

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y
PROCESO DE PAGOS A PERSONAS CON DISCAPACIDAD”
CASO: UNIDAD DE ATENCIÓN PARA PERSONAS CON DISCAPACIDAD DEL
GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO (G.A.M.E.A.)**

**Para optar al título de Licenciatura en Ingeniería de Sistemas
Mención: Gestión y Producción**

**Postulante: Jose Luis Laura Mamani
Tutor Metodológico: M. Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Revisor: Lic. Beatriz Colque Condori
Tutor Especialista: M.Sc. Ing. Felix Reynaldo Pérez Romero**

EL ALTO – BOLIVIA

2024

DEDICATORIA

Dedico este trabajo a todas las personas que me apoyaron desde el momento de comenzar mi carrera, a mi madre, por brindarme su apoyo incondicional y confianza, cuidándome y dándome fortaleza para continuar mis estudios, a mi padre y a mis hermanas por motivarme y ayudarme siempre, a mi familia que me dieron esas ganas de seguir adelante.

AGRADECIMIENTOS

Agradecer primeramente a Dios, por iluminar mi camino y cuidarme durante el transcurso de mi carrera, por darme la fortaleza en los momentos que más lo necesite, y por otorgarme a mi familia y cuidarlos siempre.

A mi padre y madre por inculcarme los valores esenciales para la vida, que son un pilar importante en mi formación, también agradecer a mis hermanas por la motivación y apoyo brindado.

Agradecer a todo el plantel docente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto, por los conocimientos brindados a lo largo de mi formación.

A mis tutores Metodológico, Especialista y Revisor: M. Sc. Marisol Arguedas Balladares, M. Sc. Ing. Felix Reynaldo Perez Romero, Lic. Beatriz Colque Condori, quienes supieron guiarme con mucha paciencia, sugerencias y observaciones para el desarrollo y conclusión del presente proyecto de grado.

A mis amigos de carrera por su amistad, ayuda y apoyo moral durante la vida universitaria, muchas a gracias a todos por marcar mi vida de manera positiva.

RESUMEN

En el proceso de informar y mejorar la atención a las personas con discapacidad en el municipio de El Alto, se analizó las diferentes situaciones y necesidades que deben pasar las personas que requieren este servicio, realizado en la dirección de desarrollo integral. La tecnología es un pilar fundamental en las entidades públicas, ya que les permite resolver problemas que se presenten por el manejo inadecuado de la información. Con este fin el presente proyecto " SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y PROCESO DE PAGOS A PERSONAS CON DISCAPACIDAD " tiene como finalidad automatizar el manejo de la información que con el tiempo crece y cada vez es más complicada su manejo, facilitando con ello la obtención de datos para consultas, datos estadísticos, reportes y pagos, Para su implementación se utilizó la metodología ágil SCRUM, misma que nos guiará desde la recopilación de los requerimientos, planificación de tareas, las fechas de cada tarea y los responsables para su desarrollo. Dado que es un sistema de información se utilizó herramientas útiles en este tipo de sistemas, como el framework Codeigniter basado en PHP y Javascript, el gestor de base de datos Mysql y el servidor Apache. Para obtener la calidad de software del proyecto se basó en la norma ISO 25010 analizando la funcionabilidad, fiabilidad, usabilidad y mantenibilidad. Para la seguridad se usó la norma ISO 27000 donde especifica las técnicas de seguridad del sistema para el ingreso autorizado.

Palabra clave: Control, Pagos, framework, Codeigniter

ÍNDICE

1 MARCO PRELIMINAR	1
1.1 INTRODUCCIÓN.....	2
1.2 ANTECEDENTES.....	3
1.2.1 Antecedentes institucionales	3
1.2.2 Antecedentes afines al proyecto de grado	4
1.2.2.1 Antecedentes Internacionales	4
1.2.2.2 Antecedentes Nacionales.....	6
1.3 PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1 Problema Principal.....	7
1.4 PROBLEMAS SECUNDARIOS	8
1.4.1 Formulación del Problema	8
1.5 OBJETIVOS	9
1.5.1 Objetivo General.....	9
1.5.2 Objetivos Específicos	9
1.6 JUSTIFICACIÓN.....	10
1.6.1 Justificación Técnica.....	10
1.6.2 Justificación Económica.....	11
1.6.3 Justificación Social	11
1.7 METODOLOGÍA.....	11
1.7.1 Método de desarrollo	11
1.7.1.1 Metodología SCRUM	12
1.7.2 Métricas de Calidad	12
1.7.2.1 Modelo de calidad del software ISO 25010	12
1.7.3 Costos	13
1.7.3.1 COCOMO II	13
1.7.4 Seguridad	13
1.7.4.1 Modelo de seguridad ISO 27002.....	13
1.7.5 Pruebas de software.....	13
1.8 MÉTODOS DE RECOLECCIÓN DE DATOS.....	14
1.8.1 Técnicas de investigación.....	14
1.9 HERRAMIENTAS	14
1.10 LÍMITES Y ALCANCES	15
1.10.1 Límites.....	15
1.10.2 Alcances.....	16
1.11 APORTES	17

2	MARCO TEÓRICO	18
2.1	INTRODUCCIÓN.....	19
2.2	SISTEMA.....	19
2.3	SISTEMA DE INFORMACION.....	19
2.3.1	Características de un Sistema de Información.....	20
2.4	SISTEMA WEB.....	20
2.5	CONTROL.....	20
2.6	PAGOS.....	21
2.6.1	Proceso de Pagos.....	21
2.6.2	Proceso de pagos a beneficios sociales.....	21
2.7	NORMA LEGAL EN EL SERVICIO DE PAGOS.....	22
2.7.1	Ley 977.....	23
2.8	INGENIERIA DE SOFTWARE.....	23
2.8.1	Actividades en el proceso de desarrollo de software.....	24
2.9	METODOLOGIA PARA EL DESARROLLO DE SOFTWARE.....	25
2.9.1	Metodologías Tradicionales.....	25
2.9.2	Metodologías Agiles.....	26
2.9.3	Diferencias entre metodologías Tradicionales y Agiles.....	26
2.10	METODOLOGIA SCRUM.....	27
2.11	HERRAMIENTAS.....	28
2.11.1	Herramientas para el desarrollo de software.....	28
2.11.2	Gestor de Base de Datos.....	32
2.12	METRICAS DE CALIDAD.....	33
2.12.1	Norma ISO 25010.....	33
2.13	METÓDO DE ESTIMACIÓN DE COSTOS.....	34
2.13.1	COCOMO II.....	34
2.13.2	Método de Estimación.....	35
2.14	SEGURIDAD DE LA INFORMACION.....	37
2.14.1	Norma ISO 27002.....	37
2.15	PRUEBAS DE SOFTWARE.....	38
3	MARCO APLICATIVO	39
3.1	INTRODUCCIÓN.....	40
3.1.1	Esquema del Funcionamiento del Sistema.....	40
3.2	ANÁLISIS DE LA SITUACIÓN ACTUAL.....	41
3.2.1	Descripción de la Institución.....	41
3.3	PRE-GAME.....	42

3.3.1	Product Backlog	42
3.3.2	Especificación.....	45
3.4	GAME.....	48
3.4.1	SPRINT 1: ACCESO AL SISTEMA.....	48
3.4.2	SPRINT 2: REALIZACIÓN DE PAGOS Y REGISTROS	51
3.4.3	SPRINT 3: REPORTE.....	57
3.4.4	SPRINT 4: DIGITALIZACIÓN Y CONTROL.....	59
3.4.5	SPRINT 5: CRUD DEL SISTEMA.....	63
3.4.6	SPRINT 6: CARGADO DE DATOS.....	67
3.4.7	SPRINT 7: REVISIÓN DE DATOS	70
3.4.8	SPRINT 8: ESTADÍSTICAS.....	73
3.5	POST-GAME	77
3.5.1	PRUEBAS	77
4	METRICAS DE CALIDAD Y SEGURIDAD	80
4.1	INTRODUCCIÓN	81
4.2	CALIDAD DE SOFTWARE	81
4.2.1	Funcionalidad	81
4.2.2	Fiabilidad	87
4.2.3	Usabilidad.....	87
4.2.4	Mantenibilidad	88
4.2.5	Portabilidad.....	89
4.3	SEGURIDAD DE LA INFORMACIÓN.....	90
4.3.1	ORGANIZACIÓN DE LA SEGURIDAD.....	90
4.3.2	CONTROL DE ACCESOS.....	91
4.3.3	Encriptación y Validación.....	92
4.3.4	Cierre de Sesiones	92
5	ANÁLISIS DE COSTO	93
5.1	INTRODUCCIÓN.....	94
5.2	CALCULO DE COSTOS COCOMO II.....	94
5.3	Análisis de Costos	94
5.4	Costo de Desarrollo de Software	95
6	CONCLUSIONES Y RECOMENDACIONES	97
6.1	INTRODUCCIÓN.....	98
6.2	CONCLUSIONES.....	98
6.3	RECOMEDACIONES	98
	REFERENCIAS BIBLIOGRÁFICAS.....	100

ÍNDICE DE TABLAS

Tabla 1 Metodologías tradicionales y metodologías ágiles.	27
Tabla 2 Descripción de las características de la ISO 25010	33
Tabla 3 Detalle de Coeficiente de COCOMO II.....	36
Tabla 4 Ecuaciones de Método de COCOMO II	36
Tabla 5 Actividades de usuarios	42
Tabla 6 Identificación de Actores.....	46
Tabla 7 Planificación del Sprint 1.....	48
Tabla 8 Historia de Usuario: Crear BD.....	50
Tabla 9 Historia de Usuario: Acceso al sistema.....	50
Tabla 10 Planificación del Sprint 2.....	51
Tabla 11 Historia de Usuario: Módulo de pagos	54
Tabla 12 Historia de Usuario: Módulo del Historial de pago.....	54
Tabla 13 Historia de Usuario: Módulo de Registro.....	55
Tabla 14 Planificación del Sprint 3.....	57
Tabla 15 Historia de Usuario: Reportes	59
Tabla 16 Planificación del Sprint 4.....	60
Tabla 17 Historia de Usuario: Digitalizar documentos.....	62
Tabla 18 Historia de Usuario: Observados	62
Tabla 19 Planificación del Sprint 5.....	64
Tabla 20 Historia de Usuario: CRUD Usuarios	66
Tabla 21 Historia de Usuario: CRUD Zonas	66
Tabla 22 Planificación del Sprint 6.....	67
Tabla 23 Historia de Usuario: Cargado de datos	69
Tabla 24 Planificación del Sprint 7.....	70
Tabla 25 Historia de Usuario: Verificación	72
Tabla 26 Planificación del Sprint 8.....	73
Tabla 27 Historia de Usuario: Panel Administrador.....	75
Tabla 28 Historia de Usuario: Panel Cajero	75
Tabla 29 Historia de Usuario: Estadísticas	76
Tabla 30 Pruebas del software	77
Tabla 31 Lista de entradas del usuario al sistema	81
Tabla 32 Lista de salidas de usuario del sistema.....	82
Tabla 33 Lista de peticiones del usuario al sistema	82
Tabla 34 Lista de archivos Lógicos del sistema	83
Tabla 35 Factor de ponderación de la funcionabilidad	83
Tabla 36 Valores de puntos función	84
Tabla 37 Cálculo del Factor de Complejidad	84
Tabla 38 Evaluación de usabilidad	88
Tabla 39 Evaluación de Mantenibilidad	88
Tabla 40 Rangos para evaluar la portabilidad.....	89
Tabla 41 Valoración para la Portabilidad	89
Tabla 42 Conversión de Puntos de Función a KDLC.....	95
Tabla 43 Tabla comparativa con el sistema implementado.....	99

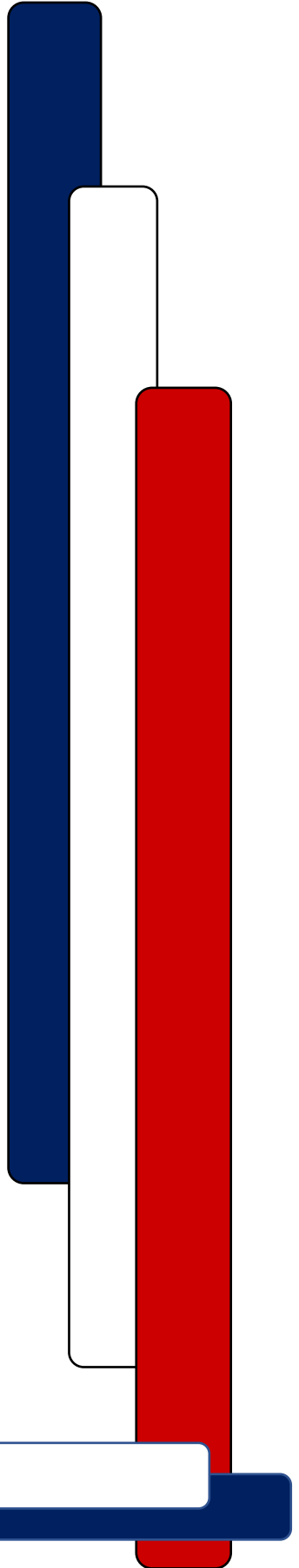
INDICE DE FIGURAS

Figura 1 Organigrama Órgano Ejecutivo del GAMEA.....	4
Figura 2 Fases del modelo en cascada.....	26
Figura 3 Fases del Sprint	28
Figura 4 Esquema del Flujo del sistema.....	40
Figura 5 Diagrama de flujo de datos pago a personas con discapacidad	41
Figura 6 Listado de Sprints planificados.....	45
Figura 7 Diagrama de Casos de Uso del sistema.....	46
Figura 8 Modelo de Base de Datos del sistema	47
Figura 9 Diagrama casos de uso acceso al sistema.....	48
Figura 10 Diagrama de secuencia acceso al sistema.....	49
Figura 11 Sprint-Backlog Acceso al sistema	49
Figura 12 Interfaz gráfica Sprint 1	51
Figura 13 Diagrama casos de uso realización de pagos y registros	52
Figura 14 Diagrama de secuencia registro de beneficiario	52
Figura 15 Diagrama de secuencia realización de pagos	53
Figura 16 Sprint-Backlog Pagos y Registros	53
Figura 17 Interfaz gráfica Sprint 2 Tarea 2	55
Figura 18 Interfaz gráfica Sprint 2 Tarea 6	56
Figura 19 Interfaz gráfica Sprint 2 Tarea 7	56
Figura 20 Diagrama casos de uso reportes.....	57
Figura 21 Diagrama de secuencia reportes.....	58
Figura 22 Sprint-Backlog Reportes	58
Figura 23 Interfaz gráfica Sprint 3	59
Figura 24 Diagrama casos de uso digitalización y control	60
Figura 25 Diagrama de secuencia digitalización	61
Figura 26 Sprint-Backlog Digitalización e inhabilitación.....	61
Figura 27 Interfaz gráfica Sprint 4 Tarea 2	63
Figura 28 Interfaz gráfica Sprint 4 Tarea 4	63
Figura 29 Diagrama casos de uso CRUD de datos	64
Figura 30 Diagrama de secuencia CRUD de datos	65
Figura 31 Sprint-Backlog CRUD de datos	65
Figura 32 Interfaz gráfica Sprint 5 Tarea 2	66
Figura 33 Interfaz gráfica Sprint 5 Tarea 3	67
Figura 34 Diagrama casos de uso cargado de datos	68
Figura 35 Diagrama de secuencia Cargado de datos.....	68
Figura 36 Sprint-Backlog Cargado de datos.....	69
Figura 37 Interfaz gráfica Sprint 6	70
Figura 38 Diagrama casos de uso acceso consulta de información	71
Figura 39 Diagrama de secuencia Consulta de información.....	71
Figura 40 Sprint-Backlog Consulta de información.....	72
Figura 41 Interfaz gráfica Sprint 7	73
Figura 42 Diagrama casos de uso Estadísticas.....	74
Figura 43 Diagrama de secuencia Estadísticas.....	74
Figura 44 Sprint-Backlog Estadísticas.....	75
Figura 45 Interfaz gráfica Sprint 8	76
Figura 46 Prueba de carga y estrés con JMeter.....	79
Figura 47 Prueba de seguridad en el envío de peticiones	79
Figura 48 Autenticación de usuario	91
Figura 49 Encriptación de contraseña	92

CAPITULO I

1 MARCO PRELIMINAR

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO



1.1 INTRODUCCIÓN

Actualmente la tecnología de información se convirtió en un medio fundamental para nuestra sociedad ya que nos ayuda de gran manera a la gestión y manejo de la información, debido al aumento de la población varias instituciones públicas o privadas requieren este tipo de tecnologías que mejoran el entorno de trabajo y mejora las relaciones con el público, además ayuda a optimizar procesos de administración con herramientas informáticas accesibles desde cualquier navegador, sea a través de internet o a través de una red local.

El Ministerio de trabajo a través de la plataforma “Eustaquio Moto Méndez” realiza la habilitación de la persona con discapacidad para realizar el cobro del beneficio, antes se toma en cuenta la calificación de discapacidad realizada por el Ministerio de salud, después se envía el listado a cada municipio del País. El Gobierno Autónomo Municipal de El Alto (G.A.M.E.A.) es la encargada de realizar el pago de discapacidad en el municipio de El Alto. La unidad de atención a personas con discapacidad realiza los registros de datos y el pago del beneficio.

Actualmente, se presenta inconvenientes con la atención al público, las personas que no tienen acceso a información de forma sencilla, además que la unidad solicitante requiere de funciones que le permita una mejor administración de sus registros.

Con el presente proyecto se tiene como finalidad desarrollar e implementar un sistema de control de la información y de los procesos de pagos a personas con discapacidad, que brinde un mejor servicio a la población facilitando la atención, tramites, registros y reportes.

El sistema será desarrollado en el lenguaje de programación PHP con el uso del framework Codeigniter 3, con la base de datos MySQL y usando la metodología de desarrollo SCRUM.

1.2 ANTECEDENTES

1.2.1 Antecedentes institucionales

El Gobierno Autónomo Municipal de El Alto es el nombre oficial de la instancia de GAMEA, que administra el territorio comprendido por el municipio. La entidad inicio su funcionamiento bajo el nombre de Alcaldía tras la creación de la entidad territorial en 1985, previamente este territorio era administrado como parte del municipio vecino de La Paz.

El GAMEA, se compone del poder ejecutivo, representado por el alcalde y su equipo y el concejo municipal constituido por representantes elegidos igualmente por voto popular a través de elecciones municipales cada 5 años.

El municipio de El Alto se halla dividido para su administración, en 14 distritos, 4 de ellos rurales, cada uno bajo la tuición de una sub-alcaldía y una autoridad denominada sub-alcalde, a través de esas entidades se descentralizan algunas actividades administrativas e impositivas.

Misión

El Alto ciudad emergente con alma y espíritu de unidad, con planificación del bienestar social, participativo, desconcentrado, incluyente que garantiza derechos, seguridad del espacio público, con acceso a la salud, educación y protagonista del desarrollo nacional.

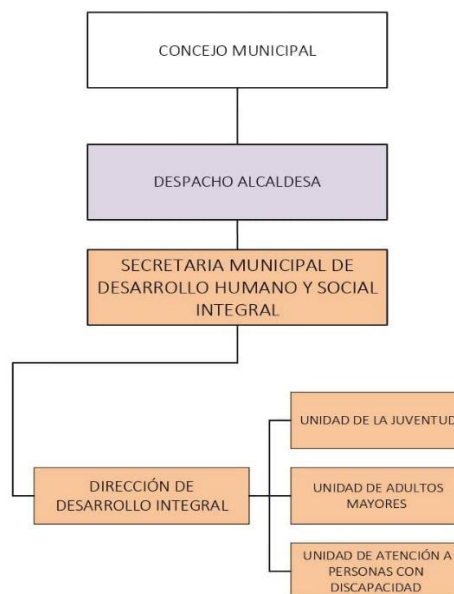
Visión

Gobierno Autónomo Municipal de el Alto, con identidad, ciudad ajayu donde se garantiza el derecho a la salud, educación y sin violencia, con empleo, modernidad, industrias y emprendimientos. Para el desarrollo sustentable y planificado, articulador de la región metropolitana.

ORGANIGRAMA

Figura 1

Organigrama Órgano Ejecutivo del GAMEA



Nota. Manual de Funciones, 2022

1.2.2 Antecedentes afines al proyecto de grado

1.2.2.1 Antecedentes Internacionales

(Zapata, 2019) “Implementación de un sistema web para el control del pago de pensiones escolares en la institución educativa particular San Juan El Obrero S.A.C – 2018.”, objetivo Implementar un sistema web para el control de pagos de matrículas y pensiones escolares en la Institución Educativa Particular San Juan el Obrero S.A.C. Investigación de tipo cuantitativo con un diseño descriptivo, se realizó con la finalidad de ver como ayudaría la implementación de un sistema web para el pago de pensiones escolares en la Institución Educativa. Una muestra de 162 estudiantes en los tres niveles educativos. Las herramientas que se utilizó el lenguaje unificado de modelado. El resultado que se obtuvo es que con un 77% se acepta que la institución educativa use un sistema web para los pagos.

(Buitrón, Carlo, 2021) “Sistema web para mejorar el control de pagos en la I.E.P. Los Salesianos, 2021.”. En el presente proyecto se tiene como objetivo determinar la influencia del sistema web en el control de pagos en la I.E.P. Se tiene una metodología utilizando SCRUM una metodología ágil para reducir el tiempo de elaboración. El tipo de investigación es aplicada, con diseño de investigación preexperimental y un enfoque es cuantitativo. Para la recolección de datos se tuvo una población para el promedio medio de cobro es de 44 alumnos y para promedio medio de pago es de 4 proveedores. Se tomará como muestra la misma cantidad de la población para ambos indicadores. Las herramientas usadas para la programación y el manejo de la base de datos fueron PHP, JQUERY, AJAX, JAVA SCRIPT, CSS y MySQL. Se obtuvo como resultado, mejorar el control de pagos que beneficia a los colaboradores de la empresa, específicamente al área administrativa que podrá tomar mejores decisiones, así como también para la consolidación y crecimiento empresarial.

(Nuñez, Ventura, Zapata, Dioses, 2022) “Implementación de un sistema web, utilizando metodología Scrum para mejorar los procesos de matrícula y control de pagos en la Institución Educativa Particular El Triunfo” cuyo objetivo principal del estudio fue responder a la pregunta: ¿Cómo se puede mejorar el proceso de matrículas y control de pagos? En consecuencia, buscar posibles soluciones a los problemas que enfrenta la institución. La metodología usada es la SCRUM. Las herramientas que se implementó al sistema de proceso de matrículas y control de pagos, un motor de base de datos PostgreSQL desarrollado en Java. La muestra tomada en esta investigación son la cantidad de estudiantes que tiene la Institución Educativa EL TRIUNFO. Como resultado, se decidió implementar un sistema WEB utilizando metodología SCRUM para el proceso de matrículas y control de pagos, para así mejorar sus procesos y reducir el tiempo de atención, además de mejorar el servicio a sus clientes. Se indica que la usabilidad del nuevo sistema con metodología SCRUM tiene un porcentaje del 90% en aceptación por parte de los padres de familia porque poseería mejoras en su atención.

1.2.2.2 **Antecedentes Nacionales**

(Mamani, 2020) “Sistema web de control y seguimiento de servicios y gestión de clientes para la Empresa Consultora Contadores Públicos & Auditores AYS S.R.L”. El presente proyecto estudia la problemática de automatizar los servicios brindados, manejo de información sencilla y facilitar el diálogo razonable con los clientes, tiene como objetivo principal ofrecer a los empleados automatizar los procesos de registro y administración de servicios, gestión de usuarios y almacenamiento de información. Se uso la metodología ágil Kanban y la metodología UWE, esta combinación es un aporte para futuros proyectos. Kanban nos permite visualizar el trabajo en un tablero dividido en bloques y UWE detalla el proceso de las aplicaciones, las herramientas usadas fueron el lenguaje PHP, JavaScript, MySQL y un framework Codeigniter. Se obtuvo resultados donde se redujo costos o gastos, se realizó procesos de control y seguimiento, y un módulo de atención a clientes.

(Condori, 2019) “Sistema web de inscripciones y control de pagos caso: kumon unidad semilla”. El presente proyecto presenta requerimientos de informes e inscripciones para un manejo digital, tiene como objetivo facilitar el proceso de inscripciones y control de pago de mensualidades mediante la implementación de un Sistema Web. Como metodología se usó el SCRUM y la metodología UWE. Las herramientas usadas un framework Codeigniter, gestor de base de datos MySQL y programación en PHP. Como resultado se implementó un sistema de inscripción y control de manera que reduce el tiempo, mejora el control de pagos, centralizar la información de la institución con metodologías adecuadas.

1.3 PLANTEAMIENTO DEL PROBLEMA

El Gobierno Autónomo Municipal de El Alto con la Unidad de Atención a Personas con Discapacidad, está encargada de distintos programas en beneficio a este sector, con el programa del pago a personas con discapacidad grave y muy grave se encarga de realizar la atención y el pago del bono en el Municipio de El Alto. Se observó los siguientes inconvenientes: indisposición de la información lo que ocasiona que las personas que desean obtener información emitidos por la unidad como algún trámite respecto al pago del bono deben asistir de forma presencial, lo que le genera complicaciones y molestias, además de largas filas durante los días de pago y complicaciones en la elaboración de informes.

1.3.1 *Problema Principal*

Con el uso actual de la tecnología, la interrupción o demora en la atención al público afecta la reputación de una institución pública, tener la capacidad de responder las necesidades en menor tiempo posible mejora la relación con la ciudadanía y el prestigio de la institución.

La mala gestión en los procesos de registros y pagos, afecta el nivel de atención a las personas con discapacidad, así también en el ámbito financiero a la unidad, disminuyendo el nivel de satisfacción del usuario.

Se presenta la necesidad de contar con un sistema que ayude a mejorar el control y manipulación de la información, con el registro y actualizaciones de datos, que en muchos casos genera inconsistencia y demoras en la elaboración de informes lo que provoca el descontento de las personas con discapacidad.

1.4 PROBLEMAS SECUNDARIOS

- La información se encuentra desorganizada ya que los documentos se encuentran dispersos en papel sin un orden respectivo.
- Para la obtención de información por parte de las personas con discapacidad como trámites, habilitación de cobro y fechas, deben apersonarse a la unidad de atención de discapacidad lo que conlleva a complicaciones de traslado.
- La información de las personas con discapacidad se encuentra desactualizada, lo que ocasiona dificultad en la localización rápida de datos.
- Riesgo en la pérdida de información durante la interacción con la base de datos.
- Demora en la elaboración de informes, debido a la poca información que se tiene acceso afectando a la toma de decisiones.
- Demora en el proceso de recolección y verificación de las personas con discapacidad habilitadas, prestándose a errores involuntarios.

1.4.1 *Formulación del Problema*

Por lo tanto, habiendo tomado en cuenta las consideraciones y punto de vista tratando el problema identificado se anuncia: ¿Cómo se puede lograr una atención eficaz a las personas con discapacidad que permita brindar información precisa y accesible que coadyuve a un mejor servicio y toma de decisiones en el Gobierno Autónomo Municipal de El Alto?

1.5 OBJETIVOS

1.5.1 Objetivo General

Desarrollar el Sistema de Información Web para el Control y Proceso de Pagos a Personas con Discapacidad del Gobierno Autónomo Municipal de El Alto, para un óptimo manejo de la información, con el fin de coadyuvar a la toma de decisiones.

1.5.2 Objetivos Específicos

- Organizar de manera óptima la información en una base de datos para una óptimo manejo de la información.
- Crear el registro y edición de información que permita un mejor control para la disponibilidad y organización de datos.
- Mejorar los procesos operativos que permita actualizar la información en la base de datos para reducir demoras.
- Generar informes de estados para tener un control de manera precisa.
- Recolectar información de las personas con discapacidad para permitir el pago en las entidades bancarias facilitando la forma de pago.
- Elaborar un módulo de informaciones, donde brinde información para las personas con discapacidad con una interfaz amigable.
- Implementar medidas de seguridad para proteger la información.

1.6 JUSTIFICACIÓN

El desarrollo de un sistema de información para el control y proceso de pagos es un punto importante, porque a través de este se realiza un beneficio social a la población que sufre algún tipo de discapacidad, que le impide realizar algunas labores o la obtención de empleos para sustentar sus gastos diarios, el bono mensual se brinda a más de 4.000 beneficiarios en el Municipio de El Alto. El sistema se convierte en un medio informativo eficaz y eficiente para recabar información de manera rápida y sencilla.

1.6.1 *Justificación Técnica*

Para este proyecto se pretende dar una mayor accesibilidad de datos actuales, datos en tiempo real, referentes a la información que se requiera. Además, el desarrollo del sistema contempla el uso del modelo de programación orientada a objetos y la metodología SCRUM lo cual permite realizar futuras modificaciones del presente sistema.

