



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE
CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

Joel Estuardo Rodríguez Santos

Asesorado por el M.Sc. Marco Tulio Aldana Prillwitz

Guatemala, noviembre de 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE
CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE
GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JOEL ESTUARDO RODRÍGUEZ SANTOS

ASESORADO POR EL M.SC. MARCO TULIO ALDANA PRILLWITZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. José Francisco Gómez Rivera (a. i.)
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Ing. Kevin Vladimir Cruz Lorente
VOCAL V	Ing. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. José Francisco Gómez Rivera (a. i.)
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADOR	Ing. Sergio Leonel Gómez Bravo
EXAMINADOR	Ing. Carlos Alfredo Azurdia Morales
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 10 de febrero de 2024.

A handwritten signature in black ink, featuring a stylized 'J' and 'R' with a horizontal line extending to the right.

Joel Estuardo Rodríguez Santos

Guatemala, 18 de septiembre de 2024


Ingeniero
Oscar Argueta Hernández
Director de la Unidad de EPS
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero Argueta:

Por este medio hago conocimiento que he revisado y aprobado el informe final de EPS titulado: **“INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por la estudiante **JOEL ESTUARDO RODRÍGUEZ SANTOS** quien se identifica con el registro académico **No. 201115018** y CUI **2098807910101**, autorizando su publicación sin ningún inconveniente.

Sin otro particular, me es grato suscribirme.

Atentamente,



Marco Tulio Aldana Prillwitz
Master in Business Intelligence and Data Analytics
Colegio de Humanidades 30747

Msc. Marco Tulio Aldana Prillwitz
Asesor de EPS
Escuela de Ingeniería en Ciencias y Sistemas

Universidad de San Carlos de
Guatemala



Facultad de Ingeniería
Unidad de EPS

Guatemala, 23 de septiembre de 2024.
REF.EPS.DOC.415.09.2024.

Ing. Oscar Argueta Hernández
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Argueta Hernández:

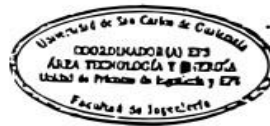
Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Joel Estuardo Rodríguez Santos, Registro Académico 201115018 y CUI 2098 80791 0101** procedí a revisar el informe final, cuyo título es **INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"



Inga. Floriza Felipa Ávila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA

Universidad de San Carlos de
Guatemala



Facultad de Ingeniería
Unidad de EPS

Guatemala, 23 de septiembre de 2024.
REF.EPS.D.393.09.2024.

Ing. Carlos Gustavo Alonzo
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Alonzo:

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, que fue desarrollado por el estudiante universitario **Joel Estuardo Rodríguez Santos, Registro Académico 201115018 y CUI 2098 80791 0101** quien fue debidamente asesorado por el Msc. Marco Tulio Aldana Prillwitz y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y la Supervisora de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"


Ing. Oscar Argueta Hernández
Director Unidad de EPS



/ra



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 2 de octubre de 2024

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **JOEL ESTUARDO RODRÍGUEZ SANTOS** carné **201115018** y CUI **2098807910101**, titulado: **"Integración del ecosistema informático de la Facultad de Ciencias Jurídicas y Sociales de la Universidad de San Carlos de Guatemala"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación





SIST.LNG.DIRECTOR.114.EICCSS.2024

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área del trabajo de graduación titulado: **Integración del ecosistema informático de la Facultad de Ciencias Jurídicas y Sociales de la Universidad de San Carlos de Guatemala**, presentado por: **Joel Estuardo Rodríguez Santos**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”



Ingeniero Carlos Gustavo Alonzo
DIRECTOR
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, octubre de 2024



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad de Ingeniería

24189101- 24189102

LNG.DECANATO.OIE.672.2024

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **INTEGRACIÓN DEL ECOSISTEMA INFORMÁTICO DE LA FACULTAD DE CIENCIAS JURÍDICAS Y SOCIALES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por: **Joel Estuardo Rodríguez Santos** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. José Francisco Gómez Rivera
Decano a.i.



Guatemala, noviembre de 2024

Para verificar validez de documento ingrese a <https://www.ingenieria.usac.edu.gt/firma-electronica/consultar-documento>

Tipo de documento: Correlativo para orden de impresión Año: 2024 Correlativo: 672 CUI: 2098807910101

Escuelas: Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, - Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS). Postgrado Maestría en Sistemas Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas. Licenciatura en Matemática. Licenciatura en Física. Centro de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad

ACTO QUE DEDICO A:

Mis padres

Por siempre, apoyarme incondicionalmente y a quienes les debo todo lo que tengo en esta vida. Por brindarme una segunda oportunidad de convertirme en un profesional a pesar de tener una edad en la cual ya no tendrían la obligación de apoyar.

Mi hermana

Por ayudarme a adentrarme en la carrera en los inicios, en la cual me sentía perdido y sin muchas esperanzas.

Mi abuela

Por haber sido una gran persona siempre, y particularmente por haberme brindado las herramientas que me ayudaron a conocer lo que realmente me ayudaría a tratar los problemas que tuve en el transcurso de la carrera.
(q. e. p. d.)

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por proporcionarme la oportunidad de poder mejorar como persona y convertirme en un profesional.
Facultad de Ingeniería	Por ser el lugar específico que me proporcionó sus instalaciones y profesionales, que me ayudaron a cultivar y desarrollar mi conocimiento en la profesión.
M.Sc. Marco Aldana	Por su gran ayuda, experiencia y guía en todo el proceso de realización del proyecto.
Lic. Nelson Luc	Por su apoyo y gran disposición en el transcurso del proyecto.
Mis amigos de la Facultad	Edin Montenegro y Javier Alfaro.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN	XIII
OBJETIVOS	XV
INTRODUCCIÓN	XVII
1. FASE DE INVESTIGACIÓN	1
1.1. Antecedentes de la institución	1
1.1.1. Reseña histórica	1
1.1.2. Misión de la empresa	2
1.1.3. Visión de la empresa	2
1.1.4. Servicios que realiza	3
1.2. Descripción de las necesidades	4
1.3. Priorización de las necesidades	4
2. FASE TÉCNICO-PROFESIONAL	7
2.1. Descripción del proyecto	7
2.2. Investigación preliminar para la solución del proyecto	9
2.2.1. Arquitecturas	9
2.2.1.1. Service Oriented Architecture (SOA)	9
2.2.1.2. Enterprise service bus (ESB)	11
2.2.1.3. Microservicios	12
2.2.1.4. Arquitectura clean	13
2.2.2. Patrones de diseño	14

	2.2.2.1.	Modelo vista controlador (MVC).....	14
	2.2.2.2.	Cadena de responsabilidades.....	15
	2.2.3.	Seguridad.....	16
	2.2.3.1.	Json Web Tokens (JWT)	16
	2.2.3.2.	Bycript	18
	2.2.4.	Experiencia del usuario	19
	2.2.5.	Herramientas.....	20
2.3.		Presentación de la solución al proyecto	23
	2.3.1.	Arquitectura SOA.....	23
	2.3.2.	Arquitectura clean.....	23
	2.3.3.	Patrón MVC.....	23
	2.3.4.	Patrón middleware.....	24
	2.3.5.	Lenguaje y herramientas de codificación	24
	2.3.6.	Manejo de sesiones.....	24
	2.3.7.	Replicación y manejo de bases de datos.....	25
	2.3.8.	Arquitectura de la solución	25
	2.3.9.	Pruebas.....	26
	2.3.9.1.	Pruebas unitarias.....	26
	2.3.9.2.	Pruebas de rendimiento.....	27
	2.3.9.3.	Pruebas de integración	27
	2.3.9.4.	Pruebas funcionales	27
	2.3.9.5.	Pruebas de aceptación	28
2.4.		Costos del proyecto.....	30
2.5.		Beneficios del proyecto	31
3.		FASE ENSEÑANZA APRENDIZAJE	33
	3.1.	Capacitación propuesta.....	33
	3.2.	Material elaborado	34
	3.2.1.	Manual de usuario	34

3.2.2.	Manual técnico.....	34
3.2.3.	Vídeo explicativo.....	35
3.2.4.	Repositorio del código y manual md	35
CONCLUSIONES		37
RECOMENDACIONES.....		39
REFERENCIAS		41
APÉNDICES.....		45

ÍNDICE DE ILUSTRACIONES

FIGURAS

Figura 1.	Pirámide de priorización de necesidades.....	6
Figura 2.	Arquitectura de software	26

TABLAS

Tabla 1.	Caso de prueba 1	28
Tabla 2.	Caso de prueba 2	29
Tabla 3.	Caso de prueba 3	29
Tabla 4.	Caso de prueba 4	30
Tabla 5.	Presupuesto del proyecto	31

LISTA DE SÍMBOLOS

Símbolo	Significado
\$	Dólar
Gb	Giga Bite
Mb	Mega Bite

GLOSARIO

AWS	Representa a Amazon Web Services, una colección de servicios en la nube proporcionado por Amazon.com
Backend	Parte lógica de un sitio que se encarga del procesamiento de datos y la lógica del negocio.
Casos de uso (CDU)	Técnica utilizada para la recopilación de requisitos funcionales en un proyecto y desde la perspectiva de un usuario. Es una descripción detalla de las actividades que realizan los usuarios en un sistema.
CDUE	Caso de uso extendido
Diagrama ER	El diagrama de entidad-relación es una herramienta visual que se utiliza para representar un modelo de datos principalmente en entidades y relaciones.
Diagrama actividades	Herramienta visual que es utilizada para representar gráficamente un flujo de control de actividades y procesos dentro de un sistema.
DOM	Document Object Model que hace referencia a la estructura de un documento HTML conformado por etiquetas HTML y sus características.

Framework	Es un conjunto estandarizado de conceptos y prácticas enfocados a resolver una problemática en particular.
Frontend	Parte lógica de un sitio que se encarga de interactuar con el usuario final. Comúnmente es también llamada la interfaz del usuario.
Historias de usuario	Técnica utilizada para la recopilación de requisitos funcionales en un proyecto, al igual que los CDU, desde la perspectiva de un usuario. A diferencia de los CDU, utilizan un lenguaje más natural y son usadas principalmente dentro de metodologías ágiles.
Log	Es un registro secuencial de un archivo o de datos sobre eventos o acontecimientos, y que se guarda típicamente en una base de datos.
Nginx	Es un servidor web/proxy inverso utilizado comúnmente como balanceador de carga.
Nube	Hace referencia al uso de un conjunto de servidores remotos conectados a internet con el fin de realizar alguna actividad informática.
SSL	Secure Sockets Layer lo cual es un protocolo de seguridad de internet que utiliza cifrado entre un sitio web y un navegador web.

SSO	Single Sing On es un procedimiento de autenticación unificado que permite a los usuarios acceder a varios servicios desde un único inicio de sesión.
Token	Es un elemento informático que sustituye a un elemento de datos sensible y lo vuelve un elemento equivalente, pero no sensible.
UI	Expresión que se deriva de <i>user interface</i> , que significa interfaz de usuario. Se refiere a todo lo gráfico que interactúa con el usuario directamente.

RESUMEN

En el presente trabajo se detallan todos los elementos que fueron necesarios para la planificación y desarrollo del proyecto de integración de los sistemas informáticos existentes y en desarrollo de la Facultad de Ciencias Jurídicas y Sociales de la Universidad de San Carlos de Guatemala.

El problema principal se presentó al momento de que los primeros proyectos fueran finalizados y algunos otros estuvieran en el transcurso de su construcción. Se determinó que existiría una gran complejidad para los usuarios al momento de tener la disponibilidad de utilizar los servicios informáticos, ya que estos estaban completamente aislados unos de otros y los usuarios tendrían que crear cuentas separadas en cada uno de ellos a pesar de que algunos sí que tendrían alguna relación en común.

El mayor inconveniente era lograr una solución que congeniara con todos los servicios, con todas sus similitudes y diferencias. Se construyeron tablas con los datos en común y diferencias manejadas por cada servicio, y finalmente se determinó la arquitectura necesaria para resolver el problema, la cual consiste principalmente en una solución SOA orientada a la utilización de la teoría de un ESB adaptado a las necesidades y la reutilización de los mecanismos de tokens ya manejados por los servicios.

La solución logra el cometido de integrar todos los servicios a un mecanismo único de inicio de sesión, creación de cuentas y modificación de datos, siempre manteniendo las mejores prácticas de seguridad y reutilizando servicios ya existentes desde los módulos desarrollados.

OBJETIVOS

General

Desarrollar un sistema que integre el ecosistema informático de la Facultad de Ciencias Jurídicas y Sociales de la Universidad de San Carlos de Guatemala, tomando en cuenta los principios esenciales de una correcta transformación digital.

Específicos

1. Implementar un módulo para el inicio de sesión centralizado utilizando principios personalizados SSO que se adecuen a técnicas utilizadas por las especialidades en desarrollo y que permita a los usuarios acceder a múltiples servicios con una única autenticación.
2. Establecer y administrar permisos de acceso a los distintos módulos mediante la habilitación de las especialidades a demanda por parte del usuario final, asegurando que estos tengan acceso adecuado basado en sus necesidades.
3. Diseñar un único módulo de registro de usuarios que gestione eficientemente la asignación de roles, representando adecuadamente la diversidad de roles requeridos por las distintas especialidades en desarrollo.

INTRODUCCIÓN

En el presente trabajo se detalla la documentación específica del proyecto de Integración del ecosistema informático de la Facultad de Ciencias Jurídicas y Sociales de la USAC; nacido de la necesidad de mejorar la experiencia de usuario en la facultad, ya que las autoridades se percataron de la dificultad que tendrían los usuarios para poder utilizar los servicios dentro de la plataforma actual.

El objetivo principal del proyecto es la centralización de las acciones que tenga que realizar el usuario para poder acceder a todos los sistemas. En esta centralización se estarían incluyendo los manejos de sesiones, la creación de cuentas y la actualización de información de los usuarios. La plataforma proporcionará un espacio ordenado y minimalista para lograr este cometido.

Al tratarse de más de 10 soluciones en desarrollo dentro de la facultad, se encontró con varios desafíos, el principal: la unificación o estandarización de datos a utilizar. Varios proyectos se encuentran en fase final por lo que la modificación de estos se tornó como una opción poco viable, y lo que finalmente se decidió es recurrir a una solución que pueda adaptarse a todos.

Dentro de la documentación se describe la solución final, así como la forma en que los sistemas de seguridad, manejo de sesiones, modificación de datos y creación de cuentas son reutilizados desde cada una de las soluciones ya establecidas, para evitar modificaciones abrumadoras que terminen por cambiar la lógica de los sistemas en desarrollo y en fase de finalización.

