Iteración 2

Análisis:

* Los diagramas no cambian.

Diseño de la aplicación:

* Índices y consultas:
* RFC1

SELECT h.id AS habitacion\_id,

COALESCE(SUM(c.costo), 0) AS dinero\_recolectado

FROM

habitaciones h

LEFT JOIN

consumos c ON h.id = c.habitaciones\_id

WHERE

c.fecha\_consumo >= ADD\_MONTHS(CURRENT\_DATE, -12)

GROUP BY

h.id;

No es necesario añadir un índice extra ya que lo único critico que usa es el id de la habitación.

* RFC2

SELECT

s.tipo AS servicio,

COUNT(c.tiposservicio\_tipo) AS cantidad\_consumos

FROM

consumos c

INNER JOIN

tiposservicio s ON c.tiposservicio\_tipo = s.tipo

WHERE

c.fecha\_consumo >= TRUNC(SYSDATE - INTERVAL '15' MONTH) -- Período de 6 meses

GROUP BY

s.tipo

ORDER BY

cantidad\_consumos DESC

FETCH FIRST 20 ROWS ONLY;

* RFC3

WITH ReservasEnUltimoAnio AS (

SELECT

r.habitaciones\_id,

GREATEST(rs.fecha\_inicio, (SYSDATE - INTERVAL '1' YEAR)) AS fecha\_entrada,

LEAST(rs.fecha\_salida, SYSDATE) AS fecha\_salida

FROM

reservan r

INNER JOIN

reservas rs ON r.reservas\_id = rs.id

WHERE

rs.fecha\_inicio <= SYSDATE

AND rs.fecha\_salida >= (SYSDATE - INTERVAL '1' YEAR)

)

SELECT

h.id AS habitacion\_id,

ROUND(COALESCE(SUM(TO\_NUMBER(fecha\_salida - fecha\_entrada) + 1), 0)) AS dias\_ocupados\_en\_ultimo\_anio,

ROUND(COALESCE((SUM(TO\_NUMBER(fecha\_salida - fecha\_entrada) + 1) / 365) \* 100, 0)) AS porcentaje\_ocupacion\_en\_ultimo\_anio

FROM

habitaciones h

LEFT JOIN

ReservasEnUltimoAnio ro ON h.id = ro.habitaciones\_id

GROUP BY

h.id;

* RFC5

SELECT

u.num\_documento AS numero\_documento,

u.nombre,

SUM(c.costo) AS suma\_consumos

FROM

reservas r

INNER JOIN

habitaciones h ON r.id = h.id

INNER JOIN

consumos c ON h.id = c.habitaciones\_id

INNER JOIN

usuarios u ON r.usuarios\_num\_documento = u.num\_documento AND r.usuarios\_tipo\_documento = u.tipo\_documento

WHERE

u.num\_documento = '1765432' -- Reemplaza con el número de documento del usuario que deseas buscar

AND c.fecha\_consumo BETWEEN '01/01/2023' AND '30/12/2023' -- Reemplaza con el rango de fechas deseado

GROUP BY

u.num\_documento, u.nombre;

* RFC6  
    
  WITH FechasReservas AS (

SELECT DISTINCT fecha AS fecha\_reserva

FROM (

SELECT fecha\_inicio AS fecha FROM reservas

UNION ALL

SELECT fecha\_salida AS fecha FROM reservas))

SELECT fr.fecha\_reserva, COUNT(r.id) AS reservas\_activas

FROM FechasReservas fr

JOIN reservas r ON fr.fecha\_reserva BETWEEN r.fecha\_inicio AND r.fecha\_salida

GROUP BY fr.fecha\_reserva

ORDER BY reservas\_activas DESC

FETCH FIRST 10 ROWS ONLY;

--- B)

WITH FechasConsumos AS (

SELECT DISTINCT fecha\_consumo AS fecha

FROM consumos)

SELECT fc.fecha, SUM(c.costo) AS costo\_consumos, COUNT(c.costo) AS consumos\_realizados

FROM FechasConsumos fc

JOIN consumos c ON fc.fecha = c.fecha\_consumo

GROUP BY fc.fecha

ORDER BY costo\_consumos DESC

FETCH FIRST 10 ROWS ONLY;

--- C

WITH FechasReservas AS (

SELECT DISTINCT fecha AS fecha\_reserva

FROM (

SELECT fecha\_inicio AS fecha FROM reservas

UNION ALL

SELECT fecha\_salida AS fecha FROM reservas))

