

Sesión 08
Primer Examen Parcial.

Objetivo

Determinar el nivel de aprendizaje alcanzado por los participantes con respecto a los temas desarrollados entre las sesiones 1 al 7 de la unidad didáctica, mediante el desarrollo de un caso de implementación del diseño de una base de datos, en una sesión práctica.

Requerimientos

- PC o Equivalente con Windows o Linux.
- Navegador web (Firefox, Chrome, Brave, Opera, etc.)
- Gestor de base de datos MySql/MariaDB/PostgreSQL
- Cliente de base de datos

Requisitos

- Haber asistido a las sesiones 3, 4, 5, 6 y 7 de la unidad didáctica.

Sesión 08
Primer Examen Parcial.

Caso Evaluativo: “La Clínica San Vital”

Crónica del Problema

La Clínica San Vital, ubicada en una zona urbana en crecimiento, ha experimentado un aumento significativo en la cantidad de pacientes atendidos diariamente. Hasta hace poco, los registros de atención se realizaban en hojas de cálculo y formularios físicos, lo que generaba múltiples inconvenientes:

- Dificultad para acceder al historial médico de los pacientes.
- Errores en la asignación de citas y duplicación de horarios.
- Imposibilidad de generar reportes sobre diagnósticos frecuentes o médicos más solicitados.
- Falta de trazabilidad sobre tratamientos y medicamentos administrados.

Ante esta situación, la administración decidió implementar un sistema de base de datos que permita gestionar eficientemente la atención médica, los profesionales de salud, y el seguimiento clínico de los pacientes.

Requerimientos del Sistema

Funcionales:

- Registrar pacientes con sus datos personales y contacto.
- Registrar médicos con especialidad y disponibilidad.
- Registrar citas médicas con fecha, hora, médico y paciente.
- Registrar diagnósticos y tratamientos asociados a cada cita.
- Consultar historial clínico por paciente.
- Generar reportes por especialidad, frecuencia de diagnósticos y carga de atención por médico.

Sesión 08
Primer Examen Parcial.

No funcionales:

- Seguridad y confidencialidad de los datos médicos.
- Escalabilidad para incluir más especialidades y sedes.
- Automatización de validaciones mediante triggers.

Diseño Conceptual

Entidades:

- Paciente
- Médico
- Cita
- Diagnóstico
- Tratamiento

(Implemente el diagrama)

Diseño Lógico

Tablas:

- Paciente(id_paciente, nombre, dni, telefono)
- Medico(id_medico, nombre, especialidad)
- Cita(id_cita, id_paciente, id_medico, fecha, hora)
- Diagnostico(id_diagnostico, id_cita, descripcion)
- Tratamiento(id_tratamiento, id_diagnostico, medicamento, dosis)

(Implemente el diagrama)

Sesión 08
Primer Examen Parcial.

Diseño Físico (SQL)

sql

```
CREATE TABLE Paciente (
    id_paciente SERIAL PRIMARY KEY,
    nombre VARCHAR(100),
    dni VARCHAR(10),
    telefono VARCHAR(15)
);
```

```
CREATE TABLE Medico (
    id_medico SERIAL PRIMARY KEY,
    nombre VARCHAR(100),
    especialidad VARCHAR(50)
);
```

```
CREATE TABLE Cita (
    id_cita SERIAL PRIMARY KEY,
    id_paciente INT REFERENCES Paciente(id_paciente),
    id_medico INT REFERENCES Medico(id_medico),
    fecha DATE,
    hora TIME
);
```

```
CREATE TABLE Diagnostico (
    id_diagnostico SERIAL PRIMARY KEY,
    id_cita INT REFERENCES Cita(id_cita),
    descripcion TEXT
);
```

Sesión 08
Primer Examen Parcial.

```
CREATE TABLE Tratamiento (
    id_tratamiento SERIAL PRIMARY KEY,
    id_diagnostico INT REFERENCES Diagnostico(id_diagnostico),
    medicamento VARCHAR(100),
    dosis VARCHAR(50)
);
```

Triggers y Procedimientos

Validar que un médico no tenga dos citas a la misma hora

```
sql
CREATE OR REPLACE FUNCTION validar_cita_medico()
RETURNS TRIGGER AS $$

BEGIN
    IF EXISTS (
        SELECT 1 FROM Cita
        WHERE id_medico = NEW.id_medico
        AND fecha = NEW.fecha
        AND hora = NEW.hora
    ) THEN
        RAISE EXCEPTION 'El médico ya tiene una cita en ese horario.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_validar_cita_medico
BEFORE INSERT ON Cita
FOR EACH ROW
EXECUTE FUNCTION validar_cita_medico();
```

Sesión 08
Primer Examen Parcial.