Las instalaciones de la alcaldía en la cual se implementará el actual proyecto cuentan con los equipos necesarios para el uso del sistema web. Los requerimientos del hardware no serán altos solo se requerirá un navegador web (Chrome, Firefox, Edge, Opera).

Para el desarrollo se usará herramientas que no son ajenas en la parte de desarrollo de páginas web, software de uso libre como el lenguaje de programación PHP con el uso del framework Codeigniter, Javascript, y un gestor de base de datos MySQL; para el diseño se usará herramientas web como CSS3 y Bootstrap. Además, el establecimiento cuenta con un servidor propio que almacenará el dominio del sistema.

1.6.2 Justificación Económica

Desde el punto de vista económico, disminuirá costos gracias a la optimización de procesos que reducirá el tiempo del personal en la realización de pagos, al disminuir el tiempo de atención se obtiene una mejor administración de información, reducir largas filas mejorando el proceso de cobros mensuales.

Los equipos necesarios para su uso mencionados en la Justificación Técnica, no realizarán gastos adicionales de hardware, la implementación usará software libre para evitar usos de licencias.

1.6.3 Justificación Social

El proyecto beneficia en la atención que brinda la Alcaldía Municipal de El Alto y también beneficia a la población Alteña que recibe este servicio, con un sistema de información que esté a su alcance, lo cual resultan de gran beneficio a la población y a la unidad solicitante. Mejoran también los procesos que se realizan actualmente con nuevas herramientas y preservando la información para su uso adecuado.

1.7 METODOLOGÍA

1.7.1 Método de desarrollo

Este método corresponde a una metodología de desarrollo perteneciente a las metodologías ágiles, su objetivo es desarrollar y gestionar proyectos con eficiencia, flexibilidad y control, se basa en la comunicación, reutilización del código desarrollado y realimentación, la metodología SCRUM.

1.7.1.1 Metodología SCRUM

En las metodologías ágiles se encuentra la metodología SCRUM el cual sirve para el desarrollo de software, haciéndose de manera simple, en poco tiempo y con los documentos requeridos a comparación de las metodologías tradicionales. Además, se acomoda a cualquier tipo de cambio la cual la hace preferida por el cliente.

Esta metodología se escoge para proyectos con requisitos que cambien de manera consecuente, ya que se basa en la retroalimentación continua.

La metodología SCRUM también permite desarrollar vínculos interpersonales, con buen ambiente de trabajo, con un adiestramiento colectivo, asegurando el éxito del proyecto.

1.7.2 Métricas de Calidad

1.7.2.1 Modelo de calidad del software ISO 25010

El modelo de calidad permitirá la evaluación del software que interpreta el grado de satisfacción del producto, si cumple con los requisitos del usuario. Se determinan mediante la ISO 25010 que es un estándar internacional para la evaluación de calidad del software, la norma se compone por sus siguientes características:

- Funcionabilidad
- Fiabilidad
- Usabilidad
- Mantenibilidad
- Portabilidad

1.7.3 Costos

1.7.3.1 COCOMO II

Para determinar el costo de desarrollo del software se utilizará el modelo constructivo de costos más conocido como el COCOMO II (Constructive Cost Model).

COCOMO II es uno de los modelos de estimación de coste más utilizados y estudiados, los objetivos toman en cuenta enfoques más modernos para el desarrollo del software, tales como el desarrollo rápido que usa lenguajes dinámicos y el uso de la programación de base de datos.

1.7.4 Seguridad

1.7.4.1 Modelo de seguridad ISO 27002

La Norma ISO 27002 se encarga de establecer estándares y guías relacionados con sistemas de Gestión de la seguridad de la información (SGSI), aplicables a cualquier tipo de organización, orientadas a la mejora continua y a la reducción de riesgos. Este modelo no es solo una norma de gestión sino es una serie de normas orientadas a la implementación de mejores prácticas recomendadas en seguridad de la información.

1.7.5 Pruebas de software

Existe algunos métodos para realizar pruebas a distintos softwares que se realizan, para el presente proyecto se hará el uso de las pruebas funcionales, que garantiza el cumplimiento de las historias de usuario como ser las pruebas unitarias, pruebas de aceptación, pruebas de regresión y las pruebas no funcionales como las pruebas de estrés y de seguridad.

Se nombran las siguientes pruebas a realizar:

- Funciones incorrectas o ausente

- Errores de interfaz
- Errores de rendimiento
- Errores de inicialización y finalización

1.8 MÉTODOS DE RECOLECCIÓN DE DATOS

1.8.1 *Técnicas de investigación*

Para este proyecto se recurrirá a usar la técnica de recolección de datos, se la hará mediante registros a las personas con discapacidad.

1.9 HERRAMIENTAS

PHP: Es un lenguaje de programación de código abierto, diseñado para la creación de páginas web dinámicas que incluyen entre etiquetas especiales de comienzo y final, procesa información de formularios generales, páginas de contenidos dinámicos.

MySQL: Sistema de Gestión de base datos de código abierto, sirve para almacenar y administrar datos relacionales utilizando de entre los diferentes artilugios con los que cuenta con tablas, vistas, procedimientos almacenados, funciones, etc. Además de ser fácil de utilizar y veloz en realizar las operaciones.

JAVASCRIPT: JavaScript es un lenguaje de programación orientado a objetos, diseñado para el desarrollo de aplicaciones cliente/servidor a través de Internet. Permite desarrollar programas que se ejecutan directamente en el navegador (cliente) de manera que éste pueda ejecutar determinadas operaciones o tomar decisiones sin necesidad de acceder al servidor.

HTML: son etiquetas de marcado donde va el código, hace referencia al lenguaje de mercado para la elaboración de páginas web. Es un estándar que sirve de referencia del software, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

CSS: son hojas de estilo en cascada, tecnología que permite crear páginas web de manera más exacta, aplica estilos (colores, formas, márgenes, etc.). CSS utiliza el concepto de separación y presentación de contenidos, intentando que los documentos HTML incluyan información y datos relativos al significado de la información a transmitir.

BOOTSTRAP: Es un framework de desarrollo web de código abierto, diseñado para facilitar el proceso de desarrollo de sitios web responsivos y orientados a los dispositivos móviles. Ayuda a desarrollar sitios web rápidamente, consta de scripts basados en HTML, CSS y JS.

CODEIGNITER: Es un framework para el desarrollo de aplicaciones en PHP, permite a los programadores Web mejorar la forma de trabajar y realizarlo en menor tiempo posible. Codeigniter usa un diseño Modelo Vista Controlador, que permite separar en forma lógica y presentación, esto puede llegar a beneficiar de forma de trabajo en equipo. El framework funciona con un URL, para dirigirse al controlador, como unidad central entre la vista y el modelo.

1.10 LÍMITES Y ALCANCES

1.10.1 Límites

- El sistema se encarga del control de la información, registro y proceso de pagos a personas con discapacidad, que son realizados por la unidad encargada ubicada en la Alcaldía Municipal de El Alto;

- El sistema de realización de pagos solo será accesible por una red interna en instalaciones de la Unidad de Discapacidad por razones de seguridad.
- La actualización de información se realiza solo por el personal autorizado.
- El módulo de información no contiene interacción directa con encargados de la unidad, cualquier otro reclamo deberá apersonarse a las instalaciones.
- El sistema no realizará la encriptación del documento generado en Excel para el cobro en el banco.
- El sistema no está enlazado con el sistema bancario ni con el sistema del Ministerio de Trabajo o de Finanzas.

1.10.2 Alcances

El presente sistema web pretende llegar a cumplir las expectativas que se espera por parte de la unidad solicitante, para llegar al objetivo de este proyecto se plantea los siguientes alcances:

- Obtener una base de datos actualizada donde se realizarán los registro y almacenamiento de toda la información.
- El sistema podrá hacer búsquedas, registro, asignación, modificación de información, realización de pagos.
- Generar un reporte detallado que contendrá información necesaria para implementar los pagos mensuales a través de una entidad bancaria.
- Módulo de cargado mensual, esta permitirá la actualización mensual de información para la habilitación de beneficiarios, que es enviada desde el ministerio de trabajo.
- Módulo de administración de usuarios, que permitirá el control del personal de la unidad y designación de sus funciones.

- Módulo de generación de reportes.
- Módulo de registro para beneficiarios y tutores.
- Módulo de consulta para beneficiarios, lo que les permitirá tener acceso a sus datos y a información emitida por la unidad, la cual podrán acceder desde cualquier sitio.

1.11 APORTES

Se desarrollará una herramienta para facilitar la administración de información y seguimiento del proceso de pagos que realiza la alcaldía de El Alto, facilitando procesos de registros, asignaciones, control de información y operaciones de pagos.

Se otorgará un control más preciso de información centralizándola en una única base de datos, funcional, disponible para la unidad y para las personas con discapacidad, con una interfaz comprensible, con un bajo consumo de recursos y compatible en las plataformas más comunes.

Se desarrolla el manejo de la metodología ágil SCRUM, una metodología adecuada para una mejor organización durante el desarrollo del proyecto, el uso del modelo de calidad ISO 25010 y el manejo del modelo de estimación de costos Cocomo II.

El sistema de información web para el control y proceso de pagos a personas con discapacidad, se desarrolló con el uso de software libre, el uso de un lenguaje de programación más utilizado, Codeigniter 3 basado en php y javascript.

CAPITULO II

2 MARCO TEÓRICO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



2.1 INTRODUCCIÓN

En este capítulo se describe los conceptos mas importantes acerca del desarrollo del proyecto, conceptos amplios ya mencionados en el capítulo I.

2.2 SISTEMA

Un sistema es un conjunto de elementos en interacción mutua, el comportamiento de cada elemento está organizado y dependerá de su objetivo definido, el cual procesa algún tipo de información. (Pressman, 2002).

Un sistema es un conjunto de componentes que interactúan entre sí para lograr un objetivo en común". (Senn, 1992, p.19).

Un sistema es un conjunto de dos o más elementos interrelacionados que conforman un todo, donde un comportamiento en uno de estos elementos afecta a las demás partes. (Montilva, 1986).

Podemos definir a un sistema como la interacción de varios elementos del mismo tipo que se unen para formar una estructura funcional que alcance objetivos planeados.

2.3 SISTEMA DE INFORMACION

Se basa en el proceso de entrada de datos, relacionar información y producir salidas como reportes, con la mutua conexión de hardware y software. (Seen, 1992).

Un sistema de información es un sistema hombre máquina para el proceso y captura de datos a fin de registrar detalles durante algún evento, para proporcionar información que facilite la ejecución de alguna actividad de la organización. (Montilva, 1986).

Con ello podemos definir que se refiere a una estructura organizada que administra y almacena datos e información dentro de una organización, de manera que puedan ser utilizadas fácilmente en cualquier momento.

2.3.1 Características de un Sistema de Información

Se caracteriza por la eficiencia en el proceso de datos, tener un conjunto de reglas que organizan la información para ponerlas a disposición de la organización y los usuarios.

También se destaca por el soporte que brinda a la organización institucional, para una pronta toma de decisiones, facilitan la relación entre usuarios de la organización, facilitan el manejo de información para empleados, control de información, clasificación de datos y un aspecto social que incluye a las personas, informaciones y procesos.

2.4 SISTEMA WEB

Un sistema web es una aplicación de software que se almacena en un servidor y que se puede tener acceso por medio de un navegador web a través de una red. (Senn, 1992).

Un sistema web es un conjunto de componentes relacionados que trabajan en el lado cliente o el servidor por medio del internet. (López et al., 2014)

Es un software que utiliza tecnologías web para brindar funcionalidad y servicios a través de internet. (Pressman, 2010).

Por ello lo podemos definir como un software que se ejecuta en un navegador web proporcionando servicios a usuarios a través de internet o intranet sin la necesidad de alguna instalación, utilizando una conexión a una base de datos para alojar y mostrar información.

2.5 CONTROL

Es el proceso de regular actividades que aseguren que se están cumpliendo como fueron planificados y corrigiendo cualquier desviación significativa (Robbins, 1996, p.654).

Consiste en verificar si todo se realiza conforme al programa adoptado, a las ordenes impartidas y a los principios administrativos. Tiene la finalidad de señalar las faltas y los errores a fin de que se pueda repararlos y evitar su repetición (Melinkoff, 1990, p.62).

Se define control como la función de supervisar alguna actividad, darle seguimiento a la espera de resultados, con una supervisión se pueden establecer estándares con la finalidad de corregir posibles fallas.

2.6 PAGOS

Pago es toda aquella acción que realizamos para extinguir o cancelar una obligación. Se basa en la entrega de un bien, servicio o activo financiero a cambio de otro bien, servicio o activo financiero. (Pedrosa, 2020)

Con ello lo definimos como a una transacción monetaria, una deuda que se es saldada por algún servicio prestado poniéndole fin a alguna obligación, también se entiende que puede realizarse la acción de pagar a través de la entrega de un bien o activo, del que suponemos tiene un valor similar a la obligación contraída.

2.6.1 Proceso de Pagos

Un proceso de pago es una secuencia de pasos y acciones que se realizan para completar una transacción financiera en la que una parte (salida de la transacción) entrega un valor económico, como dinero o medios de pago electrónicos, a otra parte (el beneficiado), por algún servicios o deuda.

2.6.2 Proceso de pagos a beneficios sociales

Existen distintos tipos de pagos, como por beneficios sociales a través de bonos o recompensas, este puede variar según la institución que lo realice:

- **Determinación de Elegibilidad:** Se determina quienes son aptos de recibir el beneficio del bono, basado por el rendimiento, antigüedad, logro o algún impedimento físico.

- Cálculo del Monto del Bono: A través de una o varias organizaciones se calcula el monto que se otorgará a las personas elegibles y la forma que se procederá.
- Notificación: Los beneficiarios serán notificados sobre el bono que recibirán, junto con detalles sobre el monto y cualquier otro requisito o información relevante.
- Aprobación: El proceso de pago puede requerir la aprobación de una organización o autoridades de gobierno.
- Generación de Documentos: Se generan los documentos necesarios para formalizar el proceso de pago.
- Pago: Se realiza el pago del bono según la forma establecida. Depositado en la cuenta bancaria del beneficiario, cheques o de forma física.
- Registro y Seguimiento: La organización registra los pagos realizados y mantiene un seguimiento de los registros para fines contables y de presentación de informes.
- Confirmación: Los beneficiarios reciben una confirmación del pago de su bono por medios brindados por la organización.
- Cierre del Proceso: Una vez que se ha realizado el pago y se ha realizado la confirmación, el proceso de pago del bono se da por terminado.

2.7 NORMA LEGAL EN EL SERVICIO DE PAGOS

Se define la norma legal como una regla que prescribe cómo debe comportarse una persona o entidad y que es respaldada por una autoridad reconocida. (Kelsen, 2018)

Entonces son las que permiten acceder a pagos de beneficios sociales, son una regulación o ley establecida por una autoridad gubernamental que define los criterios, requisitos y procedimientos que las personas deben cumplir para ser elegibles y recibir un bono o beneficio específico.

Estas normas legales son utilizadas por los gobiernos o instituciones para administrar programas de asistencia financiera y garantizar que los beneficios lleguen a las personas que cumplen con ciertos criterios como situación familiar, por edad, discapacidad y otros.

2.7.1 Ley 977

Con la promulgación de la ley 977 de inserción laboral y ayuda económica para personas con discapacidad, el Gobierno del Estado Plurinacional de Bolivia fortalece las políticas sociales que contribuye a la reducción de la pobreza y a mejorar la calidad de vida de las personas con discapacidad. Se otorga el beneficio desde el 01 de febrero del 2019 a nivel nacional, el bono se beneficia a personas con discapacidad grave y muy grave a través de los Gobiernos Autónomos Municipales.

Por medio del Ministerio de Salud quienes califican el nivel de gravedad, siendo admitido un porcentaje mayor o igual al 50%, se le proporcionará un carnet que le dará paso a ser acceder al servicio de pago.

2.8 INGENIERIA DE SOFTWARE

La ingeniería de software es una disciplina que se encarga de todos los aspectos en la producción de software, ofrece técnicas y métodos para el desarrollo, además de implementar técnicas de calidad desde los primeros momentos del sistema hasta el mantenimiento de los sistemas, cumplir con las necesidades principales, fácil de usar confiable y proveer un rendimiento adecuado. (Pressman, 2010)

Por ello podemos considerar a la ingeniería de software como los medios sistematizados con herramientas preestablecidas para el desarrollo de programas sistematizados.

Por lo tanto, la ingeniería de software, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento e implementación del sistema. El proceso debe tener una organización sistemática y bien definida.

2.8.1 Actividades en el proceso de desarrollo de software

- **Análisis de requisitos:** Esto implica comprender qué debe hacer el software, identificar y recopilar requisitos y qué características debe tener para satisfacer las necesidades de los usuarios.
- **Planificación del Proyecto:** Se desarrolla un plan de proyecto que incluye la definición de los objetivos del proyecto, el alcance, el cronograma, los recursos necesarios y los riesgos potenciales.
- **Diseño:** En esta etapa, se crea un diseño detallado del sistema. Esto incluye la arquitectura del software, la definición de componentes, módulos y la planificación de la interfaz de usuario.
- **Programación:** El ingeniero se encarga de elaborar el código del software de acuerdo con los diseños previamente establecidos.
- **Prueba y Verificación:** En esta etapa se llevan a cabo las pruebas para garantizar que el software funcione como se esperaba y cumpla con los requisitos establecidos. Se puede llegar a probar cada módulo por separado.
- **Documentación:** Se crea documentación técnica y de usuario para el software. Esto incluye manuales de usuario, documentación de diseño y otros recursos, todo con el propósito de correcciones, usabilidad y mejoras futuras.

- **Implementación y Mantenimiento:** El software se implementa para el uso del usuario, y con el tiempo requerirá correcciones de errores, con un mantenimiento que asegure que el software aún siga útil y eficaz.

2.9 METODOLOGIA PARA EL DESARROLLO DE SOFTWARE

La metodología de desarrollo se considera como una estructura utilizada para planificar y controlar el procedimiento en cuestión a la elaboración de un proyecto, tomando en cuenta la calidad y la satisfacción del cliente.

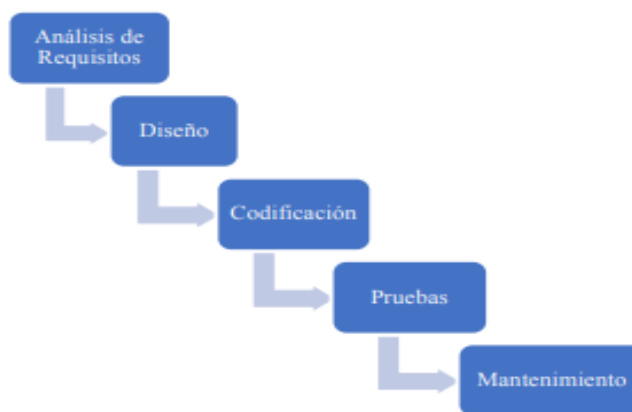
2.9.1 Metodologías Tradicionales

Estas metodologías también llamadas modelos de proceso prescriptivo, que fueron planificados para la organización en el desarrollo de software. Propone la elaboración de manera única con una estructura definida, en una sola dirección y sin marcha atrás, el proceso es rígido y no cambia, los requerimientos serán acordados durante la etapa de planificación, demandando mayor tiempo de desarrollo y poca comunicación con el cliente una vez terminada.

Se basa principalmente en las etapas de análisis de requerimientos, diseño, codificación, pruebas y mantenimiento. Además, las metodologías tradicionales no son aptas a los cambios, por lo que estos métodos no son adecuados cuando se trabaja en un entorno donde se requiera cambios o no se estime lo requerido.

Figura 2

Fases del modelo en cascada



Nota. Montero, B. M., Cevallos, H. V., & Cuesta, J. D, 2018 (p.116)

2.9.2 Metodologías Ágiles

Las metodologías ágiles presentan como principal particularidad la flexibilidad con soluciones de manera ágil, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una comunicación constante con el usuario para una respuesta oportuna, son altamente colaborativos y es mucho más adaptable a los cambios.

Entre las más notables metodologías ágiles se encuentran la SCRUM, XP, Crystal Clear, Mobile-D, ASD y Lean Development.

2.9.3 Diferencias entre metodologías Tradicionales y Ágiles

La diferencia más resaltante es que la metodología tradicional impone una disciplina de desarrollo fundamentada en la documentación del proceso, se realiza un hincapié en la planificación del proyecto, una vez detallado comienza el proceso de desarrollo hasta su fase final. Caso contrario la metodología ágil en muchas veces puede obviar la documentación realizada a inicios, dando prioridad y centrándose en las necesidades que se presenten.

Tabla 1

Metodologías tradicionales y metodologías ágiles.

METODOLOGIAS TRADICIONALES	METODOLOGIAS AGILES
Predictivos orientado a procesos	Adaptativos, orientados a personas
Proceso rígido, se concibe como un proyecto	Proceso flexible, un proyecto es subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente.	Comunicación constante con el cliente.
Entrega de software al finalizar el desarrollo.	Entregas constantes de software.
Documentación extensa.	Poca documentación.

Nota. Diferencia entre Metodología tradicional y ágil

2.10 METODOLOGIA SCRUM

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación. (Schwaber & Sutherland, 2016)

SCRUM define tres roles, los cuales son:

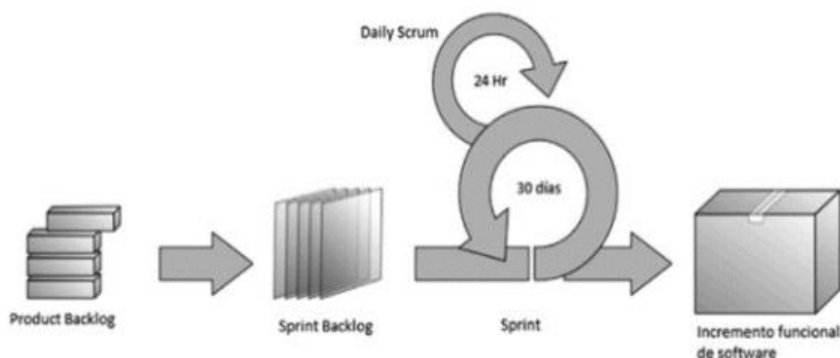
- **El Product Owner:** es el dueño del producto y representa a los que usan el software.
- **El Scrum Master:** es la persona que lidera el equipo asegurándose que el equipo cumpla las reglas y procesos de la metodología.

- **Team:** el equipo de desarrollo, grupo de profesionales encargados de convertir la lista de requerimientos (Product Backlog) en funcionalidades del software.

Scrum utiliza un elemento representativo llamado Sprint (Figura 3) que corresponde a una etapa de trabajo donde se crea una versión utilizable del producto.

Figura 3

Fases del Sprint



Nota: (Cohn, 2004) tomado de (Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018))

Cada sprint es considerado como un proyecto individual; cada sprint está compuesto por lo siguiente: reunión de planeación del Sprint, Daily Scrum o reunión diaria, trabajo de desarrollo, revisión y retrospectiva del Sprint.

2.11 HERRAMIENTAS

2.11.1 Herramientas para el desarrollo de software

PHP

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataforma sin ningún costo. (Flores y Hernandez, 2021).

("Hypertext Preprocessor") es un lenguaje de programación ampliamente utilizado para el desarrollo de sitios web dinámicos y aplicaciones web. Se destaca por su capacidad para generar contenido web en tiempo real y trabajar en conjunto con HTML para crear páginas web interactivas.

HTML

Es un lenguaje estándar para crear páginas web. Se trata de la sigla que corresponde a Hipertexto Markup Language, es decir, Lenguaje de Marcado de Hipertexto. El HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc.); dichos elementos están representados por etiquetas las cuales son interpretadas por el navegador para visualizar el contenido de la página, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).

También permite que ciertos códigos que se conocen como scripts, puedan interactuar con las etiquetas que brindan instrucciones específicas a los navegadores que se encarga de generar un resultado a partir de un dato como el Javascript y PHP. (Flores y Hernandez, 2021).

JAVASCRIPT

Javascript habilita páginas webs interactivas y es una parte esencial de las aplicaciones web. Cada navegador web tiene un motor de Javascript dedicado a ejecutarlo ya que Javascript admite estilos de programación impulsados por eventos, funcionales e imperativos. Javascript permite crear funciones para implementar algoritmos, creando un archivo .js el cual pueda incluirse en una página HTML. (Flores y Hernandez, 2021).

CSS

Cascading Style Sheets, es un lenguaje que permite describir como los elementos HTML son desplegados en el navegador Web. La estructura de un estilo CSS consiste en los elementos: selector, atributo y valor. Esta estructura permitirá a los desarrolladores el control total sobre estilo y formato de sus módulos. (Flores y Hernandez, 2021).

JQUERY

JQuery es una biblioteca de JavaScript liviana, el propósito de jQuery es hacer que sea mucho más fácil usar JavaScript en su sitio web. JQuery toma muchas tareas comunes que requieren muchas líneas de código JavaScript para lograrlo, y las envuelve en métodos que puede llamar con una sola línea de código. La biblioteca jQuery contiene las siguientes características: manipulación HTML y CSS y posee un método útil para trabaja "Ajax", método configurado a través de un objeto, el cual contiene todas las instrucciones que necesita jQuery para completar una petición.

APACHE

Es un servidor web HTTP de código abierto, para la creación de páginas y servicios web, es parte primordial de cualquier sitio web. Apache es uno de los servidores de páginas web más utilizados junto a MySQL y lenguajes de programación como PHP y PYTHON, posee varias ventajas la cual facilita su uso, como su instalación, el soporte, extensible, de código abierto, gratuito y destacar por su seguridad y rendimiento.

Este servidor puede conectarse directamente a una base de datos Oracle, varios programadores lo utilizar como un servidor local para visualizar el producto al momento de desarrollar una página web, este servidor web utiliza el protocolo (<http://>). Apache tiene bastante tiempo en su salida y ha ido desarrollando diferentes soluciones para evitar la ineficiencia en el servicio.

FRAMEWORK CODEIGNITER

Es un potente framework PHP muy liviano de código abierto, desarrollada por primera vez en 2006. Permite la creación de aplicaciones web más rápida en comparación con otras framework. CodeIgniter utiliza la función de Modelo Vista Controlador (MVC), el cual es útil y necesario para optimizar el trabajo de programación de las aplicaciones web.

FRAMEWORK BOOTSTRAP

Es un Framework de CSS y JavaScript. Fue ideado por Twitter en 2010 para simplificar la creación de interfaces limpias y con diseño responsive, y ofrece una amplia gama de herramientas y funciones que permiten a los usuarios dar vida a web sites y apps responsivas.

Utiliza componentes vitales para los desarrolladores tales como HTML5, CSS3, jQuery o GitHub, entre otros. Se integra con librerías JavaScript y utiliza Less, lenguaje de las hojas de estilo CSS preparado para enriquecer los estilos del website.

APACHE JMETER

Apache JMeter es un software de código abierto se puede utilizar para probar el rendimiento tanto en estática como dinámica. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, etc. Además de una red u objeto para probar su fuerza para analizar el rendimiento general bajo diferentes tipos de carga.

JMeter no es un navegador, funciona a nivel de protocolo. En lo que respecta a los servicios web y los servicios remotos JMeter parece un navegador (o más bien, múltiples navegadores) sin embargo JMeter no realiza todas las acciones compatibles con los navegadores.

2.11.2 Gestor de Base de Datos

Es un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y salida de información.

Existen actualmente muchas formas de bases de datos, que van desde bibliotecas hasta vastos conjuntos de datos de usuarios, organización, empresas, con el fin de proporcionar información que facilitará el manejo de datos.

➤ MYSQL

MySQL es el sistema de gestión de bases de datos relacional más extendido con un modelo cliente-servidor, en la actualidad está basada en código abierto relacional, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente. Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL.

Se puede acceder a MySQL a través de una interfaz grafica de usuario (GUI) o herramientas de la interfaz de línea de comandos.

Además de gozar con una gran flexibilidad, facilidad de uso y optimo rendimiento.

Características del MySQL:

- Arquitectura Cliente y Servidor
- Compatibilidad con SQL
- Vistas
- Procedimientos almacenados
- Desencadenantes
- Transacciones

2.12 METRICAS DE CALIDAD

Las métricas de calidad indican la manera en que el proceso de control de calidad medirá el trabajo del producto, diversos aspectos del software como funcionalidad, el rendimiento, la confiabilidad, la seguridad y otros atributos clave. Se destaca la importancia de utilizar métricas de calidad para obtener información objetiva y medible sobre la calidad del software en otras palabras son medidas cuantitativas que se utilizan para evaluar y asegurar la calidad del software durante su desarrollo y mantenimiento. (Pressman, 2010).

2.12.1 Norma ISO 25010

La norma ISO 25010 es un estándar internacional para la evaluación del software y es parte de una familia de normas ISO 25000, y tiene como objetivo la creación de un marco de trabajo para evaluar la calidad del producto.

