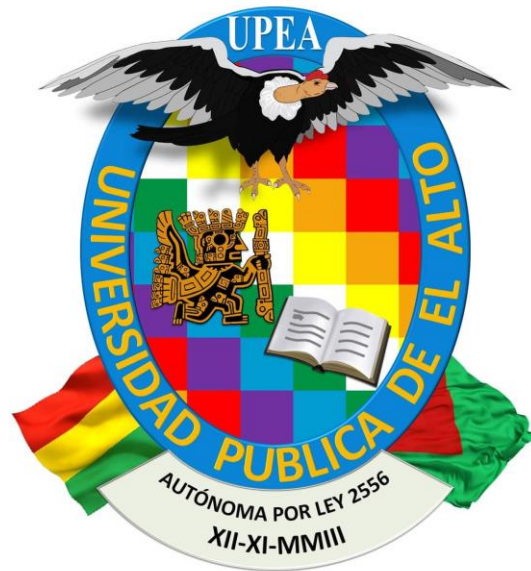


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES”

CASO: UNIVERSIDAD PÚBLICA DE EL ALTO

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
Mención: **INFORMÁTICA Y COMUNICACIONES**

Postulante: Jhonny Apaza Chambi

Tutor Metodológico: Lic. Maricel Yarari Mamani

Tutor Revisor: Ing. Sergio Ramiro Rojas Saire

Tutor Especialista: Ing. Francisco Luis Paez Rocha

EL ALTO – BOLIVIA

2021

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

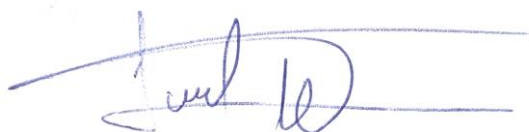
Yo, **Jhonny Apaza Chambi** Estudiante con C.I. 6878067 L.P. mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, Noviembre del 2021.



Jhonny Apaza Chambi
C.I. 6878067 L.P.
jhonnyapazach@gmail.com

DEDICATORIA

A Dios, por haberme dado la vida, salud e iluminación en todos los momentos más difíciles y quien guía siempre nuestra vida y a quien debo todo lo que tengo. A mis padres Victoriano y Margarita quienes me brindaron su amor, cariño y comprensión y quienes me inculcaron valores y principios, sus consejos y recomendaciones los tomo en cuenta siempre y las aplico en mi vida, además del apoyo constante que me brindan.

AGRADECIMIENTOS

Mi sincero agradecimiento a Dios por guiar mis pasos, por llevarme a alcanzar mis metas y objetivos, por todas las bendiciones brindadas, por haberme encaminado de tal manera que me permitió culminar este sueño.

A la Universidad Pública del Alto y la Carrera de Ingeniería de Sistemas en donde se me abrieron las puertas para poder alcanzar una meta más en mi camino.

Agradecer a mi tutor especialista Ing. Francisco Luis Paez Rocha por su apoyo, por brindarme su conocimiento que me ayudo a construir este proyecto.

Agradecer a mi tutor metodológico Ing. Maricel Yarari Mamani, por su tiempo comprensión, paciencia y orientación con sus conocimientos en la realización de este proyecto.

Agradecer a mi tutor revisor Ing. Sergio Ramiro Rojas Saire, por brindarme su tiempo, su conocimiento, experiencia y por su paciencia, que me ayudaron a culminar el presente proyecto

ÍNDICE GENERAL

CAPÍTULO I

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES DE LA INVESTIGACIÓN	1
1.2.1. Antecedentes institucionales	1
1.2.2. Antecedentes de trabajos afines	2
1.3. PLANTEAMIENTO DEL PROBLEMA	3
1.3.1. Problema general	4
1.3.2. Problemas específicos	4
1.4. OBJETIVOS	5
1.4.1. Objetivo general	5
1.4.2. Objetivos específicos	5
1.5. JUSTIFICACIÓN DEL TEMA.....	6
1.5.1. Justificación técnica	6
1.5.2. Justificación económica	6
1.5.3. Justificación social.....	7
1.6. METODOLOGÍA.....	7
1.6.1. UWE	7
1.6.2. Métricas de calidad	8
1.6.3. Métodos de recolección de datos.....	8
1.6.4. Estimación de costos	8
1.7. HERRAMIENTAS.....	9
1.7.1. Hypertext Pre-Processor “PHP”	9
1.7.2. Apache.....	9
1.7.3. Mysql.....	9
1.7.4. Laravel	9
1.7.5. Javascript	10
1.7.6. Jquery	10
1.7.7. CSS.....	10

1.7.8. Materialize	10
1.8. LÍMITES Y ALCANCES	10
1.8.1. Límites	10
1.8.2. Alcances.....	11
1.9. APORTES	11

CAPÍTULO II

2. MARCO TEORICO

2.1. CONCEPTOS BÁSICOS	12
2.1.1. SEGUIMIENTO	12
2.1.2. TRÁMITE	12
2.1.3. SISTEMA DE INFORMACIÓN	12
2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE	13
2.3. UWE	13
2.3.1. Modelos de UWE	14
2.4. HERRAMIENTAS	19
2.4.1. Laravel	19
2.4.2. JQuery.....	20
2.4.3. Materialize	20
2.5. ISO/IEC 9126	21
2.5.1. Funcionalidad.....	23
2.5.2. Confiabilidad	23
2.5.3. Usabilidad	24
2.5.4. Eficiencia.....	25
2.5.5. Capacidad de mantenimiento.....	26
2.5.6. Portabilidad	27
2.5.7. Calidad de uso	27
2.6. PRUEBAS DEL SOFTWARE	28
2.6.1. Método de la caja negra.....	29
2.6.2. Método de la caja blanca	30
2.7. COCOMO	31
2.7.1. COCOMO II.....	32

2.8. SEGURIDAD	33
2.8.1. Normas ISO/IEC 27000.....	33
2.8.2. Aplicación desde el usuario.....	33
2.8.3. Acceso al sistema	33
2.8.4. Acceso a la base de datos	34
2.8.5. Backups	34

CAPÍTULO III

3. MARCO APLICATIVO

3.1. INTRODUCCION.....	35
3.2. METODOLOGÍA UWE – CONSTRUCCIÓN E IMPLEMENTACIÓN.....	35
3.2.1. Modelo de requerimientos.....	36
3.2.1.1. Actores	36
3.2.1.2. Obtención de requisitos	37
3.2.1.3. Lista de requerimientos del sistema	38
3.2.1.3.1. Requerimientos funcionales	38
3.2.1.3.2. Requerimientos no funcionales	39
3.2.1.4. Diagrama de casos de uso general	39
3.2.1.5. Diagrama de caso de uso: Administración del sistema	40
3.2.1.6. Diagrama de Caso de Uso específicos: Administración de Unidades	41
3.2.1.7. Diagrama de Caso de Uso específicos: Administración de Usuarios	42
3.2.1.8. Diagrama de Caso de Uso: Modulo de Trámites	43
3.2.1.9. Diagrama de Caso de Uso: Modulo de Notificaciones.....	44
3.2.1.10. Diagrama de Caso de Uso: Cuenta de Usuario.....	45
3.2.2. Modelo de Contenido	46
3.2.3. Modelo de Navegación.....	47
3.2.3.1. Modelo de navegación: Inicio	47
3.2.3.2. Diagrama de navegación: Administración	48
3.2.3.3. Diagrama de navegación: Trámites	49

3.2.3.4.	Diagrama de navegación: Cuenta de usuario.....	50
3.2.4.	Modelo de Presentación.....	51
3.2.4.1.	Modelo de presentación: Inicio	51
3.2.4.2.	Modelo de presentación: Inicio de usuario	52
3.2.4.3.	Modelo de presentación: Trámites pendientes de recepción.	53
3.2.4.4.	Modelo de presentación: Trámites recibidos	54
3.2.4.5.	Modelo de presentación: Trámites despachados	55
3.2.4.6.	Modelo de presentación: Trámites finalizados.....	56
3.2.4.7.	Modelo de presentación: Cuenta de usuario	57
3.2.4.8.	Modelo de presentación: Seguimiento detallado de trámites	58
3.2.5.	Modelo de Procesos.....	59
3.2.5.1.	Diagrama de procesos.....	59
3.2.5.2.	Diagrama de actividades: Inicio de sesión.....	60
3.2.5.3.	Diagrama de actividades: Recuperación de cuenta.....	61
3.2.5.4.	Diagrama de actividades: Seguimiento de trámite.....	62
3.2.5.5.	Diagrama de actividades: Seguimiento básico de trámite	63
3.2.5.6.	Diagrama de actividades: Seguimiento detallado de trámite .	64
3.2.5.7.	Diagrama de actividades: Creación de trámites	65
3.2.5.8.	Diagrama de actividades: Recepción de trámites.....	66
3.2.5.9.	Diagrama de actividades: Despacho de trámites.....	67
3.2.5.10.	Diagrama de actividades: Finalización de trámites.....	68
3.3.	MÉTRICAS DE CALIDAD.....	69
3.3.1.	Estándar ISO/IEC 9126.....	70
3.3.1.1.	Funcionalidad	70
3.3.1.2.	Confiabilidad.....	74
3.3.1.3.	Usabilidad.....	75
3.3.1.4.	Mantenibilidad.....	77
3.3.1.5.	Portabilidad.....	77
3.3.2.	Seguridad.....	78
3.3.2.1.	Seguridad a nivel de Sistema Operativo.....	78
3.3.2.2.	Seguridad a nivel de Base de Datos.....	78

3.3.2.3. Seguridad a nivel de Software	79
3.3.3. Costos	79
3.3.3.1. Método de estimación COCOMO II	79
3.3.3.2. Costos de proyecto.....	83
3.3.3.2.1. Costo del personal	83
3.3.3.2.2. Costo de elaboración del proyecto.....	83
3.3.3.2.3. Costo de implementación del proyecto	84
3.3.3.2.4. Costo total del proyecto.....	84
3.3.4. Pruebas de Software.....	84
3.3.4.1. Pruebas de Caja Blanca	84
3.3.4.2. Pruebas de Caja Negra	87
3.4. IMPLEMENTACIÓN DEL SISTEMA.....	89
3.4.1. Página de inicio	89
3.4.2. Ingreso al sistema	90
3.4.3. Recuperación de cuenta	91
3.4.4. Página inicial al iniciar sesión.....	93
3.4.5. Página de creación de trámites	93
3.4.6. Administración de unidades	93
3.4.7. Administración de usuarios	94
CAPÍTULO IV	
4. CONCLUSIONES Y RECOMENDACIONES	
4.1. CONCLUSIONES.....	96
4.2. RECOMENDACIONES	96
BIBLIOGRAFÍA	97
ANEXOS	99

ÍNDICE DE FIGURAS

Figura 1: Diagrama de caso de uso	15
Figura 2: Diagrama de actividades.....	15
Figura 3: Diagrama de clases	16
Figura 4: Diagrama de navegación	16
Figura 5: Diagrama de presentación	17
Figura 6: Modelo de estructura de procesos que describe las relaciones entre las diferentes clases de procesos	18
Figura 7: Modelo de Flujo de Procesos que especifica las actividades relacionadas con cada proceso	19
Figura 8: Gráfico de Norma de Evaluación ISO/IEC 9126	22
Figura 9: Gráfico de Evaluación Interna, Externa y Calidad de Uso ISO/IEC 9126	22
Figura 10: Gráfico de Característica de funcionalidad.....	23
Figura 11: Gráfico Característica de Confiabilidad	24
Figura 12: Gráfico de Característica de Usabilidad	25
Figura 13: Característica de Eficiencia.....	25
Figura 14: Gráfico de Característica de Mantenimiento	26
Figura 15: Gráfico de Característica de Portabilidad.....	27
Figura 16: Gráfico de Característica Calidad de uso.....	28
Figura 17: Diagrama de casos de uso general.....	40
Figura 18: Caso de uso “Administración del Sistema“	41
Figura 19: Diagrama de Caso de Uso “Administración de unidades”	42
Figura 20: Diagrama de Caso de Uso “Administración de usuarios“	43
Figura 21: Diagrama de Caso de Uso “Módulo de trámites“	44
Figura 22: Diagrama de Caso de Uso “Módulo de notificación“	45
Figura 23: Diagrama de Caso de Uso “Cuenta de Usuario“	46
Figura 24: Base de datos	47
Figura 25: Diagrama de navegación “Inicio“	48
Figura 26: Diagrama de navegación “Administración“	49

Figura 27: Diagrama de navegación “Trámites“	50
Figura 28: Diagrama de navegación “Cuenta de usuario“	51
Figura 29: Modelo de presentación “Inicio“	52
Figura 30: Modelo de presentación “Inicio de usuario“	53
Figura 31: Modelo de presentación “Trámites pendientes de recepción“	54
Figura 32: Modelo de presentación “Trámites recibidos“	55
Figura 33: Modelo de presentación “Trámites despachados“	56
Figura 34: Modelo de presentación “Trámites finalizados“	57
Figura 35: Modelo de presentación “Cuenta de usuario“	58
Figura 36: Modelo de presentación “Seguimiento detallado de trámites“	59
Figura 37: Diagrama de procesos	60
Figura 38: Diagrama de actividades “inicio de sesión“	61
Figura 39: Diagrama de actividades “recuperación de cuenta“	62
Figura 40: Diagrama de actividades “seguimiento de trámite“	63
Figura 41: Diagrama de actividades “seguimiento básico de trámite“	64
Figura 42: Diagrama de actividades “seguimiento detallado de trámite“	65
Figura 43: Diagrama de actividades “Creación de trámites“	66
Figura 44: Diagrama de actividades “Recepción de trámites“	67
Figura 45: Diagrama de actividades “Despacho de trámites“	68
Figura 46: Diagrama de actividades “Finalización de trámites“	69
Figura 47: Caja blanca	85
Figura 48: Página inicial del sistema	90
Figura 49: Ingreso al sistema	91
Figura 50: Recuperación de cuenta	91
Figura 51: Página inicial después de iniciar sesión	92
Figura 52: Página de creación de trámites	93
Figura 53: Página de administración de unidades.....	94
Figura 54: Página de administración de usuarios.....	95

ÍNDICE DE TABLAS

Tabla 1: Lista de actores	37
Tabla 2: Obtención de requisitos.....	37
Tabla 3: Requerimientos funcionales del sistema	39
Tabla 4: Requerimientos no funcionales del sistema	39
Tabla 5: Descripción de caso de uso “Administración del Sistema”	41
Tabla 6: Descripción de caso de uso “Administración de Unidades”	42
Tabla 7: Descripción de caso de uso “Administración de Usuarios”.....	43
Tabla 8: Descripción de caso de uso “Módulo de trámites”	44
Tabla 9: Descripción de caso de uso “Módulo de notificación”.....	45
Tabla 10: Descripción de caso de uso “Cuenta de Usuario”	46
Tabla 11: Características de la Funcionalidad.....	71
Tabla 12: Calculo del punto de función (Factores de Ponderación)	71
Tabla 13: Parámetros de medición.....	73
Tabla 14: Parámetros de medición.....	76
Tabla 15: Medidas de Seguridad.....	76
Tabla 16: Conversión de puntos de función	80
Tabla 17: Modelo COCOMO	81
Tabla 18: Conductores de costo	81
Tabla 19: Costo de elaboración del proyecto	84
Tabla 20: Costo total del proyecto.....	84
Tabla 21: Caso de prueba “Ingreso al sistema”	87
Tabla 22: Caso de prueba “Registro de trámites”.....	88
Tabla 23: Caso de prueba “Seguimiento de trámites”	89

CAPÍTULO I

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Actualmente los sistemas de fácil manejo se han convertido en una necesidad para las empresas e instituciones, esto se da por la cantidad de información que se maneja, un sistema web que maneje esta información de forma rápida y automática, y además de que los usuarios que lo utilicen necesiten un navegador y conexión a internet para su manejo es una solución ideal que permitirá cubrir esta necesidad.

El presente proyecto hace uso del flujo que sigue un trámite desde su inicio hasta su finalización, permitiendo registrar y hacer un seguimiento de hoja de ruta, mejorando la gestión de los trámites en la institución, además de ser flexible cuando un trámite no tenga un flujo definido.

El sistema que será desarrollado, acompañara al proceso que se hace en la institución para la recepción, avance, y finalización de los trámites, haciendo un registro de cada paso que siguen estos.

Tomando en cuenta la importancia sobre el conocimiento del proceso o avance del trámite para las personas que lo inician o solicitan, uno de los módulos que se desarrollaran permitirá a cualquier persona que haya iniciado un trámite, hacer un seguimiento del mismo, pudiendo ingresar en la aplicación web su número de CI, y un código de seguimiento asignado a su trámite, y de acuerdo al flujo del trámite si tiene un flujo definido o no podrá informarse del tiempo aproximado para la finalización de su trámite.

1.2. ANTECEDENTES DE LA INVESTIGACIÓN

1.2.1. Antecedentes Institucionales

La Universidad Pública de El Alto es una Universidad pública y autónoma de Bolivia, con sede en la ciudad de El Alto del departamento de La Paz. La institución cuenta con 8 áreas y 37 carreras en las áreas de medicina, ingeniería, socio-político-

económica y tecnología. Su funcionamiento se enmarca en el estatuto orgánico aprobado en 2007.

La sede central de la institución está ubicada en la Av. Sucre y Av. Juan Pablo II del municipio de El Alto del departamento de La Paz, cuenta además con sedes en diferentes municipios del departamento de La Paz como ser: Achacachi, Caranavi, Viacha, Ancoraimes, Cruz Loma, Mapiri, Palos Blancos, San Antonio y Batallas.

- Misión

La Universidad Pública de El Alto tiene la misión de formar profesionales altamente capacitados en todas las disciplinas del conocimiento científico-tecnológico, con conciencia crítica y reflexiva; capaz de crear, adaptar y transformar la realidad en que vive; desarrollar una investigación productiva para fomentar el desarrollo local, regional y nacional para que responda al encargo social y las necesidades de las nacionalidades de manera eficiente y oportuna hacia la transformación revolucionaria de la sociedad.

- Visión

La Universidad Pública de El Alto, es una institución que se proyecta al desarrollo de sus actividades académico-productivas, científicas, tecnológicas de interacción social contemporáneo, para priorizar la investigación científica en todos los campos del conocimiento relacionando la teoría con la práctica para transformar la estructura económica, social, cultural u política vigente en favor de las necesidades originarias y clases populares.

1.2.2. Antecedentes De Trabajos Afines

El Sistema Web de Administración Académica incorpora nuevos elementos conceptuales y para consolidarlos se busca antecedentes de prácticas con las metodologías utilizadas destacando los siguientes:

- (Chávez, 2014) “Sistema de Información vía Web para el seguimiento y Control de trámites CASO: Escuela Superior de Formación de Maestros Simón Bolívar”, Proyecto desarrollado en la Universidad Mayor de San Andrés para la Escuela Superior de Formación de Maestros Simón Bolívar. Su objetivo es proporcionar información oportuna, integra y consolidada sobre el estado de los trámites solicitados por los maestros o apoderados.
- (Choque, 2009), “Sistema de control y seguimiento de trámites vía web para el Ministerio de Trabajo”, Proyecto desarrollado en la Universidad Mayor de San Andrés, tiene como objetivo mejorar la atención a los tramitantes proporcionándoles información óptima y confiable respecto a su trámite.
- (Bastidas, 2016), “Desarrollo e implementación del Sistema de Tramite Documentado en la Municipalidad Provincial de Huancayo para la atención de expedientes”, Proyecto desarrollado en la Universidad Nacional del Centro del Perú, tiene como objetivo desarrollar e implementar el sistema de trámites documentado en la Municipalidad Provincial de Huancayo para la atención de expedientes.
- (Calle, 2014), “Sistema de registro y seguimiento de trámites para personalidades jurídicas, Gobierno Autónomo Departamental de La Paz”, Proyecto desarrollado en la Universidad Mayor de San Andrés, tiene por objetivo el registro, otorgación de resolución administrativa para la Personería Jurídica y custodia de la documentación.

1.3. PLANTEAMIENTO DEL PROBLEMA

De acuerdo a un análisis que se hace al proceso que sigue un trámite en la Universidad Pública de El Alto, se percibe que para recibir un trámite y asignarle una hoja de ruta, se utiliza un formulario hecho en Excel para ingresar los datos del trámite, y reimprimir hojas de ruta de trámites ya existentes, este programa se utiliza en cada unidad de la institución, y además se registran trámites que salen o entran a la unidad de forma manual en libros de entrada y salida.

En una situación ideal existiría un sistema que unifique y facilite el registro de trámites y sus hojas de ruta sin depender de un programa para cada unidad, que permita a cada unidad contar con un registro automático sobre la salida o entrada de trámites a sus unidades, además de acceso oportuno al estado y seguimiento sobre los mismos trámites.

De continuar sin cambios en la institución, se continuará destinando recursos económicos en libros de registro físicos para cada unidad y su almacenamiento, persistirá el riesgo de pérdida de información al utilizar un programa Excel que puede ser eliminado, perdido, robado o corrompido, y no se tendrá un acceso oportuno a la información sobre los trámites, cuando esta sea requerida tanto por el personal administrativo como por la persona que solicita el trámite.

Con el presente proyecto se pretende, crear un sistema web que unifique el registro de trámites, registre la salida y entrada de los mismos en cada unidad, permita el acceso oportuno a la información de cada trámite, y este disponible fácilmente para los administrativos.

1.3.1. Problema Principal

Al analizar el sistema actual para el manejo de trámites de la Universidad Pública de El Alto, se evidencia que los procesos manuales empleados no contemplan metodologías de manejo de información, haciendo difícil que la persona que solicita el trámite tenga conocimiento del avance del mismo.

¿De qué manera se puede brindar información oportuna y confiable sobre el avance de tramites a las personas que lo solicitan?

1.3.2. Problemas Secundarios

- Frustración por parte de la comunidad universitaria que solicita trámites debido a lentitud del proceso.

- Mayor consumo de tiempo de los administrativos al gestionar los registros, también al obtener información sobre los trámites.
- Debido a la forma manual de manejo de los trámites, las unidades que conforman la institución no tienen un acceso fácil y rápido cuando se desea buscar el registro o ruta de un trámite.
- Al tener un cuaderno independiente, el personal de cada unidad no puede conocer con anticipación que tramites arribaran.
- Con el proceso manual, la persona que solicita el trámite no puede obtener información sobre el estado y el avance del mismo.
- Decaída de la imagen institucional.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un Sistema Web de seguimiento de trámites para la Universidad Pública de El Alto, destinado a proporcionar acceso a información confiable y oportuna sobre el avance de trámites, y facilitar al personal administrativo el registro y manejo de trámites que reciben.

1.4.2. Objetivos Específicos

- Análisis de Requerimientos para reconocer y entender las verdaderas necesidades que el sistema debe solucionar.
- Diseño del Sistema para definir la estructura que tendrá el sistema web.
- Codificación del Software para llevar a código fuente la estructura del sistema.
- Pruebas para verificar y asegurar el funcionamiento del sistema bajo distintas circunstancias.
- La Instalación o Fase de Implementación para la realización de las pruebas iniciales y correcciones necesarias en caso de ser necesario.

- El Mantenimiento correctivo para la corrección de errores y preventivo para asegurar el buen funcionamiento del sistema.

1.5. JUSTIFICACIÓN DEL TEMA

De acuerdo a los problemas que presenta el proceso manual que se utiliza en la institución, es necesario diseñar un Sistema de Seguimiento de Trámites, para dar solución a los problemas existentes.

1.5.1. Justificación Técnica

Para la implementación de este sistema web, el usuario no necesitará equipos de última tecnología para acceder al sistema, simplemente se necesitará un ordenador o dispositivo de los que ya dispone la institución. Por otra parte, la institución ya dispone de servidores preparados y que tienen las herramientas instaladas para alojar el sistema.

1.5.2. Justificación Económica

El sistema permitirá la optimización del tiempo que el personal administrativo destina a:

- Registrar trámites y crear su hoja de ruta.
- Obtener información sobre el estado y avance del trámite.
- Registrar la salida y entrada de trámites a su unidad.

Todo esto debido a que para la implementación del sistema no será necesario la adquisición de nuevos equipos, y al ser un sistema WEB incluso podrá utilizarse desde un dispositivo móvil que tenga un navegador y acceso a internet.

Además, el sistema también ahorrara tiempo a la persona que solicite el trámite, cuando esta necesite información sobre el estado y avance de su trámite, al poder acceder a esta información desde cualquier dispositivo u ordenador con un navegador instalado y acceso a internet.

Por otra parte, el sistema será desarrollado en PHP (*HYPertext PRE-PROCESSOR*) y el framework Laravel, ambos de distribución libre, lo que no significa un costo en la adquisición de licencias por parte de la institución.

1.5.3. Justificación Social

El sistema agilizará el manejo de trámites en la institución, beneficiando a:

- Personal administrativo. - al reducir la dificultad y la frustración generada con el actual proceso que se sigue en la institución.
- Comunidad universitaria. - tanto estudiantes, docentes o los mismos administrativos, al permitirles hacer un seguimiento oportuno a los trámites que soliciten, y reducir la frustración generada al tratar de hacer seguimiento a sus trámites en la situación actual.

1.6. METODOLOGÍA

1.6.1. UWE

UWE es una metodología de ingeniería del software para el desarrollo de aplicaciones web basado en UML prestando especial atención en la sistematización y personalización. Los modelos más usados de esta metodología son el modelo lógico-conceptual, navegacional, presentación, visualización de escenarios web, interacción temporal y diagrama de estado, secuencia, colaboración, actividad, entre otros. (München, 2016)

Las fases que contempla la metodología son:

- Análisis de Requerimientos.
- Diseño del Sistema.
- Codificación del Software.
- Pruebas.
- La Instalación o Fase de Implementación.
- El Mantenimiento.

1.6.2. Métricas de calidad

La ingeniería del software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad algún producto de software. (Pressman, 2010).

1.6.2.1. ISO/IEC 25000

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto.

1.6.2.2. Seguridad del software ISO-27000

Es un Sistema de Gestión la Seguridad de la Información

ISO 27000 es una norma internacional que permite el aseguramiento, la confidencialidad e integridad de los datos y de la información, así como de los sistemas que la procesan.

1.6.3. Métodos de recolección de datos

- La entrevista, es en esencia, una conversación bien planificada. En ella, el investigador plantea una serie de preguntas o temas de debate a una o varias personas, con el fin de obtener información específica. (Morella, 2010, pag.150)
- La observación es una técnica que consiste precisamente en observar el desarrollo del fenómeno que se desea analizar. Éste método puede usarse para obtener información cualitativa o cuantitativa de acuerdo con el modo en que se realiza. (Aguilar, 2013, pag.204)

1.6.4. Estimación de costos – COCOMO II

El Modelo Constructivo de Costes es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres sub modelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que

avanza el proceso de desarrollo del software: básico, intermedio y detallado. (Calero, 2010).

1.7. HERRAMIENTAS

1.7.1. Hypertext Pre-Processor “PHP”

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (The PHP Group, 2018).

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

1.7.2. Apache

Es un servidor web HTTP de código abierto para plataformas Unixlike (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. (GDR Soluciones, 2017).

1.7.3. Mysql

Es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para crear base de datos con acceso desde páginas webs dinámicas, para la creación de sistemas de transacciones on-line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas. (Cobo, Gómez, Pérez, & Rocha, 2005)

1.7.4. Laravel

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. (Andrés, 2014).

1.7.5. Javascript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

1.7.6. jQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

1.7.7. CSS

En español "*Hojas de estilo en cascada*" es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado

1.7.8. Materialize

Materialize es un framework CSS basado en Material Design, un concepto creado por Google para unificar el diseño de sus aplicaciones sin importar la plataforma en que se muestren.

1.8. LIMITES Y ALCANCES

1.8.1. Limites

- El sistema no podrá reemplazar el uso de documentos físicos adicionados a los trámites (informes, etc.) en su ruta.

- El sistema no podrá reemplazar el uso de “Hojas de Ruta” en formato físico.

1.8.2. Alcances

El sistema propone acceso a información sobre el seguimiento de un trámite a la persona que lo solicite, una manera sencilla de registrar los trámites nuevos, así como su despacho, recepción, o finalización:

- Módulo de seguimiento, que permite obtener información sobre la ruta que siguió un trámite, y verificar si ya finalizó o no.
- Módulo de notificación, que permite notificar en tiempo real a los administrativos sobre el despacho de un trámite a su persona.
- Módulo de mensajería, que permite la interacción mediante mensajes de texto en tiempo real a los administrativos de la institución.
- Módulo de manejo de Hoja de Ruta, que permite la creación, despacho, recepción, o finalización de la Hoja de Ruta de los trámites.

1.9. APORTES

El principal aporte es el de crear un sistema web de información para el seguimiento y control de trámites, que maneje la información sobre estos, y proporcione información oportuna sobre el seguimiento de este, además de permitir a los administrativos de cada unidad de la institución contar con un registro digital sobre los trámites que salen o ingresan a su unidad y permitir mejor comunicación entre las unidades a las que pertenecen.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se definirán los conceptos y definiciones para lograr los objetivos, en el desarrollo del proyecto con bases teóricas de diferentes fuentes donde se extrajo la información.

2.1. CONCEPTOS BÁSICOS

2.1.1. SEGUIMIENTO

El seguimiento consiste en el análisis y recopilación sistemáticos de información a medida que avanza un trámite. Su objetivo es mejorar la eficacia y efectividad de un trámite. Se basa en metas establecidas y actividades planificadas durante las distintas fases del trabajo de planificación. Ayuda a que se siga una línea de trabajo, y además permite a la administración conocer cuando algo no está funcionando. (Alianza Mundial para la Participación Ciudadana, s.f.)

2.1.2. TRÁMITE

Un trámite es cualquier solicitud o entrega de información que las personas físicas o morales del sector público o privado hagan ante una dependencia u organismo, ya sea para cumplir una obligación, obtener un beneficio o servicio o, en general, afín de que se emita una resolución, así como cualquier documento que dichas personas estén obligadas a conservar. (COMISIÓN FEDERAL DE MEJORA REGULATORIA, 2011)

2.1.3. SISTEMA DE INFORMACIÓN

Un sistema de información, es un conjunto integrado de componentes para recopilar, almacenar y procesar datos y para proporcionar información, conocimiento y productos digitales. Las empresas comerciales y otras organizaciones dependen de los sistemas de información para llevar a cabo y administrar sus operaciones, interactuar con sus clientes y proveedores y competir en el mercado. Los sistemas de información se utilizan para ejecutar cadenas de suministro inter-organizacionales y mercados electrónicos.

El primer sistema de información mecánico a gran escala fue el tabulador del censo de Herman Hollerith. Inventado a tiempo para procesar el censo estadounidense de 1890, la máquina de Hollerith representó un gran paso en la automatización, así como una inspiración para desarrollar sistemas de información computarizados. (Zwass, 2001)

2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

La metodología de desarrollo de software implica dividir la carga de trabajo en segmentos específicos para su gestión más rápida. También describe los métodos de gestión utilizados para aumentar la eficiencia de la entrega, la cooperación entre los equipos de desarrollo, diseño, pruebas y otros, así como el propietario del producto. Las metodologías también regulan la participación de partes involucradas fuera del equipo, como clientes e interesados. Los participantes del proyecto saben cómo y cuándo controlar el proyecto sin interferir con el equipo.

Cada metodología presenta un conjunto de métricas y umbrales críticos que muestran la eficiencia del equipo. Además, lo último, pero no menos importante: la metodología no es solo un conjunto de instrumentos, sino una mentalidad para la transformación digital.(Naumenko, s.f.)

2.3. UWE

UWE es un método de ingeniería del software para el desarrollo de aplicaciones web basado en UML.

El enfoque UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de UML en "lingua franca", sino que también usa XMI como formato de intercambio de modelos, MOF para meta-modelado, los

principios basados en modelos del enfoque MDA, el lenguaje de transformación de modelos QVT y XML.

Las principales razones para utilizar los mecanismos de extensión de UML en lugar de técnicas de modelado propietarias son la aceptación de UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio web: un llamado perfil UML, y Amplio soporte de modelado visual mediante herramientas UML CASE existentes.

UWE utiliza notación UML "pura" y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones web, es decir, sin extensiones de ningún tipo. Para las características específicas de la Web, como nodos y enlaces de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre aspectos de navegación, presentación, procesos comerciales y adaptación. La notación UWE se define como una extensión "ligera" de UML. (München, 2016)

2.3.1. Modelos de la UWE

El método UWE consiste en la construcción de los siguientes modelos de análisis y diseño:

2.3.1.1. Modelo de requerimientos: En UWE el modelado de requerimientos consiste en dos partes:

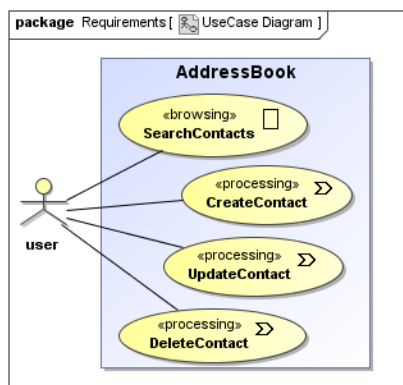


Figura 1 Diagrama de caso de uso

Fuente: (München, 2016)

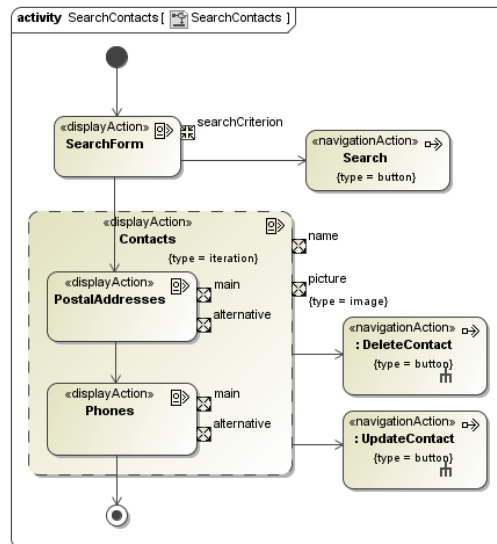


Figura 2 Diagrama de actividades

Fuente: (München, 2016)

2.3.1.2. Modelo de contenido: El modelo de contenido para aplicaciones Web en UWE, utiliza modelos estándares UML para modelar estructuras de clases, asociaciones y paquetes.

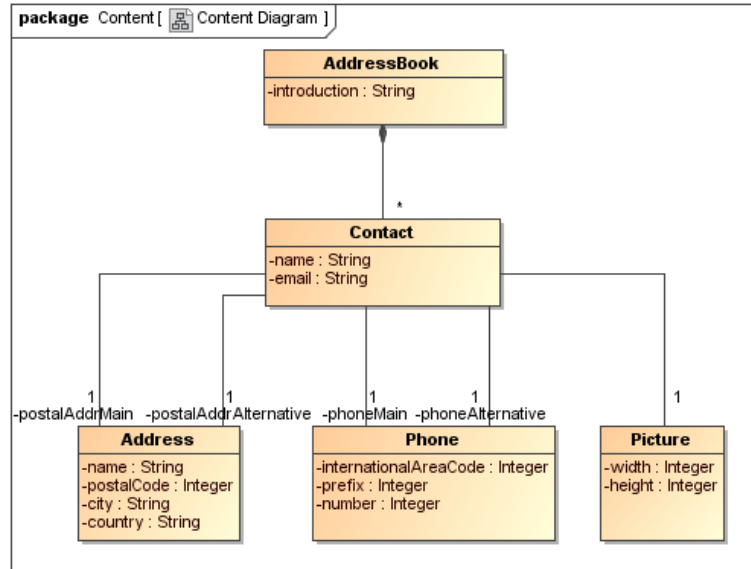


Figura 3 Diagrama de clases
Fuente: (München, 2016)

2.3.1.3. Modelo de navegación: El objetivo del modelo de navegación es el de especificar las rutas de navegación del contenido de una aplicación Web.

