



Instituto Tecnológico De Costa Rica

Escuela de Ingeniería en Computación

Bases de Datos I - GR 20

Anteproyecto del Proyecto 1

Aplicación Web con persistencia en una base de datos
relacional

Tema: Sistema de Gestión Clínica para Doctores

Estudiantes:

Luis Daniel Barboza Alfaro 2024211470

Jose Mario Castro Cruz 2022437423

Profesor:

Maria Auxiliadora Mora Cross

II Semestre

2025

Contenido

Introducción	3
Descripción del sistema	3
Objetivos y alcance del sistema	3
Objetivo general:	3
Objetivos específicos:	4
Alcance	4
Conceptualización del sistema	4
Casos de uso a implementar	5
Diseño del sistema	6
Arquitectura propuesta	6
Modelo de clases	6
Diagrama físico de la base de datos	7
Diccionario de datos	7
Uso de una función	9
Procedimiento almacenado	9
Trigger	9

Introducción

Consiste en una aplicación web diseñada exclusivamente para el uso particular de médicos y profesionales de la salud en su práctica clínica diaria. El sistema centraliza en una única plataforma segura todas las operaciones esenciales que un médico requiere para gestionar su consulta de manera autónoma y eficiente.

La plataforma permite administrar de forma integral el historial médico de cada de pacientes, manteniendo sus historiales médicos completos y actualizados. Incorpora un sistema de agenda personalizado para la gestión de citas, dando al médico control total sobre su disponibilidad y horarios de consulta.

Incluye funcionalidades para el registro detallado de cada visita médica, con la posibilidad de generar recetas digitales vinculadas directamente al historial del paciente. Todo el sistema está orientado a optimizar el flujo de trabajo clínico, garantizando que la información esté disponible de manera inmediata y segura durante las consultas.

La solución está desarrollada bajo un enfoque responsivo, asegurando su correcto funcionamiento tanto en equipos de escritorio como en dispositivos móviles, lo que permite al médico acceder a la información necesaria en cualquier momento y lugar.

Descripción del sistema

Esta es una aplicación web para médicos que les permite gestionar todas sus tareas desde el celular, tablet o computadora.

Los doctores pueden ver su agenda del día, con la lista de sus citas programadas. Desde ahí mismo, pueden acceder al historial médico de cada paciente para revisar sus consultas anteriores antes de la nueva visita.

Durante la consulta, usan la app para registrar las notas de la visita actual y pueden generar recetas médicas digitales al instante, las cuales quedan guardadas en el historial del paciente.

Objetivos y alcance del sistema

Objetivo general:

- Desarrollar una aplicación web integral que optimice la gestión clínica y la atención al paciente para los doctores.

Objetivos específicos:

- Implementar un sistema centralizado de administración de agenda médica e historiales clínicos.
- Facilitar el registro digital de consultas médicas durante la atención al paciente.
- Automatizar la generación y almacenamiento de recetas médicas digitales.
- Asegurar la integridad de los datos mediante un sistema de base de datos relacional normalizado.
- Facilitar búsquedas de historial por paciente.

Alcance

El sistema abarcará la **gestión integral de la atención médica** dentro de una clínica o consultorio, enfocado principalmente en el **trabajo de los médicos**.

Incluye:

- Registro y administración de pacientes.
- Gestión de agenda y citas médicas.
- Consulta y actualización de historiales clínicos.
- Registro digital de consultas médicas en tiempo real.
- Generación y almacenamiento de recetas médicas digitales.
- Acceso seguro y rápido a la información desde dispositivos móviles o de escritorio.
- Uso de una base de datos relacional normalizada para garantizar la integridad y consistencia de los datos.

Queda fuera del alcance:

- Procesos administrativos como facturación, pagos o seguros médicos.
- Integración con sistemas externos de salud.
- Módulos avanzados de análisis estadístico o inteligencia artificial.

