



**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB
DE GESTIÓN DE SOLICITUDES ONLINE DE HELPDESK
EMPRESARIAL, ORIENTADO A LA ATENCIÓN AL CLIENTE
EN TODOS LOS SERVICIOS BRINDADOS POR LA EMPRESA
WICOM DEL ECUADOR**

AUTOR: ROSERO VEGA, DANNY ARCELIANO

DIRECTOR: ING. DÍAZ, OSWALDO

CODIRECTOR: ING. CAMPAÑA, MAURICIO

SANGOLQUÍ

MAYO - 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
CARRERA INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

Ing. Oswaldo Díaz (Director)

Ing. Mauricio Campaña (Codirector)

CERTIFICAN

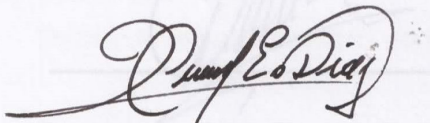
Que el trabajo titulado **“Diseño e implementación de un sistema web de gestión de solicitudes online de helpdesk empresarial, orientado al cliente en todos los servicios brindados por la empresa Wicom del Ecuador”**, realizado por el Sr. Danny Arceliano Rosero Vega, ha sido guiado y revisado periódicamente y cumple con las normas estatutarias establecidas por la ESPE, el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas.

Debido a que la tesis concluye satisfactoriamente y cumple con las disposiciones institucionales, metodología y técnicas para su ejecución recomiendan su publicación.

El mencionado trabajo consta de dos documentos empastados y dos discos compactos, el cual contiene los archivos en formato portátil de Acrobat (pdf).

Autorizan al Sr. Danny Arceliano Rosero Vega que los entregue al Ing. Mauricio Campaña en calidad de Director de Carrera de Ingeniería en Sistemas e Informática.

Salgolquí, 14 de Mayo 2015.



Ing. Oswaldo Díaz
DIRECTOR TESIS



Ing. Mauricio Campaña
CODIRECTOR TESIS

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
CARRERA INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Danny Arceliano Rosero Vega **DECLARO QUE:**

El proyecto de grado denominado titulado "**Diseño e implementación de un sistema web de gestión de solicitudes online de helpdesk empresarial, orientado al cliente en todos los servicios brindados por la empresa Wicom del Ecuador**", ha sido desarrollado en base a una investigación exhaustiva respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las paginas correspondientes, cuyas fuentes se incorporen en las bibliografías.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo de este contenido, veracidad y alcance científico del proyecto de grado mencionado.

Salgolquí, 14 de Mayo 2015.



Danny Arceliano Rosero Vega

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
CARRERA INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Danny Arceliano Rosero Vega

Autorizo a la Universidad de las Fuerzas Armadas la publicación en la biblioteca virtual de la institución del trabajo **"Diseño e implementación de un sistema web de gestión de solicitudes online de helpdesk empresarial, orientado al cliente en todos los servicios brindados por la empresa Wicom del Ecuador"**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Salgolquí, 14 de Mayo 2015.

A handwritten signature in blue ink, consisting of a large 'D' followed by several loops and a final flourish, positioned above a horizontal line.

Danny Arceliano Rosero Vega

DEDICATORIA

A mi Madre, mujer símbolo de lucha y entrega por sus hijos, por sus caricias, por sus cuidados, por enseñarme que en la vida primero siempre estará Dios y la Familia, por inculcarme buenos valores, honestidad y responsabilidad, siempre a mi lado y por ese motivo yo siempre estaré al tuyo, por eso y mucho más esta tesis es tuya mama.

A mi Padre, no sé cómo expresar la gratitud que te tengo, mis ojos se llenan de lágrimas al escribir esto, el simple motivo de que ya no estas a mi lado, que no pude darte la satisfacción de verme recibir mi título, pero sé que desde el cielo lo estás haciendo y estas orgulloso de mi. Tu ejemplo de trabajo y dedicación han logrado que llegue a ser el profesional que hoy en día lo soy, este título complementa la carrera profesional que tiempo atrás al ver tu ejemplo me propuse llegar a tener.

A mi Esposa, quiero agradecerte por el apoyo que me das día a día, por el amor y entrega que recibo de tu parte, por la satisfacción de saber que estás conmigo en los altibajos de la vida, por ser mi apoyo para lograr todos los objetivos de vida y familia que nos hemos planteado, por esto y mucho más te digo que te Amo y quiero envejecer a tu lado.

A mi hija, mi princesa, Christell Alizée, ahora a tus 5 años sé que no me entenderás si te lo digo, pero quiero que lo leas cuando hayas aprendido a leer, toda mi vida te la entrego y está dedicada a ti, tu eres el combustible que mueve mi vida, espero ser tu ejemplo y que en mi puedas confiar siempre, quiero que sepas que siempre estaré para ti y si alguna vez me pongo exigente, simplemente es porque quiero lo mejor para que llegues a ser una gran persona y una gran profesional, nunca dejes de soñar, nunca dejes de luchar, nunca dejes que te digan que no puedes lograrlo, nadie puede quitarte las ganas de llegar a ser lo que tú quieras ser, no temas equivocarte de esos tropezones aprenderás a superar las barreras que la vida te ponga solo quiero dejarte dos frases de Walt Disney, creador de todas las caricaturas y princesas que tanto te gustan, él dijo: “Si puedes soñarlo, puedes hacerlo” y “Camina hacia el futuro”. Te Amo.

Danny Arceliano Rosero Vega

AGRADECIMIENTO

En primer lugar a Dios por llenarme de bendiciones cada día de mi vida y saber que siempre estará ahí en los malos y buenos momentos, a él es a quien he considerado mi verdadero amigo.

A la Universidad de las Fuerzas Armadas por ser parte integral de mi formación académica y haberme permitido estudiar y obtener mi título en tan prestigioso establecimiento educativo.

A cada uno de los profesores que impartieron sus conocimientos durante mi paso por las aulas de tan noble institución, en especial al Ing. Oswaldo Díaz e Ing. Mauricio Campaña por su compromiso, dedicación y consejos en el desarrollo de este proyecto.

A toda mi Familia en general, por demostrar unidad y fuerza ante las adversidades, por mantenernos siempre unidos y ser un apoyo constante.

Danny Arceliano Rosero Vega

ÍNDICE DE CONTENIDO

CERTIFICADO	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
RESUMEN	xxi
ABSTRACT	xxii
CAPÍTULO 1.....	1
INTRODUCCIÓN	1
1.1 Planteamiento del Problema	2
1.2 Justificación.....	2
1.3 Objetivos	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos Específicos	3
1.4 Alcance	4
1.5 Factibilidad	5
1.5.1 Factibilidad Técnica	5
1.5.2 Factibilidad Tecnológica	5
1.5.3 Factibilidad Operativa	6
1.5.4 Factibilidad Operacional	6
1.5.5 Factibilidad Económica.....	7
1.5.6 Factibilidad Legal.....	7
1.6 Metodología	9
1.7 Herramientas.....	9

CAPÍTULO 2.....	11
MARCO TEÓRICO	11
2.1 Metodología del Desarrollo de Sistemas.....	11
2.1.1 Lenguaje Unificado de Modelado “UML”	12
2.1.1.1 Modelos	13
2.1.1.2 Elementos comunes en los Diagramas.....	14
2.1.1.3 Diagramas Estáticos	19
2.1.1.4 Diagramas Dinámicos	28
2.1.2 Proceso unificado de software “RUP”	32
2.1.2.1 Características	33
2.1.2.2 Ciclo de vida del proceso unificado.....	33
2.1.2.3 Fases del ciclo de vida del proceso unificado	35
2.1.3 Metodología de Diseño Hipermedia Orientado a Objetos..	39
2.1.4 Herramientas para análisis y diseño orientado a objetos ..	41
2.1.4.1 StarUML.....	41
2.1.4.2 Rational Rose.....	42
2.1.4.3 BoUML	43
2.1.4.4 Análisis de las herramientas case.....	44
2.2 Aplicaciones web.....	44
2.3 Arquitectura de software de las aplicaciones web.....	47
2.3.1 Arquitectura tres capas	47
2.4 Tecnologías Web	49
2.4.1 Java Server Page “JSP”	49
2.4.2 Active Server Pages “ASP”	51
2.4.3 PHP: Hypertext Preprocessor “PHP”	52
2.4.4 Análisis de Tecnología Web	52

2.4.5	JavaScript	53
2.4.6	Ajax.....	54
2.4.7	Entorno Integrado de desarrollo	55
2.4.7.1	Netbeans.....	55
2.4.7.2	PHP Edit	57
2.4.7.3	DreamWeaver.....	58
2.4.7.4	Análisis de herramientas IDE para PHP	59
2.5	Sistemas de administración de base de datos	59
2.5.1	MySql.....	59
2.5.2	Postgre SQL	60
2.5.3	Oracle	61
2.5.4	Análisis Motor Base de Datos.....	62
2.5.5	Gestión Base de Datos	63
2.5.5.1	PHPMyAdmin.....	63
2.5.5.2	Navicat	64
2.5.5.3	Mysql Workbench.....	64
2.5.5.4	Análisis Herramienta de gestión de Base de Datos.	64
2.6	Servidor Web	65
2.6.1	WAMPServer	66
2.6.2	Portable WebServer	67
2.6.3	XAMPP	67
2.6.4	Análisis del Servidor Web.....	68
2.7	Protocolos de comunicaciones.....	69
2.7.1	Transfer Control Protocol / Internet Protocol (TCP/IP).....	69
2.7.2	Hypertext Transfer Protocol.....	70
2.8	Plataforma / Sistema Operativo	71

2.8.1	Windows	71
2.8.2	Linux	72
2.8.3	MacOS.....	72
2.8.4	Análisis de Plataforma / Sistema Operativo.....	73
CAPÍTULO 3.....		74
ANÁLISIS Y DISEÑO		74
3.1	Desarrollo de la Aplicación	74
3.1.1	Modelado del Negocio	74
3.2	Obtención de Requisitos	79
3.2.1	Entrevista a Cliente.....	79
3.2.2	Resultado Entrevistas	80
3.2.2.1	Recepción / Ingreso de solicitudes.....	81
3.2.2.2	Análisis y Gestión de solicitudes	82
3.2.2.3	Finalización de solicitudes	82
3.2.3	Especificación de Requisitos de Software G.S.O.	83
3.2.3.1	Introducción	83
3.2.3.2	Descripción General.....	84
3.2.3.3	Requerimientos funcionales	92
3.2.3.4	Requerimientos No Funcionales	93
3.3	Análisis y Diseño	94
3.3.1	Diagramas casos de uso	95
3.3.1.1	Caso de uso Login	95
3.3.1.2	Caso de uso Cambiar Clave.	97
3.3.1.3	Caso de uso Gestionar Tablas.....	99
3.3.1.4	Caso de Uso Gestionar Clientes	104
3.3.1.5	Caso de Uso Gestionar Usuarios.....	110

3.3.1.6	Caso de Uso Gestionar Empresas.....	116
3.3.1.7	Caso de Uso Gestionar Solicitudes.....	121
3.3.1.8	Caso de Uso Gestionar Técnicos.....	135
3.3.1.9	Caso de Uso Calificar Atención.....	141
3.3.1.10	Caso de Uso Generar Reportes.....	142
3.3.2	Diagramas de Secuencia.....	144
3.3.2.1	Diagramas de Secuencia Login	144
3.3.2.2	Diagramas de Secuencia Cambiar Clave	145
3.3.2.3	Diagramas de Secuencia Gestionar Tablas	145
3.3.2.4	Diagramas de Secuencia Gestionar Clientes.....	147
3.3.2.5	Diagramas de Secuencia Gestionar Usuario	148
3.3.2.6	Diagramas de Secuencia Gestionar Empresas	150
3.3.2.7	Diagramas de Secuencia Gestionar Solicitudes	151
3.3.2.8	Diagramas de Secuencia Gestionar Técnicos	156
3.3.2.9	Diagramas de Secuencia Calificar Atención	157
3.3.2.10	Diagramas de Secuencia Generar Reportes	158
3.3.3	Diagramas de Colaboración	158
3.3.3.1	Diagrama de Colaboración Login.....	158
3.3.3.2	Diagrama de Colaboración Cambiar Clave	159
3.3.3.3	Diagrama de Colaboración Gestionar Tablas	159
3.3.3.4	Diagrama de Colaboración Gestionar Clientes	160
3.3.3.5	Diagrama de Colaboración Gestionar Usuario.....	161
3.3.3.6	Diagrama de Colaboración Gestionar Empresas	162
3.3.3.7	Diagrama de Colaboración Gestionar Solicitudes.....	163
3.3.3.8	Diagrama de Colaboración Gestionar Técnicos.....	166
3.3.3.9	Diagrama de Colaboración Calificar Atención.....	167

3.3.3.10 Diagrama de Colaboración Generar Reportes	167
3.3.4 Diagramas de Estado	168
3.3.4.1 Diagrama de Estado Clientes	168
3.3.4.2 Diagrama de Estado Usuarios	168
3.3.4.3 Diagrama de Estado Empresas	169
3.3.4.4 Diagrama de Estado Solicitudes	170
3.3.4.5 Diagrama de Estado Técnicos	171
3.3.5 Diagrama de Despliegue	171
3.3.6 Diagrama de Clases	172
3.3.7 Diagrama Lógico Base de Datos	173
3.3.8 Diagrama Físico Base de Datos	174
3.3.9 Diagrama Navegacional G.S.O.....	175
CAPÍTULO 4.....	176
IMPLEMENTACIÓN	176
4.1 Arquitectura del sistema.....	176
4.2 Codificación.....	177
4.3 Pruebas.....	179
4.3.1 Pruebas de Caja Blanca “Secuencia”	179
4.3.2 Prueba de Caja Negra “Proceso”	183
4.4 Seguridades	185
4.5 Manuales G.S.O.....	188
4.5.1 Manual de Referencia Técnica	188
4.5.2 Manual de Usuario Final.....	188
CAPÍTULO 5.....	189
CONCLUSIONES Y RECOMENDACIONES.....	189
5.1 Conclusiones.....	189

5.2	Recomendaciones.....	190
BIBLIOGRAFÍA.....		191

ÍNDICE TABLAS

Tabla 1	Factibilidad Económica	7
Tabla 2	Multiplicidad de asociaciones.....	17
Tabla 3	Cuadro Comparativo Herramientas Case	44
Tabla 4	Cuadro Comparativo Tecnologías.....	52
Tabla 5	Cuadro comparativo de motores de Bases de Datos	62
Tabla 6	Cuadro comparativo herramientas de gestión de BD.....	64
Tabla 7	Cuadro Servidor Web.....	68
Tabla 8	Cuadro Sistemas Operativos	73
Tabla 9	Subproceso 1	76
Tabla 10	Subproceso 2	77
Tabla 11	Subproceso 3	77
Tabla 12	Proceso recepción / Ingreso Solicitudes	81
Tabla 13	Proceso Análisis y Gestión de Solicitudes	82
Tabla 14	Finalización de Solicitudes	82
Tabla 15	Requerimientos Funcionales GSO.....	92
Tabla 16	Requerimientos No Funcionales	93

ÍNDICE DE CUADROS

Cuadro 1 Detalle caso de uso Login	96
Cuadro 2 Detalle caso de uso Cambiar Clave	98
Cuadro 3 Detalle Caso de Uso Buscar Tablas.....	99
Cuadro 4 Detalle Caso de Uso Crear Registros	100
Cuadro 5 Detalle Caso de Uso Editar Registros	102
Cuadro 6 Detalle Caso de Uso Valida Campos Registro	103
Cuadro 7 Detalle Caso de Uso Crear Clientes.....	105
Cuadro 8 Detalle Caso de Uso Buscar Clientes	106
Cuadro 9 Detalle Caso de Uso Editar Clientes	107
Cuadro 10 Detalle Caso de Uso Valida Campos Cliente	109
Cuadro 11 Detalle Caso de Uso Crear Usuarios	110
Cuadro 12 Detalle Caso de Uso Buscar Usuarios	112
Cuadro 13 Detalle Caso de Uso Editar Usuarios	113
Cuadro 14 Detalle Caso de Uso Valida Campos Usuario	115
Cuadro 15 Detalle Caso de Uso Crear Empresas	116
Cuadro 16 Detalle Caso de Uso Buscar Empresas	118
Cuadro 17 Detalle Caso de Uso Editar Empresas	119
Cuadro 18 Detalle Caso de Uso Valida Campos Empresa	120
Cuadro 19 Detalle Caso de Uso Crear Solicitud	122
Cuadro 20 Detalle Caso de Uso Validar Campos Solicitud.....	123
Cuadro 21 Detalle Caso de Uso Envío Email	124
Cuadro 22 Detalle Caso de Uso Buscar Solicitudes	125
Cuadro 23 Detalle Caso de Uso Agregar Observaciones	127
Cuadro 24 Detalle Caso de Uso reasignar Cliente	128
Cuadro 25 Detalle Caso de Uso	129
Cuadro 26 Detalle Caso de Uso Finalizar Ticket	130
Cuadro 27 Detalle de Caso de Uso Finalizar Solicitud	131
Cuadro 28 Detalle Caso de Uso Anular Solicitud.....	133
Cuadro 29 Detalle de Caso de Uso Imprimir Solicitud	134
Cuadro 30 Detalle Caso de Uso Crear Técnicos	135
Cuadro 31 Detalle Caso de Uso Buscar Técnicos	137

Cuadro 32 Detalle Caso de Uso Editar Técnicos	138
Cuadro 33 Detalle Caso de Uso Valida Campos Técnico	139
Cuadro 34 Detalle Caso de Uso Calificar Solicitud	141
Cuadro 35 Detalle Caso de Uso Generar Reportes	143
Cuadro 36 Proceso de interacción creación de técnico	179
Cuadro 37 Prueba Webmaster creación de técnico	180
Cuadro 38 Proceso de interacción creación de Solicitud	181
Cuadro 39 Prueba Drosero creación de Solicitud	182
Cuadro 40 Proceso interacción finalizar solicitud	183
Cuadro 41 Prueba Drosero finalizar solicitud	183
Cuadro 42 Proceso interacción Calificar Solicitud	184
Cuadro 43 Prueba Webmaster calificación de solicitud	184

ÍNDICE DE FIGURAS

Figura 1. Representación de Notas en UML	14
Figura 2. Relaciones de Dependencia	15
Figura 3. Modelo de una clase en UML	16
Figura 4. Representación de asociaciones	17
Figura 5. Representación de multiplicidad	17
Figura 6. Representación de roles en una asociación	18
Figura 7. Representación de agregación	18
Figura 8. Modelo de herencia	19
Figura 9. Caso de Uso	20
Figura 10. Representación de relación “Uses”	21
Figura 11. Representación con relación “Extends”	22
Figura 12. Diagrama de clases	23
Figura 13. Atributos.....	24
Figura 14. Operación	25
Figura 15. Generalización	25
Figura 16. Dependencia.....	26
Figura 17. Diagrama de Objetos	26
Figura 18. Diagrama de Componentes	27
Figura 19. Diagrama de Despliegue.	28
Figura 20. Diagrama de Secuencia.....	29
Figura 21. Diagrama de Colaboración.	30
Figura 22. Diagrama de Estado.	31
Figura 23. Diagrama de Actividad	31
Figura 24. Ciclo de vida de RUP.....	33
Figura 25. Fases del Ciclo de vida de RUP.	34
Figura 26. Gráfico Arquitectura Tres Capas.....	47
Figura 27. Comunicación en Modelo 3 Capas	48
Figura 28. Modelado Contextual del Negocio	75
Figura 29. Flujograma de Gestión de Solicitudes Online Helpdesk	78
Figura 30. Plantilla Interface Gráfica GSO	87
Figura 31. Infraestructura de comunicación	89

Figura 32. Diagrama caso de uso requisitos Funcionales GSO.....	95
Figura 33. Caso de uso Login	96
Figura 34. Caso de uso cambiar Clave	97
Figura 35. Caso de uso Gestionar Tablas.....	99
Figura 36. Caso de Uso Gestionar Clientes.....	104
Figura 37. Caso de Uso Gestionar Usuarios.....	110
Figura 38. Caso de Uso Gestionar Empresas.....	116
Figura 39. Caso de Uso Gestionar Solicitud	121
Figura 40. Caso de Uso Gestionar Técnicos	135
Figura 41. Caso de Uso Calificar Atención	141
Figura 42. Caso de Uso Generar Reportes.....	143
Figura 43. Diagrama Secuencia Login	144
Figura 44. Diagrama de Secuencia Cambio de Clave	145
Figura 45. Diagrama de Secuencia Buscar Tablas	145
Figura 46. Diagrama de Secuencia Crear Registros.....	146
Figura 47. Diagrama de Secuencia Editar Registros	146
Figura 48. Diagrama de Secuencia Crear Clientes.....	147
Figura 49. Diagrama de Secuencia Buscar Clientes.....	147
Figura 50. Diagrama de Secuencia Editar Clientes	148
Figura 51. Diagrama de Secuencia Crear Usuarios.....	148
Figura 52. Diagrama de Secuencia Buscar Usuario	149
Figura 53. Diagrama de Secuencia Editar Usuario	149
Figura 54. Diagrama de Secuencia Crear Empresa	150
Figura 55. Diagrama de Secuencia Buscar Empresa	150
Figura 56. Diagrama de Secuencia Editar Empresa	151
Figura 57. Diagrama de Secuencia Crear Solicitudes	151
Figura 58. Diagrama de Secuencia Buscar Solicitudes	152
Figura 59. Diagrama de Secuencia Agregar Observaciones	152
Figura 60. Diagrama de Secuencia Reasignar Cliente	153
Figura 61. Diagrama de Secuencia Reasignar Técnico.....	153
Figura 62. Diagrama de Secuencia Finalizar Ticket.....	154
Figura 63. Diagrama de Secuencia Finalizar Solicitud.....	154

