RECOLECCIÓN DE BASURA

ENTREGA DEL PROYECTO FINAL - SQL

CANDELA WETTSTEIN - Comisión 34985

CODERHOUSE - Marzo 2023

PROFESOR: Juan Pérez

TUTORA: Nancy Elizabeth Villena Reines

Índice

Introducción

La ciudad “Green City” cuenta con 50 barrios, y en cada esquina de las calles que los conforman, hay colocados contenedores de color verde para los residuos reciclables, y de color anaranjado para los no reciclables.

Los contenedores son recogidos por camiones recolectores, que se encargan de llevar los residuos a destino para proceder con el reciclaje, reutilización y/o reducción de los mismos.

Objetivo

El sistema de recolección se implementó recientemente, por lo que se debe crear una base de datos que permita gestionar la higiene urbana, georreferenciando los contenedores de residuos, y brindando información respecto a los servicios realizados.

Situación problemática

El municipio de “Green City” no cuenta con información pertinente a la generación de residuos que desechan sus habitantes, por lo tanto, la toma de decisiones respecto a políticas medioambientales se dificulta.

Modelo de negocio

“Green City” vela por la sustentabilidad y sostenibilidad de sus espacios, siendo una ciudad limpia y segura para sus habitantes. Este modelo tiene como pilar reducir, reciclar y reutilizar: la regla de las tres “R”, a favor del planeta.

Imagen 1. Regla de las "tres R"

Diagrama Entidad-Relación

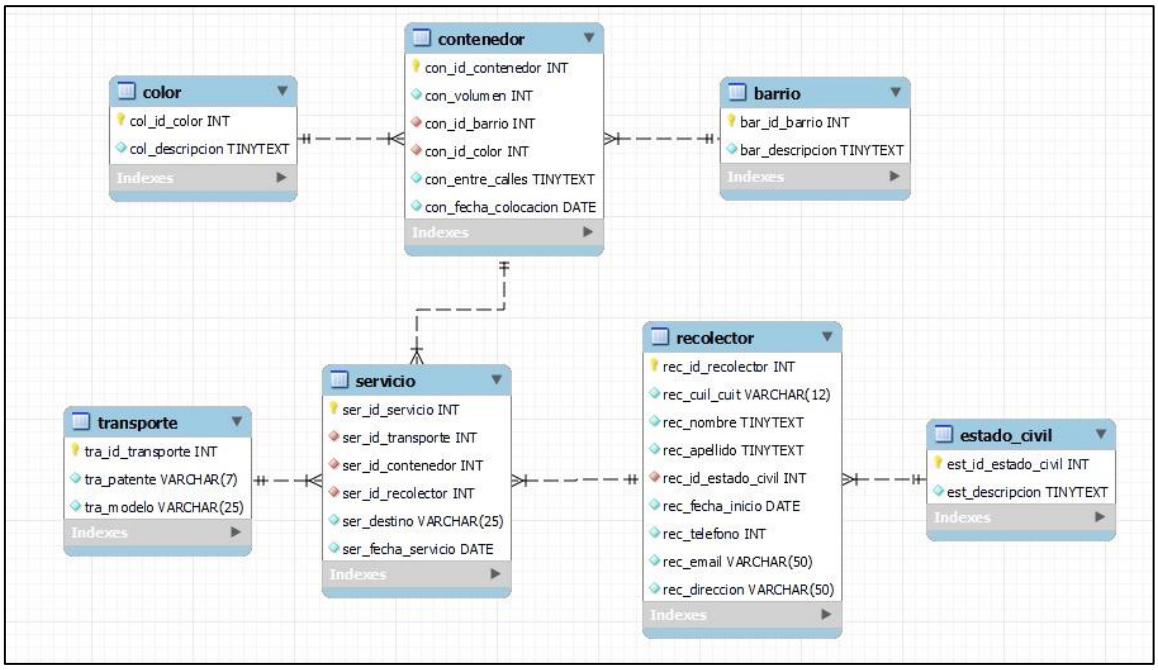


Imagen 2. Diagrama E-R

Creación de objetos

TABLAS

La base de datos se compone de las siguientes tablas:

1. Barrio
2. Color
3. Contenedor
4. Estado_civil
5. Recolector
6. Servicio
7. Transporte

Además, de las tablas bitácora:

1. Contenedor_bitacora
2. Servicio_bitacora

A continuación, se realiza una descripción de cada una de ellas. Cabe aclarar que dentro de la columna observaciones se determina si el campo es clave primaria (PK), clave foránea (FK), autoincremental (AI) y/o no nulo (NN).

TABLA **CONTENEDORES**		
Contiene información de los 100 contenedores que hay en la ciudad, como su ubicación, fecha de colocación, volumen, y si contiene residuos recuperables o no según sean de color verde o anaranjado.		
Nombre del Campo	Tipo de dato	Observaciones
CON_ID_CONTENEDOR	INT	PK NN AI
CON_VOLUMEN	INT	NN
CON_ID_BARRIO	INT	FK NN
CON_ID_COLOR	INT	FK NN
CON_ENTRE_CALLES	VARCHAR(100)	NN
CON_FECHA_COLOCACION	DATE	NN

TABLA **BARRIO**		
Contiene la descripción de los 50 barrios que hay en la ciudad.		
Nombre del Campo	Tipo de dato	Observaciones
BAR_ID_BARRIO	INT	PK NN AI
BAR_DESCRIPCIÓN	TINYTEXT	NN

TABLA **COLOR**		
Contiene el código 1 para los contenedores de color verde y el id 2 para los anaranjados.		
Nombre del Campo	Tipo de dato	Observaciones
COL_ID_COLOR	INT	PK NN AI
COL_DESCRIPCIÓN	TINYTEXT	NN

TABLA **SERVICIO**		
Contiene información de los 300 servicios realizados, a través de su respectivo transporte y recolector, recogiendo basura del contenedor y llevándolo a su destino correspondiente.		
Nombre del Campo	Tipo de dato	Observaciones
SER_ID_SERVICIO	INT	PK NN AI
SER_ID_TRANSPORTE	INT	FK NN
SER_ID_CONTENEDOR	INT	FK NN
SER_ID_RECOLECTOR	INT	FK NN
SER_DESTINO	VARCHAR(100)	NN
SER_FECHA_SERVICIO	DATE	NN

