

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE GERENCIAL PARA LOS
PROCESOS ADMINISTRATIVOS DEL ÁREA DE POSTGRADO DEL NÚCLEO
DE ANZOÁTEGUI DE LA UNIVERSIDAD DE ORIENTE.”

Realizado Por:

Lugo R., Eliannys A.

Pino M., Maritzabel del V.

Trabajo de grado presentado como requisito parcial para optar al título de
INGENIERO EN COMPUTACIÓN

BARCELONA, Octubre de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE GERENCIAL PARA LOS
PROCESOS ADMINISTRATIVOS DEL ÁREA DE POSTGRADO DEL NÚCLEO
DE ANZOÁTEGUI DE LA UNIVERSIDAD DE ORIENTE.”

Asesor:

M.S.c. Francy Ríos
ASESOR ACADÉMICO

BARCELONA, Octubre de 2009

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DESARROLLO DE UN SOFTWARE GERENCIAL PARA LOS
PROCESOS ADMINISTRATIVOS DEL ÁREA DE POSTGRADO DEL NÚCLEO
DE ANZOÁTEGUI DE LA UNIVERSIDAD DE ORIENTE.”

Jurado Calificador

M.S.c. Francy Ríos

Asesor Académico

Ing. Claudio Cortínez .

Jurado Principal

Ing. Mónica Saettone, M.s.c

Jurado Principal

BARCELONA, Octubre de 2009

RESOLUCIÓN

De acuerdo con el artículo 44 del reglamento de trabajo de grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y solo podrán ser utilizados a otros fines solo con el consentimiento del consejo de núcleo respectivo, quien lo participara al Consejo Universitario”.

DEDICATORIA

Este trabajo, hecho con mucho esfuerzo y dedicación para culminar con una larga trayectoria de estudios y llegar a ser una profesional se lo dedico a mi familia especialmente a:

Mi Mami *Elia Rodríguez* gracias por todo tu amor, apoyo y confianza que siempre me has brindado, *TE AMO*.

Mi Padre gracias por tus palabras de aliento.

Mi Prima *Francis*, nos dejastes muy pronto pero sé que desde el cielo estas feliz por mi y disfrutando este éxito tanto como yo, te extraño un mundo “espaguettico”.

Mis Abuelos por siempre estar a mi lado y darme ánimo.

Mis Tías *Milagros, Esperanza, Alida, Nancy, Nelly, Rosa, Guadalupe, Fanny y Rosa Milagros* gracias por su apoyo y ayuda en todo momento; espero disfruten de este éxito.

Mis Tíos *Nelson, José G., Carlos, Julián, Ramón, Edgar y Orlando*, espero disfruten este éxito, gracias por su apoyo y palabras de aliento.

A *Rafael Grau* gracias por darme ánimo cuando más lo necesite, y estar allí siempre, simplemente gracias por ser mi segundo papá.

Mis Primos *Christian, Rosangela, Joselyn, Francis M., Adrian y Winston* espero que disfruten de este éxito y a la vez les sirva como ejemplo y motivación para luchar en la vida por lo que quieren recordando que los mejores logros requieren esfuerzo.

Mis Primas *Cristina, Neorly y Nelsy* simplemente gracias por estar a mi lado y apoyarme de alguna u otra manera en la culminación de esta tesis.

Eliannys A. Lugo Rodríguez

A Dios todo Poderoso y la Virgen del Valle, por estar en todo momento a mi lado y darme las herramientas necesarias para seguir adelante y lograr esta meta tan importante.

A Mis Padres *Agustín y Elis* por el apoyo y la confianza, por ayudarme a culminar la realización de este uno de mis sueños dándome la fortaleza necesaria para superar los obstáculos y seguir adelante.

A mi Abuela *Isabel* por darme ánimo en los momentos más difíciles haciéndome saber que los triunfos que obtenga en la vida serán mi mejor herencia y orgullo para ella.

A mis Hermanas *Elimar, Marielis y Karelys* espero que disfruten de este éxito y a la vez le sirva como ejemplo y motivación para luchar en la vida por lo que quieren recordando que los mejores logros requieren esfuerzo.

A mis Tíos a *Joel, Myhalis, Magalis, Andrés y Adonis* por su apoyo y ayuda en todo momento.

Maritzabel Del V Pino Martínez.

AGRADECIMIENTOS

Agradezco enormemente:

A mi *MAMI*, por apoyarme siempre y ayudarme a culminar la realización de éste, uno de mis sueños, dándome la fortaleza necesaria para superar todos los obstáculos y seguir adelante. Sin ti este logro no hubiese sido posible, Gracias madre... TE AMO DEMASIADO.

A mi amiga y compañera de tesis *Marit*, simplemente has sido la compañera perfecta, gracias por ser ante todo mi amiga, por entenderme y siempre estar allí tanto en las buenas como en las malas. Hicimos un excelente equipo, valió la pena tantos sacrificios y noches en vela que amerito este proyecto.... Lo logramos Mana!!!

A la Profesora *Francy Ríos*, por su asesoría y colaboración para el desarrollo de este trabajo de grado.

A los Profesores. *Claudio Cortínez* y *Yulitza Mujica* por su predisposición permanente e incondicional en aclarar las dudas y por sus substanciales sugerencias durante la redacción de la tesis.

A *Jean Carlos* por su contribución enorme en la realización de esta tesis.

A *Luisana* y *Angeles*, lo logre muchachas y no me cambie de carrera; no lograron convencerme (jeje), una amistad que surgió desde el inicio de nuestra carrera universitaria y espero que perdure por muchísimo tiempo.. Gracias por ser MIS AMIGAS!!

A *Maga* que te puedo decir, una excelente amistad a pesar de la distancia, sé que puedo contar contigo en todo momento y demás está decirte que cuentas conmigo.. QUE DERROCHE¡! Jajaja

A *Erika, Claudia y Lilibeth*, cuantas veces hablamos y soñamos con este momento, en culminar nuestra carrera universitaria; miles de veces; lo logre muchachas y así como yo pude, se que ustedes también; y allí estaré siempre para apoyarlas... ANIMO!!!

A *Enoes y Luis Cordero* (el chino); simplemente gracias por estar allí siempre, por brindarme su amistad incondicional, por estar conmigo tanto en las buenas como en las malas.... SE LES QUIERE MUCHO MIS LOCOS!!

A *Luis Martínez*; gracias por hacerme reír a carcajadas con tus ocurrencias, y en tus momentos lúcidos gracias por esas palabras de apoyo y ánimo. Todavía recuerdo la canción que me dedicaste cuando comenzamos con esta amistad, que loco, nunca cambies tu manera de ser, ni dejes que lo hagan.

A *Gazi El Halabi*, gracias por hacerme reír cuando más lo he necesitado, y animarme en todo momento, gracias por tu amistad y toda la ayuda que me has brindado siempre... Te Quiero Mucho Gazito!!

A *José Francisco*, gracias por ser mi amigo, por brindarme tu mano cuando lo he necesitado, por calarte mis lágrimas tantas veces y siempre animarme.. Te Quiero Mucho Loquito =).

A *Osmar* una amistad que surgió en clases de programación I, lo recuerdo como si fuera ayer, y esa linda amistad todavía considero que se mantiene sólida..

Sapito Gracias por ser mi Amiguito!!

A *Irene y Tarik*, gracias por su amistad muchachos. Ire gracias por todo tu apoyo y palabras de aliento; *Turquis* gracias por enseñarme a ser odiosa. Se Les Quiere!!

A *Franco* gracias por ayudarme cuando más lo he necesitado, y por hacerme saber que siempre puedo contar contigo en todo momento, recuerda que conmigo también cuentas, gracias por ser mi amigo... Te Quiero un Mundo mi Loco!!

A *Jovito, Roger y Jose Angel*, gracias por ser mis PANAS, y por su ayuda en los momentos que más lo necesite, gracias por estar allí siempre... Se les Quiere Mucho Muchachones!!

A *Marcel*, no recuerdo exactamente en qué momento nos hicimos tan buenos amigos, pero me alegra mucho contar con tu amistad, gracia por toda la ayuda que me has brindado siempre, TQM.

A *Raquel, Karla, Lidamel, Yelu y Rita*, mis amigas de la infancia gracias por todo su apoyo, espero que esta bonita amistad se mantenga SIEMPRE, sin importar los años que puedan pasar.. Se les Quiere Mucho Loquitas!

A *Raiza, Aquino y Lesmen*, una amistad un poco rara pero bonita, espero que esta loca amistad perdure por muuucho tiempo. Recuerden; nunca se den por vencidos mis locos así como yo pude, se que ustedes también pueden; y allí estaré siempre para apoyarlos... ANIMO!!!

A los tocayo A. *Paulo y A Zabala, Angel Ramos, Carlos Jimenez, Sergio Goschenko, Segnini, Luisaine, M Cristina*, gracias por su apoyo y ayuda a lo largo de mi carrera... Se Les Quiere y Aprecia Muchachos.

A *Cesar Llinas y Ricardo Lopez*; colegas, gracias por su amistad y apoyo, Se les Aprecia Mucho Fantasmas!!

A todos aquellos profesores a quienes les debo gran parte de mis conocimientos, gracias por su paciencia y enseñanza.

A todas aquellas personas que no mencione y que de alguna u otra forma, me ayudaron a lo largo de toda mi carrera, hago extensivo mi más sincero agradecimiento.

A los que *NO CREYERON* en mi, GRACIAS TOTALES!!

Eliannys A. Lugo Rodríguez

Agradezco a *Mis Padres, Hermanas, Abuela, Tíos* y demás *Familia*, por darme la estabilidad emocional, económica, sentimental; para poder llegar hasta este logro, que definitivamente no hubiese podido ser realidad sin ustedes.

A la Profesora *Francy Ríos*, por su asesoría y colaboración para el desarrollo de este trabajo de grado.

Al Profesor *Claudio Cortínez* y a la Profesora *Yulitza Mujica* por su predisposición permanente e incondicional en aclarar las dudas y por sus substanciales sugerencias durante la redacción de la tesis.

A mi amiga y compañera de tesis *Eliannys*, por su dedicación, responsabilidad, esfuerzo, que siempre estuvo allí tanto en las buenas como en las malas durante la elaboración de nuestro proyecto de tesis.

A *Manuel* que eres parte muy importante en mi vida, gracias por estar siempre allí con tu amor, cariño y comprensión apoyándome en todo incondicionalmente.

A *Daniel, Nancy, Jovito y Jean Carlos Díaz* por su apoyo y su ayuda en los momentos que más lo necesite.

A mis amigos, *Marcel, Jorge Luis, María Alejandra, Gabriela, Karina, Dayhana, Amieluz, Alberto y Norelkys* por brindarme su amistad a lo largo de este camino y compartir momentos buenos y malos.

A todos aquellos profesores a quienes les debo gran parte de mis conocimientos, gracias por su paciencia, enseñanza y finalmente un eterno agradecimiento a esta prestigiosa universidad la cual abre sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como personas de bien.

Y a todas aquellas personas que de una u otra forma, colaboraron o participaron en la realización de esta tesis y al que no mencione, hago extensivo mi más sincero agradecimiento.

Maritzabel Del V Pino Martínez.

ÍNDICE

RESOLUCIÓN	iv
DEDICATORIA	v
AGRADECIMIENTOS	vii
ÍNDICE	xii
ÍNDICE DE TABLAS	xviii
ÍNDICE DE FIGURAS	xx
RESUMEN.....	xxiii
CAPÍTULO I.....	24
EL PROBLEMA	24
1.1 Planteamiento del Problema	24
1.2 Propósito	26
1.3 Alcance	27
1.4 Importancia	28
1.5 Originalidad	28
1.6 Objetivos.....	28
1.6.1 Objetivo General.....	28
1.6.2 Objetivos Específicos	29
CAPÍTULO II	30
MARCO TEÓRICO.....	30
2.1 Antecedentes.....	30
2.2 Marco Conceptual.....	32
2.2.1 Ingeniería de Software [9]	32
2.2.1.1 Paradigma del Ciclo de Vida Clásico para la Ingeniería de Software	33
2.2.2 Proceso Unificado de Desarrollo de Software [13]	34
2.2.2.1 La Vida del Proceso Unificado.....	35

2.2.2.2 Fases dentro de un Ciclo del Proceso Unificado	35
2.2.2.3 Flujos de Trabajo en el Proceso Unificado.....	38
2.2.3 Lenguaje Unificado de Modelado (Unified Modeling Lenguaje, UML)	
[12]	41
2.2.4 Programación Orientada a Objetos [4]	43
2.2.4.1 Principios de los Lenguajes Orientados a Objetos	44
2.2.5 Bases de Datos [10]	45
2.2.5.1 Sistema de Gestión de Base de Datos (SGBD)	46
2.2.5.2 Modelo de Datos.....	47
2.2.6 HTML (Lenguaje de Etiquetas por Hipertexto) [7].....	48
2.2.6.1. Servidor Web	48
2.2.7 PHP (Hypertext Preprocessor).....	48
2.2.7.1 Características Fundamentales de PHP	49
2.2.8 ORACLE [1].....	50
2.2.8.1 Seguridad de la Base de Datos de Oracle	51
2.2.8.2 Oracle Tecnología Integrada	51
2.2.8.3 Oracle de Gestión de Contenidos	52
2.2.8.4 Oracle Arquitecturas Orientadas a Servicios (SOA)	52
2.2.8.5 Oracle Aplicación e Integración de Arquitectura	53
2.2.8.6 Oracle de Gestión de Identidades	53
2.2.8.7 Aplicabilidad de Oracle	54
2.2.9 Adobe Dreamweaver (Dw) [2]	56
2.2.9.1 Características de Adobe Dreamweaver (Dw)	56
2.2.10 JAVASCRIPT [3].....	57
CAPITULO III	58
FASE DE INICIO	58
3.1 Introducción.....	58
3.2 Planificación de la Fase de Inicio	59
3.3 Contexto del Sistema	60

3.3.1 Modelo de Dominio.....	60
3.3.2 Clases de Dominio.....	61
3.4 Identificación de Riesgos.....	63
3.5 Recopilar Requisitos.....	64
3.5.1 Enumerar los Requisitos Candidatos.....	64
3.5.1.1 Capturar Requisitos Funcionales.....	65
3.5.1.2 Capturar Requisitos No Funcionales.....	66
3.5.2 Modelo de Casos de Uso.....	67
3.5.2.1 Identificación de Actores.....	67
3.5.2.2. Identificación de Casos de Uso.....	69
3.5.2.3 Diagrama de Casos de Uso.....	69
3.5.2.4 Casos de Uso Detallados.....	71
3.6 Análisis.....	85
3.6.1 Análisis de la Arquitectura.....	85
3.6.1.1 Identificación de Paquetes de Análisis.....	85
3.6.2 Análisis de los Casos de Uso.....	89
3.6.2.1. Identificación de las Clases de Análisis.....	90
3.6.2.2. Diagrama de Clases de Análisis.....	96
3.6.2.3. Diagrama de Colaboración.....	100
3.7. Diseño.....	109
3.7. 1 Identificación de los Subsistemas.....	109
3.8 Planificación de las Sigüientes Fases.....	111
3.8.1 Fase de Elaboración.....	111
3.8.2 Fase de Construcción.....	112
3.8.3 Fase de Transición.....	112
3.9 Resumen de la Fase de Inicio.....	112
CAPÍTULO IV.....	114
FASE DE ELABORACIÓN.....	114
4.1 Introducción.....	114

4.2	Planificación de la Fase de Elaboración	115
4.3	Identificación de Riesgos.....	116
4.4	Captura de Requisitos.....	116
4.4.1	Lista de Requisitos.....	116
4.4.1.1	Requisitos Funcionales.....	117
4.4.1.2	Requisitos No Funcionales.....	117
4.4.2	Modelo de Caso de Uso.....	117
4.4.2.1	Identificación de Caso de Uso.....	117
4.4.2.2	Diagrama de Caso de Uso.....	118
4.4.2.3	Casos de Uso Detallados.....	118
4.5	Análisis.....	121
4.5.1	Análisis de la Arquitectura.....	122
4.5.1.1	Identificación de Paquetes de Análisis.....	122
4.5.2	Análisis de los Casos de Uso.....	123
4.5.2.1	Identificación de las Clases de Análisis.....	123
4.5.2.2	Diagrama de Clase de Análisis.....	126
4.5.2.3	Diagrama de Colaboración.....	127
4.5.3	Análisis de las Clases.....	136
4.6	Diseño.....	141
4.7	Diseño del prototipo de la Interfaz de Usuario.....	141
4.7.1	Prototipo de la Interfaz Inicio de Sesión.....	142
4.7.2	Prototipo de la Interfaz Principal.....	143
4.7.3	Modelo de Diseño.....	144
4.7.3.1	Identificación de las Clases de Diseño.....	145
4.7.3.2	Diagrama de Clase de Diseño.....	152
4.7.3.3	Diagrama de Secuencia.....	160
4.7.3.4	Modelo de Despliegue.....	183
4.7.4	Diseño de la Base de Datos.....	185
4.7.4.1	Modelo Relacional.....	186

4.8 Planificación de las Sigüientes Fases.	194
4.8.1 Fase de Construcción.....	194
4.8.2 Fase de Transición.....	194
4.9 Resumen de la Fase De Elaboración.	195
CAPITULO V	196
FASE DE CONSTRUCCIÓN	196
5.1 Introducción.....	196
5.2 Planificación de la Fase de Construcción.	197
5.3 Justificación de la Elección de las Herramientas de Construcción del Sistema.	198
5.4 Interfaces Principales del Sistema.	199
5.4.1 Interfaz de Inicio de Sesión.	199
5.4.2 Interfaz de Bienvenida al Sistema.	200
5.5 Primera Iteración.	201
5.5.1. Implementación de Subsistemas.....	201
5.5.2 Diagrama de Componente.	202
5.5.3. Implementación de las Clases.....	203
5.5.4 Pruebas por Unidad.	214
5.5.5 Pruebas de Integración.....	219
5.6 Segunda Iteración.	222
5.6.1 Implementación de Subsistemas.....	222
5.6.2 Diagrama de Componente.	223
5.6.3 Implementación de las Clases.....	224
5.6.4 Pruebas por Unidad.	235
5.6.5 Pruebas de Integración.....	240
5.8 Planificación de la Sigüiente Fase.	243
5.8.1. Fase de Transición.	243
5.9. Resumen de la Fase de Construcción.	243
CAPÍTULO VI.....	245

FASE DE TRANSICIÓN	245
6.1 Introducción.....	245
6.2 Preparación de la Versión Beta.....	246
6.3 Reacción a los resultados de las Pruebas.....	246
6.4 Ventajas del Manual de Usuario.....	247
6.5 Resumen de la Fase	247
CONCLUSIONES	248
RECOMENDACIONES	250
BIBLIOGRAFÍA	251
METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:.....	253

ÍNDICE DE TABLAS

TABLA 3.1. Clases de Dominio (1/2)	62
TABLA 3.2. Clases de Dominio (2/2)	62
TABLA 3.3. Riesgos del Sistema	67
TABLA 3.4. Actores del Sistema	68
TABLA 3.5. Casos de Uso del Sistema	70
TABLA 3.6. Clases de Control del Caso de Uso Administrar Orden de Compras ..	91
TABLA 3.7. Clases de Control del Caso de Uso Administrar Pagos	91
TABLA 3.8. Clases de Control del Caso de Uso Administrar Depósitos.....	92
TABLA 3.9. Clases de Control del Caso de Uso Administrar Productos.....	92
TABLA 3.10. Clases de Control del Caso de Uso Administrar Viáticos	93
TABLA 3.11. Clases de Control del Caso de Uso Generar Reportes.....	93
TABLA 3.12. Clases de Control del Caso de Uso Configurar Sistema.....	94
TABLA 3.13. Clases de Entidad del Sistema	95
TABLA 3.14 Clases de Interfaz del Sistema	95
TABLA 3.15. Mensajes del Diagrama de Colaboración Administrar Orden de Compra.....	103
TABLA 3.16. Mensajes del Diagrama de Colaboración Administrar Pagos	105
TABLA 3.17. Mensajes del Diagrama de Colaboración Generar Reportes.	107
TABLA 3.18. Mensajes del Diagrama de Colaboración Configurar Sistema	109
TABLA 4.1. Caso de Uso del Sistema.....	118
TABLA 4.2. Clases de Control del Caso de Uso Administrar Usuarios	124
TABLA 4.3. Clases de Entidad del Sistema	125
TABLA 4.4. Clases de Interfaz del Sistema	125
TABLA 4.5. Mensajes del Diagrama de Colaboración Administrar Depósitos	129
TABLA 4.6. Mensajes del Diagrama de Colaboración Administrar Productos.....	131
TABLA 4.7. Mensajes del Diagrama de Colaboración Administrar Viáticos.....	133

TABLA 4.8. Mensajes del Diagrama de Colaboración Administrar Usuarios.....	135
TABLA 4.9. Clases de Diseño a partir de Clases de Control (1/4)	145
TABLA 4.10. Clases de Diseño a partir de Clases de Control (2/4)	146
TABLA 4.11. Clases de Diseño a partir de Clases de Control (3/4)	147
TABLA 4.12. Clases de Diseño a partir de Clases de Control (4/4)	148
TABLA 4.13. Clases de Diseño a partir de Clases de Interfaz (1/2)	149
TABLA 4.14. Clases de Diseño a partir de Clases de Interfaz (2/2)	149
TABLA 4.15. Clases de Diseño a partir de Clases de Entidad (1/2)	150
TABLA 4.16. Clases de Diseño a partir de Clases de Entidad (2/2)	151
TABLA 4.17. Campos de la Tabla Administrar_Orden_de_Compra.....	187
TABLA 4.18. Campos de la Tabla Administrar_Pagos.....	188
TABLA 4.19. Campos de la Tabla Administrar _ depósitos.	189
TABLA 4.20. Campos de la Tabla Administrar_Productos.	190
TABLA 4.22. Campos de la Tabla Administrar_Usuarios	190
TABLA 4.21. Campos de la Tabla Administrar_Viaticos.	191
TABLA 4.23. Campos de la Tabla Usuario.....	192
TABLA 4.24. Campos de la Tabla Tipo_Usuarios.....	192
TABLA 4.25. Campos de la Tabla Dependencias.	193
TABLA 4.26. Campos de la Tabla Nucleos.	193
TABLA 4.27. Campos de la Tabla Tipo_Beneficiarios.....	193
TABLA 4.28. Campos de la Tabla Tipo_ Transporte.....	194
TABLA 5.1. Clases de Equivalencia para Administrar Orden de Compra (1/2).....	216
TABLA 5.2. Clases de Equivalencia para Administrar Orden de Compra (2/2).....	217
TABLA 5.3. Casos de Prueba para Administrar Orden de Compra (1/2).	218
TABLA 5.4. Casos de Prueba para Administrar Orden de Compra (2/2).	218
TABLA 5.5. Clases de Equivalencia para Administrar Pagos (1/2).....	237
TABLA 5.6. Clases de Equivalencia para Administrar Pagos (2/2).....	238
TABLA 5.7. Casos de Prueba para Administrar Pagos (1/2)	239
TABLA 5.8. Casos de Prueba para Administrar Pagos (2/2)	240

ÍNDICE DE FIGURAS

FIGURA 2.1. Proceso de Desarrollo de Software	35
FIGURA 2.2. Fases y Flujos de Trabajo.....	38
FIGURA 3.1. Flujos de Trabajo fundamental en la Fase de Inicio.....	59
FIGURA 3.2. Modelo de Dominio de SAAP	61
FIGURA 3.3. Diagrama de Casos de Uso del Sistema SAAP.....	71
FIGURA 3.4. Caso de Uso Administrar Orden de Compra.....	72
FIGURA 3.7. Caso de Uso Administrar Productos	78
FIGURA 3.9. Caso de Uso Generar Reportes.....	82
FIGURA 3.10 Caso de Uso Configurar Sistema	84
FIGURA 3.11. Paquete de Análisis Orden de Compra.....	86
FIGURA 3.12. Paquete de Análisis Pagos.....	87
FIGURA 3.13. Paquete de Análisis Depósitos	87
FIGURA 3.14. Paquete de Análisis Productos	88
FIGURA 3.15. Paquete de Análisis Viáticos	88
FIGURA 3.16. Paquete de Análisis Reportes	89
FIGURA 3.17. Paquete de Análisis Configurar.....	89
FIGURA 3.18. Diagrama de Clase de Análisis Administrar Orden de Compra.....	97
FIGURA 3.19. Diagrama de Clase de Análisis Administrar Pagos.....	98
FIGURA 3.20. Diagrama de Clase de Análisis Generar Reportes	99
FIGURA 3.21. Diagrama de Clase de Análisis Configurar Sistema	100
FIGURA 3.22. Diagrama de Colaboración Administrar Orden de Compra.....	101
FIGURA 3.24. Diagrama de Colaboración Generar Reportes.....	106
FIGURA 3.25. Diagrama de Colaboración Configurar Sistema.....	108
FIGURA 3.26. Distribución de los Subsistemas.....	111
FIGURA 4.1. Flujos de Trabajo Fundamental en la Fase de Elaboración.....	114
FIGURA 4.2. Diagrama de Casos de Uso del Sistema SAAP.....	119

FIGURA 4.3. Caso de Uso Administrar Usuarios	120
FIGURA 4.4. Paquete de Análisis Configurar.....	122
FIGURA 4.5. Paquete de Análisis Datos del Programa.....	123
FIGURA 4.6. Diagrama de Clase de Análisis Administrar Usuarios	126
FIGURA 4.7. Diagrama de Colaboración Administrar Depósitos	127
FIGURA 4.8. Diagrama de Colaboración Administrar Productos	130
FIGURA 4.9. Diagrama de Colaboración Administrar Viáticos	132
FIGURA 4.10. Diagrama de Colaboración Administrar Usuarios	134
FIGURA 4.11. Clases de Análisis del Paquete Administrar Orden de Compras	136
FIGURA 4.12. Clases de Análisis del Paquete Administrar Pagos	137
FIGURA 4.13. Clases de Análisis del Paquete Administrar Depósitos.....	138
FIGURA 4.14. Clases de Análisis del Paquete Administrar Productos.....	138
FIGURA 4.15. Clases de Análisis del Paquete Administrar Viáticos	139
FIGURA 4.16. Clases de Análisis del Paquete Generar Reportes.....	139
FIGURA 4.17. Clases de Análisis del Paquete Configurar Sistema.....	140
FIGURA 4.18. Clases de Análisis del Paquete Administrar Usuarios	140
FIGURA 4.19. Prototipo de la Interfaz Inicio de Sesión	142
FIGURA 4.20. Prototipo de la Interfaz de Usuarios.....	143
FIGURA 4.21. Distribución de los Subsistemas.....	144
FIGURA 4.22. Diagrama de Clases de Diseño Administrar Orden de Compras.....	153
FIGURA 4.23. Diagrama de Clases de Diseño Administrar Pagos.....	154
FIGURA 4.24. Diagrama de Clases de Diseño Administrar Depósitos.	155
FIGURA 4.25. Diagrama de Clases de Diseño Administrar Productos.	156
FIGURA 4.26. Diagrama de Clases de Diseño Administrar Viáticos.	157
FIGURA 4.27. Diagrama de Clases de Diseño Generar Reportes.....	158
FIGURA 4.28. Diagrama de Clases de Diseño Configurar Sistema.....	159
FIGURA 4.29. Diagrama de Clases de Diseño Administrar Usuarios.	160
FIGURA 4.30. Diagrama de Secuencia del caso de uso Administrar Orden de	

Compra.....	164
FIGURA 4.31. Diagrama de Secuencia del caso de uso Administrar Pagos.....	167
FIGURA 4.32. Diagrama de Secuencia del caso de uso Administrar Depósitos. ...	170
FIGURA 4.33. Diagrama de Secuencia del caso de uso Administrar Productos. ...	173
FIGURA 4.34. Diagrama de Secuencia del caso de uso Administrar Viáticos.	176
FIGURA 4.35. Diagrama de Secuencia del caso de uso Generar Reportes.....	178
FIGURA 4.36. Diagrama de Secuencia del caso de uso Configurar Sistema.....	179
FIGURA 4.37. Diagrama de Secuencia del caso de uso Administrar Usuarios	183
FIGURA 4.38. Modelo de Despliegue del Sistema Administrativo del Área de Postgrado (SAAP).....	196
FIGURA 4.39. Modelo Relacional de la Base de Datos de SAAP.....	186
FIGURA 5.1. Flujos de Trabajo Fundamental en la Fase de Construcción.....	196
FIGURA 5.2. Vista de la interfaz de Inicio de Sesión del Sistema	200
FIGURA 5.3. Vista de la Interfaz de Bienvenida	201
FIGURA 5.4. Subsistema de Implementación Administrar Orden de Compras	202
FIGURA 5.5. Diagrama de Componente para Administrar Orden de Compras	203
FIGURA 5.6. Vista de ventana de la IU Principal Administración de Orden de Compras	204
FIGURA 5.7. Diagrama de Implementación para Administrar Orden de Compras.	220
FIGURA 5.8. Ingreso de datos Orden de Compras	222
FIGURA 5.9. Subsistema de Implementación Administrar Pagos	223
FIGURA 5.10. Diagrama de Componente para Administrar Pagos	223
FIGURA 5.11. Vista de la Ventana de la IU Principal Administración de Pagos....	224
FIGURA 5.12. Diagrama de Implementación para Administrar Pagos.....	241
FIGURA 5.13. Ingreso de datos de Pagos.	243
FIGURA 6.1. Flujos de Trabajo Fundamentales en la Fase de Transición.....	245

RESUMEN

El presente proyecto nace con la finalidad de mejorar la integración y control de los procesos Administrativos del Área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente, desarrollando una aplicación gerencial bajo ambiente web. Dicha aplicación se realizó siguiendo la Metodología del Proceso Unificado de Desarrollo de Software, la cual proporciona una manera ordenada y eficiente de llevar a cabo la ejecución del proyecto a través del uso de sus fases: Inicio, Elaboración, Construcción y Transición; además el empleo de Técnicas Orientadas a Objetos específicamente el Lenguaje de Modelado Unificado (UML) y la Técnica del Modelo de Entidad Relación para la estructuración de la base de datos. Luego terminada estas dos primeras y fundamentales etapas del desarrollo de software se programó y depuró con la ayuda del usuario los procesos básicos de cada objeto de los módulos, logrando después de varias pruebas el producto final que alcanzó los objetivos planeados para este proyecto. Se aplicó también un lenguaje de programación basado en herramientas con licencias de software libre PHP (*Personal Hypertext Processor*) y para el diseño de la base de datos se utilizó herramientas de última generación soportado por el manejador Oracle8i. Así mismo se logro la creación de una plataforma que permite hacer más eficiente y dinámica la operatividad y productividad del Área de Postgrado que se involucran en el nuevo sistema y a los usuarios finales del mismo.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento del Problema

En los últimos tiempos el hombre ha vivido una impresionante transformación debido al desarrollo de la informática, por ello no ha cesado en la creación de máquinas y métodos para procesar, manejar y almacenar la información. Esta situación ha surgido como resultado de la necesidad de transmisión de datos en forma rápida y confiable. Hoy en día, el campo de aplicación o uso de los computadores es muy amplio y continúa expandiéndose rápidamente; sus velocidades de respuesta y capacidades de almacenamiento, igualmente mejoran y aumentan constantemente. Esto entre otras características ha hecho de ello una herramienta indispensable y extremadamente útil en casi todos los ambientes de la vida diaria.

El ritmo acelerado del crecimiento social en Venezuela, plantea un desafío ineludible al sector educativo en la formación de los profesionales y técnicos necesarios para el crecimiento de diversos sectores económicos del país. La consolidación del recurso humano de cuarto nivel, es decir, de postgrado, constituye una respuesta adecuada a tal necesidad.

La educación de postgrado es, en la actualidad, una actividad académica de gran importancia para los profesionales de la nación por su vinculación con los sistemas científicos, tecnológicos y productivos, además de fomentar las bases de la carrera ya culminada en la etapa de pregrado. Por esta razón y con la finalidad de

promover y desarrollar la preparación del estudiante de alto nivel y atender las necesidades de la sociedad en las áreas de docencia, investigación y capacitación para el ejercicio laboral, la Universidad de Oriente (UDO) es una de las casas de estudios más importantes para la continuación de la formación académica de los egresados en las diferentes instituciones de educación superior del país.

La Universidad de Oriente apertura los estudios de postgrado en el año 1971 con un solo programa. Actualmente ofrece una amplia variedad de cátedras en áreas que durante años se han venido consolidando hasta los más altos niveles de excelencia, a fin de contribuir al desarrollo de las regiones donde la universidad ha sentado sus bases, formando especialistas con los debidos conocimientos teóricos, metodológicos y técnicos que su labor requiere.

Los Centros de Estudios de Postgrado pertenecientes a la Universidad de Oriente se encargan de supervisar, velar y organizar periódicamente la ejecución de los distintos programas de postgrado, y de las actividades administrativas de las diferentes organizaciones que lo integran.

Actualmente, la Coordinación de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente no posee entre sus herramientas de trabajo un sistema de información que permita centralizar la data correspondiente a los procesos administrativos.

Los problemas que se generan en el ámbito del control de sus procesos como consecuencia de la falta de un sistema para realizar de forma aislada el almacenamiento de los datos, que cubra los requerimientos específicos de los usuarios afectan directamente a la comunidad universitaria, a la institución ante el gobierno y a los trabajadores de la universidad, son los siguientes:

Generan esfuerzos duplicados en la obtención de reportes; resultando información con bajo porcentaje de confiabilidad.

Elevados tiempos de ejecución o retraso en el desarrollo de todas las actividades que se llevan a cabo.

Falta de una base de datos para el almacenamiento de información, que permitan una mayor seguridad, control y fácil acceso a la información que se maneja.

Dificultad para controlar la privacidad, consistencia, integridad, respaldo y acceso de los datos.

Ausencia de una planificación semestral o anual de sus actividades administrativas, como consecuencia de la falta de organización de la información o generación de reportes que especifiquen el desarrollo de las actividades.

1.2 Propósito

Esta dificultad laboral incentiva el desarrollo del trabajo de investigación que aquí se presenta, proponiendo establecer la integración de la información y la comunicación de manera organizada entre coordinaciones de postgrado de los distintos Núcleos de la Universidad de Oriente, que a su vez puedan ser revisadas y modificadas por el usuario autorizado para tal fin, mediante el desarrollo de una herramienta de software que integre y gestione de manera automatizada las solicitudes y requerimientos de los sistemas administrativos en un ambiente WEB, teniendo como ventaja la interactividad, funcionalidad, transparencia y el manejo integral de la información, manejada a través de una aplicación gerencial administrativa, que formará parte de un elemento innovador dentro del ambiente de

trabajo de la Universidad de Oriente.

Con el software se permitirá la unificación de los datos necesarios para producir los indicadores que conlleven a la toma de decisiones de manera eficaz y confiable, así como también el resguardo de la información que se almacena en la base de datos, evitando los accesos no autorizados que puedan violar la integridad de la misma.

1.3 Alcance

Asimismo, este proyecto tendrá un alcance para todos los Núcleos del Centro de Estudios de Postgrado así como para la comunidad estudiantil de cuarto nivel, al permitir con el software una facilidad de conexión bajo una tecnología Web.

Durante la realización de este trabajo de grado se manejará el Proceso Conjugado de desarrollo de software a través de la Metodología del Proceso Unificado, con lo cual se cubrirán las fases de inicio, elaboración y construcción del programa; además, se contará con el Lenguaje Unificado de Modelado (UML), herramienta gráfica que permitirá la representación, abstracción y conceptualización del sistema para su análisis.

El lenguaje de programación a utilizar para la elaboración del sistema será PHP (*Personal Hypertext Processor*), un lenguaje basado en herramientas con licencia de software libre, que ofrece funciones como el acceso y gestión de bases de datos, creación de foros, libros de visita, gestión de archivos, etc. Para la creación del diseño de la base de datos se utilizará una herramienta de última generación soportado por el manejador ORACLE, que es muy rápido, seguro, fácil de usar, capaz de manipular cantidades enormes de registros, y compagina perfectamente con el

leguaje de programación a utilizar (PHP).

1.4 Importancia

La importancia de la creación, instalación y uso de este Sistema Gerencial radica en una notable mejora en la integración de los datos y aplicaciones, con una ganancia en la relación Tiempo-Trabajo que permitirá un mejor desempeño en la funcionalidad del Sistema de Postgrado. En tal sentido, al desarrollarse la aplicación web también se permitirá el control de la actividad administrativa, autorizando la actualización del manejo de la base de datos al operador del sistema donde se suministrará dicha información.

1.5 Originalidad

La originalidad de este proyecto resulta novedosa dentro del Núcleo de Anzoátegui de la Universidad de Oriente, ya que no cuenta con una aplicación gerencial que agrupe los datos correspondientes a los procesos administrativos de la coordinación de postgrado.

1.6 Objetivos

1.6.1 Objetivo General

Desarrollar un software gerencial para los procesos administrativos del área de postgrado del Núcleo de Anzoátegui de la Universidad de Oriente.

1.6.2 Objetivos Específicos

1. Describir las características y especificaciones de los procesos administrativos del área de postgrado de la Universidad de Oriente Núcleo de Anzoátegui.

2. Determinar los requerimientos del software.

3. Diseñar la base de datos del sistema para un correcto almacenamiento de la información.

4. Diseñar los diferentes módulos del sistema.

5. Codificar los diferentes módulos diseñados en el software.

6. Diseñar la interfaz de la aplicación que permita al usuario interactuar con el software de una manera intuitiva.

7. Realizar la integración, prueba y documentación de funcionamiento de todos los módulos que conforman el sistema.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Con la finalidad de tener una orientación y un soporte bibliográfico para la elaboración del proyecto se recurrió a trabajos de grado sobre temas y metodologías similares, elaborados con anterioridad en la Universidad de Oriente del Núcleo de Anzoátegui, debido a que presentan visualización de gráficas de variables, automatización de actividades relacionadas con servicios a usuarios y utilización de tecnología Web, respectivamente.

