

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Sistema de Compras y Abastecimiento

Avance 2

**Integrantes: Duvan Fuentes Alvarez
Nicolas Lampe Huenul**

Asignatura: Proyecto 4

Profesor(es): Diego Aracena Pizarro

ARICA, 26 DE NOVIEMBRE DE 2022

CONTENIDOS

1. INTRODUCCIÓN	4
2. Objetivos	5
2.1. Objetivo General	5
2.2 Objetivos Específicos	5
3. Definición del proyecto	6
3.1 Contexto	6
3.2 Problema o necesidad	6
3.3 Solución	6
3.4 Definición de requisitos	7
3.4.1 Requisitos funcionales	7
3.4.1 Requisitos no funcionales	8
4. PLANIFICACIÓN DEL PROYECTO	
4.1 Carta Gantt	9
4.2 Metodología de desarrollo	10
4.3. Herramientas	11
5. Casos de uso	15
6. Diagrama BPM	21
6.1 Diagrama BPMN “Aceptar solicitud”	22
7. Conclusión	23

TABLAS

Tabla 1: Requisitos funcionales.	7
Tabla 2: Requisitos no funcionales.	8
Tabla 3: Caso de Uso "Crear solicitud".	16
Tabla 4: Caso de Uso "Editar solicitud".	17
Tabla 5: Caso de Uso "Ver seguimiento".	18
Tabla 7: Caso de Uso "Solicitud etapa".	19
Tabla 8: Caso de Uso "Confirmar etapa".	19
Tabla 9: Caso de Uso "Rechazar etapa".	20

FIGURAS

Figura 1: Carta Gantt.	9
Figura 2: Modelo de contexto.	12
Figura 3: Diagrama de interacción	13
Figura 4: Modelo de contexto	14
Figura 5: Modelo de contexto	14
Figura 6: Diagrama de casos	15
Figura 7: Diagrama BPMN "Aceptar solicitud".	22

1. INTRODUCCIÓN

Este documento presenta la planificación y diseño de un sistema de software destinado a gestionar eficientemente las solicitudes de compras en la Facultad de Odontología. El objetivo principal es desarrollar un sistema funcional que cumpla con los requisitos específicos del cliente, utilizando Laravel para el backend y React para el frontend.

El sistema se centrará en la gestión de solicitudes de compras y en el proceso de aprobación y flujo de trabajo. Se utilizarán componentes reutilizables en React para optimizar la interfaz y garantizar la consistencia en la entrada de datos. Además, se integrarán funcionalidades en Laravel para gestionar el proceso de aprobación de solicitudes de compras y se configurarán flujos de trabajo personalizables.

El proyecto se alojará en GitHub para facilitar la colaboración, el control de versiones y el seguimiento del progreso del desarrollo. La elección de Laravel y React se basa en su productividad, la reutilización de componentes, la amplia adopción en la industria y la abundante documentación y soporte de la comunidad.

En resumen, este proyecto busca implementar un sistema de información eficiente y fácil de usar que mejore la gestión de las solicitudes de compras en la Facultad de Odontología. Se espera que el sistema resultante sea intuitivo, eficiente y adaptable a las necesidades cambiantes de los usuarios.

2. Objetivos

2.1. Objetivo General

El objetivo general de este proyecto es desarrollar e implementar un sistema de software eficiente y funcional que cumpla con los requisitos específicos del cliente, utilizando las mejores prácticas de desarrollo de software y herramientas tecnológicas modernas. El proyecto estará alojado en GitHub para facilitar la colaboración, control de versiones y seguimiento del progreso del desarrollo.

2.2 Objetivos Específicos

- Seleccionar herramientas o frameworks de software adecuados para la implementación del proyecto, considerando las necesidades y requisitos específicos.
- Establecer el repositorio del proyecto en GitHub, asegurando un control de versiones efectivo y una colaboración eficiente.
- Definir el alcance del producto a desarrollar, detallando las funcionalidades clave de acuerdo con las herramientas y frameworks seleccionados.
- Crear diagramas de interacción o diagramas colaborativos de subsistemas para cada caso de uso, identificando las secuencias de interacciones entre los objetos participantes. Incluir rutas alternativas en los diagramas de interacción.
- Modelar los datos del sistema y seleccionar la herramienta para el almacenamiento de datos, asegurando un diseño eficiente y coherente con los requisitos del proyecto.
- Implementar el sistema con un enfoque de refinamiento sucesivo, permitiendo la iteración y mejora continua.
- Garantizar que cada diagrama colaborativo de subsistemas sea coherente con los diagramas de interacción correspondientes, incluyendo todas las rutas alternativas identificadas.
- Describir de manera general el modelo de análisis, destacando las decisiones de diseño tomadas y cualquier suposición realizada durante el proceso.

