

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE**



**“SISTEMA COLABORATIVO PARA LA
REDACCIÓN DE CONTRATOS MEDIANTE EL
SERVICIO AZURE FLUID RELAY”**

DIEGO ALBERTO SAA BRUNA

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

**Profesor Guía: Cecilia Reyes Covarrubias
Correferente: Carlos Chesta Rivas**

Abril - 2024

DEDICATORIA

Con todo cariño y admiración
dedico este trabajo a mis padres.
Cuando los miro, veo esfuerzo,
veo ese sacrificio silencioso de tantos años,
veo las huellas del sol en su piel
y el paso del tiempo en sus rostros.
Muchas veces vivimos pensando en crecer
y olvidamos que ellos envejecen.
Este logro también es de ellos.

AGRADECIMIENTOS

Quiero dedicar unas líneas para agradecer a todas las personas que han sido parte de este viaje conmigo.

A mis amigos y compañeros, les agradezco por su compañía y apoyo incondicional. Como siempre les digo, un barco no se rema solo, y ustedes han sido los marineros que han hecho más llevadera esta travesía.

Agradezco a mi familia por estar siempre ahí, especialmente a mis padres Rosa y Rubén, y mis hermanos Santiago y Melissa, brindándome su constante apoyo.

También deseo expresar mi gratitud a mis compañeros de trabajo Carlos y Javier, integrantes del equipo que llevó a cabo este proyecto.

Y por supuesto, a todos los profesores tanto del Colegio como de la Universidad que han contribuido a mi formación y crecimiento.

RESUMEN

Resumen—Actualmente, la falta de transparencia y cambios no autorizados en contratos ocasiona importantes disputas financieras y legales en el ámbito empresarial, en especial dentro del rubro de la minería. Para abordar este problema, se plantea la creación de un sistema colaborativo que facilite la negociación, comunicación e interacción entre las partes interesadas al momento de redactar un contrato con proveedores. Este sistema, se integrará con SOFI (Sistema Online de Firma de Contratos), un sistema de gestión de contratos desarrollado para una importante empresa minera en Chile, agregando funcionalidades para los roles de Ingenieros de Contratos, Representantes de Área Legal, Abastecimiento y Proveedores. Además, los componentes de interfaz usuaria desarrollados para fomentar la colaboración se incorporarán al Envision Framework, una biblioteca de código utilizada para el desarrollo de software en la empresa patrocinadora de esta memoria, permitiendo así, su implementación en proyectos futuros de manera sencilla y eficaz.

Palabras Clave— Azure Fluid Relay; Desarrollo de Software; Sistema Colaborativo; Contratos.

ABSTRACT

Abstract—Currently, the lack of transparency and unauthorized changes in contracts cause important financial and legal disputes in the business environment, especially within the mining sector. To address this problem, the creation of a collaborative system is proposed that facilitates negotiation, communication and interaction between interested parties when drafting a contract with suppliers. This system will be integrated with SOFI (Online Contract Signature System), a contract management system developed for an important mining company in Chile, adding functionalities for the roles of Contract Engineers, Legal Area Representatives, Supply and Suppliers. In addition, the user interface components developed to promote collaboration will be incorporated into the Envision Framework, a code library used for software development in the company sponsoring this report, thus allowing their implementation in future projects in a simple and effective way.

Keywords— Azure Fluid Relay; Software Development; Collaborative System; Contracts.

GLOSARIO

A continuación, se definirán las siglas mencionadas en el trabajo y su explicación, por orden alfabético:

- A
 - AA: Área de Abastecimiento
 - AL: Área Legal
 - ALC: Área Legal Corporativa
- B
 - BI: *Business Intelligence*, o en español, Inteligencia de Negocios.
 - BPMN: *Business Process Model and Notation*, o en español, Modelo y Notación de Procesos de Negocio.
- D
 - DOM: *Document Object Model*, o en español, Modelo de Objetos del Documento
- E
 - EFW: Envision FrameWork
- F
 - F7: Framework 7
- H
 - HTML: *HyperText Markup Language*, o en español, Lenguaje de Marcado de Hipertexto
- I
 - IA: Inteligencia Artificial
 - IdC: Ingeniero de Contratos
- L
 - LabUX: Laboratorio de Experiencia de Usuario
- M
 - ML: *Machine Learning*, o en español, Aprendizaje de Máquina o Aprendizaje Automático.
- N
 - NDA: *Non-Disclosure Agreement*, o en español, Acuerdo de No Divulgación
 - NPM: *Node Package Manager*, o en español, Manejador de Paquetes de Node

- P
 - PA: Power Automate
- S
 - SOFI: Sistema Online de Firma de Contratos
 - SPO: Acrónimo de SharePoint Online
 - SUS: System Usability Scale, o en español, Escala de Usabilidad del Sistema
- U
 - UTFSM: Universidad Técnica Federico Santa María
 - UI: *User Interface*, o en español, Interfaz de Usuario
 - UX: *User eXperience*, o en español, Experiencia del Usuario
- Q
 - QA: *Quality Assurance*

ÍNDICE DE CONTENIDO

RESUMEN.....	4
ABSTRACT.....	4
GLOSARIO.....	5
ÍNDICE DE CONTENIDO.....	7
ÍNDICE DE FIGURAS.....	10
ÍNDICE DE TABLAS.....	12
INTRODUCCIÓN.....	13
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA.....	15
1.1 Contexto.....	15
1.2 Situación Actual.....	16
1.3 Problema a Resolver.....	17
1.3.1 Árbol del Problema.....	18
1.4 Objetivos.....	20
1.4.1 Objetivo General.....	20
1.4.2 Objetivos Específicos.....	20
1.5 Alcance y Limitaciones.....	20
CAPÍTULO 2: MARCO CONCEPTUAL.....	21
2.1 Algoritmo.....	21
2.1.1 Algoritmos de Consenso.....	21
2.1.2 Algoritmos de Exclusión Mutua Distribuida.....	21
2.2 Contrato.....	21
2.2.1 Cláusula.....	21
2.3 BPMN.....	22
2.4 Experiencia de Usuario (UX) e Interfaz de Usuario (UI).....	22
2.4.1 Pruebas de Usabilidad.....	22
2.4.2 Entrevista.....	24
2.4.3 System Usability Scale (SUS).....	24
2.5 Groupware.....	24
2.6 Rol.....	24
2.7 Tecnologías.....	25
2.7.1 Envision Framework.....	25
2.7.2 Microsoft 365.....	25
2.7.3 Microsoft Power Automate.....	25
2.7.4 Microsoft SharePoint.....	25
2.7.5 Microsoft Azure.....	25

2.7.6 Azure Fluid Relay.....	25
2.7.7 Fluid Framework.....	25
CAPÍTULO 3: ESTADO DEL ARTE.....	26
3.1 Algoritmos Concurrentes.....	26
3.1.1 Reloj Lógico de Lamport.....	26
3.1.2 Vector Lógico de Lamport.....	28
3.2 Algoritmos de Consenso y Exclusión Mutua.....	29
3.2.1 Algoritmo de Berkeley.....	29
3.2.2 Algoritmo del Semáforo de Dijkstra.....	32
3.2.3 Spinlocks con Test And Set.....	33
3.2.4 Algoritmo de Peterson.....	35
3.3 Sistema de Mensajería y Colaboración.....	35
3.3.1 Mensajería Instantánea en Redes Sociales y Chatbots.....	36
3.3.2 Colaboración Empresarial.....	36
3.3.3 API's de Mensajería.....	37
3.4 Conflicto de Versiones y Colisiones en la escritura.....	37
3.4.1 Last Write Wins (LWW).....	38
3.4.2 Librerías de Combinación de Texto.....	38
3.5 Análisis del Estado del Arte.....	39
3.5.1 Análisis Comparativo.....	40
CAPÍTULO 4: PROPUESTA DE SOLUCIÓN.....	43
4.1 Propuesta de Diseño.....	43
4.1.1 Diagrama de Procesos.....	44
4.1.2 Arquitectura de la Solución.....	47
4.1.3 Propuesta de Diseño no Funcional.....	48
4.1.4 Diagrama de Casos de Uso.....	52
4.2 Editor Colaborativo.....	54
4.2.1 Creación y eliminación de Secciones del Contrato de forma dinámica.....	54
4.2.2 Control de Acceso a Sección Crítica con Azure Fluid Relay.....	57
4.2.3 Avatares.....	60
4.2.4 Manejo de Inactividad del Usuario.....	60
4.2.5 Guardar Cambios y Colisiones.....	63
4.2.6 Revisiones.....	64
4.3 Sistema de Mensajería.....	66
4.4 Máquina Virtual.....	67
4.5 Implementar Componentes UI en Envision Framework.....	68
4.5.1 Conexión a Azure Fluid Relay.....	68

4.5.2 Chat.....	68
4.5.3 Documentación.....	69
4.5.4 Funcionalidades e Interfaz de Usuarios.....	73
4.5.4.1 Múltiples Usuarios Conectados.....	73
4.5.4.2 Revisar perfil de los Usuarios.....	73
4.5.4.3 Compartir Archivos.....	74
4.5.4.4 Reaccionar a mensajes.....	75
4.5.4.5 Responder Mensajes.....	76
4.5.4.6 Eliminar mensajes.....	77
CAPÍTULO 5: VALIDACIÓN DE LA SOLUCIÓN.....	78
5.1 Feedback QA.....	78
5.2 Descripción del Proceso de Validación.....	78
5.2.1 Invitación y Consentimiento Informado.....	78
5.2.2 Entrevista Inicial.....	79
5.2.3 Prueba de Usabilidad.....	79
5.2.4 Formulario System Usability Scale (SUS).....	79
5.2.5 Entrevista Final.....	80
5.3 Respuestas y Resultados.....	80
5.3.1 Entrevistas.....	81
5.3.2 Prueba de Usabilidad.....	83
5.3.3 Formulario SUS.....	89
5.3.3.1 Respuestas.....	90
CAPÍTULO 6: CONCLUSIONES.....	92
6.1 Cumplimiento de Objetivos.....	92
6.2 Contribuciones y Aplicaciones del Trabajo Realizado.....	94
6.3 Trabajo Futuro.....	95
6.4 Comentarios Personales.....	95
REFERENCIAS BIBLIOGRÁFICAS.....	97
ANEXOS.....	99
Invitación para Participar en Prueba de Usabilidad.....	100
Consentimiento Informado.....	101
Validación de Usabilidad.....	102

ÍNDICE DE FIGURAS

A continuación, se presenta el Índice de Figuras utilizadas en este documento.

Figura 1: Actos Impugnados 2010-2016.....	16
Figura 2: Tipos de Contratos Licitados 2010-2016.....	17
Figura 3: Porcentaje de Entidades demandadas por Sector 2010-2016.....	17
Figura 4: Árbol del Problema.....	18
Figura 5: Proceso de una Prueba de Usabilidad.....	23
Figura 6: Ejemplo de relaciones entre distintos eventos.....	27
Figura 7: Valor del reloj lógico para cada evento.....	28
Figura 8: Relojes Vectoriales de Lamport iniciados con todos sus elementos en 0.....	28
Figura 9: Relojes Vectoriales de Lamport.....	29
Figura 10: Servidor inicia algoritmo difundiendo su reloj $T_0 = 10$ [UT].....	30
Figura 11: Nodos clientes notifican las diferencias en el reloj D_i al Servidor S.....	30
Figura 12: Servidor S notifica los ajustes.....	31
Figura 13: Clientes ajustan sus relojes.....	31
Figura 14: Algoritmo del Semáforo de Dijkstra.....	33
Figura 15: Diagrama BPMN Algoritmo Spinlock con Test and Set.....	34
Figura 16: Algoritmo de Peterson.....	35
Figura 17: Avatares de Google Docs.....	37
Figura 18: Ejemplo de LWW.....	38
Figura 19: Diagrama de Procesos SOFI Actual.....	45
Figura 20: Diagrama de Procesos SOFI mejorado - Etapa de Redacción.....	46
Figura 21: Arquitectura de la Solución.....	47
Figura 22: Interfaz del Editor Colaborativo para el Administrador SOFI.....	49
Figura 23: Panel de Cláusulas.....	50
Figura 24: Interfaz del Editor Colaborativo - Sección que requiere revisión.....	51
Figura 25: Diagrama de Casos de Uso.....	52
Figura 26: Editor activo con botones “Agregar Sección” y “Eliminar Sección”.....	55
Figura 27: Lógica de Crear Sección del Contrato en ambiente colaborativo.....	56
Figura 28: Revisión de Estados de la Sección del Contrato.....	58
Figura 29: Ejemplo de sección en estado NO editable.....	58
Figura 30: Ejemplo de sección en estado editable.....	59
Figura 31: Ejemplo de sección durante la edición.....	59
Figura 32: Avatares de Usuarios Conectados.....	60

Figura 33: Notificar actividad durante proceso de edición.....	61
Figura 34: Desactivar modo edición al detectar inactividad.....	62
Figura 35: Liberar SC al detectar inactividad sin desactivar el modo edición.....	62
Figura 36: Conflicto al guardar cambios de la Sección del Contrato.....	63
Figura 37: Etiquetas con información sobre la revisión de la Sección.....	65
Figura 38: Sección que requiere revisión vista como Área Legal.....	65
Figura 39: Chat Interno y Externo.....	66
Figura 40: Lógica de invitación y control de acceso al proveedor.....	67
Figura 41: Ver perfil del usuario.....	73
Figura 42: Compartir archivos.....	74
Figura 43: Documentos enviados.....	74
Figura 44: Reaccionar a un mensaje.....	75
Figura 45: Responder un mensaje.....	76
Figura 46: Eliminar un mensaje.....	77
Figura 47: Representación de los resultados de un SUS.....	80
Figura 48: Problemas que aborda la aplicación según los usuarios.....	82
Figura 49: Corrección error en editor Tarea 4.....	85
Figura 50: Listado de funcionalidades en el botón “Opciones” de Tarea 5.....	86
Figura 51: Listado de funcionalidades en el botón “Más Funcionalidades”.....	87
Figura 52: Popup con listado de usuarios invitados al ambiente colaborativo.....	88
Figura 52: Historial de Cambios en Popup Registro de Actividad.....	89
Figura 53: Respuestas Formulario SUS.....	90

ÍNDICE DE TABLAS

A continuación, se presenta el Índice de Tablas utilizado en este documento.

Tabla 1: Escala para criterios en la evaluación de algoritmos.....	41
Tabla 2: Análisis de los algoritmos.....	42
Tabla 3: Evolución de los cambios en la Sección del Contrato.....	64
Tabla 4: Importar módulos para EFWChat.....	69
Tabla 5: Instanciar objeto usuario para la conexión.....	70
Tabla 6: Definir el esquema con las estructuras de estado compartido en el contenedor.....	70
Tabla 7: Conectarse al servicio Azure Fluid Relay.....	71
Tabla 8: Instanciar EFWChat.....	71
Tabla 9: Parámetros del EFWChat.....	72
Tabla 10: Escala de Likert.....	80
Tabla 11: Perfil de los Usuarios.....	81
Tabla 12: Puntuaciones Formulario SUS.....	91
Tabla 13: System Usability Scale (SUS).....	99
Tabla 14: Tareas más importantes.....	102
Tabla 15: Estados de Ánimo Entrevista Inicial.....	104
Tabla 16: Tareas Prueba de Usabilidad.....	105
Tabla 17: Análisis de Comportamiento Esperado.....	106
Tabla 18: Estados de Ánimo Entrevista Inicial.....	107

INTRODUCCIÓN

En un entorno empresarial dinámico y altamente regulado, la redacción de contratos entre proveedores y empresas mineras emerge como una tarea crítica que exige confianza, colaboración eficiente y transparencia. La complejidad inherente a estos acuerdos, junto con la necesidad de adaptarse a las cambiantes regulaciones y condiciones del mercado, ha impulsado la búsqueda continua de herramientas innovadoras para facilitar y optimizar este proceso.

En este contexto, el presente trabajo de título se centra en el desarrollo de un sistema colaborativo diseñado para agilizar y mejorar la redacción de contratos entre proveedores e importante empresa minera que es cliente de la consultora informática Envision Ltda. El propósito principal es fortalecer las relaciones comerciales, reducir riesgos legales y fomentar la eficiencia operativa. De modo que este sistema se presenta como una respuesta estratégica a los desafíos inherentes a la gestión contractual, incorporando tecnologías modernas para potenciar la colaboración entre las partes involucradas y garantizar la coherencia y legalidad de los acuerdos establecidos.

Para lograr esto, se llevará a cabo la integración de herramientas colaborativas que permitan la creación, revisión y gestión efectiva de contratos, contribuyendo así al desarrollo sostenible y al éxito de las operaciones en el sector minero.

Al concluir el desarrollo del proyecto, los componentes de la interfaz de usuario (UI) desarrollados durante el proyecto y que contemplen funcionalidades colaborativas se implementarán en el Envision Framework (EFW). Esta integración permitirá su uso y aplicación en futuros proyectos de índole colaborativa de manera sencilla y eficaz, asegurando la continuidad y la versatilidad de las soluciones desarrolladas.

A continuación, se presenta la estructura del trabajo, organizada en capítulos diseñados para abordar a fondo el problema y describir el proceso de desarrollo.

➤ Capítulo 1: Definición del Problema

Se contextualiza el problema al proporcionar una visión general de la situación actual en cuanto a la redacción de contratos. Además, se definen de manera clara y precisa los objetivos generales y específicos que guiarán este trabajo de título.

➤ Capítulo 2: Marco Conceptual

Se abordan los conceptos, metodologías, tecnologías y enfoques fundamentales

que sustentan y proporcionan un contexto para el desarrollo del proyecto.

➤ Capítulo 3: Estado del Arte

Se presenta un análisis exhaustivo del estado del arte de diversos algoritmos distribuidos, de exclusión mutua y consenso, explorando sus posibles aplicaciones en el desarrollo del software además de explorar librerías de código que sean de ayuda en la implementación de estos algoritmos acorde a los requerimientos del sistema.

➤ Capítulo 4: Propuesta de Solución

Se detalla el proceso que condujo a la solución propuesta, incorporando propuestas de diseño y diagramas que facilitan la comprensión y uso de la herramienta desarrollada.

➤ Capítulo 5: Validación de la Solución

Corresponde a la validación de la propuesta de solución, llevando a cabo pruebas de usabilidad y entrevistas para medir la experiencia usuaria, evaluando la herramienta desde la perspectiva de quienes la utilizan, proporcionando retroalimentación valiosa sobre su efectividad y usabilidad.

➤ Capítulo 6: Conclusiones

Se presenta la conclusión del trabajo, donde se destaca el cumplimiento de los objetivos definidos al inicio del proyecto. Además, se analiza el impacto concreto que tuvo la implementación de la propuesta, proporcionando una visión global de su relevancia y eficacia en el contexto abordado.

Con este trabajo de título, no solo se abordan desafíos contractuales inmediatos, sino que también sienta las bases para prácticas comerciales más eficientes y adaptativas en el futuro apuntando a fortalecer la capacidad de negociación y gestión contractual, contribuyendo así al crecimiento sostenible de la industria minera.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Contexto

Dentro del conglomerado de clientes de Envision, está una importante empresa minera chilena que inmersa en su proceso de Transformación Digital presenta la necesidad de mejorar el proceso de redacción de contratos en la plataforma SOFI (Sistema Online de Firma de Contratos), específicamente se busca mejorar la trazabilidad y transparencia del proceso, abarcando desde las modificaciones contractuales hasta los finiquitos y las comunicación con los proveedores. Hasta la fecha, SOFI se erige como una solución integral que satisface necesidades cruciales, tales como:



Trazabilidad
de un contrato



Digitalización de
documentos



Firma
electrónica



Contrato
virtual

En SOFI, un usuario con el rol de Ingeniero de Contratos se encarga de crear un contrato ingresando diversos datos en un formulario. Con estos datos, el sistema completa una plantilla en documento Word según el tipo de contrato y la envía para revisión de los abogados del Área de Abastecimiento y/o Área Legal de la empresa minera según corresponda. De aprobarse, se utiliza Adobe Sign para que los proveedores y apoderados agreguen su firma electrónica finalizando el proceso. De rechazarse, vuelve al Ingeniero de Contratos para que realice las correcciones pertinentes y deberá ser enviado a revisión nuevamente.

A pesar de sus años de funcionamiento, en SOFI se han identificado oportunidades de mejora, especialmente en el contexto de las complejas operaciones mineras y los riesgos asociados con la modificación de contratos. Con el objetivo de mitigar estos riesgos, las mejoras se enfocan en implementar funcionalidades colaborativas que faciliten una interacción más eficiente entre las partes involucradas.

Adicionalmente, se persigue la meta de lograr una escalabilidad que se adapte a las cambiantes necesidades de la empresa contribuyendo a la transparencia en los procesos contractuales y mejorando el flujo de negociación entre todas las partes comprometidas en un determinado contrato. Dichas mejoras, corresponden a cambios específicos en la plataforma, y no serían aplicadas sobre SOFI en su totalidad.

1.2 Situación Actual

Con el objetivo de obtener una comprensión más profunda de la problemática, extendemos nuestra investigación a la industria chilena. Es importante recalcar que en Chile, los tribunales se declaran incompetentes en situaciones como las re-adjudicaciones, revocaciones, invalidaciones y actos ocurridos durante la ejecución del contrato.

Según datos proporcionados por [Tribunal de Contratación Pública, 2016] entre 2010 y 2016 (*Figura 1*), se identifican tendencias en procesos licitatorios relevantes respecto a lo contemplado en el artículo 24 de la Ley N°19.886¹, donde señalan a la Resolución de Adjudicación como el acto administrativo del proceso licitatorio más reclamado.

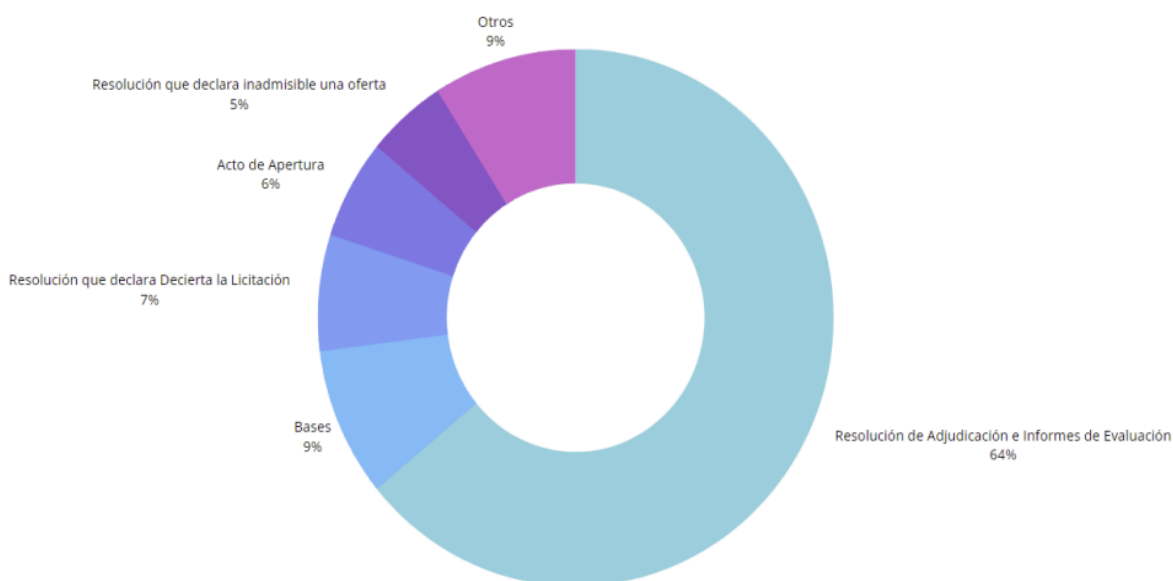


Figura 1: Actos Impugnados 2010-2016.

Fuente: Elaboración propia en base a datos del Tribunal de Contratación Pública (TCP).

Otro dato importante, es que el 7% de las licitaciones se declaran desiertas, es decir que ninguna de las empresas o individuos que participaron en la licitación cumplió con los requisitos u otros criterios establecidos en el documento de licitación.

En la *figura 2*, se destaca que el 70% de las demandas presentadas ante el TCP están orientadas a impugnar licitaciones de contratos de prestación de servicios (44%) o de bienes inmuebles (18% Bienes y 8% Fármacos).

¹ Ley 19.886. (2022, 28 de abril). Ley de Bases sobre Contratos Administrativos de Suministro y Prestación de Servicios.

Asimismo, dentro de los datos registrados entre los años 2010 y 2016, se puede obtener un listado con las entidades más demandadas (*figura 3*), siendo liderado por Municipalidades con un 38% de las demandas, seguido de Servicios Autónomos con personalidad jurídica y patrimonio propio con un 19%, en tercer y cuarto lugar se encuentran el sector de Salud con un 17% de las demandas y el Gobierno Central con un 14%. Le siguen las Fuerzas Armadas y Carabineros y el Ministerio de Obras Públicas con un porcentaje del 5% cada uno.

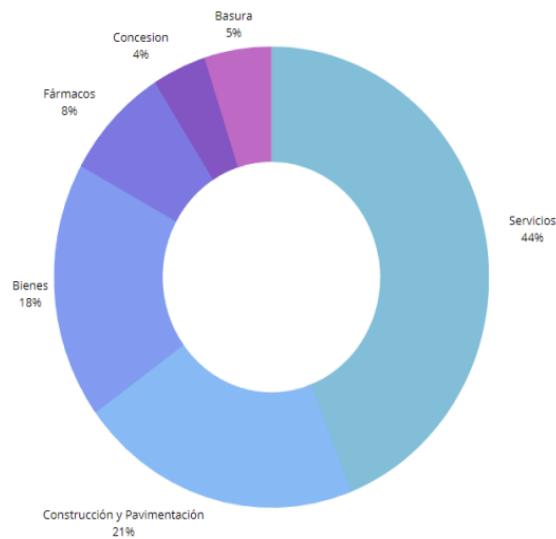


Figura 2: Tipos de Contratos Licitados 2010-2016.
Fuente: Tribunal de Contratación Pública (TCP).

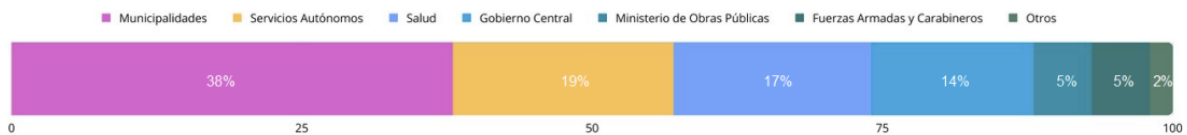


Figura 3: Porcentaje de Entidades demandadas por Sector 2010-2016.
Fuente: Elaboración propia en base a datos del Tribunal de Contratación Pública (TCP).

Con esto en mente, al explorar el mercado, existen diferentes servicios que facilitan la redacción de un contrato, llevando la trazabilidad y acelerando el proceso de firma, sin embargo, no se han realizado muchos avances significativos que permitan la redacción de forma colaborativa promoviendo la comunicación entre Ingenieros de Contratos y representantes de Proveedores.

1.3 Problema a Resolver

En el complejo entorno empresarial, las empresas independientemente del sector, confían en contratos de prestación de servicios para establecer acuerdos y obligaciones. Estos contratos, que incorporan cláusulas legales y/o comerciales, buscan salvaguardar los intereses de las partes involucradas, sin infringir la ley o principios éticos y morales donde una correcta elaboración de estos contratos no solo es esencial para cumplir con los requisitos legales, sino que también sienta las bases para relaciones comerciales duraderas basadas en la confianza mutua, ofreciendo diversos beneficios a las empresas implicadas.

Sin embargo, este escenario no está exento de desafíos, especialmente en el ámbito de las macro-empresas. Es más común de lo que se piensa encontrarse con problemas, tales como la modificación de un contrato sin el consentimiento de las partes involucradas o la firma de contratos por parte de representantes legales sin el pleno conocimiento de su contenido. Estos inconvenientes pueden desencadenar conflictos de intereses, disputas e incluso conducir al incumplimiento o resolución del contrato. En algunos casos, la falta de un acuerdo sobre cómo proceder suele derivar en demandas o acciones legales, siendo los tribunales los encargados de tomar decisiones sobre la cuestión.

Esta problemática se intensifica, especialmente en empresas de la industria minera, donde los montos involucrados en los contratos suelen ser significativos, oscilando entre cientos o miles de millones de pesos o dólares. La magnitud de estas transacciones agrega una capa adicional de complejidad y riesgo a la gestión contractual, destacando la necesidad crítica de abordar de manera efectiva los desafíos asociados a la redacción y modificación de contratos.

1.3.1 Árbol del Problema

En la *figura 4* se entrega el árbol del problema, el cual permite visualizar de manera jerárquica las causas y efectos relacionados al problema:

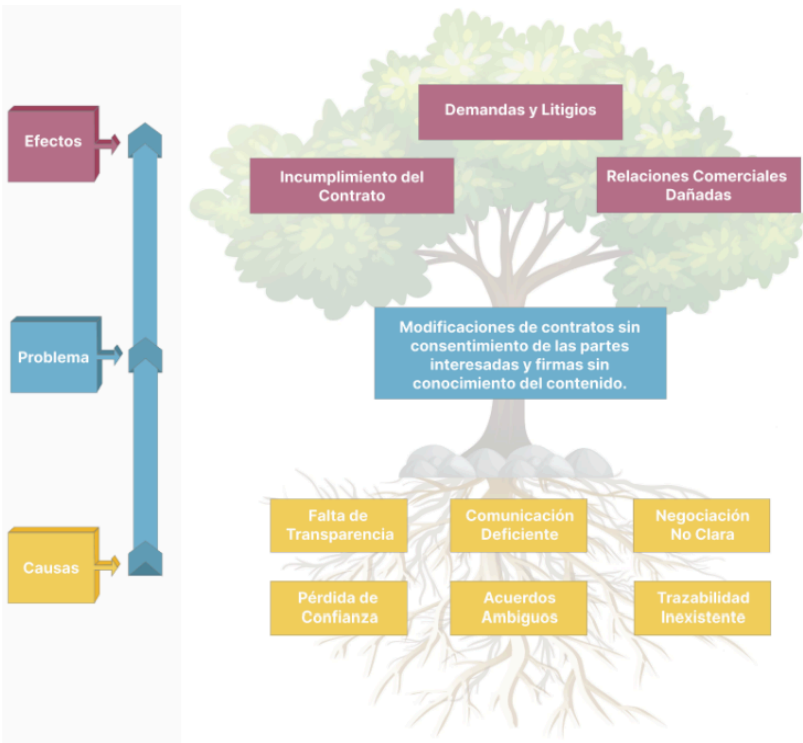


Figura 4: Árbol del Problema.
Fuente: Elaboración propia.

