



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

TEMA:

“SISTEMA DE GESTIÓN DE PROYECTOS, PARA UN MANEJO
ADECUADO DE LA INFORMACIÓN EN LA UOCENI – ISEI DE LA
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL”

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

Autor: Alex Ricardo Paucar Medina.

Tutor: Ing. David Guevara.

Ambato - Ecuador

Octubre, 2011

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “Sistema de Gestión de Proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”, del señor Alex Ricardo Paucar Medina, egresado de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad al Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad técnica de Ambato.

Ambato Octubre 26, 2011

EL TUTOR

Ing. David Guevara

AUTORÍA

El presente trabajo de investigación titulado: “Sistema de Gestión de Proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Octubre 26, 2011

Alex Ricardo Paucar Medina
CC: 060464611-7

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. M.Sc. Oswaldo Paredes, Ing. Edison Alvarez, Ing. Hernando Buenaño, revisó y aprobó el Informe Final del trabajo de graduación titulado: “Sistema de Gestión de Proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial” , presentado por el señor Alex Ricardo Paucar Medina de acuerdo al Art. 18 del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ing. Oswaldo Paredes, M.Sc.
PRESIDENTE DEL TRIBUNAL

Ing. Edison Alvarez
DOCENTE CALIFICADOR

Ing. Hernando Buenaño
DOCENTE CALIFICADOR

DEDICATORIA

Quiero dedicarle este trabajo a Dios por haberme dado la vida y fortaleza para poder cumplir con esta meta.

A mis padres por el apoyo, consejos y confianza que día a día ponen en mí.

A mis hermanos y a mi novia Rocío que con su amistad y cariño impulsan valor en mi vida.

Alex Ricardo Paucar Medina

AGRADECIMIENTO

A Dios por ser la luz guiadora y fortalecedora de mi vida.

A mis padres Washington y Marlene por ser un ejemplo de responsabilidad y emprendimiento quienes con cariño y apoyo tanto moral como económico permitieron el cumplimiento de esta meta.

A todos los profesores que a lo largo de mi vida estudiantil me proporcionaron de conocimientos y valores, en especial quiero agradecer al Ing. David Guevara quien supo guiarme en todo el proceso de desarrollo del Temi.

Alex Ricardo Paucar Medina

ÍNDICE

CONTENIDO	PÁGINA
Aprobación del tutor	i
Autoría	ii
Aprobación de la comisión calificadora	iii
Dedicatoria	iv
Agradecimiento	v
Índice	vi

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de investigación	1
1.2 Planteamiento del problema	1
1.2.1 Contextualización	1
1.2.2 Análisis crítico	2
1.2.3 Prognosis	3
1.2.4 Formulación del problema	3
1.2.5 Preguntas directrices	3
1.2.6 Delimitación del problema	3
1.3 Justificación	4
1.4 Objetivos	5
1.4.1 Objetivo general	5
1.4.2 Objetivos específicos	5

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos	6
2.2 Fundamentación Legal	7
2.3 Categorías fundamentales	8
2.3.1 Categorías fundamentales de la Variable Independiente: Sistema de gestión de proyectos	9
2.3.1.1 Sistemas operativos	9
2.3.1.2 Sistemas de Gestión de Base de Datos (SGBD)	14
2.3.1.3 Servidores Web	22
2.3.1.4 Aplicaciones web	27
2.3.2 Categorías fundamentales de la Variable Dependiente: Manejo adecuado de la información en la UOCENI – ISEI	36
2.3.2.1 Gestión de proyectos	36
2.3.2.2 Automatización	37
2.3.2.3 Software de gestión de proyectos	37
2.4 Hipótesis	38
2.5 Variables	38
2.5.1 Variable Independiente.- Sistema de gestión de proyectos	38
2.5.2 Variable Dependiente.- Manejo adecuado de la información en la UOCENI – ISEI	38

CAPÍTULO III

METODOLOGÍA

3.1 Enfoque	39
3.2 Modalidad de la investigación	39
3.3 Niveles de investigación	39
3.4 Población y muestra	40
3.5 Operacionalización de variables	41
3.5.1 Variable independiente: Sistema de gestión de proyectos	41

3.5.2 Variable dependiente: Manejo adecuado de la información en la UOCENI-ISEI	43
3.6 Técnicas e instrumentos de investigación	45
3.7 Procesamiento de la información	45

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis e interpretación de resultados	46
4.1.1 Análisis de los resultados de las encuestas	46

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones	56
5.2 Recomendaciones	57

CAPÍTULO VI

PROPUESTA

6.1 Tema	58
6.2 Datos informativos	58
6.3 Antecedentes	58
6.4 Justificación	59
6.5 Objetivos	60
6.5.1 Objetivo general	60
6.5.2 Objetivos específicos	60
6.6 Análisis de factibilidad	60
6.6.1 Factibilidad operativa	60
6.6.2 Factibilidad económica	61
6.6.3 Factibilidad técnica	61
6.7 Fundamentación	62
6.7.1 Asp.net	62
6.7.2 MonoDevelop	62
6.7.3 XSP	62

6.7.4 JavaScript	63
6.8 Metodología	63
6.9 Modelo operativo	65
6.9.1 Análisis del sistema	65
6.9.1.1 Análisis y requerimientos del sistema	65
6.9.1.2 Diagramas UML	66
6.9.1.2.1 Diagrama de casos de uso	66
6.9.1.2.2 Diagrama de estados	80
6.9.1.2.3 Diagrama de secuencias	82
6.9.1.2.4 Diagrama de actividades	90
6.9.1.2.5 Diagrama de clases	101
6.9.1.2.6 Diagrama de despliegue	103
6.9.2 Diseño del sistema	103
6.9.2.1 Diseño de la base de datos	104
6.9.2.2 Diccionario de datos	108
6.9.2.3 Diseño de la interfaz	114
6.9.2.3.1 Diseño de salidas	116
6.9.2.3.2 Diseño de entradas	124
6.9.2.4 Diseño de Comunicación	126
6.9.3 Implementación	127
6.9.3.1 Extracto de código fuente	127
6.9.4 Pruebas	135
6.9.4.1 Pruebas de caja negra	136
6.9.4.2 Pruebas de caja blanca	136
6.9.4.2.1 Prueba del camino básico	137
6.9.5 Implantación	139
6.9.5.1 Conversión del sistema antiguo al nuevo	139
6.9.5.1.1 Conversión de datos	139
6.9.5.1.2 Capacitación al personal sobre el uso y administración del nuevo sistema	140
6.9.5.2 Instalación y Configuración	140
6.9.5.2.1 Instalación Mod_mono y XSP	141

6.9.5.2.1.1 Requisitos	141
6.9.5.2.1.2 Instalación	141
6.9.5.2.1.3 Configuración	143
6.9.5.2.2 Instalación del motor de base de datos	143
6.9.5.2.3 Instalación del sistema de gestión de proyectos	148
6.9.5.3 Aplicación de pruebas en el nuevo sistema	150
6.9.5.4 Evaluaciones	151
6.10 Conclusiones y Recomendaciones	152
6.10.1 Conclusiones	152
6.10.2 Recomendaciones	153
Bibliografía	154
Información bibliográfica de libros	154
Información bibliográfica de páginas web	154
Glosario de términos	157

ANEXOS

ANEXO 1: Decreto presidencial para el uso de software libre en la república del Ecuador	160
ANEXO 2: Estructura del cuestionario	163
ANEXO 3: Respaldo y recuperación de una Base de Datos	166

ÍNDICE DE FIGURAS

Figura 2.1 Interacción entre el SO con el resto de las partes	10
Figura 4.1. Pregunta 1	47
Figura 4.2. Pregunta2	48
Figura 4.3. Pregunta 3	49
Figura 4.4. Pregunta 4	50
Figura 4.5. Pregunta 5	51
Figura 4.6. Pregunta 6	52
Figura 4.7. Pregunta 7	53
Figura 4.8. Pregunta 8	54
Figura 4.9. Pregunta 9	55
Figura 6.1. Estructura del modelo en cascada	64
Figura 6.2. Diagrama de casos de uso Sistema de Gestión de Proyectos	67
Figura 6.3. Diagrama de estados proyecto	80
Figura 6.4. Diagrama de estados objetivo	81
Figura 6.5. Diagrama de estados lista de tareas	81
Figura 6.6. Diagrama de estados tareas	82
Figura 6.7. Diagrama de secuencias ingresar al sistema	82
Figura 6.8. Diagrama de secuencias permisos	83
Figura 6.9. Diagrama de secuencias administrar usuarios	83
Figura 6.10. Diagrama de secuencias administrar presupuesto	84
Figura 6.11. Diagrama de secuencias administrar proyectos	85
Figura 6.12. Diagrama de secuencias editar perfil	86
Figura 6.13. Diagrama de secuencias tratar proyecto	86
Figura 6.14. Diagrama de secuencias tratar objetivos	87
Figura 6.15. Diagrama de secuencias tratar tareas	87
Figura 6.16. Diagrama de secuencias tratar archivos	88
Figura 6.17. Diagrama de secuencias tratar actividades	88
Figura 6.18. Diagrama de secuencias tratar presupuesto del proyecto	89
Figura 6.19. Diagrama de secuencias salir sistema	89
Figura 6.20. Diagrama de actividades ingresar al sistema	90
Figura 6.21. Diagrama de actividades administrar usuarios	91

Figura 6.22. Diagrama de actividades administrar presupuesto	92
Figura 6.23. Diagrama de actividades administrar proyectos	93
Figura 6.24. Diagrama de actividades editar perfil	94
Figura 6.25. Diagrama de actividades tratar proyecto	95
Figura 6.26. Diagrama de actividades tratar objetivos	96
Figura 6.27. Diagrama de actividades tratar tareas	97
Figura 6.28. Diagrama de actividades tratar archivos	98
Figura 6.29. Diagrama de actividades tratar actividades	99
Figura 6.30. Diagrama de actividades tratar presupuesto del proyecto	100
Figura 6.31. Diagrama de clases	101
Figura 6.32. Diagrama de clases y sus relaciones	102
Figura 6.33. Diagrama de despliegue	103
Figura 6.34. Diagrama entidad- relación	105
Figura 6.35. Modelo relacional lógico	106
Figura 6.36. Modelo relacional físico	107
Figura 6.37. Pantalla inicio de sesión	115
Figura 6.38. Página inicial	116
Figura 6.39. Menú proyecto	117
Figura 6.40. Menú perfil	118
Figura 6.41. Menú administración	119
Figura 6.42. Información presentada en pantalla	120
Figura 6.43. Generación de reportes	121
Figura 6.44. Reporte información de usuario	121
Figura 6.45. Reporte registro de actividades	122
Figura 6.46. Reporte movimiento de partidas	123
Figura 6.47. Adición de información	124
Figura 6.48. Actualización de información	125
Figura 6.49. Filtración de información	126
Figura 6.50. Diseño de Comunicación	127
Figura 6.51. Grafo de validación solo números	138
Figura 6.52. Instalar PostgreSQL	144
Figura 6.53. Directorio de instalación	144

Figura 6.54. Directorio de datos	145
Figura 6.55. Contraseña del usuario postgres	145
Figura 6.56. Puerto	146
Figura 6.57. Opciones Avanzadas	146
Figura 6.58. Listo para instalar	147
Figura 6.59. Instalando	147

ÍNDICE DE TABLAS

Tabla 2.1. Comparativa entre PostgreSQL, My SQL, SQL Server y Oracle	21
Tabla 2.2. Comparativa entre Apache e Internet Information Services	26
Tabla 2.3. Comparativa entre ASP.NET Y PHP	35
Tabla 3.1. Operacionalización de variable independiente	42
Tabla 3.2. Operacionalización de variable dependiente	44
Tabla 4.1. Cuadro porcentual pregunta 1.....	46
Tabla 4.2. Tabla porcentual pregunta 2	47
Tabla 4.3. Tabla porcentual pregunta 3	48
Tabla 4.4. Tabla porcentual pregunta 4	49
Tabla 4.5. Tabla porcentual pregunta 5	50
Tabla 4.6. Tabla porcentual pregunta 6	51
Tabla 4.7. Tabla porcentual pregunta 7	52
Tabla 4.8. Tabla porcentual pregunta 8	53
Tabla 4.9. Tabla porcentual pregunta 9	54
Tabla 6.1. Especificación ingresar al sistema	68
Tabla 6.2. Especificación administrar usuarios	69
Tabla 6.3. Especificación permisos	70
Tabla 6.4. Especificación administrar presupuesto	70
Tabla 6.5. Especificación administrar proyectos	72
Tabla 6.6. Especificación editar perfil	72
Tabla 6.7. Especificación tratar proyecto	73
Tabla 6.8. Especificación tratar objetivos	75
Tabla 6.9. Especificación tratar tareas	76
Tabla 6.10. Especificación tratar archivos	77
Tabla 6.11. Especificación tratar actividades	78
Tabla 6.12. Especificación tratar presupuesto del proyecto	79
Tabla 6.13. Especificación salir sistema	80
Tabla 6.14. Descripción de la tabla roles	108
Tabla 6.15. Descripción de la tabla usuario	109
Tabla 6.16. Descripción de la tabla proyecto	109
Tabla 6.17. Descripción de la tabla usuarioProyecto	109

Tabla 6.18. Descripción de la tabla hito	110
Tabla 6.19. Descripción de la tabla listaTarea	110
Tabla 6.20. Descripción de la tabla tarea	110
Tabla 6.21. Descripción de la tabla usuarioTarea	111
Tabla 6.22. Descripción de la tabla archivo	111
Tabla 6.23. Descripción de la tabla usuarioArchivo	111
Tabla 6.24. Descripción de la tabla partida	112
Tabla 6.25. Descripción de la tabla proyectoPartida	112
Tabla 6.26. Descripción de la tabla bien	112
Tabla 6.27. Descripción de la tabla registroActividad	113
Tabla 6.28. Descripción de la tabla movimientoPartida	114

RESUMEN EJECUTIVO

La UOCENI – ISEI es la encargada de recibir proyectos para coordinar, planificar y posteriormente presentar al consejo universitario para su aprobación, luego de su aprobación se encarga de llevar un seguimiento minucioso de cada una de las actividades que se realizan en el mismo, de tal manera que su avance no tenga ningún contratiempo.

Este tema de investigación surge como respuesta a la necesidad de la UOCENI – ISEI de contar con un Sistema Informático que facilite y optimice el proceso de administración de los proyectos.

Es importante el desarrollo del Sistema de Gestión de Proyectos ya que a través de él se podrá mejorar la velocidad de procesamiento, la recuperación de la información será rápida y eficiente, existirá seguridad en la información almacenada, se realizará un control y seguimiento de los proyectos automáticamente, y por ende se conseguirá facilitar y optimizar la gestión de proyectos.

Por la razón expuesta anteriormente a continuación se muestra el proceso de investigación que permitió el desarrollo de la solución informática para la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

INTRODUCCIÓN

El propósito de la presente investigación es elaborar una solución informática que satisfaga las necesidades del departamento UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial con la tecnología y herramientas de desarrollo más adecuadas.

La investigación se ha elaborado de acuerdo a la organización de información que se detalla a continuación:

En el capítulo I “EL PROBLEMA DE INVESTIGACIÓN”, se identifica el problema a investigar, además se plantea la justificación y los objetivos.

En el capítulo II “MARCO TEÓRICO”, se presentan los antecedentes investigativos, la fundamentación legal, hipótesis y el señalamiento de las variables de la hipótesis.

En el capítulo III “METODOLOGÍA”, se determina la metodología de investigación a utilizar, el enfoque, la modalidad básica de la investigación, el tipo de investigación, la población y muestra.

En el capítulo IV “ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS”, se procede al análisis e interpretación de los resultados.

En el capítulo V “CONCLUSIONES Y RECOMENDACIONES”, se presenta las conclusiones y recomendaciones del trabajo desarrollado.

En el capítulo VI “PROPUESTA”, se presenta el desarrollo de la propuesta ante el problema investigado.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de investigación

Sistema de gestión de proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.2 Planteamiento del problema

1.2.1 Contextualización

La tecnología en Ecuador es bastante escasa la cual ha traído graves consecuencias con respecto al desarrollo, ya que la falta de ésta no ha permitido que el país avance de una manera óptima en el mercado competitivo a nivel mundial. Ecuador se encuentra en una notoria desventaja con los demás países, generando un gigantesco abismo con países tecnológicamente más desarrollados.

Por su lado en la provincia de Tungurahua, existe todavía despreocupación en las instituciones ante la actualización de procesos, procesos manuales que años atrás fueron de gran ayuda pero que en este momento pasaron a ser obsoletos, impidiendo o posponiendo el desarrollo de las mismas y colocándolas en una posición desfavorable.

El departamento UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial es el encargado de realizar un almacenamiento, control y seguimiento minucioso de cada uno de los proyectos. Esta tarea representa la

ejecución de muchos procesos implícitos como: entrega de reportes e informes, elaboración de controles de horarios de los empleados, etc. que efectuados manualmente resultan tediosos y repetitivos.

Además, como en toda organización en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial el almacenamiento de información se lo realiza continuamente, ocasionando un constante crecimiento de la misma, al no estar automatizado este proceso da apertura a que cada vez sea más compleja la manipulación de los datos.

1.2.2 Análisis crítico

La falta de personal es una de las causas de que hasta la actualidad la administración de los proyectos se la siga realizando manualmente. El desempeño de cada uno de sus integrantes influye directamente en el correcto funcionamiento del departamento, por tanto al no existir una asignación adecuada de recursos humanos se ha generado las dificultades para que la administración de los proyectos no pueda ser automatizada.

Como en toda institución pública del Ecuador, la carencia de recursos económicos es un problema muy común y la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas Electrónica e Industrial no es la excepción. La automatización del proceso de gestión de proyectos implica gastos como: transporte, alimentación, pago de salarios, etc. que como mencionados anteriormente el departamento no puede costearlos.

Una solución factible para automatizar el proceso de gestión de proyectos debería basarse en la tecnología más adecuada a la infraestructura con que se cuenta en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial para aprovechar al máximo sus recursos. Esta tecnología es la web, la que además permitirá tener un sistema de fácil acceso desde cualquier lugar del mundo a través del Internet.

1.2.3 Prognosis

Si en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial no se soluciona el problema a corto plazo produciría que se disminuya el desempeño en la gestión de los proyectos, ocasionando un retroceso en el desarrollo del departamento.

1.2.4 Formulación del problema

¿La gestión manual de proyectos en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial incide en el manejo adecuado de la información?

1.2.5 Preguntas directrices

¿Cómo se maneja la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial?

¿Con qué herramientas se trabajará en el desarrollo del Sistema de gestión de proyectos?

¿Qué requerimientos exige el manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial?

1.2.6 Delimitación del problema

Desarrollo de un Sistema de gestión de proyectos para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la UTA, ubicada en la provincia de Tungurahua, cantón Ambato, ciudadela universitaria (predios Huachi), en el año 2011.

1.3 Justificación

En la UOCENI - ISEI el proceso de administración de proyectos se lo realiza manualmente, técnica que no brinda las facilidades necesarias para una buena gestión de información y que además en la actualidad está siendo dejada de lado por las bondades que nos ofrece la informática en el campo de desarrollo de sistemas.

La realización de la presente investigación busca aportar beneficios, como agilidad en el tratamiento a la información, optimización en el manejo de los procesos, generación oportuna de reportes; y a través de éstos beneficios dar solución a los problemas que ocasiona la administración manual de los proyectos en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

La realización del sistema además busca contribuir a que el personal del departamento cuente con una herramienta de trabajo que cumpla con sus expectativas y traiga satisfacción en los mismos.

El proyecto es factible técnicamente realizarse porque se posee conocimientos de trabajos similares realizados anteriormente, además se cuenta con el apoyo de profesores y se dispone de la infraestructura adecuada para el desarrollo e implantación del mismo.

El proyecto es factible operativamente realizarse porque el personal que interactuará con el sistema posee conocimientos informáticos de nivel medio y avanzado, además éste contará con interfaces sencillas y amigables, manual de usuario, y se efectuarán conferencias para la utilización del mismo.

El proyecto es factible económicamente realizarse debido a que para su desarrollo se pretende utilizar herramientas de código abierto, las mismas que se caracterizan por su bajo costo de implementación y mantenimiento.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un sistema de gestión de proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4.2 Objetivos específicos

1.4.2.1 Realizar el análisis del manejo de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistema, Electrónica e Industrial.

1.4.2.2 Determinar las herramientas adecuadas con las que se va a trabajar en el desarrollo del Sistema de Gestión de Proyectos.

1.4.2.3 Desarrollar e implantar el Sistema informático que cumpla con los requerimientos que exige la gestión de proyectos en la UOCENI – ISEI de la Faculta de Ingeniería en Sistemas, Electrónica e Industrial.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos

En la biblioteca de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de Universidad Técnica de Ambato en el año 2001 Chicaiza Marco y Martínez Gustavo elaboraron un trabajo de investigación con el tema “Sistema de Base de Datos documental de control de solicitudes y comunicaciones del departamento de operación y mantenimiento de la Empresa Eléctrica de Ambato S.A”, este trabajo servirá como apoyo en la búsqueda de la solución al problema investigado.

En la biblioteca de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de Universidad Técnica de Ambato en el año 1998 Elsa Pilar Urrutia Urrutia elaboró un trabajo de investigación con el tema “Sistema para la planificación y control de proyectos Informáticos”, este trabajo servirá como apoyo en la búsqueda de la solución al problema investigado.

En la biblioteca de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de Universidad Técnica de Ambato en el año 2002 Pérez Salinas Carmen Elizabeth y Villacis Gallo Lupe Aracely elaboraron un trabajo de investigación con el tema “Sistema Informático de Control del avance de tesis, proyectos y prácticas profesionales de la Facultad de Ingeniería en Sistemas”, este trabajo servirá como apoyo en la búsqueda de la solución al problema investigado.

2.2 Fundamentación Legal

De conformidad con la Ley de Educación Superior, expedida mediante Decreto No. 2000-16 promulgado el 4 de Mayo del 2000, publicado en el Registro Oficial No. 77 del Lunes, 15 de Mayo del 2000; y el Reglamento General a la Ley promulgada el 23 de Octubre del año 2000 y publicada en el Registro Oficial No. 195 del 31 de Octubre del 2000. El H. Concejo Universitario de la Universidad Técnica de Ambato, resuelve expedir el siguiente Estatuto:

Los artículos que confieren la creación del UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial son:

Art. 13. Nivel de Apoyo Académico

El Nivel Académico está constituido por las diferentes unidades de que brindan asistencia técnico – académica, a los niveles directivo, ejecutivo y operativo de la UTA, para la ejecución de los planes y proyectos institucionales, dentro del ámbito de sus componentes. Se conforma por:

- a. Concejos Académicos de la Facultad (CAF);
- b. Concejos Académicos de Postgrados de la Facultad (CAPF);
- c. Centro de Estudios de Postgrado (CEPOS);
- d. Centro de Investigaciones (CENI);**
- e. Centro de Desarrollo de la Docencia (CEDED);
- f. Centro de Vinculación con la Colectividad (CEVIC); Y,
- g. Otras que se crearen, sobre la base de proyectos.

Art. 14. Nivel Operativo

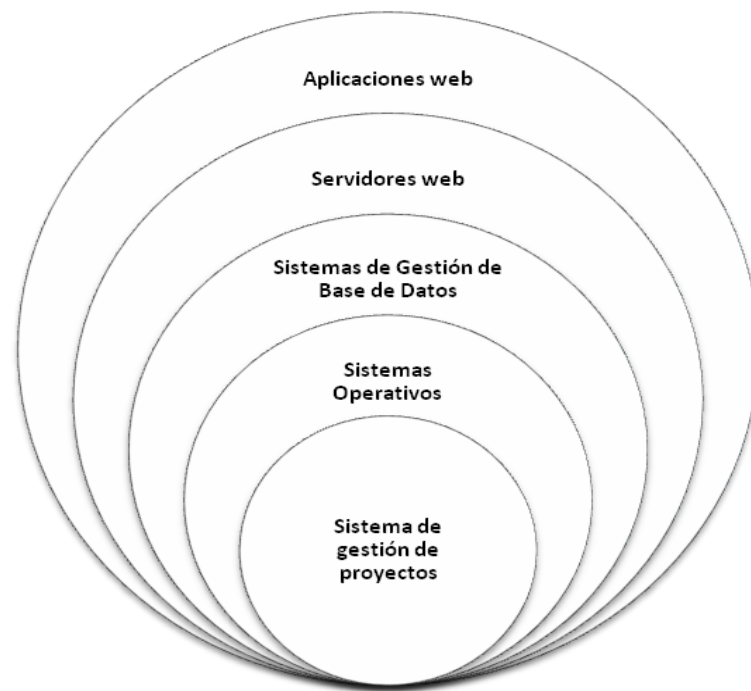
El nivel Operativo está constituido por las unidades que ejecutan los programas, planes y proyectos institucionales, dentro del ámbito de sus competencias. Está conformado por:

- a. Facultades de: Ciencias Administrativas, Contabilidad y Auditoría, Ciencias Humanas y de la Educación, Ciencia e Ingeniería Agronómica, Ingeniería Civil y Mecánica, **Ingeniería en Sistemas**

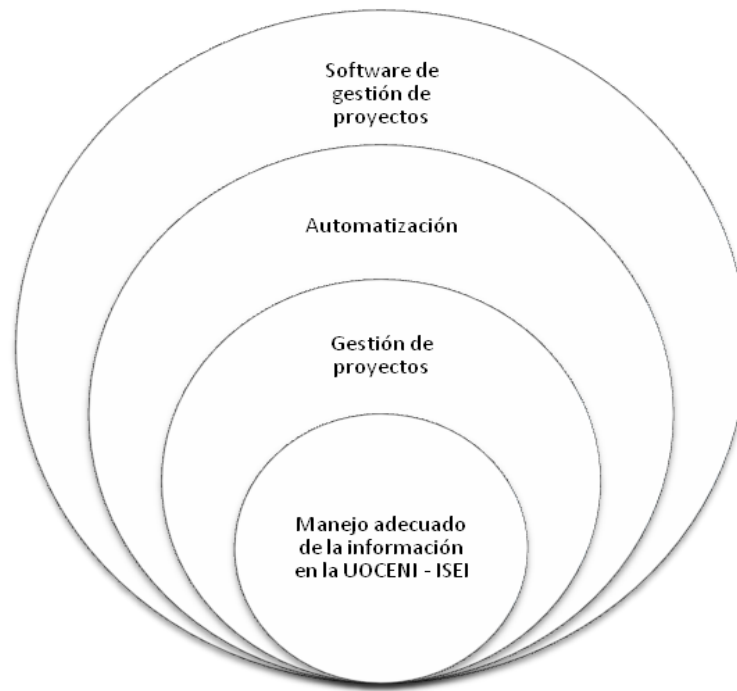
Electrónica e Industrial, Jurisprudencia y Ciencias Sociales, Ciencias de la Salud, y las que se crearen sobre la base de proyectos específicos;

- b. Carreras que son obligatoriamente parte de una Facultad;
- c. Departamentos Especializados (Idiomas, Cultura Física y las que se crearen, sobre la base de proyectos específicos);
- d. Programas de Postgrado que son parte de la Facultad;
- e. Dirección de Cultura y Comunicación Institucional;
- f. Unidades operativas de investigación que son parte de la Facultad; y,
- g. Otras que se crearen, sobre la base de proyectos específicos.

2.3 Categorías fundamentales



Variable Independiente



Variable Dependiente

2.3.1 Categorías fundamentales de la Variable Independiente: Sistema de gestión de proyectos

2.3.1.1 Sistemas operativos

Un **Sistema operativo** (S.O.) es un software que actúa de interfaz entre los dispositivos de hardware y los programas de usuario o el usuario mismo para utilizar un computador. Es responsable de gestionar, coordinar las actividades y llevar a cabo el intercambio de los recursos y actúa como intermediario para las aplicaciones que se ejecutan, el funcionamiento básico de un Sistema Operativo se muestra en la figura 2.1.



Figura 2.1 Interacción entre el SO con el resto de las partes.

Sistema Operativo. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Sistema_operativo

Funciones básicas de un S.O.

- Permite el arranque y puesta en marcha de un sistema de computación.
- Administra de manera adecuada la memoria.
- Controla y administra todos los periféricos del sistema.
- Manipula al sistema a través de comandos
- Actualiza fecha y hora del sistema.

CAZAR, Héctor (2002) Compendio de Computación Siglo XXI. Editorial Española

Clasificación de los Sistemas Operativos

Sistemas Operativos Multitarea: Procesa varias tareas al mismo tiempo.

Sistemas Operativos Monotarea: Son más primitivos y es todo lo contrario al visto anteriormente, es decir, solo pueden manejar un proceso en cada momento o solo pueden ejecutar las tareas de una en una.

Sistemas Operativos Monousuario: Son aquellos que nada más pueden atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando.

Sistemas Operativos Multiusuario: Es todo lo contrario a monousuario; y en esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten los mismos recursos. Este tipo de sistemas se emplean especialmente en redes.

Clasificación Sistemas Operativos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.mitecnologico.com/Main/ClasificacionSistemasOperativos>

Sistemas Operativos GNU/Linux y Microsoft Windows

GNU / Linux

GNU/Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, escrito por Linus Torvalds con la ayuda de miles de programadores en Internet. Unix es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de ordenadores, por lo que existen versiones de Unix para casi todos los tipos de ordenadores, desde PC y Mac hasta estaciones de trabajo y superordenadores.

Aprende Linux como si estuvieras en primero. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.taringa.net/posts/linux/1455491/aprende-linux--como-si-estuvieras-en-primero.html>

Características

- Se distribuye su código fuente, lo cual permite a cualquier persona que así lo desee hacer todos los cambios necesarios para resolver problemas que se puedan presentar, así como también agregar funcionalidad. El único requisito que esto conlleva es poner los cambios realizados a disposición del público, esto debido a su licencia.

- Es desarrollado en forma abierta por cientos de usuarios distribuidos por todo el mundo, los cuales utilizan el Internet como medio de comunicación y colaboración. Esto permite un rápido y eficiente ciclo de desarrollo.
- Cuenta con un amplio y robusto soporte para comunicaciones y redes, lo cual hace que sea una opción atractiva tanto para empresas como para usuarios individuales.
- Da soporte a una amplia variedad de hardware y se puede correr en una multitud de plataformas: PC's convencionales, computadoras Macintosh, etc., así como costosas estaciones de trabajo.
- Es un sistema operativo multitarea, multiusuario y multiplataforma.

Características GNU/Linux. (s.f). Extraído el 10 de Octubre del 2010 desde http://gbtcr.chileforge.cl/info_web/node42.html

Microsoft Windows

Es el nombre de una serie de sistemas operativos desarrollados por Microsoft desde 1981, año en que el proyecto se denominaba «*Interface Manager*».

Anunciado en 1983, Microsoft comercializó por primera vez el entorno operativo denominado *Windows* en noviembre de 1985 como complemento para MS-DOS, en respuesta al creciente interés del mercado en una interfaz gráfica de usuario (GUI). Microsoft Windows llegó a dominar el mercado de ordenadores personales del mundo, superando a Mac OS, el cual había sido introducido previamente a Windows. En octubre de 2009, Windows tenía aproximadamente el 91% de la cuota de mercado de sistemas operativos en equipos cliente que acceden a Internet. Las versiones más recientes de Windows son Windows 7 para equipos de escritorio, Windows Server 2008 R2 para servidores y Windows Phone 7 para dispositivos móviles.

Microsoft Windows. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Microsoft_Windows

Ventajas:

- Reutilización de código al ejecutar programas, uso eficiente de tareas y transparencia en la ejecución de procesos.
- Fácil manejo e instalación de dispositivos y gran compatibilidad con hardware existente en el mercado.
- Es el sistema operativo más popular en el mundo y además cuenta con un excelente soporte técnico a nivel mundial.

Desventajas:

- Los sistemas operativos de clase Windows, tienen limitaciones en cuanto a conectividad y a la adhesión a estándares se refiere. En otras palabras, son sistemas propietarios y cerrados. Si bien hay mejoras de versión a versión, el costo de propiedad y de actualización de dichas versiones es sumamente alto.
- En relación a seguridad, los sistemas operativos de Microsoft se destacan por ser ineficaces, al ser sujetos a daños por muy diversas razones (virus, correos electrónicos destructivos -gusanos, troyanos, etc.-, puertas abiertas para acceso de intrusos al sistema, etc.).
- El tiempo de respuesta del fabricante para corregir problemas de seguridad es históricamente malo e inclusive no se considera, en muchos de los casos, que hay una reacción adecuada y a la altura de las circunstancias cuando llega a haber respuesta, dejando a los usuarios de Windows en situaciones de alto riesgo.
- Windows está basado en el esquema de "seguridad a través de la obscuridad": El usuario no tiene acceso al código, por tanto no le es tan fácil encontrar errores. Sin embargo, cuando estos llegan a ser encontrados no aparece un parche sino hasta meses después, con el "service pack" correspondiente.
- Todos los programas tienen que leer y escribir en el registro del sistema constantemente, por lo cual llega a corromperse cada cierto tiempo. Además, viene codificado de tal manera que el usuario no pueda editarlo manualmente.

- En Windows y otros sistemas operativos comerciales, como ya lo hemos mencionado, la estabilidad del sistema depende de que los programadores contratados por la compañía desarrolladora encuentren los defectos y los corrijan.
- Consta de pocos programas grandes y complejos que realizan muchas tareas, cuando estos programas llegan a fallar, la estabilidad y consistencia general del sistema se ve muy afectada.

Microsoft Windows. (s.f). Extraído el 10 de Octubre del 2010 desde http://html.rincondelvago.com/microsoft-windows_2.html

2.3.1.2 Sistemas de Gestión de Base de Datos (SGBD)

Los sistemas de Gestión de Bases de Datos, son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporcionar un acceso controlado a la misma.

El SGBD es la aplicación que interactúa con los usuarios de los programas de aplicación y la base de datos.

Objetivos de un SGBD

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el Lenguaje de Manejo de Datos, lo que permite resolver el problema que presentan los sistemas de archivos, donde hay que trabajar con un conjunto fijo de consultas o la necesidad de tener muchos programas de aplicaciones.
- Proporcionar acceso controlado a la base de datos mediante:
 - Seguridad: los usuarios no autorizados no pueden acceder a la base de datos.

- Integridad: mantiene la integridad y consistencia de la base de datos.
- Control de Recurrencia: permite el acceso compartido a la base de datos.
- Control de Recuperación: restablece la base de datos después de producirse un fallo de software o hardware.
- Diccionario de datos o Catálogo: contiene la descripción de los datos de la base de datos y es accesible por el usuario.
- Gestionar la estructura física de los datos y su almacenamiento, proporcionando eficiencia en las operaciones de la base de datos y el acceso al medio de almacenamiento.
- Proporcionar un mecanismo de vistas, que permita a cada usuario tener su propia vista o visión de la base de datos.
- Eliminar la redundancia de datos, establecer una mínima duplicidad en los datos y minimizar el espacio en disco utilizado.
- Proveer interfaces procedimentales y no procedimentales, permitiendo la manipulación por usuarios interactivos y programadores.
- Independizar la estructura de la organización lógica de los datos (Independencia física).
- Independizar la descripción lógica de la Base de datos y las descripciones particulares de los diferentes puntos de vistas de los usuarios.
- Permitir una fácil administración de los datos.

