

Documento de Requerimientos del Sistema

Sistema de Reservas - Soumaya Beauty Bar

Versión: 1.0

Fecha: 02 de Febrero de 2026

Proyecto: Sistema de Gestión de Citas con WhatsApp y Google Calendar

1. INTRODUCCIÓN

1.1 Propósito

Este documento especifica los requerimientos funcionales y no funcionales del Sistema de Reservas para Soumaya Beauty Bar, que permite a las clientas agendar citas de servicios de belleza, recibir confirmaciones por WhatsApp, gestionar cancelaciones mediante encuestas interactivas, y sincronizar automáticamente los eventos con Google Calendar.

1.2 Alcance

El sistema incluye:

- Gestión de usuarios (registro, login, autenticación JWT)
- Agendamiento de citas con selección de servicio, fecha y hora
- Integración con WhatsApp vía Twilio Sandbox para notificaciones y encuestas
- Integración con Google Calendar API mediante Service Account
- Sistema de encuestas automatizadas por WhatsApp (cancelación y reagendamiento)
- Panel de administración para visualizar reservas
- Recordatorios automáticos 24 horas antes de la cita

1.3 Definiciones, Acrónimos y Abreviaturas

- **API:** Application Programming Interface
 - **JWT:** JSON Web Token
 - **CRUD:** Create, Read, Update, Delete
 - **MX:** México
 - **SMS:** Short Message Service (incluye WhatsApp en este contexto)
 - **WABA:** WhatsApp Business Account
 - **SLA:** Service Level Agreement
-

2. DESCRIPCIÓN GENERAL

2.1 Perspectiva del Producto

Sistema web full-stack que automatiza el proceso de agendamiento de citas en un salón de belleza, eliminando la necesidad de gestión manual por llamadas o mensajes directos.

2.2 Funciones del Producto

1. Registro y autenticación de usuarias
2. Visualización de servicios disponibles con precios y duración
3. Agendamiento de citas con verificación de disponibilidad en tiempo real
4. Notificación automática por WhatsApp al agendar
5. Encuesta de cancelación interactiva por WhatsApp
6. Encuesta de reagendamiento tras cancelación
7. Sincronización bidireccional con Google Calendar
8. Recordatorios automáticos programados
9. Notificaciones al salón sobre nuevas citas y cancelaciones

2.3 Características de los Usuarios

- **Cientas:** Usuarios finales que agendan citas. Requieren cuenta registrada.
- **Administrador del Salón:** Visualiza calendario y recibe notificaciones WhatsApp.
- **Sistema:** Procesos automatizados (webhooks, cron jobs, APIs).

2.4 Restricciones

- Uso de Twilio Sandbox para WhatsApp (limitación de producción hasta activar WABA)
- Horario de atención: 10:00 AM - 8:00 PM
- Zona horaria: America/Mexico_City (GMT-6)
- Se requiere número de teléfono mexicano (+52) para registro
- Google Calendar Service Account con permisos de escritura

2.5 Suposiciones y Dependencias

- Las clientas tienen acceso a smartphone con WhatsApp instalado
- El salón mantiene actualizado su calendario de Google
- Twilio Sandbox está activo y configurado
- Conexión a internet estable en servidor backend
- MongoDB como base de datos

3. REQUERIMIENTOS FUNCIONALES

RF-01: Gestión de Usuarios

RF-01.1 Registro de Usuario

Prioridad: Alta

Descripción: El sistema debe permitir a nuevas usuarias crear una cuenta.

Criterios de aceptación:

- Campos requeridos: nombre completo, teléfono (+52), email, contraseña
- Validación de formato de teléfono: 10 dígitos después del código de país
- Validación de email único en el sistema
- Contraseña mínimo 6 caracteres
- Hash de contraseña con bcrypt (salt rounds: 10)
- Creación de token JWT al registrarse exitosamente
- Mensaje de bienvenida mostrado tras registro

Entrada:

```
{
  "nombreCompleto": "María González",
  "telefono": "3511270276",
  "email": "maria@ejemplo.com",
  "contraseña": "mipassword123"
}
```

Salida:

```
{
  "token": "eyJhbGc...",
  "usuario": {
    "_id": "507f1f77bcf86cd799439011",
    "nombreCompleto": "María González",
    "telefono": "3511270276",
    "email": "maria@ejemplo.com"
  }
}
```

RF-01.2 Inicio de Sesión

Prioridad: Alta

Descripción: El sistema debe autenticar usuarios existentes.