El modelo de calidad del producto definido por la ISO 25010 se encuentra compuesto por las características de calidad que se muestra en la siguiente tabla:

Tabla 2

Descripción de las características de la ISO 25010

CARACTERÍSTICA	DESCRIPCIÓN
FUNCIONABILIDAD	El correcto funcionamiento del producto, la capacidad de un producto para proveer las funciones requeridas para resolver aquellas necesidades para las que fue diseñado o adquirido.
USABILIDAD	Capacidad del software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

CARACTERÍSTICA	DESCRIPCIÓN
FIABILIDAD	Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados, una expresión usada la del ing. Lourival Tavares, el cual la confiabilidad está en función del tiempo medio entre fallas y tiempo medio de reparación.
MANTENIBILIDAD	Esta característica representa la capacidad del software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
PORTABILIDAD	Capacidad del producto o componente de ser transferido de forma efectiva y eficiente a un entorno diferente (hardware, software, operacional o de uso).

Nota. Conceptos del proceso de calidad

2.13 METÓDO DE ESTIMACIÓN DE COSTOS

Es el método que nos permitirá realizar un análisis de costos del producto, en la que se evalúan los costos materiales, el esfuerzo realizado y la planificación de la implementación del proyecto de software.

2.13.1 COCOMO II

COCOMO II es actualización y mejora del modelo COCOMO original, desarrollado por Barry Boehm es un modelo de estimación de costos y esfuerzos, se estima los costos necesarios para llevar a cabo proyectos de desarrollo de software.

COCOMO II comprende varios modelos y submodelos que abordan diferentes niveles de detalle y precisión en la estimación, desde las primeras etapas del proyecto hasta etapas más avanzadas con más información disponible.

COCOMO II posee tres Modelos de estimación:

- **Composición de Aplicación:** Se emplea en las primeras etapas de requerimientos de desarrollo de software, durante el diseño de prototipo.
- **Modelo diseño Temprano:** Se utiliza en las primeras etapas del desarrollo en las que se evalúan la alternativa de hardware y software de un proyecto. En estas etapas se tiene poca información, lo que concuerda con el uso de Puntos Función, para estimar tamaño y el uso de un número reducido de factores de costo.
- **Modelo Post-Arquitectura:** Se aplica en la etapa de desarrollo, después definido la arquitectura del sistema y en la etapa de mantenimiento. Este modelo utiliza Puntos Función y/o líneas de código para estimar el tamaño.

2.13.2 Método de Estimación

La estimación hace referencia al costo total del sistema se considera los siguientes costos: costo del software desarrollado y costo de implantación del sistema.

Por otro lado, existen diferentes modelos que define COCOMO:

- **Proyectos Simples u orgánicos:** Aplicaciones bien entendidas desarrolladas por equipos pequeños.
- **Proyectos Moderados o semi-acoplado:** Aplicaciones más complejas, en las que el equipo de desarrollo tiene experiencia limitada en el tipo de sistema.
- **Proyectos Empotrados:** Proyectos complejos donde la aplicación es parte de un fuerte acoplamiento de software, hardware y reglas operacionales.

Tabla 3*Detalle de Coeficiente de COCOMO II*

MODO	BASICO				INTERMEDIO			
	A	B	C	D	a	B	C	D
Orgánico	2,4	1,05	2,5	0,38	3,2	1,05	2,5	0,34
Semi – orgánico	3,0	1,12	2,5	0,35	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32	2,8	1,20	2,5	0,32

Nota. (Roger Pressman, 2010)

Para la implementación del modelo de COCOMO II previamente necesitamos conocer el número de líneas de código, posteriormente para poder realizar los cálculos del modelo de estimación de costos, usamos las siguientes ecuaciones:

Tabla 4*Ecuaciones de Método de COCOMO II*

VARIABLE	ECUACIÓN	TIPO / UNIDAD
Esfuerzo	$E = a * (KLDC) B$	Personas - mes
Tiempo	$T = C * (E)^d$	Meses
Número de personas	$P = E/T$	Personas
Costo total por mes	$CM = NP * Salario$	\$us - Bs
Costo total	$CT = CM * T$	\$us - Bs

Nota. (Roger Pressman, 2010)

2.14 SEGURIDAD DE LA INFORMACION

La seguridad de la información tiene como fin proteger la información que almacena nuestro sitio web. La seguridad se trata de indicar los procedimientos necesarios para resguardar los datos almacenados, disminución de peligros o daño de todo aquello que afecte los resultados que se obtienen. Tomando en cuenta que no existe hasta el momento algún método que proteja por completo a un sistema, el riesgo siempre estará presente. Pero existe formas de brindar un modelo que minimice el riesgo, como ser la norma ISO 27002.

2.14.1 Norma ISO 27002

Las organizaciones internacionales ISO/IEC (The International Organization for Standardization) e IEC (International Electrotechnical Commission) dieron origen a un grupo de normas que consolidan directrices relacionadas al alcance de la seguridad de información, siendo representada por la serie 27000. En este grupo se encuentra la norma ISO/IEC 27002, puede ser utilizada para apoyar la implantación del SGSI en cualquier tipo de organización como el control de acceso, la autenticación, el cifrado, encriptación y la seguridad de los recursos humanos. Esta norma sirve de modelo práctico para las organizaciones que deseen proteger eficazmente sus activos de información contra las ciberamenazas. Mientras que ISO/IEC 27001 especifica los requisitos para implantar un SGSI, ISO/IEC 27002 proporciona las buenas prácticas y controles detallados que pueden aplicarse dentro del SGSI

Para brindar la protección de la información, la norma 27002 contempla los siguientes objetivos:

- Preservar la confidencialidad de los datos
- Conservar la integridad de los datos
- Disponibilidad de la información protegida

2.15 PRUEBAS DE SOFTWARE

Es el proceso de evaluar y verificar que una aplicación de software hace lo que tiene que hacer que el software debe funcionar correctamente. Beneficia principalmente en la prevención de errores, la reducción de los costos de desarrollo y sobre todo mejora el rendimiento.

Pruebas funcionales

Son las pruebas que se realizan a los requerimientos establecidos por el usuario, cada funcionalidad del sistema se prueba proporcionando la información adecuada. Esto implica la verificación de la interfaz de usuario, las API, la base de datos, la seguridad, las aplicaciones cliente/servidor y la funcionalidad de la aplicación bajo prueba.

Entre las pruebas funcionales se encuentran, la prueba unitaria, prueba de humo, aceptación de usuario, pruebas de integración, pruebas de regresión, aplicaciones móviles.

Pruebas no funcionales

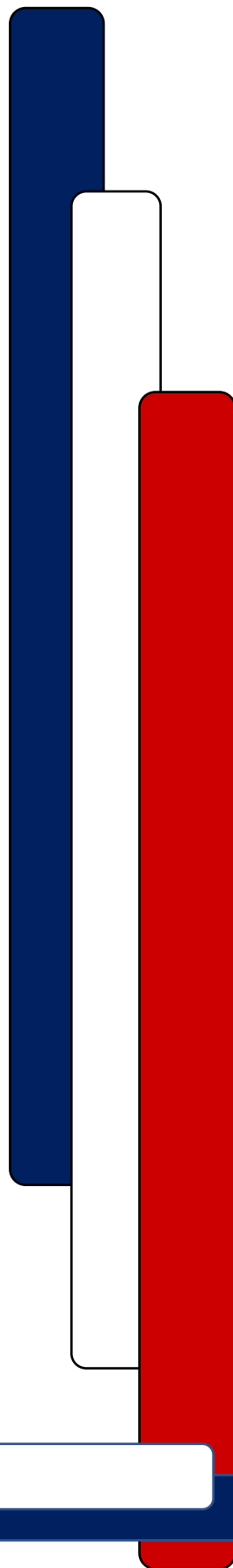
Es un tipo de prueba para comprobar aspectos no funcionales (rendimiento, usabilidad, confiabilidad, etc.) de una aplicación de software, es decir no requeridas por el usuario final.

Entre las pruebas no funcionales se encuentran, test de rendimiento, pruebas de usabilidad, prueba de carga, prueba de estrés, prueba de conformidad, prueba de seguridad.

CAPITULO III

3 MARCO APLICATIVO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



3.1 INTRODUCCIÓN

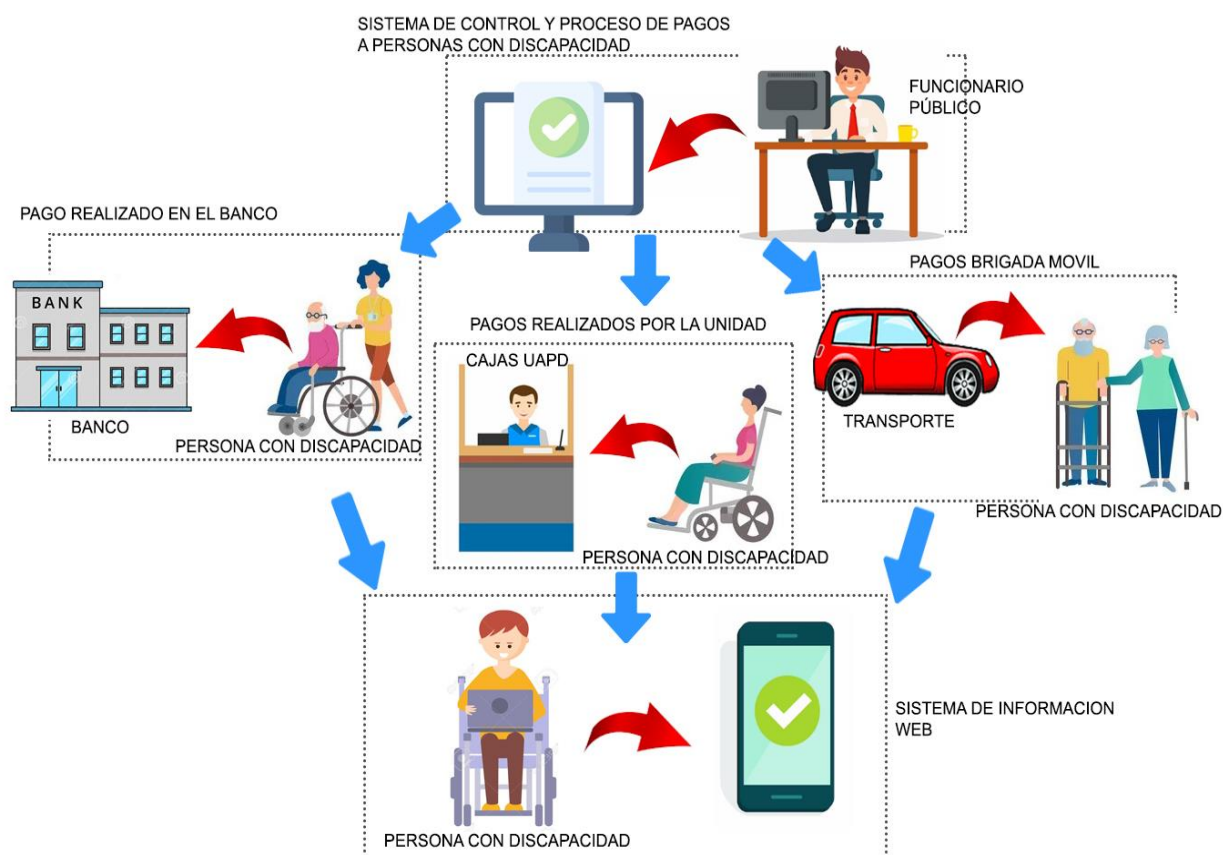
En este capítulo se describe las etapas del desarrollo, la fase de obtención de requerimientos, descripción de funciones, diseño del sistema, pruebas y la fase de implementación. En todo el proceso se aplicó la metodología de desarrollo SCRUM, siguiendo el proceso mencionado en el capítulo II.

3.1.1 Esquema del Funcionamiento del Sistema

Se demuestra el funcionamiento del sistema y el proceso de control de información que se requiere en la unidad.

Figura 4

Esquema del Flujo del sistema

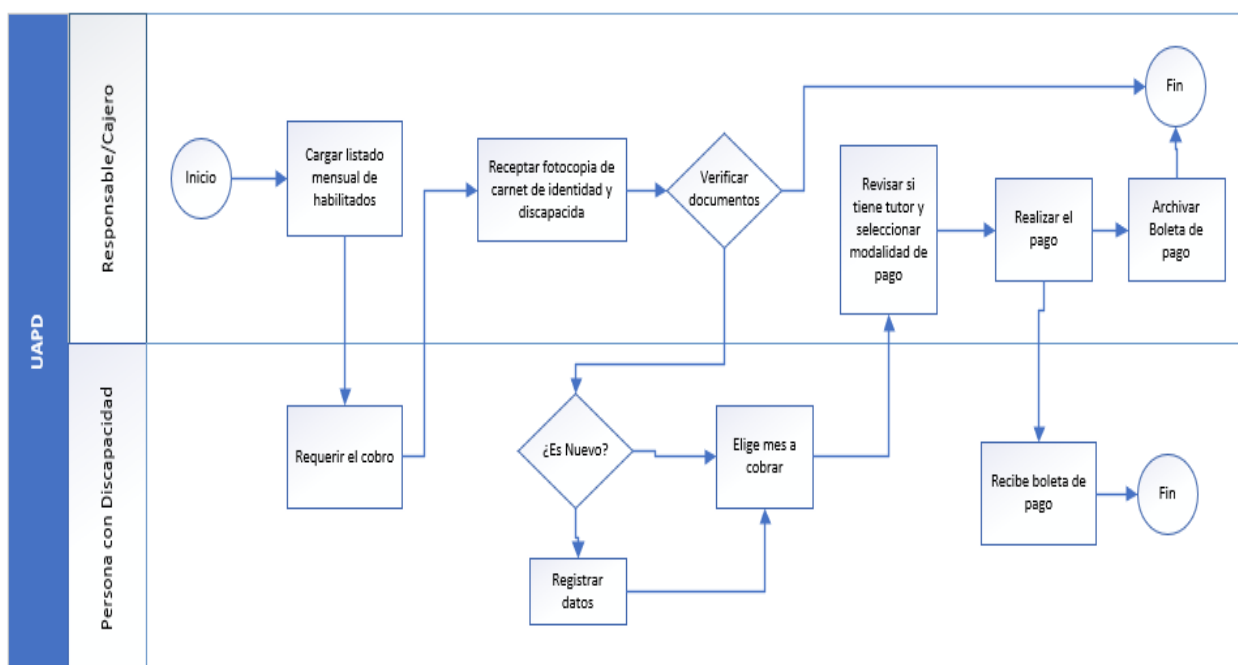


3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL

La Unidad de Atención a personas con discapacidad se encarga de realizar los registros de beneficiarios del bono discapacidad en el Municipio de El Alto, realiza pagos mensuales, la recepción de documentos y la verificación de información. Los documentos se almacenan de forma física en la unidad, eso hace que se demore la revisión de información.

Figura 5

Diagrama de flujo de datos pago a personas con discapacidad



Nota. Proceso en la realización de pagos mensuales

3.2.1 Descripción de la Institución

La Unidad de Atención a Personas con Discapacidad (UAPD), se ubica en la Avenida Costanera, Nro: 5022 Urbanización Libertad, entre calle J.J. Torrez y calle Hernán Siles Zuaso (Jach'a Uta) en la planta baja del Gobierno Autónomo Municipal de El Alto (GAMEA).

3.3 PRE-GAME

Para el desarrollo del sistema web se obtuvo los requerimientos del sistema tras la reunión con la responsable y entrevistas con los usuarios que harán el respectivo uso, con la finalidad de obtener información del funcionamiento y de sus actividades.

Tabla 5

Actividades de usuarios

ACTIVIDADES	DESCRIPCION
EFFECTUAR PAGO	Los usuarios (cajeros) se encargan de efectuar los pagos cada mes, por 7 días calendario, a través de dos modalidades, brigada móvil y punto fijo.
REGISTRO DE DATOS	El usuario se encarga de registrar o actualizar los datos de cada beneficiario habilitado para permitirle realizar el cobro.
ELABORACION DE REPORTE	El Encargado realiza la tarea de elaborar reportes mensuales tras concluir las fechas de pagos,
GESTIONAR INFORMACION DE BENEFICIARIO	El Encargado tiene la responsabilidad de tener el control de toda la información actual referente al beneficiario, desde su edad y ubicación.
ACTUALIZACION DE REGISTROS PARA PAGOS	El Encargado o Administrador tiene la función de actualizar la base de datos, para ingresar el listado de beneficiarios habilitados para el cobro del mes.

Nota. Descripción de actividades

3.3.1 **Product Backlog**

Se obtendrá los requisitos funcionales y no funcionales, lo que nos permite realizar una lista de requerimientos detallados que vienen a construir el product Backlog.

Requerimientos funcionales:

HU1: Registro de usuarios: creación de la base de datos.

HU2: Acceso al sistema: Autenticación de usuarios según sus roles.

HU3: CRUD de usuarios: El “Administrador” podrá tener acceso a este panel, para poder agregar, editar, actualizar, inhabilitar y asignar roles a los nuevos usuarios.

HU4: Panel Administrador – Encargado: Accesos directos a todos los módulos asignados.

HU5: Panel Cajero: Accesos directos a todos los módulos asignados.

HU6: Módulo Pago a Beneficiario: Los usuarios podrán acceder a este módulo para realizar el pago del beneficiario.

HU7: Módulo Historial de pagos: Los usuarios tendrán acceso a revisar su historial de pagos realizados durante un día en específico.

HU8: Módulo Registrar datos: Los usuarios tendrán acceso a registrar o editar los datos a nuevos beneficiarios y tutores si se requieren.

HU9: Beneficiarios Observados: Los usuarios tendrán acceso al control de habilitación o inhabilitación del beneficiario si este sea menor de edad o en caso de mayores de 60 años.

HU10: Búsqueda de datos: Los usuarios tendrán la opción de realizar búsquedas de datos que requieran en el momento, como números de identificación, nombres y apellidos.

HU11: Digitalizar Documentos: Los usuarios podrán digitalizar los documentos originales del beneficiario, como ser su certificado de nacimiento, carnet de discapacidad y otros.

HU12: CRUD Zonas: Acceso a la creación y edición de distritos y zonas.

HU13: Reportes: El usuario con permiso de “Administrador” y “Encargado” tendrá la opción de generar reportes de la cantidad de pagos del mes, registros de pagos y reporte generado de beneficiarios habilitados para el envío a la entidad bancaria.

HU14: Cargado de datos: El usuario con permiso de “Administrador” y “Encargado” tendrá la opción de importar la lista de habilitados del mes a la base de datos.

HU15: Modulo de verificación de datos: Se desarrolla un módulo para los beneficiarios, en el que ellos podrán acceder con el ingreso de su cédula de identidad, y revisar sus datos e historial de cobro.

Requerimientos no funcionales:

- RQNF - 01: El sistema deberá ejecutarse en cualquier navegador Web.
- RQNF - 02: El sistema deberá ejecutarse con el mínimo de recursos.
- RQNF - 03: El sistema deberá mostrarse con una interfaz de usuario fácil de entender y usar.
- RQNF - 04: El sistema deberá contener accesos directos a sus módulos.
- RQNF - 05: El sistema deberá tener un tiempo de respuesta no mayor a los 3 segundos.
- RQNF - 06: El sistema deberá contar con una BD estable y bien definida con sus respectivas credenciales y protocolos de seguridad para garantizar la integridad de los datos almacenados.
- RQNF - 07: Se requerirá una conexión cableada a internet para el uso del sistema.
- RQNF - 08: Se requerirá una impresora para la impresión de boletas de pago y formularios de datos.

A partir de los requerimientos funcionales encontrados, se genera una lista de Sprints:

Figura 6*Listado de Sprints planificados*

nro Sprint	Descripción	Historias de Usuario	Prioridad	Riesgo	Duración (días)
0	Iniciar el proyecto con Codeigniter				2
1	Desarrollo del acceso al sistema y Base de datos	HU01	1	1	10
		HU02	1	2	
2	Desarrollo del módulo referente al pago, historial y registros	HU06	1	1	12
		HU07	1	2	
		HU08	1	2	
3	Desarrollo del módulo reportes	HU13	1	1	10
4	Desarrollo módulo control de información del beneficiario	HU11	2	2	10
		HU09	2	2	
5	Desarrollo del CRUD	HU03	2	2	7
		HU12	2	3	
6	Desarrollo del módulo de cargado de datos	HU14	2	1	10
7	Desarrollo el módulo de consultas	HU15	2	3	7
8	Desarrollo de los módulos de ayuda	HU04	3	3	7
		HU05	3	3	
		HU10	3	3	
TOTAL					75

Nota: Clasificación de las historias de usuario según su prioridad

3.3.2 Especificación

En base a la arquitectura del sistema MVC, el modelo está conformado por Scripts PHP que manipulan los datos interactuando con la base de datos, la vista está conformada por páginas HTML y el controlador conformado por Scripts de JavaScript.

3.3.2.1 Actores del sistema

Para el desarrollo del sistema se identifican los siguientes actores en la tabla que realizarán el uso y la evaluación:

Tabla 6

Identificación de Actores

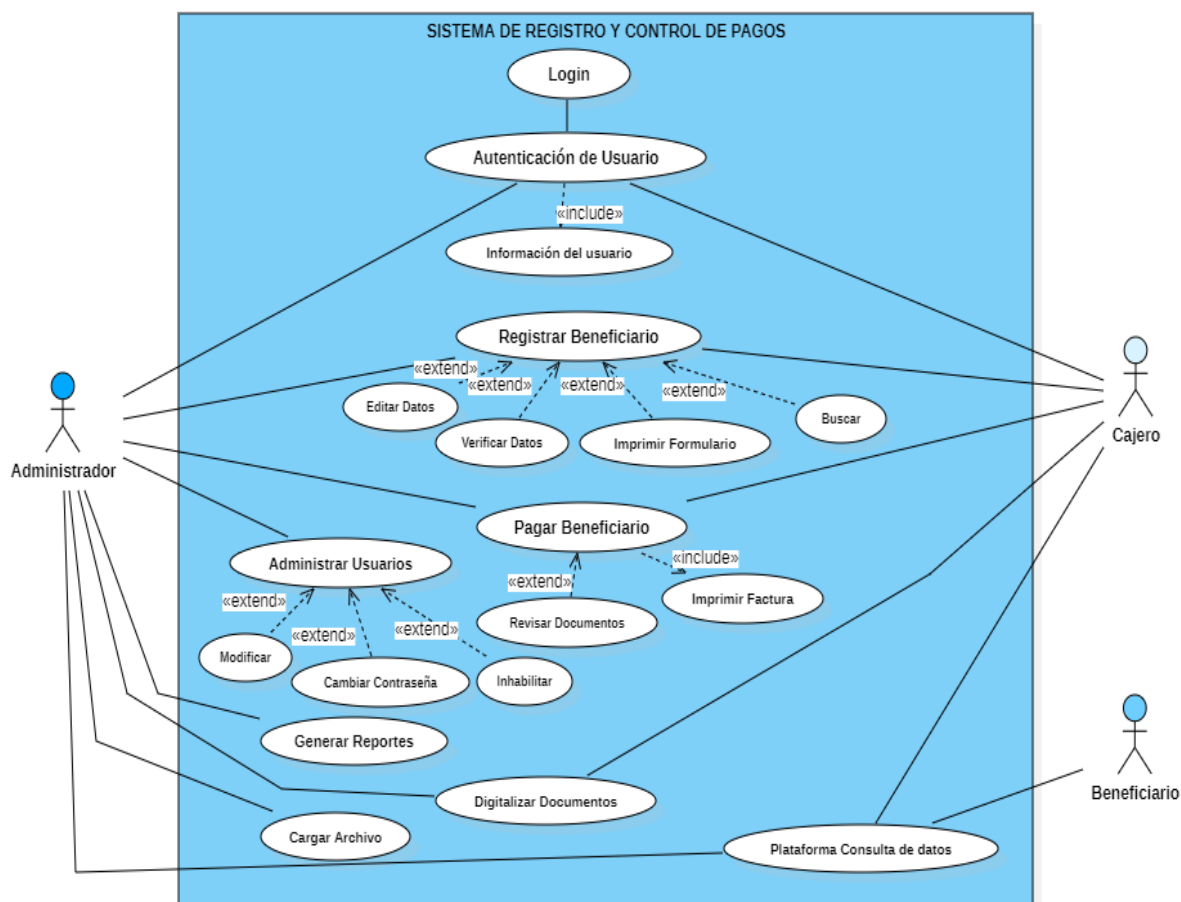
ACTOR	DESCRIPCION
ADMINISTRADOR O ENCARGADO	Tiene la función de administrar a los usuarios, elaborar reportes y actualización de registros, gestionar datos y efectuar el pago.
CAJERO	Tiene la función de registrar beneficiarios y efectuar el pago.

3.3.2.2 Diagrama de casos de uso

Se realiza el modelado del sistema donde se puede apreciar la forma en la cual los actores podrán interactuar con los módulos del sistema.

Figura 7

Diagrama de Casos de Uso del sistema

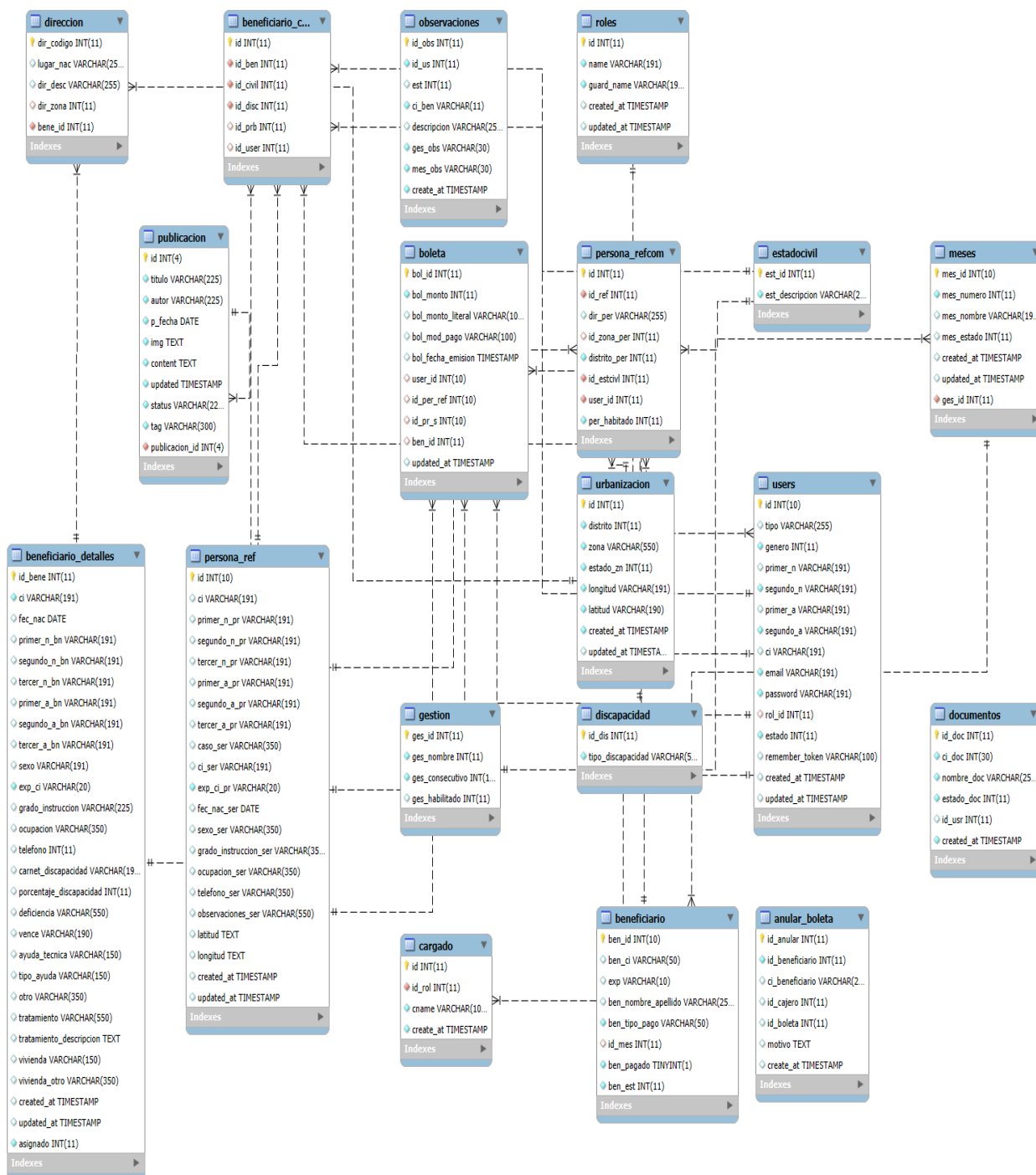


3.3.2.3 BASE DE DATOS

Los requerimientos funcionales permiten modelar una base de datos sólida.