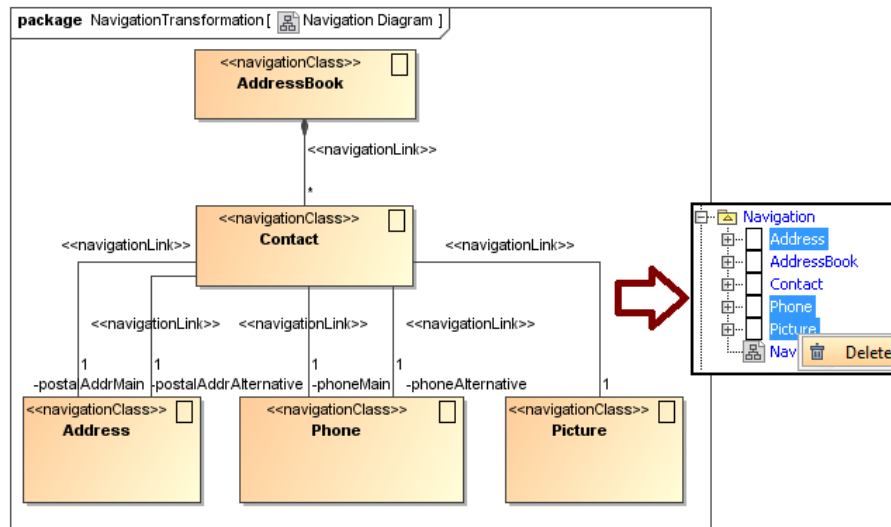


Figura 4 Diagrama de navegación
Fuente: (München, 2016)

2.3.1.4. Modelo de presentación: El modelo de presentación define la apariencia que pueden tomar los modelos de navegación.

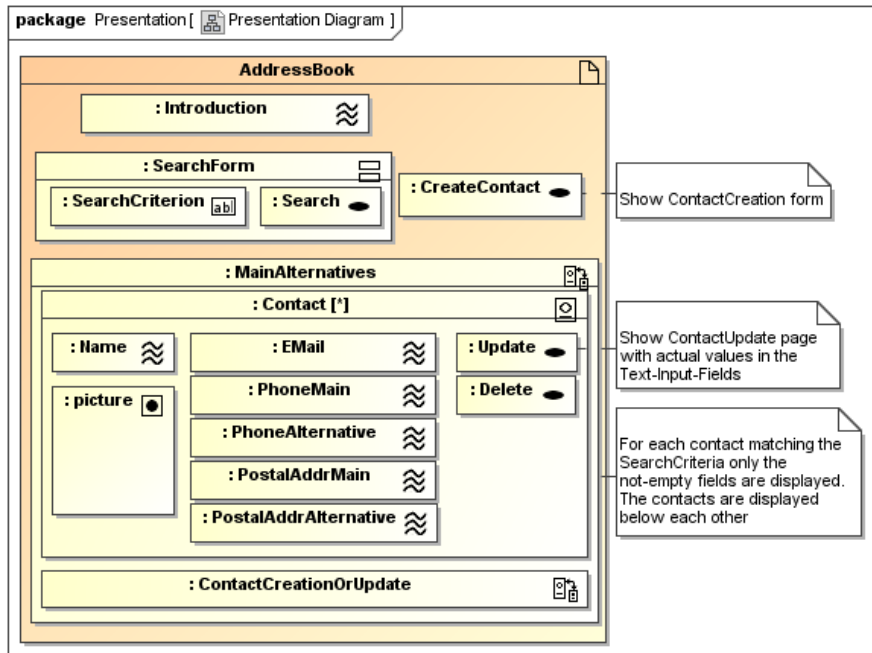


Figura 5 Diagrama de presentación

Fuente: (München, 2016)

2.3.1.5. Modelo de procesos: Los procesos proveen a una aplicación Web aspectos dinámicos. Un proceso está compuesto por uno o más subprocessos y/o acciones que el usuario ejecuta para alcanzar una meta. El modelo de procesos comprende:

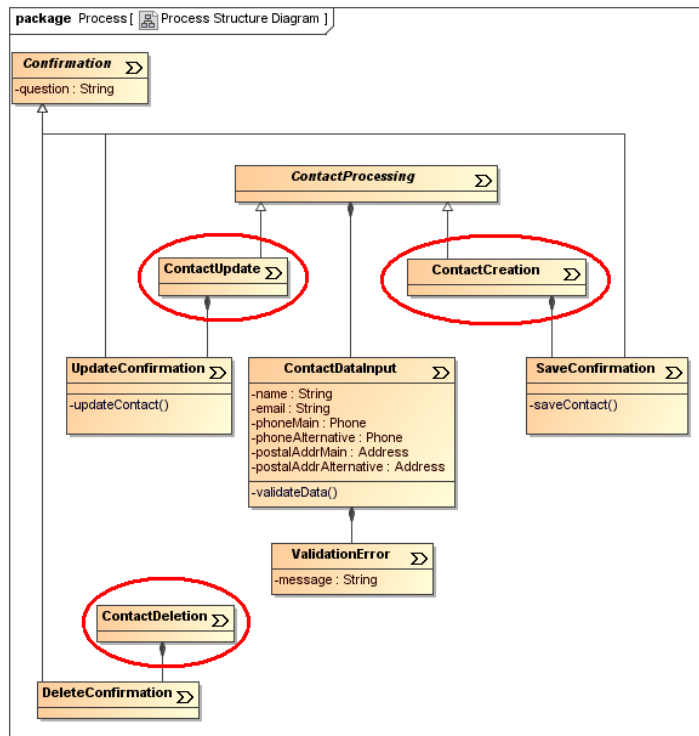


Figura 6 Modelo de estructura de procesos que describe las relaciones entre las diferentes clases de procesos

Fuente: (München, 2016)

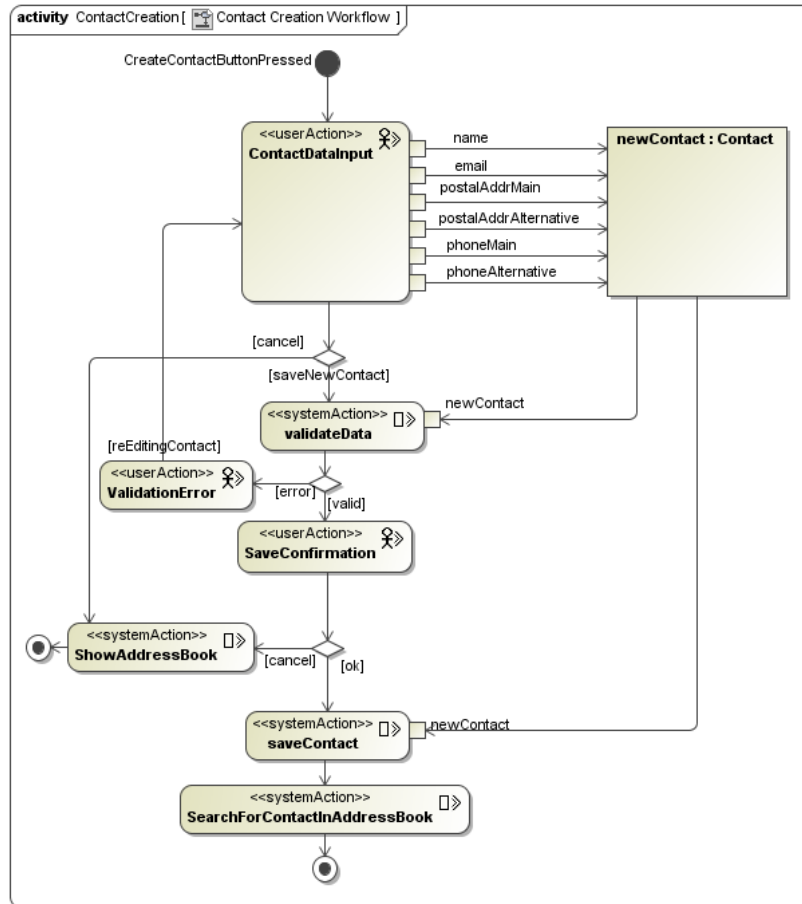


Figura 7 Modelo de Flujo de Procesos que especifica las actividades relacionadas con cada proceso

Fuente: (München, 2016)

2.4. HERRAMIENTAS

2.4.1. Laravel

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. (Andrés, 2014).

Características

- Consola artisan, que permite realizar tareas de programación repetitivas.
- Autenticación, que proporciona una forma sencilla de organizar la lógica de autorización y controlar el acceso a los recursos.

- Blade, que es un sistema de procesamiento de plantillas.
- ORM Eloquent (Mapeo objeto-relacional), para manejar los procesos correspondientes al manejo de bases de datos.

2.4.2. jQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

Características

- Constructor JQuery(\$) que permite el acceso a los elementos del DOM(Modelo de Objeto de Documento).
- Trabajo por grupos, los métodos pueden realizar consultas mediante iteración implícita.
- Anidación de consultas o tareas de forma encadenada.

2.4.3. Materialize

Materialize es un framework CSS basado en Material Design, un concepto creado por Google para unificar el diseño de sus aplicaciones sin importar la plataforma en que se muestren.

Características

- La interacción de los objetos en el espacio es transmitida a través de principios de la luz, la superficie y el movimiento.
- Permite la creación de interfaces que se adaptan a distintas resoluciones de pantalla y dispositivos.

- Influenciado por elementos de diseño de impresión que da al usuario una experiencia inmersiva.

2.5. ISO/IEC 9126

“Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation-Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126). Este estándar describe 6 características 41 generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad para el software se describen así:

Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores

del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así:



Figura 8 Gráfico de Norma de Evaluación ISO/IEC 9126

Fuente: (ARDILA, 2013)

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en varios indicadores; estas se pueden medir por métrica interna o externa.



Figura 9 Gráfico de Evaluación Interna, Externa y Calidad de Uso ISO/IEC 9126

Fuente: (ARDILA, 2013)

Las definiciones se dan para cada característica y sub-característica de calidad del software que influye en la calidad. Para cada característica y sub-característica, la capacidad del software es determinada por un conjunto de atributos internos que

pueden ser medidos. Las características y sub-características se pueden medir externamente por la capacidad del sistema que contiene el software.” (Borbón Ardila , 2013)

2.5.1. Funcionalidad

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. A continuación, se muestra la característica de Funcionalidad y las sub-características que cubre:



Figura 10 Gráfico de Característica de funcionalidad.

Fuente: (ARDILA, 2013)

Ecuación de funcionalidad:

$$PF = CuentaTotal * (0,65 + 0,01 * \sum Fi)$$

Donde:

PF: Medida de funcionalidad

Cuenta total: es la suma del producto del factor de ponderación y valores de los parámetros.

$\sum Fi$: Es la sumatoria de los valores de ajuste de la complejidad.

2.5.2. Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este

caso a la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.



Figura 11 Gráfico Característica de Confiabilidad.

Fuente: (ARDILA, 2013)

Es posible expresar la confiabilidad de acuerdo a la siguiente ecuación:

$P(T \leq t) = F(t)$: Probabilidad de fallas (el termino en el cual el sistema trabaja sin falla)

$P(T \leq t) = 1 - F(t)$: Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el que se ejecuta y se obtiene muestras.

$$F(t) = f * e^{-\mu * t}$$

Dónde:

f : Funcionalidad del sistema.

μ : Es la probabilidad de error que puede tener el sistema.

t : Tiempo que dura una gestión en el sistema.

2.5.3. Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.



Figura 12 Gráfico de Característica de Usabilidad.

Fuente: (ARDILA, 2013)

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

$$FU = [(\sum Xi n) * 100$$

Dónde:

Xi: es la sumatoria de valores

n: es el número de preguntas

2.5.4. Eficiencia

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.



Figura 13 Característica de Eficiencia.

Fuente: (ARDILA, 2013)

2.5.5. Capacidad de mantenimiento

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.



Figura 14 Gráfico de Característica de Mantenimiento

Fuente: (ARDILA, 2013)

Se determina la siguiente ecuación:

$$IMS = [Mt - (Fa + Fb + Fc)] / Mt$$

Dónde:

MT= Número de módulos en la versión actual.

Fa= Número de módulos en la versión actual que se han cambiado.

Fb= Número de módulos en la versión actual que se han añadido.

Fc= Número de módulos en la versión anterior que se han borrado en la versión actual.

A medida que el sistema se aproxima a 1 el producto se pone más estable según la siguiente relación:

75% <= IMS <= 100% → Optima

50% <= IMS <= 75% → Buena

25% <= IMS <= 50% → Suficiente

0% <= IMS <= 25% → Deficiente

2.5.6. Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro.



Figura 15 Gráfico de Característica de Portabilidad.

Fuente: (ARDILA, 2013)

Para poder medir la portabilidad del sistema usaremos la siguiente fórmula que indica el grado de portabilidad que tiene un software:

$$GP = 1 - (I ER)$$

Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

2.5.7. Calidad de uso

Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud. La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.



Figura 16 Gráfico de Característica Calidad de uso.

Fuente: (ARDILA, 2013)

2.6. PRUEBAS DE SOFTWARE

La industria del software reconoce hoy la importancia de llevar a cabo pruebas de software, como instrumento para asegurar la calidad de los productos desarrollados así, las empresas que forman esta industria intentan protegerse, en cierta medida, de los riesgos de litigios, sobrecostos y pérdida de reputación ante los clientes. La industria del software está conformada, en su mayoría, por pequeñas organizaciones (el término “pequeñas organizaciones” en este artículo equivale al de “very small entities” –vses–, que denota a una empresa, organización, departamento, grupo o proyecto que cuenta con menos de 25 personas); entonces, es importante abordar trabajos que fortalezcan el quehacer de este tipo de empresas, con el fin de que puedan tener mayor competitividad. Las pequeñas organizaciones del software necesitan llevar a cabo sus pruebas de manera planeada y sistemática, con el fin de promover el desarrollo de software de calidad y, así, obtener ventajas competitivas. El mercado valora, cada día más, la calidad, por lo tanto, las compañías exigen la disminución de errores, y penalizan los retrasos en entregas y las cancelaciones de proyectos. Esta situación conduce a que las pequeñas organizaciones cambien su orientación hacia las pruebas de software, y las consideren un proceso necesario para apoyar la calidad de sus productos. La ingeniería del software cuenta con la premisa: “la calidad del producto depende en gran medida de la calidad del proceso”, que destaca la importancia de

los procesos para proporcionar software de alta calidad. Estos procesos (incluyendo el de pruebas de software), que involucran personas, aspectos organizacionales y procedimientos utilizados para definirlos, ejecutarlos y mantenerlos, forman la base sobre la cual se sustenta el desarrollo exitoso de productos software de alta calidad, por lo que contribuir a su implantación adecuada por pequeñas organizaciones debe ser un propósito fundamental. (Rojas-Montes, Pino-Correa, & Martínez, 2015)

2.6.1. Método de la caja negra

Es un método con enfoque principalmente funcional, se centra en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiende a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario sin que esto signifique que no sean útiles en cualquier módulo del sistema

Las pruebas de caja negra se apoyan en la especificación de requisitos del módulo. De hecho, se habla de "cobertura de especificación" para dar una medida del número de requisitos que se han probado. Es fácil obtener coberturas del 100% en módulos internos, aunque puede ser más laborioso en módulos con interfaz al exterior. En cualquier caso, es muy recomendable conseguir una alta cobertura en esta línea.

El problema con las pruebas de caja negra no suele estar en el número de funciones proporcionadas por el módulo (que siempre es un número muy limitado en diseños razonables); sino en los datos que se les pasan a estas funciones. El conjunto de datos posibles suele ser muy amplio (por ejemplo, un entero).

Los beneficios permiten identificar claramente las entradas, salidas, estudiar las relaciones que existen entre ellas, permitiendo así maximizar la eficiencia, se

centran en lo que se espera de un módulo, concentra su atención en el dominio de la información.

Mediante las técnicas de prueba de la caja negra se derivan un conjunto de casos de prueba que satisfacen los siguientes criterios:

Casos de prueba que reducen, en un coeficiente que es mayor que uno, el número de casos de prueba adicionales que se deben diseñar para alcanzar una prueba razonable,

Casos de prueba que nos dicen algo sobre la presencia o ausencia de clases de errores asociados solamente con la prueba en particular que se encuentra disponible. (Vásquez, 2012)

2.6.2. Método de la caja blanca

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El probador escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.

Al estar basadas en una implementación concreta, si ésta se modifica, por regla general las pruebas también deberán rediseñarse.

Aunque las pruebas de caja blanca son aplicables a varios niveles, habitualmente se aplican a las unidades de software. Su cometido es comprobar los flujos de ejecución dentro de cada unidad (función, clase, módulo, etc.) pero también pueden testear los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas de sistema.

A pesar de que este enfoque permite diseñar pruebas que cubran una amplia variedad de casos de prueba, podría pasar por alto partes incompletas de la

especificación o requisitos faltantes, pese a garantizar la prueba exhaustiva de todos los flujos de ejecución del código analizado.

Las principales técnicas de diseño de pruebas de caja blanca son:

- Pruebas de flujo de control
- Pruebas de flujo de datos
- Pruebas de bifurcación (branchtesting)
- Pruebas de caminos básicos

Dentro de sus beneficios garantizan que se ejercitan al menos una vez todos los caminos independientes de cada módulo, se ejercitan todas las decisiones lógicas en sus caras verdaderas y falsas, ejecutan todos los bucles en sus límites y con sus límites operacionales, se ejercitan las estructuras de datos internas para asegurar su validez. (Vásquez, 2012)

2.7. COCOMO

Modelo Constructivo de Costos COCOMO es una jerarquía de modelos de estimación de costes de software que incluye sub-modelos básico, intermedio y avanzado.

El modelo COCOMO es uno de los sistemas de estimación de costes más utilizados en proyectos de desarrollo de software. La estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos.

Modelos de estimación

COCOMO básico, calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado.

COCOMO intermedio, Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

COCOMO avanzado, incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase.

2.7.1. COCOMO II

Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto. COCOMO define tres modos de desarrollo o tipos de proyectos:

- Orgánico: proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- Semi-acoplado: proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- Empotrado: proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Y por otro lado existen diferentes modelos que define COCOMO:

- Modelo básico: Se basa exclusivamente en el tamaño expresado en LDC. 31
- Modelo intermedio: Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- Modelo avanzado: Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Para el caso el modelo intermedio se usa las fórmulas siguientes:

- $E = \text{Esfuerzo} = a \text{ KLDC } e * \text{FAE (persona x mes)}$
- $T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo } d \text{ (meses)}$
- $P = \text{Personal} = E/T \text{ (personas)}$

Dónde:

KDLC: es Cantidad de líneas de código, en miles

a,e,c,d: Son constantes con valores definidos, según cada sub-modelo.

FAE: Multiplicador que depende de 15 atributos, que se obtiene de la tabla de valores

T: Tiempo requerido por el proyecto, en meses

P: Número de personas requerido por el proyecto.

2.8. SEGURIDAD

2.8.1. Normas ISO/IEC 27000

La norma ISO/IEC 27000, contiene las mejores prácticas recomendadas en seguridad de la información para desarrollar, implementar y mantener especificaciones para los SGSI. (ISO/IEC 27001, 2005)

2.8.2. Aplicación desde el usuario

A.11.2.1. Inscripción del usuario. (Control)

Debe existir un procedimiento formal para la inscripción y des-inscripción para otorgar acceso a todos los sistemas y servicios de información. (ISO/IEC 27001:2005, p.31)

A.11.2.2. Gestión de privilegios. (Control)

Se debe restringir y controlar la asignación y uso de los privilegios. (ISO/IEC 27001:2005, p.31)

A.11.2.3. Gestión de la clave del usuario. (Control)

La asignación de claves se debe controlar a través de un proceso de gestión formal. (ISO/IEC 27001:2005, p.31)

A.11.2.4. Revisión de los derechos de acceso del usuario. (Control)

La gerencia debe revisar los derechos de acceso de los usuarios a intervalos regulares utilizando un proceso formal. (ISO/IEC 27001:2005, p.31)

2.8.3. Acceso al sistema, contraseñas

A.11.1.1. Política de control de acceso. (Control)

Se debe establecer, documentar y revisar la política de control de acceso en base a los requerimientos de seguridad y comerciales. (ISO/IEC 27001:2005, p.31)

2.8.4. Acceso a la base de datos

A.10.7.1. Gestión de los medios removibles. (Control)

Deben existir procedimientos para la gestión de medios removibles. (ISO/IEC 27001:2005, p.30)

A.10.7.2. Eliminación de medios. (Control)

Los medios deben ser eliminados utilizando procedimientos formales y de una manera segura cuando ya no se les requiere. (ISO/IEC 27001:2005, p.30)

2.8.5. Backups

Objetivo: mantener la integridad y disponibilidad de los servicios de procesamiento de información y comunicaciones.

A.10.5.1. Back-up o respaldo de la información (Control)

Se deben realizar copias de back-up o respaldo de la información comercial y software esencial y se deben probar regularmente de acuerdo a la política de la empresa. (ISO/IEC 27001:2005, p.29).

CAPÍTULO III

3. MARCO APLICATIVO

En este capítulo se presenta: el análisis, desarrollo e implementación del “Sistema Web Para el Seguimiento de Trámites. Caso: Universidad Pública de El Alto”, utilizando la metodología UWE.

3.1. INTRODUCCIÓN

El presente capítulo se efectúa el proceso de desarrollo del sistema software denominado “Sistema Web para el Seguimiento de Trámites”, caso Universidad Pública de El Alto, aplicando un marco de trabajo para estructurar, planificar y controlar el mismo. Donde se engloba el enfoque de un proceso de desarrollo de software y un conjunto de herramientas libres, modelos y técnicas para desarrollar un software de buena calidad, donde se desarrolla las etapas y modelos correspondientes a la fase de obtención de requisitos, análisis y diseño del sistema, codificación, pruebas y la fase de implantación siguiendo el proceso de desarrollo de la metodología UWE ya detalladas en el capítulo II. Aplicando las diversas fases del modelo y desarrollo del modelo UWE (UMLBased Web Engineering, “Ingeniería Web basado en UML”). Y está compuesta por las siguientes fases: Análisis, Diseño, Desarrollo o codificación, pruebas, implementación y mantenimiento. El modelo que propone UWE son: Modelo de Requerimientos, Modelo Conceptual, Modelo Navegación y Modelo de Presentación que nos representan diversos diagramas y esquemas en un proceso iterativo e incremental dando apoyo al modelo de la aplicación.

3.2. METODOLOGÍA UWE – CONSTRUCCIÓN E IMPLEMENTACIÓN

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una

metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que permite modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

3.2.1. Modelo de Requerimientos

3.2.1.1. Actores

Los actores representan un rol que puede desempeñar alguien o algo que interactúa o que necesita intercambiar información con el sistema.

Para identificar a los Actores se permitió conocer a las personas encargadas en el manejo de trámites de la Universidad Pública de El Alto. En la siguiente tabla se muestra la lista de actores, junto con una descripción de sus actividades relacionadas con el sistema.

ACTOR	DESCRIPCIÓN
JEFE DE UNIDAD / DIRECTOR / DECANO DE AREA	Estos actores están a la cabeza de las distintas unidades, carreras y áreas de la institución, normalmente son los destinatarios de los trámites, y emiten respuestas a los mismos.
SECRETARIA / AUXILIAR	Están encargados de la gestión de los trámites dentro de sus unidades, carreras o áreas. Ellos reciben y registran los trámites que provienen de unidades externas, despachan trámites, generan un registro y control de los trámites que se ingresan o salen de su unidad, y generan nuevos trámites dentro de su unidad.

TÉCNICOS / OTROS ADMINISTRATIVOS	Se les asigna instrucciones como emitir informes, preparar respuestas, agendar reuniones y otras más dependiendo de la unidad
COMUNIDAD UNIVERSITARIA	Conformado por estudiantes, docentes, administrativos, etc de la institución. Pueden solicitar e iniciar trámites dentro de la institución.
PERSONAS O INSTITUCIONES EXTERNAS	Están fuera de la institución, y pueden solicitar e iniciar trámites dentro de la institución

Tabla 1 Lista de actores

Fuente: Elaboración propia

3.2.1.2. Obtención de requisitos

La obtención de requisitos es fundamental para que un sistema sea exitoso para lo cual se realizaron actividades en el que se muestra en la siguiente tabla.

TAREA	CARACTERISTICA
ENTREVISTA	Se realizaron entrevistas con: <ul style="list-style-type: none"> • Jefes de unidad • Auxiliares
OBSERVACION	Presentan dificultades para hacer seguimiento, mantener los registros de hoja de ruta (perdida de información) de los trámites, registro nuevos trámites y creación de nuevos trámites al estar apoyados en un programa en Excel para el proceso, esto sucede en cada unidad.
DOCUMENTACIÓN	Se obtuvo documentación física de trámites y hojas de ruta.

Tabla 2 Obtención de requisitos

Fuente: Elaboración propia

3.2.1.3. Lista de requerimientos del Sistema

A la obtención correcta de los requerimientos puede llegar a describir con claridad en forma consistente, el comportamiento del sistema por ello que se toman en cuenta para diferenciar los conceptos de los requisitos:

3.2.1.3.1. Requerimientos funcionales

Los requisitos funcionales se describen a continuación en la siguiente tabla, donde se muestran las características que necesita el sistema a partir de la información obtenida como parte de las tareas de obtención de requisitos.

DESCRIPCIÓN	FUNCIÓN
ADMINISTRACIÓN DE USUARIOS	Permite administrar los usuarios del sistema, asignarles roles y permisos, además de crear nuevos usuarios, modificarlos, asignarlos a una unidad y eliminarlos.
ADMINISTRACIÓN DE UNIDADES	Permite administrar las unidades del sistema, modificarlas y crear nuevas.
MODULO DE RECUPERACIÓN	Permite a los usuarios cambiar su contraseña si no pueden acceder al sistema, utilizando su dirección de correo electrónico institucional para recibir un email con un enlace de recuperación.
MODULO DE TRÁMITES	Permite crear, recibir, despachar y finalizar trámites.
MODULO DE SEGUIMIENTO	Permite hacer seguimiento de trámites a cualquier persona que esté registrada dentro de la base de datos de la institución.
MODULO DE SEGUIMIENTO BASICO	Permite hacer seguimiento de trámites a las personas que tengan una cuenta de usuario en el sistema.

MODULO DE SEGUIMIENTO DETALLADO	Permite hacer seguimiento detallado de trámites a los usuarios del sistema con el permiso necesario.
MODULO DE CONSULTAS DE TRÁMITES	Permite generar reportes del módulo de trámites.
MODULO DE NOTIFICACIONES	Permite a los usuarios dentro del sistema ser notificados en tiempo real dentro del sistema, cuando se les envíe un trámite.
CUENTA DE USUARIO	Permite al usuario dentro del sistema, agregar una foto, cambiar su contraseña, ver información sobre sus permisos y ver sus datos personales.
OCULTO	Inicio y cierre de sesión.

Tabla 3 Requerimientos funcionales del sistema

Fuente: Elaboración propia

3.2.1.3.2. Requerimientos no funcionales

Los requerimientos no funcionales son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los requerimientos funcionales que especifican los comportamientos específicos.

FUNCIÓN
El sistema debe visualizarse y funcionar correctamente en cualquier navegador como ser internet Explorer, Mozilla, Chrome, etc.
Respaldo energético del servidor, para asegurar la disponibilidad del sistema.
Soporte y mantenimiento periódico para asegurar el buen rendimiento del sistema.

Tabla 4 Requerimientos no funcionales del sistema

Fuente: Elaboración propia

3.2.1.4. Diagrama de Casos de Uso General

Diagrama de Casos de Uso General.

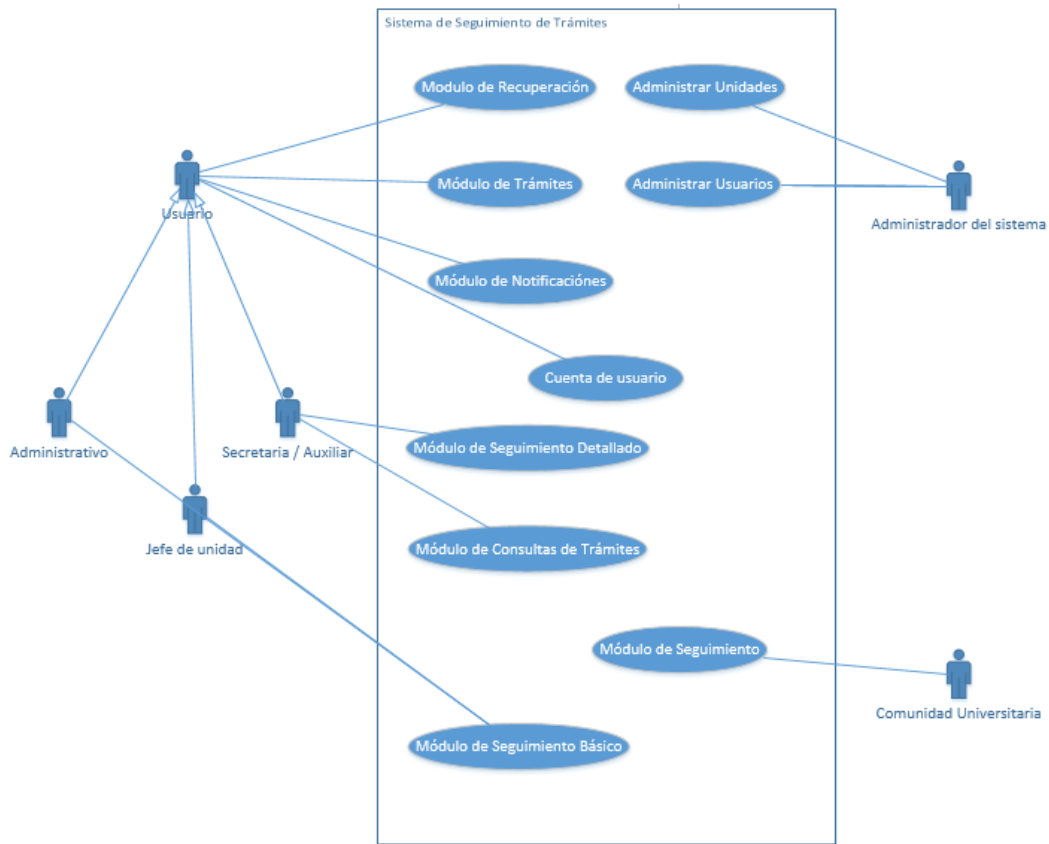


Figura 17 Diagrama de casos de uso general

Fuente: Elaboración propia

3.2.1.5. Diagrama de Caso de Uso: Administración del Sistema

Se muestra el modelo donde se puede apreciar el comportamiento del actor súper administrador sobre la manipulación del sistema: Administración de usuarios, almacén, compras, ventas, caja y reportes y ver las diferentes acciones que realiza.

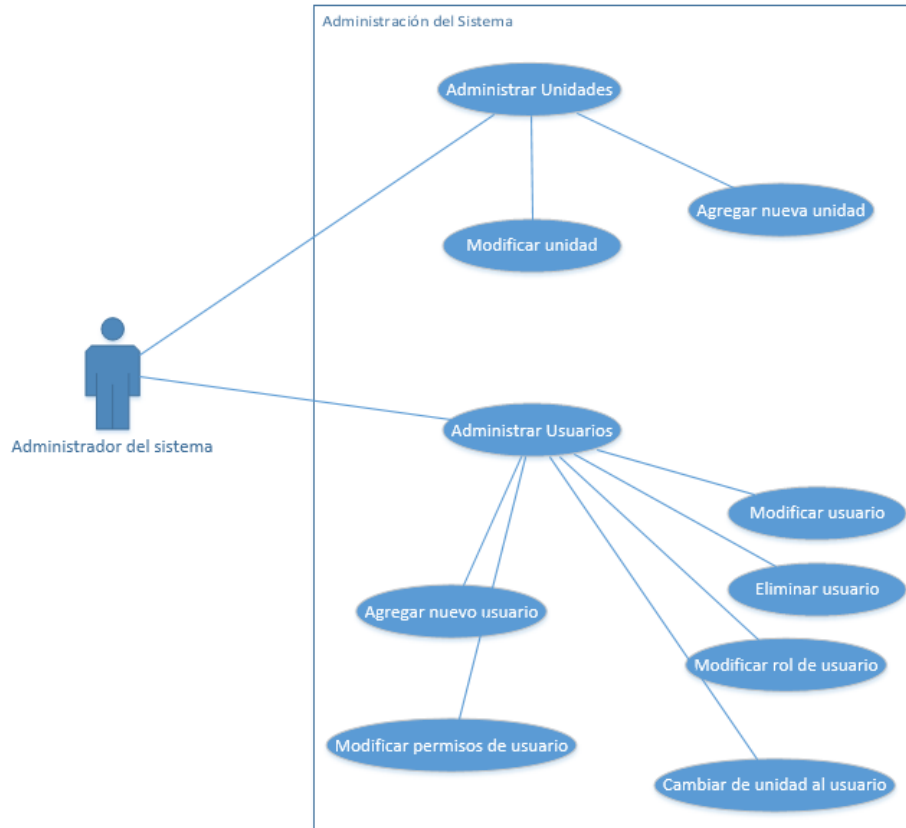


Figura 18 Caso de uso “Administración del Sistema”

Fuente: Elaboración propia

CASO DE USO ADMINISTRACIÓN DEL SISTEMA	
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	<p>Este usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> • Módulo de administración de unidades: Modificar y crear unidades. • Módulo de administración de usuarios: Modificar, crear, eliminar, cambiar rol, cambiar permisos y cambiar de unidad a los usuarios

Tabla 5 Descripción de caso de uso: Administración del Sistema

Fuente: Elaboración propia

3.2.1.6. Diagrama de Caso de Uso específicos: Administración de Unidades

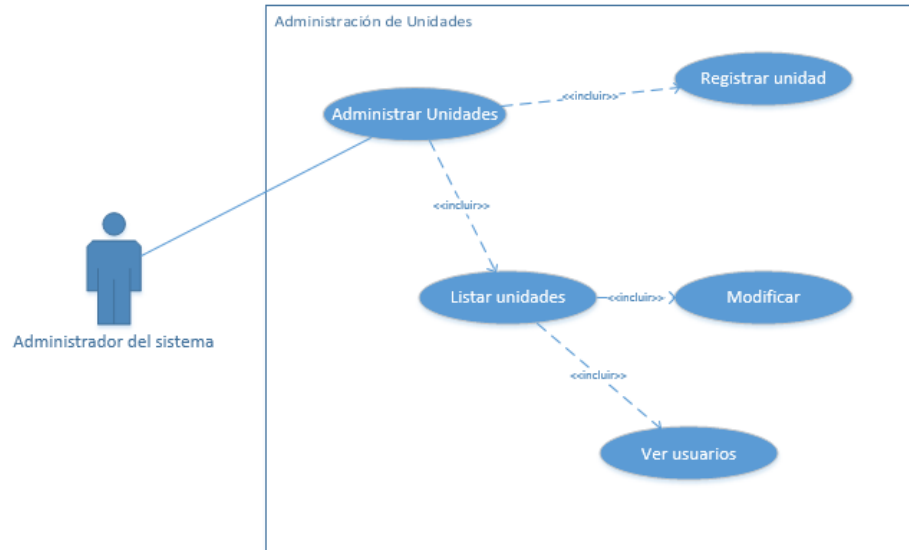


Figura 19 Diagrama de Caso de Uso: Administración de unidades

Fuente: Elaboración propia

CASO DE USO ADMINISTRACIÓN DE UNIDADES	
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	El administrador del sistema registra unidades, lista las unidades existentes, modifica la unidad y puede listar los usuarios de la unidad

Tabla 6 Descripción de caso de uso “Administración de Unidades”

