object  
19  Propietario_Vehiculo   37640 non-null  int64  
20  DIURNIO/NOCTURNO       37640 non-null  int64  
dtypes: int64(20), object(1)
memory usage: 6.3+ MB
```

```
ORDEN [ 1 2 3 ... 39190 39191 39193]
AÑO [2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023]
MES [ 1 2 3 4 5 6 7 8 9 10 11 12]
DÍA [7 1 2 3 4 5 6]
GRAVEDAD [2 1 3]
PEATON [0 1 2 3 4 6 5 7]
AUTOMOVIL [ 1 0 2 3 4 17 5 10 13 29]
CAMPERO [0 1 2]
CAMIONETA [0 1 2 3]
MICRO [0 1 2]
BUSETA [0 1 2]
BUS [0 1 2]
CAMION [0 1 2 3]
VOLQUETA [0 1 2]
MOTO [0 1 2 3 4 8 5 7 6]
BICICLETA [0 1 2 3]
OTRO [0 1 2]
ENTIDAD [1 2]
COMUNA ['17. MUTIS' '02. NORORIENTAL' '12. CABECERA DEL LLANO'
'03. SAN FRANCISCO' '11. SUR' '06. LA CONCORDIA' '09. LA PEDREGOSA'
'08. SUR OCCIDENTE' '01. NORTE' '10. PROVENZA' '15. CENTRO'
