

Batalla de los Vecindarios (Semana 2)

Análisis de la diversidad culinaria en la industria de restaurantes en Buenos Aires: una perspectiva geográfica

Por: MSc. Willer Torrico



I. INTRODUCCION

Este proyecto de investigación tiene como objetivo ayudar a los inversores interesados en el mercado de restaurantes de Buenos Aires a identificar oportunidades comerciales mediante el análisis de la diversidad culinaria y la distribución geográfica de los restaurantes en la ciudad.

El mercado de restaurantes en Buenos Aires es muy competitivo y diverso, con una gran cantidad de establecimientos que ofrecen una variedad de cocinas y estilos. Esto puede hacer que sea difícil para los inversores determinar dónde invertir su capital y qué tipo de restaurantes tienen el potencial de tener éxito en la ciudad.

Los resultados de esta investigación serán beneficiosos tanto para inversores locales como internacionales que buscan maximizar su rentabilidad en la industria de restaurantes. Al tener una comprensión clara de la distribución geográfica y diversidad culinaria de los restaurantes en Buenos Aires, los inversores podrán tomar decisiones más informadas y estratégicas sobre dónde invertir su capital y qué tipo de restaurantes tienen el potencial de tener éxito en la ciudad.

Además, los resultados de esta investigación también serán útiles para otros actores en la industria de restaurantes, como propietarios y gerentes de restaurantes, turistas y residentes de Buenos Aires que buscan explorar la oferta gastronómica de la ciudad. En resumen, este proyecto de investigación tiene el potencial de ofrecer información valiosa para una amplia gama de personas interesadas en la industria de restaurantes en Buenos Aires.

II. DATOS Y USO DE LOS DATOS

Los datos que se utilizaron para resolver el problema de identificar oportunidades comerciales en el mercado de restaurantes de Buenos Aires provienen del archivo geoJson proporcionado por la Dirección General de Estadística y Censos de la Ciudad de Buenos Aires. Este archivo contiene información detallada sobre la ubicación y categoría de los restaurantes en la ciudad, lo que permite analizar la distribución geográfica y la diversidad culinaria de los mismos.

En particular, el archivo geoJson incluye información como el nombre del establecimiento, su dirección, la categoría culinaria a la que pertenece (por ejemplo, cocina italiana, parrilla argentina, sushi, etc.), el número de teléfono y el sitio web, entre otros datos. Además, el archivo geoJson también proporciona la ubicación exacta de cada restaurante en forma de coordenadas geográficas, lo que permitirá visualizar la distribución espacial de los establecimientos en la ciudad.

Se decidió utilizar exclusivamente el archivo geoJson debido a que este contiene información más detallada que la API de Foursquare (se puede observar en el trabajo "Batalla de los barrios semana1"), que originalmente se consideró como una posible fuente de datos. Además, se espera que la información del archivo geoJson sea más precisa y actualizada, ya que proviene directamente de la fuente oficial de datos del gobierno de la Ciudad de Buenos Aires.

III. METODOLOGIA

En esta sección se describirá la metodología utilizada para analizar los datos del archivo GeoJSON obtenido como fuente de datos para resolver el problema planteado. El objetivo de la metodología es identificar patrones en la distribución geográfica de los restaurantes y en la diversidad culinaria en Buenos Aires.

Para ello, se utilizó una combinación de análisis exploratorio, pruebas estadísticas y aprendizaje automático para analizar los datos del archivo GeoJSON y encontrar patrones en la distribución geográfica de los restaurantes y en la diversidad culinaria en Buenos Aires. Esta metodología permitió identificar áreas de la ciudad con mayor potencial para invertir en la industria de restaurantes y recomendaciones para inversores interesados en el mercado de restaurantes en Buenos Aires.

3.1 Analisis exploratorio de datos

En primer lugar, se realizó un análisis exploratorio de los datos del archivo GeoJSON, mediante el uso de técnicas de visualización como mapas de calor, gráficos de pie y barras y un mapa interactivo. Se identificaron las principales áreas de la ciudad con mayor concentración de restaurantes y se analizó la diversidad culinaria en cada área.

3.1.1 Datos (dataframe de la gastronomía de Buenos Aires)

En el trabajo denominado "Batalla de los Barrios de la Semana 1", se obtuvo a partir del archivo GeoJson denominado "CapitalFederal.GeoJson" un archivo csv de nombre `Cordinates_of_Gastronomy_in_Buenos_Aires.csv`, donde se cuenta con los datos seleccionados y ordenados para realiza la presente investigacion.

In [122]:

```
import pandas as pd

# lee el archivo csv y lo carga en un dataframe
```

```
df = pd.read_csv("Coordinates_of_Gastronomy_in_Buenos_Aires.csv")

# muestra el dataframe
df.head()
```

Out[122]:

	nombre	categoria	cocina	comuna	barrio	lat	long	geometry
0	GUANTANAMERA	RESTAURANTE	CUBANA	Comuna 1	Puerto Madero	-34.622963	-58.363307	POINT (-58.3633067544777 -34.6229634698018)
1	2080 EXPRESS	RESTAURANTE	TRADICIONAL	Comuna 13	Belgrano	-34.560311	-58.453542	POINT (-58.4535422659946 -34.5603113496909)
2	22	CAFE	MINUTAS	Comuna 8	Villa Lugano	-34.677051	-58.475491	POINT (-58.4754910533597 -34.6770507014194)
3	4 X 4 CAFE	BAR	MINUTAS	Comuna 14	Palermo	-34.571843	-58.422262	POINT (-58.4222623518952 -34.571843204592)
4	5TA AVENIDA	RESTAURANTE	MINUTAS	Comuna 1	San Nicolas	-34.604797	-58.378309	POINT (-58.3783093871916 -34.6047966507886)

3.1.2 Técnicas de visualizacion.

a. Mapa de calor

Para este caso se creo un mapa de calor, utilizando la biblioteca **Seaborn** de Python y muestra el resultado utilizando la biblioteca **Matplotlib**. En particular, el mapa de calor muestra la distribución de las categorías de sitios gastronomicos en diferentes barrios de Buenos Aires. Los colores en el mapa de calor indican la cantidad de restaurantes o cafés en cada barrio y en cada categoría, con los tonos más oscuros representando una mayor cantidad. Además, los valores numéricos en cada celda del mapa de calor indican el número exacto de restaurantes o cafés en esa categoría y barrio en particular.

b. Gráficos de análisis y comparación

Esta metodología consiste en contar los valores únicos de dos variables (categoría y cocina) en un conjunto de datos, ordenarlos de mayor a menor y graficar los resultados. Primero se crea un gráfico de pastel para mostrar la distribución porcentual de las categorías. Luego se crean dos gráficos de barras para mostrar la cantidad de cada tipo de cocina, separando los valores menores o iguales al 1% en un segundo gráfico. En cada barra se agrega el porcentaje correspondiente y se giran los nombres de las categorías para mejorar la visualización. Esta metodología nos ayuda a tener una idea clara de la distribucion de las categorías y tipos de cocina en nuestro conjunto de datos y es útil para cualquier proyecto que requiera visualizar la distribución de datos categóricos.

c. Mapa interactivo

El mapa interactivo creado se utiliza para visualizar de manera interactiva y en tiempo real la ubicación de diferentes establecimientos según su categoría en una determinada zona geográfica. Esta herramienta es útil para identificar patrones y relaciones espaciales entre diferentes tipos de negocios y su ubicación en la ciudad, lo que puede ser útil para analizar el mercado y tomar decisiones empresariales, como la selección de ubicaciones estratégicas para la apertura de un negocio. Además, el mapa interactivo permite personalizar la visualización de los datos mediante el uso de marcadores y una leyenda personalizada, lo que facilita la interpretación de la información para los usuarios.

