

Documentación del Sistema de Evaluación S-FMDRS

Fecha: 22 de Diciembre, 2025 **Versión:** 1.0.0

1. Introducción

1.1 Propósito

La aplicación **S-FMDRS Evaluation App** es una plataforma integral diseñada para la gestión, evaluación y seguimiento de pacientes con Trastornos de Movimiento Funcional (FMD). Digitaliza la Escala de Calificación del Trastorno del Movimiento Funcional Simplificada (S-FMDRS), permitiendo a los médicos registrar puntuaciones de severidad y duración en tiempo real, visualizar el progreso del paciente y asistir el diagnóstico mediante Inteligencia Artificial.

1.2 Alcance

El sistema abarca tres roles principales:

- **Médicos:** Realizan evaluaciones, gestionan tratamientos, visualizan estadísticas y validan análisis de IA.
- **Pacientes:** Visualizan su historial de evaluaciones, progreso y recomendaciones validadas.
- **Administradores:** Gestionan usuarios y supervisan el sistema global.

2. Arquitectura Técnica

El sistema sigue una arquitectura cliente-servidor moderna y desacoplada.

2.1 Stack Tecnológico

- **Frontend (Cliente):**
 - **Framework:** Next.js 14 (App Router)
 - **Lenguaje:** TypeScript
 - **Estilos:** Tailwind CSS
 - **Componentes UI:** Shadcn/UI (Radix Primitives)
 - **Gráficos:** Recharts
 - **Librerías Clave:** Axios, Lucide-React, Date-fns.
- **Backend (Servidor):**
 - **Framework:** NestJS
 - **Lenguaje:** TypeScript
 - **ORM:** Sequelize
 - **Base de Datos:** PostgreSQL
 - **Autenticación:** JWT (JSON Web Tokens) con Guards RBAC.
- **Inteligencia Artificial:**
 - **Orquestador:** N8N (Webhook)
 - **Modelos:** DeepSeek AI y Google Gemini.

2.2 Diagrama de Flujo de Datos (IA)

1. **Solicitud:** El médico solicita análisis desde el Frontend.
 2. **Procesamiento:** El Backend envía los datos anónimos de la evaluación al Webhook de N8N.
 3. **Generación:** N8N consulta a dos LLMs (DeepSeek y Gemini) en paralelo.
 4. **Almacenamiento:** Las respuestas se estructuran y guardan en PostgreSQL ([análisis_ia](#)).
 5. **Validación:** El médico selecciona la mejor respuesta y agrega feedback.
 6. **Visualización:** El paciente accede únicamente a la versión validada.
-

3. Módulos y Funcionalidades

3.1 Módulo de Autenticación

- **Login Seguro:** Validación de credenciales y generación de JWT.
- **Protección de Rutas:** Middleware para restringir acceso según rol ([ADMIN](#), [MEDICO](#), [PACIENTE](#)).

3.2 Módulo del Médico

- **Dashboard Clínico (KPIs):**
 - Total de pacientes y evaluaciones.
 - Puntaje promedio S-FMDRS global.
 - Distribución de severidad (Leve/Moderado/Severo).
 - **Región Crítica:** Identificación automática de la zona corporal más afectada.
 - Estadísticas de uso y preferencia de modelos de IA.
- **Gestión de Tratamientos:**
 - Creación de nuevos tratamientos (validación de tratamiento activo único).
 - Listado de pacientes asignados.
- **Evaluación S-FMDRS:**
 - Formulario interactivo de 9 regiones corporales.
 - Puntuación independiente de **Severidad (0-4)** y **Duración (0-4)**.
 - Cálculo automático de subtotales y total global (max 54).
- **Módulo de IA:**
 - Generación de análisis comparativos.
 - Interfaz de selección y caja de texto para feedback médico.

3.3 Módulo del Paciente

- **Dashboard Personal:**
 - Estado del tratamiento actual.
 - Gráfico de progreso histórico.
- **Historial:** Acceso de lectura a todas las evaluaciones pasadas.
- **Análisis IA:** Vista simplificada del informe validado por su médico, con recomendaciones terapéuticas claras.

3.4 Módulo de Administración

- **Gestión de Usuarios:**
 - Crear, editar y desactivar usuarios.
 - Asignación de roles y especialidades médicas.
- **Visión Global:** Estadísticas de todo el sistema.

4. Estructura de Base de Datos (Schema)

Entidades principales en PostgreSQL:

- **Usuario**: Almacena credenciales, roles y datos personales.
- **Tratamiento**: Relaciona `medico_id` y `paciente_id`. Controla el estado (`ACTIVO`, `TERMINADO`). Atributos meta (`meta_objetivo`).
- **Cuestionario**: Representa una sesión de evaluación. Vinculada a un **Tratamiento**. Almacena `fecha`, `puntaje_total`, `observaciones`.
- **RespuestaRegion**: Detalle granular. 9 registros por cada Cuestionario. Almacena `severidad`, `duracion`, `nombre_region`.
- **AnalisisIa**: Almacena las respuestas de los modelos (`contenido`), el proveedor (`modelo`), y datos de validación (`seleccionado`, `feedback_medico`).

5. Guía de Instalación y Despliegue

5.1 Requisitos Previos

- Node.js v18+
- PostgreSQL v14+
- NPM o Yarn

5.2 Configuración del Backend

1. Navegar a `app-backend`.
2. Instalar dependencias: `npm install`.
3. Configurar variables de entorno en `.env`:

```
DB_HOST=localhost
DB_PORT=5432
DB_USERNAME=postgres
DB_PASSWORD=secret
DB_DATABASE=s_fmdrs_db
JWT_SECRET=clave_super_secreta
N8N_WEBHOOK_URL=https://...
```

4. Iniciar servidor: `npm run dev` (Puerto predeterminado: 4000).

5.3 Configuración del Frontend

1. Navegar a `app-frontend-web`.
2. Instalar dependencias: `npm install`.
3. Configurar variables de entorno en `.env.local`:

```
NEXT_PUBLIC_API_URL=http://localhost:4000/api
```

4. Iniciar servidor: `npm run dev` (Puerto predeterminado: 3000).

5.4 Exportación a PDF

Este documento está escrito en Markdown. Para generar el PDF final:

1. Abrir este archivo en VS Code.
 2. Usar una extensión como "Markdown PDF".
 3. Ejecutar comando: `Convert Markdown to PDF`.
-