Criterios de aceptación:

- Login mediante email y contraseña
- Verificación de credenciales contra base de datos
- Generación de token JWT con expiración de 30 días

- Mensaje de error descriptivo si credenciales incorrectas
- Rate limiting: máximo 5 intentos fallidos por IP en 15 minutos

Entrada:

```
{  
  "email": "maria@ejemplo.com",  
  "contraseña": "mipassword123"  
}
```

Salida:

```
{  
  "token": "eyJhbGc...",  
  "usuario": { /* datos del usuario */ }  
}
```

RF-01.3 Protección de Rutas

Prioridad: Alta

Descripción: Solo usuarios autenticados pueden acceder a funcionalidades de reserva.

Criterios de aceptación:

- Middleware de autenticación verifica token JWT en header **Authorization: Bearer <token>**
 - Retorna HTTP 401 si token inválido o expirado
 - Retorna HTTP 403 si token válido pero sin permisos
 - Adjunta información de usuario a **req.user** para uso posterior
-

RF-02: Gestión de Servicios

RF-02.1 Catálogo de Servicios

Prioridad: Alta

Descripción: El sistema debe mostrar los servicios disponibles del salón.

Servicios definidos:

ID	Nombre	Duración	Precio
unas-gel	Uñas de Gel	60 min	\$450 MXN
unas-acrilicas	Uñas Acrílicas	90 min	\$600 MXN

pedicure	Pedicure Premium	90 min	\$500 MXN
keratina	Tratamiento de Keratina	180 min	\$1200 MXN
tinte	Tinte Profesional	180 min	\$800 MXN
pestanas	Extensión de Pestaña	60 min	\$900 MXN
cejas	Diseño de Cejas	30 min	\$350 MXN

Criterios de aceptación:

- Información visible en interfaz de usuario antes de iniciar reserva
 - Incluye emoji visual por servicio para mejor UX
 - Duración y precio claramente indicados
-

RF-03: Agendamiento de Citas

RF-03.1 Selección de Servicio, Fecha y Hora

Prioridad: Alta

Descripción: La usuaria debe poder seleccionar servicio, fecha y hora en un flujo de 4 pasos.

Flujo de pasos:

1. **Paso 1:** Selección de servicio (grid de tarjetas con emoji, nombre, duración y precio)
2. **Paso 2:** Selección de fecha (calendario semanal navegable)
3. **Paso 3:** Selección de hora (slots de 30 min desde 10:00 AM hasta 19:30)
4. **Paso 4:** Confirmación y creación de reserva

Criterios de aceptación:

- No se puede avanzar sin completar paso actual
- Indicador visual de progreso (1/4, 2/4, 3/4, 4/4)
- Botón "Atrás" para regresar a paso anterior sin perder datos
- Fechas pasadas aparecen deshabilitadas
- Horarios ocupados aparecen marcados como "Ocupada"
- Horarios que excedan las 20:00 según duración del servicio no son seleccionables

RF-03.2 Verificación de Disponibilidad

Prioridad: Crítica

Descripción: El sistema debe verificar disponibilidad antes de confirmar reserva.

Criterios de aceptación:

- Query a base de datos de reservas existentes con estado **confirmada**

- Verificación de solapamiento de horarios considerando duración del servicio
- Bloqueo de horario si:
 - Ya existe reserva en ese horario
 - El servicio terminaría después de las 20:00
 - La fecha es anterior al día actual
- Mensaje de error claro: "El horario ya está ocupado"

Algoritmo de solapamiento:

```
const solapamiento = (inicioA, finA, inicioB, finB) => {
  return inicioA < finB && finA > inicioB;
};
```

RF-03.3 Creación de Reserva

Prioridad: Crítica

Descripción: Al confirmar, el sistema debe crear la reserva en MongoDB, Google Calendar y generar deep link de WhatsApp.

Proceso secuencial:

1. Validar datos de entrada
2. Verificar disponibilidad (última validación en servidor)
3. Generar `cancelToken` único con crypto (32 bytes hex)
4. Crear registro en MongoDB con estado `confirmada` y `estadoEncuesta: 'pendiente_conexion'`
5. Crear evento en Google Calendar con Service Account
6. Almacenar `googleCalendarEventId` en la reserva
7. Notificar al salón por WhatsApp sobre nueva cita
8. Generar deep link de WhatsApp con mensaje prellenado "join <keyword>"
9. Retornar deep link al frontend

Entrada:

```
{
  "servicio": "unas-gel",
  "fecha": "2026-02-15",
  "horaInicio": "14:00"
}
```

Salida:

```
{
  "_id": "507f1f77bcf86cd799439011",
  "nombreCliente": "María González",
  "telefonoCliente": "3511270276",
  "servicio": "unas-gel",
}
```

```
"fecha": "2026-02-15",
"horaInicio": "14:00",
"horaFin": "15:00",
"duracion": 60,
"estado": "confirmada",
"estadoEncuesta": "pendiente_conexion",
"googleCalendarEventId": "abc123xyz",
"whatsappDeepLink":
"https://api.whatsapp.com/send?phone=14155238886&text=join%20movement-limited",
"createdAt": "2026-02-02T14:30:00.000Z"
}
```

RF-03.4 Apertura Automática de WhatsApp

Prioridad: Alta

Descripción: El frontend debe abrir el deep link de WhatsApp automáticamente tras crear la reserva.