Conceptualización del sistema

El Sistema de Gestión Clínica es una **plataforma web centralizada** que permitirá a los médicos administrar, desde cualquier dispositivo, la información de sus pacientes y su agenda de citas de forma segura y eficiente.

El sistema se organiza en **módulos interconectados**:

- **Módulo de Pacientes:** registro, actualización y consulta de la información personal y médica de cada paciente.
- **Módulo de Citas:** programación, modificación y visualización de la agenda médica diaria.
- **Módulo de Historial Médico:** acceso al historial completo del paciente para revisar consultas previas, diagnósticos y tratamientos anteriores.
- **Módulo de Recetas Médicas:** generación, almacenamiento y consulta de recetas digitales.

Cada módulo estará respaldado por una **base de datos relacional normalizada**, con reglas y validaciones que garanticen la integridad de la información.

La aplicación ofrecerá un **acceso seguro** y permitirá a los médicos realizar todas estas tareas de manera rápida y sencilla desde el celular, tableta o computadora.

Casos de uso a implementar

Caso de uso 1: Registrar Paciente

Actor: Doctor

Precondición: El doctor inició sesión.

Postcondición: Se crea un nuevo paciente con id único disponible para citas e historiales.

Flujo básico: El doctor abre "Pacientes" → "Nuevo Paciente", ingresa datos obligatorios (nombre, primer apellido, fecha de nacimiento, sexo, dirección, teléfono, correo) y no obligatorios (segundo apellido), le da a "Confirmar", luego el sistema valida formatos, guarda y muestra confirmación con id.

Flujos alternos: Si el correo/teléfono existe se advierte posible duplicado y se puede revisar antes de guardar; si faltan datos obligatorios no permite continuar hasta completarlos.

Resultado: Paciente registrado correctamente en el sistema.

Caso de uso 2: Agendar Cita

Actor: Doctor

Precondición: Doctor inició sesión y el paciente existe en el sistema (o se registra en ese momento).

Postcondición: Se crea una cita asociada al paciente y al doctor en su calendario.

Flujo básico: El doctor accede a "Citas" → "Agendar Nueva Cita", busca paciente, elige fecha y hora disponible, ingresa motivo opcional, confirma; el sistema valida disponibilidad, guarda y muestra confirmación.

Flujos alternos: Si hay conflicto de horario el sistema no permite crear la cita; si el paciente no existe permite registrarlo y continuar.

Resultado: Cita agendada.

Caso de uso 3: Registrar Historial Médico por Visita

Actor: Doctor

Precondición: Doctor inició sesión y existe un paciente (con cita asociada).

Postcondición: Se crea un registro de diagnóstico y tratamiento en la historia clínica del paciente; opcionalmente se vinculan recetas.

Flujo básico: El doctor entra a "Historiales Médicos", busca paciente (o selecciona cita), elige "Nueva Visita", ingresa diagnóstico y tratamiento, agrega recetas si aplica, confirma; el sistema vincula historial con paciente y cita.

Flujos alternos: Si hay prescripción médica, el sistema abre flujo de "Emitir Receta" y vincula la receta al historial creado.

Resultado: Historial médico registrado y disponible para futuras consultas.

Caso de uso 4: Emitir Receta

Actor: Doctor

Precondición: Doctor inició sesión y hay un historial médico activo al que asociar la receta.

Postcondición: Se crea una o más recetas asociadas al historial médico del paciente.

Flujo básico: Durante o después del registro del historial, el doctor selecciona “Agregar Receta”, ingresa medicamento, dosis, frecuencia y duración, confirma; el sistema guarda y vincula la receta al historial del paciente.

Flujos alternos: N/A.

Resultado: Receta emitida y asociada al historial médico.

Diseño del sistema

Arquitectura propuesta

El sistema es **modular y de tres capas**:

- **Capa de presentación:** interfaz web responsiva (desktop y móvil) donde el **doctor** gestiona pacientes, citas, historiales y recetas.
- **Capa de lógica de negocio:** valida datos, controla flujos (registro de pacientes, agendamiento, historial médico y recetas), gestiona especialidades del doctor y asegura integridad de la información.
- **Capa de datos:** base de datos centralizada con tablas para Pacientes, Citas, Historiales Médicos, Recetas y Especialidades. Garantiza persistencia y consistencia.