1. FASE DE INVESTIGACIÓN

1.1. Antecedentes de la institución

En esta fase se presenta y detalla toda la información relevante sobre la Facultad de Ciencias Jurídicas y Sociales, incluyendo su historia, antecedentes, valores y funciones.

1.1.1. Reseña histórica

Alrededor de 60 años antes de la fundación de la universidad de San Carlos, en 1620, ya se habían asignado recursos para la instauración de la primera cátedra en Derecho Canónico en el colegio Santo Tomás de Aquino que había sido fundado en 1562 como uno de los primeros centros de enseñanza en leyes, así como también en arte, filosofía, gramática, teología, entre otros.

La elección de estas cátedras serviría como inspiración de los planes iniciales para la fundación de la primera universidad de Guatemala, en donde se incluían una cátedra prima de Leyes y una prima de Cánones. Fue hasta 2 meses después de su fundación que se impartieron los primeros cursos relacionados, el 10 de febrero de 1681, los de Derechos Canónicos y Civiles.

El primer acto de leyes fue sustentado por el Bachiller Ignacio de Mármol el 20 de julio de 1681, dedicado al presidente Gobernador y Capitán General de las Provincias de Guatemala, licenciado Don Lepe de Sierra y Osorio. Desde entonces, la facultad experimentó diversos cambios y reformas, destacando una de las primeras importantes, la "Reforma Universitaria de Goicoechea" a finales

del siglo XVIII y principios del XIX, que introdujo nuevas materias y amplió el estudio de las Ciencias Jurídicas (Ortiz et al., 2007).

Actualmente, la Facultad de Ciencias Jurídicas y Sociales otorga los títulos profesionales de Abogado y Notario, en el grado académico de Licenciado en Ciencias Jurídicas y Sociales. La duración de la carrera es de 5 años, la cual está organizada en un plan de estudios semestral, con cinco cursos por ciclo (Ortiz et al., 2007).

1.1.2. Misión de la empresa

La misión de la institución es la búsqueda y obtención del conocimiento científico relacionado con el Derecho (Facultad de Ciencias Jurídicas y Sociales, 2021).

También constituye su misión la formación de profesionales del Derecho, con conocimientos científicos sobre el mismo, competencia en el desempeño de su profesión y portador de valores morales (Facultad de Ciencias Jurídicas y Sociales, 2021).

Por último, su misión se complementa con el servicio a la comunidad y su proyección cultural y deportiva. Todo lo anterior se apoya en decisiones políticas correctas fundamentadas en la ley y en una administración eficiente y eficaz (Facultad de Ciencias Jurídicas y Sociales, 2021).

1.1.3. Visión de la empresa

La visión de la Facultad de Ciencias Jurídicas y Sociales es constituir el centro de investigación jurídica de donde emane el conocimiento del Derecho y

se disemine a nivel nacional e internacional (Facultad de Ciencias Jurídicas y Sociales, 2021).

Los egresados de la Facultad de Ciencias Jurídicas y Sociales son profesionales del Derecho con alto conocimiento científico, competencia jurídica a nivel nacional e internacional, valores éticos y morales y fortalecedores, desde su respectivo quehacer, del Estado de Derecho (Facultad de Ciencias Jurídicas y Sociales, 2021).

La Facultad de Ciencias Jurídicas y Sociales sirve eficiente y eficazmente a la sociedad guatemalteca mediante el servicio legal a personas de escasos recursos; promueve asimismo la difusión de la cultura y la práctica del deporte (Facultad de Ciencias Jurídicas y Sociales, 2021).

1.1.4. Servicios que realiza

La Facultad de Ciencias Jurídicas y Sociales de la Universidad de San Carlos de Guatemala tiene como función general la búsqueda y obtención del conocimiento científico relacionado con el Derecho (Ortiz et al., 2007).

Así también constituye la formación de profesionales del Derecho, con conocimientos científicos sobre el mismo, competencia en el desempeño de su profesión y portador de valores morales. Se complementa con el servicio a la comunidad y su proyección cultural y deportiva (Ortiz et al., 2007).

Todo lo anterior se apoya en decisiones políticas correctas fundamentadas en la ley y en una administración eficiente y eficaz (Ortiz et al., 2007).

1.2. Descripción de las necesidades

El día de la reunión inicial con el encargado del proyecto, el Lic. Nelson Luc, se comentó que en la nueva administración se estaba preparando un cambio de sistemas informáticos, ya que los actuales se han quedado obsoletos y no tienen el rendimiento necesario para rendir con los nuevos requerimientos de esta nueva administración.

Se planeó construir diversos sistemas informativos para poder llevar a cabo todas las actividades administrativas disponibles; sin embargo, en el transcurso de los desarrollos se percataron que sería complicado y tedioso para los usuarios estar transitando entre servicios con, posiblemente, diferentes credenciales y diferentes sesiones, por lo que se decidió preparar un nuevo proyecto.

La necesidad principal que nació de este análisis es la de la unificación de todos los nuevos servicios informáticos, el manejo de sesiones, credenciales únicas y la consistencia de la información entre todos los servicios. Esta información debe poder ser manejada independientemente de los roles de los usuarios, aunque en este proyecto se tiene como prioridad a los roles de los estudiantes.

1.3. Priorización de las necesidades

Las necesidades han sido priorizadas en 3 grupos: de lógica, diseño y actividades complementarias y de visualización y reportería.

La Facultad tiene como asignación primordial el manejo de sesiones únicas en todos los servicios que se encuentran dentro del ecosistema

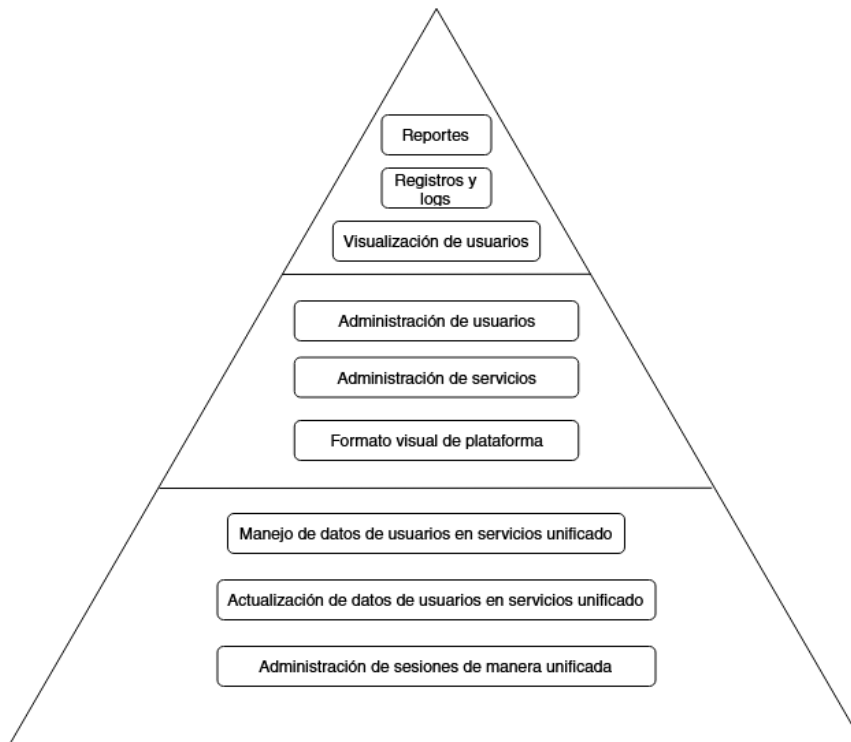
informático de la institución. Esto conlleva a que también los datos de los usuarios tienen que ser administrados obligatoriamente con las sesiones y presentar un conjunto de información coherente y conectada. Todos los cambios en cualquier servicio deben de replicarse satisfactoriamente en el resto de servicios habilitados.

En el segundo nivel de prioridades, se encuentran actividades administrativas, opciones que los usuarios administradores tendrán para manejar a los usuarios y la disponibilidad de los servicios. También se encuentra el formato y diseño de la plataforma, el cual debe de ser adaptado al formato actual manejado por la facultad de manera institucional por la actual administración.

En el último nivel de prioridades se encuentran los reportes, los registros y otras opciones para los administradores en el manejo de usuarios. Los reportes son importantes para visualizar el resultado final. Sin embargo, es necesario primero tener lista toda la lógica y la plataforma debe de funcionar totalmente para que se refleje en los reportes y en los registros. Las opciones extra para los administradores también son apartados en los cuales tendrán actividades disponibles extra que no influyen significativamente en el flujo más importante de la plataforma, el manejo de sesiones y servicios.

Figura 1.

Pirámide de priorización de necesidades



Nota. Visualización de prioridades en el proyecto. Elaboración propia, realizado en draw.io.

2. FASE TÉCNICO-PROFESIONAL

En el presente capítulo se enumeran las características del proyecto y la solución aplicada para resolverlo. Se describen las tecnologías utilizadas, así como los patrones de diseño aplicados.

2.1. Descripción del proyecto

El día de la reunión inicial con el encargado del proyecto, el Lic. Nelson Luc, se comentó que en la nueva administración se estaba preparando un cambio de sistemas informáticos, ya que los actuales se han quedado obsoletos y no tienen el rendimiento necesario para rendir con los nuevos requerimientos de esta nueva administración.

Se planeó construir diversos sistemas informativos para poder llevar a cabo todas las actividades administrativas disponibles; sin embargo, en el transcurso de los desarrollos se percataron que sería complicado y tedioso para los usuarios estar transitando entre servicios con, posiblemente, diferentes credenciales y diferentes sesiones, por lo que se decidió preparar un nuevo proyecto.

Este nuevo proyecto se encargaría de unificar todos los nuevos servicios informáticos y así manejar sesiones y credenciales únicas. Este problema finalmente consiste en resolver el tema de la unificación de sesiones, independientemente de los roles que estos tengan, aunque se tiene como primordial a los roles de los estudiantes en esta primera etapa.

La plataforma también debe permitir la creación de cuentas, modificación de datos y de estados de cuentas; replicación de datos y eliminación de silos de información que fueron generados por la falta de planificación en el resto de los servicios.

Otras funcionalidades que obtendrán los roles administradores serán las de la visualización y construcción de reportes, gráficas y estadísticas de los usuarios dentro del sistema y dentro de cada servicio, así como también la posibilidad de verificar el funcionamiento del servidor por medio de “logs” que se cargan y actualizan automáticamente en el apartado de “logs”.

Todo el trabajo se realiza de lunes a viernes, con 4 horas diarias de aplicación al proyecto, utilizando un ambiente de desarrollo local en el software de visual code, configurado con un servidor nginx para trabajar bajo el mismo dominio y poder realizar las pruebas en conjunto con el resto de los servicios. Las reuniones con los representantes y el asesor de la institución se realizan por medio de conferencias en Meet para presentar avances, ver problemas o modificaciones tanto del proyecto de integración como del resto de proyectos.

Luego de diseñar y de construir el sistema, se debe también configurar la plataforma en conjunto con el resto de los servicios para poder trabajar bajo el dominio que utiliza la institución para servir su sitio oficial, el cual también se encuentra firmado con certificados SSL, por lo que estos también deben ser utilizados por la plataforma y el resto de los servicios. Las pruebas se realizan en un servidor alternativo dentro de la nube de AWS.

2.2. Investigación preliminar para la solución del proyecto

Para desarrollar una solución de software eficaz ante el problema planteado, es esencial identificar las herramientas más adecuadas. Por lo tanto, la investigación sobre arquitecturas, patrones de diseño y librerías se convierte en un paso fundamental para el avance del proyecto. A continuación, se detalla la teoría que sustenta la solución propuesta.

2.2.1. Arquitecturas

Las arquitecturas son estructuras previamente planificadas que organizan los componentes e interacciones de un sistema. Definen cómo se implementará la solución de software, asegurando escalabilidad y mantenibilidad. Su correcta selección es clave para cumplir con los requisitos del proyecto.

2.2.1.1. Service Oriented Architecture (SOA)

SOA es una estrategia arquitectónica que permite a las organizaciones crear sistemas de software adaptables y escalables utilizando una presentación desacoplada de servicios empresariales (Endrei et al, 2004).

Al utilizar esta arquitectura, es posible la reutilización de servicios, lo que también reduce la duplicación de recursos y los costos, además de aumentar la flexibilidad y la capacidad de respuesta de la solución a las necesidades cambiantes del mundo.

SOA también facilita la integración del sistema y la gestión de la complejidad al proporcionar una capa de abstracción y especificaciones de servicio bien definidas.

La arquitectura SOA tiene cuatro elementos principales, según Endrei, et al. (2004), que trabajan juntos para aumentar la flexibilidad y la eficiencia de la gestión de servicios empresariales. En primer lugar, se encuentran los servicios, que son módulos especializados que proporcionan funciones específicas y se pueden combinar según las necesidades del negocio. Otro elemento clave es la colaboración, que se refiere a la interacción entre proveedores de servicios y consumidores siguiendo un proceso de "buscar, conectar y llamar".

El siguiente elemento por resaltar es el registro de servicios, el cual almacena información importante sobre los servicios disponibles, como contratos e identificadores de ubicación, lo que facilita la búsqueda y el acceso a esos servicios. Finalmente, se encuentra la gestión de servicios, que se refiere a aspectos críticos como la seguridad, el monitoreo, la escalabilidad y la disponibilidad de los servicios para garantizar su integridad y eficiencia. Juntos, estos elementos forman el núcleo de SOA, que buscan satisfacer las necesidades de un entorno empresarial cambiante.

- Componentes y servicios

Un componente es una pieza de código ejecutable que se puede reutilizar en múltiples áreas de una aplicación para aislar una funcionalidad específica. Por lo general, incluye reglas lógicas y comerciales que corresponden a las entidades que el componente pretende representar.

Un servicio es una pieza de software que proporciona cierta funcionalidad a través de una interfaz bien definida y que es completamente independiente de su implementación (Huaca, 2011).

Los servicios se utilizan para exponer la vista externa del sistema y son más generales y detallados que los componentes. Las aplicaciones y los servicios interactúan entre sí mediante un modelo de comunicación basado en mensajes, y los servicios están diseñados para ser expuestos y consumidos individualmente (Endrei et al, 2004).

SOA proporciona funciones estructuradas como servicios que son independientes y pueden implementarse mediante componentes de software que encapsulan determinadas funciones. Estos componentes se comunican entre sí a través de interfaces específicas y se adhieren a principios de diseño arquitectónico como la reutilización y la modularización en un esfuerzo por aprovechar la flexibilidad y mejorar la interoperabilidad del sistema.

