
# Análisis del sistema de reparación y mantenimiento de acueducto: Área metropolitana del Valle de Aburrá

Integrantes: Kevin Andres Pineda Sarmiento

# Análisis del Sistema de Reparación y Mantenimiento de Acueducto: Área Metropolitana Valle de Aburrá

**Resumen:** Análisis de las interrupciones del servicio de acueducto en el área metropolitana del Valle de Aburrá. Este análisis comprende datos que corresponden a las interrupciones que ha realizado EPM durante el 2019 hasta la semana 18 del 2022. El set de datos cuenta con información acerca del motivo, impacto, contratista, duración y localización exacta de los sectores involucrados.

# Objetivos del proyecto

Basado en la información pública de intervenciones al acueducto y alcantarillado en toda el área metropolitana del Valle de Aburrá por parte de EPM, se tiene como objetivo:

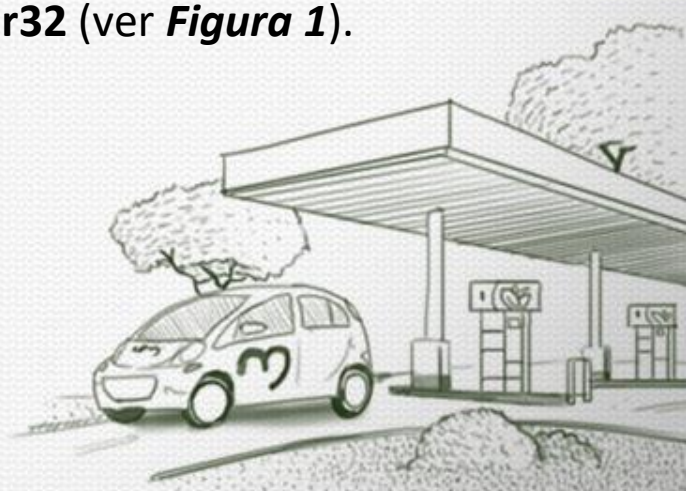
- Realizar un estudio descriptivo para organizar, depurar y presentar los datos de forma impoluta.
- Efectuar un reporte del tiempo promedio que requieren las intervenciones desglosado por las principales variables predictoras.
- Hacer una comparación de los resultados por sectores (municipio, circuito y barrio) y definir si existe correlación con alguna variable socioeconómica.
- Calcular los tiempos de intervención para cada municipio y sector del que se tenga reporte: valor máximo, mínimo, promedio y varianza.
- Definir una estrategia de visualización de los datos que de soporte a los hallazgos obtenidos y a las conclusiones del proceso.

# Proceso de codificación y análisis

- **Preprocesado de la base de datos:**

Primeramente, se importaron las librerías requeridas para el proyecto (pandas, numpy, math, seaborn, etc.).

Seguidamente, se cargaron los datos directamente desde la página web www.datos.gov.co, posteriormente, se reemplazaron y eliminaron ciertos registros para mantener una lógica estructural de la base de datos. Asimismo, se modificaron las columnas **Fecha de registro, Inicio y Fin** a un tipo de datos **Date time**, de igual forma las variables Horas a **float32** y Número de instalaciones a **integer32** (ver ***Figura 1***).



# Proceso de codificación y análisis

```
[1]  1 import pandas as pd
     2 import math
     3 import numpy as np
     4 import seaborn as sns
     5 import matplotlib.pyplot as plt
     6 import statistics as sta
     7 import requests
     8 import json
     9 import folium
    10 from folium.plugins import HeatMap
```

```
[2]  1 epm = pd.read_csv('https://www.datos.gov.co/api/views/r9fv-awbc/rows.csv?accessType=DOWNLOAD')
     2 epm['Motivo'].replace({'0':'#No registra motivo#'}, inplace = True)
```

```
[3]  1 epm['Fecha de registro'] = pd.to_datetime(epm['Fecha de registro'], errors='ignore', dayfirst=False)
     2 epm['Inicio'] = pd.to_datetime(epm['Inicio'], errors='ignore', dayfirst=False)
     3 epm['Fin'] = pd.to_datetime(epm['Fin'], errors='ignore', dayfirst=False)
     4 epm['Fecha y hora esperada'] = pd.to_datetime(epm['Fecha y hora esperada'], errors='ignore', dayfirst=False)
```

```
[4]  1 epm.drop((epm[(epm['Horas'] == '2020-12-22T04:00:00.000') | (epm['Horas'] == '2020-12-22T11:00:00.000') | (epm['Horas'] == '2020-12-23T06:00:00.000')
     2 | (epm['Horas'] == '2020-12-22T06:00:00.000')].index), axis = 0, inplace = True)
```

```
[5]  1 epm.drop(epm[(epm['Fecha de registro'] == '05/21/2012')].index, axis = 0, inplace = True)
     2 epm.drop(epm[(epm['Impacto'] == 'S16')].index, axis = 0, inplace = True)
     3 epm['Número de instalación'].replace({'-':0}, inplace = True)
```

```
[6]  1 epm = epm.astype({'Horas' : 'float32', 'Número de instalación' : 'int32'}
```

**Figura 1.** Codificación referente al preprocesado de los datos.

# Proceso de codificación y análisis

- **Exploración de datos:**

Para iniciar la exploración de los datos, se ajustó un histograma a la variable respuesta (**Horas**) con la finalidad de alcanzar una primera visión general. Con base a este gráfico, se puede inferir que la duración de las intervenciones oscilaba mayoritariamente entre 4 a 10 horas (ver ***Figura 2***).

Seguidamente, se dio paso a la inspección de los valores faltantes, tipos de las variables predictoras y la exploración general de las variables categóricas y numéricas (ver ***Figura 3***).

De igual forma, se llevó a cabo un análisis de correlación entre la variable **Número de instalaciones** (única variable predictora de carácter numérico) y la variable respuesta **Horas**, donde se pudo apreciar que no hay ningún tipo de correlación entre ambas variables (ver ***Figura 4 y 5***).

# Proceso de codificación y análisis

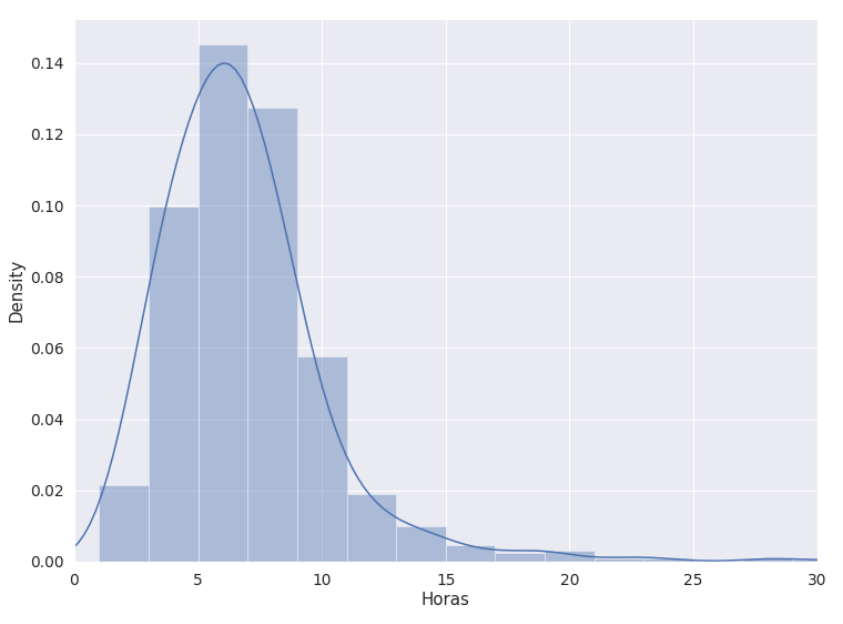



**Figura 2.** Histograma de frecuencia de la variable respuesta (**Horas**).

```
1 NA = epm.isna().sum() #Identificar valores NA
2 NA[NA!=0] #Solamente la columna 'Nombre del contratista' presenta valores faltantes (Solo 3)
```

```
1 epm.dtypes
```

```
1 epm.info()
```

```
1 epm._get_numeric_data().describe().T
```

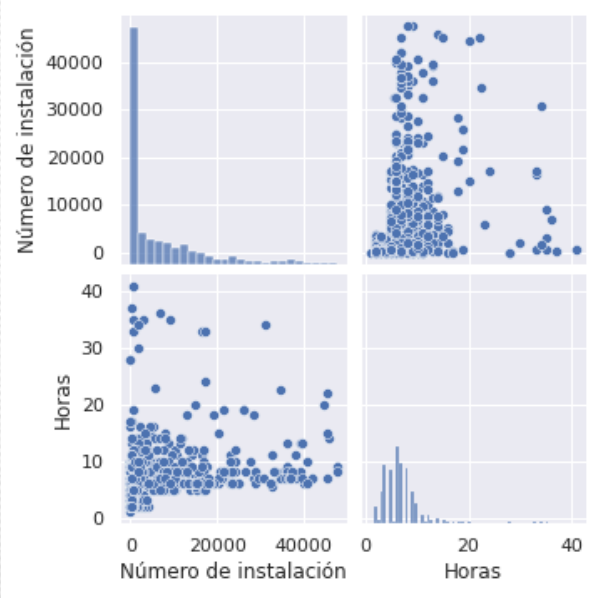
**Figura 3.** Codificación análisis exploratorio.

