

Memoria del Proyecto: Clinica3S

Sistema de Gestión para Clínicas Odontológicas

Asignatura: Sistemas de Información **Titulación:** Máster en Ingeniería Informática (MEI) **Fecha:** Enero 2026 **Autor** Edgar Conde

1. Descripción del Contexto de la Aplicación

1.1 Introducción

Clinica3S es un sistema de gestión integral diseñado para clínicas dentales que permite administrar de forma centralizada todos los aspectos operativos y administrativos del negocio. El sistema aborda la necesidad de digitalizar y optimizar los procesos de una clínica odontológica, desde la gestión de citas y pacientes hasta el análisis financiero mediante Business Intelligence.

1.2 Problemática que Resuelve

Las clínicas dentales tradicionales enfrentan diversos retos en su gestión diaria:

- **Gestión manual de citas:** Agendas en papel o sistemas desconectados que dificultan la coordinación
- **Falta de visibilidad financiera:** Dificultad para analizar ingresos, costes y rentabilidad
- **Control de pagos deficiente:** Seguimiento manual de deudas y pagos pendientes
- **Información de pacientes dispersa:** Historiales y datos de contacto no centralizados
- **Asignación de recursos:** Complejidad para gestionar dentistas con diferentes especialidades

1.3 Solución Propuesta

Clinica3S proporciona una plataforma web completa que integra:

- **Módulo de autenticación** con control de acceso basado en roles (RBAC)
- **Gestión integral de citas** con calendario visual y estados de seguimiento
- **Administración de pacientes** con datos de contacto y búsqueda avanzada
- **Catálogo de servicios** con control de costes y márgenes
- **Dashboard de Business Intelligence** con KPIs y gráficas interactivas
- **Control financiero** con seguimiento de facturación y pagos pendientes

1.4 Usuarios del Sistema

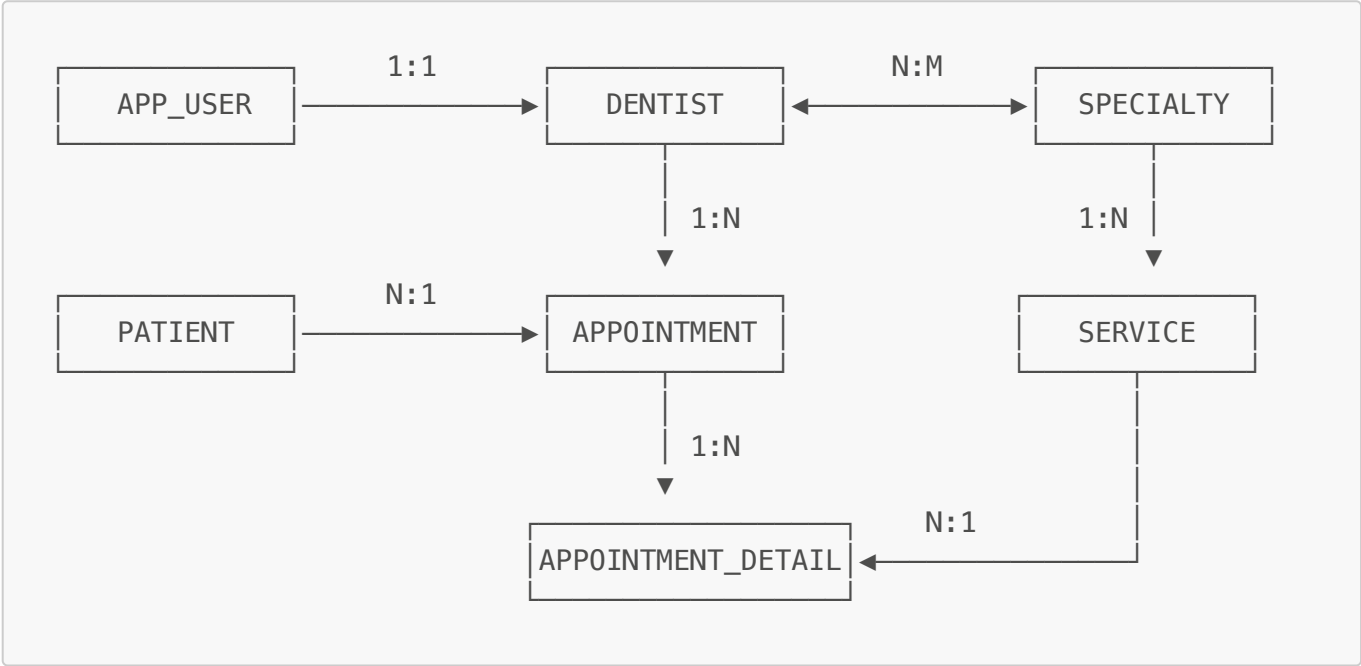
El sistema define tres roles con diferentes niveles de acceso:

Rol	Descripción	Permisos Principales
ADMIN	Administrador del sistema	Acceso completo: usuarios, configuración, reportes, todas las operaciones

Rol	Descripción	Permisos Principales
RECEPTIONIST	Personal de recepción	Gestión de citas, pacientes, servicios (sin acceso a usuarios ni reportes completos)
DENTIST	Dentistas de la clínica	Consulta de su propia agenda y citas asignadas

2. Descripción de Entidades y Relaciones

2.1 Diagrama del Modelo de Datos



2.2 Entidades Principales

APP_USER (Usuarios del Sistema)

Gestiona la autenticación y autorización de usuarios.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
username	String	Nombre de usuario único
name	String	Nombre real del usuario
password	String	Contraseña encriptada (BCrypt)
role	Enum	Rol: ADMIN, DENTIST, RECEPTIONIST
enabled	Boolean	Estado activo/inactivo

DENTIST (Dentistas)

Información profesional de los dentistas de la clínica.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
licenseNumber	String	Número de colegiado
commissionRate	BigDecimal	Tasa de comisión (%)
user_id	Long	FK → APP_USER (relación 1:1 obligatoria)

SPECIALTY (Especialidades)

Catálogo de especialidades odontológicas.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
name	String	Nombre de la especialidad

Ejemplos: Ortodoncia, Endodoncia, Cirugía Oral, Periodoncia, Odontopediatría, Odontología General, Prostodoncia, Odontología Conservadora.

SERVICE (Servicios/Tratamientos)

Catálogo de servicios ofrecidos por la clínica.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
name	String	Nombre del servicio
standardCost	BigDecimal	Coste para la clínica
listPrice	BigDecimal	Precio de venta al público
specialty_id	Long	FK → SPECIALTY

PATIENT (Pacientes)

Datos de los pacientes de la clínica.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
name	String	Nombre completo
birthDate	LocalDate	Fecha de nacimiento
gender	String	Género
phone	String	Teléfono (obligatorio)

Campo	Tipo	Descripción
email	String	Correo electrónico

APPOINTMENT (Citas)

Registro de citas médicas.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
dateTime	LocalDateTime	Fecha y hora de la cita
status	Enum	Estado: PENDING, COMPLETED, NO_SHOW
totalAmount	BigDecimal	Importe total calculado
patient_id	Long	FK → PATIENT
dentist_id	Long	FK → DENTIST

APPOINTMENT_DETAIL (Detalle de Citas)

Servicios aplicados en cada cita.

Campo	Tipo	Descripción
id	Long	Identificador único (PK)
quantity	Integer	Cantidad del servicio
priceApplied	BigDecimal	Precio en el momento de la cita
paymentDate	LocalDateTime	Fecha de pago (null si no pagado)
appointment_id	Long	FK → APPOINTMENT
service_id	Long	FK → SERVICE

2.3 Relaciones Clave

Relación	Tipo	Descripción
APP_USER → DENTIST	1:1	Cada dentista tiene un usuario vinculado obligatoriamente
DENTIST ↔ SPECIALTY	N:M	Un dentista puede tener múltiples especialidades
SPECIALTY → SERVICE	1:N	Los servicios se categorizan por especialidad
PATIENT → APPOINTMENT	1:N	Un paciente puede tener múltiples citas
DENTIST → APPOINTMENT	1:N	Un dentista atiende múltiples citas

Relación	Tipo	Descripción
APPOINTMENT → APPOINTMENT_DETAIL	1:N	Una cita contiene múltiples servicios
SERVICE → APPOINTMENT_DETAIL	1:N	Un servicio puede usarse en múltiples citas

2.4 Decisiones de Diseño Destacadas

- **priceApplied en APPOINTMENT_DETAIL**: Almacena el precio del servicio en el momento de la cita, preservando el histórico para análisis de BI aunque los precios cambien posteriormente.
- **paymentDate en APPOINTMENT_DETAIL**: Permite control granular de pagos parciales, identificando qué servicios de una cita están pagados.
- **commissionRate en DENTIST**: Facilita el cálculo automático de comisiones para reportes financieros.

