**DESARROLLO DE APLICACIONES WEB.**

**Título: Sistema de Gestión Médica para Clínica Oftalmológica Bienestar.**

**COMPONENTES DEL GRUPO:**

* TARSICIO SANTANDER MORGADES.
* ZABULON SIMA OLUY.

**PROFESOR DE LA MATERIA:** FERMÍN COPOBORÚ LOERI.

FECHA:24/03/2025.

**Introducción.**

El proyecto consiste en un sistema web para la gestión de una clínica oftalmológica, facilitando la administración de citas médicas, información de especialistas y la interacción con pacientes.

**Capítulo I: Planteamiento del Problema.**

Desarrollo de una plataforma web para la gestión integral de servicios médicos oftalmológicos, incluyendo gestión de citas, perfiles médicos y servicios especializados.

**1.2 Justificación del problema.**

* Necesidad de digitalizar procesos médicos.
* Mejora en la gestión de citas.
* Acceso fácil a información de especialistas.
* Optimización de procesos administrativos.

**1.3 Objetivo general.**

Implementar un sistema web que optimice la gestión de servicios médicos y mejore la experiencia de pacientes en la Clínica Bienestar.

**1.4 Objetivos específicos.**

* Desarrollar un sistema de gestión de citas médicas.
* Implementar perfiles detallados de especialistas.
* Crear una interfaz intuitiva para pacientes.
* Facilitar la administración de servicios médicos.

**1.5 Metodología de investigación.**

* Metodología ágil.
* Desarrollo iterativo.
* Enfoque centrado en el usuario.

**Capitulo II: solución del problema.**

**2.1 Breve historial de la empresa.**

Clínica Bienestar es un centro oftalmológico especializado en servicios de salud visual.

**2.2 Organigrama de la empresa.**

DIRECTOR GENERAL

ADMINISTRADOR

ESPECIALISTAS

RECEPCIONISTA

**2.3. Diagrama de la vida útil de tu proyecto de investigación (ciclo de vida).**

**MODELO EN V**

Planificación

Análisis

Diseño

Implementación

Pruebas

Instalación

¿SE HA HECHO?

