

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN PARA LA ASIGNACIÓN Y
CONTROL DE VIVIENDAS FUNCIONALES”
CASO: DIRECCIÓN GENERAL DE DESARROLLO Y
BIENESTAR SOCIAL (DIGEDEBIS)**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POR: FERNANDO LAURA TORREZ.
DOCENTE: M. SC. GROVER ALEX RODRÍGUEZ RAMÍREZ
TUTOR METODOLÓGICO: PH.D. JAVIER HUGO REYES PACHECO**

**LA PAZ – BOLIVIA
2021**

HOJA DE CALIFICACIÓN
UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Proyecto de grado:

**“SISTEMA DE INFORMACIÓN PARA LA ASIGNACIÓN Y CONTROL DE VIVIENDAS
FUNCIONALES”**
**CASO: DIRECCIÓN GENERAL DE DESARROLLO Y BIENESTAR SOCIAL
(DIGEDEBIS)**

Presentado por: Fernando Laura Torrez

Para optar el grado Académico de Licenciatura en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral :.....

Nota Literal:

Ha sido:

Director de la carrera de Informática: Ph.D. José María Tapia Baltazar

Docente: M. SC. Grover Alex Rodríguez Ramírez

Tutor Metodológico: Ph.D. Javier Hugo Reyes Pacheco

Tribunal:

Tribunal:

Tribunal:



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

Dedico el presente trabajo primeramente a Dios que me dio la oportunidad de vivir y regalarme una familia maravillosa.

A mis padres Mario y Teresa que, con su amor y dedicación, me dieron la fuerza para seguir adelante.

A mis princesas mágicas Fernanda y Luciana que son la razón más importante de superación de mi vida.

Y a esa persona especial en mi vida mi esposa Raquel, que siempre me apoyo y me animó a seguir adelante pese a las dificultades.

Con mucho cariño: Fernando Laura

fer.nano_@hotmail.com

Agradecimientos

A Dios por darme la fuerza y voluntad, permitiéndome llegar hasta aquí y culminar esta etapa tan importante de mi vida.

A mis queridos padres Mario y Teresa por impulsarme a seguir adelante, apoyarme en los momentos difíciles y porque siempre confiaron en mí.

A mí querida y adorada esposa Raquel Parrado y a mis princesas mágicas Fernanda y Luciana por apoyarme siempre en todo momento.

A mi tutor metodológico Ph. D. Javier Hugo Reyes Pacheco, quien me brindo de su conocimiento y sabiduría para realizar este proyecto hasta su finalización, gracias por su paciencia para revisar una y otra vez este trabajo.

A mi Docente M. Sc. Grover Alex Rodríguez Ramírez por el tiempo y colaboración, aportando con observaciones y sugerencias para el desarrollo de este proyecto.

A todos los docentes de la carrera de informática por la guía y formación académica.

Al Cnl. DAEN. MSc. Henry Claros Vela Director General de Desarrollo y Bienestar Social por otorgarme la oportunidad de desarrollar el sistema.

A los amigos que siempre me han estado apoyando y ayudando durante mi estancia en la Universidad y en mi vida.

fer.nano_@hotmail.com

RESUMEN

El contar con un sistema de información es una necesidad inevitable en nuestro contexto laboral, puesto que es esencial para la gestión de la información, el cual nos ayuda generar datos confiables y oportunos, y estas nos permiten apoyar en la toma de decisiones.

El desplazamiento de las distintas organizaciones hacia sitios web, ha traído en la actualidad una constante evolución en las aplicaciones Web. La transferencia de información confidencial y ejecución de procesos online, entre otros, exigen funcionalidad, confiabilidad, usabilidad y eficiencia como algunas de las características principales de calidad de Software.

Tomando en cuenta estas consideraciones, es necesario implementar un “Sistema de Información Para la Asignación y Control de Viviendas Funcionales Caso: Dirección General de Desarrollo y Bienestar Social”. Este reto implica: ofrecer un servicio de información que permita realizar seguimiento y generar reportes sobre deudas acumuladas por concepto de pago de alquiler y que los mismos sean confiables, pertinentes y además que la información acerca del comportamiento de los alquileres esté almacenada en una base de datos, segura en un servidor y protegida por un acceso restringido para su posterior utilización.

Para la implementación del sistema de información se utilizaron los lenguajes de programación Java juntamente al entorno de desarrollo NetBeans y Framework JSF, Primeface, una herramienta para el desarrollo de aplicaciones que crea webs utilizando Java Enterprise y la Gestión de Base de Datos de MySQL, los cuales se adecuan a las necesidades y requerimientos de la Institución.

Palabras claves: Sistema, registro, control, metodología, software.

Metodología: Para iniciar con el proyecto se utilizó la metodología Ágil Scrum para el desarrollo y proceso del software; planificando las tareas de acuerdo a las

historias de usuario que son contadas por el Dueño del Producto en este caso el Director de la Institución, identificando las fechas de cada tarea y los responsables para que el producto sea terminado.

Para obtener la calidad de software del proyecto se basó en la norma ISO/IEC 25010 que hace referencia a la calidad de Software y se utilizó las siguientes fases: Funcionalidad, fiabilidad, usabilidad y mantenibilidad. La parte de seguridad se basó en la norma ISO 17799 que se especifica y recomienda acerca de la seguridad del software, además se trabajó con técnica de seguridad como es el algoritmo MD5 en el inicio y control de Usuarios.

Para determinar el costo y el beneficio que me pueda aportar el proyecto se aplicó el modelo de costos COCOMO II que fue creado por Boehm a finales de los años 70.

ABSTRACT

Having an information system is an inevitable need in our work context, since it is essential for the management of information, which helps us generate reliable and timely data, and these allow us to support in decision-making.

The movement of different organizations towards websites has now brought about a constant evolution in Web applications. The transfer of confidential information and execution of online processes, among others, demand functionality, reliability, usability and efficiency as some of the main characteristics of software quality.

Taking into account these considerations, it is necessary to implement an "Information System for the Allocation and Control of Functional Housing Case: General Directorate of Development and Social Welfare". This challenge implies: to offer an information service that allows to track and generate reports on accumulated debts for the payment of rent and that they are reliable, relevant and also that the information about the behavior of the rentals is stored in a database, secure on a server and protected by a restricted access for its subsequent use.

For the implementation of the information system, the programming languages: Java and the frameworks were used: JSF, Primefaces and a database created in Mysql, which are adapted to the needs and requirements of the Institution.

Keywords: System, registration, control, methodology, software.

Methodology: To start with the project, the Agile Scrum methodology was used for the development and process of the software; planning the tasks according to the user stories that are told by the Product Owner in this case the Director of the Institution, identifying the dates of each task and those responsible for the product to be finished.

To obtain the software quality of the project was based on the ISO/IEC 25010 standard that refers to the quality of Software and the following phases were used:

Functionality, reliability, usability and maintainability. The security part was based on the ISO 17799 standard that is specified and recommended about the security of the software, in addition to working with security technique such as the MD5 algorithm in the startup and control of Users.

To determine the cost and benefit that the project can bring me, the COCOMO II cost model was applied, which was created by Boehm in the late 70.

CONTENIDO

CAPITULO I MARCO REFERENCIAL	1
1.1. Introducción	1
1.2. Problema	2
1.2.1. Antecedentes	2
1.2.2. Formulación del Problema	4
1.3. Objetivos	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos	5
1.4. Justificaciones	6
1.4.1. Justificación Social	6
1.4.2. Justificación Técnica	6
1.4.3. Justificación Económica	6
1.5. Límites y Alcances.....	6
1.5.1. Limites	7
1.5.2. Alcances.....	7
1.6. Aportes	8
1.7. Metodología.....	9
CAPITULO II MARCO TEÓRICO	10
2.1. Introducción	10
2.2. Marco Institucional.....	10
2.2.1. Misión	11

2.2.2.	Visión	11
2.2.3.	Organigrama	11
2.2.4.	Procedimientos y Funciones.....	12
2.2.5.	Funcionamiento del Sistema Actual.....	13
2.3.	Metodología.....	14
2.3.1.	Metodologías Ágiles	15
2.3.2.	Scrum.....	15
2.3.2.1.	Características	16
2.3.2.2.	Elementos de la Metodología	17
2.3.2.3.	El Proceso.....	19
2.3.2.4.	Roles y Responsabilidades	22
2.4.	Tecnologías de Software	23
2.4.1.	Lenguaje de Programación Java	24
2.4.2.	Framework	25
2.4.2.1.	Framework Primefaces.....	26
2.4.2.2.	Base de Datos.....	26
2.4.2.3.	Características de Base de Datos	27
2.4.2.4.	Tipos de Base de Datos	27
2.4.2.5.	Gestor de Base de Datos Mysql.....	28
2.4.2.6.	Características de Datos Mysql	29
2.5.	Métricas de Calidad.....	30
2.5.1.	Norma ISO/IEC 25010.....	31

2.5.2.	Adecuación Funcional	31
2.5.3.	Confiabilidad.....	32
2.5.4.	Usabilidad.....	33
2.5.5.	Mantenibilidad	34
2.5.6.	Portabilidad	35
2.6.	Costos y/o Beneficios	35
2.6.1.	COCOMO.....	36
2.6.2.	Modelo COCOMO II	37
2.7.	Seguridad.....	39
2.7.1.	Seguridad Física.....	39
2.7.2.	Seguridad Lógica.....	40
	CAPITULO III MARCO APLICATIVO	45
3.1.	Introducción.....	45
3.2.	Pre-Game.....	45
3.2.1.	Recopilación de Requerimientos	45
3.2.2.	Pila de Producto (PRODUCT BACKLOG)	45
3.2.3.	Definición del cronograma de trabajo	47
3.2.4.	Análisis de riesgo	47
3.3.	Game	48
3.3.1.	Sprint Backlog	48
3.3.2.	Primera iteración (SPRINT BACKLOG 1)	49
3.3.2.1.	Planificación del desarrollo del sistema (BACKLOG 1).....	49

3.3.2.2.	Análisis de requerimientos del product Backlog (BACKLOG 2)	50
3.3.2.3.	Descripción de roles y responsabilidades (BACKLOG 3)	50
3.3.2.4.	Historia de Usuario. (ID BACKLOG 3)	53
3.3.2.5.	Diseño de la arquitectura del sistema (ID BACKLOG 4)	58
3.3.2.6.	Casos de uso (ID BACKLOG 5)	60
3.3.2.7.	Diagrama de Secuencia. (ID BACKLOG 6)	73
3.3.2.8.	Modelo Entidad Relación. (ID BACKLOG 7).....	77
3.3.2.9.	Diagrama de Clases. (ID BACKLOG 8)	77
3.3.2.10.	Diagrama Físico. (ID BACKLOG 9)	78
3.3.2.11.	Diseño de interfaces de los usuarios. (ID BACKLOG 10)	79
3.3.3.	Segunda Iteración (SPRINT BACKLOG 2)	89
3.3.3.1.	Módulo Afiliación (ID BACKLOG 1)	89
3.3.3.2.	Modulo Afiliación de Vivienda (ID BACKLOG 2)	90
3.3.3.3.	Módulo Estado de las Viviendas (ID BACKLOG 3).....	91
3.3.3.4.	Módulo Búsqueda de Locatario (ID BACKLOG 4)	92
3.3.3.5.	Módulo Historial de Alquileres (ID BACKLOG 5)	93
3.3.3.6.	Módulo Asignación de Vivienda (ID BACKLOG 6).....	94
3.3.3.7.	Módulo de Extracto de pago. (ID BACKLOG 7).....	97
3.3.4.	Tercera Iteración (SPRINT BACKLOG 3)	99
3.3.4.1.	Módulo generar descuentos mensuales (ID BACKLOG 1)	100
3.3.4.2.	Modulo Registro de Alquiler (ID BACKLOG 2).....	102
3.3.4.3.	Modulo Reporte de Moras (ID BACKLOG 3)	103

3.4.	Post-Game	104
CAPITULO IV CONTROL DE CALIDAD Y SEGURIDAD.....		110
4.1.	Introducción.....	110
4.2.	Factores de Calidad ISO/IEC 25010.....	110
CAPITULO V EVALUACIÓN DE COSTO Y BENEFICIO		123
5.1.	Introducción.....	123
5.2.	Costo Del Sistema.....	124
5.2.1.	Costo De Desarrollo Del Software	124
5.2.2.	Costo De Implantación	126
5.2.3.	Costo De Elaboración Del Proyecto.....	126
5.2.4.	Costo Total Del Software.....	126
5.3.	Beneficio.....	127
5.3.1.	Valor Actual Neto.....	127
5.3.2.	Tasa Interna De Retorno	129
5.4.	Costo / Beneficio	130
CAPÍTULO VI SEGURIDAD DEL SISTEMA		131
6.1.	Seguridad.....	131
6.2.	Organización de la Seguridad.....	131
6.3.	Clasificación y Control de Activos	131
6.4.	Gestión de Operación y de Comunicación.....	131
6.5.	Control de Accesos	131
6.5.1.	Algoritmo Md5	131

6.5.2. Desarrollo y Mantenimiento de Sistema	133
6.5.3. Seguridad Administrativa	133
6.5.4. Seguridad Lógica.....	134
CAPÍTULO VII CONCLUSIONES Y RECOMENDACIONES	135
7.1. Conclusiones	135
7.2. Recomendaciones.....	135
BIBLIOGRAFÍA	137
ANEXO	140
ANEXO “A”	141
ÁRBOL DE PROBLEMAS.....	141
ANEXO “B”	142
ÁRBOL DE OBJETIVOS.....	142

ÍNDICE DE FIGURAS

Figura 2.1. Organigrama de la Dirección General Desarrollo y Bienestar Social .	12
Figura 2.2. Esquema general de Scrum.	16
Figura 2.3. Roles que el equipo debe realizar	19
Figura 2.4. El proceso de Scrum.	20
Figura 2.5. Ejemplo de Pila de Sprint con hoja de cálculo	21
Figura 2.6. Roles Scrum.....	23
Figura 3.1. Diseño de la arquitectura del sistema – Administrador	59
Figura 3.2. Diseño de la arquitectura del sistema – Responsable de viviendas...	59
Figura 3.3. Diagrama de caso de uso: Registro de Locatario.	60
Figura 3.4. Diagrama de caso de uso: Registro de vivienda.....	61
Figura 3.5. Diagrama de caso de uso: Estado de vivienda	62
Figura 3.6. Diagrama de caso de uso: Historial de alquiler.....	63
Figura 3.7. Diagrama de caso de uso: Asignación de viviendas.....	64
Figura 3.8. Diagrama de caso de uso: Descuentos Mensuales.	65
Figura 3.9. Diagrama de caso de uso: Registro de pagos	67
Figura 3.10. Diagrama de caso de uso: Control de pago.....	69
Figura 3.11. Diagrama de caso de uso: Ingresos por alquiler	70
Figura 3.12. Diagrama de caso de uso: Estimación de ingresos	71
Figura 3.13. Diagrama de caso de uso: Administración de usuario	72
Figura 3.14. Diagrama De Secuencia: Registro de Locatario	73

Figura 3.15. Diagrama De Secuencia: Estado de vivienda	73
Figura 3.16. Diagrama De Secuencia: Historial de alquiler	74
Figura 3.17. Diagrama De Secuencia: Asignación de viviendas	74
Figura 3.18. Diagrama De Secuencia: Descuentos Mensuales	75
Figura 3.19. Diagrama De Secuencia: Registro de pagos	75
Figura 3.20. Diagrama De Secuencia: Control de pago.....	76
Figura 3.21. Diagrama De Secuencia: Estimación de ingresos	76
Figura 3.22. Modelo Entidad Relación.....	77
Figura 3.23. Diagrama de Clases	78
Figura 3.24. Diagrama Físico de la base de datos	79
Figura 3.25. Registro de Datos del Locatario	80
Figura 3.26. Registro de Datos de vivienda	80
Figura 3.27. Reporte de los estados de las Viviendas	81
Figura 3.28. Búsqueda de locatario	81
Figura 3.29. Reporte de historial de alquileres	82
Figura 3.30. Extracto de pagos	82
Figura 3.31. Registro de la Solicitud.....	83
Figura 3.32. Asignación de Vivienda	83
Figura 3.33. Plan de pago	84
Figura 3.34. Confirmación de datos.....	84
Figura 3.35. Generar planilla de descuentos	85
Figura 3.36. Importar Planilla	85

Figura 3.37. Registro Descuento	86
Figura 3.38. Morosos Vigentes.....	86
Figura 3.39. Reporte de Ingresos	87
Figura 3.40. Moras Históricas.....	87
Figura 3.41. Estimación de Ingresos	88
Figura 3.42. Suma de Saldo.....	88
Figura 3.43. Afiliación de Locatario	90
Figura 3.44. Afiliación de Vivienda	91
Figura 3.45. Estado de las Viviendas	92
Figura 3.46. Búsqueda de Locatario.....	93
Figura 3. 47. Búsqueda de Locatario.....	94
Figura 3. 48. Historial de Alquileres.....	94
Figura 3. 49. Registro de Solicitud.....	95
Figura 3.50. Asignación de Vivienda	96
Figura 3.51. Calculo del Pago	96
Figura 3.52. Registrar Datos	97
Figura 3. 53. Búsqueda de Locatario.....	98
Figura 3. 54. Historial de Alquileres.....	98
Figura 3. 55. Extracto de pago	99
Figura 3. 56. Generar descuentos mensuales	101
Figura 3. 57. Generar descuentos mensuales.....	101
Figura 3. 58. Registro de Pago de Alquiler	102

Figura 3.59. Registro de Pago de Alquiler	103
Figura 3. 60. Reporte de Moras.....	104

ÍNDICE DE TABLAS

Tabla 2.1. Ejemplo de la Pila de Producto	21
Tabla 3.1. Product Backlog.....	46
Tabla 3.2. Cronograma de trabajo para el desarrollo del producto software	47
Tabla 3.3. Tabla de Riesgos y sus efectos	48
Tabla 3.4. Sprint Backlog 1	49
Tabla 3.5. StackHolder Usuario 1	51
Tabla 3.6. StackHolder Usuario 2	51
Tabla 3.7. StackHolder Desarrollador	52
Tabla 3.8. Usuario – Responsable de viviendas	52
Tabla 3.9. Usuario – Auxiliar vivienda.....	53
Tabla 3.10. Usuario – Administrador.....	53
Tabla 3.11. Registro de Locatario	54
Tabla 3.12. Registro de vivienda.....	54
Tabla 3.13. Gestión de Reportes – Estado de vivienda	54
Tabla 3.14. Búsqueda de locatario	55
Tabla 3.15. Historial de alquiler	55
Tabla 3.16. Asignación de viviendas.....	55
Tabla 3.17. Extractos de pagos de alquiler	56
Tabla 3.18. Generar Descuentos Mensuales	56
Tabla 3.19. Registro de pago.....	56
Tabla 3.20. Control de pago	57

Tabla 3.21. Reporte de ingresos por alquiler	57
Tabla 3.22. Calculo de deudas	57
Tabla 3.23. Resumen estimación de ingresos	58
Tabla 3.24. Administra usuario	58
Tabla 3.25. Especificación de Caso de uso: Registro de Locatario.....	60
Tabla 3.26. Especificación de Caso de uso: Registro de vivienda.	62
Tabla 3.27. Especificación de Caso de uso: Estado de vivienda	63
Tabla 3.28. Especificación de Caso de uso: Historial de alquiler	64
Tabla 3.29. Especificación de Caso de uso: Asignación de viviendas.	65
Tabla 3.30. Especificación de Caso de uso: Descuentos Mensuales.....	66
Tabla 3.31. Especificación de Caso de uso: Registro de pagos.....	68
Tabla 3.32. Especificación de Caso de uso: Control de pago	69
Tabla 3.33. Especificación de Caso de uso: Ingresos por alquiler	70
Tabla 3.34. Especificación de Caso de uso: Estimación de ingresos.....	71
Tabla 3.35. Especificación de Caso de uso: Administración de usuario.....	72
Tabla 3.36. Sprint Backlog 2.....	89
Tabla 3.37. Sprint Backlog 3.....	100
Tabla 3.38. Prueba de desarrollo sprint 2 Módulo afiliación	105
Tabla 3.39. Prueba de desarrollo sprint 2 Módulo afiliación de vivienda	105
Tabla 3.40. Prueba de desarrollo sprint 2 Módulo reporte de viviendas	106
Tabla 3.41. Prueba de desarrollo sprint 2 Módulo consulta alquiler	106
Tabla 3.42. Prueba de desarrollo sprint 2 Módulo historial de alquiler	107

Tabla 3.43. Prueba de desarrollo sprint 2 Módulo asignación de vivienda	107
Tabla 3.44. Prueba de desarrollo sprint 2 Módulo extracto de pago	108
Tabla 3.45. Prueba de desarrollo sprint 3 Módulo generar descuentos mensuales	108
Tabla 3.46. Prueba de desarrollo sprint 3 Módulo registro masivo de amortizaciones.....	109
Tabla 3.47. Prueba de desarrollo sprint 3 Módulo registro de amortización	109
Tabla 4.1. Entradas de usuario	112
Tabla 4.2. Salidas de usuario	112
Tabla 4.3. Peticiones de usuario.....	113
Tabla 4.4. Archivos del sistema	113
Tabla 4.5. Interfaces externas	114
Tabla 4.6. Factores de ponderación	114
Tabla 4.7. Valores de Ajuste de Complejidad	115
Tabla 4.8. Valores de Ajuste de Complejidad	117
Tabla 4.9. Evaluación de usabilidad	119
Tabla 4.10. Valores requerida por el IMS.....	120
Tabla 4.11. Calidad global del sistema	122
Tabla 5.1. Coeficientes a, b, c y d (COCOMO II)	124
Tabla 5.2. Conversión de Puntos de Función a KDLC	125
Tabla 5.3. Costo de elaboración del proyecto.....	126
Tabla 5.4. Costo total del software.....	127
Tabla 5.5. Calculo del VAN.....	128

Tabla 5.6. Criterio de interpretación del VAN.....	128
Tabla 5.7. Cálculo de la tasa interna de retorno.....	129

MARCO REFERENCIAL

1.1. Introducción

Los sistemas de información han favorecido el desarrollo de metodologías que analizan la realidad, diseñando e implementando sistemas que buscan incrementar cada vez más la obtención de procesos de información y aminorar la complejidad en el cual se lleva la documentación, cuando se presentan estados y relaciones múltiples en la información, obstáculo que reviste grandes esfuerzos de obtención, procesamiento, almacenamiento y modificación, necesarios para garantizar el normal funcionamiento de sus procesos y una adecuada toma de decisiones.

El proceso de administrar la información generada por computadora difiere de manera considerable del proceso de manejar los datos producidos en forma manual; por lo general debemos administrar una mayor cantidad de información computacional. Los costos de administración y mantenimiento pueden aumentar a ritmos alarmantes, y a menudo los usuarios consideran este tipo de información con menos escepticismo que la que se obtiene de otras fuentes (Kendall & Kendall, 2011).

En nuestra sociedad la utilización de computadoras y de los sistemas de información se ha convertido en un aspecto indispensable para todas las instituciones, puesto que ayudan a los gerentes a mantener ordenada su compañía, a analizar todo lo que por ella pasa y a crear nuevos productos que coloquen en un buen lugar a la institución. Los sistemas de información recolectan, procesan, almacenan y distribuyen información y de esta manera mantienen a la institución actualizada.

Dentro de las Fuerzas Armadas y en específico de la Fuerza Aérea Boliviana, existen instituciones que velan por el bienestar social de sus afiliados y una de sus funciones, es alquilar las Viviendas Funcionales al personal de la

Institución. Tal es el caso de la Dirección General de Desarrollo y Bienestar Social (DIGEDEBIS), la cual tiene como una de sus actividades, el de poder brindar una mejor calidad de vida a los miembros de la Fuerza Aérea Boliviana (FAB), facilitándoles la asignación de Viviendas.

El presente trabajo está enfocado a DIGEDEBIS, con el fin de coadyuvar a que este proceso se lo pueda realizar de una manera más rápida, segura y cómoda, se plantea desarrollar un sistema de información para gestión y administración de Viviendas para que los usuarios puedan tener un acceso a la información más rápida, eficiente y de esta manera prestar un mejor servicio a los miembros de la FAB., en cuanto a la otorgación de vivienda, generación de reportes, consultas sobre sus extractos de pagos, registro de pagos por planilla, control de pago por alquiler hasta su cancelación y evitar la duplicidad en la asignación de viviendas.

1.2. Problema

1.2.1. Antecedentes

Con el propósito de dar mayor eficiencia al quehacer profesional en los ámbitos de desarrollo y bienestar social de los miembros de la institución, el Comando General de la Fuerza Aérea Boliviana, dispuso la creación de la “Dirección General de Desarrollo y Bienestar Social” (DIGEDEBIS), el 15 de marzo de 1988, mediante Orden del Día N° 42/88 con el objetivo de alcanzar un desarrollo sostenido y equilibrado que viabilice programas y proyectos dirigidos a coadyuvar prioritariamente el bienestar social de la institución y la familia aeronáutica. (MANUAL, 2021).

En la actualidad la DIGEDEBIS, cuenta 404 (cuatrocientos cuatro) viviendas funcionales, con ubicación en las regionales de: La Paz, Uyuni, Cochabamba, Santa Cruz, Tarija, Trinidad, Cobija, Puerto Suarez, Riberalta, Robore y Villa Montes. Con el pasar del tiempo la dirección llega a manejar y acumular una gran cantidad de información, debido a que el personal militar es cambiado de

destino constantemente, es así que al momento de asignar o buscar viviendas libres o verificar la cancelación total del alquiler, este presenta muchos problemas debido a que el proceso es realizado de manera manual, ocasionando errores en la asignación de vivienda, deficiencia en el control de las cancelaciones del alquiler y mala admiración en proceso de los descuentos.

Actualmente no existe un sistema integrado para el control y asignación de las viviendas funcionales, por lo que acudieron a utilizar una hoja de cálculos en hoja Excel, en la cual son registradas toda la información, ocasionando que no se pueda contar la información completa, fidedigna y exista retraso en los procesos de control, seguimiento y asignación de viviendas, debido a que la información almacenada no esté completa y exista duplicidad de registro.

Antecedentes de proyectos similares

Proyecto de Grado “Sistema de Información Para el Control y Seguimiento de Recursos Financieros Caso: Condominio LA JOYA” presentado por el postulante Juan Carlos Olivares Chiara, cuyo objetivo general es implementar un sistema de información, para un adecuado control y seguimiento del flujo de efectivo y procesos relacionados en la administración de los recursos económicos del Condominio “La Joya”, el cual hace uso de herramientas tradicionales para el desarrollo de este sistema. (Juan Carlos Olivares, 2017).

Proyecto de Grado Sistema Web de Seguimiento de Ventas y Cobranzas Caso: Agencia de Viajes “COSMOS TRAVEL AND SERVICES S.R.L.” presentado por el postulante Luis Alfredo Colmena Vargas, cuyo objetivo general es Implementar un Sistema Web de Seguimiento de Venta y Cobranza para la empresa “COSMOS TRAVEL AND SERVICES S.R.L.” que permita realizar un monitoreo confiable, oportuno de su movimiento económico y la generación de información rápida y actualizada para la toma de decisiones y proyecciones futuras de la empresa. (Luis Alfredo Colmena Vargas, 2015).

De acuerdo a la información recopilada y estudiada todas las necesidades en cuanto a la información que maneja la Dirección General de Desarrollo y Bienestar Social y de acuerdo al árbol de problemas (Ver Anexo "A"), se identificaron los siguientes problemas específicos:

- No se tiene un registro adecuado y organizado del personal de la F.A.B., ocasionando demora en la búsqueda de sus datos.
- La información con la que se cuenta no es completa ni actualizada lo que provoca demora en la asignación o rechazo de vivienda.
- Existe retraso en la emisión de reportes de extractos individuales de pago de alquiler, ocasionado desconfianza en el proceso.
- No existe confianza en los descuentos por alquiler enviados al Ministerio de Defensa, proceso que es realizado en hoja Excel.
- Los registros de los descuentos mensuales por alquiler de viviendas son realizados manual mente, provocando que el proceso devuelva datos importunamente.
- Los procesos para el control de pagos y verificación de las deudas pendientes son realizados manual mente, provocando la obtención de datos incompletos.
- No se cuenta con reportes necesarios y actualizados del estado de las viviendas funcionales, para proporcionar apoyo en la toma de decisiones en la dirección.

1.2.2. Formulación del Problema

¿Cómo mejorar el proceso de asignación y control de las viviendas funcionales para la Dirección General de Desarrollo y Bienestar Social (DIGEDEBIS) para que le permita realizar un trabajo eficiente y prestar un mejor servicio a sus afiliados?

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema de información automatizado que permita realizar un mejor servicio, confiable, oportuno y facilitar información actualizada en la asignación y control de las viviendas funcionales para la Dirección General de Desarrollo y Bienestar Social.

1.3.2. Objetivos Específicos

De acuerdo al árbol de objetivos (Ver Anexo "B"), se identificaron los siguientes objetivos específicos:

- Desarrollar una base de datos que nos facilite el registro, consulta y control del personal de la F.A.B., a los cuales se le asignara una vivienda en un tiempo determinado.
- Agilizar el proceso de asignación de viviendas a través de la obtención de información actualizada y en forma automática.
- Proporcionar confianza en el proceso mediante la emisión de extractos individuales de pagos por alquiler actualizados.
- Automatizar el proceso para proporcionar seguridad en los descuentos enviados al Ministerio de Defensa.
- Facilitar el proceso mediante un registro automático de los descuentos por alquiler mensuales, para la obtener datos confiables.
- Proporcionar reportes que son necesarios para el control de pago y deudas pendientes de la asignación de viviendas, y así obtener datos completos.
- Desarrollar el módulo de reportes.

1.4. Justificaciones

1.4.1. Justificación Social

El desarrollo del nuevo sistema propuesto beneficiará a los usuarios de la Dirección General de Desarrollo y Bienestar Social, puesto que con el mismo se obtendrá la información confiable y real para agilizarán los trámites de asignación de vivienda, proporcionando así un mejor control.

1.4.2. Justificación Técnica

El proyecto se justifica tecnológicamente ya que se realizara en el lenguaje de programación Java juntamente al entorno de desarrollo NetBeans y Framework Primeface, una herramienta para el desarrollo de aplicaciones que crea webs utilizando Java Enterprise y la Gestión de Base de Datos de MySQL, en cuanto a los equipos se utilizaran los que actualmente cuenta la Dirección General de Desarrollo y Bienestar Social con un procesador I3 de segunda generación con una memoria RAM de 4 GB, y un red interna que ayudan a que se implemente el Sistema de Información.

1.4.3. Justificación Económica

El desarrollo del sistema será implementado sobre una plataforma web de libre distribución, donde el lenguaje de programación como el gestor de la base de datos no presenta costos en su adquisición y/o licencias de funcionamiento, por lo que en la ejecución y puesta en marcha del sistema, la Dirección General de Desarrollo y Bienestar Social no necesitaran realizar inversiones de software.