Figura 8

Modelo de Base de Datos del sistema



3.4 GAME

3.4.1 SPRINT 1: ACCESO AL SISTEMA

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 7

Planificación del Sprint 1

SPRINT 1	
FECHA DE INICIO:	18/09/2023
FECHA DE FIN:	27/09/2023
REVISIÓN DE LOS AVANCES:	1 semana y 3 días
TAREAS DE DESARROLLO:	HU01: Creación de la BD HU02: Acceso al sistema

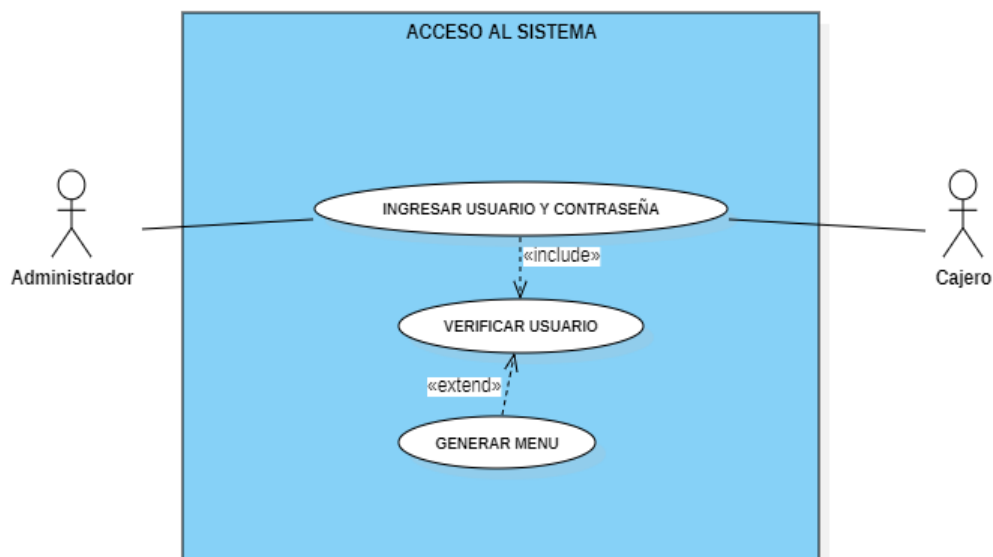
Nota. Fechas de inicio y fin del desarrollo (Sprint 1)

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 9

Diagrama casos de uso acceso al sistema

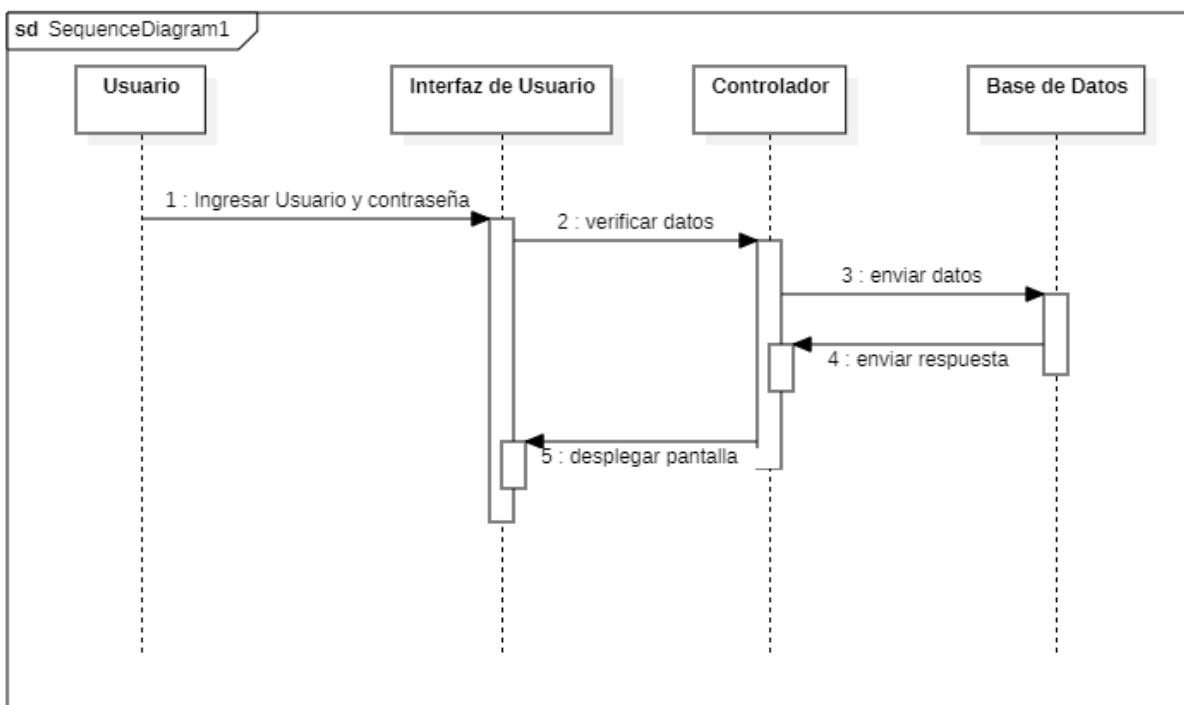


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 10

Diagrama de secuencia acceso al sistema



C. Sprint Backlog

Figura 11

Sprint-Backlog Acceso al sistema

Sprint 1			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Iniciar el proyecto con Codeigniter 3	Jose Luis Laura Mamani	1
T2	Crear la Base de Datos	Jose Luis Laura Mamani	5
T3	Crear la interfaz para inicio de sesión	Jose Luis Laura Mamani	3
T4	Realizar las pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			10

Desarrollo: Listamos las historias de usuario que describen los subprocesos:

Tabla 8*Historia de Usuario: Crear BD*

Número de historia: 01	Usuario: Administrador
Nombre de historia: Creación de la BD	
Prioridad: ALTA	Riesgo en desarrollo: ALTA
Puntaje de H.U: 8	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se modela y crea la base de datos para el almacenamiento de la información en tablas relacionadas.	
Observaciones: NINGUNA	

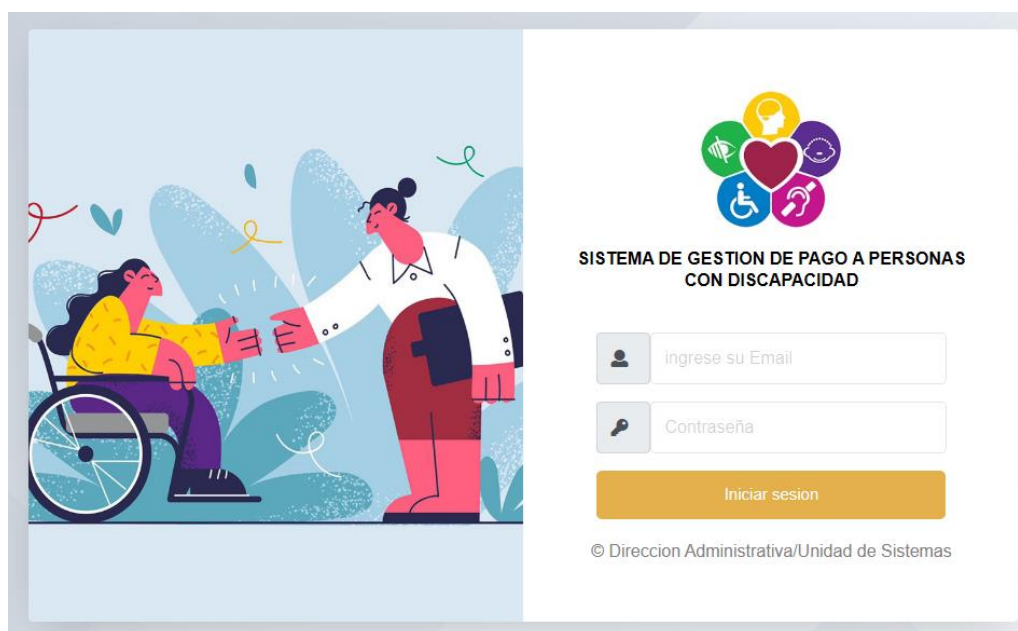
Tabla 9*Historia de Usuario: Acceso al sistema*

Número de historia: 02	Usuario: Administrador, Responsable, Cajero
Nombre de historia: Acceso al sistema	
Prioridad: ALTA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 4	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se crea la página de login, para autenticar a los usuarios según sus roles y otorgarle los permisos necesarios.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 12

Interfaz gráfica Sprint 1



3.4.2 SPRINT 2: REALIZACIÓN DE PAGOS Y REGISTROS

Planeación: Planificación de tareas del Sprint 2.

Tabla 10

Planificación del Sprint 2

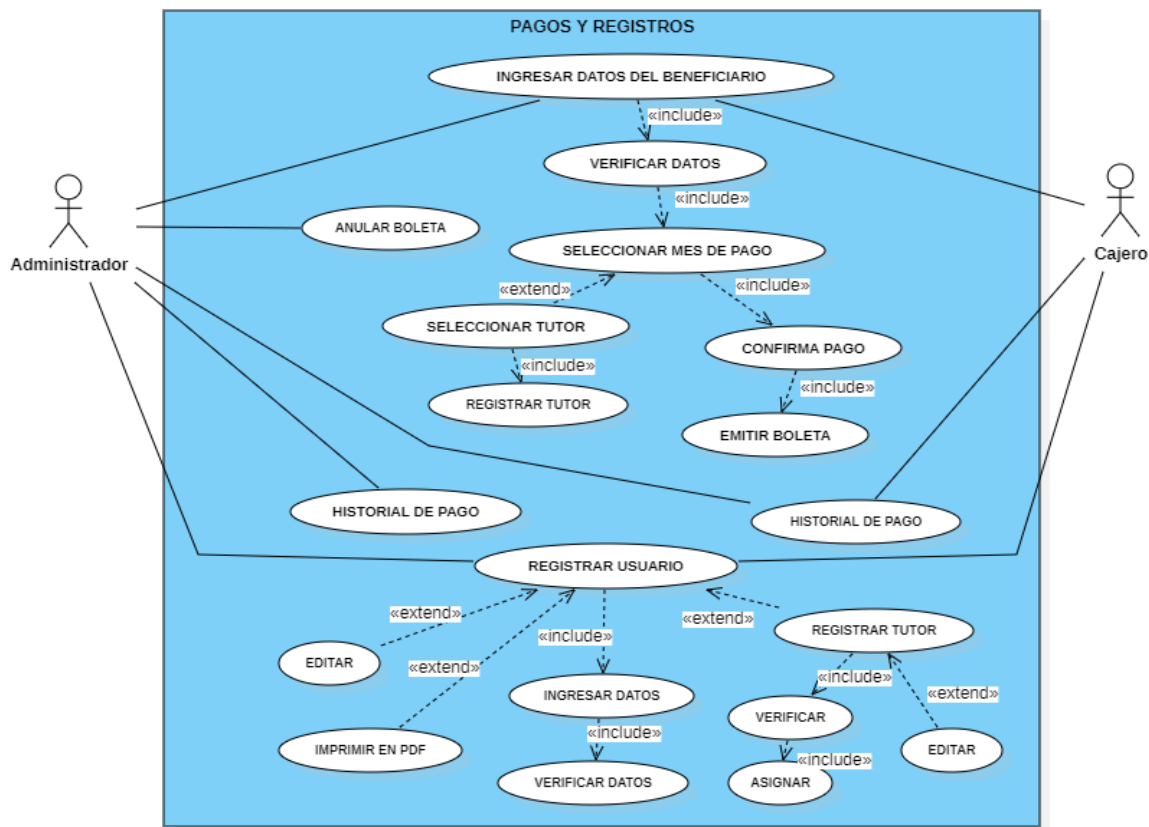
SPRINT 2	
FECHA DE INICIO:	28/09/2023
FECHA DE FIN:	09/10/2023
REVISIÓN DE LOS AVANCES:	1 semana y 5 días
TAREAS DE DESARROLLO:	HU06: Módulo pago a beneficiarios HU07: Módulo historial de pagos HU08: Módulo registro de datos

A. Diagrama de casos de uso

Diagrama de casos de uso de los procesos de pagos y de registro de beneficiarios:

Figura 13

Diagrama casos de uso realización de pagos y registros



B. Diagrama de secuencia

Muestra los procesos necesarios para llevar a cabo el pago y los registros:

Figura 14

Diagrama de secuencia registro de beneficiario

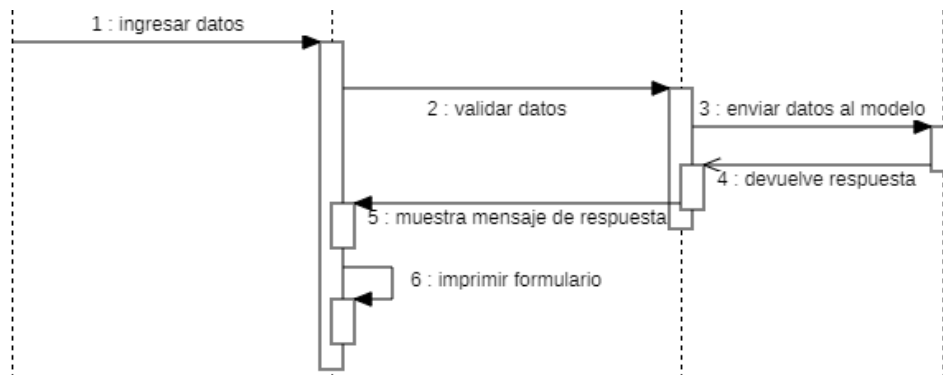
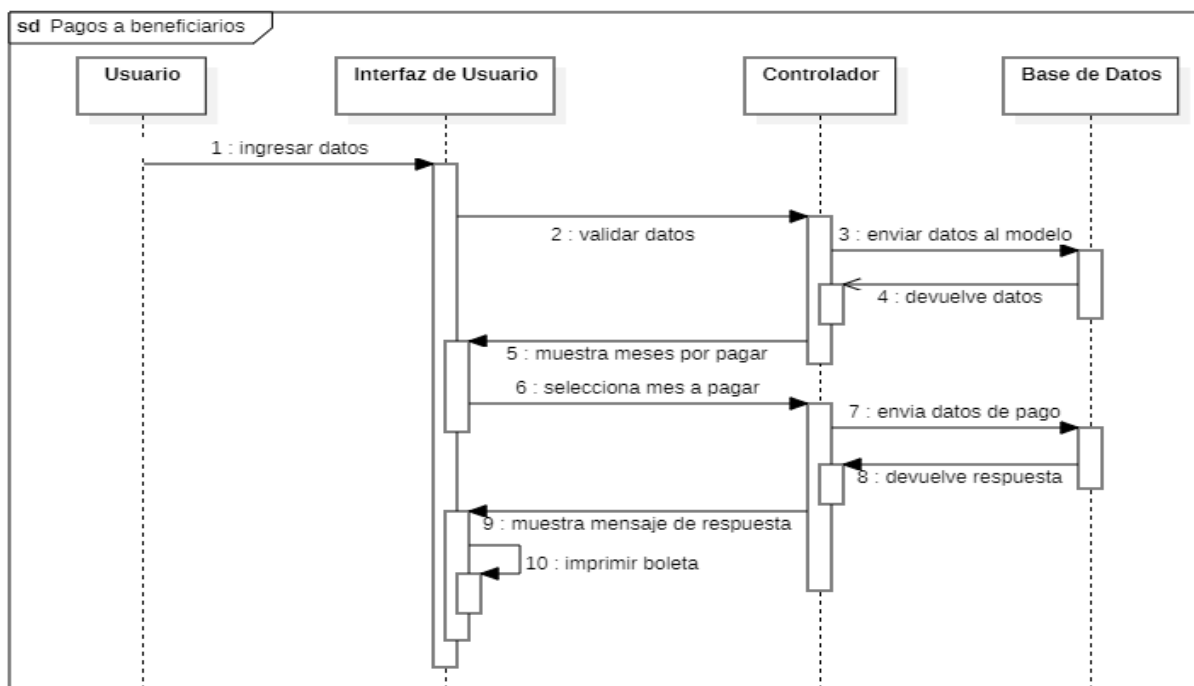


Figura 15

Diagrama de secuencia realización de pagos



C. Sprint Backlog

Figura 16

Sprint-Backlog Pagos y Registros

Sprint 2			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz de pago a beneficiarios	Jose Luis Laura Mamani	2
T2	Desarrollar lógica para pagar beneficiario	Jose Luis Laura Mamani	1
T3	Desarrollar lógica para anular e imprimir boleta	Jose Luis Laura Mamani	1
T4	Desarrollar opción de observaciones	Jose Luis Laura Mamani	1
T5	Crear la interfaz de revisar el historial de pagos	Jose Luis Laura Mamani	1
T6	Desarrollar lógica para revisar el historial	Jose Luis Laura Mamani	1
T7	Crear la interfaz de registro, impresión de formulario de datos y asignación de tutor.	Jose Luis Laura Mamani	2
T8	Desarrollar lógica para el guardado de datos	Jose Luis Laura Mamani	2
T9	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			12

Desarrollo: El Sprint Backlog nos permite generar historias de usuarios para el Sprint2:

Tabla 11*Historia de Usuario: Módulo de pagos*

Número de historia: 06	Usuario: Administrador, Responsable, Cajero
Nombre de historia: Módulo pago a beneficiarios	
Prioridad: ALTA	Riesgo en desarrollo: ALTA
Puntaje de H.U: 7	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Los usuarios podrán acceder a este módulo para realizar el pago del beneficiario.	
Observaciones: Incluir opción para imprimir boleta, anular boleta y añadir una observación.	

Tabla 12*Historia de Usuario: Módulo del Historial de pago*

Número de historia: 07	Usuario: Administrador, Responsable, Cajero
Nombre de historia: Módulo Historial de Pagos	
Prioridad: ALTA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 5	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Los usuarios tendrán acceso a revisar su historial de pagos del día o de días pasados.	
Observaciones: El Administrador y Responsable deben tener la opción de revisar el historial de pago de los demás cajeros.	

Tabla 13

Historia de Usuario: Módulo de Registro

Número de historia: 08	Usuario: Administrador, Encargado, Cajero
Nombre de historia: Módulo Registro de Datos	
Prioridad: ALTA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 5	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Acceso a registrar o editar los datos a nuevos beneficiarios habilitados, también el registro y asignación a un tutor si el caso amerita.	
Observaciones: Tener la opción de imprimir los datos en un formulario.	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 17

Interfaz gráfica Sprint 2 Tarea 2

Nro. Docum.	Apellidos y Nombres	Tipo de Pago	Mes	Gestion	Pagar Beneficiario	Ver Boleta de Pago	Anular Boleta de Pago	Adicionar Per.Ref.	Editar Per.Ref.	Obs.
14157198	GUTIERREZ QUISBERT LUZ RENATA	COBRO REGULAR	DICIEMBRE	2023	Beneficiario Pagado	Boleta de Pago	Anular Boleta	+		Adicionar Observación
14157198	GUTIERREZ QUISBERT LUZ RENATA	COBRO REGULAR	ENERO	2024	Beneficiario Pagado	Boleta de Pago	Anular Boleta	+		Adicionar Observación

Nota: Interfaz gráfica pago a los beneficiarios

Figura 18

Interfaz gráfica Sprint 2 Tarea 6

Historial de pagos [Menu Principal](#) / Historial de pagos

[Arqueo de Caja Diario](#)

Mi historial de Pagos Historial de Pagos Cajeros

17 Elija Fecha de Pago:

dd/mm/aaaa

[Buscar](#)

Nro.	Fecha de Pago	Nro. Docum. y Exp.	Apellidos y Nombres	Tipo de Pago	Monto (Bs.)	Modalidad de Pago	MES	GESTION	Nro. Boleta
SIN RESULTADOS									

Nota: Interfaz gráfica revisar historial de pago

Figura 19

Interfaz gráfica Sprint 2 Tarea 7

Completar datos del Beneficiario [Menu Principal](#) / Datos del Beneficiario

[Regresar Pago Discapacidad](#)

[IMPRIMIR FORMULARIO](#) [Persona de Referencia \(TUTOR\)](#) Nro. documento [Buscar](#)

Datos Beneficiario (Complemento)

Datos Registrados

Nro. Documento (Sin extension)

Datos Personales

Primer Nombre (obligatorio)
 Segundo Nombre OPCIONAL
 Tercer Nombre OPCIONAL
 Primer Apellido (obligatorio)
 Segundo Apellido OPCIONAL
 Tercer Apellido OPCIONAL

Fecha de nacimiento (obligatorio)
 Lugar de Nacimiento (obligatorio)
 C.I. Expedido
 Sexo (obligatorio)

Datos

Nota: Interfaz gráfica registrar o editar beneficiario

3.4.3 SPRINT 3: REPORTEES

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 14

Planificación del Sprint 3

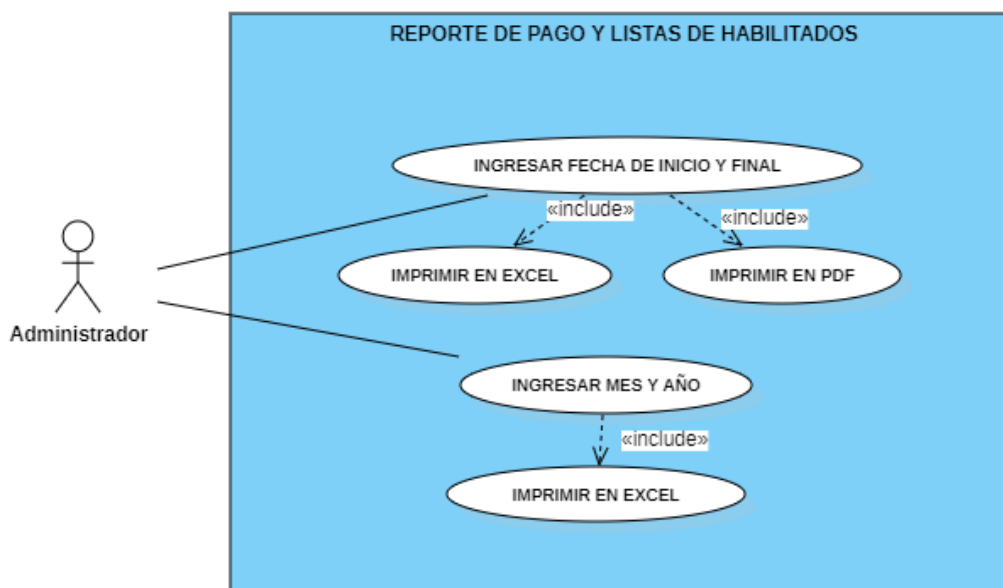
SPRINT 3	
FECHA DE INICIO:	10/10/2023
FECHA DE FIN:	19/10/2023
REVISIÓN DE LOS AVANCES:	1 semana y 3 días
TAREAS DE DESARROLLO:	HU13: Módulo de reportes

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 20

Diagrama casos de uso reportes

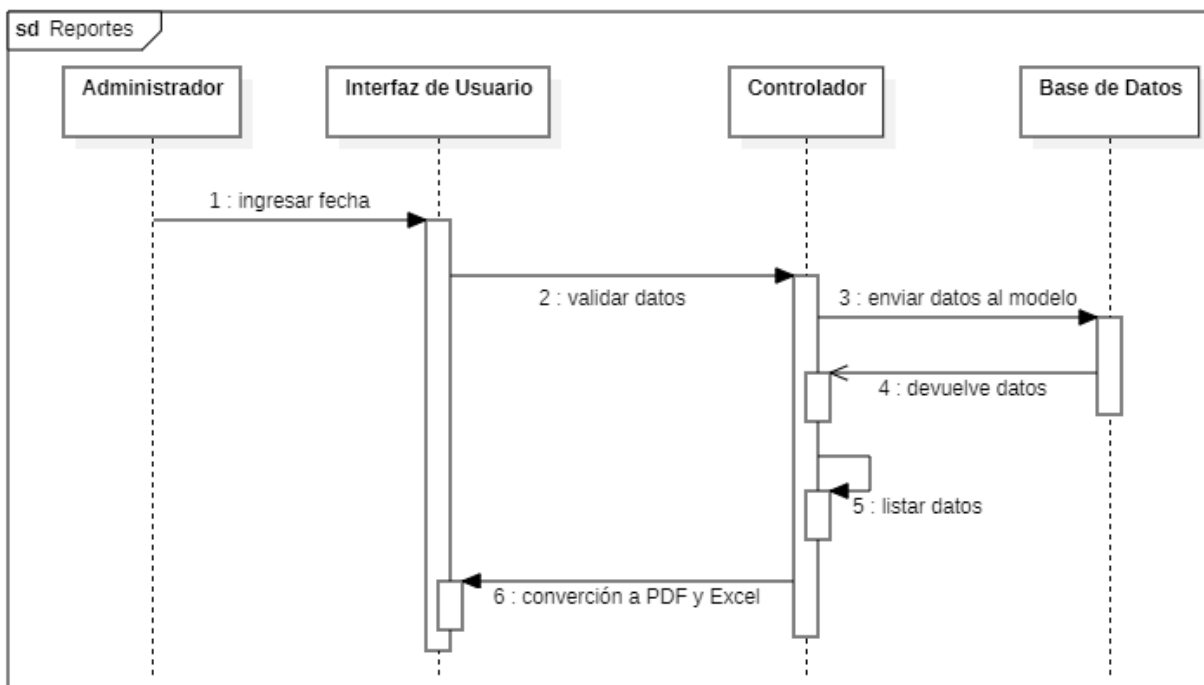


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 21

Diagrama de secuencia reportes



C. Sprint Backlog

Figura 22

Sprint-Backlog Reportes

Sprint 3			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz del módulo de reportes	Jose Luis Laura Mamani	2
T2	Desarrollar lógica para imprimir reporte de pagos en formato PDF y Excel	Jose Luis Laura Mamani	5
T3	Desarrollar lógica imprimir lista de habilitados en formato Excel, reportes fallecidos y tutores en PDF.	Jose Luis Laura Mamani	2
T4	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			10

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 15

Historia de Usuario: Reportes

Número de historia: 13	Usuario: Administrador, Responsable
Nombre de historia: Módulo Reportes	
Prioridad: ALTA	Riesgo en desarrollo: ALTA
Puntaje de H.U: 8	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Generar reportes, pagos del mes, registros de pagos, fallecidos y reporte generado de beneficiarios habilitados para el envío a la entidad bancaria.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 23

Interfaz gráfica Sprint 3



3.4.4 SPRINT 4: DIGITALIZACIÓN Y CONTROL

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 16

Planificación del Sprint 4

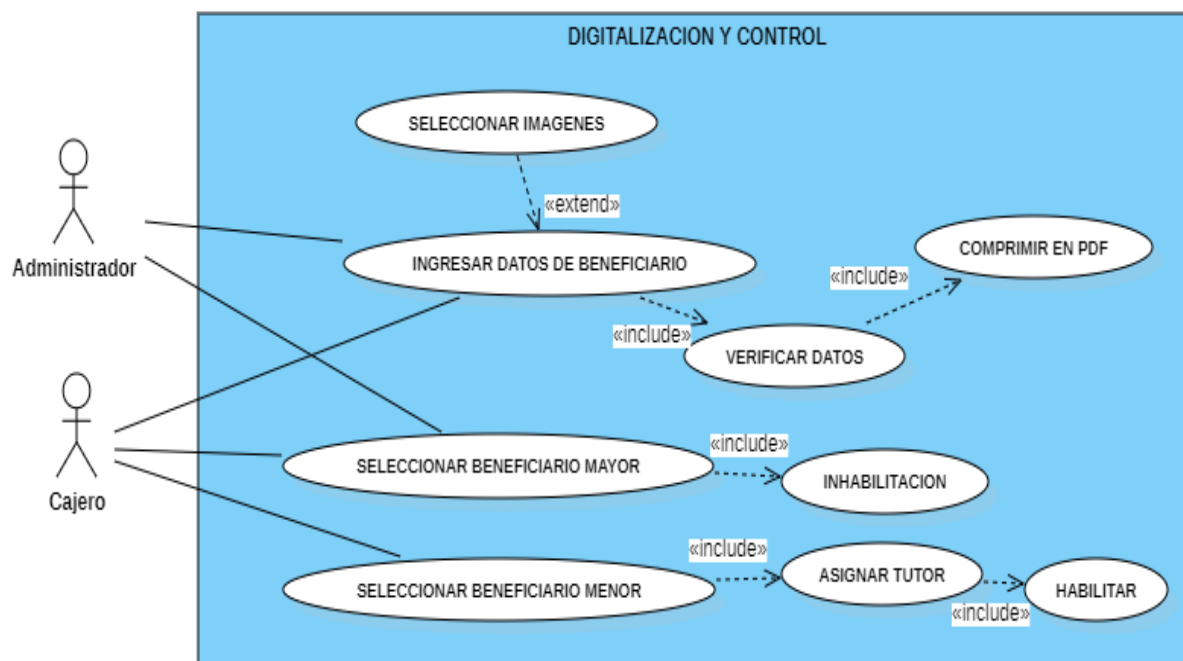
SPRINT 4	
FECHA DE INICIO:	20/10/2023
FECHA DE FIN:	30/10/2023
REVISIÓN DE LOS AVANCES:	1 semana y 3 días
TAREAS DE DESARROLLO:	HU11: Digitalizar documentos HU09: Módulo de beneficiarios observados

