

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



PROYECTO DE GRADO

**“SISTEMA DE ENCUESTA WEB PARA EL SEGUIMIENTO A
LOS TITULADOS DE LA UNIVERSIDAD MAYOR DE SAN
ANDRÉS”**

**Proyecto de Grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos**

POR: HAROLD EDUARDO RODRIGUEZ POMA

TUTORA METODOLÓGICA: Ph.D. FATIMA CONSUELO

DOLZ SALVADOR

ASESOR: P.Ph.D. YOHONI CUENCA SARZURI

**LA PAZ - BOLIVIA
2021**



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



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

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA

El presente proyecto está dedicado a mi abuela Valentina Ronquillo, quien fue la persona que me motivo a estudiar esta grandiosa carrera.

A mis padres Ladislao Rodriguez y Lily Poma por haber sido mi apoyo a lo largo de mi carrera universitaria y a lo largo de mi vida.

A mi familia y amigos por el gran apoyo que me brindaron en los buenos y malos momentos.

A mis Docentes que gracias a sus consejos y correcciones hoy puedo culminar este trabajo.

GRACIAS

AGRADECIMIENTOS

Este Proyecto de Grado realizado en la Universidad Mayor de San Andrés es un esfuerzo en el cual, directa o indirectamente, participaron distintas personas opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañándome en los momentos de crisis y en los momentos de felicidad. Este trabajo me ha permitido aprovechar la competencia y la experiencia de muchas personas que deseo agradecer en este apartado.

A Dios por la fuerza y la sabiduría que me dio para cumplir con todas mis metas y sueños también por la grandiosa familia que me dio y me apoyo en todo momento.

A mi Asesor y gran Docente de la carrera de Informática P.Ph.D. Yohoni por sus sugerencias, correcciones, paciencia y el tiempo que invirtió en este proyecto. A mi Tutora Metodológica Ph. D. Fátima por su dedicación con este proyecto.

Al jefe de la División de Sistemas de Información y Estadística (D.S.I.E) Lic. Nicanor Delgado por su confianza y apoyo en la elaboración de este proyecto también por lo que me enseñó para que cada día sea una mejor persona, a mis amigos Henry, Marco, Miguel, Rubén por brindarme ayuda y conocimiento para la elaboración de este proyecto y a todos mis compañeros de la división por hacer que todos los días de trabajo sean agradables.

A mis amigos de la universidad que hicieron divertido todo este trayecto, ya que con su amistad y apoyo me alientan a seguir mejorando como amigo, estudiante y persona.

Y para finalizar a todas aquellas personas que me dieron los últimos alientos para acabar con este proyecto.

MUCHAS GRACIAS.

harold.ro.po@gmail.com

RESUMEN

La tecnología se ha vuelto indispensable en el día a día, y cada vez es más común que esté involucrada en todos los aspectos de la vida. Es por esto que avanza pasos agigantados, con el objetivo de hacer la vida más llevadera y sencilla.

Con el avance de la tecnología ya nos es necesario perder horas y horas esperando a que los encuestadores pregunten personas por personas todas y cada una de las preguntas, del mismo modo ya no es necesario esperar a tener todas las respuestas para empezar a recibir los resultados de las mismas. Todo lo mencionado anteriormente se traduce no solo en una reducción de los tiempos necesarios, sino también en un ahorro económico, ya no son necesarios los intermediarios. El envío, la recepción y la manipulación de resultados se hace más sencillo y, por ende, se recurre en menos gastos totales.

El presente proyecto “Sistema de Encuesta Web para el Seguimiento a los Titulados de la Universidad Mayor de San Andrés” con los responsables encargados de la División de Sistemas de Información y Estadística (DSIE), ha sido implementado para poder realizar estas encuestas de forma Online y así poder obtener los resultados de forma rápida y segura. Mejorando así la experiencia de todas las personas tanto para el encuestado como para el encuestador, teniendo así la comodidad para poder responder de forma libre y sin sentirse observado y también aumentando el alcance de las encuestas.

Para el desarrollo de este proyecto se aplicaron las metodologías de desarrollo SCRUM combinándola con la metodología de desarrollo Web UWE para el análisis y el diseño.

Para determinar la calidad del proyecto, se empleó la metodología de evaluación de calidad WebQEM.

Para el desarrollo del Software se utilizó el lenguaje de programación JAVA con la plataforma Eclipse IDE, gestor de base de datos PostgreSQL 9.4 y para la seguridad y administración de roles se usó Spring Security.

ABSTRACT

Technology has become indispensable on a daily basis, and it is becoming more and more common for it to be involved in all aspects of life. That is why it is taking giant steps forward, with the aim of making life easier and more bearable.

With the advancement of technology, it is already necessary for us to waste hours and hours waiting for the interviewers to ask people for each and every question, in the same way it is no longer necessary to wait for all the answers to start receiving the results of the same. All of the aforementioned translates not only into a reduction in the time required, but also into economic savings, as intermediaries are no longer necessary. Sending, receiving and handling results is made easier and, therefore, it is used in less total expenses.

This project "Web Survey System for the Follow-up of Graduates of the Universidad Mayor de San Andrés" with the heads of the Information Systems and Statistics Division (DSIE), has been implemented to be able to carry out these surveys online and thus be able to obtain the results quickly and safely. Thus, improving the experience of all people for both the respondent and the interviewer, thus having the comfort to be able to respond freely and without feeling observed and also increasing the scope of the surveys.

For the development of this project, the SCRUM development methodologies were applied, combining it with the UWE Web development methodology for analysis and design.

To determine the quality of the project, the WebQEM quality evaluation methodology was used.

For the development of the Software, the JAVA programming language was used with the Eclipse IDE platform, PostgreSQL 9.4 database manager and Spring Security was used for security and role administration.

ÍNDICE

CAPÍTULO 1	1
MARCO INTRODUCTORIO	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES	2
1.2.1 ANTECEDENTES INSTITUCIONALES.....	2
1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES	3
1.3 PLANTEAMIENTO DEL PROBLEMA	4
1.3.1 PROBLEMA CENTRAL.....	4
1.3.2 PROBLEMAS SECUNDARIOS	4
1.4 DEFINICIÓN DE OBJETIVOS.....	5
1.4.1 OBJETIVO GENERAL	5
1.4.2 OBJETIVOS ESPECÍFICOS	5
1.5 JUSTIFICACIÓN.....	5
1.5.1 JUSTIFICACIÓN SOCIAL	5
1.5.2 JUSTIFICACIÓN TECNOLÓGICA	6
1.5.3 JUSTIFICACIÓN ECONÓMICA.....	6
1.6 ALCANCES Y LÍMITES	6
1.6.1 ALCANCES	6
1.6.2 LÍMITES	7
1.7 APORTES	8
1.7.1 PRÁCTICO	8
1.7.2 TEÓRICO.....	8
1.8 METODOLOGÍA.....	8
CAPÍTULO 2	10
MARCO TEÓRICO	10
2.1 INGENIERÍA DE SOFTWARE	10
2.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	10
2.2.1 METODOLOGIA TRADICIONAL	10
2.2.2 METODOLOGIAS ÁGILES	11
2.2.3 DIFERENCIAS ENTRE METODOLOGIA TRADICIONAL Y ÁGIL	11
2.3 METODOLOGIAS ÁGIL SCRUM.....	12

2.3.1	ROLES DE SCRUM	13
2.3.2	ELEMENTOS DE SCRUM	16
2.3.3	FASES DEL PROCESO SCRUM	19
2.4	INGENIERÍA WEB	21
2.4.1	METODOLOGÍAS DE LA INGENIERÍA WEB	22
2.4.2	ETAPAS DE LA METODOLOGIA	22
2.4.3	MÉTODOS DE LA INGENIERÍA WEB	23
2.5	PATRON DE DISEÑO MODELO VISTA CONTROLADOR (MVC)	23
2.5.1	MODELOS	24
2.5.2	VISTAS	24
2.5.3	CONTROLADORES	25
2.5.4	ARQUITECTURA DE APLICACIONES MVC	25
2.5.5	LÓGICA DE NEGOCIO	26
2.5.6	LOGICA DE LA APLICACIÓN	26
2.6	METODOLOGÍA UWE	27
2.6.1	FASES DE LA METODOLOGIA UWE	27
2.7	CALIDAD DEL SOFTWARE	32
2.8	SEGURIDAD	32
2.8.1	OWASP	32
2.9	HERRAMIENTAS DE DESARROLLO	34
CAPÍTULO 3		36
MARCO APLICATIVO		36
3.1	INTRODUCCIÓN	36
3.2	PRE – GAME	36
3.2.1	PLANIFICACIÓN	36
3.2.2	CLASIFICACIÓN E IDENTIFICACIÓN DE LOS USUARIOS (ROLES)	38
3.3	GAME	39
3.3.1	PRIMER SPRINT	40
3.3.2	SEGUNDO SPRINT	46
3.3.3	TERCER SPRINT	51
3.3.4	CUARTO SPRINT	57
3.3.5	QUINTO SPRINT	61
3.3.6	DISEÑO FINAL	66

3.4	POST GAME.....	68
3.4.1	DISEÑO DE INTERFACES	69
3.4.2	PRUEBAS UNITARIAS.....	72
3.4.3	PRUEBAS DE ESTRESS	73
CAPÍTULO 4.....		74
CALIDAD Y SEGURIDAD.....		74
4.1	METODOLOGIA WEBQEM.....	74
4.1.1	MODELO DE AGREGACIÓN Y PUNTAJE.....	74
4.1.2	CARACTERISTICAS DE WEBQEM.....	75
4.1.3	CALIDAD TOTAL	81
4.2	SEGURIDAD	81
4.2.1	APLICACIÓN DE OWASP.....	81
CAPÍTULO 5		83
ANÁLISIS COSTO BENEFICIO.....		83
5.1	INTRODUCCIÓN.....	83
5.2	COCOMO II.....	83
5.2.1	COSTO DEL SISTEMA DESARROLLADO.....	83
5.2.2	COSTO ELABORACIÓN DEL SISTEMA.....	87
5.2.3	COSTOS DE IMPLEMENTACIÓN DEL SISTEMA.....	88
5.2.4	COSTO TOTAL DEL SISTEMA	88
5.3	CÁLCULO BENEFICIO VAN Y TIR	88
5.3.1	VALOR ACTUAL NETO VAN.....	89
5.3.2	TASA INTERNA DE RETORNO TIR	90
5.4	COSTO BENEFICIO	91
CAPÍTULO 6.....		93
CONCLUSIONES Y RECOMENDACIONES		93
6.1	CONCLUSIONES.....	93
6.2	RECOMENDACIONES	93
BIBLIOGRAFÍA		95

ÍNDICE DE FIGURAS

<i>Figura 1.1.</i> Organigrama de la División de Sistemas de Información y Estadística.....	3
<i>Figura 2.1</i> Comparación entre metodología ágil y tradicional	12
<i>Figura 2.2</i> Ciclo del SCRUM	13
<i>Figura 2.3</i> Product Backlog	18
<i>Figura 2.4</i> Sprint backlog.....	19
<i>Figura 2.5</i> Fase del Proceso SCRUM	19
<i>Figura 2.6</i> Metodología SCRUM.....	21
<i>Figura 2.7</i> Modelo Vista Controlador.....	25
<i>Figura 2.8</i> Modelo de contenido	28
<i>Figura 2.9</i> Estereotipos Modelo Navegacional.....	29
<i>Figura 2.10</i> Modelo Navegacional.....	29
<i>Figura 2.11</i> Estereotipos Modelo de Presentación.....	30
<i>Figura 2.12</i> Modelo de Presentación	30
<i>Figura 2.13</i> Modelo de Estructura de Proceso	31
<i>Figura 2.14</i> Modelo de Flujos de Procesos.....	31
<i>Figura 3.1</i> Diagrama de caso de uso Gestión de Usuarios.....	41
<i>Figura 3.2</i> Diagrama Navegacional Gestión de Usuarios.....	43
<i>Figura 3.3</i> Diagrama de presentación – página de ingreso al sistema	44
<i>Figura 3.4</i> Diagrama de presentación – página de administrador Roles.....	45
<i>Figura 3.5</i> Diagrama de Procesos del Administrador	45
<i>Figura 3.6</i> Diagrama de Caso de Uso Administración de Encuesta	47
<i>Figura 3.7</i> Diagrama Navegacional Administración de Encuesta	50
<i>Figura 3.8</i> Diagrama de Presentación Administración de Encuesta	50
<i>Figura 3.9</i> Diagrama de Procesos Administración de Encuesta	51
<i>Figura 3.10</i> Diagrama de caso de uso Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.....	53
<i>Figura 3.11</i> Diagrama Navegacional de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.....	55
<i>Figura 3.12</i> Diagrama de presentación de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.....	56

<i>Figura 3.13</i> Diagrama de procesos de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.....	57
<i>Figura 3.14</i> Diagrama de caso de uso Llenado de encuesta.	58
<i>Figura 3.15</i> Diagrama Navegacional Llenado de encuesta.....	60
<i>Figura 3.16</i> Diagrama de presentación del Estudiante Titulado para el llenado de encuesta. .	60
<i>Figura 3.17</i> Diagrama de Proceso de Llenado de Encuesta.....	61
<i>Figura 3.18</i> Diagrama de caso de uso Realización de Reportes.	63
<i>Figura 3.19</i> Diagrama Navegacional Realización de Reportes.	64
<i>Figura 3.20</i> Diagrama de presentación Realización de Reportes.	65
<i>Figura 3.21</i> Diagrama de procesos de Realización de Reportes.....	66
<i>Figura 3.22</i> Diagrama Entidad Relación.....	67
<i>Figura 3.23</i> Diagrama Relacional.	68
<i>Figura 3.24</i> Página de Inicio de Sesión.....	69
<i>Figura 3.25</i> Agregación de usuario y roles.	70
<i>Figura 3.26</i> Agregación de Encuesta.	70
<i>Figura 3.27</i> Monitoreo de información del estudiante.....	71
<i>Figura 3.28</i> Llenado de Encuesta.....	71
<i>Figura 3.29</i> Software Jmeter.....	72
<i>Figura 3.30</i> Grafica de prueba de estress del sistema.	73
<i>Figura 4.1</i> Escala de Preferencia.....	75

ÍNDICE DE TABLAS

Tabla 2.1 Herramientas de Desarrollo.....	34
Tabla 3.1 Requerimientos del Sistema de Encuesta.....	37
Tabla 3.2 Roles de los usuarios del sistemas.....	38
Tabla 3.3 Descripción de los sprints.....	39
Tabla 3.4 Prioridades del Product Backlog	39
Tabla 3.5 Primer Sprint Product Backlog de Gestión de usuarios	40
Tabla 3.6 Descripción de los casos de uso Gestión usuario	42
Tabla 3.7 Segundo Sprint Product Backlog Administración de Encuesta	46
Tabla 3.8 Descripción de los casos de uso Administración de Encuesta	48
Tabla 3.9 Tercer Sprint Product Backlog Muestreo de la Información y estado de llenado ...	52
Tabla 3.10 Descripción de los casos de Muestreo de la Información del estudiante y estado de llenado de la encuesta del estudiante.....	54
Tabla 3.11 <i>Cuarto Sprint Product Backlog de Llenado de la Encuesta.</i>	57
Tabla 3.12 <i>Descripción del Caso de Uso Llenado de encuesta</i>	59
Tabla 3.13 Cuarto Sprint Product Backlog de Realización de Reportes	62
Tabla 3.14 Descripción del Caso de Uso Realización de REportes	63
Tabla 4.1 Cálculo de entradas para el Punto Función	77
Tabla 4.2 Escala de niveles de influencia.....	77
Tabla 4.3 Cálculo de punto función Factores de Complejidad.....	77
Tabla 4.4 Cálculo de Usabilidad del sistema de Encuesta	79
Tabla 4.5 Resultados de las métricas de calidad del sistema de encuestas	81
Tabla 5.1 Factor LCD/PF de lenguajes de programación	83
Tabla 5.2 Modelo básico para tipos de proyectos	85
Tabla 5.3 Tabla de resultados COCOMO II.....	87
Tabla 5.4 Costo de elaboración del sistema	87
Tabla 5.5 Costo total del sistema.....	88
Tabla 5.6 Flujo de Caja proyectado en 5 años.....	89
Tabla 5.7 Proyección del Beneficio Neto.....	92

CAPÍTULO 1 MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

La tecnología se ha vuelto indispensable en el día a día, y cada vez es más común que esté involucrada en todos los aspectos de la vida. Es por esto que avanza pasos agigantados, con el objetivo de hacer la vida más llevadera y sencilla.

Toda organización necesita manejar una gran cantidad de información debido a las actividades que realizan en el día a día. La información es un conjunto de datos que al ser organizados previamente pueden crear un mensaje basado en hechos.

La información en las organizaciones es de vital importancia debido a que les permite a las personas adquirir el conocimiento necesario para la toma de decisiones oportunas.

La **UNIVERSIDAD MAYOR DE SAN ANDRÉS**, como una casa de estudio superior no se queda al margen de usar la tecnología en sus diferentes procesos dentro de sus instalaciones, como el desarrollo y uso de sus sistemas de registro universitario, sistema de seguimiento académico y sistema kardex digitalizado que facilitan al personal administrativo en el manejo de la información académica de la población estudiantil

Los estudiantes que terminan la carrera Universitaria pueden convertirse en grandes promotores o detractores de una institución educativa. Qué mejor que conocer el punto de vista, por ejemplo, del cronograma de estudio, vida estudiantil, de las oportunidades de trabajo.

Por lo anterior mencionado, el presente proyecto pretende el desarrollo e implementación de un sistema de seguimiento a los estudiantes titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS**, en el lenguaje de programación JAVA haciendo un inicio de sesión para dar acceso a los estudiantes titulados a la encuesta, realizada por los encargados de cada carrera, y para el almacenamiento de los datos se hará uso del sistema de gestión de base de datos PostgreSQL para que al final se pueda realizar los cuadros estadísticos.

Concluyendo se va a utilizar las nuevas tecnologías para ayudar a automatizar los distintos procesos que va a necesitar el sistema, para tener la información de los estudiantes

titulados en forma digital en cada momento y que nos permitirá acceder a esta información con la exactitud y disponibilidad necesarios.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

La **U.M.S.A.** es la principal universidad pública del Estado plurinacional de Bolivia. Fue establecida en 1830 en la ciudad de La Paz.

Dentro de la Universidad Mayor de San Andrés, la División de Sistemas de Información y Estadísticas (D.S.I.E.) dependiente del Departamento de Tecnologías e Información (D.T.I.C.), es una unidad que presta servicios para la Universidad Mayor de San Andrés (U.M.S.A.) desde la década de los 80 como referente en la historia Informática. Junto al Centro de Cálculo dependiente de la facultad de ingeniería, desde entonces conocido como Centro de Procesamiento de Datos e Información (C.P.D.I.), fueron las unidades responsables del desarrollo e implementación de sistemas informáticos permitiendo la sistematización, manejo ágil y oportuno de la información académico financiera de la universidad.

Misión

La misión de la División de Sistemas de Información y Estadísticas (D.S.I.E.) es de brindar información de los estudiantes en línea referentes a los datos e identificación personal, rendimiento académico, número de estudiantes, por carreras y sedes dentro de la universidad además de procesar y difundir la información estadística de la Universidad Mayor de San Andrés.

Para una mejor comprensión de las áreas que conforman, el Organigrama de la División de Sistemas de Información y Estadística a ser aprobado tiene la siguiente estructura orgánica como se observa en la Figura 1.1.

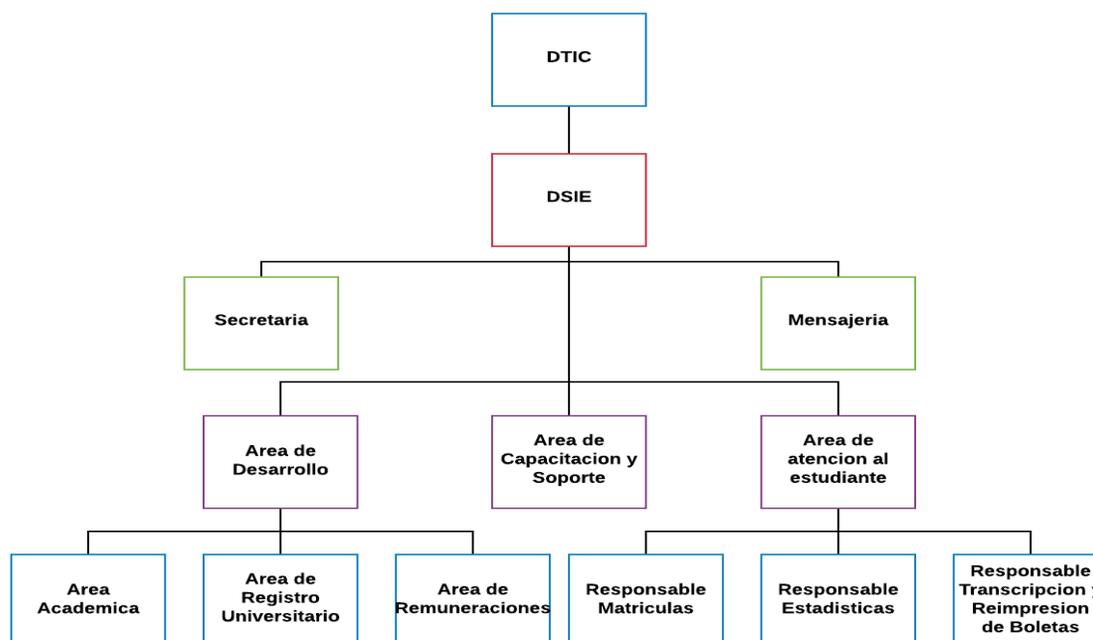


Figura 1.1. Organigrama de la División de Sistemas de Información y Estadística.
Fuente: División de Sistemas de información y Estadística, 2020

1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES

Los trabajos citados a continuación fueron realizados en la carrera de Informática de la Universidad Mayor de San Andrés

- **Tema: Sistema de información y seguimiento académico para el instituto NUEVA ESPERAZA**, elaborado por Freddy Julio Colque Carvajal publicado en el año 2010 en el cual coadyuve en los procesos de inscripción, registro de asignación de materias, listado de alumnos, pago de mensualidades, control de notas y seguimiento académico de forma automatizada de modo que se mejore el manejo de la información.
- **Tema: Sistema de Seguimiento y Gestión Administrativa Caso: Unidad de Postgrado en Informática - UMSA**, elaborado por Nestor Calixto Chura Aguilar publicado en el año 2014 en el cual desarrolla e implementar un sistema de seguimiento académico y gestión administrativa, para la unidad de postgrado en informática para organizar y optimizar los procesos de información académicos, el control de depósitos bancarios y la posibilidad de realizar trámites en línea.

- **Tema: Sistema Web de Administración y Seguimiento Académico para la Unidad de Postgrado de la Carrera de Contaduría Pública**, elaborado por Maria Estela Chambi Quispe publicado en el año 2016 el cual tiene como objetivo desarrollar e implementar un sistema web de administración y seguimiento académico para la Unidad de Postgrado de la Carrera de Contaduría Pública, el cual ayuda en los procesos en el área académica y proporcionar información oportuna y confiable.

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 PROBLEMA CENTRAL

En la **UNIVERSIDAD MAYOR DE SAN ANDRÉS**, no se cuenta con un sistema de encuesta para el seguimiento a los estudiantes titulados de Pre y Post para poder sacar cuadros evaluativos, es por lo cual se propone desarrollar un sistema de encuesta para la universidad utilizando las nuevas tecnologías.

¿Cómo poder obtener la información de los estudiantes titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS**, en el tramo final de la licenciatura y el acceso al mundo laboral, la situación laboral actual, formación de algún curso de postgrado?

1.3.2 PROBLEMAS SECUNDARIOS

- Retardo en las correcciones de las preguntas para el cuestionario por parte los encargados de carrera.
- Demora en la realización de los cuadros evaluativos de las encuestas realizadas por estudiantes titulados de la UMSA.
- Consulta dificultosa de la información de los datos del estudiante titulado, para la información o capacitación sobre la encuesta.
- Dificultad a la hora de revisar las encuestas realizadas por los estudiantes titulados.
- Perdida de información a la hora de la realizar los cuadros evaluativos de la encuesta.
- Poca organización a la hora de realizar la encuesta para los estudiantes titulados.

1.4 DEFINICIÓN DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar un sistema web de encuestas para los titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS**, que permita guardar toda la información de los que realizaron la encuesta y así poder realizar los cuadros evaluativos.

1.4.2 OBJETIVOS ESPECÍFICOS

- Desarrollar una interfaz que permita agregar, borrar, modificar y ver los listados de las preguntas de la encuesta.
- Hacer uso del sistema de encuesta a los titulados para poder realizar reportes de cuadros evaluativos, mostrando el porcentaje y toda la información necesaria de cada pregunta respondida.
- Facilitar la información de los estudiantes titulados mediante el sistema de encuesta.
- Presentar una interfaz que permita ver a tiempo real el porcentaje de las preguntas respondidas a los estudiantes titulados.
- Asegurar toda la información de las encuestas realizadas a los estudiantes titulados en el sistema de encuesta.
- Asignación de roles para una mejor organización en el sistema de encuestas.

1.5 JUSTIFICACIÓN

1.5.1 JUSTIFICACIÓN SOCIAL

El desarrollo del sistema ayudara a las distintas carreras de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS (U.M.S.A.)** a la recolección de datos sobre el tramo final de la licenciatura y el acceso al mundo laboral, la situación laboral actual, formación de algún curso de postgrado para poder obtener cuadros evaluativos de cada gestión de los estudiantes titulados.

Esto implica a su vez el ahorro de tiempo y dinero tanto a la U.M.S.A. como a sus carreras poniendo a su disposición la información recolectada de sus estudiantes titulados.

Tanto el personal de administrativo encargado del sistema de encuesta de la División de Sistemas de Información y estadística como el encargado de la carrera podrán hacer el manejo del sistema en distintas tareas de forma rápida eficaz y eficiente

1.5.2 JUSTIFICACIÓN TECNOLÓGICA

Para desarrollar el sistema se hará uso del lenguaje de programación JAVA. Que en la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de internet como en la informática en general. Una de las principales características es que es multiplataforma, es orientada a objetos, tiene seguridad por que se va actualizando para corregir los problemas que se encuentra y por último cuenta con bastante documentación.

