



```

In [ ]: # Filtramos escuelas
df_escuelas = filtrar_dataframe(file_path_escuelas, columnas_para_filtrar, [

In [ ]: # Filtramos parques
terminos_parques = ['parques', 'jardin', 'plaza', 'bosque']
df_parques = filtrar_dataframe(file_path_parques, columnas_para_filtrar, ter

In [ ]: # Salud es complicado, porque queremos hospitales, publicas y privadas separ
terminos_privadas = ['consultorio', 'privado']
terminos_publicas = ['imss', 'publico', 'isste']
terminos_hospitales = ['hospital', 'cirugia']

df_salud_privadas = filtrar_dataframe(file_path_salud, columnas_para_filtrar
df_salud_publicas = filtrar_dataframe(file_path_salud, columnas_para_filtrar
df_salud_hospitales = filtrar_dataframe(file_path_salud, columnas_para_filtr

In [ ]: # Para los negocios de alcohol tenemos que unir todos los dataframes al fina
terminos_alcohol = ['alcohol', 'licor', 'cerveza', 'bar', 'cantina', 'vinos']

df_alcohol_list = []

for path in paths_negocios_denue:
    df_filtrado = filtrar_dataframe(path, columnas_para_filtrar, terminos_al
    df_alcohol_list.append(df_filtrado)

df_alcohol = pd.concat(df_alcohol_list, ignore_index=True)

In [ ]: # Fabricas
terminos_fabricas = ['fabrica', 'fabricacion', 'industria', 'industrias', 'i
df_fabricas = filtrar_dataframe(file_path_mayoreo, columnas_para_filtrar, te

```

## EDA

```

In [ ]: import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))

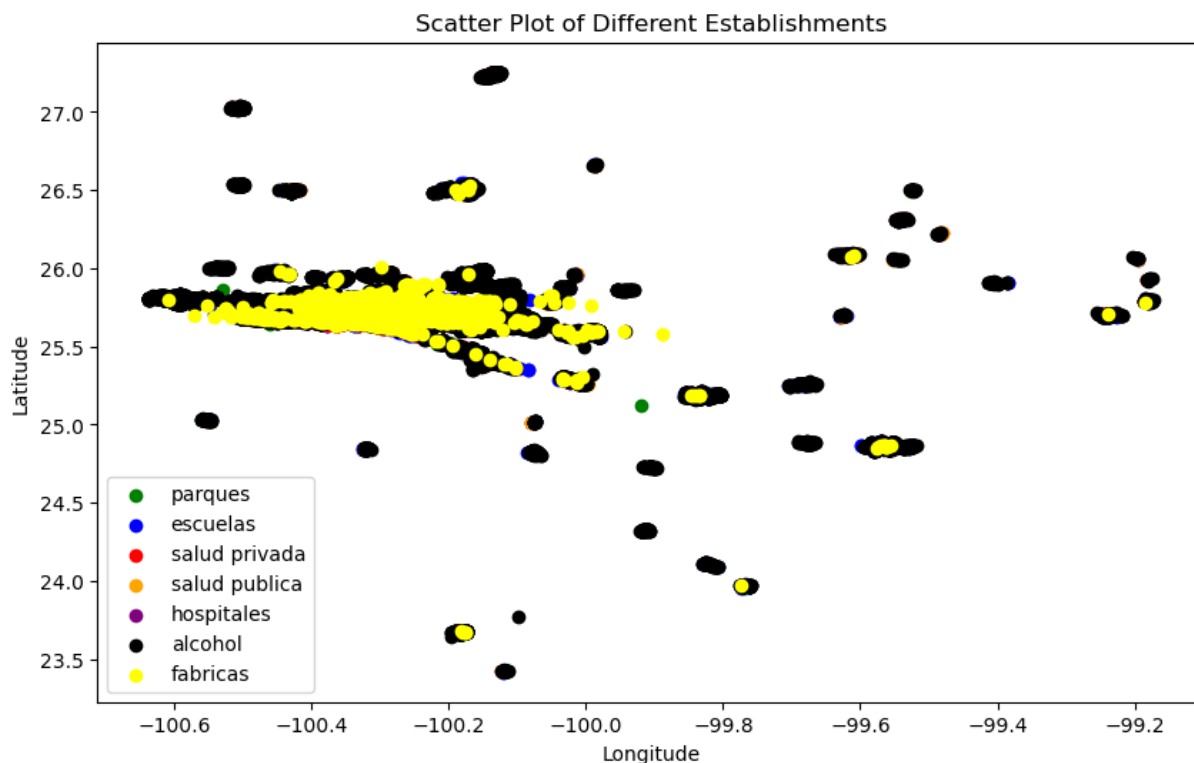
# Plot each dataframe with a different color
plt.scatter(df_parques['longitud'], df_parques['latitud'], color='green', la
plt.scatter(df_escuelas['longitud'], df_escuelas['latitud'], color='blue', l
plt.scatter(df_salud_privadas['longitud'], df_salud_privadas['latitud'], col
plt.scatter(df_salud_publicas['longitud'], df_salud_publicas['latitud'], col
plt.scatter(df_salud_hospitales['longitud'], df_salud_hospitales['latitud'],
plt.scatter(df_alcohol['longitud'], df_alcohol['latitud'], color='black', la
plt.scatter(df_fabricas['longitud'], df_fabricas['latitud'], color='yellow',

# Add a title and labels
plt.title('Scatter Plot of Different Establishments')
plt.xlabel('Longitude')
plt.ylabel('Latitude')

```

```
# Add a legend to identify the datasets
plt.legend()

# Show the plot
plt.show()
```



Todavia no queda, esta en testing

Se crea un KDTree

Se guarda el indice y la distancia del establecimiento mas cercano a la escuela

```
In [ ]: # Encuentra el establecimiento más cercano a una coordenada dado un arbol de
def find_nearest(row, tree, establishments):
    distance, index = tree.query([row['latitud'], row['longitud']])
    return establishments.iloc[index]['nom_estab'], distance

# Tree para aplicar busqueda
alcohol_tree = KDTree(alcohol[['latitud', 'longitud']])

# simon
schools['nearest_alcohol'], schools['distance_to_nearest_alcohol'] = zip(*sc
```