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 24

Diagrama casos de uso digitalización y control

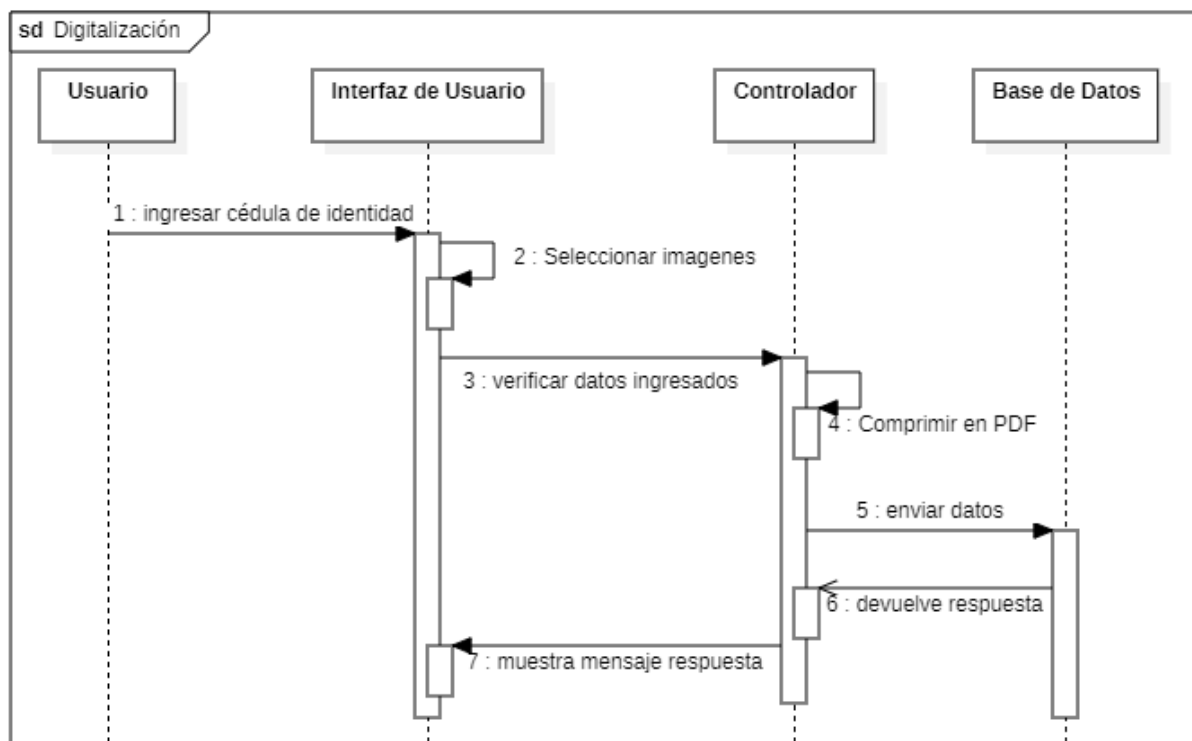


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 25

Diagrama de secuencia digitalización



C. Sprint Backlog

Figura 26

Sprint-Backlog Digitalización e inhabilitación

Sprint 4			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz de tabla beneficiarios observados	Jose Luis Laura Mamani	2
T2	Desarrollar lógica para listar observados: menores, mayores y fallecidos	Jose Luis Laura Mamani	2
T3	Desarrollar lógica para añadir un tutor de referencia	Jose Luis Laura Mamani	1
T4	Crear la interfaz de digitalizar documentos	Jose Luis Laura Mamani	2
T5	Desarrollar el guardado y conversión de imágenes a un archivo PDF	Jose Luis Laura Mamani	2
T6	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			10

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 17*Historia de Usuario: Digitalizar documentos*

Número de historia: 11	Usuario: Administrador, Encargado, Cajero
Nombre de historia: Digitalizar Documentos	
Prioridad: MEDIA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 8	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Digitalizar y subir los documentos de los beneficiarios, como ser su certificado de nacimiento, carnet de discapacidad y otros, para tener una constancia del documento original.	
Observaciones: NINGUNA	

Tabla 18*Historia de Usuario: Observados*

Número de historia: 09	Usuario: Administrador, Encargado, Cajero
Nombre de historia: Módulo Beneficiarios Observados	
Prioridad: MEDIA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 5	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Acceso a la inhabilitación del beneficiario, que sea menor de edad asignar un tutor para realizar el cobro, en caso de mayores de 60 años deberán ser inhabilitados para el cobro e inhabilitar fallecidos.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 27

Interfaz gráfica Sprint 4 Tarea 2

HABILITAR - INHABILITAR BENEFICIARIO [Menu Principal](#) / Beneficiarios Observados

i Asigne un tutor si el beneficiario es MENOR de edad

Beneficiario menor de Edad Beneficiario mayor a 60 años Beneficiarios Inhabilitados (Fallecido / Obsevado)

Mostrar 10 Entradas Buscar:

C.I.	Nombre y Apellido	Fecha-Nacimiento	Edad (Años)	Estado	Acciones (Asignar Tutor)
10008975	MARTHA GABRIEL CHOQUE	2019-07-29	5	OBSERVADO	+ Añadir + Asignar
10022828	TELMO CRUZ MAMANI	2020-09-16	4	OBSERVADO	+ Añadir + Asignar
10062579	DANIEL BALTAZAR CHOQUE	2021-09-15	3	OBSERVADO	+ Añadir + Asignar
10076027	PAULINA MURGA ZAMBRANA	2019-12-05	4	OBSERVADO	+ Añadir + Asignar
10076030	MONICA PATON MARTHA	2019-12-05	4	OBSERVADO	+ Añadir + Asignar

Figura 28

Interfaz gráfica Sprint 4 Tarea 4

Registrar Documentos [Menu Principal](#) / Registrar Documentos

[Regresar Pago Discapacidad](#)

Beneficiario:

Datos de búsqueda

Cédula de Identidad: Nombre Completo:

Documentación Reciente:

Archivos

Pulsa o Arrastra aquí las imágenes para subirlas

3.4.5 SPRINT 5: CRUD DEL SISTEMA

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 19

Planificación del Sprint 5

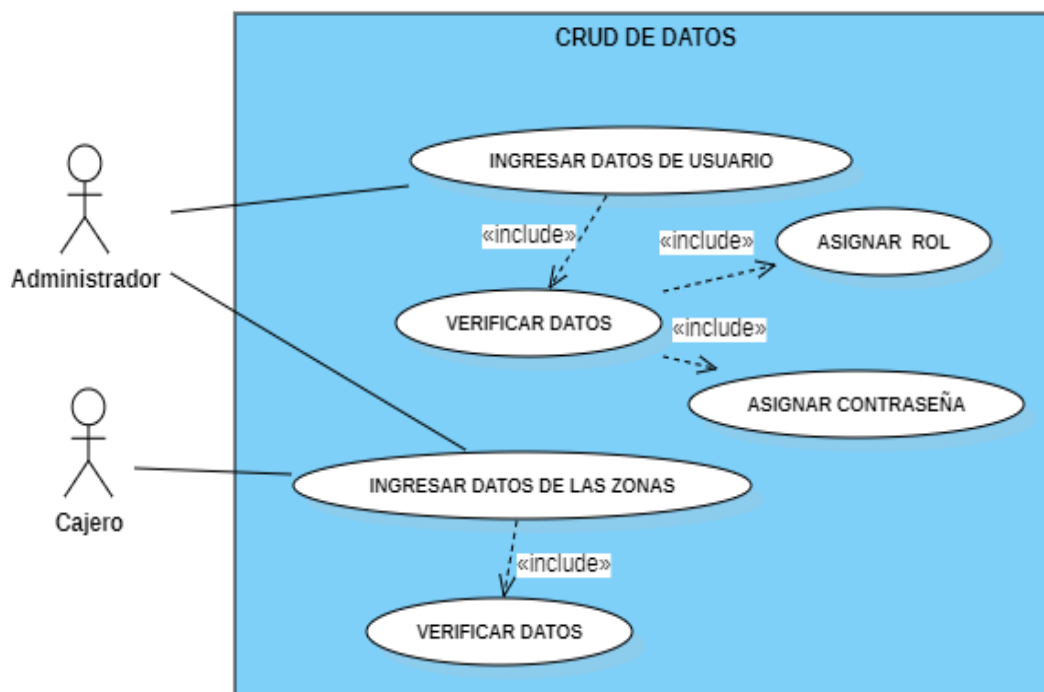
SPRINT 5	
FECHA DE INICIO:	31/10/2023
FECHA DE FIN:	06/11/2023
REVISIÓN DE LOS AVANCES:	1 semana
TAREAS DE DESARROLLO:	HU03: CRUD para usuarios HU12: CRUD para zonas

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 29

Diagrama casos de uso CRUD de datos

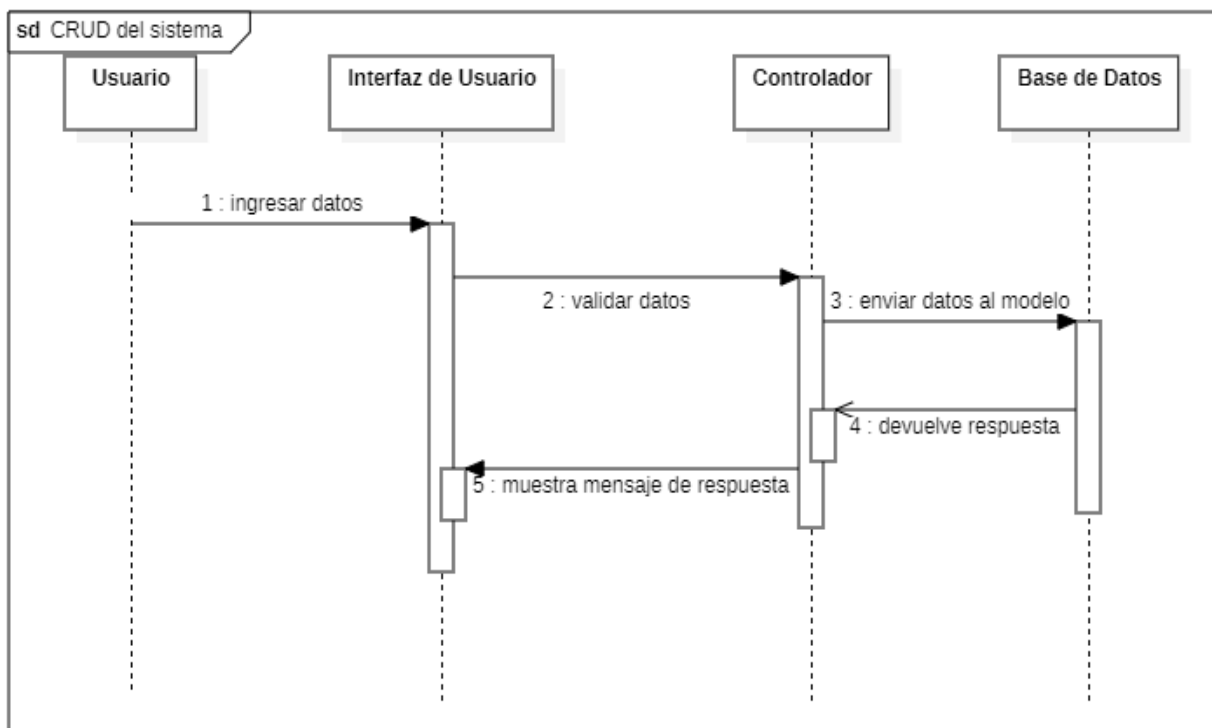


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 30

Diagrama de secuencia CRUD de datos



C. Sprint Backlog

Figura 31

Sprint-Backlog CRUD de datos

Sprint 5			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz del CRUD usuarios	Jose Luis Laura Mamani	2
T2	Desarrollar lógica del CRUD usuarios	Jose Luis Laura Mamani	1
T3	Crear la interfaz del CRUD zonas y distritos	Jose Luis Laura Mamani	2
T4	Desarrollar lógica del CRUD zonas	Jose Luis Laura Mamani	1
T5	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			7

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 20

Historia de Usuario: CRUD Usuarios

Número de historia: 03	Usuario: Administrador
Nombre de historia: CRUD para usuarios	
Prioridad: MEDIA	Riesgo en desarrollo: MEDIA
Puntaje de H.U: 3	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Tener acceso a este módulo, para poder agregar, editar, actualizar, inhabilitar y asignar roles a los nuevos usuarios.	
Observaciones: NINGUNA	

Tabla 21

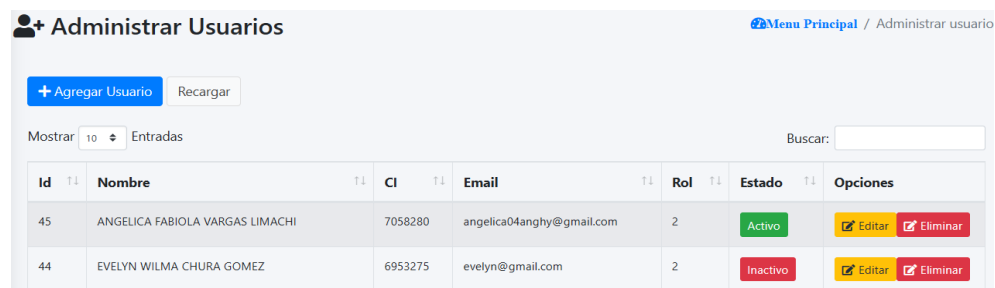
Historia de Usuario: CRUD Zonas

Número de historia: 12	Usuario: Administrador, Cajero
Nombre de historia: CRUD Zonas	
Prioridad: MEDIA	Riesgo en desarrollo: BAJA
Puntaje de H.U: 4	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Acceso a la creación y edición de nuevos distritos y zonas.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 32

Interfaz gráfica Sprint 5 Tarea 2



+ Administrar Usuarios Menu Principal / Administrar usuario

[+ Agregar Usuario](#)

Mostrar Entradas Buscar:

Id	Nombre	CI	Email	Rol	Estado	Opciones
45	ANGELICA FABIOLA VARGAS LIMACHI	7058280	angelica04anghy@gmail.com	2	Activo	Editar Eliminar
44	EVELYN WILMA CHURA GOMEZ	6953275	evelyn@gmail.com	2	Inactivo	Editar Eliminar

Figura 33

Interfaz gráfica Sprint 5 Tarea 3

Estado	Distrito	Zona	Latitud	Longitud	Opciones
Activo	8	SAN AGUSTIN	-68.2031052435239	-16.5760989060712	Editar Cambiar
Activo	1	VILLA BOLIVAR YKK	-68.1706640497398	-16.5326061991032	Editar Cambiar
Activo	6	FERROPETROL O BORIS BANZER	-68.166655837558	-16.5027463107444	Editar Cambiar
Activo	6	FERROPETROL O BORIS BANZER	-68.1642830867123	-16.5050136091931	Editar Cambiar
Activo	7	URB. LOTEO SAN LORENZO	-68.2780276299525	-16.4745930663652	Editar Cambiar
Activo	7	EL PROGRESO SECTOR 16 DE JULIO	-68.2496044272869	-16.4788323331297	Editar Cambiar

3.4.6 SPRINT 6: CARGADO DE DATOS

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 22

Planificación del Sprint 6

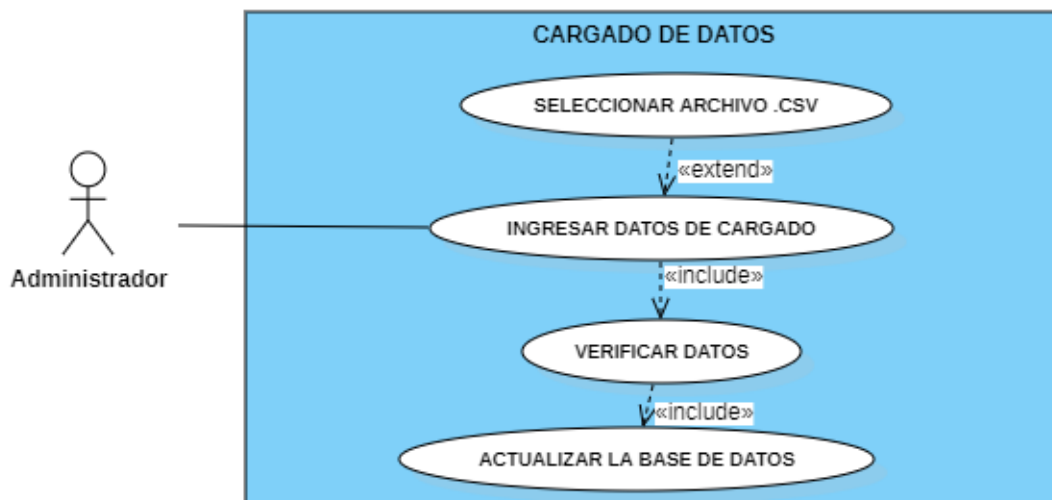
SPRINT 6	
FECHA DE INICIO:	07/11/2023
FECHA DE FIN:	17/11/2023
REVISIÓN DE LOS AVANCES:	1 semana y 3 días
TAREAS DE DESARROLLO:	HU14: Módulo de cargado de datos

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 34

Diagrama casos de uso cargado de datos

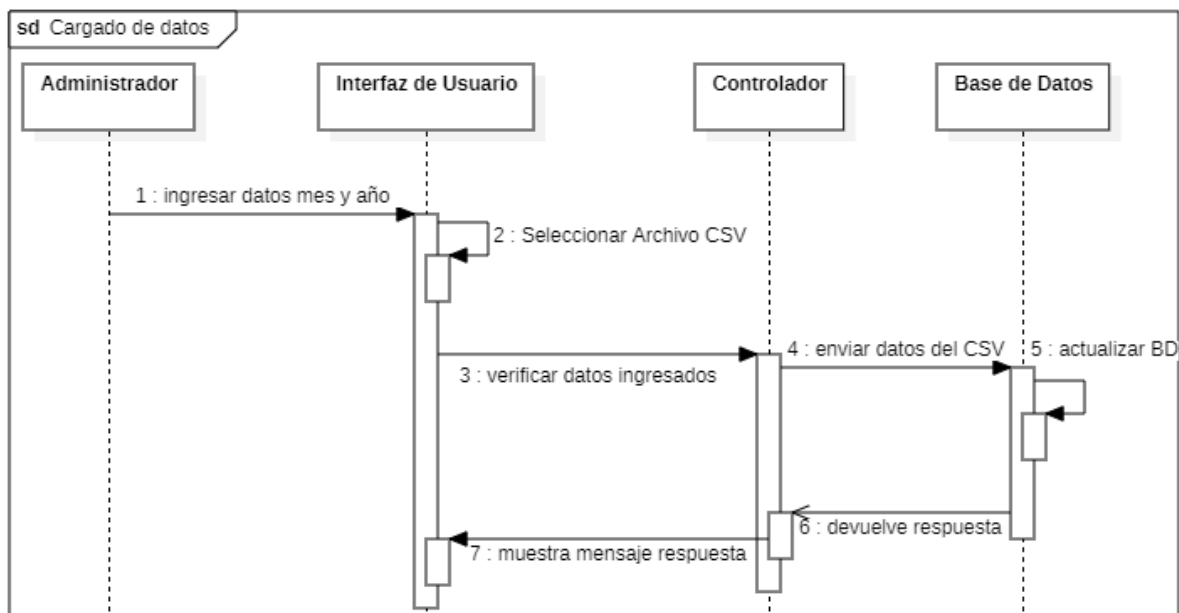


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 35

Diagrama de secuencia Cargado de datos



C. Sprint Backlog

Figura 36

Sprint-Backlog Cargado de datos

Sprint 6			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz del cargado de datos	Jose Luis Laura Mamani	2
T2	Desarrollar lógica importar lista de datos a la BD	Jose Luis Laura Mamani	5
T3	Implementar guias de ayuda como imagen y video	Jose Luis Laura Mamani	2
T4	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			10

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 23

Historia de Usuario: Cargado de datos

Número de historia: 14	Usuario: Administrador, Encargado
Nombre de historia: Módulo de Cargado de Datos	
Prioridad: MEDIA	Riesgo en desarrollo: ALTA
Puntaje de H.U: 8	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Importar la lista de habilitados del mes a la base de datos.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 37

Interfaz gráfica Sprint 6

✓ Seguir las instrucciones para realizar el cargado de datos.

COMO REALIZAR LA CARGA DE DATOS AL SISTEMA

IMPORTAR DATOS
Nuevo Mes

📅 Seleccione el mes: Seleccione el año:

Seleccione archivo CSV

Seleccionar archivo Ningún archivo seleccionado

Transforme el archivo .xls a .csv
Siga los siguientes pasos o reproduzca el tutorial

3.4.7 SPRINT 7: REVISIÓN DE DATOS

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 24

Planificación del Sprint 7

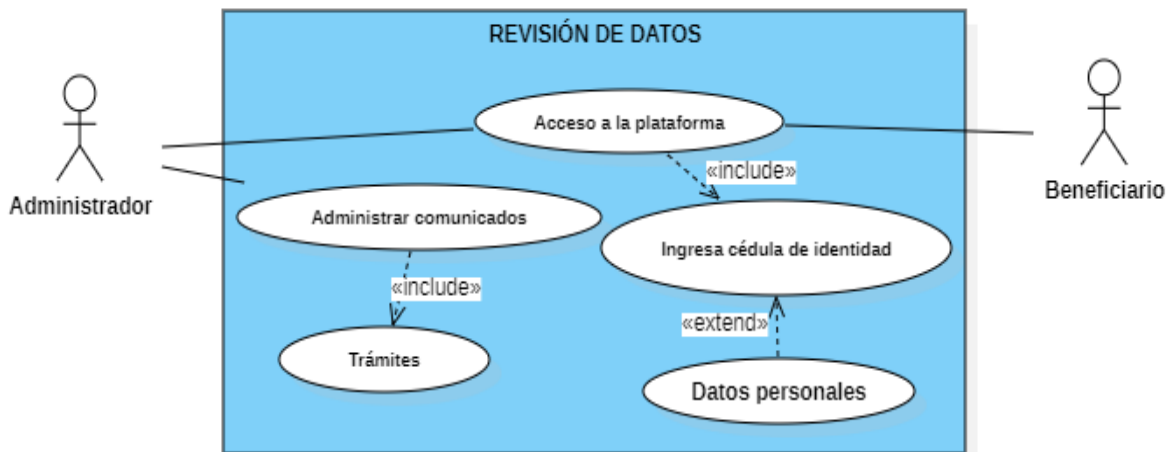
SPRINT 7	
FECHA DE INICIO:	20/11/2023
FECHA DE FIN:	04/12/2023
REVISIÓN DE LOS AVANCES:	2 semanas
TAREAS DE DESARROLLO:	HU15: Módulo verificación de datos

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 38

Diagrama casos de uso acceso consulta de información

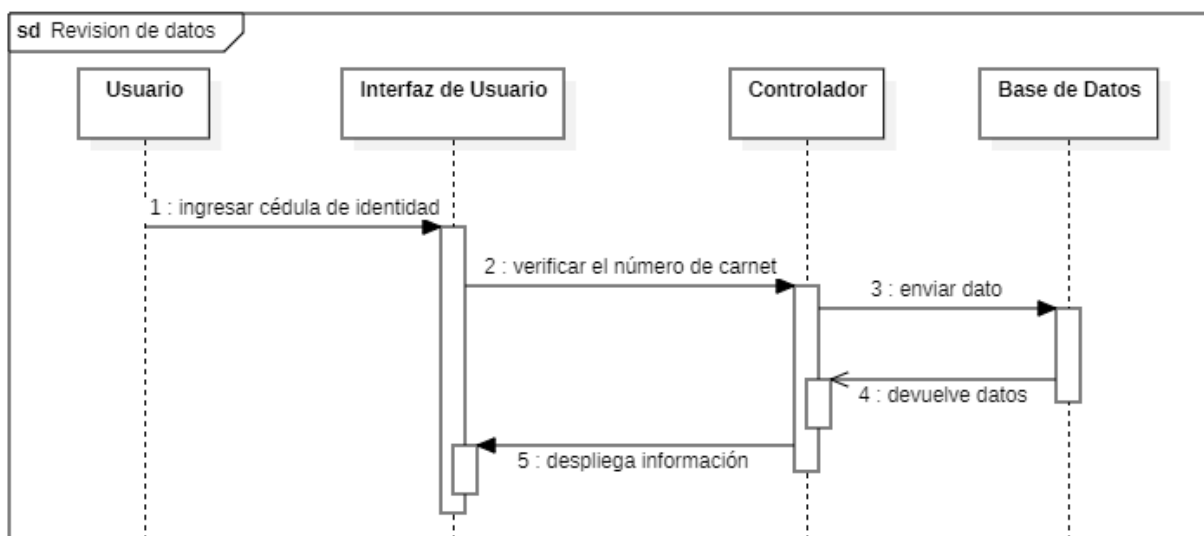


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 39

Diagrama de secuencia Consulta de información



C. Sprint Backlog

Figura 40*Sprint-Backlog Consulta de información*

Sprint 7			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz del consulta de datos	Jose Luis Laura Mamani	2
T2	Implementar acceso con el carnet de identidad	Jose Luis Laura Mamani	1
T2	CRUD administrados de comunicados	Jose Luis Laura Mamani	2
T3	Desarrollar lógica de obtención de datos	Jose Luis Laura Mamani	1
T4	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			7

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 25*Historia de Usuario: Verificación*

Número de historia: 15	Usuario: General
Nombre de historia: Verificación de Datos	
Prioridad: MEDIA	Riesgo en desarrollo: BAJA
Puntaje de H.U: 6	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se desarrolla un módulo para los beneficiarios, en el que ellos podrán acceder con el ingreso de su cédula de identidad, y revisar sus datos e historial de cobro, además de revisar comunicados y trámites realizados en la unidad.	
Observaciones: El beneficiario debe tener un ingreso al sistema con su número de cédula de identidad para el acceso a su información.	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 41

Interfaz gráfica Sprint 7



3.4.8 SPRINT 8: ESTADÍSTICAS

Planeación: Se puntualizan las tareas necesarias para llevar a cabo este Sprint.

Tabla 26

Planificación del Sprint 8

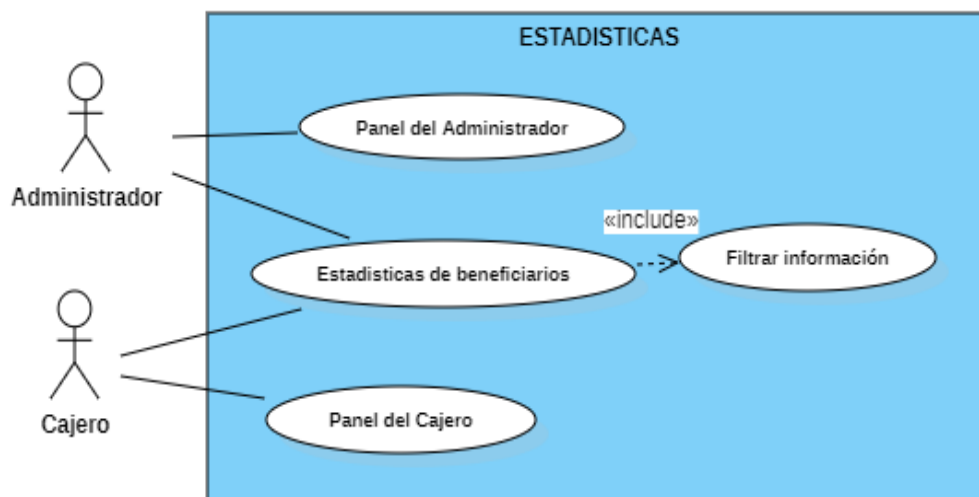
SPRINT 8	
FECHA DE INICIO:	05/12/2023
FECHA DE FIN:	12/12/2023
REVISIÓN DE LOS AVANCES:	1 semana
TAREAS DE DESARROLLO:	HU04: Panel Administrador - Responsable HU05: Panel Cajero HU10: Módulo búsqueda de datos

A. Diagrama de casos de uso

Diagrama de casos de uso Acceso al sistema de los usuarios:

Figura 42

Diagrama casos de uso Estadísticas

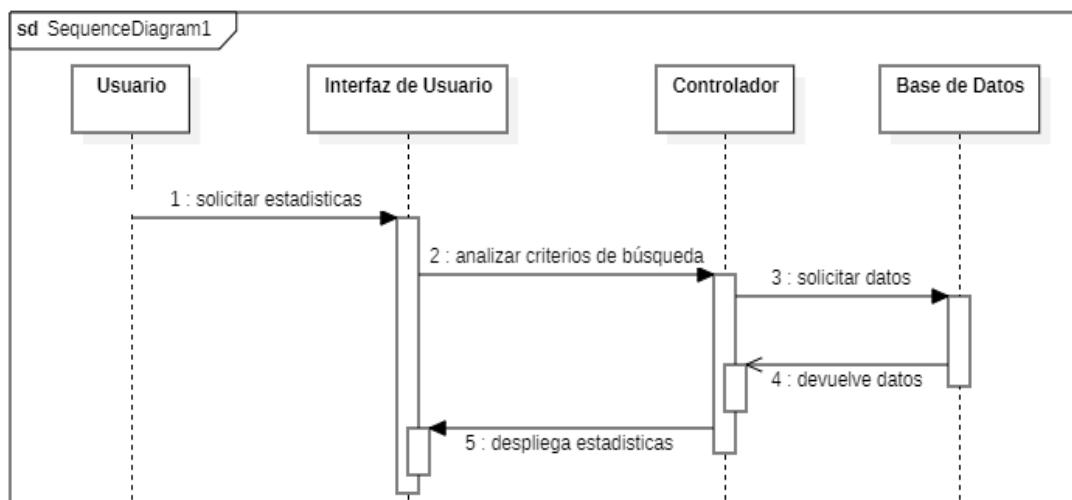


B. Diagrama de secuencia

Muestra los procesos de ingreso al sistema por el cual los usuarios atraviesan.