Fuente: Elaboración propia

3.2.1.7. Diagrama de Caso de Uso específicos: Administración de Usuarios

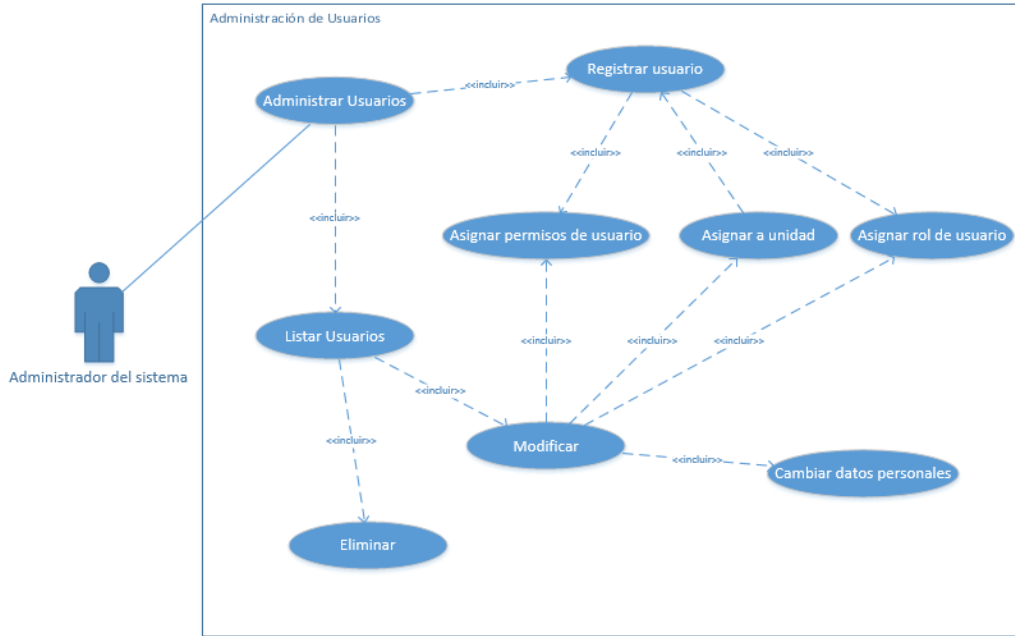


Figura 20 Diagrama de Caso de Uso “Administración de usuarios”

Fuente: Elaboración propia

CASO DE USO	ADMINISTRACIÓN DE USUARIOS
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	El administrador del sistema registra usuarios, lista los usuarios existentes, modificar datos personales, asignar permisos, asignar rol, asignar unidad y eliminar usuarios

Tabla 7 Descripción de caso de uso “Administración de Usuarios”

Fuente: Elaboración propia

3.2.1.8. Diagrama de Caso de Uso: Modulo de Trámites

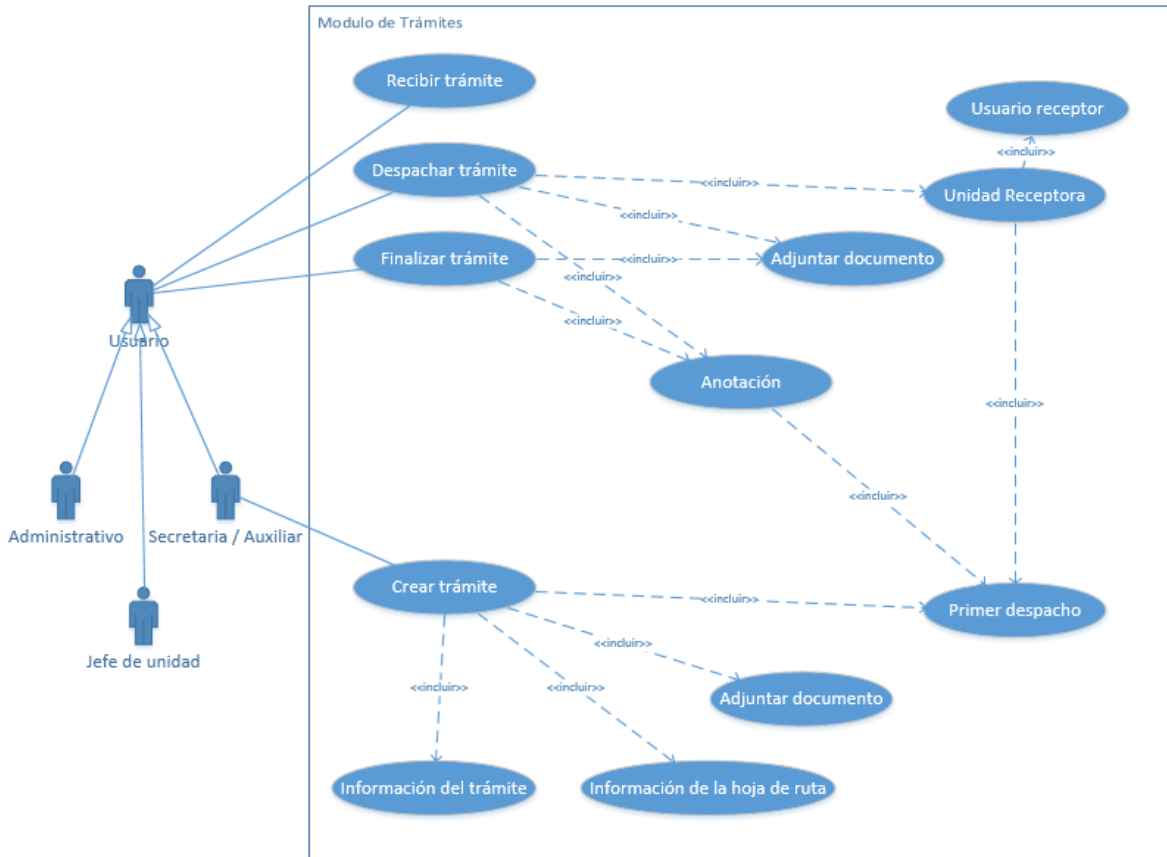


Figura 21 Diagrama de Caso de Uso “Módulo de trámites”

Fuente: Elaboración propia

CASO DE USO	MÓDULO DE TRÁMITES
ACTOR	Usuarios dentro del sistema
TIPO	Primario
DESCRIPCIÓN	<p>Todos los usuarios pueden:</p> <ul style="list-style-type: none"> • Recibir trámites que otros usuarios les hayan enviado • Despachar trámites a otros usuarios del sistema • Finalizar trámites <p>Los usuarios con rol “Secretaria/Auxiliar” pueden:</p> <ul style="list-style-type: none"> • Crear trámites

Tabla 8 Descripción de caso de uso “Módulo de trámites”

Fuente: Elaboración propia

3.2.1.9. Diagrama de Caso de Uso: Modulo de Notificaciones

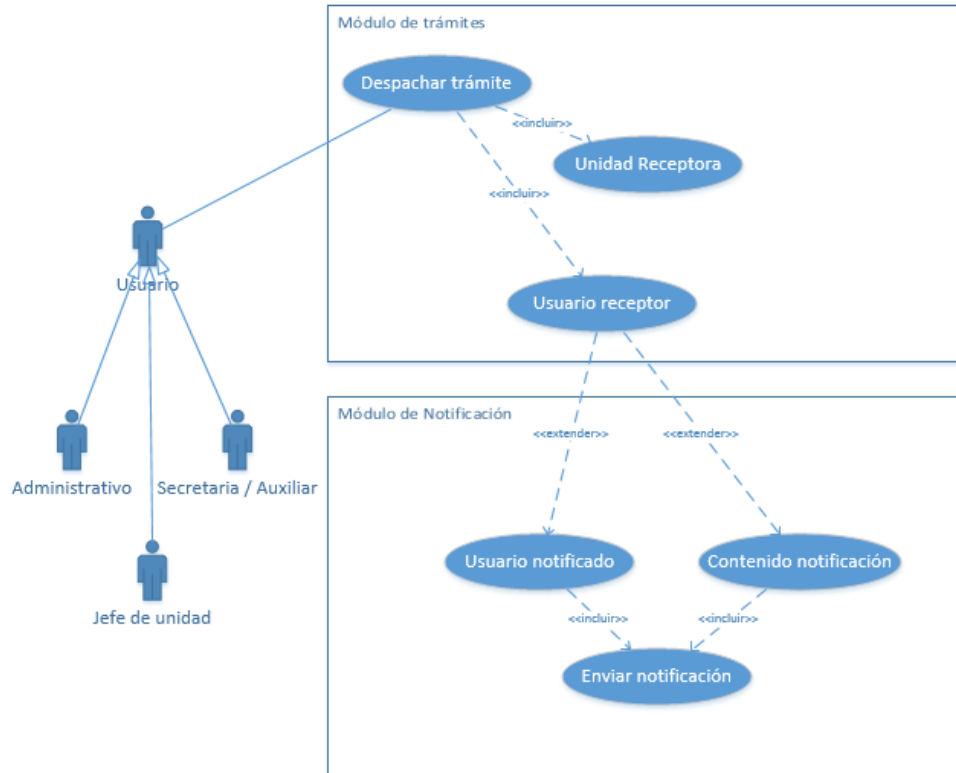


Figura 22 Diagrama de Caso de Uso “Módulo de notificación”

Fuente: Elaboración propia

CASO DE USO	MÓDULO DE NOTIFICACIÓN
ACTOR	Usuarios dentro del sistema
TIPO	Primario
DESCRIPCIÓN	Todos los usuarios pueden reciben notificaciones automáticamente, cuando otro usuario les despacha un trámite, esto incluye el primer despacho de un trámite

Tabla 9 Descripción de caso de uso “Módulo de notificación”

Fuente: Elaboración propia

3.2.1.10. Diagrama de Caso de Uso: Cuenta de Usuario

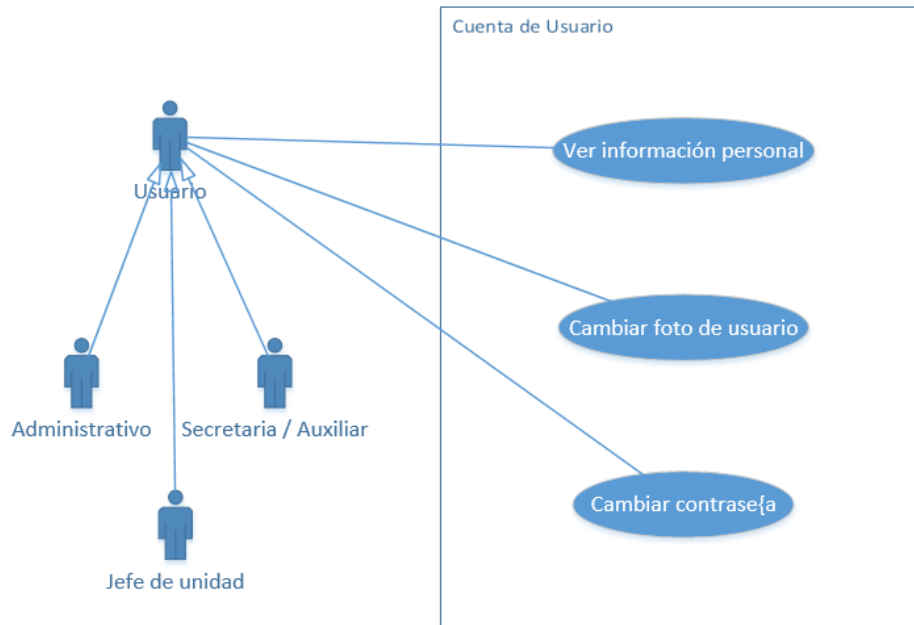


Figura 23 Diagrama de Caso de Uso “Cuenta de Usuario”

Fuente: Elaboración propia

CASO DE USO		CUENTA DE USUARIO
ACTOR		Usuarios dentro del sistema
TIPO		Secundario
DESCRIPCIÓN		Los usuarios pueden ver su información personal, cambiar su contraseña y cambiar su foto de usuario

Tabla 10 Descripción de caso de uso “Cuenta de Usuario”

Fuente: Elaboración propia

3.2.2. Modelo de Contenido

A continuación se muestra o visualiza las relaciones entre las clases que involucran el sistema.

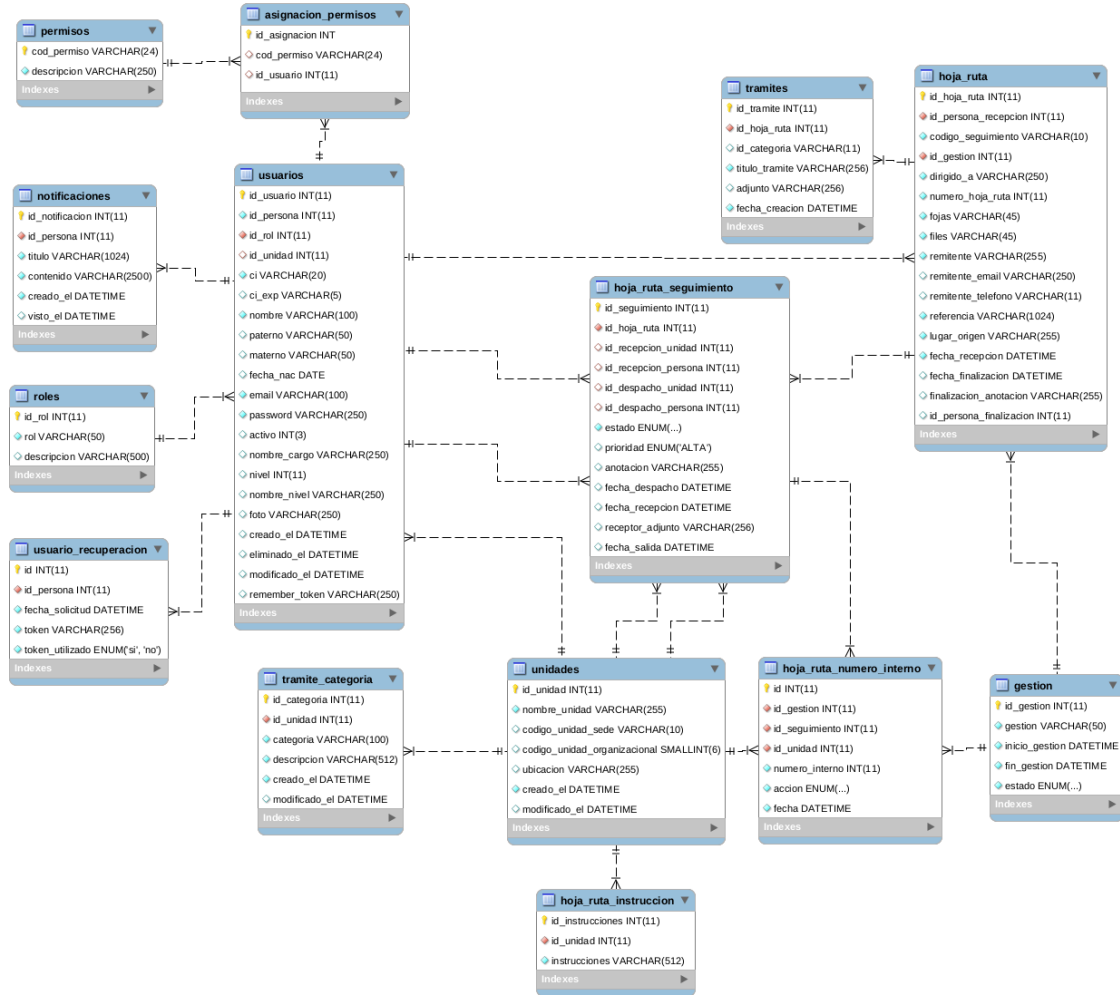


Figura 24 Base de datos
Fuente: Elaboración propia

3.2.3. Modelo de Navegación

A continuación, se hace el modelado donde se aprecia la interacción de los usuarios en la navegación del sistema.

3.2.3.1. Modelo de navegación: Inicio

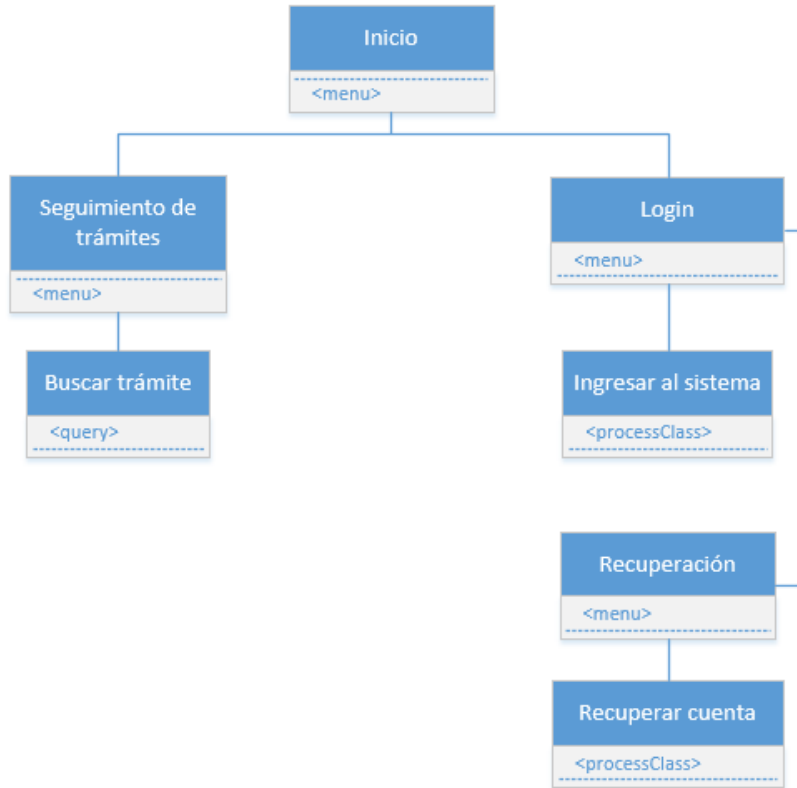


Figura 25 Diagrama de navegación "Inicio"

Fuente: Elaboración propia

3.2.3.2. Diagrama de navegación: Administración

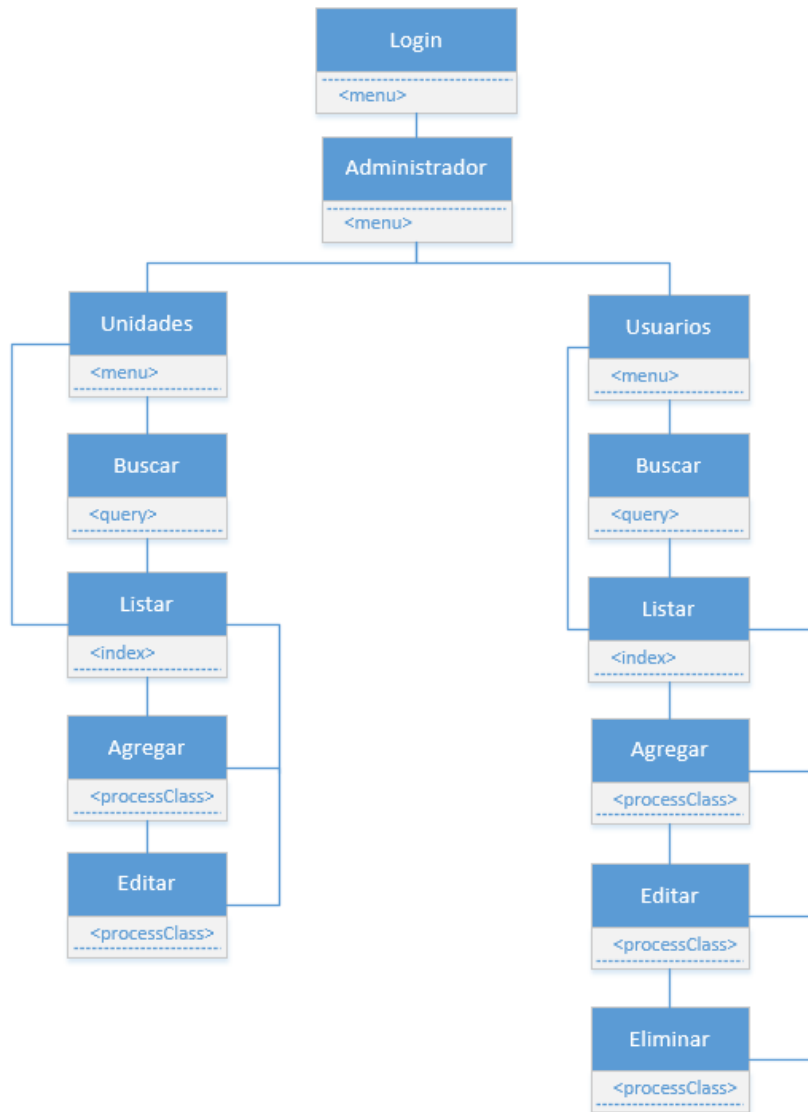


Figura 26 Diagrama de navegación "Administración"

Fuente: Elaboración propia

3.2.3.3. Diagrama de navegación: Trámites

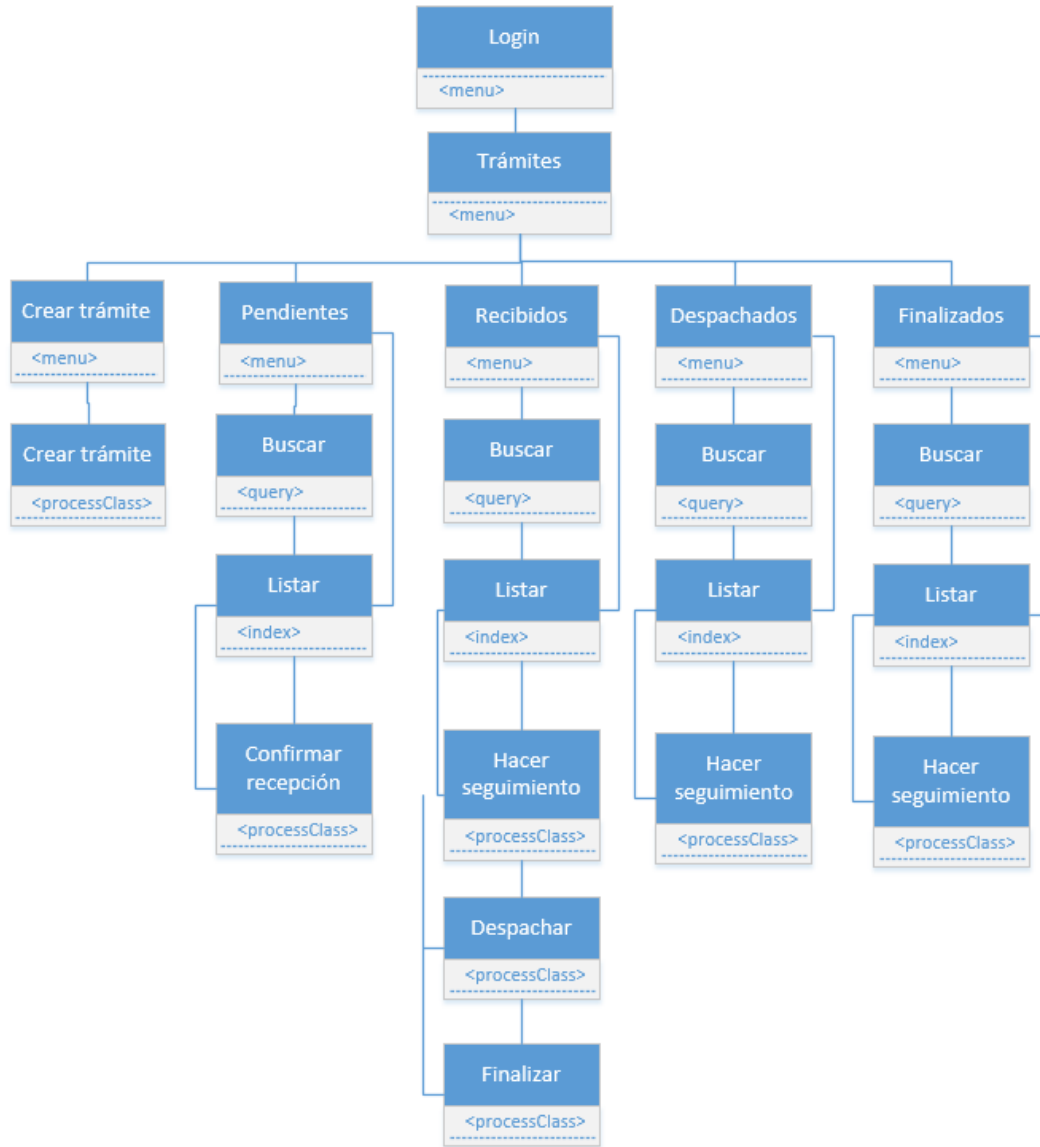


Figura 27 Diagrama de navegación "Trámites"

Fuente: Elaboración propia

3.2.3.4. Diagrama de navegación: Cuenta de usuario

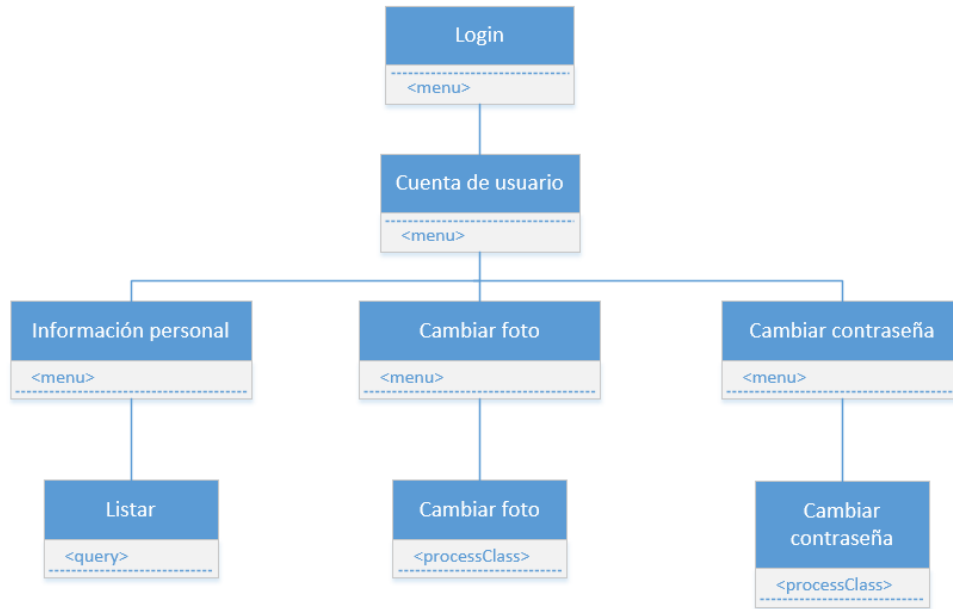


Figura 28 Diagrama de navegación "Cuenta de usuario"

Fuente: Elaboración propia

3.2.4. Modelo de Presentación

Los modelos de presentación según UWE proponen para la construcción de páginas en forma de bosquejos, donde se muestra como los usuarios podrán acceder al sistema mostrando los menús correspondientes según el tipo de usuario.

3.2.4.1. Modelo de presentación: Inicio

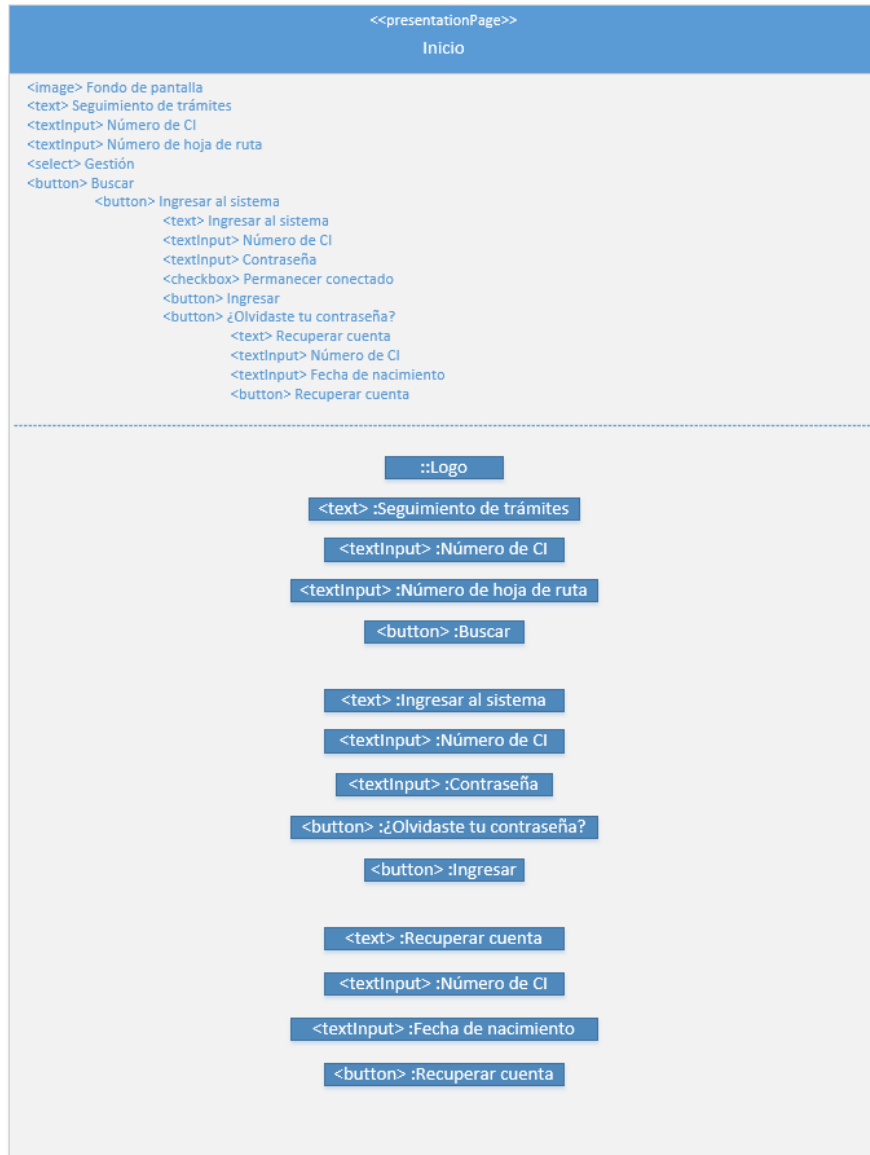


Figura 29 Modelo de presentación "Inicio"

Fuente: Elaboración propia

3.2.4.2. Modelo de presentación: Inicio de usuario

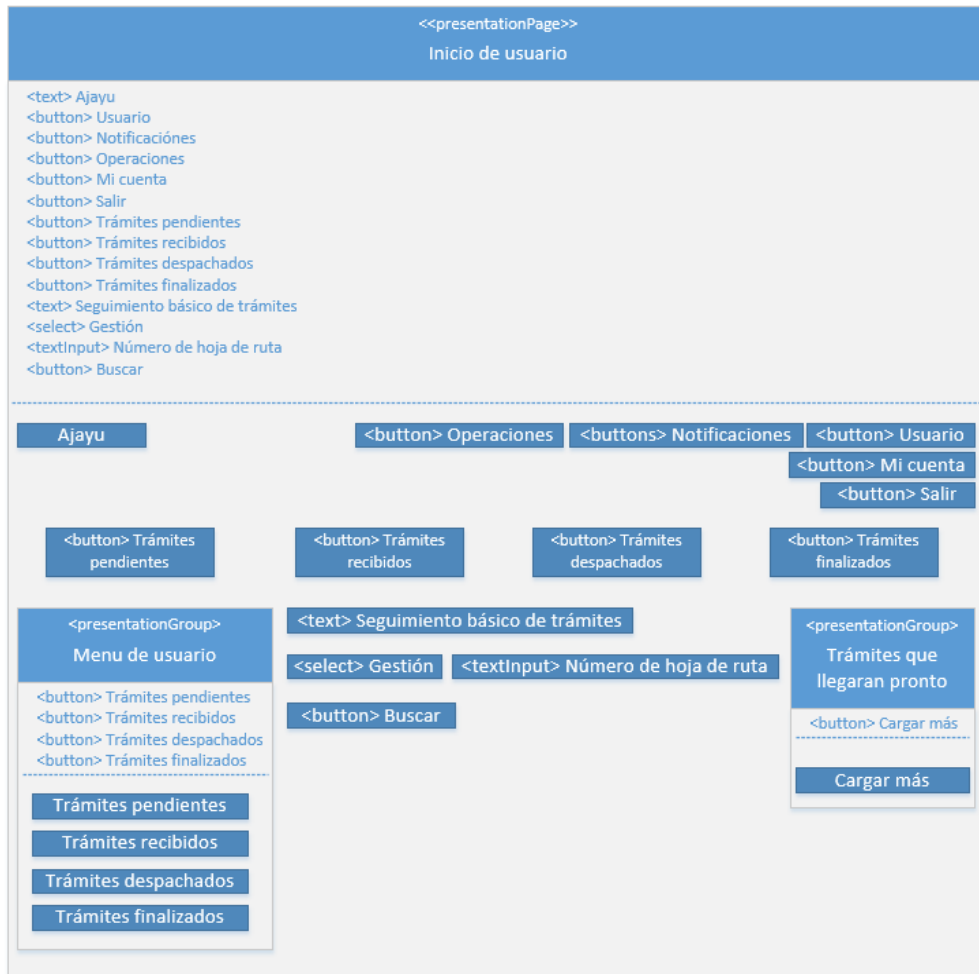


Figura 30 Modelo de presentación "Inicio de usuario"

Fuente: Elaboración propia

3.2.4.3. Modelo de presentación: Trámites pendientes de recepción

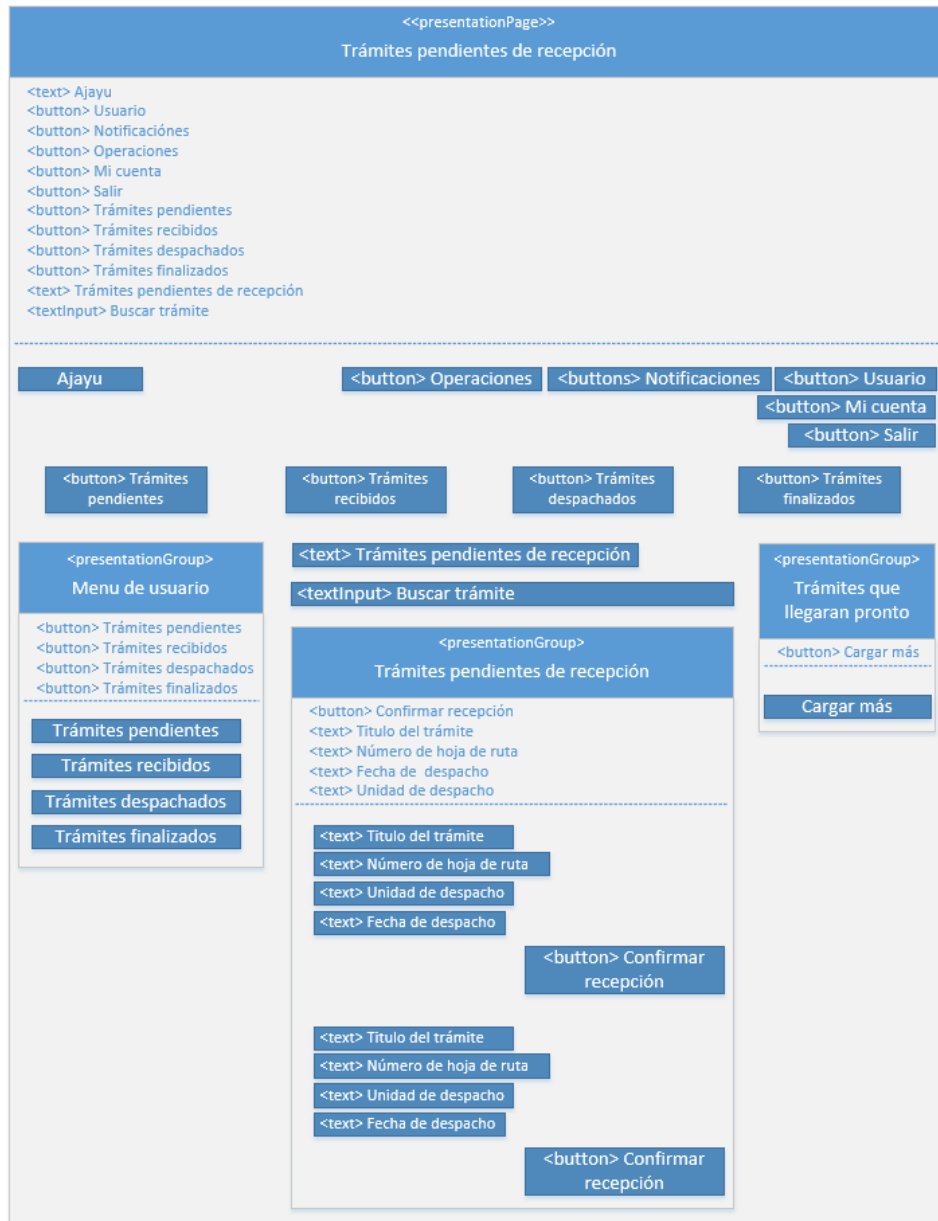


Figura 31 Modelo de presentación "Trámites pendientes de recepción"