# Proceso de codificación y análisis

```
[10]  1 cols = ['Número de instalación', 'Horas']
      2 sns.set()
      3 sns.pairplot(epm[cols]);
```

```
1 cm = np.corrcoef(epm[cols].values.T)
2 f, ax = plt.subplots(figsize=(12, 9))
3 sns.set(font_scale=1.25)
4 hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size': 10}, yticklabels=cols, xticklabels=cols)
5 plt.show()
```

**Figura 4.** Análisis de correlación entre la variable **Número de instalaciones** y **Horas.**



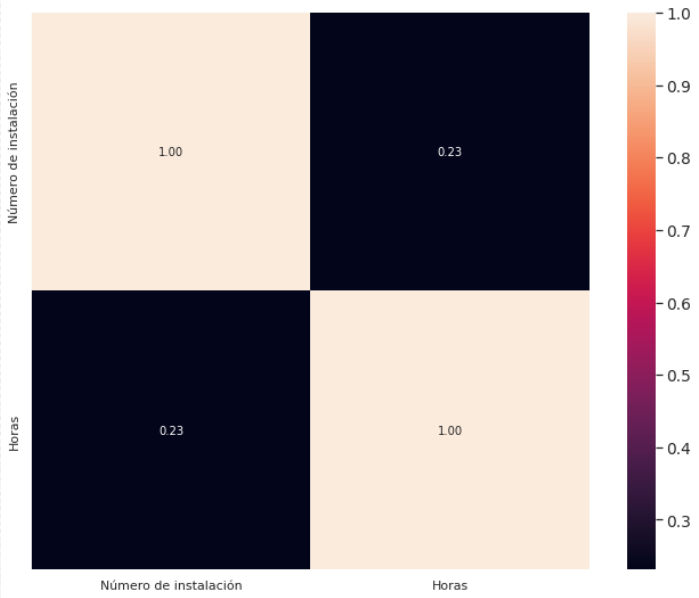



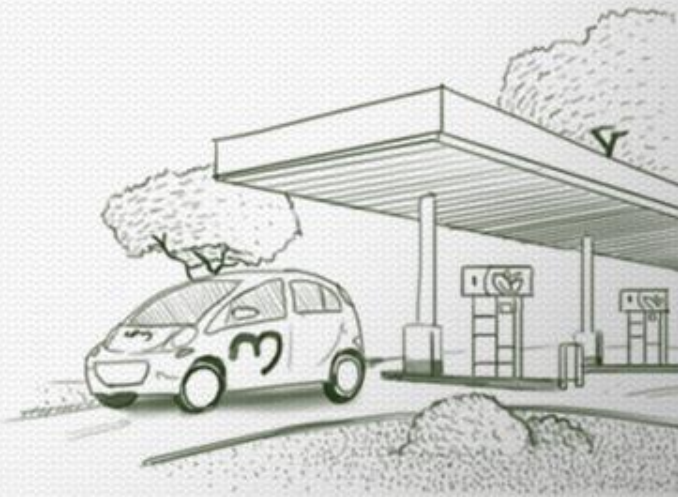

**Figura 5.** Análisis de correlación entre la variable **Número de instalaciones** y **Horas.**

# Proceso de codificación y análisis

- **Análisis de variables categóricas:**

En este apartado se realizó un estudio de las variables categóricas del Data set, se analizaron los valores únicos de estas y finalmente se graficaron las principales variables (**Impacto, Motivo, Municipio Y Nombre del responsable**) en histogramas y diagramas de barras para conocer la distribución del tiempo empleado en las intervenciones y observar su frecuencia dentro del set de datos (ver ***Figura 6 y 7***).

Por otro lado, la variable categórica **Nombre de contratistas** se analizó exclusivamente, ya que está presenta un número considerable de valores únicos, por tanto, solo se graficaron los 6 mayores contratistas según su frecuencia dentro del set de datos (ver ***Figura 8 y 9***).



# Proceso de codificación y análisis

```
[17]  1 disc_col = [i for i in epm.columns if not i in epm._get_numeric_data()]
      2 print (disc_col)
```

```
1 disc_col_net = ['Impacto', 'Motivo', 'Municipio', 'Nombre del responsable']
2 for i in disc_col_net:
3     print("%10s"%i, np.unique(epm[i].dropna()))
```

```
1  plt.figure(figsize=(40,20))
2  for i,c in enumerate(disc_col_net):
3      plt.subplot(2,4,i+1)
4      k = epm[[c,"Horas"]].dropna()
5      for v in epm[c].dropna().unique():
6          sns.distplot(k.Horas[k[c]==v], label=v);
7          plt.title(c)
8      plt.yticks([])
9      plt.legend()
10
11     plt.subplot(2,4,i+5)
12     vc = k[c].value_counts()
13     sns.barplot(vc.index, vc.values)
14     plt.xticks(range(len(vc)), vc.index, rotation="vertical");
```

**Figura 6.** Codificación análisis de variables categóricas.

# Proceso de codificación y análisis

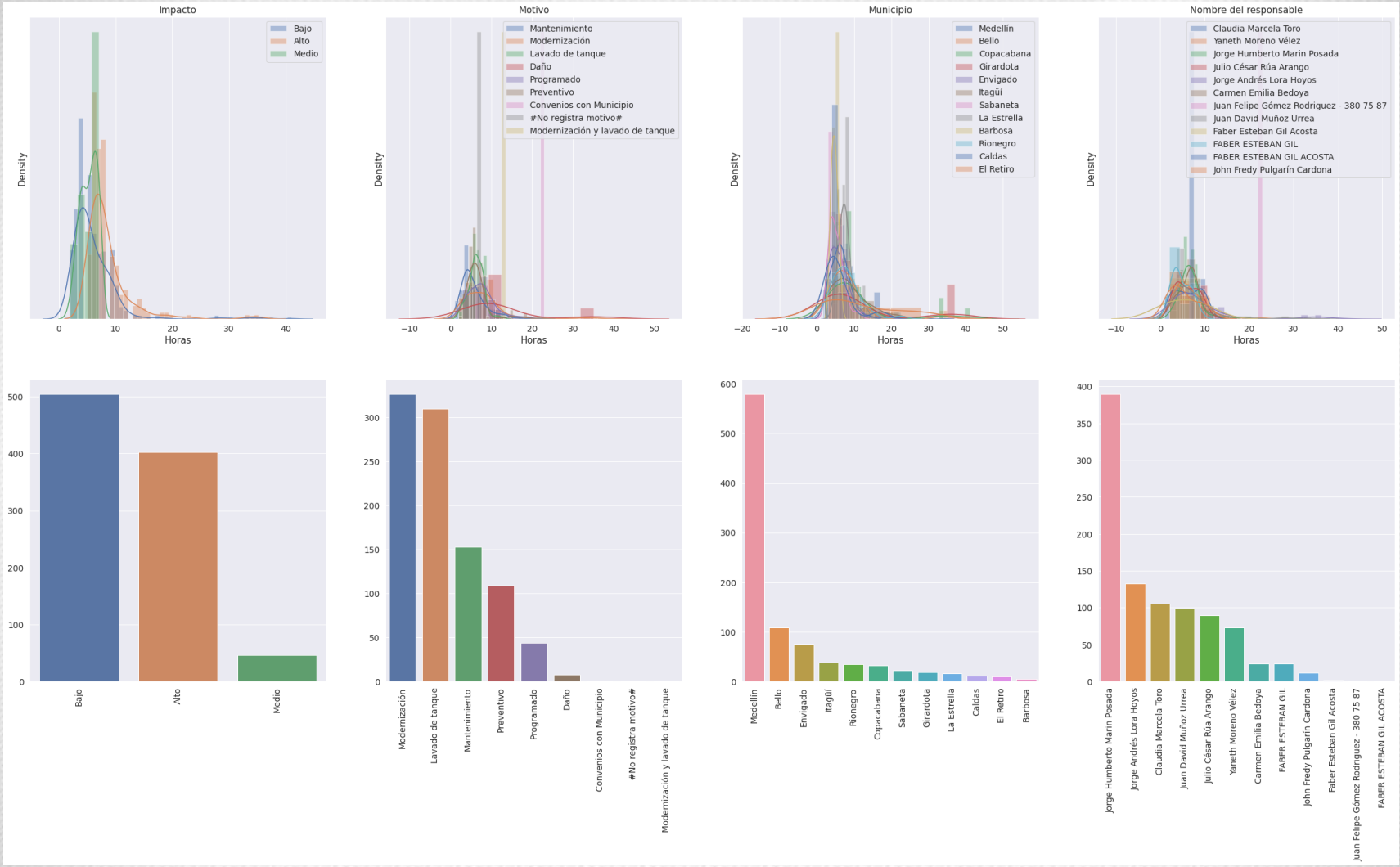



**Figura 7.** Análisis de variables categóricas.

# Proceso de codificación y análisis

```
[22]  1 top6_contratistas = epm['Nombre del contratista '].value_counts().head(6)
```

```
[23]  1 top6_contratistas
```

```
1 # sns.distplot(top6_contratistas, label=top6_contratistas.index)
2 plt.figure(figsize=(40,30))
3 plt.subplot(2,2,1)
4 for i in (list(top6_contratistas.index)):
5     sns.distplot(epm.Horas[epm['Nombre del contratista ']==i], label=i);
6 plt.title('Contratistas')
7 plt.legend(fontsize=20)
8 plt.subplot(2,2,2)
9 sns.barplot(top6_contratistas.index, top6_contratistas.values);
10 plt.xticks(range(len(top6_contratistas)), top6_contratistas.index, rotation="vertical");
```