3. Frameworks

Los frameworks que utilizaremos serán dos, Laravel para el backend y React para el frontend. A continuación se detallará los motivos de la elección:

Laravel

Productividad y Desarrollo Rápido: Laravel proporciona numerosas herramientas y funciones listas para usar, lo que acelera el desarrollo y reduce el tiempo de implementación. Esto es esencial en un sistema de información, donde la velocidad de desarrollo es crucial para cumplir con los plazos y los cambios rápidos en los requisitos.

Comunidad Activa y Documentación Abundante: Laravel cuenta con una comunidad activa de desarrolladores y una documentación extensa. Esto facilita la resolución de problemas, la obtención de soporte y la adopción de las mejores prácticas en el desarrollo del sistema de información.

React:

Reutilización de Componentes: React sigue el concepto de componentes, lo que facilita la reutilización de código. En un sistema de información, donde es común tener elementos de interfaz similares en diferentes partes del sistema, la reutilización de componentes agiliza el desarrollo y garantiza la coherencia visual.

Amplia Adopción en la Industria: React es ampliamente adoptado en la industria y respaldado por Facebook. Esta popularidad asegura la disponibilidad de recursos, bibliotecas y herramientas de terceros que pueden ser útiles en el desarrollo y mantenimiento del sistema de información.

Al elegir Laravel para el backend y React para el frontend, se logra un equilibrio entre la eficiencia en el desarrollo, la claridad en el código, la experiencia del usuario y la capacidad de mantenimiento, elementos fundamentales en la implementación de un sistema de información.

3. Alcance

El alcance del proyecto, es lo siguiente:

1. **Selección de Herramientas y Frameworks:** El proyecto implica la selección de las herramientas o frameworks de software más adecuados para satisfacer las necesidades y requisitos específicos del cliente.
2. **Control de Versiones y Colaboración:** Se establecerá un repositorio en GitHub para el proyecto, lo que permitirá un control de versiones efectivo y facilitará la colaboración entre los miembros del equipo de desarrollo.
3. **Creación de Diagramas de Interacción:** se crearán diagramas de interacción o diagramas colaborativos de subsistemas, identificando las secuencias de interacciones entre los objetos participantes.
4. **Modelado de Datos:** Se modelaron los datos del sistema y se seleccionará la herramienta adecuada para el almacenamiento de datos, asegurando un diseño eficiente y coherente con los requisitos del proyecto.
5. **Implementación del Sistema:** El sistema se implementará utilizando un enfoque de refinamiento sucesivo, lo que permitirá la iteración y la mejora continua a lo largo del desarrollo del proyecto.
6. **Coherencia entre Diagramas:** Se garantizará que cada diagrama colaborativo de subsistemas sea coherente con los diagramas de interacción correspondientes, incluyendo todas las rutas alternativas identificadas.
7. **Descripción del Modelo de Análisis:** Se describe de manera general el modelo de análisis, destacando las decisiones de diseño tomadas y cualquier suposición realizada durante el proceso.

4. Diagrama de interacción

A continuación se visualiza el diagrama de interacción (**ver figura 1**). en el que por medio de un navegador web, se interactúa con el subsistema interfaz de usuario y este solicita datos de manera asíncrona a la base de datos y la base de datos le entrega los datos pertinentes, ya se para validar credenciales de autenticación, información de solicitudes o información de usuario.

