



**Universidad
Tecnológico**

Campus
Campus Cúmbres

Materia
Base de Datos

Actividad
Proyecto Final

Maestra
Edwige Jazzmín Novelo Villegas

Alumnos

Juan Porfirio Torres Rojas
#07099471

Jesus David Marroquin Peña
#07101520

Descripción General	3
Tablas y Relaciones	3
Lógica del Negocio	3
Código para creación de tablas	4
Código para insertar datos a las tablas	5
Joins	7
Union	9
Order by	11
Group by	13
Fechas	15
Procedimientos Almacenados	17
Triggers	20
Implementacion JDBC (Java)	22

Descripción General

La base de datos app_medica está diseñada para gestionar un sistema de citas médicas, permitiendo la administración de pacientes, médicos, especialidades y el control de citas médicas.

Tablas y Relaciones

1. PACIENTES

Almacena la información personal de los pacientes del sistema.

Campos principales: id_paciente, nombre, apellidos, fecha_nacimiento, género, contacto

Relaciones: Tiene citas (1:N) y puede tener un usuario (1:1)

2. ESPECIALIDADES

Catálogo de especialidades médicas disponibles.

Campos principales: id_especialidad, nombre, descripción

Relaciones: Es asignada a médicos (1:N)

3. MÉDICOS

Registro de profesionales médicos del sistema.

Campos principales: id_medico, nombre, apellidos, contacto, especialidad

Relaciones: Atiende citas (1:N) y tiene una especialidad (N:1)

4. CITAS

Gestiona las citas médicas programadas.

Campos principales: id_cita, paciente, médico, fecha, hora, estado

Relaciones: Conecta pacientes y médicos

5. USUARIOS

Sistema de autenticación para diferentes tipos de usuarios.

Campos principales: id_usuario, credenciales, tipo_usuario, referencias

Relaciones: Puede estar asociado a paciente o médico

Lógica del Negocio

Los pacientes pueden agendar citas con médicos específicos

Cada médico tiene una especialidad definida

Las citas tienen estados: Pendiente, Confirmada, Cancelada

Sistema de usuarios diferenciado por roles (paciente, médico, admin)

Código para creación de tablas

```
CREATE DATABASE app_medica;  
USE app_medica;
```

-- Tabla PACIENTES

```
CREATE TABLE pacientes (  
  id_paciente INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  ap_paterno VARCHAR(50) NOT NULL,  
  ap_materno VARCHAR(50),  
  fecha_nacimiento DATE,  
  genero ENUM('M','F'),  
  telefono VARCHAR(20),  
  email VARCHAR(100) UNIQUE  
);
```

-- Tabla ESPECIALIDADES

```
CREATE TABLE especialidades (  
  id_especialidad INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  descripcion TEXT  
);
```

-- Tabla MÉDICOS

```
CREATE TABLE medicos (  
  id_medico INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  ap_paterno VARCHAR(50) NOT NULL,  
  ap_materno VARCHAR(50),  
  telefono VARCHAR(20),  
  email VARCHAR(100) UNIQUE,  
  id_especialidad INT NOT NULL,  
  FOREIGN KEY (id_especialidad) REFERENCES especialidades(id_especialidad)  
);
```

-- Tabla CITAS

```
CREATE TABLE citas (  
  id_cita INT AUTO_INCREMENT PRIMARY KEY,  
  id_paciente INT NOT NULL,  
  id_medico INT NOT NULL,  
  fecha DATE NOT NULL,  
  hora TIME NOT NULL,  
  estado ENUM('Pendiente','Confirmada','Cancelada') DEFAULT 'Pendiente',  
  FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente),  
  FOREIGN KEY (id_medico) REFERENCES medicos(id_medico)  
);
```

-- Tabla USUARIOS

```
CREATE TABLE usuarios (  
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
  usuario VARCHAR(50) UNIQUE NOT NULL,
```

```

contrasena VARCHAR(100) NOT NULL,
tipo_usuario ENUM('paciente','medico','admin') NOT NULL,
id_paciente INT,
id_medico INT,
FOREIGN KEY (id_paciente) REFERENCES pacientes(id_paciente),
FOREIGN KEY (id_medico) REFERENCES medicos(id_medico)
);

```

Código para insertar datos a las tablas

-- PACIENTES

```

INSERT INTO pacientes (nombre, ap_paterno, ap_materno, fecha_nacimiento, genero, telefono, email) VALUES
('Juan','Pérez','Lopez','1990-05-10','M','555123456','juanperez@email.com'),
('María','García','Martínez','1985-02-20','F','555987654','maria.garcia@email.com'),
('Luis','Hernández','Torres','2000-07-12','M','555654321','luisht@email.com'),
('Ana','Ramírez','Díaz','1995-09-25','F','555321987','ana.rdz@email.com'),
('Pedro','Sánchez','Flores','1988-11-30','M','555111222','pedrosf@email.com'),
('Sofía','López','Ramírez','1993-03-15','F','555666777','sofia.lopez@email.com'),
('Miguel','Torres','Hernández','1980-06-10','M','555777888','miguel.torres@email.com'),
('Camila','Gómez','Ruiz','2002-01-05','F','555888999','camila.gomez@email.com'),
('Javier','Martínez','Castillo','1998-12-20','M','555999000','javier.mtz@email.com'),
('Fernanda','Vargas','Santos','1991-04-18','F','555000111','fernanda.vs@email.com');

