

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SISTEMA DE GESTIÓN Y SEGUIMIENTO DE
CONVENIOS DE LA EPN CON ENTIDADES PÚBLICAS Y PRIVADAS
UTILIZANDO LA METODOLOGÍA SCRUM**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN
SOFTWARE**

LIZBETH ADRIANA SANTAMARIA HERRERA

`lizbeth.santamaria@epn.edu.ec`

LUDWING JAIR GOMEZ TIPANTIZA

`ludwing.gomez@epn.edu.ec`

DIRECTOR: RODRIGO FABIAN CHANCUSIG CHUQUILLA

`rodrigo.chancusig@epn.edu.ec`

Quito, agosto 2023

CERTIFICACIONES

Nosotros, Lizbeth Adriana Santamaría Herrera y Ludwing Jair Gomez Tipantiza declaramos que el trabajo de integración curricular aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

**LIZBETH ADRIANA SANTAMARIA
HERRERA**

LUDWING JAIR GOMEZ TIPANTIZA

Certifico que el presente trabajo de integración curricular fue desarrollado por Lizbeth Adriana Santamaría Herrera y Ludwing Jair Gomez Tipantiza, bajo mi supervisión.

**RODRIGO FABIAN CHANCUSIG CHUQUILLA
DIRECTOR**

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Lizbeth Adriana Santamaría Herrera

Ludwing Jair Gomez Tipantiza

Ing. Rodrigo Fabián Chancusig Chuquilla, MSc.

DEDICATORIA

Dedicado para mis padres, pues a ellos le debo todo lo que soy. Crecí viéndolos superar varios obstáculos difíciles, pero nunca me soltaron ni me dejaron de apoyar. Este trabajo es el resultado de lo que ellos me enseñaron, es lo mínimo que puedo hacer para retribuir todo su soporte y enseñanza.

Ludwing Jair Gómez Tipantiza

Dedico este trabajo a mis padres, que han sido un constante ejemplo de perseverancia y trabajo duro. He sido testigo de su dedicación y esfuerzo, y me han demostrado que uno mismo es el que se pone límites. Agradezco sinceramente el amor y el apoyo incondicional que siempre me han brindado, infundiendo en mí la determinación para nunca rendirme. A mis hermanas y a mi abuela materna, por su apoyo y por ser un pilar fundamental en mi vida. Me han dado cariño y siempre han estado ahí para mí, sin importar las circunstancias.

Lizbeth Adriana Santamaría Herrera

AGRADECIMIENTO

Agradezco a cada una de las personas que Dios puso en mi camino universitario, a las personas que me apoyaron y me motivaron a ingresar a la mejor universidad pública del Ecuador, a mis compañeros que hicieron este camino más cómodo, a mis amigos que siempre creyeron en mí, y para los que no también. Cada uno a su manera me supieron alimentar ya sea con palabras de aliento, o con sus críticas motivándome para demostrarme a mí mismo que si puedo, y siempre podré...

Ludwing Jair Gómez Tipantiza

Quiero expresar mi profunda gratitud a mis padres por su amor incondicional, su apoyo constante y su comprensión. Agradezco enormemente su creencia en mí y por proporcionarme las herramientas necesarias para alcanzar mis sueños.

A mi abuelita materna, agradezco su sabiduría, consejos y cariño, siendo para mí como una segunda madre y una fuente inagotable de ayuda.

A mis hermanas, valoro su amor, compañía y apoyo, ya que han sido mis mejores amigas y pilares inquebrantables.

Mi madre merece un reconocimiento especial por su amor inquebrantable, sus sacrificios y su respaldo constante. Ella ha creído en mí incondicionalmente y me ha respaldado sin reservas.

Agradezco a los MSc. Rodrigo Chancusig y Jorge Miño por su orientación, profesionalismo, apoyo y valiosos consejos, a lo largo del desarrollo del presente trabajo.

Lizbeth Adriana Santamaría Herrera

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	IX
ABSTRACT	X
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	3
1.4 Marco teórico	4
2 METODOLOGÍA.....	14
2.1 METODOLOGÍA DE DESARROLLO	15
2.2 Levantamiento de requisitos	17
2.3 Product Backlog.....	19
2.4 PLANIFICACIÓN.....	26
2.5 Sprint 0.....	26
2.6 Sprint 1.....	36
2.7 Sprint 2.....	40
2.8 Sprint 3.....	43
2.9 Sprint 4.....	46
2.10 Documentación	50
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	50
3.1 Resultados	50

3.2	Conclusiones.....	55
3.3	Recomendaciones.....	56
4	REFERENCIAS BIBLIOGRÁFICAS	57

ÍNDICE DE FIGURAS

<i>Figura 1 JSON WEB TOKEN [Autoría propia]</i>	<i>10</i>
<i>Figura 2 Esquema de arquitectura por capas [14]</i>	<i>11</i>
<i>Figura 3 Arquitectura de alto nivel [Autoría propia]</i>	<i>28</i>
<i>Figura 5 Flujo de scrum tomado de [41]</i>	<i>36</i>
<i>Figura 6 Esquema de formularios [Autoría Propia]</i>	<i>39</i>
<i>Figura 7 Esquema principal del seguimiento por convenio [Auditoría Propia].....</i>	<i>42</i>
<i>Figura 8 Formato de informe de ejecución y seguimiento [Autoría Propia].....</i>	<i>45</i>
<i>Figura 9 Proceso de automatización de informes [Autoría Propia].....</i>	<i>45</i>
<i>Figura 10 Módulo de administración donde se gestiona los usuarios [Autoría Propia]</i>	<i>49</i>
<i>Figura 11 Realización de pruebas para cada componente de cada capa presente en el proyecto [Autoría Propia]</i>	<i>51</i>
<i>Figura 12 Esquema de reportes [Autoría Propia]</i>	<i>51</i>
<i>Figura 13 Pruebas de Integracion – Frontend</i>	<i>52</i>
<i>Figura 14 Pruebas unitarias – Frontend</i>	<i>53</i>
<i>Figura 15 Burndown chart.....</i>	<i>54</i>

ÍNDICE DE TABLAS

<i>Tabla 1 Herramientas utilizadas para el desarrollo de la página web</i>	11
<i>Tabla 2 Tecnologías y estándares de desarrollo</i>	12
<i>Tabla 3 Elección de enfoques en la metodología</i>	16
<i>Tabla 4 Roles del equipo Scrum</i>	17
<i>Tabla 5 Tipos de usuario</i>	19
<i>Tabla 6 Product Backlog</i>	20
<i>Tabla 7 Planificación de ejecución de sprints</i>	26
<i>Tabla 8 Sprint Backlog 0</i>	27
<i>Tabla 9 Comparación frameworks para el frontend</i>	30
<i>Tabla 10 Comparación frameworks de acuerdo a las necesidades del proyecto</i>	32
<i>Tabla 11 Integración de Firebase con lenguajes de programación [38]</i>	33
<i>Tabla 12 Bases de datos comparadas de acuerdo con [39] y [40]</i>	34
<i>Tabla 13 Done ejecución sprint 0</i>	35
<i>Tabla 14 Sprint Backlog # 1</i>	36
<i>Tabla 15 Done sprint # 1</i>	39
<i>Tabla 16 Sprint Backlog # 2</i>	41
<i>Tabla 17 Done Sprint 2</i>	42
<i>Tabla 18. Sprint Backlog # 3</i>	43
<i>Tabla 19 Done Sprint 3</i>	44
<i>Tabla 20 Sprint Backlog # 4</i>	46
<i>Tabla 21 Done Sprint 4</i>	48
<i>Tabla 22 Tabla de puntajes SUS [43]</i>	53

RESUMEN

Actualmente, las organizaciones de instrucción superior tienen un proceso de gestión documental deficiente, ya que se realiza de forma física, lo cual conlleva desventajas como la posibilidad de perder documentos, escaso control de acceso, falta de respaldos de información y ausencia de protección contra deterioro. Estos problemas impactan la eficiencia del proceso de gestión documental llevado a cabo por los funcionarios administrativos en la organización. En la Escuela Politécnica Nacional (EPN), la gestión de documentos de convenios y acuerdos se realiza de forma manual, lo que expone a la institución a los problemas mencionados anteriormente. Para resolver esta situación, se propone el desarrollo de un sistema web con un enfoque Open-Source, a través de metodologías ágiles, que permita la gestión documental y seguimiento de los convenios de las diferentes dependencias de la EPN. Esto proporcionará un mayor control de la documentación, reducirá el tiempo de búsqueda y evitará la pérdida de documentos físicos.

PALABRAS CLAVE: Gestión documental, Convenios, Seguimiento y control, Digitalización, Sistema Web, React, Scrum, Capas.

ABSTRACT

Currently, higher education organizations have a poor document management process, as it is done physically, which entails disadvantages such as the possibility of losing documents, scarce access control, lack of information backups, and absence of protection against deterioration. These problems impact the efficiency of the document management process carried out by administrative officials in the organization. At the Escuela Politécnica Nacional (EPN), the management of documents of agreements and agreements is carried out manually, which exposes the institution to the problems mentioned above. To address this situation, the development of a web system with an open-source approach, through agile methodologies, is proposed, which allows document management and tracking of agreements from the different dependencies of the EPN. This will provide greater control over documentation, reduce search time, and prevent the loss of physical documents.

KEYWORDS: Document Management, Agreements, Monitoring and Control, Digitization, Web System, React, Scrum.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Para asegurar una información organizada y eficaz, que favorezca la productividad empresarial es necesario utilizar un sistema documental, con el cual se espera tener control, organización y documentos debidamente centralizados, que facilitara el almacenamiento y recuperación efectiva de la información. Además, las organizaciones y dependencias podrán ser más rápidas en el uso de la información, lo cual mejorará la planificación y programación de las tareas enfocadas e involucradas en el proceso que corresponde al seguimiento a los convenios, consecuentemente, se tendrá un ahorro de tiempo y un incremento de la productividad [1].

Los sistemas informáticos son vulnerables a los ataques de terceros. Estos ataques pueden comprometer la información y los recursos de la organización [2]. Por lo tanto, es fundamental tomar medidas para proteger la información.

Las medidas de seguridad de la información serán implementadas en el sistema, ya que principalmente buscan mantener la confidencialidad, integridad y disponibilidad (popularmente conocida en acrónimo como la tríada CIA por sus siglas en inglés) de los activos de la organización. En relación con la confidencialidad, se espera que únicamente individuos debidamente autorizados puedan acceder a este sistema, lo que asegura que la información pueda ser alterada bajo las apropiadas autorizaciones y exclusivamente durante los procesos predefinidos. Esta barrera no solo evita la degradación de los accesos, sino que también garantiza que la red y los datos permanezcan accesibles exclusivamente para usuarios con los permisos pertinentes.

Con la finalidad de asegurar la disponibilidad y una pronta recuperación de la información, se optará por almacenar los datos en un repositorio respaldado de manera adecuada. Este respaldo desempeña un rol fundamental en la protección de la información ante posibles daños o pérdidas, lo que, a su vez, asegura una pronta recuperación tanto en eventos desencadenados por desastres naturales como por acciones humanas [3].

La EPN al ser una organización pública del Ecuador maneja una gran cantidad de documentación. Sin embargo, la gestión de documentación de convenios y acuerdos de la EPN no está digitalizada y se lo realiza manualmente. Esto significa que los involucrados en los procesos asociados a los convenios deben archivar documentos, darles seguimiento, anotar fechas de inicio y finalización, recordar información importante relacionada con el convenio como su tipo, estado, etc. Es complicado porque los convenios duran mucho, por lo que los encargados legales podrían pasar por alto alguno de los tantos

convenios a los que deben dar seguimiento, generando resultados desfavorables con los objetivos de la EPN.

Para solucionar el problema presentado, se plantea un aplicativo web que permita la gestión y seguimiento documental de los convenios de las diferentes dependencias de la EPN, para controlar más la documentación relacionada a los convenios, para reducir el tiempo de búsqueda, evitarse perdido de documentos físicos y acceder desde cualquier navegador. Además de mejorar los procesos asociados al seguimiento de los convenios, a partir de la automatización de las tareas asociadas, como las observaciones en las que se encuentran información en lo que respecta a los avances de las actividades planificadas y efectuadas, las cuales podrán ser ocupadas para los informes que cada administrador debe cumplir con cierta periodicidad, con el fin de que la unidad o área requirente pueda evaluar los mismos, y de ser el caso, tomar los correctivos que fueren necesarios, resultando estos procesos más eficientes y productivos.

El sistema de gestión y seguimiento de convenios se desarrollará utilizando una metodología ágil. La rápida adaptabilidad a los cambios, la flexibilidad en la planificación, la máxima satisfacción del cliente, la entrega temprana y la mejora continua son los pilares de esta metodología [4].

1.1 Objetivo general

Realizar el análisis, diseño e implementación de un sistema web para la gestión y seguimiento documental de convenios de la EPN, utilizando la metodología SCRUM.

1.2 Objetivos específicos

1. Identificar las necesidades de los usuarios mediante la elicitación de requerimientos.
2. Automatizar procesos relacionados al seguimiento a los convenios y acuerdos de la EPN.
3. Definir la metodología, tecnología y arquitectura de software que se utilizará en el proyecto.
4. Diseñar y desarrollar interfaces de usuario intuitivas y accesibles.
5. Implementar medidas de seguridad como: controles de acceso, validaciones de entrada de datos, uso del principio de privilegio mínimo, configurar una gestión de

archivos adecuada [5]. De tal manera garantice la integridad, confidencialidad y disponibilidad de la información.

6. Verificar el funcionamiento del componente construido.

1.3 Alcance

El alcance de este componente es el desarrollo del frontend y backend de un sistema que permita dar seguimiento a convenios y acuerdos que se encuentren suscritos en la EPN. El Sistema tendrá un enfoque Open Source, desarrollado con metodologías ágiles. En función del marco de trabajo seleccionado (Scrum) el desarrollo del frontend y backend se realizará en 5 sprints, los mismos que contarán con entregas periódicas de software funcional.

Se contempla finalizar el proyecto con el diseño e implementación de un sistema en el cual se puedan llevar a cabo las siguientes actividades:

- Registro de convenios.
- Visualización de elementos importantes como: fecha de inicio, tipo de convenio, estado del convenio, organizaciones que realizarán el convenio, representantes legales de las organizaciones, lugar en el que están ubicadas las organizaciones, encargados de dar control y seguimiento al convenio, frecuencia con la que se informa sobre el cumplimiento del convenio, duración del convenio, fecha en la que el convenio entró en vigor, razón de finalización del convenio, firmas de los responsables, fecha prevista de finalización, versión de convenio, dependencia, e idioma del convenio.
- Visualización de un histórico en el cual se reflejen observaciones realizadas a la información del convenio.
- Realización de extensiones de los convenios mediante la incorporación de la correspondiente adenda.
- Uso de etiquetas para la búsqueda eficiente de documentos.

El proyecto comenzará con la planificación y el análisis de los requisitos del sistema. Esto implicará investigar documentos e instructivos de convenios, así como las herramientas y tecnología que se utilizarán en el desarrollo. También se efectuarán reuniones con los entes involucrados con los convenios para obtener comentarios sobre los requisitos del sistema y asegurar que el sistema satisfaga sus necesidades.

La segunda parte del proyecto involucra el desarrollo del software, que se dividirá en el frontend y el backend. Luego se llevará a cabo la instalación, las pruebas y el análisis de riesgos, con el correspondiente establecimiento de políticas de seguridad.