'13. ORIENTAL' '07. LA CIUDADELA' '04. OCCIDENTAL' 'CORREGIMIENTO'
'16. LAGOS DEL CACIQUE' '14. MORRORICO' '05. GARCIA ROVIRA']
Propietario_Vehiculo [1 2]
DIURNIO/NOCTURNO [1 0]
```

```
ORDEN 0
AÑO 0
MES 0
DÍA 0
GRAVEDAD 0
PEATON 0
AUTOMOVIL 0
CAMPERO 0
CAMIONETA 0
MICRO 0
BUSETA 0
BUS 0
CAMION 0
VOLQUETA 0
MOTO 0
BICICLETA 0
OTRO 0
ENTIDAD 0
COMUNA 0
Propietario_Vehiculo 0
DIURNIO/NOCTURNO 0
```

# ALGUNAS GRÁFICAS

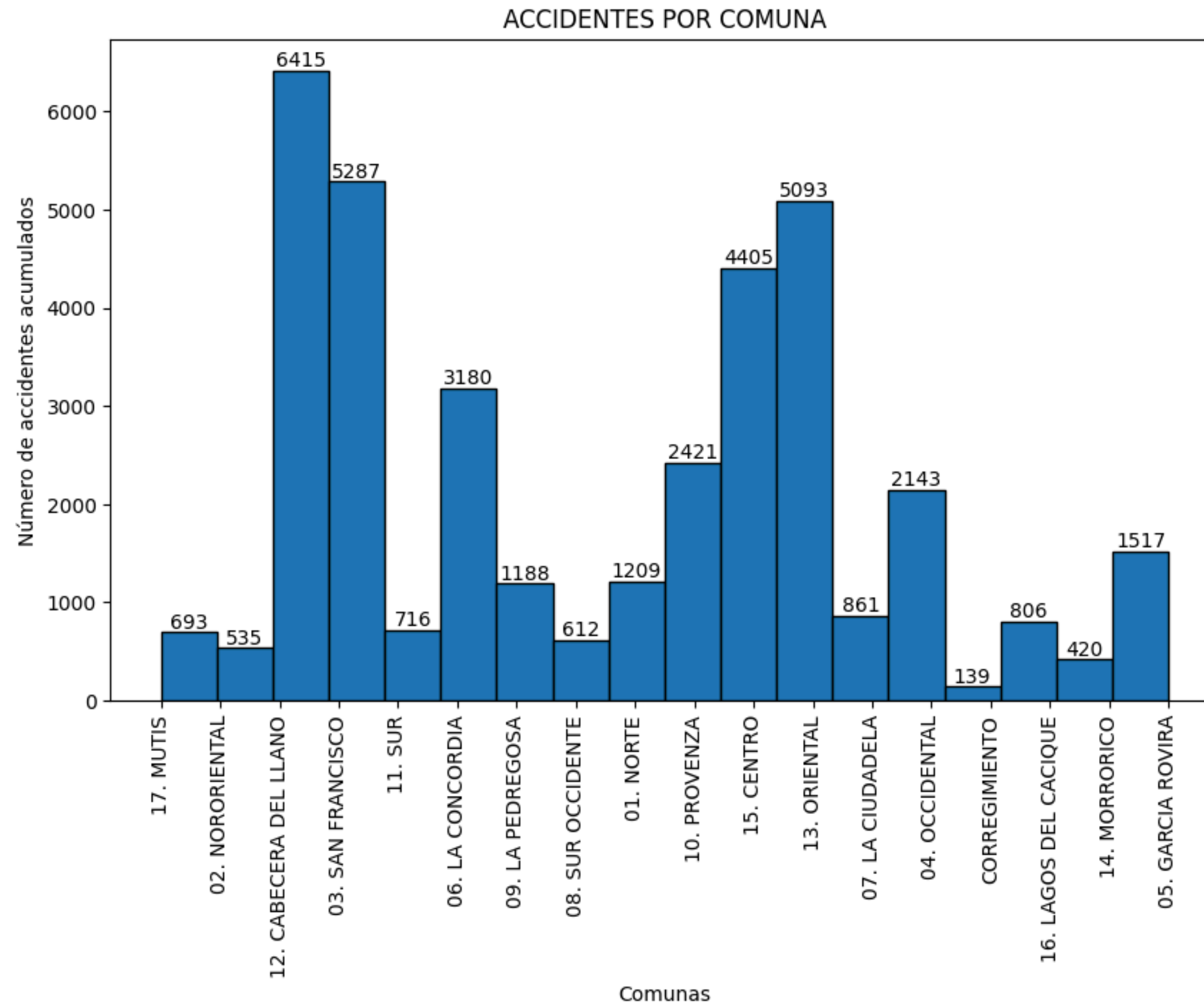


- Diurno: 1
- Nocturno: 0





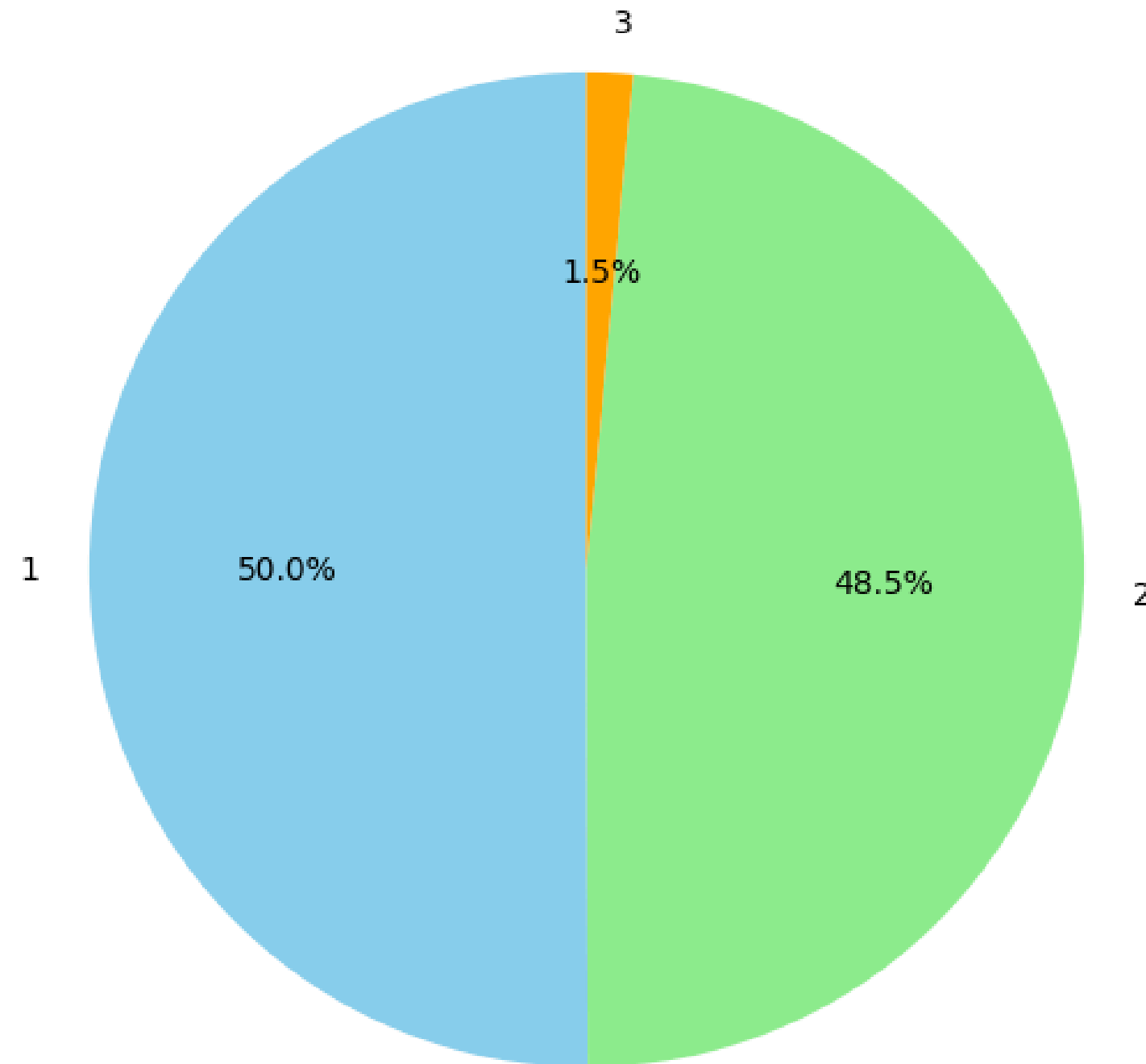
# ALGUNAS GRÁFICAS





# ALGUNAS GRÁFICAS

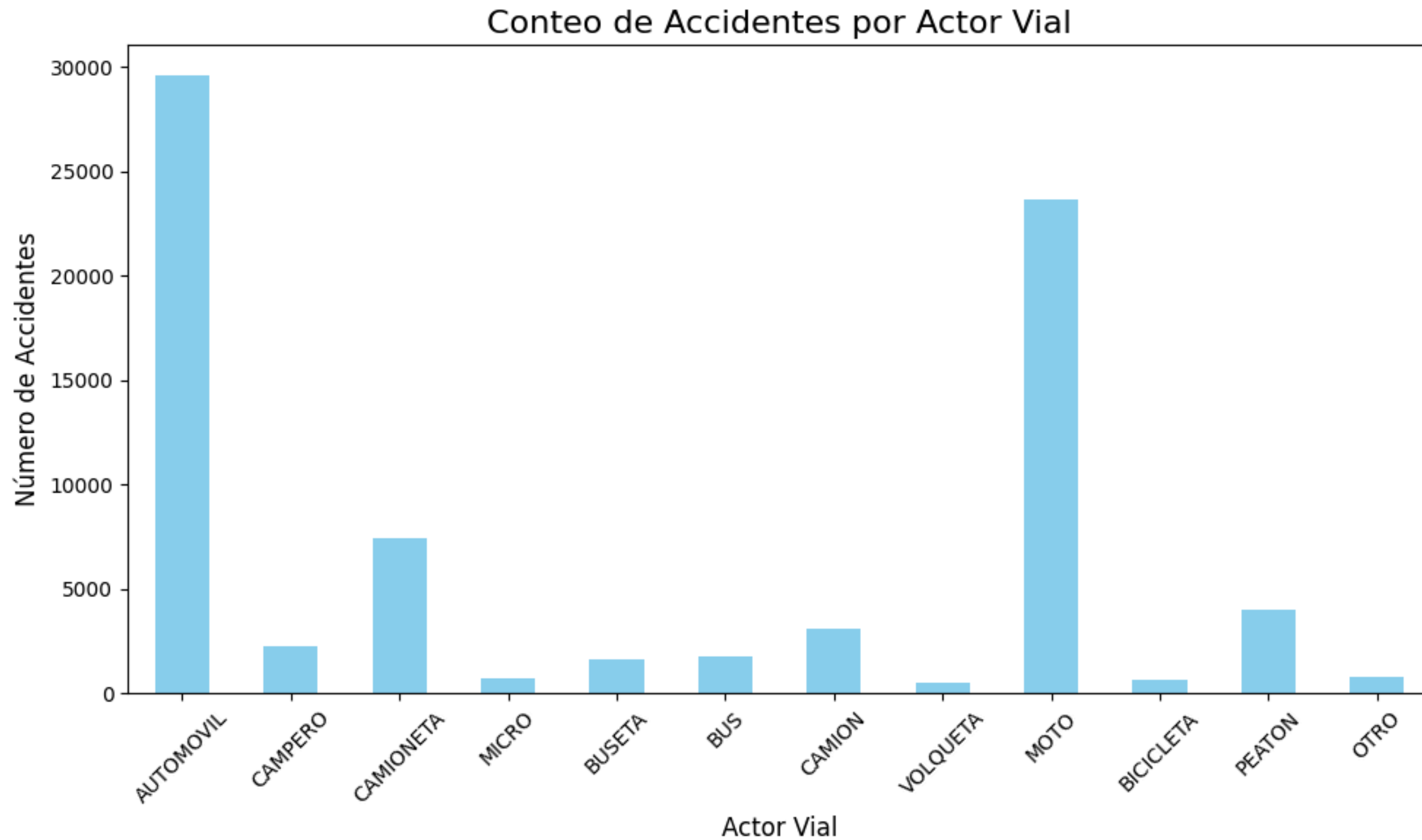
Cantidad de accidentes por gravedad



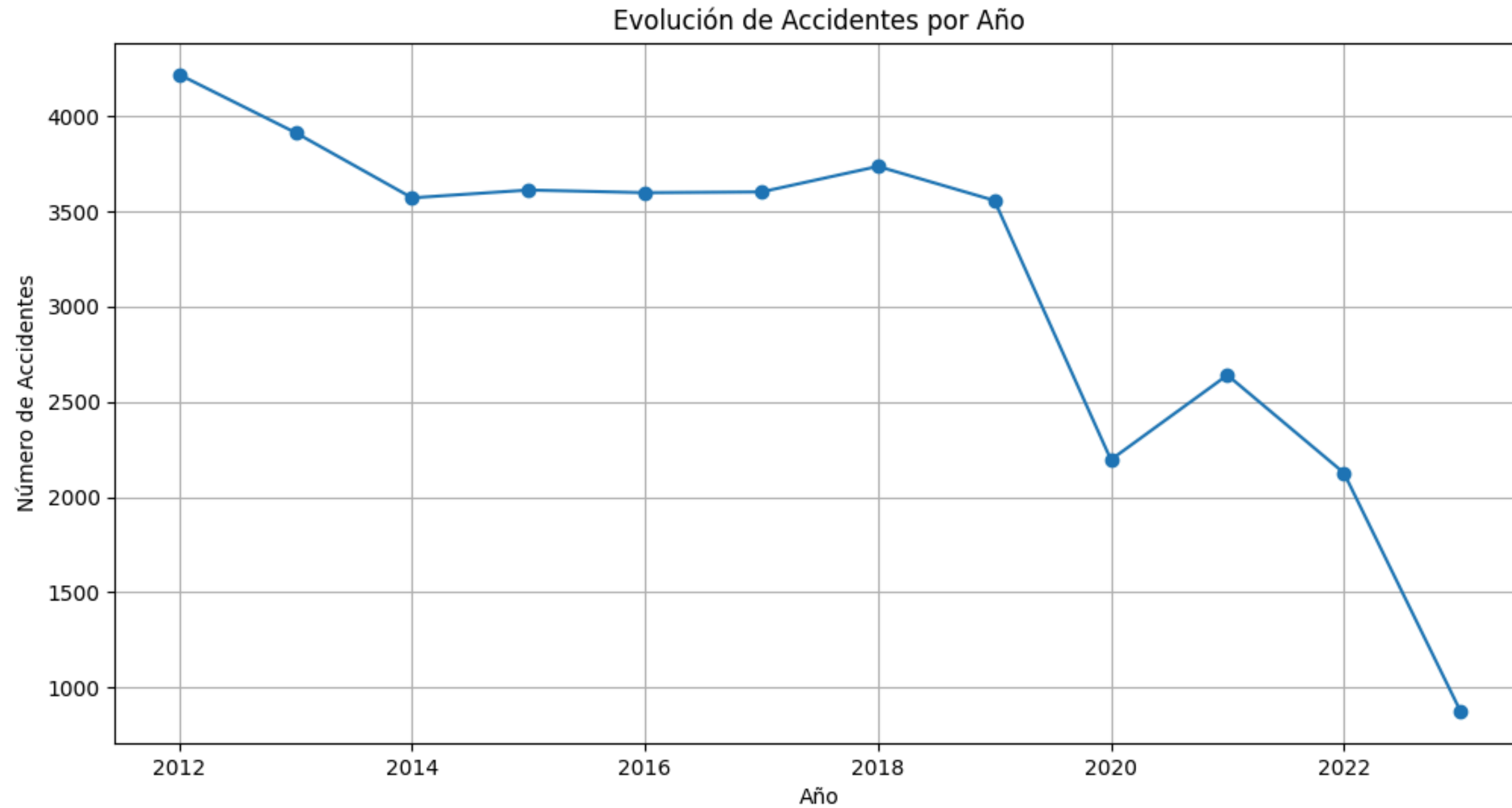
- Solo daños: 1
- Con heridos: 2