**Figura 8.** Codificación análisis de variable categórica **Nombre del contratista.**

# Proceso de codificación y análisis

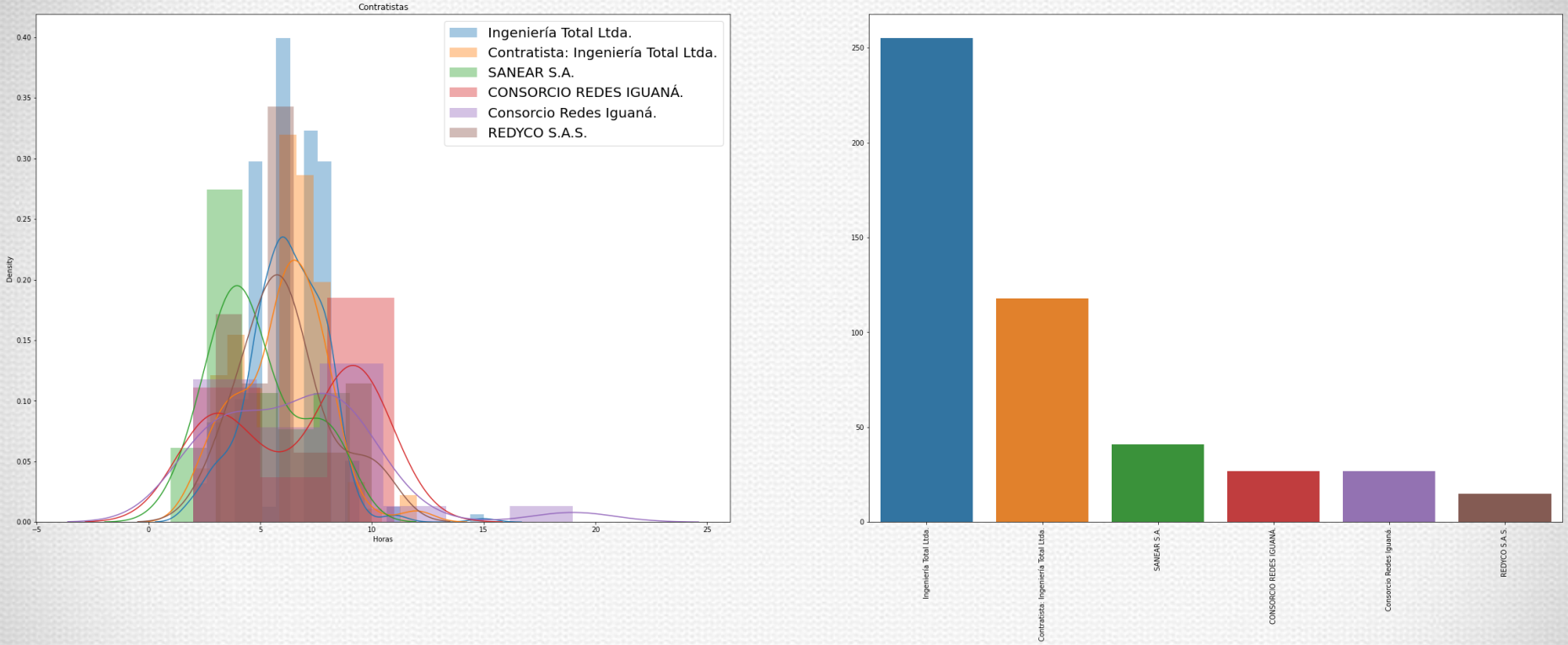
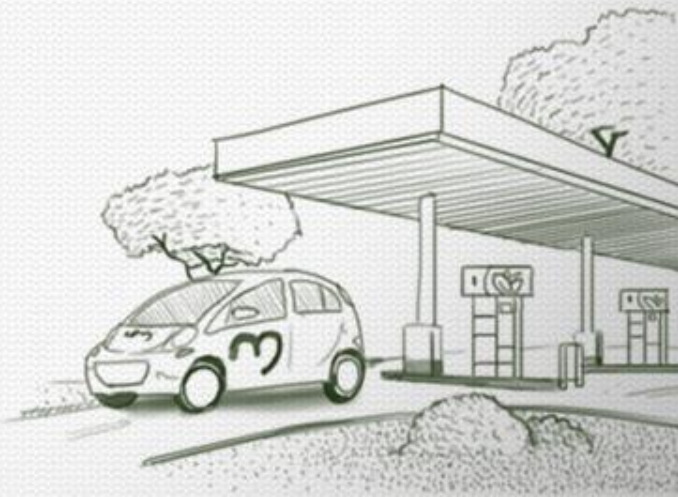



**Figura 9.** Análisis de variable categórica **Nombre del contratista.**

# Proceso de codificación y análisis

- **Punto 1:** Efectuar un reporte del tiempo promedio que requieren las intervenciones desglosado por las principales variables predictoras.

Se realizaron diversos reportes los cuales se agrupan según el promedio de la variable respuesta (**Horas**) y se desglosa conforme a las siguientes variables predictoras**: Mes de registro, Año de registro, Impacto y Motivo** (ver ***Figura 10 y 11***).



# Proceso de codificación y análisis

```
[ ]  1 epm_by_mes = epm.groupby(pd.PeriodIndex(epm['Fecha de registro'], freq='Q-DEC'), axis=0).mean()['Horas'].sort_index(axis = 0)
```

```
[ ]  1 fig, ax = plt.subplots()
     2 sns.barplot(epm_by_mes.index, epm_by_mes.values, palette="Blues_d")
     3 plt.xticks(range(len(epm_by_mes)), epm_by_mes.index, rotation="vertical")
     4 ax.plot(range(len(epm_by_mes)), epm_by_mes.values)
     5 plt.ylabel('Horas', fontsize=12)
     6 plt.xlabel('Fecha de registro', fontsize=12);
```

```
[ ]  1 epm_by_year = epm.groupby(pd.PeriodIndex(epm['Fecha de registro'], freq='Y'), axis=0).mean()['Horas'].sort_index(axis = 0)
```

```
[ ]  1 fig, ax = plt.subplots()
     2 sns.barplot(epm_by_year.index, epm_by_year.values, palette="Blues_d")
     3 plt.xticks(range(len(epm_by_year)), epm_by_year.index, rotation="vertical")
     4 ax.plot(range(len(epm_by_year)), epm_by_year.values)
     5 plt.ylabel('Horas', fontsize=12)
     6 plt.xlabel('Fecha de registro', fontsize=12);
```

```
[ ]  1 epm_by_impacto = epm.groupby('Impacto').mean()['Horas'].sort_values(ascending=False)
```

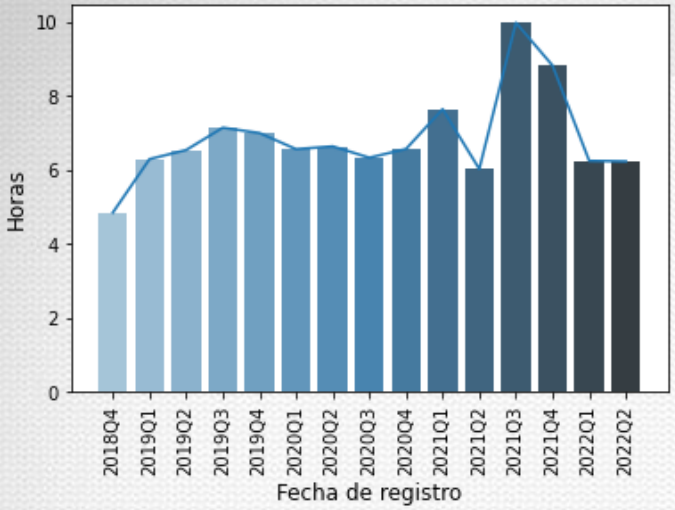
```
[ ]  1 fig, ax = plt.subplots()
     2 sns.barplot(epm_by_impacto.index, epm_by_impacto.values, palette="Blues_d")
     3 plt.xticks(range(len(epm_by_impacto)), epm_by_impacto.index, rotation="vertical")
     4 ax.plot(range(len(epm_by_impacto)), epm_by_impacto.values)
     5 plt.ylabel('Horas', fontsize=12)
     6 plt.xlabel('Fecha de registro', fontsize=12);
```

