



UNIVERSIDAD  
DE  
CÓRDOBA

**Instituto de Estudios de Posgrado  
Universidad de Córdoba**

**MÁSTER UNIVERSITARIO EN INTELIGENCIA COMPUTACIONAL E  
INTERNET DE LAS COSAS**

**TRABAJO CDR'S DE MILÁN**

*Análisis, Diseño y Procesamiento de Datos Aplicados a las Ciencias y a las  
Tecnologías*

**Autora:**

Alba Márquez-Rodríguez

**Profesores:**

Gonzalo Cerruela García  
Dormingo Ortiz Boyer  
Juan A. Romero del Castillo

Córdoba, Noviembre 2023

# Índice

1. Enunciado	2
2. Datos y Carga	3
3. Encuentra los países con los que se interactúa	6
4. Encuentra que país es con el que más se interactúa ademas de Italia	8
5. ¿Qué celda comunica más con el extranjero?	9
6. Encuentra la celda con más actividad de smsin, smsout callin, callout, internet y la total	10
7. Crea una colección con un documento por celda en el que aparezcan los acumulados de los diferentes campos	13
8. Crea una colección con una documento por celda y hora en el que aparezcan los acumulados de los diferentes campos	15
9. Realiza un estudio de las celdas indicadas	17
9.1. Celdas Seleccionadas . . . . .	17

# 1. Enunciado

Para profundizar en el uso de MongoDB para el tratamiento de datos procedentes de los registros de datos o CDR (Call Detail Record) almacenados por las operadoras de telecomunicaciones, utilizaremos los datos de la ciudad de Milán tomados por Telecom Italia desde el 1 de noviembre de 2013 hasta el 1 de enero de 2014 disponibles en el siguiente enlace:

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV>

Cada alumno elegirá una semana completa de datos que incluya la totalidad de los datos tomados de lunes a domingo y realizará las siguientes acciones:

1. Descargar los ficheros “.txt” correspondientes y crear un programa en Python para cargar los datos en la colección “Milan\_CDR\_c” de la base de datos “Milan\_CDR\_db”.
2. Encuentra los países con los que se interactúa.
3. Encuentra que país es con el que más se interactúa además de Italia
4. ¿Qué celda comunica más con el extranjero?
5. Encuentra la celda con más actividad de smsin, smsout callin, callout, internet y la total
6. Crea una colección con una documento por celda en el que aparezcan los acumulados de los diferentes campos.
7. Crea una colección con una documento por celda y hora en el que aparezcan los acumulados de los diferentes campos.
8. Realiza un estudio de las celdas 4259 (Bocconi), 4456 (Navigli), 5060 (Duomo), 1419 (terreno agrícola), 2436 (área industrial), 4990 (aeropuerto de Linate), 945 (residencial aislado) y 5048 (residencial céntrico)

## 2. Datos y Carga

Los datos con los que se trabajará son los datos comprendidos entre los días 16 y 22 de diciembre de 2013 del conjunto de datos de CDR's Milan [1]. Se han descargado los archivos .TXT correspondientes del enlace proporcionado en el enunciado de la práctica.

Los datos descargados están en formato .TXT con los valores separados por tabulaciones, en la figura 1 se muestra el contenido de uno de los archivos .TXT para ver su formato:

```
1 1387148400000 0 0.07957531340575938
1 1387148400000 39 0.21607763779169026 0.3479277997260229 0.05343788914147278
0.026137424264286602 9.806615168593751
1 1387149000000 0 0.08368870473705797
1 1387149000000 39 0.3740652239903095 0.21375155656589112 0.052274848528573205
9.859893820650818
1 1387149000000 86 0.02730046487718618
```

Figura 1: Contenido del archivo `sms-call-internet-mi-2013-12-16.txt`

Tras transformarlo en un archivo CSV en el que aquellos datos que no se tenían fueron ocupados por un valor *null* quedó como se muestra en la figura 2.

```
1,1387148400000,0,0.07957531340575938,null,null,null,null
1,1387148400000,39,0.21607763779169026,0.3479277997260229,0.05343788914147278,0.026137424264286602,9.806615168593751
1,1387149000000,0,0.08368870473705797,null,null,null,null
1,1387149000000,39,0.3740652239903095,0.21375155656589112,0.052274848528573205,null,9.859893820650818
1,1387149000000,86,null,null,null,0.02730046487718618,null
```