Es un sistema **para un único doctor** ya registrado, que puede añadir nuevas especialidades cuando lo requiera y que atiende a todos sus pacientes desde un solo perfil.

Modelo de clases

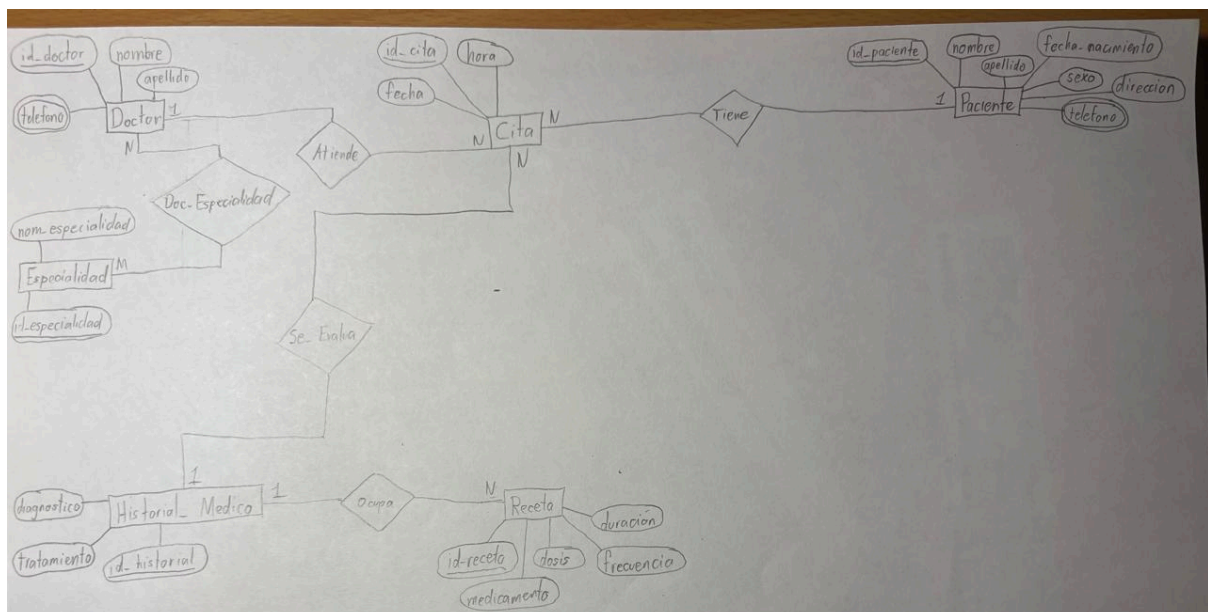
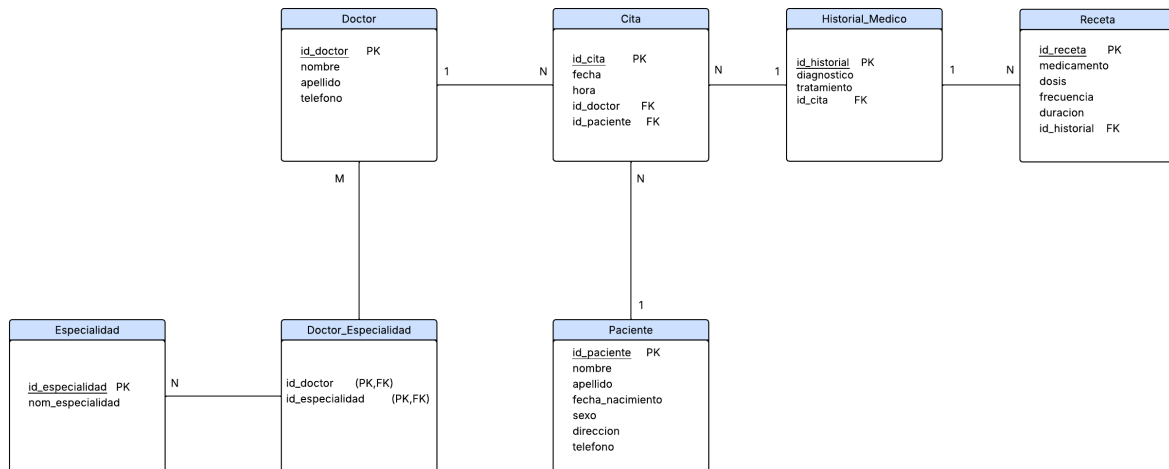


