

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

## 1. INTRODUCCIÓN:

El presente documento de especificación de requerimientos de software para el sistema de citas técnicas tiene como objetivo y/o propósito detallar con diagramas y casos de uso el desarrollo del producto de software. Este sistema está diseñado para gestionar eficientemente las citas para servicios de mantenimiento de aires acondicionados, incluyendo mantenimiento preventivo, correctivo e instalaciones. El sistema permitirá a los usuarios, tanto clientes como técnicos y administradores, interactuar de manera clara y efectiva, optimizando los procesos y mejorando la comunicación entre todas las partes involucradas.

Este documento servirá como guía fundamental para los desarrolladores, proporcionando una descripción clara de las funcionalidades requeridas, los procesos y la interacción entre los diferentes componentes del sistema y los usuarios. Además, se incluirán especificaciones técnicas detalladas para facilitar la comprensión y la implementación del sistema propuesto.

### 1.1. PROPÓSITO:

El propósito de este documento es describir de forma clara, concisa, específica, detallada e inequívoca los requerimientos del software de gestión de citas técnicas para mantenimiento de aires acondicionados. Este documento se destina a asegurar que todos los desarrolladores, partes interesadas y usuarios finales comprendan completamente las funcionalidades, interacciones y operaciones que el software debe soportar. Además, servirá como un contrato formal que define lo que el equipo de desarrollo debe entregar y lo que el cliente espera del producto final. Este documento facilitará la comunicación efectiva entre todos los involucrados en el proyecto, minimizando malentendidos y estableciendo expectativas claras para todos los aspectos del sistema.

### 1.2. ÁMBITO DEL SISTEMA:

El sistema de citas técnicas para mantenimiento de aires acondicionados se implementará como una aplicación web, accesible tanto desde dispositivos móviles como de escritorio a través de navegadores web modernos. Esto permite a los usuarios acceder al sistema en cualquier momento y desde cualquier lugar, siempre que dispongan de una conexión a internet.

La aplicación ofrecerá una interfaz intuitiva y fácil de usar tanto para los clientes como para los técnicos y administradores del sistema. Además, se contempla la posibilidad de desarrollar una aplicación móvil dedicada en fases futuras del proyecto para proporcionar una experiencia más optimizada y funciones específicas para dispositivos móviles.

La infraestructura del sistema se alojará en servidores en la nube, garantizando alta disponibilidad, escalabilidad y seguridad. Esto asegura que el sistema puede manejar un alto volumen de usuarios y citas sin degradar el rendimiento ni la experiencia del usuario.

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

**1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS:** Para asegurar una comprensión clara y uniforme de los términos técnicos y de industria utilizados en este documento, a continuación se proporciona un glosario de las definiciones, acrónimos y abreviaturas más relevantes:

SGBD	Sistema Gestor de Base de Datos. Software que permite crear, modificar y administrar una base de datos, garantizando la seguridad, la integridad, la persistencia y el mantenimiento de los datos.
IEEE	Institute of Electrical and Electronics Engineers. Organización profesional técnica dedicada al avance de la tecnología relacionada con la electricidad.
UML	Lenguaje Unificado de Modelado. Es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para modelar negocios y otros sistemas no software.
JAVASCRIPT	Lenguaje de programación interpretado, dialecto del estándar ECMAScript, utilizado principalmente en el lado del cliente para scripts de páginas web pero también en muchos entornos fuera del navegador web.
API	Interfaz de Programación de Aplicaciones. Conjunto de rutinas, protocolos y herramientas para construir software y aplicaciones, permitiendo la comunicación entre productos o servicios de software.
HTML	Lenguaje de Marcado de Hipertexto. Es el lenguaje estándar utilizado para crear y diseñar páginas web.
CSS	Hojas de Estilo en Cascada. Lenguaje utilizado para definir y crear la presentación del contenido estructurado en HTML o XML.
SCRUM	Marco de trabajo para el desarrollo ágil de software que organiza el desarrollo en ciclos repetitivos e incrementales llamados sprints.
MVP	Producto Mínimo Viable. Versión más simple de un producto que permite al equipo de desarrollo recolectar la máxima cantidad de aprendizajes validados sobre los clientes con el menor esfuerzo.
UX/UI	Experiencia de Usuario / Interfaz de Usuario. Referentes al diseño y la interactividad del producto, donde UX se enfoca en la usabilidad y la experiencia integral del usuario, y UI en la interfaz gráfica y visual que interactúa con el usuario.