Figura 2: Contenido del archivo `sms-call-internet-mi-2013-12-16.csv`

Por último se ha transformado en un archivo .JSON con el formato que se muestra en la figura 3.

```
[
  {
    "cellid": 1.0,
    "time": 1387148400000.0,
    "countrycode": 0.0,
    "smsin": 0.07957531340575938,
    "smsout": "null",
    "callin": "null",
    "callout": "null",
    "Internet": "null"
  },
  {
    "cellid": 1.0,
    "time": 1387148400000.0,
    "countrycode": 39.0,
    "smsin": 0.21607763779169026,
    "smsout": 0.3479277997260229,
    "callin": 0.05343788914147278,
    "callout": 0.026137424264286602,
    "Internet": 9.806615168593751
  },
  {
    "cellid": 1.0,
    "time": 1387149000000.0,
    "countrycode": 0.0,
    "smsin": 0.08368870473705797,
    "smsout": "null",
    "callin": "null",
    "callout": "null",
    "Internet": "null"
  },
  {
    "cellid": 1.0,
    "time": 1387149000000.0,
    "countrycode": 39.0,
    "smsin": 0.3740652239903095,
    "smsout": 0.21375155656589112,
    "callin": 0.052274848528573205,
    "callout": "null",
    "Internet": 9.859893820650818
  },
  {
    "cellid": 1.0,
    "time": 1387149000000.0,
    "countrycode": 86.0,
    "smsin": "null",
    "smsout": "null",
    "callin": "null",
    "callout": 0.02730046487718618,
    "Internet": "null"
  },
  ...
]
```

Figura 3: Contenido del archivo sms-call-internet-mi-2013-12-16.json

Finalmente, a partir de los archivos .JSON se utiliza un programa de python que los carga en la colección *Milan\_CDR\_c* de la base de datos *Milan\_CDR.db*. En la figura 14 se muestra la base de datos con la información cargada.

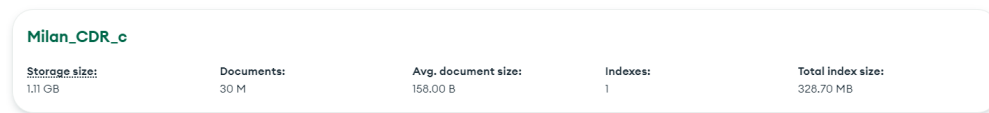


Figura 4: Base de Datos en MongoDB

El código utilizado para cargar los datos de los archivos .JSON en la base de datos es el siguiente, proporcionado por uno de los profesores de la asignatura:

Figura 5: Código para subir los datos a la Base De Datos

### 3. Encuentra los países con los que se interactúa

Para obtener los países con los que se interactúa se ha realizado una búsqueda con el siguiente código:

```
pipeline = [
    {"$group": {"_id": "$countrycode"}},
    {"$sort": SON([["_id", 1]])}
]

distinct_countries = Milan_CDR_c.aggregate(pipeline)
distinct_countries = [doc['_id'] for doc in distinct_countries]
print("Países con los que se interactúa:", distinct_countries)
```

Figura 6: Código para obtener los países con los que se interactúa

La primero agrupa los documentos según el campo `countrycode`. Como resultado, se obtiene un conjunto de documentos que representan países distintos con los que se ha interactuado. Después se ordena los resultados en orden ascendente según el campo `_id`, asegurando que la lista de países esté ordenada alfanuméricamente, así estarán ordenados de menor número del código de país a mayor.

Así se obtiene el código de los países que han tenido alguna interacción. El resultado es el siguiente:

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 66, 81, 82, 84, 86, 90, 91, 92, 93, 94, 95, 98, 211, 212, 213, 216, 218, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 291, 297, 298, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 370, 371, 372, 373, 374, 375, 376, 377, 378, 380, 381, 382, 385, 386, 387, 389, 420, 421, 423, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 670, 672, 673, 675, 676, 677, 678, 679, 680, 681, 687, 688, 689, 690, 850, 852, 853, 855, 856, 870, 880, 881, 886, 960, 961, 962, 963, 964, 965, 966, 967, 968, 970, 971, 972, 973, 974, 975, 976, 977, 992, 993, 994, 995, 996, 998, 1139, 1204, 1214, 1226, 1242, 1246, 1250, 1264, 1268, 1284, 1289, 1306, 1340, 1345, 1403, 1416, 1418, 1438, 1441, 1450, 1473, 1506, 1514, 1519, 1579, 1581, 1587, 1604, 1613, 1641, 1647, 1649, 1670, 1671, 1684, 1705, 1709, 1721, 1758, 1767, 1778, 1780, 1784, 1787, 1807, 1808, 1809, 1819, 1829, 1849, 1902, 1905, 1907, 1926, 1929, 1939, 7700, 7701, 7702, 7705, 7707, 7710, 7711, 7712, 7713, 7714, 7715, 7717, 7721, 7725, 7727, 7729, 7771, 7775, 7776, 7777, 7778, 8816, 12684, 12687, 12845, 14413, 14415, 14417, 17672, 18092, 18093, 18094, 18096, 18097, 18098, 18099, 18682, 18683, 18684, 18686, 18762, 18763, 18764, 18768, 18769, 50936, 50937, 50938, 50947, 50948, 88216, 88239, 97256, 97259]

