

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

**“SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL”**

**CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, UNIDAD DE FORMACIÓN ACADEMICA CONTINUA (UFAC)**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas**

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante: Sergio Luis Yujra Torrez**

**Tutor Metodológico: Lic. Ing. Helen Fanny Suntura Escobar**

**Tutor Revisor: Lic. Ing. Freddy Salgueiro Trujillo**

**Tutor Especialista: Ing. Walter Emilio Paco Siles**

**EL ALTO - BOLIVIA**

**2024**

## **DEDICATORIA**

Dedicado con profundo agradecimiento a aquellos que han sido parte fundamental en este importante logro:

A mi padre, quien desde el cielo me observa y protege. Gracias, padre por darme la vida y por regalarme tu amor incondicional, el cual sigue guiando mis pasos cada día.

A mi querida mamá y hermanos, por su inquebrantable apoyo, confianza y motivación. Su constante aliento es el motor que impulsa mi constante superación y crecimiento.

Sin ustedes, este logro no sería posible. Su presencia en mi vida ha sido el pilar sobre el cual he construido mis sueños y metas. ¡Gracias por ser mi fuente de inspiración y fortaleza!

## **AGRADECIMIENTOS**

A Dios, por concederme la vida y fortaleza necesaria para completar este trabajo de grado.

Al ingeniero Walter Emilio Paco Siles, mi tutor especialista, por su incansable paciencia, comprensión y constante orientación a lo largo de este proyecto. Su apoyo incondicional no solo en lo académico, sino también en lo personal, ha sido invaluable.

Al licenciado Freddy Salgueiro Trujillo, mi tutor revisor, por su colaboración, tiempo dedicado y valiosas recomendaciones que han enriquecido mi trabajo.

A la Ingeniería Helen Suntura Escobar, mi tutora metodológica, por su confianza, paciencia y orientación durante todo el proceso de desarrollo de este trabajo.

A la Dirección de Posgrado de la Universidad Pública de El Alto, por brindarme la oportunidad de realizar mis prácticas profesionales y llevar a cabo este proyecto de grado.

A mi amada carrera de Ingeniería de Sistemas y a la universidad, por la formación y conocimientos impartidos, que han sido fundamentales para el éxito de este trabajo.

## ÍNDICE

CAPÍTULO I.....	1
1    MARCO PRELIMINAR .....	2
1.1  INTRODUCCIÓN.....	2
1.2  ANTECEDENTES.....	3
1.2.1 Antecedentes Institucionales.....	3
1.2.2 Antecedentes Afines al Proyecto de Grado .....	5
1.3  PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1 Problema Principal.....	8
1.3.2 Problemas Secundarios .....	8
1.3.3 Formulación del Problema.....	9
1.4  OBJETIVOS.....	10
1.4.1 Objetivo General.....	10
1.4.2 Objetivos Específicos .....	10
1.5  JUSTIFICACIÓN.....	10
1.5.1 Justificación Técnica.....	10
1.5.2 Justificación Económica .....	11
1.5.3 Justificación Social .....	11
1.6  METODOLOGÍA .....	12
1.6.1 Metodología UWE.....	12
1.6.2 Metodología de Desarrollo SCRUM.....	13
1.6.3 Métricas de Calidad.....	13
1.6.4 Costos .....	14
1.6.5 Seguridad.....	15
1.6.6 Pruebas de Software .....	15
1.7  HERRAMIENTAS .....	16
1.8  LIMITES Y ALCANCES.....	17
1.8.1 Limites.....	17
1.8.2 Alcances.....	17
1.9  APORTES.....	19
CAPÍTULO II.....	20
2    MARCO TEÓRICO.....	21
2.1  SISTEMA INTERACTIVO.....	21
2.2  CURSOS EN LÍNEA.....	21
2.3  INGENIERÍA DE SOFTWARE .....	22

2.4	INGENIERIA WEB.....	22
2.5	MOODLE.....	22
2.6	PLATAFORMA VIRTUAL.....	23
2.7	METODOLOGIAS.....	24
2.7.1	METODOLOGIA UWE.....	25
2.7.2	SCRUM.....	29
2.8	MÉTRICAS DE CALIDAD DEL SOFTWARE.....	37
2.8.1	Factores de Calidad ISO 9126.....	37
2.9	SEGURIDAD DEL SOFTWARE.....	45
2.9.1	Principios de Seguridad.....	45
2.10	ESTIMACIÓN DE COSTOS.....	47
2.10.1	Modelo COCOMO.....	47
2.11	PRUEBAS DE FUNCIONAMIENTO.....	50
2.11.1	Pruebas de Caja Negra.....	50
2.11.2	Pruebas de Caja Blanca.....	51
2.11.3	Pruebas de Estrés.....	52
2.12	HERRAMIENTAS DE DESARROLLO.....	53
2.12.1	Definición de Servidor Web.....	53
2.12.2	Gestor de Base de Datos.....	54
2.12.3	Gestor de Base de Datos MariaDB.....	55
2.12.4	Lenguaje de Programación PHP.....	56
2.12.5	Framework Codeigniter.....	57
2.12.6	Definición de JQuery.....	61
2.12.7	Definición de JavaScript.....	62
2.12.8	Definición de AJAX.....	62
2.12.9	Bootstrap.....	63
2.12.10	Definición de HTML5.....	64
2.12.11	Definición de CSS3.....	65
CAPÍTULO III.....		67
3	MARCO APLICATIVO.....	68
3.1	ANÁLISIS DE REQUERIMIENTO.....	68
3.1.1	Recolección de Información.....	69
3.1.2	Especificación de Requerimientos.....	69
3.2	DISEÑO DEL SISTEMA.....	73
3.2.1	Diseño de Casos de Uso.....	73

3.2.2	Modelo de Contenido.....	87
3.2.3	Modelo de Navegación.....	89
3.2.4	Modelo de Presentación.....	98
3.3	SCRUM DIARIO .....	109
3.3.1	Comunicación de Sprint Backlogs.....	109
3.4	REVISIÓN DEL SCRIPT Y LA IMPLMENTACIÓN.....	110
3.4.1	Planificación de Entregas.....	110
3.4.2	Diseño de la Base de Datos.....	110
3.4.3	Diseño de Administración.....	112
3.4.1	Diseño de Usuario .....	119
3.5	RETROSPECTIVA DEL SPRINT.....	119
CAPÍTULO IV .....		121
4	SEGURIDAD, CALIDAD, COSTO Y PRUEBAS DE SOFTWARE.....	122
4.1	SEGURIDAD DE SOFTWARE.....	122
4.2	METRICA DE CALIDAD DEL SOFTWARE.....	123
4.2.1	Funcionalidad.....	123
4.2.2	Confiabilidad .....	126
4.2.3	Usabilidad .....	127
4.2.4	Mantenibilidad.....	129
4.2.5	Portabilidad.....	130
4.2.6	Definición de Medición.....	130
4.3	ESTIMACIÓN DE COSTO .....	131
4.3.1	COCOMO .....	131
4.4	PRUEBAS DE SOFTWARE.....	136
4.4.1	Pruebas de Caja Blanca.....	136
4.4.2	Pruebas de Caja Negra .....	139
4.4.3	Pruebas de Estrés .....	142
CAPÍTULO V .....		144
5	CONCLUSIONES Y RECOMENDACIONES.....	145
5.1	CONCLUSIONES.....	145
5.2	RECOMENDACIONES.....	146
BIBLIOGRAFÍA.....		147
ANEXOS.....		152

## ÍNDICE DE FIGURAS

Figura 1. Organigrama de la institución .....	4
Figura 2. Actividades del Diseño Conceptual UWE .....	26
Figura 3. Diagrama de Casos de Uso.....	27
Figura 4. Diagrama de Clases .....	27
Figura 5. Modelo de Navegación – Directorio .....	28
Figura 6. Modelo Presentación .....	28
Figura 7. Modelo de Proceso .....	29
Figura 8. Procesos del scrum.....	30
Figura 9. Atributos de Calidad del Estándar ISO/IEC. 9126 .....	38
Figura 10. Gráfico de Características de Funcionalidad .....	40
Figura 11. Gráfico de Características de Confiabilidad .....	41
Figura 12. Gráfico de Características de Usabilidad .....	43
Figura 13. Gráfico de Características de Mantenimiento .....	44
Figura 14. Gráfico de Características de Portabilidad.....	45
Figura 15. Métodos de Prueba de Caja Negra .....	50
Figura 16. Tabla de Decisión .....	51
Figura 17. Métodos de Prueba de Caja Blanca.....	52
Figura 18. Gráfico de Prueba de Estrés .....	53
Figura 19. Servidor Web Apache .....	54
Figura 20. Diagrama de Secuencias de MVC .....	59
Figura 21. Caso de Uso Web de Administración.....	73
Figura 22. Inicio de Sesión.....	75
Figura 23. Administrar Usuario .....	76
Figura 24. Administrar Participante .....	77
Figura 25. Administrar Roles y Permisos.....	78
Figura 26. Administrar Cursos y Publicación.....	79
Figura 27. Administrar Certificación.....	80
Figura 28. Administrar Participantes .....	81
Figura 29. Administrar Docente .....	82
Figura 30. Caso Web Pública.....	83
Figura 31. Inicio de Sesión.....	84
Figura 32. Registro de Usuario .....	85
Figura 33. Listado de Cursos .....	86
Figura 34. Perfil del Participante .....	87

Figura 35. Modelo de Contenido de la Base de Datos.....	88
Figura 36. Inicio de Sesión.....	89
Figura 37. Administrar Usuario y Grupos (administrador).....	89
Figura 38. Administrar Grupos (administrador).....	90
Figura 39. Administrar cursos (administrador, publicador).....	90
Figura 40. Administrar certificados (administrador, editor).....	91
Figura 41. Administrar Participantes (administrador, publicador).....	91
Figura 42. Administrar Inscritos y Preinscritos (administrador, publicador).....	92
Figura 43. Administrar Docentes (administrador, publicador).....	92
Figura 44. Administrar Blog (administrador, publicador, usuario, docente).....	93
Figura 45. Inicio de Sesión (participante).....	93
Figura 46. Registro Usuario (participante).....	94
Figura 47. Cursos (Usuario común, participante).....	94
Figura 48. Información o Detalles de Curso (Usuario común, participante).....	95
Figura 49. Perfil del Usuario (participante).....	95
Figura 50. Mis Cursos y Certificados (participante).....	96
Figura 51. Mis Cursos y Certificados (docente).....	97
Figura 52. Inicio de Sesión.....	98
Figura 53. Panel Inicio.....	98
Figura 54. Administrar Cursos (administrador, publicador).....	99
Figura 55. Administrar Configuración de Publicación (administrador, publicador).....	99
Figura 56. Editar Configuración de Publicidad Curso (administrador, publicador).....	100
Figura 57. Administrar Docentes (administrador, publicador).....	100
Figura 58. Listado de Docentes (administrador, publicador).....	101
Figura 59. Configuración de Certificados (administrador, publicador).....	102
Figura 60. Entrega de Certificados (administrador, publicador).....	103
Figura 61. Administrar Inscripciones (administrador, publicador).....	103
Figura 62. Administrar Preinscritos (administrador, publicador).....	104
Figura 63. Administrar Participantes (administrador, publicador).....	104
Figura 64. Administrar Blog (administrador, publicador, participante, docente).....	105
Figura 65. Inicio de Sesión (usuario).....	105
Figura 66. Nuestros Cursos (usuario).....	106
Figura 67. Inicio de Inscripción (usuario).....	106
Figura 68. Inicio de Blog (usuario).....	107
Figura 69. Diagrama de Actividad del Inicio de Sesión del Sistema.....	107



Figura 70. Diagrama de Actividad para la Pre-inscripción a un Curso .....	108
Figura 71. Diagrama de Actividad Para Agregar Docente a un Curso .....	108
Figura 72. Tablero de Tareas.....	110
Figura 73. Modelo físico de la base de datos .....	111
Figura 74. Inicio de Sesión.....	112
Figura 75. Login de la Pagina .....	112
Figura 76. Muestra de Código (inicio de sesión) .....	113
Figura 77. Administrar Curso (rol de administrador) .....	113
Figura 78. Muestra de Código (Administrar cursos).....	114
Figura 79. Administrar Certificación (rol de administrador) .....	114
Figura 80. Muestra de Código (Administrar certificados) .....	115
Figura 81. Administrar Inscripción y Preinscripción (rol de administrador).....	115
Figura 82. Muestra de Código (Administrar inscripción) .....	116
Figura 83. Administrar Participantes del Curso (rol de administrador) .....	116
Figura 84. Muestra de Código (Administrar participante).....	117
Figura 85. Administrar Blog (rol de administración) .....	117
Figura 86. Muestra de Código (Administrar blog).....	118
Figura 87. Administrar Live (rol de administración).....	118
Figura 88. Muestra de Código (Administrar live) .....	119
Figura 89 Autenticación Login.....	136
Figura 90 Grafo de Autenticación de Usuario.....	137
Figura 91. Diagrama de Inscripción a Cursos .....	138
Figura 92. Grafo De Flujo Registro de Un Inscrito a un Curso.....	138
Figura 93. Formulario de Autenticación de Usuarios .....	140
Figura 94. Registro a un Curso .....	141
Figura 95. Prueba de estrés de la página de inicio de los usuarios.....	143
Figura 96. Calculando las Pruebas de Estrés en el Inicio de la Pagina de Usuarios .....	143

## ÍNDICE DE TABLAS

Tabla 1. Roles del Scrum .....	30
Tabla 2. Características del ISO 9126 .....	38
Tabla 3. Tabla de Referencia Para el Cálculo de la Funcionalidad .....	39
Tabla 4. Traducción de la Complejidad Para Puntos de Función .....	40
Tabla 5. Tabla Para el Cálculo de la Confiabilidad.....	41
Tabla 6. Tabla Referencial para el Cálculo de la Usabilidad.....	42
Tabla 7. Tabla de Referencia del Mantenimiento .....	43
Tabla 8. Tabla de Referencia Para la Portabilidad.....	44
Tabla 9. Tabla de ponderación en porcentajes .....	45
Tabla 10. Tabla de Coeficientes COCOMO .....	47
Tabla 11. Tabla de Referencia Para Cálculo del Esfuerzo .....	48
Tabla 12. Tabla para el Cálculo del Tiempo de Desarrollo de Software.....	49
Tabla 13. Tabla de Referencia para el Cálculo del Personal Necesario .....	49
Tabla 14. Requerimientos Funcionales Parte de Administración .....	69
Tabla 15. Requisitos Funcionales Pública.....	70
Tabla 16 Requisitos no Funcionales .....	71
Tabla 17. Lista de Actores.....	71
Tabla 18. Descripción de Funciones.....	73
Tabla 19. Caso de Uso general Web de Administración.....	74
Tabla 20. Inicio de Sesión .....	75
Tabla 21. Administrar Usuario.....	76
Tabla 22. Administrar Participante .....	77
Tabla 23. Administrar Roles y Permisos.....	78
Tabla 24 Administrar Cursos y Publicación .....	79
Tabla 25. Administrar Certificación .....	80
Tabla 26. Administrar Participantes .....	81
Tabla 27. Administrar Docente.....	82
Tabla 28. Caso de Uso Web Pública .....	83
Tabla 29. Inicio de Sesión .....	84
Tabla 30. Registro de Usuario .....	85
Tabla 31. Cursos .....	86
Tabla 32. Perfil del Participante .....	87
Tabla 33 Tabla Del Estudio Módulos Fueron Satisfactorios .....	119

Tabla 34. Calculos de Valores de Ponderación.....	124
Tabla 35. Tabla de Ajuste de Complejidad.....	124
Tabla 36. Escala de Valores de las Preguntas.....	128
Tabla 37. Preguntas para Determinar la Usabilidad del Sistema .....	128
Tabla 38. Valores para Determinar la Mantenibilidad .....	129
Tabla 39. Conversión de puntos de Función a KDLC .....	131
Tabla 40. Tabla de atributos FAE .....	132
Tabla 41. Tabla de decisión caso de prueba de Caja Negra del Login del Sistema .....	140
Tabla 42. Registro a un Curso .....	141
Tabla 43. Tabla de Decisión Prueba de Caja Negra para el Registro de Activos .....	142

**ÍNDICE DE ECUACIONES**

( 1 ). Ecuación para el cálculo de la métrica de funcionalidad .....	39
( 2 ). Ecuación Funcionalidad del Sistema.....	39
( 3 ). Ecuación para la Calidad de Portabilidad .....	41
( 4 ). Ecuación Funcionalidad del Sistema.....	41
( 5 ). Función Exponencial.....	42
( 6 ). Ecuación de la Usabilidad .....	42
( 7 ). Ecuación de Madurez .....	43
( 8 ). Ecuación de Grado de Portabilidad.....	44
( 9 ). Ecuación de Esfuerzo .....	48
( 10 ). Ecuación Cantidad de Personal .....	49
( 11 ). Ecuación Punto de Función.....	123
( 12 ). Ecuación de Punto de Función.....	125
( 13 ). Ecuación de Tiempo de Funcionamiento .....	126
( 14 ). Ecuación de Confiabilidad .....	127
( 15 ). Ecuación de Facilidad de Uso .....	127
( 16 ). Ecuación de Mantenibilidad.....	129
( 17 ). Ecuación Grado de Portabilidad.....	130
( 18 ). Ecuacion para Kilo Lines of Delivered Code.....	131
( 19 ). Ecuación para Esfuerzo.....	134
( 20 ). Ecuación de Tiempo de Desarrollo .....	134
( 21 ). Ecuación Para Personas .....	135
( 22 ). Ecuación Para Costo del Software.....	135
( 23 ). Cálculo de la Complejidad .....	139

## RESUMEN

Se está trabajando en el desarrollo de un Sistema Interactivo de Gestión y Administración de Cursos en Línea para la Dirección de Posgrado, específicamente para la Unidad de Formación Académica Continua (UFAC). Este proyecto nace de la necesidad de modernizar y optimizar los procesos de la UFAC, que actualmente enfrenta desafíos en el control de registros académicos y financieros debido al uso de herramientas obsoletas como hojas de cálculo, formularios de Google y registros en papel. Estas herramientas no solo dificultan la obtención de información precisa y oportuna, sino que también representan un riesgo de pérdida de datos valiosos y falta de seguridad.

El nuevo sistema se desarrolla con el objetivo principal de mejorar la eficiencia en la gestión de la información académica y económica. Para ello, se ha optado por utilizar metodologías ágiles de desarrollo web como Uwe y Scrum. A nivel tecnológico, el sistema está basado en PHP como lenguaje de programación, MySQL como gestor de base de datos, y utiliza los frameworks CodeIgniter y JavaScript junto con sus diversas librerías.

Una de las principales funcionalidades del sistema es facilitar la gestión de cursos virtuales, optimizando el tiempo y proporcionando una amplia gama de funciones que permiten la interacción entre docentes, estudiantes y el público en general. Además, el sistema cuenta con una integración directa con Moodle, una plataforma educativa reconocida. También ofrece herramientas adicionales que enriquecen la experiencia educativa, como la posibilidad de generar reportes en diferentes formatos según las necesidades del usuario.

Con estas características y funcionalidades, el nuevo sistema busca no solo mejorar la eficiencia y la eficacia en la gestión y administración de cursos dentro de la UFAC, sino también brindar a los usuarios una experiencia más sencilla, segura y enriquecedora.

**Palabras claves:** Ufac, Scrum, Uwe, Sistema, Usuarios.

## ABSTRACT

Work is being done on the development of an Interactive Online Course Management and Administration System for the Graduate Management, specifically for the Continuing Academic Training Unit (UFAC). This project was born from the need to modernize and optimize UFAC processes, which currently faces challenges in the control of academic and financial records due to the use of obsolete tools such as spreadsheets, Google forms and paper records. These tools not only make it difficult to obtain accurate and timely information, but also pose a risk of loss of valuable data and lack of security.

The new system is developed with the main objective of improving efficiency in the management of academic and economic information. To do this, we have chosen to use agile web development methodologies such as Scrum, Uwe. At a technological level, the system is based on PHP as a programming language, MySQL as a database manager, and uses the CodeIgniter frameworks for the backend and JavaScript along with its various libraries.

One of the main functionalities of the system is to facilitate the management of virtual courses, optimizing time and providing a wide range of functions that allow interaction between teachers, students and the general public. In addition, the system has direct integration with Moodle, a recognized educational platform. It also offers additional tools that enrich the educational experience, such as the possibility of generating reports in different formats according to the user's needs.

With these features and functionalities, the new system seeks not only to improve efficiency and effectiveness in the management and administration of courses within the UFAC, but also to provide users with a simpler, safer and more enriching experience.

**Keywords:** Ufac, Scrum, Uwe, System, Users.

## LISTADO DE ABREVIATURAS

- **UFAC:** Unidad de Formación Académica Continua
- **UPEA:** Universidad Pública de El Alto
- **UWE:** UML Web Engineering
- **PHP:** Hypertext Preprocessor
- **Mysql:** SQL Structured Query Language
- **MSF:** Microsoft Solution Framework
- **HTML:** Hyper Text Markup Language
- **HTTP:** Hypertext Transfer Protocol
- **UML:** Unified Modeling Language
- **KLDC:** Kilo Lines of Delivered Code
- **TDES:** Tiempo de Desarrollo
- **FLDC:** Factor de Conversión LDC
- **PF:** Punto de Función

# CAPÍTULO I



## **1 MARCO PRELIMINAR**

### **1.1 INTRODUCCIÓN**

En la actualidad, la educación es un ámbito que requiere de una constante actualización diaria y dinamismo que se adapte a las necesidades de los usuarios que buscan una formación y facilitación idónea. Con ese propósito, entra la necesidad de crear un Sistema Interactivo de Gestión y Administración de Cursos en Línea, con Certificación Personalizada e Integración Social este sistema ayuda a la administración y gestión de los cursos con el intercambio de información y la interacción de los usuarios con el sistema.

Dentro de la Dirección de Posgrado, en la Unidad de Formación Académica Continua (UFAC) forman estudiantes interesados en la superación y capacitación personal. Ofrece diversos cursos educativos, pero enfrenta desafíos en el control de registros académicos y financieros. Actualmente, utilizan herramientas como hojas de cálculo, formularios de Google y registros en papel, lo que dificulta la obtención de información precisa, oportuna y confiable. Esto conduce a la pérdida de datos valiosos y una seguridad mínima.

En ese sentido, se está desarrollando un sistema con el objetivo de mejorar la eficiencia en la gestión de información académica y económica. Este sistema utiliza metodologías de desarrollo web como UWE y SCRUM. Está basado en PHP como lenguaje de programación, Mysql como gestor de base de datos, y los frameworks CodeIgniter por la parte de backend y JavaScript junto a sus diversas librerías con las que cuenta.

Se tiene como finalidad facilitar la gestión de cursos virtuales, optimizando el tiempo y proporcionando una amplia gama de funciones. Permite la interacción de docentes, estudiantes y el público en general; cuenta también con una integración a moodle. Además, ofrece herramientas adicionales para una educativa. Con estas características, se busca mejorar la eficiencia y la eficacia en la gestión y administración de cursos dentro de la unidad, brindando a los usuarios una experiencia más sencilla y enriquecedora.

## **1.2 ANTECEDENTES**

### **1.2.1 *Antecedentes Institucionales***

La Unidad de Formación Académica Continua (UFAC), dependiente de la Dirección de Posgrado de la Universidad Pública de El Alto, nace el 06 de febrero del año 2023 con la finalidad de desarrollar y consolidar el nivel académico del conocimiento, ofreciendo cursos cortos de calidad en todos los niveles de la formación académica, logrando de este modo una mejor articulación entre conocimientos prácticos y teóricos.

El desarrollo de los diferentes cursos cortos en las diferentes especialidades, buscan la excelencia personal de los participantes, ofreciéndoles temas de actualidad que son dictados por reconocidos docentes especialistas en las diferentes áreas.

#### **1.2.1.1 Visión.**

Nuestra VISIÓN, es lograr posicionamiento como unidad que da respuesta eficiente a demandas educativas y tecnológicas, estableciendo interacción entre los participantes, docentes y profesionales para intercambiar experiencias y contribuir con soluciones a las múltiples problemáticas que afrontamos.

#### **1.2.1.2 Misión.**

Nuestra MISIÓN es propiciar y generar conocimiento, a través de la capacitación teórica y práctica en cursos cortos, formando especialistas para solucionar problemas emergentes y difundir sus conocimientos.

#### **1.2.1.3 Objetivo.**

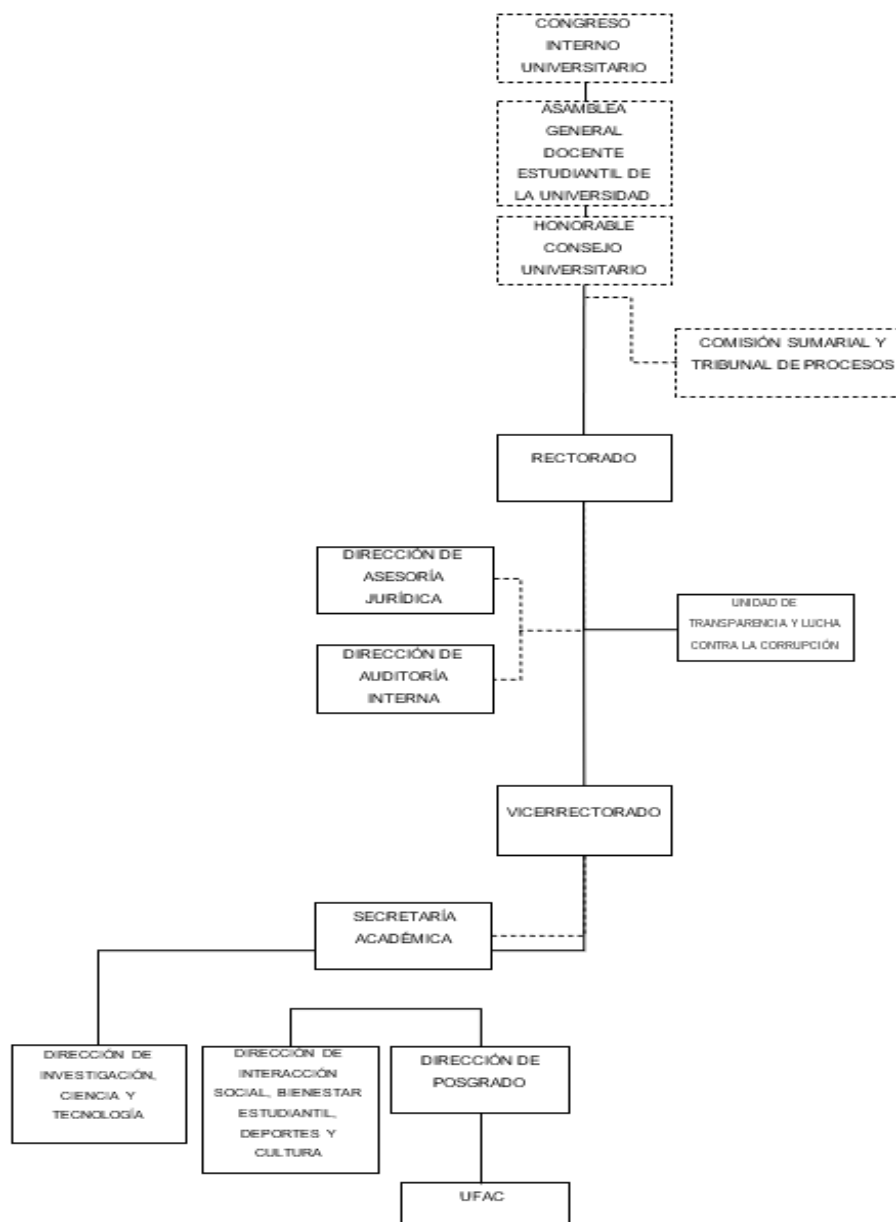
Promover la formación y especialización en el más alto nivel académico, mediante cursos cortos de actualidad.

### 1.2.1.4 Organigrama.

El organigrama mostrado en la figura es de la unidad de formación académica continua de posgrado, como se muestra hay varias unidades dentro de posgrado.

**Figura 1.**

**Organigrama de la institución**



*Nota.* El gráfico representa el esquema organizacional de las funciones de la Unidad de Formación Académica Continua, Dirección de posgrado, 2023.

## 1.2.2 Antecedentes Afines al Proyecto de Grado

### 1.2.2.1 Antecedentes Internacionales

- Según Yannick Warnier (2010), "*Plataforma E-learning & Colaboration Software Chamillo*". El objetivo Consiste en ampliar la disponibilidad de educación y promover la difusión del conocimiento a nivel mundial. El desarrollado del sistema esta con el lenguaje de programación PHP y depende de un sistema LAMP en el servidor, la creación fue en la Asociación Chamillo en Bélgica.
- Martin Dougiamas (2002), "*Plataforma E-learning Moodle*". Dice El propósito principal de Moodle es proporcionar a los profesores una plataforma en línea para establecer comunidades de aprendizaje. Esta desarrollado en PHP bajo GNU GPL versión III, soporta los principales gestores de base de datos, fue creada en la cuarta Universidad Tecnológica de Curtin, ubicada en Australia, utilizando la metodología UWE.
- (Espinoza Manchego, 2014), "*Análisis, diseño e implementación de e-learning para la tienda por departamentos ripley S.A.*". El objetivo resumido es implementar un sistema de aula virtual interactivo y personalizado, en lugar de la capacitación presencial, para reducir el tiempo y los costos asociados. Esto contribuirá a la instrucción oportuna de los colaboradores de la empresa, monitoreando su progreso y asegurando que se alineen con los objetivos estratégicos de la organización. Utilizo la metodología "MSF (Microsoft Solution Framework)" y las herramientas utilizadas para su desarrollo son php, javascript, como gestor de base de datos MySQL. Fue desarrollada en la facultad de ingeniería (USMP).

### 1.2.2.2 Antecedentes Nacionales

- Quispe Serrano (2020), "*Sistema de información web para el seguimiento académico*". En el proyecto de la Unidad Educativa Maranata se desarrolló un sistema web para

seguimiento académico, usando la metodología UWE, basada en el Proceso Unificado de aplicaciones web. Se empleó el servidor de base de datos MySQL, lenguaje de programación PHP, Framework CodeIgniter y tecnologías de diseño como HTML, JavaScript y CSS. La calidad y seguridad se aseguraron mediante los estándares ISO/IEC9126 y ISO-27002, respectivamente. Su desarrollado fue en la Universidad Pública de El Alto.

- Danilo Casati (2015), "*Plataforma E-learning Academia*". Tiene como objetivo brindar a todas las personas, incluso en comunidades remotas, la oportunidad de adquirir conocimientos en diversas áreas sin tener que asistir físicamente a un centro educativo. Esta desarrollada en el lenguaje de programación PHP, JavaScript con estilos bootstrap. Fue desarrollado en la Universidad Autónoma Gabriel René Moreno.
- (Velasco, 2009), "*Aulas virtuales en la educación boliviana un nuevo paradigma*". El proyecto tiene como objetivo desarrollar, implementar y administrar aulas virtuales que abarquen conferencias, servicios de asesoría y cursos en línea, todo ello en línea con la filosofía de la web 2.0, que se basa en la colaboración activa de los usuarios para crear, compartir y coproducir contenido.

### **1.2.2.3 Antecedentes Locales**

- (Callisaya, 2023, p. 7), "Aplicaciones educativas virtuales en procesos de enseñanza en el Centro Educativo Mutual La Paz en la ciudad de El Alto Distrito 2". Las aplicaciones educativas virtuales han tenido un impacto significativo en el Centro Educativo Mutual La Paz de El Alto Distrito 2, facilitando el aprendizaje interactivo y la

evaluación lúdica. Los beneficiarios principales son los maestros y estudiantes, mejorando su compromiso y participación en las clases.

- (Quispe, 2020, p. 9), "*Aplicación web para la gestión de procesos educativos*". Se creó una aplicación web para gestionar los procesos educativos en la Unidad Educativa Calama, permitiendo la organización eficiente de actividades curriculares. La aplicación fue desarrollada con tecnologías como React y JavaScript, y usó Cloud Firestore para el almacenamiento de datos. La metodología ágil Kanban, que optimiza el flujo de trabajo y la eficiencia, fue implementada en la gestión del proyecto.
- (Orosco Sullcata, 2020), "*Plataforma para la gestión de cursos virtuales*". Hoy en día la tecnología ha realizado enormes avances, día a día va aumentando su uso y esto genera importantes cambios en el mundo, las formas de comunicación han evolucionado bastante logrando que la sociedad tenga una gran aceptación de estas nuevas tecnologías, tomando en cuenta estos aspectos las instituciones han optado en la implementación de sistemas de información vía web, con el objetivo de difundir y promocionar los servicios que ofrecen.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

En el actual contexto educativo, el aprendizaje en línea ha adquirido una importancia sin precedentes debido a la pandemia. Sin embargo, tanto los usuarios, educadores y estudiantes se enfrentan a un desafío importante: la falta de un sistema que facilite la interacción de manera fluida y sencilla.

La Unidad de Formación Académica (UFAC) se ve afectada por esta carencia, lo que compromete su eficiencia en la promoción, inscripción y administración de los cursos. Actualmente, la dependencia de herramientas básicas como Excel, formularios de Google y

hojas tradicionales refleja una gestión deficiente que no aprovecha las oportunidades tecnológicas disponibles en la actualidad.

El uso de estas herramientas limitadas no solo implica una administración ineficiente, sino que también dificulta la automatización, el seguimiento de datos, la personalización de la experiencia del usuario y el análisis de métricas relevantes. Además, la falta de un sistema de cursos en línea puede obstaculizar la capacidad de la UFAC para competir en un mercado cada vez más aplicado en la actualidad.

Es crucial abordar este problema y buscar soluciones modernas que permitan a la UFAC contar con una plataforma eficiente y segura para respaldar los datos de sus estudiantes. Esto no solo protegerá la integridad de la información, sino que también impulsará el crecimiento de la institución al ofrecer una experiencia de inscripción y promoción de cursos más fluida y satisfactoria a los clientes.

### **1.3.1 *Problema Principal***

En la unidad se ha identificado una problemática fundamental que abarca la falta de organización en la gestión de la información y el trabajo, tanto en los ámbitos académicos como en lo económico. Además, se destaca la carencia crítica de un sistema adecuado para el control de los participantes, lo cual dificulta considerablemente la generación de reportes, informes y datos estadísticos, y acarrea la pérdida de datos valiosos para el área de marketing.

### **1.3.2 *Problemas Secundarios***

- Falta de seguimiento a los interesados, inscritos generando un control inefectivo de registro.
- Demora en las entregas de certificados a los participantes que optaron por pasar los cursos. Lo que genera desconformidad y reclamos a los coordinadores o administradores del curso.

- Falta de implementación de certificados con código QR. Actualmente, los certificados emitidos no incluyen esta tecnología adicional que permite la autenticación y verificación de manera más eficiente.
- Falta de una actualización de información detallada e inmediata sobre los cursos ofrecidos por la unidad, lo que lleva a los usuarios a buscar otras instituciones para su aprendizaje.
- Limitada interacción y colaboración. La carencia de herramientas de comunicación efectivas, como blogs colaborativos, transmisiones en vivo puede restringir las oportunidades de interacción y limitar el enriquecimiento del aprendizaje a través del trabajo en equipo.
- Insuficiente protección de la seguridad y privacidad de los datos personales en el manejo de información de estudiantes y profesores. Actualmente, existen fallos en los procesos y medidas de seguridad que comprometen la confidencialidad y protección de los datos sensibles.
- Se ha identificado la dificultad en la generación de consultas, reportes y datos estadísticos en archivos de Excel. Actualmente, este proceso requiere mucho tiempo y esfuerzo por parte del personal, lo que afecta la capacidad de tomar decisiones informadas de manera eficiente.

### **1.3.3 *Formulación del Problema***

En base a los problemas mencionados se llega a formular la siguiente interrogante:

¿De qué manera se puede optimizar la gestión y administración de cursos en línea con certificación personalizada e integración social para mejorar los procesos académicos de la UFAC?



## **1.4 OBJETIVOS**

### **1.4.1 *Objetivo General***

Desarrollar un Sistema Interactivo de Gestión y Administración de Cursos en Línea, con Certificación Personalizada e Integración Social, para la Unidad de Formación Académica Continua de la Dirección de Posgrado UPEA, para sistematizar y mejorar de manera eficaz y eficiente el control y manejo de la información dentro de la unidad.

### **1.4.2 *Objetivos Específicos***

- Realizar un análisis de la situación actual de la Unidad de Formación Académica Continua con el objetivo de identificar y comprender las necesidades específicas de la institución.
- Diseñar un sistema eficiente, intuitivo y accesible que brinde comodidad y facilidad de uso tanto para los administrativos, docentes y usuarios del sistema.
- Elaborar un sistema con una seguridad óptima para mejorar la integridad, confidencialidad y disponibilidad de la información de los usuarios.
- Realizar las pruebas de funcionamiento, identificar posibles fallos y aplicar correcciones necesarias para mejorar el rendimiento del sistema.
- Evaluar la calidad de software usando las métricas de calidad y funcionalidad del sistema.

## **1.5 JUSTIFICACIÓN**

### **1.5.1 *Justificación Técnica***

El sistema se desarrollará con tecnologías de software libres utilizando la metodología UWE y SCRUM como una unificación para un mejor trabajo. El ingreso al sistema se lo podrá realizar de cualquier dispositivo que cuente con una navegación a internet lo cual estará alojado en un servidor propio que se encuentra ya presente en la Dirección de Posgrado donde

facilitará un mayor control y protección de la información almacenada. Esta implementación también permitirá optimizar recursos y garantizar un entorno virtual de aprendizaje eficiente y seguro.

### **1.5.2 *Justificación Económica***

La implementación del sistema conlleva beneficios económicos significativos, ya que se ahorra en infraestructura al no ser necesario adquirir licencias o contratar servicios externos. Además, el sistema elimina los gastos asociados con los materiales de escritorio utilizados por el personal administrativo; asimismo, se logra una ampliación del alcance educativo y se brinda mayor flexibilidad a los estudiantes en cuanto a horarios y ubicación. Estas mejoras contribuyen a una mejor calidad educativa y generan ahorros económicos para la unidad. estudiantes a la institución, ya que se destaca en términos de Marketing y ofrece una experiencia moderna y eficiente.

### **1.5.3 *Justificación Social***

El sistema permitirá un acceso amplio a la educación en línea, sin importar la ubicación geográfica o las limitaciones de tiempo de las personas. Esto fomenta la inclusión social al brindar oportunidades educativas a aquellos que no pueden acceder a la educación presencial, como aquellos que viven en áreas remotas o tienen responsabilidades familiares o laborales.

La educación en línea permitirá a los estudiantes adaptar su aprendizaje a sus necesidades y horarios individuales. Esto beneficia especialmente a aquellos que trabajan o tienen otros compromisos, ya que les brinda la flexibilidad de estudiar a su propio ritmo y gestionar su tiempo de manera más eficiente.

## 1.6 METODOLOGÍA

### 1.6.1 Metodología UWE

Según Rossainz (2016), UWE es una metodología basada en el Proceso Unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones Web, este cubre todo el ciclo de vida de las aplicaciones Web. El objetivo principal del enfoque UWE es proporcionar: un lenguaje de modelado de dominio específico basado en UML, una metodología basada en modelos, herramientas de soporte para el diseño sistemático y herramientas de soporte para aplicaciones web automáticas de semigeneración.

La notación UWE proporciona un perfil UML para un dominio específico de la Web y se define como una ligera extensión de UML. Centra su atención en aplicaciones personalizadas o adaptativas y cubre todo el ciclo de vida de este tipo de aplicaciones. Análisis de necesidades, diseño de sistemas, codificación de software, pruebas. La fase de implementación y mantenimiento.

