



# 71.88 Gestión de Datos

Entrega 5 - Generación de DW

Grupo 02

Yasmin Ohana - 65098

Isidro Perasso - 65595

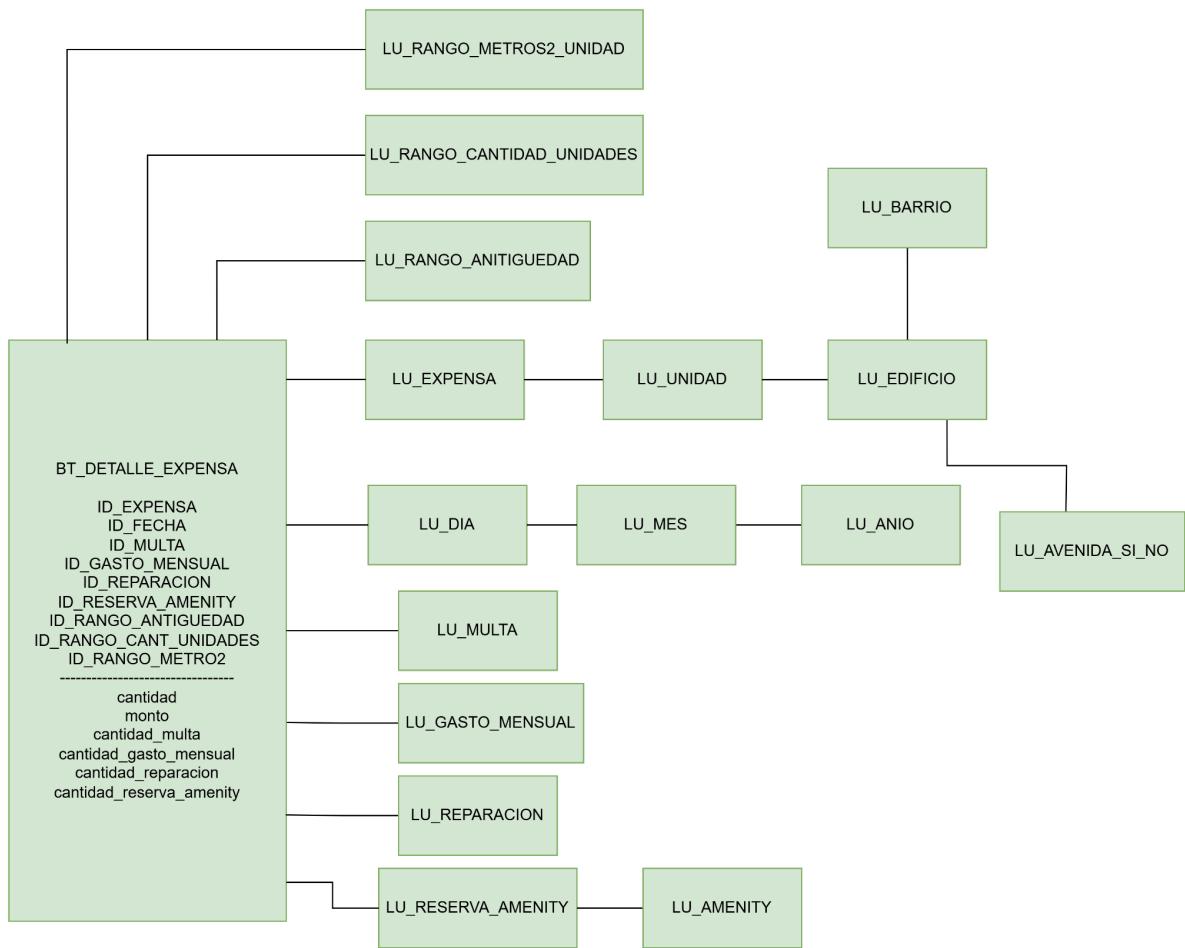
Andrés Glauberman - 65635

Julia Sexe - 65669

Fecha de entrega: Miércoles 18 de junio

---

## Modelo multidimensional:



## Tabla base de Detalle Expensa:

```
CREATE TABLE bt_detalle_expensa (
    ID_Expensa int NOT NULL,
    ID_Fecha date NOT NULL,
    ID_Multa int NULL,
    ID_GastoMensual int NULL,
    ID_Reparacion int NULL,
    ID_ReservaAmenity int NULL,
    ID_RangoAntiguedad int NOT NULL,
```

```

ID_RangoCantidadUnidades int NOT NULL,
ID_RangoMetros2 int NOT NULL,
cantidad tinyint NOT NULL DEFAULT 1,
monto int NOT NULL,
cantidad_multa int not NULL,
cantidad_gasto_mensual int not NULL,
cantidad_reparacion int not NULL,
cantidad_reserva_amenity int not NULL,
);

```

```

INSERT INTO bt_detalle_expensa

SELECT      M.ID_EXPENSA,      D.DIA_ID,      M.ID_MULTA,      NULL,      NULL,      NULL,
RA.ID_RangoAntiguedad,                               RCU.ID_RangoCantidadUnidades,
RM.ID_RANGO_METROS2_UNIDAD, 1,M.MONTO,1,0,0,0

FROM MULTA M

JOIN LU_DIA D

ON M.FECHA=D.DIA_ID

JOIN UNIDAD U

ON M.ID_UNIDAD=U.ID_UNIDAD

JOIN EDIFICIO E

ON U.ID_EDIFICIO=E.ID_EDIFICIO

JOIN LU_RANGO_ANTIGUEDAD RA

ON          RA.limite_inferior<=DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())           AND
DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())<RA.limite_superior

```

```

JOIN LU_RANGO_CANTIDAD_UNIDADES RCU
ON RCU.limite_inferior<=E.CANTIDAD_UNIDADES AND
E.CANTIDAD_UNIDADES<RCU.limite_superior

JOIN LU_RANGO_METROS2_UNIDAD RM
ON RM.LIM_INFERIOR<=U.METROS2 AND U.METROS2<RM.LIM_SUPERIOR

union

SELECT EX.ID_EXPENSA, D.DIA_ID, NULL, G.ID_GASTO, NULL, NULL,
RA.ID_RangoAntiguedad, RCU.ID_RangoCantidadUnidades,
RM.ID_RANGO_METROS2_UNIDAD, 1,EX.MONTO_GASTO,0,1,0,0
FROM GASTO_MENSUAL G

JOIN LU_DIA D
ON G.FECHA_GASTO=D.DIA_ID

JOIN EXPENSA EX
ON G.ID_GASTO=EX.ID_GASTO

JOIN UNIDAD U
ON EX.ID_UNIDAD=U.ID_UNIDAD

JOIN EDIFICIO E
ON U.ID_EDIFICIO=E.ID_EDIFICIO

JOIN LU_RANGO_ANTIGUEDAD RA
ON RA.limite_inferior<=DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE()) AND
DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())<RA.limite_superior

JOIN LU_RANGO_CANTIDAD_UNIDADES RCU
ON RCU.limite_inferior<=E.CANTIDAD_UNIDADES AND
E.CANTIDAD_UNIDADES<RCU.limite_superior

JOIN LU_RANGO_METROS2_UNIDAD RM