- Con muertos: 3



# ALGUNAS GRÁFICAS



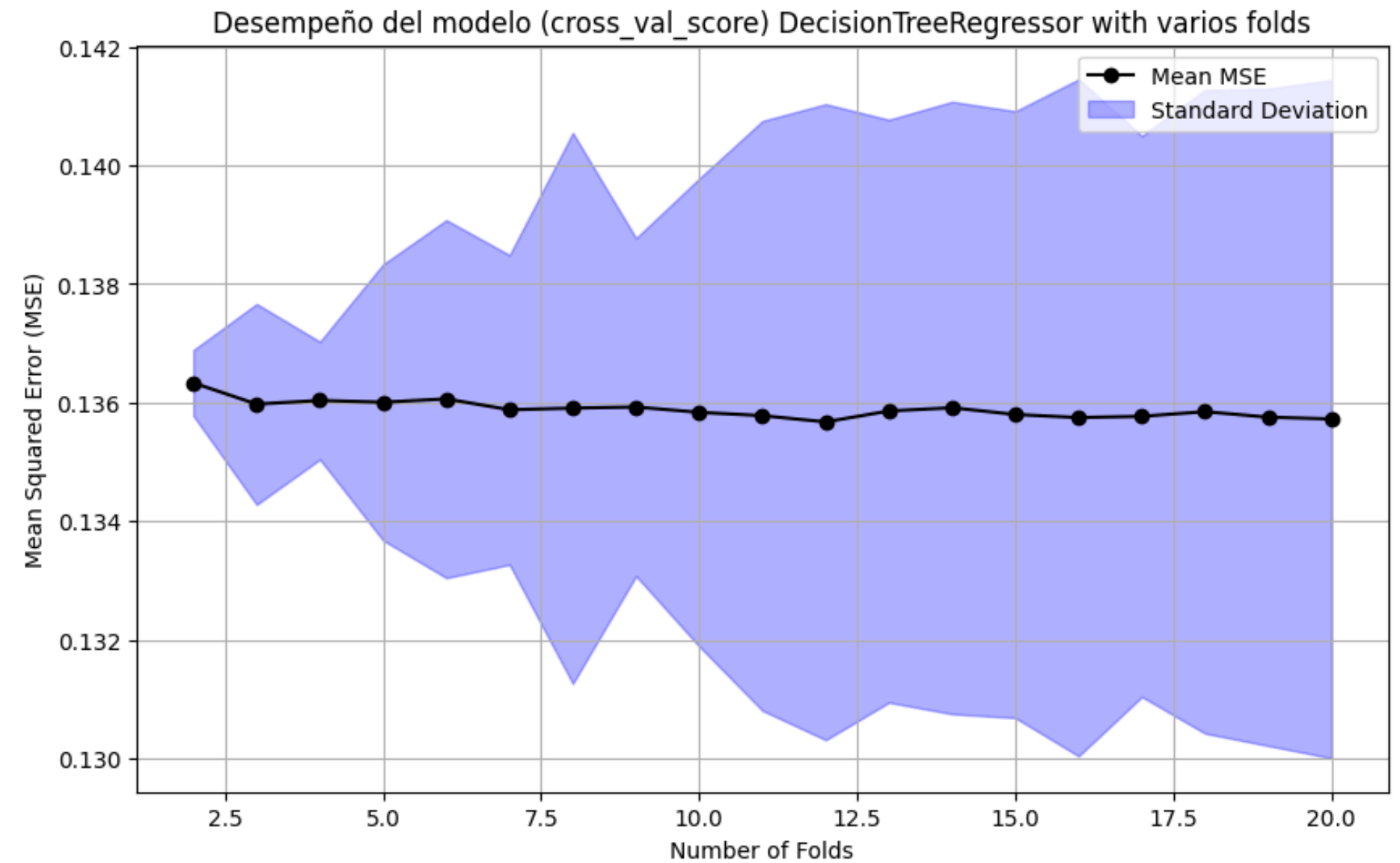
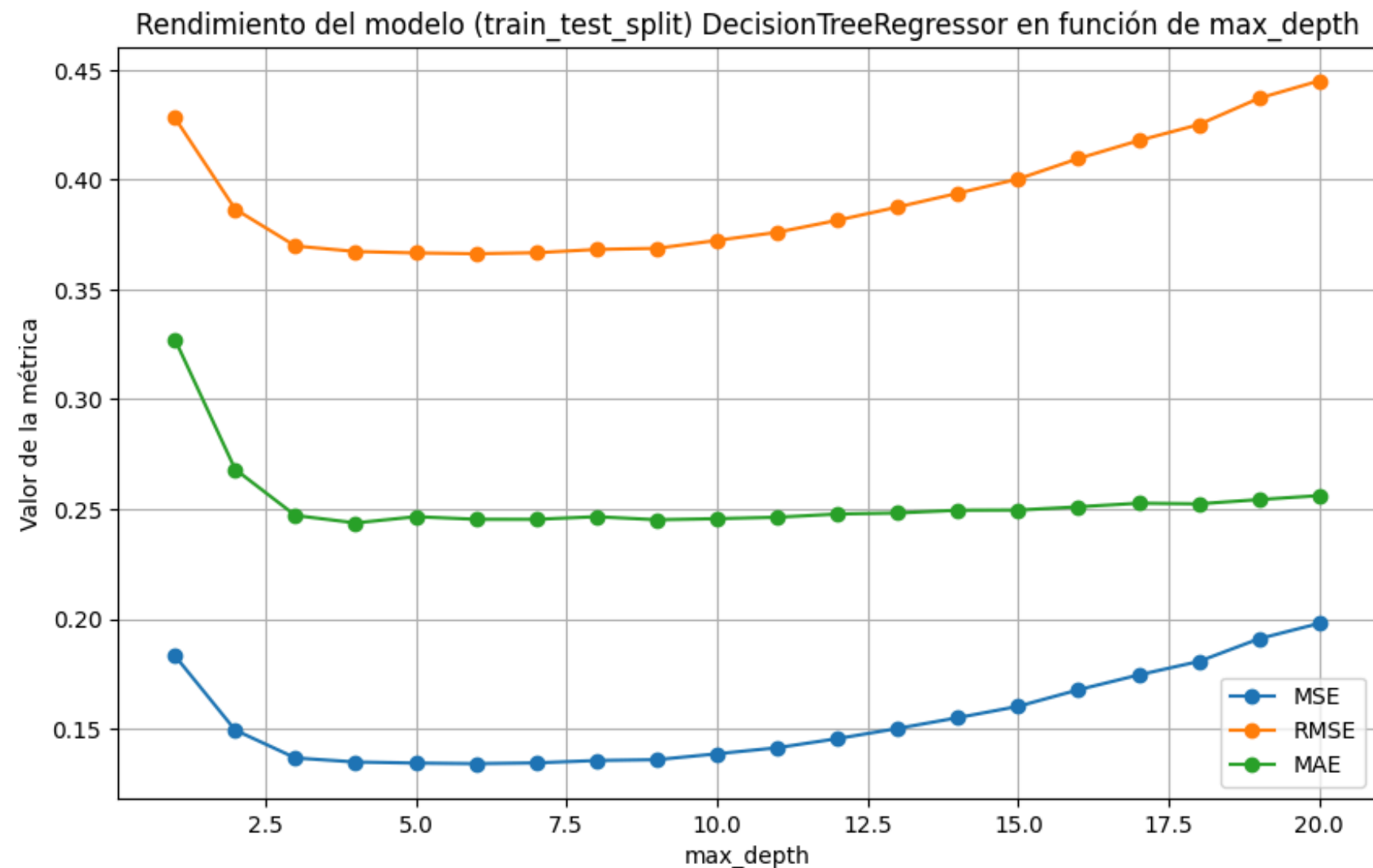
# ALGUNAS GRÁFICAS



# MODELOS USADOS

## DecisionTreeRegressor

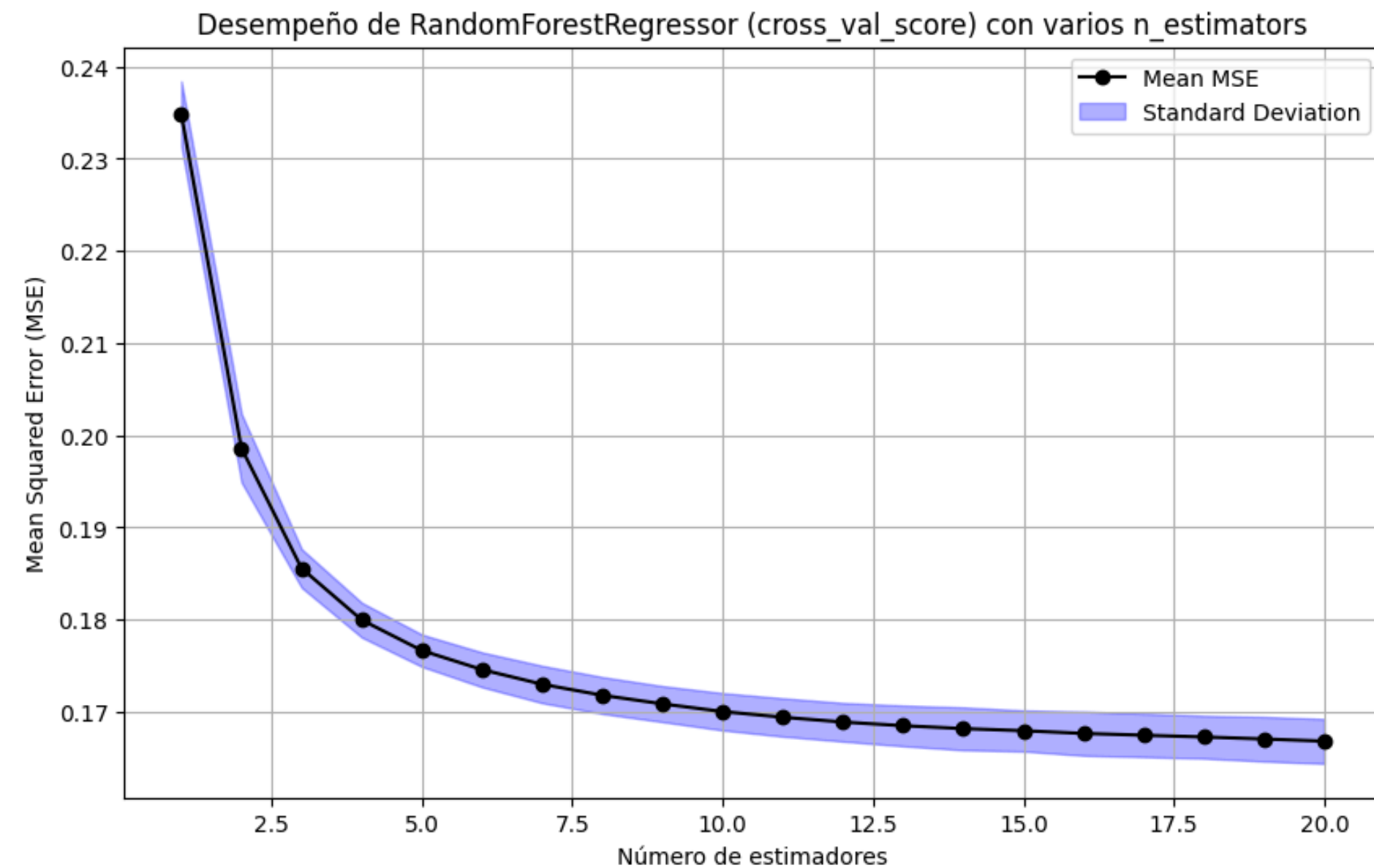
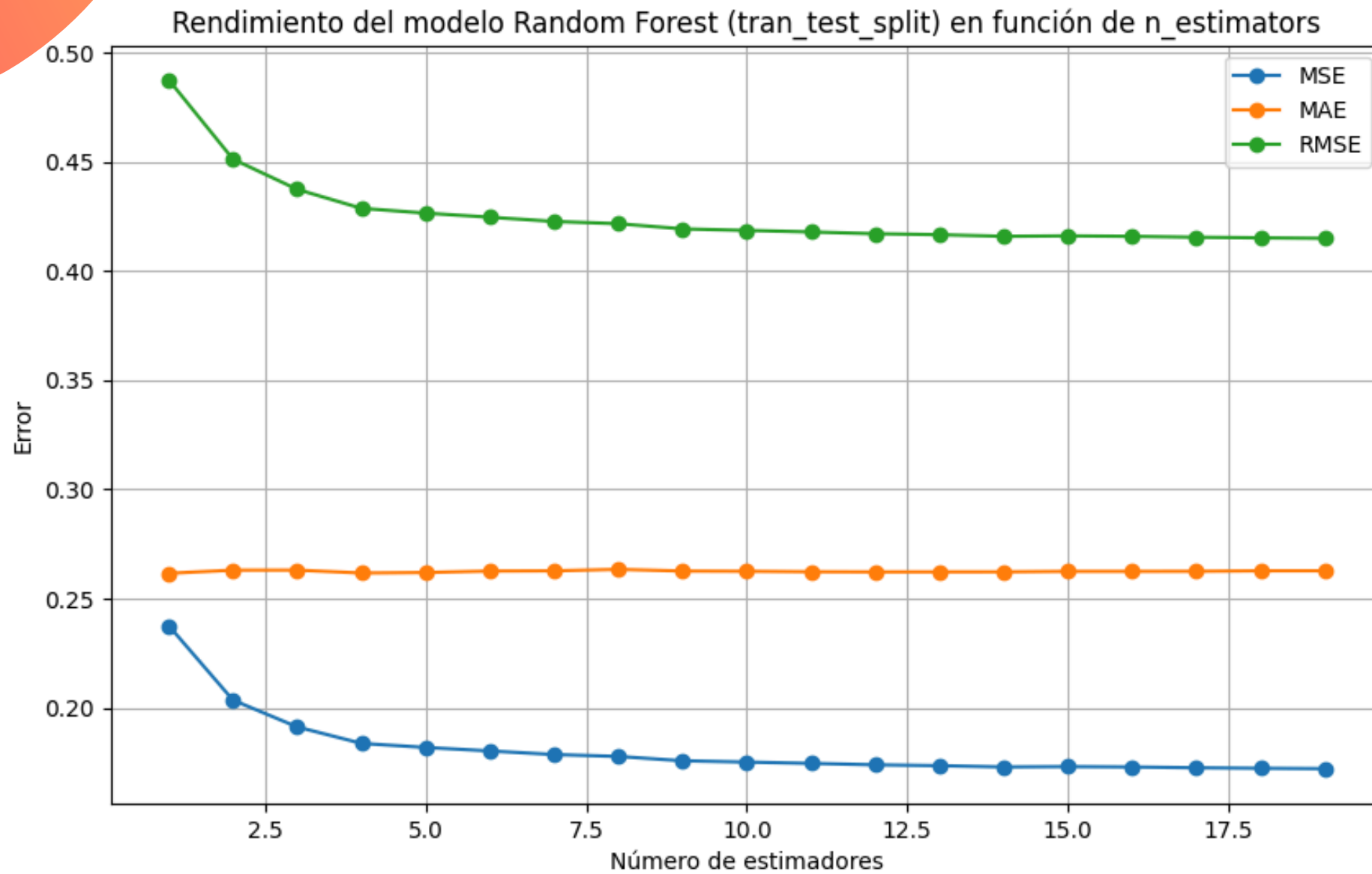
train\_test\_split vs cross\_val\_score



# MODELOS USADOS

## RandomForest

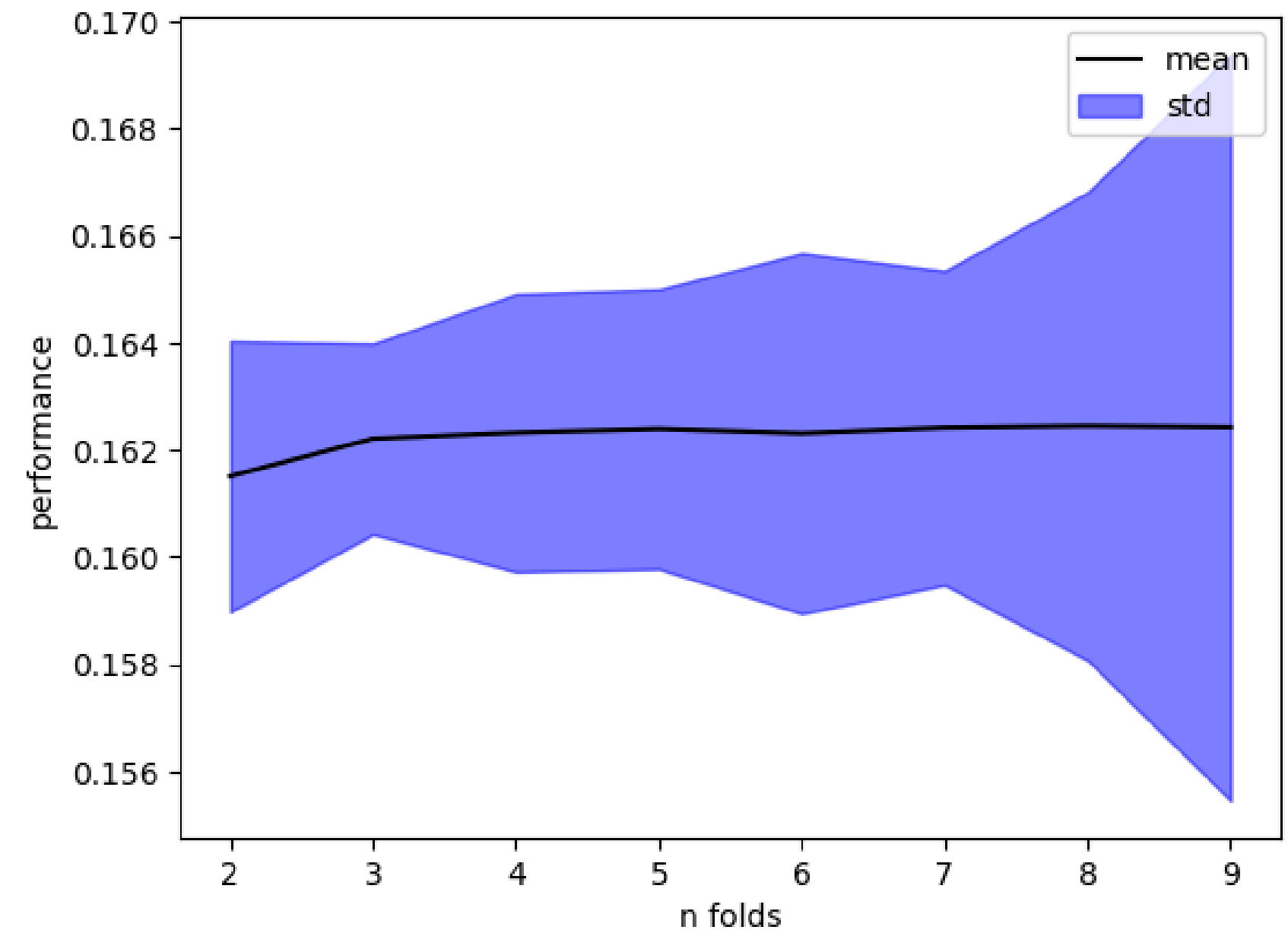
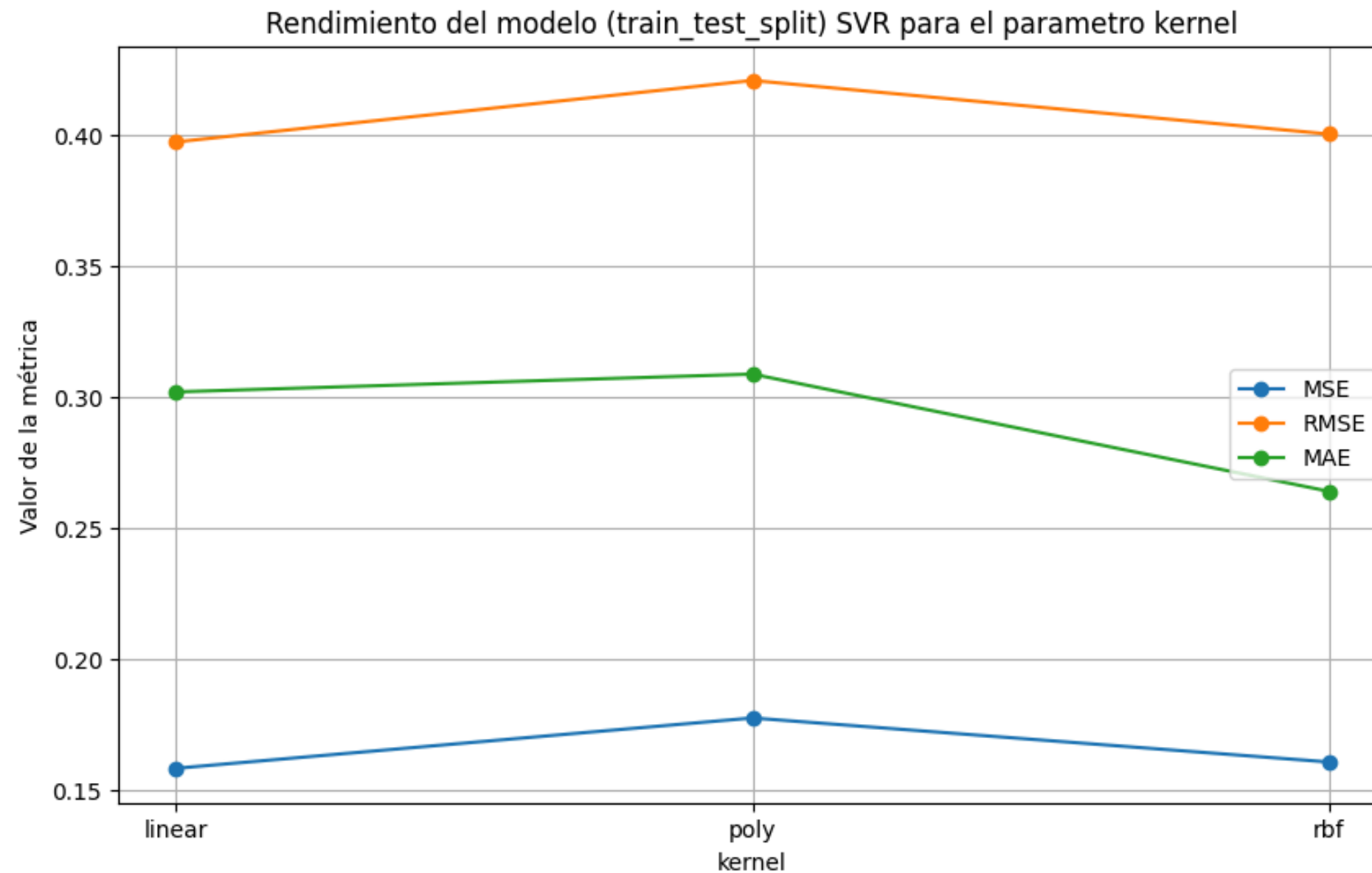
train\_test\_split vs cross\_val\_score



# MODELOS USADOS

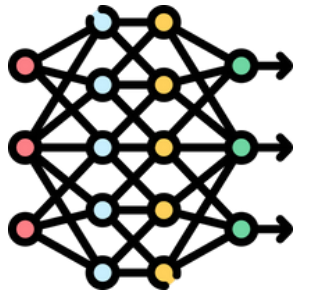
## SVR

train\_test\_split vs cross\_val\_score



**NOTA:** con el parámetro gamma, la ejecución demora mucho tiempo, por lo que no se tienen resultados

# REDES NEURONALES DENSAS

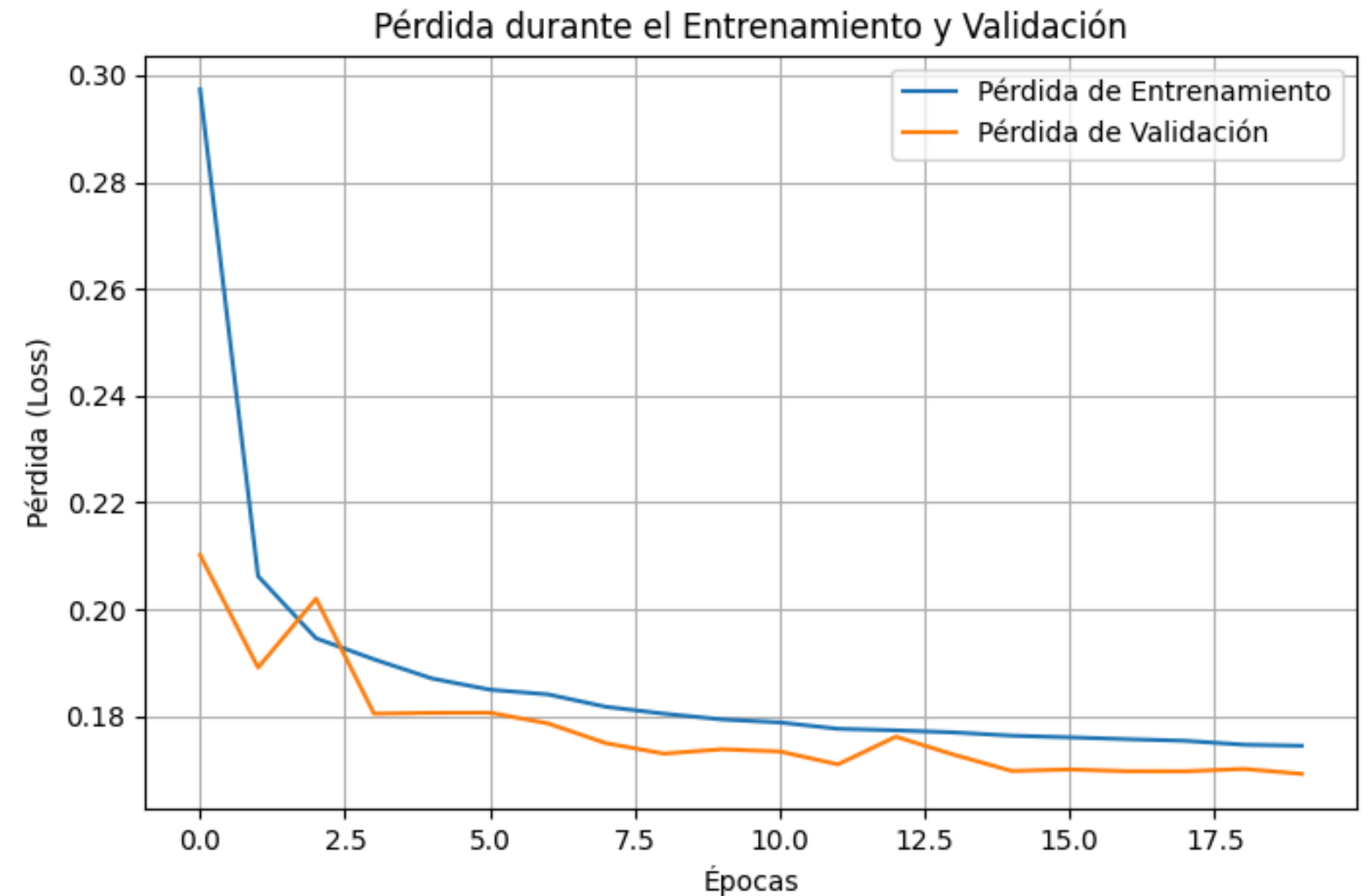