Sistemas de Gestión de Base de Datos. (2005). Extraído el 10 de Octubre del 2010 desde http://machtiani.icyt.df.gob.mx/file.php/1/moddata/forum/1/9240/SGBD_2.pdf

SGBD Propietario

Es un software privativo (también llamado propietario, de código cerrado o software no libre), una distribución de este tipo solamente entrega al comprador

una copia del programa ejecutable, junto con la autorización de ejecutar dicho programa en un número determinado de computadoras.

SQL Server (versión 2008)

Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

Características de Microsoft SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Oracle (versión 11g)

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.

- Estabilidad.
- Escalabilidad.
- Soporte multiplataforma.

Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos del 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros SGBD con licencia libre como PostgreSQL, MySql o Firebird.

Sistemas gestores de Base de datos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://admin-info.galeon.com/gestor.pdf>

SGBD Libres

Es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, estudiar, modificar el software y distribuirlo modificado.

Software libre. (s.f). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Software_libre

PostgreSQL y MySQL son los SGBD de mayor representación dentro del mundo de Software Libre, principalmente por su alta integración con herramientas de programación WEB.

Características principales de PostgreSQL (versión 9.0)

- Licencia BSD. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Incorpora una estructura de datos array.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Se pueden realizar varias operaciones al mismo tiempo sobre la misma tabla sin necesidad de bloquearla.

Características principales de My SQL (versión 5.5.9)

- My SQL está disponible bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.
- Lo mejor de My SQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Consume muy pocos recursos ya sea de CPU como así también de memoria.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Mejor integración con PHP.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

- Tiene soporte para transacciones y además posee una característica única de My SQL que es poder agrupar transacciones.

Comparativa entre PostgreSQL y MySQL. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.taringa.net/posts/info/1073108/Comparativa-entre-Postgres-y-MySQL.html>

A continuación se muestra una tabla comparativa entre PostgreSQL, My SQL, SQL Server y Oracle:

Tabla comparativa entre PostgreSQL, My SQL, SQL Server y Oracle

	PostgreSQL (versión 9.0)	My SQL (versión 5.5.9)	SQL Server (versión 2008)	Oracle (versión 11g)
Tipo de licencia	Licencia BSD. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.	Licencia libre GNU GPL, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.	Software propietario perteneciente a Microsoft	Software propietario perteneciente a Oracle Corporation.
ODBC, JDBC, ADO.NET controladores disponibles	Sí.	Sí.	Sí.	Si.
Multiplataforma	Si	Si	No, desarrollado para ejecutarse en plataformas Windows.	Si
Ver diseñador gráfico (por ejemplo, se puede ver las tablas y seleccionar los campos, arrastrarlos para hacer uniones).	No	No	Sí a través de SQL Server Management Studio y Express	No
Procedimientos almacenados	Soporta funciones almacenadas, que en la práctica es muy similar.	Soporta procedimientos almacenados	Soporta procedimientos almacenados	Soporta procedimientos almacenados

Triggers	<p>Soporta</p> <p>Un disparador PostgreSQL puede ejecutar cualquier función de usuario en cualquiera de sus lenguas de procedimiento, no sólo PL / pgsql.</p>	<p>Soporta</p> <p>Los disparadores no se activan ante las actualizaciones y borrados en cascada, incluso cuando es causado por una sentencia SQL, ya que son una característica del motor InnoDB en lugar de la base de datos en su conjunto. En estos casos, MySQL ignora silenciosamente los disparadores sin emitir una advertencia.</p>	<p>Soporta</p> <p>Creados mediante T-SQL</p>	<p>Soporta</p> <p>Creados mediante PL/ SQL.</p>
Replicación	<p>Soporta la replicación: Maestro esclavo, Maestro Maestro, Asíncrona y sincrónica.</p>	<p>Soporta la replicación: Semi-sincrónica</p>	<p>Soporta 3 tipos de replicaciones: instantánea, transaccional y mezcla.</p>	<p>Soporta 2 tipos de replicaciones: básica y avanzada.</p>
Desarrollo	<p>No es controlado por una sola compañía, sino que depende de una comunidad global de desarrolladores y empresas para su desarrollo</p>	<p>MySQL es propiedad y está patrocinado por una sola empresa con fines de lucro, Oracle. MySQL AB que posee derechos sobre la mayor parte del código base.</p>	<p>Es desarrollado por la compañía de Microsoft.</p>	<p>Es desarrollado por la compañía Oracle.</p>

Tabla 2.1. Comparativa entre PostgreSQL, My SQL, SQL Server y Oracle

Por las prestaciones mencionadas y por ser la herramienta que más se adecua a la infraestructura con que se cuenta en la UOCENI – ISEI para la realización de la propuesta se seleccionó a PostgreSQL 9.0.

2.3.1.3 Servidores Web

Son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo **HTTP** (hypertext markup language).

Básicamente un servidor WEB consta de un intérprete **HTTP** el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla.

Además los servidores pueden disponer de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código **HTML** de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como **ASP.NET**, **PHP**, **Perl**, **Ajax**. Las ventajas de utilizar estos lenguajes radica en la potencia de los mismos ejecutando tareas más complejas como, por ejemplo acceder a bases de datos abstrayendo al cliente de toda la operación.

*Servidores Web. (s.f). Extraído el 10 de Octubre del 2010 desde
<http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>*

Algunos servidores web importantes son:

Apache

Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en el 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo

Ventajas

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)

Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- mod_ssl - Comunicaciones Seguras vía TLS.
- mod_rewrite - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).
- mod_dav - Soporte del protocolo WebDAV (RFC 2518).
- mod_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- mod_auth_ldap - Permite autenticar usuarios contra un servidor LDAP.
- mod_proxy_ajp - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod_cband - Control de tráfico y limitador de ancho de banda.
- mod_perl - Páginas dinámicas en Perl.
- mod_php - Páginas dinámicas en PHP.
- mod_python - Páginas dinámicas en Python.
- mod_rexx - Páginas dinámicas en REXX y Object REXX.
- mod_ruby - Páginas dinámicas en Ruby.
- mod_aspdotnet - Páginas dinámicas en .NET de Microsoft (**Módulo retirado**).
- mod_mono - Páginas dinámicas en Mono
- mod_security - Filtrado a nivel de aplicación, para seguridad.

Servidor http Apache. (s.f). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

Internet Information Services (IIS)

Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Los servicios de Internet Information Services proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Internet Information Services. (s.f). Extraído el 10 de octubre del 2010 desde http://es.wikipedia.org/wiki/Internet_Information_Services.

A continuación se muestra una tabla comparativa entre los servidores web apache e Internet Information Services:

Tabla comparativa entre Apache e Internet Information Services

	Apache (versión 2.0.64)	Internet Information Services (versión 7.5)
Tipo de licencia	Licencia ASF. Es una licencia de software libre creada por la Apache Software Foundation.	Software propietario perteneciente a Microsoft.
Multiplataforma	Si	No, solo puede ejecutarse en plataformas Windows.
Autenticación básica de acceso	Si	Si
https	Si	Si
Virtual hosting	Si	Si
CGI	Si	Si
FastCGI	Si	Si
ASP.NET	Si, con la adición de módulos	Si
GUI	No	Si
Administración por consola	Si	Si
IPv6	Si	Si

Tabla 2.2. *Comparativa entre Apache e Internet Information Services*

Una vez analizada la información acerca de las características que nos brindan los servidores web apache e Internet Information Services y con el conocimiento de que su ejecución se la realizará en un sistema operativo Linux, se escogió como software servidor web a Apache 2.0.64.

2.3.1.4 Aplicaciones web

Son aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Lenguajes de programación

Existen numerosos lenguajes de programación empleados para el desarrollo de aplicaciones web en el servidor, entre los que destacan:

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Características de PHP

Ventajas

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados *ext's* o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista

Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Inconvenientes

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

*PHP. (s.f). Extraído el 10 de octubre del 2010 desde
<http://es.wikipedia.org/wiki/PHP>*

ASP.NET

ASP.NET es una de las múltiples tecnologías que forman parte del .NET Framework, es utilizada para desarrollar aplicaciones Web. El componente fundamental de ASP.NET es el formulario Web Form. Un formulario Web Form es la página Web que los usuarios visualizan en un navegador. Una aplicación Web ASP.NET consta de uno o más formularios Web Form. Un Web Form es una página dinámica que puede acceder a recursos del servidor.

Características principales de ASP.NET

Eficiencia

El proceso de compilación hará uso automáticamente de todas las características del microprocesador

Soporte de Lenguajes

Debido a que ASP.NET forma parte del .NET Framework, podemos desarrollar aplicaciones Web ASP.NET en cualquier lenguaje basado en .NET, como

VisualBasic.Net (VB) y C#, el nuevo lenguaje creado por Microsoft con la intención de aprovechar la potencia del C++ y combinarlo con las facilidades que brinda a la programación en Internet un lenguaje como Java.

Contenido y Código, por separado

Muchos desarrolladores de sitios web han tenido que lidiar con el inconveniente de tener que crear la interfaz de usuario y el código ASP todo junto. Esta mezcla de imágenes, botones y tablas en código HTML con pedazos de código en VBScript o Jscript llegaba a ser algo muy molesto para el desarrollador.

ASP.NET viene a solucionar este problema, utilizando un criterio similar al que utiliza Visual Basic, es decir, separar la interfaz de usuario con el código.

Compatibilidad con Navegadores

ASP.NET permite crear una página web que funcionará correctamente en todos los navegadores. Esta mejora está dada especialmente por los controles de servidor incluidos en ASP.NET. Cuando un control es procesado, este automáticamente chequea el tipo de navegador que lo está ejecutando, generando una página adecuada para ese navegador.

Código Compilado

El código es compilado **just-in-time**, logrando un enorme aumento en el rendimiento, a través de soporte nativo y servicios de caché.

Controles de Servidor

Uno de los aspectos más importantes dentro del .NET Framework es su librería de clases. Esta librería es común en toda la plataforma .NET, lo que le brinda al programador una herramienta ideal para crear aplicaciones, con un considerable ahorro de líneas de código.

Los controles de servidor están divididos en dos categorías: Controles Web y Controles HTML. Posiblemente sean los Controles Web, los más atractivos para el desarrollador, ya que permiten crear automáticamente controles que realicen tareas importantes en el servidor como validar la entrada de formularios, verificar las capacidades de los navegadores o implementar un sistema de banners rotativos.

Introducción a ASP.NET. (s.f). Extraído el 10 de Octubre del 2010 desde http://antiguo.itson.mx/dii/mrodriguez/Optativa2/Sesiones/2/1.-Introduccion_a ASP.NET.pdf

C Sharp (C#)

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

El nombre C Sharp fue inspirado por la notación musical, donde # (sostenido, en inglés *sharp*) indica que la nota (C es la nota do en inglés) es el semitono más alto, sugiriendo que C# es superior a C/C++.

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (API), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

C Sharp. (s.f). Extraído el 10 de octubre del 2010 desde: http://es.wikipedia.org/wiki/C_Sharp

Las ventajas que ofrece C# frente a otros lenguajes de programación son:

Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.

Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.

Atributos: cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.

Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.

Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.

Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.

Inicializador: un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.

Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

Programación I. (s.f). Extraído el 10 de octubre del 2010 desde:

<http://programacion1abundiz.blogspot.com/2009/09/ventajas-del-c-y-desventajas.html>

Mono

Es una plataforma de software diseñada para permitir a los desarrolladores crear fácilmente aplicaciones de plataforma cruzada. Se trata de una implementación de código abierto. Net Framework de Microsoft basado en el estándar ECMA estándares para C # y el Common Language Runtime.

Ventajas

Hay muchos beneficios para la elección de Mono para el desarrollo de aplicaciones:

Popularidad: Construido sobre el éxito de la Red, hay millones de desarrolladores que tienen experiencia en la creación de aplicaciones en C #. También hay decenas de miles de libros, sitios web, tutoriales y ejemplos de código fuente para ayudar con cualquier problema imaginable.

Base Class Library: Tener una biblioteca de clases a completa disposición de miles, de construir en las clases para aumentar la productividad.

Plataforma cruzada: Mono está diseñado para ser multiplataforma. Mono se ejecuta en Linux, Microsoft Windows, Mac OS X, BSD y Sun Solaris, Nintendo Wii, Sony PlayStation 3, el iPhone de Apple.

What is Mono. (s.f). Extraído el 10 de octubre del 2010 desde: http://www.mono-project.com/What_is_Mono

Mono utiliza como IDE (Entorno de Desarrollo Integrado) a MonoDevelop para la creación de aplicaciones.

¿Mono soporta ASP.NET?

Sí. Mono soporta ASP.NET 1.1 y ASP.NET 2.0

A continuación se muestra una comparativa entre ASP.NET Y PHP:

Tabla comparativa entre ASP.NET Y PHP

	ASP.NET (versión 2.0)	PHP (versión 5.3.3)
Código compilado	El código es compilado just-in-time , logrando un enorme aumento en el rendimiento.	No lo soporta El código es interpretado
Code-behind	La página de desarrollo se compone de dos ficheros: el de la interfaz de usuario y el de código.	No lo soporta Tanto el código html, java script y php constan en el mismo fichero.
Documentación	Existe gran cantidad de información sobre ASP.NET en la web, libros, artículos, foros en el que se puede fácilmente investigar.	Posee una amplia documentación en la web.
Soporte a SGBD	MySql, PostgreSQL, Oracle, SQL Server, etc.	MySql, PostgreSQL, Oracle, SQL Server, etc.

IDE	Visual Studio, MonoDevelop, Visual Web Developer Express, etc.	Netbeans, PhpStorm, Eclipse, etc.
Multipataforma	Si lo soporta a través de Mono	Fue diseñado para ser multipataforma.
Soporte a lenguajes	ASP.NET soporta la programación en lenguajes potentes cómo, VisualBasic.Net (VB), C# y J#	Php

Tabla 2.3. *Comparativa entre ASP.NET Y PHP*

Por las características y por ser una tecnología conocida por el investigador se escogió la versión libre de ASP.NET, la misma que pertenece al **proyecto mono**, con la finalidad de que la aplicación desarrollada sea compatible con el S.O. en la cual será implantada (CentOS 5.4).

2.3.2 Categorías fundamentales de la Variable Dependiente:

Manejo adecuado de la información en la UOCENI - ISEI

2.3.2.1 Gestión de proyectos

La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos. Un proyecto es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único

Gestión de proyectos. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos

Técnicas y Herramientas de Gestión de Proyectos

Hay una grand variedad de técnicas utilizadas en el planteamiento y control de proyecto, a saber, los Gráficos de Gantt (incluyendo cronogramas, gráficos de carga, entre otros), el PERT (Programe Evaluation and Review Technique) y el CPM (Critical Path Method). A nivel informático también han sido desarrolladas varias aplicaciones de apoyo al proceso de gestión de proyectos.

Gestión de proyectos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.deltaasesores.com/articulos/gestion-de-proyectos>

2.3.2.2 Automatización

Es el sistema de fabricación diseñado con el fin de usar la capacidad de las máquinas para llevar a cabo determinadas tareas anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana.

"Automatización." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008.

Uso de la informática

La informática es tan específica que es posible desarrollar o montar paquetes de software tan fácilmente como la imaginación lo permita. La gran dificultad es identificar correctamente las necesidades, independientemente de cualquier solución técnica, y seleccionar un proveedor o un equipo de desarrollo interno para llevarlo a cabo.

JEFF. (2008). Introducción a la gestión de proyectos. Extraído el 10 de Octubre del 2010 desde <http://es.kioskea.net/contents/projet/projetintro.php3>

2.3.2.3 Software de gestión de proyectos

El software de gestión de proyectos, es un término que cubre muchos tipos de software, incluso programación, asignación de recursos, software de colaboración, comunicación y sistemas de documentación, que están acostumbrados al trato con la complejidad de proyectos grandes.

Las herramientas de software son requeridas para automatizar y facilitar la aplicación de la metodología particular de la organización para la administración de proyectos. Esta metodología incluye cómo se organiza para manejar sus proyectos, qué prácticas son necesarias para llevar a cabo su administración y sus

procesos, y además que requerimientos se tienen en relación con su cultura organizacional.

Una vez que se determinan y documentan estas necesidades, es posible evaluar que aplicaciones de software cumplen con dichos criterios y en base a esta evaluación realizar una selección.

La nueva generación de herramientas de Tecnologías de Información de administración de proyectos combinan las tres S's: *scope*, *scheduling* y *status*. Es decir, herramientas para administrar el alcance, la programación de tareas y el estado en que se encuentran.

CABALLERO, Omar. (2006). Tecnologías de información y herramientas para la administración de proyectos de software. Extraído el 10 de Octubre del 2010 desde <http://www.revista.unam.mx/vol.7/num6/int47/art47.htm>

2.4 Hipótesis

El desarrollo de un Sistema de Gestión de Proyectos permitirá un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

2.5 Variables

2.5.1 Variable Independiente.- Sistema de gestión de proyectos.

2.5.2 Variable Dependiente.- Manejo adecuado de la información en la UOCENI - ISEI.

CAPÍTULO III METODOLOGÍA

3.1 Enfoque

El presente trabajo de investigación está enmarcado dentro del paradigma crítico propositivo, por tanto tiene un enfoque cuali-cuantitativo ya que se trabaja con sentidos holísticos y participativo considerando una realidad dinámica, pero al mismo tiempo está orientada a la comprobación de hipótesis y con énfasis en los resultados.

3.2 Modalidad de la investigación

En el desarrollo del proceso investigativo se empleó la información bibliográfica para la elaboración del marco teórico y la investigación de campo para la recolección de información, lo cual fue de gran ayuda para obtener elementos de juicio necesarios para la investigación y sirvió de base para la elaboración de la propuesta.

3.3 Niveles de investigación

La investigación abarca desde el nivel exploratorio hasta el nivel explicativo pues se reconocieron las variables que competen al problema, se establecieron las características de la realidad a investigarse, las causas y consecuencias del problema y se llegó a la comprensión de la hipótesis. Además la investigación pasa por un nivel correlacional porque se determinó el grado de relación que existe entre las variables que corresponden al problema.

3.4 Población y muestra

El presente trabajo de investigación se lo realizó en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial con una población de 6 personas las cuales están directamente involucradas con el proceso de gestión de proyectos.

Se trabajó con todo el universo investigativo considerando que la muestra es pequeña.

3.5 Operacionalización de variables

3.5.1 Variable independiente: Sistema de gestión de proyectos.

Conceptualización	Categoría	Indicadores	Ítems	Instrumento
<p>Sistema de gestión de proyectos: Es un sistema informático que permite el almacenamiento, control y seguimiento de todo lo que conlleva el desarrollo de proyectos.</p>	<p>Procedimientos</p>	<p>Administración</p>	<p>¿De qué manera se maneja la gestión de proyectos en la UOCENI -ISEI?</p> <p><input type="checkbox"/> Manualmente</p> <p><input type="checkbox"/> Sistema Informático</p> <p><input type="checkbox"/> Excel</p> <p><input type="checkbox"/> Otro programa</p>	<p>Cuestionario</p>
		<p>Control</p>	<p>¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite un control adecuado del proceso de desarrollo de los proyectos?</p> <p><input type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p>	<p>Cuestionario</p>

		<p>Reportes</p>	<p>¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite la generación de reportes oportunos?</p> <p><input type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p>	<p>Cuestionario</p>
		<p>Tecnología</p>	<p>¿Cree Ud. que sería conveniente poner en marcha un plan de renovación tecnológica para mejorar y facilitar la administración de los proyectos?</p> <p><input type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p>	<p>Cuestionario</p>

Tabla 3.1. Operacionalización de variable independiente

3.5.2 Variable dependiente: Manejo adecuado de la información en la UOCENI-ISEI.

Conceptualización	Categoría	Indicadores	Ítems	Instrumento
<p>Manejo adecuado de la información en la UOCENI - ISEI: En cualquier actividad productiva en la que estemos inmersos, día a día debemos tomar decisiones que indicarán el rumbo de nuestra empresa. Pero para tomar una decisión lo más acertada posible es necesario basarnos en información de calidad, de ahí la importancia del cómo y con qué herramientas estamos manejando la información.</p>	<p>Información</p>	<p>Almacenamiento</p>	<p>¿Cómo está almacenada la información de los proyectos actualmente?</p> <p><input type="checkbox"/> Base de Datos Manual</p> <p><input type="checkbox"/> Base de Datos Automatizada</p> <p>¿La información almacenada de los proyectos es confiable?</p> <p><input type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p> <p>¿Cree usted que toda la información almacenada de los proyectos es necesaria?</p> <p><input type="checkbox"/> Si</p> <p><input type="checkbox"/> No</p>	<p>Cuestionario</p> <p>Cuestionario</p> <p>Cuestionario</p>

		Acceso	<p>¿La generación y obtención de información acerca de los avances de cada proyecto es?</p> <p><input type="checkbox"/> Buena</p> <p><input type="checkbox"/> Aceptable</p> <p><input type="checkbox"/> Regular</p> <p><input type="checkbox"/> Pésima</p>	Cuestionario
	Recursos	Recursos humanos	<p>¿El seguimiento y control del personal que trabaja en los proyectos es?</p> <p><input type="checkbox"/> Muy bueno</p> <p><input type="checkbox"/> Aceptable</p> <p><input type="checkbox"/> Regular</p> <p><input type="checkbox"/> Pésimo</p>	Cuestionario

Tabla 3.2. Operacionalización de variable dependiente

3.6 Técnicas e instrumentos de investigación

Las técnicas que se emplearon en el proceso de la investigación son la encuesta, y la observación.

La encuesta se empleo para obtener datos significativos referentes a la información de la gestión de los proyectos que maneja la institución, para lo cual se estructuró un cuestionario que fue un instrumento que permitió obtener los datos requeridos de primera fuente.

La observación fue de gran valor en la apreciación directa de la realidad, circunstancias que permitieron confrontar los hechos e imprimir un sello de transparencia e imparcialidad a la investigación.

3.7 Procesamiento de la información

Una vez aplicados los instrumentos de investigación y analizada la validez, se procedió al análisis de los datos obtenidos los cuales fueron parte modular para la propuesta.

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Análisis e interpretación de resultados

Para la realización del análisis e interpretación de resultados es importante recalcar que las encuestas se realizaron a 6 personas que están directamente involucradas con el proceso de gestión de proyectos en la UOCENI – ISEI, como se menciono anteriormente en el capítulo III.

4.1.1 Análisis de los resultados de las encuestas

Pregunta N° 01

¿De qué manera se maneja la gestión de proyectos en la UOCENI - ISEI?

RESPUESTA	CANTIDAD	PORCENTAJE
Manualmente	6	85,7
Sistema informático	0	0
Excel	1	14,3
Otro programa	0	0
Total	7	100

TABLA 4.1. Cuadro porcentual pregunta 1

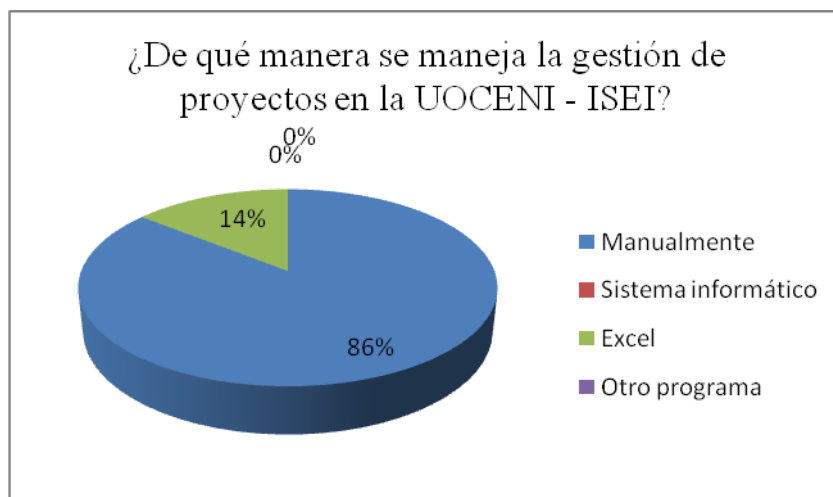


Figura 4.1. Pregunta 1

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 86% de las personas encuestadas respondieron que manejan la información de los proyectos manualmente y el 14% contestaron que manejan la información mediante Excel.

Análisis: Se demuestra que el proceso de gestión de proyectos casi en su totalidad se lo está realizando manualmente.

Pregunta N° 02

¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite un control adecuado del proceso de desarrollo de los proyectos?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	1	16,7
No	5	83,3
Total	6	100,0

Tabla 4.2. Tabla porcentual pregunta 2

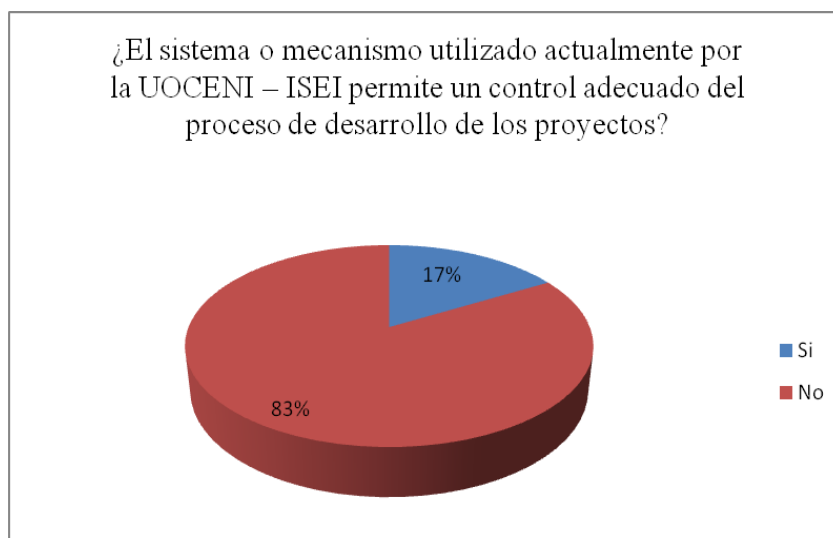


Figura 4.2. *Pregunta2*

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 83 % de las personas encuestadas respondió que el sistema o mecanismo actual utilizado en la UOCENI – ISEI no permite un control adecuado del proceso de desarrollo de los proyectos y el 17% respondió que si lo permite.

Análisis: Se concluye que la técnica de control del proceso de desarrollo de los proyectos no está realizando un buen trabajo.

Pregunta N° 03

¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite la generación de reportes oportunos?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	1	16,7
No	5	83,3
Total	6	100,0

Tabla 4.3. *Tabla porcentual pregunta 3*

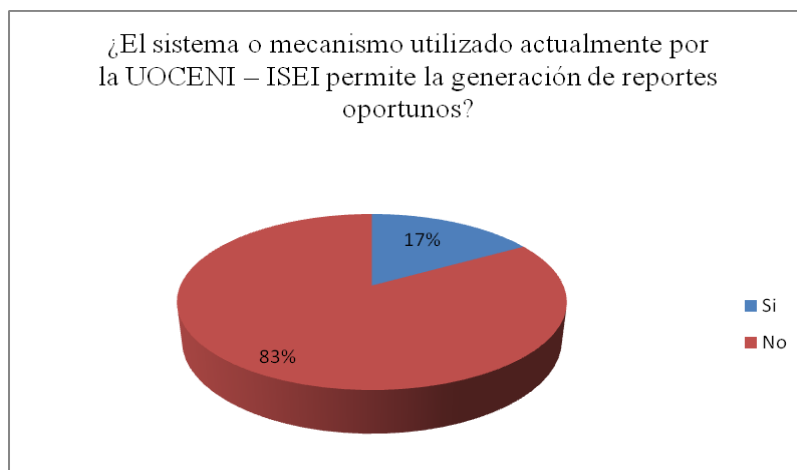


Figura 4.3. Pregunta 3

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 83% de las personas encuestadas respondió que el sistema o mecanismo utilizado actualmente en la UOCENI – ISEI no permite la generación de reportes oportunos y el 17 % contestaron que si lo permite.

Análisis: La mayoría de las personas encuestadas manifiesta que el mecanismo actual de gestión de proyectos no genera reportes oportunos.

Pregunta N° 04

¿Cree Ud. que sería conveniente poner en marcha un plan de renovación tecnológica para mejorar y facilitar la administración de los proyectos?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	5	83,3
No	1	16,7
Total	6	100,0

Tabla 4.4. Tabla porcentual pregunta 4

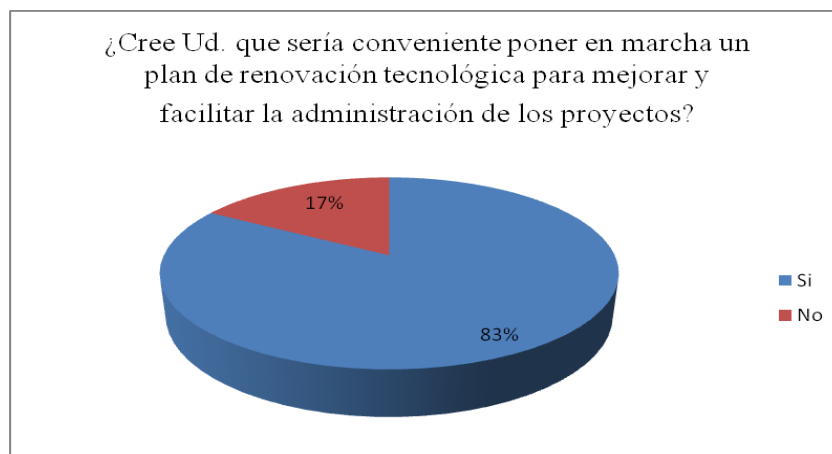


Figura 4.4. *Pregunta 4*

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 83% de las personas encuestadas respondió que sí sería conveniente poner en marcha un plan de renovación tecnológica para mejorar y facilitar la administración de los proyectos y el 17% contestó que no sería conveniente poner en marcha un plan de renovación tecnológica.

Análisis: La mayoría de las personas encuestadas manifiesta que sí sería conveniente poner en marcha un plan de renovación tecnológica para mejorar y facilitar la administración de los proyectos.

Pregunta N° 05

¿Cómo esta almacenada la información de los proyectos actualmente?

RESPUESTA	CANTIDAD	PORCENTAJE
Base de datos manual	6	100,0
Base de datos automatizada	0	0,0
Total	6	100,0

Tabla 4.5. *Tabla porcentual pregunta 5*

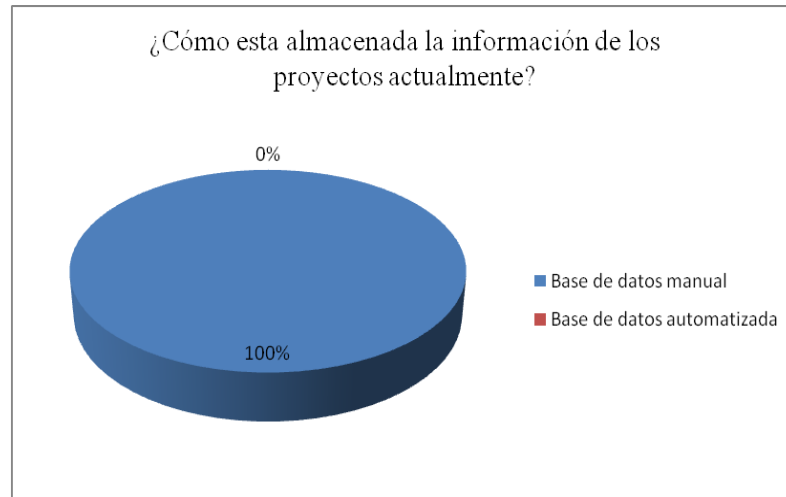


Figura 4.5. Pregunta 5

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 100% de las personas encuestadas respondió que la información de los proyectos se encuentra almacenada en una base de datos manual.

Análisis: La totalidad de las personas encuestadas manifiesta que la información de los proyectos esta almacenada en una base de datos manual.

Pregunta N° 06

¿La información almacenada de los proyectos es confiable?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	3	50,0
No	3	50,0
Total	6	100,0

Tabla 4.6. Tabla porcentual pregunta 6

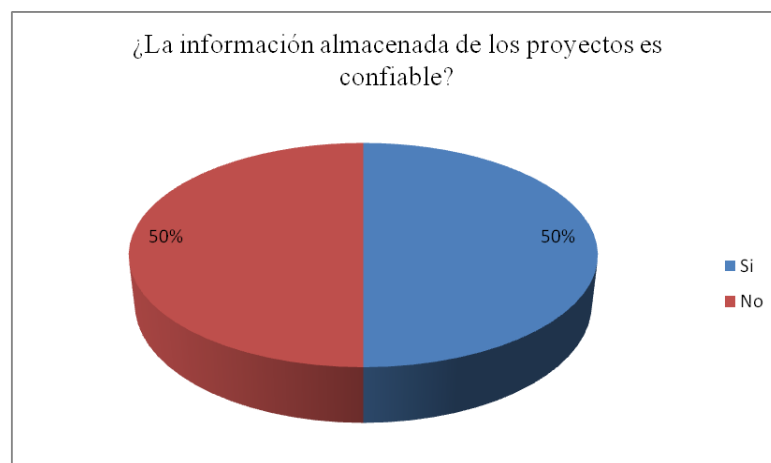


Figura 4.6. *Pregunta 6*

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 50% de las personas encuestadas respondió que la información almacenada de los proyectos sí es confiable y el otro 50% contestó que no lo es.

Análisis: Las personas encuestadas demuestran tener una opinión dividida, por lo que se tendrá que realizar un estudio técnico para verificar la confiabilidad de la información.

Pregunta N° 07

¿Cree usted que toda la información almacenada de los proyectos es necesaria?

RESPUESTA	CANTIDAD	PORCENTAJE
Si	6	100,0
No	0	0,0
Total	6	100,0

Tabla 4.7. *Tabla porcentual pregunta 7*

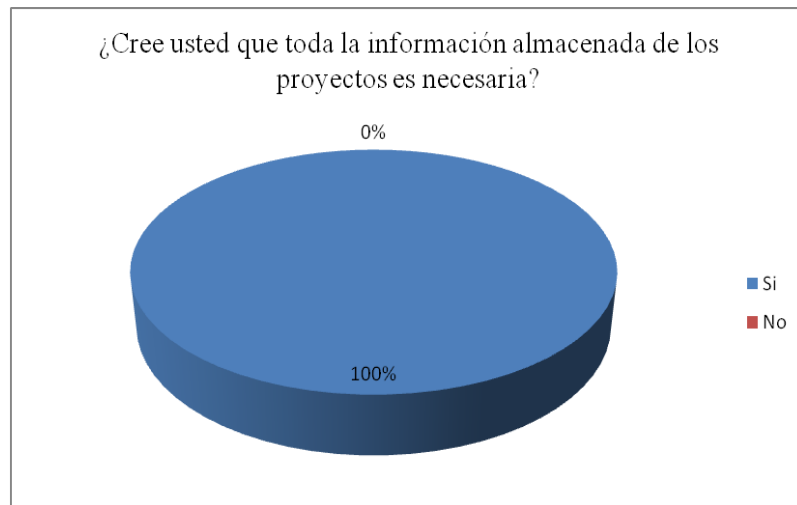


Figura 4.7. *Pregunta 7*

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 100% de las personas encuestadas respondió que toda la información almacenada de los proyectos es necesaria.

