

# **Documento de Especificación de Requisitos de Software (SRS)**

**Proyecto:** A point date

**Versión:** 1.0

**Fecha:** 01/octubre/2025

**Autores:** Calleja Jiménez Fernando Yahir, De la rosa Valdés Itzli Fernanda

---

## **1. Introducción**

### **1.1 Propósito**

Para todo aquel que trabaje en un negocio independiente el manejo de las citas y la administración de las mismas resulta una parte importante para llevar a cabo un registro de ingresos, gastos para los materiales que se usa para cada servicio y el no llevar a cabo una organización de tu negocio puede llevar a consecuencias graves, tales como la perdida de clientes, no saber cuales horarios tienes ocupados y todo eso lleva a que estos estresado y no aproveches tu día a día de una buena manera.

Para los que buscamos un servicio que requiera de agendar una cita es problemático cuando a uno se le llega a olvidar sus citas pendientes.

Es por eso que este proyecto va dirigido a todo aquel que quiera mejorar su organización y llevar un mejor manejo de todos sus clientes y para los que prefieren tener una aplicación en la que te ayuda a contactar de una manera fácil a todos estos proveedores que están esperando a estos clientes.

### **1.2 Alcance**

El proyecto esta enfocado en manjera una agenda para los proveedores de servicios, los cuales podrán ver no solo sus citas pendientes, podrás ver la peticiones que le han estado haciendo, tendrán un apartado de analytics en el cual podrán ver el crecimiento de su negocio, clientes por mes, ingresos, clientes frecuentes, clientes nuevos, el nuevo alcance que han logrado.

El proyecto no planea reemplazar por completo a formas de administración, ya que esto no será una total base de datos para manejar y monitorear clientes.

---

## **2. Descripción general**

### **2.1 Perspectiva del producto**

El sistema es una plataforma web que conecta a clientes con proveedores de servicios locales. Funciona como un marketplace donde los usuarios pueden buscar, reservar y gestionar citas con distintos prestadores (ejemplo: quiroprácticos, estilistas, doctores, servicios de belleza).

El sistema ofrecerá dos vistas principales:

- **Vista cliente:** búsqueda de servicios, reserva de citas y gestión de historial.
- **Vista proveedor:** administración de agenda, control de citas, estadísticas de negocio.
- **Vista administrador:** monitoreo de usuarios, control de calidad del contenido y métricas globales.

### **2.2 Funciones del producto**

Las funciones generales del sistema incluyen:

- Registro y autenticación de usuarios (clientes y proveedores).
- Búsqueda y filtrado de servicios cercanos.
- Reservación y cancelación de citas en línea.
- Gestión de agenda por parte del proveedor.
- Generación de métricas y reportes (número de citas, ingresos, clientes recurrentes).
- Panel de administración para control del sistema.

### **2.3 Características de los usuarios**

**Cliente:** persona que busca un servicio y agenda citas. Generalmente requiere una interfaz sencilla, rápida y con acceso desde dispositivos móviles.

**Proveedor de servicios:** profesional o negocio que ofrece servicios y requiere gestionar su agenda, mostrar disponibilidad y acceder a estadísticas de su negocio.

**Administrador del sistema:** usuario con permisos de control total sobre la plataforma, encargado de la seguridad, soporte, gestión de usuarios y supervisión general.

### **2.4 Restricciones**

El sistema debe ser accesible desde navegadores web modernos.

Se requiere conexión a internet para todas las funcionalidades principales.

Los pagos en línea estarán sujetos a la integración con pasarelas externas (ej. Stripe, PayPal).

La interfaz deberá estar inicialmente en español, con posibilidad de multilenguaje a futuro.

## **2.5 Suposiciones y dependencias**

Los usuarios cuentan con dispositivos con acceso a internet (PC o smartphone).

El proveedor es responsable de mantener actualizada su disponibilidad en la agenda.

El sistema depende de servicios externos para la geolocalización y pasarela de pagos.

Se asume que los usuarios aceptarán términos y condiciones de uso para registrarse.

---

## **3. Requisitos específicos**

### **3.1 Requisitos funcionales**

#### **Gestión de usuarios**

- RF-01: El sistema permitirá el registro de clientes mediante correo, teléfono y contraseña.
- RF-02: El sistema permitirá el registro de proveedores con información de negocio (nombre, dirección, categoría, horarios).
- RF-03: El sistema permitirá a los administradores gestionar (activar, suspender, eliminar) cuentas de usuarios.
- RF-04: El sistema permitirá la autenticación de usuarios con recuperación de contraseña por correo.

#### **Gestión de servicios**

- RF-05: El proveedor podrá registrar los servicios que ofrece, incluyendo nombre, descripción, precio, lugar y método de pago
- RF-06: El cliente podrá buscar servicios cercanos mediante filtros (categoría, ubicación, precio).
- RF-07: El sistema mostrará servicios disponibles en formato de tarjetas con información básica.