Con 3 capas ocultas

```
19 model = tf.keras.Sequential([
20     tf.keras.layers.InputLayer(input_shape=(X_train.shape[1],)),
21     tf.keras.layers.Dense(32, activation='relu'),
22     tf.keras.layers.Dense(64, activation='relu'),
23     tf.keras.layers.Dense(64, activation='relu'),
24     tf.keras.layers.Dense(1, activation='linear')
25 ])
```

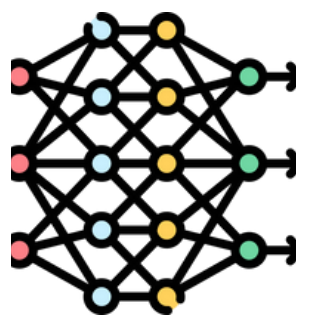
Mean Absolute Error en los valores continuos: 0.1734  
Mean Absolute Error en los valores redondeados: 0.1707

	Real	Predicción Continua	Predicción Redondeada
0	2	2.000021	2.0
1	1	1.989504	2.0
2	1	1.994445	2.0
3	1	1.001891	1.0
4	2	1.995798	2.0
5	1	1.001681	1.0
6	2	2.001130	2.0
7	2	1.992389	2.0
8	1	1.992027	2.0
9	3	2.026539	2.0
10	2	1.994155	2.0