Fuente: Elaboración propia

3.2.4.4. Modelo de presentación: Trámites recibidos

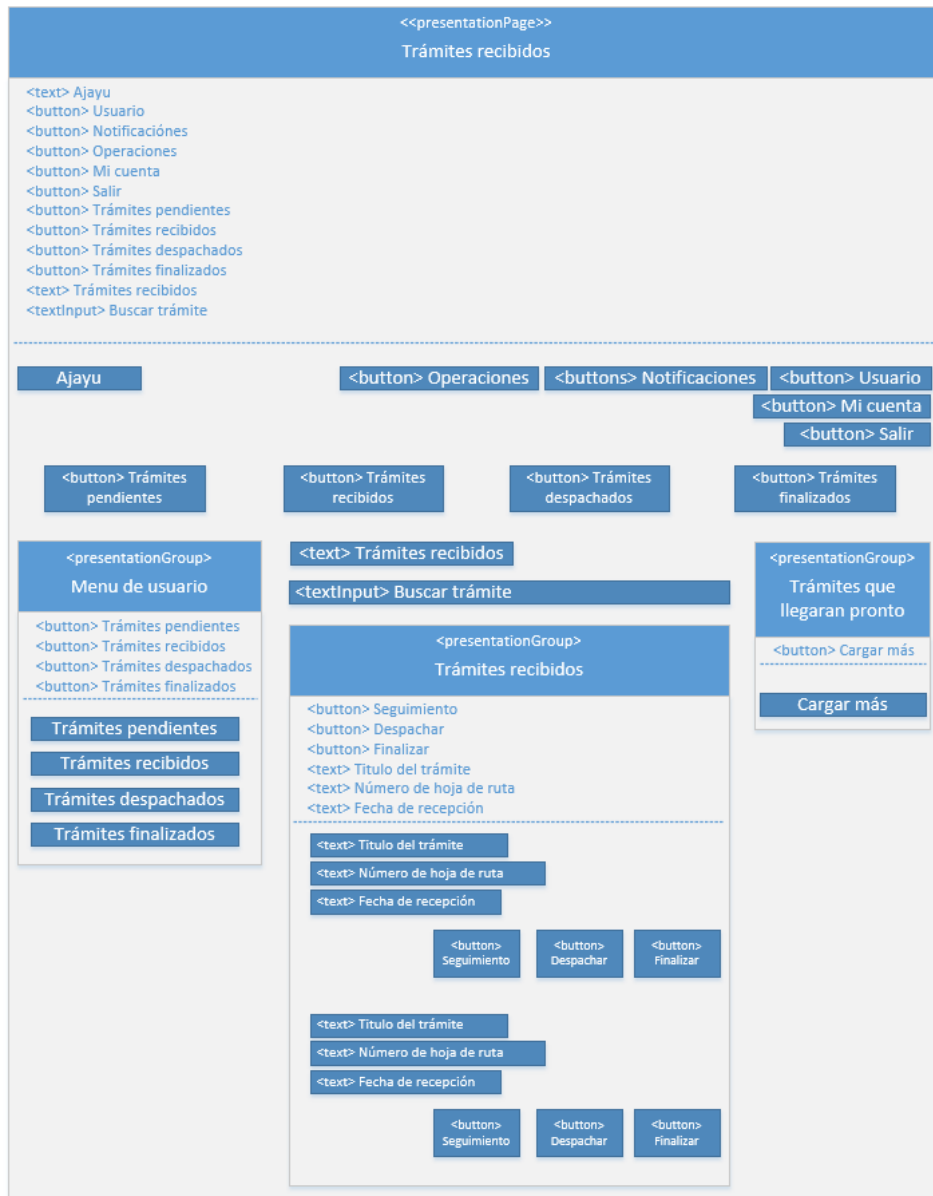


Figura 32 Modelo de presentación "Trámites recibidos"

Fuente: Elaboración propia

3.2.4.5. Modelo de presentación: Trámites despachados

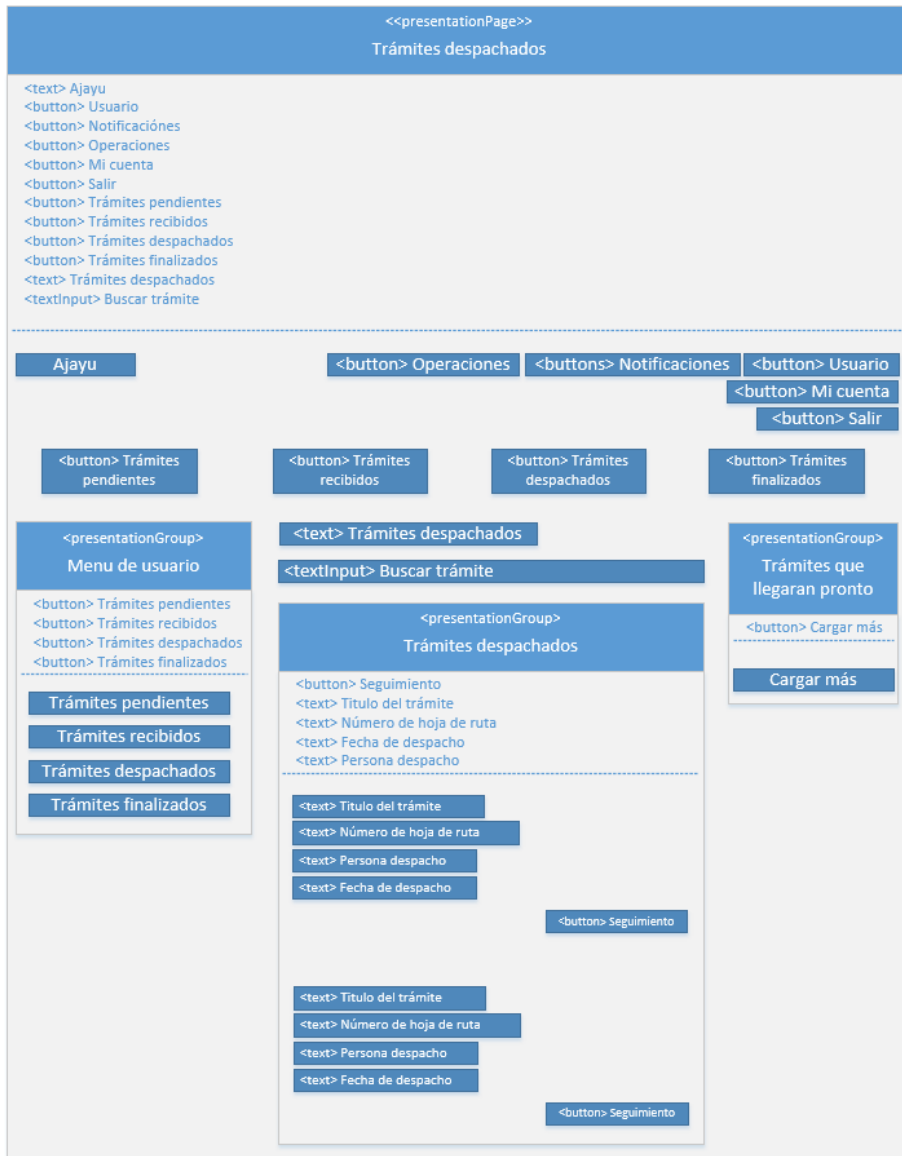


Figura 33 Modelo de presentación "Trámites despachados"

Fuente: Elaboración propia

3.2.4.6. Modelo de presentación: Trámites finalizados

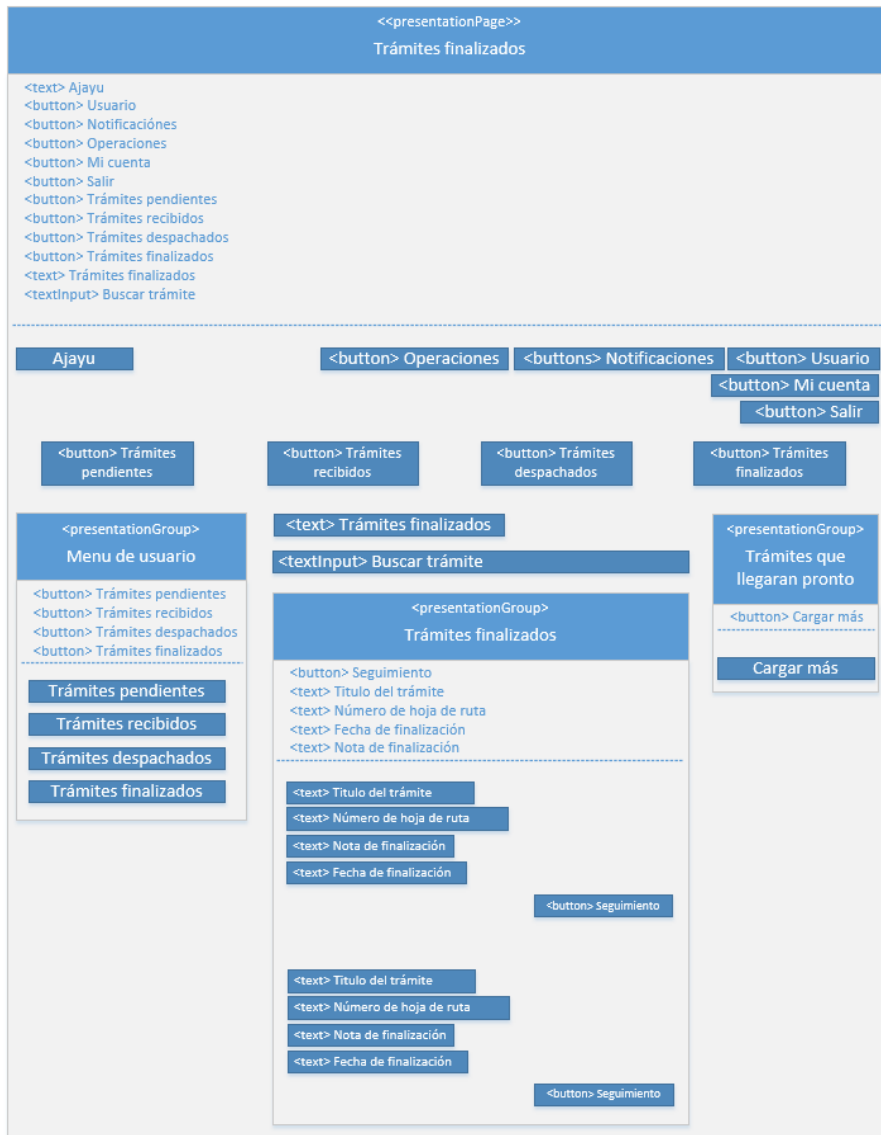


Figura 34 Modelo de presentación "Trámites finalizados"

Fuente: Elaboración propia

3.2.4.7. Modelo de presentación: Cuenta de usuario

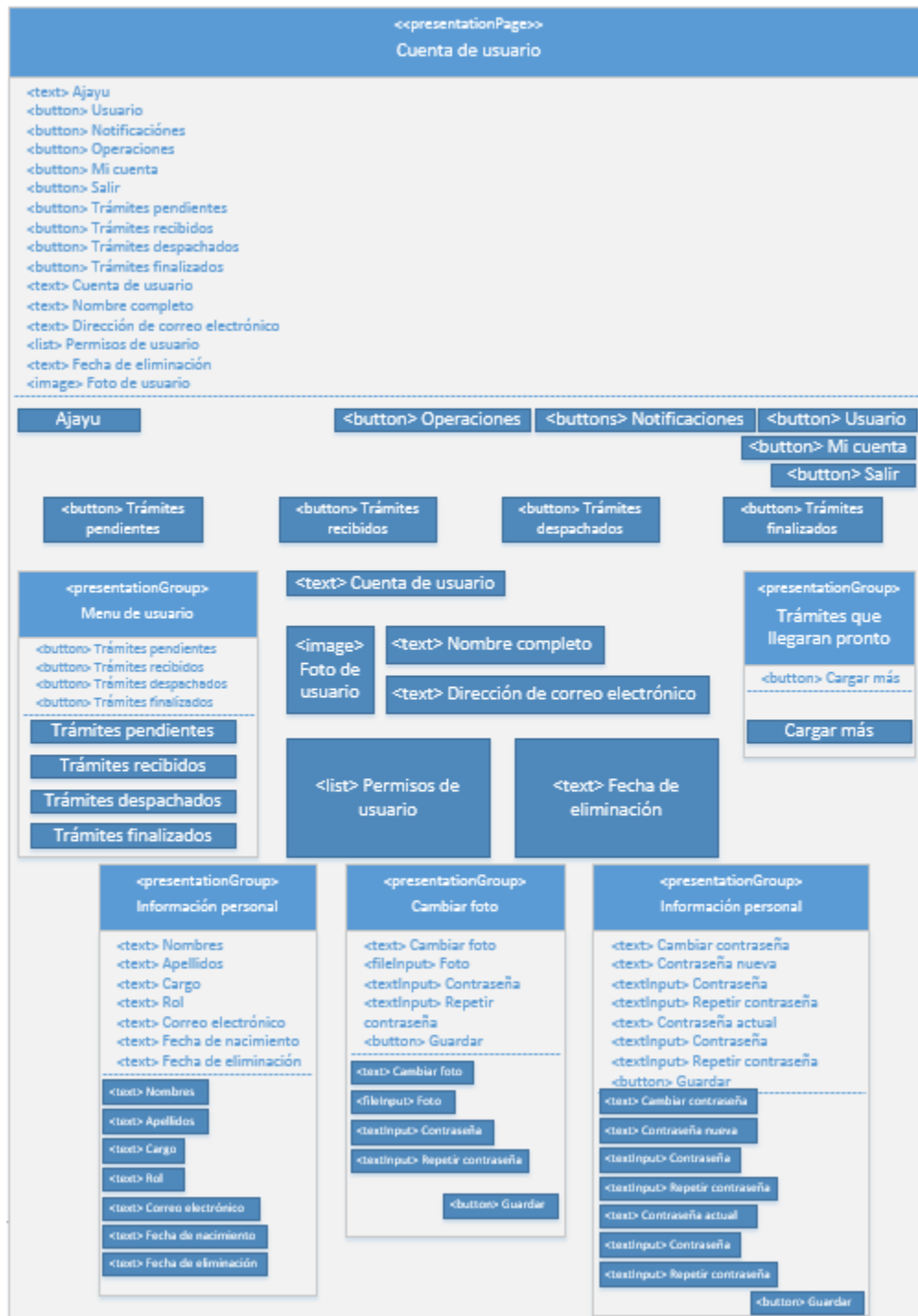


Figura 35 Modelo de presentación "Cuenta de usuario"

Fuente: Elaboración propia

3.2.4.8. Modelo de presentación: Seguimiento detallado de trámites

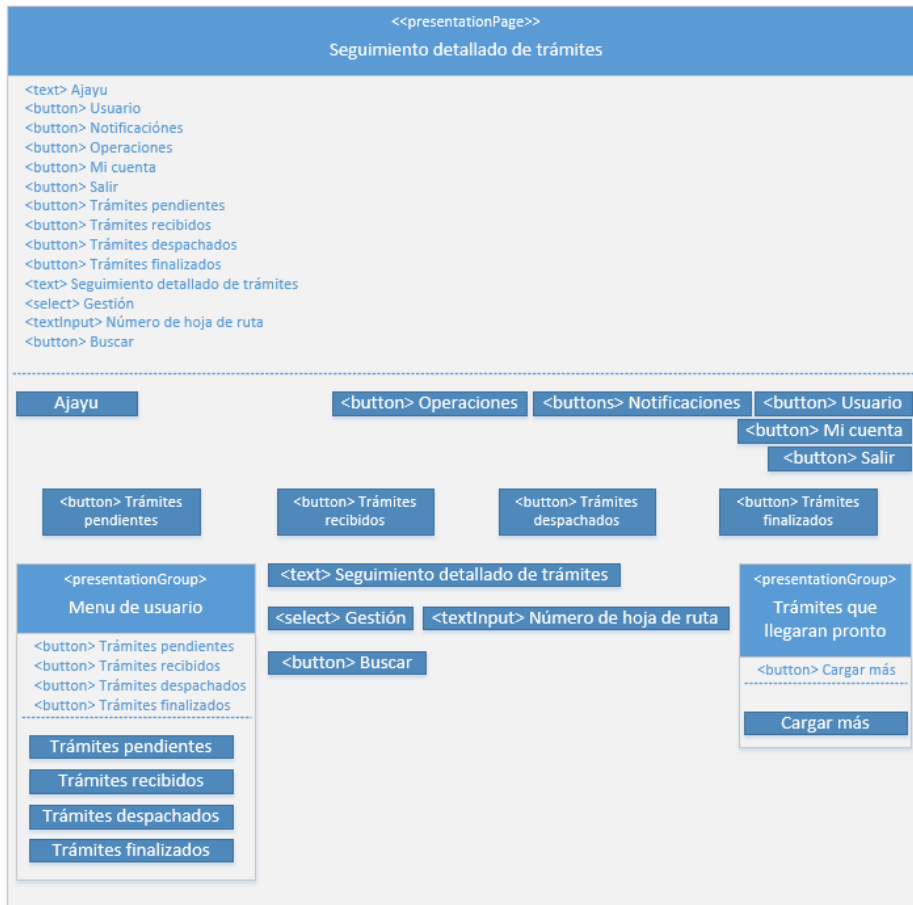


Figura 36 Modelo de presentación "Seguimiento detallado de trámites"

Fuente: Elaboración propia

3.2.5. Modelo de Procesos

Este modelo describe las relaciones entre las diferentes clases del proceso.

3.2.5.1. Diagrama de procesos

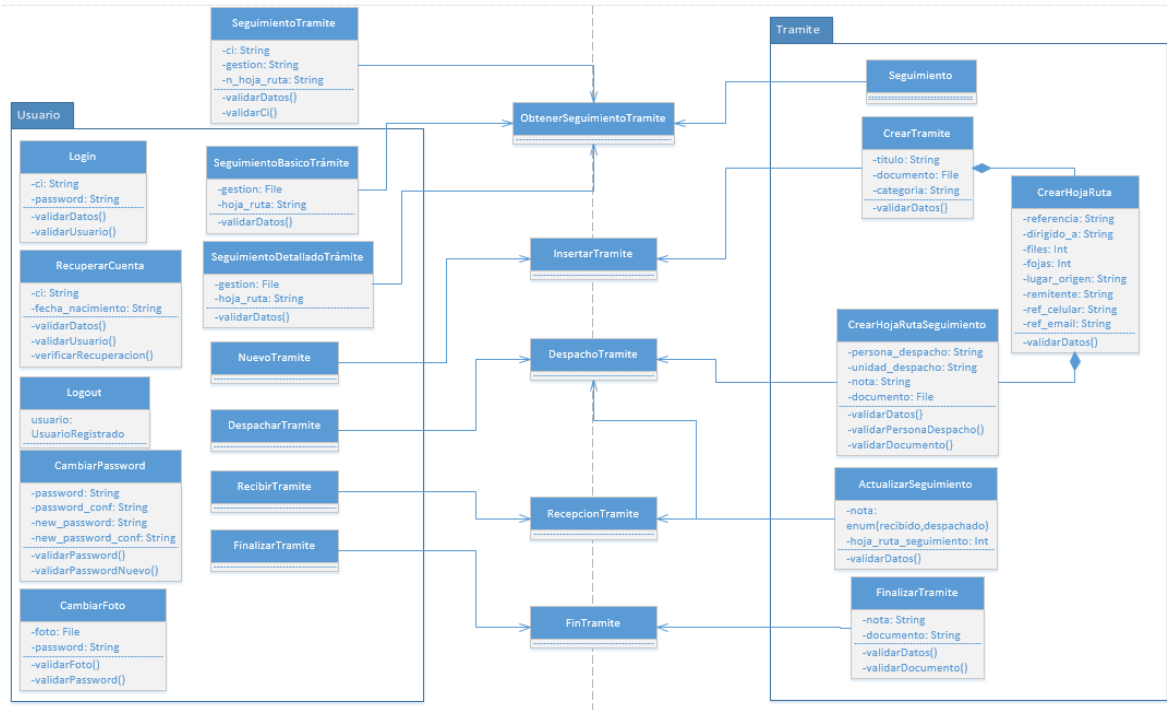


Figura 37 Diagrama de procesos

Fuente: Elaboración propia

3.2.5.2. Diagrama de actividades: Inicio de sesión

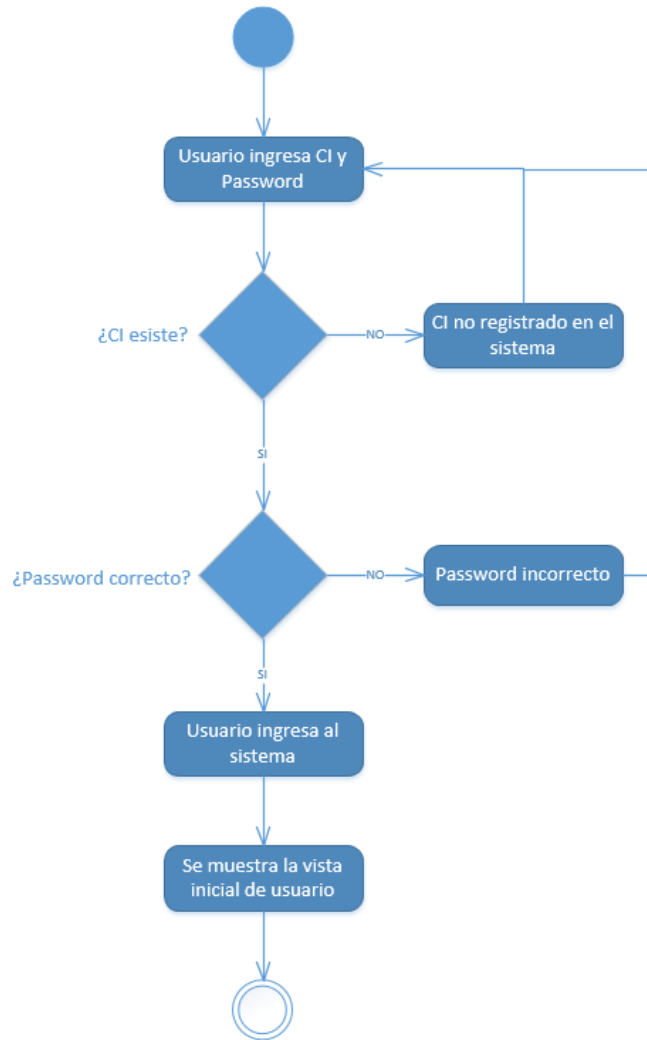


Figura 38 Diagrama de actividades "inicio de sesión"

Fuente: Elaboración propia

3.2.5.3. Diagrama de actividades: Recuperación de cuenta

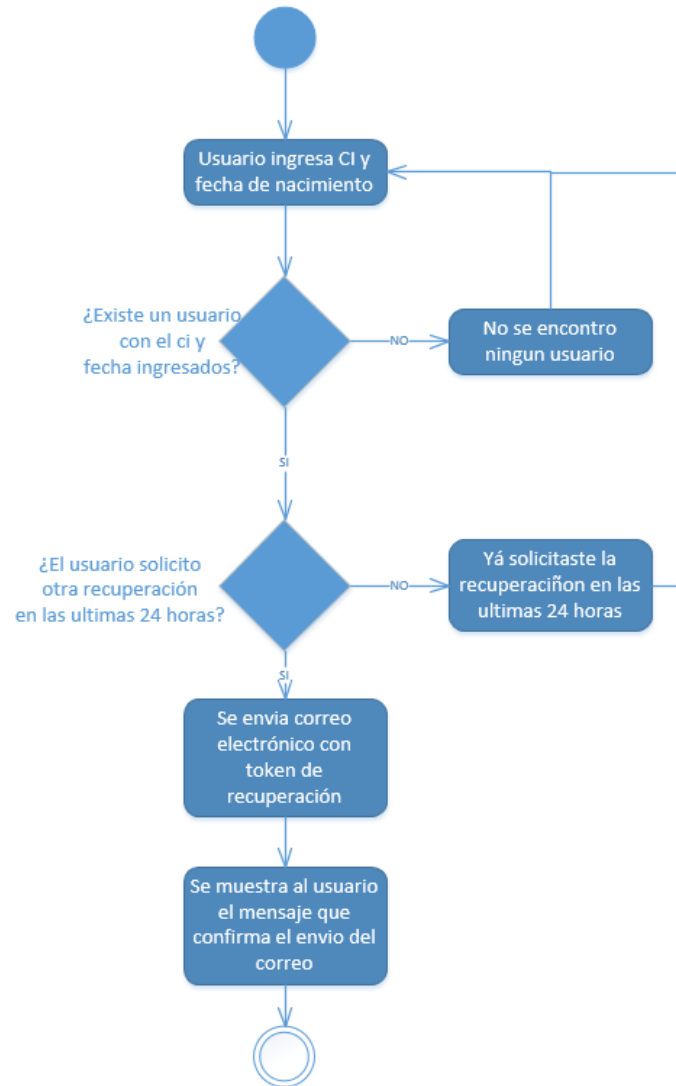


Figura 39 Diagrama de actividades "recuperación de cuenta"

Fuente: Elaboración propia

3.2.5.4. Diagrama de actividades: Seguimiento de trámite

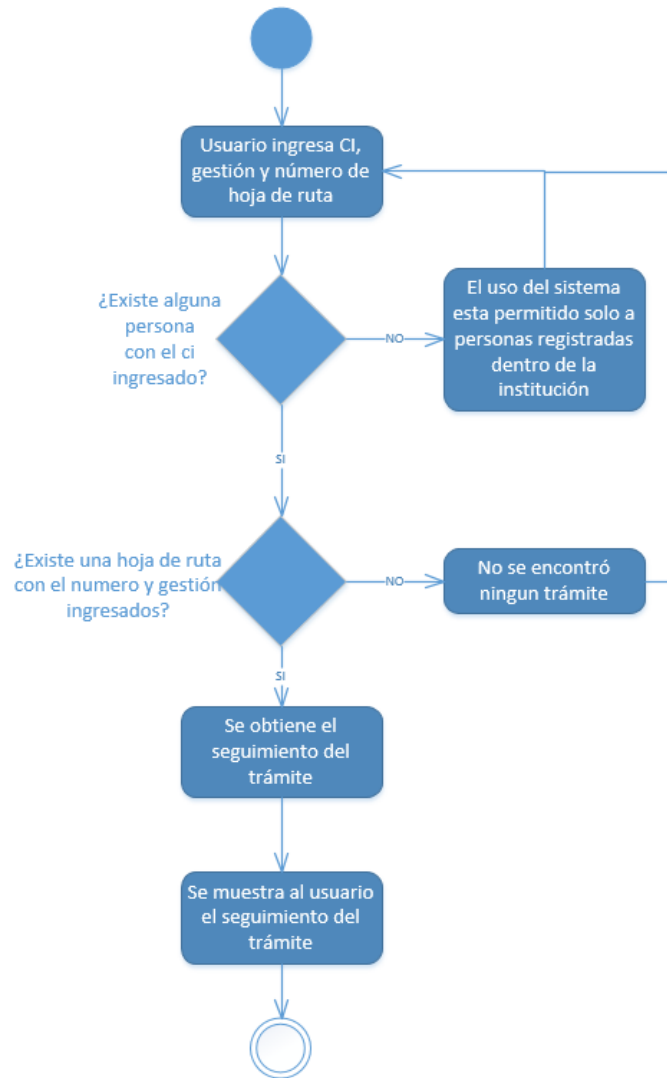


Figura 40 Diagrama de actividades "Seguimiento de trámite"

Fuente: Elaboración propia

3.2.5.5. Diagrama de actividades: Seguimiento básico de trámite

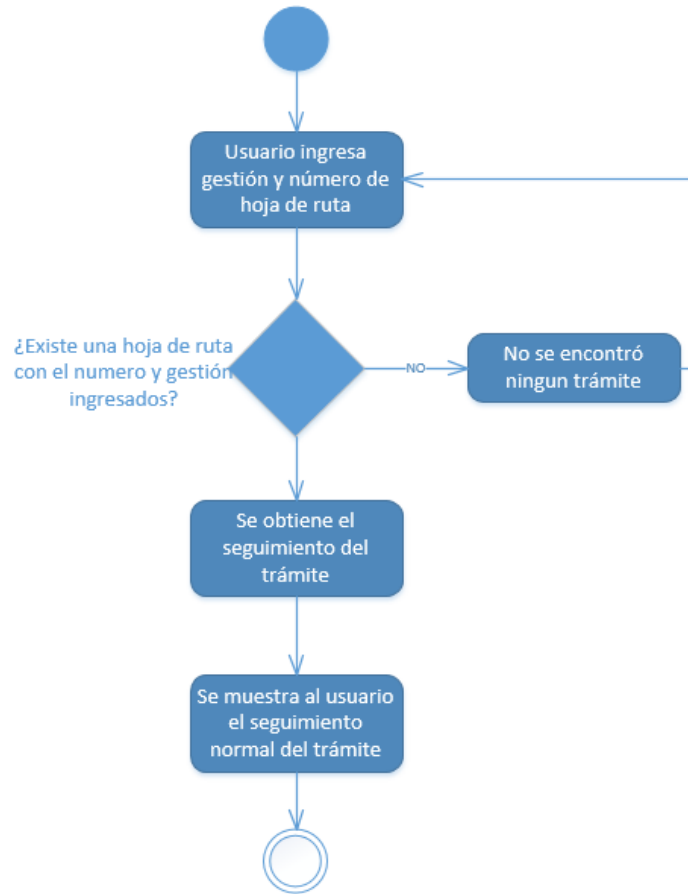


Figura 41 Diagrama de actividades "Seguimiento básico de trámite"

Fuente: Elaboración propia

3.2.5.6. Diagrama de actividades: Seguimiento detallado de trámite

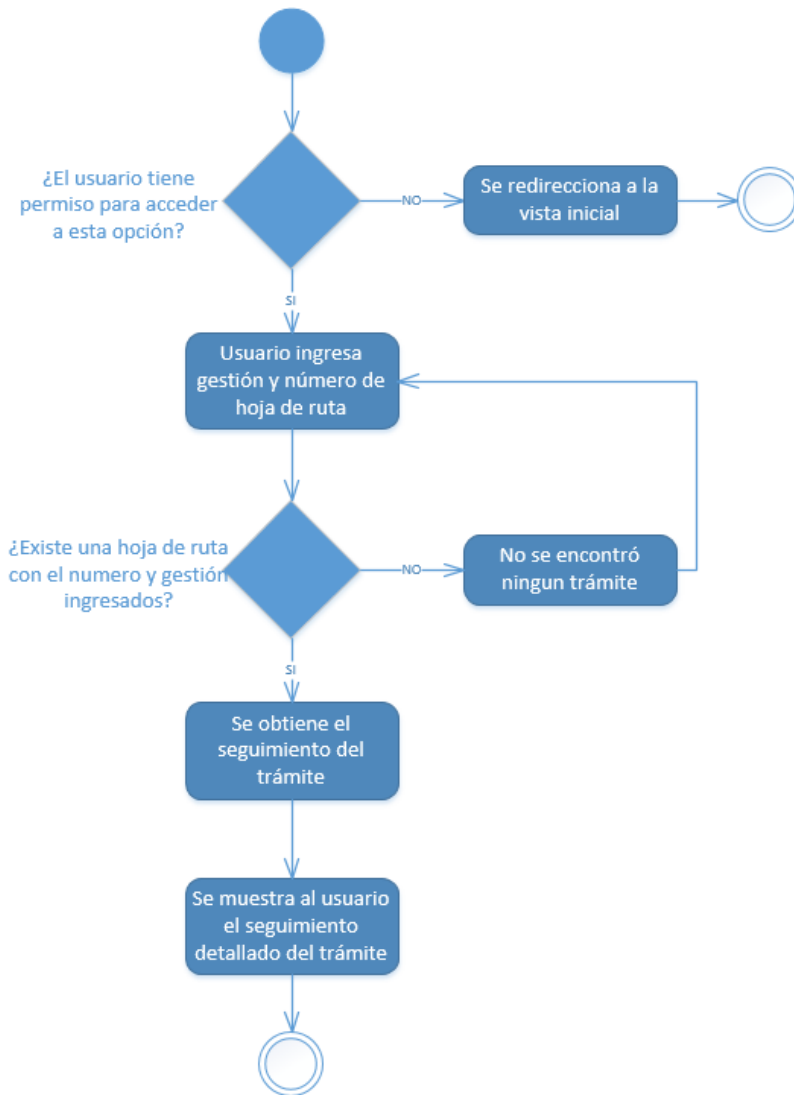


Figura 42 Diagrama de actividades "Seguimiento detallado de trámite"

Fuente: Elaboración propia

3.2.5.7. Diagrama de actividades: Creación de trámites

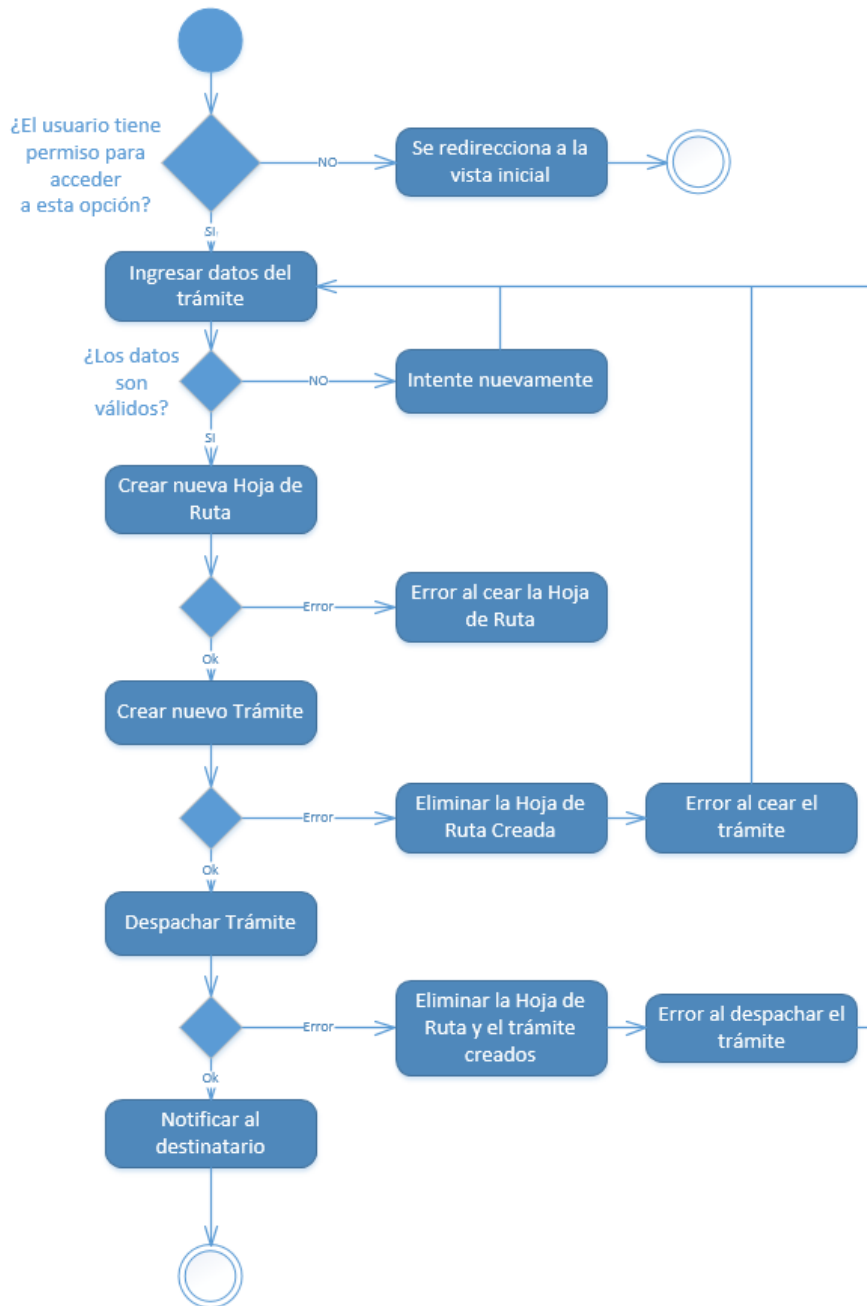


Figura 43 Diagrama de actividades "Creación de trámites"

