

## Documento de Requisitos Funcionales (RF)

### 1. Definición

Los **Requisitos Funcionales (RF)** definen las funcionalidades y el comportamiento que el sistema debe ejecutar. Cada requisito ha sido catalogado basándose en los **Endpoints de la API** y la interfaz de usuario implementada, asegurando la trazabilidad.

### 2. Requisitos Funcionales por Categoría

#### 2.1. Categoría 1: Autenticación y Acceso

RF ID	Descripción Detallada	Referencia de Implementación	Actor Principal
RF-01	<b>Registro de Nuevo Usuario:</b> Permitir el registro de nuevos usuarios, asignando por defecto el rol de <b>paciente</b> .	POST /api/auth/register <sup>1</sup> / Lógica en backend/src/controllers/authController.js <sup>2</sup> .	Todos
RF-02	<b>Inicio de Sesión Seguro:</b> Autenticar al usuario mediante email/contraseña, verificando credenciales y generando un <b>token JWT</b> .	POST /api/auth/login <sup>3</sup> / Route: backend/src/routes/auth.routes.js <sup>4</sup> .	Todos
RF-03	<b>Obtención de Perfil de Sesión:</b> Obtener los datos del usuario autenticado (Nombre, Rol, Email) para la personalización de la interfaz.	GET /api/auth/me <sup>5</sup> . Usado por authAPI.getMe() en web/js/api.js <sup>6</sup> .	Todos
RF-04	<b>Redirección por Rol y Protección:</b> Redirigir al <i>Dashboard</i> correspondiente y proteger todas las rutas privadas mediante verificación de token y rol.	Función redirectToDashboard() y protectPage() en web/js/api.js <sup>7</sup> .	Todos
RF-05	<b>Restablecimiento de Contraseña:</b> Permitir al usuario solicitar un restablecimiento de contraseña mediante email o tarjeta sanitaria.	Interfaz implementada en web/password_recovery.html.	Todos

## 2.2. Categoría 2: Gestión de Citas y Disponibilidad

RF ID	Descripción Detallada	Referencia de Implementación	Actor Principal
RF- 06	<b>Búsqueda de Disponibilidad:</b> Permitir la búsqueda por <b>Especialidad, Centro Médico y Médico</b> para la reserva.	Interfaz en web/book_new_appointment.html. Se apoya en: GET /api/centers <sup>8</sup> .	Paciente
RF- 07	<b>Listado de Médicos Filtrado:</b> Listar <i>Médicos</i> disponibles mediante filtro de especialidad.	GET /api/users/doctors?specialty =X <sup>9</sup> . Función: usersAPI.listDoctors en web/js/api.js <sup>10</sup> .	Paciente
RF- 08	<b>Creación de Cita:</b> Crear una nueva cita, registrando todos los datos necesarios (profesional, centro, fecha, hora).	POST /api/appointments <sup>11</sup> . Función: appointmentsAPI.create en web/js/api.js <sup>12</sup> .	Paciente
RF- 09	<b>Listado de Citas (Paciente):</b> Listar todas las citas asociadas al <i>Paciente</i> autenticado.	GET /api/appointments <sup>13</sup> (filtrado por <i>patientId</i> ). Lógica <i>frontend</i> en web/js/modules/patient-dashboard.js.	Paciente
RF- 10	<b>Cancelación de Cita (Paciente):</b> Permitir al <i>Paciente</i> eliminar una cita pendiente desde su <i>Dashboard</i> .	DELETE /api/appointments/:id <sup>14</sup> . Función: appointmentsAPI.cancel en web/js/api.js <sup>15</sup> .	Paciente
RF- 11	<b>Cierre de Cita (Médico):</b> Permitir al <i>Médico</i> cambiar el estado de una cita a <b>completada</b> .	PATCH /api/appointments/:id/status <sup>16</sup> . Lógica en web/js/modules/doctor-dashboard.js.	Médico