```

```

ON RM.LIM_INFERIOR<=U.METROS2 AND U.METROS2<RM.LIM_SUPERIOR

union

SELECT R.ID_EXPENSA, D.DIA_ID, NULL, NULL, R.ID_REPARACION, NULL,
RA.ID_RangoAntiguedad, RCU.ID_RangoCantidadUnidades,
RM.ID_RANGO_METROS2_UNIDAD, 1,R.MONTO,0,0,1,0

FROM REPARACION R

JOIN LU_DIA D

ON R.FECHA=D.DIA_ID

JOIN UNIDAD U

ON R.ID_UNIDAD=U.ID_UNIDAD

JOIN EDIFICIO E

ON U.ID_EDIFICIO=E.ID_EDIFICIO

JOIN LU_RANGO_ANTIGUEDAD RA

ON RA.limite_inferior<=DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())      AND
DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())<RA.limite_superior

JOIN LU_RANGO_CANTIDAD_UNIDADES RCU

ON RCU.limite_inferior<=E.CANTIDAD_UNIDADES      AND
E.CANTIDAD_UNIDADES<RCU.limite_superior

JOIN LU_RANGO_METROS2_UNIDAD RM

ON RM.LIM_INFERIOR<=U.METROS2 AND U.METROS2<RM.LIM_SUPERIOR

UNION

SELECT R.ID_EXPENSA, D.DIA_ID, NULL, NULL, NULL, R.ID_RESERVA,
RA.ID_RangoAntiguedad, RCU.ID_RangoCantidadUnidades,
RM.ID_RANGO_METROS2_UNIDAD, 1,AE.COSTO,0,0,0,1

FROM RESERVA_AMENITY R

```

```
JOIN LU_DIA D
ON CONVERT(DATE, R.FECHA_HORA)=D.DIA_ID

JOIN UNIDAD U
ON R.ID_UNIDAD=U.ID_UNIDAD

JOIN EDIFICIO E
ON U.ID_EDIFICIO=E.ID_EDIFICIO

JOIN AMENITY_X_EDIFICIO AE
ON R.ID_EDIFICIO=AE.ID_EDIFICIO AND R.ID_AMENITY=AE.ID_AMENITY

JOIN LU_RANGO_ANTIGUEDAD RA
ON RA.limite_inferior<=DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE()) AND
DATEDIFF(YEAR,E.FECHA_ALTA,GETDATE())<RA.limite_superior

JOIN LU_RANGO_CANTIDAD_UNIDADES RCU
ON RCU.limite_inferior<=E.CANTIDAD_UNIDADES AND
E.CANTIDAD_UNIDADES<RCU.limite_superior

JOIN LU_RANGO_METROS2_UNIDAD RM
ON RM.LIM_INFERIOR<=U.METROS2 AND U.METROS2<RM.LIM_SUPERIOR

JOIN EXPENSA EX
ON R.ID_EXPENSA=EX.ID_EXPENSA

order by 2
```

SQLQuery2.sql - 1...(GD Grupo02M (64)) \* X

```
SELECT * from bt_detalle_expensa
GO
```

100 %

Results Messages

ID_Expensa	ID_Fecha	ID_Multa	ID_GastoMensual	ID_Reparacion	ID_ReservaAmenity	ID_RangoAntiguedad	ID_RangoCantidadUnidades	ID_RangoMetros2	cantidad	monto	cantidad_multa	cantidad_gasto_mensual	cantidad_reparacion	cantidad_reserva_amenity	
1	158	2025-01-01	NULL	NULL	18	NULL	2	2	1	15918	0	0	1	0	
2	1	2025-01-03	NULL	NULL	35	NULL	3	5	2	10000	0	0	1	0	
3	2	2025-01-03	NULL	NULL	33	NULL	3	5	1	10000	0	0	1	0	
4	2	2025-01-04	NULL	NULL	34	NULL	3	5	1	10000	0	0	1	0	
5	70	2025-01-06	8	NULL	NULL	2	2	3	1	100000	1	0	0	0	
6	79	2025-01-07	9	NULL	NULL	3	2	2	1	30000	1	0	0	0	
7	121	2025-01-07	15	NULL	NULL	2	3	2	1	100000	1	0	0	0	
8	186	2025-01-07	22	NULL	NULL	2	2	3	1	100000	1	0	0	0	
9	216	2025-01-07	25	NULL	NULL	2	2	2	1	120000	1	0	0	0	
10	89	2025-01-10	NULL	NULL	11	NULL	2	2	3	1	18115	0	0	1	0
11	115	2025-01-15	NULL	NULL	14	NULL	2	2	2	1	54597	0	0	1	0
12	281	2025-01-15	30	NULL	NULL	2	5	4	1	30000	1	0	0	0	
13	266	2025-01-16	29	NULL	NULL	3	4	2	1	100000	1	0	0	0	
14	138	2025-01-17	17	NULL	NULL	2	5	3	1	120000	1	0	0	0	
15	46	2025-01-25	NULL	NULL	6	NULL	2	4	3	1	43758	0	0	1	0
16	40	2025-01-26	5	NULL	NULL	2	2	4	1	40000	1	0	0	0	
17	40	2025-01-27	NULL	NULL	5	NULL	2	2	4	1	14803	0	0	1	0

Query executed successfully.

172.16.0.14 (16.0 RTM) | GD Grupo02M (64) | GD Grupo02M | 00:00:00 | 1.295 rows

SQLQuery2.sql - 1...(GD Grupo02M (64)) \* X

```
SELECT COUNT(*) |from bt_detalle_expensa
GO
```

100 %

Results Messages

(No column name)
1

Query executed successfully.

172.16.0.14 (16.0 RTM) | GD Grupo02M (64) | GD Grupo02M | 00:00:00 | 1 rows

**Tabla con todas las fechas que se utilizaron (no es parte del modelo multidimensional):**

Esta tabla se creó ya que no se contaba con todas las fechas de los detalles de la expensa en una misma tabla y para crear el código de las lookups de tiempo se solicitaba una tabla que contenga a estas. Cabe mencionar que también se podría haber utilizado una vista en lugar de la creación de la tabla.

```
create table FECHA  
(fecha date not null)  
  
insert into fecha  
  
select distinct fecha  
  
from REPARACION  
  
union  
  
select distinct FECHA_GASTO  
  
from GASTO_MENSUAL  
  
union  
  
select distinct fecha  
  
from MULTA  
  
union  
  
select distinct FECHA_HORA  
  
from RESERVA_AMENITY
```

#### Tablas de lu\_fecha, mes y año:

```
CREATE TABLE LU_DIA (  
  
DIA_ID DATE NOT NULL,  
  
DESCRIPCION CHAR(10) NOT NULL,  
  
MES_ID INT NOT NULL  
);
```

```

insert into LU_DIA

select distinct

CAST(FECHA AS DATE) as FECHA,

concat(format(year(FECHA), '0000'), '-', (format(month(FECHA), '00')), '-',
format(day(FECHA), '00')) as DESCRIPCION,

concat(format(year(FECHA), '0000'), format(month(FECHA), '00')) as MES_ID

from fecha

ORDER BY FECHA;

```

```

SQLQuery1.sql - 1...(GD_Grupo02M (72))*
SELECT * from LU_DIA
GO
SELECT COUNT(*) from LU_DIA
GO

Results Messages
DIA_ID DESCRIPCION MES_ID
1 2025-01-01 202501
2 2025-01-03 202501
3 2025-01-04 202501
4 2025-01-06 202501
5 2025-01-07 202501
6 2025-01-09 202501
7 2025-01-10 202501
8 2025-01-15 202501
9 2025-01-16 202501
10 2025-01-17 202501
11 2025-01-25 202501
12 2025-01-26 202501
13 2025-01-27 202501
13 2025-01-29 202501
(No column name)
1 61

1 Query executed successfully.