“Desarrollo de un Sistema Automatizado para el Control de Estudios del Área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente”. Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero en Computación en junio de 2002, por Thamara Isabel Rojas B. en la Universidad de Oriente – Núcleo Anzoátegui. Este proyecto nace con la finalidad de automatizar el sistema manual existente en la actualidad en el área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente. Así mismo, se logró la creación de una plataforma que permite hacer más eficiente y dinámica la operatividad y productividad del área de postgrado que involucra al nuevo sistema y a los usuarios finales del mismo. El diseño del sistema propuesto se realizó siguiendo la metodología de Ingeniería de Software y aplicando las técnicas orientadas a objetos a través del Lenguaje Unificado para el Modelado de Objeto (UML).[11]

“Diseño de un Sistema de Información para el Proceso de Transcripción de Notas en el Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui, Implantando un Sistema de Control de Seguridad de la Base de Datos por Autenticación de la Clave Pública”. Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero de Sistemas en septiembre de 2004, por Daniel J. Rapino Fuentes y Wadih Zouhairi Hamdan, en la Universidad de Oriente – Núcleo Anzoátegui. El Sistema de Información diseñado realiza todas las operaciones que se llevan a cabo en el proceso de transcripción de notas, reduciendo los tiempos de respuestas y eliminando los posibles errores en el momento de la transcripción que es uno de los principales problemas. Este sistema presenta una herramienta de seguridad muy poderosa que hace imposible el acceso no autorizado a la información almacenada, resguardando la integridad de los mismos. El diseño y el estudio del sistema se llevaron a cabo utilizando los diagramas del Lenguaje Unificado para el Modelado (UML). [8]

“Desarrollo de un Software como Soporte para la Automatización de las Actividades Administrativas que se llevan a cabo en una Institución Educativa”. Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero en Computación en septiembre de 2006, por Adriana María Arcila Benavides y Elic Nomar Zacarías Rodríguez en la Universidad de Oriente – Núcleo Anzoátegui. Este trabajo permitió la automatización de los procesos administrativos de la guardería preescolar “Trencito mi Rosita Mística”, llevándose a cabo las siguientes gestiones: Inscripción de Alumnos, Facturación, Contratación de Personal, Pago de Nóminas, Relación de Ingresos vs Egresos, entre otras. El desarrollo del proyecto se guió por la metodología de Proceso Unificado de Proceso de Software(UML).[6]

“Desarrollo de un Sistema de Información Gerencial bajo Plataforma Web para mejorar el Control de Gestión en el Centro de Estudio de Postgrado del

Núcleo de Anzoátegui de la Universidad de Oriente”. Trabajo de Grado presentado como requisito parcial para optar al grado de Magister Scientiarum en Informática Gerencial en abril de 2008, por Susana Cáceres Montero en la Universidad de Oriente – Núcleo Anzoátegui. El sistema de información desarrollado permite automatizar las transacciones administrativas referentes al área de postgrado y además proporciona las herramientas necesarias para apoyar las tareas gerenciales que se realizan en el CEPNA a fin de acomodarse en forma ventajosa a las exigencias cambiantes del entorno. El desarrollo del proyecto se guió por la metodología de Proceso Unificado de Desarrollo de Software.[5]

2.2 Marco Conceptual

2.2.1 Ingeniería de Software [9]

Surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos claves: *métodos*, *herramientas* y *procedimientos* que facilitan al gestor controlar el proceso de desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva. A continuación se examinan brevemente cada uno de estos elementos:

Los *métodos* de la ingeniería de Software indican cómo construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructura de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento.

Las *herramientas* de la ingeniería del software suministran un soporte automático o semiautomático para los métodos. Hoy existen herramientas para

soportar cada uno de los métodos mencionados anteriormente. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo de software, llamado *ingeniería de software asistida por computadora* (del inglés, CASE).

Los *procedimientos* de la ingeniería de software unen los métodos y las herramientas facilitando un desarrollo racional y oportuno del software de computadora como son: los procedimientos, la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que ayudan a los gestores de software a evaluar el progreso.

2.2.1.1 Paradigma del Ciclo de Vida Clásico para la Ingeniería de Software

Exige un enfoque sistemático y secuencial del desarrollo de software. Abarca las siguientes actividades:

Ingeniería y análisis del sistema: abarca los requisitos globales a nivel del sistema con una pequeña cantidad de análisis y diseño a un nivel superior.

Análisis de los requisitos del software: para comprender la naturaleza de los programas que hay que construir, el ingeniero de software (“analista”) debe interpretar el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el software.

Diseño: se enfoca sobre cuatro atributos distintos del programa (la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de

la interfaz). El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

Prueba: una vez generado el código, comienza la prueba de programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: no, suele representar una actividad específica, sino que consiste normalmente en repetir o rehacer partes de las actividades de las fases anteriores para introducir cambios en una aplicación de software ya entregada al cliente y puesta en explotación.

2.2.2 Proceso Unificado de Desarrollo de Software [13]

Es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado. Ver figura 2.1.



FIGURA 2.1. Proceso de Desarrollo de Software

[Fuente: Propia]

2.2.2.1 La Vida del Proceso Unificado

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes y consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones.

2.2.2.2 Fases dentro de un Ciclo del Proceso Unificado

Fase de Inicio: el objetivo principal de esta fase es establecer el análisis de negocio, para decidir si se sigue adelante con el proyecto. No representa un estudio completo del sistema propuesto, sino que en ella buscamos el porcentaje de casos de uso necesarios para fundamentar el análisis de negocio inicial. Para realizar este análisis se siguen cuatro pasos:

a. Delimitar el ámbito del sistema propuesto, es decir, definir los límites del sistema y empezar a identificar las interfaces con sistemas relacionados que están fuera de los límites.

b. Describir o esbozar una propuesta de la arquitectura del sistema que pueda soportar el ámbito del mismo (sistema), y en especial de aquellas partes que son nuevas, arriesgadas o difíciles.

c. Identificar riesgos críticos, es decir, los que afectan a la capacidad de construir el sistema, y determinar si se puede encontrar una forma de mitigarlos, quizás en una etapa posterior.

d. Demostrar a usuarios o clientes potenciales que el sistema propuesto es capaz de solventar sus problemas o de mejorar sus objetivos de negocio construyendo un prototipo.

Fase de Elaboración: el objetivo principal de esta fase es formular la línea base de la arquitectura; esto implica desarrollar alrededor del 80% de los casos de usos y abordar los riesgos que interfieran en la consecución de los siguientes objetivos:

a. Recopilar la mayor parte de los requisitos que aún quedan pendientes, formulando los requisitos funcionales como los casos de usos.

b. Establecer una base de arquitectura sólida (la línea base de la arquitectura) para guiar el trabajo durante la fase de construcción y transición así como en las posteriores generaciones del sistema.

c. Continuar la observación y el control de los riesgos críticos que aún queden, e identificar los riesgos significativos hasta el punto de que podamos estimar su impacto en el análisis de inicios.

d. Completar los detalles del plan de proyecto.

Fase de Construcción: su tarea fundamental es dejar listo un producto software en su versión operativa inicial, a veces llamada “versión beta”. El producto debería tener la calidad adecuada para su aplicación y asegurarse de cumplir los requisitos. La construcción debería tener lugar dentro de los límites del plan de negocio. Entre sus actividades se encuentran:

a. Detalla cada uno de los casos de uso y escenarios restantes, si es necesario modifica la descripción de la arquitectura y continúa los flujos de trabajo a través de iteraciones adicionales.

b. Finaliza los modelos de análisis, diseño e implementación, además integra los sistemas y subsistemas y los prueba.

c. Mantiene la integridad de la arquitectura, modificándola cuando sea necesario.

d. La monitorización de los riesgos críticos y significativos arrastrados desde las dos primeras fases, para su mitigación.

Fase de Transición: comienza a menudo con la entrega de la versión beta del sistema, es decir, un número reducido de usuarios con experiencia prueba el sistema e informa de defectos y deficiencias. La organización distribuye un producto software capaz ya de un funcionamiento inicial a una muestra representativa de la comunidad de usuarios. Entre las actividades de la transición se incluyen:

a. Preparar las actividades, como la preparación del lugar.

b. Aconsejar al cliente sobre la actualización del entorno en que el software va a funcionar.

- c. Preparar los manuales y otros documentos para la entrega del producto.
 - d. Ajustar el software para que funcione con los parámetros actuales del entorno de usuario.
 - e. Corregir los defectos encontrados a lo largo de las pruebas hechas a la versión beta.
 - f. Modificar el software al detectar problemas que no habían sido previstos.
- Ver figura 2.2.

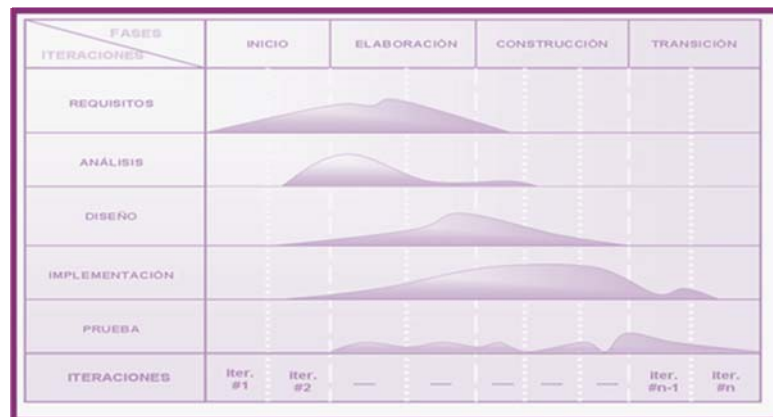


FIGURA 2.2. Fases y Flujos de Trabajo

[Fuente: Jacobson, I.]

2.2.2.3 Flujos de Trabajo en el Proceso Unificado.

Requisitos: el propósito fundamental es guiar el desarrollo hacia el sistema correcto, mediante una descripción de los requisitos del sistema, es decir, las condiciones o capacidades que debe cumplir como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre qué

debe y qué no debe hacer el sistemas

Análisis: durante la fase de análisis, se analizan los requisitos que fueron descritos en el flujo anterior. El objetivo de hacerlo es encontrar una comprensión más precisa de los requisitos y una descripción de los mismos, que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su estructura.

El resultado del *flujo de trabajo del análisis* es el modelo de análisis, que es un modelo de objetos conceptual que analiza los requisitos mediante su refinamiento y estructuración, proporcionando una comprensión detallada de los requisitos. El modelo de análisis incluye los paquetes del análisis y paquetes de servicio, las clases de análisis, las realizaciones de casos de uso-análisis y por último la vista de la arquitectura del modelo de análisis. Pueden considerarse como una primera aproximación al modelo de diseño, y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación, esto se debe a que debería ser mantenible el sistema en su conjunto, y no solo la descripción de sus requisitos.

Diseño: el propósito fundamental es modelar el sistema, mediante la comprensión profunda de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de interfaz de usuario, tecnologías de distribución y concurrencia, tecnología de gestión de transacciones, entre otros. Además de crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción; esto contribuye a una arquitectura estable y sólida y a crear un plano del modelo de implementación. Más tarde, durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la

implementación. El principal resultado del diseño es el modelo de diseño que se esfuerza en conservar la estructura del sistema impuesta por el modelo de análisis, y que sirve como esquema para la implementación. El diseño también obtiene como resultado un modelo de despliegue, que describe todas las configuraciones de red sobre las cuáles debería implantarse el sistema.

Implementación: la implementación se inicia con el resultado del diseño e implementando el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El propósito fundamental de la implementación es desarrollar la arquitectura y el sistema como un todo; mediante la planificación de las integraciones del sistema necesarias en cada iteración seguida para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables. Distribución del sistema asignando componentes ejecutables a nodos en el diagrama de despliegue la cual se basa en las clases activas encontradas durante el diseño. Implementación de las clases y subsistemas encontrados durante el diseño en particular, las clases se implementan como componentes de ficheros que contienen código fuente. La prueba de los componentes individualmente para integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema. El resultado principal de la implementación es el modelo de implementación. La implementación es el centro durante las iteraciones de construcción, aunque también se lleva a cabo trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura del sistema, y durante la fase de transición, para tratar defectos tardíos como los encontrados con distribuciones beta del sistema ya que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistema de implementación, es natural mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.

Prueba: en el *flujo de trabajo de la prueba* se verifica el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros. Los objetivos de la prueba son planificar las pruebas necesarias en cada iteración incluyendo las pruebas de integración y las pruebas del sistema; diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar, creando los procedimientos de pruebas que especifican como realizar las pruebas, y creando si es posible componentes de pruebas ejecutables par automatizar las pruebas; y por ultimo realizar las diferentes pruebas manejando los resultados de cada prueba sistemáticamente. El resultado principal de la prueba es el modelo de prueba, el cual describe como ha sido probado el sistema, la cual incluye los casos de prueba, procedimientos de prueba y los componentes de prueba. La prueba da también como resultado un plan de prueba, evaluaciones de las pruebas realizadas y los defectos que pueden ser pasados como entrada a flujos de trabajos anteriores, como el diseño y la implementación.

2.2.3 Lenguaje Unificado de Modelado (Unified Modeling Languaje, UML) [12]

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Es el lenguaje utilizado para expresar gráficamente el proceso de la generación de software. Mediante este lenguaje los analistas tienen un estándar con el cual expresarse. UML es independiente del lenguaje de implementación del software.

Diagrama de Casos de Uso: captura el comportamiento de un sistema o subsistema. Es un diagrama que se utilizan para el modelado de los aspectos dinámicos del sistema. Son importantes para visualizar, especificar y documentar el comportamiento de cada elemento del sistema. Suelen realizarse en una primera

etapa, en la especificación del software. Con ellos podemos determinar qué es lo que debe hacer el sistema.

Diagrama de Clases: se hace para definir el vocabulario del sistema. Su punto de vista de modelado del sistema es estático. En él vamos a representar las distintas clases que generarán los objetos correspondientes que hay en el sistema. Hay que tener en cuenta que hay tres agrupaciones en las clases, las que forman y tienen una relación con la base de datos directamente; las que sirven de control de las primeras, y las que se utilizan para la visualización de los contenidos. Es muy similar al modelo de Entidad Relación Extendida (EER) que se utiliza para el diseño de bases de datos. Posee las mismas relaciones y características.

Diagrama de Clases de Análisis: es utilizado por los desarrolladores de software para determinar los requerimientos funcionales, considerando una o varias clases, o sub-sistemas del sistema a desarrollar. Los casos de usos se describen mediante clases de análisis y sus objetos. El diagrama de clases de análisis se construye examinando los casos de usuarios, cerrando sus reacciones e identificando los roles de los clasificadores.

Diagrama de Clases de Diseño: se emplean para modelar la estructura estática de las clases en el sistema, sus tipos, sus contenidos y las relaciones que se establecen entre ellos. A través de este diagrama se definen las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización.

Diagrama De Secuencia: modela la relación entre los objetos desde un punto de vista temporal. El tiempo es la principal característica de este diagrama. Como su nombre indica, marca la secuencia que se produce en las llamadas de los métodos de los diferentes objetos. En caso de haber condiciones, es decir, que en un momento el diagrama pudiese derivar de una forma u otra, lo más conveniente es hacer dos

diagramas diferentes; no porque no se pueda hacer sino porque la representación acaba confusa ya que realmente el diagrama de secuencia sólo marca una secuencia.

2.2.4 Programación Orientada a Objetos [4]

La programación orientada a objetos es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representan una instancia de alguna clase, y cuyas clases son todos miembros de una jerarquía de clases unidas mediante relaciones de herencias. Existen tres importantes partes en la definición de la programación orientada a objetos:

- a. Utiliza objetos, no algoritmos, como bloques de construcción lógicos (jerarquía de objetos).
- b. Cada objeto es una instancia de una clase.
- c. Las clases se relacionan unas con otras por medio de relaciones de herencia.

Un programa puede parecer orientado a objetos, pero si cualquiera de estos elementos no existe, no es un programa orientado a objetos. Específicamente, la programación sin herencia es distinta de la programación orientada a objetos; se denomina programación con tipos abstractos de datos o programación basada en objetos.

2.2.4.1 Principios de los Lenguajes Orientados a Objetos

Según Bjarne Stroustrup, autor del lenguaje de programación C++, para que un lenguaje se llame a sí mismo orientado a objetos debe soportar tres conceptos: objetos, clases y herencia. Sin embargo, ha llegado a pensarse más comúnmente que los lenguajes orientados a objetos son lenguajes contruidos sobre el trípode encapsulación, herencia y polimorfismo. La razón de este cambio de filosofía es que con el paso de los años, los desarrolladores de software han llegado a darse cuenta que la encapsulación y el polimorfismo son partes tan integrantes de la construcción de sistemas orientados a objetos como la clase y la herencia.

Encapsulación:

Es una característica muy potente, y junto con la ocultación de la información, representan el concepto avanzado de objeto, que adquiere su mayor relevancia cuando encapsula e integra datos, más las operaciones que manipulan los datos en dicha entidad.

Herencia:

Es una propiedad que le permite a los objetos ser contruidos a partir de otros objetos. Dicho de otro modo, la capacidad de un objeto para utilizar las estructuras de datos y los métodos previstos en antepasados o ascendientes. El objetivo final es la reutilizabilidad o reutilización, es decir, reutilizar código anteriormente ya desarrollado. La herencia se apoya en el significado de ese concepto en la vida diaria, así las clases básicas o fundamentales se dividen en subclases, es decir, supone una clase base y una jerarquía de clases que contiene las clases derivadas de la clase base. Las clases derivadas pueden heredar el código y los datos de su clase base, añadiendo su propio código especial y datos a ellas, incluso cambiar aquellos elementos de la

clase base que necesita sean diferentes.

Polimorfismo:

Es la propiedad por la cual un mismo mensaje puede actuar de diferente modo cuando actúa sobre objetos diferentes ligados por la propiedad de la herencia. La gran ventaja ofrecida por el polimorfismo es permitir que los nuevos tipos de datos sean manipulados de forma similar que los tipos de datos predefinidos, logrando así ampliar el lenguaje de programación de una forma más ortogonal, en un sentido más general supone que un mismo mensaje puede producir acciones (resultados) totalmente diferentes cuando se recibe por objetos diferentes.

2.2.5 Bases de Datos [10]

Una base de datos es una colección de datos relacionados. Por datos, se quiere decir hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades implícitas:

Una base de datos representa algunos aspectos del mundo real, en ocasiones denominado mini mundo o Universo del Discurso (UdD) y sus cambios se reflejan en la base de datos.

Una base de datos es una colección coherente de datos con significados inherentes. Un conjunto aleatorio de datos no puede considerarse como una base de datos.

Una base de datos se diseña, construye y puebla con datos para un propósito específico. Está destinada a un grupo de usuarios concreto y tiene algunas aplicaciones preconcebidas en las cuales están interesados dichos usuarios.

2.2.5.1 Sistema de Gestión de Base de Datos (SGBD)

Es una colección de programas que permiten a los usuarios crear y mantener una base de datos. El SGBD es por tanto un sistema software de propósito general que facilita los procesos de definición, construcción y manipulación de base de datos para distintas aplicaciones. A continuación se describen cada uno de estos procesos:

Definición: consiste en especificar los tipos de datos, las estructuras y restricciones para los datos que se van a almacenar en dicha base de datos.

La construcción: es el proceso de almacenar los datos concretos sobre algún medio de almacenamiento controlado por el SGBD.

La manipulación: incluye funciones tales como consultar la base de datos para recuperar unos datos específicos, actualizar la base de datos para reflejar los cambios ocurridos en el mini mundo, y generar informes a partir de los datos.

Arquitectura de un Sistema de Gestión de Base De Datos (SGBD):

Una arquitectura para los sistemas de base de datos es la denominada arquitectura de tres esquemas. El objetivo de ésta es separar las aplicaciones del usuario y la base de datos física. En esta arquitectura se definen esquemas en los tres siguientes niveles:

El nivel interno: tiene un esquema interno que describe la estructura física de almacenamiento de la base de datos. El esquema interno emplea un modelo de datos físico y describe todos los detalles para su almacenamiento, así como los caminos de

acceso para la base de datos.

El nivel conceptual: tiene un esquema conceptual que describe la estructura de la base de datos completa para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras físicas de almacenamiento y se concentra en describir entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones. En este nivel podemos usar un modelo de datos de alto nivel o uno de implementación.

El nivel externo: o de vistas incluye varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado, y oculta a ese grupo el resto de la base de datos.

2.2.5.2 Modelo de Datos

Una característica fundamental del enfoque de base datos es que proporciona cierto nivel de abstracción de los datos; al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer. Un modelo de datos (colección de conceptos que sirven para describir la estructura de una base de datos) proporciona los medios necesarios para conseguir dicha abstracción. Con el concepto de estructura de una base de datos se refiere a los tipos de datos, los vínculos y las restricciones que deben cumplirse para esos datos. La mayoría de los modelos de datos contienen un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la base de datos. Entre los modelos de datos ampliamente utilizados se encuentran el modelo de datos relacional, el modelo de red y el modelo jerárquico

2.2.6 HTML (Lenguaje de Etiquetas por Hipertexto) [7]

Es un lenguaje de descripción de hipertextos compuestos por una serie de comandos, marcas o etiquetas también denominado “TAGS” (etiquetas) que permite definir la estructura lógica de un documento web y establecer los atributos de él mismo (color de textos, contenidos, multimedia e hipervínculos etc.).

2.2.6.1. Servidor Web

Es un amplio sistema de multimedia de acceso a la información heterogénea distribuida por toda la red en forma de documentos hipertextuales, este servicio surgió en 1990 en el CERN (Centren Européen de Recherche Nucleáire), con el objetivo de facilitar la distribución de información entre equipos investigadores geográficamente, es decir, se busca que los recursos disponibles en formatos electrónicos fuesen accesibles para cada investigador desde su propia terminal de forma clara y simple, posibilitando el salto entre elementos de información conexos. En definitiva trataba de integrar todos los recursos existentes en una red hipertextual.

2.2.7 PHP (Hypertext Preprocessor)

Es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source), que se caracteriza por su versatilidad, potencia, robustez y modularidad. Los programas escritos en PHP son integrados directamente en el código HTML y ejecutados por el servidor web a través de un intérprete.

2.2.7.1 Características Fundamentales de PHP

Por su flexibilidad, PHP resulta un lenguaje muy sencillo de aprender; especialmente para programadores familiarizados con lenguajes como C, Perl o Java, debido a similitudes de sintaxis entre ellos.

Es un lenguaje multiplataforma, los programas funcionan iguales sobre diferentes plataformas, trabajando la mayoría sobre servidores web y estando preparados para interactuar con más de 20 tipos de base de datos.

Además permite realizar múltiples tareas para el desarrollo de una web; entre éstas tenemos:

1. Facilidad para la conectividad con sistemas de gestores de bases de datos a través de un gran número de funciones para los datos, almacenamiento de información permanente en el servidor (Oracle, Postgress, Informix, MySQL, Server).

2. Funciones de correo electrónico que pueden ser utilizados para programar sistemas completos de correos electrónicos por vía web.

3. Función de administración y gestión de base de datos especificados para la mayoría de gestores comerciales; y funciones para conexiones ODBC con base de datos en sistemas Microsoft.

4. Funciones de gestión de dirección y ficheros, incluso para la transferencia de FTP.

5. Funciones de tratamientos de imágenes y librerías de funciones gráficas.

6. Función de generación y lecturas cookies

7. Funciones para documentos PDF.

2.2.8 ORACLE [1]

Oracle ofrece un completo e integrado conjunto de desarrollo de aplicaciones y herramientas de inteligencia empresarial que apoya cualquier enfoque de desarrollo, plataformas tecnológicas y sistemas operativos. Hacia un pensamiento arquitectónico los desarrolladores están utilizando estas herramientas hoy en día para abordar la complejidad de su aplicación y entornos de TI (Tecnología de la Información) con las Arquitecturas Orientadas a Servicios (SOA), lo que facilita el desarrollo de aplicaciones empresariales como los servicios a las empresas modular que puede ser fácilmente integrado y reutilizado, creando una verdadera flexibilidad e adaptable infraestructura de la TI(Tecnología de la Información).

Entre las características fundamentales para el uso de este manejador de base de datos tenemos:

1. Aporta los beneficios de la informática con más autonomía de gestión y automatización, lo que lo hace más fácil.

2. Cambia a los sistemas de TI (Tecnología de la Información) sin riesgo mediante la aplicación real de prueba.

3. Comprime y particionan las tablas para almacenar más datos y ejecutar consultas más rápido.

4. Protege con seguridad y permite total recuperación de los datos.
5. Integra y administra el ciclo de vida de todas las empresas de información.
6. Las prestaciones de Oracle establecen récord en Windows, Linux y UNIX.

2.2.8.1 Seguridad de la Base de Datos de Oracle

Presenta a la industria una tecnología más avanzada para la protección de los datos donde se almacenan en la base de datos.

Ofrece un amplio portafolio de soluciones de seguridad para garantizar la privacidad de los datos, protección contra las amenazas de abuso de información privilegiada, y permitir el cumplimiento regulatorio.

Los usuarios privilegiados y múltiples factores de control de acceso, la clasificación de los datos, transparente cifrados de datos, auditoría, seguimiento, y de ocultación de datos; los clientes pueden desplegar fiable y rentable soluciones que no requieren cambios a las solicitudes.

2.2.8.2 Oracle Tecnología Integrada

Muchos sistemas de hoy en día se han desplegado en el borde de los centros de datos, de la red, fuera en el entorno móvil, e incluso los consumidores en los hogares. Cada vez más, estos son para fines especiales en las aplicaciones, dispositivos y equipos que necesita para funcionar sin supervisión y administración de sus propios datos.

2.2.8.3 Oracle de Gestión de Contenidos

Ofrece una plataforma completa de gestión de contenidos.

Proporciona una mejor experiencia de usuario que se extiende por todo el espectro de contenido no estructurado, el desarrollo, reducción de los costos de administración y los aumentos individuales y equipo de productividad.

Superioridad e usabilidad construido en apoyo a los usuarios finales, grupos de trabajo, expertos de contenido, proceso de propietarios, administradores y webmasters.

Optimización de una arquitectura unificada para la gestión segura de documentos, archivos, contenido Web y activos digitales.

2.2.8.4 Oracle Arquitecturas Orientadas a Servicios (SOA)

Las empresas líderes están ganando la eficiencia operacional y la agilidad de negocios a través de adaptables, reutilizables procesos de servicios basados en una verdadera flexibilidad de las Arquitecturas Orientadas a Servicios (SOA), fundación que construye, implementa y administra lo mejor en su clase en las tecnologías, con lo cual dispone lo siguiente:

Ampliada e integrada plataforma de un conjunto completo de servicios y procesos de los componentes de la infraestructura para la construcción, despliegue y gestión de SOA.

Completa, de extremo a extremo del ciclo de vida de los servicios de gestión

de los asuntos públicos.

Extremo rendimiento y escalabilidad de las operaciones de memoria, en tiempo real de los procesos de eventos, y de alto volumen de transferencia de datos en la cima de una alta escalabilidad del servidor de aplicaciones.

2.2.8.5 Oracle Aplicación e Integración de Arquitectura

Ofrece una arquitectura abierta basada en normas para la creación de la aplicación transversal de los procesos de negocios que apoyan el modo en que ejecuten. Su aplicación independiente le permite utilizar su elección para crear compuestos de los procesos de negocio único, de una forma flexible e arquitectónica orientada al servicio. También ofrece proceso de Integración de embalaje, lo que permite la rápida aplicación de misión crítica de los procesos.

2.2.8.6 Oracle de Gestión de Identidades

Mejor en su clase.

Permite a las empresas gestionar de extremo a extremo el ciclo de vida de identidades de usuarios a través de todos los recursos tanto dentro como fuera del firewall.

Puede desplegar aplicaciones más rápido, eliminar automáticamente los privilegios de acceso latente.

Ofrece el más alto retorno de su inversión con la mejor seguridad, disminuir

los costes administrativos y de alta escalabilidad.

Proveen sostenibilidad para el cumplimiento de automatización de los procesos claves para reducir los ciclos de auditoría y los costes de conformidad.

2.2.8.7 Aplicabilidad de Oracle

Aplicaciones de Comunicaciones: ofrece los paquetes de software que proporcionan soluciones de extremo a extremo para apoyar la clave en los procesos de negocio de las empresas de comunicación, de creación de servicios, oferta de gestión, orquestación y el orden, mediante la provisión y prestación de servicios, para la facturación, garantía de ingresos, y la presentación de informes.

Las Solicitudes de Bancos: dependen de Oracle con el fin de ayudarles a simplificar los procesos de negocio, satisfacer las crecientes demandas de cumplimiento de la normativa y mejorar la eficiencia de costes proporcionando al mismo tiempo un mayor valor a los clientes y ofrece la más amplia funcionalidad para ayudar a los bancos de manera efectiva a crecer sus negocios optimizando el rendimiento.

Servicios Financieros: ofrece una potente combinación de tecnología y ampliada, pre integrada aplicaciones de negocio, incluyendo las principales funcionalidades construidos específicamente para la banca, los seguros, los mercados de capitales y organizaciones.

Ciencias de la Salud: ayuda a los productos farmacéuticos, biotecnología, dispositivos médicos y las compañías, así como la entrega integrada de redes y organizaciones de salud para descubrir, desarrollar y comercializar productos y

servicios innovadores para prevenir y curar enfermedades, mejorar la calidad de vida y satisfacer las expectativas de los accionistas.

Altas Aplicaciones de la Tecnología: sólo Oracle ofrece un modelo de datos unificado que reúne y sincroniza la información en tiempo real a través de su red de abastecimiento.

Las Solicitudes del Sector Público: ofrece tecnología y las aplicaciones para ayudar a los gobiernos a mejorar la eficiencia y la rendición de cuentas. Además todos los principales componentes de base de datos y aplicaciones para transformar las organizaciones en el sector público. También ofrece una completa solución integrada de los recursos humanos y gestión financiera para la adquisición y manejo de casos.

Venta al por menor de Aplicaciones: ofrece a la industria y la más completa suite integrada de aplicaciones de software para visión impulsada por la venta al por menor, todos los principales componentes de base de datos, y aplicaciones para transformar su negocio al por menor. Además proporciona una potente plataforma basada en estándares de la industria que permite a los minoristas aprovechar las inversiones existentes de TI y acelerar el rendimiento de las empresas con agilidad, flexibilidad y rentabilidad. Esta flexibilidad le ayuda a crear a corto plazo el valor, al tiempo que se persigue el crecimiento a largo plazo y la renovación.

Utilidades de Aplicaciones: ofrecer una solución completa e integrada para todos los segmentos de la industria de los servicios públicos de generación y transmisión a la distribución al por menor y servicios para poder ejecutar aplicaciones así como la velocidad de aplicación, optimización en el rendimiento y maximización en el retorno de la inversión.

2.2.9 Adobe Dreamweaver (Dw) [2]

Es una aplicación en forma de estudio (Basada en la forma de estudio de Adobe Flash®) pero con más parecido a un taller destinado para la construcción y edición de sitios y aplicaciones Web basados en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Sistemas). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es y tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas

La gran base de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en JavaScript-C lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de JavaScript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo ponga a su gusto.

Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG, sin embargo, versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos frameworks del lado servidor.

2.2.9.1 Características de Adobe Dreamweaver (Dw)

Además de sus capacidades WYSIWYG, tiene las funciones típicas de un editor de código fuente para la web:

1. Un administrador de sitios, para agrupar los archivos según el proyecto al que pertenezcan.

2. Un cliente FTP integrado, que permite subir los archivos editados inmediatamente al sitio en Internet.

3. Función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP.

2.2.10 JAVASCRIPT [3]

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.

CAPITULO III

FASE DE INICIO

3.1 Introducción

El Proceso Unificado de Desarrollo de Software es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

La vida del Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes y consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a su vez en iteraciones, que pasan por una serie de flujos de trabajo como son: Requisito, Análisis, Diseño, Implementación y Prueba.

En este capítulo se desarrollará la primera fase del Proceso Unificado, llamada *Fase de Inicio*, como lo indica la *figura 3.1*. Donde, se realizara un análisis del proyecto, creando un modelo inicial de la arquitectura del software y definir una planificación del proyecto que incluya los criterios de éxito, la evaluación de los riesgos que afecten la capacidad de construir el sistema, y determinar la forma de solventarlos a medida que se vayan presentando.

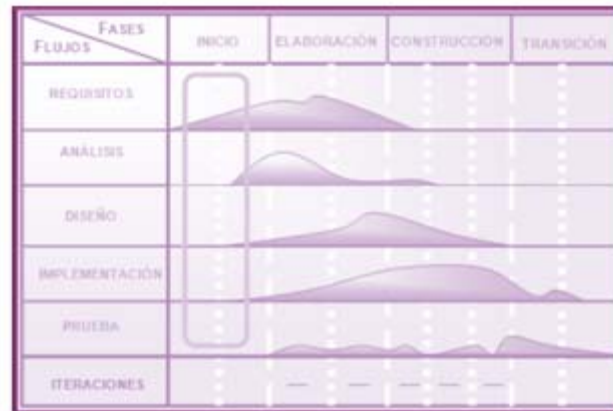


FIGURA 3.1. Flujos de Trabajo fundamental en la Fase de Inicio

[Fuente: Jacobson, I.]

3.2 Planificación de la Fase de Inicio

Para iniciar esta fase, se comenzará realizando un análisis del sistema, obteniendo los aspectos fundamentales más resaltantes; en primer lugar se estudiará el contexto del sistema, analizando la evolución y procedimientos involucrados para así definir los límites del sistema e identificar las interfaces (objetos y clases de dominios) que están relacionados, embozándolos a través del modelo de dominio. Como segundo punto, se identificarán los riesgos críticos que afectan la capacidad de construir el sistema, que serán disminuidos a medida que se avanza en la fase de elaboración del proyecto.

Luego en el tercer punto, se capturarán los requisitos que describirán el comportamiento del sistema así como también se identificarán los actores externos y la naturaleza de su interacción con él (sistema) a través del diseño para el modelado de los aspectos dinámicos del sistema llamados caso de uso ya que son importantes para visualizar, especificar y documentar el comportamiento de cada elemento del sistema.

Seguidamente se realizará el diagrama de paquetes de análisis para encapsular los casos de usos que fueron definidos al realizar el análisis del sistema. Posteriormente se procederá a modelar los casos de uso identificados, a través de los diagramas de clases de análisis. Por último, se elaborarán los diagramas de colaboración, para representar las interacciones entre los objetos.

3.3 Contexto del Sistema

Los Centros de Estudios de Postgrado pertenecientes a la Universidad de Oriente del Núcleo de Anzoátegui se encargan de supervisar, velar y organizar periódicamente la ejecución de los distintos programas de postgrado, y de las actividades administrativas de las diferentes organizaciones que lo integran.

El contexto del Sistema Administrativo del Área de Postgrado (*SAAP*) será representado a través de un Modelo de Dominio, que permitirá describir y comprender las clases más importante dentro del contexto del sistema y como se relacionan unas con otras mediante asociaciones.

3.3.1 Modelo de Dominio

Un modelo del dominio, captura los tipos más importantes de objetos en el contexto del sistema permitiendo enlazar estos objetos unos con otros. Tiene como objetivo contribuir a la comprensión del contexto del sistema, y por lo tanto, también contribuir a la comprensión de los requisitos del sistema que se desprenden de éste contexto. El modelo del dominio se describe mediante diagramas UML (especialmente mediante diagramas de clases).

seguimiento.

Sucesos que ocurrirán o han ocurrido.

En el análisis del contexto del sistema propuesto, se logró identificar y asignar un nombre a cada objeto del dominio, esto con la finalidad de facilitar el desarrollo de una lista de clases de dominio, que permitirá una mejor comunicación con todos los que trabajarán en el sistema. La lista de clases de dominio resultante se describe a continuación:

TABLA 3.1. Clases de Dominio (1/2)

CLASE DE DOMINIO	DESCRIPCIÓN
Universidad de Oriente	Casa de estudios importantes para la continuación de la formación académica de los egresados en las diferentes instituciones de educación superior del país.
Coordinación	Se refiere al acto de gestionar las interdependencias entre actividades que corresponden a las distintas Coordinaciones de Postgrado, de la Universidad De Oriente, Núcleo de Anzoátegui.
Funcionario	Abarca el operador que representa a las distintas coordinaciones de la Universidad de Oriente del Núcleo de Anzoátegui, para interactuar con los procesos administrativos del sistema.
Profesor	Son las personas que laboran dentro de la Universidad de Oriente, con el objetivo de enseñar o compartir los conocimientos orientados a profesionales universitarios

[Fuente: Propia]

TABLA 3.2. Clases de Dominio (2/2)

CLASE DE DOMINIO	DESCRIPCIÓN
Honorarios	Representa la información relacionada con los distintos pagos que se deban realizar a los profesores.
Cohorte	Son los programas de postgrados que se dictan en un año o semestre determinado.

Reportes	Son informes específicos, a través de los cuales las Coordinaciones pueden llevar una memoria y cuenta de las actividades que desarrollan.
Compras	Es el proceso que permite manipular y controlar la elaboración de compras, así como también la solicitud de orden de compra.
Pagos	Es el proceso que permite registrar los pagos de honorarios a los profesores, y diversos pagos que se realizan en la compra de algún producto que sea necesario.
Viáticos	Se refiere al pago que se proporciona para cubrir los gastos en el cumplimiento de comisiones de los profesores.
Cursos	Abarca los diferentes programas que se dictan en las distintas coordinaciones de postgrados.
Producto	Son los artículos tangibles que necesitan las coordinaciones de postgrados.
Compras	Es el proceso que permite manipular y controlar la elaboración de compras, así como también la solicitud de orden de compra.
Pagos	Es el proceso que permite registrar los pagos de honorarios a los profesores, y diversos pagos que se realizan en la compra de algún producto que sea necesario.

[Fuente: Propia]

3.4 Identificación de Riesgos

Los riesgos del desarrollo del software son todas aquellas contingencias que se puedan presentar durante la elaboración del mismo. La identificación de estos, permitirá que sean tomadas las acciones correctivas que prevengan retrasos y eventualidades que impidan el correcto desarrollo y funcionamiento final del sistema. Inicialmente se obtendrá una lista tentativa de riesgos, la misma ira incrementándose a medida que se avance en cada fase de desarrollo, e irán eliminándose a medida que estos sean solucionados. Los riesgos que se pueden considerar en un principio dependen de la fase de inicio y de elaboración del sistema y de elementos externos como en hardware que se deben considerar. Algunos de los riesgos iniciales identificados son, ambigüedad en la captura de los requisitos funcionales del software, que conllevaría a una errada construcción del mismo, diseño de una

arquitectura inadecuada a las necesidades funcionales del sistema, que no soporte los diferentes flujos de trabajos e iteraciones correspondientes al proceso unificado de desarrollo del software; debido a la naturaleza distribuida de la aplicación, la inadecuada implementación de la arquitectura cliente-servidor, representa un riesgo para el correcto funcionamiento de la aplicación. **Ver tabla 3.3 (*Riesgos del Sistema*)**.

3.5 Recopilar Requisitos

En la Fase de Inicio, el énfasis reside principalmente en el primer flujo de trabajo, el de *requisitos*. El propósito fundamental es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema, es decir, que las condiciones o capacidades que el sistema debe cumplir, sean suficientemente buenas como para que pueda llegarse a un acuerdo entre los usuarios y los desarrolladores sobre qué debe y qué no debe hacer el sistema.

Existen diferentes puntos de partida para la captura de los requisitos correctos y para la construcción del sistema deseado. Para ello, es necesario que los desarrolladores, posean un firme conocimiento del contexto en el que se emplaza el sistema, pudiendo tomar en consideración para lograrlo, la ejecución de los siguientes pasos:

3.5.1 Enumerar los Requisitos Candidatos

Son un conjunto de ideas que proporcionan inicialmente y de forma general los posibles requisitos funcionales del sistema, los mismos deberán someterse a una cuidadosa evaluación para verificar que estos lleven el cumplimiento del objetivo

final, que no es más que, la automatización de las actividades administrativas del área de postgrado. La lista de requisitos recopilada se muestra más adelante.

3.5.1.1 Capturar Requisitos Funcionales

Los requisitos funcionales, son aquellos que muestran la forma en la que el sistema interactúa con él o los usuarios finales, es decir, se toman en cuenta los procesos y métodos que se necesitan para ejecutar la interacción.

A continuación se muestran los requisitos funcionales identificados en el sistema, para esta fase:

Registro de Usuario: el sistema permitirá al administrador de la base de datos registrar a los diferentes usuarios que se encargaran de manipular la aplicación.

Administración de Registros: el sistema permitirá a las distintas coordinaciones de las áreas de postgrado ingresar los datos de las compras, pagos, viáticos, depósitos y productos, para que sean almacenados en la base de datos y, posteriormente puedan ser examinados por la Coordinación General de postgrado cuando lo desee.

Actualización de Registro: el sistema permitirá la modificación de los diferentes registros cuando el operador o funcionario correspondiente, envíe una solicitud de modificación, por causa de algún dato inválido ingresado, de forma que, éste sea corregido posteriormente en el sistema; y se proceda a actualizar el estado del mismo.

Generación de Reportes: las diferentes coordinaciones de las áreas de postgrado

podrán contar con reportes, ajustados a sus requerimientos, para su memoria y cuenta o planificaciones futuras.

