

# Especificación de Requisitos de Software (SRS)

---

## Sistema de Gestión de Servicios

---

### Tabla de Contenidos

- 1. [Introducción](#)
  - [1.1 Propósito](#)
  - [1.2 Alcance](#)
  - [1.3 Definiciones, Acrónimos y Abreviaturas](#)
  - [1.4 Referencias](#)
- 2. [Descripción General](#)
  - [2.1 Perspectiva del Producto](#)
  - [2.2 Funcionalidades del Producto](#)
  - [2.3 Características de los Usuarios](#)
  - [2.4 Restricciones](#)
- 3. [Requisitos Específicos](#)
  - [3.1 Requisitos Funcionales](#)
  - [3.2 Requisitos No Funcionales](#)
- 4. [Interfaces del Sistema](#)
  - [4.1 Interfaz de Usuario](#)
  - [4.2 Interfaces de Hardware](#)
  - [4.3 Interfaces de Software](#)
- 5. [Casos de Uso](#)
- 6. [Requisitos de Rendimiento](#)
- 7. [Requisitos de Seguridad](#)
- 8. [Otras Restricciones](#)
- 9. [Análisis de Riesgos](#)
- 10. [Aprobación y Seguimiento de Cambios](#)
- 11. [Anexos](#)
- 12. [Autor](#)

---

## 1. Introducción

### 1.1 Propósito

El propósito de este documento es describir los requisitos del **Sistema de Gestión de Servicios**. Este sistema permite registrar y gestionar información sobre clientes y servicios que la empresa ofrece, incluyendo la asignación de servicios específicos a cada cliente.

### 1.2 Alcance

Este proyecto consiste en un sistema de información que facilita la administración de clientes y servicios. El sistema cuenta con un backend para la gestión de datos y un frontend para la interacción con el usuario. Los

principales actores del sistema son los empleados o administradores que registran, editan y consultan información de clientes y servicios.

### 1.3 Definiciones, Acrónimos y Abreviaturas

- **API:** Interfaz de programación de aplicaciones (Application Programming Interface).
- **CRUD:** Crear, Leer, Actualizar, Eliminar.
- **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers).

### 1.4 Referencias

- Documentación oficial de [Node.js](#)
  - Documentación oficial de [MariaDB](#)
  - Documentación de IEEE 830
- 

## 2. Descripción General

### 2.1 Perspectiva del Producto

El sistema se divide en dos módulos principales:

- **Backend:** Gestiona la lógica del negocio y proporciona una API para interactuar con la base de datos.
- **Frontend:** Proporciona una interfaz gráfica para que los usuarios gestionen los datos de clientes y servicios.

### 2.2 Funcionalidades del Producto

El sistema permite:

- Registrar clientes con sus datos personales.
- Definir servicios disponibles, incluyendo nombre, descripción y costo.
- Asignar servicios a los clientes, registrando detalles como fecha y monto total.

### 2.3 Características de los Usuarios

El sistema está diseñado para ser utilizado por empleados administrativos o de soporte que necesiten gestionar la información de clientes y servicios.

### 2.4 Restricciones

- El sistema debe funcionar en un entorno local o en un servidor configurado adecuadamente.
  - Requiere una conexión a una base de datos MariaDB.
- 

## 3. Requisitos Específicos

### 3.1 Requisitos Funcionales

#### 1. Gestión de Clientes:

- El sistema debe permitir registrar nuevos clientes con su nombre, apellido, documento, email y teléfono.
- El sistema debe permitir editar y eliminar información de clientes.

## 2. Gestión de Servicios:

- El sistema debe permitir registrar nuevos servicios con nombre, descripción y costo.
- El sistema debe permitir editar y eliminar información de servicios.

## 3. Asignación de Servicios a Clientes:

- El sistema debe permitir asignar servicios específicos a un cliente.
- El sistema debe registrar la fecha, el estado y el monto total de cada servicio asignado a un cliente.