Para almacenar la información se hará uso de PostgreSQL es un sistema gestor de base de datos relacionales, está orientado a objetos, es multiplataforma, es de código abierto y escalable esto quiere decir que puede manejar bases de datos enormes.

Hoy en día un gran porcentaje de los estudiantes titulados cuenta con un celular y acceso a internet esto ayudará al llenado de la encuesta ya que este mismo estará implementada en los servidores de nuestra casa de estudio superior.

1.5.3 JUSTIFICACIÓN ECONÓMICA

Para el desarrollo del sistema de encuesta se hará uso del lenguaje de programación JAVA y del gestor de base de datos PostgreSQL, ambos son de código abierto, lo que nos permite total libertad a la hora del desarrollo y también de licencias.

Las encuestas en papel generan costos por impresión, fotocopias y de envío aparte de que el procesamiento de los datos lleva mucho tiempo y esfuerzo.

Por lo cual se sistematizará y será una encuesta online esto hará del proceso de análisis más eficiente esto quiere decir que no hay costes de papel, impresión ni de envío. Con esto de la encuesta online la distribución será instantánea, la recolección en tiempo real y la tabulación de resultados permite que el análisis de datos sea instantáneo.

1.6 ALCANCES Y LÍMITES

1.6.1 ALCANCES

El sistema web para la encuesta a los estudiantes titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS** se implementará bajo los servidores de este mismo.

Los módulos que alcanzara el siguiente proyecto de grado son:

- **Módulo del Administrador**, el encargado del sistema de encuesta de la División de Sistemas de Información y Estadística (DSIE) podrá dar acceso al módulo de encuesta, al encargado de la carrera que lo solicito.
- **Módulo de Encuesta**, en la cual el encargado de la carrera podrá realizar las preguntas pertinentes para que los estudiantes titulados puedan responderlo.
- **Módulo del Encuestador**, donde el encuestador podrá ver la información limitada, como el nombre(s), apellido(s), numero de celular y correo electrónico del estudiante titulado, aparte podrá observar que estudiante titulado ya lleno la encuesta, quien no lo lleno y quien está en proceso de llenado.
- **Módulo de Inicio de Sesión**, donde el administrador, encuestador, estudiante titulado o el encargado de la encuesta tendrá que iniciar sesión para poder trabajar en la encuesta aparte de que dará seguridad al mismo sistema.
- **Módulo de Muestreo**, en el cual se ira graficando toda la información obtenida de las encuestas realizadas por los estudiantes titulados.
- **Módulo de Carga de Usuarios**, donde el encargado del sistema de encuesta podrá realizar las habilitaciones a los estudiantes titulados de la **UMSA** para que puedan realizar la encuesta.

1.6.2 LÍMITES

El sistema web de encuestas para los titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS** tiene las siguientes limitaciones:

El sistema web de encuestas será de uso exclusivo para los estudiantes titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS (U.M.S.A.)** y no podrá ser utilizada por otros estudiantes de otras universidades.

- El manejo de la información de los estudiantes titulados de la **U.M.S.A.** estará estrictamente a cargo de la División de Sistemas de Información y Estadística.
- EL sistema solo dará la información del llenado de la encuesta de los estudiantes titulados de la **U.M.S.A.** al encargado de la carrera.
- Para el uso del sistema, los encargados de carrera tendrán que enviar la solicitud de uso del sistema a la División de Sistemas de Información y Estadística.

1.7 APORTES

1.7.1 PRÁCTICO

El aporte practico serán los mismos módulos del sistema y el sistema de encuestas a los estudiantes titulados estos modulo serán:

- Módulo del Administrador.
- Módulo de Encuesta
- Módulo del Encuestador
- Módulo de Inicio de Sesión
- Módulo de Muestreo
- Módulo de Carga de Usuarios

1.7.2 TEÓRICO

El aporte teórico que dará el presente proyecto es el uso de distintas metodologías y estándares de seguridad, las cuales serán:

- Metodología ágil SCRUM, para poder lograr una mejor organización.
- Metodología de desarrollo web UWE que es una variación de los diagramas UML, pero de forma más fácil y clara
- Para la calidad de software se hará uso del estándar internacional ISO 9126, para el desarrollo óptimo del sistema

1.8 METODOLOGÍA

La metodología es una estructura sistemática para la recolección, ordenamiento y análisis de la información, que permite la interpretación de los resultados en función del problema que se investiga.

Dado que el objetivo del proyecto es obtener la información de los estudiantes titulados de la **U.M.S.A.** se recurrió a un diseño no experimental, ya que el diseño no experimental observa los fenómenos tal como se dan en su contexto natural, para posteriormente analizarlos.

El presente proyecto será diseñado bajo el planteamiento metodológico del enfoque cuantitativo, puesto que este es el mejor que se adapta a las características y necesidades del proyecto.

Del enfoque cuantitativo se tomará la técnica de encuestas para obtener la información de los estudiantes titulados de la **UNIVERSIDAD MAYOR DE SAN ANDRÉS**.

Al final, la técnica que se utilizará en el procesamiento de los datos será la estadística descriptiva que consiste en un conjunto de procedimientos que tienen por objeto presentar una gran cantidad de datos por medio de tablas, gráficos.

Se aplicará la metodología ágil SCRUM, al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llaman “Sprint”.

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el *feedback* con los clientes.

Aparte se hará uso de la metodología de desarrollo web UWE, el principal objetivo del enfoque es proporcionar un lenguaje de modelado específico del dominio basado en UML.

Y para evaluar la calidad del Software se hará uso del estándar internacional **ISO 9126**, es un estándar que provee un entorno para que las organizaciones definan un modelo de calidad para el producto del software.

En cuanto a herramientas de desarrollo de software se establece:

- Lenguaje de Programación Java versión 8.
- Plataforma de desarrollo Eclipse IDE 2020-06
- Frameworks Spring, JPA.
- Gestor de base de datos PostgreSQL 9.4

CAPÍTULO 2

MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza (Sommerville, 2005).

En la disciplina de la ingeniería, los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones (Sommerville, 2005).

La ingeniería del software no solo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

2.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software son indispensables para crear o actualizar software la calidad que cumpla con los requisitos de los usuarios, son una parte, fundamental de la ingeniería de software la cual denomina metodología a un conjunto de métodos coherentes y relacionados por unos principios comunes (Rivas, Corona, Gutiérrez, & Hernández, 2015).

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado (Maida & Pacienza, 2015).

2.2.1 METODOLOGIA TRADICIONAL

Las metodologías tradicionales son denominadas, a veces, de forma despectiva, como metodologías pesadas.

Centran su atención en llevar a una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto (Maida & Pacienza, 2015).

Estas metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que esta todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar (Maida & Pacienza, 2015).

2.2.2 METODOLOGIAS ÁGILES

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo (Maida & Pacienza, 2015).

Un modelo de desarrollo ágil, generalmente es un proceso, incrementales entregas frecuentes con ciclos rápidos, cooperativo clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante, sencillo el método es fácil de aprender y modificar para el equipo y finalmente adaptativo capaz de permitir cambios de último momento. Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías (Maida & Pacienza, 2015).

2.2.3 DIFERENCIAS ENTRE METODOLOGIA TRADICIONAL Y ÁGIL

En las metodologías tradicionales el principal problema es que nunca se logra planificar bien el esfuerzo requerido para seguir la metodología. Pero entonces, si logramos

definir métricas que apoyen la estimación de las actividades de desarrollo, muchas prácticas de metodologías tradicionales podrían ser apropiadas. El no poder predecir siempre los resultados de cada proceso no significa que estemos frente a una disciplina de azar. Lo que significa es que estamos frente a la necesidad de adaptación de los procesos de desarrollo que son llevados por parte de los equipos que desarrollan software (Maida & Pacienza, 2015).

Es impórtate tener en cuenta que el uso de un método ágil no vale para cualquier proyecto. Sin embargo, una de las principales ventajas de los métodos ágiles es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante agradables (Maida & Pacienza, 2015), en la Figura 2.1 que se muestra a continuación aparece una comparativa entre dos grupos de metodologías.

Metodologías ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
Poca documentación	Documentación exhaustiva
Muchos ciclos de entrega	Pocos ciclos de entrega

Figura 2.1 Comparación entre metodología ágil y tradicional
Fuente: (Maida & Pacienza, 2015)

2.3 METODOLOGIAS ÁGIL SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (Albaladejo, 2019).

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas pro el beneficio que aportan el receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Albaladejo, 2019).

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas), límite máximo de feedback de producto real y reflexión. Cada iteración que proporciona un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite (Albaladejo, 2019), como se muestra en la Figura 2.2.



Figura 2.2 Ciclo del SCRUM

Fuente: (Albaladejo, 2019)

2.3.1 ROLES DE SCRUM

Un equipo Scrum se compone de tres roles: Product Owner, Equipo de Desarrollo y Scrum Master.

2.3.1.1 PRODUCT OWNER

El Product Owner es la persona responsable de maximizar el valor, tanto del producto como del trabajo realizado por el equipo de Desarrollo (Alaimo, 2015).

Su trabajo se refleja, principalmente, en la gestión del Product Backlog. Esto incluye:

- Asegurar la claridad de los ítems del Product Backlog
- Asegurar la visibilidad, transparencia y comprensión del mismo por parte de todos los involucrados.
- Ordenar los ítems del Product Backlog para lograr los objetivos de la mejor manera posible.
- Asegura que los miembros del equipo de Desarrollo comprenden los ítems del Product Backlog al nivel que sea requerido.

El Product Owner puede delegar este trabajo en el equipo de Desarrollo, sin embargo, el Product Owner sigue siendo el responsable (Alaimo, 2015).

Las actividades más habituales de un Product Owner son:

- Facilitar la co-creación de la visión del producto
- Gestionar las expectativas de los stakeholders
- Determinar y conocer en detalle las características funcionales de alto y de bajo nivel
- Generar y mantener el plan de entregas (reléase plan): fechas de entrega y contenidos de cada una
- Maximizar la rentabilidad del producto.
- Determinar las prioridades de cada una de las características
- Cambiar las prioridades de las características según avanza el proyecto, acompañado así los cambios en el negocio y la tecnología.

El Product Owner se focaliza en maximizar la rentabilidad del producto. La principal herramienta con la cuenta para poder realizar esta tarea es la priorización. De esta manera puede reordenar la col de trabajo del equipo de desarrollo para que éste construya con mayor anticipación las características o funcionalidades más requeridas por el mercado o la competitividad comercial (Alaimo, 2015).

2.3.1.2 EQUIPO DE DESARROLLO

El Equipo de Desarrollo está formado por todos los individuos necesarios para la construcción de un incremento “terminado” de producto al final de cada Sprint. Es el único responsable por la construcción del producto (Alaimo, 2015).

El Equipo de Desarrollo es auto-organizado. Esto significa que no existe un líder externo que asigne las tareas ni que determine la forma en la que serán resueltos los problemas. Nadie, ni siquiera el Scrum Master, tiene autoridad para decirle al Equipo de Desarrollo la forma en la que se debe hacer su trabajo (Alaimo, 2015).

El Equipo de Desarrollo es multifuncional. El equipo, representado por conjunto de sus miembros, posee todas las habilidades necesarias para construir un incremento de producto “terminado” al finalizar cada Sprint. Dentro del equipo de desarrollo no existen especialistas exclusivos, sino más bien individuos generalistas con capacidades especiales. Lo que se espera de un miembro del Equipo de Desarrollo es que no solo realice las tareas en las cuales se especializa sino también todo lo que esté a su alcance para colaborar con el éxito del equipo (Alaimo, 2015).

El equipo de desarrollo de no menos de tres personas y no más de nueve personas, tiene tres responsabilidades tan fundamentales como indelegables. La primera es proveer las estimaciones de cuánto esfuerzo será requerido para cada una de las características del producto. La segunda responsabilidad es comprometerse al comienzo de cada Sprint a construir un determinado incremento de producto para cumplir con el objetivo buscado. Y finalmente, también es responsable por la entrega del producto terminado al finalizar cada sprint.

2.3.1.3 SCRUM MASTER

La principal función del Scrum Master es asegurar el entendimiento y seguimiento de Scrum tanto por el Equipo Scrum como por los profesionales con los que este equipo se relaciona (Alaimo, 2015).

El Scrum Master es el líder del equipo y es quien ayuda a alcanzar su máximo nivel de productividad posible. Tomando algunas referencias de Leonardo Wolk podemos decir que el Scrum Master, es:

- Líder por ser un ejemplo a seguir.
- Facilitador por fomentar contextos de apertura y discusión donde todos pueden expresar sus opiniones y lograr consensos comunes.

- Provocador por desafiar las estructuras rígidas y las antiguas concepciones sobre cómo saben hacerse las cosas.
- Detective por involucrarse activamente en la búsqueda e identificación de indicios y pistas en la narrativa de equipos y los individuos.
- Soplador de brasas un socio facilitador del aprendizaje, que acompaña al otro en una búsqueda de su capacidad de aprender para generar nuevas respuestas.

El rol del Scrum Master también incluye asegurar que el desarrollo del producto tenga la mayor probabilidad de ser completado de forma exitosa. Para lograr ese cometido. Trabaja de cerca con el Product Owner asegurando una correcta priorización de los requerimientos, por un lado, y con el equipo de desarrollo para convertir los requerimientos en un producto funcionando (Alaimo, 2015)

2.3.2 ELEMENTOS DE SCRUM

2.3.2.1 SPRINT

Las iteraciones en Scrum se conocen como Sprints. Scrum, como todos los enfoques ágiles, es un proceso de construcción incremental e iterativo. Esto significa que el producto se construye en incrementos funcionales entregados en periodos cortos para obtener feedback frecuente.

En general, Scrum recomienda una duración de Sprint de no más de un mes, siendo 2 o 3 semanas lo más habitual que encontraremos en la industria. Una de las decisiones que debemos tomar al comenzar un proyecto o al adoptar Scrum es justamente la duración de los Sprints. Luego, el objetivo será mantener esta duración constante a lo largo del desarrollo del producto (Alaimo, 2015).

Durante el Sprint, no se deben hacer cambios que pongan en peligro el objetivo de Sprint, la expectativa de calidad no debe reducirse y el alcance debe ser clarificado y negociado entre el Product Owner y el Equipo de Desarrollo a medida que se va dando el aprendizaje (Alaimo, 2015).

- Retrasos y adelantos en un Sprint, muchas veces podremos encontrar situaciones en donde el equipo de desarrollo se atrase o se adelante. En este caso, la regla del timeboxing no

nos permitirá modificar (adelantar o postergar) la fecha de entrega o finalización del Sprint. La variable de ajuste en estos casos será el alcance del Sprint, esto es, en el caso de adelantarnos deberemos incrementar el alcance del Sprint agregando nuevos PBIs y reducirlos en el caso de retrasos.

- Cancelación de un Sprint, un Sprint puede ser cancelado antes del cumplimiento del time-box. Solo el Product Owner tiene la autorización para cancelar un Sprint. Tanto el Equipo de Desarrollo, como el Scrum Master o los Stakeholders pueden recomendar la cancelación.

2.3.2.2 OBEJTIVO DEL SPRINT

El Objetivo del Sprint es una meta que será alcanzada durante el Sprint por medio de la implementación del Product Backlog, y provee una guía para que el Equipo de Desarrollo comprenda la razón por la cual está construyendo este Incremento de Producto (Alaimo, 2015).

2.3.2.3 SCRUM DIARIO

Uno de los beneficios de Scrum está dado por el incremento de la comunicación dentro del equipo de proyecto. Esto facilita la coordinación de acciones entre los miembros del equipo de desarrollo y el conocimiento “en vivo” de las dependencias de las actividades que realizan. Por otro lado, se requiere además aumentar y explicitar los compromisos asumidos entre los miembros del equipo de desarrollo y dar visibilidad a los impedimentos que surjan del trabajo que está siendo realizando y que muchas veces nos impiden lograr los objetivos (Alaimo, 2015).

Todos y cada uno de los miembros toman turnos para responder las siguientes tres, y de esa manera comunicarse entre ellos:

- ¿Qué hice desde la última reunión diaria hasta ahora que nos ayuda a alcanzar el objetivo del Sprint?
- ¿En qué voy a estar trabajando desde ahora hasta la próxima reunión diaria para ayudarnos a alcanzar el objetivo del Sprint?
- ¿Qué problemas o impedimentos me impiden, o nos impiden, alcanzar el objetivo del Sprint?

Es importante destacar que en ningún momento se trata de una reunión de reporte de avance o status al Scrum Master ni a otras personas. Por el contrario, es un espacio de estricta comunicación entre los miembros del equipo de desarrollo (Alaimo, 2015).

2.3.2.4 PRODUCT BACKLOG

El Product Backlog es básicamente un listado ordenado de todo aquello que es necesario que forme parte del producto y es la única fuente de requerimientos o cambios a realizar sobre el producto. El Product Backlog se forma de ítems (Product Backlog Ítems, PBIs) o características del producto a construir, mantenido y ordenado por el Product Owner. Es importante que exista una clara priorización, ya que es esta priorización la que determinará el orden en el que el equipo de desarrollo transformará las características (ítems) en un producto funcional acabado (Alaimo, 2015).

Cada Backlog de productos Scrum tiene ciertas propiedades que lo diferencian de una simple lista de tareas.

- Una entrada en el Backlog del producto Scrum siempre agrega valor para el cliente.
- Las entradas en el Backlog del producto tienen prioridad y se ordenan en consecuencia.
- El nivel de detalle depende de la posición de la entrada dentro del Backlog del producto Scrum.
- Todas las entradas son estimadas.
- La cartera de productos de Scrum es un documento vivo.
- No hay elementos de acción o tareas de bajo nivel en el Backlog de producto Scrum.

Como se observa en la siguiente Figura 2.3.

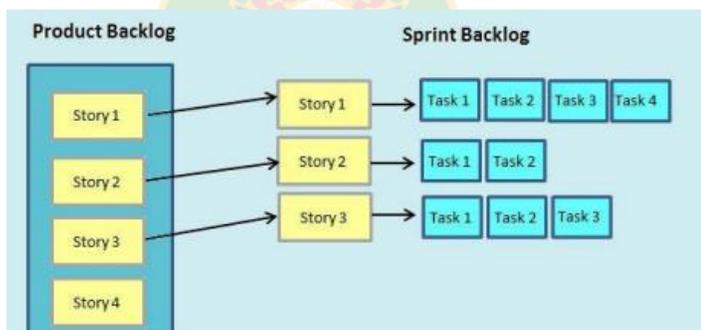


Figura 2.3 Product Backlog

Fuente: Alaimo, 2015

2.3.2.5 SPRINT BACKLOG

El Sprint Backlog es el conjunto de Ítems que fueron seleccionados para trabajar durante un determinado Sprint, conjuntamente con el plan para lograr la entrega del incremento de Producto al finalizar el Sprint y alcanzar el objetivo del Sprint (Alaimo, 2015).

Este plan, y las tareas que su ejecución requiera, es elaborado por el Equipo de Desarrollo. El Sprint Backlog hace visible y transparente todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el objetivo del Sprint, como se muestra en la siguiente Figura 2.4.



Figura 2.4 Sprint backlog

Fuente: Alaimo, 2015

2.3.3 FASES DEL PROCESO SCRUM

Los procesos en Scrum están enmarcados en cajas de tiempo que son unos de los principios del marco de trabajo y es lo que nos permite manejar eficazmente la planeación y ejecución del proyecto como se observa en la Figura 2.5.

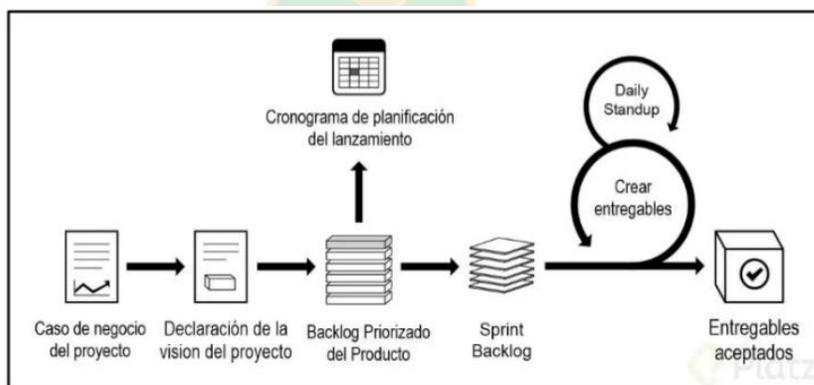


Figura 2.5 Fase del Proceso SCRUM

Fuente: Peralta, 2003

2.3.3.1 PRE – GAME

La fase PRE – GAME incluye dos subfases: planificación y arquitectura

A. PLANIFICACIÓN

Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La lista de Product Backlog es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems (Peralta, 2003).

B. ARQUITECTURA

El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la lista de Product Backlog. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifica los cambios necesarios para implementar los elementos que aparecen en la lista de Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una reunión de revisión de diseño para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada lanzamiento (Peralta, 2003).

2.3.3.2 GAME

En esta fase se espera que ocurran cosas impredecibles. Para evitar el caos scrum define prácticas para observar y controlar las variables técnicas y de entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías, y herramientas de implementación. En lugar de tenerlas en consideración al comienzo del desarrollo, Scrum propone controlarlas constantemente para poder adaptarse a los cambios en forma flexible (Peralta, 2003).

2.3.3.3 POST – GAME

Contiene el cierre del lanzamiento. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados. El sistema está listo para ser liberado y es en esta etapa en la que realiza integración, pruebas del sistema y documentación (Peralta, 2003), como se muestra en la Figura 2.6.

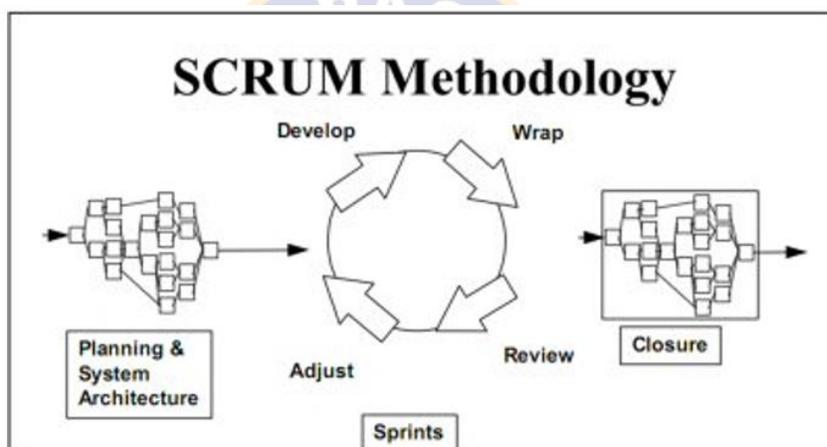


Figura 2.6 Metodología SCRUM

Fuente: Peralta, 2003

2.4 INGENIERÍA WEB

La Ingeniería Web (IWeb) aplica sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinarios y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad. En muchas ocasiones, en el desarrollo de WebApps, se aplicaron (o se aplican), herramientas de ingeniería de Software, sin tener en cuenta que contienen características especiales como: usabilidad, navegabilidad, seguridad, mantenimiento, escalabilidad, entre otras (Valle Rodríguez, 2009).

Roger Pressmanix indica que el uso de soluciones de Ingeniería de Software, conduciría a una “web enmarañada” que entraña un cúmulo de aplicaciones basadas en web más desarrolladas y que tienen muy altas probabilidades de fracaso. Por lo que sugiere que para lograr mayor éxito en el desarrollo y la aplicación de sistemas basados en Web complejos y a gran escala, existe una apremiante necesidad de enfoques disciplinarios y nuevos métodos y herramientas con que desarrollar, desplegar y evaluar los sistemas y aplicaciones basados en Web (Valle Rodríguez, 2009).

2.4.1 METODOLOGÍAS DE LA INGENIERÍA WEB

La ingeniería de software proporciona herramientas de análisis y diseño necesarias para que los desarrolladores puedan representar de forma abstracta un problema y a partir de ésta generar una solución informática, pero los métodos de análisis y diseño de WebApps contienen esos objetivos, más los añadidos por la dimensión de hipertexto como la navegación a través de la información y su presentación (Valle Rodríguez, 2009).

2.4.2 ETAPAS DE LA METODOLOGIA

La inmediatez, evolución y crecimiento continuos, son características de las aplicaciones Web, esto nos lleva a un proceso incremental y evolutivo, que permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan a sus requerimientos. Pressman enumera siete actividades que forman parte del proceso de la IWeb y que son aplicables a cualquier WebApp independientemente de su tamaño y complejidad (Valle Rodríguez, 2009). Las mismas son:

- La Formulación identifica objetivos y establece el alcance de la primera entrega.
- La Planificación genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.
- El Análisis especifica los requerimientos e identifica el contenido.
- La Modelización se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.
- En la Generación de Páginas se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación: las páginas.
- El Test busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que interoperen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.
- Finalmente, el resultado es sometido a la Evaluación del Cliente.

2.4.3 MÉTODOS DE LA INGENIERÍA WEB

Los métodos de la Ingeniería Web definen las etapas y actividades necesarias para efectuar la construcción completa de una aplicación Web (Valle Rodríguez, 2009).

En menor o mayor medida y a veces con diferentes nombres o sub-fases, la mayoría de los métodos coinciden en las siguientes etapas:

- **Diseño Conceptual:** Trata de la especificación del dominio del problema, a través de la definición de datos y sus relaciones.
- **Diseño Navegacional:** Establece los caminos de acceso a la información y sus permisos de visibilidad.
- **Diseño de la presentación o diseño de Interfaz:** Define cómo se muestra la información en la interfaz de usuario.
- **Implementación:** Es la construcción del software a partir de los artefactos generados en las etapas previas.

2.5 PATRON DE DISEÑO MODELO VISTA CONTROLADOR (MVC)

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas (Fernández Romero & Díaz Gonzáles, 2012).