1.5. Límites y Alcances

El desarrollo del presente sistema propuesto se propone los siguientes límites y alcances:

1.5.1. Limites

La información de la Dirección General de Desarrollo y Bienestar Social en sus predios en la ciudad de La Paz tendrá la información necesaria en las diferentes oficinas de esta dirección de acuerdo a restricciones y atribuciones según requiera cada usuario de esta institución.

1.5.2. Alcances

El sistema de información para la asignación y control de las viviendas funcionales comprende los módulos que se especifican a continuación:

- Agilizar el proceso de asignación de viviendas.
 - Registro de solicitante.
 - Asignación de vivienda.
 - Verificación del locatario en filiación.
 - Verificación de historial de viviendas asignadas del solicitante.
 - Determinar costo del alquiler y plazo.
 - Dar de alta el alquiler.
- Reportes de extractos de pagos de alquiler individuales
 - Buscar locatario.
 - Visualizar historial de viviendas asignadas.
 - Visualizar vivienda alquilada.
- Automatizar el proceso de descuentos.

- Generar reportes de los descuentos mensuales por alquiler de vivienda.
- Exportar listas de los descuentos en formato txt.
- Facilitar el registro de los descuentos.
 - Registro automático de los descuentos por vivienda
 - Generar reportes de los pagos registrados.
- Proporcionar reportes que son necesarios para el control de pagos y deudas pendiente.
 - Proporcionar reportes mensuales del incumplimiento de pago de vivienda.
 - Descontar las deudas pendientes.
- Emisión de reportes.
 - Detalle de las viviendas alquiladas y libres.

1.6. Aportes

El aporte principal del presente proyecto es la Implementación del sistema el cual contribuirá a una mejor gestión y control en la asignación de viviendas funcionales de la F.A.B., facilitando el manejo de información en tiempo real, con información confiable al personal administrativo y miembros de esta institución.

Además, el presente proyecto de grado tiene como objeto facilitar información oportuna para la asignación y control de las viviendas funcionales para la Dirección General de Desarrollo y Bienestar Social, a través del desarrollo del sistema de información automatizado.

1.7. Metodología

La metodología SCRUM es una manera de trabajo en equipo en el que se busca obtener resultados funcionales de manera efectiva para proyectos de programación Web, Aplicaciones Móviles, incluso utilizado para hackatones.

A Scrum se considera como un paradigma de Metodología de Desarrollo Ágil, definiendo la forma de abordar un proceso de desarrollo de software de forma ágil y liviana, a través de la descripción de un conjunto de roles, componentes y organización de la actividad diaria. El punto de partida de esta metodología consiste en la división del trabajo completo, en distintos módulos, que puedan ser abordados dentro el rango establecido y/o factible a la complejidad que este lleva (Trigas, 2008).

SCRUM, es aplicable a proyectos con tiempo y recursos limitados. Algunas consideraciones a tener en cuenta cuando se usa esta metodología son:

- La entrega del proyecto debe realizarse a tiempo, no sobrepasando los costes estimados y entregando software de calidad.
- Solo es necesario que un paso del desarrollo esté lo suficientemente completo como para poder pasar al siguiente, así la siguiente iteración podrá ser comenzada sin retrasos.
- La metodología incluye técnicas de desarrollo y gestión de proyectos.
- Puede ser aplicado tanto a proyectos nuevos como a proyectos ya existentes para su ampliación.
- La evaluación de riesgos se debe centrar en los entregables y no en el proceso de desarrollo ni en la documentación o requisitos.
- Las estimaciones se deben realizar según las funcionalidades y no según las líneas de código.

MARCO TEÓRICO

2.1. Introducción

El presente capítulo detalla algunos de los conceptos y definiciones que se necesitan para el desarrollo del proyecto, con el fin de ayudar a la familiarización del tema y son la base para la descripción y explicación del problema a solucionar.

El marco teórico se divide en varias partes como ser el marco institucional de la organización en donde se realizará el proyecto de grado de la cual se destaca la visión, misión y organigrama de la institución, en este apartado se detalla cómo funciona el sistema actual y quienes son los actores directos que hacen que funcione. Así también se detalla la metodología que utilizaremos que en nuestro caso será SCRUM y las herramientas que se utilizarán para dicho proyecto como ser el gestor de base de datos MySQL, Java juntamente al entorno de desarrollo NetBeans y framework Primeface que hacen que el trabajo sea menor. Otra parte muy importante que se detalla de manera minuciosa son las métricas de calidad de la cual nos basaremos en la norma ISO/IEC 25010 el cual identifica seis atributos claves para calidad como ser funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y eficiencia.

Finalizando veremos cómo obtener los costos y beneficios del sistema en esta parte utilizaremos el modelo desarrollado por Barry Boehm COCOMO básico y además conoceremos la seguridad que será implementada en el sistema a nivel físico y lógico.

2.2. Marco Institucional

Con el propósito de dar mayor eficiencia al quehacer profesional en los ámbitos de desarrollo y bienestar social de los miembros de la institución, el Comando General de la Fuerza Aérea Boliviana, dispuso la creación de la “Dirección

General de Desarrollo y Bienestar Social" (DIGEDEBIS), el 15 de marzo de 1988, mediante Orden del Día N° 42/88 con el objetivo de alcanzar un desarrollo sostenido y equilibrado que viabilice programas y proyectos dirigidos a coadyuvar prioritariamente el bienestar social de la institución y la familia aeronáutica. (MANUAL, 2021).

DIGEDEBIS presta sus servicios en los predios del ex Comando General de la Fuerza Aérea Boliviana ubicada en la ciudad de La Paz a todo el personal militar y civil del servicio activo, en la asignación de viviendas en las regionales de: La Paz, Uyuni, Cochabamba, Santa Cruz, Tarija, Trinidad, Cobija, Puerto Suarez, Riberalta, Robore y Villa Montes. En la actualidad la F.A.B. cuenta con cuatro mil ciento ochenta y ocho afiliados.

2.2.1. Misión

"Planificar, preparar, desarrollar y controlar las actividades de Desarrollo y Bienestar Social de la F.A.B., a fin de lograr el potenciamiento de las Institución y el bienestar de sus miembros".

2.2.2. Visión

Prestar un servicio más eficiente en la asignación de viviendas funcionales acorde a requerimiento y necesidades del personal de la F.A.B., aperturar Hoteles y Círculos Aeronáuticos en todas las capitales de departamentos para ofrecer servicios de excelencia basados en el bienestar social y calidad humana.

2.2.3. Organigrama

La Dirección General de Desarrollo y Bienestar Social está conformada como se muestra en la figura 2.1.

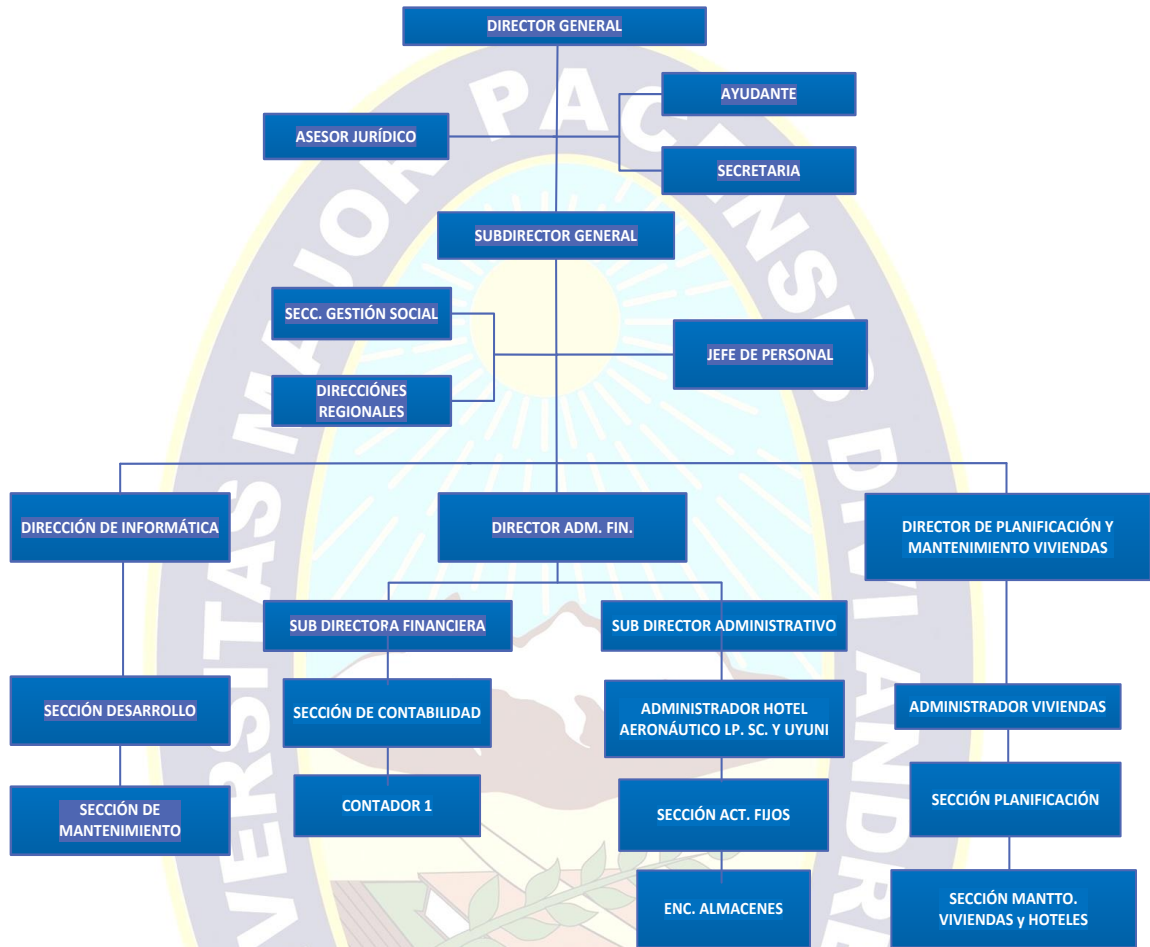


Figura 2.1. Organigrama de la Dirección General Desarrollo y Bienestar Social

Fuente: [DIGEDEBIS, 2021]

2.2.4. Procedimientos y Funciones

Las actividades que se realizan en la DIGEDEBIS siempre están en el marco del trabajo coordinado para la asignación y control de las viviendas funcionales, a todo el personal militar y civil del servicio activo.

Los recursos humanos con los que cuenta para atender a todo el personal son de la siguiente manera:

Administrador de Viviendas: Establece y coordina los procedimientos de control, para un manejo óptimo de los recursos con los que se cuenta para la asignación de viviendas y sus respectivos cobros. Programa las asignaciones periódicamente, de manera que permitan cubrir la demanda para la asignación de viviendas.

Secretaria: Es la encargada directa de atender y registrar las solicitudes de viviendas y brinda información al personal sobre los requisitos.

Contador: Parte operativa de la Dirección Administrativa Financiera, encargado de registrar los descuentos por alquiler, y generar el reporte de pagos.

Sección de Mantenimiento Vivienda y Hoteles: Es responsable de registrar y controlar los pagos por vivienda mensuales por papeleta de pago, registrar en el sistema las solicitudes de viviendas y generar reportes individuales de los extractos de pago, también es el encargado de enviar mensualmente los descuentos al Ministerio de Defensa y proporcionar información a gerencia general.

2.2.5. Funcionamiento del Sistema Actual

En la actualidad la Dirección General de Desarrollo y Bienestar Social no cuenta con un sistema informático, ocasionando que los procesos para la asignación y control de viviendas, generación de descuentos, control de pagos y reportes, se realicen de forma manual, produciendo equivocaciones, retraso en el proceso y trabajo excesivo por parte de los empleado, causando susceptibilidad y desconfianza, debido a que no se cuenta con suficiente información, lo que produce que se acuda a consultar tablas realizadas en Excel y sea de forma manual la revisión de la documentación.

Una vez que el Afiliado solicite la asignación de una vivienda del tipo familiar o soltero, la secretaria recepciona todos los datos necesarios y le asigna una hoja de ruta.

Una vez resepcionada la documentación se procede a verificar su historial de las viviendas asignadas, reportar si la vivienda solicita está disponible, en caso de que tuviera alguna deuda, se informa al solicitante que su trámite demorara una semana.

Mensualmente se revisa las listas de las viviendas asignadas y determinan quiénes tienen que ser descontados el proceso es realizado mensualmente y demora una a dos semanas.

Cuando retorna los descuentos por alquiler se procede a registrar en el Excel determinando a quien corresponde dicho descuento.

2.3. Metodología

Metodología es el estudio o elección de un procedimiento pertinente para alcanzar un determinado objetivo. Algunas de estas metodologías han venido siendo iguales o han tenido sus cambios respectivos según la situación lo amerite, ya que se hacen presentes diferentes factores como el tiempo, eficacia, dinero, requerimientos, entre otros.

Existen metodologías que han venido acompañando el avance tecnológico desde sus inicios y más en el desarrollo de software. Tanto así que han proporcionado la manera de solucionar situaciones de forma práctica obteniendo así los resultados esperados en menos tiempo. Este ha sido uno de los objetivos principales de los programadores, para hacer que la información sea utilizada en cualquier hora y lugar.

Las metodologías han sido utilizadas para la realización de proyectos tecnológicos y científicos desde su invención, consideradas como un proceso de planificación, la cual consiste en un conjunto de actividades que se encuentran interrelacionadas y coordinadas. La razón de un proyecto es alcanzar objetivos específicos dentro de los límites que imponen un presupuesto, calidades establecidas previamente y un lapso de tiempo previamente definido. La gestión de proyectos es la aplicación de

conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto.

2.3.1. Metodologías Ágiles

Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental. La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Dentro de las metodologías existentes para la realización de proyectos a nivel de programación de software, se encuentre la Metodología Scrum, la cual se utilizó en la generación de productos de software durante el desarrollo de la pasantía de investigación y la cual será descrita a lo largo de este documento.

2.3.2. Scrum

Scrum es un proceso ágil para desarrollar software que fue aplicado por primera vez por Ken Schwaber y Jeff Sutherland, quienes lo documentaron en detalle en el libro Agile Software Development with Scrum. Esta metodología centra su atención en las actividades de Gerencia y no especifica prácticas de Ingeniería. Fomenta el surgimiento de equipos autodirigidos cooperativos y aplica

inspecciones frecuentes como mecanismo de control. (Schwaber y Sutherland, 2013)

Scrum parte de la base de que los procesos definidos funcionan bien solo si las entradas están perfectamente definidas y el ruido, ambigüedad o cambio es muy pequeño. Por lo tanto, resulta ideal para proyectos con requerimientos inestables, ya que fomenta el surgimiento de los mismos.

El ciclo de vida definido por Scrum es incremental iterativo y se caracteriza por ser muy adaptable, ver Figura 2.2.



Figura 2.2. Esquema general de Scrum.

Fuente: [Sv, 2016]

2.3.2.1. Características

Existen varias características de la metodología de desarrollo Scrum, a continuación, se listará las principales características (Peralta, 2003):

- Equipos auto dirigidos
- Utiliza reglas para crear un entorno agile de administración de procesos
- No prescribe prácticas específicas de ingeniería
- Los requerimientos se capturan como ítems de la lista Product Backlog
- El producto se construye en una serie de Sprints desde una semana a un mes de duración

2.3.2.2. Elementos de la Metodología

Scrum no requiere ni provee prácticas de Ingeniería. En lugar de eso, especifica prácticas y herramientas de gerencia que se aplican en sus distintas fases para evitar el caos originado por la complejidad e imposibilidad de realizar predicciones [SCRUMstudy, 2013].

a) Product Backlog List

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint.

b) Sprint

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

Un Sprint tiene una duración planificada de entre una semana y un mes. No es posible introducir cambios durante el Sprint, por lo tanto para planificar su

duración hay que pensar en cuanto tiempo puedo comprometerme a mantener los cambios fuera del Sprint. Dependiendo del tamaño del sistema, la construcción de un reléase puede llevar entre 3 y 8 Sprints.

c) Sprint Backlog

Es una lista que tiene los ítems de la Product Backlog List que van a ser implementados en el siguiente Sprint y es el punto de entrada de cada Sprint.

Los ítems son seleccionados por el Scrum Team, el Scrum Master y el Product Owner en la Sprint Planning Meeting a partir de la priorización de los ítems y los objetivos que se marcaron para ese Sprint. A partir de los objetivos a cumplir durante el Sprint el Scrum Team determina que tareas debe desempeñar para cumplir el objetivo. El Manager no asigna tareas a los individuos y tampoco toma decisiones por el equipo. El equipo puede agregar nuevas tareas o remover tareas innecesarias en cualquier momento si lo considera necesario para cumplir el objetivo. Pero el Sprint Backlog solo puede ser modificado por el equipo. Las estimaciones se actualizan cada vez que aparece una nueva información.

d) Stabilization Sprint

En estos Sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad. Suelen aplicarse cuando se prepara un producto para el release. Son útiles cuando se están realizando pruebas beta, se está introduciendo a un equipo en la metodología de Scrum o cuando la calidad de un producto no alcanza los límites esperados.

e) Scrum of Scrums o Metascrum

Los equipos de Scrum suelen tener entre 5 y 10 personas, sin embargo esta metodología ha sido aplicada en proyectos que involucran a más de 600 personas. Esto ha sido llevado a cabo dividiendo a los accionistas en equipos pequeños de hasta 10 personas aproximadamente. Y definiendo

jerárquicamente personas que pertenecen a dos equipos, es decir, además de su rol específico dentro de un equipo tienen el rol de enlace en un equipo superior donde tienen que tomar en cuenta la pila de productos, los Scrum diario y sprint luego la demostración como se ve en la figura 2.3.



Figura 2.3. Roles que el equipo debe realizar

Fuente: [Proyscrum]

2.3.2.3. El Proceso

Según (Palacios, Juan, 2008). Scrum consta de tres fases: Pre Game, Development y Postgame que se describen en la Figura 2.4. La fase del Pre Game incluye dos sub fases: Plannig y Architecture.

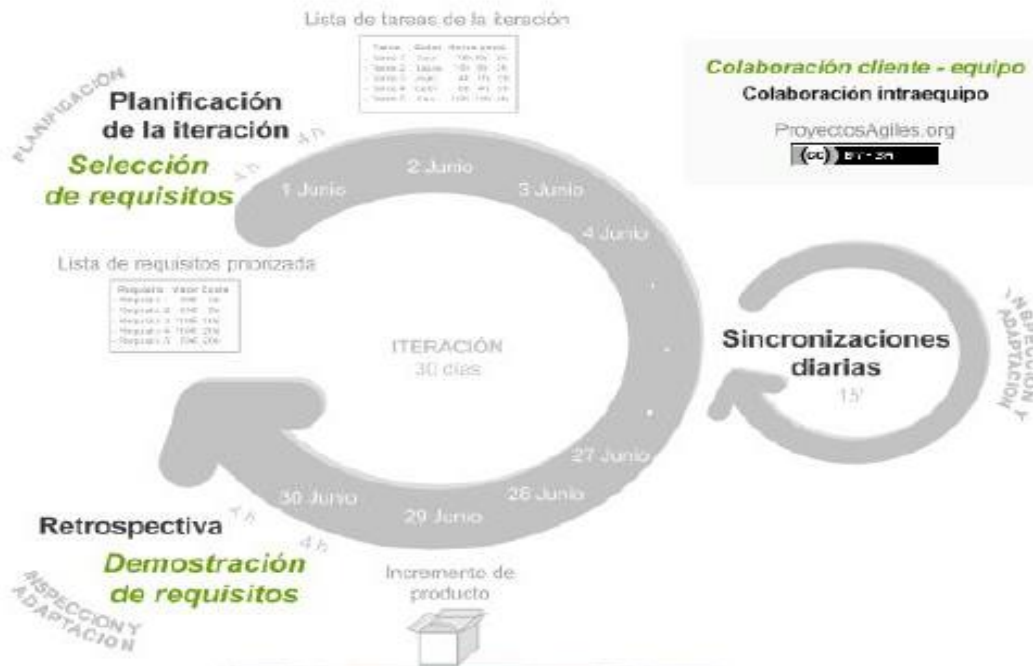


Figura 2.4. El proceso de Scrum.

Fuente: [PROYSCRUM]

Pre-Game

Es la primera fase del proceso de Scrum donde se realiza el planning (planeación), architecture (arquitectura).

Planning

Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. Un ejemplo de pila de producto se ilustra en la tabla 2.1.

Tabla 2.1. Ejemplo de la Pila de Producto

ID	PRIORIDAD	DESCRIPCIÓN	EST.	POR
1	Muy Alta	Plataforma Tecnológica	30	AR
2	Muy Alta	Interfaz de usuario	40	LM
3	Muy Alta	Un usuario se registra en el sistema	40	LM
4	Alta	El operador define el flujo y texto de una asignación	60	AR
5	Alta	XXX	999	CC

Fuente: (Palacios, 2008)

Game

En esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. Una pila de Sprint se ilustra en la Figura 2.5.

PILA DEL PRODUCTO					OBJETIVO DEL SPRINT			
Categoría	Tarea	Responsable	Estimado en horas	Estado	Crear y publicar versión básica del sitio web público			
Diseño	Crear diseño de base de datos	Juan	24	Completo	24	16	8	4
Diseño	Validar diseño de base de datos	Pedro	4	Completo	4	4	4	4
Desarrollo	Contratar servicio de hosting	Pedro	4	Completo	4	2		
Desarrollo	Crear layout y estilos de sitio web	Maria	16	Activo	8	8	4	2
Desarrollo	Crear página principal	Maria	24	Pendiente	24	24	24	24
Desarrollo	Mostrar resúmenes de noticias por sección	Juan	16	Pendiente	16	16	16	8
Desarrollo	Crear banners de publicidad	Luis	24	Pendiente	24	24	24	24
Desarrollo	Visualizar un Artículo	Luis	8	Pendiente	8	8	8	8
Desarrollo	Imprimir un Artículo	Luis	4	Pendiente	4	4	4	4

Figura 2.5. Ejemplo de Pila de Sprint con hoja de cálculo

Fuente: (Palacios, 2008)

Post-Game

Cierre: Esta fase trata sobre la preparación para la liberación, incluyendo la documentación final, las pruebas preparadas antes de la liberación y la liberación. El sistema está listo para ser liberado y es en esta etapa en la que se realiza la integración, pruebas del sistema y documentación.

2.3.2.4. Roles y Responsabilidades

Scrum Master

Es un rol de administración debe asegurar que el proyecto se está llevando a cabo de acuerdo con las prácticas, valores y reglas de Scrum y que funciona según lo planeado. Su principal trabajo es remover impedimentos y reducir riesgos del producto. Este rol suele ser desempeñado por un Gerente de Proyecto o Líder de equipo. (Palacios, Juan, 2008).

Product Owner

Es el responsable del proyecto, administra, controla y comunica la Backlog List. Es el responsable de encontrar la visión del producto y reflejarla en la Backlog List. Generalmente esta persona puede ser el Product Manager, Marketing, Internal Customer (Peralta, 2003).

Scrum Team

Típicamente es un equipo de entre 5 y 10 personas cada una especializada en algún elemento que conforma los objetivos a cumplir, por ejemplo: Programadores, Diseñadores de Interfaz de usuarios. La dedicación de los miembros del equipo debería ser full-time con algunas excepciones (Peralta, 2003).

Customer

El cliente participa en las tareas que involucran la lista Product Backlog List.

Management

Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. Participa en la selección de objetivos y requerimientos y en la selección del Scrum Owner. Tiene la responsabilidad de controlar el progreso y trabaja junto con el Scrum Master en la reducción de la Product Backlog List (Peralta, 2003).

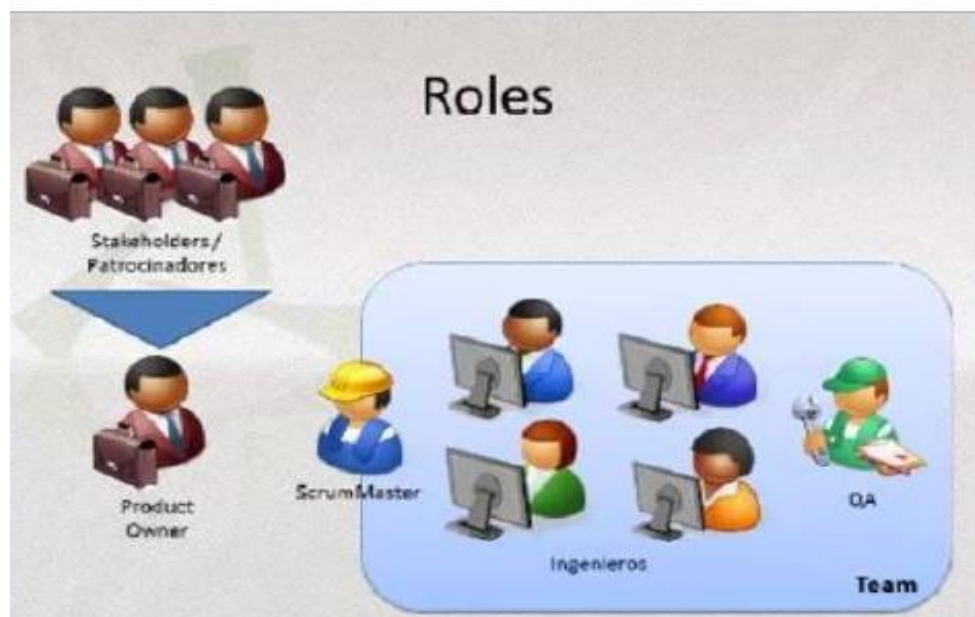


Figura 2.6. Roles Scrum

Fuente: (Autentia, 2013)

2.4. Tecnologías de Software

Las herramientas CASE son software de apoyo al desarrollo, mantenimiento y documentación informatizados de software.

Quedan, pues, principalmente las herramientas que ayudan a aplicar técnicas concretas de desarrollo y mantenimiento de software y por eso gestionan información sobre los elementos y conceptos que se utilizan en los métodos de desarrollo.

2.4.1. Lenguaje de Programación Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados. (Java, 2021).

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. (Java, 2021).

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath. (Java, 2021).

2.4.2. Framework

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírseles. (coursehero.com, 2021).

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (coursehero.com, 2021).

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. (coursehero.com, 2021).

- Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

Java Server Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML- based User-interface Language, lenguaje basado en XML para la interfaz de usuario). JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir

un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para Java Server Pages que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

2.4.2.1. Framework Primefaces

PrimeFaces es una librería de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces. (es.wikipedia.org, 2021).

2.4.2.2. Base de Datos

Según (maestrosdelweb.com, 2021), se tiene:

- Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación, te presentamos una guía que te explicará el concepto y características de las bases de datos.

- Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.4.2.3. Características de Base de Datos

Según (maestrosdelweb.com, 2021). Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.4.2.4. Tipos de Base de Datos

Según (maestrosdelweb.com, 2021). Entre los diferentes tipos de base de datos, podemos encontrar los siguientes:

- MySQL: es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

- PostgreSQL y Oracle: Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- Access: Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo mdb con la estructura ya explicada.
- Microsoft SQL Server: es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

2.4.2.5. Gestor de Base de Datos Mysql

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.¹ MySQL AB - desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 - desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

2.4.2.6. Características de Datos Mysql

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad. (ictea.com, 2021).

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar: (ictea.com, 2021).

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos

juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada. (ictea.com, 2021).

2.5. Métricas de Calidad

La medición es un elemento clave en cualquier proceso de ingeniería. Las medidas se emplean para comprender mejor los atributos de los modelos que se crean y evaluar la calidad de los productos de la ingeniería o de los sistemas que se construyen. Pero a diferencia de otras disciplinas de la ingeniería, la del software no se basa en las leyes cuantitativas básicas de la física.

Hasta los desarrolladores de software exhaustivos están de acuerdo en que es importante crear software de alta calidad. Pero, ¿Cómo se define la calidad? En el sentido más amplio, calidad del software es el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos de los estándares de desarrollo explícitamente documentados y de las características implícitas que se esperan de todo software desarrollado profesionalmente (Pressman,2005).

La calidad del software se define también, en términos de ausencia de errores en el funcionamiento del sistema. El ajuste a las necesidades del usuario, el sistema debe ser flexible u susceptible a modificaciones que se puedan realizar de manera rápida y oportuna. El sistema debe alcanzar un desempeño apropiado en términos de tiempo, volumen y espacio. Un sistema debe cumplir de la mejor forma los estándares internacionales establecidos, en lo que a la calidad de software se refiere a:

2.5.1. Norma ISO/IEC 25010.

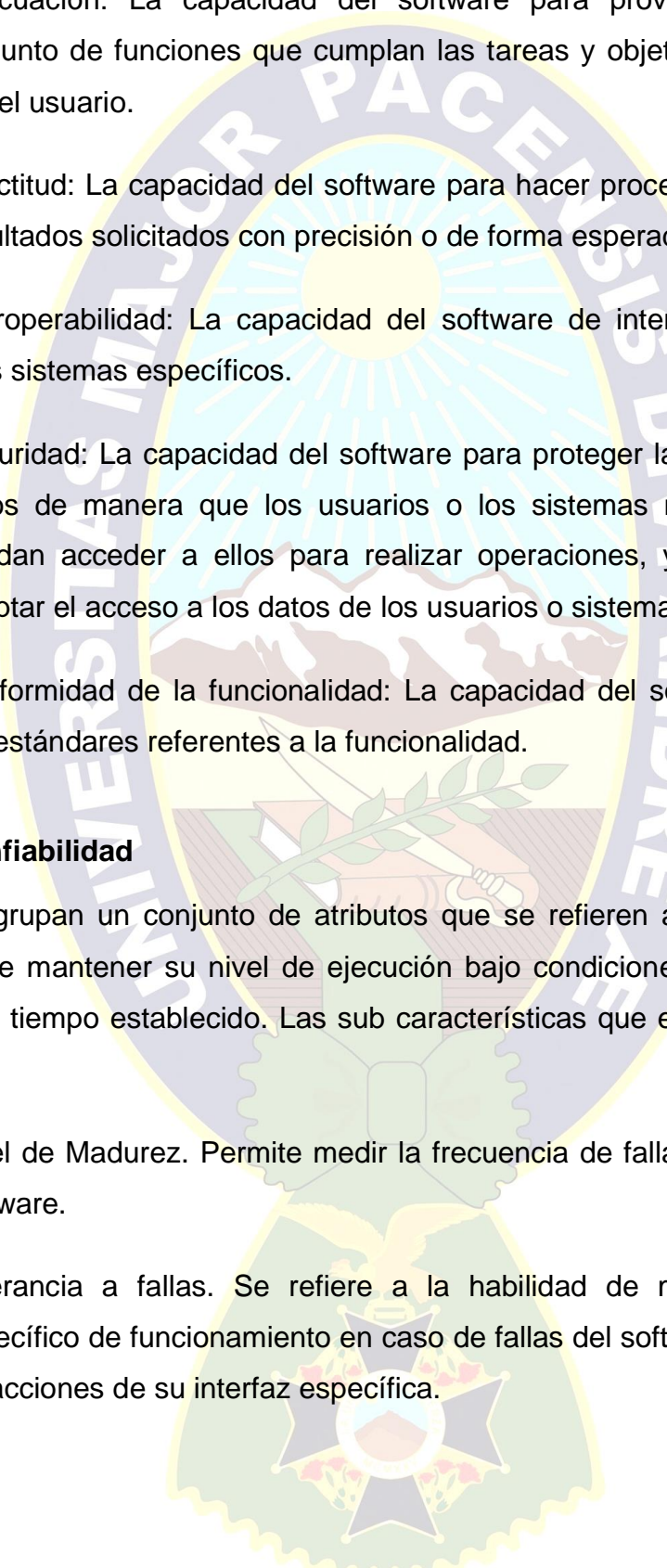
El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y sub características. (ISO25000, 2021)

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las características de calidad que se muestran a continuación:

2.5.2. Adecuación Funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. A continuación, se muestra la característica de Funcionalidad:

- 
- Adecuación: La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
 - Exactitud: La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
 - Interoperabilidad: La capacidad del software de interactuar con uno o más sistemas específicos.
 - Seguridad: La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados
 - Conformidad de la funcionalidad: La capacidad del software de cumplir los estándares referentes a la funcionalidad.