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

**1.4. REFERENCIAS:** Se listan a continuación otros documentos a los que se hace referencia desde éste:

# TÍTULO	NÚMERO	FECHA
1 IEEE Guide for Software Requirements Specification IEEE	Std 830-84	1994
2 OMG Unified Modeling Language Specification Version 1.4	formal/2001-09-67	2001
3 C4 model Simon Brown	C4	2011

## **2. DESCRIPCIÓN GENERAL:**

El software a desarrollar es un sistema de gestión de citas técnicas diseñado específicamente para la administración de servicios de mantenimiento de aires acondicionados. Este sistema permitirá a los usuarios, ya sean administradores, técnicos o clientes, interactuar eficientemente para gestionar citas de mantenimiento preventivo, correctivo e instalaciones de equipos de aire acondicionado.

La plataforma será una aplicación web accesible a través de navegadores estándar en computadoras y dispositivos móviles. El sistema facilitará las siguientes funcionalidades clave:

- **Administración de citas:** Los usuarios podrán programar, modificar y cancelar citas técnicas. Los administradores podrán configurar la disponibilidad de los técnicos y gestionar el catálogo de servicios ofrecidos.
- **Notificaciones automáticas:** Tanto los técnicos como los clientes recibirán notificaciones automáticas vía email y WhatsApp sobre el estado y los detalles de sus citas programadas, cambios y recordatorios.
- **Gestión de usuarios:** El sistema permitirá a los administradores gestionar cuentas de usuarios, asignar roles específicos y controlar el acceso a diversas funciones del sistema.
- **Interfaz de usuario intuitiva:** Diseñada para ser fácil de usar, permitiendo a los usuarios navegar y realizar tareas en el sistema sin necesidad de formación técnica avanzada.

Este sistema busca optimizar la coordinación y la comunicación entre todos los participantes, mejorar la eficiencia operativa, reducir los tiempos de respuesta y aumentar la satisfacción del cliente. Además, se diseñará para ser escalable y seguro, asegurando que pueda adaptarse a las crecientes demandas del negocio y proteger la información sensible de los usuarios.

### **2.1. PERSPECTIVA DEL PRODUCTO:**

El producto de software, un sistema de gestión de citas técnicas para el mantenimiento de aires acondicionados, está diseñado para facilitar la organización y comunicación eficiente entre administradores, técnicos y clientes.

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

Se espera que el sistema cumpla con varias funciones críticas que abordan directamente las necesidades operativas y de servicio al cliente en la gestión de mantenimiento técnico. Las expectativas del producto incluyen:

- **Automatización de la Programación de Citas:** El sistema permitirá a los clientes elegir y programar servicios de mantenimiento a través de una interfaz web intuitiva, basándose en la disponibilidad real de los técnicos. Esto incluye la capacidad de seleccionar el tipo de servicio deseado (preventivo, correctivo, instalación) y la fecha y hora que mejor se ajusten a sus necesidades.
- **Notificaciones y Recordatorios:** Se espera que el producto envíe notificaciones automáticas y recordatorios a los clientes y técnicos mediante email y WhatsApp. Estas notificaciones incluirán confirmaciones de citas, recordatorios previos al servicio y alertas de cambios o cancelaciones.
- **Gestión y Seguimiento en Tiempo Real:** El software proporcionará a los administradores herramientas para monitorear y gestionar las actividades del día a día, incluyendo la asignación de tareas a técnicos, seguimiento del progreso de los trabajos y manejo de cambios en la programación. Esto ayudará a mantener un flujo de trabajo organizado y a maximizar la eficiencia operativa.
- **Interfaz Amigable y Accesible:** El diseño del sistema será centrado en el usuario, asegurando que la interfaz sea fácil de usar para personas con diferentes niveles de habilidad tecnológica. Esto es crucial para asegurar que los clientes y técnicos puedan interactuar con el sistema de manera efectiva sin necesidad de formación extensiva.

