

Requerimientos FURPS+ para un Sistema Médico

Funcionales (F)

1. Validación de usuario:

- o El usuario debe ingresar su nombre de usuario y contraseña.
- o Al usuario registrado se enviará un código para una autenticación 2FA.
- o La implementación de **2FA** será por **JavaMailSender** y genera un código aleatorio en el backend.
- o El sistema permitirá al usuario ya autenticado identificarse sin mantener sesiones en el servidor con **JWT (Json Web Tokens)**.

2. Gestión de pacientes:

- o Los datos de los pacientes registrados son:
 - Nombre, Apellidos, Fecha de Nacimiento, Género.
 - Dirección, Teléfono de contacto, Correo electrónico.
 - Contacto y Teléfono de emergencia, Enfermedades crónicas, Alergias conocidas.
- o El paciente puede:
 - Visualizar su información personal y su historial médico, así como modificar algunos parámetros de contacto como correo, teléfono, número de emergencia.
 - Programar citas (sujeto a disponibilidad) con opción de reprogramación y cancelación.
 - Recibir notificaciones de la agendación de citas.
- o El personal médico (asociado a ese paciente) puede:
 - Visualizar su información general del paciente y su historial médico.
 - Consultar y actualizar historiales clínicos.

3. Gestión de personal médico:

- o Los datos del personal médico registrados son:
 - Nombre, Apellidos, Fecha de Nacimiento, Género, Cédula.
 - Dirección, Teléfono de contacto, Correo electrónico.
 - Contacto y Teléfono de emergencia.
- o El personal médico puede:
 - Visualizar su información personal, así como modificar algunos parámetros de contacto.
 - Visualizar sus citas médicas agendadas en su horario de trabajo.
 - Recibir notificaciones de la agendación de citas.
- o La administración puede:
 - Dar de alta, baja, cambios asociados a personal médico y paciente.

4. Solicitudes de ausencias:

- o Permitir al personal médico realizar solicitudes de ausencias con:
 - Fecha de inicio y fin del permiso.
 - Descripción de la causa.
- o La Administración autorizará las solicitudes.
- o El personal médico y administración podrá visualizar el estado de la solicitud (pendiente, aprobada, rechazada).

5. Consultas médicas:

- o Los datos de registro de consulta son:
 - Síntomas del paciente, medicamentos y dosis prescritos.
 - Fecha de próxima consulta.
- o El personal médico (asociado a esa consulta de paciente) puede:
 - Registrar diagnósticos y prescripciones médicas.
 - Gestionar citas (sujeto a disponibilidad de horario) y registrar consultas.
 - Programación de seguimiento médico (sujeto a disponibilidad).
- o La administración puede:
 - Registro y edición de consultas médicas.
 - Carga y gestión de documentos de consentimiento informado.

6. Avisos (Notificaciones):

- o Visualizar notificaciones ordenadas cronológicamente, con:
 - Fecha, Título, y Descripción del aviso.

7. Interacción:

- o **Chat integrado:** Comunicación en tiempo real entre personal médico y paciente.
- o **Sistema de tickets:**
 - Crear, asignar y gestionar solicitudes (consultas específicas y problemas técnicos)
 - Seguimiento de estados de ticket: abierto, en progreso, resuelto.

8. Cumplimiento normativo:

- o Incorporar lineamientos de normas aplicables: NOM-004-SSA3-2012 , SO 27001*, y Ley Federal de Protección de Datos Personales en Posesión de Particulares (LFPDPPP).

9. Consentimiento informado:

- o Los médicos podrán subir documentos de consentimiento informado, detallando los procedimientos a realizar.
- o Los pacientes podrán visualizar los consentimientos asociados a sus consultas.
- o Utilizar bases de datos orientadas a documentos para almacenar y gestionar estos consentimientos, permitiendo búsquedas rápidas y acceso eficiente.

10. Endpoints REST: Proveer un conjunto de endpoints REST para interactuar con el sistema, como:

- o **Pacientes:**
 - POST /api/patients: Registrar un nuevo paciente.
 - GET /api/patients: Consultar la lista de pacientes.
 - GET /api/patients/{id}: Obtener información detallada de un paciente.
 - PUT /api/patients/{id}: Editar información de un paciente.
 - DELETE /api/patients/{id}: Eliminar un paciente.
- o **Consultas:**
 - POST /api/consultations: Registrar una nueva consulta médica.
 - GET /api/consultations: Consultar todas las consultas.

- GET /api/consultations/{id}: Obtener detalles de una consulta específica.
 - o **Tickets:**
 - POST /api/tickets: Crear un nuevo ticket.
 - GET /api/tickets: Listar tickets generados.
 - GET /api/tickets/{id}: Consultar el estado y detalles de un ticket.
 - PUT /api/tickets/{id}: Actualizar el estado de un ticket.
-

Usabilidad (U)

1. **Interfaz amigable:**
 - o Botones y etiquetas descriptivas con un diseño limpio y moderno.
 - o Uso de colores que representen estados claros (verde para éxito, rojo para errores).
 2. **Flujo intuitivo:**
 - o Menús jerárquicos que permitan acceder rápidamente a cualquier funcionalidad.
 - o Tutorial interactivo para los usuarios al iniciar sesión por primera vez.
 3. **Acceso multiplataforma:**
 - o Compatibilidad con navegadores.
 4. **Manual de usuario:**
 - o Incluir guías con capturas de pantalla y pasos detallados.
 5. **Accesibilidad:**
 - o Texto escalable, compatibilidad con lectores de pantalla, responsive.
-

Confiabilidad (R)

1. **Respaldo automático:**
 - o Copias de seguridad cada cierto periodo de tiempo, con opciones de recuperación.
 2. **Gestión de errores:**
 - o Informes detallados en caso de fallas, con tickets generados automáticamente.
 3. **Autenticación segura:**
 - o Contraseñas cifradas, autenticación 2FA, recuperación de contraseña mediante un correo verificado.
-

Rendimiento (P)

1. **Optimización:**
 - o Todas las acciones críticas (como guardar datos) deben ejecutarse en menos de 3 segundos.
 - o Optimización de sesiones en el servidor implementando JWT, así evitar almacenar información de autenticación en una base de datos o memoria.
2. **Escalabilidad:**
 - o Capacidad para manejar al menos 10,000* usuarios concurrentes sin pérdida de rendimiento.

3. **Eficiencia:**

- o Carga rápida de pantallas y consultas al historial médico.
-

Soporte (S)

1. **Compatibilidad:**

- o Garantizar el funcionamiento en sistemas operativos actuales como Windows 10+ y navegadores modernos.

2. **Mantenimiento:**

- o Actualizaciones frecuentes para mejorar seguridad y funcionalidad.

3. **Sistema de soporte:**

- o Gestión de tickets para solicitudes y problemas reportados por los usuarios.

4. **Documentación técnica:**

- o Manuales detallados para desarrolladores y usuarios avanzados.
-

Extensiones (+)

1. **Analítica avanzada:**

- o Generar reportes automáticos basados en datos médicos para análisis y seguimiento.

2. **Cumplimiento legal:**

- o Proteger los datos sensibles conforme a normativas como la Ley Federal de Protección de Datos Personales en Posesión de Particulares (LFPDPPP).