Este modelo de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.

- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

2.5.1 MODELOS

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo (Fernández Romero & Díaz Gonzáles, 2012).

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor". Es opcional, pues las reglas de negocio, pueden estar también en los controladores, directamente en las acciones.
- Notificará a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc.).

2.5.2 VISTAS

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo (Fernández Romero & Díaz Gonzáles, 2012).

Las vistas son responsables de:

- Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado.

- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes.

2.5.3 CONTROLADORES

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo (Fernández Romero & Díaz Gonzáles, 2012).

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta)".

2.5.4 ARQUITECTURA DE APLICACIONES MVC

A continuación, en la Figura 2.7, se mostrará un diagrama que servirá para entender un poco mejor como colaboran las distintas capas que componen la arquitectura de desarrollo de software en el patrón MVC (Alvarez, 2020).

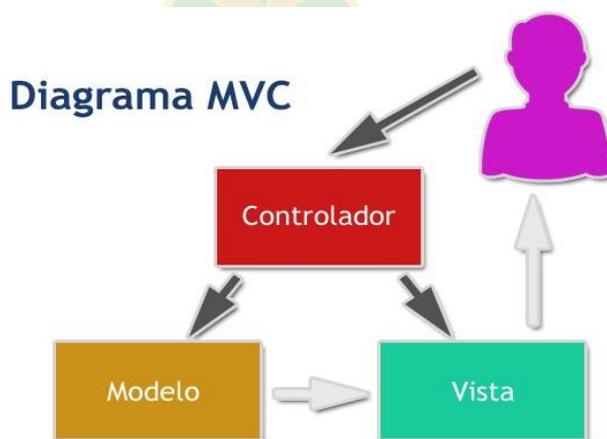


Figura 2.7 Modelo Vista Controlador

Fuente: Alvarez, 2020

El flujo de trabajo característico en un esquema MVC se denota de la siguiente manera:

- El usuario realiza una solicitud a nuestro sitio web. Generalmente estará desencadenado por acceder a una página de nuestro sitio. Esa solicitud le llega al controlador.
- El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda a realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.
- Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación.
- Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente.

2.5.5 LÓGICA DE NEGOCIO

Es un conjunto de reglas que se siguen en el software para reaccionar ante distintas situaciones. En una aplicación el usuario se comunica con el sistema por medio de una interfaz, pero cuando acciona esa interfaz para realizar acciones con el programa, se ejecutan una serie de procesos que se conocen como la lógica del negocio. Este es un concepto de desarrollo de software en general (Alvarez, 2020).

La lógica del negocio, aparte de marcar un comportamiento cuando ocurren cosas dentro de un software, también tiene normas sobre lo que se puede hacer y lo que no se puede hacer. Eso también se conoce como reglas del negocio. Bien, pues en el MVC la lógica del negocio queda del lado de los modelos. Ellos son los que deben saber cómo operar en diversas situaciones y las cosas que pueden permitir que ocurran en el proceso de ejecución de una aplicación (Alvarez, 2020).

2.5.6 LOGICA DE LA APLICACIÓN

Este es otro concepto que se usa en la terminología del MVC que es la “lógica de la aplicación”, que es al que pertenece a los controladores. Por ejemplo, cuando me piden ver el resumen de datos de un usuario. Esa acción le llega al controlador, que tendrá que acceder al

modelo del usuario para pedir sus datos. Luego llamará a la vista apropiada para poder mostrar esos datos del usuario. Si en el resumen del usuario queremos mostrar los artículos que ha publicado dentro de la aplicación, quizás en controlador tendrá que llamar al modelo de artículos, pedirle todos los publicados por ese usuario y con ese listado de artículos invocar a la vista correspondiente para mostrarlos. Todo ese conjunto de acciones que se realiza invocados métodos de los modelos y mandando datos a las vistas forman parte de la lógica de la aplicación (Alvarez, 2020).

2.6 METODOLOGÍA UWE

UWE (UML-Based web Engineering) es una propuesta basada en UML y en el proceso unificado para modelar aplicaciones web. Esta propuesta está formada por una notación para especificar el dominio (basado en UML) y un modelo para llevar a cabo el desarrollo del proceso de modelado. Los sistemas adaptativos y la sistematización son dos aspectos sobre los que se enfoca UWE (Blanco, 2020).

UWE define una extensión del Lenguaje Unificado de Modelado (UML). Ésta, es considerada como una extensión ligera de peso e incluye en su definición tipos, etiquetas de valores y restricciones para las características específicas del diseño Web, las cuales, unidas a las definiciones de UML forman el conjunto de objetos de modelado que se usarán para el desarrollo del modelado utilizado en UWE (Blanco, 2020).

2.6.1 FASES DE LA METODOLOGIA UWE

2.6.1.1 MODELO DE REQUERIMIENTOS

EL primer paso para el desarrollo de un sistema web que se especificará con UWE, es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UM. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados (Blanco, 2020).

2.6.1.2 MODELO DE CONTENIDO

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web (Aurazo, 2017).

A continuación, se en la Figura 2.8 se muestra en modelo de contenido

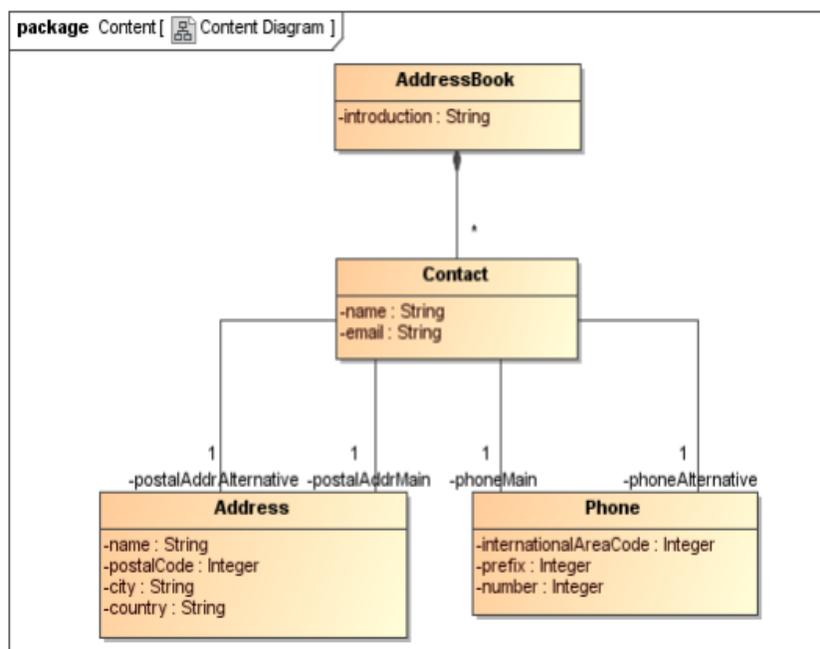


Figura 2.8 Modelo de contenido
Fuente: Maximilians, 2016

En este modelo se puede ver representado que el contenido web está formado por una agenda básica de contactos, esta agenda representada por la clase AddressBook contiene un conjunto de uno o más contactos (clase Contact), cada uno de ellos tiene un nombre, un email, una dirección y un teléfono. De los cuales los dos primeros son de tipo String y los dos últimos son estructuras de otros atributos, representadas por las clases Address y Phone, cada contacto puede tener una dirección y un teléfono principal y otros secundarios (Aurazo, 2017).

2.6.1.3 MODELO DE NAVEGACIÓN

Este modelo indica cómo el sistema de páginas web del sitio está relacionado internamente. Es decir, cómo se enlazan los elementos de navegación (Aurazo, 2017),

Para ello se utilizan unidades de navegación llamadas “nodos” conectadas por enlaces de navegación. Estos nodos pueden ser mostrados en la misma página web, no tiene por qué estar en páginas diferentes (Aurazo, 2017).

Podemos ir viendo los distintos elementos que introduce la metodología UWE, en la Figura 2.9 se muestran los elementos introducidos.

stereotype-names and their icons

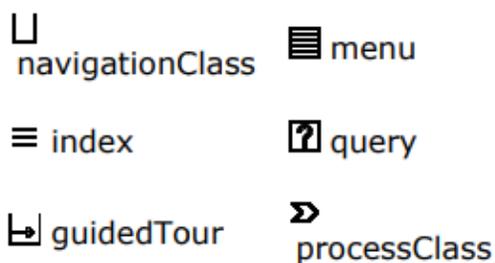


Figura 2.9 Estereotipos Modelo Navegacional
Fuente: Maximilians, 2016

En la siguiente Figura 2.10 se muestra un ejemplo de navegación del sitio web.

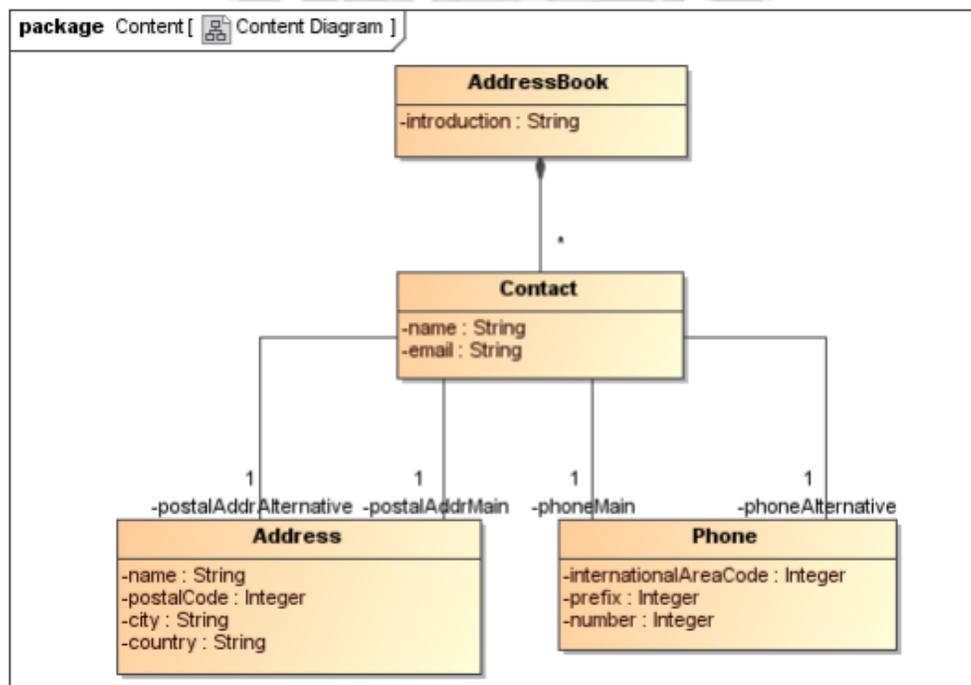


Figura 2.10 Modelo Navegacional
Fuente: Maximilians, 2016

2.6.1.4 MODELO DE PRESENTACIÓN

En la Figura 2.11 se representan las clases de navegación y de procesos que pertenecen a cada página web. Estos son los elementos que introduce la metodología UWE en este modelo:

stereotype-names and their icons

 presentationClass	 presentationPage
 text	 textInput
 anchor	 anchoredCollection
 button	 image
 form	 presentationGroup

Figura 2.11 Estereotipos Modelo de Presentación
Fuente: Maximilians, 2016

A continuación, se muestra el Modelo de presentación Figura 2.12:

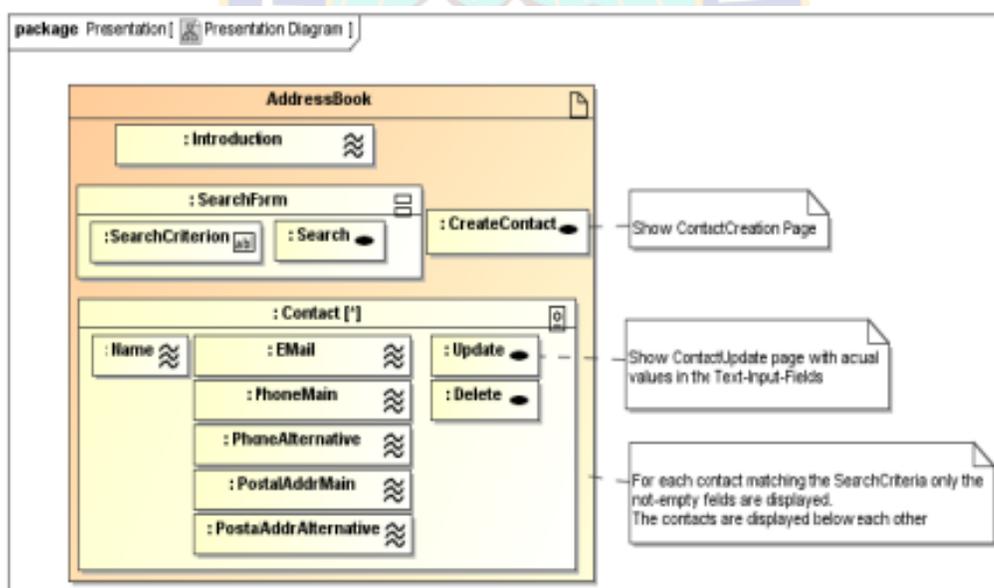


Figura 2.12 Modelo de Presentación
Fuente: Maximilians, 2016

2.6.1.5 MODELO DE PROCESO

Este modelo especifica las acciones que realiza cada clase de proceso, en este modelo se incluye:

- Modelo de Estructura de Proceso: Este define las relaciones entre las diferentes clases de proceso como se muestra en la Figura 2.13.

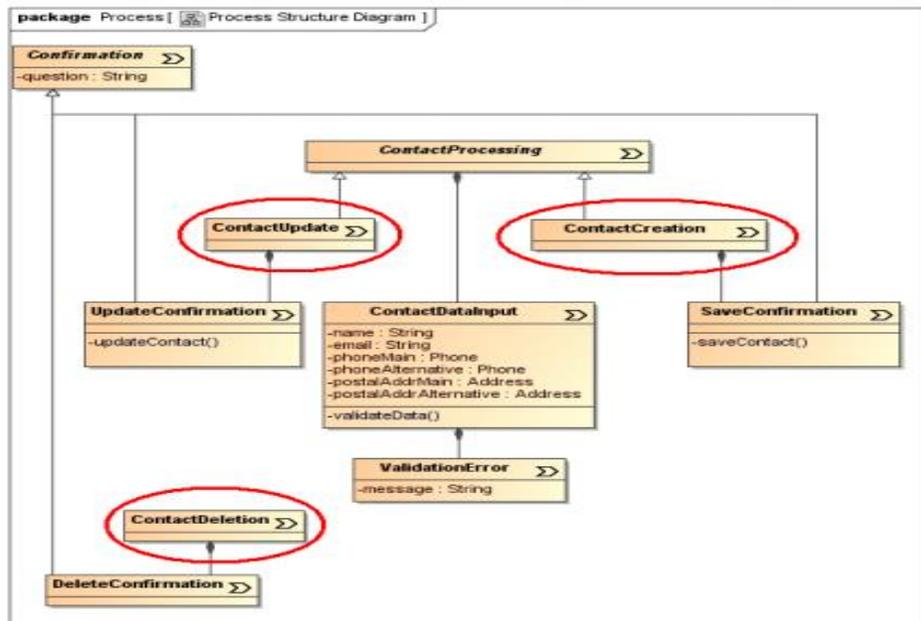


Figura 2.13 Modelo de Estructura de Proceso

Fuente: Maximilians, 2016

- Modelo de flujo de Procesos: que especifica las actividades conectadas con cada proceso. En la Figura 2.14 describe los comportamientos de una clase proceso. Lo que ocurre en detalle dentro de cada una.

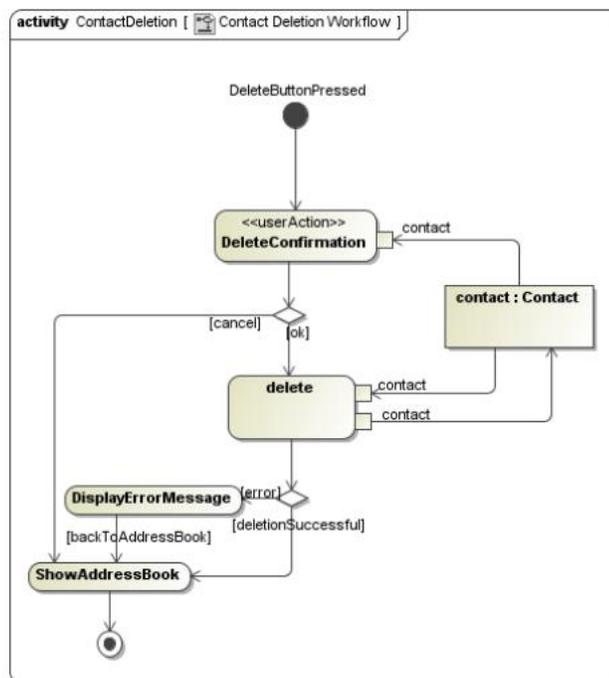


Figura 2.14 Modelo de Flujos de Procesos

Fuente: Maximilians, 2016

2.7 CALIDAD DEL SOFTWARE

La ISO 9126 es un estándar internacional para la evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-características consta de un conjunto de atributos que son medidos por una serie de métricas (Moreno, Toledo, López, & Cruz, 2020) .

WEBQEM toma las métricas del modelo de calidad ISO 9126 la cual da referencia a las siguientes características:

- **Usabilidad.** Grado en el que el software es fácil de usar, según lo indican los siguientes sub atributos: Entendible, Aprendible y Operable.
- **Funcionalidad.** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- **Facilidad de recibir mantenimiento.** Facilidad con la que pueden efectuarse reparaciones del software, según lo indican los siguientes subatributos: analizable, confiable, estable y susceptible de someterse a pruebas.
- **Portabilidad.** Facilidad con la que el software puede llevarse de un ambiente a otro, según lo indican los atributos siguientes: adaptable, instalable, conformidad y sustituible.

Estos factores de calidad ISO 9126 del software estudiados anteriormente no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas para evaluar la calidad del sistema.

2.8 SEGURIDAD

2.8.1 OWASP

Open Web Application Security Project. También conocido como OWASP, es un documento no oficial ni estándar donde se recopilan las peores vulnerabilidades, y las mas peligrosas de los últimos años. Como es habitual, cada 4 años, esta plataforma publica un informe actualizado con las vulnerabilidades más preocupantes y más explotadas por piratas informáticos. Igual que los años 2004, 2007, 2010 y 2013, en esta ocasión, la plataforma acaba

de hacer público su nuevo documento donde se habla del Top-10 de las peores vulnerabilidades de 2017 (Velasco, 2020).

A continuación, vamos a ver en detalle este Top-10 de vulnerabilidades de OWASP.

- A1 Injection, en esta categoría se engloban las vulnerabilidades que permiten la inyección de código en herramientas como SQL, NoSQL, OS o LDAP, lo que permite a los servidores interpretar una cadena como si fuese código, pudiendo acceder a la base de datos sin autorización.
- A2 Broken Authentication, fallos en las implementaciones de inicio o gestión de sesión que permite a los atacantes hacerse con contraseñas, claves o cualquier otra información para autenticarse en un sistema temporalmente o de forma permanente.
- A3 Sensitive Data Exposure, debido a fallos en la implementación de distintas APIs, muchas veces no se protege correctamente la información sensible, lo que permite a piratas informáticos hacerse, por ejemplo, con datos personales, bancarios o de salud de usuarios.
- A4 XML External Entities (XXE), por procesadores XML mal configurados pueden procesar ciertas referencias como si no se tratasen de entradas XML, lo que puede permitir revelar ficheros y recursos ocultos, e incluso ejecutar código o causar un ataque DoS.
- A5 Broken Access Control, errores en la configuración de los sistemas de control de acceso puede permitir a un atacante acceder a recursos y archivos para los que no debería tener permiso.
- A6 Security Misconfiguration, aquí se engloban todo tipo de fallos relacionados con la configuración de todo tipo de sistemas de seguridad, desde las conexiones HTTPS hasta las aplicaciones de seguridad de cualquier sistema o servidor.
- A7 Cross-Site Scripting (XSS), todo tipo de ataques XSS que se originan cuando una web incluye datos no validados para ejecutar algún script peligroso desde una web que, supuestamente, es de confianza, permitiendo así que la web en cuestión ejecute código en el equipo de la víctima.

- A8 Insecure Deserialization, Fallos en la serialización que pueden permitir la ejecución de código directamente en la memoria.
- A9 Using Components with Known Vulnerabilities, Como su nombre indica, en esta categoría se recogen todos los usos de librerías, frameworks y otros recursos de software que tienen vulnerabilidades, por lo que, al utilizarlos en una plataforma o un proyecto, automáticamente este se vuelve vulnerable, quedando en peligro a través de estos recursos.
- A10 Insufficient Logging & Monitoring, Debido a la falta de monitorización, la mayoría de las veces se tarda hasta 200 días en detectar una vulnerabilidad en un software o una plataforma web, tiempo que se podría reducir notablemente simplemente configurando mayores controles, mejor monitorización y más registros a estas plataformas.

2.9 HERRAMIENTAS DE DESARROLLO

En la Tabla 2.1 se mostrará las herramientas tecnológicas para implementación de este proyecto.

Tabla 2.1
Herramientas de Desarrollo

Herramientas	Versión	Descripción
Eclipse IDE	2020-06	Es una plataforma de desarrollo y herramienta para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.
Java	8	Es un lenguaje de programación es rápido, seguro y fiable.
PostgreSQL	9.4	Es un gestor de base de datos relacional orientado a objetos y de código abierto.
Thymeleaf	5	Es un motor de plantillas web.
Spring	2.3.4.	Es un framework para el desarrollo de aplicaciones web.
Spring Security	5.3.4.	Es un marco de autenticación y control de acceso potente y altamente personalizable. Es el estándar de facto para proteger las aplicaciones basadas en Spring.

JPA	2.3.4.	Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java
Bootstrap	4.4.	Es un framework front-end utilizado para desarrollar aplicaciones web.

Fuente: Elaboración Propia



CAPÍTULO 3

MARCO APLICATIVO

3.1 INTRODUCCIÓN

En el presente capítulo se realizará el desarrollo a la solución del problema planteado en el capítulo introductorio, el de implementar un sistema de encuesta a los titulados de la UMSA. Para el desarrollo del sistema se utilizará la metodología ágil SCRUM aplicando las tres fases para su desarrollo y la metodología UWE aplicando las 4 fases para su desarrollo.

En el Pre – Game es donde se analizará los requerimientos obtenidos para poder construir el Product Backlog del sistema de encuesta de la UMSA.

En el Game a partir del Product Backlog construido de la fase Pre – Game el sistema de encuesta se dividirá en 3 Sprints. En cada iteración se implementará la metodología UWE en el cual usaremos 4 de sus fases para que describa el desarrollo del sistema:

- Diagramas de casos de uso
- Modelo navegacional
- Modelo de presentación
- Modelo de procesos

En el Post – Game se realizará las pruebas de estrés, finalizando con las interfaces de pantalla para el documento final.

3.2 PRE – GAME

3.2.1 PLANIFICACIÓN

En la planificación se creó la lista priorizada del Product Backlog a partir del análisis de requerimientos. Los requerimientos funcionales que describen la interacción entre el sistema y el entorno son representados en la Tabla 3.1.

Tabla 3.1
Requerimientos del Sistema de Encuesta

Código	Requerimientos
RE 01	El sistema tiene que contar con autenticación de usuarios.
RE 02	El sistema debe manejar los siguientes roles: Administrador de roles, Encargado de la encuesta, Encargado de la información del titulado, Usuario titulado.
RE 03	El administrador debe poder agregar usuarios.
RE 04	El sistema debe de permitir crear, modificar y eliminar encuestas.
RE 05	El sistema debe permitir crear, modificar y eliminar secciones para la encuesta.
RE 06	El sistema debe permitir crear, modificar y eliminar preguntas para las secciones.
RE 07	El sistema debe permitir crear, modificar y eliminar respuestas para las preguntas.
RE 08	El sistema debe poder agregar, modificar y eliminar estudiantes titulados.
RE 09	El sistema debe poder mostrar información básica de los estudiantes titulados.
RE 10	El sistema debe poder mostrar el estado de llenado de la encuesta.
RE 11	El sistema debe permitir al estudiante titulado llenar la encuesta
RE 12	El sistema debe generar reportes mostrando a todos los estudiantes titulados que llenaron la encuesta, que no llenaron la encuesta y que no acabaron de llenar la encuesta.
RE 13	El sistema debe generar reportes mostrando los porcentajes de las preguntas realizadas.

Fuente: Elaboración Propia

3.2.2 CLASIFICACIÓN E IDENTIFICACIÓN DE LOS USUARIOS (ROLES)

En la Tabla 3.2 identificaremos los usuarios que interactuarán con el sistema para poder clasificarlos y diseñar las respectivas restricciones o tareas que tendrán dentro del sistema de encuesta.

Tabla 3.2
Roles de los usuarios del sistema

Usuario	Función
Administrador	Es el responsable de asignar los roles y agregar a los encargados.
Encargado de la encuesta	<p>Es el responsable de agregar, modificar y si es necesario eliminar las preguntas al sistema.</p> <p>Es el responsable de agregar a todos los estudiantes titulados para que realicen la encuesta.</p> <p>Podrá ver los reportes del porcentaje de las preguntas realizadas.</p>
Encargado de la información del titulado (Encuestador)	<p>Es el encuestador, es encargado de comunicarse con el estudiante titulado para brindarle toda la información de la encuesta.</p> <p>Es responsable de ver el estado en que se encuentra la encuesta, si está acabada o en proceso de llenado.</p> <p>Tiene la opción de ver la información de los estudiantes titulados.</p>
Estudiante titulado	Solo podrá realizar la encuesta y si es necesario podrá modificar su información.

Fuente: Elaboración Propia

3.3 GAME

En esta fase vamos a desarrollar todos los requerimientos de nuestro Product Backlog obtenidos de la Tabla 3.1, para esto se identificó los 5 Sprints que nos ayudara a desarrollar el sistema de una manera más ordenada como se muestra en la Figura 3.3.