SELECT fr.fecha\_reserva, COUNT(r.id) AS reservas\_activas

FROM FechasReservas fr

JOIN reservas r ON fr.fecha\_reserva BETWEEN r.fecha\_inicio AND r.fecha\_salida

GROUP BY fr.fecha\_reserva

ORDER BY reservas\_activas ASC

FETCH FIRST 10 ROWS ONLY;

* RFC7

WITH BuenClienteTiempo AS (

SELECT DISTINCT r.usuarios\_num\_documento AS num\_documento\_cliente

FROM reservas r

WHERE (r.fecha\_salida - r.fecha\_inicio) >= 14), BuenClienteConsumo AS (

SELECT r.usuarios\_num\_documento AS num\_documento\_cliente, SUM(c.costo) AS suma\_consumos

FROM reservas r

JOIN consumos c ON r.id = c.habitaciones\_id

WHERE c.fecha\_consumo >= (SELECT MAX(fecha\_consumo) FROM consumos) - INTERVAL '1' YEAR

GROUP BY r.usuarios\_num\_documento

HAVING SUM(c.costo) > 15000000)

SELECT u.num\_documento AS num\_documento\_cliente, u.nombre AS nombre\_cliente,

CASE

WHEN u.num\_documento IN (SELECT num\_documento\_cliente FROM BuenClienteConsumo) THEN TO\_CHAR((SELECT suma\_consumos FROM BuenClienteConsumo WHERE num\_documento\_cliente = u.num\_documento), '$999,999,999.99')

ELSE 'Buen usuario por tiempo'

END AS Razon

FROM usuarios u

WHERE u.num\_documento IN (

SELECT num\_documento\_cliente FROM BuenClienteConsumo

UNION

SELECT num\_documento\_cliente FROM BuenClienteTiempo);

* RFC8

WITH ConsumosPorSemana AS (

SELECT t.tipo, TO\_CHAR(c.fecha\_consumo, 'IYYY-IW') AS semana, COUNT(\*) AS veces\_consumido

FROM consumos c INNER JOIN tiposservicio t ON c.tiposservicio\_tipo = t.tipo

WHERE c.fecha\_consumo BETWEEN TO\_DATE('2023-01-01', 'YYYY-MM-DD') AND TO\_DATE('2023-12-31', 'YYYY-MM-DD')

GROUP BY t.tipo, TO\_CHAR(c.fecha\_consumo, 'IYYY-IW') )

SELECT s.tipo, s.semana, s.veces\_consumido

FROM ConsumosPorSemana s

WHERE s.veces\_consumido < 3;

* RFC9

SELECT

r.usuarios\_num\_documento AS num\_documento\_cliente,

r.usuarios\_nombre AS nombre\_cliente,

c.tiposservicio\_tipo AS tipo\_servicio,

COUNT(c.tiposservicio\_tipo) AS veces\_utilizado

FROM consumos c

LEFT JOIN reservan rv ON c.habitaciones\_id = rv.habitaciones\_id

LEFT JOIN reservas r ON rv.reservas\_id = r.id

WHERE c.tiposservicio\_tipo = 'spa' AND c.fecha\_consumo BETWEEN TO\_DATE('2023-01-01', 'YYYY-MM-DD') AND TO\_DATE('2023-12-31', 'YYYY-MM-DD')

GROUP BY r.usuarios\_num\_documento, r.usuarios\_nombre, c.tiposservicio\_tipo

ORDER BY r.usuarios\_nombre ASC;

* RFC10

SELECT

u.num\_documento AS num\_documento\_cliente,

u.nombre AS nombre\_cliente

FROM

usuarios u

WHERE

u.num\_documento NOT IN (

SELECT

r.usuarios\_num\_documento AS num\_documento\_cliente

FROM

consumos c

LEFT JOIN

reservan rv ON c.habitaciones\_id = rv.habitaciones\_id

LEFT JOIN

reservas r ON rv.reservas\_id = r.id

WHERE

c.tiposservicio\_tipo = 'supermercado'

AND c.fecha\_consumo BETWEEN TO\_DATE('2023-01-01', 'YYYY-MM-DD') AND TO\_DATE('2023-12-31', 'YYYY-MM-DD')

GROUP BY r.usuarios\_num\_documento, r.usuarios\_nombre, c.tiposservicio\_tipo

);