Figura 43

Diagrama de secuencia Estadísticas



C. Sprint Backlog

Figura 44

Sprint-Backlog Estadísticas

Sprint 8			
Nro Tarea	Descripción	Responsable	Esfuerzo en días
T1	Crear la interfaz del panel de Administrador	Jose Luis Laura Mamani	1
T2	Crear la interfaz del panel de Cajero	Jose Luis Laura Mamani	1
T3	Crear la interfaz del módulo de búsquedas	Jose Luis Laura Mamani	2
T4	Desarrollar lógica de obtención de datos por búsqueda	Jose Luis Laura Mamani	2
T5	Realizar pruebas de funcionamiento	Jose Luis Laura Mamani	1
Total			7

Desarrollo: El Sprint Backlog permite generar historias de usuario para el proceso.

Tabla 27

Historia de Usuario: Panel Administrador

Número de historia: 04	Usuario: Administrador, Responsable
Nombre de historia: Panel Administrador - Responsable	
Prioridad: BAJA	Riesgo en desarrollo: BAJA
Puntaje de H.U: 2	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se desarrolla el dashboard con accesos directos y datos estadísticos.	
Observaciones: NINGUNA	

Tabla 28

Historia de Usuario: Panel Cajero

Número de historia: 05	Usuario: Cajero
Nombre de historia: Panel Cajero	
Prioridad: BAJA	Riesgo en desarrollo: BAJA
Puntaje de H.U: 2	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se desarrolla el dashboard a los módulos que puede acceder el cajero.	
Observaciones: NINGUNA	

Tabla 29

Historia de Usuario: Estadísticas

Número de historia: 10	Usuario: Administrador, Encargado, Cajero
Nombre de historia: Módulo Búsqueda de Datos	
Prioridad: BAJA	Riesgo en desarrollo: BAJA
Puntaje de H.U: 2	Iteración asignada: 1
Responsable: Jose Luis Laura Mamani	
Descripción: Se tendrá la opción de consultar los distintos datos de los beneficiarios que requieran en el momento, como números de identificación, nombres, cantidad de beneficiarios por distritos y por edades.	
Observaciones: NINGUNA	

Revisión: El desarrollo del Sprint en base a las historias de usuario presentada en las tablas 6 y 8 genera la interfaz gráfica y la interpretación de órdenes.

Figura 45

Interfaz gráfica Sprint 8



3.5 POST-GAME

3.5.1 PRUEBAS

Las pruebas de funcionalidad del sistema se realizan acorde a los requerimientos que se estableció con el cliente, que fue realizado durante el desarrollo del proyecto al finalizar cada iteración.

3.5.1.1 PRUEBA DE FUNCIONALIDAD

Son las pruebas que se llevan a cabo sobre la interfaz del usuario, pruebas del funcionamiento correcto de cada uno de los módulos del sistema establecidos en las historias de usuario. Se realizaron bajo los siguientes puntos:

- **Prueba unitaria:** se realizó durante el desarrollo de cada módulo del sistema verificando su funcionalidad con la aprobación del Product Owner.
- **Prueba de integración:** esta prueba se realizó después de cada Sprint diseñado y codificado, para su posterior prueba en general.

Tabla 30

Pruebas del software

Iteración	Caso de prueba	Resultado Esperado	Resultado de la prueba
1	Inicio de sesión	de Formulario de inicio de sesión con las medidas de autenticación.	Exitoso
1	Acceso sistema	al El Usuario puede ingresar al sistema con su usuario y contraseña correcta.	Exitoso
1	Administración de usuarios	El Administrador puede gestionar la información del usuario, crear, editar o inhabilitar a los usuarios.	Exitoso

Iteración	Caso de prueba	Resultado Esperado	Resultado
1	Pagar Beneficiarios	El Usuario puede realizar la acción de pagar, generar boleta de pago, anular boleta (Administrador), adicionar y asignar tutor.	Exitoso
2	Historial de pagos	El Usuario puede observar su historial de pago, generar su arqueo diario, el Administrador puede observar el historial de todos los cajeros.	Exitoso
2	Realizar reportes	El Administrador puede generar reportes del pago realizado durante el periodo establecido, generar listado de beneficiarios habilitados y fallecidos.	Exitoso
2	Registrar Beneficiarios	Se puede ingresar nuevos beneficiarios, modificar, revisar y verificar sus datos. Además de la acción de registrar a un tutor y asignarlo al beneficiario si el caso amerita.	Exitoso
2	Digitalizar Documentos	El sistema permite subir imágenes en formato jpg, png y jpeg que permite digitalizar documentos de los beneficiarios y almacenarlos en un documento PDF para su respectiva revisión.	Exitoso
3	Búsqueda de datos	El Usuario puede realizar la búsqueda de datos, por medio del nombre del beneficiario, nombre del tutor, por medio de su edad, sexo y distrito.	Exitoso
3	Subir documento	El Administrador puede subir un documento con el listado de beneficiarios habilitados del mes, en archivo con formato .csv	Exitoso
3	Cerrar Sesión	El Usuario puede cerrar sesión satisfactoriamente.	Exitoso
3	Consulta de datos	El usuario y el público en general pueden acceder al módulo de consultas, ingresando la cédula de identidad y fecha de nacimiento para así revisar sus datos, historial de cobros, comunicados y trámites.	Exitoso

En la tabla 30 se muestra las pruebas unitarias funcionales las cuales resultaron exitosas, cumpliendo con los requerimientos esperados por el usuario.

3.5.1.2 PRUEBAS NO FUNCIONALES

Son las pruebas no requeridas por el usuario final.

- **Prueba de carga y estrés:** Con JMeter observó la cantidad de usuarios que puedan conectarse al mismo tiempo y cuantas peticiones sea capaz de soportar.
- **Prueba de seguridad:** se realizó pruebas en los datos de entrada que evite el envío de caracteres en las peticiones, tanto en el front-end y back-end.

Figura 46

Prueba de carga y estrés con JMeter

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
490	15:23:17.828	USUARIO-S 1-473	HTTP Request	8052	✓	56646	261	7919	2044
491	15:23:17.830	USUARIO-S 1-474	HTTP Request	8040	✓	56646	261	7946	2042
492	15:23:17.816	USUARIO-S 1-466	HTTP Request	8048	✓	56646	261	7925	2041
493	15:23:17.794	USUARIO-S 1-454	HTTP Request	8020	✓	56646	261	7942	2063
494	15:23:17.811	USUARIO-S 1-461	HTTP Request	7986	✓	56646	261	7925	2046
495	15:23:17.832	USUARIO-S 1-475	HTTP Request	8348	✓	56646	261	8250	2056
496	15:23:17.848	USUARIO-S 1-482	HTTP Request	8336	✓	56646	261	8258	2040
497	15:23:17.859	USUARIO-S 1-488	HTTP Request	8332	✓	56646	261	8252	2029
498	15:23:17.877	USUARIO-S 1-498	HTTP Request	8342	✓	56646	261	8283	2011
499	15:23:17.863	USUARIO-S 1-490	HTTP Request	8363	✓	56646	261	8287	2025
500	15:23:17.875	USUARIO-S 1-497	HTTP Request	8375	✓	56646	261	8276	2013

Nota. Según las pruebas el servidor soporta hasta 500 u/s conectados con estabilidad.

Figura 47

Prueba de seguridad en el envío de peticiones

BACK

```
$nro_documento = strip_tags(trim($this->input->post('nro_documento')));
```

FRONT

```
function EIniciales(input) {
  input.value = input.value.trim();
  const valor = input.value;
  const patron = /^[a-zA-Z0-9\-\]*$/;
  if (!patron.test(valor)) {
    input.value = valor.slice(0, -1);
  }
}
```

CAPITULO IV

4 METRICAS DE CALIDAD Y SEGURIDAD

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PUBLICA DE EL ALTO



4.1 INTRODUCCIÓN

En el desarrollo de este capítulo se describe el uso de la calidad del software y la seguridad del sistema, donde se medirá el nivel de calidad del software mediante la ISO 25010 y seguridad se demuestra por la ISO 27002, las cuales califican la confiabilidad del sistema.

4.2 CALIDAD DE SOFTWARE

Mediante la norma ISO 25010, se efectúa los siguientes análisis:

4.2.1 Funcionalidad

Mediante la funcionabilidad se podrá medir la capacidad del software de cumplir con las funciones especificadas por el usuario. Evaluaremos una valoración mediante el cálculo del punto función en base a la evaluación de un conjunto de características y capacidades del sistema.

Número de entradas de usuario: Se lista las entradas que proporciona datos al sistema.

Tabla 31

Lista de entradas del usuario al sistema

Nro.	Entradas de Usuario
1	Ingreso al sistema
2	Registro de usuarios
3	Registro de beneficiarios del bono discapacidad
4	Registro de beneficiarios habilitados
5	Registro de pagos

Número de salidas de usuario: Se refiere a la información elaborada por el sistema para ser mostrada al usuario como ser informes y demás.

Tabla 32

Lista de salidas de usuario del sistema

Nro	Salidas de Usuario
1	Información del usuario
2	Información de los beneficiarios y sus tutores
3	Listado de meses pagados a discapacitados
4	Listados de zonas y distritos
5	Listado de observaciones

Número de peticiones de usuario: Son las entradas interactivas entre el usuario y el sistema, donde la salida es inmediata.

Tabla 33

Lista de peticiones del usuario al sistema

Nro	Peticiones de Usuario
1	Autenticación de usuario
2	Informe de Beneficiarios habilitados
3	Detalles de pagos
4	Reportes de pagos

Numero de archivos: se cuenta cada archivo maestro lógico, como las tablas de la base de datos.

Tabla 34*Lista de archivos Lógicos del sistema*

Nro	Archivos Lógicos
1	Usuario
2	Beneficiarios detalles
3	Personas de Referencia
4	Boletas
5	Zonas
6	Meses
7	Gestión
8	Archivos

Numero de interfaces externas: Se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir información a otro sistema.

Tabla 35*Factor de ponderación de la funcionabilidad*

PARAMETROS DE ENTRADA	CUENTA	FACTOR DE COMPLEJIDAD	RESULTADO
Número de entrada de usuario	5	5	25
Número de salidas de usuario	5	6	30
Número de peticiones de usuario	4	5	20
Número de archivos	8	4	32

PARAMETROS DE ENTRADA	CUENTA	FACTOR DE COMPLEJIDAD	RESULTADO
Número de interfaces externas	1	3	3
Cuenta total			110

Para el ajuste del factor complejidad se asignan valores de acuerdo a los siguientes casos:

Tabla 36

Valores de puntos función

FACTOR	VALOR
Sin Importancia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Fuente: Pressman, 2002

Tabla 37

Cálculo del Factor de Complejidad

FACTORES DE COMPLEJIDAD	Fi
¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
¿Se requiere comunicación de datos para transferir información?	5

FACTORES DE COMPLEJIDAD	Fi
¿Existen funciones de procesamiento distribuido?	4
¿Es crítico el rendimiento?	4
¿El sistema correrá en un entorno operativo existente enormemente utilizado?	5
¿El sistema requiere entradas de datos en línea?	4
¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?	3
¿Se actualiza los archivos maestros de forma interactiva?	4
¿Las entradas, salidas, archivos o consultas son complejos?	4
¿El procesamiento interno es complejo?	3
¿El código se diseña para ser reutilizable?	5
¿La conversión y la instalación se incluye en el diseño?	5
¿El sistema se diseña para instalaciones múltiples en referentes organizaciones?	4
¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?	4
Σ Fi	59

Fuente: Pressman, 2010

La funcionalidad es medida a través del punto función (PF), que proporciona una medida objetiva, cuantitativa y auditable. (Pressman, 2002).

$$PF = Cuenta\ Total * (X + Min(Y) * \Sigma Fi)$$

PF: Medida de funcionalidad

Cuenta total: es la suma de los siguientes datos (N° De Entradas, N° De Salidas, N° De Peticiones, N° De Archivos, N° De Interfaces Externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%

MIN (Y): Error mínimo aceptable al de la complejidad.

ΣFi : Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$

Para calcular el PF usaremos la ecuación:

$$PF = cuenta\ total * [0.65 + (0.01 * \Sigma Fi)]$$

$$PF = 110 * [0.65 + (0.01 * 59)]$$

$$PF = 136.4$$

Ahora se calcula el PF ideal:

$$\Sigma Fi = 5 * 14$$

$$\Sigma Fi = 70$$

$$PF\ ideal = 110 * [0.65 + (0.01 * 70)]$$

$$PF = 148.5$$

Entonces la funcionalidad del sistema es:

$$Funcionalidad = (PF/PF\ ajuste) * 100$$

$$Funcionalidad = (136.4 / 148.5) * 100$$

$$\mathbf{Funcionalidad = 91.8 \%}$$

4.2.2 *Fiabilidad*

Para determinar la fiabilidad del sistema se evaluó durante 8 horas continuas, durante el periodo de evaluación se presentó una falla por razón del internet. Se tiene la siguiente formula:

$$TMEF = (Ht * p) * 100$$

$$TMDR = (Hp * p) * 100$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Ht: Son las horas trabajadas o de marcha durante el periodo de evaluación.

Hp: Son las horas de paro durante el periodo de evaluación.

p: Numero de paros durante el periodo de evaluación.

$$TMEF = (8 / 1) * 100 = 800$$

$$TMDR = (0.15 / 1) * 100 = 15$$

$$Confiabilidad = (TMEF / (TMEF + TMDR)) * 100$$

$$Confiabilidad = (800 / (800 + 15)) * 100 = 98.16$$

Entonces, el sistema tiene una confiabilidad de un 98.16 %

4.2.3 *Usabilidad*

La usabilidad representa facilidad de uso que el usuario final percibirá del sistema. Esta métrica nos muestra el esfuerzo necesario para aprender a manipular el sistema (Pressman, 2022)

Tabla 38*Evaluación de usabilidad*

Nro.	PREGUNTA	Valor 0 – 100
1	¿Es entendible?	94
2	¿La interfaz es agradable a la vista?	92
3	¿Es fácil de manipular?	87
4	¿Contiene toda la información que necesita?	96
5	¿Facilita el trabajo?	92
6	¿La navegación es fluida?	90
Promedio		91.8

Entonces la usabilidad del sistema es del 91.8 lo que indica que 9 de cada 10 usuarios pueden usar el sistema sin dificultad.

4.2.4 Mantenibilidad

Para su evaluación se desarrolló algunas preguntas, estas preguntas son valoradas en porcentaje por el desarrollador del sistema al momento de culminar con el sistema, tomando en cuenta la experiencia y el conocimiento de cualquier desarrollador (Pressman, 2002).

Tabla 39*Evaluación de Mantenibilidad*

FACTOR DE AJUSTE	VALOR
¿Puede ser modificado el sistema?	98%

FACTOR DE AJUSTE	VALOR
¿Permite implementar una modificación específica?	99%
¿Deja identificar las partes que deben ser modificadas?	95%
¿Presenta efectos inesperados como posibles errores?	96%
Total	97%

Entonces, la mantenibilidad del sistema es de un 97% por tanto, el esfuerzo necesario para el mantenimiento del sistema es mínimo.

4.2.5 Portabilidad

La portabilidad es la capacidad que tiene el software para ser trasladado de un entorno a otro, se valoró a cada sub-atributo con el rango de valores de la siguiente tabla:

Tabla 40

Rangos para evaluar la portabilidad

ESCALA	VALOR	ESCALA	VALOR
Excelente	5	Deficiente	2
Bueno	4	Pésimo	1
Aceptable	3		

Tabla 41

Valoración para la Portabilidad

NRO	PREGUNTAS SOBRE PORTABILIDAD	VALOR
1	¿Se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio?	4

NRO	PREGUNTAS SOBRE PORTABILIDAD	VALOR
2	¿El sistema es fácil para ser instalado en un entorno específico o por el usuario final?	5
3	¿El sistema puede coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo?	5
4	¿El sistema puede ser remplazado por otro software del mismo tipo, y para el mismo objetivo?	4
5	¿El sistema cumple con los estándares relacionados a la portabilidad?	4

En base a esto se puede tener una idea cuantitativa de calcular la portabilidad:

$$\text{Portabilidad} = ((4 + 5 + 5 + 4 + 4) / 5) * (100 / 5) = 88$$

Por tanto, se obtiene una portabilidad de un 88 %

4.3 SEGURIDAD DE LA INFORMACIÓN

La seguridad de la información no es más que la protección de los datos y trata de evitar su pérdida y modificaciones no autorizadas. La protección debe garantizar en primer lugar la confidencialidad, integridad y la disponibilidad de la información del sistema, para ello aplicamos los puntos 1, 4, 6 de la norma IEC/ISO 27002.

4.3.1 ORGANIZACIÓN DE LA SEGURIDAD

Solo el administrador del sistema puede habilitar a nuevos usuarios, asignación de roles y de contraseñas, además de realizar copias de respaldo de la base de datos.

4.3.2 CONTROL DE ACCESOS

Seguridad Administrativa: De acuerdo a las funciones del usuario se le asignará un rol con su restricción de acceso a módulos del sistema, solo el administrador tendrá acceso total, además de realizar copias de la base de datos, migración de datos y la desactivación de cuentas.

Desactivación de cuentas: En caso de que el usuario ya no pertenezca a la institución se inhabilitara su usuario permaneciendo sus datos con el fin de registro.

Control de usuario: Para poder tener un control de acceso al sistema evitando el acceso de terceros; el sistema autenticará el usuario mediante sesiones por medio de su nombre de usuario y su contraseña encriptado con el algoritmo BCrypt.

Autenticación: El acceso al sistema es controlado por la autenticación, donde el usuario debe introducir los datos correctos: correo y contraseña, estos datos son validados del lado del cliente que controla la sintaxis de los campos, así como también con el código del servidor. En la siguiente figura se muestra la parte donde se captura los datos y valida los datos introducidos:

Figura 48

Autenticación de usuario

```

$hab_nombreusuario = strip_tags(trim($this->input->post('name')));
$hab_password = trim($this->input->post('pass'));
if ($this->ModelLogin->login_user($hab_nombreusuario)) {
    $row_users = $this->ModelLogin->login_user($hab_nombreusuario);
    if (password_verify($hab_password, $row_users->password)) {
        $result = $this->ModelLogin->data_user($row_users->id);
        if ($result->estado == 0) {
            $data_session = array(
                'usu_tipo' => $result->tipo,
                'usu_id' => $result->usu_id,
                'rol_id' => $result->rol_id
            );
            $this->session->set_userdata($data_session);
            echo 'Usuario correcto!';
        }else {
            echo 'Usuario Inhabilitado';
        }
    }
}

```

4.3.3 *Encriptación y Validación*

Encriptación BCrypt

Se hace uso de BCrypt, que es un algoritmo hash que transforma una cadena de texto en una cadena de caracteres de longitud fija, llamada hash. El hashing es un proceso unidireccional, lo que significa que es fácil de generar un hash, pero difícil de recuperar la contraseña del hash. Este algoritmo está diseñado para ser lento e intensivo, la cual lo hace resistente a los ataques de fuerza bruta que intentan adivinar contraseñas mediante hashing de muchas combinaciones posibles.

Figura 49

Encriptación de contraseña

```
'email' => strip_tags(trim($this->input->post('email'))),  
'password' => password_hash($this->input->post('pass'), PASSWORD_BCRYPT),  
'rol_id' => $this->input->post('rol_id'),
```

Validación de caracteres en las entradas

La validación en los caracteres de entrada será para garantizar que solo datos válidos sean procesados por el sistema, evitando el riesgo de inyección de código y scripts maliciosos.

Con el uso del protocolo HTTPS que garantiza la comunicación cifrada entre servidor y cliente, con certificados SSL/TLS y el puerto 443.

4.3.4 *Cierre de Sesiones*

Se El cierre de sesiones asegura que las sesiones iniciadas de los usuarios sean finalizadas adecuadamente para evitar accesos no autorizados.

Se utiliza las funciones de gestión de sesiones en Codeigniter para iniciar, mantener y destruir sesiones. Implementar un tiempo de expiración de sesión y asegurar que las sesiones se destruyan completamente al cerrar sesión con `session_destroy`.

CAPITULO V

5 ANALISIS DE COSTO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



5.1 INTRODUCCIÓN

En el desarrollo de este capítulo, se dará a conocer el análisis de costo del sistema usando el método COCOMO II en base al costo del sistema y otros gastos se determina la rentabilidad del sistema con el cálculo del valor neto actual.

5.2 CALCULO DE COSTOS COCOMO II

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad del desarrollo de software.

5.3 Análisis de Costos

Para el análisis de costos COCOMO II, utiliza las siguientes ecuaciones para el cálculo del esfuerzo, tiempo de desarrollo y productividad de un determinado proyecto.

$$E = a * (KLDC) B \quad (1)$$

$$T = C * (E)^d \quad (2)$$

$$P = E/T \quad (3)$$

Donde:

E: Es el esfuerzo requerido por el proyecto, en persona – mes.

T: Es el tiempo requerido por el proyecto, en meses.

P: Es el número de personas requerido por el proyecto, en personas.

A, B, C y d: Son constantes con valores definidos en una tabla, según el tipo de proyecto.

KLDC: Es la cantidad de líneas de código, en miles.

El costo total del sistema está dado por el costo de desarrollo de software, la implementación no requerirá de costo ya que la institución posee servidores propios.

5.4 Costo de Desarrollo de Software

Para hallar el costo de construcción software utilizamos el Punto Función (PF) obtenido anteriormente y además utilizamos el factor de conversión adecuado al lenguaje de programación usado en el desarrollo del sistema.

Tabla 42

Conversión de Puntos de Función a KDLC

LENGUAJE	NIVEL	FACTOR LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL / I	4	80
Ansi Cobol 74	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Nota: Pressman, 2002

Entonces, realizando los cálculos y escogiendo el valor del lenguaje de PHP, tenemos:

$$LCD = PF * Factor LCD/PF (4)$$

$$LCD = 148,5 * 29 = 4306,5$$

Convirtiendo a KLDC, número estimado de líneas de código distribuidas en miles:

$$KLCD = 4306,5 / 1000$$

$$KLCD = 4,30$$

Así que se tiene: 4,30 miles de líneas de código o KLDC

Ahora, para hallar el esfuerzo reemplazamos los valores obtenidos hasta ahora:

$$E = 2,4 * (4,30)^{1,05} = 11,10$$

El esfuerzo calculado es de 11 personas / mes.

Calculando T (tiempo requerido por el proyecto, en meses) con C = 2,5 y d=0.38

$$T = 2,5 * (11,10)^{0,38} = 6,23$$

El proyecto deberá tener un desarrollo aproximado a 6 meses.

Calculamos el personal requerido, es decir el número de programadores estimado:

$$P = 11,10 / 6,23 = 1,78 \cong 2 \text{ Programadores}$$

Estimando que el salario medio de un programador es de 2.500.

Costo del software desarrollado = N^o.de,programador (NP) * Salario

Costo de Software desarrollado por persona = 2 * 2.500 = 5000

Costo total del Software desarrollado = 5000 * 6 = 30000

En resumen, se requiere 2 personas, estimando un trabajo de 6 meses y con un costo total de 30000 Bs.

CAPITULO VI

6 CONCLUSIONES Y RECOMENDACIONES

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



6.1 INTRODUCCIÓN

Durante la realización del presente proyecto y luego de haber concluido la elaboración del proyecto, el cual demuestra el alcance de los objetivos planteados en el capítulo I, se llega a las siguientes conclusiones y recomendaciones.

6.2 CONCLUSIONES

En la unidad de discapacidad se encuentra la necesidad de mejorar la atención a las personas con discapacidad con una eficaz administración de información, con un servicio estable brindando su estado actual de cobros e información personal. De tal manera que el actual proyecto surge como una respuesta a la necesidad por la población, donde se concluye que este proyecto contribuye con efectividad y claridad la mejora de atención.

El objetivo general del proyecto ha sido alcanzado satisfactoriamente y los objetivos específicos fueron satisfactoriamente cumplidos de la siguiente manera:

- Se desarrollo una base de datos sólida y óptima para su uso.
- Se desarrollo un módulo de control y registros de manera adecuada y sencilla.
- Se implemento una manera rápida y sencilla importar la información.
- Se desarrollo un módulo para generar informes para obtener listados y reportes de pagos, así brindar un oportuno control de información.
- Se logró implementar el sistema con una interfaz amigable lo cual facilita de gran manera al usuario pueda realizar sus tareas de forma rápida.

6.3 RECOMEDACIONES

Con el fin de buscar mejorar el sistema se propone las siguientes recomendaciones:

- Capacitar a los nuevos usuarios (administrador o cajero) para la correcta administración o manejo del sistema.
- Se recomienda al usuario cambiar continuamente su contraseña para la seguridad y así proteger el acceso de personas ajenas y malintencionadas.
- Realizar evaluaciones periódicas de la información brindada por el sistema, para poder determinar nuevas necesidades.
- Se recomienda realizar evaluación de satisfacción a los beneficiarios del bono acerca de la plataforma de consultas para así realizar mejoras de acuerdo a sus necesidades.
- Se recomienda realizar copias de la base de datos cada que se realice el cargado de personas habilitadas cada mes, para evitar cualquier fallo al momento de realizar el cargado.

Tabla 43

Tabla comparativa con el sistema implementado

SIN EL SISTEMA	CON EL SISTEMA IMPLEMENTADO
Carpetas dispersas con fotocopias del carnet y certificados de nacimiento.	Almacenamiento en el servidor en forma digital de los documentos.
Largas filas por el pago en instalaciones.	Proceso de pago a través del banco.
Información de las personas con discapacidad inconsistentes.	Mayor organización de datos con una estructura normalizada.
Sin medios de difundir información a las personas con discapacidad.	Módulo de consulta, donde se accede a información de pagos y trámites.