En resumen, se espera que nuestro producto de software simplifique y mejore la gestión de citas para servicios técnicos, aumente la satisfacción del cliente mediante una comunicación efectiva y proporcione a los administradores las herramientas necesarias para optimizar las operaciones y recursos disponibles.

**2.2. FUNCIONES DEL PRODUCTO:** Descripción del producto en base al Modelo C4 y detallar en casos de uso, como se representa a continuación.

### 2.2.1. Diagrama de contexto

	Proceso Actividad Tarea	
Producto	Especificación de requisitos de software	
Emitido por		Estado

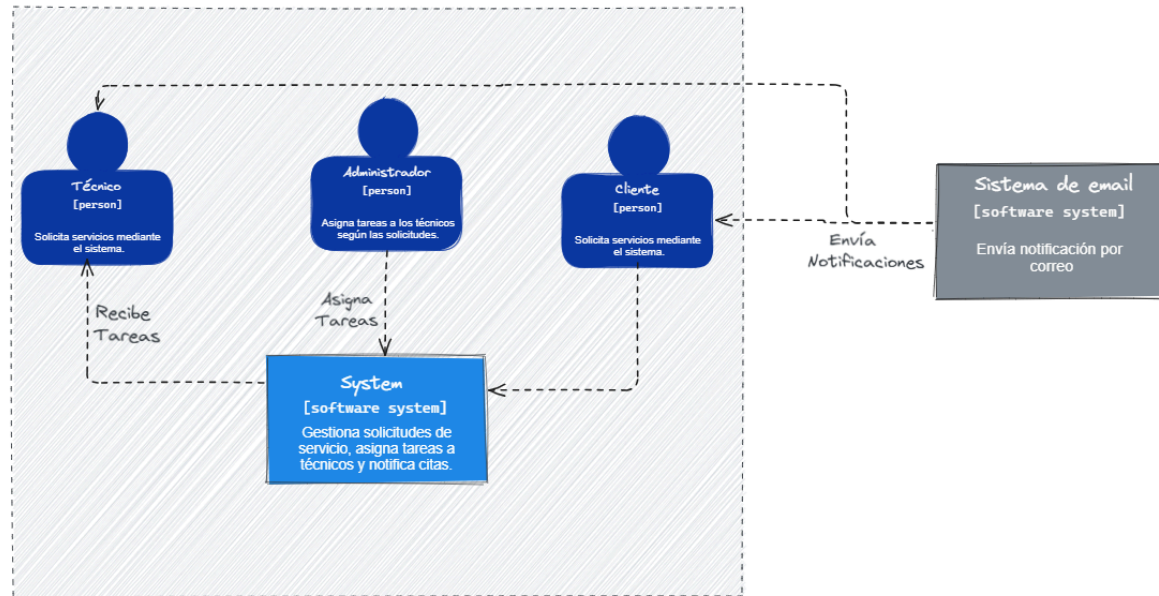


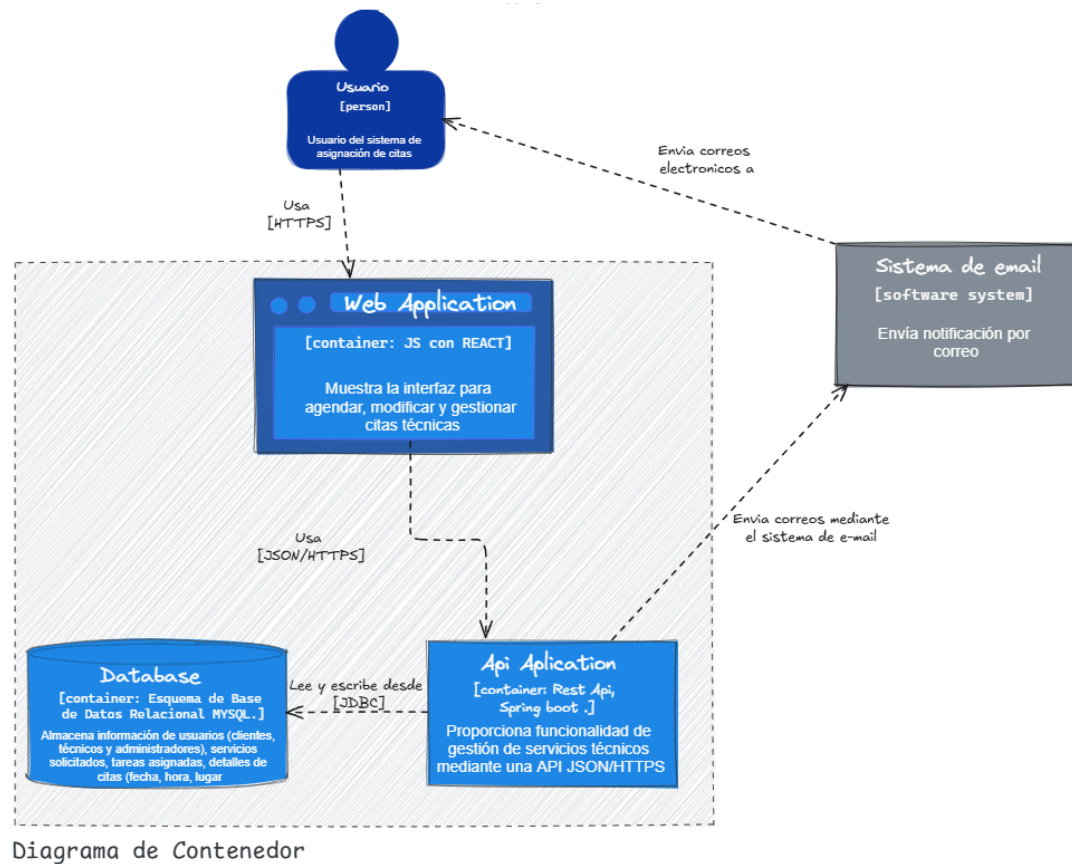
Diagrama de contexto

**Figura 1: Diagrama de contexto**

Un diagrama de Contexto del Sistema proporciona un punto de partida, mostrando cómo el sistema de software en el alcance encaja en el mundo que lo rodea.

### 2.2.2. Diagramas de contenedores

	Proceso Actividad Tarea	
Producto	Especificación de requisitos de software	
Emitido por		Estado