```

CREATE TABLE LU\_MES (

MES\_ID INT NOT NULL,

DESCRIPCION CHAR(8) NOT NULL,

ANIO\_ID INTEGER

);

insert into LU\_MES

select distinct

concat(format(year(FECHA), '0000'), format(month(FECHA), '00')) as MES\_ID,

concat(format(year(FECHA), '0000'), '-', (format(month(FECHA), '00'))) as DESCRIPCION,

year(FECHA) as ANIO\_ID

from fecha

ORDER BY concat(format(year(FECHA), '0000'), '-', (format(month(FECHA), '00')));

The screenshot shows a SQL Server Management Studio window. The query window contains the following T-SQL code:

```
SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_MES
GO
SELECT COUNT(*) from LU_MES
GO
```

The results pane displays the output of the first two queries:

	MES_ID	DESCRIPCION	ANIO_ID
1	202501	2025-01	2025
2	202502	2025-02	2025
3	202503	2025-03	2025
4	202504	2025-04	2025

Below the results, there is a single row labeled '(No column name)' with value '4'.

At the bottom of the results pane, a status bar indicates: 'Query executed successfully.' followed by connection information: '172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 5 rows'.

CREATE TABLE LU\_ANIO (

ANIO\_ID INTEGER,

DESCRIPCION CHAR(4)

);

```

insert into LU_ANIO

select distinct

year(FECHA), 

convert(varchar(4), year(FECHA))

from fecha

order by year(FECHA);

```

```

SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_ANIO
GO
SELECT COUNT(*) from LU_ANIO
GO

100 %
Results Messages
ANIO_ID DESCRIPCION
1 2025 2025

1 (No column name)
1

Query executed successfully.
172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 2 rows

```

## LU\_RANGO\_ANTIGUEDAD

```

create table LU_RANGO_ANTIGUEDAD (
ID_RangoAntiguedad int not null,
limite_inferior int not null,
limite_superior int not null,
detalle varchar(100) not null)

```

go

insert into LU\_RANGO\_ANTIGUEDAD

values

(1, -1, 11, 'menos de 11 años (-inf;11)'),

(2, 11, 13, 'entre 11 y 13 años [11;13)'),

(3, 13, 10000, 'mas de 13 años [13, inf)')

go

The screenshot shows a SQL query window titled 'SQLQuery1.sql' with the following content:

```
SELECT * from LU_RANGO_ANTIGUEDAD
GO
SELECT COUNT(*) from LU_RANGO_ANTIGUEDAD
GO
```

Below the query window is the 'Results' tab of the SSMS interface, displaying the data from the LU\_RANGO\_ANTIGUEDAD table:

ID_RangoAntiguedad	limite_inferior	limite_superior	detalle
1	-1	11	menos de 11 años (-inf;11)
2	11	13	entre 11 y 13 años [11;13)
3	13	10000	mas de 13 años [13, inf)

At the bottom of the results pane, there is a message bar indicating: 'Query executed successfully.' followed by connection information: '172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 4 rows'.

## LU\_RANGO\_CANTIDAD\_UNIDADES

create table LU\_RANGO\_CANTIDAD\_UNIDADES (

    ID\_RangoCantidadUnidades int not null,

    limite\_inferior int not null,

    limite\_superior int not null,

```
detalle varchar(100) not null
```

```
)
```

```
go
```

```
INSERT INTO LU_RANGO_CANTIDAD_UNIDADES
```

```
VALUES (1, -1, 5,'Menos de 5 unidades [-inf,5]),
```

```
(2,5,10,'Entre 5 y 10 unidades [5,10']),
```

```
(3,10,15,'Entre 10 y 15 unidades [10,15']),
```

```
(4,15,20,'Entre 15 y 20 unidades [15,20']),
```

```
(5,20,25,'Entre 20 y 25 unidades [20,25']),
```

```
(6,25,9999999,'Mas de 25 unidades [25,inf)')
```

```
GO
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - 1...(GD Grupo02M (72))". The query is:

```
SELECT * from LU_RANGO_CANTIDAD_UNIDADES
GO
SELECT COUNT(*) from LU_RANGO_CANTIDAD_UNIDADES
GO
```

The results pane displays the data inserted into the table:

ID_RangoCantidadUnidades	limite_inferior	limite_superior	detalle
1	-1	5	Menos de 5 unidades [-inf,5)
2	5	10	Entre 5 y 10 unidades [5,10)
3	10	15	Entre 10 y 15 unidades [10,15)
4	15	20	Entre 15 y 20 unidades [15,20)
5	20	25	Entre 20 y 25 unidades [20,25)
6	25	9999999	Mas de 25 unidades [25,inf)

Below the results, a message indicates: "Query executed successfully." The status bar at the bottom shows: 172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 7 rows.

## LU\_RANGO\_METROS2\_UNIDAD

```

create table LU_RANGO_METROS2_UNIDAD
(
ID_RANGO_METROS2_UNIDAD int not null,
LIM_INFERIOR int not null,
LIM_SUPERIOR int not null,
DETALLE varchar(100) not null)

```

```
INSERT INTO LU_RANGO_METROS2_UNIDAD
```

```
VALUES
```

```

(1,-1,50,'Menos de 50 metros (-inf,50)'),
(2,50,100,'Entre 50 y 100 metros [50,100'),
(3,100,150,'Entre 100 y 150 metros [100,150'),
(4,150,1000000,'Mas de 150 metros [150,inf')

```

```
go
```

The screenshot shows a SQL query window and its results pane.

**SQL Query Window:**

```

SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_RANGO_METROS2_UNIDAD
GO
SELECT COUNT(*) from LU_RANGO_METROS2_UNIDAD
GO

```

**Results Pane:**

ID_RANGO_METROS2_UNIDAD	LIM_INFERIOR	LIM_SUPERIOR	DETALLE
1	-1	50	Menos de 50 metros (-inf,50)
2	50	100	Entre 50 y 100 metros [50,100)
3	100	150	Entre 100 y 150 metros [100,150)
4	150	1000000	Mas de 150 metros [150,inf)

**Status Bar:**

Query executed successfully. 172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD.Grupo02M | 00:00:00 | 5 rows

## LU\_AMENITY

CREATE TABLE LU\_AMENITY

(ID\_AMENITY INT NOT NULL,

TIPO\_AMENITY VARCHAR(50));

INSERT INTO LU\_AMENITY

SELECT \* FROM AMENITY;

The screenshot shows a SQL query window with two queries:

```
SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_AMENITY
GO
SELECT COUNT(*) from LU_AMENITY
GO
```

Below the queries is a results grid showing the data inserted into the table:

ID_AMENITY	TIPO_AMENITY
1	Piletta
2	Gimnasio
3	Parrilla
4	Salon
5	Cancha de tenis
6	Solarium

At the bottom of the results pane, it says "Query executed successfully." and shows connection information: 172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 7 rows.

## LU\_RESERVA\_AMENITY

CREATE TABLE LU\_RESERVA\_AMENITY

(ID\_RESERVA INT NOT NULL,

ID\_AMENITY INT NOT NULL);

INSERT INTO LU\_RESERVA\_AMENITY

```
SELECT ID_RESERVA, ID_AMENITY FROM RESERVA_AMENITY  
WHERE ID_EXPENSA IS NOT NULL;
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - 1... (GD Grupo02M (72))". The query is:

```
SELECT * from LU_RESERVA_AMENITY
GO
SELECT COUNT(*) from LU_RESERVA_AMENITY
GO
```

The results pane displays two tables. The first table, "Results", contains 18 rows of data:

ID_RESERVA	ID_AMENITY
1	3
2	5
3	5
4	5
5	5
6	5
7	5
8	5
9	3
10	5
11	5
12	6
13	2
14	3
15	5
16	5
17	3
18	5
19	5
20	5
21	5
22	5
23	6
24	2

The second table, "Messages", contains one row:

(No column name)
17

At the bottom of the results pane, a message indicates: "Query executed successfully." and shows connection details: "172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 18 rows".

## LU\_MULTA

```
create table LU_MULTA  
(id_multa int not null,  
motivo varchar(100) not null)
```

```
insert into LU_MULTA
```

```
select ID_MULTA, MOTIVO  
from MULTA
```

```

SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_MULTA
GO
SELECT COUNT(*) from LU_MULTA
GO

```

100 %

Results Messages

id_multa	motivo
1	Ruidos molestos
2	Mal uso del ascensor
3	Uso indebidio del SUM
4	Fiestas no autorizadas
5	Mal estacionamiento
6	Basura fuera de horario
7	Mascota sin autorizacion
8	Ruidos molestos
9	Mal uso del ascensor
10	Uso indebidio del SUM
11	Fiestas no autorizadas
12	Mal estacionamiento
13	Basura fuera de horario

(No column name)

1 44

Query executed successfully.

172.16.0.14 (16.0 RTM) | GD Grupo02M (72) | GD Grupo02M | 00:00:00 | 45 rows

## LU\_GASTO\_MENSUAL

```

create table LU_GASTO_MENSUAL

(id_gasto int not null)

```

```

insert into LU_GASTO_MENSUAL

select id_gasto

from GASTO_MENSUAL

where ID_GASTO in (select ID_GASTO from EXPENSA)

```

The screenshot shows a SQL Server Management Studio window with the following content:

```
SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_GASTO_MENSUAL
GO
SELECT COUNT(*) from LU_GASTO_MENSUAL
GO
```

The Results tab displays the following data:

id_gasto
1
2
3
4
5
6
7
8
9
10
11
12
13
(No column name)
1

The Messages tab shows the message "Query executed successfully."

## LU\_REPARACION

```
create table LU_REPARACION
```

```
(id_reparacion int not null,  
tipo_reparacion varchar(50) null,  
calificacion tinyint null)
```

```
insert into LU_REPARACION
```

```
select ID_REPARACION, TIPO_REPARACION, CALIFICACION  
from REPARACION
```

SQLQuery1.sql - 1...(GD Grupo02M (86))\*

```

SELECT * from LU_REPARACION
GO
SELECT COUNT(*) from LU_REPARACION
GO

```

100 %

Results Messages

	id_reparacion	tipo_reparacion	calificacion
1	1	Perdida de agua	NULL
2	2	Perdida de agua	4
3	3	Perdida de agua	4
4	4	Perdida de agua	7
5	5	Perdida de agua	7
6	6	Perdida de agua	8
7	7	Perdida de agua	NULL
8	8	Bombilla de luz quemada	NULL
9	9	Bombilla de luz quemada	NULL
10	10	Bombilla de luz quemada	NULL
11	11	Bombilla de luz quemada	2
12	12	Bombilla de luz quemada	NULL
13	13	Bombilla de luz quemada	NULL

(No column name)

1 35

Query executed successfully.

172.16.0.14 (16.0 RTM) | GD Grupo02M (86) | GD Grupo02M | 00:00:00 | 36 rows

## LU\_EXPENSA

```

CREATE TABLE LU_EXPENSA (
    ID_EXPENSA INTEGER NOT NULL,
    ID_UNIDAD INTEGER NOT NULL)

```

```
INSERT INTO LU_EXPENSA
```

```
SELECT ID_EXPENSA, ID_UNIDAD FROM EXPENSA
```

The screenshot shows a SQL Server Management Studio window with the following content:

```
SQLQuery1.sql - 1...(GD Grupo02M (72))*
SELECT * from LU_EXPENSA
GO
SELECT COUNT(*) from LU_EXPENSA
GO
```

Results grid:

ID_EXPENSA	ID_UNIDAD
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13

Message grid:

(No column name)
1200

Message bar: Query executed successfully.

## LU\_UNIDAD

```
CREATE TABLE LU_UNIDAD (
ID_UNIDAD INTEGER NOT NULL,
ID_EDIFICIO SMALLINT NOT NULL,
UNIDAD_FUNCIONAL VARCHAR(20) NOT NULL)
```

```
INSERT INTO LU_UNIDAD
```

```
SELECT ID_UNIDAD, ID_EDIFICIO, UNIDAD_FUNCIONAL FROM UNIDAD
```

```

SQLQuery1.sql - 1...(GD_Grupo02M (72))*
SELECT * from LU_UNIDAD
GO
SELECT COUNT(*) from LU_UNIDAD
GO

```

100 %

Results Messages

ID_UNIDAD	ID_EDIFICIO	UNIDAD_FUNCIONAL
1	1	T1-1A
2	1	T1-1B
3	1	T1-2A
4	1	T1-2B
5	1	T1-3A
6	1	T1-3B
7	1	T1-4A
8	1	T1-4B
9	1	T1-5A
10	1	T1-5B
11	1	T2-1A
12	1	T2-1B
13	1	T2-2A

(No column name)

1 300

Query executed successfully.