Esto quiere decir que, teléfonos con el código de los países anteriormente indicados han hecho llamada, han recibido llamada, han enviado un SMS, han recibido un SMS o han hecho uso de Internet estando en la zona de dónde se han recogido los datos.

Entre estos valores hay algunos que parecen ser erróneos pues son demasiado grandes como para ser un código de un país. En un principio puede parecer que haya habido un problema al convertir los datos en otro tipo de ficheros o al subirlos a la base de datos, pero si se acude a los archivos originales en formato .TXT y se busca, por ejemplo, el último valor, se encuentra que efectivamente existen este tipo de valores en `countrycode`. En la figura 7 podemos ver uno de estos valores en uno de los archivos .TXT.

```

1765206 L387224000000 380 0.47563738597862076
1765207 L387224000000 39 25.865849873395398 14.928667475838672 26.852608839570753 33.140951918550066 631.8886738883117
1765208 L387224000000 44 0.47563738597862076 0.3157432106229328
1765209 L387224000000 49 0.3386447697064928
1765210 L387224000000 97259 0.12102679910652635
1765211 L387224600000 0 9.842145585259575 0.3157432106229328
1765212 L387224600000 20 0.2638313797847479
1765213 L387224600000 34 1.502328764522357
1765214 L387224600000 39 24.541414791551986 24.985001423012893 23.781747170181205 21.887626901034433 625.6902943524742

```

Figura 7: Valores archivo .TXT



## 4. Encuentra que país es con el que más se interactúa además de Italia

Para encontrar que país es con el que más se interactúa además de Italia se ha realizado una búsqueda con el siguiente código. Se ha tenido que excluir para ello el código del país perteneciente a Italia (39) y el 0 que es para llamadas internas y no pertenece a ningún país.

```
# Italia = 39
# Llamadas internas = 0
excluded_countries = [39, 0]

pipeline_top_countries = [
    {"$match": {"countrycode": {"$nin": excluded_countries}}},
    {"$group": {"_id": "$countrycode", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}},
    {"$limit": 1}
]

top_interacted_country = Milan_CDR_c.aggregate(pipeline_top_countries)
top_interacted_country = list(top_interacted_country)[0]['_id']

print("País con el que más se interactúa:", top_interacted_country)
```

Figura 8: Código para encontrar el país con el que más se interactúa

Primero, se define la lista `excluded_countries` que contiene los códigos de país que deben excluirse de la consideración (Italia y llamadas internas).

Luego, se crea un pipeline de agregación en MongoDB para realizar la siguiente secuencia de operaciones:

1. Filtrar los documentos en la colección `Milan_CDR_c` para excluir aquellos con códigos de país presentes en la lista `excluded_countries`.
2. Agrupar los documentos resultantes según el campo `countrycode`.
3. Contar la cantidad de documentos para cada código de país.
4. Ordenar los resultados en orden descendente según la cantidad de interacciones.
5. Limitar la salida a solo el primer documento (el país con más interacciones).

Así se obtiene el código de los países que han tenido alguna interacción. El resultado es el siguiente:

País con el que más se interactúa (además de Italia (39) y llamadas internas (0)): 49

El país con el que más se interactúa es el que tiene el código 49, que es Alemania.

## 5. ¿Qué celda comunica más con el extranjero?

Para encontrar que la celda que más se comunica con el extranjero se ha realizado una búsqueda con el siguiente código. Se ha tenido que excluir para ello el código del país perteneciente a Italia (39) y el 0 que es para llamadas internas y no pertenece a ningún país.

```
# Italia = 39
# Llamadas internas = 0
excluded_countries = [39, 0]

pipeline_foreign_cell = [
    {"$match": {"countrycode": {"$nin": excluded_countries}}},
    {"$group": {"_id": "$cellid", "count": {"$sum": 1}}},
    {"$sort": SON([("count", -1)])},
    {"$limit": 1}
]

most_foreign_communicating_cell = Milan_CDR_c.aggregate(pipeline_foreign_cell)
most_foreign_communicating_cell = list(most_foreign_communicating_cell)[0]['_id']
print("Celda que más comunica con el extranjero:", most_foreign_communicating_cell)
```

Figura 9: Código para encontrar la celda que más interactúa con el extranjero

Primero, se define la lista `excluded_countries` que contiene los códigos de país que deben excluirse de la consideración (Italia y llamadas internas).