Causas

- Falta de transparencia:
La ausencia de procesos claros y abiertos contribuye a la opacidad en la redacción y modificación de contratos.
- Comunicación deficiente:
La falta de canales efectivos de comunicación puede dar lugar a malentendidos y a la falta de claridad en los términos contractuales.
- Negociación no clara:
Si las negociaciones no son precisas y comprensibles, resultando en la omisión de detalles legales importantes o puntos de interés sin considerar.
- Pérdida de confianza:
La falta de transparencia y comunicación puede resultar en la pérdida de confianza entre las partes involucradas.
- Acuerdos ambiguos:
La falta de claridad en la redacción de los acuerdos propicia malentendidos y posibles interpretaciones erróneas.
- Trazabilidad inexistente:
La ausencia de un seguimiento claro y documentado dificulta la identificación de cambios y decisiones en el proceso contractual.

Problema:

- Modificaciones de contratos sin consentimiento de las partes interesadas y firmas sin conocimiento del contenido:
Estos eventos son la manifestación directa de las causas subyacentes. La falta de transparencia, comunicación deficiente y negociaciones no claras contribuyen a la realización de cambios no autorizados y a la firma de contratos sin una comprensión completa de su contenido.

Efectos:

- Incumplimiento del contrato:
La falta de claridad y consenso en los contratos puede conducir a su incumplimiento.

- Demandas y litigios:
La falta de integridad contractual puede dar lugar a disputas legales y demandas entre las partes involucradas.
- Relaciones comerciales dañadas:
El conjunto de problemas mencionados afecta negativamente la confianza y colaboración, dañando las relaciones comerciales a largo plazo.

1.4 Objetivos

Luego de analizar la problemática, nacen los objetivos de este trabajo de título.

1.4.1 Objetivo General

Desarrollar un sistema colaborativo para la elaboración de contratos entre proveedores y empresa minera, que permita la comunicación e interacción en tiempo real, con el fin de generar transparencia y eficiencia al proceso de negociación inserto en todo contrato.

1.4.2 Objetivos Específicos

1. Implementar algoritmos de consenso que permitan una eficiente edición colaborativa de un contrato.
2. Construir un sistema de comunicación mediante chats para debatir el contenido de las cláusulas comerciales o legales insertas en el contrato.
3. Extender el Envision Framework con el desarrollo de componentes de interfaz de usuario que puedan ser reutilizados en otros proyectos de ámbito colaborativo.
4. Validar la experiencia de usuario del sistema colaborativo desarrollado, con énfasis en comprobar la transparencia y eficiencia que provea a la generación de contratos.

1.5 Alcance y Limitaciones

En términos de alcance, se llevará a cabo un análisis detallado de la problemática y desafíos contractuales pertinentes, destacando cómo las tecnologías seleccionadas proporcionan soluciones al problema. Adicionalmente, se abordará el desarrollo, implementación y evaluación del sistema colaborativo propuesto, desde las fases iniciales de su diseño hasta su transición al ambiente de producción. Para finalizar, se pondrá énfasis en el impacto y la eficiencia logrados al crear componentes reutilizables de fácil implementación para proyectos futuros. En relación con las limitaciones de este trabajo, por razones de confidencialidad y NDA firmado por Envision con su cliente, se abstendrá de divulgar información asociada a la empresa cliente, como nombres, datos de usuarios, logotipos o referencias específicas vinculadas al desarrollo del sistema.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Algoritmo

El término Algoritmo se refiere a un “conjunto de reglas que hay que seguir para realizar una tarea o resolver un problema” [Team, 2022]. En el campo de la informática, un programa informático es un algoritmo escrito en un lenguaje de programación y que le indica al ordenador qué pasos debe realizar y en qué orden para llevar a cabo una tarea específica.

2.1.1 Algoritmos de Consenso

Según [Binance Academy, 2018], un algoritmo de consenso es un “mecanismo que permite a los usuarios o máquinas coordinarse en un entorno distribuido. Debe garantizar que todos los agentes del sistema puedan ponerse de acuerdo respecto a una fuente única de verdad, incluso en el caso de que algunos de ellos fallen. En otras palabras, el sistema debe ser tolerante a fallas.”

2.1.2 Algoritmos de Exclusión Mutua Distribuida

[Rojas et al. 2005] describe los algoritmos de exclusión mutua distribuida, como algoritmos que proveen acceso sincronizado a recursos en un sistema distribuido.

2.2 Contrato

Según [RAE, s.f], el término contrato se refiere a un acto o convenio, oral o escrito, entre partes que se obligan sobre materia o cosa determinada, y a cuyo cumplimiento pueden ser compelidas².

2.2.1 Cláusula

Una cláusula en un contrato es una disposición o término específico que establece derechos, obligaciones, restricciones o condiciones acordadas entre las partes que firman el contrato.

Las cláusulas son esenciales para definir y regular diversos aspectos del acuerdo contractual, asegurando que ambas partes comprendan claramente sus responsabilidades y derechos.

Existen diversos tipos de cláusulas, pueden ser comerciales, legales, de confidencialidad, etc.

² Compeler: Obligar a alguien, con fuerza o por autoridad, a que haga lo que no quiere.

2.3 BPMN

Un diagrama de procesos, también conocido como BPMN (*Business Process Model and Notation*), es una representación gráfica de los pasos o actividades que componen un proceso empresarial. BPMN es un estándar ampliamente utilizado para modelar procesos de negocios de manera clara y comprensible.

Para comprender un diagrama BPMN es necesario conocer algunos conceptos clave:

- Proceso: Un conjunto de actividades o tareas interrelacionadas que se llevan a cabo para lograr un objetivo específico.
- Actividad: Una tarea o conjunto de tareas que se realizan dentro de un proceso.
- Flujo: La secuencia de actividades y eventos que definen el orden en que se deben realizar las tareas.
- Evento: Un suceso que inicia, interrumpe o finaliza una actividad o un proceso.
- *Gateway* o Portal: Representa puntos de decisión en un proceso, donde se elige entre diferentes caminos basados en condiciones específicas.

2.4 Experiencia de Usuario (UX) e Interfaz de Usuario (UI)

La Experiencia de Usuario (UX), se refiere al área interdisciplinaria que estudia el conjunto de factores que repercuten en la interacción con un servicio o producto por parte de usuarias y usuarios [Hassan-Montero, 2015].

Según la norma ISO 9241-2010 sobre diseño centrado en las personas indica que “la experiencia de usuario son las percepciones y respuestas de una persona como resultado del uso o de la anticipación del uso de un producto, sistema o servicio. [...] La experiencia de usuario incluye todas las emociones, creencias, preferencias, percepciones, respuestas físicas y fisiológicas, comportamientos y logros de los usuarios que ocurren antes, durante y después del uso [Organización Internacional de Normalización, 2019].

2.4.1 Pruebas de Usabilidad

Una prueba de usabilidad es una técnica para evaluar la facilidad de uso de un diseño, producto, servicio o sistema con un grupo de usuarios representativos. Por lo general, implica observar a los usuarios mientras se les solicita completar tareas (*figura 5*).

Permite identificar problemas de usabilidad como dificultades para encontrar información, botones o acciones mal etiquetadas, problemas de diseño visual o de navegación, que pueden dificultar la interacción de los usuarios con el producto.



Figura 5: Proceso de una Prueba de Usabilidad.
Fuente: Elaboración propia.

Para extraer el máximo valor de la prueba de usabilidad, es esencial contar con un grupo de usuarios que se alineen con el perfil adecuado, es decir, aquellos individuos o profesionales que representen fielmente a la audiencia prevista para la aplicación. La selección cuidadosa de participantes garantiza que los escenarios de prueba sean relevantes y que los problemas de usabilidad identificados sean verdaderamente representativos de las experiencias del usuario. Para la mayoría de las pruebas de usabilidad, 5 o 6 usuarios son suficiente para descubrir aproximadamente del 80% a 85% de los problemas de usabilidad en un producto.

Es esencial que el facilitador guíe la prueba de usabilidad de manera efectiva, destacando que la evaluación se enfoca en el sistema y no en el usuario. Para establecer un entorno claro y colaborativo, se deben comunicar previamente las condiciones de la prueba, como la posibilidad de grabar la sesión, y obtener el consentimiento del participante.

También, se recomienda realizar una prueba piloto antes de la implementación completa, ya que permite evaluar la efectividad del proceso de usabilidad, verificar la claridad de las tareas, establecer un punto de retorno ante posibles fallos del sistema y gestionar contratiempos de manera eficiente. Sin embargo, es importante abordar con precaución la presencia de errores, ya que estos pueden generar frustración en los usuarios.

2.4.2 Entrevista

Las entrevistas en pruebas de usabilidad son una parte esencial del proceso de evaluación de la experiencia del usuario (UX). Estas entrevistas proporcionan información valiosa al permitir a los investigadores comprender más profundamente las experiencias, percepciones y pensamientos de los usuarios mientras interactúan con un producto o sistema.

Podemos indagar en el perfil del usuario, su experiencia, qué tan familiarizado se encuentra con el producto o aplicación que se va a probar.

2.4.3 *System Usability Scale (SUS)*

En el contexto de Experiencia del Usuario, "SUS" se refiere a "*System Usability Scale*" (Escala de Usabilidad del Sistema), una herramienta estándar utilizada para evaluar la usabilidad de un sistema, producto o servicio.

La Escala de Usabilidad del Sistema consta de 10 preguntas que los usuarios responden después de interactuar con el sistema en cuestión. Estas preguntas se centran en evaluar la facilidad de uso, la eficiencia y la satisfacción general del usuario. Las respuestas se califican en una escala de cinco puntos, desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo". La puntuación total se calcula y se ajusta para proporcionar una medida cuantitativa de la usabilidad del sistema.

El SUS es valioso porque proporciona una evaluación rápida y general de la usabilidad, lo que permite a los diseñadores y desarrolladores obtener información sobre la experiencia del usuario de manera eficiente.

2.5 Groupware

El término "*Groupware*", hace referencia a tecnología de software y hardware para asistir la interacción de grupos en un tarea o proyecto en común para alcanzar sus objetivos. Facilita la comunicación, colaboración y coordinación entre los miembros de un grupo, incluso si están geográficamente dispersos. Los sistemas *groupware* suelen incluir una variedad de herramientas y funciones enfocadas en facilitar el trabajo en equipo, algunos ejemplos de ellas son: sistemas de mensajería y videoconferencias, editar documentos de forma colaborativa, calendarios compartidos o bien listas de tareas.

2.6 Rol

Conjunto de funciones, responsabilidades y comportamientos asociados con una posición específica en un grupo, organización o sistema.

2.7 Tecnologías

2.7.1 Envision Framework

Biblioteca de librerías de Javascript basada en framework7 desarrollada por Envision para desarrollar proyectos sobre SharePoint Online.

2.7.2 Microsoft 365

Microsoft 365 es una línea de servicios por suscripción ofrecidos por Microsoft. Entre los que se encuentran programas populares como Word, Excel, Powerpoint, Outlook, Teams, Exchange, Publisher, Access y SharePoint. Siendo este último de nuestro interés.

2.7.3 Microsoft Power Automate

Servicio de Microsoft para el desarrollo de flujos de trabajo y automatización de procesos en la nube.

2.7.4 Microsoft SharePoint

Producto y servicio de Microsoft 365, que permite almacenar información y documentos, provee de portales e *intranets* para grupos de usuarios con diversas utilidades.

2.7.5 Microsoft Azure

Es una plataforma de computación en la nube creada por Microsoft para construir, probar, desplegar y administrar aplicaciones y servicios mediante el uso de sus centros de datos.

2.7.6 Azure Fluid Relay

Servicio de Microsoft Azure, provee un conjunto de funciones de servidor de Fluid integradas para facilitar el aprovisionamiento y la administración de las experiencias de colaboración de Fluid.

2.7.7 Fluid Framework

Fluid Framework, es una colección de bibliotecas cliente para distribuir y sincronizar estados compartidos [Fluid Framework Documentation, s.f]. Estas bibliotecas permiten que varios clientes creen y operen simultáneamente en estructuras de datos compartidas utilizando patrones de codificación similares a los utilizados para trabajar con datos locales.

CAPÍTULO 3: ESTADO DEL ARTE

En el contexto de este proyecto, fue necesario adentrarse en la vanguardia de los algoritmos distribuidos, de consenso y exclusión mutua, con el objetivo de encontrar soluciones que optimicen la edición colaborativa de documentos contractuales y mejoren la eficacia de nuestro sistema de mensajería.

Se parte con la hipótesis de que cada usuario conectado al entorno colaborativo debe comportarse como un nodo lógico en un sistema distribuido. En este escenario, hubo que enfrentar desafíos como mantener la coherencia y trazabilidad del contrato cuando múltiples usuarios realizan ediciones simultáneas. O bien, si se restringe la edición a un usuario a la vez, se deben abordar cuestiones relacionadas con la gestión de conflictos y control de cambios.

Adicionalmente, es fundamental que estas soluciones sean de implementación sencilla, considerando las restricciones temporales inherentes a la planificación del proyecto.

A continuación se detallarán algunos de los algoritmos concurrentes más conocidos.

3.1 Algoritmos Concurrentes

Los algoritmos concurrentes representan conjuntos de instrucciones o procedimientos diseñados para enfrentar la concurrencia, un escenario en el cual múltiples tareas o procesos comparten recursos y operan de manera independiente.

Dados los requerimientos del proyecto, es necesario abordar algoritmos que sean aplicables en el desarrollo de un sistema colaborativo. En dicho sistema, los usuarios pueden redactar un contrato de forma paralela, lo que implica la necesidad de gestionar simultáneamente múltiples contribuciones para garantizar una ejecución fluida y sin interferencias en la redacción del contrato.

3.1.1 Reloj Lógico de Lamport

El reloj lógico o *logical clock* (LC) [Lamport, 1978] es una manera de ordenar eventos en sistemas distribuidos. El cual es muy útil en aplicaciones distribuidas que solo requieren saber si un evento ha sucedido antes que otro.

Para comprenderlo hay que tener en cuenta los distintos tipos de relaciones y propiedades:

Dado dos eventos (a, b) , si a y b son eventos de un mismo nodo y a ocurre antes que b , su relación se denota de la siguiente manera:

$$a \rightarrow b$$

Existiendo transitividad entre dos eventos:

$$\text{Si } a \rightarrow b \text{ y } b \rightarrow c, \text{ entonces } a \rightarrow c$$

Dos eventos x e y son concurrentes si NO se puede decir cual de ellos ocurre antes, simbolizado como:

$$x \parallel y$$

La figura 6 es un claro ejemplo de cómo se denota la relación entre los eventos $\{a, b, c, d, e, f, g, h\}$ para los distintos nodos $\{P_1, P_2, P_3\}$.

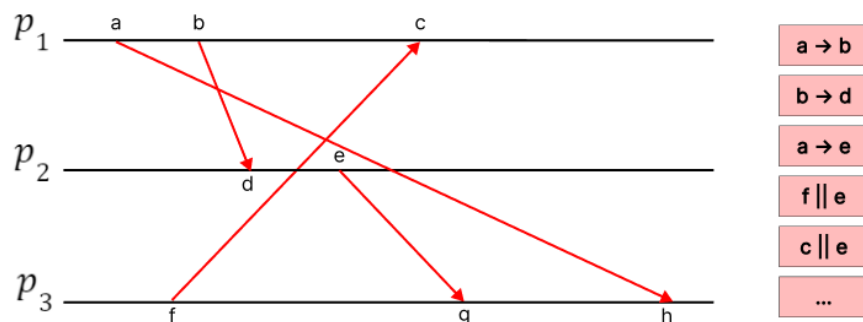


Figura 6: Ejemplo de relaciones entre distintos eventos.
Fuente: Elaboración propia.

El algoritmo consiste en:

1. Cada nodo p dispone de un contador o reloj lógico inicializado en 0.

$$C_p = 0$$

2. Por cada ejecución de un evento o envío de mensaje en un nodo p , se incrementa el valor de su reloj lógico en 1. Es decir:

$$C_p += 1$$

3. Cuando un nodo p recibe un mensaje, entonces actualiza su reloj de la siguiente manera:

$$C_p = \max(C_p, C_m) + 1$$

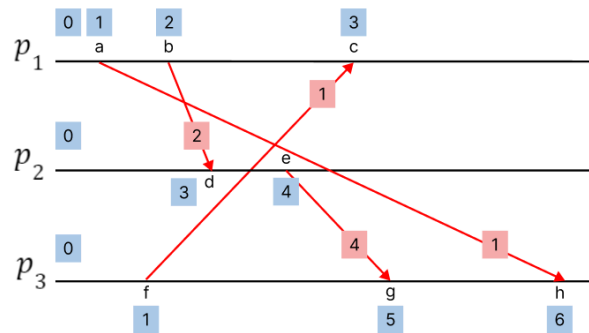


Figura 7: Valor del reloj lógico para cada evento.
Fuente: Elaboración propia.

3.1.2 Vector Lógico de Lamport

Los relojes vectoriales corresponden a una técnica utilizada para ordenar eventos en un sistema distribuido. Fueron propuestos por Leslie Lamport en 1978 como una extensión a los relojes lógicos. En un sistema distribuido, donde múltiples eventos trabajan de forma independiente y pueden comunicarse entre sí, es fundamental poder establecer un orden temporal de los eventos para comprender la causalidad y consistencia de las operaciones.

Es por ello, que cada proceso mantiene un vector de relojes que se actualiza e incrementa cada vez que se ejecuta un evento local, como por ejemplo, enviar un mensaje.

Dados N nodos, cada nodo p mantiene un vector de N marcas temporales V_p , de forma que:

- $V_p[q]$ es el número de eventos que han ocurrido en p
- Si $V_p[q] = k$, entonces p sabe que al menos han ocurrido k eventos en el nodo q

El algoritmo consiste en:

1. Cada nodo dispone de un reloj vectorial V_p iniciado con todos sus elementos en 0.

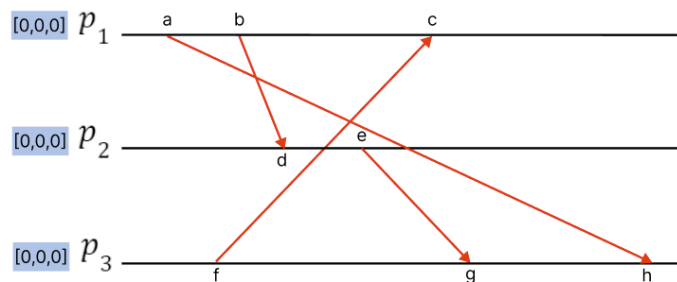


Figura 8: Relojes Vectoriales de Lamport iniciados con todos sus elementos en 0.
Fuente: Elaboración propia.

2. Un nodo p incrementa en 1 su valor $V_p[p]$ cada vez que envía un mensaje, considerando además:

$$\forall i, 1 \leq i \leq N, V_p[i] = \max(V_p[i], V_m[i])$$

Aplicando el algoritmo hasta el final, se obtiene el siguiente resultado (ver *Figura 9*):

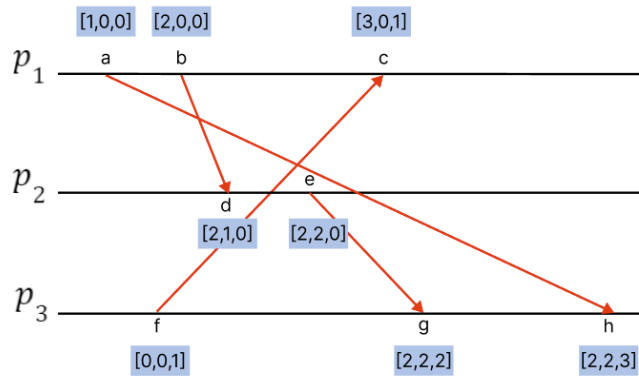


Figura 9: Relojes Vectoriales de Lamport.

Fuente: Elaboración propia.

3.2 Algoritmos de Consenso y Exclusión Mutua

Al hablar de algoritmos de consenso, se refiere al conjunto de reglas o procedimientos que permiten a un grupo de participantes llegar a un acuerdo o consenso sobre un valor o decisión, incluso cuando alguno de los participantes puede fallar o comportarse de manera maliciosa.

Por otra parte, cuando se plantean algoritmos de exclusión mutua se está hablando de procedimientos diseñados para asegurar que dos o más procesos no accedan simultáneamente a una sección crítica en programación concurrente. La sección crítica representa la porción de código que manipula o accede a un recurso compartido, y los algoritmos de exclusión mutua garantizan que esta sección sea ejecutada de manera única y sin interferencias, evitando conflictos y garantizando la consistencia de los datos compartidos entre los procesos.

3.2.1 Algoritmo de Berkeley

El algoritmo de Berkeley [Gusella & Zatti., 1987], es un algoritmo que permite sincronizar relojes, está especialmente diseñado para coordinar el tiempo entre diferentes computadoras en un sistema distribuido.

Dado un grupo de nodos formado por:

- Un servidor S
- N clientes, donde cada cliente es C_i

Cada reloj dispone de un reloj lógico de manera local. A su vez, cada nodo tiene un tiempo real Tr_i . La finalidad del algoritmo es que, dada la iniciativa del servidor, todos los nodos clientes sincronizan sus relojes lógicos. El algoritmo consiste en los siguientes pasos:

1. El Servidor S difunde su reloj en T_0 , para simplificar, se asumirá que los mensajes llegarán a los clientes en 2 unidades de tiempo [UT]³ fijas y constantes, este es un supuesto fuerte, lejano a la realidad, sin embargo podría facilitar el entendimiento.

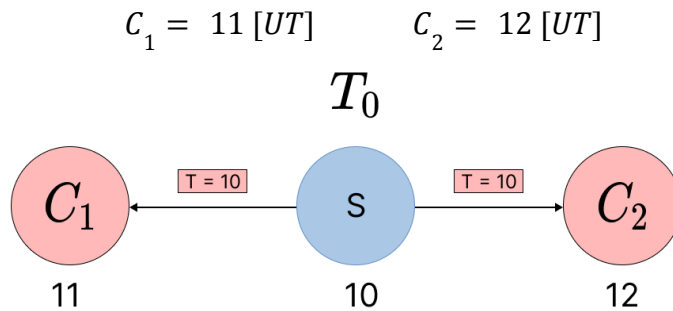


Figura 10: Servidor inicia algoritmo difundiendo su reloj $T_0 = 10$ [UT].

Fuente: Elaboración propia.

2. Cada uno de los clientes calcula la diferencia D_i entre su reloj local y el tiempo que ha sido notificado por el servidor en el mensaje. Luego notifica dicha diferencia.

$$D_1 = 13 - 10 = 3 \quad D_2 = 14 - 10 = 4$$

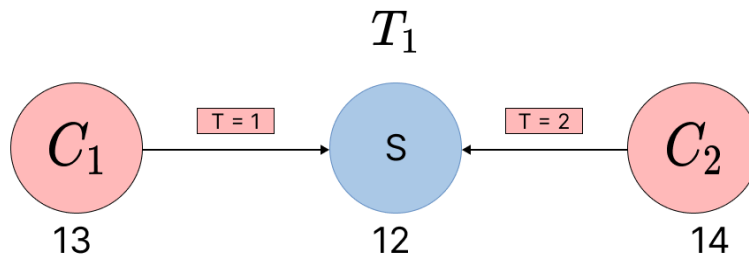


Figura 11: Nodos clientes notifican las diferencias en el reloj D_i al Servidor S .

Fuente: Elaboración propia.

³ UT: Unidad de Tiempo

3. Se ajusta la diferencia notificada tal que:

$$D_i' = D_i - \frac{T_{1i} - T_0}{2} \quad (1)$$

$$D_1' = 3 - \frac{14-10}{2} = 1 \quad D_2' = 4 - \frac{14-10}{2} = 2$$

Se obtiene el promedio de la diferencia:

$$D = \frac{1+2+0}{2} = \frac{3}{2}$$

4. Se notifica el ajuste a cada nodo cliente:

$$A_i = D - D_i' \quad (2)$$

Se obtiene:

$$A_1 = \frac{3}{2} - 1 = \frac{1}{2} \quad A_2 = \frac{3}{2} - 2 = -\frac{1}{2}$$

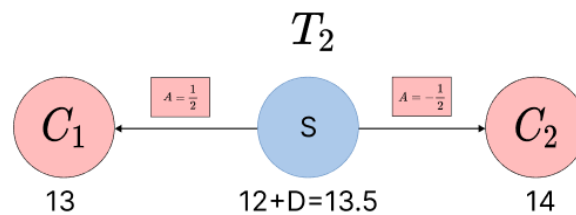


Figura 12: Servidor S notifica los ajustes.

Fuente: Elaboración propia.

5. Cada cliente ajusta su reloj incrementando en A_i

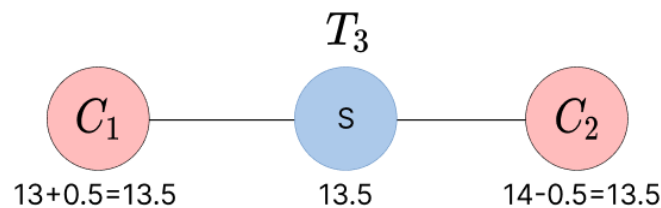


Figura 13: Clientes ajustan sus relojes.

Fuente: Elaboración propia.

3.2.2 Algoritmo del Semáforo de Dijkstra

[Dijkstra, 1968] propuso un algoritmo en 1968 que da solución al problema de la exclusión mutua para el acceso a recursos compartidos, evitando condiciones de carrera⁴ en programación concurrente mediante la implementación de semáforos.

Dijkstra abarca el problema del Productor-Consumidor, donde considera 2 procesos:

- Productor: Proceso cíclico que en cada ciclo produce una porción de información que ha de ser procesada por el “Consumidor”.
- Consumidor: Proceso cíclico que en cada ciclo procesa la siguiente porción de información producida por el “Productor”.

La relación entre estos dos procesos, Productor y Consumidor, implica una manera o canal de comunicación entre ellos, a través de los cuales pueden transmitir las porciones de información.

Algunos supuestos son:

- Ambos procesos están conectados a través de un buffer de capacidad ilimitada.
- Las porciones de información producidas no necesitan ser consumidas inmediatamente.

Notar que se utiliza un supuesto fuerte: un buffer con capacidad ilimitada lo cual es irrealista, sin embargo no debiera tener un mayor impacto para comprender el algoritmo. Con esto en mente, el productor sólo puede ofrecer su siguiente porción de información cuando la porción anterior ya ha sido consumida.

Explicando el algoritmo, un productor agrega elementos a un buffer y un consumidor los retira. La exclusión mutua se implementa mediante semáforos para garantizar que solo un proceso pueda modificar el buffer a la vez. Los semáforos son los siguientes:

- **Número de porciones en espera:** Controla cuántas porciones están disponibles para ser consumidas.
- **Número de posiciones vacías:** Rastrea cuántas posiciones vacías quedan en el buffer para que el productor pueda agregar porciones.
- **Manipulación del buffer:** Un semáforo que asegura la exclusión mutua, permitiendo que solo un proceso a la vez modifique el buffer.

El productor produce porciones y las agrega al buffer. Antes de agregar, adquiere el semáforo **Número de posiciones vacías** y **Manipulación del buffer**. Luego, libera el semáforo **Manipulación del buffer** y aumenta el **Número de porciones en espera**.

⁴ Condiciones de Carrera: Situación en la que el resultado de un programa depende del orden de ejecución de las instrucciones.

El consumidor retira porciones del buffer. Antes de retirar, adquiere el semáforo **Número de porciones en espera** y **Manipulación del buffer**. Luego, libera el semáforo **Manipulación del buffer** y aumenta el **Número de posiciones vacías**.

Esta estructura garantiza que el productor y el consumidor no accedan al buffer simultáneamente, evitando condiciones de carrera y asegurando la integridad de los datos compartidos.

El pseudocódigo del algoritmo corresponde al descrito en la [figura 14](#):

Algorithm 1 Manipulación de buffer con exclusión mutua

```
1. Número de porciones en espera ← 0;
2. Número de posiciones vacías ← N;
3. Manipulación del buffer ← 1;
4.
5. parbegin
6.     productor:
7.         repeat
8.             Producir próxima porción;
9.             Wait(Número de posiciones vacías);
10.            Wait(Manipulación del buffer);
11.            Agregar porción al buffer;
12.            Signal(Manipulación del buffer);
13.            Signal(Número de porciones en espera);
14.        until false
15.    until false
16.    consumidor:
17.        repeat
18.            Wait(Número de porciones en espera);
19.            Wait(Manipulación del buffer);
20.            Tomar porción del buffer;
21.            Signal(Manipulación del buffer);
22.            Signal(Número de posiciones vacías);
23.            Procesar porción tomada;
24.        until false
25.    parend
```

Figura 14: Algoritmo del Semáforo de Dijkstra.