1.4 Marco teórico

1.4.1 Dominio

El sistema de seguimiento de convenios surge en respuesta a la necesidad de optimizar, flexibilizar y mejorar la trazabilidad de los procesos involucrados en el seguimiento de convenios. Esta iniciativa surge debido a diversos problemas que se presentan al acceder y generar información relacionada con los convenios y su respectivo seguimiento.

Se ha identificado cuatro procesos clave que engloban la gestión de los convenios:

- a) Solicitud del convenio.
- b) Suscripción del convenio.
- c) Seguimiento del convenio.
- d) Finalización del convenio.

Cada uno de los procesos conllevan un gran análisis y depende de muchos factores. Como menciona [6], los procesos pueden variar según los atributos de los convenios, su origen en áreas de investigación, académicas u otras dependencias en la EPN, y la presencia de erogación económica de la universidad.[6]

Independientemente del tipo de convenio (marco, específico, prácticas, cooperación), carta de intención o compromiso, memorando de entendimiento, adenda u otros documentos que impliquen compromisos para la EPN, el desafío real radica en representar de manera precisa los flujos de los 4 procesos anteriores, los cuales varían según las entidades participantes dentro de la EPN.

Los convenios o acuerdos se realizan entre dos o más participantes. En los cuatro procesos mencionados anteriormente, la EPN es el participante fijo y principal para todos los convenios. El otro participante dependerá de cada convenio, y puede ser una gran y diversa cantidad de personas o entidades. A estos participantes se les denominará la contraparte externa de cada convenio.

Sin embargo, es crucial resaltar que la presencia de la contraparte externa no altera el flujo del sistema. En otras palabras, el proceso de seguimiento de los convenios es el mismo

independientemente de si el convenio se realiza con una entidad pública, privada, nacional o extranjera.

Es necesario entender todo el flujo de manera global para poder entender cómo puede afectar o como afecta el proceso de seguimiento del convenio. Este análisis se puede ver reflejado en varios diagramas de flujo que se pueden encontrar en el [Anexo I](#).

1.4.2 Ejecución y seguimiento de acuerdos y convenios institucionales

En el artículo 11 de la RESOLUCIÓN ADMINISTRATIVA Nro. 0026-2019, que corresponde al instructivo de convenios de la EPN [6], se establece que la ejecución y seguimiento de convenios y acuerdos se llevara a cabo de acuerdo con la metodología planteada en el Informe Técnico de Viabilidad.

A continuación, se describen las actividades que cada ente debe llevar a cabo en el proceso de seguimiento y ejecución de convenios y acuerdos [6]:

- Parte interesada: En el caso de que exista erogación económica deberá informar detallada y documentadamente sobre la calidad del gasto.
- Dirección de Relaciones Institucionales: Da seguimiento periódico a las condiciones establecidas en el documento suscrito.
- Jefe Departamento o director:
 - Debe conocer, evaluar y controlar el cumplimiento de todos los Convenios asignados a su departamento.
 - Es su deber consolidar la información proporcionada por los administradores de los documentos suscritos.
 - Se encarga de reportar de manera semestral el estado de ejecución de los convenios vigentes. Y será el Decano quien reciba estos reportes.
- Decano: reporta la gestión de los documentos suscritos y en ejecución a través de un informe de gestión, que será enviado a las Primera Autoridad Ejecutiva.
- Administrador del convenio.
 - Una vez que se solicite un informe financiero a la Dirección Financiera, a la Unidad de Gestión de Investigación y Proyección Social (UGIPS) o a una Unidad desconcentrada de la EPN (EOD), se considerará que el proceso ha finalizado al recibir el informe solicitado.

- En caso de no haber solicitado un informe financiero, es importante considerar si se ha solicitado la asignación de recursos.
 - Si es así, se cuenta con un plazo de 30 días antes de que finalice la vigencia del documento suscrito para procesar dicha solicitud. Una vez transcurrido este período, la Dirección Financiera, UGIPS o EOD no se hará responsable de los fondos que no se hayan utilizado en la ejecución del convenio.
 - Si no es así, es necesario generar informes de seguimiento y ejecución al menos una vez al año, y reportar la gestión al Jefe de Departamento o Director. Si se actualiza el expediente del convenio, el informe lo recibirá la Primera Autoridad Ejecutiva.

1.4.3 Entendimiento del negocio y posible solución

Para dar seguimiento a los convenios dentro de la EPN, se presentó la estrategia de un historial de procesos dentro de una página web. Esta página web almacenará y mostrará toda la información relacionada a los convenios.

1.4.3.1 Página Web

Según menciona [7], una aplicación web es una aplicación que se ejecuta en un navegador y se accede a través de una conexión de red utilizando HTTP. El protocolo HTTP es un protocolo de aplicación que se utiliza para transferir información entre un servidor y un cliente. Es un protocolo rápido y eficiente que es adecuado e ideal para servicios web.

1.4.3.2 Historial de procesos

Un historial de procesos es una estrategia que permite a los usuarios de una aplicación web revisar las acciones que han realizado en el pasado, incluyendo las fechas y horas en que se realizaron. Esta información puede ser útil para monitorear los procesos y registrar las acciones realizadas en la aplicación relacionadas a un convenio.

La implementación de un historial de procesos en una aplicación web puede variar según las necesidades específicas de la aplicación. En este proyecto, se utilizarán bases de datos para registrar las acciones de los usuarios y luego mostrar estos datos en una sección específica de la aplicación.

1.4.4 Marco de trabajo SCRUM

Scrum es un marco de trabajo iterativo e incremental utilizado en la gestión de proyectos y desarrollo de productos o aplicaciones software, además se caracteriza por ser ágil, simple y genérico. Inicia con la creación de un Product Backlog y su objetivo principal es ofrecer

características funcionales, integradas, probadas y valiosas para el negocio en cada iteración, lo que permite obtener resultados rápidos y fomentar la innovación en la entrega de productos y servicios [8].

1.4.4.1 Prácticas de SCRUM

Roles

Product Owner (voz del cliente): Además de compartir el objetivo inicial del sprint, presentar el product backlog priorizado y responder a las preguntas del equipo sobre los ítems del backlog de productos de alto nivel, también debe estar disponible para realizar aclaraciones, revisar el trabajo y proporcionar retroalimentación al equipo [8].

Development Team: Determina lo que se puede ofrecer y realizar, se autoorganiza y decide la mejor manera de cumplir con el objetivo establecido en el sprint planning. Además, se compromete de manera realista al finalizar el sprint planning [8].

Scrum Master (Facilitador): Actúa como el coach del equipo, observando la actividad de planificación, formulando preguntas de sondeo y ayudando a lograr un resultado satisfactorio. Si bien el Scrum Master hace todo lo posible para que el equipo tenga éxito, no asigna tareas al equipo ni dicta cómo realizar el trabajo [8].

Actividades

Sprint: Scrum organiza el trabajo en iteraciones o ciclos de hasta un mes calendario llamados sprints. Durante cada sprint, se realiza un incremento que debe ser usable y que consiste en la suma de todos los requisitos completados al final del sprint. Para visualizar el estado del trabajo restante de la pila del sprint, se utiliza una gráfica conocida como burn down [8].

Sprint planning: Al inicio de cada sprint, que generalmente tiene una duración de 2 o 3 semanas, se lleva a cabo el sprint planning. Durante esta fase, el Product Owner y el equipo de desarrollo acuerdan el objetivo del sprint, que establece lo que se espera lograr en el próximo sprint. Luego, el equipo se encarga de preparar el Sprint Backlog, donde se detallan las tareas necesarias para alcanzar el objetivo del sprint, y se informa y comunica el trabajo potencial a realizar [8].

Daily scrum: Es una breve reunión que se realiza cada día durante el sprint, la duración es de 15 minutos o menos, preferiblemente a la misma hora y participa el equipo de desarrollo. Durante la reunión, el equipo responde a tres preguntas: "¿Qué has hecho desde ayer?", "¿Qué es lo que harás hasta la reunión de mañana?" y "¿Has tenido algún problema que te haya impedido completar tu objetivo?" [8].

Sprint review: Durante esta actividad, se fomenta la conversación entre el equipo Scrum, los interesados, los patrocinadores, los clientes y miembros interesados de otros equipos. El objetivo es revisar las características recién completadas en el contexto del desarrollo, examinar y ajustar el producto en construcción y verificar si el trabajo fue o no completado [8].

Sprint retrospective: Es la segunda actividad de inspección y adaptación que toma lugar al finalizar el sprint, de hecho, ocurre después del sprint review y antes de siguiente sprint planning. Durante la retrospectiva de sprint, el equipo de desarrollo, el ScrumMaster y el product owner se reúnen para discutir lo que está funcionando y lo que no está funcionando. Después de que se completa la retrospectiva de sprint, todo el ciclo se repite nuevamente, comenzando con el siguiente sprint planning [8].

Artefactos

Product backlog: Es una lista priorizada y ordenada de tareas, la cual es gestionada por el product owner, quien colabora con el equipo de desarrollo y los interesados para reunir y definir estos elementos. Además, incluye características, cambios, defectos y mejoras técnicas. El product owner se asegura de que los elementos estén secuenciados correctamente según factores como valor, costo, conocimiento y riesgo, colocando los elementos de alto valor en la parte superior. El product backlog es actualizado continuamente por el product owner basado en la retroalimentación y puede ser modificado a lo largo del proceso de desarrollo [8].

Sprint backlog: Es una lista de requisitos comprometidos por el equipo de desarrollo para un sprint específico. Se describe a través de un conjunto de tareas detalladas que delinean el plan del equipo para diseñar, construir, integrar y probar el subconjunto de funciones seleccionadas del product backlog durante ese sprint en particular [8].

1.4.5 Bases de Datos Relacionales

Se espera que el sistema documental maneje una gran cantidad de información, por lo que se decidió usar bases de datos relacionales, ya que ofrecen diversas opciones para asegurar un rendimiento óptimo. Además de que una de las fortalezas destacadas del equipo de desarrollo es su experiencia en la gestión de este tipo de bases. Según [9], los puntos de datos relacionados entre sí se almacenan y acceden en una base de datos relacional.

1.4.6 Interfaz de programación de aplicaciones (API)

Las API son interfaces que permiten a dos programas comunicarse entre sí y transmitir información. Las APIs utilizan los endpoints (URLs) gestionados por los controladores para exponer toda la funcionalidad del backend con el frontend. Es en estos controladores donde se almacena el acceso a toda la lógica programada.

La arquitectura de software REST (Representational State Transfer) establece pautas sobre cómo debería funcionar una API. REST ofrece una comunicación confiable y de alto rendimiento, al mismo tiempo que es flexible y fácil de implementar, lo que permite modificaciones y garantiza la portabilidad entre plataformas. Las API REST emplean peticiones HTTP para llevar a cabo las operaciones convencionales de base de datos, como la creación, lectura, actualización y eliminación de registros (conocidas como operaciones CRUD). Para obtener un registro, una API REST recurre a la petición GET; para crear, utiliza POST; para actualizar, utiliza PUT; y para eliminar, emplea DELETE. [10], [11].

Los servicios web que implementan la arquitectura REST se denominan servicios web RESTful. El término API RESTful se utiliza generalmente para referirse a las API web RESTful, las cuales han sido muy utilizadas en estos tiempos en la mayoría de los proyectos web [10].

La función básica de una API RESTful es permitir al cliente comunicarse con el servidor para acceder a recursos. Los pasos generales para hacer una llamada a una API REST son los siguientes: El cliente envía una solicitud al servidor. El servidor autentica al cliente y verifica sus derechos para realizar la solicitud. El servidor recibe y procesa la solicitud internamente. El servidor envía una respuesta al cliente, indicando si la solicitud fue exitosa y proporcionando la información solicitada. Los detalles de la solicitud y la respuesta pueden variar según el diseño de la API por parte de los desarrolladores [10].

1.4.7 SEGURIDAD

A medida que la tecnología avanza a un ritmo vertiginoso, las vulnerabilidades también lo hacen. Esto crea oportunidades para que los atacantes aprovechen las vulnerabilidades y alcancen sus objetivos. Según [12], la seguridad de los datos confidenciales ya sea información organizacional o personal, es un factor importante que preocupa en cuanto a su protección. Las API REST no son una excepción, ya que forman parte de sistemas esenciales que requieren protección contra amenazas e infracciones de seguridad que podrían suceder principalmente al usar HTTP como el protocolo de comunicación.

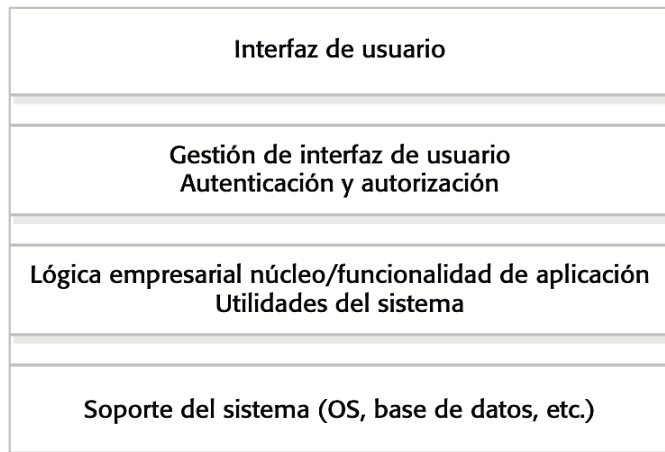


Figura 2 Esquema de arquitectura por capas [14]

Las capas dependen de los servicios que brinda la capa inferior inmediata. Los principales beneficios es que permite un desarrollo continuo y que las capas son portátiles siempre y cuando las interfaces iniciales permanezcan estables y sin cambios. Solo las capas que dependen de una interfaz entre capas se ven afectadas cuando se produce un cambio en ella. También se facilita la implementación de aplicaciones multiplataforma al tener dependencias en capas inferiores.

1.4.9 Herramientas de desarrollo

El desarrollo de aplicaciones web por lo general se dividen en el desarrollo frontend y backend. Cada una ocupa herramientas y tecnologías detalladas en la Tabla 1 y Tabla 2 respectivamente.

Tabla 1 Herramientas utilizadas para el desarrollo de la página web

Herramienta	Descripción
Web Storm	JetBrains, un proveedor de software especializado en la creación de herramientas de desarrollo inteligentes desarrolló tanto WebStorm como IntelliJ IDEA, que son dos entornos de desarrollo integrados [15].
Intellij Idea	WebStorm es un IDE de JetBrains diseñado para desarrolladores web. Es compatible con TypeScript y React, los cuales serán utilizados a la hora de realizar la parte del frontend. En lo que respecta al backend, se utiliza IntelliJ IDEA, un IDE que ofrece asistencia avanzada en la codificación, refactorización confiable y una navegación instantánea por el código.