Luego, se crea un pipeline de agregación en MongoDB para realizar la siguiente secuencia de operaciones:

1. Filtrar los documentos en la colección `Milan_CDR_c` para excluir aquellos con códigos de país presentes en la lista `excluded_countries`.
2. Agrupar los documentos resultantes según el campo `cellid`.
3. Contar la cantidad de documentos para cada celda.
4. Ordenar los resultados en orden descendente según la cantidad de interacciones.
5. Limitar la salida a solo el primer documento (la celda que más comunica con el extranjero).

Así se obtiene la celda que ha tenido más interacciones con países que no son ni Italia (39) ni llamadas internas (0):

Celda que más comunica con el extranjero: 6064

La celda que más interactúa con el extranjero es 6064.

## **6. Encuentra la celda con más actividad de smsin, smsout callin, callout, internet y la total**

Para encontrar que la celda con más actividad de smsin, smsout callin, callout, internet y la total se ha realizado una búsqueda con el siguiente código:

```

# Encuentra la celda con más actividad de smsin
pipeline_smsin = [
    {"$group": {"_id": "$cellid", "total_smsin": {"$sum": "$smsin"}}},
    {"$sort": {"total_smsin": -1}},
    {"$limit": 1}
]

most_active_smsin_cell = Milan_CDR_c.aggregate(pipeline_smsin)
most_active_smsin_cell = list(most_active_smsin_cell)[0]['_id']

# Encuentra la celda con más actividad de smsout
pipeline_smsout = [
    {"$group": {"_id": "$cellid", "total_smsout": {"$sum": "$smsout"}}},
    {"$sort": {"total_smsout": -1}},
    {"$limit": 1}
]

most_active_smsout_cell = Milan_CDR_c.aggregate(pipeline_smsout)
most_active_smsout_cell = list(most_active_smsout_cell)[0]['_id']

# Encuentra la celda con más actividad de callin
pipeline_callin = [
    {"$group": {"_id": "$cellid", "total_callin": {"$sum": "$callin"}}},
    {"$sort": {"total_callin": -1}},
    {"$limit": 1}
]

most_active_callin_cell = Milan_CDR_c.aggregate(pipeline_callin)
most_active_callin_cell = list(most_active_callin_cell)[0]['_id']

# Encuentra la celda con más actividad de callout
pipeline_callout = [
    {"$group": {"_id": "$cellid", "total_callout": {"$sum": "$callout"}}},
    {"$sort": {"total_callout": -1}},
    {"$limit": 1}
]

most_active_callout_cell = Milan_CDR_c.aggregate(pipeline_callout)
most_active_callout_cell = list(most_active_callout_cell)[0]['_id']

# Encuentra la celda con más actividad de Internet
pipeline_internet = [
    {"$group": {"_id": "$cellid", "total_internet": {"$sum": "$Internet"}}},
    {"$sort": {"total_internet": -1}},
    {"$limit": 1}
]

most_active_internet_cell = Milan_CDR_c.aggregate(pipeline_internet)
most_active_internet_cell = list(most_active_internet_cell)[0]['_id']

# Encuentra la celda con más actividad total
pipeline_total_activity = [

```

Este código utiliza operaciones de agregación en MongoDB para calcular la suma de distintas métricas (SMS entrantes, SMS salientes, llamadas entrantes, llamadas salientes, uso de Internet) agrupadas por celda. Luego, se selecciona la celda con el valor máximo en cada métrica.

Las celdas con más actividad en cada caso obtenidas son las siguientes:

```
Celda con más actividad de smsin: 5161
Celda con más actividad de smsout: 5059
Celda con más actividad de callin: 5059
Celda con más actividad de callout: 5059
Celda con más actividad de Internet: 5161
Celda con más actividad total: 5161
```

Estos resultados quieren decir que la celda que más sms entrantes ha tenido ha sido 5191 mientras que la que más sms salientes ha tenido ha sido 5059. Por otro lado la que más llamadas ha recibido ha sido 5059 y la que más llamadas ha realizado también ha sido 5059. La celda con más actividad de Internet ha sido la celda 5161. Teniendo en cuenta todas las acciones posibles, la celda con más actividades (sms entrante y saliente, llamada entrante y saliente e Internet) ha sido la celda 5161.

