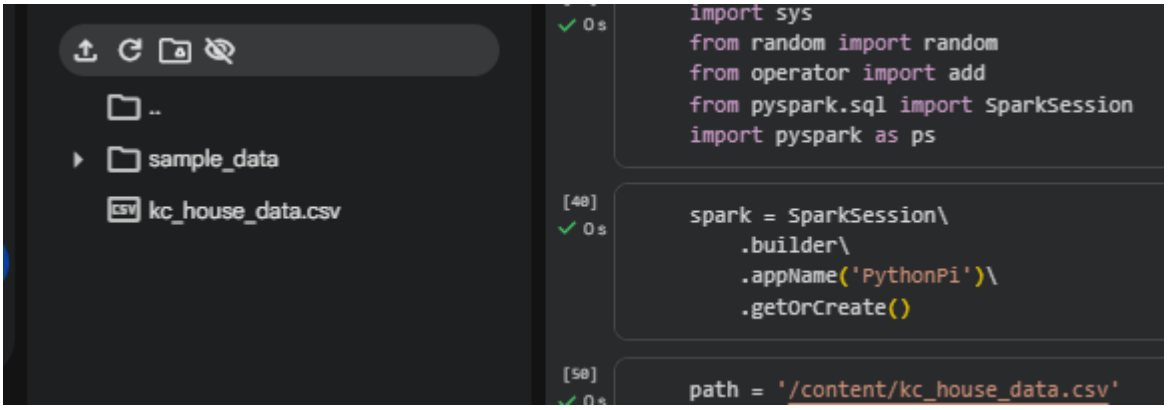
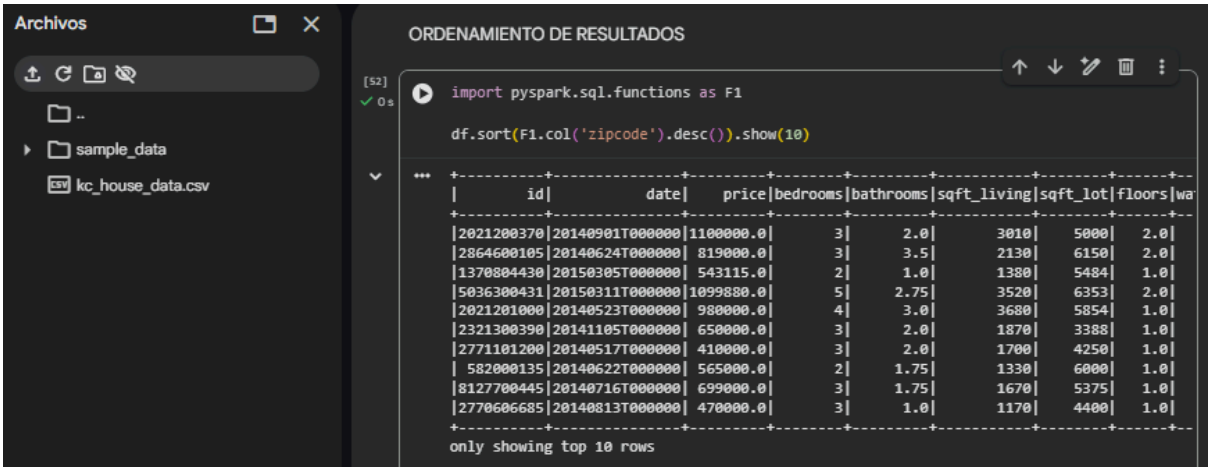
~ Configuración de plataforma Spark

En mi caso utilice google colab, ya que en la sección de comentarios del módulo encontre que es muy dificil configurar spark en Windows 11.

~ Importación de datos de Housing a una estructura Spark



~ Selección de datos de housing con filtros simples:
1) listado completo de columnas ordenado por zipcode



2) para el zipcode con mayor número de casas, calcular el promedio de precio, y tamaño en m2

En este caso use una tabla general, pero podemos ver en el top1 que es el zipcode correcto

```
# Conversión de pies² a m²
SQFT_TO_M2 = 0.092903

# Estadísticas por zipcode
stats_zipcode = (
    df.groupBy('zipcode')
      .agg(
          F.round(F.avg('price'), 2).alias('PrecioPromedio'),
          F.round(F.avg(df['sqft_living'] * SQFT_TO_M2), 2).alias('Promedio_m2')
      )
)

# Conteo de casas por zipcode
zipcode_counts = df.groupBy('zipcode').count()

# Unir estadísticas + conteo
stats_conteo = stats_zipcode.join(zipcode_counts, on='zipcode')

# Ordenar por mayor número de casas
stats_ordenado = stats_conteo.orderBy(F.col('count').desc())

print("Estadísticas por Zipcode (ordenado por número de casas):")
stats_ordenado.show()
```

```
Estadísticas por Zipcode (ordenado por número de casas):
+-------+-------------+-----------+-----+
|zipcode|PrecioPromedio|Promedio_m2|count|
+-------+-------------+-----------+-----+
|98103.0|     584919.21|     153.37|  602|
|98038.0|      366867.6|     199.53|  590|
|98115.0|     619900.55|      170.5|  583|
|98052.0|     645231.46|     219.59|  574|
|98117.0|     576795.01|      157.2|  553|
```

```python
# Promedio de precio por zipcode
precio_promedio_zipcode = df.groupBy('zipcode').agg(F1.round(F1.avg('price'),2).alias('Precio Promedio'))
# Visualizar
print('Precio promedio por Zipcode:')
precio_promedio_zipcode
precio_promedio_zipcode.show()
```

```
Precio promedio por Zipcode:
+-------+--------------+
|zipcode|Precio Promedio|
+-------+--------------+
|  98002|      234284.04|
|  98155|       423725.7|
|  98198|      302878.88|
|  98146|      359483.24|
|  98122|      634360.18|
|  98077|      682774.88|
|  98006|      859684.78|
|  98001|      280804.69|
|  98005|      810164.88|
|  98112|     1095499.34|
|  98115|      619900.55|
|  98059|      493552.53|
|  98075|      790576.65|
|  98023|      286732.79|
|  98109|      879623.62|
|  98136|      551688.67|
|  98052|      645231.46|
|  98011|      490351.47|
|  98014|      455617.11|
|  98058|      353608.64|
+-------+--------------+
only showing top 20 rows
```

Agrupamiento en Spark, por número de habitaciones y baños, del precio.

```python
df_agrupado = df.groupBy('zipcode', 'bedrooms', 'bathrooms').agg(
    F.round(F.avg('price'), 2).alias('PrecioPromedio')
)

df_agrupado.show()
```

```
+-------+--------+---------+--------------+
|zipcode|bedrooms|bathrooms|PrecioPromedio|
+-------+--------+---------+--------------+
|98119.0|     3.0|      1.0|     681881.25|
|98040.0|     3.0|      2.5|      889000.0|
|98030.0|     4.0|      2.5|     347197.22|
|98042.0|     4.0|     2.25|     371188.46|
|98122.0|     4.0|      3.0|      664125.0|
|98052.0|     3.0|      2.0|     517635.36|
|98058.0|     4.0|     3.25|      583000.0|
|98065.0|     2.0|      2.5|      786000.0|
|98178.0|     1.0|     0.75|      231000.0|
|98040.0|     5.0|     2.75|     1225587.0|
```