Análisis: La totalidad de las personas encuestadas manifiestan que toda la información almacenada de los proyectos es necesaria.

Pregunta N° 08

¿La generación y obtención de información acerca de los avances de cada proyecto es?

RESPUESTA	CANTIDAD	PORCENTAJE
Buena	0	0,0
Aceptable	4	66,7
Regular	2	33,3
Pésima	0	0,0
Total	6	100,0

Tabla 4.8. *Tabla porcentual pregunta 8*

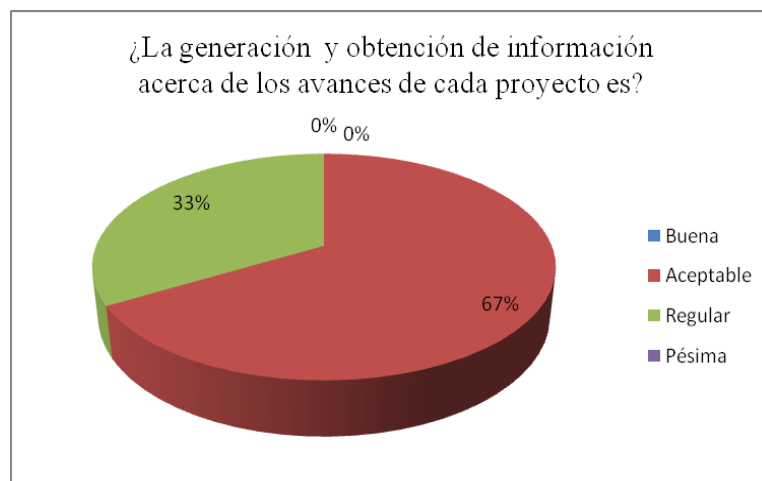


Figura 4.8. Pregunta 8

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 67% de las personas encuestadas respondió que la generación y obtención de información acerca de los avances de cada proyecto es Aceptable y el 33% contestó que es regular.

Análisis: La mayoría de personas encuestadas manifiestan que la generación y obtención de información acerca de los avances de cada proyecto es Aceptable

Pregunta N° 09

¿El seguimiento y control del personal que trabaja en los proyectos es?

RESPUESTA	CANTIDAD	PORCENTAJE
Muy Bueno	0	0,0
Aceptable	2	33,3
Regular	3	50,0
Pésimo	1	16,7
Total	6	100,0

Tabla 4.9. Tabla porcentual pregunta 9

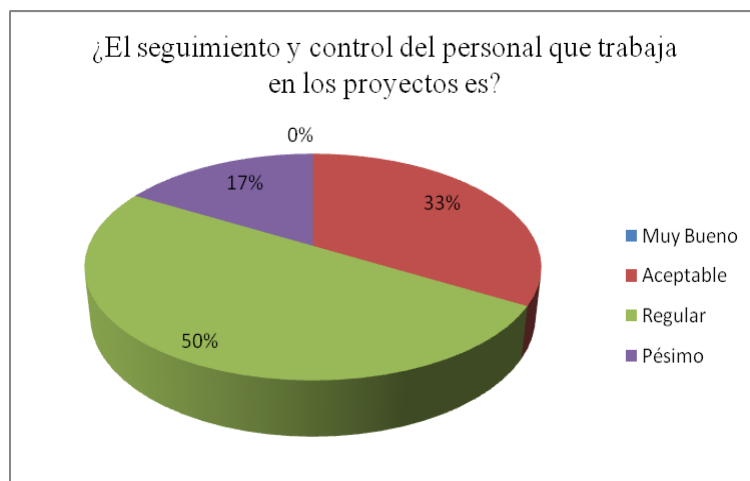


Figura 4.9. *Pregunta 9*

Desarrollado por: Alex Paucar

Interpretación: La gráfica refleja que el 50% de las personas encuestadas respondió que el seguimiento y control del personal que trabaja en los proyectos es regular; el 33% contestó que es Aceptable y el 17% contestó que es pésimo.

Análisis: La mayor parte de las personas encuestadas manifestó que el seguimiento y control del personal es regular.

CONCLUSIÓN

Una vez procesados y analizados los datos resultantes de las encuestas efectuadas a las personas que trabajan en la UOCENI – ISEI se llegó a determinar que la forma actual de manejo de los proyectos no brinda confianza y dificulta la administración de los mismos.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La forma como se está manejando la gestión de proyectos en la UOCENI – ISEI es manual, esta forma no permite un control adecuado del proceso de desarrollo de los proyectos y carece de generación oportuna de reportes.
- La información referente a los proyectos se encuentra almacenada en una Base de Datos manual; y según la encuesta realizada al personal que integra el departamento dicha información en su totalidad es necesaria, pero existe una opinión dividida sobre si ésta es confiable.
- El personal que integra el departamento UOCENI – ISEI ha dado a conocer su inconformidad por la forma actual de manejo de los proyectos y creen conveniente poner en marcha un plan de renovación tecnológica con la finalidad de mejorar y facilitar la administración de los mismos.
- Como IDE (Entorno de Desarrollo Integrado) se escogió a MonoDevelop, herramienta de código abierto que permite trabajar con tecnología también de código abierto como ASP.NET de Mono, las mismas que por sus características permitirán el desarrollo de un software potente y de alta calidad.

- Como Sistema Gestor de Base de Datos (SGBD) se escogió a PostgreSQL debido a su alto desempeño en el manejo de grandes cantidades de información, factor que garantizará una mayor estabilidad, integridad y seguridad en el sistema.

5.2 Recomendaciones

- Se recomienda desarrollar e implantar el sistema informático en el departamento UOCENI – ISEI como solución al problema investigado.
- Se recomienda diseñar una base de datos relacional que brinde seguridad y confianza en el almacenamiento de información correspondiente a la gestión de proyectos.
- Se recomienda incluir en el sistema mecanismos de filtrado y generación de reportes según la información requerida por el usuario.
- Las herramientas en las cuales se pretende desarrollar el sistema, se han establecido según las necesidades y requerimientos del departamento.
- Se recomienda revisar exhaustivamente las ayudas y los manuales de cada una de las herramientas escogidas para el desarrollo del sistema, con el fin de aprovechar y aplicar todas las bondades que éstas nos ofrecen.

CAPITULO VI

PROPUESTA

6.1 Tema

Sistema de gestión de proyectos, para un manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

6.2 Datos informativos

Institución: Departamento UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Ciudad: Ambato

Dirección: Ciudadela universitaria (predios Huachi) de la Universidad Técnica de Ambato.

Investigador: Alex Ricardo Paucar Medina

Tutor: Ing. David Guevara

6.3 Antecedentes

La práctica investigativa en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial se inicia formalmente a partir del año 2003, con la creación de un departamento destinado a fomentar y desarrollar proyectos de investigación denominado UOCENI - ISEI (Unidad Operativa del Centro de investigaciones de la FISEI), el mismo que viene desarrollando sus actividades desde entonces.

El almacenamiento, control y seguimiento de los proyectos en la UOCENI - ISEI siempre se los ha realizado manualmente creando u originando una serie de inconvenientes como: información inexacta, procesos tediosos y repetitivos, un inadecuado control de recursos requeridos para el desarrollo de los proyectos, etc. Partiendo de estos inconvenientes surge la necesidad de desarrollar un sistema automatizado de gestión de proyectos que cumpla con los requerimientos de la UOCENI – ISEI y contribuya con el desarrollo de la misma.

Teniendo en cuenta la exigencia del departamento de que la información deba ser segura, fiable, precisa; se decidió escoger como Sistema Gestor de Base de Datos a PostgreSQL que es un motor que cumple con las características. Para la parte de desarrollo web se escogió como IDE (Entorno de desarrollo Integrado) a monodevelop, herramienta de código abierto (OPEN SOURCE) que permite trabajar con tecnología ASP.NET de Mono.

6.4 Justificación

El mecanismo utilizado para la administración de los proyectos en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial no entrega los resultados esperados, por esta razón se decidió aprovechar los beneficios de la tecnología y desarrollar un sistema informático que a través de la optimización de procesos, recuperación rápida y eficiente de la información, seguridad en los datos, permita dar solución al problema.

Además, la elaboración de la presente propuesta es factible y viable para utilizarla como instrumento de cambio y mejoramiento para la entidad, a la vez que permitirá al investigador aplicar conocimientos adquiridos en el aprendizaje de toda la carrera universitaria.

6.5 Objetivos

6.5.1 Objetivo general

- Desarrollar e implantar un Sistema de Gestión de Proyectos utilizando herramientas de código abierto, para el manejo adecuado de la información en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

6.5.2 Objetivos específicos

- Analizar los requerimientos obtenidos por el personal que labora en la UOCENI – ISEI para definir el funcionamiento que tendrá el Sistema de Gestión de Proyectos.
- Desarrollar e implementar el Sistema de Gestión de Proyectos en base a los requerimientos de la UOCENI – ISEI con la tecnología ASP.NET de Mono.
- Implantar el Sistema de Gestión de Proyectos en el servidor del departamento UOCENI - ISEI.

6.6 Análisis de factibilidad

6.6.1 Factibilidad operativa

Gracias a que el sistema informático fue realizado a medida, los usuarios no necesitarán de una capacitación extensa para utilizarlo, además que éste cuenta con interfaces sencillas y amigables para su manipulación, a más de permitir una gerencia total del sistema mediante el menú de administración.

El sistema permite la creación de roles y usuarios según el administrador lo requiera, cada usuario es asignado a un role, dependiendo de la personalización del role se podrá añadir, modificar, eliminar, filtrar y generar reportes; necesariamente el usuario debe loguearse para poder realizar cualquier actividad en el sistema informático.

6.6.2 Factibilidad económica

El proyecto es factible económicamente debido a que para su desarrollo se ha utilizado herramientas de código abierto (OPEN SOURCE), las mismas que se caracterizan por su bajo costo de implementación y mantenimiento.

6.6.3 Factibilidad técnica

Software

Para la construcción del sistema informático se adquirió:

- Herramienta de desarrollo web: MonoDevelop con tecnología ASP.NET de Mono
- Motor de Base de Datos: PostgreSQL
- Servidor web: apache
- Sistema Operativo: Centos 5.4

Hardware

El departamento UOCENI – ISEI al ser un centro de investigación y desarrollo de proyectos informáticos cuenta con el equipo necesario para la implantación del sistema, aspecto que beneficiará en el desarrollo del mismo.

6.7 Fundamentación

6.7.1 Asp.net

Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Un framework puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

6.7.2 MonoDevelop

Es un IDE diseñado principalmente para C # y otros lenguajes .NET.

MonoDevelop permite a los desarrolladores crear rápidamente aplicaciones de escritorio y Web ASP.NET en Linux, Windows y Mac OSX con soporte completo de finalización de código y prueba en XSP, el servidor web Mono.

6.7.3 XSP

Es un servidor web sencillo que ejecuta de forma independiente código en C#. El mismo permite hospedar en Linux y otros sistemas operativos UNIX sitios desarrollados utilizando ASP.NET. Además de ejecutarse sobre la plataforma Mono para Linux también permite su ejecución sobre la plataforma .NET, posibilitando que sea utilizado como un servidor web ligero en cualquier plataforma que soporte .NET.

XSP fue originalmente el nombre del proyecto de Microsoft que luego se transformó en ASP.NET. El nombre es un homenaje al nombre original de lo que se transformó en ASP.NET.

6.7.4 JavaScript

Es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML. Al ser la más sencilla, es por el momento la más extendida.

JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor. Un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente, es decir, en el cliente, los eventos generados por el usuario. Estos eventos pueden ser el recorrido del propio documento HTML o la gestión de un formulario.

Características

- Es interpretado (no compilado) por el cliente, es decir, directamente del programa fuente se pasa a la ejecución de dicho programa, con lo que al contrario que los lenguajes compilados no se genera ni código objeto ni ejecutable para cada máquina en el que se quiera ejecutar dicho programa.
- Su código se integra en las páginas HTML, incluido en las propias páginas.
- No es necesario declarar los tipos de variables que van a utilizarse ya que como se verá más adelante, JavaScript realiza una conversión automática de tipos.
- Las referencias a objetos se comprueban en tiempo de ejecución. Esto es consecuencia de que JavaScript no es un lenguaje compilado.

6.8 Metodología

Para la realización del sistema de gestión de proyectos se ha decidido utilizar la metodología de desarrollo de software denominada **modelo en cascada**, paradigma que consiste en el desarrollo del software en fases, las mismas que se deben ejecutar en forma secuencial, es decir una después de la otra, de tal forma que el inicio de cada etapa deba esperar a la finalización de la anterior;

permitiendo alcanzar el objetivo propuesto y un correcto trabajo en cada fase de desarrollo.

Ventajas de la Metodología Modelo de Cascada:

- Modelo y planificación fácil y sencillos
- La calidad del producto resultante es alta.
- Produce mejores resultados en proyectos donde los requerimientos del software son claramente entendidos.

Estructura del modelo en cascada

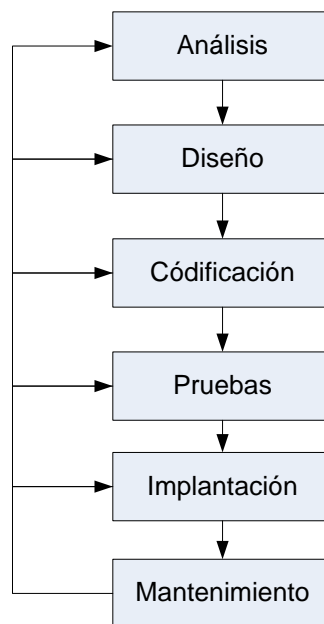


Figura 6.1. *Estructura del modelo en cascada*

Una de las mejores herramientas para especificar, visualizar, construir y documentar un sistema de software y que además es independiente de la metodología de desarrollo que se esté utilizado, es UML (Lenguaje unificado de modelado), motivo por el cual se ha escogido utilizarlo como ayuda en el análisis y diseño del sistema.

6.9 Modelo operativo

6.9.1 Análisis del sistema

6.9.1.1 Análisis y requerimientos del sistema

El Departamento UOCENI - ISEI de la facultad de Ingeniería en Sistemas, Electrónica e Industria está teniendo dificultades en la administración de los proyectos que ejecuta, motivo por el cual se desarrollará un software a medida que facilite dicho proceso.

Luego de haber analizado los requerimientos que solicita el departamento se ha determinado que el sistema deberá contar con:

- Un módulo para el ingreso, modificación, eliminación y control de: objetivos, tareas, archivos y actividades que pertenecen al proyecto. Recordando que un proyecto consta de varios objetivos y un objetivo consta de varias tareas; además para dar por concluido un proyecto sus objetivos deben estar finalizados, de igual manera para terminar un objetivo sus tareas deben estar finalizadas.
- Un módulo que gestione el presupuesto de cada proyecto, mediante el cual se garantice un correcto consumo y asignación de recursos.
- Un módulo que administre el personal que trabaja en cada proyecto.
- Un patrón que facilite la medición y avance de cada uno de los proyectos.
- Un módulo de seguridad que permita el acceso a la información solo a personas autorizadas.
- Generación de reportes de: registro de actividades diarias y mensuales de los usuarios, información personal de cada usuario y en los proyectos que se encuentra asignado.

- Además, el sistema deberá contar con interfaces sencillas y amigables para su manipulación.

Para el desarrollo del sistema se utilizará herramientas de código abierto (OPEN SOURCE).

6.9.1.2 Diagramas UML

Considerando que hoy en día, UML (Lenguaje unificado de modelado) está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo, se ha procedido a utilizarlo con el fin de establecer requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código.

6.9.1.2.1 Diagrama de casos de uso

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

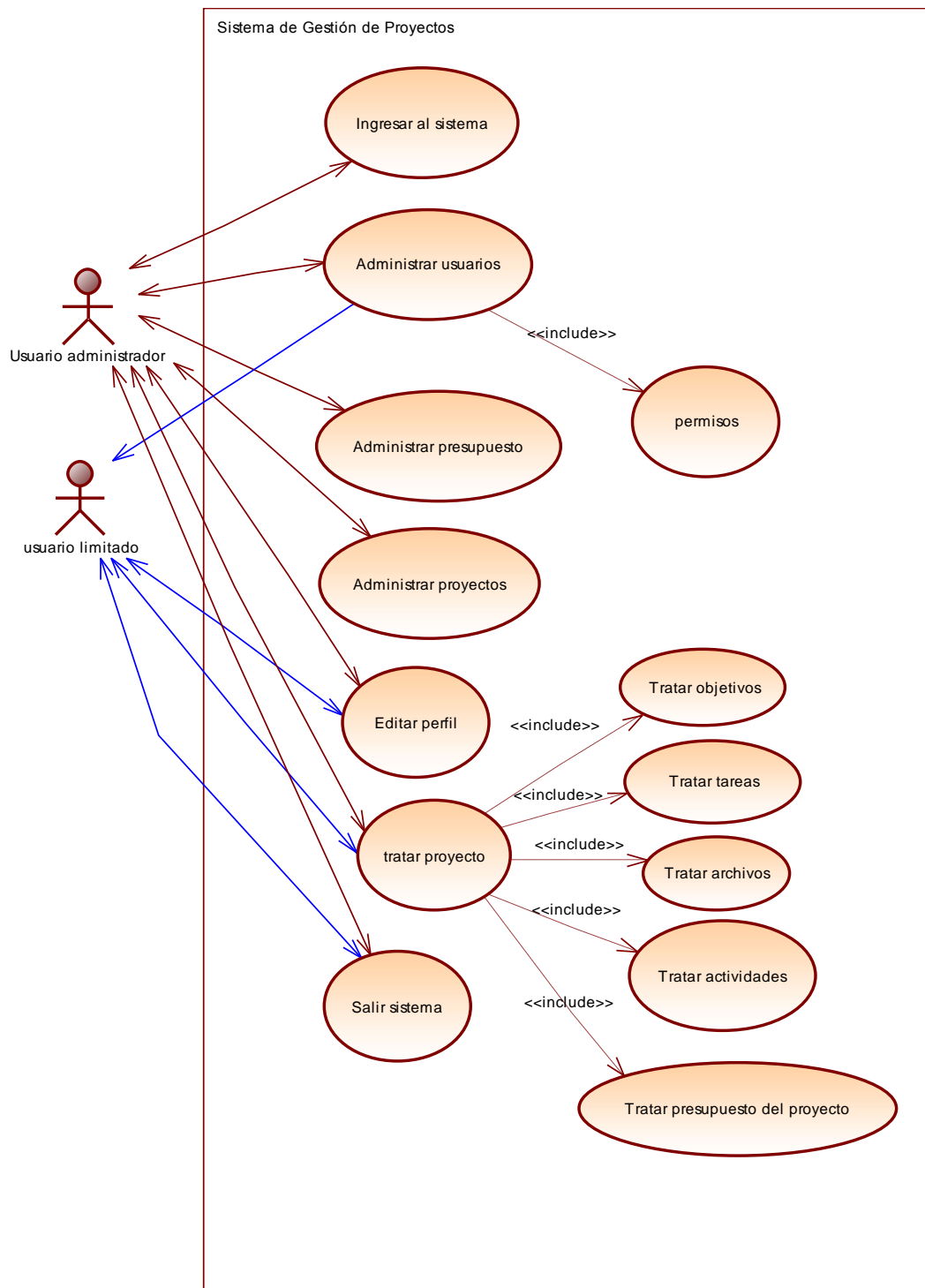


Figura 6.2. Diagrama de casos de uso Sistema de Gestión de Proyectos

Especificación de casos de uso

	Caso de uso: Ingresar al sistema
Actores:	Usuario administrador Usuario limitado
Descripción:	Validar a un usuario ya registrado para el uso del sistema.
Precondiciones:	El usuario debe estar registrado en el sistema
Pos condiciones	Validación realizada con éxito
Flujo normal de eventos:	validar <ol style="list-style-type: none"> 1. El usuario ingresa el nombre de usuario y contraseña 2. El sistema valida los datos ingresados por el usuario. 3. Una vez validado el usuario, el sistema muestra el menú de opciones
Flujos alternos:	usuario no registrado <ol style="list-style-type: none"> 1. En el paso 2 del Flujo Normal, si el usuario no existe en el sistema se muestra un mensaje donde se indica que los datos de usuario o contraseña son erróneos.

Tabla 6.1. *Especificación ingresar al sistema*

	Caso de uso: Administrar usuarios
Actores:	Usuario administrador
Descripción:	Añadir, modificar, eliminar usuarios del sistema.
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado como administrador • Datos de permisos ingresados con anterioridad
Pos condiciones	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	añadir <ol style="list-style-type: none"> 1. El usuario escoge la opción de Administrar usuarios 2. El sistema muestra las opciones disponibles: añadir, modificar, eliminar 3. El usuario escoge la opción añadir 4. El sistema muestra los campos para ingresar la información de los usuarios 5. El usuario ingresa la información 6. El sistema verifica el ingreso correcto de información

Flujos alternos:	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario selecciona la opción modificar 2. El sistema muestra los campos con información del usuario a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada.
-------------------------	---

Tabla 6.2. Especificación administrar usuarios

	Caso de uso: Permisos
Actores:	Usuario administrador
Descripción:	Añadir, modificar, eliminar permisos
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado como administrador.
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción de permisos 2. El sistema muestra las opciones disponibles: añadir, modificar, eliminar 3. El usuario escoge la opción añadir 4. El sistema muestra los campos para ingresar la información de los permisos 5. El usuario ingresa la información 6. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 5. En el paso 3 del Flujo Normal, el usuario escoge la opción modificar 6. El sistema muestra los campos con información del permiso a modificar. 7. El usuario modifica la información 8. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 5. En el paso 3 del Flujo Normal, el usuario escoge la opción eliminar

	<ol style="list-style-type: none"> 6. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 7. El usuario toma una decisión 8. El sistema realiza la operación indicada.
--	--

Tabla 6.3. *Especificación permisos*

	Caso de uso: Administrar presupuesto
Actores:	Usuario administrador
Descripción:	Añadir, modificar, eliminar presupuesto
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado como administrador.
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>Añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción de administrar presupuesto. 2. El sistema muestra las opciones disponibles: añadir, modificar, eliminar. 3. El usuario escoge la opción añadir. 4. El sistema muestra los campos para ingresar la información del presupuesto. 5. El usuario ingresa la información. 6. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del presupuesto a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada.

Tabla 6.4. *Especificación administrar presupuesto*

	Caso de uso: Administrar proyectos
Actores:	Usuario administrador
Descripción:	Añadir, modificar, eliminar, cerrar y asignar proyectos
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado como administrador. • Datos de usuarios ingresados con anterioridad
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción de administrar proyectos 2. El sistema muestra las opciones disponibles: añadir, modificar, eliminar, cerrar, usuarios del proyecto 3. El usuario escoge la opción añadir 4. El sistema muestra los campos para ingresar la información de los proyectos 5. El usuario ingresa la información 6. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del proyecto a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada. <p>cerrar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción cerrar 2. El sistema realiza la operación indicada <p>usuarios del proyecto</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción usuarios del proyecto 2. El sistema muestra un formulario con los usuarios registrados en el sistema

	<ol style="list-style-type: none"> 3. El usuario escoge los usuarios para el proyecto 4. El sistema verifica el ingreso correcto de información.
--	--

Tabla 6.5. *Especificación administrar proyectos*

	Caso de uso: editar perfil
Actores:	Usuario administrador Usuario limitado
Descripción:	Modificar perfil
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado.
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	añadir <ol style="list-style-type: none"> 1. El usuario escoge la opción mi perfil 2. El sistema muestra las opciones disponibles: editar, exportar 3. El usuario escoge la opción editar 4. El sistema muestra un formulario con información del usuario solicitante 5. El usuario modifica la información 6. El sistema verifica el ingreso correcto de información.
Flujos alternos	Exportar <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción exportar 2. El sistema genera un archivo .PDF con información del usuario y los proyectos en los cuales está trabajando.

Tabla 6.6. *Especificación editar perfil*

	Caso de uso: tratar proyecto
Actores:	Usuario administrador Usuario limitado
Descripción:	Añadir, modificar, eliminar y cerrar proyectos
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario con las opciones disponibles: añadir, modificar, eliminar, cerrar 3. El usuario escoge la opción añadir 4. El sistema muestra los campos para ingresar la información de los proyectos 5. El usuario ingresa la información 6. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>Modificar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del proyecto a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 3 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada. <p>cerrar</p> <ol style="list-style-type: none"> 3. En el paso 3 del Flujo Normal, el usuario escoge la opción cerrar 4. El sistema realiza la operación indicada

Tabla 6.7. Especificación tratar proyecto

	Caso de uso: tratar objetivos
Actores:	Usuario administrador Usuario limitado
Descripción:	Añadir, modificar, eliminar y cerrar objetivos
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios • El usuario debe estar asignado al proyecto contenedor del objetivo a tratar
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario 3. El usuario escoge un proyecto 4. El sistema muestra un menú con las opciones: objetivos, tareas, archivos, registro de actividad y presupuesto 5. El usuario escoge la opción objetivos 6. El sistema muestra la opciones disponibles: añadir, modificar, eliminar, cerrar 7. El usuario escoge la opción añadir 8. El sistema muestra los campos para ingresar la información de los objetivos 9. El usuario ingresa la información 10. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del objetivo a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información. <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada.

	<p>cerrar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción cerrar 2. El sistema realiza la operación indicada
--	--

Tabla 6.8. *Especificación tratar objetivos*

	Caso de uso: tratar tareas
Actores:	<p>Usuario administrador Usuario limitado</p>
Descripción:	Añadir, modificar, eliminar y cerrar tareas
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios • El usuario debe estar asignado al proyecto contenedor de la tarea a tratar • El proyecto contenedor de la tarea debe tener información de objetivos
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario 3. El usuario escoge un proyecto 4. El sistema muestra un menú con las opciones: objetivos, tareas, archivos, registro de actividad y presupuesto 5. El usuario escoge la opción tareas 6. El sistema muestra la opciones disponibles: añadir, modificar, eliminar, cerrar 7. El usuario escoge la opción añadir 8. El sistema muestra los campos para ingresar la información de las tareas 9. El usuario ingresa la información 10. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información de la tarea a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información

	<p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada. <p>cerrar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción cerrar 2. El sistema realiza la operación indicada
--	--

Tabla 6.9. Especificación tratar tareas

	Caso de uso: tratar archivos
Actores:	Usuario administrador Usuario limitado
Descripción:	Añadir, modificar, descargar y eliminar archivos
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios • El usuario debe estar asignado al proyecto contenedor del archivo a tratar
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario 3. El usuario escoge un proyecto 4. El sistema muestra un menú con las opciones: objetivos, tareas, archivos, registro de actividad y presupuesto 5. El usuario escoge la opción archivos 6. El sistema muestra la opciones disponibles: añadir, modificar, eliminar, descargar 7. El usuario escoge la opción añadir 8. El sistema muestra los campos para ingresar la información de los archivos 9. El usuario ingresa la información 10. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del archivo a modificar.

	<p>3. El usuario modifica la información</p> <p>4. El sistema verifica el ingreso correcto de información</p> <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada. <p>descargar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción descargar 2. El sistema permite la descarga del archivo.
--	---

Tabla 6.10. Especificación tratar archivos

	Caso de uso: tratar actividades
Actores:	Usuario administrador Usuario limitado
Descripción:	Añadir, modificar, eliminar, filtrar y exportar actividades
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios • El usuario debe estar asignado al proyecto contenedor de las actividades a tratar
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario 3. El usuario escoge un proyecto 4. El sistema muestra un menú con las opciones: objetivos, tareas, archivos, registro de actividad y presupuesto 5. El usuario escoge la opción registro de actividad 6. El sistema muestra la opciones disponibles: añadir, modificar, eliminar, filtrar, exportar 7. El usuario escoge la opción añadir 8. El sistema muestra los campos para ingresar la información de la actividad 9. El usuario ingresa la información 10. El sistema verifica el ingreso correcto de información.

<p>Flujos alternos</p>	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información de la actividad a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada. <p>filtrar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción filtrar 2. El sistema muestra los criterios de búsqueda 3. El usuario ingresa los criterios de búsqueda 4. El sistema realiza un filtro de la información según los criterios de búsqueda <p>exportar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción exportar 2. El sistema genera un archivo .PDF con información de las actividades del usuario según el criterio de búsqueda.
-------------------------------	---

Tabla 6.11. *Especificación tratar actividades*

	Caso de uso: tratar presupuesto del proyecto
Actores:	Usuario administrador Usuario limitado
Descripción:	Añadir, modificar, eliminar presupuesto del proyecto
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe contar con los permisos necesarios • El usuario debe estar asignado al proyecto contenedor del presupuesto a tratar
Pos condiciones:	<ul style="list-style-type: none"> • Resultados almacenados
Flujo normal de eventos:	<p>añadir</p> <ol style="list-style-type: none"> 1. El usuario escoge la opción mis proyectos 2. El sistema muestra los proyectos asignados al usuario 3. El usuario escoge un proyecto 4. El sistema muestra un menú con las opciones: objetivos, tareas, archivos, registro de actividad y presupuesto 5. El usuario escoge la opción presupuesto 6. El sistema muestra la opciones disponibles: añadir, modificar, eliminar 7. El usuario escoge la opción añadir 8. El sistema muestra los campos para ingresar la información del presupuesto 9. El usuario ingresa la información 10. El sistema verifica el ingreso correcto de información.
Flujos alternos	<p>modificar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción modificar 2. El sistema muestra los campos con información del presupuesto a modificar. 3. El usuario modifica la información 4. El sistema verifica el ingreso correcto de información <p>eliminar</p> <ol style="list-style-type: none"> 1. En el paso 7 del Flujo Normal, el usuario escoge la opción eliminar 2. El sistema muestra un mensaje donde se indica si el usuario está seguro de la eliminación. 3. El usuario toma una decisión 4. El sistema realiza la operación indicada.

Tabla 6.12. Especificación tratar presupuesto del proyecto

	Caso de uso: Salir sistema
Actores:	Usuario administrador Usuario limitado
Descripción:	El usuario sale del sistema
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar registrado en el sistema • El usuario debe estar con la sesión iniciada en el sistema
Pos condiciones	Operación realizada con éxito
Flujo normal de eventos:	validar 1. El usuario escoge la opción cerrar sesión 2. El sistema realiza la operación indicada

Tabla 6.13. Especificación salir sistema

6.9.1.2.2 Diagrama de estados

Los diagramas de estado son una técnica importante en la descripción del comportamiento de un sistema. Detallan todos los estados posibles por los que puede pasar un objeto a lo largo de su vida y los estímulos que lo provocan.