## 7. Crea una colección con un documento por celda en el que aparezcan los acumulados de los diferentes campos

La colección consiste en crear los acumulados de cada campo (`smsin`, `smsout`, `callin`, `callout`, `Internet`) para cada celda. Cada celda tendrá un documento con sus acumulados de estos. El código para realizar la colección es el siguiente:

```
pipeline_accumulated_by_cell = [  
  {"$group": {"_id": "$cellid",  
    "total_smsin": {"$sum": "$smsin"},  
    "total_smsout": {"$sum": "$smsout"},  
    "total_callin": {"$sum": "$callin"},  
    "total_callout": {"$sum": "$callout"},  
    "total_internet": {"$sum": "$Internet"}}},  
  {"$out": "accumulated_by_cell"}  
]  
  
Milan_CDR_c.aggregate(pipeline_accumulated_by_cell)
```

Figura 11: Código para crear la colección con los acumulados de los diferentes campos

Este código realiza una operación de agregación en MongoDB sobre la colección `Milan_CDR_c`, con el objetivo de calcular la suma acumulada de diversas métricas de comunicación, como SMS entrantes, SMS salientes, llamadas entrantes, llamadas salientes y uso de Internet. El resultado de esta operación se almacena en una nueva colección llamada `accumulated_by_cell`.

El pipeline de agregación consta de dos etapas principales:

### 1. Agrupación por Celda

- Se agrupan los documentos según el campo `cellid`.
- Para cada celda, se calcula la suma de SMS entrantes (`smsin`), SMS salientes (`smsout`), llamadas entrantes (`callin`), llamadas salientes (`callout`), y uso de Internet (`Internet`).
- El resultado es un conjunto de documentos, uno por cada celda, que contiene las sumas acumuladas de las métricas mencionadas.

### 2. Almacenamiento en una Nueva Colección:

- Se almacena en una nueva colección llamada `accumulated_by_cell`.

La colección `accumulated_by_cell` resultante contiene documentos con las sumas acumuladas de SMS entrantes, SMS salientes, llamadas entrantes, llamadas salientes y uso de Internet para cada celda en la colección original `Milan_CDR_c`.

El resultado es el siguiente:

## Milan\_CDR\_db.accumulated\_by\_cell

10.0k 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Filter  Type a query: { field: 'value' } or [Generate query](#) 

Explain Reset Find  Options 

 ADD DATA  EXPORT DATA 

1 - 20 of 9998     

```
_id: 3418
total_smsin: 4272.0773918382165
total_smsout: 1848.936232270327
total_callin: 2976.114542293891
total_callout: 3087.1268823940395
total_internet: 72714.25468892521
```

```
_id: 8921
total_smsin: 3752.515694693393
total_smsout: 2429.9764499989893
total_callin: 2748.363866771274
total_callout: 2928.755019193977
total_internet: 55535.55883263793
```

```
_id: 7711
total_smsin: 3845.896528981709
total_smsout: 1692.9885178295654
total_callin: 2364.3207973580475
total_callout: 2774.7179190713628
total_internet: 29859.053554894464
```

Figura 12: Colección acumulados por celda

## 8. Crea una colección con una documento por celda y hora en el que aparezcan los acumulados de los diferentes campos

La colección consiste en crear los acumulados de cada campo (smsin, smsout, callin, callout, Internet) para cada celda. Cada celda tendrá un documento con sus acumulados de estos. El código para realizar la colección es el siguiente:

```
pipeline_accumulated_by_cell = [  
  {"$group": {"_id": "$cellid",  
    "total_smsin": {"$sum": "$smsin"},  
    "total_smsout": {"$sum": "$smsout"},  
    "total_callin": {"$sum": "$callin"},  
    "total_callout": {"$sum": "$callout"},  
    "total_internet": {"$sum": "$Internet"}}},  
  {"$out": "accumulated_by_cell"}  
]  
  
Milan_CDR_c.aggregate(pipeline_accumulated_by_cell)
```

Figura 13: Código para crear la colección con los acumulados de los diferentes campos

En el contexto de la base de datos MongoDB, la transformación de un campo de tiempo expresado en milisegundos a la componente de hora se logra mediante el uso de las funciones de agregación proporcionadas por MongoDB. En este código, el campo "time", que representa el tiempo de inicio del intervalo en milisegundos, se transforma utilizando la función `$toDate` para convertirlo en un objeto de fecha. Luego, la función `$hour` se aplica a este objeto de fecha para extraer la componente de hora. El resultado se utiliza como parte de la clave de agrupación junto con el identificador de celda (`cellid`).