### 2.3. Categoría 3: Administración y Mantenimiento (CRUD)

RF ID	Descripción Detallada	Referencia de Implementación	Actor Principal
RF-12	<b>Listado de Usuarios (Admin):</b> Mostrar una tabla con <b>todos los usuarios</b> del sistema para su gestión.	GET /api/users <sup>17</sup> . Módulo: web/js/modules/administrator-dashboard.js.	Administrador
RF-13	<b>Creación de Usuarios (Admin):</b> Permitir crear nuevos usuarios con asignación de <b>roles y especialidades</b> .	POST /api/users <sup>18</sup> . Lógica en web/js/modules/administrator-dashboard.js.	Administrador
RF-14	<b>Edición de Usuarios (Admin):</b> Modificar los datos de cualquier usuario (nombre, email, rol, etc.).	PUT /api/users/:id <sup>19</sup> . Función: usersAPI.update en web/js/api.js <sup>20</sup> .	Administrador
RF-15	<b>Eliminación de Usuarios (Admin):</b> Eliminar usuarios del sistema.	DELETE /api/users/:id <sup>21</sup> . Función: usersAPI.delete en web/js/api.js <sup>22</sup> .	Administrador
RF-16	<b>CRUD de Centros (Admin): Crear, Editar y Eliminar</b> centros médicos.	POST, PUT, DELETE /api/centers/:id <sup>23</sup> .	Administrador
RF-17	<b>Toggle de Estado (Centros):</b> Cambiar el estado de un <i>Centro Médico</i> a <i>Activo</i> o <i>Inactivo</i> .	PATCH /api/centers/:id/status <sup>24</sup> . Función: centersAPI.toggleStatus en web/js/api.js <sup>25</sup> .	Administrador
RF-18	<b>Búsqueda en Tablas (Frontend):</b> Las tablas de <i>Usuarios</i> y <i>Centros</i> deben permitir la <b>búsqueda en tiempo real</b> por nombre, email o rol.	Lógica de filtrado y renderizado en web/js/modules/administrator-dashboard.js.	Administrador

#### 2.4. Categoría 4: Auxiliares y Experiencia de Usuario (RF-19 a RF-25)

RF ID	Descripción Detallada	Referencia de Implementación	Actor Principal
RF-19	<b>Gestión de Órdenes Médicas:</b> El Médico debe poder <b>crear y listar órdenes médicas</b> para sus pacientes.	Interfaz implementada en web/doctor_dashboard.html (Panel "Órdenes Médicas").	Médico
RF-20	<b>Mensajería Interna (Chat):</b> Proporcionar una <b>interfaz de chat</b> para la comunicación Paciente-Médico.	Interfaz en web/_chat.html y web/doctor_dashboard.html.	Paciente, Médico
RF-21	<b>Centro de Notificaciones:</b> Mostrar un centro de notificaciones con información sobre citas, recordatorios y alertas.	Interfaz implementada en web/notification_center.html.	Todos
RF-22	<b>Gestión de Pagos Online:</b> Procesar los <b>pagos online</b> de las citas mediante diversos métodos (Tarjeta, PayPal, Transferencia).	Interfaz implementada en web/online_payment_screen.html.	Paciente
RF-23	<b>Cálculo de Pagos Detallado:</b> Mostrar el resumen de pagos, incluyendo subtotal, IVA (21%), y la aplicación de descuentos por códigos y puntos de fidelidad.	Lógica de cálculo y descuentos en web/online_payment_screen.html.	Paciente
RF-24	<b>Sistema de Puntos y Recompensas:</b> Mostrar el <b>nivel de fidelidad y puntos disponibles</b> del Paciente para canje por descuentos.	Lógica <i>frontend</i> en web/js/modules/patient-dashboard.js.	Paciente
RF-25	<b>Panel de Métricas (Analytics):</b> Acceso a un panel de <i>Analytics</i> que muestre estadísticas clave del sistema.	Interfaz implementada en web/healthcare_analytics_dashboard.html.	Administrador