#### **Agenda de citas**

- RF-08: El cliente podrá reservar citas seleccionando fecha, hora y servicio.
- RF-09: El proveedor podrá aceptar, modificar o cancelar citas desde su panel.
- RF-10: El sistema enviará notificaciones por correo al cliente y proveedor al crear, modificar o cancelar una cita.
- RF-11: El cliente podrá consultar el historial de citas pasadas y próximas.
- RF-12: El proveedor podrá bloquear horarios en su agenda para evitar reservas.

#### **Analytics y reportes**

- RF-13: El sistema generará un reporte mensual de número de citas atendidas.

- RF-14: El sistema calculará los ingresos mensuales por proveedor.
- RF-15: El sistema mostrará métricas gráficas (ej. crecimiento de citas, ingresos por mes, clientes nuevos vs recurrentes).

## Administración del sistema

- RF-16: El administrador podrá visualizar estadísticas globales (número de usuarios, número de citas activas, categorías más usadas).
- RF-17: El administrador podrá gestionar reportes de usuarios (comentarios ofensivos, fraudes, etc.).

## 3.2 Requisitos de interfaces externas

### Interfaz de usuario

- IU-01: La aplicación deberá ser accesible desde navegadores web en dispositivos móviles y de escritorio.
- IU-02: La interfaz será responsive y deberá adaptarse automáticamente al tamaño de pantalla.
- IU-03: Los formularios de registro, login y reserva deberán validar datos antes de enviarse.

### Interfaces de software

- IS-01: El sistema deberá conectarse a una pasarela de pagos (Stripe o PayPal).
- IS-02: El sistema deberá conectarse a un servicio de mapas/geolocalización (ej. Google Maps API).

### Interfaces de comunicación

- IC-01: El sistema usará HTTPS para todas las transacciones.
- IC-02: El sistema enviará notificaciones por correo electrónico mediante un servicio externo (ej. SendGrid, AWS SES).

## 3.3 Requisitos de rendimiento

- RP-01: El sistema deberá permitir al menos 500 usuarios concurrentes en su primera versión.
- RP-02: El tiempo de respuesta de cualquier solicitud no deberá exceder los 3 segundos bajo carga normal.
- RP-03: El sistema deberá procesar al menos 50 reservas por minuto sin fallos.

## 3.4 Requisitos de diseño

- RD-01: El sistema deberá estar desarrollado bajo una arquitectura cliente-servidor.
- RD-02: El backend deberá implementar una API RESTful para comunicación con el frontend.
- RD-03: La base de datos deberá soportar integridad referencial y escalabilidad horizontal.

- RD-04: El sistema deberá seguir principios de diseño modular para facilitar mantenibilidad.

### **3.5 Requisitos de calidad**

#### **Seguridad**

- RQ-01: Todas las contraseñas deberán almacenarse cifradas (bcrypt o equivalente).
- RQ-02: El sistema deberá implementar autenticación basada en tokens (JWT).

#### **Usabilidad**

- RQ-03: La interfaz deberá ser intuitiva, con navegación clara y mínima curva de aprendizaje.
- RQ-04: El sistema deberá ofrecer accesibilidad básica (contraste de colores, etiquetas para lectores de pantalla).

#### **Disponibilidad**

- RQ-05: El sistema deberá estar disponible al menos el 99% del tiempo mensual.

#### **Mantenibilidad**

- RQ-06: El código deberá estar documentado y versionado en un sistema de control (ej. Git).

#### **Portabilidad**

- RQ-07: El sistema deberá poder desplegarse en servidores Linux y Windows.

### **3.6 Requerimientos No Funcionales (RNF)**

1. **RNF-01 (Seguridad):** Todas las contraseñas deberán almacenarse con un algoritmo de hashing seguro (bcrypt o Argon2).
2. **RNF-02 (Seguridad):** Toda comunicación entre cliente y servidor deberá realizarse bajo el protocolo HTTPS.
3. **RNF-03 (Rendimiento):** El tiempo de respuesta promedio del sistema no deberá exceder los 3 segundos bajo carga normal.
4. **RNF-04 (Disponibilidad):** El sistema deberá estar disponible al menos el 99% del tiempo en un mes calendario.
5. **RNF-05 (Usabilidad):** La interfaz gráfica deberá ser responsive y adaptarse automáticamente a dispositivos móviles, tablets y escritorio.
6. **RNF-06 (Accesibilidad):** El sistema deberá cumplir con estándares básicos de accesibilidad (etiquetas en formularios, contraste adecuado).
7. **RNF-07 (Mantenibilidad):** El código fuente deberá estar versionado en un sistema de control de versiones (ej. Git).
8. **RNF-08 (Escalabilidad):** El sistema deberá poder aumentar su capacidad para manejar hasta 1,000 usuarios concurrentes sin cambios en la arquitectura base.
9. **RNF-09 (Confiabilidad):** El sistema deberá realizar respaldos automáticos diarios de la base de datos.
10. **RNF-10 (Portabilidad):** El sistema deberá ser compatible con los navegadores más usados: Chrome, Edge, Firefox y Safari.