DBeaver	Es una herramienta de gestión de bases de datos, que ofrece la posibilidad de completar el código y resaltar la sintaxis. De acuerdo con [16], una de las ventajas de DBeaver es su facilidad de uso, su capacidad para trabajar con varias bases de datos y su apoyo para el desarrollo de SQL, la administración de bases de datos y el mantenimiento.
Notion	Es una herramienta que sirve para tomar notas, planificar y gestionar proyectos. La ventaja dentro del proyecto es que permite el trabajo en equipo [17].
GitHub	Es una herramienta que permite la colaboración grupal, el alojamiento de código y el control de versiones basado en Git [18].
Postman	La plataforma permite la creación y el uso de API's, así como la colaboración en grupo [19]. Esto facilita la comunicación entre los desarrolladores de backend y frontend, lo que ayuda a garantizar que el proyecto se desarrolle de manera fluida y eficiente.
Jasper Reports	Es una herramienta para generar reportes automatizados que conecta perfectamente con el framework de Spring. Jasper tiene la ventaja de que puede generar informes en varios formatos

Tabla 2 Tecnologías y estándares de desarrollo

Tecnología	Descripción
MySQL	Considerando la naturaleza del sistema documental y requisitos como la escalabilidad, compatibilidad con la infraestructura, rendimiento y seguridad, se llegó a la conclusión de que MySQL es la opción más adecuada. Esto se debe a que, en escenarios en los que los registros de datos experimentan un aumento constante, es esencial contar con un sistema de gestión de bases de datos (SGBD) apropiado. MySQL es uno de estos SGBD, conocido por su capacidad para procesar, almacenar y mantener grandes volúmenes de datos de manera eficiente.
Next Js	Next.js representa un marco JavaScript de código abierto y ágil, desarrollado sobre la base de React. Su objetivo es simplificar la creación de aplicaciones y sitios web de alto rendimiento y usuario amigables. Next.js emplea tanto el renderizado del lado del servidor (SSR) como la generación de sitios web estáticos (SSG) para

	<p>producir contenido HTML en respuesta a solicitudes al servidor o durante el proceso de compilación, respectivamente. Esta estrategia posibilita la ejecución fluida de aplicaciones y sitios web de Next.js en diversos dispositivos, facilitando su distribución a través de múltiples canales [20].</p>
SpringBoot	<p>El Framework de SpringBoot ofrece un enfoque completo tanto en la programación como en la configuración integral para aplicaciones empresariales desarrolladas en Java. Se enfoca en manejar las preocupaciones de infraestructura, permitiendo que los desarrolladores se concentren en la lógica del negocio. Además, proporciona características esenciales como la inyección de dependencias, soporte para pruebas, acceso a datos, frameworks web, capacidades de integración y soporte para múltiples lenguajes [21].</p>
GitFlow	<p>Es un modelo de ramificación de Git que utiliza ramas de características y múltiples ramas principales. Siguiendo este modelo los desarrolladores crean una rama y posponen la fusionan con la rama principal hasta que la característica está completa. Gitflow ofrece ventajas como solicitudes de extracción, experimentos aislados y una colaboración más eficiente [22].</p>
Firebase	<p>Firebase es una plataforma desarrollada por Google para crear aplicaciones móviles y web. Ofrece servicios como: Cloud Firestore, autenticación, mensajería en la nube, funciones en la nube, Crashlytics, monitoreo de rendimiento y Google Analytics, etc [23].</p> <p>En este caso, utilizaremos Firebase Authentication, un servicio diseñado para facilitar la creación de sistemas de autenticación seguros, además, proporciona diversos métodos de autenticación, como correo electrónico y contraseña, verificación de número de teléfono, inicio de sesión mediante redes sociales y autenticación personalizada [24].</p>
Material UI	<p>Es una librería de componentes de React de código abierto que implementa el diseño de materiales de Google. Proporciona una amplia gama de componentes listos para usar para crear interfaces de usuario atractivas de forma rápida [25].</p>

2 METODOLOGÍA

El diseño del componente se basa en un enfoque mixto en cuanto a su metodología de investigación, es decir, se combina elementos cualitativos y cuantitativos, donde el primero prevalece sobre el último. Este enfoque es sumamente útil, ya que combina tanto la comprensión de las necesidades y expectativas de los usuarios (enfoque cualitativo), como la recopilación de datos mediante pruebas de usabilidad, unitarias y de integración en el frontend y el backend para evaluar el rendimiento del componente en términos de eficiencia, velocidad y estabilidad (enfoque cuantitativo).

El trabajo por realizar es experimental, ya que mediante la práctica junto a la experimentación se desarrollará y probará una aplicación web para la gestión y seguimiento de convenios de la EPN. Se llevarán a cabo actividades como el diseño de interfaces mediante mockups, configuración de base de datos, implementación de tecnologías y su posterior desarrollo.

A medida que se avance en el desarrollo del proyecto, se realizarán pruebas para evaluar la funcionalidad de la aplicación, lo que facilita la detección temprana de problemas. Al tratarse de un proyecto experimental, tendremos la oportunidad de explorar y aprender más sobre los elementos que contribuyen a la calidad de un software y cómo lograr un resultado final de alta calidad.

Las técnicas de recolección de información que se utilizarán son:

- **Análisis documental:** mediante el estudio de la documentación se podrá obtener información importante sobre convenios, herramientas, tecnologías y buenas prácticas.
- **Focus Group:** reuniones con el cliente (Administrativos de convenios) para discutir temas relacionados a la aplicación web y si los avances van de acuerdo con las necesidades del cliente.
- **Fuentes abiertas:** recursos documentales públicos que contenga información relacionada al tema de este proyecto.

Antes de comenzar a trabajar de lleno en este proyecto de software, se debe realizar un análisis que tenga en cuenta las ideas que se presentaron inicialmente, tanto el alcance como las limitaciones del proyecto y el tiempo disponible, todo esto con el fin de comenzar a construir las funcionalidades que se espera que el sistema pueda cumplir.

Para cumplir con el proyecto, es importante definir una estructura que nos guíe en el desarrollo. Esta estructura debe incluir la elección de una metodología de desarrollo de software, que nos ayudará a comprender mejor las necesidades del cliente, una arquitectura adaptada a las necesidades, las tecnologías a utilizar, y el proceso de implementación o codificación.

2.1 METODOLOGÍA DE DESARROLLO

La elección de una metodología de desarrollo de software a menudo se basa en la falta de conocimiento y comprensión de las metodologías ágiles. A medida que estas técnicas se han vuelto más populares, muchos equipos y organizaciones se han apresurado a adoptar enfoques ágiles sin una comprensión completa de lo que implican.

La falta de comprensión ha llevado a la aplicación incorrecta e incompleta de las metodologías ágiles. Muchos equipos toman superficialmente algunos aspectos de las metodologías ágiles, como las reuniones diarias o la entrega incremental, pero no abordan adecuadamente otros aspectos importantes, como la colaboración constante con la parte interesada, la retroalimentación continua o la adaptación flexible a los cambios. De tal manera que estas acciones dan cabida a interpretación de que se está ocupando un tipo de metodología tradicional adaptada, pues se pierden todos los beneficios de ser una metodología, y en lugar se convierte en una mezcla confusa de enfoques que no se ajusta completamente a ninguna metodología en particular.

Las metodologías tradicionales de desarrollo de software se basan en un flujo lineal de procesos con etapas bien definidas. Para que funcionen correctamente, es importante tener una idea clara de lo que se quiere lograr y definir los requisitos del proyecto. Las metodologías tradicionales son una buena opción para proyectos que tienen requisitos bien definidos desde el principio y que no necesitan cambios frecuentes, pero se debe tener en cuenta que pueden ser demasiado inflexibles para proyectos que cambian rápidamente.

La elección de la metodología de desarrollo de software es una decisión importante. Hay dos grandes caminos a seguir: las metodologías tradicionales y las metodologías ágiles. La metodología adecuada para el proyecto dependerá del tipo de este proyecto y de sus respectivas proyecciones, ver Tabla 3.

Tabla 3 Elección de enfoques en la metodología

Característica	Enfoque ágil	Enfoque tradicional
<p>El alcance del proyecto se centra en desarrollar un sistema que permita el seguimiento efectivo de los convenios en la EPN, facilitando la gestión, comunicación y cumplimiento de las obligaciones establecidas en dichos convenios. El proyecto durará poco y requiere menos recursos y personal.</p> <p>El tamaño de este proyecto se clasifica como mediano debido a las consideraciones anteriores, lo que favorece tener ciclos de desarrollo más cortos, que pueden ir de la mano con las iteraciones cortas que existen en enfoques ágiles.</p>	X	
<p>Debido a que el producto final debe desarrollarse e implementarse en el transcurso del semestre académico, la planificación es muy rigurosa y poco flexible.</p>		X
<p>Para la creación del sistema de seguimiento de convenios de la EPN se tiene un conjunto de necesidades las cuales se transformarán a requisitos, aun así, no se puede afirmar que estos requisitos estén bien definidos pues no está muy claro las funcionalidades y las operaciones que al final realizará el sistema.</p>	X	
<p>A pesar de que este proyecto requiere esclarecer las funcionalidades y consecuentemente los requisitos, no hay una completa disponibilidad del cliente para tratarlos en varias reuniones, debido a su agenda. La implicación del cliente es baja, lo que se refiere a una participación no constante y poco activa del cliente a lo largo del proceso de desarrollo. Esto es similar al enfoque tradicional, que permite que el cliente participe solo al principio del proyecto, proporcionando generalmente los requisitos iniciales y esperando el resultado final.</p>		X
<p>Se espera que este proyecto tenga como prioridad el producto, que debe evaluarse de manera iterativa para obtener comentarios y ajustarse a las necesidades del cliente. Por lo tanto, la documentación, aunque es importante, pasa a un segundo plano.</p>	X	

Como se pudo notar la mayoría de los puntos se decantan por un enfoque ágil, aun así, teniendo puntos que encajan en el enfoque tradicional, se espera lidiar con estos para poder integrarlos y así poder trabajar en el marco de trabajo de una metodología de enfoque ágil.

2.1.1 MÉTODOLOGÍA AGIL

Una vez escogido el agilismo como la metodología de desarrollo, se espera evaluar y ajustar la implementación para que sea coherente con las mejores prácticas y principios fundamentales, los cuales deben ser entendidos desde la colaboración, la adaptabilidad y la entrega temprana de valor, puesto que son esenciales para la implementación exitosa de una metodología ágil.

Para lograr el objetivo del presente trabajo se utilizará el marco de trabajo SCRUM, el cual promueve la colaboración y facilita la entrega incremental. Además, el equipo de desarrollo y los participantes tienen una amplia experiencia tanto académica como profesional, por lo que tienen conocimientos teóricos y prácticos que ayudan a implementar este marco de trabajo.

El análisis de la información se realizará de manera continua durante cada sprint. Esto incluye revisiones de historias de usuario, backlog y sprint, retrospectivas de sprint, y retroalimentación del cliente para verificar que sus necesidades están siendo cubiertas.

2.2 Levantamiento de requisitos

Una vez escogido SCRUM como marco de trabajo, se empieza a utilizar varias de sus herramientas y técnicas para recolectar la mayoría de información que pueda servir para esclarecer las necesidades del cliente.

Pero para ello primero se debe especificar los roles de cada miembro del equipo Scrum. Los roles de cada miembro del equipo se describen en la Tabla 4:

Tabla 4 Roles del equipo Scrum

ROLES	INDIVIDUO	DETALLES
PRODUCT OWNER	Msc. Rodrigo Chancusig	Profesor de la Escuela Politécnica Nacional, susceptible a ser administrador de varios convenios.
SCRUM MASTER	Msc Jorge Miño	Director del proyecto

DEVELOPMENT TEAM	Ludwing Gómez Lizbeth Santamaría	Desarrolladores del proyecto
---------------------	-------------------------------------	------------------------------

Para el levantamiento de requisitos, se realizaron varias entrevistas, las cuales fueron necesarias para analizar el trabajo asociado a los convenios en la EPN. Durante estas entrevistas, se abordaron aspectos clave, como la suscripción, la edición y la gestión de actividades en relación con cada convenio de la EPN.

El enfoque principal se centra en la creación de una aplicación web que simplifique y optimice la gestión y el seguimiento de los convenios que se suscriben en la EPN. Esto implica que los usuarios podrán realizar diversas acciones, tales como visualizar, registrar, actualizar y supervisar los convenios. Además, podrán generar informes e historial de procesos, así como agregar observaciones para el seguimiento correspondiente. Con dicha información fue posible descubrir los diversos procedimientos que implementaría la aplicación, que incluyen:

- **Gestión de usuarios.** La aplicación deberá soportar el registro de todos los trabajadores de la EPN, los cuales tendrán de una u otra manera una relación con un convenio, todo depende de los diferentes roles. Cada vez que ingrese un trabajador a la EPN se le deberá crear un usuario y contraseña el cual le permitirá ingresar a la aplicación.
- **Gestión de convenios.** Todo el flujo principal de la aplicación comienza con la creación de un convenio, en donde se ingresará la mayoría de la información la cual posteriormente servirá para que la búsqueda pueda ser más refinada, y eficiente. En este proceso se relaciona un convenio con un trabajador, a partir de la designación como administrador o fiscalizador. Además, en ocasiones muchos de los convenios sufren actualizaciones a partir de adendas, lo que permite principalmente cambiar los administradores, objetivos y obligaciones.
- **Gestión de observaciones y seguimiento.** Cada administrador requiere llevar un control de las actividades de su convenio, según menciona el instructivo de convenios de la EPN. A partir de los informes se debe detallar las actividades realizadas, y qué beneficios trae o con que obligación está relacionada, entre otras cosas que permiten llevar un control detallado, de tal manera que pueda ser útil para la generación de reportes automáticamente, específicamente para la automatización del informe de seguimiento.

La definición de los diferentes tipos de usuario que utilizará la aplicación surgió simultáneamente con la revelación de los diferentes procesos. Estos tipos de usuarios se enumeran en la Tabla 5:

Tabla 5 Tipos de usuario

Tipo de usuario	Descripción
Administrador	El único rol que se agrega automáticamente al desplegar la aplicación además tendrá las opciones de agregar y crear usuarios, es el administrador del sistema como tal.
Iniciador	El usuario que se ocupa de cargar el convenio, y que puede asignar su administrador, fiscalizadores correspondientes.
Encargado	Es el trabajador que está ligado a un convenio, a partir de las actividades que hace en pro de los convenios podrá crear observaciones, e informes que sirven para enviar los resultados obtenidos en función del objeto del convenio suscrito.
Fiscalizador	Supervisan como van todos sus convenios, a partir de las observaciones que va realizando el encargado. Se pueden denominar los encargados de los encargados del convenio.
Externo	Solo tiene permisos de lectura al repositorio de convenios.

2.3 Product Backlog

El Product Backlog es el conjunto de todas las historias de usuario, son creadas y trabajadas juntos por el Product Owner y el equipo de desarrollo para describir las funcionalidades del sistema.

El desarrollo del proyecto inició el lunes 8 de mayo del 2023 y concluyó el viernes 19 de agosto del mismo año, teniendo una duración aproximada de 3 meses y 11 días.

En la Tabla 6 se presenta el Product Backlog, el cual ha sido elaborado tomando en cuenta los requisitos de los usuarios y será de utilidad para la implementación de cada sprint. Cabe recalcar que, debido a la extensión, no pueden ser incorporadas en la tabla los criterios de aceptación de cada historia de usuario, aun así, se la puede encontrar en el [Anexo II](#). Las historias de usuario han sido divididas según su alcance en el sistema.