Los documentos se agrupan por esta combinación de celda y hora, y se realizan operaciones de suma en varias métricas, como SMS entrantes ("`total_smsin`"), SMS salientes ("`total_smsout`"), llamadas entrantes ("`total_callin`"), llamadas salientes ("`total_callout`"), y uso de Internet ("`total_internet`"). El resultado se almacena en una nueva colección denominada `.accumulated_by_cell_and_hour`.

El resultado es el siguiente:



# Milan\_CDR\_db.accumulated\_by\_cell\_and\_hour

240.0k 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Filter ⓘ ⓘ Type a query: { field: 'value' } or [Generate query](#) ⚡

Explain Reset Find ⏪ ⏩ Options ▾

ADD DATA EXPORT DATA

1 - 20 of 239950 🔍 ⚙️ 📄

```
{
  "_id": Object,
  "cellid": 1,
  "hour": 0,
  "total_smsin": 6.006195927066063,
  "total_smsout": 5.504772124127329,
  "total_callin": 0.5147288975331441,
  "total_callout": 1.096106205532247,
  "total_internet": 325.7031158762117
}
```

```
{
  "_id": Object,
  "cellid": 1,
  "hour": 1,
  "total_smsin": 1.9005393950004377,
  "total_smsout": 1.6619469109810645,
  "total_callin": 0.30300177451700665,
  "total_callout": 0.14705959288382752,
  "total_internet": 290.21998645835527
}
```

```
{
  "_id": Object,
  "cellid": 1,
  "hour": 2,
  "total_smsin": 1.5497467429283158,
  "total_smsout": 1.042972769332798
}
```

Figura 14: Colección acumulados por celda y hora

## 9. Realiza un estudio de las celdas indicadas

En este análisis, se seleccionan varias celdas de la colección Milan CDR para realizar un estudio detallado de las interacciones telefónicas y de datos asociadas con cada celda. En algunos casos se excluyen las interacciones internas en Italia y las llamadas internas, identificadas por los códigos de país 39 (Italia) y 0, respectivamente.

Se han utilizado algunas de las consultas hechas anteriormente en esta actividad, adaptándolas a las necesidades de este apartado, en concreto, filtrándolas por la celda indicada en cada caso.

### 9.1. Celdas Seleccionadas

Las celdas seleccionadas para el análisis son las siguientes:

- 4259 (Bocconi)
- 4456 (Navigli)
- 5060 (Duomo)
- 1419 (Terreno Agrícola)
- 2436 (Área Industrial)
- 4990 (Aeropuerto de Linate)
- 945 (Residencial Aislado)
- 5048 (Residencial Céntrico)

Para cada celda seleccionada, se lleva a cabo el siguiente análisis:

- **Países con los que se Interactúa:** Se identifican los distintos códigos de país con los que la celda seleccionada ha interactuado.
- **País con el que más se Interactúa:** Se determina el país con el que la celda ha tenido la mayor cantidad de interacciones, excluyendo Italia y las llamadas internas.
- **Acumulados de Interacciones:** Se calculan los acumulados de diversas métricas, como SMS entrantes y salientes, llamadas entrantes y salientes, y uso de Internet.

A continuación se presenta el código elaborado para obtener los datos del análisis anteriormente explicado:

```

excluded_countries = [39, 0]

selected_cells = [4259, 4456, 5060, 1419, 2436, 4990, 945, 5048]

for cell in selected_cells:
    print("-----")
    print("          Análisis de la celda:", cell)
    print("-----\n")

    # CountryCode con los que interactúa:
    pipeline_interacted_countries = [
        {"$match": {"cellid": cell}},
        {"$group": {"_id": "$countrycode"}},
        {"$sort": SON([("_id", 1)])}
    ]

    distinct_countries = Milan_CDR_c.aggregate(pipeline_interacted_countries)
    distinct_countries = [doc['_id'] for doc in distinct_countries]
    print("Países con los que se interactúa:", distinct_countries)

    # País con el que más se interactúa
    pipeline_top_countries = [
        {"$match": {"$and": [{"countrycode": {"$nin": excluded_countries}},
                               {"cellid": cell}]}},
        {"$group": {"_id": "$countrycode", "count": {"$sum": 1}}},
        {"$sort": {"count": -1}},
        {"$limit": 1}
    ]

    top_interacted_country = Milan_CDR_c.aggregate(pipeline_top_countries)
    top_interacted_country = list(top_interacted_country)[0]['_id']

    print("País con el que más se interactúa (además de Italia y llamadas internas):", top_interacted_country)

    #Acumulados
    pipeline_cell_analysis = [
        {"$match": {"cellid": cell}},
        {"$group": {"_id": "$cellid",
                    "total_smsin": {"$sum": "$smsin"},
                    "total_smsout": {"$sum": "$smsout"},
                    "total_callin": {"$sum": "$callin"},
                    "total_callout": {"$sum": "$callout"},
                    "total_internet": {"$sum": "$Internet"}}}
    ]

    cell_analysis_result = Milan_CDR_c.aggregate(pipeline_cell_analysis)
    cell_analysis_result = list(cell_analysis_result)[0]
    print(f"Acumulados:")
    print(pd.Series(cell_analysis_result)) 18

    print("\n\n ===== \n\n")