#### 1.6.1.1 Fases de la Metodología UWE

- **Análisis de Requerimientos:** En esta etapa de la metodología se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir el sistema.
- **Diseño del Sistema:** Se baja en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define como se cumplirán estos requisitos y la estructura que tendrá el sistema.
- **Codificación de Software:** Durante esta etapa se realizan las tareas de programación del sistema, que consiste en llevar al código fuente el diseño del sistema.
- **Pruebas:** Las pruebas son utilizadas para asegurar el correcto funcionamiento de secciones de código.

- **Fase de Implementación:** Es el proceso por el cual los programas desarrollados son transferidos al computador de destino, inicializados y eventualmente configurados para ser utilizados por el usuario final.
- **El Mantenimiento,** en esa etapa se mejora y optimiza el software ya desarrollado e instalado, donde también se pueden corregir y depurar errores que puedan haberse encontrado.

### **1.6.2 Metodología de Desarrollo SCRUM**

Según (Sutherland, 2022 p.2), Scrum es un marco ligero que ayuda a las personas, equipos a generar valor a través de soluciones adaptables para problemas complejos.

Scrum se basa en la inteligencia colectiva de las personas que lo utilizan, en lugar de proporcionar a las personas instrucciones detalladas, las reglas de Scrum guían sus relaciones e interacciones, en el marco se pueden emplear diversos procesos, técnicas y métodos. Scrum envuelve las prácticas existentes o las hace innecesarias también visibiliza la eficacia relativa de la gestión actual, el entorno y las técnicas de trabajo, de modo que se pueden realizar mejoras.

### **1.6.3 Métricas de Calidad**

#### **1.6.3.1 Norma ISO-9126.**

ISO 9126 es un estándar internacional para la evaluación del software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software. Los requisitos del software constituyen el funcionamiento para medir la calidad. El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad como la funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de recibir mantenimiento, portabilidad. Igual que otros factores de la calidad del software estudiados en las subsecciones

anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa (Pressman, 2010, p. 343).

- **Funcionalidad:** Esta característica se enfoca en la capacidad que el software o sistema tiene para cumplir con las funciones y requerimientos solicitados por el usuario o cliente final.
- **Fiabilidad:** Es la capacidad que el software tiene para mantener su rendimiento donde también se evalúan la tolerancia a fallos y la disponibilidad.
- **Usabilidad:** Esta característica se enfoca en la facilidad de uso del sistema por el usuario final, es decir que el sistema sea fácilmente comprensible por el usuario y que el sistema también sea atractivo visualmente.
- **Eficiencia:** La eficiencia está directamente relacionada con la capacidad del software de utilizar la menor cantidad de recursos en hardware para cumplir sus tareas y operaciones.
- **Mantenibilidad:** Es la capacidad que tiene el software para ser modificado, es decir que al realizar modificaciones al software este pueda continuar funcionando y permita ser mejorado de manera efectiva.
- **Portabilidad:** Es la capacidad que tiene el sistema para poder ser transferido a otro entorno de ejecución sin la necesidad de realizar grandes modificaciones.

#### **1.6.4 Costos**

La estimación de costos de un producto de software se basa en distintos métodos como ser la Estimación de costos basada en la Experiencia, Estimación de costos Delphi, Estimación de costos COCOMO, estimación de costos de Monte Carlo, entre otros.

#### **1.6.4.1 COCOMO.**

El método COCOMO es un modelo constructivo para la estimación de costes y se han creado fórmulas para medirlos teniendo en cuenta factores como el tamaño del sistema, el esfuerzo requerido, el tiempo empleado y el número de personas necesarias (Sommerville, 2011, p. 637).

#### **1.6.5 Seguridad**

En cuanto a Sommerville, (Sommerville, 2011, p. 302), La seguridad es una de las características más importantes que debe contener un sistema debido a que de ello dependerá que la accesibilidad de los datos solo sea a usuarios autorizados.

Como características importantes de la seguridad estan: La Disponibilidad, Confidencialidad, Integridad, Autenticación.

#### **1.6.6 Pruebas de Software**

La creación de estrategias de pruebas efectivas es importante para identificar la mayor cantidad de errores en el menor tiempo y esfuerzo posible. Estas estrategias pueden aprovechar tanto el enfoque de pruebas de caja blanca como el de caja negra y así mismo las pruebas de estrés.

##### **1.6.6.1 Caja Blanca**

Según Pressman, Roger (2010), la prueba de Caja Blanca, en ocasiones llamada prueba de caja de vidrio, consiste en probar el código. Se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.

- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los métodos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

#### **1.6.6.2 Caja Negra.**

Las pruebas de Caja Negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de Caja Negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman, 2002, p. 423).

#### **1.6.6.3 Pruebas de Estrés.**

Las pruebas de estrés de software, también conocidas como pruebas de resistencia son un recurso utilizado durante la etapa de testing, con el objetivo de probar los límites de un sistema y prever escenarios de riesgo ante cargas extremas (Sanches, 2015, p. 55).

### **1.7 HERRAMIENTAS**

Se utilizarán las siguientes herramientas durante el proceso de desarrollo e implementación del sistema:

- **PHP:** es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo (González, 2006).

- **JAVASCRIPT:** JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, que se utiliza principalmente en el desarrollo web para añadir interactividad y funcionalidades dinámicas a las páginas web (Simpson, 2015).
- **MariaDB:** Es un sistema de administración de bases de datos relacionales (DBMS) de código abierto que es un reemplazo directo compatible para la tecnología de base de datos MySQL ampliamente utilizada (Bartholomew, 2014).
- **CODEIGNITER 4:** Es un framework de desarrollo de aplicaciones web, escrito en PHP, que facilita la creación de aplicaciones web dinámicas, seguras y eficientes siguiendo el patrón de diseño Modelo-Vista-Controlador (MVC). CodeIgniter es conocido por su simplicidad y facilidad de uso, así como por su rendimiento y velocidad en comparación con otros frameworks PHP (Ezell, 2021).

## 1.8 LIMITES Y ALCANCES

### 1.8.1 *Limites*

El presente proyecto se limitará a subir tareas al sistema por parte de los estudiantes, y docentes, también a la descarga individual de cada certificado por parte del estudiante, otra de las limitantes son las transmisiones en vivo que solo se realizaran en global y no por curso y el tema de la certificación no incluirá la firma digital de la Universidad Pública de El Alto.

### 1.8.2 *Alcances*

El sistema propone lo siguiente:

- El blog educativo proporciona un espacio en línea para compartir información y recursos educativos relevantes y actualizados.
- Gestión de Marketing facilita que los docentes suban videos introductorios para promocionar los temas de sus cursos y captar el interés de los usuarios.



- La inscripción en línea permite a los usuarios registrarse rápidamente, ingresar datos y pagos. Hace la gestión de registros más fácil, genera reportes y facilita la comunicación.
- La comunicación tendrá posibilidad de transmitir eventos gratuitos o educativos, que se pueden promocionar en la plataforma para atraer nuevos usuarios y mejorar la visibilidad de la unidad tecnológica UFAC.
- El sistema permite a usuarios pedir información sobre cursos disponibles, ya sea en general o específicos. Tras solicitar, recibirán detalles del curso y mensajes personalizados vía WhatsApp para Marketing.
- Permitir a los administradores publicar cursos en la plataforma y asignarles las siguientes características como: fecha de publicación, fecha de inicio y fecha fin para el curso, horarios, carga horaria, inversión, descuentos.
- Será capaz de generar certificados con diferentes formatos de tamaño y con diferentes niveles de seguridad y autenticación como el QR.
- Se tendrá reportes de los montos de cada curso, el módulo proporciona informes de montos pagados por inscritos en cursos y permite visualizar datos estadísticos relacionadas con estos montos.
- Este módulo permitirá al administrador del sistema establecer qué acciones o funciones pueden realizar los usuarios en función de sus roles.
- En la gestión de notas se podrá ver las calificaciones de los participantes en curso y así también todos los cursos que se llevaron con sus respectivas notas.
- Contará con una integración a Moodle para un fácil acceso y una mejor conexión entre ambas plataformas.
- Se contará con una certificación personalizada con verificación del usuario.

## **1.9 APORTES**

Durante los años de formación en la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto, se ha adquirido un sólido conjunto de conocimientos. Aprovechando esta experiencia, se ha desarrollado un software que brindará un valioso apoyo en el ámbito de la tecnología para la unidad de formación académica.

El sistema creado beneficiará a toda la comunidad de la institución. Los usuarios podrán acceder a información actualizada sobre los cursos que se ofrecen al público en general, y aquellos que se inscriban podrán obtener certificaciones con autenticación única. Los docentes podrán planificar sus cursos de manera eficiente, mientras que los administrativos tendrán un acceso sencillo a la información de docentes y estudiantes, permitiéndoles hacer un seguimiento de las actividades académicas y generar reportes económicos.

La implementación de esta plataforma mejorará el prestigio de la UFAC ante la población en general, lo cual se traducirá en un aumento en la demanda de estudiantes inscritos. Esto, a su vez, impulsará el crecimiento y desarrollo de la institución en general.

# CAPÍTULO II

## **2 MARCO TEÓRICO**

La base teórica que sustenta este proyecto de grado permite al lector comprender mejor este sistema. Aquí encontrará bases fundamentos y adicionales y se centrará en temas específicos. En este contexto, se discutirán los principios de la metodología de desarrollo Uwe y Scrum junto con las herramientas utilizadas:

Lenguaje de programación PHP, soportado por el framework CodeIgniter para el lado del servidor, javascript, jquery. Todo esto en conjunto con el sistema de gestión de bases de datos MariaDB. También se detallarán los términos y pasos correspondientes al modelo de métricas de calidad ISO/IEC 9126, reconocido internacionalmente, así como al modelo de estimación de costos COCOMO.

### **2.1 SISTEMA INTERACTIVO**

Un entorno tecnológico que facilita la interacción y comunicación bidireccional entre los usuarios y el sistema, permitiendo la entrada, visualización y manipulación de información en tiempo real, con el objetivo de lograr una experiencia de usuario más eficiente y efectiva en diversas aplicaciones y contextos. (Norman, 1986, p. 130).

### **2.2 CURSOS EN LÍNEA**

(TANAKA OHASHI, 2021) Se entiende por educación en línea a aquella en la que los docentes y estudiantes participan en un entorno digital a través de las nuevas tecnologías y de las redes de computadoras, haciendo uso intensivo de las facilidades que proporciona Internet y las tecnologías digitales.

(Schrum, 1998, p. 78) La literatura de la educación a distancia para cursos donde se usa correspondencia, video, la transmisión por televisión y por satélite, no necesariamente describen el proceso incluido en los cursos en línea.

### **2.3 INGENIERÍA DE SOFTWARE**

(Maida, 2015, p. 16) Sintetizando los diversos conceptos acerca de la Ingeniería de Software, podemos describirla como el campo informático que aplica de manera inteligente los fundamentos de la ingeniería. Su meta es lograr soluciones informáticas viables desde una perspectiva económica, alineadas con las demandas empresariales concretas. Esto involucra considerar los procesos de creación y mantenimiento de software a lo largo del tiempo, tomando en cuenta tanto la evolución constante como los costos estimados.

### **2.4 INGENIERIA WEB**

Según (Díaz, 2005, p. 31) La ingeniería Web toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un Sistema basado en computadora.

(TORRES, 2003, p. 6) La ingeniería web es un área que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

### **2.5 MOODLE**

Según (Dougiamas, 2003) "Plataforma E-learning Moodle" Objetivo General de Moodle es brindar a los educadores una plataforma web para crear comunidades de aprendizaje en línea. El código de esta plataforma está escrito en PHP bajo GNU GPL versión

3, soporta los principales gestores de base de datos, fue creada en la 3 Universidad Tecnológica de Curtin, ubicada en Australia, utilizando la metodología UWE.

Como vemos en la definición del nombre de Moodle, este se refiere a “objetos de aprendizaje”, normalmente de tamaño pequeño y diseñados para distribuirse en internet posibilitando el acceso simultaneo a la información por parte de múltiples usuarios. Este hecho es fundamental ya que a partir de ahora no nos basaremos en la mera lectura de unos apuntes sino en la creación de estos “objetos de aprendizaje”, plenos de significado que siguen secuencias didácticas en las que el profesor guía a los alumnos posibilitando su autoaprendizaje.

## **2.6 PLATAFORMA VIRTUAL**

Es un entorno informático en el que nos encontramos con muchas herramientas agrupadas y optimizadas para fines docentes. Su función es permitir la creación y gestión de cursos completos para internet sin que sean necesarios conocimientos profundos de programación. Las plataformas deben poseer unas aplicaciones mínimas, que se pueden agrupar en:

- Herramientas de gestión de contenidos, que permiten al profesor poner a disposición del alumno información en forma de archivos (que pueden tener distintos formatos: pdf, xls, doc, txt, html) organizados a través de distintos directorios y carpetas.
- Herramientas de comunicación y colaboración, como foros de debate e intercambio de información, salas de chat, mensajería interna del curso con posibilidad de enviar mensajes individuales y/o grupales.
- Herramientas de seguimiento y evaluación, como cuestionarios editables por el profesor para evaluación del alumno y de autoevaluación para los mismos, tareas, informes de la actividad de cada alumno, plantillas de calificación.

- Herramientas de administración y asignación de permisos. Se hace generalmente mediante autenticación con nombre de usuario y contraseña para usuarios registrados.

Por ello queda claro que las plataformas deben de tener herramientas de gestión, de comunicación, de evaluación, de administración y herramientas de hipertexto. Si en un software educativo faltase alguna de esta herramienta, no se podría considerar plataforma virtual. (Diaz, 2009)

## **2.7 METODOLOGIAS**

Se ha combinado las metodologías UWE y SCRUM para garantizar el desarrollo óptimo de nuestro sistema. Esta fusión permite aprovechar lo mejor de cada uno, asegurándonos de un enfoque estructurado de ingeniería del software con UWE al mismo tiempo que nos brinda la flexibilidad y adaptabilidad características del marco ágil de SCRUM. Gracias a esta decisión estratégica, podemos alcanzar nuestros objetivos de desarrollo priorizando la calidad, el trabajo en equipo y la entrega continua de valor para nuestros usuarios. En los siguientes subtítulos, se explorarán las características de cada una de estas metodologías y cómo se combinarán para la implementación del sistema. Se analizará la metodología UWE en la fase uno, que se enfoca en el análisis de requerimientos debido a su eficaz manejo al solicitar los requisitos necesarios. Para la fase dos, se llevará a cabo el diseño del sistema, también dentro del marco de UWE. La fase tres se abordará con el Scrum diario para una revisión regular del progreso del trabajo. La fase cuatro destacará la metodología UWE para las pruebas del sistema. En la fase cinco, se considerará tanto la implementación como la retrospectiva del sprint, integrando elementos de ambas metodologías para una evaluación completa del proceso.

### 2.7.1 METODOLOGIA UWE

UWE es una metodología basada en el Proceso Unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones Web, este cubre todo el ciclo de vida de las aplicaciones Web. El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML, una metodología dirigida por modelos, herramientas de soporte para el diseño sistemático y herramientas de soporte para la generación semiautomática de Aplicaciones Web. (Lopez, 2016).

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas. (Minguez, 2011). Las fases o etapas a utilizar son:

- **Análisis de Requerimientos**, en esta etapa de la metodología se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir el sistema.
- **Diseño del Sistema**, Se baja en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define como se cumplirán estos requisitos y la estructura que tendrá el sistema.
- **Codificación de Software**, durante esta etapa se realizan las tareas de programación del sistema, que consiste en llevar al código fuente el diseño del sistema.
- **Pruebas**, las pruebas son utilizadas para asegurar el correcto funcionamiento de secciones de código.
- **Fase de Implementación**, es el proceso por el cual los programas desarrollados son transferidos al computador de destino, inicializados y eventualmente configurados para ser utilizados por el usuario final.
- **Mantenimiento**, es el proceso de control en el sistema ya desarrollado e instalado que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas.



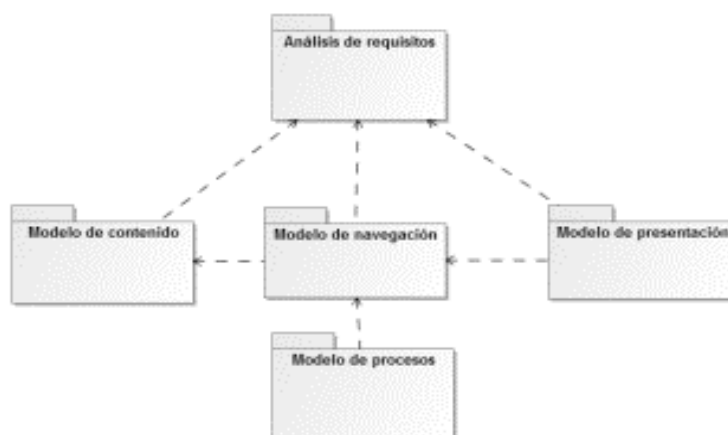
### 2.7.1.1 Diseño Conceptual.

Corresponde a plasmar y diseñar en diagramas los requisitos que se descubrieron en la fase de análisis de requerimientos. Esto incluirá establecer estructura requerida para la aplicación web.

Los diagramas a utilizarse serán:

**Figura 2.**

**Actividades del Diseño Conceptual UWE**



*Nota.* Actividades del diseño conceptual de UWE, Nieves Guerrero, 2018, Revista Latinoamericana de Software

#### 2.7.1.1.1 Diseño de Casos de Uso.

UML Está destinado a el apoyo de herramientas de modelado visuales e interactivas que generan código e informes. La especificación UML no define un proceso estándar, pero pretende ser útil en un proceso de desarrollo iterativo. Tiene la intención de apoyar la comprensión del Software para cualquier persona común

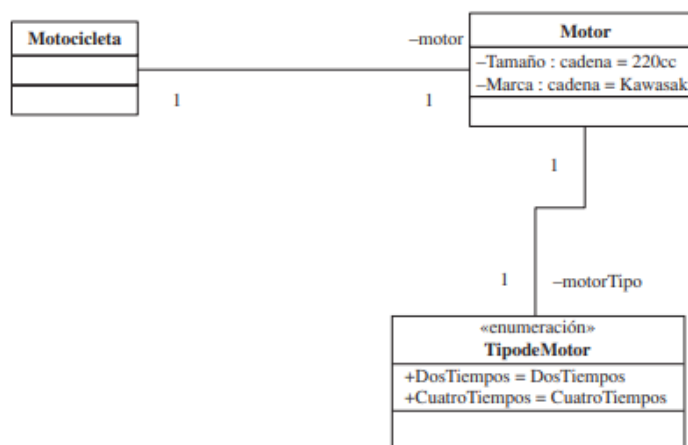
Entre los diagramas más comunes y utilizados de UML tenemos:

**Figura 3.****Diagrama de Casos de Uso**

*Nota.* Diagrama de casos de uso en UML, Kimmel, 2007, McGraw-Hill.

**2.7.1.1.2 Modelo de Contenido.**

Los diagramas de clases describen la estructura estática de un sistema

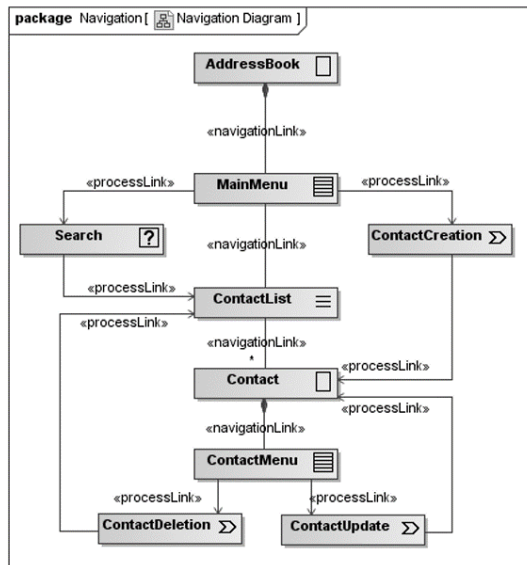
**Figura 4.****Diagrama de Clases**

*Nota.* Diagrama de clases de UML, Kimmel, 2007, tomado de McGraw-Hill.

**2.7.1.1.3 Modelo Navegacional.**

El Modelo de Navegación se muestra detalladamente cuáles son las clases de navegación y de proceso que pertenecen a un sitio web.

**Figura 5.**  
**Modelo de Navegación – Directorio**

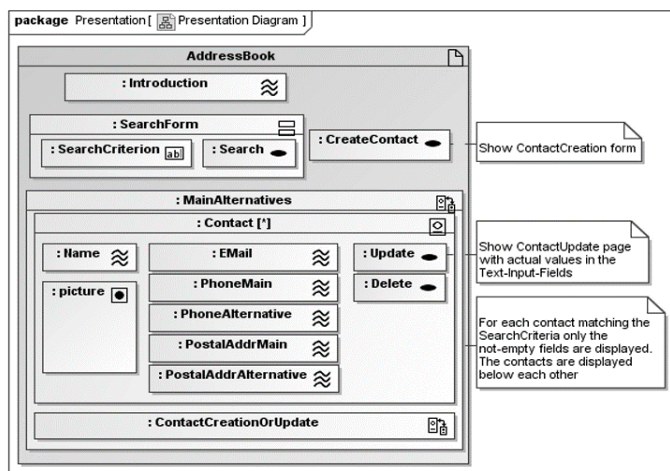


Nota. Modelo de navegación, Maximilians, 2016, UWE - UML-based Web Engineering.

**2.7.1.1.4 Modelo de Presentación.**

El Modelo Navegacional se muestra cuáles son las clases de navegación y de proceso que pertenecen a una sitio web.

**Figura 6.**  
**Modelo Presentación**



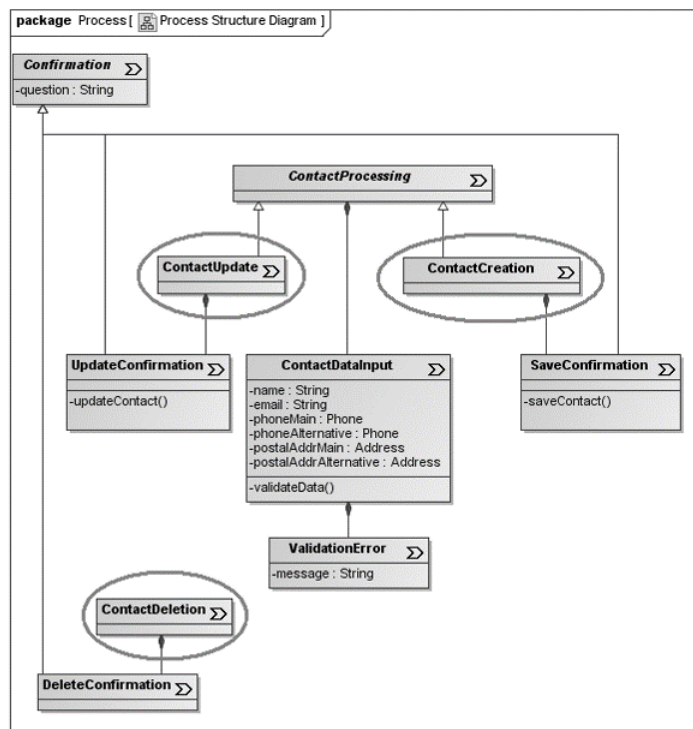
Nota. Modelo de presentación, Maximilians, 2016, UWE - UML-based Web Engineering

### 2.7.1.1.5 Diagrama de Actividades.

Este modelo representa la parte dinámica del sistema web, especificando la funcionalidad de las transacciones y de los flujos de trabajo complejos de las actividades.

**Figura 7.**

**Modelo de Proceso**

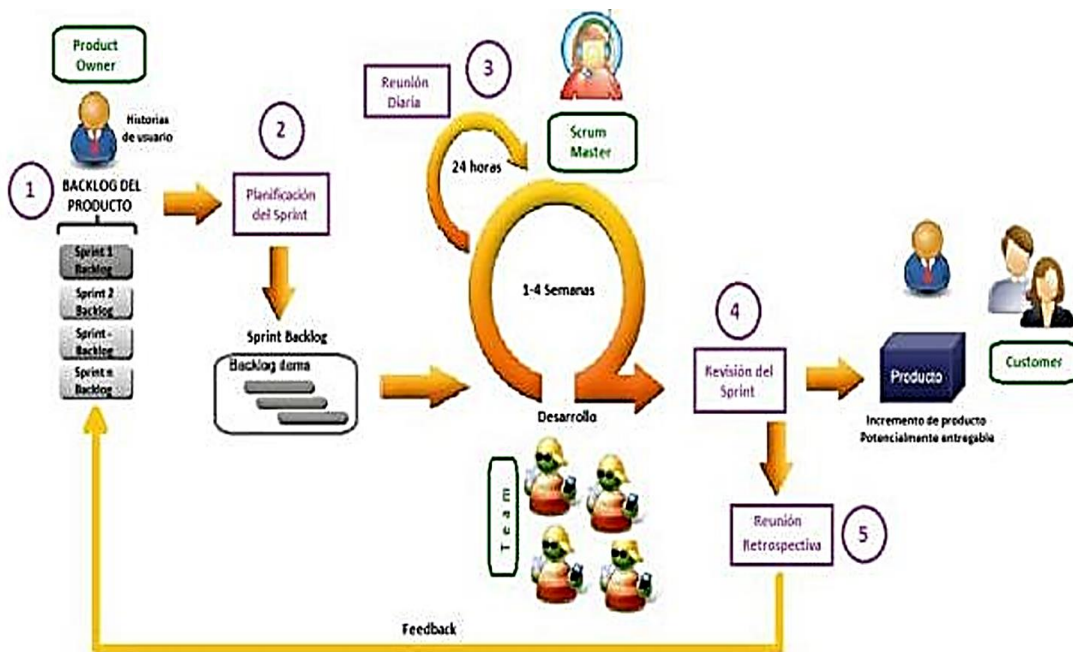


*Nota.* Modelo de proceso Maximilians, 2016, UWE - UML-based Web Engineering

### 2.7.2 SCRUM

Esta metodología es un marco diseñado para lograr una colaboración efectiva en equipo en proyectos, utilizando un conjunto de reglas y artefactos y definiendo roles que crean la estructura necesaria para que funcione.

**Figura 8.**  
**Procesos del scrum**



Nota. Muestra el proceso del scrum, Rodriguez, 2008, Estudio de la aplicación de metodologías ágiles para la evolución de productos software

- **Roles Scrum**

(Schwaber & Sutherland, 2013) Descripción de roles en la tabla

**Tabla 1.**

**Roles del Scrum**

Nombre de los roles Scrum	Descripción
<b>Scrum master</b>	El Scrum Master es el responsable en asegurar que se entienda y se ejecute el Scrum. Los Scrum Masters realizan estas acciones asegurándose de que el Equipo trabajen ajustándose a la teoría, prácticas y reglas de que se tiene en el Scrum.
<b>Dueño del producto</b>	Su responsabilidad principal es optimizar el valor del producto y supervisar el trabajo del equipo de desarrollo. Esto incluye la gestión y priorización de la lista ordenada de funcionalidades requeridas, conocida como Product Backlog.
<i>//...</i>	

...//

---

**Equipo de Desarrolladores**

El equipo de desarrollo está formado por individuos sin jerarquías, todos son desarrolladores con igualdad de roles y responsabilidades. El tamaño ideal del equipo oscila entre tres y nueve personas.

---

*Nota.* Roles scrum, Schwaber, 2016, La guía de Scrum

En lo visto en la tabla anterior observa que el Modelo Scrum está conformado por las siguientes fases:

- **FASE N°1:** Definir backlog del producto
- **FASE N°2:** Planificación del backlog
- **FASE N°3:** Scrum diario
- **FASE N°4:** Revisión del Sprint
- **FASE N°5:** Retrospectiva del Sprint

Según (Sutherland, 2016, p. 8) Existen diferentes eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en scrum. Todo evento es de compartimiento o periodos limitados.

Una vez que se inicia un Sprint, su duración queda establecida y no puede ser modificada ni acortada. En cuanto a los demás eventos, concluyen en el momento en que se logra su objetivo, garantizando la asignación adecuada de tiempo sin permitir que se derroche este recurso en el proceso. Más allá del Sprint es un marco para diversos eventos, cada uno de estos eventos dentro del marco de Scrum representa una oportunidad formal para llevar a cabo inspecciones y adaptaciones en aspectos específicos. Estos eventos fueron concebidos con el propósito expreso de respaldar los pilares fundamentales de transparencia e inspección. La omisión de algunos de estos eventos conlleva una disminución en la claridad de la situación y resulta en la pérdida de oportunidades para la revisión y ajuste necesarios.

A continuación, iremos detallando las fases del scrum:

### **2.7.2.1 Definir Backlog del Producto (fase 1).**

Es donde empieza todo, básicamente es una lista priorizada de requisitos, historias o funcionalidades que el cliente desea, descritas en terminología del usuario final, Se llama a esto historias de usuario o a veces simplemente elementos de la pila.

### **2.7.2.2 Planificación de Backlog (fase 2).**

El propósito de la planificación de Sprint es proporcionar al equipo suficiente información como para que puedan trabajar en paz y sin interrupciones durante las semanas, y para ofrecer al dueño de producto suficiente confianza como para permitirselo. Una planificación de Sprint produce, concretamente:

- Una meta de Sprint
- Una lista de miembros (y su nivel de dedicación, si no es del 100%)
- Una Pila de Sprint o Sprint Backlog (lista de historias incluidas en el Sprint)
- Una fecha concreta para la Demo del Sprint
- Un lugar y momento definidos para el Scrum Diario

Es importante que el encargado del sistema producto asista a la planificación del Sprint.

De acuerdo a (Sutherland, 2016, p.10) El trabajo a realizar durante el Sprint se planifica en la reunión de planificación del Sprint (Sprint Planning). Este plan se crea mediante el trabajo colaborativo de todo el equipo Scrum.

### **2.7.2.3 Scrum Diario (fase 3).**

El Scrum diario según (Sutherland, 2016, p.11) es una reunión con un bloque de tiempo de 15 a 20 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan

para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente. El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad.

- **Prioridad 1:** Una meta de Sprint y una fecha para la demo. Esto es lo mínimo que se necesita para comenzar un Sprint. El equipo tiene una meta y una fecha de finalización, y puede trabajar directamente con la pila de producto. Se debe considerar seriamente organizar una nueva reunión de planificación de Sprint mañana mismo, pero si realmente se necesita que el Sprint comience entonces probablemente se pueda hacer con esto.
- **Prioridad 2:** Lista de qué historias ha aceptado terminar el equipo en este Sprint.
- **Prioridad 3:** Una estimación para cada historia del Sprint.
- **Prioridad 4:** “Cómo probarlo”, relleno para cada historia del Sprint.
- **Prioridad 5:** Cálculos de velocidad y recursos, como chequeo de la planificación del Sprint. Incluyendo una lista de los miembros del equipo y sus compromisos.
- **Prioridad 6:** Un sitio y hora específicos para la realización del Scrum diario. Sólo se necesita un momento para decidirlo, pero si se queda sin tiempo el Scrum Master puede simplemente decidir esto después de la reunión y mandar un correo a todo el mundo.
- **Prioridad 7:** Historias divididas en tareas. Esta división puede sin embargo hacerse diariamente durante los Scrum diarios, pero interferirá levemente el flujo del Sprint.

#### **2.7.2.4 Revisión del Sprint (fase 4).**

La revisión del Sprint, involucra la presencia del equipo Scrum, Scrum Master, Product Owner con todas las personas implicadas en el proyecto. La duración máxima de esta reunión



es de cuatro horas, y el objetivo es presentar al propietario del producto y a las nuevas funcionalidades implementadas.

Una demo de Sprint bien ejecutada, aunque parezca poco espectacular, tiene un efecto muy profundo:

- El equipo de desarrollo obtiene reconocimiento por sus logros con los objetivos.
- Otras personas se enteran de lo que está haciendo el equipo.
- La demo consigue feedback vital de los interesados.
- Las demos son (o deberían ser) un evento social donde diferentes equipos pueden interactuar unos con otros y discutir su trabajo. Esto es muy valioso.
- Hacer una demo fuerza al equipo a acabar realmente las cosas y entregarlas (incluso aunque sea sólo en entorno de pruebas). Sin las demos, se sigue consiguiendo enormes montones de cosas terminadas al 99%. Con las demos puede que se consigan menos cosas terminadas, pero estas están realmente terminadas, lo que es mucho mejor que tener una enorme pila de cosas que están más o menos listas y que se pulirán en el próximo Sprint.
- Si un equipo se ve obligado a realizar una demo de Sprint, incluso aunque no tengan mucho que realmente esté funcionando, la demo será embarazosa. El equipo tartamudeará y tropezará mientras hace la demo y el aplauso de después será frío. La gente sentirá un poco de pena por el equipo, algunos se enfadarán con ellos por hacerles perder el tiempo con una demo pésima. Esto duele, pero el efecto es similar al de una amarga medicina. En el próximo Sprint, el equipo intentará por todos los medios tener

cosas terminadas. El equipo sabe que tendrán que hacer una demo pase lo que pase, lo que incrementa significativamente las posibilidades de que haya algo útil que demostrar.

Lista de comprobación para demos de Sprint

- Se debe asegurar de presentar claramente el objetivo del Sprint. Si hay personas en la demo que no saben nada sobre tu producto, tómate un par de minutos para describirlo.
- No se debe perder mucho tiempo preparando la demo, especialmente en llamadas presentaciones. Se debe concentrar en mostrar código funcionando.
- Mantener el paso rápido, es decir, se debe concentrar la preparación en hacer que la demo sea rápida en lugar de bonita.
- Mantenga la demostración a nivel de negocio, dejando a un lado los detalles técnicos. Se debe enfocarse en "qué se hizo" en lugar de "cómo se hizo".
- Se debe permitir que la audiencia pruebe el producto lo más posible.
- No se deben mostrar muchos errores solucionados y funcionalidades insignificantes. Se puede mencionar, pero no exhibir, ya que generalmente es demasiado lento y distrae de las historias más importantes.

Al final de la reunión, se realiza una encuesta individual a todos los participantes para recopilar impresiones, recomendaciones para cambios y mejoras, así como su relevancia. Y el propietario del producto discute con los asistentes y el equipo las modificaciones potenciales en el backlog del producto.

#### **2.7.2.5 Retrospectiva del Sprint (fase 5).**

Lo más importante de una retrospectiva de Sprint es asegurarse de que tienen lugar. Por alguna razón, los equipos no siempre parecen inclinados a hacer retrospectivas. Sin un empujón, la mayoría de los equipos usualmente se saltarían la retrospectiva y continuarían

con el próximo Sprint. Pero, todo el mundo coincide en que las retrospectivas son extremadamente útiles. De hecho, se puede afirmar que la retrospectiva es el segundo evento más importante de Scrum (siendo el primero la reunión de planificación de Sprint), ya que es la mejor oportunidad para mejorar.

Por supuesto, una reunión de retrospectiva no es necesaria para obtener buenas ideas; sin embargo, si una idea surge durante la retrospectiva y viene del equipo, entonces todos pueden participar y discutirla. Será evidente que el equipo sigue cometiendo los mismos errores una y otra vez sin retrospectivas. El formato general para las retrospectivas varía un poco, pero generalmente se hace algo como esto:

- Se reserva 1 - 3 horas, dependiendo de cuánto análisis se tenga.
- Participantes: el Dueño de Sistema, el Equipo Scrum y el Scrum Master.
- Se conducen a una reunión cerrada, que se pueda tener una discusión sin interrupciones.
- Normalmente no se hace retrospectivas en la sala del equipo, ya que la atención de la gente suele diluirse.
- Alguien es designado secretario.
- El Scrum Master muestra la pila de Sprint y, con ayuda del equipo, resume el Sprint. Eventos importantes, decisiones, etc.
- Se realiza "la ronda". Cada persona tiene la oportunidad de expresar libremente sus opiniones sobre su desempeño, posibles mejoras y cómo creen que deberían actuar de manera diferente en el próximo Sprint.
- Se observa la velocidad estimada frente a la real. Si hay una gran diferencia, intentamos analizar por qué.
- Cuando el tiempo casi se ha acabado, el Scrum Master trata de resumir las sugerencias concretas sobre qué puede hacerse mejor en el próximo Sprint.

Las retrospectivas generalmente no están muy estructuradas. No obstante, el tema subyacente es siempre el mismo: “qué se puede realizar mejor el próximo Sprint”.

## 2.8 MÉTRICAS DE CALIDAD DEL SOFTWARE

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

### 2.8.1 *Factores de Calidad ISO 9126*

Este estándar según (Pressman, 2002, p. 343) se creó para identificar los aspectos clave de calidad del software. Define seis atributos esenciales de calidad:

- **Funcionalidad:** Grado en que el software cumple con necesidades específicas, como adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- **Confiabilidad:** Tiempo durante el cual el software está disponible para su uso, relacionado con madurez, tolerancia a fallos y capacidad de recuperación.
- **Usabilidad:** Facilidad de uso del software, evaluada por su comprensibilidad, facilidad de aprendizaje y operabilidad.
- **Eficiencia:** Uso óptimo de los recursos del sistema, medido por el rendimiento en términos de tiempo y recursos.
- **Mantenibilidad:** Facilidad de realizar reparaciones en el software, relacionada con su analizabilidad, capacidad de cambio, estabilidad y facilidad de prueba.
- **Portabilidad:** Facilidad de transferir el software entre entornos, caracterizada por su adaptabilidad, capacidad de instalación, conformidad y sustituibilidad.

Estos atributos no siempre se pueden medir directamente, pero ofrecen una base para mediciones indirectas y una lista de verificación útil para evaluar la calidad del sistema.

**Tabla 2.**  
**Características del ISO 9126**

ISO 9126		ISO/IEC 25012
CARACTERISTICA	SUBCARACTERISTICA	CARACTERISTICA
FUNCIONALIDAD	Idoneidad	Consistencia Actualidad Compleitud Precisión
	Exactitud	Exactitud
	Interoperabilidad	
	Seguridad	Seguridad
FIABILIDAD	Madurez	Disponibilidad
	Tolerancia a fallos	Recuperabilidad
USABILIDAD	Facilidad de comprensión	Recuperabilidad
EFICIENCIA	Operatividad	Accesibilidad Manejabilidad
MANTENIBILIDAD	Facilidad de análisis	
	Facilidad de cambio	Facilidad de cambio
	Estabilidad	
PORTABILIDAD	Facilidad de prueba	
	Facilidad de instalación	
	Facilidad de ajuste Facilidad de adaptación al cambio	Portabilidad

Nota. Evaluación de Software ardilla, 2013, actividad reconocimiento

**Figura 9.**  
**Atributos de Calidad del Estándar ISO/IEC. 9126**



Nota. Atributos de calidad del estándar ISO 9126, Cervantes Maceda et al., 2016, CENGAGE Learning

### 2.8.1.1 Funcionalidad.

La funcionalidad evalúa la capacidad del software para cumplir con los requisitos del usuario final y las normas de protección de datos. Para calcular la funcionalidad se deben calcular los valores de las cinco características, que son el número de entradas, el número de salidas, las solicitudes realizadas por el usuario, los archivos y las interfaces de usuario. Calculando el punto de función con la siguiente relación:

**Tabla 3.**

**Tabla de Referencia Para el Cálculo de la Funcionalidad**

Variable	Descripción
PF	Producto de la funcionalidad
ct	Suma de todas las entradas cuenta total
0.65	Valor de ajuste de complejidad mínimo
0.01	Factor de conversión con error del 1%
Fi	Valores de complejidad del sistema

*Nota.* Cálculo de funcionalidad, Calle, 2020

Ecuación para el cálculo de la métrica de la Funcionalidad es la siguiente:

**( 1 ). Ecuación para el cálculo de la métrica de funcionalidad**

$$PF = ct * [0.65 + 0.01 * \sum(Fi)] \quad (1)$$

Ecuación para calcular la funcionalidad del sistema:

**( 2 ). Ecuación Funcionalidad del Sistema**

$$Funcionalidad = \frac{PF}{PF_{max}} \quad (2)$$

**Figura 10.****Gráfico de Características de Funcionalidad**

*Nota.* Evaluación de Software, ardilla, 2013, actividad reconocimiento

**Tabla 4.****Traducción de la Complejidad Para Puntos de Función**

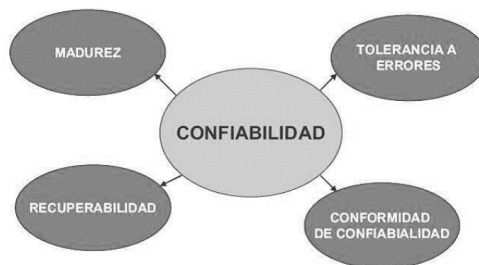
Complejidad	Factor		
	Simple	Medio	Complejo
Entradas del Usuario	x3	x4	x6
Salidas del usuario	x4	x5	x7
Peticiones del usuario	x3	x4	x6
Archivos	x7	x10	x15
Interfaces externas	x5	x7	x10

*Nota.* Valores Estándar de la IFPUG, Busquelle, 2010, Function Point User Group

### **2.8.1.2 Confiabilidad.**

En la confiabilidad se evalúa la capacidad de recuperación del software ante fallas o errores que puedan presentarse. La confiabilidad contiene diferentes propiedades como ser la disponibilidad, fiabilidad protección y la seguridad.