TABLA **TRANSPORTE**		
Contiene la información de los 20 transportes que realizan los servicios de recolección.		
Nombre del Campo	Tipo de dato	Observaciones
TRA_ID_TRANSPORTE	INT	PK NN AI
TRA_PATENTE	VARCHAR(7)	NN
TRA_MODELO	VARCHAR(25)	NN

TABLA **RECOLECTOR**		
Contiene la información de los 80 recolectores que realizan los servicios de recolección.		
Nombre del Campo	Tipo de dato	Observaciones
REC_ID_RECOLECTOR	INT	PK NN AI
REC_CUIL_CUIT	VARCHAR(12)	NN
REC_NOMBRE	VARCHAR(100)	NN
REC_APELLIDO	VARCHAR(100)	NN
REC_ID_ESTADO_CIVIL	INT	FK NN
REC_FECJA_INICIO	DATE	NN
REC_TELEFONO	VARCHAR(50)	NN
REC_EMAIL	VARCHAR(100)	NN
REC_DIRECCION	VARCHAR(100)	NN

TABLA **ESTADO CIVIL**		
Contiene la descripción de los 5 estados civiles posibles de los recolectores (ej.: soltero, casado, divorciado, etc).		
Nombre del Campo	Tipo de dato	Observaciones
EST_ID_ESTADO_CIVIL	INT	PK NN AI
EST_DESCRIPCION	TINYTEXT	NN

✓ Script de creación de tablas

https://github.com/candelaw01/ProyectoFinal/blob/811029801298a75a116d50756d677c947f586cf3/01.%20Tablas.sql

INSERCIÓN DE DATOS

Inserción manual mediante script

Se utilizó la inserción manual mediante código para los datos de las tablas “color” y “estado civil”.

```
insert into color (col_id_color, col_descripcion)
values
(1, 'verde'),
(2, 'anaranjado');
```

Imagen 3. Inserción de datos tabla "color".

```
insert into estado_civil (est_id_estado_civil,est_descripcion)
values
(1, 'solterx'),
(2, 'casadx'),
(3, 'viudx'),
(4, 'divorciadx'),
(5, 'conviviente');
```

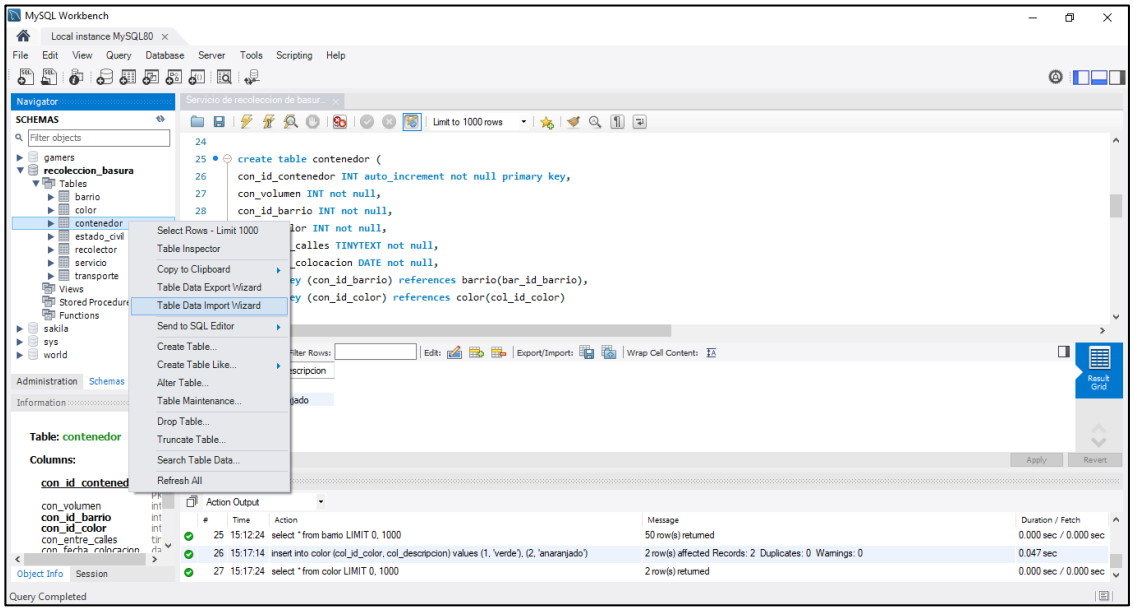
Imagen 4. Inserción de datos tabla "estado civil".

Importación de datos automática

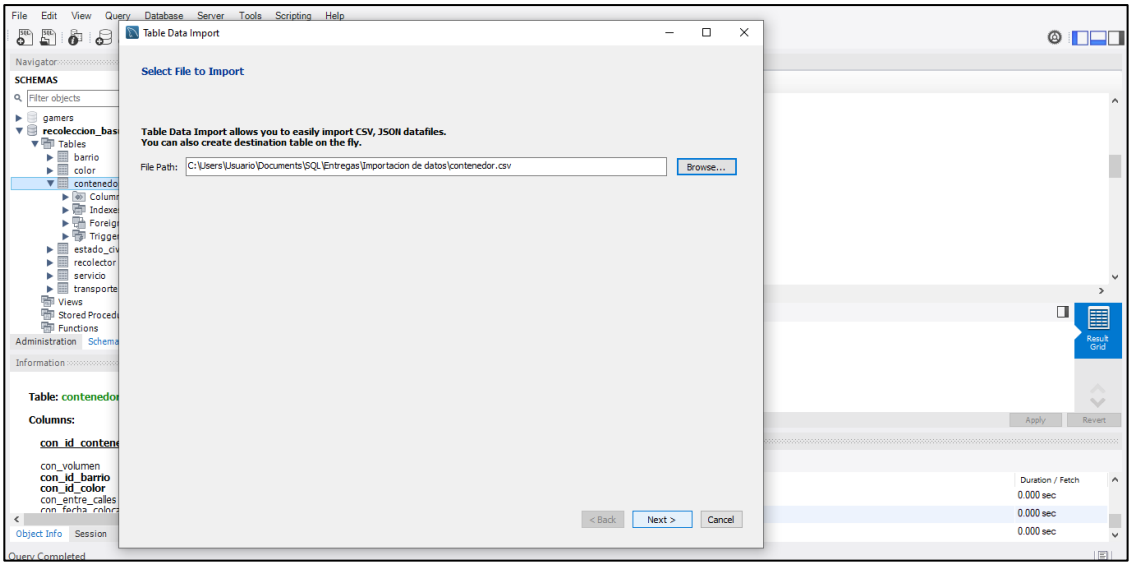
Se utilizó la importación de archivos .csv para insertar datos en las tablas “contenedor”, “barrio”, “servicio”, “transporte” y “recolector”.