```

Los resultados obtenidos de cada celda son los siguientes:

-----  
Análisis de la celda: 4259  
-----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 57, 60, 61, 62, 63, 64, 66, 81, 82, 84, 86, 90, 91, 94, 98, 212, 213, 216, 220, 221, 226, 228, 230, 234, 242, 244, 251, 256, 291, 351, 352, 353, 355, 356, 357, 358, 359, 370, 371, 372, 373, 377, 380, 381, 382, 385, 386, 420, 421, 503, 505, 507, 591, 593, 852, 880, 886, 961, 965, 966, 971, 972, 974, 975, 995, 1204, 1438, 1514, 1604, 1613, 1780, 7777, 18092, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 49

Acumulados:

_id	4259.000000
total_smsin	12724.153334
total_smsout	6468.609191
total_callin	8935.102460
total_callout	10299.850557
total_internet	233453.127646

dtype: float64

=====  
-----  
Análisis de la celda: 4456  
-----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 65, 66, 81, 82, 86, 90, 91, 92, 94, 98, 212, 213, 216, 221, 222, 226, 229, 230, 232, 233, 234, 237, 238, 242, 243, 251, 254, 255, 261, 351, 352, 353, 354, 355, 356, 357, 358, 359, 370, 371, 372, 373, 375, 377, 378, 380, 381, 382, 385, 386, 420, 421, 503, 507, 591, 593, 595, 598, 678, 852, 880, 886, 961, 962, 966, 967, 968, 970, 971, 972, 974, 995, 1204, 1289, 1403, 1416, 1514, 1604, 1778, 1780, 1829, 1907, 1929, 7701, 7702, 7777, 50937, 88239, 97259]

País con el que más se interactúa (además de Italia y llamadas internas): 44

Acumulados:

_id	4456.000000
total_smsin	33537.435050
total_smsout	14909.557291
total_callin	23917.864365
total_callout	23001.288772
total_internet	568506.598233

dtype: float64

=====  
-----  
Análisis de la celda: 5060  
-----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41,

43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 66, 81, 82, 84, 86, 90, 91, 92, 94, 98, 212, 213, 216, 218, 221, 223, 226, 228, 229, 230, 233, 234, 237, 238, 241, 242, 243, 245, 250, 254, 255, 256, 258, 260, 261, 262, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 370, 371, 372, 373, 375, 377, 378, 380, 381, 382, 385, 386, 387, 389, 420, 421, 423, 503, 507, 509, 592, 593, 595, 598, 673, 852, 853, 880, 886, 960, 961, 962, 964, 965, 966, 967, 968, 970, 971, 972, 973, 974, 976, 994, 995, 996, 1204, 1214, 1403, 1416, 1418, 1450, 1514, 1587, 1604, 1613, 1647, 1787, 1829, 1905, 7700, 7701, 7717, 7721, 7777, 7778, 18763, 18764, 18768, 50937, 88239, 97256, 97259]

País con el que más se interactúa (además de Italia y llamadas internas): 44

Acumulados:

_id	5060.000000
total_smsin	69090.062240
total_smsout	33268.123900
total_callin	42971.304990
total_callout	52596.482770
total_internet	612496.942316

dtype: float64

=====

-----  
Análisis de la celda: 1419  
-----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 52, 53, 55, 57, 61, 62, 63, 66, 82, 86, 90, 91, 92, 94, 98, 212, 213, 216, 220, 221, 229, 244, 254, 256, 351, 352, 353, 355, 357, 358, 359, 370, 371, 372, 373, 375, 376, 380, 381, 385, 386, 387, 389, 420, 421, 503, 593, 852, 880, 961, 966, 971, 996, 1416, 1808, 1819, 1829, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 33