Figura 11.

**Gráfico de Características de Confiabilidad**

Nota. Evaluación de Software ardilla, 2013, actividad reconocimiento

Para el cálculo de la confiabilidad en un software se puede aplicar las siguientes referencias:

**Tabla 5.****Tabla Para el Cálculo de la Confiabilidad**

Variable	Descripción
PF	Punto de Fusión
F(t)	Probabilidad de la ejecución con fallas
1-F(t)	Probabilidad de ejecución sin fallas
f	Funcionalidad del sistema
<i>u</i>	Probabilidad de falla en un periodo de tiempo
t	Tiempo total en el que se realiza el cálculo del fallo

Nota. Calculo de la portabilidad, Calle, 2020.

La ecuación para el cálculo de la portabilidad es la siguiente según a la tabla que esta definida:

**( 3 ). Ecuación para la Calidad de Portabilidad**

$$P (T \geq t) = 1 - F(t) \quad (3)$$

**( 4 ). Ecuación Funcionalidad del Sistema**

$$\text{Funcionalidad} = \frac{PF}{PF_{max}} \quad (4)$$



Para hallar la probabilidad de falla en el sistema  $F(t)$  en el periodo de tiempo  $t$ , se utiliza la función exponencial definida por:

**( 5 ). Función Exponencial**

$$F(t) = f * e^{(-u*t)} \quad (5)$$

**2.8.1.3 Usabilidad.**

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido, mirándolo gráficamente quedaría así: (Norma ISO 9000, 2018).

La siguiente ecuación se determina la usabilidad de un sistema:

**( 6 ). Ecuación de la Usabilidad**

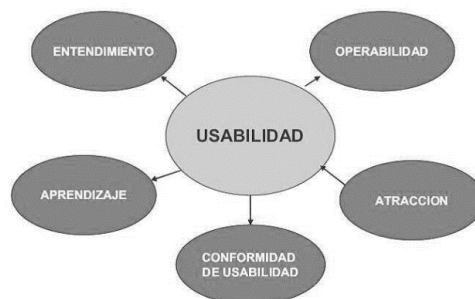
$$FU = [(\sum \frac{Xi}{n}) * 100] \quad (6)$$

**Tabla 6.**

**Tabla Referencial para el Cálculo de la Usabilidad**

Variable	Descripción
$X_i$ :	Es la sumatoria de Valores.
$n$	Indica el número de preguntas Las preguntas contienen una escala de valor.
FU	Facilidad de uso

*Nota.* Calculo de la usabilidad, Claudio, 2020, Medium

**Figura 12.****Gráfico de Características de Usabilidad**

*Nota.* Evaluación de Software ardilla, 2013, Actividad reconocimiento

**2.8.1.4 Mantenibilidad.**

La mantenibilidad es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales. En la siguiente figura se observa las características de Mantenimiento: (Norma ISO 9000, 2018).

Para hallar el índice de madurez del software se calcula con la siguiente formula:

**( 7 ). Ecuación de Madurez**

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt} \quad (7)$$

**Tabla 7.****Tabla de Referencia del Mantenimiento**

<b>Variable</b>	<b>Descripción</b>
IMS	Índice de mantenimiento del sistema
Mt	Número de módulos en la versión actual
Fa	Número de módulos en la versión actual que se han cambiado.
Fb	Número de módulos en la versión actual que se han añadido.
Fc	Número de módulos que se modificaron.

*Nota.* Métrica del mantenimiento del software, Calle, 2020

**Figura 13.****Gráfico de Características de Mantenimiento**

Nota. Evaluación de Software ardilla, 2013, Actividad reconocimiento

**2.8.1.5 Portabilidad.**

La "portabilidad del software" implica su capacidad de moverse entre entornos, y se detalla en la figura de las características de mantenimiento. (Norma ISO 9000, 2018).

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

**( 8 ). Ecuación de Grado de Portabilidad**

$$GP = 1 - \frac{ET}{ER} \quad (8)$$

**Tabla 8****Tabla de Referencia Para la Portabilidad**

Variable	Descripción
GP	Grado de portabilidad
ET	Costo de transporte del sistema
ER	Costo de Redesarrollo
GP>0	La portabilidad del software es las rentable que el desarrollo
GP=1	La portabilidad del sistema es optima
GP<0	El redesarrollo es más rentable que la portabilidad.

Nota. Métrica de la portabilidad, Calle, 2020

**Figura 14.****Gráfico de Características de Portabilidad**

*Nota.* Evaluación de Software ardilla, 2013, actividad reconocimiento

### 2.8.1.6 Definición de Criterios Elementales y Procedimientos de Medición.

En este punto, se establecerá una sólida base de criterios esenciales para la medición y el fundamento para llevar a cabo la medición y calificación de los valores, con el objetivo de evaluar la calidad del software. En este contexto, se definirán los siguientes criterios:

**Tabla 9.****Tabla de ponderación en porcentajes**

Características de calidad	Mala	Regular	Buena	Muy bueno
Funcionalidad	0-30%	31-50 %	51-90 %	91-100 %
Confiabilidad	0-30%	31-50 %	51-90 %	91-100 %
Usabilidad	0-30%	31-50 %	51-90 %	91-100 %
Mantenibilidad	0-30%	31-50 %	51-90 %	91-100 %
Portabilidad	0-30%	31-50 %	51-90 %	91-100 %
<b>Total</b>	0-30%	31-50 %	51-90 %	91-100 %

*Nota.* Ponderación de porcentaje, Castillo, 2020, Sistema De Educacion Virtual (E-Learning)

## 2.9 SEGURIDAD DEL SOFTWARE

### 2.9.1 Principios de Seguridad

La seguridad del software es una actividad del aseguramiento del software que se centra en la identificación y evaluación de los peligros potenciales que podrían afectarlo negativamente y que podrían ocasionar que falle todo el sistema. Si los peligros se identifican

al principio del proceso del software, las características de su diseño se especifican de modo que los eliminen o controlen. La seguridad en informática es la capacidad de ofrecer confidencialidad, integridad, disponibilidad y control de la información conocida solo por los individuos autorizados (Pressman, 2010, p. 378)

Una de las características más importantes que debe tener un software para sistemas informáticos es la seguridad, ya que dependerá de que usuarios no autorizados puedan acceder a la información y garantizar que el sistema funcione correctamente.

Para Sommerville (2011), "El número de capas de protección que se necesitan en cualquier aplicación particular depende del carácter crítico de los datos. No todas las aplicaciones requieren protección a nivel registro". Por lo tanto, en un sistema no debería usarse las mismas credenciales de acceso para cada nivel de permisos, el método más adecuado es utilizar un sistema basado en el uso de contraseñas.

En la Seguridad de software existen cuatro funciones fundamentales para ofrecer una adecuada protección de datos los cuales son:

- **Confidencialidad:** La confidencialidad busca garantizar que solo personas autorizadas tengan acceso a la información.
- **Integridad:** La integridad busca garantizar que la información y los datos que proporciona el sistema no se dañen, modifiquen o sufran pérdidas.
- **Disponibilidad:** la disponibilidad busca asegurar que el sistema esté disponible cuando se le solicite, procurando evitar fallos, interrupciones y caídas no programadas.
- **Autenticación:** la autenticación busca garantizar la verificación de la identidad de los usuarios antes de que accedan a la información a la que el usuario tiene permitido el acceso.

## 2.10 ESTIMACIÓN DE COSTOS

### 2.10.1 Modelo COCOMO

Es un modelo matemático de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Modelo Constructivo de Coste (Constructive Cost Model), siendo uno de los modelos de estimación de costo de software más utilizado y estudiado en la industria (Bustamante, 2021).

Existe distintos modelos de COCOMO:

- **El básico:** Generalmente aplicados a proyectos pequeños denominados proyectos orgánicos, los cuales no superen las 50.000 líneas de código.
- **El intermedio:** Enfocado en proyectos medianos denominados semiacoplados, en los cuales la cantidad de líneas de código sean menores a 300.000.
- **El avanzado:** Utilizado para proyectos con una complejidad alta denominados Incrustados, en los cuales se cuenta con mucha experiencia.

Para un cálculo más acertado también se utilizan distintos coeficientes dependiendo del modelo COCOMO a utilizar, a continuación, se muestra en la imagen los coeficientes definidos para el cálculo del costo.

**Tabla 10**

**Tabla de Coeficientes COCOMO**

PRODUCTO SOFTWARE	A	B	C	D	DESCRIPCIÓN
Simple	3,2	1,05	2,5	0,38	Aplicaciones bien comprendidas desarrolladas por equipos pequeños
Moderada	3,0	1,12	2,5	0,35	Proyectos más complejos donde los miembros del equipo tienen experiencia en sistemas relacionados
Avanzada	2,8	1,20	2,5	0,32	Proyectos complejos donde el software es parte de un complejo fuertemente acoplado de hardware, software, reglas y procedimientos operacionales.

*Nota.* Coeficiente de COCOMO, Arroyo, 2010, Slideshare

### 2.10.1.1 COCOMO Básico

En relación al desarrollo del sistema, considerando el uso de frameworks y librerías que promueven la reutilización de código, se utilizará el método COCOMO básico. Este método se basa en una única variable (líneas de código), permitiendo una evaluación centrada en el esfuerzo requerido.

El modelo COCOMO ofrece varias fórmulas que se aplican en diferentes etapas del desarrollo de software. Estas ecuaciones están diseñadas para estimar aproximadamente el esfuerzo requerido, el costo del software y el tiempo estimado de desarrollo.

### 2.10.1.2 Esfuerzo.

Para calcular el esfuerzo de un software, se emplearán los datos presentados en las siguientes tablas:

**Tabla 11**

**Tabla de Referencia Para Cálculo del Esfuerzo**

Variable	Descripción
E	Estimación del esfuerzo en (meses–hombre)
a, b, c, d	Coefficientes obtenidos mediante estudios en proyectos
KLDC	Número de líneas de código
FAE	Factor de esfuerzo
p	personas
m	mes

*Nota.* Cálculo del esfuerzo, Arroyo, 2010, Slideshare

Para el cálculo del **Esfuerzo** utiliza la siguiente formula:

**( 9 ). Ecuación de Esfuerzo**

$$E = a * KLDC^b * FAE = [p/m] \quad (9)$$

### 2.10.1.3 Tiempo de Desarrollo.

La fórmula para utilizar para el tiempo de desarrollo es:

$$TDES = b * E^d$$

**Tabla 12**

**Tabla para el Cálculo del Tiempo de Desarrollo de Software**

Variable	Descripción
TDES	Tiempo de duración de desarrollo (meses)
E	Esfuerzo
a, b, c, d	Coefficientes obtenidos mediante estudios en proyectos

*Nota.* Desarrollo del software, Arroyo, (2010), Slideshare

### 2.10.1.4 Personal Necesario Para el Desarrollo.

Para el desarrollo de un software según el modelo COCOMO está dada con la siguiente ecuación:

*( 10 ). Ecuación Cantidad de Personal*

$$CP = \frac{E}{TDES} \quad (10)$$

Para calcular la cantidad de personal necesario para el desarrollo del software de hará uso de los parámetros:

**Tabla 13**

**Tabla de Referencia para el Cálculo del Personal Necesario**

Variable	Descripción
CP	Personal necesario para el desarrollo del proyecto (personas)
E	Esfuerzo
TDES	Tiempo de duración de desarrollo (meses)

*Nota.* Cálculo del personal necesario, Arroyo, 2010, Slideshare



## 2.11 PRUEBAS DE FUNCIONAMIENTO

### 2.11.1 Pruebas de Caja Negra

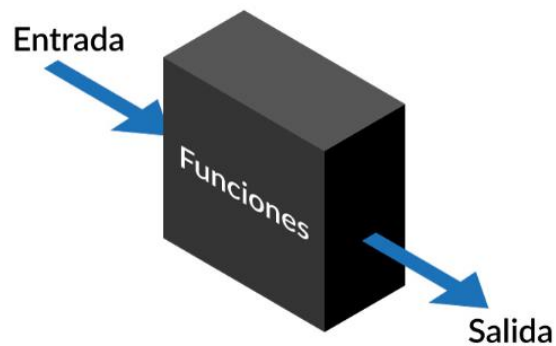
Las pruebas de caja negra, también conocidas como pruebas de comportamiento, verifican que el software cumple con los especificaciones funcionales. Esto significa que permiten que la entrada de software sea procesada a través de conjuntos de condiciones que satisfacen por completo todos los requisitos funcionales de un programa. (Pressman, 2010).

Las pruebas de caja negra intentan encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de estructura de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de iniciación y determinación.

**Figura 15.**

**Métodos de Prueba de Caja Negra**



*Nota.* Caja negra para pruebas Pressman, 2010, University of Connecticut

- **Las tablas de decisión** están compuestas de condiciones que darán como resultado verdadero o falso en base a las entradas de los datos.

**Figura 16.****Tabla de Decisión**

<b>Conditions</b>	<b>TC1</b>	<b>TC2</b>	<b>TC3</b>	<b>TC4</b>
Request login	0	1	1	1
Valid user name entered	X	0	1	1
Valid password entered	X	X	0	1
<b>Actions</b>				
Offer recovery credentials	0	1	1	0
Activate entrybox user name	0	1	1	0
Activate entrybox password	0	0	1	0
Enter privileged area	0	0	0	1

*Nota.* Decisión para la verificación de las salidas o resultados, Dano, 2016, Academia EDU

### **2.11.2 Pruebas de Caja Blanca**

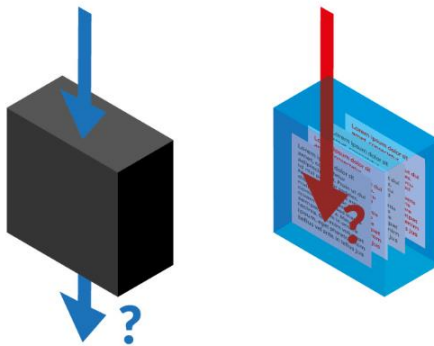
La prueba de caja blanca, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que se ejercita por los menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los bucles en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez (Pressman, 2011)

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba como:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

**Figura 17.**

**Métodos de Prueba de Caja Blanca**



*Nota.* Caja blanca para pruebas, Pressman, 2010, University of Connecticut

### **2.11.3 Pruebas de Estrés**

La prueba de rendimiento es una técnica de prueba de software no funcional que determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación se mantiene bajo una determinada carga de trabajo. Es un paso clave para asegurar la calidad del software, pero desafortunadamente a menudo se ve como una reflexión posterior en aislamiento y para comenzar una vez que se completan las pruebas funcionales, y en la mayoría de los casos después de que el código está listo para ser distribuido (Micro Focus, 2022).

**Figura 18.**

**Gráfico de Prueba de Estrés**



*Nota.* Pruebas de estrés para el sistema app

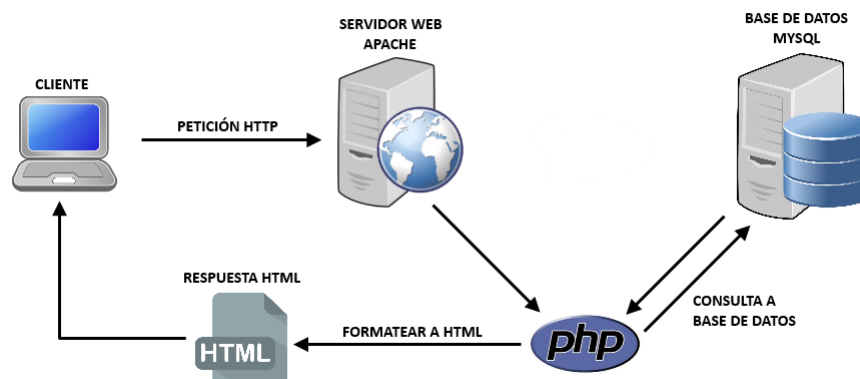
## **2.12 HERRAMIENTAS DE DESARROLLO**

### **2.12.1 Definición de Servidor Web**

Un servidor web como su nombre lo indica, es un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización. Existen varios tipos de servidores web, Apache es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows. Al instalarlo en nuestra PC dispondremos de un servidor completo, con todos los requisitos para ejecutarlo de forma local. (OsGroup, 2019)

La herramienta tecnológica a utilizar será el PHP, cuyas siglas responden a un acrónimo recursivo (PHP: hypertext preprocessor) que es una herramienta orientada a objetos que nos proporciona un nivel extremadamente alto de velocidad y rendimiento. (Guioteca, 2011)

**Figura 19.**  
**Servidor Web Apache**



*Nota.* Definición de servidor web Fuente: OsGroup, 2019, O.S. Group

### 2.12.2 Gestor de Base de Datos

Los sistemas de gestión de bases de datos (SGBD) según (Espinoza, 2016) son un “tipo de software específico dedicado a servir de interfaz entre las bases de datos, el usuario y las aplicaciones que la utilizan”. El objetivo principal de la base de datos es el de unificar los datos que se manejan y los programas o aplicaciones que lo manejan. Anteriormente los programas se codificaban junto con la base de datos, es decir, se diseñaban para la aplicación concreta que lo iba a manejar, lo desembocaba en una dependencia de los programas respecto a los datos, ya que la estructura de los ficheros va incluida dentro del programa, cualquier cambio en la estructura en el fichero provocaba modificar y recompilar programas. Además, que cada aplicación utiliza ficheros que pueden ser comunes a otros de la misma organización, por la que se produce una redundancia de la información, que provoca mayor ocupación de memoria, laboriosos programas de actualización (unificar datos recogidos por

las aplicaciones de los diferentes departamentos), e inconsistencia de datos (no son correctos) si los datos no fueron bien actualizados en todos los programas.

### **2.12.3 Gestor de Base de Datos MariaDB**

Es un sistema gestor de bases de datos (SGBD), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe.

El sistema de base de datos que proviene de MySQL, pero con licencia GPL, desarrollado por Michael Widenius, fundador de MySQL y la comunidad de desarrolladores de software libre.

Como hemos comentado anteriormente, MariaDB se trata de un “sustituto” de MySQL que corre bajo una licencia GPL. Por lo tanto, incorpora todas las mejoras de la versión de MySQL en la que se encuentre y además incluye diferentes mejoras:

Lo primero que tiene que quedar totalmente claro es que al ser MariaDB compatible con MySQL, la migración a MariaDB es simple y directa, no hay que adaptar el código ni nada.

#### **Ventajas de MariaDB.**

(ZEOKAT, 2013), nuevos motores de almacenamiento, para la mayoría de usuarios lo interesante es, que viene a reemplazar a MyISAM y también tenemos XtraDB que reemplaza a InnoDB. Los nuevos motores de almacenamiento son:

- Aria: Un motor de almacenamiento a prueba de fallos basado en MyISAM.
- XtraDB: El reemplazo del motor InnoDB basado en el plug-in de InnoDB.
- PBXT: Un motor de almacenamiento transaccional con una gran cantidad de nuevas y bonitas características.
- FederatedX: El reemplazo del motor Federated.

Mejoras de velocidad sobre todo en consultas complejas cuando se usa el motor de almacenamiento Aria, ya que Aria cachea los datos de tablas temporales en memoria, lo que supone un rendimiento frente al uso del disco duro (que es lo que emplea MyISAM).

Se añaden nuevas tablas de sistema (INFORMATION\_SCHEMA) para almacenar estadísticas que nos pueden ayudar a optimizar las bases de datos.

El sistema para manejar las conexiones se ha mejorado, ya que implementa el sistema pool-of-threads de MySQL 6.0 con el que podemos tener más de 200.000 conexiones a MariaDB.

En general se han hecho muchas modificaciones para mejorar el rendimiento, velocidad e incluso implementar características nuevas. Desventajas de MariaDB. La única desventaja es la propia de tener que desinstalar MySQL e instalar MariaDB, pero la migración es muy sencilla

#### **Desventajas de MariaDB.**

La única desventaja es la propia de tener que desinstalar MySQL e instalar MariaDB, pero la migración es muy sencilla (ZEOKAT, 2013)

#### **2.12.4 Lenguaje de Programación PHP**

PHP son las siglas en inglés de “Hypertext Pre-Processor” que al traducirlo al español pierde un poco el sentido, mejor lo analizamos y encontramos que significa “Lenguaje de Programación Interpretado”. Este lenguaje es al que le debemos la visualización de contenido dinámico en las páginas web.

Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos. Un lenguaje del lado del servidor es aquel que

se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones 50 en red, y otras tareas para crear la página final que verá el cliente.

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. (Alvarez, 2017)

#### **Ventajas:**

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.

Según (Welling, 2005), esto hace que la programación en PHP sea segura y confiable. Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con varios gestores de base de datos. ∞ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).

#### **Desventajas:**

Como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código, pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución.

#### **2.12.5 Framework Codeigniter**

Codeigniter es un framework para el desarrollo de aplicaciones en PHP, que utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar,



además de dar una mayor velocidad a la hora de crear páginas Webs. Algunas ventajas de usar el framework CodeIgniter:

- Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- Es sencillo de instalar, basta con subir los archivos al ftp y tocar un archivo de configuración para definir el acceso a la base de datos.
- Existe abundante documentación en la red.
- Facilidad de edición del código ya creado.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.
- Estandarización del código.
- Separación de la lógica y arquitectura de la web, el MVC.
- Cualquier servidor que soporte PHP + MySQL sirve para CodeIgniter.
- CodeIgniter se encuentra bajo una licencia open source, es código libre.

#### **2.12.5.1 Definición de MVC.**

Según (Burbeck, 1993), es una capa de abstracción que permite realizar aplicaciones más potentes de una manera más rápida y utilizable las funciones, sin tener que saber cómo se ha programado.

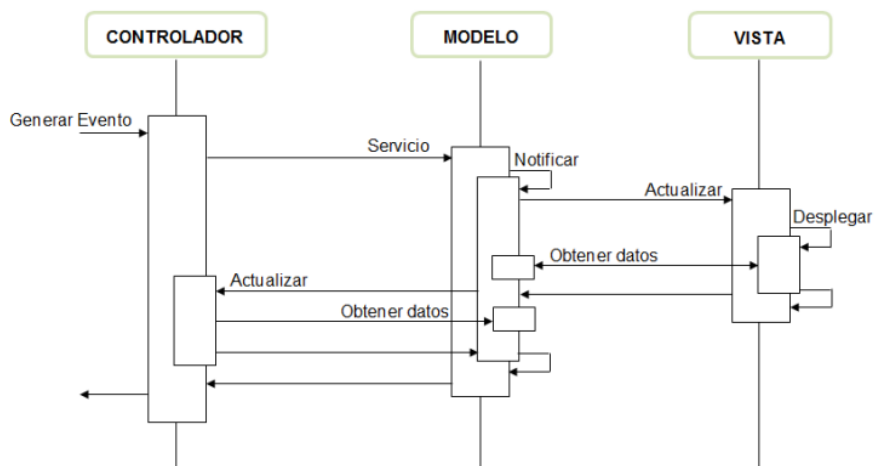
Un framework MVC es una forma estandarizada de dividir la información en carpetas en un proyecto. MVC es una guía para el diseño de arquitecturas de aplicaciones que ofrecen una fuerte interactividad con usuarios.

Según (Rouse, 2016) este patrón “organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es el conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesan las peticiones de los

usuarios y controla el flujo de ejecución del sistema”. Una gran mayoría de los frameworks implementan este patrón.

**Figura 20.**

**Diagrama de Secuencias de MVC**



*Nota.* Diagrama de MVC, Rouse, 2016, Universidad Pontificia de Salamanca

**Modelo:** Según (Burbeck, 1993), es la “parte del código que se encarga de la gestión de los datos, implementada en para una gran cantidad de frameworks, existe una carpeta llamada Model. Es un conjunto de código que hace referencia, a una base de datos, o a una conexión con JSON, XML, otros”.

Gracias a la separación de las dependencias al modelo del núcleo del programa, en el futuro se puede cambiar de base de datos ejemplo de MySQL a Oracle, solo tiene que cambiarse el objeto o las funciones que se encuentran en un modelo, no afectando a la aplicación principal. El modelo representa la estructura de datos; es decir toda la información con la que opera la aplicación; gestiona el comportamiento y los datos de dominio; típicamente

las clases del modelo contendrán las funciones que ayudan a devolver, insertar, actualizar y eliminar información de la base de datos, independientemente de la base de datos a utilizar.

Encapsular el modelo de una aplicación en componentes facilita la depuración, mejora la calidad y favorece la reutilización del código, a los cambios que el sistema pudiera sufrir a futuro.

**Vista:** Dependiendo de la aplicación si esta por carpetas, según (Burbeck, 1993) se encarga de todo lo que tenga que ver con el HTML, es decir, es la estructura de la Web. Gracias a la separación en una carpeta diferente se puede administrar con facilidad la información que se presenta al usuario.

La vista gestiona la presentación de la información de la aplicación; todo lo relativo a la interfaz de usuario, los datos necesarios para que pueda seguir interactuando con la aplicación.

Una vista normalmente es una página web, que representa la lógica de presentación de una aplicación. Los componentes de la misma extraen el estado actual del sistema del modelo y proporcionan la interfaz de usuario para el protocolo que se utiliza. Una vista puede ser un fragmento de página (encabezado, pie de página, cuerpo), así como una página RSS, o cualquier otro tipo "pagina".

**Controlador:** Según (Burbeck, 1993) un controlador es el núcleo del programa, aplicación o framework, es donde realmente se ubica toda la lógica de la programación para el perfecto funcionamiento de un sistema. Un controlador puede cargar cualquier Modelo y

cualquier cantidad de vista(s) si son requeridas. Encargado de generar contenido HTML, con datos que obtiene dinámicamente.

Es el encargado de responder a los eventos invocados desde la vista; y as u vez llamar a los métodos correspondientes, el modelo para procesar y producir las respuestas necesarias con los datos adecuados para la interacción de la aplicación.

Sirve como intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web. El controlador es responsable de recibir los eventos, determinar el procesador del evento, invocar al procesador y finalmente provocar la generación de la vista apropiada. Tareas que debe de realizar un controlador:

- Control de seguridad
- Identificación de eventos
- Preparar el modelo
- Procesar el evento
- Manejar los errores
- Provocar la generación de la respuesta.

#### **2.12.6 Definición de JQuery**

Se denomina jQuery a una librería o biblioteca de JavaScript que facilita la programación en este lenguaje. Por lo general se emplea para añadir elementos interactivos a una página web sin necesidad de tener que programar demasiado. En el caso de jQuery, la biblioteca es de código abierto (todos pueden contribuir al desarrollo ya que el acceso al código fuente es libre) y está basada en el lenguaje JavaScript. Lo que hace es ofrecer diversas

funcionalidades que, de otro modo, exigirían desarrollar más código, por lo tanto, ayudan a ahorrar tiempo. Según: (Definiciones, 2019).

### **2.12.7 Definición de JavaScript**

JavaScript consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones en el lado del cliente, estando embebido en el código fuente de la página web.

Técnicamente, constituye un dialecto del estándar ECMAScript, propuesto por la entidad internacional de estándares de información y comunicación ECMA International y diseñado inicialmente por Netscape y, posteriormente, por la Fundación Mozilla.

También constituye un estándar ISO. Si bien recibe en su sintaxis y forma de trabajo una mayor herencia del lenguaje C, también adopta nombres y convenciones que encontramos en Java, aunque se quedan en eso, en parecidos “cosméticos”. JavaScript permite, en una página web, crear elementos como cuadros de diálogo, recoger información entrada por el usuario y mandarla al servidor para ser procesada

### **2.12.8 Definición de AJAX**

AJAX son las siglas de Asynchronous JavaScript And XML, (Javascript asíncrono y XML). No es en sí un lenguaje de programación, sino una nueva técnica que combina varios lenguajes de programación.

La ventaja de ajax respecto a otros lenguajes de programación web es la asincronía. Esto consiste en que cuando queremos intercambiar datos con el servidor (por ejemplo, enviar

o comprobar un formulario, consultar una base de datos, etc), la página no se queda parada esperando la respuesta, sino que se pueden seguir ejecutando acciones mientras tanto.

Con ajax podemos crear páginas interactivas. En éstas solicitamos datos al servidor, los cuales podemos tener guardados en otras páginas o en bases de datos. El servidor devuelve los datos, los cuales se cargan en la misma página y en segundo plano.

Lo de "segundo plano" significa que mientras esperamos que se reciban los datos la página no se queda parada, y el usuario o la programación de la página pueden seguir haciendo otras

cosas. Para poder entender este manual debemos tener conocimientos de HTML, CSS, y Javascript. Es conveniente también tener nociones de XML y acceso al DOM. Para las últimas páginas necesitamos también tener conocimientos de PHP, ya que es el lenguaje que se emplea para procesar los datos en el servidor. (Definiciones, 2019).

### **2.12.9 *Bootstrap***

Bootstrap es conjunto conceptos, prácticas y criterios (framework) desarrollado por Mark Otto y Jacob Thornton dentro de Twitter con la intención de estandarizar el conjunto de herramientas que utilizaban todos los involucrados en el desarrollo del front-end.

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo.

Esta técnica de diseño y desarrollo se conoce como "responsive design" o diseño adaptativo. De esta manera crearon un conjunto de librerías JavaScript y CSS que toda la compañía debía usar evitando que las partes desarrolladas por un equipo no pudiesen ser mantenidas por otros. Bootstrap nos ayuda a maquetar un sitio web con rapidez y, sobre todo,

ayudándonos a que el diseño sea correcto y usable tanto en dispositivos convencionales con en los táctiles (responsive web design).

Para hacerlo, nos ofrece una serie de estilos CSS y librerías JavaScript que nos ayudarán de una manera rápida a desarrollar nuestro sitio web y sobre todo es recomendable para el desarrollo de prototipos y tener un tiempo de respuesta realmente bueno. (Benjamin Jakobus, 2018).

#### **2.12.10      *Definición de HTML5***

Según (MariaDB, 2019) HTML5 (HyperText Markup Language) es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML5 podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5.

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5. Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta página

referencia numerosos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función.

- **Semántica:** Permite describir con mayor precisión cuál es su contenido.
- **Conectividad:** Permite comunicarse con el servidor de formas nuevas e innovadoras.
- **Sin conexión y almacenamiento:** Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.
- **Multimedia:** Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- **Gráficos y efectos 2D/3D:** Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- **Rendimiento e Integración:** Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- **Acceso al dispositivo:** Proporciona para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.
- **CSS3:** Nos ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados

### **2.12.11      *Definición de CSS3***

CSS es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML.

CSS3 es una tecnología que ha tenido una evolución en el tiempo, que actualmente se encuentra en su versión 3, como su propio nombre indica.

Sus siglas corresponden a “Cascading Style Sheets”, que tiene el siguiente significado:



- Cascading, que significa que los estilos que aplicamos a los elementos de una página web se propagan a los elementos que contiene, se propagan en cascada.
- Style, porque mediante CSS lo que hacemos es aplicar estilos visuales a los distintos elementos de nuestra página web. Sheets, que significa hojas, porque los estilos de una página web se añaden en ficheros aparte, en ficheros con la extensión .css de manera general. CSS ha ido evolucionando desde hace 25 años a la versión actual, que es CSS3, pero su función es la misma, sirve para dar estilos visuales a las páginas web.

# CAPÍTULO III

### **3 MARCO APLICATIVO**

Este capítulo detalla la organización del sistema y los métodos de trabajo, utilizando una variedad de herramientas y metodologías para optimizar el desarrollo de sus módulos. Las fases del proceso se han organizado minuciosamente para maximizar las ventajas de cada enfoque.

Durante las fases iniciales, 1 y 2, nos guiamos por la metodología UWE para adquirir y priorizar los requisitos del sistema. Esto nos permite definir de manera precisa el backlog del producto y planificar las tareas a realizar en cada iteración.

Al avanzar a la fase 3, adoptamos Scrum como nuestra metodología principal. Aquí, aprovechamos las prácticas ágiles para gestionar el trabajo diario del equipo, priorizar el backlog del producto y realizar entregas incrementales durante los sprints.

En la fase 4, integramos tanto Scrum como UWE, especialmente durante la etapa de implementación. Esto nos permite aprovechar los modelos y diseños detallados de UWE como referencia para el desarrollo del sistema, mientras seguimos los principios ágiles de Scrum para gestionar el proceso de implementación de manera efectiva.

Finalmente, en la fase 5, nos sumergimos en la retrospectiva del Sprint, donde reflexionamos sobre nuestro trabajo, identificamos áreas de mejora y acordamos acciones concretas para implementar mejoras en los futuros sprints. Esta fase nos permite cerrar el ciclo de desarrollo de manera efectiva, asegurándonos de aprender y mejorar continuamente.

#### **3.1 ANÁLISIS DE REQUERIMIENTO**

Para el desarrollo de software en esta primera etapa usamos UWQ por la fácil y correcta obtención de requerimientos de la organización puede llegar a describir claramente

en forma consistente para el diseño del sistema web, por esta razón se toman en cuenta, para diferenciar los conceptos de requisitos.

### 3.1.1 *Recolección de Información*

Durante la recolección de información para su posterior análisis y definición de requerimientos se realizó diferentes actividades que se detallan a continuación:

<b>Actividad</b>	<b>Descripción</b>
Observación	Se pudo evidenciar que no existe un control exacto de los cursos que ofrecen y una falta de un sistema que agilice los procesos.
Entrevistas	Se realizaron entrevistas al personal de la UFAC tanto como escrita y verbal.
Documentación	Se pudo obtener documentación como: <ul style="list-style-type: none"> <li>• Documentos de los cursos.</li> <li>• Documentos de registro de cursos y participantes en formato Excel.</li> </ul>

### 3.1.2 *Especificación de Requerimientos*

En base a la información obtenida se pudo definir los siguientes requerimientos:

#### 3.1.2.1 **Requerimientos Funcionales.**

Los requerimientos funcionales son todas las características que necesita el sistema.

**Tabla 14.**

#### **Requerimientos Funcionales Parte de Administración**

<b>N.</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
R1-1	Ingreso al sistema	Acceder al sistema por tipos de usuario tales como: Administrador, Publicador, Docente, Participante.
R1-2	Administración de usuarios	Crear, Listar, Actualizar, eliminar usuarios.
R1-3	Administrar roles y permisos	Permite la asignación de roles a usuarios.
R1-4	Administración de los cursos	Crear, listar, actualizar y elimina. Los cursos deben estar asignados con una certificación.
R1-5	Administraciones participantes	Crear, listar, actualizar y eliminar. Los participantes deben estar asignados a una o más cursos.
//...		

		...//
R1-6	Administraciones docentes	Agrega docentes a un curso en específico, listar docentes, eliminar docente.
R1-7	Administrar la certificación	Agregar, Editar Actualizar, Eliminar, finalizar la configuración de la certificación.
R1-8	Registro de publicaciones de cursos en línea.	Editar visualizar los registros y detalles de los cursos finaliza.
R1-9	Registro de blogs	Agregar, editar, eliminar. Todos los usuarios registrados podrán postear un post.
R1-10	Administración de transmisiones en vivo	Agregar, visualizar la transmisión comentar y eliminar.
R1-11	Reportes	Generar reportes en diferentes niveles tanto en Excel y PDF reportes según el usuario requiera.

**Tabla 15.****Requisitos Funcionales Pública**

<b>N.</b>	<b>TIPO</b>	<b>OBSERVACIONES</b>
R2-1	Inicio de sesión del usuario	Acceder al sistema con el rol participante
R2-2	Perfil de usuario	Permite modificar sus datos de su cuenta
R2-3	Cambiar o actualizar contraseña	Permite modificar su contraseña de su cuenta
R2-4	Listado de los cursos de un participante	Listado de todos los cursos que tomo el participante con las notas incluidas.
R2-5	Registro de usuario a un curso	Ingresa sus datos para que sea registrado a un curso para su confirmación y posterior ingreso al sistema.
R2-6	Solicitud de Información	Usuario pide información sobre un curso.
R2-7	Post	Los usuarios registrados al sistema podrán comentar un post.
R2-8	Detalle de los cursos	Muestra toda la información que tiene cada curso.
R2-9	Transmisión en vivo de cursos	Permite interactuar con los usuarios en las transmisiones.

### 3.1.2.2 Requisitos No Funcionales.

Los requisitos no funcionales son aquellas que no forman parte de la funcionalidad en sí del sistema, pero son requisitos que forman parte general del sistema.

**Tabla 16**

**Requisitos no Funcionales**


ROL	FUNCION
R3-1	El sistema web debe funcionar correctamente en cualquier navegador web, como ser: Internet Explorer, Mozilla Firefox, Chrome y Otros.
R3-2	El sistema web debe estar implementada en un servidor web y base de datos, para estar disponible las 24 horas del día.
R3-3	Dar soporte y mantenimiento periódico del sistema web, para resguardar el buen funcionamiento del mismo.
R3-4	Almacenar copias de seguridad de manera sencilla de la base de datos.
R3-5	El sistema web debe contar con conexión a internet.


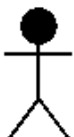


### 3.1.2.3 Definición de Actores.

En primera se identificará los actores que interactuarán con el sistema y posteriormente los roles y permisos de los mismos. En la siguiente tabla se muestran la lista de actores y a su lado una descripción de sus actividades relacionadas con el sistema web.