3. Historias de Usuario / Casos de Uso

3.1 Autenticación y Acceso

ID	Historia de Usuario
HU-01	Como usuario, quiero iniciar sesión con mis credenciales para acceder al sistema
HU-02	Como usuario, quiero cambiar mi contraseña desde mi perfil
HU-03	Como administrador, quiero crear usuarios con diferentes roles
HU-04	Como administrador, quiero habilitar/deshabilitar usuarios

3.2 Gestión de Citas

ID	Historia de Usuario
HU-05	Como recepcionista, quiero ver el calendario de citas del día con todos los dentistas
HU-06	Como recepcionista, quiero crear una nueva cita asignando paciente, dentista, fecha/hora y servicios
HU-07	Como recepcionista, quiero ver el listado de citas con filtros por fecha, estado, dentista y paciente
HU-08	Como recepcionista, quiero ver el detalle de una cita con los servicios aplicados
HU-09	Como recepcionista, quiero cambiar el estado de una cita (PENDING, COMPLETED, NO_SHOW)
HU-10	Como recepcionista, quiero registrar el pago de una cita

ID	Historia de Usuario
HU-11	Como dentista, quiero ver mi agenda personal de citas

3.3 Gestión de Pacientes

ID	Historia de Usuario
HU-12	Como recepcionista, quiero buscar pacientes por nombre, teléfono o email
HU-13	Como recepcionista, quiero registrar un nuevo paciente con sus datos de contacto
HU-14	Como recepcionista, quiero editar los datos de un paciente existente
HU-15	Como administrador, quiero eliminar pacientes del sistema

3.4 Gestión de Dentistas

ID	Historia de Usuario
HU-16	Como administrador, quiero ver el listado de dentistas con sus especialidades
HU-17	Como administrador, quiero crear un nuevo dentista asociándolo a un usuario
HU-18	Como administrador, quiero editar los datos de un dentista (matrícula, comisión, especialidades)
HU-19	Como administrador, quiero filtrar dentistas por especialidad

3.5 Catálogo de Servicios y Especialidades

ID	Historia de Usuario
HU-20	Como administrador, quiero gestionar el catálogo de especialidades
HU-21	Como administrador, quiero crear servicios con coste y precio de venta
HU-22	Como administrador, quiero ver el margen de ganancia de cada servicio
HU-23	Como administrador, quiero asignar servicios a especialidades

3.6 Dashboard y Reportes (Business Intelligence)

ID	Historia de Usuario
HU-24	Como administrador, quiero ver KPIs de facturación, costes y ganancias
HU-25	Como administrador, quiero ver la distribución de estados de citas en una gráfica
HU-26	Como administrador, quiero ver los ingresos y comisiones por dentista
HU-27	Como administrador, quiero ver la tendencia mensual de ingresos y citas

ID	Historia de Usuario
HU-28	Como administrador, quiero filtrar las estadísticas por año
HU-29	Como administrador, quiero ver el importe de pagos pendientes de cobro

4. Estructura y Elementos del Proyecto

4.1 Arquitectura General

El proyecto sigue una arquitectura cliente-servidor con:

- **Backend:** API REST con Spring Boot (arquitectura en capas)
- **Frontend:** SPA (Single Page Application) con React
- **Base de Datos:** H2 (desarrollo) / PostgreSQL (producción)

4.2 Backend (clinica3s-backend)

Estructura de Paquetes

```
io.github.edconde.clinica3s_backend/
├── Clinica3sBackendApplication.java      # Clase principal
├── config/                             # Configuración Spring
├── controller/                         # Controladores REST
│   ├── AuthController.java
│   ├── UserController.java
│   ├── DentistController.java
│   ├── PatientController.java
│   ├── AppointmentController.java
│   ├── ServiceController.java
│   ├── SpecialtyController.java
│   └── ReportController.java
├── service/                           # Lógica de negocio
│   ├── AuthenticationService.java
│   ├── DentistService.java
│   ├── PatientService.java
│   ├── AppointmentService.java
│   ├── ServiceService.java
│   ├── SpecialtyService.java
│   └── ReportService.java
├── repository/                       # Acceso a datos (JPA)
├── entity/                           # Entidades del dominio
│   ├── AppUser.java
│   ├── Dentist.java
│   ├── Patient.java
│   ├── Appointment.java
│   ├── AppointmentDetail.java
│   ├── Service.java
│   ├── Specialty.java
│   ├── Role.java (Enum)
│   └── AppointmentStatus.java (Enum)
```

└─ dto/	# Data Transfer Objects
└─ security/	# Configuración JWT
└─ JwtService.java	
└─ JwtAuthenticationFilter.java	
└─ SecurityConfig.java	
└─ exception/	# Manejo de excepciones