Fuente: Elaboración propia

3.2.5.8. Diagrama de actividades: Recepción de trámites

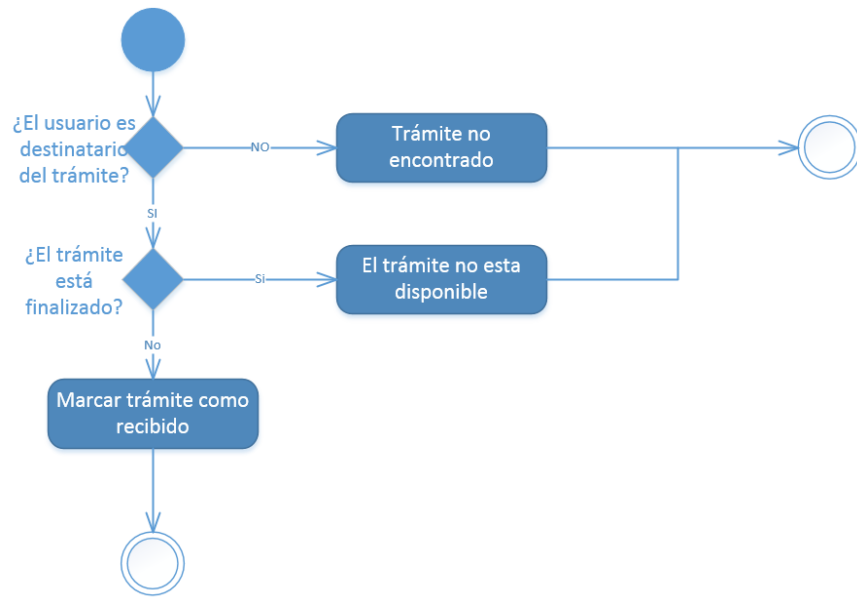


Figura 44 Diagrama de actividades "Recepción de trámites"

Fuente: Elaboración propia

3.2.5.9. Diagrama de actividades: Despacho de trámites

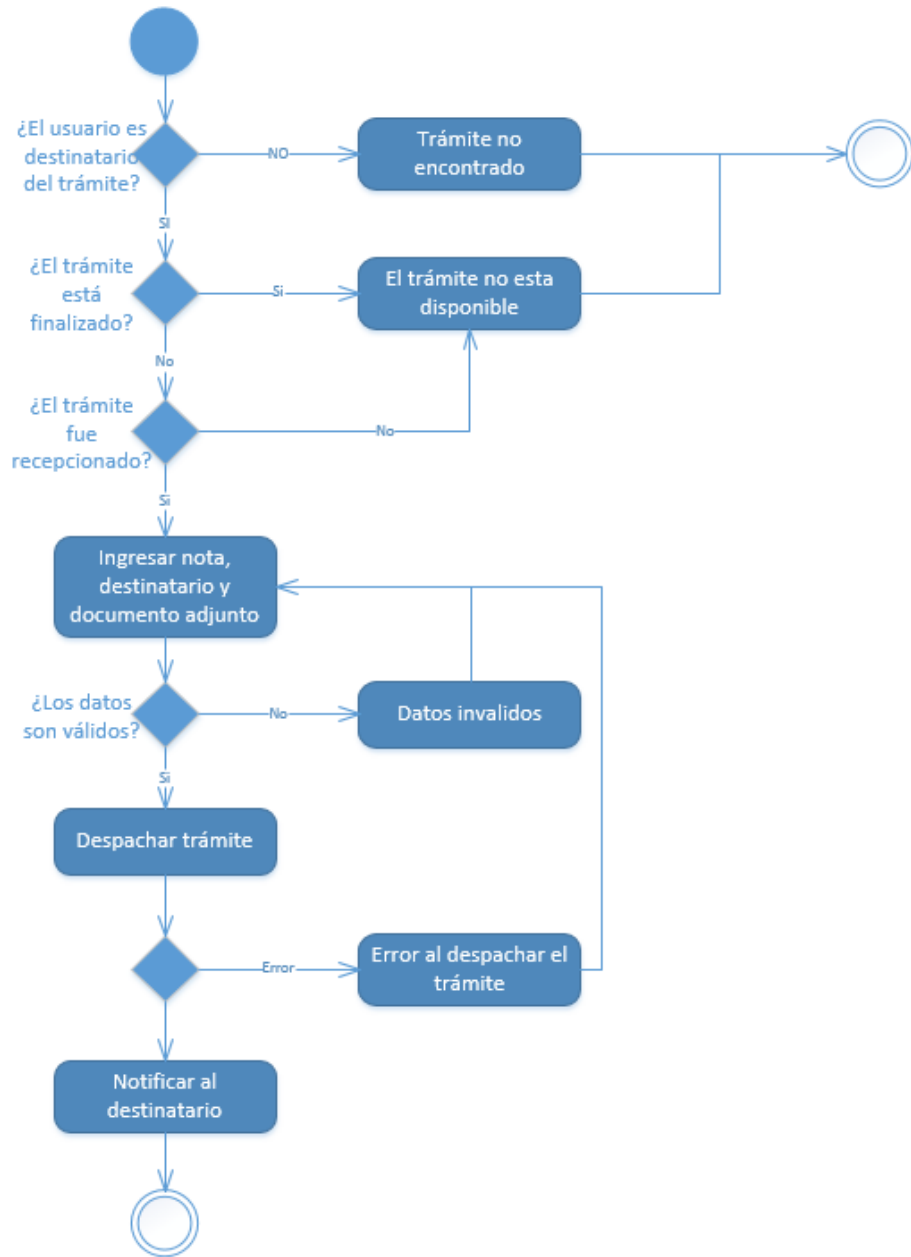


Figura 45 Diagrama de actividades "Despacho de trámites"

Fuente: Elaboración propia

3.2.5.10. Diagrama de actividades: Finalización de trámites

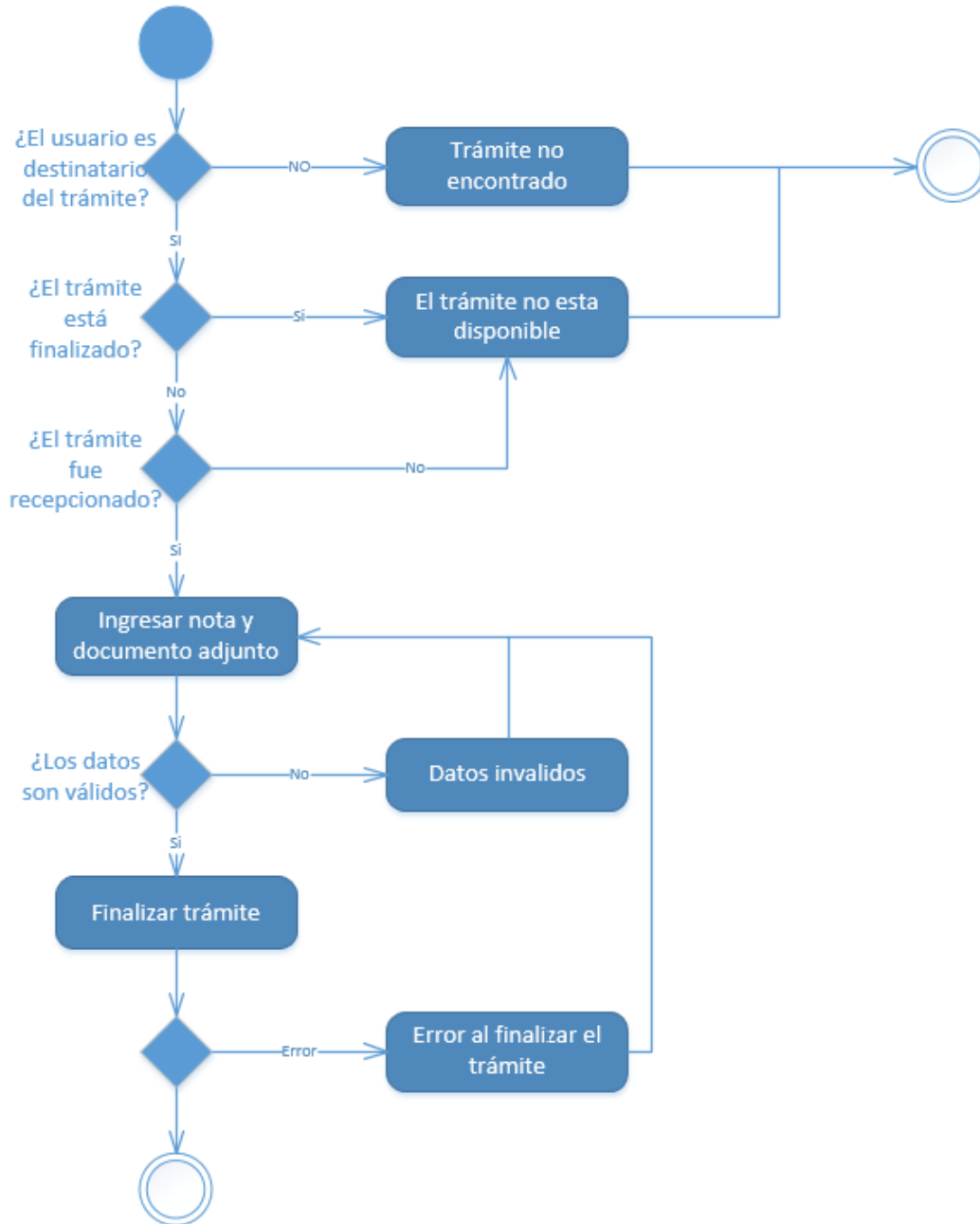


Figura 46 Diagrama de actividades "Finalización de trámites"

Fuente: Elaboración propia

3.3. MÉTRICAS DE CALIDAD

Se hará la medición de calidad del software mediante la métrica de ISO 9126, que establece cualquier componente de la calidad de software puede ser descrito en

términos de una de seis características básicas las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenimiento y portabilidad.

3.3.1. Estándar ISO/IEC 9126

La norma ISO/IEC 9126 define la calidad interna como:

“la totalidad de las características del producto software desde una perspectiva interna. La calidad interna es medida y evaluada en base a los requerimientos de calidad interna. Los detalles de la calidad del producto software pueden ser mejorados durante la implementación, revisión y prueba del código software, pero la naturaleza fundamental de la calidad del producto software representada por la calidad interna permanece sin cambios a menos que sea re diseñado”; y a la calidad externa como: “la totalidad de las características del producto software desde una perspectiva externa. Es la calidad cuando el software es ejecutado, la cual es típicamente medida y evaluada mientras se prueba en un ambiente simulado con datos simulados y usando métricas externas. Durante las pruebas, muchas fallas serán descubiertas y eliminadas. Sin embargo, algunas fallas todavía pueden permanecer después de las pruebas. Como es difícil corregir la arquitectura de software u otros aspectos fundamentales del diseño del software, el diseño fundamental permanece sin cambios a través de las pruebas”.

3.3.1.1. Funcionalidad

Este atributo valora las características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global. La funcionalidad es el grado en que el sistema satisface las necesidades que indican los siguientes sub atributos: estabilidad, exactitud, interoperabilidad, cumplimientos de seguridad.

Punto función. Para el cálculo de punto función se toma en cuenta cinco características, el dominio de información, como son números de entrada, salida, condiciones, archivos e interfaz externa. Luego se realiza el cálculo de punto de función 109 hallando la suma de estas características, parámetros de medición y el factor de ponderación también llamado punto medio de ponderación.

CARACTERÍSTICAS	DESCRIPCIÓN
Número de entradas de Usuario	Se cuenta cada entrada de usuario que proporciona datos al sistema.
Número de salidas de Usuario	Se refiere cada salida que proporciona el sistema al usuario. Entre estos pueden ser informes, reportes y mensajes advertencia, notificaciones y errores.
Número de peticiones de Usuario	Es una entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software.
Número de Archivos	Se toma en cuenta cada archivo, estos pueden ser grupos lógicos de datos (tablas de base de datos).
Número de interfaces externas	Se cuentan todas las interfaces legibles por la máquina.

Tabla 11 Características de la Funcionalidad

Fuente: Elaboración propia

Aplicando lo anterior al proyecto se tiene los siguientes datos:

PARÁMETROS DE MEDICIÓN	CUENTA	FACTOR	TOTAL
Número de entradas de Usuario	21	4	84
Número de salidas de Usuario	14	5	70
Número de peticiones de Usuario	10	5	50
Número de Archivos	17	10	170
Número de interfaces externas	1	7	7
TOTAL			381

Tabla 12 Calculo del punto de función (Factores de Ponderación)

Fuente: Elaboración propia

En la tabla anterior se muestra la cuenta total que se obtiene de la sumatoria de los factores de ponderación a los parámetros de medición.

Para determinar los valores de ajustes de complejidad se indica según se corresponda a las preguntas de la siguiente tabla:

IMPORTANCIA	0%	20%	40%	60%	80%	100%	Fi
FACTOR	0	1	2	3	4	5	
1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?						X	5
2. ¿Se requiere comunicación de datos?				X			3
3. ¿Existen funciones de procesamiento distribuido?			X				2
4. ¿Es crítico el rendimiento?						X	5
5. ¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?					X		4
6. ¿Requiere el sistema entrada de datos interactiva?						X	5
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?			X				2
8. ¿Se actualizan los archivos maestros de forma interactiva?			X				2

9. ¿Son complejos las entradas, las salidas, los archivos o peticiones?			X				2
10. ¿Es complejo el procesamiento interno?					X		4
11. ¿Se ha utilizado el código para ser reutilizable?					X		4
12. ¿Están incluidas en el diseño la conversión y la instalación?				X			3
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?			X				2
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					X		4
TOTAL							49

Tabla 13 Parámetros de medición

Fuente: Elaboración propia

Calculando el punto de función mediante la siguiente ecuación:

Ecuación de funcionalidad:

$$PF = Cuenta\ Total * (0.65 + 0.01 * \sum Fi)$$

Considerado el máximo valor de ajuste de complejidad: $\sum Fi = 70$

Cuenta Total: es la sumatoria del producto del factor de ponderación y valores de los parámetros.

\sum : Es la sumatoria de los valores de ajuste de la complejidad.

Calculando:

$$PF = 381 * (0.65 + 0.01 * 49)$$

$$PF = 381 * 1.14$$

$$PF = 434,34$$

Considerando el máximo ajuste de la complejidad $\sum Fi = 70$ calculamos al 100% el nivel de confianza de la siguiente manera:

$$PFmax = Cuenta\ Total * (0.65 + 0.01 * \sum Fi)$$

$$PFmax = 381 * (0.65 + 0.01 * 70)$$

$$PFmax = 381 * 1.35$$

$$PFmax = 514,35$$

La relación obtenida entre ambos es la funcionalidad:

$$Funcionalidad = PF / PFmax$$

$$Funcionalidad = 434,34 / 514,35$$

$$Funcionalidad = 0,844 * 100 = 84,4 \%$$

Por lo que se concluye que la funcionalidad del sistema es un 84,4%, esto requiere decir que el sistema tiene un 84.4% de funcionar sin riesgos a fallar con operatividad constante y un 15,6% aproximadamente de colapso del sistema.

3.3.1.2. Confiabilidad

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos de un programa o computadora.

Donde se encuentra:

$(T \leq t) = (t)$ Probabilidad de fallas (el termino en el cual el sistema trabaja sin falla)

$(T \leq t) = 1 - (t)$ Probabilidad de trabajo sin fallas (Tiempo en el cual no falla el sistema)

Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el que se ejecuta y se obtiene muestras.

$$F(t) = f * e^{-\mu * t}$$

Donde:

f : Funcionalidad del sistema.

μ : Es la probabilidad de error que puede tener el sistema.

t : Tiempo de duración de gestión en el sistema.

Para lo que consideramos un periodo de 20 días como tiempo de prueba donde se define que cada 10 ejecuciones se presenta 1 falla.

Calculando:

$$F(t) = f * e^{(-\mu/10) * t}$$

$$f(t) = 0.84 * e^{(-1/10) * 20}$$

$$F(t) = 0.1136 * 100 = 11.4\%$$

Reemplazando en las fórmulas de probabilidades:

$$P(T \leq t) = f(t) \rightarrow P(T \leq t) = 0.1136 = 11.4\%$$

$$P(T \leq t) = 1 - f(t) \rightarrow P(T \leq t) = 1 - 0.1136$$

$$P(T \leq t) = 0,864 = 88.6\%$$

Por lo tanto, la confiabilidad del sistema es el 88.6 % en un periodo de 20 días como tiempo de prueba.

3.3.1.3. Usabilidad

Para conocer si el sistema satisface los requerimientos establecidos por el usuario, se realiza una evaluación del mismo en base a encuestas planteadas a los usuarios del sistema, los cuales califican en una ponderación al 100% los usuarios tienen conocimiento de los procesos que realizan y los resultados se refleja en la Tabla 16

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

$$FU = [(\sum (Xi / n)) * 100]$$

Donde:

FU: Facilidad de Uso

Xi: Es la sumatoria de valores

n: Es el número de preguntas

Para responder a las preguntas se debe considerar la siguiente tabla:

ESCALA	VALOR
Muy Bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Tabla 14 Parámetros de medición

Fuente: Elaboración propia

	PARÁMETROS DE MEDICIÓN	SI	NO	EVALUACIÓN
1	¿Puede utilizar con fiabilidad el sistema?	5	0	1
2	¿Puedo controlar operaciones que el sistema solicite?	4	1	0.8
3	¿El sistema permitió la retroalimentación de información?	4	1	0.8

4	¿El sistema cuenta con interfaz amigable a la vista?	5	0	1
5	¿La respuesta del sistema es satisfactoria?	5	0	1
6	¿Le parecen complicadas las funciones del sistema?	3	2	0.6
7	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	1
8	¿Durante el uso del sistema se produjo errores?	2	3	0.4
TOTAL				6.6

Tabla 15 Medidas de seguridad

Fuente: Elaboración propia

Calculando la usabilidad:

$$FU = [(\sum (Xi / n)) * 100]$$

$$FU = [(6.6 / 8) * 100]$$

$$FU = 82.5\%$$

Se concluye que el sistema tiene una usabilidad del 82.5% de acuerdo a los datos obtenidos.

3.3.1.4. Mantenibilidad

El mantenimiento se da las modificaciones del sistema a los nuevos requerimientos según los usuarios administrativos de la Universidad Pública de El Alto.

Por lo que el índice de madurez del software (IMS) se determina con la siguiente ecuación:

$$IMS = [Mt - (Fa + Fc + Fd)] / Mt$$

Donde:

Mt = Numero de módulos de la versión actual

Fc = Número de módulos en la versión actual que se han modificado

Fa = Número de módulos en la versión actual que se han añadido

Fd = Número de módulos de la anterior versión que se han borrado en la versión actual.

Calculando:

$$IMS = [7 - (0 + 1 + 0)] / 7$$

$$IMS = 0.85 * 100 = 85\%$$

Por lo tanto, se puede decir que el sistema tiene un índice de estabilidad de 85% que es la facilidad de mantenimiento, el 15% restante es el margen de error correspondiente a los cambios y modificaciones.

3.3.1.5. Portabilidad

El sistema actual está en plataforma de Linux Debian ejecutable desde cualquier plataforma debido a su diseño adaptable el único requisito es que el dispositivo cuente con internet y un navegador.

El sistema por estar diseñado en un entorno de acceso vía web mide la portabilidad en lado del servidor y lado del cliente, la portabilidad del software se enfoca en tres aspectos:

- Hardware del servidor
- Sistema operativo del servidor
- Software del servidor

Por lo mencionado anteriormente el sistema es portable en sus diferentes entornos tanto en hardware y software.

3.3.2. Seguridad

La seguridad es la capacidad de protección de la información y los datos de manera que las personas no autorizadas puedan leerlas y/o modificarlas. Esta característica se subdivide a su vez en las siguientes sub características:

3.3.2.1. Seguridad a nivel Sistema Operativo

El servidor en el que se alojara el presente proyecto y que es de propiedad de la institución ya tiene las siguientes configuraciones de seguridad:

- Conexión SSH inhabilitada para el usuario ROOT.
- Usuario SSH/FTP con acceso restringido (a escritura y ejecución) a cualquier directorio del sistema operativo exceptuando el directorio donde se encuentra alojado el proyecto y sin permisos para ejecutar comandos como súper usuario (SUDO o SU).
- Conexión remota mediante SSH/FTP solo accesible mediante la red local del servidor que es accesible a usuarios autorizados mediante VPN.

3.3.2.2. Seguridad a nivel Base de Datos

La seguridad en las bases de datos y la aplicación web se encuentra profundamente relacionados. En aplicaciones web toda entrada al sistema debe ser filtrada, y toda salida escapada. Lo mismo aplica cuando las entradas o salidas son de o hacia una base de datos. De esta manera corresponde mitigar los siguientes riesgos:

- Descubrimiento de información acerca de los datos de conexión al servidor (usuario y contraseña), información sensible almacenada en la base de datos o información sobre la estructura de la base de datos.
- Identificación y autenticación: Permite prevenir el ingreso de personas que no son usuarios, para ello el sistema cuenta con un control estricto en el ingreso con un Usuario y una contraseña estrictamente controlada.
- Encriptación: Se aplica la encriptación de seguridad para la contraseña, un dato de suma importancia para el ingreso al sistema. de este modo se está utilizando lo que es el algoritmo de SHA-256 es un hash de 64 dígitos

hexadecimales, SHA es una de las muchas funciones hash, una encriptación de alta seguridad.

Por lo tanto, para eliminar estas vulnerabilidades de seguridad, se tiene un gestor de base de datos ya implementado que es MariaDB, cuyo nivel de seguridad permite establecer usuarios y permisos, con privilegios asignados a los objetos de la base de datos como son: tablas, consultas, ejecución de triggers, etc.

3.3.2.3. Seguridad a nivel del Software

Para preservar la seguridad a nivel de software el sistema tiene las siguientes características:

- El sistema exige contraseñas que tengan como mínimo 8 caracteres, con al menos una letra mayúscula, una letra minúscula y un número.
- El módulo de inicio y cierre de sesión permite el acceso solamente a personal administrativo de la institución.
- Los usuarios administrativos tienen una contraseña por defecto para acceder a al sistema basado en datos personales del mismo.
- La primera vez que el usuario administrativo acceda al sistema, este estará obligado a cambiar su contraseña.
- El usuario administrativo podrá recuperar el acceso a su cuenta mediante el uso del módulo de recuperación que enviará un enlace por correo electrónico que le permitirá cambiar su contraseña.

3.3.3. Costos

Existen distintos métodos para la estimación de costes de desarrollo de software, estos métodos no son otra cosa que establecer una relación matemática entre el esfuerzo y el tiempo de desarrollo.

3.3.3.1. Método de estimación COCOMO II

Para determinar el costo de la aplicación se usa el modelo COCOMO II orientado en los puntos de función, para calcular el Esfuerzo, necesitaremos hallar la variable KDLC (Kilolíneas de código), donde los PF son 434,34 y las líneas por cada PF equivalen a 53 según vemos en la tabla que se ilustra a continuación:

LENGUAJE DE PROGRAMACIÓN	LOC/PF
Lenguaje Ensamblador	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
php	67
Java	31
Lenguajes orientados a objetos	30
Lenguajes de cuarta generación (4GLs)	20
Generadores de código	15
Hojas de calculo	6
Lenguajes gráficos	4

Tabla 16 Conversión de puntos de función

Fuente: (Taina, 2005)

Así pues, tras saber que son 67 LDC por cada PF, el resultado de los KDLC será el siguiente:

$$KLDC = (PF * \text{Lineas de codigo por cada PF}) / 1000$$

$$KLDC = (434,34 * 67) / 1000$$

$$KLDC = 29,1$$

Así pues, en nuestro caso el tipo semi-acoplado será el más apropiado ya que el proyecto tiene una complejidad media, por consiguiente, los coeficientes que usaremos serán las siguientes:

PROYECTO DE SOFTWARE	a	e	c	d
Orgánico	3,2	1,05	2,5	0,38
Semi-acoplado	3,0	1,12	2,5	0,35
Empotrado	2,8	1,20	2,5	0,32

Tabla 17 Modelo COCOMO

Fuente: (Pressman, 2010)

De igual manera se debe de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente tabla:

CONDUCTORES DE COSTE	Valor					
	Muy Alto	Bajo	Nominal	Alto	Muy Alto	Extr. Alto
Atributos de Software						
Fiabilidad requerida del software	0,75	0,88	1,00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de Hardware						
Restricciones del tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1,00	1,07	1,15	-
Atributos de personal						
Capacidad del analista	1,46	1,19	1,00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	-

Experiencia en S. O. utilizado	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1,00	0,95	-	-
Atributos del proyecto						
Prácticas de programación moderna	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,24	1,08	1,00	1,04	1,10	-

Tabla 18 Conductores de costo

Fuente: (Castillo, 2017)

Justificación:

- Fiabilidad requerida del software: Si se produce un fallo por el registro o envío de trámites puede ocasionar para el seguimiento del mismo (Valoración Muy alta).
- Tamaño de la base de datos: La base de datos será estándar (Valoración Nominal).
- Complejidad del producto: La aplicación no va a realizar cálculos complejos (Valoración Media).
- Restricciones del tiempo de ejecución: Para que exista el menor retraso posible en el procesamiento de trámites se exige alto rendimiento (Valoración Alta).
- Restricciones del almacenamiento principal: Los documentos que se almacenarán no son pesados (Valoración Nominal).
- Volatilidad de la máquina virtual: Se usará cualquier dispositivo con navegador y conexión a internet (Valoración Nominal).
- Tiempo de respuesta del ordenador: Deberá ser rápida para facilitar el flujo de trabajo del usuario (Valoración Alta).
- Capacidad del analista: Capacidad muy alta, debido a la complejidad del manejo de trámites en la institución (Valoración Muy alta)

- Experiencia en la aplicación: Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).
- Capacidad de los programadores: Teóricamente deberá tenerse una capacidad alta por la experiencia en anteriores proyectos de similar complejidad (Valoración Alta).
- Experiencia en S.O. utilizado: Con Linux la experiencia es alta (Valoración Alta).
- Experiencia en el lenguaje de programación: Es alta, dado se trabajó en proyectos de similar complejidad en el mismo lenguaje (Valoración Alta).
- Prácticas de programación modernas: Se usarán prácticas de programación modernas (Valoración Alta).
- Utilización de herramientas software: Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Nominal).
- Limitaciones de planificación del proyecto: Existen pocos límites de planificación. (Valoración Baja).

$$\begin{aligned}
 FAE &= 1,40 * 1,00 * 1,00 * 1,11 * 1,00 * 1,00 * 1,07 * 0,71 * 0,82 * 0,86 * 0,90 \\
 &\quad * 0,95 * 0,91 * 1,00 * 1,08 \\
 FAE &= 0,699578
 \end{aligned}$$

Cálculo del Esfuerzo del Desarrollo:

$$\begin{aligned}
 E &= a KLD C^e \times FAE \\
 E &= 3,0 \times 29,1^{1.12} \times 0.699578 \\
 E &= 91,52 \Rightarrow \underline{\underline{92 \text{ personas/mes}}}
 \end{aligned}$$

Cálculo de Tiempo de Desarrollo:

$$\begin{aligned}
 T &= c \text{ Esfuerzo}^d \\
 T &= 2,5 \times 91,52^{0.35} \\
 T &= 12,14 \text{ meses} \Rightarrow \underline{\underline{1 \text{ año}}}
 \end{aligned}$$

Personal promedio:

$$P = E / T$$

$$P = 91,52 / 12,14$$

$$P = 7,38 \Rightarrow \underline{\mathbf{7 \text{ personas}}}$$

3.3.3.2. Costos del proyecto

3.3.3.2.1. Costo de Personal

Según los resultados obtenidos será necesario un equipo de 7 personas trabajando alrededor de 1 año.

Ahora pues sabemos que el salario promedio de un programador junior es de aproximadamente Bs. 3,200 se tiene lo siguiente:

*Costo de la aplicación = Numero de programadores * Salario de un programador*

$$\text{Costo de la aplicación} = 7 * 3200$$

$$\text{Costo de la aplicación} = 22\ 400 \text{ Bs}$$

3.3.3.2.2. Costo de elaboración del proyecto

Por otra parte, el desarrollo como tal del proyecto comprende la siguiente serie de gastos que se detallan a continuación:

DETALLE	IMPORTE (Bs)
Análisis y diseño de la aplicación	500
Material de escritorio	100
Internet	360
TOTAL	960

Tabla 19 Costo de elaboración del proyecto

Fuente: Elaboración propia

3.3.3.2.3. Costo de implementación del proyecto

Debido a que el proyecto está desarrollado con software libre y que será implementada en un servidor que utiliza software libre el costo de implementación es de 0 (cero) bolivianos.

3.3.3.2.4. Costo total del proyecto

El costo total del proyecto es la sumatoria del costo del personal, costo de elaboración del proyecto y costo de implementación del proyecto.

DESCRIPCIÓN	COSTO TOTAL (Bs)
Costo del personal	22 400
Costo de elaboración del proyecto	960
Costo de implementación del proyecto	0
TOTAL	23 360

Tabla 20 Costo total del proyecto

Fuente: Elaboración propia

Por tanto, el costo total del proyecto es de 23,360 bolivianos.

3.3.4. Pruebas de Software

Para las pruebas de software se utilizará las pruebas de caja blanca y las pruebas de caja negra, las cuales probaran que se obtiene una determinada salida para una entrada determinada y además de probar la funcionalidad en conjunto del sistema.

3.3.4.1. Pruebas de Caja Blanca

Esta prueba se basa en el cálculo de las regiones que deben ser consideradas como caminos independientes del sistema, y estableciendo cuáles con las entradas para que se ejecutan cada una de las regiones. A continuación de forma general se muestra los caminos independientes del sistema:

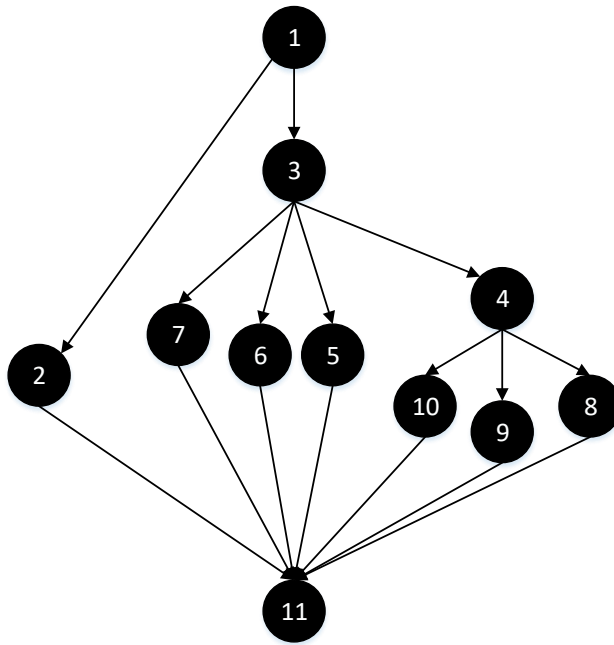


Figura 47 Caja Blanca

Fuente: Elaboración propia

Donde:

1. Inicio del Sistema
2. Seguimiento de trámites
3. Página principal de usuario
4. Administración de Trámites
5. Cuenta de usuario
6. Administrar usuarios
7. Administrar unidades
8. Creación de Trámites
9. Correspondencia de trámites
10. Reporte de Trámites
11. Fin del sistema

Ya realizado el grafo generado de acuerdo de las características del sistema, se procede a determinar la complejidad ciclo matica del grafo de acuerdo a la siguiente formula:

$$V(G) = A - N + 2$$

Donde:

$$A = 16 \text{ (Aristas)}$$

$$N = 11 \text{ (Nodos)}$$

Por tanto:

$$V(G) = 16 - 11 + 2 = 7$$

Camino 1: 1-2-11

Camino 2: 1-3-7-11

Camino 3: 1-3-6-11

Camino 4: 1-3-5-11

Camino 5: 1-3-4-10-11

Camino 6: 1-3-4-9-11

Camino 7: 1-3-4-8-11

Estos caminos representan condiciones en las que al menos se ejecuta los nodos establecidos en el camino, la especificación de estos caminos son los siguientes:

- **Camino 1:** El sistema recibe los datos requeridos para consultar la información de un trámite y muestra el seguimiento y estado de este.
- **Camino 2:** Se muestra un listado de unidades existentes en el sistema y la opción de crear nuevas.
- **Camino 3:** Se muestra un listado de usuarios existentes en el sistema y la opción de crear nuevos usuarios.
- **Camino 4:** Se muestra los permisos del usuario, la opción de cambiar su foto de perfil y cambio de contraseña.
- **Camino 5:** Se ejecutan los reportes sobre los trámites de la unidad de acuerdo a la solicitud.
- **Camino 6:** Se muestra un listado de todos los trámites que ingresaron o salieron de la unidad.
- **Camino 7:** Se muestra un formulario para registrar un nuevo trámite.

3.3.4.2. Pruebas de Caja Negra

La prueba de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

Para realizarla prueba de caja negra se tomará en cuenta los casos de ingreso al sistema, registro de trámites y seguimiento de trámites.

Caso de prueba: Ingreso al sistema	
Descripción	Acceso al sistema mediante email y contraseña. Se permite el acceso al sistema una vez verificado los datos de acceso.
Condición de ejecución	El formulario debe ser enviado por el método POST, con todos los campos presentes y el token CSRF.
Entradas	<ul style="list-style-type: none">• Usuario (número de CI).• Contraseña.• Opcionalmente marcar el checkbox para mantener la sesión activa
Resultados esperados	El Sistema verifica el token CSRF, verifica que los campos "Usuario" y "Contraseña" estén presentes, verifica en la base de datos los datos ingresados para permitir al usuario acceder al sistema. En caso de que el checkbox "Permanecer conectado" este marcado, se genera un token que permitirá al usuario acceder al sistema saltando el inicio de sesión
Resultados obtenidos	Se realiza el ingreso al sistema de forma satisfactoria

Tabla 21 Caso de prueba "Ingreso al sistema"

Fuente: Elaboración propia

Caso de prueba: Registro de trámites	
Descripción	Registro de un nuevo trámite en el sistema.

Condición de ejecución	El formulario debe ser enviado por el método POST, con todos los campos presentes y el token CSRF, y debe ser realizado solamente por usuarios que tengan los permisos necesarios.
Entradas	<ul style="list-style-type: none"> • Referencia del trámite. • Documento (input File). • Categoría (opcional). • Dirigido a. • N. de Hojas. • N. de Files. • Lugar de origen. • Referencia. • Remitente. • Correo electrónico del remitente (opcional). • Teléfono del remitente (opcional). • Unidad donde despachar el trámite. • Usuario a quien despachar el trámite. • Anotación (opcional). • Marcar como alta prioridad (opcional).
Resultados esperados	El Sistema verifica el token CSRF, que el usuario tenga los permisos requeridos, que todos los campos no opcionales sean válidos, que el usuario haya subido un documento en formato válido y de ser válidos todos los datos se registra en la base de datos además se envía una notificación al usuario al que se despachó el trámite.
Resultados obtenidos	Se realiza el registro de trámites con resultados satisfactorios.

Tabla 22 Caso de prueba “Registro de trámites”

Fuente: Elaboración propia

Caso de prueba: Seguimiento de trámites

Descripción	Consulta sobre el seguimiento y estado de un trámite..
Condición de ejecución	El proceso debe ser realizado por una persona registrada en la institución, el formulario debe ser enviado por el método POST, con todos los campos presentes y el token CSRF..

Entradas	<ul style="list-style-type: none"> • Número de CI de la persona que consulta. • Gestión del trámite. • Código único del trámite
Resultados esperados	El sistema verifica que el número de CI pertenezca a una persona registrada en la institución, que el token CSRF sea válido, que la gestión y el código único de trámite correspondan a un trámite existente y de pasar la verificación se muestre el estado y detalles del trámite.
Resultados obtenidos	Se realiza el seguimiento de trámites con resultados satisfactorios.