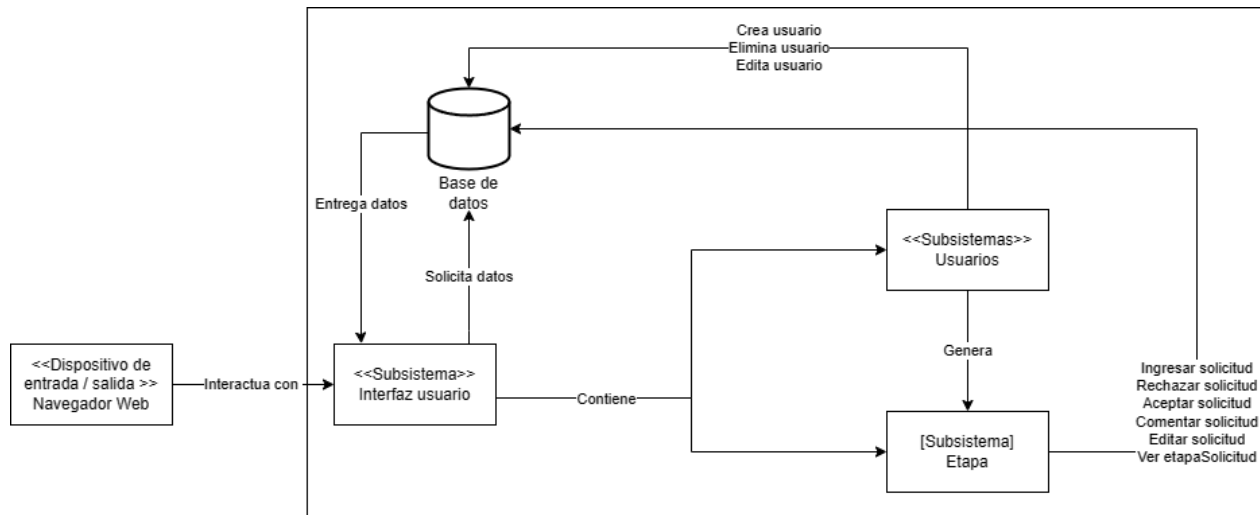


Figura 1: Diagrama de interacción.

5. Modelamiento de datos

MongoDB es una base de datos NoSQL, lo que significa que no utiliza el modelo relacional que se encuentra en las bases de datos SQL tradicionales[1]. En lugar de tablas, filas y columnas, MongoDB utiliza una estructura de datos basada en documentos. Estos documentos se almacenan en BSON, que es una representación binaria de JSON [2].

En MongoDB, lo que serían las “tablas” en una base de datos relacional, se llaman “colecciones”. Cada colección contiene documentos, que son registros de datos. Cada documento puede tener un esquema diferente, lo que significa que los documentos dentro de una misma colección no necesariamente tienen que tener la misma estructura, los mismos campos ni el mismo tipo de datos para cada campo[3].

Aunque no se puede crear un diagrama relacional para MongoDB como si se podría para una base de datos SQL, se puede crear un bosquejo o diagrama de documentos para visualizar cómo están estructurados los datos. El bosquejo en el que nos basamos es el siguiente (ver figura 2):

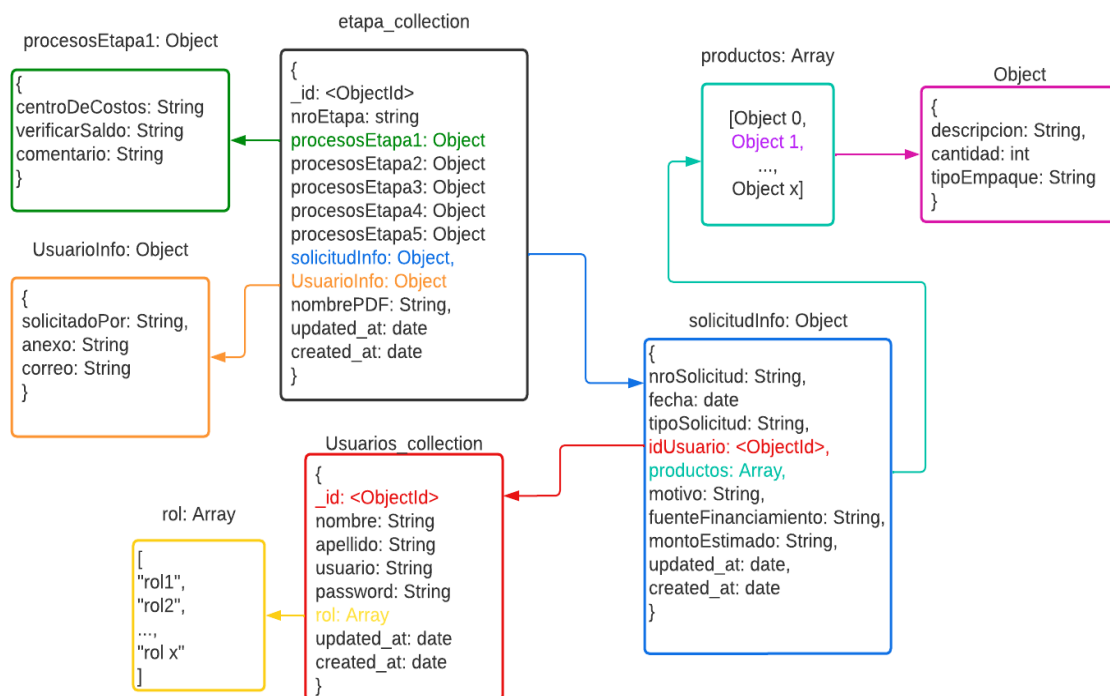


Figura 2: Diagrama de interacción.