Criterios de aceptación:

- Mostrar pantalla de éxito con detalles de la cita
 - Abrir `window.open(whatsappDeepLink, '_blank')` con delay de 1.5 segundos
 - Mostrar botón manual "Abrir WhatsApp ➡️" como fallback
 - Instrucción visible: "Se abrirá WhatsApp para enviar tu confirmación"
-

RF-04: Integración con Google Calendar

RF-04.1 Creación de Evento en Google Calendar

Prioridad: Alta

Descripción: Cada reserva confirmada debe crear un evento automáticamente en Google Calendar del salón.

Criterios de aceptación:

- Autenticación mediante Service Account (credentials JSON en variable de entorno)
- Scope requerido: <https://www.googleapis.com/auth/calendar>
- Calendar ID configurable (default: `primary` del Service Account)

Formato de evento:

```
{ summary: '💅 Uñas de Gel — María González', description: 'Cliente: María González\nTeléfono: 3511270276\nServicio: Uñas de Gel\nPrecio: $450 MXN\nID Reserva: 507f...', location: 'Soumaya Beauty Bar', start: { dateTime: '2026-02-15T14:00:00', timeZone: 'America/Mexico_City' }, end: { dateTime: '2026-02-15T15:00:00', timeZone: 'America/Mexico_City' }, colorId: '9' }
```

-
- Retornar `eventId` para almacenar en MongoDB
- Log de éxito/error en consola del servidor

RF-04.2 Eliminación de Evento al Cancelar

Prioridad: Alta

Descripción: Si la cita es cancelada, el evento debe eliminarse de Google Calendar.

Criterios de aceptación:

- Verificar que existe `googleCalendarEventId` en la reserva
 - Llamar a `calendar.events.delete({ calendarId, eventId })`
 - Manejar error si evento ya fue eliminado manualmente (no fallar la operación)
 - Log de éxito/error
-

RF-05: Integración con WhatsApp (Twilio)

RF-05.1 Configuración de Twilio Sandbox

Prioridad: Crítica

Descripción: Uso de Twilio Sandbox para WhatsApp en fase de desarrollo/testing.

Configuraciones requeridas:

- Account SID y Auth Token en variables de entorno
- Número de sandbox: `whatsapp:+14155238886` (compartido por todos los usuarios de Twilio)
- Keyword del sandbox: configurable (ej: `join movement-limited`)
- Webhook URL configurada en Twilio Console:
 - **Endpoint:** `POST`
`https://soumaya-beauty-salon.onrender.com/api/whatsapp/webhook`
 - **Evento:** "When a message comes in"

Limitaciones del Sandbox:

- Solo usuarios que envíen "join <keyword>" pueden recibir mensajes
- Sesión expira en 3 días (re-join requerido)
- No soporta plantillas personalizadas
- Máximo 50 mensajes/día en cuenta trial de Twilio

Migración a Producción (futuro):

- Registrar WhatsApp Business Account (WABA) vía Meta
- Usar número propio (+523511270276)
- Aprobar plantillas de mensajes

- Sin límite de sesión

RF-05.2 Generación de Deep Link de WhatsApp

Prioridad: Alta

Descripción: Crear deep link que abra WhatsApp con mensaje prellenado.

Formato del deep link:

`https://api.whatsapp.com/send?phone=<SANDBOX_NUMBER>&text=<URL_ENCODED_KEYWORD>`

Ejemplo:

`https://api.whatsapp.com/send?phone=14155238886&text=join%20movement-limited`

Criterios de aceptación:

- Número sin prefijo **whatsapp**: ni + en el deep link
- Keyword URL-encoded correctamente
- Compatible con Android e iOS (abre app nativa o WhatsApp Web)

RF-05.3 Detección de Conexión al Sandbox (Webhook)

Prioridad: Alta

Descripción: Cuando la clienta envía el mensaje "join <keyword>", el webhook debe detectar que se conectó y enviar la confirmación.