2.5.3. Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub características que el estándar sugiere son:

- Nivel de Madurez. Permite medir la frecuencia de falla por errores en el software.
- Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.

- Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

Los primeros trabajos sobre confiabilidad del software trataban de extrapolar la teoría matemática de la confiabilidad del hardware a la predicción de la confiabilidad del software. La mayor parte de modelos relacionados con el hardware se abocan a la falla debida al uso, en lugar de a la que tiene su origen en los defectos de diseño. En el hardware, las fallas debidas al uso físico (por ejemplo, los efectos de temperatura, corrosión y golpes) son más probables que las debidas al diseño. Desafortunadamente, con el software ocurre lo contrario. En realidad, todas las fallas del software pueden rastrearse en problemas de diseño o de implementación.

2.5.4. Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 25010 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

- Entendimiento: La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- Aprendizaje: La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

- Operabilidad: La manera como el software permite al usuario operarlo y controlarlo.
- Atracción: La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- Conformidad de uso: La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

2.5.5. Mantenibilidad

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

- Capacidad de ser analizado: La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- Cambiabilidad: La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- Estabilidad: La forma como el software evita efectos inesperados para modificaciones del mismo.
- Facilidad de prueba: La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- Conformidad de facilidad de mantenimiento: La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

2.5.6. Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro, se divide en cinco criterios:

- **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados ya sea en hardware o software sistemas operativos sin que implique reacciones negativas ante el cambio
- **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Coexistencia:** La capacidad que tiene el software para coexistir con otros, compartiendo recursos comunes.
- **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- **Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

2.6. Costos y/o Beneficios

Existen tres principales parámetros que se deben usar al calcular los costos de un proyecto de desarrollo de software:

- **Costos de esfuerzo** (los costos de pagar a los ingenieros y administradores de software).
- **Costos de hardware y software**, incluido el mantenimiento.
- **Costos de viajes y capacitación.**

Para la mayoría de los proyectos, el mayor costo es el primer rubro. Debe estimarse el esfuerzo total (en meses-hombre) que es probable se requiera para completar el trabajo de un proyecto. Desde luego, se cuenta con datos limitados para realizar tal valoración, de manera que habrá que hacer la mejor evaluación posible y a continuación agregar contingencia significativa (tiempo y esfuerzo adicionales) en caso de que la estimación inicial sea optimista.

2.6.1. COCOMO

Constructive Cost Model del (COCOMO) es un algorítmico Modelo de la valoración del coste del software convertido cerca Barry Boehm. El modelo utiliza una básica regresión fórmula, con los parámetros que se derivan de datos históricos del proyecto y de características actuales del proyecto.

COCOMO consiste en una jerarquía de tres cada vez más detallados y de formas exactas. El primer nivel, COCOMO básico es buena para aprisa, la orden temprana, áspera de las estimaciones de la magnitud de los costes del software, pero su exactitud debe limitado a su carencia de factores explicar diferencia en cualidades del proyecto (Conductores del coste). COCOMO intermedio toma estos conductores del coste en consideración y COCOMO detallado explica además la influencia de las fases del proyecto individual.

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

a) Modelos

- COCOMO básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LDC.

- COCOMO intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- COCOMO detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

2.6.2. Modelo COCOMO II

El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo, llamado COCOMO II. Como su predecesor, COCOMO II en realidad es una jerarquía de modelos de estimación que aborda las áreas siguientes:

- Modelo de composición de aplicación. Se usa durante las primeras etapas de la ingeniería de software, cuando son primordiales la elaboración de prototipos de las interfaces de usuario, la consideración de la interacción del software y el sistema, la valoración del rendimiento y la evaluación de la madurez de la tecnología.
- Modelo de etapa temprana de diseño. Se usa una vez estabilizados los requisitos y establecida la arquitectura básica del software.
- Modelo de etapa post arquitectónica. Se usa durante la construcción del software.

Como todos los modelos de estimación para software, los modelos COCOMO II requieren información sobre dimensionamiento. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objeto, puntos de función y líneas de código fuente.

Fórmula para hallar el factor de complejidad TCF

$$TCF=(0.65+0.01 \times PF)$$

El procesamiento de datos del punto función se basa en la formula siguiente:

$$PF=Cuenta\ Total \times TCF$$

Factor LDC/PF se calcula con la fórmula:

$$LDC=PF \times FactorLDC/PF$$

Número estimado de líneas de código distribuidas en miles se calcula con la siguiente fórmula:

$$KLCD=LDC/1000$$

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E=ab(KLDC)^b$$

$$D=cbD^d$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses cronológicos.

KLDC: Número estimado de líneas de código distribuidas (en miles).

El número de programadores (Nº Prog) se obtiene con la siguiente fórmula:

$$N^{\circ}\ Prog=E/D$$

Costo del software desarrollado por persona = Numero de programadores *
salario de un programador

2.7. Seguridad

La seguridad es un atributo del sistema que refleja la habilidad de este para protegerse a sí mismo de ataques externos, que podrían ser accidentales o deliberados. Estos ataques externos son posibles puesto que la mayoría de las computadoras de propósito general ahora están en red y, en consecuencia, son accesibles a personas externas (Sommerville, 2011).

La seguridad del software es una actividad de aseguramiento de la calidad del software que se enfoca en la identificación y evaluación de los peligros potenciales que pueden afectar negativamente al software.

2.7.1. Seguridad Física

Según (Bravo, 2015). La seguridad física de un sistema informático consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial". Se refiere a los controles y mecanismos de seguridad dentro y alrededor del Centro de Cómputo así como los medios de acceso remoto al y desde el mismo; implementados para proteger el hardware y medios de almacenamiento de datos.

Este tipo de seguridad está enfocado a cubrir las amenazas ocasionadas tanto por el hombre como por la naturaleza del medio físico en que se encuentra ubicado el sistema. Las principales amenazas que se prevén son:

- Desastres naturales, incendios accidentales y cualquier variación producida por las condiciones ambientales.
- Amenazas ocasionadas por el hombre como robos o sabotajes.
- Disturbios internos y externos deliberados.

Evaluar y controlar permanentemente la seguridad física del sistema es la base para comenzar a integrar la seguridad como función primordial del mismo.

Tener controlado el ambiente y acceso físico permite disminuir siniestros y tener los medios para luchar contra accidentes.

2.7.2. Seguridad Lógica

La seguridad lógica de un sistema informático consiste en la aplicación de barreras y procedimientos que protejan el acceso a los datos y a la información contenida en él.

El activo más importante de un sistema informático es la información y, por tanto, la seguridad lógica se plantea como uno de los objetivos más importantes.

La seguridad lógica trata de conseguir los siguientes objetivos:

- Restringir el acceso a los programas y archivos.
- Asegurar que los usuarios puedan trabajar sin supervisión y no puedan modificar los programas ni los archivos que no correspondan.
- Asegurar que se estén utilizados los datos, archivos y programas correctos en y por el procedimiento correcto.
- Verificar que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y que la información recibida sea la misma que la transmitida.
- Disponer de pasos alternativos de emergencia para la transmisión de información.

a) Seguridad a Nivel Sistema Operativo

Cuando usted es un usuario de la computadora, la seguridad es un gran problema. Los desarrolladores de sistemas operativos de saber que la seguridad del sistema también son importantes. Es por eso que todos los

sistemas operativos han incorporado características de seguridad que hacen que sea segura tanto para navegar por el internet, así como mantener a los usuarios no autorizados utilicen su ordenador.

La seguridad de sistema operativo se basa en dos principios:

- El sistema operativo proporciona acceso a una serie de recursos, directa o indirectamente, como los archivos en un disco local, las llamadas privilegiadas del sistema, la información personal sobre los usuarios, y los servicios ofrecidos por los programas que se ejecutan en el sistema.
- El sistema operativo es capaz de distinguir entre algunos solicitantes de estos recursos que están autorizados o se permite para acceder a los recursos, y otros que no están autorizados o prohibido. Aunque algunos sistemas solo puede distinguir entre privilegiados y no privilegiados, los sistemas suelen tener una forma de identidad solicitante, tales como un nombre de usuario.

Además de permitir que el / no permitir modelo de seguridad, un sistema operativo con un alto nivel de seguridad también se ofrecen opciones de auditoria. Esto permitirá el seguimiento de las solicitudes de acceso a recursos tales como “que ha estado leyendo este archivo”. La seguridad del sistema operativo más se puede dividir en dos subsecciones en lo que respecta a los solicitantes:

- Seguridad Interna – un programa en ejecución. En algunos sistemas, un programa una vez que se ejecuta no tiene limitaciones. Sin embargo, más comúnmente, el programa tiene una identidad que se guarda y se utiliza para comprobar todas sus solicitudes de recursos.
- Seguridad Externa – una nueva solicitud desde fuera de la computadora como un login en una consola conectada o algún tipo de conexión de red. Para establecer la identidad, puede haber un proceso de

autenticación. A menudo, un nombre de usuario debe ser citado y cada usuario puede tener una contraseña.

Otros métodos de autenticación, tales como tarjetas magnéticas o los datos biométricos podrán utilizarse en su lugar. En algunos casos, especialmente con las conexiones de una red, los recursos se puede acceder si autenticación en absoluto.

La seguridad del sistema operativo ha sido durante mucho tiempo una preocupación por los datos altamente sensibles celebra en equipos de carácter personal, comercial, e incluso militares. Es por eso que los programadores del sistema operativo que preste especial atención a la seguridad de los sistemas operativos que están desarrollando. Ellos quieren asegurarse de que los datos delicados contenidos en un sistema se mantienen como privado y solo se le permite ser visto por aquellos que están autorizados a hacerlo.

b) Seguridad a Nivel de Base de Datos

Es la capacidad del sistema para proteger Datos, Servicios y Recursos de usuarios no autorizados. El fin de la seguridad es garantizar la protección o estar libre de todo peligro y/o daño, y que en cierta manera es infalible.

- **Confidencialidad:** nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades.
- **Integridad:** Significa que los objetos solo pueden ser modificados por elementos autorizados, y de una manera controlada.
- **Disponibilidad:** indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados; es el contrato de la negación de servicio.

¿De qué nos queremos proteger?

A continuación, se presenta una relación de los elementos que potencialmente pueden amenazar a nuestro sistema.

Personas

- Pasivos: aquellos que husmean por el sistema, pero no lo modifican y/o destruyen.
- Activos: aquellos que dañan el objetivo atacado o lo modifican en su favor.

Describiremos algunos ataques que realizan las personas:

- Personal
- Ex empleados
- Curiosos
- Hackers
- Terroristas

Amenazas Lógicas

- Software incorrecto
- Herramientas de seguridad
- Puertas traseras
- Canales cubiertos
- Virus
- Gusanos

- Caballos de Troya

Encriptación

La mayor parte de bases de datos contiene la información sensible, propia, y/o probada.

Esto puede incluir información de cliente, salarios de empleado, registros pacientes, números de la tarjeta de crédito. La llave al mantenimiento de esta información en una manera segura es la confidencialidad – y las empresas que no pueden asegurar la seguridad (el valor) para la vergüenza del riesgo confidencial de la información, penas financieras, y a veces aun el negocio mismo.

c) Seguridad a Nivel del Software

La seguridad del software aplica los principios de la seguridad de la información al desarrollo de software. Esto se refiere a la seguridad de información contra el acceso desautorizado y la modificación de información, se está en una fase de procesamiento, almacenamiento o tránsito.

MARCO APLICATIVO

3.1. Introducción

En este capítulo se efectúa el diseño y desarrollo del sistema correspondiente, dado que SCRUM es una metodología simple y concisa pero que exige un gran esfuerzo y trabajo duro si se quiere lograr un producto de calidad, tenemos que entender que no se sigue un plan, sino que es una adaptación constante a las necesidades y circunstancias de la evolución del crecimiento del proyecto.

3.2. Pre-Game

3.2.1. Recopilación de Requerimientos

El presente proyecto como es una solución para la DIGEDEBIS, los requerimientos y objetivos fueron proporcionados por los funcionarios de dicha institución, se elaboró una lista priorizada de requerimientos en base a toda la información, documentación y normativas recolectadas y definidas como en Product Backlog para el sistema.

Las Reuniones. - Esta reunión nos genera el “sprint backlog” o la lista de las tareas que se tiene que realizar y desarrollar, además en ella se determinaran los requisitos del sistema para su funcionamiento total.

Requerimientos del Sistema. - Es parte de la visión del resultado que parte del jefe de proyecto que desea obtener, y estos requerimientos evolucionan durante el desarrollo del sistema. El inventario o la lista de características del sistema que desea obtener el propietario serán ordenados por orden de prioridad.

3.2.2. Pila de Producto (PRODUCT BACKLOG)

En la siguiente lista se encuentran los requisitos generales detallados por el propietario del producto. (Ver Tabla 3.1).

Tabla 3.1. Product Backlog

ID	DESCRIPCIÓN	PRIORIDAD	ESTADO	RESPONSABLE
R1	Planificación para el desarrollo del Sistema.	Alta	100%	Fernando Laura Torrez
R2	Desarrollo del Sistema con modelos UML.	Alta	100%	Fernando Laura Torrez
R3	Creación de la Base de datos, para el almacenamiento de datos del sistema.	Alta	100%	Fernando Laura Torrez
R4	Desarrollo de página web para la la Dirección General de Desarrollo y Bienestar Social.	Alta	100%	Fernando Laura Torrez
R5	Desarrollo de login, para acceso al sistema.	Alta	100%	Fernando Laura Torrez
R6	Desarrollo del formulario de registro de locatario.	Alta	100%	Fernando Laura Torrez
R7	Desarrollar el formulario de registro y modificación de viviendas	Alta	100%	Fernando Laura Torrez
R8	Desarrollar el formulario de visualización de viviendas ocupadas y disponibles.	Alta	100%	Fernando Laura Torrez
R9	Desarrollo de la búsqueda de locatario	Alta	100%	Fernando Laura Torrez
R10	Desarrollo del módulo, visualización del historial de viviendas alquiladas al locatario	Alta	100%	Fernando Laura Torrez
R11	Desarrollo del módulo de asignación de viviendas	Alta	100%	Fernando Laura Torrez
R12	Desarrollo del formulario de extracto de pago por alquiler de vivienda	Alta	100%	Fernando Laura Torrez
R13	Elaborar el reporte de descuentos mensuales que se realizará a los locatarios, por pago del alquiler, la cual será remitida al Ministerio de Defensa.	Alta	100%	Fernando Laura Torrez
R14	Elaboración del módulo de registro de pago de alquiler, por depósito bancario y descuento por papeleta de pago.	Alta	100%	Fernando Laura Torrez
R15	Desarrollo del módulo control de moras, proceso que mostrara una lista con los locatarios, que no cumplieron su pago.	Alta	100%	Fernando Laura Torrez
R16	Desarrollo del reporte mensual de ingreso monetario por alquiler.	Alta	100%	Fernando Laura Torrez
R17	Calculo de las deudas de pago de alquiler pendiente.	Alta	100%	Fernando Laura Torrez
R18	Resumen estimación de ingresos.	Alta	100%	Fernando Laura Torrez
R19	Elaborar reportes de viviendas que concluyen su contrato.	Alta	100%	Fernando Laura Torrez

Fuente: (Elaboración Propia)

3.2.3. Definición del cronograma de trabajo

El cronograma de trabajo fue establecido en base al tiempo de vida de la metodología SCRUM, en el cual se identifica 3 etapas principales. Pre - Game, Game y Postgame, según se detalla en la Tabla 3.2.

Tabla 3.2. Cronograma de trabajo para el desarrollo del producto software

Actividades	Fecha Inicio	Fecha Final	2021																			
			Febrero				Marzo				Abril				Mayo				Junio			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Recopilación de Requerimientos	15/2/2021	26/2/2021																				
Primera Iteración	1/3/2021	18/3/2021																				
Segunda Iteración	22/3/2021	23/4/2021																				
Tercera Iteración	03/5/2021	8/6/2021																				
Control de calidad y corrección de fallas / errores	01/4/2021	24/6/2021																				
Implementación del sistema	21/6/2021	30/6/2021																				

Fuente (Elaboración Propia)

3.2.4. Análisis de riesgo

Un riesgo es la probabilidad de que ocurra algo adverso o en contra de la conclusión de sistema, existen tres tipos de riesgo.

- Riesgo del proyecto, que afecta al calendario o al rendimiento del sistema que se está desarrollando.
- Riesgo del producto, afecta a la calidad o al rendimiento del sistema que se está desarrollando.
- Riesgo de negocio, afectan a la organización que desarrolla o suministra el sistema.

Los riesgos que se encuentran para el desarrollo del sistema además de sus efectos se describen a continuación (Ver Tabla 3.3).

Tabla 3.3. Tabla de Riesgos y sus efectos

RIESGO	DESCRIPCIÓN	PROBABILIDAD	EFEECTO	ESTRATEGIA
Incumplimiento de fecha del cronograma	Es probable que el cronograma del diagrama de Gantt no se cumplan las fechas	ALTA	TOLERABLE	Realizar un segundo cronograma más flexible.
Cambios en los requerimientos del sistema	Puede haber nuevos requerimientos y/o cambio total de algunos módulos ya desarrollados	MODERADA	TOLERABLE	Programar reuniones con funcionarios de DIGEDEBIS.
No exista la infraestructura, para el sistema	Es posible que no se cuente o exista una demora, en los materiales y/o infraestructura para el sistema.	ALTA	IMPORTANTE	Utilizar materiales propios para no retrasar, el desarrollo del sistema.

Fuente: (Elaboración Propia)

3.3. Game

3.3.1. Sprint Backlog

El desarrollo del sistema, en conformidad a la metodología que estamos usando para el desarrollo del sistema, es realizado en 7 Sprint.

Cada Sprint se realiza con el previo análisis, planificación y respectiva codificación con la etapa de pruebas.

La estrategia para desarrollar el sistema es realizar modelos UML, la codificación se realiza con el lenguaje de programación Java, utilizando el framework Primeface, con una base de datos MySql. A continuación, se describen cada uno de los sprints realizados con el detalle técnico de la implementación.

3.3.2. Primera iteración (SPRINT BACKLOG 1)

El Sprint 1 fue desarrollado bajo el siguiente detalle técnico, las actividades que fueron realizadas durante el tiempo del Sprint, están detalladas en la tabla 3.4.

Tabla 3.4. Sprint Backlog 1

ID	TAREA	TIPO	ESTADO	RESPONSABLE	DÍAS DE TRABAJO
1	Planificación del desarrollo del sistema y la página web	Análisis	Terminado	Fernando Laura Torrez	1
2	Análisis de requerimientos del Product Backlog	Análisis	Terminado	Fernando Laura Torrez	2
3	Descripción de Roles y Responsabilidades	Análisis	Terminado	Fernando Laura Torrez	1
4	Diseño de la arquitectura del sistema	Análisis	Terminado	Fernando Laura Torrez	1
5	Diseño de los diagramas de los casos de uso	Análisis	Terminado	Fernando Laura Torrez	2
6	Diseño de los diagramas de los casos de secuencia	Análisis	Terminado	Fernando Laura Torrez	2
7	Diseño de los diagramas de Entidad – Relación, Clases y Físico de la base de datos.	Análisis	Terminado	Fernando Laura Torrez	2
8	Diseñar interfaces de los usuarios.	Análisis	Terminado	Fernando Laura Torrez	3
SPRINT		INICIO		FIN	
1		01/03/2021		14 días	

Fuente: (Elaboración Propia)

3.3.2.1. Planificación del desarrollo del sistema (BACKLOG 1)

Variables de campo

- **Tiempo.**- El sistema tiene planeado empezar el 01 de marzo del presente año.

- **Calidad.**- El desarrollo del sistema está bajo criterios de mejores prácticas para el desarrollo de aplicaciones web.
- **Retro alimentación.**- Las observaciones y/o errores, que se encuentren en el desarrollo del sistema, ayudaran un mejoramiento constante del sistema.

Control de evolución del proyecto

- **Revisión.**- Cada vez que se acabe un módulo y/o módulos se realizara la presentación pertinente a los funcionarios de Dirección General de Desarrollo y Bienestar Social.
- **Observación.**- Al final de cada iteración se pueda inspeccionar y evaluar cada parte del proyecto. Donde se observará la mejor y peor parte del producto; si supera las pruebas, se entrega el producto, caso contrario se realiza una iteración para corregir errores.

3.3.2.2. Análisis de requerimientos del product Backlog (BACKLOG 2)

Se realiza una reunión, para poder analizar cada uno de los requerimientos del product backlog, logrando conseguir más claridad en cada uno de los objetivos, de tal manera minimizar los contratiempos que podrían darse en el transcurso del desarrollo del sistema.

3.3.2.3. Descripción de roles y responsabilidades (BACKLOG 3)

A continuación, se describen a las personas que intervienen en el proyecto, los comprometidos y los implicados.

Descripción de Stackholders

Para desarrollar el sistema es necesario definir cada uno de los actores que intervendrán en el proyecto, estos son los que representan al cliente y a los

usuarios, esto nos permitirá identificar los requerimientos y aceptar las soluciones propuestas ante el problema.

➤ **StakeHolder Cliente**

Tabla 3.5. StackHolder Usuario 1

Representante	Cnl, DAEN. Henry Claros Vela
Descripción	Coordinador del proyecto “Sistema de Información pasar la asignación y control de vivienda”.
Tipo	Cliente.
Responsabilidades	Representa al cliente. Encargado de brindar los requerimientos generales del proyecto. Responsable de obtener el mayor valor del producto.
Grado de Participación	Requerimientos de usuario
Comentario	Ninguno

Fuente: (Elaboración Propia)

Tabla 3.6. StackHolder Usuario 2

Representante	Sof, My. DEMA Héctor Tintaya Altamirano
Descripción	Coordinador administrativo del proyecto “Sistema de Información pasar la asignación y control de vivienda”.
Tipo	Cliente, Usuario
Responsabilidades	Representa al usuario. Encargado de brindar la información necesaria de los requisitos del sistema. Realizara observaciones de acuerdo a mejoraras del sistema.
Grado de Participación	A detalle del cliente.
Comentario	Ninguno

Fuente: (Elaboración Propia)

➤ **StakeHolder Desarrollador**

A continuación, se muestra la tabla que describe al equipo de desarrollo y las responsabilidades que tienen que cumplir. Ver tabla 3.7

Tabla 3.7. StackHolder Desarrollador

Representante	Fernando Laura Torrez
Descripción	Encargado de desarrollo del Sistema de Información pasar la asignación y control de vivienda.
Tipo	Analista/Desarrollador del sistema.
Responsabilidades	Responsable de recabar los requerimientos del sistema. Seleccionar las herramientas de trabajo. Realizar preguntas y solicita aclaraciones necesarias. Propone modificaciones y soluciones alternativas.
Grado de Participación	Planificación del sistema de información y control
Comentario	Ninguno

Fuente: (Elaboración Propia)

➤ **Usuarios**

A continuación, se muestra las tablas que describen usuarios que requieren y brindan la información del sistema. Ver tablas 3.8, 3.9 y 3.10

Tabla 3.8. Usuario – Responsable de viviendas

Representante	Usuario
Descripción	Encarga de gestionar, controlar la asignación de las viviendas (registro, evaluación y aprobación) y generar los descuentos mensuales por alquiler. Realiza el control de pago por alquiler, hasta su cancelación y realizar consulta de los mismos.
Tipo	Usuarios – Responsable de viviendas.
Responsabilidades	Encargado de clasificar y analizar la viabilidad de las solicitudes de viviendas, verificar el cumplimiento de los requisitos y realizar el cobro de los alquileres pendientes, según normas establecidas por la Dirección General.
Grado de Participación	A definir por el cliente.
Comentario	Ninguno

Fuente: (Elaboración Propia)

Tabla 3.9. Usuario – Auxiliar vivienda

Representante	Usuario
Descripción	Tiene el rol de registrar los descuentos por alquiler, reporte que es emitido por el Ministerio de Defensa. Llevar el control de pagos por alquiler hasta su cancelación y generar reportes de moras, deudas pendientes. Proporcionar extracto de pago de alquiler por locatario.
Tipo	Usuarios – Auxiliar vivienda
Responsabilidades	Registrar correctamente los descuentos por alquiler del Ministerio de Defensa y controlar los pagos hasta su cancelación.
Grado de Participación	A definir por el cliente.
Comentario	Ninguno

Fuente: (Elaboración Propia)

Tabla 3.10. Usuario – Administrador

Representante	Usuario
Descripción	Es encargado de registrar, modificar y dar de baja a los usuarios del sistema y asignar contraseña. Encargado de la gestión del registro de afiliado.
Tipo	Usuarios – Administrador
Responsabilidades	Registrar correctamente los datos del usuario y asignarle una contraseña. Registrar los datos correctos del locatario.
Grado de Participación	A definir por el cliente.
Comentario	Ninguno

Fuente: (Elaboración Propia)

3.3.2.4. Historia de Usuario. (ID BACKLOG 3)

El objetivo de este punto es de realizar un listado de las historias de usuario, que se detallaron en la reunión con el cliente.

A continuación, en las tablas 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19 y 3.20 se describen las historias de usuario del sistema Web con sus respectivas descripciones.

Tabla 3.11. Registro de Locatario

Historia de Usuario	
Historia de Usuario Nro.:1	
Usuario: Responsable de viviendas/Administrador Prioridad: Alta	Nombre: Registro de Locatario
Descripción	Podrá registrar la información necesaria, para poder identificar a los locatarios, según los requerimientos necesarios. Realizar actualizaciones de datos.
Observación	Si el locatario es nuevo, se crea un registro, caso contrario no lo dejara crearlo.

Fuente: (Elaboración Propia)

Tabla 3.12. Registro de vivienda

Historia de Usuario	
Historia de Usuario Nro.:2	
Usuario: Responsable de viviendas/Administrador Prioridad: Alta	Nombre: Registro de Vivienda
Descripción	Podrá registrar la información necesaria de la nueva vivienda, para prestar servicios de alquiler. Realizar modificaciones de los datos registrados.
Observación	Si la vivienda es nueva, se creará un nuevo registro. No se realizará el registro de la baja de una vivienda habitada.

Fuente: (Elaboración Propia)

Tabla 3.13. Gestión de Reportes – Estado de vivienda

Historia de Usuario	
Historia de Usuario Nro.:3	
Usuario: Responsable de viviendas/ Auxiliar vivienda Prioridad: Alta	Nombre: Gestión de Reportes – Estado de vivienda
Descripción	Permitirá generar y visualizar el reposte general de las viviendas, su estado actual y la fecha de conclusión de alquiler.
Observación	Previamente el usuario debe estar registrado en el sistema.

Fuente: (Elaboración Propia)

Tabla 3.14. Búsqueda de locatario

Historia de Usuario	
Historia de Usuario Nro.:4	
Usuario: Responsable de viviendas/ Auxiliar vivienda Prioridad: Alta	Nombre: Búsqueda de locatario.
Descripción	Logrará realizar la búsqueda del locatario de acuerdo a las posibilidades de rastreo (CI, Carnet Militar, Paterno, Materno y Nombre).
Observación	Previamente el usuario debe estar registrado en el sistema.

Fuente: (Elaboración Propia)

Tabla 3.15. Historial de alquiler

Historia de Usuario	
Historia de Usuario Nro.:5	
Usuario: Responsable de viviendas/ Auxiliar vivienda Prioridad: Alta	Nombre: Historial de alquiler.
Descripción	Permite ubicar y visualizar el historial de las viviendas alquiladas de un determinado locatario.
Observación	Previamente el usuario debe estar registrado en el sistema.

Fuente: (Elaboración Propia)

Tabla 3.16. Asignación de viviendas

Historia de Usuario	
Historia de Usuario Nro.:6	
Usuario: Responsable de viviendas Prioridad: Alta	Nombre: Asignación de viviendas
Descripción	Permitirá registrar las solicitudes de viviendas y el dato del interesado. Permitirá visualizar las viviendas disponibles y poder asignar. Permitirá generar el plan de pagos luego de asignarle el monto del alquiler. Registrará la aprobación de la solicitud.
Observación	Previamente la solicitud debe estar registrada.

Fuente: (Elaboración Propia)

Tabla 3.17. Extractos de pagos de alquiler

Historia de Usuario	
Historia de Usuario Nro.:7	
Usuario: Responsable de viviendas/ Auxiliar vivienda Prioridad: Alta	Nombre: Extractos de pagos de alquiler.
Descripción	Podrá realizar la búsqueda del socio y visualizar la información de la vivienda alquilada sus pagos realizados.
Observación	El locatario deberá contar con la asignación de vivienda.

Fuente: (Elaboración Propia)

Tabla 3.18. Generar Descuentos Mensuales

Historia de Usuario	
Historia de Usuario Nro.:8	
Usuario: Auxiliar/ Responsable de viviendas Prioridad: Alta	Nombre: Generar Descuentos Mensuales
Descripción	Podrá generar mensualmente el reporte de los descuentos que se realizaran a los locatarios (personal de la FAB), que habitan las viviendas funcionales, y serán enviados al Ministerio de Defensa. Permitirá generar reportes por moras.
Observación	Previamente el Auxiliar de vivienda debe estar registrado e ingresar al sistema.

Fuente: (Elaboración Propia)

Tabla 3.19. Registro de pago

Historia de Usuario	
Historia de Usuario Nro.:9	
Usuario: Auxiliar vivienda Prioridad: Alta	Nombre: Registro de pago
Descripción	Registrará la lista de los descuentos mensuales que se realizaron a los locatarios por alquiler, lista emita por el Ministerio de Defensa. Registrar los pagos del alquiler, mediante depósitos bancarios.
Observación	Previamente el Auxiliar de viviendas debe estar registrado e ingresar al sistema.

Fuente: (Elaboración Propia)

Tabla 3.20. Control de pago

Historia de Usuario	
Historia de Usuario Nro.:10	
Usuario: Auxiliar vivienda. Prioridad: Alta	Nombre: Control de pago
Descripción	Podrá obtener información de las viviendas y el responsable que no cumplió con el pago. Verificara el cumplimiento del plazo y cuota.
Observación	Previamente el Auxiliar de vivienda debe estar registrado.