Actividad de Evaluación

Instrucciones:

A partir del caso “La Clínica San Vital”, responde las siguientes preguntas:

- Diseño Conceptual
 - a. ¿Qué entidad representa el historial clínico de un paciente?
 - b. ¿Qué relación existe entre diagnóstico y tratamiento?
- Diseño Lógico
 - a. ¿Qué claves foráneas están presentes en la tabla Cita?
 - b. ¿Cómo se relaciona un tratamiento con una cita médica?
- Diseño Físico
 - a. ¿Qué tipo de dato se usa para el campo hora en la tabla Cita?
 - b. ¿Cuál es el propósito del trigger trg_validar_cita_medico?
- Consultas SQL
 - a. Escribe una consulta para obtener todos los tratamientos prescritos a un paciente específico.
 - b. Escribe una consulta para listar los médicos con más citas en el último mes.

Sesión 08

Primer Examen Parcial.

Rúbrica de Evaluación – Caso “La Clínica San Vital”

Criterio	Excelente (5 pts)	Bueno (4 pts)	Regular (3 pts)	Insuficiente (1-2 pts)
Análisis del caso	Identifica claramente los problemas, necesidades y contexto del caso.	Reconoce los problemas principales con poca profundidad contextual.	Muestra comprensión parcial del caso y omite aspectos clave.	Presenta una interpretación superficial o incorrecta del caso.
Diseño conceptual	Define correctamente entidades, atributos y relaciones con claridad y coherencia.	Define entidades y relaciones con algunos errores menores.	Presenta entidades incompletas o relaciones poco claras.	Confunde entidades con atributos o presenta relaciones incorrectas.
Diseño lógico	Traduce el modelo conceptual a tablas relacionales con claves bien definidas.	Traduce el modelo con algunos errores en claves o normalización.	Presenta tablas incompletas o con errores estructurales.	No logra representar adecuadamente el modelo lógico.
Diseño físico (SQL)	Sintaxis correcta, uso adecuado de tipos de datos, claves y restricciones.	Sintaxis mayormente correcta, con pequeños errores en tipos o claves.	Código funcional pero con errores que afectan la integridad o eficiencia.	Código incompleto, con errores graves o no ejecutable.
Uso de triggers y procedimientos	Implementa correctamente lógica avanzada con triggers y funciones bien estructuradas.	Implementa lógica con algunos errores o simplificaciones.	Uso limitado o incorrecto de triggers/procedimientos.	No incluye lógica avanzada o presenta errores críticos.
Consultas SQL avanzadas	Consultas precisas, eficientes y bien estructuradas para generar reportes útiles.	Consultas funcionales con algunos errores de lógica o sintaxis.	Consultas incompletas o poco relevantes para los reportes solicitados.	Consultas incorrectas, irrelevantes o no ejecutables.
Presentación y documentación	Presenta el trabajo de forma clara, ordenada y bien documentada.	Presentación adecuada con documentación parcial.	Presentación desordenada o poco clara.	Presentación deficiente o sin documentación.
Interpretación de resultados	Analiza correctamente los resultados obtenidos en consultas y procedimientos.	Interpreta resultados con algunas imprecisiones.	Interpretación superficial o incompleta.	No interpreta los resultados o lo hace de forma errónea.

Sesión 08
Primer Examen Parcial.

Puntaje total posible: 40 puntos, dividido entre dos, genera la siguiente escala de calificación :

- 18–20: Excelente
- 15–17: Bueno
- 13–14: Regular
- <13: Insuficiente

Bibliografía

- "Desarrollo de Aplicaciones Android con App Inventor 2", TOMAS DOMINGUEZ MINGUEZ, Ed. Marcombo, 2022.
- "Crea tus Aplicaciones Android con app Inventor 2", José Manuel Cabello García Ed. IC.
- "Android - Guía para desarrolladores", Ableson, F., Sen, R., King, C. Ed. Anaya Multimedia.
- "El Gran Libro de Android", Gironés, J.
Ed. Alfaomega. Ediciones de la U, 2023;
ISBN 9587924649, 9789587924640; N.º de páginas:316.
- <https://docs.oracle.com/en/database/>