Controladores y Endpoints Principales

Controlador	Endpoints	Descripción
AuthController	POST /api/auth/login, POST /api/auth/register	Autenticación
UserController	GET/POST/PUT/DELETE /api/users	Gestión de usuarios (ADMIN)
DentistController	GET/POST/PUT/DELETE /api/dentists	Gestión de dentistas
PatientController	GET/POST/PUT/DELETE /api/patients	Gestión de pacientes
AppointmentController	GET/POST/PUT /api/appointments	Gestión de citas
ServiceController	GET/POST/PUT/DELETE /api/services	Catálogo de servicios
SpecialtyController	GET/POST/PUT/DELETE /api/specialties	Especialidades
ReportController	GET /api/reports/dashboard	Dashboard BI

Tecnologías Backend

- **Spring Boot 3.5.9** - Framework base
- **Spring Security + JWT** - Autenticación stateless
- **Spring Data JPA + Hibernate** - Persistencia
- **Liquibase** - Migraciones de base de datos
- **SpringDoc OpenAPI** - Documentación Swagger
- **Lombok** - Reducción de boilerplate
- **Datafaker** - Generación de datos de prueba

4.3 Frontend (clinica3s-frontend)

Estructura de Directorios

src/	
└─ main.jsx	# Punto de entrada
└─ App.jsx	# Componente raíz con rutas
└─ index.css	# Estilos globales
└─ api/	
└─ api.js	# Configuración Axios e interceptores
└─ context/	
└─ AuthContext.jsx	# Contexto de autenticación


```
├── components/
│   ├── layout/
│   │   └── MainLayout.jsx      # Layout principal con menú
│   ├── common/
│   │   └── ProtectedRoute.jsx  # Protección de rutas
│   ├── dashboard/
│   │   └── KpiCard.jsx         # Tarjeta de KPI reutilizable
│   └── appointments/
│       └── AppointmentDetailDialog.jsx # Dialog de detalles
├── pages/
│   ├── LoginPage.jsx          # Página de login
│   ├── HomePage.jsx           # Calendario de citas
│   ├── DashboardPage.jsx      # Dashboard de BI
│   ├── AppointmentsPage.jsx    # Listado de citas
│   ├── NewAppointmentPage.jsx  # Formulario nueva cita
│   ├── PatientsPage.jsx        # Gestión de pacientes
│   ├── DentistsPage.jsx        # Gestión de dentistas
│   ├── ServicesPage.jsx        # Catálogo de servicios
│   ├── SpecialtiesPage.jsx     # Gestión de especialidades
│   ├── UsersPage.jsx           # Gestión de usuarios
│   └── ProfilePage.jsx         # Perfil de usuario
├── services/                  # Servicios API
│   ├── authService.js
│   ├── appointmentService.js
│   ├── patientService.js
│   ├── dentistService.js
│   ├── serviceService.js
│   ├── specialtyService.js
│   ├── userService.js
│   └── reportService.js
└── utils/
    ├── constants.js           # Constantes y colores
    └── formatters.js           # Funciones de formateo
```

Vistas React Principales

Página	Ruta	Descripción
LoginPage	/login	Formulario de autenticación
HomePage	/	Calendario visual de citas del día
DashboardPage	/dashboard	KPIs y gráficas de BI
AppointmentsPage	/appointments	Listado filtrable de citas
NewAppointmentPage	/appointments/new	Formulario de nueva cita
PatientsPage	/patients	CRUD de pacientes
DentistsPage	/dentists	CRUD de dentistas
ServicesPage	/services	Catálogo de servicios

Página	Ruta	Descripción
SpecialtiesPage	/specialties	Gestión de especialidades
UsersPage	/users	Administración de usuarios
ProfilePage	/profile	Perfil y cambio de contraseña

Tecnologías Frontend

- **React 19** - Biblioteca UI
- **Vite 7** - Build tool
- **PrimeReact 10** - Componentes UI (DataTable, Dialog, Calendar, etc.)
- **PrimeFlex** - Sistema de rejilla CSS
- **React Router DOM 7** - Enrutamiento SPA
- **Axios** - Cliente HTTP con interceptores
- **Chart.js** - Gráficas interactivas
- **date-fns** - Manipulación de fechas
- **TailwindCSS** - Framework CSS utilitario