Fuente: (Elaboración Propia)

Tabla 3.21. Reporte de ingresos por alquiler

Historia de Usuario	
Historia de Usuario Nro.:11	
Usuario: Responsable de viviendas. Prioridad: Alta	Nombre: Reporte de ingresos por alquiler
Descripción	Se podrá contar con el reporte de pagos registrados y calcular la sumatoria total del ingreso mensual.
Observación	Previamente el usuario debe estar registrado e ingresar al sistema.

Fuente: (Elaboración Propia)

Tabla 3.22. Calculo de deudas

Historia de Usuario	
Historia de Usuario Nro.:12	
Usuario: Auxiliar vivienda. Prioridad: Alta	Nombre: Calculo de deudas
Descripción	Le permitirá calcular el acumulado de las cuotas no canceladas, por alquiler.
Observación	Previamente el Auxiliar de vivienda debe estar registrado. Previa autorización del Responsable de viviendas no podrá mandar sus descuentos.

Fuente: (Elaboración Propia)

Tabla 3.23. Resumen estimación de ingresos

Historia de Usuario	
Historia de Usuario Nro.:13	
Usuario: Responsable de viviendas. Prioridad: Alta	Nombre: Resumen estimación de ingresos.
Descripción	Permitirá generar un resumen de ingresos estimados para el próximo mes.
Observación	Previamente el usuario debe estar registrado.

Fuente: (Elaboración Propia)

Tabla 3.24. Administra usuario

Historia de Usuario	
Historia de Usuario Nro.:14	
Usuario: Administrador. Prioridad: Alta	Nombre: Administración de usuario
Descripción	Encargado de gestionar el control de usuario y asignarle una contraseña.
Observación	Previamente el Administrador debe estar registrado en el sistema.

Fuente: (Elaboración Propia)

3.3.2.5. Diseño de la arquitectura del sistema (ID BACKLOG 4)

La arquitectura del sistema muestra todos los módulos y sus respectivas opciones a donde se puede acceder cada usuario.

El Administrador cuenta con 3 módulos, en base a los requerimientos y las historias de usuarios ya predefinidos. (Ver figura 3.1.)



Figura 3.1. Diseño de la arquitectura del sistema – Administrador

Fuente: (Elaboración Propia)

El Responsable de viviendas cuenta con 5 módulos, en base a los requerimientos y las historias de usuarios ya predefinidos. (Ver figura 3.2.).

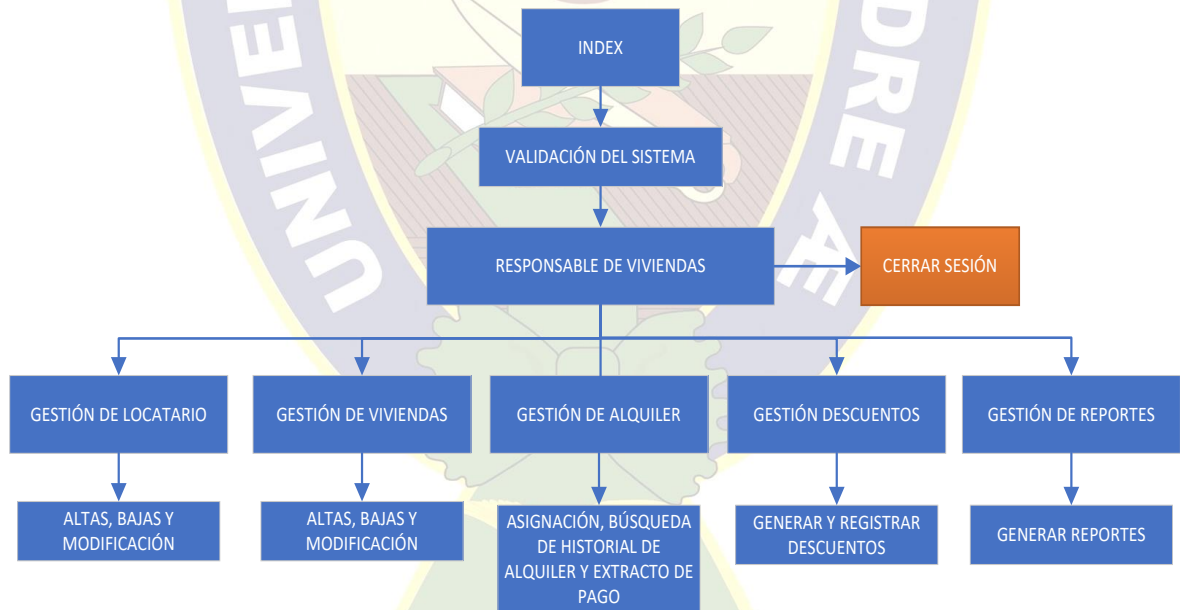


Figura 3.2. Diseño de la arquitectura del sistema – Responsable de viviendas

Fuente: (Elaboración Propia)

3.3.2.6. Casos de uso (ID BACKLOG 5)

Después de efectuarse las reuniones con el cliente y todo el personal involucrado se llegó a identificar los siguientes casos de uso por cada usuario, las siguientes figuras y sus respectivas especificaciones se muestran los pasos y actividades que deben realizarse para llevar a cabo procesos, esto por los usuarios: Encargado de viviendas, Auxiliar de viviendas y Administrador.

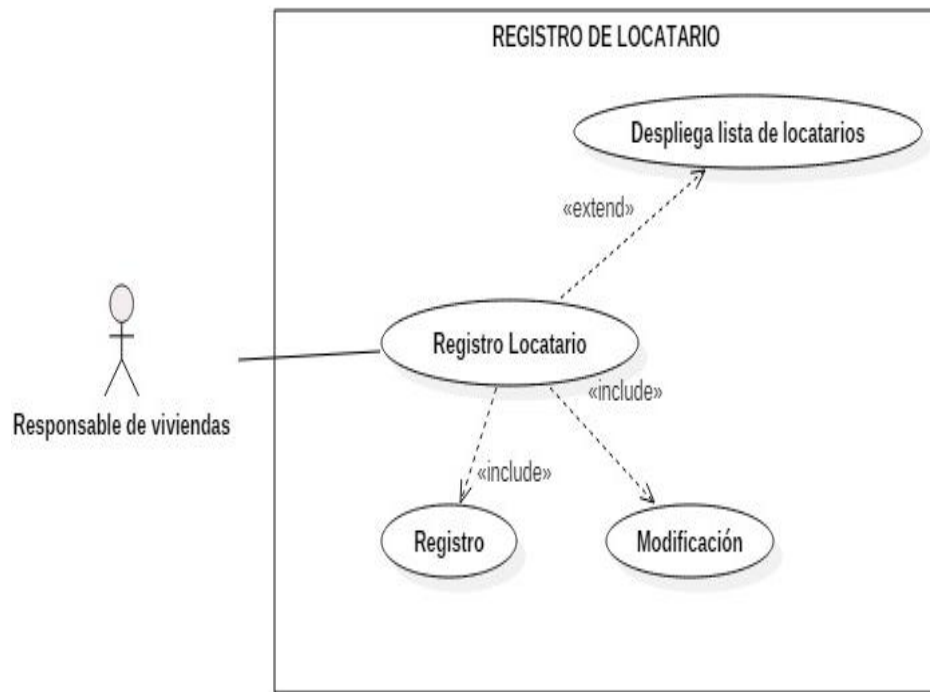


Figura 3.3. Diagrama de caso de uso: Registro de Locatario.

Fuente: (Elaboración Propia)

Tabla 3.25. Especificación de Caso de uso: Registro de Locatario.

Nombre	Registro de Locatario	
Nro. Caso de uso	1	
Actores	Responsable de viviendas	
Propósito	Registrar la información personal de los locatarios.	
Descripción	El usuario ingresara al sistema y registrara los datos personales del afiliado y podrá realizar modificaciones de datos.	
Flujo principal	Eventos del actor	Eventos del sistema
	1. El usuario ingresa al sistema y elige la opción registro de locatario.	2. El sistema despliega la interfaz gráfica para realizar el registro.

	3. Realizara el registro de los datos personales del locatario y destino actual.	
	4. Elige opción guardar.	5. El sistema captura los datos para validarlos.
	7. Elige opción de confirmación.	6. El sistema despliega ventana de confirmación.
		8. El sistema realiza el almacenamiento de dicha información.
Flujo alternativo	3. El formulario es llenado de forma errónea. 5. El sistema despliega una ventana de error.	
Precondición	El locatario no debe estar registrado en el sistema.	
Postcondición	El sistema registrar la información del nuevo locatario.	

Fuente: (Elaboración Propia)

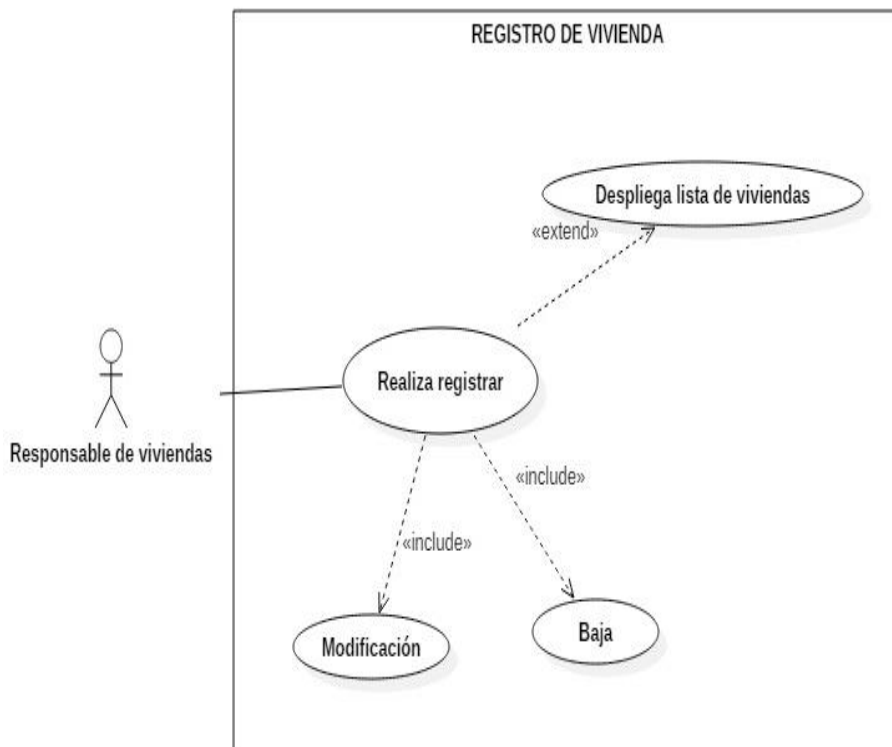


Figura 3.4. Diagrama de caso de uso: Registro de vivienda

Fuente: (Elaboración Propia)

Tabla 3.26. Especificación de Caso de uso: Registro de vivienda.

Nombre	Registro de vivienda	
Nro. Caso de uso	2	
Actores	Responsable de viviendas	
Propósito	Almacenar y modificar la información de las viviendas.	
Descripción	Podrá registrar la información necesaria de la vivienda, para prestar servicios de alquiler. Realizar modificaciones de los datos registrados.	
Flujo principal	Eventos del actor	Eventos del sistema
	1. El usuario ingresa al sistema y elige la opción registro.	2. El sistema despliega la interfaz gráfica para realizar el registro de la nueva vivienda.
	3. Realizara el registro de los datos de la vivienda y elige opción guardar.	4. Capturara los datos y luego lo valida. 5. El sistema despliega ventana de confirmación.
	6. Elige opción de confirmación.	7. El sistema realiza el almacenamiento de dicha información.
Flujo alternativo	3. El formulario es llenado de forma errónea. 5. El sistema despliega una ventana de error.	
Precondición	La vivienda no tiene que estar registrada.	
Postcondición	Se almacenara en el sistema los datos de la vivienda.	

Fuente: (Elaboración Propia)

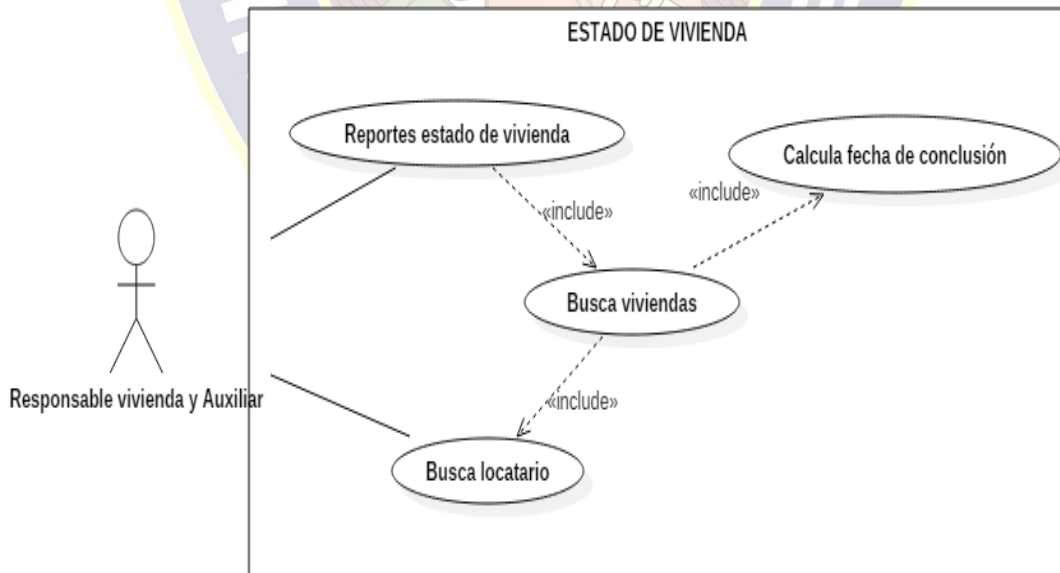


Figura 3.5. Diagrama de caso de uso: Estado de vivienda

Fuente: (Elaboración Propia)

Tabla 3.27. Especificación de Caso de uso: Estado de vivienda

Nombre	Estado de vivienda	
Nro. Caso de uso	3	
Actores	Responsable de vivienda / Auxiliar vivienda	
Propósito	Obtener información actualizada de los estados de las viviendas, responsable y la fecha de conclusión de alquiler.	
Descripción	El Usuario podrá generar el reporte del estado de las viviendas ocupadas, disponibles, responsable y la fecha de conclusión del alquiler.	
Flujo principal	Eventos del actor	Eventos del sistema
	1. El usuario ingresa al sistema y elige la opción.	2. El sistema despliega la interfaz gráfica solicitada.
	3. Elige opción generar.	4. Busca viviendas y su estado. 5. El sistema despliega la información requerida.
	6. Elige opción de imprimir.	7. El sistema realiza la opción requerida.
Precondición	El usuario ha ingreso correctamente el al sistema.	
Postcondición	El sistema debe generar el reporte estado de viviendas.	

Fuente: (Elaboración Propia)

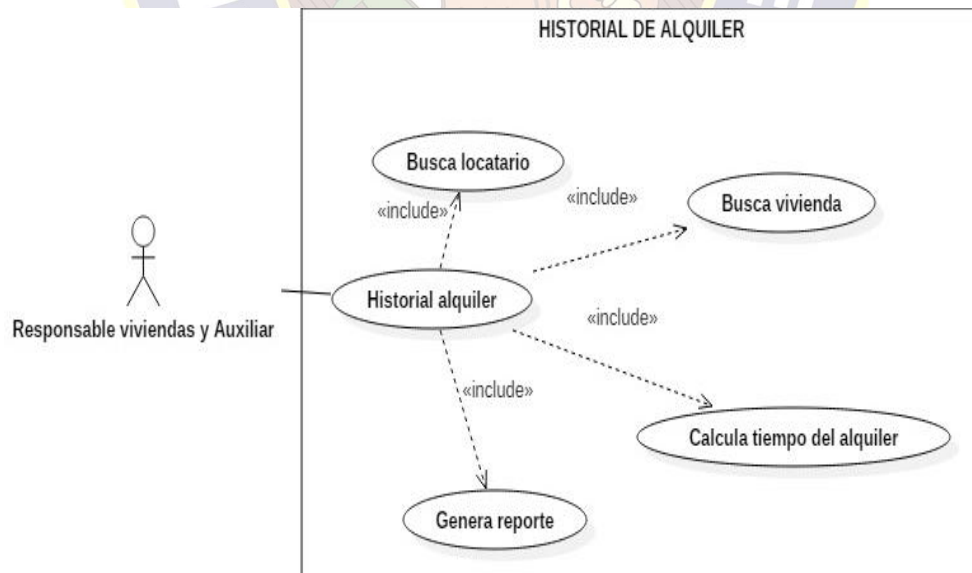


Figura 3.6. Diagrama de caso de uso: Historial de alquiler.

Fuente: (Elaboración Propia)

Tabla 3.28. Especificación de Caso de uso: Historial de alquiler

Nombre		Historial de alquiler	
Nro. Caso de uso	4		
Actores	Responsable de vivienda / Auxiliar vivienda		
Propósito	Generar reporte de las viviendas alquiladas al locatario.		
Descripción	El Usuario podrá buscar al locatario y generar su historial de viviendas alquiladas.		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige la opción.	2. El sistema despliega la interfaz gráfica solicitada.	
	3. Ingresa datos del locatario.	4. Captura y busca los datos del locatario.	
		5. Busca las viviendas alquiladas.	
		6. El sistema despliega la información requerida.	
	7. Elige opción imprimir.	8. Realiza opción requerida.	
Flujo alternativo	4. No encuentra locatario.		
Precondición	El Locatario debe de estar registrado en el sistema.		
Postcondición	El sistema debe generar el reporte de alquileres histórico		

Fuente: (Elaboración Propia)

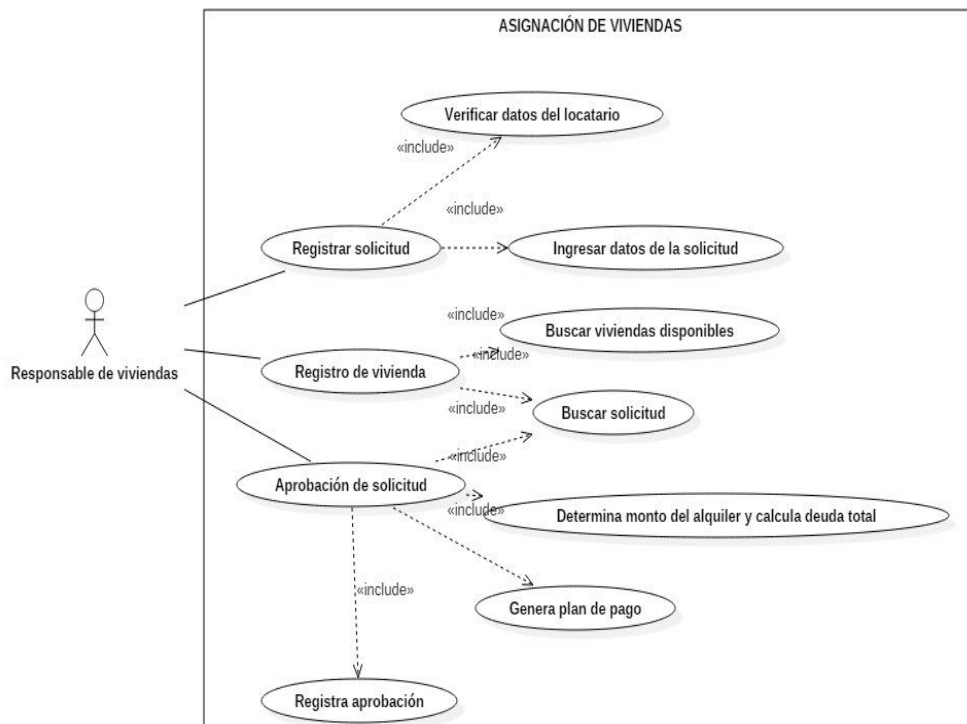


Figura 3.7. Diagrama de caso de uso: Asignación de viviendas

Fuente: (Elaboración Propia)

Tabla 3.29. Especificación de Caso de uso: Asignación de viviendas.

Nombre		Asignación de viviendas.	
Nro. Caso de uso	5		
Actores	Responsable de vivienda		
Propósito	Registrar las solicitudes de viviendas, asignarle vivienda y registrar la aprobación de la solicitud.		
Descripción	<p>El responsable registra los datos de la solicitud y del solicitante.</p> <p>Podrá visualizar las viviendas disponibles para su asignación.</p> <p>Calculara la cuota del alquiler y la deuda total.</p> <p>Generar un plan de pagos y registrará la aprobación de la solicitud.</p>		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige la opción. 3. Ingresa los datos del solicitante.	2. El sistema despliega la opción solicitada. 4. Captura y busca los datos.	
		5. Muestra los datos y despliega formulario re registro.	
	6. Ingresa datos.	7. Captura los datos y muestra una ventana de confirmación.	
	8. Elige opción.	9. Almacena los datos.	
Flujo alternativo	5. No encuentra ningún dato.		
Precondición	El solicitante debe estar registrado en el sistema.		

Fuente: (Elaboración Propia)

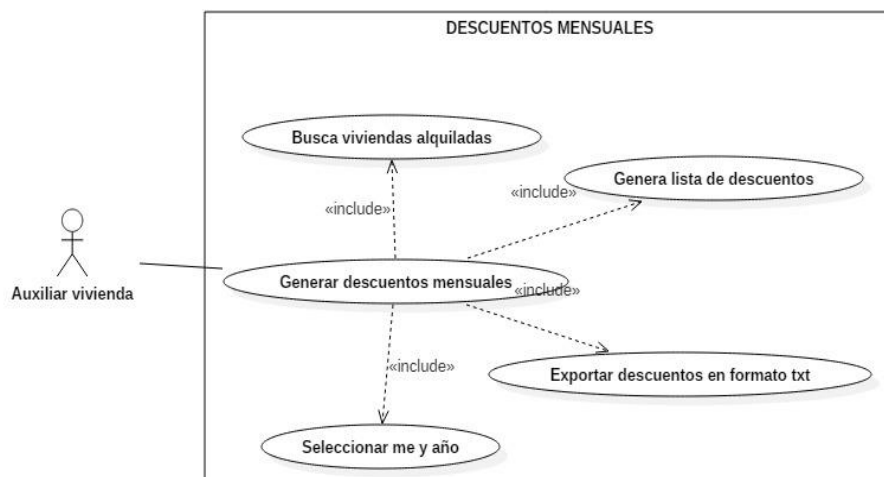


Figura 3.8. Diagrama de caso de uso: Descuentos Mensuales.

Fuente: (Elaboración Propia)

Tabla 3.30. Especificación de Caso de uso: Descuentos Mensuales

Nombre		Descuentos Mensuales	
Nro. Caso de uso	6		
Actores	Auxiliar vivienda		
Propósito	Generar mensualmente el reporte de los descuentos que se realizaran a los locatarios (personal de la F.A.B.) por concepto de alquiler de vivienda y su posterior envío al Ministerio de Defensa.		
Descripción	<p>El responsable generara mensualmente el reporte de los descuentos que se realizarán a los locatarios (personal de la F.A.B.), por alquiler de vivienda.</p> <p>El mencionado reporte también se generará en un archivo plano, según los datos requeridos por el Ministerio de Defensa, quien se encargará de realizar los descuentos.</p>		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige la opción requerida.	2. El sistema despliega la opción solicitada.	
	3. Selecciona la fecha a generar los descuentos.	4. Busca las viviendas alquiladas y el responsable que cuentas con saldo a la fecha seleccionada.	
		5. Despliega un reporte con las viviendas y responsable que tendrán que ser descontados.	
	6. Elige opción guardar y generara listado en formato de archivo plano.	7. Procesa opción seleccionada y muestra opción de conformidad.	
	8. Acepta opción de conformidad.	9. Se almacena los datos en el sistema.	
Flujo alternativo	5.- Muestra un error de fecha.		
Precondición	El Auxiliar ha realizado correctamente al ingreso al sistema.		
Postcondición	El sistema debe generar el reporte de los descuentos.		

Fuente: (Elaboración Propia)

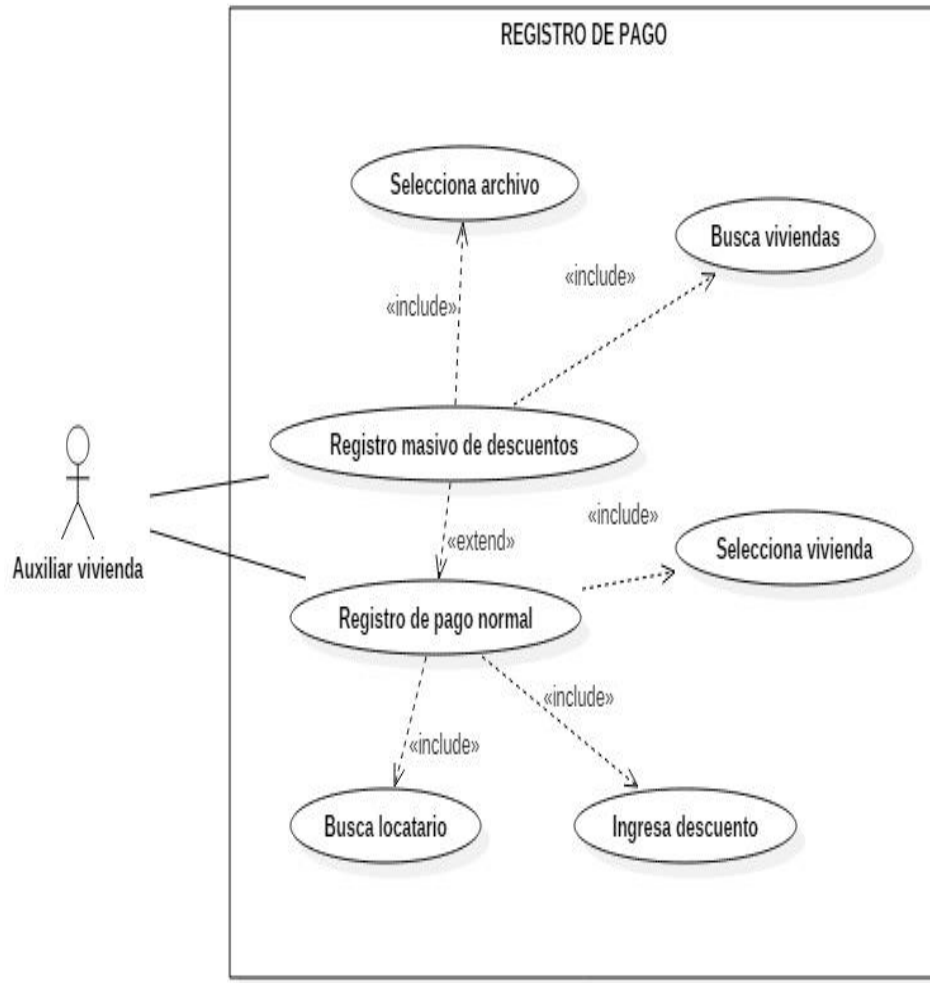


Figura 3.9. Diagrama de caso de uso: Registro de pagos

Fuente: (Elaboración Propia)

Tabla 3.31. Especificación de Caso de uso: Registro de pagos.

Nombre		Registro de pagos	
Nro. Caso de uso	7		
Actores	Auxiliar de vivienda		
Propósito	Registrar los pagos de alquiler por depósitos bancarios y lista de descuentos, emitidos por el Ministerio de Defensa.		
Descripción	Registrará la lista de los descuentos mensuales que se realizaron a los locatarios por alquiler, lista emita por el Ministerio de Defensa. Registrar los pagos del alquiler, mediante depósitos bancarios.		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige la opción requerida.	2. El sistema despliega la opción seleccionada.	
	3. Introduce datos del locatario y selecciona la opción buscar.	4. Captura datos y busca la vivienda asignada.	
		5. Despliega la información.	
	6. Selecciona la vivienda.	7. El sistema muestra los pagos realizados.	
	8. Introduce datos del depósito bancario (monto y número de cuenta) y elige opción procesar.	9. Captura los datos y determina el saldo a la fecha.	
		10. Muestra opción de conformidad.	
	11. Acepta opción de conformidad.	12. Almacena la información en el sistema.	
Flujo alternativo	5. El sistema no encuentra los datos de la vivienda alquilada. 9. El sistema verifica que el depósito no cubre el pago del alquiler.		
Precondición	El locatario debe contar con una vivienda alquilada, para el registro de su pago.		
Postcondición	El sistema registrara el depósito bancario.		

Fuente: (Elaboración Propia)



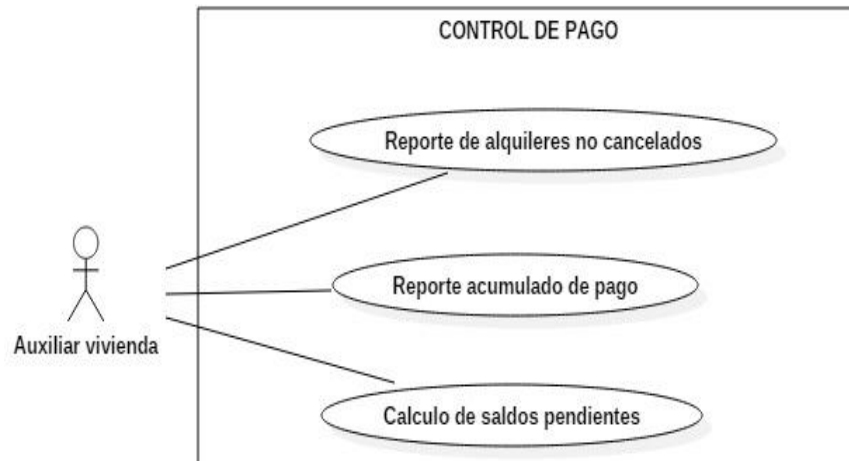


Figura 3.10. Diagrama de caso de uso: Control de pago

Fuente: (Elaboración Propia)

Tabla 3.32. Especificación de Caso de uso: Control de pago

Nombre		Control de pago
Nro. Caso de uso	8	
Actores	Auxiliar vivienda.	
Propósito	Llevar el control de los pagos por alquiler mensual que realizan los locatarios.	
Descripción	El actor realizara el control de pagos mensuales por alquiler hasta su cancelación, contara con reportes de alquiler no cancelados, obtener información de pagos acumulados y generar reporte de saldos pendientes.	
Flujo principal	Eventos del actor	Eventos del sistema
	1. El usuario ingresa al sistema y elige opción requerida.	2. El sistema muestra la opción solicitada.
	3. Elige opción generar reporte de alquileres no cancelados.	4. Captura mes y año para generar reporte.
		5. Muestra reporte solicitado.
	6. Elige opción imprimir.	7. Ejecuta opción solicitada.
Flujo alternativo	4. Ingresa datos erróneos.	
Precondición	El usuario debe estar registrado en el sistema.	
Postcondición	El sistema muestra los datos solicitados.	