Proyecto

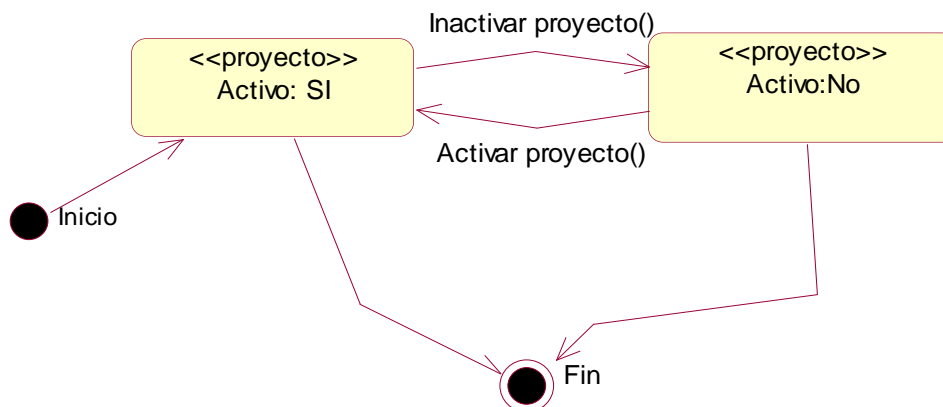


Figura 6.3. Diagrama de estados proyecto

Objetivo

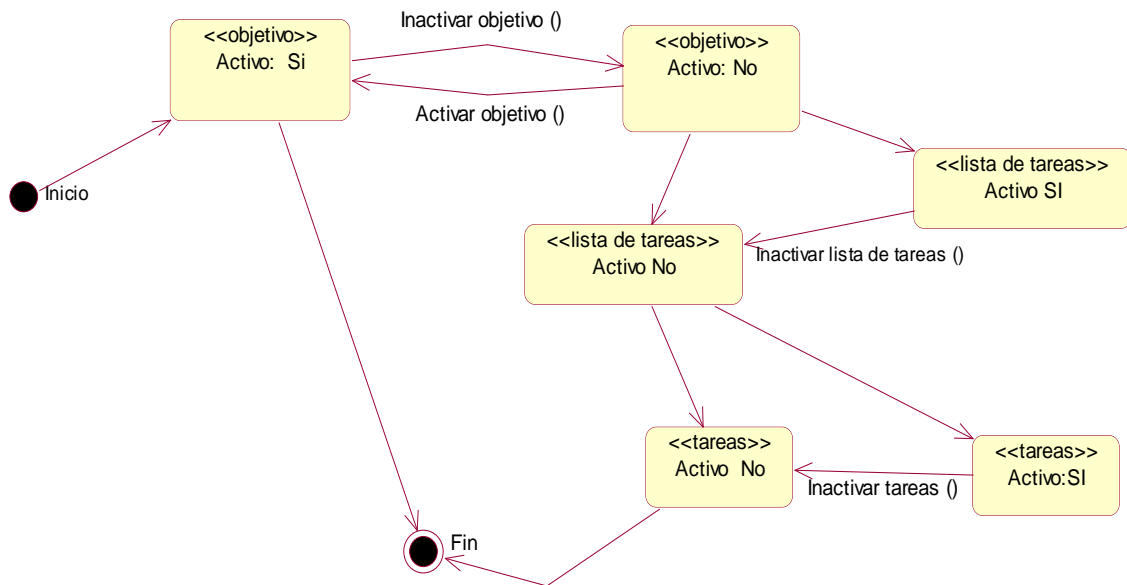


Figura 6.4. Diagrama de estados objetivo

Lista de tareas

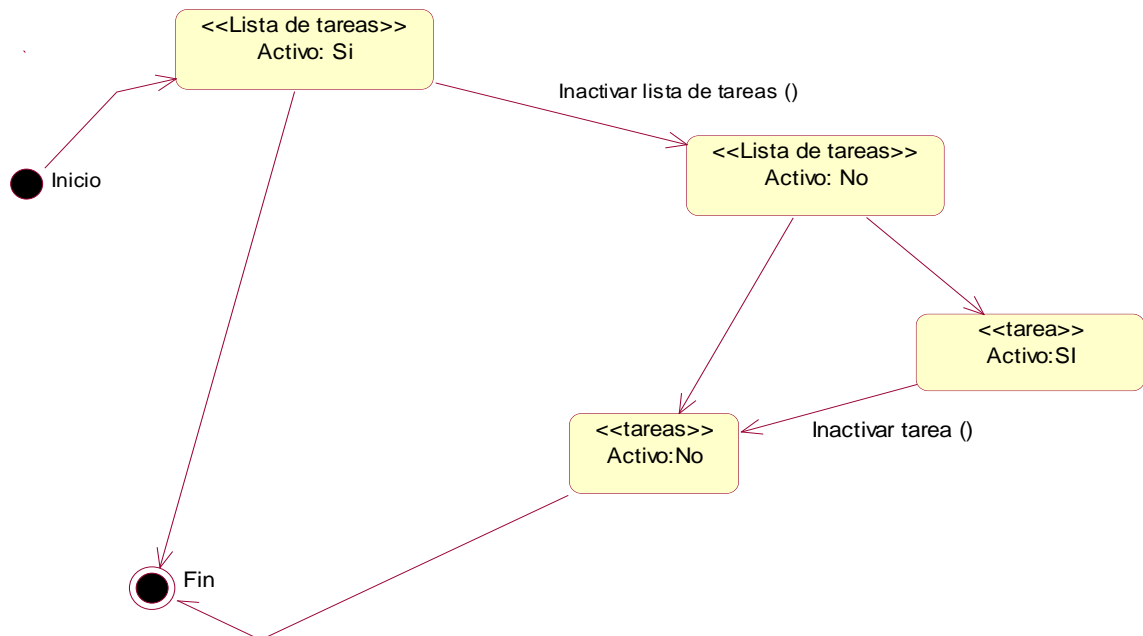


Figura 6.5. Diagrama de estados lista de tareas

Tareas

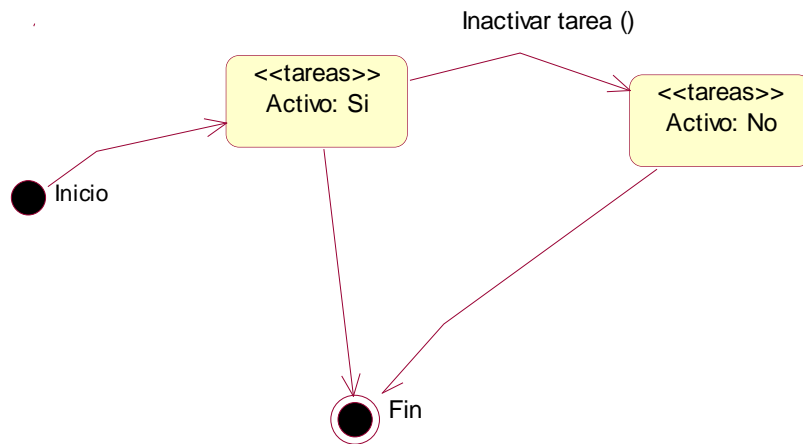


Figura 6.6. Diagrama de estados tareas

6.9.1.2.3 Diagrama de secuencias

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

Ingresar al sistema

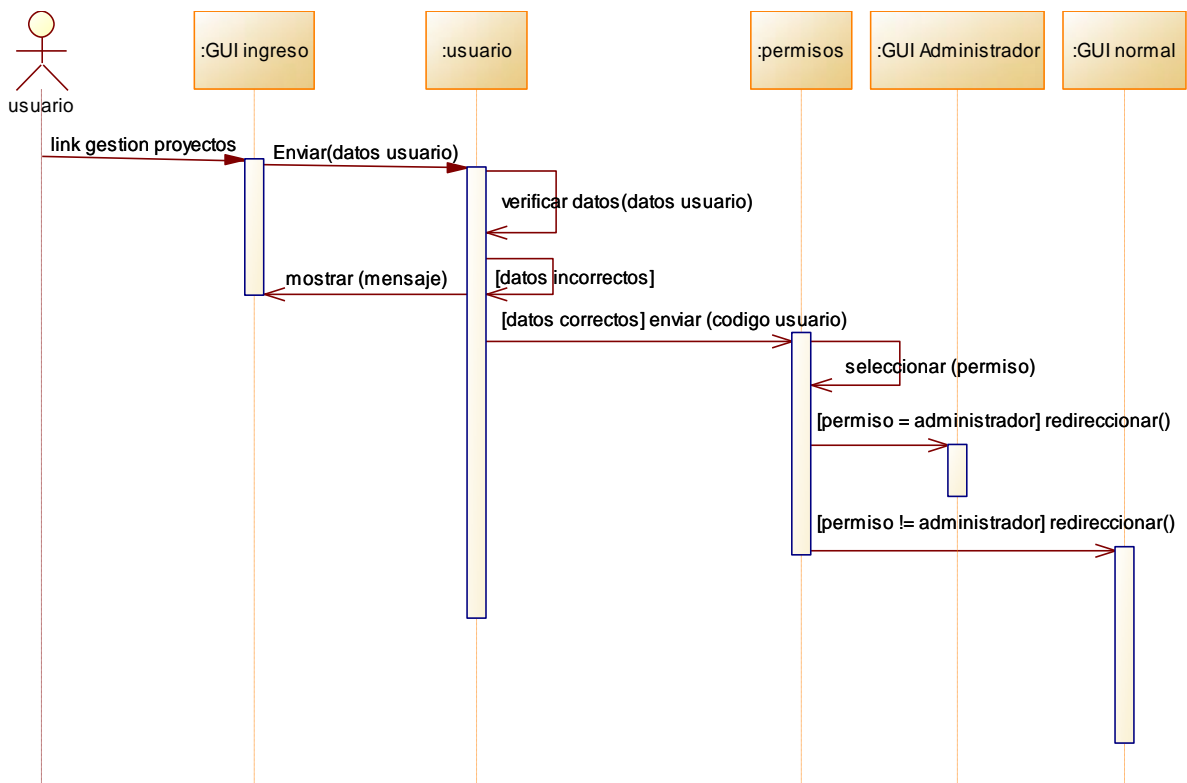


Figura 6.7. Diagrama de secuencias ingresar al sistema

Permisos

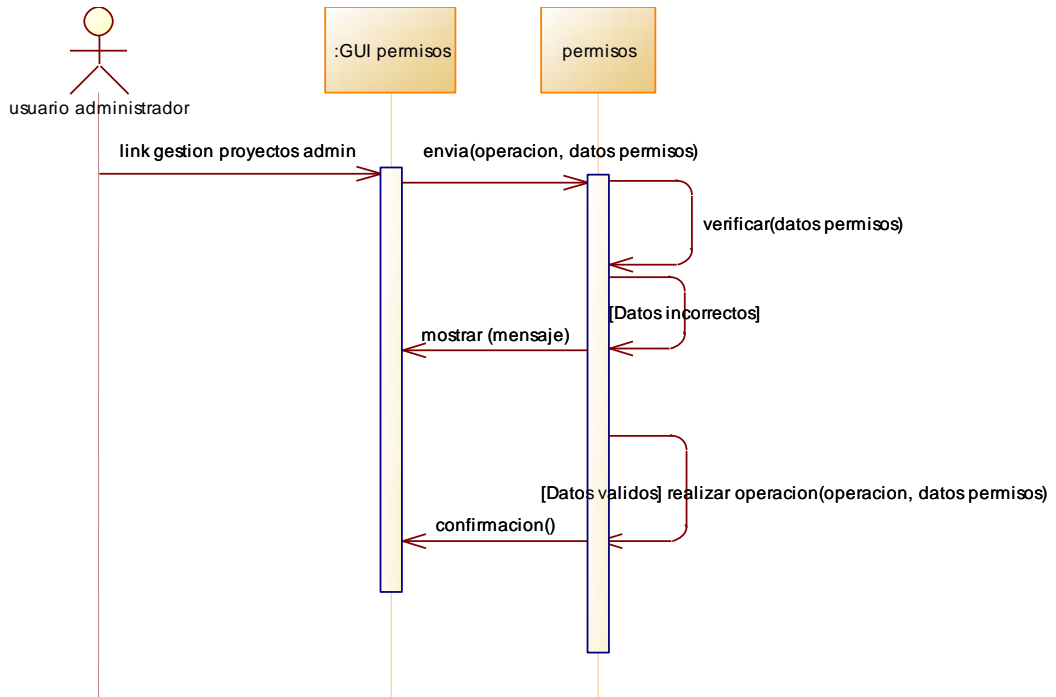


Figura 6.8. Diagrama de secuencias permisos

Administrar usuarios

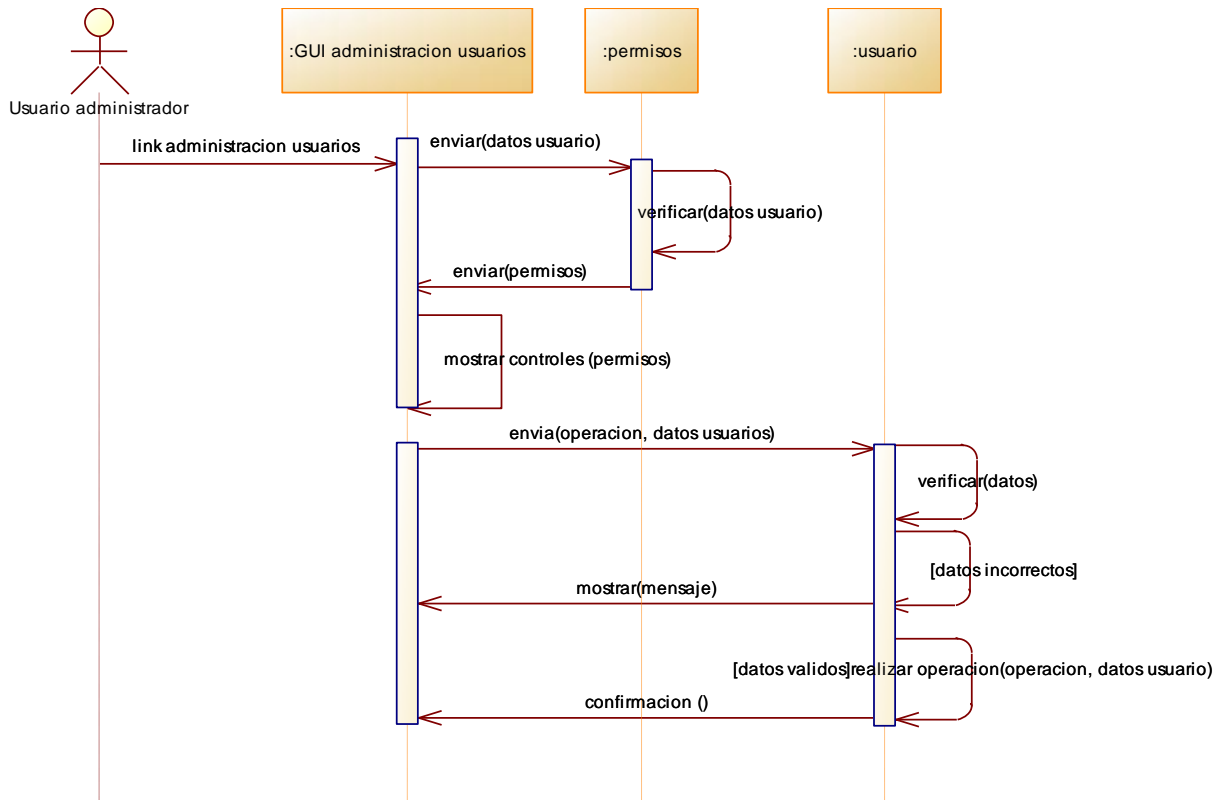


Figura 6.9. Diagrama de secuencias administrar usuarios

Administrar presupuesto

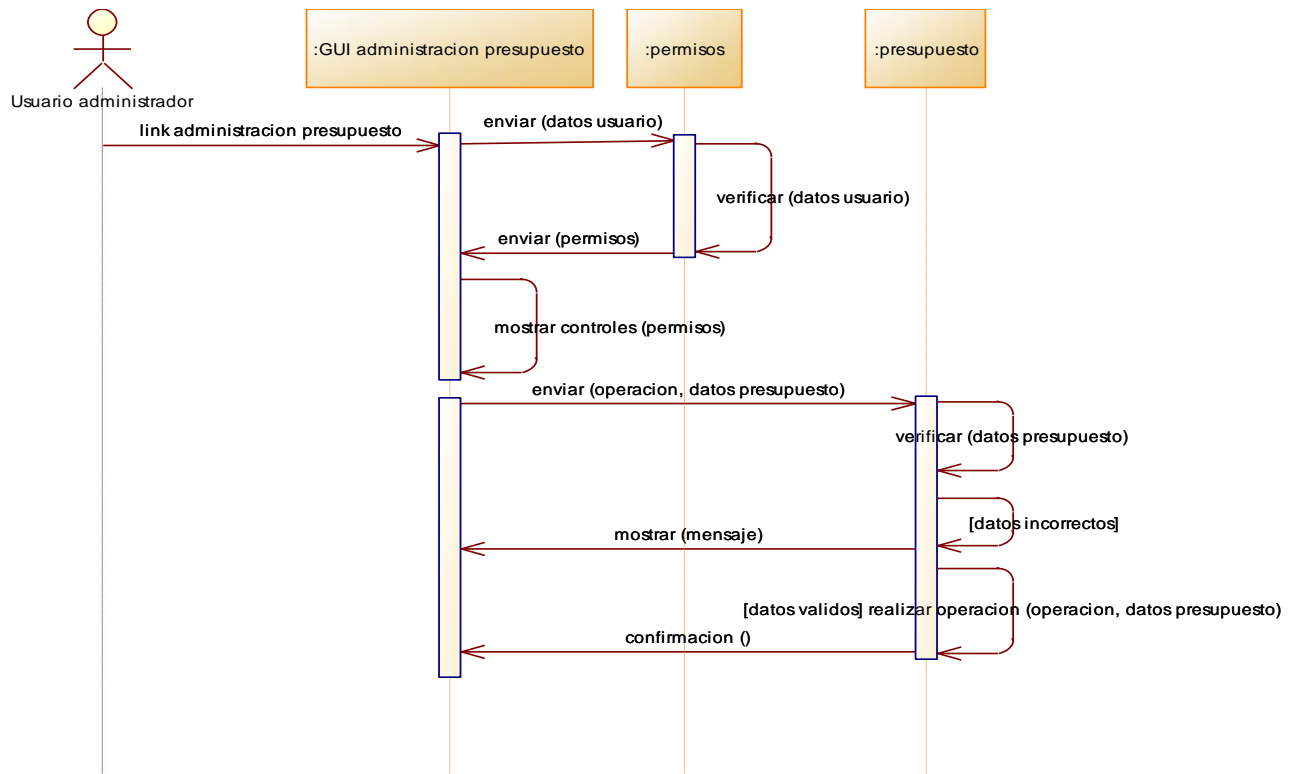


Figura 6.10. Diagrama de secuencias administrar presupuesto

Administrar proyectos

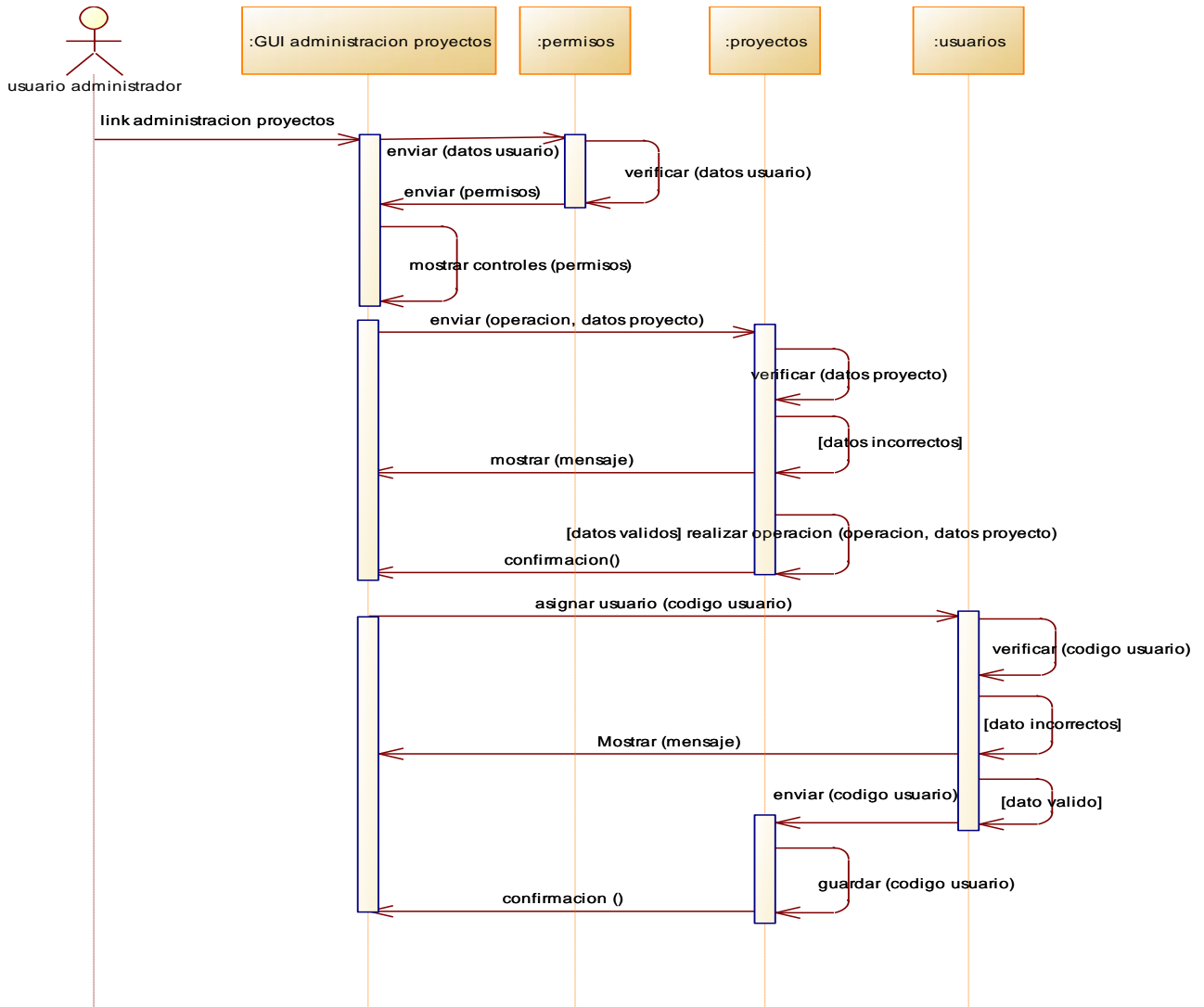


Figura 6.11. Diagrama de secuencias administrar proyectos

Editar perfil

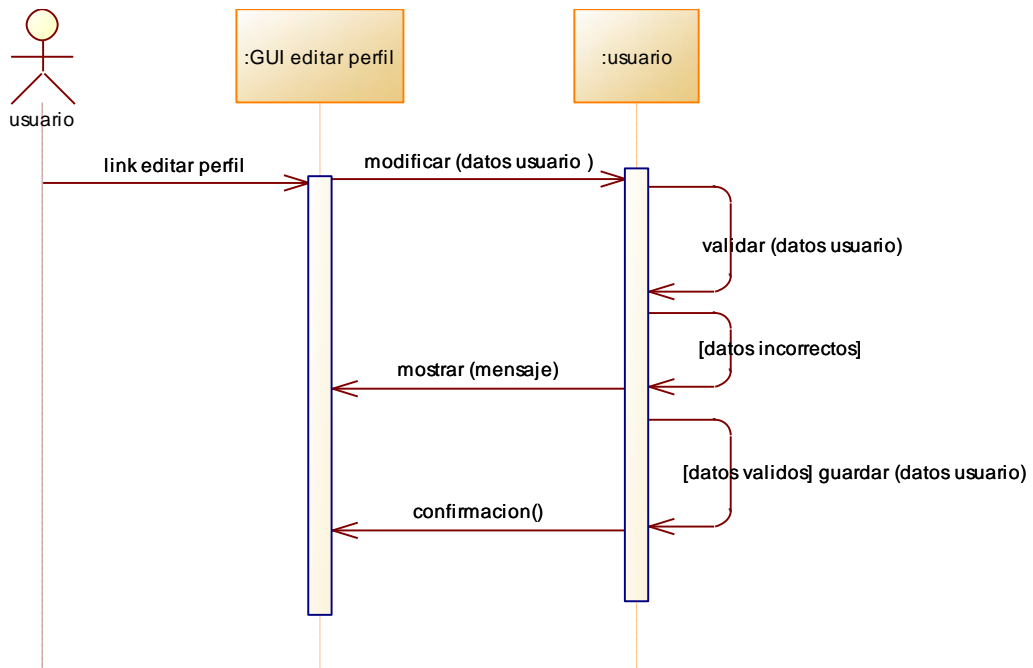


Figura 6.12. Diagrama de secuencias editar perfil

Tratar proyecto

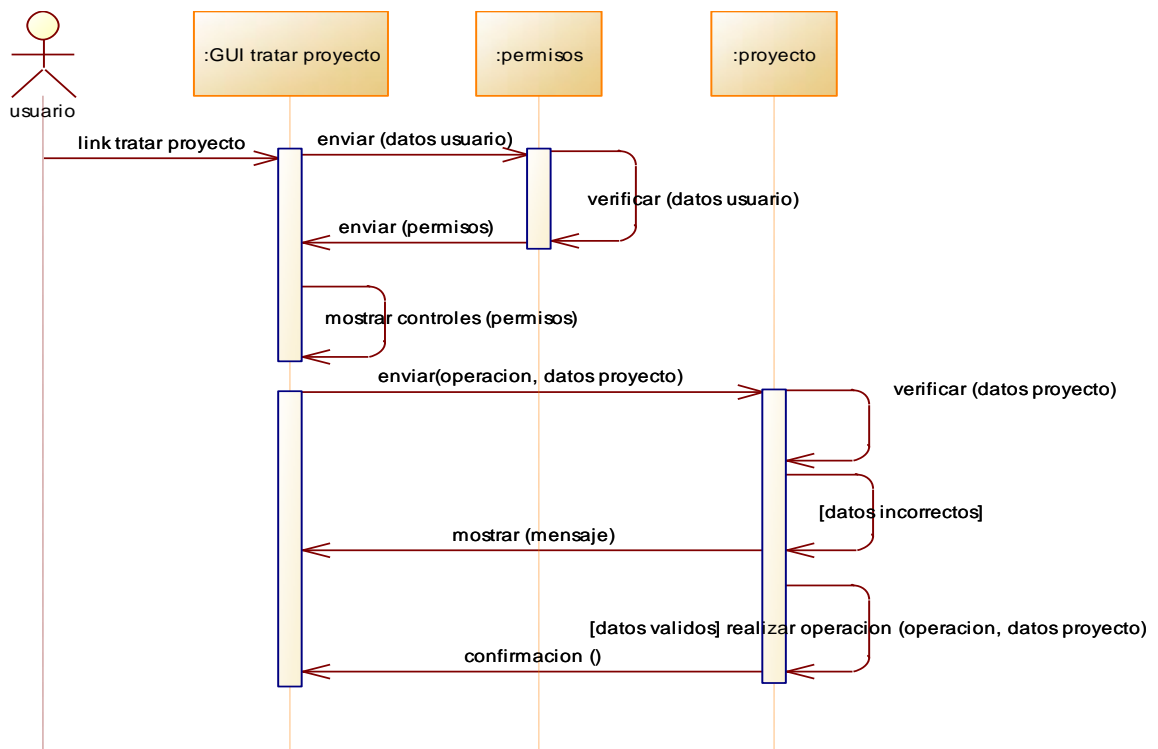


Figura 6.13. Diagrama de secuencias tratar proyecto

Tratar objetivos

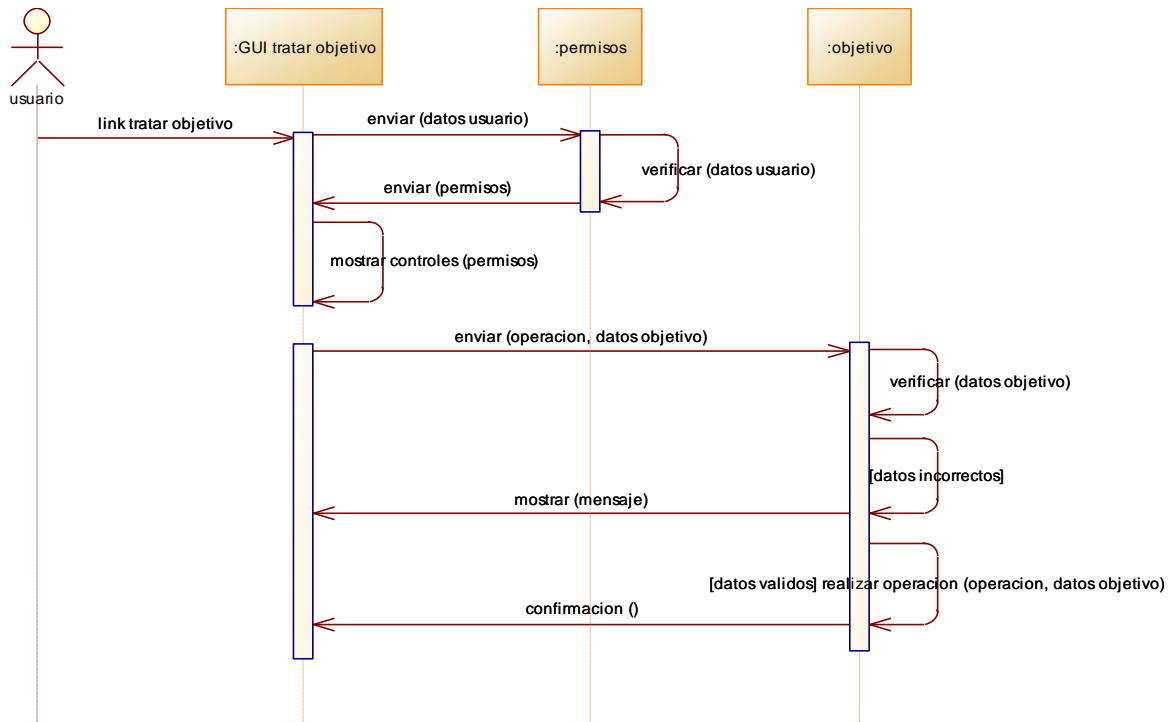


Figura 6.14. Diagrama de secuencias tratar objetivos

Tratar tareas

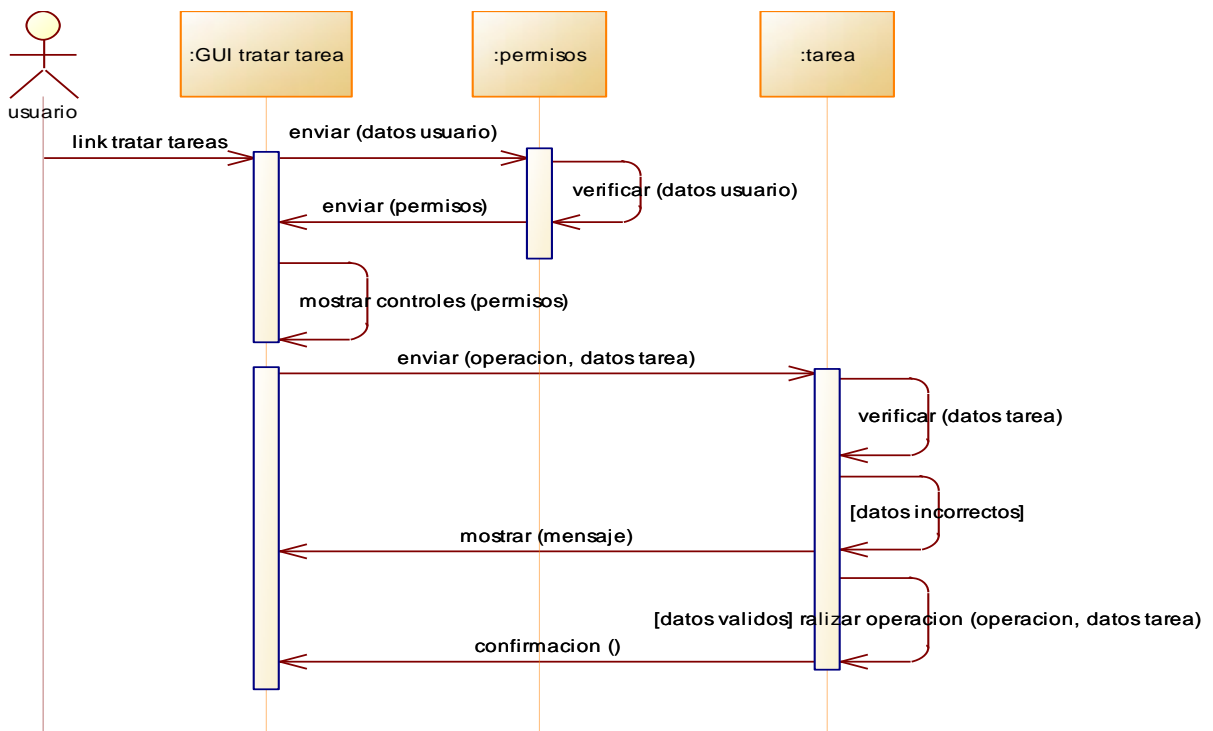


Figura 6.15. Diagrama de secuencias tratar tareas

Tratar archivos

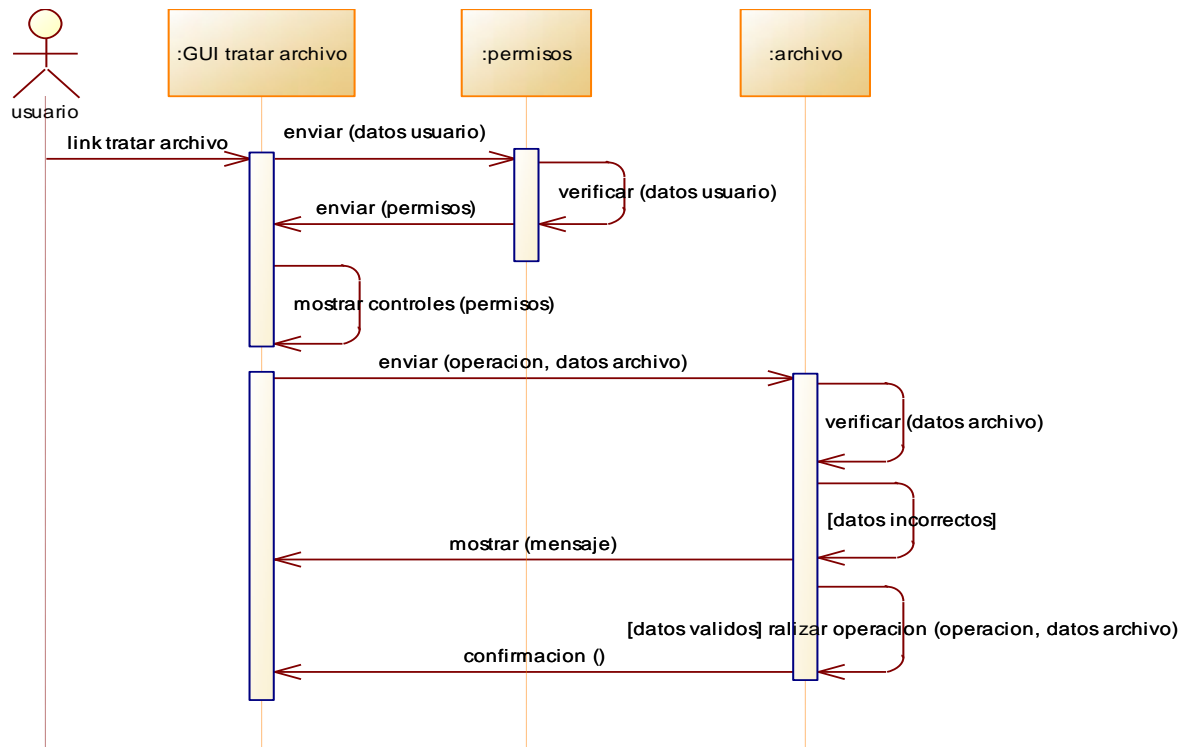


Figura 6.16. Diagrama de secuencias tratar archivos

Tratar actividades

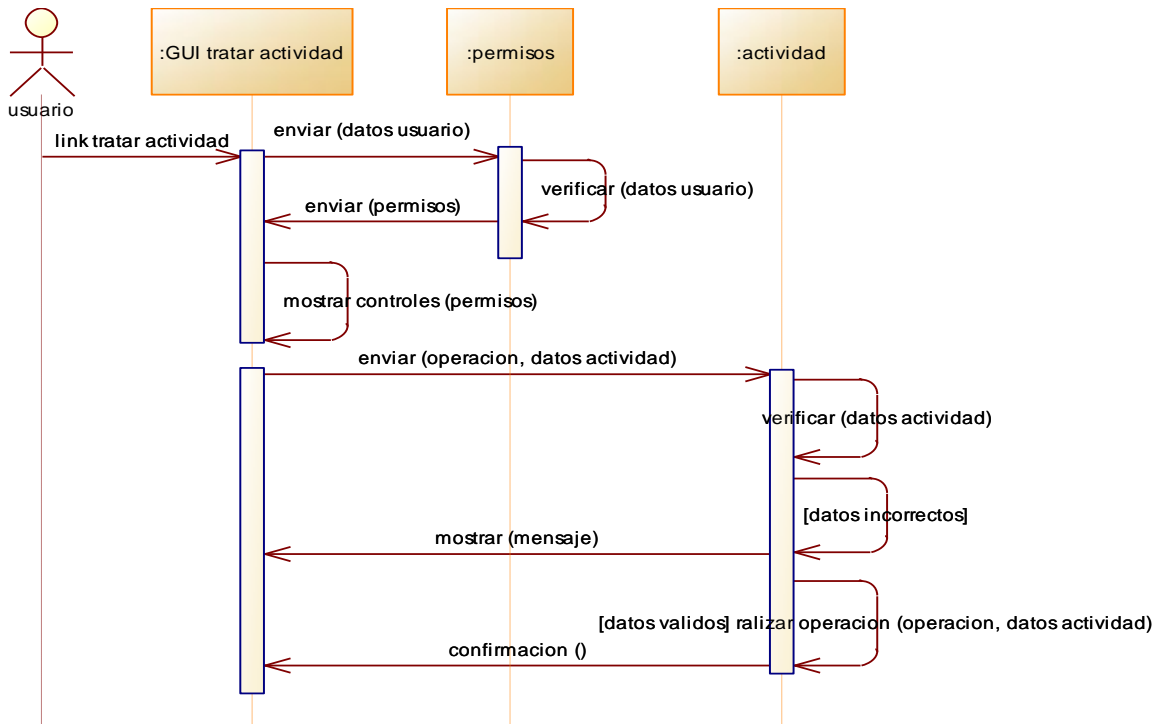


Figura 6.17. Diagrama de secuencias tratar actividades

Tratar presupuesto del proyecto

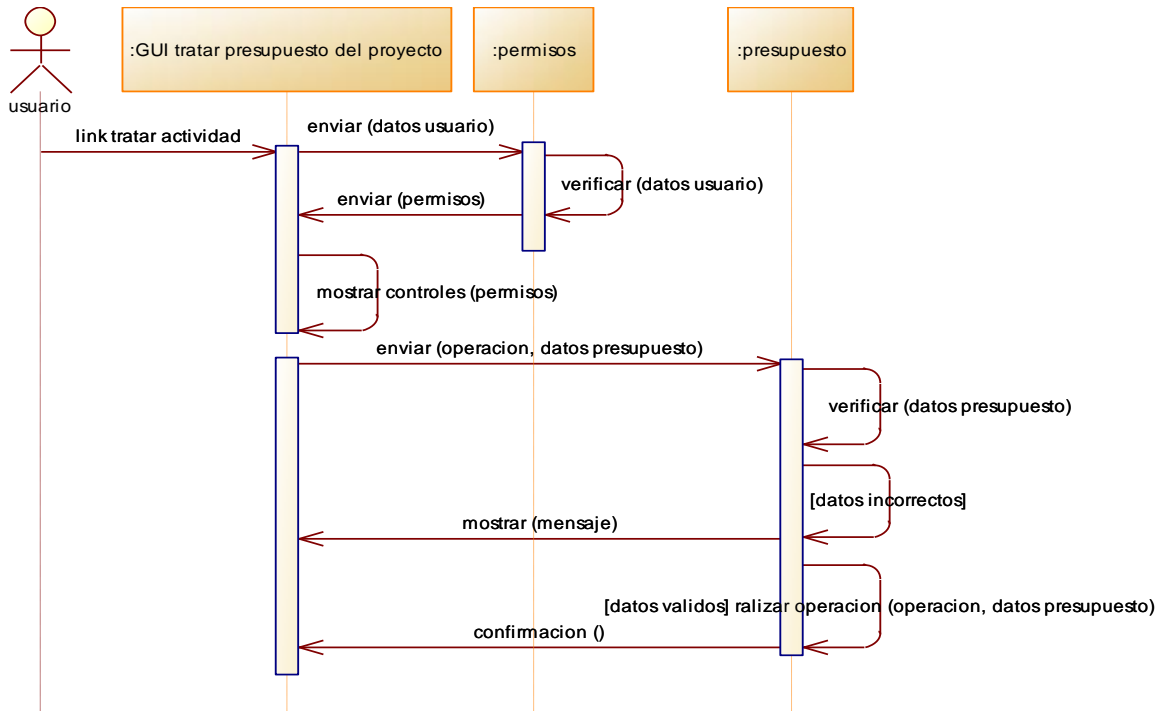


Figura 6.18. Diagrama de secuencias tratar presupuesto del proyecto

Salir sistema

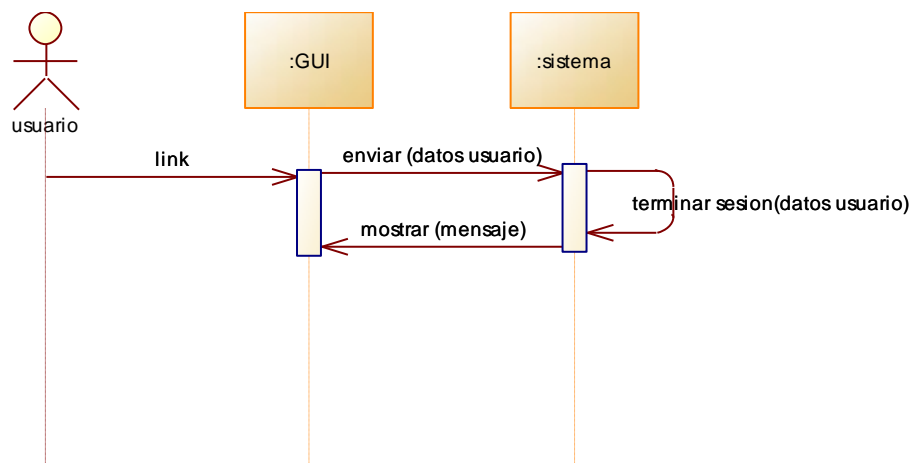


Figura 6.19. Diagrama de secuencias salir sistema

6.9.1.2.4 Diagrama de actividades

Un diagrama de actividad permite mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Para lograr una visión más detallada se procedió a graficar diagramas híbridos, los mismos que permiten agregar objetos en la descripción de actividades.

