

DISEÑO DE MÓDULOS DEL PROCEDIMIENTO PEVReS PARA SU INTEGRACIÓN
A LA PLATAFORMA DE GESTIÓN DE REQUISITOS CHAMÍ

HOSSEN ADLAY BARAJAS PEREA
1.077.477.160

DIRECTOR DEL PROYECTO
ALONSO TORO LAZO, MSc, PhD.

UNIVERSIDAD CATÓLICA DE PEREIRA
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
PEREIRA - RISARALDA
NOVIEMBRE 2022

TABLA DE CONTENIDO

RESUMEN.....	8
ABSTRACT.....	8
1. INTRODUCCIÓN	9
2. PLANTEAMIENTO DEL PROBLEMA.....	11
3. DELIMITACIÓN.....	12
4. JUSTIFICACIÓN.....	13
5. OBJETIVOS	14
5.1. Objetivo General	14
5.2. Objetivos Específicos.....	14
6. MARCO TEÓRICO.....	15
6.1. Software de Calidad	15
6.2. Gestión de Requerimientos	15
6.3. Especificación	16
6.4. Diseño y Arquitectura	16
6.5. Procedimiento para la Especificación y Validación de Requisitos de Software (PEVReS) 16	
6.6. Documento de Especificación de Requerimientos (SRS).....	18
6.7. Lenguaje Unificado de Modelado (UML).....	19
6.8. Sistema Gestor de Requisitos “CHAMÍ”	19
6.9. Metodologías de desarrollo de software.....	20
6.10. Metodología basada en Prototipos	21
7. METODOLOGÍA	22
7.1. Planificación del Proyecto (Cronograma de Actividades)	23
8. DESARROLLO DEL PROYECTO.....	24
8.1. Antecedentes de proyectos Similares	24
8.2. Aplicación de Metodología de Desarrollo.....	28
9. ANÁLISIS DE RESULTADOS	30
9.1. Módulo para la Verificación y Validación de Prototipos	31
9.2. Módulo para la Especificación y Verificación de Casos de Usos	41
9.2. Módulo para Casos de Pruebas	56
9.3. Módulo para Historias de Usuarios	70
9.4. Integración de herramientas UML online al sistema de información CHAMÍ	87

10.	CONCLUSIONES	94
11.	RECOMENDACIONES	95
12.	REFERENCIAS	96
	APÉNDICES	98

LISTA DE ILUSTRACIONES

Ilustración 1. Procedimiento PEVReS [1].....	17
Ilustración 2. Especificación de requerimientos en lenguaje natural [1].....	18
Ilustración 3. Sistema gestor de requisitos “CHAMÍ” [8].....	20
Ilustración 4. Fases de la metodología basada en Prototipos.[12].....	21
Ilustración 5. Metodología del proyecto.	22
Ilustración 6. Sistema Integral de Administración de Requerimientos “SIAR” [13].....	25
Ilustración 7.Rational Doors Next Generation.[15]	25
Ilustración 8.Sistema de Visual Requirements.....	26
Ilustración 9.Sistema Raquest	27
Ilustración 10.Sistema Doc Sheeta.....	28
Ilustración 11. Diagrama de casos de uso de las nuevas funcionalidades para la gestión del sistema Chamí.	30
Ilustración 12. Requerimientos del módulo prototipos implementados en el sistema Chamí.....	32
Ilustración 13. Diagrama de actividad para el módulo de prototipos	33
Ilustración 14. Diagrama de casos de uso para el módulo de prototipos.	34
Ilustración 15. Diagrama de clases módulo prototipos.	37
Ilustración 16. Mockups página de inicio - módulo prototipos.....	38
Ilustración 17. Mockups ver o crear nuevo proyecto - módulo prototipos	38
Ilustración 18.Mockups crear proyecto - módulo prototipos	39
Ilustración 19.Mockups lista de prototipos - módulo prototipos.....	39
Ilustración 20.Mockups pestaña de verificación de prototipos - módulo prototipos.....	40
Ilustración 21.Requerimientos del módulo casos de uso implementados en el sistema Chamí	42
Ilustración 22.Diagrama de actividad consultar CU- módulo casos de uso	43
Ilustración 23.Diagrama de actividad actualizar CU- módulo casos de uso	44
Ilustración 24.Diagrama de actividad eliminar CU- módulo casos de uso	45
Ilustración 25.Diagrama de casos de uso – módulo casos de uso	46
Ilustración 26.Especificación de caso de uso - módulo caso de uso	47
Ilustración 27.Especificación de caso de uso - módulo caso de uso	48
Ilustración 28.Diagrama de clases – módulo casos de uso.....	49
Ilustración 29.Mockups registro de usuario - módulo casos de uso.....	50
Ilustración 30.Mockups inicio de sesión - módulo casos de uso.....	50

Ilustración 31.Mockups dashboard de usuario - módulo casos de uso.....	51
Ilustración 32.Mockups creación de nuevo proyecto- módulo casos de uso	51
Ilustración 33. Mockups dashboard de proyecto - módulo casos de uso	52
Ilustración 34.Mockups formularios de especificación - módulo casos de uso	53
Ilustración 35.Mockups ventana emergente para adicción de imagen - módulo casos de uso.....	54
Ilustración 36.Mockups formularios de verificación - módulo casos de uso	55
Ilustración 37.Requerimientos del módulo casos de pruebas implementados en el sistema Chamí .	57
Ilustración 38.Diagrama de actividad caso de prueba – módulo casos de pruebas	58
Ilustración 39.Diagrama de actividad walkthroughs– módulo casos de pruebas	59
Ilustración 40.Diagrama de caso de uso gestión del módulo – módulo casos de pruebas	60
Ilustración 41.. Diagrama de caso de uso, gestión de casos de pruebas – módulo casos de pruebas	61
Ilustración 42.Diagrama de caso de uso, casos de pruebas – módulo casos de pruebas	61
Ilustración 43.Diagrama de caso de uso, gestión de casos walkthroughs – módulo casos de pruebas	62
Ilustración 44.Diagrama de caso de uso, casos de walkthroughs – módulo casos de pruebas.....	62
Ilustración 45.Diagrama de clase– módulo casos de pruebas	67
Ilustración 46.Mockup Login. - módulo casos de pruebas.....	68
Ilustración 47.Mockup home. - módulo casos de pruebas	68
Ilustración 48.Mockup administrador. - módulo casos de pruebas	69
Ilustración 49.Requerimientos del módulo historias de usuarios implementados en el sistema Chamí	71
Ilustración 50.Diagrama de actividad – módulo historias de usuarios.....	72
Ilustración 51.Diagrama de casos de uso – módulo historias de usuarios.....	73
Ilustración 52.Especificación para verificar historia de usuario - módulo historias de usuarios	74
Ilustración 53.Especificación para consultar el estado - módulo historias de usuarios.....	75
Ilustración 54.Especificación para login - módulo historias de usuarios	76
Ilustración 55.Especificación para buscar historia de usuario - módulo historias de usuarios	77
Ilustración 56.Especificación para crear historia de usuario - módulo historias de usuarios.....	78
Ilustración 57.Especificación para actualizar historia de usuario - módulo historias de usuarios	79
Ilustración 58.Especificación para eliminar historia de usuario - módulo historias de usuarios.....	80
Ilustración 59.Especificación para registrar nuevo usuario - módulo historias de usuarios.....	81
Ilustración 60.Especificación para crear proyecto- módulo historias de usuarios	82
Ilustración 61.Diagrama de clases – módulo historias de usuarios	83

Ilustración 62.Mockup login. - módulo historias de usuarios	84
Ilustración 63.Mockup home. - módulo historias de usuarios.....	84
Ilustración 64.Mockup formulario. - módulo historias de usuarios	85
Ilustración 65.Mockup verificación. - módulo historias de usuarios	85
Ilustración 66.Mockup actualizar. - módulo historias de usuarios.....	86
Ilustración 67.Mockup buscar. - módulo historias de usuarios	86
Ilustración 68.Mockup eliminar. - módulo historias de usuarios	87
Ilustración 69.Apix-Drive Marketplace [16].....	88
Ilustración 70.Atlassian Marketplace[18]	88
Ilustración 71. Sistema JointJS.....	91
Ilustración 72. Sistema JSUML2.	92
Ilustración 73.Sistema MXGRAPH.	93

LISTA DE TABLAS

Tabla 1. Cronograma de actividades.....	23
Tabla 2. Requerimientos para el módulo de prototipos.....	31
Tabla 3.Especificación detallada del primer requisito	35
Tabla 4.Especificación detallada del segundo requisito.....	35
Tabla 5.Especificación detallada del tercer requisito	35
Tabla 6.Especificación detallada del cuarto requisito	36
Tabla 7.Especificación detallada del quinto requisito.....	36
Tabla 8.Especificación detallada del sexto requisito.....	36
Tabla 9. Requerimientos – módulo casos de uso	41
Tabla 10.Requerimientos – módulo casos de pruebas	56
Tabla 11.Especificación Inicio sesión CP	63
Tabla 12.Especificación CU-002 – Ingresar CP	64
Tabla 13.Especificación CU-002 – Actualizar CP.....	65
Tabla 14.Especificación CU-003 – Consultar CP.....	66
Tabla 15.Requerimientos – módulo historias de usuarios.....	70

RESUMEN

Este informe determina el diseño requerido para reforzar el sistema de gestión de requisitos CHAMÍ mediante la integración de los módulos de procedimientos PEVReS, junto a las herramientas de diseño UML-Online, con la finalidad de ayudar a las MIPYMES desarrolladoras de software de la región con el aseguramiento de la calidad de software desde el desarrollo y gestión de requerimientos, lo cual permitirá mejorar su productividad, economizar tiempo y recursos.

Palabras Clave: RQA, Gestión de Requisitos, Calidad de software, Diseño de software, MiPymes.

ABSTRACT

This report determines the design required to strengthen the CHAMÍ requirements management system through the integration of the PEVReS procedures modules together with the UML-Online design tools, in order to help MYPIMES software developers in the region with the software quality assurance from the development and management of requirements, which will improve their productivity, save time and resources.

Keywords: *RQA, Requirements Management, Software Quality, Software Design, MiPymes.*

1. INTRODUCCIÓN

La calidad del software se determina por el cumplimiento de los requerimientos especificados y la satisfacción de necesidades identificadas en el cliente y el usuario final. A lo largo del tiempo la tecnología ha avanzado a un ritmo apresurado y por ende los sistemas son más complejos, en consecuencia, las exigencias de calidad sobre su desarrollo incrementaron.

Para alcanzar un sistema de software de alta calidad, es importante que su desarrollo sea más rígido, lo que reduciría en mayor proporción la cantidad de errores durante su elaboración. De igual manera, identificar y solucionar errores en una etapa temprana del proyecto podría traducirse en disminución de costos y prevención de fallas en fases finales.

La gestión de requisitos es una de las etapas fundamentales para poder desarrollar un software de calidad, los requisitos deben cumplir con ciertas propiedades tales como completo, correcto, realizable y verificable, entre otras. Sin embargo, en algunas ocasiones los requisitos no satisfacen dichas propiedades, lo que se traduce en errores y en mala calidad del producto que se está desarrollando. [1]

Partiendo de la necesidad de explorar alternativas que permitan a la industria del software implementar procesos de aseguramiento de la calidad, se crea el sistema de gestión de requisitos CHAMÍ, el cual nace de un trabajo de tesis doctoral y tiene como propósito, a grandes rasgos, fomentar las buenas prácticas en términos de construcción de software.

El sistema de información CHAMÍ, está en capacidad de acompañar el desarrollo de proyectos de software en la fase de gestión de requisitos, su diseño se ajusta a las necesidades identificadas en los productores de software del segmento Mipyme.

Con el tiempo, se ha logrado demostrar que la gestión de requisitos es de vital importancia en un proyecto de desarrollo de software porque es la fase donde se tiene el primer contacto o interacción con el cliente y se conocen e identifican sus necesidades, esto permite establecer el punto de partida para analizar estas necesidades y determinar cómo abordar el problema.

El presente proyecto tiene como principal objetivo, diseñar el sistema de información integrado PEVReS-CHAMÍ, de tal manera que la herramienta logre proporcionar una cobertura completa de los procesos de desarrollo y gestión de requerimientos.

2. PLANTEAMIENTO DEL PROBLEMA

Actualmente el sistema CHAMÍ, brinda una cobertura en los procesos de desarrollo y gestión de requerimientos debido a que de momento solo cuenta con el módulo de la especificación de requisitos en lenguaje natural, que permite realizar la identificación de requerimientos y especificación de requisitos.

Para garantizar la calidad del software se requiere ampliar la funcionalidad del sistema agregando nuevos módulos del procedimiento PEVReS tales como: Los casos de uso, que son aquellos que describen las actividades a realizar para un actor o algo para llevar a cabo algún proceso, también los casos de pruebas, que son la agrupación de diferentes condiciones bajo las cuales se decreta si el sistema cumple con lo requerido y sus especificaciones, y determinar si es aprobado o desaprobado. A su vez, también el módulo de prototipos, el cual se emplea para validar un prototipo una vez terminado y que este cumpla las necesidades, expectativas y la información recopilada durante el proceso de licitación y pensamiento del diseño. Así como el módulo de historia de usuario, que representa un requisito escrito con frases cortas empleando el mismo lenguaje natural del usuario.

Adicionalmente, se pretende integrar el servicio de las herramientas UML-Online para permitirle a los usuarios una mayor y mejor experiencia al obtener todas las herramientas necesarias dentro del mismo sistema.

3. DELIMITACIÓN

El sistema gestor de requisitos CHAMÍ, hoy en día, funciona en su primera fase de desarrollo, la cual se basa en la identificación, análisis y especificación de requisitos y requerimientos en lenguaje natural para proyectos de desarrollo de software, el cual, podrá ser utilizado por las diferentes empresas desarrolladoras de software en el departamento de Risaralda.

El presente proyecto se enfocó principalmente en la investigación de los módulos PEVReS, el estudio de las herramientas UML y en el análisis y la especificación de requerimientos con sus respectivos diseños estructurales basados en diagramas. Por lo tanto, no se realizó ningún tipo de codificación.

4. JUSTIFICACIÓN

Un buen desarrollo de requisitos es de vital importancia, debido a que es la base para abordar las necesidades, empezar una planificación y concluir con el desarrollo del software. El sistema gestor de requisitos CHAMÍ requiere la integración de nuevos módulos PEVReS para poder ampliar su funcionalidad y la correcta especificación de los requisitos para poder realizar el diseño respectivo de las nuevas funcionalidades que se tendrán dentro del sistema.

La intervención realizada en este proyecto se basó principalmente en asistir el proceso de mejoramiento de las funcionalidades del sistema de gestión de requisitos CHAMÍ, mediante la licitación, análisis y especificación de requisitos con sus respectivos diseños estructurales mediante el uso de herramientas de diagramas UML, para la integración de los módulos de procedimientos PEVReS.

Todo esto, recurriendo siempre a los conocimientos adquiridos durante el proceso universitario, especialmente a aquellas asignaturas relacionadas con la ingeniería de software, acatar las recomendaciones sugeridas por el asesor del proyecto y contar siempre con la mejor actitud y disposición para continuar aprendiendo.

5. OBJETIVOS

5.1. Objetivo General

Diseñar el sistema de información integrado PEVReS-CHAMÍ para el desarrollo y gestión de requisitos.

5.2. Objetivos Específicos

- ❖ Especificar los requerimientos para el desarrollo del sistema de información integrado.
- ❖ Diseñar el sistema de información integrado de acuerdo con los requerimientos identificados.
- ❖ Investigar la integración de herramientas UML on-line al sistema de información.

6. MARCO TEÓRICO

El software es un conjunto de instrucciones o programas que le dicen a un sistema que hacer y cómo hacerlo, haciendo esto que sea totalmente independiente del hardware y hacer que un sistema sea programable, Sabiendo esto, se puede decir que, el desarrollo de software es una mezcla de actividades informáticas enfocadas al proceso de creación, diseño, implementación y soporte de software. [2]

Para mejorar la funcionabilidad del sistema integrado CHAMÍ es fundamental tomar con base las teorías, conceptos y tecnologías adecuadas que permitan garantizar el proceso y que el producto final sea tomado como un producto de calidad.

Por tal razón, a continuación, se presentan algunos conceptos y supuestos de diversos autores reconocidos en la disciplina en los que se basa este proyecto.

6.1. Software de Calidad

EL software de calidad es aquel que realiza las funciones requeridas y a su vez satisfaga las necesidades del cliente; por ende, se puede decir que la calidad de software es “El proceso eficaz de software que se aplica de manera, que crea un producto útil que proporciona valor medible quienes lo producen y a quienes lo utilizan”, del cual cuenta con unos factores como; adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad. [3]

6.2. Gestión de Requerimientos

La gestión de requerimientos “es el proceso de comprensión de y definición de qué servicios se requieren del sistema y de identificación de las restricciones de funcionamiento y desarrollo del mismo” [4]

La gestión de requerimientos es una etapa fundamental durante el proceso de desarrollo de software “ya que los errores en esta etapa originan inevitablemente problemas posteriores en el diseño e implementación del sistema” [4]

6.3. Especificación

Es la tarea de escribir detalladamente el software a ser desarrollado, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.[5]

6.4. Diseño y Arquitectura

Se refiere a determinar cómo funcionará el software de forma general sin entrar en detalles. Consisten en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc. Se definen los casos de uso para cubrir las funciones que realizará el sistema, y se transformarán las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.[5]

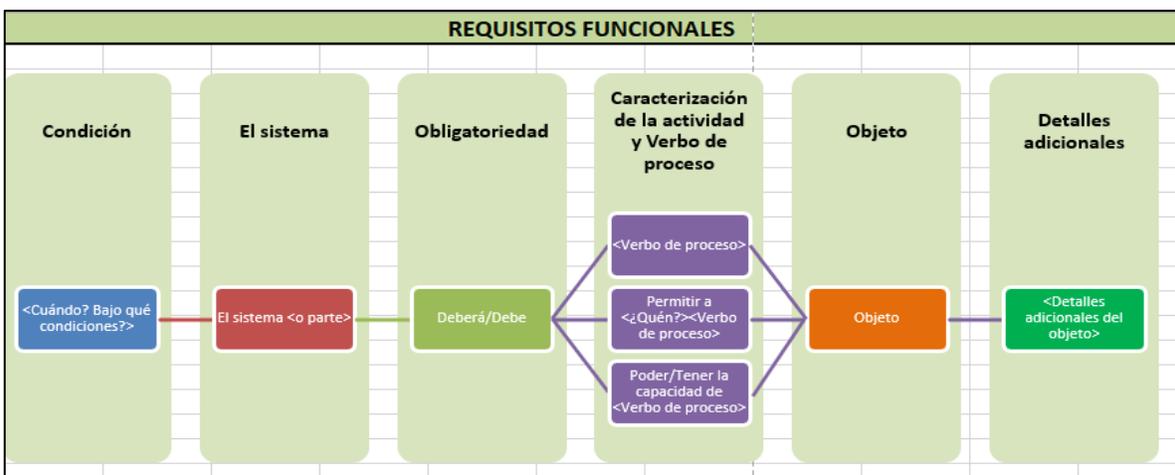
6.5. Procedimiento para la Especificación y Validación de Requisitos de Software (PEVReS)

El Procedimiento para especificar y validar requisitos de software conocido como (PEVReS) [6] proporciona toda la documentación necesaria para la especificación de requisitos de software, lo que permite que se cumpla con los propósitos para el desarrollo de un sistema, en donde el trabajo que se da entre el analista y el cliente refleja, al final del proceso, las necesidades de los usuarios finales a través de una detallada especificación de requerimientos. [6]



Ilustración 1. Procedimiento PEVReS [1]

Por otro lado, dicho procedimiento toma a consideración que toda especificación debe cumplir con ciertas características y cualidades con el fin de dar calidad al producto final: trazable, modificable, consistente, completa; además de la inclusión de la documentación 19 necesaria. La figura 2, muestra un ejemplo de la estructura gramatical que se debe seguir para la redacción de los requerimientos funcionales y no funcionales, de acuerdo con la técnica de especificación de requerimientos en lenguaje natural.[6]



Escenario condicional	Obligatoriedad	Interacción	Objeto	Detalles
	El sistema debe	Permitir a los estudiantes <matricular>	Un curso	Con el cual cumplan los prerrequisitos
	El sistema debe	Permitir a los profesores <consultar>	La lista de estudiantes	Registrado en cada uno de sus cursos
Cuando se ha cumplido fecha de inscripción a un curso o cuando se ha completado el cupo máximo de estudiantes	El sistema debe	<Cerrar>	El curso	
	El sistema debe	Tener la capacidad de <notificar>	Al sistema de pagos	Cuando un estudiante haya realizado el proceso de pago de matrícula
	El sistema debe	Permitir a los administradores <crear>	Cursos	
Tan pronto como un estudiante haya realizado el pago de matrícula		<Imprimir>	La factura del curso	

Ilustración 2. Especificación de requerimientos en lenguaje natural [1].

6.6. Documento de Especificación de Requerimientos (SRS)

Para la estructuración y documentación de los requisitos, PEVReS emplea el documento (SRS), que bajo los estándares de la IEE 830 se conoce como el documento de especificación de requerimientos de software, del cual se conforma con una lista de los requisitos y el contexto de la solución, así como una descripción general del diseño a través de los casos de uso y escenarios.

A la hora de especificar requisitos, se necesita ser muy claro y preciso al documentar toda la información, por esta razón el modelo PEVReS desarrolló diferentes técnicas para poder especificar cada uno los diferentes tipos de requisitos ya sean para los requisitos del sistema, requisitos de negocio y requisitos del usuario; Por ejemplo:

Para la especificación de requisitos del sistema se emplea la especificación en lenguaje natural de requisitos no funcionales y restricciones y la especificación en lenguaje natural de requisitos funcionales.

En la especificación de requisitos de negocio surgieron diferentes técnicas como son: diagramas de flujo de datos (DFD), diagrama de actividades, diagrama de estados, diagrama de secuencia, diagrama de entidad-relación (ERD) y diagrama de clases.

Y para la especificación de requisitos de usuario se emplea los; diagrama de casos de uso, historias de usuario y prototipos. [6]

6.7. Lenguaje Unificado de Modelado (UML)

El lenguaje unificado de modelado (UML), es un fase clave durante el desarrollo tanto de la especificación de requisitos como la codificación, porque hacen que el procedimiento, los requerimientos y necesidades del cliente sean más entendible y se puedan evitar muchos errores y también encontrar soluciones o mejoras al sistema, pero que es UML, “ es un lenguaje modelado, y no un método” [7]

“EL lenguaje unificado modulado define una representación gráfica para visualizar, especificar, construir y documentar los artefactos en un sistema, a lo largo de toda la metodología unificada de desarrollo” [8] UML se emplea en la especificación de requerimientos para realizar el levantamiento de requisitos que describe los actores, el propósito y, por lo tanto, los casos de uso.

6.8. Sistema Gestor de Requisitos “CHAMÍ”

Por otro lado, CHAMÍ, “es un sistema de información diseñado y desarrollado, con el cual se espera que las MIPYMES desarrolladoras de software de la región mejoren la calidad de sus documentos de especificación de requisitos (SRS- Software Requirements Specification).

Este sistema de información, que se comporta en forma de plataforma web, permite realizar el registro y la gestión de los requerimientos de usuario de un proyecto de software, para posteriormente llevar a cabo sobre ellos un conjunto de actividades que, guiadas por el modelo propuesto, permiten asegurar su calidad” [9]

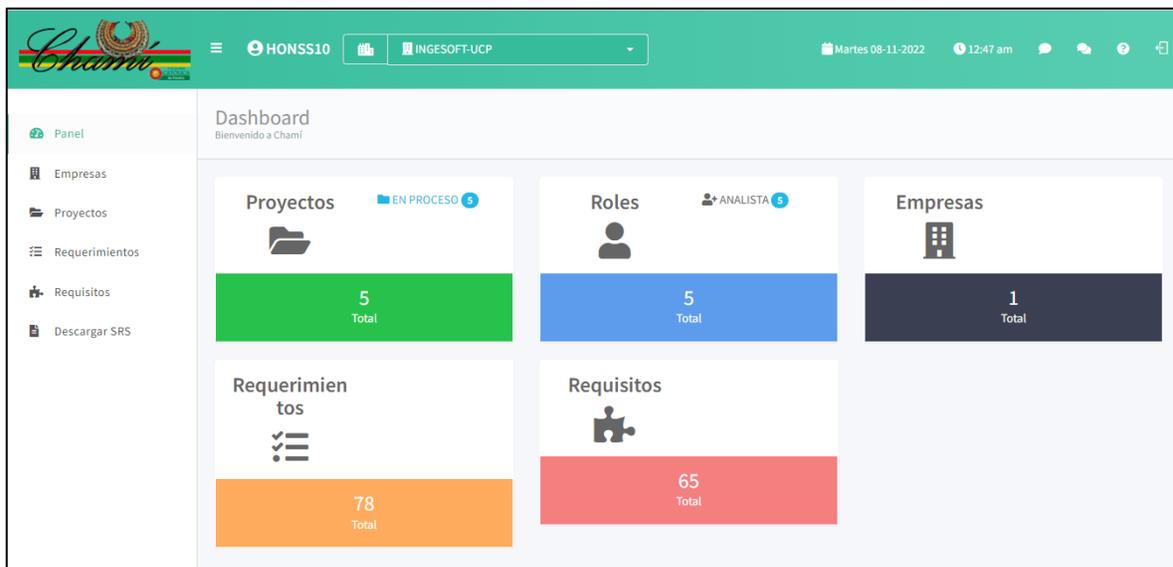


Ilustración 3. Sistema gestor de requisitos “CHAMÍ” [8]

6.9. Metodologías de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. [10]

6.10. Metodología basada en Prototipos

El modelo de prototipos se centra en un diseño rápido que representa las características principales del programa que el usuario podrá ver o utilizar. De esta manera pueden probarlo y dar su opinión sobre distintos aspectos como la usabilidad, la utilidad o el rendimiento, entre otras. El prototipo se puede modificar cuando sea necesario y todos los resultados obtenidos de las presentaciones y pruebas se deben anotar para utilizar posteriormente como ayuda en el desarrollo del producto final.[11].



Ilustración 4. Fases de la metodología basada en Prototipos.[12]

7. METODOLOGÍA

Para la elaboración del proyecto, primero se comenzó con la elicitación de requerimientos (conocido como levantamientos de requisitos o necesidades), mediante reuniones y llamadas frecuentes con el administrador del sistema “CHAMÍ”, con el fin de identificar aquello que requerían y gestionar las posibles soluciones.

Luego de precisar sus requerimientos y necesidades, que correspondían específicamente con el diseño para la integración de nuevos módulos, se procedió a realizar la investigación acerca del sistema para poder conocerlo a fondo y posteriormente pasar a la fase de la especificación de cada uno de los requisitos, de esta forma determinar concretamente lo necesario para el diseño de cada módulo.