A continuación, se detalla el procedimiento realizado en cada tabla:

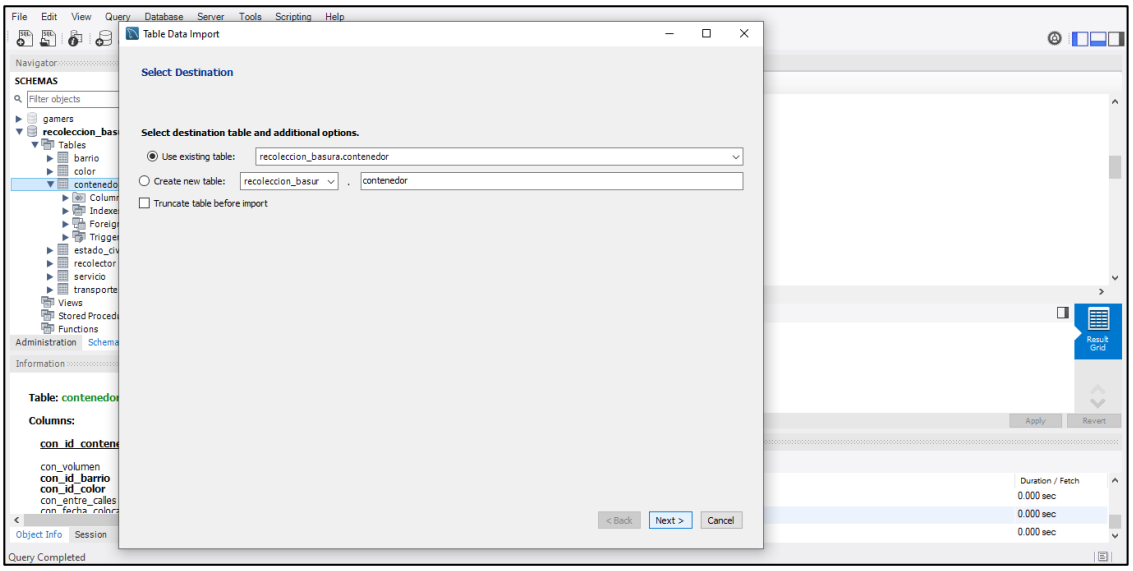
1. Hacer clic derecho en la tabla y seleccionar “Table Data Import Wizard”.



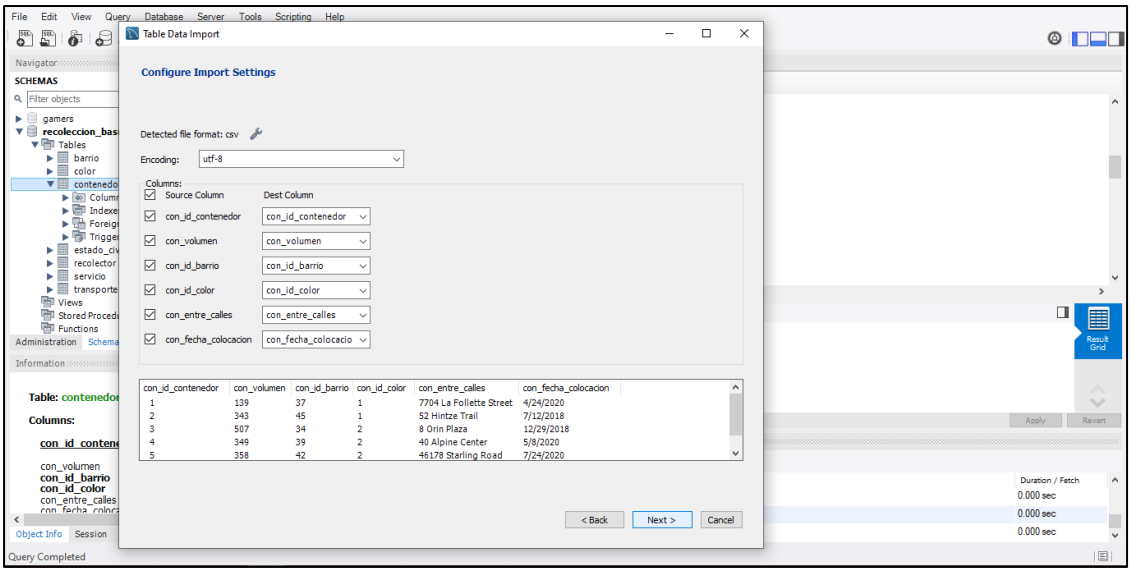
2. Seleccionar el archivo .csv a importar.