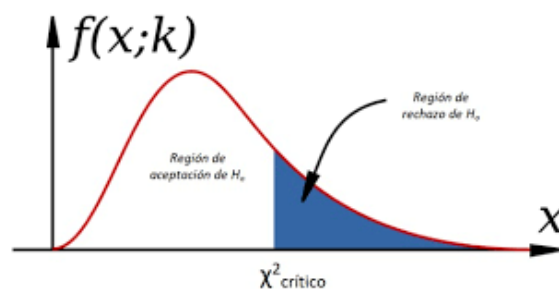
3.2 Prueba de hipotesis

Posteriormente, se realizó una prueba estadística inferencial para determinar si existía una relación significativa entre la diversidad culinaria y la ubicación geográfica de los restaurantes. Para esto, se evaluó si la distribución de los tipos de cocina en cada área de la ciudad era significativamente diferente de lo que se esperaría si la distribución fuera aleatoria.

3.2.1 Prueba de Chi 2

Para determinar si existe una relación significativa entre la diversidad culinaria (categoría y cocina) y la ubicación geográfica de los restaurantes (barrio), se empleó la prueba de chi-cuadrado. Esta prueba es una técnica estadística que se utiliza para evaluar la relación entre dos variables categóricas.

- En este caso, se planteó la **hipótesis alternativa** de que existe una asociación entre las variables de interés, es decir, que la diversidad culinaria está relacionada con la ubicación geográfica de los restaurantes.
- Por otro lado, la **hipótesis nula** establece que no existe relación entre las variables y que cualquier asociación observada es resultado del azar.



Para llevar a cabo la prueba de chi-cuadrado, se creó una tabla de contingencia que relaciona las variables de categoría, cocina y barrio. Se calculó el chi-cuadrado y el valor p para cada combinación de variables. Un valor alto de chi-cuadrado indica una mayor diferencia entre las frecuencias observadas y las esperadas, lo que sugiere una asociación entre las variables. Por su parte, un valor bajo de p indica que la probabilidad de obtener un resultado tan extremo como el observado es baja, lo que sugiere que la hipótesis nula puede ser rechazada. En este proyecto, se utilizó la prueba de χ^2 para determinar si existe una relación significativa entre la diversidad culinaria y la ubicación geográfica de los restaurantes.

3.3 Inclusion de Modelo Cluster

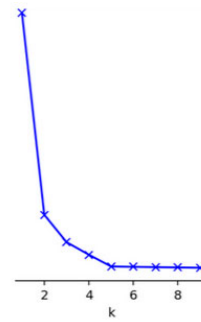
Finalmente, se utilizó aprendizaje automático para construir un modelo de clustering basado en la ubicación geográfica y los tipos de cocina de los restaurantes. Se utilizó el algoritmo de K-Means para agrupar los restaurantes en diferentes categorías según su ubicación y tipos de cocina, y se evaluó la calidad del modelo utilizando medidas de evaluación como la silueta y la inercia.

3.3.1 Método de Elbow y Silueta para Kmeans

El gráfico de "Elbow Method y Silueta para Kmeans" se utiliza para encontrar el número óptimo de clusters en un modelo de clustering KMeans. La inercia representa la suma de las distancias al cuadrado de cada punto al centroide de su cluster correspondiente, y su objetivo es minimizarse. La silueta es una medida de la cohesión y separación de los clusters, y su objetivo es maximizarse.



Método de la silueta (Silhouette Method)



Método del codo (Elbow Method)

Para este proyecto, se empleó la metodología del Elbow Method y Silueta para KMeans con el objetivo de realizar la segmentación de los restaurantes en clusters según sus coordenadas geográficas. Para ello, se siguieron varios pasos, tales como la selección de características del dataset, la estandarización de los datos, el cálculo de la inercia y la silueta, la visualización de ambas medidas en función del número de clusters, la selección del número óptimo de clusters, la aplicación del algoritmo KMeans y la visualización de los resultados en un scatter plot. Esta metodología permitió determinar el número óptimo de clusters y segmentar los restaurantes en grupos homogéneos según su ubicación geográfica, lo que facilitó el análisis posterior de los diferentes tipos de cocina presentes en cada zona.

IV. Resultados y discusion.

4.1 Exploracion de datos

En este apartado se explora los datos del estudio, que son variables del donde se muestran sus propiedades, distribuciones y correlaciones de los sitios gastronomicos en ciudad de Buenos Aires.

a. Mapa de calor

Se realizo un mapa de calor de la categoria por barrio, cuyo procedimiento y resultados se muestran a continuacion:

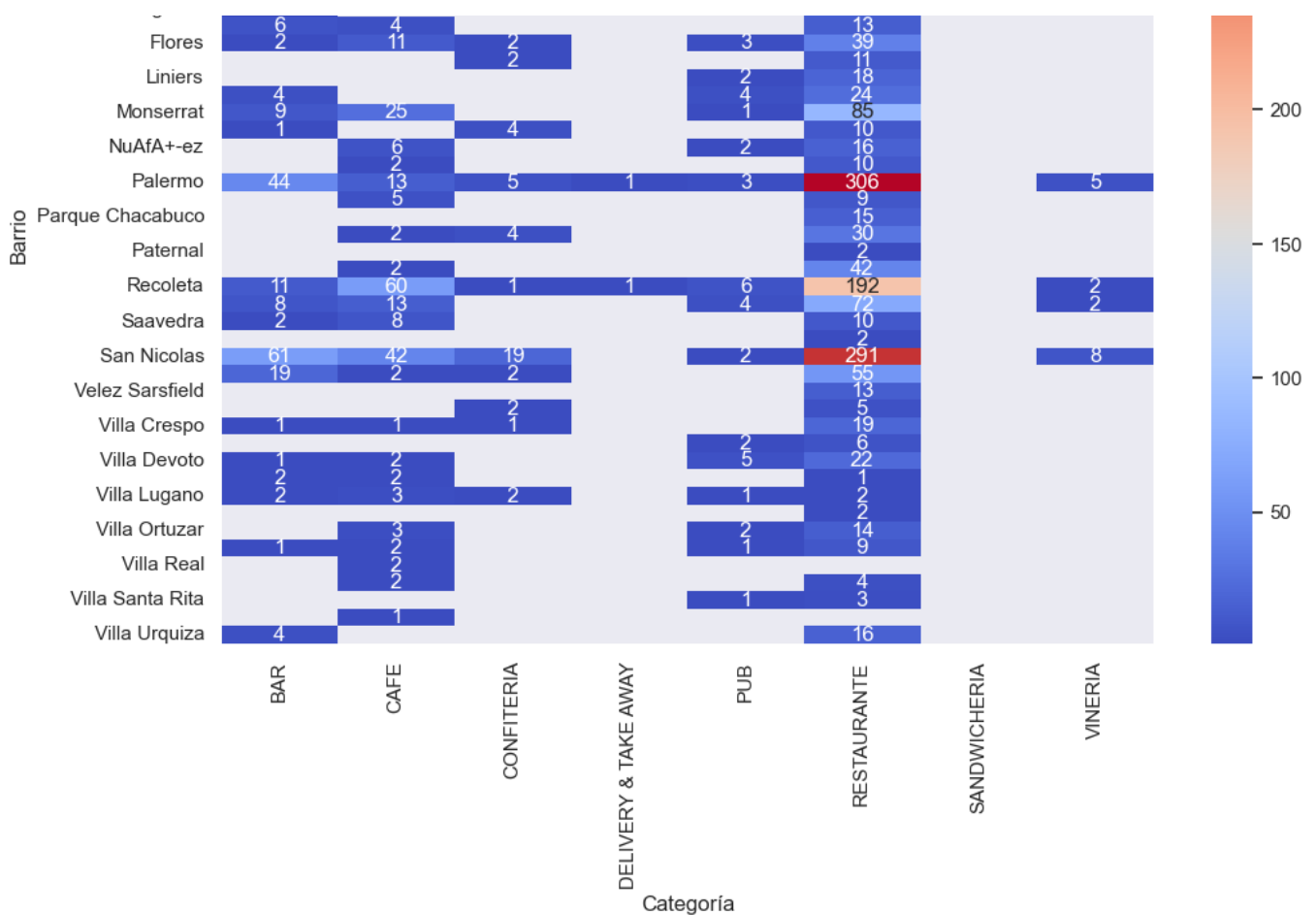
In [9]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Crear el mapa de calor
sns.set(rc={'figure.figsize': (11.7, 8.27)})
sns.heatmap(df.groupby(['barrio', 'categoria']).size().unstack(), cmap='coolwarm',
            annot=True, fmt='g')

plt.title('Mapa de calor de la categoría por barrio')
plt.xlabel('Categoría')
plt.ylabel('Barrio')
plt.show()
```





Se observa una alta concentración de negocios de comida y bebida en los barrios de Balvanera, San Nicolás, Recoleta y Palermo, especialmente en la categoría de restaurantes y cafés. Por otro lado, hay una distribución más homogénea en la categoría de bares y pubs en varios barrios de la ciudad. Cabe destacar la ausencia de negocios de comida y bebida en algunas zonas de la ciudad, como Villa Luro, Villa Santa Rita y Villa Soldati.

b. Resultados de los graficos de analisis y comparación

A continuación, se analizan y visualizan los porcentajes y cantidades de diferentes categorías y tipos de cocina en un dataset.

In [26]:

```
# Contar los tipos de categorías y ordenarlos de mayor a menor
categoria_counts = df['categoria'].value_counts().sort_values(ascending=False)

# Calcular los porcentajes de cada tipo de categoría
categoria_perc = categoria_counts / categoria_counts.sum() * 100

# Crear el gráfico de pie
fig, ax1 = plt.subplots(figsize=(8, 8))
_, _, autotexts = ax1.pie(categoria_perc.values, labels=None, autopct='%1.1f%%')
ax1.set_title('Porcentaje de categorías')
ax1.legend(categoria_perc.index, loc='best')

# Modificar los textos dentro del gráfico de pie
for i, autotext in enumerate(autotexts):
    autotext.set_text(f"{categoria_perc.values[i]:.1f}%")

# Contar los tipos de cocina y ordenarlos de mayor a menor
cocina_counts = df['cocina'].value_counts().sort_values(ascending=False)

# Crear los gráficos de barras
fig, (ax2, ax3) = plt.subplots(nrows=1, ncols=2, figsize=(16, 8))
```

```
# Separar los valores menores o iguales a 1% en un segundo gráfico
cocina_counts_1 = cocina_counts[cocina_counts / cocina_counts.sum() * 100 <= 1]

# Gráfico 1
cocina_counts = cocina_counts[cocina_counts / cocina_counts.sum() * 100 > 1]
ax2.bar(cocina_counts.index, cocina_counts.values, color='steelblue')

# Añadir porcentaje en la cima de cada barra
for i, v in enumerate(cocina_counts.values):
    ax2.text(i, v + 1, f"{cocina_counts[i]/len(df)*100:.1f}%", ha='center')

ax2.set_xlabel('Tipo de cocina')
ax2.set_ylabel('Cantidad')
ax2.set_title('Comparación de cantidad por tipo de cocina')

# Gráfico 2
ax3.bar(cocina_counts_1.index, cocina_counts_1.values, color='crimson')

# Añadir porcentaje en la cima de cada barra

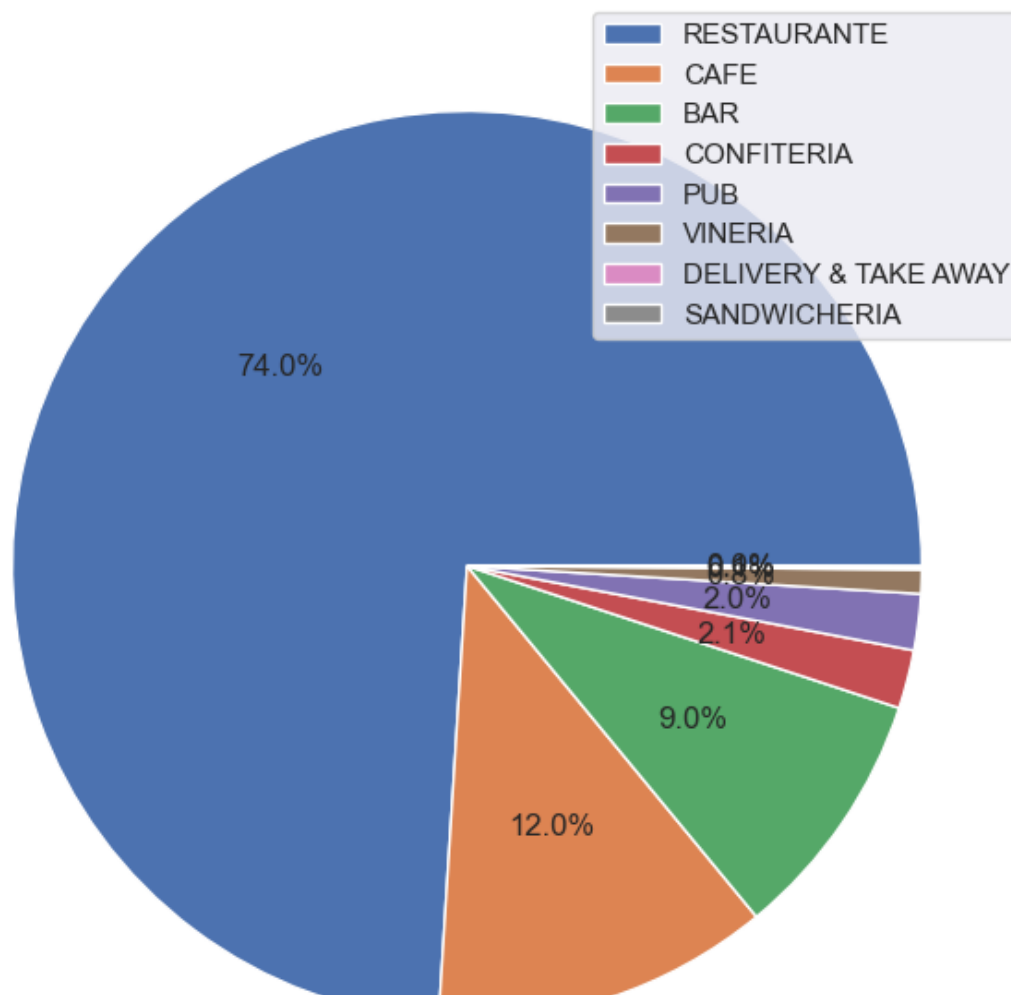
ax3.set_xlabel('Tipo de cocina')
ax3.set_ylabel('Cantidad')
ax3.set_title('Tipos de cocina con menos de 1%')