Utilización de tecnología WEB en el desarrollo del sistema.

3.5.1.2 Capturar Requisitos No Funcionales

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad, y fiabilidad.

1. Los requisitos que no son vitales para el funcionamiento del sistema se señalan a continuación:

2. La codificación o construcción del sistema debe adaptarse a los estándares requeridos por las Coordinaciones de Postgrado, como lo son: los lenguajes de programación y las políticas para la integridad de los datos.

3. La transmisión debe ser segura, es decir, sólo las personas autorizadas pueden tener acceso a manipular la información.

4. El sistema debe ser diseñado siguiendo los requisitos y normativas establecidas por la Coordinación.

5. Diseño de una arquitectura robusta que permita al sistema ser sometida a cambio cuando sea necesario.

6. El sistema debe contar con una interfaz gráfica agradable de tal forma que, permita interactuar, de forma fácil, segura y cómoda; entre los usuarios y el sistema.

3.5.2 Modelo de Casos de Uso

El desarrollo de un modelo de caso de uso permitirá la captura de los requisitos funcionales del Sistema de Coordinación, puesto que facilitará a los desarrolladores del software y a los clientes, establecer el conjunto de condiciones y posibilidades que debe cumplir el sistema para cada tipo de usuario.

Un modelo de caso de uso, representa un modelo del sistema que contiene actores, casos de uso y sus relaciones, proporcionando la entrada fundamental para el análisis, el diseño y las pruebas. UML permite presentar el modelo en diagramas que muestran los actores y los casos de uso desde diferentes puntos de vista y con diferentes propósitos.

3.5.2.1 Identificación de Actores

La mayoría de los sistemas tienen muchos tipos de usuarios, cada tipo de usuario se representa mediante un actor, que se comunican con el sistema mediante el envío y recepción de mensajes hacia y desde el sistema, de acuerdo a los casos de uso Ver tabla 3.4 (*Actores del Sistema*) .

TABLA 3.3. Riesgos del Sistema

CLASE DE DOMINIO	DESCRIPCIÓN
Correcta Identificación De Los Requisitos Funcionales Y No Funcionales	Se basa esencialmente en identificar los casos de usos correctos con las interfaces de usuarios correctas.

Herramientas de Programación	Se debe tener el conocimiento de la robustez del lenguaje de programación seleccionado, para que sea adaptable a múltiples plataformas, y capaz de realizar los procedimientos requeridos por el sistema.
Hardware	Equipo computacional sobre el cual se vaya a ejecutar la aplicación representa un riesgo, ya que éste puede ocasionar ineficiencias entre el sistema, tal sería el caso si se implementa el sistema en un equipo con memoria insuficiente o de baja capacidad.
Diseño inadecuado de la Base de Datos del Sistema	Un diseño inadecuado de la base de datos puede causar inconvenientes en el manejo de los datos y generar dificultades en el desarrollo del sistema.
Construcción de una Arquitectura incorrecta	El diseño de la arquitectura del sistema, debe ser flexible, pudiendo evolucionar por las siguientes fases o durante su tiempo de vida. Este riesgo es considerado explícitamente durante la fase de inicio y elaboración.
Diseño de una Interfaz poco amigable	Las interfaces de usuario deben diseñarse de tal forma, que el usuario se sienta cómodo y seguro al usar el software.
Configuración	Se define este riesgo como el error al iniciar por primera vez el sistema el cual se hace a partir de su configuración. El mantenimiento del sistema se puede ver afectado por una errónea configuración.
Ayuda	Consiste en un módulo de ayuda que no sea compatible para el usuario o que no trate los aspectos que verdaderamente interesen a éste para el manejo del sistema.
Plataforma	Representa un riesgo para el sistema ya que debido a su naturaleza, éste pudiese presentar incompatibilidad total con la plataforma con el cual se pretenda cargar, haciendo imposible su utilización.
Operadores del Sistema	Un vago conocimiento de las capacidades del sistema por parte de los operadores ocasionarían salidas que no son las deseadas y podrían representar un riesgo sobre todo al tener acceso al modulo de configuración.

[Fuente: Propia]

TABLA 3.4. Actores del Sistema

ACTOR	DESCRIPCIÓN
Operador y/o Funcionario	Este actor puede interactuar con los procesos de administrar orden de compras, administrar pagos, administrar depósitos, administrar productos, administrar viáticos, generar reportes y configurar el sistema.

Administrador	Tiene bajo su cargo la responsabilidad de atender los procesos de configurar el sistema y además de vincularse con todas las funciones asociadas al operador.
Base de Datos SAAP	Tiene bajo su responsabilidad proporcionar todos los datos necesarios que cuando los diferentes subsistemas (registro compra, depósito, etc.), requieran dicha data.
Base de Datos Externa	El sistema Base de Datos Externa enviará los datos necesarios que requiera el sistema SAAP.

[Fuente: Propia]

3.5.2.2. Identificación de Casos de Uso

Los casos de usos son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores es decir, específica una secuencia de acciones que el sistema puede llevar acabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. Los casos de uso que describen la funcionalidad del sistema SAAP se identifican en la tabla 3.5 (**Ver tabla *Casos de Uso del Sistema***)

3.5.2.3 Diagrama de Casos de Uso

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada actor/caso de uso, que interactúan. Estos diagramas representan un conjunto de escenarios unidos por un objeto común, cada escenario describe una secuencia de eventos y cada secuencia a su vez, es iniciada por un actor u otro sistema.

TABLA 3.5. Casos de Uso del Sistema

CASO DE USO	DESCRIPCIÓN	ACTOR
Administrar Orden de Compras	Este proceso le permite al actor manipular y controlar la elaboración de compra.	Operador y/o Funcionario, Base de Datos SAAP
Administrar Pagos	Este proceso permite al actor realizar los pagos de honorarios entre otros.	Operador y/o Funcionario, Base de Datos SAAP, Base de Datos Externa
Administrar Depósitos	Este procedimiento permite al actor registrar los pagos respectivos de las cohortes de los inscritos en el postgrado.	Operador y/o Funcionario, Base de Dato SAAP, Base de Datos Externa
Administrar Productos	Este proceso permite al actor manipular y controlar la elaboración de una lista de los artículos tangibles que necesitan las coordinaciones de postgrados.	Operador y/o Funcionario, Base de Datos SAAP
Administrar Viáticos	Este proceso permite manipular, controlar los viáticos de los profesores que laboran en las distintas Coordinaciones del Área de Posgrado.	Operador y/o Funcionario, Base de Datos SAAP, Base de Datos Externa
Generar Reportes	Permite generar reportes de acuerdo a los parámetros especificados, por el actor que sirvan para la memoria y cuenta de la actividades que se dirijan dentro del área de postgrado.	Administrador, Base de Datos SAAP
Configurar Sistema	Permite controlar y/o modificar las variables y parámetros que se manejan dentro del sistema.	Administrador, Base de Datos SAAP

[Fuente: Propia]

El diagrama de caso de uso general para el Sistema Administrativo del Área de Postgrado (*SAAP*), que proporciona la estructura de la vista externa del sistema, se muestra a continuación.

En la *figura 3.3*. Se muestra la representación general del Sistema Administrativo del Área de Postgrado (*SAAP*). En este caso de uso general detallado

se observa que el sistema poseerá cuatro (4) actores: Operador y/o Funcionario, Administrador, Base de Datos SAAP y Base de Datos Externa. El *Operador*, tiene bajo su cargo la administración de los datos del sistema como compra, viáticos, pagos, depósitos entre otros. El *Administrador*, se encarga de la configuración de los aspectos globales del sistema, y las funciones que realiza el Operador. Ambas Base de Datos se encargaran de proporcionar la información necesarias que se requieran en el sistema.

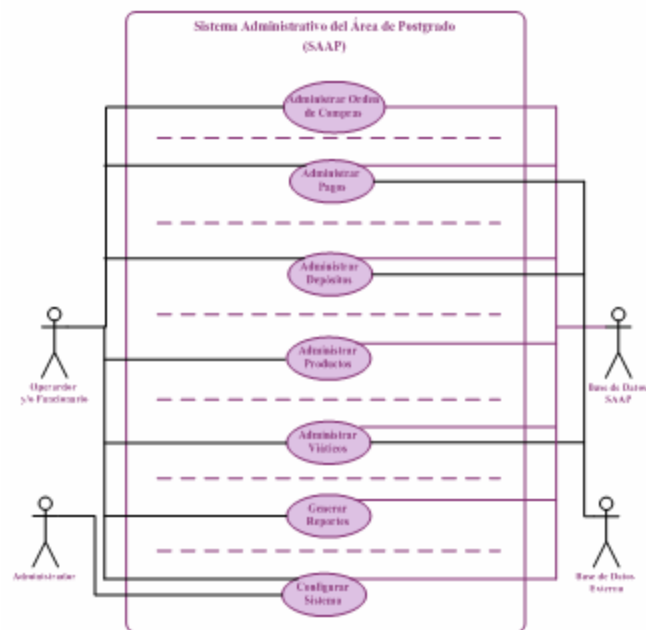


FIGURA 3.3. Diagrama de Casos de Uso del Sistema SAAP

[Fuente: Propia]

3.5.2.4 Casos de Uso Detallados

A partir del caso de uso general detallado del sistema, se hace necesario desglosar de manera individual los principales procesos que el SAAP desempeñará,

enfocándolos en casos de uso detallados, de tal forma, que se pueda descomponer en las distintas acciones que el sistema ejecutará. A continuación se presentan a detalle los diagramas de casos de uso más significativos del sistema:

Caso de Uso Administrar Orden de Compra

Este diagrama de caso de uso *figura 3.4*, describe el proceso de Administración de Orden de Compra, el cual será operado por el actor Operador y/o Funcionario y Base de Datos SAAP; para Registrar, Modificar y Eliminar las órdenes de Compras.



FIGURA 3.4. Caso de Uso Administrar Orden de Compra

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Orden de Compra.

Caso de Uso 1: ADMINISTRAR ORDEN DE COMPRA.

Actores Participantes: Operador y/o Funcionario, Base de Datos SAAP.

Descripción: Administrar los datos de las ordenes de compras de las diferentes Coordinaciones del Área de Postgrado de la Universidad de Oriente.

Condiciones de Entrada: El actor Operador debe seleccionar del menú principal la opción Administrar Orden de Compra.

Flujo de Eventos:

Flujo Principal

1. El actor Operador activa el Caso de Uso, seleccionando la opción Administrar Orden de Compra del Menú Principal.

2. Luego de esto, se carga en pantalla el listado de las órdenes de compras, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del *Código de Orden de Compra*, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado, se lleve a cabo satisfactoriamente.

4. En el caso de que el actor Operador desee realizar un nuevo registro, selecciona la opción Registrar Orden de Compras del menú Opciones que se mostrará en una interfaz principal.

5. El actor Operador, ingresa los datos mínimos requeridos, dentro del formulario establecido para el registro de orden de compras.

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida la data cargada para completar el registro.

8. Finalmente, el registro del postulante es cargado a la Base de Datos del sistema SAAP.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón

Cerrar Sesión.

Caso de Uso Administrar Pagos

Este diagrama de caso de uso *figura 3.5*, describe el proceso de Administración de Pagos, el cual será operado por el actor Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa, para Registrar Pagos Honorarios y Otros Pagos, Modificar y Eliminar Pagos.

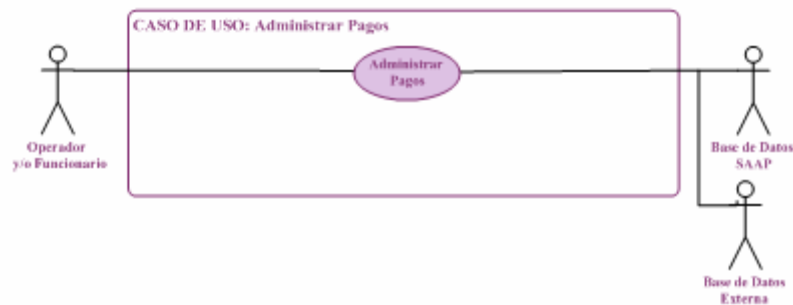


FIGURA 3.5. Caso de Uso Administrar Pagos

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Pagos

Caso de Uso 2: *ADMINISTRAR PAGOS*.

Actores Participantes: Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa.

Descripción: Administrar los pagos honorarios de los profesores y otros pagos.

Condiciones de Entrada: El actor Operador debe seleccionar del menú principal la opción Administrar Pagos.

Flujo de Eventos:

Flujo Principal

1. El actor Operador activa el Caso de Uso, seleccionando la opción Administrar Pagos del Menú Principal.

2. Luego de esto, se carga en pantalla el listado de los pagos, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del *Número de Pago*, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado, se lleve a cabo satisfactoriamente.

4. En el caso de que el actor Operador desee realizar un nuevo registro, selecciona la opción Registrar Pagos de Honorarios de Profesores o Registrar Otros Pagos del menú Opciones que se mostrará en una interfaz principal.

5. El actor Operador ingresa los datos requeridos.

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida que los datos hayan sido correctamente ingresados.

8. Finalmente, el registro es almacenado dentro de la Base de Datos del sistema *SAAP*.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón Cerrar Sesión.

Caso de Uso Administrar Depósitos

Este diagrama de caso de uso *figura 3.6*, describe el proceso de Administración de Depósitos, el cual será operado por el actor Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa; para Registrar, Modificar y Eliminar Depósitos.

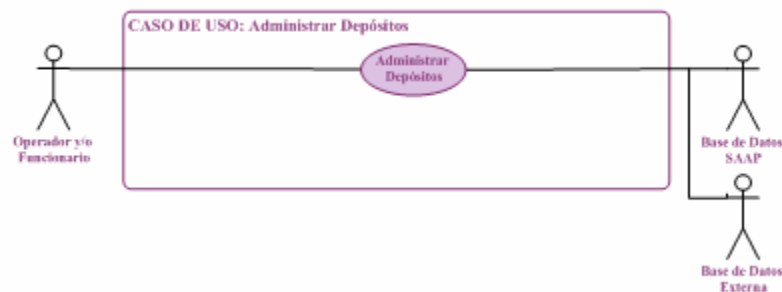


FIGURA 3.6. Caso de Uso Administrar Depósitos [Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Depósitos.

Caso de Uso 3: *ADMINISTRAR DEPÓSITOS*.

Actores Participantes: Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa.

Descripción: Administrar los datos de los depósitos del pago de las cohortes que realizan las personas inscritas en el Postgrado de la Universidad de Oriente Núcleo de Anzoátegui.

Condiciones de Entrada: El actor Operador debe seleccionar del menú principal la opción Administrar Depósitos.

Flujo de Eventos:

Flujo Principal

1. El actor Operador activa el Caso de Uso, seleccionando la opción Administrar Depósitos del Menú Principal.

2. Luego de esto, se carga en pantalla el listado de los depósitos, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del *Número de Depósito*, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado, se lleve a cabo satisfactoriamente.

4. En el caso de que el actor Operador desee realizar un nuevo registro, selecciona la opción Registrar Depósitos del menú Opciones que se mostrará en una interfaz principal.

5. El actor Operador, ingresa los datos mínimos requeridos, dentro del formulario establecido para el registro de depósitos.

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida la data cargada para completar el registro.

8. Finalmente, el registro del postulante es cargado a la Base de Datos del sistema SAAP.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón

Cerrar Sesión

Caso de Uso Administrar Productos

Este diagrama de caso de uso *figura 3.7*, describe el proceso de Administración de Productos, el cual será operado por el actor Operador y/o Funcionario, Base de Datos SAAP; para Registrar, Modificar y Eliminar Productos.

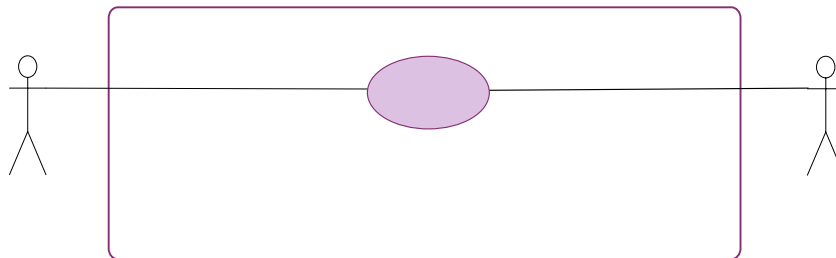


FIGURA 3.7. Caso de Uso Administrar Productos

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Productos.

Caso de Uso 4: *ADMINISTRAR PRODUCTOS*.

Actores Participantes: Operador y/o Funcionario, Base de Datos SAAP.

Descripción: Administrar los datos de los productos que necesitan las distintas Coordinaciones del Área de Postgrado de la Universidad de Oriente Núcleo de Anzoátegui. **CASO DE USO: Administrar**

Condiciones de Entrada: El actor Operador debe seleccionar del menú principal la opción Administrar Productos.

**Operador
y/o Funcionario**

Flujo de Eventos:

Flujo Principal

1. El actor Operador activa el Caso de Uso, seleccionando la opción Administrar Productos del Menú Principal.

2. Luego de esto, se carga en pantalla el listado de los productos, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del *Código del Producto*, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado, se lleve a cabo satisfactoriamente.

4. En el caso de que el actor Operador desee realizar un nuevo registro, selecciona la opción Registrar Productos del menú Opciones que se mostrará en una interfaz principal.

5. El actor Operador, ingresa los datos mínimos requeridos, dentro del formulario establecido para el registro de productos.

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida la data cargada para completar el registro.

8. Finalmente, el registro del postulante es cargado a la Base de Datos del sistema SAAP.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón Cerrar Sesión.

Caso de Uso Administrar Viáticos

Este diagrama de caso de uso figura 3.8, describe el proceso de Administración de Viáticos, el cual será operado por el actor Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa; para Registrar, Modificar y Eliminar Viáticos.

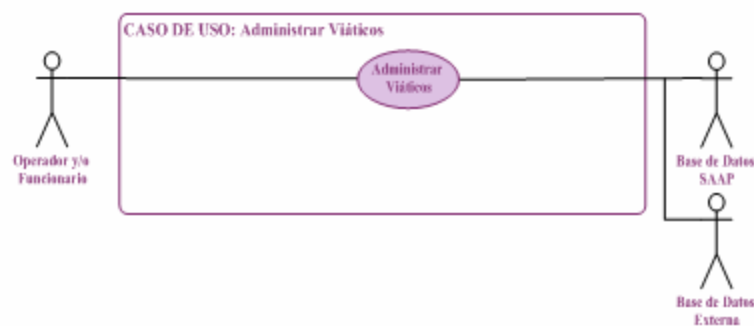


FIGURA 3.8. Caso de Uso Administrar Viáticos [Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Viáticos.

Caso de Uso 5: *ADMINISTRAR VIÁTICOS*.

Actores Participantes: Operador y/o Funcionario, Base de Datos SAAP y Base de Datos Externa.

Descripción: Administrar los datos de los viáticos de los profesores que laboran en la distintas Coordinaciones de Postgrado de la Universidad de Oriente Núcleo de Anzoátegui.

Condiciones de Entrada: El actor Operador debe seleccionar del menú principal la opción Administrar Viáticos.

Flujo de Eventos:

Flujo Principal

1. El actor Operador activa el Caso de Uso, seleccionando la opción Administrar Viáticos del Menú Principal.

2. Luego de esto, se carga en pantalla el listado de los viáticos, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del *Número de Viático*, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado se lleve a cabo satisfactoriamente.

4. En el caso de que el actor Operador desee realizar un nuevo registro, selecciona la opción Registrar Viáticos del menú Opciones que se mostrará en una interfaz principal.

5. El actor Operador, ingresa los datos mínimos requeridos, dentro del formulario establecido para el registro de viáticos

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida la data cargada para completar el registro.

8. Finalmente, el registro del postulante es cargado a la Base de Datos del sistema SAAP.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón Cerrar Sesión

Caso de Uso Generar Reportes

Este diagrama de caso de uso *figura 3.9*, describe el proceso de Generación de Reportes, el cual será operado por el actor Operador y/o Funcionario, Administrador, Base de Datos SAAP y Base de Datos Externa; con el fin de obtener reportes de las orden de compras, registros de pagos, registro de productos, registro de depósitos y registro de viáticos.

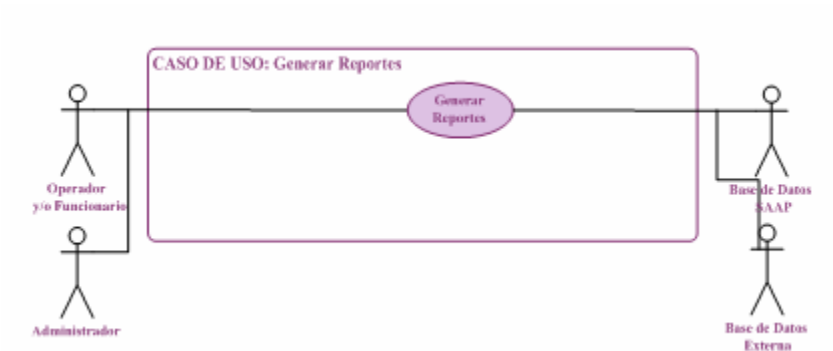


FIGURA 3.9. Caso de Uso Generar Reportes

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Generar Reportes.

Caso de Uso 6: GENERAR REPORTES.

Actores Participantes: Operador y/o Funcionario, Administrador, Base de Datos SAAP y Base de Datos Externa.

Descripción: Generar los reportes relacionados con la información que maneja el sistema respecto a los diferentes registros del sistema SAAP.

Condiciones de Entrada: El usuario debe seleccionar del menú principal la opción Generar Reportes.

Flujo de Eventos:

Flujo Principal

1. El usuario activa el Caso de Uso, seleccionando la opción Generar Reportes del Menú Principal.

2. Luego escoge una opción de las aplicaciones del módulo, de acuerdo al tipo de reporte que desea obtener como Generar Reporte de Orden de Compras, Generar Reporte de Pagos, Generar Reporte de Depósitos, Generar Reporte de Productos ó Generar Reporte de Viáticos.

3. El actor selecciona los datos que desee para generar *El Reporte* que desee según sus requerimientos.

4. El usuario presiona el botón *Mostrar* para obtener el listado correspondiente según las especificaciones establecidas.

5. Luego el actor puede presionar el botón *Imprimir Reporte* para generar el reporte.

6. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón Cerrar Sesión.

Caso de Uso Configurar Sistema

Este diagrama de caso de uso *figura 3.10*, describe el proceso de Configuración de Sistema, el cual será operado por el actor Operador y/o Funcionario, Administrador y Base de Datos *SAAP*; con el fin de manejar los datos utilizados por el sistema.

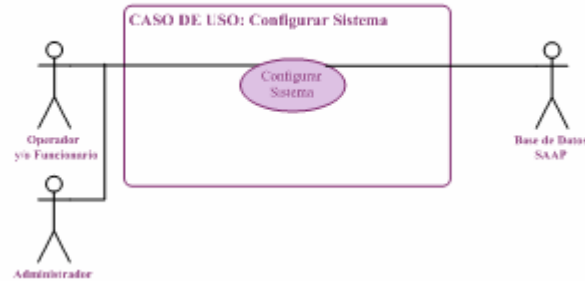


FIGURA 3.10 Caso de Uso Configurar Sistema

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Configurar Sistema.

Caso de Uso 7: *CONFIGURAR SISTEMA*.

Actores Participantes: Operador y/o Funcionario, Administrador y Base de Datos *SAAP*.

Descripción: Establecer los parámetros de configuración, de forma que se puedan manipular ciertos datos utilizados por el sistema.

Condiciones de Entrada: El actor debe seleccionar del menú la opción *Configurar Sistema*.

Flujo de Eventos:

Flujo Principal

1. El actor activa el Caso de Uso, seleccionando la opción *Configurar Sistema*.
2. El actor ejecuta las acciones correspondientes.
3. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú principal o presionar

el botón Cerrar Sesión.

3.6 Análisis

El flujo de trabajo análisis permite analizar y examinar los requisitos que se describieron durante la captura de requisitos, refinándolos y estructurándolos en un modelo de objetos que sirva como primera impresión del modelo de diseño; en esta fase, el resultado es un modelo inicial de análisis. El modelo de análisis permite crear un sistema robusto y flexible, incluyendo una arquitectura que emplea la reutilización de componentes de manera considerable; suele ser transitorio y sobrevivir sólo al primer par de iteraciones.

3.6.1 Análisis de la Arquitectura

El propósito del análisis de la arquitectura es, esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis, clases de análisis evidentes, y requisitos especiales comunes, hasta determinar que existe una arquitectura factible.

3.6.1.1 Identificación de Paquetes de Análisis.

Los paquetes del análisis proporcionan un medio para organizar el modelo de análisis en piezas pequeñas y manejables. Pueden identificarse inicialmente como una forma de dividir el trabajo de análisis, o pueden encontrarse a medida que el modelo de análisis evoluciona; esto se hace basándose en los requisitos funcionales y en el dominio del problema, es decir, en la aplicación que estamos considerando.

Debido a que capturamos los requisitos funcionales en la forma de caso de uso, una forma directa de identificar paquetes del análisis es asignar la mayor parte de un cierto número de caso de uso a un paquete concreto y después realizar la funcionalidad correspondiente dentro de ese paquete.

El contenido de los paquetes de análisis debe estar fuertemente relacionados y, deben ser débilmente acoplados, es decir, sus dependencias unos de otros deben ser mínima. A partir de los casos de usos más relevantes del Sistema Administrativo del Área de Postgrado (*SAAP*), se suscitaron los siguientes paquetes de análisis:

Paquete de Análisis Orden de Compras.

Este paquete de análisis contiene las acciones que permiten gestionar las órdenes de compras. Sostiene una relación directa con el caso de uso Administrar Orden de Compras.

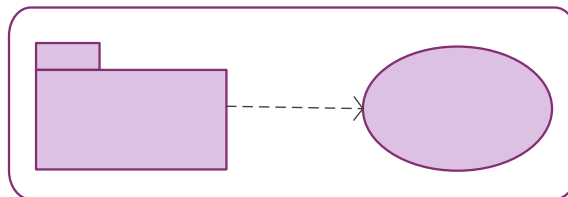


FIGURA 3.11. Paquete de Análisis Orden de Compra

[Fuente: Propia]

Paquete de Análisis Pagos.

Este paquete de análisis permite gestionar las acciones para la Administración de Pagos. Sostiene una relación directa con el caso de uso Administrar Pagos.

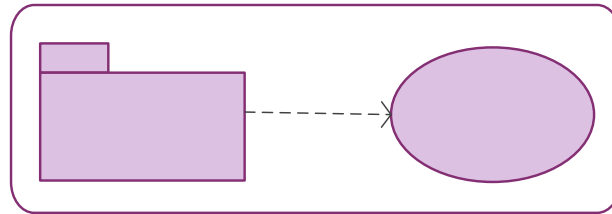


FIGURA 3.12. Paquete de Análisis Pagos

[Fuente: Propia]

Paquete de Análisis Depósitos.

Este paquete de análisis permite gestionar las acciones para la Administración de Depósitos. Sostiene una relación directa con el caso de uso Administrar Depósitos.

Pagos

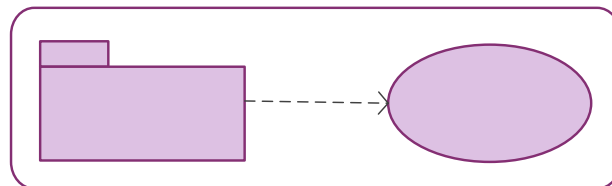


FIGURA 3.13. Paquete de Análisis Depósitos

[Fuente: Propia]

Paquete de Análisis Productos

Este paquete de análisis permite gestionar las acciones para la Administración de Productos. Sostiene una relación directa con el caso de uso Administrar Productos.

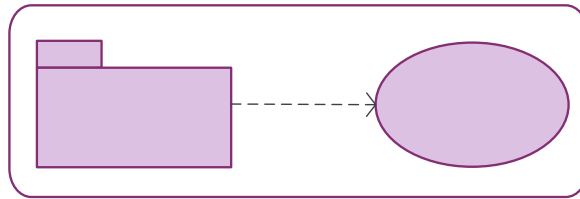


FIGURA 3.14. Paquete de Análisis Productos

[Fuente: Propia]

Paquete de Análisis Viáticos

Este paquete de análisis permite gestionar las acciones para la Administración de Viáticos. Sostiene una relación directa con el caso de uso Administrar Viáticos.

Productos

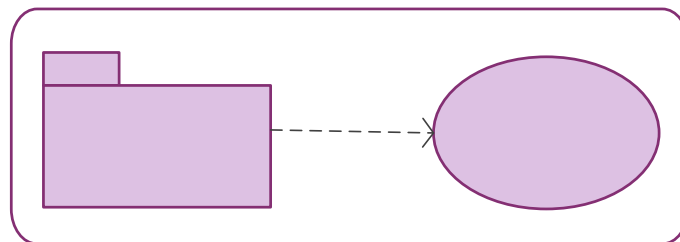


FIGURA 3.15. Paquete de Análisis Viáticos

[Fuente: Propia]

Paquete de Análisis Reportes.

Este paquete de análisis contiene las acciones para gestionar el proceso de Generación de los Reportes del sistema. Sostiene una relación inmediata hacia el caso de uso Generar Reportes.

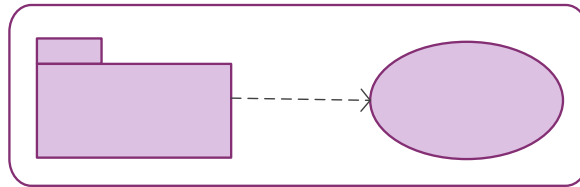


FIGURA 3.16. Paquete de Análisis Reportes

[Fuente: Propia]

Paquete de Análisis Configurar Sistema.

Este paquete de análisis contiene las acciones para gestionar el proceso de Configuración del Sistema SAAP. Sostiene una relación inmediata hacia el caso de uso Configurar Sistema.

Reportes

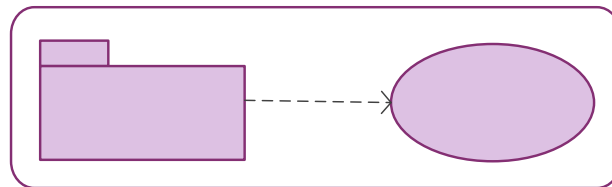


FIGURA 3.17. Paquete de Análisis Configurar

[Fuente: Propia]

3.6.2 Análisis de los Casos de Uso.

El análisis de los casos de uso se realiza para:

1. Identificar las clases del análisis, cuyos objetos son necesarios para llevar a cabo el flujo de sucesos de un caso de uso determinado.
2. Distribuir el comportamiento del caso de uso entre los objetos del análisis

que interactúan.

3. Capturar requisitos especiales sobre la realización del caso de uso.

3.6.2.1. Identificación de las Clases de Análisis.

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Existen tres estereotipos de clases estándar que se utilizan para ayudar a los desarrolladores a distinguir el ámbito de las diferentes clases, ellas son: clases de control, de interfaz, e identidad.

Durante este paso, basándose en su papel en las realizaciones de los casos de uso del sistema, se determinaron los correspondientes nombres, atributos, responsabilidades y relaciones de los tres estereotipos de clases de análisis que forman parte de la arquitectura del Sistema Administrativo del Área de Postgrado (*SAAP*) y, que a continuación se reseñan:

Clases de Control.

Las clases de control representan coordinación, secuencia, transacciones, y control de otros objetos. Con frecuencia se usan para encapsular el control de un caso de uso en concreto.

TABLA 3.6. Clases de Control del Caso de Uso Administrar Orden de Compras

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Orden de Compras	Permite el registro de una orden de compras a través de un formulario a partir del cual, los operadores podrán registrar la data mínima requerida para llevar a cabo el registro de orden de compras (código de compra, código de producto, descripción, fecha de compra, dependencia, núcleo).
Gestor Modificar Orden de Compras	Permite la modificación de la data de la orden de compras correspondiente a los datos de códigos de compras, códigos de productos, fecha de compra, dependencia y núcleo.
Gestor Eliminar Orden de Compras	Permite llevar a cabo la eliminación de una orden de compras específica.

[Fuente: Propia]

TABLA 3.7. Clases de Control del Caso de Uso Administrar Pagos

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Pagos	Permite registrar un pago mediante un formulario a partir del cual, los operadores podrán registrar la data mínima requerida para llevar a cabo los registros de pagos (Pagos de Honorario de Profesores y Otros Pagos).
Gestor Modificar Pagos	Permite modificar los datos e información referente a un pago de un registro específico.
Gestor Eliminar Pagos	Permite realizar la eliminación de un registro de pago.

[Fuente: Propia]

TABLA 3.8. Clases de Control del Caso de Uso Administrar Depósitos

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Depósitos	Permite el registro de los depósitos mediante un formulario a partir del cual, los operadores podrán registrar la data mínima requerida para llevar a cabo los registros de depósitos (Número Depósito, Número Cuenta, Fecha Depósito, Cédula, Nombre, Apellido, Dependencia, Núcleo, Programa, Cohorte, Sección , Monto del Depósito, Motivo, Observación, Total a Pagar, Total Pagado y Total por Pagar).
Gestor Modificar Depósitos	Permite modificar los datos e información referente a un depósito de un registro específico.
Gestor Eliminar Depósitos	Permite realizar la eliminación de un registro de depósito.

[Fuente: Propia]

TABLA 3.9. Clases de Control del Caso de Uso Administrar Productos

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Productos	Permite registrar un producto mediante un formulario a partir del cual, los operadores podrán registrar la data mínima requerida para llevar a cabo los registros de los productos (Código del producto, Unidad de Venta).
Gestor Modificar Productos	Permite modificar los datos e información referente a un producto de un registro específico.
Gestor Eliminar Productos	Permite realizar la eliminación de un registro de un producto.

[Fuente: Propia]

TABLA 3.10. Clases de Control del Caso de Uso Administrar Viáticos

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Viáticos	Permite registrar los viáticos mediante un formulario a partir del cual, los operadores podrán registrar la data mínima requerida para llevar a cabo los registros de los viáticos (Numero de Viático, Dependencia , Núcleo, Fecha Viático, Duración del Viaje, Fecha Destino, Motivo del Viaje, Destino del Viaje, Asignación, Tipo de Transporte, Línea Aérea, Valor del Boleto, Ruta de Viaje, Línea Terrestre, Carro Particular, Carro Por Puesto, Taxi, Otros Gastos, Precio de Carro Particular, Precio de Carro por Puesto, Precio de Carro taxi, Precio de Otros Gastos, Total de Asignación, Total de Pasaje, Total de Otros)
Gestor Modificar Viáticos	Permite modificar los datos e información referente a un viático de un registro específico.
Gestor Eliminar Viáticos	Permite realizar la eliminación de un registro de viáticos.

[Fuente: Propia]

TABLA 3.11. Clases de Control del Caso de Uso Generar Reportes

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Generar Reportes	Permite Generar un reporte según sean las especificaciones requeridas por la Coordinación Administrativa de Postgrado.

[Fuente: Propia]

TABLA 3.12. Clases de Control del Caso de Uso Configurar Sistema

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Configurar Sistema	Permite controlar y/o modificar las variables y parámetros que se manejan dentro del sistema.

[Fuente: Propia]

Clases de Entidad.

Las clases de entidad se utilizan para modelar la información y el comportamiento asociado a cierto fenómeno o concepto, como una persona, objeto o suceso del mundo real, que posee una vida larga y que es a menudo persistente.

Clases de Interfaz

Las clases de interfaz permiten modelar la interacción entre el sistema y sus actores (es decir, usuarios y sistemas externos). Esta interacción a menudo implica recibir y presentar información, peticiones de y hacia los usuarios y los sistemas externos. Las clases de interfaz modelan las partes del sistema que depende de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema. Por tanto, un cambio en una interfaz de usuario o en una interfaz de comunicaciones queda normalmente aislado en una o más clases de interfaz. Representan a menudo abstracciones de ventanas, formularios, paneles, interfaces de impresoras y de comunicaciones, entre otras.

TABLA 3.13. Clases de Entidad del Sistema

CLASE DE ENTIDAD	DESCRIPCIÓN
Reporte	Representa la información que es generada por la Coordinación Administrativa de Postgrado, de acuerdo a los parámetros especificados.
Depósitos	Representa la información relacionada con los diferentes depósitos que se realicen de acuerdo a la inscripción de algún curso o cohorte.
Productos	Representa los productos que se necesitan para el abastecimiento de las diferentes Coordinaciones de Postgrado.
Pagos	Representa la información relacionada con los distintos pagos que se deban realizar a los diferentes servicios ofrecidos a la Coordinación Administrativa de Postgrado.
Compras	Representa la información relacionada con las distintas compras de materiales que deban realizar a las distintas coordinaciones.
Honorario de Profesor	Representa la información relacionada con los distintos pagos que se deban realizar a los profesores.
Configuración	Representa los parámetros de configuración de los datos usados en el sistema.

[Fuente: Propia]

TABLA 3.14 Clases de Interfaz del Sistema

CLASES DE INTERFAZ	DESCRIPCIÓN
IU Principal Administración Orden de Compras	Es la interfaz principal para el modulo ADMINISTRAR ORDEN DE COMPRAS, donde los usuarios con acceso al modulo, puedan interactuar con las aplicaciones o procesos asociados.

IU Principal Administración de Pagos	Es la interfaz principal para el modulo ADMINISTRAR PAGOS, donde los usuarios con acceso al modulo, puedan interactuar con las aplicaciones o procesos asociados.
IU Principal Administración de Depósitos	Es la interfaz principal para el modulo ADMINISTRAR DEPÓSITOS, donde los usuarios con acceso al modulo, puedan interactuar con las aplicaciones o procesos asociados.
IU Principal Administración de Productos	Es la interfaz principal para el modulo ADMINISTRAR PRODUCTOS, donde los usuarios con acceso al modulo, puedan interactuar con las aplicaciones o procesos asociados.
IU Principal Administración de Viáticos	Es la interfaz principal para el modulo ADMINISTRAR VIÁTICOS, donde los usuarios con acceso al modulo, puedan interactuar con las aplicaciones o procesos asociados.
IU Generación de Reportes	Es la interfaz principal para el modulo GENERAR REPORTES, permite generar reportes según las especificaciones de la Coordinación Administrativa de Postgrado.
IU Configuración de Sistema	Es la interfaz principal para el modulo CONFIGURAR SISTEMA, el cual se encarga de procesar los parámetros que contienen las variables que brindan apoyo a la seguridad, control y configuración el sistema.

[Fuente: Propia]

3.6.2.2. Diagrama de Clases de Análisis.

Dentro del modelo de análisis, los diagramas de clases de análisis describen como se lleva a cabo y se ejecuta un caso de uso determinado, en términos de las clases de análisis y de sus objetos del análisis en interacción; centrándose en los requisitos funcionales. Los diagramas de clase de análisis que se diseñaron para el Sistema Administrativo del Área de Postgrado (SAAP) son:

Diagrama de Clase de Análisis para el Caso de Uso Administrar Orden de

Compra.

Este caso de uso permite la Administración de la Orden de Compra, enfocando su funcionalidad en el registro, modificación, y eliminación, con el fin de controlar la información referente a las órdenes de compras.

En la *figura 3.18*, se muestra el diagrama de clase de análisis que describe como se lleva a cabo el caso de uso Administrar Orden de Compra.