Ingresar al sistema

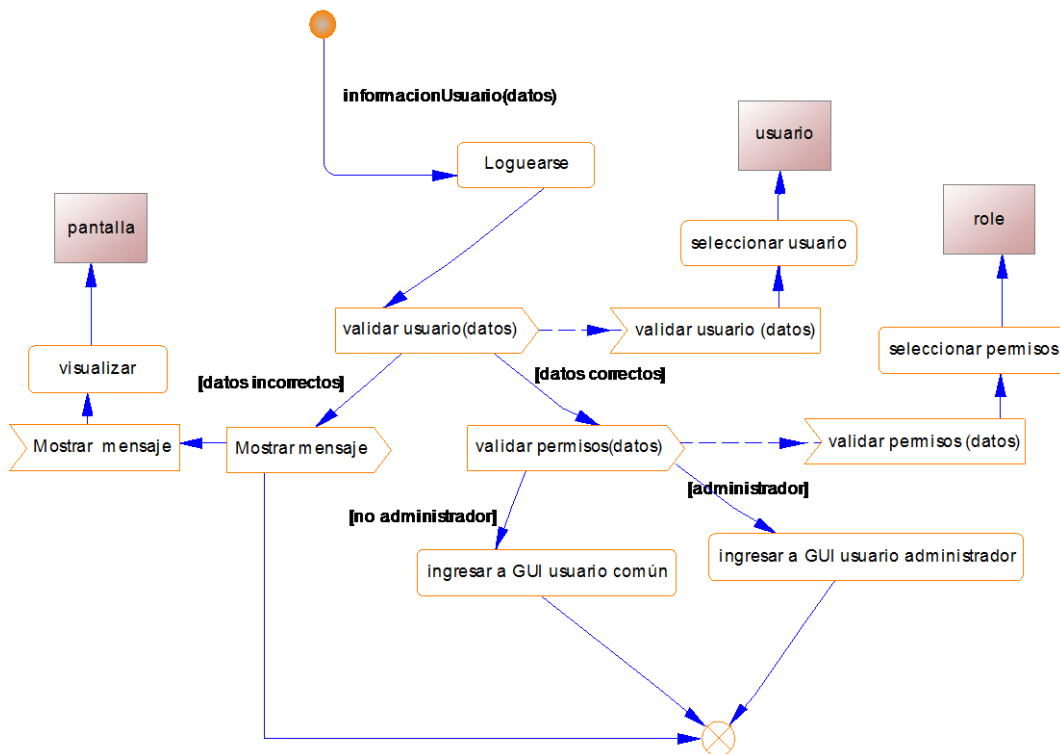


Figura 6.20. Diagrama de actividades ingresar al sistema.

Administrar usuarios

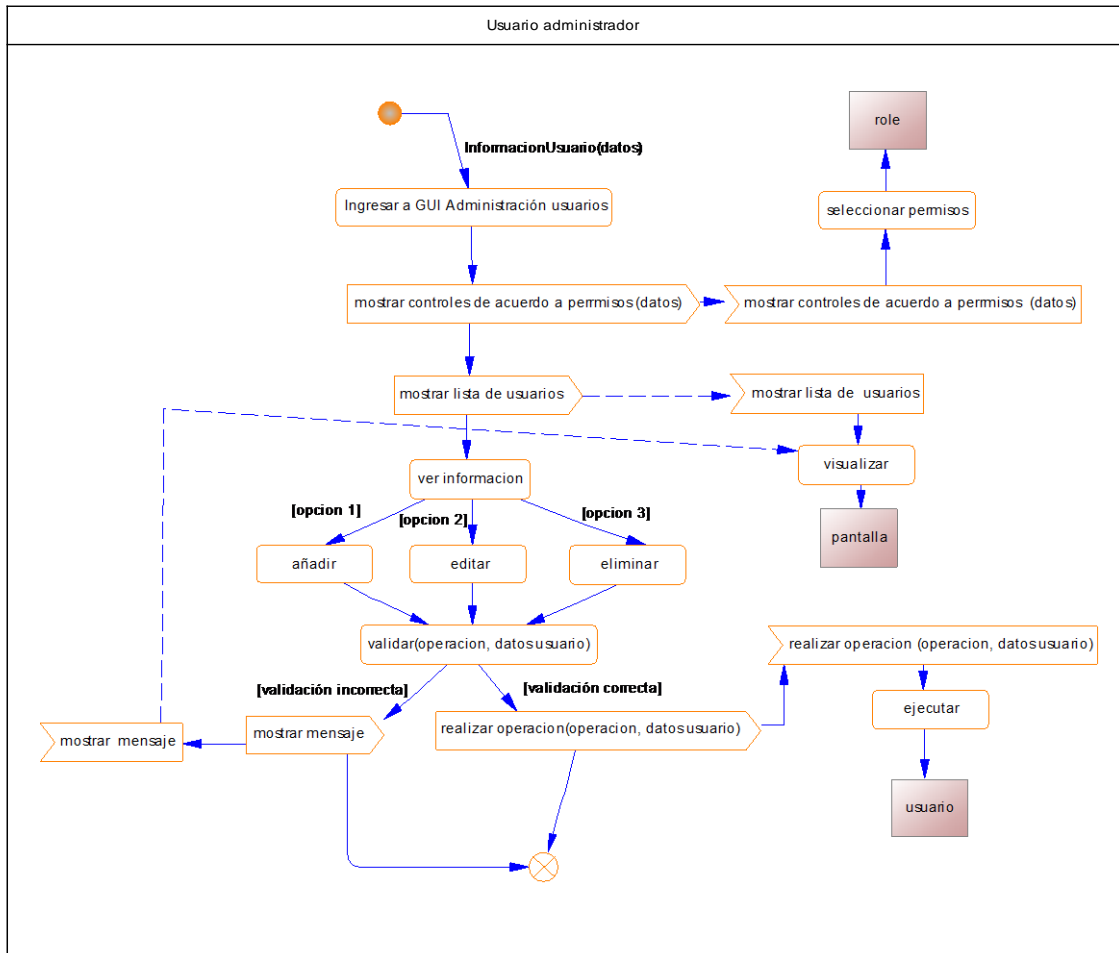


Figura 6.21. Diagrama de actividades administrar usuarios

Administrar presupuesto

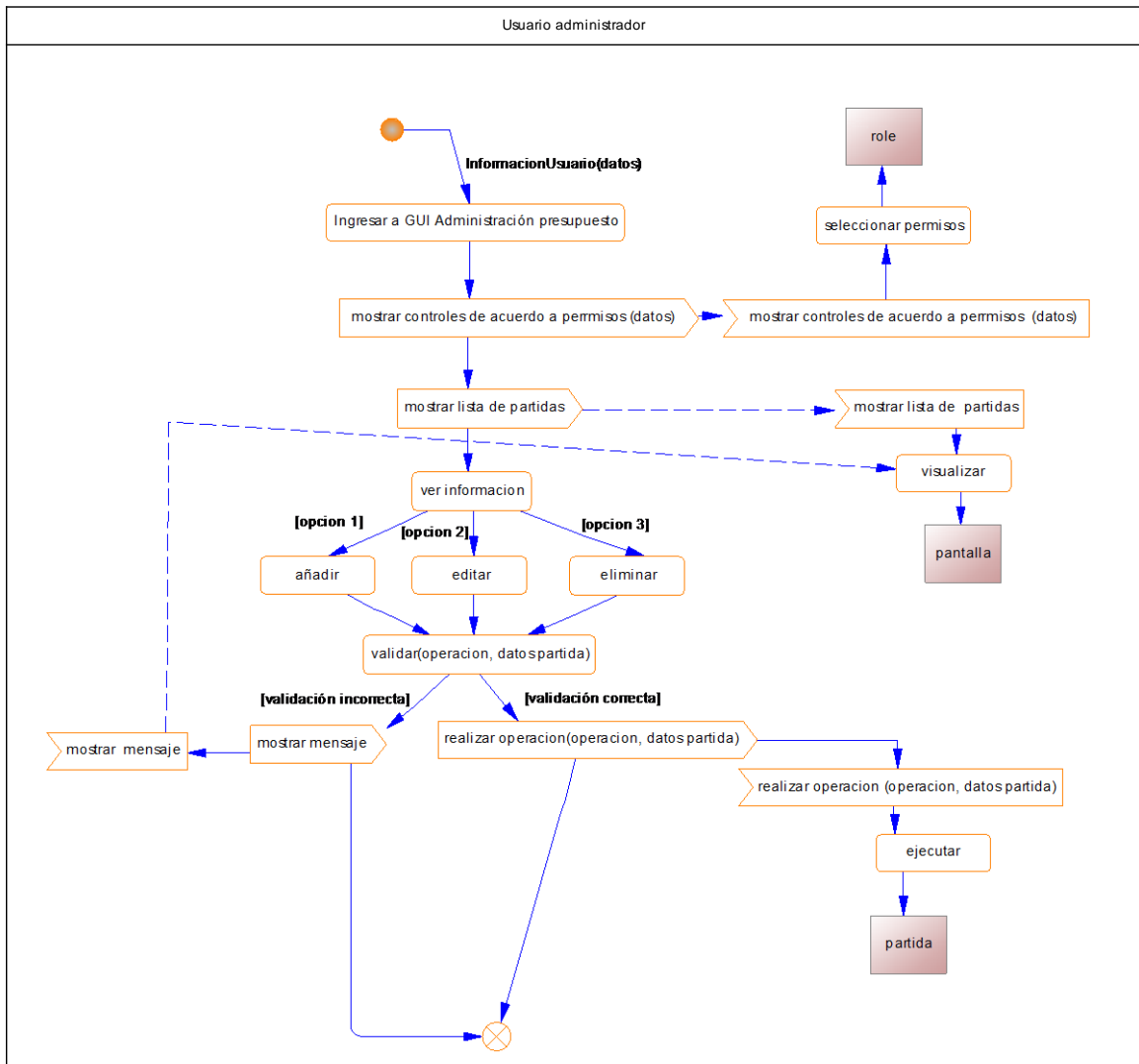


Figura 6.22. Diagrama de actividades administrar presupuesto

Administrar proyectos

Editar perfil

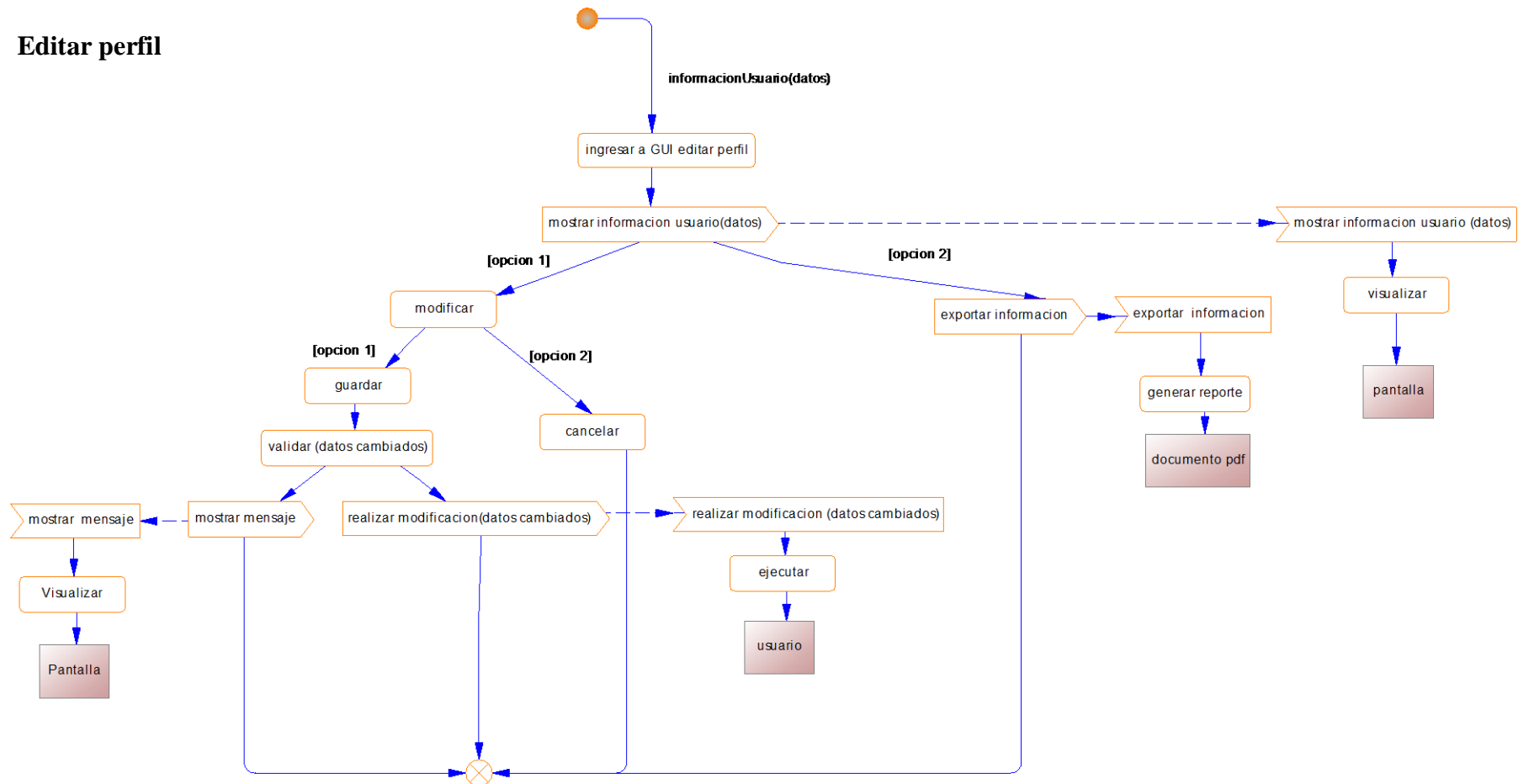


Figura 6.24. Diagrama de actividades editar perfil

Tratar proyecto

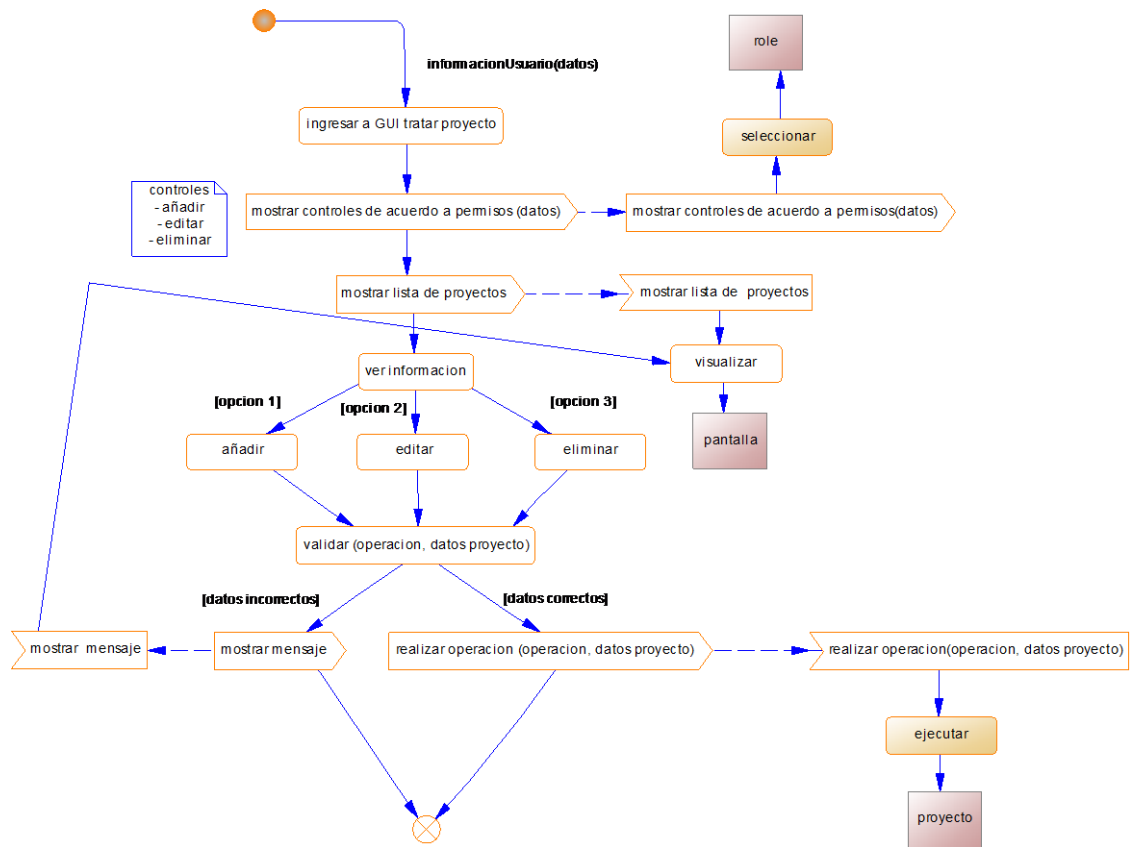


Figura 6.25. Diagrama de actividades tratar proyecto

Tratar objetivos

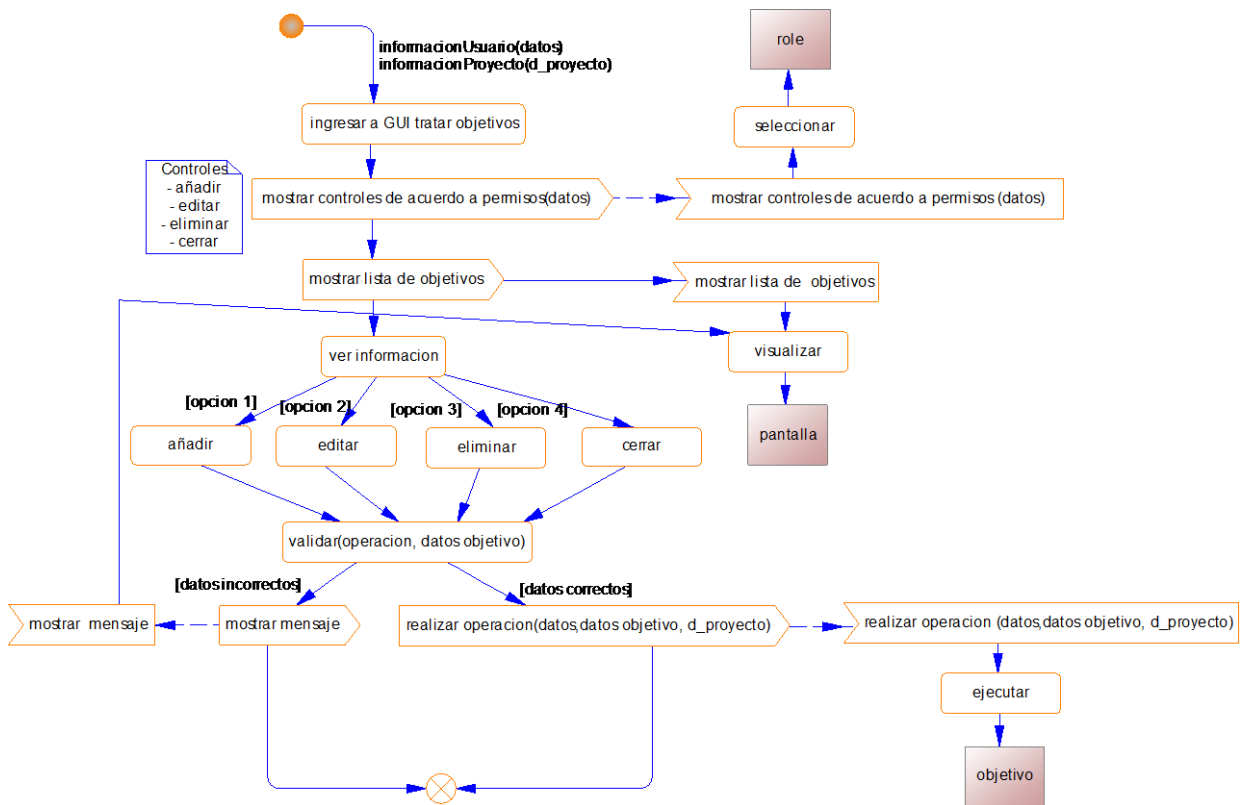


Figura 6.26. Diagrama de actividades tratar objetivos

Tratar tareas

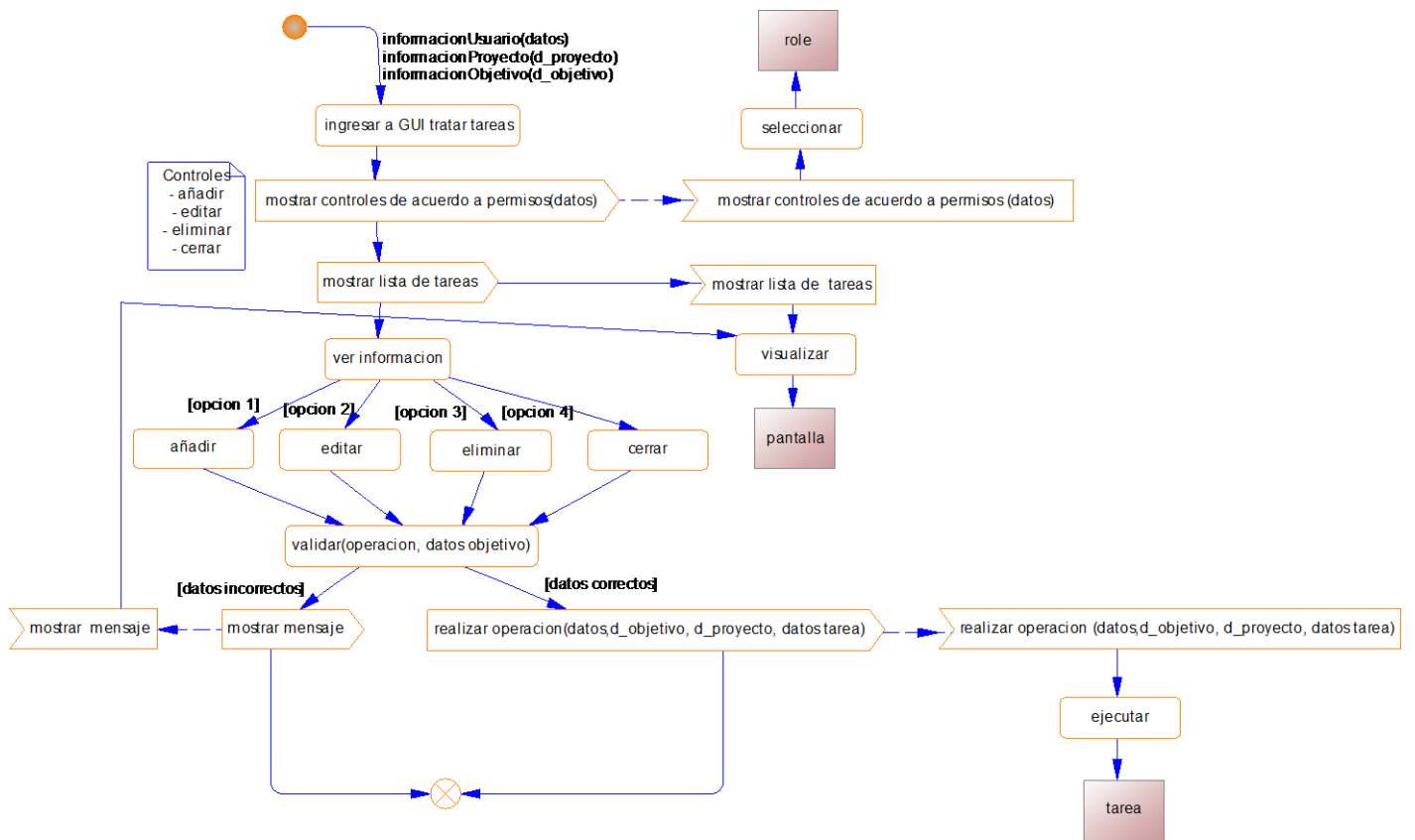


Figura 6.27. Diagrama de actividades tratar tareas

Tratar archivos

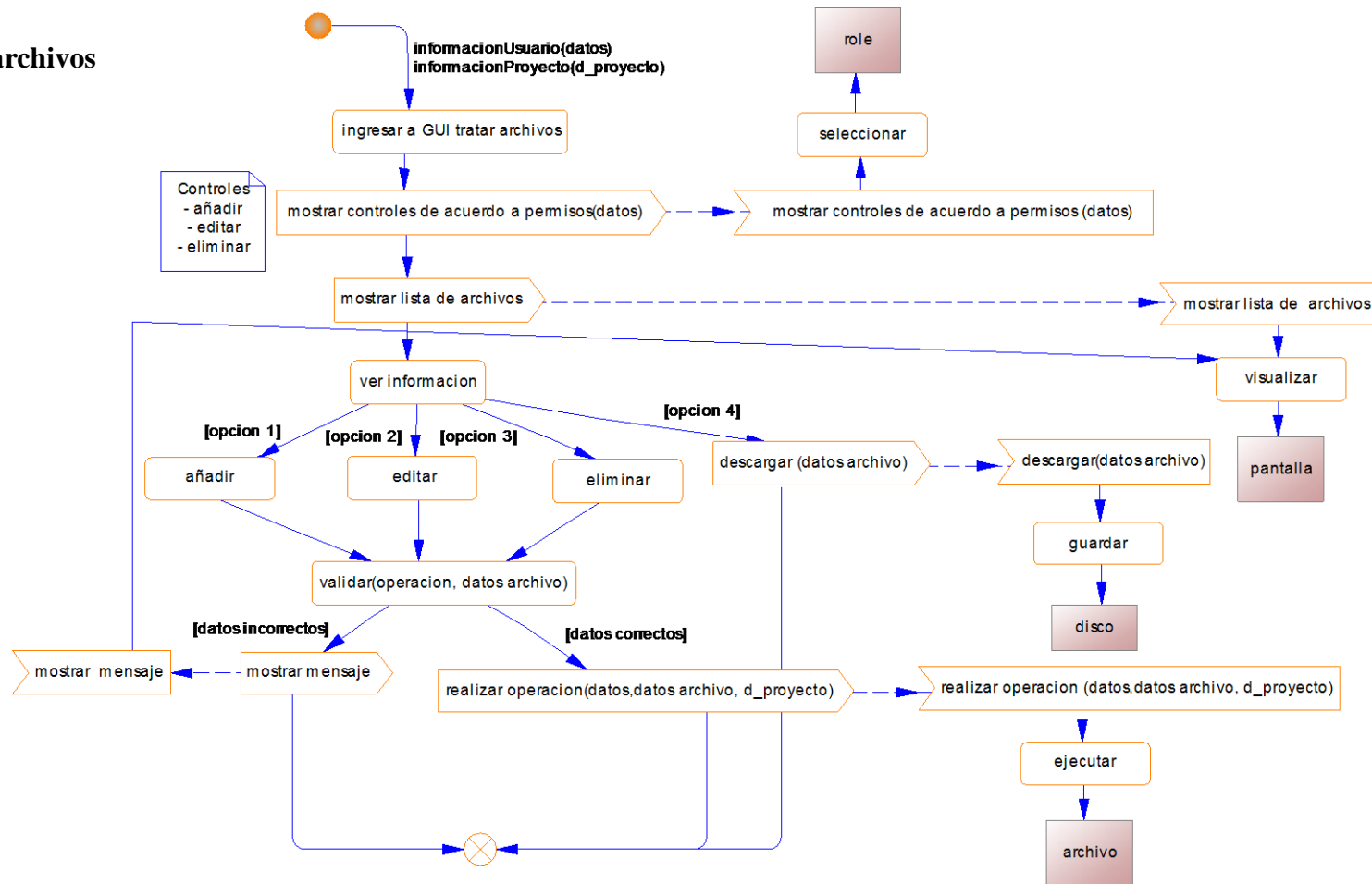


Figura 6.28. Diagrama de actividades tratar archivos

Tratar actividades

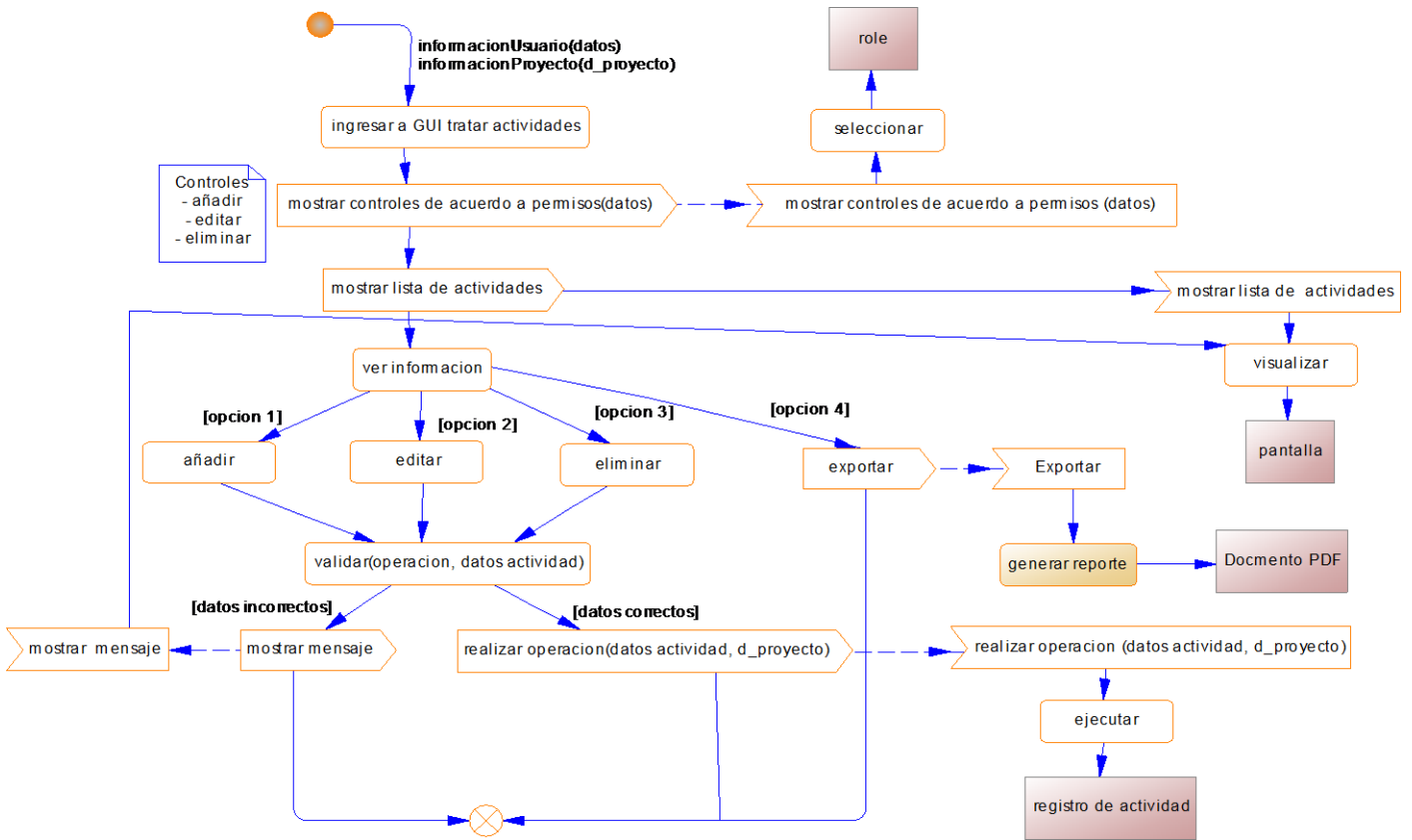
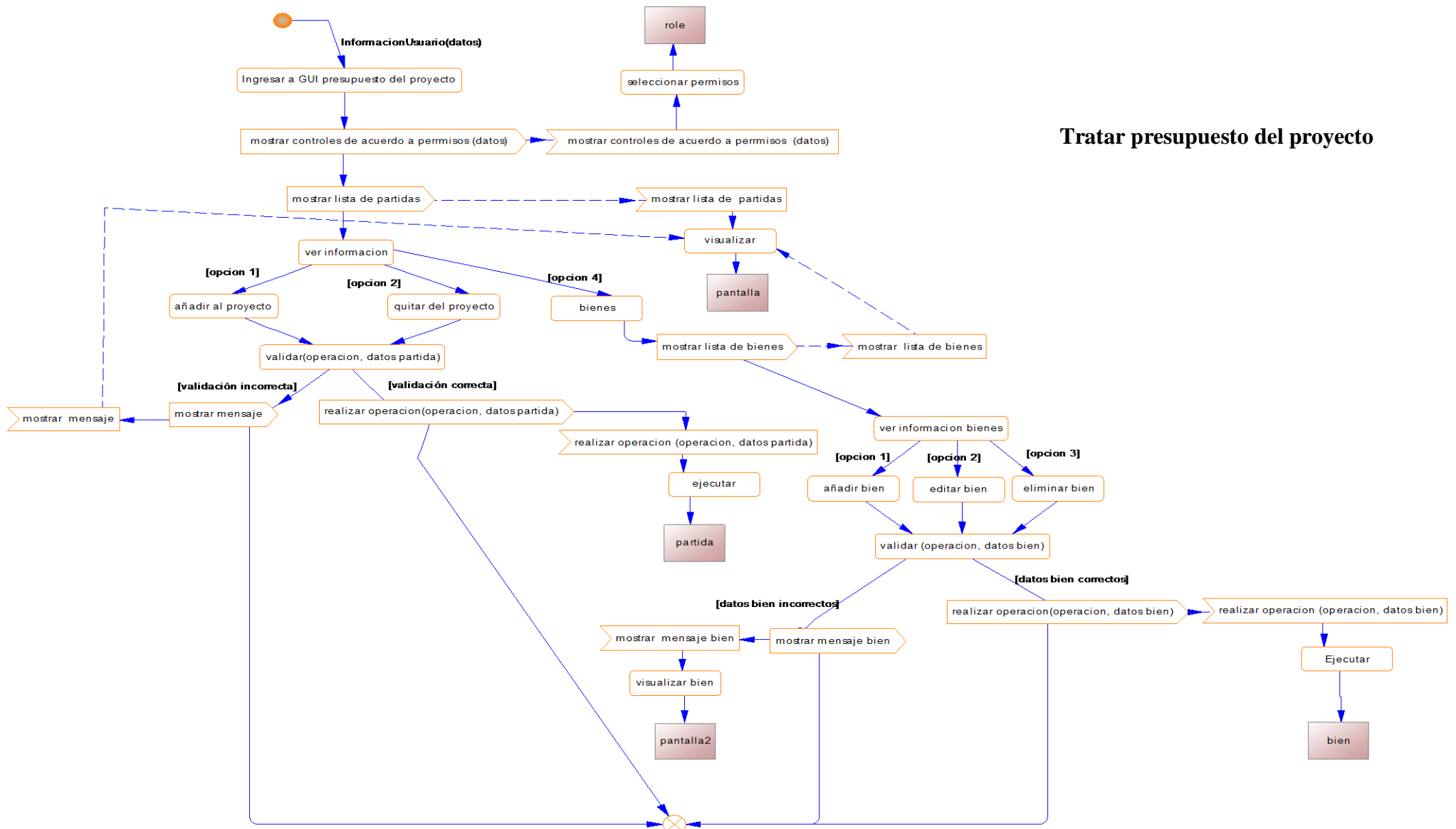


Figura 6.29. Diagrama de actividades tratar actividades



Tratar presupuesto del proyecto

Figura 6.30. Diagrama de actividades tratar presupuesto del proyecto

6.9.1.2.5 Diagrama de clases

¿Qué es un Diagrama de clases?