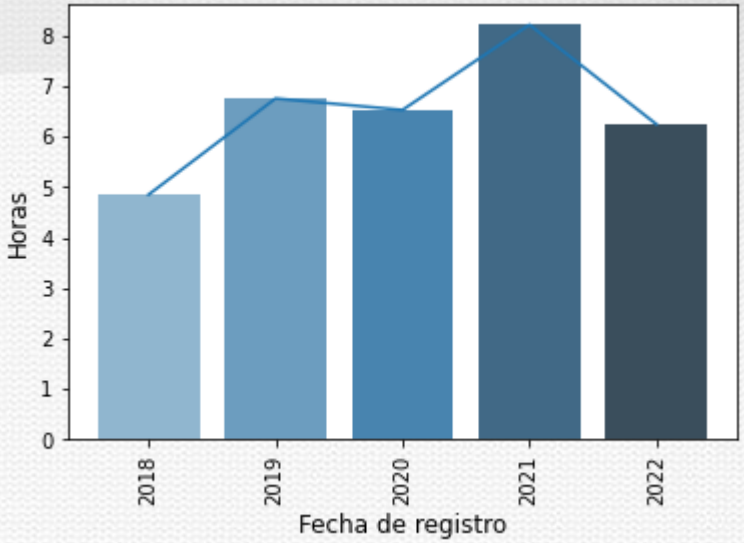
```
[ ]  1 epm_by_motivo = epm.groupby('Motivo').mean()['Horas'].sort_values(ascending=False)
```

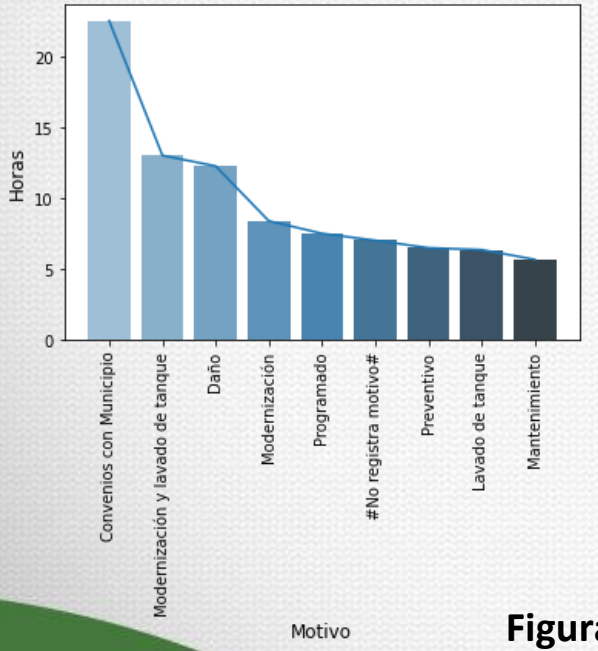
```
[ ]  1 fig, ax = plt.subplots()
     2 sns.barplot(epm_by_motivo.index, epm_by_motivo.values, palette="Blues_d")
     3 plt.xticks(range(len(epm_by_motivo)), epm_by_motivo.index, rotation="vertical")
     4 ax.plot(range(len(epm_by_motivo)), epm_by_motivo.values)
     5 plt.ylabel('Horas', fontsize=12)
     6 plt.xlabel('Motivo', fontsize=12);
```

**Figura 10.** Codificación Punto 1.

# Proceso de codificación y análisis



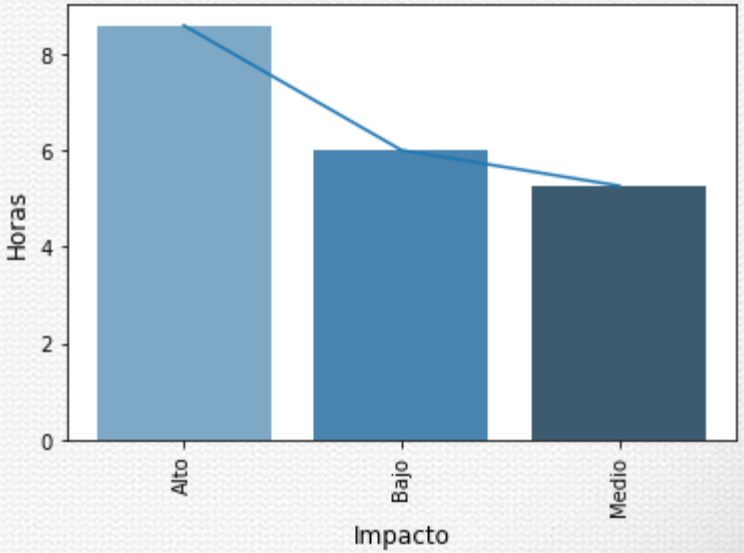




**Figura 11.** Reporte agrupado según el promedio de la variable respuesta y desglosado conforme a las principales variables predictoras.

# Proceso de codificación y análisis

- **Punto 2:** Hacer una comparación de los resultados por sectores (municipio, circuito y barrio) y definir si existe correlación con alguna variable socioeconómica.

Primeramente, se empleó un diagrama de caja para analizar la variable **Municipio**, ya que es la única variable referente a localización que presenta un número reducido de valores únicos. Basándonos en este gráfico, se puede afirmar que hay un valor considerable de intervenciones que requirieron un tiempo atípico, fácilmente identificadas, puesto que su duración se encuentra por encima del umbral máximo establecido. Por otro lado, el rango intercuartílico que exponen la mayor parte de los municipios es reducido, lo cual está relacionado con un nivel de dispersión bajo, es decir que, la duración de las intervenciones presenta (aproximadamente) un rango de tiempo estándar (ver ***Figura 12 y 13***).

Por otra parte, no fue posible establecer si existe correlación (desde un punto de vista técnico) entre la variable respuesta **Horas** y alguna variable socioeconómica, ya que, la base de datos entregada no presenta una variable numérica que detalle el nivel socioeconómico del **Circuito/Barrio** donde se realizó la intervención. No obstante, se optó por agrupar según el **Circuito/Barrio** el tiempo promedio de las intervenciones, tomando el top 6 de los mayores y menores tiempos promedios de intervención, esto se empleó con el fin de visualizar si existe algún tipo de relación entre el tiempo de intervención y el nivel socioeconómico del sector donde se llevó a cabo esta, teniendo como premisa el hecho de que potencialmente, en los sectores más marginados, el tiempo de intervención sería mayor. A continuación, se exponen los gráficos (ver ***Figura 14 y 15***).

# Proceso de codificación y análisis

```
[ ]  1 epm_by_municipio = epm.groupby('Municipio').mean()['Horas'].sort_values(ascending=False)
```

```
[ ]  1 plt.figure(figsize=(40,30))
     2 var = 'Municipio'
     3 data = pd.concat([epm['Horas'], epm[var]], axis=1)
     4 f, ax = plt.subplots(figsize=(8, 6))
     5 fig = sns.boxplot(x=var, y='Horas', data=data)
     6 fig.axis(ymin=0, ymax=40)
     7 plt.xticks(range(len(epm_by_municipio)), epm['Municipio'].unique(), rotation="vertical");
```