Tabla 3.3
Descripción de los sprint

Iteraciones	Descripción
Primer Sprint	Gestión de Usuarios
Segundo Sprint	Administración de Encuesta
Tercer Sprint	Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante
Cuarto Sprint	Llenado de la Encuesta
Quinto Sprint	Realización de Reportes

Fuente: Elaboración Propia

El Product Backlog se prioriza por valor, según el valor se dará prioridad a cada ítem, la descripción de cada valor se describe en la Tabla 3.4

Tabla 3.4
Prioridades del Product Backlog

Prioridad	Descripción
Prioridad 1	Un ítem que tiene prioridad baja, ítem que puede esperar para dar prioridad a otros ítems de mayor grado si así lo requiere.
Prioridad 2	Un ítem que tiene la prioridad media, se tiene que trabajar en él.

Prioridad 3

Un ítem que tiene la prioridad alta, se tiene que trabajar en él lo antes posible.

Fuente: Elaboración Propia

3.3.1 PRIMER SPRINT

Para el desarrollo del Primer Sprint utilizaremos la metodología UWE para el desarrollo del módulo gestión de Usuarios, utilizando los requerimientos obtenidos de la Tabla 3.1

RE 01 El sistema tiene que contar con autenticación de usuarios.

RE 02 El sistema debe manejar los roles de Administrador de roles, Encargado de la encuesta, Encargado de la información del titulado, Usuario titulado.

RE 03 El administrador debe poder agregar usuarios.

A continuación, en la Tabla 3.5 se detallará el Product Backlog del Primer Sprint.

Tabla 3.5
Primer Sprint Product Backlog de Gestión de usuarios

Sprint		Historia	Inicio	Duración (Días)
Primer Sprint		Gestión de Usuarios	1-10-2020	15

ID	Tareas	Tipo	Prioridad	Estimación en días	Estado
1.1	Construir la base de datos en el gestor de base de datos.	Análisis	Prioridad 1	2	Terminado
1.2	Diseño del diagrama de casos de uso.	Análisis	Prioridad 1	1	Terminado
1.3	Diseño del modelo navegacional.	Análisis	Prioridad 2	1	Terminado

1.4	Diseño del modelo de presentación.	Análisis	Prioridad 2	1	Terminado
1.5	Diseño del modelo de procesos.	Análisis	Prioridad 2	1	Terminado
1.6	CRUD de los roles de usuario.	Codificación	Prioridad 2	2	Terminado
1.7	Autenticación de Usuarios.	Codificación	Prioridad 3	3	Terminado
1.8	Encriptado de contraseñas con BCryptPasswordEncoder.	Codificación	Prioridad 3	2	Terminado
1.9	Desarrollo de interfaz.	Codificación	Prioridad 3	2	Terminado

Fuente: Elaboración Propia

3.3.1.1 DIAGRAMAS DE CASOS DE USO

En el primer Sprint, el caso de uso detallara la gestión de usuarios en el cual se tendrá en cuenta, el responsable administrador podrá gestionar a los distintos usuarios para que sean admitidos en el sistema como se observa en la siguiente Figura 3.1.

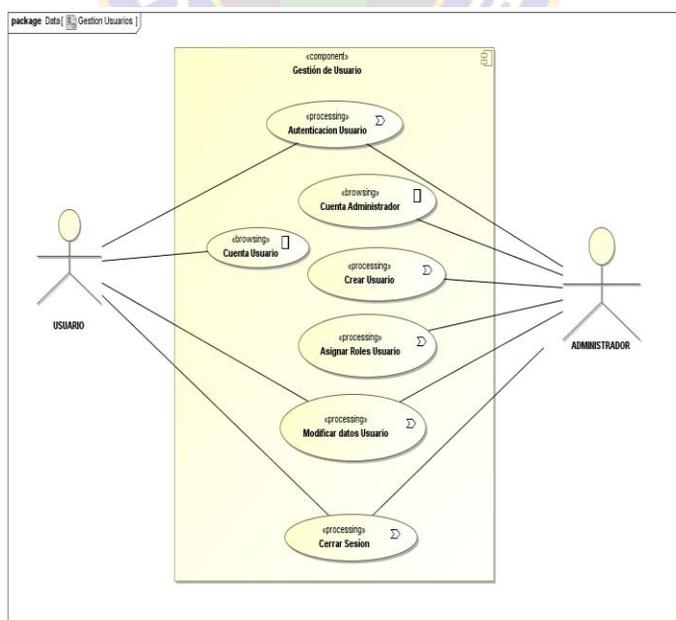


Figura 3.1 Diagrama de caso de uso Gestión de Usuarios

Fuente: Elaboración Propia

A continuación, se describirá de forma extendida el diagrama de caso de uso para el proceso de gestión de usuarios donde el usuario realizará las acciones e interacciones con el sistema definido en la siguiente Tabla 3.6

Tabla 3.6
Descripción de casos de uso Gestión usuario

Nombre	Gestión Usuario	Código	SE-01
Usuario	Administrador, Encargado de la Encuesta, Encuestador, Titulado.		
Descripción	Acceso al sistema, Administrar a los usuarios del sistema.		
Precondición	Esta registrado por el responsable del Sistema de Encuesta en la base de datos.		
Evento Actor	Evento Sistema		
Introducir Usuario y Contraseña para acceder al sistema.	<ul style="list-style-type: none"> • Compara el nombre de usuario con el de la base de datos. • Compara la contraseña encriptada con el de la base de datos. • No se acepta campos vacíos. 		
Ingreso al sistema.	<ul style="list-style-type: none"> • Muestra la interfaz adecuada para el administrador o usuario. 		
Agregar Usuario.	<ul style="list-style-type: none"> • El administrador completa un formulario con los campos necesarios para agregar un nuevo usuario. • El administrador puede asignar roles a los usuarios. 		
Eliminación de usuario.	<ul style="list-style-type: none"> • El administrador eliminara a los usuarios que ya no necesiten el usar sistema. 		
Cerrar Sesión.	<ul style="list-style-type: none"> • Destruye la sesión activa. 		
Post condición	Una vez accedido al sistema se tendrá los datos de los usuarios.		

Presunción

El usuario que ingrese al sistema tiene que estar registrado en la base de datos.

Fuente: Elaboración Propia

3.3.1.2 MODELO NAVEGACIONAL

En el modelo navegacional se detallará la información con respecto a las páginas y muestra la forma de interacción del usuario con el sistema como se muestra en la Figura 3.2

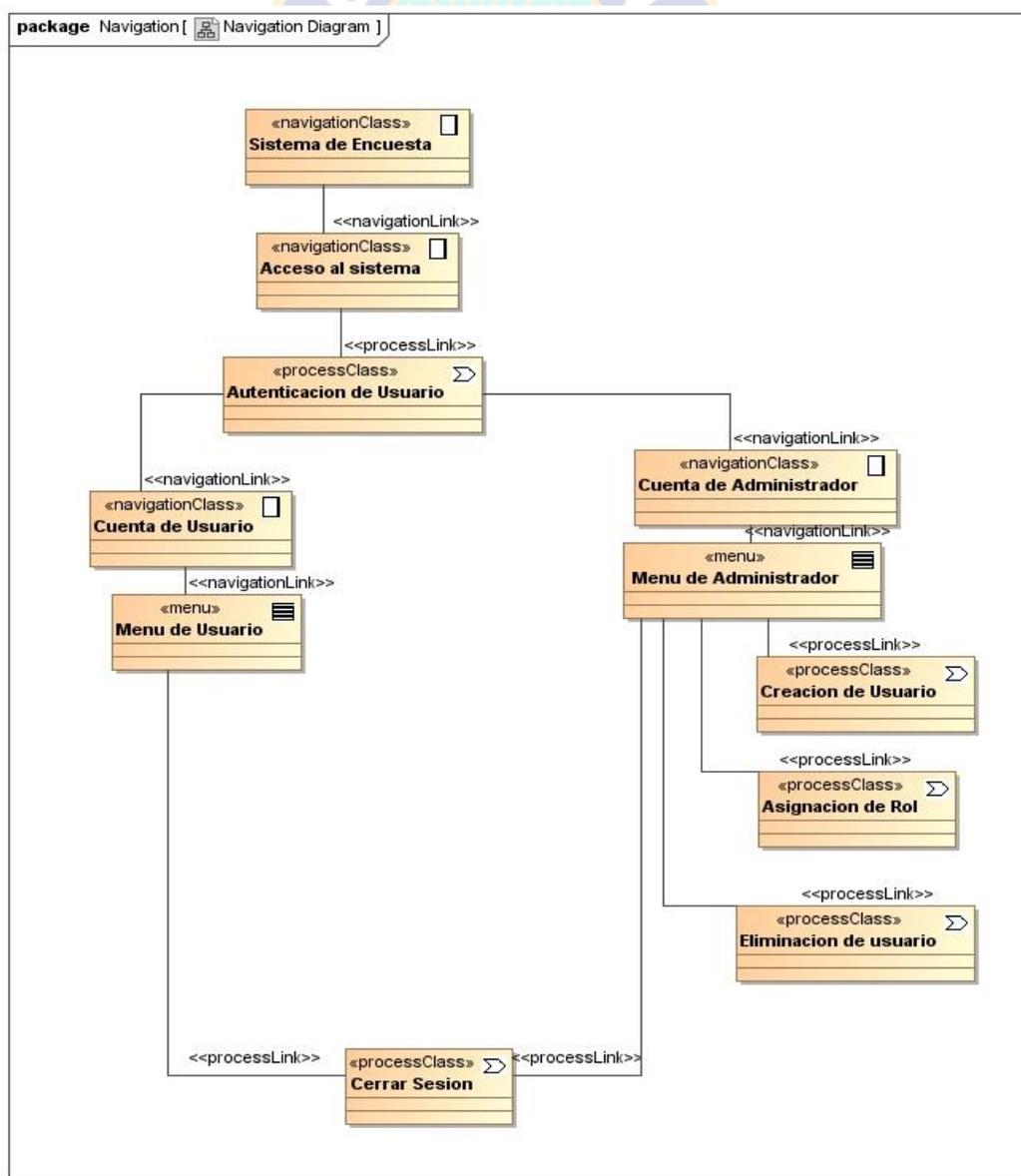


Figura 3.2 Diagrama Navegacional Gestión de Usuarios

Fuente: Elaboración Propia

3.3.1.3 MODELO DE PRESENTACIÓN

En el modelo de presentación nos mostrara la interfaz gráfica y sus elementos que formara el sistema.

- **Login de Usuario:** El usuario podrá ingresar al sistema con sus credenciales correctas que luego los llevará a las ventanas asignadas como se muestra en la Figura 3.3

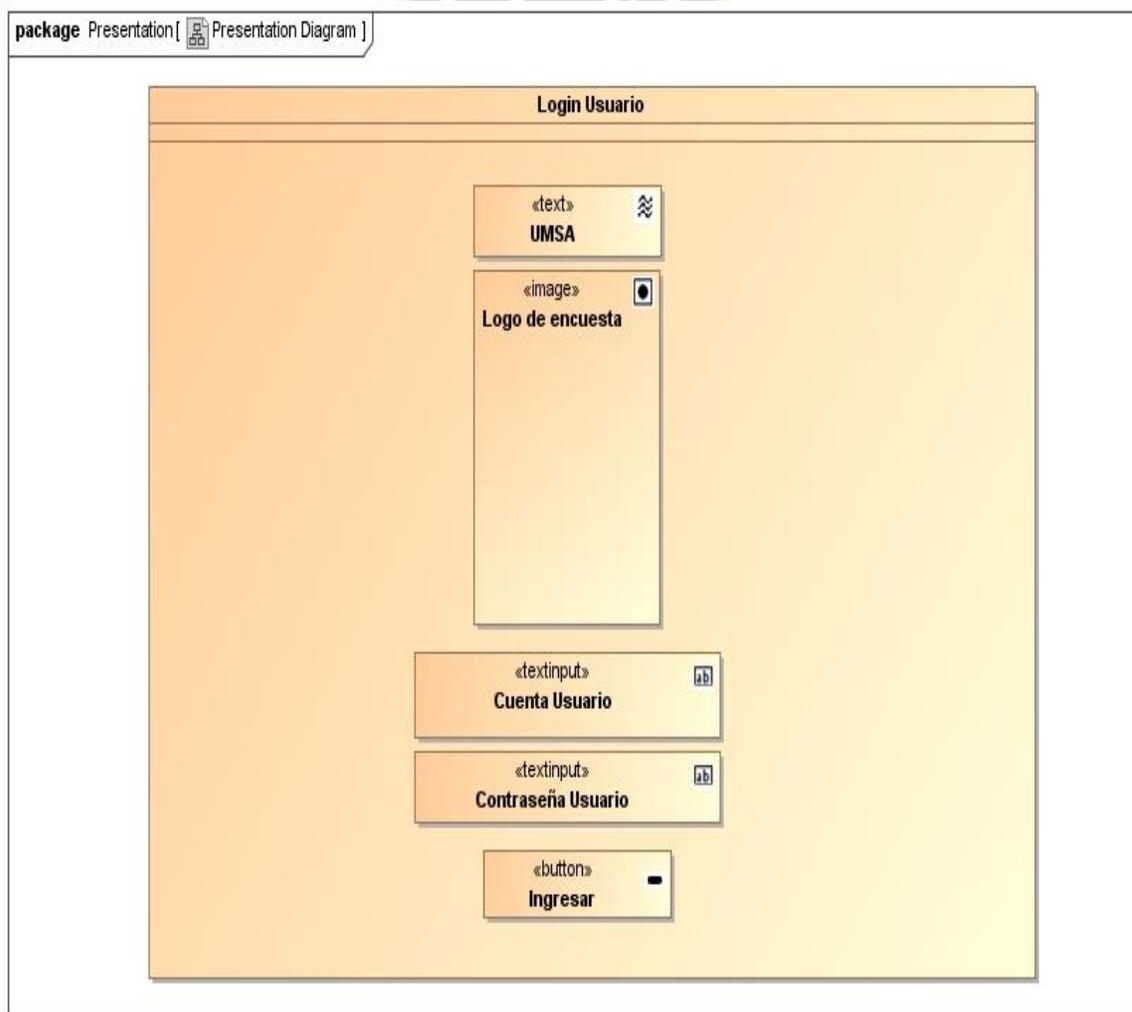


Figura 3.3 Diagrama de presentación – página de ingreso al sistema

Fuente: Elaboración Propia

- **Página del administrador:** En esta ventana el administrador podrá realizar la adición de nuevos usuarios y dar el rol respectivo para que puedan ingresar al sistema de encuestas.

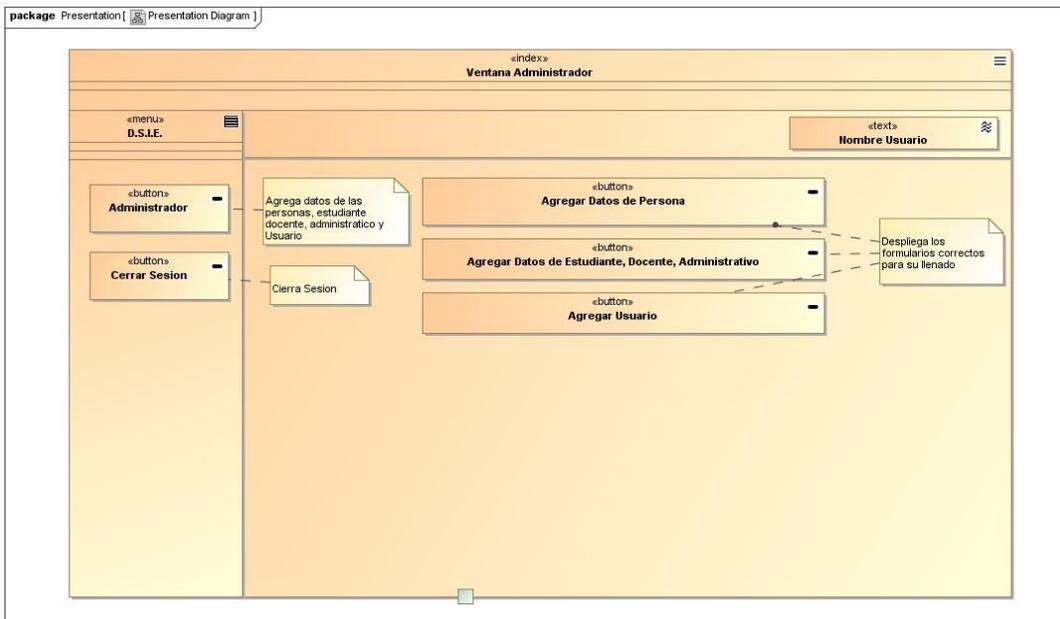


Figura 3.4 Diagrama de presentación – página de administrador Roles

Fuente: Elaboración Propia

3.3.1.4 MODELOS DE PROCESOS

Se refleja como es el flujo de acciones que realiza el administrador en la gestión de usuarios como se muestra en la Figura 3.5

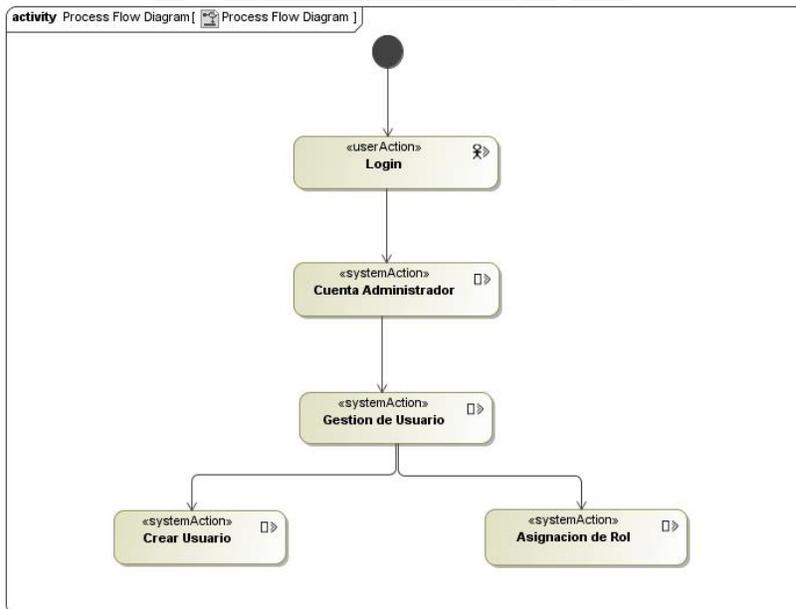


Figura 3.5 Diagrama de Procesos del Administrador

Fuente: Elaboración Propia

3.3.2 SEGUNDO SPRINT

Para el desarrollo de este sprint utilizaremos de apoyo la metodología UWE, también los requerimientos del Sistema de Encuesta que se mostró en la Tabla 3.1

RE 04 El sistema debe de permitir crear, modificar y eliminar encuestas.

RE 05 El sistema debe permitir crear, modificar y eliminar secciones para la encuesta.

RE 06 El sistema debe permitir crear, modificar y eliminar preguntas para las secciones.

RE 07 El sistema debe permitir crear, modificar y eliminar respuestas para las preguntas.

RE 08 El sistema debe poder agregar, modificar y eliminar estudiantes titulados.

A continuación, en la Tabla 3.7 se detallará el Product Backlog del Segundo Sprint.

Tabla 3.7
Segundo Sprint Product Backlog Administración de Encuesta

Sprint		Historia	Inicio	Duración (Días)	
Segundo Sprint		Administración de Encuesta	16-10-2020	15	
ID	Tareas	Tipo	Prioridad	Estimación en días	Estado
2.1	Planificación de Tareas	Análisis	Prioridad 1	1	Terminado
2.2	Diseño de casos de uso	Análisis	Prioridad 1	1	Terminado
2.3	Diseño del modelo navegacional	Análisis	Prioridad 2	1	Terminado
2.4	Diseño del modelo de presentación	Análisis	Prioridad 2	1	Terminado
2.5	Diseño del modelo de procesos	Análisis	Prioridad 2	1	Terminado
2.6	CRUD de la encuesta	Codificación	Prioridad 3	1	Terminado

2.7	CRUD de las secciones de la encuesta	Codificación	Prioridad 3	1	Terminado
2.8	CRUD de las preguntas de la sección	Codificación	Prioridad 3	2	Terminado
2.9	CRUD de las respuestas de las preguntas	Codificación	Prioridad 3	2	Terminado
2.10	CRUD de los estudiantes titulados	Codificación	Prioridad 3	2	Terminado
2.11	Diseño de la interfaz	Codificación	Prioridad 3	2	Terminado

Fuente: Elaboración Propia

3.3.2.1 DIAGRAMA DE CASO DE USO

Para este Sprint de administración de encuesta se tomará en cuenta al Encargado de encuesta, Como se muestra en la Figura 3.6

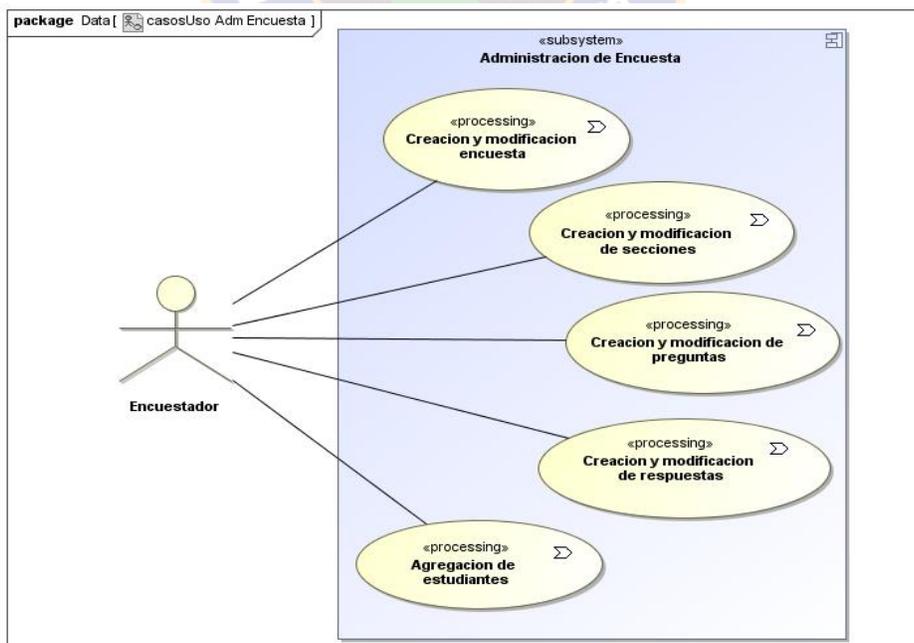


Figura 3.6 Diagrama de Caso de Uso Administración de Encuesta

Fuente: Elaboración Propia

A continuación, se describe en la Tabla 3.8 el caso de uso que sigue el encuestador al interactúa con el sistema en la parte de Administración de encuesta como se mostró en la Figura 3.6

Tabla 3.8
Descripción de caso de uso de Administración de Encuesta

Nombre	Administración de Encuesta	Código	SE-02
Usuario	Encargado de la Encuesta.		
Descripción	Acceso al sistema, Creación de la encuesta y agregación de estudiantes.		
Precondición	El encargado debe estar registrado en la base de datos y contar con el rol de encuestador.		
Evento Actor	Evento Sistema		
Creación y modificación de la encuesta.	<ul style="list-style-type: none"> • El encuestador podrá adicionar distintas encuestas en el sistema. • El encuestador podrá modificar las distintas encuestas que adiciono en el sistema. • El encuestador podrá eliminar la encuesta. 		
Creación y modificación de las secciones	<ul style="list-style-type: none"> • El encuestador podrá adicionar secciones en la encuesta para que esta se pueda mantener ordenada. • El encuestador podrá modificar las secciones de la encuesta que adiciono. • El encuestador podrá eliminar la sección. 		
Creación y modificación de las preguntas	<ul style="list-style-type: none"> • El encuestador podrá adicionar las preguntas en la sección. 		

	<ul style="list-style-type: none"> • El encuestador podrá modificar las preguntas que adicione. • El encuestador podrá eliminar las preguntas.
Creación y modificación de las respuestas	<ul style="list-style-type: none"> • El encuestador podrá adicionar las respuestas para las preguntas. • El encuestador podrá modificar las respuestas. • El encuestador podrá eliminar las respuestas.
Agregación de Estudiantes	<ul style="list-style-type: none"> • El encuestador podrá adicionar las a los estudiantes titulados para que realicen la encuesta.
Post condición	El sistema guarda toda la información generada por el encuestador y despliega de forma dinámica la información.
Presunción	El usuario que ingresa debe tener el rol de Encuestador.

Fuente: Elaboración Propia.

3.3.2.2 MODELO NAVEGACIONAL

En la siguiente Figura 3.7 Se detallarán la información con respecto a las páginas y muestra la forma de interacción del encuestador con el sistema de encuesta.

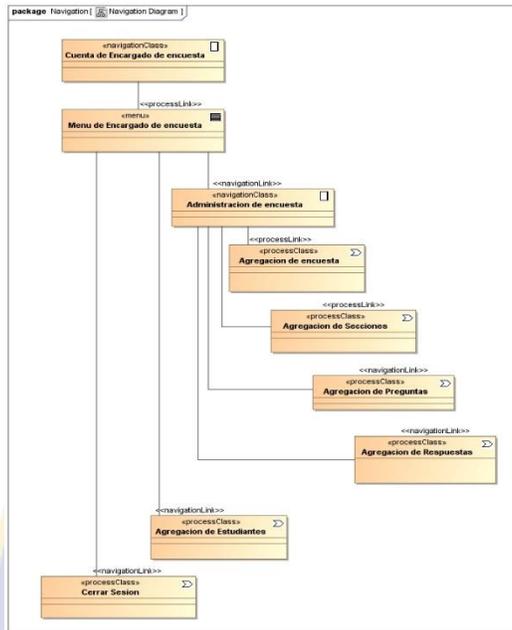


Figura 3.7 Diagrama Navegacional Administración de Encuesta
Fuente: Elaboración Propia.