Una vez concretada la fase de especificación, se pasa a diseñar los módulos mediante diagramas UML, empleando en este caso, la herramienta Enterprise Architect. Luego de finalizado el diseño de los módulos, se presenta el informe al cliente para poder recibir su aprobación.

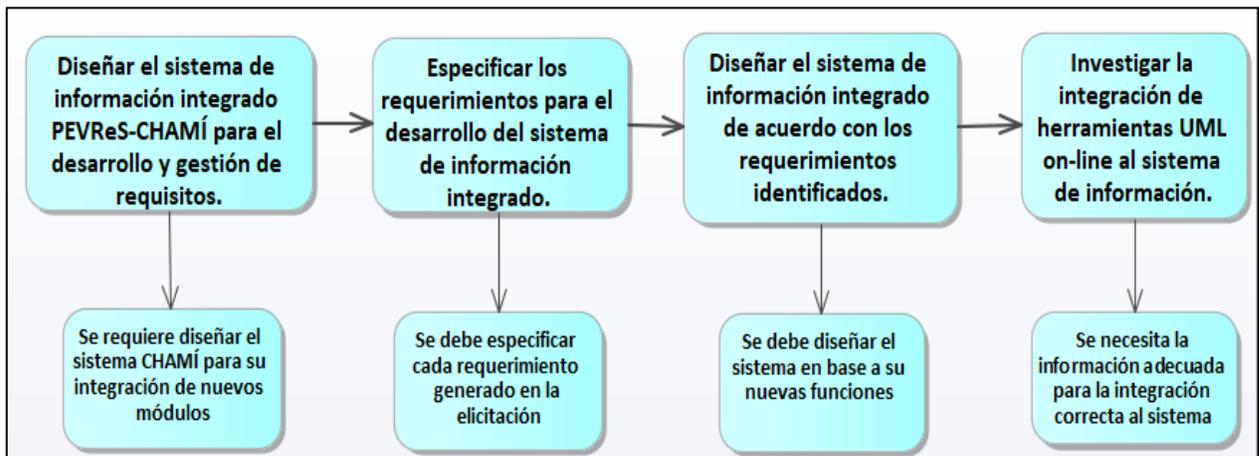


Ilustración 5. Metodología del proyecto.

Para cada objetivo planteado, se necesitaba especificar lo que se requería, para que de cierta manera todo el proceso durante el desarrollo de este, fuera más claro y específico, y así mismo lograr el resultado esperado. Como se demuestra en la ilustración 5.

7.1. Planificación del Proyecto (Cronograma de Actividades)

Para el desarrollo del proyecto se estableció un tiempo estimado para cumplir cada objetivo planteado, como se demuestran a continuación en la tabla 1, correspondiente al cronograma de actividades.

Tabla 1. Cronograma de actividades.

OBJETIVO/Actividad	ABR	MAY	JUN	AGO	SET	OCT	NOV
OBJETIVO 1							
Recopilar información sobre el sistema de gestión de requisitos CHAMI							
Identificación de requerimientos							
Realizar Prototipos							
OBJETIVO 2							
Diseñar módulo para Diagramas de Clase							
Diseñar módulo para Diagramas de Casos de Uso							
Diseñar módulos para Casos de Prueba							
Diseñar módulo para Prototipos							
Diseñar módulos para Historias de usuarios							
Diseñar el diagrama Relacional para la base de datos							
OBJETIVO 3							
Investigar la integración de herramientas UML con CHAMÍ							
Entrega de informe							

8. DESARROLLO DEL PROYECTO

Para el desarrollo del proyecto también era importante investigar o consultar, la existencia de algún proyecto similar, para poder evidenciar su funcionamiento y diferencias con respecto al sistema CHAMÍ y además de establecer e incorporar una metodología de desarrollo con respecto al proyecto.

8.1. Antecedentes de proyectos Similares

“SIAR (Sistema Integral de Administración de Requerimientos); El presente presenta una línea de investigación en donde se describe una aplicación web denominada SIAR (Sistema Integral de Administración de Requerimientos) que administra y valida los requerimientos del Modelado Conceptual de un sistema de información mediante Casos de Uso, según los lineamientos de UML (Lenguaje Unificado de Modelado). La finalidad principal de SIAR es la administración de Casos de Uso con una herramienta informática que agilice su registración, normalice su contenido y posibilite implementar validaciones funcionales, como por ejemplo un método automatizado de análisis de consistencia de Casos de Uso.”[13]

The screenshot displays the SIAR web application interface, which is organized into several sections:

- Trabajar con Sistemas:** A header section with a search field labeled "Nombre".
- System List:** A table listing systems. The first entry is "ADMINISTRADOR DE CURSOS" with ID 1, created on 22/06/2013 by MOYANO ENRIQUE.
- Versiones Sistema:** A sub-section for "ADMINISTRADOR DE CURSOS" with a "Sis Ver Cod" field. It contains a table with one entry: "1 - 1" created on 12/08/13 by MOYANO ENRIQUE.
- Casos de Uso:** A section for "ADMINISTRADOR DE CURSOS Versión: 1 - 1 Alcance: INGRESO DE DATOS" with a "Fecha de Alta" field. It contains a table listing use cases:

	Caso Uso Id	Fecha Alta	Nombre	Tipo	Complejidad	Usuario Creador	Versiones
	1 - 1 - 7 - 1	12/08/13	CONSULTAR CURSO	Concreto	Media	MOYANO ENRIQUE	
	1 - 1 - 7 - 2	12/09/13	PRUEBA	Abstracto	Alta	MOYANO ENRIQUE	

Ilustración 6. Sistema Integral de Administración de Requerimientos “SIAR” [13]

“**Rational Doors Next Generation**; desarrollada por IBM, esta herramienta tiene por objetivo la especificación de los requerimientos por medio de unidades lógicas conocidas como artefactos, las cuales describen los requerimientos y todos los detalles correspondientes. La Rational Doors Next Generation es muy usada debido a que permite obtener una perspectiva muy detallada de los casos de uso.”[14]

The screenshot displays the Rational Doors Next Generation interface for an "AMR Stakeholder Requirements Specification". The main content area shows a list of requirements with the following details:

ID	Contents	Artifact Type								
63	-1 Introduction	Heading								
17	ANSI 1252 Latin 1 for CSV Export	Information								
177	-1.1 Purpose of the Document	Heading								
303	The Stakeholder requirements describe the user needs for a water meter reading system (AMR) including the meter, the method of collecting data from the device, and the means by which the customers will be billed for usage.	Information								
357	The AMR product is intended to allow water providers to lower their cost of operation by more accurately measuring consumption and more quickly gather data.	Information								
132	-1.2 Intended Use	Heading								
258	The AMR is intended for gathering and reporting of water consumption data to facilitate measurement of residential and business customers.	Information								
144		Diagrams and sketches								
134	-1.3 Definitions Acronyms, and Abbreviations	Heading								
315	<table border="1"> <thead> <tr> <th>Term</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>AMR</td> <td>Automatic Meter Reader</td> </tr> <tr> <td>MMIU</td> <td>Multi-channel Meter Interface Unit</td> </tr> <tr> <td>CTRL</td> <td>Central Control Center</td> </tr> </tbody> </table>	Term	Definition	AMR	Automatic Meter Reader	MMIU	Multi-channel Meter Interface Unit	CTRL	Central Control Center	Information
Term	Definition									
AMR	Automatic Meter Reader									
MMIU	Multi-channel Meter Interface Unit									
CTRL	Central Control Center									

The right-hand sidebar shows details for the selected artifact "8: AMR Stakeholder Requirements Specification", including its description, ownership, and creation/modification history.

Ilustración 7. Rational Doors Next Generation. [15]

Visure Requirements: Es un software muy flexible a la hora de hablar ingeniería de requerimientos. Capaz de racionalizar y gestionar los procesos de gestión de requerimientos por ende es posible alcanzar una excelente calidad. Al igual que otro software permite gestionar (requisitos, pruebas, solicitudes de cambio, etc) y está diseñado para un fácil uso con excelentes resultados.

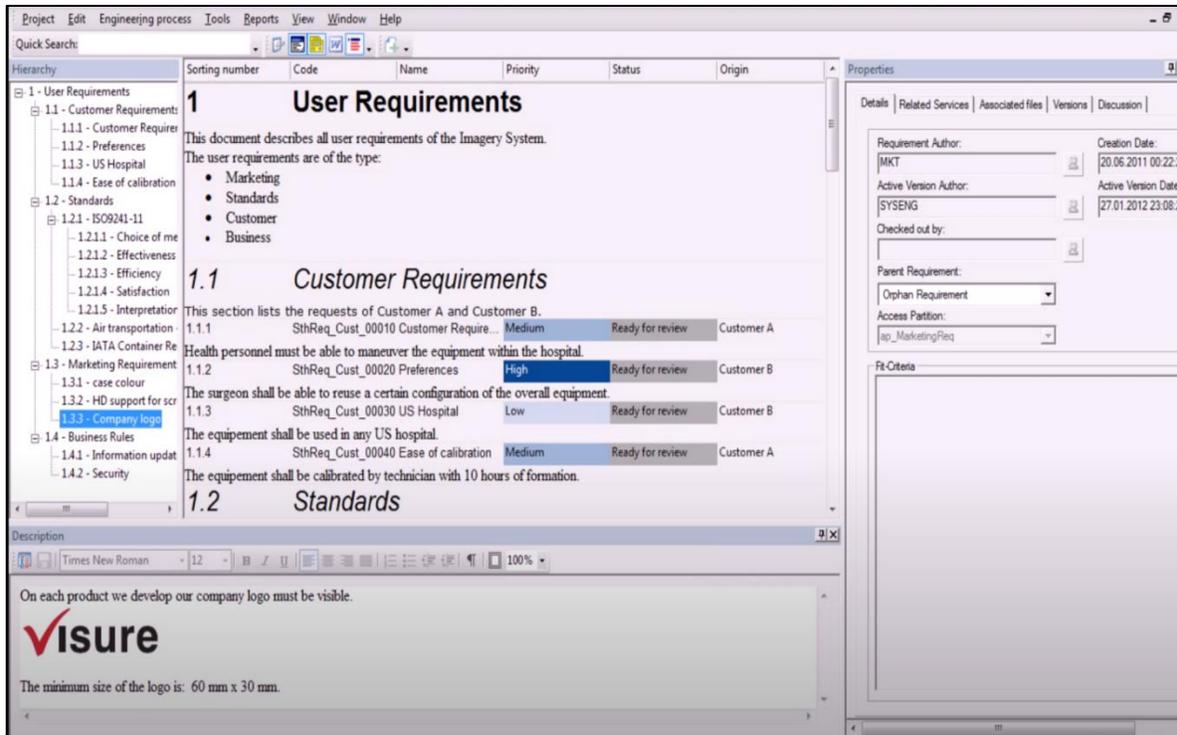


Ilustración 8. Sistema de Visual Requirements

RaQuest: Es una herramienta sobre la gestión de requerimientos para el modelado UML. Se puede realizar un seguimiento a los requisitos con una serie de características. Las características principales de este software son:

- o Definir y gestionar elementos de requisitos.
- o Crear paquetes para gestionar artículos de requisito.
- o Generar documentos para una parte o todo el proyecto en los siguientes formatos (HTML, CSV, Word, Excel, RTF).

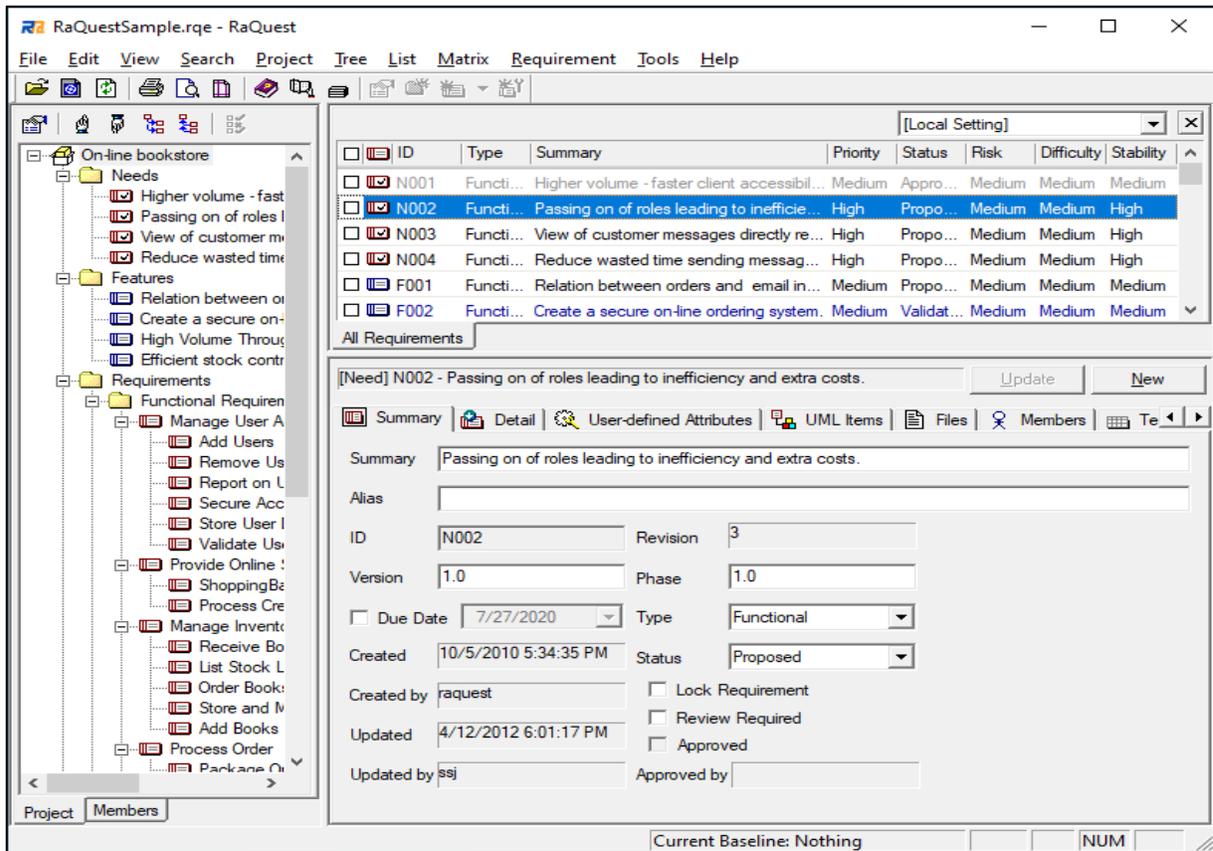


Ilustración 9.Sistema Raquest

Doc Sheets: es un software de gestión de requisitos único, con inigualable facilidad de uso, intuición y flexibilidad, se puede trabajar con requisitos utilizando múltiples vistas (árbol, cuadrícula, flujo, tarjeta y documento). Permite ahorrar tiempo importando automáticamente documentos desde hojas de cálculo y documentos de Word y generar documentos de especificación con una tabla de contenido automáticamente.

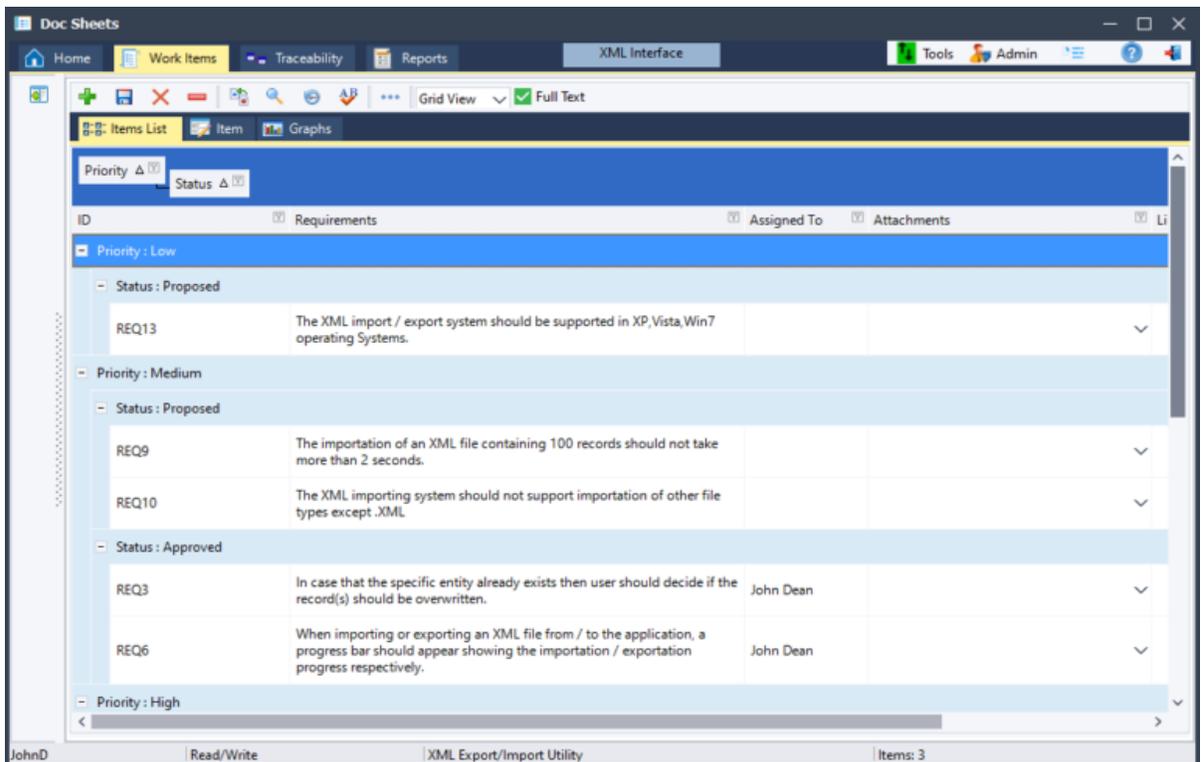


Ilustración 10. Sistema Doc Sheeta

8.2. Aplicación de Metodología de Desarrollo

Si bien el alcance del proyecto no incluyó las fases de codificación y validación, ya que está enfocado en el diseño de los módulos, se siguió la metodología en Prototipos. Esta metodología propone el desarrollo de un sistema de información siguiendo 6 pasos. A continuación, se presentan de manera resumida las actividades realizadas en proyecto por cada uno de estos pasos:

- 1) **Recolección y refinamiento de requisitos:** En este paso se recolectó cada uno de los requerimientos expresados por el cliente para luego ser especificados.
- 2) **Diseño Rápido:** Se realizó un pequeño mockup para dar una idea más clara de lo que se requiere y se va a desarrollar.

- 3) **Construcción del prototipo:** En este punto se agrupó los requerimientos ya especificados con el diseño rápido para ahora si establecer una idea más real o definitiva a lo que se quiere llegar.
- 4) **Evaluación del prototipo por el cliente:** Al concluir la construcción del prototipo se le enseña al cliente el trabajo realizado para obtener una evaluación de este, para ver si cumple con los requerimientos propuestos.
- 5) **Refinamiento del prototipo:** Una vez finalizada la evaluación por el cliente, se procede a realizar un refinamiento del prototipo, donde se puede modificar o mejorar algunas funciones para que este pueda realizar su objetivo o cumplir las necesidades del cliente.
- 6) **Producto de ingeniería:** Una vez terminado todo el proceso de los pasos anteriormente mencionados, se envía el resultado al área de desarrollo para culminar el proyecto.

Es de anotar, que estas actividades se llevaron a cabo para cada uno de los módulos propuestos, utilizando una secuencia en basada en iteraciones, de la siguiente manera:

- Iteración 1: Módulo para la Verificación y Validación de Prototipos
- Iteración 2: Módulo para la Especificación y Verificación de Casos de Usos
- Iteración 3: Módulo para Casos de Pruebas
- Iteración 4: Módulo para Historias de Usuarios

9. ANÁLISIS DE RESULTADOS

Durante el desarrollo de este proyecto, se analizó la forma de integrar algunas técnicas del procedimiento PEVReS al sistema CHAMÍ, de tal manera que se logre mejorar su alcance. En este sentido, las nuevas funcionalidades a integrar al sistema corresponden a las presentadas en la Ilustración 11.

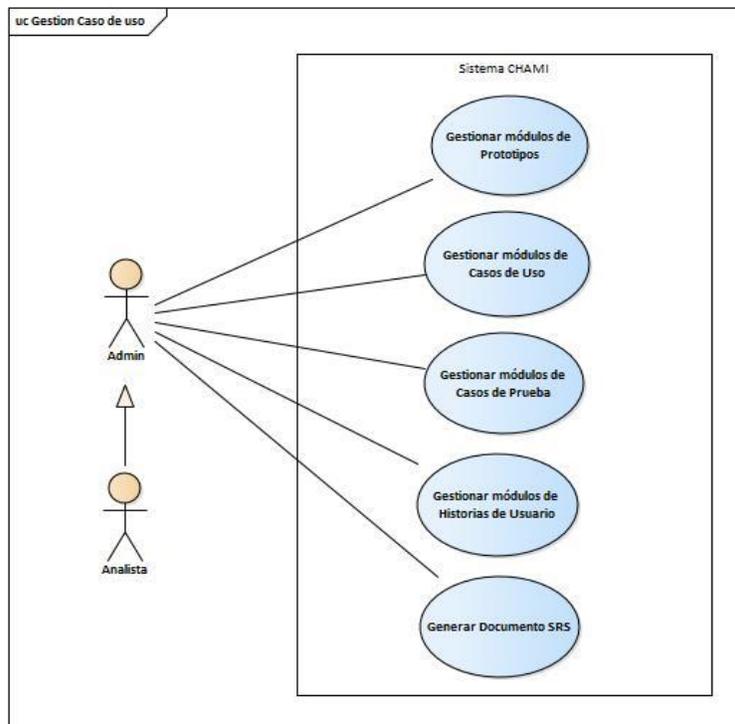


Ilustración 11. Diagrama de casos de uso de las nuevas funcionalidades para la gestión del sistema Chamí.

Con la intención de integrar las nuevas funcionales al sistema, se realizó el diseño de los siguientes módulos, los cuales serán descritos en la siguiente sección:

- a) Módulo para la Verificación y Validación de Prototipos.
- b) Módulo para la Especificación y Verificación de Casos de Usos
- c) Módulo para Casos de Pruebas
- d) Módulo para Historias de Usuarios

9.1 Módulo para la Verificación y Validación de Prototipos

En el desarrollo de un proyecto de software, el módulo de prototipos sirve para validar un prototipo una vez terminado, tales que este cumpla las necesidades, expectativas y la información recopilada durante el proceso de licitación y pensamiento del diseño.

Para la integración del módulo de verificación y validación de prototipos, se realizó el siguiente diseño estructural:

a. Requerimientos

Para la integración de este módulo se establecieron por parte del cliente los siguientes requerimientos, representados en la Tabla 2

Tabla 2. Requerimientos para el módulo de prototipos

ID	TIPO	REQUISITOS DE NEGOCIO
BR-PRO-1	Objetivo de negocio	Permitir de forma ágil y rápida la gestión y validación de prototipos para proyectos de software
BR-PRO-2	Objetivo de negocio	Validación y aprobación de prototipos del cliente final de forma ágil
BR-PRO-3	Objetivo de negocio	Aumentar la capacidad de modelo de negocio a medida que se vayan registrando más administradores y usuarios.
BR-PRO-4	Objetivo de negocio	Disponibilidad para permitir una continuidad de negocio
BR-PRO-5	Regla de negocio	El sistema deberá tener 2 Loguin para separar el diseñador de prototipos y el usuario que valida.

Los requerimientos solicitados para este módulo se han diligenciado en el sistema CHAMÍ para demostrar el uso funcional que tiene hasta la fecha.

Gestión de Requerimientos

PROY. G1 - PROTOTIPOS

Buscar

ID	Fecha	Proyecto	Prioridad	Tipo de Usuario	Medio de Recepción	Nombre	Versión	Estado
344	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	VALIDACIÓN DE PROTOTIPO	0	RECIBIDO
345	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	VALIDACIÓN POR PARTE DEL CLIENTE	0	RECIBIDO
346	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	AUMENTO DE CAPACIDAD	0	RECIBIDO
347	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	DISPONIBILIDAD	0	RECIBIDO
348	2022-10-26	PROY. G1 - PROTOTIPOS	ALTA	GENERAL	OTRO	LOGIN	0	RECIBIDO
349	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	ACCESO AL SISTEMA	0	RECIBIDO
350	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	REGISTRARSE	0	RECIBIDO
351	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	CREAR PROYECTO	0	RECIBIDO
352	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	LISTADO DE PROTOTIPOS	0	RECIBIDO
353	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	VALIDAR PROTOTIPO	0	RECIBIDO
354	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	DISPONIBILIDAD	0	RECIBIDO
355	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	USABILIDAD	0	RECIBIDO
356	2022-10-26	PROY. G1 - PROTOTIPOS	MEDIA	GENERAL	OTRO	RESTRICCIÓN DE PROTOTIPOS	0	RECIBIDO

Mostrando registros del 1 al 13 de un total de 13 registros

Ilustración 12. Requerimientos del módulo prototipos implementados en el sistema Chamí

b. Diagrama de Actividades

Se representó los requerimientos mediante un diseño de diagrama de actividades reflejado en la siguiente Ilustración 13.

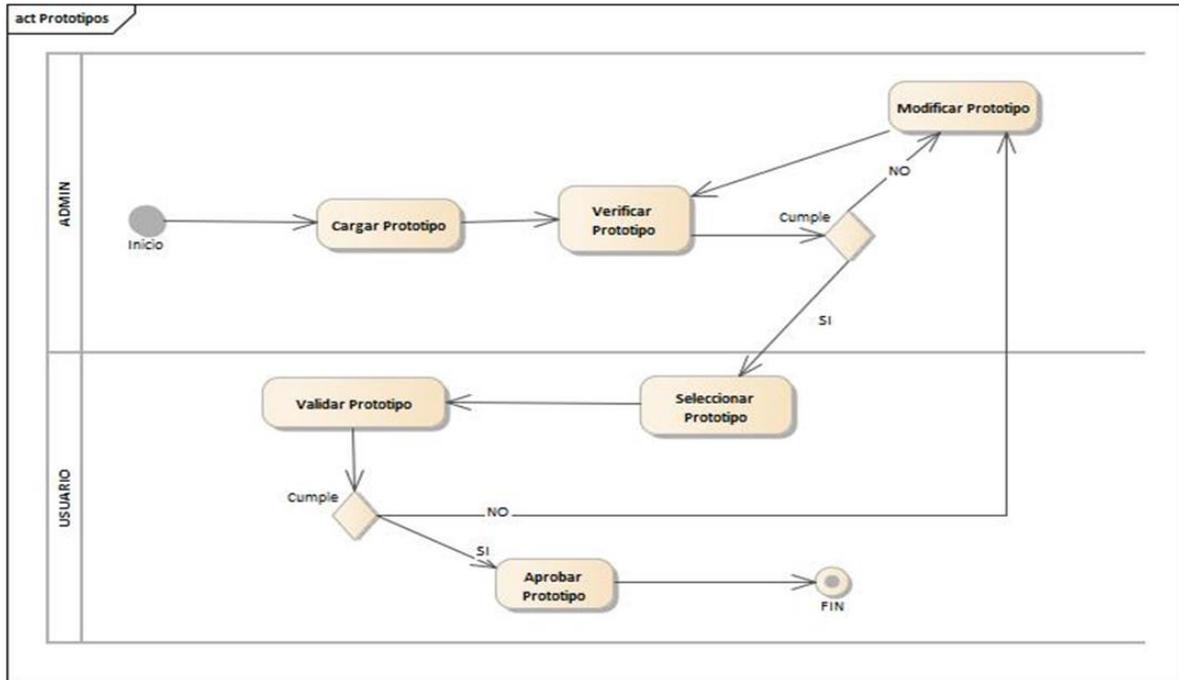


Ilustración 13. Diagrama de actividad para el módulo de prototipos

c. Diagrama de Casos de uso

Para determinar la actividad que le correspondía a cada uno de los actores dentro del sistema, se implementaron los requerimientos mediante un diseño de diagrama de casos de uso, reflejado en la siguiente Ilustración 14.