Tabla 23 Caso de prueba “Seguimiento de trámites”

Fuente: Elaboración propia

3.4. IMPLEMENTACIÓN DEL SISTEMA

3.4.1. Página de inicio



Figura 48 Página inicial del sistema

Fuente: Elaboración propia

3.4.2. Ingreso al sistema

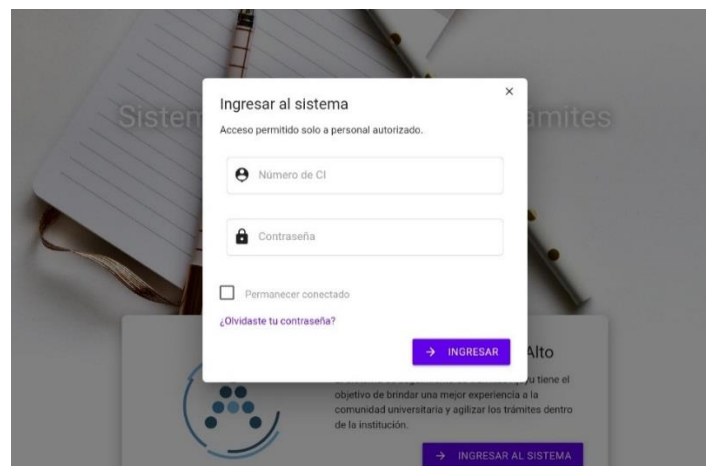


Figura 49 Ingreso al sistema

Fuente: Elaboración propia

3.4.3. Recuperación de cuenta

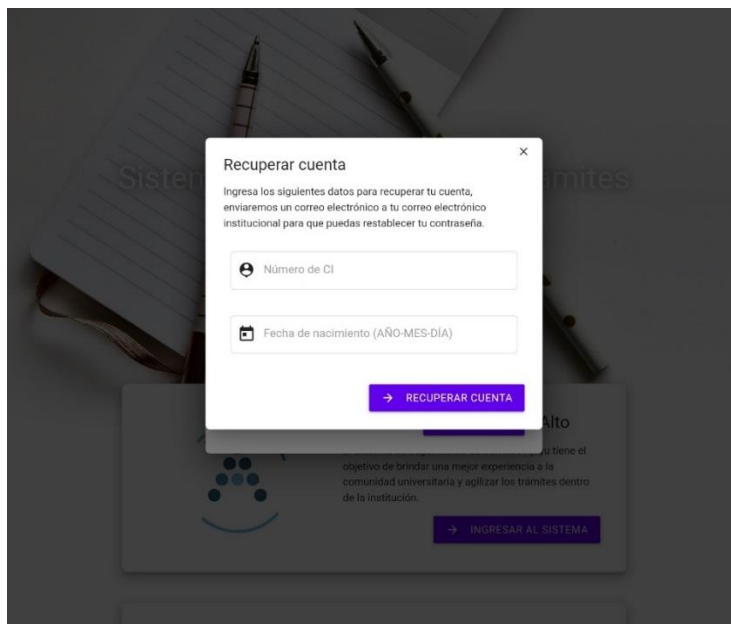


Figura 50 Recuperación de cuenta

Fuente: Elaboración propia

3.4.4. Página inicial al iniciar sesión

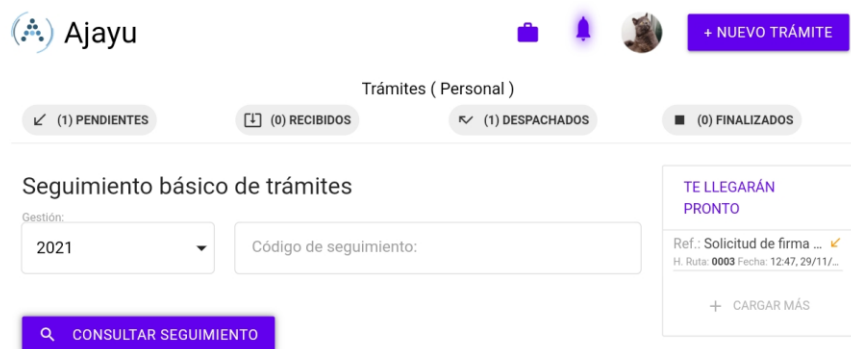


Figura 51 Página inicial después de iniciar sesión

Fuente: Elaboración propia

3.4.5. Página de creación de trámites

Ajayu

Trámites (Personal)

(1) PENDIENTES (0) RECIBIDOS (1) DESPACHADOS (0) FINALIZADOS

Trámites / Nuevo trámite

Nuevo trámite

- 1 Información del trámite**
Ingrese los datos del nuevo trámite
- 2 Hoja de ruta
Ingrese los datos para la hoja de ruta
- 3 Datos del remitente
Ingrese los datos del remitente
- 4 Despacho
¿A donde se despachará este trámite?

Referencia

Selecciona el documento del trámite.

Opcionalmente selecciona una categoría para este trámite.

Categoría:

Sin categoría

CONTINUAR

TE LLEGARÁN PRONTO

Ref.: Solicitud de firma ...
H. Ruta: 0003 Fecha: 12:47, 29/11/...

+ CARGAR MÁS

Figura 52 Página de creación de trámites

Fuente: Elaboración propia

3.4.6. Administración de unidades

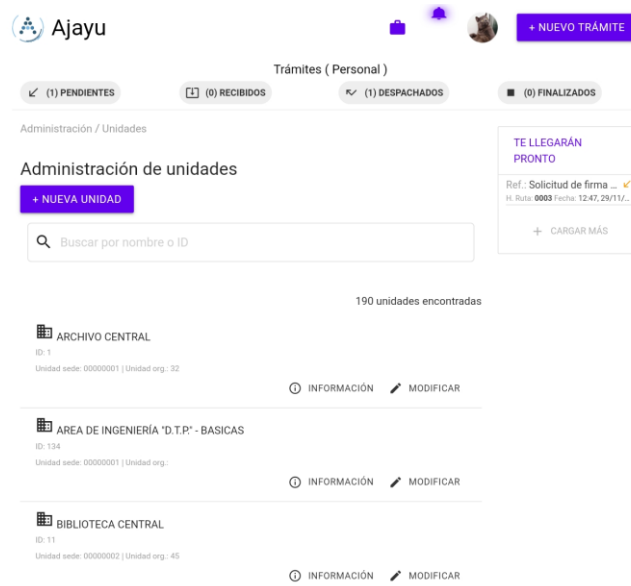


Figura 53 Página de administración de unidades
Fuente: Elaboración propia

3.4.7. Administración de usuarios

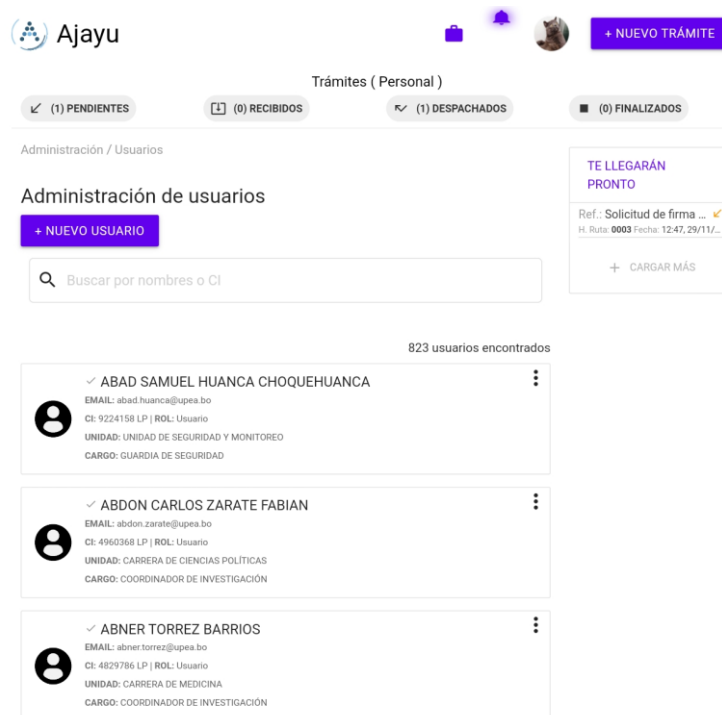


Figura 54 Página de administración de usuarios
Fuente: Elaboración propia

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

El sistema web permite a la comunidad universitaria que solicita y desea hacer seguimiento a sus trámites información rápida y oportuna sobre los mismos, así también permite al personal administrativo de la institución un manejo fácil y confiable sobre los trámites que se registran y mueven dentro de la institución.

4.1. CONCLUSIONES

Finalizado el desarrollo y despliegue del “**SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES**” para la Universidad Pública de El Alto, se ha logrado alcanzar los objetivos planteados, bajo los requerimientos de la institución.

Se llegaron a las siguientes conclusiones:

- Se logró diseñar una interfaz amigable y de fácil manipulación para el usuario
- El sistema agiliza los tiempos de acceso a la información.
- Se logró un control sobre el registro y manejo de trámites.
- Se logró realizar el manual de usuario correspondiente al sistema.
- Se logró disminuir los tiempos en la generación de reportes.

De esta forma, se alcanzó el objetivo principal de proporcionar acceso a información confiable y oportuna sobre el estado y avance de trámites, además de facilitar al personal administrativo facilidad para el manejo y registro de trámites.

4.2. RECOMENDACIONES

A partir del presente trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Se recomienda delegar trámites simples por completo al sistema y dejar la documentación física.
- Se recomienda la elaboración de un sistema web que permita al personal administrativo la redacción, modificación y almacenamiento de documentos.
- Se recomienda la implementación de la firma digital para documentos de la institución.

BIBLIOGRAFÍA

- Allianza Mundial para la Participación Ciudadana. (s.f.). *civicus*. Obtenido de https://www.civicus.org/view/media/Seguimiento%2520y%2520evaluacion.pdf&ved=2ahUKEwjuzP3p-PzyAhW_GFkFHYECDeEQFnoECAYQBg&usg=AOvVaw0qnWqouw5q31Ao03KghEwC
- ARDILA, N. I. (12 de Marzo de 2013). *actividadreconocimiento-301569-8.blogspot.com*. Obtenido de <http://actividadreconocimiento-301569-8.blogspot.com/2013/>
- Baez, S. (20 de Octubre de 2012). *knowdo.org*. Obtenido de <http://www.knowdo.org/knowledge/39-sistemas-web>
- Baquero. (Diciembre de 2015). *Arsys*. Obtenido de <https://www.arsys.es/>
- Calero, W. (07 de octubre de 2010). *COCOMO*. Obtenido de Ingeniería de Software: <http://ingenieraupoliana.blogspot.com/2010/10/cocomo.html>
- CHICUNQUE, J. M. (2014). *senadsi2014.wordpress.com*. Obtenido de <https://senadsi2014.wordpress.com/21-ejemplos-y-ejercicios-resueltos-de-diagramas-de-caso-de-uso/>
- Choque, M. A. (2009). Sistema de control y seguimiento de trámites vía web para el Ministerio de Trabajo. *Sistema de control y seguimiento de trámites vía web para el Ministerio de Trabajo*. La Paz: Universidad Mayor de San Andres.
- Cobo, Gómez, Pérez, & Rocha. (2005). *PHP y MySQL: Tecnología Para el Desarrollo de Aplicaciones Web*. España: Díaz de Santos.
- COMISIÓN FEDERAL DE MEJORA REGULATORIA. (28 de Septiembre de 2011). *conamer*. Obtenido de <https://www.conamer.gob.mx/contenido.aspx?contenido=22>
- Cover, R. (10 de Agosto de 2000). *coverpages.org*. Obtenido de <http://xml.coverpages.org/webML.html>
- Damorelos. (Febrero de 2019). *scio*. Obtenido de <https://www.scio.com.mx>
- Deshpande, Y. (Octubre de 2002). *acm.org*. Obtenido de <https://dl.acm.org/doi/10.5555/20111098.2011101>
- Fernandez. (2016). *Repositorio UMSA*. Obtenido de <https://repositorio.umsa.bo/bitstream/handle/123456789/10000/T.3247.pdf?sequence=1&isAllowed=y>.
- ibm. (s.f.). *ibm.com*. Obtenido de <https://www.ibm.com/topics/software-testing>

Jústiz-Núñez, D. (2014). Proceso de pruebas para productos de software en un. *Instituto Superior Politécnico José Antonio*, 15.

Marquez, A. (17 de Marzo de 2020). *testermoderno.com*. Obtenido de <https://www.testermoderno.com/prueba-de-carga-vs-prueba-de-estres-load-test-vs-stress-test/>

micarrerauniversitaria. (s.f.). *micarrerauniversitaria.com*. Obtenido de <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-web/>

München, L.-M.-U. (10 de 08 de 2016). *uwe.pst.ifi.lmu.de*. Obtenido de <https://uwe.pst.ifi.lmu.de/aboutUwe.html>

Naumenko, V. (s.f.). *jelvix.com*. Obtenido de <https://jelvix.com/blog/software-development-methodologies>

Parada. (Octubre de 2019). *Openwebinars*. Obtenido de <https://openwebinars.net>

Pressman. (2010). *Ingeniería de Sistemas*. México: McGrall Hill.

Pressman, R. (2010). *Ingeniería del software Un enfoque práctico*. McGraw-Hill.

Rojas-Montes, M. L., Pino-Correa, F. J., & Martínez, J. M. (1 de Febrero de 2015). *Proceso de pruebas para pequeñas organizaciones*. Obtenido de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292015000200006

Romero. (s.f.). *Espacios*. Obtenido de <https://www.espacios.media>

Salas. (Junio de 2017). *Blogger*. Obtenido de <http://marcelosalasvargas.blogspot.com>

Stringfellow, A. (21 de Septiembre de 2017). *stackify.com*. Obtenido de <https://stackify.com/web-application-architecture/>

Taina, J. (5 de Octubre de 2005). *helsinki.fi*. Obtenido de <https://www.cs.helsinki.fi/u/taina/ohtu/fp.html>

Vásquez, T. N. (2012). */repositorio.puce.edu.ec*. Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/6360/9.21.001138.pdf%3Bsequence%3D4&ved=2ahUKEwjYrKHPgP3yAhVjD1kFHYftCccQFnoECACQAQ&usq=AOvVaw1VKLIhfaR07784gMMj4DF0>

Zamora. (s.f.). *Scribd*. Obtenido de <https://es.scribd.com/document/136176327/Webml>

Zwass, V. (12 de 10 de 2001). *Information system*. Obtenido de [britannica.com: https://www.britannica.com/topic/information-system](https://www.britannica.com/topic/information-system)

ANEXOS

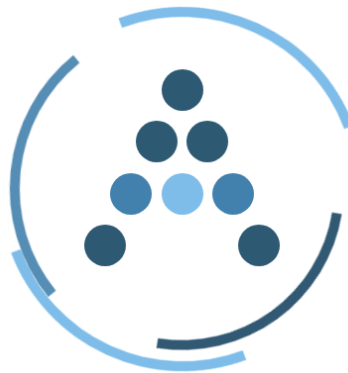


Universidad Pública de El Alto

Creada por Ley 2115 del 5 de septiembre de 2000 y Autónoma por Ley 2556 del 12 de noviembre de 2003



UNIDAD SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA



MANUAL DE USUARIO

Sistema de Seguimiento de Trámites “AJAYU”

Versión 1.0

Descripción	Designación de tarea(s)	Tarea(s) realizada(s) por:	Fecha
Sistema de Seguimiento de Trámites “Ajayu”	Elaboración y presentación	Jhonny Apaza Chambi	2021

ÍNDICE

1. DESTINATARIOS	
2. TERMINOS	
3. FUNCIONALIDADES DEL SISTEMA	
3.1. Ingreso al sistema	
3.2. Hacer seguimiento a los trámites	
3.3. Ingreso al sistema del personal administrativo	
3.4. Recuperar el acceso a la cuenta	
3.5. Secciones de la vista principal de usuario	
3.6. Cuenta de usuario	
3.7. Salir del sistema	
3.8. Módulos del sistema	

ÍNDICE DE FIGURAS

Figura 1: Barra de navegación	
Figura 2: Vista principal del sistema	
Figura 3: Sección para hacer seguimiento de trámites	
Figura 4: Llenado de campos para hacer seguimiento de trámites	
Figura 5: Resultado del seguimiento de un trámite	
Figura 6: Ventana emergente para acceder al sistema como administrativo	
Figura 7: Ventana emergente para cambiar la contraseña si se ingresa al sistema por primera vez	
Figura 8: Vista principal al acceder al sistema	
Figura 9: Ventana emergente para recuperar la cuenta de usuario	
Figura 10: Confirmación de la solicitud para recuperar la cuenta de usuario	
Figura 11: Correo electrónico para recuperar cuenta	
Figura 12: Ventana emergente para cambiar la contraseña	
Figura 13: Confirmación de cambio de contraseña	
Figura 14: Secciones de la vista principal de usuario	
Figura 15: Barra lateral de operaciones	

Figura 16: Barra de contadores Personal.....	
Figura 17: Barra de contadores Unidad	
Figura 18: Acceder a la vista “Mi Cuenta”	
Figura 19: Vista “Mi cuenta”	
Figura 20: Pestaña “Información Personal” de la vista “Mi Cuenta”	
Figura 21: Pestaña “Cambiar Foto de Perfil” de la vista “Mi Cuenta”	
Figura 22: Pestaña “Cambiar Contraseña” de la vista “Mi Cuenta”	
Figura 23: Salir del sistema.....	
Figura 24: Notificación de envió de trámite a un usuario.....	
Figura 25: Vista de Trámites pendientes de recepción	
Figura 26: Ventana emergente para confirmar la recepción de un trámite.....	
Figura 27: Vista de los Trámites Recibidos	
Figura 28: Ventana emergente para hacer seguimiento al trámite recibido	
Figura 29: Ventana emergente para Finalizar un trámite	
Figura 30: Ventana emergente Despachar el trámite a otro usuario	
Figura 31: Vista de Trámites Despachados a otros usuarios	
Figura 32: Vista de Trámites Finalizados	
Figura 33: Acceso a los trámites desde la barra de contadores.....	
Figura 34: Acceso a los trámites desde la barra lateral de operaciones	

1. Destinatarios

Comunidad universitaria

2. Terminos

- **Usuario.** Se refiere a las personas que podrán interactuar con el sistema
- **Dominio.** Es el nombre que identifica al sitio Web.
- **Base de datos.** “Almacén” de grandes cantidades de datos.
- **Vista.** Es el formato de texto y diseño de página que se muestra.
- **Ventana emergente.** Ventana dentro del navegador que se superpone al resto de contenido
- **Sesión.** Actividad del usuario dentro del sistema

3. Funcionalidades del Sistema

3.1. Ingreso al sistema

Para ingresar al sistema, se debe escribir el dominio en la barra de navegación, que puede ser cambiado y que mientras no sea cambiado es el siguiente ajayu.upea.bo.

Para ingresar al sistema el usuario debe escribir el dominio en el navegador.



Figura 1. Barra de navegación, donde se escribe el dominio para ingresar al sistema

Al poner e ingresar el dominio en el navegador se le mostrará la vista inicial del sistema.

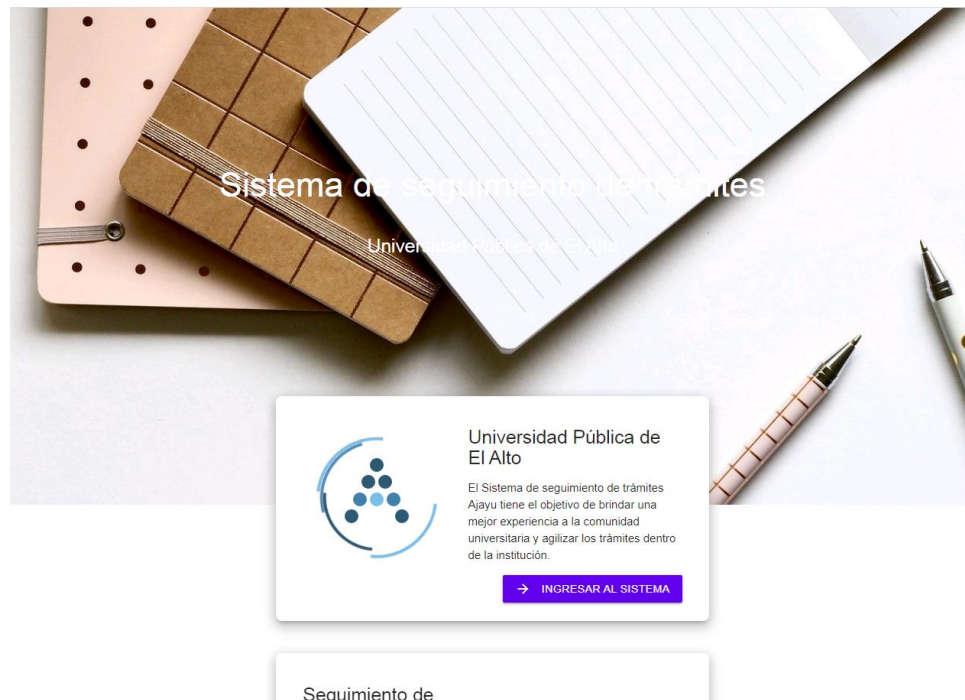


Figura 2. Vista principal del sistema

3.2. Hacer seguimiento a los trámites.

Para hacer el seguimiento a un trámite, es necesario estar registrado en la base de datos de la Universidad Pública de El Alto, y conocer el número de hoja de ruta del trámite y su gestión.

En la vista principal del sistema localice el enunciado “Seguimiento de trámites”.

Figura 3. Sección para hacer seguimiento de trámites

Ingrese su número de Carnet de Identidad (CI) sin la extensión, seleccione la gestión (año) a la que corresponde el trámite y por ultimo ingrese el código de seguimiento, después haga clic en el botón “BUSCAR”.

Seguimiento de trámites

NÚMERO DE CI

Número de CI (sin expedido)
6878587

Gestión
2021

Código de segui...
LK5SP

CÓDIGO DE SEGUIMIENTO

GESTIÓN

BUSCAR

Detailed description: The image shows a web form titled 'Seguimiento de trámites'. It contains three input fields: a text box for 'Número de CI (sin expedido)' with the value '6878587', a dropdown menu for 'Gestión' with '2021' selected, and a text box for 'Código de segui...' with the value 'LK5SP'. Below these fields is a purple button with a magnifying glass icon and the text 'BUSCAR'. Red lines with labels point to each field: 'NÚMERO DE CI' points to the CI field, 'GESTIÓN' points to the dropdown, and 'CÓDIGO DE SEGUIMIENTO' points to the tracking code field.

Figura 4. Llenado de campos para hacer seguimiento de trámites

Si los datos son correctos el sistema le mostrará los datos relacionados al trámite, como ser las unidades que ha recorrido, la fecha de ingreso a cada unidad, y si el trámite está pendiente de ser finalizado o ya lo está.

Seguimiento de trámites

Número de CI (sin expedido)
6878786

Gestión: 2021 Código de segui...: 2BR1U

BUSCAR

Resultado

✓ Jhonny, encontramos esto para ti :

Trámite finalizado
 Anotación: **SOLICITUD RECHAZADA**
 Finalizado a las 20:25 del 30-11-2021

Número de Hoja de Ruta: 0001

Gestión: 2021

Referencia: SERVICIO DE CERTIFICACION FIRMA DIGITAL

Remitente: MARIA JANNETT IBAÑEZ

Dirigido a: JUAN REGIS MUÑEZ SIRPA - JEFE DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

Fecha de recepción: 30/11/2021

Fecha de finalización: 30/11/2021

Seguimiento

1 UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

✓ Recibido el 30 de Noviembre de 2021

Trámite finalizado
 Anotación: **SOLICITUD RECHAZADA**
 Finalizado a las 20:25 del 30-11-2021

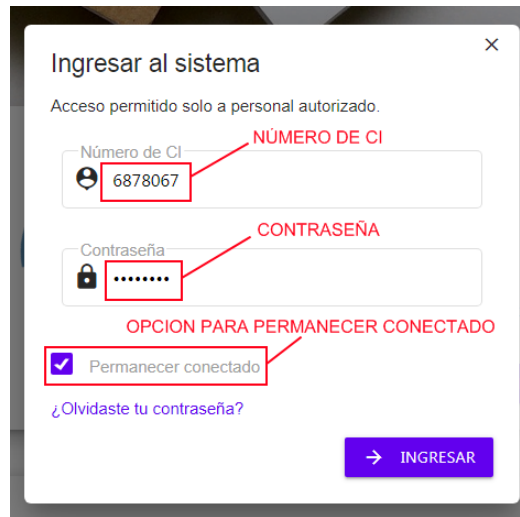
Figura 5. Resultado del seguimiento de un trámite

3.3. Ingreso al sistema del personal administrativo

Si pertenece al personal administrativo de la institución, usted tiene una cuenta en el sistema para acceder a varias funciones, para acceder como administrativo haga clic en el botón “INGRESAR AL SISTEMA”, se mostrará una ventana emergente en la que podrá ingresar sus datos para acceder al sistema.

Debe ingresar su número de Carnet de Identidad (CI) sin la extensión, y su contraseña, opcionalmente puede marcar opción “Permanecer conectado” para que su sesión permanezca abierta de forma permanente en el navegador.

Si es la primera vez que ingresa al sistema, su contraseña será su fecha de nacimiento en el formato DIA-MES-AÑO, por ejemplo: 26-11-1989.



The image shows a login window titled "Ingresar al sistema" with a close button (X) in the top right corner. Below the title, it says "Acceso permitido solo a personal autorizado." There are three input fields: "Número de CI" containing "6878067", "Contraseña" containing ".....", and a checkbox labeled "Permanecer conectado" which is checked. A link "¿Olvidaste tu contraseña?" is below the checkbox. A purple button labeled "INGRESAR" with a right arrow is at the bottom right. Red annotations with arrows point to the "Número de CI" field (labeled "NÚMERO DE CI"), the "Contraseña" field (labeled "CONTRASEÑA"), and the "Permanecer conectado" checkbox (labeled "OPCION PARA PERMANECER CONECTADO").

Figura 6. Ventana emergente para acceder al sistema como administrativo

Si es la primera vez que ingresas al sistema, inmediatamente se te mostrará una ventana emergente para que cambies tu contraseña de forma obligatoria, debes ingresar tu nueva contraseña en ambos campos, ten en cuenta que tu nueva contraseña debe tener como mínimo 8 caracteres entre ellos al menos un número, una letra mayúscula y otra minúscula.

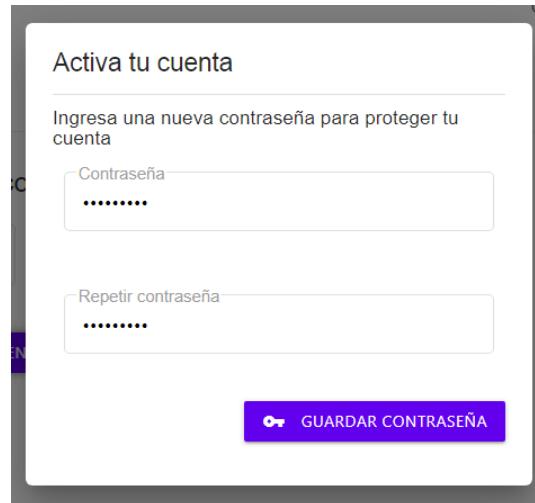


Figura 7. Ventana emergente para cambiar la contraseña si se ingresa al sistema por primera vez

Una vez acceda al sistema, se le mostrara la vista inicial de usuario.

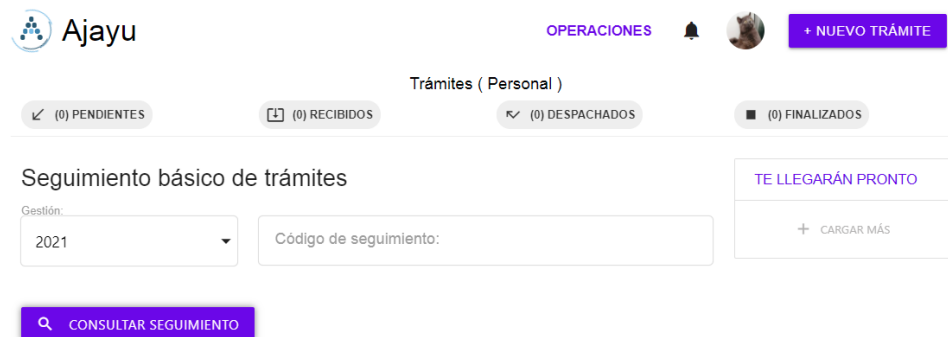


Figura 8. Vista principal al acceder al sistema

3.4. Recuperar el acceso a cuenta de usuario

En caso de que hayas olvidado tu contraseña y no puedas acceder a tu cuenta, en la vista inicial del sistema, en la ventana emergente para ingresar tu CI y contraseña, encontraras el enlace “¿Olvidaste tu contraseña?”, al hacer clic en ese enlace se mostrara una nueva ventana

emergente en la que podrás ingresar tu número de Carnet de Identidad (CI) y tu fecha de nacimiento (en el formato AÑO-MES-DÍA, por ejemplo: 1979-11-25) para que se envíe un correo electrónico a tu dirección de correo electrónico institucional xxxxxxx@upea.bo que contiene un enlace mediante el cual puedes cambiar tu contraseña. Ten en cuenta que solo puedes solicitar recuperar tu contraseña una vez cada 24 horas, y que el enlace que se te enviará tiene una validez de 1 hora.

Recuperar cuenta

Ingresa los siguientes datos para recuperar tu cuenta, enviaremos un correo electrónico a tu correo electrónico institucional para que puedas restablecer tu contraseña.

Número de CI **NÚMERO DE CI**
4807896

Fecha de nacimiento (AÑO-MES-DÍA)
1985-07-21

**FECHA DE NACIMIENTO
AÑO-MES-DÍA** → RECUPERAR CUENTA

Figura 9. Ventana emergente para recuperar la cuenta de usuario

Recuperar cuenta

✓ Se envió un email a tu correo carlos.condori@upea.bo con un enlace para cambiar tu contraseña, este enlace tiene una validez de una hora.

Figura 10. Confirmación de la solicitud para recuperar la cuenta de usuario

Recibirás un correo electrónico como el siguiente:

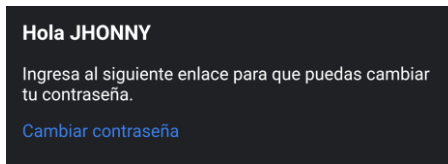


Figura 11. Correo electrónico para recuperar la cuenta

Al ingresar al enlace se mostrará la siguiente ventana para que puedas cambiar tu contraseña, ingresa tu nueva contraseña y presiona el botón “Cambiar contraseña”.

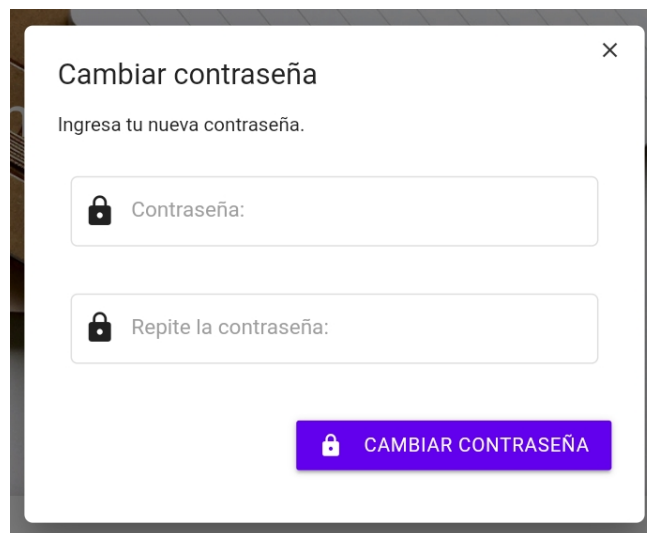


Figura 12. Ventana emergente para cambiar la contraseña

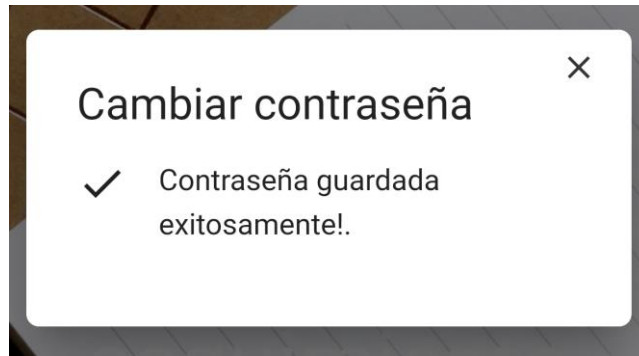


Figura 13. Confirmación de cambio de contraseña

3.5. Secciones de la vista principal de usuario

Cuando accedes a tu cuenta notarás que existen diferentes secciones en la pantalla, estos tienen funciones específicas y se describen a continuación:



Figura 14. Secciones de la vista principal de usuario

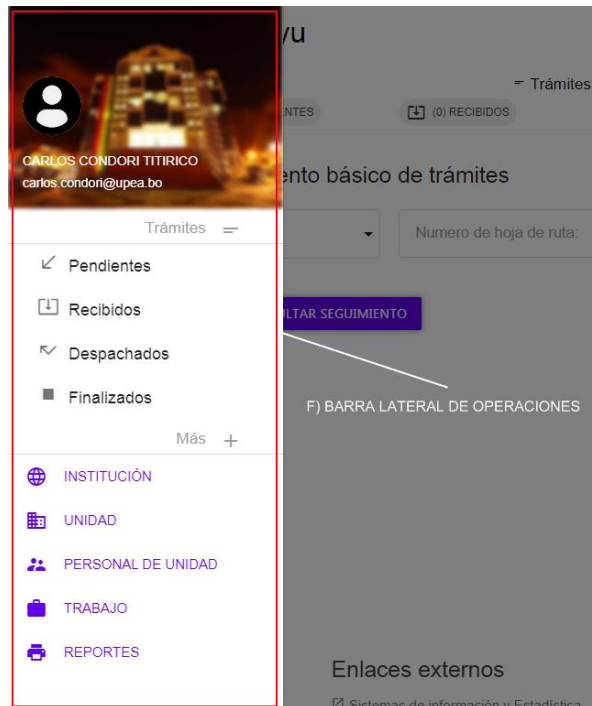


Figura 15. Barra lateral de operaciones

A). Barra de navegación, Contiene las opciones de usuario para navegar dentro del sistema, desde ella puedes accionar la Barra lateral de operaciones, ver tus notificaciones, ver y acceder a opciones de tu cuenta (como Salir del sistema, o ver los detalles de tu cuenta), y si tu cuenta tiene los permisos necesarios podras ver el botón “+ NUEVO TRÁMITE” para crear trámites.

B). Barra de contadores, En esta sección se muestran la cantidad de tramites en forma de botones de la siguiente forma:

a. Trámites Pendientes. - Son los trámites que se enviaron a tu persona, pero su recepción aún no está confirmada, al hacer clic en este botón podrás ver todos los detalles de los

trámites que otros usuarios te enviaron y confirmar su recepción.

- b. Trámites Recibidos.** - Son los trámites que confirmaste como recibidos, es decir que ya recibiste la documentación física.
- c. Trámites Despachados.** - Son los trámites que despachaste a otros usuarios del sistema.
- d. Trámites Finalizados.** – Son los trámites que finalizaste, es decir trámites que tu hayas puesto fin.

Esta barra de contadores se actualiza automáticamente en tiempo real, si algún usuario te envía un trámite el contador de trámites “Pendientes” se incrementará.

Dependiendo de los permisos de tu cuenta de usuario, se te mostrara los contadores de trámites Pendientes, Recibidos, Despachados y Finalizados de tu persona o de tu unidad.

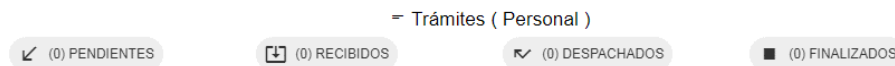


Figura 16. Barra de contadores Personal

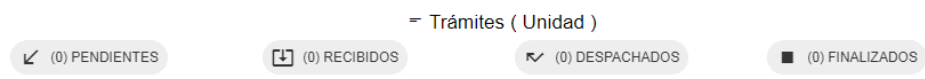


Figura 17. Barra de contadores de la Unidad

C). Barra de acceso rápido, En esta sección se muestran de forma resumida listas de:

- a. Trámites pendientes de recepción.** - Para tu persona o para tu unidad de acuerdo a los permisos de tu cuenta.
- b. Tareas pendientes de finalizar.** - Son tareas que te fueron asignadas y aun no las finalizaste.

Estas listas se actualizan de forma automática en tiempo real, además puedes hacer clic en los elementos de ambas listas para confirmar la recepción del trámite seleccionado en el caso de la lista de Trámites pendientes de recepción, y finalizar la tarea seleccionada en el caso de la lista de Tareas pendientes.