172.16.0.14 (16.0 RTM) | GD\_Grupo02M (72) | GD\_Grupo02M | 00:00:00 | 301 rows

## LU\_EDIFICIO

```

CREATE TABLE LU_EDIFICIO (
    ID_EDIFICIO SMALLINT NOT NULL,
    NOMBRE VARCHAR (50) NOT NULL,
    ID_BARRIO INTEGER NOT NULL,
    ID_AVENIDA INTEGER NOT NULL)

```

```
INSERT INTO LU_EDIFICIO
```

```

SELECT E.ID_EDIFICIO, E.NOMBRE, B.ID_BARRIO, L.ID_AVENIDA FROM EDIFICIO E
INNER JOIN LU_BARRIO B
ON      RIGHT(E.DIRECCION,LEN(E.DIRECCION)-CHARINDEX(',',E.DIRECCION)-1) =
B.Nom_Barrio
INNER JOIN LU_AVENIDA_SI_O_NO L

```

ON CASE

WHEN LEFT(E.DIRECCION,3) LIKE 'Av.' THEN 1

ELSE 0 END = L.Es\_Avenida

GO

The screenshot shows a SQL query window titled "SQLQuery1.sql - 1...(GD\_Grupo02M (72))". The query is:

```
SELECT * from LU_EDIFICIO
GO
SELECT COUNT(*) from LU_EDIFICIO
GO
```

The results pane displays the data from the LU\_EDIFICIO table:

ID_EDIFICIO	NOMBRE	ID_BARRIO	ID_AVENIDA
1	Palacio Raggio	3	2
2	Torre Renoir	9	2
3	Le Parc Figueira Alcorta	8	2
4	Muleria Towers	9	2
5	Torre Al Rio	13	1
6	Quartier Dorrego	5	2
7	Forum Puerto Madero	9	2
8	Alvear Tower	9	2
9	Dome Residence	1	1
10	Torre Chateau	7	1
11	Torre Aquatina	12	2
12	Torre Grand Bourg	8	2
13	Torre La Pasc	2	1

Below the table, there is a single row labeled "(No column name)" with value 30.

## LU\_BARRIO

CREATE TABLE LU\_BARRIO (

    ID\_BARRIO int IDENTITY(1,1) ,

    Nom\_Barrio varchar(30));

INSERT INTO LU\_BARRIO

SELECT DISTINCT

RIGHT(DIRECCION,LEN(DIRECCION)-CHARINDEX(',',DIRECCION)-1) as Nombre\_Barrio  
FROM EDIFICIO;

The screenshot shows a SQL query window titled "SQLQuery1.sql - 1... (GD Grupo02M (72))". The query is:

```

SELECT * from LU_BARRIO
GO
SELECT COUNT(*) from LU_BARRIO
GO

```

The results grid displays 14 rows of data:

ID_BARRIO	Nom_Barrido
1	Balvanera
2	Belgrano
3	Boedo
4	Caballito
5	Colegiales
6	Microcentro
7	Nunes
8	Palermo
9	Puerto Madero
10	Recoleta
11	Retro
12	Rosario
13	Vicente Lopez

(No column name)

1 13

Query executed successfully.

## LU\_AVENIDA\_SI\_O\_NO

CREATE TABLE LU\_AVENIDA\_SI\_O\_NO (

    ID\_Avenida INT,

    Es\_Avenida BIT,

    Descripcion VARCHAR(50));

INSERT INTO LU\_AVENIDA\_SI\_O\_NO

(ID\_Avenida, Es\_Avenida, Descripcion)

VALUES (1,1, 'La direccion contiene el prefijo Av.'),(2,0, 'La direccion no contiene el prefijo Av.');

```

SQLQuery2.sql - 1...(GD Grupo02M (64))*
select * from LU_AVENIDA_SI_O_NO
GO
select COUNT(*) from LU_AVENIDA_SI_O_NO
GO

```

ID_Avenida	Es_Avenida	Descripcion
1	1	El edificio se encuentra sobre una avenida
2	0	El edificio no se encuentra sobre una avenida

(No column name)

1 2.

Query executed successfully.

## Tablero en Power BI:

Para comenzar, el trabajo realizado en Power BI cuenta con cuatro páginas que siguen un camino lógico a considerar. El primer panel es una página que la plataforma va a proveer a cada unidad en particular con el objetivo de que los usuarios tengan una experiencia enriquecedora donde puedan conocer mejor cómo interactúan con el edificio. El segundo panel tiene un objetivo similar pero relacionado a los edificios, para que los administradores puedan controlar y tomar mejores decisiones en base a los datos recopilados por la plataforma. Estos paneles son información que la empresa dueña de la aplicación concede con el objetivo de mejorar cómo se relacionan estos con la aplicación y aumente su satisfacción en relación al servicio ofrecido por la app.

En cuanto a los dos paneles siguientes, de multas y reparaciones, posiciona a la empresa con un amplio abanico de posibilidades. Tiene la chance de ampliar sus operaciones hacia nuevos negocios y oportunidades de generar ingresos en base a la información que generan los edificios. Además, le permite seguir ofreciendo recomendaciones de negocio a sus clientes, los edificios. Así, la plataforma proporciona valor a aquellos que confían en desplegar su información en ella y profundiza la fidelización con ellos.

Previo a la explicación de cada panel, es necesario mencionar que al aplicar ciertos filtros de manera extrema, como filtrar un rango de fechas y que el periodo sea de dos días, es probable que al no haber datos para todas las fechas, las visualizaciones en tal caso

aparezcan vacías. Esto se debe a una cuestión lógica de que al aplicar ciertos filtros no haya datos para esas condiciones. Aunque sea un problema inherente al hecho de visualizar datos, creemos necesario hacer tal mención.

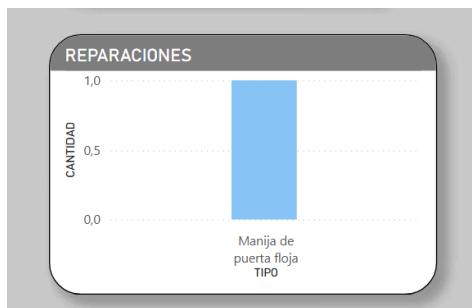
### Panel 1: reporte sobre una unidad particular

En el primer panel se realiza un reporte sobre una unidad particular especificada por el usuario. El objetivo es poder conocer en detalle ciertos aspectos de una unidad específica, como las reparaciones, reserva de amenities, desglose de expensas, entre otros. En este panel existen tres filtros: primero, un filtro desplegable de selección única para elegir el nombre del edificio; segundo, un filtro de selección única en forma de mosaico para seleccionar la unidad funcional; tercero, un filtro deslizante para seleccionar el rango de fechas. A su vez, al posicionarse y cliquear en un lugar en particular en cada gráfico, también se puede filtrar. Por último, consideramos que este panel será visible tanto para la administración del edificio como para los dueños de la unidad.

Además, se programó para que en caso de que una unidad no tenga multas o el monto de algún rubro en el desglose de expensas sea \$0 (cero unidades monetarias), el panel refleje eso de manera elegante y no con un mensaje predeterminado por la herramienta. Por ejemplo, que si no hay multas muestre “no hay multas” y el monto de multas sea \$0, en vez de que aparezca un cartel del estilo de “(En blanco)”, que a fines estéticos daña el dashboard.