Figura 64. Diagrama de Secuencia Anular Solicitud	155
Figura 65. Diagrama de Secuencia Imprimir Solicitud	155
Figura 66. Diagrama de Secuencia Crear Técnicos	156
Figura 67. Diagrama de Secuencia Buscar Técnicos	156
Figura 68. Diagrama de Secuencia Editar Técnicos	157
Figura 69. Diagrama de Secuencia Calificar Solicitud	157
Figura 70. Diagrama de Secuencia Generar Reportes	158
Figura 71. Diagrama de Colaboración Login.....	158
Figura 72. Diagrama de Colaboración Cambiar Clave.....	159
Figura 73. Diagrama de Colaboración Buscar Tablas	159
Figura 74. Diagrama de Colaboración Crear Registros	159
Figura 75. Diagrama de Colaboración Editar Registros.....	160
Figura 76. Diagrama de Colaboración Buscar Clientes	160
Figura 77. Diagrama de Colaboración Crear Clientes.....	160
Figura 78. Diagrama de Colaboración Editar Clientes	161
Figura 79. Diagrama de Colaboración Buscar Usuarios	161
Figura 80. Diagrama de Colaboración Crear Usuarios	161
Figura 81. Diagrama de Colaboración Editar Usuarios.....	162
Figura 82. Diagrama de Colaboración Buscar Empresas	162
Figura 83. Diagrama de Colaboración Crear Empresas	162
Figura 84. Diagrama de Colaboración Editar Empresas.....	163
Figura 85. Diagrama de Colaboración Buscar Solicitudes	163
Figura 86. Diagrama de Colaboración Crear Solicitudes	163
Figura 87. Diagrama de Colaboración Agregar Observaciones.....	164
Figura 88. Diagrama de Colaboración Reasignar Cliente.....	164
Figura 89. Diagrama de Colaboración Reasignar Técnico.....	164
Figura 90. Diagrama de Colaboración Finalizar Ticket	165
Figura 91. Diagrama de Colaboración Finalizar Solicitud	165
Figura 92. Diagrama de Colaboración Anular Solicitud.....	165
Figura 93. Diagrama de Colaboración Imprimir Solicitud.....	166
Figura 94. Diagrama de Colaboración Buscar Técnicos.....	166
Figura 95. Diagrama de Colaboración Crear Técnicos	166

Figura 96. Diagrama de Colaboración Editar Técnicos.....	167
Figura 97. Diagrama de Colaboración Calificar Solicitud.....	167
Figura 98. Diagrama de Colaboración Generar Reportes.....	167
Figura 99. Diagrama de Estado Clientes	168
Figura 100. Diagrama de Estado Usuarios	169
Figura 101. Diagrama de Estado Empresas	169
Figura 102. Diagrama de Estado Solicitudes	170
Figura 103. Diagrama de Estado Técnicos	171
Figura 104. Diagrama de Despliegue Sistema G.S.O.....	171
Figura 105. Diagrama de Clases G.S.O.....	172
Figura 106. Diagrama Lógico Base de Datos G.S.O.	173
Figura 107. Diagrama Físico Base de Datos G.S.O	174
Figura 108. Diagrama Navegacional G.S.O.....	175
Figura 109. Configuración Básica de la Clase de Base de Datos.....	177
Figura 110. Función de conexión privada a la Base de Datos	177
Figura 111. Función Cambio de conexión.....	178
Figura 112. Función conexión publica a la base de datos	178
Figura 113. Función que define el tipo de resultado	178
Figura 114. Función Pública Query.....	179
Figura 115. Creación de llave SSL	185
Figura 116. Creación de CSR.....	186
Figura 117. Creación de Certificado.....	186
Figura 118. Copia certificado.	186
Figura 119. Copia llave.	186
Figura 120. Inicio de servidor SSL.	187
Figura 121. Información de certificado.....	187
Figura 122. Detalle de certificado.	188

RESUMEN

En la actualidad una falencia de las empresas es la burocracia sobre sus procesos en cada una de sus áreas, el uso excesivo de documentos compartidos para llevar un supuesto control de su trabajo diario y la falta de sistemas específicos, ha llevado a que muchas empresas presenten pérdidas económicas y baja calidad en la atención a sus clientes. El objetivo principal de la sistematización de procesos, es proveer mayor fluidez a los mismos dentro de un flujo organizado definiendo claramente los alcances del proceso y darle un valor agregado al trabajo que se realiza en dicha área mejorando los indicadores de atención al cliente. Por estas razones se implementó un sistema web en la empresa WICOM del Ecuador, utilizando metodología OOHDM para el análisis y diseño de información destinado a la programación orientada a objetos y sistematizar el proceso “GESTIÓN DE SOLICITUDES DE HELPDESK EMPRESARIAL” para obtener una mejora en los indicadores de “ATENCIÓN AL CLIENTE” mediante un sistema utilizando tecnologías de programación web actuales, que permite obtener un rendimiento y adaptabilidad alta, garantizando la posterior satisfacción en su amplia cartera de clientes, obteniendo resultados como la reducción de tiempos de atención, subir el nivel de satisfacción del cliente y demostrar el alto rendimiento y funcionalidad de las herramientas de libre distribución para generar proyectos de gama amplia.

PALABRAS CLAVE:

- **SISTEMATIZACIÓN**
- **PROCESOS**
- **METODOLOGÍA OOHDM**
- **DESARROLLO**
- **SISTEMAS**
- **WEB**

ABSTRACT

At present, one shortcoming of the companies is the bureaucracy about their processes in each of their areas, excessive use of shared documents to keep control of their daily course work and lack of specific systems, has led many companies present economic losses and low quality of care to their clients. The main objective of the systematization of processes, is to provide greater fluidity to them within an organized flow clearly defining the scope of the process and give added value to the work done in this area by improving customer service indicators. For these reasons a web system was implemented in the company WICOM of Ecuador, using OOHDM methodology for the analysis and design of information intended for OOP and systematize the process "HELPDESK MANAGEMENT BUSINESS APPLICATIONS" for an improvement in the indicators "CUSTOMER" by a system using current web programming technologies that allows us to obtain high performance and adaptability, ensuring satisfaction in his subsequent large client, obtaining results like reducing service times, raise the level of customer satisfaction and demonstrate high performance and functionality freely available tools to generate projects range.

KEYWORDS:

- **SYSTEMATIZATION**
- **PROCESSES**
- **METHODOLOGY**
- **DEVELOPMENT**
- **SYSTEMS**
- **WEB**

CAPÍTULO 1

INTRODUCCIÓN

En la actualidad contar con Sistemas de Información aumenta la competitividad en empresas de toda índole, esto implica conocer en todo momento qué es lo que sucede en los diferentes pasos de un determinado proceso, permite delimitar controles para eventos propios o ajenos a la compañía, establecer políticas y normas que llevan a obtener, productos o servicios de calidad.

La implementación de un Sistema Informático lleva a la compañía a tener una gestión adecuada y de calidad con su Información, reducción de tiempos para obtención de informes, estadísticas, indicadores, etc. Facilitando de esta manera la toma de decisiones, realizando proyecciones basados en los históricos, los mismos que estarán disponibles a toda hora y en todo momento.

En nuestro mercado, se comercializan sistemas de información, que pretenden satisfacer las necesidades de empresas, sistemas genéricos predispuestos a adaptarse al giro del negocio, pero estos no cubren todas las expectativas, tienden a alejarse de la necesidad real, especialmente en los procesos propios de cada compañía; por tanto adquirir un sistema con estas características no tendría sentido si se lo analiza desde el punto de vista del beneficio real.

Se necesita elaborar un sistema que tome en cuenta otros factores personalizados que satisfagan las perspectivas del nivel gerencial y apoyen al nivel medio, por lo que se ha visto indispensable la construcción de una herramienta informática, que permita estructurar un Sistema de Control eficaz y eficiente.

Esta solución está orientada a proporcionar esa herramienta, utilizando métodos y técnicas de Ingeniería de Software que logren un producto de calidad, en funcionalidad, oportunidad y costo.

1.1 Planteamiento del Problema

En la empresa WICOM del Ecuador, se ha detectado que la metodología para el control de Soporte brindado a sus clientes en sus diferentes canales de servicios, no es la adecuada, obteniendo así resultados negativos con sus clientes, descontrol en la carga de trabajo de sus técnicos de soporte, por ende no se conoce si todas sus solicitudes son finalizadas adecuadamente y existiendo valores inexactos de facturación por dichos trabajos.

Causas

- Bitácora en Excel compartida mediante red.
- Técnicos de soporte olvidan sus pendientes.
- No hay un registro exacto en tiempo, desde que se abrió una solicitud hasta que se la da por terminada.
- Duplicidad de solicitudes.
- Duplicidad de solicitudes en facturación a los clientes.
- Desconocimiento de carga de trabajo en los técnicos de soporte.
- Desconocimiento absoluto de histórico en soluciones a clientes.

De continuar este manejo ineficiente en el canal de soporte a clientes, la compañía tendrá una pérdida importante de clientes y por ende de ingresos económicos.

Por tal motivo se ha solicitado la automatización de este proceso fundamental en el giro de negocio de la empresa WICOM del Ecuador.

1.2 Justificación

A continuación se detallará las razones que justifican este proyecto:

- Los clientes podrán tener información acerca de sus solicitudes abiertas de manera rápida y oportuna.

- Se podrá tener información histórica por cliente, con el fin de realizar seguimientos constantes de los casos, mejorar los niveles de atención y los tiempos de respuestas.
- Los clientes podrán obtener información de sus contratos de soporte, fechas de expiración y prevenir una nueva renovación del mismo.
- El cliente podrá revisar los ítems que se le está facturando, ya que dispondrá de la base completa de todas sus solicitudes.
- En caso de no implementar el sistema el canal de Soporte al Cliente se volverá una debilidad de WICOM del Ecuador, generando pérdidas económicas importantes.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar e Implementar un Sistema Web aplicando la metodología de desarrollo OOHDM, para el registro, almacenamiento y gestión de información precisa y confiable de las solicitudes de soporte al cliente de la empresa WICOM del Ecuador, realizando un análisis del efecto de la metodología sobre este tipo de proyectos.

1.3.2 Objetivos Específicos

- Analizar y computarizar el proceso de Soporte al Cliente realizado en la empresa WICOM del Ecuador.
- Construir una aplicación de ambiente WEB para que Supervisores, Técnicos y Clientes puedan acceder a una información en línea, generando una interactividad real entre los participantes.
- Administrar una Base de Clientes y una base de Conocimiento con la resolución de varios tipos de problemas ya tratados en Soporte al Cliente de WICOM del Ecuador.

- Analizar el efecto de la aplicación de la metodología indicada en este tipo de proyectos orientados al entorno web.

1.4 Alcance

El sistema Web que se implementa para la empresa WICOM del Ecuador cuenta con las siguientes funcionalidades:

- Esquema lógico de seguridad de usuarios robusta.
- Caducidad de sesiones de trabajo.
- Manejo de perfiles de usuarios a nivel de menús y acciones.
- Interface de usuario amigable de fácil uso.
- Generación de reportes estadísticos gerenciales para el control de tiempos de atención y calificación de atención.
- Creación de n Departamentos o Áreas de trabajo a las que pertenecen sus clientes “Gerencia, Contabilidad, Sistemas, Marketing, etc.”
- Creación de n Cargos para usuarios “Ingeniero, Doctor, Abogado, Gerente, Administrador, etc.”.
- Creación de n Usuarios, Empresas, Clientes y Categorías de Solicitudes.
- Caducidad de tiempo de soporte por empresa o usuario.
- Control de acceso a la web.
- Ingreso de solicitudes mediante personal en call center.
- Ingreso de solicitudes mediante intranet/internet de sus clientes.
- Asignación de solicitudes a personal de soporte con control de tiempos de asignación, atención y finalización.
- Charla interactiva mediante un mini foro por solicitud ingresada.
- Impresión de solicitudes.
- Alertas de solicitudes vía mail en el ingreso, atención, finalización, etc.
- Reportes de usuarios, empresas, solicitudes, calificaciones, etc.
- Filtros de búsqueda y ordenamiento interactivo.

- Auditoria.
- Instalación y configuración de servidor web y base de datos.

1.5 Factibilidad

1.5.1 Factibilidad Técnica

El desarrollo de un Sistema Web para la Gestión de Solicitudes Online para servicio al Cliente de la empresa WICOM del Ecuador, requiere de una plataforma tecnológica y conocimiento que de soporte a todas las necesidades del mismo.

Por la experiencia recabada durante 8 años de trabajo en diseño y desarrollo de sistemas informáticos, y la instrucción recibida en la Universidad de las Fuerzas Armadas ESPE avala los conocimientos suficientes para la construcción del sistema en mención.

En relación a Sistemas Web el manejo de recursos y la integración se recomienda para que su correcta implementación y funcionamiento este garantizado, usar un servidor web de arquitectura LAMP (Linux, Apache, MySQL, PHP).

Apache Friends "<http://www.apachefriends.org>" proveen servidores de uso libre y con mucha estabilidad, por lo cual se utiliza XAMPP 1.6.8

1.5.2 Factibilidad Tecnológica

La empresa WICOM del Ecuador cuenta con una infraestructura completa ya que es una organización que se dedica a consultoría informática:

- Hardware.- La empresa cuenta con equipos de una moderna tecnología, capaz de soportar la implantación del sistema web.
- Software.- Para el desarrollo del sistema se va a utilizar Software libre.

- Sistema de Comunicación.- Se cuenta con una infraestructura de red capaz de soportar la implementación del sistema web a realizar.
- Información.- Existe toda la información y la disposición a entregarla en fin de agilizar el desarrollo del proyecto.

1.5.3 Factibilidad Operativa

La operatividad del Sistema de Gestión de Solicitudes Online, está garantizada mediante su tolerancia orientada a soportar fallos críticos, una política clara de respaldos diarios programados de la Base de Datos así como su replicación en varios medios.

Su diseño modular y distribuido lo hace más consistente al momento de presentar un fallo, su auditoria interna da una garantía de contar con un registro de los incidentes que podrían generar errores.

Estas características harán del Sistema de Gestión de Solicitudes Online tenga una disponibilidad alta a través del tiempo.

1.5.4 Factibilidad Operacional

El sistema de Gestión de Solicitudes Online Para la Empresa WICOM del Ecuador, tendrá una interface amigable, con pocas opciones y procesos bien definidos, los mismos que compondrán una interface sencilla de uso, el manejo de plantillas para el diseño, dará la facilidad de cambiar el entorno visual muy fácilmente, de igual manera se cuenta con burbujas de ayuda en procesos críticos o de poca comprensión, formularios claros y bien organizados, un motor de reportería general que mantendrá la metodología de uso en todo el sistema.

Por cuanto al personal de la empresa WICOM del Ecuador recibirá una completa capacitación en un ambiente de pruebas y al final del mismo recibirá un adecuado manual de uso el mismo que les servirá de apoyo en su trabajo diario.

La mayoría de personal que trabaja en el área de servicio al cliente oscila entre las edades 22 a 26 años, gente que está acostumbrada a manejo de información en ambientes web y buscadores, lo que da a entender que la oposición al cambio será mínima.

1.5.5 Factibilidad Económica

A continuación se detallan los valores correspondientes a los recursos que se van a emplear en el proyecto:

Tabla 1

Factibilidad Económica

RECURSO	COSTO
Humano	2800.00 <i>usd</i>
Servidor Web	1200.00 <i>usd</i>
1 Portátil	800.00 <i>usd</i>
Base de Datos “MySql 5”	0.00 <i>usd</i>
Sistema Operativo “Centos 6 x64”	0.00 <i>usd</i>
Herramientas de Diseño “NetBeans”	0.00 <i>usd</i>
Varios	1000.00 <i>usd</i>
TOTAL	5800.00 <i>usd</i>

La empresa WICOM del Ecuador cuenta con todo el apoyo para la inversión e implantación del proyecto que requiera el equipo de trabajo.

1.5.6 Factibilidad Legal

Para la construcción del Sistema de Gestión de Solicitudes Online para la empresa WICOM del Ecuador, se ha seleccionado la utilización de

herramientas y lenguajes bajo licencias GNU, FSF y OSI, esto garantiza la legalidad total en el proyecto, este tipo de proyectos son bien vistos por el Estado Ecuatoriano, el mismo que cuenta con una ley sobre la utilización de software libre en las empresas públicas.

En cuanto a Derechos de Autoría se puede referir a la ley Ecuatoriana sobre derechos de Autor Informático la misma que dicta:

- Art. 28.- Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa. (Ley de la propiedad intelectual, 1998)
- Art. 29.- Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. (Ley de la propiedad intelectual, 1998)

De lo que se obtiene como conclusión que la empresa auspiciante tendrá los derechos de autor. De igual forma dentro del plan del buen vivir 2013-2017 se puede destacar el artículo 9.2 en el párrafo e que dicta lo siguiente:

- ART. 9.2 e.- Priorizar el desarrollo de iniciativas económicas vinculadas al desarrollo de las TIC, aprovechando las capacidades desarrolladas en software y de los recursos de la biodiversidad, creando espacios e infraestructura pertinente, que sustenten su productividad. (Plan Nacional del Buen Vivir, 2013)

- Art. 12.3 f.- Promocionar a nivel internacional la oferta de servicios ecuatorianos dando énfasis a los sectores priorizados como el turismo, transporte y logística, servicios ambientales y software. (Plan Nacional del Buen Vivir, 2013)

1.6 Metodología

Los siguientes estándares y metodologías que se detallan a continuación son los que se aplicarán en el proyecto:

- De la metodología OOHDM u Object Oriented Hypermedia Design Methodology, para diseño de aplicaciones hipermedia y para la Web, que fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa y es una extensión de HDM con orientación a objetos, se utilizará para los diagramas navegacionales.
- Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.
- IEEE 830 es un estándar para las especificaciones de los requisitos del software.

1.7 Herramientas

Las herramientas a utilizar en el desarrollo de la aplicación web se detallan a continuación y su función:

- *Netbeans*.- IDE de desarrollo, principalmente enfocado al lenguaje JAVA, pero su gran variedad de módulos y versatilidad, su libre uso, gratuito y sin restricciones lo hace el candidato perfecto para el desarrollo del sistema.

- *Apache*.- Servidor web de código abierto, ideal para la publicación de aplicaciones web desarrolladas en PHP, su robusta funcionalidad permite tener un ambiente seguro y estable para nuestra aplicación.
- *PHP*.- Lenguaje de programación del lado de servidor, muy versátil tanto para la creación de ambientes gráficos o ambientes de línea de comandos, con mucha información y librerías de código abierto.
- *JavaScript*.- Lenguaje del lado del cliente utilizado principalmente para mejorar la interfaz gráfica y la validación de la información pre procesamiento en el lado del servidor, se convierte en una herramienta indispensable dentro del proyecto.
- *Ajax*.- JavaScript asincrónico y XML, técnica de programación sobre JavaScript necesario para el desarrollo de aplicaciones, debido a la facilidad que brinda al usuario al momento de utilizar el portal, el programa se ejecuta del lado del cliente, de esta manera se dispone de velocidad de respuesta, rendimiento e interactividad para el usuario.
- *MySQL*.- Sistema de gestión de base de datos, robusta y confiable, mantiene la información íntegra. Este Motor de Base de Datos es relacional multihilo y multiusuario, características necesarias para la elaboración de un sistema que aproveche recursos de software y hardware.
- *MySQL Workbench*.- Herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.
- *StarUML*.- Herramienta case intuitiva y estructurada, que permite el modelamiento de datos con UML, arquitectura, metadatos, etc., para el diseño y análisis del software de manera fácil y ágil,

CAPÍTULO 2

MARCO TEÓRICO

En la actualidad la globalización y los nuevos avances tecnológicos ha permitido que las pequeñas y grandes empresas se proyectaron a un futuro novedoso, competitivo y productivo, el Internet formo parte de la vida cotidiana, exponiendo a cada una de las empresas a publicar información confiable, segura y productiva para todo público obteniendo una herramienta para gestionar planes estratégicos de mercadeo, producción y ventas en línea.

El internet brinda una alta gama de servicios donde se puede publicar y explorar información, optimizando tiempo y recursos en las empresas. La (World Wide Web) más conocida como WWW es un protocolo que ayuda para la publicación de Sitios Web.

2.1 Metodología del Desarrollo de Sistemas

Para desarrollar una aplicación de estas características se ha considerado un enfoque disciplinado y sistemático, en referencia a de las diferentes metodologías que se pueden aplicar, con el fin de dividir los procesos de modelamiento de metodologías orientadas a objetos.

La programación orientada a objetos es muy importante en el aspecto de “Modelado del sistema” puesto que es un conjunto de objetos, que actúan entre sí, empleando términos tales como abstracción, encapsulamiento, herencia y polimorfismo.

La metodología orientada a objetos se basa en construir objetos que contienen atributos y operaciones de manera que cubra las necesidades que requiere la aplicación, una de las principales ventajas es el concepto de encapsulamiento conocido también como protección de datos mediante el cual solo se pueden modificar los datos de un objeto accediendo a través de sus métodos y operaciones.

Las metodologías orientadas a objetos que se ajustan a las especificaciones y requerimientos de la empresa WICOM del Ecuador son las siguientes: para el modelado conceptual se utilizará RUP (Proceso Unificado de Software) más UML (Lenguaje Unificado de Modelado) que en conjunto constituyen una metodología estándar utilizada para el análisis, implementación y documentación de sistemas, para el modelo de navegación OOHDM (Metodología de diseño hipermedia orientado a objetos), para la especificación de los requisitos del software (IEEE830).

El proceso de desarrollo de software es un conjunto de información que se adaptan para transformar necesidades de un usuario a los requerimientos específicos para la obtención de un sistema orientado a objetos.

2.1.1 Lenguaje Unificado de Modelado “UML”

UML es un lenguaje de notación y meta modelo utilizado para el desarrollo de sistemas por medio de conceptos orientados a objetos. UML unifica una serie de métodos de análisis y diseños orientados a objetos, UML es un lenguaje de modelo pero no una metodología.