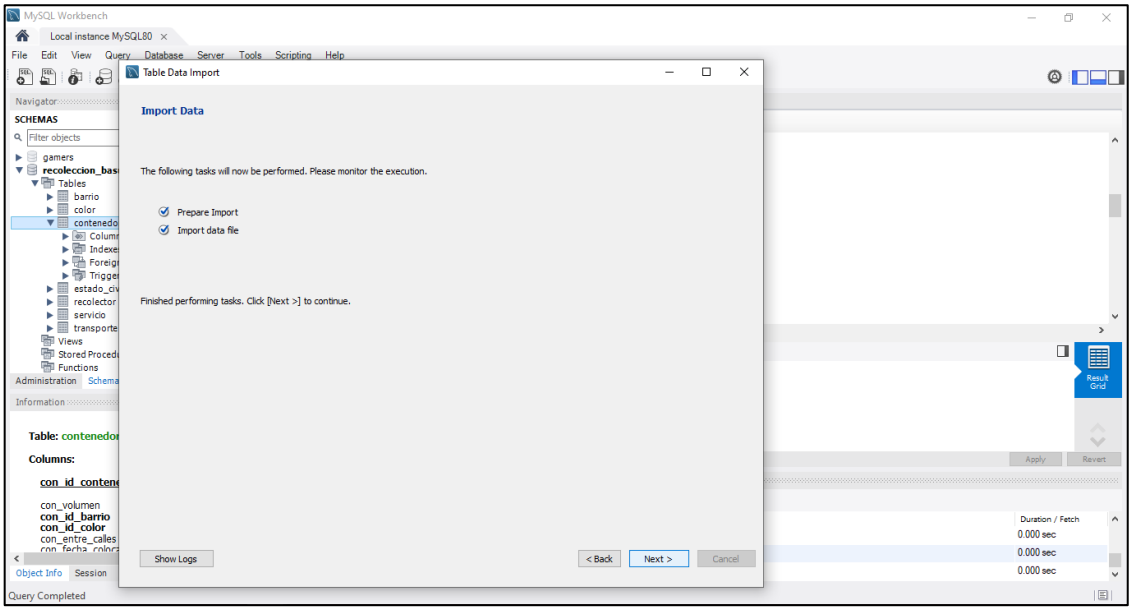
3. Elegir la tabla existente donde se deben insertar los datos.



4. Revisar que las columnas coincidan con los campos que deseamos importar.



5. Importar datos.



✓ Script de inserción de datos

https://github.com/candelaw01/ProyectoFinal/blob/811029801298a75a116d50756d677c947f586cf3/02.%20Insercion%20de%20datos.sql

✓ Archivos .csv importados

- https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/A1.%20barrio.csv
- https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/A2.%20contenedor.csv
- https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/A3.%20recolector.csv
- https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/A4.%20servicio.csv
- https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/A5.%20transporte.csv

VISTAS

La base de datos se compone de las siguientes vistas:

1. Serxtra: “servicios por transporte”. Su objetivo es visualizar la cantidad de servicios realizados por cada transporte, para sacar conclusiones acerca de qué camión realizó más recolecciones.
2. Serxrec: “servicios por recolector”. Su objetivo es visualizar la cantidad de servicios realizados por cada recolector, para sacar conclusiones acerca de qué recolector realizó mayor cantidad de recolecciones.
3. Volverde: “volumen verde”. Su objetivo es mostrar la cantidad total de residuos verdes (reciclables) recolectados. A medida que se realizan servicios de recolección de contenedores verdes, se suma el volumen de los mismos.
4. Volanaranjado: “volumen anaranjado”. Su objetivo es mostrar la cantidad total de residuos anaranjados (no reciclables) recolectados. A medida que se realizan servicios de recolección de contenedores anaranjados, se suma el volumen de los mismos.
5. Volxbarrio: “volumen por barrio”. Su objetivo es mostrar la cantidad total de residuos recolectados en cada barrio.

✓ Script de creación de vistas

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/03.%20Vistas.sql

FUNCIONES

La base de datos se compone de las siguientes funciones:

1. No_de_años: “número de años”. Obtiene la cantidad de años que hace que se colocó cada contenedor, restando la fecha actual con la fecha de colocación.
2. Con_viejo: “contenedor viejo”. Si el contenedor se colocó en la ciudad hace cuatro años o más, aparece la leyenda “realizar mantenimiento”.
3. Eco_friendly: “amigable con el medioambiente”. Si el volumen de residuos recolectados en el barrio es menor o igual a 2.000 kgs, entonces el barrio es amigable con el medioambiente. De lo contrario, la municipalidad deberá implementar políticas que incentiven el reciclaje.

✓ Script de creación de funciones

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/04.%20Funciones.sql

STORED PROCEDURES

La base de datos se compone de los siguientes Stored Procedures:

1. Sextra: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de servicios que realizó cada transporte.
2. Sexrec: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de servicios que realizó cada recolector.
3. Volanaranjado: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de residuos no reciclables que se recolectaron.
4. Volverde: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de residuos reciclables que se recolectaron.
5. Volxbarrio: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de residuos que se recolectaron en cada barrio.
6. Volxbarriofiltro: Tiene como objetivo visualizar rápidamente mediante “call”, la cantidad de residuos que se recolectaron en cada barrio, permitiendo filtrar según el ID del barrio que deseo obtener información.
7. Insert_servicio: Tiene como objetivo insertar rápidamente los servicios que se van realizando.
8. Insert_contenedor: Tiene como objetivo insertar rápidamente los contenedores nuevos que se colocan en la ciudad.

✓ Script de creación de stored procedures

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/05.%20Stored%20Procedures.sql

TRIGGERS

La base de datos se compone de los siguientes Triggers:

1. Check_servicio: Tiene como objetivo validar los nuevos IDs que se agregan a la tabla servicio.
2. Insert_servicio: Tiene como objetivo respaldar en la tabla “servicio_bitacora” los “insert” que se realicen en la tabla servicio.
3. Udate_servicio: Tiene como objetivo respaldar en la tabla “servicio_bitacora” los “update” que se realicen en la tabla servicio.
4. Delete_servicio: Tiene como objetivo respaldar en la tabla “servicio_bitacora” los “delete” que se realicen en la tabla servicio.
5. Check_contenedor: Tiene como objetivo validar los nuevos IDs que se agregan a la tabla contenedor.
6. Insert_contenedor: Tiene como objetivo respaldar en la tabla “contenedor_bitacora” los “insert” que se realicen en la tabla contenedor.
7. Udate_contenedor: Tiene como objetivo respaldar en la tabla “contenedor_bitacora” los “update” que se realicen en la tabla contenedor.
9. Delete_contenedor: Tiene como objetivo respaldar en la tabla “contenedor_bitacora” los “delete” que se realicen la tabla contenedor.

✓ Script de creación de triggers

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/06.%20Triggers.sql

DATA CONTROL LANGUAGE

Se crearon dos usuarios nuevos a los cuales se les asignó una serie de permisos: uno de los usuarios creados tiene permiso de sólo lectura sobre todas las tablas; y el otro usuario tiene permisos de lectura, inserción y modificación de datos. Ninguno de ellos puede eliminar registros de ninguna tabla.

