

TRABAJO PRÁCTICO NRO 2

SEMINARIO DE PRÁCTICA

DE INFORMATICA

Sistema de Gestión Integral

para una Clínica de Salud

Antonella Diaz  
DNI 28.910.424  
VINFOR12606

## Índice

1. Etapa de análisis.
2. Etapa de diseño.
3. Etapa de implementación.
4. Etapa de pruebas.
5. Definición de base de datos para el sistema.
6. Diagrama entidad-relación de la base de datos.
7. Creación de las tablas MySQL.
8. Inserción, consulta y borrado de registros.
9. Presentación de las consultas SQL.
10. Definiciones de comunicación.

# 1. Etapa de Análisis

## 1. Recolección de Información:

- Entrevistas con Personal: Realizar entrevistas con médicos, enfermeras, administrativos y otros usuarios clave del sistema.
- Análisis de Procesos Existentes: Revisar los procedimientos actuales para identificar puntos de mejora.
- Revisión de Documentación: Examinar documentos, registros y sistemas actuales usados por la clínica.

## 2. Requerimientos del Sistema

### 1. Gestión de Citas

- Programación de citas médicas
- Recordatorios automáticos
- Cancelación o reprogramación de citas

### 2. Historias Clínicas Electrónicas

- Creación y gestión de historias clínicas
- Seguridad y confidencialidad de datos

### 3. Gestión de Inventarios

- Registro de inventarios
- Alertas automáticas

### 4. Facturación y Pagos

- Generación de facturas
- Integración con sistemas de pago

### 5. Reportes de Gestión

- Generación de reportes personalizables

### 3. Casos de Uso

#### 1. Programación de Citas

- Actor Principal: Paciente
- Descripción: El paciente programa una cita médica seleccionando médico, fecha y hora.
- Precondiciones: El paciente debe estar registrado en el sistema.
- Postcondiciones: La cita queda registrada en el sistema y el paciente recibe un recordatorio.

#### 2. Gestión de Historias Clínicas

- Actor Principal: Médico
- Descripción: El médico registra una nueva historia clínica o actualiza una existente.
- Precondiciones: El paciente debe tener una cita con el médico.
- Postcondiciones: La información médica queda registrada en el sistema de forma segura.

## 2. Etapa de Diseño

### Arquitectura del Sistema

- Frontend: Aplicación web con HTML, CSS y JavaScript.
- Backend: API RESTful con Node.js y Express.
- Base de Datos: MySQL.

### Modelo de Datos Relacional

#### 1. Pacientes

- id (PK)
- nombre
- apellido
- fecha\_nacimiento
- dirección
- teléfono

- correo\_electrónico

## 2. Médicos

- id (PK)
- nombre
- apellido
- especialidad
- teléfono
- correo\_electrónico

## 3. Citas

- id (PK)
- paciente\_id (FK)
- medico\_id (FK)
- fecha\_hora
- estado

## 4. Historias\_Clinicas

- id (PK)
- paciente\_id (FK)
- medico\_id (FK)
- fecha
- detalles

## 5. Inventarios

- id (PK)
- producto
- cantidad
- fecha\_vencimiento
- proveedor\_id (FK)

## 6. Facturas

- id (PK)

- paciente\_id (FK)
- fecha
- total
- estado

#### 7. Proveedores

- id (PK)
- nombre
- contacto

#### 8. Reportes

- id (PK)
- tipo\_reporte
- fecha\_creacion
- detalles

### 3. Etapa de Implementación

#### 1. Desarrollo del Sistema:

- Creación de Módulos: se desarrolla cada módulo del sistema de acuerdo con los requerimientos funcionales.
- Integración de Módulos: nos aseguramos que los módulos se comuniquen e integren correctamente.

#### 2. Base de Datos:

- Definición de Tablas MySQL: se crea las tablas de la base de datos de acuerdo con el modelo de datos relacional.
- Implementación de Relaciones y Restricciones: se definen claves primarias, claves foráneas y restricciones para mantener la integridad de los datos.

#### 3. Inserción, Consulta y Borrado de Registros:

- Operaciones CRUD: se implementa funciones para crear, leer, actualizar y eliminar registros en la base de datos.

## 4. Etapa de Pruebas

### Documentación de Pruebas

#### 1. Pruebas Unitarias

- **Objetivo:** Verificar que cada unidad de código funcione correctamente.
- **Método:** Uso de frameworks de prueba como Mocha y Chai para Node.js.

#### 2. Pruebas de Integración

- **Objetivo:** Verificar que las diferentes unidades de código funcionen juntas correctamente.
- **Método:** Simulación de interacciones entre los módulos del sistema.

#### 3. Pruebas de Sistema

- **Objetivo:** Verificar que el sistema en su conjunto funcione según lo esperado.
- **Método:** Ejecución de casos de uso completos.

#### 4. Pruebas de Usuario

- **Objetivo:** Asegurar que el sistema cumpla con las expectativas de los usuarios finales.
- **Método:** Realización de pruebas con usuarios reales y recopilación de feedback.

