



Base de datos con SQL SERVER

Proyecto Final

AUTOR(Autores):

- Huaman Santana Karla Patricia
- Lalupu Imán José Junior
- Medina Montoya Jhordan Brayan
- Tapara Quispe Fabiola Grissel
- Vásquez Loro Gerardo Fernando

Docente:

Ing. Kevin Rivera Vergaray

Lima, Perú

2023

- I. Iniciación: Definición de los objetivos y el alcance inicial de la base de datos.

Título del Problema: Gestión de líneas telefónicas

Descripción del Problema:

La dificultad que nos concierne radica en la necesidad de administrar de manera efectiva los datos asociados a las líneas telefónicas en un contexto que abarca a clientes, planes de servicio, recargas y las facturas generadas. La implementación de esta base de datos se vuelve esencial para lograr una gestión eficiente que permita rastrear a los clientes con una organización adecuada de la información.

Tablas y Campos Relevantes:

- **Clientes:**

Nombre
DNI
Dirección
Número de Teléfono
Correo Electrónico

- **Líneas Telefónicas:**

Número de Teléfono
Tipo de Línea
Estado
Fecha de Activación/Desactivación

- **Planes:**

Nombre del Plan
Tarifas
Límites de Datos
Minutos Incluidos

- **Factura:**

Monto de factura
Fecha de emisión
Fecha de pago
Fecha de vencimiento

- **Recargas:**

Monto de Recarga

Fecha y Hora de Recarga

Método de Pago

Desafíos:

- Actualización de Planes: Se necesita eventualmente actualizar las descripciones de los beneficios a los planes de las líneas telefónicas.
- Gestión de las líneas de teléfono: Se necesita tener el detalle de las líneas por cliente para verificar la continuidad del servicio.
- Seguimiento de las facturaciones: Es relevante tener un monitoreo detallado de las facturaciones para poder verificar los montos y fechas tanto de emisión o vencimiento de estas mismas.
- Registro de Recargas: Se necesita llevar un registro de las recargas de cada línea para ofrecer planes a clientes con recargas recurrentes.

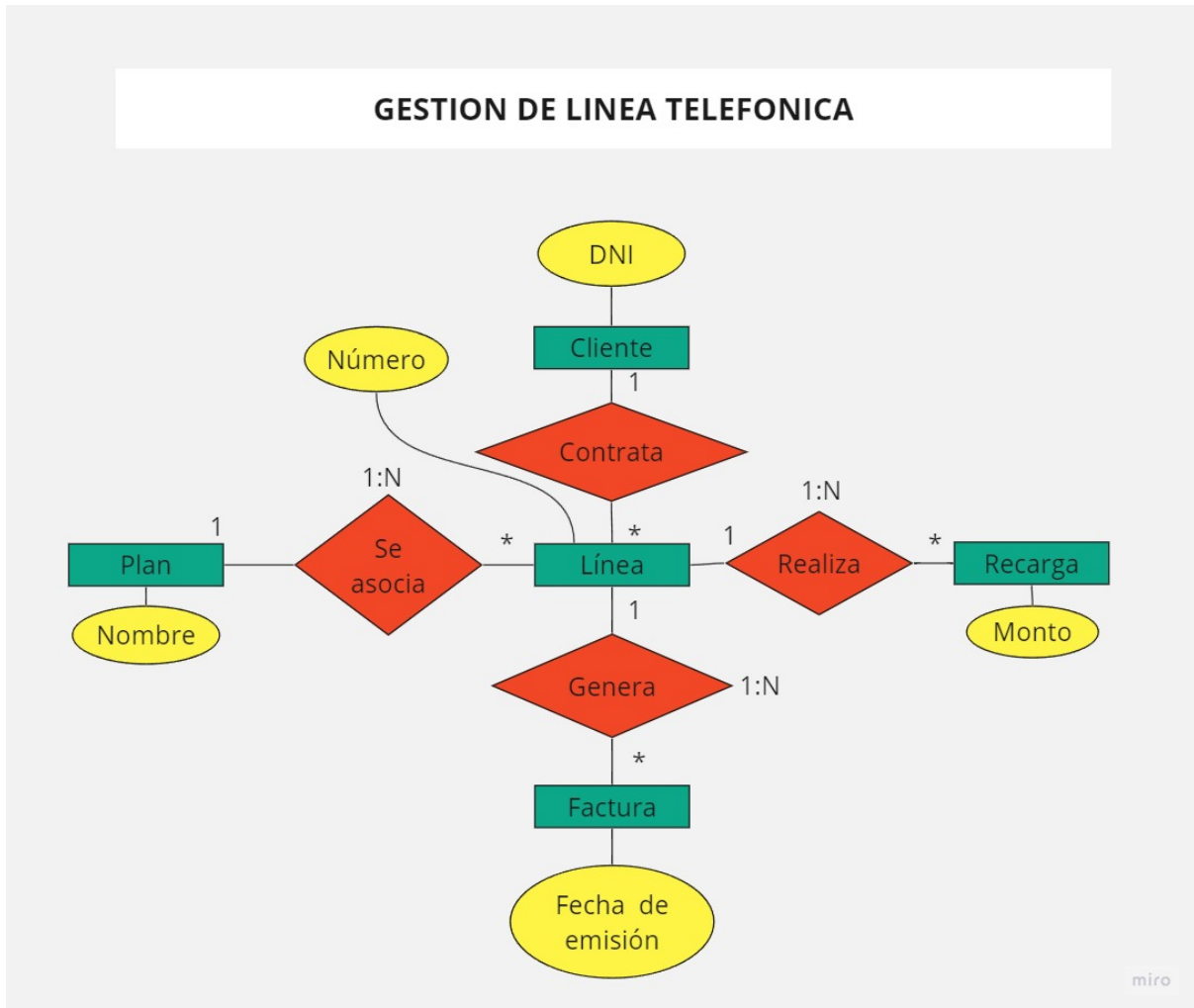
Objetivos:

- Optimizar la actualización de los beneficios y descripción de los planes de las líneas telefónicas.
- Obtener información del cliente para poder ofrecer diferentes servicios. Así como permitir una supervisión efectiva del estado de la línea.
- Garantizar un seguimiento preciso de las facturaciones de los clientes y el cumplimiento de pagos.
- Obtener la información de consumo de los clientes prepago.

Beneficios Esperados:

- Ahorro de Tiempo: La actualización de los planes se haría con facilidad y en poco tiempo.
- Mejora en la precisión: Registro preciso del estado de las líneas telefónicas.
- Reducción de Pérdidas por pagos no realizados: El seguimiento ayuda a identificar los pagos pendientes y poder tomar medidas antes que se acumulen a deudas significativas asegurando la rentabilidad.
- Incrementar la relación de clientes afiliados a un plan: Ofrecer al cliente migrar de línea prepago a una línea postpago.

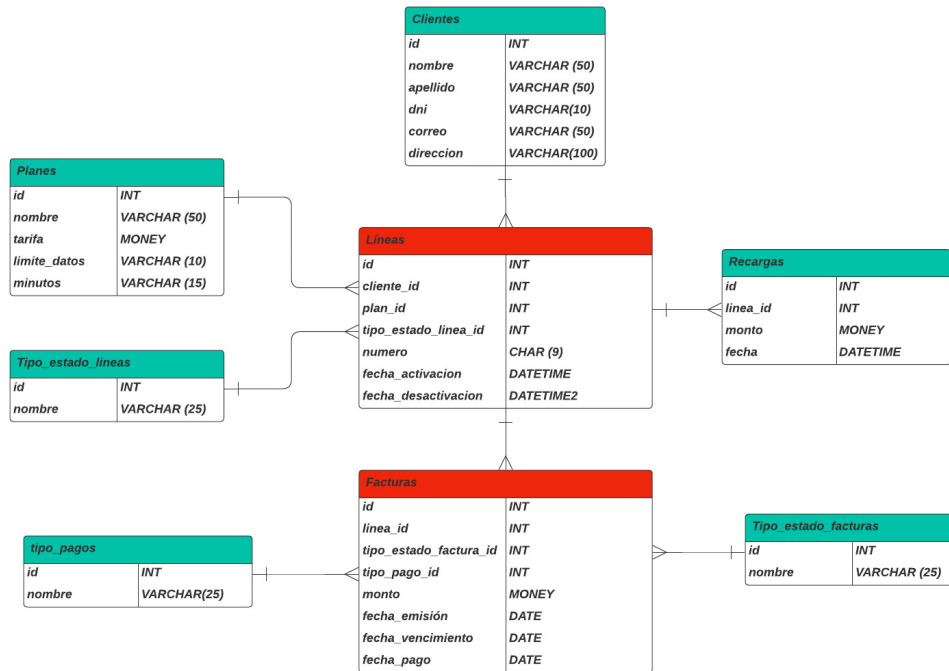
II. Análisis: Diseño Conceptual de la base de datos.