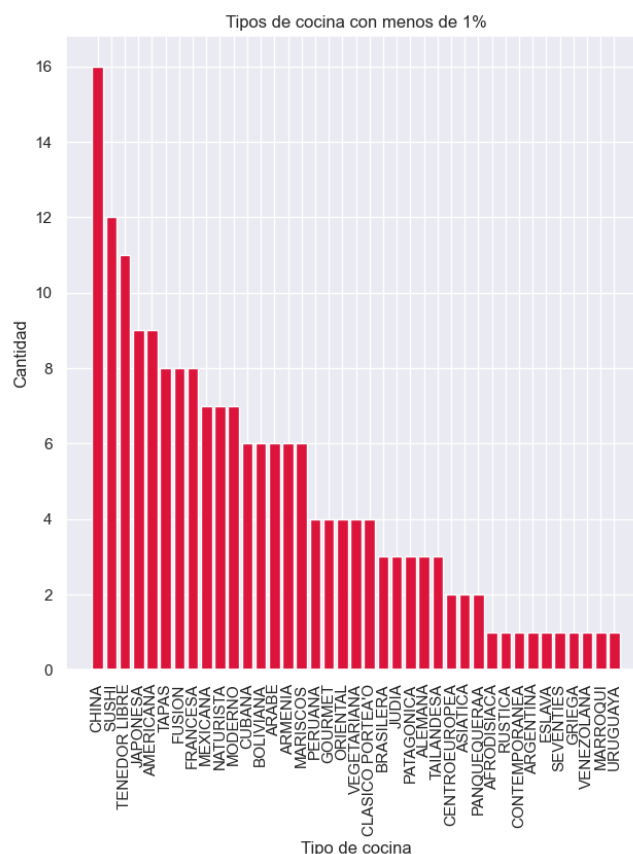
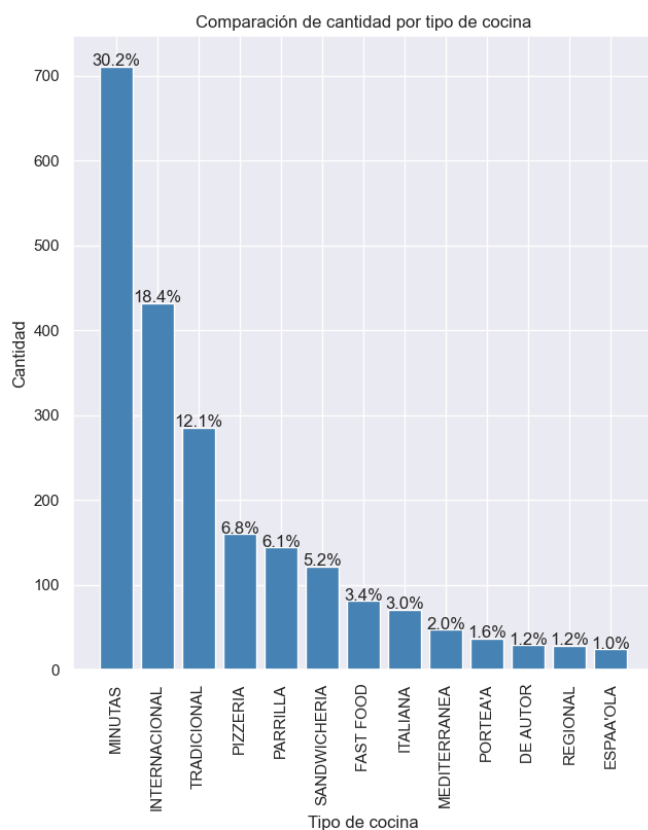
# Girar nombres de las categorías en ambos gráficos
ax2.set_xticklabels(cocina_counts.index, rotation=90)
ax3.set_xticklabels(cocina_counts_1.index, rotation=90)

plt.show()
```

```
C:\Users\wille\AppData\Local\Temp\ipykernel_15704\444830460.py:48: UserWarning: FixedFormatter should only be used together with FixedLocator
    ax2.set_xticklabels(cocina_counts.index, rotation=90)
C:\Users\wille\AppData\Local\Temp\ipykernel_15704\444830460.py:49: UserWarning: FixedFormatter should only be used together with FixedLocator
    ax3.set_xticklabels(cocina_counts_1.index, rotation=90)
```

Porcentaje de categorías





Según los graficos, se puede observar que el 74% de los locales pertenecen a la categoría "Restaurante", mientras que la categoría "Café" representa el 12% y la categoría "Bar" representa el 9%. Además, se puede ver que la mayoría de los locales ofrecen comida del tipo "Minutas" (30%), seguida de la categoría "Internacional" (18%) y "Tradicional" (12%). En menor medida, se pueden encontrar locales que ofrecen comida regional, italiana, mediterránea, entre otras.

c. Resultados del mapa interactivo

A continuacion, se crea un mapa interactivo con marcadores de diferentes categorías y una leyenda personalizada.

In [123]:

```
from ipyleaflet import Map, CircleMarker, basemaps, WidgetControl
from IPython.display import display, HTML
from ipywidgets import HTML, Box

# Crear un mapa centrado en la ubicación promedio de los puntos
center_lat = df['lat'].mean()
center_long = df['long'].mean()
m = Map(center=(center_lat, center_long), zoom=12, basemap=basemaps.OpenStreetMap.Mapnik)

# Crear una lista de colores y nombres de categorías
categories = ['RESTAURANTE', 'CAFE', 'BAR', 'CONFITERIA', 'PUB', 'VINERIA', 'DELIVERY & TAKE AWAY', 'SANDWICHERIA']
colors = ['blue', 'green', 'red', 'orange', 'purple', 'brown', 'pink', 'gray']
color_dict = dict(zip(categories, colors))