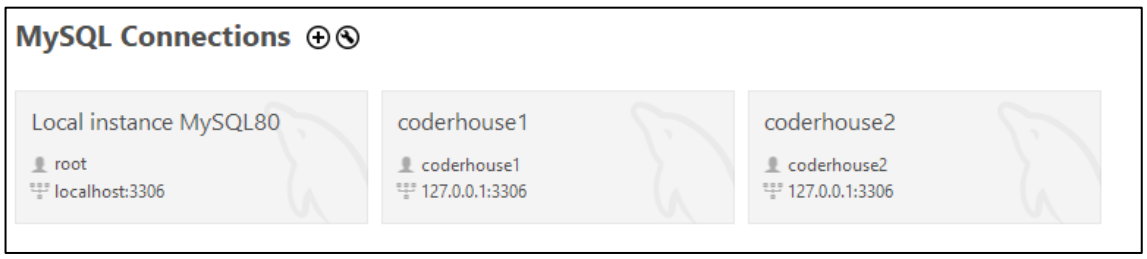


Imagen 5. Usuarios creados con DDL.

✓ Script de implementación de sentencias

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/07.%20Data%20Control%20Language.sql

TRANSACTION CONTROL LANGUAGE

Se eligieron las tablas principales “servicio” y “contenedor” para realizar modificaciones en los registros, controladas por transacciones.

En la tabla “servicio” se elimina uno de los registros iniciando previamente la transacción. En la tabla “contenedor” se insertan ocho nuevos registros, también iniciando la transacción, y agregando un savepoint luego del registro #4 y del #8.

✓ Script de implementación de sentencias

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/08.%20Transaction%20Control%20Language.sql

BACKUP

Se generó un backup de la base de datos, incluyendo sólo los datos de las tablas.

✓ Script de creación del backup

https://github.com/candelaw01/ProyectoFinal/blob/0159c9e5a31b1a34592b2c7a21ffbb8712709534/09.%20Backup.sql

Informes generados

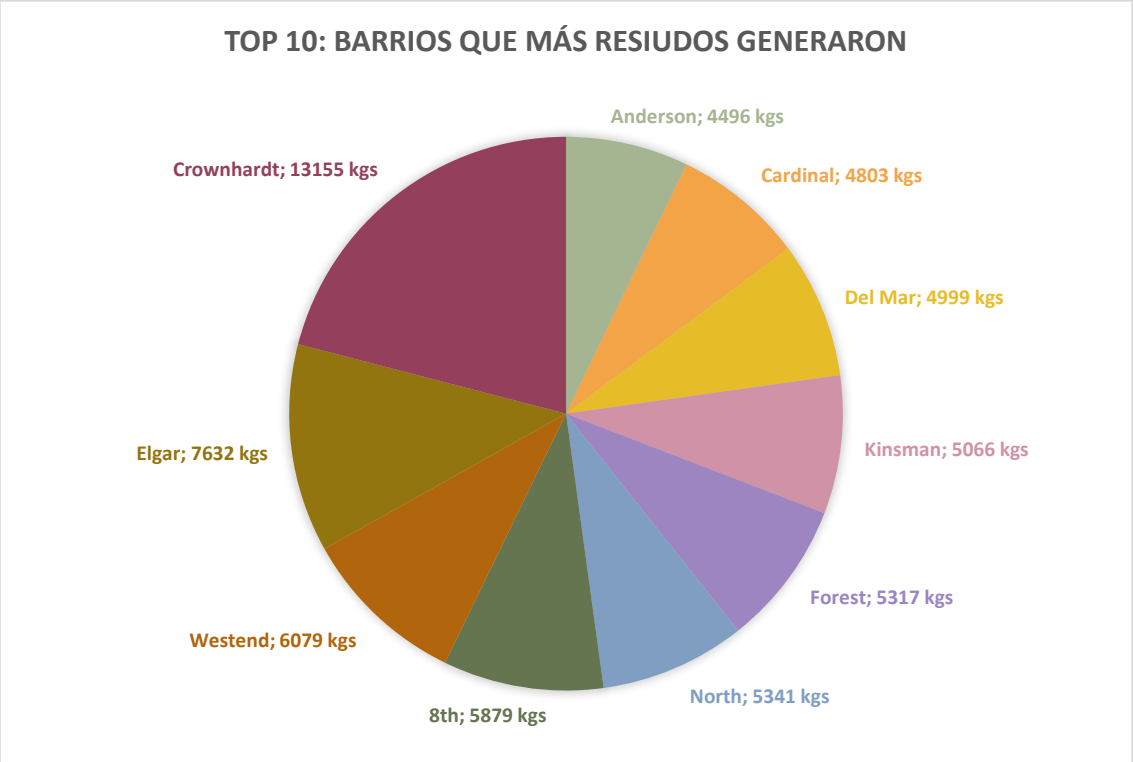


Imagen 6. Los diez barrios que más residuos generaron.