2.2.1.2. Enterprise service bus (ESB)

El bus de servicios empresariales (ESB) es, según Endrei, et al. (2004), una infraestructura de software diseñada principalmente para proporcionar virtualización de recursos empresariales. Como resultado, la lógica empresarial se puede crear y gestionar sin depender de la infraestructura, red o la prestación de esos servicios empresariales.

El ESB modela los recursos como servicios que ofrecen una o varias operaciones comerciales. Según Endrei, et al. (2004), para la implementación de un ESB es necesario un conjunto integrado de servicios de middleware que soporten varios diseños arquitectónicos: arquitecturas orientadas a servicios, arquitecturas basadas en mensajes y arquitecturas basadas en eventos.

Entre las comunicaciones de los consumidores y los proveedores de servicios, el ESB convierte los mensajes, dirige las solicitudes al proveedor

adecuado y modifica los protocolos de transporte. El ESB también utiliza un registro de servicios para realizar un seguimiento de las descripciones de los servicios y otra información que los proveedores de servicios pueden utilizar para anunciar sus servicios.

2.2.1.3. Microservicios

Según Daya, et al. (2015) los microservicios son una arquitectura de software que ofrecen mayor flexibilidad, escalabilidad y mantenibilidad que las arquitecturas monolíticas tradicionales y mayor granularidad e independencia que SOA, porque son componentes de software pequeños e independientes que pueden comunicarse entre sí a través de API.

- **Beneficios**

Entre los beneficios resaltados por Daya, et al. (2015) se encuentra:

- Mejoran la disponibilidad de los servicios y elimina cuellos de botella
- Agilizan el desarrollo, la actualización y el despliegue de los servicios.
- Reduce los costos al eliminar la duplicidad de servicios
- Son independientes de los lenguajes de programación o herramientas
- Mejora la colaboración e integración de los equipos de trabajo al subdividirse en desarrollos más específicos

Mientras que las desventajas más importantes explicadas por Endrei, et al. (2004) son:

- Aumento de complejidad al tener que manejar más componentes móviles del sistema
- Aumenta el uso de recursos ya que la arquitectura necesita manejar múltiples procesos independientes
- Puede llegar a requerir una infraestructura de monitoreo y operaciones de alta calidad
- Es necesario manejar tamaños adecuados para los microservicios y que estos no lleguen a ser demasiado grandes ya que puede llegar a ser difíciles para trabajar.
- Aumenta el tiempo de revisión de código al utilizar servicios muy pequeños e independientes, por lo que en el peor de los casos se tendrían que revisar varios componentes.

2.2.1.4. Arquitectura clean

Fernández (2016) indica que este tipo de arquitectura es “un conjunto de principios cuya finalidad es ocultar los detalles de implementación a la lógica de dominio de la aplicación. De esta manera mantenemos aislada la lógica, consiguiendo tener una lógica mucho más mantenible y escalable en el tiempo” (párr. 1).

Este enfoque sigue los fundamentos de otras arquitecturas bien conocidas, como la hexagonal o la cebolla. Su objetivo primordial es separar la lógica en capas independientes, lo que facilita su mantenimiento y permite que cada parte del sistema evolucione de manera autónoma sin afectar a otras capas. Una ventaja importante es esta misma división que permite que las capas sean más comprensibles y simples de modificar o reemplazar.

Un esquema bien conocido de esta arquitectura incluye capas como la interfaz de usuario (UI), controladores, casos de uso y entidades. Sin embargo, no es una guía absolutista, permite la adición de otras capas según las necesidades específicas del proyecto. Las entidades representan el estado y las reglas del negocio, los casos de uso reflejan las acciones que los usuarios desean realizar, los controladores gestionan estas acciones y la UI finalmente interactúa con el sistema a través de los métodos proporcionados por los controladores (Cea, 2021).

2.2.2. Patrones de diseño

Los patrones de diseño son soluciones probadas para problemas recurrentes en el diseño de soluciones de software. Una de sus grandes ventajas es que han sido utilizadas con éxito en situaciones específicas, lo que las convierte en una guía confiable que puede adaptarse a problemas similares.

2.2.2.1. Modelo vista controlador (MVC)

Este concepto representa un modelo arquitectónico que divide la organización de una aplicación en tres elementos principales lógicos e independientes: modelos, vistas y controladores. Cada uno de estos componentes está diseñado para manejar un aspecto específico y diferente dentro del desarrollo (Maboud, 2020).

Este diseño es utilizado frecuentemente en el desarrollo de aplicaciones web, debido a que las interacciones de los usuarios en estas generan solicitudes que un servidor debe procesar para conocer un resultado y presentarlo al cliente. Estas solicitudes a menudo implican la manipulación de datos que el servidor guarda en una capa de almacenamiento de datos, antes de conocer la respuesta.

Las vistas son interfaces que interactúan directamente con los usuarios, muestran el HTML renderizado y, a través de las acciones de los usuarios, generan peticiones que son enviadas al servidor y procesadas por medio de controladores.

Los controladores son los encargados de ejecutar todas las acciones necesarias para resolver las peticiones que reciben desde las vistas. Finalmente, los modelos son interfaces construidas para permitir la interacción de los controladores con la capa de datos y así lograr modificar la información almacenada si es necesario.

Uno de los principales beneficios de este patrón es su capacidad para proporcionar flexibilidad a las aplicaciones, ya que al dividir la aplicación en tres partes independientes se facilita su comprensión y gestión. Esto también contribuye al mejoramiento del mantenimiento y la escalabilidad del sistema, ya que cada parte puede ser modificada sin tener efectos significativos sobre las otras.

2.2.2.2. Cadena de responsabilidades

La cadena de responsabilidad es un patrón de diseño que permite que una solicitud pase a lo largo de una serie de objetos, hasta que uno finalmente pueda manejar la solicitud o la cadena termine (Yüksel, 2023).

Este patrón se enfoca en la interconexión de varios objetos que tienen como función realizar una acción específica a una petición entrante en el sistema y reportar el resultado, ya sea como satisfactorio al siguiente objeto o como error y respuesta final. Se usan regularmente para temas de autenticación, verificación

de datos o permisos, seguridad, entre otros. Tienen la ventaja de poder realizar acciones en una petición antes de que esta sea procesada por el servidor.

Se puede determinar que son desacoplables ya que no dependen de otros objetos para ser utilizados, tienen un inicio y fin específico; son reutilizables y pueden ser aplicables de manera individual a distintas peticiones o incluso a rutas específicas en la aplicación, y son flexibles ya que se pueden agregar o quitar sin afectar el comportamiento de otros objetos o de la cadena misma.

2.2.3. Seguridad

En la seguridad de la aplicación se toman en cuenta los mecanismos que se utilizan para proteger los datos de los usuarios y las defensas que evitan ingresos no autorizados.

2.2.3.1. Json Web Tokens (JWT)

Según lo indicado en su documentación oficial, JWT.IO (s. f.), JSON Web Token (JWT) es un estándar de código abierto que se encuentra definido en el RFC 7519. Su función principal radica en proporcionar un método eficaz y autónomo para la transferencia segura de datos entre distintas partes. La información que se transmite, presentada en formato JSON, goza de confiabilidad y autenticidad gracias a su firma digital. Estos tokens pueden ser autenticados utilizando una clave, a través de algoritmos como HMAC, o mediante un par de claves público/privado, utilizando RSA o ECDSA.

El uso de JSON Web Tokens (JWT) es relevante en diversas situaciones, siendo la autorización una de las más comunes. Una vez que un usuario inicia sesión, cada solicitud posterior incorporará un JWT que otorga al usuario la

posibilidad de acceder a rutas, servicios y recursos específicos habilitados por el token. Además, JWT es una herramienta utilizada para el intercambio seguro de información entre partes. Dado que los tokens se pueden firmar, se puede establecer un nivel adicional de confianza verificando que la transmisión del emisor sea auténtica.

- Estructura

En su forma compacta, los JWT constan de tres partes: Encabezado, Payload y Firma.

El encabezado normalmente consta de dos elementos: el tipo de token, que es JWT, y el algoritmo de firma utilizado, como HMAC SHA256 o RSA.

El payload contiene afirmaciones sobre una entidad, generalmente un usuario, así como información adicional. En el “JWT Handbook” de Peyrott (2016) se indica que son posibles tres tipos de reclamos: registrados, que son un conjunto predefinido recomendado por compatibilidad y aseveraciones útiles; público, que puede ser definido por los usuarios de JWT; y privado, que son afirmaciones personalizadas creadas para compartir información entre partes y que no son afirmaciones públicas o registradas.

La firma está constituida por el payload y el encabezado codificado en base64. Se produce mediante el uso de una clave secreta y el algoritmo de firma especificado en el encabezado para combinarlos. Esta firma es necesaria para confirmar la identidad del remitente y garantizar que el mensaje se mantenga sin alteraciones durante la transmisión.

2.2.3.2. Bcrypt

Es un algoritmo de hash que mapea datos de cualquier tamaño a una salida de tamaño fijo. Se utilizan en sistemas informáticos principalmente para verificar la integridad de datos, como en contraseñas. Es un algoritmo construido a propósito para ser lento y resistente a GPU, con el fin de ser difícil de “crackear”. Para lograr este cometido utiliza hashes Blowfish junto con un “salt” y un costo (Grigutyte, 2023).

El “salt” es denominado un hash con sal que tiene como fin agregar una cadena aleatoria a la entrada antes de “hashearla”. El costo indica cuántas iteraciones de procesamiento se necesitan para calcular cada hash, haciendo que este sea más tardado de calcular.

- **Funcionamiento**

Grigutyte (2023) explica que el proceso inicia generando un valor aleatorio de 16 bytes (salt) que se obtiene utilizando un generador de números aleatorios. Luego, este salt se utiliza como prefijo y se añade a la entrada, que el usuario ingresa usando un valor ASCII. La combinación se procesa mediante el algoritmo hash Blowfish, produciendo un valor hash de 64 bits. Esta función de entrada ya recibe la información, el salt y el costo (Arias, 2021).

Más adelante se activa el bucle iterativo y los hashes de Blowfish resultantes se someten a un proceso de hash iterativo. El número de rondas en este ciclo está determinado por el costo seleccionado, que generalmente oscila entre 4 y 31. En cada ronda, se vuelven a combinar resultados anteriores, lo que aumenta la complejidad del proceso. Cuando se completan todas las rondas, se obtiene un hash final de 192 bits, que es una combinación de tres hashes de 64

bits. El resultado es un hash Bcrypt final que se puede almacenar y utilizar en futuras confirmaciones de contraseña.

2.2.4. Experiencia del usuario

Uno de los principios básicos del diseño de la experiencia del usuario, según Krug (2014), es que el esfuerzo mental asociado con el uso del sistema debe ser mínimo. La primera ley de la usabilidad es: No hagas pensar al usuario. La interfaz debe explicarse por sí misma y no requerir mayor esfuerzo por comprenderla, lo que puede ser especialmente importante cuando se construye una plataforma centralizada que proporciona acceso a múltiples aplicaciones o servicios.

Como punto de entrada a múltiples sistemas, es fundamental que el proceso sea simple e intuitivo, como un inicio de sesión centralizado o un registro global de usuarios. Krug (2014) recomienda utilizar diferentes métodos para tener en cuenta el comportamiento de los usuarios reales, como:

- Facilita la navegación visual rápida por opciones sin contenido saturado.
- Mejorar el proceso de recopilación de información.
- Definir claramente las áreas y partes de la plataforma
- Evite la confusa ambigüedad del lenguaje.
- Reduzca las distracciones visuales para que los usuarios puedan concentrarse en las tareas.
- Utilizar convenciones establecidas para elementos clave del mercado, como los formularios de información necesarios para iniciar sesión y registrarse.

La presencia de múltiples puntos de acceso en diferentes ubicaciones y con diferentes credenciales hace que la plataforma sea compleja y complicada para los usuarios, mientras que, simplificando la plataforma, según el supuesto de Krug (2014), se puede lograr un acceso más estandarizado y optimizado a los servicios, permitiendo a los usuarios utilizarla sin ningún esfuerzo adicional y sentirse cómodos y confiados al ingresar a una opción que maneja convenciones conocidas.

2.2.5. Herramientas

Herramientas de software que fueron seleccionadas para el desarrollo de la solución.

- NodeJS

Es una tecnología que está construida sobre el motor de JavaScript V8 de Chrome y utiliza programación asíncrona de entrada/salida, lo que la hace muy eficiente y rápida al momento de realizar operaciones de E/S como hacer requests en la red o acceder a bases de datos. Es una de las tecnologías más utilizadas en el mundo, con millones de soluciones utilizándola, y tiene una gran comunidad detrás que puede ser favorable al momento de encontrar y resolver problemas.

Dziuba (2023) resalta que entre las ventajas más importantes se encuentran: que es una solución óptima en relación con el costo-beneficio; es muy escalable; tiene una gran comunidad de las mejores para los dispositivos móviles; está muy bien adaptada para el desarrollo de front-end y back-end; gran compatibilidad cross-platform, entre otros.

- ExpressJS

Es un framework de nodejs que se presenta como minimalista y flexible, que tiene como fin proveer al desarrollador un conjunto de herramientas robustas para el desarrollo web y móvil. Varios de los frameworks más populares están contruidos con base a ExpressJS, como NestJS, Blueprint, FoalTS, entre otros. Con esta herramienta se construyen los servidores que sostienen a todas las aplicaciones.

Este framework aporta una gran variedad de beneficios, por lo cual es una de las herramientas más seleccionadas y mejor posicionadas en el mercado. Estos son: simplicidad, flexibilidad, modularidad, comunidad robusta, escalabilidad, rendimiento, funcionalidad multiplataforma, entre otros (Taha, 2023).

- ReactJS

Framework desarrollado por Facebook y utilizado específicamente para el desarrollo web. Es declarativo lo cual facilita el desarrollo y construcción de interfaces para el usuario que a su vez están conformadas por componentes individuales. Con su modo de renderización por medio de DOM virtual minimiza las operaciones de actualización de componente haciéndolo una solución de alto rendimiento. Estaba basado en uno de los lenguajes más populares de la actualidad, javascript/typescrit, y es junto a nodejs uno de los frameworks más utilizados en el mundo.

- Docker

Es una plataforma para el manejo de contenedores que permite acelerar la entrega de aplicaciones seguras las cuales contendrán todas las dependencias necesarias para poder ejecutarse. Los contenedores son uno de los pilares de la arquitectura de los microservicios y son de gran ayuda para proyectos grandes en donde se tienen muchas soluciones con tecnologías distintas. Ayuda a los desarrolladores a desligarse de las dependencias y a poder utilizar e integrar las soluciones en otros proyectos.