Fuente: Elaboración propia en base al algoritmo del artículo "*Co-operating sequential processes*".

3.2.3 Spinlocks con Test And Set

Los *spinlocks* o cerraduras de giro son un mecanismo de exclusión mutua utilizado en programación concurrente. A diferencia de otros mecanismos que hacen que un hilo que no puede adquirir la exclusión mutua espere de manera pasiva, un *spinlock* hace que el hilo espere activamente repitiendo un bucle ("girando") hasta que adquiere la cerradura. Estos bucles de espera activa son conocidos como "spinning".

En la literatura, no hay un autor a quien se le atribuya el concepto de *spinlock*, sin embargo diversos libros asociados a sistemas operativos, algoritmos de concurrencia, o teoría de la programación concurrente discuten su implementación. Ejemplo de ello es el artículo presentado por [Anderson, 1990], que utilizó *spinlocks* y *Test and Set* para garantizar la consistencia de las operaciones de acceso a memorias compartidas por multiprocesos.

En el contexto de *spinlocks*, un hilo o proceso que desea adquirir el *spinlock* utiliza *Test and Set* para verificar si el *spinlock* está bloqueado y, al mismo tiempo, lo bloquea para evitar que otros hilos o procesos lo adquieran. Este proceso se repite activamente hasta que el hilo o proceso logra adquirir el *spinlock*, evitando la espera pasiva y girando en un bucle hasta que el *spinlock* esté disponible.

Una descripción básica de su funcionamiento es:

1. Iniciar el *lock* como desbloqueado.
2. Adquirir(*lock*):

Cuando un hilo o proceso quiere acceder a la sección crítica, debe adquirir el *spinlock*:

- Si el *spinlock* está desbloqueado, entonces lo adquiere.
- Si el *spinlock* está bloqueado, repite el bucle hasta que el *spinlock* esté desbloqueado.

3. Liberar(*lock*):

Cuando el proceso o hilo ha terminado de ejecutar la sección crítica, libera el *spinlock*, cambiando su estado a desbloqueado. Una vez el *spinlock* ha sido liberado, otros hilos o procesos pueden adquirirlo.

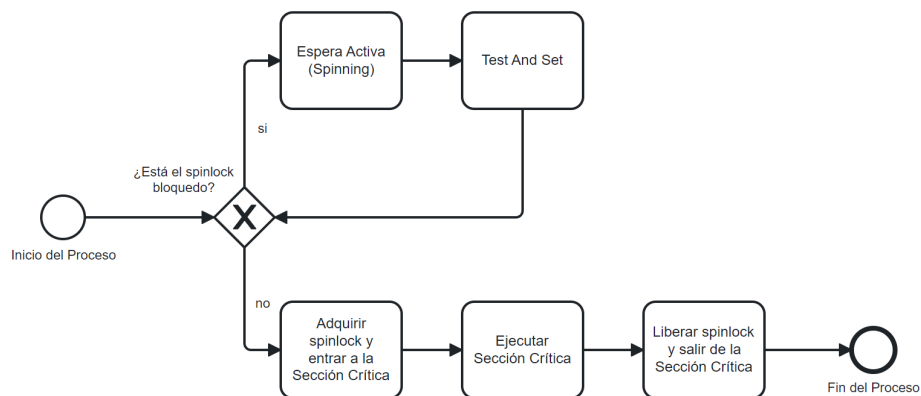


Figura 15: Diagrama BPMN Algoritmo *Spinlock* con *Test and Set*.

Fuente: Elaboración propia.

Es importante señalar que este algoritmo requiere un importante uso de recursos, que los procesos están constantemente esperando y revisando que el *spinlock* se desbloquee.

3.2.4 Algoritmo de Peterson

Presentado en el artículo "*Myths About the Mutual Exclusion Problem*" [Peterson, 1981] el algoritmo de Peterson utiliza dos variables booleanas compartidas para cada proceso:

- *flag*: Indica si el proceso se encuentra listo para entrar a la sección crítica.
- *turn*: Indica cuál de los procesos tiene el derecho a entrar en la sección crítica.

Algorithm 2 Algoritmo de Peterson con Variables Flag y Turno

```
1. boolean flag[2] = {false,false}
2. int turno = 0;
3. Process P_i:
4. repeat
5.     flag[i]=true;
6.     turno = j
7.     while (flag[j] && turno == j) {
8.         // do nothing, only waiting
9.     }
10.    enterCriticalSection()
11.    flag[i] = false;
12.    releaseCriticalSection()
13. forever
```

Figura 16: Algoritmo de Peterson.
Fuente: Elaboración propia.

El algoritmo de Peterson es relativamente simple y fácil de entender (*figura 16*), y puede ser implementado en diversos entornos, sin embargo está específicamente diseñado para trabajar con dos procesos. Si se quisiera extender se necesitaría implementar modificaciones.

Adicionalmente, cuando un proceso no puede entrar a la sección crítica entra en espera activa⁵, lo que puede consumir recursos de manera ineficiente o bien afectar a su rendimiento.

3.3 Sistema de Mensajería y Colaboración

Esta investigación en sistemas de mensajería y colaboración se enfoca en comprender las funciones cruciales y la evolución de estas plataformas en el contexto tecnológico actual.

Se exploraron características esenciales, innovaciones recientes y tendencias emergentes para mejorar la eficacia, seguridad y versatilidad de estos sistemas.

⁵ Espera activa: Revisa periódicamente mientras espera su turno.

3.3.1 Mensajería Instantánea en Redes Sociales y Chatbots

En la actualidad, existen muchas aplicaciones de mensajería instantánea que son bastante populares entre la comunidad. Algunas de ellas son WhatsApp, Telegram, Facebook, Instagram, X (antes Twitter) y Signal.

[WhatsApp, s.f], especialmente, destaca en su sitio web al poner un fuerte énfasis en la privacidad de los mensajes. Además de realizar copias de seguridad, la plataforma ofrece una funcionalidad interesante, enviar mensajes temporales, que desaparecen después de haberlos enviado.

Por el lado de los chatbots, se suelen utilizar para automatizar servicios de atención al cliente, respondiendo preguntas frecuentes o realizando actividades específicas sin la intervención humana. Todo esto se ha logrado gracias a los avances en inteligencia artificial (IA) y machine learning (ML), lo que ha llevado a mejorar la calidad de las respuestas y experiencia del usuario.

Existen variadas herramientas y servicios que ofrecen la creación de *chats bots online*, sin embargo la implementación de uno, va más allá de lo requerido en el proyecto que se busca abordar en este trabajo de título.

3.3.2 Colaboración Empresarial

En el ámbito empresarial, plataformas como Slack y Microsoft Teams desempeñan un papel fundamental al facilitar la colaboración y la comunicación efectiva. Microsoft Teams, diseñada específicamente para el trabajo híbrido, contiene diversas funcionalidades en un único espacio, como chats, grupos, calendario y aplicaciones. Esto crea un entorno integrado para el uso sencillo y fiable de las aplicaciones y servicios de Microsoft 365.

Entre las características destacadas de los chats se encuentran la capacidad de agregar compañeros de equipo, reacciones, emojis, GIFs, memes, archivos y menciones. Además, permite establecer estados para que los demás usuarios conozcan tu disponibilidad y actividades actuales.

En el ámbito de las herramientas colaborativas, tanto Microsoft como Google han sido pioneros con herramientas como Google Docs y Microsoft Word Online. Más allá de la edición de documentos, estas plataformas ofrecen diversas funcionalidades prácticas para el trabajo conjunto.

Además estas aplicaciones muestran avatares (*figura 17*), generalmente ubicados en la parte superior de la pantalla, indican qué usuarios están conectados en el entorno colaborativo, a menudo representados por su imagen de perfil, nombres de animales o bien con las iniciales de la cuenta.

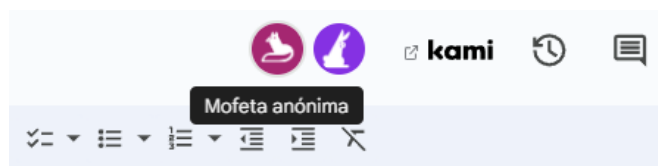


Figura 17: Avatares de Google Docs.
Fuente: Screenshot sobre Google Docs.

3.3.3 API's de Mensajería

Las API's de mensajerías son un servicio que permite a desarrolladores integrar fácilmente funciones de mensajería en aplicaciones o servicios personalizados.

[Telesign, s.f], es una API de SMS que más allá de las funcionalidades típicas de todos los sistemas de mensajería, ofrece la detección de bots y usuarios falsos, fraude de identidad sintética y fraude de apropiación de cuentas.

[Twilio, s.f] es otra API de Mensajería, ofrece tecnología CustomerAI para combinar datos de interacción en tiempo real con IA y conocer mejor a los clientes.

Otras características que se encuentran en diversas API incluyen la capacidad de realizar análisis de sentimientos en los mensajes, lo que permite evaluar la tonalidad emocional de las conversaciones.

Además, algunas plataformas ofrecen la posibilidad de crear encuestas y formularios directamente en la interfaz de chat, facilitando la recopilación de datos de manera interactiva. La integración con servicios de traducción automática también es una funcionalidad valiosa para facilitar la comunicación entre usuarios que hablan diferentes idiomas.

3.4 Conflicto de Versiones y Colisiones en la escritura

La gestión efectiva de versiones y mitigación de colisiones en la escritura son aspectos críticos en entornos colaborativos y sistemas de control de versiones.

A medida que varios usuarios contribuyen simultáneamente en un mismo conjunto de datos o documentos surge la necesidad de abordar eficazmente los conflictos de versiones y colisiones en la escritura.

En este contexto, se investigan algunos mecanismos para lidiar con estas problemáticas y algunas librerías de código que permitan detectar diferencias dada una colisión y obtener algún indicio u orientación sobre cómo resolverlo.

3.4.1 Last Write Wins (LWW)

Last Write Wins [Martin, S., Ahmed-Nacer, M., & Urso, P, 2011] es un mecanismo donde cada elemento está asociado a una marca de tiempo o *timestamp*. Dicho elemento solo puede ser modificado si la marca de tiempo de la operación de actualización es mayor a la marca de tiempo asociada al elemento.

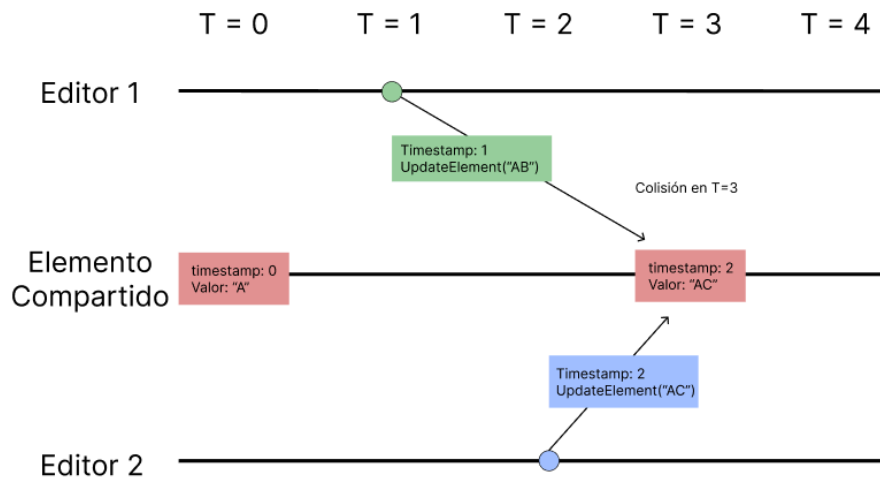


Figura 18: Ejemplo de LWW.

Fuente: Elaboración propia.

En la *figura 18*, asumiendo un retardo de mensajes, se observa que en el momento $T = 3$ se producen dos modificaciones sobre el elemento compartido.

Por un lado, el Editor 1 actualiza el valor del elemento compartido a **"AB"**, mientras que, por otro lado, el Editor 2 actualiza su valor a **"AC"**. Dado que ambas actualizaciones colisionan en el tiempo, el elemento compartido adopta el valor propuesto por el Editor 2, es decir, **"AC"**, debido a que posee el *timestamp* más reciente.

LWW destaca por su simplicidad tanto en comprensión como en implementación, sin embargo no resuelve los conflictos, simplemente adopta la última escritura sin evaluar opciones alternativas como combinar las escrituras. Esta falta de resolución puede conducir a pérdida de datos, como en el caso de ejemplo donde se descartó el valor propuesto por el Editor 1, **"AB"**.

3.4.2 Librerías de Combinación de Texto

Resolviendo el problema de conflictos existente en LWW, se procede a investigar diferentes librerías de código para detectar diferencias entre dos textos con sintaxis HTML5.

Diff-match-path [NPM, s.f] es una librería desarrollada por Google que ofrece algoritmos eficientes para realizar diferencias y parches luego de comparar dos textos. Clasifica las diferencias en tres categorías:

-1: Eliminación:

Un *substring*⁶ se encuentra en el primer texto, pero no en el segundo, es decir, fue eliminado.

0: Igualdad:

El *substring* se encuentra en ambos textos.

1: Inserción:

El *substring* se encuentra en el primer texto, pero no en el segundo, es decir, fue agregado.

Diff-Dom [NPM, s.f] es un algoritmo de diferenciación implementado en Javascript que permite obtener las modificaciones, ya sea inserción o eliminación entre dos nodos del DOM HTML.

3.5 Análisis del Estado del Arte

La aplicación de los algoritmos propuestos por Leslie Lamport en el contexto de la edición colaborativa de contratos, ofrece valiosas perspectivas para gestionar cambios concurrentes de manera eficiente. Aunque en el sistema desarrollado no se emplean nodos físicos en el sentido tradicional, se puede conceptualizar a cada usuario conectado como un "nodo lógico" en el entorno colaborativo. En este modelo, los eventos se traducen en acciones que implican la notificación mediante transmisiones (*broadcast*⁷) de cambios específicos realizados en una sección particular del contrato.

Una idea de adaptación de estos algoritmos a nuestro sistema es abordar la concurrencia en las ediciones del contrato de una manera que refleje la noción de eventos concurrentes en sistemas distribuidos. Cada usuario, como un "nodo lógico", mantiene su propio vector de relojes que se actualiza en función de las acciones realizadas. Al notificar cambios a través de transmisiones, se establece un marco que facilita la comprensión del orden temporal de las ediciones y ayuda a prevenir conflictos, similar a un control de versiones.

A su vez, los algoritmos de exclusión mutua dan una buena base si se desea restringir la edición de una sección del contrato a un usuario en específico. Asegurando que no hayan conflictos de versiones o múltiples usuarios sobrescribiendo sus cambios.

Desde la perspectiva de los sistemas de mensajería, se pueden destacar diversas

⁶ Substring: del inglés subcadena, en este contexto se refiere a una frase, oración o palabras perteneciente a un grupo de texto más grande.

⁷ Broadcast: Acto de enviar un mensaje, señal o información a múltiples destinatarios o receptores simultáneamente.

funcionalidades y características esenciales que contribuyen a una experiencia de usuario en un chat. Entre estas, se incluyen la capacidad de enviar mensajes de texto, multimedia y documentos, así como la opción de realizar llamadas de voz y video. Es fundamental que la interfaz de usuario sea intuitiva y personalizable, que la sincronización de mensajes sea eficiente en múltiples dispositivos y la seguridad robusta para proteger la privacidad del usuario. Asimismo, la integración con emojis, stickers y la capacidad de realizar búsquedas rápidas en el historial de conversaciones añaden un valor adicional.

3.5.1 Análisis Comparativo

A continuación, se procederá a comparar los algoritmos investigados en el estado del arte, evaluándolos según los siguientes criterios:

Dificultad: Este criterio mide la complejidad del algoritmo y su capacidad de comprensión. Se busca determinar qué tan accesible es para los desarrolladores comprender la lógica subyacente de cada algoritmo.

Utilidad: La utilidad del algoritmo se evalúa en función de su capacidad para cumplir con los requisitos específicos del proyecto. Se busca determinar en qué medida cada algoritmo satisface las necesidades y objetivos establecidos en la propuesta de diseño.

Escalabilidad: Este criterio mide la capacidad del algoritmo para manejar contratos con contenido extenso. Se busca determinar cómo cada algoritmo escala en términos de rendimiento a medida que aumenta la complejidad y el volumen de datos.

Mantenibilidad: Se evalúa la complejidad y dificultad asociadas con el mantenimiento del algoritmo en un proyecto. Considerando qué tan fácil o difícil puede ser para un nuevo desarrollador comprender su funcionamiento y abordar posibles problemas o errores.

Factibilidad: Este criterio mide la factibilidad de implementar cada algoritmo en el marco temporal establecido durante la planificación del proyecto. Se busca determinar cuánto tiempo y recursos serán necesarios para incorporar y poner en funcionamiento cada algoritmo.

Esta comparación proporcionará una base sólida para la selección del algoritmo más adecuado, considerando tanto las demandas específicas del proyecto como los recursos disponibles durante su desarrollo.

Cada criterio se evaluará en una escala de 1 a 5, donde cada valor en la escala se detalla en la Tabla 1:

Tabla 1: Escala para criterios en la evaluación de algoritmos.
Fuente: Elaboración propia.

Criterio	Escala
Dificultad (D)	1: Muy fácil de comprender. 2: Relativamente fácil de comprender. 3: Moderadamente difícil de comprender. 4: Difícil de comprender. 5: Muy difícil de comprender.
Utilidad (U)	1: No cumple con los requisitos del proyecto. 2: Cumple parcialmente con los requisitos. 3: Cumple con la mayoría de los requisitos. 4: Cumple con todos los requisitos. 5: Supera los requisitos del proyecto.
Escalabilidad (E)	1: No es escalable en absoluto. 2: Poco escalable. 3: Moderadamente escalable. 4: Bastante escalable. 5: Altamente escalable.
Mantenibilidad (M)	1: Muy poco costoso de mantener. 2: Poco costoso de mantener. 3: Moderadamente costoso de mantener. 4: Costoso de mantener. 5: Muy costoso de mantener.
Factibilidad (F)	1: Muy largo y poco factible. 2: Largo y difícil de cumplir en el tiempo previsto. 3: Adecuado en términos de tiempo. 4: Relativamente corto y factible. 5: Muy corto y altamente factible.

Para obtener un puntaje, se utilizará la siguiente fórmula:

$$Puntaje = 0.2(5 - D) + 0.2U + 0.1E + 0.2(5 - M) + 0.3F$$

Los pesos asignados a cada criterio dependen específicamente de los requerimientos del proyecto y prioridades del equipo de desarrollo. Por lo que la dificultad y el costo de mantención tienden a restar y la utilidad, escalabilidad y factibilidad suman, siendo esta última la que tiene un mayor impacto en la fórmula.

Finalmente, los puntajes asignados a cada algoritmo se encuentran en la siguiente tabla

Tabla 2: Análisis de los algoritmos.
Fuente: Elaboración propia.

Algoritmo	Dificultad	Utilidad	Escalabilidad	Mantenibilidad	Factibilidad	Puntaje
Algoritmo Relojes Lógicos	1	3	5	3	5	3.8
Algoritmo de Vectores Lógicos	2	5	3	3	4	3.6
Algoritmo de Berkeley	3	1	1	2	2	1.9
Algoritmo del Semáforo de Dijkstra	1	5	5	4	5	4.0
<i>Spinlocks con Test and Set</i>	1	5	1	3	2	2.9
Algoritmo de Peterson	1	1	2	4	5	2.9

Analizando la tabla, el algoritmo del semáforo de Dijkstra parece ser una buena opción para permitir la edición colaborativa. Considerando dividir el contenido de cada contrato en estructuras modulares llamadas “Secciones” se puede implementar este algoritmo para permitir que un solo usuario edite la Sección a la vez.

Sin embargo, implementar un algoritmo puede no ser suficiente, más adelante se verá que hay situaciones como la inactividad del usuario que no pueden ser abordadas correctamente con el semáforo, por lo que se ha de implementar un algoritmo que sigue la idea de los relojes y vectores lógicos para evitar colisiones y asegurar que el usuario siempre esté al tanto de la última versión del contrato, evitando así pérdidas de información.

Por otro lado, la idea de los *Spinlock* es interesante, sin embargo al utilizarlos con *Test and Set* se vuelve poco escalable, por lo tanto infactible.

CAPÍTULO 4: PROPUESTA DE SOLUCIÓN

4.1 Propuesta de Diseño

Actualmente, SOFI cuenta con un sistema desarrollado que simplifica la generación de documentos contractuales mediante el llenado de una plantilla en Word con datos proporcionados a través de un formulario. Este enfoque automatizado agiliza el proceso, permitiendo la trazabilidad del documento a medida que avanza por distintas áreas, como Legal y Abastecimiento.

Sin embargo, al revisar el diagrama BPMN del proceso actual en la *figura 19*, se observa que no contempla un paso específico para la redacción del documento, ya que este se genera automáticamente. Este método automatizado presenta un desafío significativo, ya que excluye la participación activa de las partes interesadas durante la redacción involucrando a los otros roles solo en el proceso de revisión y firma.

Esta situación es motivo de preocupación, especialmente al considerar la cantidad de contratos que se firman sin que las partes tengan un conocimiento detallado de su contenido. Para abordar esta problemática, se propone incorporar un proceso de redacción más transparente y colaborativo en SOFI.

Esto puede lograrse mediante la introducción de etapas específicas que involucren a los distintos roles en la creación del documento antes de llegar a la fase de firma, promoviendo así una comprensión más profunda y una participación más activa en la redacción del contrato. De esta manera, los interesados podrán tener una mayor claridad de los documentos generados así como mayor confianza y satisfacción en el proceso.

Para alcanzar este objetivo, es imperativo implementar una serie de modificaciones en el sistema existente. Estas modificaciones implican la incorporación de nuevos componentes y servicios dentro la arquitectura de la solución, así como ampliar las funcionalidades de todos los roles presentes en el actual SOFI.

Para finalizar la propuesta, dada la complejidad inherente de estas mejoras, se prepara una demostración no funcional que exhibe el diseño de la interfaz gráfica del sistema. Esta demostración destaca las nuevas funcionalidades y ajustes adicionales a implementar, con el fin de asegurar que el sistema, en su versión mejorada, cumpla con las necesidades operativas y estratégicas específicas de la empresa minera aportando valor al proceso de redacción de contratos.

4.1.1 Diagrama de Procesos

En la figura 19, se visualiza el diagrama de procesos BPMN actual de SOFI, que en pocas palabras separa el proceso en 4 etapas:

1. Ingeniero de Contratos completa el formulario de creación del contrato
2. Revisión del contrato por parte de Área Legal y/o Área de Abastecimiento
3. Firma del Proveedor
4. Firma del Apoderado

La propuesta de solución consiste en agregar un editor post formulario de creación como se detalla en la figura 20. Este editor permitirá al IdC invitar a usuarios de diferentes roles para revisar, editar y debatir su contenido mediante chats, de modo que el proceso sea el siguiente:

1. Ingeniero de Contratos completa el formulario de creación del contrato
2. **Redacción Colaborativa (Proveedor, Área Legal y Área de Abastecimiento)**
3. Revisión del contrato por parte de Área Legal y/o Área de Abastecimiento
4. Firma del Proveedor
5. Firma del Apoderado

Adicionalmente, es fundamental destacar algunas reglas de negocios que se mantienen entre las dos versiones de SOFI y que se pueden observar en ambos BPMN:

- Aunque en su defecto, el contrato esté configurado para no requerir revisión por parte del Área Legal, si el valor del contrato excede los USD \$5 millones, es obligatorio que el Área Legal revise.
- Un Ingeniero de Contratos que completa el formulario de creación de Contrato, puede derivar dicho contrato a otro Ingeniero de Contratos.
- El Área Legal, a su vez, se divide en: Área Legal Corporativa y Área Legal Externa, entre ellas, también se puede delegar el contrato para su revisión.
- Si durante la revisión del contrato, el Área Legal o el Área de Abastecimiento indican que “Hay observaciones”, el contrato vuelve al Ingeniero de Contrato para ser modificado.
- Si el contrato es rechazado por el Apoderado o por el representante legal del Proveedor, entonces finaliza el proceso.

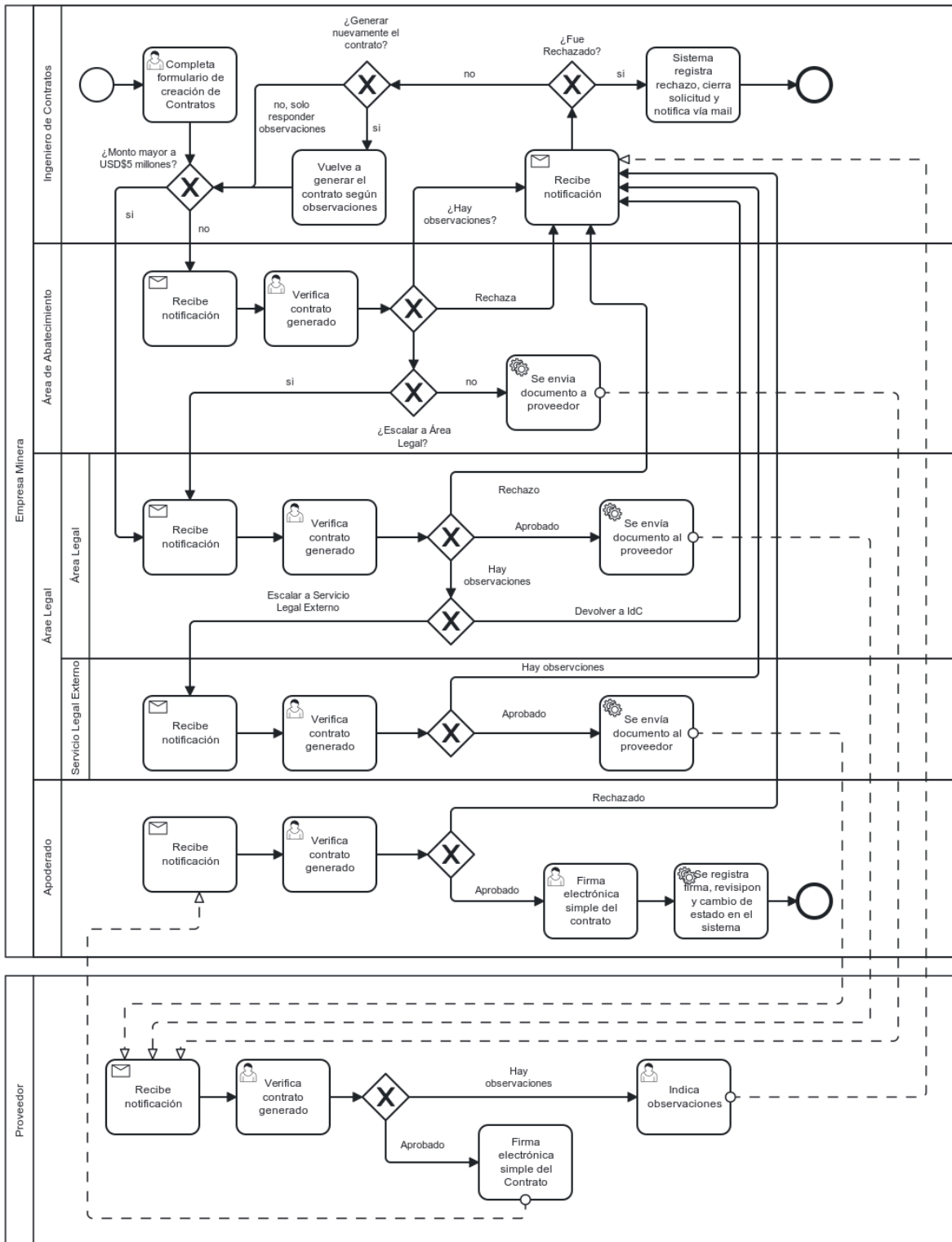


Figura 19: Diagrama de Procesos SOFI Actual.

Fuente: Elaboración propia.