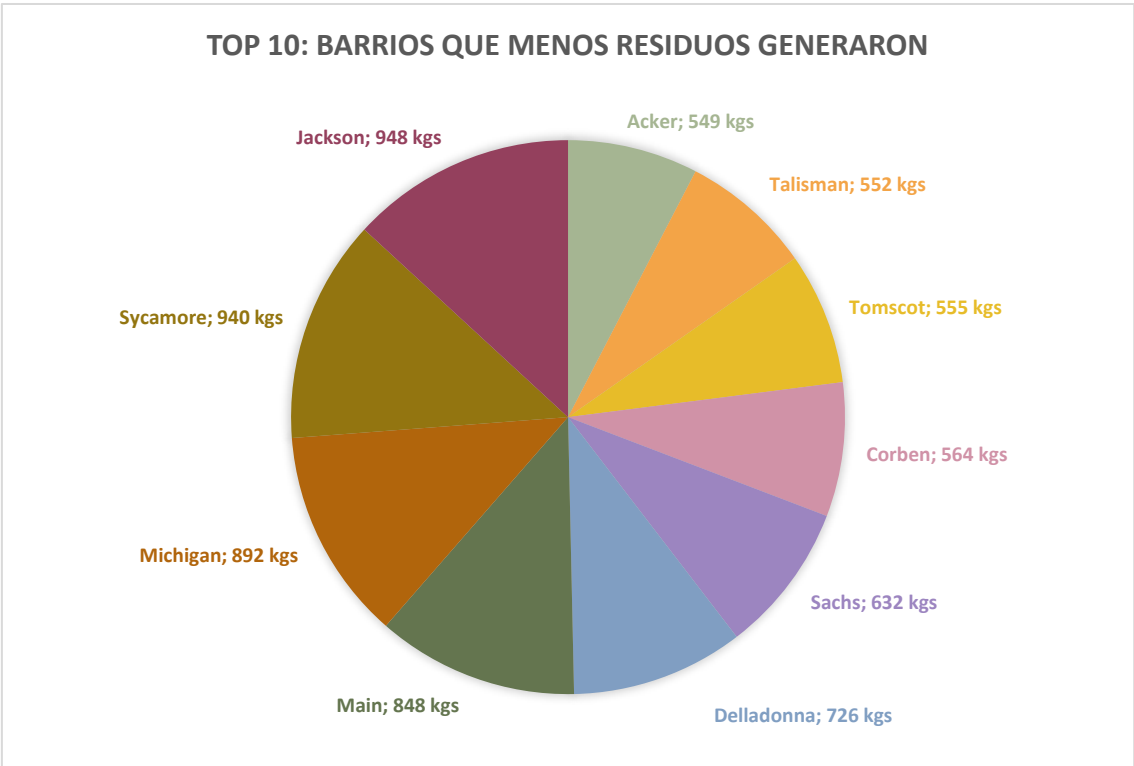


Imagen 7. Los diez barrios que menos residuos generaron.

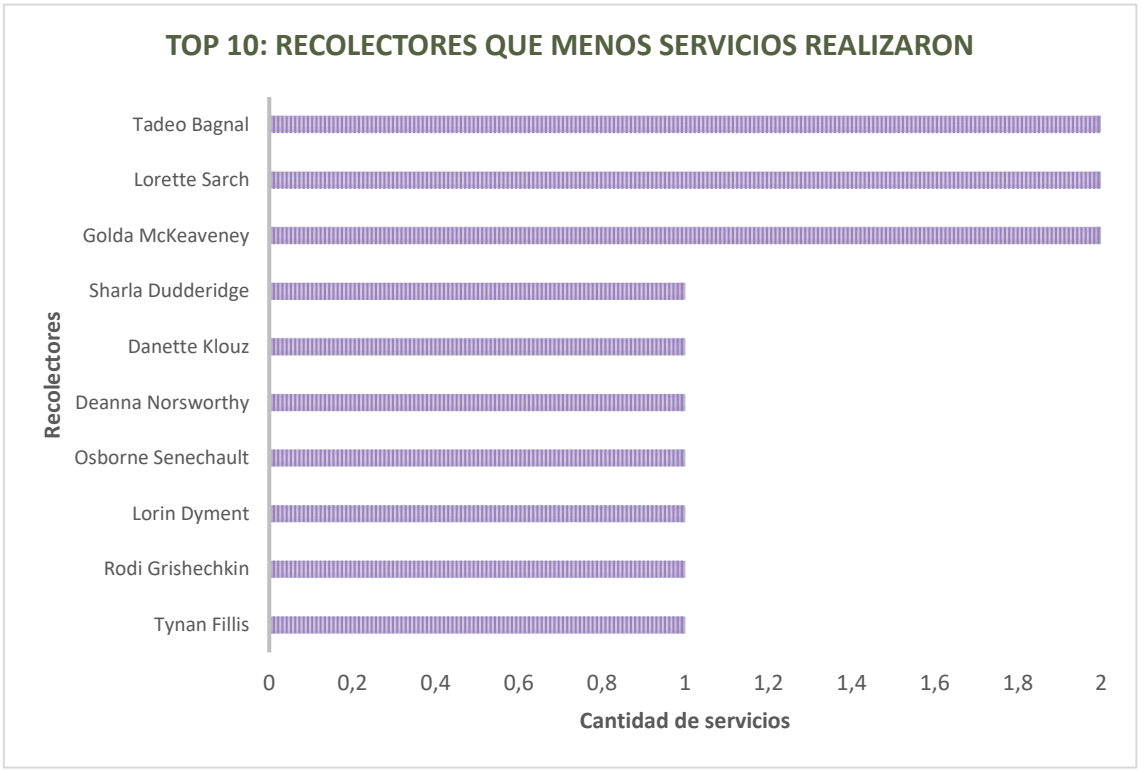


Imagen 8. Los diez recolectores que menos servicios realizaron.