* RFC11

WITH Semanas AS (

SELECT

TO\_CHAR(fecha\_inicio, 'IYYY-IW') AS semana,

reservan.reservas\_id AS reserva\_id,

consumos.tiposservicio\_tipo AS tipoServicio,

reservan.habitaciones\_id AS habitacion\_id

FROM reservas

JOIN reservan ON reservan.reservas\_id = reservas.id

JOIN consumos ON reservan.habitaciones\_id = consumos.habitaciones\_id),

ServiciosConsumidos AS (

SELECT

semana,

tipoServicio,

COUNT(\*) AS total\_consumos

FROM Semanas

GROUP BY semana, tipoServicio),

HabitacionesSolicitadas AS (

SELECT

semana,

habitacion\_id,

COUNT(\*) AS total\_solicitudes

FROM Semanas

GROUP BY semana, habitacion\_id),

MaxMinServicios AS (

SELECT

semana,

MAX(total\_consumos) AS max\_consumos,

MIN(total\_consumos) AS min\_consumos

FROM ServiciosConsumidos

GROUP BY semana

), MaxMinHabitaciones AS (

SELECT

semana,

MAX(total\_solicitudes) AS max\_solicitudes,

MIN(total\_solicitudes) AS min\_consumos

FROM HabitacionesSolicitadas

GROUP BY semana)

SELECT

s.semana,

MAX(CASE WHEN sc.total\_consumos = mms.max\_consumos THEN tipoServicio END) AS servicio\_mas\_consumido,

MIN(CASE WHEN sc.total\_consumos = mms.min\_consumos THEN tipoServicio END) AS servicio\_menos\_consumido,

MAX(CASE WHEN hs.total\_solicitudes = mmh.max\_solicitudes THEN habitacion\_id END) AS habitacion\_mas\_solicitada,

MIN(CASE WHEN hs.total\_solicitudes = mmh.min\_consumos THEN habitacion\_id END) AS habitacion\_menos\_solicitada

FROM (SELECT DISTINCT semana FROM Semanas) s

JOIN MaxMinServicios mms ON s.semana = mms.semana

JOIN MaxMinHabitaciones mmh ON s.semana = mmh.semana

LEFT JOIN ServiciosConsumidos sc ON s.semana = sc.semana

LEFT JOIN HabitacionesSolicitadas hs ON s.semana = hs.semana

GROUP BY s.semana

ORDER BY s.semana;

* RFC12

--- Reservas por trimestre

SELECT usuarios\_nombre AS nombre\_cliente, usuarios\_num\_documento AS num\_documento\_cliente, usuarios\_correo AS correo

FROM reservas

WHERE fecha\_inicio BETWEEN TO\_DATE('2023-01-01', 'YYYY-MM-DD') AND TO\_DATE('2023-03-31', 'YYYY-MM-DD')

AND usuarios\_num\_documento IN (

SELECT usuarios\_num\_documento

FROM reservas

WHERE fecha\_inicio BETWEEN TO\_DATE('2023-05-01', 'YYYY-MM-DD') AND TO\_DATE('2023-08-30', 'YYYY-MM-DD')

) AND usuarios\_num\_documento IN (

SELECT usuarios\_num\_documento

FROM reservas

WHERE fecha\_inicio BETWEEN TO\_DATE('2023-09-01', 'YYYY-MM-DD') AND TO\_DATE('2023-12-30', 'YYYY-MM-DD'));

---Por lo menos un consumo de servicio costoso

SELECT

r.usuarios\_nombre AS nombre\_cliente,

r.usuarios\_num\_documento AS num\_documento\_cliente,

r.usuarios\_correo AS correo

FROM reservas r

WHERE r.usuarios\_num\_documento IN (

SELECT r2.usuarios\_num\_documento

FROM reservas r2

WHERE EXISTS (

SELECT 1

FROM consumos c

WHERE c.fecha\_consumo BETWEEN r2.fecha\_inicio AND r2.fecha\_salida AND c.costo >= 300000))

GROUP BY r.usuarios\_num\_documento, r.usuarios\_nombre,r.usuarios\_correo ;

-------

SELECT

r.usuarios\_nombre AS nombre,

r.usuarios\_num\_documento AS num\_doc,

r.usuarios\_correo

FROM RESERVAS r

WHERE r.usuarios\_num\_documento NOT IN (

SELECT DISTINCT re.usuarios\_num\_documento

FROM

RESERVASSERVICIO rs

INNER JOIN RESERVAN ON rs.habitaciones\_id = reservan.habitaciones\_id

INNER JOIN RESERVAS re ON rs.dia BETWEEN re.fecha\_inicio AND re.fecha\_salida

WHERE rs.duracion\_hora < 4)