```

-- ESPECIALIDADES

```

INSERT INTO especialidades (nombre, descripcion) VALUES
('Cardiología','Especialidad del corazón'),
('Pediatría','Atención médica de niños'),
('Dermatología','Tratamiento de la piel'),
('Neurología','Enfermedades del sistema nervioso'),
('Ginecología','Salud reproductiva femenina'),
('Oftalmología','Tratamiento de enfermedades oculares'),
('Ortopedia','Tratamiento de huesos y articulaciones'),
('Psiquiatría','Atención de trastornos mentales'),
('Endocrinología','Trastornos hormonales y metabólicos'),
('Oncología','Diagnóstico y tratamiento del cáncer');

```

-- MÉDICOS

```

INSERT INTO medicos (nombre, ap_paterno, ap_materno, telefono, email, id_especialidad) VALUES
('Carlos','Ruiz','Fernández','555111111','carlos.ruiz@medico.com',1),
('Laura','Mendoza','Ortega','555222222','laura.mendoza@medico.com',2),
('José','Morales','Castro','555333333','jose.morales@medico.com',3),
('Elena','Soto','Ramirez','555444444','elena.soto@medico.com',4),
('Ricardo','Vega','Campos','555555555','ricardo.vega@medico.com',5),

```

```
('Patricia','Reyes','Moreno','555666111','patricia.reyes@medico.com',6),
('Andrés','Domínguez','Flores','555777222','andres.dominguez@medico.com',7),
('Claudia','Núñez','García','555888333','claudia.nunez@medico.com',8),
('Héctor','Rojas','Martínez','555999444','hector.rojas@medico.com',9),
('Gabriela','Cruz','Fernández','555000555','gabriela.cruz@medico.com',10);
```

-- CITAS

```
INSERT INTO citas (id_paciente, id_medico, fecha, hora, estado) VALUES
(1,1,'2025-08-30','09:00:00','Confirmada'),
(2,2,'2025-08-30','10:30:00','Pendiente'),
(3,3,'2025-08-31','11:00:00','Confirmada'),
(4,4,'2025-09-01','12:00:00','Cancelada'),
(5,5,'2025-09-01','14:30:00','Pendiente'),
(6,6,'2025-09-02','09:15:00','Pendiente'),
(7,7,'2025-09-02','10:45:00','Confirmada'),
(8,8,'2025-09-03','11:30:00','Pendiente'),
(9,9,'2025-09-03','12:45:00','Confirmada'),
(10,10,'2025-09-04','15:00:00','Pendiente');
```

-- USUARIOS

```
INSERT INTO usuarios (usuario, contrasena, tipo_usuario, id_paciente, id_medico) VALUES
('juanp','12345','paciente',1,NULL),
('maria_g','12345','paciente',2,NULL),
('carlosr','12345','medico',NULL,1),
('lauram','12345','medico',NULL,2),
('admin','admin123','admin',NULL,NULL),
('sofia_l','clave123','paciente',6,NULL),
('miguel_t','clave123','paciente',7,NULL),
('camila_g','clave123','paciente',8,NULL),
('javier_m','clave123','paciente',9,NULL),
('fernanda_v','clave123','paciente',10,NULL),
('patricia_r','clave123','medico',NULL,6),
('andres_d','clave123','medico',NULL,7),
('claudia_n','clave123','medico',NULL,8),
('hector_r','clave123','medico',NULL,9),
('gabriela_c','clave123','medico',NULL,10);
```

Joins

-- JOIN para mostrar información completa de citas

```
SELECT
  c.id_cita,
  p.nombre AS paciente_nombre,
  p.ap_paterno AS paciente_apellido,
  m.nombre AS medico_nombre,
  m.ap_paterno AS medico_apellido,
  e.nombre AS especialidad,
  c.fecha,
  c.hora,
  c.estado
FROM citas c
JOIN pacientes p ON c.id_paciente = p.id_paciente
JOIN medicos m ON c.id_medico = m.id_medico
JOIN especialidades e ON m.id_especialidad = e.id_especialidad;
```

id_cita	paciente_nombre	paciente_apellido	medico_nombre	medico_apellido	especialidad	fecha	hora	estado
1	Juan	Pérez	Carlos	Ruiz	Cardiología	2025-08-30	09:00:00	Confirmada
2	María	García	Laura	Mendoza	Pediatría	2025-08-30	10:30:00	Pendiente
3	Luis	Hernández	José	Morales	Dermatología	2025-08-31	11:00:00	Confirmada
4	Ana	Ramírez	Elena	Soto	Neurología	2025-09-01	12:00:00	Cancelada
5	Pedro	Sánchez	Ricardo	Vega	Ginecología	2025-09-01	14:30:00	Pendiente
6	Sofía	López	Patricia	Reyes	Oftalmología	2025-09-02	09:15:00	Pendiente
7	Miguel	Torres	Andrés	Domínguez	Ortopedia	2025-09-02	10:45:00	Confirmada
8	Camila	Gómez	Claudia	Núñez	Psiquiatría	2025-09-03	11:30:00	Pendiente
9	Javier	Martínez	Héctor	Rojas	Endocrinología	2025-09-03	12:45:00	Confirmada
10	Fernanda	Vargas	Gabriela	Cruz	Oncología	2025-09-04	15:00:00	Pendiente