Tabla 6 Product Backlog

ID	Nombre de la épica	ID	Descripción	Prioridad
E1	Definición y Selección del Stack Tecnológico para el Desarrollo Frontend y Backend	HU0	Como desarrollador necesito definir la tecnología y stack del frontend y backend para asegurar una base sólida y escalable para la funcionalidad principal de la aplicación.	ALTA
		HU1	Como desarrollador, quiero investigar y evaluar diversas herramientas y tecnologías de frontend y backend para elegir las más adecuadas para el proyecto.	ALTA
		HU2	Como desarrollador principal, quiero establecer estándares y directrices de codificación de backend para garantizar un código base uniforme y mantenible para todo el equipo.	MEDIA
		HU3	Como desarrollador, quiero implementar el frontend y el backend de forma iterativa e incremental, centrándome en características y funcionalidades individuales en iteraciones, para garantizar un progreso constante y una validación temprana de los componentes.	ALTA
		HU4	Como desarrollador, necesito gestionar los datos de la empresa, para estructurar su modelo de negocio.	ALTA
E2	Desarrollo de Login	HU5	Como usuario, necesito iniciar sesión en el sistema, para acceder de manera segura y personalizada a mis datos y funcionalidades de acuerdo con mi rol.	ALTA
E3		HU6	Como administrador, necesito poder registrar nuevos usuarios proporcionando la información requerida (email, contraseña),	ALTA

			para que puedan acceder a la aplicación con sus cuentas.	
	Desarrollo del módulo de administración	HU7	Como administrador, necesito poder asignar roles de usuario (Administrador, Iniciador, Encargado, Fiscalizador, Externo) para diferenciar los niveles de acceso y responsabilidades.	ALTA
		HU8	Como administrador, quiero ver una lista de todos los convenios registrados en el sistema, para acceder rápidamente a información relevante sobre cada convenio, como título, fechas, estado, tipo, dependencia, institución y estado del convenio.	MEDIA
		HU9	Como administrador, deseo poder crear, editar, actualizar y ver las dependencias, instituciones, estados del convenio y tipos de convenio, para asegurarme de que la información sea precisa y actualizada.	MEDIA
E4	Desarrollo del módulo de registro de convenios	HU10	Como iniciador, quiero iniciar sesión en el sistema utilizando mis credenciales para acceder a las funciones de registro de convenios.	MEDIA
		HU11	Como iniciador, quiero subir un convenio al sistema, proporcionando los siguientes detalles: <ul style="list-style-type: none"> ● Número de convenio. ● Título del convenio ● Nacionalidad ● Estado del convenio ● Tipo de convenio 	ALTA

		<ul style="list-style-type: none"> ● Institución ● Archivo PDF del convenio. ● Dependencia a la que pertenece el convenio. ● Administrador asignado al convenio. ● Fiscalizador asignado al convenio ● Objetivo/s ● Fecha de inicio, suscripción y prevista finalización <p>para que pueda ser buscado fácilmente.</p>	
	HU12	Como iniciador, quiero subir una adenda relacionada al convenio para que este pueda ser editado sin ningún problema.	MEDIA
	HU13	Como iniciador necesito que la información relacionada a un convenio pueda ser editada siempre y cuando esté cargada adenda correspondiente. Se deberá implementar una pantalla para cargar la adenda. Si el convenio no cuenta con una adenda registrada, al intentar realizar una actualización, se mostrará una alerta que indique que es necesario cargar una adenda previamente.	MEDIA
	HU14	Como iniciador, quiero poder visualizar una lista de todos los convenios que se han cargado en el sistema, para poder realizar un seguimiento y gestionarlos de manera efectiva.	MEDIA
	HU15	Como iniciador, quiero finalizar los convenios en caso de que haya sido informado como terminado por las debidas autoridades con un acta de finiquito, para poder gestionar solo los	BAJA

			convenios activos. (" Luego de que la documentación de finalización pasa por los jurídicos y rectorado, se debe tener el acta de finiquito, el cual será subido por la DRI y se actualiza el convenio como finalizado").	
E5	Acceso a los convenios sin requerir el registro o inicio de sesión	HU16	Como usuario externo, quiero tener la posibilidad de visualizar y descargar los convenios de la EPN sin tener que registrarme o iniciar sesión, para facilitar el acceso rápido y sin restricciones a la información de los convenios.	ALTA
		HU17	Como usuario externo, deseo tener la capacidad de filtrar los convenios según dependencia, institución, nacionalidad, fechas, tipo y estado, para realizar búsquedas más específicas y eficientes.	MEDIA
E6	Desarrollo del módulo de seguimiento	HU18	Como encargado, quiero acceder a un menú de seguimiento donde pueda ver todos los convenios asignados a mí, para poder darles el seguimiento respectivo.	MEDIA
		HU19	Como encargado, quiero poder crear observaciones en los convenios asignados, proporcionando información detallada sobre el proceso de seguimiento, coordinación, control y evaluación del convenio.	ALTA
		HU20	Como encargado quiero que las observaciones puedan ser un conjunto de sub-observaciones, para tener la libre posibilidad de segmentar según mi conveniencia.	ALTA
		HU21	Como encargado, quiero tener un historial de procesos en cada convenio, donde se registren todas las observaciones y acciones	ALTA

			realizadas a nivel administrativo, técnico y logístico para el cumplimiento de los objetivos del convenio.	
		HU22	Como encargado, quiero generar informes automáticos, al menos un informe ejecutivo anual, que muestre los resultados obtenidos en función del objeto del convenio suscrito para presentar a mis autoridades.	MEDIA
		HU23	Como encargado, quiero tener la posibilidad de solicitar la autorización o aprobación de las máximas autoridades de la universidad para la prórroga, ampliación, adenda o terminación del convenio, a través de los informes generados.	MEDIA
		HU24	Como encargado, deseo tener la opción de editar el informe generado para poder aumentar más información en caso de que fuera el caso.	MEDIA
		HU25	Como encargado, deseo tener la opción de guardar el informe generado para que repose como documentación asociada al convenio y poder revisarlo cuando desee.	BAJA
		HU26	Como encargado, quiero registrar puntos importantes tratadas en todas las reuniones realizadas con los representantes del otro organismo para que sirva como un insumo adicional para las observaciones.	MEDIA
		HU27	Como encargado, deseo recibir una notificación que el contrato del convenio está próximo a acabar en un plazo de 2 meses, para poder realizar las acciones debidas dentro del proceso de finalización.	MEDIA

		HU28	Como fiscalizador, quiero recibir notificaciones por correo electrónico cuando sea asignado a un nuevo convenio, para estar al tanto de las acciones hechas en el convenio por parte del encargado.	MEDIA
		HU29	Como fiscalizador, deseo acceder a un menú de seguimiento donde pueda ver todos los convenios asignados a mí, para poder revisar las acciones realizadas por el encargado.	ALTA
		HU30	Como fiscalizador, necesito tener acceso al historial completo de observaciones y acciones realizadas en cada convenio, para verificar la transparencia y el cumplimiento de los procedimientos establecidos.	ALTA
		HU31	Como fiscalizador, deseo recibir una notificación cada vez que el encargado guarde un informe, para revisar los resultados obtenidos en función del objeto del convenio.	ALTA
		HU32	Como fiscalizador, quiero recibir notificaciones 2 meses antes de cuando se acerque la fecha límite de finalización de un convenio, para tomar acciones pertinentes en caso de que el encargado no esté al tanto.	MEDIA
		HU33	Como fiscalizador, deseo recibir una notificación cuando un convenio haya finalizado oficialmente, para verificar que se haya seguido el proceso adecuado y se haya cumplido con todas las obligaciones establecidas.	MEDIA
E7	Corrección de bugs	HU34	Como desarrollador, necesito identificar y corregir cualquier error tanto en el backend como en el frontend de la aplicación, con el fin	MEDIA

			de garantizar una ejecución exitosa de la aplicación en producción.	
--	--	--	---	--

2.4 PLANIFICACIÓN

Como menciona [8], el trabajo dentro de SCRUM se lleva a cabo en iteraciones o ciclos de hasta un mes, con el fin de que cada secuencia de trabajo debe generar algo de valor tangible para el cliente. La Tabla 7 muestra la planificación y segmentación de los diferentes sprints, tomando en cuenta las limitaciones del proyecto asociadas a la planificación académica de un periodo académico.

Tabla 7 Planificación de ejecución de sprints

	Semana 1,2 Sprint 0	Semana 3,4,5 Sprint 1	Semana 6,7,8 Sprint 2	Semana 9,10,11 Sprint 3	Semana 12,13,14 Sprint 4
Arquitectura del proyecto	8 DE MAYO				
Convenios		22 DE MAYO			
Historial de procesos			12 DE JUNIO		
Informes				3 DE JULIO	
Autenticación y autorización					24 DE JULIO

Se espera acabar el desarrollo el 6 de agosto, a partir de ahí realizar las respectivas pruebas para confirmar la calidad del producto.

2.5 Sprint 0

SPRINT PLANNING

La arquitectura de la aplicación, el modelo de entidad relación, el diseño de la vista principal de la aplicación y la configuración del marco de trabajo son los objetivos de este sprint. El sprint backlog se presenta en la tabla 8.

Tabla 8 Sprint Backlog 0

HU	Código	Descripción	Responsable
HU 1	TA01	Investigar las diferentes herramientas y tecnologías de frontend y backend que se pueden usar para el proyecto.	Lizbeth Santamaría Ludwing Gomez
	TA02	Evaluar las diferentes herramientas y tecnologías en función de los requisitos, limitaciones y características.	Lizbeth Santamaría Ludwing Gomez
	TA03	Realizar la arquitectura de la aplicación.	Ludwing Gómez
HU 2	TA04	Investigar y recopilar las buenas prácticas de codificación	Ludwing Gomez
HU 3	TA05	Configurar el entorno de desarrollo y la estructura del proyecto.	Lizbeth Santamaría Ludwing Gomez
HU4	TA06	Diseñar el modelo entidad relación base de la aplicación.	Ludwing Gómez
	TA07	Crear el script de la base de datos	Ludwing Gómez
	TA08	Pruebas de funcionamiento de la base de datos	Ludwing Gómez
	TA09	Diseño estructural de la vista principal de la aplicación.	Lizbeth Santamaría

Ejecución del sprint 0

Análisis de arquitectura

Se espera elegir el diseño arquitectónico con el fin de seguir un conjunto de lineamientos que hagan óptima y sencilla la obtención del producto final. La arquitectura de software tiene la misma importancia que la arquitectura en el mundo de la edificación, pues a partir del diseño arquitectónico se plantean las bases y se conforma la estructura de lo que será el producto final.

Como menciona [14], la estructura arquitectónica que se elija para un sistema dependerá de los requisitos no funcionales. En este proyecto sobresale la importancia de la seguridad, además de que la parte interesada siempre está con la predisposición de que este sistema pueda funcionar para diferentes universidades. Se puede concluir que la escalabilidad y mantenibilidad, son muy importantes también. Existen señales que sugieren que la arquitectura debería adoptar un enfoque de estructura en capas. En esta estructura, cada

componente funcionaría de manera independiente y sería fácil de modificar. Los elementos más críticos estarían resguardados en las capas internas, y un riguroso nivel de validación de seguridad sería implementado en estas áreas. La Figura 3 ofrece una descripción de la arquitectura de alto nivel.

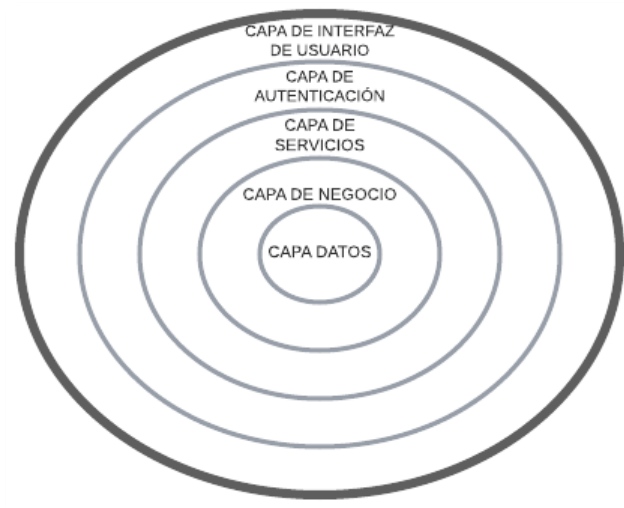


Figura 3 Arquitectura de alto nivel [Autoría propia]

Para el sistema de control y seguimiento de convenios se ha destinado las siguientes capas principales:

La capa de interfaz de usuario, la cual se encuentra en la capa externa del sistema y está compuesta por las partes que se encargan de presentar la interfaz gráfica y permitir la interacción con el usuario. Los elementos visuales que permiten al usuario interactuar con el sistema, como pantallas, formularios, botones y menús, se implementan en la capa de interfaz de usuario. Esta capa recibe las acciones del usuario, como clics, entradas de texto o selecciones, y las envía a las capas inferiores para procesarlas.

Antes de llegar a la capa de servicios o negocios, estas solicitudes pasan por la capa de autenticación y autorización, en donde se gestionan los datos de los usuarios, como credenciales de inicio de sesión, roles y permisos. Se verifica la identidad y los permisos del usuario, la autenticación se realiza para asegurarse de que el usuario es quien afirma ser y que tiene las credenciales correctas, para después realizar la autorización para determinar si el usuario tiene los permisos necesarios para acceder a las funcionalidades o realizar las operaciones solicitadas.

Los clientes pueden acceder a varios procesos y funcionalidades del sistema a través de una interfaz o API que se encuentra en la capa de servicios. Aquí se definen y ponen en práctica los servicios que se ofrecen a los clientes. La capa de servicios puede recibir las

solicitudes del cliente, validar los datos, llamar a la capa de negocio para implementar la lógica adecuada y devolver las respuestas al cliente.

La capa de negocio donde se encuentra la lógica de negocio. Aquí se implementan las reglas y las operaciones que definen cómo funcionan y se manipulan los datos en el sistema.

Finalmente, la capa de acceso que se encarga de interactuar con la base de datos. Aquí se implementan las operaciones de lectura, escritura, actualización y eliminación de datos. La capa de acceso a datos recibe las solicitudes de la capa de negocio, ejecuta las consultas correspondientes en la base de datos y devuelve los resultados. Además, puede incluir la lógica de mapeo entre los objetos de dominio y las entidades de la base de datos.

Justificación tecnologías

FireBase

Firebase es una plataforma desarrollada por Google que permite crear aplicaciones móviles y web de manera eficiente. Destaca por sus múltiples ventajas, como la facilidad de uso, la estabilidad, la escalabilidad y un rendimiento óptimo incluso bajo altas cargas de trabajo.

La elección de Firebase se fundamenta en su destacado servicio de autenticación, que se complementa perfectamente con las reglas de seguridad de Firebase. Estas reglas ofrecen un control de acceso y validación de datos altamente adaptable a diferentes niveles de complejidad [26].

La autenticación identifica a los usuarios que solicitan acceso a sus datos y proporciona esa información como una variable, la cual contiene la siguiente información [26]:

- uid: Un ID de usuario único asignado al usuario que hace la solicitud.
- Token: Es un mapa de valores recopilados durante el proceso de autenticación, que incluye los siguientes datos: email, email_verified, phone_number, name, sub (UID del usuario), firebase.identities, firebase.sign_in_provider y firebase.tenant. Además, es posible agregar atributos de autenticación personalizados [26].

Frontend

Inicialmente, se consideró el lenguaje de programación JavaScript debido a su amplio soporte en los navegadores. Como menciona [27], JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas,

especialmente desde la parte del cliente (Frontend). Además, es el lenguaje base para frameworks populares como React, Angular, Vue e Ionic, sin olvidarnos de estándares ampliamente usados como Web Components y librerías basadas en ellos como Stencil o Lit [28].

Ahora que se han nombrado las tecnologías más famosas, procedemos a filtrarlas considerando si es un software libre o no. En este sentido, identificamos a React como una opción adecuada, ya que permite crear páginas dinámicas, interfaces de usuario eficaces y cuenta con una comunidad activa que brinda soporte y numerosas tecnologías, plugins y dependencias complementarias.