Diagrama físico de la base de datos



Diccionario de datos

Tabla: paciente

Nombre del Campo	Descripción	Tipo	Restricciones
id_paciente	Identificador único del paciente.	SERIAL	PK, NOT NULL
nombre	Primer nombre del paciente.	VARCHAR(100)	NOT NULL
apellido	Apellido del paciente.	VARCHAR(100)	NOT NULL
fecha_nacimiento	Fecha de nacimiento del paciente.	DATE	NOT NULL
sexo	Sexo biológico del paciente (M: Masculino, F: Femenino, O: Otro).	ENUM('M','F','O')	NOT NULL (Validado por dominio)
direccion	Dirección de residencia del paciente.	TEXT	NULL (Opcional)
telefono	Número de teléfono del paciente.	VARCHAR(20)	NOT NULL

Tabla: doctor

Nombre del Campo	Descripción	Tipo	Restricciones
id_doctor	Identificador único del doctor.	SERIAL	PK, NOT NULL
nombre	Nombre completo del doctor.	VARCHAR(100)	NOT NULL
apellido	Apellido del doctor.	VARCHAR(100)	NOT NULL

especialidad	Especialidad médica principal del doctor.	VARCHAR(100)	NOT NULL
--------------	---	--------------	----------

Tabla: especialidad

Nombre del Campo	Descripción	Tipo	Restricciones
id_especialidad	Identificador único del número de especialidad.	SERIAL	PK (Llave Primaria), NOT NULL
nom_especialidad	Nombre de la especialidad	VARCHAR(100)	NOT NULL

Tabla: doc_especialidad

Nombre del Campo	Descripción	Tipo	Restricciones
id_especialidad	Identificador del número de especialidad.	INT	FK → especialidad.id_especialidad, NOT NULL
id_doctor	Identificador del número del doctor.	INT	FK → doctor.id_doctor, NOT NULL

Tabla: historial_medico

Nombre del Campo	Descripción	Tipo	Restricciones
id_historial	Identificador único del registro en el historial médico.	SERIAL	PK, NOT NULL
diagnostico	Diagnóstico resultante de la consulta.	TEXT	NOT NULL
tratamiento	Tratamiento prescrito al paciente.	TEXT	NULL (Opcional)
id_cita	Cita que se agrega al historial del paciente.	INT	NOT NULL

Tabla: cita

Nombre del Campo	Descripción	Tipo	Restricciones
id_cita	Identificador único de la cita.	SERIAL	PK, NOT NULL
fecha	Fecha de la cita.	DATE	NOT NULL
hora	Hora de la cita.	TIME	NOT NULL
id_paciente	Paciente que tiene la cita.	INT	FK → paciente.id_paciente, NOT NULL
id_doctor	Doctor que atiende la cita.	INT	FK → paciente.id_paciente, NOT NULL

Tabla: receta

Nombre del Campo	Descripción	Tipo	Restricciones
id_receta	Identificador único de la receta.	SERIAL	PK, NOT NULL
medicamento	Nombre comercial/generico del medicamento.	VARCHAR(200)	NOT NULL
dosis	Dosis por toma (ej. '500 mg').	VARCHAR(100)	NOT NULL
frecuencia	Frecuencia (ej. '3 veces/día').	VARCHAR(100)	NULL (Opcional)
duracion	Duración recomendada en días.	INT	NULL (Opcional)
id_historial	Historial al que se le asigna la receta.	INT	FK → historial_medico.id_historial, NOT NULL