Propósito: Muestra todas las citas con información detallada

Resultado: ID de cita, nombres del paciente y médico, especialidad, fecha, hora y estado

```
-- LEFT JOIN para incluir todos los médicos aunque no tengan citas
SELECT
    m.id_medico,
    m.nombre,
    m.ap_paterno,
    m.ap_materno,
    e.nombre AS especialidad,
    COUNT(c.id_cita) AS total_citas
FROM medicos m
LEFT JOIN especialidades e ON m.id_especialidad = e.id_especialidad
LEFT JOIN citas c ON m.id_medico = c.id_medico
GROUP BY m.id_medico, m.nombre, m.ap_paterno, m.ap_materno, e.nombre;
```

id_medico	nombre	ap_paterno	ap_materno	especialidad	total_citas
1	Carlos	Ruiz	Fernández	Cardiología	1
2	Laura	Mendoza	Ortega	Pediatría	1
3	José	Morales	Castro	Dermatología	1
4	Elena	Soto	Ramirez	Neurología	1
5	Ricardo	Vega	Campos	Ginecología	1
6	Patricia	Reyes	Moreno	Oftalmología	1
7	Andrés	Domínguez	Flores	Ortopedia	1
8	Claudia	Núñez	García	Psiquiatría	1
9	Héctor	Rojas	Martínez	Endocrinología	1
10	Gabriela	Cruz	Fernández	Oncología	1

Propósito: Lista TODOS los médicos, incluso los que no tienen citas

Resultado: Médicos con su contador de citas (0 si no tienen ninguna)

Union

-- UNION para combinar nombres de pacientes y médicos

```
SELECT
    nombre,
    ap_paterno,
    ap_materno,
    'Paciente' AS tipo_persona
FROM pacientes
UNION
SELECT
    nombre,
    ap_paterno,
    ap_materno,
    'Médico' AS tipo_persona
FROM medicos
ORDER BY tipo_persona, ap_paterno, nombre;
```

nombre	ap_paterno	ap_materno	tipo_persona
Gabriela	Cruz	Fernández	Médico
Andrés	Domínguez	Flores	Médico
Laura	Mendoza	Ortega	Médico
José	Morales	Castro	Médico
Claudia	Núñez	García	Médico
Patricia	Reyes	Moreno	Médico
Héctor	Rojas	Martínez	Médico
Carlos	Ruiz	Fernández	Médico
Elena	Soto	Ramirez	Médico
Ricardo	Vega	Campos	Médico
María	García	Martínez	Paciente
Camila	Gómez	Ruiz	Paciente
Luis	Hernández	Torres	Paciente
Sofía	López	Ramírez	Paciente
Javier	Martínez	Castillo	Paciente
Juan	Pérez	Lopez	Paciente
Ana	Ramírez	Díaz	Paciente
Pedro	Sánchez	Flores	Paciente
Miguel	Torres	Hernández	Paciente
Fernanda	Vargas	Santos	Paciente

Propósito: Crear un directorio unificado de todas las personas en el sistema

Resultado: Lista única con nombres completos de pacientes y médicos, identificados por tipo

-- UNION para combinar información de contacto

```
SELECT
    nombre,
    ap_paterno,
    email,
    telefono,
    'Paciente' AS tipo
FROM pacientes
UNION
SELECT
    nombre,
    ap_paterno,
    email,
    telefono,
    'Médico' AS tipo
FROM medicos
ORDER BY tipo, ap_paterno;
```

nombre	ap_paterno	email	telefono	tipo
Gabriela	Cruz	gabriela.cruz@medico.com	555000555	Médico
Andrés	Domínguez	andres.dominguez@medico.com	555777222	Médico
Laura	Mendoza	laura.mendoza@medico.com	555222222	Médico
José	Morales	jose.morales@medico.com	555333333	Médico
Claudia	Núñez	claudia.nunez@medico.com	555888333	Médico
Patricia	Reyes	patricia.reyes@medico.com	555666111	Médico
Héctor	Rojas	hector.rojas@medico.com	555999444	Médico
Carlos	Ruiz	carlos.ruiz@medico.com	555111111	Médico
Elena	Soto	elena.soto@medico.com	555444444	Médico
Ricardo	Vega	ricardo.vega@medico.com	555555555	Médico
María	García	maria.garcia@email.com	555987654	Paciente
Camila	Gómez	camila.gomez@email.com	555888999	Paciente
Luis	Hernández	luisht@email.com	555654321	Paciente
Sofía	López	sofia.lopez@email.com	555666777	Paciente
Javier	Martínez	javier.mtz@email.com	555999000	Paciente
Juan	Pérez	juanperez@email.com	555123456	Paciente
Ana	Ramírez	ana.rdz@email.com	555321987	Paciente
Pedro	Sánchez	pedrosf@email.com	555111222	Paciente
Miguel	Torres	miguel.torres@email.com	555777888	Paciente
Fernanda	Vargas	fernanda.vs@email.com	555000111	Paciente