REFERENCIAS BIBLIOGRÁFICAS

- Angel Robledano (2019) Bases de Datos <https://openwebinars.net/blog/mysql/>
- Alicia. (2019). Implementación de un sistema web para el control del pago de pensiones escolares en la institución educativa particular San Juan El Obrero S.A.C. Recuperado de <https://alicia.concytec.gob.pe/vufind/Record/ULAD//>
- Apache JMeter (2024): <https://jmeter.apache.org/>
- Bascon, E. (2004). El patron de diseño Modelo Vista Controlador. Jalasoft
- Codeigniter 3 (2022): <https://codeigniter.es>
- COCOMO II. Gomez, A., Lopez C., A., Migani, S., & Otazu, A. (2010). COCOMO Un modelo de estimación de proyectos de software. México.
- College Derkra (2020). El control como fase del proceso administrativo. Recuperado <https://www.gestiopolis.com/el-control-como-fase-del-proceso-administrativo/>
- Fernández, H. F., & Rodríguez, J. H. (2021). Aplicaciones web con Php. Ra-Ma Editorial.
- Hans Kelsen (2018). Teoría general de las normas. Traducción y presentación Miguel Angel Rodilla.
- ISO25000 (2019). ISO 25000 Calidad de Software y datos. Recuperado de <https://www.iso27000.es/iso27002.html>
- ISO27000 (2013). ISO 27002 Dominios de seguridad y controles. Recuperado de <https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&limitstart=0>
- Kendall, K. (1997). Análisis y Diseño de Sistemas.
- López S. M., Vara M. J., Verde M. J., Sánchez F. D., Jiménez H. J., & Castro V. (2014). Desarrollo Web en Entorno Servidor.
- Ministerio de Salud y Deportes (2019). Salud. Recuperado de <https://www.minsalud.gob.bo/3654-pago-bono-discapacidad//>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. Espirales revistas multidisciplinaria de investigación, 2(17), 114-121.
- Montilva J. A. (1986) Desarrollo de sistemas de información.
- Nuvei. (2017). Procesos de pagos. Recuperado de <https://nuvei.com/es/insights/resources/how-does-payment-processing-work>
- Pedrosa S. Jorge (2020) Sistema de pago. Recuperado de <https://economipedia.com/definiciones/pago.html>

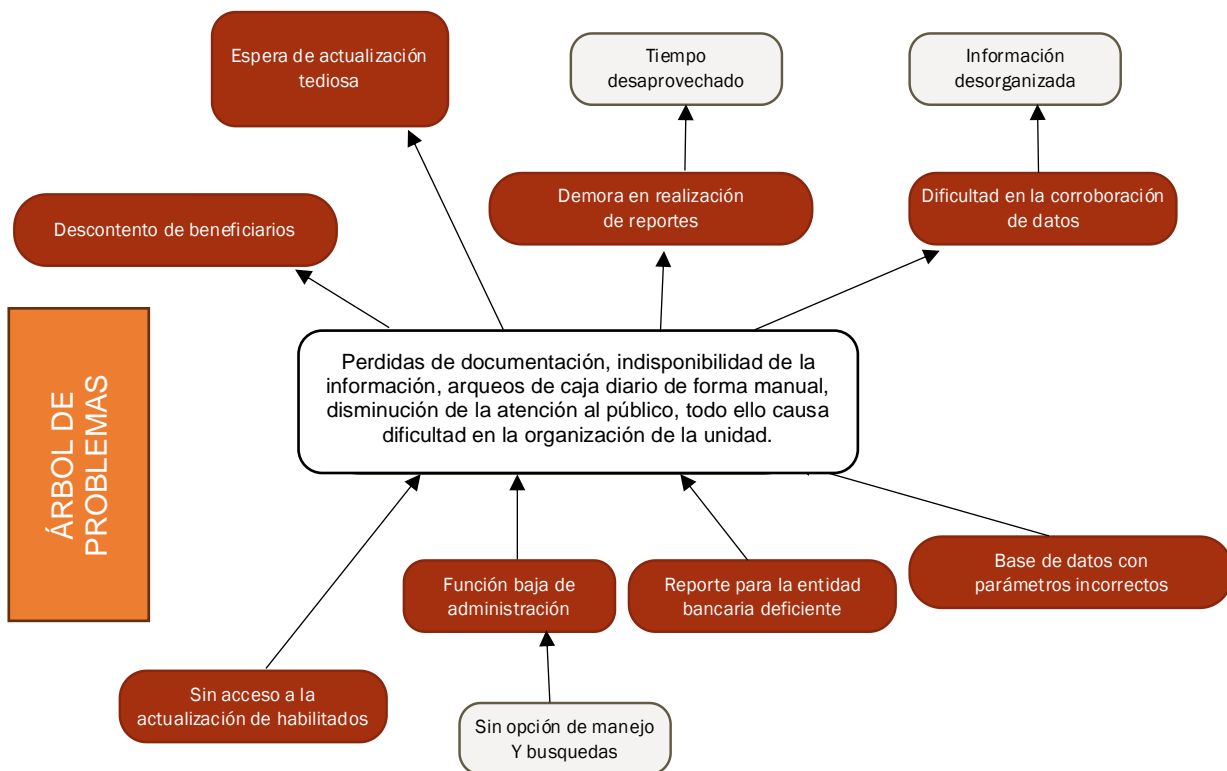
- Pérez, J., & Gardey, A. (15 de Mayo de 2012). Definición de procedimientos administrativos. Obtenido de <http://definicion.de/php/>
- Pressman, R. S. (2002). Ingeniería del software. Un enfoque práctico. (5ta edición).
- Pressman, R. S. (2010). Ingeniería del software. Un enfoque práctico. (7ma edición)
- Repositorio Digital Institucional Universidad César Vallejo. (2021). Sistema web para mejorar el control de pagos en la I.E.P. Los Salesianos, 2021. Recuperado de <https://repositorio.ucv.edu.pe/handle/20.500.12692/101302//>
- Repositorio Institucional UMSA. (2015). Sistema web de inscripciones y control de pagos caso: kumon unidad semilla. Recuperado de <https://repositorio.umsa.bo/handle/123456789/8141//>
- Repositorio Institucional UMSA. (2020). Sistema web de control y seguimiento de servicios y gestión de clientes para la Empresa Consultora Contadores Públicos & Auditores AYS S.R.L. Recuperado de <https://repositorio.umsa.bo/handle/123456789/27967//>
- Repositorio Institucional UNP. (2022). Implementación de un sistema web, utilizando metodología Scrum para mejorar los procesos de matrícula y control de pagos en la Institución Educativa Particular El Triunfo. Recuperado de <https://repositorio.unp.edu.pe/handle/20.500.12676/3688//>
- Robert, M. (1995). Apache HTTP Server proxy pass module. Obtenido de <https://httpd.apache.org/docs/2.0/>
- Schwaber K., & Sutherland J. (2016) La guía definitiva de Scrum. Recuperado de <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>
- Seen J. (1992). Análisis y diseño de sistemas de información. (2. edición).
- Software Guru. (mayo, 2017). La importancia de las metodologías ágiles [Mensaje en un blog]. Recuperado de <https://sg.com.mx/buzz/la-importancia-lasmetodologias-agiles#.WBDzii3hCM>
- Stefaniak. (2019). Back-End y Front-End obtenido de <https://descubrecomunicacion.com/que-es-backend-y-frontend/>
- Tecnoin (enero, 2018). Proceso Scrum [Mensaje en un blog]. Recuperado de <http://tecnoin.com.sv/blog/metodologia-scrum/>
- Thomas H. (2024) Pruebas funcionales y pruebas no funcionales. Obtenido de <https://www.guru99.com/es/functional-testing-vs-non-functional-testing.html>

ANEXOS

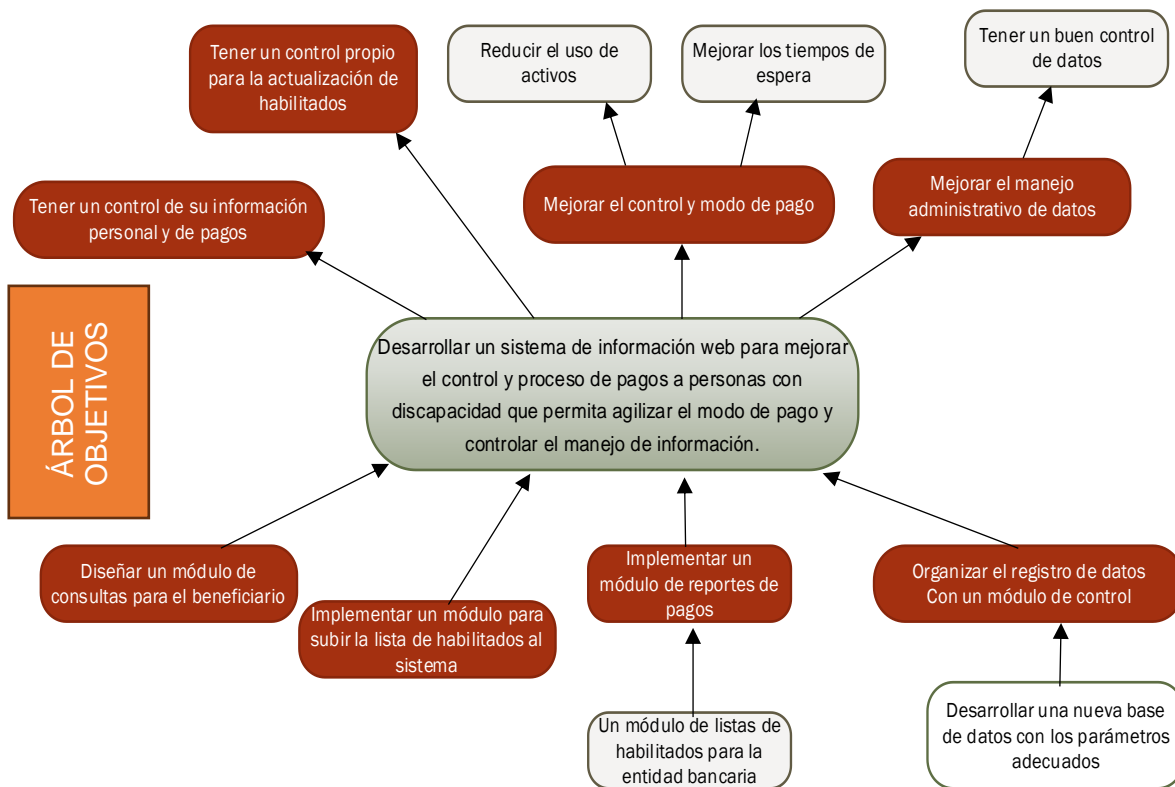
**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO A



ANEXO B





MANUAL DE USUARIO

SISTEMA DE GESTION DE
PAGOS A PERSONAS CON
DISCAPACIDAD
“SIPADIS.”
G.A.M.E.A.

Versión V1
UNIDAD DE SISTEMAS

UNIDAD DE SISTEMAS
MANUAL DE USUARIO SI.PA.DIS

INDICE

1. INTRODUCCION	3
2. OBJETIVO	3
3. REQUERIMIENTOS.....	3
3.1. CONOCIMIENTOS BASICOS.....	3
3.2. REQUERIMIENTOS DE HARDWARE	3
3.3. REQUERIMIENTOS DE SOFTWARE.....	3
3.4. REQUERIMIENTOS DE RED	3
4. GENERALIDADES.....	4
4.1. INGRESO AL SISTEMA.....	4
4.2. INICIO DE SESION.....	4
4.3. CERRAR SESION.....	5
5. ROL CAJERO	6
5.1. MENU LATERAL IZQUIERDO	6
5.2. PAGINA PRINCIPAL.....	7
5.3 PAGO A BENEFICIARIOS	7
5.4 HISTORIAL DE PAGOS.....	9
5.5 DETALLES DEL BENEFICIARIO	9
5.6 COMPLETAR DATOS DEL BENEFICIARIO.....	10
5.7 BENEFICIARIOS OBSERVADOS.....	12
5.8 ADMINISTRAR ZONAS Y DISTRITOS	13
5.9 DETALLES DEL BENEFICIARIO	14
5.10 SUBIR DOCUMENTOS	14
5.11 MÓDULO Manual de uso.....	15
5 ROL ADMINISTRADOR.....	16
6.1 MENU LATERAL IZQUIERDO	16
6.2 PAGINA PRINCIPAL ADMINISTRADOR	16
6.3 MÓDULO ADMINISTRAR USUARIOS.....	17
6.4 MÓDULO REPORTES Boletas PAGADAS.....	18
6.4.1 MÓDULO reportes para el banco	18
6.5 MÓDULO CARGADO DE ARCHIVO AL SISTEMA	19
6.5.1 PASOS PARA CARGAR EL ARCHIVO.....	20

1. INTRODUCCION

El sistema "SI.PA.DIS." Sistema de gestión de pagos a personas con discapacidad es una herramienta informática fundamental para la administración eficiente de información de los beneficiarios.

2. OBJETIVO

Instruir al usuario en el uso correcto del sistema y la solución de los problemas que puedan suceder en la cada operación.

3. REQUERIMIENTOS

3.1. **CONOCIMIENTOS BASICOS**

- Manejos de navegadores de Internet.

3.2. **REQUERIMIENTOS DE HARDWARE**

Los requisitos mínimos de hardware para el correcto funcionamiento del sistema son:

- Computadora de escritorio o personal (Laptop).
- Conexión a internet

3.3. **REQUERIMIENTOS DE SOFTWARE**

Sistema Operativo:

- Windows (Versión 7 o superior)

Navegadores:

- Google Chrome (Versión 3.5 o superior)
- Opera (Versión 3.5 o superior)
- Microsoft Edge (Versión 3.5 o superior)
- Mozilla Firefox (Versión 3.5 o superior)
- O navegadores de distribuciones Linux, de versiones mencionados.

3.4. **REQUERIMIENTOS DE RED**

- Conexión a la intranet del G.A.M.E.A.
- Conexión a Internet y servicio de datos de cualquier proveedor de internet.

4. GENERALIDADES

4.1. INGRESO AL SISTEMA

En la URL de su navegador, ingrese el siguiente enlace: <http://sipadis.elalto.gob.bo/>, aparecerá una ventana como la mostrada en (Fig.1)



Figura 1: Dirección url del Sistema de SIPADIS.

4.2. INICIO DE SESION

Seguidamente aparecerá la ventana de ingreso, mostrada en (Fig.2), donde cada usuario deberá ingresar sus datos correctamente.

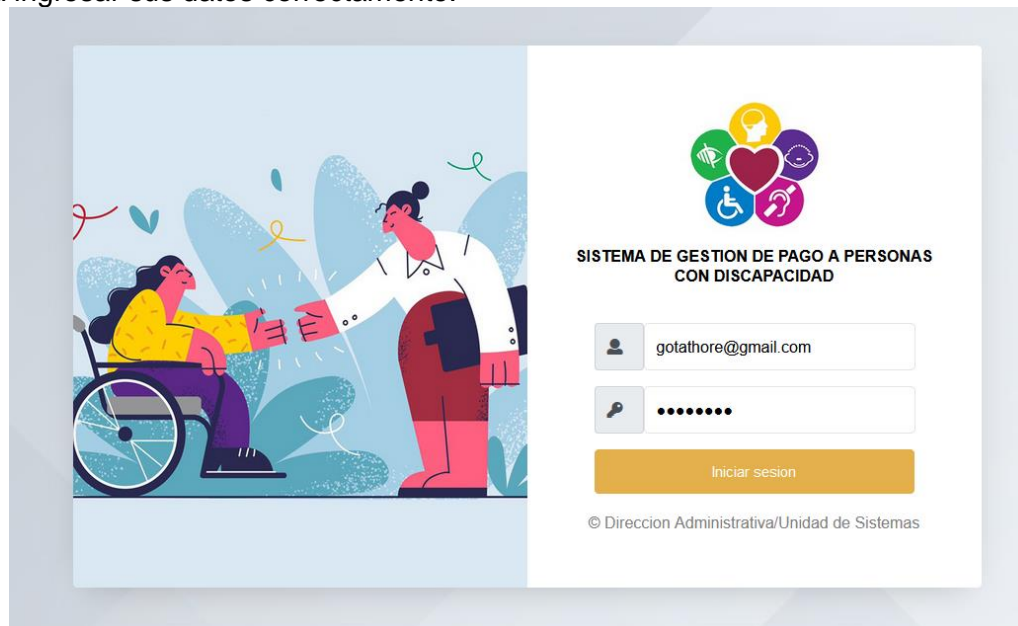


Figura 2: Login (Inicio de sesión) del Sistema SIPADIS.
IMPORTANTE.



Introducir el nombre de Usuario, Contraseña, presionar el botón “Iniciar sesión”. El sistema validara los datos introducidos, si sus datos son correctos ingresara al panel principal del sistema, caso contrario el sistema lo hará saber con un mensaje, como se muestra en la (Fig.3) y deberá intentar nuevamente. En caso de Ingresar los datos correctos inicia el sistema y vera el panel principal (Fig.6)

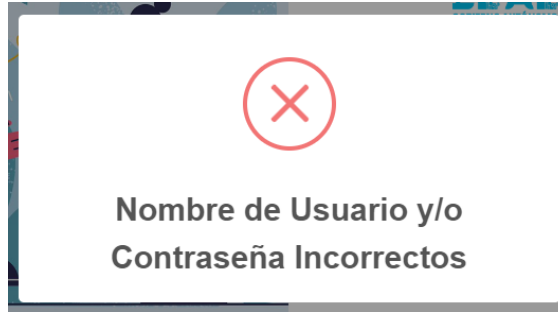


Figura 3: Errores de Autenticación.

4.3. CERRAR SESION

Para cerrar la sesión nos dirigimos a la parte superior derecha de la ventana para desplegar las opciones y elegimos Cerrar Sesión (Fig.4).

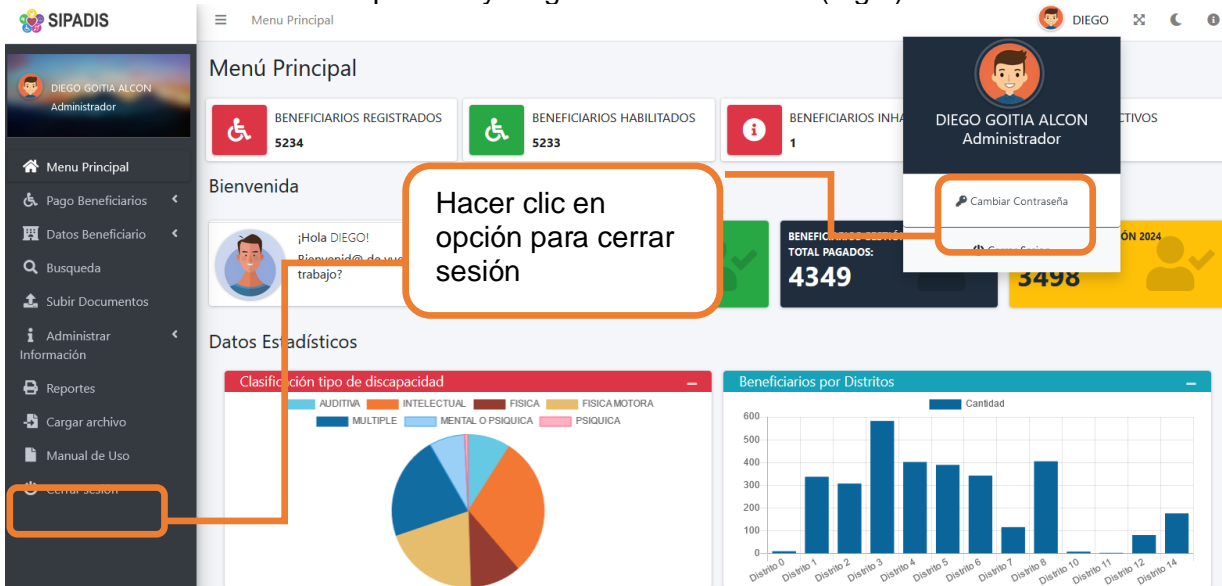


Figura 4: Cerrar Sesión.

IMPORTANTE.



Una vez finalizada la sesión nos reenviará a la ventana del “Login”, y nos volverá a pedir nuestro nombre de usuario y contraseña.
Cierra la sesión cada vez que dejara de manipular el sistema.

5. ROL CAJERO

5.1. MENU LATERAL IZQUIERDO

En este apartado muestra el nombre del Sistema, el nombre del usuario que inicio sesión, y el menú de opciones (Fig.5).



Figura 5: Menú, lateral Izquierdo Encargado.

5.2. PAGINA PRINCIPAL

En la parte central de la página se muestra los accesos directos a cada opción del sistema y datos del usuario. (Fig.6).

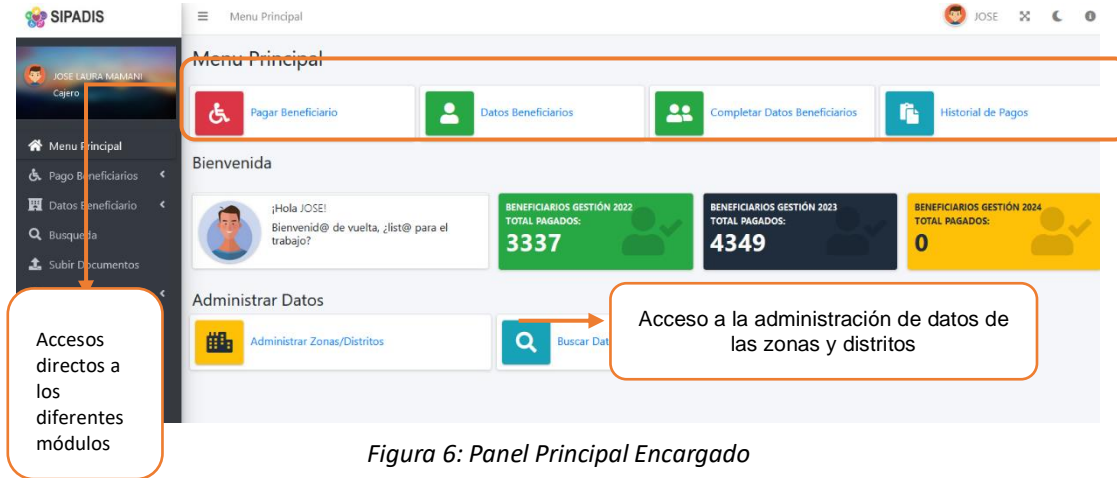


Figura 6: Panel Principal Encargado

5.3 PAGO A BENEFICIARIOS

En este módulo se podrá realizar el pago de beneficiarios. (Fig.7)

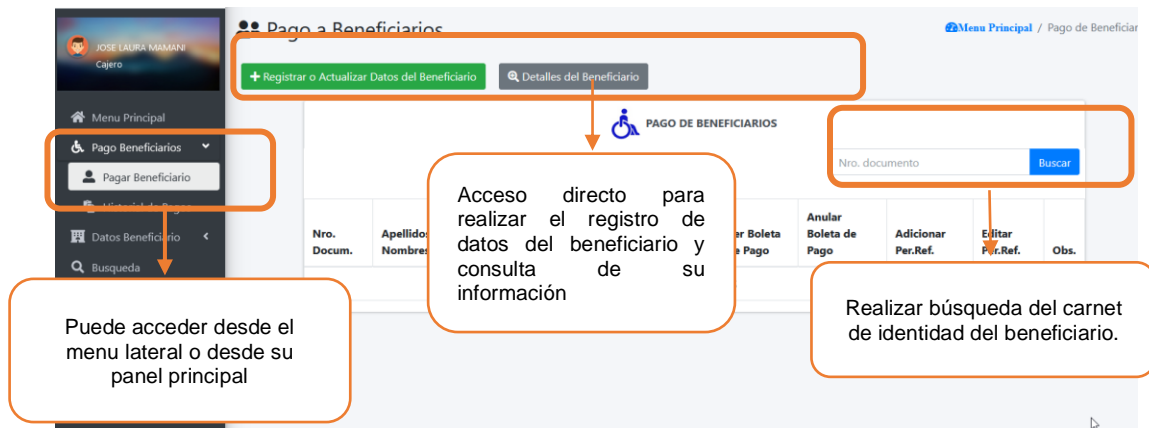


Figura 7: Módulo de pago a beneficiarios.

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS


Pago de Beneficiarios

SIN DOCUMENTOS REGISTRADOS!

4837139

Nro. Docum.	Apellidos y Nombres	Tipo de Pago	Mes	Gestion	Pagar Beneficiario	Boleta de Pago	Boleta de Pago	Adicionar Per.Ref.	Editar Per.Ref.
4837139	SOLDADO CHAMBI FABIANA	COBRO REGULAR	NOVIEMBRE2023		<div style="border: 1px solid green; padding: 2px; display: inline-block; color: green;">Pagar Beneficiario</div>	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">Boleta de Pago</div>		<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">+</div>	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">✎</div>

Seleccione para imprimir boleta

Seleccione para adicionar o editar a una persona de referencia para el cobro

Seleccione pagar beneficiario

Adicionar Observacion

GOBIERNO MUNICIPAL DE EL ALTO

BOLETA DE PAGO

Bs. 250
Periodo: NOVIEMBRE/2023

DATOS DEL BENEFICIARIO

NOMBRES Y APELLIDOS: SOLDADO CHAMBI FABIANA
 NRO. DOCUMENTO: 4837139 LPZ
 ACOMPAÑANTE:

SELECCION ACOMPAÑANTE

DATOS DE DISCAPACIDAD

Tipo de Discapacidad: FISICA Carnet Discapacidad (C.D.): 19710120P5C
 Deficiencia: VICIAL Porcentaje de deficiencia: 55%
 Vencimiento de C.D.: 2022-01-05 Usa ayuda tecnica: NO

DATOS DE PAGO

MODALIDAD DE PAGO: SELECCION MODALIDAD

TIPO DE PAGO: COBRO REGULAR
 FECHA DE PAGO: 22/11/2023 09:00:03

Cancelar

Pagar Beneficiario

Seleccione los campos de acompañante y modalidad de pago

Seleccione Pagar Beneficiario

Figura 8: Realizar pago a beneficiarios.

ADICIONAR PERSONA DE REFERENCIA para el beneficiario

Datos Personales

Primer Nombre (obligatorio) Segundo Nombre Tercer Nombre OPCIONAL
 Primer Apellido (obligatorio) Segundo Apellido Tercer Apellido OPCIONAL

Vivulo familiar (obligatorio) Cédula de identidad (obligatorio) Exp. Sexo (obligatorio) Fecha de Nacimiento (obligatorio)
 DD / MM / AAAA

Datos Complementarios

Domicilio (obligatorio) Deseño (obligatorio) Zona (obligatorio) (Puede seleccionar distrito)

Estado Civil (obligatorio) Ocupación (obligatorio)

Grado de instrucción Observaciones

Cancelar

Adicionar Persona de Referencia

Llene los espacios y luego pulse Adicionar

ACTUALIZAR PERSONA DE REF. de LUGARANI RODRIGUEZ CRESCENCIO(6048179):

MIRIAM LUGARANI FLORES(TUTOR)

Pagar de Beneficiarios

Seleccione el nombre del tutor y prosiga al llenado de datos

Figura 9: Modal Adicionar y Editar Tutor de la tabla Pagar.

5.4 HISTORIAL DE PAGOS

En este módulo podrás elegir una determinada para ver el historial de pagos del usuario. (Fig.10)

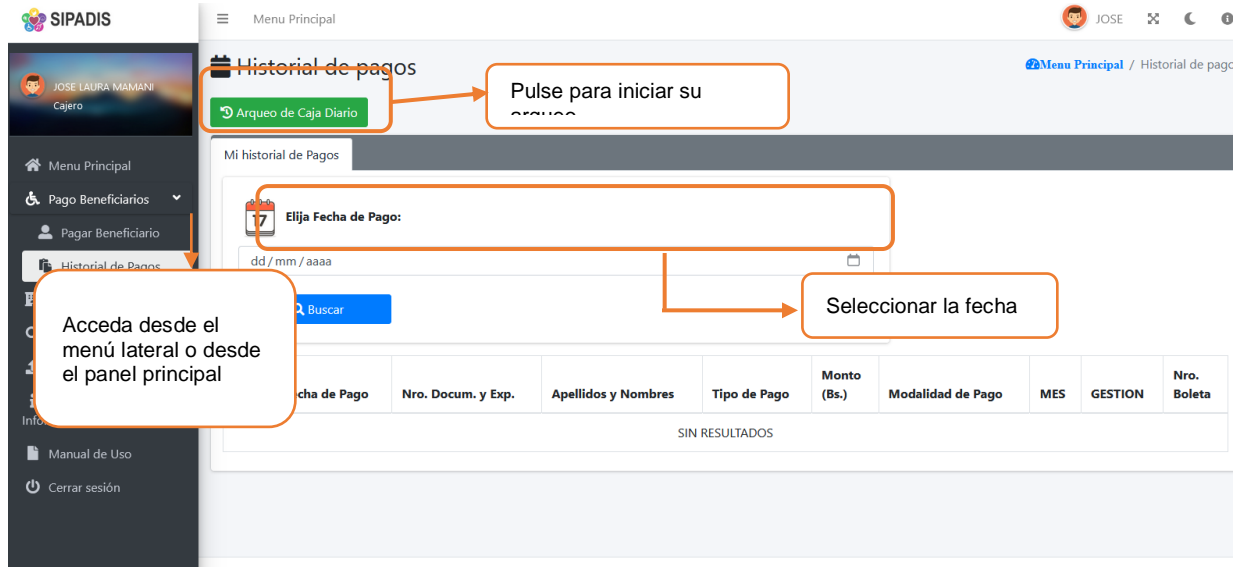


Figura 10: historial de pagos del usuario.

5.5 DETALLES DEL BENEFICIARIO

En este módulo podrás consultar los datos registrados de los beneficiarios, además de poder registrar o editar a las personas de referencia del beneficiario. (Fig.11)

Podrás asignar una persona de referencia al beneficiario si este así lo desea. (Fig.12)

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS



Figura 11: Ver detalles del beneficiario.

Puede realizar la asignación de beneficiario en el mismo módulo, pulse “Asignar o cambiar persona de referencia”

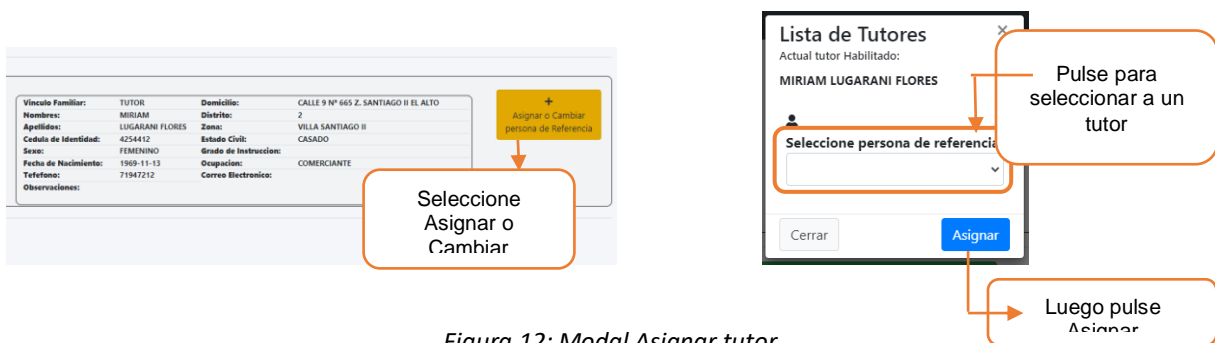


Figura 12: Modal Asignar tutor.