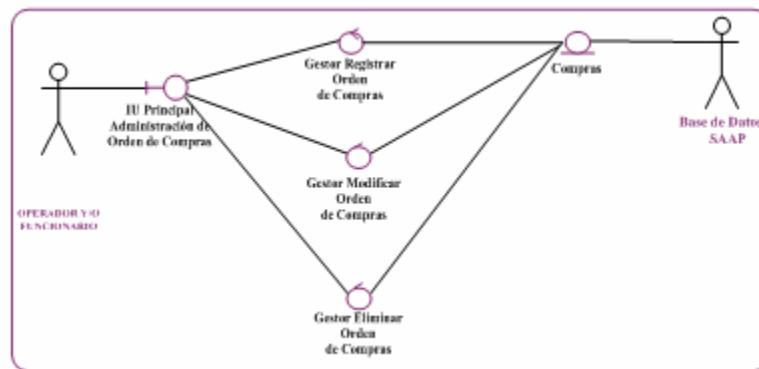


FIGURA 3.18. Diagrama de Clase de Análisis Administrar Orden de Compra

[Fuente: Propia]

Como se puede observar, el usuario Operador y/o Funcionario activa el caso de uso mediante la: IU Principal Administración Orden de Compra donde se muestra la lista de Orden de Compras y las diferentes opciones que le permiten interactuar con los gestores :Gestor Registrar Orden de Compra, :Gestor Modificar Orden de Compra, y el :Gestor Eliminar Orden de Compra. Para los procesos de registro, modificación y eliminación de orden de compras, la entidad vinculada es Compra.

Diagrama de Clase de Análisis para el Caso de Uso Administrar Pagos.

Este caso de uso permite la Administración de los Pagos, incluyendo los procesos Registrar, Modificar y Eliminar Pagos, con el fin de controlar la información referente a los pagos suscritos.

En la *figura 3.19* se muestra el diagrama de clase de análisis que describe como se lleva a cabo el caso de uso Administrar Pagos.

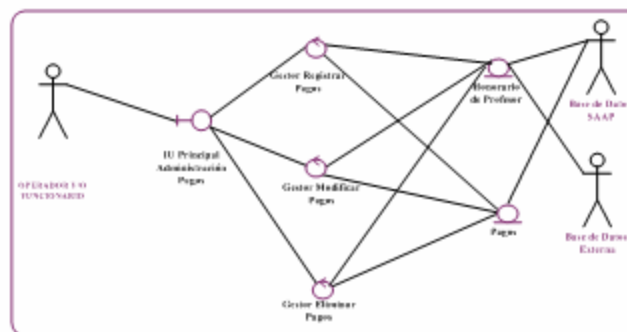


FIGURA 3.19. Diagrama de Clase de Análisis Administrar Pagos

[Fuente: Propia]

Como se puede observar, el usuario Operador y/o Funcionario activa el caso de uso mediante la: IU Principal Administración Pagos. Donde se muestra la lista de Pagos y las diferentes opciones que le permiten interactuar con los gestores: Gestor Registrar Pagos, :Gestor Modificar Pagos y el :Gestor Eliminar Pagos. Para los procesos de registro, modificación y eliminación de Pagos, las entidades vinculadas son Honorario de Profesor y Pagos.

Diagrama de Clase de Análisis para el Caso de Uso Generar Reporte.

Este caso de uso le facilita al usuario correspondiente la generación de reportes, que le permitirán llevar un control estadístico de las actividades gestionadas por el sistema.

En la *figura 3.20.*, se muestra el diagrama de clase de análisis referente al caso de uso Generar Reportes.

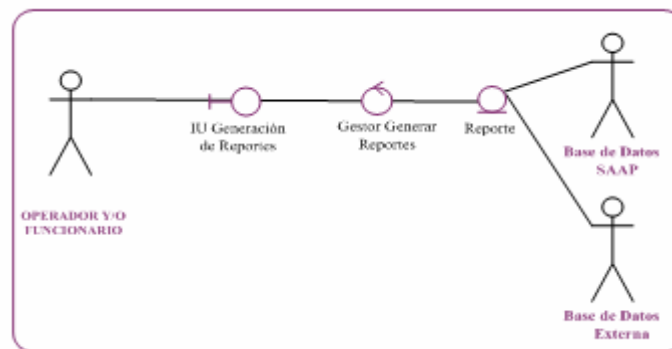


FIGURA 3.20. Diagrama de Clase de Análisis Generar Reportes

[Fuente: Propia]

Se observa que el usuario correspondiente activa el caso de uso mediante la interfaz Generación de Reportes. El usuario activa el Gestor Generar Reportes para finalmente adquirir el tipo de reporte que desea obtener como Generar Reporte de Orden de Compras, Generar Reporte de Pagos, Generar Reporte de Depósitos, Generar Reporte de Productos ó Generar Reporte de Viáticos.

Diagrama de Clase de Análisis para el Caso de Uso Configurar Sistema.

Este caso de uso le permite que ciertos parámetros del sistema puedan ser configurables según las necesidades y requerimientos que se susciten, para así garantizar la vida útil del sistema.

En la *figura 3.21*, se muestra el diagrama de clase de análisis referente al caso de uso Configurar Sistema.

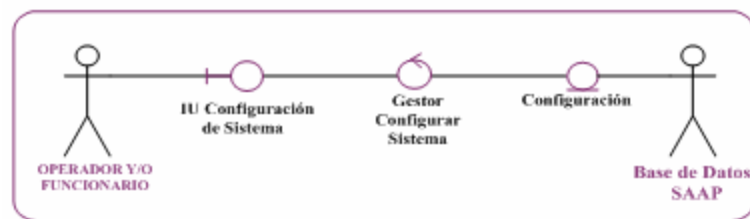


FIGURA 3.21. Diagrama de Clase de Análisis Configurar Sistema

[Fuente: Propia]

Se observa que el usuario Administrador activa el caso de uso mediante la interfaz principal del sistema. El Administrador establece los parámetros de funcionamiento del sistema a través de la: IU Configurar Sistema, posteriormente la clase: Gestor Configurar Sistema se encarga de procesar los parámetros y almacenarlos en la clase de entidad Configuración, que contienen las variables que brindan apoyo a la seguridad, control y configuración del sistema.

3.6.2.3. Diagrama de Colaboración.

Los diagramas de colaboración tienen como fundamento identificar los requisitos y las responsabilidades sobre los objetos del análisis, mostrando las interacciones y creando enlaces entre ellos, mediante la agregación de mensajes a

esos enlaces. El nombre de un mensaje, denota el propósito del objeto invocante en la interacción con el objeto invocado.

Dentro del modelo de análisis, los diagramas de colaboración del SAAP que permiten mostrar los casos de uso más relevantes en términos de interacciones de objetos del análisis, son los siguientes:

Diagrama de Colaboración para el Caso de Uso Administrar Orden de Compra.

En la *figura 3.22*. Se observa el diagrama de colaboración del caso de uso *Administrar Orden de Compra*. Mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

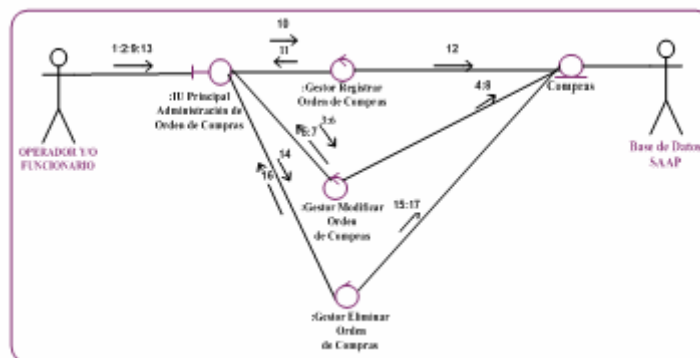


FIGURA 3.22. Diagrama de Colaboración Administrar Orden de Compra

[Fuente: Propia]

Un operador del sistema puede realizar una búsqueda de orden de compras, seleccionándola del listado; con el fin de modificarla. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración Orden de Compra. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Orden de Compra a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través del código de compras) para seleccionar el Registro de Orden de Compras con los datos. Posteriormente a través del mensaje (4) se busca el Registro de la Orden de Compras seleccionado en la entidad Compras.

El :Gestor Modificar Orden de Compra mediante el mensaje (5) muestra el Registro de Orden de Compras seleccionado. Seguidamente se suministra los nuevos datos del registro Orden de Compras modificados, según el mensaje (6).

El :Gestor Modificar Orden de Compra valida la información modificada según el mensaje (7), para que finalmente sea almacenada dentro de la entidad Compras a través del mensaje (8).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 3.15.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Administrar Orden de Compra.

TABLA 3.15. Mensajes del Diagrama de Colaboración Administrar Orden de Compra

LEYENDA	
1. Activar IU Principal Administrar Orden de Compra.	9. Solicitar Registro de una nueva Orden de Compra.
2. Solicitar Modificación de las Órdenes de Compras.	10. Cargar data del formulario de Registro.
3. Cargar dato para seleccionar el Registro de Ordenes de Compras para modificar.	11. Validar Registro de Orden de Compra.
4. Buscar Registro de Orden de Compra seleccionado.	12. Registrar los datos de la nueva Orden de Compra
5. Mostrar Registro de Orden de Compra seleccionado.	13. Solicitar Eliminación de Orden de Compra.
6. Cargar nuevos datos de la Orden de Compra seleccionado.	14. Cargar dato para seleccionar el Registro a eliminar.
7. Validar nuevos datos de Orden de Compra.	15. Buscar registro de la Orden de Compra seleccionada.
8. Guardar Registro de la Orden de Compra modificada.	16. Mostrar Registro de la Orden de Compra seleccionada.
	17. Eliminar Orden de Compra.

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Administrar Pagos.

En la *figura 3.23* se observa el diagrama de colaboración del caso de uso Administrar Pagos, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

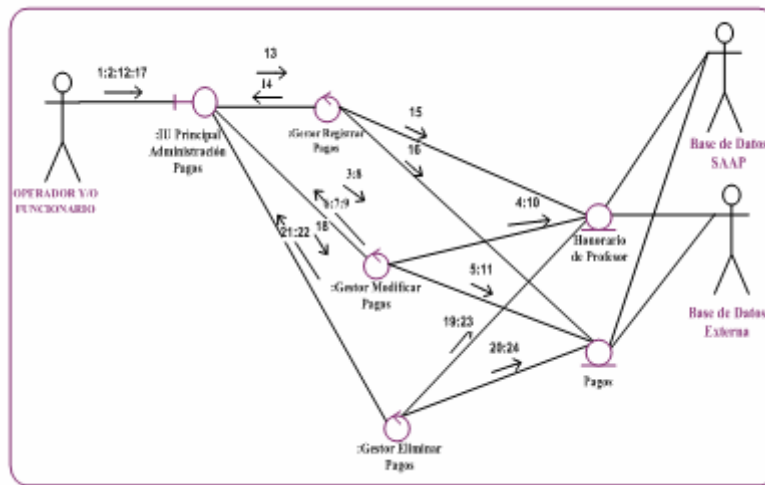


FIGURA 3.23. Diagrama de Colaboración Administrar Pagos

[Fuente: Propia]

Un operador del sistema puede realizar una búsqueda avanzada de pagos, con el fin de modificar específicamente, algún dato de los Pagos Honorarios de Profesores y Otros Pagos. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración de Pagos. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Pagos a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través de numero de pago) para seleccionar el Registro con los datos correspondientes. Posteriormente a través del mensaje (4 y 5) se busca el Registro ya sea de Honorario de Profesor u Otros Pagos seleccionado en las entidades de Honorario de Profesor y Pagos.

El :Gestor Modificar Pagos mediante el mensaje (6 y 7) muestra el Registro seleccionado. Seguidamente se suministra los nuevos datos del registro modificado, según el mensaje (8).

El :Gestor Modificar Pagos valida la información modificada según el mensaje (9), para que finalmente sea almacenada dentro de las entidades Honorario de Profesor a través del mensaje (10) y Pagos a través del mensaje (11).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 3.16.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Administrar Pagos.

TABLA 3.16. Mensajes del Diagrama de Colaboración Administrar Pagos

LEYENDA	
1. Activar IU Principal Administrar Pago.	13. Cargar data para el formulario de Registro de Pago.
2. Solicitar Modificación de Pago.	14. Validar Registro de Pago.
3. Cargar data para seleccionar el Pago (Honorario de Profesor u otros Pagos).	15. Registrar Pago de Honorario de Profesor.
4. Buscar Registro de pagos de Honorario de profesor seleccionado.	16. Registrar Pagos.
5. Buscar Registro de otros Pagos seleccionado.	17. Solicitar la Eliminación de un Pago (Honorario de Profesor o Pagos).
6. Mostrar Registro de Pago de Honorario de Profesor seleccionado.	18. Cargar dato para seleccionar el registro a eliminar.
7. Mostrar Registro de otros Pagos seleccionados.	19. Buscar Registro de Pago de Honorario de profesor seleccionado a eliminar.
8. Cargar datos nuevos (Honorario de Profesor u otros Pagos).	20. Buscar Registro de Otros Pagos seleccionado a eliminar.
9. Validar nuevos datos de pagos (Honorario de Profesor u otros Pagos).	21. Mostrar Registro de Pagos de Honorario de Profesor seleccionado.
	22. Mostrar registro de pago seleccionado.

<p>10. Guardar Registro modificado de pagos de Honorario de Profesor.</p> <p>11. Guardar registro modificado de otros pagos.</p> <p>12. Solicitar Registro de un nuevo Pago ya sea de Honorario de Profesor o Pagos especificados.</p>	<p>23. Eliminar Registro de Pago de Honorario de Profesor.</p> <p>24. Eliminar Registro de otros Pagos.</p>
--	---

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Generar Reportes.

En la figura 3.24. Se observa el diagrama de colaboración del caso de uso Generar Reportes, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de este caso de uso

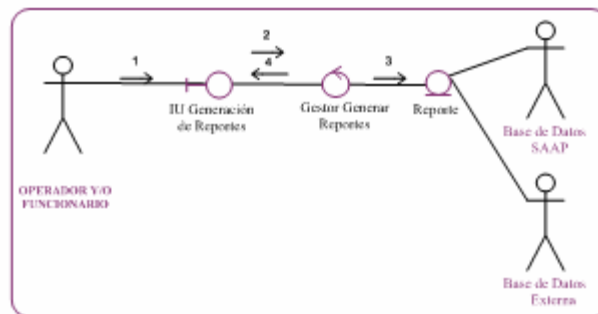


FIGURA 3.24. Diagrama de Colaboración Generar Reportes

[Fuente: Propia]

Un operador del sistema puede generar un reporte específico ya sea de:

Orden de Compras, Pagos, Depósitos, Productos o Viáticos. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Generación de Reportes. En ésta interfaz, el usuario a través del mensaje (2), solicita Generar un Reporte, activando de esta manera, al: Gestor Generar Reporte.

Seguidamente, mediante el mensaje (3) el: Gestor Generar Reporte, busca el reporte, procesa la información para que finalmente se muestre el reporte al usuario.

En la *tabla 3.17.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Generar Reportes.

TABLA 3.17. Mensajes del Diagrama de Colaboración Generar Reportes.

LEYENDA
1. Solicitar Generación de Reportes.
2. Procesar Generar Reportes.
3. Buscar Generar Reporte
4. Mostrar Generación de Reportes.

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Configurar Sistema.

En la *figura 3.25*. Se observa el diagrama de colaboración del caso de uso Configurar Sistema, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de este caso de uso.

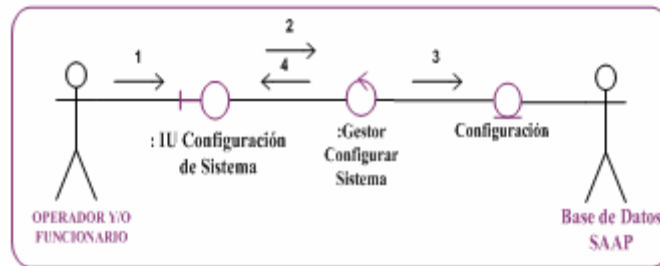


FIGURA 3.25. Diagrama de Colaboración Configurar Sistema

[Fuente: Propia]

El actor Administrador puede configurar datos del sistema. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Configuración Sistema. En ésta interfaz, el usuario a través del mensaje (2), solicita configuración del sistema, activando de esta manera, al: Gestor Configurar Sistema.

Seguidamente, mediante el mensaje (3) el: Gestor Configurar Sistema, busca y procesa la información para que finalmente se muestre al usuario.

En la *tabla 3.18.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Configurar Sistema.

TABLA 3.18. Mensajes del Diagrama de Colaboración Configurar Sistema

LEYENDA	
1.	Solicitar Configuración de Sistema.
2.	Procesar Configurar Sistema.
3.	Buscar Configurar Sistema.
4.	Mostrar Configuración del Sistema.

[Fuente: Propia]

3.7. Diseño.

En la fase de inicio, el objetivo principal del flujo de trabajo de diseño, es desarrollar un esbozo inicial del modelo de diseño, es decir, un primer paso para la vista de la arquitectura del modelo de diseño que realice los casos de uso anteriormente identificados en el flujo de trabajo de requisitos, como colaboraciones entre subsistemas o clases.

3.7. 1 Identificación de los Subsistemas.

El Sistema Administrativo del Área de Postgrado (*SAAP*) será desarrollado bajo el Sistema Operativo Microsoft Windows. Para realizar la codificación de los módulos que forman parte del sistema *SAAP*, se hará uso de la herramienta de Programación PHP, un lenguaje de programación utilizado en la elaboración de

aplicaciones para ambientes Web.

Las aplicaciones son ejecutadas a través del Servidor EasyPHP, un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol). La manipulación de los datos del sistema se realizara con la ayuda del manejador de base de datos ORACLE. En el siguiente diagrama muestra la arquitectura del sistema, y engloba los diferentes subsistemas existentes en el Sistema Administrativo del Área de Postgrado.

En la *Figura 3.26.*, se representan las cuatro capas donde se distribuye la funcionalidad del sistema, junto con los diferentes subsistemas que lo conforman. La *Capa Específica de la Aplicación*, engloba los paquetes de análisis que constituyen el SAAP. En la *Capa General*, se encuentra el paquete de datos de sistema, donde se encuentran las Base de Datos relacionadas con el proyecto. En la *Capa Intermedia*, se encuentra el servidor Web (EasyPHP), el lenguaje de programación para aplicaciones web PHP, Dreamweaver la biblioteca de código abierto para php, el sistema de gestión de bases de datos (ORACLE) , y por ultimo en la *Capa de Software* se ubica el Sistema Operativo y el protocolo TCP/IP.

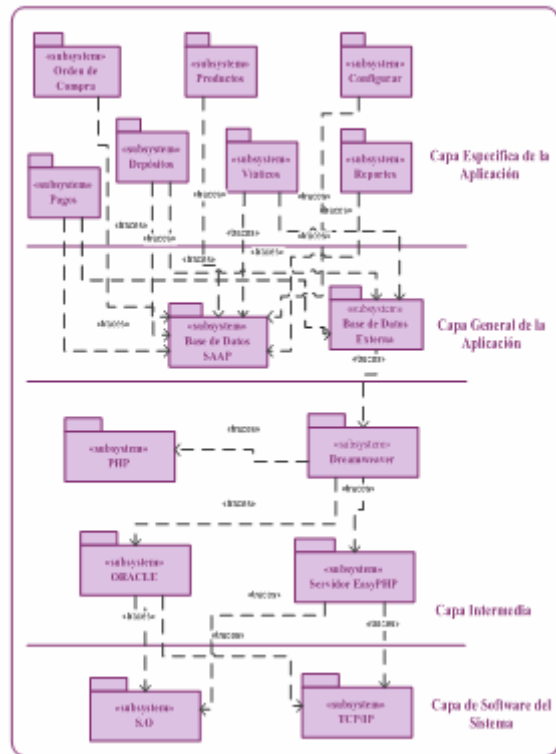


FIGURA 3.26. Distribución de los Subsistemas

[Fuente: Propia]

3.8 Planificación de las Sigüientes Fases.

3.8.1 Fase de Elaboración.

En la fase de elaboración se formula la línea base de la arquitectura del sistema, completando el modelo de análisis y el modelo de diseño realizado durante la fase de inicio, esto implica recopilar el mayor porcentaje de requisitos pendientes, formulando los requisitos funcionales como casos de uso, y abordando los riesgos que interfieran en la consecución de este objetivo. Además se diseña, se crea y

construye la base de datos que requiere el sistema.

3.8.2 Fase de Construcción.

En la fase de construcción se deja listo un producto software en su versión operativa inicial, mediante la codificación de los módulos generados a través de los flujos de trabajo realizado en la fase de elaboración. Para ello, se detallan los casos de uso y escenarios restantes, modificando si es necesaria la descripción de la arquitectura. En esta fase, se dejan cerrados los modelos de análisis, diseño e implementación.

3.8.3 Fase de Transición.

En la fase de transición se coloca en funcionamiento el sistema, incorporando e integrando sus componentes. Además se debe seleccionar y adiestrar al personal encargado de probar las versiones beta, preparar las instrucciones de pruebas y realizar una versión preliminar del manual de usuario, detallado para guiar a los usuarios de la beta.

3.9 Resumen de la Fase de Inicio.

El fin principal de la fase de inicio, es realizar un análisis inicial del proyecto, para determinar el ámbito del sistema propuesto y esbozar una arquitectura candidata. Basándose, en la ejecución de los flujos de trabajo fundamentales para el desarrollo de las fases de desarrollo de software, se llevó a cabo el primer flujo de trabajo, el de *requisitos*, donde residió la mayor parte de trabajo de ésta fase, ya que se enumeraron los requisitos candidatos, se esbozó un modelo de dominio que permitiera comprender el contexto del sistema, y se elaboró una lista de riesgos críticos que

pondrían en peligro la viabilidad del sistema, por último, en este flujo se representaron los requisitos funcionales mediante la identificación y descripción de los casos de uso más relevantes y de sus actores principales, para obtener como resultado un primer modelo de casos de uso detallado.

Una vez completados los aspectos que engloban la recopilación de requisitos, se avanzó al siguiente flujo de trabajo, el de *análisis*, donde los casos de usos ya diseñados, fueron detalladamente analizados por medio de los paquetes de análisis, de las clases de análisis, y de los diagramas de colaboración, con el fin de obtener un modelo inicial de análisis, que servirá como primer paso para la vista de la arquitectura del modelo de análisis.

Ya cumplidos los objetivos planteados para la fase de inicio, se pudieron obtener resultados favorables que establecían que es factible la puesta en marcha del sistema propuesto.

CAPÍTULO IV

FASE DE ELABORACIÓN

4.1 Introducción

Cumpliendo satisfactoriamente con los objetivos previstos en la Fase de Inicio, se prosigue a darle entrada a la segunda fase del Proceso Unificado de Desarrollo de Software llamada *Fase de Elaboración*. El propósito de esta etapa es crear la línea base de la arquitectura para así disponer de unos cimientos sólidos sobre los que se basaran el diseño y la implementación durante la Fase de Construcción. La arquitectura evoluciona considerando los requisitos más significativos (aquellos que tienen un fuerte impacto en la arquitectura del sistema) y la evaluación de riesgo.

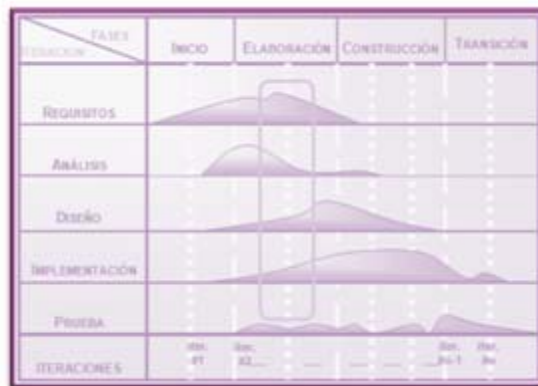


FIGURA 4.1. Flujos de Trabajo Fundamental en la Fase de Elaboración

[Fuente: Jacobson, I.]

Dentro de esta fase, el objetivo primordial es formular la línea base de la

arquitectura del proyecto, para guiar el trabajo durante las Fases de Construcción y Transición, así como en las posteriores generaciones del sistema. Esto implica, desarrollar el mayor porcentaje de los casos de usos, basándose en la comprensión del sistema en su totalidad, su ámbito y sus requisitos funcionales y no funcionales, abordando los riesgos que interfieran en la consecución de los objetivos.

4.2 Planificación de la Fase de Elaboración

Durante el desarrollo de la Fase de Elaboración se abordaran los flujos de trabajo fundamentales correspondientes a ésta fase, tomando como base los resultados obtenidos de la fase anterior. De esta manera, se recopilará, analizará, diseñará, implementará y probará únicamente los requisitos relevantes desde el punto de vista de la arquitectura.

Para la *recopilación de requisitos*, se estructuran el conjunto de casos de uso y actores adicionales a aquellos descritos en la fase de inicio, para ello, es preciso que sean identificados la mayor cantidad de estos. Otra de las actividades que tiene lugar en este flujo es la identificación de las interfaces de usuario.

En el segundo flujo, el de *análisis* de la arquitectura, se analizan los casos de uso, las clases y los paquetes respectivos hasta el extremo de que pueda darle pie a una línea base de la arquitectura ejecutable.

Siguiendo con el tercer flujo de trabajo, el de *diseño* de la arquitectura, donde se diseñarán los casos de uso, clases y subsistemas que sean arquitectónicamente significativos.

Luego, el flujo de *implementación* de la arquitectura, donde se implementará

menos del 10% de los casos de uso, además de las clases y la implementación del subsistema así como también la integración del sistema.

Por último, se cubrirán las *pruebas*, donde el objetivo es asegurarse de que los subsistemas de todos los niveles (subsistemas del servicio y los subsistemas del diseño) y de todas las capas, desde las capas del sistema hasta las capas específicas de la aplicación funcionen adecuadamente.

4.3 Identificación de Riesgos.

Durante la Fase de Elaboración se tratará de atenuar o reducir a un nivel aceptable para la fase de construcción la mayoría de los riesgos identificados durante la Fase de Inicio. En esta fase se afrontarán los riesgos técnicos de la arquitectura del sistema; teniendo en cuenta que no serán eliminados hasta que los casos de usos relacionados con cada uno de ellos estén implementados.

4.4 Captura de Requisitos.

A medida que se avanza en el desarrollo del Sistema Administrativo del Área de Postgrado (SAAP) surgen nuevos requisitos que se traducen en casos de usos adicionales a aquellos, que fueron identificados en la fase de inicio. Dentro de esta fase, el flujo de trabajo Requisitos busca identificar el mayor porcentaje de los casos de uso que permitan llegar a la arquitectura final deseada del sistema.

4.4.1 Lista de Requisitos.

A continuación se señalan cada uno de los requisitos funcionales que se han adicionado en esta fase.

4.4.1.1. Requisitos Funcionales.

Para completar totalmente la descripción del sistema en base a su operabilidad y necesidades específicas, se determino el siguiente requisito:

Administrar Usuarios: el sistema debe permitir el control de los usuarios de las distintas Coordinaciones de Postgrado; también debe permitir que los usuarios puedan cambiar su contraseña cuando sea necesario.

4.4.1.2. Requisitos No Funcionales.

Para la Fase de Elaboración no se encontraron requisitos no funcionales, ya que los requerimientos necesarios fueron identificados en la fase de inicio.

4.4.2 Modelo de Caso de Uso.

Los requisitos que se han adicionado en esta fase, traen consigo la identificación de nuevos casos de uso para el Sistema SAAP. En el diagrama de casos de uso del sistema (*tabla 4.1*) se puede observar resaltado el nuevo caso de uso (general).

4.4.2.1 Identificación de Caso de Uso.

TABLA 4.1. Caso de Uso del Sistema.

CASO DE USO	DESCRIPCIÓN	ACTOR
Administrar Usuarios	Este caso de uso permite el control de cada uno de los usuarios que desee ingresar al sistema; así como también el usuario puedan solicitar el cambio de su contraseña cuando sea necesario.	Operador y/o Funcionario, Bases de Datos SAAP

[Fuente: Propia]

4.4.2.2. Diagrama de Caso de Uso.

En esta fase los actores mantienen las mismas relaciones con los procesos vinculados al sistema, la diferencia se establece debido a que se ha identificado e incorporado el caso de uso *Administrar Usuarios* al diagrama de caso de uso general del sistema *SAAP*. En la *figura 4.2*. Se muestra la representación general del Sistema Administrativo del Área de Postgrado (*SAAP*) con el nuevo caso de uso. (Ver figura 4.2)

4.4.2.3. Casos de Uso Detallados.

El diagrama de caso de uso anterior es una representación general del sistema, a continuación se detallará el caso de uso que fue agregado.

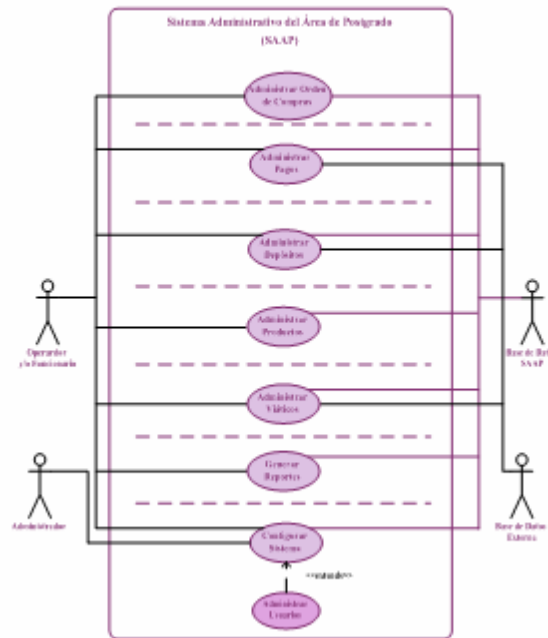


FIGURA 4.2. Diagrama de Casos de Uso del Sistema SAAP

[Fuente: Propia]

Caso de Uso Administrar Usuarios.

Se puede observar en el diagrama de caso de uso de la *figura 4.3.*, como se desarrolla el proceso de Administrar Usuarios. En este caso de uso el usuario podrá realizar cambios a los datos que son usados como parámetro para su registro.



FIGURA 4.3. Caso de Uso Administrar Usuarios

[Fuente: Propia]

Descripción del Flujo de Sucesos del Caso de Uso Administrar Usuarios.

Caso de Uso 9: *ADMINISTRAR USUARIOS*

Actores Participantes: Operador y/o Funcionario, Administrador, Base de Datos *SAAP*.

Descripción: Corresponde al proceso de administración de los usuarios que interactúan dentro del sistema *SAAP*; será operado por el actor, quien tendrá a su cargo la ejecución de los procesos de registro, modificación y/o eliminación de los usuarios; así como también el cambio de contraseña cuando sea necesario.

Condiciones de Entrada: El actor activa el Caso de Uso, seleccionando el modulo Configurar Sistema del Menú Principal, la opción Administrar Usuarios.

Flujo de Eventos:

Flujo Principal

1. El actor activa el Caso de Uso, seleccionando la opción Administrar Usuarios.
2. Luego de esto, se carga en pantalla el listado de los usuarios registrados en el

sistema, o si el usuario desea realizar una búsqueda más específica, procede a seleccionar en la lista de menú, los parámetros de búsqueda y se muestra en pantalla el formulario con los datos de la búsqueda especificada.

3. A partir del formulario retornado a través del número de cédula, el usuario podrá detallar su contenido, modificar la data, actualizar o eliminarlo; ajustándose a las acciones necesarias para que el proceso que haya seleccionado, se lleve a cabo satisfactoriamente.

4. En el caso de que el actor desee realizar un nuevo registro, selecciona la opción Registrar Usuarios del menú de Opciones que se mostrará en una interfaz principal.

5. El actor ingresa los datos requeridos.

6. El actor confirma que los datos ingresados sean los correctos presionando el botón Guardar.

7. El sistema valida que los datos hayan sido correctamente ingresados.

8. Finalmente, el registro es almacenado dentro de la Base de Datos del sistema SAAP.

9. Fin del Caso de Uso.

Condiciones de Salida: Seleccionar otra opción del menú ó presionar el botón Cerrar Sesión.

4.5. Análisis.

En la fase de inicio se analizó un pequeño porcentaje del sistema, por lo que durante la fase de elaboración, se profundizará en esta actividad y se analizarán los nuevos casos de uso que se identifiquen desde el punto de vista de la arquitectura, para lograr su comprensión en función de las clases del análisis y sus objetos en iteración. Por último, se abordará la actividad del análisis de paquetes del sistema.

4.5.1. Análisis de la Arquitectura.

En esta fase se ampliará el análisis de la arquitectura hasta encontrar la línea base de una arquitectura ejecutable. El propósito es proyectar el modelo de análisis, mediante la identificación de paquetes de análisis. Se realizaran los paquetes de análisis, partiendo de la vista de la arquitectura del modelo de caso de uso, se elaborará una arquitectura en capas, explicando los paquetes específicos de la aplicación y los paquetes generales.

4.5.1.1. Identificación de Paquetes de Análisis.

Los paquetes de análisis identificados en la fase anterior no sufrieron ningún cambio, a excepción del paquete de análisis Configurar.

Paquete de Análisis Configurar

Este paquete de análisis contiene los procesos ó métodos para gestionar las acciones de Configurar Sistema y el proceso adicionado en esta fase Administrar Usuarios. Manteniendo una relación directa hacia sus respectivos casos de uso.

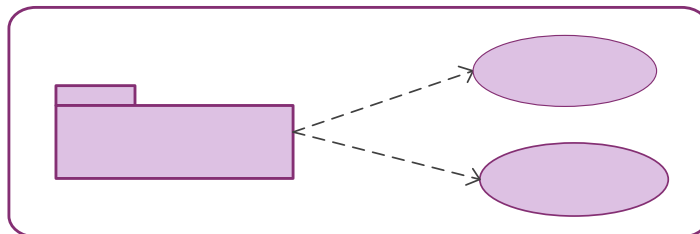


FIGURA 4.4. Paquete de Análisis Configurar

[Fuente: Propia]

Paquete de Análisis Datos del Sistema.

Este paquete de análisis contiene la información necesaria para la realización de cada uno de los procesos que contempla el sistema *SAAP*.



FIGURA 4.5. Paquete de Análisis Datos del Programa

[Fuente: Propia]

4.5.2. Análisis de los Casos de Uso.

Con la identificación de nuevos casos de usos, es necesario refinar el modelo de análisis para el Sistema Administrativo del Área de Postgrado *SAAP*, por lo que se realizarán modificaciones en uno de los diagramas mostrados en la fase anterior, y se detallará la realización de los otros casos de uso del sistema.

4.5.2.1. Identificación de las Clases de Análisis.

A continuación se identifican las clases de análisis correspondientes al caso de uso Administrar Usuarios, el cual se incorporo en esta fase.

Clases de Control.

Las clases de control que encapsulan el control de los casos de usos que se incorporaron al modelo de análisis para esta fase, son las descritas en las tablas mostrada a continuación.

TABLA 4.2. Clases de Control del Caso de Uso Administrar Usuarios

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Registrar Usuarios	Permite el registro de un nuevo usuario, pero solo el de usuarios de clase interno.
Gestor Modificar Usuarios	Permite modificar los datos e información referente a un usuario de un registro específico.
Gestor Eliminar Usuarios	Permite la eliminación de un usuario del sistema de la clase interno.
Gestor Cambiar Contraseña	Este gestor permite validar el cambio de la contraseña un usuario.

[Fuente: Propia]

Clases de Entidad.

Las clases de entidad que se describen en la *Tabla 4.4*. Corresponden a las identificadas dentro de la iteración Análisis para esta fase.

TABLA 4.3. Clases de Entidad del Sistema

CLASE DE ENTIDAD	DESCRIPCIÓN
Usuarios	Son los actores que van a interactuar con los distintos módulos y aplicaciones del sistema, según el tipo de usuario al cual correspondan.
Contraseña	Representa la forma de autenticación que utiliza información secreta para controlar el acceso de los usuarios hacia el sistema.

[Fuente: Propia]

Clases de Interfaz

Las clases de interfaz que permiten modelar la interacción entre el sistema y los actores que interactúan con los casos de uso identificados dentro de esta fase, son detalladas a continuación.

TABLA 4.4. Clases de Interfaz del Sistema

CLASE DE INTERFAZ	DESCRIPCIÓN
:IU Principal Administración de Usuarios	Representa la interfaz donde el usuario podrá interactuar con las aplicaciones asociadas con la administración de los usuarios del sistema; así como también en el cambio de contraseña.

[Fuente: Propia]

4.5.2.2. Diagrama de Clase de Análisis.

Dentro del modelo de análisis en la Fase de Elaboración, los diagramas de clases de análisis que describen como se lleva a cabo y se ejecuta el caso de uso Administrar Usuarios se detalla a continuación:

Diagrama de Clase de Análisis para el Caso de Uso Administrar Usuarios.

Este caso de uso permite la Administración de Usuarios, enfocando su funcionalidad en el registro, modificación, eliminación y cambio de contraseña, con el fin de controlar la información referente a los usuarios.

En la *figura 4.6*, se muestra el diagrama de clase de análisis que describe como se lleva a cabo el caso de uso Administrar Usuarios.

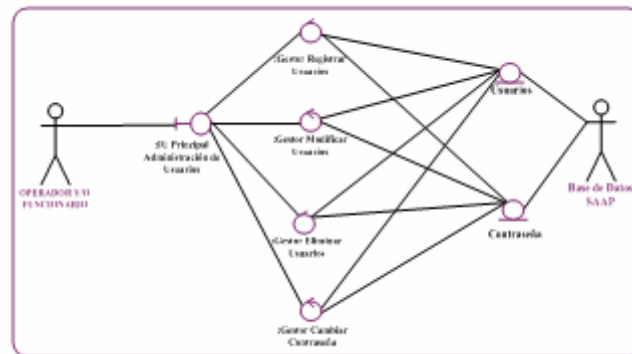


FIGURA 4.6. Diagrama de Clase de Análisis Administrar Usuarios

[Fuente: Propia]

Como se puede observar, el usuario Operador y/o Funcionario activa el caso de uso mediante la: IU Principal Administración de Usuarios donde se muestra la lista de

los usuarios que están registrado en el sistema, además las diferentes opciones que le permiten interactuar con los gestores: Gestor Registrar Usuarios, Gestor Modificar Usuarios, Gestor Eliminar Usuarios y el Gestor Cambiar Contraseña. Para los procesos de registro, modificación, eliminación de usuarios y cambio de contraseña, las entidades vinculadas son Usuarios y Contraseña.

4.5.2.3. Diagrama de Colaboración.

Una vez detallada la estructura interna del sistema mediante los diagramas de clases de análisis identificados, se modelan las secuencias de las acciones e interacciones que existen entre los distintos objetos de análisis y se muestran los mensajes que pasan de un objeto a otro, a través de los *diagramas de colaboración*. Dichos diagramas se visualizan a continuación.

Diagrama de Colaboración para el Caso de Uso Administrar Depósitos.

En la *figura 4.7* se observa el diagrama de colaboración del caso de uso Administrar Depósitos, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

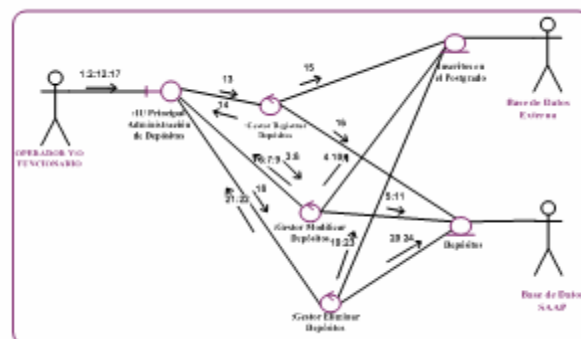


FIGURA 4.7. Diagrama de Colaboración Administrar Depósitos

[Fuente: Propia]

Un operador del sistema puede realizar una búsqueda avanzada de los depósitos, con el fin de modificar específicamente algún dato de los inscritos en el postgrado y sus respectivos depósitos. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración de Depósitos. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Depósitos a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través de numero de depósitos) para seleccionar el Registro con los datos correspondientes. Posteriormente a través del mensaje (4 y 5) se busca el Registro ya sea de los Inscritos en Postgrado y Depósitos seleccionado en las entidades de Inscritos en Postgrado y Depósitos.