FRAMEWORK

En la búsqueda del framework para el desarrollo del frontend de nuestra aplicación web de seguimiento de convenios, hemos evaluado y comparado cuatro frameworks populares compatibles con React: Next.js, Gatsby.js y Remix.js. Cada uno de estos frameworks presenta características únicas y enfoques particulares en aspectos cruciales como el tipo de renderizado, optimización para motores de búsqueda (SEO), escalabilidad, soporte de la comunidad, etc. A través de la siguiente tabla comparativa, se busca identificar el framework que mejor se adapte a las necesidades del sistema y brinde una base sólida para un desarrollo exitoso y eficiente de la aplicación, ver Tabla 9.

Tabla 9 Comparación frameworks para el frontend

Característica	Next.js [29], [30]	Gatsby.js [31], [32]	Remix.js [33], [34]
Tipo de renderizado	Generación estática (SSG) y Renderizado del lado del servidor (SSR).	Renderizado del lado del servidor (SSR), Generación de sitios estáticos diferida (DSG), Generación de sitios estáticos (SSG).	Renderizado híbrido: Combina el renderizado del lado del servidor (SSR) y el renderizado del lado del cliente (CSR).
SEO (Optimización para motores de búsqueda)	Fuerte soporte nativo para optimización SEO.	Especialmente diseñado para SEO.	Puede requerir medidas adicionales para optimizar el SEO, como agregar meta etiquetas.
Escalabilidad	Alta	Alta	Moderada
Curva de aprendizaje	Baja	Moderada	Moderada

Licencia	Open-source	Open-source	Open-source
-----------------	-------------	-------------	-------------

Next.js es un framework diseñado específicamente para crear aplicaciones web dinámicas, interactivas y de alto rendimiento mediante el uso de componentes de React [35]. Sus características principales abarcan un enrutador basado en el sistema de archivos que permite diseños personalizados, enrutamiento anidado, gestión de estados de carga, manejo de errores y más. Ofrece un rendimiento óptimo gracias al renderizado tanto en el lado del cliente como en el servidor. Además, facilita la obtención de datos con el uso de `async/await` en componentes de React, y brinda opciones versátiles de estilización con soporte para CSS Modules, Tailwind CSS y CSS-in-JS. También destaca por su soporte para TypeScript, garantizando una verificación de tipos más eficiente con un complemento personalizado y un verificador de tipos [35].

Sus diversas características hacen que sea una elección destacada para el desarrollo del frontend del proyecto. Además, el equipo de desarrolladores cuenta con una amplia experiencia en el uso de este framework, lo que facilita el proceso de aprendizaje y permite redirigir recursos hacia otras áreas del proyecto.

Backend

La elección de las tecnologías es muy importante en el backend, ya que determina cómo se implementará la arquitectura elegida. Antes que elegir un lenguaje específico en el cual se pueda desarrollar las APIs, se elegirá un framework que se ajuste a las características de este proyecto, entre las que destaca la línea de seguridad. Al fin y al cabo, la mayoría de los lenguajes cuentan con marcos y librerías que facilitan la creación de APIs REST, por ello se puede afirmar que no existe un lenguaje que sea más beneficioso que otro, sino que depende de las necesidades del proyecto.

Se tiene la siguiente lista de los frameworks más conocidos por proporcionar una estructura y un conjunto de herramientas que facilitan el desarrollo de la API REST. Hay que destacar que se toma muy en cuenta la popularidad de estos frameworks, así como si incurre en software propietario o libre, pues a partir de ello se tendrá una gama de beneficios entre las cuales la principal es que dispone de una amplia comunidad de apoyo en cual los desarrolladores puedan respaldarse. Además, es posible clasificar los frameworks de acuerdo con las necesidades de este proyecto siguiendo la estructura presentada en la Tabla 10.

Tabla 10 Comparación frameworks de acuerdo a las necesidades del proyecto

FRAMEWORK	LENGUAJE	SEGURIDAD	ARQUITECTURA	CURVA DE APRENDIZAJE
NEST	JAVASCRIPT	NEST ACCESS CONTROL (<u>NestAC</u>)	<ul style="list-style-type: none"> ● MICROSERVICIOS ● MVC 	BAJO
EXPRESS	JAVASCRIPT	NO DISPONE	<ul style="list-style-type: none"> ● SOA ● SPA ● MONOLÍTICA ● CAPAS 	BAJO
DJANGO	PYTHON	DJANGO AUTHENTICATION FRAMEWORK	<ul style="list-style-type: none"> ● MVC ● MICROSERVICIOS ● SPA ● CAPAS 	MEDIO
SPRINGBOOT	JAVA	SPRING SECURITY	<ul style="list-style-type: none"> ● MICROSERVICIOS ● CAPAS ● MVC 	BAJO
LARAVEL	PHP	LARAVEL PASSPORT	<ul style="list-style-type: none"> ● MVC ● SOA ● CAPAS 	ALTO
ASP.NET MVC	C#	ASP.NET IDENTITY	<ul style="list-style-type: none"> ● MVC 	MEDIO

La mejor opción que se acoge a las necesidades y características del proyecto es el framework de Spring Boot, pues la popularidad de este se debe a su capacidad para soportar la implementación de aplicaciones con una arquitectura por capas, a partir de la facilitación de inyección de dependencias de manera automática en los componentes de cada capa. Los componentes pueden ser segmentados fácilmente a partir de simples anotaciones, las cuales ayudan a establecer una estructura clara y coherente dentro de la aplicación. Lo que promueve la separación de responsabilidades entre las diferentes capas, además de contribuir al mantenimiento de un código modular, fácil de entender y de mantener. Así también mediante las anotaciones puede soportar nativamente el desarrollo de APIs RESTful, pues pueden usarse para configurar los endpoints y administrar las solicitudes y respuestas HTTP de manera efectiva [36].

Otros frameworks, como Express y Django, también pueden usar una arquitectura de capas. Sin embargo, la elección del marco apropiado depende de muchos factores, y uno de ellos es la seguridad. Spring Boot ocupa otro producto de su misma línea en lo que respecta en seguridad, Spring Security, el cual tiene disponible la protección contra ataques comunes como Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF), y la configuración de políticas de seguridad a nivel de aplicación [37].

Spring Boot se integra fácilmente con otras herramientas y librerías de seguridad, lo cual es importante puesto que en este proyecto se necesita conectar con Firebase, que es el encargado de la capa de autenticación. A partir de la autenticación realizada, se espera que Spring Boot controle la autorización y demás etapas de seguridad, esto le permite aumentar las capacidades de seguridad de la aplicación como se ilustra en la Figura 4.

Tabla 11 Integración de Firebase con lenguajes de programación [38]

Función	Node.js	Java	Python	Go	C#
Creación de tokens personalizados	X	X	X	X	X
Verificación de token de ID	X	X	X	X	X
Administración de usuarios	X	X	X	X	X
Control del acceso con reclamaciones personalizadas	X	X	X	X	X
Revocación de tokens de actualización	X	X	X	X	X
Importación de usuarios	X	X	X	X	X
Administración de cookies de sesión	X	X	X	X	X

Generación de vínculos de acciones de correo electrónico	X	X	X	X	X
Administración de configuraciones del proveedor SAML/OIDC	X	X	X	X	X
Compatibilidad con los multiusuarios	X	X	X	X	X
Realtime Database	X	X	X	X	
Firebase Cloud Messagin	X	X	X	X	X
FCM Multicast	X	X	X	X	X
Administración de suscripciones a temas de FCM	X	X	X	X	X
Cloud Storage	X	X	X	X	
Cloud Firestore	X	X	X	X	
Administración de proyectos	X	X	X		

Finalmente, debido al ajustado tiempo del cronograma que se tiene dentro del proyecto, la elección de Spring Boot resulta acertada pues los desarrolladores tienen experiencia en esta tecnología y en el lenguaje que maneja (JAVA). Y si ese no fuera el caso, Spring Boot se destaca por su enfoque en la simplicidad y la facilidad de uso, además de tener una activa comunidad de desarrollo y soporte técnico, lo cual significa que se puede acceder a una amplia gama de recursos, instrucciones y ejemplos prácticos [21].

Base de Datos

Para la comparación, se consideraron tres de las bases de datos relacionales de código abierto más populares: PostgreSQL, MariaDB y MySQL. La Tabla 12 muestra si estas bases de datos cumplen con ciertos criterios.

Tabla 12 Bases de datos comparadas de acuerdo con [39] y [40]

Base de Datos	Alto Volumen de Datos	Basada en la comunidad	Velocidad de lectura de datos	Compatibilidad con SpringBoot y arquitectura	Fácil aprender
PostgreSQL	X	X		X	X
MariaDB	X	X	X	X	

MySQL	X		X	X	X
-------	---	--	---	---	---

Después de este análisis, se concluye que MySQL es la mejor opción debido a su curva de aprendizaje mínima y su compatibilidad con SpringBoot.

Preparación para el desarrollo

Estándar de desarrollo

Previo al desarrollo y con el fin de contribuir a la mantenibilidad y estabilidad, los cuales son atributos de calidad que impulsan nuestro proyecto, se estableció la norma de desarrollo que se utilizará en el desarrollo actual y las futuras implementaciones y modificaciones. El estándar detallado se encuentra contenido en [Anexo III](#).

Con ello en la Tabla 13 obtenemos los siguientes resultados del primer sprint:

Tabla 13 Done ejecución sprint 0

Entregable
Manual de estándares de programación.
Creación de proyectos tanto para el Frontend como para el Backend con el estándar base a seguir.
Creación de manuales de instalación de todas las tecnologías en una versión específica.
Creación de documentación de las tecnologías principales.

Retroalimentación (Resultado de Sprint Review y Sprint Retrospective)

Es poco común realizar una Sprint Retrospective al final del Sprint 0, ya que normalmente se utiliza para planificar y preparar un proyecto, tal cual en este caso.

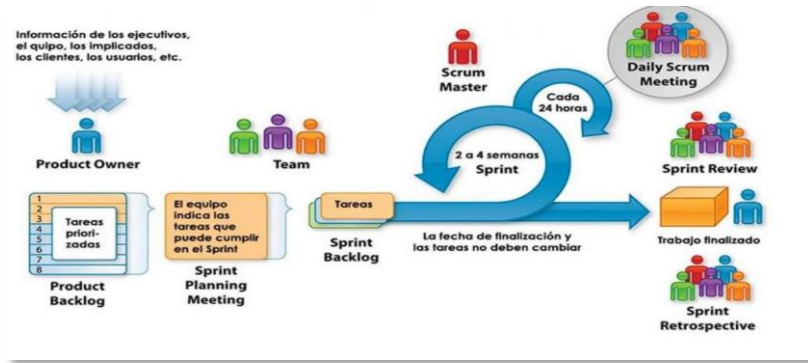


Figura 4 Flujo de scrum tomado de [41]

2.6 Sprint 1

Sprint Planning

Se espera cargar convenios en el sistema para su respectiva asignación a los diferentes trabajadores. Con ello se espera tener un repositorio general de los convenios en donde se puede filtrar a partir de sus detalles. Esto será la base para que en siguientes sprints cuando los trabajadores sean asignados los trabajadores se espera que los diferentes usuarios comiencen a ocupar los distintos roles, ver Tabla 14.

Tabla 14 Sprint Backlog # 1

ID	Código	Descripción	Responsable
HU08	TA01	Diseñar la interfaz de usuario que incluya un repositorio con una lista de los convenios registrados en el sistema.	Lizabeth Santamaría
	TA02	Agregar funcionalidad de búsqueda y filtrado.	Lizabeth Santamaría
	TA03	Habilitar opciones de ordenamiento.	Lizabeth Santamaría
	TA04	Proporcionar botones para acceder a información detallada de cada convenio.	Lizabeth Santamaría
	TA05	Implementar un servicio que posibilite la obtención de los convenios almacenados en la base de datos.	Ludwing Gómez
HU09	TA01	Diseñar pantallas para la gestión de dependencias, instituciones, estados y tipos de convenio.	Lizabeth Santamaría
	TA02	Implementar servicios CRUD y sus respectivos controladores para:	Ludwing Gómez

		<ul style="list-style-type: none"> • Tipo Convenio • Estado Convenio • Dependencia • Institución • Trabajador 	
	TA03	Definir roles y permisos para asegurar que solo el administrador tenga la autoridad para realizar estas acciones de gestión.	Ludwing Gómez Lizbeth Santamaría
HU11	TA01	Diseñar e implementar en base de datos las entidades tipo de convenio, estado de convenio, dependencia e institución que será usada para generar un registro de convenio.	Ludwing Gómez
	TA02	Diseñar e implementar en base de datos las entidades convenio, y adjunto.	Ludwing Gómez
	TA03	Implementar servicio que permita el registro del convenio, tomando en consideración que debe subir el documento asociado.	Ludwing Gómez
	TA04	Crear vista para el registro de convenio	Lizbeth Santamaría
HU12	TA01	Diseñar e implementación de base de datos con las entidades adendas, y adjunto	Ludwing Gómez
	TA02	Implementar servicio que permita cargar la información de la adenda junto a su archivo, el cual está relacionado a un convenio.	Ludwing Gómez
	TA03	Crear vista para la carga de una adenda junto a su motivo del cambio.	Lizbeth Santamaría
HU13	TA01	Implementar servicios que: <ul style="list-style-type: none"> • Verifique si es que un convenio tiene una adenda vigente. Permita la edición de un convenio, tomando a consideración que debe subir un adjunto.	Ludwing Gómez
	TA02	Crear vista para que pueda editar la información relacionada con un convenio, controlando si es que el convenio tiene una adenda vigente.	Lizbeth Santamaría

HU14	TA01	Implementar servicio que muestre la información relacionada a un o muchos convenios, junto a su adjunto relacionado.	Ludwing Gómez
	TA02	Crear vista para que pueda mostrarse todos los convenios creados junto a su información, y las opciones disponibles para un iniciador.	Lizbeth Santamaría
HU15	TA01	Implementar servicio para que una vez cumplido la fecha de finalización del convenio el estado pase a un estado terminado.	Ludwing Gómez
HU16	TA01	Crear vista para que pueda mostrarse todos los convenios creados junto a su información.	Lizbeth Santamaría
HU17	TA01	Crear vista para escoger diferentes opciones de filtrado del repositorio de convenios.	Lizbeth Santamaría
HU27	TA01	Creación de componentes para programar el envío de notificaciones como recordatorios de la finalización del convenio. Establecer una fecha de recordatorio dos meses antes de la fecha prevista de finalización.	Ludwing Gómez
	TA02	Mostrar una notificación al usuario cuando la fecha del recordatorio sea igual a la fecha actual.	Lizbeth Santamaría
HU28	TA01	Establecer un servicio que permita enviar notificaciones por correo electrónico a los fiscalizadores cuando sean asignados a un nuevo convenio.	Ludwing Gómez
HU31	TA01	Crear un servicio en el backend que genere la notificación para el fiscalizador cuando se guarde un nuevo informe.	Ludwing Gómez
	TA02	Definir cómo se enviarán las notificaciones al fiscalizador (por correo electrónico, mensaje en la aplicación, etc.).	Lizbeth Santamaría Ludwing Gómez
HU32	TA01	Establecer una fecha de recordatorio dos meses antes de la fecha prevista de finalización.	Ludwing Gómez
	TA02	Mostrar una notificación al fiscalizador cuando la fecha del recordatorio sea igual a la fecha actual.	Lizbeth Santamaría

HU33	TA01	Crear un servicio en el backend que genere la notificación para el fiscalizador cuando un convenio haya finalizado oficialmente.	Ludwing Gómez
------	------	--	---------------

Ejecución sprint

Los daily scrums proporcionaron información que contribuyó a la finalización de las tareas. Los resultados de los daily scrums se presentan en la Tabla 15.