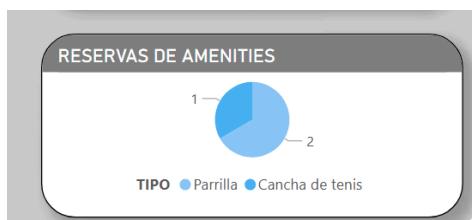


Visualización 1: Gráfico de barras



En este gráfico de barras se realiza una comparación de la cantidad de reparaciones por tipo, para una unidad funcional específica de un edificio particular filtrado previamente. En el eje X se observan los tipos de reparaciones que están disponibles actualmente en el sistema y esa unidad funcional requirió, y en el eje Y las cantidades correspondientes a cada tipo de reparación. De esta manera, el objetivo de la visualización es poder comparar las cantidades de reparaciones de distintos tipos para una misma unidad funcional. Así, se puede tener un conocimiento concreto acerca de cuáles son los tipos de reparaciones más y menos frecuentes para una unidad particular. Esto permite saber qué sectores físicos de una unidad funcional necesitan más mantenimiento que otros para evitar futuras reparaciones, sin que el residente deba llevar registro de eso por su propia cuenta. Una decisión puntual que podría ser tomada por el residente de una unidad funcional tras analizar esta visualización es que al observar que, por ejemplo, el tipo de reparación “inodoro no descarga correctamente” tuvo la mayor cantidad de reparaciones en su unidad, se decide realizar un mantenimiento con mayor frecuencia en el inodoro para disminuir el número de reparaciones de ese tipo a futuro.

Visualización 2: Gráfico de torta



En este gráfico de torta se realiza un desglose de la composición de las reservas de amenities, para una unidad funcional específica de un edificio particular previamente filtrado. En la leyenda se pueden observar los tipos de amenity disponibles para el edificio de esa unidad, y dentro de la torta se ve la composición de la totalidad de reservas por tipo de amenity. De esta manera, el objetivo de la visualización es poder conocer las cantidades de reservas por tipo de amenity y la composición de la totalidad de las reservas para una misma unidad funcional. Así, se puede tener un conocimiento concreto acerca de cuáles

son los tipos de amenities más y menos reservados para una unidad particular. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, el tipo de amenity “parrilla” tuvo la mayor cantidad de reservas en una unidad, el administrador del edificio puede ir a pedirle feedback sobre comentarios y opiniones a los dueños de esa unidad para mejorar el servicio ofrecido de ese tipo de amenity a futuro. En cuanto al usuario, este gráfico le indica al mismo cuál es el valor que el edificio le ofrece a él como residente a través de las amenities que le provee.

#### Medida 1: Composición del monto total de expensa



En esta medida compuesta por tarjetas agrupadas se realiza un desglose de la composición del monto de la expensa de una unidad funcional específica de un edificio particular que fue filtrado previamente. Se puede observar que la composición del monto total se divide en: monto de reparaciones, monto de reservas, monto de multas y monto de gasto mensual. De esta manera, el objetivo de la medida es poder conocer la composición del monto total de una expensa para una unidad particular. Así, se puede tener un conocimiento concreto acerca de cuáles son los aspectos que involucran mayores y menores gastos para una unidad particular. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, el monto de multas es muy grande en una unidad, los dueños de esa unidad deciden controlar la cantidad de acciones indebidas que realizan ya que se pueden observar que el monto es muy elevado y está siendo significativo con respecto al monto total de la expensa. Además, los residentes podrían chequear cuánto de lo que pagan en las expensas se va hacia gastos fuera del gasto mensual, que puede considerarse fijo. De esa manera pueden alertarse de aspectos que estén demandando mucho dinero en las expensas para reducirlo para los meses entrantes.

#### Medida 2: Cantidad total de multas



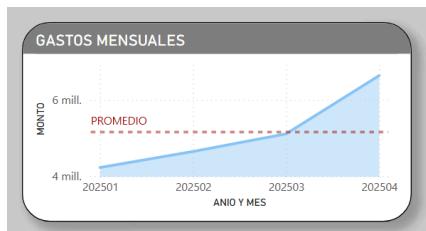
En esta medida se observa la cantidad total de multas de una unidad funcional específica de un edificio particular que fue filtrado previamente. De esta manera, el objetivo de la medida es poder conocer cuántas multas realizó una unidad para que tanto la administración como los dueños de la unidad puedan tener conocimiento de este número de acciones indebidas. Así, tanto la administración como los dueños pueden tomar diferentes medidas en el asunto. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, la cantidad de multas es muy grande en una unidad, los dueños de esa unidad deciden controlar la cantidad de infracciones ya que verán este número reflejado en el monto de las multas y la administración podrá notificar a la unidad acerca del exceso de multas que tuvieron para alertarlos acerca de esta situación y evitar que este número siga aumentando a futuro. También, considerando que tras cierta cantidad de multas el valor de la misma viene con recargo, puede considerarse importante que el residente de la unidad esté al tanto de cuántas infracciones realizó y que le sirva como señal de alerta.

#### Panel 2: reporte sobre un **edificio particular**

En el segundo panel se realiza un reporte sobre un edificio particular especificado por el usuario. El objetivo es poder conocer en detalle ciertos aspectos de un edificio específico, como la cantidad de unidades funcionales, reserva de amenities, gastos mensuales, entre otros. En este panel existen dos filtros: primero, un filtro desplegable de selección única para elegir el nombre del edificio y segundo, un filtro con deslizador para seleccionar el rango de fechas. A su vez, al posicionarse en un lugar en particular en cada gráfico, también se puede filtrar. Por último, consideramos que este panel será visible únicamente para el rédito de la administración del edificio.

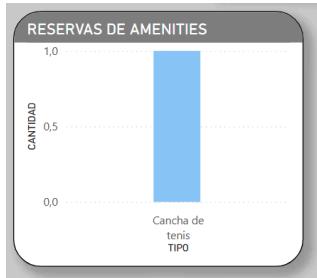


Visualización 1: Gráfico de líneas con línea constante en eje Y



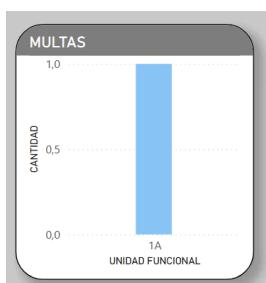
La visualización muestra el gasto mensual de un edificio particular previamente filtrado y su evolución mes a mes. El gasto mensual de cada edificio está compuesto por la suma de los monto de gasto mensual de cada residente de ese edificio. A través de la misma el edificio puede analizar si sus gastos van en aumento o no. La línea constante es el promedio de gasto mensual de ese edificio. Así, es posible establecer una comparación de acuerdo a los meses anteriores y posteriores pero teniendo en cuenta el promedio general como referencia. A partir de este análisis, el edificio puede hallar la necesidad de revisar si está registrando meses con gastos elevados, en pos de profundizar por qué está ocurriendo eso. ¿Se debe a un incremento previsto por construcciones en el edificio? ¿Se debe a gastos inesperados de agua y luz? Eso dependerá del edificio. Pero la visualización sirve como una señal de alarma ante gastos del edificio fuera de lo común. Una decisión particular podría ser la de direccionar una alerta a los residentes para que regulen sus gastos ya que el edificio se encontró con un patrón incremental en los últimos meses y no se debe a motivos que ellos tengan previstos.