El :Gestor Modificar Depósitos mediante el mensaje (6 y 7) muestra el Registro seleccionado. Seguidamente se suministra los nuevos datos del registro modificado, según el mensaje (8).

El :Gestor Modificar Depósitos valida la información modificada según el mensaje (9), para que finalmente sea almacenada dentro de las entidades Inscritos en Postgrado a través del mensaje (10) y Depósitos a través del mensaje (11).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 4.5.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso

Administrar Depósitos.

TABLA 4.5. Mensajes del Diagrama de Colaboración Administrar Depósitos

LEYENDA	
1. Activar IU Principal Administración de Depósitos.	12. Solicitar Registro de un nuevo Depósito.
2. Solicitar Modificar Registro de Depósitos.	13. Cargar data para el formulario del nuevo Registro de Depósitos.
3. Cargar data para seleccionar el Registro de Depósito (inscrito en postgrados u otros depósitos).	14. Validar Registro de Depósitos.
4. Buscar Registro de los depósitos de los inscritos en postgrado seleccionado.	15. Registrar Inscritos en el Postgrado.
5. Buscar Registro de depósito seleccionado.	16. Registrar Depósitos.
6. Mostrar Registro de los depósitos de los Inscritos en postgrado seleccionado.	17. Solicitar Eliminación de un Registro de Depósito.
7. Mostrar Registro de los depósitos seleccionados.	18. Cargar datos para seleccionar el registro a eliminar.
8. Cargar datos nuevos.	19. Buscar Registro de los depósitos de los inscritos en postgrado seleccionado.
9. Validar datos nuevos.	20. Buscar Registro de depósito seleccionado.
10. Guardar registro modificado de los depósitos de los inscritos en el postgrado.	21. Mostrar Registro de los depósitos de los Inscritos en postgrado seleccionado.
11. Guardar registro modificados de los depósitos.	22. Mostrar Registro de los depósitos seleccionados.
	23. Eliminar registro de los depósitos de los Inscritos en el Postgrado.
	24. Eliminar Depósitos.

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Administrar Productos.

En la *figura 4.8* se observa el diagrama de colaboración del caso de uso Administrar Productos, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

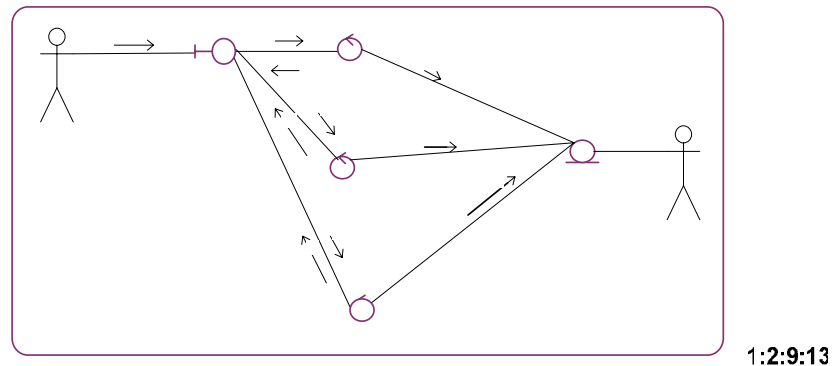


FIGURA 4.8. Diagrama de Colaboración Administrar Productos

[Fuente: Propia]

:IU Principal
Administración de
Productos

Un operador del sistema puede realizar una búsqueda avanzada de los productos, con el fin de modificar específicamente algún dato de los mismos. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración Productos. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Productos a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través del código de productos) para seleccionar el Registro de Productos con los datos correspondientes. Posteriormente a través del mensaje (4) se busca el Registro Productos seleccionado en la entidad Productos.

El :Gestor Modificar Productos mediante el mensaje (5) muestra el Registro Productos seleccionado. Seguidamente se suministra los nuevos datos del registro productos modificados según el mensaje (6).

El :Gestor Modificar Productos valida la información modificada según el

mensaje (7), para que finalmente sea almacenada dentro de la entidad Productos a través del mensaje (8).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 4.6.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Administrar Productos.

TABLA 4.6. Mensajes del Diagrama de Colaboración Administrar Productos.

LEYENDA	
1. Activar IU Principal Administración de Productos.	10. Cargar data para el formulario del Registro de Producto.
2. Solicitar Modificación de un Producto.	11. Validar Registro del Producto.
3. Cargar data para seleccionar el Registro de Producto a modificar.	12. Registrar Productos.
4. Buscar Registro de Producto Seleccionado.	13. Solicitar Eliminación de un Registro de Producto.
5. Mostrar Registro de Producto Seleccionado	14. Cargar data para seleccionar el Registro de Producto a eliminar
6. Cargar nuevos datos	15. Buscar Registro de Producto Seleccionado.
7. Validar nuevos datos de Producto.	16. Mostrar Registro de Producto Seleccionado
8. Guardar Registro de Producto modificado	17. Eliminar Productos.
9. Solicitar Registro de un nuevo Producto.	

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Administrar Viáticos.

En la *figura 4.9* se observa el diagrama de colaboración del caso de uso Administrar Viáticos, mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

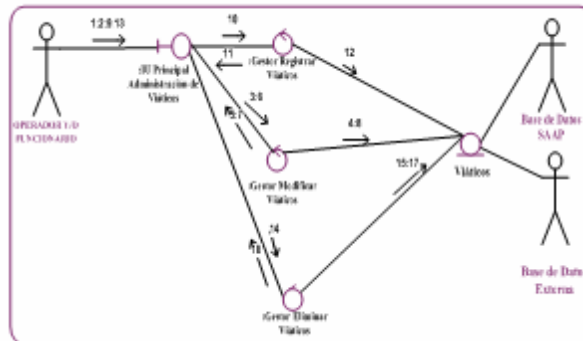


FIGURA 4.9. Diagrama de Colaboración Administrar Viáticos

[Fuente: Propia]

Un operador del sistema puede realizar una búsqueda avanzada de los viáticos, con el fin de modificar específicamente algún dato de los mismos. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración Viáticos. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Viáticos a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través del número de Viáticos) para seleccionar el Registro de Viáticos con los datos correspondientes. Posteriormente a través del mensaje (4) se busca el Registro Viáticos seleccionado en la entidad Viáticos.

El :Gestor Modificar Viáticos mediante el mensaje (5) muestra el Registro Viáticos seleccionado. Seguidamente se suministra los nuevos datos del registro viáticos modificados según el mensaje (6).

El :Gestor Modificar Viáticos valida la información modificada según el

mensaje (7), para que finalmente sea almacenada dentro de la entidad Viáticos a través del mensaje (8).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 4.7.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Administrar Viáticos.

TABLA 4.7. Mensajes del Diagrama de Colaboración Administrar Viáticos

LEYENDA	
1. Activar IU Principal Administración de Viáticos.	10. Cargar data para el formulario del Registro de Viático.
2. Solicitar Modificación de un Registro de Viáticos.	11. Validar Registro nuevo del Viático.
3. Cargar data para seleccionar el Registro del Viático para modificar.	12. Registrar los nuevos datos del Viáticos.
4. Buscar Registro de Viáticos seleccionado.	13. Solicitar Eliminación de un Registro de Viáticos.
5. Mostrar Registro de Viático seleccionado.	14. Cargar data para seleccionar el Registro del Viático para eliminar.
6. Cargar nuevos datos.	15. Buscar Registro de Viáticos seleccionado.
7. Validar nuevos datos del Viático.	16. Mostrar Registro de Viático seleccionado.
8. Guardar registro de Viáticos modificado.	17. Eliminar Viáticos.
9. Solicitar Registro de un nuevo Viático.	

[Fuente: Propia]

Diagrama de Colaboración para el Caso de Uso Administrar Usuarios

En la *figura 4.10.* Se observa el diagrama de colaboración del caso de uso *Administrar Usuarios.* Mostrando las interacciones entre las distintas clases de análisis que intervienen en la realización de éste caso de uso.

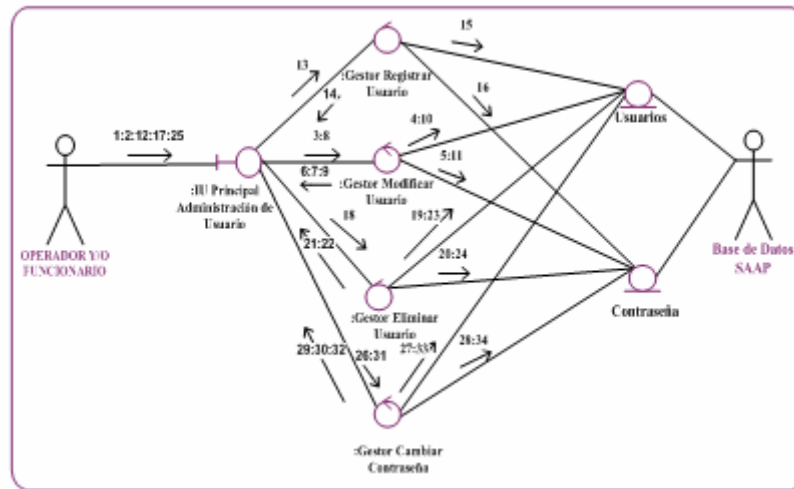


FIGURA 4.10. Diagrama de Colaboración Administrar Usuarios

[Fuente: Propia]

Un operador del sistema puede realizar una búsqueda de un usuario con el fin de modificarlo, seleccionándolo del listado. A continuación se explica paulatina y detalladamente la secuencia de interacciones entre los objetos de análisis que actúan para llevar a cabo ésta petición del usuario:

El usuario mediante el mensaje (1) activa la: IU Principal Administración de Usuarios. En ésta interfaz, el usuario activa de esta manera, el :Gestor Modificar Usuarios a través del mensaje (2).

Luego mediante el mensaje (3), se indica que cargue el dato (a través de número de usuario) para seleccionar el Registro con los datos correspondientes. Posteriormente a través del mensaje (4 y 5) se busca el Registro ya sea de los usuarios y contraseña seleccionado en las entidades de Usuarios y Contraseña.

El :Gestor Modificar Usuarios mediante el mensaje (6 y 7) muestra el Registro seleccionado. Seguidamente se suministra los nuevos datos del registro modificado, según el mensaje (8).

El :Gestor Modificar Usuarios valida la información modificada según el mensaje (9), para que finalmente sea almacenada dentro de las entidades Usuarios a través del mensaje (10) y Contraseña a través del mensaje (11).

De acuerdo a la secuencia de interacciones descrita, se da cumplimiento a la petición del usuario.

En la *tabla 4.8.*, se listan los mensajes que se envían entre los objetos del análisis, dentro del diagrama de colaboración para la realización del caso de uso Administrar Usuarios.

TABLA 4.8. Mensajes del Diagrama de Colaboración Administrar Usuarios

LEYENDA	
1. Activar IU Principal Administración de Usuarios.	20. Buscar registro de contraseña seleccionado a eliminar.
2. Solicitar Modificación de un Usuario.	21. Mostrar Registro de Usuario seleccionado.
3. Cargar dato para seleccionar el Registro de Usuario a modificar.	22. Mostrar Registro de Contraseña seleccionado.
4. Buscar registro de Usuario seleccionado.	23. Eliminar el registro de Usuario seleccionado.
5. Buscar registro de contraseña seleccionado	24. Eliminar el registro de Contraseña seleccionado.
6. Mostrar Registro de Usuario seleccionado	25. Solicitar cambio de Contraseña.
7. Mostrar Registro de Contraseña seleccionado.	26. Cargar datos para seleccionar El registro a modificar.
8. Cargar datos nuevos(usuarios y contraseñas).	27. Buscar registro de usuario seleccionado a cambiar.
9. Validar Datos Nuevos.	28. Buscar registro de contraseña seleccionado a cambiar.
10. Guardar registro de Usuario modificado.	29. Mostrar Registro de Usuario seleccionado.
11. Guardar registro de contraseña modificados.	30. Mostrar Registro de Contraseña seleccionado.
12. Solicitar registro de un nuevo usuario o contraseña.	31. Cargar datos nuevos.
13. Cargar data para El formulario del registro de usuario o contraseña.	32. Validar datos nuevos.
14. Validar Registro.	33. Guardar registro de usuario.
15. Registrar los datos del Usuario.	34. Guardar registro de contraseña.
16. Registrar Contraseña.	
17. Solicitar eliminación de Usuarios.	
18. Cargar data para seleccionar el registro a eliminar.	
19. Buscar registro de usuario seleccionado a eliminar.	

[Fuente: Propia]

4.5.3 Análisis de las Clases.

Básicamente, este análisis consiste en identificar y mantener las responsabilidades de una clase de análisis, e identificar y mantener los atributos y relaciones de las clases de análisis.

4.5.4 Análisis de los Paquetes.

El objetivo de este análisis es identificar y mantener las responsabilidades de una clase de análisis, identificar y mantener los atributos y relaciones de las clases de análisis.

Análisis del Paquete Administrar Orden de Compras.

Este paquete está vinculado con el caso de uso *Administrar Orden de Compras*. Posee las siguientes clases de análisis: IU Principal Administración de Orden de Compras, Gestor Registrar Orden de Compras, Gestor Modificar Orden de Compras, Gestor Eliminar Orden de Compras y la entidad Compras.

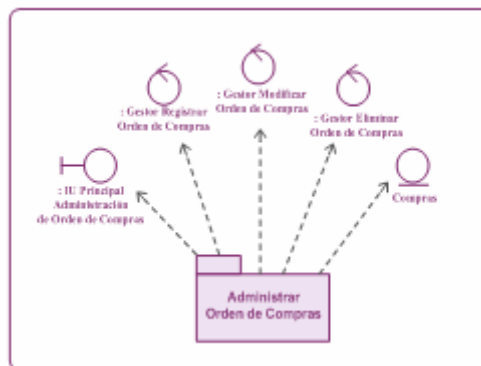


FIGURA 4.11. Clases de Análisis del Paquete Administrar Orden de Compras

[Fuente: Propia]

Análisis del Paquete Administrar Pagos.

Este paquete está vinculado con el caso de uso *Administrar Pagos*. Posee las siguientes clases de análisis: IU Principal Administración de Pagos, Gestor Registrar Pagos, Gestor Modificar Pagos, Gestor Eliminar Pagos y las entidades Proveedor, Honorario de Profesor y Pagos.

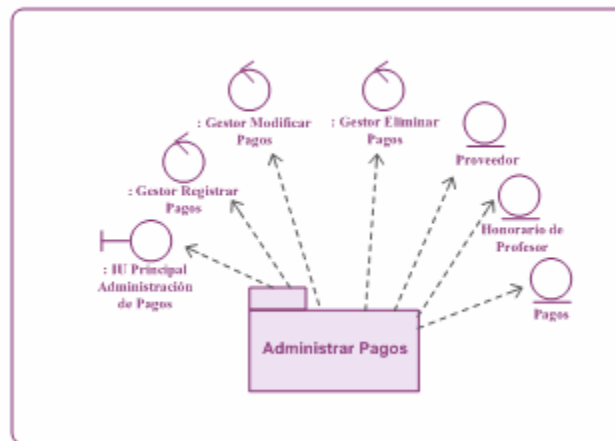


FIGURA 4.12. Clases de Análisis del Paquete Administrar Pagos

[Fuente: Propia]

Análisis del Paquete Administrar Depósitos.

Este paquete está vinculado con el caso de uso *Administrar Depósitos*. Posee las siguientes clases de análisis: IU Principal Administración de Depósitos, Gestor Registrar Depósitos, Gestor Modificar Depósitos, Gestor Eliminar Depósitos y las entidades Inscritos en el Postgrado y Depósitos.

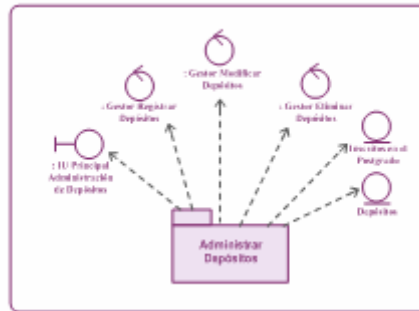


FIGURA 4.13. Clases de Análisis del Paquete Administrar Depósitos

[Fuente: Propia]

Análisis del Paquete Administrar Productos.

Este paquete está vinculado con el caso de uso *Administrar Productos*. Posee las siguientes clases de análisis: IU Principal Administración de Productos, Gestor Registrar Productos, Gestor Modificar Productos, Gestor Eliminar Productos y la entidad Productos.

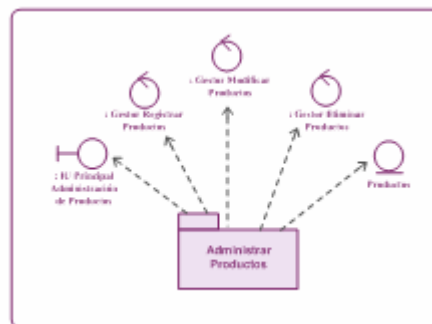


FIGURA 4.14. Clases de Análisis del Paquete Administrar Productos

[Fuente: Propia]

Análisis del Paquete Administrar Viáticos.

Este paquete está vinculado con el caso de uso *Administrar Viáticos*. Posee las siguientes clases de análisis: IU Principal Administración de Viáticos, Gestor

Registrar Viáticos, Gestor Modificar Viáticos, Gestor Eliminar Viáticos y la entidad Viáticos.

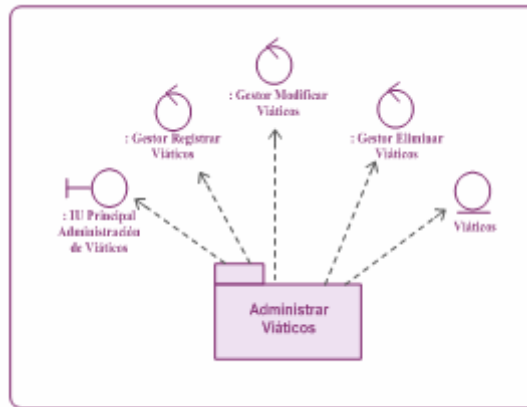


FIGURA 4.15. Clases de Análisis del Paquete Administrar Viáticos

[Fuente: Propia]

Análisis del Paquete Generar Reportes.

Este paquete está vinculado con el caso de uso *Generar Reportes*. Posee las siguientes clases de análisis: IU Generación de Reportes, Gestor Generar Reportes, y la entidad Reporte.

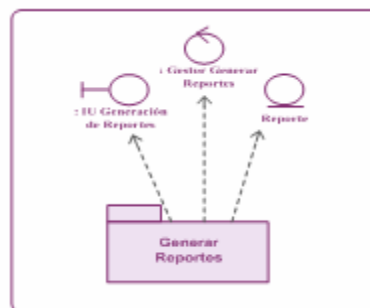


FIGURA 4.16. Clases de Análisis del Paquete Generar Reportes

[Fuente: Propia]

Análisis del Paquete Configurar Sistema.

Este paquete está vinculado con el caso de uso Configurar Sistema. Posee las siguientes clases de análisis: IU Configuración de Sistema, Gestor Configurar Sistema y la entidad Configuración.

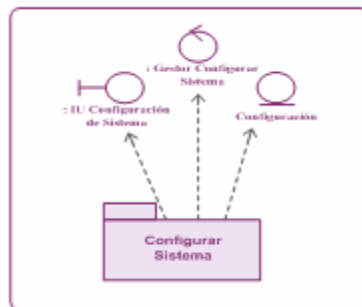


FIGURA 4.17. Clases de Análisis del Paquete Configurar Sistema

[Fuente: Propia]

Análisis del Paquete Administrar Usuarios.

Este paquete está vinculado con el caso de uso *Administrar Usuarios*. Posee las siguientes clases de análisis: IU Principal Administración de Usuarios, Gestor Registrar Usuarios, Gestor Modificar Usuarios, Gestor Eliminar Usuarios, Gestor Cambiar Contraseña, y las entidades Usuario y Contraseña.

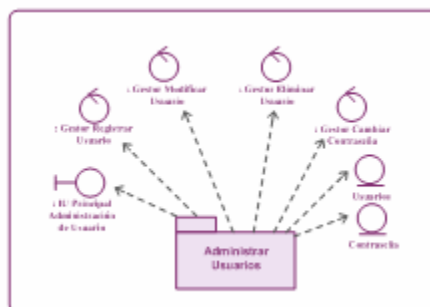


FIGURA 4.18. Clases de Análisis del Paquete Administrar Usuarios

[Fuente: Propia]

4.6 Diseño.

Durante el diseño, el resultado del análisis es expandido a una solución técnica. Se agregan nuevas clases que proveen la infraestructura del sistema: interfaces de usuario e interfaces de solicitud, manejo de bases de datos para almacenar objetos, comunicaciones con otros sistemas, etc. Se modelará el sistema y se dará forma, a fin de que soporte todos los requisitos, tomando como punto de partida el análisis que previamente fue realizado.

El diseño del sistema *SAAP* deberá crear una entrada apropiada y un punto de partida para las actividades de implementación, esto quiere decir que dicho diseño se utiliza para visualizar la implementación y para soportar las técnicas de programación.

Se realizará el diseño de la arquitectura del sistema, partiendo de la identificación de paquetes de diseño, capas y las dependencias de los subsistemas. Luego se elaborará el diagrama de clases de diseño, definiendo así una vista estática del sistema. Posteriormente, se realizará el diagrama de secuencia, que muestra una vista dinámica del sistema.

También se realizará el diseño de la base de datos del sistema *SAAP*, siguiendo los lineamientos del modelo relacional y de las formas normales y se presentará una descripción de las tablas que conforman la base de datos y sus relaciones.

4.7. Diseño del prototipo de la Interfaz de Usuario.

Al llegar a este punto, ya se conocen las principales necesidades de la interfaz

de usuario, gracias a la descripción de los casos de uso que permitieron obtener gran parte de los requerimientos del sistema, entre ellos los usuarios involucrados con el sistema y la forma como estos interactúan dentro de él. De éste modo, se hace posible proseguir a realizar una esquematización de cómo deberían ser las principales interfaces, es decir, el diseño de un prototipo de la interfaz de usuario que permita comprender y especificar las interacciones entre los usuarios y el sistema.

A continuación se muestran dos prototipos de interfaces, de inicio de sesión y la interfaz principal del sistema *SAAP*.

4.7.1 Prototipo de la Interfaz Inicio de Sesión.

Inicialmente al acceder al sistema se muestra la pantalla de logueo o inicio de sesión que le permitirá a un usuario empezar a interactuar con el sistema. Simplemente se divide en distintas áreas, el área superior se encuentra el logo de la Universidad y el logo del sistema *SAAP* y en el centro estará ubicado el área de autenticación de los usuarios. Observe el prototipo de la interfaz inicio de sesión en la *figura 4.19*.



FIGURA 4.19. Prototipo de la Interfaz Inicio de Sesión

[Fuente: Propia]

4.7.2 Prototipo de la Interfaz Principal.

Se ha diseñado un prototipo de interfaz principal que permita al usuario invocar todos los casos de uso del sistema e interactuar con las interfaces de cada uno de ellos, esto a través un menú desplegable que brinda un rápido acceso a las operaciones básicas que ofrece el sistema. Observe la *figura 4.20*.

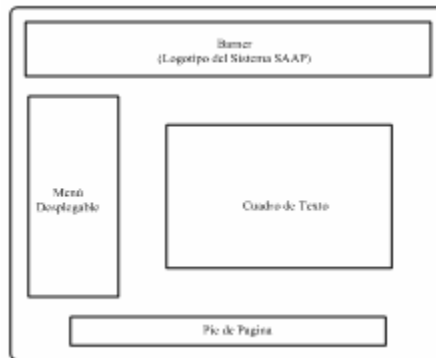


FIGURA 4.20. Prototipo de la Interfaz de Usuarios

[Fuente: Propia]

Identificación de los Subsistemas.

A continuación se identifican los subsistemas que forman parte de las capas: específica, general e intermedia de la aplicación, y de software del sistema *SAAP*, cabe destacar que la capa específica de la *figura 3.21* de la fase de inicio fue modificada en esta Fase de Elaboración

La *figura 4.21*, muestra la distribución de los subsistemas que conforman el sistema *SAAP*, dichos subsistemas se dividen en cuatro capas donde se distribuye la funcionalidad del sistema *SAAP*, En la *Capa Específica de la Aplicación* se despliegan los diferentes paquetes de análisis que constituyen el proyecto agregando el subsistema y administrar usuarios en el cual está contenido en el subsistema

configurar.

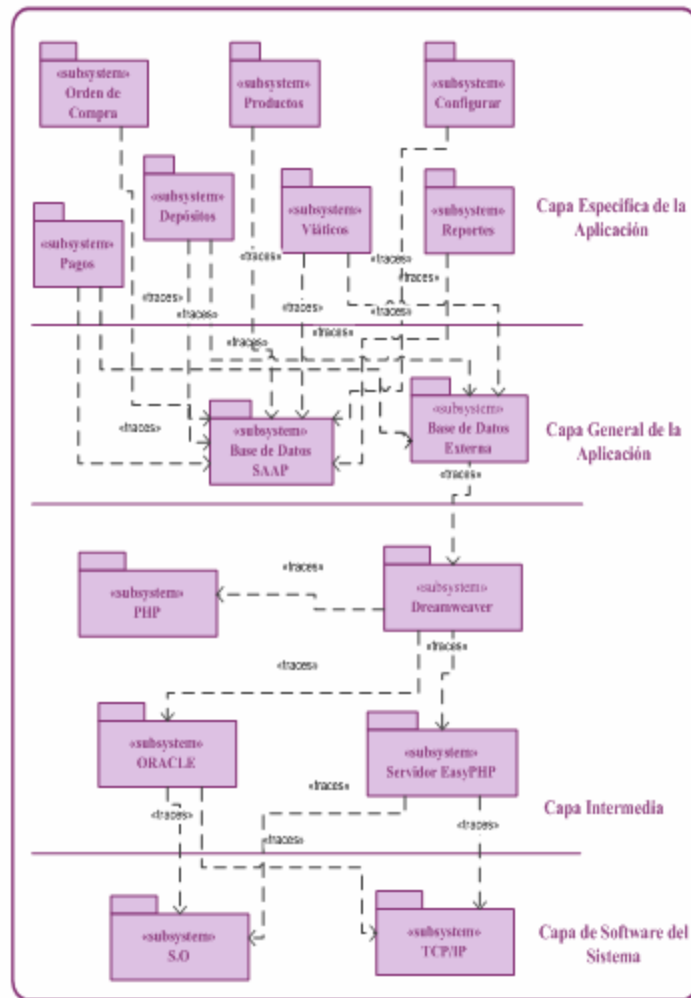


FIGURA 4.21. Distribución de los Subsistemas

[Fuente: Propia]

4.7.3 Modelo de Diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de usos centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema en estudio. En este modelo, los casos de usos son realizados por las clases de diseño y sus objetos.


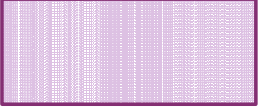



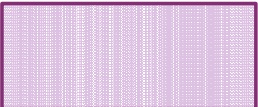
4.7.3.1. Identificación de las Clases de Diseño.

Las clases de diseño definen los atributos y operaciones con que debe contar cada clase de análisis para llevar a cabo sus responsabilidades, así como las relaciones existentes entre ellas. En este paso se convierten las clases de análisis más significativas para la arquitectura del sistema *SAAP*, en clases de diseño.

A continuación se esbozan las clases activas, considerando como entrada los resultados obtenidos del análisis, para determinar la correspondencia entre los diseños de las respectivas clases.

Clases de Diseño a partir de Clases de Control.

TABLA 4.9. Clases de Diseño a partir de Clases de Control (1/4)

CLASES DE ANALISIS	CLASES DE DISEÑO
	
	
	

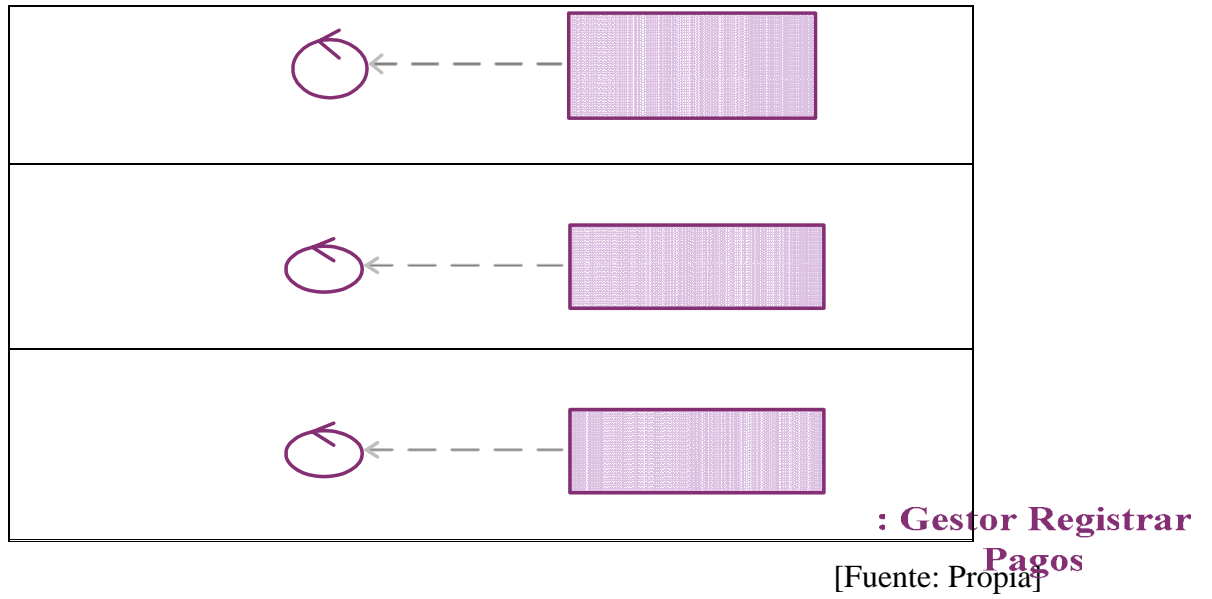
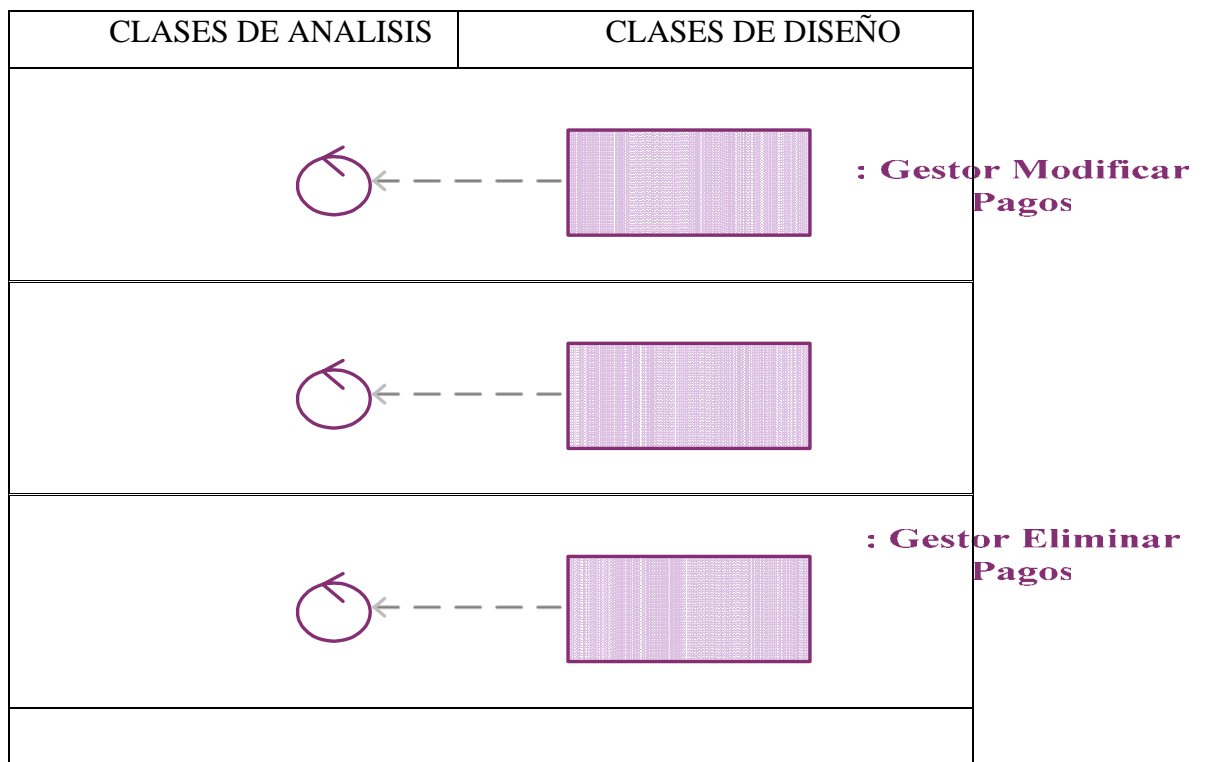
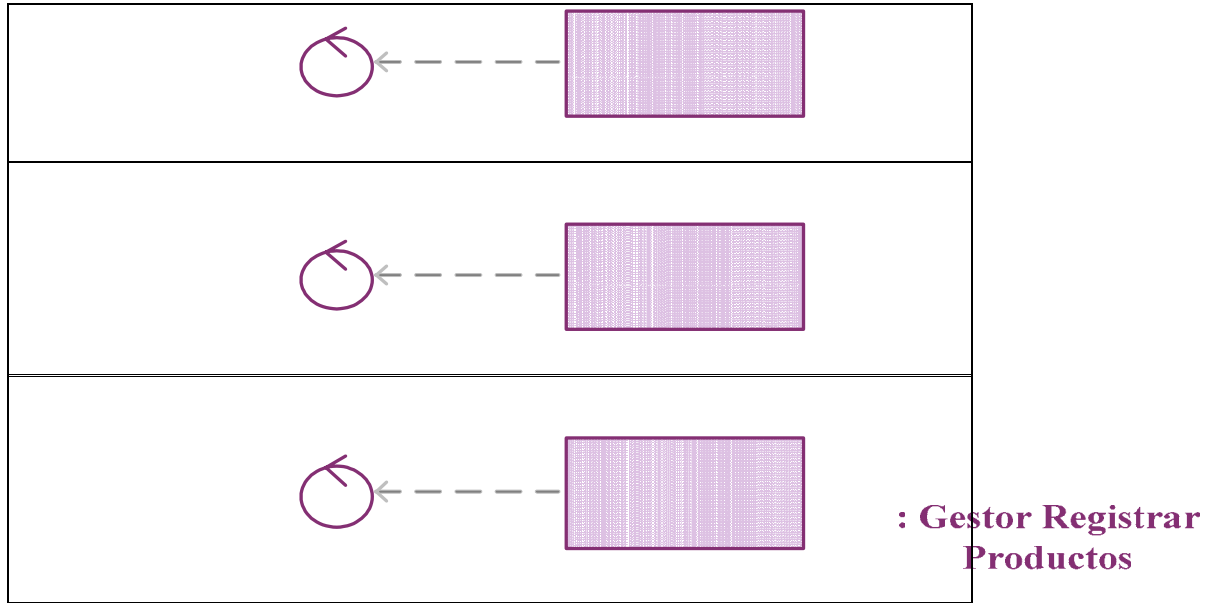







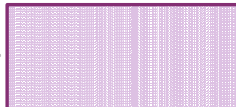
TABLA 4.10. Clases de Diseño a partir de Clases de Control (2/4)

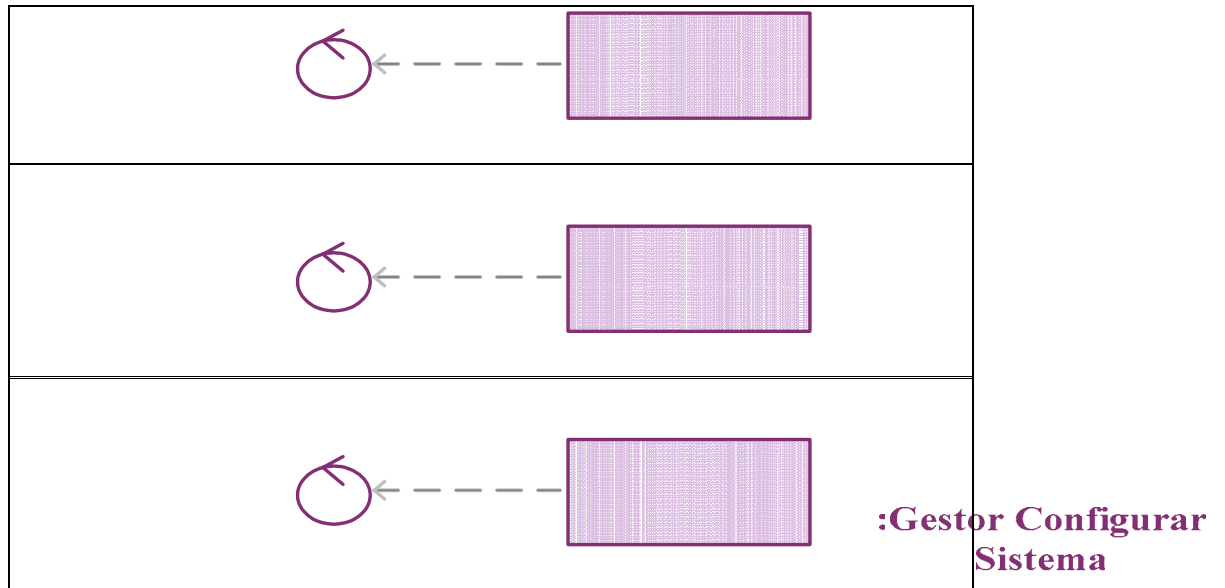




[Fuente: Propia]


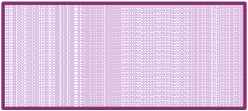

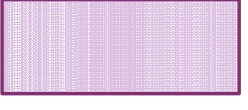
TABLA 4.11. Clases de Diseño a partir de Clases de Control (3/4)

CLASES DE ANALISIS	CLASES DE DISEÑO	
		: Gestor Modificar Productos
		
		: Gestor Eliminar Productos



[Fuente: Propia]







TABLA 4.12. Clases de Diseño a partir de Clases de Control (4/4)

CLASES DE ANALISIS	CLASES DE DISEÑO
	 : Gestor Registrar Usuarios
	 : Gestor Modificar Usuarios

[Fuente: Propia]





: Gestor Modificar Usuarios

TABLA 4.13. Clases de Diseño a partir de Clases de Interfaz (1/2)

CLASES DE ANALISIS	CLASES DE DISEÑO
	
	
	

[Fuente: Propia]

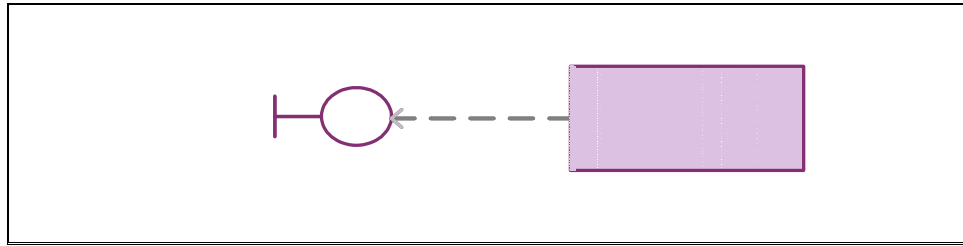
TABLA 4.14. Clases de Diseño a partir de Clases de Interfaz (2/2)

**: IU Principal
Administración de
Compras**











**: IU Principal
Administración de
Pagos**

**: IU Principal
Administración de
Depósitos**



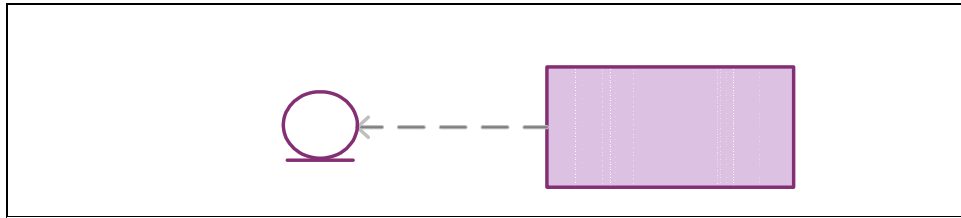
[Fuente: Propia]

TABLA 4.15. Clases de Diseño a partir de Clases de Entidad (1/2)

CLASES DE ANALISIS	CLASES DE DISEÑO
	
	
	
	
	




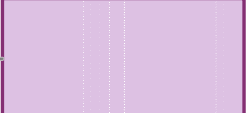

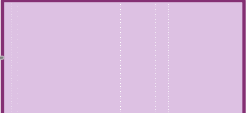



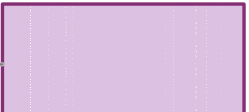
«
: IU Principal
Administración de
Usuarios

«tra
Compras



[Fuente: Propia]

TABLA 4.16. Clases de Diseño a partir de Clases de Entidad (2/2)

CLASES DE ANALISIS	CLASES DE DISEÑO
	
	
	
	
	

[Fuente: Propia]

Productos

Viáticos

<<tr

<<tr

4.7.3.2. Diagrama de Clase de Diseño.

El diagrama de clase de diseño permite visualizar las clases que componen el sistema, la forma como se relacionan dichas clases y la estructura estática de los casos de uso. De modo que, permitan mostrar lo que el sistema puede hacer, además de cómo puede ser construido.