GROUP BY r.usuarios\_nombre, r.usuarios\_num\_documento, r.usuarios\_correo;

* Índices:

Para mejorar el acceso a las tablas por medio de la agrupación de tipos de servicio:

CREATE

INDEX idx\_consumos\_tiposservicio

ON consumos(tiposservicio\_tipo);

Para mejorar la consulta cuando tienen que acceder por la fecha a las reservas:

CREATE

INDEX idx\_consumos\_fecha

ON consumos(fecha\_consumo);

Para la agrupación de reservas por usuario:

CREATE

INDEX idx\_reservas\_num\_doc

ON reservas(usuarios\_num\_documento);

Para acceder a las reservas por habitación y fecha:

CREATE

INDEX idx\_consumos\_habitaciones\_fecha

ON consumos(habitaciones\_id, fecha\_consumo);

Agrupación de reservas por el id de la habitación:

CREATE

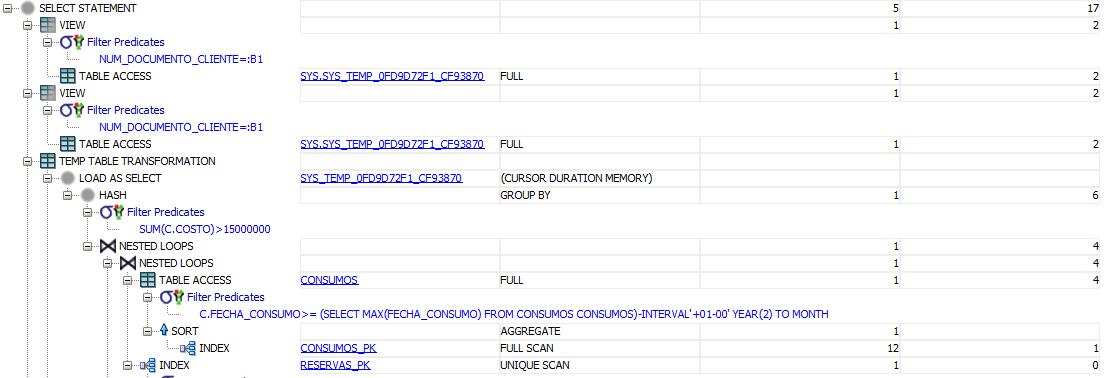
INDEX idx\_reservan\_habitaciones\_id

ON reservan(habitaciones\_id);

* Comparación de planes de consulta:

Para RFC7:

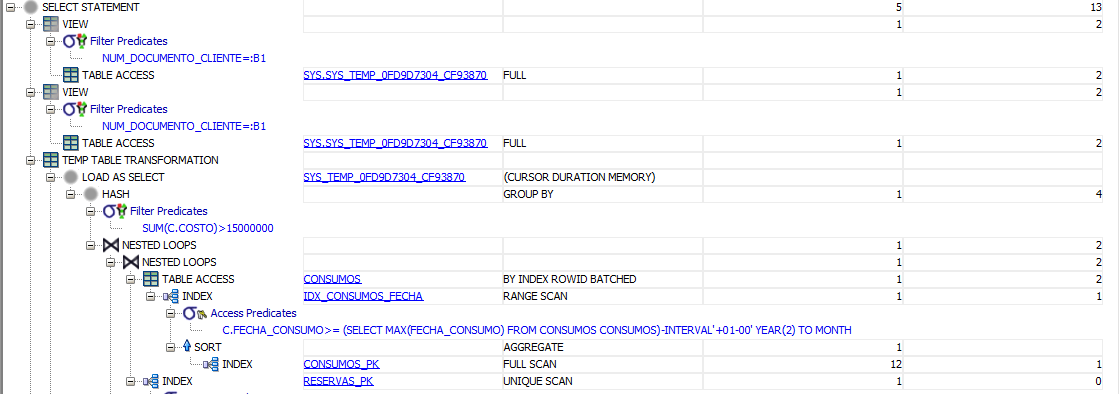
Sin indicadores



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Con indicadores:

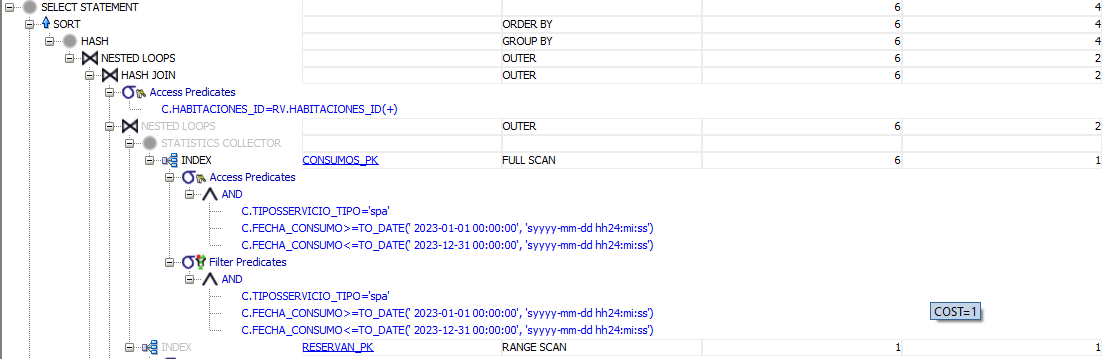


Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para RFC9:

Sin indicadores



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Con indicadores

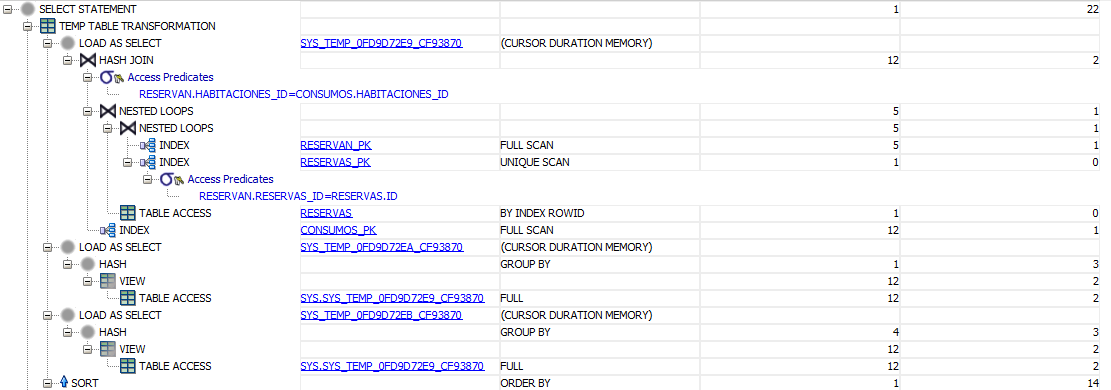
Interfaz de usuario gráfica, Aplicación

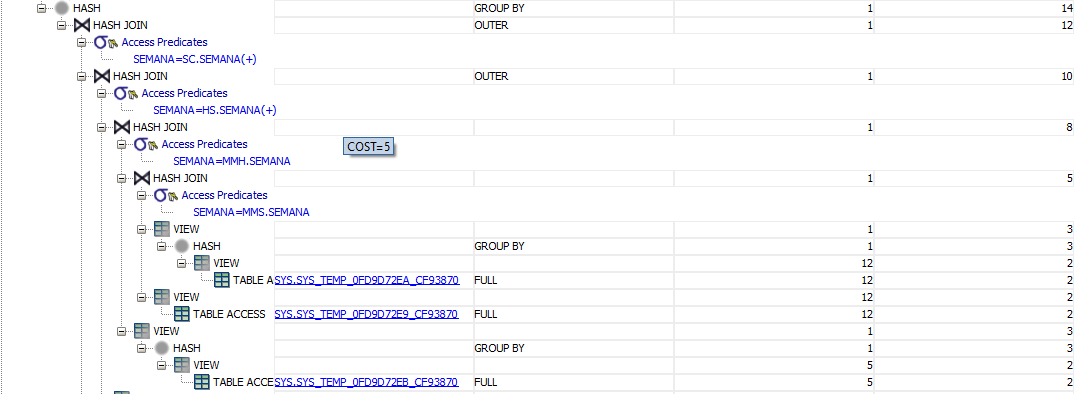
Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