- Github

Es solución para el control de versiones más popular en el mundo y que permite tener control total del código a través de todo el desarrollo. Posee una gran comunidad, debido a su popularidad. Permite proyectos tanto públicos como privados; permite la colaboración entre usuarios y proporcionar herramientas interesantes como copilot u organizaciones que permiten mantener diversos repositorios en un mismo espacio, haciendo posible que un proyecto de integración pueda entregar también los repositorios de los servicios individuales.

- Mysql

Solución de base de datos que proporciona gran escalabilidad, rendimiento, alta disponibilidad, buena seguridad, flexibilidad, facilidad de uso y popularidad al ser una de las bases de datos más utilizadas y mejor respaldados debido a que actualmente es mantenida por la empresa de ORACLE. Al ser software libre también se evitan los costos extra por licencias y está integrada en los servicios proporcionados por RDS en AWS.

2.3. Presentación de la solución al proyecto

Arquitecturas, modelos y patrones de diseño utilizados para la construcción de la solución de software. Manejo de sesiones, técnica de administración y replicación de datos utilizados en el proyecto.

2.3.1. Arquitectura SOA

La arquitectura seleccionada para la solución es SOA utilizando específicamente conceptos del ESB ya que se adapta bien a los requisitos del proyecto. Es una arquitectura que debe utilizar virtualización de servicios, transformar de datos, rutear peticiones e información, convertir información, entre otros. La principal característica por utilizar es la adaptabilidad, una exigencia del proyecto al tener un número significativo de servicios que manejan datos distintos y que deben de ser adaptados para poder ser utilizados en la plataforma y en otros servicios.

2.3.2. Arquitectura clean

Esta arquitectura esta aplicada y adaptada al frontend ya que ayuda a mejorar la organización del proyecto en capas independientes, y sobre todo ayuda a la implementación de características como el lazy loading para enriquecer la carga de elementos, la navegación del usuario y el trabajo en los servidores.

2.3.3. Patrón MVC

Se decide implementar este patrón debido a la naturaleza del proyecto, en el cual se espera a un usuario que realice acciones que desemboquen en

solicitudes hacia los servidores para el retorno de resultados de operaciones realizadas por controladores utilizando también una capa de datos y de esa manera cumplir con el ciclo que se adapta al funcionamiento que modela este patrón.

2.3.4. Patrón middleware

Se selecciona este último patrón para ser implementado en el servidor ya que se espera tener varias capas que realicen ciertas operaciones a las peticiones antes de poder finalmente ser evaluadas y solucionadas por los controladores. Entre estas capas se pueden mencionar las que se encargaran de verificar datos correctos, estado de sesiones, crear logs, verificación de credenciales y permisos, entre otros.

2.3.5. Lenguaje y herramientas de codificación

El lenguaje seleccionado es javascript por ser un lenguaje enfocado principalmente al desarrollo web; por la cantidad de funcionalidades que puedo ofrecer a una web; por su capacidad de trabajar del lado del cliente; por la posibilidad que tiene de trabajar en la mayoría de los navegadores más utilizados; por su gran comunidad y basta documentación; por ser uno de los lenguajes más utilizados, mejor mantenidos y con gran cantidad de librerías; por ser el núcleo de herramientas de desarrollo como React o Angular; por poder obtener todos los beneficios de una arquitectura web basada en componentes, entre otros.

2.3.6. Manejo de sesiones

En el estudio previo de los servicios en desarrollo se observó que todos tenían como factor común la utilización de la metodología JWT y almacenamiento

local en el cliente para la construcción y mantenimiento de las sesiones de los usuarios, por lo que se decidió continuar con la misma dinámica y crear las configuraciones y los datos necesarios para abrir las sesiones en los servicios habilitados para cada usuario.

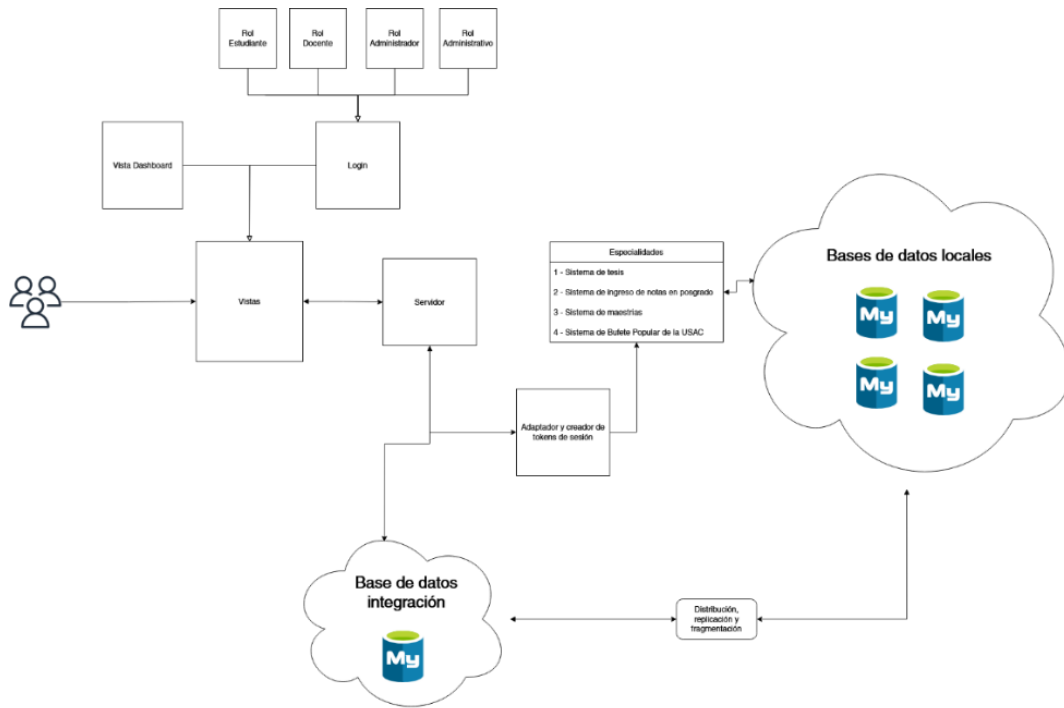
2.3.7. Replicación y manejo de bases de datos

Se decidió utilizar la replicación de base de datos a nivel interno tomando principalmente ventaja del futuro despliegue en una instancia de RDS. Debido a que las tablas de usuarios son en cierta medida diferentes no se pudo realizar una replicación convencional en mysql, por lo que se decidió aplicarla utilizando conexiones entre las bases dentro de la misma instancia para eliminar silos de información y mantener todos los datos actualizados tanto en la base principal como en las copias locales de los servicios y que estas últimas son necesarias para el correcto funcionamiento de estos servicios.

2.3.8. Arquitectura de la solución

Arquitectura basada en una orientación a servicios con un componente bus de servicio empresarial que comunica al resto de servicios integrados, tanto los internos como los externos.

Figura 2.
Arquitectura de software



Nota. Diagrama de la arquitectura construida para la solución de software. Elaboración propia, realizado con draw.io

2.3.9. Pruebas

Las pruebas abarcan temas como la verificación del correcto funcionamiento de los métodos, la toma de datos, funcionamiento de las tecnologías de seguridad o el rendimiento de la solución.

2.3.9.1. Pruebas unitarias

Son las pruebas más cercanas a la fuente de la aplicación realizadas al proyecto. Su finalidad es verificar el correcto funcionamiento de los métodos de

la aplicación. Estas fueron ejecutadas en las funciones relacionadas a la creación de usuarios, actualización de datos de usuarios, creación de tokens de acceso y sesión, permisos, actualización de estados de usuarios y obtención de datos.

2.3.9.2. Pruebas de rendimiento

Se seleccionaron las pruebas de estrés para verificar el comportamiento de la plataforma con altas cargas de trabajo; observar la velocidad de respuesta, estabilidad y precisión de resultados. El tipo de pruebas de rendimiento realizadas fueron: de aplicación, en búsqueda de fallos de conexión o cuellos de botella, y transaccionales, para verificar el correcto intercambio de información.

2.3.9.3. Pruebas de integración

Tipo de pruebas destinadas a verificar el correcto funcionamiento de aplicación complejas y que interactúan u operación con diversos módulos o servicios.

Se verificaron las correctas conexiones entre los servicios internos y externos de cada aplicación que interviene en la plataforma. Reutilizando rutas de los servicios externos se verificó que las respuestas necesarias fueran exitosas, en los procesos de creación y actualización de información.

2.3.9.4. Pruebas funcionales

Destinadas a confirmar el correcto funcionamiento de la aplicación según los requisitos iniciales de software. Se ejecutaron los diversos recorridos posibles en la plataforma para resolver las funcionalidades requeridas como: la creación de cuentas, inicio de sesión, recuperación de contraseña, activación de servicios

externos, modificación de datos, recuperación y obtención de datos y el manejo de datos de sesión.

2.3.9.5. Pruebas de aceptación

Tipo de pruebas funcionales en la cual los usuarios finales verifican que el sistema funcione y cumpla con lo previsto. Para verificar el éxito o fracaso de estas pruebas se desarrollaron 4 casos que fueron reproducidos en el servidor de pruebas por una muestra de usuarios estudiantes y administrativos de la facultad de derecho, los casos fueron los siguientes:

Tabla 1.

Caso de prueba 1

Descripción:	Creación de cuenta estudiante en la plataforma
Pasos de prueba:	1) Ingresar a la plataforma y seleccionar crear cuenta 2) Seleccionar el rol de estudiante 3) Rellenar el formulario con los datos del estudiante 4) Confirmar la creación de cuenta 5) Verificar correo y copiar el código de verificación 6) Verificar la cuenta en verificación de cuentas
Resultado esperado:	Creación de cuenta exitoso
Estado:	Aprobado

Nota. Descripción del caso de prueba 1. Elaboración propia, realizado con Excel.

Tabla 2.*Caso de prueba 2*

Descripción:	Inicio de sesión en plataforma de integración
Pasos de prueba:	1) Ingresar a la plataforma 2) Ingresar datos de usuario 3) Enviar formulario para inicio de sesión
Resultado esperado:	Inicio de sesión exitoso
Estado:	Aprobado

Nota. Descripción del caso de prueba 2. Elaboración propia, realizado con Excel.

Tabla 3.*Caso de prueba 3*

Descripción:	Recuperación de contraseña
Pasos de prueba:	1) Ingresar a la plataforma 2) Seleccionar recuperación de contraseña 3) Ingresar datos de usuario y generar código de verificación 4) Revisar correo y copiar código de verificación 5) Ingresar código de verificación y enviar el formulario 6) Ingresar y confirmar nueva contraseña 7) Enviar formulario y aceptar el cambio de contraseña 8) Iniciar sesión con la nueva contraseña
Resultado esperado:	Recuperación de contraseña exitosa
Estado:	Aprobado

Nota. Descripción del caso de prueba 3. Elaboración propia, realizado con Excel.

Tabla 4.*Caso de prueba 4*

Descripción:	Activación de servicio de tesis de prueba
Pasos de prueba:	<ol style="list-style-type: none">1) Ingresar a la plataforma2) Iniciar sesión3) Seleccionar la opción de activar en servicio de tesis4) Confirmar la activación5) Seleccionar la opción de abrir6) Seleccionar la opción de perfil7) Verificar que los datos son correctos8) Cerrar sesión desde tesis9) Regresar a la plataforma y verificar que la sesión esté finalizada.
Resultado esperado:	Activación de servicio de tesis de prueba exitosa
Estado:	Aprobado

Nota. Descripción del caso de prueba 4. Elaboración propia, realizado con Excel.

2.4. Costos del proyecto

Presupuesto realizado para un proyecto de 6 meses con la utilización de servicios en la nube de Amazon, honorarios de un ingeniero asesor, servicios y bienes requeridos por el proyecto y el desarrollo.

Tabla 5.*Presupuesto del proyecto*

Presupuesto de proyecto de 131 días (6 meses)				
Rubro	Unidad	Cantidad (Semestre)	Precio unitario. (Q)	Subtotal (Q)
A. Recursos humanos				
Honorarios	Hora	524	100.61	52,719.64
Asesor de ECYS	Hora	60	386.95	23,217.00
B. Bienes				
Laptop	Individual	1	398.79	398.79
C. Servicios				
Internet	Mes	6	1,995.18	11,971.08
Luz	Hora	524	1.39	728.36
D. Servicios en AWS				
EC2 large	Hora	30	1.61	48.30
ESB SSD (gp2)	Mes	6	3.48	20.88
VPN	Hora	120	0.077	9.24
RDS large	Hora	2160	0.13	280.80
RDS x GB	GB	60	0.89	53.40
Route53	Mes	6	8.51	51.06
S3 Standard	GB	120	0.18	21.60
TOTAL Q				89,520.15

Nota. Detalle del presupuesto para la realización del proyecto de software. Elaboración propia, realizado con Excel.

2.5. Beneficios del proyecto

- Plataforma que integra el manejo de sesiones e información de cuentas de los servicios informáticos en la institución. El proyecto estará montado en contenedores Docker para evitar problemas con dependencias.
- Manuales de usuario y técnico que detallan el funcionamiento de la plataforma tanto del lado del cliente como del lado de los servidores.

- Repositorio con el código utilizado para el desarrollo del proyecto, junto con un manual MD con su explicación general.
- Vídeo explicativo que contendrá las funcionalidades de la plataforma.

3. FASE ENSEÑANZA APRENDIZAJE

3.1. Capacitación propuesta

La metodología utilizada para realizar la capacitación son las pruebas en tiempo real de la plataforma por usuarios finales. Los encargados del proyecto utilizarán la plataforma que estará montada en un servidor de pruebas en un servicio en la nube para que puedan utilizar todas las funciones que esta provee.

En las primeras pruebas se encontrará presente el desarrollador del proyecto para ir indicando como se maneja la plataforma y cómo se puede acceder a todas las funciones. Todo lo relacionado al manejo interno de cada servidor no será tomado en cuenta en estas pruebas, ya que están fuera de los alcances del proyecto.

Las pruebas consistirán en seguir un flujo completo que comprende desde la creación de una nueva cuenta hasta la activación de los servicios utilizando roles como docente, administrativo o estudiante. Para el rol de administrador se probarán todas sus funciones desde el dashboard de administración, como la modificación de estados de usuario, visualización de gráficos o verificación de logs de actividades dentro del servidor o de las views de la plataforma.

3.2. Material elaborado

Los distintos materiales físicos y virtuales contruidos, y entregados a la institución como parte de la solución final. Entre ellos se detallan los manuales, el vídeo explicativo y la documentación del proyecto.