**Tabla 17.**

**Lista de Actores**

ACTOR	FUNCIONES
Administrador 	<ul style="list-style-type: none"> <li>• Acceso total en el sistema Web.</li> <li>• Registra a los nuevos participantes.</li> <li>• Confirma a los participantes.</li> <li>• Lista a todos los participantes</li> <li>• Asigna los roles y permisos de los usuarios.</li> <li>• Registra la información de un curso para su posterior publicación.</li> <li>• Registra docentes de cada curso.</li> <li>• Listar docentes de cada curso</li> <li>• Registra la publicación de un curso.</li> <li>• Verifica los blogs.</li> <li>• Configura los certificados.</li> <li>• Genera los certificados.</li> <li>• Administra las transmisiones en vivo.</li> </ul>

	<ul style="list-style-type: none"> <li>• Genera reportes.</li> <li>• Cambio de contraseña</li> <li>• Asigna usuarios y roles</li> <li>• Agrega contenido a los cursos</li> </ul>
<p>Publicador</p> 	<ul style="list-style-type: none"> <li>• Registra los participantes para cada curso.</li> <li>• Registra la información de un curso para su posterior publicación.</li> <li>• Realiza entrega de certificados.</li> <li>• Confirma preinscritos al curso.</li> <li>• Genera reportes</li> <li>• Agrega docentes a un curso.</li> <li>• Listar docentes de cada curso.</li> <li>• Genera certificados.</li> <li>• Administra la transmisión en vivo</li> <li>• Verifica los posts y postea material educativo.</li> <li>• Cambio de contraseña</li> <li>• Agregar contenido a los cursos</li> </ul>
 <p>Docente</p>	<ul style="list-style-type: none"> <li>• Registra su contenido de su curso para su posterior publicación.</li> <li>• Agrega un blog.</li> <li>• Verifica notas de sus participantes.</li> <li>• Cambio de contraseña</li> </ul>
<p>Participante</p> 	<ul style="list-style-type: none"> <li>• Verifica la cantidad de cursos tomados, información básica del curso.</li> <li>• Cambio de contraseña</li> <li>• Actualización de datos</li> <li>• Verificación de sus notas</li> <li>• Agrega un blog</li> <li>• Verificación de sus cursos.</li> </ul>
<p>Moodle</p> 	<ul style="list-style-type: none"> <li>• Tiene acceso a las notas de los participantes</li> <li>• Listado de participantes</li> <li>• Verificación de los cursos</li> <li>• Cambio de contraseña</li> </ul>

### 3.1.2.4 Descripción de Funciones.

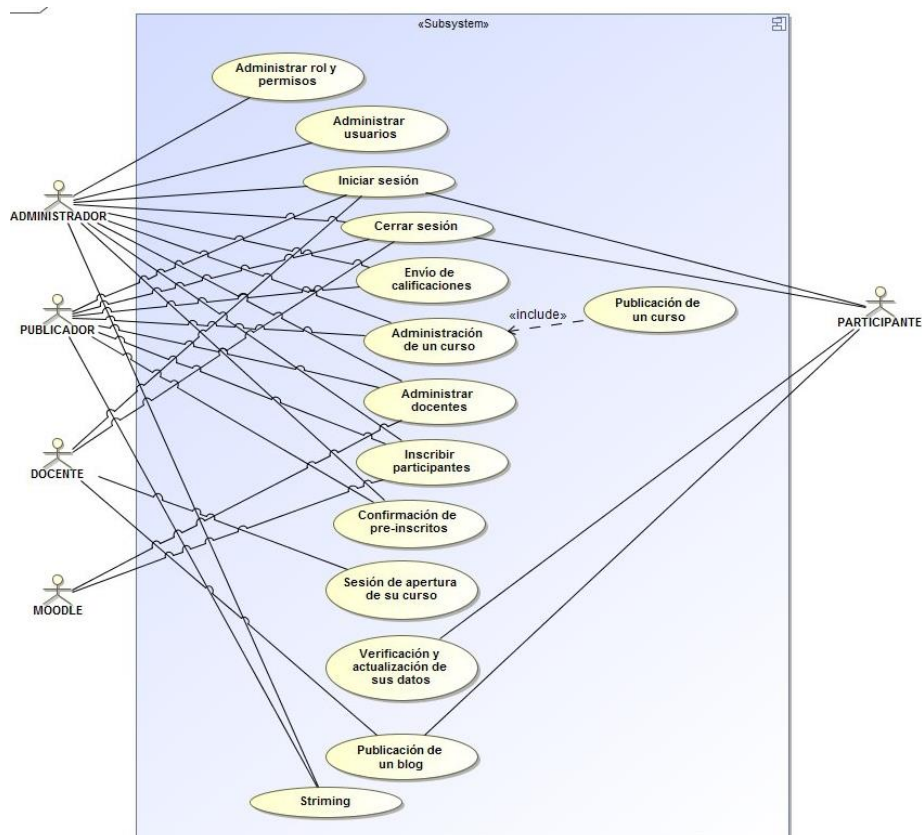
En las siguientes secciones, presentaremos un desglose detallado de las responsabilidades y funcionalidades asociadas a cada actor o usuario dentro de nuestra plataforma web.

**Tabla 18.****Descripción de Funciones**

ACTOR	FUNCIONES
Administrador	La función del administrador del sistema, es estar ligado directamente con el encargado de área, esta tiene todas las funciones del sistema
Publicador	Su función es registrar al sistema los pre inscritos, verificar si existen interesados para confirmarlos realiza los certificados, entregan certificados, inscriben nuevos participantes, asignan docentes a un curso, crean la información de los cursos, realizan transmisiones en vivo y sube blogs educativos y los verifica.
Docente	Las funciones del docente son tener acceso para un publicador de blog y la carga de sus videos introductorios y su contenido del curso un listado de sus cursos y la verificación de sus notas de sus estudiantes.
Participante	Visualizara sus notas y los cursos que tomó.
Moodle	La función de Moodle es dar acceso a su base de datos para ir haciendo una conexión con ella para generar o facilitar la administración.

**3.2 DISEÑO DEL SISTEMA****3.2.1 Diseño de Casos de Uso****3.2.1.1 Casos de Uso Administración.****Figura 21.****Caso de Uso Web de Administración**

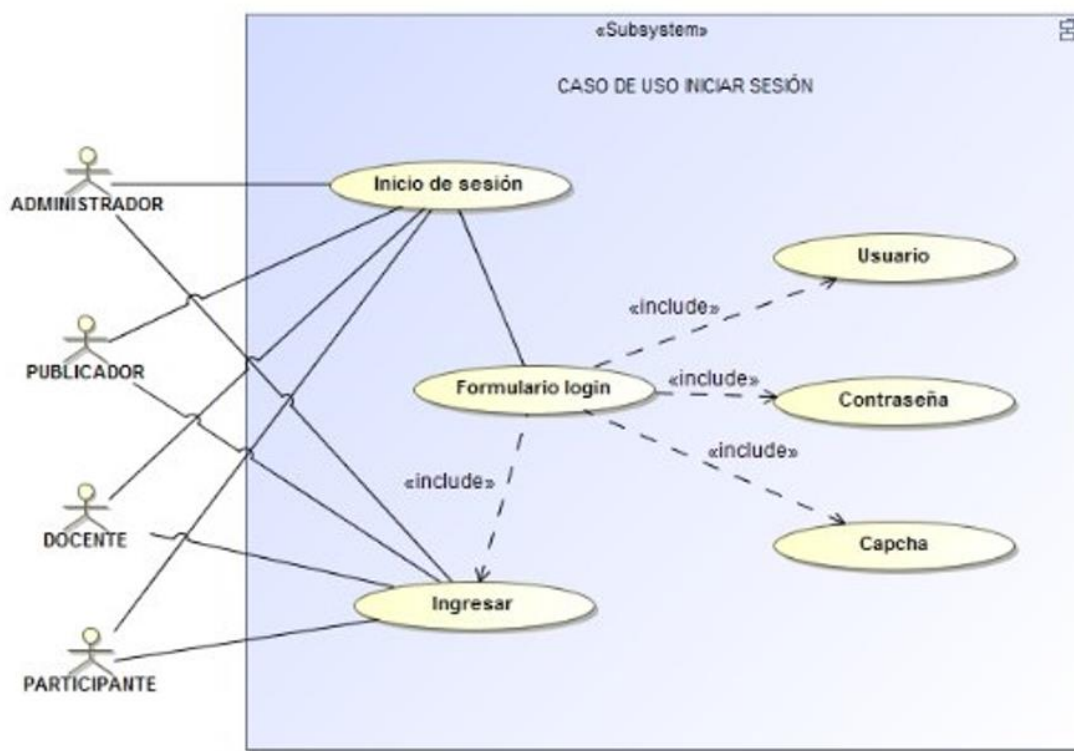




**Tabla 19.**  
**Caso de Uso general Web de Administración**

Caso de uso	Caso de uso general web de administración
Actores	Administrador, Docente, Publicador, Participante
Tipo	Primario esencial
Descripción	<p>El administrador cuenta con el privilegio de registrar, editar eliminara, cambiar estado y otros dentro del sistema en los diferentes módulos.</p> <p>El publicador puede registrar, editar, cambiar estados y otros dentro del sistema asignado con los privilegios que le asigna el administrador.</p> <p>El docente puede subir solamente su video de introducción a su curso y la verificación de sus notas de sus estudiantes y cursos dictados.</p> <p>El usuario tendrá acceso al sistema para que pueda verificar a la cantidad de cursos que anda inscrito y descargar sus certificados de manera automática.</p>

**Figura 22.**  
**Inicio de Sesión**



**Tabla 20**  
**Inicio de Sesión**

Caso de uso	Inicio de sesión
Actores	Administrador, Participante, publicador, Docente
Tipo	Primario esencial
Descripción	El administrador, publicador, Docente, Participante deben ingresar con su usuario, contraseña y captcha para ingresar al sistema.

Figura 23.

## Administrar Usuario

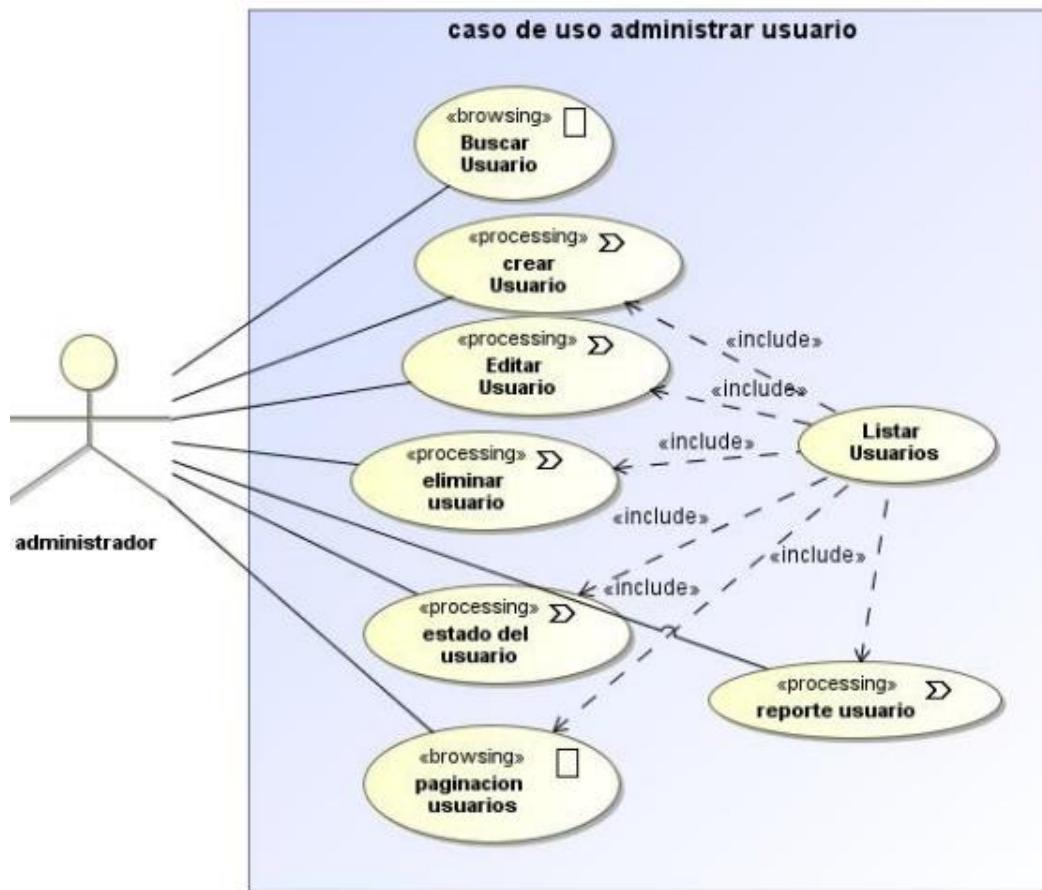
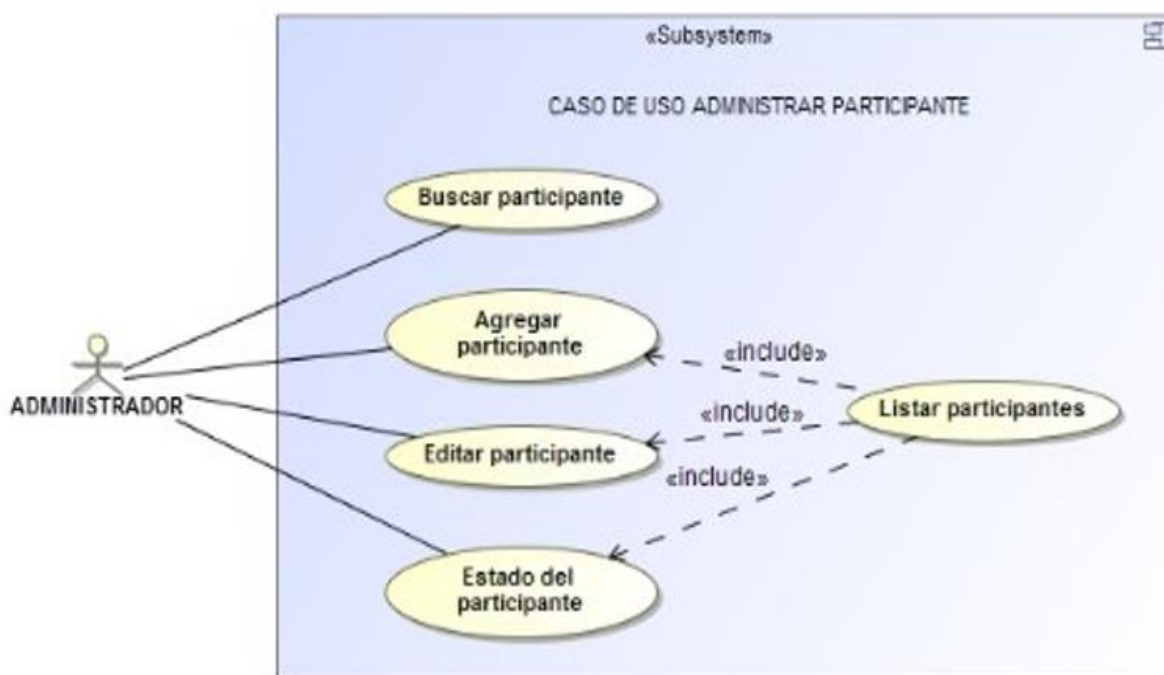


Tabla 21.

## Administrar Usuario

Caso de uso	Administrar usuario
Actores	Administrador
Tipo	Primario esencial
Descripción	El administrador, registrar, buscar, editar, cambiara el estado usuario.

**Figura 24.**  
**Administrar Participante**



**Tabla 22.**  
**Administrar Participante**

Caso de uso	Administrar participante
Actores	Administrador
Tipo	Primario esencial
Descripción	El administrador, registrara, buscara, editara, cambiara el estado del participante.

Figura 25.

## Administrar Roles y Permisos

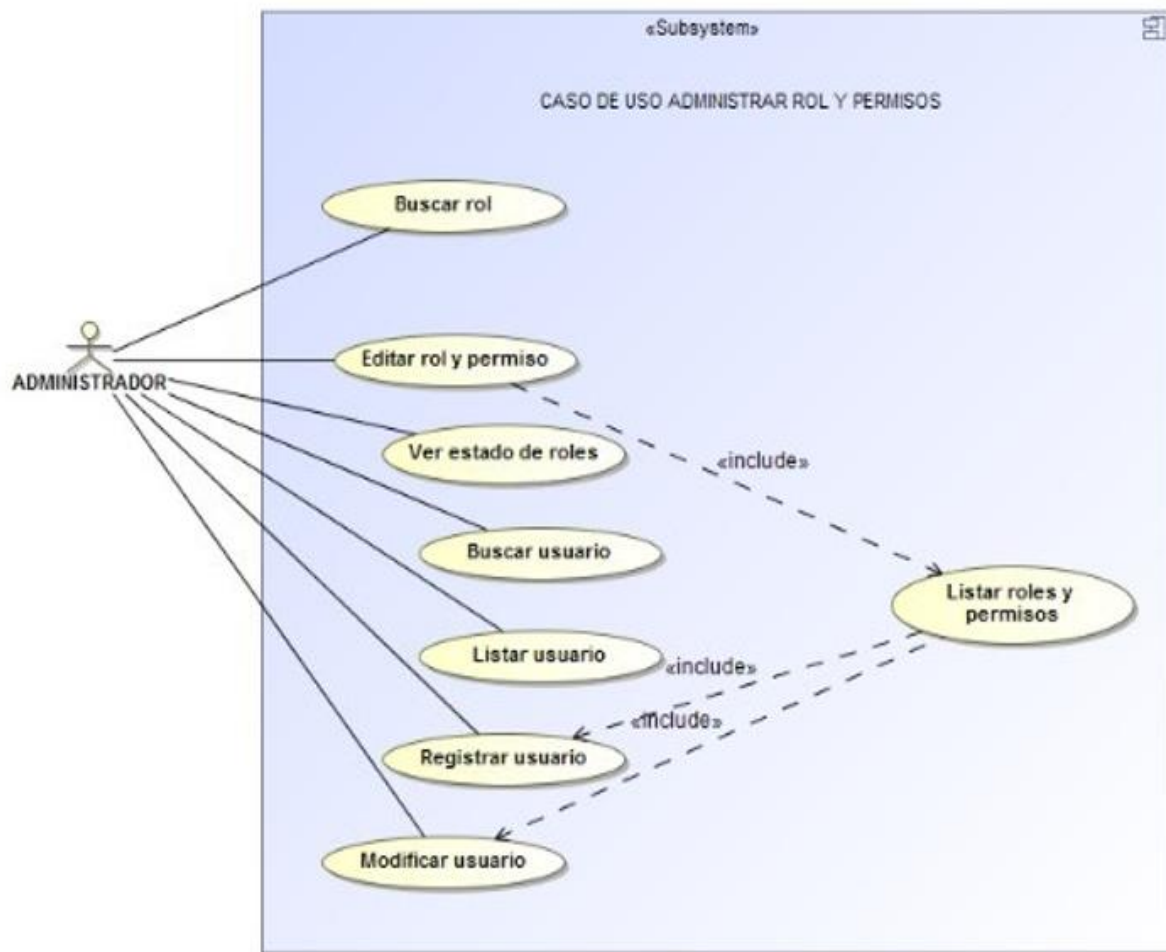
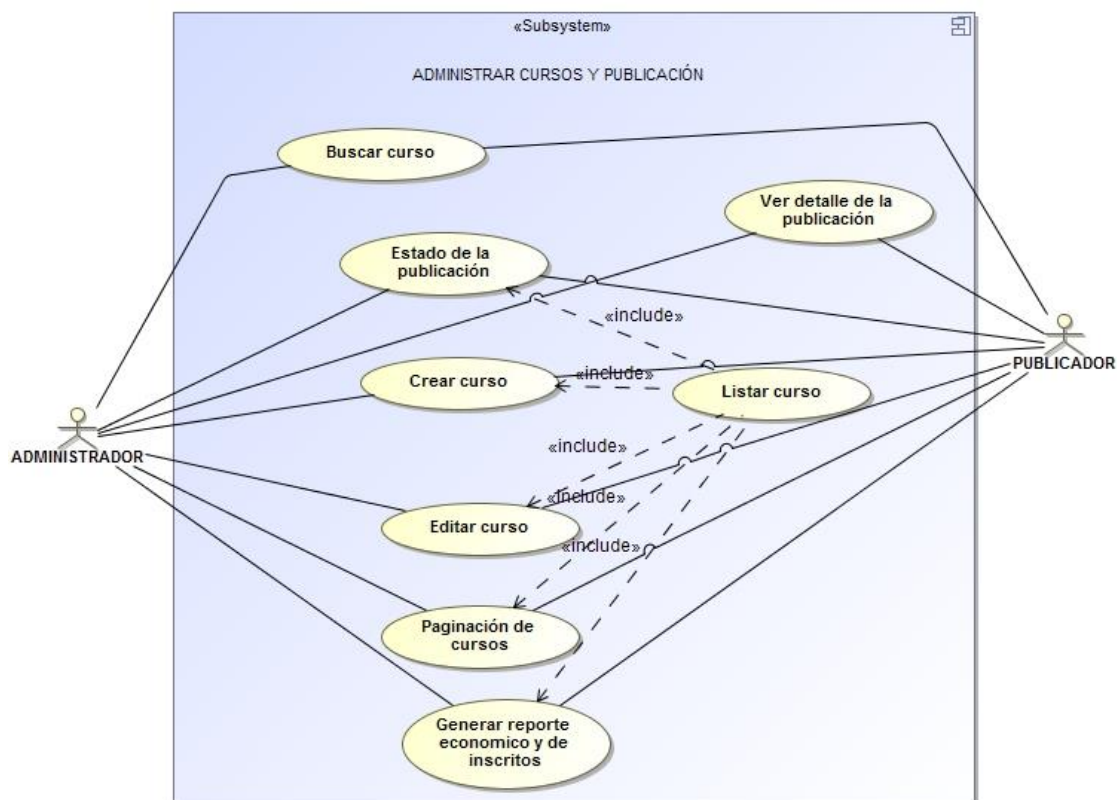


Tabla 23.

## Administrar Roles y Permisos

Caso de uso	Administrar roles y permisos
Actores	Administrador
Tipo	Primario esencial
Descripción	El administrador, edita el rol y los permisos del usuario

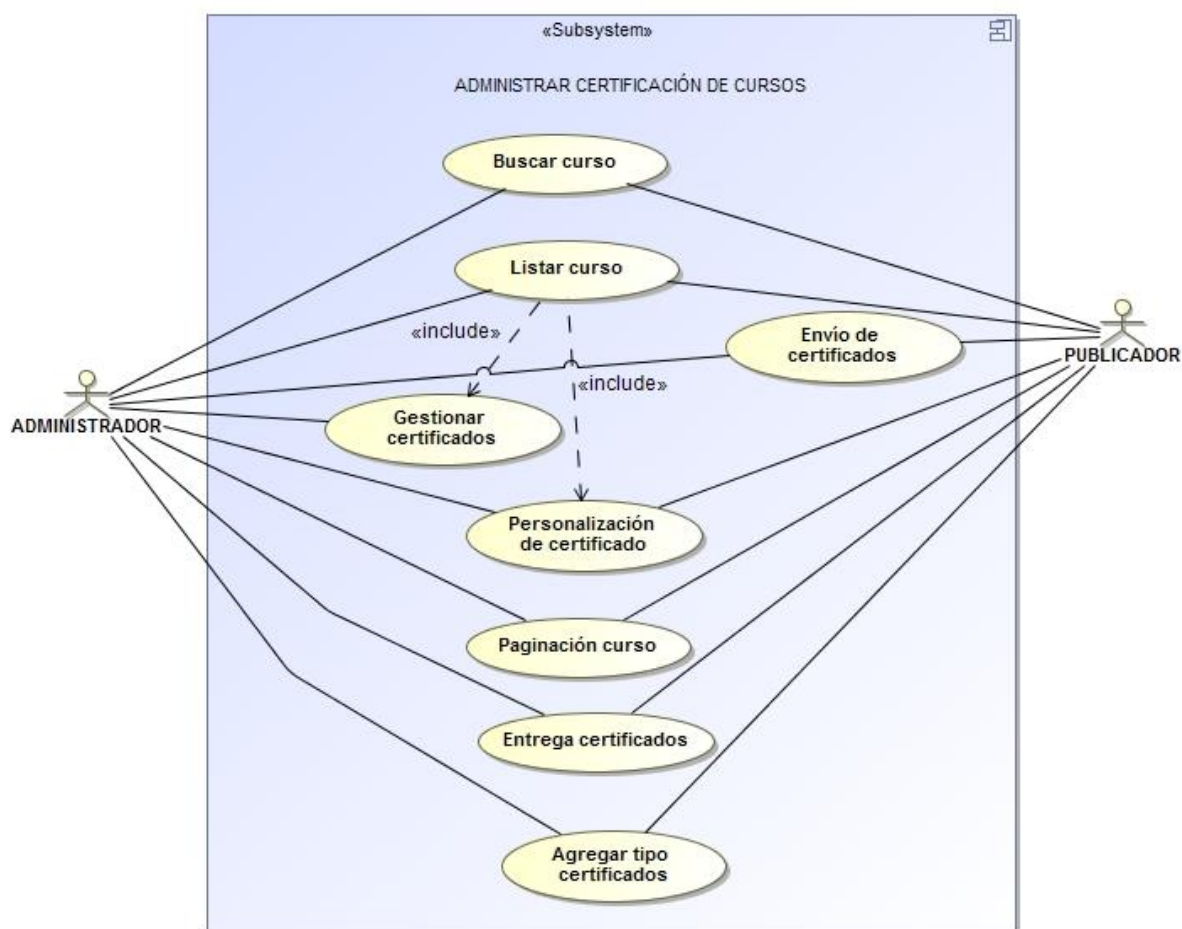
**Figura 26.**  
**Administrar Cursos y Publicación**



**Tabla 24**  
**Administrar Cursos y Publicación**

Caso de uso	Administrar cursos, publicación
Actores	Administrador, publicador
Tipo	Primario esencial
Descripción	<p>El administrador tiene los privilegios de configurar. Crear un curso, editar un curso, buscar un curso, reportes económicos, reporte de inscritos, gestionar certificación.</p> <p>EL publicador tiene los privilegios asignador por el administrador como: privilegios de configurar. Crear un curso, editar un curso, buscar un curso, reportes económicos, reporte de inscritos, gestionar certificación.</p>

**Figura 27.**  
**Administrar Certificación**

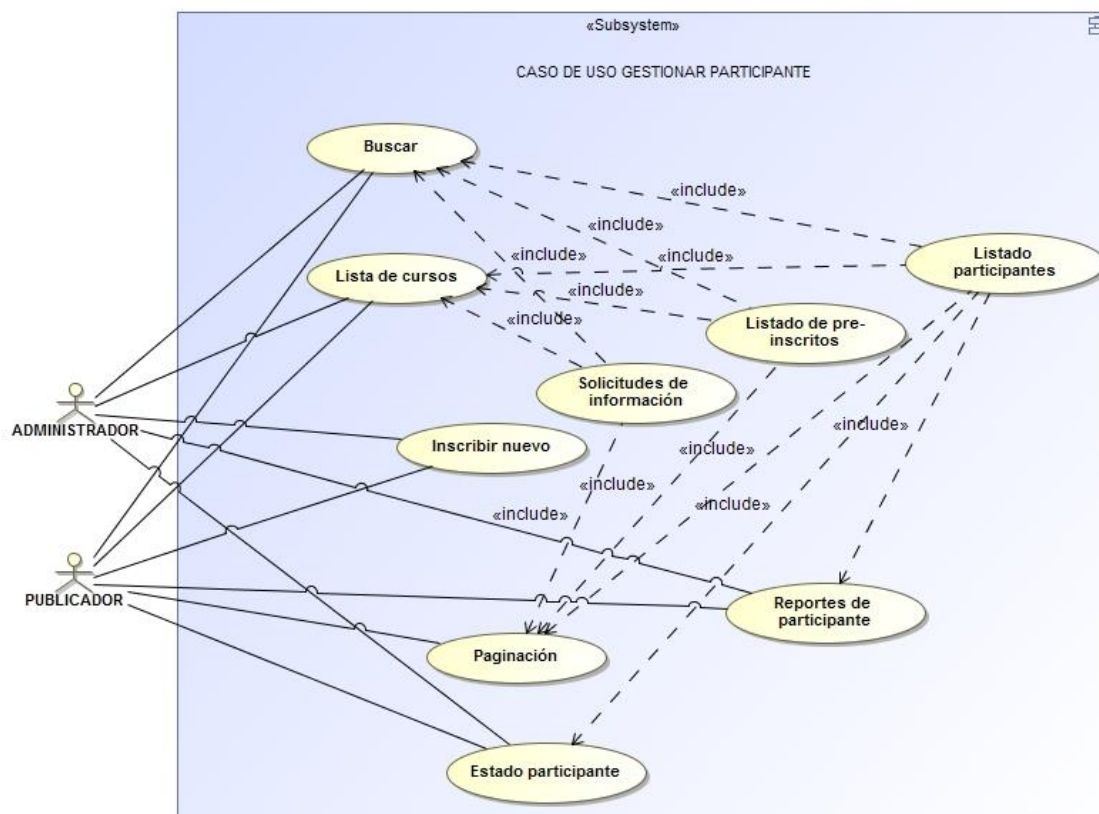


**Tabla 25.**  
**Administrar Certificación**

Caso de uso	Administrar certificación
Actores	Administrador, publicador
Tipo	Primario esencial
Descripción	<p>El administrador tiene los privilegios de configurar los cursos editar certificación, personalizar certificación, editar entrega de certificado.</p> <p>EL publicador tiene los privilegios asignador por el administrador como: privilegios de configurar. Crear un curso, gestionar certificados finalizar configuración de certificación, entrega de certificados y envíos.</p>



**Figura 28.**  
**Administrar Participantes**

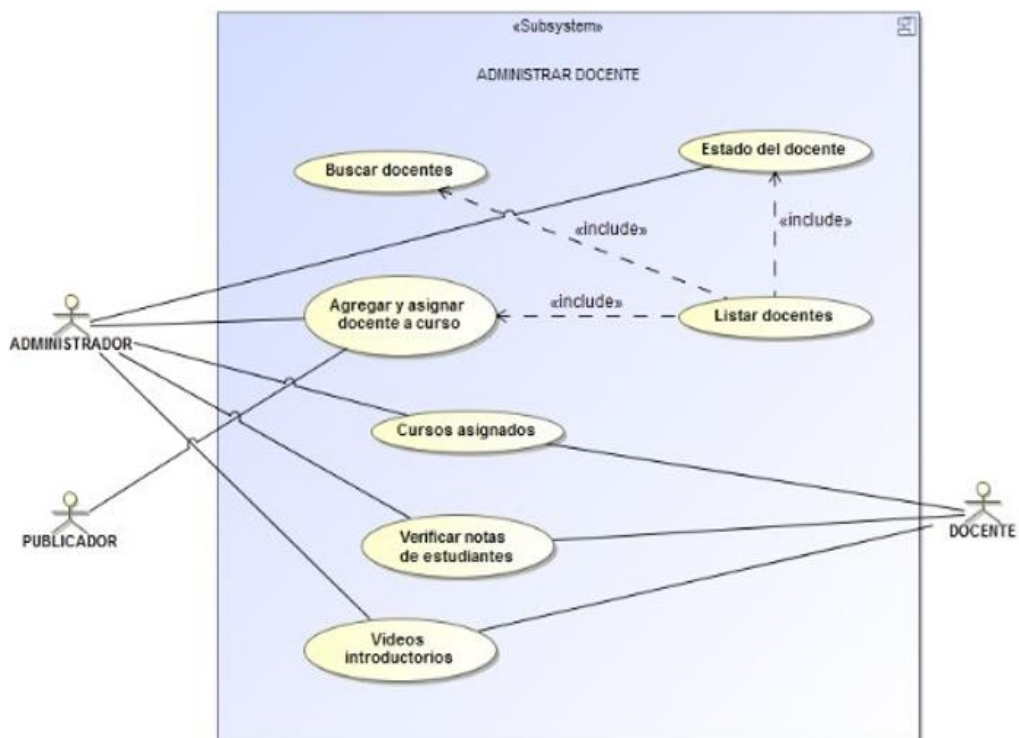


**Tabla 26**  
**Administrar Participantes**

Caso de uso	Administrador publicador
Actores	Administrador, publicador
Tipo	Primario esencial
Descripción	El administrador tiene los privilegios de inscribir, confirmar pre inscritos, verificar, gestionar participantes, reportes. El publicador tiene el privilegio de ser asignador por el administrador como: inscribir, confirmar pre inscritos, verificar, gestionar participantes, reportes.



**Figura 29.**  
**Administrar Docente**



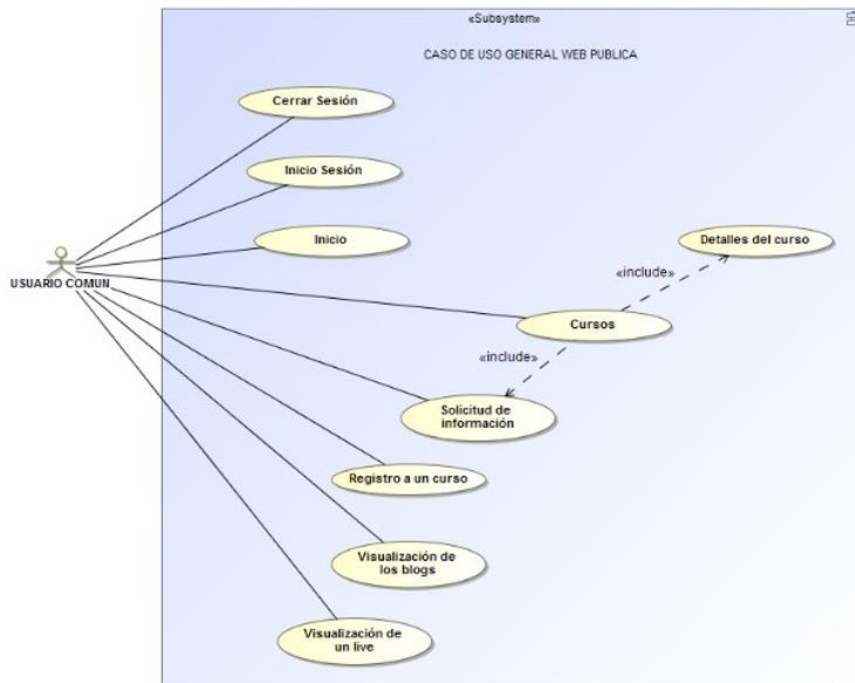
**Tabla 27**  
**Administrar Docente**

Caso de uso	Administrar docente
Actores	Administrador, docente
Tipo	Primario esencial
Descripción	El administrador tiene los privilegios de crear, eliminar, editar, cambiar estado. El docente tiene como rol poder subir su video introductorio para llamar la atención de los participantes, verificación de sus cursos y participantes.

### 3.2.1.2 Casos de Uso Publica.

**Figura 30.**

**Caso Web Pública**

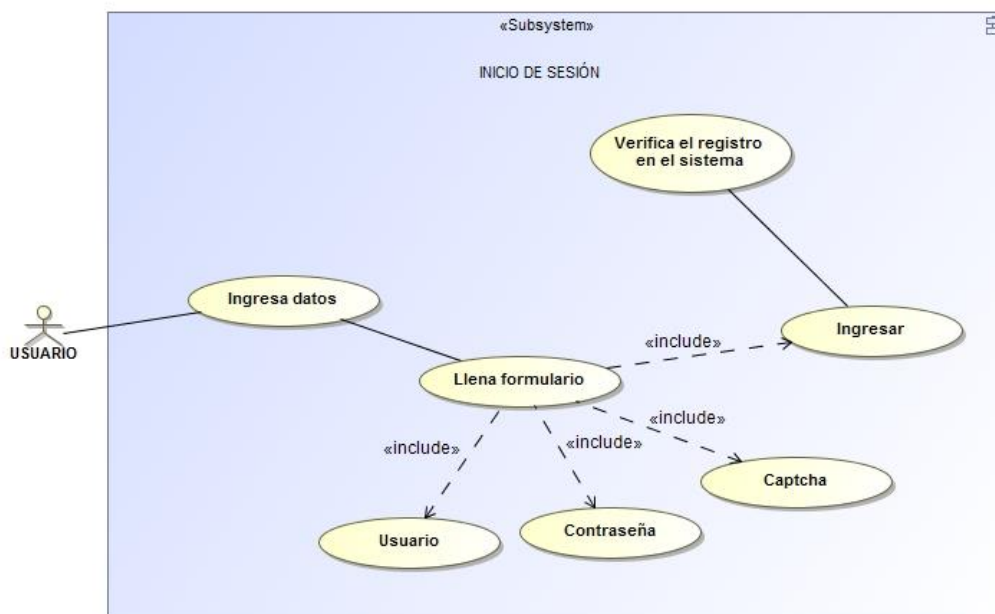


**Tabla 28.**

**Caso de Uso Web Pública**

Caso de uso	Web publica
Actores	Usuario común
Tipo	Primario esencial
Descripción	El usuario común puede ver los cursos disponibles, solicitar información, ver los blogs educativos, detalles de la información, visualizar live si es que están en transmisión.

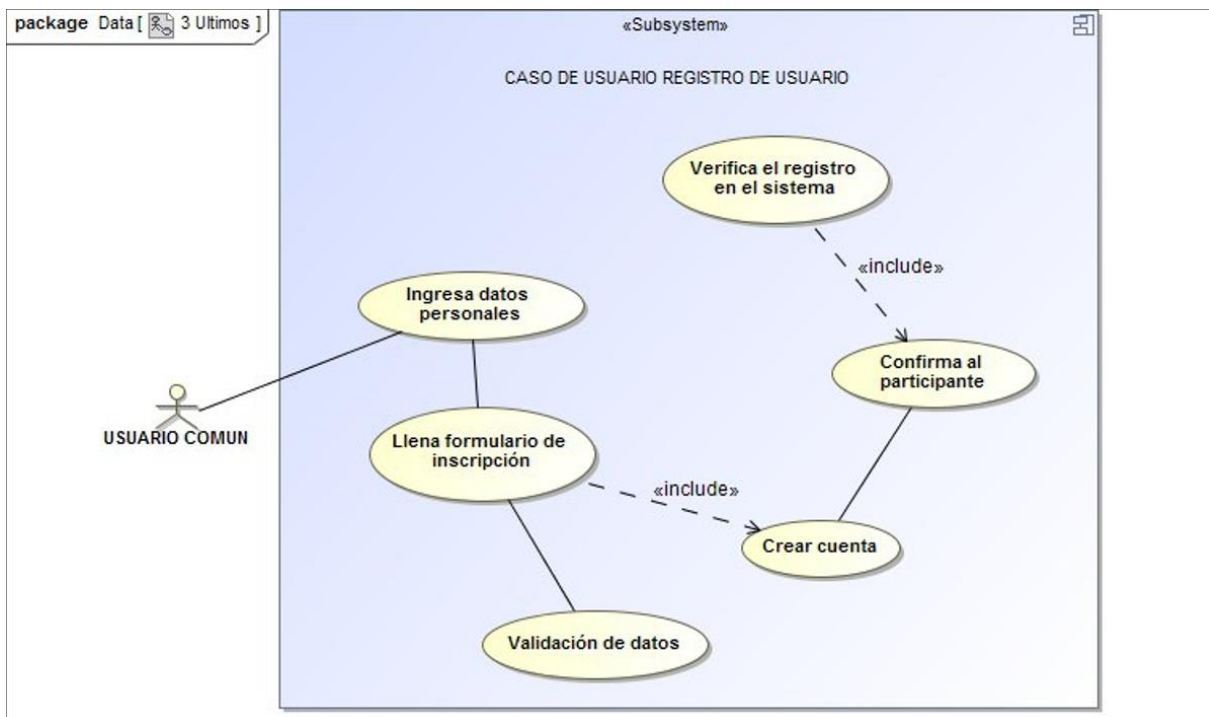
**Figura 31.**  
**Inicio de Sesión**



**Tabla 29**  
**Inicio de Sesión**

Caso de uso	Inicio de sesión
Actores	Usuario común
Tipo	Primario esencial
Descripción	El usuario común antes de ingresar al sistema debe estar registrado y confirmado por el publicador posteriormente recién ingresara con su usuario y contraseña asignado.