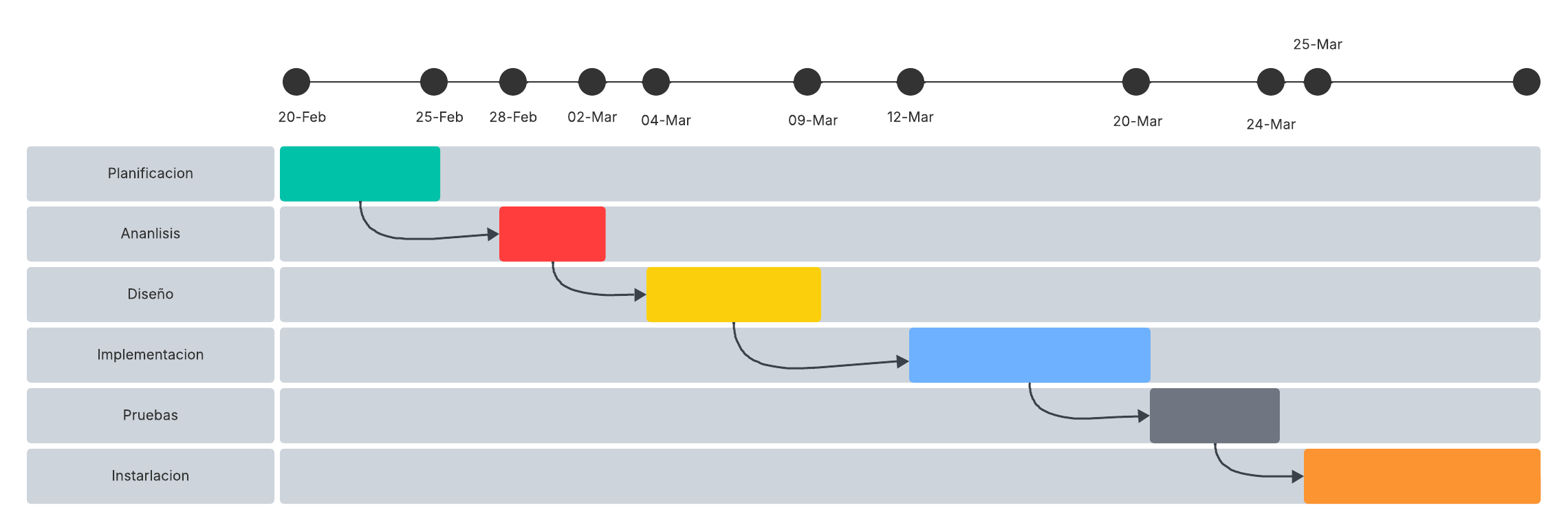
Mantenimiento

Validar

Verificar

¿SE HA HECHO BIEN?

**2.4. Diagrama de Gantt.**



**2.5. Diseño del sistema**

**2.5.1.Casos de uso.**













**2.5.2. Modelo conceptual.**

## Entidades y Relaciones

### 1. ROL

- \*\*Atributos:\*\*

- id (PK)

- nombre

### 2. PACIENTES

- \*\*Atributos:\*\*

- id (PK)

- nombre

- apellido

- fecha\_nacimiento

- genero

- direccion

- telefono

- edad

- email

- alergias

- \*\*Relaciones:\*\*

- Tiene un ROL (N:1)

- Tiene CITAS (1:N)

- Tiene HISTORIAL\_MEDICO (1:N)

- Tiene USUARIO (1:1)

### 3. EMPLEADOS

- \*\*Atributos:\*\*

- id (PK)

- perfil

- nombre

- apellido

- telefono

- nacionalidad

- email

- direccion

- fecha\_registro

- \*\*Relaciones:\*\*

- Tiene un ROL (N:1)

- Puede ser MEDICO (1:1)

- Tiene USUARIO (1:1)

### 4. MEDICOS

- \*\*Atributos:\*\*

- id (PK)

- especialidad

- titulo\_profesional

- experiencia

- idiomas

- \*\*Relaciones:\*\*

- Es un EMPLEADO (1:1)

- Tiene CITAS (1:N)

- Crea HISTORIAL\_MEDICO (1:N)

### 5. CITAS

- \*\*Atributos:\*\*

- id (PK)

- fecha\_cita

- hora\_cita

- motivo

- tipo (Consulta/Analisis/Revision)

- estado (Programada/Realizada/Cancelada/Reprogramada)

- fecha\_registro

- \*\*Relaciones:\*\*

- Pertenece a PACIENTE (N:1)

- Asignada a MEDICO (N:1)

- Genera HISTORIAL\_MEDICO (1:1)

### 6. HISTORIAL\_MEDICO

- \*\*Atributos:\*\*

- id (PK)

- fecha

- diagnostico

- tratamiento

- receta

- \*\*Relaciones:\*\*

- Pertenece a PACIENTE (N:1)

- Creado por MEDICO (N:1)

- Relacionado con CITA (N:1)

### 7. USUARIO

- \*\*Atributos:\*\*

- cod\_usuario (PK)

- correo

- passwd

- estado

- \*\*Relaciones:\*\*

- Puede ser PACIENTE (1:1)

- Puede ser EMPLEADO (1:1)

## Cardinalidades Principales

- ROL ← (1:N) → PACIENTES

- ROL ← (1:N) → EMPLEADOS

- EMPLEADOS ← (1:1) → MEDICOS

- PACIENTES ← (1:N) → CITAS

- MEDICOS ← (1:N) → CITAS

- CITAS ← (1:1) → HISTORIAL\_MEDICO

- USUARIO ← (1:1) → PACIENTES

- USUARIO ← (1:1) → EMPLEADOS

**2.5.2. Modelo Logico.**

CREATE DATABASE ClinicaBienestar

    DEFAULT CHARACTER SET = 'utf8mb4';

USE ClinicaBienestar;

CREATE TABLE rol(

    id INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(100) NOT NULL

);

    CREATE TABLE pacientes (

        id INT AUTO\_INCREMENT PRIMARY KEY,

        nombre VARCHAR(100) NOT NULL,

        apellido VARCHAR(100) NOT NULL,

        fecha\_nacimiento DATE,

        genero ENUM('Masculino', 'Femenino', 'Otro'),

        direccion TEXT,

        telefono VARCHAR(20),

        edad INT,

        email VARCHAR(100) UNIQUE,

        alergias TEXT,

        id\_rol INT,

        FOREIGN KEY (id\_rol) REFERENCES rol(id)