Tabla 15 Done sprint # 1

Entregable
9 tablas de base de datos
9 servicios API
9 interfaces de usuario
1 reporte de pruebas unitarias JUnit

El sprint se completó con éxito, cumpliendo el objetivo de proporcionar los principales formularios para la gestión de las entidades que rodean a un convenio. Estos formularios permiten crear, editar, modificar y eliminar entidades. El esquema de modelo de base de datos se puede encontrar en el [Anexo IV](#), y el esquema principal de los formularios se muestra a continuación en la Figura 6:

NOMBRE DE LA DEPENDENCIA	ACCIONES
1 Facultad De Ingeniería civil y ambiental	EDITAR ELIMINAR
2 Facultad de ciencias	EDITAR ELIMINAR
3 Facultad De Ingeniería eléctrica y electrónica	EDITAR ELIMINAR
4 Facultad De Geología Y Petróleos	EDITAR ELIMINAR
5 Facultad De Ingeniería De Sistemas	EDITAR ELIMINAR
6 Facultad De Ingeniería Civil Y Ambiental editado	EDITAR ELIMINAR
7 Facultad de Ciencias Administrativas	EDITAR ELIMINAR
8 Escuela Politécnica Nacional	EDITAR ELIMINAR
9 ESFOT	EDITAR ELIMINAR
10 Departamento de Ingeniería Civil y Ambiental (DICA)	EDITAR ELIMINAR

Figura 5 Esquema de formularios [Autoría Propia]

Las tareas se completaron y se determinó que cumplen con los criterios de finalización acordados por el equipo de desarrollo. Como resultado, se ha creado un nuevo artefacto dentro del marco de trabajo, que es un incremento de producto potencialmente enviable.

Retroalimentación (Resultado de Sprint Review y Sprint Retrospective)

Se convocó a los dos representantes encargados de este proyecto para realizar la revisión de sprint del módulo. La validación correspondiente se realizó utilizando los criterios de aceptación de cada historia de usuario, que se muestran en la Tabla 14, y el detalle se encuentra en el [Anexo II](#).

- En la vista del repositorio externo en la parte de filtrar los criterios de búsqueda, la parte de las fechas se lo realice por un conjunto de rangos.
- En la vista del repositorio externo aumentar la información de fechas al encontrar los convenios filtrados.
- En la vista de registro de convenio con la subida de su adjunto, se espera que sean varios archivos de respaldo, y no solo uno como está ahora.
- Se espera que pueda haber varias adendas activas que corresponden a un convenio.
- En lo que respecta a finalización, no se espera que automáticamente se finalice, sino que el iniciador también pueda finalizar el convenio, pero solo después de haber pasado el debido proceso, para ello el sistema deberá realizar recordatorios de 2 meses, 1 mes, 15 días.
- Se espera una vista de carga de finiquito.
- Se espera que por cada institución se pueda ingresar información del administrador del convenio de la contraparte (correo electrónico).

2.7 Sprint 2

Sprint Planning

El principal objetivo es aumentar la funcionalidad del producto tomando en cuenta la retroalimentación del sprint anterior. Además, de los convenios generados, se espera que el encargado pueda ingresar distintas observaciones que sucedan a lo largo de la vigencia del convenio. Cada observación puede tener múltiples sub-observaciones con el fin de que toda la información de actividades pueda ser detallada, para su posterior utilización en la generación de reportes e informes automáticos que será tomado en cuenta para el siguiente sprint, ver Tabla 16.

Tabla 16 Sprint Backlog # 2

ID	Código	Descripción	Responsable
HU18	TA01	Desarrollar e implementar base de datos intermedia para las tablas de convenio y trabajador, la cual tendrá atributos para identificar si es encargado o fiscalizador.	Ludwing Gómez
	TA02	Implementar servicio que liste los convenios que esta relacionados a un encargado de la tabla trabajador.	Ludwing Gómez
	TA03	Crear vista para listar todos los convenios asignados a un encargado, además de las diferentes opciones que puede tener para gestionar su convenio.	Lizbeth Santamaría
HU19	TA01	Desarrollar e implementar base de datos de observaciones.	Ludwing Gómez
	TA02	Implementar servicio que permita agregar observaciones las cuales estan asociadas a un convenio en específico.	Ludwing Gómez
	TA03	Crear vista de historial de procesos que pertenece a un convenio en donde se podrá crear varias observaciones.	Lizbeth Santamaría
HU20	TA01	Desarrollar e implementar base de datos de sub-observaciones.	Ludwing Gómez
	TA02	Implementar servicio que permita agregar sub-observaciones las cuales están asociadas a una observación.	Ludwing Gómez
	TA03	Aumentar opciones a la vista de observaciones para que pueda tener una lista de observaciones.	Lizbeth Santamaría
HU21 HU26	TA01	Depurar la vista de observaciones y sub-observaciones para que se pueda ver como un historial de procesos.	Lizbeth Santamaría
HU29	TA01	Implementar servicio que liste los convenios que esta relacionados a un encargado de la tabla fiscalizador.	Ludwing Gómez
	TA02	Crear vista para listar todos los convenios asignados a un fiscalizador, además de las diferentes opciones que puede tener para gestionar su convenio.	Lizbeth Santamaría

HU30	TA01	Crear acceso al historial de procesos del convenio, en donde solo podrá ver las observaciones y sub-observaciones mas no crear.	Lizbeth Santamaría
HU31	TA	Actualizar en el repositorio de los convenios en donde es fiscalizador para que se pueda ver el documento, de tal manera que pueda acceder a las obligaciones.	Lizbeth Santamaría

Ejecución sprint

Durante este sprint se obtuvo los resultados según la Tabla 17.

Tabla 17 Done Sprint 2

Entregable
3 tablas de base de datos
3 servicios API
5 interfaces de usuario
1 reporte de pruebas unitarias JUnit

Estos elementos contribuyen a la implementación de la funcionalidad base de esta aplicación, que es llevar a cabo el seguimiento de los convenios. Como se ha mencionado, se siguió el diseño de un historial de procesos, que podría considerarse un posible entregable, sujeto a las observaciones y comentarios que se reciban, como se muestra en la Figura 7.



Figura 6 Esquema principal del seguimiento por convenio [Auditoría Propia]

Retroalimentación (Resultado de Sprint Review y Sprint Retrospective)

Se contó con el apoyo del Product Owner y el Scrum Master para seguir el mismo procedimiento que en el sprint anterior. Los comentarios detallados se pueden encontrar en la lista a continuación.

- De ser posible plantearse la creación de un nuevo módulo, denominado Notas, donde se detallará el seguimiento que hace el fiscalizador al encargado.
- Se debe corregir los botones los cuales deben cumplir con los estándares y principios web. En la vista de seguimientos, los botones que amplían las observaciones para mostrar las sub-observaciones deberían usar una flecha en lugar de un símbolo más.
- Se espera que las observaciones y sub-observaciones también puedan ser editadas y eliminadas, en el último caso se espera que sea una eliminación lógica.

2.8 Sprint 3

Sprint Planning

En este sprint se espera corregir los errores encontrados en el sprint anterior. Una vez corregido se espera implementar la generación de reporte automático a partir de los formatos ya establecidos por el cliente, ver Tabla 18.

Tabla 18. Sprint Backlog # 3

ID	Código	Descripción	Responsable
HU19	TA01	Implementar servicio que permita editar y eliminar observaciones.	Ludwing Gómez
	TA02	Actualizar vista de historial de procesos que pertenece a un convenio en donde las observaciones podrán ser editadas y eliminadas.	Lizbeth Santamaría
H20	TA01	Implementar servicio que permita editar y eliminar sub-observaciones.	Ludwing Gómez
	TA02	Actualizar vista de historial de procesos donde las sub-observaciones pertenecientes a una observación puedan ser editadas y eliminadas.	Lizbeth Santamaría
HU21 HU26	TA01	Depurar la vista de observaciones y sub-observaciones para que se pueda ver como un historial de procesos siguiendo los estándares web.	Lizbeth Santamaría

HU22	TA01	Crear servicio para poder generar el informe en distintos tipos de documento a partir de la información ingresada en el módulo de historial de proceso.	Ludwing Gómez
	TA02	Crear vista para el módulo de informes.	Lizbeth Santamaría
	TA03	Crear vista donde se encuentre el boton que genere el informe y posteriormente pueda ser visualizado.	Lizbeth Santamaría
HU23 HU24	TA01	Crear opción para que pueda descargar el archivo en distintos formatos y así que la información del archivo pueda ser editado y posteriormente enviar a las autoridades.	Lizbeth Santamaría
HU25	TA01	Actualizar en el mismo servicio de generar el informe, se guarde en la carpeta de adjuntos donde se debe relacionar con un convenio específico.	Ludwing Gómez

Ejecución sprint

Después de completar todas las tareas, se obtuvieron los entregables, que se muestran en la Tabla 19.

Tabla 19 Done Sprint 3

Entregable
1 tabla nueva y actualización de 3 tablas de base de datos
6 servicios API
2 interfaces de usuario
1 reporte de pruebas unitarias JUnit

Los resultados de este sprint han ayudado a complementar el seguimiento mediante la generación de informes automáticos. El informe de ejecución y seguimiento ha sido una prioridad, ya que es utilizado por cada uno de los encargados para enviar el detalle de las acciones que han realizado con sus respectivos convenios. En la Figura 8, se presenta el

informe original en formato xlsx el cual se debe automatizar, mientras que en la Figura 9 se presenta el espacio de trabajo para automatizarlo:

ESCUELA POLITÉCNICA NACIONAL GESTIÓN DE CONVENIOS, ACUERDOS, CARTAS DE COMPROMISO O SIMILAR, DONDE LA EPN TENGA COMPROMISOS INSTITUCIONALES			
INFORME DE EJECUCIÓN Y SEGUIMIENTO			
Unidad Requerente Facultad / Dirección		Fecha de suscripción	Periodo a reportar
Título del Convenio, Acuerdo, Carta de Compromiso			
Institución con la que se suscribió el convenio		TIPO DE CONVENIO	CONVENIO ESPECÍFICO
A. DATOS DEL ADMINISTRADOR O RESPONSABLE DEL CONVENIO			
Cédula	Apellidos y Nombres	Departamento / Unidad	Cargo actual en la EPN
B. OBJETO DEL CONVENIO			
Objetivo General	Cumplimiento	Descripción de los beneficios que obtuvo la EPN, con la ejecución el convenio.	
	CUMPLE PARCIALMENTE		
C. COMPROMISOS / OBLIGACIONES			
Compromisos / Obligaciones que constan en el convenio por la EPN	Cumplimiento	Actividades desarrolladas (Anexar los respaldos)	

Figura 7 Formato de informe de ejecución y seguimiento [Autoría Propia]

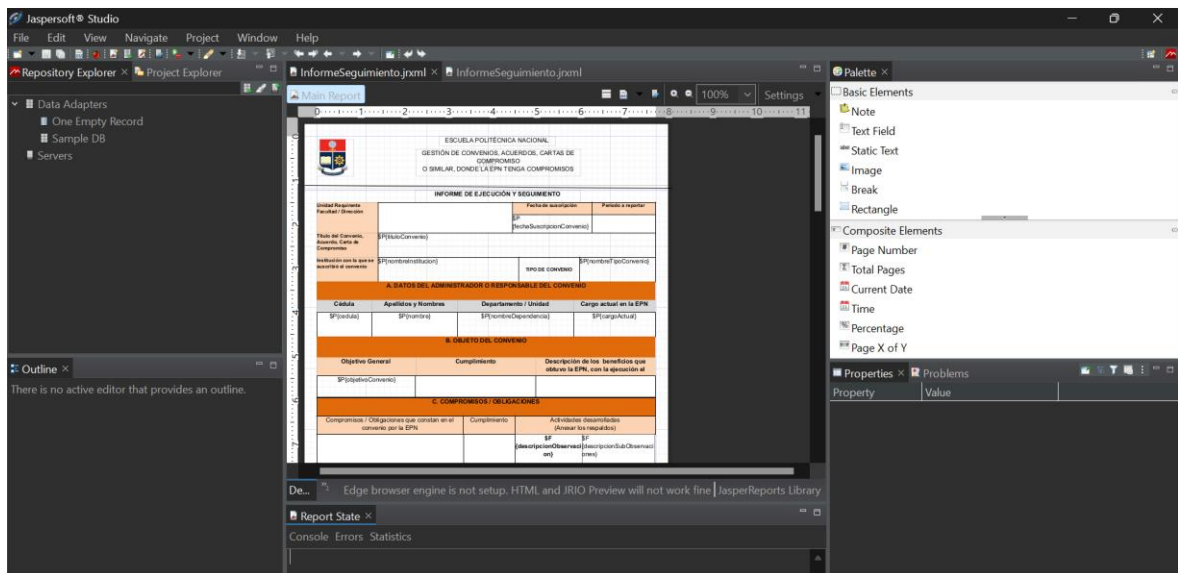


Figura 8 Proceso de automatización de informes [Autoría Propia]

Retroalimentación (Resultado de Sprint Review y Sprint Retrospective)

Con el apoyo del Product Owner y el Scrum Master, recibimos retroalimentación que se detalla en la siguiente lista.

- Se debe aumentar las opciones de descarga de documentos. Inicialmente, los documentos se pueden descargar en formato PDF, pero también debe ser posible descargarlos en los formatos Excel (.xlsx) y Word (.docx) para facilitar la edición.
- Es necesario mejorar el diseño de los informes, pues en algunos casos el informe toma inconsistencias visuales como líneas sobrepuestas, celdas fuera de la tabla, algunos campos que no siguen el estándar de justificación del documento y más.
- De ser posible plantearse la opción de crear un formulario de donde se obtenga la información extra que no es automatizada dentro del informe, como por ejemplo la opción de erogación económica.

2.9 Sprint 4

Sprint Planning

Se espera gestionar la autenticación y la autorización de los recursos que dispondrá el sistema, para ello se espera ocupar el ambiente de Firebase, para posteriormente realizar las conexiones con la capa de servicios y la capa de interfaz de usuario. Para tener usuarios, se debe partir de la tabla trabajador, la cual es gestionada por los administradores tal y como se pudo ver en el primer sprint, por ello es necesario implementar el módulo de administración, ver Tabla 20.