III. Diseño: Definición del Modelo Lógico de la Base de Datos.

SISTEMA DE GESTIÓN DE LINEAS DE TELÉFONO

SQL SERVER - GRUPO 7



- IV. Construcción: Desarrollo de las sentencias SQL que permiten la construcción de las tablas y demás estructuras de la Base de Datos en SQL SERVER.

```
-- Creación de la base de datos

CREATE DATABASE BD_LINEAS_TELEFONICAS

USE BD_LINEAS_TELEFONICAS

--Creación de la tabla Planes

CREATE TABLE PLANES(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    tarifa MONEY NOT NULL,
    limite_datos VARCHAR(10) NOT NULL,
    minutos VARCHAR(15) NOT NULL
)

-- Creación de la tabla Clientes

CREATE TABLE CLIENTES(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    dni VARCHAR(10) UNIQUE NOT NULL,
    correo VARCHAR(50) UNIQUE NOT NULL,
    direccion VARCHAR(100) NOT NULL
)

-- Creación de la tabla Tipo_Pagos

CREATE TABLE TIPO_PAGOS(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    nombre VARCHAR(25) NOT NULL
)
```

```

-- Creación de la tabla Tipo_Estado_Lineas

CREATE TABLE TIPO_ESTADO_LINEAS(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    nombre VARCHAR(25) NOT NULL
)

-- Creación de la tabla Tipo_Estado_Facturas

CREATE TABLE TIPO_ESTADO_FACTURAS(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    nombre VARCHAR(15) NOT NULL
)

-- Creación de la tabla Lineas

CREATE TABLE LINEAS(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    cliente_id INT NOT NULL,
    plan_id INT NOT NULL,
    tipo_estado_linea_id INT NOT NULL,
    numero CHAR(9) UNIQUE NOT NULL,
    fecha_activacion DATETIME NOT NULL,
    fecha_desactivacion DATETIME2 NULL,
    CONSTRAINT fk_lineas_cliente FOREIGN KEY (cliente_id) REFERENCES clientes(id),
    CONSTRAINT fk_lineas_plan FOREIGN KEY (plan_id) REFERENCES planes(id),
    CONSTRAINT fk_lineas_estado FOREIGN KEY (tipo_estado_linea_id) REFERENCES tipo_estado_lineas(id)
)

-- Creación de la tabla Recargas

CREATE TABLE RECARGAS(
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    linea_id INT NOT NULL,
    monto MONEY NOT NULL,
    fecha DATETIME NOT NULL,
    CONSTRAINT fk_recarga_linea FOREIGN KEY (linea_id) REFERENCES lineas(id),
)

-- Creación de la tabla Facturas

CREATE TABLE FACTURAS (
    id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    linea_id INT NOT NULL,
    tipo_estado_factura_id INT NOT NULL,
    tipo_pago_id INT NULL,
    monto MONEY NULL,
    fecha_emision DATETIME NOT NULL,
    fecha_vencimiento DATETIME NOT NULL,
    fecha_pago DATETIME NULL,
    CONSTRAINT fk_factura_linea FOREIGN KEY (linea_id) REFERENCES lineas(id),
    CONSTRAINT fk_tipo_estado_factura FOREIGN KEY (tipo_estado_factura_id) REFERENCES tipo_estado_facturas(id),
    CONSTRAINT fk_tipo_pago FOREIGN KEY (tipo_pago_id) REFERENCES tipo_pagos(id)
);

