



Universidad Metropolitana de Honduras

Diseño de base de datos 2

Docente: Allan Lopez

Entregable N2 proyecto

ALUMNOS:

Andres Alessandro Viera Espinal

<https://github.com/Ales2704/-UMH-BD2-2025P1-202301761/tree/main/Proyecto%20Umaps>

Carlos Roberto Diaz

<https://github.com/dzroberto72/UMH-BD2-2025P1-202401986/tree/main/Proyecto%20Umaps>

Kilvet Alonso Barahona Martinez

<https://github.com/kilvet/UMH-BD2-2025P1-202100176/tree/main/Proyecto%20Umaps>

Dagoberto Fernandez Valenzuela

<https://github.com/dago20201/UMH-BD2-2025P1-202202541/tree/main/Proyecto%20Umaps>

1-Creación de tablas

```
CREATE DATABASE sistema_facturacion_agua;  
USE sistema_facturacion_agua;
```

```
CREATE TABLE usuarios (  
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    direccion TEXT NOT NULL,  
    categoria ENUM('Residencial', 'Comercial', 'Industrial') NOT NULL,  
    fecha_registro DATE NOT NULL  
);
```

```
CREATE TABLE lecturas (  
    id_lectura INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario INT NOT NULL,  
    lectura_anterior INT NOT NULL,  
    lectura_actual INT NOT NULL,  
    consumo_m3 INT GENERATED ALWAYS AS (lectura_actual - lectura_anterior) STORED,  
    fecha_lectura DATE NOT NULL,  
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```

```
CREATE TABLE facturas (  
    id_factura INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario INT NOT NULL,  
    fecha_emision DATE NOT NULL,  
    fecha_vencimiento DATE NOT NULL,  
    total_a_pagar DECIMAL(10,2) NOT NULL,  
    saldo_pendiente DECIMAL(10,2) NOT NULL DEFAULT 0,  
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```

```
CREATE TABLE conceptos (  
    id_concepto INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL UNIQUE,  
    descripcion TEXT  
);
```

```
CREATE TABLE recibos_de_pago (  
    id_pago INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario INT NOT NULL,  
    id_factura INT NOT NULL,  
    monto_pagado DECIMAL(10,2) NOT NULL,  
    fecha_pago DATE NOT NULL,  
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),  
    FOREIGN KEY (id_factura) REFERENCES facturas(id_factura)  
);
```

```
CREATE TABLE detalle_factura (  
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,  
    id_factura INT NOT NULL,  
    id_concepto INT NOT NULL,  
    monto DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (id_factura) REFERENCES facturas(id_factura) ON DELETE CASCADE,  
    FOREIGN KEY (id_concepto) REFERENCES conceptos(id_concepto) ON DELETE CASCADE  
);
```

2-Inserción de Datos

USE sistema_facturacion_agua;

```
INSERT INTO usuarios (nombre, direccion, categoria, fecha_registro) VALUES
('Juan Pérez', 'Av. Principal #123, Ciudad', 'Residencial', '2024-01-10'),
('María López', 'Calle Secundaria #456, Ciudad', 'Comercial', '2023-11-25'),
('Carlos Ramírez', 'Colonia Centro #789, Ciudad', 'Residencial', '2022-07-05');
```

```
INSERT INTO lecturas (id_usuario, lectura_anterior, lectura_actual, fecha_lectura) VALUES
(1, 100, 120, '2024-03-01'), -- Consumo: 20 m3
(2, 250, 270, '2024-03-01'), -- Consumo: 20 m3
(3, 500, 530, '2024-03-01'); -- Consumo: 30 m3
```

```
INSERT INTO facturas (id_usuario, fecha_emision, fecha_vencimiento, total_a_pagar, saldo_pendiente) VALUES
(1, '2024-03-05', '2024-03-20', 300.50, 0.00),
(2, '2024-03-05', '2024-03-20', 450.75, 50.00), -- Saldo pendiente de meses anteriores
(3, '2024-03-05', '2024-03-20', 600.25, 20.00);
```

```
INSERT INTO conceptos (nombre, descripcion) VALUES
('Agua potable', 'Cobro por el consumo de agua potable'),
('Alcantarillado sanitario', 'Servicio de alcantarillado'),
('Mantenimiento medidor', 'Costo de mantenimiento del medidor'),
('Intereses moratorios', 'Intereses por mora en el pago'),
('Costo fijo', 'Tarifa fija mensual'),
('(-) Desc. adulto mayor', 'Descuento para adultos mayores'),
('Acta de compromiso', 'Pago de acuerdo a acta de compromiso'),
('Otros', 'Otros cobros adicionales'),
('Consumo del mes', 'Cargo por el consumo del mes'),
('Saldo Pendiente', 'Monto no pagado de periodos anteriores')
ON DUPLICATE KEY UPDATE descripcion = VALUES(descripcion);
```

```
INSERT INTO detalle_factura (id_factura, id_concepto, monto) VALUES
(1, 1, 150.00), -- Agua potable
(1, 2, 50.00), -- Alcantarillado sanitario
(1, 5, 100.50), -- Costo fijo
```

```
(2, 1, 200.00), -- Agua potable
(2, 2, 75.00), -- Alcantarillado sanitario
(2, 4, 25.75), -- Intereses moratorios
(2, 5, 100.00), -- Costo fijo
```

```
(3, 1, 300.00), -- Agua potable
(3, 2, 100.00), -- Alcantarillado sanitario
(3, 5, 150.25); -- Costo fijo
```

```
INSERT INTO recibos_de_pago (id_usuario, id_factura, monto_pagado, fecha_pago) VALUES
(1, 1, 300.50, '2024-03-10'), -- Pago total
(2, 2, 400.00, '2024-03-15'), -- Pago parcial
(3, 3, 600.25, '2024-03-18'); -- Pago total
```

```
select * from sistema_facturacion_agua.usuarios;
```

```
select * from sistema_facturacion_agua.lecturas;
```

```
select * from sistema_facturacion_agua.detalle_factura;
```

```
select * from sistema_facturacion_agua.facturas;
```

```
select * from sistema_facturacion_agua.conceptos;
```