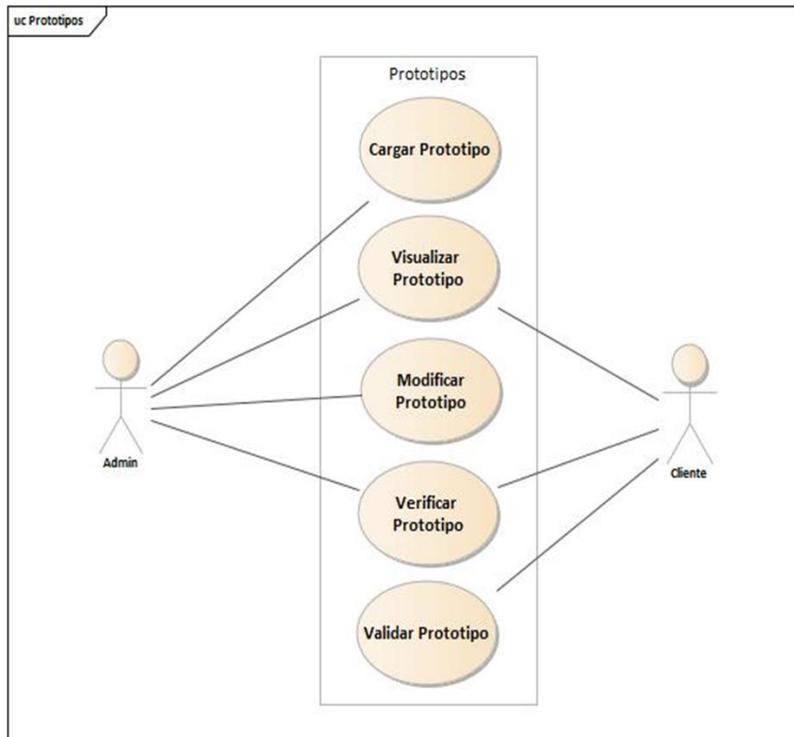


Ilustración 14. Diagrama de casos de uso para el módulo de prototipos.

d. Especificación de requisitos en Lenguaje Natural

Tabla 3. Especificación detallada del primer requisito

ID	REQ-VP-1
Descripción	Acceder al sistema mediante al botón de Loguin haciendo clic
Datos de Entrada	Datos de usuario y contraseña
Salidas	Ingreso al aplicativo web
Excepciones	Si el usuario no está registrado debe registrarse (REQ-VP-2) antes de poder ingresar

Tabla 4. Especificación detallada del segundo requisito

ID	REQ-VP-2
Descripción	Acceder al formulario de registro dando clic en el botón de registro
Entradas	Datos de nombre, usuario, correo electrónico, numero celular y contraseña
Salidas	Se muestra una ventana de registro exitoso
Excepciones	El usuario ya está registrado

Tabla 5. Especificación detallada del tercer requisito

ID	REQ-VP-3
Descripción	Al momento de crear un proyecto nuevo para cargar los prototipos debemos previamente dar clic en crear proyecto y llenar la información solicitada y cargar el archivo.
Entradas	Nombre de proyecto, código de prototipo y archivo
Salidas	Mensaje del sistema (proyecto creado exitosamente y las opciones de salir y agregar más prototipos)
Excepciones	<ul style="list-style-type: none"> • Si el proyecto ya está creado se alerta al usuario informando que el nombre del proyecto ya se encuentra creado • Si la información no está completa se alerta al usuario que complete la información y cargue el archivo nuevamente

Tabla 6. Especificación detallada del cuarto requisito

ID	REQ-VP-4
Descripción	Visualizar los prototipos subidos por el diseñador.
Entradas	Dar clic al botón prototipos subidos
Salidas	Lista de prototipos
Excepciones	Si el diseñador no ha subido ningún prototipo al usuario le aparece un mensaje no hay ningún prototipo subido

Tabla 7. Especificación detallada del quinto requisito

ID	REQ-VP-5
Descripción	El diseñador debe poder verificar el prototipo. Esto se hace en las listas de prototipos donde se muestra una casilla para verificar el prototipo.
Entradas	Hacer clic en el link del segundo paso para verificar.
Salidas	Se visualiza el test de preguntas junto al prototipo que se subió.
Excepciones	<ul style="list-style-type: none"> • Se debe de responder todas las preguntas marcando. Si no responde el test no puede seguir al siguiente paso. • Si responde no, se debe de hacer las observaciones correspondientes para que se vuelva a realizar el prototipo.

Tabla 8. Especificación detallada del sexto requisito

ID	REQ-VP-6
Descripción	El usuario debe poder validar el prototipo después de que se verifique el prototipo.
Entradas	Hacer clic en el link del tercer paso para validar.
Salidas	Se visualiza el test de preguntas junto al prototipo que se subió.
Excepciones	<ul style="list-style-type: none"> • Se debe de responder todas las preguntas marcando. Si no responde el test no puede seguir al siguiente paso. • Si responde no, se debe de hacer las observaciones correspondientes para que el diseñador haga los ajustes necesarios para que quede listo.

e. Diagrama de Clases

Se ha representado la estructura de este módulo, para establecer sus clases, atributos, relaciones y operaciones, mediante el siguiente diagrama de clases, como se demuestra a través de la Ilustración 15.

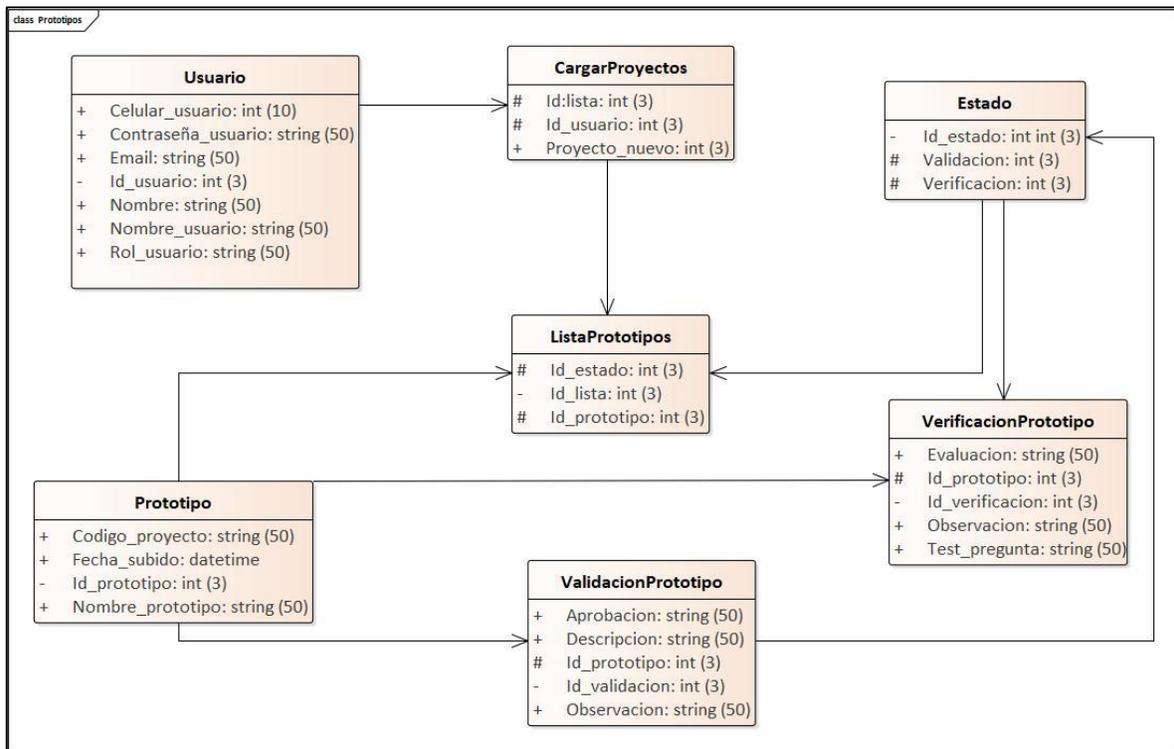


Ilustración 15. Diagrama de clases módulo prototipos.

f. Mockups

Para tener una idea más clara y tener un modelo de cómo será el resultado final o esperado de este módulo, se ha diseñado el siguiente mockup.

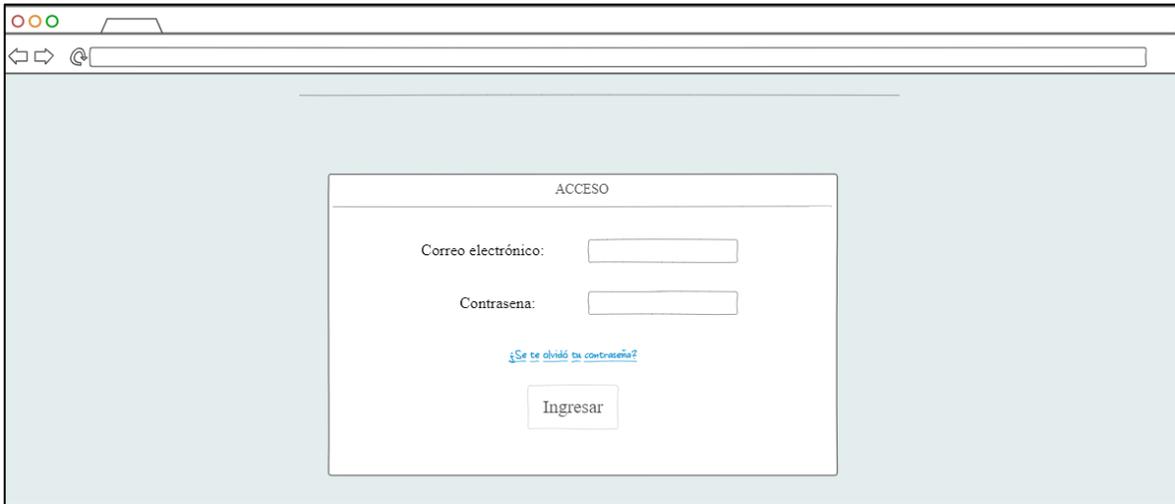


Ilustración 16. Mockups página de inicio - módulo prototipos

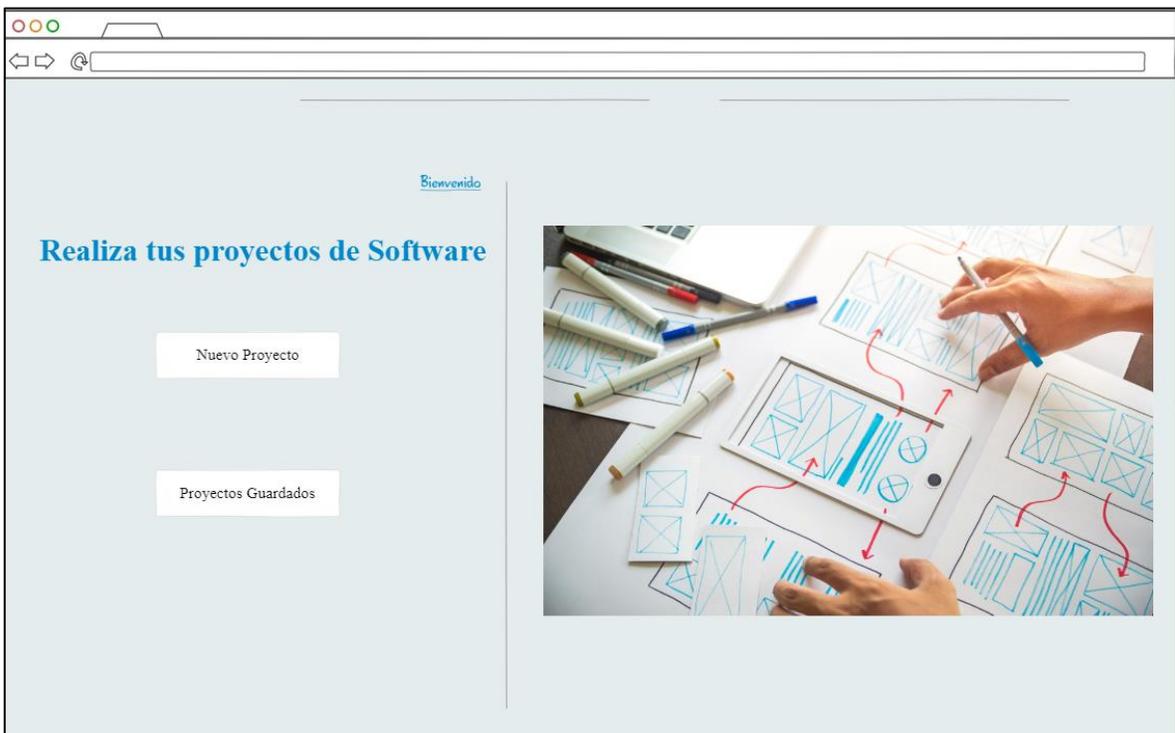


Ilustración 17. Mockups ver o crear nuevo proyecto - módulo prototipos

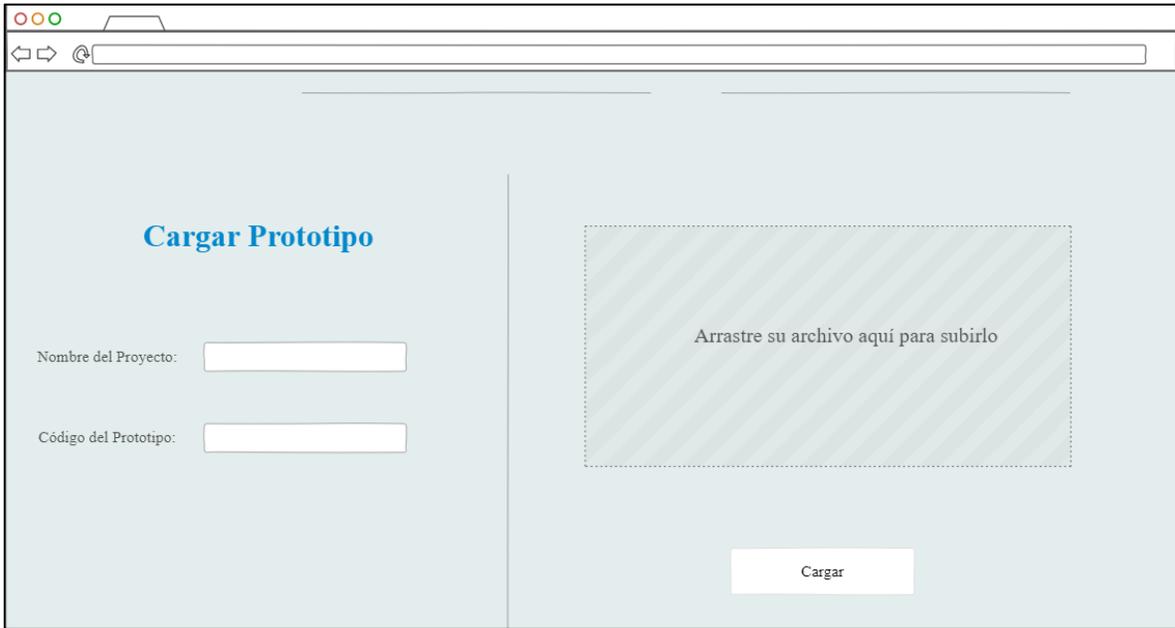


Ilustración 18. Mockups crear proyecto - módulo prototipos

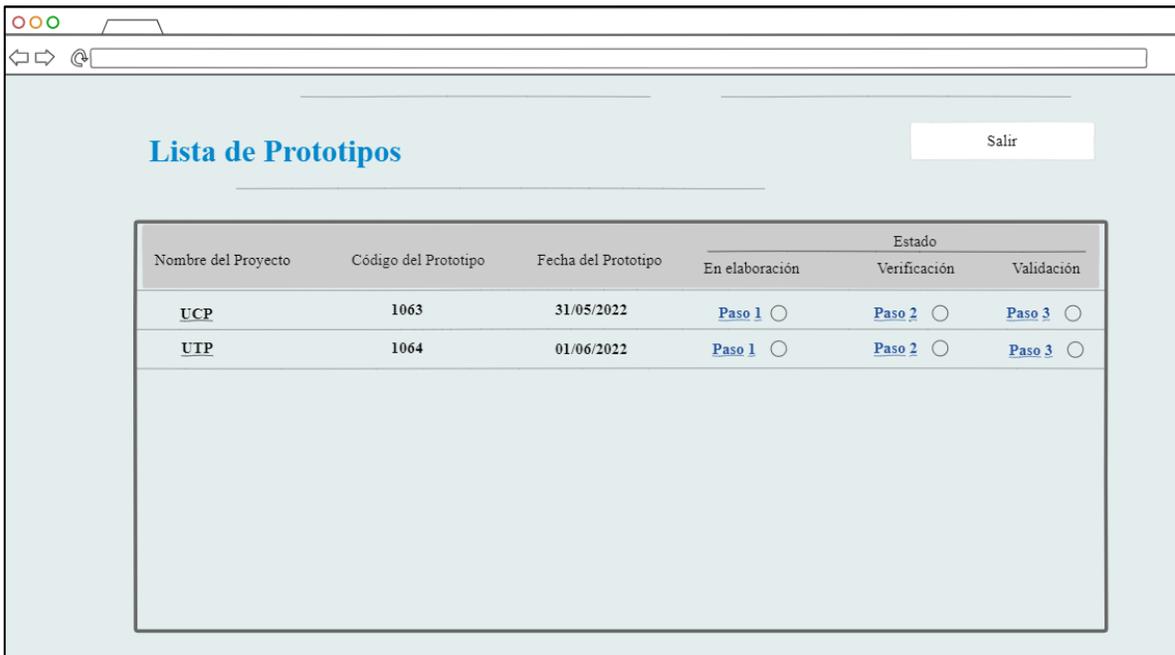


Ilustración 19. Mockups lista de prototipos - módulo prototipos

Regresar

CHECKLIST PARA LA VERIFICACIÓN DE LOS PROTOTIPOS				
No.	PREGUNTA	EVALUACIÓN		OBSERVACIONES
		SI	NO	
1	El prototipo está relacionado con algún caso de uso, historia de usuario o requisito.	<input type="radio"/>	<input type="radio"/>	
2	El prototipo es coherente con el caso de uso, historia de usuario o requisito.	<input type="radio"/>	<input type="radio"/>	
3	Los títulos y leyendas presentes en el prototipo son coherentes con lo especificado en el caso de uso, historia de usuario o requisito.	<input type="radio"/>	<input type="radio"/>	
4	Los campos y elementos del prototipo corresponden con los mencionados en el caso de uso, historia de usuario o requisito.	<input type="radio"/>	<input type="radio"/>	
5	Los campos de entrada usados en el prototipo parecen lógicos para el tipo de dato a usar.	<input type="radio"/>	<input type="radio"/>	
6	Los términos y lenguaje usados en el prototipo son adecuados para el tipo de Actor o Rol que lo va a utilizar.	<input type="radio"/>	<input type="radio"/>	
7	Los términos y lenguaje usados en el prototipo son adecuados para el caso de negocio o requisito que se presenta.	<input type="radio"/>	<input type="radio"/>	
8	El orden de los elementos presentes en el prototipo parecen lógicos y permiten representar el flujo de eventos del caso de uso o la historia de usuario.	<input type="radio"/>	<input type="radio"/>	

Guardar

Ilustración 20. Mockups pestaña de verificación de prototipos - módulo prototipos

9.2. Módulo para la Especificación y Verificación de Casos de Usos

El módulo de caso de uso es la fase en la cual se especifican la interacción de cada uno de los actores con el sistema. Para la integración del módulo de especificación y verificación de casos de usos, se realizó el siguiente diseño.

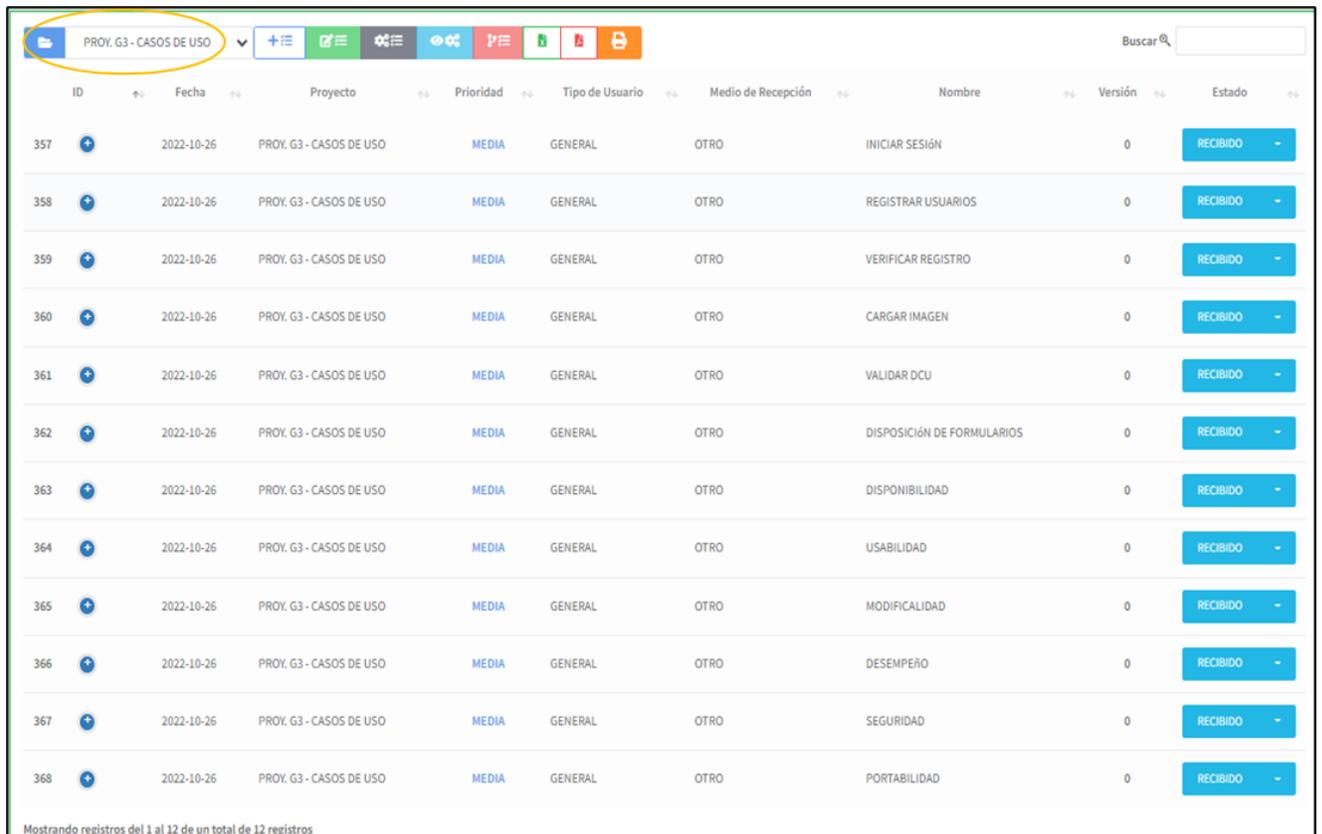
b) Requerimientos

Para la integración de este módulo se establecieron por parte del cliente los siguientes requerimientos, representados en la Tabla 9

Tabla 9. Requerimientos – módulo casos de uso

ID	TIPO	REQUISITOS DE NEGOCIO
RQN-1	Objetivo de negocio	Automatizar el proceso de validación de los casos de uso
RQN-2	Objetivo de negocio	Verificar los casos de uso para que, al momento del desarrollo, garantizar software de calidad
RQN-3	Objetivo de negocio	Cumplir con las políticas de los tratamientos de datos de los casos de uso diligenciados por el usuario
RQN-4	Objetivo de negocio	Estandarizar los formularios en la página web para evitar el uso de archivos locales (documentos de Office)
RQN-5	Objetivo de negocio	Reducir el tiempo en la elaboración de los formularios de validación
RQN-6	Regla de negocio	Registrar y almacenar los usuarios en la nube
RQN-7	Regla de negocio	Permitir acceso en cada momento a los formularios diligenciados desde cualquier dispositivo y lugar
RQN-8	Regla de negocio	Presentar la página web en un hosting y dominio gratuito

Los requerimientos solicitados para este módulo se han diligenciado en el sistema CHAMÍ para demostrar el uso funcional que tiene hasta la fecha.



ID	Fecha	Proyecto	Prioridad	Tipo de Usuario	Medio de Recepción	Nombre	Versión	Estado
357	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	INICIAR SESIÓN	0	RECIBIDO
358	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	REGISTRAR USUARIOS	0	RECIBIDO
359	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	VERIFICAR REGISTRO	0	RECIBIDO
360	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	CARGAR IMAGEN	0	RECIBIDO
361	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	VALIDAR DCU	0	RECIBIDO
362	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	DISPOSICIÓN DE FORMULARIOS	0	RECIBIDO
363	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	DISPONIBILIDAD	0	RECIBIDO
364	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	USABILIDAD	0	RECIBIDO
365	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	MODIFICABILIDAD	0	RECIBIDO
366	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	DESEMPEÑO	0	RECIBIDO
367	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	SEGURIDAD	0	RECIBIDO
368	2022-10-26	PROY. G3 - CASOS DE USO	MEDIA	GENERAL	OTRO	PORTABILIDAD	0	RECIBIDO

Mostrando registros del 1 al 12 de un total de 12 registros

Ilustración 21. Requerimientos del módulo casos de uso implementados en el sistema Chamí

c) Diagrama de Actividades

Se representó los requerimientos mediante un diseño de diagrama de actividades reflejado en las siguientes ilustraciones.