UML se encuentra definido por dos partes esenciales notación y meta modelo. La Notación a la parte gráfica que se puede visualizar en los modelos, es decir la sintaxis del lenguaje del modelado como por ejemplo las clases. El meta modelo es la aplicación de varias notaciones en el que se puede expresar a través de diagramas como por ejemplo: diagramas de clases.

UML permite la modificación de todos sus miembros mediante estereotipos y restricciones. El estereotipo permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. La restricción identifica el comportamiento forzado de una clase o relación, es decir mediante la restricción se está forzando el comportamiento que debe tener el objeto al que se le aplica.

Cada diagrama de UML utiliza una notación pertinente y la suma de estos diagramas crean diferentes vistas. Estas vistas existentes en UML son:

- **Vista de casos de uso:** Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- **Vista de diseño:** Se forma con los diagramas de clases, objetos, colaboración, estado y actividades.
- **Vista de procesos:** Se forma con los diagramas de las vistas de diseño recalando las clases y objetos referentes a procesos.
- **Vista de implementación:** Se forma con los diagramas de componentes, colaboración, estados y actividades.
- **Vista de despliegue:** Se forma con los diagramas de despliegue, interacción, estados y actividades.

2.1.1.1 Modelos

Un modelo representa una vista específica para la construcción de un sistema de software, cada modelo representa fases de procesos diferentes pero consecuentes en línea de información.

UML dispone de dos tipos de diagramas, diagramas estáticos y diagramas dinámicos.

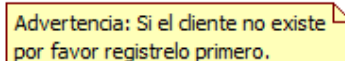
- Diagramas Estáticos
 - Diagramas de Casos de Uso
 - Diagramas de Clases
 - Diagramas de Objetos
 - Diagramas de Componentes
 - Diagramas de Despliegue
- Diagramas Dinámicos
 - Diagramas de interacción
 - Diagramas de Estados
 - Diagramas de Actividad

2.1.1.2 Elementos comunes en los Diagramas

A. Notas

Una nota es un símbolo que puede ser usado en todos los modelos de diagramas, tienen la función de contener anotaciones, comentarios, restricciones, el cuerpo de un procedimiento o un valor rotulado (tagged value).

Una nota se representa con un rectángulo y en la parte superior derecha posee una de sus esquina dobladas. Como se muestra en la Figura 1.



Advertencia: Si el cliente no existe
por favor regístrelo primero.

Figura 1. Representación de Notas en UML

B. Agrupación de elementos mediante paquetes

Un paquete es un mecanismo de propósito general para organizar elementos en grupos. Los grupos de elementos son estructurales o de comportamiento estos elementos pueden incluirse en un paquete, pero lo interesante de esto, es que se pueden agrupar paquetes dentro de otro paquete.

Un paquete se representa como un rectángulo grande y en su esquina superior izquierda tiene un pequeño rectángulo.

Se pueden indicar relaciones de dependencia entre paquetes mediante una flecha con la línea entre cortada, pero si el paquete A depende del paquete B, la simbología será una flecha de A a B como se ilustra en la Figura 2.

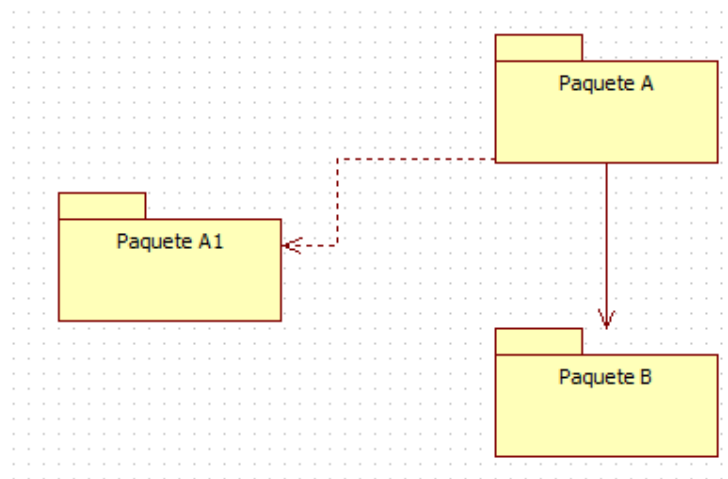


Figura 2. Relaciones de Dependencia

C. Diagramas de Estructura Estática

Los Diagramas de Estructura Estática engloba al Modelo Conceptual de la fase de Diseño como al Diagrama de Clases de Diseño. Estos diagramas conceptualmente son distintos, el primero modela elementos de dominio mientras que el segundo presenta los elementos de la solución del sistema. Sin embargo, ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

D. Clases

Una clase es un artefacto de modelado que describe una serie de objetos que comparten los mismos atributos, operaciones, relaciones y semántica, define la estructura y comportamiento de los objetos denominados instancia de la clase.

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones, como se representa en la Figura 3.

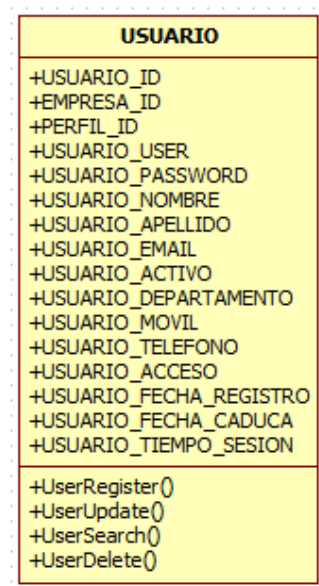


Figura 3. Modelo de una clase en UML

E. Objeto

Es una instancia de una clase, que se caracteriza por tener una identidad única, definido por un conjunto de valores de atributos y un comportamiento representado por sus operaciones y métodos.

F. Asociaciones

Las asociaciones son los mecanismos por el cual permiten a los objetos mantener una comunicación, una asociación puede ser unidireccional o bidireccional, es decir, establece una relación en la cual pueden intercambiar mensajes entre sí, estableciendo un valor de multiplicidad, indica cuantos objetos se encuentran relacionados de un extremo con un objeto del extremo contrario.

Las asociaciones se representan mediante una línea continua, que unen las relaciones que existen entre las clases. Esta línea puede tener una serie de elementos gráficos como: nombre, multiplicidad, rol y agregación, que expresan características particulares de la asociación, como se muestra en la Figura 4.

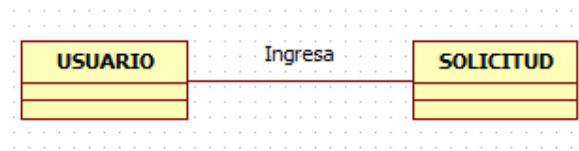


Figura 4. Representación de asociaciones

G. Multiplicidad

La multiplicidad es una restricción que se aplica a una asociación, limita el número de instancias de una clase con otra. La multiplicidad define la cantidad de objetos que participan en la relación y pueden ser:

Tabla 2

Multiplicidad de asociaciones

Multiplicidad	Descripción
1	Solo 1
1..*	Asociación de 1 al infinito
1..n	Asociación de 1 a n
*	Asociación de 0 al infinito

En la Figura 5 se puede observar la relación existente entre dos clases, con una instancia de 1 una asociación a 1* varias asociaciones.

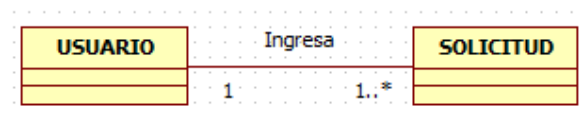


Figura 5. Representación de multiplicidad

H. Roles

Indica el papel que desempeña una clase en una asociación por el cual se especifica el nombre del rol. La representación de un rol, es en el extremo de la asociación junto a la clase que desempeña dicho rol como se muestra en la Figura 6.

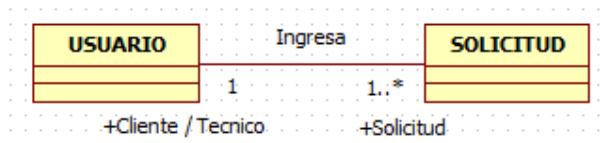


Figura 6. Representación de roles en una asociación

I. Agregación

Representa una relación que parte de objetos. El símbolo de agregación es un diamante colocado en el extremo de la clase que representa el "todo" como se muestra en la Figura 7.

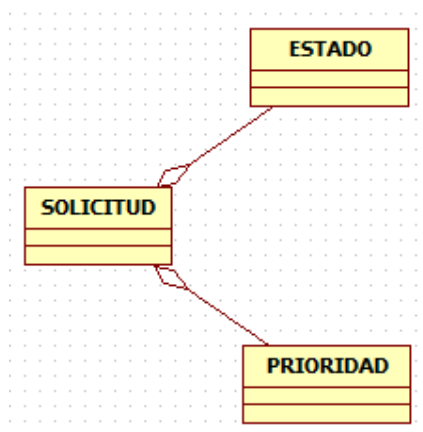


Figura 7. Representación de agregación

J. Navegabilidad

Se representan por flechas en un extremo de una asociación donde indican la responsabilidad o la capacidad de iniciar una actividad de navegabilidad. Esto significa que es posible "navegar" desde el objeto de la

clase origen hasta el objeto de la clase destino. Este concepto de diseño indica que un objeto de la clase origen conoce al objeto de la clase destino, por tanto puede llamar a alguna de sus operaciones.

K. Herencia

La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre” como se muestra en la Figura 8.

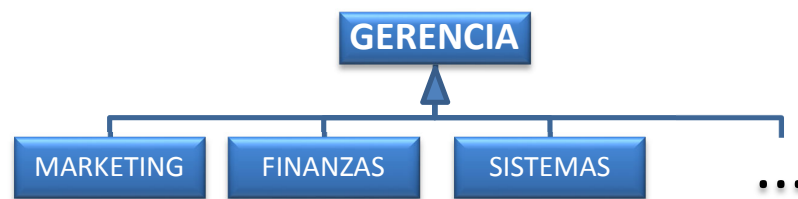


Figura 8. Modelo de herencia

En el caso de tener una relación de herencia con varias clases subordinadas no es necesario poner todas las clases por lo que, se pueden plasmar las más importantes y las otras representarlas como puntos suspensivos.

2.1.1.3 Diagramas Estáticos

A. Diagramas de Casos de Uso

Los diagramas de casos de uso son una pieza de funcionalidad con la que el usuario interactúa con el sistema.

- Un caso de uso capta los requisitos específicos de los usuarios
- Un caso de uso se diseña, se implementa y se prueba.
- Un caso de uso alcanza el objetivo que requiere el usuario.

Un caso de uso es desarrollado de acuerdo a la información recopilada mediante reuniones realizadas con cliente.

Los elementos que componen los casos de uso son: Actores, casos de uso y relación entre casos de uso.

- **Actores**

Un actor es una entidad externa al sistema que interacciona con el sistema. Los actores se representan mediante una figura humana, con el fin de definir el papel o rol, al que pertenece dentro del sistema. Los actores pueden ser gente real (por ejemplo, usuarios del sistema) que inclusive interactúan en otros sistemas.

- **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, es necesario tener en cuenta que el actor realiza una tarea específica dentro de la aplicación, expresando una unidad coherente y funcional.

Los diagramas de Casos de Uso se representan mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema, como se muestra en la Figura 9.

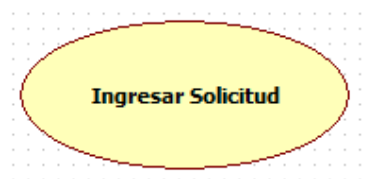


Figura 9. Caso de Uso

- **Relaciones entre casos de Uso**

Los términos “Uses” y “Extends” representan las relaciones entre dos casos de uso. “Extends” se utiliza entre los casos de uso que realizan las mismas funciones, pero una especializa a la otra extendiendo su funcionalidad.

La relación “Uses” ocurre cuando un caso de uso utiliza un comportamiento similar en varios casos de uso por lo que no requiere repetir la misma funcionalidad en otro caso de uso.

La representación gráfica de “Uses” y “Extends” es una flecha en forma de triángulo y con una etiqueta <<extends>> o <<uses>> según el tipo de relación que se desee representar.

Es importante mencionar que la relación “extends” describe variaciones de una conducta normal mientras la relación “uses” se da cuando existen repeticiones de conductas de varios casos de uso.

A continuación se muestra en la Figura 10 la representación de relación “uses” y en la Figura 11 la relación de “extends”

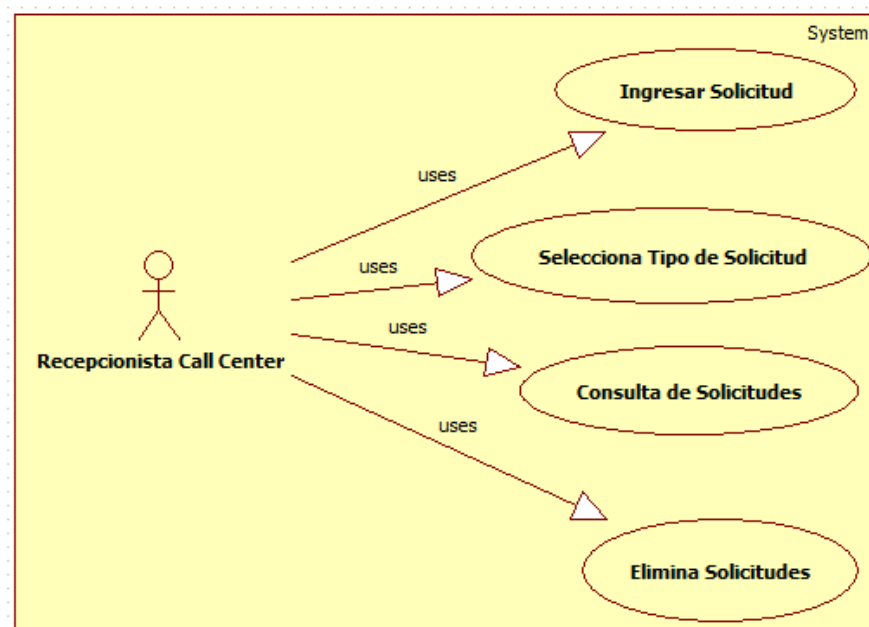


Figura 10. Representación de relación “Uses”

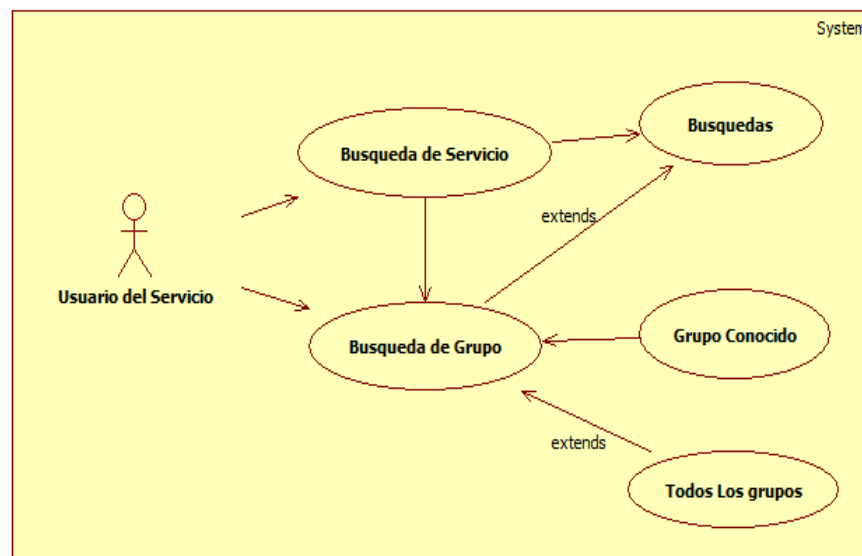


Figura 11. Representación con relación “Extends”

B. Diagramas de Clases

Los diagramas de clases describen los tipos de objetos, la variedad de clases y relaciones que existe en un sistema.

Los diagramas de clases también se los conoce como diagramas estáticos, ya que se componen de un conjunto de asociaciones de objetos, atributos, métodos, operaciones, navegabilidad, dependencias y restricciones para la construcción de una clase como se muestra en la Figura 12.

▪ Asociación

Representa la relación entre instancias de clase, esta relación tiene una multiplicidad la cual define la cantidad de objetos que participan en la relación.

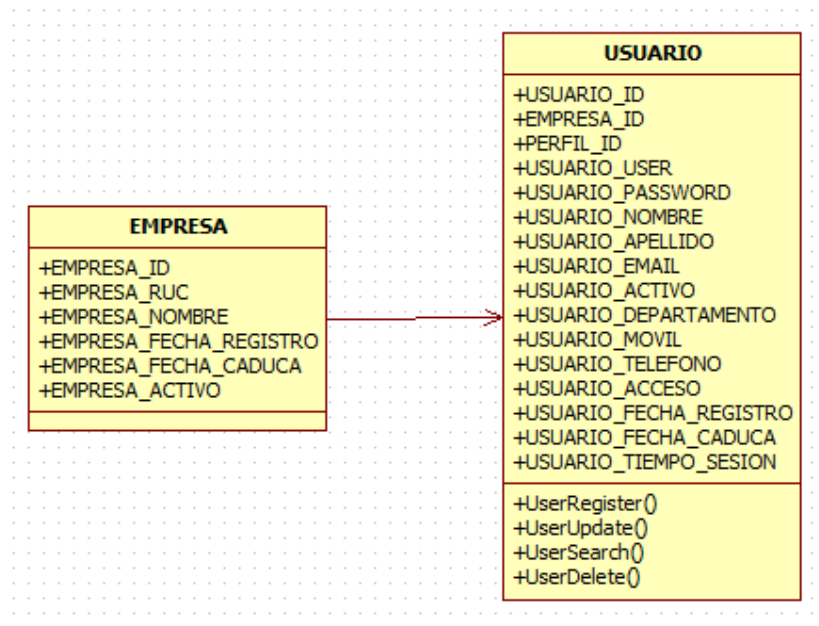


Figura 12. Diagrama de clases

▪ Navegabilidad

La navegabilidad es una propiedad de un rol, pueden ser unidireccionales o bidireccionales a través de la asociación, desde objetos de la clase origen a objetos de la clase destino. Poseen responsabilidad o capacidad para iniciar una actividad.

▪ Atributos

Representa una propiedad o característica de una clase. Cada atributo de un objeto tiene un valor que pertenece a un dominio de valores determinados, como se muestra en la Figura 13.

Su sintaxis es:

Visibilidad nombre: tipo = valor inicial

La visibilidad puede ser:

- + Indica atributos públicos

- # Indica atributos protegidos
- - Indica atributos privados

USUARIO
+USUARIO_ID
+EMPRESA_ID
+PERFIL_ID
+USUARIO_USER
+USUARIO_PASSWORD
+USUARIO_NOMBRE
+USUARIO_APELLIDO
+USUARIO_EMAIL
+USUARIO_ACTIVO
+USUARIO_DEPARTAMENTO
+USUARIO_MOVIL
+USUARIO_TELEFONO
+USUARIO_ACCESO
+USUARIO_FECHA_REGISTRO
+USUARIO_FECHA_CADUCA
+USUARIO_TIEMPO_SESION

Figura 13. Atributos

▪ Operación

Son los procesos que definen el comportamiento de los objetos de una clase, como se muestra en la Figura 14.

Su sintaxis es:

Visibilidad nombre (lista de parámetros): tipo de dato que retorna {propiedades}

La visibilidad puede ser:

- + público
- # protegido
- - privado

La operación invoca al objeto mientras que el método es el cuerpo de la operación esta diferencia es propia del polimorfismo.

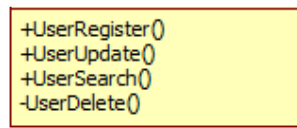


Figura 14. Operación

▪ Generalización

La generalización denomina una clase llamada Superclase que es la clase general, y otra llamada súper tipo, estas dos clases poseen características similares.

La relación de generalización denota una relación de herencia entre clases. Se representa dibujando un triángulo sin rellenar en el lado de la superclase como se muestra en la Figura 15. La subclase hereda todos los atributos y mensajes descritos a la clase súper tipo.

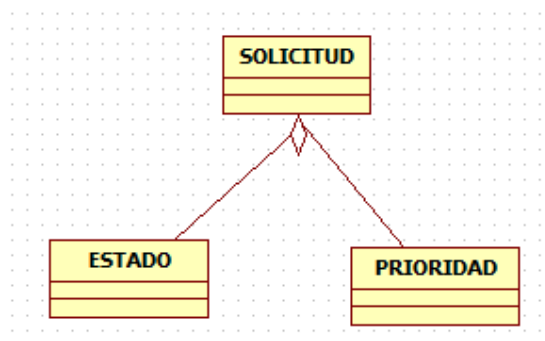


Figura 15. Generalización

▪ Dependencia

Es la relación en la que existen entidades independientes y dependientes, lo que implica que al cambiar el elemento independiente puede requerir cambios en los dependientes. Se representa con una línea punteada direccional, indicando el sentido de la dependencia, como se muestra en la Figura 16.



Figura 16. Dependencia

▪ Reglas de Restricción

Las restricciones definen las limitaciones que pueden existir en las clases. Se representan dibujando llaves a los extremos y la sintaxis puede ser escrita en lenguaje informal.

C. Diagramas de Objetos

Los diagramas de Objetos modelan las instancias de elementos contenidos en los diagramas de clases, muestran un conjunto de objetos y sus relaciones en un situación concreta en un momento determinado.

Los diagramas de objetos usan un sub conjunto de elementos de un diagrama de clase para enfatizar la relación entre las instancias de las clases, no muestran nada diferente en su arquitectura a los diagramas de secuencia, pero reflejan multiplicidad y roles, como se muestra en la Figura 17.