Propósito: Consolidar información de contacto de pacientes y médicos

Resultado: Directorio telefónico/email único del hospital

Order by

-- ORDER BY para pacientes por fecha de nacimiento (más jóvenes primero)

```
SELECT
  id_paciente,
  nombre,
  ap_paterno,
  ap_materno,
  fecha_nacimiento,
  TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE()) AS edad,
  genero
FROM pacientes
ORDER BY fecha_nacimiento DESC;
```

id_paciente	nombre	ap_paterno	ap_materno	fecha_nacimiento	edad	genero
8	Camila	Gómez	Ruiz	2002-01-05	23	F
3	Luis	Hernández	Torres	2000-07-12	25	M
9	Javier	Martínez	Castillo	1998-12-20	26	M
4	Ana	Ramírez	Díaz	1995-09-25	30	F
6	Sofía	López	Ramírez	1993-03-15	32	F
10	Fernanda	Vargas	Santos	1991-04-18	34	F
1	Juan	Pérez	Lopez	1990-05-10	35	M
5	Pedro	Sánchez	Flores	1988-11-30	36	M
2	María	García	Martínez	1985-02-20	40	F
7	Miguel	Torres	Hernández	1980-06-10	45	M

Propósito: Listar pacientes del más joven al mayor

Resultado: Pacientes ordenados por fecha de nacimiento con edad calculada

- ORDER BY múltiple para citas

```
SELECT
  id_cita,
  (SELECT nombre FROM pacientes WHERE id_paciente = c.id_paciente) AS paciente,
  (SELECT nombre FROM medicos WHERE id_medico = c.id_medico) AS medico,
  fecha,
  hora,
  estado
FROM citas c
ORDER BY
  CASE estado
    WHEN 'Confirmada' THEN 1
    WHEN 'Pendiente' THEN 2
    WHEN 'Cancelada' THEN 3
    ELSE 4
  END,
  fecha ASC,
  hora ASC;
```

id_cita	paciente	medico	fecha	hora	estado
1	Juan	Carlos	2025-08-30	09:00:00	Confirmada
3	Luis	José	2025-08-31	11:00:00	Confirmada
7	Miguel	Andrés	2025-09-02	10:45:00	Confirmada
9	Javier	Héctor	2025-09-03	12:45:00	Confirmada
2	María	Laura	2025-08-30	10:30:00	Pendiente
5	Pedro	Ricardo	2025-09-01	14:30:00	Pendiente
6	Sofía	Patricia	2025-09-02	09:15:00	Pendiente
8	Camila	Claudia	2025-09-03	11:30:00	Pendiente
10	Fernanda	Gabriela	2025-09-04	15:00:00	Pendiente
4	Ana	Elena	2025-09-01	12:00:00	Cancelada

Propósito: Priorizar citas por importancia (confirmadas primero)

Resultado: Lista de citas ordenada por estado, fecha y hora

Group by

-- GROUP BY para contar citas por médico

```
SELECT
  m.id_medico,
  m.nombre,
  m.ap_paterno,
  e.nombre AS especialidad,
  COUNT(c.id_cita) AS total_citas,
  SUM(CASE WHEN c.estado = 'Confirmada' THEN 1 ELSE 0 END) AS citas_confirmadas,
  SUM(CASE WHEN c.estado = 'Pendiente' THEN 1 ELSE 0 END) AS citas_pendientes
FROM medicos m
JOIN especialidades e ON m.id_especialidad = e.id_especialidad
LEFT JOIN citas c ON m.id_medico = c.id_medico
GROUP BY m.id_medico, m.nombre, m.ap_paterno, e.nombre
ORDER BY total_citas DESC;
```

id_medico	nombre	ap_paterno	especialidad	total_citas	citas_confirmadas	citas_pendientes
1	Carlos	Ruiz	Cardiología	1	1	0
2	Laura	Mendoza	Pediatría	1	0	1
3	José	Morales	Dermatología	1	1	0
4	Elena	Soto	Neurología	1	0	0
5	Ricardo	Vega	Ginecología	1	0	1
6	Patricia	Reyes	Oftalmología	1	0	1
7	Andrés	Domínguez	Ortopedia	1	1	0
8	Claudia	Núñez	Psiquiatría	1	0	1
9	Héctor	Rojas	Endocrinología	1	1	0
10	Gabriela	Cruz	Oncología	1	0	1

Propósito: Generar reporte de productividad de cada médico

Resultado: Tabla con total de citas, confirmadas y pendientes por médico

```

-- GROUP BY para análisis demográfico de pacientes
SELECT
    genero,
    CASE
        WHEN TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE()) < 18 THEN 'Menor
de edad'
        WHEN TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE()) BETWEEN 18 AND
30 THEN '18-30 años'
        WHEN TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE()) BETWEEN 31 AND
50 THEN '31-50 años'
        ELSE 'Mayor de 50 años'
    END AS grupo_edad,
    COUNT(*) AS cantidad_pacientes,
    ROUND(AVG(TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE())), 1) AS
edad_promedio
FROM pacientes
GROUP BY genero, grupo_edad
ORDER BY genero,
    CASE grupo_edad
        WHEN 'Menor de edad' THEN 1
        WHEN '18-30 años' THEN 2
        WHEN '31-50 años' THEN 3
        ELSE 4
    END;