Flujo:

1. Twilio recibe "join <keyword>" de la clienta
2. Twilio confirma la conexión automáticamente (mensaje propio de Twilio)
3. Twilio llama al webhook con el mensaje "join <keyword>"
4. Webhook detecta que es un mensaje de tipo "join"
5. Busca reserva con **estadoEncuesta**: 'pendiente_conexion' y teléfono coincidente (últimos 10 dígitos)
6. Cambia **estadoEncuesta** a 'encuesta_cancelacion_pendiente'
7. Envía mensaje de confirmación con encuesta

Criterios de aceptación:

- Detección case-insensitive de "join " al inicio del mensaje
- Búsqueda de reserva por últimos 10 dígitos (maneja +52, +1, etc.)
- Si no hay reserva pendiente, no enviar nada extra (Twilio ya respondió)
- Response inmediato a Twilio: **<Response></Response>** (XML vacío para evitar timeout)

RF-05.4 Mensaje de Confirmación con Encuesta de Cancelación

Prioridad: Crítica

Descripción: Tras conectarse al sandbox, enviar confirmación de la cita con encuesta "¿Desea cancelar? Sí o No".

Formato del mensaje:

Hola María González! 🌸

✅ Tu cita está confirmada

📅 sábado, 15 de febrero de 2026

🕒 14:00 - 15:00

💅 Uñas de Gel

💰 \$450 MXN

📍 Soumaya Beauty Bar

¡Te esperamos! 💕

¿Desea cancelar su cita?

Responde *Sí* o *No*

Criterios de aceptación:

- Usar caracteres Unicode correctamente (✅, 📅, 💅, etc.)
- Formato de fecha en español (día de semana + número + mes completo)
- Precio con separador de miles si aplica
- Negrita simulada con asteriscos en "Sí" y "No" (si el cliente de WhatsApp lo soporta)
- Envío mediante `client.messages.create()` de Twilio SDK

RF-05.5 Encuesta de Cancelación Interactiva

Prioridad: Crítica

Descripción: Procesar respuestas de la clienta a la encuesta de cancelación.

Respuestas aceptadas (case-insensitive):

- **Afirmativas:** "sí", "si", "yes", "y", "ok", "sip"
- **Negativas:** "no", "nope", "non"

Flujo si responde "Sí":

1. Cambiar estado de reserva a `cancelada` en MongoDB
2. Cambiar `estadoEncuesta` a `'encuesta_reagendar_pendiente'`
3. Invalidar `cancelToken` (set a `null`)
4. Eliminar evento de Google Calendar usando `googleCalendarEventId`
5. Notificar al salón por WhatsApp sobre la cancelación

Enviar mensaje a la clienta:

✓ Tu cita de Uñas de Gel el sábado, 15 de febrero de 2026 a las 14:00 ha sido cancelada. El evento también fue eliminado de tu calendario. _____ ¿Desea reagendar una nueva cita? 🌸 Responde *Sí* o *No*

6.

Flujo si responde "No":

1. Cambiar **estadoEncuesta** a 'completada'

Enviar mensaje de confirmación:

¡Genial! 🌸 Tu cita sigue confirmada. ¡Te esperamos! 💖 Soumaya Beauty Bar

2.

Flujo si responde texto no reconocido:

1. No cambiar estado de la reserva

Enviar mensaje de aclaración:

No entendí tu respuesta 😊 ¿Desea cancelar su cita? Por favor responde *Sí* o *No*

2.

RF-05.6 Encuesta de Reagendamento

Prioridad: Alta

Descripción: Tras cancelar, preguntar si desea reagendar.

Flujo si responde "Sí":

1. Cambiar **estadoEncuesta** a 'completada'

Enviar enlace a página de reservaciones:

¡Genial! 🌸 Abre el siguiente enlace para agendar tu nueva cita: <https://soumaya-beauty-salon.vercel.app/reservaciones> Selecciona el horario que prefieras. ¡Te esperamos! 💖

2.

Flujo si responde "No":

1. Cambiar **estadoEncuesta** a 'completada'

Enviar mensaje de despedida:

De acuerdo 🌸 Si en algún momento deseas agendar una cita, no dudes en visitar: <https://soumaya-beauty-salon.vercel.app/reservaciones> ¡Que tengas un día genial! 💖 Soumaya Beauty Bar

2.

RF-05.7 Notificaciones al Salón


Prioridad: Alta


Descripción: Notificar al número del salón sobre nuevas citas y cancelaciones.


Mensaje de nueva cita:


 NUEVA CITA AGENDADA


 Cliente: María González



 Teléfono: 3511270276

 Fecha: sábado, 15 de febrero de 2026

 Hora: 14:00 - 15:00

 Servicio: Uñas de Gel


 Precio: \$450 MXN


 Evento agregado a Google Calendar 

Mensaje de cancelación:

 CITA CANCELADA

 Cliente: María González



 Teléfono: 3511270276

 Fecha: sábado, 15 de febrero de 2026

 Hora: 14:00

 Servicio: Uñas de Gel

El cliente canceló desde WhatsApp (encuesta).