6. Implementación y refinamiento sucesivo

Diagrama de comunicación

En la figura 4 se muestra el diagrama de comunicación de una creación de una solicitud.

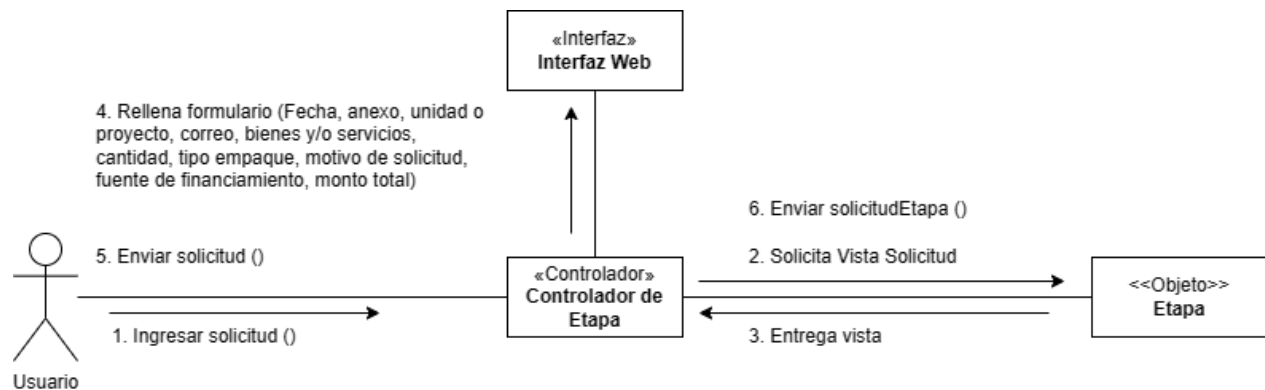


Figura 4: Diagrama de comunicación "Crear solicitud"

Una vez creada una solicitud para avanzar de etapa en etapa es necesario confirmarlas para ello en esta figura 5 Se presenta el diagrama de comunicación de "Aceptar solicitud".

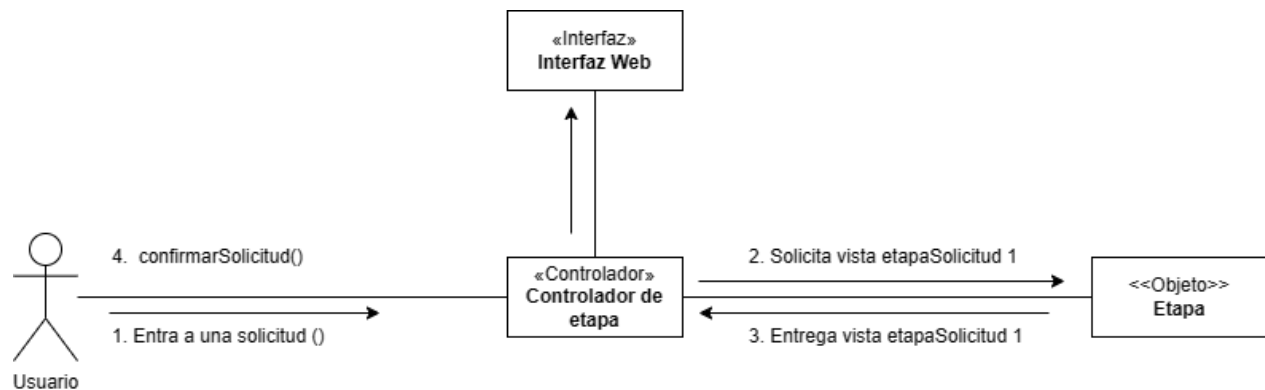


Figura 5: Diagrama de comunicación "Confirmar Etapa"

El flujo alternativo que se ha contemplado es rechazar una solicitud (**ver figura 6**), para ello el usuario rechaza la solicitud, indica el motivo de rechazo y la solicitud ingresa como rechazada.