D). Contenido principal, Esta es la única sección que cambiara de acuerdo a las acciones que quieras ejecutar, el contenido de esta sección no es permanente como el de las demás secciones en las que se muestran cosas específicas.

E). Pie de página (Footer), En esta sección se muestran enlaces a otros sistemas y sitios web de la institución.

F). Barra lateral de operaciones, En esta sección puedes acceder a todas las operaciones que puedes realizar de acuerdo a los permisos de tu cuenta, como hacer seguimiento detallado de trámites, Crear un nuevo trámite, cambiar las instrucciones de las hojas de ruta de tu unidad, etc.

3.6. Cuenta de usuario

Dentro del sistema puedes acceder a la opción “Mi cuenta” al hacer clic en la foto de usuario de la barra de navegación.

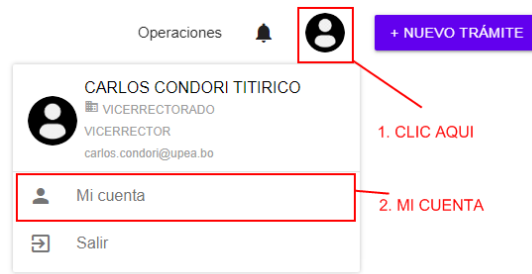


Figura 18. Acceder a la vista “Mi Cuenta”

Dentro de la vista “Mi cuenta” puedes ver los permisos que tiene asignada tu cuenta, la fecha de eliminación de la misma (no en todos los casos se muestra esta fecha), además de tres pestañas:

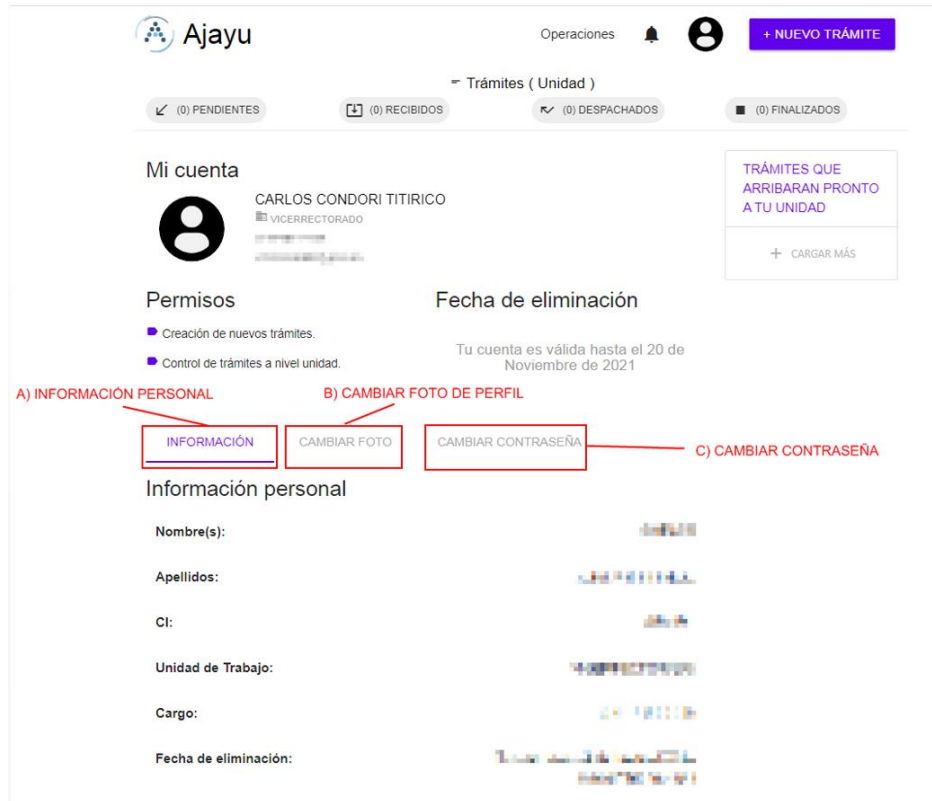


Figura 19. Vista “Mi Cuenta”

- 1) **Información Personal**, Aquí puedes consultar la información que tiene el sistema sobre ti, en caso de ser información incorrecta contacta con los administradores del sistema.



Figura 20. Pestaña “Información Personal” de la vista “Mi Cuenta”

2) Cambiar Foto de Perfil, Aquí puedes opcionalmente agregar una foto tuya para que los demás usuarios puedan saber quién eres, para cambiar tu foto tendrás que ingresar tu contraseña actual, puedes cambiar tu foto de perfil las veces que desees.



The screenshot shows a user interface for changing a profile picture. At the top, there are three tabs: 'INFORMACIÓN', 'CAMBIAR FOTO' (which is highlighted with a purple underline), and 'CAMBIAR CONTRASEÑA'. Below the tabs, the text 'Cambiar mi foto de perfil' is displayed. In the center, there is a circular placeholder for a profile picture, currently showing a camera icon. Below the placeholder, the instruction 'Ingrese su contraseña actual para guardar los cambios' is shown. There are two input fields for passwords: the first is labeled 'Contraseña' and the second is labeled 'Ingrese su contraseña nuevamente'. Both fields have a lock icon on the left. At the bottom right, there is a purple button labeled 'GUARDAR FOTO'.

Figura 21. Pestaña “Cambiar Foto de Perfil” de la vista “Mi Cuenta”

3) Cambiar Contraseña, Aquí puedes cambiar la contraseña para acceder a tu cuenta de usuario, ten en cuenta que solo puedes cambiar tu contraseña una vez cada 7 días, tendrás que ingresar tu contraseña nueva y tu contraseña actual para hacer el cambio.

INFORMACIÓN CAMBIAR FOTO CAMBIAR CONTRASEÑA

Cambiar mi contraseña

Ingresa tu nueva contraseña

Contraseña

Ingresa su contraseña nuevamente

Ingresa su contraseña actual para guardar los cambios

Contraseña

Ingresa su contraseña nuevamente

GUARDAR CAMBIOS

Figura 22. Pestaña “Cambiar Contraseña” de la vista “Mi Cuenta”

3.7. Salir del sistema

Para salir del sistema solo debe presionar la foto de perfil de la Barra de Navegación, y hacer clic en el botón “Salir” entonces su sesión será cerrada.

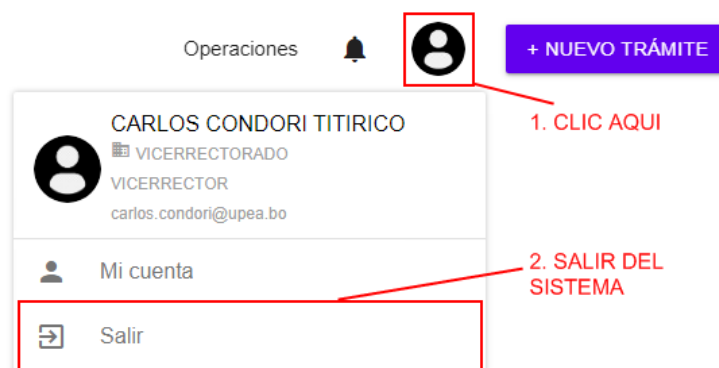


Figura 23. Salir del sistema

3.8. Módulos del sistema

El sistema cuenta con distintos módulos que permiten su funcionamiento, son los siguientes:

3.8.1. Módulo de Trámites

Es el modulo principal del sistema, todos los Administrativos de la institución tienen acceso a él, y es necesario conocer el proceso que sigue cada trámite dentro del sistema, y como un usuario interactúa con ellos dentro del sistema.

Proceso básico del trámite

- 1) **Creación.** El trámite es creado ingresando al sistema los datos del trámite, datos de la hoja de ruta, datos del remitente, y los datos del primer despacho.

Al crear el trámite debe despacharse a un usuario del sistema.

- 2) **Curso.** El trámite sigue un curso dentro de la institución por ejemplo despachándose a distintas personas dentro de una misma unidad o entre distintas unidades de la institución, adjuntándose a ella notas, informes, etc.

Dentro del sistema debe reflejarse ese proceso y es por eso que el módulo de Trámites es accesible a todos los administrativos de la institución.

- 3) **Finalización.** El trámite llega a su fin y es necesario notificar al interesado sobre esto.

Interacción del usuario con los trámites

El usuario dentro del sistema interactúa con los trámites de la siguiente forma:

- 1) **Espera el Trámite.** Que es enviado al por otro usuario y que el sistema ya ha notificado que se le ha enviado.
- 2) **Recibe el Trámite.** Recibe la documentación física del trámite y confirma en el sistema la recepción del trámite.
- 3) **Atiende el Trámite.** Cuando el usuario recibe el trámite puede suceder lo siguiente:
 - **Despachar el Trámite a otro usuario.** - Puede despachar el trámite a otro usuario añadiendo alguna anotación.
 - **Finalizar el trámite.** – Puede finalizar el trámite añadiendo una anotación final, y opcionalmente puede contactar a la persona que inicio o solicito el trámite para avisarle sobre la finalización del mismo.

Todo lo anterior se ve reflejado en 4 secciones que se muestran a todos los usuarios:

- **Trámites Pendientes de Recepción.** - Son los trámites que dentro del sistema un usuario te envía, pero están “Pendientes” porque tu aun no recibiste la documentación física del mismo, una vez que recibas esta documentación debes marcar el trámite como “Recibido”.

Cuando otro usuario del sistema te envía un trámite mediante el sistema recibirás una notificación que podrás ver en la barra de navegación haciendo clic en el botón con el icono de una Campana.

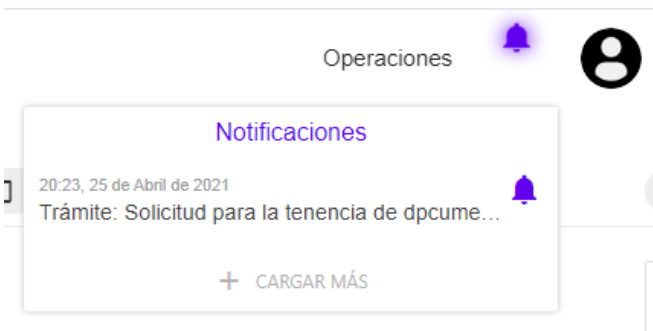


Figura 24. Notificación de envió de trámite a un usuario

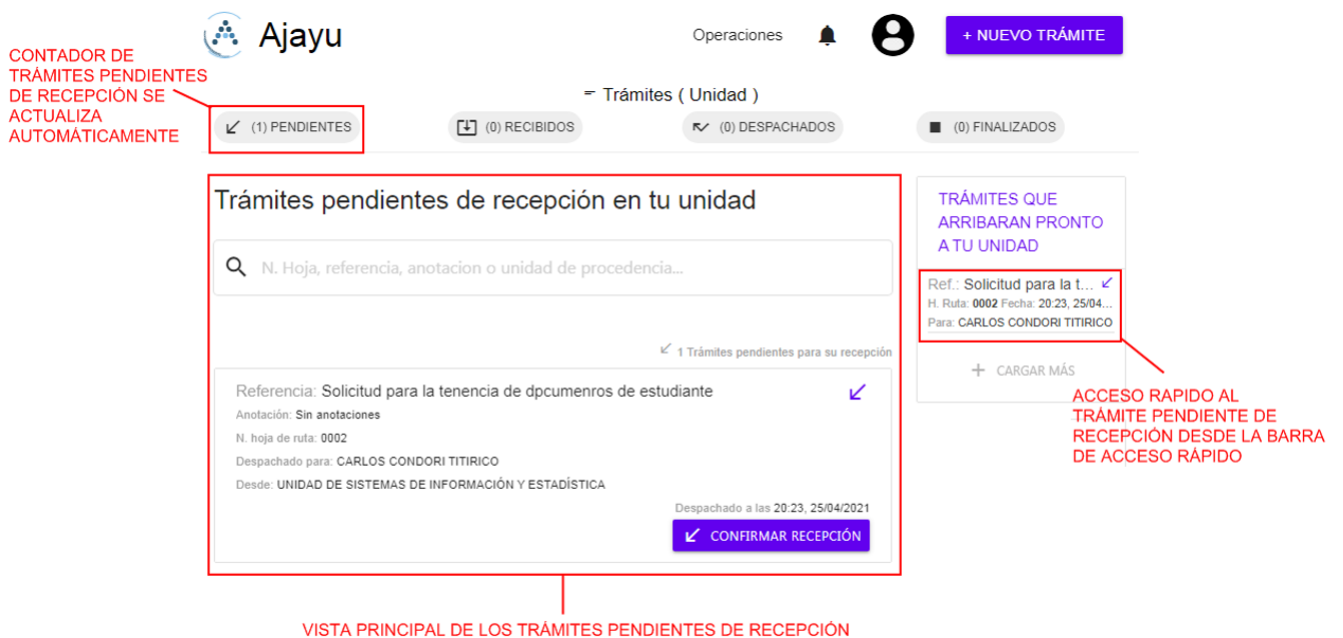


Figura 25. Vista de Trámites pendientes de recepción

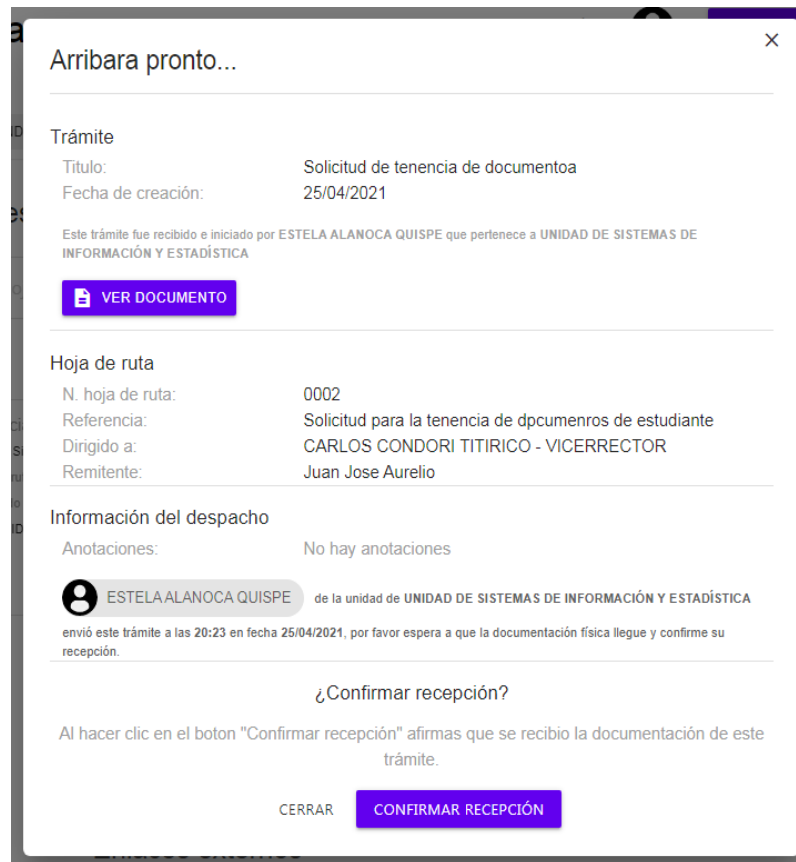


Figura 26. Ventana emergente para confirmar la recepción de un trámite

- **Trámites Recibidos.** – Son los trámites que confirmaste como “Recibidos” pero aun no hiciste nada con ellos, en esta vista puedes hacer seguimiento al trámite que recibiste, despachar el trámite a otra persona, y finalizar el trámite.

= Trámites (Personal)

(0) PENDIENTES

(1) RECIBIDOS

(0) DESPACHADOS

(0) FINALIZADOS

CONTADOR DE TRÁMITES RECIBIDOS ACTUALIZADO AUTOMATICAMENTE

Trámites recibidos

N. Hoja, remitente, dirigido a, origen, anotacion o referencia

1 Trámites recibidos

Referencia: Solicitud para la tenencia de dpcumenros de estudiante
N. hoja de ruta: 0002
Dirigido a: CARLOS CONDORI TITIRICO - VICERRECTOR
Remitente: Juan Jose Aurelio
Origen: UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA
Anotación: Sin anotaciones

FINALIZAR TRÁMITE

Recibido a las 20:37, 25/04/2021

SEGUIMIENTO



FINALIZAR

DESPACHAR

DESPACHAR TRÁMITE A OTRO USUARIO

VISTA DE LOS TRÁMITES RECIBIDOS

Figura 27. Vista de los Trámites Recibidos

Información trámite

Trámite

Título: Solicitud de tenencia de documentoa
Fecha de creación: 25/04/2021

Este trámite fue recibido e iniciado por ESTELA ALANOCA QUISPE que pertenece a UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

[VER DOCUMENTO](#)

Hoja de ruta

N. hoja de ruta: 0002
Gestión: 2021
Referencia: Solicitud para la tenencia de dpcumenros de estudiante
Dirigido a: CARLOS CONDORI TITIRICO - VICERRECTOR
Remitente: Juan Jose Aurelio
N. de fojas: 1 N. de files: 0
Fecha recepción: 25/04/2021 Hora recepción: 20:23

[IMPRIMIR HOJA DE RUTA](#)

Seguimiento


- UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA
 -  ESTELA ALANOCA QUISPE
 - ✓ Recibido a las 20:23, 25-04-2021
- VICERRECTORADO

Figura 28. Ventana emergente para hacer seguimiento al trámite recibido

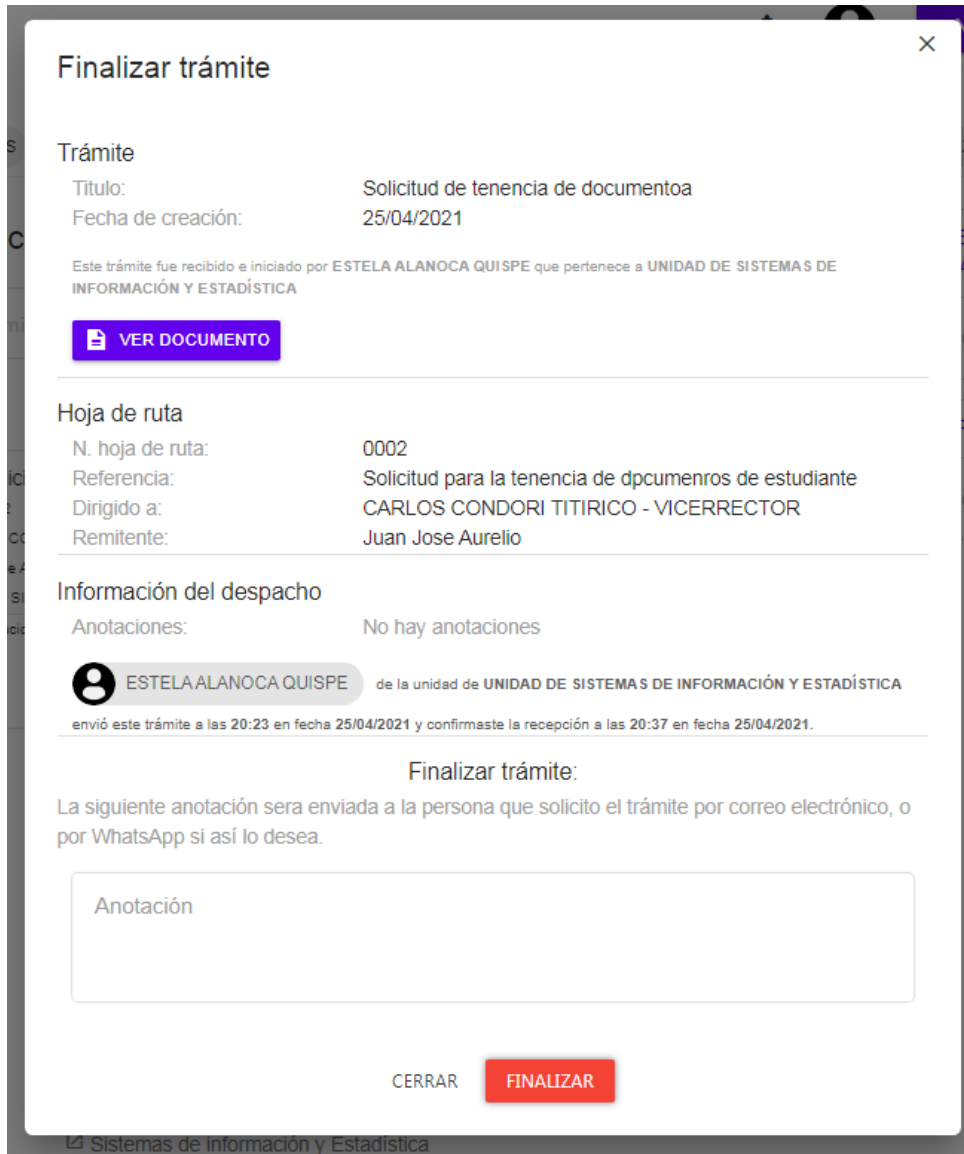


Figura 29. Ventana emergente para Finalizar un trámite

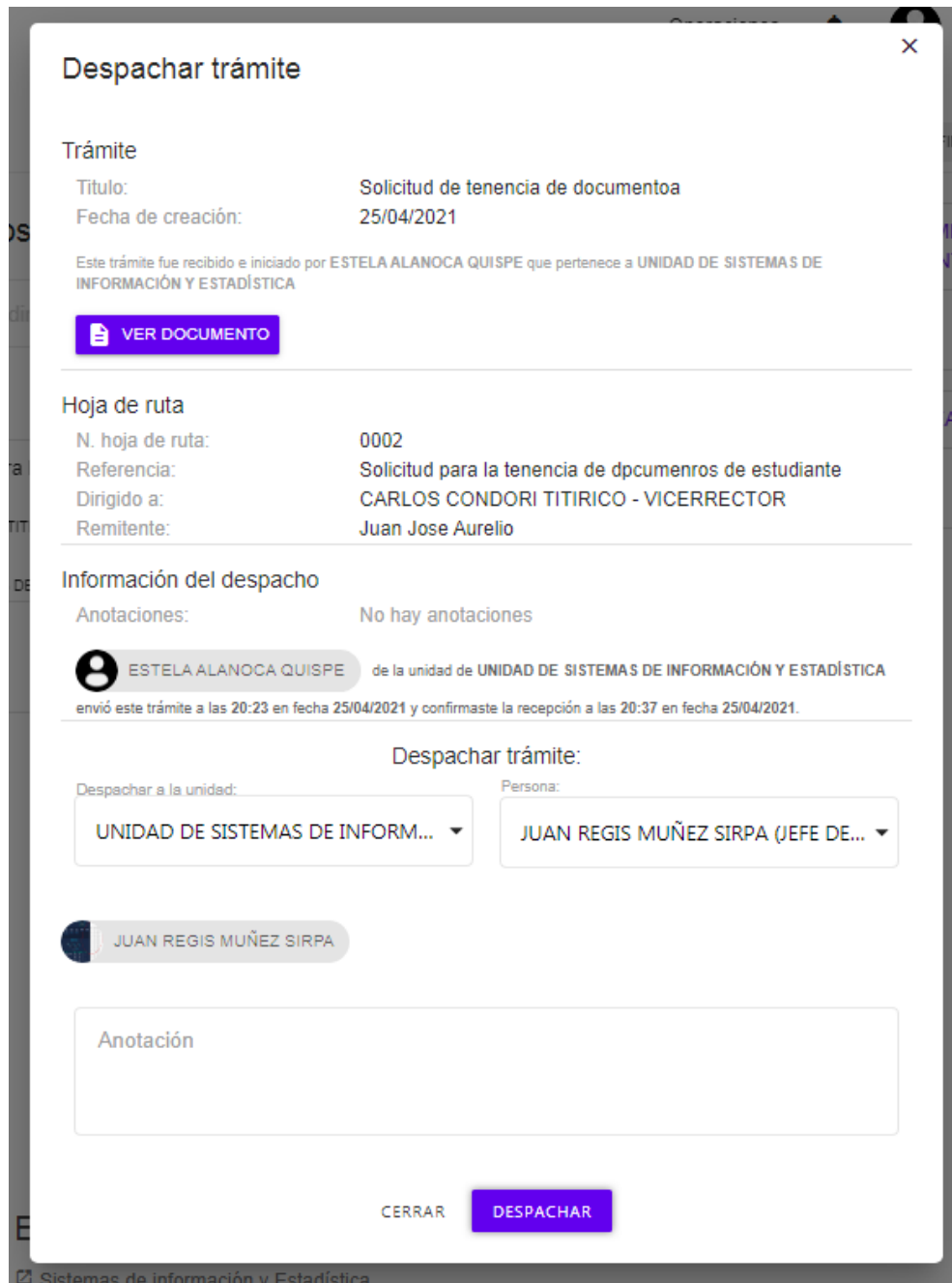


Figura 30. Ventana emergente Despachar el trámite a otro usuario

- **Trámites Despachados.** – En esta vista se muestra el historial de trámites que hayas despachado a otros usuarios del sistema, adicionalmente puede hacer el seguimiento de esos mismos

trámites de la misma manera que en la vista de Trámites Recibidos.

Ajayu

Operaciones

+ NUEVO TRÁMITE

Trámites (Personal)

(0) PENDIENTES (0) RECIBIDOS (1) DESPACHADOS (0) FINALIZADOS

CONTADOR DE TRÁMITES DESPACHADO ACTUALIZADO AUTOMÁTICAMENTE

Trámites despachados

N. Hoja, referencia, dirigido a, remitente, u origen...

+ CARGAR MÁS

1 Trámites despachados

Referencia: Solicitud para la tenencia de documentos de estudiante ✓

N. hoja de ruta: 0002

Dirigido a: CARLOS CONDORI TITIRICO - VICERRECTOR

Remitente: Juan Jose Aurelio

Origen: UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

Despachado a: UNIDAD DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

Despachado a las 20:51, 25/04/2021

SEGUIMIENTO

SEGUIMIENTO

VISTA DE TRÁMITES DESPACHADOS

Figura 31. Vista de Trámites Despachados a otros usuarios

- **Trámites finalizados.** – En esta vista puedes ver el historial de trámites que hayas finalizado, adicionalmente al igual que en la vista de Trámites Recibidos puedes hacer seguimiento de esos trámites.

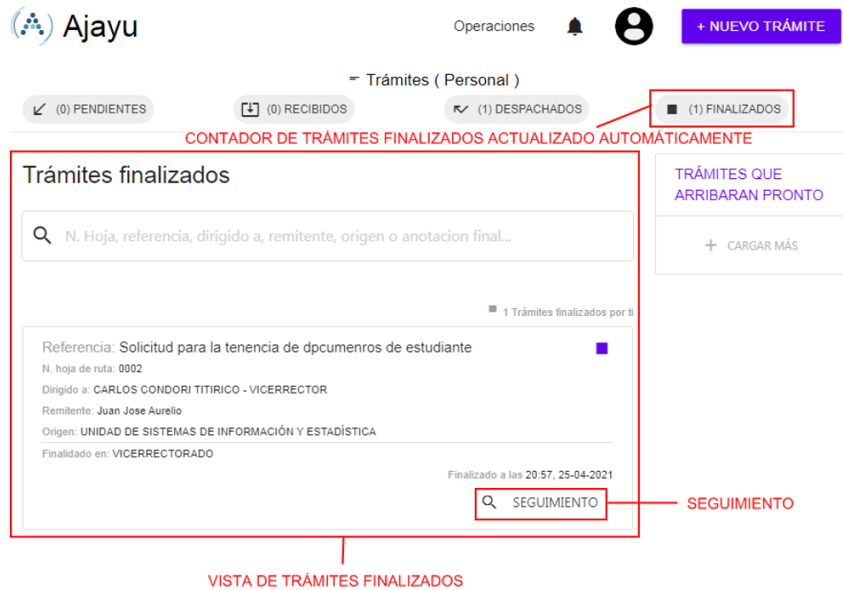


Figura 32. Vista de Trámites Finalizados

Información adicional

Puedes acceder a cualquiera de las 4 vistas (trámites pendientes de recepción, recibidos, despachados y finalizados) haciendo clic a los botones de la barra de contadores, o los de la barra lateral de operaciones.

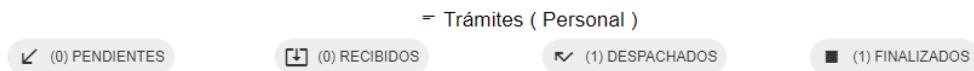


Figura 33. Acceso a los trámites desde la barra de contadores

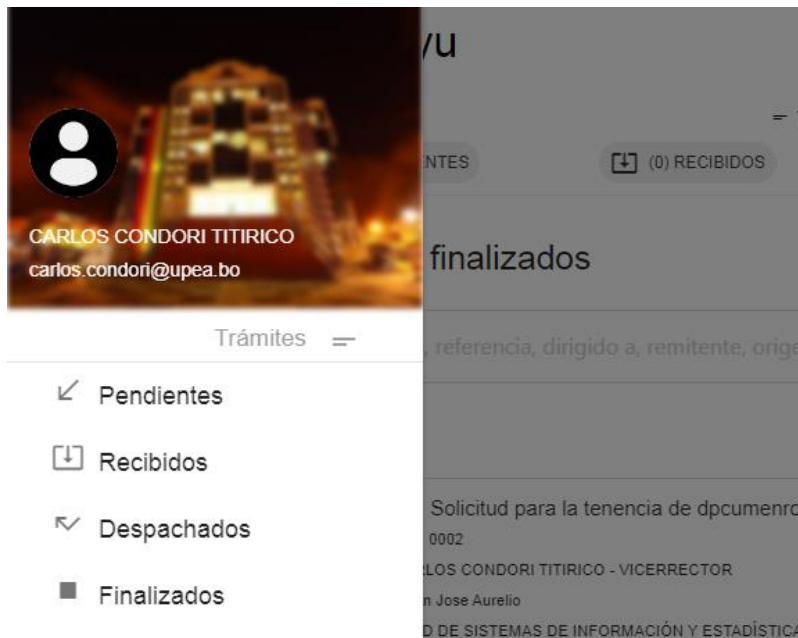


Figura 34. Acceso a los trámites desde la barra lateral de operaciones

Acceso a los trámites a nivel de unidad

Dependiendo de los permisos de tu cuenta de usuario podrás acceder a los trámites de toda tu unidad, también podrás marcar su recepción, despacharlos y finalizarlos.

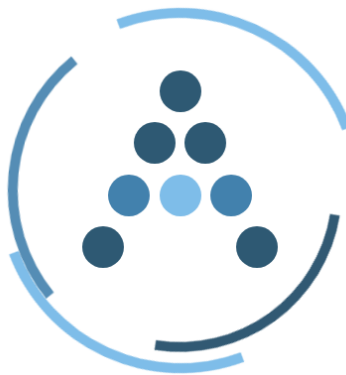


Universidad Pública de El Alto

Creada por Ley 2115 del 5 de septiembre de 2000 y Autónoma por Ley 2556 del 12 de noviembre de 2003



UNIDAD SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA



MANUAL TÉCNICO

Sistema de Seguimiento de Trámites “AJAYU”

AUTOR:

JHONNY APAZA CHAMBI

VERSIÓN:

1.0

FECHA:

24 DE NOVIEMBRE DE 2021

TABLA DE CONTENIDO

2.	PRINCIPALES REQUERIMIENTOS.....	
2.1.	Requerimientos Funcionales	
2.2.	Requerimientos no funcionales	
4.	DISEÑO	
4.1.	Diagramas de caso de uso	
5.	BASE DE DATOS	
5.1.	Diagrama de base de datos.....	
5.2.	Descripción de tablas	
4.	HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	
5.	REQUERIMIENTOS E INSTALACIÓN	
5.1.	Requerimientos	
5.2.	Instalación	

INDICE DE FIGURAS

Figura 1:	Diagrama de caso de uso general	
Figura 2:	Administración del sistema	
Figura 3:	Administración de unidades	
Figura 4:	Administración de usuarios	
Figura 5:	Módulo de trámites	
Figura 6:	Módulo de notificaciones	
Figura 7:	Cuenta de usuario	
Figura 8:	Diagrama de base de datos	

INDICE DE TABLAS

Tabla 1: Administración del sistema
Tabla 2: Administración de unidades
Tabla 3: Administración de usuarios
Tabla 4: Módulo de trámites
Tabla 5: Módulo de notificaciones
Tabla 6: Cuenta de usuario
Tabla 7: Descripción tabla “permisos”
Tabla 8: Descripción tabla “asignación_permisos”
Tabla 9: Descripción tabla “roles”
Tabla 10: Descripción tabla “usuarios”
Tabla 11: Descripción tabla “usuario_recuperacion”
Tabla 12: Descripción tabla “notificaciones”
Tabla 13: Descripción tabla “hoja_ruta”
Tabla 14: Descripción tabla “tramites”
Tabla 15: Descripción tabla “hoja_ruta_seguimiento”
Tabla 16: Descripción tabla “hoja_ruta_numero_interno”
Tabla 17: Descripción tabla “hoja_ruta_numero_interno”
Tabla 18: Descripción tabla “tramite_categoria”
Tabla 19: Descripción tabla “gestion”
Tabla 20: Descripción tabla “hoja_ruta_instrucción”

PRINCIPALES REQUERIMIENTOS

1.1. Requerimientos funcionales

A continuación se muestra el listado de requerimientos funcionales del sistema y su descripción:

- ADMINISTRACIÓN DE USUARIOS.- Administrar los usuarios del sistema, asignarles roles y permisos, además de crear nuevos usuarios, modificarlos, asignarlos a una unidad y eliminarlos.
- ADMINISTRACIÓN DE UNIDADES.- Permite administrar las unidades del sistema, modificarlas y crear nuevas.
- RECUPERACIÓN DE CUENTA DE USUARIO.- Que permita a los usuarios cambiar su contraseña si no pueden acceder al sistema, utilizando su dirección de correo electrónico institucional para recibir un email con un enlace de recuperación.
- GESTIÓN DE TRÁMITES.- Que permite crear, recibir, despachar y finalizar trámites.
- SEGUIMIENTO DE TRÁMITES.- Que permita hacer seguimiento de trámites a cualquier persona que esté registrada dentro de la base de datos de la institución.
- CONSULTAS DE TRÁMITES.- Que permita generar reportes del módulo de trámites.
- NOTIFICACIONES.- Que permita a los usuarios dentro del sistema ser notificados en tiempo real dentro del sistema, cuando se les envíe un trámite

1.2. Requerimientos no funcionales

A continuación se muestra el listado de requerimientos no funcionales del sistema:

- El sistema debe visualizarse y funcionar correctamente en cualquier navegador como ser internet Explorer, Mozilla, Chrome, etc.
- Respaldo energético del servidor, para asegurar la disponibilidad del sistema.
- Soporte y mantenimiento periódico para asegurar el buen rendimiento del sistema.

2. DISEÑO

2.1. Diagramas de caso de uso

2.1.1. Diagrama de caso de uso general

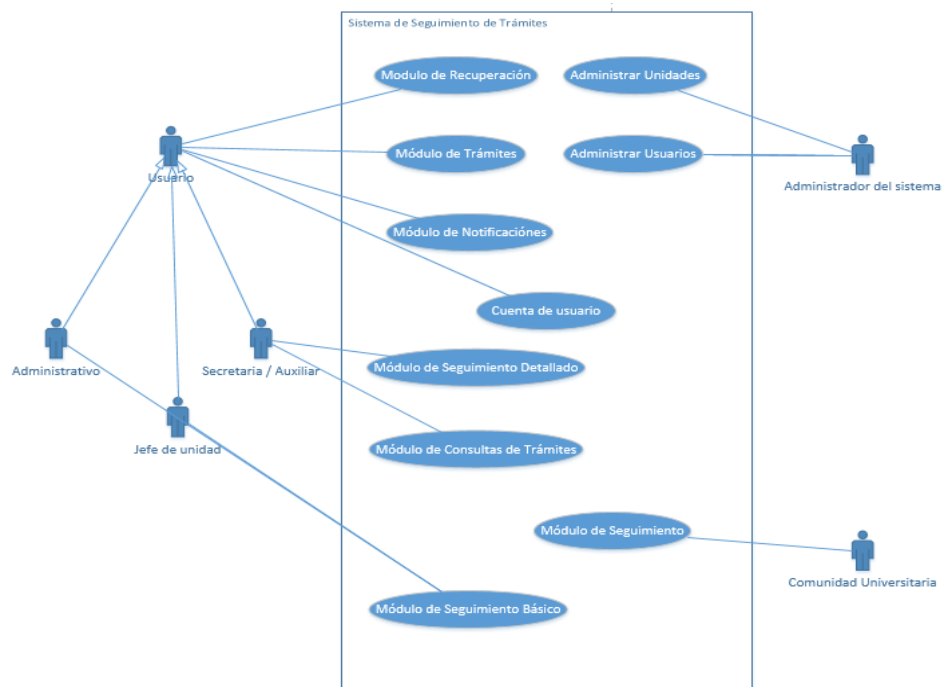


Figura 1: Diagrama de caso de uso general

2.1.2. Administración del Sistema

Se muestra el modelo donde se puede apreciar el comportamiento del actor súper administrador sobre la manipulación del sistema: Administración de usuarios, almacén, compras, ventas, caja y reportes y ver las diferentes acciones que realiza.