11	1	1.982649	2.0
12	2	1.999361	2.0
13	2	2.000268	2.0
14	1	2.000596	2.0
15	1	1.001674	1.0
16	1	1.003576	1.0
17	2	2.000260	2.0
18	1	1.002035	1.0
19	2	1.999476	2.0





# REDES NEURONALES DENSAS

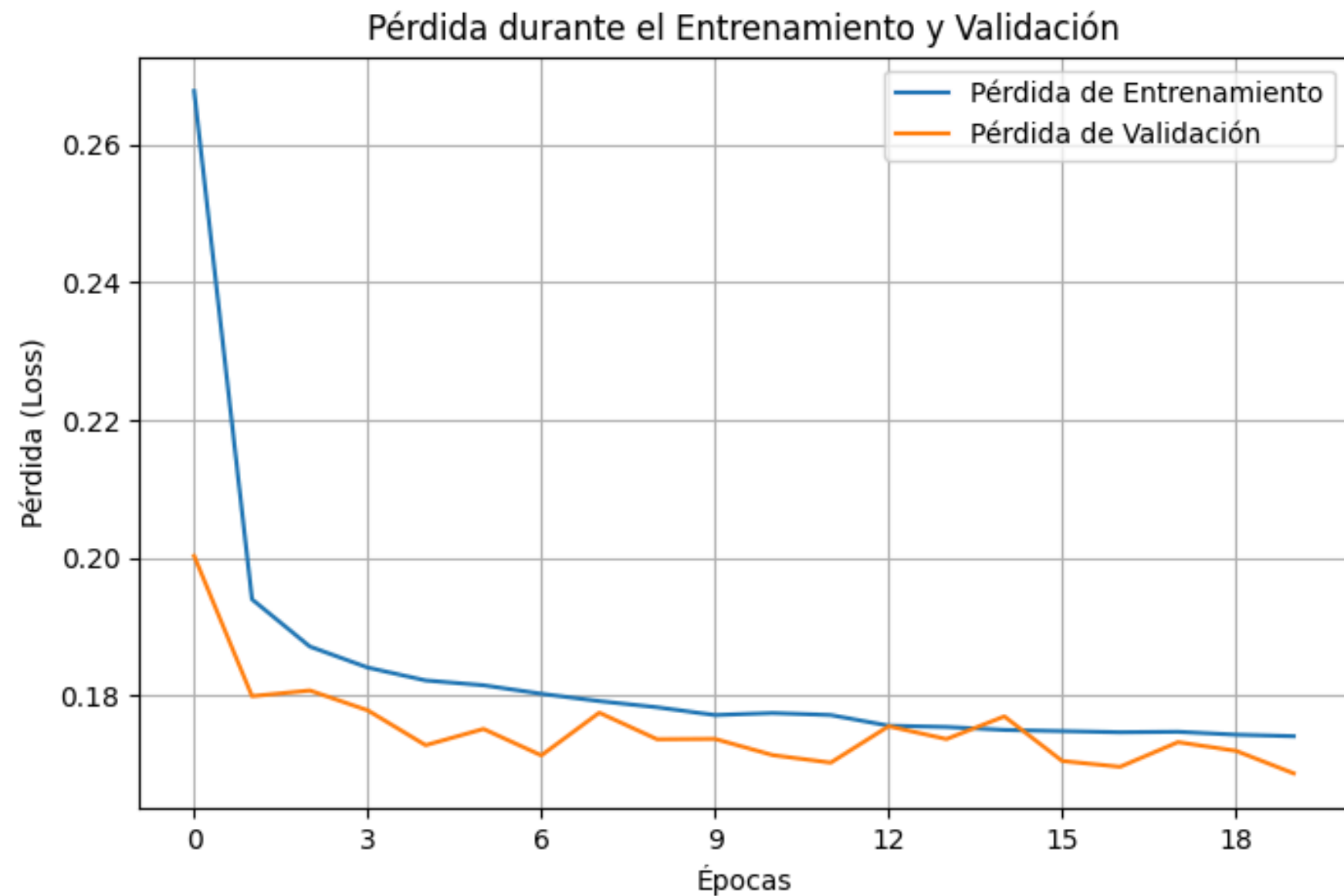


Con 6 capas ocultas

```
19 model = tf.keras.Sequential([
20     tf.keras.layers.InputLayer(input_shape=(X_train.shape[1],)),
21     tf.keras.layers.Dense(32, activation='relu'),
22     tf.keras.layers.Dense(64, activation='relu'),
23     tf.keras.layers.Dense(128, activation='relu'),
24     tf.keras.layers.Dense(64, activation='relu'),
25     tf.keras.layers.Dense(32, activation='relu'),
26     tf.keras.layers.Dense(16, activation='relu'),
27     tf.keras.layers.Dense(1, activation='linear')
28 ])
```