Para la realización de éste diagrama se toman como base, las clases de análisis establecidas en la iteración anterior (*Análisis*), y es posible agregar nuevas clases de ser necesario. Las clases de diseño se establecen tomando como referencia a las clases usadas en la implementación del software.

Los siguientes diagramas de clase de diseño que se muestran, son los correspondientes a los principales casos de uso del Sistema Administrativo del Área de Postgrado.

Diagrama de Clase de Diseño para el caso de uso Administrar Orden de Compra

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Orden de Compra; donde se dispone de una clase de interfaz: IU Principal Administración de Orden de Compra siendo las clases de gestión: Registrar Orden de Compra, Modificar Orden de Compra y Eliminar Orden de Compra, y la clase de entidad relacionada es Compras.

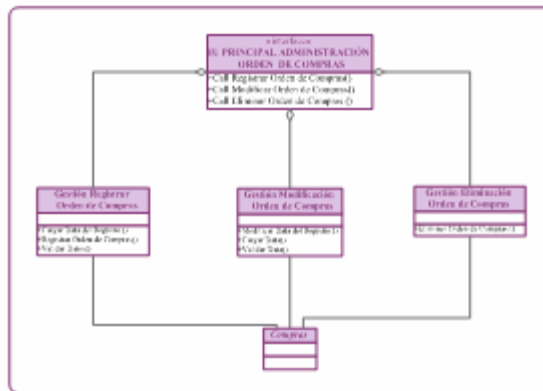


FIGURA 4.22. Diagrama de Clases de Diseño Administrar Orden de Compras.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.22*, un usuario puede registrar una nueva Orden de Compras. Para completar dicha acción, primeramente, el usuario activa la IU Principal Administración de Orden de Compra, donde al seleccionar del panel de opciones la opción *Registrar Orden de Compras*, solicita a la clase Gestión Registrar de Orden de Compra, que cargue los controles dentro del formulario destinado para el registro de las ordenes de compras, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Orden de Compra hace uso de la clase Compras.

Diagrama de Clase de Diseño para el caso de uso Administrar Pagos

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Pagos; donde se dispone de una clase de interfaz: IU Principal Administración de Pagos siendo las clases de gestión: Registrar Pagos, Modificar Pagos y Eliminar Pagos, y las clases de entidad relacionadas son:

Proveedor, Honorarios de Profesor y Pagos.

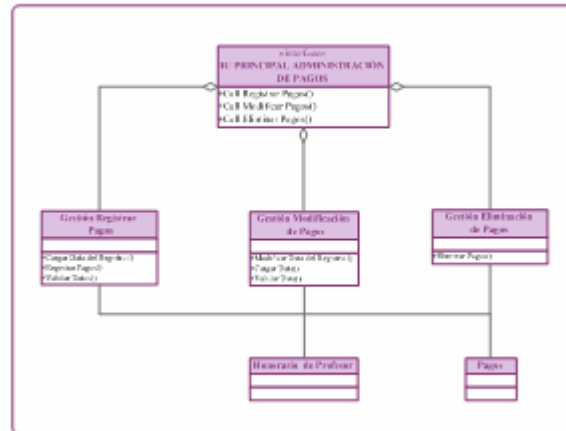


FIGURA 4.23. Diagrama de Clases de Diseño Administrar Pagos.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.23*, un usuario puede registrar un nuevo Pago. Para completar dicha acción, el usuario activa la IU Principal Administración de Pagos, donde al seleccionar del panel de opciones la opción *Registrar Pagos*, solicita a la clase Gestión Registrar Pagos, que cargue los controles dentro del formulario destinado para el registro de los pagos, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Pagos hace uso de las clases Honorario de Profesor y Pagos.

Diagrama de Clase de Diseño para el caso de uso Administrar Depósitos

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Depósitos; donde se dispone de una clase de interfaz: IU Principal Administración de Depósitos, siendo las clases de gestión:

Registrar Depósitos, Modificar Depósitos y Eliminar Depósitos, y las clases de entidad relacionadas son: Inscritos en el Postgrado y Depósitos.

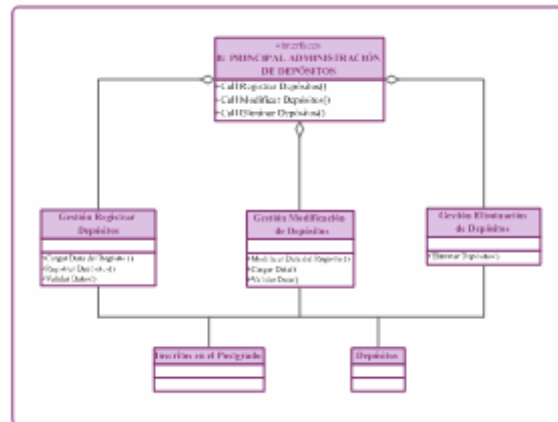


FIGURA 4.24. Diagrama de Clases de Diseño Administrar Depósitos.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.24*, un usuario puede registrar un nuevo Depósito. Para completar dicha acción, el usuario activa la IU Principal Administración de Depósitos, donde al seleccionar del panel de opciones la opción *Registrar Depósitos*, solicita a la clase Gestión Registrar Depósitos, que cargue los controles dentro del formulario destinado para el registro de los pagos, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Depósitos hace uso de las clases Inscritos en el Postgrado y Depósitos.

Diagrama de Clase de Diseño para el caso de uso Administrar Productos

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Productos; donde se dispone de una clase de

interfaz: IU Principal Administración de Productos, siendo las clases de gestión: Registrar Productos, Modificar Productos y Eliminar Productos, y la clase de entidad relacionada es Productos.

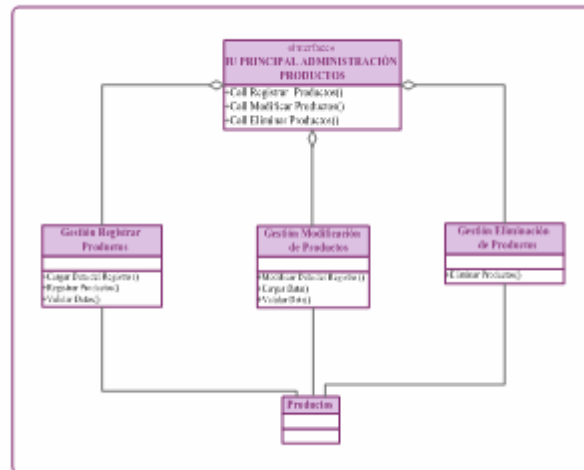


FIGURA 4.25. Diagrama de Clases de Diseño Administrar Productos.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.25*, un usuario puede registrar un nuevo Producto. Para completar dicha acción, primeramente, el usuario activa la IU Principal Administración de Productos, donde al seleccionar del panel de opciones la opción *Registrar Productos*, solicita a la clase Gestión Registrar Productos, que cargue los controles dentro del formulario destinado para el registro de los productos, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Productos hace uso de la clase Productos.

Diagrama de Clase de Diseño para el caso de uso Administrar Viáticos

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Viáticos; donde se dispone de una clase de interfaz: IU Principal Administración de Viáticos, siendo las clases de gestión: Registrar Viáticos, Modificar Viáticos y Eliminar Viáticos, y la clase de entidad relacionada es Viáticos.

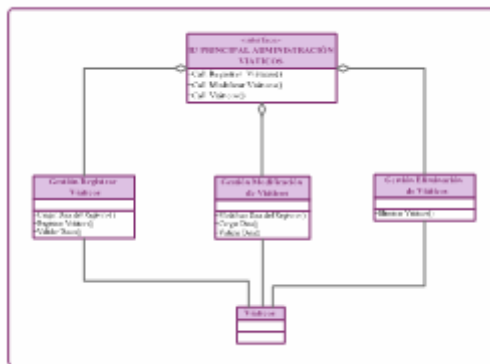


FIGURA 4.26. Diagrama de Clases de Diseño Administrar Viáticos.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.26*, un usuario puede registrar un nuevo Viático. Para completar dicha acción, primeramente, el usuario activa la IU Principal Administración de Viáticos, donde al seleccionar del panel de opciones la opción *Registrar Viáticos*, solicita a la clase Gestión Registrar Viáticos, que cargue los controles dentro del formulario destinado para el registro de los viáticos, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Viáticos hace uso de la clase Viáticos.

Diagrama de Clase de Diseño para el caso de uso Generar Reportes

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Generar Reportes; donde se dispone de una clase de

interfaz: IU Generación de Reportes, siendo la clase de gestión Generar Reportes, y la clase de entidad relacionada es Reporte.



FIGURA 4.27. Diagrama de Clases de Diseño Generar Reportes.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.27*, un usuario puede generar un nuevo Reporte. Para completar dicha acción, primeramente, el usuario activa la IU Generación de Reporte, donde al seleccionar del panel de opciones la opción *Generar Reportes*, solicita a la clase Gestión Generar Reportes, que cargue los controles dentro del formulario destinado para generar los reportes, que valide los datos ingresados por el usuario y por último que cargue la data. Para este caso, la clase Gestión Generar Reportes hace uso de la clase Reporte.

Diagrama de Clase de Diseño para el caso de uso Configurar Sistema

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Configurar Sistema; donde se dispone de una clase de interfaz: IU Configuración de Sistema, siendo la clase de gestión Configurar Sistema, y la clase de entidad relacionada es Configuración.

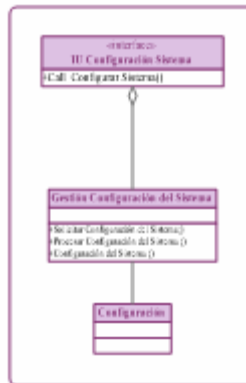


FIGURA 4.28. Diagrama de Clases de Diseño Configurar Sistema.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.28*, administrador puede configurar el sistema. Para completar dicha acción, primeramente, el usuario activa la IU Configuración del Sistema, donde al seleccionar del panel de opciones la opción *Configurar el Sistema*, solicita a la clase Gestión Configuración del Sistema, que cargue los controles dentro del formulario destinado para dicha acción, que valide los datos ingresados por el administrador y por último que cargue la configuración. Para este caso, la clase Gestión Configuración del Sistema hace uso de la clase Configuración.

Diagrama de Clase de Diseño para el caso de uso Administrar Usuarios

Este diagrama representa las interacciones existentes entre las clases de diseño que actúan en el caso de uso Administrar Usuarios; donde se dispone de una clase de interfaz: IU Principal Administración de Usuarios, siendo las clases de gestión: Registrar Usuarios, Modificar Usuarios, Eliminar Usuarios y Cambiar Contraseña; y las clases de entidad relacionadas son: Usuarios y Contraseña.

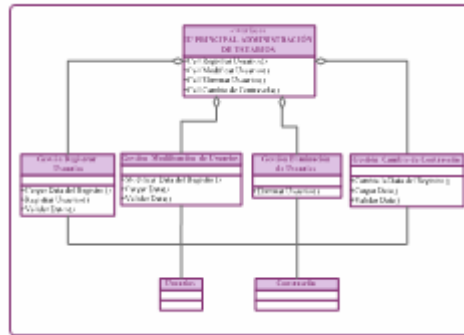


FIGURA 4.29. Diagrama de Clases de Diseño Administrar Usuarios.

[Fuente: Propia]

Tomando en consideración el diagrama de clase de diseño que se observa en la *figura 4.29.*, supongamos que un usuario desea registrar un nuevo usuario determinado. Para completar dicha acción, primeramente, el usuario activa la IU Administración de Usuarios, donde al seleccionar del panel de opciones la opción *Registrar Usuarios*, se activa a la clase Gestión Registrar Usuarios, que valide los datos ingresados por el usuario y por último que cargue el nuevo registro. Para este caso, la clase Gestión Registrar Usuarios hace uso de las clases Usuarios y Contraseña.

4.7.3.3 Diagrama de Secuencia.

Un diagrama de secuencia muestra las interacciones entre un conjunto de objetos, ordenados según el tiempo en que tienen lugar. En los diagramas de este tipo intervienen objetos, que tienen un significado parecido al de los objetos representados en el diagrama de colaboración, es decir son instancias concretas de una clase que participa en la interacción. El objeto puede existir durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción.

Estos diagramas constituyen una forma de indicar el periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento. En este tipo de diagramas también intervienen los mensajes, que son la forma como se comunican los objetos: el origen solicita (llama a) una operación del objeto destino. Existen distintos tipo de mensajes según ocurran en el tiempo: simples, síncronos y asíncronos.

Como parte de la representación gráfica para los diagramas de secuencia, los objetos se colocan de izquierda a derecha en la parte superior. Cada línea de vida de un objeto es una línea discontinua que se desplaza hacia abajo del objeto. Una línea continua con una punta de flecha conecta a una línea de vida con otra, y representa un mensaje de un objeto a otro. El tiempo se inicia en la parte superior y continúa hacia abajo. Aunque un actor es el que normalmente inicia la secuencia, su símbolo no es parte del conjunto de símbolos del diagrama de secuencias. A continuación se muestran los diagramas de secuencia correspondientes a los casos de uso fundamentales para el Sistema Administrativo del Área de Postgrado.

Diagrama de Secuencia para el Caso de Uso Administrar Orden de Compra

En el diagrama de secuencia del caso de uso Administrar Orden de Compra, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.30.*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Orden de Compra por medio del mensaje Activar Administrar Orden de Compra; el cual envía un mensaje al objeto Compras, que es la entidad que se encuentra almacenada en la Base de Datos *SAAP* para procesar el listado de todas las Ordenes de Compras; seguidamente se busca el listado; posteriormente se le

muestra dicho listado al operador .

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Orden de Compra por medio del mensaje Solicitud de Registro de Orden de Compra.

3. El objeto :IU Principal Administración de Orden de Compra activa al objeto :Gestión Registrar Orden de Compra a través del mensaje Activar Gestor Registrar Orden de Compra el cual hace un llamado para cargar el control de la data (formulario) donde posteriormente se le envía un mensaje al objeto Compras para que procese el llamado, luego busque la data correspondiente en la Base Datos SAAP y finalmente es mostrada al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para el Registro de la nueva Orden de Compras activando nuevamente al objeto Gestión Registrar Orden de Compra donde se envía un mensaje al objeto Compras para procesar los datos, para posteriormente validarlos.

5. El objeto Gestión Registrar Orden de Compra seguidamente envía un mensaje al objeto Compras, para que sea cargado el nuevo registro de la Orden de Compra, es decir, almacenando dicho registro en la Base de Datos SAAP y luego termina el proceso del registro.

6. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Orden de Compra por medio del mensaje Solicitud de Modificación de Orden de Compra.

7. El objeto :IU Principal Administración de Orden de Compra a través del mensaje Activar Gestor Modificar Orden de Compra le hace el llamado al objeto

Gestión Modificación de Orden de Compra el cual va buscar el registro seleccionado a modificar, activando el Objeto Compras, posteriormente muestra los controles con la data del registro de la Orden de Compra seleccionada (formulario) al operador y/o funcionario.

8. El actor Operador y/o Funcionario suministra los datos para la modificación de la data de la Orden de Compras activando nuevamente al objeto Gestión Modificación Orden de Compra y luego va a procesar la modificación y validarla.

9. El objeto gestión Modificación Orden de Compra nuevamente envía mensajes al objeto Compras para que cargue la modificación de la Orden de Compra, es decir, guardando dicho registro modificado en la Base de Datos; luego termina el proceso de modificación.

10. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Orden de Compra por medio del mensaje Solicitud de Eliminación de Orden de Compras.

11. El objeto :IU Principal Administración de Orden de Compra activa al objeto Gestión Eliminación de Orden de Compra a través del mensaje Activar Gestor Eliminar Orden de Compra; el cual va a buscar el registro seleccionado a eliminar, activando el Objeto Compras; posteriormente mostrando los controles con la data del registro de la Orden de Compra seleccionada (formulario) al operador y/o funcionario.

12. El Operador y/o Funcionario suministra Eliminar la Orden de Compras activando nuevamente al objeto Gestión Eliminación de Orden de Compra luego se procesa y verifica la eliminación.

13. El objeto Gestión Eliminación Orden de Compra envía nuevamente el mensaje al objeto Compras para que elimine la orden de compras, de la Base de Datos SAAP; seguidamente se envía el mensaje al operador de que se termino el proceso de Eliminación de Orden de Compra.

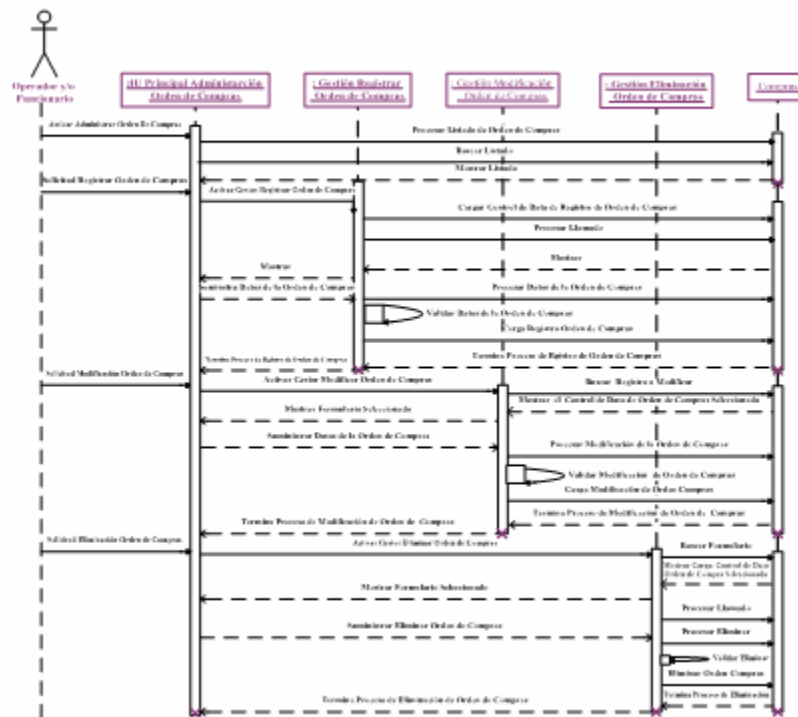


FIGURA 4.30. Diagrama de Secuencia del caso de uso Administrar Orden de Compra.

[Fuente: Propia]

Diagrama de Secuencia para el Caso de Uso Administrar Pagos

En el diagrama de secuencia del caso de uso Administrar Pagos, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver figura 4.31.).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Pagos por medio del mensaje Activar Administrar Pagos; el cual envía un mensaje al objeto Pagos y Honorario de Profesor que son las entidades que se encuentran almacenadas en la Base de datos *SAAP* y la Base de Datos Externa, para procesar el listado de todos los Pagos involucrados, seguidamente se busca el listado; posteriormente se le muestra dicho listado al operador.

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Pagos por medio del mensaje Solicitud de Registros de Pagos.

3. El objeto :IU Principal Administración de Pagos activa al objeto :Gestión Registrar Pagos a través del mensaje activar Gestor Registrar Pagos; el cual hace un llamado para cargar el control de la data (formulario); donde posteriormente se le envía un mensaje a Pagos y Honorario de Profesor para que procese el llamado en ambos objetos, luego busque la data correspondiente en las dos Bases de Datos involucradas para que seguidamente sea mostrada la información al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para el Registro de la Pagos activando nuevamente al objeto Gestión Registrar Pagos donde luego se envía un mensaje al objeto Pagos y Honorarios de Profesor para procesar los datos y posteriormente validarlos.

5.El objeto Gestión Pagos seguidamente envía un mensaje al objeto Pagos y Honorarios de Profesor para que sea cargado el nuevo registro de Pagos, es decir, guardando dicho registro en la Base de Datos *SAAP* y la Base de Datos Externa; luego termina el proceso del registro.

6.El actor Operador y/o Funcionario activa el objeto :IU Principal

Administración de Pagos por medio del mensaje Solicitud de Modificación de Pagos.

7.El objeto :IU Principal Administración de Pagos a través del mensaje Activar Gestor Modificar Pagos le hace el llamado al objeto Gestión Modificación de Pagos el cual va buscar el registro seleccionado a modificar, activando los objetos Pagos y Honorarios de Profesor, posteriormente muestra los controles con la data del registro seleccionada (formulario) al operador y/o funcionario.

8.El actor Operador y/o Funcionario suministra los datos para la modificación de la data de los pagos activando nuevamente al objeto Gestión Modificación de Pagos; luego se va a procesar la modificación y validarla.

9.El objeto Gestión Modificación Pagos nuevamente envía mensajes al objeto Pagos y Honorario de Profesor para que cargue la modificación de pagos, es decir, guardando dicho registro modificado en la Base de Datos SAAP y Base de Datos Externa; luego termina el proceso de modificación.

10.El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Pagos por medio del mensaje Solicitud de Eliminación de Pagos.

11.El objeto :IU Principal Administración de Pagos activa al objeto Gestión Eliminación de Pagos a través del mensaje Activar Gestor Eliminar Pagos el cual va a buscar el registro seleccionado a eliminar activando los objetos Pagos y Horario de Profesor; posteriormente muestra los controles con la data del registro seleccionada (formulario) al operador y/o funcionario.

12.El Operador y/o Funcionario suministra Eliminar los Pagos activando nuevamente al objeto Gestión Eliminación de Pagos luego se procesa y verifica la

eliminación.

13. El objeto Gestión Eliminación Pagos envía nuevamente el mensaje al objeto Pagos y Honorario de Profesor para que elimine los pagos de las Base de Datos SAAP y la Base de Datos Externa, seguidamente se envía el mensaje al operador de que se termino el proceso de Eliminación de Pagos.

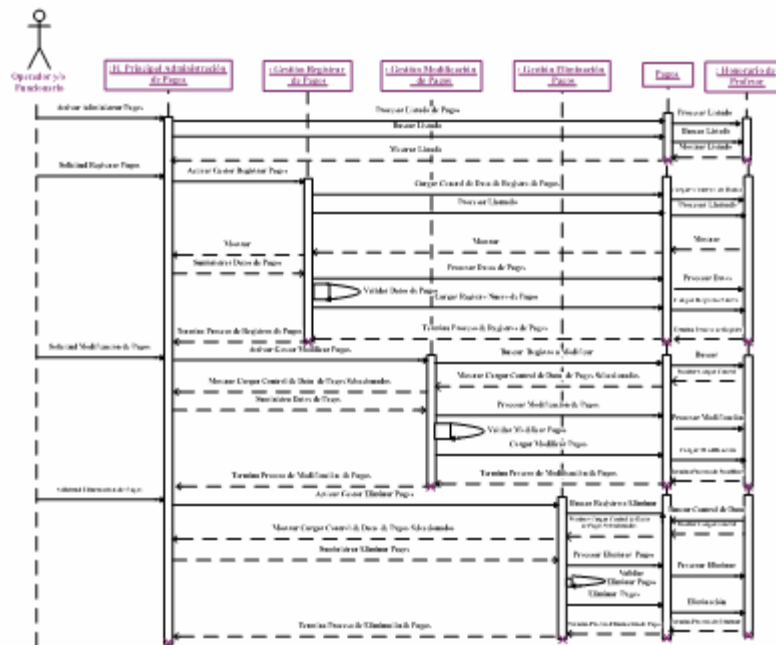


FIGURA 4.31. Diagrama de Secuencia del caso de uso Administrar Pagos.

[Fuente: Propia]

Diagrama de Secuencia para el caso de uso Administrar Depósitos

En el diagrama de secuencia del caso de uso Administrar Depósitos, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.32.*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Depósitos por medio del mensaje Activar Administrar Depósitos el cual envía un mensaje al objeto Depósitos e Inscritos en Postgrado que son las entidades que se encuentran almacenadas en la Base de datos *SAAP* y Base de Datos Externa, para procesar todo el listado de los Depósitos involucrados, seguidamente se busca el listado y luego se muestra este listado al operador y /o funcionario.

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Depósitos por medio del mensaje Solicitud de Registros de Depósitos.

3. El objeto :IU Principal Administración de Depósitos activa al objeto Gestión Registrar Depósitos a través del mensaje Activar Gestor Registrar Depósitos el cual hace un llamado para cargar el control de la data (formulario) donde posteriormente se le envía un mensaje a Depósitos e Inscritos en Postgrado para que procese el llamado en ambos objetos, luego busque la data correspondiente en la Base Datos *SAAP* y Base de datos Externa; finalmente es mostrada la información al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para el Registro de los Depósitos, activando nuevamente al objeto Gestión Registrar Depósitos donde luego se envía un mensaje al objeto Depósitos e Inscritos en Postgrado para procesar los datos, y posteriormente validarlos.

5. El objeto Gestión Depósitos seguidamente envía un mensaje al objeto Depósitos e Inscritos en Postgrado para que sea cargado el nuevo Registro de Depósitos, es decir, guardando dicho registro en la Base de Datos *SAAP* y la Base de Datos Externa; luego termina el proceso del registro.

6. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Depósitos por medio del mensaje Solicitud de Modificación de Depósitos.

7. El objeto :IU Principal Administración de Depósitos a través del mensaje Activar Gestor Modificar Depósitos le hace el llamado al objeto Gestión Modificación de Depósitos Pagos el cual va a buscar el registro seleccionado a modificar, activando los objetos Depósitos e Inscritos en Postgrado, posteriormente muestra los controles con la data del registro seleccionada (formulario) al operador y/o funcionario.

8. El actor Operador y/o Funcionario suministra los datos para la modificación de la data de los depósitos involucrados activando nuevamente al objeto Gestión Modificación de Depósitos; luego se va a procesar la modificación y validarla.

9. El objeto gestión Modificación Depósitos nuevamente envía mensajes al objeto Depósitos e Inscritos en Postgrado para que cargue la modificación, es decir, guardando dicho registro modificado en la Base de Datos SAAP y Base de Datos Externa; luego termina el proceso de modificación.

10. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Depósitos por medio del mensaje Solicitud de Eliminación de Depósitos.

11. El objeto :IU Principal Administración de Depósitos activa al objeto Gestión Eliminación de Depósitos a través del mensaje Activar Gestor Eliminar Depósitos el cual va a buscar el registro seleccionado a eliminar activando los objetos Depósitos e Inscritos en Postgrado, posteriormente muestra los controles con la data del registro seleccionado (formulario) al operador y/o funcionario.

Diagrama de Secuencia para el caso de uso Administrar Productos

En el diagrama de secuencia del caso de uso Administrar Productos, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.33.*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Productos por medio del mensaje Activar Administrar Productos el cual envía un mensaje al objeto Productos que es la entidad que se encuentra almacenada en la Base de datos *SAAP* para procesar el listado de todos los productos involucrados, seguidamente se busca el listado y se le muestra al operador.

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Productos por medio del mensaje Solicitud de Registro de Productos.

3. El objeto :IU Principal Administración de Productos activa al objeto Gestión Registrar Productos a través del mensaje Activar Gestor Registrar Productos el cual hace un llamado para cargar el control de la data (formulario), donde posteriormente se le envía un mensaje al objeto Productos para que procese el llamado, luego busque la data correspondiente en la Base Datos *SAAP*; finalmente es mostrada la información al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para un nuevo Registro de Productos activando nuevamente al objeto Gestión Registrar Productos donde luego se envía un mensaje al objeto Productos para procesar los datos y posteriormente validarlos.

5. El objeto Gestión Registrar Productos seguidamente envía un mensaje al objeto Productos para que sea cargado el nuevo registro de productos, es decir, almacenando dicho registro en la Base de Datos SAAP luego termina el proceso del registro.

6. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Productos por medio del mensaje Solicitud de Modificación de Productos.

7. El objeto :IU Principal Administración de Productos a través del mensaje Activar Gestor Modificar Productos le hace el llamado al objeto Gestión Modificación de Productos el cual va a buscar el registro seleccionado a modificar, activando el Objeto Productos, posteriormente muestra los controles con la data del registro de productos seleccionado (formulario) al operador y/o funcionario.

8. El actor Operador y/o Funcionario suministra los datos para la modificación de la data de productos activando nuevamente al objeto Gestión Modificación Productos y luego va a procesar la modificación y validarla.

9. El objeto gestión Modificación Productos nuevamente envía un mensaje al objeto Productos para que cargue la modificación del registro de productos, es decir, almacenando dicho registro modificado en la Base de Datos SAAP y luego termina el proceso de modificación.

10. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Productos por medio del mensaje Solicitud de Eliminación de Productos.

11. El objeto :IU Principal Administración de Productos activa al objeto Gestión Eliminación de Productos a través del mensaje activar Gestor Eliminar

Productos la cual va a buscar el registro seleccionado a eliminar activando el Objeto Productos, posteriormente muestra los controles con la data del registro de los productos seleccionados (formulario) al operador y/o funcionario.

12. El Operador y/o Funcionario suministra Eliminar Productos activando nuevamente al objeto Gestión Eliminación de Productos luego se procesa y verifica la eliminación.

13 El objeto Gestión Eliminación Productos envía nuevamente el mensaje al objeto Productos para que elimine el Registro de Productos, de la Base de Datos SAAP y seguidamente se envía el mensaje al operador de que termino el proceso de Eliminación de Productos.

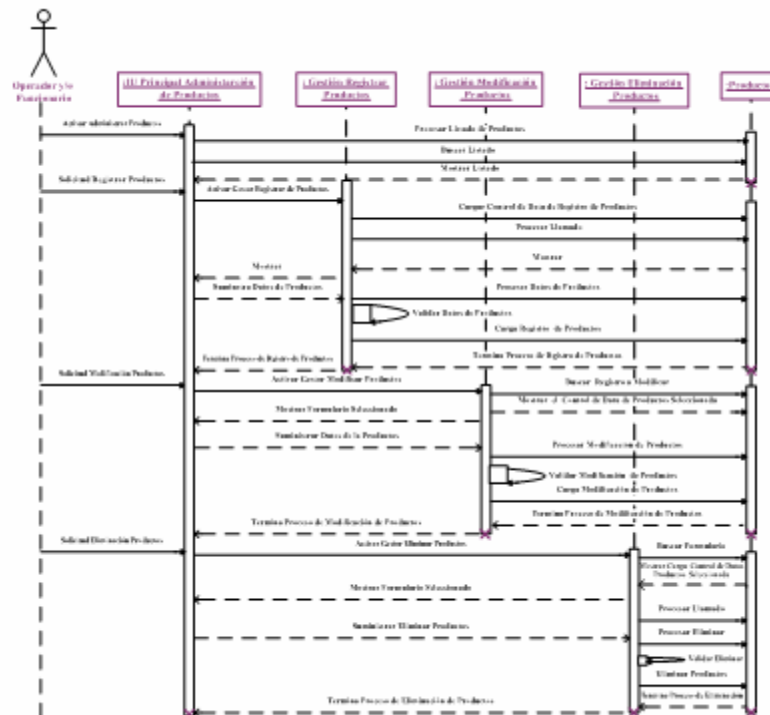


FIGURA 4.33. Diagrama de Secuencia del caso de uso Administrar Productos. [Fuente: Propia]

Diagrama de Secuencia para el caso de uso Administrar Viáticos

En el diagrama de secuencia del caso de uso Administrar Viáticos, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.34.*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Viáticos por medio del mensaje Activar Administrar Viáticos el cual envía un mensaje al objeto Viáticos que es la identidad que se encuentra almacenada en la Base de datos SAAP para procesar el listado de todos los viáticos involucrados, seguidamente se busca el listado y se le muestra al operador.

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Viáticos por medio del mensaje Solicitud de Registro de Viáticos.

3. El objeto :IU Principal Administración de Viáticos activa al objeto Gestión Registrar Viáticos a través del mensaje Activar Gestor Registrar Viáticos el cual hace un llamado para cargar el control de la data (formulario) donde posteriormente se le envía un mensaje al objeto Viáticos para que procese el llamado, luego busque la data correspondiente en la Base Datos SAAP finalmente es mostrada la información al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para un nuevo Registro de Viáticos activando nuevamente al objeto Gestión Registrar Viáticos donde luego se envía un mensaje al objeto Viáticos para procesar los datos y posteriormente validarlos.

5. El objeto Gestión Registrar Viáticos seguidamente envía un mensaje al

objeto Viáticos para que sea cargado el nuevo registro de viáticos, es decir, almacenando dicho registro en la Base de Datos *SAAP*; luego termina el proceso del registro.

6. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Viáticos por medio del mensaje Solicitud de Modificación de Viáticos.

7. El objeto :IU Principal Administración de Viáticos a través del mensaje Activar Gestor Modificar Viáticos le hace el llamado al objeto Gestión Modificación de Viáticos el cual va buscar el registro seleccionado a modificar, activando el Objeto Viáticos; posteriormente muestra los controles con la data del Registro de Viáticos seleccionada (formulario) al operador y/o funcionario.

8. El actor Operador y/o Funcionario suministra los datos para la modificación de la data de productos activando nuevamente al objeto Gestión Modificación Viáticos y luego va a procesar la modificación y validarla.

9. El objeto Gestión Modificación Viáticos nuevamente envía mensajes al objeto Viáticos para que cargue la modificación de viáticos, es decir, almacenando dicho registro modificado en la Base de Datos *SAAP*; luego termina el proceso de modificación.

10. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Viáticos por medio del mensaje Solicitud de Eliminación de Viáticos.

11. El objeto :IU Principal Administración de Viáticos activa al objeto Gestión

Eliminación de Viáticos a través del mensaje Activar Gestor Eliminar Viáticos la cual va buscar el registro seleccionado a eliminar activando el Objeto Viáticos, posteriormente muestra los controles con la data del registro de los viáticos seleccionados (formulario) al operador y/o funcionario.

12. El Operador y/o Funcionario suministra Eliminar Viáticos activando nuevamente al objeto Gestión Eliminación de Viáticos luego se procesa y verifica la eliminación.

13.El objeto Gestión Eliminación Viáticos envía nuevamente el mensaje al objeto Viáticos para que elimine el Registro de Viáticos, de la Base de Datos SAAP; seguidamente se envía el mensaje al operador de que termino el proceso de Eliminación de Viáticos.

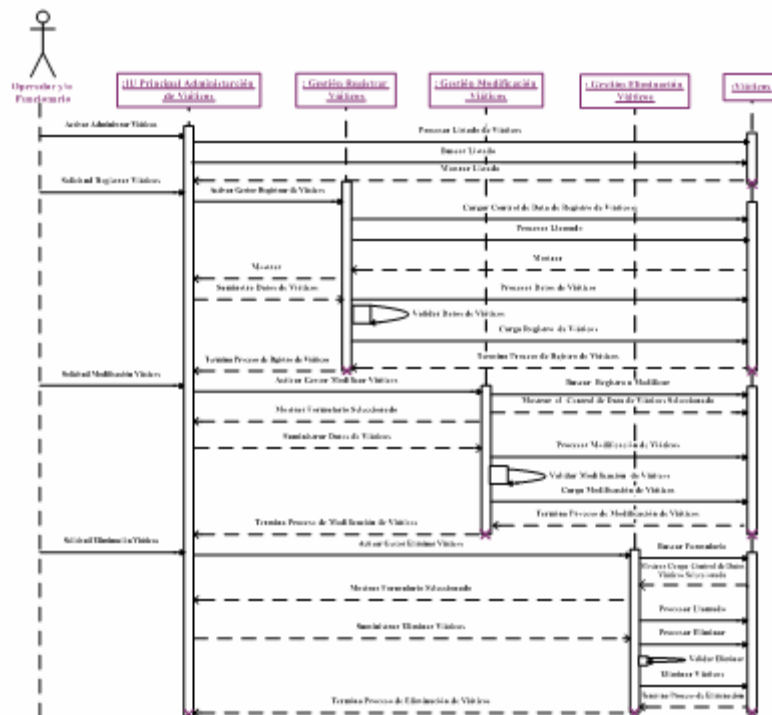


FIGURA 4.34. Diagrama de Secuencia del caso de uso Administrar Viáticos.

Diagrama de Secuencia para el caso de uso Generar Reportes

En el diagrama de secuencia del caso de uso Generar Reportes, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.35*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Generación de Reportes por medio del mensaje Activar Solicitar Generar Reportes.

2. El objeto :IU Generación de Reportes activa al objeto :Gestión Generar Reportes a través del mensaje Activar Gestor Generar Reportes seguidamente se activa el objeto Reportes para buscar el tipo de reporte seleccionado; posteriormente muestra el control de la data (formulario) al actor operador y/o funcionario.

3. El actor Operador y/o Funcionario suministra los datos para Generar el Reporte, activando nuevamente al objeto Gestión Generar Reportes donde luego se envía un mensaje al objeto Reporte para procesar los datos, y posteriormente validarlos.

4. El objeto Gestión Generar Reportes seguidamente envía un mensaje al objeto Reportes para que se Genere el Reporte deseado, tomando la información desde la Base de Datos SAAP, luego termina el proceso Generar Reportes.

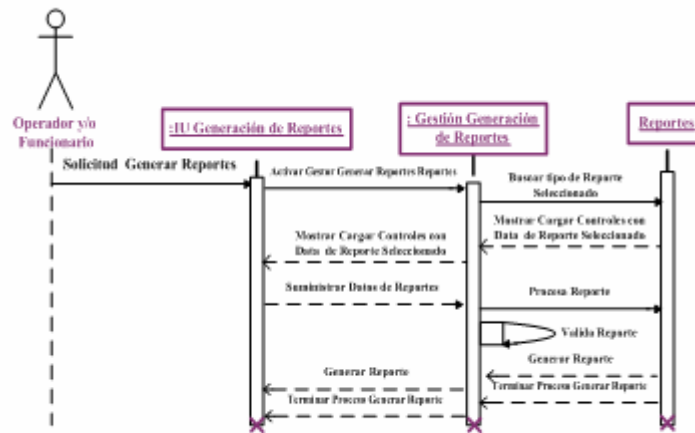


FIGURA 4.35. Diagrama de Secuencia del caso de uso Generar Reportes.