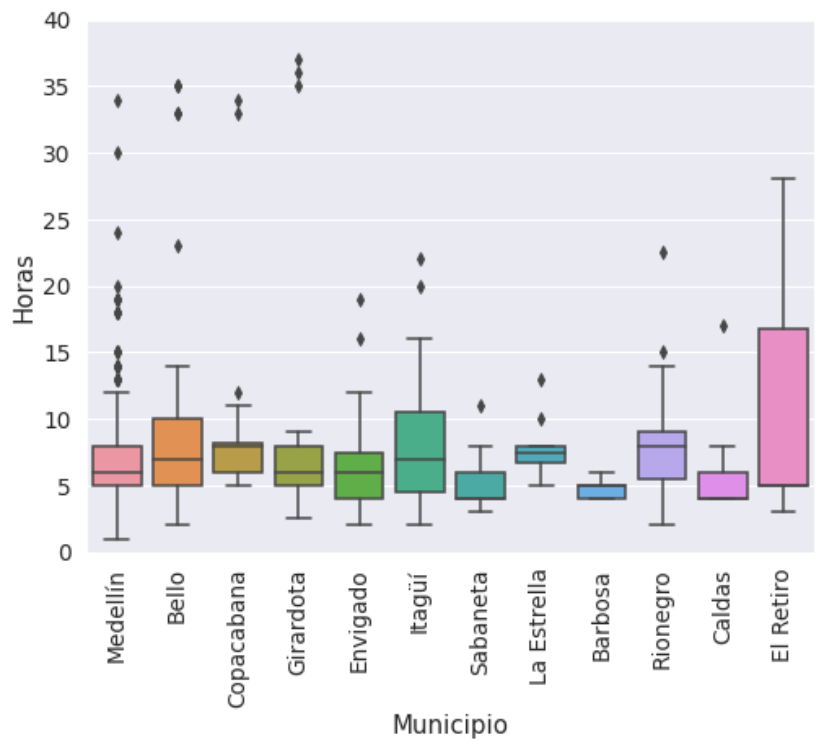
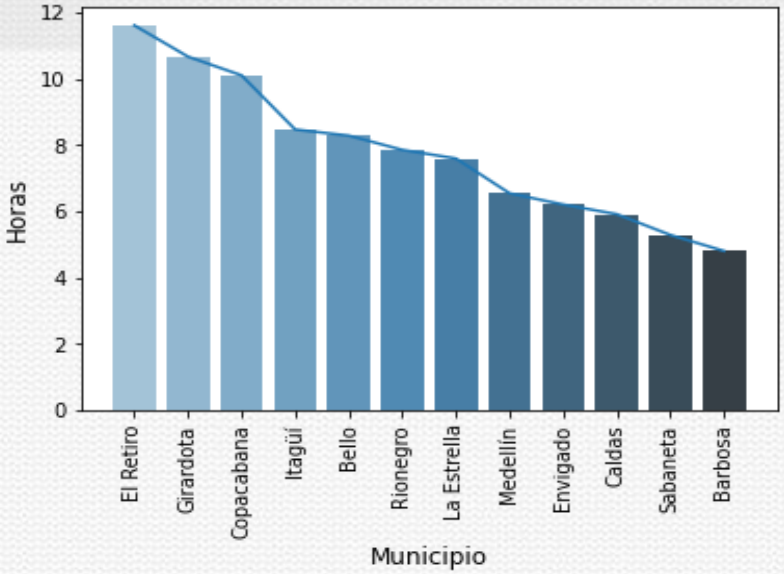



**Figura 12.** Codificación y diagrama de cajas para analizar la variable **Municipio**.

# Proceso de codificación y análisis

**Figura 13. T**iempos promedios de intervención desglosado por **Municipio**.



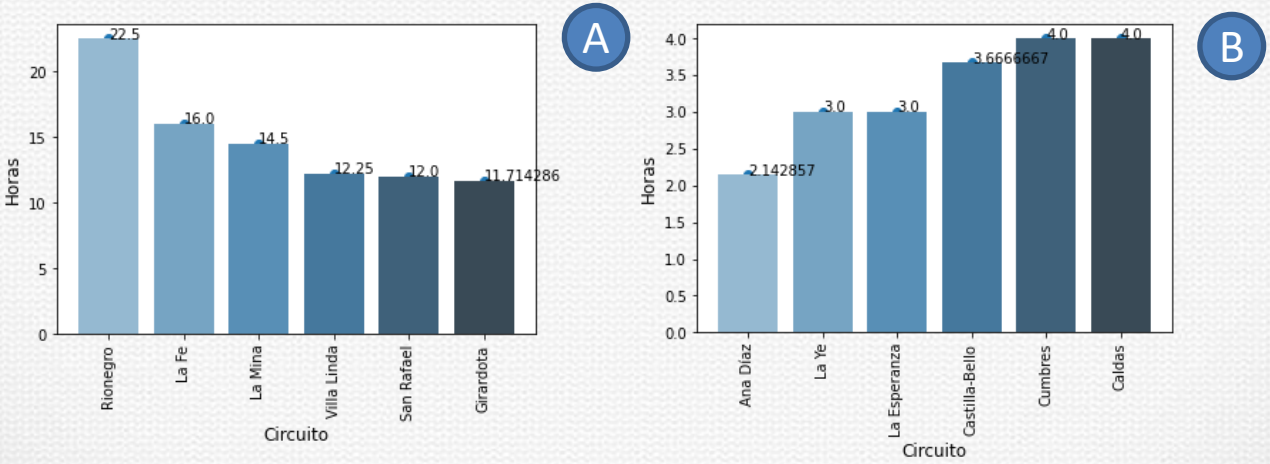




**Figura 14.** Top 6 de los mayores (A) y menores (B) tiempos promedios de intervención desglosado por **Circuito**.

# Proceso de codificación y análisis

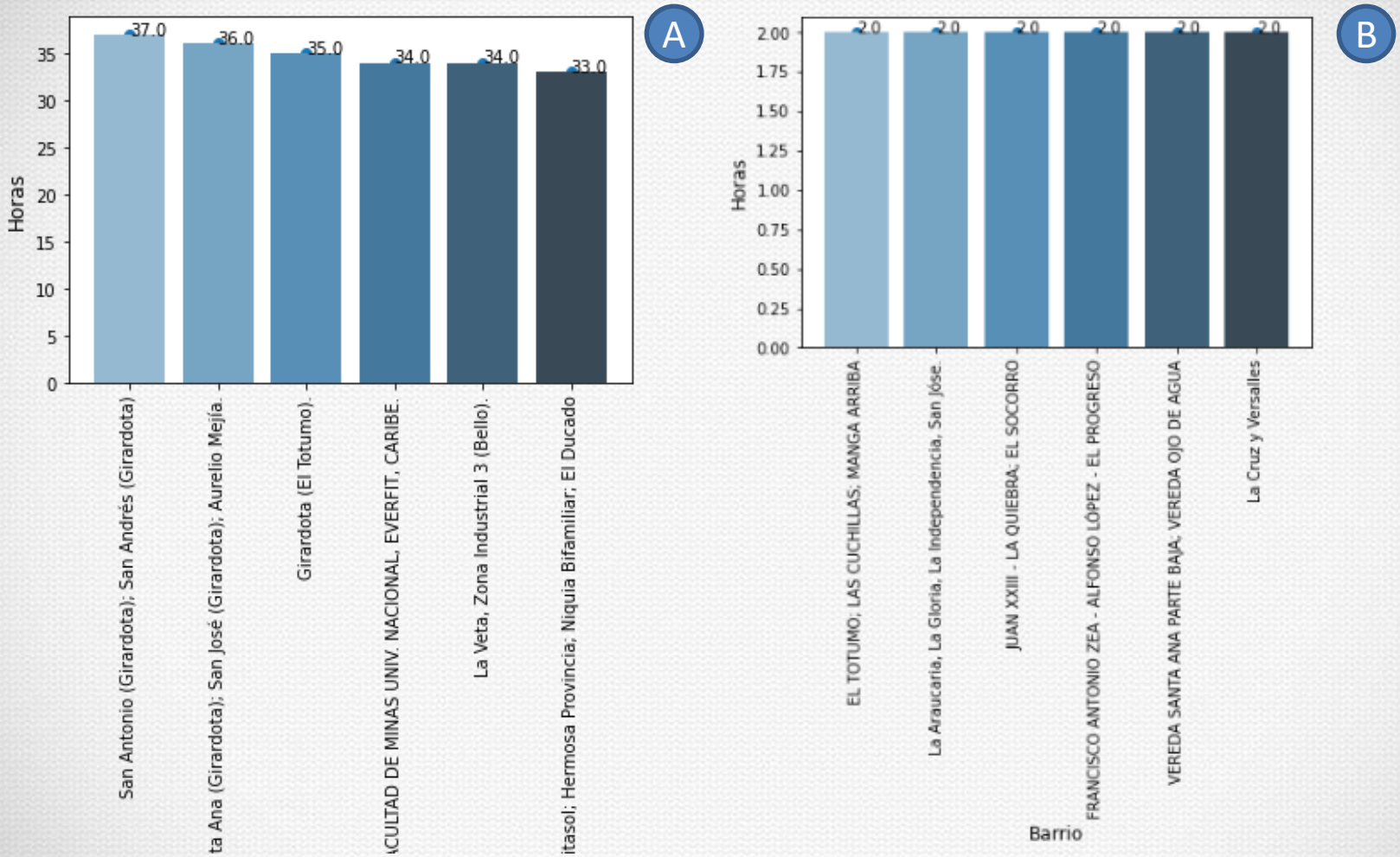
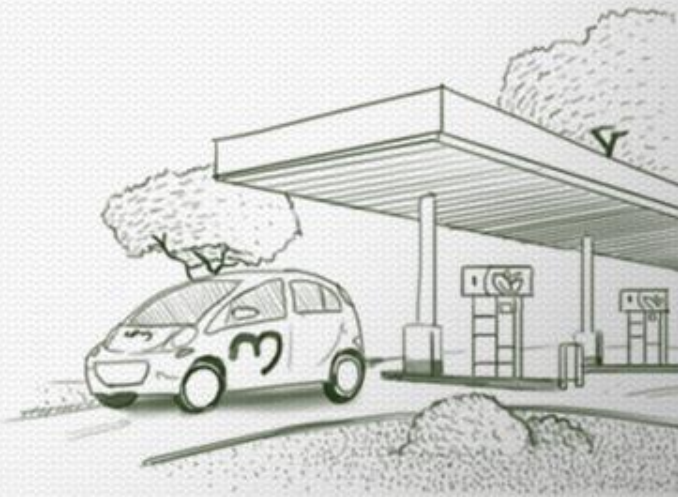