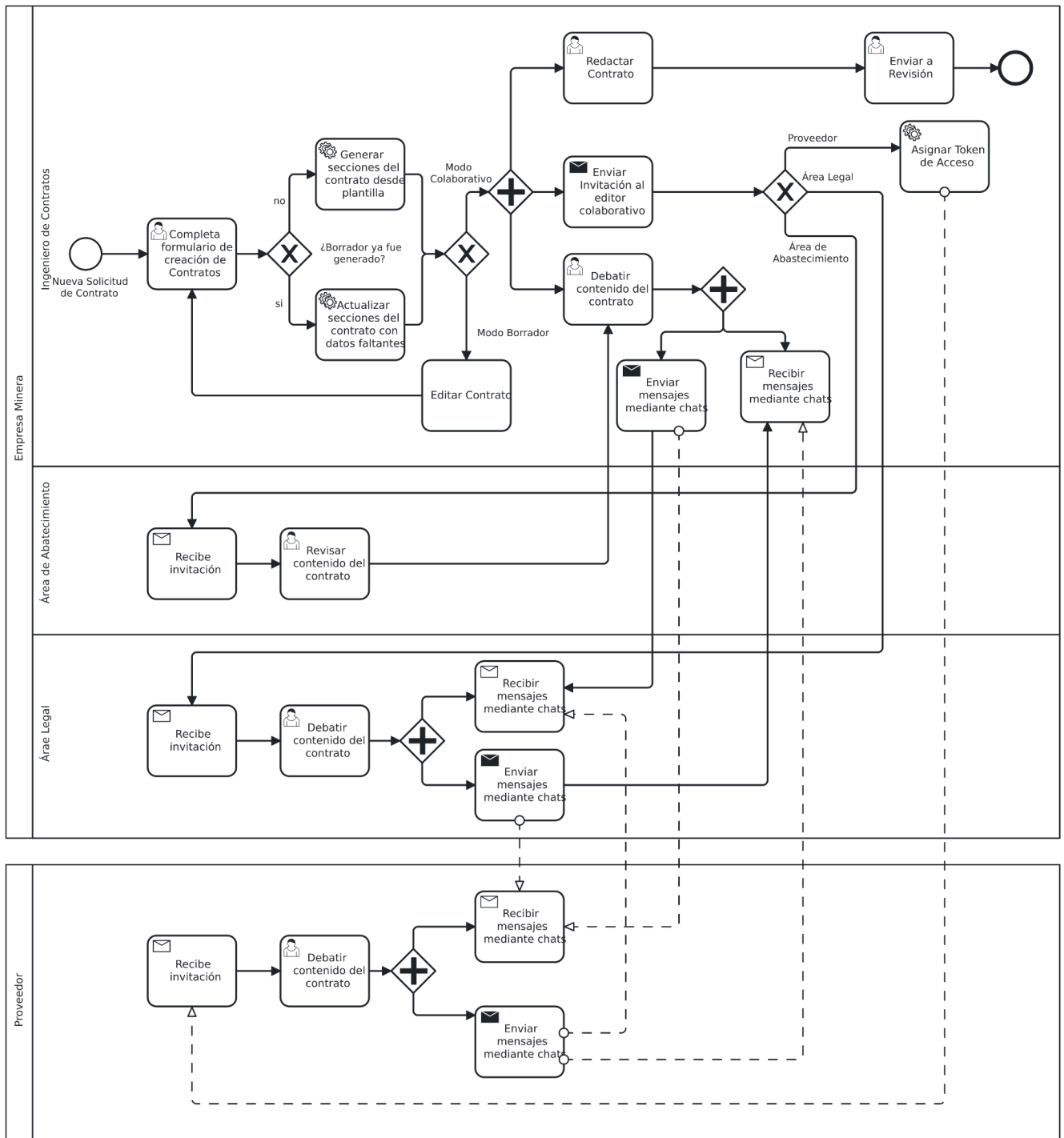


Figura 20: Diagrama de Procesos SOFI mejorado - Etapa de Redacción.
Fuente: Elaboración propia.

4.1.2 Arquitectura de la Solución

Para llevar a cabo la implementación de la solución propuesta, resulta esencial la incorporación de nuevos componentes a la arquitectura existente. En este contexto, el uso de un diagrama de arquitectura se revela como una herramienta crucial para diseñar y planificar de manera efectiva la estructura del sistema antes de su despliegue.

Este diagrama ofrece una representación visual que permite observar de manera integral los componentes propuestos, así como comprender sus interacciones y dependencias. Al proporcionar esta visión global, facilita una planificación más precisa y una toma de decisiones informada durante todas las etapas del desarrollo. Además, sirve como una guía visual que puede ser compartida y discutida entre los miembros del equipo de desarrollo y otros *stakeholders*.

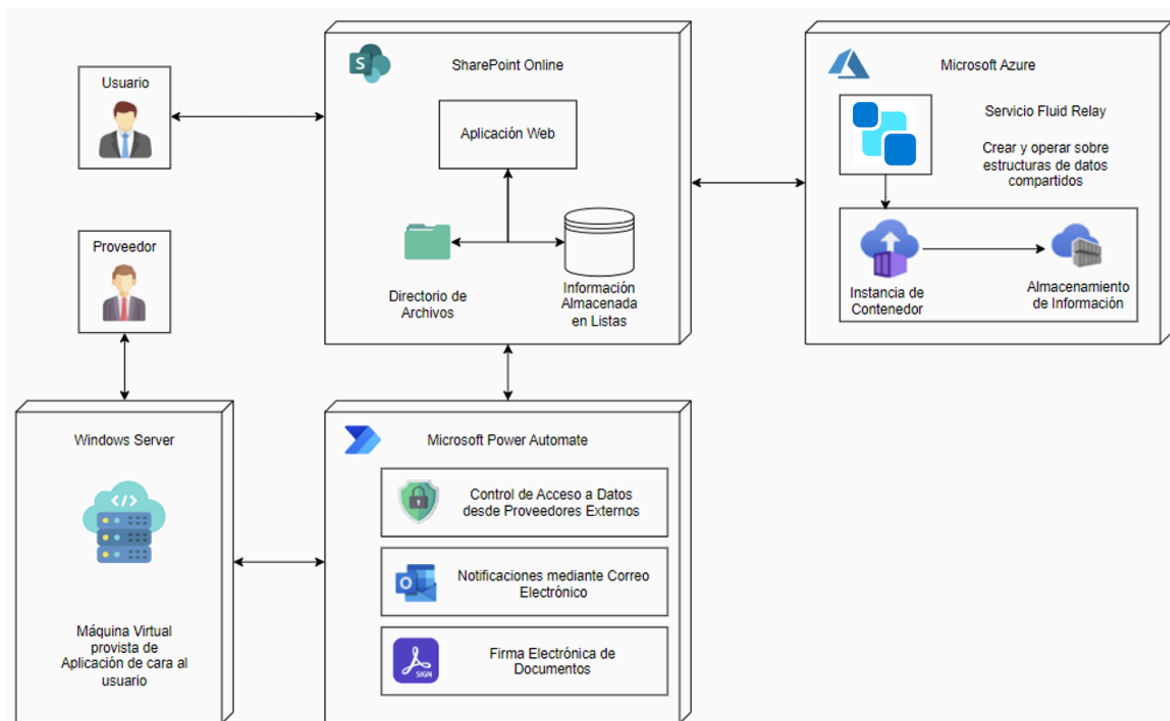


Figura 21: Arquitectura de la Solución.

Fuente: Elaboración propia.

Como se puede visualizar en la **figura 21**, los usuarios tendrán la capacidad de acceder a SOFI desde SharePoint Online mediante el inicio de sesión con sus cuentas de correo empresarial. En esta plataforma, la información relativa a los contratos y al sistema de mensajería se almacenará de manera organizada en listas, mientras que los contratos en formato .docx o .pdf podrán ser almacenados en un directorio de archivos provisto por la misma plataforma.

Para abordar los aspectos de mensajería y edición colaborativa, se implementará el servicio Azure Fluid Relay, el cual permite comunicación de baja latencia. Este servicio permitirá la creación de un objeto compartido alojado en un contenedor, accesible para todos los usuarios conectados. Cualquier modificación realizada en este objeto se notificará de manera instantánea a través de un *broadcast* a los demás usuarios.

En relación a los contratos, se establecerán notificaciones automatizadas mediante Microsoft Power Automate. Estas notificaciones se desencadenarán en eventos específicos, como la finalización de la edición, la revisión por parte del Área de Abastecimiento o Área Legal, y cuando el contrato se envía para que sea firmado por parte del proveedor o Apoderado.

En cuanto a la interacción con proveedores, que no tienen acceso directo a SharePoint Online debido a que no pertenecen a la organización, se ha implementado una solución mediante una máquina virtual con Windows Server. Esta máquina se comunica con Microsoft Power Automate como Middleware para gestionar de manera segura y eficiente el acceso a los datos pertinentes en SharePoint Online.

4.1.3 Propuesta de Diseño no Funcional

Previo al inicio del proyecto, se elaboró una propuesta de diseño no funcional para presentar a los *stakeholders*, brindándoles una visión clara de la mejora planificada. La propuesta abarca diferentes tipos de documentos contractuales:

- Contrato de Servicios
- Modificación de Contrato
- Contrato de Suministros
- Finiquito de Contrato
- Contrato de Construcción
- Contrato de Servicios de Ingeniería

Cada uno de estos documentos cuenta con una plantilla específica que se proporcionará al Ingeniero de Contratos, lo que agiliza el proceso evitando que comience la redacción desde cero.

En la figura 22, se destaca una sección específica del contrato denominada "TÍTULO Y CUERPO SUSTANTIVO", que corresponde a la portada. Cada sección sigue un formato uniforme, presentando un editor de texto enriquecido a la izquierda para modificar el contenido y una sección de comentarios a la derecha para discutir el contenido de esa sección en particular.

Asimismo, se incluyen funcionalidades adicionales mediante botones claves, como "GUARDAR CAMBIOS", "ELIMINAR SECCIÓN" y "AJUSTES". En este último, se puede especificar la capacidad de edición de la sección o si se requiere aprobación por parte del Área Legal o del Área de Abastecimiento.



Figura 22: Interfaz del Editor Colaborativo para el Administrador SOFI.
Fuente: Elaboración propia.

Es relevante destacar la presencia de los botones "NUEVA SECCIÓN" en la parte superior e inferior de cada sección, facilitando la adición dinámica de secciones al contrato.

Un elemento crucial es la barra de navegación (*navbar*), que muestra avatares para identificar a los usuarios que están conectados y editando simultáneamente el documento. Además, se incorporan botones que permiten acceder al historial del documento y a un Chat Global.

Otro aspecto fundamental, es el uso de variables, identificadas entre llaves: `{{Variable}}` o `{{=Variable.Propiedad}}`. Este tipo de sintaxis será sustituida por los valores proporcionados por el Ingeniero de Contratos al momento de completar el formulario de creación de contrato.

A modo de ejemplo, si el Ingeniero de Contratos ingresa en el campo RUT del Proveedor el valor 11.111.111-1, entonces la variable `{{=Proveedor.Rut}}` será reemplazada por dicho valor.

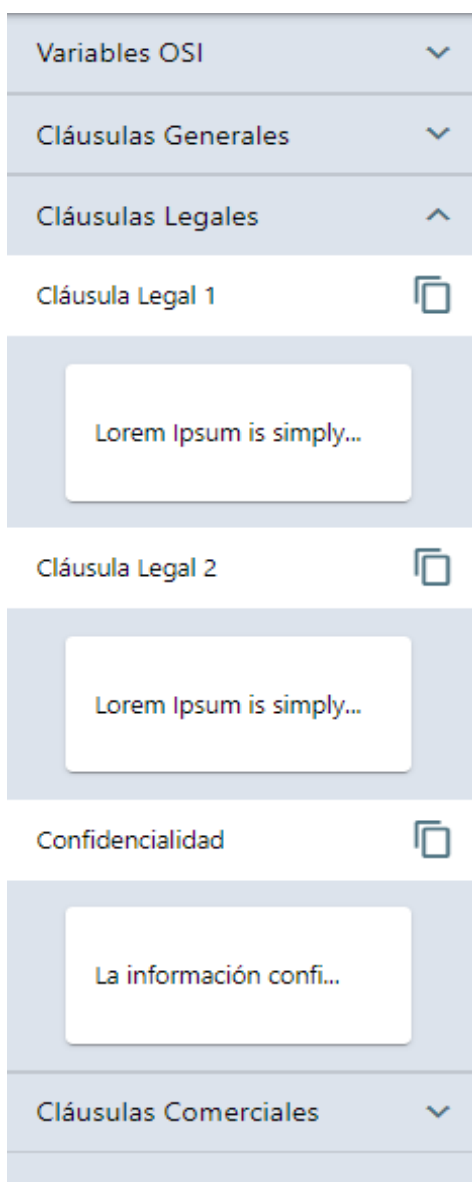


Figura 23: Panel de Cláusulas.
Fuente: Elaboración propia.

Otra herramienta valiosa para optimizar el proceso de redacción del contrato es un panel lateral que alberga cláusulas predefinidas por la empresa, abarcando diversos aspectos contractuales (*figura 23*). Estas cláusulas se dividen en tres categorías: Generales, Legales y Comerciales, y permiten al usuario copiar y pegar fácilmente su contenido en las secciones que estime conveniente con tan solo un clic.

Además de las cláusulas, se incluyen las Variables OSI, exclusivas para el Administrador SOFI. Estas variables contienen datos fundamentales compartidos por todos los contratos, como las entidades involucradas, representantes legales, códigos del contrato, entre otros.

Naturalmente, con el fin de asegurar que el contenido de este panel permanezca constantemente actualizado, se considera la implementación de un mantenedor. Este mantenedor, reservado exclusivamente para el rol de Administrador SOFI, proporciona la capacidad de crear, editar y eliminar tanto cláusulas como variables OSI según sea necesario.

También, la interfaz del editor se presenta de manera consistente para todos los roles, variando principalmente en las funcionalidades proporcionadas. Por esta razón, se ha desarrollado una vista específica para cada rol y para el mismo contrato, destacando las características más relevantes y pertinentes a las responsabilidades particulares de cada usuario.

Adicionalmente, se propone que las secciones que necesiten ser revisadas se destaquen con un color rojizo claro, fácilmente identificable. Además, se sugiere resaltar los cambios efectuados por el Ingeniero de Contratos en el contenido predeterminado suministrado en la plantilla, siendo de color verde el contenido agregado, y de color rojo lo quitado. Esta funcionalidad deberá estar disponible para los roles con capacidad de revisión, es decir, Área Legal y Área de Abastecimiento (*figura 24*).



Figura 24: Interfaz del Editor Colaborativo - Sección que requiere revisión.

Fuente: Elaboración propia.

Además, cada contrato, deberá contar con dos Chats Globales que permitan la comunicación entre:

- Chat Global Interno: IdC + AL
- Chat Global Externo: IdC + AL + Proveedor

De la misma forma, se propone que cada sección del contrato deberá contar con 3 chats:

- Chat 1: IdC + AL
- Chat 2: IdC + AL + Proveedor
- Chat 3: IdC + Proveedor

En este punto, es importante resaltar que durante el desarrollo del proyecto, se optó por reducir la cantidad de chats e incorporar al Área de Abastecimiento, quedando de la siguiente manera tanto para el chat global como para los chats de cada sección del contrato:

- Chat Interno: IdC + AA + AL
- Chat Externo: IdC + Proveedor

4.1.4 Diagrama de Casos de Uso

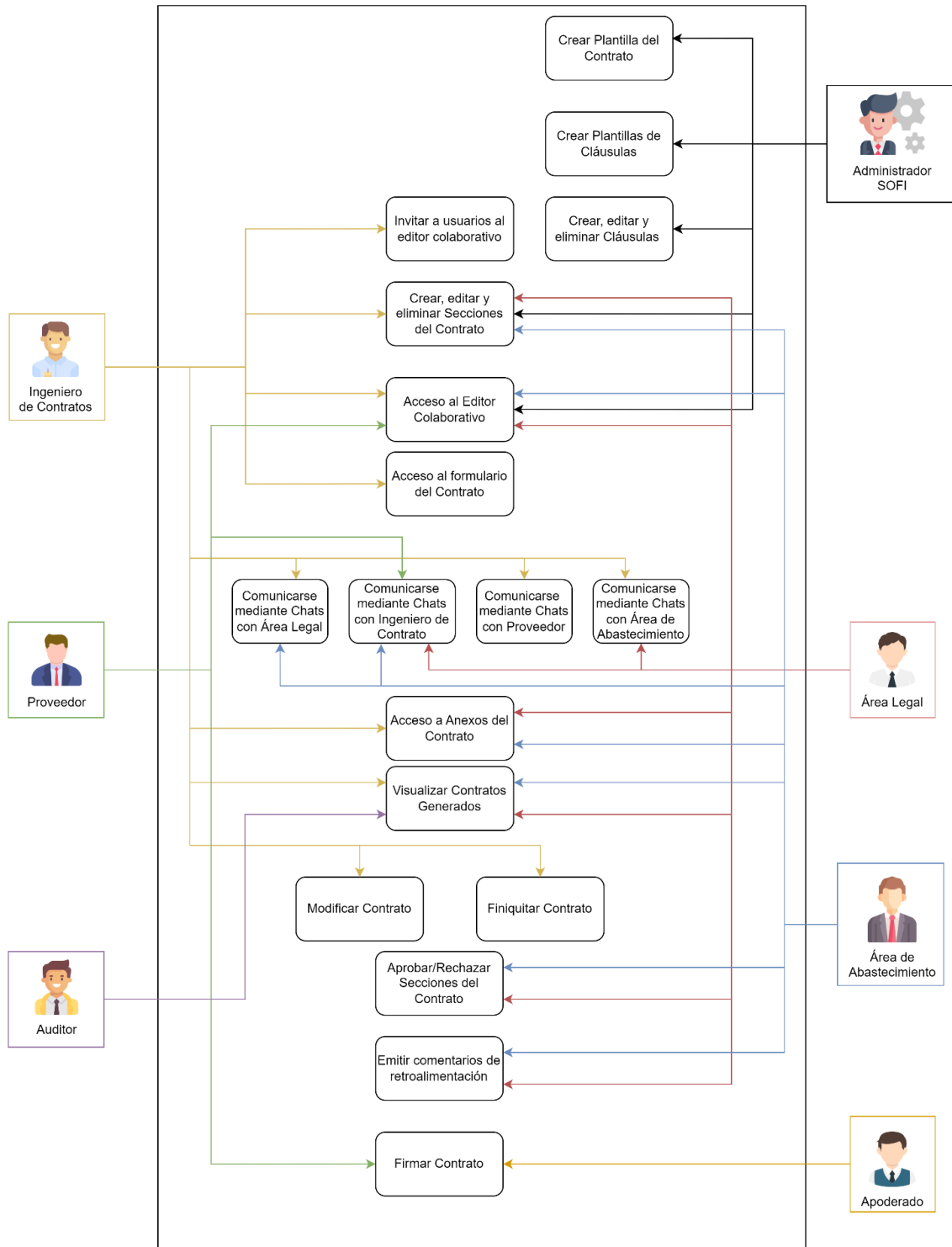


Figura 25: Diagrama de Casos de Uso.

Fuente: Elaboración propia.

➤ Administrador SOFI o editor de plantillas

Corresponde a un perfil de administrador, que podrá configurar la plantilla de un contrato, indicando las secciones que tendrá por defecto, si son editables o bien si requieren aprobación, ya sea del Área Legal o del Área de Abastecimiento. Además estará a cargo de mantener actualizadas las diversas cláusulas que se pueden incluir en el contrato.

➤ Ingeniero de Contratos

El Ingeniero de Contrato será el encargado de crear los contratos, indicando el tipo de contrato, sociedades, anexos, entre otros mediante un formulario. Una vez completado el formulario, podrá acceder al editor, donde el sistema tomará los datos ingresados en el formulario y generará un documento usando como base una plantilla, en ella podrá modificar el contenido del contrato, crear y eliminar secciones, añadir cláusulas e invitar a un tercero, ya sea del Área Legal, Área de Abastecimiento o Proveedor.

Durante el proceso de redacción, podrá interactuar mediante chats con el Área Legal, Área de Abastecimiento y Proveedor.

➤ Área Legal y Área de Abastecimiento

Las funcionalidades del Área Legal y del Área de Abastecimiento estarán enfocadas a la revisión del contrato, donde una vez finalizada la redacción por parte del Ingeniero de Contratos, podrá revisar las secciones que incluyan cláusulas legales, sugerir cambios o bien aprobar o rechazar. Sin embargo, si durante la redacción el Ingeniero de Contratos lo invita al editor, podrá ser partícipe de los chats para discutir el contenido de las secciones del contrato. También podrá editar las secciones del contrato, sin embargo no es el objetivo principal de este rol.

Adicionalmente, hay otros roles, Interlocutor, Auditor SOFI y Apoderado. Sin embargo, sólo pueden visualizar el contrato o ingresar firma, no son partícipes en el proceso de redacción.

4.2 Editor Colaborativo

Para el desarrollo del editor, se ha de tener en cuenta lo siguiente:

- Un contrato, consta de una o más estructuras modulares denominadas “Secciones”.
- Una sección, corresponde a una porción del contrato, podría ser un capítulo, una tabla o bien, un conjunto de cláusulas.
- Cada sección del contrato puede incluir cláusulas legales o comerciales.
- Si el contrato contiene cláusulas legales, requiere aprobación del Área Legal.

Dado el tiempo estimado para llevar a cabo el proyecto, fue necesario seleccionar una ruta para realizar el trabajo colaborativo.

Se exploraron dos alternativas. La primera consistió en permitir que todos los usuarios editaran simultáneamente el contenido de la sección del contrato. No obstante, la implementación de esta opción podría afectar a la trazabilidad del contrato, si todos editan a la vez, entonces ¿Quién editó? ¿Qué editó? ¿Cuándo?.

Además, producto de la latencia se podrían dar condiciones de carrera, ya que cuando se estaba realizando la implementación del sistema se observó que el contenido ingresado por un usuario sobrescribía por completo los cambios ingresados por otro usuario previamente, generando inconsistencias. Adaptar el código para manejar este tipo de situaciones podría tomar demasiado tiempo y esfuerzo. Por este motivo, se optó por adaptar diversos algoritmos investigados en el estado del arte, considerando a cada usuario conectado al editor de contratos como un nodo lógico en un sistema distribuido.

Finalmente, el algoritmo escogido para la edición fue el semáforo de Dijkstra, sobre el cual se incorporaron algunas modificaciones para el control de acceso a la sección crítica y control de versiones.

4.2.1 Creación y eliminación de Secciones del Contrato de forma dinámica

Al acceder al contrato en la plataforma, el editor se encontrará activo siempre que la revisión del Área Legal y Área de Abastecimiento no se hayan concretado para una sección específica, permitiendo al Ingeniero de Contratos realizar ediciones incluso durante el proceso de revisión.

Además, se mostrarán los botones “Agregar Sección” antes y después de cada sección del contrato y el botón “Eliminar Sección” dentro de la misma sección (*figura 26*).



Figura 26: Editor activo con botones "Agregar Sección" y "Eliminar Sección".
Fuente: Elaboración propia.

Para facilitar la creación y eliminación dinámica de secciones, y garantizar que todos los usuarios tengan visibilidad de estos cambios, se ha incorporado la propiedad "Contexto" en el Objeto Compartido. Esta propiedad opera como una cola de cambios donde los usuarios registran las modificaciones que realizan en el contrato.

Cuando un usuario agrega, actualiza o elimina una sección del contrato, la acción se lleva a cabo mediante la API de SharePoint Online, que agrega, edita o elimina el registro en la lista de SharePoint Online.

Una vez que se recibe la respuesta de esta solicitud, se ingresa el cambio al "Contexto" y Fluid Relay se encarga de notificar el cambio a los demás usuarios.

Este flujo de eventos se detalla en la [figura 27](#), que ilustra el proceso del sistema después de presionar el botón "Agregar Sección".

Es relevante señalar que la notificación se emite de manera similar independiente del tipo de evento (*create, delete, update*).

Entorno Colaborativo

Usuarios A, B, C se encuentran conectados al contenedor creado para el contrato en Azure Fluid Relay.

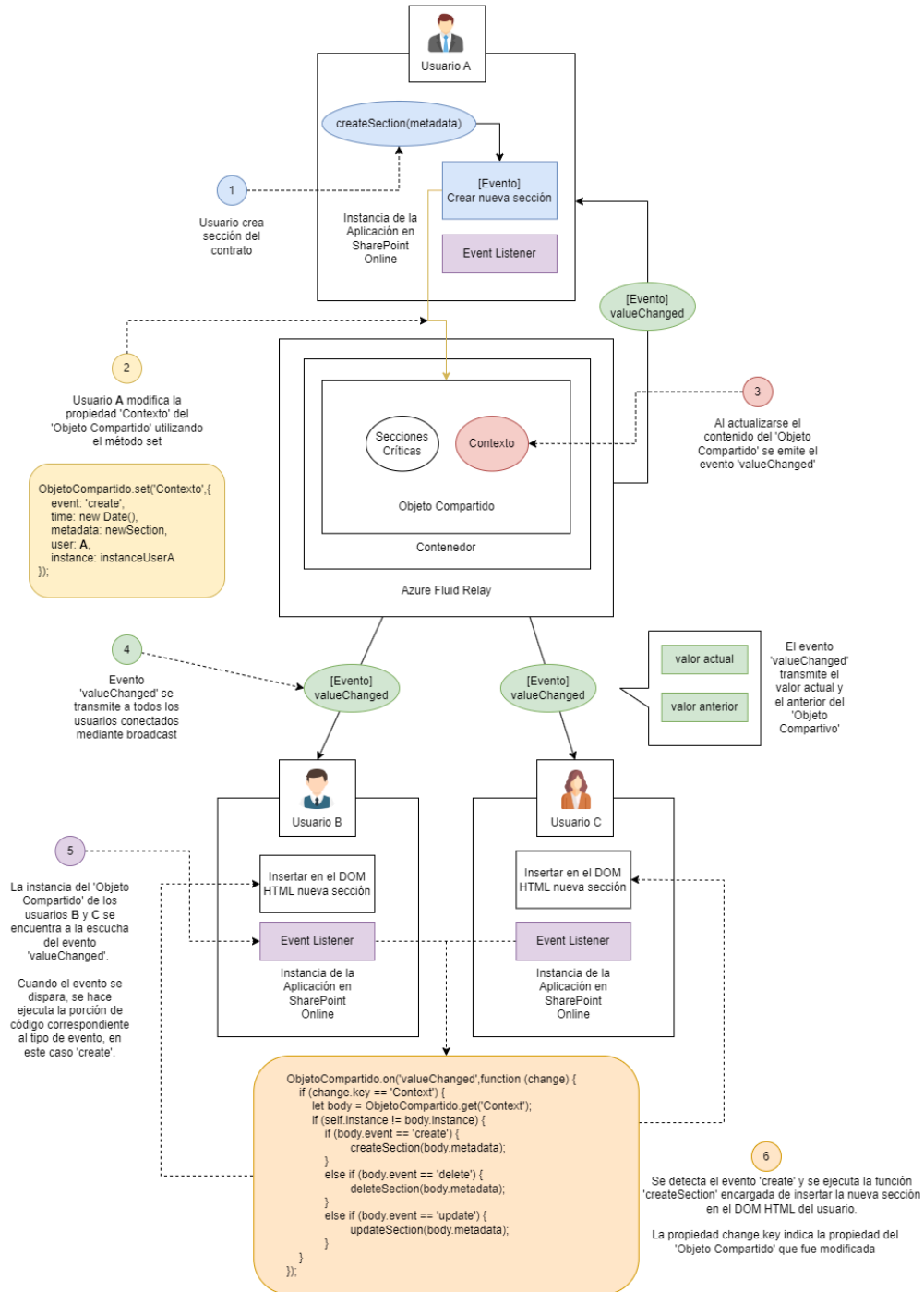


Figura 27: Lógica de Crear Sección del Contrato en ambiente colaborativo.

Fuente: Elaboración propia.

4.2.2 Control de Acceso a Sección Crítica con Azure Fluid Relay

La idea es sencilla, cada sección del contrato se considera una **sección crítica**. Para que un usuario pueda realizar ediciones, deberá acceder a la sección crítica correspondiente. Una vez que un usuario inicia la edición, se implementa un algoritmo que impide a cualquier otro usuario realizar ediciones simultáneas en la misma sección. Azure Fluid Relay y su biblioteca Fluid Framework se presentan como la solución ideal para esta dinámica.

El proceso comienza conectándose al servicio en Azure y creando un contenedor dedicado para cada contrato. Dentro de este contenedor, se establece un Objeto Compartido, que a través de *websockets* facilita la apertura de un canal bidireccional que permite a los usuarios mantenerse constantemente actualizados sobre los cambios en el Objeto Compartido.

Este objeto compartido opera de manera análoga a un *hash map* en JavaScript, donde se utiliza el método *SET* para asignar un valor y el método *GET* para obtenerlo. Cada vez que un usuario actualiza el valor de este Objeto Compartido, se inicia un algoritmo de propagación que notifica a todos los usuarios conectados al contenedor el cambio realizado mediante un evento "*valueChanged*".

Así, cada vez que un usuario se conecta al contenedor en Azure Fluid Relay, se guarda de manera local una copia del Objeto Compartido que se mantiene a la escucha de los cambios sobre el objeto original almacenado en el contenedor de Azure. Mientras la conexión permanezca estable, cualquier usuario que utilice el método *SET* para actualizar una propiedad del objeto compartido activará el evento "*valueChanged*" que se propagará a todos los usuarios conectados. El evento, señala la propiedad "*key*" o clave del objeto que fue modificada, así cada usuario, al recibir el evento, podrá utilizar el método *GET* para obtener los cambios de dicha propiedad y ejecutar la lógica correspondiente.

Una vez explicado el funcionamiento de Azure Fluid Relay y el Objeto Compartido se puede comprender la lógica de edición que se representa en la [figura 28](#), cuando el usuario abre la aplicación en SharePoint Online, el sistema hace un revisión periódica del estado compartido para cada sección del contrato, considerando que:

- Si la sección se encuentra ocupada, se actualiza el DOM para que la muestre como NO editable, e indique quién la está editando.
- Si la sección se encuentra libre, actualiza el DOM para que se muestre editable.
- Si el usuario desea editar y la sección se encuentra libre, entonces puede entrar a la sección crítica ingresando en la "*key*" asignada a la sección sus credenciales y una marca de tiempo del momento que indica año, mes, día, minuto, segundo y fracciones de segundo. Así los demás usuarios sabrán que la sección se encuentra ocupada, quién la está editando y desde cuándo.