```

genero	grupo_edad	cantidad_pacientes	edad_promedio
M	18-30 años	2	25.5
M	31-50 años	3	38.7
F	18-30 años	2	26.5
F	31-50 años	3	35.3

Propósito: Analizar la distribución de pacientes por género y edad

Resultado: Estadísticas demográficas con rangos de edad y promedios

Fechas

```
SELECT
  id_cita,
  fecha,
  hora,
  DAYNAME(fecha) AS dia_semana,
  MONTHNAME(fecha) AS mes,
  DAY(fecha) AS dia_mes,
  YEAR(fecha) AS año,
  CONCAT(
    CASE
      WHEN HOUR(hora) < 12 THEN 'Mañana'
      WHEN HOUR(hora) BETWEEN 12 AND 17 THEN 'Tarde'
      ELSE 'Noche'
    END,
    '(',
    DATE_FORMAT(hora, '%h:%i %p'),
    ')'
  ) AS turno,
  estado
FROM citas
-- WHERE fecha >= CURDATE() -- Comentar esta línea
ORDER BY fecha, hora;
```

id_cita	fecha	hora	dia_semana	mes	dia_mes	año	turno	estado
1	2025-08-30	09:00:00	Saturday	August	30	2025	Mañana (09:00 AM)	Confirmada
2	2025-08-30	10:30:00	Saturday	August	30	2025	Mañana (10:30 AM)	Pendiente
3	2025-08-31	11:00:00	Sunday	August	31	2025	Mañana (11:00 AM)	Confirmada
4	2025-09-01	12:00:00	Monday	September	1	2025	Tarde (12:00 PM)	Cancelada
5	2025-09-01	14:30:00	Monday	September	1	2025	Tarde (02:30 PM)	Pendiente
6	2025-09-02	09:15:00	Tuesday	September	2	2025	Mañana (09:15 AM)	Pendiente
7	2025-09-02	10:45:00	Tuesday	September	2	2025	Mañana (10:45 AM)	Confirmada
8	2025-09-03	11:30:00	Wednesday	September	3	2025	Mañana (11:30 AM)	Pendiente
9	2025-09-03	12:45:00	Wednesday	September	3	2025	Tarde (12:45 PM)	Confirmada
10	2025-09-04	15:00:00	Thursday	September	4	2025	Tarde (03:00 PM)	Pendiente

Propósito: Analizar patrones de citas por día de semana y horarios

Resultado: Citas con día de semana, mes y clasificación por turno (Mañana/Tarde/Noche)

-- Cálculos con fechas para pacientes

```
SELECT
  id_paciente,
  nombre,
  ap_paterno,
  fecha_nacimiento,
  TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE()) AS edad_actual,
  DATE_ADD(fecha_nacimiento, INTERVAL 65 YEAR) AS fecha_jubilacion_65,
  TIMESTAMPDIFF(YEAR, CURDATE(), DATE_ADD(fecha_nacimiento, INTERVAL 65
YEAR)) AS años_para_jubilacion,
  DAYOFYEAR(fecha_nacimiento) AS dia_del_año_nacimiento,
  CONCAT(
    'Nació un ',
    DAYNAME(fecha_nacimiento),
    ' en ',
    MONTHNAME(fecha_nacimiento)
  ) AS descripcion_nacimiento
FROM pacientes
ORDER BY fecha_nacimiento;
```

	id_paciente	nombre	ap_paterno	fecha_nacimiento	edad_actual	fecha_jubilacion_65	años_para_jubilacion	dia_del_año_nacimiento	descripcion_nacimiento
7	Miguel	Torres	1980-06-10	45	2045-06-10	19	162	Nació un Tuesday en June	
2	María	García	1985-02-20	40	2050-02-20	24	51	Nació un Wednesday en February	
5	Pedro	Sánchez	1988-11-30	36	2053-11-30	28	335	Nació un Wednesday en November	
1	Juan	Pérez	1990-05-10	35	2055-05-10	29	130	Nació un Thursday en May	
10	Fernanda	Vargas	1991-04-18	34	2056-04-18	30	108	Nació un Thursday en April	
6	Sofía	López	1993-03-15	32	2058-03-15	32	74	Nació un Monday en March	
4	Ana	Ramírez	1995-09-25	30	2060-09-25	34	268	Nació un Monday en September	
9	Javier	Martínez	1998-12-20	26	2063-12-20	38	354	Nació un Sunday en December	
3	Luis	Hernández	2000-07-12	25	2065-07-12	39	194	Nació un Wednesday en July	
8	Camila	Gómez	2002-01-05	23	2067-01-05	41	5	Nació un Saturday en January	