 Evento eliminado de Google Calendar 

Criterios de aceptación:

- Enviado al número del salón: [whatsapp :+523511270276](https://wa.me/523511270276)
- Envío asíncrono (no debe bloquear respuesta a la cliente)
- Error en notificación no debe interrumpir flujo principal

RF-06: Sistema de Recordatorios

RF-06.1 Cron Job de Recordatorios Diarios







Prioridad: Media

Descripción: Enviar recordatorios automáticos 24 horas antes de cada cita.

Criterios de aceptación:

- Ejecución diaria a las 9:00 AM (zona horaria: America/Mexico_City)
- Query de reservas con:
 - `estado: 'confirmada'`
 - `fecha` igual a mañana (fecha actual + 1 día)

Por cada reserva encontrada, enviar WhatsApp:

 RECORDATORIO DE CITA Hola María González!  Mañana tienes tu cita:  17 sábado, 15 de febrero de 2026  14:00  Uñas de Gel ¡No olvides asistir!  Soumaya Beauty Bar

-
- Log de cada recordatorio enviado
- Si falla envío a una clienta, continuar con las demás

Librería: `node-cron`

Sintaxis: `cron.schedule('0 9 * * *', callback, { timezone: 'America/Mexico_City' })`

RF-07: Cancelación por Link (Fallback)

RF-07.1 Link de Cancelación Único

Prioridad: Baja (Fallback)

Descripción: Mantener compatibilidad con links de cancelación antiguos.

Formato del link:

<https://soumaya-beauty-salon.onrender.com/api/cancel/507f1f77bcf86cd799439011?token=abc123...>

Criterios de aceptación:

- Verificar que `cancelToken` coincida con el almacenado en DB
 - Verificar que reserva esté en estado `confirmada`
 - Si válido:
 - Cancelar reserva (estado → `cancelada`)
 - Eliminar evento de Google Calendar
 - Notificar al salón
 - Enviar WhatsApp con pregunta de reagendar
 - Redirigir a página HTML de confirmación
 - Si inválido:
 - Redirigir a página HTML de error
-

RF-08: Visualización de Reservas

RF-08.1 Listado de Reservas del Usuario

Prioridad: Media

Descripción: La usuaria debe poder ver su historial de citas.

Endpoint: GET /api/reservations

Autenticación: Requerida (JWT)

Criterios de aceptación:

- Retornar solo reservas del usuario autenticado
- Ordenadas por fecha descendente (más recientes primero)
- Incluir campos:
 - ID de reserva
 - Servicio (ID y nombre legible)
 - Fecha en formato legible (ej: "sábado, 15 de febrero de 2026")
 - Hora de inicio y fin
 - Duración en minutos
 - Estado (confirmada / cancelada / completada)
 - Nombre del cliente
- Filtros opcionales: por estado, por rango de fechas

Salida:

```
[
  {
    "_id": "507f...",
    "servicio": "unas-gel",
    "servicioNombre": "Uñas de Gel",
    "fecha": "2026-02-15",
    "fechaLegible": "sábado, 15 de febrero de 2026",
    "horaInicio": "14:00",
    "horaFin": "15:00",
    "duracion": 60,
    "estado": "confirmada",
    "nombreCliente": "María González"
  }
]
```

RF-08.2 Cancelación Manual desde Panel

Prioridad: Media

Descripción: La usuaria puede cancelar su propia cita desde el panel web.

Endpoint: PUT /api/reservations/:id/cancel

Autenticación: Requerida (JWT)

Criterios de aceptación:

- Verificar que la reserva pertenece al usuario autenticado
 - Verificar que esté en estado **confirmada**
 - Cancelar reserva (estado → **cancelada**)
 - Eliminar evento de Google Calendar
 - NO enviar WhatsApp (cancelación manual, no encuesta)
 - Retornar reserva actualizada
-

4. REQUERIMIENTOS NO FUNCIONALES

RNF-01: Rendimiento

RNF-01.1 Tiempo de Respuesta

- **Carga de página de reservaciones:** < 2 segundos
- **Creación de reserva (backend):** < 3 segundos
- **Query de disponibilidad semanal:** < 1 segundo
- **Respuesta de webhook de WhatsApp:** < 200ms (respuesta a Twilio), procesamiento asíncrono después