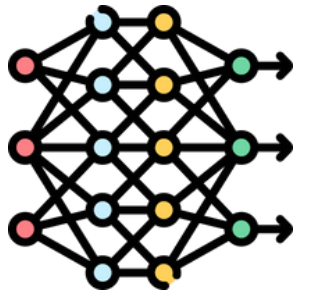
Mean Absolute Error (MAE) en valores continuos: 0.1719  
Mean Absolute Error (MAE) con valores redondeados: 0.1707

	Real	Predicción Continua	Predicción Redondeada
0	2	1.993756	2.0
1	1	1.994242	2.0
2	1	1.993515	2.0
3	1	0.999884	1.0
4	2	1.996244	2.0
5	1	0.999562	1.0
6	2	1.998816	2.0
7	2	1.998164	2.0
8	1	1.995112	2.0
9	3	2.001048	2.0
10	2	1.995347	2.0
11	1	2.001833	2.0
12	2	1.996393	2.0
13	2	1.998223	2.0
14	1	1.997386	2.0
15	1	0.999694	1.0
16	1	0.999464	1.0
17	2	1.998297	2.0
18	1	0.999483	1.0
19	2	2.001073	2.0



# REDES NEURONALES DENSAS

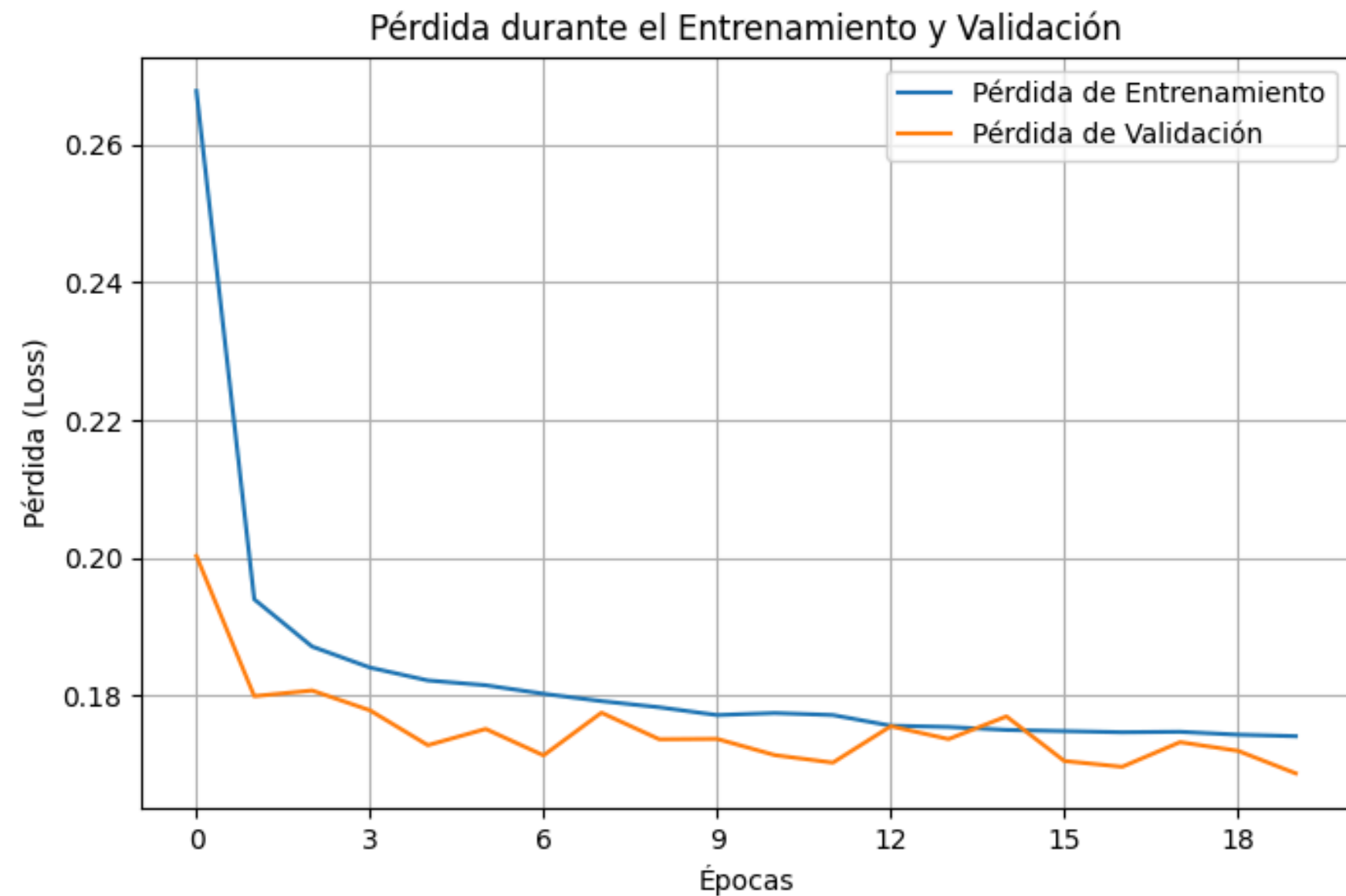
## Con 10 capas ocultas



```
18 model = tf.keras.Sequential([
19     tf.keras.layers.InputLayer(input_shape=(X_train.shape[1],)),