3.3.2.3 MODELO DE PRESENTACION

En el diseño del modelo de presentación nos mostrara la interfaz gráfica y sus elementos que formara el sistema como se muestra en la siguiente Figura 3.8

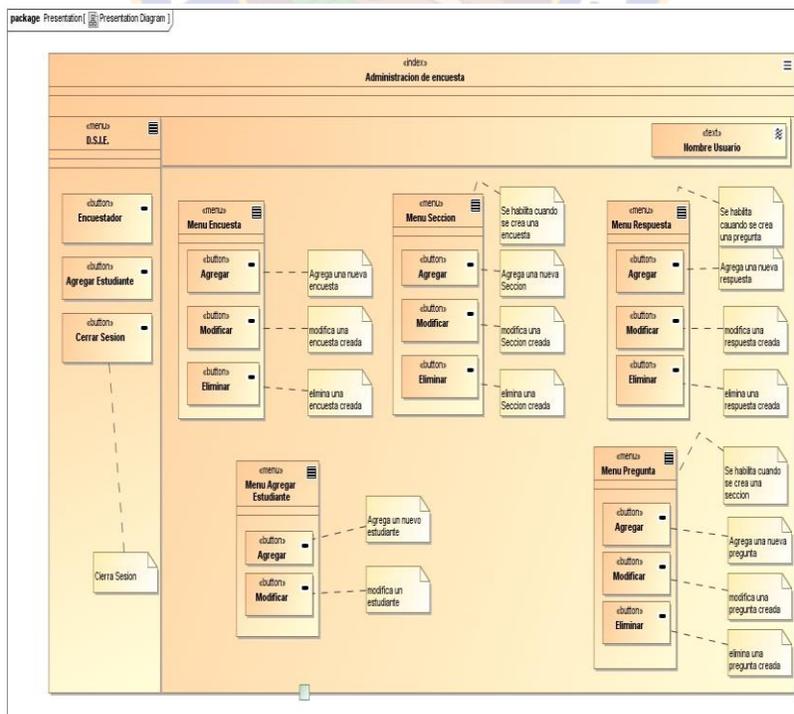


Figura 3.8 Diagrama de Presentación Administración de Encuesta
Fuente: Elaboración Propia

3.3.2.4 MODELO DE PROCESOS

En el modelo de procesos reflejaremos el flujo de procesos y acciones que realizara el Encargado de encuesta con muestra en la figura 3.9

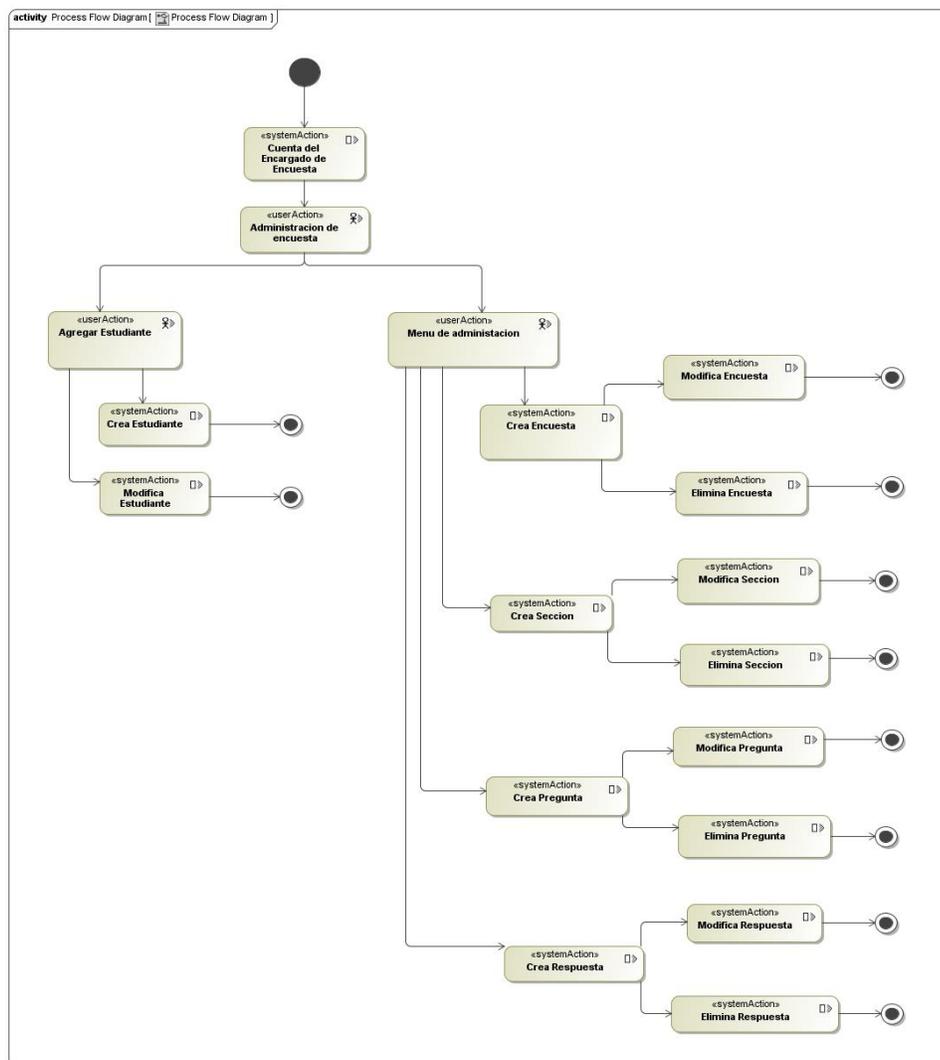


Figura 3.9 Diagrama de Procesos Administración de Encuesta

Fuente: Elaboración Propia

3.3.3 TERCER SPRINT

Para el desarrollo del Tercer Sprint muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante, utilizaremos la metodología de apoyo UWE utilizando los requerimientos obtenidos en la Tabla 3.1.

RE 9 El sistema debe poder mostrar información básica de los estudiantes titulados.

RE 10 El sistema debe poder mostrar el estado de llenado de la encuesta.

A continuación, en la Tabla 3.9 se detallará el Product Backlog del tercer Sprint.

Tabla 3.9

Tercer Sprint Product Backlog Muestreo de la información y estado de llenado

Fuente: Elaboración Propia

Sprint		Historia	Inicio	Duración (Días)
Tercer Sprint		Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.	31-10-2020	15

ID	Tareas	Tipo	Prioridad	Estimación en días	Estado
3.1	Planificación de Tareas.	Análisis	Prioridad 1	1	Terminado
3.2	Diseño de casos de uso.	Análisis	Prioridad 1	1	Terminado
3.3	Diseño del modelo navegacional.	Análisis	Prioridad 2	2	Terminado
3.4	Diseño del modelo de presentación.	Análisis	Prioridad 2	2	Terminado
3.5	Diseño del modelo de procesos.	Análisis	Prioridad 2	2	Terminado
3.6	Diseño de la página donde se mostrará la información del estudiante.	Codificación	Prioridad 3	2	Terminado
3.7	Diseño de la página donde se mostrar el	Codificación	Prioridad 3	2	Terminado

estado de llenado de la encuesta

3.8 Diseño de la interfaz Codificación Prioridad 3 3 Terminado

Fuente: Elaboración Propia

3.3.3.1. DIAGRAMAS DE CASO DE USO

Para este Sprint de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante se tomará en cuenta al Encuestador como se muestra en Figura 3.10

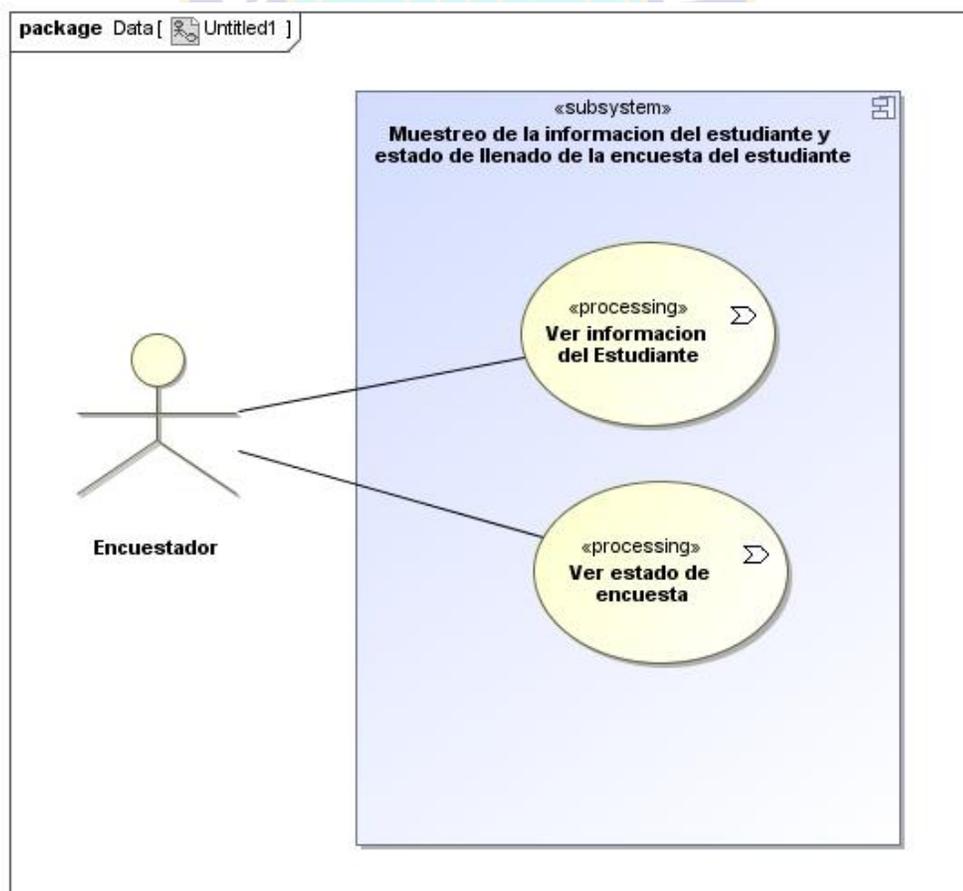


Figura 3.10 Diagrama de caso de uso Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.

Fuente: Elaboración Propia

A continuación, se describe en la Tabla 3.10 el caso de uso que sigue el encargado de la información del titulado (Encuestador), que se encarga de observar la información y la encuesta que responden los estudiantes titulados de la Figura 3.10

Tabla 3.10
Descripción de caso de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.

Nombre	Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.	Código	SE-03
Usuario	Encargado de la información del titulado (Encuestador).		
Descripción	Acceso al sistema, observar la información del estudiante y el estado de la encuesta.		
Precondición	El encuestador debe estar registrado en la base de datos y contar con el rol de encargado de la información del titulado (Encuestador).		
Evento Actor	Evento Sistema		
Ver información del estudiante	<ul style="list-style-type: none"> • Solo podrá ver la información del estudiante titulado. 		
Ver estado de la encuesta	<ul style="list-style-type: none"> • Podrá ver el estado de la encuesta si está en proceso de llenado, en llenado o no ingreso a la encuesta. 		
Presunción	El usuario que ingresa debe tener el rol de encargado de la información del titulado (Encuestador).		

Fuente: Elaboración Propia

3.3.3.1 MODELO NAVEGACIONAL

Se puede observar en la implementación del modelo navegacional del tercer sprint, los procesos son ejecutados por el responsable encargado de la información del titulado (Encuestador) como se muestra en la Figura 3.11.

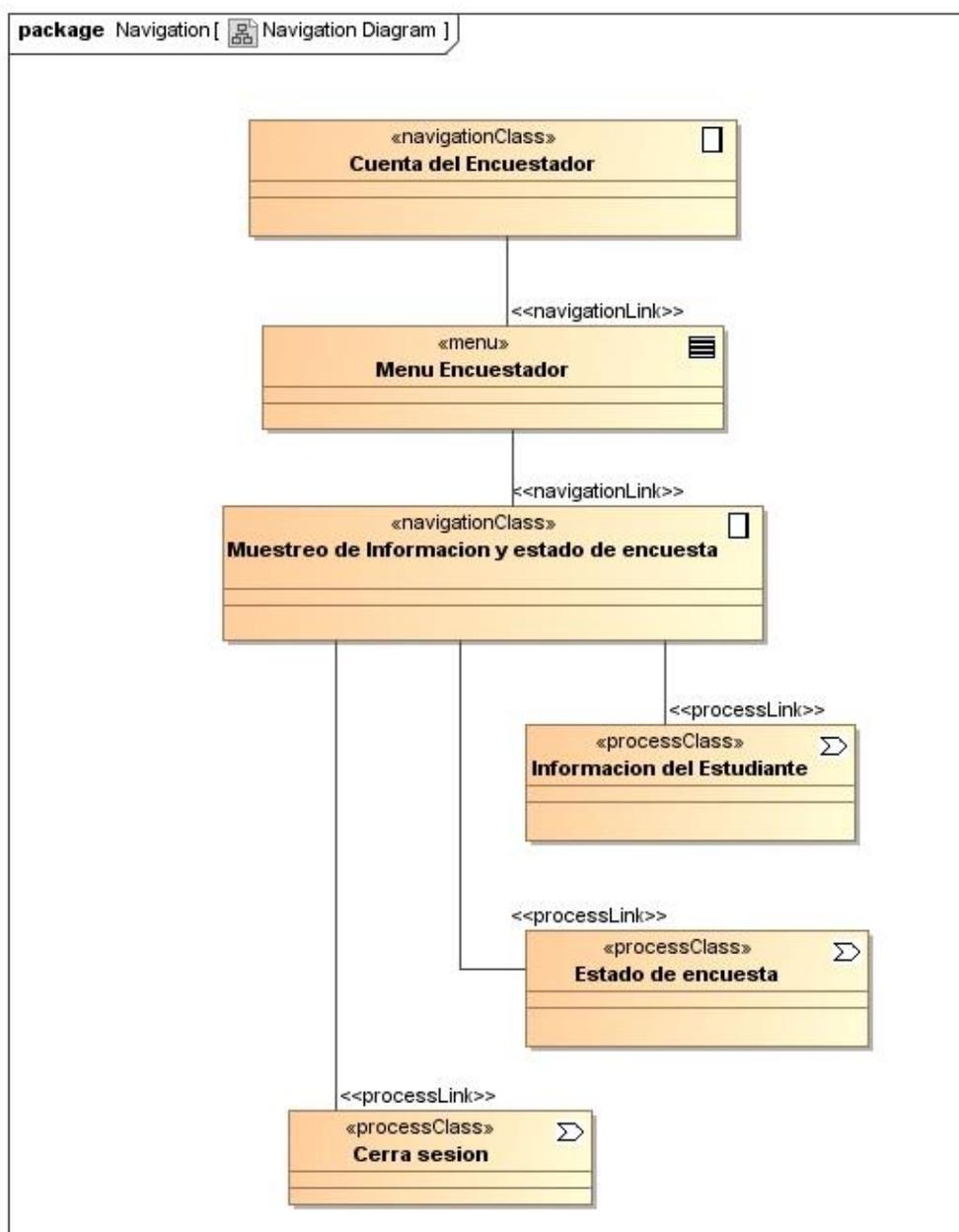


Figura 3.11 Diagrama Navegacional de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.

Fuente: Elaboración Propia

3.3.3.2 MODELO DE PRESENTACIÓN

El diseño del modelo de presentación mostrará las tareas que tendrá el Encargado de la información del titulado (Encuestador) como se muestra en la Figura 3.12.

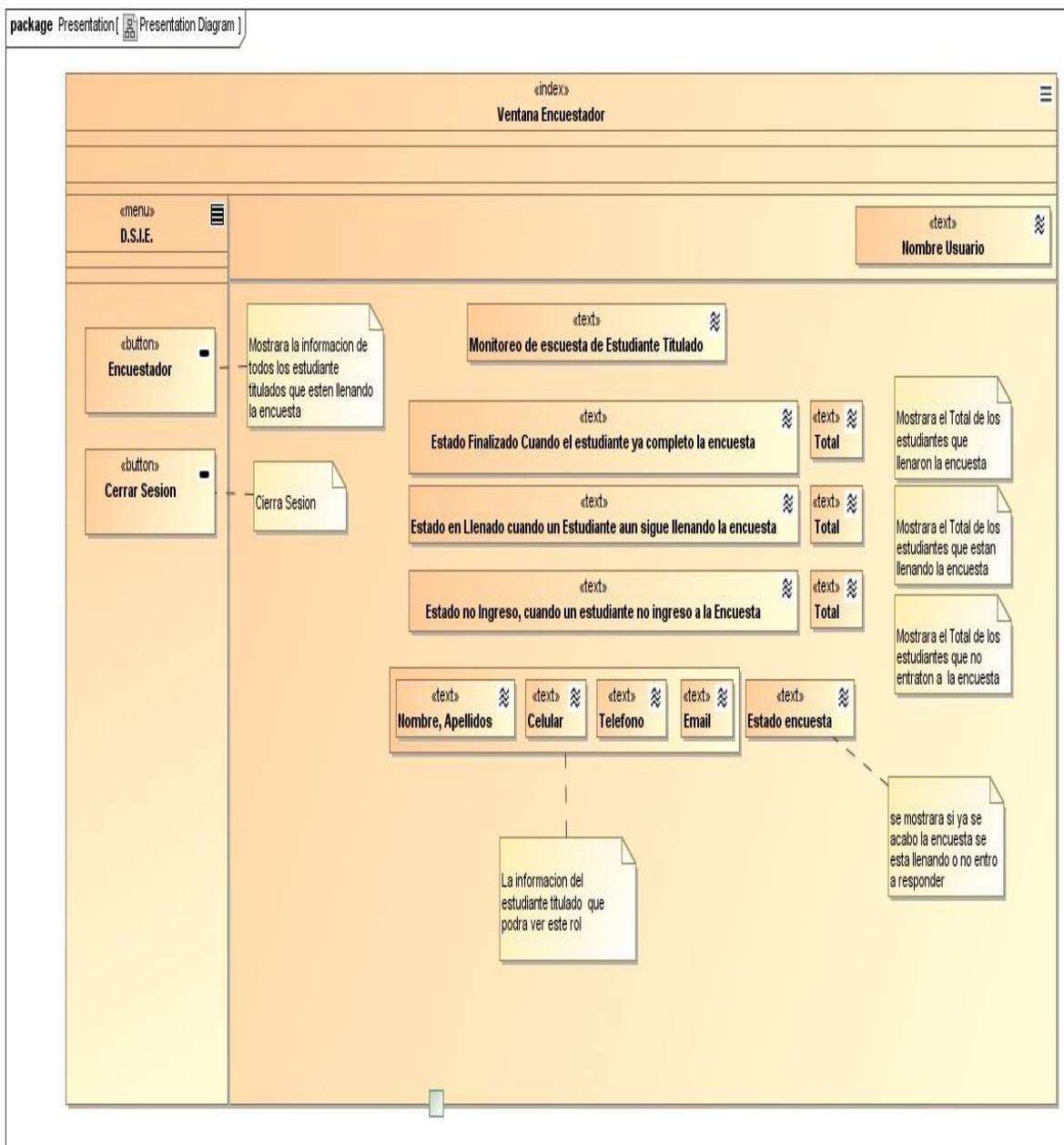


Figura 3.12 Diagrama de presentación de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.

Fuente: Elaboración Propia.

3.3.3.3 MODELO DE PROCESOS

Para el modelo de procesos veremos el flujo de acciones que realiza el Encargado de la información del titulado (Encuestador) como se muestra en la Figura 3.13

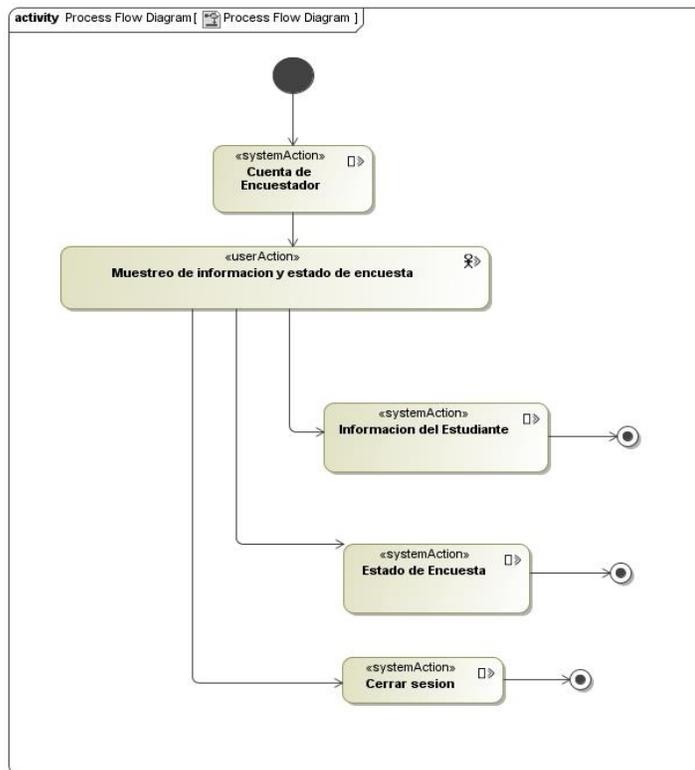


Figura 3.13 Diagrama de procesos de Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante.

Fuente: Elaboración Propia.

3.3.4 CUARTO SPRINT

Para el desarrollo del Cuarto Sprint Llenado de la Encuesta, utilizaremos la metodología de apoyo UWE utilizando los requerimientos obtenidos en la Tabla 3.1 y se detallara en la Tabla 3.11

RE 11 El sistema debe permitir al estudiante titulado llenar la encuesta.

Tabla 3.11

Cuarto Sprint Product Backlog de Llenado de la Encuesta.

Sprint	Historia	Inicio	Duración (Días)
Cuarto Sprint	Llenado de la Encuesta	15-11-2020	10

ID	Tareas	Tipo	Prioridad	Estimación en días	Estado
4.1	Planificación de Tareas.	Análisis	Prioridad 1	1	Terminado
4.2	Diseño de casos de uso.	Análisis	Prioridad 1	1	Terminado
4.3	Diseño del modelo navegacional.	Análisis	Prioridad 2	1	Terminado
4.4	Diseño del modelo de presentación.	Análisis	Prioridad 2	1	Terminado
4.5	Diseño del modelo de procesos.	Análisis	Prioridad 2	2	Terminado
4.6	Diseño de la página donde se mostrará la encuesta para que el estudiante lo pueda responder.	Codificación	Prioridad 3	2	Terminado
4.8	Diseño de la interfaz	Codificación	Prioridad 3	2	Terminado

Fuente: Elaboración Propia

3.3.4.1 DIAGRAMAS DE CASO DE USO

Para este Sprint de Llenado de la Encuesta se tomará en cuenta al Estudiante titulado como se muestra en Figura 3.14

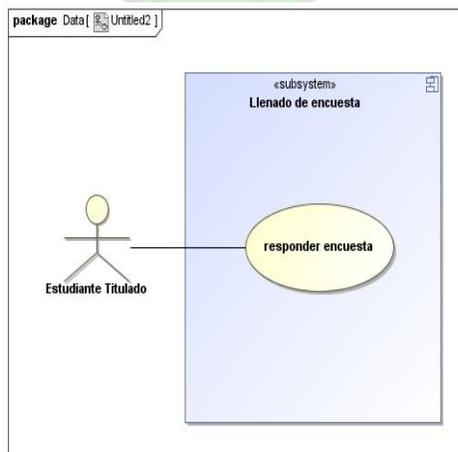


Figura 3.14 Diagrama de caso de uso Llenado de encuesta.

Fuente: Elaboración Propia

A continuación, se describe el caso de uso que sigue el estudiante titulado, que se encarga de llenar la encuesta en la Tabla 3.12

Tabla 3.12

Descripción del Caso de Uso Llenado de encuesta.

Nombre	Llenado de encuesta	Código	SE-04
Usuario	Estudiante Titulado.		
Descripción	Acceso al sistema, responder la encuesta.		
Precondición	El estudiante debe estar registrado en la base de datos y contar con el rol de Estudiante Titulado.		
Evento Actor	Evento Sistema		
Responder la Encuesta	<ul style="list-style-type: none"> El estudiante titulado deberá responder las encuestas que tiene disponible 		
Presunción	El usuario que ingresa debe tener el rol de Estudiante titulado.		
Post condición	Se guardará toda la información que proporcione el estudiante.		

Fuente: Elaboración Propia

3.3.4.2 MODELO NAVEGACIONAL

Se puede observar en la implementación del modelo navegacional del cuarto sprint llenado se encuesta, los procesos son ejecutados por el estudiante titulado como se muestra en la Figura 3.15

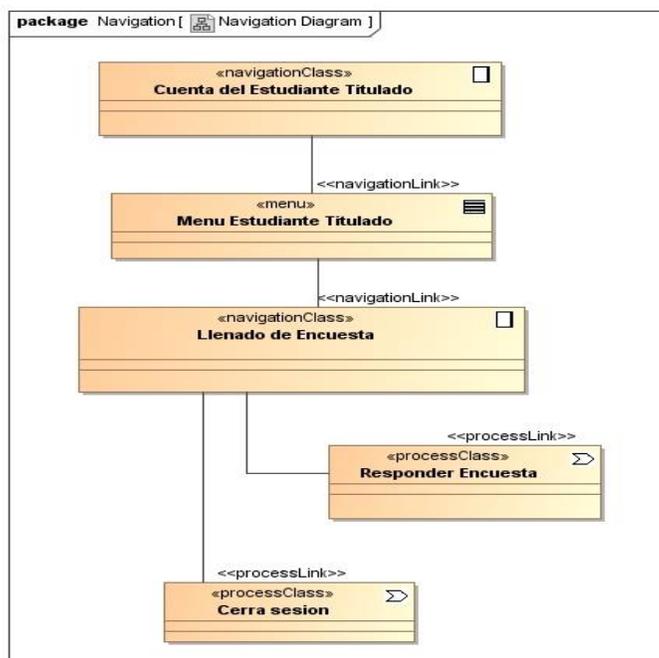


Figura 3.15 Diagrama Navegacional Llenado de encuesta.

Fuente: Elaboración Propia

3.3.4.3 MODELO DE PRESENTACIÓN

El diseño del modelo de presentación mostrará las tareas que tendrá el Estudiante titulado como se muestra en la Figura 3.16.