Figura 17. Diagrama de Objetos

D. Diagramas de Componentes

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, comunicación, ubicación y otras condiciones. Los componentes también pueden exponer

las interfaces, las cuales indican los puntos de entrada o los servicios que un componente puede ofrecer.

Un componente está compuesto por numerosas clases y paquetes de clases internas, como se muestra en la Figura 18.

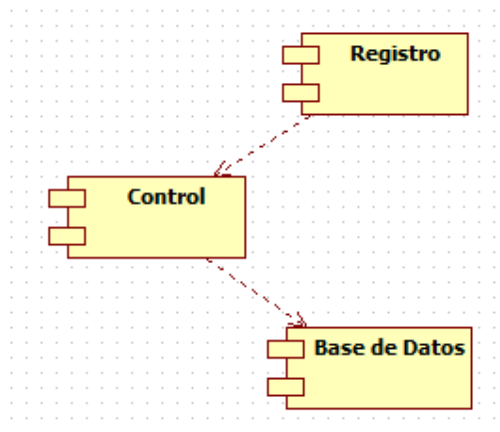


Figura 18. Diagrama de Componentes

E. Diagramas de Despliegue

El diagrama de despliegue define las relaciones físicas entre los componentes hardware y software en el sistema, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Se encuentran formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deberán detallarse en el diagrama de componentes).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación, como se muestra en la Figura 19. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (un componente puede tener más de una interfaz).

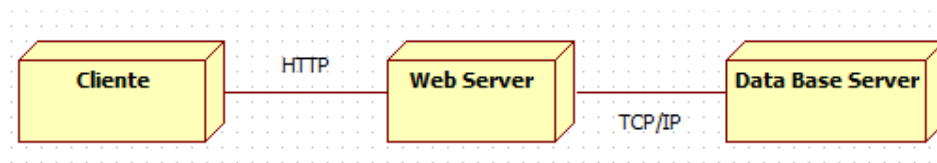


Figura 19. Diagrama de Despliegue.

2.1.1.4 Diagramas Dinámicos

A. Diagramas de interacción

Los diagramas de interacción muestran un patrón de interacción entre objetos, siendo un grupo de objetos que colaboran entre sí, mostrando cierto número de objetos y mensajes que comunican a los objetos dentro de un diagrama.

Existen dos tipos de diagrama de interacción basados en la misma información, pero cada uno enfatiza un aspecto particular.

- Diagramas de Secuencia
- Diagramas de Colaboración

B. Diagramas de Secuencia

Los diagramas de secuencia son de mucha utilidad ya que permiten visualizar a detalle cómo interactúan los objetos de una forma ordenada según la secuencia temporal de eventos. En general estos diagramas, muestran los objetos participantes en la interacción, como líneas de vida a lo largo del tiempo y sus interacciones en el tiempo, como mensajes intercambiados ordenadamente según su secuencia en el tiempo, como se muestra en la Figura 20.

Los objetos y actores participantes en la interacción se los representa por medio de cajas en el eje horizontal, en el eje vertical se encuentra la línea de vida representada por una línea punteada que sale desde la caja y los mensajes se los representa mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo, también se pueden colocar etiquetas (como

restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

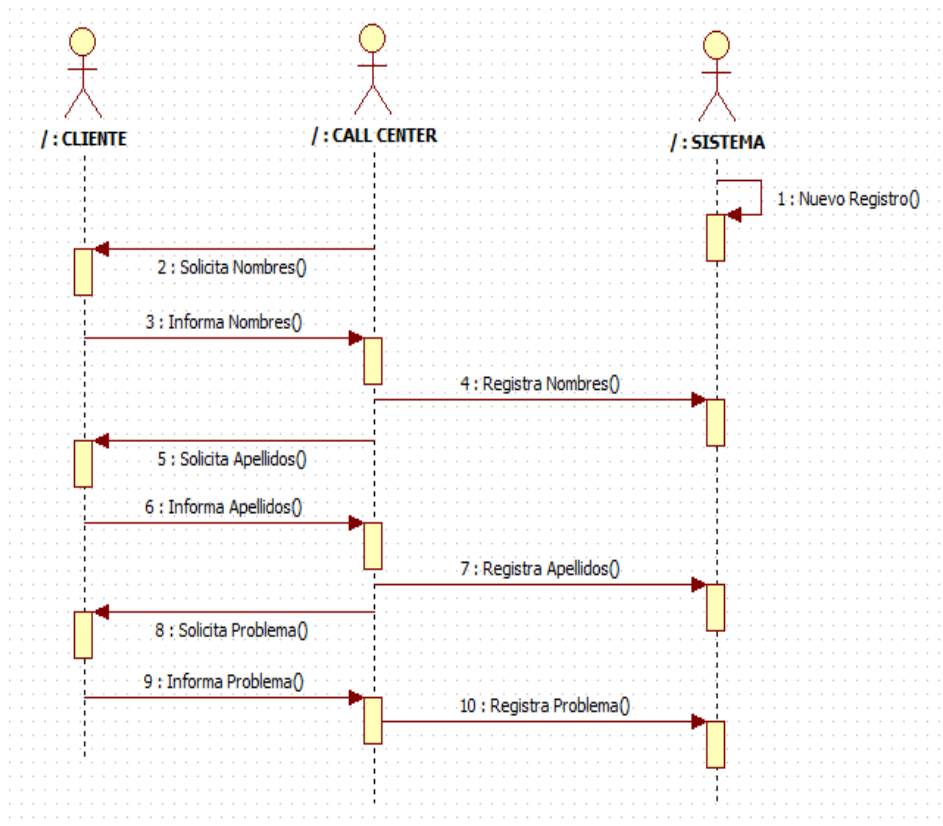


Figura 20. Diagrama de Secuencia

C. Diagramas de Colaboración

El diagrama de colaboración muestra gráficamente una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos visualizando el orden de concurrencia de las actividades, como se muestra en la Figura 21.

El diagrama de colaboración utiliza los mismos marcadores que el diagrama de secuencia.

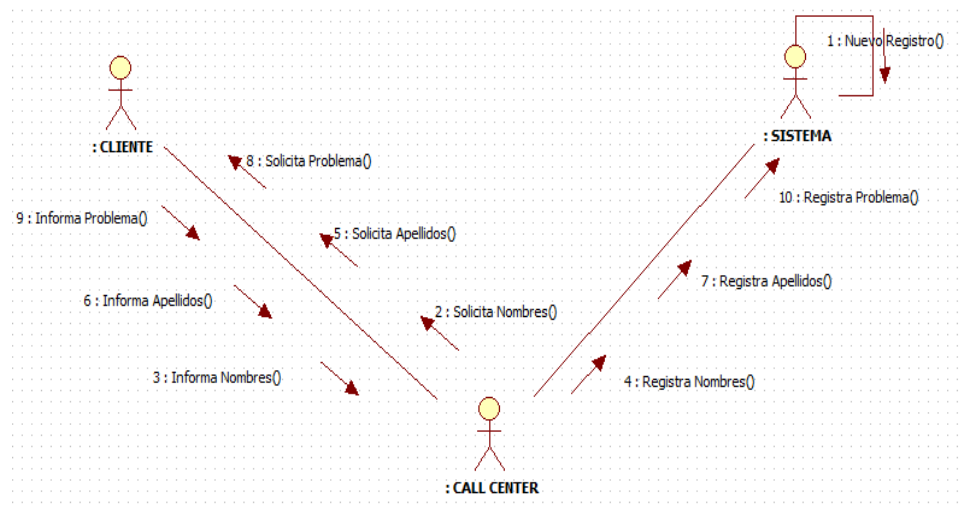


Figura 21. Diagrama de Colaboración.

D. Diagramas de Estados

Los diagramas de estado definen el comportamiento de los objetos del sistema a través de diferentes estados, eventos y transiciones.

- **Evento:** Acontecimiento importante para el sistema
- **Estado:** Condición de un objeto en un momento determinado: el tiempo que transcurre entre eventos
- **Transición:** Es una relación entre dos estados, e indica que, cuando ocurre un evento, el objeto pasa del estado anterior al siguiente.

Los diagramas de estado son utilizados para mostrar el comportamiento de un objeto a través de las diferentes etapas de vida que tiene el objeto.

Los estados se representan mediante óvalos, las transiciones mediante flechas con el nombre del evento respectivo. Estos diagramas poseen dos estados especiales uno inicial y otro final, su representación gráfica para el estado inicial es un círculo negro mientras que el estado final un círculo grande y en el centro un círculo relleno, como se muestra en la Figura 22.

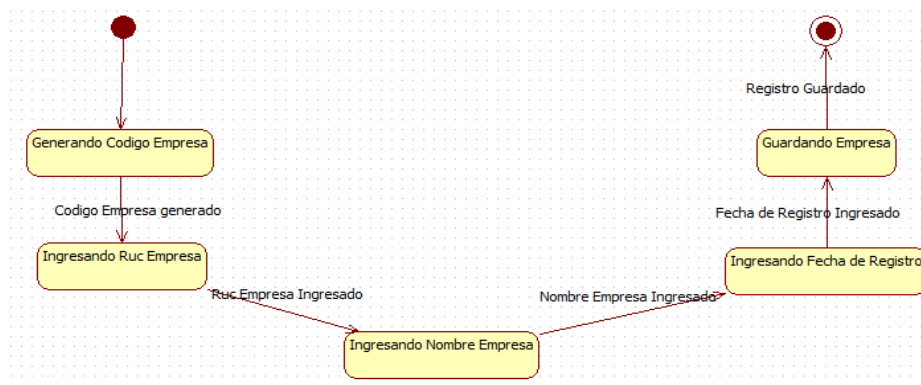


Figura 22. Diagrama de Estado.

E. Diagramas de Actividad

Los diagramas de actividad muestran el orden de cómo se va realizando las tareas dentro de un sistema, es decir define el flujo de control de actividades, también ayuda con la descripción del comportamiento para los procesos en paralelo.

La sintaxis para las transiciones es similar al diagrama de estado, como se muestra en la siguiente Figura 23.

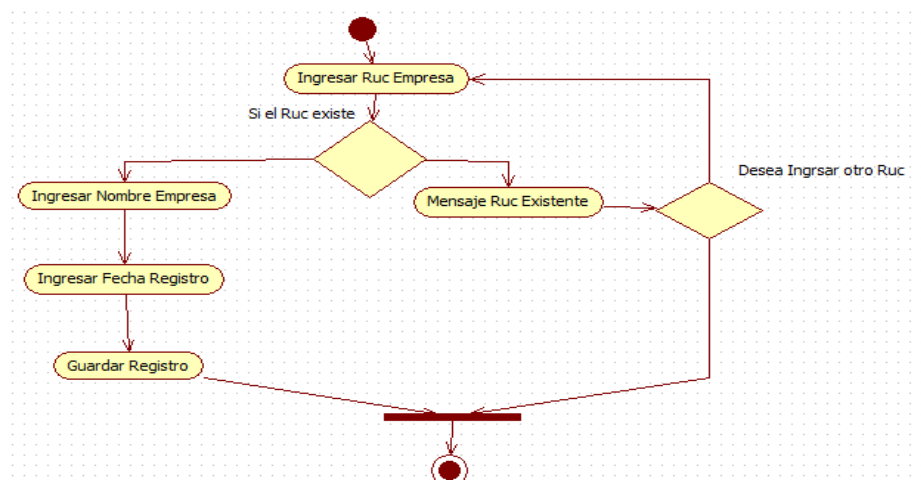


Figura 23. Diagrama de Actividad

2.1.2 Proceso unificado de software “RUP”

RUP está compuesto de seis principios fundamentales:

- **Adaptación del proceso:** Los procesos tienen que adaptarse a las características propias de la empresa, teniendo en cuenta el tamaño y las regulaciones puesto que influirán en su diseño, cabe tener en cuenta que la adaptación del proceso tiene que regirse al alcance del proyecto.
- **Balancear prioridades:** Los requerimientos de los clientes pueden ser diferentes y contradictorios, para esto se debe encontrar un balance en el que se puede satisfacer las necesidades de todos.
- **Colaborar entre equipos:** Para desarrollar un software no se debe contar con una persona, sino con múltiples equipos puesto que cada equipo tiene su función específica en el transcurso de la implementación. Es muy importante establecer una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.
- **Demostrar valor interactivamente:** Los entregables del proyecto se irán entregando de modo interno, en etapas iteradas, en cada interacción se analiza la opinión de los clientes para verificar la estabilidad y calidad que tiene el producto como también los riesgos que se podrían presentar.
- **Elevar el nivel de abstracción:** Este principio se basa en la utilización de patrones de software y esquemas de interfaces. Estos patrones de software ayudan a las prestaciones visuales de la arquitectura como por ejemplo UML.
- **Enfocarse en la calidad:** El control de calidad se lo debe realizar en cada una de las interacciones que posee el proyecto.

2.1.2.1 Características

Las características esenciales que posee el proceso unificado de software son:

- **Dirigido por los Casos de Uso:** El objetivo principal es orientar el proyecto a las necesidades específicas del cliente.
- **Centrado en la arquitectura:** Esta capa se relaciona con la toma de decisiones lo que ayuda a construir un sistema estructurado y ordenado.
- **Iterativo e incremental:** Divide el proyecto en pequeños proyectos con la finalidad de que los casos de uso y la arquitectura cumplen sus objetivos de manera más definida y depurada.

2.1.2.2 Ciclo de vida del proceso unificado

El ciclo de vida de RUP es una vista de las actividades que se repite a lo largo de una serie de ciclos que construyen la vida de un sistema. Cada ciclo es una versión del sistema en el que consta de cuatro fases, como se muestra en la Figura 24:

- Inicio (define el alcance del proyecto y define los casos de uso)
- Elaboración (planifica el proyecto)
- Construcción (define características y elabora una arquitectura base)
- Transición (crear el producto y la transición es transferir el producto a sus usuarios.)



Figura 24. Ciclo de vida de RUP.

Cada fase se subdivide en iteraciones, estas interacciones se desarrollan en secuencia, presentando un conjunto de disciplinas o flujos de trabajos, como se muestra en la Figura 25.

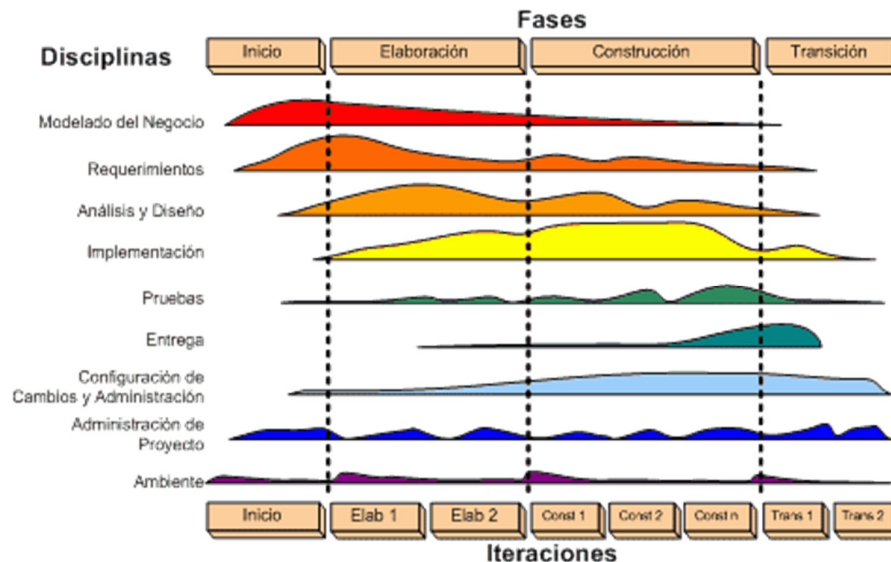


Figura 25. Fases del Ciclo de vida de RUP.

Fuente: (Juan Fuente & Cueva Lovelle, 2006)

Cada disciplina es un conjunto de actividades relacionadas (flujos de trabajo) entre si y vinculadas a un área específica dentro del proyecto.

Estas actividades son una secuencia que indica un orden con el que se debe trabajar, a este proceso se lo denomina Grupo de Trabajo.

RUP define para cada disciplina un flujo de trabajo, estos flujos de trabajo dependen de las interacciones del proceso que se va a desarrollar en la organización. Existen siete actividades que se usan en cada fase del proyecto, estas actividades son: Modelado del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas y Despliegue.

Durante el proyecto RUP especifica para su seguimiento de calidad la: Gestión de proyecto, Configuración y control de cambios y entorno.

2.1.2.3 Fases del ciclo de vida del proceso unificado

RUP se divide en cuatro fases, dentro de las cuales se realizan varias iteraciones midiendo el nivel de madurez que alcanza el producto durante cada actividad ejecutada en los entregables que se han culminado.

A. Fase Inicio

En esta fase se define principalmente las actividades de modelado de negocio y requisitos del sistema, se identifican los actores y los casos de uso más esenciales que se van a diseñar para una mejor estructuración del proyecto. Los objetivos primordiales que se trazan en esta fase son:

- Establecer las reglas de negocio y las limitaciones que posee el sistema.
- Identificar los casos de uso críticos del sistema, los escenarios básicos para establecer la funcionalidad específica del sistema.
- Contemplar una arquitectura para los escenarios principales.
- Estimar los costos, recursos y tiempo para el proyecto que se va a realizar.

En fase de inicio hay tener en cuenta, que las evaluaciones antes expuestas hayan llegado a su final, es decir se identificó y se priorizó los riesgos más importantes que existe en la obtención de requerimientos de modelado de negocio.

▪ Modelo del Negocio

El modelo del negocio define todos los procesos y procedimientos que se implantará para una determinada aplicación. Esta etapa permite identificar el funcionamiento actual de los procesos que se van a sistematizar en la empresa, entender la estructura y la dinámica del negocio con el fin de especificar un enfoque claro y conciso, de que necesita dicha organización.

- **Requisitos**

La fase de análisis de requerimientos es la más importante del desarrollo de software, puesto que aquí se detallan los requisitos y condiciones que debe cumplir el sistema que se desea implementar.

En esta etapa es donde se realiza el levantamiento de requerimientos con la finalidad de satisfacer las necesidades de los clientes, plasmando el alcance y la estimación del proyecto, para el desarrollo del sistema.

Después de haber realizado el levantamiento de cada uno de los requerimientos es necesario especificarlos en la norma IEEE830, para un mejor entendimiento formal de las necesidades del usuario final con respecto al producto final.

B. Fase Elaboración

En la fase de elaboración se capturan a detalle los requerimientos del negocio, abarca los flujos de trabajo de requerimientos, se analiza el dominio del problema, se elabora la arquitectura base, análisis y diseño de una parte de la implementación orientado a la base de la arquitectura, que se han planteado en el proyecto según el alcance que se haya definido en la fase de inicio.

Esta fase culmina cuando alcanza el hito de la arquitectura del sistema y es el principio para la fase de construcción.

- **Análisis y Diseño**

La etapa de análisis y diseño especifica los requerimientos que se van a implementar en el sistema, estos requerimientos fueron documentados en la especificación de requerimientos de Software (ERS) dentro de la IEEE830 con el fin de mantener un establecimiento formal en la (ERS).

Los requerimientos antes mencionados se transformaran a diseños del sistema con la finalidad de alcanzar una arquitectura robusta y confiable para la implementación.

C. Fase Construcción

La fase de construcción, es la fase más larga del proyecto, en la que se completa la implementación del sistema tomando como base, la arquitectura obtenida durante la fase de elaboración.

La finalidad principal de esta fase, es alcanzar la capacidad operacional y funcional del producto de forma incremental, a través de las sucesivas interacciones obteniendo una nueva versión ejecutable del producto.

La fase de construcción concluye con el hito de obtención de una funcionalidad completa, en el que el producto está listo para implantar en un entorno de producción.

▪ Implementación

Se implementan clases y objetos en ficheros fuente, binarios, ejecutables etc., dando como resultado un sistema de calidad, es necesario planificar el orden en que deberá ser implementado cada subsistema, estos subsistemas tienen que ser probados para verificar si cumplen con las funcionalidades requeridas en el sistema.

▪ Pruebas

La fase de pruebas tiene por objeto evaluar y asegurar la calidad del producto. El sistema se encontrará sometido a pruebas con la finalidad de encontrar el más mínimo error en la aplicación y corregirlo a tiempo. En el caso de que los errores no sean corregidos a tiempo, estos errores se convertirán en defectos, por lo que a futuro representaría costos elevados al momento de corregirlos.

También se valida y se prueba la especificación de requerimientos de una forma concreta, en lo que respecta al diseño se evaluará que sea una interfaz amigable para el usuario.

D. Fase Transición

La fase de transición tiene como finalidad presentar el despliegue del producto, en el entorno de los usuarios finales, para entrenamiento del usuario en el manejo del producto, en caso de cambios se contempla versiones actualizadas del sistema y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del sistema. Esta fase concluye con el hito de publicación del producto.

- **Despliegue**

El objetivo principal es realizar pruebas antes de poner el sistema en producción, además es necesario efectuar la migración de datos para poder distribuir el sistema a los usuarios finales.

También determina el proveer asistencia técnica a los usuarios es decir capacitar en el manejo del sistema.

Para el seguimiento de calidad del proyecto, RUP se basa en lo siguiente:

- Gestión del Proyecto
- Gestión del cambio y configuraciones
- Entorno

- **Gestión del Proyecto**

La gestión del proyecto ayuda a verificar el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que satisfaga las necesidades de los clientes.

- **Gestión del cambio y configuraciones**

La gestión de cambios y configuraciones define mantener la integridad del producto. Los cambios que se realicen tienen que ser previamente analizados, para que sean efectuados en el sistema y previamente documentados para conocer el procedimiento efectivo del cambio.

▪ **Entorno**

El enfoque principal de esta actividad es brindar un soporte al proyecto con las herramientas, procesos y métodos adecuados para la implementación del sistema.