3.2.1. Manual de usuario

Manual que explica, con ejemplos gráficos, todos los pasos que debe seguir un usuario, dependiendo del rol, para poder acceder al sistema y poder navegar entre las diversas opciones que este contendrá.

Se detallan las situaciones con las que un usuario se puede encontrar en la plataforma, tales como: creación de una nueva cuenta, activación de la cuenta, generación de nuevos códigos de verificación, inicio de sesión, activación de servicios, ingreso a los servicios, visualización de información de usuarios, revisión de logs, cambio de estado de cuentas de usuarios, revisión de gráficos, entre otros.

3.2.2. Manual técnico

Contiene información sobre todos los elementos referentes al código y las tecnologías de desarrollo utilizadas en el proyecto. Incluye la explicación detallada de variables, métodos, funciones, estructuras, constantes, entre otros. El manual esta desarrollado con la tecnología de Jsdoc y puede ser visualizado desde un navegador ya que se encuentra en formato html.

3.2.3. Vídeo explicativo

Son fragmentos de video en los cuales se muestra en tiempo real las principales características de la plataforma, de manera similar a como se explica en el manual de usuario, pero en este caso en formato audiovisual. Es una opción extra para poder verificar si las acciones realizadas por el usuario son correctas y llevan al resultado esperado.

3.2.4. Repositorio del código y manual md

Repositorio de Github que contiene toda la estructura del proyecto y todo el código utilizado en el desarrollo de este, junto con el manual técnico agregado en formato Markdown. El repositorio tiene todo el código utilizado para desarrollar el proyecto a excepción de datos sensibles como variables de entorno u otros datos de configuración de entornos de desarrollo.

El manual md especifica aspectos técnicos del desarrollo del proyecto, tales como las tecnologías utilizadas, las variables necesarias para correr el proyecto, las instrucciones a seguir para ejecutar el proyecto, entre otros.

CONCLUSIONES

1. La falta de una planificación previa hace que consolidar un sistema de inicio de sesión para varios proyectos que fueron desarrollados de manera aislada sea un desafío. Sin embargo, reutilizando los servicios de creación de sesión existentes, es posible construir los elementos necesarios para configurar sesiones en cada servicio y mantenerlas activas dentro de un único dominio.
2. Aunque existen 4 roles que se mantienen constantes en todos los servicios, muchos otros no lo hacen y son únicos en cada servicio, lo que dificulta la gestión de acceso y permisos en la plataforma consolidada. La solución más efectiva para este problema es utilizar los roles más abstractos de cada servicio, de esa manera el manejo de los roles específicos queda a juicio de cada servicio y se manejan de manera local, sin la necesidad de integrarlos a la plataforma consolidada.
3. El diseño de un único módulo de registro para varios servicios con características dispares es un reto. La manera más eficaz de resolver el problema es implementar un registro global que capture los factores comunes de todos los servicios y complementar el registro dentro de la plataforma con los datos que cada servicio necesita al momento de su activación. Esto evita solicitar información innecesaria en el registro global y permite crear formularios más simples y menos tediosos.

RECOMENDACIONES

1. Seguir el manual técnico para la integración de nuevos servicios a la plataforma, ya que los procesos están bien definidos y la adición de elementos adicionales ya está contemplada. No es necesaria la creación de elementos nuevos y únicos, solo se deben replicar y adaptar los existentes.
2. Evitar la creación de nuevos permisos o roles en la plataforma; en su lugar utilizar los globales existentes. Si es necesaria la creación de nuevos roles, que se utilicen de manera local en cada servicio, ya que estos no pueden ser escalados y se volvería innecesariamente complejo el manejo de todos los roles y permisos de manera individual.
3. Implementar un servicio de colas para el procesamiento de secuencias de alto volumen y manejo de eventos, evitando así caídas de rendimiento, pérdida de datos y peticiones erróneas.

REFERENCIAS

- Arias, D. (25 de Febrero de 2021). *Hashing in Action: Understanding bcrypt* [Hashing en acción: Comprendiendo bcrypt]. Auth0. <https://auth0.com/blog/ hashing-in-action-understanding-bcrypt/>
- Cea, O. (10 de Octubre de 2021). *Arquitectura CLEAN para el frontend*. daslaf.dev. <https://daslaf.dev/posts/arquitectura-clean-react>
- Daya, S., Duy, N., Eati, K., Ferreira, C., Glozic, D., Gucer, V., Gupta, M., Joshi, S., Lampkin, V., Martins, M., Narain, S., & Vennam, R. (2015). *Microservices from Theory to Practice. Creating Applications in IBM Bluemix Using the Microservices Approach* [Microservicios desde la teoría a la práctica. Creación de aplicaciones en IBM Bluemix con enfoque en microservicios]. IBM Redbooks. <https://www.redbooks.ibm.com/redbooks/pdfs/sg248275.pdf>
- Dziuba, A. (15 de Febrero de 2023). *8 Advantages of Node.js Development for Startups in 2023* [8 ventajas del desarrollo en Node.js para startups en 2023]. Relevant.software. <https://relevant.software/blog/7-benefits-of-node-js-for-startups/#Why Nodejs and How it Works>
- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M., & Newling, T. (2004). *Patterns: Service-Oriented Architecture and Web Services* [Patrones: Arquitectura orientada a servicios y servicios web. IBM Redbooks]. IBM Redbooks. <https://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>

Fernández, J. (28 de Julio de 2016). *¿Por qué utilizo Clean Architecture?*
Xurxodev. <https://xurxodev.com/por-que-utilizo-clean-architecture-en-mis-proyectos/>

Grigutyte, M. (16 de Junio de 2023). *What is bcrypt and how does it work?*
[¿Qué es bcrypt y cómo funciona?]. Nordvpn.
<https://nordvpn.com/blog/what-is-bcrypt/>

Huaca, E. (2011). *Sistema de gestion de historias clinicas para el departamento de bienestar universitario de la UTN*. [Tesis de pregrado, Universidad Técnica del Norte de Ecuador]. Archivo digital UTN.
<http://repositorio.utn.edu.ec/handle/123456789/587>

JWT.IO. (s. f.). *JSON Web Tokens introduction* [Introducción a JSON Web Tokens]. Jwt.io. <https://jwt.io/introduction>

Krug, S. (2014). *Don't Make Me Think, Revisited* [No me hagas pensar, revisitado]. New Riders.

Maboud, I. (9 de Octubre de 2020). *What is the MVC, Creating a [Node.js-Express] MVC Application* [¿Qué es MVC? Creación de una aplicación MVC (Node.js-Express)]. Medium.
<https://medium.com/@ipenywis/what-is-the-mvc-creating-a-node-js-express-mvc-application-da10625a4eda>

Ortiz, B., Barillas, M., Moreira, M., De Leon, S., Patzán, C., Palacios, L., & López, M.. (9 de abril de 2007). *División de desarrollo organizacional USAC*. Manual De organización de la facultad de ciencias jurídicas y sociales. <https://ddo.usac.edu.gt/wp-content/uploads/2024/03/M-O-FACULTAD-DE-CIENCIAS-JURIDICAS-Y-SOCIALES-2006.pdf>.

Peyrott, S. (2016). *JWT Handbook* [Manual de JWT]. Auth0, Inc. <https://auth0.com/resources/ebooks/jwt-handbook>

Facultad de Ciencias Jurídicas y Sociales. (2021). *Sobre la facultad*. Facultad de Ciencias Jurídicas y Sociales. <https://derecho.cloud/sobre-la-facultad-de-ciencias-juridicas-y-sociales-usac/>

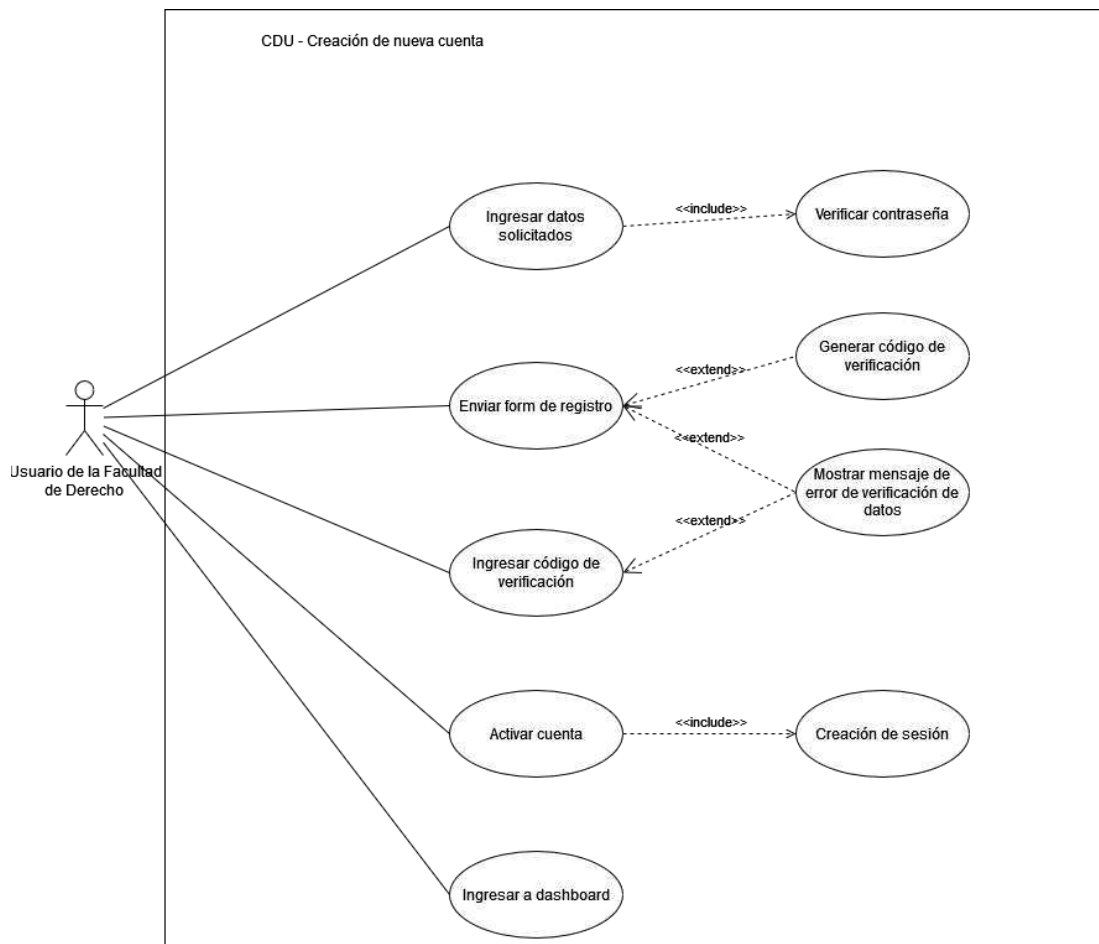
Taha, A. (10 de Junio de 2023). *Express.js: The Good, the Bad, and the Ugly* [Express.js: Lo bueno, lo malo y lo feo]. linkedin. <https://www.linkedin.com/pulse/expressjs-good-bad-ugly-aziz-taha>

Yüksel, O. (2 de Abril de 2023). *Implementing a Chain of Responsibility Design Pattern in Middleware* [Implementación de un patrón de diseño de cadena de responsabilidad en middleware]. tech.jotform. <https://tech.jotform.com/chain-of-responsibility-design-pattern-and-its-real-life-implementation-as-middleware-2c8adbad0076>

APÉNDICES

Apéndice 1.

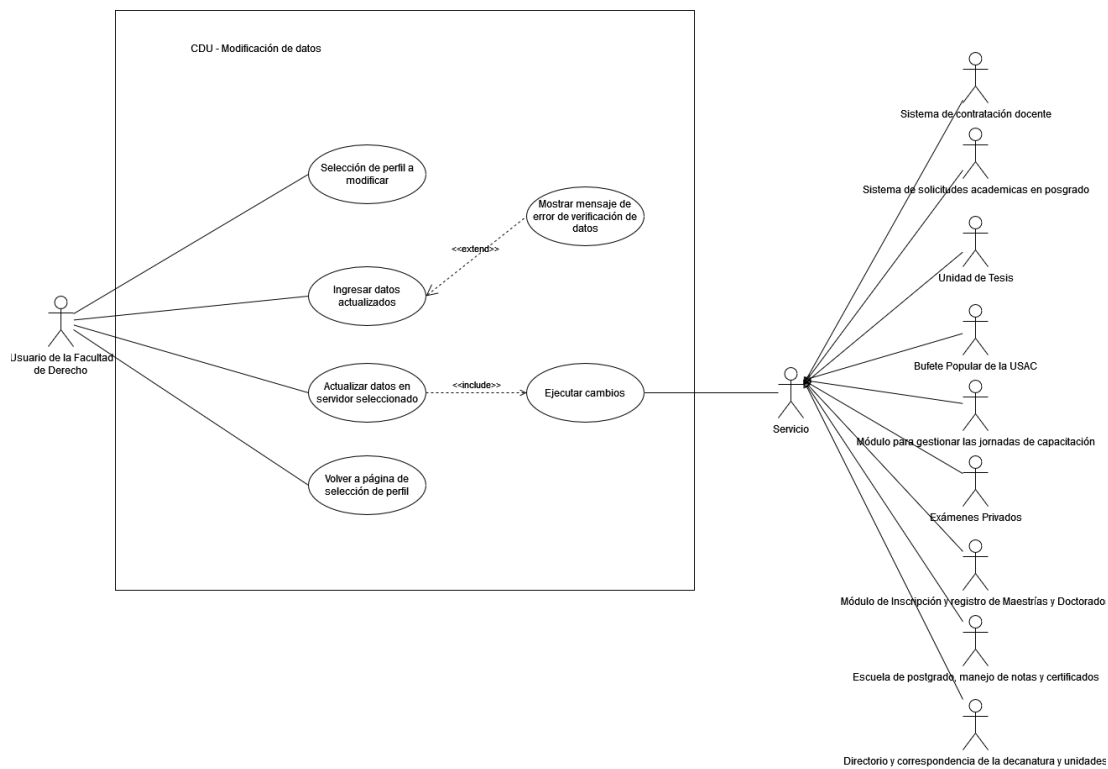
CDU001 – Creación de nueva cuenta



Nota. Detalle gráfico del CDU – Creación de nueva cuenta. Elaboración propia, realizado con draw.io.

Apéndice 2.