[Fuente: Propia]

Diagrama de Secuencia para el caso de uso Configurar Sistema.

En el diagrama de secuencia del caso de uso Configurar Sistema, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.36.*).

1. El actor Administrador realiza un llamado al objeto :IU Configuración de Sistema por medio del mensaje Activar Solicitar Configurar Sistema.

2. El objeto objeto :IU Configuración de Sistema activa al objeto Gestión Configuración de Sistema a través del mensaje Activar Gestor Generar Configurar Sistema, seguidamente se activa al objeto Configuración para buscar la información acerca de la configuración que desee el administrador; y posteriormente mostrar el control de la data (formulario) al actor.

3. El actor suministra los datos para la Configuración del Sistema activando nuevamente al objeto Gestión Configuración de Sistema donde luego se envía un mensaje al objeto Configuración para procesar la configuración y posteriormente validar.

4. El objeto Gestión Configuración de Sistema seguidamente envía un mensaje al objeto Configuración para que realice la configuración deseada, tomando la información desde la Base de Datos SAAP; luego termina el proceso de configurar sistema.

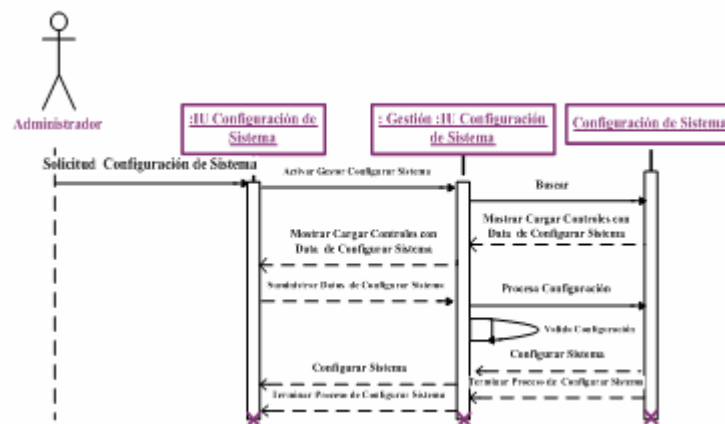


FIGURA 4.36. Diagrama de Secuencia del caso de uso Configurar Sistema

[Fuente: Propia]

Diagrama de Secuencia para el caso de uso Administrar Usuarios.

En el diagrama de secuencia del caso de uso Administrar Usuarios, el proceso de comunicación de los objetos que interactúan, se establece a través de los mensajes que se envían entre ellos. Dicho proceso se describe de la siguiente manera: (Ver *figura 4.37.*).

1. El actor Operador y/o Funcionario realiza un llamado al objeto :IU Principal Administración de Usuarios por medio del mensaje activar Administrar Usuarios en el cual envía un mensaje a los objeto Usuarios y Contraseña que son las entidades que se encuentra almacenada en la Base de datos *SAAP* para procesar el listado de todos los Registros de Usuarios y Contraseñas, seguidamente se busca el listado; posteriormente se le muestra dicho listado al operador.

2. El Actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Usuarios por medio del mensaje Solicitud de Registro de Usuarios.

3. El objeto :IU Principal Administración de Usuarios activa al objeto Gestión Registrar Usuarios a través del mensaje Activar Gestor Registrar Usuarios el cual hace un llamado para cargar el control de la data (formulario) donde posteriormente se le envía un mensaje a los objeto Usuarios y Contraseña para que procese el llamado, busque la data correspondiente en la Base Datos *SAAP* y luego es mostrada al actor operador y/o funcionario.

4. El actor Operador y/o Funcionario suministra los datos para un nuevo Registro de Usuarios activando nuevamente al objeto Gestión Registrar Usuarios donde luego se envía un mensaje a los objeto Usuarios y Contraseña para procesar los datos y posteriormente validarlos.

5. El objeto Gestión Registrar Usuarios seguidamente envía un mensaje al objeto Usuarios y Contraseña para que sea cargado el nuevo Registro de Usuarios y Contraseña, es decir, almacenando dicho registro en la Base de Datos *SAAP*; luego termina el proceso del registro.

6. El actor Operador y/o Funcionario activa el objeto :IU Principal

Administración de Usuarios por medio del mensaje Solicitud de Modificación de Usuarios.

7. El objeto :IU Principal Administración de Usuarios a través del mensaje activar Gestor Modificar Usuarios le hace el llamado al objeto Gestión Modificación de Usuarios la cual va buscar el registro seleccionado a modificar, activando los Objetos Usuarios y Contraseña; posteriormente muestra los controles con la data del registro de usuarios seleccionada (formulario) al operador y/o funcionario.

8. El actor Operador y/o Funcionario suministra los datos para la modificación de la data de productos activando nuevamente al objeto Gestión Modificación Usuarios y luego va a procesar la modificación y validarla.

9. El objeto Gestión Modificación Usuarios nuevamente envía mensajes a los objetos Usuarios y Contraseña para que cargue la modificación, es decir, guardando dicho registros modificados en la Base de Datos SAAP y luego termina el proceso de modificación.

10. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Usuarios por medio del mensaje Solicitud de Eliminación de Usuarios.

11. El objeto :IU Principal Administración de Usuarios activa al objeto Gestión Eliminación de Usuarios a través del mensaje Activar Gestor Eliminar Usuarios el cual va buscar el registro seleccionado a eliminar activando a los objetos Usuarios y Contraseña, posteriormente muestra los controles con la data del registro seleccionada (formulario) al operador y/o funcionario.

12. El Operador y/o Funcionario suministra Eliminar Usuarios activando

nuevamente a los objetos Gestión Eliminación de Usuarios, luego se procesa y verifica la eliminación.

13. El objeto Gestión Eliminación Usuarios envía nuevamente el mensaje a los objetos Usuarios y Contraseña para que elimine el registro de la Base de Datos *SAAP*; seguidamente se envía el mensaje al operador de que termino el proceso de Eliminación.

14. El actor Operador y/o Funcionario activa el objeto :IU Principal Administración de Usuarios por medio del mensaje Solicitud de Cambio de Contraseña.

15. El objeto :IU Principal Administración de Usuarios a través del mensaje activar Gestor Cambiar Contraseña hace el llamado al objeto Gestión Cambio de Contraseña el cual va a buscar el registro seleccionado a modificar, activando los Objetos Usuarios y Contraseña posteriormente muestra los controles con la data del registro seleccionada (formulario) al operador y/o funcionario.

16. El actor Operador y/o Funcionario suministra los datos para el cambio de contraseña activando nuevamente a los objetos Gestión Cambio de Contraseña; luego procesa el cambio y lo valida.

17. El objeto gestión Cambio de Contraseña nuevamente envía mensajes a los objetos Usuarios y Contraseña para que cargue el cambio, es decir, almacenando dicho registros que fueron sometidos a un cambio en la Base de Datos *SAAP*; luego termina el proceso de cambio de contraseña.

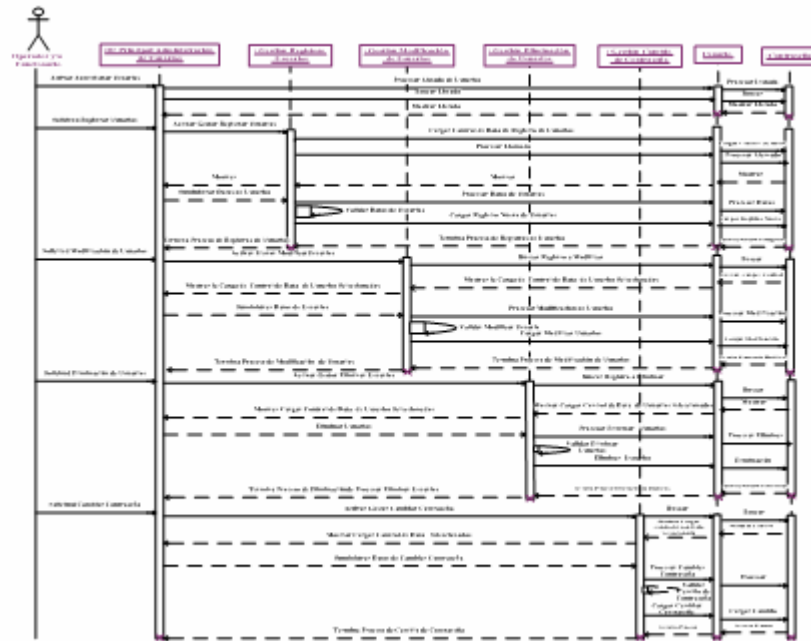


FIGURA 4.37. Diagrama de Secuencia del caso de uso Administrar Usuarios

[Fuente: Propia]

4.7.3.4 Modelo de Despliegue.

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputos. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en el diseño. Lo cual es importante destacar lo siguiente:

1. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar.

2. Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como Internet, Intranet, bus y similares.

3. El modelo de despliegue puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación.

4. La funcionalidad (los procesos) de un nodo se definen por los componentes que se distribuyen en ese nodo.

5. El modelo de despliegue en si representa una correspondencia entre la arquitectura software y la arquitectura del sistema.

A continuación se presenta el Modelo de Despliegue del Sistema (SAAP).

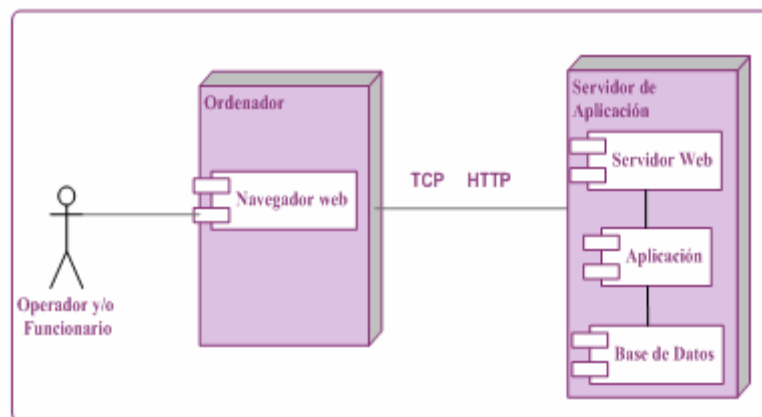


FIGURA 4.38. Modelo de Despliegue del Sistema Administrativo del Área de Postgrado (SAAP).

[Fuente: Propia]

En la *figura 4.38* se muestra que el sistema SAAP se ejecutará sobre dos nodos un servidor y un ordenador. El nodo ordenador la cual es un computador de sobremesa con cualquier sistema operativo capaz de ejecutar un navegador web *mozilla* o compatible, y conexión a una red IP con acceso al nodo servidor de aplicación, además contiene el componente de navegador web que es el software de visualización de páginas web a través de una red Ip que soportará las tecnologías HTML, XHTML, CSS y Javascript, así como el protocolo HTTP, esencial para el funcionamiento de la aplicación; el nodo ordenador es manejada por el operador y/o Funcionario. El nodo Servidor de Aplicación es un equipo informático donde se almacena y ejecuta la aplicación, contiene el componente de servidor de aplicación que depende la componente de aplicación y es el software de pasarela entre la aplicación y el Navegador WEB, debe soportar el protocolo HTTP y TPC para cumplir los requisitos de tiempo de las aplicaciones interactivas. Además contiene la componente base de datos la cual es el Software de base de datos utilizado para almacenamiento persistente de los datos de la aplicación.

4.7.4 Diseño de la Base de Datos.

El diseño de la base de datos es fundamental en el diseño del sistema, pues permite almacenar toda la información que será solicitada y a su vez manipulada por los distintos tipos de usuarios.

Para realizar el diseño de la base de datos del Sistema Administrativo del Área de Postgrado, se tomaron las clases de entidad estudiadas en el análisis; la abstracción de estos objetos será representada a través de las tablas que conformarán el Modelo Relacional de Datos.

4.7.4.1 Modelo Relacional.

Un modelo relacional permite representar conceptualmente las relaciones de todos los datos a través de tablas; siendo el punto de partida para la implementación de una Base de Datos, facilitando la abstracción y comprensión de todos los elementos que forman parte de ella.

A continuación se presenta el esquema relacional del sistema de almacenamiento de datos para la aplicación SAAP.

El modelo relacional que se puede observar en la figura 4.39 representa las relaciones lógicas de cada entidad identificada. Para cada una de las entidades existe una propiedad que las identifica únicamente y que se corresponde con la clave primaria de cada una de las tablas en la implementación relacional. Las entidades tienen otras propiedades que las describen y que están asociadas con los distintos campos de la tabla (relación).

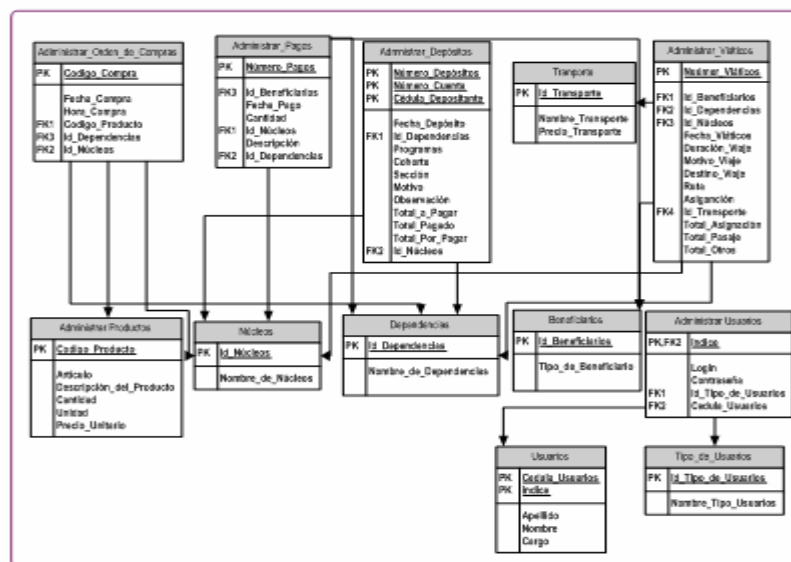


FIGURA 4.39. Modelo Relacional de la Base de Datos de SAAP

Identificación de Tablas de la Base de Datos.

Durante el proceso de definición de la estructura de las tablas, se identificaron en definitiva 12 tablas que formarán parte del sistema de datos del software, donde unas están destinadas al almacenamiento de datos relacionados con la gestión de actividades desempeñadas por la Coordinación de Postgrado, y el resto están destinadas al almacenamiento de datos relacionados con usuarios y configuración del sistema. A continuación se describen las tablas que componen la base de datos del sistema y se tabulan los campos que conforman cada una de las tablas.

Tabla Administrar_Orden_de_Compra

Dentro de esta tabla se almacenan las distintas compras y a su vez la respectiva descripción de los productos necesarios para cubrir las necesidades de los diferentes coordinaciones de postgrado. Los campos claves correspondientes a la tabla son *Codigo_Compra* y *Codigo_Producto*.

TABLA 4.17. Campos de la Tabla Administrar_Orden_de_Compra.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
Codigo_Compra	number(10,0)	Id de las orden de compras
Id_Codigo_Producto	number(10,0)	Id de los productos
Fecha_Compra	date	Fecha de la compra de algún producto
Id_Dependencia	number(20)	Id de la Dependencia
Id_Nucleo	number(20)	Id del Núcleo
Cantidad	number(20)	Cantidad de productos que se van a comprar

[Fuente: Propia]

Tabla Administrar_Pagos

Dentro de esta tabla se almacenan todos los pagos realizados por la coordinación de postgrado; tanto a los profesores y diversos pagos que se realizan en la compra de algún producto que sea necesario. El campo clave correspondiente a la tabla es el *Numero_Pago*.

TABLA 4.18. Campos de la Tabla Administrar_Pagos.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
Numero_Pago	number(10,0)	Id de los pagos
Descripcion	varchar2(100)	Descripción del concepto del pago que se realiza
Fecha_Pago	date	Fecha de los pagos que se realicen
Cantidad	number(10,0)	Monto del pago que se esta realizando .
Id_Dependencia	number(20)	Id de la Dependencia
Id_Nucleo	number(20)	Id del Núcleo
Id_Beneficiario	number(20)	Id del Beneficiario

[Fuente: Propia]

Tabla Administrar_Depositos

Dentro de esta tabla se almacenan todos los depósitos realizados por las personas interesadas en algunos de los cursos inscritos en el postgrados que se ofrecen en la Universidad de Oriente del Núcleo de Anzoátegui. Los campos claves correspondientes a la tabla son *Numero_deposito*, *Numero_cuenta* y *Cedula*.

TABLA 4.19. Campos de la Tabla Administrar _ depósitos.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
Numero_deposito	number(10,0)	Id del depósito que se realiza
Numero_cuenta	number(25,0)	Id del número de cuenta del depósito que se realiza
Cedula	number(10,0)	Cedula de quien realiza el deposito
Fecha_Deposito	date	Fecha del deposito
Id_Dependencia	number(20)	Id del la Dependencia
Id_Nucleo	number(20)	Id del Núcleo
Id_Beneficiario	number(20)	Id del Beneficiario
Programa	varchar2(100)	Nombre de los programas que se dictan
Cohorte	varchar2(100)	Son los programas de postgrados que se dictan
Seccion	varchar2(50)	Sección en la cual se inscribe
Monto_deposito	number(12,0)	Cantidad depositada
Motivo	varchar2(100)	Justificación por el cual se esta efectuando el depósito
Observacion	varchar2(100)	Aclara cierta información de un depósito
Total_apagar	number(12,0)	Cantidad a total cancelar
Total_pagado	number(12,0)	Cantidad pagada
Total_porpagar	number(12,0)	Cantidad restante por cancelar

[Fuente: Propia]

Tabla Administrar_Productos

Dentro de esta tabla se almacenan todos los productos que van suministrar a las distintas coordinaciones de postgrados de la Universidad de Oriente del Núcleo de Anzoátegui. El campo clave correspondiente a la tabla es. El campo clave correspondiente a la tabla es *CODIGO_PRODUCTO*.

TABLA 4.20. Campos de la Tabla Administrar_Productos.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
CODIGO_PRODUCTO	number(10,0)	Id del producto
Articulo	varchar2(100)	Cualquier objeto que puede ser ofrecido a un mercado que pueda satisfacer un deseo o una necesidad.
Descripción_del_Producto	varchar2(100)	Explicación, de forma del producto
Precio_Unitario	number(12,0)	Es el costo de cada unidad.

[Fuente: Propia]

Tabla Administrar _Usuarios

Dentro de esta tabla se almacenan los distintos usuarios con el objetivo de controlar el acceso al sistema. El campo clave correspondiente a la tabla es el *Id_Indice* y *Id_Usuarios*.

TABLA 4.22. Campos de la Tabla Administrar_Usuarios

CAMPOS	TIPOS DE DATOS	DESCRIPCION
indice	number(10)	Indice de la tabla.
idUsuario	number(10)	Id del usuario del sistema.
Login	Varcha2r(20)	Login de acceso del usuario.
Contraseña	Varchar2(20)	Contraseña de acceso del usuario.
IdTipo_Usuario	number(10)	Id del tipo de usuario al cual pertenece el usuario.

[Fuente: Propia]

Tabla Administrar_Viaticos

Dentro de esta tabla se almacenan todos los pagos que se proporcionan para cubrir los gastos en el cumplimiento de comisiones de los profesores de postgrados de la Universidad de Oriente del Núcleo de Anzoátegui. El campo clave correspondiente a la tabla *Numero_Viatico*.

TABLA 4.21. Campos de la Tabla Administrar_Viaticos.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
Numero_Viatico	number(10,0)	Id del viático
Id_Dependencia	number(20)	Id del Dependencia
Id_Nucleo	number(20)	Id del Núcleo
Id_Beneficiario	number(20)	Id del Beneficiario
Fecha_Viatico	date	Fecha del viaje a realizar por el profesor
Duracion_Viaje	varchar2(100)	Duración del viaje
Fecha_Destino	date	Fecha en el cual se realiza el viaje
Motivo_Viaje	varchar2(100)	Motivo por el cual realiza el viaje
Destino_Viaje	varchar2(100)	Destino del viaje que realiza
Asignacion	number(12,0)	Cantidad de dinero que se le asigna al beneficiario
Id_Tipo_Transporte	number(10)	Id Tipo de transporte
Ruta_Viaje	varchar2(100)	Es el recorrido del viaje que realiza el beneficiario
Total_asignacion	number(12,0)	Cantidad total de dinero que se le asigna al beneficiario
Total_pasaje	number(12,0)	Total a cancelar por el pasaje
Total_otros	number(12,0)	Total a cancelar por otros gastos

[Fuente: Propia]

Tabla Usuario

Dentro de esta tabla se almacenan los usuarios que están involucrados con el sistema. El campo clave correspondiente a la tabla es el *Id_Indice* y *Id_Cedula*

TABLA 4.23. Campos de la Tabla Usuario.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
indice	number(10)	Indice de la tabla.
Cedula	number(10)	Cedula del usuario .
Apellido	Varchar2(100)	Apellido del usuario
Nombre	Varchar2(100)	Nombre del usuario
Cargo	Varchar2(100)	Cargo del usuario

[Fuente: Propia]

Tabla Tipo_Usuarios

Dentro de esta tabla se almacenan las distintas tipos de usuarios que interactúan con el sistema. El campo clave correspondiente a la tabla es el *IdTipo_Usuario*.

TABLA 4.24. Campos de la Tabla Tipo_Usuarios.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
IdTipo_Usuario	number(20)	Id del Usuario
Tipo_Usuario	varchar2(20)	Tipo de Usuario

[Fuente: Propia]

Tabla Dependencias

Dentro de esta tabla se almacenan las distintas Dependencias de Coordinaciones de Postgrados de la Universidad de Oriente. El campo clave correspondiente a la tabla es el *Id_Dependencias*.

TABLA 4.25. Campos de la Tabla Dependencias.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
<u>Id_Dependencias</u>	number(20)	Id de la Dependencias
NombreDependencias	varchar2(20)	Nombre de las Dependencias.

[Fuente: Propia]

Tabla Núcleos

Dentro de esta tabla se almacenan los distintos núcleos de la Universidad de Oriente. El campo clave correspondiente a la tabla es el idNucleo.

TABLA 4.26. Campos de la Tabla Nucleos.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
idNucleo	number(20)	Id del núcleo.
NombreNucleo	varchar2(20)	Nombre del Núcleo.

[Fuente: Propia]

Tabla Tipo_Beneficiarios

Dentro de esta tabla se almacenan los distintos beneficiarios que se le realiza un pago. El campo clave correspondiente a la tabla es el *Id_TipoBeneficiario*.

TABLA 4.27. Campos de la Tabla Tipo_Beneficiarios.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
<u>Id_TipoBeneficiario.</u>	number(20)	Id Tipo de Beneficiario.
Nombre_Beneficiario	varchar2(20)	Ente a los que se le realiza el pago

[Fuente: Propia]

Tabla Tipo_Transporte

Dentro de esta tabla se almacenan los distintos transportes que utiliza cualquier beneficiario . El campo clave correspondiente a la tabla es el Id_TipoTransporte.

TABLA 4.28. Campos de la Tabla Tipo_ Transporte.

CAMPOS	TIPOS DE DATOS	DESCRIPCION
Id_TipoTransporte..	number(20)	Id Tipo de Transporte.
Nombre_ Transporte.	varchar2(20)	Nombre del transporte
Precio_Transporte	number(12,0)	Es el costo del transporte

[Fuente: Propia]

4.8 Planificación de las Sigüientes Fases.

4.8.1 Fase de Construcción.

En la fase de construcción se realiza la codificación de los módulos generados a través de los flujos de trabajo realizado en la fase de elaboración. Se crea un software de prueba del sistema para su implementación.

4.8.2 Fase de Transición.

En la fase de transición se coloca en funcionamiento el sistema, incorporando e integrando sus componentes. Además, se realizan las pruebas y correcciones necesarias del sistema para su implementación final.

4.9 Resumen de la Fase De Elaboración.

Durante el desarrollo de esta fase se adicionaron nuevos requisitos, se identificaron nuevos casos de uso que trajo consigo la reestructuración y actualización del modelo de casos de uso del sistema.

Al llegar a este punto se tiene una visión estable de la arquitectura del sistema donde los principales elementos de riesgo han sido abordados y resueltos; considerando que el diseño de las clases y subsistemas, así como el de las base de datos, son suficientemente completos como para ser utilizados en la fase de construcción durante la codificación del software.

CAPITULO V

FASE DE CONSTRUCCIÓN

5.1 Introducción

La *Fase de Construcción* dentro del Proceso Unificado, desarrolla o adquiere los componentes del software que harán que cada caso de uso sea operativo para los usuarios finales. En ésta fase, el objetivo principal es ajustar la construcción del software en función a la arquitectura del sistema, y de ser necesario modificarla para incorporar los cambios que surgen durante el proceso de construcción. Aquí se culmina el estudio del sistema y la ejecución de las primeras pruebas, haciendo mayor énfasis, principalmente en el flujo de trabajo de implementación y en el de prueba, tal y como es observable según la *figura 5.1*.



FIGURA 5.1. Flujos de Trabajo Fundamental en la Fase de Construcción

[Fuente: Jacobson, I.]

El fundamento de esta fase, es desarrollar el flujo de trabajo de implementación,

sustentado en la codificación de cada componente, los cuales formaran parte del sistema ejecutable y de prueba. Las pruebas son el mecanismo que se utiliza para validar la calidad del software.

La versión final del software es la reflejada en el modelo de análisis y en el modelo de diseño de la fase de elaboración. Por tal razón, en la fase de construcción, se planea codificar las interfaces de usuario, estableciendo finalmente la comunicación entre ellas. Se trabajará directamente en la implementación y las pruebas con base en el análisis y diseño de la fase de elaboración.

5.2 Planificación de la Fase de Construcción.

El principal objetivo de esta fase es obtener la “versión beta” del software. Es decir, una versión lista para ser entregada a los usuarios finales de la aplicación, en este caso, a los usuarios que laboran dentro del Área Administrativa de Postgrado de la Universidad de Oriente del Núcleo de Anzoátegui.

Para lograr los objetivos planteados en esta fase, en primer lugar se especificara el lenguaje de programación usado para la codificación de las páginas Web. La implementación de estas páginas implica también la construcción visual de las mismas, deben tener un aspecto elegante y un esquema de navegabilidad práctico, siguiendo el prototipo de interfaz planteado en la fase de elaboración.

En la fase de construcción se considera la realización de una serie de actividades, en función del esquema que se muestra a continuación:

1. Selección del Lenguaje de programación.

2. Diseño de las interfaces de usuario.
3. Implementación de los Subsistemas.
4. Definición de los Diagramas de Componentes.
5. Implementación de las Clases.
6. Realización de las pruebas por Unidad.
7. Realización de las pruebas por Integración.

5.3 Justificación de la Elección de las Herramientas de Construcción del Sistema.

El Sistema Administrativo del Área de Postgrado se espera que sea implementado dentro del Área Administrativa de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente, lo cual requiere de un lenguaje de programación con licencia de software libre debido al decreto presidencial 3390.

Por otra parte, podemos mencionar que la programación orientada a objetos es un paradigma de programación que se ha popularizado debido a la gran cantidad de ventajas que posee: extensibilidad, eliminación de redundancias y reutilización del código, entre otras.

Debido a lo antes mencionado se escogió PHP como lenguaje de programación, ya que es un lenguaje que soporta objetos, utilizado para la generación de páginas web dinámicas y está basado en herramientas con licencia de software libre.

El lenguaje de programación PHP (*Personal Hypertext Processor*), es el lenguaje a través del cual serán codificadas las aplicaciones que comprenden al software. Generalmente, es usado para la creación de contenido dinámico para sitios Web y el cual puede ser embebido en paginas HTML. Está basado en herramientas con licencia de software libre, cuya tecnología es del lado del servidor; ofrece funciones como el acceso a bases de datos, creación de foros, libros de visita, gestión de base de datos, gestión de archivos, etc.

Las aplicaciones realizadas en el lenguaje PHP, requieren el uso de un servidor para realizar la ejecución de la misma, por ese motivo se utilizará el servidor EasyPHP, para ejecutar el sistema *SAAP*.

Para el diseño y creación de la base de datos se utilizará Oracle, es muy rápido, seguro, maneja grandes bases de datos, capaz de manejar cantidades enormes de registros, y compagina perfectamente con el lenguaje de programación a utilizar (PHP).

5.4 Interfaces Principales del Sistema.

Siguiendo los estándares, a continuación se muestran las interfaces de usuario del Sistema Administrativo del Área de Postgrado, indicando las áreas que la conforman.

5.4.1 Interfaz de Inicio de Sesión.

Esta interfaz permite que los distintos usuarios del sistema sean autenticados. Mediante un acceso a la base de datos, se comprueba tanto el nombre de usuario y contraseña ingresada por el usuario; si los datos son correctos se le permite el acceso

a las aplicaciones que le correspondan, de lo contrario se le deniega el acceso al sistema. A continuación se muestra la interfaz de inicio de sesión del sistema:



FIGURA 5.2. Vista de la interfaz de Inicio de Sesión del Sistema

[Fuente: Propia]

5.4.2 Interfaz de Bienvenida al Sistema.

En esta interfaz se les da la bienvenida a los usuarios internos del sistema (Operadores, Administrador); a partir de esta interfaz el usuario puede comenzar a interactuar con el sistema, ya que el menú principal se carga con los módulos del sistema a los que el usuario tiene acceso como tal.



FIGURA 5.3. Vista de la Interfaz de Bienvenida

[Fuente: Propia]

5.5 Primera Iteración.

En esta iteración se implementará el caso de uso *Administrar Orden de Compras*.

5.5.1. Implementación de Subsistemas.

Se identificaran los subsistemas de implementación necesarios para realizar la implementación del caso de uso *Administrar Orden de Compras*.

Subsistema de Implementación Administrar Orden de Compras.

El subsistema de implementación *Administrar Orden de Compras* implementa al subsistema de diseño del mismo nombre.

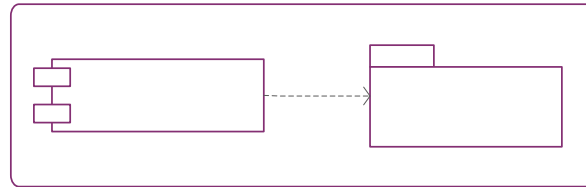


FIGURA 5.4. Subsistema de Implementación Administrar Orden de Compras

[Fuente: Propia]

Administrar Orden de Compras

5.5.2 Diagrama de Componente.

Un componente es el empaquetamiento físico de los elementos de un modelo de diseño y es conforme a un conjunto de interfaces a las que proporciona su realización.

Normalmente, un diagrama de componentes contiene componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Este diagrama muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista de este diagrama se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

A continuación se muestra el diagrama de componente del subsistema de implementación Administrar Orden de Compras:

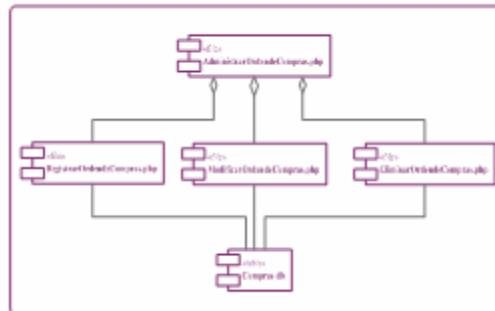


FIGURA 5.5. Diagrama de Componente para Administrar Orden de Compras

[Fuente: Propia]

En la *Figura 5.5.* Se observa que el subsistema de implementación Administrar Orden de Compras posee una relación de agregación con los componentes RegistrarOrdendeCompras.php, ModificarOrdendeCompras.php, y EliminarOrdendeCompras.php, los cuales forman parte del funcionamiento del componente Administrar Orden de Compras.

5.5.3. Implementación de las Clases.

En esta parte, se realizará la implementación en código de las clases de diseño correspondientes al caso de uso *Administrar Orden de Compras*, las cuales fueron definidas en la fase de elaboración. A continuación se detalla la clase de interfaz que posee este subsistema.

Clases de Interfaz.

La interfaz propia del caso de uso Administrar Orden de Compras es la siguiente:

:IU Principal Administración de Orden de Compras.

La clase :IU Principal Administración de Orden de Compras es implementada dentro del componente *administrarordendecompras.php*, del subsistema Administrar Orden de Compras.

Vista De La Ventana:



The screenshot displays the SAAP interface with a table of purchase orders. The table has the following columns: Código_Compra, Código_Producto, Fecha_Compra, Departamento, Estado, and Cantidad. The data rows are as follows:

Código_Compra	Código_Producto	Fecha_Compra	Departamento	Estado	Cantidad
1	6	07-07-00	adm	pend	
4	6	07-08-00	adm	pend	
10111111	4	20-01-02			1
10111111	8	20-01-02			8
1122222	4	20-01-03	adm	pend	1
1122222	8	20-01-03	adm	pend	1
201000	1	24-02-00	adm	pend	444
0000	1	22-01-00	adm	pend	1
0743	4	24-01-00	adm	pend	4
0743	20	23-01-00	adm	pend	4
0000	1	22-01-00	adm	pend	4
00	1	22-02-00	adm	pend	4
00	1	21-02-00	adm	pend	10
00	4	21-02-00	adm	pend	2
00	8	21-02-00	adm	pend	1
00	11	21-02-00	adm	pend	1
00	1111	21-02-00	adm	pend	1
00200	4	21-02-00	adm	pend	8
00200	8	21-02-00	adm	pend	8

FIGURA 5.6. Vista de ventana de la IU Principal Administración de Orden de Compras

[Fuente: Propia]

A continuación se muestra el código del componente *administrarordendecompras.php*:

```
<?php
class Compras
{
//Variables de los datos de compra
var $CodigoCompra;
var $CodigoProducto;
```

```

var $DescripcionProducto;
var $FechaCompra;
var $Cantidad;
var                                $UnidadCompra;                                ////
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
var $Dependencia; //Departamento de postgrado que solicita la compra
var $Nucleo;      //Nucleo a la cual pertenece el departamento del postgrado
//.....
.

function Compras() // Constructor de la clase
{
// Inicializando las variables
$this->CodigoCompra = 0;
$this->CodigoProducto = 0;
$this->DescripcionProducto = "x";
$this->FechaCompra = date("d-m-Y");
$this->Cantidad = 0;
$this->UnidadCompra = 0;
$this->Dependencia = "x";
$this->Nucleo = "x";
} // fin Constructor

//.....Fin.....

....

//*****
SET*****

```

```

//          *****Datos          de          las
compras*****
function SetCodigoCompra($valor)
{
    $this->CodigoCompra = $valor;
}
function SetCodigoProducto($valor)
{
    $this->CodigoProducto = $valor;
}
function SetDescripcionProducto($valor)
{
    $this->DescripcionProducto = $valor;
}
function SetFechaCompra($valor)
{
    $this->FechaCompra = $valor;
}
function SetCantidad($valor)
{
    if($valor==NULL)
    {
        $this->Cantidad = 0;
    }
    else {
        $this->Cantidad = $valor;
    }
}
function SetUnidadCompra($valor)

```

```
{
    if($valor==NULL)
    {
        $this->UnidadCompra = 0;
    }
    else {
        $this->UnidadCompra = $valor;
    }
}

function SetDependencia($valor)
{
    $this->Dependencia = $valor;
}

function SetNucleo($valor)
{
    $this->Nucleo = $valor;
}

//.....Fin.....

//*****Get *****

function GetCodigoCompra()
{
    return($this->CodigoCompra);
}

function GetCodigoProducto()
{
    return($this->CodigoProducto);
}
```

```
function GetDescripcionProducto()
{
    return($this->DescripcionProducto);
}
function GetFechaCompra()
{
    return($this->FechaCompra);
}
function GetCantidad()
{
    return($this->Cantidad);
}
function GetUnidadCompra()
{
    return($this->UnidadCompra);
}
function GetDependencia()
{
    return($this->Dependencia);
}
function GetNucleo()
{
    return($this->Nucleo);
}
//.....Fin.....

function MostrarDatosCompras($NuevaConn)
{
```



```

$query = "Select * from UDO.Registro_Compras order by fecha_compra";
$this->BuscarDatosCompras($NuevaConn,$query);
}
//.....Fin.....

function CodigosCompras($conn)
{
    $query = "select UDO.REGISTRO_COMPRAS.CODIGO_COMPRA from
UDO.Registro_Compras";
    $stmt=ociparse($conn,$query);
    ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
    return($stmt);
}
//*****

function SelectCodigosCompras($conn) // Crea un menú desplegable con todos
los códigos de las compras
{
    echo '<select name="selectCodigoCompras" size="1">';
    $stmt = $this->CodigosCompras($conn);
    $i=0;
    while(ocifetch($stmt))
    {
        $column_value = OCIResult($stmt,1);
        echo '<option value="'. $column_value. "'>
            '.$column_value.</option>';
        $i++;
    }
    echo '</select>';
}

```

```

//*****MostrarDatos desde la BD
*****

function BuscarDatosCompras($conn,$query) // Ejecuta el query que es pasado
por parámetro
{
    $stmt = ociparse($conn,$query);
    ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
    $this->DibujarTabla($conn,$stmt);
}
//.....Fin de la
funcion.....

function CargarInstancia($conn,$codigo)
{
    $query = 'select "UDO"."REGISTRO_COMPRAS"."CODIGO_COMPRA",
                "UDO"."REGISTRO_COMPRAS"."CODIGO_PRODUCTO",

"UDO"."REGISTRO_COMPRAS"."FECHA_COMPRA",

"UDO"."REGISTRO_COMPRAS"."DEPENDENCIA",

"UDO"."REGISTRO_COMPRAS"."NUCLEO",

"UDO"."REGISTRO_COMPRAS"."CANTIDAD"
                from "UDO"."REGISTRO_COMPRAS"
                where
("UDO"."REGISTRO_COMPRAS"."CODIGO_COMPRA" = '.$codigo.)';

    $stmt = ociparse($conn,$query);

```



```

    $NombreColumna=1;
    echo "</tr>";
    echo "<tr>";
}
while(ocifetch($stmt))
{
    echo "<tr>";
    for($j=1;$j<=$numcolumnas;$j++)
    {
        $column_value = OCIResult($stmt,$j);
        echo "<td >&nbsp;". $column_value."&nbsp;</td>";
    }
}
} //.....Fin de la
funcion.....

```

```

//***** Cargar Datos
desde la BD *****

```

```

function CargarDatosCompras($NuevaConn)
{
    $query = "insert into UDO.Registro_Compras (CODIGO_COMPRA,
        CODIGO_PRODUCTO,
        FECHA_COMPRA,
        CANTIDAD,
        DEPENDENCIA,

```

```

NUCLEO)

VALUES

($this->CodigoCompra,

    $this->CodigoProducto,
                                TO_DATE('$this->FechaCompra',    'DD-MM-
YYYY'),

    '$this->Cantidad',
    '$this->Dependencia',
    '$this->Nucleo');"
$stmt = ociparse($NuevaConn,$query);
ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
}
//.....Fin la funcion .....

//*****Actualizar Datos desde la
BD******/
function ActualizarDatosCompras($NuevaConn,$CodigoCompra)
{
    $query = "UPDATE    UDO.REGISTRO_COMPRAS    SET
CODIGO_COMPRA = $this->CodigoCompra ,
                                CODIGO_PRODUCTO = $this->CodigoProducto,
FECHA_COMPRA = TO_DATE('$this->FechaCompra',    'dd-mm-yyyy') ,

DEPENDENCIA = '$this->Dependencia' ,

```

```

NUCLEO = '$this->Nucleo',

CANTIDAD = $this->Cantidad

UDO.REGISTRO_COMPRAS.CODIGO_COMPRA =      $CodigoCompra";

$stmt = ociparse($NuevaConn,$query);
ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
}
//.....Fin de la funcion.....

//*****Borrar          Datos          de          la
Bd*****
function DeleteCompra($NuevaConn,$CodigoCompra)
{
    $query = "Delete from UDO.REGISTRO_COMPRAS where
UDO.REGISTRO_COMPRAS.CODIGO_COMPRA =      $CodigoCompra";
    $stmt = ociparse($NuevaConn,$query);
    ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
} }
?>

```

5.5.4 Pruebas por Unidad.

Las pruebas por unidad son empleadas con la finalidad de probar el correcto funcionamiento de un módulo del subsistema, de esta forma se garantiza que cada módulo funcione correctamente por separado.