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema.

¿Cómo surgió el diagrama de clases del proyecto investigado?

El diagrama surgió de la detección de objetos en la fase de análisis del sistema, diagramas de secuencias, procesos, y actividades permitieron su localización; posteriormente dichos objetos fueron agrupados por tipo para su clasificación en un diagrama de clase.

En la siguiente figura se muestra las clases que fueron encontradas:



Figura 6.31. *Diagrama de clases*

A continuación se procedió a diseñar las relaciones que existe entre las clases y a definir los atributos de las mismas. (Ver figura 6.32)

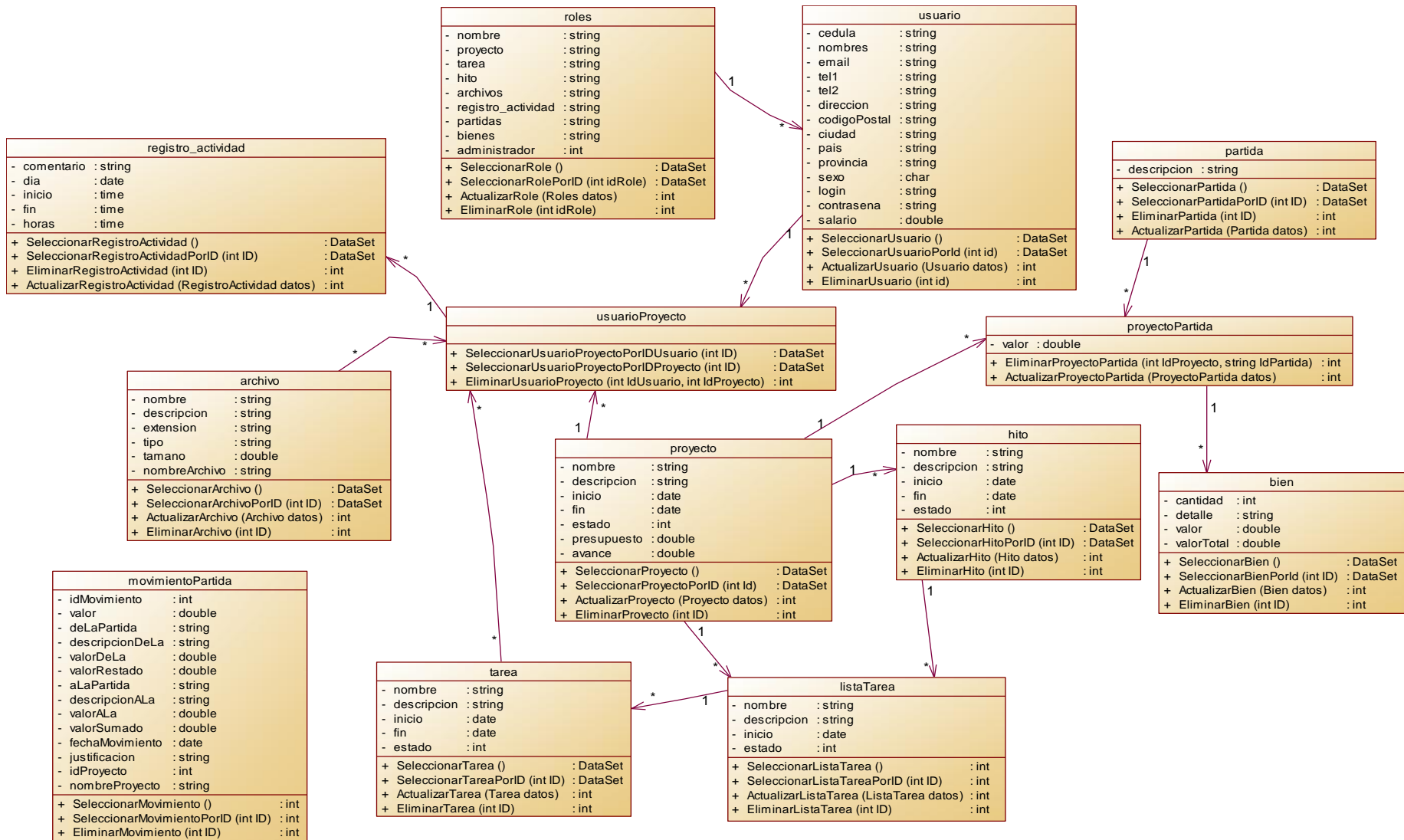


Figura 6.32. Diagrama de clases y sus relaciones.

6.9.1.2.6 Diagrama de despliegue

Los diagramas de despliegue se usan para modelar las relaciones físicas entre los componentes hardware y software en el sistema.

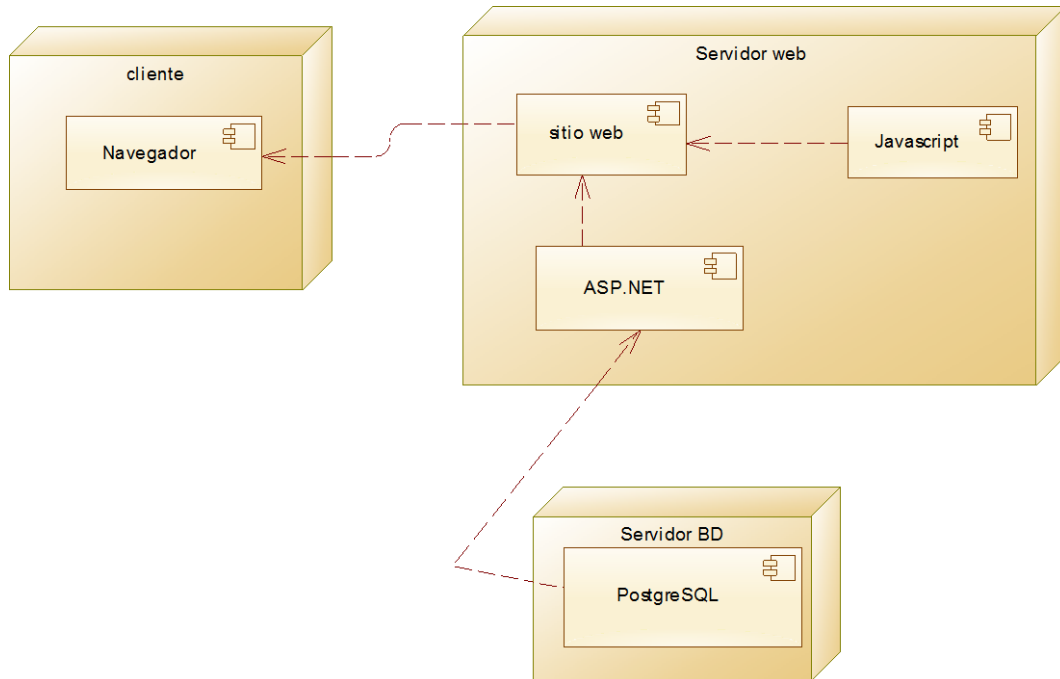


Figura 6.33. *Diagrama de despliegue*

6.9.2 Diseño del sistema

Una vez analizado el proceso de gestión de proyectos en la UOCENI – ISEI y determinado los requerimientos que deberá cumplir el sistema informático, el siguiente paso es diseñar la solución con los suficientes detalles para permitir su interpretación y posterior implementación.

6.9.2.1 Diseño de la base de datos

Se utilizó el diagrama de clases como referencia para el diseño del diagrama entidad- relación, donde las clases y sus atributos representan directamente a las entidades lógicas y sus atributos, y las asociaciones entre las clases permitieron diseñar las relaciones entre las entidades.

Ya en el diagrama entidad – relación y modelo relacional lógico, se determinó que atributos son claves primarias, foráneas y únicas. El diseño de la base de datos se puede observar en la figura “*Figura 6.34 Diagrama entidad-relación*”, “*Figura 6.35 Modelo relacional lógico*” y la “*Figura 6.36 Modelo relacional físico*”.

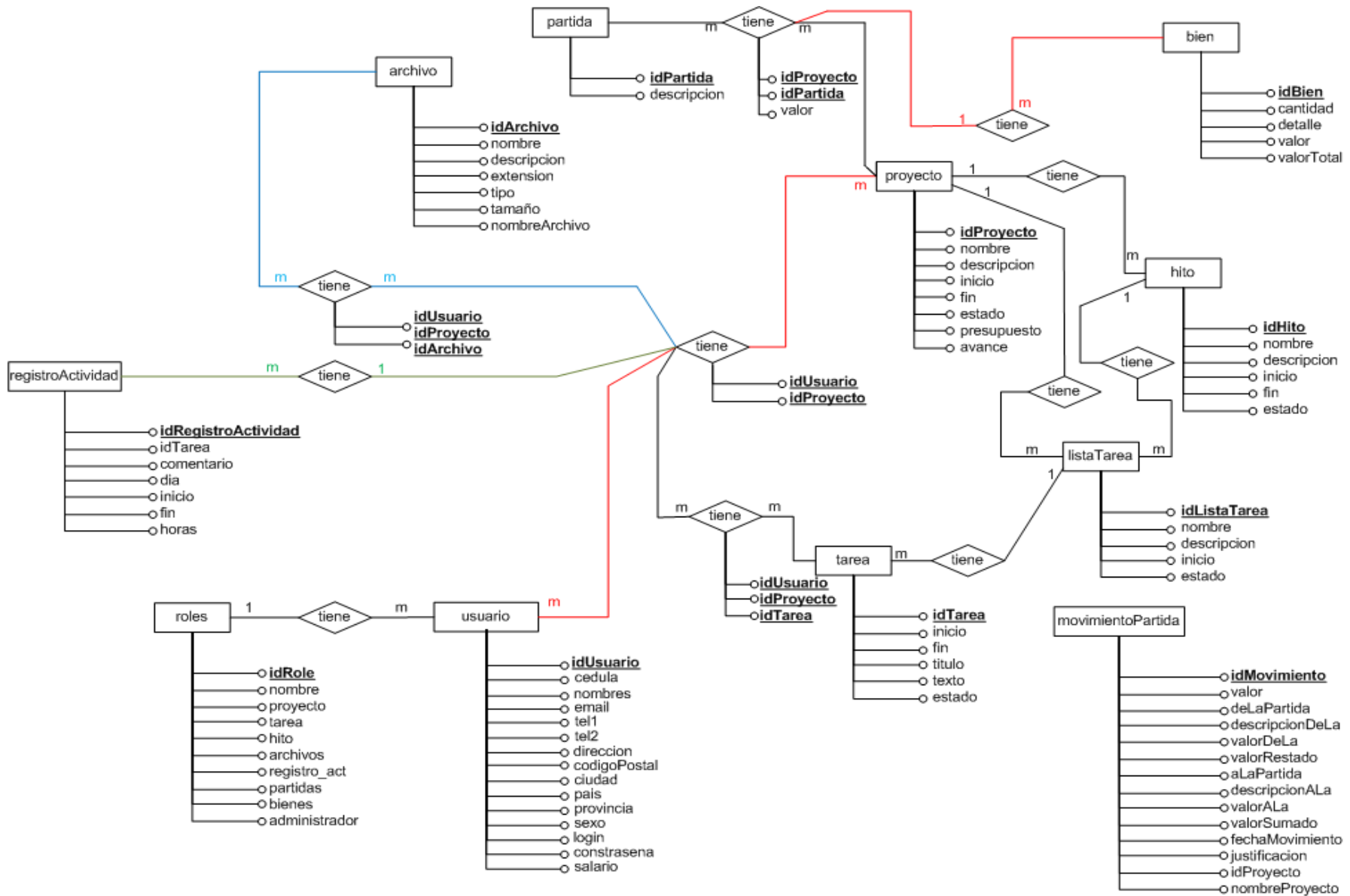


Figura 6.34. Diagrama entidad- relación

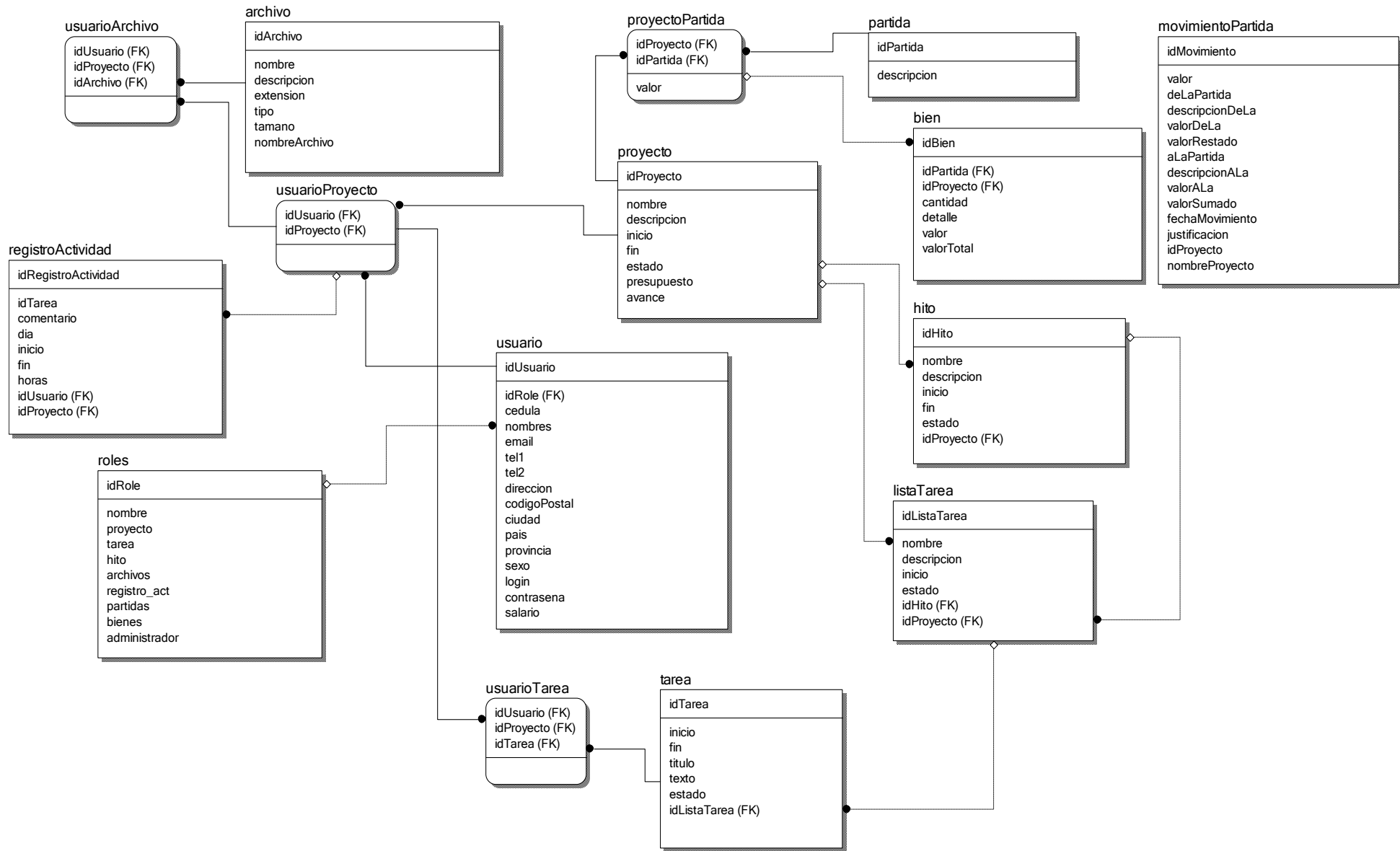


Figura 6.35. Modelo relacional lógico

6.9.2.2 Diccionario de datos

TABLA: roles			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idRole	Integer	Código del role	Primary key
nombre	Character(100)	Nombre del role	Unique
proyecto	Character(1000)	Nombre que identifica permisos sobre proyectos	Not null
tarea	Character(1000)	Nombre que identifica permisos sobre tareas	Not null
hito	Character(1000)	Nombre que identifica permisos sobre hitos	Not null
archivos	Character(1000)	Nombre que identifica permisos sobre archivos	Not null
registro_act	Character(1000)	Nombre que identifica permisos sobre actividades	Not null
partidas	Character(1000)	Nombre que identifica permisos sobre partidas	Not null
bienes	Character(1000)	Nombre que identifica permisos sobre bienes	Not null
administrador	Integer	Valor que identifica el permiso de administrador	Not null, 1 = administrador 0 = no es administrador

Tabla 6.14. Descripción de la tabla roles

TABLA: usuario			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idUsuario	Integer	Código del usuario	Primary key
idRole	Integer	Código del role	Foreign key (idRole de la tabla roles)
cedula	Character(10)	Cédula del usuario	Not null
nombres	Character(200)	Nombres y Apellidos del usuario	Not null
email	Character(200)	Correo electrónico del usuario	
tel1	Character(9)	Teléfono convencional del usuario	
tel2	Character(9)	Teléfono móvil del usuario	
direccion	Character(500)	Dirección del usuario	
codigoPostal	Character(100)	Código postal del usuario	
ciudad	Character(100)	Nombre de la ciudad donde reside el usuario	
pais	Character(100)	Nombre del país donde reside el usuario	
provincia	Character(100)	Nombre de la provincia donde reside el usuario	

sexo	Character(1)	Carácter que identifica el sexo del usuario.	Not null, m = masculino f = femenino
login	Character(100)	Login del usuario	Unique
contrasena	Character(100)	Contraseña del usuario	Unique
salario	Numeric	Salario del empleado	

Tabla 6.15. Descripción de la tabla usuario

TABLA: proyecto			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idProyecto	Integer	Código del proyecto	Primary key
nombre	Character(200)	Nombre del proyecto	Not null
descripcion	Character(500)	Descripción del proyecto	Not null
inicio	Date	Fecha inicio del proyecto	Not null
fin	Date	Fecha fin del proyecto	Not null
estado	Integer	Valor que identifica el estado del proyecto.	Not null, 1 = abierto 0 = cerrado
presupuesto	Numeric	Presupuesto del proyecto	Not null
Avance	Numeric	Porcentaje de avance del proyecto	Default = 0

Tabla 6.16. Descripción de la tabla proyecto

TABLA: usuarioProyecto			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idUsuario	Integer	Código del usuario	Primary key, Foreign key (idUsuario de la tabla usuario)
idProyecto	Integer	Código del proyecto	Primary key, Foreign key (idProyecto de la tabla proyecto)

Tabla 6.17. Descripción de la tabla usuarioProyecto

TABLA: hito			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idHito	Integer	Código del hito	Primary key
idProyecto	Integer	Código del proyecto	Foreign key (idProyecto de la tabla proyecto)
nombre	Character(100)	Nombre del hito	Not null
descripcion	Character(200)	Descripción del hito	

inicio	Date	Fecha inicio del hito	Not null
fin	Date	Fecha fin del hito	Not null
estado	Integer	Valor que identifica el estado del hito.	Not null, 1 = abierto 0 = cerrado

Tabla 6.18. Descripción de la tabla hito

TABLA: listaTarea			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idListaTarea	Integer	Código de la lista de tareas.	Primary key
idProyecto	Integer	Código del proyecto.	Foreign key (idProyecto de la tabla proyecto)
idHito	Integer	Código del hito.	Foreign key (idHito de la tabla hito)
nombre	Character(100)	Nombre de la lista de tareas.	Not null
descripcion	Character(300)	Descripción de la lista de tareas.	Not null
inicio	Date	Fecha inicio de la lista de tareas.	Not null
estado	Integer	Valor que identifica el estado de la lista de tareas.	Not null, 0 = cerrado 1 = abierto

Tabla 6.19. Descripción de la tabla listaTarea

TABLA: tarea			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idTarea	Integer	Código de la tarea	Primary key
idListaTarea	Integer	Código de la lista de tareas	Foreign key (idListaTarea de la tabla listaTarea)
inicio	Date	Fecha inicio de la tarea	Not null
fin	Date	Fecha fin de la tarea	Not null
titulo	Character(100)	Título de la tarea	Not null
texto	Character(1000)	Descripción de la tarea	
estado	Integer	Valor que identifica el estado de la tarea.	Not null, 1 = abierto 0 = cerrado

Tabla 6.20. Descripción de la tabla tarea

TABLA: usuarioTarea			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idUsuario	Integer	Código del usuario	Primary key, Foreign key (idUsuario de la tabla usuarioProyecto)
idProyecto	Integer	Código del proyecto	Primary key, Foreign key (idProyecto de la tabla usuarioProyecto)
idTarea	Integer	Código de la tarea	Primary key, Foreign key (idTarea de la tabla tarea)

Tabla 6.21. Descripción de la tabla usuarioTarea

TABLA: archivo			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idArchivo	Integer	Código del archivo	Primary key
nombre	Character(100)	Nombre del archivo	Not null
descripcion	Character(500)	Descripción del archivo	
extension	Character(300)	Extensión del archivo	
tipo	Character(200)	Tipo de archivo	
tamano	Numeric	Tamaño del archivo	
nombreArchivo	Character(200)	Nombre original del archivo	Not null

Tabla 6.22. Descripción de la tabla archivo

TABLA: usuarioArchivo			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idUsuario	Integer	Código del usuario	Primary key, Foreign key (idUsuario de la tabla usuarioProyecto)
idProyecto	Integer	Código del proyecto	Primary key, Foreign key (idProyecto de la tabla usuarioProyecto)
idArchivo	Integer	Código del archivo	Primary key, Foreign key (idArchivo de la tabla archivo)

Tabla 6.23. Descripción de la tabla usuarioArchivo

TABLA: partida			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idPartida	Character(100)	Código de la partida presupuestaria	Primary key
descripcion	Character(500)	Descripción de la partida presupuestaria	Not null

Tabla 6.24. Descripción de la tabla partida

TABLA: proyectoPartida			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idProyecto	Integer	Código del proyecto	Primary key, Foreign key (idProyecto de la tabla proyecto)
idPartida	Character(100)	Código de la partida presupuestaria	Primary key, Foreign key (idPartida de la tabla partida)
Valor	Numeric	Valor de la partida en el proyecto asignado.	Not null

Tabla 6.25. Descripción de la tabla proyectoPartida

TABLA: bien			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idBien	Integer	Código del bien	Primary key
idProyecto	Integer	Código del proyecto	Foreign key (idProyecto de la tabla proyectoPartida)
idPartida	Character(100)	Código de la partida presupuestaria	Foreign key (idPartida de la tabla proyectoPartida)
cantidad	Integer	Cantidad de bienes	Not null
detalle	Character(500)	Detalle del bien	Not null
valor	Numeric	Valor de cada bien	Not null
valorTotal	Numeric	Valor total de los bienes	Not null

Tabla 6.26. Descripción de la tabla bien

TABLA: registroActividad			
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idRegistroActividad	Integer	Código del registro de actividad	Primary key
idUsuario	Integer	Código del usuario	Foreign key (idUsuario de la tabla usuarioProyecto)
idProyecto	Integer	Código del proyecto	Foreign key (idProyecto de la tabla usuarioProyecto)
idTarea	Integer	Código de la tarea	
comentario	Character(500)	Comentario del registro de actividad	
dia	Date	día del registro de actividad	Not null
inicio	Time	Hora inicio del registro de actividad	
fin	Time	Hora fin del registro de actividad	
horas	Time	Cantidad de horas registradas	Default 00:00:00

Tabla 6.27. Descripción de la tabla registroActividad

TABLA: movimientoPartida			
Clave primaria:	idMovimiento		
CAMPO	TIPO	DESCRIPCIÓN	RESTRICCIÓN
idMovimiento	Integer	Código del movimiento de partidas	Primary key
valor	Numeric	Valor a ser movido entre las partidas	Not null
deLaPartida	Character(100)	Nombre de la partida restada	Not null
descripcionDeLa	Character(500)	Descripción de la partida restada	Not null
valorDeLa	Numeric	valor actual de la partida restada	Not null
valorRestado	Numeric	valor resultante de la partida restada	Not null
aLaPartida	Character(100)	Nombre de la partida sumada	Not null
descripcionALa	Character(500)	Descripción de la partida sumada	Not null
valorALa	Numeric	Valor actual de la partida sumada	Not null
valorSumado	Numeric	Valor resultante de la partida sumada	Not null

fechaMovimiento	Date	Fecha del movimiento de partidas	Not null
justificacion	Character(500)	Justificación del movimiento de partidas	Not null
idProyecto	Integer	Código del proyecto	Not null
nombreProyecto	Character(200)	Nombre del proyecto	Not null

Tabla 6.28. Descripción de la tabla movimientoPartida

6.9.2.3 Diseño de la interfaz

Para el diseño de la interfaz del sistema de gestión de proyectos se ha tomado como referencia los requerimientos del departamento UOCENI – ISEI de que ésta deba ser sencilla y amigable con el usuario.

Pantalla inicio de sesión

En esta página web la persona que desee entrar al sistema de gestión de proyectos deberá ingresar su usuario y contraseña obtenidos previamente en su registro, estos datos serán validados y permitirán su acceso con los respectivos permisos.

Uoceni - Isei
Gestión Proyectos

Usuario:

Contraseña:

Ingresar

Mensaje Error

1

2

3

4

Figura 6.37. *Pantalla inicio de sesión*

1. Caja de texto que permite el ingreso del nombre de usuario
2. Caja de texto que permite el ingreso de la contraseña
3. Botón ingresar: valida los parámetros ingresados en las cajas de texto y posteriormente si los datos son correctos redirecciona al Sistema de Gestión de Proyectos.
4. Etiqueta Mensaje Error: permite la visualización de los errores cometidos en la página web.

6.9.2.3.1 Diseño de salidas

En esta fase se procedió a diseñar las interfaces necesarias que permitirán al sistema informático entregar información a sus usuarios.

Página principal del sistema de gestión de proyectos

Cuando el usuario acaba de iniciar su sesión se le presenta la siguiente página maestra con los menús correspondientes:

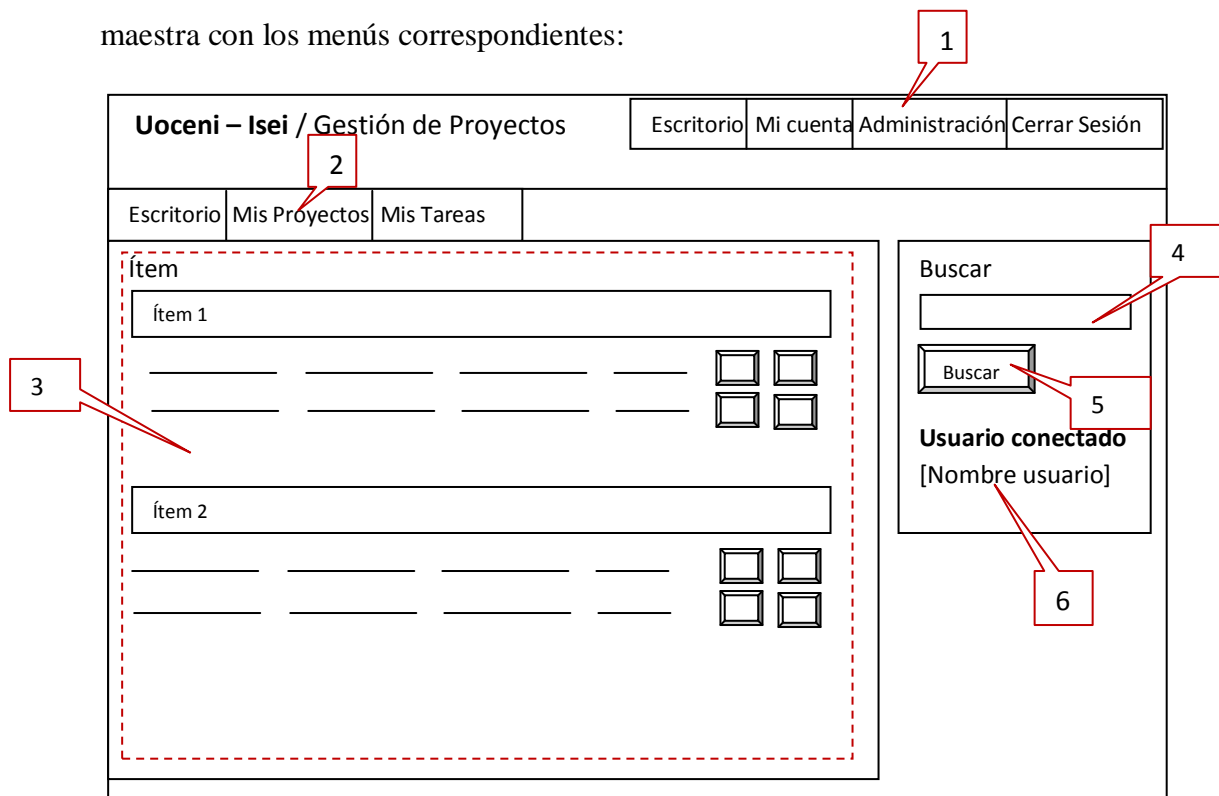


Figura 6.38. *Página inicial*

1. Menú principal. Se podrá escoger 4 opciones: Escritorio, Mi cuenta, Administración (solo usuarios administradores), Cerrar sesión.
2. Menú escritorio. Resultante de la selección “Escritorio” del menú principal, en éste se podrá escoger 3 opciones: Escritorio, Mis proyectos, Mis tareas.
 - a. Escritorio. Muestra información de los proyectos y tareas del usuario conectado.
 - b. Mis proyectos. Muestra información de los proyectos del usuario conectado.

- c. Mis tareas. Muestra información de las tareas del usuario conectado.
3. Página contenido. Espacio destinado para la gestión de información. La información mostrada dependerá directamente de las opciones marcadas en los menús.
 4. Caja de texto buscar. Se podrá ingresar texto para su respectiva búsqueda
 5. Botón buscar. Ejecuta la búsqueda del parámetro ingresado anteriormente en la caja de texto buscar
 6. Etiqueta Nombre usuario. Muestra el nombre y apellido del usuario conectado.

Para la gestión de cada uno de los proyectos el sistema cuenta con:

Uoceni – Isei / Gestión de Proyectos							Escritorio	Mi cuenta	Administración	Cerrar Sesión
Proyecto	Objetivos	Lista de tareas	Archivos	Usuario	R. Actividad	Presupuesto	Buscar <input type="text"/> <input type="button" value="Buscar"/> Usuario conectado [Nombre usuario]			
Ítem										

Figura 6.39. *Menú proyecto*

1. Menú proyecto. Resultante de la selección de un proyecto, en éste se podrá escoger 7 opciones: Proyecto, Objetivos, Lista de tareas, Archivos, Usuario, R. Actividad, Presupuesto.