## 3.2 Requisitos No Funcionales

1. **Usabilidad:** La interfaz del usuario debe ser intuitiva y fácil de usar.
  2. **Escalabilidad:** El sistema debe poder ampliarse para soportar un número mayor de registros de clientes y servicios.
  3. **Rendimiento:** Las operaciones de consulta y actualización deben ejecutarse en un tiempo razonable.
  4. **Compatibilidad:** El frontend debe ser accesible desde navegadores modernos.
  5. **Seguridad:** Debe haber validación en la API para prevenir la entrada de datos inválidos.
- 

# 4. Interfaces del Sistema

## 4.1 Interfaz de Usuario

- **Vista de Clientes:** Permite al usuario ver, crear, editar y eliminar registros de clientes.
- **Vista de Servicios:** Permite al usuario ver, crear, editar y eliminar registros de servicios.
- **Vista de Asignación de Servicios:** Permite asignar servicios a un cliente y ver el historial de servicios asignados.

## 4.2 Interfaces de Hardware

No se requiere hardware especializado, únicamente un equipo de cómputo estándar con acceso a internet.

## 4.3 Interfaces de Software

- **Base de Datos:** MariaDB, utilizada para almacenar los datos de clientes, servicios y asignaciones.
  - **Backend:** Node.js y Express para la API y lógica del servidor.
  - **Frontend:** HTML, CSS y JavaScript.
- 

# 5. Casos de Uso

## 1. Registrar Cliente:

- **Actor:** Usuario administrativo
- **Descripción:** El usuario ingresa datos de un cliente para registrarlo en el sistema.

- **Precondiciones:** El usuario debe estar autenticado.
- **Postcondiciones:** El cliente es registrado en la base de datos.

## 2. Registrar Servicio:

- **Actor:** Usuario administrativo
- **Descripción:** El usuario ingresa datos de un servicio para registrarlo en el sistema.
- **Precondiciones:** El usuario debe estar autenticado.
- **Postcondiciones:** El servicio es registrado en la base de datos.

## 3. Asignar Servicio a Cliente:

- **Actor:** Usuario administrativo
- **Descripción:** El usuario selecciona un cliente y le asigna un servicio específico.
- **Precondiciones:** El cliente y el servicio deben estar registrados.
- **Postcondiciones:** El servicio es asignado al cliente y se registra en la base de datos.

---

## 6. Requisitos de Rendimiento

- Las operaciones de consulta de clientes y servicios deben ejecutarse en menos de 2 segundos.
- El sistema debe soportar al menos 50 usuarios simultáneos en una red local sin pérdida significativa de rendimiento.

---

## 7. Requisitos de Seguridad

- El sistema debe validar y sanitizar todas las entradas de datos para prevenir inyecciones SQL.
- Los datos sensibles deben estar protegidos y solo accesibles a usuarios autorizados.

---

## 8. Otras Restricciones

- El sistema debe ser compatible con navegadores como Google Chrome, Mozilla Firefox y Microsoft Edge.
- El backend debe ejecutarse en un servidor Node.js con acceso a una base de datos MariaDB configurada.

---

## 9. Análisis de Riesgos

- **Pérdida de Datos:** Riesgo de pérdida de datos si no se configuran correctamente los backups de la base de datos.
- **Problemas de Escalabilidad:** El sistema puede experimentar problemas si el número de registros crece significativamente sin mejoras en el hardware o ajustes de configuración.

---

## 10. Aprobación y Seguimiento de Cambios

- **Aprobación:** Este documento debe ser revisado y aprobado por el equipo de desarrollo y los interesados antes de proceder con la implementación.

- **Seguimiento de Cambios:** Cualquier cambio en los requisitos debe ser documentado y aprobado antes de ser implementado en el sistema.
- 

## 11. Anexos

- **Diagramas:** Diagrama de clases de la base de datos y diagramas de flujo de datos.
  - **Glosario:** Contiene términos específicos utilizados en el sistema y su definición.
- 

