Caso del Proyecto: Clínica Dental

Este documento describe los requerimientos funcionales y técnicos de un sistema de gestión para una clínica dental, desarrollado con Angular en el Frontend y basado en una base de datos MySQL.

# 1. Base de Datos

La base de datos se llama “Clínica Dental” e incluye las siguientes tablas principales: paciente, personal, historial\_medico, radiografía, tratamiento, paciente\_tratamiento, pagos y cita.

Las claves foráneas, índices y validaciones aseguran la integridad referencial y el rendimiento.

# 2. Requerimientos Funcionales del Frontend Angular

## Gestión de Pacientes

* Crear, listar, editar y eliminar pacientes.
* Buscar pacientes por nombre o RUT.
* Ver historial clínico y radiografías asociadas.

## Historial Médico

* Mostrar historial médico del paciente.
* Añadir nueva entrada al historial.
* Eliminar registros del historial.

## Radiografías

* Subir archivo de radiografía asociada a un paciente.
* Listar y visualizar radiografías.
* Eliminar registros si es necesario.

## Tratamientos

* CRUD de tratamientos (admin).
* Mostrar tratamientos disponibles con filtros por nombre y duración.

## Asignación de Tratamientos a Pacientes

* Asignar tratamientos a un paciente.
* Editar fechas, descripción y monto aplicado.
* Finalizar un tratamiento con fecha de término.

## Pagos

* Ingresar pagos de tratamientos (por paciente\_tratamiento).
* Filtrar por estado, fecha o método.
* Mostrar resumen de pagos realizados vs monto aplicado.

## Citas

* Agendar nueva cita (vinculada a paciente y tratamiento si aplica).
* Cambiar estado de la cita (programada, completada, etc.).
* Filtrar agenda por fecha o paciente.
* Mostrar observaciones y duración.

## Personal (Admin)

* CRUD de usuarios del sistema.
* Mostrar listado por roles (doctores, recepcionistas, administrativos).
* Activar/desactivar usuarios.
* Login y control de permisos (opcional).

# 3. Recomendaciones Técnicas Angular

|  |  |
| --- | --- |
| Elemento | Herramienta Angular Recomendada |
| Formularios | ReactiveFormsModule |
| Ruteo | RouterModule con `children` y `lazy-loading` |
| Diseño | Angular Material o Tailwind |
| Comunicación API | HttpClient (servicios por entidad) |
| Autenticación | JWT opcional / Angular Guards |
| Manejo de estados | `BehaviorSubject` o `NgRx` si escala |