2.1.3 Metodología de Diseño Hipermedia Orientado a Objetos

La metodología de diseño hipermedia orientado a objetos “OOHDM” es una metodología orientada a objetos que propone un proceso de desarrollo interactivo, incremental y basado en prototipos. En una primera instancia debido al poco auge que tenía el Internet, OOHDM se aplicaba a sistemas que incluían hipertexto y multimedia como: CD’S promocionales, enciclopedias, museos virtuales, etc. Pero el avance tecnológico que ha obtenido el Internet obligó a la creación de aplicaciones hipermedia en Internet, tales como: comercio electrónico, motores de búsqueda, sitios educativos y de entretenimiento.

La metodología OOHDM permite que el diseño navegacional y el comportamiento funcional de la aplicación se integren transparentemente reduciendo el tiempo y costo de la implementación enfocándose a utilización de patrones de diseño.

OOHDM se enfoca en el desarrollo de aplicaciones hipermedia a través de cuatro fases que se desarrollan iterativamente:

- Diseño conceptual.
- Diseño navegacional.
- Diseño de interfaces abstractas.
- Implementación.

Dentro del alcance del proyecto la metodología OOHDM se aplicará para el Modelo de navegación.

Dentro de la metodología OOHDM existen varios tipos pre-definidos de clases de navegación como:

- **Nodos:** Contenedores básicos de información de aplicaciones hipermedia, poseen una combinación de atributos básicos como imágenes, sonidos y enlaces.
- **Enlaces:** Refleja la relación de navegación que puede explorar el usuario, actúa como un objeto intermedio en el proceso de navegación o también como puente de conexión entre dos nodos.
- **Estructuras de Acceso:** Actúan como índices o diccionarios que permiten al usuario encontrar de forma rápida y eficiente la información deseada.

El diseño de navegación define dos esquemas de modelamiento:

- **Esquema de clase de navegación:** Permite diseñar adecuadamente una aplicación hipermedia, teniendo en cuenta los caminos que el usuario puede seguir, de esta manera se podrá evitar la información redundante, o que el usuario se pierda en la navegación. En OOHDM un contexto navegacional está compuesto por un conjunto de nodos, de enlaces de clases de contexto y de otros contextos navegacionales.
- **Esquema del Contexto de navegación:** Describe la dinámica de la aplicación, mostrando los cambios espaciales de navegación cuando el usuario navega.

Ayuda a complementar la definición de una clase de navegación. Por ejemplo, indica qué información está accesible desde un enlace a otro.

2.1.4 Herramientas para análisis y diseño orientado a objetos

Una herramienta case es la unión de sistemas de software que utilizan técnicas de diseño y metodologías de desarrollo, para controlar y verificar la construcción del modelado de negocio, captura de requisitos, análisis y diseño, implementación, control de cambios y gestión de configuración.

Para el desarrollo y diseño de la arquitectura del presente proyecto, se realizó un análisis de algunas herramientas case disponibles para la gestión y análisis de los requisitos del sistema. Las herramientas case que serán analizadas son: StarUML, Rational Rose y BoUML.

2.1.4.1 StarUML

Es una herramienta de software libre basada en los estándares UML(Unifined Model Lenguaje) y MDA (Model Driven Architecture), ayuda a la generación de código C++,C# Java y PHP , soporta ingeniería inversa. Utiliza un método de "templates", en el cual se puede modificar el código que generará detectando la agregación y composición en los respectivos diagramas. Anteriormente Staruml era un producto comercial y hace un año paso a ser un proyecto comercial con licencia Open Source dando la facilidad de ser una herramienta robusta y libre para el análisis y diseño de software.

Las ventajas son:

- Diseña todos los diagramas basados en UML 2.0 como por ejemplo: Diagramas de casos de uso, diagrama de clase, diagramas de secuencia, diagramas de colaboración, diagramas de estado etc.
- Posee la capacidad de generar código a partir de diagramas y viceversa, actualmente se encuentra funcionando para los lenguajes c++, c#, PHP y java. Licencia Open Source.
- Soporta la tecnología MDA.
- Optimiza el código generado.

- Soporta ciertos patrones de diseño (GoF, EFB patterns).

Las desventajas son:

- Mono plataforma “Windows”.

2.1.4.2 Rational Rose

Es una herramienta establecida por Rational Software Corporation actualmente forma parte de IBM se basa en la creación de diagramas de lenguaje de notación (UML) Lenguaje Unificado de modelado, facilitar la captura de dominio de la semántica, la arquitectura y el diseño. Este software tiene la capacidad de: crear, ver, modificar y manipular, posee soporte para generación de código Java y modelos J2EE.

Las ventajas son:

- Es una herramienta de uso propietario.
- Genera código Java y CORBA, ingeniería inversa y sincronización de modelo código configurable.
- Aplica ingeniería inversa y la sincronización de modelo – código configurable.
- Admite la integración con otras herramientas de desarrollo (IDEs).
- Permite la encapsulación de los objetos.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.
- Admite notaciones como: UML, COM, OMT y Booch
- Integración con modelado de datos.
- Soporta múltiples lenguajes incluyendo C++, Visual C++ (6.0), Visual Basic, Java, CORBA, etc.).

Las desventajas son:

- Mono plataforma “Windows”.
- Ocupa mucha memoria para su ejecución.

- No es lenguaje multiplataforma.

2.1.4.3 BoUML

Es una herramienta de software libre para “Modelado Visual”, basado en los estándares de UML (Lenguaje Unificado de Modelado), forma parte de un conjunto amplio de análisis y diseño del ciclo de vida de un sistema, permite definir y generar código C++, Java, Idl y PHP y Python es compatible con Unix/Linux/Solaris, MacOS X y Windows.

Las ventajas son:

- Soporta UML2 para la generación de diagramas
- Es multiplataforma: Unix/Linux/Solaris, MacOS X y Windows.
- Soporta múltiples lenguajes para la generación de código C++, java, Idl, Php y Python, ingeniería inversa, sincronización de modelo –código configurable.
- Es rápida y no consume mucha memoria.
- Genera documentación en varios formatos (HTML, XML, ...)
- Soporta ciertos patrones de diseño (GoF, EFB patterns).
- Admite notaciones como: UML, COM, OMT y Booch.
- Integración con modelado de datos.
- Soporta tecnología MDA (Model Driven Architecture).

Las desventajas son:

- No es opensource.

2.1.4.4 Análisis de las herramientas case

Tabla 3

Cuadro Comparativo Herramientas Case

	<i>StarUML</i>	<i>Rational Rose</i>	<i>BoUML</i>
Multiplataforma			X
Soporta UML 2	X	X	X
Generación Código PHP	X		X
Ingeniería Inversa	X	X	X
Licenciamiento Libre	X		
Soporta Patrones de Diseño		X	X
Soporta MDA	X	X	X

Como se puede observar la herramienta BoUML cumple con los parámetros necesarios para el análisis y modelado del proyecto, sin embargo para su uso hay que pagar una licencia, al igual que Rational Rose, de tal manera que la herramienta que será elegida es StarUML por su licencia libre y gran desempeño en el modelado de software.

2.2 Aplicaciones web

En la ingeniería de software se denomina aplicación web a aquellas 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. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea, redes sociales, etc. siendo Facebook como un ejemplos bien conocido de aplicación web.

Hoy en día las aplicaciones web tienen una gran influencia en todas esferas de cualquier negocio, cada día que pasa las aplicaciones de escritorio son más ambiguas ya que tienen muchas más desventajas en su uso que las aplicaciones web a medida, aunque aún se siguen usando para cosas específicas.

Entre las ventajas que tienen las aplicaciones web se puede destacar algunas:

- Ahorra tiempo
- No hay problemas de compatibilidad
- No ocupan espacio en tu ordenador
- Actualizaciones inmediatas
- No consumen recursos
- Son multiplataforma
- Son portables, son independientes del ordenador
- Gran disponibilidad
- No tienen riesgo de virus

Claro no todo son ventajas, también existen algunos inconvenientes pero que cada son menos, como que las aplicaciones de escritorio son menos limitadas que las aplicaciones web, pero los navegadores hoy en día están más preparados para soportar cualquier función de las aplicaciones.

Otra de las ventajas que brindan las Aplicaciones Web, es el hecho de que se pueden ejecutar sin importar el Sistema Operativo. Esto es posible ya que lenguajes como HTML o XHTML son soportado por una gran variedad

de navegadores web, lo cual ayuda a no tener que desarrollar una aplicación para cada Sistema Operativo sea este: Windows (en cualquiera de sus versiones), Linux, Mac OS, etc.

Generalmente se entrega al cliente una página web estática, pero la secuencia de páginas les provee una experiencia interactiva a los usuarios, un usuario ingresa datos por medio de un formulario web y el navegador despliega la página dependiendo de la información que el cliente haya ingresado.

En 1995, Netscape introdujo el Script de lado del Cliente (client-side scripting) al cual se lo denominó JavaScript, permitiendo a los programadores añadir elementos dinámicos en la interfaz del usuario. En 1996, Macromedia introdujo Flash, un reproductor de animaciones que puede ser instalado en un navegador a manera de un plugin, permitiendo adjuntar animaciones a las páginas web. Esto permite el uso de Scripts para programar una interacción en el lado del cliente sin la necesidad de conectarse al servidor. En el 2004, Web 2.0 fue introducida con la finalidad de usar Internet como medio de acceso a los servidores. En el 2005 se introdujo AJAX, con lo que se puede hacer que el lado del cliente sea mucho más interactivo.

La Interface web es muy funcional ya que no posee muchas limitantes en el lado del cliente. Mediante la utilización de varios lenguajes y tecnologías actualmente existentes en el ámbito web, como por ejemplo Ajax, HTML 5, etc.

Muchas Aplicaciones Web han utilizado estas ventajas y han creado ambientes familiares los cuales adoptan apariencias conocidas por los usuarios, sistemas de ventanas, menús, etc.

En el desarrollo de Aplicaciones Web se utilizan varios scripts del lado del cliente para añadir funcionalidades, especialmente para crear una experiencia interactiva la cual no requiera que la página tenga de refrescarse

o cargarse nuevamente. En la actualidad, los Scripts del lado del cliente se han desarrollado para coordinarse con tecnologías del lado del servidor como PHP y AJAX, una técnica utilizando una combinación de varias tecnologías, es un ejemplo de una tecnología la cual crea una experiencia más interactiva.

2.3 Arquitectura de software de las aplicaciones web

2.3.1 Arquitectura tres capas

La arquitectura tres capas nace en la década de los 90's para solventar las limitaciones de la arquitectura cliente servidor. La Arquitectura tres capas introduce una capa intermedia entre la interface del cliente y el manejo de la información. Esta capa intermedia es la encargada de ejecutar el modelo y las reglas del negocio. Las ventajas de la Arquitectura en tres capas frente a la Arquitectura Cliente Servidor son:

- Incremento en el desempeño
- Flexibilidad
- Facilidad para realizar mantenimientos
- Reusabilidad de Componentes
- Escalabilidad

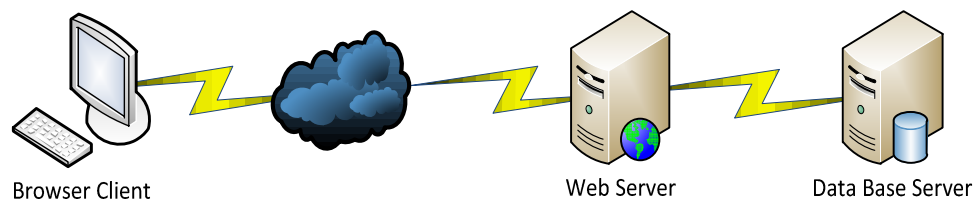


Figura 26. Gráfico Arquitectura Tres Capas

Como se muestra en la Figura 26 la Arquitectura Tres Capas consiste en dividir a la aplicación en tres partes importantes las cuales tienen una

función específica dentro de la aplicación. La capa uno o también llamada la capa del cliente es la que se encarga de la interfaz del usuario. La capa dos o es la que se encarga de las reglas del cliente, en esta capa se encuentran las reglas del negocio y donde se define el funcionamiento de nuestra aplicación. Por último se tiene la capa tres o capa de datos que es la encargada del manejo y almacenamiento de los datos, esta capa regularmente está compuesta de una base de datos.

El funcionamiento de una Aplicación Web se lo realiza de la siguiente forma:

- El cliente realiza una solicitud.
- La solicitud es procesada por el servidor web.
- El servidor verifica si necesita consultar datos desde el repositorio de información.
- El servidor realiza la consulta con el repositorio de información y envía la respuesta hacia el navegador.
- El navegador despliega la información que le envía el servidor web.

Este proceso se lo puede visualizar gráficamente como se lo muestra en la Figura 27.

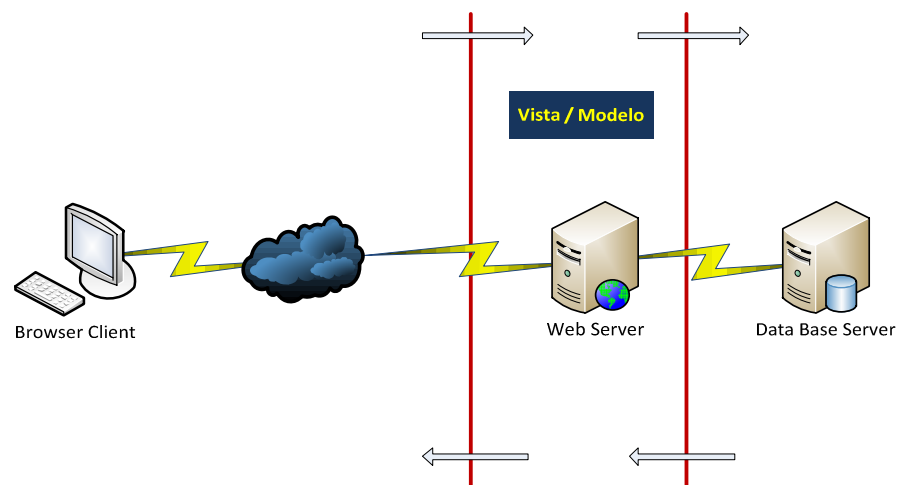


Figura 27. Comunicación en Modelo 3 Capas

Las Arquitectura en tres capas provee una funcionalidad de manejo de bases de datos. El manejo de los datos de forma separada garantiza la consistencia de los mismos a través de toda la aplicación.

La capa intermedia provee servicios de manejo de procesos los cuales pueden ser utilizados por varias aplicaciones. De la misma manera mejora el desempeño por medio de la centralización de los procesos lógicos. La centralización de los procesos lógicos hace que la administración y modificación de los mismos sea más sencilla, dado que solo se deben cambiar las reglas del negocio en la capa intermedia y no en los clientes.

2.4 Tecnologías Web

2.4.1 Java Server Page “JSP”

Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología fue desarrollada por Sun Microsystems.

JSP permite la utilización de código Java por medio del uso de Scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas externas las cuales pueden ser personalizadas.

JSP puede considerarse como una manera alternativa, y simplificada, de construir Servlets. Es por esto que una página puede hacer todo lo que un servlet puede hacer, y viceversa. Cada versión de la especificación de JSP está vinculada a una versión en particular de la especificación de servlets.

El funcionamiento general de la tecnología JSP se basa en que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento que se presentará en la pantalla del Navegador.

El rendimiento de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java.

A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera eficiente. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios. Es común incluso que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

Java Server Pages son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP o ASP programas que generan páginas web en el servidor. Sin embargo, se diferencian de ellos en otras cosas.

Los JSP se ejecutan en una máquina virtual Java, lo cual permite que se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación, un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

2.4.2 Active Server Pages “ASP”

ASP (Active Server Pages) es una tecnología del lado del servidor para la creación de páginas dinámicas. Inicialmente fue promocionado como un adicional del Internet Information Service (IIS), pero ha sido incluido como un componente gratis en la familia de Windows Server.

El desarrollo de sitios web bajo la tecnología ASP es posible por la utilización de varios objetos fundamentales previamente desarrollados. Cada objeto corresponde a un grupo de funciones frecuentemente las cuales son muy útiles en la creación de páginas web dinámicas. En la versión ASP 2.0 existen 6 objetos fundamentales los cuales son: Application, ASPError, Request, Response, Server y Session.

La mayoría de páginas web desarrolladas bajo ASP son escritas en lenguaje VBscript, sin embargo el motor ASP es capaz de utilizar otros lenguajes mediante la utilización de directivas de lenguaje, esto permite la utilización JScript, PearlScript entre otros.

ASP ha tenido tres versiones:

- ASP versión 1.0 (Distribuida con IIS 3.0) publicada en Diciembre de 1996
- ASP versión 2.0 (Distribuida con IIS 4.0) publica en Septiembre de 1997
- ASP versión 3.0 (Distribuida con IIS 5.0) publicada en Noviembre del 2000

Los avances entre la versión 3.0 y la versión 2.0 de ASP fueron mínimos, uno de sus avances más significativos fueron los métodos de ejecución del servidor y también el objeto ASPError.

La versión actual de ASP es la versión 3.0 que está disponible con IIS 6.0 bajo es Sistema Operativo Windows Server 2003 o en IIS 7.0 bajo el sistema operativo Windows Server 2008. ASP.Net es frecuentemente

confundido con la última versión de ASP, pero las tecnologías son muy diferentes, ASP .NET se basa en el Framework de y en sus lenguajes de compilación, ASP aparece estrictamente como un lenguaje de scripts.

2.4.3 PHP: Hypertext Preprocessor “PHP”

PHP es un lenguaje de servidor originalmente diseñado para el manejo de páginas web dinámicas, que mantengan una interacción con el usuario, de libre distribución y con una gama amplia de documentación en la web es una herramienta ideal para el desarrollo de aplicaciones.

PHP actúa previamente como un filtro, tomando datos de entrada desde un archivo o una secuencia de comandos PHP, y entregando una secuencia de datos que generalmente se presentan en formato HTML.

Como PHP fue creado originalmente para la creación de páginas web dinámicas, este se centra en los scripts del lado del servidor, y es similar a otras tecnologías las cuales proveen contenido dinámico desde un servidor hasta el cliente como: Microsoft ASP .NET, JavaServerPages, etc.

2.4.4 Análisis de Tecnología Web

Para visualizar las características y ventajas de las tecnologías del lado del servidor que son utilizadas para el desarrollo de páginas web dinámicas se ha realizado el siguiente cuadro.

Tabla 4

Cuadro Comparativo Tecnologías

	<i>PHP</i>	<i>ASP</i>	<i>JSP</i>
Tecnología Lado del Servidor	X	X	X
Contenido Dinámico	X	X	X
Multiplataforma	X		X

Continúa →

Portabilidad	X	X
Soporte Online	X	X
Opensource	X	X

Para el desarrollo de la aplicación web se ha escogido la tecnología PHP debido a que es una tecnología la cual brinda todas las facilidades para el desarrollo de la misma. Esta tecnología cumple con características como desarrollo de páginas web dinámicas, multiplataforma, portabilidad y personalización. Estas son las razones por las cuales se ha escogido la tecnología PHP para el desarrollo de la aplicación web.

2.4.5 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se

proporciona al lenguaje JavaScript una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.4.6 Ajax

Asynchronous JavaScript and XML (AJAX) es un método de desarrollo basado en JavaScript para el desarrollo de Aplicaciones Web. Este código se ejecuta en el lado del cliente, en el navegador del usuario mientras se mantiene una comunicación asíncrona con el servidor. De esta manera es factible realizar interacciones sobre las páginas web sin necesidad de que estas se recarguen, lo que aumenta la velocidad e interacción en los sistemas web.

AJAX es una tecnología asíncrona en el sentido de que los datos adicionales que se requieren del servidor se cargan en un segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es un lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de AJAX mientras que el acceso a los datos se realiza por medio de XMLHttpRequest, que es un objeto disponible en los navegadores actuales.

AJAX es la combinación de cuatro tecnologías existentes:

- HTML o XHTML y hojas de estilo en cascado (CSS) para el diseño de la interfaz del usuario.
- Document Object Model (DOM) al cual se puede acceder mediante JavaScript y JScript para mostrar e interactuar con la información presentada.

- El objeto XMLHttpRequest para interactuar con los datos asíncronamente con el servidor web.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor.

En la actualidad casi todos los navegadores soportan la tecnología AJAX, existen muy pocos navegadores que no lo soportan y en su mayoría versiones antiguas de navegadores que si lo soportan.

2.4.7 Entorno Integrado de desarrollo

Un IDE (Integrated Development Environment) es una aplicación compuesta de un conjunto de herramientas útiles para el programador. Un entorno IDE puede ser exclusivo para un lenguaje de programación o puede ser para varios lenguajes, un IDE regularmente está compuesto por un editor de código, un compilador, un depurador y un constructor para la interfaz gráfica.

A continuación se analiza el IDE seleccionado para el proyecto.

2.4.7.1 Netbeans

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones

basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El IDE NetBeans es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Sun Studio, Sun Java Studio Enterprise, y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE.

2.4.7.2 PHP Edit

Es un IDE de desarrollo basado en PHP el cual fue creado en 1999. Actualmente es utilizado por cientos de personas alrededor del mundo. Entre las principales características de PHP Edit se encuentran:

Mejor entendimiento de código: PHPEdit provee herramientas que ayudan al entendimiento rápido del código como: Resaltado de Código mediante cambio de color para ayudar a identificar palabras reservadas tanto de PHP como de HTML, CSS, JavaScript, XML etc. Browser de Código mediante el cual se puede tener conocimiento de todos los archivos abiertos sobre los cuales se esté trabajando.

Salto a declaraciones: esta utilidad permite navegar rápidamente a través del código mediante saltos desde un llamado de una función hasta su declaración.

Rápido Desarrollo: Posee herramientas que pueden ayudar a escribir código más rápido como autocompletar, esta herramienta permite escribir menos código por parte del desarrollador ya que completa código que siempre se complementa como las comillas o los paréntesis, además muestra las posibles opciones de escritura de palabras reservadas luego de escribir la primera letra.