```

```

CREATE OR ALTER PROCEDURE pa_montoFacturas
    @linea_id INT,
    @tipo_estado INT,
    @tipo_pago INT,
    @monto MONEY,
    @fecha_emision DATETIME,
    @fecha_vencimiento DATETIME,
    @fecha_pago DATETIME
AS
BEGIN
    DECLARE @montoPlan MONEY;
    DECLARE @totalMonto MONEY;

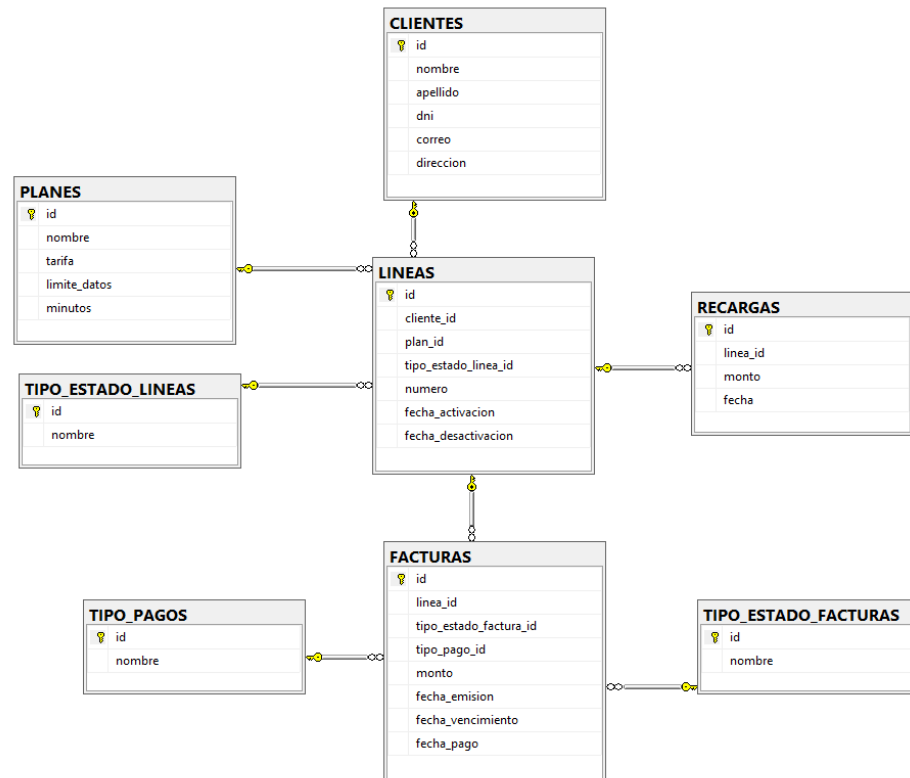
    -- Obtener el monto del plan para la línea específica
    SELECT @montoPlan = p.tarifa
    FROM planes p
    INNER JOIN lineas l ON l.plan_id = p.id
    WHERE l.id = @linea_id;

    -- Calcular el total del monto de la factura sumando el monto del plan y las recargas del mes de la fecha de emisión
    SET @totalMonto = @montoPlan + ISNULL((
        SELECT SUM(r.monto)
        FROM recargas r
        WHERE r.linea_id = @linea_id
        AND MONTH(r.fecha) = MONTH(@fecha_emision)
        AND YEAR(r.fecha) = YEAR(@fecha_emision)
    ), 0);

    INSERT INTO FACTURAS(linea_id, tipo_estado_factura_id, tipo_pago_id, monto, fecha_emision, fecha_vencimiento, fecha_pago)
    VALUES(@linea_id, @tipo_estado, @tipo_pago, @totalMonto, @fecha_emision, @fecha_vencimiento, @fecha_pago)
END;

```


- V. Producto: Diseño Físico, Script de la Base de datos con inserción de datos (Mínimo 5 registros por tablas independientes o no transaccionales y mínimo 15 registros por cada tabla transaccional).