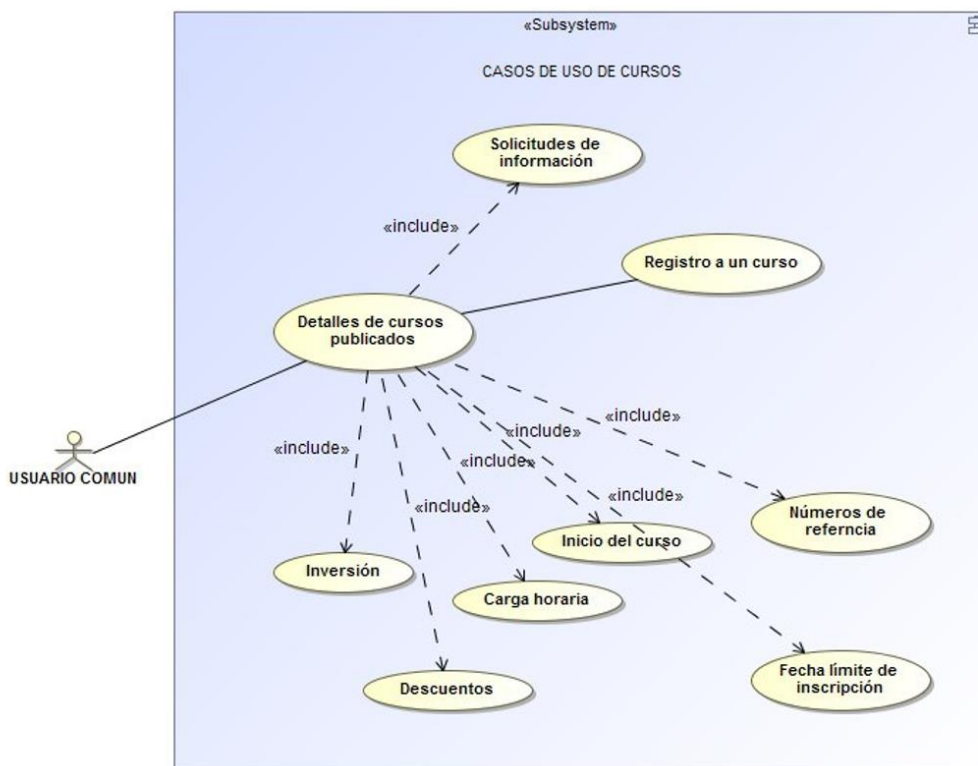
**Figura 32.**  
**Registro de Usuario**



**Tabla 30**  
**Registro de Usuario**

Caso de uso	Registro de usuario
Actores	Usuario común
Tipo	Primario esencial
Descripción	El usuario común debe ingresar sus datos como: debe ingresar sus datos personales para el llenado de una inscripción de un curso y según a ello se registrarán sus datos y se le irán asignando las credenciales de acceso, pero eso pasara una vez que el administrador o publicador verifique esos datos registrados.

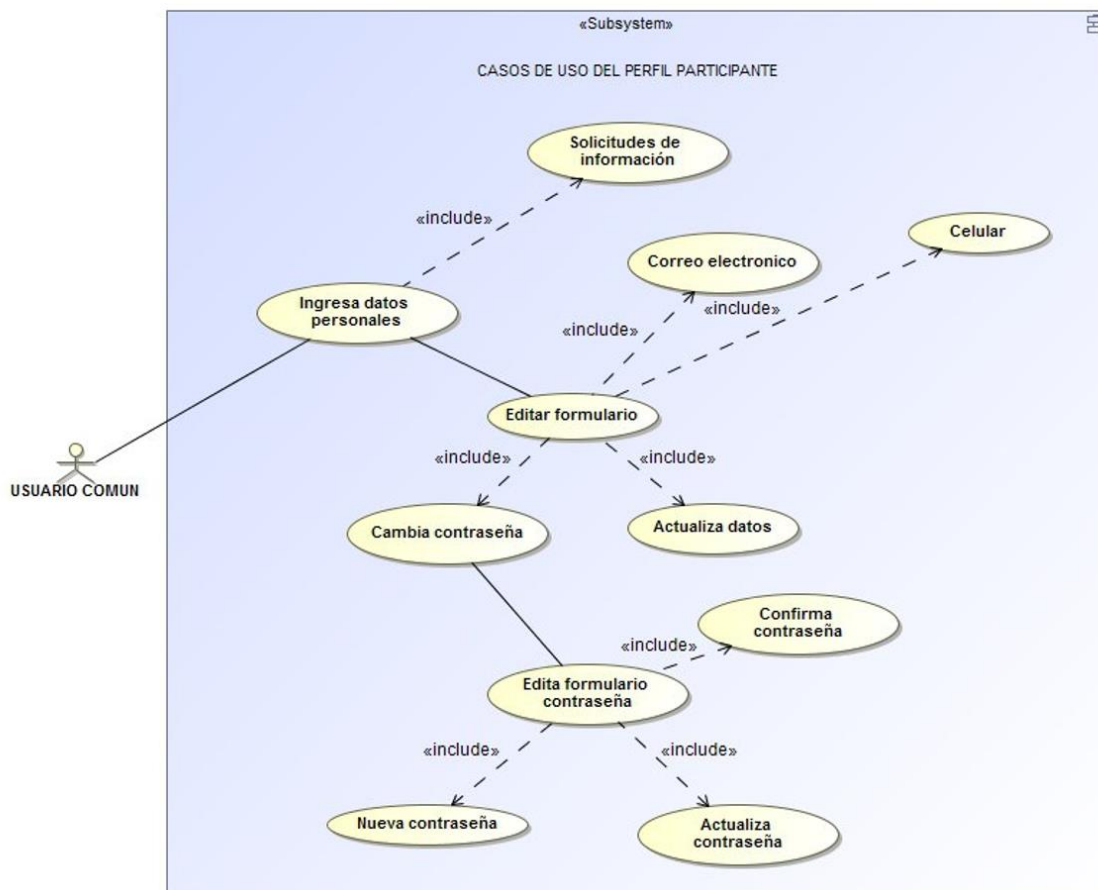
**Figura 33.**  
**Listado de Cursos**



**Tabla 31**  
**Cursos**

Caso de uso	Detalles del curso
Actores	Usuario común
Tipo	Primario esencial
Descripción	El usuario común puede ver los detalles del curso sin una cuenta de usuario y registro del curso si está interesado.

**Figura 34.**  
**Perfil del Participante**



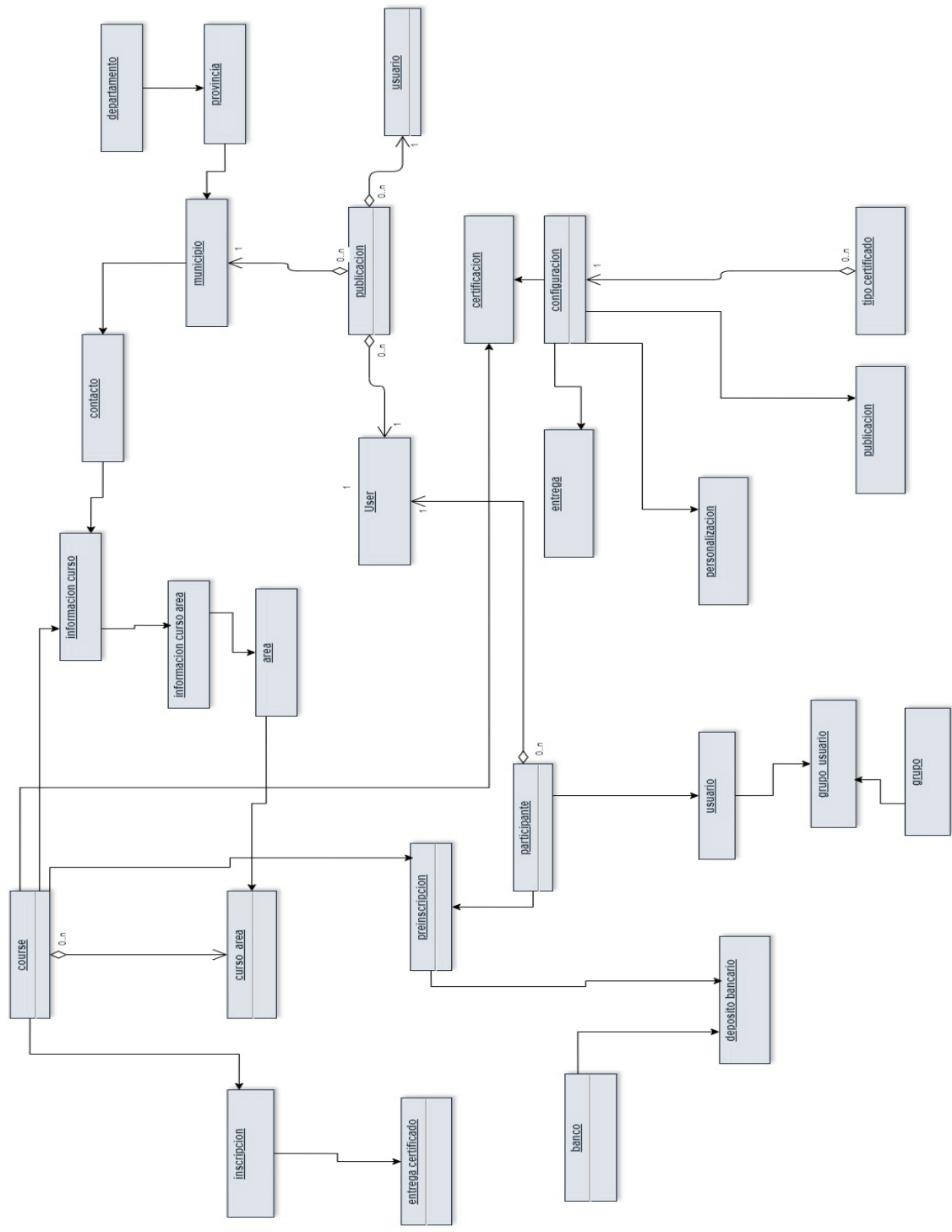
**Tabla 32.**  
**Perfil del Participante**

Caso de uso	Perfil del participante
Actores	Participante
Tipo	Primario esencial
Descripción	El participante una vez ingresando al sistema podrá ver sus datos personales actualizarlos y cambiar de contraseña por seguridad.

### 3.2.2 Modelo de Contenido

Se muestra el diseño del modelo de contenido del sistema:

**Figura 35.**  
**Modelo de Contenido de la Base de Datos**



### 3.2.3 Modelo de Navegación

#### 3.2.3.1 Navegación Administrativa.

Figura 36.

Inicio de Sesión

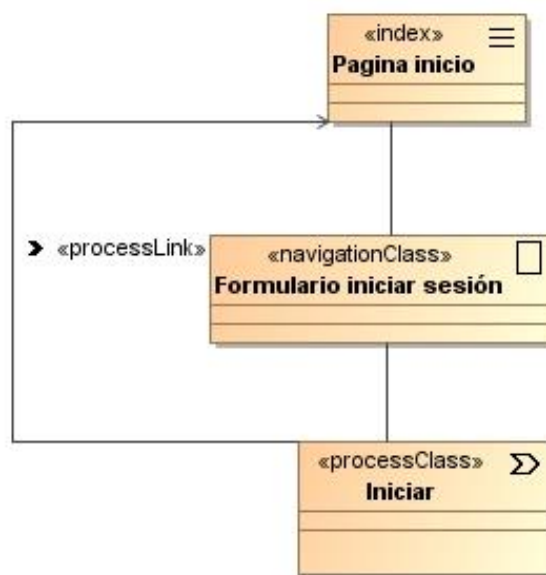
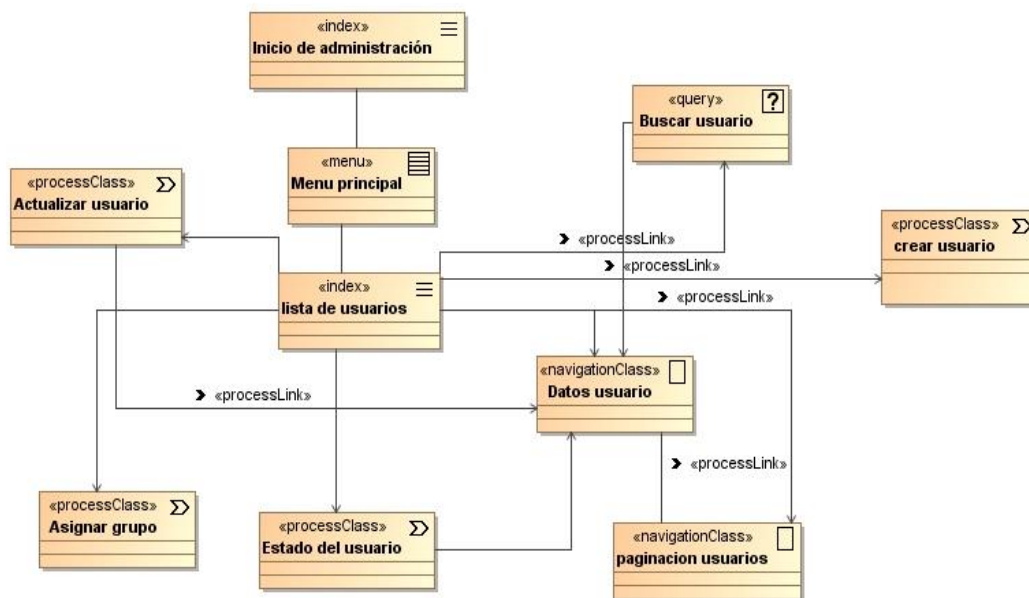


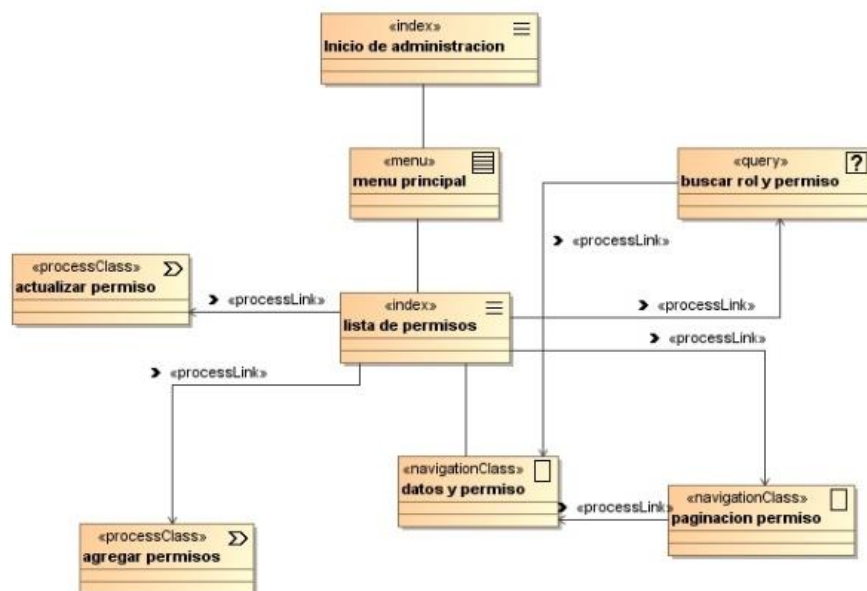
Figura 37.

Administrar Usuario y Grupos (administrador)

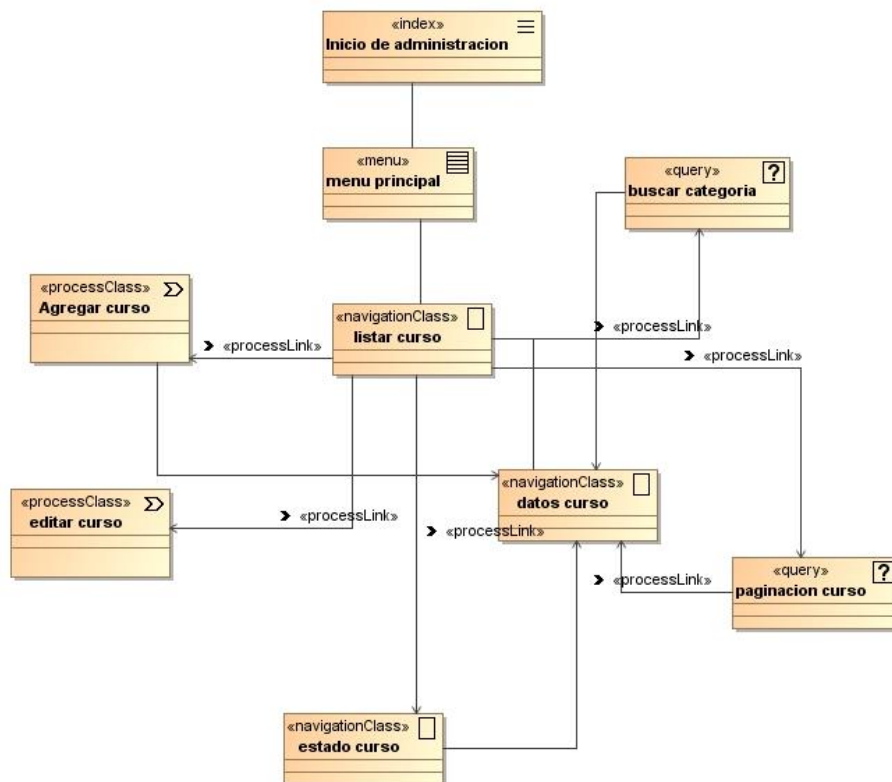




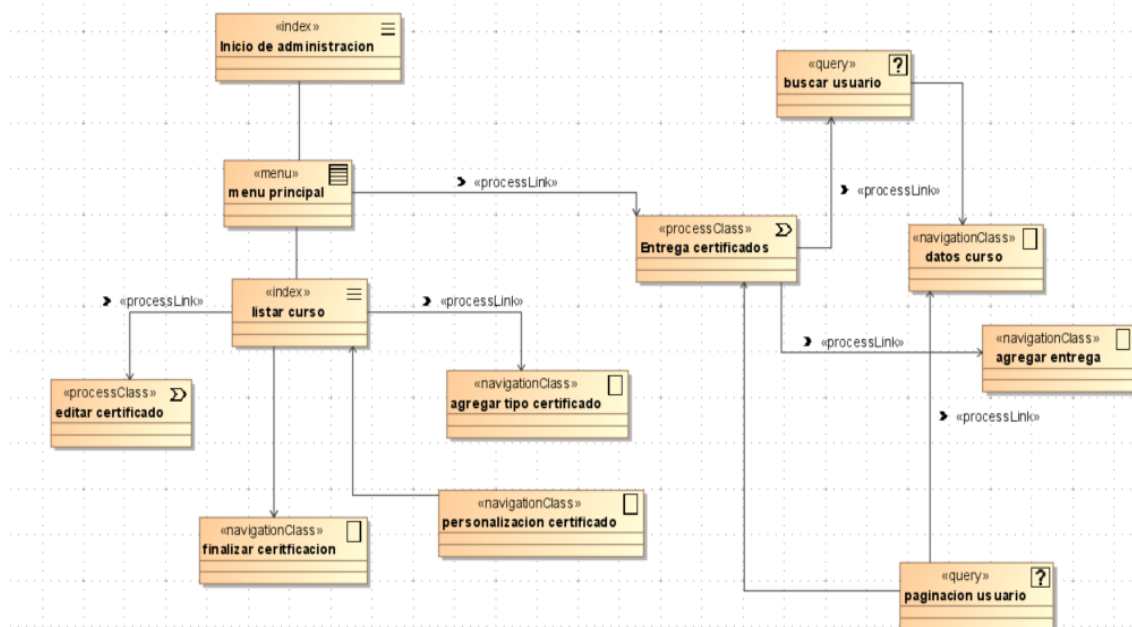
**Figura 38.**  
**Administrar Grupos (administrador)**



**Figura 39.**  
**Administrar cursos (administrador, publicador)**



**Figura 40.**  
**Administrar certificados (administrador, editor)**



**Figura 41.**  
**Administrar Participantes (administrador, publicador)**

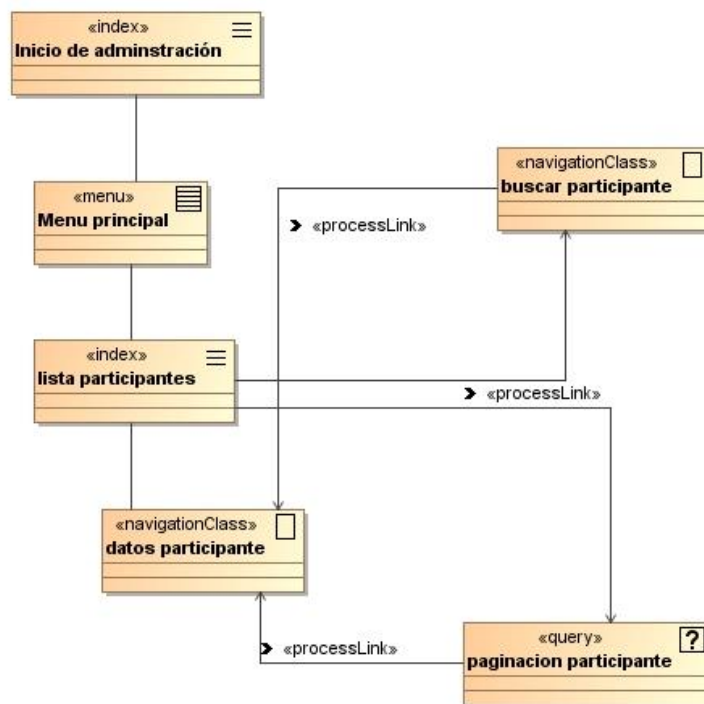


Figura 42.

## Administrar Inscritos y Preinscritos (administrador, publicador)

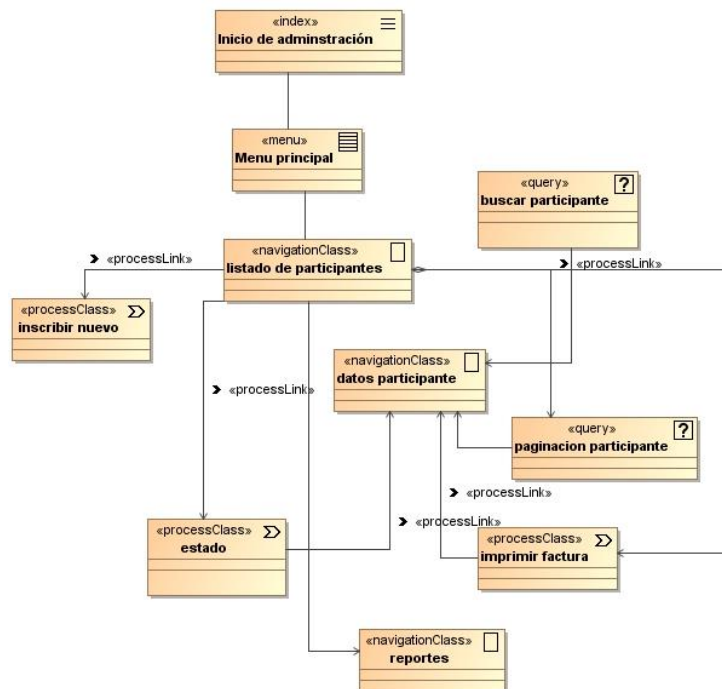


Figura 43.

## Administrar Docentes (administrador, publicador)

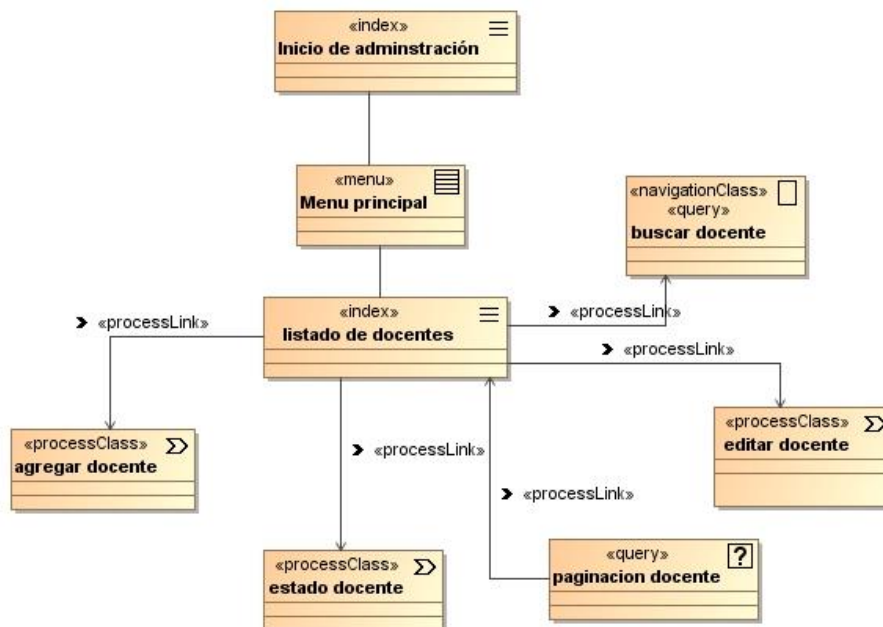
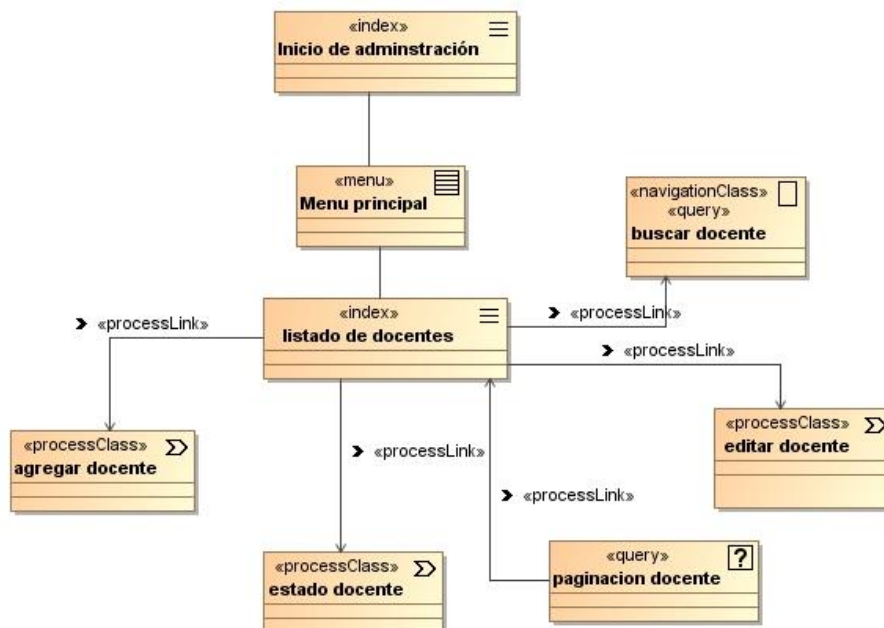


Figura 44.

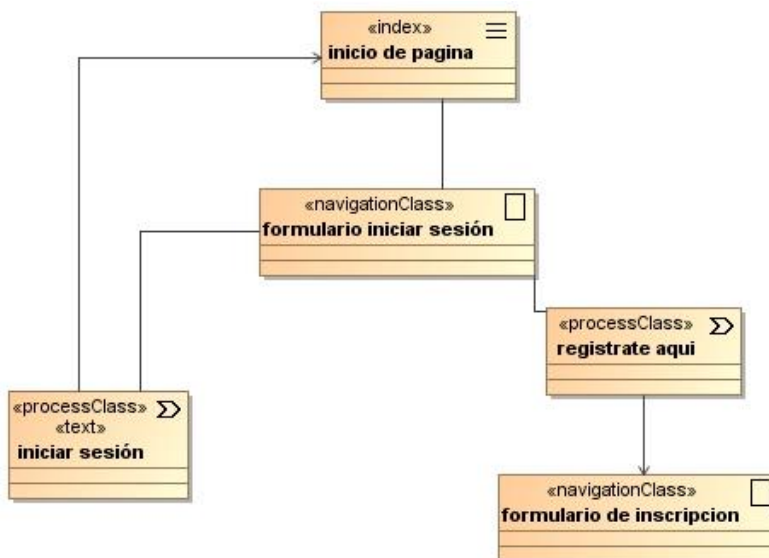
Administrar Blog (administrador, publicador, usuario, docente)



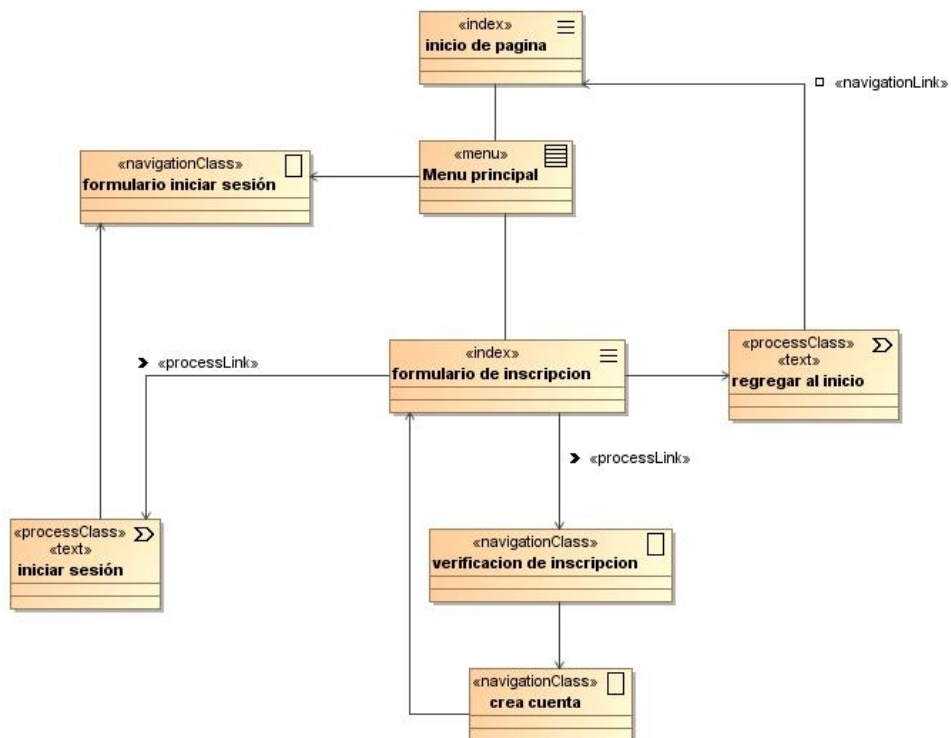
### 3.2.3.2 Navegación Pública.

Figura 45.

Inicio de Sesión (participante)



**Figura 46.**  
**Registro Usuario (participante)**



**Figura 47.**  
**Cursos (Usuario común, participante)**

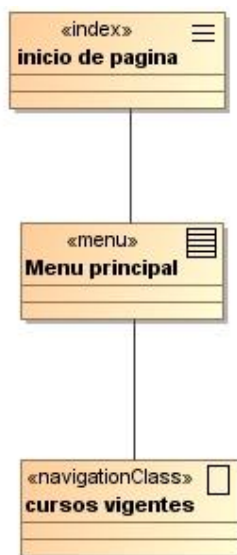


Figura 48.

Información o Detalles de Curso (Usuario común, participante)

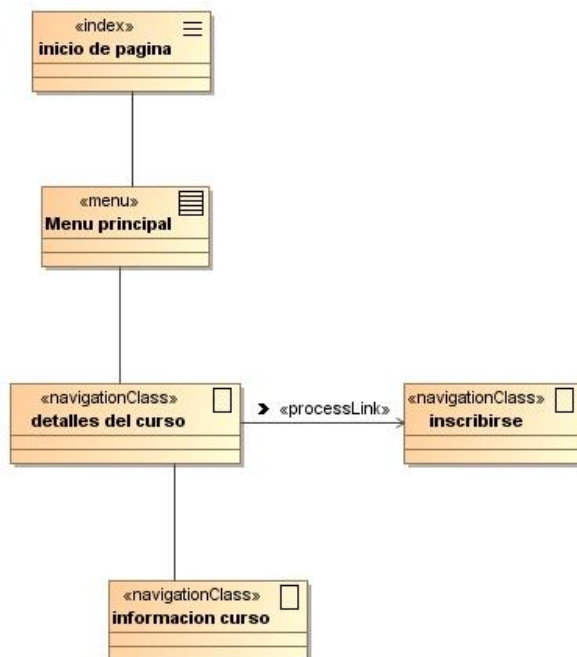


Figura 49.

Perfil del Usuario (participante)

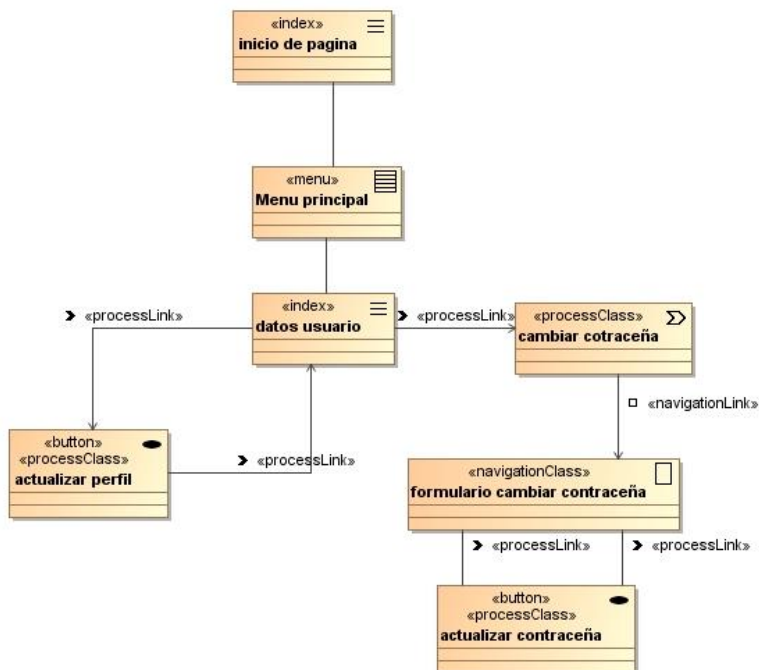
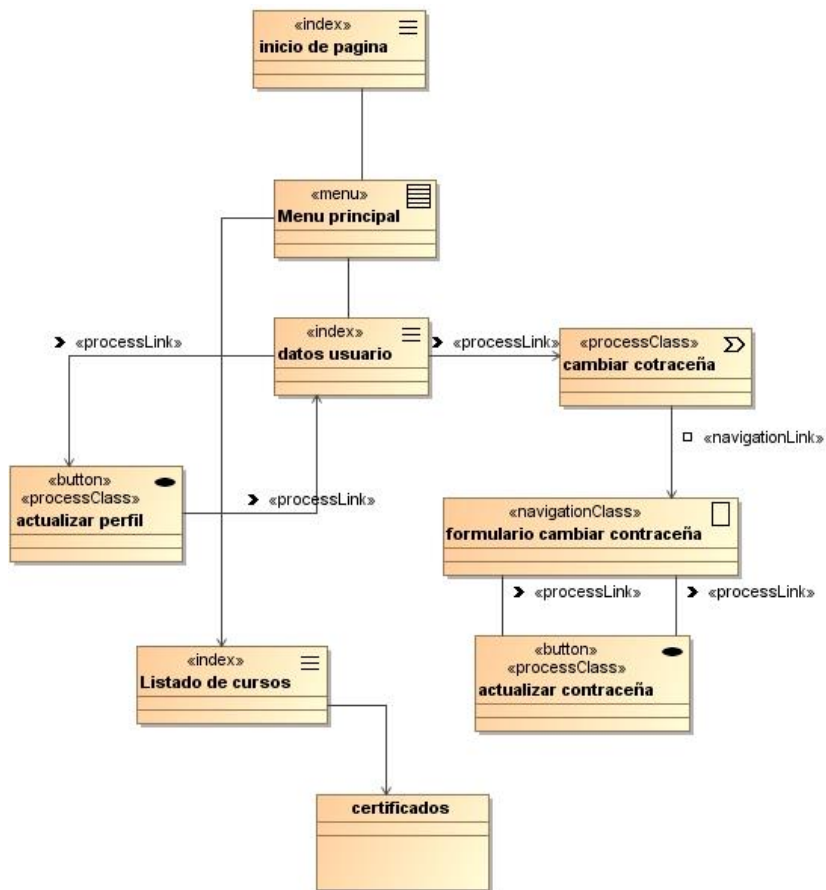
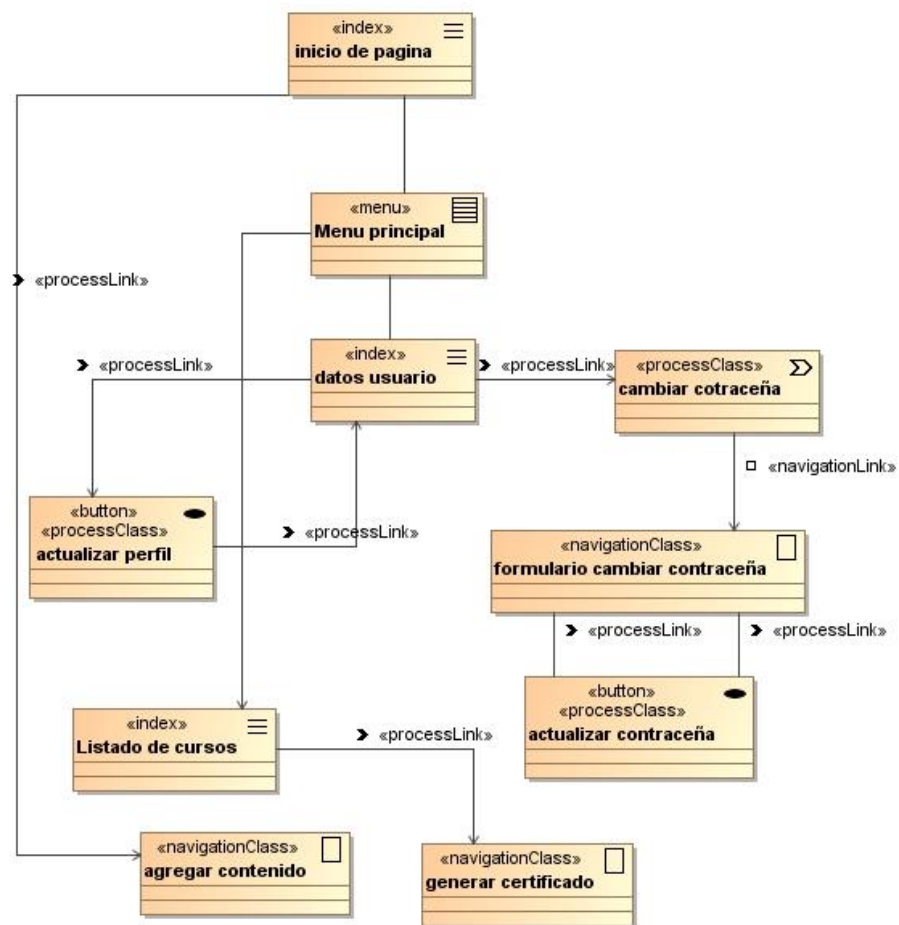


Figura 50.

## Mis Cursos y Certificados (participante)



**Figura 51.**  
**Mis Cursos y Certificados (docente)**





### 3.2.4 Modelo de Presentación

#### 3.2.4.1 Modelo de Presentación Administración.

Figura 52.