    );

SELECT \* FROM pacientes;

CREATE TABLE empleados(

    id INT AUTO\_INCREMENT PRIMARY KEY,

    perfil VARCHAR(100) NOT NULL,

    nombre VARCHAR(100) NOT NULL,

    apellido VARCHAR(100) NOT NULL,

    telefono VARCHAR(20),

    nacionalidad VARCHAR(100) NOT NULL,

    email VARCHAR(100) NOT NULL UNIQUE,

    direccion TEXT,

    id\_rol INT ,

    fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (id\_rol) REFERENCES rol(id)

);

CREATE TABLE medicos(

    id INT AUTO\_INCREMENT PRIMARY KEY,

    especialidad VARCHAR(100) NOT NULL,

    titulo\_profesional VARCHAR(100) NOT NULL,

    experiencia INT NOT NULL,

    idiomas VARCHAR(100) NOT NULL,

    id\_empleado INT,

    FOREIGN KEY (id\_empleado) REFERENCES empleados(id)

);

CREATE TABLE citas (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    paciente\_id INT NOT NULL,

    medico\_id INT NOT NULL,

    fecha\_cita DATE NOT NULL,

    hora\_cita TIME NOT NULL,

    motivo TEXT,

    tipo ENUM('Consulta', 'Analisis', 'Revision') DEFAULT 'Consulta',

    estado ENUM('Programada', 'Realizada', 'Cancelada', 'Reprogramada') DEFAULT 'Programada',

    fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (paciente\_id) REFERENCES pacientes(id),

    FOREIGN KEY (medico\_id) REFERENCES medicos(id)

);

CREATE TABLE historial\_medico (

    id INT PRIMARY KEY AUTO\_INCREMENT,

    paciente\_id INT NOT NULL,

    id\_medico INT NOT NULL,

    id\_cita INT NOT NULL,

    fecha DATE NOT NULL,

    diagnostico TEXT NOT NULL,

    tratamiento TEXT NOT NULL,

    Receta TEXT NOT NULL,

    FOREIGN KEY (paciente\_id) REFERENCES pacientes(id),

    FOREIGN KEY (id\_medico) REFERENCES medicos(id),

    FOREIGN KEY(id\_cita) REFERENCES citas (id);

);

create table usuario(

    cod\_usuario int primary key auto\_increment,

    correo varchar(45) not null UNIQUE,

    passwd varchar(100) not null,

    estado BOOLEAN,

    id\_paciente int default null, foreign key(id\_paciente) references pacientes(id),

    id\_empleado int default null, foreign key(id\_empleado) references empleados(id)

);

**2.6 Técnicas de Programación, Tecnologías y Métodos**

**2.6.1 Tecnologías y Herramientas Utilizadas**

1. Frontend

- HTML5

- CSS3

- JavaScript

- Bootstrap 5.3.3

- Font Awesome (Iconografía)

- SweetAlert2 (Notificaciones)

2. Backend

- PHP

- MySQL

- FPDF (Generación de PDFs)

- AJAX (Comunicación asíncrona)

3. Servidor

- XAMPP

- Apache

- MySQL Server

4. Herramientas de Desarrollo

- Visual Studio Code

- phpMyAdmin

- Git (Control de versiones)

**2.6.2 Arquitectura del sistema o la aplicación**

**MODELO CLIENTE-SERVIDOR**

Que consiste en que el CLIENTE hace una petición, dicha petición es procesada por el Back-End y enviada a la Base de Datos, a su ves el Back-End recibe la respuesta de la Base de Datos y la devuelve al CLIENTE

**2.6.3 Arquitectura de implementación**

**- Capa de Presentación**

**- Interfaces responsivas**

**- Validación de formularios**

**- Manejo de sesiones**

**- Generación de reportes PDF**

**- Capa de Negocio**

**- Gestión de citas**

**- Control de usuarios**

**- Manejo de historiales médicos**

**- Administración de personal**

**- Capa de Datos**

**- Base de datos relacional MySQ**