Tabla 20 Sprint Backlog # 4

HU	Código	Descripción	Responsable
HU5	TA01	Crear la página principal del aplicativo.	Lizabeth Santamaría
	TA02	Crear proyecto en FireBase.	Lizabeth Santamaría
	TA03	Crear vista para ver formulario de inicio de sesión.	Lizabeth Santamaría
	TA04	Configurar y conectar la vista con Firebase Authentication.	Lizabeth Santamaría
	TA05	Crear vista para restablecer la contraseña del usuario.	Lizabeth Santamaría
	TA06	Utilizar SDK Firebase para leer los tokens que vendrán en cada petición.	Ludwing Gómez

		Diseñar e implementación de base de datos con las entidades rol, menú, y sus respectivas tablas intermedias.	Ludwing Gómez
		Implementar filtros y configuración de seguridad a partir del token JWT.	Ludwing Gómez
		Implementar servicios que de acuerdo a la información del usuario pasado en el JWT token muestre los roles y módulos que pueden ser accedidos.	Ludwing Gómez
HU6	TA07	Diseñar la interfaz para la creación de nuevos usuarios con campos para email y contraseña.	Lizbeth Santamaría
	TA08	Crear servicio en el backend que actualice el uid de Firebase en la tabla dato personal.	Ludwing Gómez
	TA09	Validar que el email proporcionado sea único, pertenezca a la EPN y cumpla con el formato adecuado.	Lizbeth Santamaría
	TA10	Realizar pruebas para asegurar que el proceso de creación de usuario funcione correctamente en Firebase y en la base de datos.	Lizbeth Santamaría
Ludwing Gómez			
HU7	TA01	Crear servicios en el backend que permita la lista, asignación, actualización y eliminación de roles, en donde en principio estarán los roles definidos.	Ludwing Gómez
	TA02	Diseñar una interfaz para que el administrador pueda crear, ver, editar y eliminar roles.	Lizbeth Santamaría
	TA03	Crear servicios en el backend que permita la asignación de un rol tanto a un usuario como a una lista de menús que podrán ser accedidos.	Ludwing Gómez

	TA04	Implementar una página de administración de roles para que el administrador pueda asignar roles a usuarios existentes.	Lizbeth Santamaría
		Implementar una página de administración de menús por rol para que el administrador pueda asignar los menús que pueden ser accedidos por un rol.	Lizbeth Santamaría
	TA05	Hay que asegurar que solo el administrador tenga acceso a la funcionalidad de creación y asignación de roles.	Lizbeth Santamaría
			Ludwing Gómez
TA06	Probar que cada usuario pueda acceder a los recursos que el rol disponga.	Lizbeth Santamaría	
HU10	TA01	Crear un servicio en el backend que permita verificar las credenciales ingresadas por el iniciador y autenticar al usuario.	Ludwing Gómez
HU34	TA01	Depurar las pruebas unitarias, y de integración, con el fin de realizar las últimas pruebas de usabilidad para identificar y resolver posibles errores en el sistema.	Lizbeth Santamaría
			Ludwing Gómez
	TA02	Realizar una revisión exhaustiva del código para identificar los errores relacionados con el manejo de datos tanto en el backend como en el Frontend.	Lizbeth Santamaría
			Ludwing Gómez
	TA03	Aplicar las correcciones necesarias en el frontend y backend para solucionar los problemas de manejo de datos.	Lizbeth Santamaría
			Ludwing Gómez

Ejecución sprint

Después de completar todas las tareas, se obtuvieron los entregables, que se muestran en la Tabla 21.

Tabla 21 Done Sprint 4

Entregable

Modelo entidad relación
15 servicios API
5 interfaces de usuario
1 reporte de pruebas unitarias JUnit

Los resultados de este sprint ayudan a asegurar la seguridad de la aplicación, a partir de la autenticación y autorización de los diferentes roles de usuario que tendrá el sistema. Para ello se ocupó la ayuda del componente de Firebase para la gestión de usuarios.

Además de diversas configuraciones adicionales en el Backend y Frontend, que incluye la configuración de rutas tanto públicas-privadas, con restricciones correspondientes basadas en los roles y permisos. La creación de un módulo de administración plenamente funcional ha sido un éxito. La Figura 10 muestra uno de los módulos del sistema de administración, que permite la creación de usuarios.



Figura 9 Módulo de administración donde se gestiona los usuarios [Autoría Propia]

Retroalimentación (Resultado de Sprint Review y Sprint Retrospective)

Para realizar la revisión del sprint del módulo, se convocó al Product Owner y al Scrum Master. Se recibieron comentarios detallados, que se enumeran a continuación.

- El menú que contiene los módulos a los que puede acceder un usuario debe optimizarse, ya que está ralentizando el flujo óptimo para el usuario.
- Se espera que se pueda gestionar la creación de usuarios trabajadores a partir de una api o base de datos que sea proporcionada por el departamento técnico de la universidad, que sería la Dirección de Gestión de la Información y Procesos “DGIP”.

- No es conveniente que, al asignar un encargado a un convenio en el registro, se le provea automáticamente sus permisos, sino que debe haber un módulo de asignación de roles.
- En concordancia con el punto anterior, se espera que este sistema sea mantenible y escalable por lo que se necesita que esa parte de la administración de roles y permisos pueda ser gestionado por una persona a partir de la aplicación misma.

2.10 Documentación

Para facilitar las futuras implementaciones, la parte interesada necesitó una documentación adecuada para poder aumentar nuevas funcionalidades si así lo quisiera. Esto se debe a que esta aplicación puede ser utilizada en otras instituciones, especialmente en entidades públicas. Para cumplir con este requisito, se creó una documentación del API REST que incluye un documento de Swagger, que se puede encontrar en el [Anexo V](#), además esa información también puede ser utilizada a partir de un repositorio de la herramienta POSTMAN, almacenado en el [Anexo VI](#), así también un manual del programador generado al empezar el desarrollo. Debido a esto, se espera que este sistema no tenga ningún problema para que sea mantenible y escalable.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Se escogió los frameworks y tecnologías que ayudaron a cumplir con el objetivo del aplicativo, entre las más importantes figuran: Firebase, SpringBoot, Next.js, las cuales fueron la base, y que aportaron significativamente a la optimización de los procesos de los participantes en el flujo de seguimiento de un convenio.

A lo largo de los distintos sprints, el equipo de desarrollo se aseguró de asignar el tiempo necesario para llevar a cabo las pruebas, tanto unitarias como de integración. Estas pruebas son fundamentales dentro de las buenas prácticas y los principios que deben adherirse al implementar la metodología SCRUM. Además, se destinó un tiempo adicional para llevar a cabo una revisión exhaustiva de los puntos clave y críticos, con el propósito de garantizar la calidad de la aplicación. Este enfoque asegura que la calidad no sea sacrificada en ningún momento del proceso de desarrollo en aras de acelerar el proceso.

La Figura 11 y la Figura 12, junto con el [Anexo VII](#), muestran el formato de las pruebas unitarias y sus reportes a nivel de backend. Estas pruebas se implementaron utilizando el framework JUnit, junto con varias librerías, especialmente Mockito.

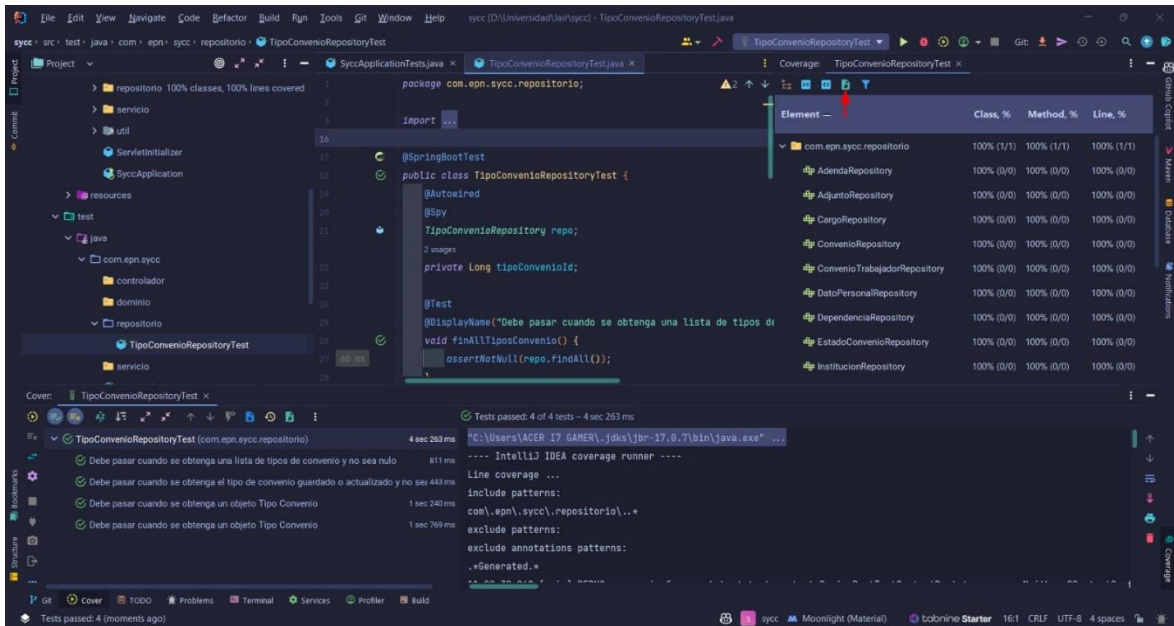


Figura 10 Realización de pruebas para cada componente de cada capa presente en el proyecto [Autoría Propia]

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
BACK_SYCCPARAMETRO	0 of 1,060	100 %	0 of 4	100 %	0	90	0	271	0	88	0	9
Total	0 of 1,060	100 %	0 of 4	100 %	0	90	0	271	0	88	0	9

Figura 11 Esquema de reportes [Autoría Propia]

Para el frontend se empleó el marco de pruebas de JavaScript llamado Jest para llevar a cabo las pruebas unitarias y de integración. Jest se destaca por su enfoque en la simplicidad y es compatible con diversos proyectos que utilizan tecnologías como Babel, TypeScript, Node, React, Angular, Vue, entre otras.

Dado que se está trabajando con Next.js en el frontend, es crucial tener en mente que, en el contexto específico de esta aplicación, existen tres enfoques distintos para adoptar Jest como framework de pruebas. El primero consiste en utilizar ejemplos de inicio rápido, lo que implica generar un proyecto de Next.js preconfigurado con Jest y sus dependencias. El segundo enfoque requiere la configuración de Jest, con el compilador Rust. Por último, el tercer enfoque se centra en la configuración manual de Jest con Babel. La elección del compilador Rust se basó en la estructura preexistente del proyecto. En este sentido, crear un nuevo proyecto con Jest no resultaba una opción viable. Además, el proyecto generado mediante este primer enfoque carecía de los directorios "src" y "app". En el caso del enfoque con Babel, se requería un ajuste manual de la configuración de Jest.

Mientras que en la configuración de Jest con compilador Rust, se especifica que a partir de Next.js 12, se ha incorporado una configuración predefinida para Jest. Esta actualización ha simplificado significativamente la implementación de Jest. Bajo esta nueva configuración, se abordan varios aspectos, como la transformación a través de SWC, auto-mocking de hojas de estilo, la adaptación de variables de entorno, así como la exclusión de los directorios `node_module` y `.next` durante las pruebas [42]. Esta integración de funcionalidades reforzó la elección de la configuración basada en el compilador Rust.

Es importante resaltar que tanto las pruebas de integración, que se ilustran en la Figura 13, como las pruebas unitarias, presentadas en la Figura 14, se realizaron exitosamente.

```
describe( name: "Servicios Dependencias", fn: () : void => {
  test( name: "Verifica que se listen las dependencias - /dependencia/listar", fn: async () : Promise<void> => {
    // Llamamos a la función y obtenemos la respuesta de la API
    const respuestaListaDependencias : Dependencia[] = await getDependencias( props: {token: test_token});
    expect(respuestaListaDependencias).not.toBeUndefined();
    expect(Array.isArray(respuestaListaDependencias)).toBe( expected: true);
    expect(respuestaListaDependencias.length).toBeGreaterThan( expected: 0);
  }, timeout: 10000);

  test( name: "Obtener dependencia por ID - /dependencia/${codDependencia}", fn: async () : Promise<void> => {
    const dependenciaID : Dependencia = await getDependenciasByID( props: {codDependencia: 40, token: test_token});
  });
});
```

Run dev Servicios Dependencias

Test Results 4 sec 828 ms

- Dependencia.test.tsx 4 sec 828 ms
 - Servicios Dependencias 4 sec 828 ms
 - Verifica que se listen las dependencias - /c1 1 sec 7 ms
 - Obtener dependencia por ID - /dependencia/ 724 ms
 - Crear Dependencia - /dependencia/crear 643 ms
 - Actualizar Dependencia - /dependencia/ 1 sec 296 ms
 - Eliminar Dependencia - /dependencia/\$ 1 sec 158 ms

Tests passed: 5 of 5 tests - 4 sec 828 ms

```
jest Servicios Dependencias Dependencia.test.tsx
- info Loaded env from D:\GitHub\seguimiento_control_convenios\front_end\.env
```

Figura 12 Pruebas de Integración – Frontend

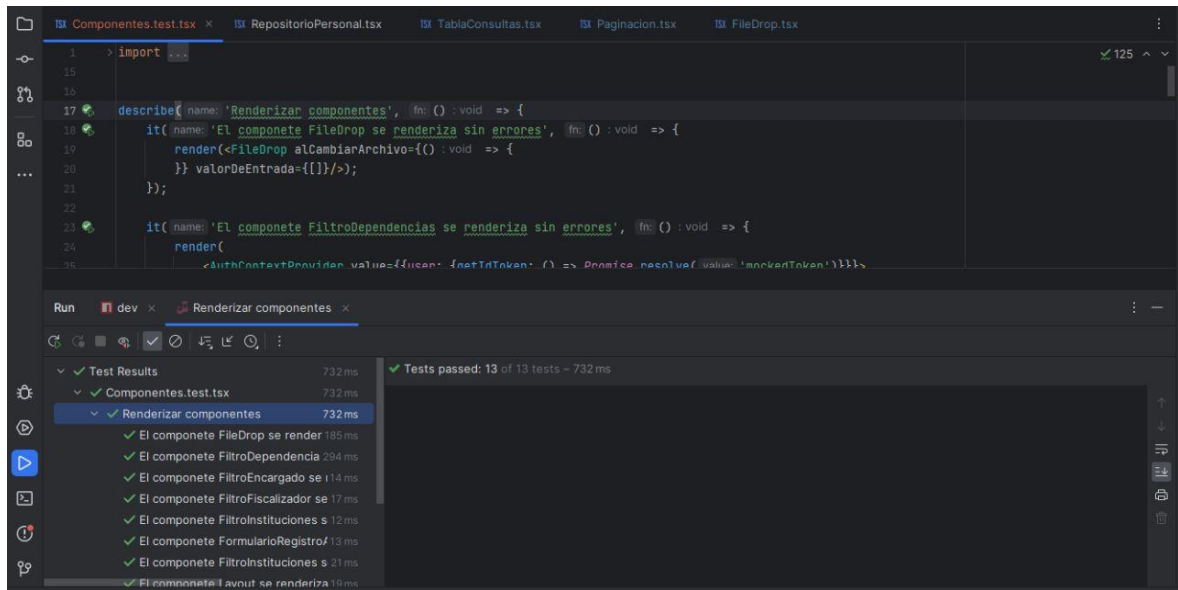


Figura 13 Pruebas unitarias – Frontend

Finalmente, se llevaron a cabo pruebas funcionales y de usabilidad en las que se contó con la participación de 15 usuarios. El procedimiento se realizó a partir de un protocolo de pruebas, incorporando los aspectos y casos de uso específicos del sistema. Estas pruebas se realizaron con el objetivo de evaluar la eficacia, eficiencia y satisfacción de los usuarios al interactuar con el sistema en situaciones representativas de uso. En el documento de pruebas de usabilidad que se encuentra en el [ANEXO XII](#), se detallan las tareas asignadas a los usuarios, los criterios de éxito, los escenarios de prueba y las métricas de rendimiento que se utilizaron para medir el desempeño del sistema en cada caso. El proceso de recopilación de datos se llevó a cabo mediante la administración de encuestas SUS y una encuesta de satisfacción, como se ha descrito en la sección de Criterios de Medición. La combinación de los resultados obtenidos en las pruebas y las respuestas de los usuarios en las encuestas permitió obtener una evaluación comprehensiva de la usabilidad del sistema y proporcionó información valiosa para futuras iteraciones y mejoras.