Fuente: (Elaboración Propia)

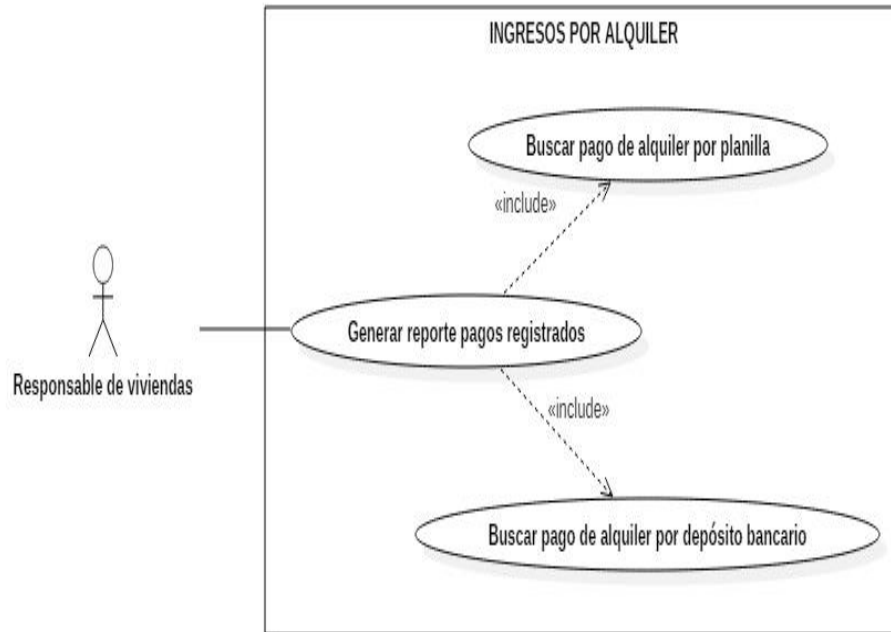


Figura 3.11. Diagrama de caso de uso: Ingresos por alquiler

Fuente: (Elaboración Propia)

Tabla 3.33. Especificación de Caso de uso: Ingresos por alquiler

Nombre		Ingresos por alquiler	
Nro. Caso de uso	9		
Actores	Responsable de viviendas.		
Propósito	Llevar el control mensual de los pagos registrados por alquiler.		
Descripción	El usuario obtendrá el reporte de los locatarios que cumplieron con su pago de alquiler y poder determinar el monto total cancelado.		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige opción requerida.	2. El sistema muestra la opción solicitada.	
	3. Elige opción generar reporte de ingresos.	4. Captura mes y año para generar reporte.	
		5. Muestra reporte solicitado.	
	6. Elige opción imprimir.	7. Ejecuta opción solicitada.	
Flujo alternativo	4. Ingresa datos erróneos.		
Precondición	El usuario debe estar registrado en el sistema.		
Postcondición	El sistema muestra los datos solicitados.		

Fuente: (Elaboración Propia)

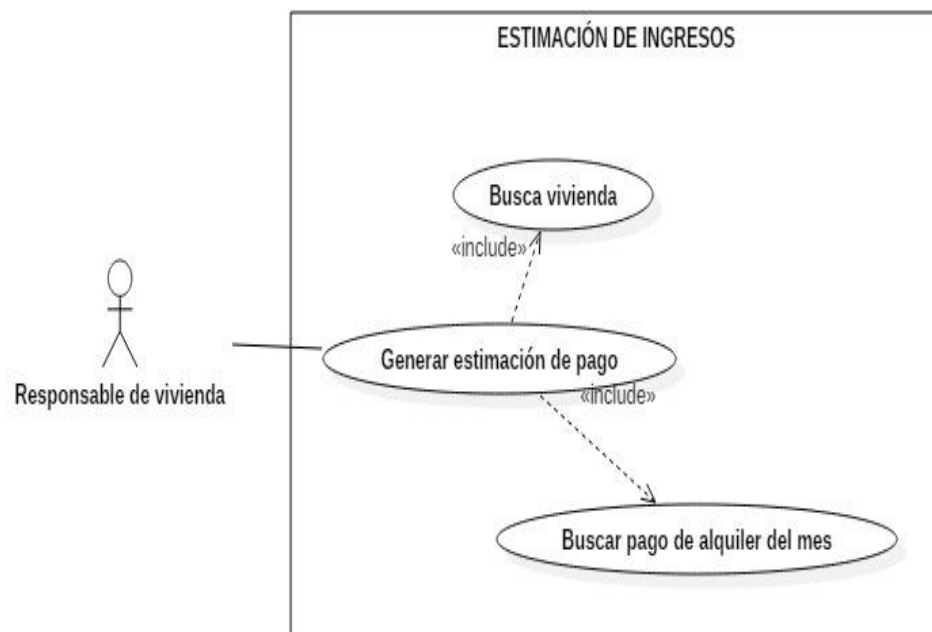


Figura 3.12. Diagrama de caso de uso: Estimación de ingresos

Fuente: (Elaboración Propia)

Tabla 3.34. Especificación de Caso de uso: Estimación de ingresos

Nombre		Estimación de ingresos	
Nro. Caso de uso	10		
Actores	Responsable de viviendas.		
Propósito	Obtener una estimación de los ingresos del próximo mes.		
Descripción	El usuario podrá generar la estimación de los ingresos que deberán percibir la DIGEDEBIS por alquiler del próximo mes.		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El usuario ingresa al sistema y elige opción requerida.	2. El sistema muestra la opción solicitada.	
	3. Elige opción generar estimación de ingresos.	4. Captura mes y año para generar reporte.	
		5. Muestra reporte solicitado.	
	6. Elige opción imprimir.	7. Ejecuta opción solicitada.	
Flujo alternativo	4. Ingresa datos erróneos.		
Precondición	El usuario debe estar registrado en el sistema.		
Postcondición	El sistema muestra los datos solicitados.		

Fuente: (Elaboración Propia)

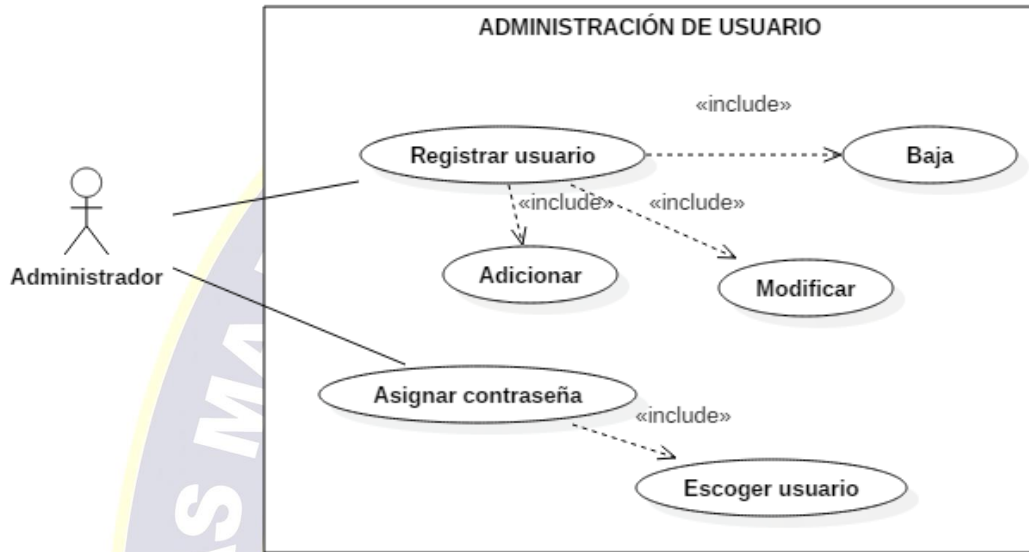


Figura 3.13. Diagrama de caso de uso: Administración de usuario

Fuente: (Elaboración Propia)

Tabla 3.35. Especificación de Caso de uso: Administración de usuario

Nombre		Administración de usuario	
Nro. Caso de uso	11		
Actores	Administrador		
Propósito	Gestionar y asignar contraseña a los usuarios		
Descripción	El administrador gestionara y registrara los datos del usuario y le asignara una contraseña, para poder acceder al módulo que le corresponda.		
Flujo principal	Eventos del actor	Eventos del sistema	
	1. El administrador ingresa al sistema con su nombre de usuario y contraseña.	2. El sistema autentifica el usuario y despliega el menú principal.	
	3. Selecciona la opción nuevo usuario.	4. Despliega formulario para el nuevo usuario.	
	5. Llenan el formulario con los datos requeridos.		
	6. Acepta guardar el registro del nuevo usuario.	7. El sistema inserta el nuevo registro en la base de datos.	
Flujo alternativo	2. El usuario ingresa su nombre de usuario o contraseña errónea. 5. No se llena el formulario de manera adecuada.		
Precondición	El administrador ha realizado correctamente el ingreso al sistema.		
Postcondición	El sistema debe registrar al nuevo usuario.		

Fuente: (Elaboración Propia)

3.3.2.7. Diagrama de Secuencia. (ID BACKLOG 6)

La secuencia de eventos donde se observa la iteración de los diferentes componentes del sistema se muestra en las figuras siguientes, mostrando las secuencias respectivas en las operaciones.



Figura 3.14. Diagrama De Secuencia: Registro de Locatario

Fuente: (Elaboración Propia)



Figura 3.15. Diagrama De Secuencia: Estado de vivienda

Fuente: (Elaboración Propia)

HISTORIAL ALQUILER

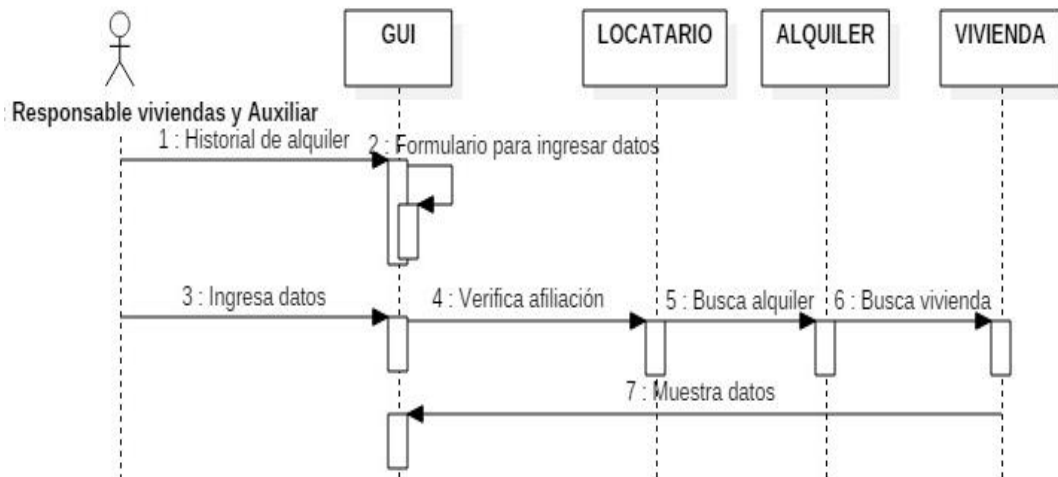


Figura 3.16. Diagrama De Secuencia: Historial de alquiler

Fuente: (Elaboración Propia)

ASIGNACIÓN DE VIVIENDA

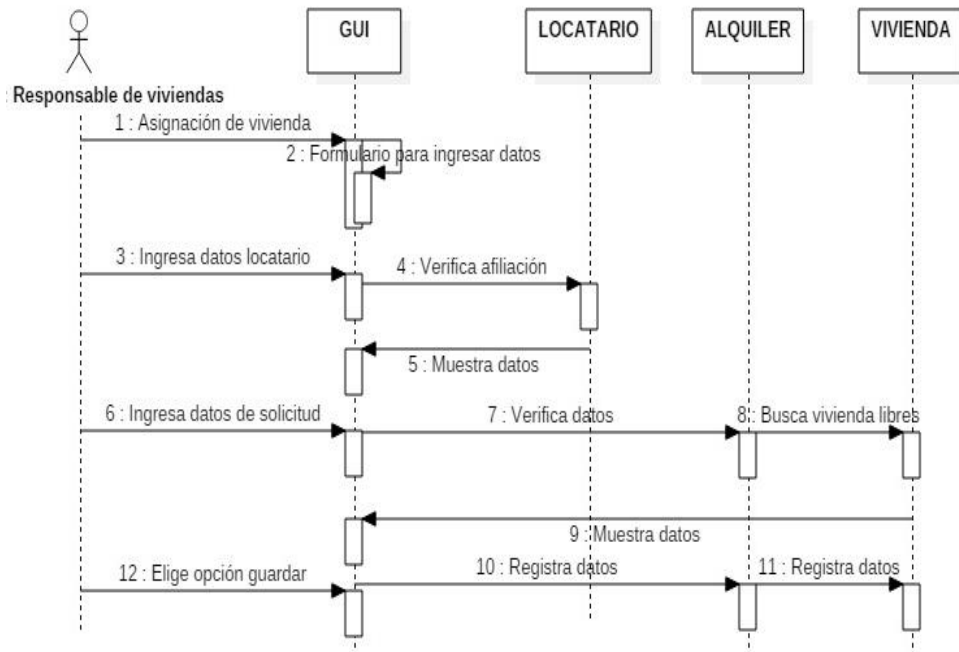


Figura 3.17. Diagrama De Secuencia: Asignación de viviendas

Fuente: (Elaboración Propia)

DESCUENTOS MENSUALES

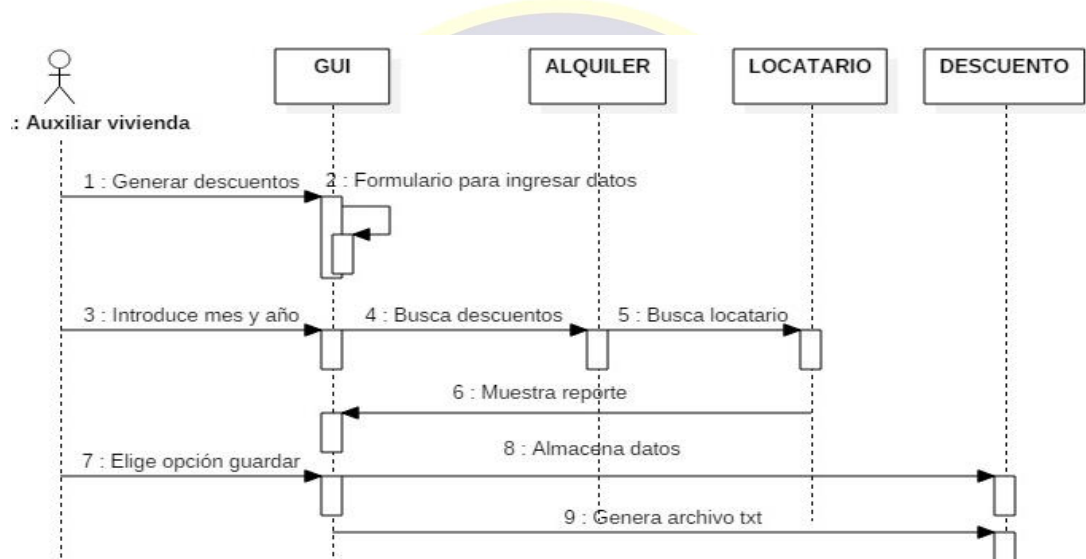


Figura 3.18. Diagrama De Secuencia: Descuentos Mensuales

Fuente: (Elaboración Propia)

REGISTRO DE PAGOS

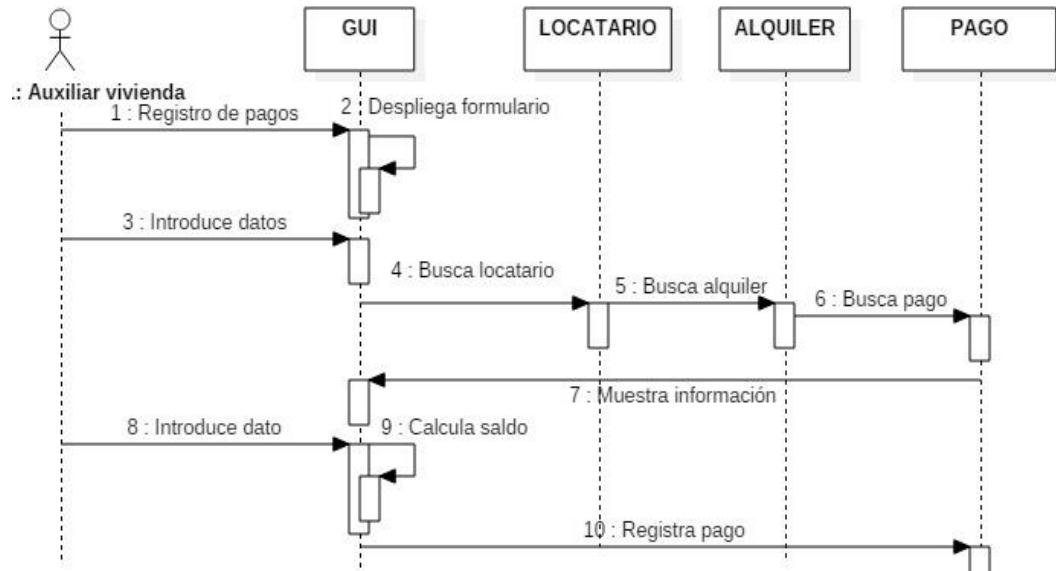


Figura 3.19. Diagrama De Secuencia: Registro de pagos

Fuente: (Elaboración Propia)

CONTROL DE PAGO

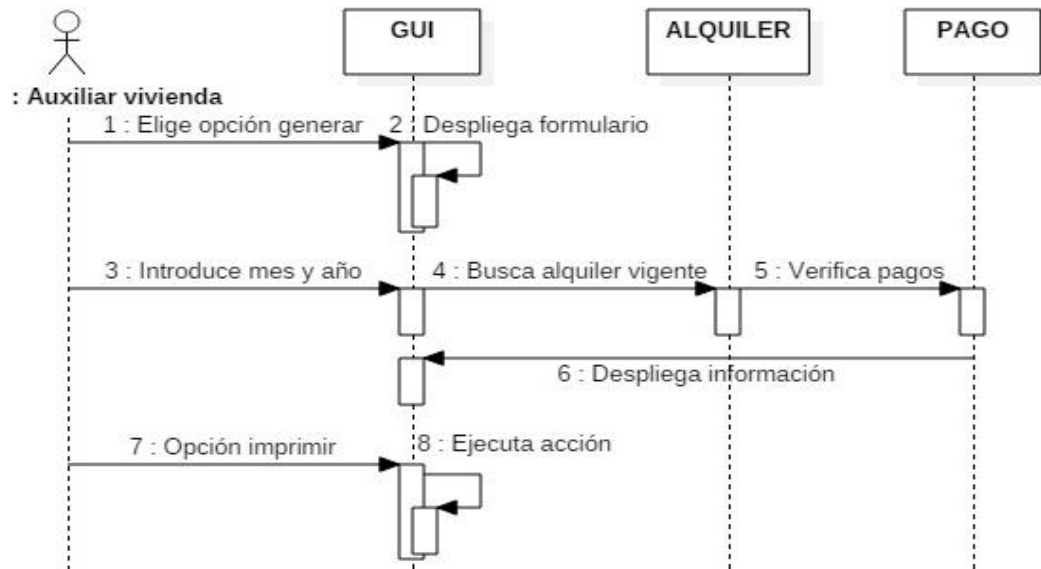


Figura 3.20. Diagrama De Secuencia: Control de pago

Fuente: (Elaboración Propia)

ESTIMACIÓN DE INGRESOS

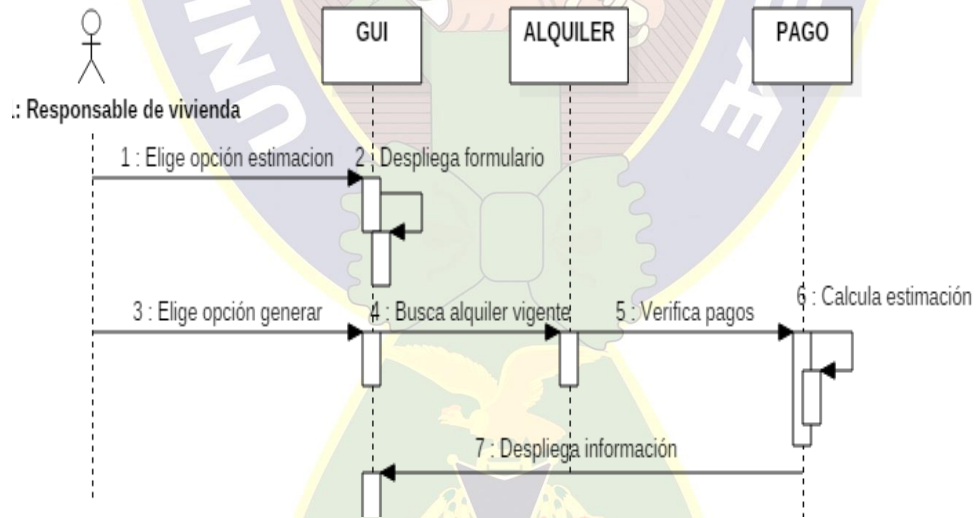


Figura 3.21. Diagrama De Secuencia: Estimación de ingresos

Fuente: (Elaboración Propia)

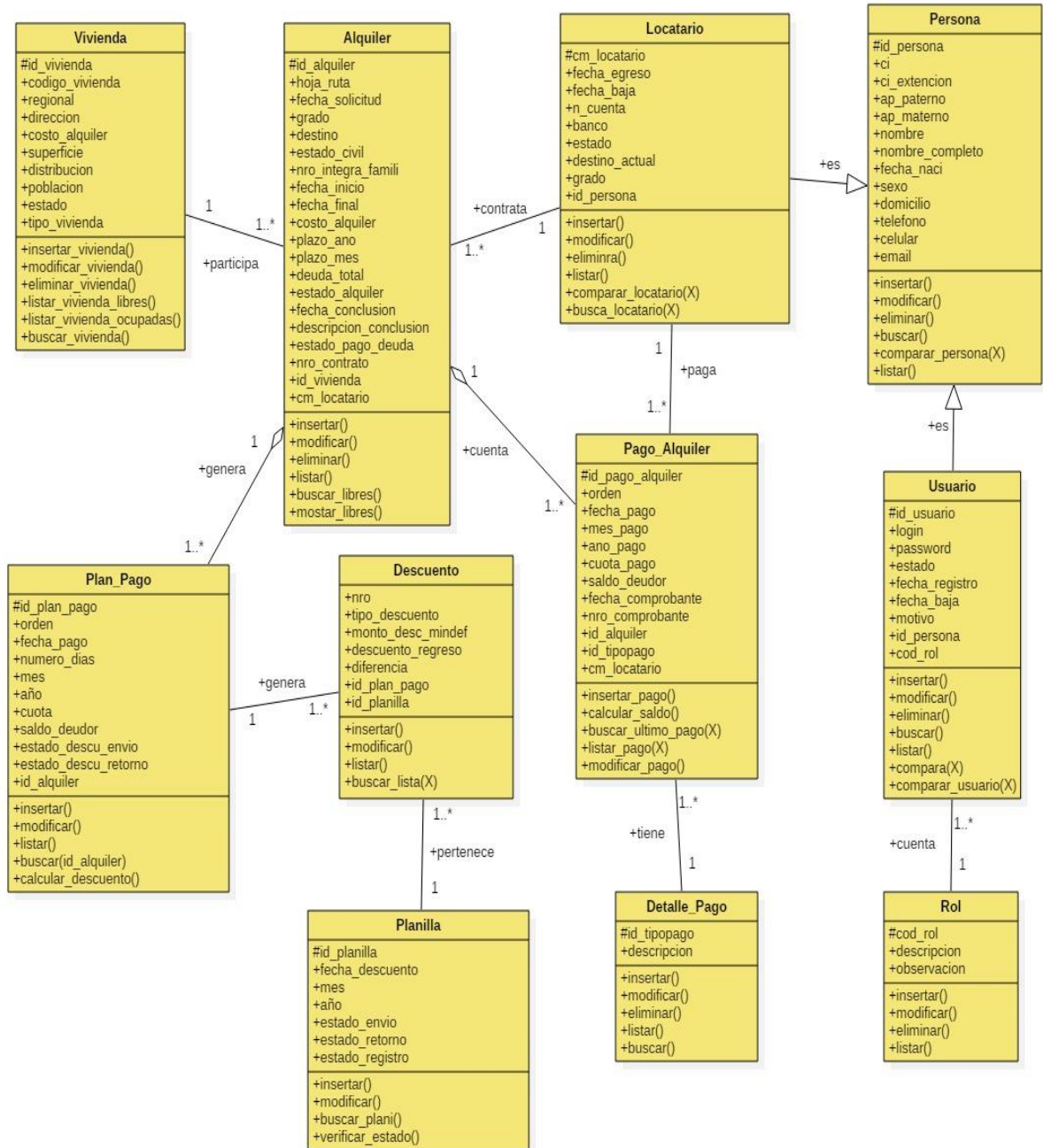


Figura 3.23. Diagrama de Clases

Fuente: (Elaboración Propia)

3.3.2.10. Diagrama Físico. (ID BACKLOG 9)

A continuación, en la figura 3.24 se detalla el diagrama físico de la base de datos.

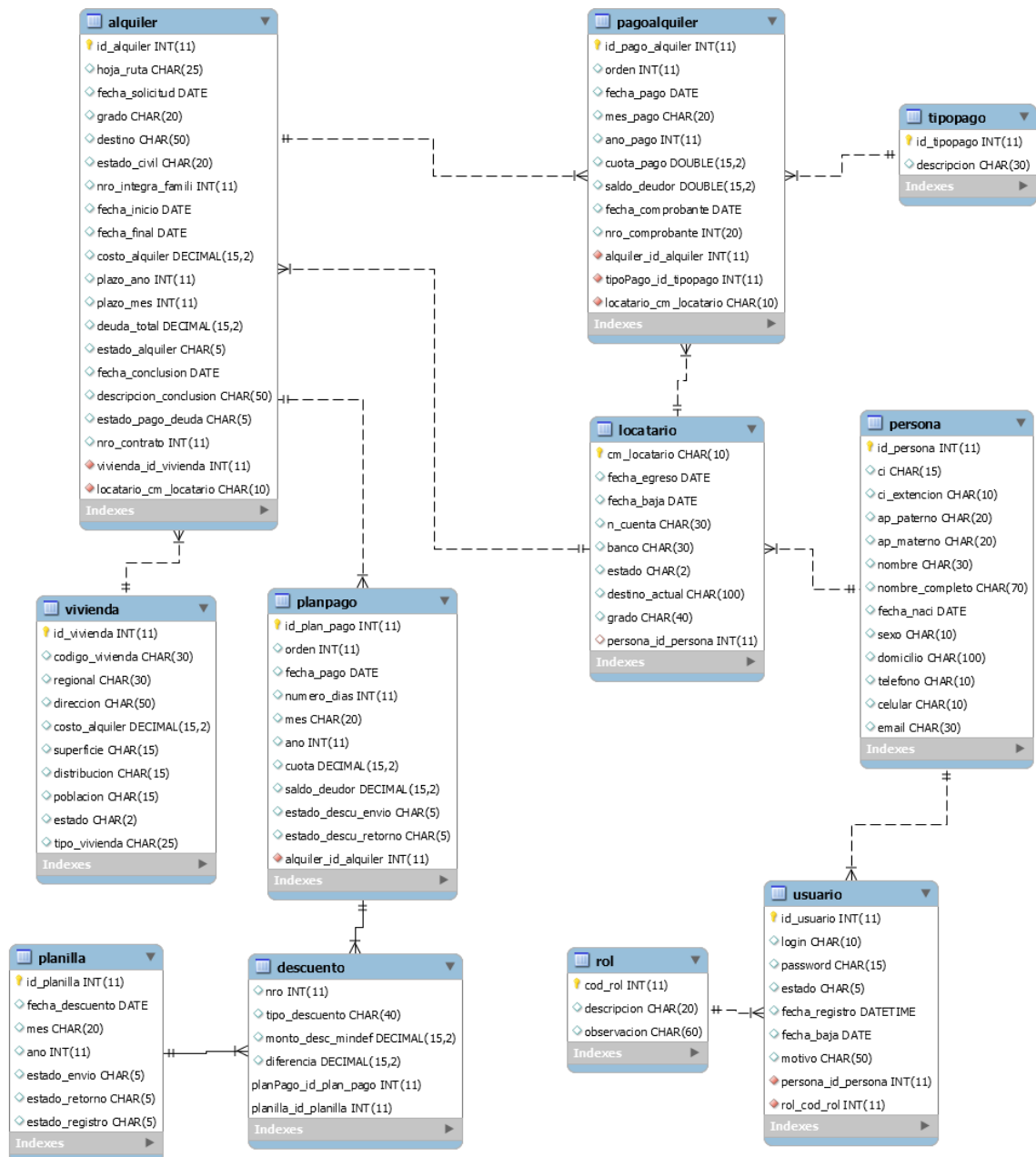


Figura 3.24. Diagrama Físico de la base de datos

Fuente: (Elaboración Propia)

3.3.2.11. Diseño de interfaces de los usuarios. (ID BACKLOG 10)

- Responsable de viviendas - Registro de Locatario.