Propósito: Calcular información relacionada con la edad de pacientes

Resultado: Edad actual, fecha de jubilación a los 65 años y años restantes

Procedimientos Almacenados

-- Procedimiento Almacenado - Reporte de Clientes Vigentes
DELIMITER //

```
CREATE PROCEDURE CalcularIngresosDiarios(IN fecha_consulta DATE)
BEGIN
    SELECT
        c.fecha,
        COUNT(*) as total_consultas,
        SUM(500) as ingresos_totales -- Precio fijo de $500 por consulta
    FROM citas c
    LEFT JOIN medicos m ON c.id_medico = m.id_medico
    WHERE c.fecha = fecha_consulta
    AND c.estado = 'Confirmada'
    GROUP BY c.fecha;
END //
DELIMITER ;
```

30 01:05:19 CREATE PROCEDURE CalcularIngresosDiarios(IN fecha_consulta DATE) BEGIN SELECT c.fecha, COUNT(*) as tot... 0 row(s) affected

-- Ejecutar el procedimiento almacenado

CALL CalcularIngresosDiarios('2025-08-30');

fecha	total_consultas	ingresos_totales
2025-08-30	1	500

Propósito: Calcular ingresos económicos de un día específico

Resultado: Fecha, número de consultas confirmadas e ingresos totales (\$500 por consulta)

-- Manejo de Excepción - Restricción Única de Email
DELIMITER //

```
CREATE PROCEDURE ReportePacientesTrimestre(IN anio INT)
BEGIN
    SELECT
        p.id_paciente,
        CONCAT(p.nombre, ' ', p.ap_paterno, ' ', p.ap_materno) as nombre_completo,
        p.email,
        p.telefono,
        COUNT(c.id_cita) as total_citas
    FROM pacientes p
    INNER JOIN citas c ON p.id_paciente = c.id_paciente
    WHERE c.fecha BETWEEN CONCAT(anio, '-07-01') AND CONCAT(anio, '-09-30')
    AND c.estado != 'Cancelada'
    GROUP BY p.id_paciente, nombre_completo, p.email, p.telefono
    ORDER BY total_citas DESC;
END //
```

DELIMITER ;

43 01:10:33 CREATE PROCEDURE ReportePacientesTrimestre(IN anio INT) BEGIN SELECT p.id_paciente, CONCAT(p.nombre, '...', p.ap_paterno, '...', p.ap_materno) as nombre_completo, p.email, p.telefono, COUNT(c.id_cita) as total_citas FROM pacientes p INNER JOIN citas c ON p.id_paciente = c.id_paciente WHERE c.fecha BETWEEN CONCAT(anio, '-07-01') AND CONCAT(anio, '-09-30') AND c.estado != 'Cancelada' GROUP BY p.id_paciente, nombre_completo, p.email, p.telefono ORDER BY total_citas DESC; 0 row(s) affected

-- Ejecutar el procedimiento almacenado
CALL ReportePacientesTrimestre(2025);

id_paciente	nombre_completo	email	telefono	total_citas
1	Juan Pérez Lopez	juanperez@email.com	555123456	1
2	María García Martínez	maria.garcia@email.com	555987654	1
3	Luis Hernández Torres	luisht@email.com	555654321	1
5	Pedro Sánchez Flores	pedrosf@email.com	555111222	1
6	Sofía López Ramírez	sofia.lopez@email.com	555666777	1
7	Miguel Torres Hernández	miguel.torres@email.com	555777888	1
8	Camila Gómez Ruiz	camila.gomez@email.com	555888999	1
9	Javier Martínez Castillo	javier.mtz@email.com	555999000	1
10	Fernanda Vargas Santos	fernanda.vs@email.com	555000111	1

Propósito: Identificar pacientes más activos en el tercer trimestre

Resultado: Lista de pacientes con sus datos de contacto y número de citas

-- Manejo de Excepción - Restricción Única de Email

```
ALTER TABLE pacientes ADD CONSTRAINT uk_pacientes_email UNIQUE (email);
```

```
DELIMITER //
```

```
CREATE PROCEDURE AgregarPacienteConValidacion(
    IN p_nombre VARCHAR(100),
    IN p_ap_paterno VARCHAR(100),
    IN p_ap_materno VARCHAR(100),
    IN p_fecha_nacimiento DATE,
    IN p_genero CHAR(1),
    IN p_telefono VARCHAR(15),
    IN p_email VARCHAR(150)
)
BEGIN
    DECLARE EXIT HANDLER FOR 1062
    BEGIN
        SELECT 'Error: El email ya está registrado, intente con otro email' as mensaje_error;
    END;

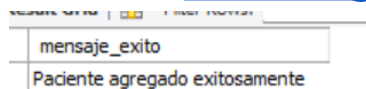
    INSERT INTO pacientes (nombre, ap_paterno, ap_materno, fecha_nacimiento, genero,
telefono, email)
    VALUES (p_nombre, p_ap_paterno, p_ap_materno, p_fecha_nacimiento, p_genero,
p_telefono, p_email);

    SELECT 'Paciente agregado exitosamente' as mensaje_exito;
END //
```

```
DELIMITER ;
```

-- Intento exitoso

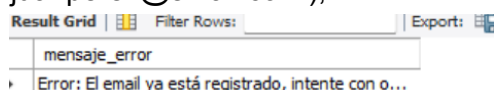
```
CALL AgregarPacienteConValidacion('Roberto', 'Silva', 'Mendez', '1992-08-15', 'M',
'555123999', 'roberto.silva@email.com');
```



mensaje_exito
Paciente agregado exitosamente

-- Intento que generará error (email duplicado)

```
CALL AgregarPacienteConValidacion('Ana', 'Maria', 'Lopez', '1990-01-01', 'F', '555000000',
'juanperez@email.com');
```



mensaje_error
Error: El email ya está registrado, intente con o...