Ayudas de Código: Muestra ayudas sobre las funciones y sus parámetros con una descripción tanto en funciones nativas de PHP como en funciones creadas por el desarrollador.

Publicación de Proyectos: PHP Edit ayuda a la fácil publicación de proyectos mediante la utilización de herramientas como Manejador de proyectos, Manejador de Bases de Datos, Manejador de Archivos mediante la utilización de FTP.

Es una herramienta muy útil a la hora de desarrollar proyectos en PHP ya que ayuda a mantener el código organizado y la fácil navegación en el código.

2.4.7.3 DreamWeaver

Es un editor de aplicaciones web en el cual se puede desarrollar aplicaciones en distintos lenguajes tales como: PHP, HTML, CSS, JavaScript etc. Dreamweaver es un editor muy poderoso el cual brinda un ambiente amigable de desarrollo tanto en la parte gráfica como en la parte de código.

En la parte gráfica brinda ayuda con barra de herramientas, paleta de colores, barra de diseño y barra de objetos mediante las cuales se facilita y agiliza el diseño de una aplicación web en su parte gráfica.

En la parte de desarrollo de código brinda un apoyo con herramientas que permiten:

Navegar rápidamente a través del código mediante buscador de palabras claves, resaltado de palabras reservadas con diferentes colores que facilitan la identificación del código que se está escribiendo.

Rápida escritura de código a través de la utilización de la herramienta de autocompletar la cual brinda ayudas para completar las palabras, las opciones de las funciones y los posibles valores de un atributo de una etiqueta.

Organización y búsqueda rápida de los archivos mediante la utilización de un navegador de archivos y sitios web.

2.4.7.4 Análisis de herramientas IDE para PHP

De los IDE presentados anteriormente, se ha optado trabajar con NetBeans 7.x debido a sus características que lo vuelven una herramienta muy completa y confiable, con mucho soporte a en internet brindado por su comunidad, además de su licenciamiento GPL ayuda a no incurrir en gastos para el desarrollo del proyecto.

2.5 Sistemas de administración de base de datos

Los Sistemas de Administración de Bases de Datos (DBMS) son un conjunto de programas las cuales permiten crear, acceder y administrar bases de datos. Estos sistemas de administración se componen básicamente de: un lenguaje de definición de datos conocidos como Data Definition Language (DDL), por un lenguaje de manipulación de datos conocido como Data Manipulation Language (DML) y un lenguaje de consulta de datos conocido como Structure Query Language (SQL).

2.5.1 MySql

Es un servidor de bases de datos de código abierto más popular en el mundo. Su arquitectura lo hace extremadamente rápida y fácil de adaptar a un sistema.

El servidor MySQL fue desarrollado originalmente para el manejo de bases de datos de una manera más rápida que las soluciones existentes en el mercado y ha estado siendo usado de manera exitosa en ambientes de producción exigentes por varios años. A pesar de que se encuentra en un proceso de constante desarrollo, el servidor MySQL un conjunto muy extenso y variado de funciones muy útiles. Su conectividad, velocidad y seguridad hacen de MySQL un servidor apropiado para ser utilizado como motor de base de datos para una Aplicación Web.

El uso extenso que tiene MySQL en el desarrollo de Aplicaciones Web hace que exista una gran cantidad de manuales, tutoriales, foros, artículos, libros sobre la misma lo cual facilita el desarrollo.

Sus principales características son:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos, permite la creación de tipos de datos definidos por el usuario.
- Dispone de API's para gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla, lo cual agiliza sus respuestas en la búsqueda de datos.
- Gestión de usuarios y contraseñas, manteniendo un nivel alto de seguridad en los datos.

2.5.2 Postgre SQL

Es un servidor de base de datos relacional libre, por lo que no se necesita comprar una licencia para su utilización. Es una alternativa a otros sistemas de bases de datos de código abierto como: MySQL, Firebird o MaxDB, así como sistemas propietarios como Oracle, SQL Server o DB2.

Postgre SQL es considerado como una de las bases de datos de código abierto más completas. Posee algunas de las características de las bases de datos propietarias las cuales se detallan en la siguiente lista:

- Data Base Management System (DBMS): Posee la capacidad de aproximar los datos a un modelo objeto-relacional, y es capaz de manejar complejas reglas del negocio.
- Altamente Extensible: Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

- Integridad Referencial: Utilizada para la verificación y validación de los datos.
- Control de Concurrencia Multi Versión (MVCC): Empleada para evitar bloqueos innecesarios de la Base de Datos.

Cabe recalcar que debido a que Postgre SQL es una herramienta gratuita, existe una gran cantidad de información en Internet.

2.5.3 Oracle

Oracle es una herramienta creada por Oracle Corporation para la administración y gestión de Bases de Datos. Esta herramienta tiene costo, es decir que para su utilización se debe comprar las licencias respectivas. Debido a su elevado costo no es una herramienta muy popular aunque muchos desarrolladores destacan su seguridad y gran versatilidad en el manejo y administración de datos.

Debido a su elevado costo esta herramienta puede ser adquirida por grandes empresas, las cuales pueden explotar las características avanzadas de la base de datos Oracle, obteniendo una herramienta completa para la administración de la información.

En la actualidad Oracle es considerada la base de datos más robusta del mercado, mostrando gran estabilidad y velocidad con grandes volúmenes de información; cabe recalcar que cuenta con un amplio soporte mediante manuales, tutoriales, libros y ayuda en línea con la compra de la licencia del producto.

2.5.4 Análisis Motor Base de Datos

Tabla 5

Cuadro comparativo de motores de Bases de Datos

	<i>Postgre Sql</i>	<i>MySql</i>	<i>Oracle</i>
Integridad de Datos	X	X	X
Stored Procedures	X	X	X
Vistas	X	X	X
Triggers	X	X	X
Secuencias	X		X
Cursores	X	X	X
Tipos de Datos	X	X	X
Índices Single Colum	X	X	X
Índices Multi Colum	X	X	X
Primary y Foreign Key	X	X	X
Replicación		X	X
Distribución Libre	X	X	

De acuerdo a la tabla 5, se puede observar una alta similitud entre los 3 motores de bases de datos, sin embargo uno de los factores importantes para la selección del motor de base de datos adecuado para el proyecto, es la incursión en gastos por licencias, si bien Oracle es una buena opción su alto costo de licenciamiento, infraestructura y soporte necesario es un factor determinante al momento de realizar este análisis.

Postgre SQL es una base de datos muy sólida y robusta, posee características muy similares a MySQL como se puede apreciar en la Tabla 5, las diferencias principales son las secuencias y la réplica que en Postgre SQL tiene un costo ya en una licencia comercial, su instalación y mantenimiento requiere de un cierto nivel de experticia por lo cual se puede incurrir en costos de soporte especializado.

MySQL una de las bases de datos más populares a nivel mundial ofrece muchas facilidades para su uso, la búsqueda de documentación sobre esta base de datos es amplia debido a su amplio uso, una de sus características principales es ser de distribución libre, por lo cual no se incurre en gastos altos en licenciamiento, infraestructura y soporte, diseño de bases de datos relaciones, capacidad de establecer replica y soporte de transaccionalidad alta, es por esto que ha sido seleccionada para el desarrollo del proyecto. Además de ser la base de datos más utilizada para desarrollo web y tener una alta integración con el lenguaje de servidor PHP lo que facilitara su uso.

2.5.5 Gestión Base de Datos

Como la base de datos elegida es MySQL se va a definir la o las herramientas de gestión de base de datos que se puede utilizar para el desarrollo de nuestros queries SQL para el sistema de Gestión de Solicitudes OnLine.

Entre las herramientas de gestión de base de datos para MySQL se puede nombrar a PHPMyAdmin, MySQL Workbench y Navicat.

2.5.5.1 PHPMyAdmin

Esta es una de las herramientas de gestión de base de datos más populares y utilizadas a nivel mundial, desarrollado en PHP y de uso libre con una interfaz web muy intuitiva, capacidad para gestionar todos los aspectos y características de MySQL, la capacidad de crear consultas “Queries” complejas mediante el uso de QBE “Query By Example” se

convierte en una de las principales herramientas a utilizar en el desarrollo del sistema.

2.5.5.2 Navicat

Navicat posee una gran variedad de herramientas dedicadas a la gestión de bases de datos, entre ellas se encuentran Navicat Premium, Navicat MySQL y Navicat Lite, de estos 3 productos orientados a la base de datos MySQL, los dos primeros son versiones de pago con licencias un poco altas en costos y la versión Lite que es de uso libre, aunque no posee todas las características de las versiones de pagos es una buena opción para realizar consultas “Querys” complejas, gestionar tablas, vistas, trigger, funciones y usuarios, disponible en versiones multiplataforma Windows, MacOS y Linux.

2.5.5.3 Mysql Workbench

Si bien tiene un cliente para gestión de base de datos, su uso está más enfocado al diseño de la base de datos, de una manera visual, se puede diseñar nuestra base y en un siguiente paso poderla aplicar a nuestro motor MySQL directamente, cuenta con un editor de consultas para gestionar la base de datos y específicamente diseñar consultas “Querys” aunque con un limitado set de herramientas a diferencia de los anteriormente mencionados.

2.5.5.4 Análisis Herramienta de gestión de Base de Datos.

Tabla 6

Cuadro comparativo herramientas de gestión de BD

	<i>PHPMysqlAdmin</i>	<i>Navicat</i>	<i>MySQL Workbench</i>
Multi Plataforma	X	X	
Multi DB Engine		X	
Multi Conexión	X	X	X

Continúa →

Editor SQL	X	X	X
Query Builder		X	X
Multi Idioma	X	X	
WEB	X		
Modelador DB	X		X
Importación	X	X	
Exportar	X	X	
Backups	X	X	
DB Compare		X	

De acuerdo a la Tabla 6 la combinación de las 3 herramientas de acuerdo al escenario es muy útil, PHPMyAdmin para soporte remoto del sistema ya implementado por la facilidad de acceso y calidad de herramientas disponibles en su arquitectura, para el diseño y modelado de la base de datos la mejor herramienta seria MySQL WorkBench, y para el desarrollo del sistema, la generación de consultas “Querys” complejas la mejor opción es Navicat Lite. Estas 3 herramientas al ser de uso libre podrán ser utilizadas para el desarrollo e implantación del sistema de Gestión de Solicitudes Online.

2.6 Servidor Web

La utilización del protocolo HTTP “Hypertext Transfer Protocol” y HTTPS “Hypertext Transfer Protocol Secure” en los servidores web permite tener páginas web de texto enriquecido, imágenes, audio, video, formularios, etc.

En la actualidad con el uso de HTML 5 se puede tener una alta interactividad con el usuario e infinidad de servicios multimedia para

aumentar la experiencia final, claro está que para lograr mantener esta alta interactividad se depende mucho del cliente de nuestro servidor web, los navegadores “Browsers” como son conocidos comúnmente.

El servidor mantiene abierto un puerto abierto a la espera de una petición HTTP “HTTP Request” enviado por un cliente “Navegador”, a lo que el servidor responderá enviado código HTML el mismo que será interpretado por el cliente “Navegador” y desplegado como una Página Web.

El servidor web puede alojar dos tipos de páginas o aplicaciones web, estas se las puede dividir en dos grupos importantes que son, las que se ejecutan del lado del cliente y las que se ejecutan del lado del servidor.

Aplicaciones / Páginas del lado del cliente: Son las que están compuestas netamente de lenguaje que interpreta el lado del cliente, como es HTML y JavaScript, estos lenguajes son interpretados por los navegadores “Browsers” para que ser presentado directo en la pantalla.

Aplicaciones / Páginas del lado del servidor: Son las que están compuestas de lenguaje que solo interpreta el servidor, en nuestro caso es PHP, en él se puede realizar operaciones directamente de interacción del cliente contra la base de datos o el servidor propiamente, para construir lenguaje HTML que será devuelto al cliente.

En la actualidad se dispone de aplicaciones que pueden brindarnos un servidor integrado como ejemplo se analiza tres de los cuales se elegirá uno para la utilización en nuestro proyecto.

2.6.1 WAMPServer

Wamp Server o Wamp como es más conocido se trata de un conjunto de Apache + Mysql + PHP para Windows perfectamente integradas y listo para su funcionamiento, además trae herramientas como PHPMyAdmin para la gestión de base de datos y webgrind como un profiler para analizar el rendimiento de nuestras aplicaciones en PHP.

Tiene una interface sencilla y es de fácil administración, actualmente dispone de versiones para 32Bits y 64 Bits, es de libre distribución bajo la licencia GPML y desarrollado por la empresa Alter Way de Francia.

2.6.2 Portable WebServer

Al igual forma que WampServer es un servidor integrado que facilita su instalación y puesta en funcionamiento para probar los desarrollos antes de que sean subidos al internet.

Está conformado por Apache + Mysql + PHP, es un servidor muy ligero y con una característica que le da una pequeña ventaja ante sus otros similares, es portable y se lo puede llevar incluso en una Flash Memory, desarrollado exclusivamente para Windows, cuenta con una interface simple que con un solo clic se puede activar p desactivar módulos y servicios de nuestro servidor.

2.6.3 XAMPP

Uno de los servidores integrados más conocidos en el ámbito de los desarrolladores web, producido por la empresa Apache Friends bajo licencia GNU permite su libre uso.

Cuenta con Apache + Mysql + PHP + Perl cuyas iniciales forman parte del nombres del servidor, en donde su primera letra X indica que está disponible para varias plataformas Windows, Linux, MacOS y Solaris.

Cuenta con un sistema de gestión de base de datos, soporte para gestionar cuentas FTP, bases de datos SQLite y varias características que lo vuelve un servidor muy poderoso.

También incluye un servidor de correo “Mercury” y un servidor Tomcat para servlets y JSP. Todo esto lo vuelve un servidor muy robusto y completo para desarrollar una aplicación Web.

2.6.4 Análisis del Servidor Web

Tabla 7

Cuadro Servidor Web

	<i>WampServer</i>	<i>Portable WebServer</i>	<i>XAMPP</i>
Multi Plataforma			X
Apache	X	X	X
MySQL	X	X	X
PHP	X	X	X
Fácil Instalación	X	X	X
Fácil Administración	X	X	X
Documentación Amplia	X		X
PHPMyAdmin	X		X
Ftp			X
Correo			X

Las Frameworks de servidor tienen características similares y cumple con los requerimientos básicos para el desarrollo de nuestro proyecto que es Apache, Mysql y PHP, pero un factor determinante de uno de los tres sobresale ante los demás, lo tiene el servidor XAMPP y es la posibilidad de probar el sistema en los tres principales sistemas operativos que son Windows, Linux y MacOS, por este motivo ha sido el elegido para alojar el sistema Web.

2.7 Protocolos de comunicaciones

Los protocolos definen las políticas y estándares mediante las cuales podrán establecer una comunicación dos o más dispositivos en una red, permitiéndoles transmitir información.

Existen una gran variedad de protocolos, cada uno tiene su propósito en una infraestructura de comunicaciones, cada protocolo es característico de un servicio de red como por ejemplo correo, ftp, video, voz, etc. En el proyecto para el desarrollo web se ocupa dos protocolos principalmente.

2.7.1 Transfer Control Protocol / Internet Protocol (TCP/IP)

Fue desarrollado por el Departamento de Defensa de los Estados Unidos de América en los años 70 consiste en una variedad de protocolos para el desarrollo de las redes de datos. Los protocolos más conocidos de este conjunto son los protocolos TCP y el protocolo IP.

El protocolo TCP/IP está diseñado para enrutar y posee un grado elevado de confiabilidad, es adecuado para redes medianas y grandes. Se utiliza a nivel mundial para conectarse a Internet y a los Servidores Web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

El protocolo IP es el que se encarga de determinar hacia donde son encaminados los paquetes, en función de su dirección de destino. El protocolo TCP se encarga de determinar si los paquetes lleguen correctamente a su destino, en el caso de que un paquete no llegue a su destino intentará reenviarlo hasta que llegue correctamente a su destino. Su arquitectura se representa en cuatro capas, y cada capa maneja protocolos diferentes, algunas de las ventajas que ofrece el protocolo TCP/IP son:

- Independencia de Fabricante
- Soporta múltiples tecnologías
- Funciona en cualquier tipo de dispositivo

- Es un estándar en EEUU desde 1983

La arquitectura TCP/IP tiene una serie de objetivos entre los cuales se encuentran:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador
- Conectividad Universal por medio de la red
- Reconocimientos de extremo a extremo
- Protocolos mundiales estandarizados

2.7.2 Hypertext Transfer Protocol

El protocolo HTTP funciona con la World Wide Web, que es la parte de crecimiento más rápido y más utilizada en Internet. Una de las principales razones de este crecimiento sorprendente de la Web es la facilidad con la que permite acceder a la información.

Este protocolo permite usar una serie de métodos para indicar la finalidad de la petición. Se basa en otros conceptos y estándares como Uniform Resource Identifier “URI”, Uniform Resource Location “URL” y Uniform Resource Name “URN”, para indicar el recurso al que hace referencia la petición del usuario. Los formatos viajan con un formato similar al usado por Internet Mail y el Multipurpose Internet mail Extension “MIME”.

HTTP se basa en sencillas operaciones de solicitud y respuesta, es un protocolo sin estado lo que no permite guardar ninguna información de conexiones anteriores, por lo cual los desarrolladores de aplicaciones web utilizan los Cookies, que es información del servidor guardada del lado del cliente, esto hace que la aplicación web pueda mantener sesiones sin necesidad de estados del protocolo.

2.8 Plataforma / Sistema Operativo

La definición de Sistema Operativo, es la un programa o conjuntos de programas de un sistema informático que ayuda a gestionar recursos de hardware el ordenador y provee de servicios a los programas de aplicaciones que en el son ejecutados.

Existen varios tipos de sistemas operativos en los cuales se puede realizar la implantación de nuestro servidor web así como de nuestro aplicativo de los cuales se puede nombrar a Windows, Linux y MacOS.

2.8.1 Windows

Microsoft Corporation es el propietario creador de la amplia gama de sistemas operativos Windows en versiones de escritorio y servidor. Todos los Sistemas Operativos tienen Licenciamiento para su utilización y tienen como principal característica una excelente interfaz gráfica de usuario aunque con un poco de inestabilidad en su funcionamiento.

Actualmente las versiones que cuentan con soporte por parte de la empresa Microsoft Corporation son Windows 7 en versiones de 32 y 64 Bits, y el más actual Windows 8 en versiones de 32 y 64 Bits con una interface totalmente renovada.

En 1985 Microsoft Corporation publico la primera versión de Windows que fue la Microsoft Windows 1.0 con una interfaz gráfica de usuario para su propio sistema operativo MS-DOS. Que había sido incluido en el IBM PC y ordenadores compatibles desde 1981. La interfaz gráfica fue creada imitando sistema operativo MAC OS de la empresa Apple.

En la actualidad existe una gran variedad de sistemas operativos dentro de la familia Windows tanto para computadores personales como para servidores los cuales se encuentran en constante evolución, lo que permite tener actualizaciones constantes del sistema operativo. El mayor inconveniente de la familia Microsoft son sus altos costos debido a lo que la

mayoría de servidores han empezado a utilizar sistemas operativos gratuitos como UNIX y LINUX.

2.8.2 Linux

La popularidad de los sistemas operativos Linux ha ido en un aumento constante, su libre distribución lo ha hecho el favorito de muchas empresas quienes no desean invertir en licencias costosas para armar sus infraestructuras de red.

En un inicio fue desarrollado para procesadores i386 y con un ambiente grafico muy limitado, en la actualidad Linux cuenta con una amplia compatibilidad con los procesadores actuales y cuenta con un ambiente grafico muy elaborado.

Linux desarrollado en base a Unix es considerado en la actualidad como uno de los sistemas operativos para servidores más estable con características principales como multitarea, multiusuario, gestión de memoria eficiente y soporte de redes multiprotocolo.

2.8.3 MacOS

Es el sistema operativo icono de la empresa Apple, considerado uno de los más seguros y estables, ha mantenido su innovadora línea desde su lanzamiento en 1976 dirigido siempre al usuario final, es uno de los primeros sistemas operativos en contar con ventanas, menús e iconos.

Su característica principal es que solo se lo puede utilizar en computadores de la marca de Apple y hasta 1999 solo se podía utilizar con componentes fabricados por la misma empresa, algo que cambio en las siguientes versiones después de esta fecha ya que Apple reestructuro completamente su sistema operativo basándolo en el sistema operativo Unix lo que le dio más compatibilidad con procesadores de otras marcas, pero aun solo puede ser usado en computadores Apple.

Su licenciamiento no es muy costoso pero como contraparte su hardware es uno de los más costosos del mercado.

2.8.4 Análisis de Plataforma / Sistema Operativo

Hemos visto tres sistemas operativos que se puede utilizar como servidor de aplicaciones web, las diferencias son mínimas y la compatibilidad entre los tres es alta, sus diferencias están más basadas en rendimiento y costos, a continuación la tabla 8 muestra la comparativa S.O.

Tabla 8

Cuadro Sistemas Operativos

	<i>Windows</i>	<i>Linux</i>	<i>MacOS</i>
Multi Plataforma	X	X	X
Multi Tarea	X	X	X
Multi Usuario	X	X	X
Ambiente Grafico	X	X	X
Libre Distribución		X	
Más Popular	X	X	

Como se puede observar en la Tabla 8 la diferencia entre los sistemas operativos es mínima, pero de los tres sobresale uno que cumple con todas las características solicitadas, este es LINUX, en la actualidad es uno de los favoritos para implementar servicio servidor Web, además de ser gratuito y tener una alta estabilidad son los motivos por lo que ha sido escogido para alojar la aplicaciones web a implementar.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1 Desarrollo de la Aplicación

Para el desarrollo de nuestro proyecto se ha tomado a consideración el uso de metodologías OOHDH (Metodología de hipermedia orientada a objetos) para un desarrollo disciplinado y estructural. Además del uso de metodologías de desarrollo ágil para mantener una iteración constante con el cliente y realizarlo en distintas fases de entrega.