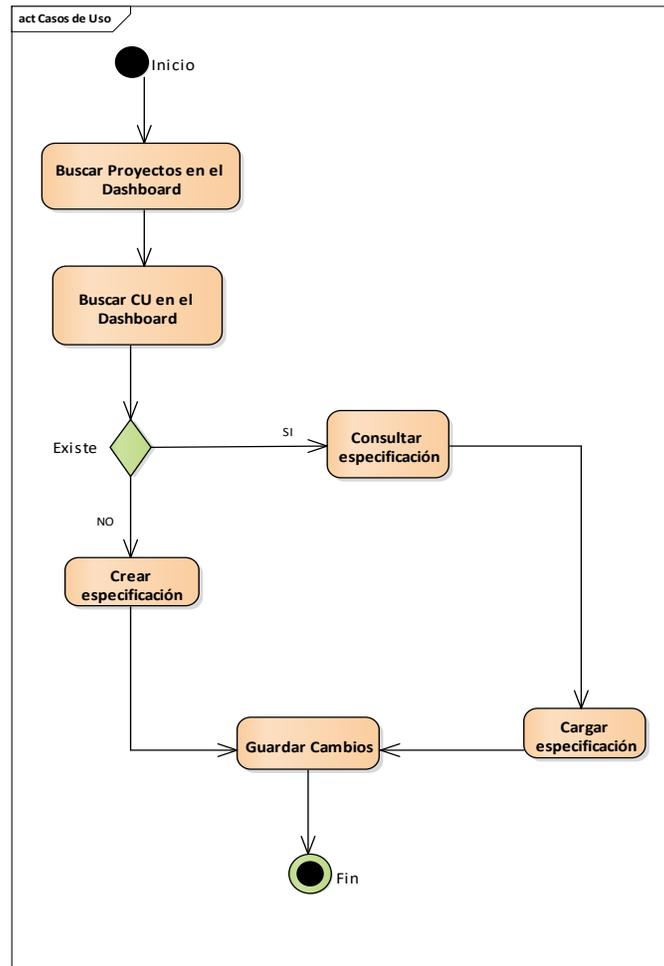


Ilustración 22. Diagrama de actividad consultar CU- módulo casos de uso

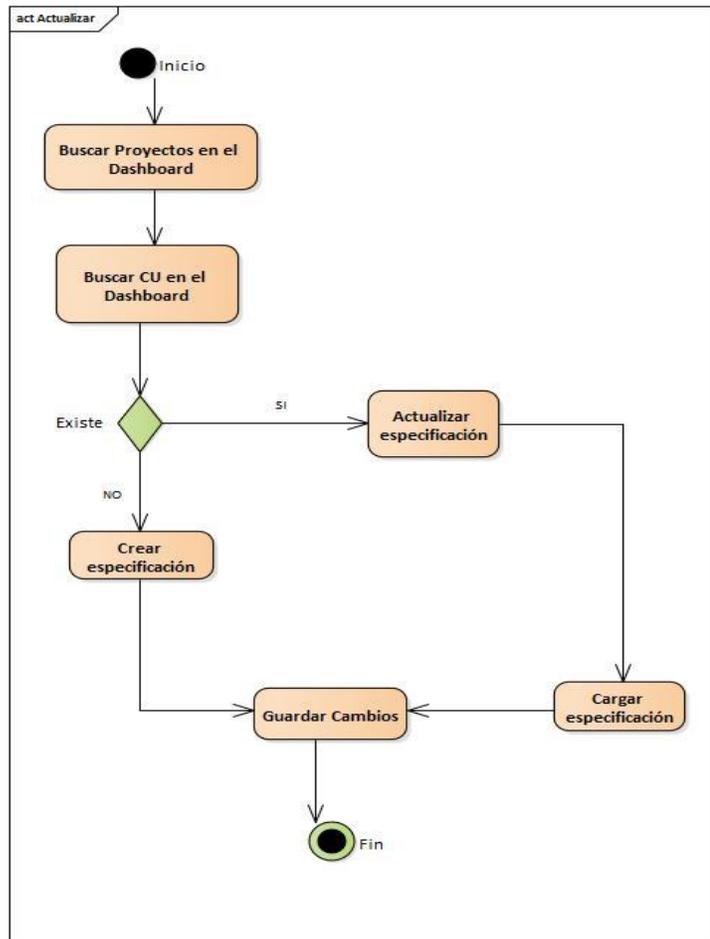


Ilustración 23. Diagrama de actividad actualizar CU- módulo casos de uso

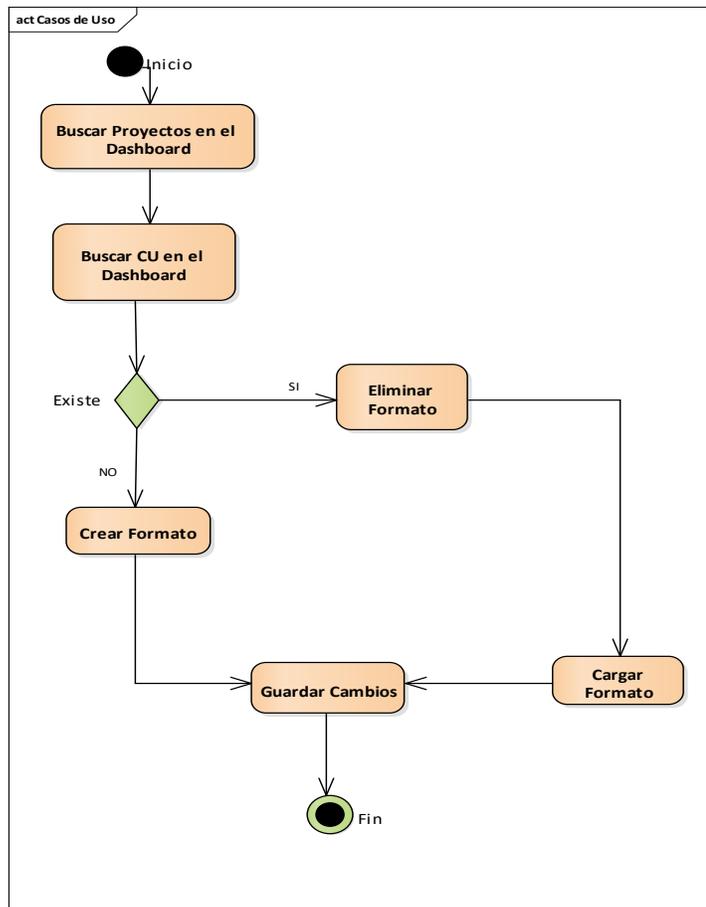


Ilustración 24. Diagrama de actividad eliminar CU- módulo casos de uso

d) Diagrama de Casos de Uso

Para determinar la actividad que le correspondía a cada uno de los actores dentro del sistema, se implementaron los requerimientos mediante un diseño de diagrama de casos de uso, reflejado en la siguiente.

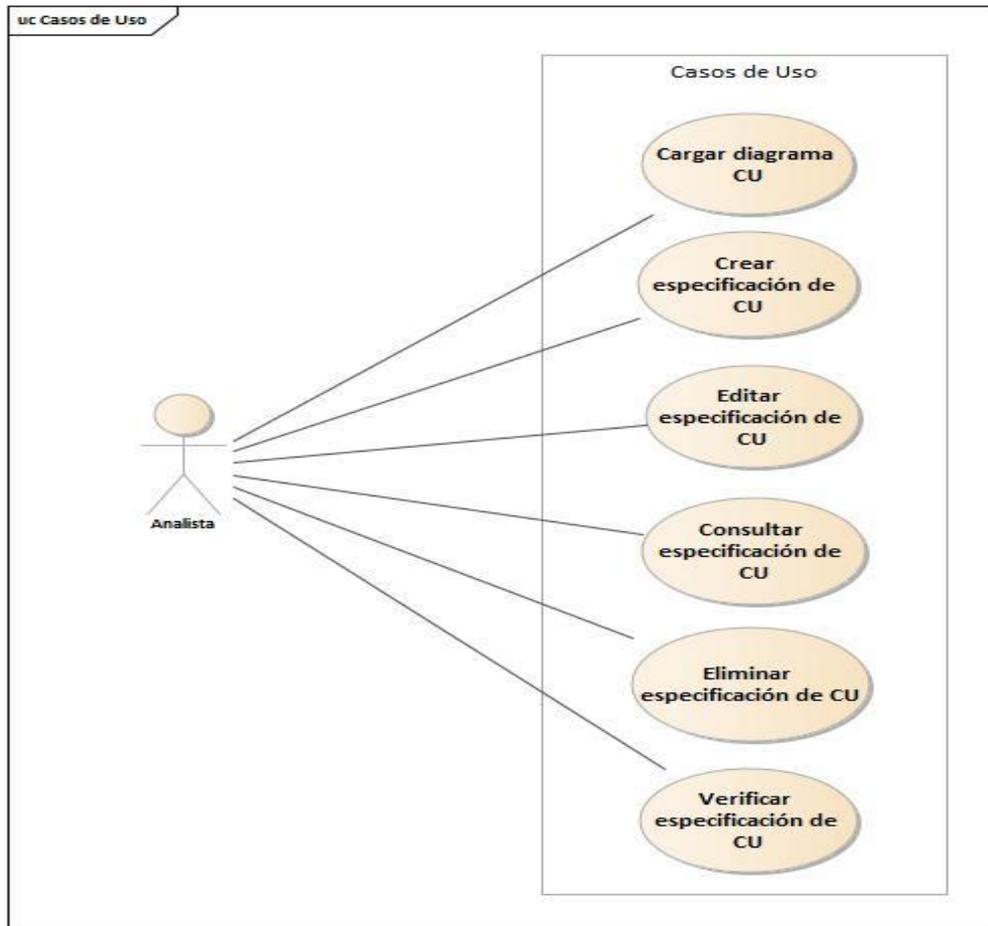


Ilustración 25. Diagrama de casos de uso – módulo casos de uso

e) Especificación de Casos de Uso

Se procedio a especificar las acciones que se establecio en los casos de uso para determinar el debido proceso que debe hacer cada actor.

Tipo de Artefacto: Formato de Especificación de casos de uso			
Nombre del sistema a revisar:			
FORMATO ESPECIFICACIÓN DE CASOS DE USO			
ID	1		Media
Nombre	Login		
Descripción	Permitir al usuario iniciar sesion con nombre de usuario y contraseña		
Fecha creación	6/05/2022	Fecha última modificación	6/05/2022
Actores	Usuario		
Precondicion es	Haberse registrado como un usuario del sistema MEVCU		
Poscondicion es	ninguna		
Flujo normal de eventos			
<Acciones>			
1. El actor ingresa nombre de usuario y contraseña			
2. El sistema permite acceder al actor			
Flujos alternos			
El usuario no esta registrado			
1. El actor ingresa nombre de usuario y contraseña			
2. El sistema no detecta al actor como un usuario del sistema			
El usuario digito mal la contraseña			
1. El actor ingresa nombre de usuario y contraseña			
2. El sistema le pide al actor que digite correctamente la contraseña			
El usuario digito mal la contraseña			
1. El actor ingresa nombre de usuario y contraseña			
2. El sistema le pide al actor que digite correctamente la contraseña			
El actor no lleno uno o ambos campos			
1. El actor no lleno el nombre de usuario, la contraseña o ninguno de los dos			
2. El sistema le pide que llene los campos correspondientes			
Excepciones			
o <Excepción: cuándo se presenta>			
<Acciones>			
...			

Ilustración 26.Especificación de caso de uso - módulo caso de uso

Tipo de Artefacto: Formato de Especificación de casos de uso			
Nombre del sistema a revisar:			
FORMATO ESPECIFICACIÓN DE CASOS DE USO			
ID	2		Prioridad Media
Nombre	Insertar Imagen		
Descripción	Permitir al usuario insertar una imagen en la elaboración del FER		
Fecha creación	6/05/2022	Fecha última modificación	6/05/2022
Actores	Usuario		
Precondición	Crear un nuevo proyecto		
Póscondicio	ninguna		
Flujo normal de eventos			
<Acciones>			
1. El actor crea un nuevo proyecto			
2. El actor inserta una imagen			
3. El sistema permite ver de forma asincrónica la imagen junto con el formato a diligenciar			
Flujos alternos			
El usuario no inserta la imagen			
1. El actor intenta seguir sin insertar la imagen			
2. El sistema no detecta la imagen y le pide al actor que inserte la imagen para poder seguir con el formulario			
Excepciones			
o <Excepción: cuándo se presenta>			
<Acciones>			
...			

Ilustración 27. Especificación de caso de uso - módulo caso de uso

f) Diagrama de Clases

Se ha representado la estructura de este módulo, para establecer sus clases, atributos, relaciones y operaciones, mediante el siguiente diagrama de clases, como se demuestra a través de la Ilustración 28

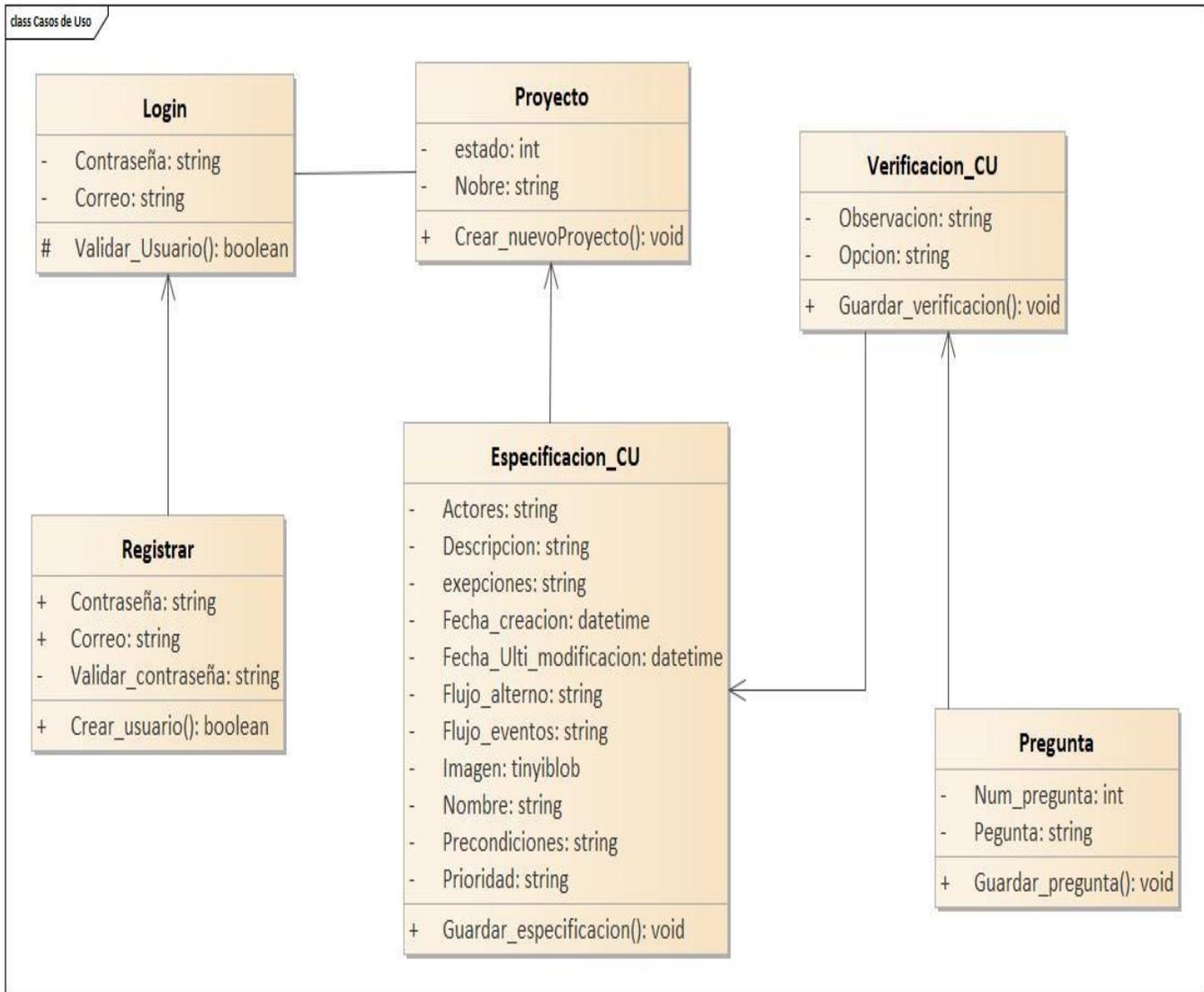
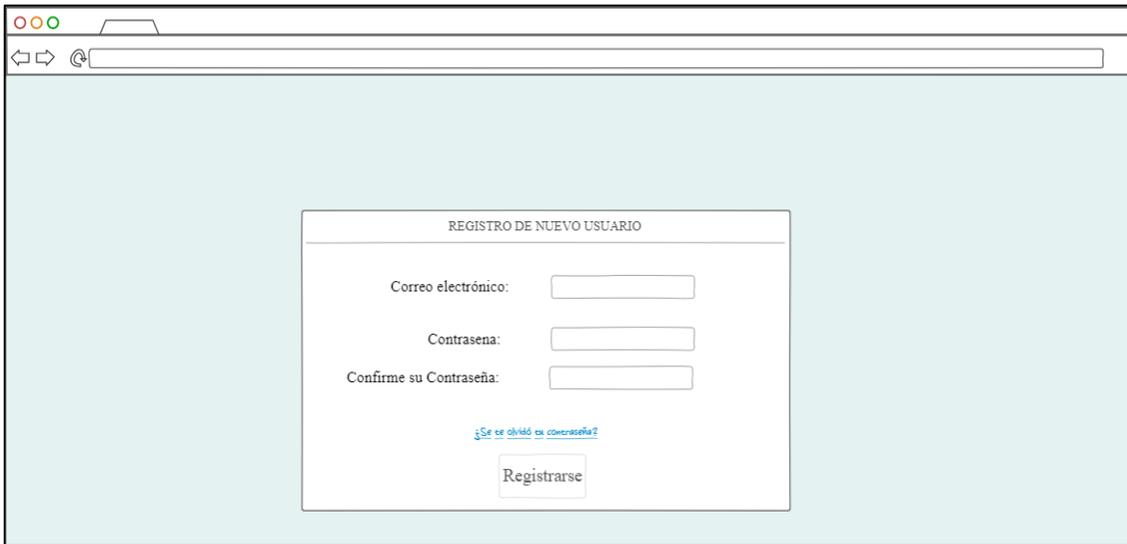


Ilustración 28. Diagrama de clases – módulo casos de uso

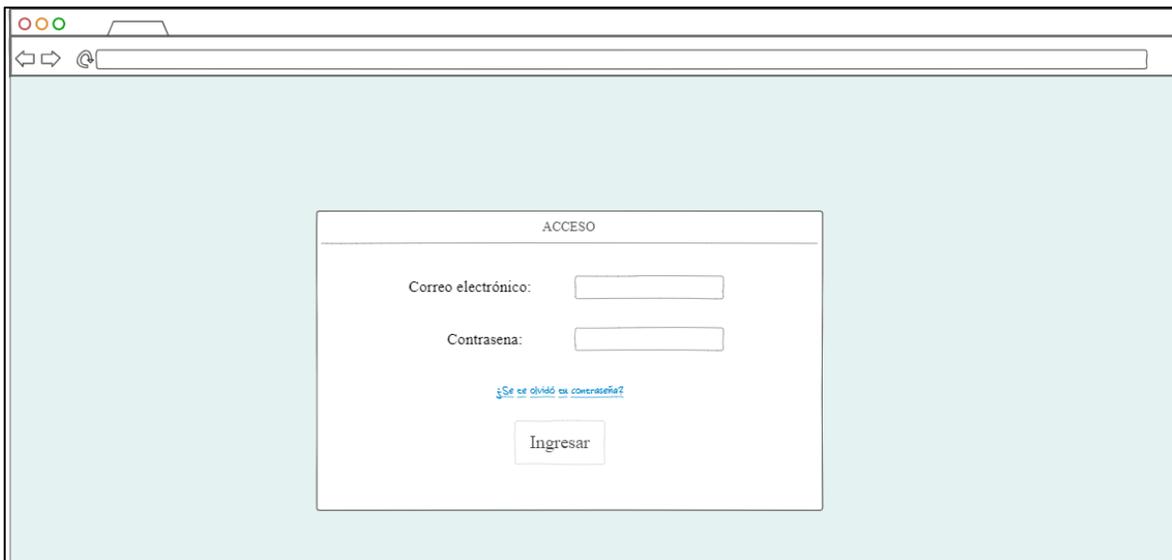
g) Mockups

Para tener una idea más clara y tener un modelo de cómo será el resultado final o esperado de este módulo, se ha diseñado el siguiente mockup.



The mockup shows a web browser window with a light blue background. In the center is a white rectangular form titled "REGISTRO DE NUEVO USUARIO". The form contains three input fields: "Correo electrónico:", "Contraseña:", and "Confirme su Contraseña:". Below the fields is a blue link that says "¿Se te olvidó tu contraseña?". At the bottom of the form is a button labeled "Registrarse".

Ilustración 29. Mockups registro de usuario - módulo casos de uso



The mockup shows a web browser window with a light blue background. In the center is a white rectangular form titled "ACCESO". The form contains two input fields: "Correo electrónico:" and "Contraseña:". Below the fields is a blue link that says "¿Se te olvidó tu contraseña?". At the bottom of the form is a button labeled "Ingresar".

Ilustración 30. Mockups inicio de sesión - módulo casos de uso

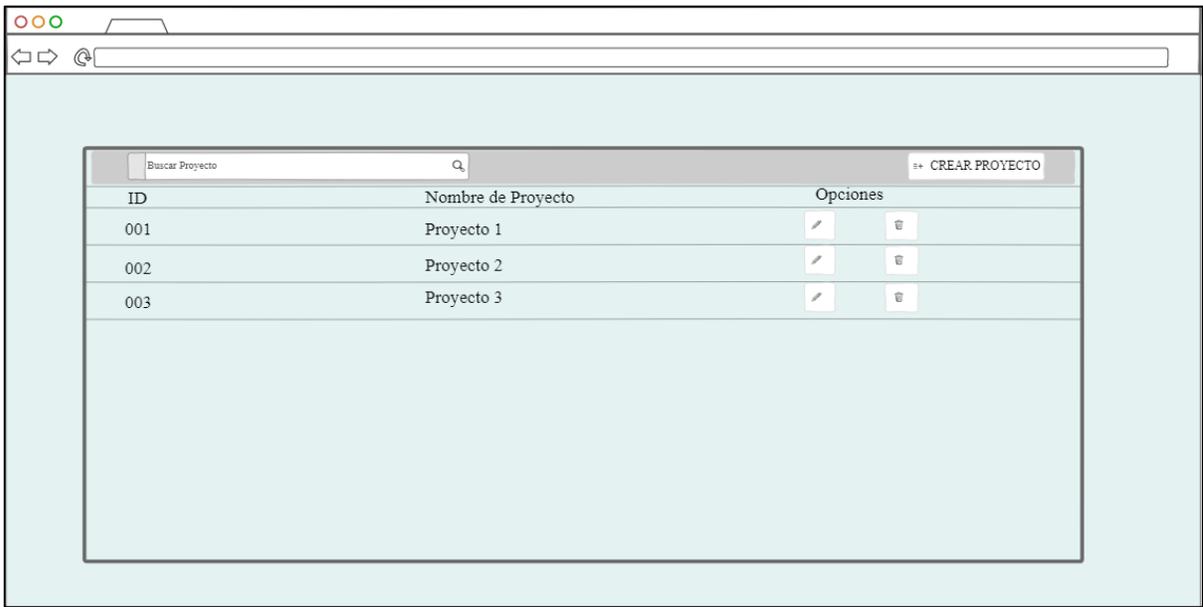


Ilustración 31. Mockups dashboard de usuario - módulo casos de uso

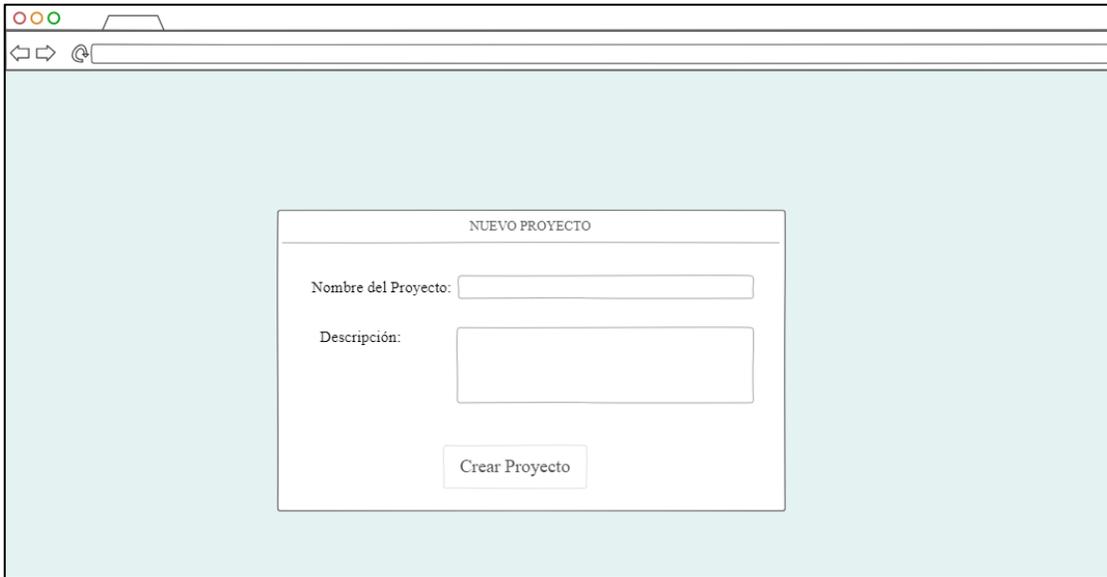


Ilustración 32. Mockups creación de nuevo proyecto- módulo casos de uso

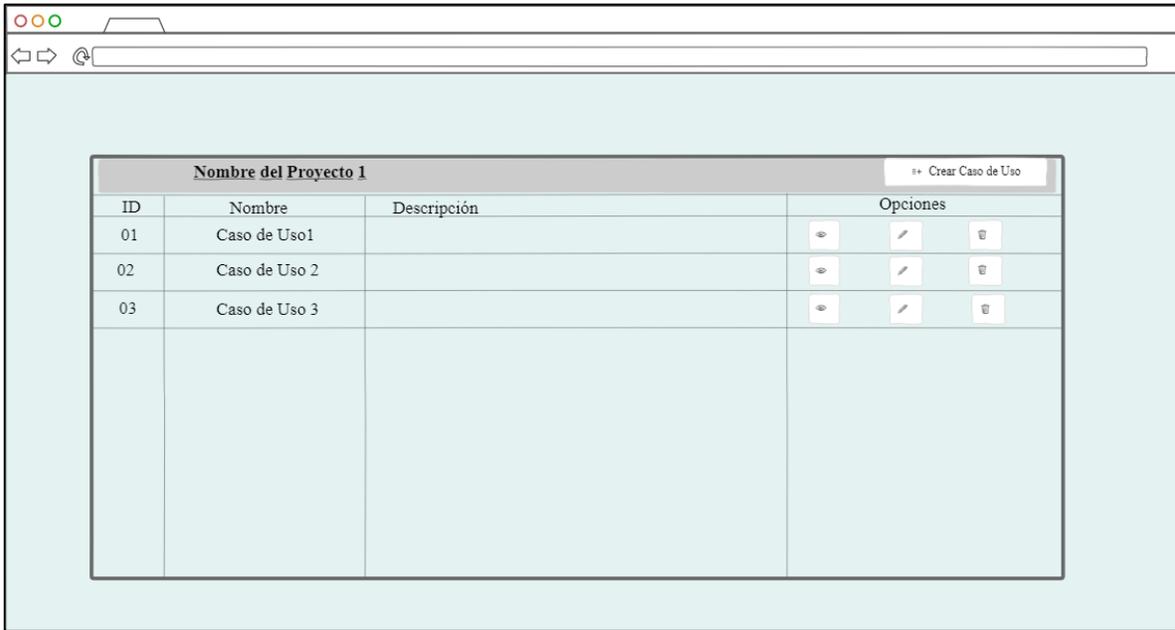


Ilustración 33. Mockups dashboard de proyecto - módulo casos de uso

FER

☆ Logotype Nombre de la empresa

Formulario De Especificación De Requisitos

ID	<input type="text" value=" <Identificador del caso de uso>"/>	Prioridad	<input type="text" value=" <Alta, Media o Baja>"/>
Nombre	<input type="text" value=" <Nombre descriptivo del caso de uso>"/>		
Descripción	<input type="text" value=" <Breve descripción del caso de uso, cuál es su funcionalidad>"/>		
Fecha creación	<input type="text" value=" <Fecha en la cual se creó la especificación>"/>	Fecha última modificación	<input type="text" value=" <Fecha en la cual se realizó algún cambio a la especificación>"/>
Actores	<input type="text" value=" <Lista de actores que pueden utilizar este caso de uso>"/>		
Precondiciones	<input type="text" value=" <Condiciones que deben cumplirse antes de ejecutar el caso de uso>"/>		
Poscondiciones	<input type="text" value=" <Condiciones que deben cumplirse después de ejecutar el caso de uso>"/>		
Flujo normal de eventos	<input type="text" value=" <Acciones>"/>		
	<ol style="list-style-type: none"> 1. El actor ... 2. El sistema ... 3. ... 		
Flujos alternos	<input type="text" value=" o <Nombre del flujo alterno>"/>		
	<input type="text" value=" <Acciones>"/>		
	...		
Excepciones	<input type="text" value=" o <Excepción: cuándo se presenta>"/>		
	<input type="text" value=" <Acciones>"/>		
	...		