Inicio de Sesión

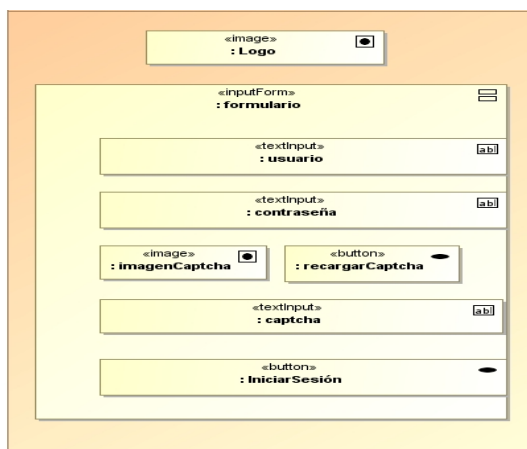
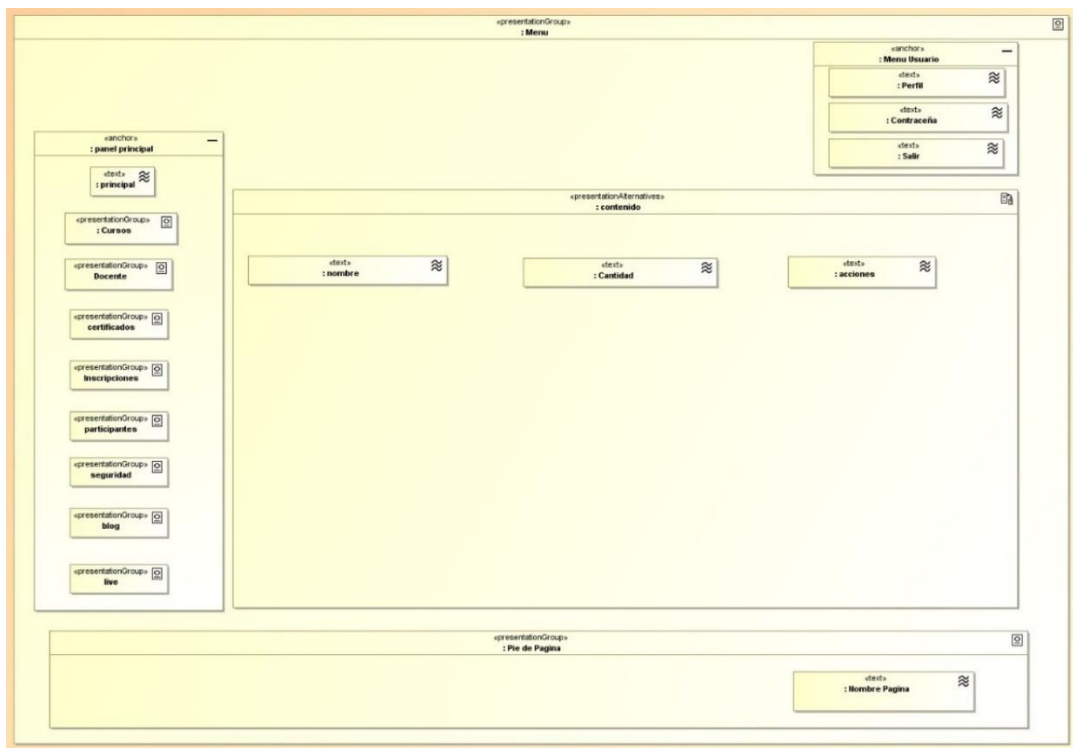
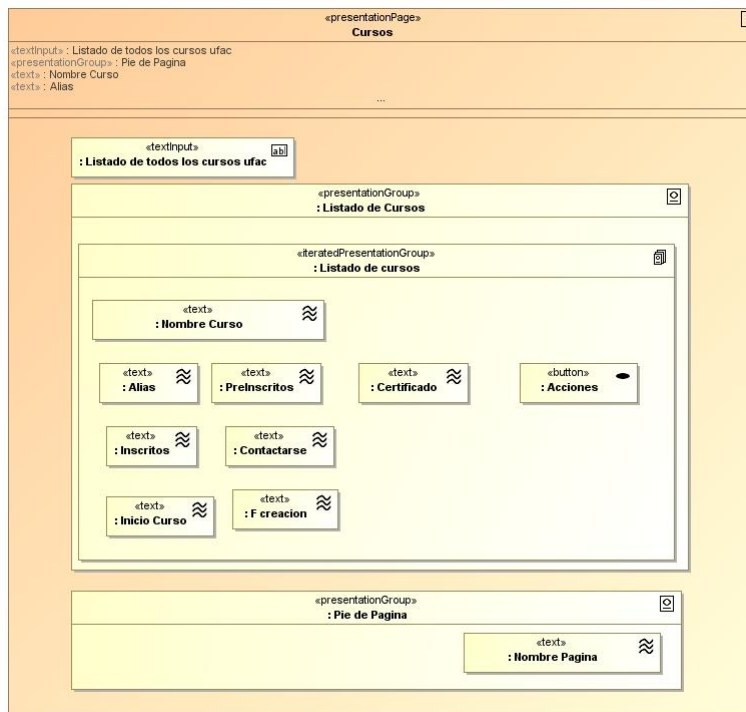


Figura 53.

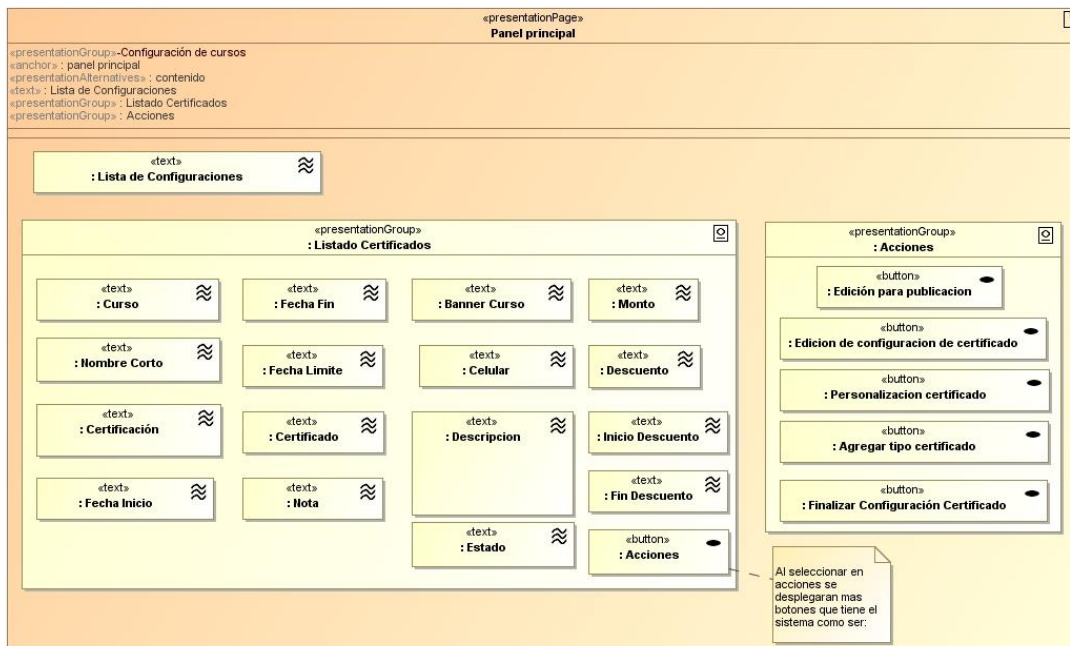
Panel Inicio



**Figura 54.**  
**Administrar Cursos (administrador, publicador)**



**Figura 55.**  
**Administrar Configuración de Publicación (administrador, publicador)**



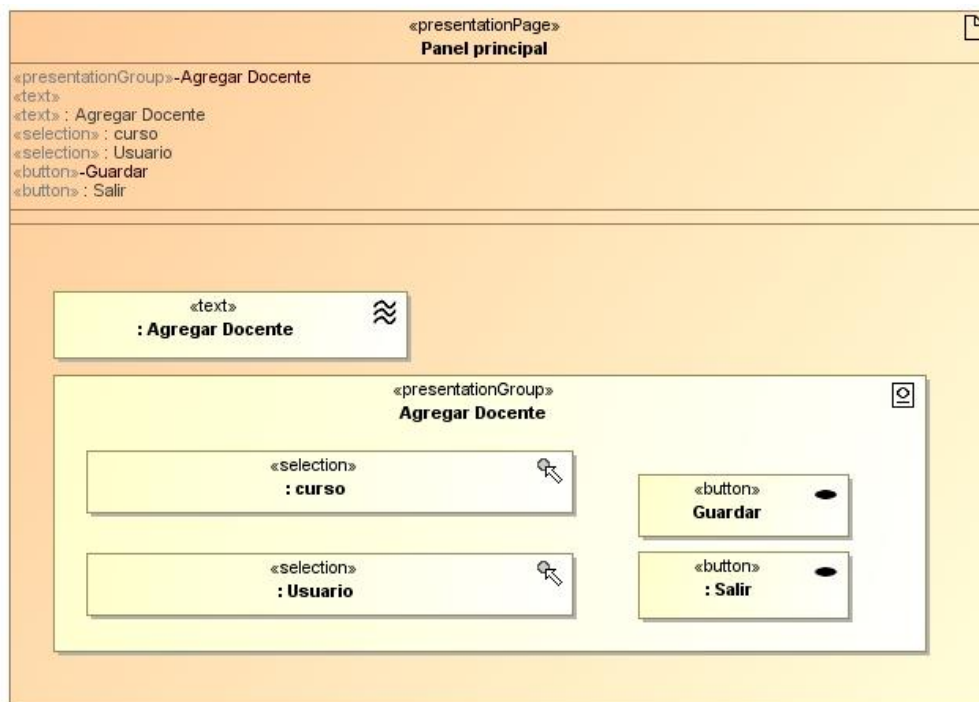
**Figura 56.**

**Editar Configuración de Publicidad Curso (administrador, publicador)**



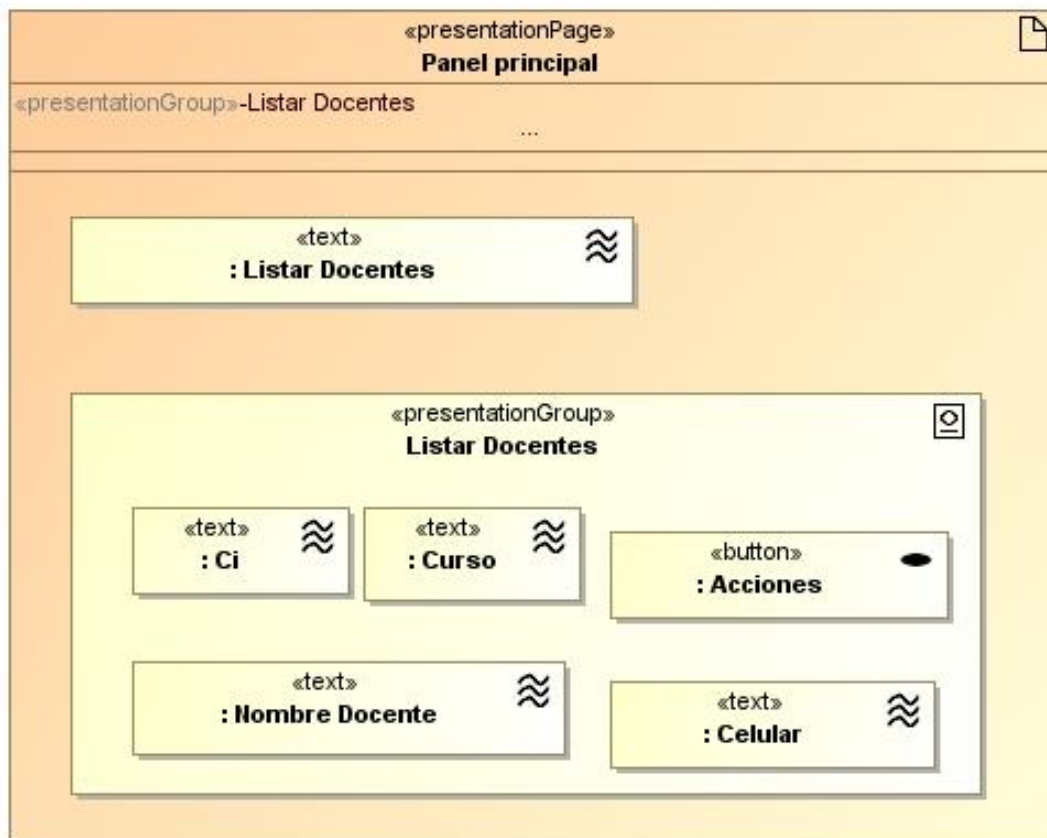
**Figura 57.**

**Administrar Docentes (administrador, publicador)**



**Figura 58.**

**Listado de Docentes (administrador, publicador)**



**Figura 59.**  
**Configuración de Certificados (administrador, publicador)**

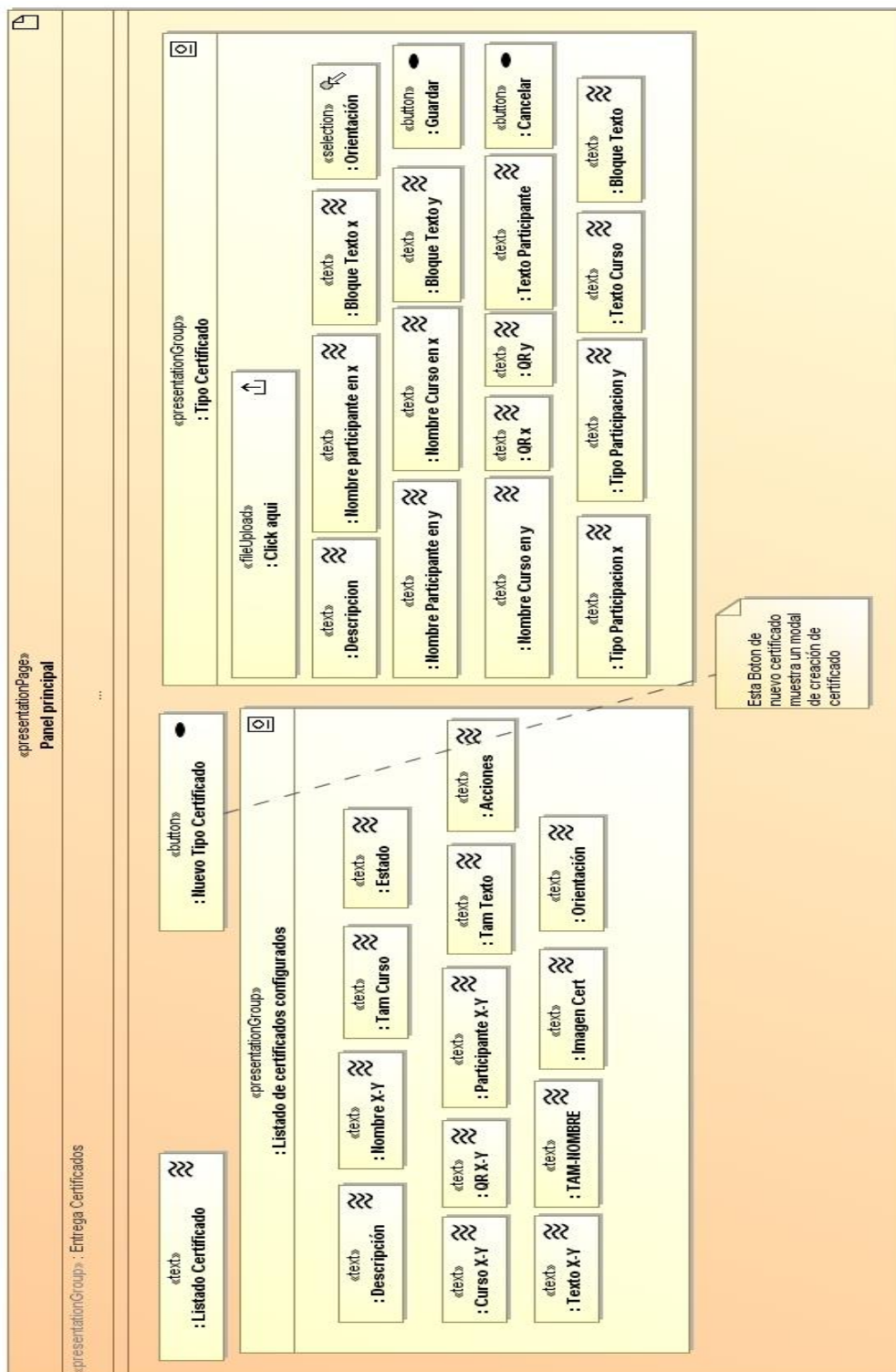


Figura 60.

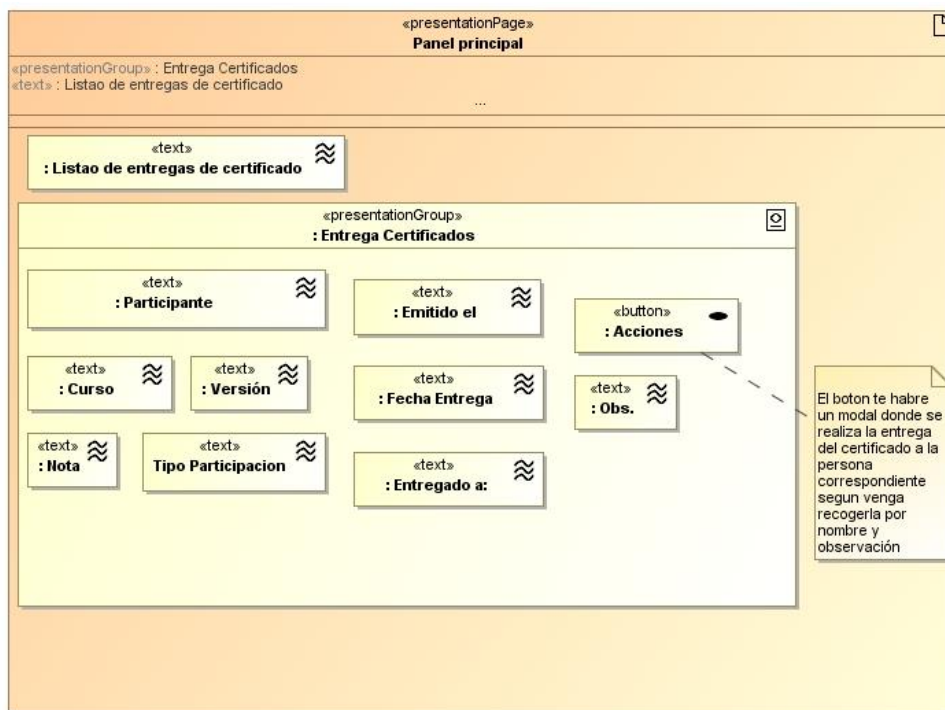
**Entrega de Certificados (administrador, publicador)**

Figura 61

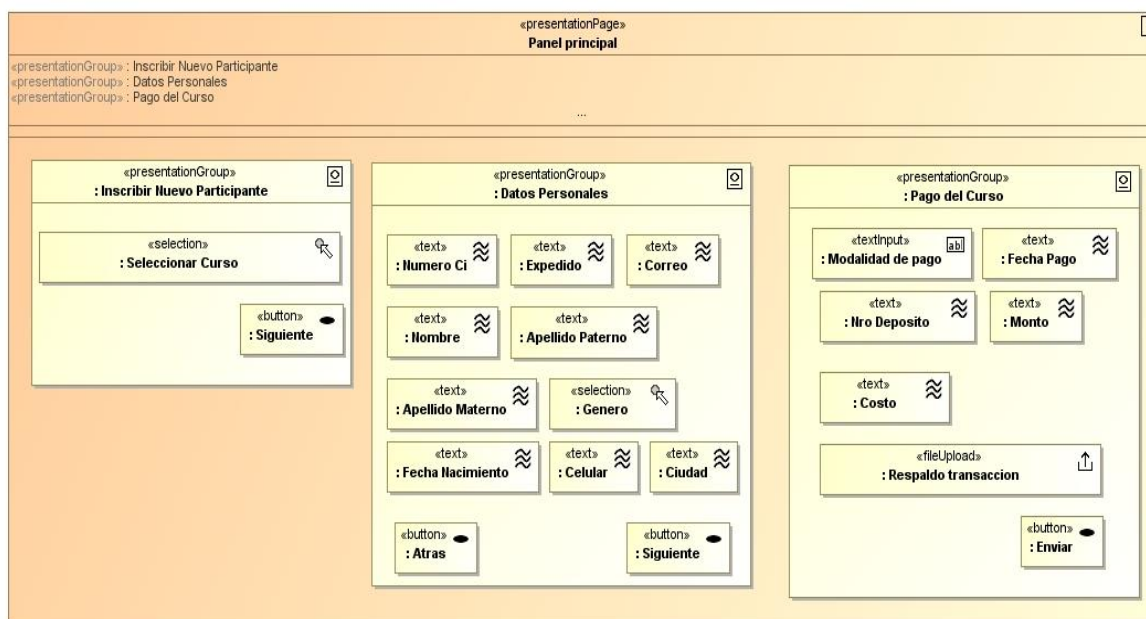
**Administrar Inscripciones (administrador, publicador)**

Figura 62.

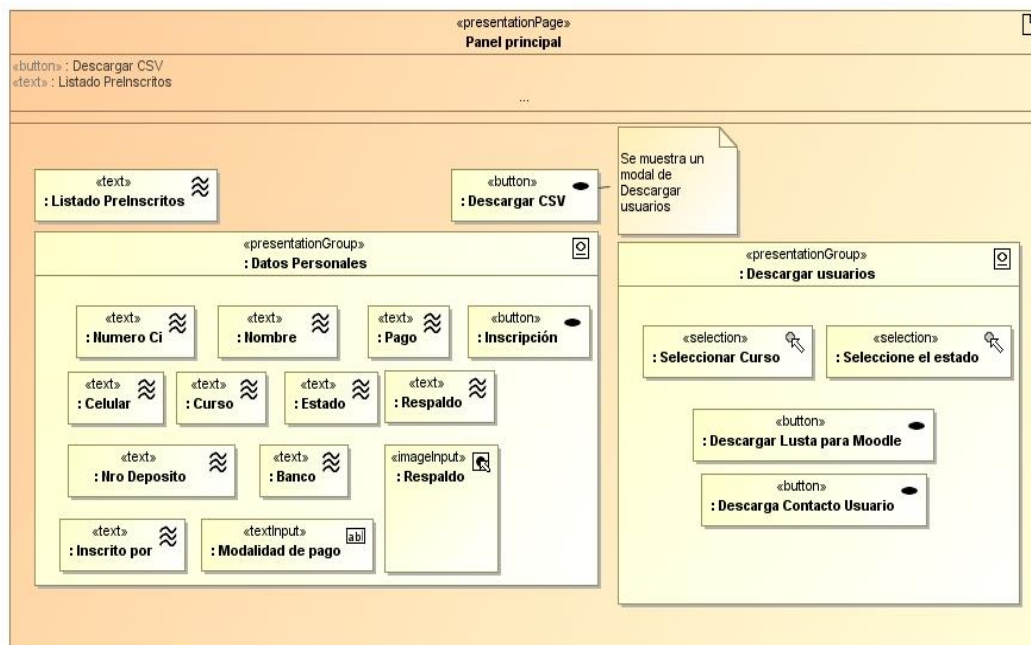
**Administrar Preinscritos (administrador, publicador)**

Figura 63.

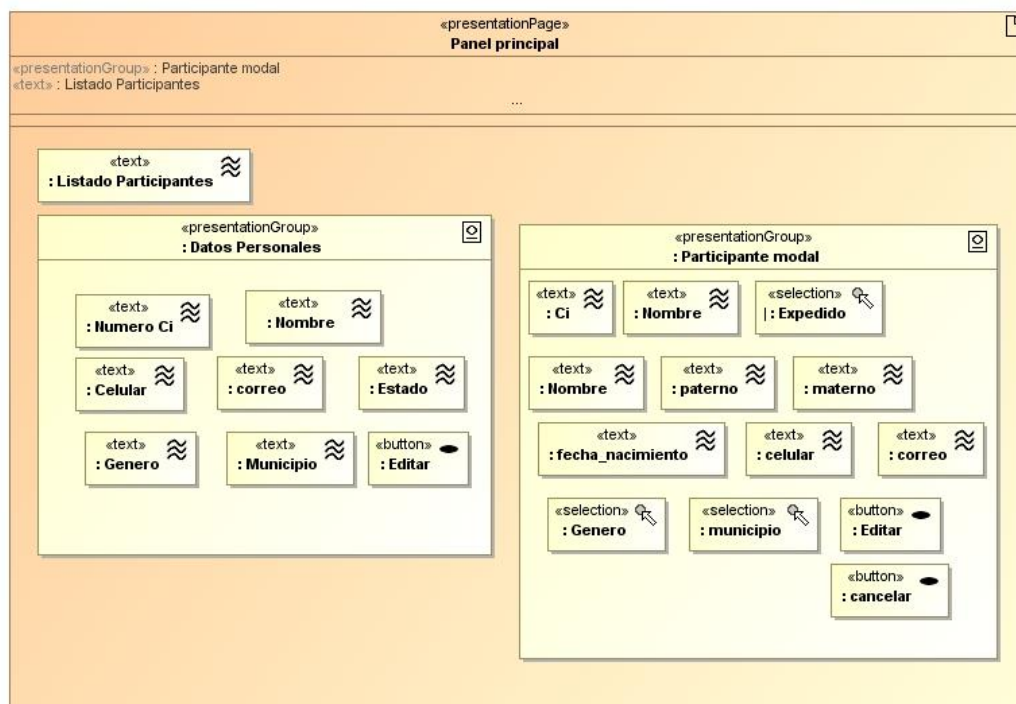
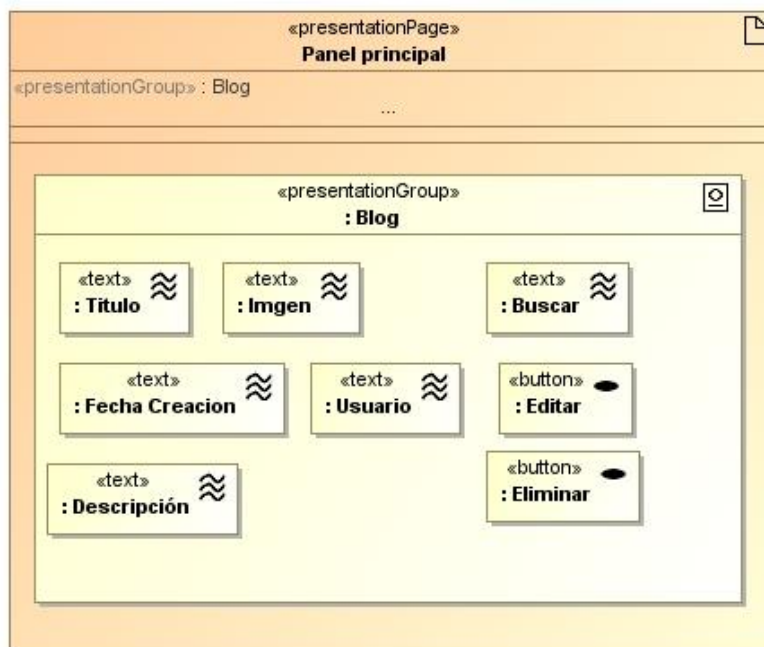
**Administrar Participantes (administrador, publicador)**



Figura 64.

**Administrar Blog (administrador, publicador, participante, docente)**



### 3.2.4.2 Modelo de Presentación Pública.

Figura 65.

**Inicio de Sesión (usuario)**

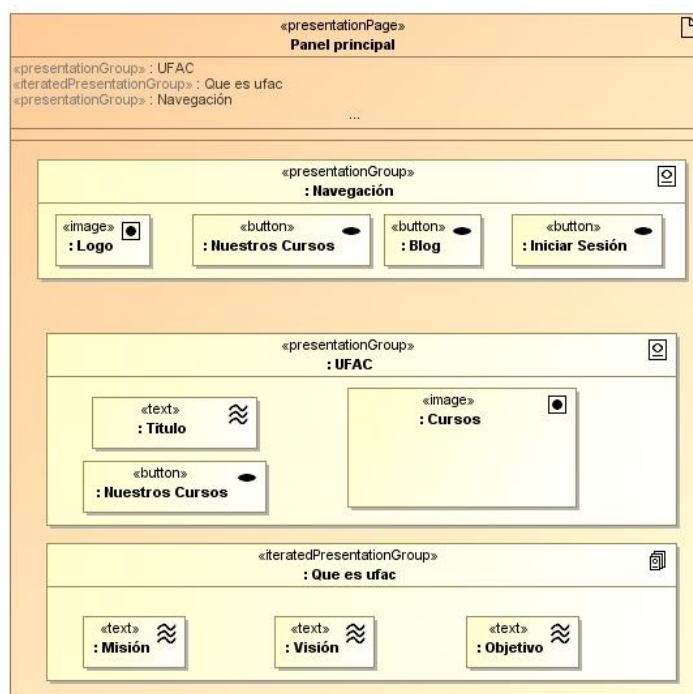




Figura 66.

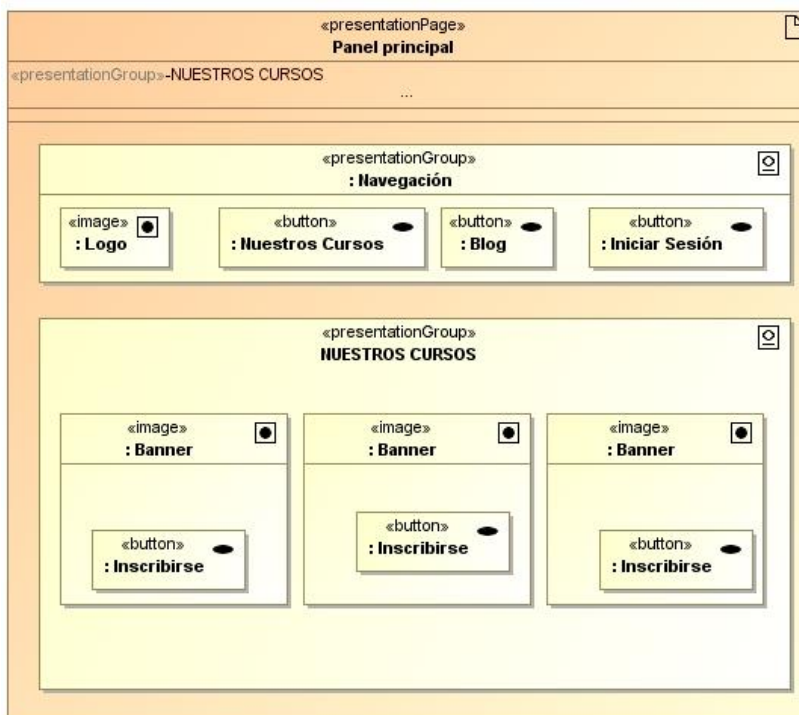
**Nuestros Cursos (usuario)**

Figura 67.

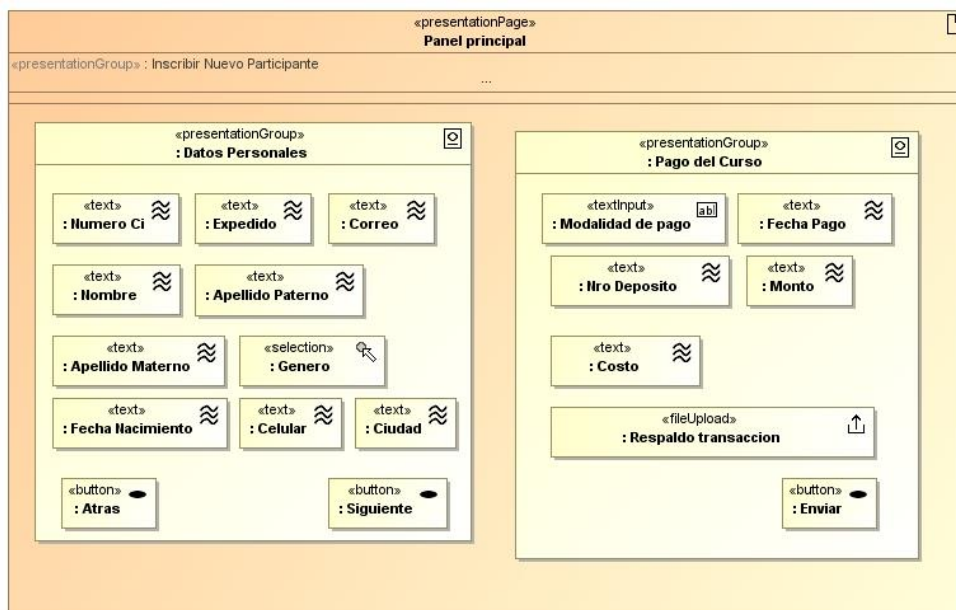
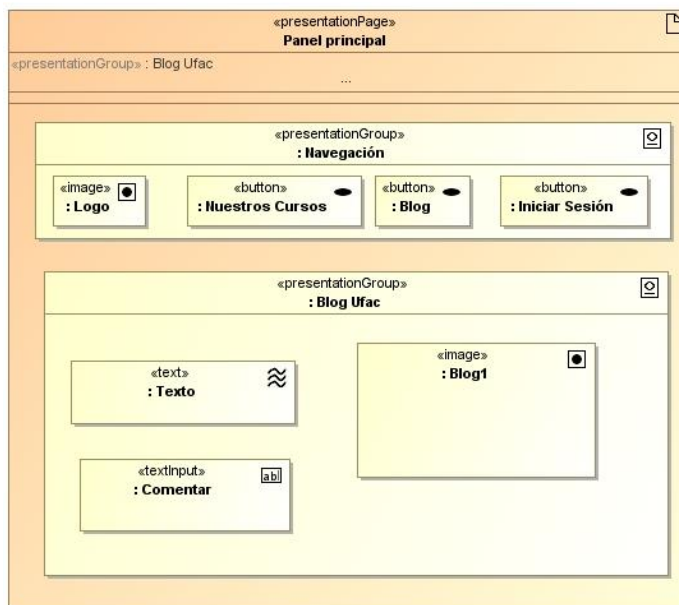
**Inicio de Inscripción (usuario)**

Figura 68.

**Inicio de Blog (usuario)****3.2.4.3 Diagrama de Actividades.**

A continuación, se mostrará el modelado del flujo de los procesos más importantes del sistema para poder comprender la lógica del funcionamiento de las actividades y procesos que realiza el sistema.

Figura 69.

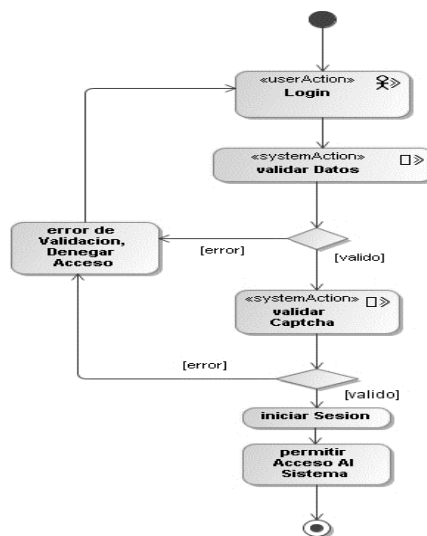
**Diagrama de Actividad del Inicio de Sesión del Sistema**

Figura 70

Diagrama de Actividad para la Pre-inscripción a un Curso

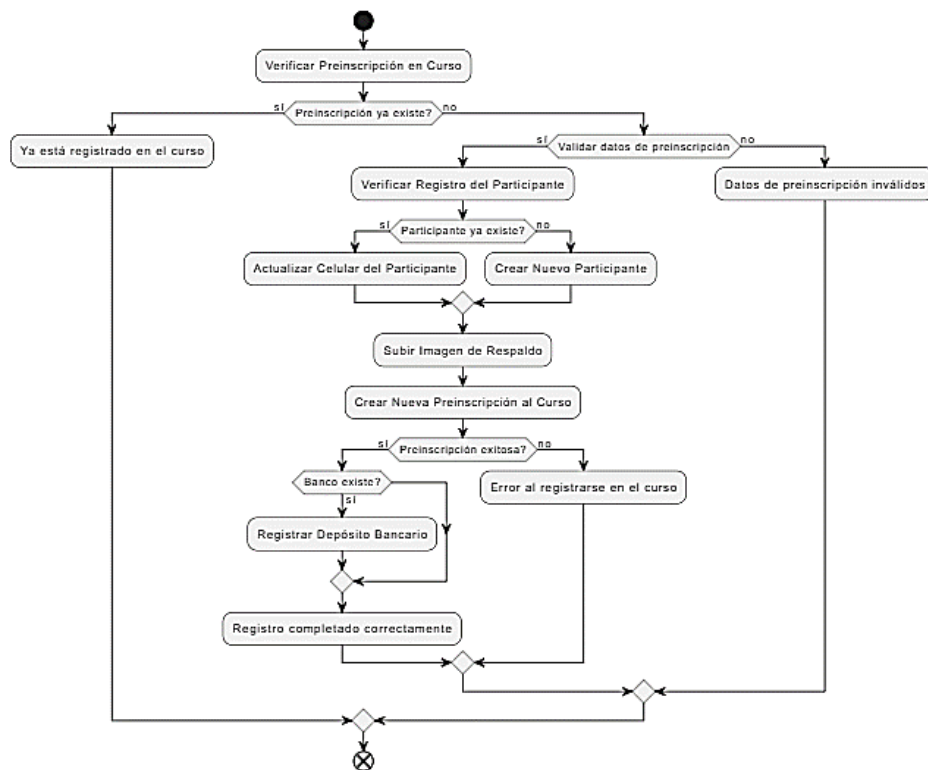
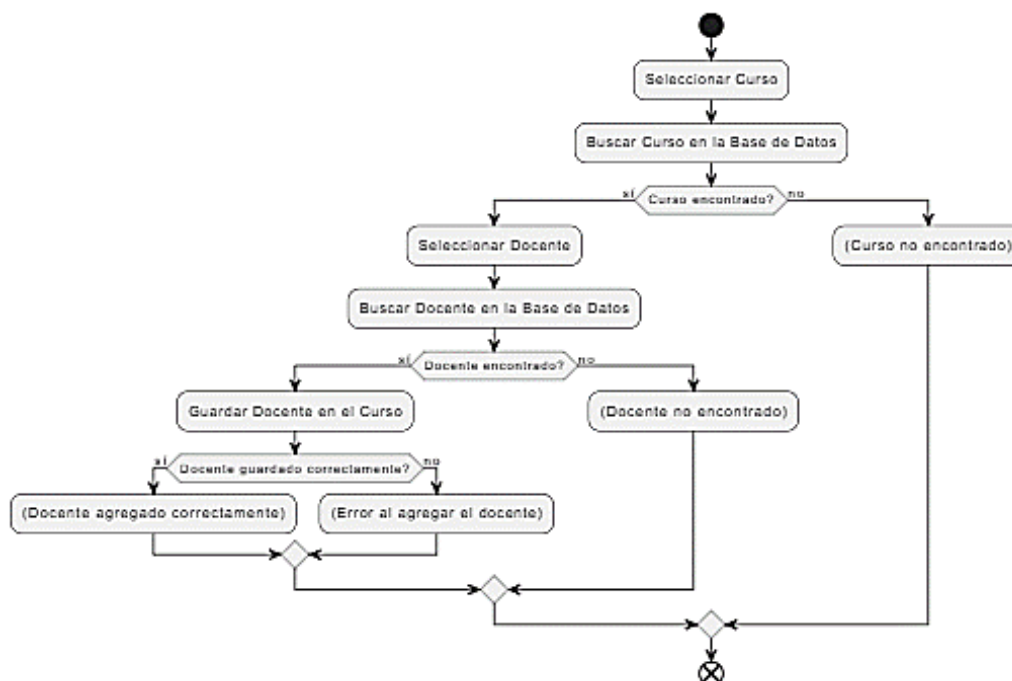


Figura 71

Diagrama de Actividad Para Agregar Docente a un Curso



### 3.3 SCRUM DIARIO

#### 3.3.1 *Comunicación de Sprint Backlogs*

Para poder comunicar el avance de cada uno de los requisitos que se definieron en el análisis de requerimiento, se realizan los scrum diarios o reuniones diarias donde participan el Scrum Master y el Desarrollador principalmente para verificar y evaluar el avance realizado de las tareas asignadas. La finalidad de ello es que ninguna tarea sea un cuello de botella que impida la culminación del proyecto.

Sobre un tablero se vieron los avances del día y los problemas que se tuvo y que se realizara mañana las reuniones con el scrum master no superaron los 20 minutos diarios.