RNF-01.2 Capacidad

- Soportar mínimo 100 reservas activas simultáneas
- Procesar hasta 50 webhooks de WhatsApp por minuto
- Soportar hasta 20 usuarias navegando simultáneamente

RNF-01.3 Escalabilidad

- Arquitectura preparada para escalar horizontalmente (múltiples instancias)
 - Base de datos con índices en campos críticos:
 - **telefonoCliente + estadoEncuesta + estado** (compuesto)
 - **fecha + estado**
 - **usuario**
 - Cache de disponibilidad semanal (futuro)
-

RNF-02: Seguridad

RNF-02.1 Autenticación y Autorización

- Contraseñas hasheadas con bcrypt (salt rounds: 10)
- Tokens JWT con expiración de 30 días
- Secret key de JWT almacenada en variable de entorno (no en código)
- Middleware de autenticación en todas las rutas protegidas

- Validación de propiedad de recursos (un usuario solo ve/modifica sus propias reservas)

RNF-02.2 Protección de Datos Sensibles

- Variables de entorno para:
 - `TWILIO_ACCOUNT_SID`
 - `TWILIO_AUTH_TOKEN`
 - `GOOGLE_SERVICE_ACCOUNT_JSON`
 - `MONGODB_URI`
 - `JWT_SECRET`
- Archivo `.env` excluido de control de versiones (`.gitignore`)
- No exponer stack traces en producción

RNF-02.3 Validación de Entrada

- Validar formato de teléfono (10 dígitos después de +52)
- Validar formato de fecha (YYYY-MM-DD)
- Validar formato de hora (HH:MM)
- Validar enum de servicios (solo IDs válidos)
- Sanitización de entrada para prevenir inyección NoSQL

RNF-02.4 Rate Limiting

- Login: máximo 5 intentos fallidos por IP en 15 minutos
- Creación de reservas: máximo 10 por usuario por día
- Webhooks: sin rate limiting (controlado por Twilio)

RNF-02.5 HTTPS

- Todas las comunicaciones sobre HTTPS en producción
 - Certificado SSL válido en dominio de producción
-

RNF-03: Disponibilidad

RNF-03.1 Uptime

- Objetivo: 99% uptime mensual (permite ~7 horas de downtime por mes)
- Servidor backend en plataforma cloud con auto-restart (Render / Railway / Heroku)
- MongoDB con replicación (MongoDB Atlas con cluster M0 o superior)

RNF-03.2 Manejo de Errores

- Try-catch en todos los endpoints críticos
- Logs estructurados de errores con timestamp
- Mensajes de error amigables al usuario (no técnicos)
- Rollback de transacciones parciales:

- Si falla creación de Google Calendar, no crear reserva
- Si falla envío de WhatsApp, crear reserva igual pero loggear error

RNF-03.3 Recuperación ante Fallos

- Si Twilio está caído, permitir crear reserva pero marcar como "sin notificación"
 - Si Google Calendar está caído, crear reserva sin `googleCalendarEventId`
 - Si MongoDB está caído, retornar HTTP 503 Service Unavailable
-

RNF-04: Usabilidad

RNF-04.1 Interfaz de Usuario

- Diseño responsive (móvil, tablet, desktop)
- Breakpoints: 320px (móvil), 768px (tablet), 1024px (desktop)
- Colores del salón:
 - Primario: `#d53f8c` (rosa)
 - Fondo: `#fdf2f8` (rosa claro)
 - Texto: `#1a202c` (gris oscuro)
- Emojis para identificación visual rápida de servicios
- Indicador de progreso visible en flujo de 4 pasos

RNF-04.2 Accesibilidad

- Contraste suficiente para cumplir WCAG AA (4.5:1 para texto normal)
- Labels en formularios asociados correctamente
- Botones con estados hover y active visibles
- Mensajes de error claros y descriptivos

RNF-04.3 Idioma

- Todo el sistema en español (México)
 - Formato de fecha: "día de semana, D de mes de YYYY" (ej: "sábado, 15 de febrero de 2026")
 - Formato de hora: 24 horas (HH:MM)
 - Zona horaria: America/Mexico_City
-







RNF-05: Mantenibilidad

RNF-05.1 Código

- Estructura modular (MVC adaptado):
 - `models/` - Modelos de Mongoose
 - `controllers/` - Lógica de negocio

- `routes/` - Definición de endpoints
- `utils/` - Servicios auxiliares (WhatsApp, Google Calendar, cron)
- `config/` - Configuración de BD
- Comentarios en funciones complejas
- Nombres de variables descriptivos en español/inglés mixto
- Separación de lógica de negocio y lógica de presentación