Acumulados:

_id	1419.000000
total_smsin	530.395946
total_smsout	229.710593
total_callin	428.327276
total_callout	464.698276
total_internet	5016.234490

dtype: float64

=====

-----  
Análisis de la celda: 2436  
-----

Países con los que se interactúa: [0, 1, 7, 20, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 66, 81, 82, 84, 86, 90, 91, 92, 94, 212, 213, 216, 221, 226, 230, 251, 254, 261, 291, 351, 352, 353, 355, 356, 358, 359, 370, 371, 372, 373, 375, 380, 381, 382, 385, 386, 387, 389, 420, 421, 503, 593, 852, 880, 960, 963, 968, 971, 974, 1416, 1514, 1808, 1829, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 33

```

Acumulados:
_id          2436.000000
total_smsin  2844.990743
total_smsout 1782.894877
total_callin 2308.122751
total_callout 3207.435899
total_internet 38375.181471
dtype: float64

```

=====

-----  
 Análisis de la celda: 4990  
 -----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 65, 66, 81, 82, 84, 86, 90, 91, 92, 93, 94, 98, 211, 212, 213, 216, 218, 221, 223, 225, 226, 228, 229, 230, 231, 233, 234, 236, 237, 241, 242, 243, 249, 250, 254, 255, 256, 257, 258, 261, 262, 264, 351, 352, 353, 354, 355, 356, 357, 358, 359, 370, 371, 372, 373, 374, 375, 377, 380, 381, 382, 385, 386, 387, 389, 420, 421, 502, 503, 504, 507, 591, 593, 598, 852, 880, 886, 961, 962, 964, 965, 966, 967, 968, 971, 972, 973, 974, 994, 995, 1214, 1403, 1416, 1418, 1438, 1450, 1514, 1519, 1587, 1604, 1613, 1647, 1778, 1780, 1905, 1907, 7701, 7705, 7771, 7777, 7778, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 44

```

Acumulados:
_id          4990.000000
total_smsin  15035.045725
total_smsout  6834.801858
total_callin  9611.194039
total_callout 11901.054162
total_internet 123992.107314
dtype: float64

```

=====

-----  
 Análisis de la celda: 945  
 -----

Países con los que se interactúa: [0, 1, 7, 20, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 58, 60, 61, 62, 63, 66, 81, 82, 84, 86, 90, 91, 92, 94, 98, 212, 213, 216, 218, 221, 225, 226, 228, 230, 233, 234, 351, 352, 353, 355, 359, 370, 371, 373, 375, 377, 380, 381, 385, 386, 387, 389, 420, 421, 506, 590, 593, 690, 852, 880, 965, 971, 972, 994, 1416, 1647, 1808, 1829, 1849, 7701, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 33

```

Acumulados:
_id          945.000000
total_smsin  1705.516010
total_smsout 1010.916955
total_callin 1201.980322

```

```
total_callout      1348.251826
total_internet     23311.239413
dtype: float64
```

=====

-----  
Análisis de la celda: 5048  
-----

Países con los que se interactúa: [0, 1, 7, 20, 27, 30, 31, 32, 33, 34, 36, 39, 40, 41, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 57, 58, 60, 61, 62, 63, 65, 66, 81, 82, 86, 90, 91, 92, 93, 94, 98, 212, 213, 216, 218, 221, 225, 230, 234, 237, 241, 243, 248, 250, 251, 253, 254, 255, 261, 351, 352, 353, 355, 357, 358, 359, 370, 371, 372, 373, 377, 380, 381, 385, 386, 387, 389, 420, 421, 503, 591, 593, 598, 852, 880, 886, 961, 962, 965, 966, 968, 971, 972, 974, 1204, 1214, 1246, 1416, 1514, 1604, 1647, 1829, 1905, 1907, 7705, 7707, 18096, 18764, 88239]

País con el que más se interactúa (además de Italia y llamadas internas): 33

Acumulados:

```
_id      5048.000000
total_smsin  30029.934795
total_smsout  22762.239429
total_callin  20290.734409
total_callout  20565.383951
total_internet  386280.132139
dtype: float64
```

## Referencias

- [1] R. L. A. C. C. G. T. F. A. A. V. A. P. . B. L. Gianni Barlacchi, Marco De Nadai, “A multi-source dataset of urban life in the city of milan and the province of trentino.” <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV>, 2015. Accessed: November 2023.