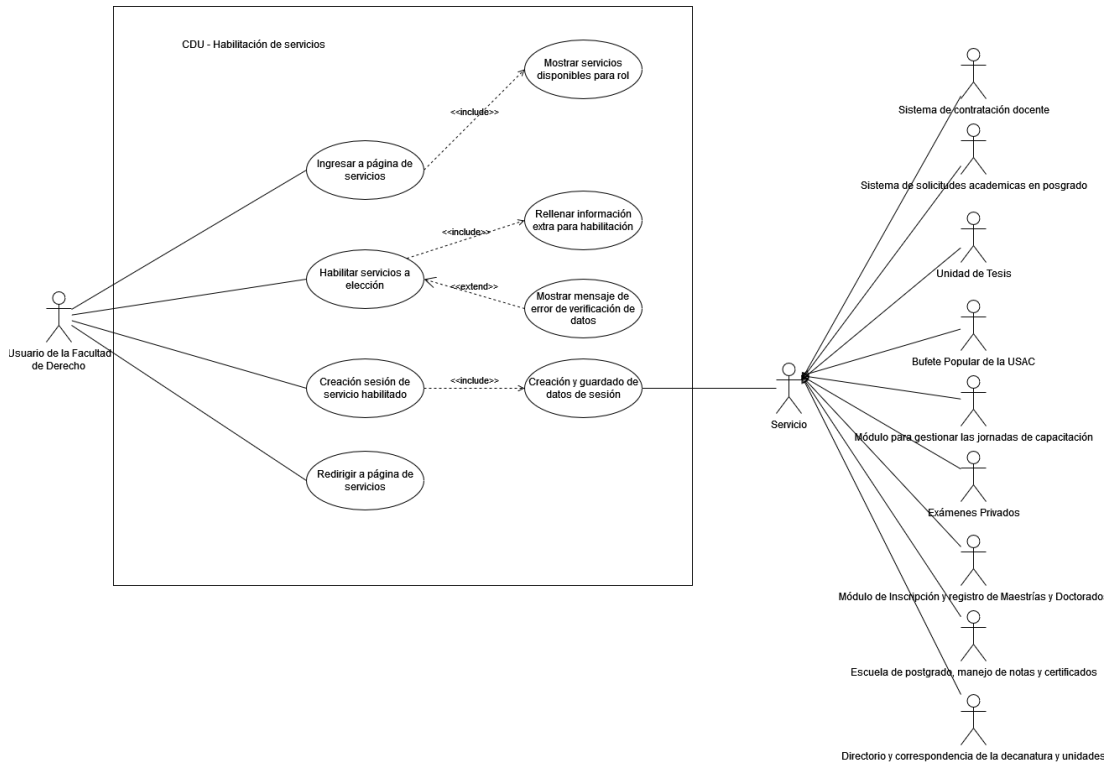
CDU002 – Modificación de datos



Nota. Detalle gráfico del CDU – modificación de datos. Elaboración propia, realizado con draw.io.

Apéndice 3.

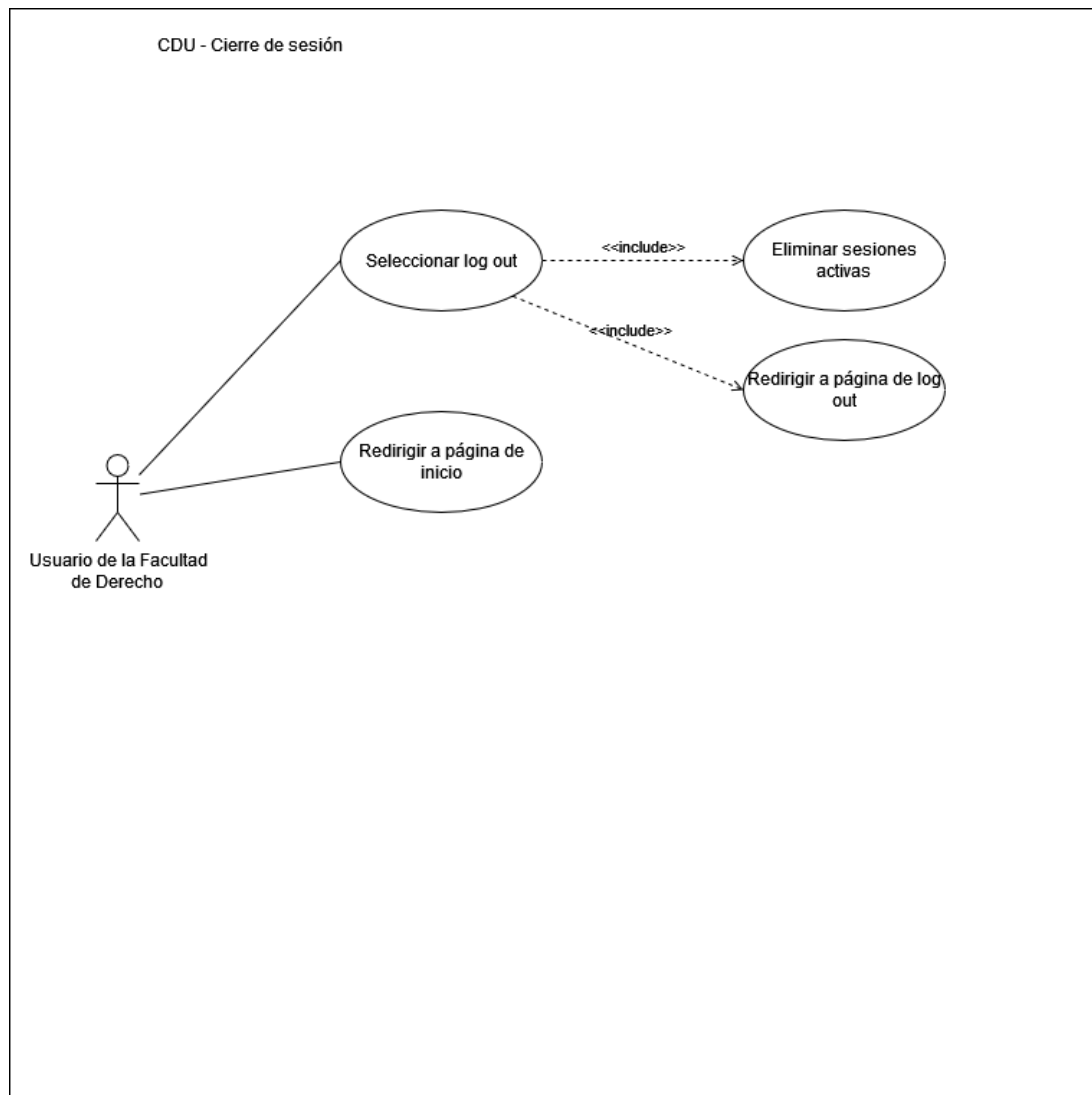
CDU003 – Habilitación de servicios



Nota. Detalle gráfico del CDU – habilitación de servicios. Elaboración propia, realizado con draw.io.

Apéndice 4.

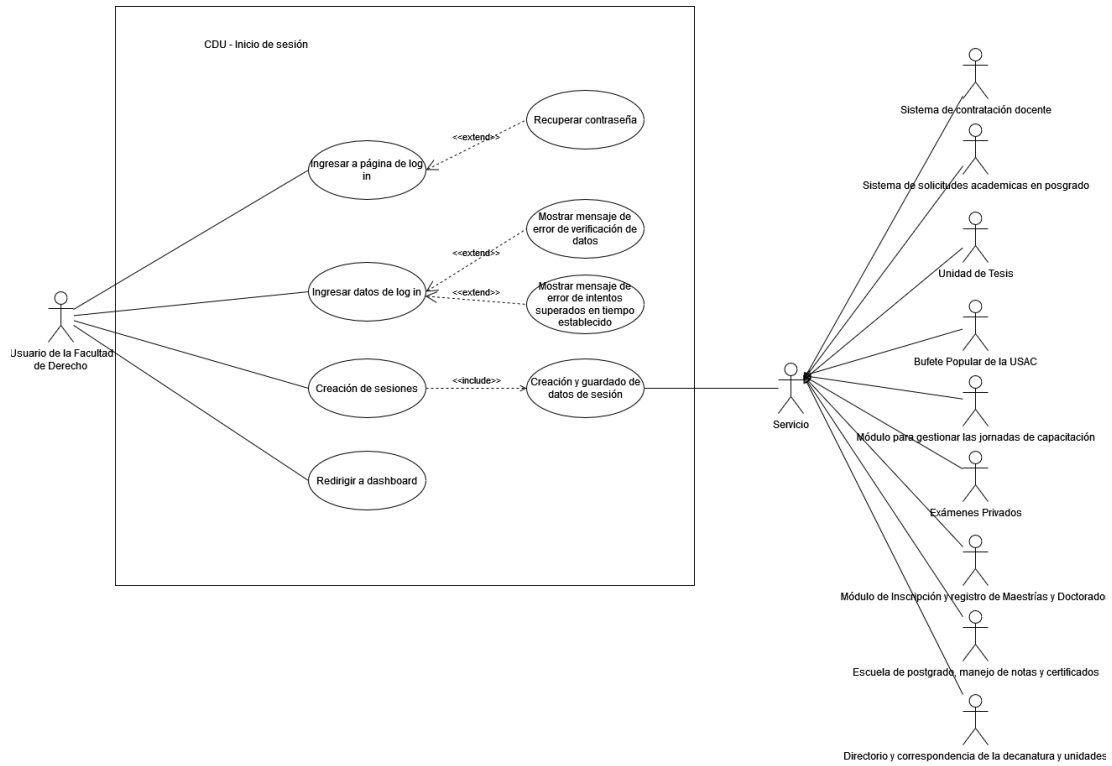
CDU004 – Cierre de sesión



Nota. Detalle gráfico del CDU – cierre de sesión. Elaboración propia, realizado con draw.io.

Apéndice 5.

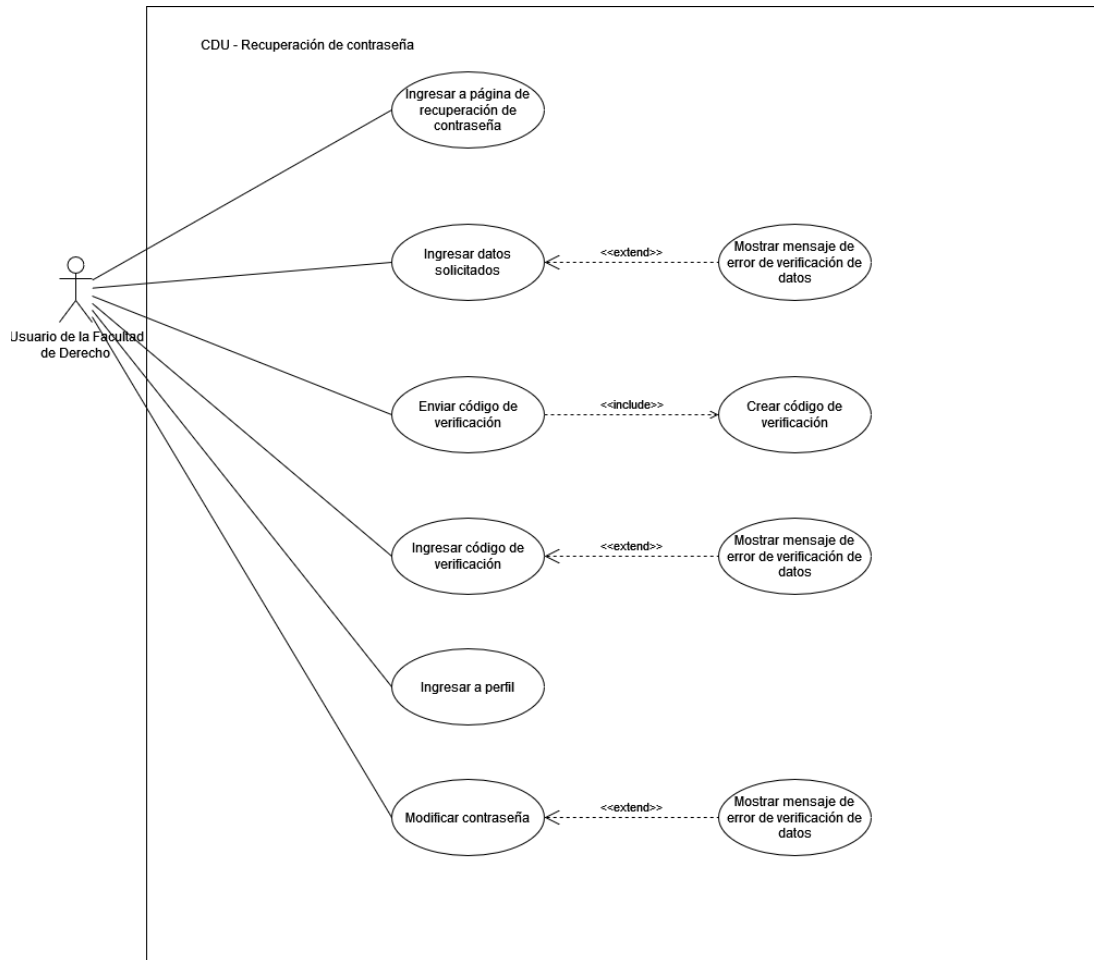
CDU005 – Inicio de sesión



Nota. Detalle gráfico del CDU – inicio de sesión. Elaboración propia, realizado con draw.io.

Apéndice 6.

CDU006 – Recuperación de contraseña



Nota. Detalle gráfico del CDU – recuperación de contraseña. Elaboración propia, realizado con draw.io.

Apéndice 7.

CDUE001 – Creación de cuenta

Nombre CDU	CDUE001_Creacion_Cuenta
Descripción	Los usuarios crear una nueva cuenta en el sistema consolidado.
Actores	Usuario de la Facultad de Derecho.
Secuencia Normal	El usuario ingresa a la página de inicio del sistema y selecciona la opción de crear una nueva cuenta. Más adelante ingresa los datos solicitados en el form de registro y confirma la creación de la cuenta.
Pasos	<ul style="list-style-type: none">• El usuario ingresa a la página de inicio y da clic sobre la opción de crear nueva cuenta.• El usuario ingresa los datos solicitados en el form de registro.• El usuario confirma la creación de la nueva cuenta• Se ejecuta el CDUE002_Confirmacion_Nueva_Cuenta
Excepciones	E1: El usuario ingresa datos erróneos en el form de registro. Solo puede avanzar hasta que ingrese los datos correctos.

Nota. Detalle técnico del CDUE – Creación de cuenta. Elaboración propia, realizado con Word.

Apéndice 8.