A continuación se muestra el formulario de registro de los Datos del locatario. Ver Figura 3.25

Figura 3.25. Registro de Datos del Locatario

Fuente: (Elaboración Propia)

➤ Responsable de viviendas – Registro de Vivienda. Ver Figura 3.26

Figura 3.26. Registro de Datos de vivienda

Fuente: (Elaboración Propia)

- Responsable de viviendas y Auxiliar - Estado de vivienda. Ver Figura 3.27

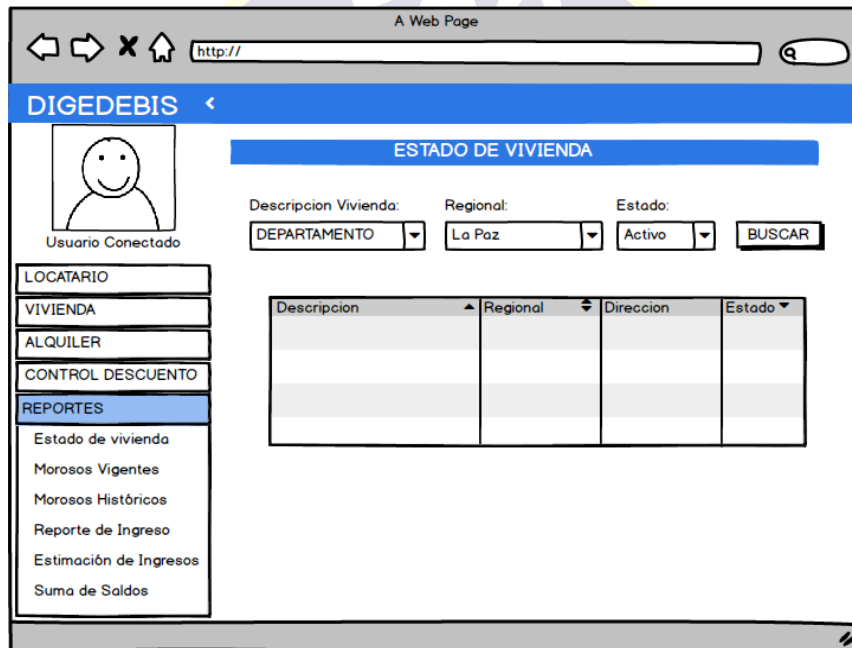


Figura 3.27. Reporte de los estados de las Viviendas
Fuente: (Elaboración Propia)

- Responsable de vivienda y Auxiliar – Búsqueda. Ver Figura 3.28

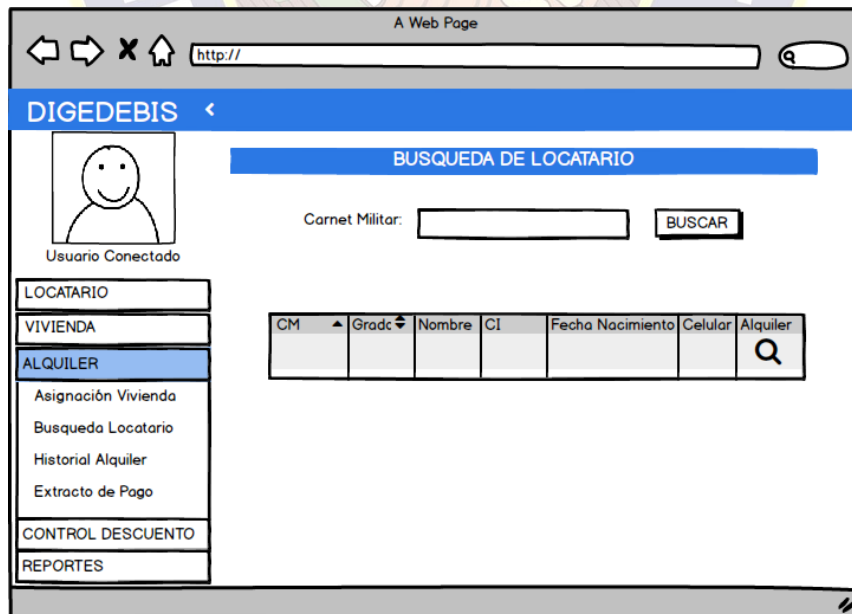


Figura 3.28. Búsqueda de locatario
Fuente: (Elaboración Propia)

- Responsable de vivienda y Auxiliar - Historial de Alquiler. Ver Figura 3.29

A Web Page

http://

DIGEDEBIS <

Usuario Conectado

HISTORIAL DE ALQUILER

Carnet Militar: Grado: Nombre Completo:

C.I.: Celular: Destino:

ALQUILER VIGENTE

Regional	Direccio	Tipo Vivient	Fecha Inicic	Fecha Final	Costo	Plazo Mensual	Estado	Extracto

ALQUILER HISTORICOS

Regional	Direccio	Tipo Vivient	Fecha Inicic	Fecha Final	Costo	Plazo Mensual	Estado	Extracto

LOCATARIO

VIVIENDA

ALQUILER

Asignación Vivienda

Busqueda Locatario

Historial Alquiler

Extracto de Pago

CONTROL DESCUENTO

REPORTES

Figura 3.29. Reporte de historial de alquileres

Fuente: (Elaboración Propia)

- Responsable de vivienda y Auxiliar – Extracto de Pago. Ver Figura 3.30

A Web Page

http://

DIGEDEBIS <

Usuario Conectado

EXTRACTO DE PAGO

Carnet Militar: Nombre Completo: C.I.: Celular:

Regional: Direccion: Fecha inicio: Fecha final:

Costo Alquiler: Plazo Anual: Plazo mensual:

ALQUILER VIGENTE

Nro	Tipo Pago	Año	Meso	Cuota	Saldo Deudor

LOCATARIO

VIVIENDA

ALQUILER

Asignación Vivienda

Busqueda Locatario

Historial Alquiler

Extracto de Pago

CONTROL DESCUENTO

REPORTES

Figura 3.30. Extracto de pagos

Fuente: (Elaboración Propia)

- Responsable de vivienda - Asignación de vivienda.

A continuación se muestra el diseño de la asignación de vivienda. Ver las Figuras 3.31, 3.32, 3.33 y 3.34.

Figura 3.31. Registro de la Solicitud

Fuente: (Elaboración Propia)

- Responsable de vivienda – Asignación de Vivienda. Ver Figura 3.32

Figura 3.32. Asignación de Vivienda

Fuente: (Elaboración Propia)

- Responsable de vivienda – Plan de pagos. Ver Figura 3.33

The screenshot shows a web browser window with the address bar containing 'http://'. The page title is 'DIGEDEBIS'. On the left, there is a navigation menu with the following items: LOCATARIO, VIVIENDA, ALQUILER (highlighted), Asignación Vivienda, Búsqueda Locatario, Historial Alquiler, Extracto de Pago, CONTROL DESCUENTO, and REPORTES. The main content area is titled 'ASIGNACION VIVIENDA' and has four tabs: Reg. Solicitante, Asignación Vivienda, Calculo Pago, and Confirmacion Datos. The 'Asignación Vivienda' tab is active, displaying a form with the following fields: Fecha ini. Contra, Fecha Fin. Contra, Plazo Mensual, Nro. Contrato, Conto Mensual, Fecha primera cuota, and Deuda Total. A 'Generar plan pago' button is located at the bottom right of this form. Below this is the 'PLAN DE PAGO' section, which contains a table with the following columns: Nro. Cu., Fecha, Dia, Mes, Año, Descuento, and Saldo. The table is currently empty. Navigation buttons '< Back' and 'Next >' are located at the bottom of the page.

Figura 3.33. Plan de pago

Fuente: (Elaboración Propia)

- Responsable de vivienda – Confirmación de datos. Ver Figura 3.34

The screenshot shows the same web browser window as Figure 3.33. The 'Confirmacion Datos' tab is active, displaying a form with two main sections: 'Datos del Solicitante' and 'Datos de la Vivienda'. The 'Datos del Solicitante' section includes fields for Hoja de ruta, Fecha solicitud, Solicitante, Carnet militar, Destino, Celular, Estado civil, Nro. Int. Fila., and Plazo anual. The 'Datos de la Vivienda' section includes fields for Regional, Tipo vivienda, Direccion, Costo, Fecha Ini. Contr., Fecha Fin. Contr., Plazo mensual, and Nro. Contrato. At the bottom of the form are two buttons: 'REGISTRAR' and 'CANCELAR'. Navigation buttons '< Back' and 'Next >' are located at the bottom of the page.

Figura 3.34. Confirmación de datos

Fuente: (Elaboración Propia)

- Auxiliar de Vivienda - Generar planilla de descuentos. Ver Figura 3.35

The screenshot shows a web browser window with the URL 'http://'. The page title is 'DIGEDEBIS'. On the left is a navigation menu with options: LOCATARIO, VIVIENDA, ALQUILER, CONTROL DESCUENTO (highlighted), and REPORTES. Under 'CONTROL DESCUENTO' are sub-options: Generar Descuentos, Registro Descuentos, and Registro de pago. The main content area is titled 'GENERAR PLANILLA DE DESCUENTOS'. It features a 'Usuario Conectado' profile icon and the instruction 'Elija la fecha para generar los descuentos'. There are three input fields: 'Mes:' (dropdown), 'Año:' (dropdown), and 'Nro. planilla:' (text). Below these are buttons for 'GENERAR', 'GUARDAR', and 'CANCELAR'. A table titled 'DESCUENTOS MENSUALES' is displayed with the following columns: Nro., CM, Locatari, Vivienc, Regional, Fecha Desc, Deudado, Saldo, and Descu.

Figura 3.35. Generar planilla de descuentos

Fuente: (Elaboración Propia)

- Auxiliar de Vivienda – Registro de Descuentos. Ver Figura 3.36 y 3.37

The screenshot shows a web browser window with the URL 'http://'. The page title is 'DIGEDEBIS'. On the left is a navigation menu with options: LOCATARIO, VIVIENDA, ALQUILER, CONTROL DESCUENTO (highlighted), and REPORTES. Under 'CONTROL DESCUENTO' are sub-options: Generar Descuentos, Registro Descuentos, and Registro de pago. The main content area is titled 'IMPORTAR PLANILLA DE DESCUENTOS'. It features a 'Usuario Conectado' profile icon and the instruction 'Seleccione el archivo a registrar.'. There are buttons for 'Escoger', 'Cargar', and 'Cancelar'. A file selection box shows 'DescuentoAbril2021.txt' with a close button 'X'. There are three input fields: 'Mes:' (dropdown), 'Año:' (dropdown), and 'Nro. planilla:' (dropdown). Below these are buttons for 'IMPORTAR', 'GUARDAR', and 'CANCELAR'. A table titled 'DESCUENTOS MENSUALES' is displayed with the following columns: Nro.,Codigo Desc., CM, Nombre Completo, and Descuento.

Figura 3.36. Importar Planilla

Fuente: (Elaboración Propia)

- Auxiliar de Vivienda – Registro Descuento. Ver Figura 3.37

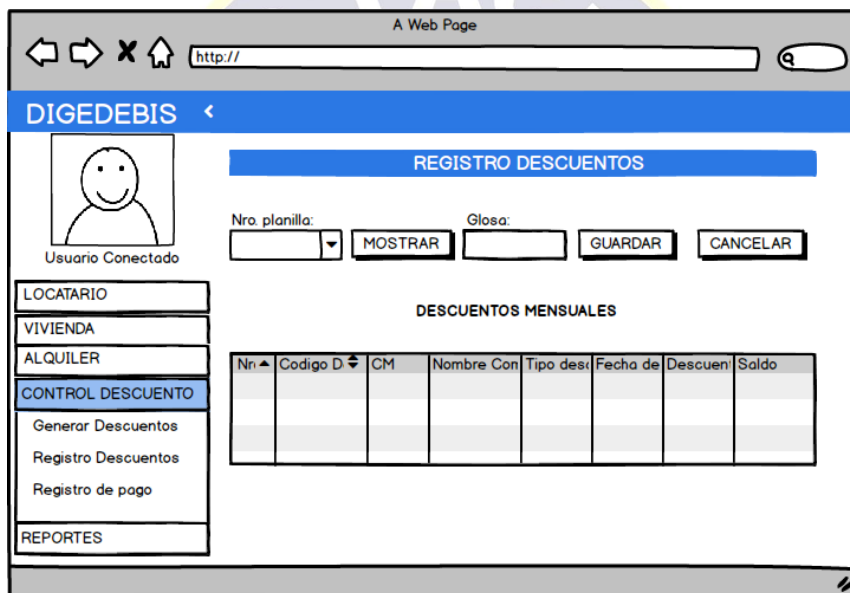


Figura 3.37. Registro Descuento

Fuente: (Elaboración Propia)

- Responsable y Auxiliar de Vivienda – Morosos Vigentes. Ver Figura 3.38

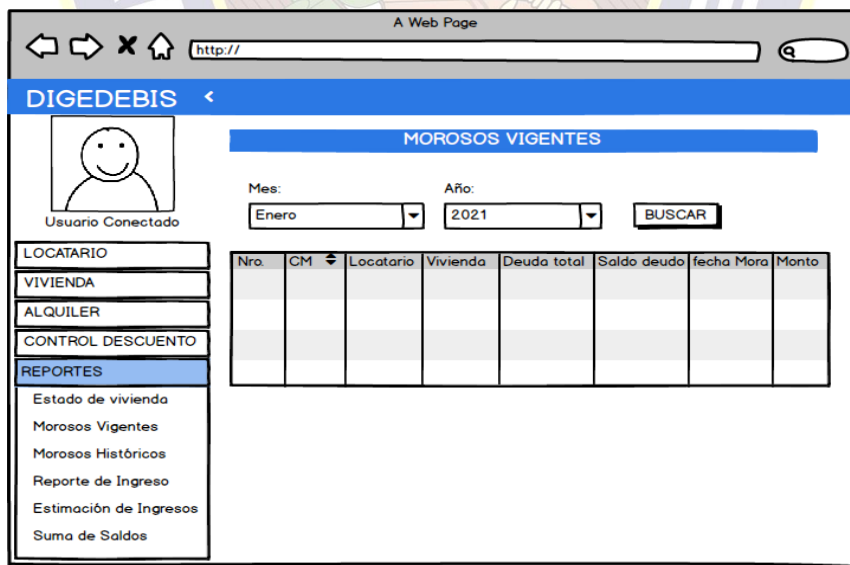


Figura 3.38. Morosos Vigentes

Fuente: (Elaboración Propia)

- Responsable de Vivienda – Reporte de Ingresos. Ver Figura 3.39

Figura 3.39. Reporte de Ingresos

Fuente: (Elaboración Propia)

- Responsable y Auxiliar de Vivienda – Moras Históricas. Ver Figura 3.40

Figura 3.40. Moras Históricas

Fuente: (Elaboración Propia)

- Responsable de Vivienda – Estimación de Ingresos. Ver Figura 3.41

The screenshot shows the 'ESTIMACIÓN DE INGRESOS' page in the DIGEDEBIS system. The interface includes a user profile section on the left, a search area for month and year, and a table of income estimates.

Regional^	Cantidad	Monto de Ingresos	Fecha
La Paz	40	1.500.00 Bs	Enero/2021
Santa Cruz	38	2.230.12 Bs	Enero/2021
Cochabamba	41	950.74 Bs	Enero/2021
TOTAL	119	4.680.86 Bs	

Figura 3.41. Estimación de Ingresos

Fuente: (Elaboración Propia)

- Responsable de Vivienda – Suma de Saldo. Ver Figura 3.42

The screenshot shows the 'SUMA DE SALDOS' page in the DIGEDEBIS system. The interface includes a user profile section on the left, buttons for 'GENERAR', 'EXPORTAR', and 'IMPRIMIR', and a table with columns for Nro., CM, Locatario, Vivienda, Deuda Total, Saldo Deudor, fecha Inicio, fecha Final, and Monto Cancelado.

Nro.	CM	Locatario	Vivienda	Deuda Total	Saldo Deudor	fecha Inicio	fecha Final	Monto Cancelado

Figura 3.42. Suma de Saldo

Fuente: (Elaboración Propia)

3.3.3. Segunda Iteración (SPRINT BACKLOG 2)

El sprint 2 fue desarrollado bajo el siguiente detalle técnico. Las actividades realizadas durante esta iteración fueron planificadas y desarrolladas según lo planeado, además que se consideran módulos necesarios para el funcionamiento total del sistema. Ver tabla 3.36

Tabla 3.36. Sprint Backlog 2

ID	TAREA	TIPO	ESTADO	RESPONSABLE	DÍAS DE TRABAJADO
1	Módulo, afiliación de locatario.	Desarrollo	Terminado	Fernando Laura Torrez	3
2	Módulo, afiliación de viviendas	Desarrollo	Terminado	Fernando Laura Torrez	5
3	Módulo, visualización de viviendas ocupadas y disponibles.	Desarrollo	Terminado	Fernando Laura Torrez	5
4	Módulo búsqueda de locatario	Desarrollo	Terminado	Fernando Laura Torrez	3
5	Módulo, histórico de alquiler, de locatario.	Desarrollo	Terminado	Fernando Laura Torrez	3
6	Módulo, asignación de vivienda.	Desarrollo	Terminado	Fernando Laura Torrez	3
7	Extracto de pago de alquiler.	Desarrollo	Terminado	Fernando Laura Torrez	3
SPRINT		INICIO		FIN	
2		22/3/2021		25	

Fuente: (Elaboración Propia)

3.3.3.1. Módulo Afiliación (ID BACKLOG 1)

En base a los requisitos e historia de usuario, el Administrador de la Vivienda, dispone de un módulo, donde podrá registrar y editar los datos del Locatario.

a) Módulo Afiliación – Análisis

El módulo se encarga de realizar el registro de los datos del nuevo Locatario, actualizar y registrar su baja.

b) Módulo Afiliación – Desarrollo

A continuación, se despliega el modulo donde se realiza la filiación. Ver figuras 3.43

The screenshot shows a web browser window displaying the DIGEBEBIS application. The page title is 'FORMULARIO DE REGISTRO DE LOCATARIO'. The form is divided into two main sections. The first section contains fields for: Nombres (*), Ap. Paterno (*), Ap. Materno (*), Sexo (*), Cl: (*), Extencion: (*), Fecha Nacimiento (*), Telefono:, Celular: (*), Email:, and Domicilio: (*). The second section contains fields for: Carnet Militar (*), Grado (*), Destino (*), Fecha Egreso (*), Banco (*), and Nro. Cuenta (*). The 'Registrar' button is highlighted in blue.

Nombres: *	Ap. Paterno: *	Ap. Materno: *
Mario Gonzalo	Mamani	Tola
Sexo: *	Cl: *	Extencion: *
<input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	6158972	La Paz
Fecha Nacimiento *	Telefono:	Celular: *
02-Sep-1990		79628684
Email:	Domicilio: *	
Mario@gmail.com	Zona: Achachicala calle: Eusebio Gutiérrez Nro. 661	
Carnet Militar *	Grado *	Destino *
31208021	Sfo 1ro	Primara Brigara
Fecha Egreso *	Banco *	Nro. Cuenta *
01-Ene-2021	Union	123456789

Figura 3.43. Afiliación de Locatario

Fuente: (Elaboración Propia)

3.3.3.2. Modulo Afiliación de Vivienda (ID BACKLOG 2)

En base a los requisitos e historia de usuario, el Administrador de las Viviendas requiere de un módulo donde puede realizar el registro de la nueva Vivienda.

a) Módulo Afiliación de Vivienda – Análisis

El módulo se encarga de registrar los datos de la nueva Vivienda y poder realizar actualizaciones a estos registros.

b) Módulo Afiliación de Vivienda – Desarrollo

A continuación, se muestra el módulo de Afiliación de Vivienda. Ver figuras 3.44

The screenshot shows a web browser window with the URL `localhost:8080/SisVivienda/faces/prottegido/Operador/RegistraViviendaOP.xhtml`. The application header displays the logo 'DIGEDEBIS' and a user profile 'ra ra ra OPERADOR'. A left sidebar contains navigation options: 'Locatario', 'Vivienda' (selected), 'Registrar', 'Actualizar', 'Alquiler', 'Control Descuentos', and 'Reportes'. The main content area is titled 'REGISTRO VIVIENDA FUNCIONAL' and contains the following form fields:

- Codigo Vivienda ***: Input field with value 'co-a'
- Regional: ***: Dropdown menu with value 'La Paz'
- Costo del Alquiler: ***: Input field with value '500,00 Bs.'
- Superficie: ***: Input field with value '50 M2'
- Distribución: ***: Input field with value 'Sala, Cocina, Baño y 3 Dormitorios'
- Poblacion: ***: Input field with value '4 Personas'
- Descripción de Vivienda ***: Dropdown menu with value 'Departamento'
- Dirección: ***: Input field with value 'Zona los Olivos calle primero Nro. 456'

At the bottom of the form are two buttons: 'Registrar' and 'Cancelar', both with checkmark icons.

Figura 3.44. Afiliación de Vivienda

Fuente: (Elaboración Propia)

3.3.3.3. Módulo Estado de las Viviendas (ID BACKLOG 3)

En base a los requisitos e historia de usuario, el Administrador dispone del módulo de listado de las viviendas.

a) Módulo Estado de las Viviendas – Análisis

En este módulo despliega la información de las viviendas libres y alquiladas, hasta la fecha.

b) Módulo Estado de las Viviendas – Desarrollo

A continuación, se despliega el modulo donde muestra el estado de las viviendas. Ver figuras 3.45

ESTADO DE VIVIENDA

Regional: * La Paz Tipo de Vivienda: * Habitación Estado del Alquiler: * Alquilado

TABLA DE LOCATARIOS

Página: (1 de 1)

Codigo	Regional	Tipo Vivienda	Direccion	Costo	Estado	Ver
1	LA PAZ	DEPARTAMENTO	CALACOTO	700.00	Alquilado	
31	LA PAZ	VIVIENDAS	EL ALTO	700.00	Alquilado	
91	LA PAZ	HABITACION	SOLTEROS LP	700.00	Alquilado	

Página: (1 de 1)

Figura 3.45. Estado de las Viviendas

Fuente: (Elaboración Propia)

3.3.3.4. Módulo Búsqueda de Locatario (ID BACKLOG 4)

En base a los requisitos e historia de usuario, el de administrador dispone de un módulo donde puede realizar la búsqueda del locatario y si existe se podrá actualizar su información.

a) Módulo Búsqueda de Locatario – Análisis

En este módulo se puede realizar la búsqueda por diferentes filtros que mencionaremos a continuación:

- Búsqueda por su carnet de identidad.
- Búsqueda por nombres, apellido paterno o apellido materno
- Sección de búsqueda por su carnet militar.

b) Módulo Búsqueda de Locatario – Desarrollo

A continuación, se muestra el entorno donde se muestra los diferentes filtros de búsqueda ayudándole al usuario a realizar una búsqueda efectiva y eficiente. Ver figuras 3.46

The screenshot displays the DIGEBEBIS web application interface for tenant search. The top navigation bar shows the user profile 'TERAN POMIER OMAR OPERADOR'. The main content area is titled 'BÚSQUEDA DE LOCATARIO' and features a search bar with a dropdown menu set to 'CARNET MILITAR' and a search input field containing '31208018'. A search button is visible to the right of the input field. Below the search bar, a table titled 'DATOS DE LOCATARIOS' is displayed. The table has a pagination control showing 'Página: (1 de 1)' and a dropdown menu set to '5'. The table columns are: CM, Grado, Nombre, Carnet Identidad, fecha Nacimiento, Celular, and Alquiler. The table contains one row of data: CM 31208018, Grado SOF.INL, Nombre CHIRINOS MACHACA FAVIO, Carnet Identidad 6019070, LPZ, fecha Nacimiento 28/02/1984, Celular 0, and Alquiler with a search icon. The bottom of the table also shows a pagination control for 'Página: (1 de 1)' with a dropdown menu set to '5'.

Figura 3.46. Búsqueda de Locatario

Fuente: (Elaboración Propia)

3.3.3.5. Módulo Historial de Alquileres (ID BACKLOG 5)

En base a los requisitos e historia de usuario, del administrador, dispone de un módulo donde puede visualización el historial de las viviendas asignadas a un locatario.

a) Módulo Historial de Alquileres – Análisis

En este módulo se despliega un reporte con el historial de las viviendas asignadas a un locatario.

b) Módulo Historial de Alquileres – Desarrollo

A continuación, se despliega el módulo donde se muestra el extracto de pago. Ver figuras 3.47 y 3.48

BÚSQUEDA DE LOCATARIO

CARNET MILITAR * 11090015

DATOS DE LOCATARIOS

CM	Grado	Nombre	Carnet Identidad	fecha Nacimiento	Celular	Alquiler
11090015	CNL. DAENMG.	ZENTENO PARDO JOSE	1087813, CH	24/07/1968	0	

Figura 3. 47. Búsqueda de Locatario

Fuente: (Elaboración Propia)

HISTORIAL DE ALQUILER

Carnet Militar: 11090015 | Grado: CNL. DAENMG. | Nombre Completo: ZENTENO PARDO JOSE

Carnet d Identidad: 1087813 | Telefono: 0 | Destino: IV-BRIG.AEREA

ALQUILER VIGENTE

Regional	Direccion	Tipo de Vivienda	Fecha Inicio Contrato	fecha final del Contrato	Costo del Alquiler	Plazo Mensual	Deuda Total	Saldo Deudor	Estado	Extracto de Pago
LA PAZ	SOLTEROS LP	HABITACIO	10/05/2021	31/12/2022	700.00	20	14000.00	5249.40	Vigente	

Figura 3. 48. Historial de Alquileres

Fuente: (Elaboración Propia)

3.3.3.6. Módulo Asignación de Vivienda (ID BACKLOG 6)

En base a los requisitos e historia de usuario, el administrador requiere de un módulo donde puede realizar el registro de la solicitud de viviendas, asignarle la vivienda disponible, calcular la deuda total y generar un plan de pagos.

a) Módulo Historial de Alquileres – Análisis

El módulo se encarga de registrar la solicitud de préstamo y los datos del solicitante, asignarle una vivienda disponible, calcular su deuda total y generar un plan de pagos.

b) Módulo Historial de Alquileres – Desarrollo

A continuación, se muestra el módulo de registro de la solicitud. Ver figuras 3.49, 3.50, 3.51 y 3.52

The screenshot shows a web browser window with the URL `localhost:8080/SisVivienda/faces/protegido/Operador/BuscaUsuarioExtracto.xhtml`. The application header includes the logo 'DIGEDEBIS' and the user profile 'TERAN POMIER OMAR OPERADOR'. A left sidebar contains navigation options: Locatario, Vivienda, Alquiler (selected), Asignación de Alquiler, Consulta Alquiler, Control Descuentos, and Reportes. The main content area is titled 'HISTORIAL DE ALQUILER' and contains a form with the following fields:

- Carnet Militar: 11090015
- Grado *: CNL. DAENMG.
- Nombre Completo *: ZENTENO PARDO JOSE
- Carnet d Identidad *: 1087813
- Telefono *: 0
- Destino *: IV-BRIG.AEREA

Below the form, there are tabs for 'VIGENTE' and 'HISTORIAL'. The 'VIGENTE' tab is active, displaying a table titled 'ALQUILER VIGENTE'. The table has the following data:

Regional	Direccion	Tipo de Vivienda	Fecha Inicio Contrato	fecha final del Contrato	Costo del Alquiler	Plazo Mensual	Deuda Total	Saldo Deudor	Estado	Extracto de Pago
LA PAZ	SOLTEROS LP	HABITACIO	10/05/2021	31/12/2022	700.00	20	14000.00	5249.40	Vigente	

Figura 3. 49. Registro de Solicitud

Fuente: (Elaboración Propia)

Registro de Solicitante **Asignación Vivienda** Calculo de Pago Confirmación de datos

Datos Vivienda

Selección Regional: * Selección Vivienda: * Plazo: *

Regional: Vivienda: Dirección: Costo:

TABLA DE VIVIENDA

Página: (1 de 14) 1 2 3 4 5 6 7 8 9 10 4

Regional	Tipo Vivienda	Dirección	Costo	Asignar
LA PAZ	VIVIENDAS	MIRAFLORES VIVIENDA CMDTE.	700.00	<input type="button" value="+"/>
LA PAZ	VIVIENDAS	EL ALTO	700.00	<input type="button" value="+"/>
LA PAZ	VIVIENDAS	EL ALTO	700.00	<input type="button" value="+"/>
LA PAZ	VIVIENDAS	EL ALTO	700.00	<input type="button" value="+"/>

Página: (1 de 14) 1 2 3 4 5 6 7 8 9 10 4

Figura 3.50. Asignación de Vivienda

Fuente: (Elaboración Propia)

Registro de Solicitante **Asignación Vivienda** **Calculo de Pago** Confirmación de datos

Información de Pago

Fecha de Inicio del Contrato: * Fecha de Final del Contrato: * Plazo Mensual: * Nro. del Contrato de Alquiler: *

Costo Mensual del Alquiler: Fecha del Primer Descuento: * Deuda Total a Cancelar:

PLAN DE PAGOS

Nro. Cuota	Fecha	Días	Mes	Año	Descuento Mensual Bs.	Saldo Ac. Bs.
0	04-05-2021	0	Mayo	2021	700,00	14,000,00
1	31-05-2021	27	Mayo	2021	700,00	13,300,00
2	30-06-2021	30	Junio	2021	700,00	12,600,00
3	31-07-2021	31	Julio	2021	700,00	11,900,00
4	31-08-2021	31	Agosto	2021	700,00	11,200,00
5	30-09-2021	30	Septiembre	2021	700,00	10,500,00
6	31-10-2021	31	Octubre	2021	700,00	9,800,00
7	30-11-2021	30	Noviembre	2021	700,00	9,100,00
8	31-12-2021	31	Diciembre	2021	700,00	8,400,00
9	31-01-2022	31	Enero	2022	700,00	7,700,00
10	28-02-2022	28	Febrero	2022	700,00	7,000,00

Figura 3.51. Calculo del Pago

Fuente: (Elaboración Propia)

Registro de Solicitante	Asignación Vivienda	Calculo de Pago	Confirmación de datos
Confirmación			
Datos del Solicitante			
Nro. de Hoja de Ruta:	Fecha de Solicitud:	Solicitante:	
<input type="text" value="124"/>	<input type="text" value="04-05-2021"/>	<input type="text" value="CAP. MACEDO MOYA JUAN"/>	
Carnet Militar del Solicitante:	Destino Actual:	Celular:	
<input type="text" value="11008025"/>	<input type="text" value="G.A.ENTR. 21"/>	<input type="text" value="0"/>	
Estado Civil:	Nro. Integrantes Familiares :	Plazo Anual Solicitado:	
<input type="text" value="CASADO"/>	<input type="text" value="3 Integrantes"/>	<input type="text" value="2 Año(s)"/>	
Datos de la Vivienda Asignada			
Regional:	Vivienda:	Dirección :	Costo :
<input type="text" value="LA PAZ"/>	<input type="text" value="VIVIENDAS"/>	<input type="text" value="EL ALTO"/>	<input type="text" value="700,00 Bs."/>
Fecha de Inicio del Contrato:	Fecha de Final del Contrato:	Plazo Mensual:	Nro. del Contrato de Alquiler:
<input type="text" value="04-05-2021"/>	<input type="text" value="31-12-2022"/>	<input type="text" value="20 Meses"/>	<input type="text" value="152"/>
<input type="button" value="REGISTRA"/>		<input type="button" value="CANCELAR"/>	

[← Back](#)

Figura 3.52. Registrar Datos

Fuente: (Elaboración Propia)

3.3.3.7. Módulo de Extracto de pago. (ID BACKLOG 7)

En base a los requisitos e historia de usuario, el Administrador, dispone de un módulo donde visualizara los pagos realizados.

a) Módulo Extracto de pago – Análisis

En este módulo se despliega un reporte de los pagos realizados por alquileres vigentes e históricos.

b) Módulo Extracto de pago – Desarrollo

A continuación, se despliega el módulo donde se muestra el extracto de pago. Ver figuras 3.53, 3.54 y 3.55

BÚSQUEDA DE LOCATARIO

Aviso: Usuario encontrado

CARNET MILITAR * 11090015

DATOS DE LOCATARIOS

Página: (1 de 1)

CM	Grado	Nombre	Carnet Identidad	fecha Nacimiento	Celular	Alquiler
11090015	CNL. DAENMG.	ZENTENO PARDO JOSE	1087813, CH	24/07/1968	0	

Página: (1 de 1)

Figura 3. 53. Búsqueda de Locatario

Fuente: (Elaboración Propia)

HISTORIAL DE ALQUILER

Carnet Militar: 11090015
 Grado *: CNL. DAENMG.
 Nombre Completo *: ZENTENO PARDO JOSE
 Carnet d Identidad *: 1087813
 Telefono *: 0
 Destino *: IV-BRIG.AEREA

ALQUILER VIGENTE

Página: (1 de 1)

Regional	Direccion	Tipo de Vivienda	Fecha Inicio Contrato	fecha final del Contrato	Costo del Alquiler	Plazo Mensual	Deuda Total	Saldo Deudor	Estado	Extracto de Pago
LA PAZ	SOLTEROS LP	HABITACIO	10/05/2021	31/12/2022	700.00	20	14000.00	5249.40	Vigente	

Página: (1 de 1)

Figura 3. 54. Historial de Alquileres

Fuente: (Elaboración Propia)

EXTRACTO DE PAGO

Regional * LA PAZ Dirección * SOLTEROS LP Fecha Inicio Contrato * 10/05/2021 Fecha Final Contrato * 31/12/2022

Costo Alquiler 700,00 Bs. Plazo Anual * 2 Año(s) Plazo Mensual * 20 Meses

TABLA DE PAGOS

Página: (1 de 3)

Nro.	Tipo Pago	fecha del Pago	Mes	Año	Cuota Bs.	Saldo Deudor Bs.
1	Min. Def. Planilla	28/02/2022	Febrero	2022	700,10	13.299,90
2	Min. Def. Planilla	31/07/2022	Julio	2022	700,00	12.599,90
3	Min. Def. Planilla	28/02/2022	Febrero	2022	700,10	11.899,80
4	Min. Def. Planilla	28/02/2022	Febrero	2022	700,10	11.199,70
5	Min. Def. Planilla		Febrero	2022	700,10	10.499,60
13					8.750,60 Bs.	5.249,40 Bs.