**Figura 15.** Top 6 de los mayores (A) y menores (B) tiempos promedios de intervención desglosado por **Barrio**.

# Proceso de codificación y análisis

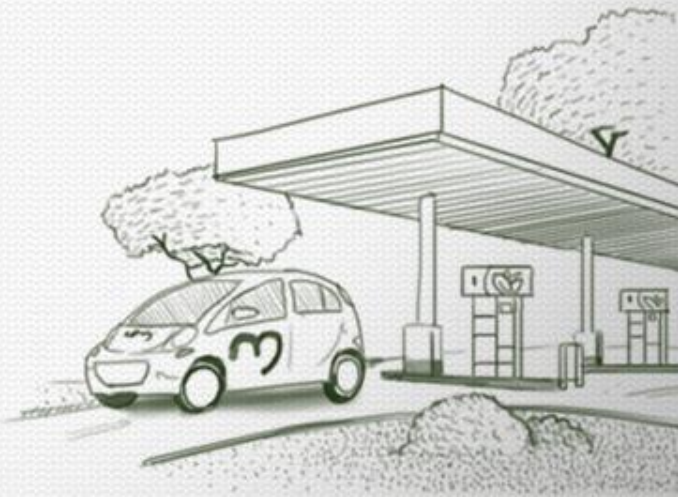
Según lo visto en las gráficas (ver ***Figura 14 y 15***) y realizando un análisis exhaustivo de los datos, se puede afirmar que no hay ningún tipo de relación entre el estrato socioeconómico y el tiempo que duran las intervenciones, ya que tanto sectores marginados como sectores de alto poder adquisitivo presentan de igual forma tiempos de intervención promedio altos, como bajos. Esto se puede ratificar al analizar el mapa de calor llevado a cabo con la librería **Folium**, donde se puede visualizar que los puntos críticos (zonas con un tiempo de intervención promedio alto) demarcados con un color de gradiente rojizo están extendidos por todo el Valle de Aburrá sin exponer ningún tipo de patrón relacionado con el estrato socioeconómico, asimismo, los puntos con un tiempo de intervención bajo (demarcados con un color de gradiente azuloso) se encuentran dispersos a lo largo del territorio antioqueño.



# Proceso de codificación y análisis

- **Punto 3:** Calcular los tiempos de intervención para cada municipio y sector del que se tenga reporte: valor máximo, mínimo, promedio y varianza.

Para este apartado, se crearon los dataframe **Tabla_municipio** y **Tabla_circuito**, para los cuales, mediante un ciclo para (**For**) se iban agregando a las respectivas tablas el nombre del municipio, el valor máximo, valor mínimo, promedio y varianza asociados al valor único en cuestión. A continuación, se expondrá la codificación realizada y las salidas obtenidas (ver ***Figura 16 y 17***) .



# Proceso de codificación y análisis

```
[22]  1 tabla_municipio = pd.DataFrame(columns=['Municipio', 'Max', 'Min', 'Promedio', 'Varianza'])
      2 tabla_circuito = pd.DataFrame(columns=['Circuito', 'Max', 'Min', 'Promedio', 'Varianza'])
```

```
[25]  1 tabla_municipio
```

```
1 #Municipio
2 for i in epm['Municipio'].unique():
3   datos = np.array(epm[(epm['Municipio'] == i)]['Horas'])
4   try:
5     max = datos.max()
6     min = datos.min()
7     promedio = np.mean(datos)
8     varianza = datos.var()
9   except ValueError:
10    pass
11  tabla_municipio = tabla_municipio.append({'Municipio' : str(i) , 'Max' : float(round(max,2)), 'Min' : float(round(min,2)),
12                                            'Promedio': float(round(promedio,2)), 'Varianza': float(round(varianza,2))} , ignore_index=True)
```

```
[27]  1 #Circuito
      2 for i in epm['Circuito'].unique():
      3   datos = np.array(epm[(epm['Circuito'] == i)]['Horas'])
      4   try:
      5     max = datos.max()
      6     min = datos.min()
      7     promedio = np.mean(datos)
      8     varianza = datos.var()
      9   except ValueError:
     10     pass
     11   tabla_circuito = tabla_circuito.append({'Circuito' : str(i) , 'Max' : float(round(max,2)), 'Min' : float(round(min,2)),
     12                                           'Promedio': float(round(promedio,2)), 'Varianza': float(round(varianza,2))} , ignore_index=True)
```

**Figura 16.** Codificación del dataframe **Tabla_municipio** y **Tabla_circuito**.

# Proceso de codificación y análisis

```
1 tabla_municipio
```

| | Municipio | Max | Min | Promedio | Varianza |
|---|---|---|---|---|---|
| 0 | Medellín | 34.0 | 1.0 | 6.55 | 10.780000 |
| 1 | Bello | 35.0 | 2.0 | 8.31 | 35.380001 |
| 2 | Copacabana | 41.0 | 5.0 | 10.09 | 74.080002 |
| 3 | Girardota | 37.0 | 2.5 | 10.66 | 122.629997 |
| 4 | Envigado | 19.0 | 2.0 | 6.20 | 8.750000 |
| 5 | Itagüí | 22.0 | 2.0 | 8.46 | 22.969999 |
| 6 | Sabaneta | 11.0 | 3.0 | 5.28 | 3.500000 |
| 7 | La Estrella | 13.0 | 5.0 | 7.62 | 3.230000 |
| 8 | Barbosa | 6.0 | 4.0 | 4.80 | 0.560000 |
| 9 | Rionegro | 22.5 | 2.0 | 7.96 | 14.920000 |
| 10 | Caldas | 17.0 | 4.0 | 5.91 | 14.080000 |
| 11 | El Retiro | 28.0 | 3.0 | 11.60 | 89.239998 |

```
1 tabla_circuito
```

| | Circuito | Max | Min | Promedio | Varianza |
|---|---|---|---|---|---|
| 0 | San Cristóbal | 14.0 | 1.0 | 6.09 | 10.910000 |
| 1 | Castilla-Bello | 5.0 | 2.0 | 3.67 | 1.560000 |
| 2 | Aguas Frías | 30.0 | 6.0 | 9.44 | 53.139999 |
| 3 | Villa Hermosa | 7.0 | 7.0 | 7.00 | 0.000000 |
| 4 | Altos De Niquia | 33.0 | 3.0 | 7.75 | 61.189999 |
| ... | ... | ... | ... | ... | ... |
| 106 | San Antonio De Pereira | 9.0 | 5.0 | 7.17 | 2.140000 |
| 107 | Llanogrande | 10.0 | 4.0 | 7.67 | 6.890000 |
| 108 | La Fe | 28.0 | 3.0 | 16.00 | 100.330002 |
| 109 | La Quiebra | 11.5 | 4.0 | 7.83 | 9.390000 |
| 110 | La Honda | 10.0 | 5.0 | 7.00 | 3.000000 |

111 rows × 5 columns

**Figura 17. Tabla_municipio** y **Tabla_circuito**.

# Proceso de codificación y análisis

**Punto 4:** Definir una estrategia de visualización de los datos que de soporte a los hallazgos obtenidos y a las conclusiones del proceso.

La estrategia de visualización de los datos se basó en el uso de diversos tipos de gráficos cuyo fin era soportar las conclusiones realizadas basados en el set de datos analizado. Los gráficos utilizados en la estrategia fueron: histograma de densidad, histograma de frecuencia, diagrama de pares, diagrama de dispersión, diagrama de caja y diagrama de barras. De igual forma, mediante la API **Geocoding** (desarrollada por Mapquest Developer) se lograron obtener las coordenadas geográficas referentes a la ubicación exacta donde se efectuaron las intervenciones en cuestión, mediante el suministro de la **Dirección** (nuevo campo creado en el set de datos), el cual comprendía la concatenación de las variables **Circuito** y **Municipio** más la constante **"Antioquia, Colombia"**(ver ***Figura 18***). Seguidamente, mediante la implementación de la librería **Folium**, se elaboró un mapa de calor (**Heatmap**) con las coordenadas generadas anteriormente, con el objetivo de ilustrar la magnitud del tiempo empleado para ejecutar las intervenciones en el servicio de acueducto, donde la variación de colores denota la intensidad de la variable respuesta, es decir que, a un mayor número promedio de horas la tonalidad tiende a un color rojizo y en caso contrario (menor número promedio de horas) la tonalidad tiende a un color azuloso. Este tipo de gráficos expone de forma precisa el tiempo promedio que duraron cierto número de intervenciones en cualquier zona comprendida dentro del área metropolitana del Valle de Aburrá (ver ***Figura 19 y 20***) . Este tipo de iniciativas podrían ser provechosas para EPM, ya que a través de estas visualizaciones tipo mapa de calor, es posible detectar en que zonas se han llevado a cabo intervenciones de manera ineficientes, generando de esta forma interrogantes referentes a posibles factores que podían influenciar en un tiempo de mantenimiento mayor, de esta forma EPM podría planificar de mejor forma el cómo abordar las intervenciones en zonas críticas, ya sea a través de mayor personal, maquinaria más eficiente, etc.