5.6 COMPLETAR DATOS DEL BENEFICIARIO

Módulo de registro y edición de datos de los beneficiarios (Fig.13)

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS



Figura 13: Panel para el registro y edición de datos.

Al terminar el registro presione "insertar/Actualizar Beneficiario" para guardar cambios (Fig.14)

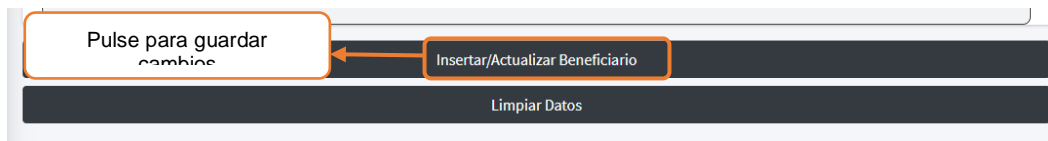


Figura 14: Botón Guardar.

Al pulsar "Persona de Referencia" saldrá la siguiente ventana emergente (Fig.15)

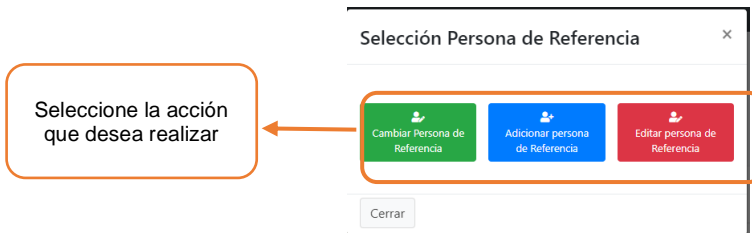


Figura 15: Ventana emergente Persona de Referencia.

Al pulsar "Cambiar Persona de referencia" obtendrá la siguiente ventana, asegúrese de seleccionar a uno de sus tutores (Fig.16)

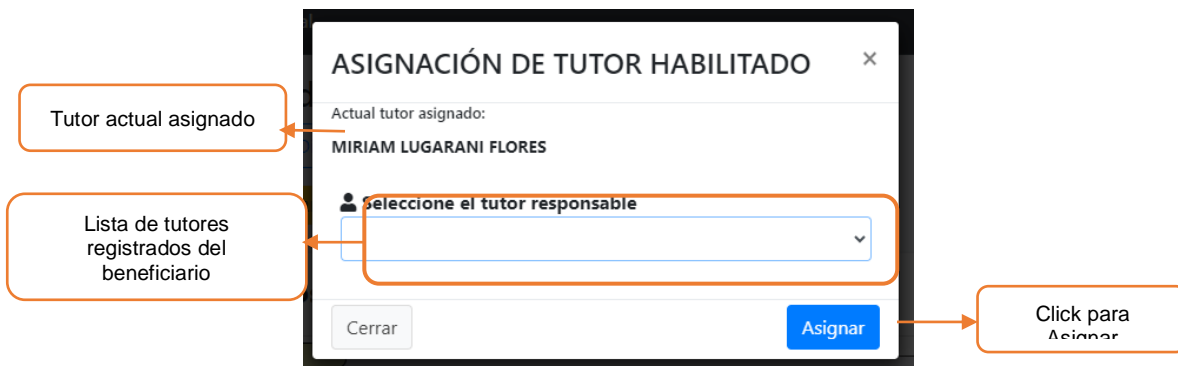


Figura 16: Modal Asignar tutor.

Al pulsar “Adicionar Persona de referencia” obtendrá la siguiente ventana, asegúrese de llenar los campos que son requeridos y luego pulse “Adicionar” (Fig.17)

The image displays two overlapping screenshots of a web application interface. The top screenshot is a modal window titled "ADICIONAR PERSONA DE REFERENCIA para el beneficiario". It contains two main sections: "Datos Personales" and "Datos Complementarios". The "Datos Personales" section includes fields for "Primer Nombre", "Segundo Nombre", "Tercer Nombre" (marked as "OPCIONAL"), "Primer Apellido", "Segundo Apellido", and "Tercer Apellido" (marked as "OPCIONAL"). It also has dropdowns for "Vínculo familiar", "Cédula de Identidad", "Exp.", "Sexo", and a date field for "Fecha de Nacimiento" (DD/MM/AAAA). The "Datos Complementarios" section includes "Domicilio", "Distrito", "Zona" (with a "Mostrar selección distinta" link), "Estado Civil", "Ocupación", "Tel./Cel.", "Grado de instrucción", "Correo", and "Observaciones". At the bottom are "Cancelar" and "Adicionar Persona de Referencia" buttons. An orange callout box points to the "Adicionar Persona de Referencia" button with the text "Click para Adicionar".

The bottom screenshot is a modal window titled "ACTUALIZAR PERSONA DE REF. de:" for "MIRIAM LUGARANIFLORES (TUTOR)". It shows a list of references with columns for "PERSONA REFERENCIADA", "Persona de Referencia", and "6048179". Below this is a "Datos Beneficiario (Complemento)" section with "Datos Registrados" (Nro. Documento: 6048179) and "Datos Personales" (Primer Nombre: CRESCENCIO, Segundo Nombre, Primer Apellido: LUGARANI, Segundo Apellido: RODRIGUEZ, Tercer Apellido: OPCIONAL). An orange callout box points to the "PERSONA REFERENCIADA" column with the text "Al pulsar 'Editar Persona de referencia' pulse el nombre si existiera, luego proseguir con el llenado de datos."

Figura 17: Modal Adicionar y Editar Persona de referencia.

5.7 BENEFICIARIOS OBSERVADOS

En este módulo se encontrará las listas de beneficiarios que llegan a tener alguna observación debido a su edad, menores de 18 podrán ser habilitados una vez que se le asigne un tutor. Y beneficiarios mayores a 60 ya no podrán realizar el cobro. (Fig.18)

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS



Figura 18: Lista de beneficiarios observados.

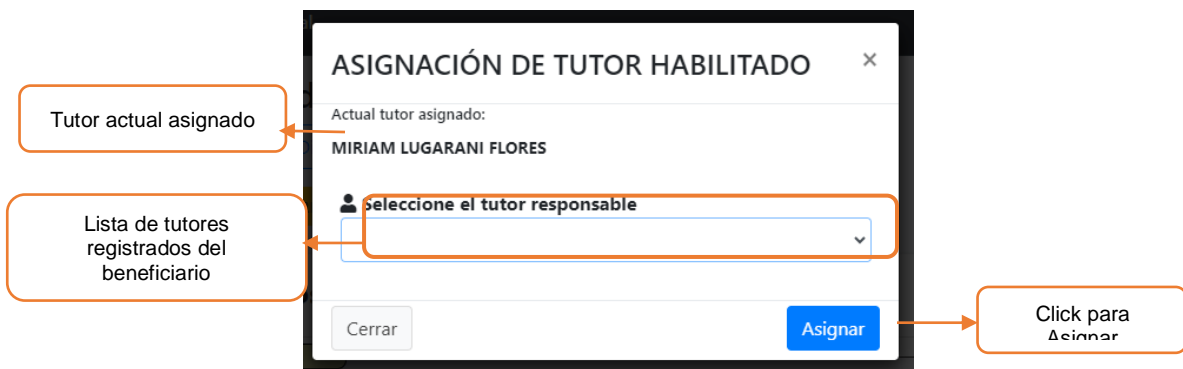


Figura 19: Asignar persona de referencia.

5.8 ADMINISTRAR ZONAS Y DISTRITOS

Si desea agregar o corregir los datos de las zonas entonces diríjase a este módulo para la administración de las zonas y distritos (Fig.20).

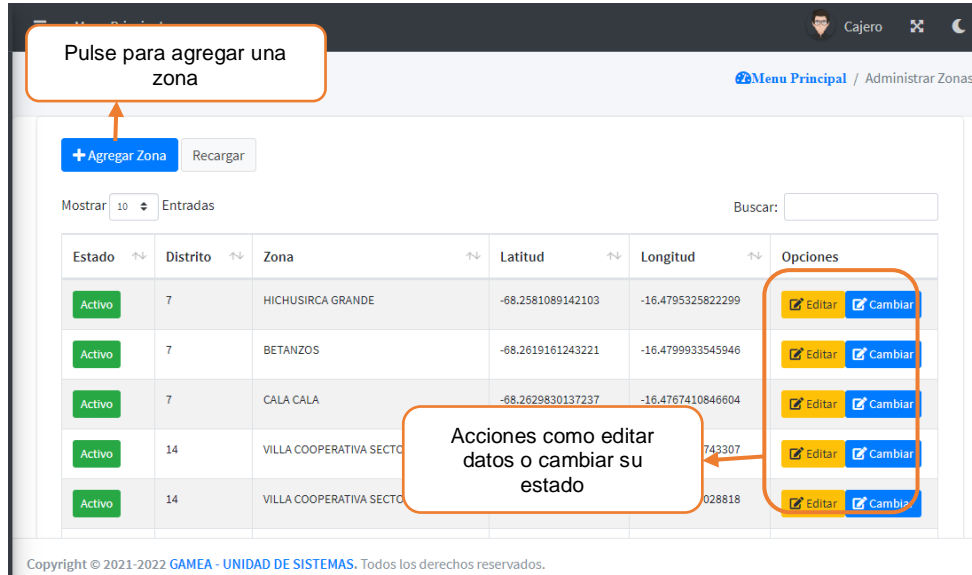


Figura 20: Tabla de Zonas y Distritos.

5.9 DETALLES DEL BENEFICIARIO

En este módulo podrás consultar los datos registrados de los beneficiarios, además de poder registrar o editar a las personas de referencia del beneficiario. (Fig.21)



Figura 21: Ver detalles del beneficiario.

5.10 SUBIR DOCUMENTOS

En este módulo podrás subir imágenes escaneadas de los documentos originales de los beneficiarios, en formato JPEG, JPG y PNG. (Fig.22)

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS

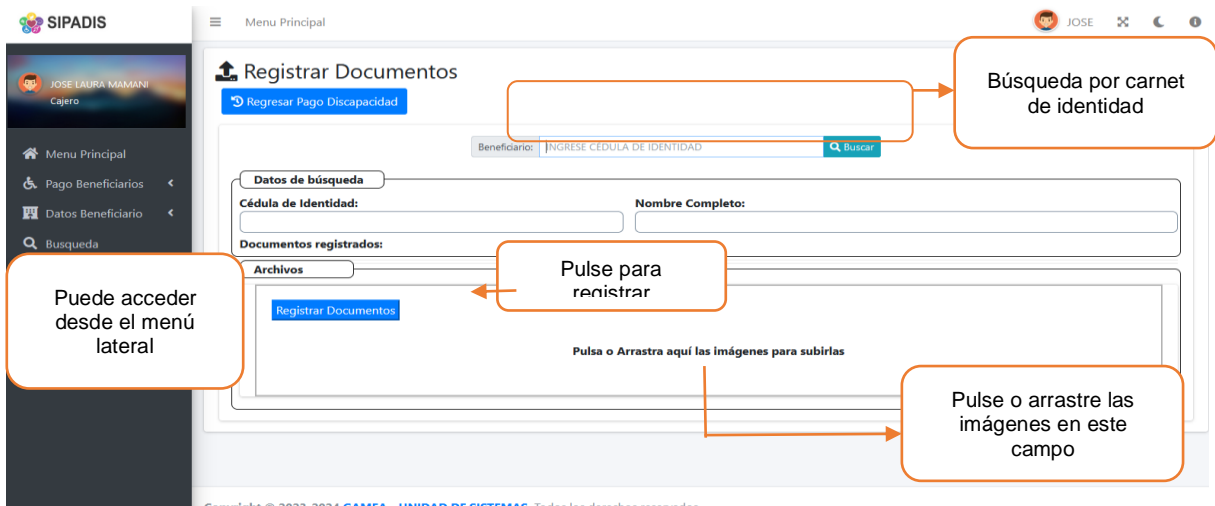


Figura 22: Subir documentos.

5.11 MÓDULO Manual de uso

En este módulo se podrá descargar o consultar el manual de uso y también poder ver videos tutoriales de procesos del sistema que sean poco complejos. (Fig.23).

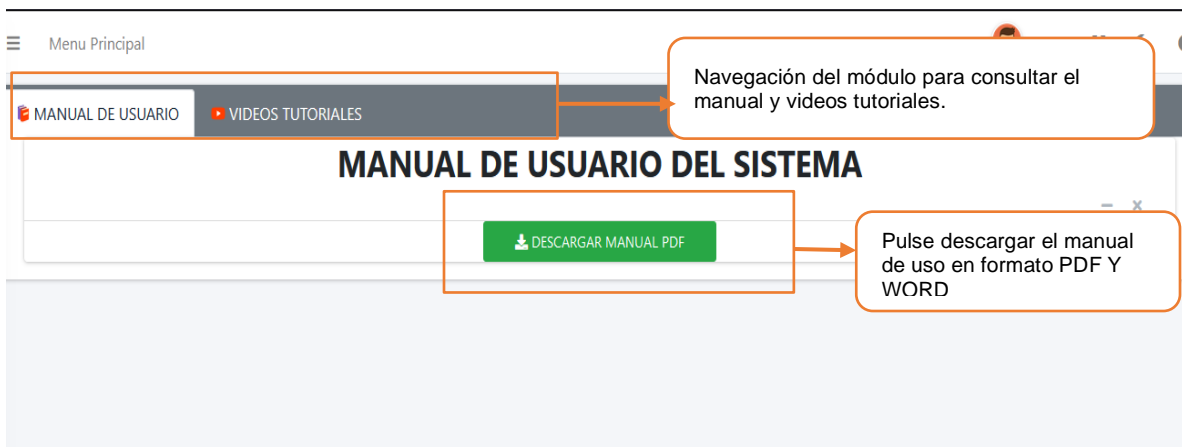


Figura 23: Módulo Manual de uso.

6. ROL ADMINISTRADOR

6.1. MENU LATERAL IZQUIERDO

En este apartado muestra el nombre del Sistema, el nombre del administrador que inicio sesión, y el menú de opciones (Fig.24).

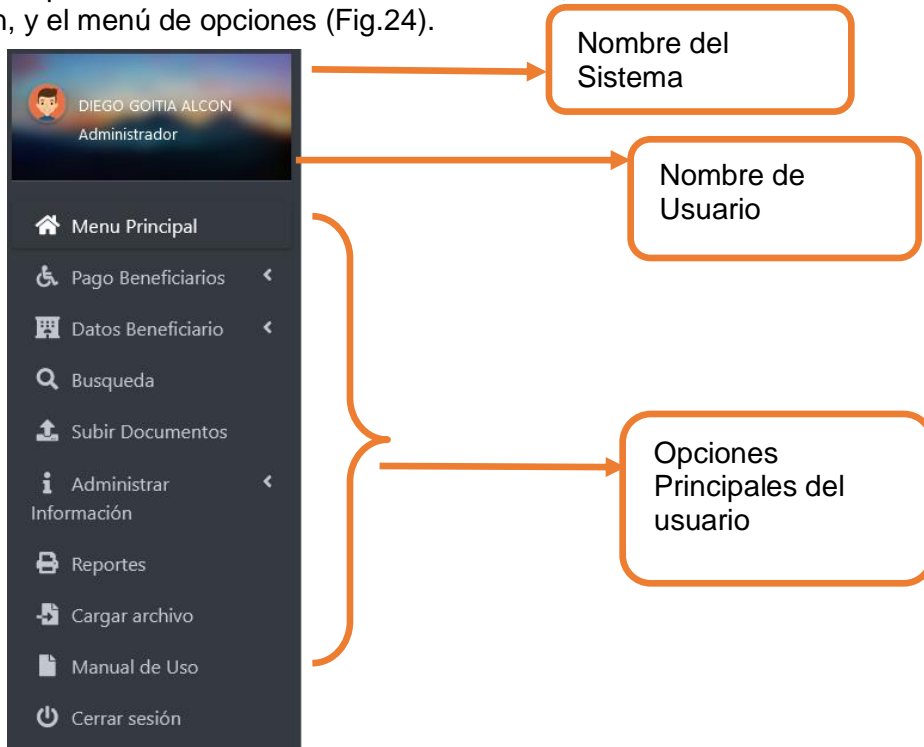


Figura 24: Menú, lateral Izquierdo Administrador.

6.2. PAGINA PRINCIPAL ADMINISTRADOR

En la parte central de la página se muestra los accesos directos a cada opción del sistema y datos a los que este rol tendrá acceso. (Fig.25).

UNIDAD DE SISTEMAS MANUAL DE USUARIO SI.PA.DIS



Figura 25: Panel Principal Administrador

El “Administrador” tendrá todas las funciones ya adquiridas por el rol “Cajero” de esta manera puede consultar los módulos ya mencionados anteriormente.

6.3. MÓDULO ADMINISTRAR USUARIOS

En este módulo se podrá realizar la administración de usuarios que tendrán acceso al sistema, se realizará la edición o la inhabilitación del usuario “Encargado”. (Fig.26).

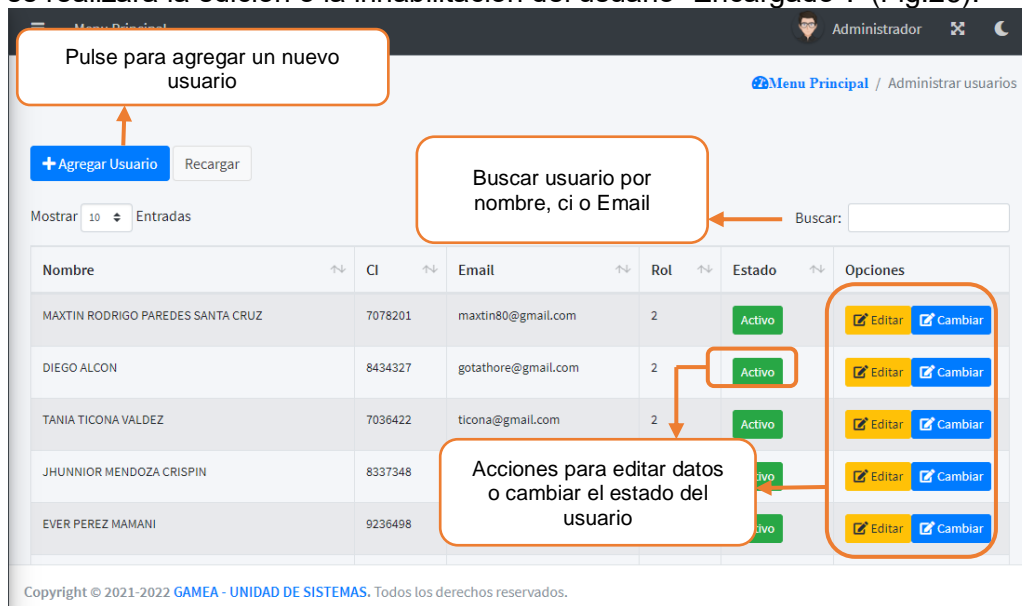


Figura 26: Tabla de administración de usuarios.

Al adicionar o editar datos, pulsar el botón “Guardar” para el correcto registro (Fig.27)



Figura 27: Modal Usuario.

6.4 MÓDULO REPORTES Boletas PAGADAS

En este módulo se podrá realizar reportes que son requeridos por la unidad, como los reportes de boletas pagadas. (Fig.28).

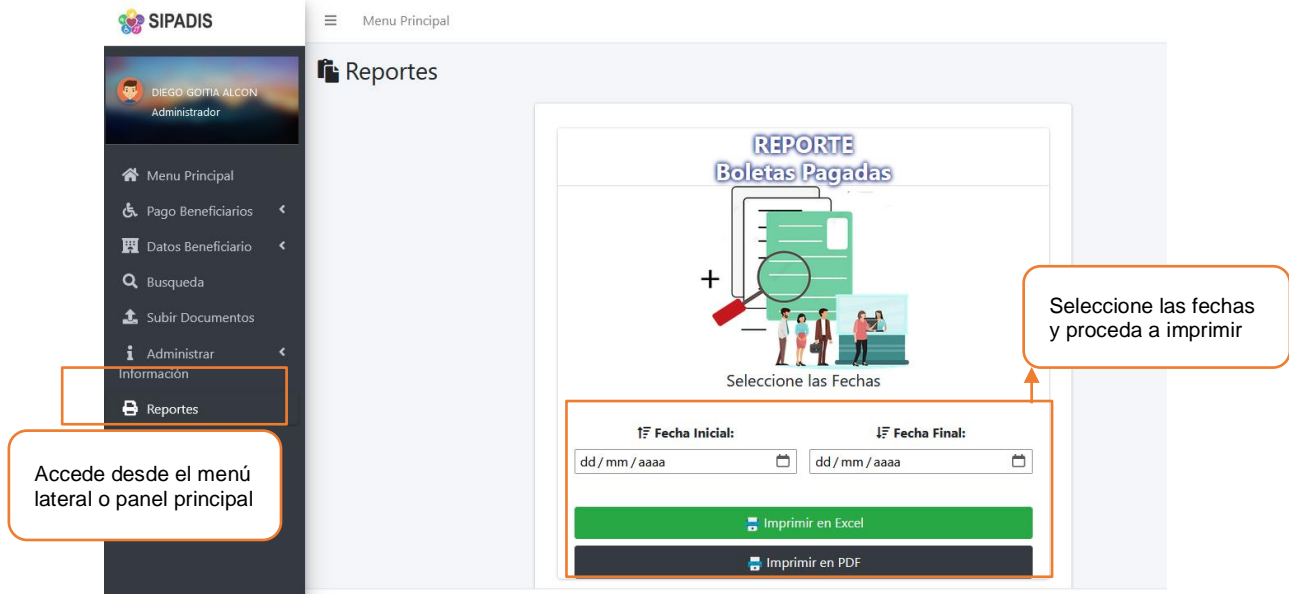


Figura 28: Módulo de reporte boletas pagadas.

6.4.1 MÓDULO reportes para el banco

En este módulo se podrá realizar reportes que son requeridos por la entidad bancaria para poder realizar el pago a beneficiarios. (Fig.29).

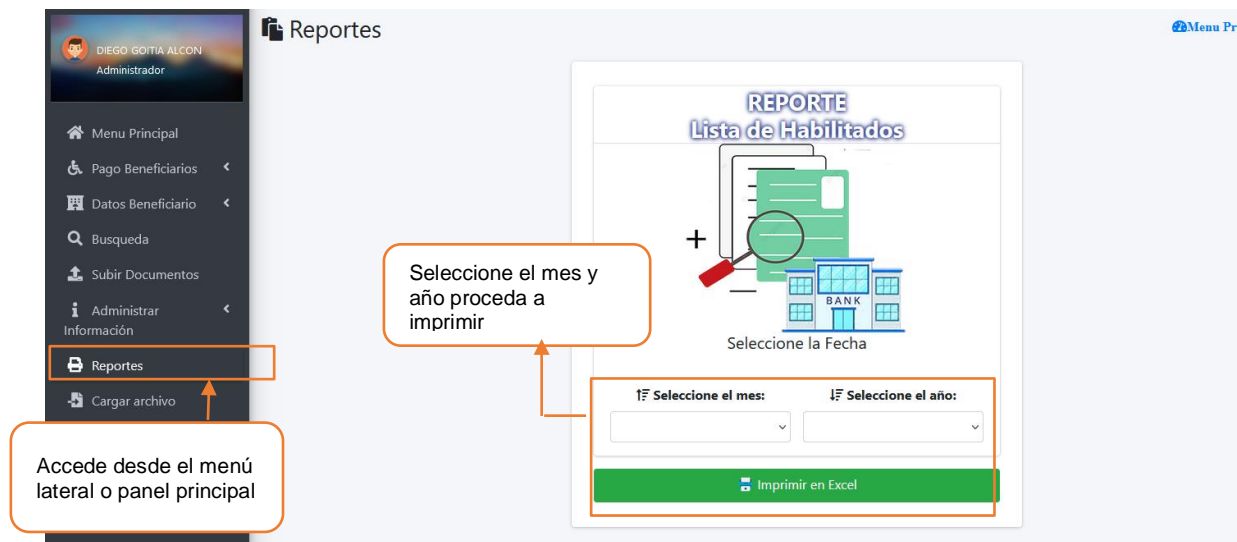


Figura 29: Módulo reporte para el banco.

6.5 MÓDULO CARGADO DE ARCHIVO AL SISTEMA

En este módulo se podrá realizar el cargado del archivo que se realiza por cada mes. Se llegará a cargar en un formato propio de Excel que es el formato “.CSV”, y para que no exista algún fallo en el cargado deberá seguir las indicaciones que se encontrará en el módulo o dirigirse al módulo del manual de usuario “tutoriales” para ver un video (Fig.30).



Figura 30: Módulo Cargado de archivo al sistema.

6.5.1 PASOS PARA CARGAR EL ARCHIVO

Cuando se llega a tener la lista de beneficiarios habilitados del mes que corresponde, se obtiene un archivo Excel con los datos. Se tendrá que modificar las cabeceras de las columnas para una inserción correcta, y eliminar filas y columnas que no serán necesarios como el TITULO, NRO, MONTO y el TOTAL. (Fig.31)

Eliminar filas y columnas no necesarias

El documento quedará de esta manera.

CEDULA DE IDENTIDAD	APELLIDOS Y NOMBRES DE BENEFICIARIOS
4369856	CALLE NOLBERTO
4837776	CHAMBI JUANA
4978837	CHUQUIMIA CARINA LOURDES
16281294	COLQUE LUCIO EMANUEL
16454598	CRUZ GEREMY GABRIEL
16126660	QUIROGA MARIELA
9951179	QUISBERT MONICA ALEJANDRINA
14823561	RIVEROSCARLOSALBERTO
2708102	RODRIGUEZ JUAN TITO
3326445	VALENCIA REMIGIO ALBERTO
15869555	YANIQUE LISBETH ALEXA
9957786	ACARAPI CRUZ ALEX
9974354	ACARAPI HUANCA RUBEN CARLOS
13297398	ACARAPI LAURA JOSE BASILIO

Figura 31: Editar documento

Luego deberá cambiar el nombre de las columnas restantes, con CI, NOMBRE (EXP si el documento llegara a tener esta fila). (Fig.32). luego seleccione “Guardar como” y guardar en formato “.CSV” con el nombre del mes que desea cargar (Fig.32)

Cambie el nombre de estas columnas con lo respectivo

Guarda con el nombre del mes en MAYUSCULA con la extensión “CSV delimitado por comas”

CI	NOMBRE
4369856	CALLE NOLBERTO
4837776	CHAMBI JUANA
4978837	CHUQUIMIA CARINA LOURDES
16281294	COLQUE LUCIO EMANUEL
16454598	CRUZ GEREMY GABRIEL
16126660	QUIROGA MARIELA
9951179	QUISBERT MONICA ALEJANDRINA
14823561	RIVEROSCARLOSALBERTO
2708102	RODRIGUEZ JUAN TITO
3326445	VALENCIA REMIGIO ALBERTO
15869555	YANIQUE LISBETH ALEXA
9957786	ACARAPI CRUZ ALEX
9974354	ACARAPI HUANCA RUBEN CARLOS
13297398	ACARAPI LAURA JOSE BASILIO

Figura 32: Guardar Archivo.

El archivo quedará de la siguiente manera (Fig.33) y seguir luego seguir las indicaciones (Fig.30)

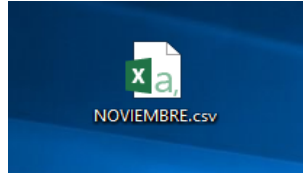


Figura 33: Archivo CSV.

6.6 ADMINISTRACIÓN DEL MÓDULO DE CONSULTAS

Pulse en el icono de usuario en la parte inferior e ingrese con su usuario y contraseña de administrador, en la url <https://consulta-pcd.elalto.gob.bo/>.

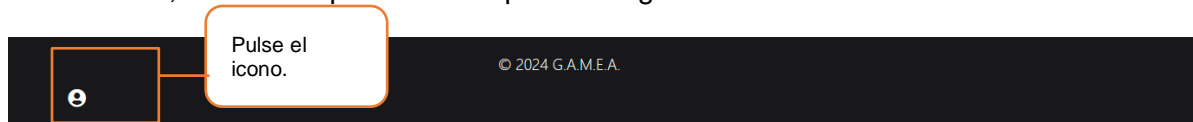


Figura 34: Ingreso de usuario.

6.7 subir una publicación



Figura 35: Cambiar una publicación.