RNF-05.2 Logs

- Logs estructurados con prefijos emoji:
 -  Operaciones exitosas
 -  Errores críticos
 -  Advertencias
 -  Mensajes enviados
 -  Búsquedas en BD
 -  Eventos de calendario
- Timestamp en todos los logs
- Logs de inicio de servidor con configuración actual

RNF-05.3 Documentación

- README con instrucciones de instalación
 - `.env.example` con todas las variables requeridas
 - Comentarios en código explicando flujos complejos
 - Este documento de requerimientos actualizado
-

RNF-06: Portabilidad

RNF-06.1 Despliegue

- Backend: Node.js v18+ compatible con Render, Railway, Heroku
- Frontend: React/Vite compatible con Vercel, Netlify
- Base de datos: MongoDB Atlas (cloud) o cualquier servidor MongoDB

RNF-06.2 Dependencias

Backend:

```
{
  "express": "^4.18.2",
  "mongoose": "^7.0.0",
  "bcryptjs": "^2.4.3",
  "jsonwebtoken": "^9.0.0",
  "twilio": "^4.20.0",
  "googleapis": "^128.0.0",
  "node-cron": "^3.0.2",
  "dotenv": "^16.0.3",
```

```
"cors": "^2.8.5"
}
```

Frontend:

```
{
  "react": "^18.2.0",
  "axios": "^1.4.0"
}
```

RNF-07: Integraciones

RNF-07.1 Twilio WhatsApp API

- **Versión:** Twilio API v1
- **SDK:** [twilio](#) npm package v4.20.0+
- **Rate limits:** Respetados por Twilio (50 msg/día en trial, ilimitado en producción)
- **Timeout webhook:** Respuesta en < 15 segundos
- **Formato de respuesta webhook:** TwiML XML vacío `<Response></Response>`

RNF-07.2 Google Calendar API

- **Versión:** v3
- **SDK:** [googleapis](#) npm package v128.0.0+
- **Autenticación:** Service Account con JSON credentials
- **Permisos:** Scope <https://www.googleapis.com/auth/calendar>
- **Quota:** 1,000,000 queries/día (más que suficiente)
- **Formato de fecha:** ISO 8601 con zona horaria

RNF-07.3 MongoDB

- **Versión:** MongoDB 5.0+
 - **Driver:** Mongoose v7.0.0+
 - **Conexión:** URI de MongoDB Atlas o servidor local
 - **Timeout:** 10 segundos de conexión inicial
-

RNF-08: Cumplimiento Legal

RNF-08.1 Protección de Datos Personales (INAI - México)

- Recolección de datos mínimos necesarios
- Consentimiento implícito al registrarse
- Uso de datos solo para operación del sistema
- No compartir datos con terceros sin consentimiento

- Derecho de acceso, rectificación y cancelación (ARCO) disponible bajo solicitud

RNF-08.2 WhatsApp Business Policy

- Mensajes transaccionales permitidos (confirmaciones, notificaciones)
 - No spam ni mensajes promocionales sin opt-in explícito
 - Opt-in requerido antes de enviar primer mensaje (join keyword en sandbox)
 - Respetar opt-out si la clienta responde "stop"
-

5. CASOS DE USO PRINCIPALES

CU-01: Agendar Cita Exitosamente

Actor: Clienta registrada

Precondición: Usuario autenticado

Flujo Principal:

1. Usuaría navega a [/reservaciones](#)
2. Selecciona servicio "Uñas de Gel"
3. Selecciona fecha "15 de febrero de 2026"
4. Selecciona hora "14:00"
5. Confirma la reserva
6. Sistema crea reserva en MongoDB
7. Sistema crea evento en Google Calendar
8. Sistema retorna deep link de WhatsApp
9. Frontend abre WhatsApp automáticamente
10. Usuaría envía "join movement-limited"
11. Twilio conecta a la usuaria al sandbox
12. Webhook detecta conexión y envía confirmación
13. Usuaría recibe mensaje de confirmación con encuesta

Postcondición: Cita confirmada, evento en calendario, encuesta enviada

CU-02: Cancelar Cita por WhatsApp

Actor: Clienta con cita confirmada

Precondición: Ya recibió mensaje de confirmación

Flujo Principal:

1. Usuaría responde "Sí" a la encuesta de cancelación
2. Webhook procesa respuesta
3. Sistema cancela reserva en MongoDB
4. Sistema elimina evento de Google Calendar
5. Sistema notifica al salón

6. Sistema envía mensaje de cancelación a la clienta
7. Sistema pregunta si desea reagendar
8. Usuaría responde "Sí"
9. Sistema envía link de reagendamiento
10. Usuaría abre link y agenda nueva cita