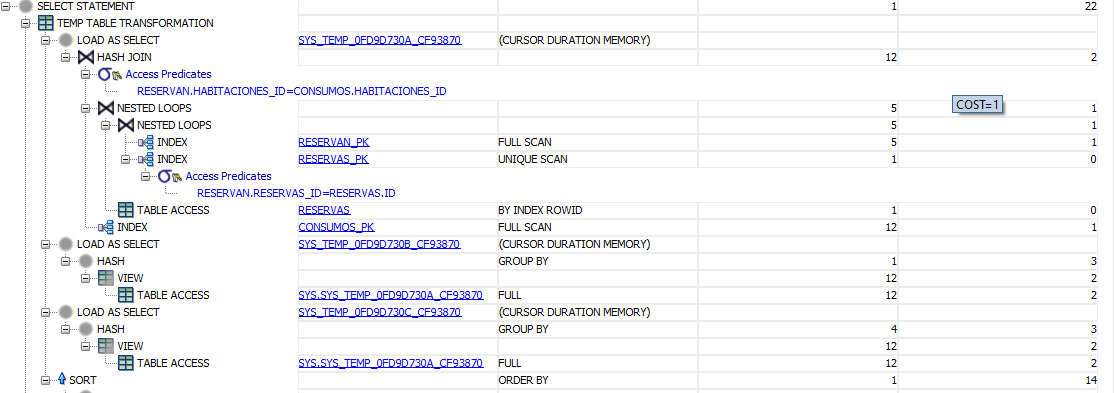
Descripción generada automáticamente

Para RFC11:

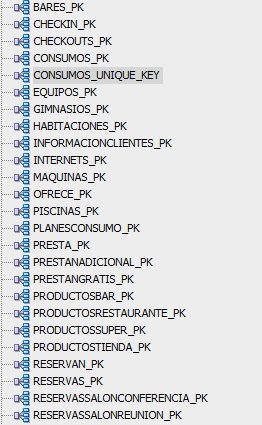
Sin indicadores  




Con indicadores:



Se nota una pequeña mejoría en los costos, sin embargo al ser consultas tan complejas el costo no baja tanto como se espera.

* Creados automáticamente por SQL:  
  

Básicamente son las llaves primarias de cada tabla, importantes para reunir la información por un valor en específico.

* Para los casos de prueba se usaron pocos registros que cumplen las características para cada requerimiento, para cada uno se recibieron las respuestas esperadas.

RFC1:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

RFC2

Rango de fechas: 01/01/2023, 30/11/2023

Tabla

Descripción generada automáticamente

RFC3

(muestra los índices redondeados)

Tabla

Descripción generada automáticamente

RFC5

Rango de fechas: 01/01/2023, 30/11/2023. Num doc = 111111112

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

RFC6

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Aplicación

Descripción generada automáticamente con confianza baja

RFC7

Interfaz de usuario gráfica

Descripción generada automáticamente

RFC8

(Se probó con 3 consumos en una misma semana, no aparecen)

Consumos:

Interfaz de usuario gráfica, Texto, Aplicación

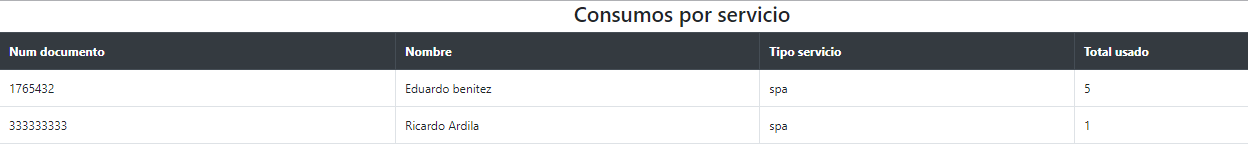
Descripción generada automáticamenteResultado:

Tabla

Descripción generada automáticamente

RFC9

Rango de fechas: 01/01/2023, 30/11/2023. Tipo de servicio = ‘spa’



RFC10

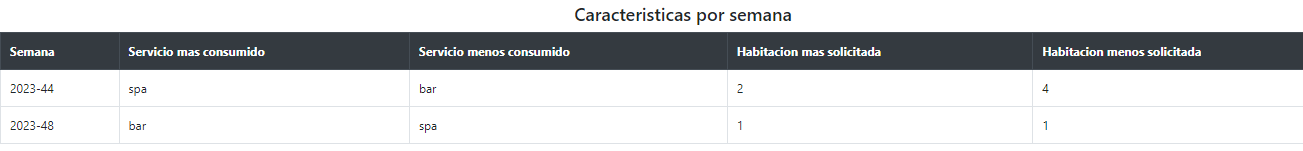
Rango de fechas: 01/01/2023, 30/11/2023. Tipo de servicio = ‘spa’

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

RFC11

Revisa las semanas en las que hubo entradas en reservas



RFC12

Una captura de pantalla de una red social

Descripción generada automáticamente

* Datos

Se generaron los datos mediante la creación de un código en Python que selecciona valores aleatorios, también se hizo el uso de páginas web como generadores de sentencias de SQL para datos un poco más fáciles de crear, todo con el fin de generar el volumen de datos suficiente.