- a. Proyecto. Muestra información del proyecto seleccionado con anterioridad.
- b. Objetivos. Muestra información de los objetivos del proyecto seleccionado con anterioridad.
- c. Lista de tareas. Muestra una lista de tareas del proyecto seleccionado con anterioridad.
- d. Archivos. Muestra información de los archivos del proyecto seleccionado con anterioridad.
- e. R. Actividad. Muestra información de las actividades realizadas por parte del personal del proyecto seleccionado con anterioridad.
- f. Presupuesto. Muestra información del presupuesto del proyecto seleccionado con anterioridad.

Para que cada usuario pueda gestionar su información, el sistema cuenta con:

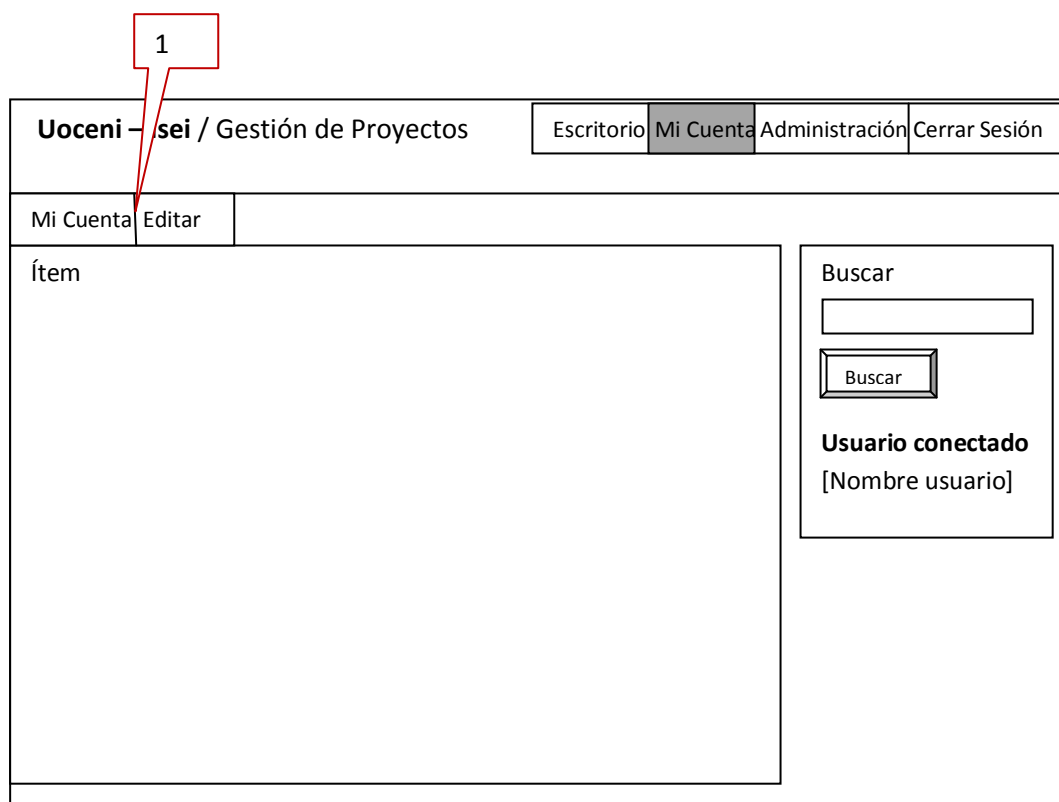


Figura 6.40. *Menú perfil*

1. Menú perfil. Resultante de la selección “Mi cuenta” del menú principal, en éste se podrá escoger 2 opciones: Mi Cuenta, Editar.

- a. Mi Cuenta. Muestra información detallada del usuario conectado
- b. Editar. Se podrá actualizar la información del usuario conectado

Para la administración total del sistema de gestión de proyectos, se cuenta con:

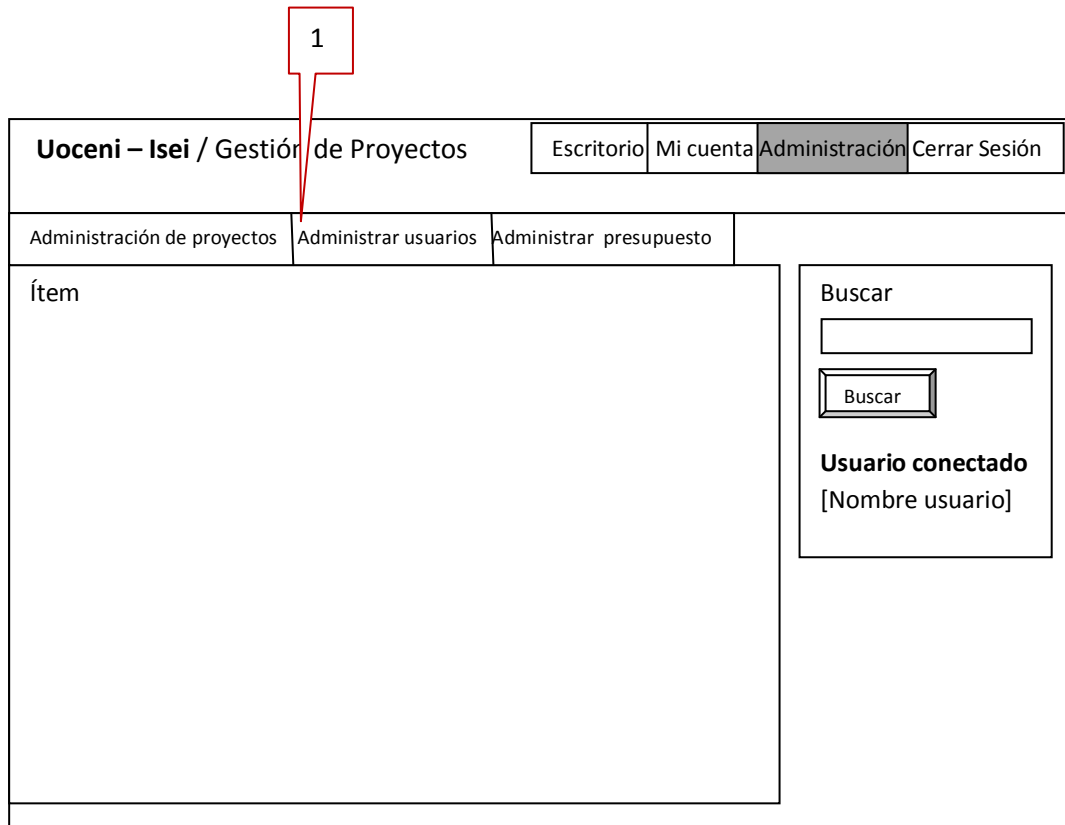


Figura 6.41. *Menú administración*

1. Menú administración: Resultante de la selección “Administración” del menú principal, en éste se podrá escoger 3 opciones: Administración de proyectos, Administrar usuarios, Administrar presupuesto.
 - a. Administración de proyectos. Muestra información de todos los proyectos ingresados en el sistema.
 - b. Administrar usuarios. Muestra información de todos los usuarios registrados en el sistema.
 - c. Administrar presupuesto. Muestra información de todas las partidas presupuestarias.

Nota: Para poder visualizar la opción “Administración” y el menú “Menú Administración” el usuario debe tener permisos de administrador.

Información traída de la Base de Datos presentada en pantalla

Esta es la descripción general para las páginas que presentan información traída de la Base de Datos en pantalla:

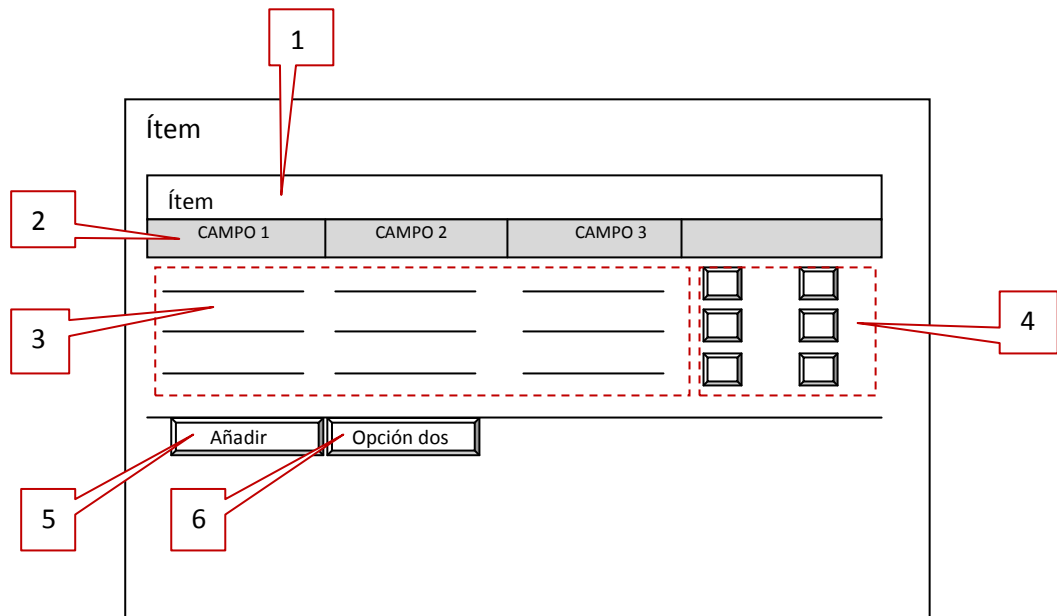


Figura 6.42. Información presentada en pantalla.

1. Barra de encabezado. Permite visualizar el título o referencia de la información a manipular.
2. Barra de campos. Permite visualizar el nombre de los campos de la información a manipular.
3. Espacio de información. Permite visualizar la información traída de la Base de Datos.
4. Espacio de manipulación. Permite ejecutar operaciones con la información traída de la Base de Datos como: Editar, Eliminar.
 - a. Eliminar. Elimina el registro seleccionado
 - b. Editar. Envía la información del registro seleccionado a otra página web para su modificación
5. Botón añadir. Visualiza una sección oculta destinada para la adición de información.

6. Botón Opción dos. Puede ser utilizado para filtrar información o para visualizar información de proyectos, objetivos y tareas terminadas.

Generación de reportes

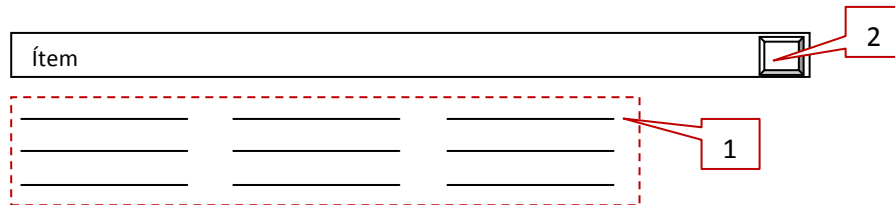


Figura 6.43. *Generación de reportes*

1. Espacio de información. Permite visualizar la información traída de la Base de Datos.
2. Botón exportar. Genera un archivo PDF con la información ubicada en el “Espacio de información”

A continuación se muestra los formatos de reportes tipo PDF que generará el sistema.

Formato de reporte tipo PDF de “información de usuario”

UNIVERSIDAD TÉCNICA DE AMBATO		
PERFIL DE USUARIO:		
Nombres: xxxxxx xxxxxx		
Correo electrónico: xxxxxxxx@xxx.xxx		
Teléfono: xxxxxx		
Móvil: xxxxxx		
Dirección: xxxxxx		
Código postal / Ciudad: xxxxxxxxxxxx		
Provincia: xxxxxx		
Sexo: xxxxxx		
Role: xxxxxx		
PROYECTOS:		
Nombre	Inicio	Fin
XXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXX	XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXX	XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXX	XXXXX

Figura 6.44. *Reporte información de usuario*

Formato de reporte tipo PDF de “registro de actividades”

UNIVERSIDAD TÉCNICA DE AMBATO					
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL					
PROYECTO DE INVESTIGACIÓN: xxxxxxxxxxxxxx					
FECHA	DESCRIPCIÓN	NOMBRES	ENTRADA	SALIDA	FIRMA
xxxxxxx	xxxxxxxxxxxxxxxxxxx	xxxxxxxxxxx	xxxxxxx	xxxxxxx	_____
xxxxxxx	xxxxxxxxxxxxxxxxxxx	xxxxxxxxxxx	xxxxxxx	xxxxxxx	_____
xxxxxxx	xxxxxxxxxxxxxxxxxxx	xxxxxxxxxxx	xxxxxxx	xxxxxxx	_____
xxxxxxx	xxxxxxxxxxxxxxxxxxx	xxxxxxxxxxx	xxxxxxx	xxxxxxx	_____

Figura 6.45. *Reporte registro de actividades*

6.9.2.3.2 Diseño de entradas

En esta fase se procedió a diseñar las interfaces que permitirán al usuario ingresar información al sistema informático para su posterior tratamiento.

La pantalla de adición de información se utilizará para: Proyectos, Objetivos, Lista de tareas, Tareas, Archivos, Actividades, Usuarios, Presupuesto.

Descripción general para la adición de información.

El diagrama muestra una interfaz de usuario con el título "Añadir ítem". Contiene cuatro campos de entrada etiquetados como "Campo 1", "Campo 2", "Campo 3" y "Campo 4". Debajo de los campos hay dos botones: "Enviar" y "Cancelar". Hay tres marcadores numerados: "1" apunta a los campos de entrada, "2" apunta al botón "Enviar" y "3" apunta al botón "Cancelar".

Figura 6.47. Adición de información

1. Campos de ingreso. Para los campos existen reglas para poder almacenar la información en la base de datos y además se puede utilizar diferentes controles para cada campo según se lo requiera.
2. Botón Guardar. Guarda la información ingresada siempre y cuando no existan errores en la validación de cada campo.
3. Botón Cancelar. Cancela el ingreso.

La pantalla de actualización de información se utilizará para: Proyectos, Objetivos, Lista de tareas, Tareas, Archivos, Actividades, Usuarios, Presupuesto.

Descripción general para la actualización de información.

El diagrama muestra una interfaz de usuario titulada "Editar Ítem". Dentro de un recuadro, se encuentran cuatro campos de texto etiquetados como "Campo 1", "Campo 2", "Campo 3" y "Campo 4". Cada campo contiene una línea de texto representada por "XXXXXXXXXXXX". Una línea punteada roja agrupa los cuatro campos. Debajo de los campos, hay dos botones: "Enviar" y "Cancelar". Tres recuadros numerados (1, 2 y 3) con líneas de conexión indican: 1 apunta a los campos de texto, 2 apunta al botón "Enviar" y 3 apunta al botón "Cancelar".

Figura 6.48. Actualización de información

1. Campos de ingreso. Para los campos existen reglas para poder almacenar la información en la base de datos y además se puede utilizar diferentes controles para cada campo según se lo requiera.
2. Botón Guardar. Guarda la información ingresada siempre y cuando no existan errores en la validación de cada campo.
3. Botón Cancelar. Cancela el ingreso.

La pantalla de filtración de información se utilizará para: actividades y el presupuesto de cada proyecto.

Descripción general para la filtración de información.

Figura 6.49. *Filtración de información*

1. Campos de ingreso. Para los campos existen reglas para poder filtrar la información de la base de datos y además se puede utilizar diferentes controles para cada campo según se lo requiera.
2. Botón Filtrar. Filtra la información de una o varias tablas de la Base de Datos siempre y cuando no existan errores en la validación de cada campo.
3. Botón Cancelar. Cancela la filtración de información.

6.9.2.4 Diseño de Comunicación

El sistema a desarrollar utilizará la arquitectura cliente-servidor, razón por la cual se necesitará de una infraestructura de red que le permita compartir sus recursos en una intranet o en la Internet.

A continuación se muestra un bosquejo general de la infraestructura de red en la UOCENI – ISEI.

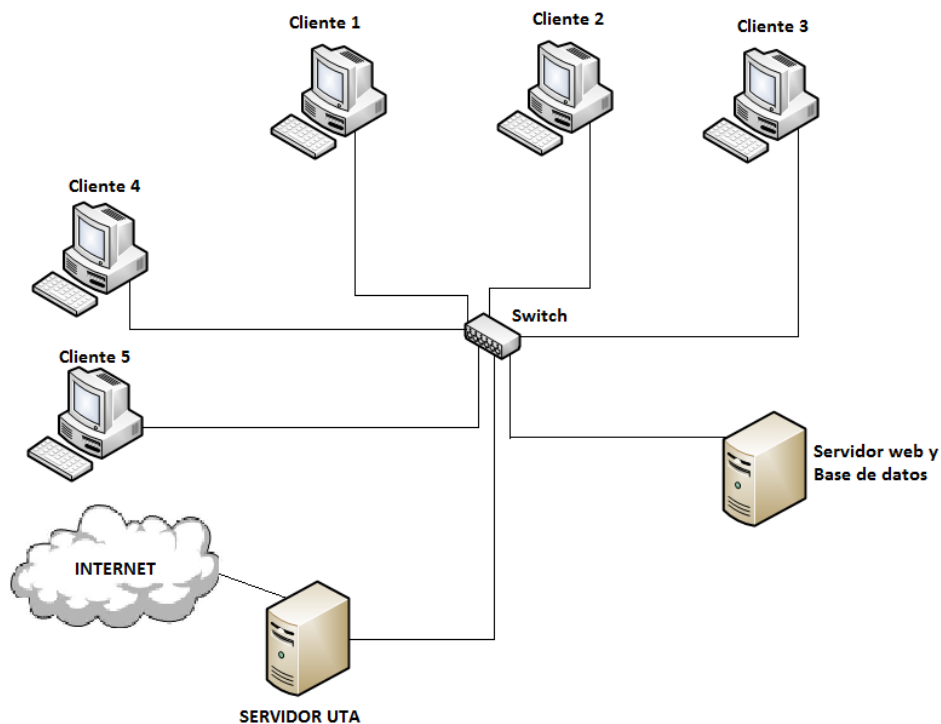


Figura 6.50. *Diseño de Comunicación*

6.9.3 Implementación

6.9.3.1 Extracto de código fuente

```

ConectarBD.cs

public static class ConectarBD
{
    #region Datos
    private static string _cadenaDeConexion;

    #endregion

    // Propiedad para manejar la variable: _cadenaDeConexion

    public static string CadenaDeConexion
    {
        get
        {
            //Devuelve
            return _cadenaDeConexion;
        }

        set
        {
            // Obtiene
            _cadenaDeConexion = value;
        }
    }
}

```

Código para iniciar sesión

```
Default.aspx.cs

//Almacenamos la cadena de conexión en la variable _cadenaDeConexión de la
//clase ConectarBD
ConectarBD.CadenaDeConexion= "Server=localhost;Port=5432; Database=gestion_proyectos;
                               User ID= postgres;Password=sa";

//Verificamos que los campos de inicio de sesión no estén vacíos
if(txtUsuario.Text!="" && txtContrasena.Text!="")
{
    try
    {
        //Declaramos e Instanciamos una variable de tipo DataBaseHelper.

        //DataBaseHelper es la clase que contiene métodos, propiedades, enumeraciones, que permiten
        //manipular la BD.
        DatabaseHelper dbhIngresar = new DatabaseHelper(ConectarBD.CadenaDeConexion,
                                                         Providers.PostgresSQL);
        // Añadimos parámetros para comparar con la Base de datos

        dbhIngresar.AddParameter("@login",txtUsuario.Text.Trim());
        dbhIngresar.AddParameter("@contrasena",txtContrasena.Text.Trim());

        //Declaramos una variable DataTable, que almacenara la información traída de la BD.

        DataTable dtUsuario = new DataTable();
        dtUsuario= dbhIngresar.ExecuteDataSet(sentenciaSQL).Tables[0];

        //Comparamos si existe algún usuario con login: txtUsuario y contraseña: txtContrasena.

        if(dtUsuario.Rows.Count == 1)
        {
            Session["usuario"] = dtUsuario.Rows[0]["Login"].ToString().Trim();

            Session["contrasena"] = dtUsuario.Rows[0]["contrasena"].ToString().Trim();

            Session["idRole"] = Convert.ToInt32(dtUsuario.Rows[0]["idRole"]);

            Session["administrador"] = Convert.ToInt32(dtUsuario.Rows[0]["administrador"]);

            Response.Redirect("~/webForms/escritorio.aspx");

        }
        else
        {
            lblMensaje.Text = "No ha sido posible iniciar la sesión.<br/>Por favor, comprueba tus datos
                               de acceso";

        }
    }
}
```

```
catch(NpgsqlException error)
{
    ControlExcepciones.TratarErroresWindows(error);

    lblMensaje.Text=ControlExcepciones.Mensaje;
}

catch(Exception error)
{
    ControlExcepciones.TratarErroresWindows(error);

    lblMensaje.Text=ControlExcepciones.Mensaje;
}
}
else
    lblMensaje.Text="No ha sido posible iniciar la sesión.<br/>Todos los campos son requeridos";
```

Código para traer datos de los proyectos

MisProyectos.aspx.cs

```
protected DataTable traerDatosProyectoAbiertos ()
{
    //Declaramos e Instanciamos una variable de tipo DataBaseHelper.

    DatabaseHelper dbhProyectos = new DatabaseHelper(ConectarBD.CadenaDeConexion,
                                                    Providers.PostgresSQL);

    try
    {
        //Declaramos instanciamos una variable de tipo DataTable
        DataTable dtValor = new DataTable();

        //Estado
        //0 -----> Cerrado
        //1 -----> Activo
        // Sentencia SQL para seleccionar los proyectos
        string seleccion = "SELECT p.idProyecto, p.nombre, p.descripcion, p.presupuesto, "+
                           "p.fin-current_date as tiempoRestante, p.estado, p.avance " +
                           "FROM proyecto p, usuarioProyecto up " +
                           "WHERE p.idProyecto = up.idProyecto " +
                           "and up.idUsuario = @idUsuario and p.estado=1 "+
                           "ORDER BY p.idProyecto DESC;";

        dbhProyectos.AddParameter("@idUsuario", Convert.ToInt32(Session["idUsuario"]));

        dtValor = dbhProyectos.ExecuteDataSet(seleccion).Tables[0];

        // retornar una tabla con información de los proyectos
        return dtValor;

    }
    catch (NpgsqlException error)
    {
        throw error;
    }
    catch (Exception error)
    {
        throw error;
    }
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        try
        {
            // proyectos Abiertos
            gvProyectos.DataSource = traerDatosProyectoAbiertos();
            gvProyectos.DataBind();
        }
        catch (NpgsqlException error)
        {
            //Utilizamos la clase llamada ControlExcepciones para tratar los errores de BD.
            ControlExcepciones.TratarErroresWindows(error);
            lblMensaje.Text = ControlExcepciones.Mensaje;
        }
    }
}
```

```

        catch (Exception error)
        {
            //Utilizamos la clase llamada ControlExcepciones para tratar los errores de CSharp.
            ControlExcepciones.TratarErroresWindows(error);
            lblMensaje.Text = ControlExcepciones.Mensaje;
        }
    }
}

```

Código para ingresar datos de los usuarios

```

administracionUsuarios.aspx.cs

protected void btnEnviar_click(object sender, EventArgs e)
{
    bool requeridos = datosRequeridos();
    DatabaseHelper dbhInsertarUsuario = new DatabaseHelper(ConectarBD.CadenaDeConexion,
        Providers.PostgresSQL);

    if(requeridos==true)
    {
        bool validacion= validacionDatos();

        if(validacion==true)
        {
            try
            {
                string sentenciaInsert = "INSERT INTO usuario (cedula, nombres, login, contrasena, "+
                    "salario, idRole) " +
                    "VALUES (@cedula, @nombres, @login, @contrasena, "+
                    "@salario, @idRole)";

                dbhInsertarUsuario.AddParameter("@cedula",txtCedula.Text.Trim());

                dbhInsertarUsuario.AddParameter("@nombres",txtNombres.Text.Trim());

                dbhInsertarUsuario.AddParameter("@login",txtUsuario.Text.Trim());

                dbhInsertarUsuario.AddParameter("@contrasena",txtContrasena.Text.Trim());

                if(txtSalario.Text.Trim()!="")
                    dbhInsertarUsuario.AddParameter("@salario",
                        Convert.ToDecimal(txtSalario.Text.Trim()));

                else
                    dbhInsertarUsuario.AddParameter("@salario",0);

                dbhInsertarUsuario.AddParameter("@idRole",rblRol.SelectedValue.ToString());

                int seInserto = dbhInsertarUsuario.ExecuteNonQuery(sentenciaInsert);

                if (seInserto == 0)
                {
                    lblMensajeError.Text = "No ha sido posible insertar los datos, verifique la información
                        ingresada";
                }
            }
            else
            {

```

```

        Response.Redirect("~/webForms/administracionUsuarios.aspx");
    }
}
catch (NpgsqlException error)
{
    ControlExcepciones.TratarErroresWindows(error);
    lblMensajeError.Text = ControlExcepciones.Mensaje;
}

catch (Exception error)
{
    ControlExcepciones.TratarErroresWindows(error);
    lblMensajeError.Text = ControlExcepciones.Mensaje;
}
}
}
else
    lblMensajeError.Text = " * Campos requeridos";
}

protected bool datosRequeridos()
{
    bool valor = false;
    int contador = 0;

    if (txtCedula.Text.Trim() != "")
        contador += 1;
    else
        lblRequeridoCedula.Text = " *";

    if (txtNombres.Text.Trim() != "")
        contador += 1;
    else
        lblRequeridoNombres.Text = " *";

    if (txtUsuario.Text.Trim() != "")
        contador += 1;
    else
        lblRequeridoUsuario.Text = " *";

    if (txtContrasena.Text.Trim() != "")
        contador += 1;
    else
        lblRequeridoContrasena.Text = " *";

    if (contador == 4)
        valor = true;

    return valor;
}

protected bool validacionDatos()
{
    int contadorValidos = 0;
    bool EsValido = false;

    // Validación cédula

```

```

string cedula = txtCedula.Text.Trim();

if (cedula.Length == 10)
{
    int x = 0;
    int x1 = 0;
    int suma = 0;
    int subTotal = 0;
    int respuesta = 0;
    int valor;

    for (int i = 1; i <= 9; i++)
    {
        x = i % 2;
        valor = (cedula[i - 1]) - 48;

        if (x != 0)
        {
            x1 = valor * 2;

            if (x1 > 9)
            {
                x1 = x1 - 9;
            }
        }
        else
        {
            x1 = valor;
        }

        suma = suma + x1;
    }

    subTotal = ((suma / 10) + 1) * 10;
    respuesta = subTotal - suma;

    if (respuesta == 10)
    {
        respuesta = 0;
    }

    if (respuesta == cedula[9] - 48)
    {
        // cedula correcta
        //aumenta los controles validos.
        contadorValidos += 1;
    }
    else
    {
        lblCedula.Text = " Cédula incorrecta";
    }
}
else
    lblCedula.Text = " Cédula incorrecta";

//Nombres

bool nombres = soloLetras(txtNombres.Text.Trim());

```

```

        if (nombres == false)
        {
            lblNombres.Text = " Solo letras";
        }
        else
            contadorValidos += 1;

// Salario
        bool salario = validarSalario(txtSalario.Text.Trim());

        if (salario == false && txtSalario.Text.Trim()!="")
        {
            lblSalario.Text = " Salario incorrecto";
        }
        else
            contadorValidos += 1;

// Contraseña
        if(txtContrasena.Text.Length<6)
        {
            lblContrasena.Text="La contraseña debe tener una longitud de 6 caracteres como
            mínimo.<br/> Escriba otra contraseña.";
        }
        else
            contadorValidos += 1;

//Contar válidos

        if (contadorValidos == 4)
        {
            EsValido = true;
        }
        return EsValido;
    }

protected bool soloLetras(string letras)
{
    int contadorError = 0;
    for (int i = 0; i < letras.Length; i++)
    {
        if (((Convert.ToInt32(letras[i]) >= 65) && (Convert.ToInt32(letras[i]) <= 90)) || /*a --- z*/
            ((Convert.ToInt32(letras[i]) >= 97) && (Convert.ToInt32(letras[i]) <= 122)) || /*A----Z*/
            (Convert.ToInt32(letras[i])==32) || /*espacio*/
            (Convert.ToInt32(letras[i])==225) || /*á*/
            (Convert.ToInt32(letras[i])==233) || /*é*/
            (Convert.ToInt32(letras[i])==243) || /*ó*/
            (Convert.ToInt32(letras[i])==237) || /*í*/
            (Convert.ToInt32(letras[i])==250) || /*ú*/
            (Convert.ToInt32(letras[i])==193) || /*Á*/
            (Convert.ToInt32(letras[i])==201) || /*É*/
            (Convert.ToInt32(letras[i])==205) || /*Í*/
            (Convert.ToInt32(letras[i])==211) || /*Ó*/
            (Convert.ToInt32(letras[i])==218) || /*Ú*/
            (Convert.ToInt32(letras[i])==241) || /*ñ*/
            (Convert.ToInt32(letras[i])==209) /*Ñ*/
            )
        {
            contadorError++;
        }
    }
}

```



```

if (contadorError == letras.Length)
    return true;
else
    return false;
}

protected bool validarSalario(string numero)
{
    int contadorError = 0;

    for (int i = 0; i < numero.Length; i++)
    {
        // 48 ----> 0
        // 57 ----> 9

        if (((Convert.ToInt32(numero[i]) >= 48) && (Convert.ToInt32(numero[i]) <= 57)) ||
            (Convert.ToInt32(numero[i]) == 46) || /* */
            (Convert.ToInt32(numero[i]) == 44) /* */)
        {
            contadorError++;
        }
    }

    if (contadorError == numero.Length)
        return true;
    else
        return false;
}

```

6.9.4 Pruebas

El proceso de pruebas es básicamente una etapa destinada a la identificación de posibles fallos de implementación y calidad del software.

A continuación se detalla cada una de las pruebas aplicadas en el sistema implementado:

6.9.4.1 Pruebas de caja negra

En este tipo de pruebas no se considera la codificación dentro de sus parámetros a evaluar, es decir, no está basado en el conocimiento interno del sistema. Estas pruebas se enfocan en la funcionalidad y requerimientos especificados del software.

Una vez aplicado este tipo de pruebas en el sistema, como conclusión se pudo determinar qué:

- Las funciones que realiza el sistema son claras y precisas.
- El ingreso al sistema es fácil y sencillo
- La seguridad de acceso y permisos de usuario se cumplen a cabalidad.
- Las interfaces del sistema no permiten confusión en su manipulación.
- Se mantiene la integridad del sistema a lo largo de su utilización.

6.9.4.2 Pruebas de caja blanca

Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como está definido.

¿Porqué usar pruebas de caja blanca?

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa.
- A veces creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando puede hacerlo de forma normal.

6.9.4.2.1 Prueba del camino básico

El método del camino básico (propuesto por McCabe) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

A continuación se muestra un ejemplo de la aplicación de esta prueba:

Grafo de flujo

Validar solo números

protected bool soloNumeros(string numero) {	0
int contadorError = 0; bool variableRetorno = false;	1
for (int i = 0; i < numero.Length; i++) {	2
if (Convert.ToInt32(numero[i]) >= 48) {	3
if (Convert.ToInt32(numero[i]) <= 57) {	4
contadorError++;	5
}	6
}	7
}	8
if (contadorError == numero.Length) {	9
variableRetorno = true;	10
}	11
return variableRetorno;	12
}	13

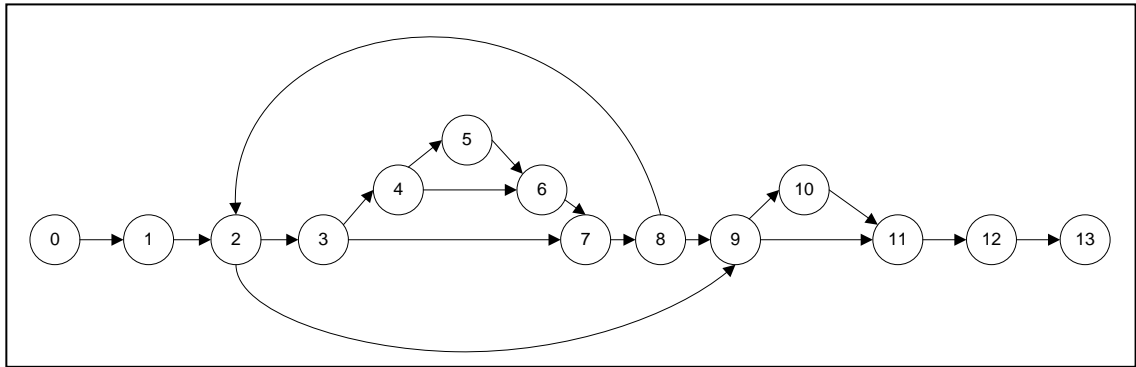


Figura 6.51. Grafo de validación solo números

Complejidad Ciclomática

$$V(G) = A \text{ (aristas)} - N \text{ (nodos)} + 2$$

$$V(G) = 18 - 14 + 2$$

$$V(G) = 6$$

Camino básicos

1) 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 2 - 9 - 10 - 11 - 12 - 13

2) 0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 2 - 9 - 11 - 12 - 13

3) 0 - 1 - 2 - 3 - 4 - 6 - 7 - 8 - 2 - 9 - 10 - 11 - 12 - 13

4) 0 - 1 - 2 - 3 - 4 - 6 - 7 - 8 - 2 - 9 - 11 - 12 - 13

5) 0 - 1 - 2 - 3 - 7 - 8 - 2 - 9 - 10 - 11 - 12 - 13

6) 0 - 1 - 2 - 3 - 7 - 8 - 2 - 9 - 11 - 12 - 13

Comprobación

Se tomará el camino básico #1 para la comprobación:

La función recibe un parámetro de tipo texto "numero" **nodo 0**

La variable "contadorError" recibe el valor "0 (cero)" y la variable "variableRetorno" recibe el valor "false" **nodo 1**

A continuación se ingresa al ciclo for, el cual se repetirá mientras que su contador "i" que empieza en "cero" sea menor o igual a la longitud del texto "numero", en cada una de las repeticiones se comparará que "numero[i]" sea número, sumando

el contador “contadorError” por todos los caracteres números encontrados. **Nodo 2 – 3 – 4 – 5 – 6 – 7 – 8.**

Posteriormente, se comprobara si “contadorError” es igual a la longitud del texto “numero”, en el caso verdadero “variableRetorno” tomara el valor “true” **Nodo: 9 – 10 – 11.**

Por último, la función retornará “variableRetorno” con el valor “true”. **Nodo 12 - 13**

6.9.5 Implantación

El proceso de implantación de un sistema informático es uno de los últimos pasos en el ciclo de vida del software, y el producto de largas horas de trabajo. Esta etapa comprende además de la instalación del sistema en sí, la conversión del sistema antiguo al nuevo.

6.9.5.1 Conversión del sistema antiguo al nuevo

Para lograr la conversión del sistema antiguo al nuevo se procedió a ejecutar las siguientes tareas:

- Conversión de datos.
- Capacitación al personal sobre el uso y administración del nuevo sistema.