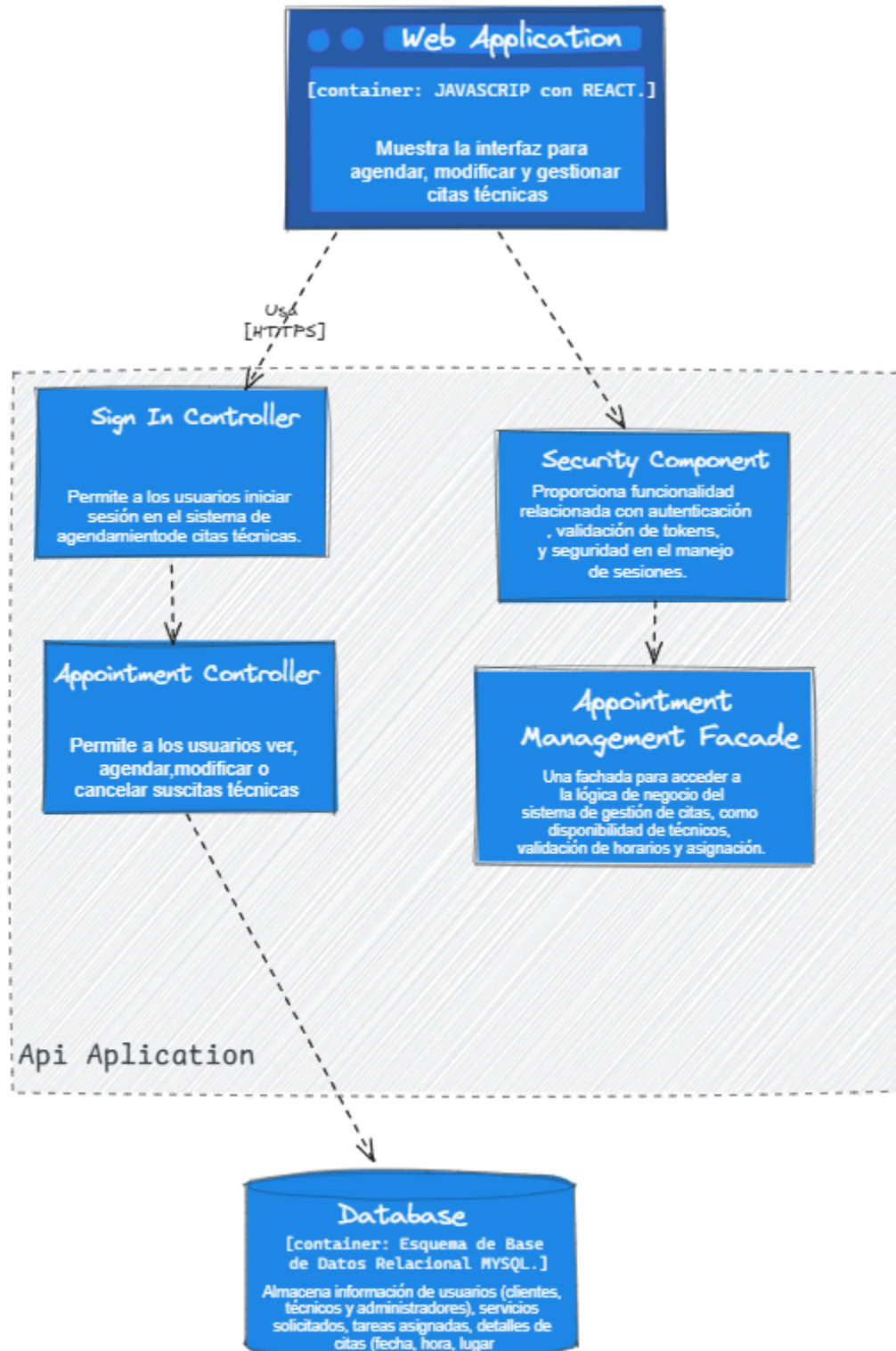
**Figura 2: Diagrama de contenedores**

Un diagrama de contenedor amplía el alcance del sistema de software y muestra los componentes básicos técnicos de alto nivel.

Es decir que pueden haber N cantidad de diagramas por contenedores dependiendo del modelo de negocio.

### 2.2.3. Diagrama de componentes

	Proceso Actividad Tarea	
Producto	Especificación de requisitos de software	
Emitido por		Estado



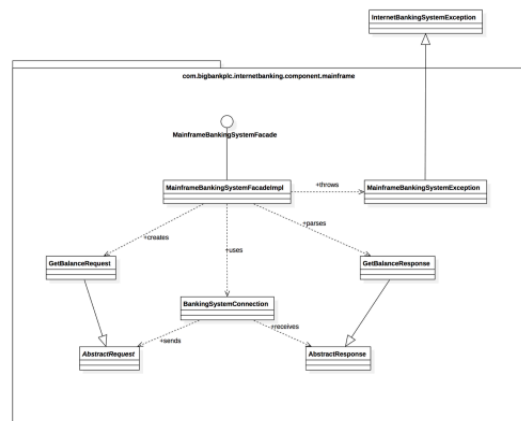
	Proceso Actividad Tarea	
Producto	Especificación de requisitos de software	
Emitido por		Estado

*Figura 3: Diagrama de componentes*

Un diagrama de componentes se acerca a un contenedor individual y muestra los componentes que contiene.

Es decir que pueden haber n cantidad de diagramas por componentes dependiendo del modelo de negocio.

#### 2.2.4. Diagrama de clases



*Figura 4: Diagrama de clases*

Se puede usar un diagrama de código (por ejemplo, clase UML) para ampliar un componente individual y mostrar cómo se implementa ese componente.