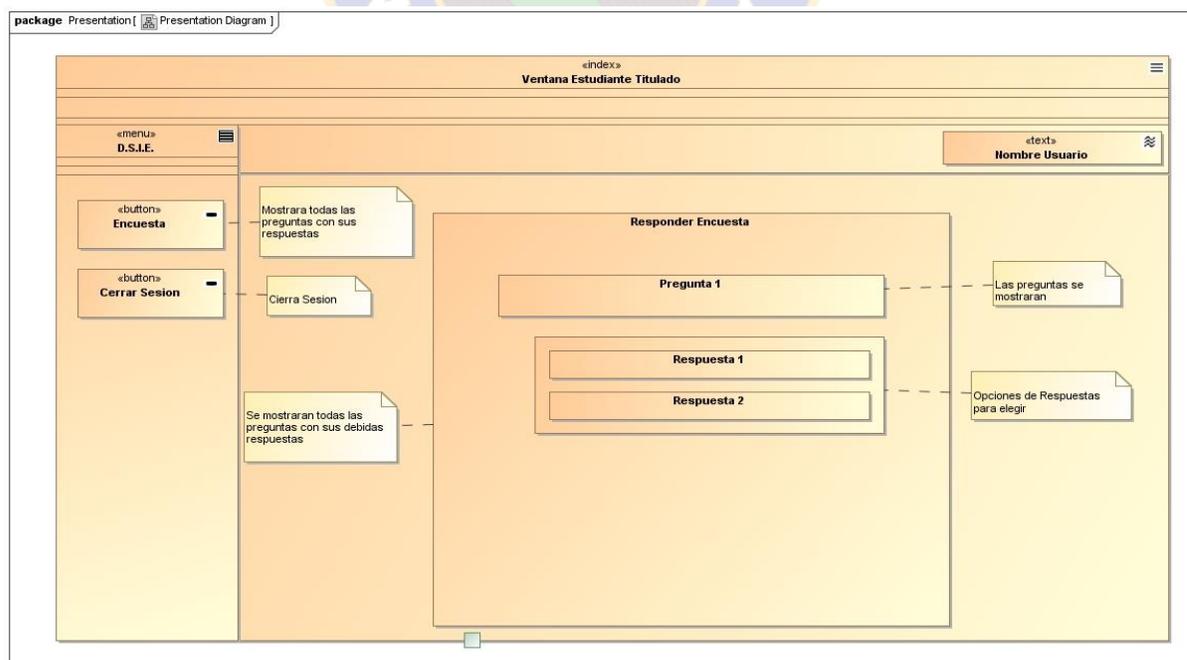


Figura 3.16 Diagrama de presentación del Estudiante Titulado para el llenado de encuesta.

Fuente: Elaboración Propia.

3.3.4.4 MODELO DE PROCESOS

Para el modelo de procesos veremos el flujo de acciones que realiza el Estudiante titulado como se muestra en la Figura 3.17

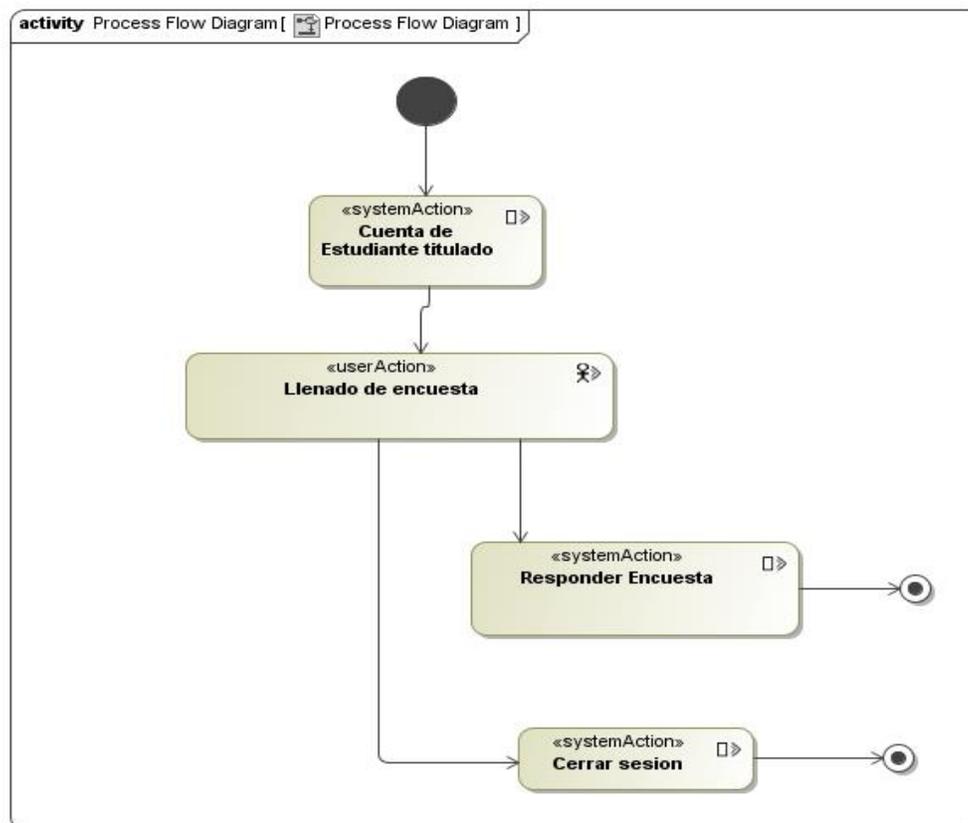


Figura 3.17 Diagrama de Proceso de Llenado de Encuesta.

Fuente: Elaboración Propia.

3.3.5 QUINTO SPRINT

Para el desarrollo del quinto Sprint Llenado de la Encuesta, utilizaremos la metodología de apoyo UWE utilizando los requerimientos obtenidos en la Tabla 3.1 y se detallara en la Tabla 3.13

RE 12 El sistema debe generar reportes mostrando a todos los estudiantes titulados que llenaron la encuesta, que no llenaron la encuesta y que no acabaron de llenar la encuesta.

RE 13 El sistema debe generar reportes mostrando los porcentajes de las preguntas realizadas.

Tabla 3.13
Cuarto Sprint Product Backlog de Realización de Reportes.

Sprint		Historia	Inicio	Duración (Días)	
Quinto Sprint		Realización de Reportes	25-11-2020	10	
ID	Tareas	Tipo	Prioridad	Estimación en días	Estado
5.1	Planificación de Tareas.	Análisis	Prioridad 1	1	Terminado
5.2	Diseño de casos de uso.	Análisis	Prioridad 1	1	Terminado
5.3	Diseño del modelo navegacional.	Análisis	Prioridad 2	1	Terminado
5.4	Diseño del modelo de presentación.	Análisis	Prioridad 2	1	Terminado
5.5	Diseño del modelo de procesos.	Análisis	Prioridad 2	1	Terminado
5.6	Generar reporte de estado de encuesta	Codificación	Prioridad 3	1	Terminado
5.7	Generar reporte de porcentaje de preguntas realizadas	Codificación	Prioridad 3	2	Terminado
5.8	Diseño de la interfaz	Codificación	Prioridad 3	2	Terminado

Fuente: Elaboración Propia

3.3.5.1 DIAGRAMAS DE CASO DE USO

Para este Sprint de Realización de Reportes se tomará en cuenta al Estudiante titulado como se muestra en Figura 3.18.

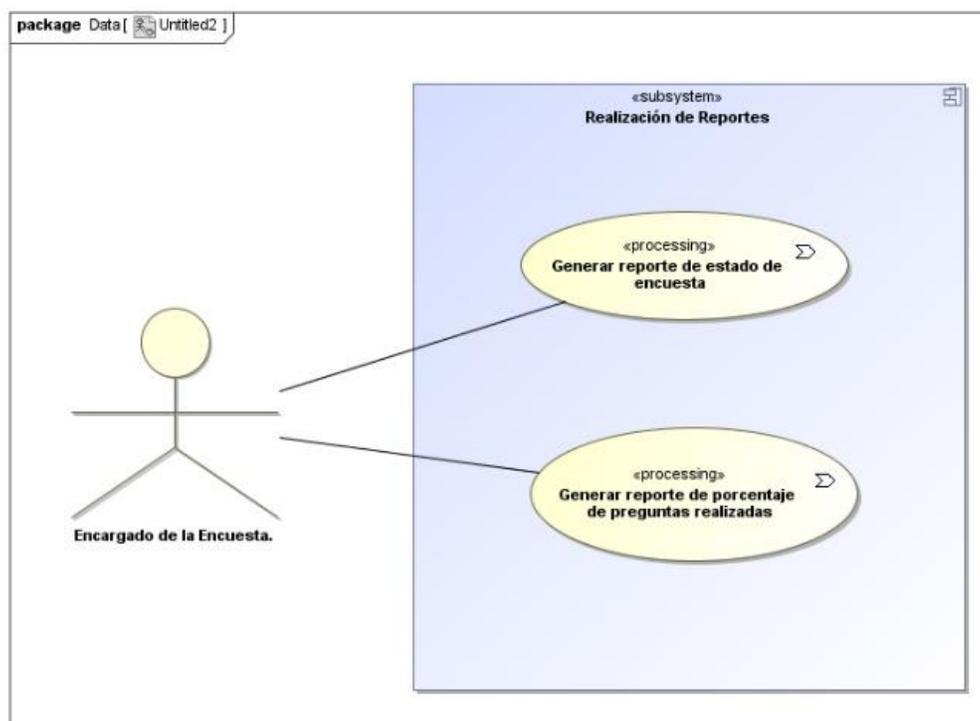


Figura 3.18 Diagrama de caso de uso Realización de Reportes.

Fuente: Elaboración Propia

A continuación, se describe el caso de uso que sigue el encargado de encuesta, que se encarga de generar reportes como se muestra en la Figura 3.14

Tabla 3.14

Descripción del Caso de Uso Realización de Reportes.

Nombre	Realización de Reportes	Código	SE-05
Usuario	Encardo de Encuesta		
Descripción	Acceso al sistema, generar reportes		
Precondición	El usuario debe estar registrado en la base de datos y contar con el rol de Encardo de Encuesta.		
Evento Actor	Evento Sistema		

Generar reportes

- El encargado de Encuesta podrá generar reportes de las encuestas de los estudiantes titulados que llenaron la encuesta, que no llenaron la encuesta y que no acabaron de llenar la encuesta.
- El encargado de Encuesta podrá generar reportes de las encuestas mostrando los porcentajes de las preguntas realizadas.

Presunción	El usuario que ingresa debe tener el rol de Encargado de Encuesta.
Post condición	Obtención de archivos de utilidad de la encuesta que se llevó a cabo.

Fuente: Elaboración Propia

3.3.5.2 MODELO NAVEGACIONAL

Se puede observar en la implementación del modelo navegacional del quinto sprint, los procesos que son ejecutados por el encargado de encuesta como se muestra en la Figura 3.19

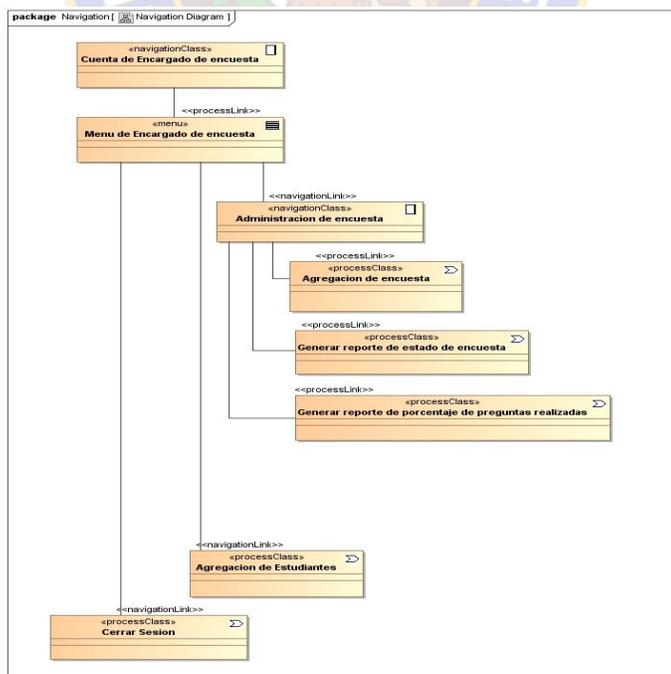


Figura 3.19 Diagrama Navegacional Realización de Reportes.

Fuente: Elaboración Propia

3.3.5.3 MODELO DE PRESENTACIÓN

El diseño del modelo de presentación mostrará los reportes que se pueden generar para el encargado de encuesta como se muestra en la Figura 3.20

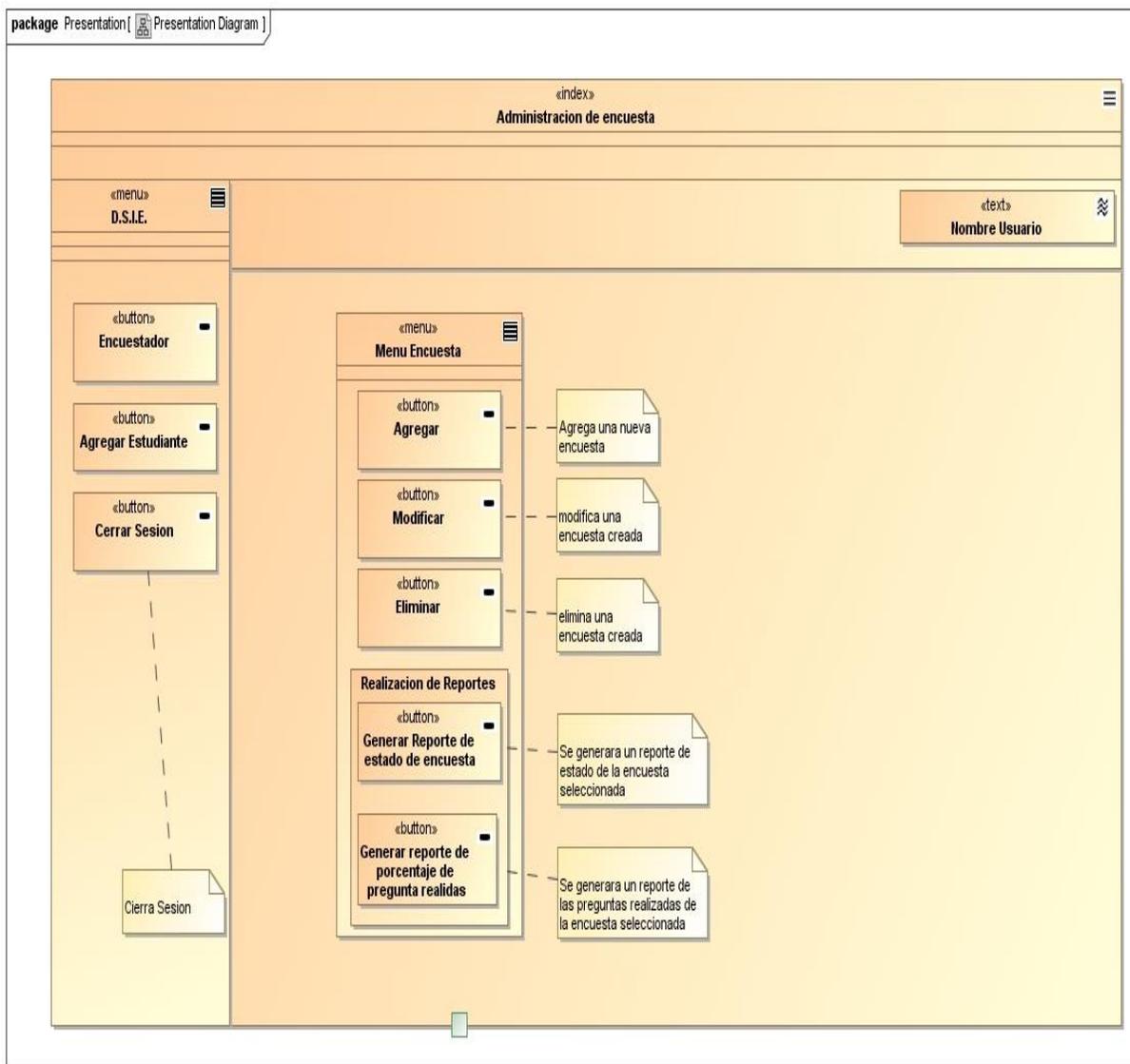


Figura 3.20 Diagrama de presentación Realización de Reportes.

Fuente: Elaboración Propia.

3.3.5.4 MODELO DE PROCESOS

Para el modelo de procesos veremos el flujo de acciones que realiza el Estudiante titulado como se muestra en la Figura 3.21

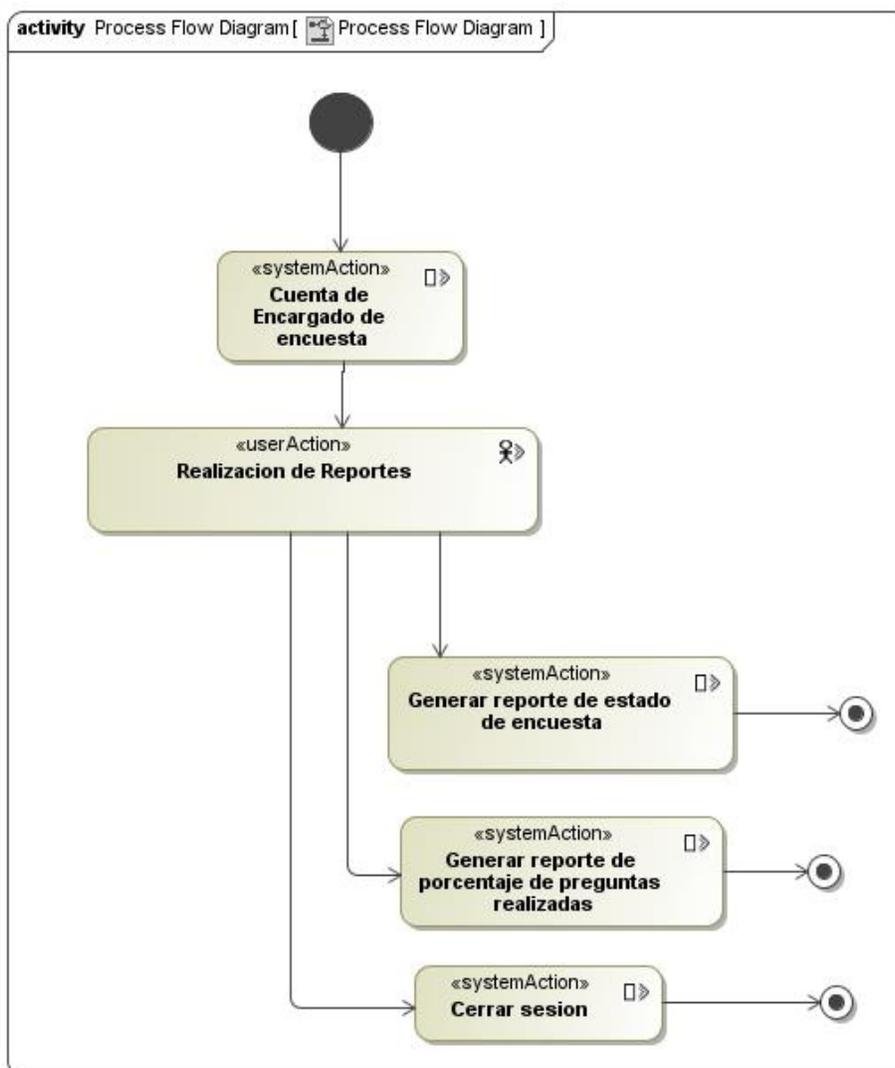


Figura 3.21 Diagrama de procesos de Realización de Reportes.

Fuente: Elaboración Propia.

3.3.6 DISEÑO FINAL

3.3.6.1 DIAGRAMA ENTIDAD RELACIÓN

Para este diagrama modelaremos los datos que permiten representar las entidades del sistema, así como sus interrelaciones y propiedades. Se observarán los atributos de cada entidad, su llave primaria y como están relacionadas las entidades.

En la siguiente Figura 3.22 se mostrará el diagrama entidad relación, que se obtuvo a partir de los requerimientos del sistema de encuestas, para que se vea el modelado de sus datos que permitirá representar las entidades relevantes.

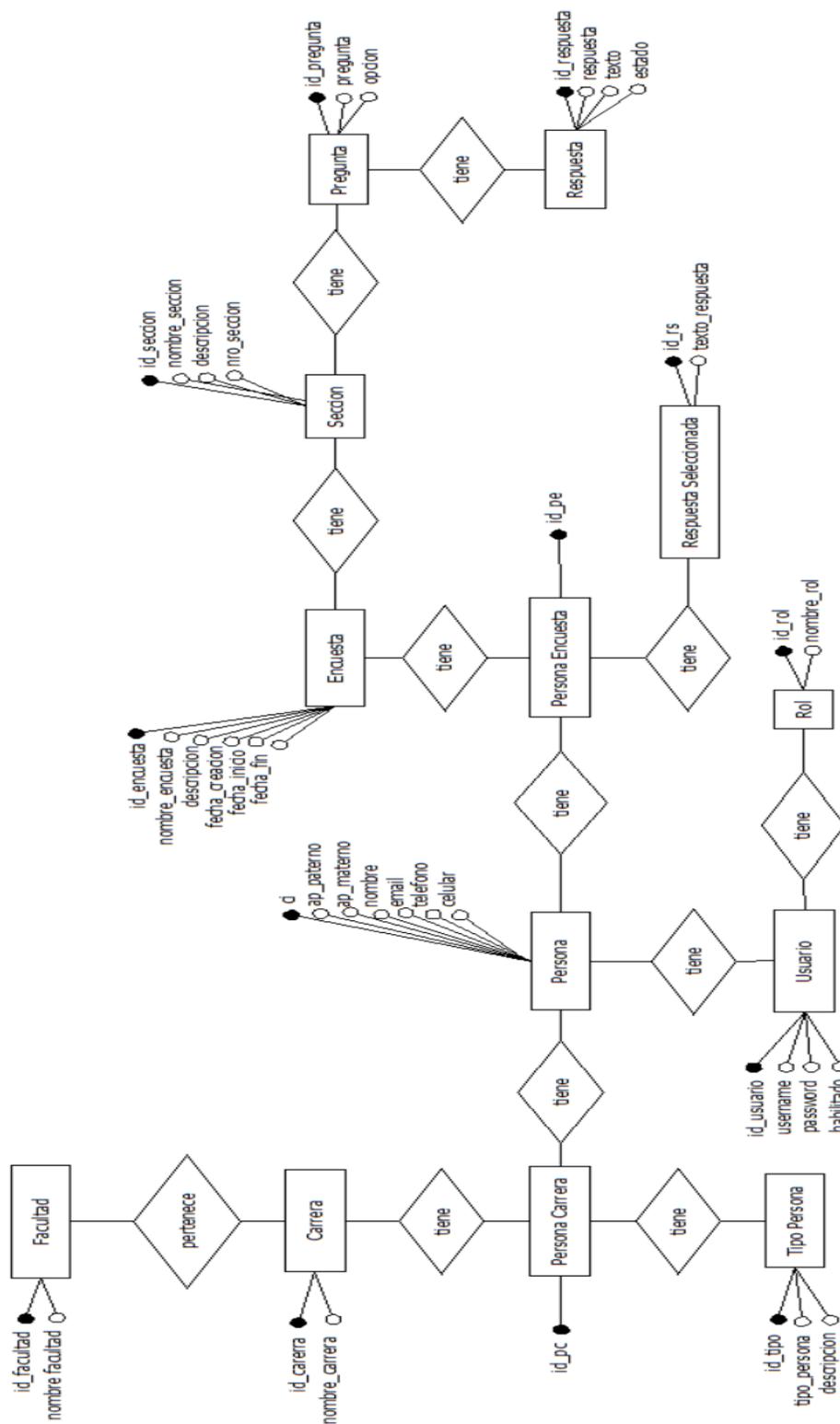


Figura 3.22 Diagrama Entidad Relación.

Fuente: Elaboración Propia.

