

1. Introducción

1.1 Propósito

Este documento describe los requisitos funcionales y no funcionales para una aplicación móvil de gestión de citas médicas. El sistema será distribuido con varias APIs que permitirán la interacción entre pacientes y médicos para agendar citas, realizar pagos, gestionar perfiles y chatear. El frontend será una aplicación móvil hecha en React Native.

1.2 Alcance

El sistema permitirá a los pacientes buscar médicos por especialidad y ubicación, gestionar citas, pagar servicios, y mantener un chat con los médicos. Por su parte, los médicos podrán gestionar su perfil, sus horarios y citas, y habilitar o deshabilitar su disponibilidad. El backend estará compuesto por múltiples APIs (REST y SOAP) para manejar las funcionalidades necesarias, integradas a través de una API Gateway.

1.3 Definiciones, acrónimos y abreviaturas

- **API:** Interfaz de programación de aplicaciones.
- **REST:** Representational State Transfer.
- **SOAP:** Simple Object Access Protocol.
- **Gateway:** Punto de acceso centralizado para las APIs.
- **Frontend:** Interfaz de usuario.
- **Backend:** Lógica de negocio y manejo de datos en el servidor.

1.4 Referencias

- Documentación de React Native.
 - Estándares para la creación de APIs REST y SOAP.
-

2. Descripción General

2.1 Perspectiva del Producto

La aplicación es un sistema distribuido compuesto de una aplicación móvil conectada a un backend que incluye varias APIs distribuidas. La API Gateway centraliza las solicitudes del frontend y las distribuye a las APIs individuales que manejan las funcionalidades específicas.

2.2 Funcionalidades Principales

- **Búsqueda de médicos:** Filtrar médicos por especialidad, estado del país y nombre.

- **Gestión de citas:** Agendar, cancelar y reagendar citas.
- **Pago de citas:** Realizar pagos por los servicios médicos.
- **Perfil de usuario:** Configuración y gestión de perfiles tanto de pacientes como de médicos.
- **Chat:** Chat entre el médico y el paciente cuando la cita haya sido agendada.
- **Horarios médicos:** Los médicos pueden establecer, bloquear o desbloquear horas de trabajo.

2.3 Usuarios del Sistema

- **Paciente:** Persona que busca y agenda citas con médicos.
- **Médico:** Profesional que ofrece sus servicios a través de la plataforma.

2.4 Suposiciones y Dependencias

- La conexión entre el frontend y el backend se realizará exclusivamente a través de la API Gateway.
 - La API SOAP se encargará de gestionar las especialidades médicas.
-

3. Requisitos Funcionales

3.1 Registro y Login

- **RF1:** El sistema debe permitir que los usuarios (pacientes y médicos) se registren utilizando su correo electrónico y contraseña.
- **RF2:** El sistema debe proporcionar una API REST para manejar el login de los usuarios.

3.2 Gestión de Perfiles

- **RF3:** Los usuarios podrán configurar su perfil personal, incluyendo nombre, correo, teléfono, dirección, foto de perfil y otros detalles.
- **RF4:** Los médicos podrán agregar información sobre su especialidad y experiencia.

3.3 Búsqueda de Médicos

- **RF5:** La API SOAP debe proporcionar una lista de especialidades médicas.
- **RF6:** El sistema debe permitir filtrar médicos por especialidad, estado del país y nombre.

3.4 Gestión de Citas

- **RF7:** Los pacientes podrán agendar, cancelar y reagendar citas médicas.
- **RF8:** Los médicos podrán ver su agenda de citas y bloquear horas específicas.

3.5 Pagos

- **RF9:** El sistema debe permitir realizar pagos por las citas agendadas mediante una API REST.

3.6 Chat Médico-Paciente

- **RF10:** El sistema debe permitir que los pacientes chateen con los médicos solo cuando hayan agendado una cita.

3.7 API Gateway

- **RF11:** La API Gateway será el punto de acceso para todas las solicitudes del frontend.
 - **RF12:** La API Gateway debe gestionar la autenticación y autorización de los usuarios.
-

4. Requisitos No Funcionales

4.1 Seguridad

- **RNF1:** El sistema debe garantizar la seguridad de los datos de los usuarios mediante protocolos de cifrado SSL/TLS.
- **RNF2:** Autenticación basada en JWT (JSON Web Token) para todas las APIs REST.