Visualización 2: Gráfico de barras



La presente visualización refleja la cantidad de reservas por tipo de amenity para un edificio. El objetivo de la misma es que el edificio tenga información de qué amenities están siendo más usados por sus residentes. Como las únicas amenities de las que se registra el uso son aquellas reservables, el análisis cobra más relevancia ya que las reservables suelen ser las amenities que tienen un costo de reserva hacia los residentes y que más inversión requiere de parte de los edificios. Un ejemplo práctico es que si la cancha de tenis es el amenity más usado del edificio, esa amenity será a la que más atención habrá que ponerle. Posiblemente sea la que necesite mayor mantenimiento de parte del edificio por su uso (como asegurarse que las condiciones de la misma sea óptima o renovar las pelotas de tenis en caso de que el edificio las provea). A la vez, filtrándose por lapso temporal puede observarse si se trata de una tendencia de tan solo unos meses o los residentes valoran mucho esa amenity. De ser así, podrían tomarse acciones como hacer propuestas sobre la ampliación de las canchas si la demanda es muy alta o expandirse y evaluar la construcción de canchas de pádel como nuevo amenity. El análisis de cada amenity abre las puertas a nuevas evaluaciones y oportunidades.

Visualización 3: Gráfico de barras



La visualización muestra la cantidad de multas por cada una de las unidades funcionales de un edificio particular que fue filtrado previamente. En el eje X, se pueden observar las unidades funcionales de ese edificio que tuvieron multas y en el eje Y las cantidades correspondientes de multas por cada unidad funcional. De esta manera, el objetivo de la visualización es poder comparar las cantidades de multas en distintas unidades funcionales para un edificio. Así, se puede tener un conocimiento concreto acerca de cuáles son las

unidades funcionales con más y menos multas. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, la unidad “T1-1” tuvo la mayor cantidad de multas, se decide enviar una advertencia para evitar que la cantidad de acciones indebidas en esa unidad siga siendo significativa a futuro.

Tabla 1:

REPARACIONES		
TIPO	FECHA	CALIFICACION
Perdida de agua	martes, 18 de febrero de 2025	7

La tabla muestra las calificaciones, fecha y tipo de reparación por cada registro de reparación para un edificio particular que fue filtrado previamente. De esta manera, el objetivo de la tabla es tener un registro detallado sobre las calificaciones por cada reparación, según su tipo. Una decisión puntual que podría ser tomada tras analizar esta tabla es que, al observar que, por ejemplo, el tipo de reparación “Bombilla de luz quemada” tuvo calificaciones bajas, se decide realizar una capacitación adicional a los empleados del edificio que están a cargo de estas reparaciones para mejorar el servicio ofrecido para este tipo de reparaciones.

Medida 1: Promedio de gasto mensual

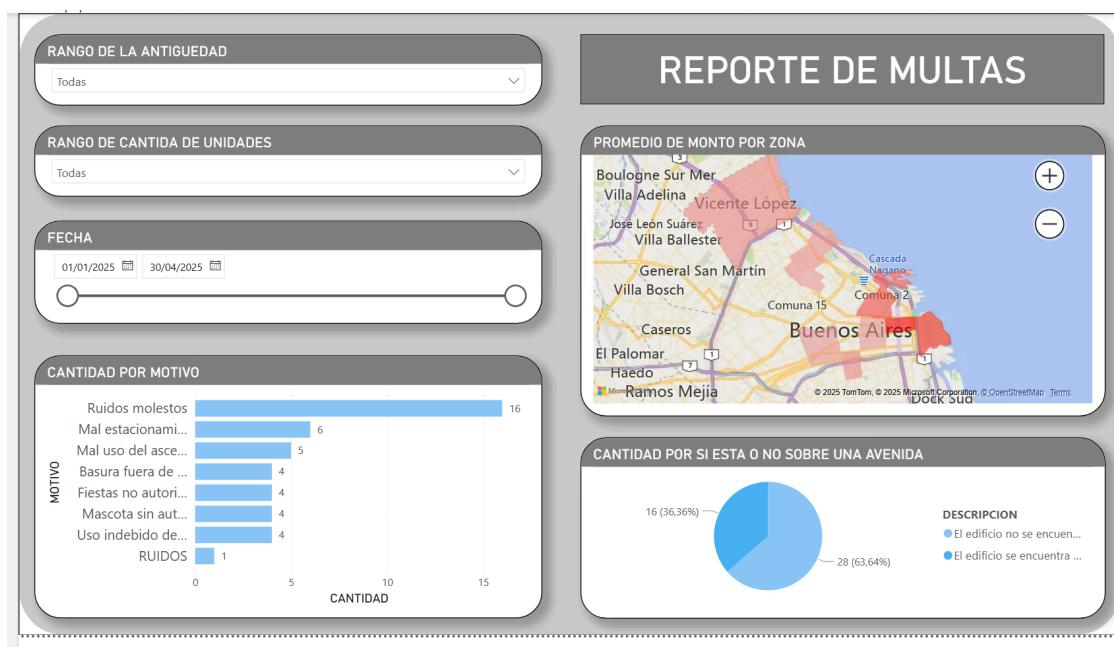


La medida de promedio de gasto mensual por edificio muestra justamente esa función de dominio agregado. Esta medida se utiliza en la visualización 1 de esta sección. El objetivo de calcularla está explicitado en la visualización anterior. Pero el motivo de mostrar este valor además de exhibirlo como línea constante en el otro gráfico es para tener la referencia de cuál es el valor de esa constante y que no quede únicamente en la línea. De esta manera se vuelve más tangible la comparación entre las barras de la visualización y el promedio de gasto mensual.

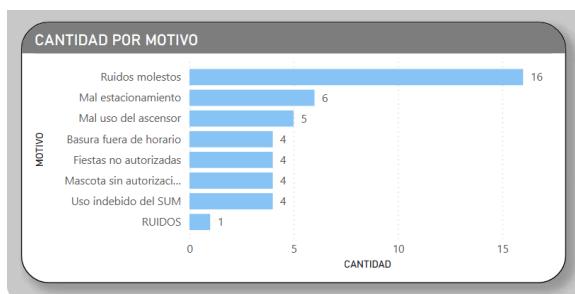
Vale mencionar también que fue programado para que en caso de que el promedio de gasto mensual del edificio sea de cero unidades monetarias, va a aparecer “\$0” y no un mensaje poco amigable y predeterminado por el sistema.

### Panel 3: reporte de multas sobre todos los edificios

En el tercer panel se realiza un reporte de multas sobre todos los edificios. El objetivo es poder conocer en detalle ciertos aspectos sobre las multas, como la cantidad, el motivo, las zonas, entre otros. En este panel existen tres filtros: primero, un filtro desplegable de selección única o múltiple para elegir el rango de antigüedad del edificio; segundo, otro filtro desplegable de selección única o múltiple para elegir el rango de cantidad de unidades del edificio; tercero, un filtro con deslizador para seleccionar el rango de fechas. A su vez, al posicionarse en un lugar en particular en cada gráfico, también se puede filtrar, permitiendo filtrar por zona del edificio, por motivo de la multa o por si el edificio está o no sobre una avenida. Por último, consideramos que este panel será visible para la empresa dueña de la aplicación.