Para la participación del sistema se tiene un equipo:

- **Product Owner:** Es el encargado del área que en este caso es el Lic. Zenón Chuquimia, responsable de la unidad de formación académica continua.
- **Scrum Master:** Es el que asegura de que las etapas de Scrum se lleven a cabo, el encargado de este rol es el Ing. Walter Emilio Paco Siles
- **Desarrollador:** Se encarga de realizar todo el proceso de creación del sistema, en este caso el desarrollador Sergio Luis Yujra Torrez

Para la participación de los scrums diarios solo conformas el scrum master y el desarrollador.

**Figura 72.****Tablero de Tareas**

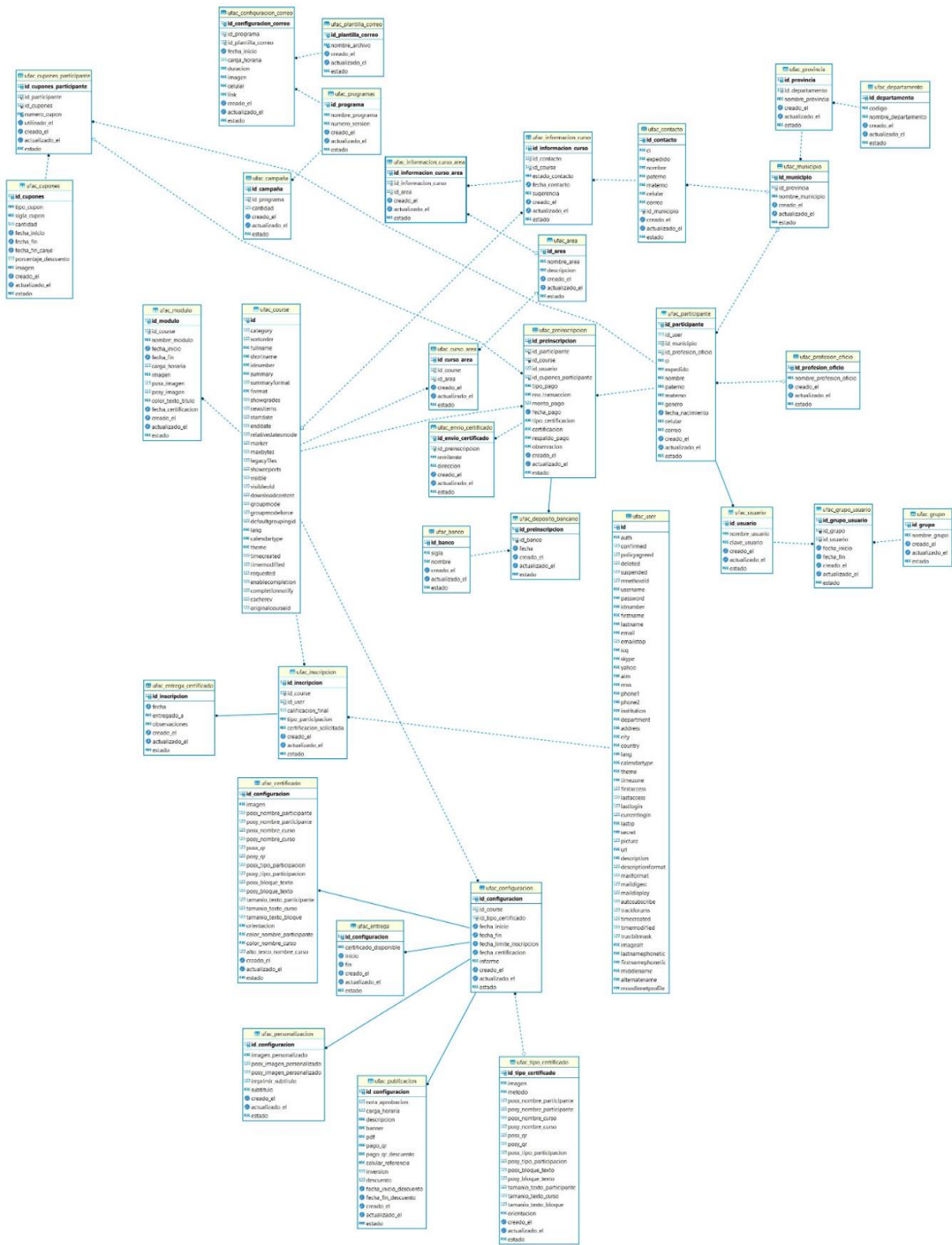
The screenshot shows a Kanban board for 'ufac' with three columns: 'Things To Do' (5 tareas), 'Doing' (3 tareas), and 'Terminado' (5 tareas). The 'Things To Do' column contains five tasks: 'Inicio con login', 'listado de cursos', 'certificados de particioantes', 'agregar docentes a un curso', and 'qr en los certificados', plus an 'AGREGAR NUEVA TAREA' button. The 'Doing' column contains three tasks: 'live en vivo', 'blog', and 'reportes', plus an 'AGREGAR NUEVA TAREA' button. The 'Terminado' column contains five tasks, each preceded by 'TERMINADO RECIENTEMENTE': 'listado de docentes', 'reportes de cursos', 'configuracion de cursos', 'analisis de requerimiento', and 'login'.

**3.4 REVISIÓN DEL SCRIPT Y LA IMPLENTACIÓN****3.4.1 Planificación de Entregas**

Se basan inicialmente en el product backlog que es la cantidad de necesidades del sistema como así también en el diseño del sistema que es la fase dos de uwe donde se visualizara lo acordado en el diseño para la implementación del sistema.

**3.4.2 Diseño de la Base de Datos**

**Figura 73.**  
**Modelo físico de la base de datos**

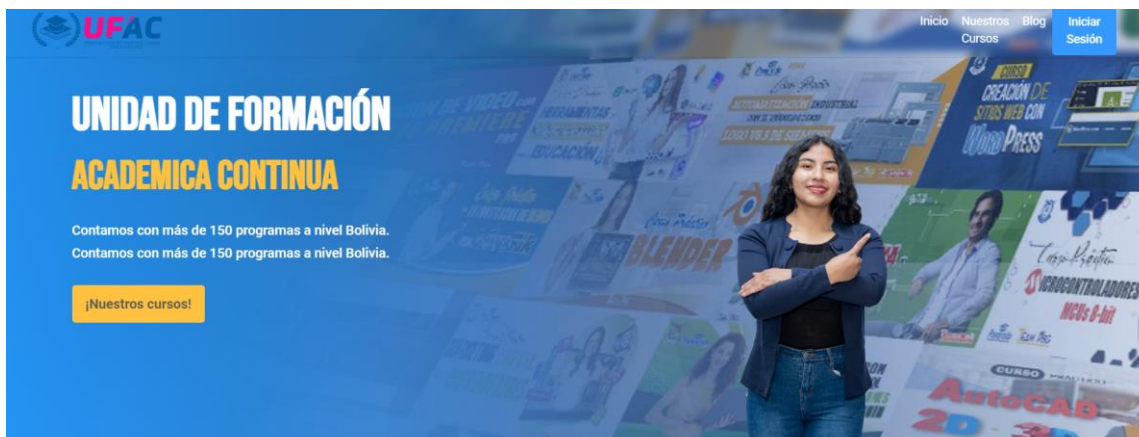


### 3.4.3 Diseño de Administración

#### 3.4.3.1 Vista de Inicio de Sesión.

Figura 74.

#### Inicio de Sesión



Que es la unidad de formación academica continua



Figura 75.

#### Login de la Pagina

The image shows the login page of the Ufac website. The page has a light gray background. In the center, there is a white box containing the login form. At the top of the box, it says 'Bienvenido al Sistema' and the Ufac logo. Below this, it says 'INGRESE SUS DATOS PARA EL INICIO DE SESIÓN'. There are three input fields: 'username', 'Password', and a captcha field with the text 'Ingrese el captcha'. The captcha field has a 'Wtk3N' logo. At the bottom of the box, there is a blue button that says 'Ingresar'.

Figura 76.

## Muestra de Código (inicio de sesión)

```

<div class="card pt-2">
  <div class="card-body">
    <div class="row">
      <div class="d-flex justify-content-center">
        <div class="auth-header">
          <h2 class="" style="color: #626262;"><b>Bienvenido al Sistema</b></h2>
          
          <p class="f-16 mt-2" style="font-weight: 600;">INGRESE SUS DATOS PARA EL INICIO DE SESIÓN</p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="form-floating mb-3">
  <input type="text" class="form-control form-control-sm form-control-solid mt-1" name="username" id="username" placeholder="Nombre de usuario" />
</div>
<div class="form-floating mb-3">
  <input type="password" class="form-control form-control-sm form-control-solid mt-1" name="password" id="password" placeholder="Contraseña" />
</div>
<div class="fv-row mb-2 d-flex align-items-center">
  <button class="btn btn-primary" id="btn-refresh" type="button" style="padding: 10px;">
    <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-arrow-repeat" viewbox="0 0 16 16">
      <path d="M11.534 7h3.932a.25.25 0 0 1 .192.411-1.966 2.36a.25.25 0 0 1-.384 0l-1.966-2.36a.25.25 0 0 1 .192-.411a.25.25 0 0 1-.192-.411" />
    </svg>
  </button>

```

## 3.4.3.2 Vista de la Administración de un Curso.

Figura 77.

## Administrar Curso (rol de administrador)

Inicio

- Inicio

Cursos

- cursos**
- configuración

Docentes

- agregar
- Listado de docentes

Certificado

- entrega
- agregar tipo de certificado

Listado de todos los cursos ufac

Show: 10 entries Search:

#	NOMBRE DEL CURSO	ALIAS	CERTIFICADO	INSCRITOS	PREINSCRITOS	CONTACTAR	F/CREACION	INICIO
3	prueba 2	prueba		0	0		30/04/2024 Inicio: 30/04/2024 Fin: 30/09/2024	30/04/2024
1	moodleufac	moodle		0	0		18/04/2024 Inicio: Fin:	
2	pruebadecurso	pruebadecurso		4	3		18/04/2024 Inicio: 18/04/2024 Fin:	18/04/2024

Showing 1 to 3 of 3 entries

Previous 1 Next



Figura 78.

## Muestra de Código (Administrar cursos)

```

<?= $this->section('css') ?>
<link href="<?= base_url('assets/plugins/custom/datatables/datatables.bundle.css') ?>" rel="stylesheet" type="te
<?= $this->endSection() ?>

<?= $this->section('content') ?>
<div class="row gy-5 g-xl-8">
  <div class="container-fluid">
    <!--begin::Card-->
    <div class="card card-custom">
      <div class="card-header">
        <div class="card-title">
          <h3 class="card-label">
            Listado de todos los cursos ufac You, last month * inicio de proyecto
          </h3>
        </div>
      </div>
      <div class="card-body">
        <table class="table table-separate table-head-custom table-checkable" id="tbl_cursos" style="width:
        <thead>
          <tr>
            <th>#</th>
            <th>nombre del curso</th>
            <th>alias</th>
            <th>certificado</th>
            <th>inscritos</th>
            <th>preinscritos</th>
            <th>contactar</th>
            <!-- <th>módulos</th> -->
            <th>f/creacion</th>
            <th>Inicio</th>

```

## 3.4.3.3 Vista de Certificación.

Figura 79.

## Administrar Certificación (rol de administrador)

The screenshot shows the 'Listado de configuración' (Configuration List) for certificates in the UFAC system. The interface includes a sidebar with navigation options like 'Inicio', 'Cursos', 'Docentes', and 'Certificado'. The main content area displays a table with columns for ID, Course Name, Short Name, Certification Status, Start Date, End Date, Limit Date, Certificate Status, and Grade. Three entries are visible, each with a certificate icon and a grade value.

#	CURSO	NOMBRE CORTO	CERTIFICACIÓN	FECHA_INICIO	FECHA_FIN	FECHA_LIMITE	CERTIFICADO	NOTA
181	prueba 2	prueba	30-04-2024	30-05-2024	29-04-2024	sin fecha certificación		75
180	pruebadecurso	pruebacurso	18-04-2024	10-08-2024	10-08-2024	18-08-2024		65
179	moodleufac	moodle	sin fecha inicio	sin fecha fin	sin limite inscripción	sin fecha certificación		65

Showing 1 to 3 of 3 entries

Previous 1 Next

Figura 80.

## Muestra de Código (Administrar certificados)

```

<?= $this->extend('base') ?>

<?= $this->section('title') ?>
Entrega de Certificados
<?= $this->endSection() ?>

<?= $this->section('subtitle') ?>
Entrega de Certificados
<?= $this->endSection() ?>

<?= $this->section('css') ?>
<link href="<?= base_url('assets/plugins/custom/datatables/datatables.bundle.css') ?>" rel="stylesheet" type="te
<?= $this->endSection() ?>

<?= $this->section('content') ?>
<div class="row gy-5 g-xl-8">
  <div class="container-fluid">
    <!--begin::Card-->
    <div class="card card-custom">
      <div class="card-header">
        <div class="card-title">
          <h3 class="card-label"> You, last month + inicio de proyecto
            listado de entrega de certificados
          </h3>
        </div>
      </div>
      <div class="card-body">
        <table class="table table-separate table-head-custom nowrap table-checkable" id="tbl_entrega_cer
          <thead>

```

## 3.4.3.4 Vista de Inscripción.

Figura 81.

## Administrar Inscripción y Preinscripción (rol de administrador)

The screenshot displays the UFAC administrative interface. On the left is a sidebar menu with categories: Inicio, Cursos, Docentes, and Certificado. The main content area is titled 'DATOS PERSONALES.' and contains the following form fields:

- Número C.I. (\*)**: A text input field with the placeholder 'Tu respuesta'.
- Expedido \***: A dropdown menu with the placeholder 'Elige'.
- Dirección de correo electrónico (\*)**: A text input field with the placeholder 'Tu dirección de correo electrónico'.
- Nombre(s) (\*)**: A text input field with the placeholder 'Tu respuesta'.
- Apellido Paterno**: A text input field with the placeholder 'Tu respuesta'.
- Apellido Materno**: A text input field with the placeholder 'Tu respuesta'.
- Género**: Radio buttons for 'Masculino' (selected) and 'Femenino'.
- Fecha de Nacimiento (\*)**: Three dropdown menus for selecting the day, month, and year.

Figura 82.

## Muestra de Código (Administrar inscripción)

```

You, 3 days ago | 1 author (You)
<?= $this->extend('base') ?>

<?= $this->section('title') ?>
Inscripción
<?= $this->endSection() ?>

<?= $this->section('subtitle') ?>
Inscripción Administración You, last month * inicio de proyecto
<?= $this->endSection() ?>

<?= $this->section('css') ?>
<link href="<?= base_url('assets/css/viewerjs/css/viewer.css') ?>" rel="stylesheet" type="text/css" />
<?= $this->endSection() ?>

<?= $this->section('content') ?>
<div class="row g-xl-8">
  <div class="container-fluid">
    <div class="card card-custom">
      <div class="card-body">
        <div class="wizard wizard-3" id="kt_wizard_v3" data-wizard-state="step-first" data-wizard-clicka
        <!--begin: Wizard Nav-->
        <div class="wizard-nav d-none">
          <div class="wizard-steps px-8 py-8 px-lg-15 py-lg-3">
            <!--begin: Wizard Step 1 Nav-->
            <div class="wizard-step" data-wizard-type="step" data-wizard-state="current">
              <div class="wizard-label">
                <h3 class="wizard-title">
                  <span>1.</span>Curso

```

## 3.4.3.5 Vista del Participante.

Figura 83.

## Administrar Participantes del Curso (rol de administrador)

UFAC

Inicio

Cursos

Docentes

Certificado

Listado de Preinscripciones

Show 10 entries

Descargar CSV

#	CI	PARTICIPANTE	CELULAR	CURSO	ESTADO	INSCRITO_POR	TIPO	BANCO	PAGO
6983	60634512 LP	VICTOR HUGO GUTIERREZ ZAPANA	73736972	pruebadecurso	PREINSCRITO		TIGO MONEY		190.00
6982	6035300 LP	WALTER EMILIO PACO SILES	63209954	pruebadecurso	INSCRITO	Sergio Yuja	TIGO MONEY		152.00
6981	11547715 LP	Sergio Yuja Torrez	73736972	pruebadecurso	INSCRITO		TIGO MONEY		152.00

Showing 1 to 3 of 3 entries

Previous 1 Next

**Figura 84.****Muestra de Código (Administrar participante)**

```

<?= $this->endSection() ?>





<?= $this->section('content') ?>
<div class="row gy-5 g-xl-8">
  <div class="container-fluid">
    <div class="card card-custom">
      <div class="card-header">
        <div class="card-title">
          <h3 class="card-label">
            Estado de participantes You, last month + inicio de proyecto
          </h3>
        </div>
      </div>
      <div class="card-body">
        <table class="table table-separate table-head-custom table-checkable" id="tbl_participante" styl
          <thead>
            <tr>
              <th>#</th>
              <th>ci</th>
              <th>nombres</th>
              <th>genero</th>
              <th>fecha_nacimiento</th>
              <th>celular</th>
              <th>correo</th>
              <th>municipio</th>
              <th>Estado</th>
              <th>Acciones</th>
            </tr>
          </thead>
        </tbody>
      </div>
    </div>
  </div>
</div>

```

**3.4.3.6 Vista del Blog.****Figura 85.****Administrar Blog (rol de administración)**

Listado de Posts Blog + Nuevo Post

Show 10 entries Search:

#	TITULO	IMAGEN
19	titulo 2	
20	titulo 3	
21	titulo 4	
22	holasihai sdias da sldasd	

Showing 1 to 4 of 4 entries < 1 >

Figura 86.

## Muestra de Código (Administrar blog)

```

: $this->section('content') ?>
.v class="row gy-5 g-xl-8"
<div class="container-fluid">
  <!--begin::Card-->
  <div class="card card-custom">
    <div class="card-header">
      <div class="card-title">
        <h3 class="card-label">
          Listado de Posts Blog          You, last week + blog
        </h3>
      </div>
    </div>
    <div class="card-toolbar">
      <button class="btn btn-primary font-weight-bolder" id="btn-agregar-post">
        <span class="svg-icon svg-icon-md">
          <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" wid
            <g stroke="none" stroke-width="1" fill="none" fill-rule="evenodd">
              <rect x="0" y="0" width="24" height="24" />
              <circle fill="#000000" opacity="0.3" cx="12" cy="12" r="10" />
              <path d="M11,11 L11,7 C11,6.44771525 11.4477153,6 12,6 C12.5522847,6 13,6.44771
            </g>
          </svg>
          Nuevo Post
        </button>
      </div>
    </div>
    <div class="card-body">
      <table class="table table-separate table-head-custom nowrap table-checkable" id="tbl_posts" style='
        <thead>

```


## 3.4.3.7 Vista de la Transmisión.

Figura 87.

## Administrar Live (rol de administración)

Listado de Comentarios

Show  entries Search:

#	COMENTARIO	FECHA CREACION	ACCIONES
149	asdadadsad	25/03/2024 19:56:40	
148	hola	31/10/2023 09:22:26	

Showing 1 to 2 of 2 entries < 1 >

**Figura 88.****Muestra de Código (Administrar live)**

```

<?= $this->section('content') ?>
<div class="row gy-5 g-xl-8">
  <div class="container-fluid">
    <!--begin::Card-->
    <div class="card card-custom">
      <div class="card-header">
        <div class="card-title">
          <h3 class="card-label">
            listado de Comentarios
          </h3>
        </div>
        <div class="card-toolbar">
          <div>
            <div class="card-body">
              <table class="table table-separate table-head-custom nowrap table-checkable" id="tbl_comments" s
                <thead>
                  <tr>
                    <th>#</th>
                    <th>Comentario</th>
                    <th>Fecha Creacion</th>
                    <th>acciones</th>
                  </tr>
                </thead>
              </table>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!--end::Card-->

```

**3.4.1 Diseño de Usuario****3.5 RETROSPECTIVA DEL SPRINT**

En esta etapa se debe realizar la retrospectiva de cada uno de los módulos definidos en la fase de planificación o diseño siempre y cuando el cliente y/o dueño del sistema establezca que el entregable proporcionado por el desarrollador no es lo que se solicitó al inicio del proyecto.

En el presente estudio las retrospectivas para los módulos fueron satisfactorias.

**Tabla 33****Tabla Del Estudio Módulos Fueron Satisfactorios**

Nro.	Modulo	Importancia	Como Probarlo	Estado Final
1	Inscripción a un curso	100	Llenando el formulario de inscripción	Realizado
2	Información a un curso	90	Solicitando información de un curso	Realizado
3	Blog Educativo	50	Cargando post a la página.	

4	Transmisión en Vivo	70	Cargando desde el lado del administrador una transmisión.	
5	Inicio de Sesión	100	Llenar los datos e ingresar	Realizado
6	Inscripción a un Participante	100	Registrando un participante a un curso	Realizado
7	Creación de un Curso	90	Creando un curso desde Moodle Y debería hacerse la vinculación	Realizado
8	Creación de un Certificado	90	Registrando y Modificando los modelos del certificado.	Realizado
9	Creación de un Docente	90	Buscando a un docente y asignándole a un curso.	Realizado
10	Asignación de nota	90	Darles nota a los participantes.	Realizado
11	Cambio de Contraseña	100	Los usuarios podrán cambiar su contraseña.	Realizado
12	Verificación de un certificado por QR	100	Con un lector de QR podrá verificar la autenticidad.	Realizado
14	Generación de reportes económicos	70	Ingresando a los cursos se podrá generar los reportes económicos	Realizado
15	Generación de reportes de participantes	70	Ingresando a la sección de preinscritos podremos generar para los participantes un reporte de inscripción	Realizado
16	Asignación de usuarios y roles	70	El administrador tiene el poder de asignar roles y permisos.	

La tabla muestra que el sistema permitió completar cada historia de usuario que se planteó al inicio del proyecto. Una vez levantadas cada una de las observaciones realizadas por el dueño del sistema al desarrollador, se procede a realizar el cierre del proyecto. Tarea que generalmente le corresponde al Scrum Master en coordinación con el encargado del quien administrara el sistema y los demás integrantes.

# CAPÍTULO IV



## **4 SEGURIDAD, CALIDAD, COSTO Y PRUEBAS DE SOFTWARE**

En este capítulo se vera la parte de seguridad y calidad del software según ISO 9126 y por último la estimación del software.

### **4.1 SEGURIDAD DE SOFTWARE**

Para mantener un buen nivel de seguridad en el software desarrollado se implementaron distintas medidas y métodos de seguridad que propone la ISO 9126 en su métrica de la funcionalidad, los cuales se describen a continuación:

- Autenticación, el sistema gestiona los accesos, verificando que solo usuarios autenticados puedan ingresar al mismo, esta medida evita el acceso a personal no autorizado.
- Encriptación, se utiliza la encriptación avanzada para las contraseñas de usuarios, esta medida proporciona una capa adicional de protección para a las credenciales de usuarios, procurando además de las contraseñas sean indescifrables, es decir que no puedan ser revertidas a sus caracteres originales.
- Confidencialidad, se han establecido medidas para que solo usuarios autorizados puedan acceder a sus módulos designados en base a sus Roles de Usuario.
- Integridad en los datos, Para asegurar que la información no se altere, ya sea de manera accidental o malintencionada, se han implementado protocolos y controles que validan y mantienen la coherencia y precisión de los datos en todo momento.

## 4.2 METRICA DE CALIDAD DEL SOFTWARE

El modelo de calidad del producto definido por la ISO 9126 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura.

### 4.2.1 *Funcionalidad*

Para realizar el cálculo de la métrica de funcionalidad del software se debe calcular el valor del punto de función que está dada por las siguiente formula:

( 11 ). *Ecuación Punto de Función*

$$PF = Cuenta\ Total \times (0.65 + 0.01 * \sum Fi) \quad (11)$$

Para lo cual primero hallaremos los valores de la “**Cuenta Total**” y la sumatoria **Fi** (valores de complejidad).

Para hallar la cuenta total se toman 5 características del dominio de la información, teniendo en cuenta:

- El Número de entradas de usuario
- El Número de salidas de usuario
- El Número de Peticiones de usuario
- El Número de archivos
- El Número de interfaces externas

Además, se debe seleccionar los factores de ponderación respecto al nivel de complejidad del sistema, para el cual en el presente sistema se hará uso del factor de ponderación con nivel de complejidad medio.

El cálculo de estos factores se realiza a continuación:

**Tabla 34.****Calculos de Valores de Ponderación**

Parámetros de Medición	Cuenta	Factor			Total
		Simple	Medio	Complejo	
Entradas del Usuario	56	x3	<b>x4</b>	x6	224
Salidas del usuario	73	x4	<b>x5</b>	x7	365
Peticiones del usuario	35	x3	<b>x4</b>	x6	140
Archivos	67	x7	<b>x10</b>	x15	670
Interfaces externas	6	x5	<b>x7</b>	x10	42
<b>Cuenta Total</b>					<b>1441</b>

En la tabla se muestra el resultado del cálculo de la cuenta total utilizando el factor de ponderación de complejidad.

Una vez obtenido el valor de la cuenta total, se realiza la determinación del ajuste de complejidad el cual se describe en la siguiente tabla:

**Tabla 35.****Tabla de Ajuste de Complejidad**

Importancia	0%	20%	40%	60%	80%	100%	Fi
Escala	No influencia	Incidencia	Moderado	Medio	Significativo	Esencial	Fi
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
1. ¿El sistema requiere copias de seguridad fiables?						Si	<b>5</b>
2. ¿Se requiere comunicación de datos?						Si	<b>5</b>
3. ¿Existen funciones de procesamiento distribuido?					Si		<b>4</b>
4. ¿Es crítico el rendimiento?						Si	<b>5</b>
5. ¿Se ejecuta en un entorno operativo existente y fuertemente utilizado?					Si		<b>4</b>

6. ¿Requiere entrada de datos interactiva?	Si	5
7. ¿Requiere la entrada de datos que las transacciones de entrada se lleven a cabo sobre variadas opciones?	Si	3
8. ¿Se actualizan los archivos maestros de forma interactiva?	Si	4
9. ¿Son complejos las entradas, las salidas, los archivos o peticiones?	Si	4
10. ¿Es complejo el procesamiento interno?	Si	4
11. ¿Utilización de código para ser reutilizable?	Si	5
12. ¿Están incluidas en el diseño la conversión y la instalación?	Si	3
13. ¿Se ha diseñado el sistema para soportar instalaciones en diferentes organizaciones?	Si	4
14. ¿Se ha diseñado la aplicación para ser fácilmente utilizada por el usuario?	Si	5
<b>TOTAL</b>		<b>60</b>

Calculando los puntos de función (PF) y reemplazando los valores hallados en la formula:

**( 12 ). Ecuación de Punto de Función**

$$PF = Cuenta\ Total \times (0.65 + 0.01 * \sum Fi) \quad (12)$$

Cuenta total = **1441**

$\sum Fi$  : Sumatoria de ajuste de complejidad = **60**

$$PF = 1441 \times (0.65 + 0.01 * 60)$$

$$PF = 1441 * 1.25$$

$$PF = 1801$$

Reemplazamos a la formula según el ajuste de complejidad máxima  $\sum Fi = 70$  para obtener el 100% del nivel de confianza.

$$PF_{max} = 1441 \times (0.65 + 0.01 * 70)$$

$$PF_{max} = 1441 * 1.35$$

$$PF_{max} = 1945$$

Ahora se utiliza la siguiente fórmula para calcular la funcionalidad real:

$$Funcionalidad = \frac{PF}{PF_{max}}$$

$$Funcionalidad = \frac{1801}{1945}$$

$$Funcionalidad = 0.925 * 100 = \mathbf{92.5\%}$$

Según el valor obtenido, la funcionalidad del presente proyecto es **92.5%**, esto da a entender que el sistema tiene un **92,5%** de realizar sus funciones sin riesgo de fallar con operatividad constante y un **7.5%** de que el sistema tenga un colapso y deje de funcionar.

#### 4.2.2 **Confiabilidad**

Para realizar el cálculo del nivel de confiabilidad en el sistema se deben tomar en cuenta la cantidad de fallas que se producen en un tiempo determinado de funcionamiento **F(t)**, para hallar **F(t)** se hace uso del valor hallado durante el cálculo de la funcionalidad del sistema realizado y se utiliza en la siguiente formula:

#### ( 13 ). **Ecuación de Tiempo de Funcionamiento**

$$F(t) = f * e^{-u*t} \tag{13}$$

Considerando un periodo de 15 días como tiempo de prueba donde se obtiene que en cada 10 ejecuciones y tareas resultan en 1 falla.

Se obtienen los siguientes datos:

**f**: Funcionalidad del sistema = 0.925

**u**: probabilidad de errores en el sistema = 1/10 = 0.1

**t**: tiempo de prueba que se le realizo al sistema = 15 días

calculando y reemplazando en la fórmula de ecuación de tiempo de funcionamiento:

$$F(t) = 0.925 * e^{-0.1*15}$$

$$F(t) = 0.925 * e^{-1.5}$$

$$F(t) = 0.206$$

Tenemos que el tiempo de funcionamiento es de un **0.206**

Ya que logramos conseguir el tiempo de funcionamiento posteriormente se calcula la confiabilidad reemplazando el valor hallado en la formula:

**( 14 ). Ecuación de Confiabilidad**

$$P(T \leq t) = 1 - F(t) \tag{14}$$

$$P(T \leq t) = 1 - 0.206$$

$$P(T \leq t) = 0.794 = 80\%$$

Para finalizar la confiabilidad del sistema es de un 80% en el tiempo de pruebas de 15 días.

#### **4.2.3 Usabilidad**

A continuación, se realiza el cálculo de la usabilidad del sistema tomando en cuenta la facilidad de uso (FU) para los usuarios y su valoración hacia el sistema, esta se determina usando la formula:

**( 15 ). Ecuación de Facilidad de Uso**

$$FU = \sum \frac{Xi}{n} * 100 \tag{15}$$

Donde:

***Xi***: Es la sumatoria de Valores

***n***: Indica el número de preguntas

Las preguntas contienen una escala de valor definidas por la siguiente tabla:

**Tabla 36**  
**Escala de Valores de las Preguntas**

#	Escala	Valor
1	Muy Bueno	5
2	Bueno	4
3	Regular	3
4	Malo	2

**Tabla 37**  
**Preguntas para Determinar la Usabilidad del Sistema**

Nro.	Preguntas	SI	NO	Evaluación
1	¿El sistema brinda la confiabilidad de uso?	4	1	80%
2	¿Las operaciones del sistema son controladas?	4	1	80%
3	¿El sistema permite la retroalimentación de la Información?	4	1	80%
4	¿El sistema cuenta con una interfaz de usuario amigable y agradable a la vista?	5	0	100%
5	¿Las respuestas del sistema son satisfactorias?	5	0	80%
6	¿Las funciones del sistema le parecen complicadas?	3	2	60%
7	¿El sistema facilita el trabajo de los usuarios?	5	0	100%
8	¿Durante el proceso de funcionamiento, el sistema produjo errores?	0	5	100%
TOTAL				85%

En la tabla se muestra el resultado de las preguntas realizadas a 5 integrantes de la ufac para medir el nivel de usabilidad del sistema.

Con los datos obtenidos, calculamos la usabilidad del sistema:

$$FU = \sum \frac{Xi}{n} * 100$$

$$FU = \sum \frac{680}{8} * 100$$

$$FU = 85\%$$

Realizando un análisis del resultado obtenido, existe un 85 % de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

#### 4.2.4 *Mantenibilidad*

El mantenimiento de un sistema se realiza con el fin de cambiar o mejorar funcionalidades, en respuesta a los nuevos requerimientos que puedan surgir.

Para obtener los datos de madurez del sistema (IMS), se emplea la siguiente ecuación.

El IMS del software se puede calcular haciendo uso de la formula:

( 16 ). *Ecuación de Mantenibilidad*

$$IMS = \frac{Mt - (Fa + Fc + Fd)}{Mt} \quad (16)$$

**Tabla 38**

**Valores para Determinar la Mantenibilidad**

Descripción	Valor
<b>Mt:</b> Número de módulos de la versión actual	10
<b>Fc:</b> Número de módulos en la versión actual que se han modificado	1
<b>Fa:</b> Número de módulos en la versión actual que se han añadido	0
<b>Fd:</b> Número de módulos de la versión anterior que se han borrado	0

Reemplazando en la formula:

$$IMS = \frac{10 - (0 + 0 + 0)}{10}$$

$$IMS = 1 = 100\%$$



Por lo tanto, el sistema tiene un índice de estabilidad de 100% de facilidad de mantenimiento, y el restante 0% es el margen de error correspondiente a los cambios y las modificaciones.

#### **4.2.5 Portabilidad**

La portabilidad define la facilidad de llevar el sistema de un entorno a otro sin mucho esfuerzo y sin ningún problema. El presente proyecto está diseñado en un entorno de acceso vía Web y mide su portabilidad en el lado del Servidor y en el lado del cliente, para este tipo de sistema se toma en cuenta 3 aspectos:

- Acceso a internet.
- El equipo debe contar con un servidor Apache que ejecute PHP.
- Computadora con sistema operativo Windows o Linux, con base de datos MySQL.

Aplicando la formula:

#### **( 17 ). Ecuación Grado de Portabilidad**

$$GP = 1 - \frac{ET}{ER} \quad (17)$$

$$GP = 1 - \frac{5}{100}$$

$$GP = 0.95$$

Por tratarse de un sistema Web, el proyecto presenta una gran portabilidad, ya que los requerimientos para la implementación y ejecución del software son mínimos, por ellos a criterio propio se brinda un 95% de Portabilidad.

#### **4.2.6 Definición de Medición**

- Funcionabilidad = 92.5%
- Confiabilidad = 80%
- Usabilidad = 85%
- Mantenibilidad = 100%

- Portabilidad = 95%

**Promedio final:** 90.5%

### 4.3 ESTIMACIÓN DE COSTO

#### 4.3.1 COCOMO

A continuación, se realiza la estimación de costos del sistema aplicando del método de estimación de costos COCOMO intermedio en base a líneas de código.

Debido a que durante el desarrollo del software se utilizaron Frameworks y librerías, estos ya contienen código por defecto, por lo cual solo se tomaran en cuenta las líneas de código que se elaboraron.

**Tabla 39**

**Conversión de puntos de Función a KDLC**

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI BASIC	5	64
JAVA	6	53
PL/I	4	80
ANSI COBOL 74	3	107
ASP	9.00	36
PHP	11.00	29
VISUAL C++	9.50	34

*Nota.* Conversión de puntos de función a KLDC de lenguajes de programación, Salas, 2020

Aplicando la conversión y reemplazando en la fórmula para hallar KLDC:

**( 18 ). Ecuacion para Kilo Lines of Delivered Code**

$$KLDC = \frac{FLDC * PF}{1000} \quad (18)$$

El valor para PHP en un sistema mediano según la tabla de conversión de puntos de función que se muestra en la tabla 50 es 29 por lo tanto, se obtiene los siguientes datos:

**FLDC:** Factor de conversión LDC = 29

**PF:** Punto de Función: **1784** (hallado durante el cálculo de la funcionalidad)

Remplazando:

$$KLDC = \frac{29 * 1784}{1000} = 51.74$$

Después de haber realizado la sumatoria total de líneas de código desarrolladas en codeigniter 4 y otras librerías, se obtuvo el siguiente dato: Líneas de Código del Proyecto: 8527 utilizando la fórmula para hallar KLDC de COCOMO intermedio.

$$KLDC = \frac{8527}{1000}$$

$$KLDC = 8.527 = 8$$

Procederemos con la selección de atributos y cálculo del factor de ajuste FAE, para realizar el análisis de costo.

**Tabla 40**

**Tabla de atributos FAE**

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Software</b>						
Fiabilidad	0.75	0.88	1	1.15	1.4	
Tamaño de Base de Datos		0.94	1	1.08	1.16	
Complejidad del sistema	0.7	0.85	1	1.15	1.3	1.65
<b>Hardware</b>						
Restricciones de tiempo de ejecución			1	1.11	1.3	1.66

Restricciones de memoria virtual			1	1.06	1.21	1.56
Volatilidad de la máquina virtual	0.87		1	1.15	1.3	
Tiempo de Respuesta	0.87		1	1.15	1.3	
<b>Personal</b>						
Capacidad de análisis	1.46	1.19	1	0.86	0.71	
Experiencia en la aplicación	1.29	1.13	1	0.91	0.82	
Calidad de los programadores	1.42	1.17	1	0.86	0.7	
Experiencia en la máquina virtual	1.21	1.1	1	0.9		
Experiencia en el lenguaje	1.14	1.07	1	0.95		
<b>Proyecto</b>						
Técnicas actualizadas de programación	1.24	1.1	1	0.91	0.82	
Herramientas de software	1.24	1.1	1	0.91	0.83	
Restricción de tiempo de desarrollo	1.23	1.08	1	1.04	1.1	
			TOTAL =	0.64		

En la tabla se muestra el cálculo de los atributos FAE los cuales se multiplicarán para hallar el valor final.

En función de los valores seleccionados se procede al cálculo de la FAE, realizando la multiplicación de todos estos valores:

$$\text{FAE} = 0.64$$

#### 4.3.1.1 Esfuerzo requerido

Para el cálculo del esfuerzo requerido se utilizó los coeficientes de tipo simple que se muestran en la tabla de coeficientes debido a que este es el que mejor se acopla al caso.

Donde:

a: Coeficiente de COCOMO simple= 3.2

**b:** Coeficiente de COCOMO simple = 1.05

**c:** Coeficiente de COCOMO simple = 2.5

**d:** Coeficiente de COCOMO simple = 0.38

**FAE:** = 0.64

**KLDC** = 8

Reemplazando valores en la formula del esfuerzo:

**( 19 ). Ecuación para Esfuerzo**

$$E = a * KLDC^b * FAE \quad (19)$$

$$E = 3.2 * 8^{1.05} * 0.64$$

$$E = 18.1 \text{ meses/hombre}$$

#### 4.3.1.2 Tiempo de desarrollo

A continuación, se calcula el tiempo de desarrollo con los siguientes datos obtenidos:

**c:** Coeficiente de COCOMO simple = 2.5

**d:** Coeficiente de COCOMO simple = 0.38

**E:** Esfuerzo = 18.1

Reemplazando valores en la formula del tiempo de desarrollo:

**( 20 ). Ecuación de Tiempo de Desarrollo**

$$TDES = c * E^d \quad (20)$$

$$TDES = 2.5 * 18.1^{0.38}$$

$$TDES = 7 \text{ meses} = 7 \text{ meses}$$

Se requiere un tiempo de desarrollo aproximado de **7 meses**.

#### 4.3.1.3 Cantidad de programadores

Teniendo los datos:

**E:** Esfuerzo = 20.5 meses/hombre

**TDES** = 7 meses

Reemplazamos en la formula

**( 21 ). Ecuación Para Personas**

$$P = \frac{E}{TDES} \quad (21)$$

$$P = \frac{20.5}{7}$$

$$P = 2.9 \text{ personas} = 3 \text{ personas}$$

Se requieren 3 personas para un tiempo de desarrollo de 7 meses.

#### 4.3.1.4 Costo del software

A continuación, se realiza el cálculo del costo total del software:

Sueldo Promedio entre programadores: 330 \$

**( 22 ). Ecuación para Costo del Software**

$$C_{sof} = \text{sueldo mes} * P * TDES \quad (22)$$

$$C_{sof} = 330 * 3 * 7$$

$$C_{sof} = 6930 \text{ $us}$$

Según los datos recopilados, se necesitarán tres programadores para un desarrollo de ocho meses. Se prevé que, durante este período, el costo del software alcanzará los 6930 dólares, lo que equivale a alrededor de 48.320 bolivianos.

## 4.4 PRUEBAS DE SOFTWARE

Se tratará de encontrar todo posible error durante un proceso antes que se entre en aplicación, mediante las siguientes pruebas realizadas.