Además, tras la finalización de los cuestionarios SUS por parte de los 15 participantes, se procedió al cálculo de los puntajes SUS. Estos puntajes constituyen una medida de la usabilidad de un sistema en relación con su efectividad, eficiencia y facilidad de uso [43]. Las respuestas pueden variar entre 0 y 100, siendo un puntaje más elevado indicativo de una mayor usabilidad, ver Tabla 22.

Tabla 22 Tabla de puntajes SUS [43]

SUS Score	Calificación Adjetival
> 80.3	Excelente

68 – 80.3	Bueno
68	Regular
51 - 68	Malo
< 51	Pésimo

El puntaje promedio SUS de los 15 participantes fue de 71.33, lo que sugiere una buena usabilidad en el sistema. Sin embargo, aún existen oportunidades para mejorar y alcanzar puntuaciones más altas en futuras versiones. Los resultados del cuestionario SUS se pueden encontrar en el [Anexo XII](#).

Basándonos en la retroalimentación de los usuarios, fue posible captar sus percepciones más significativas. Esta información permitió concluir que, gracias a la aplicación, los empleados ya no necesitan invertir tanto tiempo en tareas manuales, lo que ha incrementado su eficiencia y liberado recursos para otras actividades estratégicas de los involucrados en los convenios dentro de la EPN.

El sistema de seguimiento de convenios se entregó en tiempo y forma, cumpliendo con los requisitos establecidos al inicio del proyecto. Esto se sustenta en la Figura 15, que se puede encontrar en la herramienta JIRA, donde se detalla todo el proceso de SCRUM. El uso de SCRUM permitió una entrega iterativa y continua de funcionalidades, lo que facilitó la adaptación a los cambios y garantizó que el producto cumpliera con las necesidades del cliente. Para más detalles, consultar el [Anexo VIII](#).



Figura 14 Burndown chart

3.2 Conclusiones

Se revisaron documentos como convenios, contratos, plantillas de informes, referéndums y reglamentos internos. El objetivo era extraer toda la información necesaria y determinar cómo integrarla en la fuente de datos que contribuyó a comprender la problemática en cuestión.

Se descubrieron problemas que preocupaban a la parte interesada, como la cantidad de tiempo que se gastaba en administrar las actividades y el manejo inadecuado de los convenios. En consecuencia, se tomó la decisión de implementar una aplicación web para automatizar los múltiples procedimientos que realiza el personal que supervisa los convenios.

Tras llevar a cabo los análisis y comparaciones pertinentes, se determinó que la adopción de una arquitectura basada en capas resultó ser la opción más adecuada. Esta elección se fundamentó en su capacidad para abordar tanto las exigencias de seguridad como las necesidades de escalabilidad planteadas por las partes involucradas.

Se seleccionó un conjunto de herramientas para desarrollar una aplicación web que permitió un desarrollo rápido, organizado y que permitió la creación de un sistema de alta calidad que no generaba gastos. Se utilizaron SpringBoot, Jasper Report, Postman, Notion, Next JS y Trello. El desempeño, la baja curva de aprendizaje, la facilidad de uso, la experiencia previa y el software libre fueron las razones por las que se decidió utilizar estas herramientas.

Se creó una base de datos con veintitrés tablas relacionadas que permiten controlar la información relacionada con el seguimiento de un convenio. La cual fue utilizada y expuesta a partir de la API de SpringBoot, que utiliza la arquitectura por capas para permitir el acceso y la modificación de datos.

Dentro de cada uno de los proyectos, las pruebas unitarias permitieron verificar las operaciones principales. En el backend, el uso de la librería Junit permite realizar pruebas automatizadas en caso de que se requieran cambios en la API o la base de datos en el futuro. En el lado del front-end, se utilizó Jest, una librería de pruebas que permite escribir pruebas automatizadas para aplicaciones web. Jest es fácil de usar y se puede configurar en pocos minutos, lo que lo hace ideal para ejecutar pruebas en un entorno de desarrollo. Jest se utilizó para llevar a cabo pruebas unitarias y de integración, lo que ayudó a garantizar que la aplicación funcionara correctamente.

El sistema ha sido desarrollado de tal manera que pueda ser escalado y mantenido, debido a la facilidad y adaptabilidad de las tecnologías de desarrollo de software que se utilizó, adicionalmente a la forma que se programó el sistema, pudiendo encontrar los manuales, informes, y documentación generada y recolectada a lo largo del proceso de desarrollo.

El uso de la aplicación web resolvió principalmente el problema de la gestión de actividades que requería mucho tiempo. Se han implementado algunas funciones dentro del sistema que ayudan a optimizar el registro de actividades realizadas en pro de cada convenio, así como a notificar a los encargados y fiscalizadores la proximidad de la terminación de los contratos para que puedan planificar el proceso subsiguiente.

El sistema de convenios proporciona a la universidad una mayor visibilidad y transparencia en sus acuerdos con entidades externas. De tal manera que cualquier interesado, podrá acceder fácilmente a la información de los convenios, lo que facilita la toma de decisiones y la planificación estratégica, e inclusive los estudiantes pueden conocer más detalles acerca de los convenios en los que pueden realizar pasantías.

3.3 Recomendaciones

Se recomienda mejorar, depurar y optimizar el sistema a partir de la identificación y corrección de errores, mejorando la calidad del código, agregando nuevas características, mejorando el rendimiento y mejorando la seguridad, con el fin de cumplir un porcentaje más acercado al 100 % tanto en lo que respecta a la funcionalidad como a la calidad, para que posteriormente pueda ser desplegado, en donde se podrá realizar pruebas de aceptación, rendimiento, estrés, vulnerabilidad, y pentesting.

En caso de que se requiera seguir desarrollando, antes de comenzar, es esencial comprender la arquitectura y la comunicación de cada una de las capas, además se recomienda revisar los manuales del programador, documentación técnica, guías, y arquetipos presentados en los anexos con el fin de seguir con la misma línea del proyecto.

Se recomienda escalar este sistema, especialmente en lo que respecta a la generación de informes automáticos, pues hasta el momento solo se automatiza el informe de seguimiento y ejecución a partir de la información de las observaciones, pero se puede seguir el mismo formato para automatizar otros tipos de informes, actas, expedientes que existen en el proceso en el que participa un convenio.

Se recomienda la implementación en una aplicación móvil, pues permite a los usuarios acceder a la información y las funciones de la aplicación desde cualquier lugar y en cualquier momento. Esto sería especialmente útil para los usuarios que no tienen acceso

a una computadora o que no tienen tiempo para navegar por el sitio web de la tesis. sin olvidar la realización de los estudios previos de viabilidad y factibilidad.

Como se mencionó al principio del documento, el flujo completo de un convenio consta de cuatro largos procesos. Todavía quedan tres procesos que pueden ser automatizados, especialmente la suscripción y finalización. El proceso de suscripción es importante pues es el proceso anterior, en donde se realiza todos los estudios de viabilidad junto con informes técnicos acerca de la posible contratación del convenio. Por su parte el proceso de finalización que es el proceso posterior, el cual implica varios pasos, donde se genera una gran cantidad de documentación, para la creación del acta de finiquito.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] V. S. Abbasova, "MAIN CONCEPTS OF THE DOCUMENT MANAGEMENT SYSTEM REQUIRED FOR ITS IMPLEMENTATION IN ENTERPRISES," *ScienceRise*, Feb. 29, 2020. https://www.researchgate.net/publication/339674095_MAIN_CONCEPTS_OF_THE_DOCUMENT_MANAGEMENT_SYSTEM_REQUIRED_FOR_ITS_IMPLEMENTATION_IN_ENTERPRISES (accessed May 30, 2023).
- [2] NIST, "Security and Privacy Controls for Information Systems and Organizations," Sep. 2020. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf> (accessed May 30, 2023).
- [3] R. Paper Bitzer, "DISENTANGLING THE CONCEPT OF INFORMATION SECURITY PROPERTIES-ENABLING EFFECTIVE INFORMATION SECURITY GOVERNANCE," 2021. <https://www.fim-rc.de/Paperbibliothek/Veroeffentlicht/1160/wi-1160.pdf> (accessed May 30, 2023).
- [4] G. Waja, J. Shah, and P. Nanavati, "AGILE SOFTWARE DEVELOPMENT," *International Journal of Engineering Applied Sciences and Technology*, Apr. 01, 2021. https://www.researchgate.net/publication/354201110_AGILE_SOFTWARE_DEVELOPMENT (accessed May 30, 2023).

- [5] owasp, "Inicio - OWASP Top 10:2021," 2021. <https://owasp.org/Top10/es/> (accessed May 30, 2023).
- [6] EPN, "RESOLUCIÓN ADMINISTRATIVA Nro. 0026-2019," Dec. 31, 21AD. https://www.epn.edu.ec/wp-content/uploads/2022/02/Instructivo_de_convenios-signed-signed.pdf (accessed Jun. 05, 2023).
- [7] "¿Qué pueden hacer las aplicaciones basadas en web? - el enfoque experto - Cynoteck." <https://cynoteck.com/es/blog-post/what-can-web-based-applications-do/> (accessed Jul. 11, 2023).
- [8] B. Meyer, "Agile principles," *Agile!*, pp. 49–78, 2014, doi: 10.1007/978-3-319-05155-0_4.
- [9] "Qué es una base de datos relacional | Oracle Argentina." <https://www.oracle.com/ar/database/what-is-a-relational-database/> (accessed Jul. 11, 2023).
- [10] Amazon, "What is RESTful API? - RESTful API Explained - AWS." <https://aws.amazon.com/what-is/restful-api/> (accessed Jul. 11, 2023).
- [11] IBM, "What is a REST API? | IBM." <https://www.ibm.com/topics/rest-apis> (accessed Jul. 11, 2023).
- [12] C. Muyón and F. Montaluisa, "Métodos de seguridad de la información para proteger la comunicación y los datos de servicios web REST en peticiones HTTP utilizando JSON Web Token y Keycloak Red Hat Single Sign On".
- [13] P. Mahindrakar and U. Pujeri, "Volume-8 Issue-6," *International Journal of Recent Technology and Engineering (IJRTE)*, pp. 2277–3878, 2020, doi: 10.35940/ijrte.F7689.038620.
- [14] I. Sommerville, "Ingeniería de Software," 2011.
- [15] "JetBrains: Essential tools for software developers and teams." <https://www.jetbrains.com/> (accessed Jul. 11, 2023).
- [16] "Pros and Cons of DBeaver 2023." <https://www.trustradius.com/products/dbeaver/reviews?q=pros-and-cons#overview> (accessed Jul. 11, 2023).

- [17] “Notion – Powerful project management, without the chaos.” <https://www.notion.so/product/projects> (accessed Jul. 11, 2023).
- [18] “Top 7 benefits you get by using Github | Apiumhub.” <https://apiumhub.com/tech-blog-barcelona/using-github/> (accessed Jul. 11, 2023).
- [19] “Postman API Platform | Sign Up for Free.” <https://www.postman.com/> (accessed Jul. 11, 2023).
- [20] INFRAESTRUCTURA. DESARROLLO WEB, “NextJS: ¿el futuro de la web? ¿por qué y cuándo usarlo? - Aplyca,” Sep. 14, 2022. <https://www.aplyca.com/blog/nextjs-el-futuro-web-que-es-nextjs> (accessed Jul. 12, 2023).
- [21] VMware, “Qué es y para qué sirve Java Spring Boot.” <https://platzi.com/blog/que-es-spring-boot/> (accessed Jul. 12, 2023).
- [22] Atlassian, “Gitflow Workflow | Atlassian Git Tutorial.” <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> (accessed Jul. 16, 2023).
- [23] Firebase, “Firebase | Google’s Mobile and Web App Development Platform.” <https://firebase.google.com/> (accessed Jul. 16, 2023).
- [24] Firebase, “Firebase Authentication | Simple, no-cost multi-platform sign-in.” <https://firebase.google.com/products/auth> (accessed Jul. 16, 2023).
- [25] MUI, “Overview - Material UI.” <https://mui.com/material-ui/getting-started/> (accessed Jul. 16, 2023).
- [26] Firebase, “Security Rules and Firebase Authentication | Firebase Security Rules,” Jul. 12, 2023. <https://firebase.google.com/docs/rules/rules-and-auth> (accessed Jul. 17, 2023).
- [27] Javier Eguiluz, “Introducción a JavaScript,” Mar. 25, 2009. <https://uniwebsidad.com/libros/javascript?from=librosweb> (accessed Jun. 13, 2023).
- [28] Desarrolloweb, “Javascript.” <https://desarrolloweb.com/home/javascript> (accessed Jan. 31, 2023).

- [29] Next, “Building Your Application: Rendering | Next.js.” <https://nextjs.org/docs/pages/building-your-application/rendering> (accessed Aug. 16, 2023).
- [30] NEXT, “What is SEO? - Search Engine Optimization | Learn Next.js.” <https://nextjs.org/learn/seo/introduction-to-seo> (accessed Aug. 16, 2023).
- [31] Gatsbyjs, “SEO with Gatsby | Gatsby.” <https://v3.gatsbyjs.com/docs/how-to/adding-common-features/seo/> (accessed Aug. 16, 2023).
- [32] Gatsbyjs, “Rendering Options | Gatsby.” <https://www.gatsbyjs.com/docs/conceptual/rendering-options/> (accessed Aug. 16, 2023).
- [33] Remix, “Technical Explanation | Remix.” <https://remix.run/docs/en/main/pages/technical-explanation> (accessed Aug. 16, 2023).
- [34] Remix, “App Tutorial (long) | Remix.” <https://remix.run/docs/en/1.19.3/tutorials/jokes#seo-with-meta-tags> (accessed Aug. 16, 2023).
- [35] Vercel, “Docs | Next.js,” 2023. <https://nextjs.org/docs> (accessed Jul. 17, 2023).
- [36] “Getting Started | Building REST services with Spring.” <https://spring.io/guides/tutorials/rest/> (accessed Jul. 12, 2023).
- [37] “Añade SEGURIDAD a Java con SPRING SECURITY - //Arteco.” <https://www.arteco-consulting.com/post/securizando-una-aplicacion-con-spring-boot> (accessed Jul. 12, 2023).
- [38] “Documentación de Firebase.” <https://firebase.google.com/docs?hl=es-419> (accessed Aug. 17, 2023).
- [39] N. Purohit, “A Comparative Study on Open Source Database Management System,” *Int J Res Appl Sci Eng Technol*, vol. 6, no. 4, pp. 792–795, 2018, doi: 10.22214/ijraset.2018.4133.
- [40] “PostgreSQL vs MySQL: Descubre sus 12 diferencias críticas.” <https://kinsta.com/es/blog/postgresql-vs-mysql/> (accessed Jul. 16, 2023).
- [41] J. Fernando, M. Ayala, I. Barriga Andrade, and J. Javier, “ESCUELA POLITÉCNICA NACIONAL”.

- [42] Next.js, “Optimizing: Testing | Next.js,” 2023. <https://nextjs.org/docs/pages/building-your-application/optimizing/testing#jest-and-react-testing-library> (accessed Aug. 09, 2023).
- [43] UX Research, “Measuring and Interpreting System Usability Scale (SUS) - UIUX Trend.” <https://uiuxtrend.com/measuring-system-usability-scale-sus/#interpretation> (accessed Aug. 17, 2023).