Propósito: Registrar nuevos pacientes con validación automática de email único

Resultado: Mensaje de éxito o error si el email ya existe

Triggers

```
-- Triggers
-- Control de Citas Duplicadas
DELIMITER //

CREATE TRIGGER PrevenirCitasDuplicadas
BEFORE INSERT ON citas
FOR EACH ROW
BEGIN
    DECLARE cuenta_citas INT;

    SELECT COUNT(*) INTO cuenta_citas
    FROM citas
    WHERE id_paciente = NEW.id_paciente
    AND id_medico = NEW.id_medico
    AND fecha = NEW.fecha
    AND hora = NEW.hora;

    IF cuenta_citas > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Ya existe una cita programada para este paciente con
el mismo médico, fecha y hora';
    END IF;
END //

DELIMITER ;
```

Propósito: Evitar que se programen citas duplicadas automáticamente

Resultado: Bloquea inserción y muestra error si ya existe cita igual

DELIMITER //

```
CREATE TRIGGER SeguimientoNuevasCitas
AFTER INSERT ON citas
FOR EACH ROW
BEGIN
    DECLARE nombre_completo VARCHAR(300);

    -- Obtener nombre completo del paciente
    SELECT CONCAT(nombre, ' ', ap_paterno, ' ', ap_materno)
    INTO nombre_completo
    FROM pacientes
    WHERE id_paciente = NEW.id_paciente;

    -- Insertar en tabla de seguimiento
    INSERT INTO seguimiento_citas (id_cita, nombre_paciente, fecha_cita, hora_cita, accion)
    VALUES (NEW.id_cita, nombre_completo, NEW.fecha, NEW.hora, 'Nueva cita
registrada');
END //
```

DELIMITER ;

Propósito: Crear auditoría automática de todas las citas nuevas

Resultado: Registro automático en tabla de seguimiento con timestamp

```
-- Probar el trigger insertando una nueva cita
INSERT INTO citas (id_paciente, id_medico, fecha, hora, estado)
VALUES (1, 1, '2025-09-05', '16:00:00', 'Pendiente');
```

-- Ver el seguimiento

```
SELECT * FROM seguimiento_citas;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:							
	id_seguimiento	id_cita	nombre_paciente	fecha_cita	hora_cita	fecha_registro	accion
▶	1	11	Juan Pérez Lopez	2025-09-05	16:00:00	2025-10-02 01:13:15	Nueva cita registrada
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Si probamo el trigger insertando nuevamente la cita el sistema arroja lo siguiente:

56 01:15:18 INSERT INTO citas (id_paciente, id_medico, fecha, hora, estado) VALUES (1, 1, '2025-09-05', '16:00:00', 'Pendiente') Error Code: 1644. Error: Ya existe una cita programada para este paciente con el mismo médico, fecha y hora

Implementacion JDBC (Java)

1. DatabaseConnection.java

Propósito: Maneja la conexión a la base de datos MySQL

Funcionalidad:

Lee configuración desde config.properties

Establece conexión JDBC con MySQL

Maneja errores de conexión

Proporciona método getConnection() para otras clases

Características principales:

Carga automática del driver MySQL

Configuración externa (no hardcoded)

Manejo de excepciones

Singleton pattern para conexiones

2. Paciente.java

Propósito: Modelo de datos que representa un paciente

Funcionalidad:

Define estructura de un paciente

Getters y setters para todos los campos

Constructor con y sin parámetros

Método toString() para mostrar información

Campos que maneja:

ID del paciente

Nombre completo (nombre, apellido paterno, materno)

Fecha de nacimiento

Género (M/F)

Teléfono y email

3. PacienteDAO.java

Propósito: Data Access Object - Maneja operaciones CRUD en la base de datos

Funcionalidad:

CREATE - Agregar paciente

READ - Leer pacientes

UPDATE - Actualizar paciente

DELETE - Eliminar paciente

Características:

Prepared Statements (previene SQL injection)

Manejo de excepciones

Conversión automática ResultSet → Objeto Paciente

Validación de datos

4. SistemaMedico.java

Propósito: Aplicación principal con interfaz de usuario en consola

Funcionalidad:

Menú Principal:

Operaciones disponibles:

Opción 1 - Registrar paciente:

Solicita datos por consola

Valida información ingresada

Llama a PacienteDAO.agregarPaciente()

```
=== DAR DE ALTA UN PACIENTE ===
Nombre: Juan Torres
Apellido Paterno: Torres
Apellido Materno (opcional): Rojas
Fecha de nacimiento (yyyy-MM-dd): 2004-08-24
Género (M/F): M
Teléfono: 8124185528
Email: porfiriotr@outlook.com
Conexión establecida exitosamente
? Paciente dado de alta exitosamente

Presione Enter para continuar...
```

11	Roberto	Silva	Mendez	1992-08-15	M	555123999	roberto.silva@email.com
13	Juan Tor...	Torres	Rojas	2004-08-24	M	8124185528	porfiriotr@outlook.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Opción 2 - Listar todos:

Obtiene lista completa de pacientes

Muestra información formateada en consola