# Proceso de codificación y análisis

```
[ ]  1 #Dirección completa solo con Circuito y municipio
     2 epm['Dirección completa2'] = [0 for i in range(len(epm['Municipio']))]
     3 epm['Latitud_ubi2'] = [0 for i in range(len(epm['Municipio']))]
     4 epm['Longitud_ubi2'] = [0 for i in range(len(epm['Municipio']))]
     5 epm['Lati+Longi'] = [0 for i in range(len(epm['Municipio']))]
```

```
[ ]  1 epm = epm.astype({'Dirección completa2' : 'object', 'Latitud_ubi2' : 'object', 'Longitud_ubi2' : 'object', 'Lati+Longi' : 'object'})
```

```
[ ]  1 for i, row in epm.iterrows():
     2   apiAddress = (str(epm.at[i,'Circuito'])+','+str(epm.at[i,'Municipio']+',Antioquia,Colombia'))
     3   epm.at[i,'Dirección completa2'] = apiAddress
     4
     5   parameters = {"key": "6SVccLzFrvBJqpGguwZJMgY6mDppOrAo",
     6                 "location": apiAddress}
     7
     8   response = requests.get("http://www.mapquestapi.com/geocoding/v1/address", params=parameters)
     9   data = response.text
    10   dataJ = json.loads(data)['results']
    11   lat = (dataJ[0]['locations'][0]['latLng']['lat'])
    12   lng = (dataJ[0]['locations'][0]['latLng']['lng'])
    13
    14   #Agregar resultados al DF epm:
    15   epm.at[i,'Latitud_ubi2'] = lat
    16   epm.at[i,'Longitud_ubi2'] = lng
```

**Figura 18.** Codificación para la creación de las coordenadas (latitud, longitud) mediante la API **Geocoding**.

# Proceso de codificación y análisis

```
[ ]  1 for i, row in epm.iterrows():
     2   a = ('(' + str(epm.at[i,'Latitud_ubi2']) + ',' + str(epm.at[i,'Longitud_ubi2']) + ')')
     3   epm.at[i,'Lati+Longi'] = a
```

```
[ ]  1 tabla_mapadecalor = pd.DataFrame(columns=['Lati+Longi', 'Latitud', 'Longitud', 'Horas_prom'])
     2 tabla_mapadecalor = tabla_mapadecalor.astype({'Latitud' : 'float32', 'Longitud' : 'float32', 'Horas_prom' : 'float32'})
```

```
[ ]  1 for i in epm['Lati+Longi'].unique():
     2   datos = np.array(epm[(epm['Lati+Longi'] == i)]['Horas'])
     3   try:
     4     prom = np.mean(datos)
     5     parte = i.split(',')
     6     lat = parte[0][1:7]
     7     lon = parte[1][0:7]
     8   except ValueError:
     9     pass
    10   tabla_mapadecalor = tabla_mapadecalor.append({'Lati+Longi' : str(i) , 'Latitud' : np.float32(lat), 'Longitud' : np.float32(lon),
    11                                                 'Horas_prom': np.float32(prom)} , ignore_index=True)
```

```
[ ]  1 tabla_mapadecalor.dtypes
```

```
[ ]  1 meanLong3 = -75.567
     2 meanLat3 = 6.217
     3
     4 print(meanLong3, meanLat3);
     5
     6 # Crear mapa base
     7 mapObj3 = folium.Map(location=[meanLat3, meanLong3], zoom_start = 12.5)
     8
     9 # Crear capa de heatmap
    10 colrGradient = {0.3: 'blue', 0.6: 'yellow', 0.7: 'red'}
    11
    12 heatmap3 = HeatMap(list(zip(tabla_mapadecalor['Latitud'], tabla_mapadecalor['Longitud'], tabla_mapadecalor['Horas_prom'])),
    13                     min_opacity=0.6,
    14                     max_val = tabla_mapadecalor['Horas_prom'].max(),
    15                     radius = 50, blur = 60,
    16                     max_zoom = 1,
    17                     gradient = colrGradient,
    18                     overlay = True)
    19 # Agregar la capa de heatmap al mapa base
    20 heatmap3.add_to(mapObj3)
    21 mapObj3
```

```
[ ]  1 mapObj3.save("Heatmap2.html")
```

**Figura 19.** Codificación para la implementación de la librería **Folium** y la creación del mapa de calor.

# Proceso de codificación y análisis

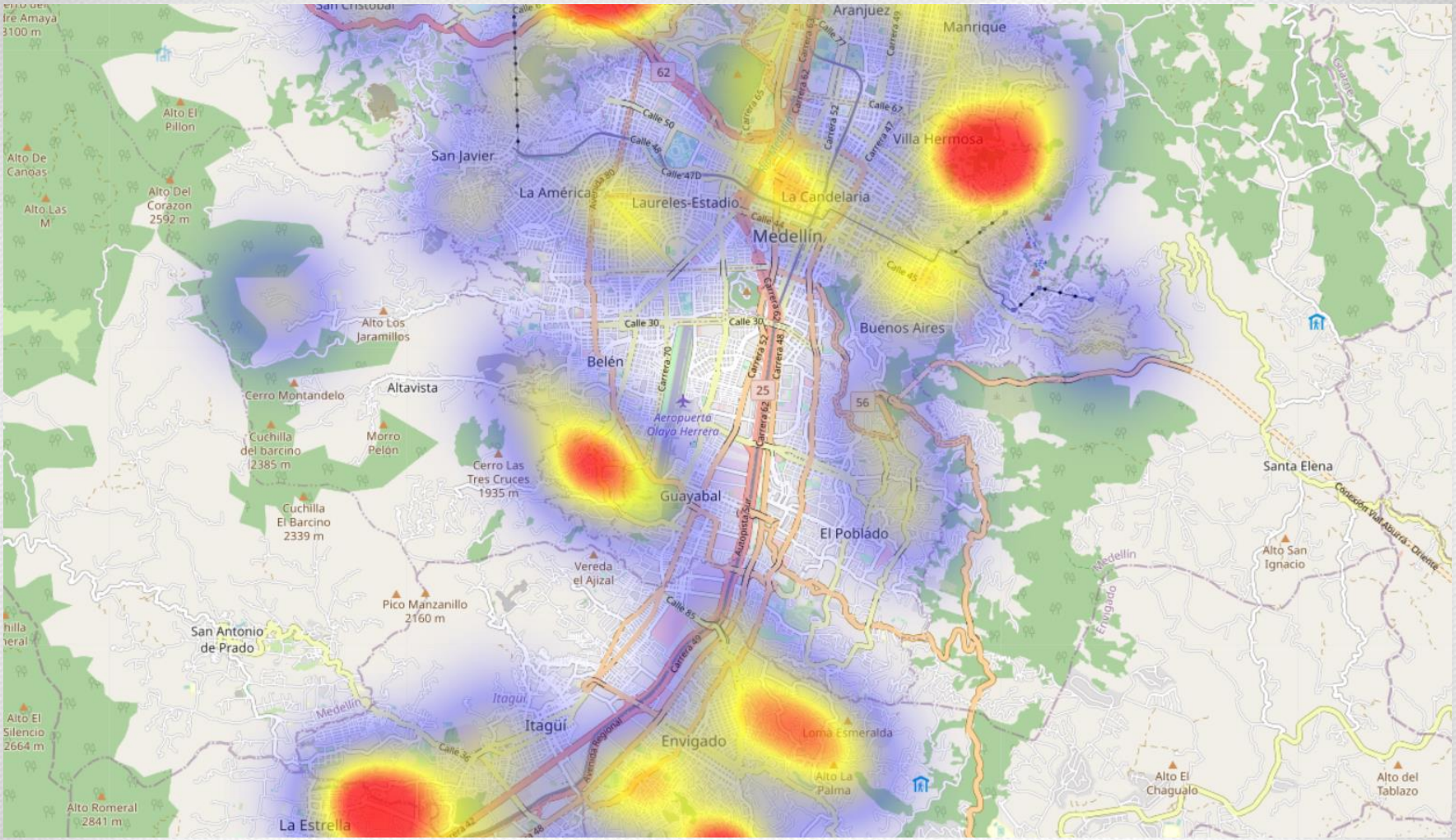
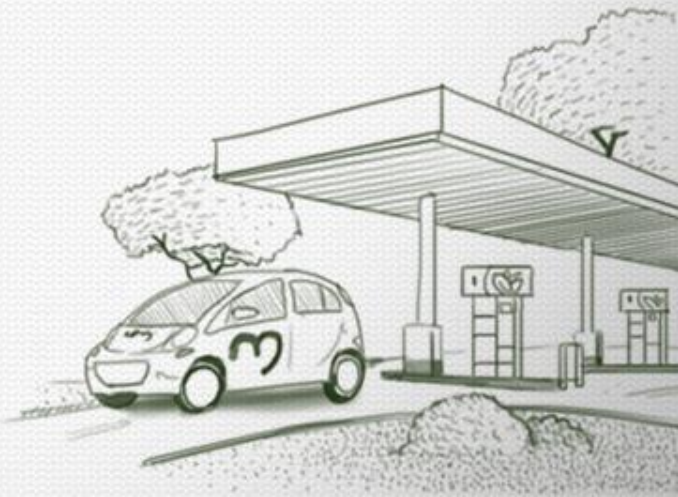