Es decir que pueden haber n cantidad de diagramas por componentes dependiendo del modelo de negocio y por cada diagrama de clases debe hacer un Caso de uso.

##### 2.2.4.1. HU

En la HU, es la descripción de una acción o actividad. Una HU se usa en la descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso, ver **Anexo 1- HU**.

#### 2.3. CARACTERÍSTICAS DE LOS USUARIOS



	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

En este literal se describe qué conocimientos básicos debe tener el usuario final o cliente para poder usar nuestro sistema, se pueden especificar los roles si es necesario, permisos y demás.

## 2.4. RESTRICCIONES

**2.4.1. Políticas de la empresa:** Son todas las restricciones que salen del modelo de negocio, como se observa en el siguiente ejemplo:

Nota: si no presenta escribir “No se contemplan”

**2.4.2. Limitaciones del hardware:** son las características mínimas que debe tener el hardware, para poder instalar o correr nuestro software ejemplo: Linux, CPU 16, RAM 128 GB, etc. Si no se presentan escribir “No se contemplan”.

**2.4.3. Operaciones paralelas:** La computación paralela es una forma de cómputo en la que se hace uso de 2 o más procesadores para resolver una tarea. La técnica se basa en el principio según el cual, algunas tareas se pueden dividir en partes más pequeñas que pueden ser resueltas simultáneamente, ejemplo:

Nota: si no presenta escribir “No se contemplan”.

**2.4.4. Funciones de auditoría:** Las funciones de auditoría se instauran para controlar y verificar los sistemas informáticos internos, Analizar los riesgos del entorno informático, sistemas operativos, redes y telecomunicaciones, estas funciones suelen estar basadas en comprobaciones del software. si no presenta escribir “No se contemplan”

**2.4.5. Funciones de control:** Las funciones de control permiten tener garantías sobre el software y sus datos, ejemplo:

Nota: si no presenta escribir “No se contemplan”.

**2.4.6. Lenguajes de programación:** Son todos los lenguajes de programación, frameworks y/o herramientas tecnológicas usadas en la elaboración del sistema, ejemplo:

- JavaScript
- PHP

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

- MySQL
- Angular
- TypeScript

Nota: si no presenta escribir "No se contemplan".

**2.4.7. Protocolos de comunicación:** Los protocolos de comunicaciones entre los diferentes nodos de la infraestructura hardware de soporte serán los siguientes, ejemplo:

- HTTPS para conexiones con el servidor web.
- TCP/IP a nivel físico.
- Ethernet 802.3 a nivel eléctrico.

**2.4.8. consideraciones de seguridad:** Son todas las consideraciones y atributos de seguridad dentro de nuestro software, deben también estar incluidos en los casos de uso, como por ejemplo:

- Contraseña robusta.
- Logueo de 2 pasos.
- Bloqueos de usuarios por IP.

Nota: si no presenta escribir "No se contemplan".

**2.4.9. Criticalidad de la aplicación:** En este literal se debe instaurar o explicar qué debe hacer la aplicación o el software en caso de que se presente algún problema o falle, ejemplo:

**2.5. SUPOSICIONES Y DEPENDENCIAS:** En la mayoría de los casos nuestras aplicaciones pueden estar expuestas para su consumo desde otros software, así mismo, nuestro software puede tener dependencias internas o consumos de otros sistemas, como por ejemplo:

- Nuestro sistema expone el total de usuarios nuevos, para que sirva de estadística para un app de marketing.
- Nuestro producto consume una API orientada a las finanzas para poder realizar las conversiones de moneda.

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

**2.6. REQUERIMIENTOS FUTUROS:** Son todos los requerimientos que no se tendrán en cuenta en esta versión del proyecto o producto.