### 4.4.1 Pruebas de Caja Blanca

Para aplicar las pruebas de caja blanca en este proyecto, se hará uso del método de camino básico, para el cual se hará uso de las líneas del código fuente extraídas de las funciones principales y más complejas del sistema.

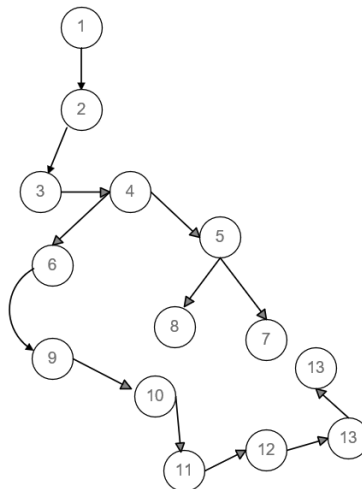
#### Figura 89

#### Autenticación Login

```

public function autenticar()
{
    @var \CodeIgniter\Validation\ValidationInterface $validation
    $validation = \Config\Services::validation();
    $validation->setRules([
        'username' => 'required',
        'password' => 'required',
    ]);
    if (!$validation->withRequest($this->request)->run()) {
        return redirect()->to(base_url(route_to('login')))->withInput()->with('errors', $validation->getEr
    }
    $nombreUsuario = $this->request->getPost('username');
    $claveUsuario = $clave = (string) $this->request->getPost('password');
    $usuarioEntidad = new UsuarioEntity(['nombre_usuario' => $nombreUsuario, 'clave_usuario' => $claveUsua
    // dd($usuarioEntidad);
    $usuario = $this->usuarioModel->obtenerUsuario($usuarioEntidad->nombre_usuario);
    if (!$usuario) {
        return redirect()->to(base_url(route_to('login')))->withInput()->with('error', 'Usuario no encontr
    }
    if (!password_verify($clave, $usuario['clave_usuario'])) {
        return redirect()->to(base_url(route_to('login')))->withInput()->with('error', 'Contraseña incorre
    }
    $persona = $this->usuarioModel->obtenerPersona($usuario['id_usuario']);
    $rol = new GrupoUsuarioModel();
    $roles = $rol->obtenerGrupoUsuario($usuario['id_usuario']);
    session()->set([
        'id_usuario' => $usuario['id_usuario'],
    ]
}

```

**Figura 90****Grafo de Autenticación de Usuario**

Posterior al diseño del grafo del flujo de la función para la autenticación e inicio de sesión de los usuarios, se puede determinar la complejidad ciclomática utilizando la fórmula respectiva y reemplazando los datos:

**A:** número de aristas del grafo = 14

**N:** número de nodos del grafo = 14

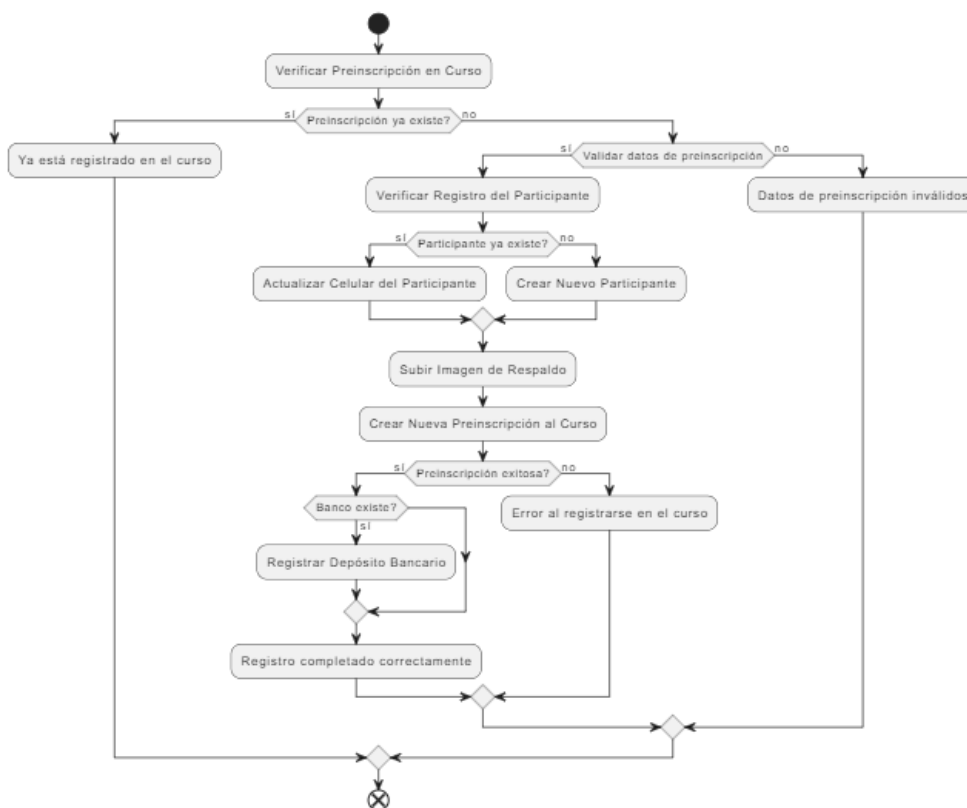
$$V(G) = A - N + 2$$

$$V(G) = 14 - 14 + 2 = 2$$

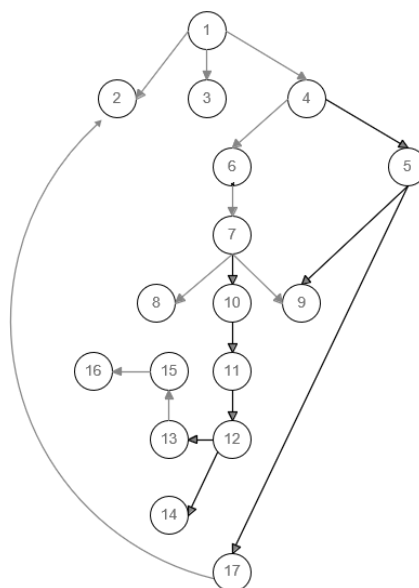
La complejidad ciclomática de la función es 2, determinando así que la función es de poco riesgo.



**Figura 91.**  
**Diagrama de Inscripción a Cursos**



**Figura 92.**  
**Grafo De Flujo Registro de Un Inscrito a un Curso**



Numero de nodos = 17

Numero de aristas = 19

Verificado el diagrama de registro de un participante a un curso y generando el grafo, ahora se procede a determinar la complejidad ciclo matica mediante la siguiente formula:

**( 23 ). Cálculo de la Complejidad**

$$(G) = A - N + 2 \quad (23)$$

(G): complejidad ciclo matica del grafo

A:(Aristas)

N:(Nodos)

Por tanto:

$$(G) = 19 - 17 + 2$$

$$(G) = 4$$

La complejidad del ciclo matica de la función es de 4, determinando así que la función es de poco riesgo.

#### **4.4.2 Pruebas de Caja Negra**

Para realizar las pruebas de caja negra se hará uso del método de las tablas de decisión, donde se podrá observar las salidas y resultados que nos devuelve el sistema.

**Figura 93.****Formulario de Autenticación de Usuarios**

Bienvenido al Sistema

**UFAc**  
UNIVERSIDAD FRANCISCA DE ASIS Y SANTO DOMINGO  
PROFESORADO UFAc

INGRESE SUS DATOS PARA EL INICIO DE SESIÓN

username

Password

 Ingrese el captcha

Ingresar

**Tabla 41****Tabla de decisión caso de prueba de Caja Negra del Login del Sistema**

Entradas	R1	R2	R3	R4	R5	R6	R7	R8
Usuario	V	V	V	V	F	F	F	F
Contraseña	V	V	F	F	V	V	F	F
Captcha	V	F	V	F	V	F	V	F
Resultado								
Permitir Acceso al sistema	SI	NO	NO	NO	NO	NO	NO	NO

**Figura 94.**  
**Registro a un Curso**

**FORMULARIO DE INSCRIPCIÓN AL CURSO: MOODLE**  
Para confirmar la inscripción llene los siguientes datos personales

[Volver](#)

INGRESE SUS DATOS PERSONALES EN EL SIGUIENTE FORMULARIO

Cedula de Identidad \*  
Solamente número sin expedido  Expedido

Nombre(s) \*  Apellido Paterno  Apellido Materno

Dirección de correo electrónico \*  Género \*  Masculino  Femenino

Fecha de Nacimiento \*  
Año  Mes  Día

Numero de celular \*  Ciudad de residencia \*

[Siguiente >](#)

**Tabla 42**  
**Registro a un Curso**

Campos	Entrada Valida	Entrada Invalida
Cedula Identidad	cadena de texto	No puede ir vacío
Nombres	cadena de texto	No puede estar vacío
Apellido paterno	cadena de texto	vacío
Apellido materno	cadena de texto	vacío
Correo	cadena de texto	No puede estar vacío
Genero	elección	No puede estar vacío
Fecha nacimiento	Fecha válida en formato DD/MM/AAAA	No puede estar vacío
Numero celular	Número válido de celular	No puede estar vacío
Ciudad	cadena de texto	No puede estar vacío
Modalidad de pago	elección	No puede estar vacío
Respaldo de transacción	selección	vacío
Numero deposito	Número decimal positivo	No puede estar vacío

		...//
Fecha pago	Fecha válida en formato DD/MM/AAAA	No puede estar vacío
Monto	Número decimal positivo	Caracteres especiales
Costo curso	Número decimal positivo	No puede estar vacío

*Nota.* En la tabla se detalla los campos, entradas y restricciones del formulario de registro de un participante a un curso.

**Tabla 43.**

**Tabla de Decisión Prueba de Caja Negra para el Registro de Activos**

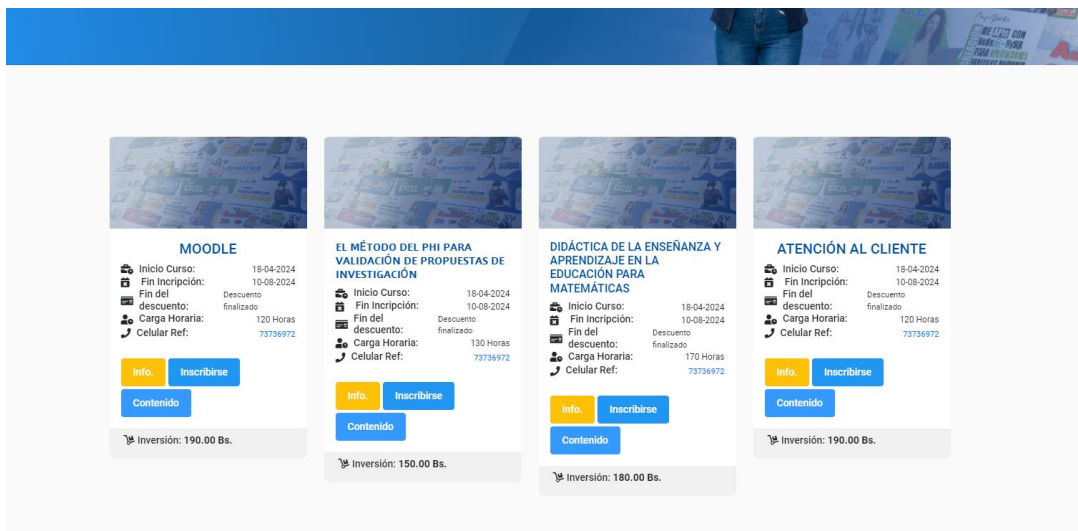
<b>Entradas</b>	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>
Datos Personales	V	V	F	F
Métodos de Pago	V	F	V	F
<b>Resultado</b>				
Enviar Registro de participante	SI	NO	NO	NO

*Nota.* En la tabla se muestran los resultados de pruebas de caja negra realizadas al formulario que se muestra, donde se validan los datos ingresados, y si estos resultan correctos se procede al registro del participante al curso.

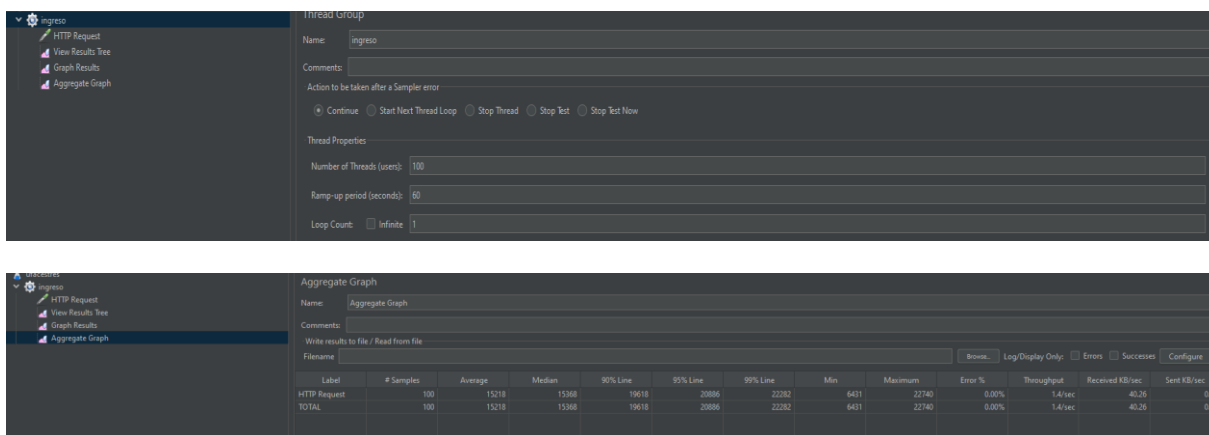
#### **4.4.3 Pruebas de Estrés**

El propósito de las pruebas de estrés es someter al sistema web a una carga extrema hasta alcanzar su límite de capacidad. Para llevar a cabo estas pruebas, se empleó la herramienta Apache JMeter 5.5. Los resultados obtenidos se presentan a continuación. Prueba de estrés de la página de inicio de los usuarios lectores:

**Figura 95.**  
**Prueba de estrés de la página de inicio de los usuarios**



**Figura 96.**  
**Calculando las Pruebas de Estrés en el Inicio de la Pagina de Usuarios**



Usuarios virtuales: 100, Tiempo de respuesta o rendimiento: 1.4/sec , Solicitudes realizadas Max: 22740, Error HTTP: 0

# CAPÍTULO V

## 5 CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

Una vez finalizado el proyecto de grado que titula “SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL” para la unidad de formación académica continua de la Universidad Pública de El Alto, se logró alcanzar los objetivos planteados, cumpliendo con las necesidades requeridas.

Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- El sistema web se desarrolló con éxito, alcanzando los objetivos de manera efectiva y utilizando los recursos de manera óptima.
- Se logró diseñar un sistema eficiente que brinde comodidad y facilidad para el personal encargado de la UFAC.
- Se logró elaborar el sistema para tener una información segura dentro de la base de datos.
- Se logró que el usuario pueda acceder al sistema con un determinado rol por medio de una autenticación con usuario y contraseña.
- La utilización de la metodología SCRUM Y UWE fue práctico para el desarrollar e implementar del sistema Web.
- Se logró mejorar el registro de los cursos en línea, se tiene las características los registros de los participantes para su posterior re marketing y así con estos datos mejorar la atención al usuarios y facilitación a los administrativos.
- Se logró disminuir los papeleos y la carga administrativa que generaba tanto en reportes de los participantes y también en la generación de certificados.
- Se logró que los usuarios puedan ver transmisiones gratuitas de eventos o cursos gratuitos donde interactúen entre sí.



- Se logro realizar la inscripción en línea para los participantes que opten en cursar un curso.

## **5.2 RECOMENDACIONES**

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

- Se recomienda realizar copias de seguridad de la base de datos periódicamente, para llevar una copia de respaldo.
- Se recomienda siempre contar con acceso a internet lo que permitirá poder estar con el sistema en línea.
- Realizar mantenimientos al sistema periódicamente al menos cada 6 meses para mantener la eficiencia del sistema.

## BIBLIOGRAFÍA

- Abud Figueroa, M. A. (2012). *Calidad en la industria del software, La norma ISO-9126*.
- ardilla, N. I. (12 de marzo de 2013). *Evaluación de Software*. Evaluación de Software: <http://actividadreconocimiento-301569-8.blogspot.com/2013/>
- Arroyo, D. (19 de junio de 2010). *Slideshare*. Retrieved 2 de septiembre de 2023, from <https://es.slideshare.net/david286/cocomo-basico>
- Bartholomew, D. (2014). *libro de MariaDB*. Packt Publishing.
- Beck, K. (24 de Abril de 1999). *Extreme Programming Explained*. [https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n\\_extrema&oldid=150750063](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_extrema&oldid=150750063)
- Boehm, B. W. (1981). *Software Engineering Economics*. Prentice-Hall.
- Boehm, B., Abts, C., Brown, A., Chulani, S., Clark, B., Horowitz, E., & Reifer, D. J. (2000). *Software cost estimation with COCOMO II*. Prentice Hall PTR.
- Booch, G., Rumbaugh, J., & Jacobson, L. (2013). *The Unified Modeling Language User Guide*. Pearson Education.
- Burbeck. (11 de 05 de 1993). *The Early History of Smalltalk*. <http://worrydream.com/EarlyHistoryOfSmalltalk/>
- Busquelle, J. (2010). *Análisis de Puntos de Función*.
- Bustamante, P. (2021). *Estimación de esfuerzo de trabajo y planificación de proyectos de desarrollo con restricciones de recursos*. repositorio UCHILE: <https://repositorio.uchile.cl/>
- Calamani Salas, J. M. (2020). *Sistema de informacion para el registro y seguimiento academico*. Universidad Publica de el alto, Bolivia. <https://repositorio.upea.bo/jspui/bitstream/123456789/223/1/P.D.G-JHANNET%20MONICA%20CALAMANI%20SALAS.pdf>
- Calle, j. (2020). Retrieved 19 de octubre de 2021, from <http://repositorio.upea.bo/handle/123456789/181>
- Callisaya Laura, N. K. (2023). *Aplicaciones educativas virtuales en procesos de enseñanza en el Centro Educativo Mutua La Paz en la ciudad de El Alto distrito 2. (Tesis de Grado)*.

Universidad mayor de san andres, La paz El alto.  
<http://repositorio.umsa.bo/xmlui/handle/123456789/31594>

- Castillo, R. D. (2020). *SISTEMA DE EDUCACION VIRTUAL (E-LEARNING) COMO* .
- Cervantes Maceda, H., Velasco Elizondo, P., & Castro Careaga, L. (2016). *ARQUITECTURA DE SOFTWARE*. Mexico: CENGAGE Learning.
- Claudio. (4 de enero de 2020). *Medium*. Retrieved 2 de septiembre de 2023, from <https://medium.com/ux-ripley/c%C3%B3mo-medir-la-usabilidad-597c8fbb48f1>
- Coderhouse. (12 de abril de 2023). [https://www.coderhouse.cl/blogs/news/que-es-react-js-y-como-funciona?utm\\_term=&utm\\_campaign=2&utm\\_source=google\\_search&utm\\_medium=cpc&gclid=CjwKCAjwgqejBhBAEiwAuWHioMHxGeUmMHqrU9iZWC7dP-R-Zlq6h-oqCBSKfAp7bC8kRofJqdv6iBoCLSkQAvD\\_BwE](https://www.coderhouse.cl/blogs/news/que-es-react-js-y-como-funciona?utm_term=&utm_campaign=2&utm_source=google_search&utm_medium=cpc&gclid=CjwKCAjwgqejBhBAEiwAuWHioMHxGeUmMHqrU9iZWC7dP-R-Zlq6h-oqCBSKfAp7bC8kRofJqdv6iBoCLSkQAvD_BwE)
- Cusi Quispe, D. G. (2020). *APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS. Proyectos de Grado - Ingeniería de Sistemas*. universidad pública de el alto, La paz el alto. <http://repositorio.upea.bo/handle/123456789/190>
- Dano, J. (s.f.). *Academia EDU*. Retrieved 3 de septiembre de 2023, from [https://www.academia.edu/35223544/Pruebas\\_de\\_caja\\_negra](https://www.academia.edu/35223544/Pruebas_de_caja_negra)
- DeMarco, T. (2010). *Métodos de Estimación*. <http://www.lsi.us.es/docencia/get.php?id=326>
- Diaz, S. (2009). *Revista científica mundo de la investigacion y el conocimiento*. [https://doi.org/10.26820/recimundo/4.\(4\).octubre.2020.199-212](https://doi.org/10.26820/recimundo/4.(4).octubre.2020.199-212)
- Donovan, A., & Kernighan, B. W. (2015). *The Go Programming Language*. Addison-Wesley Professional.
- Dougiamas, M. (2003). *Association for the Advancement of Computing in Education*. <https://www.aace.org/>
- Eich, B. (1996). *Netscape's JavaScript, version 1.1*. Netscape Communications Corporation. Berners-Lee.
- Ezell, L. (2021). *CodeIgniter 4 Foundations*.
- Galiano. (2012). *Informe de metodologías aplicadas*. <https://elproyectedeluisgaliano.blogspot.com/2012/11/metodologia-uwe>

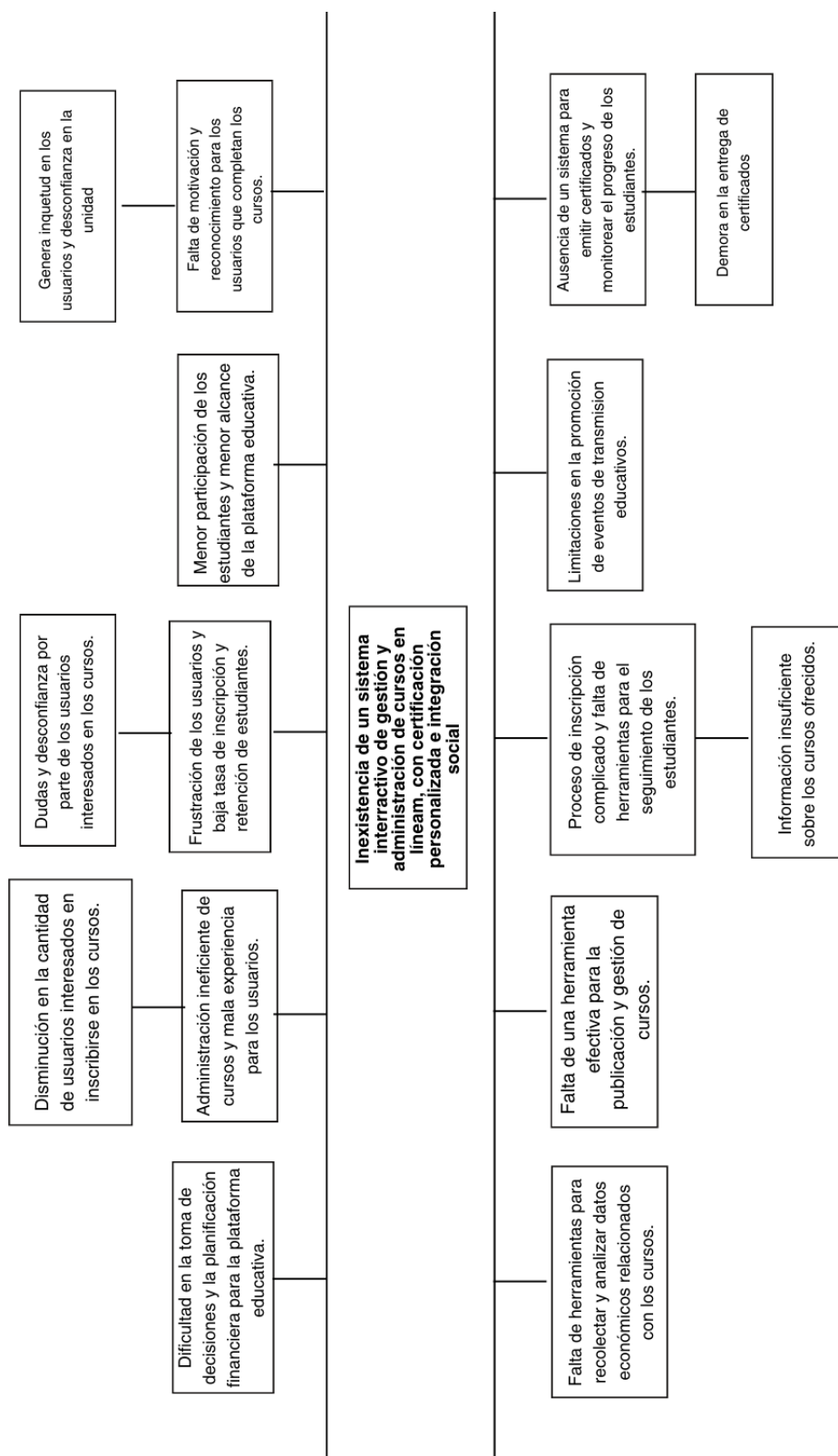
- Garcia, R., & Piattini, M. (2003). *Calidad En El Desarrollo y Mantenimiento del Software (Spanish Edition)*. Softcover.
- Gomez, A., Lopez, M., Migani, S., & Otazu, A. (2020). *COCOMO, Un modelo de estimacion de proyectos de software*.
- González, E. (2006). [https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=492:i-que-es-php-y-i-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:i-que-es-php-y-i-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193).
- Green, A. (02 de 11 de 2021). *GitMind*. <https://gitmind.com/es/metodologia-cascada.html>
- Informe de metodologías aplicadas*. (2012). <https://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe>
- ISO/IEC-27001. (2013). *Information technology, Security techniques, Information security management systems, Requirements*.
- ISO/IEC-9126:1. (2001). *Software engineering*. Product quality.
- Kaner, C., Falk, J., & Nguyen, H. Q. (2014). *Testing Computer Software*. John Wiley & Sons.
- Kimmel, P. (2007). *Manual de UML*. (H. Perez, Trad.) México: McGraw-Hill.
- Kniberg, H. (s.f.). *Scrum and XP from the Trenches*. Estados unidos: C4media.
- Lopez, M. R. (2016). *SCRIBD*. Retrieved 21 de septiembre de 2021, from METODOLOGIAS DE DESARROLLO WEB: <https://es.scribd.com/document/365638248/METODOLOGIAS-DE-DESARROLLO-WEB-pdf>
- Lopez, M. R. (2016). *SCRIBD*. <https://es.scribd.com/document/365638248/METODOLOGIAS-DE-DESARROLLO-WEB-pdf>
- Maida, E. G. (2015). *METODOLOGIAS DE DESARROLLO DE SOFTWARE*.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing*. John Wiley & Sons.
- Nieves Guerrero, C. (2018). UWE en Sistema de Recomendación de Objetos de Aprendizaje. *Revista Latinoamericana de Ingeniería de Software*.

- Norman, D. A. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*.
- Orosco Sullcata, A. A. (2020). PLATAFORMA PARA LA GESTIÓN DE CURSOS VIRTUALES. (*Proyecto de Grado*). Universidad pública de el Alto, El alto. <http://repositorio.upea.bo/handle/123456789/93>
- Palacio, J. (2014). *Scrum Manager II* (Scrum Manager® ed.). <https://www.safecreative.org/work/1404240651197?1>
- Paloma Díaz, S. M. (2005). *Ingeniería de la web y patrones de diseño*. España: Prentice Hall.
- Pressman , R. S. (2010). *Ingenieria del software un enfoque práctico*.
- Pressman, R. (2002). *Ingeniería del software*. IEEE Software, Steve McConnell [McC99] .
- Pressman, R. S. (2002). *Ingenieria del software*.
- Pressman, Roger. (2010). *Ingeniería del software*. Mexicana.
- Rodriguez Gonzales, P. (2008). *Estudio de la aplicacion de metodologias agiles para la evolucion de productos software*. Madrid, España.
- ROSSAINZ, M. (2016). *SCRIBD*. Retrieved 21 de septiembre de 2021, from METODOLOGIAS DE DESARROLLO WEB: <https://es.scribd.com/document/365638248/METODOLOGIAS-DE-DESARROLLO-WEB-pdf>
- Rouse, M. (05 de 11 de 2016). Whatis.com: <http://whatis.techtarget.com/definition/model-view-controller-MVC>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2006). *EL LENGUAJE UNIFICADO DE MODELADO MANUAL DE REFERENCIA*. (H. Castán Rodríguez, Ó. Sanjuán Martínez, & M. de la Fuente Alarcón, Trads.) España: Universidad Pontificia de Salamanca.
- Sanches Peño, J. M. (2015). *Pruebas de software*.
- Schrum, L. (1998). *On-line education: A study of emerging pedagogy. New Directions for Adult and Continuing Education*.
- Schwaber, K. (2016). *La guia del scrum*.
- Schwaber, K. (2016). *La guia del scrum*.
- Schwaber, K., & Sutherland, J. (2013). *La guia de scrum*.

- Schwaber, K., & Sutherland, J. (2017). *scrum.org*
- Schwaber, S. K. (2016). *La guía del scrum*.
- Simpson, K. (2015). *"You Don't Know JS" (No sabes JS, en español)*. O'Reilly.
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE* (9 ed.). (V. Campos Olguín, Trad.) México: Addison-Wesley.
- Stonebraker, M., & Roew, L. A. (1986). *The design of POSTGRES. Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data, 340-355*.
- Sutherland, J. (2016). *La guía del scrum*.
- Sutherland, J. (2016). *La guía del scrum*.
- Sutherland, K. S. (2016). *La Guía del Scrum*.
- TANAKA OHASHI, R. J. (2021). *PROCESO DE VIRTUALIZACIÓN DE ASIGNATURA*.
- TORRES, F. M. (2003). *DEFINICIONES DE LA INGENIERIA WEB*.
- Velasco, R. A. (2009). *aulas virtuales en la educación boliviana un nuevo paradigma*.
- ZEOKAT. (3 de diciembre de 2013). *Que es MariaDB y ventajas frente a MySQL*.  
<https://www.vozidea.com/que-es-mariadb-y-ventajas-frente-mysql>

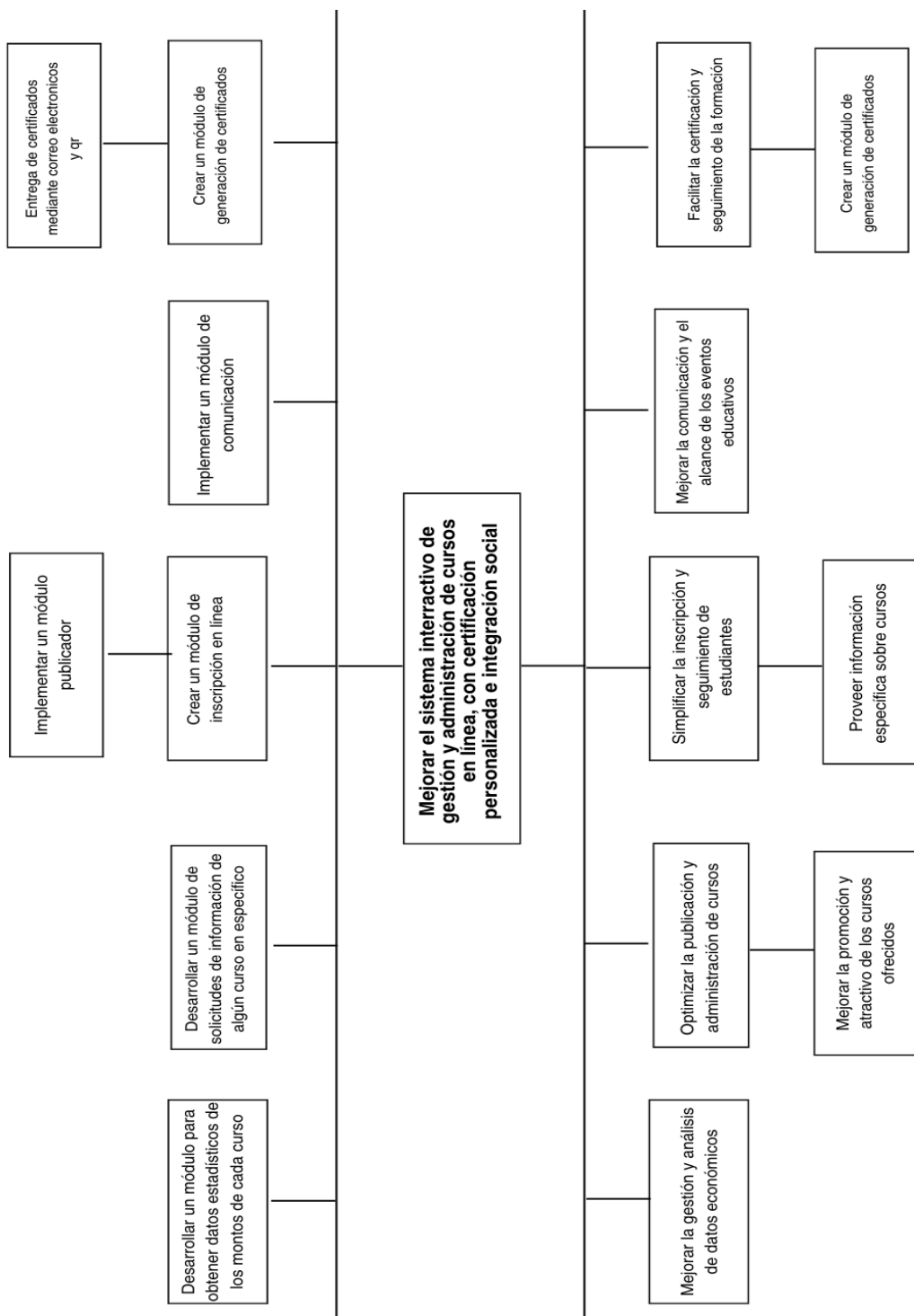
# ANEXOS

## Árbol de Problemas





## Árbol de Objetivos



## CUESTIONARIO DE CAPTURA DE REQUERIMIENTOS

### CUESTIONARIO PARA LA RECOLECCIÓN DE REQUERIMIENTOS UFAC

Este cuestionario contiene preguntas diseñadas para recopilar información esencial necesaria para analizar los requisitos del personal de la unidad de formación académica continua.

Nombre de la Institución: Unidad de formación académica continua

Encargado de la Unidad: RESPONSABLE UFAC Fecha: 15 FEBRERO 2023

1) ¿Actualmente cómo se está manejando la información dentro de la unidad?

DE MANERA INEFICIENTE, COMO SE PLANIFICA  
CADA LANZAMIENTO DE CADA CURSO

2) ¿Cómo se está gestionando el tema de marketing en los cursos?

AMAZON DE REDES SOCIALES, USANDO PRINCIPALMENTE  
EL FACEBOOK Y EL WHATSAPP.

3) ¿De qué forma se está realizando la inscripción de los participantes?

TENEMOS DOS FORMAS, UNA VIA DEPOSITO O  
TRANSFERENCIA Y OTRA DE FORMA PRESENCIAL  
EN OFICINAS UFAC.

4) ¿Se entrega certificación a los usuarios y cómo?

SE ENTREGA DE FORMA FISICA CON UN QR DE  
VERIFICACION DE ASISTENTES AL CURSO

5) ¿Qué desafíos enfrenta la unidad actualmente?

DE PODER ATRAPAR LA OPORTUNIDAD DE NUESTROS  
CURSOS Y ALCANZAR A PARTICIPANTES DE  
TODAS LAS REGIONES DEL PAIS.

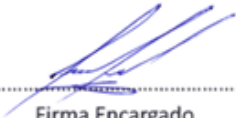
6) ¿Dónde se publica los cursos aparte de la página de Facebook?

EN OTROS ESPACIOS SOCIALES COMO WHATSAPP Y  
TIC DE LA INSTITUCIÓN.

#### CONSIDERACIONES

7) ¿Hay algún otro aspecto o requerimiento que desee agregar?

UN BLOQUE INFORMATIVO, TRANSFERENCIAS SU  
VINO ATRÁS DE LA PÁGINA DE FACEBOOK Y  
YOUTUBE.

  
Firma Encargado

## AVAL DE LA INSTITUCIÓN



*Universidad Pública de El Alto*

Creada por Ley 2115 del 5 de Septiembre de 2000 y Autónoma por Ley 2556 del 12 de Noviembre de 2003

El Alto, mayo de 2024

Lic. Ing. Helen Fanny Suntura Escobar

**TUTOR METODOLÓGICO**

**TALLER DE GRADO II**

Presente. –

### REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien de comunicarle mi conformidad del Trabajo de Grado:

TÍTULO: SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL.

CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, UNIDAD DE FORMACIÓN ACADÉMICA CONTINUA (UFAC).

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: SERGIO LUIS YUJRA TORREZ

REGISTRO UNIVERSITARIO: 200000128

CEDULA DE IDENTIDAD: 11547715

De tal forma cabe recalcar que el **SISTEMA** satisface los requerimientos de la institución, de esta forma se dio cumplimiento de los objetivos del presente.

El presente SISTEMA fue **IMPLEMENTADO** satisfactoriamente en la institución.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,

*Dr. Ricardo Jorge Torres Juaniquina Ph. D.*  
DIRECTOR DE POSGRADO  
UNIVERSIDAD PÚBLICA DE EL ALTO



**AVAL DE TUTOR METODOLÓGICO**

El Alto, mayo 2024

Señor:

Lic. Ing. William Roque Roque

**DIRECTOR DE CARRERA****INGENIERIA DE SISTEMAS**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido director de carrera:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de grado.

**TITULO: "SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL"****CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, UNIDAD DE FORMACIÓN ACADEMICA CONTINUA (UFAC)****MODALIDAD: PROYECTO DE GRADO****UNIV. SERGIO LUIS YUJRA TORREZ****REGISTRO UNIVERSITARIO: 200000128****CÉDULA DE IDENTIDAD: 11547715 LP**

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente.



.....  
Lic. Ing. Helen Fanny Suntura Escobar  
TUTOR METODOLOGICO  
TALLER DE GRADO II

**AVAL DE TUTOR REVISOR**

El Alto, mayo 2024

Señor:

Lic. Ing. Helen Fanny Suntura Escobar

**TUTOR METODOLOGICO****TALLER DE GRADO II**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

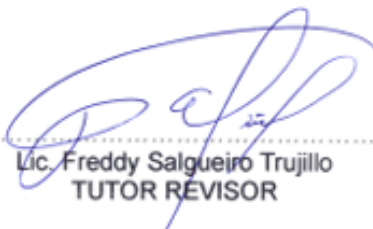
Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de grado.

**TITULO: "SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL"****CASO: UNIVERSIDAD PÚBLICA DE EL ALTO, UNIDAD DE FORMACIÓN ACADEMICA CONTINUA (UFAC)****MODALIDAD: PROYECTO DE GRADO****UNIV. SERGIO LUIS YUJRA TORREZ****REGISTRO UNIVERSITARIO: 200000128****CÉDULA DE IDENTIDAD: 11547715 LP**

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente.



.....  
Lic. Freddy Salgueiro Trujillo  
TUTOR REVISOR



**AVAL DE TUTOR ESPECIALISTA**

El Alto, mayo 2024

Señor:

Lic. Ing. Helen Fanny Suntura Escobar

**TUTOR METODOLOGICO****TALLER DE GRADO II**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de grado.

**TITULO:** "SISTEMA INTERACTIVO DE GESTIÓN Y ADMINISTRACIÓN DE CURSOS EN LÍNEA, CON CERTIFICACIÓN PERSONALIZADA E INTEGRACIÓN SOCIAL"**CASO:** UNIVERSIDAD PÚBLICA DE EL ALTO, UNIDAD DE FORMACIÓN ACADEMICA CONTINUA (UFAC)**MODALIDAD:** PROYECTO DE GRADO**UNIV.** SERGIO LUIS YUJRA TORREZ**REGISTRO UNIVERSITARIO:** 200000128**CÉDULA DE IDENTIDAD:** 11547715 LP

Para su defenza pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,



Ing. Walter Emilio Paco Siles  
TUTOR ESPECIALISTA