CDUE002 – Confirmación de nueva cuenta

Nombre CDU	CDUE002_Confirmacion_Nueva_Cuenta
Descripción	Los usuarios confirman la creación de una nueva cuenta en el sistema
Actores	Usuario de la Facultad de Derecho.
Secuencia Normal	El usuario recibe un correo con un código de verificación que deberán de ingresar en la página de confirmación que les aparecerá luego de confirmar la creación de una nueva cuenta.
Pasos	<ul style="list-style-type: none">• El usuario recibe un correo con el código de confirmación.• El usuario ingresa el código de confirmación en la página de confirmación de nuevo registro.• Se ejecuta el CDUE003_Creacion_Sesion• El usuario ingresa al sistema y visualiza el dashboard.
Excepciones	E1: El usuario ingresa un código equivocado. El contador de intentos aumenta y se bloqueará al llegar al máximo número de intentos.

Nota. Detalle técnico del CDUE – Confirmación de nueva cuenta. Elaboración propia, realizado con Word.

Apéndice 9.

CDUE003 – Creación de sesión

Nombre CDU	CDUE003_Creacion_Sesion
Descripción	Los servicios reciben una solicitud de creación de sesión de un usuario desde el sistema consolidado.
Actores	Servicios de la Facultad de Derecho
Secuencia Normal	El sistema seleccionado recibe la solicitud de inicio de sesión. Comprueba los datos y crea los datos de sesión necesarios. Retorna los datos hacia el sistema consolidado para crear la sesión consolidada.
Pasos	<ul style="list-style-type: none">• El servicio seleccionado recibe la solicitud de creación de sesión.• El servicio comprueba las credenciales.• El servicio crea los datos de sesión necesarios.• El servicio retorna los datos de sesión hacia el sistema consolidado.• El sistema consolidado crea la sesión para el servicio seleccionado.• El sistema actualiza el estado de la aplicación y agrega la nueva sesión.
Excepciones	E1: El sistema no retorna datos de sesión por algún error interno. Se reporta a los logs del sistema.

Nota. Detalle técnico del CDUE – Creación de sesión. Elaboración propia, realizado con Word.

Apéndice 10.

CDUE004 – Actualizar datos

Nombre CDU	CDUE004_Actualizar_Datos
Descripción	El usuario solicita la actualización de datos en alguno de los servicios del sistema consolidado.
Actores	Usuarios registrados de la Facultad de Derecho
Secuencia Normal	El usuario selecciona un perfil para actualizar los datos de este, modifica los datos y confirma la actualización.
Pasos	<ul style="list-style-type: none">• El usuario ingresa a su perfil.• El usuario selecciona un perfil para modificar los datos.• El usuario ingresa los nuevos datos.• El usuario confirma la actualización de los datos.• Se ejecuta el CDUE005_Ejecutar_Cambios• El usuario retorna a la página de perfil.
Excepciones	E1: El sistema no retorna un resultado satisfactorio desde algún servicio. Se reporta a los logs. E2: El usuario ingresa datos incorrectos en los forms para la actualización de datos. Se le solicita que haga una revisión de sus datos ingresados.

Nota. Detalle técnico del CDUE – Actualización de datos de usuario. Elaboración propia, realizado con Word.

Apéndice 11.

CDUE005 – Ejecutar cambios

Nombre CDU	CDUE005_Ejecutar_Cambios
Descripción	Se envían solicitudes a los servicios informáticos de la Facultad de Derecho para realizar cambios en los datos de las cuentas de los usuarios registrados que solicitan los cambios.
Actores	Servicios de la Facultad de Derecho
Secuencia Normal	El servicio recibe una solicitud de modificación de datos. Actualiza los datos y retorna un mensaje con el resultado de la operación.
Pasos	<ul style="list-style-type: none">• El servicio recibe la solicitud de actualización de datos.• El servicio realiza la actualización de datos.• El servicio retorna el resultado de la operación.
Excepciones	E1: El sistema no retorna un resultado satisfactorio desde algún servicio. Se reporta a los logs.

Nota. Detalle técnico del CDUE – Ejecutar cambios. Se refiere principalmente a cambios como la creación de solicitudes de creación de cuentas que no se crean directamente. Elaboración propia, realizado con Word.

Apéndice 12.

CDUE006 – Cerrar sesión

Nombre CDU	CDUE006_Cerrar_Sesión
Descripción	El usuario solicita la eliminación de su sesión activa en el navegador actual.
Actores	Usuarios registrados de la Facultad de Derecho
Secuencia Normal	El usuario ingresa a la opción de “log out” en cualquiera de los servicios de la Facultad de Derecho. Estos envían una solicitud al sistema consolidado para la eliminación de los datos de sesión. El sistema consolidado termina eliminando los datos y redirigiendo al usuario a la página de inicio.
Pasos	<ul style="list-style-type: none">• El servicio recibe la solicitud para la eliminación de la sesión actual.• El servicio redirige al usuario a la página de “log out” del sistema consolidado.• El sistema consolidado elimina los datos de sesión del usuario del navegador utilizado.• El sistema consolidado actualiza el estado de todas las pestañas o componentes en uso para que sean notificados de que la sesión ha sido eliminada.• El usuario es redirigido a la página de inicio del sistema consolidado.
Excepciones	E1: El sistema no redirecciona al usuario a la página de “log out” del sistema consolidado. Se reporta a los logs.

Nota. Detalle técnico del CDUE – Cierre de sesión de usuario. Elaboración propia, realizado con Word.

Apéndice 13.

CDUE007 – Iniciar sesión

Nombre CDU	CDUE007_Iniciar_Sesión
Descripción	El usuario solicita la eliminación de su sesión activa en el navegador actual.
Actores	Usuarios registrados de la Facultad de Derecho
Secuencia Normal	El usuario solicita la creación de una sesión en el sistema consolidado mediante la opción de “log in”. El sistema consolidado notifica a los servicios disponibles para que estos creen y activen sesiones para el usuario. El sistema finalmente guarda los datos de sesión para que el usuario pueda ingresar.
Pasos	<ul style="list-style-type: none">• El usuario ingresa a la página de “log in” del sistema consolidado, ingresa sus datos e inicia sesión.• El sistema comprueba los datos del usuario.• Se ejecuta el CDUE_008_Crear_Sesion para cada uno de los servicios disponibles para el usuario.• Se guardan los datos de sesión del usuario recibidos desde los servicios disponibles y se crea la sesión.• El usuario es redirigido al dashboard.
Excepciones	<p>E1: El usuario ingresa datos incorrectos para la creación de la sesión. Se le solicitan de nuevo y se aumenta el contador de intentos.</p> <p>E2: El usuario se queda sin intentos para el inicio de sesión con ese usuario. Se notifica al usuario que deberá esperar cierto tiempo para volver a intentar ingresar al sistema.</p>

Nota. Detalle técnico del CDUE – Inicio de sesión de usuario. En este caso se toma en cuenta la creación de sesiones en todos los servicios activos. Elaboración propia, realizado con Word.

Apéndice 14.

CDUE008 – Crear sesión

Nombre CDU	CDUE008_Crear_Sesion
Descripción	El sistema crear una sesión para el usuario solicitante.
Actores	Servicios de la Facultad de Derecho
Secuencia Normal	El servicio recibe una solicitud de creación de sesión desde el sistema consolidado. Comprueba los datos del usuario y crea los datos de sesión. Finalmente los retorna hacia el sistema consolidado.
Pasos	<ul style="list-style-type: none">• El servicio recibe la solicitud para la creación de una nueva sesión.• El servicio comprueba las credenciales del usuario.• El servicio crea los datos para la nueva sesión.• El servicio retorna los datos de sesión hacia el sistema consolidado.
Excepciones	E1: El servicio no logra comprobar de forma exitosa las credenciales del usuario. Se reporta a los logs.

Nota. Detalle técnico del CDUE – Creación de sesión. Se refiere principalmente a la creación de una sesión en un servicio específico. En conjunto con el CDUE07 se crea una sesión de usuario en integración. Elaboración propia, realizado con Word.

Apéndice 15.

CDUE009 – Solicitar cambio de contraseña

Nombre CDU	CDUE009_Solicitar_Cambio_Contraseña
Descripción	El usuario solicita la recuperación y cambio de contraseña.
Actores	Usuarios registrados de la Facultad de Derecho
Secuencia Normal	El usuario solicita el cambio de su contraseña desde el respectivo apartado en la página de inicio de sesión. El sistema crea y envía un código de confirmación para que el usuario pueda ingresar a su perfil y cambiar su contraseña.
Pasos	<ul style="list-style-type: none">• El usuario selecciona la opción de recuperar contraseña desde la página de inicio de sesión.• El usuario ingresa las credenciales solicitadas.• El sistema crea un código de verificación y se lo envía al usuario por correo.• El usuario ingresa el código de verificación.• El usuario es redirigido a la página de perfil para que pueda actualizar su contraseña.
Excepciones	E1: El usuario ingresa credenciales erróneas. Se notifica al usuario del error. E2: El usuario no recibe el correo con el código de verificación. Se reporta a los logs. E3: El código de verificación expira antes de que el usuario lo ingrese. Se le notifica al usuario y debe volver a intentarlo.

Nota. Detalle técnico del CDUE – Solicitud de cambio de contraseña. La contraseña se actualiza en la plataforma de integración y funciona para la creación de sesiones en todos los servicios. Elaboración propia, realizado con Word.

Apéndice 16.

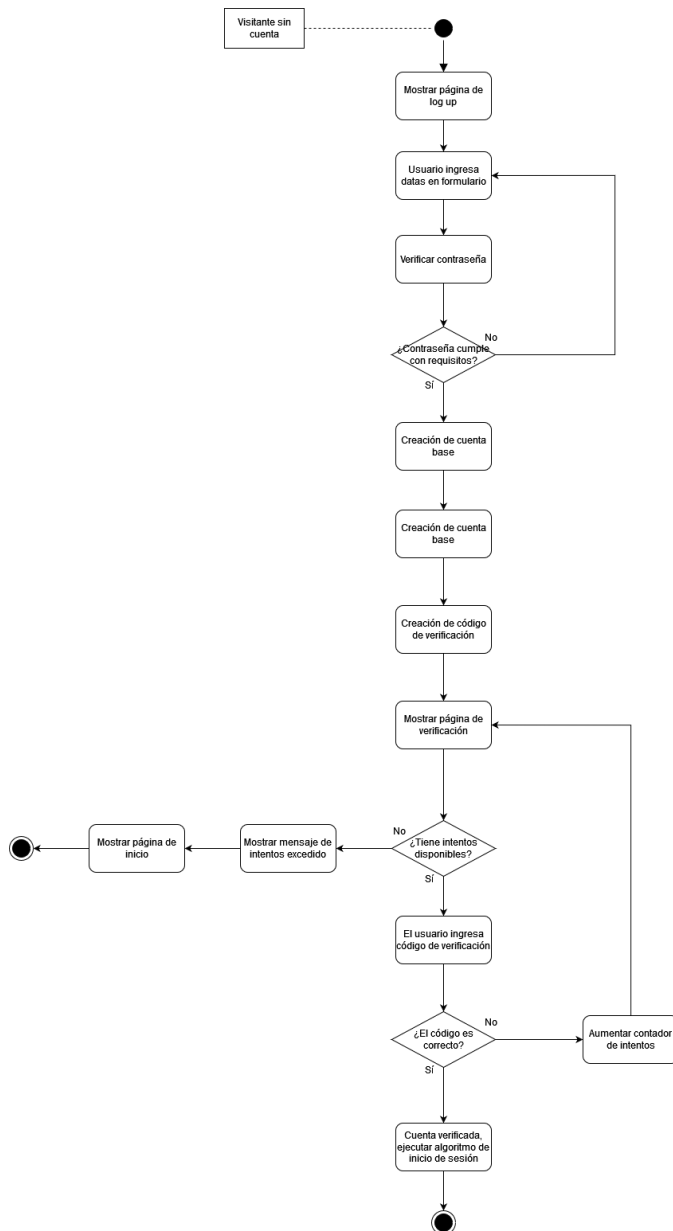
CDUE010 – Habilitar servicios

Nombre CDU	CDUE010_Habilitar_Servivios
Descripción	El usuario solicita la activación de un servicio disponible para su rol registrado.
Actores	Usuarios registrados de la Facultad de Derecho
Secuencia Normal	El usuario ingresa al apartado de servicios y solicita la activación de uno de estos. Rellena los datos necesarios para la activación y recibe la notificación de la creación de una nueva sesión.
Pasos	<ul style="list-style-type: none">• El usuario ingresa a la página de servicios.• El usuario selecciona un servicio para solicitar su habilitación.• El usuario ingresa los datos solicitados en el form emergente para la habilitación del servicio.• El usuario confirma la solicitud.• El sistema notifica al servicio selecciona para que registre al nuevo usuario.• El servicio retorna el resultado de la creación.• Se ejecuta el CDUE003_Creacion_Sesion• El usuario es redirigido a la página de servicios con el servicio habilitado y la sesión creada.
Excepciones	E1: El usuario ingresa datos erróneos. Se le notifica al usuario. E2: El servicio retorna un resultado insatisfactorio en el registro de la cuenta. Se reporta a los logs.

Nota. Detalle técnico del CDUE – Habilitación de servicios. Procedimiento que se sigue para la habilitación de algún servicio informático disponible en la plataforma. Elaboración propia, realizado con Word.

Apéndice 17.

