

## Modelo lógico: Modo esquema

### Tablas

- PACIENTE (PacienteID INT, Nombre VARCHAR(100), FechaNacimiento DATE, Telefono VARCHAR(15), Email VARCHAR(100), Direccion VARCHAR(200))
- DOCTOR (DoctorID INT, Nombre VARCHAR(100), Especialidad VARCHAR(50), Telefono VARCHAR(15), Email VARCHAR(100), HorarioConsulta VARCHAR(100))
- CITA (CitaID INT, PacienteID INT, DoctorID INT, FechaCita DATE, HoraCita TIME, Motivo VARCHAR(200), MedicamentosRecetados VARCHAR(500), Estado VARCHAR(20))
- HISTORIAL\_MEDICO (HistorialID INT, PacienteID INT, FechaVisita DATE, DoctorID INT, Diagnostico TEXT, Tratamiento TEXT)

### Relaciones

- CITA.PacienteID → PACIENTE.PacienteID (Foreign Key)
- CITA.DoctorID → DOCTOR.DoctorID (Foreign Key)
- HISTORIAL\_MEDICO.PacienteID → PACIENTE.PacienteID (Foreign Key)
- HISTORIAL\_MEDICO.DoctorID → DOCTOR.DoctorID (Foreign Key)

### Tareas

1. Identificar y corregir el error de normalización
2. Crear las tablas en SQL con las correcciones necesarias
3. Insertar al menos 3 registros en cada tabla
4. Crear una vista VistaCitasProximas que muestre:
  - CitaID, FechaCita, HoraCita, Motivo
  - NombrePaciente, TelefonoPaciente
  - NombreDoctor, Especialidad

- DíasFaltantes (diferencia entre fecha actual y FechaCita)
- Solo citas de los próximos 7 días

5. Crear función CalcularEdadPaciente con:

- Parámetro: p\_PacienteID INT
- Retorno: INT (edad en años)
- Lógica: Calcular edad desde FechaNacimiento hasta fecha actual

## Criterios de evaluación

- Corrección de Normalización (30%): Identificación y solución correcta del error
- Implementación SQL (40%): Sintaxis correcta
- Calidad de Datos (10%): Datos coherentes y completos
- Programación BD (20%): Funcionamiento correcto del procedimiento/vista/función