3-Procedimientos

DELIMITER \$\$

DROP PROCEDURE IF EXISTS ResumenAnual;

CREATE PROCEDURE ResumenAnual(

IN p_anio INT

)

BEGIN

SELECT

u.id_usuario,

u.nombre AS Nombre,

p_anio AS Anio,

COALESCE(SUM(l.consumo_m3), 0) AS Consumo_Total_m3,

COALESCE(SUM(f.total_a_pagar), 0) AS Total_Facturado,

COALESCE(SUM(r.monto_pagado), 0) AS Total_Pagado

FROM usuarios u

LEFT JOIN lecturas l ON u.id_usuario = l.id_usuario AND YEAR(l.fecha_lectura) = p_anio

LEFT JOIN facturas f ON u.id_usuario = f.id_usuario AND YEAR(f.fecha_emision) = p_anio

LEFT JOIN recibos_de_pago r ON u.id_usuario = r.id_usuario AND YEAR(r.fecha_pago) = p_anio

GROUP BY u.id_usuario, u.nombre

ORDER BY Total_Facturado DESC;

END \$\$

DELIMITER ;

CALL ResumenAnual(2025);

DELIMITER \$\$

DROP PROCEDURE IF EXISTS ListarUsuariosPorCategoriaYAnio;

CREATE PROCEDURE ListarUsuariosPorCategoriaYAnio(

IN p_categoria ENUM('Residencial', 'Comercial', 'Industrial'),

IN p_anio INT

)

BEGIN

SELECT

id_usuario,

nombre,

direccion,

categoria,

YEAR(fecha_registro) AS anio_registro

FROM usuarios

WHERE categoria = p_categoria

AND YEAR(fecha_registro) = p_anio;

END \$\$

DELIMITER ;

CALL ListarUsuariosPorCategoriaYAnio('Residencial', 2024);

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS montosporaños;
```

```
CREATE PROCEDURE montosporaños(  
IN p_anio INT  
)  
BEGIN  
  
SELECT  
u.nombre AS nombre_usuario,  
f.total_a_pagar,  
YEAR(f.fecha_emision) AS anio_factura  
FROM usuarios u  
JOIN facturas f ON u.id_usuario = f.id_usuario  
WHERE YEAR(f.fecha_emision) = p_anio;  
END $$  
  
DELIMITER ;  
  
CALL montosporaños(2024);
```

4-Funciones

DELIMITER \$\$

```
CREATE FUNCTION calcular_total_a_pagar(id_factura INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
DECLARE total DECIMAL(10,2);
DECLARE saldo DECIMAL(10,2);
DECLARE total_pagado DECIMAL(10,2);
```

```
SELECT total_a_pagar INTO total
FROM facturas
WHERE id_factura = id_factura
LIMIT 1;
```

```
SELECT saldo_pendiente INTO saldo
FROM facturas
WHERE id_factura = id_factura
LIMIT 1;
```

```
SELECT SUM(monto_pagado) INTO total_pagado
FROM recibos_de_pago
WHERE id_factura = id_factura;
```

```
RETURN total - total_pagado + saldo;
END $$
```

```
DELIMITER ;
SELECT calcular_total_a_pagar(1);
```

DELIMITER \$\$

```
CREATE FUNCTION obtener_consumo_usuario(id_usuario INT, fecha_inicio DATE, fecha_fin DATE)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE total_consumo INT;
```

```
SELECT SUM(consumo_m3) INTO total_consumo
FROM lecturas
WHERE id_usuario = id_usuario
AND fecha_lectura BETWEEN fecha_inicio AND fecha_fin;
```

```
RETURN total_consumo;
END $$
```

DELIMITER ;

```
SELECT obtener_consumo_usuario(1, '2024-01-01', '2024-03-01');
```



```
CREATE FUNCTION obtener_consumo_usuario(id_usuario INT, fecha_inicio DATE, fecha_fin DATE)
RETURNS INT
DETERMINISTIC
BEGIN
DECLARE total_consumo INT;
```

```
SELECT SUM(consumo_m3) INTO total_consumo
FROM lecturas
WHERE id_usuario = id_usuario
AND fecha_lectura BETWEEN fecha_inicio AND fecha_fin;
```

```
RETURN total_consumo;
END $$
```

```
DELIMITER ;
```

```
SELECT obtener_consumo_usuario(1, '2024-01-01', '2024-03-01'); -- Donde 1 es el ID del usuario
```

```
DELIMITER $$
CREATE FUNCTION obtener_descuento_usuario(id_usuario INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
DECLARE descuento DECIMAL(10,2);
```

```
SELECT SUM(monto) INTO descuento
FROM detalle_factura df
JOIN conceptos c ON df.id_concepto = c.id_concepto
WHERE df.id_factura IN (SELECT id_factura FROM facturas WHERE id_usuario = id_usuario)
AND c.nombre LIKE '%Desc.%';
```

```
IF descuento IS NULL THEN
SET descuento = 0;
END IF;
```

```
RETURN descuento;
END $$
```

```
DELIMITER ;
```

```
SELECT obtener_descuento_usuario(1);
```