Link del SCRIPT:

https://colab.research.google.com/drive/1JALX_PO-5p1-XAvUQPynL2e2sbZ5HlwI?usp=sharing#scrollTo=0wjspgCbBjvo

VI. ANEXOS

---MUESTRA TODAS LAS FACTURAS DEL MES DE ABRIL EN EL AÑO 2023---JUNIOR

```
SELECT c.nombre AS 'CLIENTE', p.nombre AS 'NOMBRE_PLAN', l.numero AS 'NUMERO', f.monto AS 'MONTO', f.fecha_emision AS 'FECHA_EMISION'
FROM CLIENTES AS c
INNER JOIN LINEAS AS l ON l.cliente_id=c.id
INNER JOIN PLANES AS p ON p.id=l.plan_id
INNER JOIN FACTURAS AS f ON f.linea_id=l.id
WHERE MONTH(f.fecha_emision) = 4 AND YEAR(f.fecha_emision) = 2023
```

	CLIENTE	NOMBRE_PLAN	NUMERO	MONTO	FECHA_EMISION
1	María	Plan Max 29.90	987654382	29.90	2023-04-02 00:00:00.000
2	Luis	Plan Max 39.90	987654383	39.90	2023-04-03 00:00:00.000
3	Ana	Plan Max 49.90	987654384	49.90	2023-04-04 00:00:00.000
4	Pedro	Plan Max 59.90	987654385	59.90	2023-04-05 00:00:00.000
5	Sofía	Plan Max Ilimitado 69.90	987654386	69.90	2023-04-06 00:00:00.000
6	Jorge	Plan Max Ilimitado 79.90	987654387	79.90	2023-04-07 00:00:00.000
7	Laura	Plan Max Ilimitado 89.90	987654388	89.90	2023-04-08 00:00:00.000
8	Martín	Plan Max Ilimitado 99.90	987654389	99.90	2023-04-09 00:00:00.000
9	Isabel	Plan Max Ilimitado 109.90	987654390	109.90	2023-04-10 00:00:00.000

---MUESTRE LAS LÍNEAS DE LOS CLIENTES QUE HAN SIDO BLOQUEADAS EN EL MES DE FEBRERO---JUNIOR

```
SELECT c.nombre AS 'CLIENTE', l.numero AS 'NUMERO', l.fecha_desactivacion AS 'FECHA_DESACTIVACION', tl.nombre AS 'TIPO_ESTADO'
FROM CLIENTES AS c
INNER JOIN LINEAS AS l ON l.cliente_id=c.id
INNER JOIN TIPO_ESTADO_LINEAS AS tl ON l.tipo_estado_linea_id=tl.id
WHERE tl.nombre='Bloqueado' AND MONTH(l.fecha_desactivacion) = 2
```

	CLIENTE	NUMERO	FECHA_DESACTIVACION	TIPO_ESTADO
1	Elena	987654336	2023-02-16 00:00:00.00000000	Bloqueado
2	Ricardo	987654337	2023-02-17 00:00:00.00000000	Bloqueado
3	Silvia	987654338	2023-02-18 00:00:00.00000000	Bloqueado
4	Pablo	987654339	2023-02-19 00:00:00.00000000	Bloqueado
5	Adriana	987654340	2023-02-20 00:00:00.00000000	Bloqueado

---CALCULA EL PROMEDIO MENSUAL DE RECARGAS POR CLIENTE--- JHORDAN

```
SELECT
  c.id AS cliente_id,
  c.nombre + ' ' + c.apellido AS cliente,
  AVG(r.monto) AS promedio_recargas
FROM CLIENTES c
INNER JOIN LINEAS l ON c.id = l.cliente_id
INNER JOIN RECARGAS r ON l.id = r.linea_id
GROUP BY c.id, c.nombre, c.apellido;
```

	cliente_id	cliente	promedio_recargas
1	1	JuanPérez	10.00
2	2	MaríaGonzález	10.00
3	3	LuisRamírez	21.6666
4	4	AnaMartínez	20.00
5	5	PedroTorres	16.6666
6	6	SofíaHerrera	25.00
7	7	JorgeOrtega	30.00
8	8	LauraSánchez	15.00
9	9	MartínFlores	25.00
10	10	IsabelDíaz	15.00