Nombre imagen

Ilustración 34. Mockups formularios de especificación - módulo casos de uso

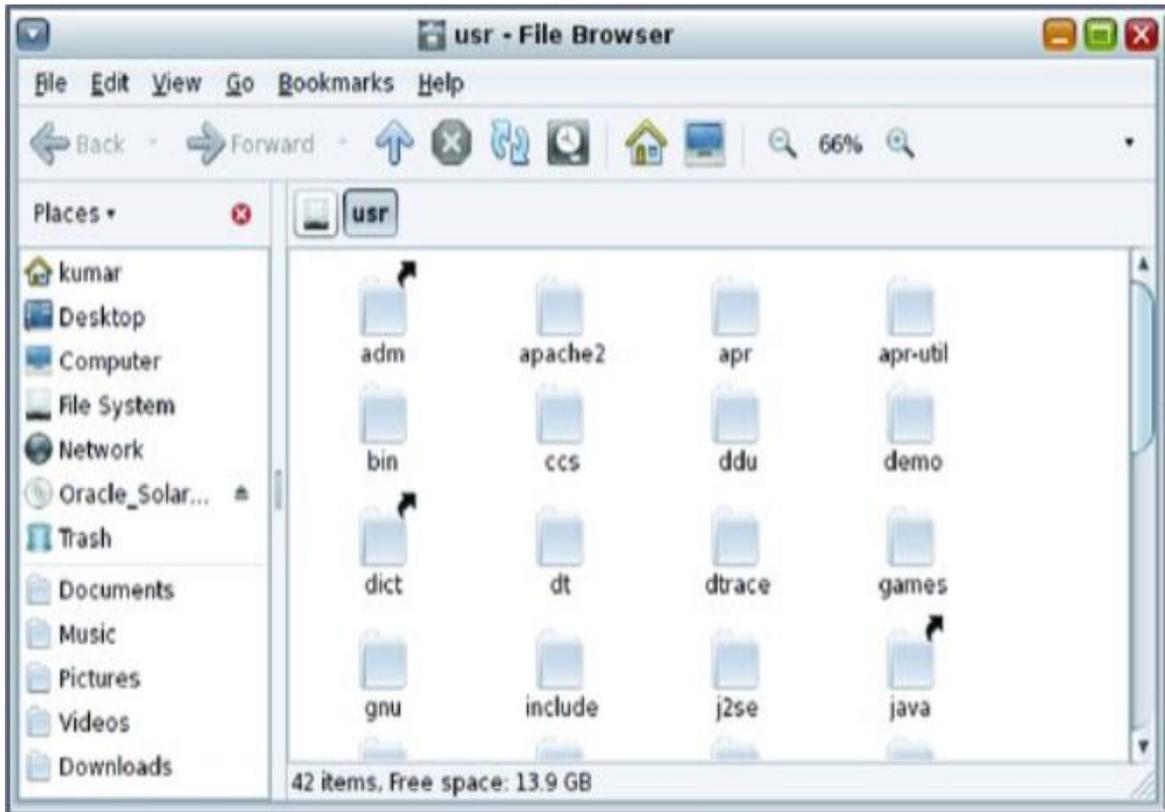


Ilustración 35. Mockups ventana emergente para adición de imagen - módulo casos de uso

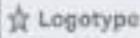
		Nombre de la empresa	
<input type="button" value="Volver"/>		Formulario De Verificación De Casos De Uso	
Correctos			
No.	PREGUNTAS	EVALUACIÓN	
		SI	NO
1	¿Cada caso de uso tiene un nombre significativo, que indica la funcionalidad que ofrece?		
2	¿Las pre-condiciones están expresadas como condiciones válidas, no como datos de entrada?		
3	¿Las pos-condiciones están expresadas como condiciones válidas, no como datos de salida?		
Completo			
4	¿Cada especificación de un caso de uso tiene toda la información que establece el estándar? (como mínimo: identificador, nombre, descripción, pre-condiciones, pos-condiciones, flujo normal de eventos y flujos alternos)		
5	¿Se definen claramente los datos de entrada que se necesitan en el caso de uso?		
6	¿Se define claramente la información de salida que debe proporcionar el sistema?		
7	¿Se definen claramente las validaciones que se deben realizar?		
8	¿Se definen claramente los cálculos o procedimientos que debe efectuar el sistema?		
9	¿Se han incluido los posibles errores o excepciones que se pueden presentar?		
10	¿En cada flujo alterno o excepción se indican concretamente las acciones que se deben realizar?		
11	En caso de tener referencias a otros documentos, ¿Están claramente identificados y pueden encontrarse fácilmente?		
Claros (No ambiguos)			
12	¿Cada caso de uso puede ser entendido por personal no técnico?		
13	¿Cada flujo alterno y excepción indica dónde se deriva del flujo normal y al finalizar las acciones, dónde continúa o si termina el caso de uso?		
14	¿Están identificadas claramente las acciones que realizan los actores y las que realiza el sistema?		
15	¿Cada especificación de un caso de uso está escrita sin errores gramaticales?		
16	¿Se hace claridad en las validaciones y no se dejan genéricas? (Por ejemplo, no se tiene algo como: "se valida que los datos sean correctos").		
Verificables			
17	¿Se puede verificar la implementación de cada caso de uso mediante pruebas, inspección, demostración o análisis?		
18	¿Todos los casos de uso se pueden implementar, con los recursos y las restricciones actuales?		
Otras consideraciones			
19	¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor?		
20	Si en el caso de uso interviene más de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?		
21	¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?		
22	El nombre del caso de uso no tiene verbos vagos o ambiguos, como son "hacer" o "procesar", en lugar de ello se deben utilizar verbos exactos: "aprobar", "notificar", "supervisar", "generar", etc.		
23	No hay casos de uso que describan varias acciones orientadas a bases de datos (crear, modificar, leer, eliminar), en un solo caso de uso.		
24	Cada caso de uso está relacionado por lo menos con un requisito funcional.		
<input type="button" value="Finalizar"/>			

Ilustración 36. Mockups formularios de verificación - módulo casos de uso

9.2 Módulo para Casos de Pruebas

Los casos de pruebas tienen demasiada importancia en un desarrollo de software, ya que estos son el conjunto de condiciones y restricciones las cuales avalan una aplicación o un sistema de software son aprobados o no, complaciendo las necesidades del cliente y los requerimientos específicos. Para la integración del módulo de casos de pruebas, se realizó el siguiente diseño estructural.

a) Requerimientos

Para la integración de este módulo se establecieron por parte del cliente los siguientes requerimientos, representados en la

Tabla 10

Tabla 10. Requerimientos – módulo casos de pruebas

ID	TIPO	REQUISITOS DE NEGOCIO
REQ-NEG-1	Objetivo de negocio.	Reducir el tiempo de diligenciamiento del caso de prueba.
REQ-NEG-2	Objetivo de negocio.	Sistematizar el diligenciamiento del caso de prueba.
REQ-NEG-3	Objetivo de negocio.	Facilitar el registro de los casos de prueba por medio de la implementación del sistema.
REQ-NEG-4	Regla de negocio.	La validación de un caso de prueba es exitosa cuando el resultado esperado es igual al resultado obtenido.

Los requerimientos solicitados para este módulo se han diligenciado en el sistema CHAMÍ para demostrar el uso funcional que tiene hasta la fecha

ID	Fecha	Proyecto	Prioridad	Tipo de Usuario	Medio de Recepción	Nombre	Versión	Estado
369	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	TIEMPO	1	RECIBIDO
371	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	SISTEMATIZACION	5	RECIBIDO
372	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	REGISTROS	4	RECIBIDO
373	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	VALIDACIÓN	0	RECIBIDO
374	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	LOGIN	0	RECIBIDO
376	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	OPERACIONES TESTER	1	RECIBIDO
377	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	OPERACIONES CLIENTE	0	RECIBIDO
378	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	LOGIN	0	RECIBIDO
380	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	INGRESAR DATOS	1	RECIBIDO
381	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	ACTUALIZAR DATOS	0	RECIBIDO
382	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	CONSULTAR DATOS	0	RECIBIDO
383	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	ELIMINAR DATOS	0	RECIBIDO
384	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	INICIAR SESIÓN	0	RECIBIDO
385	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	INGRESAR CP	0	RECIBIDO
386	2022-10-30	PROY. G2 - CASOS DE PRUEBA	MEDIA	GENERAL	OTRO	ACTUALIZAR CP	0	RECIBIDO

Ilustración 37. Requerimientos del módulo casos de pruebas implementados en el sistema Chamí

b) Diagramas de Actividades

Se representó los requerimientos mediante un diseño de diagrama de actividades reflejado en la siguiente Ilustración 38.

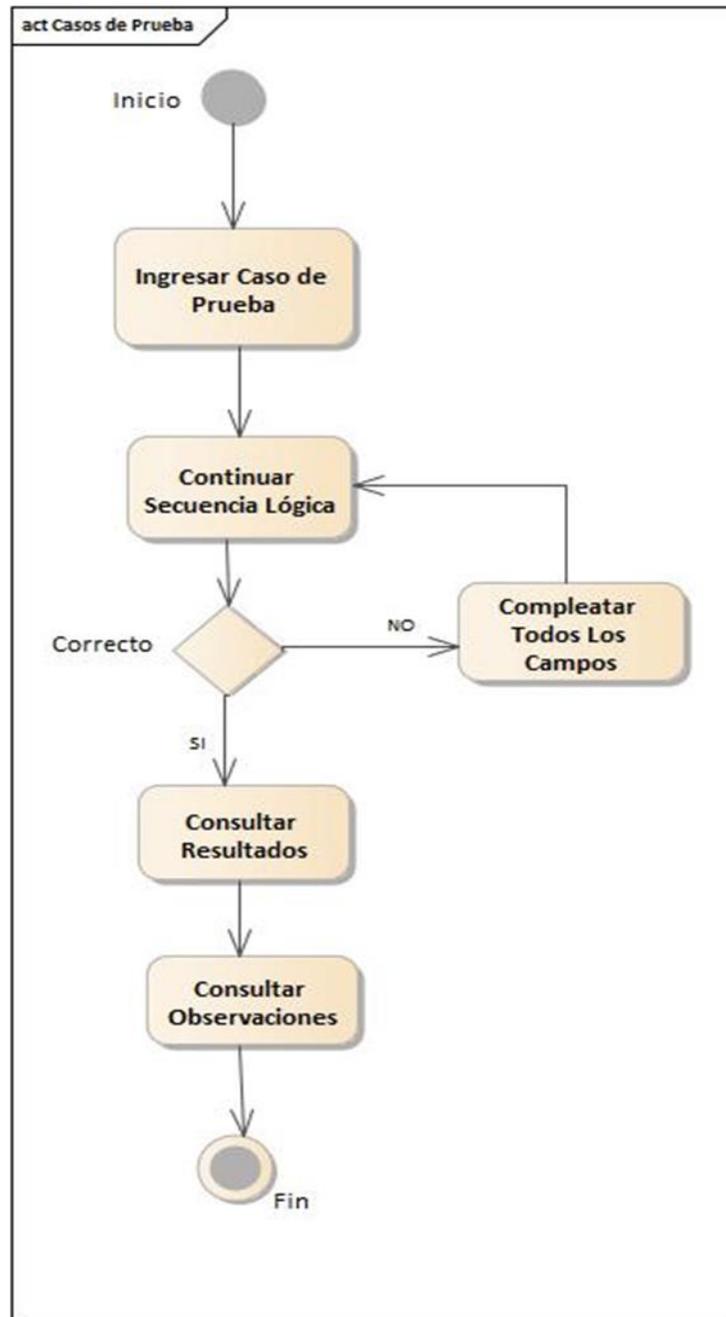


Ilustración 38. Diagrama de actividad caso de prueba – módulo casos de pruebas

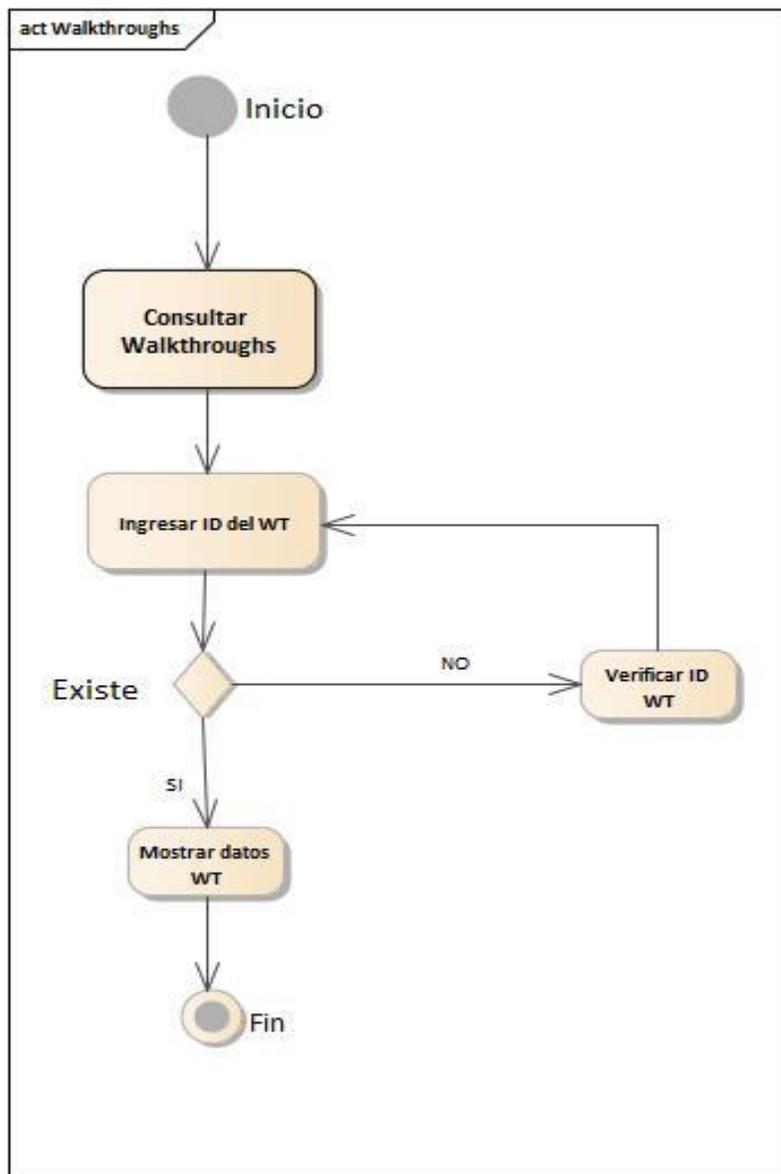


Ilustración 39. Diagrama de actividad walkthroughs- módulo casos de pruebas

c) Diagrama de Casos de Uso

Para determinar la actividad que le correspondía a cada uno de los actores dentro del sistema, se implementaron los requerimientos mediante un diseño de diagrama de casos de uso, reflejado en la siguiente Ilustración 40

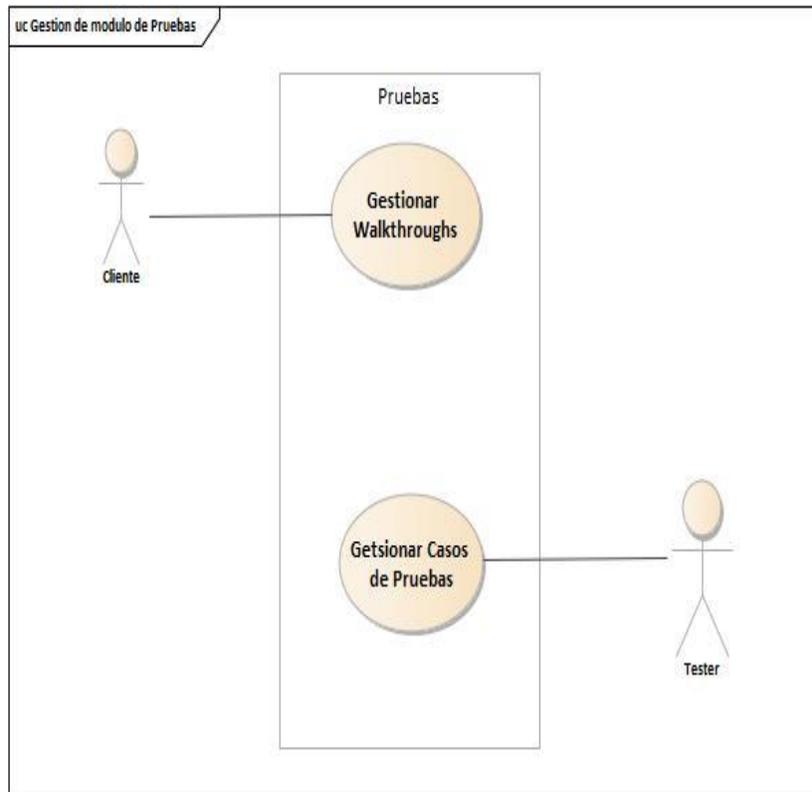


Ilustración 40. Diagrama de caso de uso gestión del módulo – módulo casos de pruebas

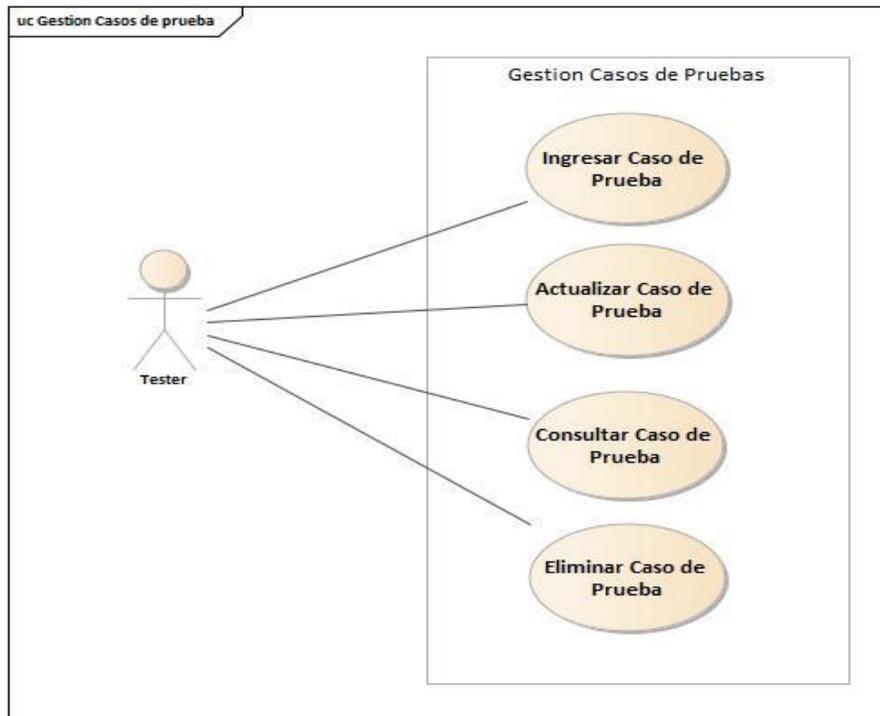


Ilustración 41.. Diagrama de caso de uso, gestión de casos de pruebas – módulo casos de pruebas

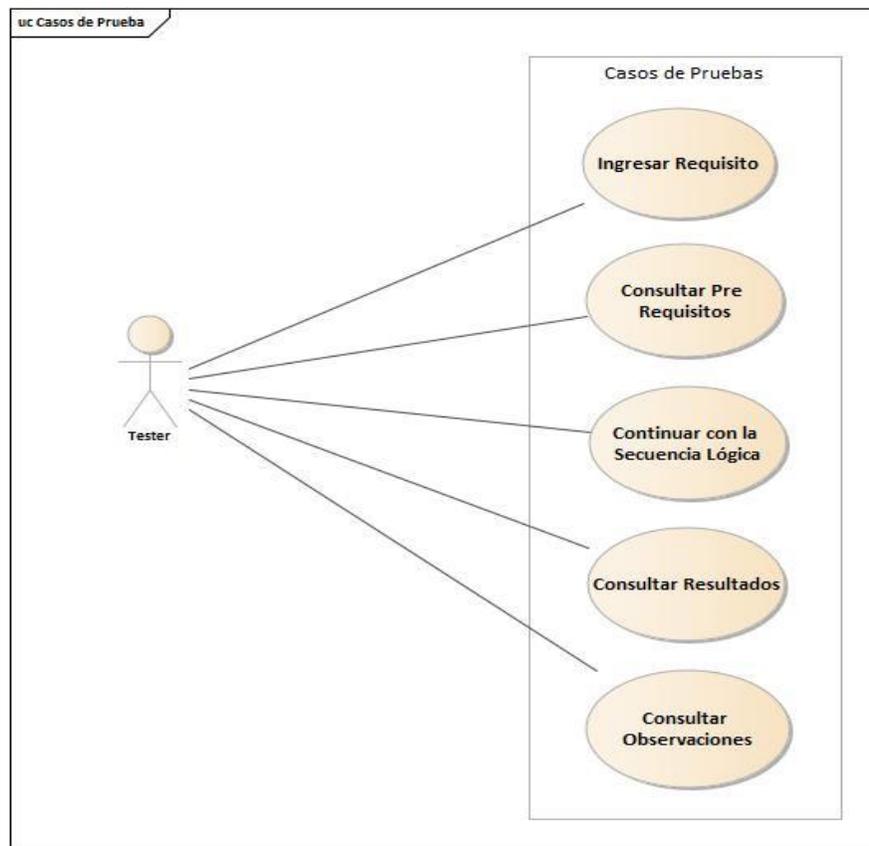


Ilustración 42. Diagrama de caso de uso, casos de pruebas – módulo casos de pruebas

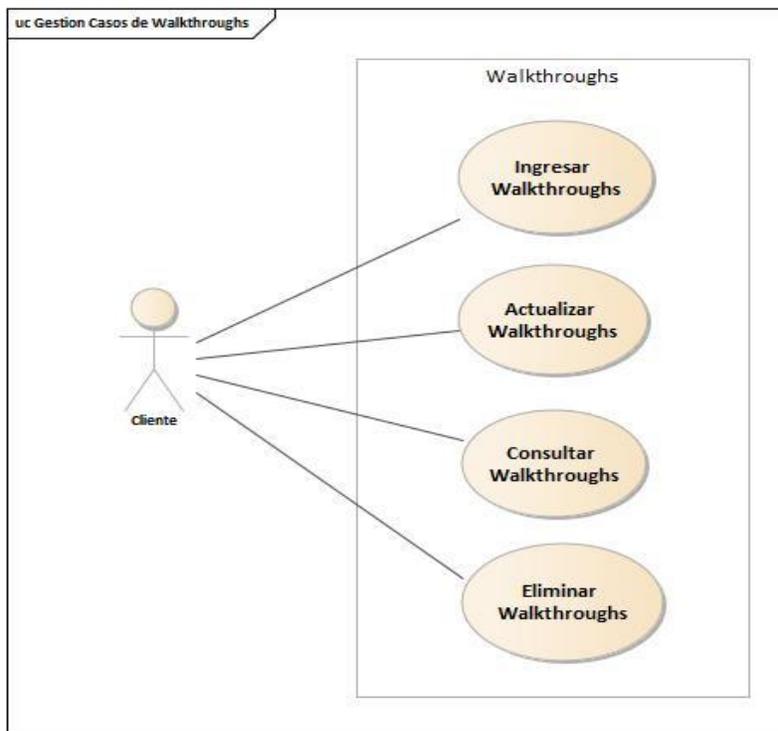


Ilustración 43. Diagrama de caso de uso, gestión de casos walkthroughs – módulo casos de pruebas

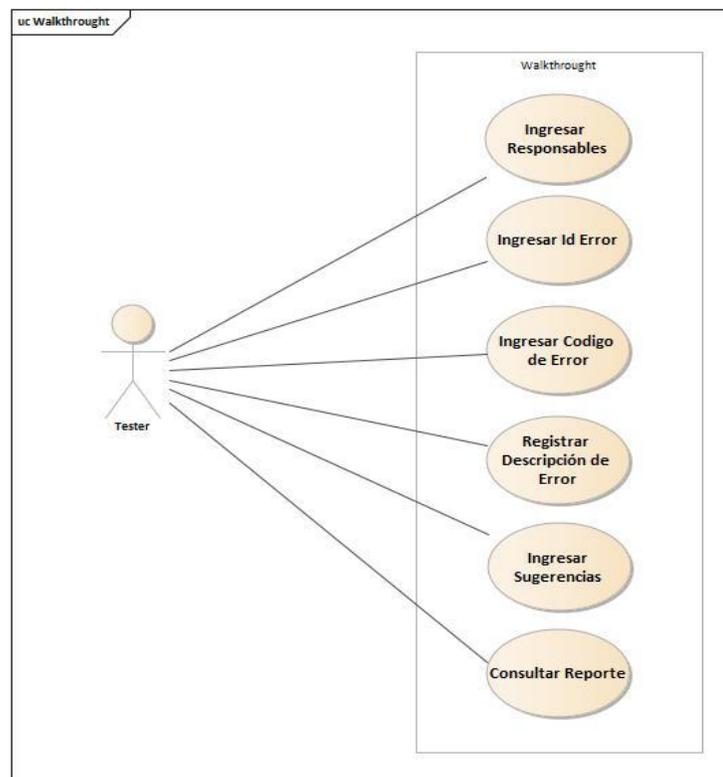


Ilustración 44. Diagrama de caso de uso, casos de walkthroughs – módulo casos de pruebas

d) Especificación de Casos de Uso

Se procedio a especificar las acciones que se establecio en los casos de uso para determinar el debido proceso que debe hacer cada actor.