4.2 Rendimiento

- **RNF3:** Las respuestas de las APIs deben tener un tiempo de respuesta inferior a 500 ms en condiciones normales de carga.
- **RNF4:** La API Gateway debe ser escalable para soportar miles de usuarios simultáneos.

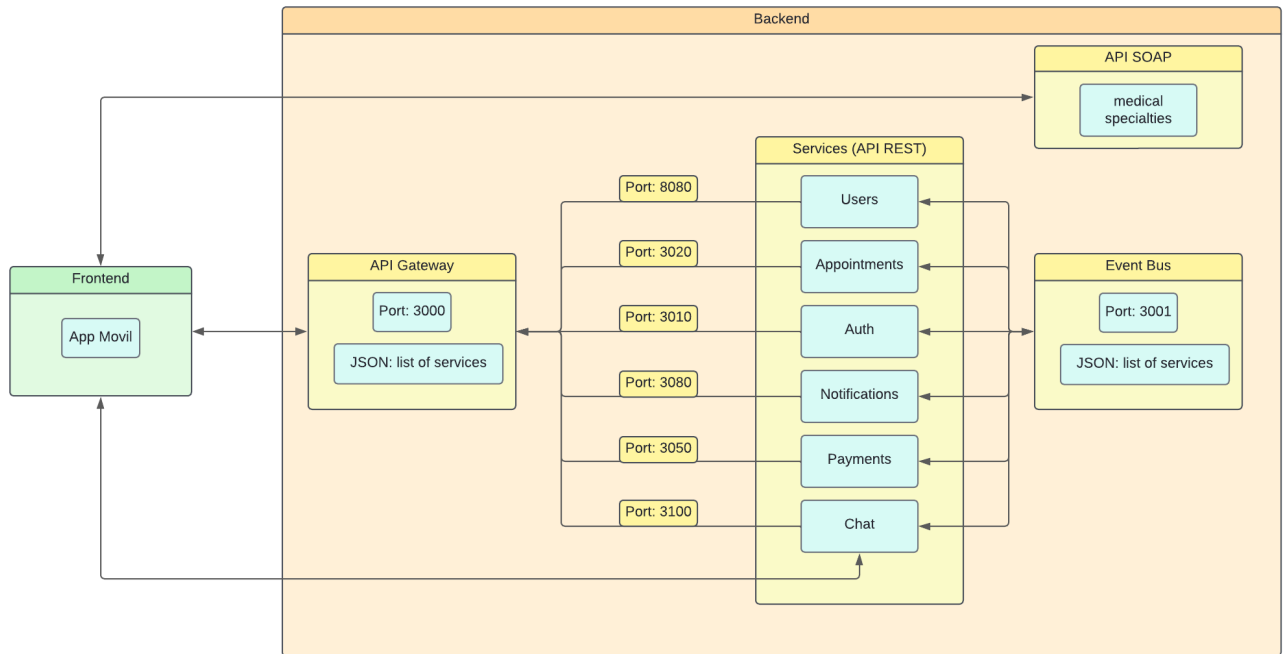
4.3 Disponibilidad

- **RNF5:** El sistema debe estar disponible al 99.9% del tiempo.

4.4 Compatibilidad

- **RNF6:** El sistema debe ser compatible con dispositivos iOS y Android.
-

5. Arquitectura del Sistema



5.1 Estilo Arquitectónico

El sistema estará diseñado bajo una arquitectura de **microservicios**, donde cada funcionalidad principal de la aplicación será gestionada por un microservicio independiente. Los microservicios se comunicarán entre sí y con la API Gateway de forma asincrónica (cuando sea necesario) y síncrona para ciertas operaciones críticas.

5.2 Componentes

- **Microservicio de Especialidades Médicas (SOAP):** Gestiona las especialidades médicas disponibles en el sistema.
- **Microservicio de Usuarios (REST):** Gestiona el registro, autenticación, login y perfil de usuarios (pacientes y médicos).
- **Microservicio de Citas (REST):** Maneja la creación, modificación y cancelación de citas médicas.
- **Microservicio de Pagos (REST):** Gestiona los pagos de las citas.
- **Microservicio de Chat (REST):** Permite la comunicación entre médicos y pacientes una vez agendada la cita.
- **API Gateway:** Orquesta las comunicaciones entre el frontend y los microservicios. Actúa como un único punto de acceso, facilitando la escalabilidad, seguridad y control del tráfico.