Las metodologías orientadas a objetos que se ajustan a las especificaciones y requerimientos del proyecto son las siguientes: para el modelado conceptual se utilizará RUP (Proceso unificado de Software) más UML (Lenguaje unificado de modelado) que en conjunto constituyen una metodología estándar utilizada para el análisis, implementación y documentación de sistemas, para el modelo navegacional OOHDH (Metodología de diseño hipermedia orientado a objetos), y para la especificación de los requisitos del software la norma IEEE830.

3.1.1 Modelado del Negocio

WICOM DEL ECUADOR es una empresa orientada a soluciones y consultoría informática, dentro de sus líneas de negocio está el soporte a nivel de un departamento de tecnología y sistemas para empresas que no cuentan con su propio departamento de TI, brindando soluciones rápidas y eficientes manteniendo sus niveles de atención al cliente en cada requerimiento brindado.

En la Figura 29 se puede observar el macro proceso de ingreso y finalización de solicitudes y tres subprocesos de alta importancia que componen todo este macro procesos.

1. Ingreso de Solicitudes
2. Análisis y Gestión de Solicitudes

3. Finalización de Solicitudes

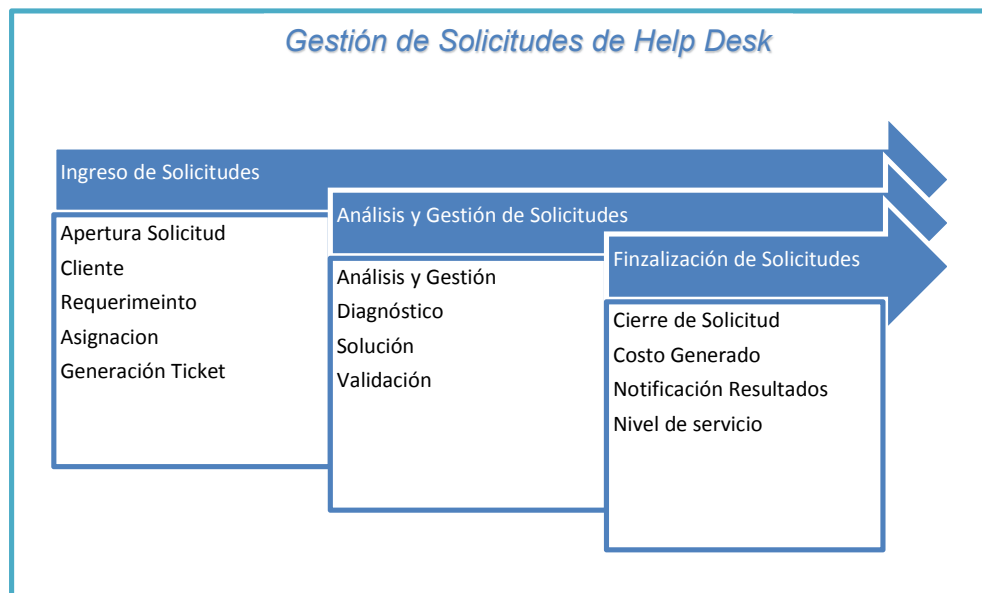


Figura 28. Modelado Contextual del Negocio

Como se puede observar dentro de los tres subprocesos se tiene varias actividades importantes para el modelado del negocio y el posterior desarrollo de la aplicación web.

A continuación se detallará más los subprocesos de Gestión de Solicitudes Online Helpdesk.

1. Ingreso de solicitudes

- Ingreso datos personales del cliente
- Ingreso de requerimiento
- Asignación de técnico responsable
- Ticket de ingreso de solicitud

2. Análisis y Gestión de Solicitudes

- Análisis y gestión de Requerimiento
- Ingreso de Diagnostico
- Ingreso de Solución Requerimiento
- Validación de la Solución

3. Finalización de Solicitudes

- a. Cierre de Solicitud
- b. Ingreso de Costo generado
- c. Notificación de resultados vía email
- d. Calificación Requerimiento, nivel de servicio

Los procesos y actividades detallados anteriormente son los que se implementaran en el desarrollo del proyecto con la finalidad de sistematizar la gestión de Solicitudes de HelpDesk en la empresa WICOM DEL ECUADOR.

A continuación se explica más a detalle cada uno de los subprocesos mencionados anteriormente.

Tabla 9

Subproceso 1

Proceso	Gestión de Solicitudes de HelpDesk
Subproceso	Ingreso de Solicitudes

Como muestra la Tabla 9, el proceso comienza en el counter del área de soporte de la empresa WICOM DEL ECUADOR, donde la recepcionista se encarga de receptar las solicitudes ya sea de manera telefónica o mediante mail de soporte.

Aquí se registra toda la información del cliente que solicita soporte y el requerimiento específico para proceder a generar un ticket de atención que será asignado a un técnico de soporte quien será responsable de procesar el requerimiento.

Tabla 10**Subproceso 2**

Proceso	Gestión de Solicitudes de Help Desk
Subproceso	Análisis y gestión de Solicitudes

Como muestra la Tabla 10, el personal técnico es el responsable de atender las solicitudes en orden cronológico, dar seguimiento al ticket generado, comunicarse con el cliente si es necesario, ingresar el diagnóstico y solución, validar que se ha cumplido con todo lo requerido y cambiar de estado al ticket.

Tabla 11**Subproceso 3**

Proceso	Gestión de Solicitudes de Help Desk
Subproceso	Finalización de Solicitudes

Como muestra la Tabla 11, el ticket finalizado en el departamento de soporte llega nuevamente a la recepción de dicho departamento, quienes son los encargados de ingresar el costo generado y enviar al cliente los resultados de la solicitud por email.

De igual forma se realiza una llamada para la validación del nivel de atención por parte del personal técnico, reciben una calificación sobre el soporte brindado, en donde se solicita una cuantificación de acuerdo a la eficacia en tiempos y solución brindada.

Esta última información brindada servirá para validar los indicadores en nivel de servicio así como para clasificar las soluciones y poder tener una

base de conocimiento para que los próximos requerimientos puedan ser resueltos en un menor tiempo posible.

A continuación en la Figura 29 se observa el flujograma de todo el proceso de gestión de Solicitudes Online de Helpdesk.

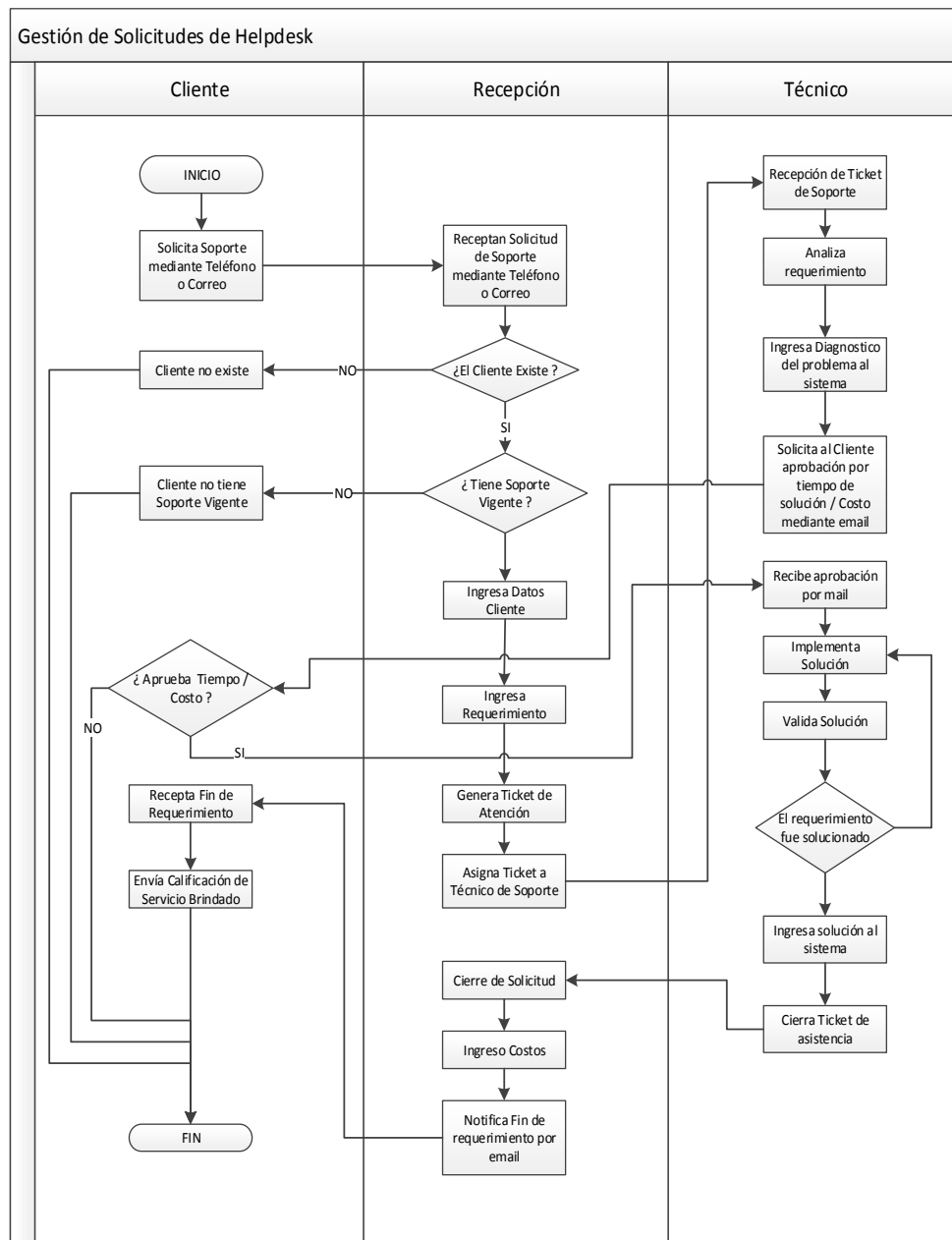


Figura 29. Flujograma de Gestión de Solicitudes Online Helpdesk

3.2 Obtención de Requisitos

Para la construcción de una aplicación, la recopilación de requisitos es fundamental, estos representan toda la funcionalidad a implementar. Toda la funcionalidad vendrá dada por todos los procesos que se desea sistematizar para cumplir nuestros objetivos.

Para la obtención de estos requisitos se utiliza el método de entrevistas con los actores involucrados en el proceso de Gestión de Solicitudes Online, este método facilitara la comprensión y recopilación de los requerimientos para cada fase de nuestro desarrollo, cada uno de estos requerimientos serán plasmados en el estándar IEEE 830 con la finalidad de crear una abstracción de la realidad del proceso y comprender de mejor manera el sistema a ser desarrollado e implementado.

3.2.1 Entrevista a Cliente

Para facilitar la recopilación de requerimientos de la aplicación a desarrollar, se formuló una entrevista al personal involucrado en el proceso con el siguiente modelo de entrevista. (Anexo A)

La presente entrevista será dirigida a todo el personal del Departamento de Soporte Técnico, los cuales están divididos de la siguiente manera.

- Administrativo
 - Gerencia Técnica
 - Jefatura
 - Supervisor
 - Recepción
 - Técnicos nivel 1
 - Técnicos nivel 2

3.2.2 Resultado Entrevistas

La entrevista fue aplicada a todo el personal del Departamento de Soporte Técnico, es en esta área es donde se desarrolla por completo el proceso de gestión de solicitudes de soporte helpdesk.

EL proceso comienza en el counter de recepción, en ese counter las dos recepcionistas de soporte reciben las solicitudes de soporte que llegan vía correo electrónico o por llamada telefónica a las oficinas, son registradas en un archivo de Excel compartido en la red y les asignan un numero de ticket, previo a esto verifican en otro archivo Excel si el cliente aún mantiene una cuenta de soporte activa y que dicha cuenta no haya caducado, de acuerdo a los paquetes de soporte que comercializa el departamento de ventas.

Esta sub área del departamento está encargada de llevar una base de datos de clientes y sus cuentas, registro de solicitudes en general, etc. De igual forma deben estar pendientes de las solicitudes sean finalizadas en un tiempo prudencial.

Además se encargan de cerrar la solicitud, colocar los costos indicados por los técnicos en caso de haberlos o de controlar las horas de soporte a descontar del paquete de horas que mantiene el cliente contratadas. Como paso final también son los encargados de registrar las calificaciones que dan los clientes a sus solicitudes terminadas para mantener un nivel de servicio adecuado.

Los niveles de satisfacción son controlados directamente por el Gerente y el Jefe del departamento respectivamente.

La asignación de Técnicos a las distintas solicitudes lo realizan las recepcionistas y el jefe del departamento de acuerdo a la carga de trabajo, importancia del cliente y nivel de prioridad.

Los técnicos reciben las solicitudes de manera verbal o por correo, ellos tienen que revisar el archivo de Excel compartido para validar su cola de

trabajo, ellos realizan todo el proceso de diagnóstico y solución del problema, registran todo lo que hacen en otra hoja del mismo archivo de Excel anotando el número de ticket para mantener una relación con la solicitud ingresada.

Piden la aprobación del cliente del costo si lo hay o de las a horas a descontar del paquete de soporte contratado. Cuando hayan dado por finalizado al soporte comunican mediante correo o verbalmente a la recepcionista para que notifique al cliente que su solicitud ha finalizado.

Como se puede apreciar si bien ocupan herramientas de ofimática el proceso se lo lleva de una manera casi manual, ocasionando muchas veces perdidas económicas y un nivel de servicio bajo.

De las entrevistas se pudo determinar los siguientes procesos:

1. Recepción / Ingreso de Solicitudes
2. Análisis y Gestión de Solicitudes
3. Finalización de Solicitudes

Para una comprensión más a detalle se procede a explicar cada uno de los procesos.

3.2.2.1 Recepción / Ingreso de solicitudes

Tabla 12

Proceso recepción / Ingreso Solicitudes

Proceso	Recepción / Ingreso de Solicitudes
Responsable	Recepcionistas Jefe Gerente
Descripción	<ul style="list-style-type: none"> • Recepción de Solicitudes • Ingreso de clientes

Continúa →

<ul style="list-style-type: none"> • Ingreso de Solicitudes • Validación de Cuentas • Asignación de Técnico • Generación de Ticket Soporte
--

3.2.2.2 Análisis y Gestión de solicitudes

Tabla 13

Proceso Análisis y Gestión de Solicitudes

Proceso	Análisis y Gestión de Solicitudes
Responsable	Técnicos Supervisor Jefe Gerente
Descripción	<ul style="list-style-type: none"> • Recepción de Solicitud • Diagnóstico / Solución del requerimiento • Registro de Solución • Solicita aprobación costo / tiempo soporte • Cierre de ticket

3.2.2.3 Finalización de solicitudes

Tabla 14

Finalización de Solicitudes

Proceso	Finalización de Solicitudes
Responsable	Recepcionistas Jefe

Continúa →

Gerente	
Descripción	<ul style="list-style-type: none"> • Recibe cierre de ticket • Ingresa costos / tiempo • Finaliza Solicitud • Notifica Cliente • Recibe calificación nivel de servicio

3.2.3 Especificación de Requisitos de Software G.S.O.

3.2.3.1 Introducción

El presente documento de especificación de requerimientos de software, se describen los requisitos que cumplirá el sistema G.S.O. “Gestión de Solicitudes Online” y su funcionalidad dentro del departamento de soporte técnico de la empresa WICOM del Ecuador. El presente documento cumple con todos los estándares y especificaciones establecidas en la norma IEEE 830.

A. Objetivo

El objetivo de la Especificación de Requerimientos de Software, es establecer parámetros definidos sobre el diseño y la funcionalidad que la aplicación G.S.O. debe cumplir dentro de la empresa WICOM del Ecuador ajustándose a su lógica de negocio.

B. Alcance

Las especificaciones están dirigidas al equipo del trabajo del proyecto, conformado por desarrolladores, jefatura y gerencia del Departamento Técnico de WICOM del Ecuador, así como a los usuarios finales del sistema, en este documento quedara plasmado de forma definitiva todas las características funcionales y los requerimientos tanto de software y

hardware, constituyendo en el documento guía sobre el cual se desarrollara el sistema G.S.O.

El sistema G.S.O. será una aplicación web para la Gestión de Solicitudes Online que estará constituido por varias funciones y característica que se procede a detallar a continuación:

- Gestionar Empresas
- Gestionar Usuarios / Clientes / Técnicos
- Gestionar Tablas
- Gestionar Solicitudes
- Reportería
- Auditoria

G.S.O. mantendrá todos los registros ingresados al sistema, disponibles en todo momento, solicitudes y la respectiva solución a los distintos requerimientos ingresados, de igual forma se puede apreciar toda la reportería de una manera ágil y en todo momento.

C. Definiciones

Para una mejor comprensión de este documento se muestra a continuación las definiciones y significados de las siglas que se usan de aquí en adelante.

- IEEE. Institute of Electrical and Electronics Engineers
- ERS. Especificación de Requerimientos del Sistema
- GSO. Gestión de Solicitudes Online

3.2.3.2 Descripción General

Se procederá a sistematizar mediante el sistema GSO el proceso de soporte “helpdesk” a clientes, servicio ofrecido por la empresa WICOM del Ecuador, permite llevar un control a detalle de los soportes brindados así

como tener indicadores de nivel de servicio brindado por el departamento y carga de trabajo por cada técnico asignado.

El sistema será desarrollado en su totalidad para un ambiente web, funcionara dentro de su intranet alojado en un servidor Linux, su acceso será mediante los navegadores Chrome, Firefox e Internet Explorer siendo los dos primeros los recomendados por su nivel de compatibilidad con las nuevas tecnologías por lado del cliente.

GSO mantendrá la administración de usuarios, empresas, clientes, solicitudes, soluciones y calificaciones, manteniendo registros históricos para esto mantendrá una eliminación lógica ningún registro será borrado de la base de datos, simplemente se maneja banderas para que pueda ser o no visualizado por los usuarios.

Se ha adoptado el nombre de GSO “Gestión de Solicitudes Online” para el sistema debido al nombre del proceso que se está implementando y por mantener la línea de negocio, a lo cual, solo se lo trata como un valor agregado al estar publicado en la intranet con la posibilidad de que sea publicado al internet si así se lo requiere.

A. Interface de GSO

GSO cuenta con una interface muy amigable e intuitiva, será parte integral del diseño contar con menús claros y simples, toda la gama de colores de acuerdo a lo solicitado por el Departamento de Soporte Técnico y manteniendo los colores corporativos de WICOM del Ecuador.

La distribución de los espacios de trabajo y las características principales del sistema estarán diseñadas exclusivamente para el usuario final, se usa una paleta de colores con pocos elementos, esto con la finalidad de no cansar la vista de los usuarios, los mismos que se lo utilizar día tras día.

Estas son algunas de las características de la interface que maneja GSO:

- Menú dinámico horizontal, conlleva una lógica de carpetas de almacenamiento para mantener organizado nuestros procesos
- Menú de navegación para movernos de sección en sección de un amañera rápida y eficiente.
- Formularios cortos y claros, con validación en línea para evitar procesamiento innecesario.
- Ayuda informativas tipo burbuja.
- Cabecera con el logo del sistema, logo de la empresa, nombre de la empresa, nombre y versión del sistema.
- Bajo el menú una franja informativa del usuario con nombres, perfil, empresa y fecha.
- Menú con iconos para identificar de mejor manera los procesos a realizar.
- Mensajes de error e información claros para el usuario.

Se revisa brevemente el esquema visual de la aplicación en la cual se desenvolverá el usuario.

Nomenclatura usada en la plantilla visual que se muestra en la Figura 30 para definir la estructura del sistema:

- a) **Cabecera:** Aquí se tiene la información del sistema como logotipo, nombre y versión, también la información de la empresa como nombre y logotipo.
- b) **Menú:** Es la sección donde ira alojado el menú dinámico.
- c) **Menú Navegación:** Es la sección fija donde se despliega el menú de navegación.
- d) **Menú Información:** Es una sección fija de carácter informativo, aquí se muestra la información del usuario como perfiles, nombres y empresa a la que pertenece.
- e) **Escritorio Trabajo:** Esta es la sección principal, será una sección móvil, aquí se despliega todos los procesos definidos, formularios, reportes, etc.

- f) **Pie:** Esta sección es fija; contendrá información de la empresa e información de derechos y autoría.