Página: (1 de 3)

Figura 3. 55. Extracto de pago

Fuente: (Elaboración Propia)

3.3.4. Tercera Iteración (SPRINT BACKLOG 3)

Las actividades realizadas durante esta iteración fueron planificadas y desarrolladas según lo planeado, además que se consideran módulos necesarios para el funcionamiento total del sistema. Ver tabla 3.37

Tabla 3.37. Sprint Backlog 3

ID	TAREA	TIPO	ESTADO	RESPONSABLE	Días de trabajos
1	Modulo Generar Descuentos Mensuales	Desarrollo	Terminado	Fernando Laura Torrez	5
2	Modulo Registro masivo de Amortizaciones	Desarrollo	Terminado	Fernando Laura Torrez	4
3	Modulo Registro de Alquiler	Desarrollo	Terminado	Fernando Laura Torrez	5
4	Modulo Control de moras	Desarrollo	Terminado	Fernando Laura Torrez	5
5	Modulo Reporte de ingresos	Desarrollo	Terminado	Fernando Laura Torrez	4
6	Módulo Administración de usuario	Desarrollo	Terminado	Fernando Laura Torrez	4
SPRINT		INICIO		FIN	
3		03/5/2021		27 días	

Fuente: (Elaboración Propia)

3.3.4.1. Módulo generar descuentos mensuales (ID BACKLOG 1)

En base a los requisitos e historia de usuario, el Auxiliar de crédito dispone del módulo donde podrá generar el reporte mensual de los descuentos que se realizará a los socios.

a) Módulo generar descuentos mensuales – Análisis

En este módulo se podrá generar el reporte mensual de los descuentos que se realizaran a los socios que cuenten con préstamos, el cual será enviado al Ministerio de Defensa para su proceso.

b) Módulo generar descuentos mensuales – Desarrollo

A continuación, se despliega el modulo donde se genera el reporte. Ver figuras 3.56 y 3.57.

GENERAR PLANILLA DE DESCUENTOS

Elija la fecha para generar los descuentos

Mes * AGOSTO Año * 2021 Nro. Planilla 55

DESCUENTOS MENSUALES

Nro.	Cod. Descuento	CM	Locatario	Vivienda	Regional	fecha Descuento	Monto Adeudado en Bs.	Saldo	Descuento Bs.
1	4	31208018	SOF.INL.CH MACHACA FAVIO	VIVIENDAS	LA PAZ	Agosto-202	13.300,00	5.298,00	700,00
2	25	11003009	MY. DEMA, KELC CERVANTES MIGUEL	DEPARTAME	LA PAZ	Agosto-202	14.000,00	5.600,00	700,00
3	46	11090015	CNL. DAENMG.Z PARDO JOSE	HABITACION	LA PAZ	Agosto-202	14.000,00	5.249,40	700,00
								16.147,40	2.100,00

Figura 3. 56. Generar descuentos mensuales

Fuente: (Elaboración Propia)

GENERAR PLANILLA DE DESCUENTOS

Elija la fecha para generar los descuentos

Mes * AGOSTO Año * 2021 Nro. Planilla 55

DESCUENTOS MENSUALES

Bloc de notas

```
*1**4**31208018**SOF.INL.CHIRINOS MACHACA FAVIO**VIVIENDAS**LA PAZ**Agosto-2021**13.300,00**5.298,00**700,00*
*2**25**11003009**MY. DEMA.,KELCA CERVANTES MIGUEL**DEPARTAMENTO**LA PAZ**Agosto-2021**14.000,00**5.600,00**700,00*
*3**46**11090015**CNL. DAENMG.,ZENTENO PARDO JOSE**HABITACION**LA PAZ**Agosto-2021**14.000,00**5.249,40**700,00*
```

Figura 3. 57. Generar descuentos mensuales

Fuente: (Elaboración Propia)

3.3.4.2. Módulo Registro de Alquiler (ID BACKLOG 2)

En base a los requisitos e historia de usuario, el Administrador de las Viviendas requiere de un módulo donde puede realizar el registro de la nueva Vivienda.

a) Módulo Registro de Alquiler – Análisis

El módulo se encarga de registrar los pagos por depósito bancario como por boleto de pago.

b) Módulo Registro de Alquiler – Desarrollo

A continuación, se muestra el módulo de Afiliación de Vivienda. Ver figuras 3.58 y 3.59.

Nro.	Tipo Pago	fecha del Pago	Mes	Año	Cuota Bs.	Saldo Deudor Bs.
1	Min. Def. Planilla	30/06/2021	Junio	2021	700,00	12,600,00
2	Min. Def. Planilla	31/07/2021	Julio	2021	700,00	11,900,00
3	Min. Def. Planilla	31/05/2022	Mayo	2022	700,00	11,200,00
4	Depósito Bancario	31/08/2022	Junio	2022	500,00	10,700,00
5	Depósito Bancario	31/08/2022	Agosto	2022	500,00	10,200,00
17					8,002,00 Bs.	5,298,00 Bs.

Figura 3. 58. Registro de Pago de Alquiler

Fuente: (Elaboración Propia)

Nro.	Tipo Pago	fecha del Pago	Mes	Año	Cuota Bs.	Saldo Deudor Bs.
1	Min. Def. Planilla	30/06/2021	Junio	2021	700,00	12.600,00
2	Min. Def. Planilla	31/07/2021	Julio	2021	700,00	11.900,00
3	Min. Def. Planilla	31/05/2022	Mayo	2022	700,00	11.200,00
4	Depósito Bancario	31/08/2022	Junio	2022	500,00	10.700,00
5	Depósito Bancario	31/08/2022	Agosto	2022	500,00	10.200,00

Figura 3.59. Registro de Pago de Alquiler

Fuente: (Elaboración Propia)

3.3.4.3. Módulo Reporte de Moras (ID BACKLOG 3)

En base a los requisitos e historia de usuario, el Administrador de las Viviendas requiere de un módulo donde puede realizar el registro de la nueva Vivienda.

a) Módulo Moras Vigentes – Análisis

El módulo se encarga de reportar el incumplimiento de pago de alquileres por meses y poder exportar dicha información.

b) Módulo Moras Vigentes – Desarrollo

A continuación, se muestra el módulo de Afiliación de Vivienda. Ver figuras 3.60

MORAS VIGENTES

Mes * AÑO *
NOVIEMBRE 2021

DESCUENTOS MENSUALES

Página: (1 de 1)

Nro.	Cod. Descuento	CM	Locatario	Vivienda	Regional	Monto Adeudado en Bs.	Saldo Deudor	fecha Mora	Descuento Bs.
1	7	31208018	SOF.INL.CHIF MACHACA FAVIO	VIVIENDAS	LA PAZ	13.300,00	3.798,00	Noviembre-2	700,00
2	28	11003009	M.Y. DEMA, KELCA CERVANTES MIGUEL	DEPARTAMEN	LA PAZ	14.000,00	5.600,00	Noviembre-2	700,00
2									

Página: (1 de 1)

Figura 3. 60. Reporte de Moras
Fuente: (Elaboración Propia)

3.4. Post-Game

Una vez concluidas las iteraciones de los sprints se procedió a realizar las actividades de pruebas de manera global en integra a todo el sistema.

3.4.1. Pruebas de Software

Durante el proceso de desarrollo del sistema se aplicaron pruebas unitarias del sistema, a la codificación y la refacción del código construido.

En las tablas 4.1 al 4.13, se muestran las pruebas de cada sprint respectivamente comenzando del segundo sprint, nótese que el primer Sprint cuenta con una explicación de la evaluación de Sprint ya que no fue un Sprint de desarrollo.

➤ Sprint 1

La evaluación del Sprint 1 se da dentro de la revisión con los requerimientos del sistema también se evaluó antes, durante y después de la reunión de presentación de este Sprint, para poder tener una retro alimentación constante

y así tener una base sólida para afrontar los siguientes Sprint que se desarrollan.

Tabla 3.38. Prueba de desarrollo sprint 2 Módulo afiliación

PRUEBA DE DESARROLLO	
Código de Prueba: 1	Nro. Historia de Usuario:1
Tipo de Prueba: Prueba exhaustiva	Módulo afiliación
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 1, correspondiente a la afiliación, registro de un nuevo locatario realizado por el administrador y Auxiliar.	
Evaluación: <ul style="list-style-type: none"> • Antes de realizar el registro se tiene que estar seguro que no se encuentra registrado el locatario. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none"> • Cuando no se registraba todos los datos del formulario, el proceso guardaba los datos incompletos. • Se podía registrar varias veces a un locatario. Corrección ante los casos de prueba. <ul style="list-style-type: none"> • Se validaron algunos campos como requeridos para realizar el registro. • Se realizó los ajustes para el registro del nuevo locatario. 	

Fuente: (Elaboración Propia)

Tabla 3.39. Prueba de desarrollo sprint 2 Módulo afiliación de vivienda

PRUEBA DE DESARROLLO	
Código de Prueba: 2	Nro. Historia de Usuario:2
Tipo de Prueba: Prueba exhaustiva	Módulo afiliación de vivienda
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 2, correspondiente a la afiliación, de una nueva vivienda realizada por el administrador y Auxiliar.	
Evaluación: <ul style="list-style-type: none"> • Cuando no se registraba todos los datos del formulario, el proceso guardaba los datos incompletos. • Se podía registrar varias veces una vivienda. Corrección ante los casos de prueba. <ul style="list-style-type: none"> • Se validaron algunos campos como requeridos para realizar el registro. • Se realizó los ajustes para el registro de la nueva vivienda. 	

Fuente: (Elaboración Propia)

Tabla 3.40. Prueba de desarrollo sprint 2 Módulo reporte de viviendas

PRUEBA DE DESARROLLO	
Código de Prueba: 3	Nro. Historia de Usuario:3
Tipo de Prueba: Prueba exhaustiva	Módulo reporte de viviendas
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 3, correspondiente a la obtención de información de los estados de las viviendas.	
Evaluación: <ul style="list-style-type: none">• Despliegue de todas las viviendas libres y alquiladas. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• Muestra las viviendas libres y alquiladas.• Problema al no haber validado algunos campos numéricos. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se creó la opción de búsqueda de viviendas por estado disponibles y alquiladas.• Validar los campos numéricos.	

Fuente: (Elaboración Propia)

Tabla 3.41. Prueba de desarrollo sprint 2 Módulo consulta alquiler

PRUEBA DE DESARROLLO	
Código de Prueba: 4	Nro. Historia de Usuario:4
Tipo de Prueba: Prueba exhaustiva	Módulo consulta alquiler
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 4, correspondiente a la búsqueda del locatario, realizado por el Oficial de Crédito.	
Evaluación: <ul style="list-style-type: none">• Selecciona opción de búsqueda (CI, Apellidos y Carnet Militar). Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• No se tiene un límite de despliegue de registros encontrados cuando estos sean en gran cantidad. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se realizó los ajustes en el sistema para que no exista despliegue de información.	

Fuente: (Elaboración Propia)

Tabla 3.42. Prueba de desarrollo sprint 2 Módulo historial de alquiler

PRUEBA DE DESARROLLO	
Código de Prueba: 5	Nro. Historia de Usuario:5
Tipo de Prueba: Prueba exhaustiva	Módulo historial de alquiler
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 5, correspondiente al historial de alquileres.	
Evaluación: <ul style="list-style-type: none">• Selecciona el tipo de búsqueda de locatario.• Despliega la información de las viviendas Asignadas e históricas. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• El sistema no muestra el historial de viviendas asignadas• No se visualiza los datos del locatario. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se validó la búsqueda de las viviendas asignadas, cuando se realice el proceso por segunda vez muestre los datos correctos.• Se realizó los ajustes en el sistema para que no exista errores en la búsqueda de locatario.	

Fuente: (Elaboración Propia)

Tabla 3.43. Prueba de desarrollo sprint 2 Módulo asignación de vivienda

PRUEBA DE DESARROLLO	
Código de Prueba: 6	Nro. Historia de Usuario:6
Tipo de Prueba: Prueba exhaustiva	Módulo asignación de vivienda
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 6, correspondiente al registro de la solicitud y aprobación realizado por el administrador.	
Evaluación: <ul style="list-style-type: none">• Ingreso de datos correspondientes dentro de la plataforma.• Selecciona la vivienda disponible.• Genera el plan de pagos.• Calcula la deuda total. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• El sistema no muestra las viviendas disponibles.• No genera el plan de pagos según lo requerido.• Problema al no haber validado algunos campos numéricos. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se validó la búsqueda de las viviendas disponibles, cuando se realice el proceso por segunda vez muestre los datos correctos.• Se añadió un calendario, con el cual podrá determinar la primera fecha de amortización.• Se realizó los ajustes en el sistema para que no exista errores en el cálculo de la deuda total.• Validar los campos numéricos.	

Fuente: (Elaboración Propia)

Tabla 3.44. Prueba de desarrollo sprint 2 Módulo extracto de pago

PRUEBA DE DESARROLLO	
Código de Prueba: 7	Nro. Historia de Usuario:7
Tipo de Prueba: Prueba exhaustiva	Módulo extracto de pago
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 7, correspondiente al extracto de pago realizado por el el administrador, Auxiliar.	
Evaluación: <ul style="list-style-type: none">• Despliega todos los préstamos del locatario, vigentes e históricos.• Despliega todos sus pagos. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• Problema al no mostrar un resumen de sus préstamos.• No se visualiza los datos de la vivienda asignada.• No se visualiza los pagos realizados. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se proporcionó un resumen de vivienda.• Se corrigió la visualización de las viviendas asignadas.	

Fuente: (Elaboración Propia)

Tabla 3.45. Prueba de desarrollo sprint 3 Módulo generar descuentos mensuales

PRUEBA DE DESARROLLO	
Código de Prueba: 8	Nro. Historia de Usuario:8
Tipo de Prueba: Prueba exhaustiva	Módulo generar los descuentos mensuales
Descripción de la prueba: Prueba aplicada a la Historia de Usuario 8, correspondiente a la generación de los descuentos mensuales realizado por el el Auxiliar.	
Evaluación: <ul style="list-style-type: none">• Despliegue de locatarios que deberán ser descontados de sus haberes por alquiler. Con varios casos de pruebas se encontraron los siguientes errores. <ul style="list-style-type: none">• También se adicionan los alquileres cancelados. Corrección ante los casos de prueba. <ul style="list-style-type: none">• Se corrigió el proceso para que solamente se envíen las deudas pendientes.	

Fuente: (Elaboración Propia)

Tabla 3.46. Prueba de desarrollo sprint 3 Módulo registro masivo de amortizaciones

PRUEBA DE DESARROLLO	
Código de Prueba: 9	Nro. Historia de Usuario:9
Tipo de Prueba: Prueba exhaustiva	Módulo registro masivo de amortizaciones
<p>Descripción de la prueba: Prueba aplicada a la Historia de Usuario 9, correspondiente al registro masivo de amortizaciones realizado por el el Auxiliar.</p>	
<p>Evaluación:</p> <ul style="list-style-type: none"> • Importación de la lista de los descuentos. • Despliegue de los locatarios que han sido descontados de sus haberes por alquiler. <p>Con varios casos de pruebas se encontraron los siguientes errores.</p> <ul style="list-style-type: none"> • Problema al momento de registrar, no muestra los descuentos no registrados. <p>Corrección ante los casos de prueba.</p> <ul style="list-style-type: none"> • Se validó el proceso para que exista la información de las cuotas no pagadas. 	

Fuente: (Elaboración Propia)

Tabla 3.47. Prueba de desarrollo sprint 3 Módulo registro de amortización

PRUEBA DE DESARROLLO	
Código de Prueba: 9	Nro. Historia de Usuario:9
Tipo de Prueba: Prueba exhaustiva	Módulo registro de amortización
<p>Descripción de la prueba: Prueba aplicada a la Historia de Usuario 9, correspondiente al registro de amortización individual realizado por el el Auxiliar.</p>	
<p>Evaluación:</p> <ul style="list-style-type: none"> • Ingresar datos del depósito. <p>Con varios casos de pruebas se encontraron los siguientes errores.</p> <ul style="list-style-type: none"> • Problema con el cálculo del nuevo saldo. • Los campos numéricos aceptan cualquier carácter. <p>Corrección ante los casos de prueba.</p> <ul style="list-style-type: none"> • Se volvió a programar el proceso de cálculo. • Se validos los campos numéricos. 	

Fuente: (Elaboración Propia)

CONTROL DE CALIDAD Y SEGURIDAD

4.1. Introducción

Para establecer la calidad del sistema, primeramente, se trabaja con la prueba exhaustiva de cada Sprint, prueba que permite el desarrollador notar los fallos y omisiones.

El control de calidad del sistema se refiere a una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del ciclo de desarrollo, esto asegura que cada producto cumpla con los requisitos que se designan.

Dentro de la ingeniería de software se cuenta con modelos y normas de calidad, que nos describen los factores de calidad a evaluar en los productos de software, el modelo que se aplica para el presente proyecto de grado es la norma ISO/IEC 25010.

4.2. Factores de Calidad ISO/IEC 25010

➤ Adecuación Funcional

La adecuación funcional es la capacidad del software de proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en ocasiones específicas, este atributo del sistema no puede medirse de forma directa, por esa razón para el cálculo de la adecuación funcional utilizaremos la métrica de punto función, para esto se debe determinar cinco características de dominios de información. Los valores de información se definen de la siguiente forma:

- **Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían de diferenciar de las peticiones las cuales se cuentan de forma separada.

- **Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto las salidas se refieren a informes, pantallas, mensajes de error, etc.
- **Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- **Numero de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser parte de una base de datos o un archivo independiente).
- **Numero de interfaces externas.** Se cuenta todas las interfaces legibles por la máquina que se utilizan para transmitir la información a otro sistema.

Para calcular los puntos función se usó la siguiente formula:

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

Donde:

PF: Medida de la adecuación funcional.

Cuenta Total: Es la suma de los siguientes datos: N° de entradas, N° de salidas, N° de peticiones, N° de archivos y N° de interfaces externas.

0.65: Confiabilidad del proyecto, varía del 1% al 100% (0 a 1)

0.01: Error mínimo aceptable de complejidad.

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

Tomando en cuenta estos pasos, inicialmente definimos los siguientes puntos:

Número de Entradas de Usuario

Es el número de los momentos en que el usuario va a introducir información al SI, se puede observar en la Tabla 4.1, las entradas de usuario que tiene el sistema.

Tabla 4.1. Entradas de usuario

Nro.	Entradas de Usuarios
1	Registro de usuario
2	Registro de locatario
3	Registro de vivienda
4	Registro de asignación de vivienda
5	Registro de pago de alquiler por depósito
6	Registro de plan de pagos
7	Registro de los descuentos Min. Def.

Fuente: (Elaboración Propia)

Número de Salidas de Usuario

Es la información elaborada por el sistema, pueden ser los reportes y/o cualquier formato de salida ya sea impreso o en pantalla que son emitidas al usuario, en la Tabla 4.2 se puede observar las salidas de usuario.

Tabla 4.2. Salidas de usuario

Nro.	Salidas de Usuario
1	Reporte de viviendas
2	Reporte de pagos registrados
3	Reporte de saldos pendientes
4	Reporte de alquiler por locatario
5	Reporte extracto de pago
6	Reporte de los descuentos mensuales

Fuente: (Elaboración Propia)

Número de Peticiones de Usuario

Es una entrada interactiva que produce la generación de alguna respuesta del software inmediata, es decir, que se espera que el programa responda al usuario, llámense consultas y similares.

Tabla 4.3. Peticiones de usuario

Nro. Peticiones de usuario	
1	Autenticación de usuario
2	Lista de usuarios
3	Edición de usuarios
4	Lista de viviendas
5	Edición de viviendas
6	Lista de asignación de viviendas
7	Listado de pagos
8	Edición de pago
9	Listado de los descuentos
10	Edición de los descuentos

Fuente: (Elaboración Propia)

Número de Archivos

Es la cantidad de archivos necesarios para que el sistema funcione, es decir un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente. Se puede observar el número de archivo en la Tabla 4.4.

Tabla 4.4. Archivos del sistema

Nro. Archivos	
1	Locatario
2	Descuentos mensuales
3	Viviendas
4	Alquiler
5	Pagos

Fuente: (Elaboración Propia)

Número de Interfaces Externas

Son todas las interfaces externas legibles que sirven para la computadora y que permiten la transmisión de la información, se puede observar las interfaces externas en la Tabla 4.5

Tabla 4.5. Interfaces externas

Nro.	Archivos
1	Internet
2	Intranet

Fuente: (Elaboración Propia)

Factor de ponderación

Una vez que los obtenidos los datos, estos son colocados en la tabla y se elige una ponderación, que representa el grado de complejidad que se prevé para cada una de las opciones, ver la Tabla 4.6.

Tabla 4.6. Factores de ponderación

Parámetro de medida	Cantidad	Factor de ponderación			Total
		Simple	Medio	Complejo	
Nº de entradas de usuario	7	*	4		= 28
Nº de salidas de usuario	6	*	5		= 30
Nº de peticiones de usuario	10	*	10		= 100
Nº de archivos	5	*	4		= 20
Nº de interfaces externas	2	*	7		= 14
Cuenta Total					192

Fuente: (Elaboración Propia)

La cuenta total de los puntos de función obtenidos se debe ajustar en función a las características ambientales del sistema. Los valores de ajuste de complejidad ***Fi*** donde ***i*** puede variar de **1** hasta **14** los valores de ajuste de complejidad basados en las respuestas a las preguntas formuladas de la siguiente tabla:

Tabla 4.7. Valores de Ajuste de Complejidad

N°	Facto de ajuste de valor	Valor
1	¿El sistema requiere respaldo y recuperación confiables?	3
2	¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?	3
3	¿Existen funciones de procesamiento distribuidas?	0
4	¿El desempeño es crucial?	3
5	¿El sistema correrá en un entorno operativo existente enormemente utilizado?	5
6	¿El sistema requiere entrada de datos en línea?	3
7	¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?	3
8	¿Los ALI se actualizan en línea?	3
9	¿Las entradas, salidas, archivos o consultas son complejos?	3
10	¿El procesamiento interno es complejo?	2
11	¿El código se diseña para ser reutilizable?	4
12	¿La conversión y la instalación se incluyen en el diseño?	3
13	¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?	4
14	¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?	3
Total ΣFi		42

Fuente: (Elaboración Propia)

Una vez que se consiguió los valores correspondientes a las variables de la fórmula de los puntos función se procedió a realizar el cálculo del mismo.

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

$$PF = 192 * (0.65 + 0.01 * 42)$$

$$PF = 192 * (1.07)$$

$$PF = 205.44$$

Para comparar los puntos función con su valor máximo, se calculó los puntos función con valores de ajuste de complejidad al máximo que es un total de 70:

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

$$PF = 192 * (0.65 + 0.01 * 70)$$

$$PF = 192 * (1.35)$$

$$PF = 259.2$$

Con máximos valores de ajuste a la complejidad se tiene que la funcionalidad real es:

$$\text{Funcionalidad:} = (205.44 / 259.2) * 100$$

$$\text{Funcionalidad:} = 0.79 * 100$$

$$\text{Funcionalidad:} = 79.25 \%$$

Este resultado quiere decir que el “SISTEMA DE INFORMACIÓN PARA LA ASIGNACIÓN Y CONTROL DE VIVIENDAS FUNCIONALES” CASO: DIRECCIÓN GENERAL DE DESARROLLO Y BIENESTAR SOCIAL (DIGEDEBIS)”, satisface las necesidades explícitas e implícitas en un 79.25%, esto quiere decir que el sistema tiene un 79% de probabilidad que funcione sin riesgos a fallos de operatividad constante y en un 21% de que el sistema colapse.

➤ **Fiabilidad**

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas, por cierto, tiempo. Para este punto se hizo el análisis del nivel de confiabilidad del

sistema, para lo cual primeramente se considera la confiabilidad de cada módulo o subsistema de forma independiente.

Para calcular la confiabilidad de cada módulo se usó de la formula

$$R(t) = e^{-\lambda t}$$

Donde:

$R(t)$: Confiabilidad de un componente o subsistema t.

λ : Tasa de constantes de fallo = N° de fallas de acceso/N° total de accesos al sistema).

t: Periodo de operación de tiempo.

$e^{-\lambda t}$: Probabilidad de falla de un componente o subsistema en el tiempo t.

Luego de realizar pruebas de cada módulo en un tiempo de 4 horas continuas se logró llenar la siguiente tabla:

Tabla 4.8. Valores de Ajuste de Complejidad

N°	Modulo		t	R(t)
1	Módulo de Usuarios y Seguridad	0.015	4hrs	0.94
2	Módulo de Vivienda	0.028	4hrs	0.89
3	Módulo de asignación de Vivienda	0.022	4hrs	0.91
4	Modulo control de descuentos	0.015	4hrs	0.94

Fuente: (Elaboración Propia)

Para hallar la confiabilidad del sistema en su conjunto se tomó en cuenta dos casos, cuando uno de los componentes falla, entonces el sistema en conjunto

falla sistema en serie y cuando todos los componentes fallan en su conjunto sistema en paralelo.

Es por eso que la confiabilidad del sistema estaría dada por la fórmula:

$$Conf = R_s * R_p.$$

Donde:

$$R_p = R_1 = 0.95$$

$$R_s = 1 - (1 - R_2)(1 - R_3)(1 - R_4)(1 - R_5)$$

$$R_s = 1 - (1 - 0.89)(1 - 0.91)(1 - 0.94)(1 - 0.97)$$

$$R_s = 0.99998218$$

Por lo tanto, la confiabilidad del sistema está dada por:

$$Conf = 0.99998218 * 0.95 = 0.94 = 94\%$$

De donde concluimos que la confiabilidad del sistema es del 94%, resultado que implica que existe un 6% de probabilidad de que el sistema pueda fallar cuando se exceda un determinado tiempo de uso continuo, debido a que puede existir fallas en la conexión al sistema, errores en la entrada de datos, uso incorrecto del sistema por parte del usuario, problemas en el servidor de base de datos o en el servidor de páginas Web, etc. Este margen de error también puede deberse a fallas en los dispositivos periféricos como ser impresoras, o la conexión de la red.

➤ Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado de forma fácil y atractiva. Para determinar el porcentaje de la usabilidad del sistema se optó por realizar una encuesta a 8 personas. La siguiente tabla nos muestra los resultados de la encuesta que se realizó:

Tabla 4.9. Evaluación de usabilidad

N°	Pregunta	Respuestas		% de SI
		SI	NO	
1	¿Aprendió a usar rápido el sistema?	8	2	80
2	¿Las pantallas que vio fueron de su agrado?	9	1	90
3	¿Las pantallas que vio fueron fáciles de comprender?	10	0	100
4	¿El sistema responde rápido a sus solicitudes?	8	2	80
5	¿El sistema le facilita el trabajo?	10	0	100
6	¿El sistema reduce su tiempo de trabajo?	10	0	100
7	¿Es fácil navegar por las distintas opciones?	10	0	100
8	¿Las operaciones que se realizan no son complicadas?	9	1	90
9	¿El sistema le proporciono las respuestas requeridas?	9	1	90
10	¿El sistema no presento errores?	9	1	90
Promedio				92

Fuente: (Elaboración Propia)

Considerando las respuestas afirmativas de los cuestionarios concluimos que el Sistema Web de Control y Asignación de viviendas, tiene un grado de usabilidad del 92%, resultado que implica que existe un 8% de los usuarios no logra comprender ni usar el software en condiciones específicas de uso, y que para 92% de los usuarios el software es atractivo y de fácil comprensión llegando a satisfacer sus expectativas y ayudando a un mejor desempeño en las tareas que estos realizan.

➤ **Mantenibilidad**

La Mantenibilidad es la calidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales, para poder medir la calidad del mantenimiento del sistema utilizaremos el índice de madurez del software (IMS), que indica la estabilidad de un producto de software. El índice de madurez del software se calcula con la siguiente fórmula:

$$IMS = [M_t - (F_a + F_b + F_c)] / M_t$$

Donde:

Mt: Número de módulos en la versión actual

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

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

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

En el sistema se obtuvieron los siguientes valores como muestra la Tabla 4.10, para la información requerida por el IMS.

Tabla 4.10. Valores requerida por el IMS

Información	Valor
Mt	5
Fa	0
Fb	0
Fc	0

Fuente: (Elaboración Propia)

Ahora calculamos el IMS, usando los valores obtenidos:

$$IMS = [5 - (0+0+0)] / 5$$

$$IMS = 5/5 = 100\%$$

Con ese resultado se concluyó que el Sistema de Control y Asignación de vivienda es estable en un 100%, que es la facilidad de mantenimiento en la corrección de fallas o errores del sistema, y su adaptación a nuevas necesidades.

➤ **Portabilidad**

La portabilidad es la capacidad que tiene el software para ser trasladado de un entorno a otro. Para poder medir la portabilidad del sistema usaremos la siguiente formula que indica el grado de portabilidad que tiene un software.

$$GP = 1 - (ET / ER)$$

Donde:

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

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

Si $GP > 0$, la portabilidad es más rentable que el re-desarrollo

Si $GP = 1$, la portabilidad es perfecta

Si $GP < 0$, el re-desarrollo es más rentable que la portabilidad.

Para llevar el sistema web a otro entorno se necesita una memoria extraíble de 8gb o más capacidad, para crear el sistema en el entorno residente se necesita inicialmente 1 servidor con sistema operativo este puede ser cualquiera (Windows, Linux o Mac OS en sus diferentes versiones), y servidor Apache, NetBeans, el gestor de base de datos MySQL los cuales deben estar instalados en los servidores y un navegador web.

Con esta información requerida por la fórmula, se procede a calcular el grado de portabilidad:

$$GP = 1 - (1/5)$$

$$GP = 1 - 0.20 = 0.8$$

GP = 80%

Por lo que se concluye que el sistema web tiene un grado de portabilidad del 80%.