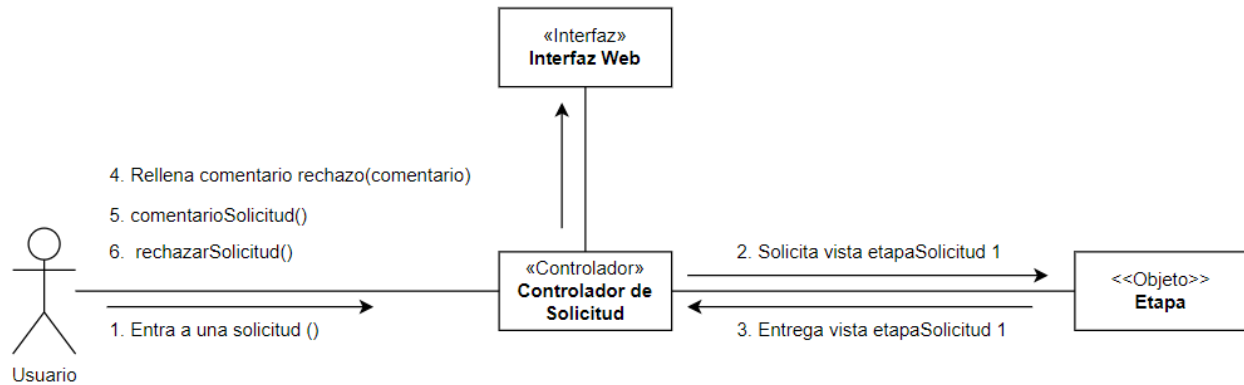


Figura 6: Diagrama de comunicación "Rechazar Etapa"

Si la solicitud fue rechazada en la etapa de chequeo (**ver figura 7**), el usuario que realizó la solicitud, podrá acceder a su solicitud rechazada y editarla en base a los motivos del rechazo de su solicitud.

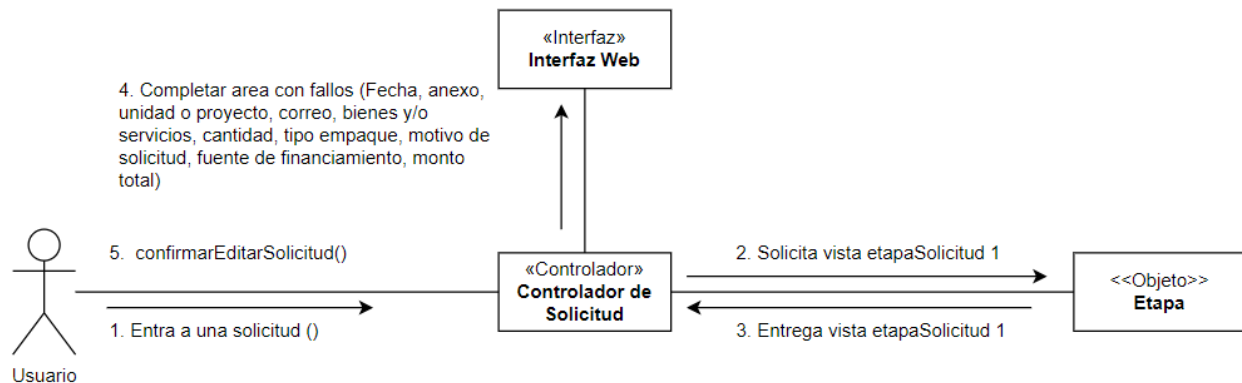


Figura 7: Diagrama de comunicación "Editar Etapa"

Todos los usuarios, tienen acceso a ver las solicitudes (**ver figura 8**), ver una solicitud, es una vista que contempla todas las etapas por las que ha pasado una solicitud, por lo que se podrá ver toda la información y las etapas que se han hecho una solicitud.