6.9.5.1.1 Conversión de datos

Por decisión del Director del departamento, que se ingrese al nuevo sistema únicamente los datos de los proyectos que iniciaren después de la fecha de implantación del mismo, no hubo la necesidad de realizar la conversión de los datos del sistema antiguo al nuevo. Decisión tomada en base a que el nuevo sistema fue diseñado con la finalidad de permitir un adecuado control y seguimiento de los proyectos que están en fase de desarrollo, más no de los proyectos que ya han sido culminados.

6.9.5.1.2 Capacitación al personal sobre el uso y administración del nuevo sistema.

A pesar de que el sistema informático fue realizado a medida y existió una constante comunicación con el personal del departamento en cada fase de desarrollo del software, se realizó una charla de capacitación al director sobre el funcionamiento y administración del mismo, y con juntamente con él se ejecutó una conferencia donde se informó al resto del personal las bondades, características y la forma correcta de utilización del sistema informático, con la finalidad de aprovechar al máximo los recursos que éste nos ofrece.

Otra forma de preparación y guía para la utilización del sistema informático fue la elaboración del manual de usuario, el mismo que permitirá resolver dudas que se presenten en el futuro, sin la presencia del desarrollador del software.

6.9.5.2 Instalación y Configuración

Como en el departamento UOCENI – ISEI se cuenta con la infraestructura necesaria para el alojamiento del sitio web no hubo la necesidad de comprar equipamiento pero si se requirió de la instalación y configuración del software que a continuación se lo describe.

El servidor del departamento trabaja con el sistema operativo CentOS 5.4, en el que se encuentra instalado y configurado como servidor web a Apache que es el que alojará el sitio.

Para poder ejecutar aplicaciones ASP.NET en Linux el servidor web Apache necesita del módulo mod_mono y componentes de mono XSP (Motor de Asp.Net hosting).

6.9.5.2.1 Instalación Mod_mono y XSP

6.9.5.2.1.1 Requisitos

Previo la instalación de Mod_mono y XSP se instaló los siguientes paquetes:

gcc-c++: Es un compilador integrado del proyecto GNU para C++

```
yum install gcc-c++
```

bison: Es un programa generador de analizadores sintácticos de propósito general perteneciente al proyecto GNU.

```
yum install bison
```

httpd-devel: Si va a instalar el servidor HTTP Apache y desea ser capaz de compilar o desarrollar módulos adicionales para Apache, es necesario instalar este paquete.

```
yum install httpd-devel
```

glib2-devel: Gnome library, proporciona los bloques básicos para construir aplicaciones y bibliotecas escritas en C

```
yum install glib2-devel
```

6.9.5.2.1.2 Instalación

Descargamos los siguientes paquetes:

```
http://ftp.novell.com/pub/mono/sources/mono/mono-2.0.1.tar.bz2
```

```
http://ftp.novell.com/pub/mono/sources/xsp/xsp-2.0.tar.bz2
```

```
http://ftp.novell.com/pub/mono/sources/mod_mono/mod_mono-2.0.tar.bz2
```

A continuación instalamos el núcleo mono

```
tar -vxjf mono-2.0.1.tar.bz2
cd mono-2.0.1
./configure
make
make install
cd ..
```

Luego viene XSP

```
tar -vxjf xsp-2.0.tar.bz
cd xsp-2.0
./configure
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
make
make install
cd ..
```

Finalmente instalamos mod_mono

```
tar -vxjf mod_mono-2.0.tar.bz2
cd mod_mono-2.0
./configure
make
make install
```


6.9.5.2.1.3 Configuración

Para que se pueda reconocer y aprovechar las nuevas funcionalidades de apache se procedió a añadir el siguiente código en el archivo httpd.conf

```
<IfModule !mod_mono.c>
  LoadModule mono_module /usr/lib/httpd/modules/mod_mono.so
  AddType application/x-asp-net .aspx
  AddType application/x-asp-net .asmx
  AddType application/x-asp-net .ashx
  AddType application/x-asp-net .asax
  AddType application/x-asp-net .ascx
  AddType application/x-asp-net .soap
  AddType application/x-asp-net .rem
  AddType application/x-asp-net .axd
  AddType application/x-asp-net .cs
  AddType application/x-asp-net .config
  AddType application/x-asp-net .Config
  AddType application/x-asp-net .dll
  DirectoryIndex index.aspx
  DirectoryIndex Default.aspx
  DirectoryIndex default.aspx
</IfModule>
```

6.9.5.2.2 Instalación del motor de base de datos

El sistema de gestión de proyectos trabaja con PostgreSQL para la parte de Base de datos motivo por el cual se procedió a instalarlo en el servidor del departamento.

Como primer paso se efectuó la descarga del paquete llamado:

```
postgresql-9.0.2-1-linux.bin
```

Luego se procedió a realizar su ejecución desde consola como super-usuario.

```
./postgresql-9.0.2-1-linux.bin
```

Una vez ejecutado el paquete se generó un asistente gráfico para seguir con la instalación



Figura 6.52. *Instalar PostgreSQL*

En este paso seleccionamos el directorio de instalación de PostgreSQL



Figura 6.53. *Directorio de instalación*

A continuación seleccionamos el directorio dentro del cual se almacenarán los datos

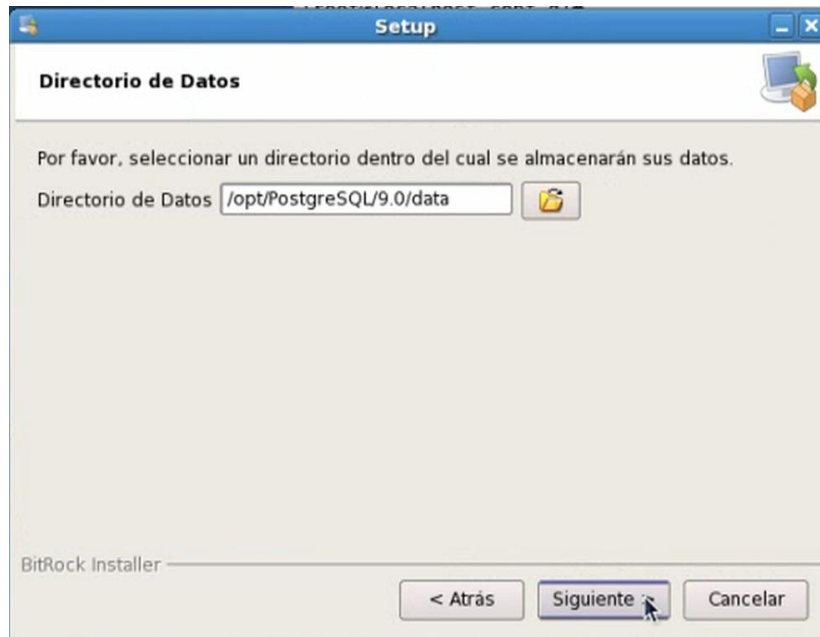


Figura 6.54. Directorio de datos

Como paso siguiente se procedió a ingresar la contraseña del usuario postgres.
Contraseña = "sa"

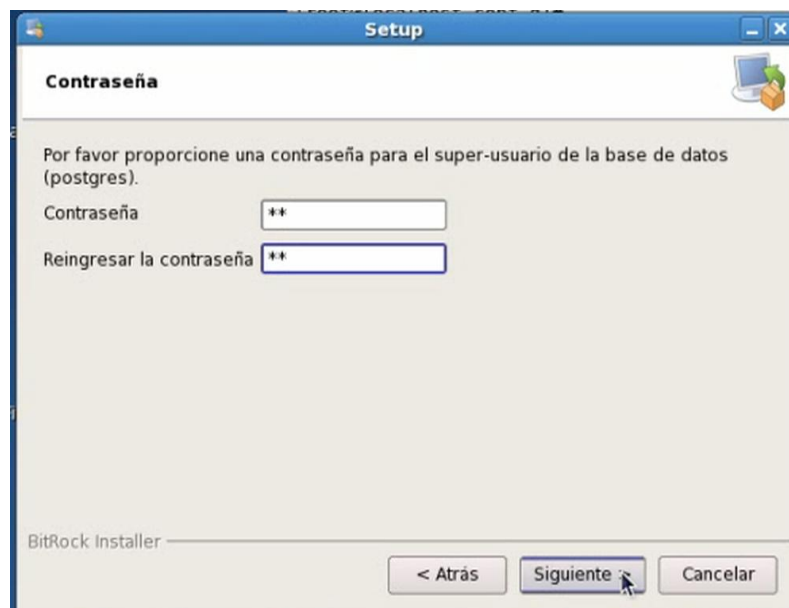


Figura 6.55. Contraseña del usuario postgres

En este paso ingresamos el número de puerto en el que el servidor debería escuchar

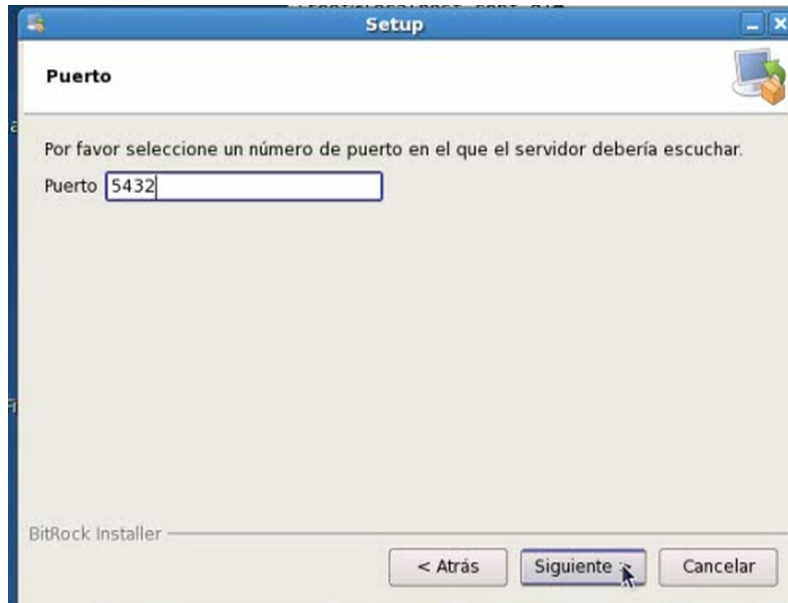


Figura 6.56. *Puerto*

Configuramos el idioma



Figura 6.57. *Opciones Avanzadas*

Como último paso ejecutamos la instalación en el servidor

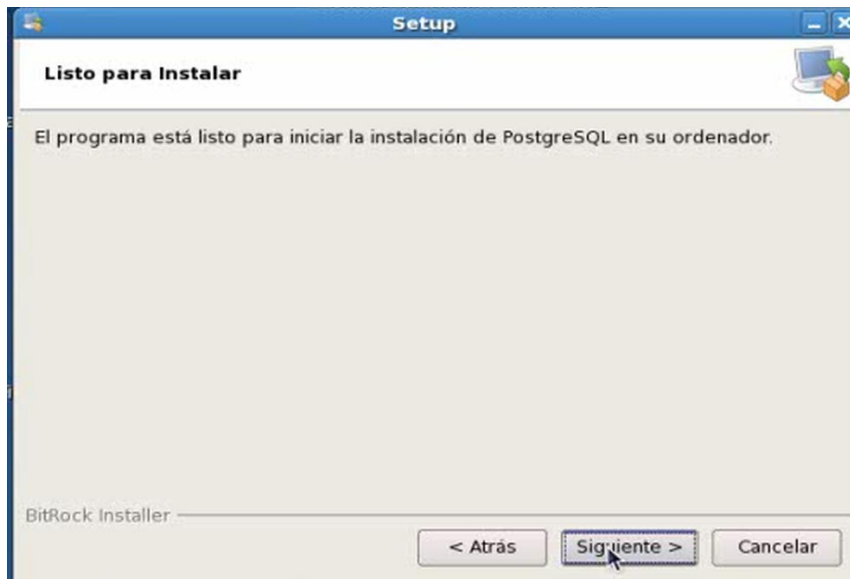


Figura 6.58. *Listo para instalar*

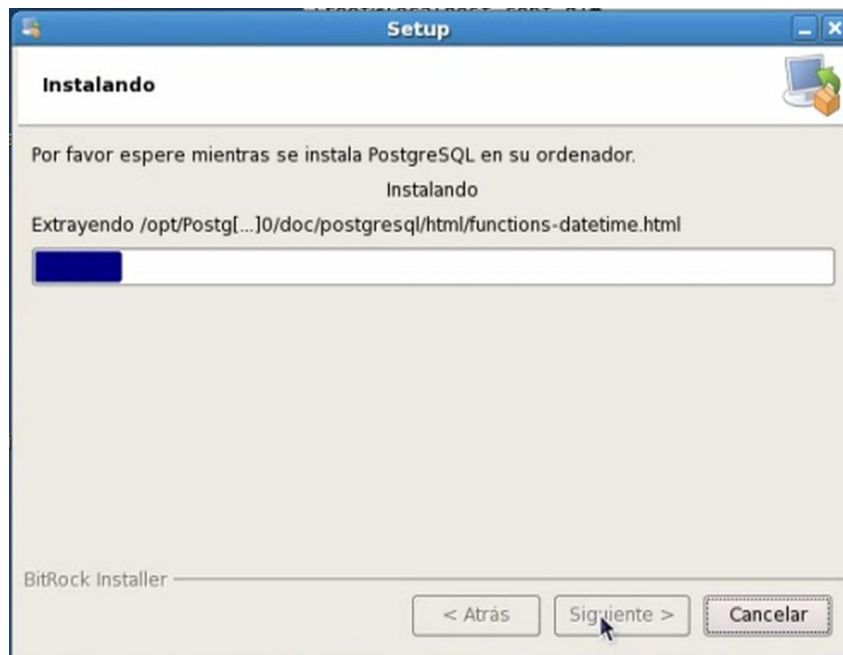


Figura 6.59. *Instalando*

6.9.5.2.3 Instalación del sistema de gestión de proyectos

Primer paso: Base de datos

Se procedió a crear la base de datos que utilizará el Sistema en PostgreSQL con el nombre “gestion_proyectos” junto con sus tablas, funciones y triggers.

La creación de las tablas, funciones y triggers se realizó mediante la ejecución de SQL en la base de datos.

Segundo paso: Subir el sistema de gestión de proyectos al servidor web

Copiamos el sistema en la dirección

```
/var/www/html
```

Asignamos los permisos de apache al sistema

```
chown apache.apache ceni
```

Agregamos la carpeta con el nombre “ArchivosServer” dentro del sistema y le asignamos los permisos de apache. En esta carpeta se almacenará los archivos de los usuarios.

```
mkdir /var/www/html/ceni/ArchivosServer
```

```
chown apache.apache /var/www/html/ceni/ArchivosServer
```

Tercer paso: Configuración de apache

Como el departamento UOCENI - ISEI cuenta con un dominio donde esta publicado el servicio E-Learning de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial se resolvió crear un subdominio para el Sistema de Gestión de proyectos.

La creación del subdominio se realizó de la siguiente manera:

- Creación el archivo aplica.conf en la dirección /etc/httpd/conf

```
nano /etc/httpd/conf/aplica.conf
```

Contenido del archivo aplica.conf

```
Alias /ceni /var/www/html/ceni
MonoApplication default "*/var/www/html/ceni"
<Directory /var/www/html/ceni>
    SetHandler mono
    DirectoryIndex Default.aspx
</Directory>
```

- Reinicio del servicio apache para almacenar la configuración en el mismo

```
service httpd restart
```

Con los pasos realizados ya se puede utilizar el sistema de gestión de proyectos que ha sido desarrollado e implantado ingresando a <http://e-fisei.uta.edu.ec/ceni>

6.9.5.3 Aplicación de pruebas en el nuevo sistema

Luego de que el sistema fue instalado y configurado en el servidor del departamento se procedió a verificar que su funcionamiento sea el mismo al ejecutado y aprobado por el Director en el ambiente de desarrollo, razón por la cual se aplicó pruebas de caja negra en el mismo y se obtuvo los siguientes resultados:

- El diseño de la interfaz del sistema no tuvo ningún inconveniente, las imágenes, colores y estilos funcionaron correctamente.
- La gestión de información de la base de datos a través del sistema no generó ningún error.
- Se verificó el funcionamiento correcto de cada uno de los formularios web pertenecientes al sistema.
- La seguridad de acceso al sistema se cumple a cabalidad.

Posteriormente el sistema fue sometido a un período de prueba en el ambiente de producción de la UOCENI – ISEI (prueba beta), con la finalidad de constatar y evaluar su correcto funcionamiento en escenarios reales.

6.9.5.4 Evaluaciones

- En cuanto a la parte técnica, el sistema se acoplo adecuadamente al servidor del departamento de la UOCENI – ISEI (Base de datos y sitio web).
- La capacitación al personal del departamento sobre el uso del sistema facilitó la conversión del mismo (sistema antiguo al sistema nuevo).
- En el lapso que el sistema estuvo a prueba en el ambiente de producción del departamento se pudo encontrar y corregir errores que pasaron desapercibidos en fases anteriores, factor que permitió cumplir con el objetivo de implantar el sistema en el departamento.
- El software implantando es de alta calidad, porque a través de él se logro satisfacer las expectativas y necesidades del departamento con respecto a la gestión de proyectos.

6.10 Conclusiones y Recomendaciones

6.10.1 Conclusiones

- Se desarrolló e implantó un Sistema web de Gestión de Proyectos que permitió dar solución al manejo inadecuado de la información en la UOCENI – ISEI.
- En cuanto al desarrollo web con ASP.NET de Mono fue muy interesante porque es una tecnología de código abierto de gran alcance comparable a la tecnología ASP.NET de Microsoft. Se pudo aplicar los conocimientos adquiridos sobre esta tecnología e investigar soluciones en la web para problemas que se presentaban a lo largo del desarrollo del sistema.
- Con la aplicación de pruebas en el sistema se pudo encontrar y corregir errores que permitieron asegurar el correcto funcionamiento en su implantación en el departamento UOCENI – ISEI.

6.10.2 Recomendaciones

- Se recomienda designar a una persona capacitada para el mantenimiento y control del sistema informático, para asegurar un mejor desempeño y el correcto funcionamiento del mismo.
- Efectuar respaldos de la Base de Datos consecutivamente y almacenarlos en dispositivos externos seguros, con la fecha y descripción que los identifique para evitar inconvenientes futuros.
- Los usuarios del sistema deben tener cuidado con el manejo de su contraseña pues el acceso de personas no autorizadas podría provocar daños en la correcta administración de los proyectos.

Bibliografía

Información bibliográfica de libros

PAZMAY, Galo (2004) Guía práctica para la elaboración de tesis y trabajos de investigación. Editorial Freire.

CAZAR, Héctor (2002) Compendio de Computación Siglo XXI. Editorial Española

Información bibliográfica de páginas web

Sistema Operativo. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Sistema_operativo

Clasificación Sistemas Operativos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.mitecnologico.com/Main/ClasificacionSistemasOperativos>

Características GNU/Linux. (s.f). Extraído el 10 de Octubre del 2010 desde http://gbtcr.chileforge.cl/info_web/node42.html

Microsoft Windows. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Microsoft_Windows

Microsoft Windows. (s.f). Extraído el 10 de Octubre del 2010 desde http://html.rincondelvago.com/microsoft-windows_2.html

Sistemas de Gestión de Base de Datos. (2005). Extraído el 10 de Octubre del 2010 desde http://machtiani.icyt.df.gob.mx/file.php/1/moddata/forum/1/9240/SGBD_2.pdf

Sistemas gestores de Base de datos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://admin-info.galeon.com/gestor.pdf>

Comparativa entre PostgreSQL y MySQL. (2008). Extraído el 10 de Octubre del 2010 desde <http://www.taringa.net/posts/info/1073108/Comparativa-entre-Postgres-y-MySQL.html>

Servidores Web. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>

Servidor http Apache. (s.f). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

World Wide Web. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/World_Wide_Web

Gestión de proyectos. (2010). Extraído el 10 de Octubre del 2010 desde http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos

Gestión de proyectos. (s.f). Extraído el 10 de Octubre del 2010 desde <http://www.deltaasesores.com/articulos/gestion-de-proyectos>

"Automatización." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008.

JEFF. (2008). Introducción a la gestión de proyectos. Extraído el 10 de Octubre del 2010 desde <http://es.kioskea.net/contents/projet/projetintro.php3>

CABALLERO, Omar. (2006). Tecnologías de información y herramientas para la administración de proyectos de software. Extraído el 10 de Octubre del 2010 desde <http://www.revista.unam.mx/vol.7/num6/int47/art47.htm>

Framework. (2011). Extraído el 1 de Junio del 2011 desde <http://es.wikipedia.org/wiki/Framework>

ASP.NET. (2011). Extraído el 1 de Junio del 2011 desde
<http://es.wikipedia.org/wiki/ASP.NET>

MonoDevelop. (s.f). Extraído el 1 de Junio del 2011 desde
<http://monodevelop.com/>

XSP. (s.f). Extraído el 2 de Junio del 2011 desde
<http://www.mono-project.com/ASP.NET>

JavaScript. (s.f). Extraído el 2 de Junio del 2011 desde
<http://www.ulpgc.es/otros/tutoriales/JavaScript/cap1.htm>

OSMOSIS LATINA. (2000-2011). Diseñado bajo estándares. Extraído el 5 de Junio del 2011 desde
<http://www.osmosislatina.com/lenguajes/uml/basico.htm>

CÁCERES, Jesús. (s.f). Diagramas de casos de uso. Extraído el 5 de Junio del 2011 desde
<http://www2.uah.es/jcaceres/uploaded/capsulas/DiagramaCasosDeUso.pdf>

Diagrama de secuencia. (2011). Extraído el 12 de Junio del 2011 desde
http://es.wikipedia.org/wiki/Diagrama_de_secuencia

Técnicas de prueba. (s.f). Extraído el 18 de junio del 2011 desde
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>

My Adventures Installing mono 2.0 on CentOS 4 to work with apache via mod_mono. (s.f). Extraído el 19 de junio del 2011 desde
http://blog.palehorse.net/2008/11/06/my-adventures-installing-mono-20-on-centos-4-to-work-with-apache-via-mod_mono/

Glosario de términos

Framework: Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de *software* concretos, con base en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

S.O: (Sistema Operativo), es el programa o conjunto de programas que efectúan la gestión de los procesos básicos de un sistema informático, y permite la normal ejecución del resto de las operaciones.

SQL: (Structured Query Language), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

PL / PGSQL: (Procedural Language/PostgreSQL Structured Query Language), es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas.

PHP: (Hypertext Pre-Processor), es un lenguaje de programación para desarrollo de aplicaciones Web.

API: (application programming interface), es un conjunto de funciones residentes en bibliotecas que permiten que una aplicación corra bajo un determinado sistema operativo.

UML: (Unified Modeling Language), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

BUCLE: es un tipo de estructura de control que permite repetir una o más sentencias múltiples veces.

Open Source: Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

Usuario, Un usuario generalmente se identifica frente al sistema utilizando un nombre de usuario (nick) y una contraseña. Un usuario registrado accede a un servicio a través de un login luego de su autenticación.

Interfaz Amigable, Cualidad de una interfaz de programa que por su forma de interactuar con el usuario es considerada de fácil uso.

ANEXOS

ANEXO 1: Decreto presidencial para el uso de software libre en la república del Ecuador

Artículo 1.- Establecer como política pública para las Entidades de la Administración Pública Central la utilización de Software Libre en sus sistemas y equipamientos informáticos.

Artículo 2.- Se entiende por Software Libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan su acceso a los códigos fuentes y que sus aplicaciones. puedan ser mejoradas. Estos programas de computación tienen las siguientes libertades:

- a) Utilización del programa con cualquier propósito de uso común.
- b) Distribución de copias sin restricción alguna.
- c) Estudio y modificación del programa (Requisito: código fuente disponible).
- d) Publicación del programa mejorado (Requisito: código fuente disponible).

Artículo 3.- Las entidades de la Administración Pública Central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de éste tipo de software.

Artículo 4.- Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de Software Libre que supla las necesidades requeridas, o cuando esté en riesgo la seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

En este caso, se concibe como seguridad nacional, las garantías para la supervivencia de la colectividad y la defensa del patrimonio nacional.

Para efectos de este decreto se entiende por un punto de no retorno, cuando el sistema o proyecto informático se encuentre en cualquiera des estas condiciones:

- a) Sistema en producción funcionando satisfactoriamente y que un análisis de costo – beneficio muestre que no es razonable ni conveniente una migración a Software Libre.
- b) Proyecto en estado de desarrollo y que un análisis de costo – beneficio muestre que no es conveniente modificar el proyecto y utilizar Software Libre.

Periódicamente se evaluarán los sistemas informáticos que utilizan software propietario con la finalidad de migrarlos a Software Libre.

Artículo 5.- Tanto para software libre como software propietario, siempre y cuando se satisfagan los requerimientos, se debe preferir las soluciones en este orden:

- a) Nacionales que permitan autonomía y soberanía tecnológica.
- b) Regionales con componente nacional.
- c) Regionales con proveedores nacionales.
- d) Internacionales con componente nacional.
- e) Internacionales con proveedores nacionales.
- f) Internacionales.

Artículo 6.- La Subsecretaría de Informática como órgano regulador y ejecutor de las políticas y proyectos informáticos en las entidades del Gobierno Central deberá realizar el control y seguimiento de éste Decreto.

Para todas las evaluaciones constantes en este decreto la Subsecretaría de Informática establecerá los parámetros y metodologías obligatorias.

Artículo 7.- Encárguese de la ejecución de este decreto a los señores Ministros Coordinadores y el señor Secretario General de la Administración Pública y Comunicación.

Dado en el Palacio Nacional en la ciudad de San Francisco de Quito, Distrito Metropolitano, el día de hoy 10 de abril del 2008.

RAFAEL CORREA DELGADO
PRESIDENTE CONSTITUCIONAL DE LA REPÚBLICA

ANEXO 2: Estructura del cuestionario

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS

Cuestionario dirigido al personal que labora en la UOCENI – ISEI de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Sírvase contestar el cuestionario de la manera más honesta posible, puesto que la información servirá exclusivamente para mejorar la productividad del departamento.

1. ¿De qué manera se maneja la gestión de proyectos en la UOCENI -ISEI?
 - Manualmente
 - Sistema Informático
 - Excel
 - Otro programa

2. ¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite un control adecuado del proceso de desarrollo de los proyectos?
 - Si
 - No

3. ¿El sistema o mecanismo utilizado actualmente por la UOCENI – ISEI permite la generación de reportes oportunos?
 - Si
 - No

4. ¿Cree Ud. que sería conveniente poner en marcha un plan de renovación tecnológica para mejorar y facilitar la administración de los proyectos?
- Si
 - No
5. ¿Cómo está almacenada la información de los proyectos actualmente?
- Base de Datos Manual
 - Base de Datos Automatizada
6. ¿La información almacenada de los proyectos es confiable?
- Si
 - No
7. ¿Cree usted que toda la información almacenada de los proyectos es necesaria?
- Si
 - No
8. ¿La generación y obtención de información acerca de los avances de cada proyecto es?
- Buena
 - Aceptable
 - Regular
 - Pésima

9. ¿El seguimiento y control del personal que trabaja en los proyectos es?

Muy bueno

Aceptable

Regular

Pésimo

ANEXO 3: Respaldo y recuperación de una Base de Datos

Debido a la gran importancia de la información que maneja el Sistema de Gestión de Proyectos y a los problemas que originaría su pérdida, es recomendable realizar periódicamente respaldos de la Base de Datos y almacenarlos en dispositivos extraíbles.

Pasos para realizar un respaldo una Base de Datos en PostgreSQL:

1. Clic derecho en la base de datos que deseamos respaldar y escogemos la opción “Backup” (Ver figura 1)

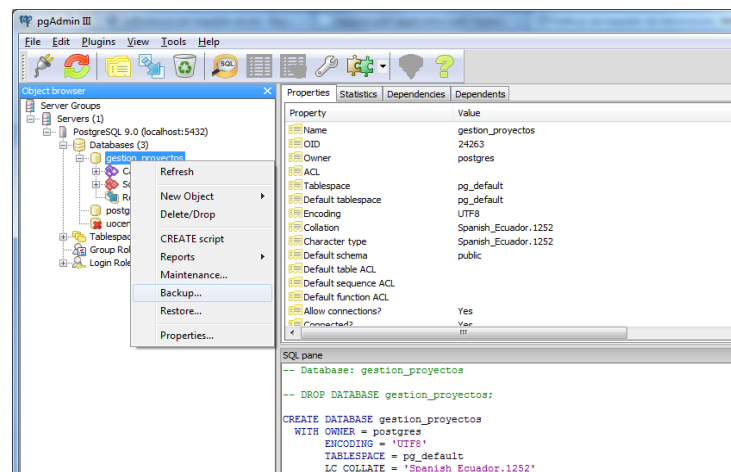


Figura 1

2. Como consecuencia del paso anterior aparecerá una ventana donde deberemos llenar los campos que PostgreSQL requiere para la realización del respaldo. Información como: nombre, ubicación, tipo, etc. (Ver figura 2)

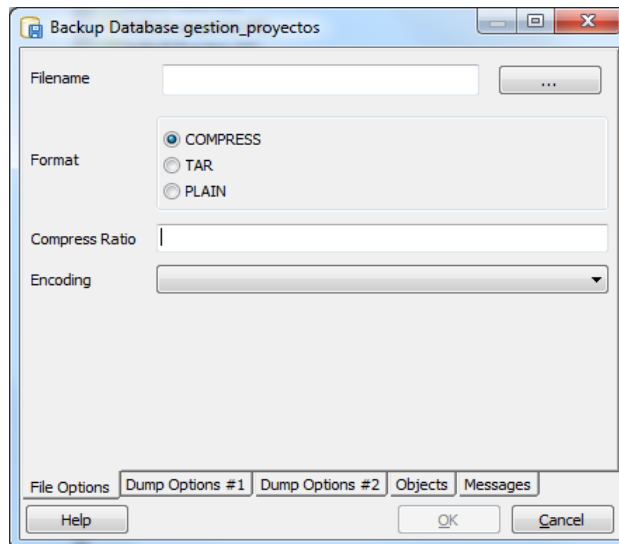


Figura 2

3. Como último paso seleccionamos “OK” y ya tenemos un respaldo de nuestra Base de Datos

Pasos para recuperar un respaldo de una Base de Datos en PostgreSQL:

1. Creamos una base de datos con el mismo nombre de la que fue respaldada anteriormente. (Ver figura 3)
 - 1.1 Clic derecho en la opción “DataBase”
 - 2.1 Seleccionamos la opción “New Database...”

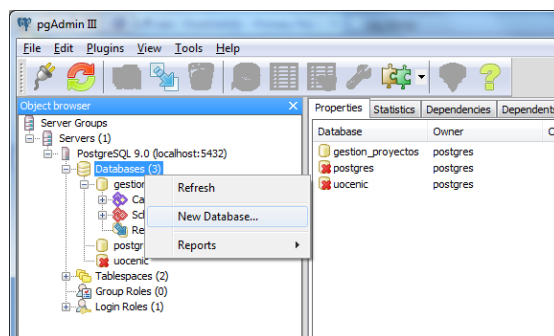


Figura 3

- 2.1 Como consecuencia del paso anterior aparecerá una ventana donde deberemos llenar los campos necesarios para la creación de la nueva base de datos. (Ver figura 4)

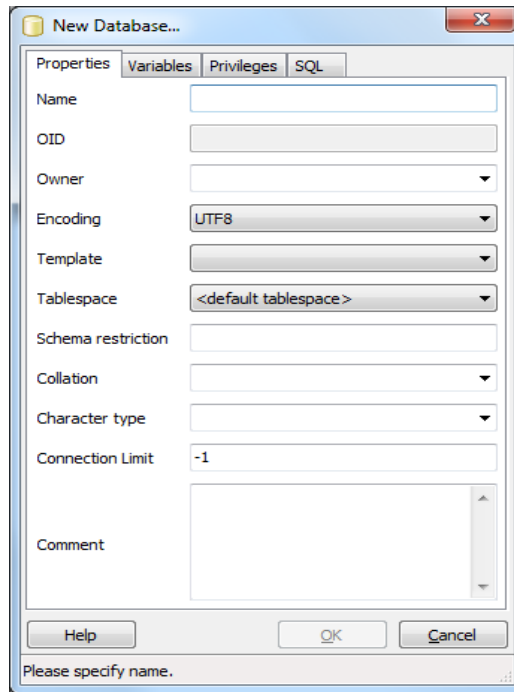


Figura 4

4. Damos clic derecho en la base de datos creada en el paso anterior y escogemos la opción “Restore”. (Ver figura 5)

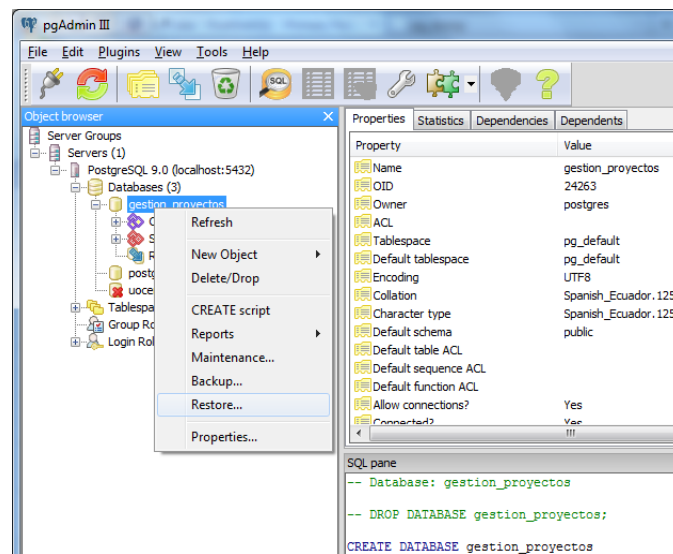


Figura 5

5. Como consecuencia del paso anterior aparecerá una ventana donde deberemos llenar los campos que PostgreSQL requiere para restaurar la Base de Datos. Información como: nombre y ubicación del respaldo. (Ver figura 6)

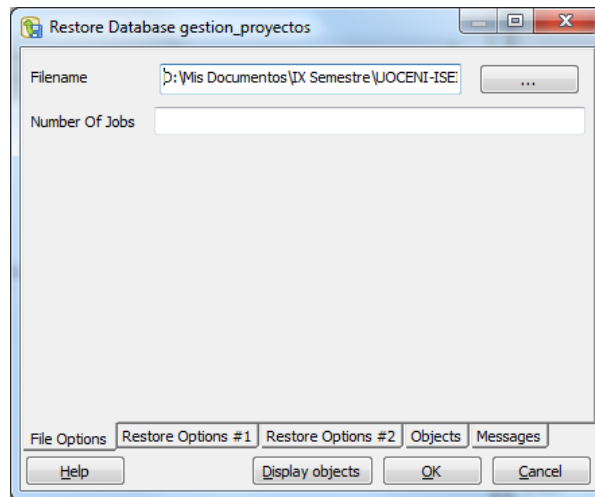


Figura 6

6. Como último paso seleccionamos "OK" y ya tenemos restaurada nuestra Base de Datos