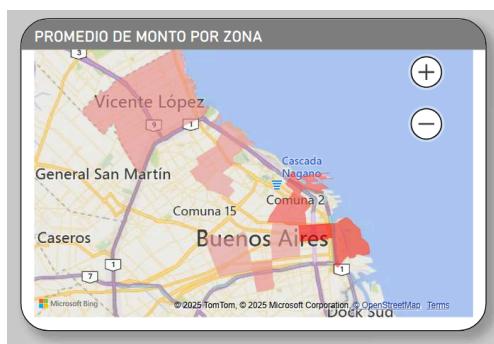
Visualización 1: Gráfico de barras horizontales



En este gráfico de barras horizontales se realiza una comparación de la cantidad de multas por motivo de todos los edificios. En el eje Y, se pueden observar los motivos de multa que

están disponibles actualmente en el sistema y en el eje X las cantidad correspondientes a cada motivo de multa. De esta manera, el objetivo de la visualización es poder comparar las cantidades de multas de distintos motivos para todos los edificios. Así, se puede tener un conocimiento concreto acerca de cuáles son los motivos de multa más y menos frecuentes para todos los edificios del sistema. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, el motivo de multa “ruidos molestos” tuvo la mayor cantidad, se decide realizar una campaña general de concientización a todos los vecinos de los edificios para disminuir la cantidad de multas por este motivo a futuro.

#### Visualización 2: Mapa



En este mapa se realiza un análisis geográfico sobre el promedio de expensas por zona. Se utiliza una gama de colores donde el color más claro corresponde al promedio de expensas más bajo, mientras que el más oscuro corresponde a promedio más alto. De esta manera, el objetivo de la visualización es poder comparar el monto promedio de las expensas por zona para todos los edificios. Así, se puede tener un conocimiento concreto acerca de cuáles son las zonas con expensas promedio más altas y más bajas para todos los edificios del sistema. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, la zona de mayor promedio de expensas es Puerto Madero, se decide establecer una estrategia de precios diferenciada donde a este área de altos costos se le ajusta la tarifa de suscripción ofreciendo más funcionalidades al sistema pero a un monto mayor.

#### Visualización 3: Gráfico de torta



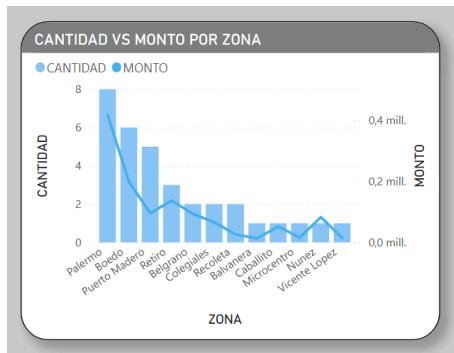
En este gráfico de torta se realiza un desglose de la composición de la cantidad de multas según si el edificio se encuentra en una avenida o no. En la leyenda se puede observar la descripción sobre la ubicación o no en una avenida del edificio y dentro de la torta, se ve la composición de la totalidad de las multas según la división previamente mencionada. De esta manera, el objetivo de la visualización es poder conocer la cantidad de multas según la ubicación o no en avenida de los edificios. Una decisión puntual que podría ser tomada tras analizar esta visualización es que al observar que, por ejemplo, la mayor cantidad de multas se generaron en edificios ubicaciones sobre avenidas, se decide implementar campañas específicas para disminuir la cantidad de multas en edificios sobre avenidas para luego comparar periódicamente la proporción de multas en avenidas para validar si las acciones reducen realmente la incidencia.

#### Panel 4: reporte de **reparaciones** sobre **todos los edificios**

En el cuarto panel se realiza un reporte de reparaciones sobre todos los edificios. El objetivo es poder conocer en detalle ciertos aspectos sobre las reparaciones, como la cantidad, el tipo, las zonas, entre otros. En este panel existen dos filtros: primero, un filtro desplegable de selección única o múltiple para elegir el rango de metros cuadrados de la unidad y segundo, un filtro deslizante para seleccionar el rango de fechas. A su vez, al posicionarse en un lugar en particular en cada gráfico, también se puede filtrar. Por último, consideramos que este panel será visible tanto para la empresa dueña de la aplicación.



Visualización 1: Gráfico de barras con línea superpuesta



La visualización muestra la suma total de dinero destinado a reparaciones por barrio de los edificios. En tanto, también muestra la cantidad de reparaciones de esos barrios. El objetivo de la visualización es clarificar qué barrios son los que experimentan más desperfectos en sus hogares. A partir de esta información se puede comparar qué barrios tienen las viviendas más nuevas y cuáles necesitan más mantenimiento. Es importante saber tanto la cantidad de reparaciones como el monto total de estas ya que una reparación puede ser de gran escala (cambiar los caños del departamento, por ejemplo) y si no tuviéramos eso en cuenta estaría generando falsas conclusiones. De esta manera contextualizamos la visualización y prevenimos interpretaciones incorrectas. Una decisión particular puede ser con el objetivo de relevar información a otras plataformas con el objetivo de que futuros potenciales compradores se orienten hacia qué ubicaciones tienen edificios en mejores condiciones. Si la empresa quisiera adherirse al lanzamiento o construcción de un edificio, también podría usar esa información para elegir barrios con edificios más nuevos. De la

misma manera, la empresa podría ofrecer o asociarse a personal de mantenimiento de ciertos barrios en particular, sumando a la plataforma personal propio de la aplicación. Bajo la información de esta visualización se garantiza la certeza de asociarse con personal de aquellos barrios que más reparaciones necesiten, teniendo la certeza de que los servicios que ellos van a ofrecer serán bien recibidos y se aseguran más ingresos por comisión de la contratación de ese personal.

Visualización 2: Gráfico de barras



Esta visualización detalla la cantidad de reparaciones de acuerdo a la antigüedad del edificio al momento de la misma, separada en rangos. El objetivo de la misma es entender si los edificios más antiguos requieren más reparaciones como señal de mayor mantenimiento. Un caso puntual en que el gráfico puede ser útil es para que la empresa oriente al personal de mantenimiento a edificios de qué antigüedad asociarse. El personal puede acercarse a la empresa dueña de la aplicación y pedirle información de qué tipo de edificios suelen requerir de más reparaciones. De esta manera, aunque la plataforma no cobrara por estas recomendaciones, la plataforma mejora la disponibilidad de personal para los edificios que administra y para los residentes que viven en ellos, aumentando la conformidad de los mismos con la aplicación. Así, a través de este tipo de recomendaciones gratuitas, la empresa busca aumentar la satisfacción sobre la plataforma y mantener la fidelidad de esos edificios.

Visualización 3: Gráfico de anillos



Esta visualización detalla el promedio del monto de reparaciones por tipo de reparación para todos los edificios del sistema. El objetivo de la misma es conocer los tipos de reparaciones que manejan un mayor volumen de dinero y, por lo tanto, demandan más gastos para los dueños de las unidades funcionales. Una decisión puntual que podría ser tomada es aconsejar a los edificios para que tomen mayores recaudos en el mantenimiento de un tipo específico de reparación que haya tenido un elevado volumen de dinero invertido. De esta manera, la visión integral de la aplicación les permite dar un feedback a los edificios que contribuya a un mayor cuidado de los mismos.