## 5. Definición de Base de Datos para el Sistema

### Descripción de Tablas y Relaciones

- **Pacientes** y **Médicos** son entidades clave con relaciones uno-a-muchos con **Citas**.
- **Historias\_Clinicas** está relacionada con **Pacientes** y **Médicos**.

- **Inventarios y Facturas** están relacionadas con **Proveedores y Pacientes**, respectivamente.

## 6. Diagrama Entidad-Relación de la Base de Datos

### Entidades y Relaciones

#### 1. Pacientes

- Atributos: id (PK), nombre, apellido, fecha\_nacimiento, dirección, teléfono, correo\_electrónico

#### 2. Médicos

- Atributos: id (PK), nombre, apellido, especialidad, teléfono, correo\_electrónico

#### 3. Citas

- Atributos: id (PK), paciente\_id (FK), medico\_id (FK), fecha\_hora, estado
- Relaciones:
  - Cada cita está asociada con un paciente.
  - Cada cita está asociada con un médico.

#### 4. Historias\_Clinicas

- Atributos: id (PK), paciente\_id (FK), medico\_id (FK), fecha, detalles
- Relaciones:
  - Cada historia clínica está asociada con un paciente.
  - Cada historia clínica está asociada con un médico.

#### 5. Inventarios

- Atributos: id (PK), producto, cantidad, fecha\_vencimiento, proveedor\_id (FK)
- Relaciones:
  - Cada inventario está asociado con un proveedor.



## 6. Facturas

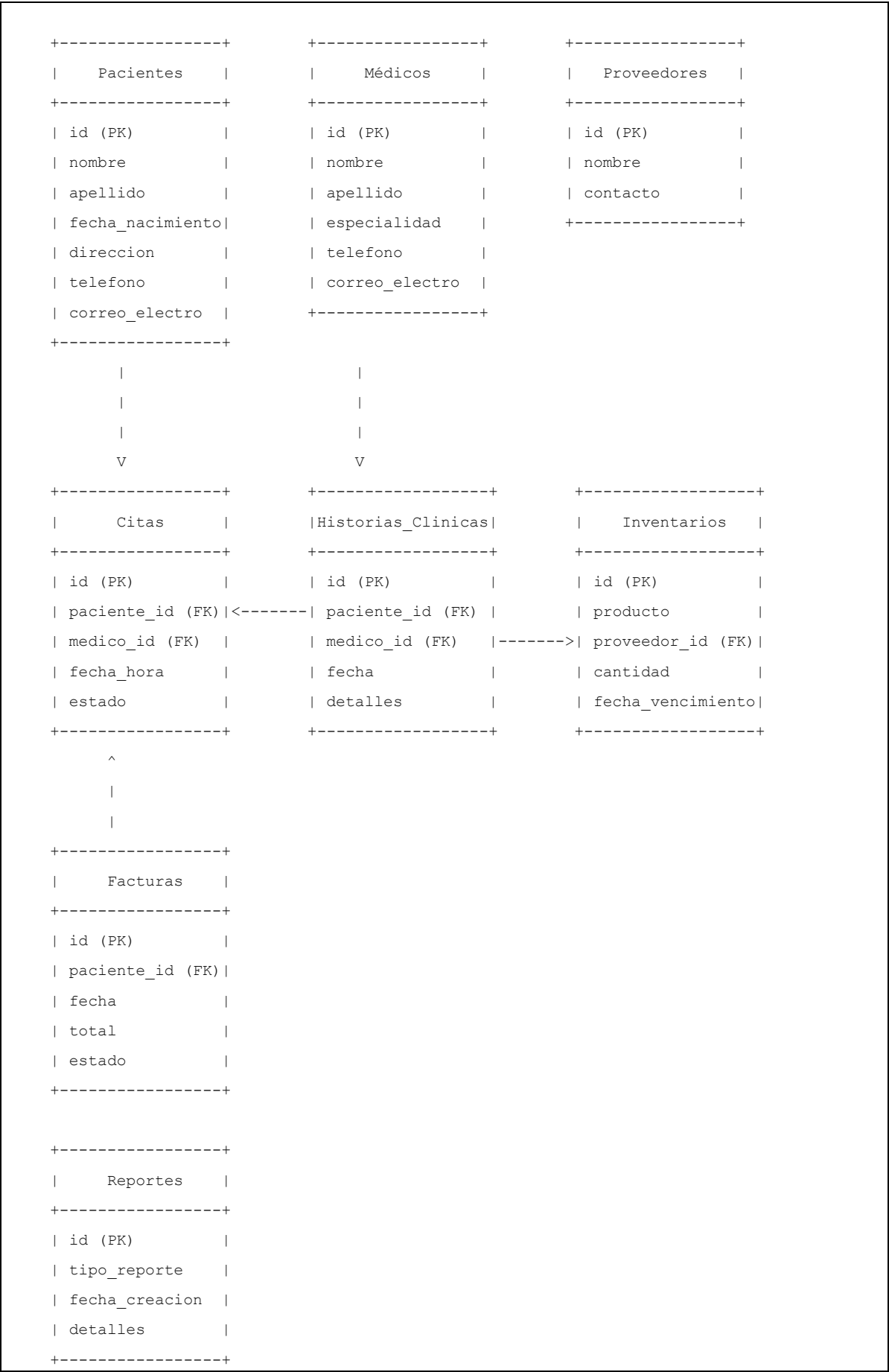
- Atributos: id (PK), paciente\_id (FK), fecha, total, estado
- Relaciones:
  - Cada factura está asociada con un paciente.