# Añadir los marcadores al mapa
for i, row in df.iterrows():
```



```

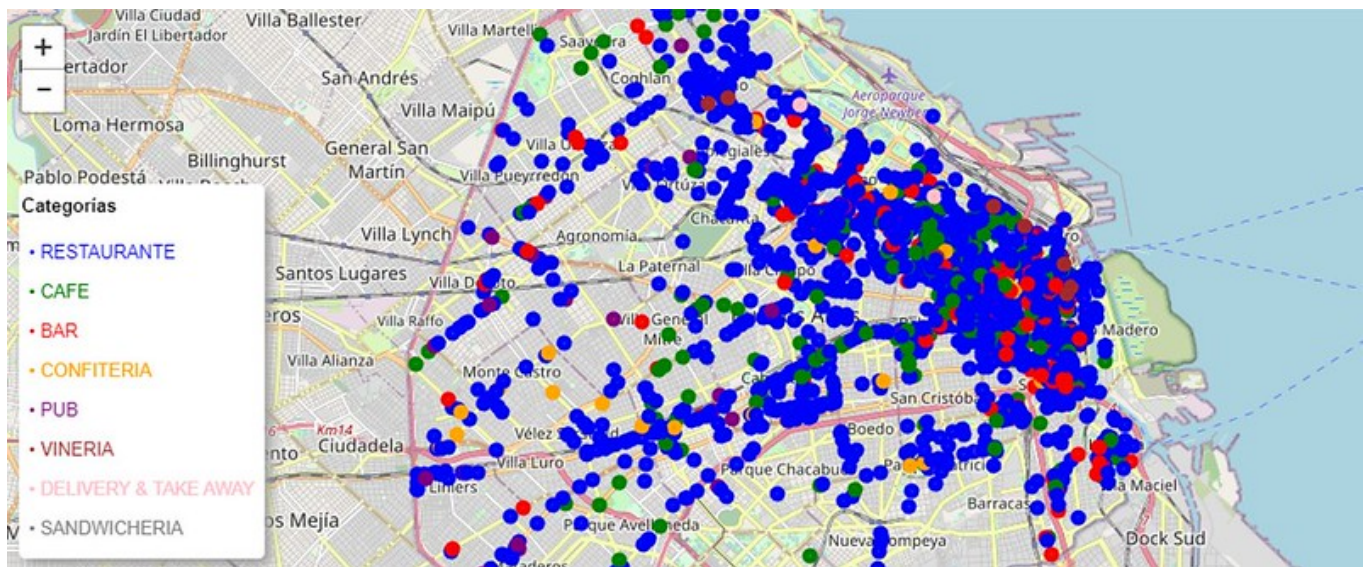
lat, long = row['lat'], row['long']
cat = row['categoria']
color = color_dict[cat]
# Utilizar CircleMarker para crear círculos con radio pequeño
marker = CircleMarker(location=(lat, long), color=color, fill_color=color, radius=3)
marker.tooltip = row['nombre']
m.add_layer(marker)

# Crear leyenda personalizada
legend_html = """
<b>Categorías</b><br>
<div style="margin: 5px;">
  <span style="color: blue;">&bull; RESTAURANTE</span><br>
  <span style="color: green;">&bull; CAFE</span><br>
  <span style="color: red;">&bull; BAR</span><br>
  <span style="color: orange;">&bull; CONFITERIA</span><br>
  <span style="color: purple;">&bull; PUB</span><br>
  <span style="color: brown;">&bull; VINERIA</span><br>
  <span style="color: pink;">&bull; DELIVERY & TAKE AWAY</span><br>
  <span style="color: gray;">&bull; SANDWICHERIA</span><br>
</div>
"""
legend = WidgetControl(widget=Box([HTML(legend_html)]), position='bottomleft')
m.add_control(legend)

m

```

Out[123]:



En el mapa se puede observar que predomina el color azul (categoría RESTAURANTE), corroborando así los gráficos analizados anteriormente, por lo tanto en la ciudad de Buenos Aires, predominan los restaurantes como sitios gastronómicos.

4.2 Prueba de hipótesis

4.2.1 Prueba de Chi 2

Este siguiente código realiza la prueba de chi-cuadrado de independencia entre variables categóricas y crea tablas de contingencia entre las variables "categoría" y "barrio" y entre "cocina" y "barrio" utilizando la función "crosstab" de pandas. Luego, utiliza la función "chi2_contingency" de scipy.stats para realizar la prueba de chi-cuadrado y obtener los valores de chi-cuadrado y p. Finalmente, imprime los resultados para ambas tablas de contingencia.

In [37]:

```
from scipy.stats import chi2_contingency
# Crear una tabla de contingencia entre la categoría y el barrio
ctg_barrio = pd.crosstab(df['categoria'], df['barrio'])

# Realizar la prueba de chi-cuadrado para la tabla de contingencia
chi2_ctg_barrio, p_ctg_barrio, _, _ = chi2_contingency(ctg_barrio)

# Imprimir los resultados
print('Chi-cuadrado para la tabla de contingencia entre categoría y barrio:', chi2_ctg_barrio)
print('Valor p para la tabla de contingencia entre categoría y barrio:', p_ctg_barrio)

# Crear una tabla de contingencia entre la cocina y el barrio
coc_barrio = pd.crosstab(df['cocina'], df['barrio'])

# Realizar la prueba de chi-cuadrado para la tabla de contingencia
chi2_coc_barrio, p_coc_barrio, _, _ = chi2_contingency(coc_barrio)

# Imprimir los resultados
print('Chi-cuadrado para la tabla de contingencia entre cocina y barrio:', chi2_coc_barrio)
print('Valor p para la tabla de contingencia entre cocina y barrio:', p_coc_barrio)
```

```
Chi-cuadrado para la tabla de contingencia entre categoría y barrio: 710.7999068087462
Valor p para la tabla de contingencia entre categoría y barrio: 2.2443631477357643e-31
Chi-cuadrado para la tabla de contingencia entre cocina y barrio: 2804.373883744627
Valor p para la tabla de contingencia entre cocina y barrio: 1.4013325810103067e-10
```

El resultado del chi-cuadrado indica que existe una relación significativa entre las variables de la tabla de contingencia. Es decir, la distribución de restaurantes en función de su categoría y cocina varía en función de su ubicación geográfica. Un valor alto de chi-cuadrado (710.80 para la tabla de contingencia entre categoría y barrio y 2804.37 para la tabla de contingencia entre cocina y barrio) indica que las frecuencias observadas difieren significativamente de las frecuencias esperadas, lo que sugiere una asociación entre las variables.

El valor p indica la probabilidad de obtener un resultado tan extremo como el observado si la hipótesis nula de independencia de las variables es verdadera. En este caso, los valores de p son muy pequeños (2.24e-31 para la tabla de contingencia entre categoría y barrio y 1.40e-10 para la tabla de contingencia entre cocina y barrio), lo que sugiere que la hipótesis nula puede ser rechazada y que la asociación entre las variables es significativa.

En resumen, los resultados indican que hay una relación significativa entre la diversidad culinaria (categoría y cocina) y la ubicación geográfica de los restaurantes (barrio). Esto sugiere que la ubicación geográfica puede ser un factor importante a considerar al momento de decidir qué tipo de restaurante abrir y qué tipo de cocina ofrecer.

4.3 Resultado de la aplicación del método de Elbow y Silueta para Kmeans

El siguiente código realiza clustering KMeans en un conjunto de datos geográficos, usando la inercia y la silueta para encontrar el número óptimo de clusters y visualiza los resultados en un scatter plot con etiquetas de cluster.

In [48]:

```
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

```

# Seleccionar las características
X = df[['lat', 'long']]

# Estandarizar los datos
X = (X - X.mean()) / X.std()

# Calcular la inercia y la silueta para diferentes valores de k
inertias = []
silhouettes = []
k_values = range(2, 10)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(X)
    inertias.append(kmeans.inertia_)
    silhouettes.append(silhouette_score(X, kmeans.predict(X)))

# Visualizar la inercia y la silueta en función del número de clusters
fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('Número de clusters')
ax1.set_ylabel('Inercia', color=color)
ax1.plot(k_values, inertias, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel('Silueta', color=color)
ax2.plot(k_values, silhouettes, color=color)
ax2.tick_params(axis='y', labelcolor=color)

ax1.set_title('Elbow Method y Silueta para KMeans')
ax1.legend(['Inercia'], loc='upper left')
ax2.legend(['Silueta'], loc='upper right')

fig.tight_layout()
plt.show()

# Entrenar el modelo con el número óptimo de clusters
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(X)

# Añadir las etiquetas de los clusters al dataframe
df['cluster'] = kmeans.labels_

# Visualizar los resultados en un scatter plot
plt.scatter(X['long'], X['lat'], c=df['cluster'])
plt.xlabel('Longitud')
plt.ylabel('Latitud')
plt.title('Resultados del modelo de clustering KMeans')
plt.legend(title='Clusters')
plt.show()

```

```

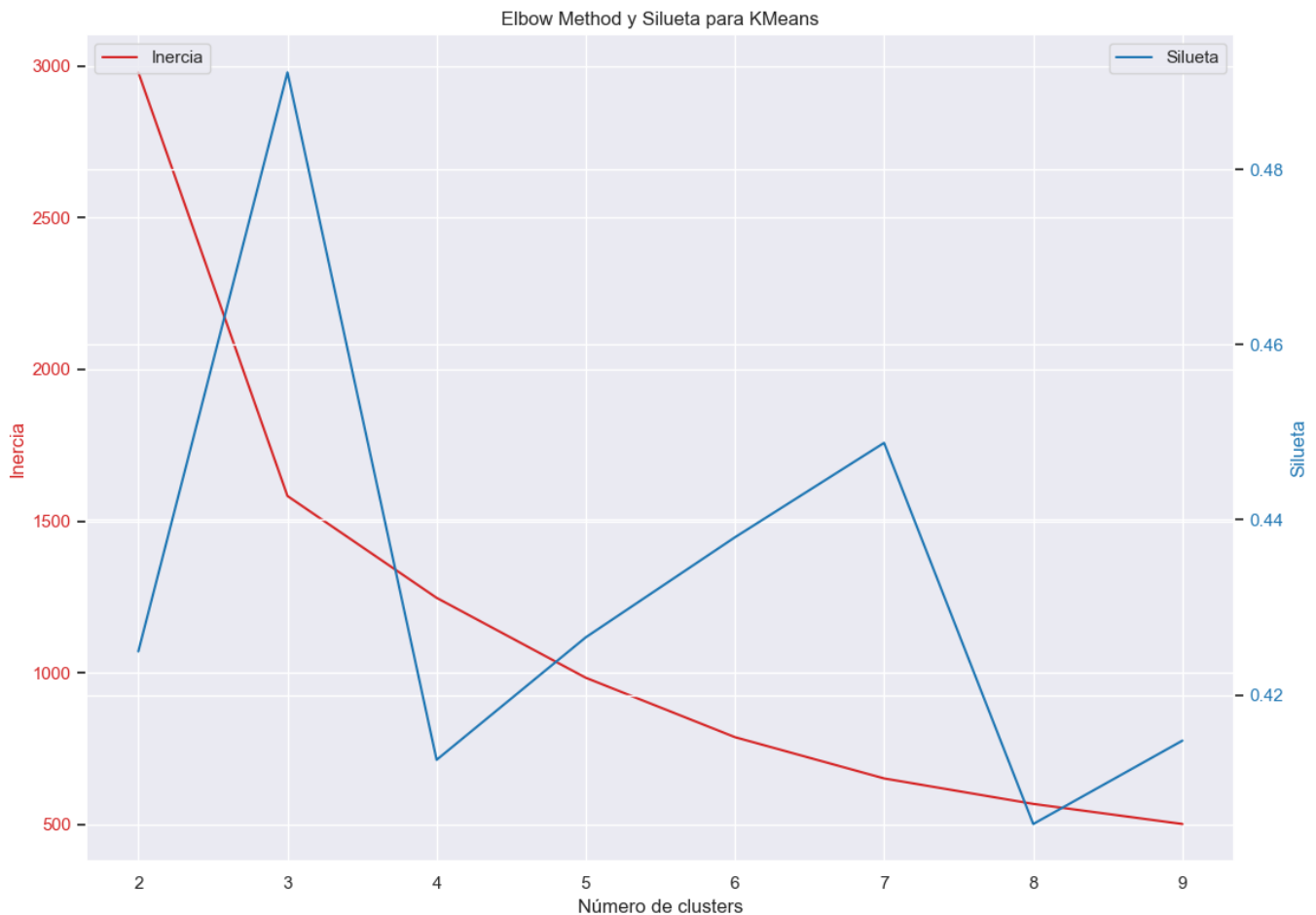
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(