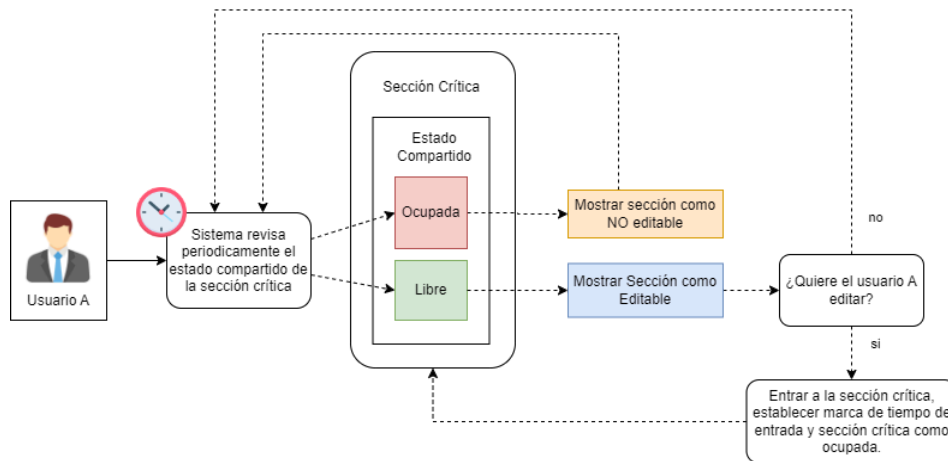


Figura 28: Revisión de Estados de la Sección del Contrato.
Fuente: Elaboración propia.

A continuación en la *figura 29*, un ejemplo de la interfaz de una sección no editable del contrato. Se observa que en la parte superior de la sección se especifica la persona que está realizando la edición.

A pesar de que la sección no está en modo de edición, el usuario aún tiene la capacidad de enviar mensajes a través del chat ubicado en el lado derecho.



Figura 29: Ejemplo de sección en estado NO editable.
Fuente: Elaboración propia.

En caso contrario, la *figura 30* muestra una sección que se encuentra editable, y ya que nadie la está editando se pueden visualizar los botones “Editar Sección” y “Eliminar Sección”. Presionar el botón “Editar Sección” le permite al usuario ingresar a la sección crítica y comenzar a editar.

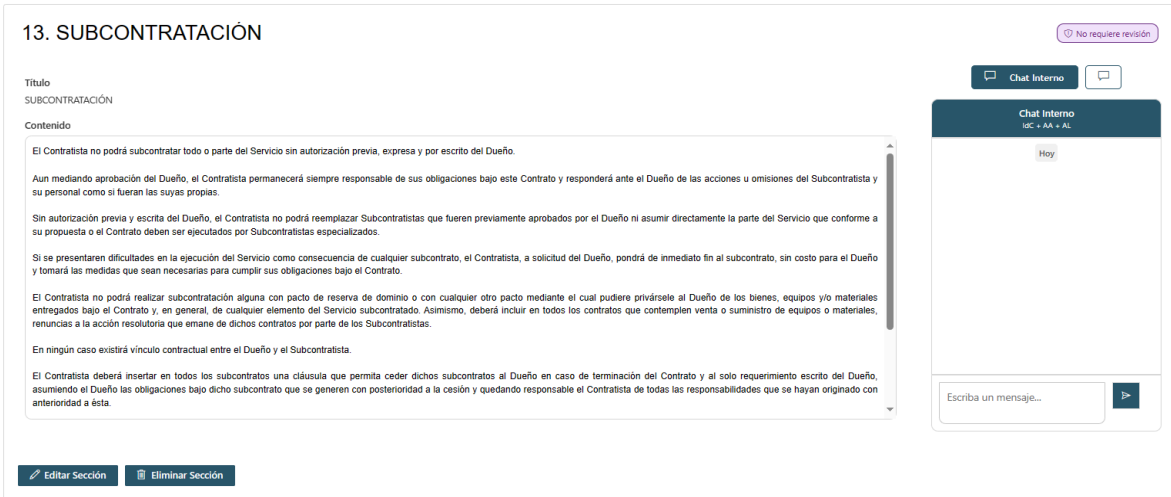


Figura 30: Ejemplo de sección en estado editable.
Fuente: Elaboración propia.

Por último, un ejemplo de una sección del contrato siendo editada (*figura 31*). En la imagen se puede apreciar que durante la edición, la sección se vuelve de color anaranjado claro, además se activan los campos Título y Contenido.

Adicionalmente, los botones cambian en comparación a la *figura 30*, el botón “Editar Sección” se oculta y se muestran los botones “Guardar Cambios” y “Cancelar Edición”.

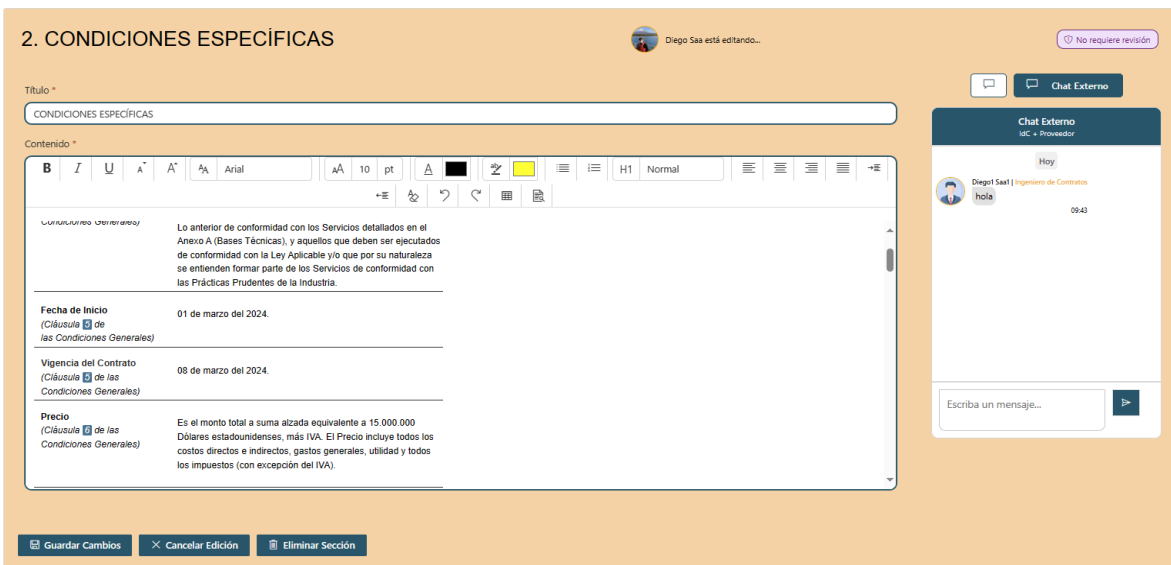


Figura 31: Ejemplo de sección durante la edición.
Fuente: Elaboración propia.

Si se presiona el botón “Guardar Cambios”, se libera la sección crítica y notifica los cambios a todos los usuarios conectados, iniciando el algoritmo para actualizar el DOM.

4.2.3 Avatares

Los avatares son una excelente manera de informar a los usuarios sobre la presencia de otros participantes en el editor. Por esta razón, se ha integrado en la barra de navegación un listado que exhibe los usuarios conectados, incluyendo su imagen de perfil, nombre y rol con la pseudo clase :hover de CSS.

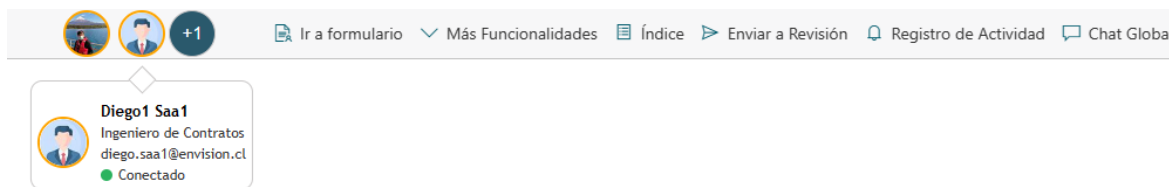


Figura 32: Avatares de Usuarios Conectados.
Fuente: Elaboración propia.

4.2.4 Manejo de Inactividad del Usuario

En la gestión de acceso a secciones críticas, es crucial considerar la variable de inactividad.

¿Qué sucede si un usuario inicia la edición de una sección y pierde la conexión? ¿O si, por alguna razón, edita la sección, pero no guarda los cambios realizados? En tales situaciones, los demás usuarios conectados podrían ver la sección bloqueada, ya que la edición estaría aparentemente disponible para otro usuario y el contenido de la sección no será actualizado hasta que se guarden los cambios y libere dicha sección crítica.

Por ende, es imperativo establecer un algoritmo que detecte la inactividad de un usuario y libere la sección crítica correspondiente. Para abordar este desafío, la implementación de marcas de tiempo emerge como una solución eficaz. Estas marcas permiten rastrear el intervalo transcurrido desde la última acción del usuario, posibilitando la liberación automática de la sección en caso de inactividad prolongada.

Entonces ¿Cómo identificar la última acción del usuario? Los eventos Javascript que se disparan al interactuar con elementos HTML son herramientas valiosas para lograr esto, destacando cuatro de ellos:

- *Input*: Este evento se dispara cuando el usuario interactúa con un campo de entrada. Es el mejor evento para detectar si un campo está siendo editado, no obstante, el evento se dispara cada vez que el usuario ingresa o elimina un carácter, utilizarlo para dar una señal de actividad genera muchas peticiones de red, lo que no sería escalable. Sin embargo, es posible aplicar un algoritmo, para que emita una señal de actividad si dentro de un determinado tiempo $t[s]^8$ se ha disparado el evento. Para ello es necesario revisar periódicamente, si se ha emitido

⁸ [s] Notación de la unidad de medida “segundos”

el evento *input* dentro de los últimos $t[s]$, en cuyo caso, el sistema asumirá que el usuario aún se encuentra editando.

- *Change*: Este evento se dispara cuando el valor de un elemento cambia, por lo que se puede utilizar para detectar modificaciones en los campos de entrada. Sirve para el propósito, dado que funciona de manera similar al evento *input*, pero con una frecuencia considerablemente menor.
- *Focus*: El evento de enfoque ocurre cuando un elemento HTML recibe la atención del usuario. Es perfecto para señalar que el usuario está enfocado en un campo.
- *Blur*: El evento de desenfoque (*blur*) sucede cuando un elemento pierde el foco, es decir, cuando el usuario deja de interactuar con él. Hace exactamente lo contrario a lo que buscamos. No es de gran utilidad.

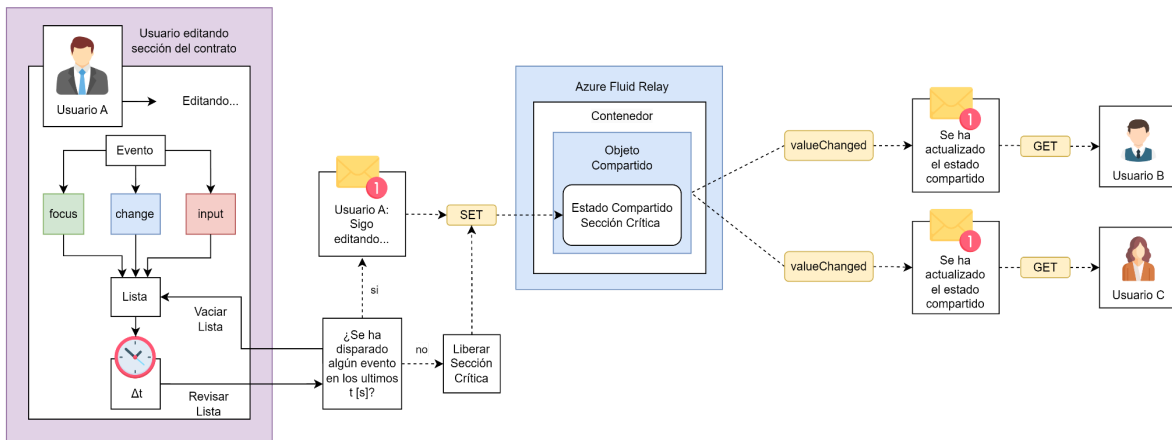


Figura 33: Notificar actividad durante proceso de edición.
Fuente: Elaboración propia.

Cada vez que el usuario dispare los eventos *focus*, *input* o *change*, de manera local, se añade un registro del evento a una lista.

Mientras el usuario se encuentre en modo edición, revisará la lista periódicamente cada $t[s]$, si hay algún registro de evento en la lista, significa que el usuario ha estado activo durante los últimos $t[s]$, por lo que actualiza su marca de tiempo en la sección crítica y vacía la lista en local para comenzar a registrar los eventos de los siguientes $t[s]$. Al actualizar su marca de tiempo, los demás usuarios sabrán que la sección crítica sigue estando ocupada.

Si pasados los $t[s]$ la lista no tiene registros de eventos y el usuario sigue en modo edición sin guardar cambios, entonces se interpretará como inactividad, y se liberará la sección crítica. Los demás usuarios, al notar que la sección crítica fue liberada, pueden volver a editar.

No obstante, hay que considerar dos situaciones:

- Si el usuario no realizó cambios antes de estar inactivo, simplemente cierra el modo de edición y libera la sección crítica (ver [figura 34](#)).

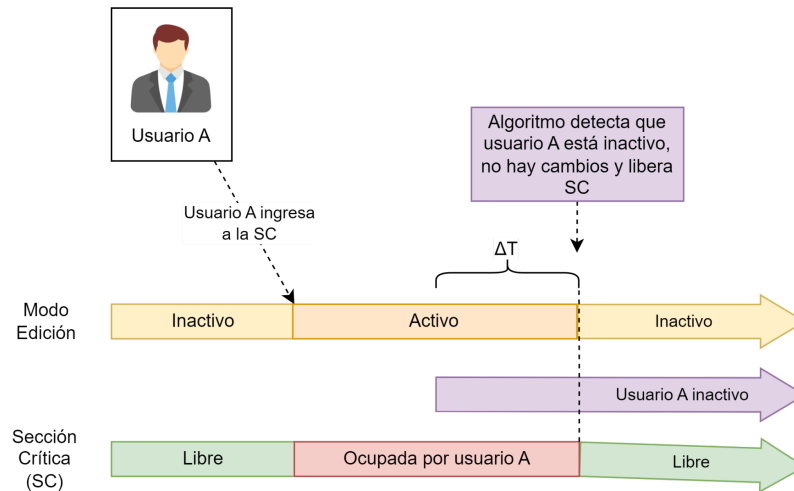


Figura 34: Desactivar modo edición al detectar inactividad.

Fuente: Elaboración propia.

- Si el usuario realizó cambios antes de estar inactivo, entonces libera la sección crítica, sin embargo el modo edición permanece activo (ver [figura 35](#)). Dentro de este proceso, se pueden generar colisiones, tema que se abordará más adelante.

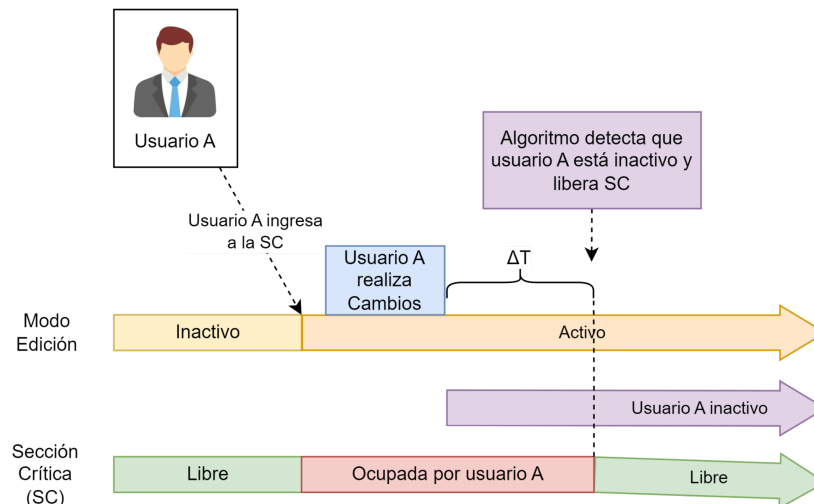


Figura 35: Liberar SC al detectar inactividad sin desactivar el modo edición.

Fuente: Elaboración propia.

4.2.5 Guardar Cambios y Colisiones

Cuando un usuario edita el contrato, la última versión se almacena localmente, y cada sección del contrato permanece atenta a nuevas versiones. Sin embargo, surgen desafíos cuando los usuarios editan el contenido de una sección y guardan cambios sin tener información sobre la última versión. Esta situación puede dar lugar a colisiones o pérdida de información, especialmente cuando se produce inactividad después de realizar modificaciones durante el modo de edición.

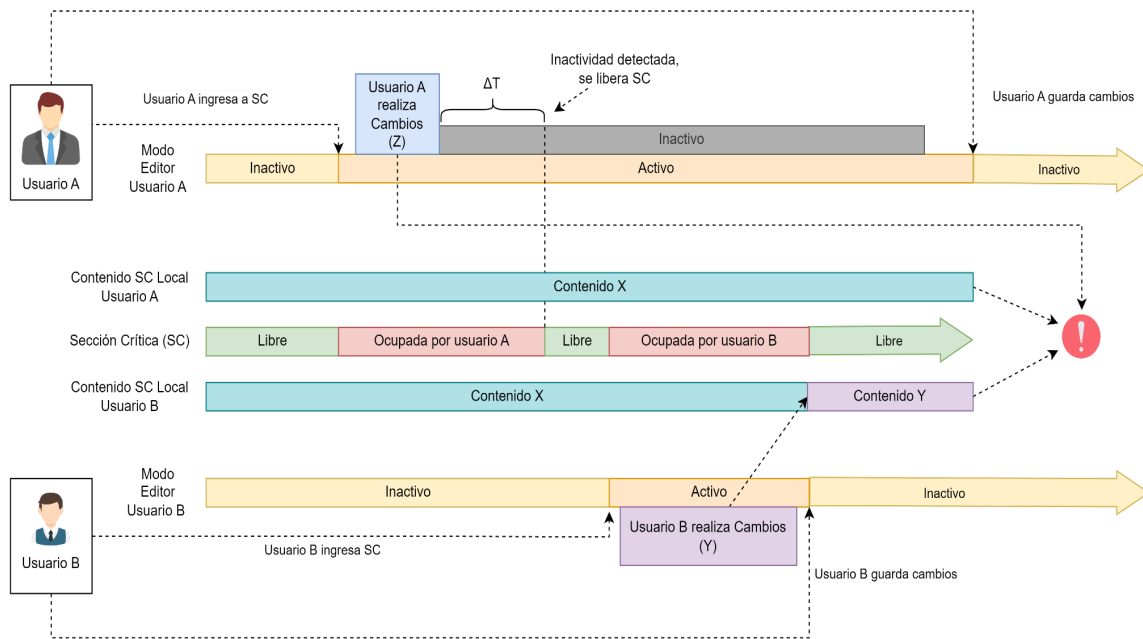


Figura 36: Conflicto al guardar cambios de la Sección del Contrato.
Fuente: Elaboración propia.

Al observar la *figura 36*. Inicialmente, el usuario **A** accede a la SC y comienza la edición del Contenido **X**. Después de un tiempo queda inactivo y el sistema al detectar la inactividad, libera la SC. No obstante, dado que el usuario **A** modificó el Contenido **X** (ahora denominado Contenido **Z**), el sistema mantiene activo el modo de edición. En paralelo, el usuario **B**, al notar que la SC está libre, decide iniciar su edición. Como el usuario **A** no guardó los cambios del Contenido **Z**, el usuario **B** comienza a editar el Contenido **X**.

El usuario **B**, sin contratiempos, realiza modificaciones creando el Contenido **Y**. Al finalizar, guarda los cambios, estableciendo así que la última versión local del usuario **B** y la última versión almacenada en SharePoint será el Contenido **Y**. Luego, el usuario **A** regresa de su periodo de inactividad y decide guardar cambios, generando la siguiente situación indicada en la *Tabla 3*:

Tabla 3: Evolución de los cambios en la Sección del Contrato.

Fuente: Elaboración propia.

Usuario	Versión Inicial	Versión después de cambios	Versión Final
A (Local)	Contenido X	Contenido Z	Contenido Z
B (Local)	Contenido X	Contenido Y	Contenido Z

Como se observa en la *Tabla 3*, se da una situación con condiciones de carrera, los usuarios A y B no tienen en cuenta las modificaciones realizadas por el otro. Si solo se registrarán los cambios, se perdería el Contenido Y introducido por el usuario B, estableciendo el Contenido Z como la última versión y definitiva, es decir, el último en guardar gana.

Esto no puede ser, dado que el contenido en la mayoría de los casos es lo suficientemente relevante como para no ser sustituido o mezclado. Es por ello, que se diseñó una solución basada en los vectores lógicos de Lamport, que permite sincronizar el contenido de las secciones y detectar cuándo se ha perdido el orden, señalando al usuario que su versión no es la más reciente, en otras palabras, se ha detectado una colisión.

El sistema en estos casos, le muestra al usuario la versión con la que ha colisionado, brindándole la autonomía para decidir, según su criterio, si desean conservar o descartar estos cambios. Aunque esta mejora no estaba contemplada en la propuesta inicial de diseño no funcional, se vuelve indispensable para asegurar el correcto funcionamiento del sistema.

4.2.6 Revisiones

Para determinar la necesidad de revisión, cada sección generada a partir de la plantilla incluye una configuración predeterminada establecida por el Administrador SOFI. En esta configuración el Administrador SOFI deberá evaluar el contenido específico o las cláusulas presentes en cada sección e indicar si la sección requiere revisión del Área Legal, del Área de Abastecimiento, ambos, o en su defecto, si no requiere revisión.

Adicionalmente y a modo de referencia visual, el usuario puede observar etiquetas en la esquina superior derecha de cada sección (*figura 37*). Estas etiquetas, presentadas con colores distintivos, indicarán claramente el estado de la sección, ya sea si requiere o no revisión, si está en espera de revisión por parte del Área Legal o Área de Abastecimiento, o si ya ha sido revisada.

También hay etiquetas que indican si “Falta Información”, es decir, el Ingeniero de Contrato debe completar algunos campos, o bien si la sección está configurada como no editable.

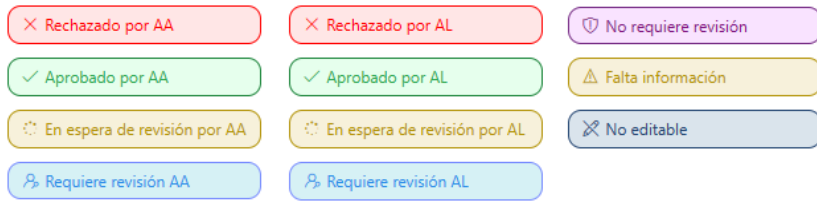


Figura 37: Etiquetas con información sobre la revisión de la Sección.
Fuente: Elaboración propia.

Todos los contratos generados a partir de una plantilla heredarán la configuración que indicó el Administrador SOFI para cada una de las secciones. Así, una vez que se ha finalizado el proceso de redacción por parte del Ingeniero de Contratos, podrá enviarlo a revisión, donde el sistema revisará automáticamente la configuración de las secciones y lo derivará al área que corresponda para continuar el proceso.

En paralelo, cuando se envía el contrato a revisión, se notificará a los responsables del Área Legal y del Área de Abastecimiento según corresponda mediante correo electrónico.

En sus interfaces, las secciones que requieran revisión se destacarán con un tono rojizo claro (*figura 38*). Además, podrán visualizar los cambios efectuados por el Ingeniero de Contratos y tendrán acceso a un panel que les permitirá indicar si aprueba o rechaza y agregar comentarios pertinentes.



Figura 38: Sección que requiere revisión vista como Área Legal.
Fuente: Elaboración propia.

4.3 Sistema de Mensajería

Dentro de los requerimientos del proyecto contemplados en la propuesta de diseño, se propuso que cada sección del contrato tuviera al menos 3 chats, donde los usuarios podrían emitir comentarios referentes al contenido de la sección del contrato. No obstante, durante el desarrollo los requerimientos cambiaron y se optó por incluir al Área de Abastecimiento y dejar solo dos chats, uno interno, para el Ingeniero de Contratos y los revisores de la empresa minera, y uno externo entre el Ingeniero de Contratos y el proveedor. Estos chats estarán disponibles a nivel de roles, lo que significa que cualquier usuario invitado al ambiente colaborativo que tenga uno de los roles especificados en los permisos de cada chat podrá realizar comentarios y visualizar mensajes. Los chats para cada Sección del contrato serán los siguientes:

- Chat Interno: IdC + AA + AL
- Chat Externo: IdC + Proveedor

Adicionalmente, habrá dos chat globales para discutir el contenido del contrato en general.

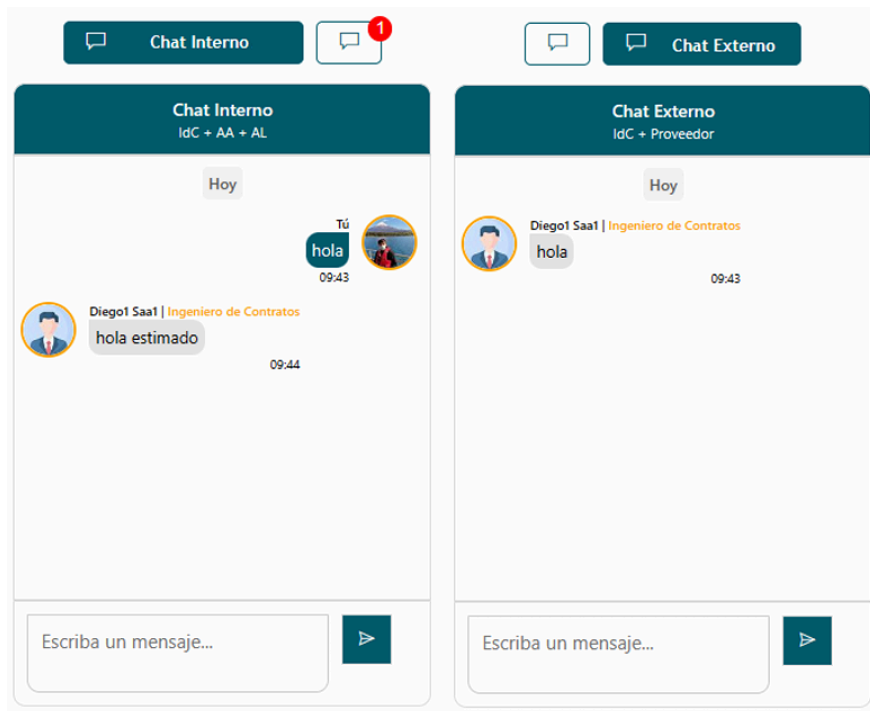


Figura 39: Chat Interno y Externo.
Fuente: Elaboración propia.

Dado que no se necesita un chat complejo, solo se permite el envío de mensajes de texto. Además, tanto los chats, como los mensajes se almacenan en listas de SharePoint Online.

4.4 Máquina Virtual

Los proveedores y sus cuentas no forman parte de la empresa minera, por lo que no pueden acceder directamente a la aplicación web alojada en SharePoint Online. Por esta razón, se proporciona una máquina virtual con Windows Server para que los proveedores puedan acceder de manera regulada a la información almacenada en SOFI.

Con el fin de garantizar la seguridad de la información contractual, se ha establecido un procedimiento exclusivo para que el proveedor acceda a través de una invitación del Ingeniero de Contratos (*figura 40*). Esta invitación incluye un enlace único y un *token* personalizado que posibilita el acceso al entorno colaborativo desde una máquina virtual segura. El *token* asignado a cada individuo funciona como el método de identificación del proveedor, asegurando su uso personal y único. Con este *token*, el proveedor tiene la capacidad de participar en conversaciones en los chats, donde puede discutir con el Ingeniero de Contratos sobre el contenido de las distintas secciones del contrato. Es esencial destacar que este *token* se emplea de manera adicional en cada mensaje o intercambio de información, fortaleciendo aún más la seguridad del proceso.

A pesar de que la aplicación externa reside en una máquina virtual, comparte similitudes en la interfaz de usuario y diseño con la aplicación alojada en SharePoint Online, además, debido al uso de Microsoft Power Automate como intermediario y a las exhaustivas validaciones de seguridad implementadas, se experimenta una latencia en el intercambio de información. Esta consideración condujo a la implementación de restricciones específicas en las funcionalidades del proveedor, permitiéndole solo el chat y visualizar el contrato, en lugar de editar directamente.

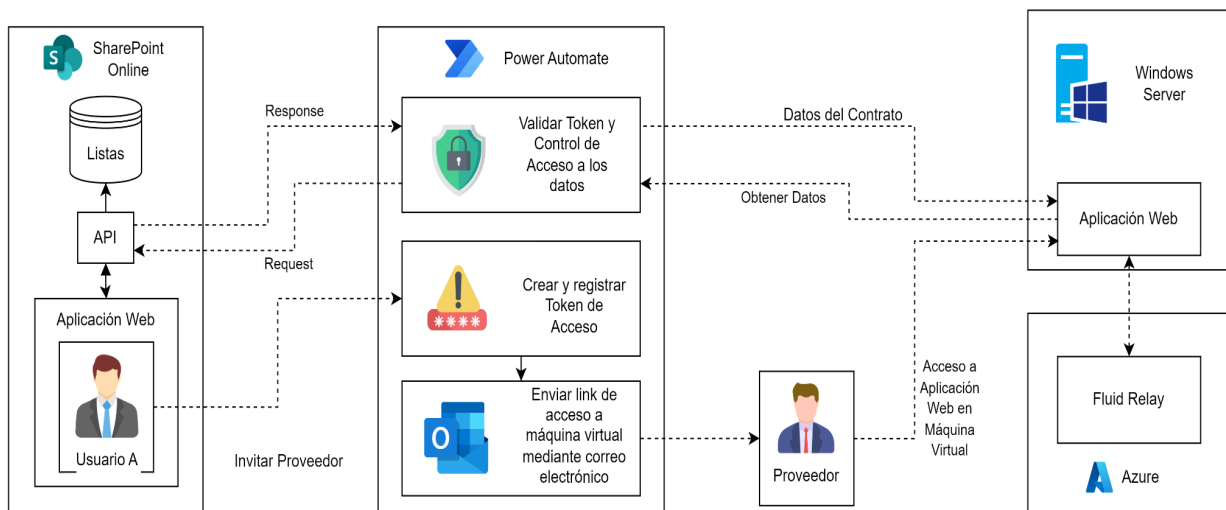


Figura 40: Lógica de invitación y control de acceso al proveedor

Fuente: Elaboración propia.