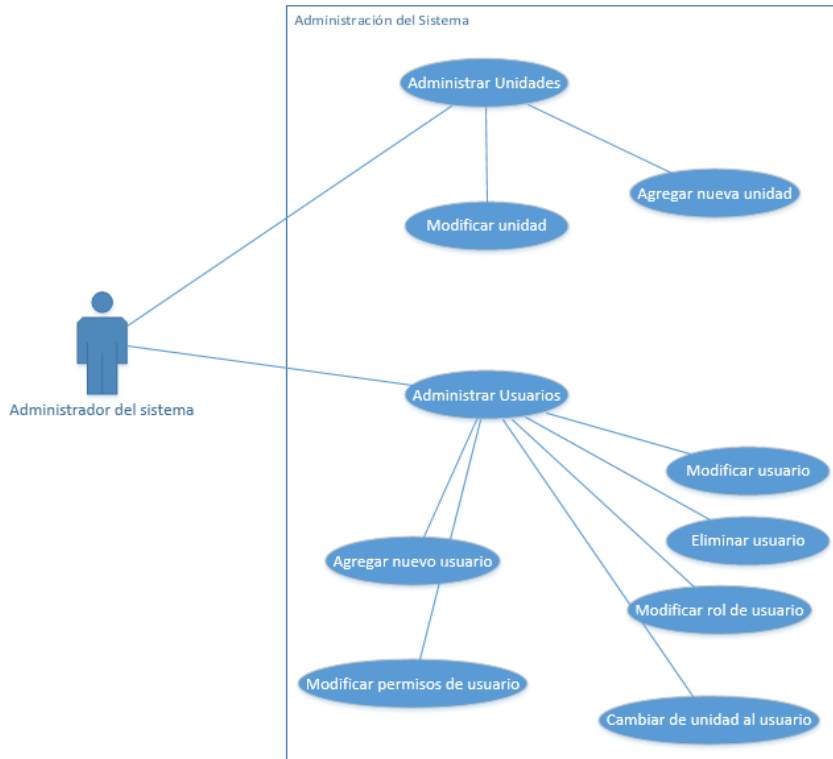


Figura 2: Administración del sistema

CASO DE USO ADMINISTRACIÓN DEL SISTEMA	
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	<p>Este usuario puede realizar las siguientes acciones:</p> <ul style="list-style-type: none"> Módulo de administración de unidades: Modificar y crear unidades. Módulo de administración de usuarios: Modificar, crear, eliminar, cambiar rol, cambiar permisos y cambiar de unidad a los usuarios

Tabla 1: Administración del sistema

2.1.3. Administración de Unidades

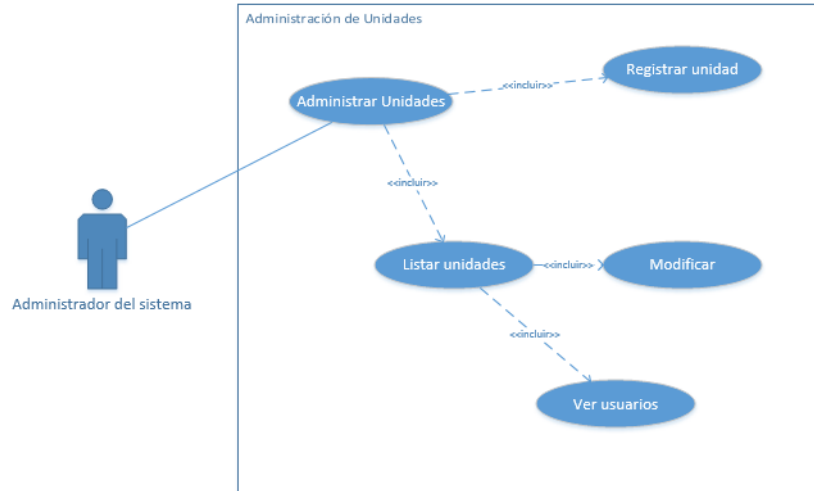


Figura 3: Administración de unidades

CASO DE USO ADMINISTRACIÓN DE UNIDADES	
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	El administrador del sistema registra unidades, lista las unidades existentes, modifica la unidad y puede listar los usuarios de la unidad

Tabla 2: Administración de unidades

2.1.4. Administración de usuarios

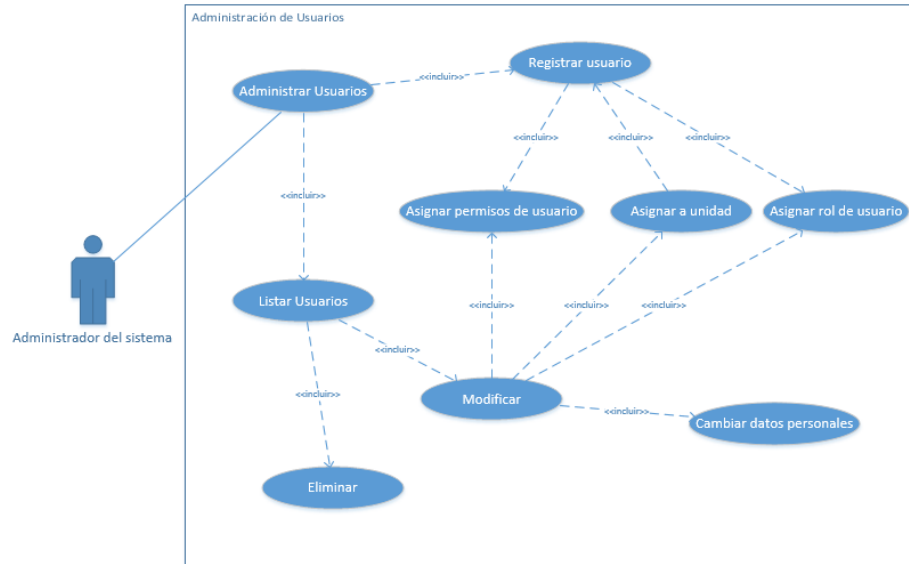


Figura 4: Administración de usuarios

CASO DE USO ADMINISTRACIÓN DE USUARIOS	
ACTOR	Administrador, Encargado de Sistemas
TIPO	Primario
DESCRIPCIÓN	El administrador del sistema registra usuarios, lista los usuarios existentes, modificar datos personales, asignar permisos, asignar rol, asignar unidad y eliminar usuarios

Tabla 3: Administración de usuarios

2.1.5. Módulo de trámites

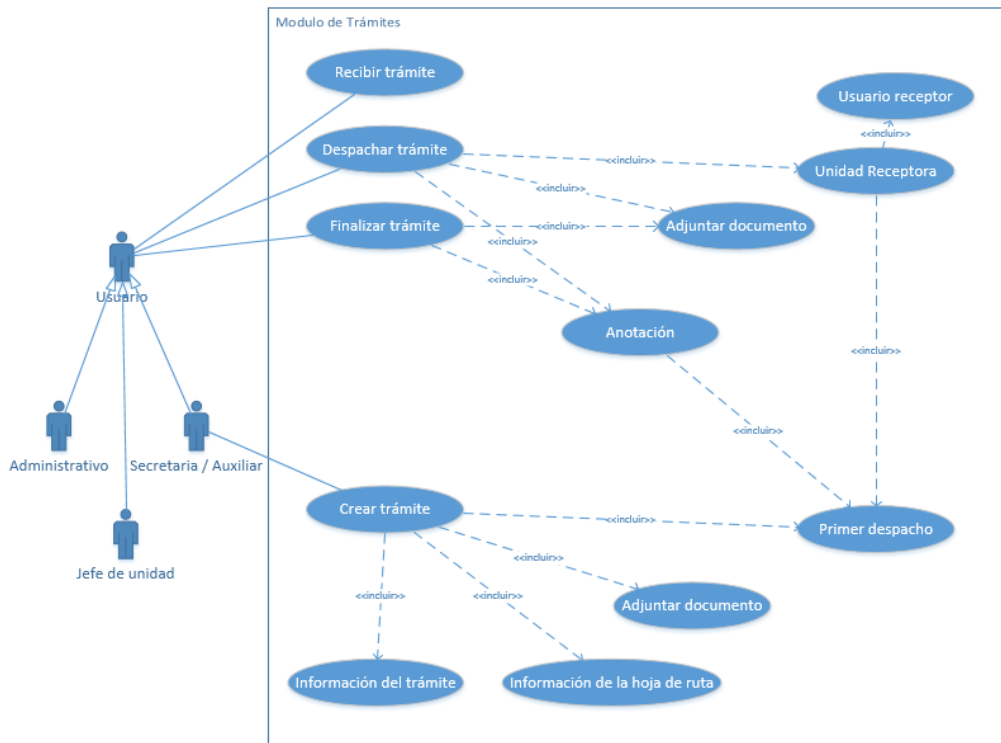


Figura 5: Módulo de trámites

CASO DE USO		MÓDULO DE TRÁMITES	
ACTOR	Usuarios dentro del sistema		
TIPO	Primario		
DESCRIPCIÓN	<p>Todos los usuarios pueden:</p> <ul style="list-style-type: none"> • Recibir trámites que otros usuarios les hayan enviado • Despachar trámites a otros usuarios del sistema • Finalizar trámites <p>Los usuarios con rol “Secretaria/Auxiliar” pueden:</p> <ul style="list-style-type: none"> • Crear trámites 		

Tabla 4: Módulo de trámites

2.1.6. Módulo de notificaciones

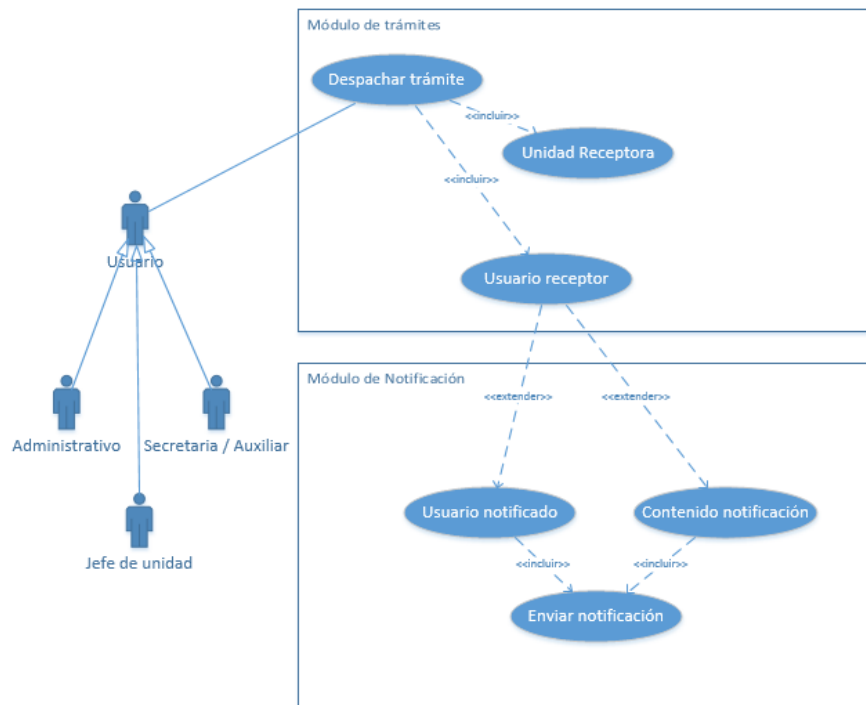


Figura 6: Módulo de notificaciones

CASO DE USO	MÓDULO DE NOTIFICACIÓN
ACTOR	Usuarios dentro del sistema
TIPO	Primario
DESCRIPCIÓN	Todos los usuarios pueden recibir notificaciones automáticamente, cuando otro usuario les despacha un trámite, esto incluye el primer despacho de un trámite

Tabla 5: Módulo de notificaciones

2.1.7. Cuenta de usuario

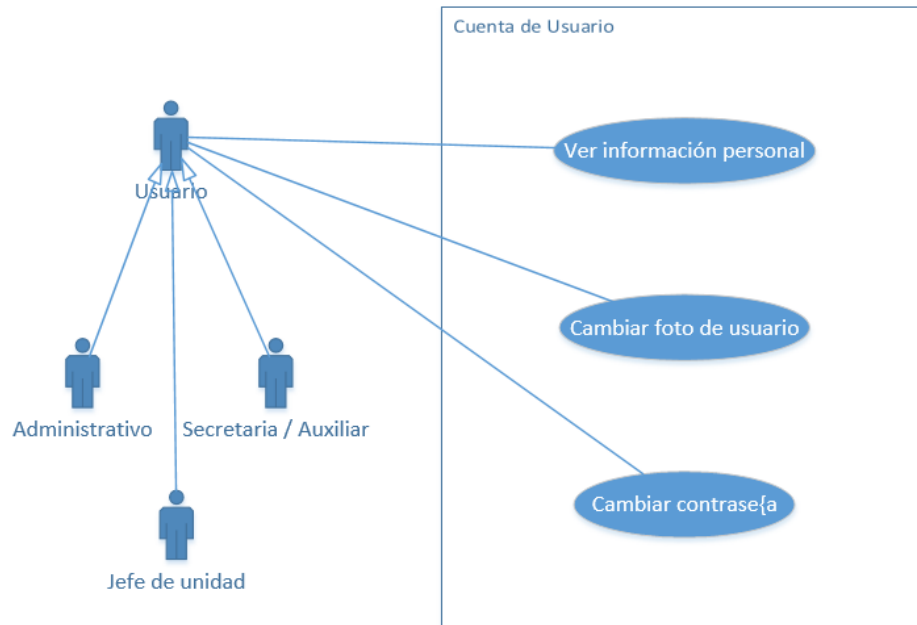


Figura 7: Cuenta de usuario

CASO DE USO		CUENTA DE USUARIO	
ACTOR	Usuarios dentro del sistema		
TIPO	Secundario		
DESCRIPCIÓN	Los usuarios pueden ver su información personal, cambiar su contraseña y cambiar su foto de usuario		

Tabla 6: Cuenta de usuario

3. BASE DE DATOS

3.1. Diagrama de base de datos

A continuación se muestra el diagrama de la base de datos del sistema:

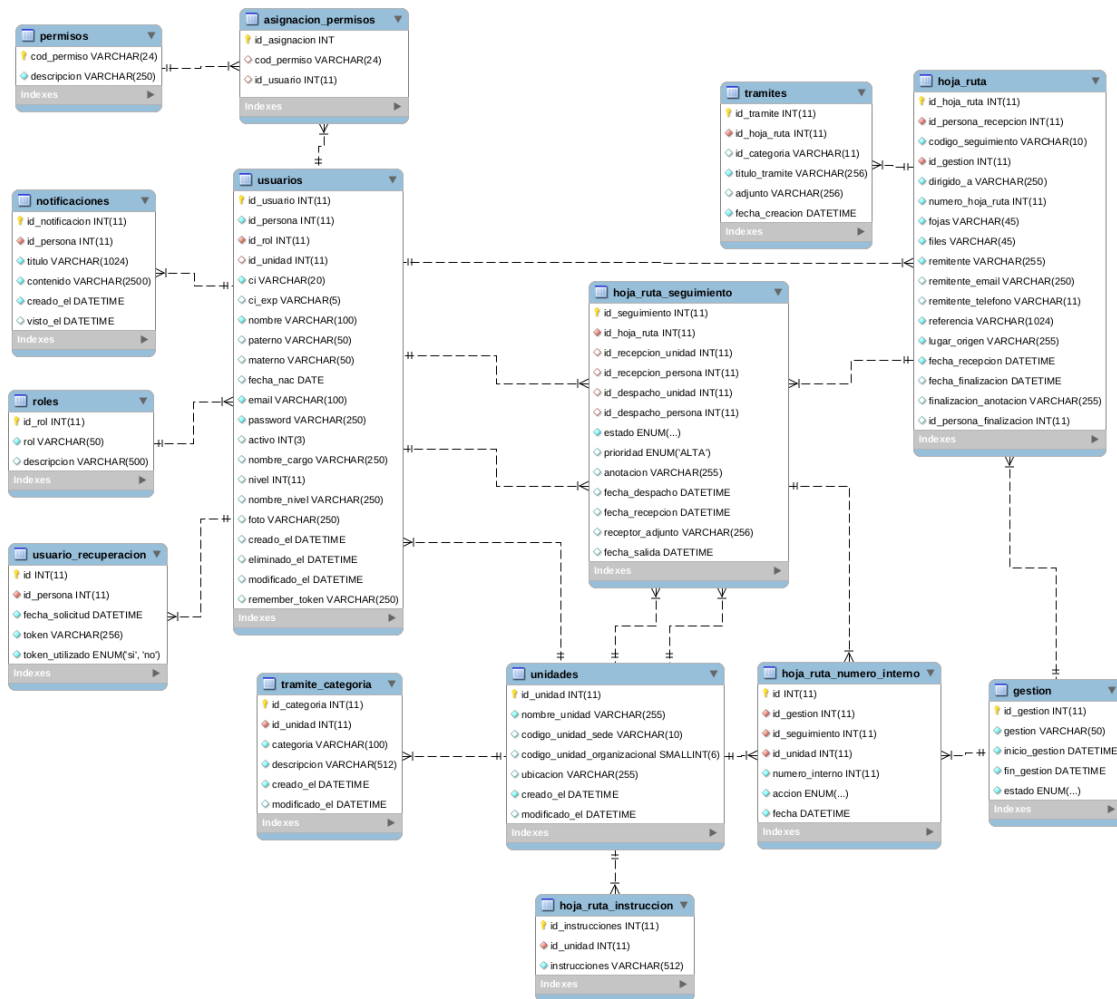


Figura 8: Diagrama de base de datos

3.2. Descripción de tablas

A continuación se muestra la descripción y utilización de cada uno de los campos de las tablas de la base de datos:

- **Tabla “permisos”**

Descripción General	Descripción de columnas
<p>Esta tabla está destinada a almacenar los permisos que se manejaran en el sistema, y está aislada de las demás tablas ya que el sistema solo la utilizará para validación de permisos.</p>	<ul style="list-style-type: none"> • “cod_permiso” VARCHAR(24).- En este campo se almacenarán los códigos de cada permiso del sistema. • “descripcion” VARCHAR(250).- Almacenara la descripción de cada permiso.

Tabla 7: Descripción tabla “permisos”

- **Tabla “asignación_permisos”**

Descripción General	Descripción de columnas
<p>Almacena la asignación de permisos de cada usuario del sistema.</p>	<ul style="list-style-type: none"> • “id_asignacion” INT(11).- Llave primaria autoincremental. • “cod_permiso” VARCHAR(24).- Llave foránea correspondiente a la tabla “permisos” en la columna “cod_permiso”. • “id_usuario” INT(11).- Llave foránea correspondiente a la tabla “usuarios” en la columna “id_usuario”.

Tabla 8: Descripción tabla “asignación_permisos”

- **Tabla “roles”**

Descripción General	Descripción de columnas

Almacenara los distintos roles que tendrán los usuarios del sistema, por ejemplo jefe de unidad, secretaria, etc.

- **“id_rol” INT(11).**- Llave primaria que almacena el id de cada rol de usuario
- **“rol” VARCHAR(50).**- Descripción corta del rol de usuario.
- **“descripcion” VARCHAR(500).**- Descripción extendida y detallada del rol de usuario.

Tabla 9: Descripción tabla “roles”

- **Tabla “usuarios”**

Descripción General	Descripción de columnas
<p>En esta tabla se registran los datos de los usuarios del sistema.</p>	<ul style="list-style-type: none"> • “id_usuario” INT(11).- Llave primaria autoincremental. • “id_persona” INT(11).- ID correspondiente a la persona dentro de la base de datos de la institución. • “id_rol” INT(11).- Llave foránea correspondiente a la tabla “roles” en la columna “id_rol”. • “id_unidad” INT(11).- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad”, mediante esta columna un usuario está asignado a una unidad. • “ci” VARCHAR(20).- Almacena el número de cedula de identidad del usuario. • “ci_exp” VARCHAR(5).- Almacena el expedido de la cedula de identidad del usuario.. • “nombre” VARCHAR(100).- Almacena el nombre o nombres del usuario.

- **“paterno” VARCHAR(50).**- Almacena el apellido paterno del usuario.
- **“materno” VARCHAR(50).**- Almacena el apellido materno del usuario.
- **“fecha_nac” DATE.**- Almacena la fecha de nacimiento del usuario.
- **“email” VARCHAR(100).**- Almacena la dirección de correo electrónico institucional del usuario.
- **“password” VARCHAR(250).**- Almacena el hash del password del usuario.
- **“activo” INT(3).**- Se almacenaran valores de 0 o 1, representando que el usuario activo o no su cuenta cambiando su contraseña por defecto.
- **“nombre_cargo” VARCHAR(250).**- Almacena el nombre del cargo que el usuario ocupa en la institución.
- **“nivel” INT(11).**- Almacena el nivel del administrativo del usuario, significando los valores menores cargos de mayor jerarquía.
- **“nombre_nivel” VARCHAR(250).**- Almacena el nombre del nivel del usuario.
- **“foto” VARCHAR(250).**- Opcionalmente nulo almacena el nombre y la extensión de la foto de un usuario.
- **“creado_el” DATETIME.**- Almacena en formato de fecha y tiempo la creación del usuario.
- **“eliminado_el” DATETIME.**- Por defecto nulo, almacena la hora y fecha de eliminación del usuario, de este modo se maneja el eliminado lógico de usuarios.

- **“modificado_el” DATETIME.-** Almacena la fecha y hora de la última modificación de los datos del usuario.
- **“remember_token” VARCHAR(250).-** Almacena opcionalmente el token para la función “Permanecer conectado” del login del sistema.

Tabla 10: Descripción tabla “usuarios”

- **Tabla “usuario_recuperacion”**

Descripción General	Descripción de columnas
<p>Almacena las solicitudes de recuperación de contraseñas hechas por los usuarios.</p>	<ul style="list-style-type: none"> • “id” INT(11).- Llave primaria autoincremental. • “id_persona” INT(11).- Llave foránea correspondiente a la tabla “usuarios” en la columna “cod_id_persona”. • “fecha_solicitud” DATETIME.- Almacena la fecha y hora en la que el usuario solicita recuperar su cuenta. • “token” VARCHAR(256).- Almacena el token que se envía por correo electrónico para el cambio de contraseña. • “token_utilizado” ENUM(‘si’, ‘no’).- Por defecto “no”, indica si el token fue utilizado.

Tabla 11: Descripción tabla “usuario_recuperacion”

- **Tabla “notificaciones”**

Descripción General	Descripción de columnas
---------------------	-------------------------

Almacena las notificaciones de los usuarios dentro del sistema.

- **“id_notificacion” INT(11).**- Llave primaria autoincremental.
- **“id_persona” INT(11).**- Llave foránea correspondiente a la tabla “usuarios” en la columna “id_persona”.
- **“titulo” VARCHAR(1024).**- Almacena el título de la notificación.
- **“contenido” VARCHAR(2500).**- Almacena el contenido de la notificación puede tener HTML incluido.
- **“creado_el” DATETIME.**- Almacena la fecha y hora del envío de la notificación.
- **“visto_el” DATETIME.**- Por defecto NULO almacena la fecha y hora en la que la notificación fue vista.

Tabla 12: Descripción tabla “notificaciones”

- **Tabla “hoja_ruta”**

Descripción General	Descripción de columnas
Almacena los datos de las hojas de ruta.	<ul style="list-style-type: none"> • “id_hoja_ruta” INT(11).- Llave primaria autoincremental. • “id_persona_recepcion” INT(11).- Llave foránea correspondiente a la tabla “usuarios” en la columna “id_persona”. • “código_seguimiento” VARCHAR(10).- Almacena el código de seguimiento de los trámites. • “dirigido_a” VARCHAR(250).- Almacena el nombre de la persona a la que va dirigida la hoja de ruta.

- **“numero_hoja_ruta” INT(11).**- Almacena el número de hoja de ruta, que re-inicia cada gestión.
- **“fojas” VARCHAR(45).**- Almacena el número de hojas adjuntado a la hoja de ruta.
- **“files” VARCHAR(45).**- Almacena el número de files del trámite.
- **“remitente” VARCHAR(255).**- Almacena el nombre del remitente.
- **“remitente_email” VARCHAR(250).**- Almacena opcionalmente la dirección de correo electrónico del remitente.
- **“remitente_telefono” VARCHAR(11).**- Almacena opcionalmente el número de teléfono del remitente.
- **“referencia” VARCHAR(1024).**- Almacena la referencia de la hoja de ruta.
- **“lugar_origen” VARCHAR(255).**- Almacena el lugar de origen.
- **“fecha_recepcion” DATETIME.**- Almacena la fecha y hora de la recepción del trámite en el sistema.
- **“fecha_finalizacion” DATETIME.**- Por defecto NULO almacena la fecha y hora de finalización del trámite.
- **“finalización_annotacion” VARCHAR(255).**- Por defecto NULO almacena la anotación de la finalización del trámite.
- **“id_persona_finalizacion” INT(11).**- Por defecto NULO almacena el id de la persona que finalizó el trámite.

Tabla 13: Descripción tabla “hoja_ruta”

- **Tabla “tramites”**

Descripción General	Descripción de columnas
<p>Almacena datos sobre los trámites.</p>	<ul style="list-style-type: none"> • “id_tramite” INT(11).- Llave primaria autoincremental. • “id_hoja_ruta” INT(11).- Llave foránea correspondiente a la tabla “hoja_ruta” en la columna “id_hoja_ruta”. • “id_categoria” VARCHAR(11).- Por defecto NULO almacena el id de la categoría del trámite. • “referencia_tramite” VARCHAR(256).- Almacena la referencia del trámite. • “adjunto” VARCHAR(256).- Almacena el nombre y extensión del archivo digitalizado del trámite. • “fecha_creación” DATETIME.- Almacena la fecha y hora de creación del trámite.

Tabla 14: Descripción tabla “tramites”

- **Tabla “hoja_ruta_seguimiento”**

Descripción General	Descripción de columnas
<p>Almacena el recorrido de una hoja de ruta.</p>	<ul style="list-style-type: none"> • “id_seguimiento” INT(11).- Llave primaria autoincremental. • “id_hoja_ruta” INT(11).- Llave foránea correspondiente a la tabla “hoja_ruta” en la columna “id_hoja_ruta”. • “id_recepcion_unidad” INT(11).- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad” que

almacena el id de unidad que recibirá el trámite.

- **“id_recepcion_persona” INT(11).**- Llave foránea correspondiente a la tabla “usuarios” en la columna “id_persona” que almacena el id de la persona que recibirá el trámite.
- **“id_despacho_unidad” INT(11).**- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad” que almacena el id de la unidad que despacha el trámite.
- **“id_despacho_persona” INT(11).**- Llave foránea correspondiente a la tabla “usuarios” en la columna “id_persona” que almacena el id de la unidad que despacha el trámite.
- **“estado” ENUM(‘recibido’, ‘despachado’, ‘pendiente’).**- Almacena el estado de cada registro.
- **“prioridad” ENUM(‘alta’).**- Por defecto NULO almacena si el envío es de alta prioridad.
- **“anotacion” VARCHAR(255).**- Opcionalmente almacena una anotación al enviarse un trámite.
- **“fecha_despacho” DATETIME.**- Almacena la fecha y hora de cuando el trámite fue despachado al usuario de la columna “id_recepcion_persona”.
- **“fecha_recepcion” DATETIME.**- Almacena la fecha y hora de cuando el trámite es marcado como recibido por el usuario.
- **“recepción_adjunto” VARCHAR(255).**- Almacena opcionalmente el nombre y

extensión de un archivo adjunto cuando un usuario despacha un trámite.

- **“fecha_salida” DATETIME.-** Almacena la fecha y hora en la que el usuario despacha el trámite a otro usuario.

Tabla 15: Descripción tabla “hoja_ruta_seguimiento”

- **Tabla “hoja_ruta_numero_interno”**

Descripción General	Descripción de columnas
<p>Almacena la entrada y salida de trámites para cada unidad.</p>	<ul style="list-style-type: none"> • “id” INT(11).- Llave primaria autoincremental. • “id_gestion” INT(11).- Llave foránea correspondiente a la tabla “gestion” en la columna “id_gestion”. • “id_seguimiento” INT(11).- Llave foránea correspondiente a la tabla “hoja_ruta_seguimiento” en la columna “id_seguimiento”. • “id_unidad” INT(11).- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad”. • “numero_interno” INT(11).- Almacena el numero correlativo de entrada o salida que se reinicia cada gestión. • “accion” ENUM(‘entrada’, ‘salida’).- Almacena si el trámite entro o salio de la unidad. • “fecha” DATETIME.- Almacena la fecha y hora de entrada o salida del trámite.

Tabla 16: Descripción tabla “hoja_ruta_numero_interno”

- **Tabla “unidades”**

Descripción General	Descripción de columnas
<p>Almacena las unidades/carreras/decanatos existentes dentro de la institución.</p>	<ul style="list-style-type: none"> • “id_unidad” INT(11).- Llave primaria auto-incremental. • “nombre_unidad” VARCHAR(255).- Almacena el nombre de la unidad. • “código_unidad_sede” VARCHAR(10).- Almacena el código asignado a cada unidad dentro de la institución. • “código_unidad_organizacional” SMALLINT(6).- Almacena el código de unidad organizacional de la unidad. • “ubicacion” VARCHAR(255).- Almacena la ubicación física de la unidad. • “creado_el” DATETIME.- Almacena la fecha y hora de la creación del registro. • “modificado_el” DATETIME.- Almacena la fecha y hora de la ultima modificación del registro

Tabla 17: Descripción tabla “hoja_ruta_numero_interno”

- **Tabla “tramite_categoria”**

Descripción General	Descripción de columnas
<p>Almacena las categorías de trámites de cada unidad.</p>	<ul style="list-style-type: none"> • “id_categoria” INT(11).- Llave primaria auto-incremental. • “id_unidad” INT(11).- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad”.

- **“categoria” VARCHAR(100).**- Almacena la categoría.
- **“descripcion” VARCHAR(512).**- Almacena la descripción de la categoría.
- **“creado_el” DATETIME.**- Almacena la fecha y hora de la creación del registro.
- **“modificado_el” DATETIME.**- Almacena la fecha y hora de la última modificación del registro

Tabla 18: Descripción tabla “tramite_categoria”

- **Tabla “gestion”**

Descripción General	Descripción de columnas
Almacena las gestiones en las que se manejarán los trámites.	<ul style="list-style-type: none"> • “id_gestion” INT(11).- Llave primaria auto-incremental. • “gestion” VARCHAR(50).- Almacena la gestión. • “inicio_gestion” DATETIME.- Almacena la fecha y hora del inicio de la gestión. • “fin_gestion” DATETIME.- Almacena la fecha y hora del fin de la gestión. • “estado” ENUM(‘finalizado’, ‘actual’).- Almacena el estado de la gestión.

Tabla 19: Descripción tabla “gestion”

- **Tabla “hoja_ruta_instruccion”**

Descripción General	Descripción de columnas
---------------------	-------------------------

Almacena las instrucciones que se mostrarán en la hoja de ruta de cada unidad

- **“id_instrucciones” INT(11).**- Llave primaria auto-incremental.
- **“id_unidad” INT(1).**- Llave foránea correspondiente a la tabla “unidades” en la columna “id_unidad”.
- **“instrucciones” VARCHAR(12).**- Almacena las instrucciones en sí.

Tabla 20: Descripción tabla “hoja_ruta_instrucción”

4. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

4.1. PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (The PHP Group, 2018).

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

4.2. MYSQL

Es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para crear base de datos con acceso desde páginas webs dinámicas, para la creación de sistemas de transacciones on-line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas. (Cobo, Gómez, Pérez, & Rocha, 2005)

4.3. APACHE

Es un servidor web HTTP de código abierto para plataformas Unixlike (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. (GDR Soluciones, 2017).

4.4. LARAVEL

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. (Andrés, 2014).

4.5. JAVASCRIPT

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

5. REQUERIMIENTOS E INSTALACIÓN

5.1. Requerimientos

A continuación, se lista los requerimientos del sistema para la implementación:

- PHP en la versión 7.2 a la 8.0 compatibles con la versión 5.8 de Laravel que se utiliza.
- Servidor Apache instalado.
- Composer.
- Acceso SSH para ejecución de comandos.
- Puerto 6001 libre para el servidor de websockets

El sistema no es exigente en cuanto a recursos de hardware, así que podría funcionar en un hosting compartido o un servidor virtual privado con las siguientes especificaciones:

- 512 Mb de memoria ram.
- 20 GB de almacenamiento.
- 1 núcleo de procesador.

5.2. Instalación

Para instalar el sistema en un servidor, se deben seguir los siguientes pasos:

- Copiar el contenido del directorio del sistema al directorio final donde se alojará el sistema.
- Crear una nueva base de datos e importar el script sql del sistema.
- Configurar el archivo “.env” con las credenciales de base de datos y email.
- Configurar el dominio del sistema.

- Iniciar el servidor local de websockets con el comando “php artisan websockets:serve”.
- Agregar en la configuración del dominio las siguientes líneas para redireccionar la conexión websockets del puerto 6001 al servidor local de websockets:

```
<VirtualHost *:80>
...
RewriteEngine on
ProxyRequests off
ProxyVia on
RewriteCond %{HTTP:Connection} Upgrade [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteRule /(.*) ws://localhost:6001/$1 [P,L]
ProxyPass      /request/path http://localhost:6001/request/path
ProxyPassReverse /request/path http://localhost:6001/request/path
</VirtualHost>
```

- Verificar el funcionamiento ingresando en el navegador el dominio asignado al sistema.

El Alto, noviembre de 2021

Señor(a):

Ing. Maricel Yarari Mamani

DOCENTE DE LA MATERIA DE TALLER DE LICENCIATURA II

Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: "**SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO**" que propone el postulante Univ.: **Jhonny Apaza Chambi**, con C.i.: **6878067 L.P.** y **R.U.:15000074**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.



Ing. Sergio Ramiro Rojas Saire

Tutor Revisor

El Alto, noviembre de 2021

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE LA CARRERA – INGENIERÍA DE SISTEMAS


Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: "**SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO**" que propone el postulante Univ.: **Jhonny Apaza Chambi**, con C.i.: **6878067 L.P.** y R.U.:**15000074**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.



Ing. Maricel Yarari Mamani
Tutor Metodológico

El Alto, noviembre de 2021

Señor(a):

Ing. Maricel Yarari Mamani

DOCENTE DE LA MATERIA DE TALLER DE LICENCIATURA II

Presente:

REF.: AVAL DE CONFORMIDAD

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con el Proyecto de Grado denominado: "**SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO**" que propone el postulante Univ.: **Jhonny Apaza Chambi**, con C.i.: **6878067 L.P.** y R.U.:**15000074**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.



Ing. ~~Francisco Luis Páez~~ Rocha

Tutor Especialista

C I 2296105 11^o

El Alto, noviembre de 2021

Señor(a):

Ing. Maricel Yarari Mamani

Docente de la materia de Taller de Licenciatura II

Presente

REF. Aval de Conformidad de implementación del Trabajo de Grado

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, mi conformidad con la implementación del Trabajo de Grado denominado: **"SISTEMA WEB PARA EL SEGUIMIENTO DE TRÁMITES – CASO: UNIVERSIDAD PÚBLICA DE EL ALTO"** que propone el postulante **Univ.: Jhonny Apaza Chambi**, con **C.i.: 6878067 L.P.** y **R.U.:15000074**, para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente:



Ing. Juan Regis Muñoz Sirpa

JEFE DE SISTEMAS DE INFORMACIÓN Y
ESTADÍSTICA - UPEA.