Uso de una función

Esta función recibe una fecha de nacimiento como parámetro y devuelve la edad de la persona en años. Para calcularla, toma la diferencia de meses entre la fecha actual y la fecha de nacimiento, y la divide entre 12 para obtener la edad en años completos.

ejemplo prototipo:

```
CREATE OR REPLACE FUNCTION calcular_edad(fecha_nac DATE)
RETURN NUMBER
IS
    v_edad NUMBER;
BEGIN
    v_edad := TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nac) / 12);
    RETURN v_edad;
END;
```

Procedimiento almacenado

El procedimiento almacenado sirve para registrar una nueva cita médica en el sistema. Primero verifica que el paciente exista en la base de datos y luego confirma que el doctor también exista. A continuación, comprueba que la fecha y hora de la cita no sea en el pasado y valida que el doctor esté disponible en ese horario, es decir, que no tenga otra cita

dentro de un margen de treinta minutos. Si todo está correcto, inserta la nueva cita en la tabla correspondiente y devuelve el identificador de la cita creada. En caso de que ocurra algún error durante el proceso, se muestra un mensaje indicando la causa del problema.

ejemplo prototipo:

```
CREATE OR REPLACE PROCEDURE agendar_cita(
    p_id_paciente IN NUMBER,
    p_id_doctor IN NUMBER,
    p_fecha_hora IN DATE,
    p_motivo IN VARCHAR2(100)
)
IS
    v_paciente_existe NUMBER;
    v_doctor_existe NUMBER;
    v_id_cita NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_paciente_existe
    FROM paciente
    WHERE id_paciente = p_id_paciente;

    IF v_paciente_existe = 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Paciente no existe');
    END IF;

    SELECT COUNT(*) INTO v_doctor_existe
    FROM doctor
    WHERE id_doctor = p_id_doctor;

    IF v_doctor_existe = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Doctor no existe');
    END IF;

    IF p_fecha_hora <= SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20003, 'La cita debe ser en fecha futura');
    END IF;

    SELECT COUNT(*) INTO v_id_cita
    FROM cita
    WHERE id_doctor = p_id_doctor
        AND fecha_hora BETWEEN p_fecha_hora - (30/1440)
        AND p_fecha_hora + (30/1440);

    IF v_id_cita > 0 THEN
        RAISE_APPLICATION_ERROR(-20004, 'Doctor no disponible en ese horario');
    END IF;

    INSERT INTO cita (id_cita, id_paciente, id_doctor, fecha_hora, motivo)
    VALUES (cita_seq.NEXTVAL, p_id_paciente, p_id_doctor, p_fecha_hora, p_motivo)
    RETURNING id_cita INTO v_id_cita;

    DBMS_OUTPUT.PUT_LINE('Cita agendada exitosamente con ID: ' || v_id_cita);
```

```

EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20005, 'Error al agendar cita: ' || SQLERRM);
END;
/

```

Trigger

Este disparador se ejecuta automáticamente después de que se inserta una nueva cita en la base de datos.

Su función es crear un registro vacío en la tabla de historial médico para ese paciente y esa cita, dejando preparado un espacio donde luego se podrán añadir el diagnóstico y el tratamiento del paciente

ejemplo prototipo:

```

CREATE OR REPLACE TRIGGER trigger_crear_historial
AFTER INSERT ON cita
FOR EACH ROW
BEGIN
    INSERT INTO historial_medico (
        id_historial,
        id_paciente,
        id_cita,
        fecha_consulta,
        diagnostico,
        tratamiento
    ) VALUES (
        historial_seq.NEXTVAL,
        :NEW.id_paciente,
        :NEW.id_cita,
        TRUNC(:NEW.fecha_hora),
        '',
        ''
    );
END;
/

```