4.5 Implementar Componentes UI en Envision Framework

Dentro del ámbito de los proyectos de Envision, la necesidad de un entorno colaborativo no es un requerimiento común, pero ha surgido en algunas instancias.

Para los desarrolladores, abordar este tipo de requisitos suele ser un desafío, por lo que se sugiere incorporar en el EFW (Envision FrameWork) una biblioteca de elementos de interfaz de usuario codificados con el fin de utilizarlos en proyectos futuros, facilitando significativamente el proceso de desarrollo.

4.5.1 Conexión a Azure Fluid Relay

A modo de facilitar la implementación del servicio Azure Fluid Relay, se introduce en el EFW un *script* que trae las funcionalidades más importantes del *fluid framework* para que los desarrolladores puedan hacer uso de ellas de forma sencilla. Dentro de las funcionalidades se encuentran:

- Conexión a Azure Fluid Relay
- Creación y eliminación de Contenedores
- Creación de Schemas: Esquema con las estructuras
- Manejo de Eventos y Audiencia: Usuarios conectados en tiempo real al contenedor, detectando conexiones y desconexiones.

4.5.2 Chat

Para la mejora de SOFI, se estableció el requisito de implementar mensajes de texto únicamente. Sin embargo, al considerar la integración con el Envision Framework y su implementación en futuros proyectos, se reconoce la necesidad de desarrollar un sistema de chat más sofisticado. Este sistema debe ser capaz de manejar diversos tipos de mensajes, incluyendo texto, imágenes y documentos comunes en el ámbito empresarial, como archivos Word, Excel y PDF.

Con este objetivo en mente, se propone la creación de una clase TypeScript que permita la generación automática de un chat al ser importada en el código. Esta clase deberá ser versátil y adaptable, capaz de ajustarse a diferentes contextos y requisitos específicos de distintos proyectos.

Algunas características esperadas de esta clase podrían incluir:

- Gestión de mensajes de texto, imágenes y archivos.
- Soporte para la carga y visualización de documentos Word, Excel y PDF.
- Funcionalidades adicionales como la posibilidad de responder a mensajes específicos, reaccionar a mensajes y uso de emojis.

4.5.3 Documentación

1. Instalación

Envision utiliza una plantilla de código generada con componentes UI del Envision Framework para iniciar sus proyectos. Una vez que los desarrolladores han clonado el repositorio y han iniciado sesión en NPM, pueden ejecutar el comando **npm install**. Este comando instala todas las dependencias especificadas en el archivo **package.json**, incluyendo el paquete **@enviongroup/envision-framework**, que contiene los componentes necesarios para renderizar el chat de manera efectiva.

Es importante tener en cuenta que el archivo **package.json** incluirá la dependencia necesaria para instalar el Fluid Framework, una colección de bibliotecas de TypeScript que permiten a los usuarios operar sobre diversas estructuras de datos compartidos instanciadas en el servicio Azure Fluid Relay. En caso de que no esté presente, se puede instalar fácilmente con el comando **npm install fluid-framework**.

2. Importar

Para instanciar el componente, el primer paso es importar sus módulos correspondientes (código [Tabla 4](#)). Primero se debe importar el componente del chat desarrollado como parte de esta memoria, y el servicio de Azure Fluid Relay, el cual fue generado en base a la documentación oficial del **fluid-framework**.

Tabla 4: Importar módulos para EFWChat.
Fuente: Elaboración propia.

Código TypeScript
<pre>// Importar módulo EFWChat desde su ruta relativa import { EFWChat } from './ruta_relativa_al_EFWChat'; // Importar servicio AzureFluidRelayService desde su ruta relativa import { AzureFluidRelayService } from './ruta_relativa_al_AzureFluidRelayService'; // Importar SharedMap desde fluid-framework import { SharedMap } from 'fluid-framework';</pre>

3. Instancia del Usuario

Para acceder al servicio Fluid Relay y crear una instancia del componente de chat, es esencial definir un objeto que permita identificar al usuario que se está conectando al servicio e incluirlo en la audiencia del contenedor junto al resto de usuarios.

Las propiedades del usuario se detallan en el código de la [Tabla 5](#):

Tabla 5: Instanciar objeto usuario para la conexión.
Fuente: Elaboración propia.

Código TypeScript
<pre>const user:User = { id: 'correo@dominio.cl', name: 'Diego', additionalDetails: { id: 1, email: 'correo@dominio.cl', fullname: 'Diego Saa Bruna', rol: { id: 1, title: 'Desarrollador', color: '#a90056' }, src: spo.getUserAvatar('correo@dominio.cl'), timestamp: new Date(), }, };</pre>

4. Instanciar y Conectarse a Azure Fluid Relay

Antes de conectar con el servicio de Azure Fluid Relay, se requiere disponer de las credenciales proporcionadas por dicho servicio, las cuales pueden obtenerse desde el Portal Azure. Posteriormente, es esencial definir un esquema (*Schema*) que describa las estructuras con estado compartido en el contenedor (código de la [Tabla 5](#)).

Tabla 6: Definir el esquema con las estructuras de estado compartido en el contenedor.
Fuente: Elaboración propia.

Código TypeScript
<pre>// Definir el esquema del contenedor en Fluid Relay Service const Schema:ContainerSchema = { initialObjects: { sharedMap: SharedMap , }, };</pre>

Finalmente, se debe proporcionar un *containerId*, una cadena de texto que actúa como identificador del contenedor. Si este parámetro no se especifica y la propiedad *isNew* es true, se creará un nuevo contenedor basado en el *Schema* indicado (código de la [Tabla 7](#)).

Tabla 7: Conectarse al servicio Azure Fluid Relay.

Fuente: Elaboración propia.

Código TypeScript
<pre>/** * Si no se le pasa un containerId entonces crea uno, sin embargo deberá indicarse el último * parámetro como true */ const AZFRService = new AzureFluidRelayService({ containerId: containerId, credentials: config[process.env.DEV_ENV].accessKeyFluidRelay, user: user, schema: FluidSchema, isNew: false }); const response = await AZFRService.methods.connect();</pre>

5. Instancia del Componente de Chat

Para instanciar el componente de chat (código de la [Tabla 8](#)), simplemente se pasan como parámetro un objeto cuyas propiedades son acorde a las especificaciones del proyecto.

Tabla 8: Instanciar EFWChat.

Fuente: Elaboración propia.

Código TypeScript
<pre>new EFWChat(params);</pre>

Las propiedades del objeto params se definirán a continuación en el punto 6.

6. Parámetros

Dentro del conjunto de parámetros del EFWChat se pueden pasar estilos CSS entre otras propiedades que ayudan al chat a ser más dinámico y responsivo.

Adicionalmente, se pueden pasar funciones que se ejecutarán automáticamente al realizar ciertas acciones sobre el chat, como enviar mensajes de texto o archivos.

Tabla 9: Parámetros del EFWChat.
Fuente: Elaboración propia.

Parámetro	Tipo			Descripción
allows	Object	allowDocuments	Boolean	Define si el chat permite Imágenes, Reacciones, Emojis, Documentos.
		allowEmojis		
		allowImages		
		allowReactions		
append	Boolean	true	Agrega el chat al final del contenedor	
		false	Sobreescribe el contenedor dejando como único nodo hijo al chat	
container	HTMLElement			Elemento HTML que contendrá al chat
currentUser	User			Usuario que se conecta al chat
description	String			Descripción del Chat
logic	Object	onDeleteMessage	Function(Message):void	Se ejecuta al eliminar un mensaje
		onEditMessage	Function(Message,Action):void	Se ejecuta cuando se responde o reacciona a un mensaje
		onReceiveMessage	Function(Message):void	Se ejecuta cuando se recibe un mensaje
		onSendMessage	Function(Message):void	Se ejecuta cuando se envía un mensaje
		onUploadFile	Function(File):string	Se ejecuta al enviar un archivo
messages	Message[]			Lista de Mensajes con los que se inicia el Chat
sharedMap	SharedMap			Objeto compartido del Fluid Framework que permite la comunicación en tiempo real
style	Object	backgroundColor	string	Define los estilos del chat

		border	String	
		borderRadius	String	
		color	String	
		height	String	
		margin	String	
		padding	String	
		width	String	
title	String			Título del Chat

4.5.4 Funcionalidades e Interfaz de Usuarios

4.5.4.1 Múltiples Usuarios Conectados

El EFWChat está diseñado para que dos o más usuarios puedan comunicarse sin inconvenientes. El componente no limita al desarrollador al manejar la audiencia.

4.5.4.2 Revisar perfil de los Usuarios

El chat permite revisar el perfil de cada usuario, indicando el avatar, nombre, rol y correo electrónico.

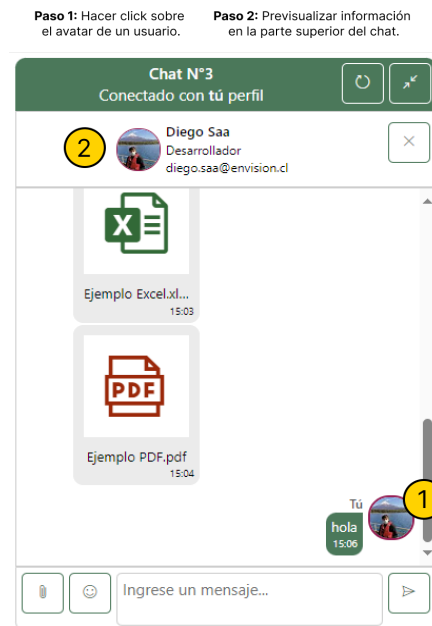


Figura 41: Ver perfil del usuario.

Fuente: Elaboración propia.

4.5.4.3 Compartir Archivos

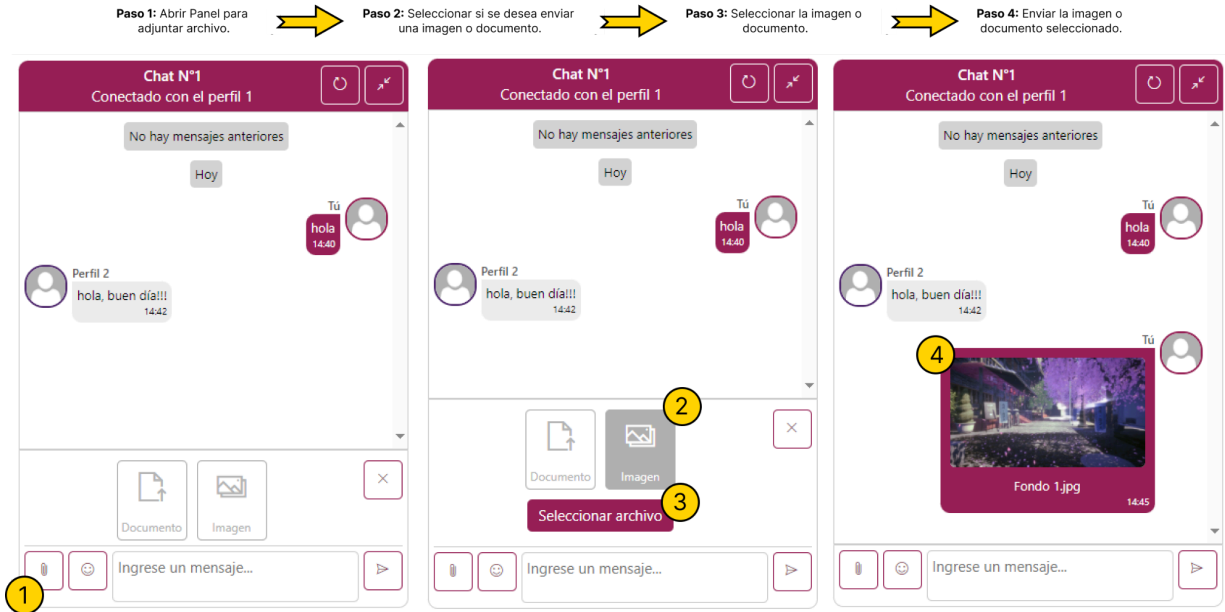


Figura 42: Compartir archivos.
Fuente: Elaboración propia.

Dentro de los documentos enviados se puede distinguir su formato (Excel, PDF, Word, etc), así como la opción de pre visualizarlo o descargarlo.

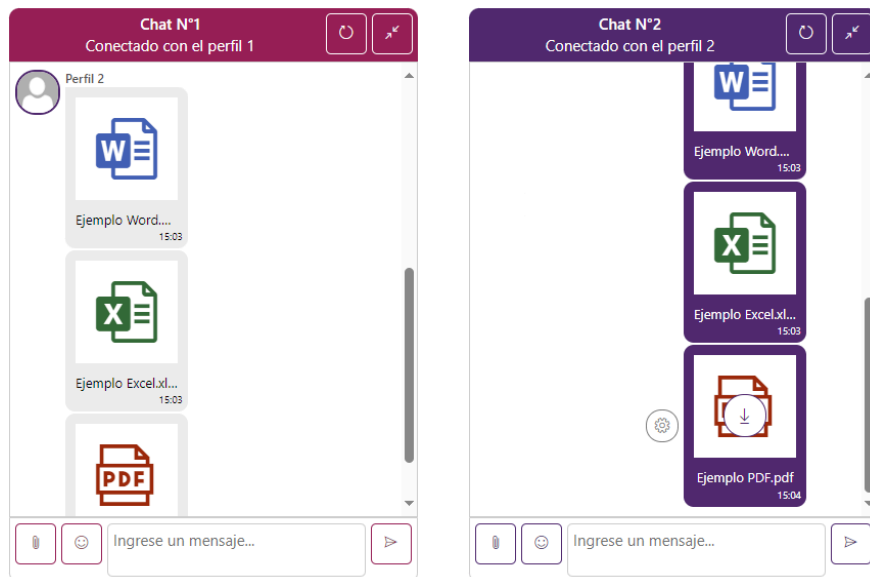
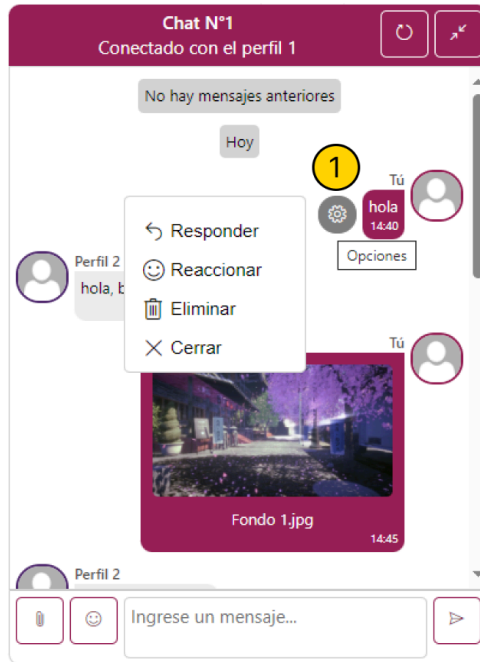


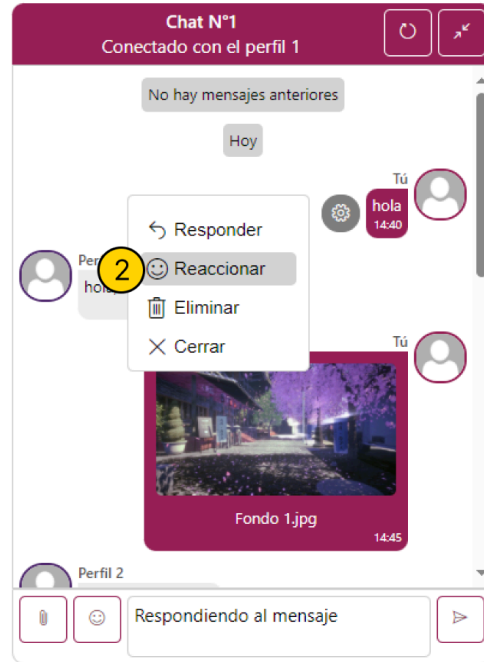
Figura 43: Documentos enviados.
Fuente: Elaboración propia.

4.5.4.4 Reaccionar a mensajes

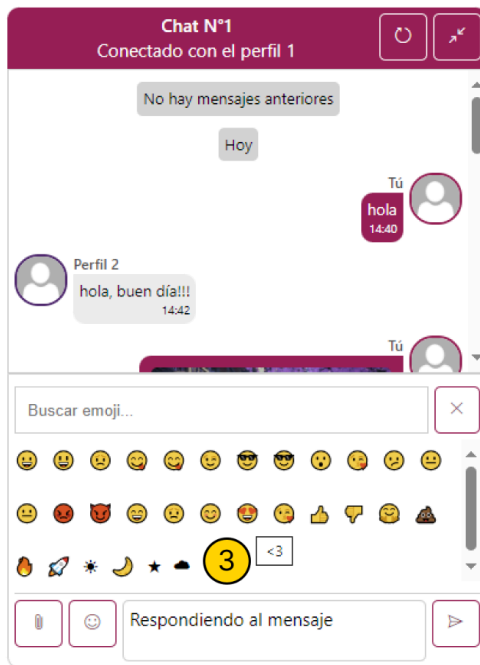
Paso 1: Al pasar el mouse sobre un mensaje se mostrará un botón que al clickearlo listará distintas opciones.



Paso 2: Seleccionar la opción Reaccionar.



Paso 3: Se desplegará un panel con Emojis para que puedas escoger con cual incluir en tu reacción al mensaje.



Paso 4: El Emoji seleccionado se mostrará en la parte inferior del mensaje.

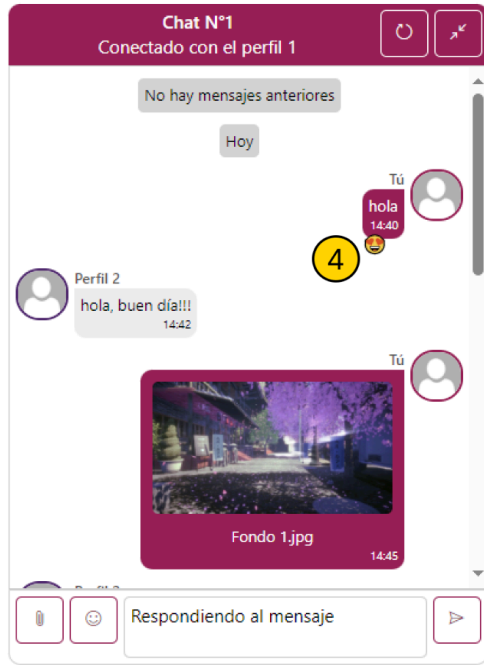
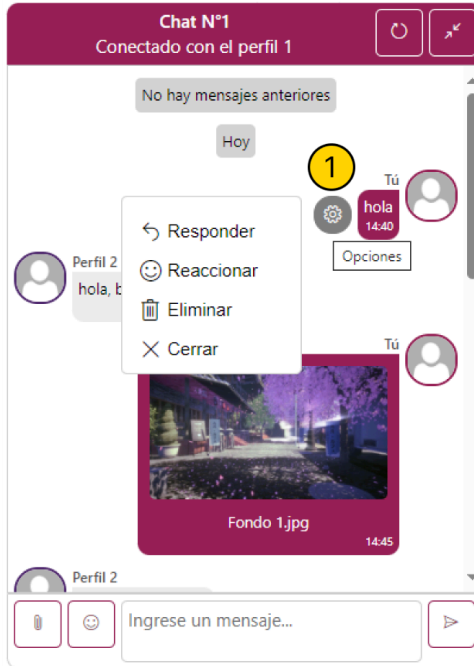


Figura 44: Reaccionar a un mensaje.

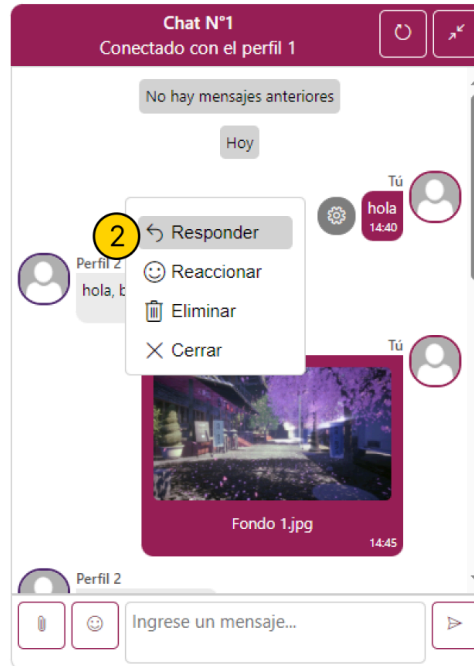
Fuente: Elaboración propia.

4.5.4.5 Responder Mensajes

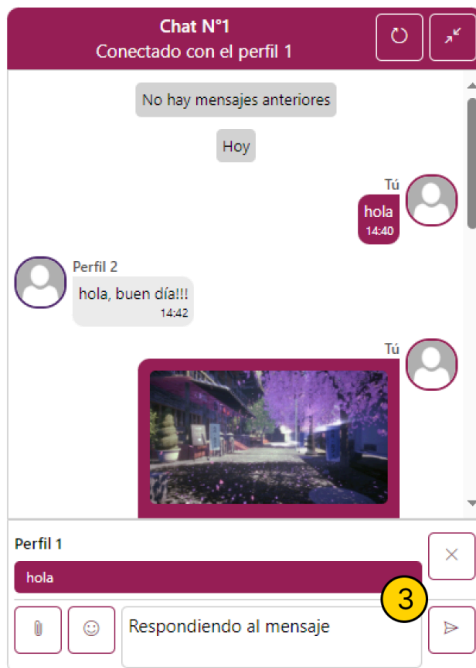
Paso 1: Al pasar el mouse sobre un mensaje se mostrará un botón que al clickearlo listará distintas opciones.



Paso 2: Seleccionar la opción Responder.



Paso 3: Se desplegará un panel con el mensaje a responder.



Paso 4: Se muestra el mensaje respondido.

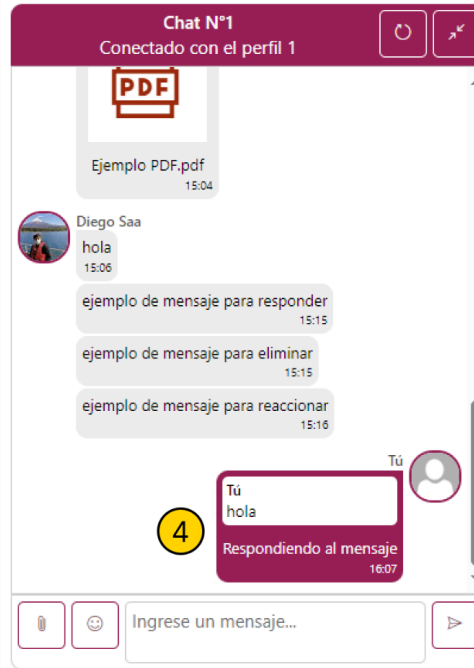


Figura 45: Responder un mensaje.

Fuente: Elaboración propia.

4.5.4.6 Eliminar mensajes

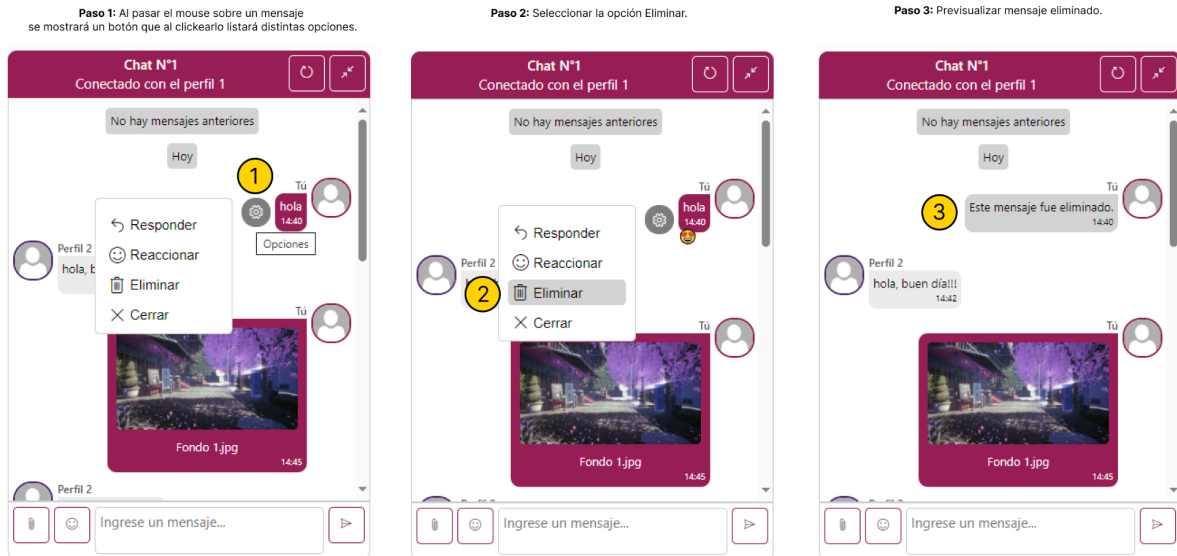


Figura 46: Eliminar un mensaje.

Fuente: Elaboración propia.

Como se puede apreciar en las figuras 41 a 46, el chat cuenta con muchas más funcionalidades que las utilizadas en el proyecto siendo a su vez bastante más completo.

El diseño está pensado para que incluso desarrolladores con poca experiencia en *fluid framework*, puedan implementarlo en minutos. Adicionalmente, el código contiene bastantes comentarios que permiten a cualquier otro desarrollador comprender su lógica y funcionamiento, para así corregir posibles errores, implementar mejoras o extender sus funcionalidades.

CAPÍTULO 5: VALIDACIÓN DE LA SOLUCIÓN

5.1 Feedback QA

Durante el desarrollo de la mejora se hicieron periódicamente despliegues de la aplicación en un ambiente de pruebas o QA (*Quality Assurance*), donde los usuarios podían ir probando las nuevas funcionalidades. Luego de cada prueba, el cliente hacía llegar un reporte con hallazgos y observaciones, donde lo que más se detectó era problemas al reemplazar variables en el contenido de la sección. Debido a la variedad de caracteres que generaba el editor de texto enriquecido fue complejo elaborar una serie de expresiones regulares que permitan capturar estas variables y realizar correctamente su reemplazo, sobre todo si estas variables representan condiciones (IF - END-IF) o bucles (FOR - END-FOR). En este caso se propone hacer es un script que haga un preprocesamiento al contenido de las secciones limpiando el texto y eliminando caracteres no deseados que son ingresados generalmente por el editor de texto utilizado para dar formato o estilos al texto.

5.2 Descripción del Proceso de Validación

Para validar la usabilidad se desarrolla un documento a modo de guía, dicho documento se encuentra en Anexos y divide el proceso de validación en 5 etapas:

5.2.1 Invitación y Consentimiento Informado

Se seleccionan 5 usuarios para participar en la evaluación del rol de Ingeniero de Contratos, el cual experimenta importantes mejoras en sus funcionalidades. Cada uno de los usuarios ha recibido una invitación vía correo electrónico, acompañada de un consentimiento informado detallado que se encuentra también adjunto en los Anexos. La invitación proporciona una visión general de la prueba, explicando en detalle los aspectos que se analizarán para mejorar la experiencia de usuario y la usabilidad del sistema.

Dentro de los 5 usuarios se escogieron algunos que ya tenían conocimiento de SOFI y habían hecho uso de ella con frecuencia, sin embargo, también se incluyeron usuarios que eran nuevos en el sistema con la finalidad de evaluar qué tan amigable es con nuevos usuarios. ¿Por qué 5 usuarios y no más? Según la regla de [Jakob Nielsen, s.f] con solo 5 usuarios es posible descubrir el 85% de los problemas de usabilidad, si bien otros expertos en el tema sugieren que más usuarios son adecuados para un análisis más profundo, consideramos que 5 es suficiente para la complejidad de SOFI, con más de 5 usuarios los hallazgos comenzarían a repetirse.

Por último, dada la naturaleza del proyecto, las pruebas han de realizarse de manera online por medio de Microsoft Teams.

5.2.2 Entrevista Inicial

Antes de iniciar la prueba de usabilidad, se lleva a cabo una entrevista inicial para recopilar información relevante sobre el perfil del usuario, incluyendo edad, profesión, nivel de familiaridad con la tecnología, entre otros aspectos relevantes.

Además, se incorpora una pregunta destinada a conocer el estado de ánimo del usuario antes de comenzar la prueba. El propósito es asegurar que la experiencia de la prueba o el uso del sistema no genere frustración para los usuarios. La respuesta a esta pregunta inicial será comparada con la respuesta obtenida al realizar la misma pregunta durante la entrevista final, una vez que el usuario haya completado la prueba y haya experimentado el uso del sistema. Este análisis permitirá evaluar cómo el sistema impacta en el estado emocional del usuario a lo largo de la interacción.

Es importante destacar que, dependiendo de las respuestas proporcionadas por el usuario, es posible que se incluyan preguntas adicionales que no forman parte del guión original de la entrevista.

5.2.3 Prueba de Usabilidad

Para las pruebas de usabilidad se escogieron un total de 9 tareas (ver [Anexos](#)) relacionadas a las distintas funcionalidades incorporadas en SOFI.

Adicionalmente, se emplea el método *Think Aloud* o “Pensar en voz alta”, donde previo a la prueba se le pide a los usuarios que puedan expresar en voz alta todos los pensamientos, impresiones, sensaciones, consideraciones, etc, lo que permite obtener una gran cantidad de información durante la prueba.