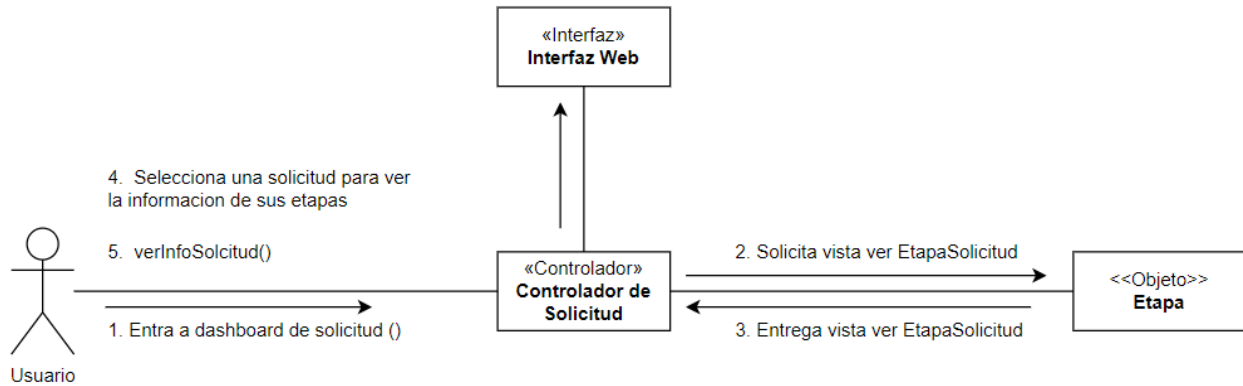


Figura 8: Diagrama de comunicación "Ver información solicitud"

En algunos casos particulares, el administrador y solo él, podrá eliminar solicitudes (**ver figura 9**). Sin ningún tipo de restricción para eliminarla.

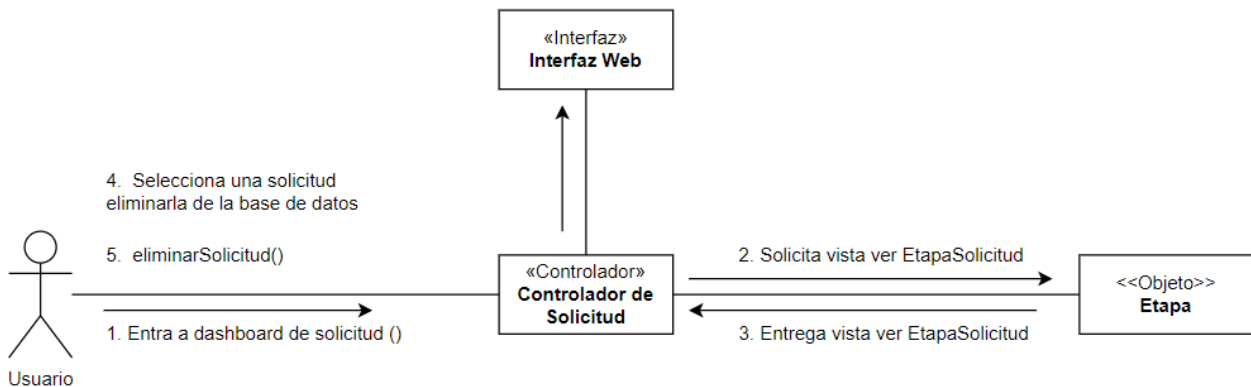


Figura 9: Diagrama de comunicación "eliminar solicitud"

7. Conclusión

En conclusión, el sistema descrito en el documento parece ser un sistema robusto y bien pensado para manejar solicitudes de usuarios. Los diagramas de interacción y comunicación proporcionan una visión clara de cómo los usuarios interactúan con el sistema y cómo se procesan las solicitudes. La implementación de MongoDB para la base de datos no relacional permite una gran flexibilidad en la gestión de los datos.

El sistema también contempla varios flujos de trabajo, incluyendo la creación de solicitudes, la confirmación de etapas, el rechazo de solicitudes y la edición de solicitudes rechazadas. Además, proporciona a todos los usuarios la capacidad de ver las solicitudes y a los administradores la capacidad de eliminar solicitudes.

Esto demuestra una consideración cuidadosa de las diferentes necesidades y roles de los usuarios dentro del sistema. Sin embargo, como con cualquier sistema, es esencial realizar pruebas exhaustivas y obtener retroalimentación de los usuarios para garantizar que el sistema funcione según lo previsto y satisfaga las necesidades de los usuarios.

8. REFERENCIAS

- [1] MongoDB documentation. (s/f). Mongoddb.com; MongoDB. Recuperado el 26 de noviembre de 2023, de <https://www.mongodb.com/docs/>
- [2] What is MongoDB? (s/f). Mongoddb.com. Recuperado el 26 de noviembre de 2023, de <https://www.mongodb.com/docs/manual/>
- [3] Mukherjee, S. (2019). The battle between NoSQL databases and RDBMS. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.3393986>
- [4] Profesor R. Herrera, Apuntes del curso de Ingeniería de Software, 2022.
- [5] Profesor M. Villalobos, PROYECTOS DE DESARROLLO DE SOFTWARE (Versión 1.0).