Tabla 11.Especificación Inicio sesión CP

ID:	CU-001
Nombre:	Iniciar sesión.
Descripción:	Permite que el usuario ingrese al sistema con sus credenciales.
Actores:	Tester, Cliente.
Precondiciones:	Estar registrado en el sistema.
Postcondiciones:	Mostrar la pantalla de inicio según el rol del usuario que inició sesión.
Flujo normal de eventos	
<ol style="list-style-type: none">1. El actor ingresa su nombre de usuario.<ul style="list-style-type: none">• usuario2. El actor ingresa la contraseña.<ul style="list-style-type: none">• contraseña3. El sistema valida que se hayan ingresado todos los datos obligatorios.4. El actor da clic en el botón "iniciar sesión".5. El sistema valida los datos.6. El sistema inicia sesión.	
Flujos alternos	
* Actor no existe en el sistema o los datos son incorrectos.: a. En el paso 4, si el actor no tiene usuario registrado o ingreso datos incorrectos, el sistema muestra un mensaje "Datos incorrectos, si sigue presentando el mismo problema por favor colóquese en contacto con el administrador".	

Tabla 12.Especificación CU-002 – Ingresar CP

ID:	CU-002
Nombre:	Ingresar CP
Descripción:	Permite al actor ingresar un caso de prueba.
Actores:	Tester.
Precondiciones:	Iniciar sesión en el sistema.
Poscondiciones:	Se ha ingresado un caso de prueba.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor elige ingresar caso de prueba. 2. El sistema muestra el módulo correspondiente. 3. El actor diligencia los datos. <ul style="list-style-type: none"> • idCP • requisitoUsuario • estadoCP • prerrequisitos • pasos • resultadosEsp • resultadosObt • observaciones • nombreEjecutor • nombreRespProy 4. El sistema valida que se hayan ingresado todos los datos obligatorios. 5. El actor da clic en el botón enviar. 6. El sistema valida los datos. 7. El sistema muestra un mensaje "Realizado" 	
Flujos alternos	
<p>* Id de Caso de Prueba ya existe:</p> <ol style="list-style-type: none"> a. En el paso 6, si el Caso de Prueba ya existe en el sistema, se muestra un mensaje "El Caso de Prueba ya existe". 	

Tabla 13. Especificación CU-002 – Actualizar CP

ID:	CU-002
Nombre:	Actualizar CP
Descripción:	Permite al actor actualizar un caso de prueba.
Actores:	Tester.
Precondiciones:	Iniciar sesión en el sistema.
Poscondiciones:	Se ha actualizado un caso de prueba.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor elige actualizar caso de prueba. 2. El sistema muestra el módulo correspondiente. 3. El actor diligencia los datos. <ul style="list-style-type: none"> • idCP • requisitoUsuario • estadoCP • prerequisites • pasos • resultadosEsp • resultadosObt • observaciones • nombreEjecutor • nombreRespProy 4. El sistema valida que se hayan ingresado todos los datos obligatorios. 5. El actor da clic en el botón enviar. 6. El sistema valida los datos. 7. El sistema muestra un mensaje "Realizado" 	
Flujos alternos	
<p>* Id de Caso de Prueba no existe:</p> <ol style="list-style-type: none"> a. En el paso 6, si el Caso de Prueba no existe en el sistema, se muestra un mensaje "El Caso de Prueba no existe". 	

Tabla 14. Especificación CU-003 – Consultar CP

ID:	CU-002
Nombre:	Consultar CP
Descripción:	Permite al actor consultar un caso de prueba.
Actores:	Tester.
Precondiciones:	Iniciar sesión en el sistema.
Poscondiciones:	Se ha consultado un caso de prueba.
Flujo normal de eventos	
<ol style="list-style-type: none"> 1. El actor elige consultar caso de prueba. 2. El sistema muestra el módulo correspondiente. 3. El actor diligencia los datos. <ul style="list-style-type: none"> • idCP 4. El sistema valida que se hayan ingresado todos los datos obligatorios. 5. El actor da clic en el botón enviar. 6. El sistema valida los datos. 7. El sistema muestra la información solicitada. <ul style="list-style-type: none"> • idCP • requisitoUsuario • estadoCP • requisitos • pasos • resultadosEsp • resultadosObt • observaciones • nombreEjecutor • nombreRespProy 	
Flujos alternos	
<p>* Id de Caso de Prueba no existe:</p> <ol style="list-style-type: none"> a. En el paso 6, si el Caso de Prueba no existe en el sistema, se muestra un mensaje "El Caso de Prueba no existe". 	

e) Diagrama de Clases

Se ha representado la estructura de este módulo, para establecer sus clases, atributos, relaciones y operaciones, mediante el siguiente diagrama de clases, como se demuestra a través de la Ilustración 45

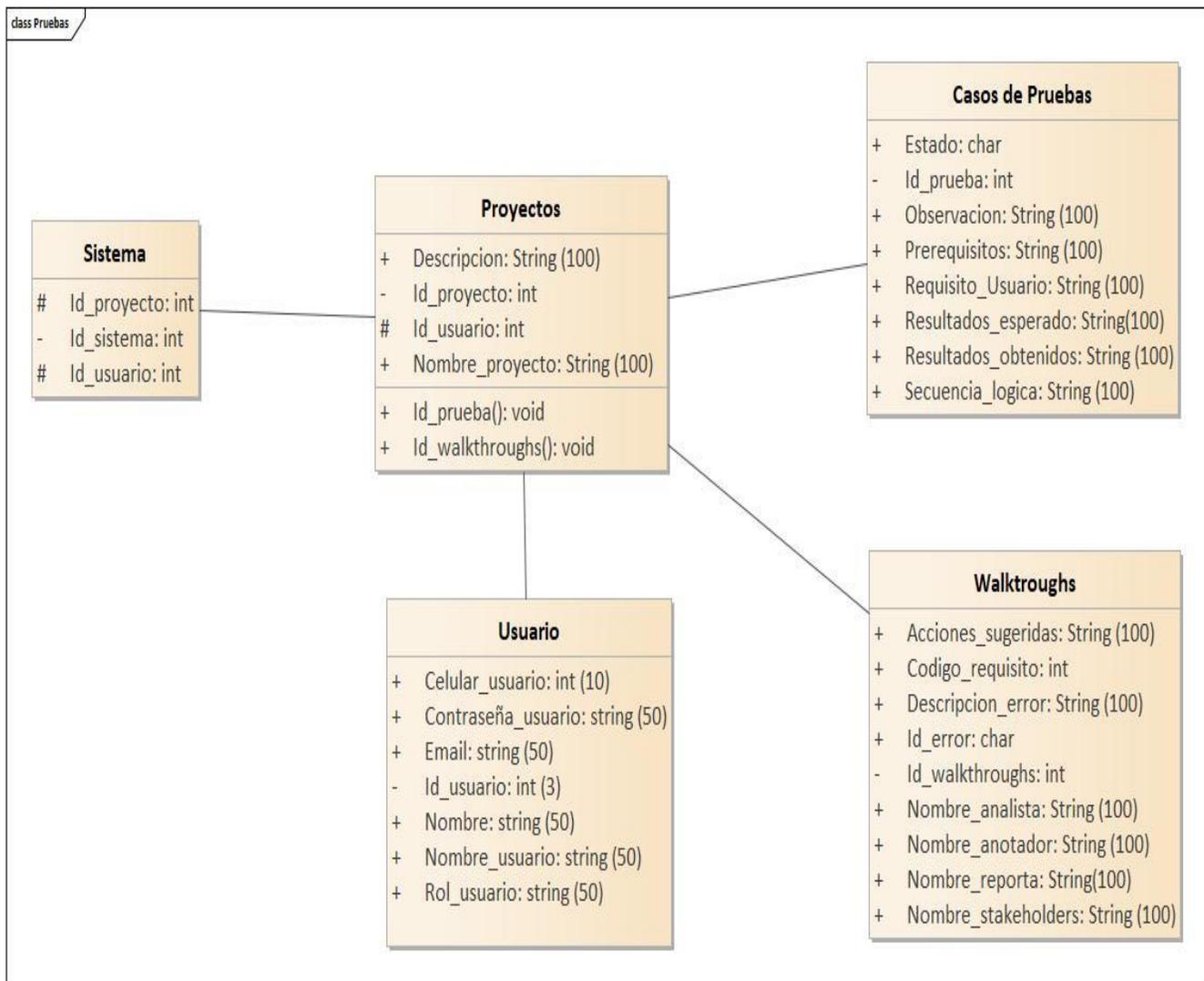


Ilustración 45. Diagrama de clase- módulo casos de pruebas

f) Mockups

Para tener una idea más clara y tener un modelo de cómo será el resultado final o esperado de este módulo, se ha diseñado el siguiente mockup.

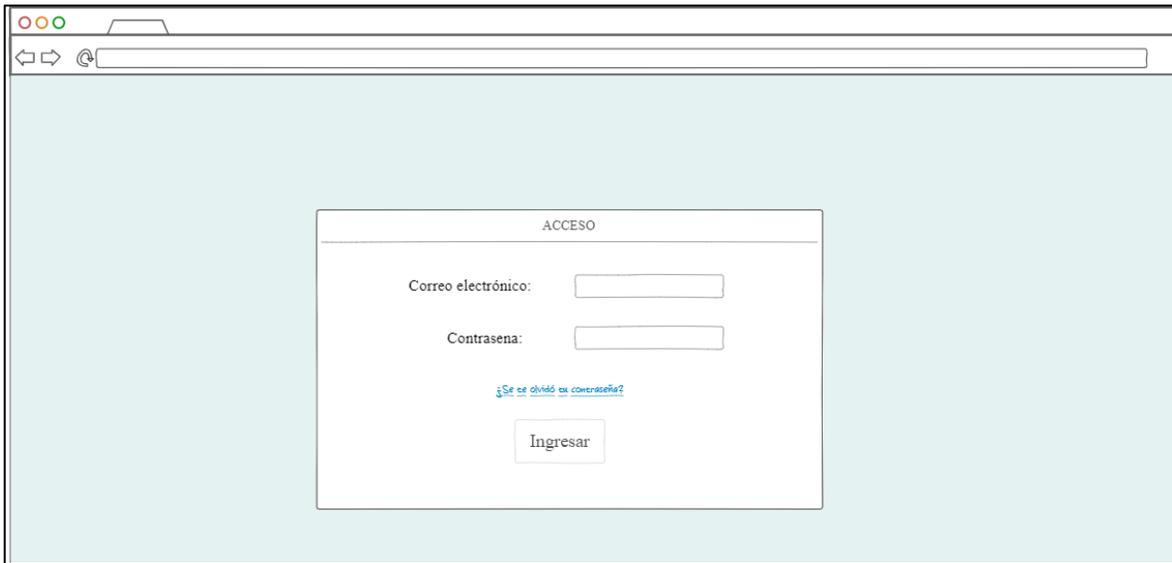


Ilustración 46. Mockup Login. - módulo casos de pruebas

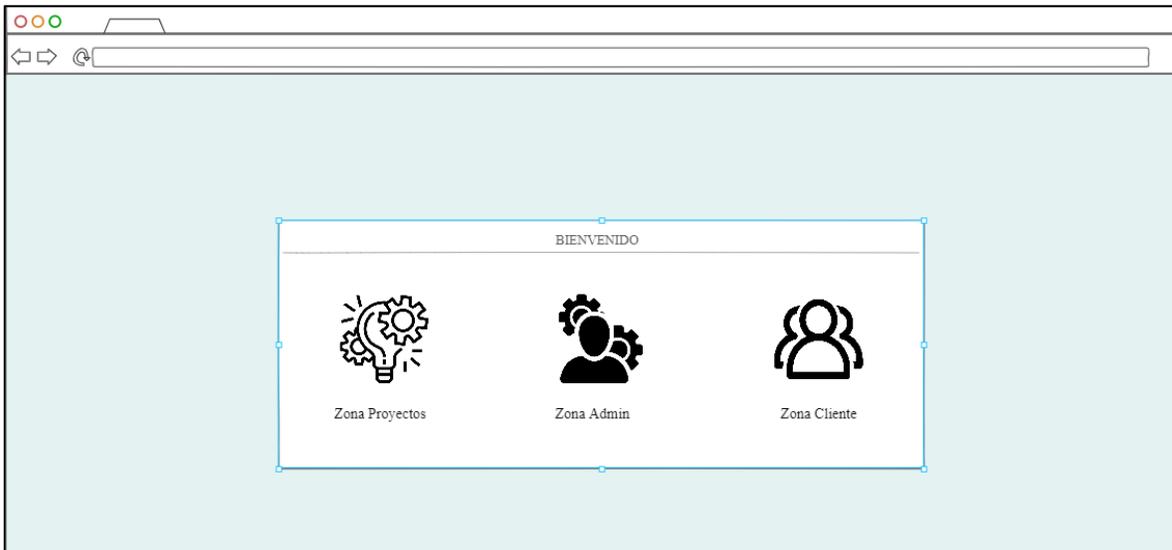


Ilustración 47. Mockup home. - módulo casos de pruebas

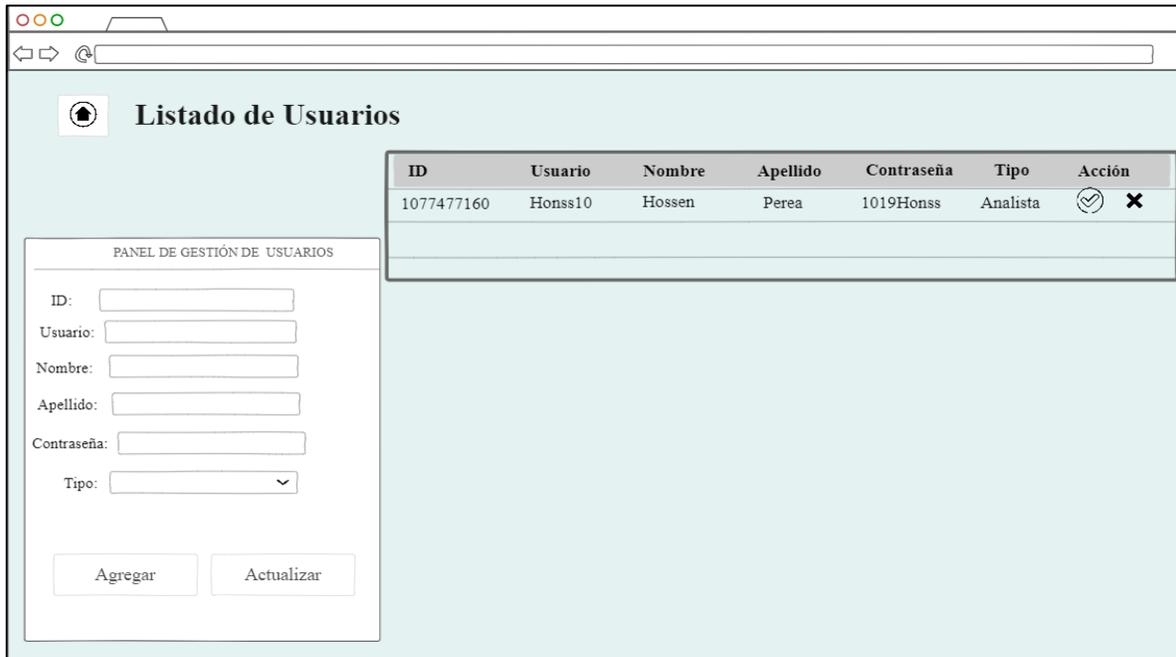


Ilustración 48. Mockup administrador. - módulo casos de pruebas

9.3 Módulo para Historias de Usuarios

Una historia de usuario es una explicación general e informal de la funcionalidad del software escrita desde el punto de vista del usuario final. Su propósito es articular cómo la funcionalidad del software ofrece valor a los clientes. Para la integración del módulo de historias de usuario, se realizó el siguiente diseño estructural.

a) Requerimientos

Para la integración de este módulo se establecieron por parte del cliente los siguientes requerimientos, representados en la Tabla 15

Tabla 15. Requerimientos – módulo historias de usuarios

ID	Tipo	Requisitos de negocio
REQ-NEG-1	Objetivo de negocio	Aumentar el nivel de satisfacción del usuario que accede a los servicios de creación, actualización, eliminación, búsqueda y verificación de historias de usuario de acuerdo con la siguiente fórmula: Nivel de Satisfacción = Rendimiento Percibido del Sistema - Expectativas.
REQ-NEG-2	Objetivo de negocio	Ampliar la capacidad de la base de datos para soportar un volumen más grande de Historias de Usuario con su Verificación correspondiente.
REQ-NEG-3	Reglas de negocio	Todas las Historias realizadas por los usuarios deben ser verificadas dentro de las siguientes 24 horas después de su radicación.

Los requerimientos solicitados para este módulo se han diligenciado en el sistema CHAMÍ para demostrar el uso funcional que tiene hasta la fecha.

ID	Fecha	Proyecto	Prioridad	Tipo de Usuario	Medio de Recepción	Nombre	Versión	Estado
392	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	SATISFACCIÓN DE USUARIO	0	RECIBIDO
393	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	CAPACIDAD	0	RECIBIDO
394	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	VERIFICACIÓN	0	RECIBIDO
395	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	ENVÍO DE DATOS	0	RECIBIDO
396	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	ESTADO	0	RECIBIDO
397	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	CORRECCIONES	0	RECIBIDO
398	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	TIEMPO	0	RECIBIDO
399	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	INGRESAR DATOS	0	RECIBIDO
400	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	VERIFICAR DATOS	0	RECIBIDO
401	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	ACTUALIZAR DATOS	0	RECIBIDO
402	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	BUSCAR DATOS	0	RECIBIDO
403	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	ELIMINAR DATOS	0	RECIBIDO
404	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	LISTADO	0	RECIBIDO
405	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	INTERFAZ	0	RECIBIDO
406	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	TAMAÑO	0	RECIBIDO
407	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	USABILIDAD	0	RECIBIDO
408	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	ACCESIBILIDAD	0	RECIBIDO
409	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	USABILIDAD	0	RECIBIDO
410	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	PORTABILIDAD	0	RECIBIDO
411	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	RESTRICCIÓN	0	RECIBIDO
412	2022-10-30	PROY. G4 - HISTORIAS DE USUARIO	MEDIA	GENERAL	OTRO	RESTRICCIÓN	0	RECIBIDO

Mostrando registros del 1 al 21 de un total de 21 registros

Ilustración 49. Requerimientos del módulo historias de usuarios implementados en el sistema Chamí

b) Diagrama de Actividades

Se representó los requerimientos mediante un diseño de diagrama de actividades reflejado en la siguiente Ilustración 50

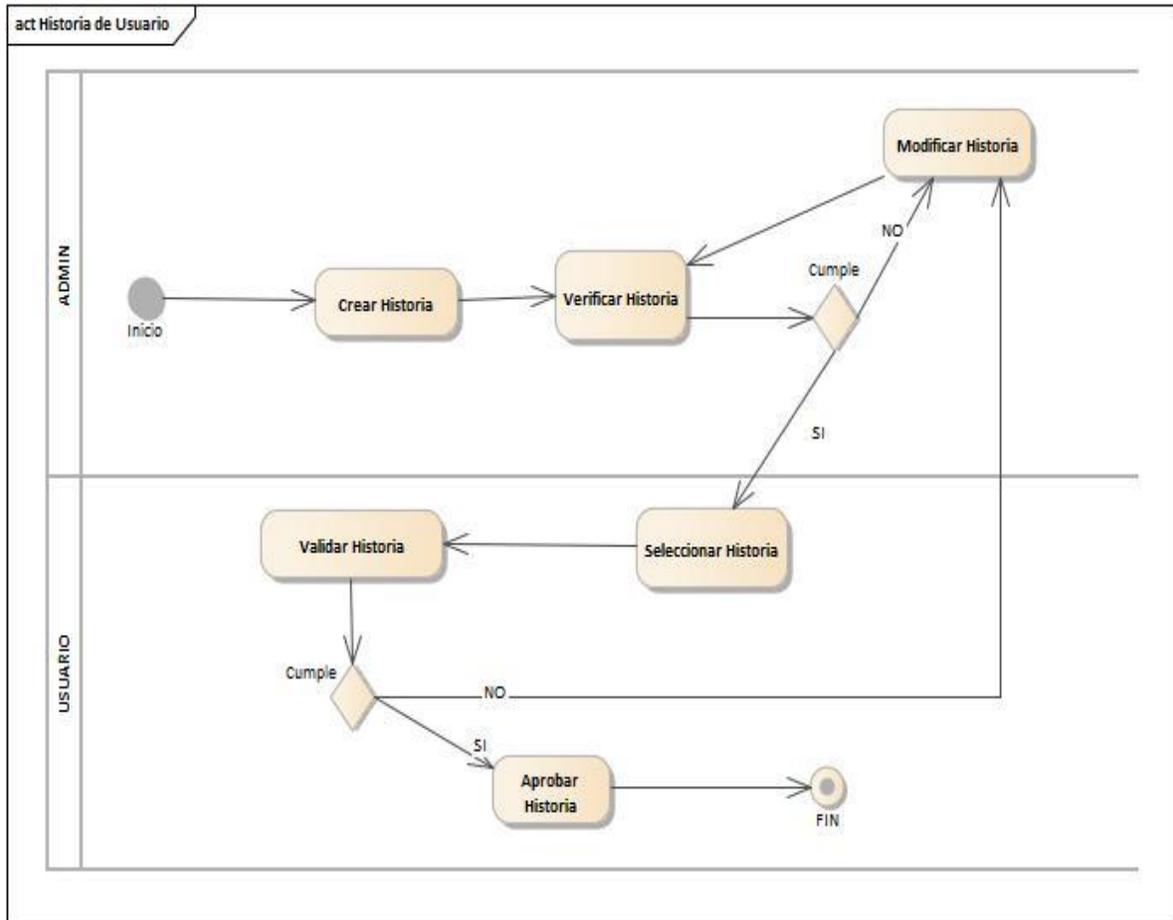


Ilustración 50. Diagrama de actividad – módulo historias de usuarios

c) Diagrama de Casos de Uso

Para determinar la actividad que le correspondía a cada uno de los actores dentro del sistema, se implementaron los requerimientos mediante un diseño de diagrama de casos de uso, reflejado en la siguiente Ilustración 51

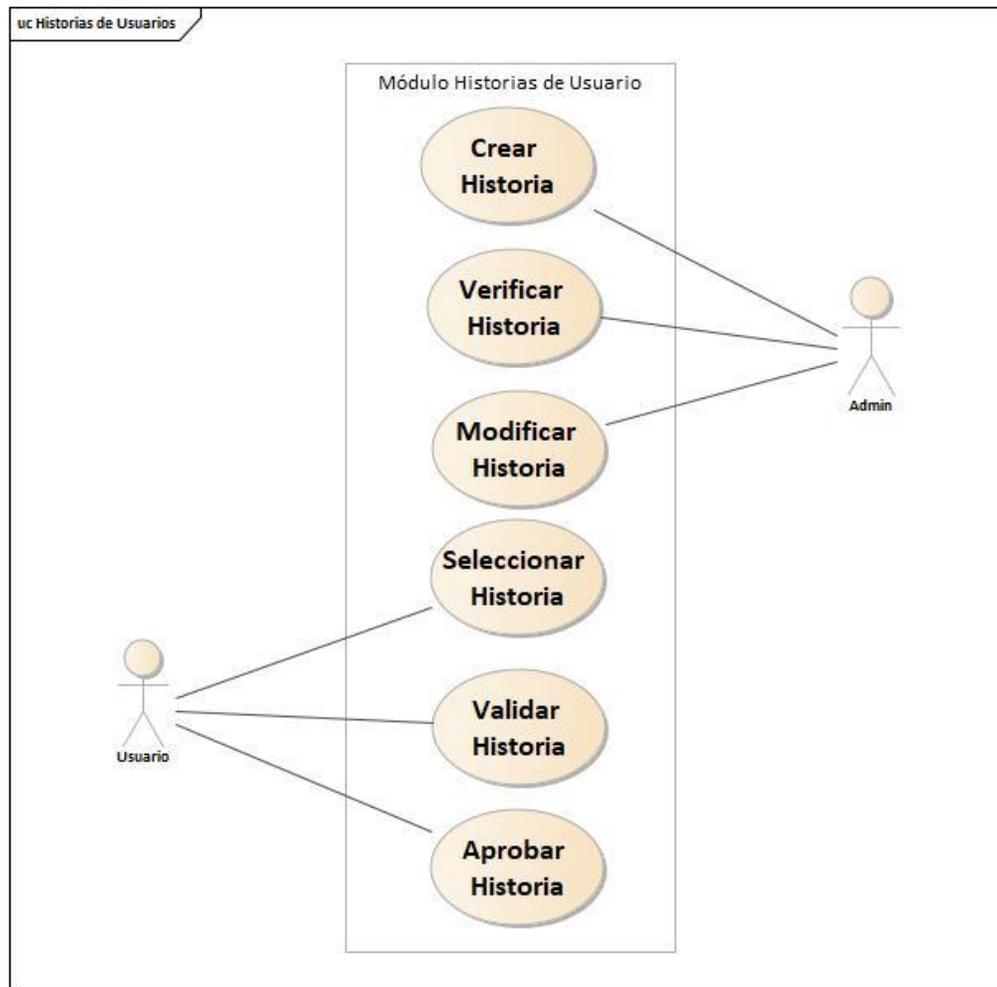


Ilustración 51. Diagrama de casos de uso – módulo historias de usuarios

d) Especificación de Casos de Uso

Se procedio a especificar las acciones que se establecio en los casos de uso para determinar el debido proceso que debe hacer cada actor.