Antes de llevar a cabo la validación de este proceso, se lleva a cabo una prueba piloto con el fin de validar el flujo y asegurar la correcta redacción e interpretación de las distintas tareas a realizar. Tras esta prueba piloto, es posible que el guión original experimente ajustes o modificaciones en función de los hallazgos y lecciones aprendidas durante la prueba.

5.2.4 Formulario *System Usability Scale* (SUS)

El formulario SUS es un método muy conocido que permite evaluar la usabilidad, eficacia, eficiencia y satisfacción de un sistema.

Consta de 10 breves enunciados ya predefinidos que el usuario deberá calificar utilizando la escala de Likert, es decir utilizando 5 opciones:

Tabla 10: Escala de Likert.
Fuente: Elaboración propia.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutro	De acuerdo	Totalmente de acuerdo

Adicionalmente, los enunciados de este formulario, tienen una particularidad, cinco de ellos son positivos (1, 3, 5, 7, 9) y los otros cinco son negativos (2, 4, 6, 8, 10).

Donde el resultado del SUS viene dado por:

- I. Sumar las respuestas de los enunciados impares y restarle 5.
- II. A 25, se le resta la suma de las respuestas de los enunciados pares.
- III. Sumar ambos resultados y multiplicarlos por 2,5.

Tras un estudio realizado a 500 webs y aplicaciones por Jeff Sauro, se concluye que el puntaje promedio es de 68. Y que un resultado por debajo de esta cifra indica que hay varios aspectos a corregir.⁹

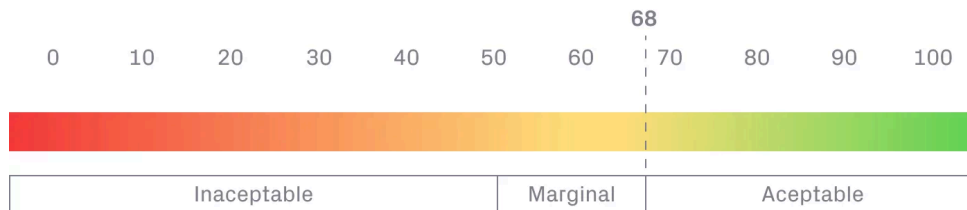


Figura 47: Representación de los resultados de un SUS.
Fuente: ui-from-mars—.

5.2.5 Entrevista Final

Después de completado el formulario SUS, se lleva a cabo una entrevista final para recopilar información sobre la experiencia del usuario al utilizar la aplicación. Durante esta entrevista, interesa conocer el estado emocional final del usuario y compararlo con su estado durante la entrevista inicial, recibir retroalimentación, opiniones y sugerencias de mejora sobre la aplicación.

5.3 Respuestas y Resultados

En esta sección se analizan los resultados obtenidos del proceso de validación, se detalla el perfil de los usuarios entrevistados.

⁹ Cómo medir la usabilidad con un SUS — uiFromMars

5.3.1 Entrevistas

Dentro de los usuarios, todos estaban dentro del Área de Ingeniería. Además, su nivel usando la tecnología era variado, así como su conocimiento sobre SOFI.

Tabla 11: Perfil de los Usuarios.
Fuente: Elaboración propia.

Cargo	Profesión	Edad	Estado de Ánimo Inicial	Estado de Ánimo Final	Nivel de manejo de tecnología	Conocimiento de SOFI
Consultor y Desarrollador	Ingeniería Civil en Informática	25	Bien	Medio	Alto	Nada
Encargado de Soporte	Estudiante Ingeniería Informática	22	Excelente	Excelente	Alto	Alto
Administración y Recursos Humanos	Ingeniería Comercial	30	Bien	Excelente	Bajo	Nada
Analista de Control de Gestión	Ingeniería Industrial	31	Bien	Bien	Medio	Bajo
Consultor y Desarrollador	Ingeniería Civil en Informática	23	Promedio	Promedio	Alto	Medio

Durante la entrevista, surgieron respuestas interesantes de dos usuarios al ser cuestionados sobre si consideraban que la aplicación mejoraba la confiabilidad y transparencia del proceso de redacción de contratos. Uno de ellos expresó: "Sí, siempre y cuando yo tenga la última palabra". Este comentario resalta una preocupación común entre los usuarios: el deseo de mantener el control final sobre los contratos generados.

En esta misma línea, se destacó que otorgar excesiva libertad al usuario podría comprometer la trazabilidad del contrato. Permitir que múltiples usuarios editen simultáneamente podría generar confusión sobre quién realizó qué cambios. Por esta razón, el sistema ha sido diseñado para permitir únicamente que un usuario edite una sección del contrato a la vez, bloqueando la edición para otros usuarios hasta que se complete la modificación.

También, en una de las pruebas se hizo una pregunta que no está en el guión: se consultó al encargado de soporte del sistema sobre los tipos de incidencias más frecuentes. Su respuesta reveló que muchos usuarios tenían un bajo nivel tecnológico y que algunos mostraban cierta inseguridad en cuanto al uso de firma digital generando dificultades para

acceder la interfaz donde se firma el contrato. Esta incertidumbre en el proceso levanta la necesidad de abordar estas preocupaciones del usuario en el corto plazo.

Es importante resaltar que, si bien todos los usuarios entrevistados en algún momento de su vida fueron parte de la redacción colaborativa de un documento que consideren importante, ya sea utilizando herramientas conocidas como Microsoft Word o Google Docs, sólo uno de ellos afirmó haberlo hecho para redactar un contrato. También destacó haber realizado una especie de negociación de su contenido. Indicando que la mayoría de las veces, alguien se encarga de redactarlo, y otro lo revisa y lo firma, muchas veces sin revisarlo en profundidad. Además mencionó: “Hoy en día, por ejemplo en internet o cuando se descargan aplicaciones en el celular, todos llegan y aceptan las condiciones de uso sin leerlas”. Este comportamiento refleja una tendencia generalizada hacia la aceptación pasiva de los términos y condiciones, sin una comprensión completa de las implicaciones legales y de responsabilidad que conllevan.

Por último, se les preguntó ¿Qué problemas se están abordando en esta aplicación? ¿Los aborda correctamente? Las respuestas de los usuarios se pueden apreciar en la *figura 48*, es importante considerar que la pregunta tenía una respuesta abierta, y no se les dió a los usuarios un listado de problemas sobre los que elegir.

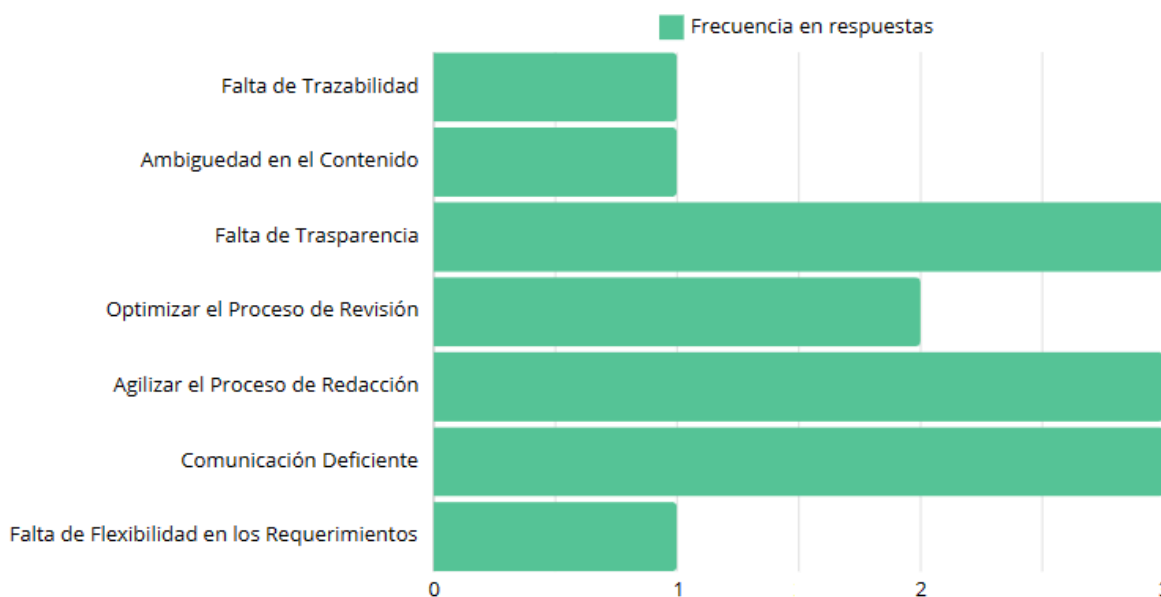


Figura 48: Problemas que aborda la aplicación según los usuarios.

Fuente: Elaboración propia.

Los usuarios indicaron que la aplicación abordaba adecuadamente estos problemas. No obstante, señalaron que algunos aspectos de la trazabilidad del contrato no eran claramente visibles en la interfaz de usuario.




A pesar de este detalle, destacaron que la aplicación mejoraba significativamente la transparencia del proceso de redacción de contratos. La capacidad de separar los diversos contenidos del contrato en secciones permitía a los proveedores dejar comentarios específicos en cada uno de ellos, lo que mejoraba la comunicación y facilitaba la comprensión mutua entre las partes interesadas.

5.3.2 Prueba de Usabilidad

Para analizar las tareas de la prueba de usabilidad se tendrá en consideración:

- Problemas de Usabilidad:
Observar las dificultades que enfrentan los usuarios al interactuar con SOFI.
- Comportamiento:
Observar estrategias de navegación, comportamientos repetitivos o situaciones que le generen confusión o frustración al usuario.
- Retroalimentación:
Analizar las opiniones, comentarios, percepciones y sugerencias menciona el usuario mientras realiza la tarea, considerando que se está empleando el método *“Think Aloud”*.

Adicionalmente, para cada error encontrado se clasificará en 3 categorías:

-  Error leve:
 - No impide que el usuario complete la tarea, sin embargo puede causar confusión o frustración.
-  Error medio:
 - Afecta a la capacidad del usuario de completar la tarea de manera eficiente y efectiva.
 - Puede requerir esfuerzo adicional por parte del usuario para comprender la interfaz.
-  Error grave:
 - Impide al usuario terminar la tarea con éxito, pudiendo generar una experiencia negativa.
 - Impide el correcto flujo de la aplicación, requiriendo intervención del facilitador o en el peor de los casos, dar por finalizada la prueba.

A continuación, se procede con el análisis de las tareas, donde de encontrarse errores se indicará su clasificación, una posible retroalimentación del usuario o los usuarios que experimentaron el error y se indicará la mejora o corrección realizada. Adicionalmente, para cada tarea se agregaron observaciones de la interacción del usuario.

➤ **Tarea 1: Modo borrador**

- [Errores]

Clasificación:  Error Leve

Descripción: Si bien el sistema le indica al usuario los campos requeridos que falta completar le dificulta encontrarlos en el formulario.

Comentarios de los Usuarios: “Acá me dice que hay datos insuficientes... debería mostrarlos en la pantalla... ¿dónde están?”

- [Observaciones]

Para esta tarea, a 2 usuarios se les dió la libertad de navegar y modificar el formulario. Al modificarlo, por regla de negocio se solicitan algunos campos extras, sin embargo, al ser tan extenso el formulario, para el usuario es complejo detectar qué campos requeridos aún no han sido completados.

- [Corrección/Mejora realizada]

Cuando el sistema muestre en una pantalla emergente el listado de campos requeridos, al hacer click en cada campo, la aplicación se desplazará hasta dicho campo.

➤ **Tarea 2: Adjudicación**

- [Observaciones]

Ningún usuario presentó dificultades al realizar esta tarea, no se observaron errores. Encontraban el botón “Continuar proceso” sin mayores inconvenientes. Y notaban la diferencia con el modo borrador.

➤ **Tarea 3: Usuarios conectados**

- [Observaciones]

Ningún usuario presentó dificultades al verificar los usuarios que se conectaban al ambiente colaborativo, encuentran inmediatamente los avatares en la barra superior e interactúan sin problemas con ellos para obtener información del usuario posicionando el puntero sobre el avatar.

➤ Tarea 4: Editor

- [Errores]
Clasificación: Error leve

Descripción: Usuario, al editar el contenido de una sección del contrato intenta destacar texto en amarillo, sin embargo no funciona intuitivamente. Usuario seleccionó el texto y usó el selector de color de la herramienta para destacar, pero el texto seleccionado no sufrió ningún cambio.

Comentarios de los Usuarios: “¿Cómo se usa esto?... no le cambió el color”

- [Observaciones]
Ningún usuario presentó problemas al Agregar, Editar (Guardar Cambios, Cancelar) y Eliminar una sección del contrato. Encontraban rápidamente los botones correspondientes y los usaban sin problemas. Sólo un usuario presentó problemas cuando utilizaba la herramienta para destacar texto en amarillo, la cuál no funcionó como se esperaba.
- [Corrección/Mejora realizada]
La causa de este error, se debía a un defecto en el código que fue corregido posterior a la prueba.

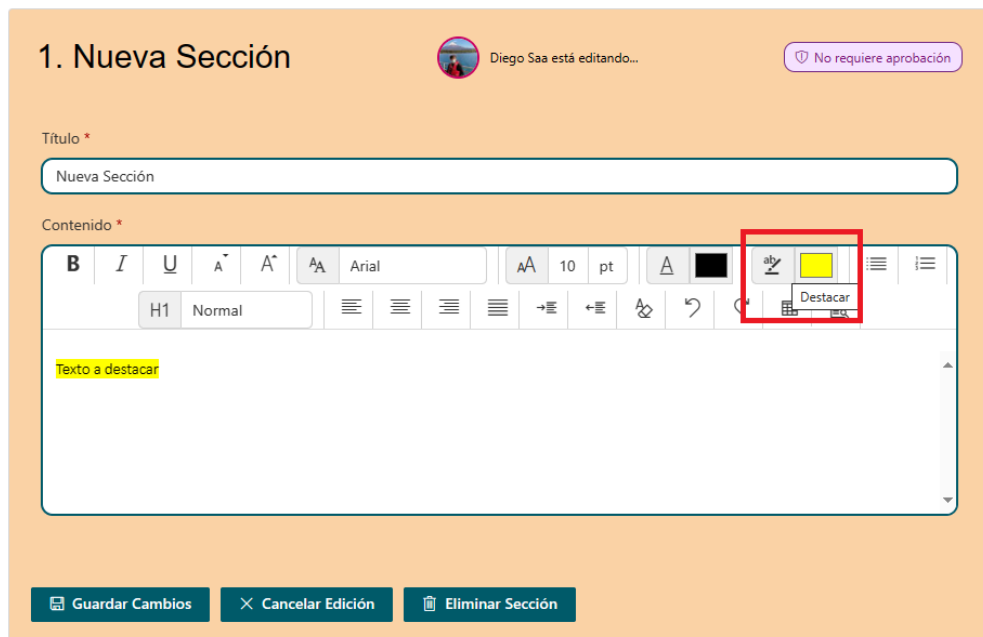


Figura 49: Corrección error en editor Tarea 4.

Fuente: Elaboración propia.

➤ Tarea 5: Visualizar Documentos

○ [Errores]

Clasificación: ■ Error medio

Descripción:

Usuarios presentan dificultad para encontrar la opción “Ver documentación” dentro del listado de funcionales desplegable al hacer clic en el botón “Opciones”. Usuarios que si presionan el botón “Opciones” tampoco hacen clic en la opción “Ver documentación”. Ningún usuario completó la tarea.

Comentarios de los Usuarios:

Cuando se les preguntó ¿por qué no presionar el botón de “Opciones”?
Usuarios respondieron:

“Me dió la impresión de que ese botón era para otras cosas”

“Por el ícono pensé que era para ajustes o configuración”

“¿Ver documentación? ¿serán manuales de uso? o la documentación del sistema?”

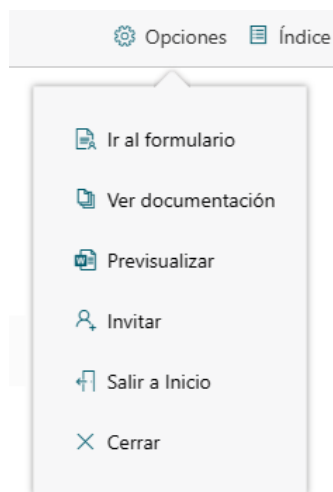


Figura 50: Listado de funcionalidades en el botón “Opciones” de Tarea 5.

Fuente: Elaboración propia.

○ [Observaciones]

El 80% de los usuarios no encuentra el botón, revisan durante tiempo suficiente el editor colaborativo y no logra completar la tarea. Se interviene la tarea y se les indica a los usuarios que el botón para “Visualizar Documentación” se encuentra visible al presionar el botón “Opciones” en

la barra superior y se aprovecha de realizar algunas preguntas. El 20% que sí presionó el botón de opciones, no hizo clic en la opción “Ver documentación”.

- [Corrección/Mejora realizada]
Se cambia ícono y título del botón a “Más Funcionalidades”. Adicionalmente, se reordenan todas las opciones del listado de mayor a menor frecuencia de uso. También se cambian algunos íconos y títulos para que sean más autodescriptivos.

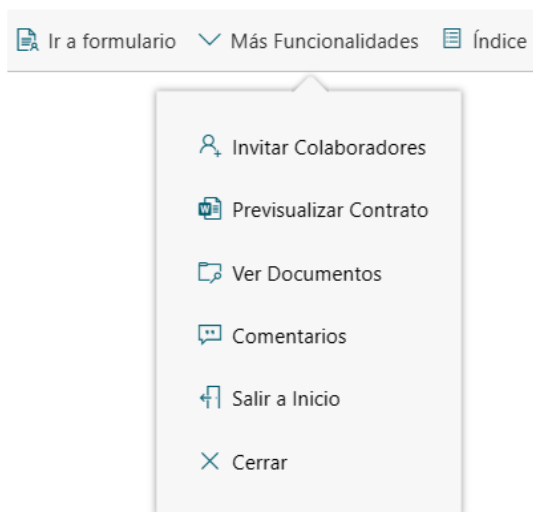


Figura 51: Listado de funcionalidades en el botón “Más Funcionalidades”.
Fuente: Elaboración propia.

➤ Tarea 6: Previsualizar Documento

- [Observaciones]
Una vez los usuarios desplegar el botón de “Opciones” encontraban sin problemas el botón para previsualizar el contrato como documento Word. Adicionalmente, se les pidió revisar el formato con el que se genera el documento y se les realizó preguntas respecto al formato del documento, sobre el cual no hubo inconvenientes.

➤ Tarea 7: Invitar a Proveedor

- [Observaciones]
Algunos usuarios presentaron dificultades para encontrar la opción “Invitar” ya que no presionaban el botón de opciones. Sin embargo, los usuarios que lo encontraban utilizaban la funcionalidad sin problemas.

- [Sugerencias de los Usuarios]
“Se podría visualizar a quienes ya se les ha enviado invitación”.
- [Corrección/Mejora realizada]
Se cambia el título a “Invitar colaboradores” y se muestran los usuarios invitados.

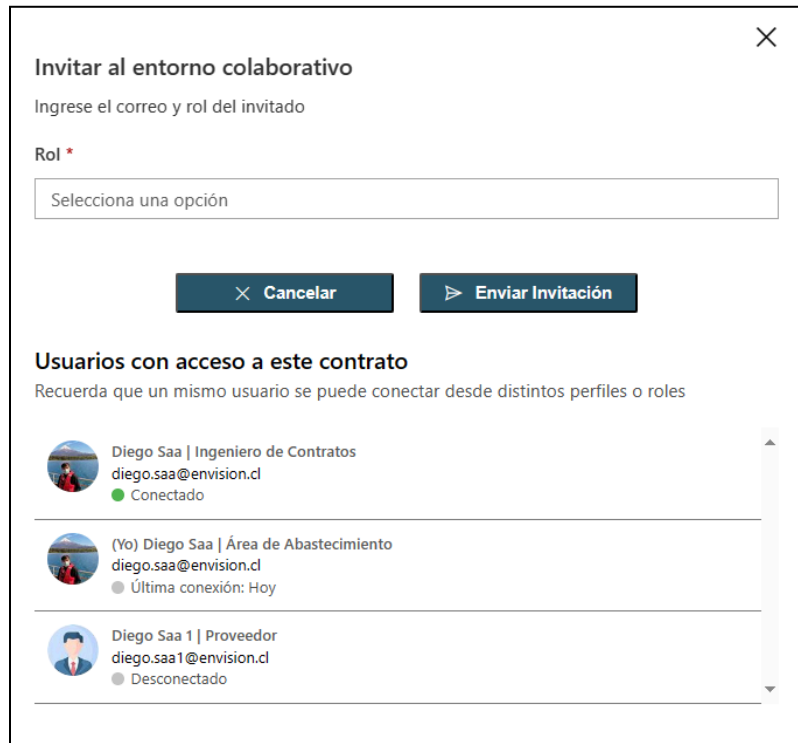


Figura 52: Popup con listado de usuarios invitados al ambiente colaborativo.
Fuente: Elaboración propia.

➤ Tarea 8: Utilizar Chats

- [Observaciones]
Todos los usuarios utilizan los distintos chats sin inconvenientes, distinguen correctamente la diferencia entre el Chat Interno y el Chat Externo con el proveedor. Adicionalmente, encuentran y envían mensajes sin problemas por los distintos canales del Chat Global.

➤ Tarea 9: Enviar a Revisión

- [Observaciones]
El usuario revisa correctamente los campos requeridos, detecta

rápidamente las secciones con fondo de color amarillo etiquetadas como “Falta Información” y modifica el texto que corresponde para completar la información faltante logrando que la sección deje de estar en amarillo, tal como se esperaba. No se observan dificultades.

Adicionalmente, en base a algunas sugerencias de mejora que realizaron los usuarios, se actualizó la interfaz de algunos paneles, para que los usuarios puedan tener una mayor trazabilidad de los cambios en el contrato.



Figura 52: Historial de Cambios en Popup Registro de Actividad.

Fuente: Elaboración propia.

Los resultados de estas pruebas de usabilidad, así como las respuestas de la entrevista y del formulario SUS, se presentan en reunión interna con el equipo de desarrollo para discutir las mejoras y correcciones que se señalaron en cada tarea.

5.3.3 Formulario SUS

A continuación, se presentan las respuestas que dieron los usuarios a los 10 enunciados del formulario SUS. Estas respuestas fueron recopiladas mediante el creador de encuestas en línea Microsoft Forms.

5.3.3.1 Respuestas

Una gran duda que surgió en la entrevista y que se vió reflejada en las respuestas del formulario SUS fue ¿qué tantas reglas de negocio debo conocer para poder utilizar el sistema? Acorde a la pregunta 4 nadie consideró necesaria la presencia de un técnico, pero el 60% de los usuarios indicó que se podría realizar una capacitación o entregar un manual de usuario.

1. Creo que me gustaría utilizar este sistema con frecuencia

● Totalmente en desacuerdo	0
● En desacuerdo	0
● Neutro	0
● De acuerdo	1
● Totalmente de acuerdo	4



6. Pensé que había demasiada inconsistencia en este sistema

● Totalmente en desacuerdo	3
● En desacuerdo	1
● Neutro	1
● De acuerdo	0
● Totalmente de acuerdo	0



2. Encontré el sistema innecesariamente complejo

● Totalmente en desacuerdo	2
● En desacuerdo	3
● Neutro	0
● De acuerdo	0
● Totalmente de acuerdo	0



7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente

● Totalmente en desacuerdo	0
● En desacuerdo	0
● Neutro	0
● De acuerdo	3
● Totalmente de acuerdo	2



3. Pensé que el sistema era fácil de usar

● Totalmente en desacuerdo	0
● En desacuerdo	1
● Neutro	1
● De acuerdo	3
● Totalmente de acuerdo	0



8. Encontré el sistema muy complicado de usar

● Totalmente en desacuerdo	2
● En desacuerdo	3
● Neutro	0
● De acuerdo	0
● Totalmente de acuerdo	0



4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema

● Totalmente en desacuerdo	2
● En desacuerdo	1
● Neutro	1
● De acuerdo	0
● Totalmente de acuerdo	1



9. Me sentí muy seguro usando el sistema

● Totalmente en desacuerdo	0
● En desacuerdo	0
● Neutro	0
● De acuerdo	2
● Totalmente de acuerdo	3



5. Encontré que las diversas funciones de este sistema estaban bien integradas

● Totalmente en desacuerdo	0
● En desacuerdo	0
● Neutro	0
● De acuerdo	2
● Totalmente de acuerdo	3



10. Necesitaba aprender muchas cosas antes de empezar con este sistema

● Totalmente en desacuerdo	1
● En desacuerdo	3
● Neutro	1
● De acuerdo	0
● Totalmente de acuerdo	0



Figura 53: Respuestas Formulario SUS.

Fuente: Microsoft Forms.

Tabla 12: Puntuaciones Formulario SUS.
Fuente: Elaboración propia.

Calificación	1	2	3	4	5	Puntaje Total Enunciados Impares	Puntaje Total Enunciados Impares	Promedio Enunciados Impares	Promedio Enunciados Pares
1				1	4	24		4.8	
2	2	3					8		1.6
3		1	1	3		17		3.4	
4	2	1	1		1		12		2.4
5				2	3	23		4.6	
6	3	1	1				8		1.6
7				3	2	22		4.4	
8	2	3					8		1.6
9				2	3	23		4.6	
10	1	3	1				10		2
Total								21.8	9.2

En la tabla anterior se detallan las respuestas obtenidas del formulario, donde cada fila representa un enunciado, en las columnas de encabezado azul su frecuencia, en las columnas con encabezado amarillo su puntuación total y en verde la puntuación promedio por usuario.

Al analizar los datos, seguimos el procedimiento indicado en la sección 5.2.4:

$$\text{Impares: } 21.8 - 5 = 16.8$$

$$\text{Pares: } 25 - 9.2 = 15.8$$

Para finalizar, el cálculo de la puntuación viene dado por:

$$(16.8 + 15.8) \times 2.5 = 81.5$$

Una puntuación de 81.5 se encuentra dentro de un rango aceptable.

CAPÍTULO 6: CONCLUSIONES

6.1 Cumplimiento de Objetivos

En primer lugar, el Objetivo General de este trabajo de título:

“Desarrollar un sistema colaborativo para la elaboración de contratos entre proveedores y empresa minera, que permita la comunicación e interacción en tiempo real, con el fin de generar transparencia y eficiencia al proceso de negociación inserto en todo contrato.”

El sistema fue concebido y desarrollado por un equipo compuesto por 3 desarrolladores, sin embargo hubo más personas involucradas en el proyecto, diseñadores, gerente de proyectos, entre otros. La mejora a SOFI consiste en dos aplicaciones web: una desplegada sobre SharePoint Online y la otra en una máquina virtual con Windows Server. Ambas aplicaciones presentan una interfaz de usuario similar, pero difieren en las funcionalidades incorporadas. Por ejemplo, la aplicación desplegada en la máquina virtual se diseñó exclusivamente para el rol de proveedores. Por lo tanto, incorporar funcionalidades adicionales resultaría innecesario, demandaría un esfuerzo adicional y podría comprometer la integridad del proceso.

Asimismo, la mejora fue especialmente diseñada para permitir que los distintos roles puedan expresar sus intereses, centrándose en estructuras modulares que hacen referencia a contenido específico de un contrato, denominadas "Secciones", con ellas se facilita la colaboración y la revisión de los aspectos relevantes, lo que permite dejar registros en el sistema cada vez que una Sección es agregada, editada, eliminada, revisada, etc. siendo el proceso completamente trazable.

En cuanto a cada uno de los objetivos específicos:

1. Implementar algoritmos de consenso que permitan una eficiente edición colaborativa de un contrato.

Implementar algoritmos de consenso fue una idea acertada para el desarrollo de este proyecto. Cuando se trabaja de manera colaborativa, existen diversos factores que pueden generar inconsistencias en el proceso, y simplemente dejar que todos los usuarios editen haría que la trazabilidad se pierda. Es por ello que fue vital implementar algoritmos de exclusión mutua, donde los usuarios conectados se comportan de manera similar a un nodo lógico en un sistema distribuido, logrando llegar a un consenso que asegure que un usuario esté editando a la vez, evitando así pérdidas de información.

2. Construir un sistema de comunicación mediante chats para debatir el contenido de las cláusulas comerciales o legales insertas en el contrato.

Como se puede apreciar en la *figura 39*, cada sección del contrato permite que los usuarios mediante chats puedan discutir su contenido en tiempo real, agilizando la negociación y toma de decisiones. Provee a los usuarios un lugar donde registrar sus discusiones y tener un respaldo de las decisiones tomadas. También promueve la colaboración, transparencia y claridad, en estos chats pueden aclarar dudas o problemas en la comprensión de alguna cláusula del contrato siendo los principales actores en la transparencia del sistema.

5. Extender el Envision Framework con el desarrollo de componentes de interfaz de usuario que puedan ser reutilizados en otros proyectos de ámbito colaborativo.

Ampliar el Envision Framework mediante el desarrollo de componentes de interfaz de usuario reutilizables representa una estrategia clave para potenciar la eficiencia en futuros proyectos de índole colaborativa.

Con este enfoque no sólo se aprovecha el tiempo invertido en la investigación de tecnologías y en la mitigación de riesgos durante el desarrollo de SOFI, sino que también capitaliza el conocimiento y la experiencia adquiridos.

Al desarrollar componentes reutilizables de interfaz de usuario que permitan una rápida integración de funcionalidades colaborativas, se promueve la agilidad en el desarrollo y la escalabilidad del sistema, permitiendo su adaptación a diversas necesidades y contextos, en este caso, se implementa en el Envision Framework componentes de usuarios que permiten implementar un sistema de mensajería mediante Chats en cosa de minutos y que haga uso de las diversas alternativas que provee el servicio Azure Fluid Relay.