```

```

ster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\clu
ster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\clu
ster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(
c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\clu
ster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(

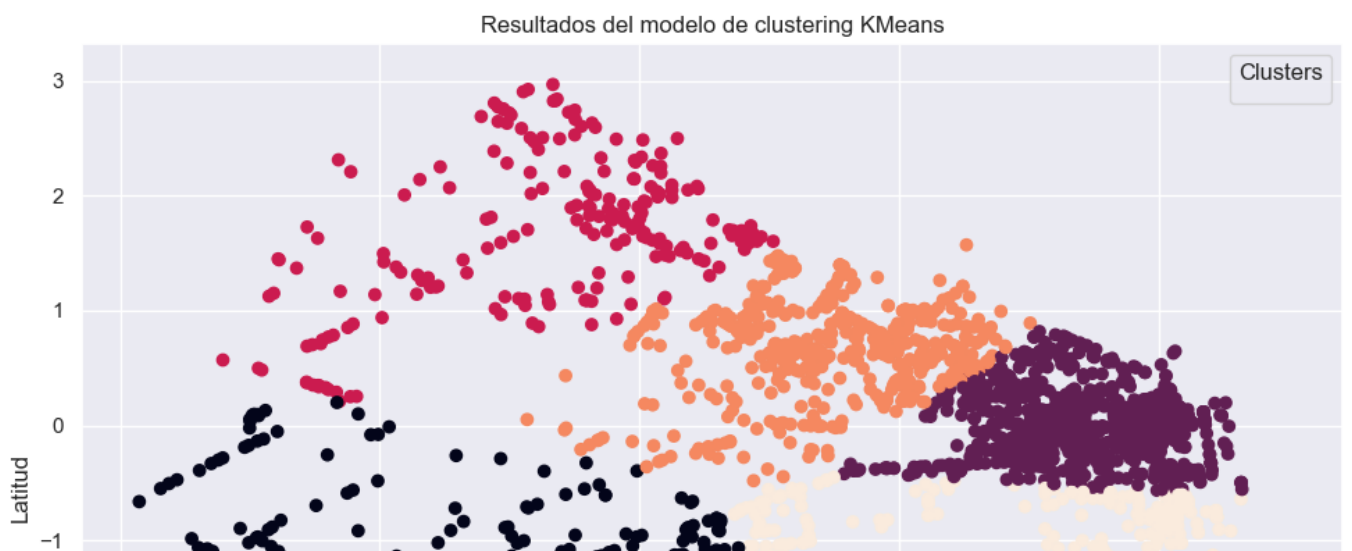
```

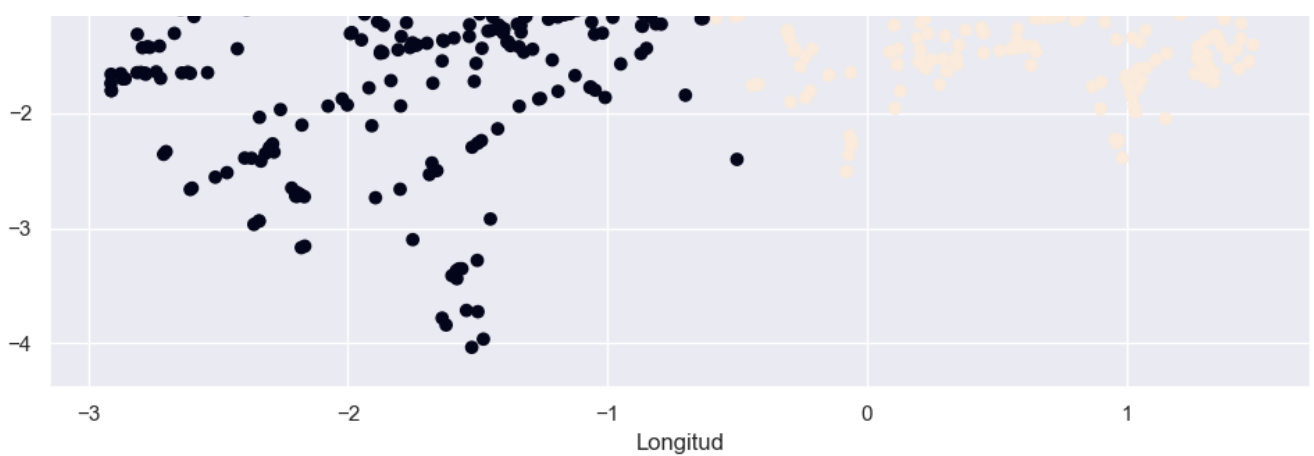


```

c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\clu
ster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(
No artists with labels found to put in legend. Note that artists whose label start
with an underscore are ignored when legend() is called with no argument.

```





La curva de Elbow Method indica que la inercia disminuye a medida que aumenta el número de clusters, lo que sugiere que hay una tendencia a la formación de grupos más pequeños de restaurantes con características similares. Sin embargo, no hay un punto claro de inflexión en la curva, lo que hace que sea difícil determinar el número óptimo de clusters.

La curva de Silueta muestra que los clusters tienen una variación en la densidad y separación de los datos, con algunos clusters con una silueta más alta que otros. Los valores de silueta sugieren que el modelo puede agrupar los restaurantes en clusters de 2 a 4 y de 7 a 9, mientras que los clusters de 5 y 6 tienen valores de silueta más bajos.

El modelo de clustering Kmeans se ajustó a 4 clusters, lo que sugiere que los datos se pueden agrupar en 4 grupos diferentes. Los puntos de color en el gráfico indican la ubicación geográfica de los restaurantes en cada cluster. La densidad de los puntos en cada cluster indica que hay áreas con más restaurantes, que pueden tener características similares en términos de tipo de comida, estilo de restaurante, precios, etc.

En general, se puede inferir que hay áreas en Buenos Aires con concentraciones de restaurantes que comparten características similares, pero es necesario un análisis más detallado para determinar qué características son esas y cómo se relacionan con la gastronomía de la ciudad.