ID	<CU1001>	Prioridad	<Alta>
Nombre	<Verificar Historia de Usuario>		
Descripción	<Breve descripción del caso de uso, cuál es su funcionalidad>		
Autor	<Hansen Arenas>		
Fecha creación	<5/05/2022>	Fecha última modificación	<5/05/2022 >
Fuente(s)	<Hansen Arenas>		
Actores	<Usuario, Administrador>		
Precondiciones	<El actor debe haber iniciado sesión>		
Poscondiciones	<Se debe cargar la consulta y ver su estado en el apartado de revisiones>		
Flujo normal de eventos			
<Acciones>			
<ol style="list-style-type: none"> 1. El actor ingresa al apartado de HU que están aún por verificar 2. El sistema lo conduce a el módulo de verificación 3. El actor procede a validar cada respuesta de la HU 4. El sistema va almacenando las validaciones hechas por el actor 5. El actor termina y da clic en el botón de "Guardar Verificación" 6. El sistema valida que todas las respuestas hayan sido validadas y guarda la verificación en la base de datos 7. El sistema presenta un mensaje de que la Verificación ha sido registrada con éxito 8. El usuario da clic en Aceptar 9. El sistema regresa al actor al módulo de verificación 			
Flujos alternos			
<ul style="list-style-type: none"> o <Respuestas sin validación> <ul style="list-style-type: none"> <El sistema informa al actor mediante un mensaje, que se deben validar todas las respuestas> <El actor da clic en aceptar y vuelve a la HU> 			
...			
Excepciones			
<ul style="list-style-type: none"> o <Excepción: cuándo no se puede registrar la verificación> <ul style="list-style-type: none"> <El sistema debe presentar un mensaje diciendo que la verificación no se ha podido guardar, por favor vuelva a intentarlo (Regresa al paso 5 del flujo normal de eventos)> 			
...			
Criterios de aceptación	<Que se registre una nueva verificación en la base de datos con el número de HU correspondiente>		
Objetivos relacionados	<El objetivo es que el actor pueda verificar una HU enviada al sistema y guardar dicha verificación>		
Reglas del negocio	<Listado de reglas del negocio que aplican para este caso de uso>		
Requerimientos No Funcionales	<Interfaz responsive con rápido tiempo de respuesta>		
Anotaciones			

Ilustración 52. Especificación para verificar historia de usuario - módulo historias de usuarios

ID	Id_caso_uso1	Prioridad	<Alta>
Nombre	Consultar estado de la verificación		
Descripción	El usuario puede consultar en qué estado está la verificación del registro		
Autor	Juan Pablo Devos y documento de especificación de requisitos UCP		
Fecha creación	05/05/2022	Fecha última modificación	05/05/2022
Fuente(s)	Juan Pablo Devos		
Actores	<ul style="list-style-type: none"> • Usuario • Administrador 		
Precondiciones	Antes de ejecutar este caso de uso en la base de datos debe de estar ya el registro guardado para cuando el usuario digite el número de historia coincida con lo guardado		
Poscondiciones	El sistema debe responder a la consulta con los siguientes datos <ul style="list-style-type: none"> • Numero de historia • Estado • Ver/editar 		
Flujo normal de eventos			
<Acciones>			
<ol style="list-style-type: none"> 1. El actor usuario o administrador pulsa el botón CONSULTAR en el menú de navegación 2. El sistema entra al interfaz de consulta 3. El actor consultara el registro mediante el número de historia y pulsa el botón BUSCAR 4. El sistema mostrara el estado de la HU correspondiente al número de historia ingresado 			
Fujos alternos: Si no se ingresa un numero de historia de usuario valido, el sistema mostrara un mensaje informado que por favor digite un numero de historia usuario			
o			
o Excepciones Si el sistema no detecta que el número de historia con los que hay en la base de datos entonces se mostrará un mensaje de texto donde informe que el número de historia no fue encontrada, el sistema deberá devolverse al menú de navegación			
Criterios de aceptación	Si el sistema permite hacer la búsqueda mediante el número de historia y si permite ver y editar la historia de usuario se asume como cumplido el caso de uso		
Objetivos relacionados	El objetivo principal de este apartado es que el actor usuario o administrador puedan ver en qué estado en tiempo real se encuentra la verificación de historia de usuario		
Reglas del negocio			
Requerimientos No Funcionales	Interfaz responsive con rápido tiempo de respuesta		
Anotaciones			

Ilustración 53. Especificación para consultar el estado - módulo historias de usuarios

ID	<CU1000>	Prioridad	<Alta>
Nombre	<Login>		
Descripción	<Se debe poder acceder al sistema haciendo uso de un usuario y una contraseña para manejar cualquier opción dentro de este>		
Autor	<Hansen Arenas>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	<Hansen Arenas>		
Actores	<Administrador, Usuario>		
Precondiciones	<Debe haberse establecido un usuario y contraseña para el administrador o el usuario debe haberse registrado con anterioridad>		
Poscondiciones	<Se debe permitir el acceso al módulo principal del sistema>		
Flujo normal de eventos			
<Acciones>			
1. El actor ingresa al sistema de manera normal			
2. El sistema presenta pantalla de Login solicitando usuario y contraseña antes de permitir el ingreso			
3. El actor completa ambos campos y da clic en INGRESAR			
4. El sistema valida la información y dirige al actor al módulo principal			
Flujos alternos			
o <Contraseña Incorrecta> <El sistema informa al actor mediante un mensaje en pantalla que la contraseña ingresada no es la correcta, por favor vuelva a intentarlo (Regresa al paso 3 del flujo normal de eventos)>			
...			
Excepciones			
o <Excepción: Usuario no existente> <El sistema debe presentar un mensaje informando al actor que no existe un registro con el usuario ingresado, por favor vuelva a intentarlo (Regresa al paso 3 del flujo normal de eventos)>			
...			
Criterios de aceptación	<Se da paso al menú principal del sistema justamente después del login>		
Objetivos relacionados	<El objetivo es que el actor pueda ingresar y hacer uso del sistema siempre y cuando cumpla con este requisito>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 54. Especificación para login - módulo historias de usuarios

ID	<CU1001>	Prioridad	<Alta>
Nombre	<Buscar Historia de Usuario>		
Descripción	<Se deben poder hacer las respectivas consultas de cada campo accediendo mediante el código de historia, presentando la información con la cual fue diligenciada.>		
Autor	<Hansen Arenas>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	<Hansen Arenas>		
Actores	<Administrador, Usuario>		
Precondiciones	<El actor debe haber iniciado sesión>		
Poscondiciones	<Se debe presentar nueva pantalla mostrando la información solicitada>		
Flujo normal de eventos			
<Acciones>			
5. El actor ingresa al módulo de Gestión de H.U			
6. El sistema dirige al actor a la vista correspondiente			
7. El actor da clic en Buscar Historia de Usuario			
8. El sistema presenta la pantalla correspondiente y solicita el código que desea consultar			
9. El actor digita el código indicado y da clic en CONSULTAR			
10. El sistema valida la información recibida y presenta los datos relacionados a tal código			
Flujos alternos			
o <Código no digitado>			
<El sistema informa al actor mediante un mensaje en pantalla que por favor ingrese un código válido para proceder con la consulta (Regresa al paso 5 del flujo normal)>			
...			
Excepciones			
o <Excepción: Código no existente>			
<El sistema debe presentar un mensaje informando al actor que no existe un registro con el código digitado previamente, por favor vuelva a intentarlo (Regresa al paso 5 del flujo normal de eventos)>			
...			
Criterios de aceptación	<Se presenta en pantalla los datos encontrados que se relacionan con el código de la búsqueda>		
Objetivos relacionados	<El objetivo es que el actor pueda visualizar la información correspondiente de una Historia de Usuario solicitada >		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 55. Especificación para buscar historia de usuario - módulo historias de usuarios

ID	<CU1002>	Prioridad	<Alta>
Nombre	<Crear HU>		
Descripción	<Se deben poder almacenar nuevos registros de HU en el sistema, para ello se solicita ingresar la información esencial correspondiente descrita en los requisitos>		
Autor	<Hansen Arenas>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	<Hansen Arenas>		
Actores	<Administrador, Usuario>		
Precondiciones	<El actor debe haber iniciado sesión y debe haber un proyecto creado por ese mismo actor>		
Poscondiciones	<Se debe generar en un nuevo espacio en memoria con el registro establecido por el actor>		
Flujo normal de eventos			
<Acciones>			
11. El actor ingresa al módulo de Proyectos y selecciona donde desea crear su nueva Historia			
12. El sistema dirige al actor a la vista correspondiente			
13. El actor da clic en Crear Nueva Historia de Usuario			
14. El sistema presenta el formato correspondiente a rellenar para proceder con la creación			
15. El actor completa cada uno de los campos presentados en pantalla y da clic en GUARDAR			
16. El sistema valida la información y presenta un mensaje en pantalla diciendo que el registro se ha efectuado con éxito			
17. El sistema carga la información nueva en el archivo			
18. El actor da clic en OK			
19. El sistema dirige al actor al módulo del Proyecto			
Flujos alternos			
o <Falta de Información> <El sistema informa al actor mediante un mensaje en pantalla que por favor complete todos los campos antes de continuar con el registro (Regresa al paso 5 del flujo normal de eventos)>			
...			
Excepciones			
o <Excepción: Código ya existente> <El sistema debe presentar un mensaje informando al actor que ya existe un registro con el mismo código digitado (Regresa al paso 5 del flujo normal de eventos)>			
...			
Criterios de aceptación	<Se genera una nueva Historia de Usuario en el archivo correspondiente dentro del sistema>		
Objetivos relacionados	<El objetivo es que el actor pueda crear nuevos registros en el sistema haciendo uso de este módulo>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 56. Especificación para crear historia de usuario - módulo historias de usuarios

ID	<CU1003>	Prioridad	<Alta>
Nombre	<Actualizar Historia de Usuario>		
Descripción	<Se deben poder BUSCAR la información de alguna Historia mediante su código y tener la posibilidad de modificar sus datos>		
Autor	<Hansen Arenas>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	<Juan Camilo Campos>		
Actores	<Administrador, Usuario>		
Precondiciones	<El actor debe haber iniciado sesión>		
Poscondiciones	<Se deben guardar los cambios efectuados por el actor con la nueva información digitada>		
Flujo normal de eventos			
<Acciones>			
20. El actor ingresa al módulo de Gestionar HU			
21. El sistema dirige al actor a la vista correspondiente			
22. El actor da clic en Actualizar Historia			
23. El sistema presenta la pantalla correspondiente y solicita el código que la historia que desea modificar			
24. El actor digita el código indicado y da clic en BUSCAR			
25. El sistema valida la información recibida y presenta la vista mostrando la información relacionada a dicho código y habilitando los campos de la historia para ser modificados			
Flujos alternos			
○ <Código no digitado> <El sistema informa al actor mediante un mensaje en pantalla que por favor ingrese un código válido para proceder con la consulta (Regresa al paso 5 del flujo normal)>			
...			
Excepciones			
○ <Excepción: Código no existente> <El sistema debe presentar un mensaje informando al actor que no existe un registro con el código digitado previamente, por favor vuelva a intentarlo (Regresa al paso 5 del flujo normal de eventos)>			
...			
Criterios de aceptación	<Se guarda un nuevo registro en el archivo con los cambios ejecutados por el actor>		
Objetivos relacionados	<El objetivo es que el actor pueda modificar la información correspondiente de x historia en específico y guardar dichos cambios para que queden con ese nuevo estado en el sistema>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 57. Especificación para actualizar historia de usuario - módulo historias de usuarios

ID	<CU1004>	Prioridad	<Alta>
Nombre	<Eliminar Historia de Usuario>		
Descripción	<Mediante el mismo método BUSCAR, al consultar la información requerida, se puede eliminar dicho registro haciendo uso de un botón ELIMINAR situado en la vista en la cual estamos presentando la información>		
Autor	<Juan Camilo Campos>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	< Juan Pablo Devos >		
Actores	<Administrador, Usuario>		
Precondiciones	<El actor debe haber iniciado sesión y debe existir al menos un registro creado previamente>		
Poscondiciones	<Se debe eliminar el registro seleccionado del archivo donde se guardó la información>		
Flujo normal de eventos			
<Acciones>			
26. El actor ingresa al módulo de Gestionar HU			
27. El sistema dirige al actor a la vista correspondiente			
28. El actor da clic en Eliminar Historias			
29. El sistema presenta la pantalla correspondiente y solicita el código que desea consultar			
30. El actor digita el código indicado y da clic en CONSULTAR			
31. El sistema valida la información recibida y presenta los datos relacionados a tal código habilitando un botón ELIMINAR por si se desea borrar el registro buscado			
32. El actor da clic en ELIMINAR			
33. El sistema pregunta al actor si está seguro de eliminar el registro			
34. El actor da clic en SI			
35. El sistema procede a borrar los datos y el registro que tenía establecido con dicho código			
36. El sistema informa al actor que el registro ha sido eliminado con éxito			
37. El actor da clic en ACEPTAR			
38. El sistema dirige al actor al módulo de gestión nuevamente			
Flujos alternos			
o <Código no digitado> <El sistema informa al actor mediante un mensaje en pantalla que por favor ingrese un código válido para proceder con la consulta (Regresa al paso 5 del flujo normal)> ...			
Excepciones			
o <Excepción: Código no existente> <El sistema debe presentar un mensaje informando al actor que no existe un registro con el código digitado previamente, por favor vuelva a intentarlo (Regresa al paso 5 del flujo normal de eventos)> ...			
Criterios de aceptación	<Se elimina toda la información correspondiente al movimiento de los archivos del sistema>		
Objetivos relacionados	<El objetivo es que el actor pueda borrar registros que ya no considere necesarios en el sistema>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 58. Especificación para eliminar historia de usuario - módulo historias de usuarios

ID	<CU1005>	Prioridad	<Alta>
Nombre	<Registrar nuevo usuario>		
Descripción	<Se deben poder crear nuevos registros de usuarios con un código y contraseña que le permitan hacer uso del sistema>		
Autor	< Juan Camilo Campos >		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	< Juan Pablo Devos >		
Actores	<Usuario>		
Precondiciones	<El actor debe poder registrarse y crear un nuevo usuario para él poder hacer uso del sistema>		
Poscondiciones	<Se debe generar en un nuevo espacio en memoria con el registro establecido>		
Flujo normal de eventos			
<Acciones>			
39. El actor ingresa a la pantalla inicial del sistema (Login)			
40. El sistema dirige al actor a la vista correspondiente			
41. El actor da clic en Registrarse			
42. El sistema presenta el formato correspondiente a rellenar para proceder con el registro			
43. El actor completa cada uno de los campos presentados en pantalla y da clic en GUARDAR			
44. El sistema valida la información y presenta un mensaje en pantalla diciendo que el registro se ha efectuado con éxito			
45. El sistema carga la información nueva en el sistema			
46. El actor da clic en OK			
47. El sistema dirige al actor al Login nuevamente para que este pueda ingresar con su cuenta ya cread			
Flujos alternos			
o <Campos incompletos>			
<El sistema informa al actor mediante un mensaje en pantalla que por favor complete todos los campos antes de continuar con el registro (Regresa al paso 5 del flujo normal de eventos)>			
...			
Excepciones			
o <Excepción: Código ya existente>			
<El sistema debe presentar un mensaje informando al actor que ya existe un registro con el mismo código digitado (Regresa al paso 5 del flujo normal de eventos)>			
...			
Criterios de aceptación	<Se genera un nuevo (Usuario, Medicamento o Laboratorio) en el archivo correspondiente>		
Objetivos relacionados	<El objetivo es que el actor pueda registrarse para empezar a hacer uso del sistema>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 59. Especificación para registrar nuevo usuario - módulo historias de usuarios

ID	<CU1006>	Prioridad	<Alta>
Nombre	<Crear Proyecto>		
Descripción	<Se debe poder generar un nuevo proyecto, este será un espacio donde los usuarios podrán guardar sus historias de usuario>		
Autor	<Hansen Arenas>		
Fecha creación	<18/05/2022>	Fecha última modificación	<18/05/2022 >
Fuente(s)	<Juan Pablo Devos>		
Actores	<Administrador, Usuario>		
Precondiciones	<El actor debe haber iniciado sesión>		
Poscondiciones	<Se debe generar en un nuevo proyecto donde el usuario puede generar nuevos registros de historia>		
Flujo normal de eventos			
<Acciones>			
48. El actor ingresa al módulo de Gestionar HU			
49. El sistema dirige al actor a la vista correspondiente			
50. El actor da clic en Crear Proyecto			
51. El sistema pregunta al actor si está seguro de crear un nuevo proyecto			
52. El actor da clic en ACEPTAR			
53. El sistema abre el nuevo proyecto y presenta el módulo correspondiente para agregar historias de usuario (el sistema debe solicitar un nombre para el proyecto que será como el usuario podrá visualizarlo)			
Flujos alternos			
o <Se cancela la creación del Proyecto>			
<El usuario responde con NO cuando el sistema ejecuta el paso 4 del flujo normal de eventos.			
Si esto pasa, el sistema debe dirigir automáticamente al usuario al paso 2 del flujo)>			
...			
Excepciones			
o <Excepción: >...			
Criterios de aceptación	<Se genera un proyecto relacionado al usuario correspondiente		
Objetivos relacionados	<El objetivo es que el actor pueda crear nuevos proyectos donde documentará las historias de usuario para dar un orden a sus registros>		
Reglas del negocio	<>		
Requerimientos No Funcionales	<Interfaz con rápido tiempo de respuesta>		
Anotaciones	<>		

Ilustración 60. Especificación para crear proyecto- módulo historias de usuarios

e) Diagrama de Clases

Se ha representado la estructura de este módulo, para establecer sus clases, atributos, relaciones y operaciones, mediante el siguiente diagrama de clases, como se demuestra a través de la Ilustración 61

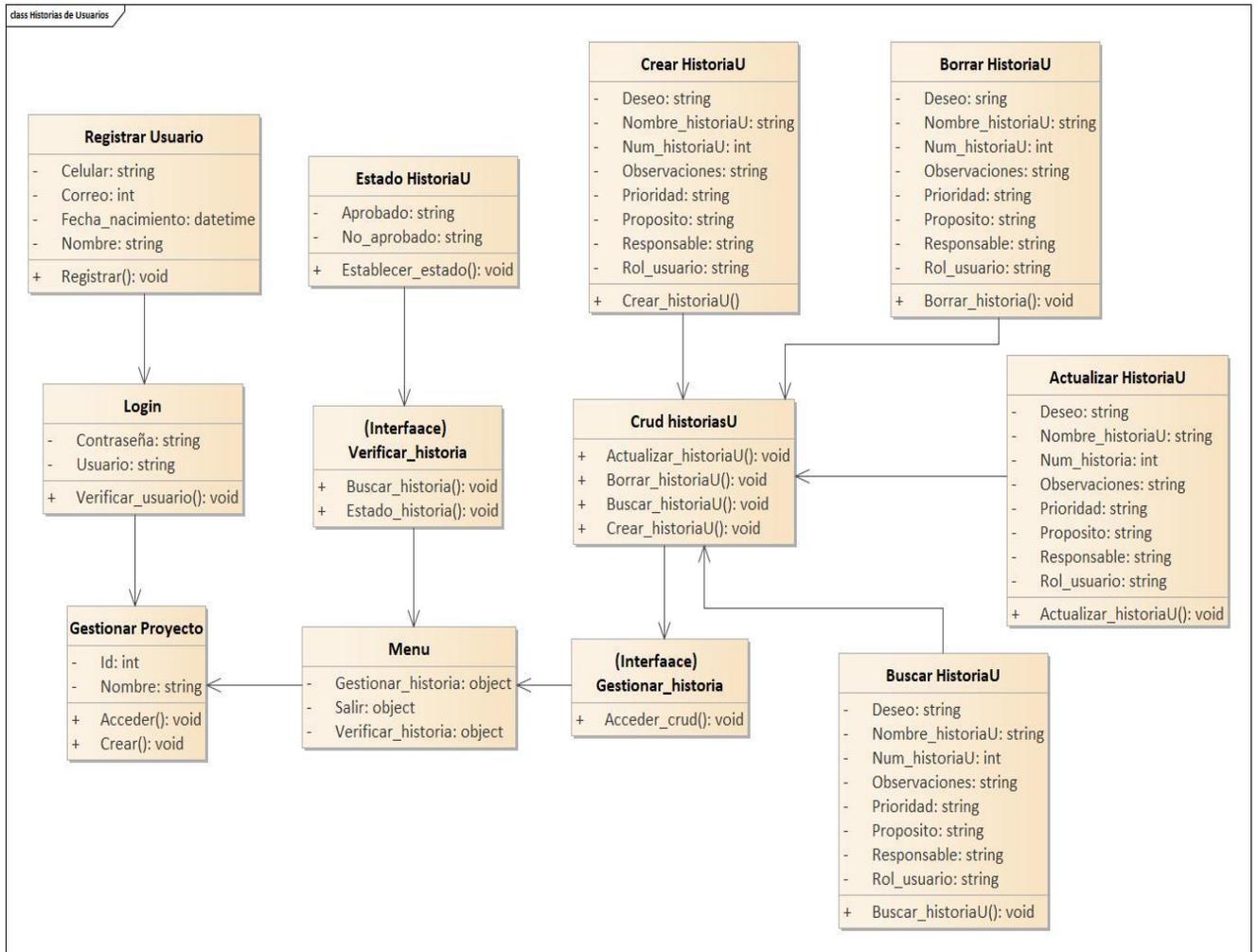


Ilustración 61. Diagrama de clases – módulo historias de usuarios

f) Mockups

Para tener una idea más clara y tener un modelo de cómo será el resultado final o esperado de este módulo, se ha diseñado el siguiente mockup.

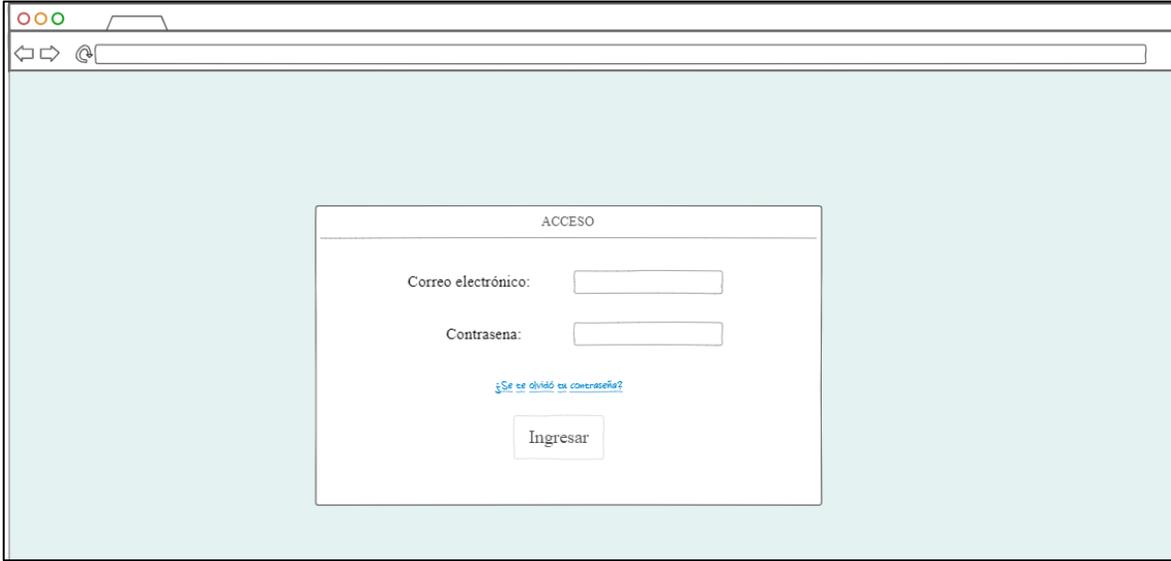


Ilustración 62. Mockup login. - módulo historias de usuarios

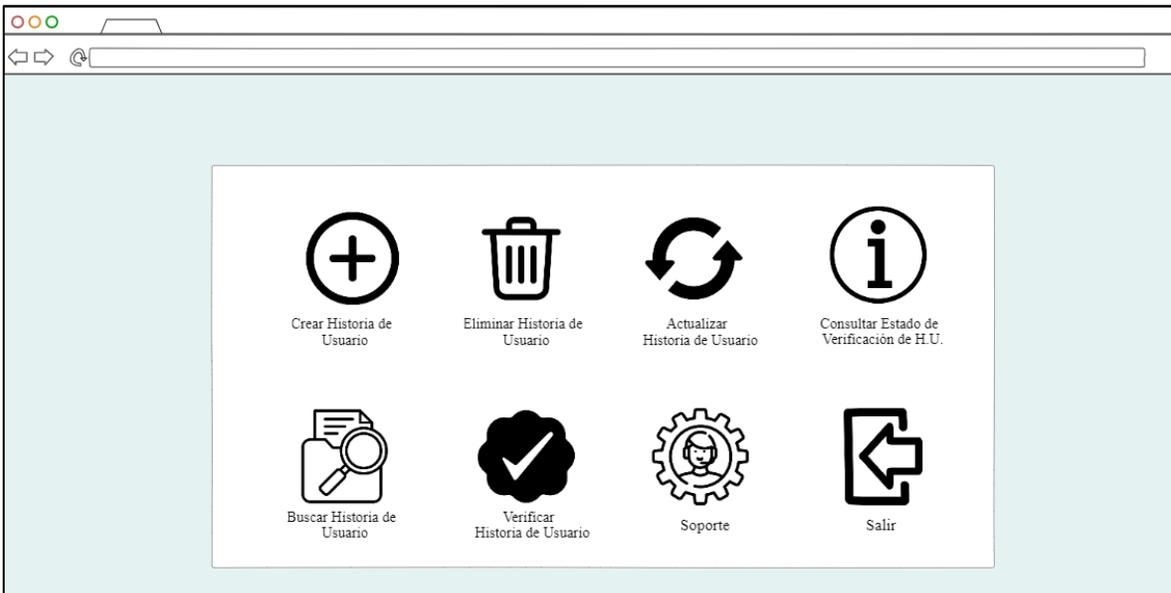


Ilustración 63. Mockup home. - módulo historias de usuarios

Formulario Historia de Usuario

Número H.U.