---ENCENTRAS LOS CLIENTES QUE HAN TENIDO RECARGAS SUPERIORES AL PROMEDIO DE RECARGAS---JHORDAN

```

SELECT
    c.nombre + ' ' + c.apellido AS cliente,
    r.monto AS monto_recargas,
    r.fecha AS fecha_recarga
FROM LINEAS l
INNER JOIN RECARGAS r ON l.id = r.linea_id
INNER JOIN CLIENTES c ON l.cliente_id = c.id
WHERE r.monto < (
    SELECT AVG(monto) FROM RECARGAS
)
ORDER BY c.nombre, c.apellido DESC;

```

	cliente	monto_recargas	fecha_recarga
1	AdrianaMoreno	15.00	2024-02-10 14:30:00.000
2	AlbertoFernández	10.00	2023-04-20 06:30:00.000
3	AnaMartínez	15.00	2023-02-09 10:45:00.000
4	ElenaMendoza	15.00	2024-01-25 10:30:00.000
5	IsabelDíaz	15.00	2023-02-05 06:30:00.000
6	JuanPérez	15.00	2023-05-01 08:30:00.000
7	JuanPérez	10.00	2023-02-06 08:30:00.000
8	JuanPérez	5.00	2023-05-06 21:45:00.000
9	LauraSánchez	15.00	2023-01-12 18:30:00.000
10	MaríaGonzález	10.00	2023-03-02 11:30:00.000

---CONSULTA PARA OBTENER EL PROMEDIO MENSUAL DE MONTOS DE FACTURAS POR CLIENTE---FABIOLA

```

SELECT
    c.nombre AS cliente_nombre,
    c.apellido AS cliente_apellido,
    AVG(f.monto) AS promedio_montos_mensuales
FROM CLIENTES c
JOIN LINEAS l ON c.id = l.cliente_id
JOIN FACTURAS f ON l.id = f.linea_id
GROUP BY
    c.nombre, c.apellido;

```

	cliente_nombre	cliente_apellido	promedio_montos_mensuales
1	Rosa	Castañeda	29.90
2	Ricardo	Chávez	79.90
3	Isabel	Díaz	109.90
4	Alberto	Fernández	59.90
5	Martín	Flores	108.2333
6	Silvia	Gómez	89.90
7	María	González	29.90
8	Sofía	Herrera	78.2333
9	Paola	Jiménez	49.90
10	Ana	Martínez	49.90

---CONSULTA PARA OBTENER CLIENTES CON RECARGAS SUPERIORES A \$10 EN EL MES DE MAYO---FABIOLA

```

SELECT
    c.nombre AS cliente_nombre,
    c.correo AS cliente_correo,
    SUM(r.monto) AS monto_total_recargas
FROM CLIENTES c
JOIN LINEAS l ON c.id = l.cliente_id
JOIN RECARGAS r ON l.id = r.linea_id
WHERE MONTH(r.fecha) = 5
GROUP BY c.nombre, c.correo
HAVING
    SUM(r.monto) > 10;

```

70 %

Results Messages

	cliente_nombre	cliente_correo	monto_total_recargas
1	Alberto	alberto.fernandez@email.com	25.00
2	Ana	ana.martinez@email.com	20.00
3	Juan	juan.perez@email.com	20.00
4	Luis	luis.ramirez@email.com	25.00
5	María	maria.gonzalez@email.com	15.00

---PROCEDIMIENTO PARA OBTENER INFORMACIÓN DE UN CLIENTE Y SUS LINEAS ACTIVAS---KARLA

```

CREATE PROCEDURE ObtenerInfoCliente
    @cliente_id INT
AS
BEGIN
    SELECT c.nombre, c.apellido, c.dni, l.numero, p.nombre as 'plan'
    FROM CLIENTES c
    JOIN LINEAS l ON c.id = l.cliente_id
    JOIN PLANES p ON l.plan_id = p.id
    WHERE c.id = @cliente_id AND l.tipo_estado_linea_id = 1; -- Línea activa
END;
EXECUTE ObtenerInfoCliente 5;

```

70 %

Results Messages

	nombre	apellido	dni	numero	plan
1	Pedro	Torres	45678901	987654325	Plan Max 59,90
2	Pedro	Torres	45678901	987654365	Plan Max 59,90
3	Pedro	Torres	45678901	987654385	Plan Max 59,90