20     tf.keras.layers.Dense(16, activation='relu'),
21     tf.keras.layers.Dense(32, activation='relu'),
22     tf.keras.layers.Dense(64, activation='relu'),
23     tf.keras.layers.Dense(128, activation='relu'),
24     tf.keras.layers.Dense(128, activation='relu'),
25     tf.keras.layers.Dense(64, activation='relu'),
26     tf.keras.layers.Dense(32, activation='relu'),
27     tf.keras.layers.Dense(16, activation='relu'),
28     tf.keras.layers.Dense(8, activation='relu'),
29     tf.keras.layers.Dense(4, activation='relu'),
30     tf.keras.layers.Dense(1, activation='linear')
31 ])
```

Mean Absolute Error (MAE) en valores continuos: 0.1736  
Mean Absolute Error (MAE) con valores redondeados: 0.1718

	Real	Predicción Continua	Predicción Redondeada
0	2	1.989275	2.0
1	1	1.985200	2.0
2	1	1.994732	2.0
3	1	1.001464	1.0
4	2	1.989411	2.0
5	1	1.001492	1.0
6	2	1.997679	2.0
7	2	1.996438	2.0
8	1	1.987274	2.0
9	3	1.994538	2.0
10	2	1.990374	2.0
11	1	1.987793	2.0
12	2	1.995396	2.0
13	2	1.999130	2.0
14	1	1.999094	2.0
15	1	1.001473	1.0
16	1	1.001464	1.0
17	2	1.999227	2.0
18	1	1.001456	1.0
19	2	1.999421	2.0



# CONCLUSIONES

A pesar de que lo ideal sea que el dataset contenga variables relacionadas a las personas involucradas en cualquier tipo de accidente para dar una mejor solidez a la predicción, el modelo está haciendo en su mayoría predicciones que no están lejanas al contexto del problema y es un modelo que a futuro está abierto a muchas mejoras si se quiere analizar la accidentalidad en Bucaramanga.