Nombre de la Historia

Prioridad en el negocio
 Alta Media Baja

Programador Responsable

Rol del Usuario que Ejecuta la Historia

Deseo (Qué)

Propósito (Para)

Observaciones

Ilustración 64. Mockup formulario. - módulo historias de usuarios

Verificación Historia de Usuario

Número H.U.

Número V.H.U.

Estado de la Verificación
 Aprobado NO Aprobado

No.	PREGUNTA	SI	NO	OBSERVACIONES	
1	la historia de usuario es independiente, es decir, que no requiere de otra.	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	Quién Elabora
2	La historia de usuario es negociable, lo que se indica que puede reemplazar por otra de diferente prioridad.	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	
3	La historia de usuario tiene valor, hace referencia a que sea necesaria y de valor para el proyecto.	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	
4	La historia de usuario es estimable, con el objetivo de que el equipo se sienta tranquilo y seguro estimandola.	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	Quién Revisa
5	La historia de usuario es small (pequeña), que no sean grande, que coste de funcionalidades pequeñas.	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	
6	La historia de usuario es testeable (verificable), para que en el futuro se le puedan realizar pruebas	<input type="radio"/>	<input type="radio"/>	<input type="text"/>	

Ilustración 65. Mockup verificación. - módulo historias de usuarios

Actualizar Historia de Usuario

Buscar Número de Historia

Nombre de la Historia

Programador Responsable

Prioridad en el negocio

Alta Media Baja

Rol del Usuario que Ejecuta la Historia

Deseo (Qué)

Propósito (Para)

Observaciones

Actualizar

Ilustración 66. Mockup actualizar. - módulo historias de usuarios

Buscar Historia de Usuario

Buscar Número de Historia

Nombre de la Historia

Programador Responsable

Prioridad en el negocio

Alta Media Baja

Rol del Usuario que Ejecuta la Historia

Deseo (Qué)

Propósito (Para)

Observaciones

Ilustración 67. Mockup buscar. - módulo historias de usuarios

Eliminar Historia de Usuario

Buscar Número de Historia

Nombre de la Historia

Programador Responsable

Prioridad en el negocio

Alta Media Baja

Rol del Usuario que Ejecuta la Historia

Deseo (Qué)

Propósito (Para)

Observaciones

Eliminar

Ilustración 68. Mockup eliminar. - módulo historias de usuarios

9.4 Integración de herramientas UML online al sistema de información CHAMÍ

Con la intención de integrar a futuro herramientas UML que funcionen de manera online al sistema de información CHAMÍ, se investigó que se pueden integrar mediante APIS, las cuales se pueden adquirir individualmente o por paquetes, servicio que ofrecen compañías como Apix-Drive, que de momento se encuentran estructurando su sistema para ofrecer APIS correspondiente a las herramientas UML.[16] [17]

Por otro lado, encontramos que la compañía australiana Atlassian, por medio de su Marketplace, a través del programa que ellos llaman como “Atlassian Confluence”, ofrecen este servicio, ya que el principio fundamental de este, es permitir a los equipos integrar las herramientas que desean utilizar, pero para esto se debe cumplir una serie de requisitos para la integración de estas herramientas, puesto que algunas son de forma gratuita para un máximo de 10 personas por equipo.[18] [19]

[Apoyamos a Ucrania] El 24 de febrero, Rusia lanzó una guerra contra Ucrania. Te instamos a no quedarte callado! ¡Únase a la acción militar de todas las formas posibles!

API Drive

¿Para quién? Integraciones listas para usar Blog Tarifas Española **Entrada** Registro

Integración Draw.io por API previsto

Después de que hagamos la integración, usted podrá conectar Draw.io con uno de **265** sistemas disponibles en 5 minutos, sin programadores.

draw.io +265 sistemas disponibles

Probar gratis

Quiere que hagamos la integración con Draw.io?

Ilustración 69. Apix-Drive Marketplace [16]

ATLASSIAN Marketplace

Buscar aplicaciones

Mostrar aplicaciones para

Todos los productos

Seleccione un tipo d...

Ordenar por

FILTRADO POR

- Aplicaciones fortificadas en la...
- Soportado por socios
- Gratis para todos los equipos
- Versión beta
- Gratis hasta 10 usuarios

CATEGORÍAS

- integraciones
- Utilidades
- Gestión de proyectos
- macros
- Herramientas de administración
- Planos
- Gestión de compilación
- CRM
- Gráficos y diagramas**
- Calidad del código
- Revisión de código
- Integración continua
- Campos Personalizados
- Gadgets del tablero
- Gestión de dependencias

Aplicaciones de gráficos y diagramas

488 resultados filtrados por Gráficos y diagramas x

<p>draw.io Diagramas y pizarras</p> <p>Cree diagramas y pizarras potentes, fáciles de usar y seguros. La solución de creación de diagramas de Confluence mejor valorada... por //SEIBERT/MEDIA - Draw.io</p> <p>★★★★★ (1.1k) 61.2k instalaciones</p> <p>NUBE FORTIFICADA</p>	<p>Diagramas Giffy para Confluence</p> <p>Aplicación Confluence n.º 1 de mayor venta desde 2005; la herramienta favorita de 16 millones de usuarios para dibujar diagramas... por Giffy</p> <p>★★★★★ (763) 20.8k instalaciones</p> <p>NUBE FORTIFICADA</p>	<p>Filtro de tabla y gráficos para confluencia</p> <p>Herramienta versátil y fácil de usar para administrar tablas y datos en Confluence, crear gráficos y hojas de cálculo por Stiltsoft Europa OU</p> <p>★★★★★ (326) 13.6k instalaciones</p> <p>NUBE FORTIFICADA</p>
<p>Informes y gráficos de eazyBI para Jira</p> <p>Analice y visualice Jira y aplicaciones populares: docenas de informes de plantilla, generador de informes personalizados... por eazyBI</p> <p>★★★★★ (149) 8.8k instalaciones</p> <p>NUBE FORTIFICADA</p>	<p>Easy Agile TeamRhythm (anteriormente User Story Maps)</p> <p>Trabaje mejor como equipo para planificar y entregar productos que sus clientes valoran, con nuestra solución intuitiva y ágil en Jira por Fácil Ágil</p> <p>★★★★★ (151) 6.2k instalaciones</p> <p>NUBE FORTIFICADA</p>	<p>Seguimiento de partes de horas para Jira</p> <p>¡Las 3 mejores aplicaciones de seguimiento de tiempo de tendencia! Informes de hojas de tiempo potentes y fáciles de usar basados e... por Aterrizaje</p> <p>★★★★★ (152) 20.7k instalaciones</p> <p>NUBE FORTIFICADA</p>
<p>Markdown Macro, HTML, PlantUML, LaTeX, diagramas, API abierta</p> <p>NUBE FORTIFICADA</p>	<p>Filtros enriquecidos para paneles de Jira</p> <p>Paneles, gadgets e informes en tiempo real</p> <p>NUBE FORTIFICADA</p>	<p>Mejor exportador de PDF para Jira</p> <p>Exportación de PDF personalizable para Jira.</p> <p>NUBE FORTIFICADA</p>

Ilustración 70. Atlassian Marketplace [18]

Además, se encontró la especificación de la Api del sistema Draw.io, la cual está dentro de una biblioteca JavaScript y esta biblioteca de JavaScript se divide en 8 paquetes.

“La clase mxClient de nivel superior incluye (o importa dinámicamente) todo lo demás. La versión actual se almacena en mxClient.VERSION. El paquete del editor proporciona las clases necesarias para implementar un editor de diagramas. La clase principal de este paquete es mxEditor. Los paquetes de vista y modelo implementan el componente gráfico, representado por McGrath. Hace referencia a un mxGraphModel que contiene mxCells y almacena en caché el estado de las celdas en un mxGraphView. Las celdas se pintan con un mxCellRenderer basado en la apariencia definida en mxStylesheet. El historial de deshacer se implementa en mxUndoManager. Para mostrar un icono en el gráfico, se puede utilizar mxCellOverlay. Las reglas de validación se definen con mxMultiplicity.

Los paquetes Handler, layout y shape contienen detectores de eventos, algoritmos de diseño y formas, respectivamente. Los detectores de eventos gráficos incluyen mxRubberband para la selección de bandas elásticas, mxTooltipHandler para información sobre herramientas y mxGraphHandler para modificaciones básicas de celdas. mxCompactTreeLayout implementa un algoritmo de diseño de árbol y el paquete de formas proporciona varias formas, que son subclases de mxShape.

El paquete util proporciona clases de utilidad que incluyen mxClipboard para copiar y pegar, <mxDataTransfer> para arrastrar y soltar, mxConstants para claves y valores de hojas de estilo, mxEvent y mxUtils para manejo de eventos entre navegadores y funciones de propósito general, mxResources para internacionalización y mxLog para salida de consola. El paquete io implementa un mxObjectCodec genérico para convertir objetos de JavaScript en XML. La clase principal es mxCodec. mxCodecRegistry es el registro global para códecs personalizados.”[20]

La cual se puede configurar mediante la siguiente forma para especificar su idioma o su ruta base.

- mxBasePath: especifica la ruta en mxClient.basePath.
- mxImageBasePath: especifica la ruta en mxClient.imageBasePath.
- mxLanguage: especifica el idioma de los recursos en mxClient.language.
- mxDefaultLanguage: especifica el idioma predeterminado en mxClient.defaultLanguage.

- mxLoadResources: especifica si se debe cargar algún recurso. El valor predeterminado es verdadero.
- mxLoadStylesheets: especifica si se debe cargar alguna hoja de estilo. El valor predeterminado es verdadero. [20]

También se encontraron unas librerías basadas en paquetes de JavaScript, las cuales se presentan a continuación:

JointJS: “JointJS es un conjunto de herramientas completo para crear aplicaciones de creación de diagramas y modelos completamente interactivas que se ejecutan en navegadores modernos. La biblioteca principal de JointJS es de código abierto. Está disponible un kit de herramientas comercial, Rappid, que implementa todos los aspectos de una aplicación de diagramación. [21]

Algunas de sus funcionalidades:

- Elementos básicos para diagramas (rectángulo, círculo, elipse)
- Paleta con soporte para notación UML, ERC, PN, predefinida
- Posibilidad de definir tus propias formas con SVG
- Elementos i links interactivos para conectarlos
- Configuración de la apariencia visual de los links
- diagramas jerárquicos
- (de)serialización a JSON
- zoom in/out”

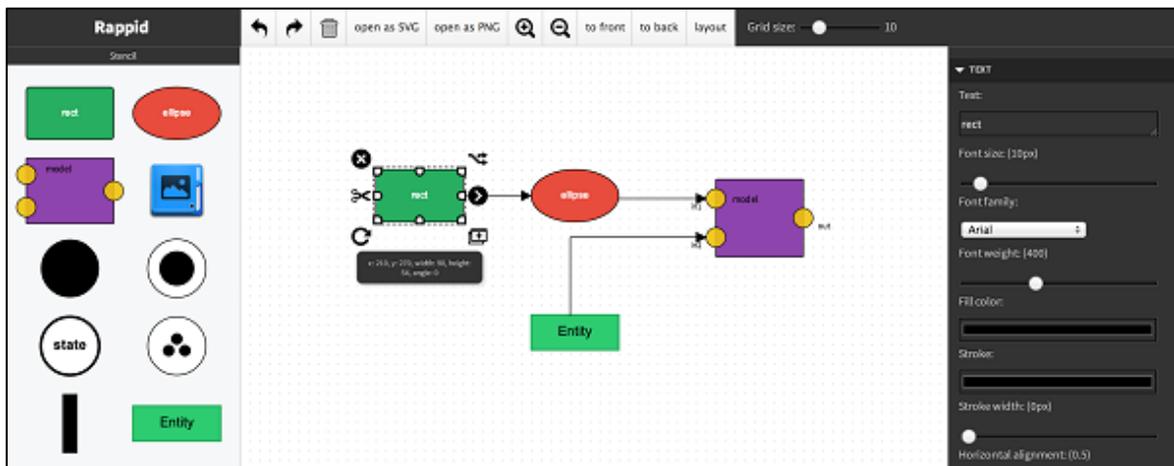


Ilustración 71. Sistema JointJS.

JSUML2: “JsUML2 es una librería HTML5/javascript específica para el dibujo de diagramas UML2. Su principal objetivo es ayudar a visualizar y editar modelos UML2 en todo tipo de aplicaciones web, sin dependencias externas y completamente ejecutable en el cliente vía el navegador.[22]

Da soporte a la modelización de todos los tipos importantes de diagramas UML:

- Casos de uso.
- Diagramas de clases
- Diagramas de componentes
- Diagramas de secuencia
- Diagramas de actividad
- Diagramas de estado”

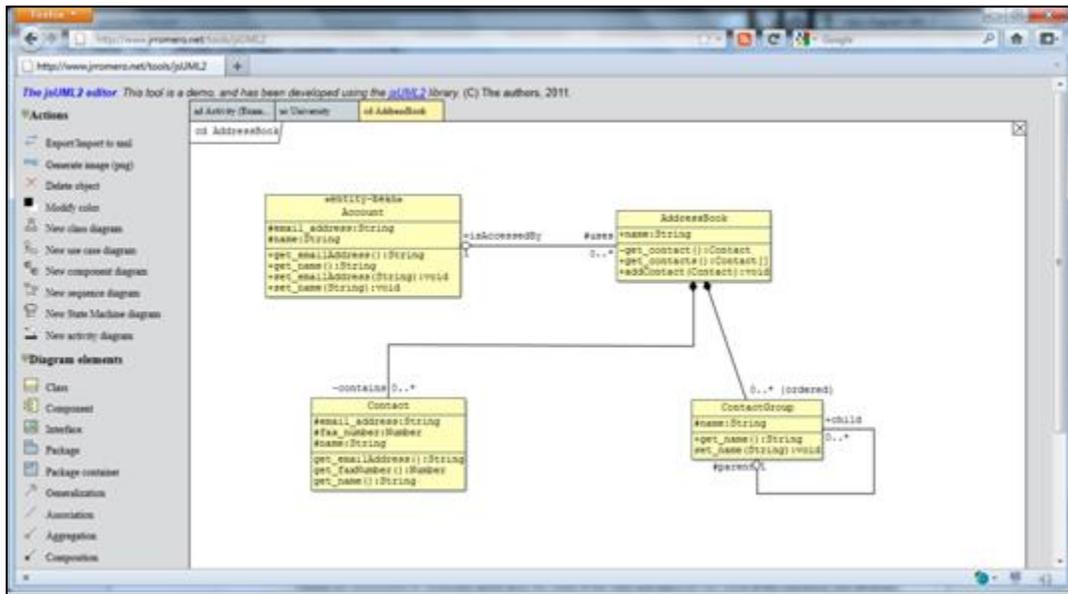


Ilustración 72. Sistema JSUML2.

MXGRAPH: “MxGraph es una librería JavaScript interactiva que hace un muy buen uso de HTML5 y SVG para renderizar los modelos. De hecho, es la librería sobre la que se construye Draw.io. La primera versión es de 2005 y sigue en desarrollo activo a día de hoy gracias a este fork del proyecto original. No utiliza software de terceros lo que facilita su integración en cualquier framework.”[22]

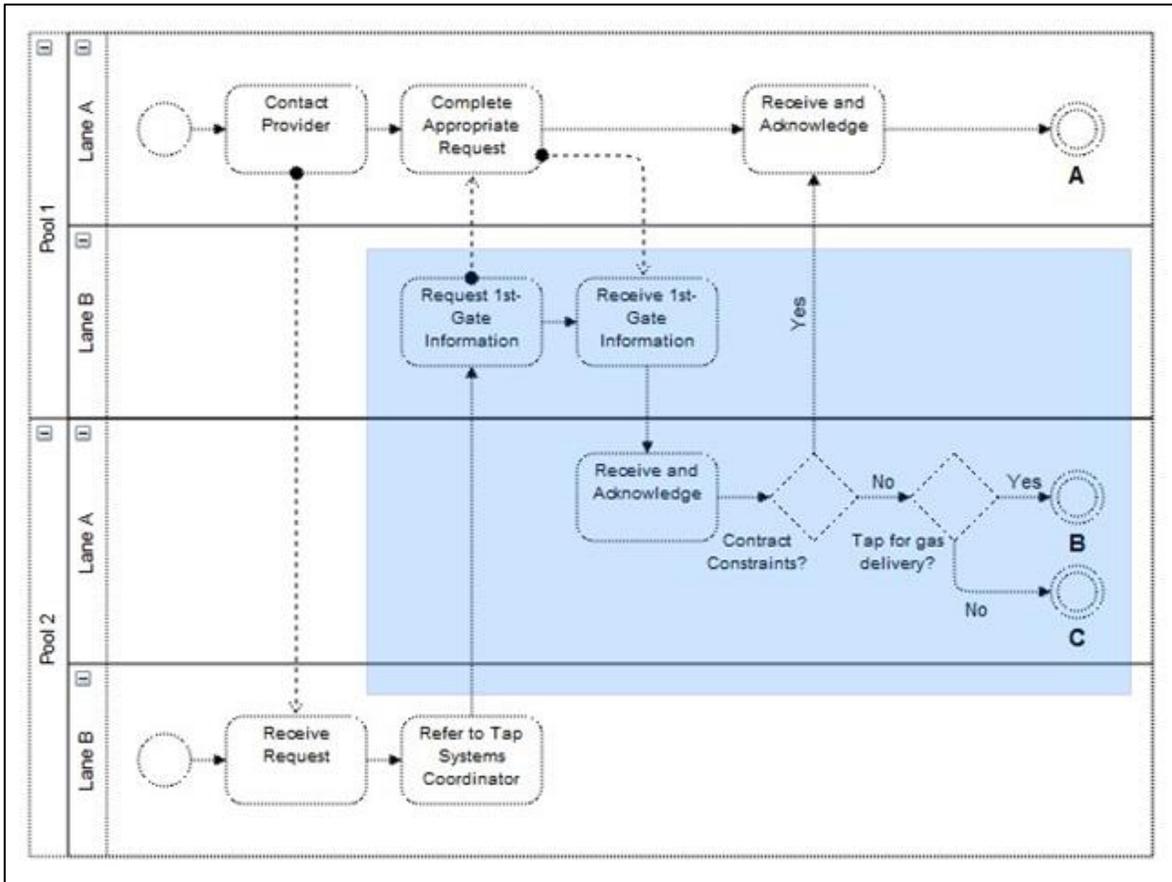


Ilustración 73.Sistema MXGRAPH.

10. CONCLUSIONES

Con base en todos los procesos y necesidades anteriormente presentados en este informe, se llevaron a cabo todas las actividades necesarias para gestionar el diseño y las especificaciones de los requerimientos, para cada uno los módulos correspondientes a las técnicas de procedimiento PEVReS, y de esta manera poder mejorar la funcionalidad del sistema gestor de requisitos CHAMÍ.

Además, se puede inferir que, el sistema CHAMÍ en un futuro será de gran ayuda e importancia para las MiPymes desarrolladoras de software de la región y de la nación, ya que cuyo interés es, facilitar el proceso de desarrollo y gestión de los requerimientos funcionales y no funcionales, usando múltiples actividades sistematizadas de acuerdo con lo que dicen las técnicas PEVReS y todo esto con el fin de permitir y asegurar el desarrollo de software de alta calidad.

Para dar comienzo al proyecto se tuvo que elicitar cada uno de los requerimientos y necesidades que nos manifestaba el cliente y se puede decir que, se cumplió con el objetivo de especificar cada uno de los requerimientos y de esa manera dar continuidad al proyecto.

De igual manera, se puede decir que se logró el objetivo de diseñar el sistema de información en base a cada uno de los módulos del procedimiento PEVReS, con lo cual podemos concluir que el diseño ya se encuentra listo para pasar a la siguiente fase, que en este caso sería el área de codificación para que pueda ser implementado.

Por último, se concluyó con la investigación acerca de, la integración al sistema de información CHAMÍ, el uso de herramientas UML con funcionamiento de manera online, de la cual se pudo determinar que la manera más eficiente es a través de medios de APIS, las cuales de momento no se podrán integrar al sistema, ya que el sistema aún no está en su total funcionamiento y no cumple con los requisitos.

11. RECOMENDACIONES

De acuerdo con el trabajo realizado, se pueden indicar a continuación una serie de recomendaciones a tener en cuenta para el desarrollo (codificación) de cada uno de los módulos especificados en este documento, las cuales se mencionan a continuación:

El sistema CHAMÍ de momento presentan una serie de errores que deben de ser corregidos ya que son fundamentales para su funcionalidad y seguridad, errores tales como; no realiza ciertas validaciones a la hora de realizar un registro de un nuevo usuario, hay campos que no se actualizan... entre otros.

A la hora de especificar los requisitos se debería agregar un filtro donde se pueda escoger primero el proyecto y luego el requerimiento, para que el usuario no tenga que devolverse al apartado anterior para escoger un proyecto y luego retomar la especificación.

Se debería codificar bajo un solo lenguaje de programación, para así evitar problemas ya sea con librerías o claridad del código a la hora de realizar una actualización o modificación y mantener la información dentro de la misma base de datos.

Para la integración de estos módulos se recomienda realizar diferentes tipos de pruebas con la cual puedan evidenciar que el sistema si este funcionando como debe ser y cumpla con cada uno de los requisitos.

Y con respecto a la integración de APIS o plataformas de UML al sistema, se recomendaría usar el servicio que ofrece la compañía Atlassian, ya que llevan mucho más tiempo en el mercado, cuenta con una amplia experiencia y además brindan un amplio apoyo de soporte.

Como trabajo futuro, se plantea la actualización de la base de datos del sistema CHAMÍ, incluido el diagrama Relacional, de tal manera que los nuevos módulos puedan ser implementados.

12. REFERENCIAS

- [1] A. T. Lazo y J. G. G. Botero, «PROCEDIMIENTO PARA ESPECIFICAR Y VALIDAR REQUISITOS DE SOFTWARE EN MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA, BASADO EN ESTUDIOS PREVIOS EN LA REGIÓN», p. 120.
- [2] R. N. Martínez, *El Proceso de Desarrollo de Software*. IT Campus Academy, 2015.
- [3] G. Pantaleo, *Calidad en el desarrollo de software*. Alpha Editorial, 2016.
- [4] I. Sommerville, *Ingeniería del software*. Pearson Educación, 2005.
- [5] M. Kloris, «Guía de ingeniería del software», Accedido: 16 de noviembre de 2022. [En línea]. Disponible en:
https://www.academia.edu/9125978/Guia_de_ingenieria_del_software
- [6] A. Alonso, «Procedimiento PEVReS». <https://pevres.ucp.edu.co/> (accedido 8 de noviembre de 2022).
- [7] M. Fowler y K. Scott, *UML gota a gota: actualizado para cubrir la version 1*. Pearson Educación, 1999.
- [8] «FAEDIS», Accedido: 26 de octubre de 2022. [En línea]. Disponible en:
http://virtual.umng.edu.co/distancia/ecosistema/odin/odin_desktop.php?path=Li4vb3ZhcY9pbmdlbnllcmlhX2luZm9ybWF0aWNhL2luZ2VuaWVyaWFfZGVfcmVxdWVyaW1pZW50b3MvdW5pZGFkXzMv#slide_5.1
- [9] L. E. P. Valencia, M. Á. C. Ávila, I. A. D. González, A. T. Lazo, y J. L. A. Vargas, «El Sistema CHAMÍ para Asistir el Aseguramiento la Calidad de los Requerimientos Funcionales y No Funcionales en la Industria del Software», *Entre Ciencia e Ingeniería*, vol. 15, n.º 30, Art. n.º 30, 2021, doi: 10.31908/19098367.2698.
- [10] E. Gabriel, «Metodologías de desarrollo de software», p. 117.
- [11] «Modelo de prototipos: ¿qué es y cuáles son sus etapas? | Blog | Hosting Plus Colombia», *Hosting Plus*, 6 de julio de 2021.
<https://www.hostingplus.com.co/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/> (accedido 16 de noviembre de 2022).
- [12] «Metodologías de desarrollo. | Ing. Desarrollo de Software UnADM». <https://woodyweb.wordpress.com/2015/08/24/metodologias-de-desarrollo/> (accedido 16 de noviembre de 2022).
- [13] O. C. Medina, M. M. Marciszack, M. A. Groppo, C. S. Castro, y E. H. Moyano, «Software para la gestión de requerimientos del modelo conceptual de un sistema de información», <http://sedici.unlp.edu.ar/handle/10915/54103>, 2016, Accedido: 26 de octubre de 2022. [En línea]. Disponible en:
<http://ria.utn.edu.ar/xmlui/handle/20.500.12272/5975>
- [14] Odin, «FAEDIS, requerimientos», Accedido: 26 de octubre de 2022. [En línea]. Disponible en: <http://virtual.umng.edu.co/distancia/ecosistema/odi>
- [15] «Engineering Requirements Management DOORS Next - Visión general», 21 de septiembre de 2020. <https://www.ibm.com/es-es/products/requirements-management-doors-next> (accedido 16 de noviembre de 2022).

- [16] Apex-Drive, «Sobre nosotros | Historia de la empresa, nuestro equipo», *Apix-Drive*. <https://apix-drive.com/es/about> (accedido 17 de noviembre de 2022).
- [17] Apex-Drive, «Integración con Draw.io | 350 integraciones con otros servicios están listas», *Apix-Drive*. <https://apix-drive.com/es/draw-io> (accedido 17 de noviembre de 2022).
- [18] «Charts & diagramming apps | Atlassian Marketplace». <https://marketplace.atlassian.com/search?category=Charts%20%26%20diagramming> (accedido 17 de noviembre de 2022).
- [19] Atlassian, «Compilar con las API de Atlassian», *Atlassian*. <https://www.atlassian.com/es/devops/build-with-apis> (accedido 17 de noviembre de 2022).
- [20] «API Specification». <http://jgraph.github.io/mxgraph/docs/js-api/files/index-txt.html> (accedido 28 de noviembre de 2022).
- [21] «JointJS: JavaScript diagramming toolkit». <https://modeling-languages.com/jointjs-javascript-diagramming-toolkit/> (accedido 28 de noviembre de 2022).
- [22] «20+ librerías JavaScript para dibujar todo tipo de diagramas y grafos (edición 2022)», 23 de febrero de 2022. <https://ingenieriadesoftware.es/10-librerias-javascript-dibujar-diagramas/> (accedido 28 de noviembre de 2022).

APÉNDICES

Los siguientes apéndices corresponden a los Documentos de especificación de requisitos (SRS) para el cada módulo, generados por la plataforma CHAMÍ.

Apéndice A: SRS módulo para la Verificación y Validación de Prototipos.

Apéndice B: SRS módulo para la Especificación y Verificación de Casos de Usos.

Apéndice C: SRS módulo para Casos de Prueba.

Apéndice D: SRS módulo para Historias de Usuarios.