---OBTENER LA CANTIDAD TOTAL DE RECARGAS REALIZADAS POR CADA CLIENTE---
KARLA

```
SELECT
    c.id as cliente_id,
    c.nombre,
    c.apellido,
    COUNT(r.id) as total_recargas
FROM CLIENTES c
LEFT JOIN LINEAS l ON c.id = l.cliente_id
LEFT JOIN RECARGAS r ON l.id = r.linea_id
GROUP BY c.id, c.nombre, c.apellido
ORDER BY total_recargas DESC;
```

70 %

Results Messages

	cliente_id	nombre	apellido	total_recargas
1	1	Juan	Pérez	3
2	2	María	González	3
3	3	Luis	Ramírez	3
4	4	Ana	Martínez	3
5	5	Pedro	Torres	3
6	11	Carlos	Luna	2
7	12	Rosa	Castañeda	2
8	13	Miguel	Vargas	2
9	14	Paola	Jiménez	2
10	15	Alberto	Fernández	2

---MOSTRAR TODOS LOS CLIENTES JUNTO CON LA CANTIDAD DE LÍNEAS ACTIVAS QUE TIENEN---GERARDO

```
SELECT
    c.id AS cliente_id,
    c.nombre AS nombre_cliente,
    c.apellido AS apellido_cliente,
    COUNT(l.id) AS cantidad_lineas_activas
FROM CLIENTES c
LEFT JOIN LINEAS l ON c.id = l.cliente_id AND l.tipo_estado_linea_id = 1
GROUP BY
    c.id, c.nombre, c.apellido;
```

70 %

Results Messages

	cliente_id	nombre_cliente	apellido_cliente	cantidad_lineas_activas
1	1	Juan	Pérez	3
2	2	María	González	3
3	3	Luis	Ramírez	3
4	4	Ana	Martínez	3
5	5	Pedro	Torres	3
6	6	Sofía	Herrera	0
7	7	Jorge	Ortega	0
8	8	Laura	Sánchez	0
9	9	Martín	Flores	0
10	10	Isabel	Díaz	0

---MOSTRAR TODAS LAS LÍNEAS ACTIVAS JUNTO CON EL NOMBRE DEL CLIENTE, EL PLAN ASOCIADO Y LA FECHA DE ACTIVACIÓN---GERARDO

```

SELECT
    l.id AS linea_id,
    c.nombre AS nombre_cliente,
    p.nombre AS nombre_plan,
    l.numero AS numero_linea,
    l.fecha_activacion AS fecha_activacion
FROM LINEAS l
INNER JOIN CLIENTES c ON l.cliente_id = c.id
INNER JOIN PLANES p ON l.plan_id = p.id
WHERE l.tipo_estado_linea_id = 1;

```

70 %

Results Messages

	linea_id	nombre_cliente	nombre_plan	numero_linea	fecha_activacion
1	1	Juan	Plan PREPAGO	987654321	2023-01-01 00:00:00.000
2	2	María	Plan Max 29.90	987654322	2023-01-02 00:00:00.000
3	3	Luis	Plan Max 39.90	987654323	2023-01-03 00:00:00.000
4	4	Ana	Plan Max 49.90	987654324	2023-01-04 00:00:00.000
5	5	Pedro	Plan Max 59.90	987654325	2023-01-05 00:00:00.000
6	21	Juan	Plan PREPAGO	987654361	2023-03-12 00:00:00.000
7	22	María	Plan Max 29.90	987654362	2023-03-13 00:00:00.000
8	23	Luis	Plan Max 39.90	987654363	2023-03-14 00:00:00.000
9	24	Ana	Plan Max 49.90	987654364	2023-03-15 00:00:00.000
10	25	Pedro	Plan Max 59.90	987654365	2023-03-16 00:00:00.000

---CALCULAR EL MONTO TOTAL DE FACTURACIÓN POR MES---KARLA

```

SELECT
    MONTH(fecha_emision) AS mes,
    YEAR(fecha_emision) AS año,
    SUM(monto) AS total_ingresos
FROM FACTURAS
GROUP BY MONTH(fecha_emision), YEAR(fecha_emision);

```

70 %

Results Messages

	mes	año	total_ingresos
1	1	2023	1353.20
2	3	2023	1258.20
3	4	2023	629.10