3.3.6.2 DIAGRAMA RELACIONAL

Ahora se procederá a estructurar el modelo relacional para la implementación de la base de datos del sistema de encuesta, como se muestra en la Figura 3.23

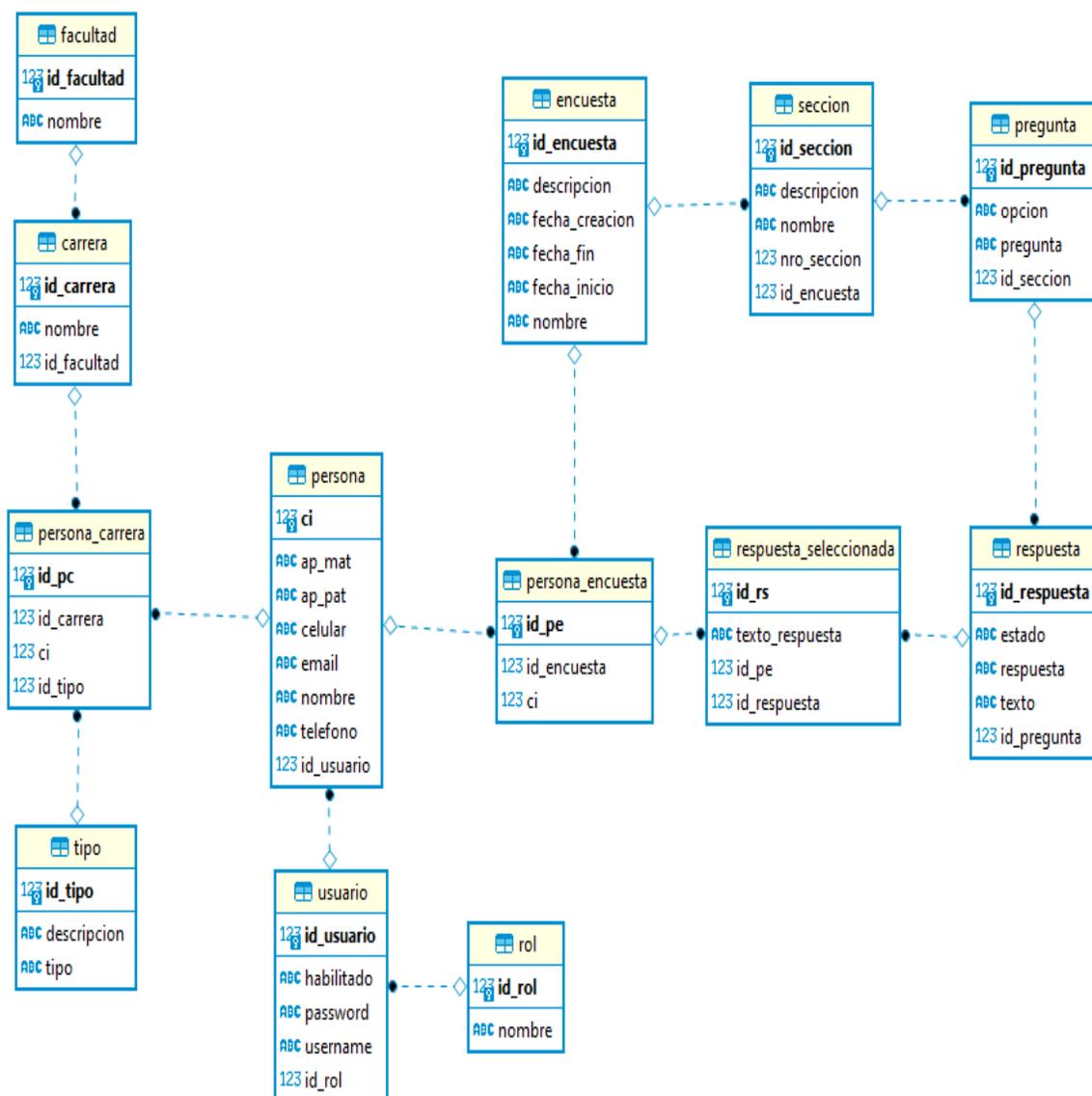


Figura 3.23 Diagrama Relacional.

Fuente: Elaboración Propia.

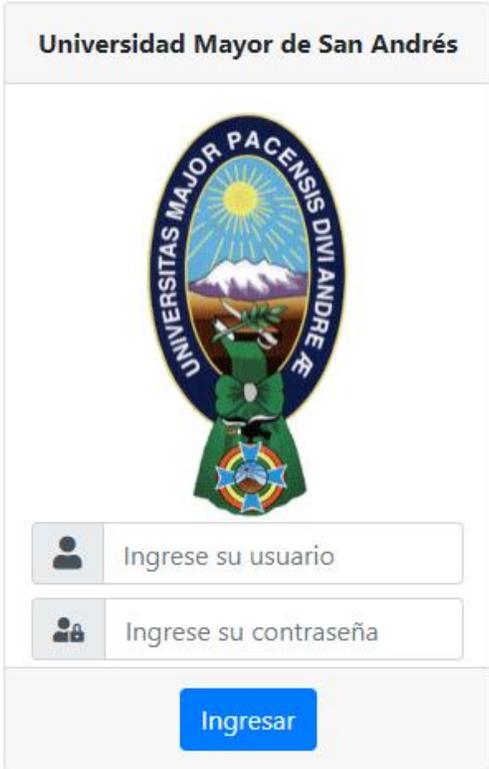
3.4 POST GAME

Iniciamos la última etapa de la metodología SCRUM en el cual se mostrará los diseños de las interfaces y las pruebas de Stress correspondientes al sistema.

3.4.1 DISEÑO DE INTERFACES

A continuación, se listará las interfaces graficas del sistema de encuesta web para el seguimiento a los titulados de la Universidad Mayor de San Andrés.

Página de Inicio. Donde el administrador, encargado de encuesta, encargado de la información del titulado (Encuestador) y estudiante titulado podrán acceder al sistema de encuesta véase la Figura 3.24



Universidad Mayor de San Andrés

UNIVERSITAS MAJOR PACENSIS DIVINDE RE

Ingrese su usuario

Ingrese su contraseña

Ingresar

Figura 3.24 Página de Inicio de Sesión.

Fuente: Elaboración Propia.

Ventana de agregación de usuario y roles. El administrador podrá agregar a usuarios con los respectivos roles para que estos puedan ingresar al sistema como se observa en la Figura 3.25.

Figura 3.25 Agregación de usuario y roles.

Fuente: Elaboración Propia.

Ventana de agregación de encuesta. Donde el encargado de la encuesta podrá agregar encuestas, como se muestra en la Figura 3.26.

Figura 3.26 Agregación de Encuesta.

Fuente: Elaboración Propia.

Ventana de monitoreo de información del estudiante. El encargado de la información del titulado (Encuestador) podrá ver la información básica del estudiante como se muestra en la siguiente Figura 3.27.



Figura 3.27 Monitoreo de información del estudiante.

Fuente: Elaboración Propia.

Ventana de la encuesta. Donde el estudiante titulado podrá responder la encuesta como observa en la siguiente Figura 3.28.

The screenshot shows a survey form titled 'Titulo Seccion: Impacto del Covid' with the subtitle 'Solo estudiantes'. The first question is:

1. ¿HAS REVISADO LA RESOLUCIÓN HCU 069/2020, QUE APRUEBA EL REGLAMENTO GENERAL DE CARÁCTER TRANSITORIO DE EDUCACIÓN A DISTANCIA Y SEMIPRESENCIAL DE LA UMSA, GESTIÓN 2020?

The response options are:

- a. SI, EN DETALLE
- b. SI, SUPERFICIALMENTE
- c. NO
- d. NO SABE/NO RESPONDE

The second question is:

2. ¿A CUÁNTAS MATERIAS TE HAS INSCRITO?

The response field is labeled 'R-' and contains the text 'Ingrese Respuesta'.

Figura 3.28 Llenado de Encuesta.

Fuente: Elaboración Propia.

3.4.2 PRUEBAS UNITARIAS

JMeter es un proyecto de Apache que utilizaremos para realizar las pruebas unitarias para el sistema de encuesta web para el seguimiento a los titulados de la Universidad Mayor de San Andrés, a continuación, se mostrara el resumen de las pruebas como se muestra en la Figura 3.29.

El informe contiene varias columnas que representan cada uno de los *Request* que se realizaron y las cuales representaran ciertas medias dentro la prueba:

Muestras: Es el número de iteraciones que se realizó por cada usuario.

Media: Es el promedio o media aritmética del tiempo en milisegundos.

Desviación estándar: Es una medida de dispensación.

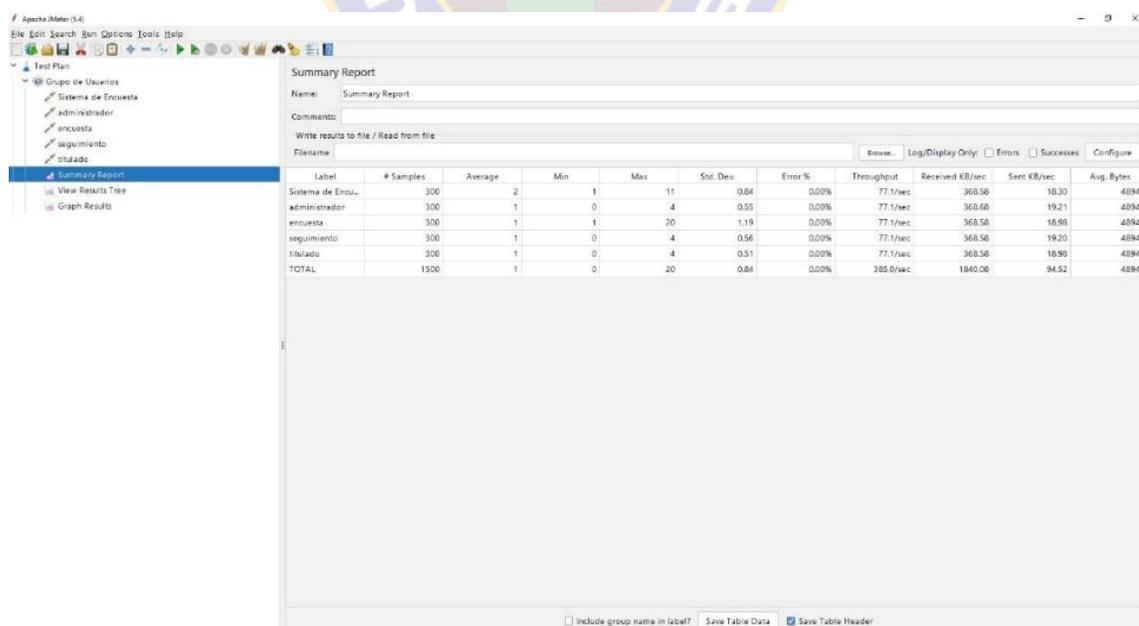
Min: Es el tiempo mínimo de todos los *Request* de ese tipo.

Max: Es el tiempo máximo de los *Request* de ese tipo,

Error: En el cual se muestra el porcentaje de *Request* fallidos,

Rendimiento: Esta medido en *Request*/segundo,

KB/sec.: Medida de velocidad de kilobytes/seg.



The screenshot shows the Apache JMeter Summary Report window. The report is titled 'Summary Report' and includes a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Sistema de Encu...	300	2	1	11	0.84	0.00%	77.1/sec	368.56	18.30	4894.0
administrador	300	1	0	4	0.55	0.00%	77.1/sec	368.68	19.21	4894.0
encuesta	300	1	1	20	1.19	0.00%	77.1/sec	368.58	18.98	4894.0
seguimiento	300	1	0	4	0.56	0.00%	77.1/sec	368.58	19.20	4894.0
titulado	300	1	0	4	0.51	0.00%	77.1/sec	368.58	18.98	4894.0
TOTAL	1500	1	0	20	0.84	0.00%	285.0/sec	1840.08	94.52	4894.0

Figura 3.29 Software Jmeter.

Fuente: Elaboración Propia.

Los datos obtenidos indican que el sistema no sufre problemas de recepción y envío de datos, por lo cual se concluye que el sistema aparentemente no presenta problemas de funcionamiento

3.4.3 PRUEBAS DE ESTRESS

Las pruebas de estrés realizadas al sistema, fueron efectuadas, simulando una concurrencia de 300 usuarios, las peticiones se realizaron con un intervalo de 2 segundos dando un poco de realismo a la prueba de estrés como se muestra en la Figura 3.29, el comportamiento obtenido del sistema en una concurrencia de 300 usuarios puede observarse en la Figura 3.30

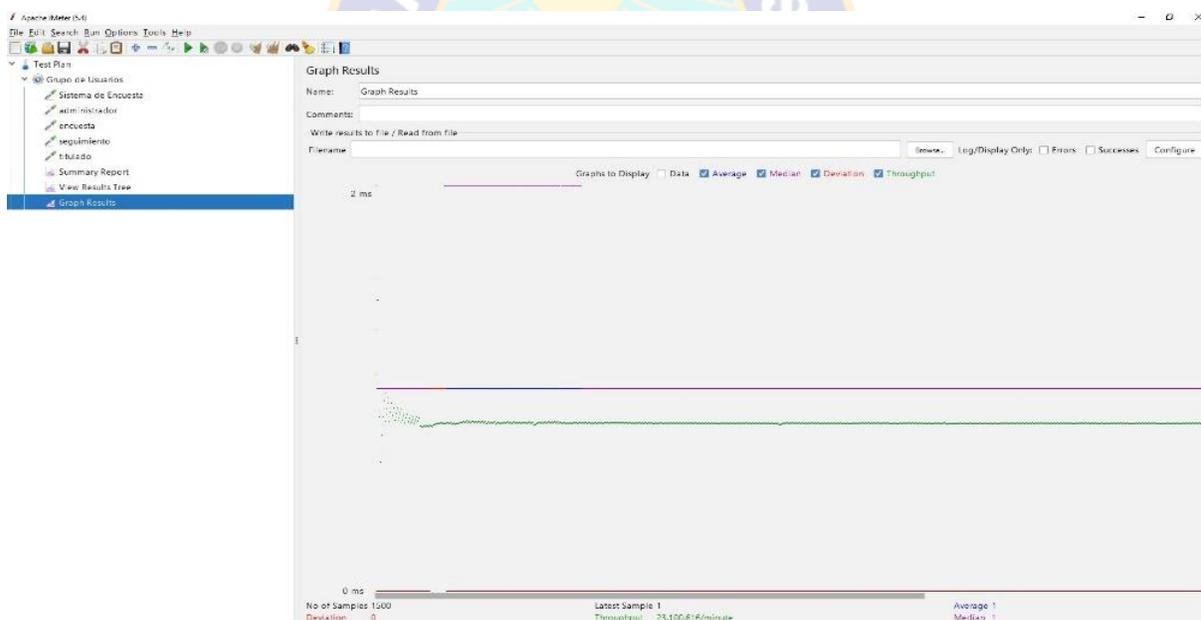


Figura 3.30 Grafica de prueba de estress del sistema.

Fuente: Elaboración Propia.

De la información obtenida se puede observar que una concurrencia de 300 peticiones al mismo tiempo el sistema tiene la capacidad de mantener una relativa estabilidad en una concurrencia de peticiones alta.

CAPÍTULO 4

CALIDAD Y SEGURIDAD

4.1 METODOLOGIA WEBQEM

El principal objetivo de la metodología para la evaluación de la calidad web (Web Quality Evaluation Method, o Web-siteQEM) consiste en evaluar y determinar el nivel de cumplimiento de las características especificadas de acuerdo a los distintos puntos de vista y objetivos de evaluación para lo cual se analizan las preferencias elementales, parciales y globales. El resultado del proceso de evaluación puede ser interpretado como el grado de requerimientos de calidad satisfechos.

4.1.1 MODELO DE AGREGACIÓN Y PUNTAJE

Respecto de la idea que sustenta a los modelos de puntajes aditivos y lineales es bastante simple. Para cada característica o sistema a evaluar y comparar se identifican n atributos necesarios, cuya preferencia o indicador elemental (IE) se debe computar.

Supongamos que los valores individuales de $IE_i \dots IEn$ están normalizados de manera que:

$$0 \leq IE_i \leq 1$$

O, en la escala del porcentaje

$$0\% \leq IE_i \leq 100\%$$

En el caso que todos los atributos intervinientes sean equi-pesados, podríamos expresar el indicador o preferencia global (IG) mediante el uso de una sumatoria. Pero si los elementos no tienen la misma importancia en el modelo de estructuración, debemos introducir la idea de peso positivo y normalizados, de manera que:

$$0 \leq P_i \leq 1 \text{ para } i = 1$$

O, expresado de otra manera

$$P_1 + P_2 + \dots + P_n = 1$$

Por lo tanto, el puntaje o indicador global puede ser expresado mediante la siguiente expresión.

$$Puntaje\ Global = \sum(\text{componente peso} * \text{indicador elemental})$$

Se debe considerar que si la cantidad de atributos es 100 el peso más bajo es casi irrelevante, en tanto que el componente peso en promedio es de 0.01 o 1% para cada indicador elemental. Esto le quita sensibilidad, flexibilidad y robustez al modelo, principalmente para evaluar sistemas de mediana o alta complejidad como se muestra en la Tabla 4.1.

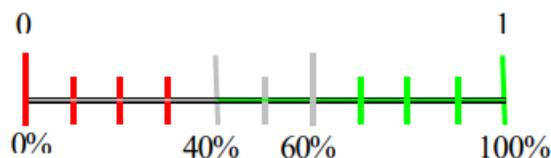


Figura 4.1 Escala de Preferencia

Fuente: Olsina, 1999

4.1.2 CARACTERÍSTICAS DE WEBQEM

WEBQEM toma las métricas del modelo de calidad ISO 9126 la cual da referencia a las siguientes características:

4.1.2.1 FUNCIONALIDAD

La funcionalidad del sistema se mide a través de su complejidad, la funcionalidad no puede ser medida directamente, entonces corresponde derivar, mediante otras medidas directas como Punto Función.

Las métricas de punto función es un método utilizado en la ingeniería del software que nos permitirá medir el tamaño del sistema, pretende medir la funcionalidad entregada al usuario independiente de la tecnología utilizada para la construcción y explotación del sistema, desde el diseño inicial hasta la implementación y el mantenimiento.

Se determinan cinco características de dominios de información estos valores se definen de la siguiente forma:

Número de entradas de usuario: Se cuenta cada entrada de usuario que proporciona diferentes datos orientados al sistema. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

Número de salidas de usuario: Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

Número de peticiones del usuario: Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

Numero de archivos lógicos internos: Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos).

Numero de archivos de interfaz externos: Se cuentan todas las interfaces legibles por la máquina, por ejemplo: archivos de datos de cinta o disco que se utilizan para transmitir información a otro sistema.

Para calcular la funcionalidad del Sistema de Encuesta utilizaremos las siguientes ecuaciones.

Funcionalidad

$$Funcionalidad = \frac{PF}{PFmax} * 100 \text{ (Ecuación 1)}$$

Punto Función

$$PF = EPF * \left(0.65 + 0.01 * \sum_i^n F_i \right) \text{ (Ecuación 2)}$$

Donde:

EPF: Es la suma de todas las entradas de PF que se obtiene de la Tabla 4.1

F_i : Resultados de los parámetros de la función.

Punto Función Máximo

$$PFmax = EPF * (0.65 + 0.01 * 70) \text{ (Ecuación 3)}$$

Donde:

EPF: Es la suma de todas las entradas de PF que se obtendrán de la Tabla 4.1

En la Tabla 4.1 para la medición de la funcionalidad, se toman en cuenta las siguientes características:

Tabla 4.1
Calculo de Entradas para el punto función

Parámetros de medición	Cuenta	Factor de ponderación			Resultados
		Simple	Medio	Complejo	
Número de entradas de usuario	15	3	4	6	60
Número de salidas de usuario	12	4	5	7	70
Número de peticiones	10	3	4	6	40
Número de Archivos	5	7	10	15	50
Número de archivos de interfaz externos	0	5	7	10	0
Cuenta Total					220

Fuente: Elaboración Propia

A continuación, se muestra en la Tabla 4.2 la escala de niveles de influencia para hallar:

Tabla 4.2
Escala de niveles de influencia

Escala	0	1	2	3	4	5
Complejidad	Sin Influencia	Poca Influencia	Moderado	Medio	Significativo	Esencial

Fuente: Elaboración Propia

Los factores de ajuste son representados en 14 puntos que se muestran en la siguiente Tabla 4.3 y se debe asignar un puntaje del 0 al 5 donde:

Tabla 4.3
Cálculo de punto función Factores de complejidad

Nro	Factor Ambiental	Puntaje
1	¿Se requiere comunicación de datos?	5
2	¿Existen funciones o procedimientos distribuidos?	4

3	¿Es crítico el rendimiento?	4
4	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado? ¿Hay restricciones de plataforma?	3
5	¿El sistema tendrá una carga transaccional alta o baja?	3
6	Nivel de Disponibilidad	4
7	Eficiencia del usuario final requerida (usabilidad)	5
8	Actualización en línea	3
9	Complejidad del Procesamiento	3
10	El sistema debe estar diseñado e implementado para ser reutilizable	5
11	¿El sistema debe ser diseñado para ser fácil de instalar y de portar?	5
12	Facilidad de Uso	5
13	¿El sistema debe soportar múltiples instalaciones en diferentes organizaciones?	3
14	¿El sistema debe estar diseñado e implementado para facilitar cambios?	5
Factor de Ajuste		57

Fuente: Elaboración Propia

Aplicando los datos obtenidos costo total y el factor de ajuste en ecuación 2 del punto función se tiene:

$$PF = 220 * (0.65 + 0.01 * 57)$$

$$PF = 268.4$$

Por tanto, el Punto Función máximo es de: 268.4

Aplicando los datos obtenidos costo total y el factor de ajuste en la ecuación 3 del punto función máximo se tiene:

$$PF_{max} = 220 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 297$$

Por tanto, el Punto Función máximo es de 297

La funcionalidad del sistema será:

$$Funcionalidad = \frac{268.4}{297} * 100 = 90.37\%$$

Por tanto, la funcionalidad del sistema se representa por el 90.37%. Lo que quiere decir que el sistema cumple con los requisitos funcionales de forma satisfactoria.

4.1.2.2 USABILIDAD

Cuando se habla de usabilidad, se espera que el software sea de fácil entendimiento y aprendizaje, la usabilidad no es afectada por la funcionalidad y eficiencia. La usabilidad está definida por los usuarios finales o asociados del Software.

Para la medición de la usabilidad se tiene el siguiente cuestionario con los ajustes necesarios que se pueden observar en la siguiente Tabla 4.4:

Tabla 4.4
Cálculo de usabilidad del sistema de encuesta

Nro.	Preguntas Respuesta Resultado	Puntaje	Porcentaje
1	¿El diseño estético del sistema es de su agrado?	4	80%
2	¿Las salidas del sistema son las esperadas?	5	100%
3	¿Le parece fácil el aprendizaje de manejo del sistema?	4	100%
4	¿Se ha cubierto todos los requerimientos establecidos por el software?	5	100%
5	¿El software tiene la seguridad necesaria?	5	100%
6	¿Está de acuerdo con el funcionamiento del software?	4	80%

7	¿El software facilita el trabajo que realiza?	5	100%
TOTAL		32	91.4%

Fuente: Elaboración Propia

Por tanto, la Usabilidad de sistema se representa por el 91.4%.

4.1.2.3 MANTENIBILIDAD

La mantenibilidad está asociada a la detección y corrección de fallas a los cambios debidos a los requerimientos del usuario, a las adaptaciones requeridas a medida que evoluciona el sistema.

Para calcular la mantenibilidad del sistema de encuestas utilizaremos la siguiente ecuación:

$$IMS = \frac{[Mt - (Fc + Fa + Fd)]}{Mt} \quad (\text{Ecuación 4})$$

Donde:

IMS: Índice de madurez de software.

Mt: Número de módulos en la versión actual.

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

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

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

Aplicando los datos en Ecuación 4 de mantenibilidad se tiene:

$$IMS = \frac{[5 - (1 + 0 + 0)]}{5} = 0.8$$

Por tanto, el sistema posee una estabilidad de un 80% lo que nos indica la facilidad del mantenimiento con la que se puede corregir el sistema si es que se puede encontrar un error, o si el cliente solicita un cambio en el sistema.

4.1.2.4 PORTABILIDAD

El hecho de que el sistema este desarrollado en el Lenguaje Java y es en el entorno web nos ayudara de sobremanera en cuanto a su portabilidad, dado que únicamente es necesario tener un navegador instalado, por lo tanto, la portabilidad es de un 100%.

4.1.3 CALIDAD TOTAL

Para poder obtener la calidad total del Software, sacaremos la media de todas las medidas expresadas en porcentaje como se muestra en la siguiente Tabla 4.5.

Tabla 4.5
Resultados de las métricas de calidad del sistema de encuesta

Métrica	Porcentaje	Puntaje
Funcionalidad		90.37%
Usabilidad		91.4%
Mantenibilidad		80%
Portabilidad		100%
<i>Total</i>		<i>90.4%</i>

Fuente: Elaboración Propia

Observando la Tabla 4.5 podemos decir que el sistema de encuesta web para el seguimiento a los titulados de la Universidad Mayor de San Andrés tiene una preferencia de calidad global satisfactoria del 90.4% en todas sus características de más alto nivel. Por tanto, se puede decir que el sistema cumple con los requisitos de calidad.

4.2 SEGURIDAD

4.2.1 APLICACIÓN DE OWASP

A continuación, detallaremos cada uno de OWASP Top 10 en el sistema de encuesta, cuyo objetivo de este es crear y concientizar sobre la seguridad en aplicaciones web mediante la identificación de algunos de los riesgos más críticos que enfrentan las organizaciones.

A1: El sistema no admite la inyección de código por medio de consultas SQL, ya que el proyecto hace uso de JPA que es un Framework de Mapeo Relacional de Objetos (MRO) por lo que los usuarios no podrán hacer inyección de código.

A2: El sistema esta implementado con Spring Security que es un marco de autenticación y control de acceso, cierra las sesiones inactivas de los usuarios, por lo que es difícil que sufra este riesgo de seguridad.

A3: El sistema cuenta con encriptado para los datos sensibles.

A4: El sistema no hace referencias a entidades externas en documentos XML por lo cual es difícil que sufra este percance.

A5: En este punto con Spring Security no habría problema con la autenticación de usuarios, ya que con este *Framework* a cada tipo de usuario se le asigna un rol o acceso, aparte de que tiene actualizaciones de seguridad.

A6: Como se trabaja con Spring todos sus *Framework* derivados tienen una actualización porque es difícil que suceda este riesgo,

A7: Para acceder al sistema se tiene que estar registrado en la base de datos y estos datos están encriptados por lo cual el sistema no admite la intromisión de personas ajenas y es muy difícil que sufra un ataque de tipo XSS.

A8: El sistema maneja el *Framework* JPA lo cual permite interactuar con la base de datos por medio de objetos y el cual es muy difícil de que haya una deserialización insegura.

A9: Todos los *Frameworks* cuentan con la actualización de seguridad después de encontrarse alguna vulnerabilidad en esta.

A10: El sistema cuenta con un control y validación de entrada del lado del servidor para identificar las cuentas sospechosas o maliciosas y así denegándolas. Para luego poder hacer un análisis.

CAPÍTULO 5

ANÁLISIS COSTO BENEFICIO

5.1 INTRODUCCIÓN

El análisis costo beneficio es una metodología para evaluar de forma exhaustiva los costes y beneficios de un proyecto, con el objetivo de determinar si el proyecto es deseable desde el punto de vista del bienestar social y, si lo es, en qué medida. Para ello, los costes y beneficios deben ser cuantificados, y expresados en unidades monetarias, con el fin de poder calcular los beneficios netos del proyecto para la sociedad en su conjunto.

En este punto cuantificaremos la inversión de los recursos que se empleó en el desarrollo del Software. Para lo cual utilizaremos el modelo COCOMO II.

5.2 COCOMO II

COCOMO II permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de coste y de escala.