Postcondición: Cita cancelada, evento eliminado, nueva cita puede ser agendada

CU-03: Recibir Recordatorio

Actor: Clienta con cita mañana

Precondición: Cita confirmada para el día siguiente

Flujo Principal:

1. Cron job se ejecuta a las 9:00 AM
2. Sistema consulta citas para mañana
3. Sistema envía WhatsApp de recordatorio
4. Usuaría recibe recordatorio

Postcondición: Clienta recordada de su cita

6. MATRIZ DE TRAZABILIDAD

Requerimiento	Prioridad	Componente	Dependencias	Estado
RF-01.1	Alta	Auth	MongoDB, bcrypt, JWT	 Implementado
RF-01.2	Alta	Auth	MongoDB, JWT	 Implementado
RF-03.3	Crítica	Reservas	MongoDB, Google Calendar, WhatsApp	 Implementado
RF-04.1	Alta	Google Calendar	googleapis, Service Account	 Implementado
RF-04.2	Alta	Google Calendar	googleapis	 Implementado
RF-05.4	Crítica	WhatsApp	Twilio SDK, Webhook	 Implementado

RF-05.5	Crítica	WhatsApp	Twilio SDK, MongoDB	✓ Implementado
RF-05.6	Alta	WhatsApp	Twilio SDK	✓ Implementado
RF-06.1	Media	Cron	node-cron, WhatsApp	✓ Implementado
RNF-01.1	Alta	Performance	Backend optimization	↻ En evaluación
RNF-02.1	Alta	Seguridad	bcrypt, JWT, env vars	✓ Implementado
RNF-03.1	Media	Disponibilidad	Cloud hosting	↻ En deployment

7. RIESGOS Y MITIGACIONES

Riesgo	Probabilidad	Impacto	Mitigación
Twilio Sandbox expira después de 3 días	Alta	Medio	Documentar proceso de re-join para clientas. Migrar a WABA en producción.
Google Calendar API quota excedido	Baja	Alto	Monitorear uso diario. Implementar retry con exponential backoff.
WhatsApp caído	Baja	Medio	Crear reserva igual, marcar como "sin notificación". Reintentar envío después.
Número de teléfono duplicado en registro	Media	Bajo	Validar unicidad de email (no teléfono), permitir que múltiples usuarios compartan teléfono.
Reservas solapadas por race condition	Baja	Alto	Implementar lock en MongoDB (uso de transacciones o índice único compuesto en fecha+hora+servicio).

8. PLAN DE TESTING

8.1 Testing Funcional

- [] Registro de usuario con datos válidos
- [] Login con credenciales correctas/incorrectas
- [] Creación de reserva con horario disponible
- [] Intento de reserva con horario ocupado (debe fallar)
- [] Envío de WhatsApp de confirmación
- [] Respuesta "Sí" a encuesta de cancelación
- [] Eliminación de evento de Google Calendar
- [] Respuesta "No" a encuesta (cita permanece confirmada)
- [] Deep link de WhatsApp abre correctamente en móvil
- [] Cron job ejecuta recordatorios a las 9 AM

8.2 Testing de Integración

- [] MongoDB → Backend → Frontend (flujo completo de reserva)
- [] Backend → Twilio → Webhook → Backend (flujo de WhatsApp)
- [] Backend → Google Calendar → Backend (creación y eliminación)

8.3 Testing de Seguridad

- [] Intento de acceso sin token JWT (debe retornar 401)
- [] Intento de cancelar reserva de otro usuario (debe retornar 403)
- [] Inyección NoSQL en campos de entrada (debe ser prevenido)

8.4 Testing de Performance

- [] Tiempo de carga de página de reservaciones < 2s
- [] Creación de reserva completa < 3s
- [] Response de webhook < 200ms

9. GLOSARIO

Término	Definición
Deep link	URL especial que abre una aplicación nativa (WhatsApp) directamente con datos prellenados.
Webhook	Endpoint HTTP que recibe notificaciones automáticas de servicios externos (Twilio, Google).
Service Account	Cuenta de Google no asociada a un usuario humano, usada para autenticación de servidor a servidor.
JWT (JSON Web Token)	Estándar abierto para transmitir información de autenticación de forma segura como objeto JSON.

Twilio Sandbox	Entorno de pruebas de Twilio WhatsApp con número compartido, sin costo adicional.
WABA (WhatsApp Business Account)	Cuenta oficial de WhatsApp Business requerida para enviar mensajes desde número propio.
Cron job	Tarea programada que se ejecuta automáticamente en intervalos específicos.
Race condition	Situación donde múltiples procesos acceden simultáneamente al mismo recurso causando inconsistencia.

Fin del Documento