```
Seleccione una opción: 2

=== CONSULTAR TODOS LOS PACIENTES ===
Conexión establecida exitosamente
Total de pacientes: 12
-----
ID: 1 | Juan Pérez Lopez | Género: M | Teléfono: 555123456 | Email: juanperez@email.com | Nacimiento: 1990-05-10
ID: 2 | María García Martínez | Género: F | Teléfono: 555987654 | Email: maria.garcia@email.com | Nacimiento: 1985-02-20
ID: 3 | Luis Hernández Torres | Género: M | Teléfono: 555654321 | Email: luisht@email.com | Nacimiento: 2000-07-12
ID: 4 | Ana Ramírez Díaz | Género: F | Teléfono: 555321987 | Email: ana.rdz@email.com | Nacimiento: 1995-09-25
ID: 5 | Pedro Sánchez Flores | Género: M | Teléfono: 555111222 | Email: pedrosf@email.com | Nacimiento: 1988-11-30
ID: 6 | Sofía López Ramírez | Género: F | Teléfono: 555666777 | Email: sofia.lopez@email.com | Nacimiento: 1993-03-15
ID: 7 | Miguel Torres Hernández | Género: M | Teléfono: 555777888 | Email: miguel.torres@email.com | Nacimiento: 1980-06-10
ID: 8 | Camila Gómez Ruiz | Género: F | Teléfono: 555888999 | Email: camila.gomez@email.com | Nacimiento: 2002-01-05
ID: 9 | Javier Martínez Castillo | Género: M | Teléfono: 555999000 | Email: javier.mtz@email.com | Nacimiento: 1998-12-20
ID: 10 | Fernanda Vargas Santos | Género: F | Teléfono: 555000111 | Email: fernanda.vs@email.com | Nacimiento: 1991-04-18
ID: 11 | Roberto Silva Mendez | Género: M | Teléfono: 555123999 | Email: roberto.silva@email.com | Nacimiento: 1992-08-15
ID: 13 | Juan Torres Torres Rojas | Género: M | Teléfono: 8124185528 | Email: porfiriotr@outlook.com | Nacimiento: 2004-08-24

Presione Enter para continuar...
```

Opción 3 - Buscar por ID:
Solicita ID del paciente
Muestra información detallada si existe

```

=== BUSCAR PACIENTE POR ID ===
Ingrese ID del paciente: 13
Conexión establecida exitosamente
Paciente encontrado:
ID: 13 | Juan Torres Torres Rojas | Género: M | Teléfono: 8124185528 | Email: porfiriotr@outlook.com | Nacimiento: 2004-08-24

Presione Enter para continuar...

```

Opción 4 - Actualizar:
Busca paciente por ID
Permite modificar cada campo
Guarda cambios en base de datos

```

=== ACTUALIZAR DATOS DE PACIENTE ===
Ingrese ID del paciente a actualizar: 13
Conexión establecida exitosamente
Paciente actual: ID: 13 | Juan Torres Torres Rojas | Género: M | Teléfono: 8124185528 | Email: porfiriotr@outlook.com | Nacimiento: 2004-08-24
Ingrese los nuevos datos (Enter para mantener el valor actual):
Nombre [Juan Torres]: Juan Porfirio
Apellido Paterno [Torres]: Torres
Apellido Materno [Rojas]: Rojas
Fecha de nacimiento [2004-08-24] (yyyy-MM-dd): 2004-08-24
Género [M] (M/F): M
Teléfono [8124185528]: 8124139205
Email [porfiriotr@outlook.com]: juantr@hotmail.com
Conexión establecida exitosamente
? Paciente actualizado exitosamente

Presione Enter para continuar...

```

Antes:

11	Roberto Silva	Mendez	1992-08-15	M	555123999	roberto.silva@email.com
13	Juan Tor...	Torres Rojas	20(1992-08-15	M	8124185528	porfiriotr@outlook.com
NULO	NULO	NULO	NULO	NULO	NULO	NULO

Despues:

[illegible]

Opción 5 - Eliminar:
Solicita ID del paciente
Confirma eliminación
Remueve de la base de datos

```

=== ELIMINAR PACIENTE ===
Ingrese ID del paciente a eliminar: 13
Conexión establecida exitosamente
Paciente a eliminar: ID: 13 | Juan Porfirio Torres Rojas | Género: M | Teléfono: 8124139205 | Email: juantr@hotmail.com | Nacimiento: 2004-08-24
¿Está seguro que desea eliminar este paciente? (s/N): s
Conexión establecida exitosamente
? Paciente eliminado exitosamente

Presione Enter para continuar...

```

[illegible]

FLUJO DE LA APLICACIÓN

SistemaMedico.main() → Inicia la aplicación

DatabaseConnection → Establece conexión con MySQL

Menú interactivo → Usuario selecciona operación

PacienteDAO → Ejecuta operación CRUD correspondiente

Modelo Paciente → Maneja los datos

Resultado → Se muestra al usuario

Liga de Github

<https://github.com/enderkai00-png/Proyecto-base-de-datos-tecmi>