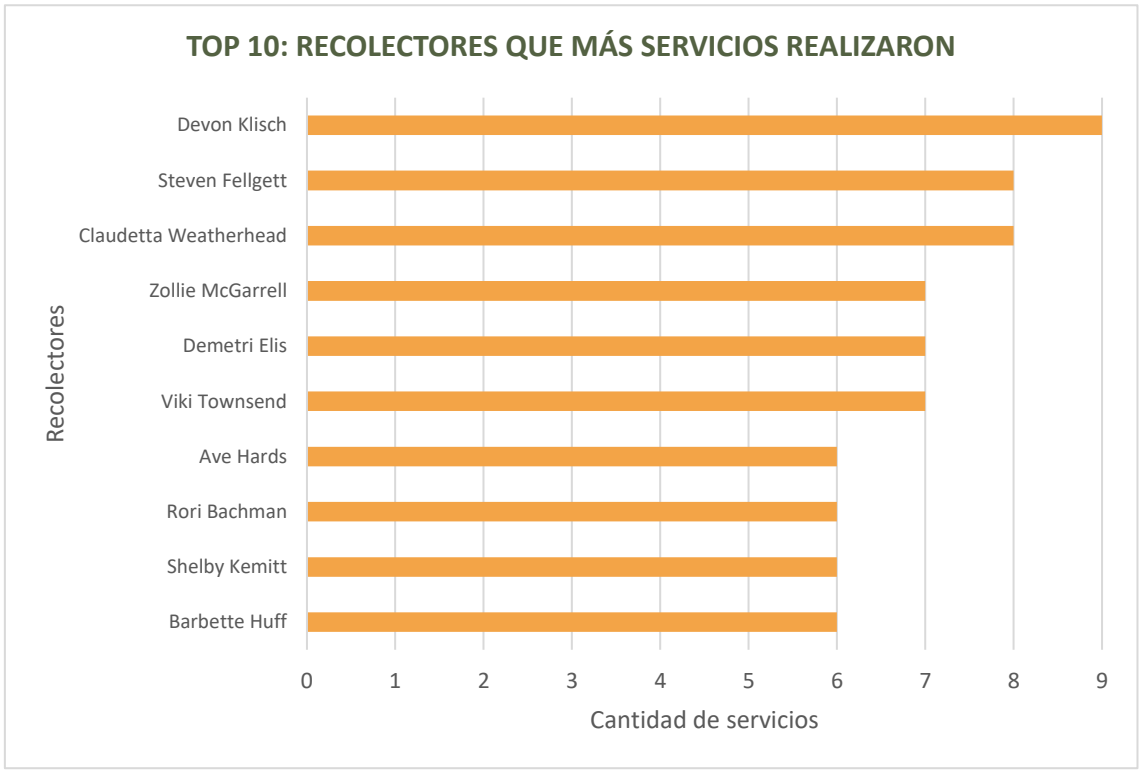


Imagen 9. Los diez recolectores que más servicios realizaron.

ID Barrio	Descripción	Volumen (kgs)	Observaciones
1	Butterfield	1137	Barrio ecofriendly
2	Melody	2760	Implementar políticas de reciclaje
3	Cordelia	3242	Implementar políticas de reciclaje
4	Nova	2435	Implementar políticas de reciclaje
5	Comanche	4227	Implementar políticas de reciclaje
6	8th	5879	Implementar políticas de reciclaje
8	Corben	564	Barrio ecofriendly
9	Old Shore	1242	Barrio ecofriendly
10	North	5341	Implementar políticas de reciclaje
11	Victoria	1988	Barrio ecofriendly
12	Carioca	2010	Implementar políticas de reciclaje
13	Steensland	1808	Barrio ecofriendly
14	Jackson	948	Barrio ecofriendly
15	Clove	1818	Barrio ecofriendly
17	Prentice	2436	Implementar políticas de reciclaje
18	Anderson	4496	Implementar políticas de reciclaje
19	Main	848	Barrio ecofriendly
20	Talisman	552	Barrio ecofriendly
21	Warbler	1038	Barrio ecofriendly
22	Kinsman	5066	Implementar políticas de reciclaje
23	Del Mar	4999	Implementar políticas de reciclaje
24	Westend	6079	Implementar políticas de reciclaje
25	Fallview	1637	Barrio ecofriendly
26	Elgar	7632	Implementar políticas de reciclaje
27	Kings	1615	Barrio ecofriendly
29	Sachs	632	Barrio ecofriendly
32	Tennessee	2188	Implementar políticas de reciclaje
33	Delladonna	726	Barrio ecofriendly
35	Fulton	3259	Implementar políticas de reciclaje
36	Crownhardt	13155	Implementar políticas de reciclaje
37	Tomscot	555	Barrio ecofriendly
38	Mosinee	3906	Implementar políticas de reciclaje
39	Acker	549	Barrio ecofriendly
40	Fordem	1440	Barrio ecofriendly
42	Michigan	892	Barrio ecofriendly
43	Cardinal	4803	Implementar políticas de reciclaje
44	Forest	5317	Implementar políticas de reciclaje
45	North	1279	Barrio ecofriendly
46	Sycamore	940	Barrio ecofriendly
48	Magdeline	1632	Barrio ecofriendly
49	Vidon	2079	Implementar políticas de reciclaje
50	Heffernan	3882	Implementar políticas de reciclaje

Imagen 10. Observaciones a los barrios según la cantidad de residuos generados.

ID contenedor	Entre calles	Años	Observaciones
3	60 Canary Drive	5	Realizar mantenimiento
5	730 Springview Pass	6	Realizar mantenimiento
7	5303 Esch Point	4	Realizar mantenimiento
8	7 Scofield Alley	6	Realizar mantenimiento
12	7 Coleman Junction	5	Realizar mantenimiento
13	02002 Oakridge Center	4	Realizar mantenimiento
14	07 Hallows Road	4	Realizar mantenimiento
15	37402 Burrows Road	5	Realizar mantenimiento
17	3 Sachtjen Place	5	Realizar mantenimiento
19	806 Mcbride Avenue	5	Realizar mantenimiento
20	928 Londonderry Junction	5	Realizar mantenimiento
23	21 Mifflin Junction	4	Realizar mantenimiento
29	712 Prairie Rose Street	5	Realizar mantenimiento
32	851 Thackeray Alley	5	Realizar mantenimiento
35	60178 Mallory Pass	5	Realizar mantenimiento
36	7988 Ilene Alley	4	Realizar mantenimiento
39	70 Melrose Place	4	Realizar mantenimiento
40	7284 Mayer Circle	6	Realizar mantenimiento
42	70 Alpine Crossing	6	Realizar mantenimiento
43	301 Grim Trail	5	Realizar mantenimiento
46	729 Havey Circle	4	Realizar mantenimiento
48	60 Village Green Circle	5	Realizar mantenimiento
50	13 Morrow Alley	5	Realizar mantenimiento
51	05 Park Meadow Junction	5	Realizar mantenimiento
53	0596 Maywood Lane	4	Realizar mantenimiento
58	5 Haas Street	5	Realizar mantenimiento
59	26777 Service Pass	4	Realizar mantenimiento
61	7725 Prentice Park	5	Realizar mantenimiento
62	6 Rutledge Park	4	Realizar mantenimiento
63	22 Grim Junction	4	Realizar mantenimiento
65	8 Lunder Way	5	Realizar mantenimiento
66	718 Del Sol Alley	5	Realizar mantenimiento
67	96 Sunnyside Junction	5	Realizar mantenimiento
71	638 6th Point	5	Realizar mantenimiento
73	94 Oak Valley Avenue	5	Realizar mantenimiento
75	3527 Green Ridge Hill	4	Realizar mantenimiento
77	86 Manitowish Park	4	Realizar mantenimiento
79	6849 Stang Circle	4	Realizar mantenimiento
81	8123 Lukken Place	4	Realizar mantenimiento
85	359 Onsgard Drive	4	Realizar mantenimiento
86	57 Onsgard Trail	4	Realizar mantenimiento
88	7 Banding Pass	5	Realizar mantenimiento
91	394 Glacier Hill Place	5	Realizar mantenimiento
96	7 Bowman Way	4	Realizar mantenimiento

97	36660 Steensland Hill	5	Realizar mantenimiento
99	69914 Trailsway Trail	5	Realizar mantenimiento

Imagen 11. Contenedores a los que hay que realizarles mantenimiento por la cantidad de años que hace que se colocaron.