5.3 Diagrama de Arquitectura

(Un diagrama aquí mostraría la interacción entre el frontend y las diferentes APIs a través de la API Gateway). Esto me falta

6. Casos de Uso

Caso de Uso 1: Registro de Paciente

1. El paciente abre la aplicación móvil.
2. Ingresa sus datos personales (nombre, correo, contraseña).
3. El sistema valida la información y registra al usuario.

Caso de Uso 2: Búsqueda de Médico

1. El paciente abre la sección de búsqueda.
2. Elige la especialidad y el estado del país o ingresa el nombre del médico.
3. El sistema muestra los médicos disponibles.

Caso de Uso 3: Agendar una Cita

1. El paciente selecciona un médico.
2. Elige la fecha y hora disponible.
3. El sistema agenda la cita.

(Continuar con más casos de uso...)

7. Requisitos de Implementación

7.1 Tecnologías

- **Frontend:** React Native.
- **Backend:** APIs distribuidas en REST y SOAP.
- **API Gateway:** Basada en Node.js

8. Normas legales a seguir

Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP)

Puntos Clave:

1. **Datos Sensibles:** La LFPDPPP clasifica los datos médicos como datos personales sensibles, lo que significa que requieren un mayor nivel de protección.
2. **Consentimiento Expreso:** Para tratar datos sensibles (como los médicos), es obligatorio obtener el consentimiento explícito y por escrito de los usuarios, salvo algunas excepciones (emergencias médicas, por ejemplo).

3. **Aviso de Privacidad:** Debes informar claramente a los usuarios mediante un aviso de privacidad sobre el uso, tratamiento, y finalidad de los datos que se recopilan. Este aviso debe ser fácil de entender y estar disponible antes de que los datos sean recolectados.
4. **Derechos ARCO:** Los usuarios tienen derecho de **Acceso, Rectificación, Cancelación y Oposición** (ARCO) respecto a sus datos. Esto significa que deben poder consultar sus datos, corregirlos, solicitar su eliminación o limitar su uso.
5. **Medidas de Seguridad:** La ley requiere la implementación de medidas de seguridad administrativas, técnicas y físicas para proteger los datos personales de accesos no autorizados o alteraciones. Esto incluye cifrado de datos, control de acceso, y auditorías de seguridad.
6. **Responsabilidad:** Las organizaciones que recolectan y manejan datos personales son responsables de su protección y deben garantizar la confidencialidad, integridad y disponibilidad de los datos.
7. **Transferencia de Datos:** Si los datos personales son transferidos a terceros, es necesario contar con el consentimiento del usuario y tomar las medidas para que estos terceros también protejan adecuadamente los datos.

Relación con Normas Internacionales

- Aunque México no está obligado a seguir **HIPAA** o **GDPR** en su totalidad, muchas organizaciones que operan internacionalmente adoptan estos estándares como buenas prácticas adicionales.
- **ISO 27001** e **ISO 27799** son adoptados en México como buenas prácticas para asegurar la protección de datos, sobre todo en empresas que buscan certificaciones internacionales.

Norma Oficial Mexicana NOM-024-SSA3-2012

Además de la LFPDPPP, en el ámbito de la salud, México también cuenta con la **NOM-024-SSA3-2012** que establece los requisitos para la interoperabilidad de sistemas de información en salud, la seguridad y la confidencialidad de la información clínica.

Puntos Clave:

1. **Interoperabilidad:** Facilita que los sistemas de información en salud puedan intercambiar datos entre sí, garantizando la continuidad de los servicios médicos.
2. **Confidencialidad:** Esta norma refuerza la necesidad de resguardar adecuadamente la información clínica del paciente, tanto en su almacenamiento como en su transferencia entre sistemas médicos.

Resumen de Requisitos Clave para Aplicaciones Médicas en México:

1. **Consentimiento informado:** Es necesario obtener el consentimiento expreso del usuario antes de recolectar datos médicos.

2. **Aviso de privacidad claro:** Informar cómo se tratarán los datos y los derechos ARCO del usuario.
3. **Medidas de seguridad robustas:** Implementar cifrado, control de acceso y monitoreo para proteger los datos sensibles.
4. **Cumplimiento de la LFPDPPP:** Garantizar que se cumpla con la ley nacional en cuanto al manejo de datos personales sensibles.