En los factores de coste se incluye aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto.

Los factores de escala incluyen la parte de escala producida a medida que un proyecto se software incrementa su tamaño.

5.2.1 COSTO DEL SISTEMA DESARROLLADO

A continuación, realizaremos la conversión de punto función a miles de líneas de código, mediante la siguiente Tabla 5.1:

Tabla 5.1
Factor LCD/PF de lenguajes de programación

Lenguaje	Factor LDC/PF
Java	53
Visual Basic	46
ASP	36

C++	34
Visual C#	34
PHP	12
Ensamblador	320
C	150

Fuente: Elaboración Propia

En el PF se muestran diferentes tipos de lenguaje a utilizar será Java con un valor de 53 LCD por PF.

$$KLCD = \frac{(EPF * Factor LDC / PF)}{1000} \quad (\text{Ecuación 5})$$

Donde:

KLCD: Número estimado de líneas de código distribuidas.

EPF: Entradas punto Función.

Reemplazando Valores en la Ecuación 5

$$KLCD = \frac{(220 * 53)}{1000} = 11.66$$

Por tanto, el sistema posee 11.66 KLCD líneas de código.

5.2.1.1 ESTIMACIÓN DE ESFUERZO

El esfuerzo se refiere a la suma de los tiempos que le dedicaran los diferentes recursos a cierta actividad o al proyecto. Se mide en horas/hombre, días/hombre, semanas/hombre, etc. No importa que el trabajo se haga de forma secuencial por un solo recurso o en paralelo por diferentes personas. Se suman los tiempos de cada uno de ellos para obtener el esfuerzo total.

Todo proyecto corresponde a uno de los siguientes tres modos:

Modo Orgánico (*Organic*): En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar es relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware.

Modo Semiacoplado (*Semidetached*): Es un modelo para productos de software de tamaño y complejidad media. Las características de los proyectos se consideran intermedias a las de los modos Orgánico y Empotrado.

Modo Empotrado (*Embedded*): En esta clasificación están incluidos proyectos de gran envergadura que operan en un ambiente complejo con altas restricciones de hardware, software y procedimientos operacionales, tales como los sistemas de tráfico aéreo.

En la Tabla 5.2 se muestran los coeficientes para los diferentes modos.

Tabla 5.2
Modelo básico para tipos de proyectos

Proyecto de software	a	b	c	d
Orgánicos	2.4	1.05	2.5	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Fuente: Elaboración Propia

Para calcular el esfuerzo aplicado por persona utilizaremos la siguiente ecuación:

$$E = a(KLCD)^b \text{ (Ecuación 6)}$$

Donde:

E: Esfuerzo aplicado por la persona.

KLCD: Número estimado de líneas de código distribuidas.

Reemplazando en la ecuación 6

$$E = 3 (11.66)^{1.12} = 46.97$$

Para calcular tiempo de desarrollo en meses utilizaremos la siguiente ecuación:

$$D = c(E)^d \text{ (Ecuación 7)}$$

Donde:

D: tiempo de desarrollo en meses.

Reemplazando en la ecuación 7.

$$D = 2.5(46.97)^{0.35} = 9.62$$

Para calcular tiempo de desarrollo en meses utilizaremos la siguiente ecuación.

$$NP = \frac{E}{D} \text{ (Ecuación 8)}$$

Donde:

NP: Numero de programadores.

D: Tiempo de desarrollo en meses

E: Esfuerzo aplicado por persona

Reemplazando en la ecuación 8.

$$NP = \frac{46.97}{9.62} = 4.88$$

Redondeando ya que se refiere a personas:

$$NP = 5$$

Realizando un estudio de mercado laboral, se toma en cuenta que un ingeniero informático percibe un salario mínimo de 300\$us (Dólares americanos) aproximadamente. Así podemos calcular el costo del sistema.

$$CS = \text{salarioMinimo} * D * NP \text{ (Ecuación 9)}$$

CS: Costo de Software

Reemplazando en la Ecuación 9 se tiene:

$$CS = (300) * 9.62 * 5$$

$$CS = 14430$$

Con los datos obtenidos se tiene el siguiente resultado descrito en la Tabla 5.3

Tabla 5.3
Tabla de resultados COCOMO II

Descripción	Total
Tiempo de desarrollo (Meses)	9.62
Numero de Programadores	5
Costo del Proyecto (expresado en \$us)	14430

Fuente: Elaboración Propia

5.2.2 COSTO ELABORACIÓN DEL SISTEMA

Los costos de elaboración del proyecto se detallan en la Tabla 5.4

Tabla 5.4
Costo de elaboración del sistema

Detalle	Importe(\$us)
Análisis y diseño del proyecto	200
Material de escritorio	20
Internet	120
Pasajes	20

Otros	15
Total	375

Fuente: Elaboración Propia

5.2.3 COSTOS DE IMPLEMENTACIÓN DEL SISTEMA

Las herramientas de software son JAVA, PostgreSQL, Spring, todas las herramientas son de uso libre por esa razón el costo es 0.

5.2.4 COSTO TOTAL DEL SISTEMA

El coste total del sistema en Dólares Americanos (\$us) es la suma de los costos, los cuales se presentan en la Tabla 5.5

Tabla 5.5
Costo total del sistema

Detalle	Importe (\$us)
Costo del Sistema Desarrollado	14430
Costo elaboración del Sistema	375
Costo de implementación del Sistema	0
Total	14805

Fuente: Elaboración Propia

5.3 CÁLCULO BENEFICIO VAN Y TIR

El VAN y el TIR son dos fórmulas financieras empleadas con frecuencia para analizar qué tan oportuno puede ser un proyecto para una empresa, independiente del área en el que opere o del tipo de producto al que se menciona.

Las siglas VAN corresponde al valor neto, mientras el TIR es la tasa interna de retorno. Ambas fórmulas se relacionan de forma directa con el flujo de caja de los negocios y buscan hacer más preciso el cálculo del tiempo en que un negocio tardara en recuperar su inversión inicial.

5.3.1 VALOR ACTUAL NETO VAN

El valor actual neto VAN es un criterio de inversión que consiste en actualizar los cobros y pagos de un proyecto o inversión para conocer cuánto se va a ganar o perder con esa inversión.

Para ello trae todos los flujos de caja al momento presente descontándolos a un tipo de interés determinado. El VAN va a expresar una medida de rentabilidad del proyecto en términos absolutos netos, es decir, en número de unidades monetarias (euros, dólares, etc.).

$VAN > 0$ se recomienda pasar a la siguiente etapa del proyecto.

$VAN = 0$ es indiferente realizar la inversión.

$VAN < 0$ se recomienda desecharlo o postergarlo.

Para calcular el valor actual neto VAN utilizaremos la siguiente ecuación:

$$VAN = -A + \sum_i^n \frac{Q_i}{(1 + K)^i} \quad (\text{Ecuación 10})$$

Donde:

A: Inversión Inicial

Q: Flujo de caja

K: Tasa de Retorno Seleccionada

n: Vida útil del proyecto

i: Periodo

En la Tabla 5.6 se muestra la caja de flujo en años con una inversión de 14805.

Tabla 5.6
Flujo de Caja proyectado en 5 años

Año	2019	2020	2021	2022	2023	2024
-----	------	------	------	------	------	------

Proyección	-14805	13028.4	14805	16581.6	18358.2	20134.8
-------------------	--------	---------	-------	---------	---------	---------

Fuente: Elaboración Propia

Aplicando los datos obtenidos del flujo de caja 5.6 con una tasa de retorno del 12% en la Ecuación 10 del VAN se tiene:

$$VAN = -(-14805) + \frac{13028.4}{(1 + 0.12)^1} + \frac{14805}{(1 + 0.12)^2} + \frac{16581.6}{(1 + 0.12)^3} + \frac{18358.2}{(1 + 0.12)^4} + \frac{20134.8}{(1 + 0.12)^5}$$

$$VAN = 73134.40$$

Como se puede observar el VAN es 73134.40 que es mayor a cero por lo tanto el proyecto es aceptable.

5.3.2 TASA INTERNA DE RETORNO TIR

Esta tasa indica la rentabilidad promedio anual que genera el capital que pertenece invertido en el proyecto.

TIR > k se recomienda pasar a la siguiente etapa.

TIR > k es indiferente invertir

TIR < k se recomienda se rechazó o postergación.

Para calcular la Tasa interna de Retorno TIR utilizaremos la siguiente ecuación

$$VAN = \sum_i^n \frac{Q_i}{(1 + TIR)^i} \quad (\text{Ecuación 11})$$

Donde:

Q: Flujo de caja.

TIR: Tasa interna de Retorno.

n: Vida útil del proyecto.

i: Periodo.

VAN: Valor actual neto.

Aplicando los datos obtenidos del flujo de caja en Tabla 5.6 con el VAN = 0 en la Ecuación 11 del TIR se tiene:

$$0 = -14805 + \frac{13028.4}{(1 + 0.12)^1} + \frac{14805}{(1 + 0.12)^2} + \frac{16581.6}{(1 + 0.12)^3} + \frac{18358.2}{(1 + 0.12)^4} + \frac{20134.8}{(1 + 0.12)^5}$$

$$= 0.95$$

Como se puede observar el TIR es 0.95 que es mayor a la tasa seleccionada de Retorno por lo tanto el proyecto es aceptable.

5.4 COSTO BENEFICIO

El costo beneficio tiene como objetivo fundamental proporcionar una medida de los costos en que se realiza un proyecto y comparar dichos costos previstos con los beneficios esperados.

Para calcular el beneficio neto utilizaremos la siguiente Ecuación.

$$\sum Beneficion\ Neto = \sum_i^n \frac{Ganancias}{(1 + k)^i} - \sum_i^n \frac{Costos}{(1 + k)^i} \text{ (Ecuación 12)}$$

Donde:

Ganancias: Ingreso de flujo anual.

Costos: Salidas de flujo anual.

n: Numero de costo.

K: tasa de descuento anual.

La proyección se realizará a 5 años con una tasa de retorno de 12% de descuento anual como se muestra en la Tabla 5.7.

Tabla 5.7
Proyección del Beneficio Neto

Año	Costo	Beneficio	Beneficio Neto
1	14805	0	0
2	3500	7500	3188.78
3	3000	10000	4982.46
4	2500	12500	6355.18
5	2000	15000	7376.55
Sumatoria	25805	45000	8684.22

Fuente: Elaboración Propia

Una vez que se calculó el beneficio neto y se obtuvo los resultados encontrados en la Tabla 5.7, se puede pasar a reemplazar los datos en la siguiente ecuación de costo y beneficio.

$$\frac{\text{Costo}}{\text{Beneficio}} = \frac{\sum \text{Beneficio}}{\sum \text{Costo}} \text{ (Ecuación 12)}$$

Donde:

$\sum \text{Beneficio}$: Senatoria de los ingresos de flujo anual.

$\sum \text{Costo}$: Sumatoria de las salidas de flujo anual.

$$\frac{\text{Costo}}{\text{Beneficio}} = \frac{45000}{25805} = 1.74 \$us$$

El resultado del costo beneficio es igual 1.74 \$us indica que por cada 1\$us se obtendrá una ganancia de 0.74 ctvs. de \$us, por tanto, el proyecto es viable.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Considerando los requerimientos del sistema de encuesta web para el seguimiento a los titulados de la Universidad Mayor de San Andrés de parte de la División de Sistemas de Información y Estadística (DSIE), los cuales fueron satisfechos, se ha cumplido con el objetivo planteado por medio de la implementación del sistema a través de los cinco módulos, los cuales son: Gestión de Usuarios, Administración de Encuesta, Muestreo de la información del estudiante y estado de llenado de la encuesta del estudiante, Llenado de la Encuesta y Realización de Reportes.

Por lo tanto, se concluye:

- Se implementó el módulo de Gestión de Usuarios donde el Administrador podrá agregar a los usuarios y asignar roles, a las personas que lo soliciten usar el sistema de encuesta.
- Se implementó el módulo de Administración de Encuesta donde el encargado de encuesta podrá crear la encuesta con sus debidas preguntas para que el usuario titulado las pueda responder, también podrá generar reportes de las encuestas.
- Se logró implementar el modulo del muestreo de la información del estudiante y el estado de llenado de la encuesta, los cuales permiten ver información básica de los Estudiante Titulados que están llenado la encuesta para así poder dar mejor seguimiento a la encuesta.
- Se implementó el módulo de Llenado de encuesta donde el Estudiante Titulado podrá realizar su encuesta y esta podrá ser guardada en la base de datos.
- Se implementó aspectos de seguridad para poder cuidar la integridad del sistema ante la pérdida de datos o mala manipulación, se implementó uso de sesiones para que los usuarios tengan que autenticarse para poder usar el sistema.

6.2 RECOMENDACIONES

A la conclusión del presente proyecto de grado “Sistema de Encuesta Web para el Seguimiento a los Titulados de la Universidad Mayor de San Andrés” se tiene las siguientes recomendaciones:

- Se recomienda de manera enfática programar el mantenimiento al Sistema de Encuesta para mantener el correcto desempeño del sistema y poder prevenir posibles fallas.
- Se recomienda a la División de Sistemas de Información y Estadística (DSIE) tener precaución con sus servidores ya que ahí se almacena toda la información del sistema de encuesta.
- Realizar ampliaciones del Sistema de encuesta que no sea únicamente para los Estudiantes Titulados sino también para los estudiantes que siguen en carrera, para docentes, estudiantes de Posgrado, y administrativos de la Universidad Mayor de San Andrés.

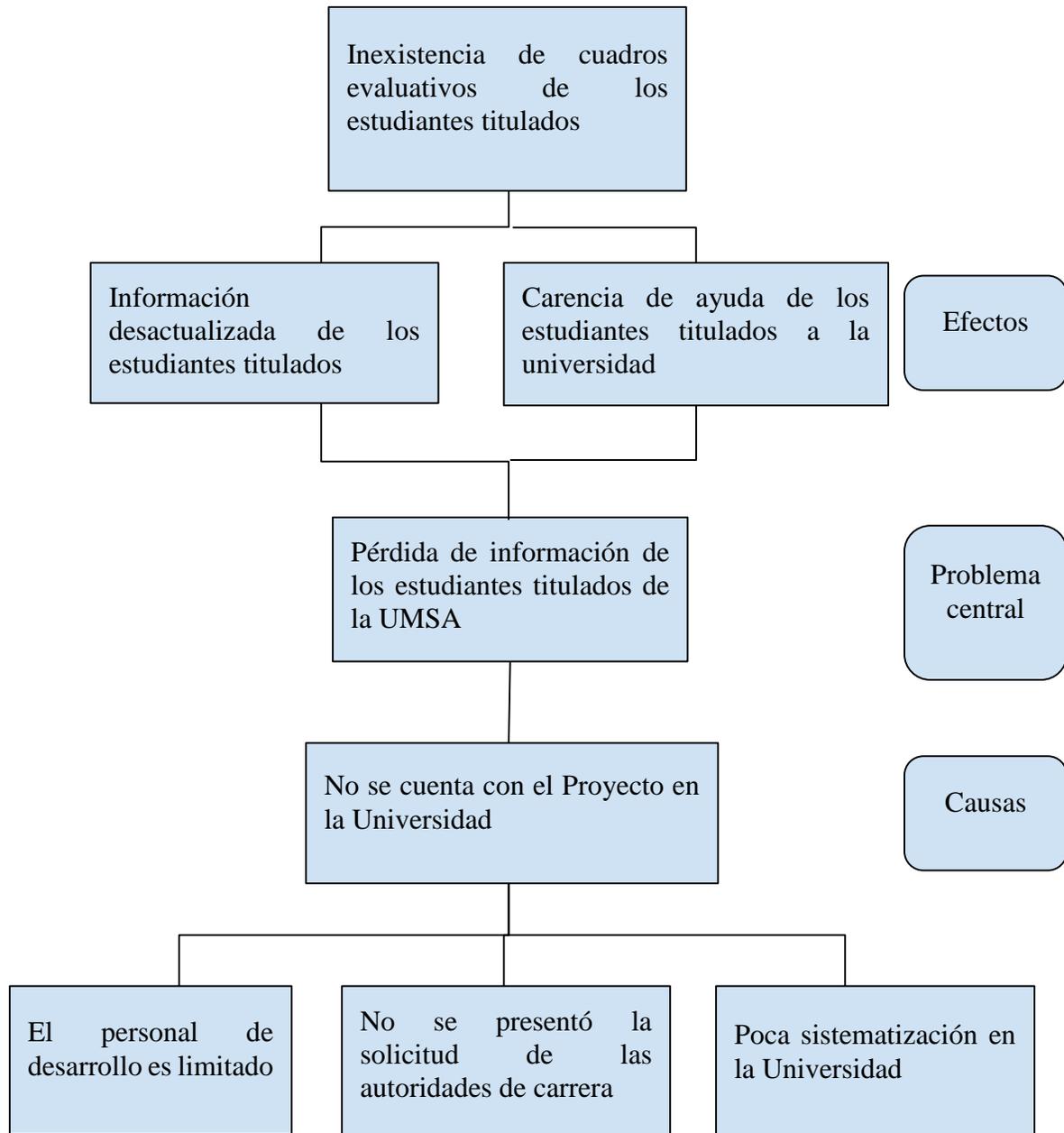


BIBLIOGRAFÍA

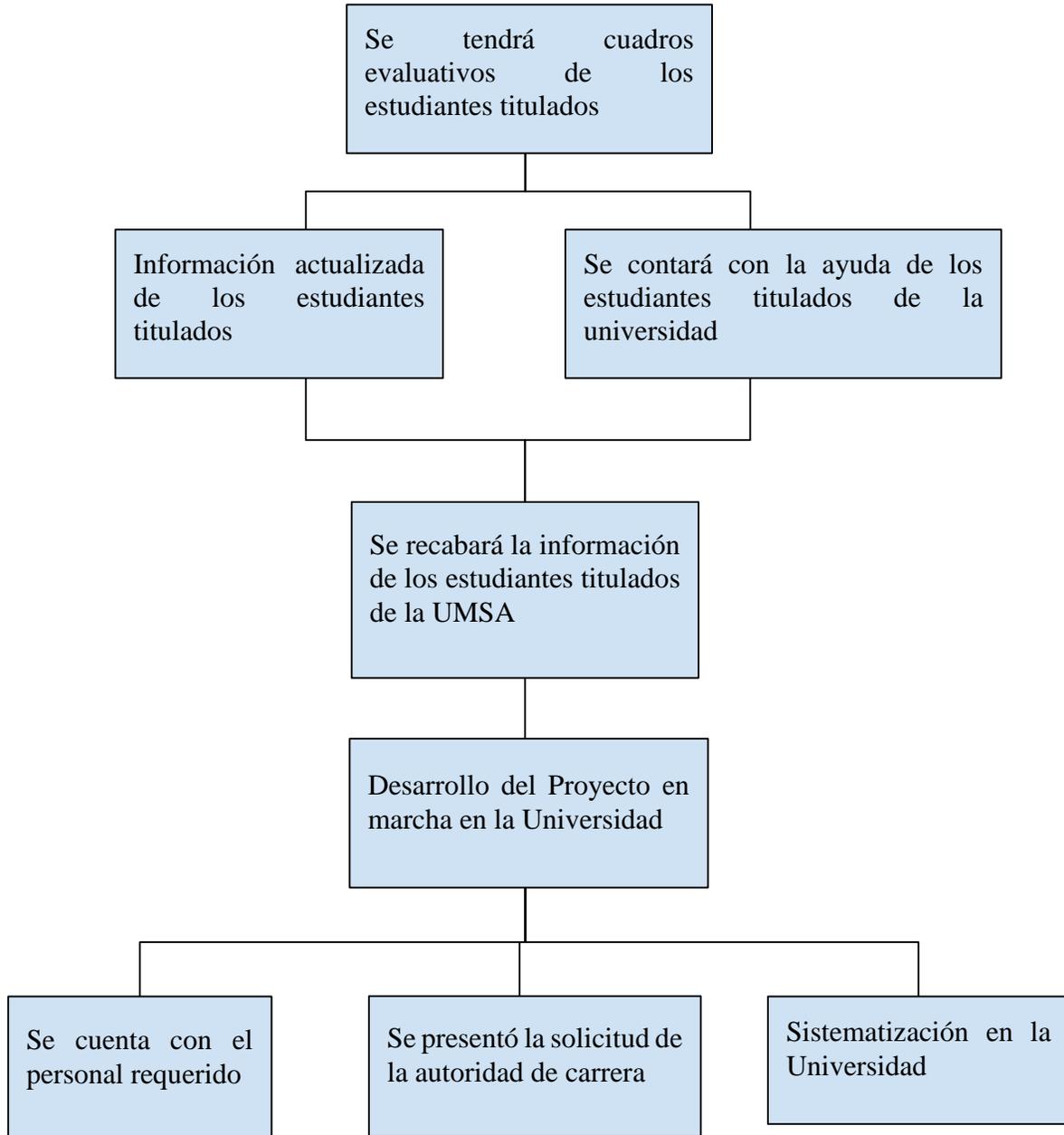
- Alaimo, D. M. (2015). *Proyectos ágiles con Scrum : flexibilidad, aprendizaje, innovación y colaboración en contextos complejos*. Buenos Aires: kloor.
- Albaladejo, X. (31 de marzo de 2019). *proyectos agiles*. Obtenido de <https://proyectosagiles.org/que-es-scrum/>
- Alvarez, M. A. (8 de 10 de 2020). *DesarrolloWeb.com*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Aurazo, E. (2017). *Estudio de UWE - Metodología de Desarrollo Web*.
- Blanco, S. D. (8 de 10 de 2020). *Metodología UWE*. Obtenido de <https://metodologiauwe.wordpress.com/2015/06/25/hello-world/>
- Fernández Romero, Y., & Díaz Gonzáles, Y. (2012). *Patrón Modelo-Vista-Controlador. TELEM@TICA*, 47-57.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Buenos aires.
- Manuel, T. G. (2012). *Metodología Scrum*.
- Maximilians, L. (2016). *UWE - UML-based Web Engineering*. Obtenido de <https://uwe.pst.ifi.lmu.de/index.html>
- Moreno, M., Toledo, A., López, C., & Cruz, N. (8 de 10 de 2020). *ISO 9126*. Obtenido de https://iso9126uts.blogspot.com/p/i_8.html
- Olsina, L. A. (1999). *Metodología Cuantitativa para la Evaluación y Comparación de la calidad de Sitios Web*. La Plata.
- Peralta, A. (2003). *Metodología Scrum*. Montevideo: Universidad ORT Uruguay.
- Rivas, C. I., Corona, V. P., Gutiérrez, J. F., & Hernández, L. (2015). *Metodologías actuales de desarrollo de software*. *Revista Tecnología e Innovación*, 980-986.
- Sommerville, I. (2005). *Ingeniería del software*. Madrid: Pearson Educación.
- Srivastava, N. (05 de febrero de 2020). *Product Backlog Explained*. Obtenido de <https://www.c-sharpcorner.com/article/product-backlog/>
- Valle Rodríguez, A. N. (s.f.).
- Valle Rodríguez, A. N. (2009). *Metodologías de diseño usadas en ingeniería web, su vinculación con las NTICS*. Buenos Aires: UNIVERSIDAD NACIONAL DE LA PLATA.
- Velasco, R. (8 de 10 de 2020). *Top-10 de vulnerabilidades de 2017 segun el nuevo informe OWASP*. Obtenido de <https://www.redeszone.net/2017/11/21/top-10-vulnerabilidades-2017-owasp/>

ANEXOS

ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS



MATRIZ DEL MARCO LOGICO

Descripción	Indicadores	Verificadores	Supuestos
<p>FIN: Se tendrá cuadros evaluativos de los estudiantes titulados</p>	<ul style="list-style-type: none"> - Lograr la sistematización de la U.M.S.A. al 100%. - Lograr el llenado de la encuesta en por lo menos un 95% de los titulados para lograr cuadros evaluativos confiables. 	<ul style="list-style-type: none"> - Los registros de la información obtenida de los titulados mediante encuestas se encontrar en los servidores de la unidad D.S.I.E. 	<ul style="list-style-type: none"> - La universidad contará con los datos actualizados de los titulados - La universidad tendrá información del comportamiento de los titulados después su educación superior.
<p>PROPOSITO Se recabará la información de los estudiantes titulados de la UMSA</p>	<ul style="list-style-type: none"> - Cantidad de información obtenida de los titulados en el tramo de la licenciatura, el acceso al mundo laboral, su situación laboral actual y su formación postgrado 	<ul style="list-style-type: none"> - Se podrá revisar la información en una página designada por la unidad D.S.I.E. 	<ul style="list-style-type: none"> - Participación de los estudiantes en el llenado de la encuesta.

<p>COMPONENTES / RESULTADOS</p> <p>Implementación del “Sistema de Seguimiento a los titulados de la U.M.S.A.”</p>	<ul style="list-style-type: none"> - Aprobación del proyecto por la unidad D.S.I.E. de la U.M.S.A. - Aprobación de la documentación del proyecto de grado. - Calidad de desarrollo del Sistema 	<ul style="list-style-type: none"> - Verificar la hoja de Ruta de en la unidad D.S.I.E. - Verificación Mediante una resolución del HCC (Honorable consejo de Carrera) - Nota del D.S.I.E. donde se aclare que el sistema cumple con los requerimientos especificados y las necesidades o expectativas 	<ul style="list-style-type: none"> - Cumplir con el plazo establecido del desarrollo del proyecto. - Capacitar a los estudiantes y encuestadores para el manejo del sistema.
<p>ACTIVIDADES</p> <ul style="list-style-type: none"> - Se contará con el personal requerido - Se presentó la solicitud de las autoridades de carrera - Mayor Sistematización en la Universidad 	<ul style="list-style-type: none"> - La División de Sistemas de Información y Estadística (D.S.I.E.) da el servicio de Desarrollo del Sistema sin costo alguno 	<ul style="list-style-type: none"> Revisión de Notas de No pago en la División de sistemas de información y estadística 	<ul style="list-style-type: none"> - Apoyo de Personal Capacitado para el desarrollo del Sistema. -Equipamiento en software y hardware para el desarrollo del proyecto.