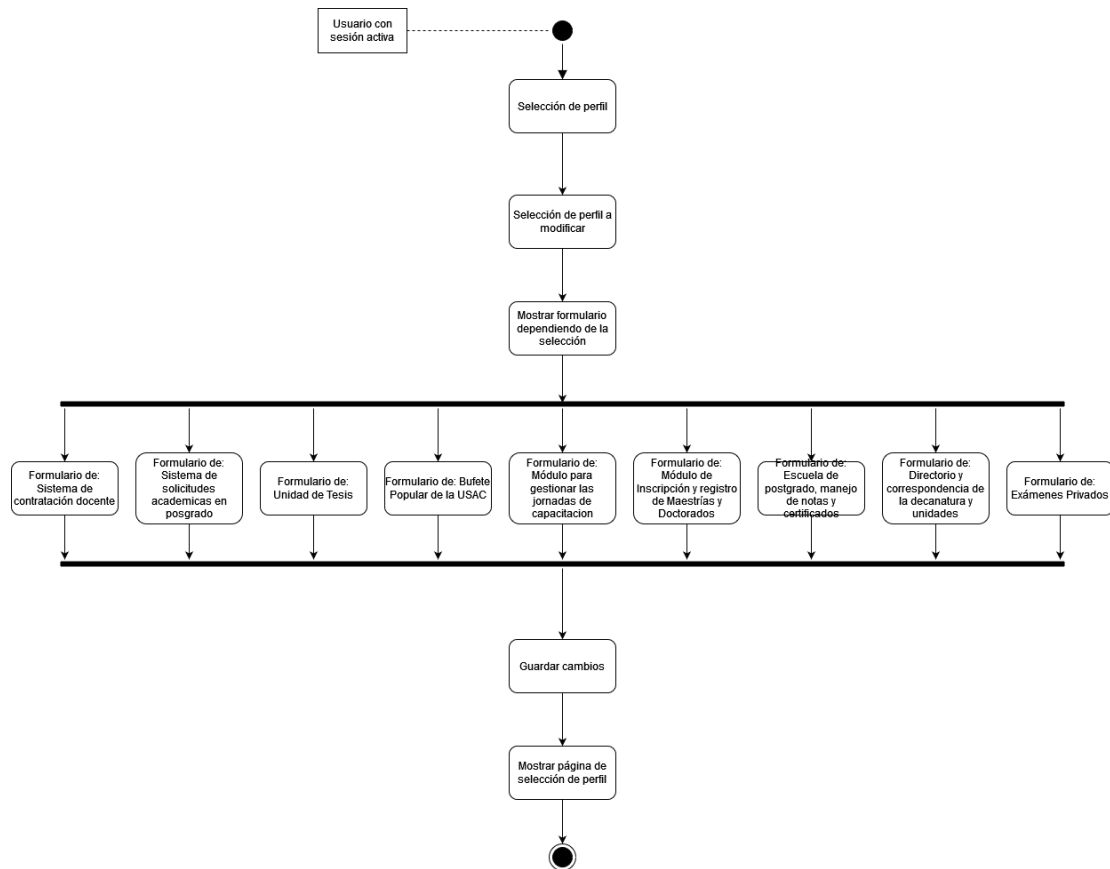
Diagramas de actividades de creación de cuenta



Nota. Diagrama de actividades que detalla la tarea de creación de una cuenta. Elaboración propia, realizado con draw.io.

Apéndice 18.

Diagramas de actividades de modificación de cuenta



Nota. Diagrama de actividades que detalla la tarea de modificación de cuentas. Elaboración propia, realizado con draw.io.

Apéndice 19.

Diagramas de actividades de inicio de sesión



Nota. Diagrama de actividades que detalla la tarea de inicio de sesión. Elaboración propia, realizado con draw.io.

Apéndice 20.

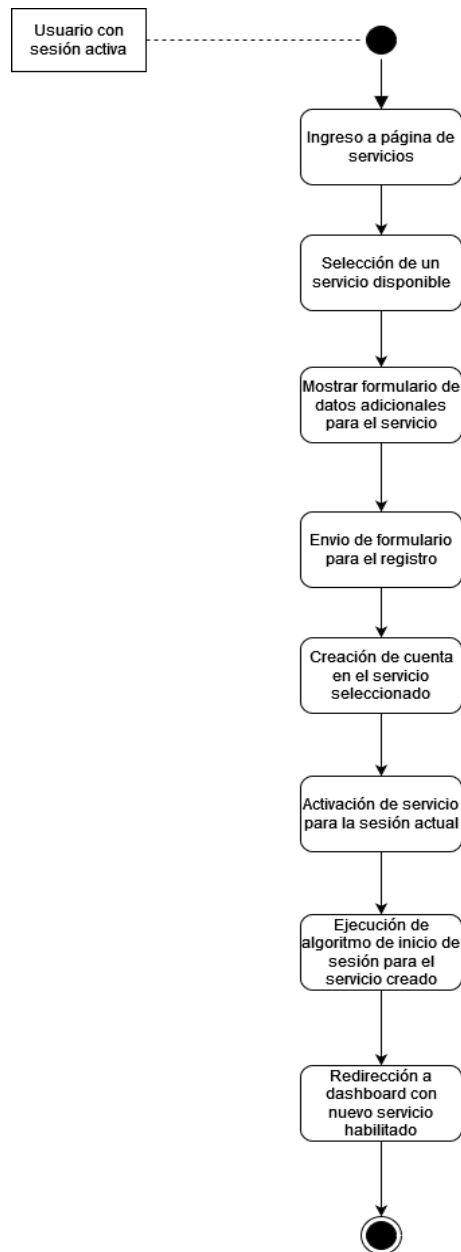
Diagramas de actividades de cierre de sesión



Nota. Diagrama de actividades que detalla la tarea de cierre de sesión. Elaboración propia, realizado con draw.io.

Apéndice 21.

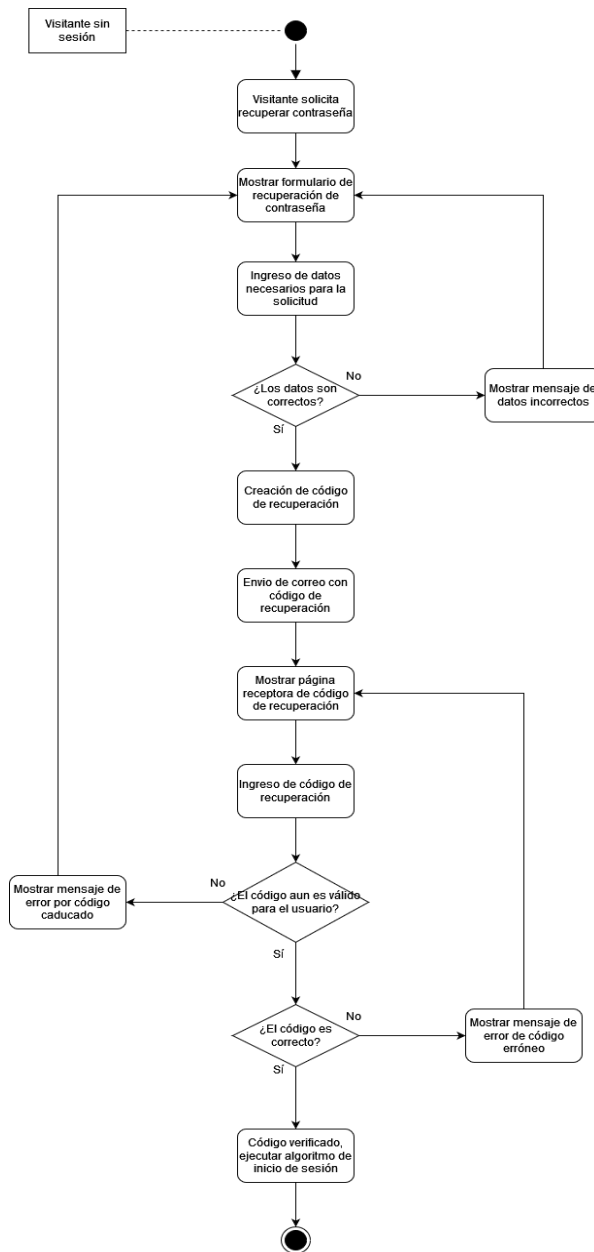
Diagramas de actividades de asignación de permisos



Nota. Diagrama de actividades que detalla la tarea de asignación de permisos. Elaboración propia, realizado con draw.io

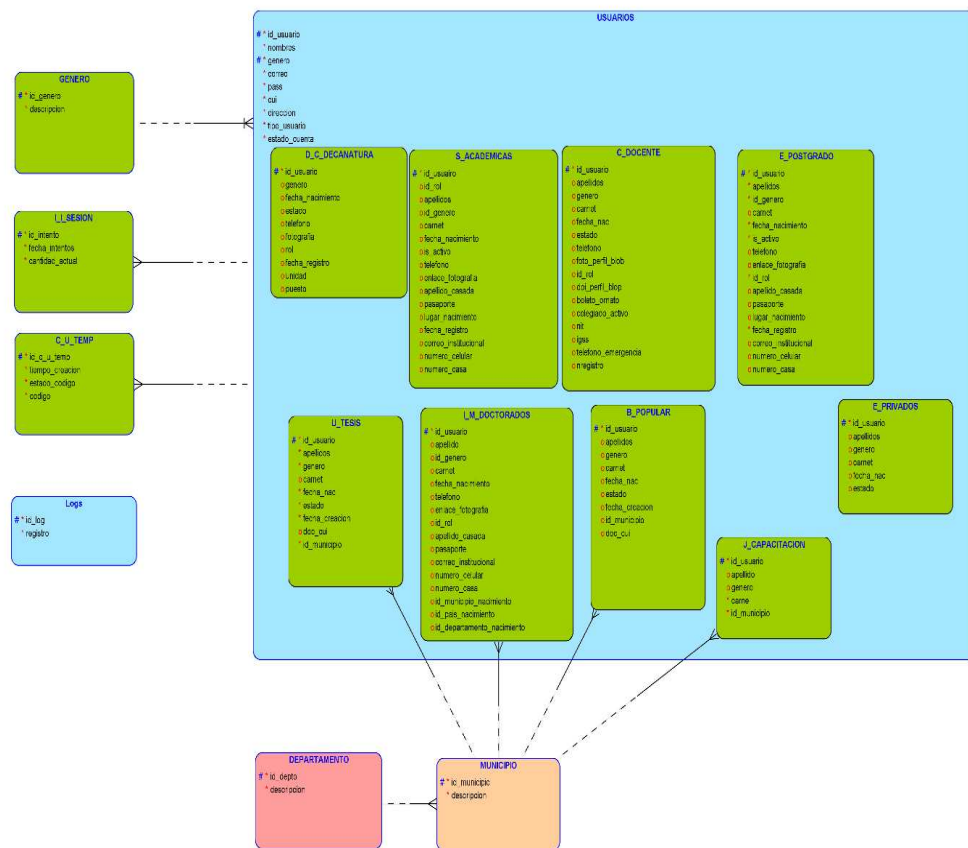
Apéndice 22.

Diagramas de actividades de recuperación de contraseña



Nota. Diagrama de actividades que detalla la tarea de recuperación de contraseñas.
Elaboración propia, realizado con draw.io.

Diagrama entidad relación



Nota. Diagrama de datos de las entidades y sus relaciones en el proyecto. Elaboración propia, realizado con el software datamodeler de Oracle.

Apéndice 24.

Historia de usuario 001

Historia de usuario	
Número: 1	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Creación de cuenta de usuario	
Prioridad en negocio: Alta	Riesgo en el desarrollo: Alta
Programador responsable: Joel Santos	
Descripción: El usuario de la facultad de derecho desea crear una cuenta para poder acceder a los servicios informáticos que proporciona la Facultad de Derecho.	
Observaciones: N/A	

Nota. Detalle de la historia de usuario de la creación de una cuenta desde cualquier rol de usuario. Elaboración propia, realizado con Word.

Apéndice 25.

Historia de usuario 002

Historia de usuario	
Número: 2	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Inicio de sesión consolidado	
Prioridad en negocio: Alta Riesgo en el desarrollo: Alta	
Programador responsable: Joel Santos	
Descripción: El usuario de la facultad de derecho desea que al ingresar al sistema todos los servicios a los cuales aplique su rol estén disponibles y con sesiones activas.	
Observaciones: Al ingresar el usuario se tiene que crear sesiones en todos los servicios en los cuales tenga una cuenta activa.	

Nota. Detalle de la historia de usuario del inicio de sesión desde el apartado consolidado en la plataforma de integración. Elaboración propia, realizado con Word.

Apéndice 26.

Historia de usuario 003

Historia de usuario	
Número: 3	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Modificación de datos de cuentas	
Prioridad en negocio: Alta Riesgo en el desarrollo: Alta	
Programador responsable: Joel Santos	
Descripción: El usuario de la facultad de derecho desea tener la posibilidad de modificar los datos registrados en su cuenta en cualquier momento y en cualquier servicio disponible para su rol.	
Observaciones: Todos los servicios tienen diferentes formas de modificar los datos de cuentas y diferentes datos a modificar.	

Nota. Detalle de la historia de usuario de la modificación de información en las cuentas de los usuarios. Elaboración propia, realizado con Word.

Apéndice 27.

Historia de usuario 004

Historia de usuario	
Número: 4	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Cierre de sesión consolidado	
Prioridad en negocio: Alta Riesgo en el desarrollo: Alta	
Programador responsable: Joel Santos	
Descripción: El usuario de la facultad de derecho quiere que, al momento de cerrar la sesión en cualquiera de los servicios en los que se encuentre navegando, se cierre sesión en todos los demás servicios en conjunto.	
Observaciones: Todos los servicios tienen diferentes formas de manejar y guardar los datos de sesiones.	

Nota. Detalle de la historia de usuario sobre el cierre de sesión consolidado, es decir, el cierre de sesión desde todos los servicios activados en una cuenta. Elaboración propia, realizado con Word.

Apéndice 28.

Historia de usuario 005

Historia de usuario	
Número: 5	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Habilitación de servicios	
Prioridad en negocio: Alta Riesgo en el desarrollo: Alta	
Programador responsable: Joel Santos	
Descripción: El usuario de la facultad de derecho quiere tener la posibilidad de habilitar otros servicios disponibles en la facultad con sus mismas credenciales, sin tener realizar de nuevo otro proceso de inscripción y utilizar diferentes credenciales en el mismo.	
Observaciones: Todos los servicios utilizan diferentes datos para los registros de nuevas cuentas.	

Nota. Detalle de la historia de usuario sobre la habilitación de servicios. El usuario tiene la posibilidad de habilitar los servicios presentados en la plataforma de manera individual. Elaboración propia, realizado con Word.

Apéndice 29.

Historia de usuario 006

Historia de usuario	
Número: 6	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Recuperación de contraseña	
Prioridad en negocio: Media	Riesgo en el desarrollo: Alta
Programador responsable: Joel Santos	
Descripción: Al existir varios servicios informáticos diferentes, con diferentes metodologías en el manejo de la recuperación de contraseñas, el usuario de la Facultad de Derecho desea tener la opción de recuperar su contraseña en un único lugar y que esta pueda ser utilizada para todos los servicios activos.	
Observaciones: El manejo de tokens, encriptación y forma de recuperación de contraseñas es diferente en cada servicio.	

Nota. Detalle de la historia de usuario sobre la recuperación de contraseña de un usuario.
Elaboración propia, realizado con Word.

Apéndice 30.

Historia de usuario 007

Historia de usuario	
Número: 7	Usuario: Usuario de la Facultad de Derecho
Nombre de historia: Visualización de reportes	
Prioridad en negocio: Media	Riesgo en el desarrollo: Alta
Programador responsable: Joel Santos	
Descripción: Los usuarios de la parte administrativa de la facultad de derecho desean tener la opción de poder visualizar estadísticas sobre el uso del sistema, el número de estudiantes inscritos en los módulos y con posibilidad de filtrarlos.	
Observaciones: Los filtros se deben de construir de tal manera que puedan ser tomados generalmente entre todos los módulos o individualmente.	

Nota. Detalle de la historia de usuario sobre la visualización de reportes desde un usuario administrador. Elaboración propia, realizado con Word.