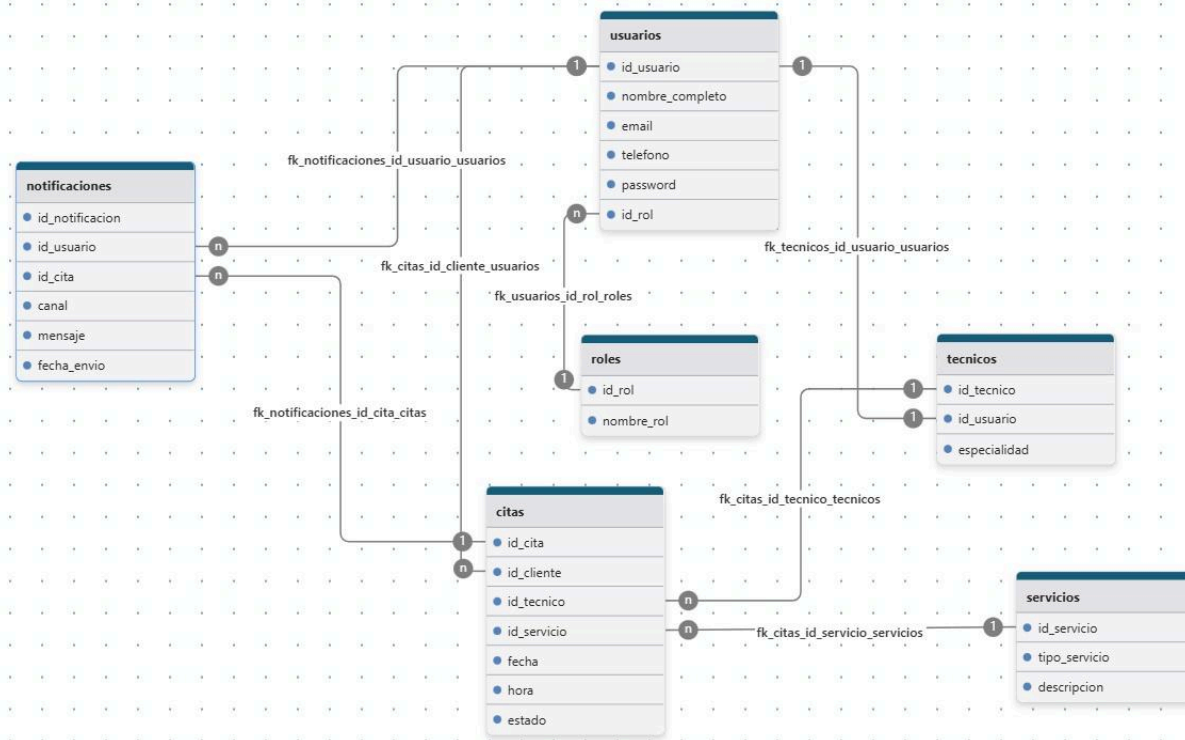
**3. INTERFACES EXTERNAS:** Son todas las interfaces que nuestro sistema debe proveer o en su defecto usar para la comunicación con el mismo o con software externos, estas interfaces deben ser a nivel de usuario, hardware, software y de comunicación.

- 3.1. Interfaces de Usuario**
- 3.2. Interfaces Hardware**
- 3.3. Interfaces Software**
- 3.4. Interfaces de Comunicación**
- 3.5. Modelo Lógico de Datos (MER):**

Enlace a los documentos:

<https://drive.google.com/drive/folders/1KQL48arU-5-eggA0DXoSJDmg3VXkRnkS?usp=sharing>

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>



#### 4. REQUERIMIENTOS DE RENDIMIENTO: requerimientos orientados al rendimiento y/o performance de nuestra aplicación

- Número de usuarios conectados en simultáneo.
- Número de peticiones al servidor.
- Número de descargas en simultáneo.
- Número de subidas de archivos en simultáneo.

5. **REQUERIMIENTOS ESPECÍFICOS:** En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema, Todos los requisitos aquí expuestos son esenciales, es decir, no será aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. **HU**

6. **CRITERIOS DE ACEPTACIÓN:** Son las condiciones que un producto de software debe satisfacer para ser aceptado por un usuario, cliente o stakeholder,

	<b>Proceso Actividad Tarea</b>	
<b>Producto</b>	<b>Especificación de requisitos de software</b>	
<b>Emitido por</b>		<b>Estado</b>

en ese orden de ideas se deben agrupar todos los requisitos tanto funcionales como no funcionales expuestos en este documento a manera de check en el **HU**