➤ **Calidad Global del Sistema Web**

Una vez calculado los porcentajes de los diferentes atributos que el sistema web tiene según lo propuesto por el estándar de calidad ISO 9126, se procedió a calcular la calidad global del sistema web, la cual se visualiza en la siguiente tabla:

Tabla 4.11. Calidad global del sistema

Atributo	Valor(%)
funcionalidad	86
Confiabilidad	94
Usabilidad	92
Mantenibilidad	100
Portabilidad	80
Calidad Global	90.4

Fuente: (Elaboración Propia)

Con el resultado obtenido se concluyó que la calidad global del sistema de información para la asignación y control de viviendas funcionales” caso: Dirección General de Desarrollo y Bienestar Social (DIGEDEBIS), es del 90.4%.

EVALUACIÓN DE COSTO Y BENEFICIO

5.1. Introducción

COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones que se detallan a continuación.

$$E = a(KLDC)^b ; \text{Persona - Mes}$$

Ecuación 4

$$D = c(E)^d ; \text{Meses}$$

Ecuación 5

$$P = E/D ; \text{Personas}$$

Ecuación 6

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código distribuidas en miles.

A la vez cada modelo se subdivide en modos, los mismos son:

- Modo orgánico: Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- Modo semi – libre o semi acoplado: Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo

puede incluir una mezcla de personas experimentadas y no experimentadas.

- Modo rígido o empotrado: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

La tabla 5.1 muestra los coeficientes del proyecto de software de acuerdo a los tres modos expuestos anteriormente.

Tabla 5.1. Coeficientes a, b, c y d (COCOMO II)

Proyecto de Software	a	b	c	d
Orgánico	2.4	1.05	1.05	0.38
Semi-Acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: (Pressman, 2005)

5.2. Costo Del Sistema

El costo del Sistema se lo planteara en tres partes: desarrollo de software, implementación y elaboración del proyecto.

5.2.1. Costo De Desarrollo Del Software

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustado valor ya encontrado en el capítulo anterior. Recordando es el siguiente:

$$PF = 220$$

Este resultado se debe convertir a KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla 5.2

a) Conversión De Los Puntos A Kdlc

Ahora se debe convertir los Puntos de Función a miles de líneas de código. Tomar en Cuenta la siguiente tabla.

Tabla 5.2. Conversión de Puntos de Función a KDLC

LENGUAJE	NIVEL	FACTOR LDC / PF
Java	5	120
Xhtml	6	107
JSF	7.00	46
Primefaces	9.00	36
Netbenas	11.00	29
MySql	9.50	34

Fuente: (Pressman, 2002)

Realizando algunos cálculos y escogiendo el lenguaje de programación en este caso Netbenas.

$$LDC = PF * \text{Factor LDC / PF}$$

$$LDC = 220 * 29$$

$$LDC = 6380$$

$$KLDC = (6380 / 1000) = 6.38$$

b) Aplicación De Las Formulas Básicas De Esfuerzo, Tiempo Calendario Y Personal Requerido.

El proyecto de software orgánico, los proyectos intermedios (en tamaño y complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidos, tal es el caso del Sistema Web de Gestión Académica desarrollado y utilizando los valores de la Tabla 5.1.

Reemplazando los valores obtenidos en las ecuaciones 5.1 y 5.2

$$E = 2.4 * (6.38)1.05 \Rightarrow 16.75$$

$$D = 2.5 * (16.75)0.35 \Rightarrow 6.70$$

Aplicando en la ecuación 5.3 para el cálculo del personal requerido en este caso el número de programadores para el desarrollo es:

$$P = (16.75 / 6.70) = 2.56 \cong 3$$

El salario de un programador puede oscilar entre los \$US 300, cifra que será tomada para la estimación siguiente por lo tanto se tiene:

Costo del Software Desarrollado = Numero de Programadores * Salario de un Programador.

Costo del Software Desarrollado por Persona = 3 * 300 =>900 \$US.

Costo total del Software Desarrollado = 900* 7 =>6300 \$US.

5.2.2. Costo De Implantación

La Unidad Educativa cuenta con equipos de computación y más adelante el servicio de internet. Por lo tanto el único costo de implementación que se tendrá será la configuración de la parte del servidor. El mismo tendrá un costo de 100 \$us.

5.2.3. Costo De Elaboración Del Proyecto

Los costos de elaboración del proyecto se refieren principalmente a los gastos que se realizan a lo largo de las diferentes fases de la metodología Ágil SCRUM. Estas las podemos ver expresadas en la tabla 5.3

Tabla 5.3. Costo de elaboración del proyecto

DETALLE	IMPORTE EN (\$US.)
Análisis y diseño del proyecto	200
Material de escritorio	30
Internet	50
Otros	20
Total	300

Fuente: (Elaboración propia)

5.2.4. Costo Total Del Software

El costo total del software se lo obtiene de la sumatoria del costo de: desarrollo, Implementación y elaboración del proyecto. La tabla 5.4 expresa estos resultados.

Tabla 5.4. Costo total del software

DETALLE	IMPORTE EN (\$US.)
Costo de desarrollo	6300
Costo de implementación	100
Costo de elaboración del proyecto	300
Total	6700

Fuente: (Elaboración propia)

5.3. Beneficio

El VAN y el TIR son dos herramientas financieras procedentes de las matemáticas financieras que nos permiten evaluar la rentabilidad de un proyecto de inversión, entendiéndose por proyecto de inversión no solo como la creación de un nuevo negocio, sino también, como inversiones que podemos hacer en un negocio en marcha, tales como el desarrollo de un nuevo producto, la adquisición de nueva maquinaria, el ingreso en un nuevo rubro de negocio, etc.

5.3.1. Valor Actual Neto

El VAN o valor actual neto es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto. La fórmula que utilizaremos para hallar el valor actual neto será:

$$VAN = \sum (Ganancias/(1 + k)^n) - \sum (Costos/(1 + k)^n)$$

Dónde:

VAN: Valor Actual Neto

Ganancias: Ingreso de flujo anual

Costos: Salidas de flujo anual

n: Numero de periodo

k: Tasa de descuento o tasa de interés al préstamo

Los gastos y ganancias que se estiman en un lapso de 4 años los mostramos en la Tabla 5.5, para este caso en particular utilizamos una de descuento del 12% ya que es la tasa actual de interés del préstamo en las entidades financieras.

Tabla 5.5. Calculo del VAN

Año	Costo	Ganancia	Costo/(1+i)	Guanacias/(1+i)	Resultado
1	6700	0	5982	0	0
2	1100	2000	876.2	1594.4	717.5
3	620	3500	441.3	2491.2	2049.9
4	300	5000	190.7	3177.6	2986.9
Σ	7670	10500	6553.5	7263.2	0
$VAN = \sum \frac{Ganancias}{(1 + 0.12)^n} - \sum \frac{Costos}{(1 + 0.12)^n}$					709.7

Fuente: (Elaboración propia)

La tabla 5.6 muestra si un proyecto es rentable y de acuerdo a ciertos criterios más el valor del VAN concluiremos si es rentable o no.

Tabla 5.6. Criterio de interpretación del VAN

VALOR DEL VAN	INTERPRETACIÓN
VAN > 0	El proyecto es rentable
VAN = 0	El proyecto también es rentable, ya que se incorpora la ganancia de la tasa de interés.
VAN < 0	El proyecto no es rentable.

Fuente: (Elaboración propia)

De aquí concluimos: considerando que el VAN = 709.7 ≈ 709 y siguiendo los criterios de la tabla 5.6 se afirma que nuestro proyecto es rentable ya que 709 es mayor a 0.

5.3.2. Tasa Interna De Retorno

Cuando en la fórmula del VAN el valor de “k” es igual a “0” pasa a llamarse T.I.R. (Tasa Interna de Retorno). La T.I.R. es la rentabilidad que nos proporciona al proyecto.

La ecuación que utilizaremos es la siguiente:

$$TIR = \sum(Ganancias - Costos)/(1 - i)^n$$

Dónde:

TIR: Tasa Interna de Retorno

Ganancias: Flujo de entrada de un periodo

Costos: Flujo de salida de un periodo

i: Tasa de interés al ahorro

n: Numero de periodo

La tabla 5.7 muestra una mejor comprensión de ecuación T.I.R. y expresando los resultados encontrados en las misma.

Tabla 5.7. Cálculo de la tasa interna de retorno

Año	Costo	Ganancias	(Ganancia - Costo)/(1+i)
1	7047	0	- 6420.50
2	1100	2000	1162.20
3	620	3500	4226.10
4	300	5000	7837.30
TIR			6805.10

Fuente: (Elaboración propia)

El proyecto nos dará una rentabilidad de 6805.1 \$us

5.4. Costo / Beneficio

Para calcular el costo beneficio usaremos la tabla 5.5 donde encontramos el total de ganancias y costos.

$$\text{Costo/Beneficio} = \Sigma \text{ Ganancias} / \Sigma \text{ Costos}$$

$$\text{Costo/Beneficio} = 10500 / 7670$$

$$\text{Costo/Beneficio} = 1.4 \text{ \$us}$$

Con este resultado interpretamos de la siguiente manera: por cada dólar invertido en el proyecto de software la institución genera una ganancia de 0.4 \$us. Y de lo cual se concluye que el proyecto es viable.



SEGURIDAD DEL SISTEMA

6.1. Seguridad

La Norma 17799 (o ISO 27002) es un estándar para la seguridad de la información, que comprende diez dominios, los más adecuados para la evaluación de este proyecto son los siguientes:

6.2. Organización de la Seguridad

Deben establecerse adecuadamente las responsabilidades para cada usuario de la organización.

6.3. Clasificación y Control de Activos

Se debe realizar inventarios de información e instalaciones, como ser: Recursos de la información (Base de Datos), software (de aplicación, Sistemas y herramientas de desarrollo).

6.4. Gestión de Operación y de Comunicación

Se debe garantizar el funcionamiento correcto y seguro de las instalaciones de procesamiento de la información.

6.5. Control de Accesos

Consiste en seguir procedimientos formales para controlar la asignación de privilegios, contraseñas de usuario, acceso a la red, restricción de acceso a la información.

6.5.1. Algoritmo Md5

MD5 comienza rellenando el mensaje a una longitud con $448 \bmod 512$, es decir la longitud del mensaje es de 64 bits, el relleno empieza con un 1, seguido de tantos 0 como sean necesarios. La codificación de MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. El

siguiente código de 28 bytes ASCII será tratado con MD5 y veremos su correspondiente salida.

Descripción del algoritmo MD5: Empezamos suponiendo que tenemos un mensaje de “b” bits de entrada, y que nos gustaría encontrar su resumen. Aquí “b” es un valor arbitrario entero no negativo, pero puede ser cero, no tiene por qué se multiplica de ocho, y puede ser muy largo. Imaginemos los bits del mensaje escrito así:

m0 m1 m2 mb-1

Los siguientes cinco pasos son efectuados para calcular el resumen del mensaje:

- Paso 1. Añadiendo bits.- El mensaje será extendido hasta que su longitud en bits sea congruente con $448 \bmod 512$. Esto es, si se le resta 448 a la longitud del mensaje tras este paso, se obtiene un múltiplo de 512. Esta extensión se realiza siempre, la extensión se realiza como sigue: un solo bit “1” se añade al mensaje y después bits “0” se añade hasta que la longitud en bits del mensaje extendido se haga congruente con $448 \bmod 512$.
- Paso 2. Longitud del mensaje.- Un entero de 64 bits que represente la longitud “b” del mensaje, se concatena al resultado del mensaje anterior. En el supuesto que “b” sea mayor que 2^{64} , entonces solo los 64 bits de menor peso que “b” se usaran.
- Paso 3. Inicializar el buffer MD.- Un buffer de cuatro palabras (A, B, C, D) se usa para calcular el resumen del mensaje aquí cada una de las letras A, B, C, D representa un registro de 32 bits. Estos registros se inicializan con los siguientes valores hexadecimales, los bits de mayor peso primero.
- Paso 4. Procesado el mensaje en bloques de 16 palabras.- Primero definimos 4 funciones auxiliares que toman como entrada 3 palabras de 32 bits y su salida es una palabra de 32 bits.

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X, Y, Z) = X + Y + Z$$

$$I(X, Y, Z) = Y + (X \vee \neg Z)$$

En cada posición de cada bit F actúa un condicional: si X entonces Y sino Z. la función F podría haber sido definida usando “+” en lugar de “v” ya que XY y $\neg X$ nunca tendrán unos en la misma posición del bit.

- Paso 5. Salida.- El resumen del mensaje es la salida producida por A, B, C y D. Se comienza el byte de menor peso de A y se acaba con el byte de mayor peso de D. Aplicación del algoritmo MD5: El algoritmo MD5 se encuentra en PHP3, PHP4 y PHP5 como una función de cifrado tipo hash que acepta una cadena de texto como entrada y devuelve un número de 128 bits. Las ventajas de este algoritmo es la imposibilidad de reconstruir la cadena original a partir del resultado, para la implementación de un método seguro para la autenticación y asignación de niveles de acceso y privilegios.

6.5.2. Desarrollo y Mantenimiento de Sistema

Este dominio comprende la Seguridad de los archivos del Sistema y seguridad de los procesos de desarrollo y soporte.

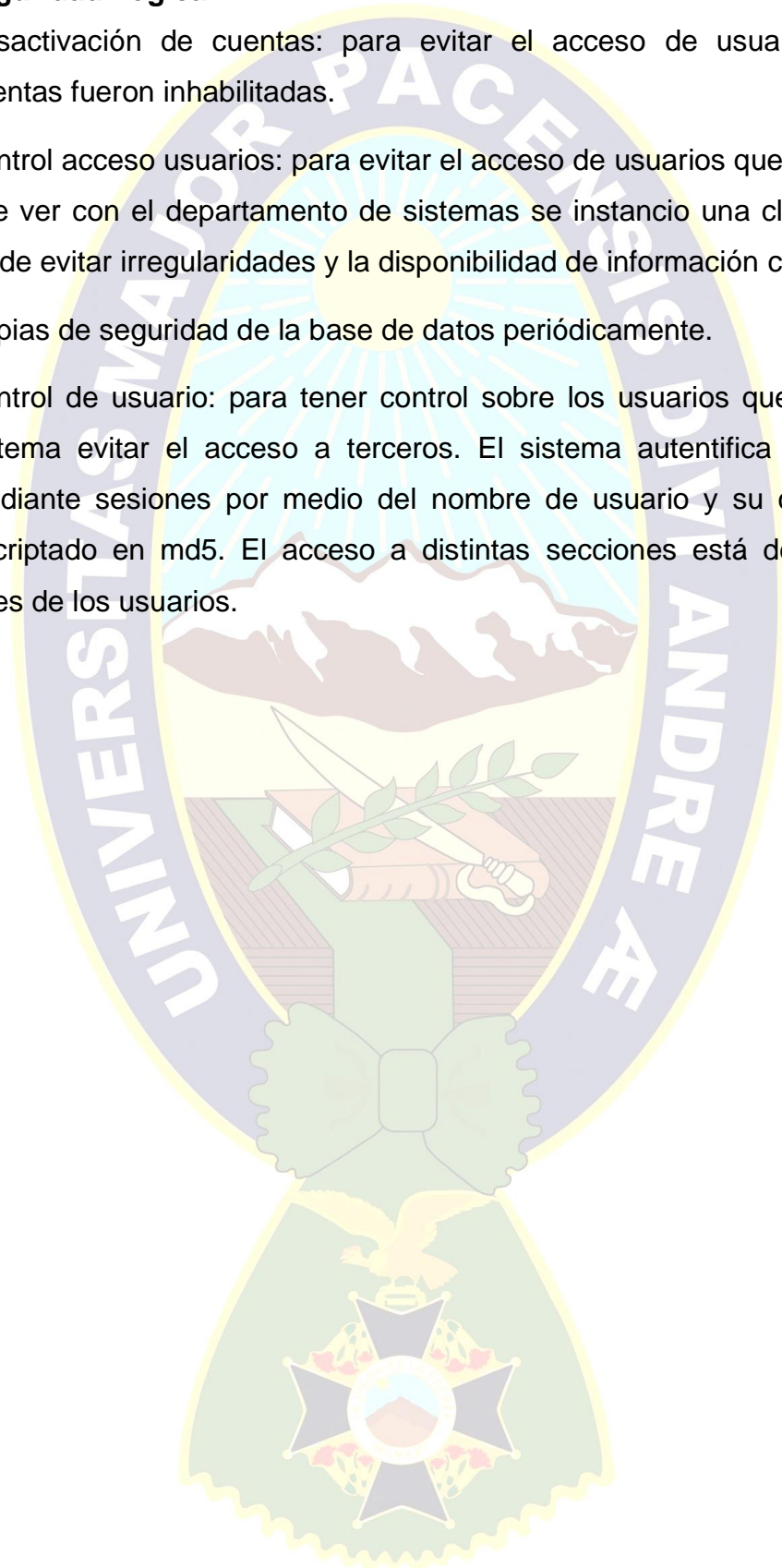
Se aplicara los incisos 1, 4, 5 y 6 de lo norma ISO para garantizar la confidencialidad, integridad y disponibilidad de la información del sistema y de su entorno, considerando los siguientes aspectos:

6.5.3. Seguridad Administrativa

- Solo el administrador del sistema debe asignar cuentas a los usuarios, realizar las copias de respaldo, restaurar el sistema y la base de datos.
- Solo el responsable de la actualización de la base de datos del sistema debe realizar la migración de datos.

6.5.4. Seguridad Lógica

- Desactivación de cuentas: para evitar el acceso de usuarios cuyas cuentas fueron inhabilitadas.
- Control acceso usuarios: para evitar el acceso de usuarios que no tengan que ver con el departamento de sistemas se instancio una clave con el fin de evitar irregularidades y la disponibilidad de información confiable.
- Copias de seguridad de la base de datos periódicamente.
- Control de usuario: para tener control sobre los usuarios que entran al sistema evitar el acceso a terceros. El sistema autentifica al usuario mediante sesiones por medio del nombre de usuario y su contraseña encriptado en md5. El acceso a distintas secciones está definido por roles de los usuarios.



CONCLUSIONES Y RECOMENDACIONES

7.1. Conclusiones

Se desarrolló e implemento el Sistema de Información para la Asignación y Control de Viviendas Funcionales cumpliendo con las exigencias establecidas.

En cuanto a la Administración y control de Usuarios, cada Usuario tiene su propia cuenta para ingresar al Sistema, dando seguridad a cada cuenta de Usuario.

Se automatizó los procesos de asignación de viviendas el cual permite guardar, actualizar los datos. Así también registrar los pagos correspondientes.

Se desarrolló una sección para el envío de los descuentos mensuales ante el Ministerio de Defensa, para lo cual se genera listas de los descuentos.

El Sistema también brinda reporte de los Extractos de pagos por alquileres.

Se desarrolló un módulo que permite obtener reportes de listas de los estados de las viviendas Funcionales como la estimación de ingresos del mes.

También se implementó el módulo de registros de amortizaciones masivas (descuentos remitidos por el ministerio de defensa), como las amortizaciones individuales por depósitos bancarios.

El objetivo principal de este proyecto era automatizar los procesos manuales y brindar información completa y confiable en todo momento, es así que se logró automatizar esos procesos manuales de forma interactiva, tal que el usuario pueda realizar consultas, las cuales se ejecutaran al instante, lo cual ayudara al momento de brindar solución a algún problema en la administración de las viviendas.

7.2. Recomendaciones

El sistema fue desarrollado cumpliendo con requisitos solicitados por los usuarios, pero aún quedan otras funcionalidades que se tiene que

complementar para tener un control absoluto de la información que genera la institución.

Se recomienda realizar lo siguiente:

- Desarrollar el subsistema contable para que así la integración con el sistema esté completa y de esta manera se pueda controlar mejor.
- Desarrollar el subsistema para el seguimiento y control de los activos fijos de las viviendas Funcionales.
- Implementar un módulo para el control y generación de alerta desalojos.



BIBLIOGRAFÍA

Dirección General de Desarrollo y Bienestar Social. (2003). *Manual de organización y funciones*. La Paz: Autor.

Dirección de Relaciones Públicas de la Fuerza Aérea Boliviana. (2005). *Resumen Histórico de las Brigadas Aéreas, Institutos y Unidades de la Aviación Militar*. La Paz: Autor.

Pressman, R. (2003). *Ingeniería del Software un enfoque práctico* (3a Ed.). España: Mc Graw Hill.

Pressman, R. (2005). *Ingeniería del Software un enfoque práctico*. España: Mc Graw Hill.

Java, O. (28 de Junio de 2021). Java. Obtenido de Conozca más Sobre la tecnología Java: <https://www.java.com/es/about/>

AprendeAProgramar.com. (04 de Julio de 2015). AprendeAProgramar.com. Obtenido de Introducción a Java: <http://www.aprendeaprogramar.com/mod/resource/view.php?id=204>

coursehero.com. (04 de mayo de 2021). coursehero.com. Obtenido de Introducción a framework: <https://www.coursehero.com/file/52010709/Frameworkpdf/>

es.wikipedia.org. (04 de mayo de 2021). es.wikipedia.org. Obtenido de Introducción a PrimeFaces: <https://es.wikipedia.org/wiki/PrimeFaces>

maestrosdelweb.com. (04 de mayo de 2021). maestrosdelweb.com. Obtenido de Introducción a Base de Datos: <http://www.maestrosdelweb.com/queson-las-bases-de-datos/>

- ictea.com. (04 de mayo de 2021). ictea.com. Obtenido de MySQL:
<https://www.ictea.com/cs/index.php?rp=/knowledgebase/3500/iQue-es-MySQL.html>
- Trigas, 2008. *Gestión de proyectos informáticos Metodología Scrum*. Virrueta Méndez A., 2010. *Metodologías de Desarrollo de Software*.
- Schmuller, J. (2004). *Aprendiendo UML en 24 Horas*. México: Prentice Hall.
- Pressman, R. (2003). *Ingeniería del Software un enfoque práctico* (3ª Ed.). España: Mc Graw Hill.
- Pressman, R. (2005). *Ingeniería del Software un enfoque práctico*. España: Mc Graw Hill.
- Rueda, J. (2006). *Aplicación de la Metodología RUP*. Guatemala: Universidad de San Carlos.
- ISO25000, (2019). *NORMAS ISO 25000*. Recuperado de <http://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&limitstart=0>
- COCOMO II (2019). *COCOMO II - Constructive Cost Model*. Recuperado de <http://csse.usc.edu/tools/cocomoii.php>
- Palenque, W. (2018). *Plan de negocios para la creación de una empresa de Courier y Delivery en zonas estratégicas de la ciudad de La Paz*. Universidad San Fransisco de Asis, Carrera de Administración de Empresas. La Paz, Bolivia.
- Pérez, A. (2014) "Software móvil de geolocalización para la banca en la ciudad de La Paz" Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales, Carrera Informática. La Paz, Bolivia.

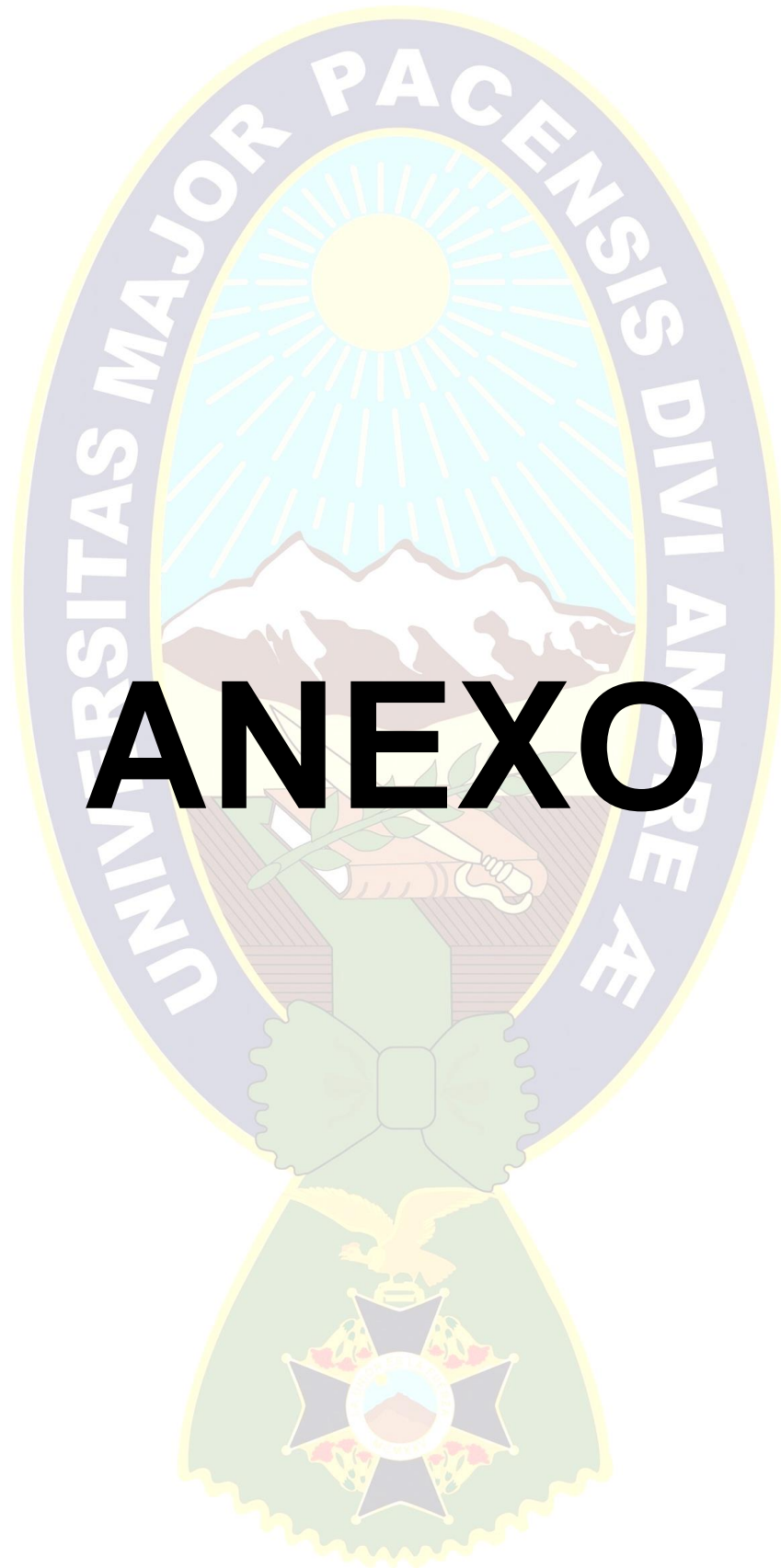
Piattini, M. (2020). Gestión de Requisitos para el Desarrollo de Software. Recuperado 20 de julio 2020, de Virtual.itca.edu website: https://virtual.itca.edu.sv/Mediadores/stis/22___gestin_de_requisitos_para_el_desarrollo_de_software.html

Pressman, R. (2010) Ingeniería de Software un Enfoque Práctico, 7th Edición, México: McGraw-Hill

Sánchez, F, (1999) Medida del Tamaño Funcional de Aplicaciones de Software, Universidad de Castilla – La Mancha, Escuela Superior de Informática de Ciudad Real

Somerville, I. (2011) Ingeniería de Software, 9no Edición, México: Pearson Educación

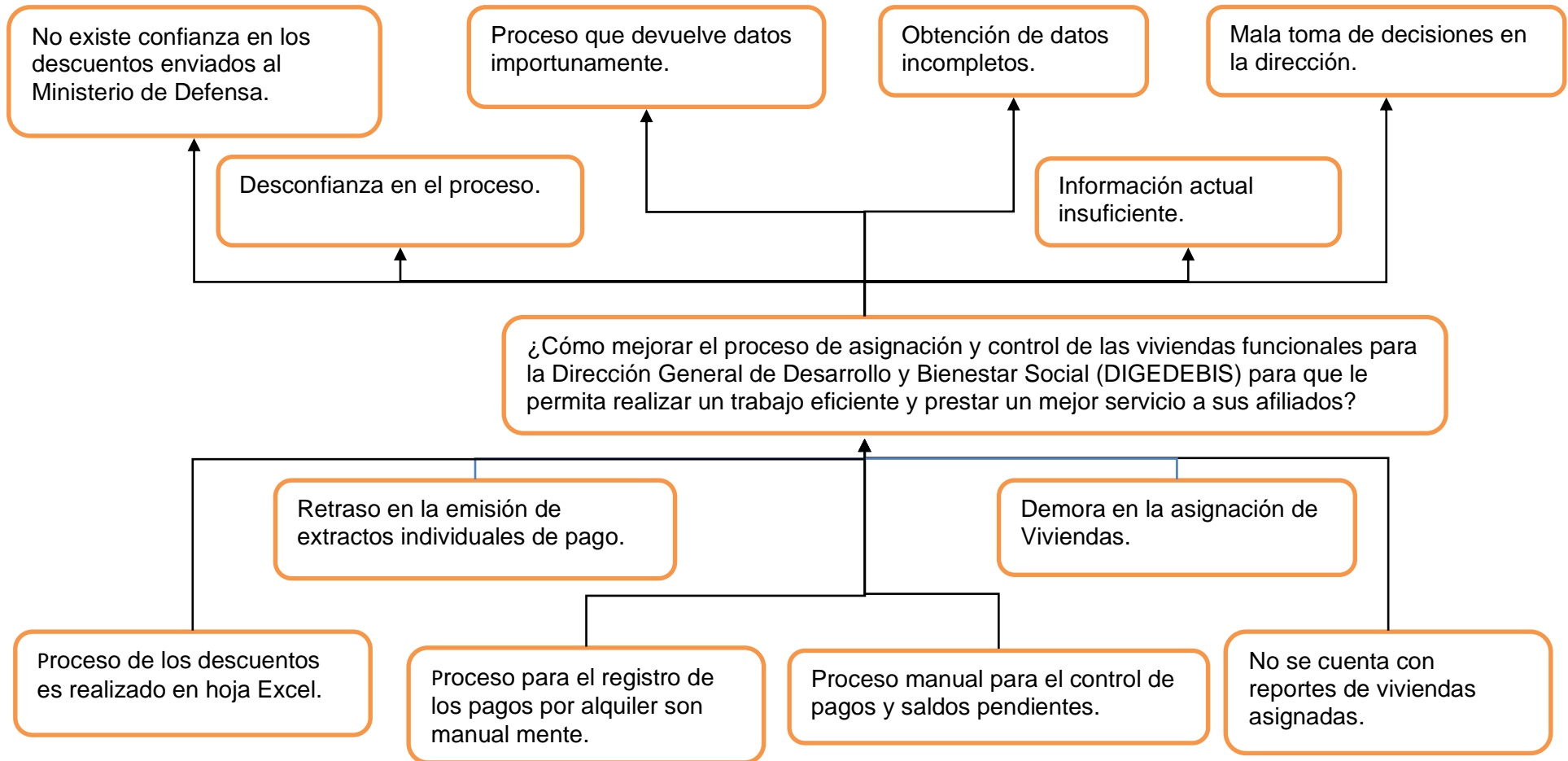
Utecno, (2020). Mezcla de Mercadotecnia Análisis de Coste-Beneficio. Recuperado 15 de septiembre 2020 de Utecno: <https://utecno.files.wordpress.com/2013/09/anacc81lisis-costo-beneficio.pdf>



ANEXO

ANEXO "A"

ÁRBOL DE PROBLEMAS



ANEXO "B"

ÁRBOL DE OBJETIVOS