In [124]:

```
import geopandas as gpd
from sklearn.cluster import KMeans
from ipyleaflet import Map, CircleMarker, WidgetControl, FullScreenControl, LayerGroup, LayersControl
from ipywidgets import HTML

# Leer los datos
gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.long, df.lat))

# crear un objeto KMeans con cuatro clusters
kmeans = KMeans(n_clusters=4)

# ajustar los datos y obtener las etiquetas de los clusters
clusters = kmeans.fit_predict(gdf[['lat', 'long']])

# imprimir el número de clusters encontrados
print("Número de clusters encontrados:", len(set(clusters)))

# asignar una etiqueta de cluster a cada punto
gdf['cluster'] = clusters

# colores para cada cluster
colors = {0:'red', 1:'blue', 2:'green', 3:'purple'}

# crear el mapa
m2 = Map(center=(-34.6053795, -58.3843273), zoom=12)

# agregar capas de puntos para cada cluster
for i in range(len(kmeans.cluster_centers_)):
    center = kmeans.cluster_centers_[i]
```



```

    color = colors[i]
    marker = CircleMarker(location=(center[0], center[1]), color=color, fill_color=c
olor, radius=10)
    marker.tooltip = f"Cluster {i}"
    m2.add_layer(marker)

    # agregar puntos a la capa correspondiente
    cluster_points = gdf[gdf['cluster'] == i]
    for _, row in cluster_points.iterrows():
        lat = row['lat']
        long = row['long']
        color = colors[row['cluster']]
        marker = CircleMarker(location=(lat, long), color=color, fill_color=color, r
adius=3)
        marker.tooltip = row['nombre']
        popup_content = f"<b>{row['nombre']}</b><br><b>Categoría:</b> {row['categori
a']}<br><b>Cocina:</b> {row['cocina']}<br><b>Cluster:</b> {row['cluster']}"
        marker.popup = HTML(popup_content)
        m2.add_layer(marker)

# Crear leyenda personalizada
legend_box = Box(
    children=[
        HTML(value=f"<span style='color:{color};'>&#9679;</span> Cluster {i}")
        for i, color in colors.items()
    ],
    layout={
        'border': '1px solid black',
        'padding': '5px',
        'width': '200px'
    }
)
legend = WidgetControl(widget=legend_box, position='bottomright')
m2.add_control(legend)

# Agregar controles adicionales al mapa
m2.add_control(FullScreenControl())
m2.add_control(LayersControl())

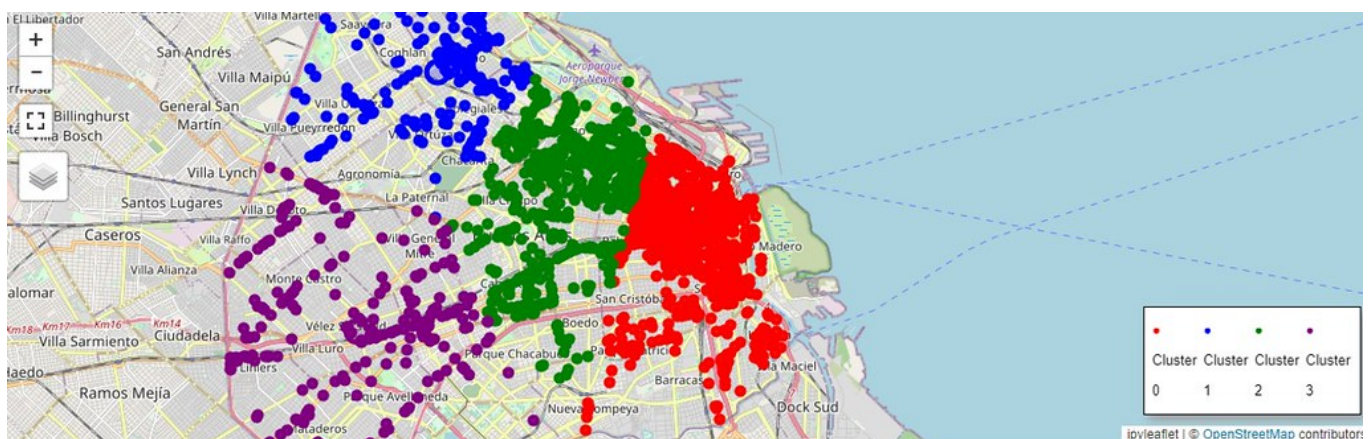
# mostrar el mapa
m2

```

c:\Users\wille\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
warnings.warn(

Número de clusters encontrados: 4

Out[124]:



En el mismo mapa se puede observar el agrupamiento de los establecimientos, pudiendose evidenciar que la region gastronomica se encuentra al este de la ciudad, comprendiendo los barrios de Recoleta, Retiro, San Nicolas, Monserrat, Balvanera, San Cristobal, Constitucion, San Telmo, La Boca, Puerto Madero. Palermo. Almaadro. Caballito. Villa Crespo v los de menor densidad en el lado oeste.

V. Debate

Los resultados del proyecto muestran una relación significativa entre la diversidad culinaria y la ubicación geográfica de los restaurantes en Buenos Aires. Es interesante observar que los barrios de Balvanera, San Nicolás, Recoleta y Palermo presentan una alta concentración de negocios de comida y bebida, especialmente en la categoría de restaurantes y cafés. Además, la categoría de restaurante es la más predominante en la ciudad, seguida por la de café y bar.

Estos hallazgos sugieren que los empresarios que planean abrir un negocio de comida y bebida en Buenos Aires deberían considerar la ubicación geográfica y la categoría de restaurante al decidir qué tipo de cocina ofrecer. Por ejemplo, si alguien quiere abrir un restaurante en la zona de Palermo, debería considerar ofrecer comida internacional o regional para diferenciarse de la alta concentración de restaurantes de la zona.

La curva de Elbow Method y la curva de Silueta mostraron que el modelo de clustering Kmeans se ajustó a 4 clusters. Estos clusters sugieren que hay grupos de restaurantes con características similares en diferentes áreas de la ciudad. Por lo tanto, los empresarios podrían considerar explorar las características de los restaurantes en cada cluster para identificar oportunidades de mercado en cada área.

En general, los resultados indican que la ubicación geográfica es un factor importante a considerar al momento de decidir qué tipo de restaurante abrir y qué tipo de cocina ofrecer. Es importante mencionar que, aunque hay zonas con alta concentración de restaurantes, también existen áreas con baja presencia de negocios de comida y bebida, como Villa Luro, Villa Santa Rita y Villa Soldati, lo que podría presentar oportunidades para nuevos emprendimientos.

VI. CONCLUSIONES

- Se observa una alta concentración de negocios de comida y bebida en los barrios de Balvanera, San Nicolás, Recoleta y Palermo, especialmente en la categoría de restaurantes y cafés. Esto indica que estos barrios pueden ser lugares ideales para abrir un negocio de comida y bebida debido a la alta demanda en la zona.
- La mayoría de los locales ofrecen comida del tipo "Minutas", seguida de la categoría "Internacional" y "Tradicional". Esto puede ser útil para los emprendedores que buscan abrir un negocio de comida y bebida en la ciudad, ya que pueden optar por ofrecer este tipo de comida para satisfacer la demanda del mercado.
- El valor del chi-cuadrado indica que existe una relación significativa entre las variables de la tabla de contingencia, lo que sugiere que la ubicación geográfica puede ser un factor importante a considerar al momento de decidir qué tipo de restaurante abrir y qué tipo de cocina ofrecer. Por lo tanto, se recomienda que los emprendedores presten atención a la ubicación geográfica y a las preferencias culinarias de los clientes potenciales antes de abrir un negocio de comida y bebida en la ciudad.
- La curva de Elbow Method sugiere que hay una tendencia a la formación de grupos más pequeños de restaurantes con características similares. Aunque no hay un punto claro de inflexión en la curva, se recomienda que los emprendedores consideren agrupar los restaurantes en clusters para identificar áreas con características similares y analizar la competencia en la zona.
- La curva de Silueta muestra que los clusters tienen una variación en la densidad y separación de los datos, con algunos clusters con una silueta más alta que otros. Se recomienda que los emprendedores presten atención a la densidad de restaurantes en los clusters con una silueta más alta para identificar áreas con alta demanda y características similares en términos de tipo de comida, estilo de restaurante, precios, etc.
- El modelo de clustering Kmeans se ajustó a 4 clusters, lo que sugiere que los datos se pueden agrupar en 4 grupos diferentes. Los puntos de color en el gráfico indican la ubicación geográfica de los restaurantes en cada cluster. Se recomienda que los emprendedores consideren la ubicación geográfica de su negocio y la competencia en la zona antes de decidir en qué cluster ubicarse.
- En general, se puede inferir que hay áreas en Buenos Aires con concentraciones de restaurantes que comparten características similares, pero es necesario un análisis más detallado para determinar qué características son esas y cómo se relacionan con la gastronomía de la ciudad. Por lo tanto, se

recomienda que los emprendedores realicen un análisis más detallado de las características de los restaurantes en cada cluster antes de tomar una decisión sobre el tipo de negocio y la ubicación geográfica.