## 7. Proveedores

- Atributos: id (PK), nombre, contacto

## 8. Reportes

- Atributos: id (PK), tipo\_reporte, fecha\_creacion, detalles



## Descripción del Diagrama

- **Pacientes** y **Médicos** son entidades principales que tienen relaciones uno-a-muchos con **Citas** y **Historias\_Clinicas**.
- **Citas** y **Historias\_Clinicas** tienen claves foráneas que se refieren a **Pacientes** y **Médicos**.
- **Inventarios** tiene una relación con **Proveedores**.
- **Facturas** está asociada con **Pacientes**.

**Reportes** es una entidad independiente que almacena información sobre los reportes generados por el sistema

## 7. Creación de las Tablas MySQL

### Scripts SQL para la Creación de Tablas

```
-- Creación de la tabla de Pacientes
CREATE TABLE Pacientes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50),
  apellido VARCHAR(50),
  fecha_nacimiento DATE,
  direccion VARCHAR(100),
  telefono VARCHAR(20),
  correo_electronico VARCHAR(50)
);

-- Creación de la tabla de Médicos
CREATE TABLE Medicos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50),
  apellido VARCHAR(50),
  especialidad VARCHAR(50),
  telefono VARCHAR(20),
  correo_electronico VARCHAR(50)
);

-- Creación de la tabla de Citas
CREATE TABLE Citas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  paciente_id INT,
  medico_id INT,
  fecha_hora DATETIME,
  estado VARCHAR(20),
  FOREIGN KEY (paciente_id) REFERENCES Pacientes(id),
  FOREIGN KEY (medico_id) REFERENCES Medicos(id)
);

-- Creación de la tabla de Historias Clínicas
CREATE TABLE Historias_Clinicas (
  id INT AUTO_INCREMENT PRIMARY KEY,
  paciente_id INT,
  medico_id INT,
  fecha DATE,
  detalles TEXT,
  FOREIGN KEY (paciente_id) REFERENCES Pacientes(id),
  FOREIGN KEY (medico_id) REFERENCES Medicos(id)
);

-- Creación de la tabla de Inventarios
CREATE TABLE Inventarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  producto VARCHAR(100),
```

```
    cantidad INT,
    fecha_vencimiento DATE,
    proveedor_id INT,
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(id)
);

-- Creación de la tabla de Facturas
CREATE TABLE Facturas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT,
    fecha DATE,
    total DECIMAL(10, 2),
    estado VARCHAR(20),
    FOREIGN KEY (paciente_id) REFERENCES Pacientes(id)
);

-- Creación de la tabla de Proveedores
CREATE TABLE Proveedores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    contacto VARCHAR(100)
);

-- Creación de la tabla de Reportes
CREATE TABLE Reportes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tipo_reporte VARCHAR(50),
    fecha_creacion DATE,
    detalles TEXT
);
```

## 8.Inserción, consulta y borrado de registros.

Inserción:

```
INSERT INTO Pacientes (nombre, apellido, fecha_nacimiento, dirección, teléfono,
correo_electrónico)
VALUES ('Juan', 'Pérez', '1985-03-15', 'Calle Falsa 123', '123456789',
'juan.perez@example.com');
```

Consulta:

```
SELECT * FROM Pacientes WHERE apellido = 'Pérez';
```

Borrado:

```
DELETE FROM Pacientes WHERE id = 1;
```

## 9.Presentación de las consultas SQL.

Consultas Ejemplos:

1. Listar citas por médico:

```
SELECT * FROM Citas WHERE medico_id = 1;
```

2. Buscar pacientes por nombre:

```
SELECT * FROM Pacientes WHERE nombre LIKE '%Juan%';
```

3. Inventario próximo a vencer:

```
SELECT * FROM Inventarios WHERE fecha_vencimiento < CURDATE() + INTERVAL 30 DAY;
```

## 10.Definiciones de Comunicación

Entorno de Red

- Protocolos de Red: Utilización de HTTP/HTTPS para la comunicación web.
- Infraestructura Física: Descripción de los servidores, routers y otros dispositivos de red necesarios para soportar el sistema.

Control de Enlace de Datos

- Seguridad en la Transmisión: Implementación de SSL/TLS para asegurar la transmisión de datos entre el sistema y los usuarios.
- Integridad y Confidencialidad: Uso de cifrado de datos y autenticación de usuarios para proteger la información sensible.

Interoperabilidad

- Estándares de Comunicación: Adopción de estándares como HL7 para asegurar la interoperabilidad con otros sistemas de salud.
- API y Web Services: Definición de APIs y servicios web para facilitar la integración con sistemas externos y permitir el acceso a funcionalidades específicas del sistema.