**Figura 20.** Vista general del mapa de calor creado.

*Nota: En la capeta del reto se anexara el mapa de calor en formato HTML (Heatmap.html).*

# ¿Por qué esta programación tiene relevancia dentro del mundo del ML?

Esta programación tiene relevancia dentro del mundo del ML, dado que realiza una sinergia de todo el proceso a seguir para ajustar modelos de inteligencia artificial a un problema en concreto, es decir, todo el preprocesado de datos, análisis exploratorio de la data, arquitectura del set de datos, creación de visualizaciones, etc. Por otra parte, desde un punto personal, esta programación ha sido significativa considerando que, brinda la oportunidad de integrar todo lo aprendido en el proceso formativo para ponerlo en práctica dentro de un proyecto concreto y real, en este caso analizar las intervenciones realizadas al acueducto en toda el área metropolitana del Valle de Aburrá por parte de **EPM**. Esto brinda un enfoque funcional del saber, permitiendo potencializar los conocimientos adquiridos.



# ¿Cómo su “producto” (código fuente) puede aportar a las industrias y/o programas y plataformas?

El contar con un sistema de abastecimiento de agua constante y eficiente es vital para la sociedad, por ello, se convierte en un imperativo para la entidad prestadora de servicio (**EPM**) y el distrito gubernamental antioqueño, garantizar la provisión de los servicios de acueducto y alcantarillado a partir de acciones de corto, mediano y largo plazo.

Es obligación de **EPM** el velar por un correcto funcionamiento del sistema de acueductos y alcantarillados dentro del área metropolitana del Valle de Aburra, por lo tanto, es necesaria la gestión de las redes de conducción de agua y establecer una estrategia para su mantenimiento que contrarreste el deterioro producido por el uso en el tiempo. Las operaciones más habituales son, básicamente, la inspección, la limpieza y las reparaciones puntuales de averías. Pero si, en determinados casos, la red presenta inconvenientes que el mantenimiento no puede superar, habrá que proceder a su rehabilitación y cuando ello resulte inviable, llevar a cabo su renovación.

Todos estos casos fueron tenidos en cuenta en el set de datos analizado, por ende, la programación codificada sirve como base para llevar un seguimiento de los tiempos de intervención desglosados por impacto del arreglo, motivo del arreglo, municipio y/o circuito donde se realizó la intervención, etc. Levar a cabo un análisis exploratorio de los datos referente a las intervenciones realizadas sumado a la estadística descriptiva obtenida, permite tener una idea clara de los factores que repercuten en la duración de estas y, por lo tanto, tener una métrica clara de la eficiencia de los procedimientos correctivos y preventivos efectuados, lo cual repercuten favorablemente en la calidad del servicio ofrecido a la comunidad antioqueña.

# ¿Cómo su “producto” (código fuente) puede aportar a las industrias y/o programas y plataformas?

El contar con un sistema de abastecimiento de agua constante y eficiente es vital para la sociedad, por ello, se convierte en un imperativo para la entidad prestadora de servicio (**EPM**) y el distrito gubernamental antioqueño, garantizar la provisión de los servicios de acueducto y alcantarillado a partir de acciones de corto, mediano y largo plazo.

Es obligación de **EPM** el velar por un correcto funcionamiento del sistema de acueductos y alcantarillados dentro del área metropolitana del Valle de Aburra, por lo tanto, es necesaria la gestión de las redes de conducción de agua y establecer una estrategia para su mantenimiento que contrarreste el deterioro producido por el uso en el tiempo. Las operaciones más habituales son, básicamente, la inspección, la limpieza y las reparaciones puntuales de averías. Pero si, en determinados casos, la red presenta inconvenientes que el mantenimiento no puede superar, habrá que proceder a su rehabilitación y cuando ello resulte inviable, llevar a cabo su renovación.

Todos estos casos fueron tenidos en cuenta en el set de datos analizado, por ende, la programación codificada sirve como base para llevar un seguimiento de los tiempos de intervención desglosados por impacto del arreglo, motivo del arreglo, municipio y/o circuito donde se realizó la intervención, etc. Levar a cabo un análisis exploratorio de los datos referente a las intervenciones realizadas sumado a la estadística descriptiva obtenida, permite tener una idea clara de los factores que repercuten en la duración de estas y, por lo tanto, tener una métrica clara de la eficiencia de los procedimientos correctivos y preventivos efectuados, lo cual repercuten favorablemente en la calidad del servicio ofrecido a la comunidad antioqueña.

# Forecast

Para el apartado de pronóstico, se procedió con el preprocesamiento de los datos, donde se seleccionaron las potenciales variables predictoras que harían parte del entrenamiento y evaluación de los modelos de Machine learnig. Estas variables fueron las siguientes: **Impacto, Motivo, Número de instalaciones y Municipio**. Seguidamente, se dividió el set de datos de la siguiente forma: **Train, Test y Performance**. Esto con el objetivo de no sesgar los indicadores de eficiencia de los modelos de ML ajustados y así optar por el mejor modelo predictivo. De igual forma, para validar la fiabilidad de las métricas que exponen los modelos ajustados, se emplearon los métodos Bootstrap y Validación cruzada.

Los modelos de ML ajustados fueron los siguientes: **Linear regression, Decision tree regressor y Random forest regressor.**

Ya con un planteamiento conciso referente a la gestión de los modelos, se procedió a ajustar los diferentes modelos predictivos y de esta forma crear un dataframe para registrar el desempeño expuesto por cada uno de estos (el cual se validó mediante las siguientes técnicas de re-muestreo: Cross Validation y Shuffle Split). La viabilidad del ajuste se evaluó basándose en la métrica **Error absoluto relativo medio (MRAE)**, el cual se interpreta como el porcentaje de error absoluto que expone la predicción al compararse con el valor real. Según esto, es evidente que debemos optar por el modelo que exponga un menor MRAE, es decir, que tienda a cero (0%).

# Forecast

A continuación, se expondrá el resumen de los modelos ajustados con su respectiva métrica de evaluación (MRAE):

| Modelo | Train score | Test score |
|---|---|---|
| RFR depth = 8 | 0.264042 | 0.385002 |
| DTR depth = 5 | 0.337599 | 0.396223 |
| DTR depth = 10 | 0.177441 | 0.404538 |
| Linear Regression | 0.424290 | 0.437725 |

**Figura 21.** Desempeño de los modelos entrenados.


**Figura 22.** Evaluación de los modelos en base al MRAE.

# Conclusión Forecast

Con base al resumen, se puede concluir que el modelo de Random Forest con un nivel de profundidad igual a 8 expone el menor MRAE y, por lo tanto, se puede concluir que es el modelo que predice de mejor forma la variable respuesta. En conclusión, el modelo seleccionado presenta un rendimiento de **MRAE = 37.30%.** Dicho rendimiento fue calculado con el set de datos denominado **“Performance”.**

# Diagnóstico del modelo predictivo

A continuación, se expondrá un diagnóstico del modelo predictivo seleccionado previamente, en donde se detalla la evolución del nivel de ajuste a medida que el set de datos de entrenamiento va aumentando.


**Figura 23.** Diagnostico del modelo ajustado.

Según el diagnóstico realizado, se puede evidenciar que el modelo se encuentra sesgado, ya que el ajuste del modelo no mejora a medida que el tamaño de la muestra aumenta. En primera instancia, para mitigar el nivel de sesgo, se optó por ajustar modelos con un nivel de complejidad mayor, sin embargo, el ajuste no mejoro, a partir de un nivel de profundidad igual o mayor a 8 el modelo no presenta mejoras en su nivel de MRAE. Por lo tanto, se puede concluir que, para mejorar el rendimiento del modelo seleccionado, se deben agregar más variables predictoras al modelo, lo cual está por fuera de nuestro alcance, ya que este proyecto se llevó a cabo mediante una base de datos extraída de un repositorio abierto.

Gracias

Grupo·epm®