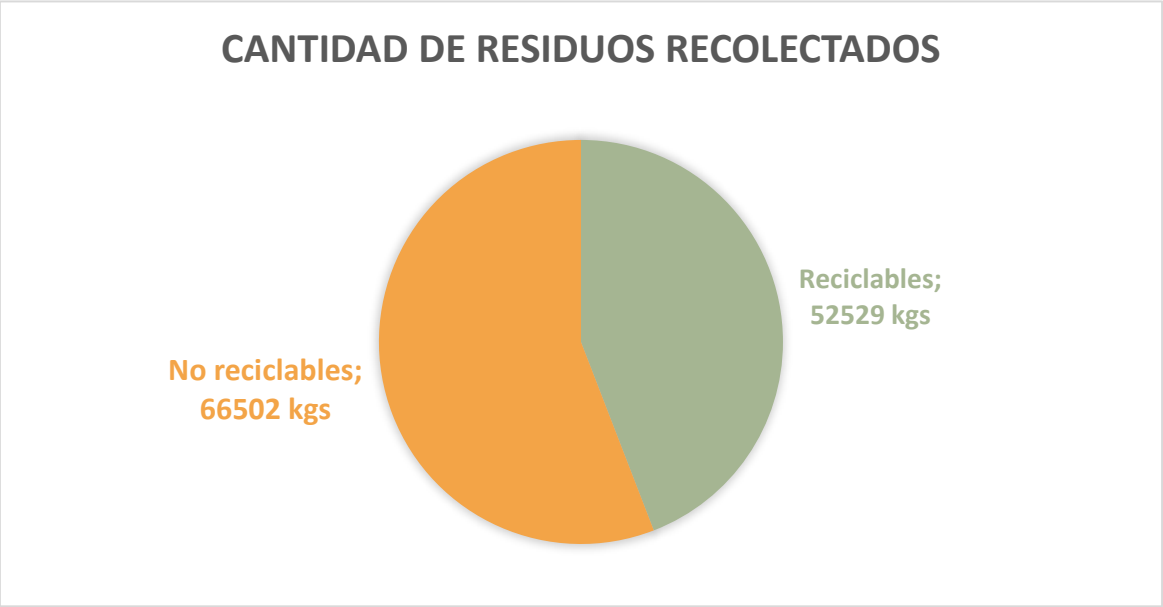


Imagen 12. Cantidad de residuos reciclables y no reciclables generados en la ciudad.

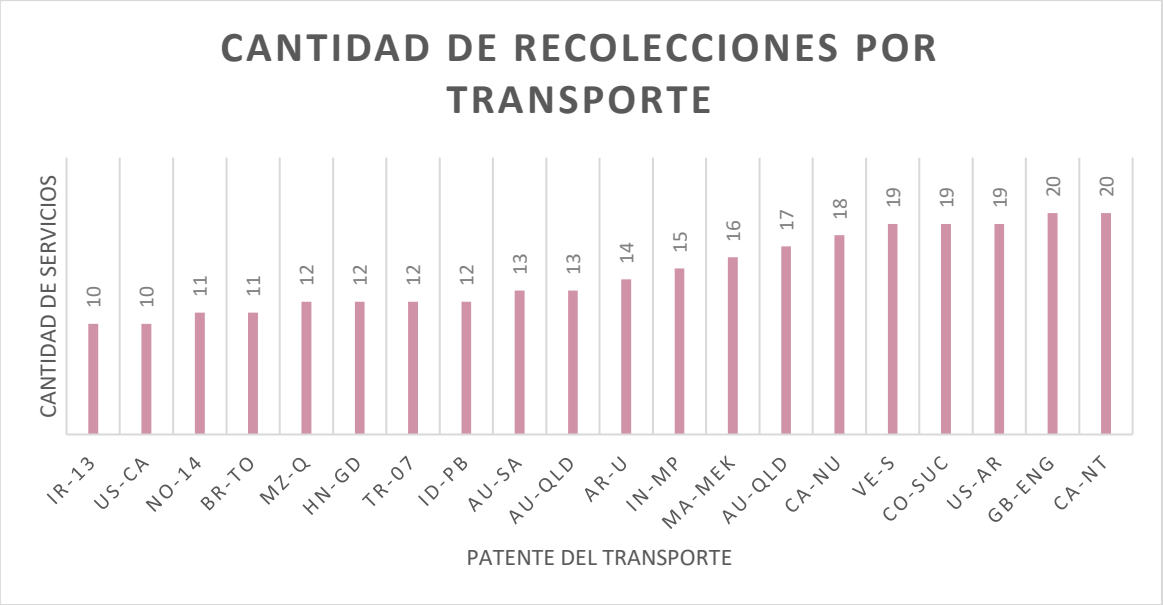


Imagen 13. Cantidad de servicios realizados según el vehículo (patente).

Herramientas utilizadas

- My SQL Workbench
- Repositorio GitHub
- Microsoft Excel

Conclusión

Las bases de datos traen aparejados múltiples beneficios, ya que pueden recopilar una gran cantidad de datos. Cuando estos datos son manipulados por un usuario que posee los conocimientos necesarios, se puede generar información valiosa.

En el caso en cuestión, se recopilaron datos sobre un sistema de recolección de basura, y a través de la generación de informes, el municipio de “Green City” puede tomar decisiones con una alta probabilidad de éxito.

Los informes generados permiten sacar conclusiones sobre cuáles barrios son más amigables con el medioambiente, y en cuáles hay que reforzar las políticas de sostenibilidad y sustentabilidad. Además, se puede premiar a los recolectores que mayores servicios realizan para contribuir a la higiene urbana, entre otras acciones.

En estos tiempos donde el cuidado del planeta es crítico, es importante que las ciudades tengan un sistema que permita el desarrollo y control de sus residuos, para que los habitantes puedan vivir plenamente, y se contribuya al bien común.