5. Instrucciones de Compilación y Uso

5.1 Requisitos Previos

- **Java 17** o superior
- **Maven 3.8+**
- **Node.js 18+** y npm
- **Git**

5.2 Instalación

```
# 1. Clonar el repositorio
git clone https://github.com/edconde/clinica3s.git
cd clinica3s

# 2. Ejecutar Backend
cd clinica3s-backend
./mvnw spring-boot:run
# Backend disponible en http://localhost:8080

# 3. Ejecutar Frontend (nueva terminal)
cd clinica3s-frontend
npm install
npm run dev
# Frontend disponible en http://localhost:5173
```

5.3 Acceso al Sistema

Credenciales por defecto:

- Usuario: `admin`
- Contraseña: `admin123`

URLs de desarrollo:

- Frontend: `http://localhost:5173`
- Backend API: `http://localhost:8080`
- Swagger UI: `http://localhost:8080/swagger-ui.html`
- H2 Console: `http://localhost:8080/h2-console`

5.4 Compilación para Producción

```
# Backend
cd clinica3s-backend
./mvnw clean package -DskipTests
# Genera: target/clinica3s-backend-0.0.1-SNAPSHOT.jar

# Frontend
cd clinica3s-frontend
npm run build
# Genera: dist/
```

5.5 Variables de Entorno (Producción)

Variable	Descripción
ENVIRONMENT	Perfil: <code>development</code> , <code>preproduction</code> , <code>production</code>
DATABASE_URL	URL de conexión PostgreSQL
DATABASE_USERNAME	Usuario de base de datos
DATABASE_PASSWORD	Contraseña de base de datos
JWT_SECRET	Clave secreta para tokens JWT
ADMIN_PASSWORD	Contraseña inicial del administrador
CORS_ALLOWED_ORIGINS	URLs del frontend permitidas

6. Conclusiones, Problemas y Mejoras

6.1 Conclusiones

El desarrollo de **Clinica3S** ha permitido implementar un sistema completo de gestión que aborda las necesidades reales de una clínica odontológica. Los principales logros incluyen:

- **Arquitectura robusta:** Separación clara entre frontend y backend con API REST bien definida
- **Seguridad implementada:** Autenticación JWT con control de acceso granular por roles

- **Experiencia de usuario:** Interfaz moderna y responsive con componentes profesionales (PrimeReact)
- **Business Intelligence:** Dashboard con métricas clave para la toma de decisiones
- **Código mantenible:** Arquitectura en capas, uso de DTOs, servicios desacoplados
- **Datos de prueba realistas:** Generación automática de datos para desarrollo y testing

6.2 Problemas Encontrados

Problema	Solución Aplicada
Complejidad del modelo de citas con múltiples servicios	Diseño de AppointmentDetail con precio histórico y control de pagos individual
Rendimiento en consultas de pacientes y citas	Implementación de paginación y filtros en backend
Gestión de estados de cita	Enums con transiciones controladas y validaciones
Sincronización de precios históricos	Campo priceApplied que preserve el precio en el momento de la cita
Control de acceso por rol	Middleware JWT con roles embebidos en el token

6.3 Posibles Mejoras Futuras

Funcionalidades Adicionales

- **Notificaciones:** Recordatorios de citas por email/SMS
- **Historial clínico:** Registro detallado de tratamientos por paciente
- **Facturación:** Generación de facturas PDF
- **Agenda de disponibilidad:** Configuración de horarios por dentista
- **Multi-clínica:** Soporte para gestionar varias sedes

Mejoras Técnicas

- **Testing:** Ampliar cobertura de tests unitarios e integración
- **Caché:** Implementar Redis para mejorar rendimiento
- **Logs estructurados:** Integración con ELK Stack para monitorización
- **CI/CD:** Pipeline automatizado de despliegue
- **PWA:** Convertir frontend en Progressive Web App para uso offline

Mejoras de UX

- **Modo oscuro:** Tema alternativo para la interfaz
- **Accesibilidad:** Mejoras WCAG para usuarios con discapacidades
- **Internacionalización:** Soporte multi-idioma
- **Dashboard personalizable:** Widgets configurables por usuario

6.4 Lecciones Aprendidas

1. **Diseño de base de datos:** La importancia de almacenar datos históricos (precios) para análisis posterior
 2. **Seguridad desde el inicio:** Implementar autenticación y autorización desde las primeras etapas
 3. **Componentes reutilizables:** El uso de bibliotecas como PrimeReact acelera significativamente el desarrollo
 4. **Documentación automática:** SpringDoc/Swagger facilita la integración frontend-backend
 5. **Datos de prueba:** Un buen generador de datos de prueba es esencial para el desarrollo
-

Documento elaborado: Enero 2026

Versión: 1.0