Para realizar las pruebas por unidad a los componentes del subsistema Administrar Orden de Compras se identifican un conjunto de valores que pueden ser ingresados por un usuario, dichos valores son formulados como clases de equivalencia que permitan abarcar las ocurrencias de un evento de inserción de datos.

Identificación de las clases de Equivalencia

- 1.Sólo números.
- 2.Sólo caracteres.
- 3.Caracteres y números.
- 4.Ningún carácter.
- 5.Formato de Fecha

Grupo de Tipos de Entradas de Datos

1. *Login*: hace referencia al campo de ingreso del login del usuario, este debe ser sólo alfanumérico.

2. *Contraseña*: hace referencia al campo de ingreso del password del usuario, este debe ser alfanumérico.

3. *Código de Compra*: hace referencia al campo de entrada del código de la orden de compra a registrar, este debe ser solo numérico.

4. *Código de Producto*: hace referencia al campo de entrada del código del producto a registrar en la orden de compra. Este debe ser numérico, el cual representa el id del elemento seleccionado en el control.

5. *Fecha*: hace referencia al campo de la fecha de la orden de compra a registrar (dd/mm/aa), este debe ser alfanumérico o vacío.

6. *Fecha de Vencimiento*: hace referencia al campo de la fecha de vencimiento de la orden de compra (dd/mm/aa), este debe ser alfanumérico o vacío.

7. *Dependencia*: hace referencia al campo donde se selecciona el departamento que realiza la gestión de orden de compra. Este debe ser alfabético, el cual representa el id del elemento seleccionado en el control.

8. *Núcleo*: hace referencia al campo donde se selecciona el núcleo que realiza la gestión de orden de compra. Este debe ser alfabético, el cual representa el id del elemento seleccionado en el control.

A continuación en la *tabla 5.1* y *5.2*, se representan las clases de equivalencia:

TABLA 5.1. Clases de Equivalencia para Administrar Orden de Compra (1/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
1	12345	1	X	
1	ASsfdg	2	X	
1	Dsg457	3	X	
1	“ ”	4		X
2	68679	1	X	
2	dfgDSGS	2	X	
2	SDF456	3	X	
2	“ ”	4		X
3	12345	1	X	
3	ASsfdg	2		X
3	Dsg457	3		X
3	“ ”	4		X

4	68679	1	X	
4	dfgDSGS	2		X
4	SDF456	3		X
4	“ ”	4		X
5	12345	1	X	
5	ASsfdg	2		X
5	Dsg457	3	X	
5	“ ”	4	X	
5	5-12-09	5	X	
6	12345	1	X	
6	ASsfdg	2		X
6	Dsg457	3	X	
6	“ ”	4	X	
6	5-12-09	5	X	

[Fuente: Propia]

TABLA 5.2. Clases de Equivalencia para Administrar Orden de Compra (2/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
7	12345	1		X
7	dfgDSGS	2	X	
7	Dsg457	3		X
7	“ ”	4		X
8	18523	1		X
8	ASsfdg	2	X	
8	Dsg457	3		X
8	“ ”	4		X

[Fuente: Propia]

En las *tabla 5.3* y *5.4* se presentan casos de prueba para incluir datos, de acuerdo a las clases de equivalencia establecidas previamente. Si se cumple las clases

involucradas con dicho dato la salida será válida.

TABLA 5.3. Casos de Prueba para Administrar Orden de Compra (1/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
1	12345	1	X	
1	Marit	2	X	
1	Marit17	3	X	
1	“ ”	4		X
2	68679	1	X	
2	Marit	2	X	

[Fuente: Propia]

TABLA 5.4. Casos de Prueba para Administrar Orden de Compra (2/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
2	Marit17	3	X	
2	“ ”	4		X
3	12345	1	X	
3	ASsfdg	2		X
3	Assfdg17	3		X
3	“ ”	4		X
4	68679	1	X	
4	dfgDSGS	2		X
4	SDF456	3		X
4	“Seleccione”	4		X
5	081209	1	X	
5	ASsfdg	2		X
5	03/05/84	3	X	
5	“ ”	4	X	

5	5-12-09	5		X
6	101209	1		X
6	ASsfdg	2		X
6	5-12-09	3	X	
6	“ ”	4	X	
6	14-11-99	5	X	
7	12345	1		X
7	Eléctrica	2	X	
7	Electrica8	3		X
7	“ Seleccione ”	4		X
8	18523	1		X
8	Anzoátegui	2	X	
8	Anzoategui15	3		X
8	“ Seleccione ”	4		X

[Fuente: Propia]

5.5.5 Pruebas de Integración.

Después de verificar el diagrama de componentes se procede a comprobar la eficiencia de un conjunto de componentes integrados, es decir, a continuación se llevan a cabo las pruebas de integración. Se explica el proceso de integración del sistema utilizando como ejemplo *registrarordendecompras.php*.

A continuación se muestra el diagrama de implementación para el subsistema Administrar Orden de Compra:

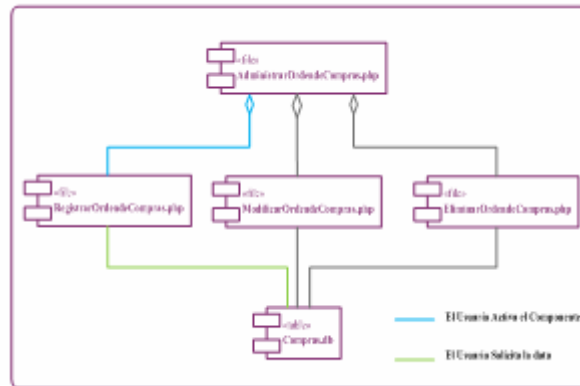


FIGURA 5.7. Diagrama de Implementación para Administrar Orden de Compras

[Fuente: Propia]

Registrar Orden de Compras.

Procedimiento de Prueba:

1.El usuario activa el caso de uso Administrar Orden de Compra y accede a la interfaz de usuario correspondiente.

2.La interfaz carga el formulario con los controles destinados para el registro de orden de compras.

3.Luego el usuario introduce en el campo de Código de Compra el número correspondiente a la orden de compra a registrar.

4. En el campo de Código de Producto selecciona el código correspondiente al producto a registrar en la orden de compras.

5. Luego para ingresar la fecha (Opcional), se deber introducir la fecha

correspondiente a la orden de compra que se está registrando.

6. Para ingresar la fecha de vencimiento (Opcional), se debe introducir la fecha de vencimiento correspondiente a la orden de compra que se está registrando.

7. El usuario procede ahora a ingresar la Dependencia. Es decir, el usuario escoge la dependencia de donde se está gestionando la orden de compra, y se carga automáticamente.

8. Luego el usuario procede a ingresar el Núcleo. Es decir, el usuario escoge el núcleo donde se gestione la orden de compra, y se carga automáticamente.

9. El usuario presiona el botón “Registrar” en la interfaz.

10. Se validan los campos requeridos. De haber un campo vacío no realizará el registro y mostrara un mensaje describiendo el problema.

11. Luego de la validación, se carga el registro de la orden de compra en la base de datos SAAP.

A continuación se muestran una serie de figuras que muestran el proceso anteriormente mencionado.

The screenshot displays the 'REGISTRAR ORDEN DE COMPRAS' form within the SAAP system. The interface includes a header with the SAAP logo and the text 'Sistema Administrativo del Área de Postgrado'. Below the header, there is a title bar for the form. The form contains several input fields: 'Fecha de Compra' (with a date picker), 'Codigo de Compra', 'Dependencia', and 'Nucleo'. A table for product selection is present, with columns for 'Selección de Cód del Producto', 'Cantidad', 'Descripción del Producto', and 'Cantidad'. The table is currently empty. At the bottom of the form, there are buttons for 'Borrar', 'Cancelar', 'Registrar', and 'Eliminar Item'.

FIGURA 5.8. Ingreso de datos Orden de Compras

[Fuente: Propia]

5.6 Segunda Iteración.

En esta iteración se implementará el caso de uso *Administrar Pagos*.

5.6.1 Implementación de Subsistemas.

Se identificarán los subsistemas de implementación necesarios para realizar la implementación del caso de uso *Administrar Pagos*.

Subsistema de Implementación Administrar Pagos.

El subsistema de implementación *Administrar Pagos* implementa al subsistema de diseño del mismo nombre.

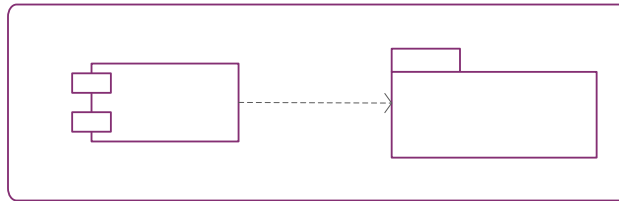


FIGURA 5.9. Subsistema de Implementación Administrar Pagos

[Fuente: Propia]

5.6.2 Diagrama de Componente.

A continuación se muestra el diagrama de componente del subsistema de **Administrar Pagos** implementación Administrar Pagos:

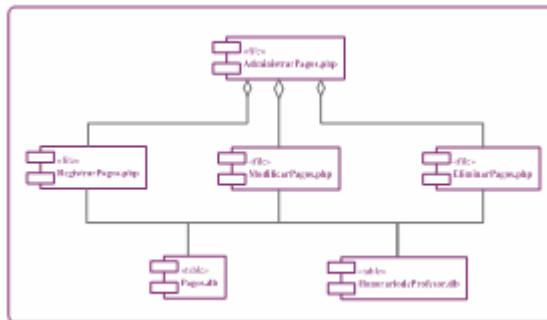


FIGURA 5.10. Diagrama de Componente para Administrar Pagos

[Fuente: Propia]

En la *Figura 5.10.* Se observa que el subsistema de implementación Administrar Pagos; posee una relación de agregación con los componentes RegistrarPagos.php, ModificarPagos.php, y EliminarPagos.php, los cuales forman parte del funcionamiento del componente Administrar Pagos.

5.6.3 Implementación de las Clases.

En esta parte, se realizará la implementación en código de las clases de diseño correspondientes al caso de uso *Administrar Pagos*, las cuales fueron definidas en la fase de elaboración. A continuación se detalla la clase de interfaz que posee este subsistema.

Clases de Interfaz.

La interfaz propia del caso de uso Administrar Pagos es la siguiente:

:IU Principal Administración de Pagos.

La clase :IU Principal Administración de Pagos es implementada dentro del componente *administrarpagos.php*, del subsistema Administrar Pagos.

Vista De La Ventana:



The screenshot shows a web application window titled "SAAP Sistema Administrativo Area de Postgrado". Below the title bar, there is a navigation menu with options like "Inicio", "Pagos", "Cuentas", and "Reportes". The main content area displays a table with the following data:

numero_pago	descripcion	fecha_pago	cantidad	dependencia	usuario	tipo_pago
22	0		0			
28			0			
16			0			
222			0			
333			0			
1000		24-03-89	0			
8		26-02-84	0			
44456		03-02-07	0			

FIGURA 5.11. Vista de la Ventana de la IU Principal Administración de Pagos

[Fuente: Propia]

A continuación se muestra el código del componente *administrarpagos.php*:

```
<?php
class Pagos
{
//Variables de los datos de compra
var $NumeroPago;
var $CodigoProducto;
var $FechaPago;
var $Descripción;
var $Cantidad;
var $Dependencia; //Departamento de postgrado que solicita la compra
var $Nucleo; //Nucleo a la cual pertenece el departamento del postgrado
var $Tipo_Beneficiario;

//.....

function Pagos() // Constructor de la clase
{
// Inicializando las variables
$this->NumeroPago = 0;
$this->CodigoProducto = 0;
$this->FechaPago= date("d-m-Y");
$this->Descripcion = "x";
$this->Cantidad = 0;
$this->Dependencia = "x";
$this->Nucleo = "x";
$this->Tipo_Beneficiario = "x";
} // fin Constructor
```

```

//.....Fin.....
....

//***** SET*****

//          *****Datos          de          las
compras*****

function SetNumeroPago($valor)
{
    $this->NumeroPago = $valor;
}
function SetCodigoProducto($valor)
{
    $this->CodigoProducto = $valor;
}
function SetFechaPago($valor)
{
    $this->FechaPago = $valor;
}
function SetDescripcion($valor)
{
    $this->Descripcion = $valor;
}
function SetCantidad($valor)
{
    if($valor==NULL)

```

```

    {
        $this->Cantidad = 0;
    }
    else {
        $this->Cantidad = $valor;
    }
}
function SetDependencia($valor)
{
    $this->Dependencia = $valor;
}
function SetNucleo($valor)
{
    $this->Nucleo = $valor;
}

function SetTipo_Beneficiario($valor)
{
    $this->Tipo_Beneficiario = $valor;
}

//.....Fin.....

//*****

Get *****

function GetNumeroPago()
{

```

```
        return($this->NumeroPago);
    }
    function GetCodigoProducto()
    {
        return($this->CodigoProducto);
    }
    function GetFechaPago()
    {
        return($this->FechaPago);
    }
    function GetDescripcion()
    {
        return($this->Descripcion);
    }
    function GetCantidad()
    {
        return($this->Cantidad);
    }
    function GetDependencia()
    {
        return($this->Dependencia);
    }
    function GetNucleo()
    {
        return($this->Nucleo);
    }

    function GetTipo_Beneficiario()
    {
```

```

return($this->Tipo_Beneficiario);
}

//.....Fin.....
function MostrarDatosPagos($NuevaConn)
{
    $query = "Select * from UDO.Registros_Pagos";
    $this->BuscarDatosPagos($NuevaConn,$query);
}

//*****MostrarDatos desde la BD
*****

function NumerosPagos($conn)
{
    $query = "select UDO.REGISTROS_PAGOS.NUMERO_PAGO from
UDO.Registros_Pagos";
    $stmt=ociparse($conn,$query);
    ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
    return($stmt);
}

//*****

function SelectNumerosPagos($conn) // Crea un menú desplegable con todos
los códigos de las compras
{
    echo '<select name="selectNumeroPagos" size="1">';
    $stmt = $this->NumerosPagos($conn);
    $i=0;
    while(ocifetch($stmt))

```

```

{
  $column_value = OCIResult($stmt,1);
  echo '<option value="'. $column_value. "'>
    '$column_value.</option>';
  $i++;
}
echo '</select>';
}

```

function BuscarDatosPagos(\$conn,\$query) // Ejecuta el query que es pasado por parámetro

```

{
  $stmt = ociparse($conn,$query);
  ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
  $this->DibujarTabla($conn,$stmt);
}
//.....Fin de la
funcion.....

```

```

function CargarInstanciaPagos($conn,$numpago)
{
  $query = 'select "UDO"."REGISTROS_PAGOS"."NUMERO_PAGO",
"UDO"."REGISTROS_PAGOS"."DESCRIPCION",
"UDO"."REGISTROS_PAGOS"."FECHA_PAGO",
"UDO"."REGISTROS_PAGOS"."CANTIDAD",
"UDO"."REGISTROS_PAGOS"."DEPENDENCIA",
"UDO"."REGISTROS_PAGOS"."NUCLEO",
"UDO"."REGISTROS_PAGOS"."TIPO_BENEFICIARIO"

```

from

```
"UDO"."REGISTROS_PAGOS"                                where
("UDO"."REGISTROS_PAGOS"."NUMERO_PAGO" = '.$numpago.');
```

```

$stmt = ociparse($conn,$query);
ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
while(ocifetch($stmt))
{
    $this->NumeroPago = OCIResult($stmt,1);
    $this->Descripcion = OCIResult($stmt,2) ;
    $this->FechaPago = OCIResult($stmt,3);
    $this->Cantidad = OCIResult($stmt,4);
    $this->Dependencia = OCIResult($stmt,5);
    $this->Nucleo = OCIResult($stmt,6);
    $this->Tipo_Beneficiario = OCIResult($stmt,7);

}
return;
}

```

```

//*****Dibuja una tabla
*****

```

```

function DibujarTabla($conn,$stmt) // Dibuja una tabla con resultado de
ejecutar un query
{
    $numcolumnas=OCIResult($stmt);

```

```

$NombreColumna=0;
if($NombreColumna==0)// La cabecera se hace sólo una vez.
{
  for($n=1;$n<=$numcolumnas;$n++)
  {
    $column_name = OCIColumnName($stmt,$n);
    echo          '<th          bgcolor="#5C9B9E"
>&nbsp;'.strtolower($column_name).'&nbsp;</th>';
  }
  $NombreColumna=1;
  echo "</tr>";
  echo "<tr>";
}
while(ocifetch($stmt))
{
  echo "<tr>";
  for($j=1;$j<=$numcolumnas;$j++)
  {
    $column_value = OCIResult($stmt,$j);
    echo "<td >&nbsp;". $column_value."&nbsp;</td>";
  }
}
}//.....Fin          de          la
funcion.....

```

```

//***** Cargar Datos
desde la BD *****

```



```
function CargarDatosPagos($NuevaConn)
{
    $query = "INSERT INTO UDO.REGISTROS_PAGOS (NUMERO_PAGO
, DESCRIPCION

,FECHA_PAGO

,CANTIDAD

,DEPENDENCIA

,NUCLEO

,TIPO_BENEFICIARIO)

VALUES

($this->NumeroPago

,$this->Descripcion'

    ,TO_DATE('$this->FechaPago','DD-MM-YYYY')

,$this->Cantidad

,$this->Dependencia'

,$this->Nucleo'

,$this-
```

```

>Tipo_Beneficiario)";
    $stmt = ociparse($NuevaConn,$query);
    ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
}

//.....Fin de la funcion.....

//*****Actualizar Datos desde la BD
*****

function ActualizarDatosPagos($NuevaConn,$NumeroPago)
{
    $query = "UPDATE UDO.REGISTROS_PAGOS SET NUMERO_PAGO =
$this->NumeroPago ,
                DESCRIPCION = '$this->Descripcion' ,

FECHA_PAGO= TO_DATE('$this->FechaPago','DD-MM-YYYY') ,

CANTIDAD = $this->Cantidad ,

DEPENDENCIA = '$this->Dependencia' ,
                NUCLEO
= '$this->Nucleo' ,

TIPO_BENEFICIARIO = '$this->Tipo_Beneficiario'
                WHERE
UDO.REGISTROS_PAGOS.NUMERO_PAGO = $NumeroPago" ;
    $stmt = ociparse($NuevaConn,$query);

```

```

        ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
    }

    //.....Fin de la
funcion.....

    /*******Borrar Datos desde la BD *****/
    function BorrarRegistro($NuevaConn,$NumeroPago)
    {
        $query = "Delete from UDO.REGISTROS_PAGOS where
UDO.REGISTROS_PAGOS.NUMERO_PAGO=$NumeroPago";
        $stmt = ociparse($NuevaConn,$query);
        ociexecute($stmt,OCI_COMMIT_ON_SUCCESS);
    }
    //.....Fin de la
funcion.....

}

```

5.6.4 Pruebas por Unidad.

Para realizar las pruebas por unidad a los componentes del subsistema Administrar Pagos se identifican un conjunto de valores que pueden ser ingresados por un usuario, dichos valores son formulados como clases de equivalencia que permitan abarcar las ocurrencias de un evento de inserción de datos.

Identificación de las clases de Equivalencia

1. Sólo números.

2. Sólo caracteres.
3. Caracteres y números.
4. Ningún carácter.
5. Formato de Fecha

Grupo de Tipos de Entradas de Datos

1. *Login*: hace referencia al campo de ingreso del login del usuario, este debe ser sólo alfanumérico.

2. *Contraseña*: hace referencia al campo de ingreso del password del usuario, este debe ser alfanumérico.

3. *Número de Pagos*: hace referencia al campo de entrada del número de pagos a registrar, este debe ser solo numérico.

4. *Fecha de Pago*: hace referencia al campo de la fecha del registro del pago (dd/mm/aa), este debe ser alfanumérico o vacío.

5. *Dependencia*: hace referencia al campo donde se selecciona el departamento que realiza la gestión del registro de pago. Este debe ser alfabético, el cual representa el id del elemento seleccionado en el control.

6. *Núcleo*: hace referencia al campo donde se selecciona el núcleo que realiza la gestión del registro de pago. Este debe ser alfabético, el cual representa el id del elemento seleccionado en el control.

7. *Beneficiario*: hace referencia al campo de entrada del tipo de beneficiario al que se le realiza el pago correspondiente. Este debe ser alfabético, el cual representa

el id del elemento seleccionado en el control.

8. *Cantidad*: hace referencia al campo de entrada de la cantidad total del pago que se le realiza al beneficiario. Este debe ser numérico.

9. *Descripción*: hace referencia al campo de entrada de la descripción del concepto del pago que se está realizando. Este debe ser alfanumérico.

A continuación en la *tabla 5.5* y *5.6* se representan las clases de equivalencia:

TABLA 5.5. Clases de Equivalencia para Administrar Pagos (1/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
1	12345	1	X	
1	ASsfdg	2	X	
1	Dsg457	3	X	
1	“ ”	4		X
2	68679	1	X	
2	dfgDSGS	2	X	
2	SDF456	3	X	
2	“ ”	4		X
3	12345	1	X	
3	ASsfdg	2		X
3	Dsg457	3		X
3	“ ”	4		X

[Fuente: Propia]

TABLA 5.6. Clases de Equivalencia para Administrar Pagos (2/2).

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
4	68679	1	X	
4	dfgDSGS	2		X
4	SDF456	3	X	
4	“ ”	4	X	
4	05-12-09	5	X	
5	1234	1		X
5	dfgDSGS	2	X	
5	Dsg457	3		X
5	“ ”	4		X
6	12345	1		X
6	ASsfdg	2	X	
6	Dsg457	3		X
6	“ ”	4		X
7	12345	1		X
7	asdfH	2	X	
7	asdfH 7	3		X
7	“ ”	4		X
8	12345	1	X	
8	ASsfdg	2		X
8	Dsg457	3		X
8	“ ”	4		X
9	12345	1	X	
9	Adfs	2	X	
9	Adfs7	3	X	

[Fuente: Propia]

En las *tabla 5.7* y *5.8* se presentan casos de prueba para incluir datos, de acuerdo a las clases de equivalencia establecidas previamente. Si se cumple las clases involucradas con dicho dato la salida será válida.

TABLA 5.7. Casos de Prueba para Administrar Pagos (1/2)

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
1	12345	1	X	
1	Marit	2	X	
1	Marit17	3	X	
1	“ ”	4		X
2	68679	1	X	
2	Marit	2	X	
2	Marit17	3	X	
2	“ ”	4		X
3	12345	1	X	
3	ASsfdg	2		X
3	Assfdg17	3		X
3	“ ”	4		X
4	081209	1	X	
4	ASsfdg	2		X
4	03/05/84	3	X	
4	“ ”	4	X	
4	5-12-09	5	X	
5	12345	1		X
5	Electrica	2	X	
5	Electrica5	3		X

[Fuente: Propia]

TABLA 5.8. Casos de Prueba para Administrar Pagos (2/2)

GRUPO	CASO DE PRUEBA	CLASE DE EQUIVALENCIA	VÁLIDO	INVÁLIDO
5	“Sleccione”	4		X
6	101209	1		X
6	Anzoategui	2	X	
6	Anzoategui5	3		X
6	“Seleccione”	4		X
7	12345	1		X
7	Profesor	2	X	
7	Profesor7	3		X
7	“ Seleccione ”	4		X
8	12345	1	X	
8	ASsfdg	2		X
8	Dsg457	3		X
8	“ ”	4		X
9	12345	1		X
9	Horas	2	X	
9	10 Horas	3	X	
9	“ ”	4		X

[Fuente: Propia]

5.6.5 Pruebas de Integración.

Después de verificar el diagrama de componentes se procede a comprobar la eficiencia de un conjunto de componentes integrados, es decir, a continuación se llevan a cabo las pruebas de integración. Se explica el proceso de integración del sistema utilizando como ejemplo *registrarpagos.php*.

A continuación se muestra el diagrama de implementación para el subsistema Administrar Pagos:

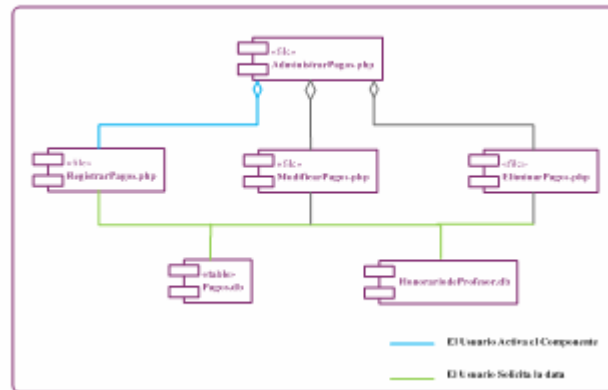


FIGURA 5.12. Diagrama de Implementación para Administrar Pagos

[Fuente: Propia]

Registrar Pagos.

Procedimiento de Prueba:

1 El usuario activa el caso de uso Administrar Pagos y accede a la interfaz de usuario correspondiente.

2. La interfaz carga el formulario con los controles destinados para el registro de pagos.

3. Luego el usuario introduce en el campo de número de pagos el número correspondiente al pago a registrar.

4. Luego para ingresar la fecha del pago (Opcional), se deber introducir la fecha

correspondiente al pago que se está registrando.

5. El usuario procede ahora a ingresar la dependencia. Es decir, el usuario escoge la dependencia donde se gestione el pago, y se carga automáticamente.

6. Luego el usuario procede a ingresar el núcleo. Es decir, el usuario escoge el núcleo donde se gestione el pago, y se carga automáticamente.

7. Posteriormente el usuario procede ahora a ingresar el beneficiario. Es decir, el usuario escoge el tipo de beneficiario al cual se le realiza su pago correspondiente, y se carga automáticamente.

8. El usuario introduce en el campo cantidad el número correspondiente a un pago que se le realiza a un tipo de beneficiario.

9. El usuario introduce en el campo descripción el concepto de un pago que se le realiza a un tipo de beneficiario.

10. El usuario presiona el botón “Registrar” en la interfaz.

11. Se validan los campos requeridos. De haber un campo vacío no realizará el registro y mostrara un mensaje describiendo el problema.

12. Luego de la validación, se carga el registro pago en la base de datos. SAAP y Base de datos externa

A continuación se muestran una serie de figuras que muestran el proceso anteriormente mencionado.

The screenshot shows the SAAP interface for recording a payment. At the top, there is a blue header with the SAAP logo and the text 'Sistema Administrativo del Área de Postgrado'. Below the header, the title 'REGISTRAR PAGO' is centered. To the right of the title is a date selection field labeled 'Fecha de Pago'. The main form consists of several input fields: 'Número de Pago', 'Dependencia', 'Núcleo', 'Beneficiario', 'Cantidad', and 'Descripción'. At the bottom of the form, there are three buttons: 'Borrar', 'Cancelar', and 'Registrar'.

FIGURA 5.13. Ingreso de datos de Pagos.

[Fuente: Propia]

5.8 Planificación de la Siguiete Fase.

5.8.1. Fase de Transición.

Al inicio de la esta fase, el Sistema Administrativo del Área de Postgrado SAAP cuenta con una versión Beta que abarca con los requerimientos establecidos durante las fases anteriores, se coloca en funcionamiento el sistema, incorporando e integrando sus componentes. Además se realizan las pruebas y correcciones necesarias del sistema para su implementación final. Al final de la fase de transición, el incremento del sistema se convierte en un software estable y confiable que puede ser utilizado por los distintos usuarios para ser probado. Además, se elabora el manual de usuario.

5.9. Resumen de la Fase de Construcción.

La fase de construcción se ha ejecutado exitosamente. Para lograrlo se

codificaron, probaron e integraron todos los subsistemas del *SAAP*, mediante una serie de construcciones que conllevaron a la capacidad operativa total del sistema.

Durante el desarrollo de esta fase, se le aplicaron al software pruebas de unidad y de integración que permitieron de esta forma mejorar su calidad, ya que con la corrección de los errores se garantiza la futura estabilidad operativa del mismo.

CAPÍTULO VI

FASE DE TRANSICIÓN

6.1 Introducción

La fase de transición, representa la cuarta fase que abarca el Proceso Unificado de Desarrollo de Software. Permite comprobar la calidad del software sobre la plataforma del entorno de operación, para ello, el software se entrega a los usuarios para que realicen las pruebas beta. Además, el equipo de software crea la información de soporte necesaria (por ejemplo, manuales de usuario) para el lanzamiento. Al final de la fase de transición, el incremento de software se convierte en un lanzamiento de software utilizable.



FIGURA 6.1. Flujos de Trabajo Fundamentales en la Fase de Transición

[Fuente: Jacobson, I.]

En la fase de inicio se establecieron requisitos que fueron analizados y diseñados en la fase de elaboración, para ser codificados en la fase de construcción y

finalmente implantados en la fase de transición. En esta fase se ha culminado el estudio del sistema y éste se encuentra operativo fundamentándose en el flujo de trabajo de pruebas.

6.2 Preparación de la Versión Beta

Se prepara la documentación que describe como se manipula el sistema. Estará dirigido al personal de la Coordinación de Postgrado.

Las pruebas del software las realizan los usuarios seleccionados, o los más capacitados para esta tarea. A éstos, le son proporcionados los manuales de usuario, acerca de cómo interactuar con el sistema, cómo realizar ciertas pruebas, etc. Esta última tarea es en la mayoría de los casos necesaria, debido a que, aunque la mayor parte de los requerimientos son proporcionados por el cliente, éste muchas veces puede desconocer la manera de interactuar con la implementación de esos requisitos.

6.3 Reacción a los resultados de las Pruebas

Durante el proceso de implementación, se pueden presentar ciertos problemas que deben ser recopilados y solucionados de la mejor forma posible para que la funcionalidad del sistema sea totalmente correcta.

En el sistema no se hallaron errores de mayor importancia que comprometieran la arquitectura del sistema ya desarrollado. Sólo se encontraron fallas, en relación a la codificación de determinadas aplicaciones, las cuales fueron corregidas sin complicación alguna, por lo que no hubo necesidad de desarrollar flujos adicionales de trabajo.

6.4 Ventajas del Manual de Usuario

Con este manual se busca que los distintos usuarios tengan las siguientes ventajas:

1. Facilitar el adiestramiento a los usuarios del sistema.
2. Interactuar fácilmente con el sistema.
3. Ayudar a corregir problemas que se puedan presentar al operar el sistema.

6.5 Resumen de la Fase

En esta fase se obtuvo la versión Beta del software de manera exitosa; se efectuaron las pruebas de aceptación con las cuales fueron localizadas fallas menores en algunos componentes. Estos fallos fueron atendidos sin afectar la arquitectura del sistema, concluyendo la fase final del proceso de desarrollo de software con una versión completa y corregida del producto, que cumple con los requerimientos y necesidades planteadas por la Coordinación de Postgrado, así como también, ha satisfecho los objetivos planteados al inicio del proyecto.

Los productos que se obtuvieron en esta fase son, una eficiente solución de software basada en web, la cual funcionará de acuerdo a lo estipulado, esperando supere las expectativas de los usuarios finales. También se obtuvo el manual de usuario del sistema que les servirá de guía a los usuarios finales en la interacción de las aplicaciones del sistema.

CONCLUSIONES

El sistema *SAAP* representa una herramienta automatizada, integral y confiable que facilita la gestión de las actividades administrativas desempeñadas dentro de la Coordinación de Postgrado.

La utilización del *Proceso Unificado de Desarrollo de Software* utilizado como marco de trabajo, junto con los diagramas del UML fueron herramientas determinantes y esenciales para que los requisitos funcionales recopilados se tradujeran en el correcto sistema software *SAAP*.

En la construcción de la base de datos *SAAP*, a través de las técnicas del Modelo Relacional de Base de Datos, permitió un efectivo diseño lógico de las tablas necesarias para el correcto funcionamiento del software.

Durante la *fase de inicio* se determinaron los requisitos funcionales necesarios para la elaboración del sistema, mediante entrevistas directas con el personal que labora dentro de la coordinación de postgrado; capturando de ésta manera, las verdaderas necesidades de quienes serán los futuros usuario del sistema y, traduciéndose en el punto de partida para la definición de una eficiente y completa funcionalidad.

En la *fase de construcción* se concretaron la mayoría de las funcionalidades a nivel de aplicación, que permitieron realizar las operaciones diseñadas y analizadas en las fases anteriores.

Las interfaces del sistema fueron diseñadas bajo la concepción de que el usuario se sienta motivado a interactuar con el sistema de forma rápida y sencilla.

Se logro eficazmente la integración, pruebas y documentación del funcionamiento del sistema, obteniendo un software altamente funcional, y un instructivo orientado al usuario que le sirva de soporte en la utilización del sistema.

RECOMENDACIONES

Efectuar un mantenimiento periódico a la base de datos, realizando respaldos y eliminando de las tablas los elementos con estado inactivo, para optimizar los tiempos de respuesta en el funcionamiento de las secciones del sistema y mantener la información actualizada.

Realizar la configuración inicial completa del sistema para su correcto funcionamiento, obteniendo así los mejores beneficios del software construido.

Mantener actualizados los documentos de diseños y manuales de usuarios, con respecto a cualquier modificación o actualización que se realice al sistema y llevar un registro de estos.

BIBLIOGRAFÍA

- [1] Oracle. “Management Excellence” <http://www.oracle.com/index.html>. (2008).
Visitado en Diciembre 2008.
- [2] Adobe Dreamweaver. <http://www.adobe.com/es/products/dreamweaver/> (2008).
Visitada en Noviembre 2008.
- [3] Introducción a JavaScript. <http://www.librosweb.es/javascript/capitulo1.html>
(2008). Visitada en Noviembre 2008.
- [4] AUTOR DESCONOCIDO. “Lenguaje Unificado de Modelo (UML)”
<http://www.nitsnets.com/uml/uml.swf>, Visitado en Julio 2008.
- [5] Cáceres S., “Desarrollar un Sistema de Información Gerencial bajo Plataforma Web para mejorar el Control de Gestión en el Centro de Estudio de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente”. Trabajo de Grado de Ingeniería de Sistema y Computación, Universidad de Oriente. Barcelona (2008).
- [6] Arcila A y Zacarías E., “Desarrollo de un Software como Soporte para la Automatización de las Actividades Administrativas que se llevan a cabo en una Institución Educativa”. Trabajo de Grado de Ingeniería de Sistema y Computación, Universidad de Oriente. Barcelona (2006).
- [7] Cobo A y Gómez. P, “PHP y MySQL: Tecnologías para Aplicaciones de Desarrollo de Web”. Primera Edición, Editorial Díaz Santos, Madrid (2005).

- [8] Rapino D y Zouhairi W., “Diseño de un Sistema de Información para el Proceso de Transcripción de Notas en el Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui, Implantando un Sistema de Control de Seguridad de la Base de Datos por Autenticación de la Clave Pública”. Trabajo de Grado de Ingeniería de Sistema y Computación, Universidad de Oriente. Barcelona. (2004).
- [9] Joyanes, L. “Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos”, Tercera Edición, Editorial Mc Graw Hill/Interamericana de España, Madrid (2003).
- [10] Elmasri R y Navathe S, “Fundamentos de Sistemas de Bases de Datos”, Tercera Edición, Editorial Pearson Educación, Madrid (2002).
- [11] Rojas T., “Desarrollo de un Sistema Automatizado para el Control de Estudios del Área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente”. Trabajo de Grado de Ingeniería de Sistema y Computación, Universidad de Oriente. Barcelona. (2002).
- [12] Jacobson I, Booch G y Rumbaugh, J., “El Proceso Unificado de Desarrollo de Software”, Primera Edición, Editorial Pearson Educación, Madrid (2000).
- [13] Pressman, R. “Ingeniería de Software Un Enfoque Práctico”, Tercera Edición, Editorial Mc Graw Hill/Interamericana de España, Madrid (1993).

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	DESARROLLO DE UN SOFTWARE GERENCIAL PARA LOS PROCESOS ADMINISTRATIVOS DEL ÁREA DE POSTGRADO DEL NÚCLEO DE ANZOÁTEGUI DE LA UNIVERSIDAD DE ORIENTE.
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Lugo R., Eliannys A.	CVLAC: V-16.182.554 E MAIL: ealr17@hotmail.com
Pino M., Maritzabel del V.	CVLAC: V-15.802.686 E MAIL: marit17_14@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Coordinación _____

Postgrado _____

Administrativo _____

Gerencial _____

ORACLE _____

UML _____

PHP _____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
Ingeniería y Ciencias Aplicadas	Ingeniería en Computación

RESUMEN (ABSTRACT):

El presente proyecto nace con la finalidad de mejorar la integración y control de los procesos Administrativos del Área de Postgrado del Núcleo de Anzoátegui de la Universidad de Oriente, desarrollando una aplicación gerencial bajo ambiente web. Dicha aplicación se realizó siguiendo la Metodología del Proceso Unificado de Desarrollo de Software, la cual proporciona una manera ordenada y eficiente de llevar a cabo la ejecución del proyecto a través del uso de sus fases: Inicio, Elaboración, Construcción y Transición; además el empleo de Técnicas Orientadas a Objetos específicamente el Lenguaje de Modelado Unificado (UML) y la Técnica del Modelo de Entidad Relación para la estructuración de la base de datos. Luego terminada estas dos primeras y fundamentales etapas del desarrollo de software se programó y depuró con la ayuda del usuario los procesos básicos de cada objeto de los módulos, logrando después de varias pruebas el producto final que alcanzó los objetivos planeados para este proyecto. Se aplicó también un lenguaje de programación basado en herramientas con licencias de software libre PHP (*Personal Hypertext Processor*) y para el diseño de la base de datos se utilizó herramientas de última generación soportado por el manejador Oracle8i. Así mismo se logro la creación de una plataforma que permite hacer más eficiente y dinámica la operatividad y productividad del Área de Postgrado que se involucran en el nuevo sistema y a los usuarios finales del mismo.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Ríos, Francy	ROL	CA	AS	TU	JU
				X	
	CVLAC:	V-12.291.177			
	E_MAIL	francyrios75@gmail.com			
	E_MAIL				
Cortínez, Claudio	ROL	CA	AS	TU	JU
					X
	CVLAC:	V-12.155.334			
	E_MAIL	cl_cortinez@cantv.net			
	E_MAIL				
Saettone, Mónica	ROL	CA	AS	TU	JU
			X		
	CVLAC:	V-10.948.010			
	E_MAIL	msaettone@yahoo.com			
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	10	22
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**ARCHIVO (S): TESIS**

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.Sistemasap.doc	APPLICATION/MSWORD

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J K
L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x y
z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pre-Grado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistema

INSTITUCIÓN:

Universidad de Oriente – Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**DERECHOS**

De acuerdo con el artículo 44 del reglamento de trabajo de grado: _____

"Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario". _____

Lugo R., Eliannys A.

AUTOR

Pino M., Maritzabel del V.

AUTOR

Ríos, Francy

TUTOR

Cortínez, Claudio

JURADO

Saettone, Mónica

JURADO

POR LA SUBCOMISIÓN DE TESIS