6. Validar la experiencia de usuario del sistema colaborativo desarrollado, con énfasis en comprobar la transparencia y eficiencia que provea a la generación de contratos.

Después de la validación de usabilidad, se identificaron varios hallazgos. Sin embargo, gracias a la retroalimentación brindada por los usuarios, las correcciones necesarias no representaron un esfuerzo significativo y fueron aceptadas e implementadas sin contratiempos en el sistema.

Además, los comentarios de los usuarios resaltaron la eficiencia y la transparencia que la aplicación ofrecía para abordar los problemas relacionados con la generación de contratos.

Por mi parte, la realización de estas pruebas es invaluable, ya que permite detectar de manera económica y oportuna problemas técnicos y de usabilidad que a menudo pasan desapercibidos durante el proceso de desarrollo y en las pruebas de ambiente QA que suelen estar más centradas en el correcto funcionamiento de las funcionalidades que en la experiencia del usuario.

En cuanto a los resultados obtenidos del formulario SUS, el puntaje de 81.5 es considerado aceptable, lo que indica una satisfacción general con la experiencia del usuario en el sistema colaborativo desarrollado. Sin embargo, también sugiere que existen oportunidades para mejorar la experiencia del usuario aún más.

6.2 Contribuciones y Aplicaciones del Trabajo Realizado

El trabajo realizado representa un avance significativo en el proceso de redacción de contratos con proveedores para la empresa minera cliente de este proyecto. La capacidad de redactar sobre un documento digital previamente completado a partir de un formulario no solo ahorra un tiempo considerable a los Ingenieros de Contratos, sino que también registra todos los cambios realizados, lo que hace que el proceso sea más trazable y transparente.

Además, la interacción permitida entre diferentes roles durante el proceso de redacción, incluyendo a los proveedores, añade transparencia y fortalece la confianza entre las empresas.

Y, dado que las empresas en este rubro generalmente utilizan sistemas para evaluar a sus proveedores y asignarles niveles de riesgo, aspectos como la comunicación efectiva y un proceso de negociación claro son fundamentales para establecer relaciones duraderas.

Adicionalmente, si bien esta mejora añade un paso adicional al proceso, en realidad lo agiliza. Esto se debe a que, considerando el proceso de revisión llevado a cabo por los abogados del Área Legal y del Área de Abastecimiento, podrán detectar rápidamente las secciones relevantes para su revisión. Además, si se requiere algún cambio menor, podrán realizarlo directamente.

Haciendo referencia al sistema en entorno productivo, los primeros contratos generados con esta mejora han arrojado resultados prometedores. La utilización de chats para revisar el contenido de cada sección ha sido fundamental, garantizando que las partes interesadas estén al tanto del contenido del contrato y puedan contribuir de manera efectiva a su desarrollo y revisión.

Por último, pero no menos importante, el sistema no solo genera los documentos, sino que también los guarda de manera segura, convirtiéndose en un repositorio único de información facilitando la gestión, accesibilidad y búsqueda de estos documentos.

6.3 Trabajo Futuro

Una vez concluido el trabajo, se visualizan numerosas oportunidades de mejora para el sistema colaborativo.

Considerando las bases establecidas en este proyecto, se podría ampliar ciertas funcionalidades como brindar una mayor flexibilidad al editor de texto enriquecido. Existe una amplia gama de funcionalidades que los editores de texto reconocidos, como Microsoft Word ofrecen y que podrían implementarse en el editor de texto enriquecido utilizado en SOFI.

Además, aunque en sus inicios, los requerimientos solo especificaban el intercambio de mensajes de texto, es factible proponer la integración del componente de chat desarrollado como parte de este trabajo de título para expandir las capacidades del chat. Esto permitiría el envío de imágenes y archivos, así como la posibilidad de responder mensajes, eliminar mensajes, reaccionar a mensajes usando emojis entre otras funcionalidades, de manera similar a las redes sociales contemporáneas.

También, existe la posibilidad de ampliar el alcance del sistema hacia la redacción de diversos tipos de documentos, los cuales no necesariamente se limitan al contexto de los contratos con proveedores y teniendo en cuenta que el sistema está desplegado en SharePoint Online, se podría explorar su integración con otras herramientas de Microsoft, como Power BI, que facilitaría el análisis de los datos generados por el sistema y la creación de informes. Asimismo, la integración de inteligencia artificial y aprendizaje automático permitiría la automatización de tareas repetitivas.

Durante el desarrollo, se identificó una mejora, que por limitaciones de tiempo y esfuerzo, se tomó la decisión de no implementarla, sin embargo habría aportado valor a SOFI, esta mejora consistía en integrar al proceso la etapa de licitación y negociación entre los Ingenieros de Contratos y diversos proveedores que se realiza antes a la adjudicación de un contrato a un proveedor en específico, todo esto previo a la redacción del contrato.

Finalmente, dado que muchos proveedores de la industria minera tienen origen extranjero, sería pertinente implementar un soporte multilingüe y una función de traducción automática para la interfaz, lo que facilita la accesibilidad y la comprensión del sistema para usuarios de diferentes regiones y lenguas.

6.4 Comentarios Personales

Desde una perspectiva personal, considero que llevar a cabo este proyecto ha sido todo un desafío y una gran oportunidad para poner en práctica una amplia gama de conocimientos adquiridos durante mis años en la carrera Ingeniería Civil en Informática, ramos como

Sistemas Distribuidos, Lenguajes de Programación y Bases de Datos fueron de gran ayuda para concretar con éxito este trabajo.

En reiteradas ocasiones hubo que explorar y aprender nuevas tecnologías o bien profundizar en las que ya conocía. En este sentido, la elección de utilizar el servicio Azure Fluid Relay resultó ser acertada. Su implementación era sencilla, permite abstraerse completamente del uso directo de *Web Sockets*. Además, el intercambio de información con baja latencia fue de gran utilidad en muchos aspectos de este proyecto.

En algún momento se consideró cambiar algunas tecnologías, producto que la aplicación del proveedor presentaba problemas de latencia al utilizar Power Automate como intermediario para cada intercambio de información con SharePoint Online. Sin embargo, a esas alturas realizar dicho cambio no era factible, era necesario realizar una nueva propuesta de diseño, lo que podría retrasar el proyecto en semanas. Así que hubo que ingeniárselas para resolver este así como muchos otros problemas.

Por otro lado, mi experiencia como ayudante en el Laboratorio de Experiencia de Usuario en la UTFSM fue de suma importancia para el proyecto. Allí adquirí un amplio conocimiento sobre diversas metodologías y técnicas centradas en el usuario, las cuales pude aplicar a lo largo de todo el desarrollo. Es más, estas herramientas resultaron fundamentales para validar la solución, lo que agilizó las pruebas de usabilidad en gran medida. De hecho, los resultados superaron nuestras expectativas como equipo, ya que algunos aspectos que generaban incertidumbre sobre la intuitividad de ciertas funcionalidades, como el uso de texto destacado para indicar cambios, se confirmaron como altamente efectivos durante las pruebas.

Por último, la estrategia de utilizar colores fácilmente reconocibles en el diseño del sistema para representar secciones del contrato con particularidades o para indicar acciones requeridas por los usuarios demostró ser muy acertada. En particular, los usuarios respondieron positivamente a estas señales visuales, lo que facilitó su interacción con el sistema.

REFERENCIAS BIBLIOGRÁFICAS

[Tribunal de Contratación Pública, 2016] Estadística General – Tribunal de Contratación Pública. (2016). Recuperado el 13 de Noviembre, 2023, desde el sitio web:

<https://tribunaldecontratacionpublica.cl/estadistica-general/>

[Biblioteca del Congreso Nacional, 2023] Congreso, del. (2023). Biblioteca del Congreso Nacional | Ley Chile. Recuperado el 13 de Noviembre, 2023, desde el sitio web:

<https://www.bcn.cl/leychile/navegar?idNorma=213004>

[Paletta, 2009] Paletta, M. (2009). Modelo de aprendizaje para entornos distribuidos colaborativos - Archivo Digital UPM. *Oa.upm.es*. <https://oa.upm.es/3278/>

[RAE, s.f] ASALE, R., & RAE. (2022). Diccionario de la lengua española | Edición del Tricentenario. Recuperado el 13 de Noviembre de 2023, from “Diccionario de la lengua española” - Edición del Tricentenario website: <https://dle.rae.es/>

[LabUX USM, 2022] Lab UX USM. (2023), Recuperado el 13 de Noviembre de 2023, desde:

<https://labux.inf.utfsm.cl/>

[Team, 2022] Team, D. (2022, June 9). Algoritmo : Definición y usos - DataScientest. Recuperado el 13 de Noviembre de 2023, desde Formation Data Science | DataScientest.com:

<https://datascientest.com/es/que-es-un-algoritmo>

[Binance Academy, 2018] Binance Academy. (2018, December 13). ¿Qué es un Algoritmo de Consenso? Recuperado el 13 de Noviembre de 2023, desde Binance Academy:

<https://academy.binance.com/es/articles/what-is-a-blockchain-consensus-algorithm>

[Rojas *et al.*, 2005] Rojas, J., & Orellana, C. (2005). Un nuevo algoritmo distribuido de exclusión mutua que minimiza el intercambio de mensajes. *Rev. Fac. Ing. -Univ. Tarapacá*, 13(1), 89–98. Recuperado el 13 de Noviembre de 2023, desde <https://www.scielo.cl/pdf/rfacing/v13n1/art10.pdf>

[Microsoft Azure, s.f] Azure Fluid Relay: soluciones de colaboración en tiempo real | Microsoft Azure. (2023). Recuperado el 13 de Noviembre de 2023, desde Microsoft.com

<https://azure.microsoft.com/es-es/products/fluid-relay#layout-container-uid189e>

[Fluid Framework Documentation, s.f] Fluid Framework Documentation. (2023, November 11). Recuperado el 13 de Noviembre de 2023, desde Fluidframework.com website: <https://fluidframework.com/docs/>

[Academic, s.f] Exclusión mutua (informática). (2023). Recuperado el 13 de Noviembre de 2023, desde: Los diccionarios y las enciclopedias sobre el Académico website:

<https://es-academic.com/dic.nsf/eswiki/466887>

[Lamport, 1978] Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7), 558–565. <https://doi.org/10.1145/359545.359563>

[Tanenbaum, s.f] *DISTRIBUTED SYSTEMS Second Edition*. (n.d.). Recuperado desde:

https://wovi.fsinf.at/images/b/bc/TU_Wien-Verteilte_Systeme_VO_%28G%C3%B6schka%29_-_Tannenbaum-distributed-systems-principles-and-paradigms-2nd-edition.pdf

[UPV, 2017] Universidad Politécnica de Valencia UPV. (2017). El algoritmo de Berkeley | | UPV [YouTube Video]. Recuperado el 14 de Noviembre de 2023, desde:

https://www.youtube.com/watch?v=z4MQC-4bScE&t=241s&ab_channel=UniversitatPolit%C3%A8cnicadeVal%C3%A8ncia-UPV

[UPV, 2017] Universidad Politécnica de València UPV. (2017). Relojes vectoriales | | UPV [YouTube Video]. Recuperado el 14 de Noviembre de 2023, desde:

https://www.youtube.com/watch?v=kXWCzFXCdG4&t=96s&ab_channel=UniversitatPolit%C3%A8cnicadeVal%C3%A8ncia-UPV

[Dijkstra, 1968] Dijkstra, E. W. (1968). Co-operating sequential processes. In Programming languages : NATO Advanced Study Institute : lectures given at a three weeks Summer School held in Villard-le-Lans, 1966 / ed. by F. Genuys (pp. 43-112). Academic Press Inc.

[Anderson, 1990] Thomas E- Anderson (1990). The Performance of Spin Lock Alternative for Shared-Memory Multiprocessors. Recuperado desde:

<http://content.udacity-data.com.s3.amazonaws.com/courses/ud923/references/ud923-anderson-paper.pdf>

[Peterson, 1981] Peterson, G. *Volume 12, number 3 INFORMATION PROCESSING LETTERS 13 June 1981 MYTHS ABOUT THE MUTUAL EXCLUSION PROBLEM*. Recuperado desde:

<https://zoo.cs.yale.edu/classes/cs323/doc/Peterson.pdf>

[NPM, s.f] diff-match-patch. (2020, May 20). Recuperado el 14 de Noviembre de 2023, desde npm website:

<https://www.npmjs.com/package/diff-match-patch>

[NPM, s.f] diff-dom JavaScript diffing algorithm for DOM elements. Recuperado el 14 de Noviembre de 2023, desde npm website: <https://www.npmjs.com/package/diff-dom>

[Microsoft Teams, s.f] Introducción a Microsoft Teams. (2023). - Soporte técnico de Microsoft. Recuperado el 14 de Noviembre de 2023, desde Microsoft.com:

<https://support.microsoft.com/es-es/office/introducci%C3%B3n-a-microsoft-teams-b98d533f-118e-4bae-bf44-3df2470c2b12#:~:text=Microsoft%20Teams%20es%20una%20aplicaci%C3%B3n,todo%20en%20un%20mis%20lugar>

[Telesign, s.f] API de SMS para la Comunicación Global - Telesign. (2023, July 6). Recuperado el 14 de Noviembre de 2023, desde Telesign:

<https://www.telesign.com/es/productos/api-de-sms#:~:text=Una%20API%20de%20SMS%20es,sus%20propiedades%20aplicaciones%20o%20sistemas>

[Twilio, s.f] CustomerAI | Twilio.. Recuperado el 14 de Noviembre de 2023, desde:

<https://www.twilio.com/es-mx/customer-ai>

[Gusella & Zatti, 1987] Gusella, R., & Zatti, S. (1987). The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD. Recuperado el 10 de Marzo de 2024

<https://apps.dtic.mil/sti/tr/pdf/ADA619395.pdf>

[WhatsApp, s.f] WhatsApp | Mensajería y llamadas Gratuitas Privadas, Seguras Y Confiables. Recuperado el 10 de Marzo de 2023 desde https://www.whatsapp.com/?lang=es_LA

[Martin, S., Ahmed-Nacer, M., & Urso, P, 2011] Abstract unordered and ordered trees CRDT. Recuperado el 10 de Marzo de 2023 desde <https://arxiv.org/pdf/1201.1784.pdf>

ANEXOS

Tabla 13: System Usability Scale (SUS).
Fuente: Elaboración propia.

1	Creo que me gustaría utilizar este sistema frecuentemente
2	Encuentro este sistema innecesariamente complejo
3	Pienso que el sistema es fácil de usar
4	Creo que necesitaría soporte técnico para hacer uso del sistema
5	Encuentro las diversas funciones del sistema bastante bien integradas
6	He encontrado demasiada inconsistencia en el sistema
7	Creo que la mayoría de la gente aprendería a hacer uso del sistema rápidamente
8	He encontrado el sistema bastante incómodo de usar
9	Me siento muy seguro haciendo uso del sistema
10	Necesitaría aprender un montón de cosas antes de poder manejar el sistema

Invitación para Participar en Prueba de Usabilidad

El propósito del presente documento es invitarlos a participar en la prueba de usabilidad del Sistema Online de Firma (SOFI), Una plataforma web enfocada en automatizar el proceso de redacción contractual.

Para que usted pueda tomar una decisión informada, le explicaremos cuáles serán los procedimientos involucrados en la ejecución de la prueba, así como en qué consistiría su colaboración:

1. El estudio busca evaluar y probar la plataforma SOFI con técnicas de diseño centrado en usuarios que permita medir de alguna manera la empatía y usabilidad de la aplicación.
2. Como herramientas de estudio se utilizarán entrevistas y/o formularios.
3. Los resultados de la prueba de usabilidad y las respuestas de la entrevista y/o formulario serán usados en el estudio directamente y existirá una reserva total de su identidad y la información que entregue. Su nombre no será revelado, y solo tendrán acceso a los datos recolectados, el conjunto de investigadores, equipo de desarrollo y el/la profesor guía.
4. La entrevista se realizará de manera online de acuerdo a la disponibilidad del participante y de los investigadores.
5. Su participación consistirá en interactuar con una plataforma web, para ello seguirá instrucciones de un facilitador (investigador), y posteriormente en responder preguntas sobre su experiencia en el proceso, esto requerirá parte de su tiempo y conocimiento de la aplicación adquirido durante la prueba de usabilidad. Estimamos que el tiempo requerido para todo el proceso será de [30-45 minutos].
6. Los resultados y hallazgos de esta prueba serán utilizados para redactar un informe, el cual será anexado como parte de un trabajo de título una vez se dé por finalizado el proceso.
7. No se prevén riesgos asociados a su participación. Cualquier potencial pérdida de confidencialidad de los datos en este estudio será minimizada al almacenarlos en una ubicación segura.
8. Como participante, usted tiene derecho de manifestar sus dudas al investigador <diego.saa@envision.cl> en cualquier momento.
9. Tiene el derecho de retirarse del estudio en el momento en que usted lo considere necesario, sin que esto le traiga consecuencias.
10. La entrevista tiene motivos netamente estudiantiles e investigativos, es por eso por lo que nos adjudicamos el derecho de poder utilizar el material para próximos proyectos y/o exhibiciones a futuros. Solo con el fin educativo y de investigación.

EN CASO DE ACEPTAR LA INVITACIÓN PARA PARTICIPAR EN LA PRUEBA DE USABILIDAD DEBE LLENAR EL SIGUIENTE CONSENTIMIENTO INFORMADO.

Consentimiento Informado

Fecha __/__/2024

Mi nombre es _____, edad _____, trabajo en la empresa/organización _____, y me desempeño como _____.

Manifiesto mi voluntad de participar en la producción de este proyecto “Sistema Online de Firma (SOFI)”, por consecuencia acepto y autorizo que se me grabe y utilice mi imagen, declarando que Diego Saa Bruna será el dueño exclusivo del material y podrá producirlo, reproducirlos, editarlo, adaptarlo, ejecutarlo públicamente, en Chile y/o el extranjero.

Acepto los términos y condiciones para participar en la actividad a la que me han invitado, indicando que:

1. He leído lo anterior, o me lo han leído, y he entendido toda la información.
2. Cuando no entendí algo, pude preguntar, y han contestado a todas mis preguntas.
3. Sé que puedo decidir no participar, y no habrá consecuencia alguna. Si tengo alguna duda en cualquier momento de la actividad, puedo preguntar todas las veces que necesite.
4. Sé que puedo elegir participar, pero después puedo cambiar de opinión en cualquier momento sin consecuencias.
5. Sé que la información que entregue en esta actividad sólo la sabrán los integrantes del equipo de desarrollo Envision asignados al proyecto SOFI, el estudiante memorista a cargo y los profesores guía designados. Si mis respuestas llegasen a ser publicadas, no estarían relacionadas con mi nombre, así que nadie sabrá cuales fueron mis decisiones o respuestas.
6. De tener alguna pregunta sobre la actividad, después podré llamar o escribir al investigador encargado, que podrá responder todas mis preguntas y comentarios. Puedo contactarlos a través del correo <diego.saa@envision.cl>
7. Si acepto participar en la actividad debo firmar el consentimiento y me entregarán una copia si tengo cualquier duda después.

Firma del Participante

Firma del Facilitador

Validación de Usabilidad

El conjunto de metodologías implementadas en esta validación se centra en el rol de Ingeniero de Contratos, cuyas responsabilidades se verán principalmente afectadas por las mejoras implementadas en el sistema SOFI. Con esta evaluación, se busca llevar a cabo un análisis rápido y directo que permita identificar posibles problemas y mejoras tanto de usabilidad como de funcionalidad derivados de los cambios realizadas en el sistema, garantizando que el Ingeniero de Contratos pueda desempeñar el proceso de redacción contractual de manera eficiente y efectiva, aprovechando al máximo las nuevas características y funcionalidades del sistema.

1. Tareas más importantes

De las propuestas de tareas que los usuarios podrán realizar en la aplicación, hay que considerar evaluar las funcionalidades ya implementadas y que sean de interés para los stakeholders y equipo de desarrollo.

Se procede a realizar una lluvia de ideas y las tareas más importantes se presentan en la siguiente Tabla:

Tabla 14: Tareas más importantes.
Fuente: Elaboración propia.

ID	Tarea	Descripción
T1	Completar Formulario	Usuario completa formulario de creación de contratos.
T2	Edición	Usuario agrega, edita y elimina secciones del contrato.
T3	Invitar usuarios	Usuario invita a otro perfil con rol Área Legal, Área de Abastecimiento y Proveedor al editor colaborativo.
T4	Previsualizar contrato	Usuario genera documento Word del contrato.
T5	Utilizar chats	Usuario envía mensajes de texto a través de los distintos chats.
T6	Enviar a revisión	Usuario envía contrato a revisión.
T7	Visualizar estados de revisión	Usuario revisa si las secciones del contrato fueron aprobadas/rechazadas por Área Legal y/o Área de Abastecimiento

2. Intereses de los *stakeholders*

Para una prueba de usabilidad efectiva, es fundamental identificar y comprender los

intereses de los *stakeholders*.

Según lo mencionado en las distintas reuniones, se busca en las revisar los siguientes puntos:

- Aplicación funcional, que no se produzcan errores ni fallas.
- La aplicación facilita la trazabilidad del proceso contractual.
- Aplicación intuitiva y fácil de utilizar.
- Flexibilidad y capacidad de personalización del contrato.
- Eficiencia en el rendimiento de la aplicación, especialmente en los tiempos de respuesta y velocidad de carga.
- Integración efectiva de la mejora sobre SOFI actual.

3. Objetivos

3.1 Objetivo General

Evaluar la experiencia del usuario y la eficiencia del sistema de redacción de contratos colaborativos para identificar áreas de mejora y garantizar su correcto funcionamiento.

3.2 Objetivos Específicos

- Evaluar la funcionalidad del sistema y garantizar que los usuarios puedan redactar y editar contratos de manera colaborativa sin errores o fallos en el proceso.
- Determinar la intuitividad y la facilidad de uso del sistema, observando cómo los usuarios interactúan con la interfaz y si pueden completar tareas de redacción de contratos de manera fluida y sin dificultades.
- Evaluar la flexibilidad y la capacidad de personalización del sistema, permitiendo a los usuarios adaptar los contratos según sus necesidades específicas y preferencias.
- Medir la eficiencia en el rendimiento del sistema y percepción del usuario, evaluando los tiempos de respuesta y la velocidad de carga de las diferentes funcionalidades y acciones dentro del sistema.

4. Procedimiento

4.1 Invitación y Consentimiento Informado

Cada usuario recibirá una invitación formal por correo electrónico junto con un consentimiento informado detallado que describe el propósito y los detalles de la prueba. Se solicitará a los usuarios que firmen el consentimiento para confirmar su participación.

4.2 Entrevista Inicial

Antes de comenzar la entrevista y las pruebas en sí, es necesario confirmar que el usuario haya leído y firmado el consentimiento informado.

Etapas:

- Presentación
- Explicación en qué consiste la prueba, dejando en claro que lo que se busca evaluar es la aplicación y no al usuario, indicar que si lo desea, se puede retirar en cualquier momento.
- Dudas
- Preguntas
 - Perfil
 - ¿Qué edad tienes?
 - ¿A qué te dedicas?
 - ¿Cómo te sientes antes de la experiencia? Usar emojis

Tabla 15: Estados de Ánimo Entrevista Inicial.

Fuente: Elaboración propia.

				
Muy mal	Mal	Normal	Bien	Excelente

- Aplicación
 - ¿Cómo consideras tu manejo de la tecnología? (Bajo/Medio/Alto)
 - ¿Has redactado o editado de forma colaborativa algún contrato o documento que consideres importante?
 - ¿Qué consideras importante al momento de redactar un contrato?
 - ¿Cuáles serían desde tu punto de vista los problemas más graves que se pueden presentar en el contexto de celebrar o firmar un contrato con proveedores?

4.3 Prueba de Usabilidad

A continuación, se definirá en detalle cada una de las tareas que se espera que el usuario realice y que no necesariamente corresponden a las tareas indicadas en la primera sección. El sistema a utilizar, estará desplegado en ambiente de desarrollo, dentro del SharePoint Online de Envision.

Para obtener resultados óptimos durante la prueba, emplearemos la técnica "Think Aloud", que requiere que el usuario verbaliza en voz alta todas las acciones que realiza y sus pensamientos mientras interactúa con el sistema.

Adicionalmente, se le proporcionará al usuario una cuenta de correo electrónico con el rol de Ingeniero de Contratos y sus respectivas credenciales para que pueda iniciar sesión en el sistema. Por motivos de seguridad, las credenciales serán cambiadas al finalizar la prueba.

Tabla 16: Tareas Prueba de Usabilidad.
Fuente: Elaboración propia.

ID	Tarea	Escenario/Instrucciones	Criterio de Término	Requerimientos
T1	Modo Borrador	[Escenario: Formulario] "Esta aplicación te permite generar un borrador sin la necesidad de completar el formulario" T1.1: Modifica el Tipo de Solicitud en el formulario y dirígete al borrador T1.2: Explora el borrador T1.3: Vuelve al formulario	T1.1: Usuario se dirige al borrador T1.2: Dejar que el usuario revise el borrador durante 1 min T1.3: Usuario vuelve al formulario del contrato	Ninguno
T2	Adjudicación	[Escenario: Formulario] "Completa el formulario y continúa el proceso"	T2: Usuario guarda el contrato y redirige a vista de Secciones	Tener el formulario completado previamente.
T3	Usuarios Conectados	[Escenario: Editor Colaborativo] "Busca que otro usuario se encuentra conectado al editor colaborativo"	T3: Usuario encuentra los avatares en la barra superior de la pantalla	Ingresar al editor con otra cuenta para tener dos cuentas conectadas al editor.
T4	Editor	[Escenario: Editor Colaborativo] T4.1 Agrega una sección después de la portada. T4.2 Edita la sección e ingresa los datos que te proporcionaré. T4.3 Elimina la sección que creaste.	T4.1 Usuario agrega la Sección al Contrato T4.2 Usuario edita la Sección del Contrato T4.3 Usuario elimina la Sección creada.	Datos de ejemplo para el título y contenido de la sección.
T5	Visualizar Documentos	[Escenario: Editor Colaborativo] "En el formulario, se pueden adjuntar documentos como Anexos. La tarea consiste en que los busques en el sistema"	T5: Usuario encuentra los Anexos y documentos adjuntados	Documentos previamente cargados en el sistema
T6	Previsualizar	[Escenario: Editor Colaborativo]	T6: Usuario ubica y	Ninguno

	Documento	“Crea una previsualización del contrato en documento word”	presiona botón “Previsualizar” en la barra superior	
T7	Invitar a Proveedor	[Escenario: Editor Colaborativo] Invita a un proveedor	T7: Usuario envía invitación al proveedor con la cuenta señalada	Dar al usuario datos de un ejemplo de Proveedor
T8	Utilizar Chats	[Escenario: Editor Colaborativo] “Ahora que has invitado a un proveedor:” T8.1: Envíale un mensaje en la sección “Antecedentes Complementarios” T8.2: Envíale un mensaje por el chat global	T8.1: Usuario envía un mensaje por el chat correspondiente T8.2: Usuario envía un mensaje por el chat global	Ninguno
T9	Enviar a revisión	[Escenario: Editor Colaborativo] “Envía a revisión el contrato”	T9: Usuario explora los campos requeridos para enviar el formulario a revisión.	Ninguno

Análisis de cada tarea y comportamiento esperado:

Tabla 17: Análisis de Comportamiento Esperado.
Fuente: Elaboración propia.

ID	Comportamiento Esperado
T1	Usuario ingresa al modo borrador, revisa el contrato y se cuestiona la edición de las secciones del contrato.
T2	Usuario ubica el botón “Continuar proceso” y se redirecciona al editor colaborativo.
T3	Usuario encuentra panel con avatares y visualiza nombre y rol de los usuarios conectados.
T4	Usuario comprende intuitivamente cómo agregar, editar y eliminar secciones.
T5	Usuario ubica botón que redirecciona al repositorio de documentos anexados al contrato.
T6	Usuario ubica el botón para generar una previsualización del contrato en documento Word.
T7	Usuario ubica el botón para invitar a un usuario externo al editor colaborativo y lo invita correctamente.
T8	Usuario nota la diferencia entre los dos canales de chats y envía mensaje al proveedor por el canal correcto.
T9	Usuario intenta enviar contrato a revisión, sin embargo el sistema le notifica que falta información, el usuario intenta completar la información faltante.

4.4 Formulario SUS

Se le pide al usuario que complete el formulario *System Usability Scale (SUS)*, una herramienta ampliamente utilizada que permite evaluar la percepción del usuario sobre la facilidad de uso y eficacia de un sistema en particular, el formulario consta de una serie de 10 preguntas enfocadas a distintos aspectos de un producto o sistema.

4.5 Entrevista Final

Preguntas

- ¿Cómo te sientes al finalizar la experiencia? Usar emojis

Tabla 18: Estados de Ánimo Entrevista Inicial.
Fuente: Elaboración propia.

				
Muy mal	Mal	Normal	Bien	Excelente

- Aplicación
 - ¿Cuál fue la tarea más fácil? ¿Por qué?
 - ¿Cuál fue la tarea más difícil? ¿Por qué?
 - ¿Qué entiendes por "Sección"?
 - ¿Qué significan los colores de las secciones? (Naranja, Blanco, Amarillo, Gris)
 - ¿Hay algo en específico que te llamó la atención de la aplicación? ¿para bien o para mal?
 - Según tu opinión, ¿Qué problemas se están abordando en esta aplicación? ¿Los aborda correctamente?
 - ¿Crees que la aplicación hace el proceso de redacción de contratos más confiable y transparente?
 - ¿Cómo mejorarías la aplicación?

4.5 Agradecimientos y despedida

Agradecer formalmente por su tiempo, dar por finalizada la prueba.