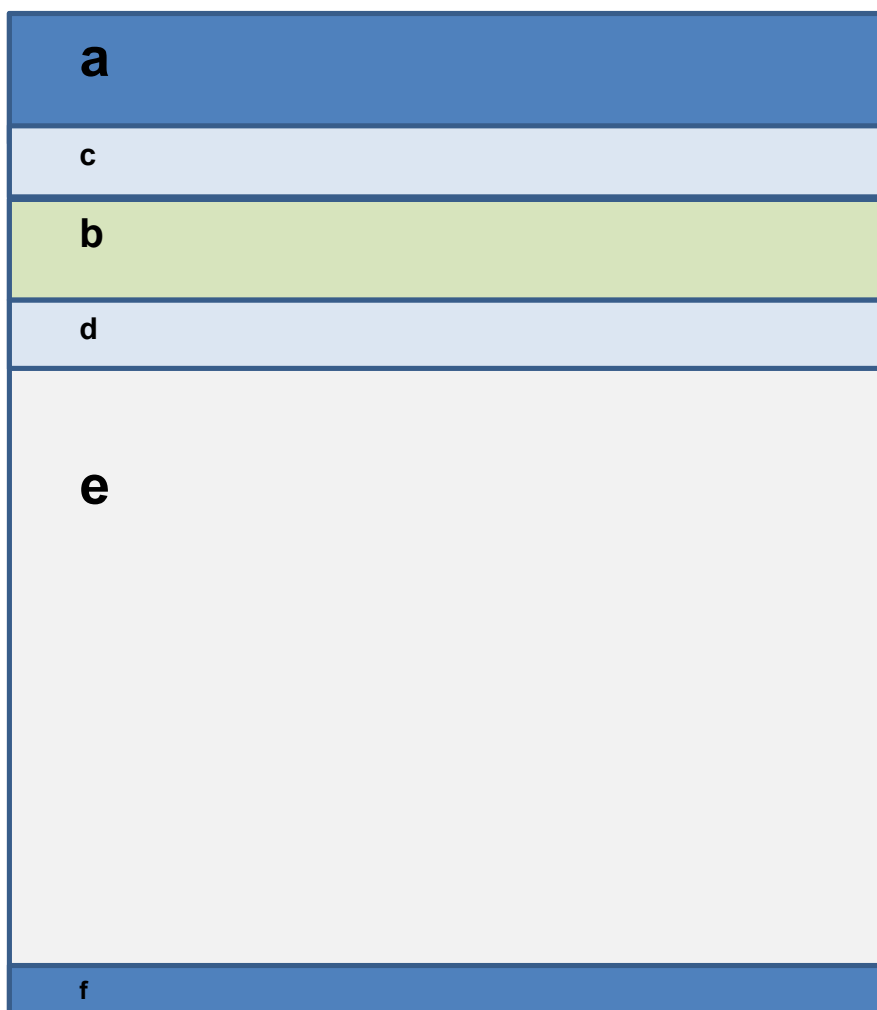


Figura 30. Plantilla Interface Gráfica GSO

El acceso al sistema GSO se lo realiza mediante cualquier navegador digitando una url en nuestra barra de direcciones, el URL estará formado de la siguiente manera <http://{Nombre del Servidor o IP}/WICOMGso/> esto abrirá automáticamente una nueva pantalla de navegación completa, esto lo realiza siempre y cuando nuestro navegador este configurado para permitir abrir popups, caso contrario se tiene que abrir manualmente presionando en el botón que aparecerá en la pantalla.

B. Hardware

Para la implementación del aplicativo web se requiere disponer de un servidor de carga media para no afectar el rendimiento de la aplicación, en este servidor se instala nuestro servicio web y nuestra base de datos.

Se puede especificar como requerimientos mínimos hardware para el servidor lo siguiente:

- 2GB memoria RAM
- 500MB Espacio en Disco
- Procesador Intel

Para los usuarios que ingresan desde los terminales con sus navegadores los requisitos mínimos de hardware serían los siguientes:

- 1GB de memoria Ram
- 500MB Espacio en Disco

C. Software

Para el desarrollo e implementación del aplicativo web GSO se utiliza las siguientes herramientas de software.

Por la parte del servidor se necesita lo siguiente:

- Linux Centos
- MySQL Server 5.0.x
- Apache 2.0
- PHP 5.2.x
- PEAR DB 1.7.6

Por el lado del desarrollador se necesita lo siguiente:

- NetBeans 7.x

Por el lado del cliente que accederá a la aplicación GSO

- Windows, Linux o MacOS
- Chrome, Firefox, Safari o Internet Explorer

D. Infraestructura Comunicación

GSO estará publicado en la intranet e internet de la empresa WICOM del Ecuador, los usuarios y clientes podrán acceder mediante sus navegadores y consumir la aplicación, la interacción con el servidor web se lo realiza utilizando el protocolo TCP/IP por el puerto por defecto 80 y este a su vez con la base de datos lo realizará de igual forma por el protocolo TCP/IP utilizando el puerto por defecto de MySQL 3306.

En la Figura 31 se puede observar más a detalle lo expuesto anteriormente.

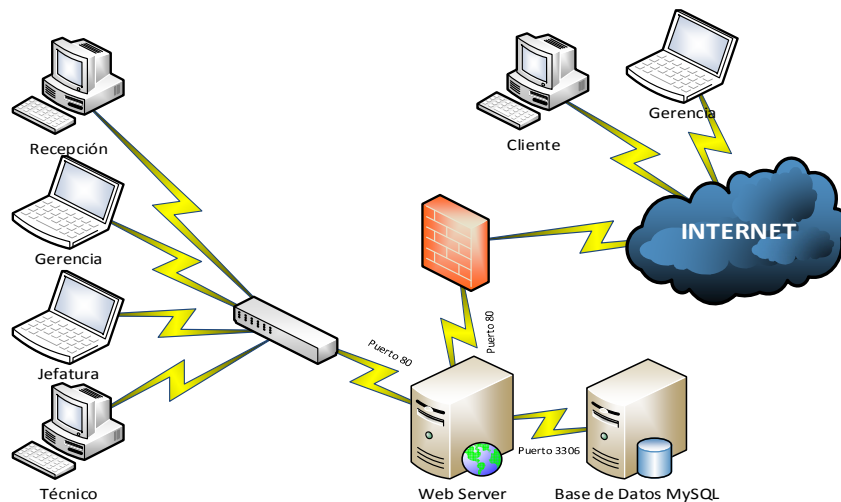


Figura 31. Infraestructura de comunicación

E. Características Funcionales de GSO

GSO cuenta con las siguientes características funcionales:

- Esquema de seguridad de robusta basada en sesiones.
- Caducidad de sesiones de trabajo.
- Manejo de perfiles de usuarios a nivel de menús.

- Interface de usuario amigable de fácil uso.
- Generación de reportes estadísticos gerenciales para el control de tiempos de atención y calificación de atención.
- Creación de n Departamentos o Áreas de trabajo a las que pertenecen sus clientes “Gerencia, Contabilidad, Sistemas, Marketing, etc.”
- Creación de n Cargos para usuarios “Ingeniero, Doctor, Abogado, Gerente, Administrador, etc.”
- Creación de n Usuarios.
- Creación de n Empresas Clientes.
- Creación de n tipos de solicitudes para una mejor categorización de las mismas.
- Caducidad de tiempo de soporte por empresa o usuario.
- Control de acceso a la web.
- Ingreso de solicitudes mediante personal en call center.
- Ingreso de solicitudes mediante intranet/internet de sus clientes.
- Asignación de solicitudes a personal de soporte.
- Mini foro por solicitud ingresada.
- Impresión de solicitudes.
- Alertas automáticas de solicitudes vía mail en el ingreso, atención, finalización.
- Reportes de usuarios, empresas, solicitudes, calificaciones, etc.
- Filtros de búsqueda y ordenamiento interactivo.
- Impresión de reportes.

F. Perfiles de Usuarios

El sistema GSO cuenta con 4 perfiles activos para el control de roles dentro del mismo, estos son:

- **Webmaster.** Dentro de los privilegios que tiene este usuario está el poder revisar absolutamente toda la información del sistema,

gestionar usuarios, empresas, departamentos, cargos, solicitudes, etc.

- **Técnico.** Podrá gestionar solicitudes y todo lo que respecta con la solución del requerimiento, tendrá acceso a información limitada como clientes, solicitudes así mismo asignadas, no podrá visualizar solicitudes de otros técnicos.
- **Cliente.** Solo podrá ingresar solicitudes de soporte siempre y cuando su cuenta este activa, revisar sus soportes actuales o históricos, calificar por los soportes recibidos, esta última opción será obligatoria para poder ingresar nuevos requerimientos.
- **Supervisor.** Podrá gestionar todo lo referente a solicitudes, empresas clientes, departamentos, cargos, etc. re asignar técnicos solicitudes abiertas, validar la carga de trabajo por técnico así como ver los niveles de satisfacción de servicio.

G. Auditoria

GSO cuenta con una auditoria, en la cual se registra todas las transacciones que se realizan a la base de datos, además de contar con auditoria de acceso al sistema para llevar un control y evitar posibles fraudes o manipulación de la información.

La auditoría será guardada en la base de datos y de igual forma replicada a archivos tipo log como respaldo de la auditoria.

H. Seguridad

GSO cuenta con las siguientes funciones de seguridad y control:

- Validación de Ingreso de usuario
- Control de id único de sesión
- Validación de perfiles
- Validación de alertas de información, error y control

3.2.3.3 Requerimientos funcionales

Mediante las entrevistas realizadas a todo el personal del departamento de soporte de la empresa WICOM del Ecuador, se pudo recopilar la información necesaria y condensarla en los siguientes requerimientos funcionales:

Tabla 15

Requerimientos Funcionales GSO

Sistema	Proceso	Procedimiento
GSO	Iniciar Sesión	Valida Usuario Crea Sesión
	Cambiar Clave	Cambiar clave Cerrar Sesión
	Gestionar Tablas	Crear Editar Buscar
	Gestionar Usuarios	Crear Editar Buscar
	Gestionar Empresas	Crear Editar Buscar
	Gestionar Clientes	Crear Editar Buscar
	Gestionar Técnicos	Crear Editar Buscar
	Gestionar Solicitudes	Crear Buscar Agregar Observaciones Reasignar Cliente Reasignar Técnico Finalizar Ticket Finalizar Solicitud

Continúa →

	Anular Solicitud Imprimir Solicitud
Generar Reportes	Usuarios Empresas Clientes Solicitudes Ficha Técnica Nivel de Servicio Carga de Trabajo Tiempos de Atención Auditoria
Calificar Atención	Valida Solicitud Finalizada Calificar Solicitud

3.2.3.4 Requerimientos No Funcionales

Estos requerimientos describen como debe comportarse el sistema, no están sujetos a cambios y deben formar obligatoriamente parte de la solución de desarrollo, para esto se lo dividirá en 4 aspectos principales dentro de la creación del software G.S.O. que son:

Tabla 16

Requerimientos No Funcionales

Atributo	Descripción	Medida
Desempeño	Consulta de información	La transaccionalidad del sistema con la base de datos no debe tener demoras ni interrumpir los demás procesos en el sistema, el tiempo garantizado por transacción deberá ser los 3 Segundos.
Disponibilidad	Fallos Externos	El sistema debe garantizar que durante una pérdida de conexión con la base de datos en cualquier espacio de tiempo la integridad de

Continúa →

la información se mantenga.		
Modificabilidad	Reglas de Negocio	El desarrollador debe ser capaz de agregar reglas de negocio de una manera fácil y simple, sin muchas líneas de programación.
Seguridad	Accesos no Autorizados	El sistema debe garantizar la seguridad de la información, ningún módulo o documento debe ser abierto sin antes haberse logeado en el sistema.

3.3 Análisis y Diseño

Los requerimientos para el desarrollo del software GSO, están especificados de la norma IEEE-830 y documentado con la especificación de requerimientos de software, de esta manera se puede obtener diseños adecuados a partir de los requerimientos recabados que permita realizar un desarrollo uniforme y claro del sistema.

Para el diseño de los modelos del sistema se dispone de los dos tipos de diagramas especificados anteriormente en el capítulo II, diagramas estáticos y diagramas dinámicos.

A continuación estos son los diagramas que serán utilizados para el desarrollo del sistema GSO por la contabilidad con los requisitos propuestos.

- Diagramas Estáticos
 - Diagramas de Casos de Uso
 - Diagramas de Objetos
 - Diagramas de Componentes
 - Diagramas de Despliegue
- Diagramas Dinámicos
 - Diagramas de Estados
 - Diagramas de Secuencia

3.3.1 Diagramas casos de uso

Los diagramas de Casos de Uso se los usa para definir claramente y visualmente la funcionalidad del sistema como lo muestra la Figura 32 de la funcionalidad del sistema GSO.

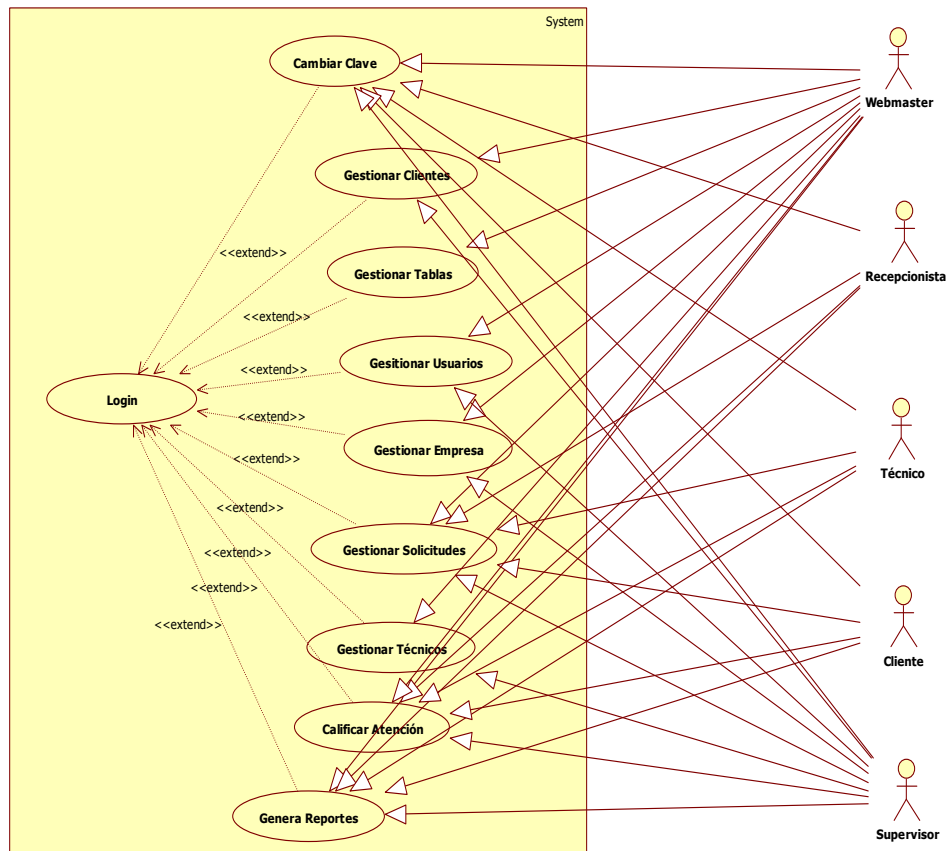


Figura 32. Diagrama caso de uso requisitos Funcionales GSO

Basado en la Figura 32 y en la Tabla 16 se procede a crear los casos de uso específico y detallado de todos los requisitos funcionales del sistema GSO.

3.3.1.1 Caso de uso Login

A continuación se detalla el caso de uso Login con la Figura 33 y la interacción con el sistema en el Cuadro 1.

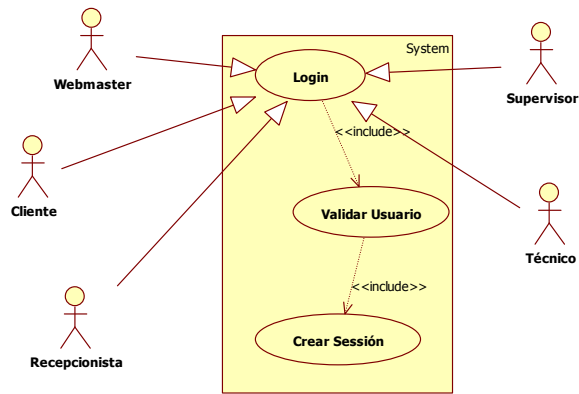


Figura 33. Caso de uso Login

Cuadro 1
Detalle caso de uso Login

CASO DE USO	
Identificación	1.1
Nombre	Login
Descripción	Permite al usuario iniciar sesión en el sistema, valida su perfil de acuerdo al cual se muestra la cinta de opciones disponibles en el menú del sistema, valida si el usuario existe, si está activo, si tiene acceso a la web y que su contraseña está correctamente ingresada y crea una sesión para control de tiempo dentro del sistema.
Actores	Webmaster, Supervisor, Técnico, Recepcionista, Cliente
Frecuencia	Siempre
Precondiciones	Los usuarios deben estar registrados en la base de datos.

Flujo Normal	
	<ol style="list-style-type: none"> 1. El sistema muestra el formulario de login. 2. El actor ingresa: Usuario y Contraseña. 3. El sistema Valida si el usuario existe y la contraseña esta correcta. 4. El sistema procede a crear la sesión y dar acceso normal. 5. El sistema despliega el menú de acuerdo a los permisos otorgados en su perfil.
Flujo Alternativo	
	<ol style="list-style-type: none"> 3.A. El sistema valido el usuario y la contraseña si no son correctas, muestra un mensaje al actor para que pueda volver a ingresarlos.

3.3.1.2 Caso de uso Cambiar Clave.

A continuación se detalla el caso de uso Cambiar Clave con la Figura 34 y la interacción con el sistema en el Cuadro 2.

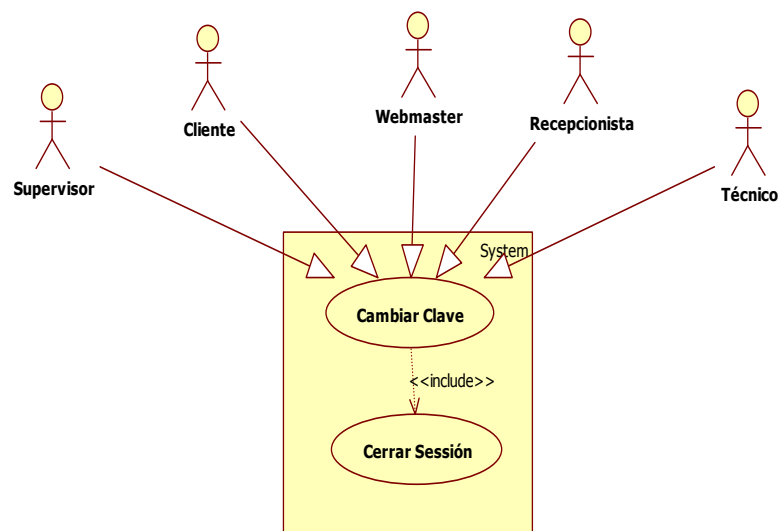


Figura 34. Caso de uso cambiar Clave

Cuadro 2**Detalle caso de uso Cambiar Clave**

CASO DE USO	
Identificación	1.2
Nombre	Cambiar Clave
Descripción	<p>Permite que el actor ya dentro del sistema pueda cambiar su contraseña, si el cambio es exitoso el sistema cierra la sesión para volver a ingresar con la nueva contraseña.</p>
Actores	<p>Webmaster, Supervisor, Técnico, Recepcionista, Cliente</p>
Frecuencia	Ocasional
Precondiciones	<p>El actor debe haber realizado login en el sistema (Login 1.1).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona cambiar contraseña en el menú. 3. Ingresa campo obligatorio Clave Actual. 4. Ingresa campo obligatorio Clave nueva. 5. Ingresa campo obligatorio Confirmar Clave Nueva. 6. El sistema valida datos ingresados y modifica la nueva clave. 7. El sistema cierra la sesión par que vuelva a realizar (Login 1.1) en el sistema.
Flujo Alternativo	

Continúa →

6.A. El sistema valida los datos ingresados, si no es correcto, muestra un mensaje al actor para que pueda volver a ingresarlos y no modifica la clave actual.

3.3.1.3 Caso de uso Gestionar Tablas

A continuación se detalla el caso de uso Gestionar Tablas con la Figura 35 y la interacción con el sistema en el cuadro .3, 4, 5, 6.

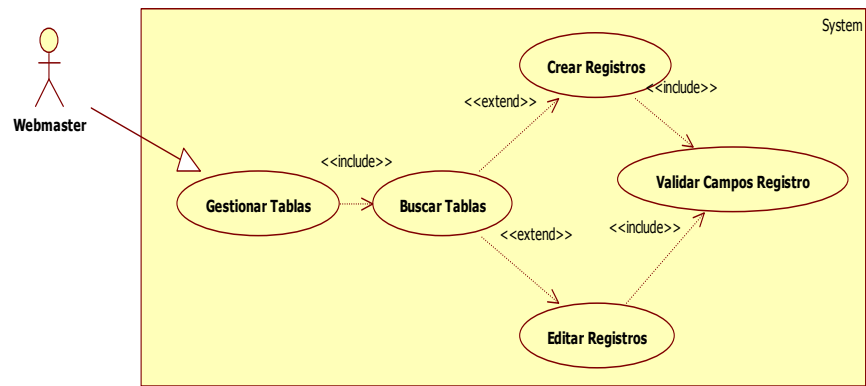


Figura 35. Caso de uso Gestionar Tablas

Cuadro 3
Detalle Caso de Uso Buscar Tablas

CASO DE USO	
Identificación	1.3
Nombre	Buscar Tablas
Descripción	Permite que el actor pueda buscar una tabla especifica en el maestro de tablas, y desplegar información como tipos de solicitudes, configuraciones del sistema, etc.
Actores	

Continúa →

Webmaster	
Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado login en el sistema (Login 1.1).	
Flujo Normal	
1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Configuración > Tablas. 3. El sistema despliega formulario de búsqueda. 4. Dentro de la lista desplegable se puede seleccionar la tabla deseada. 5. El sistema Muestra una lista con los registros disponibles en dicha tabla.	
Flujo Alternativo	
5.A. Si no existen registros dentro de la tabla no se despliega nada en pantalla.	

Cuadro 4

Detalle Caso de Uso Crear Registros

CASO DE USO	
Identificación	1.4
Nombre	Crear Registros
Descripción	Permite que el actor pueda buscar una tabla específica en el maestro de tablas e insertar registros de acuerdo a su necesidad.
Actores	

Continúa →

Webmaster	
Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado Buscar Tablas (Buscar Tablas 1.3).	
Flujo Normal	
<p>1. Realizada la Búsqueda de tablas (Buscar Tablas 1.3).</p> <p>2. Selecciona la opción nuevo registro.</p> <p>3. El sistema despliega formulario de ingreso de registros con los siguientes campos:</p> <ul style="list-style-type: none"> • Código (Requerido) • Dato 1 • Dato 2 • Dato 3 • Dato 4 • Valor 1 • Valor 2 • Valor 3 • Valor 4 • Texto 1 <p>4. El sistema valida datos de registro (Valida Campos Registro 1.6).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	
<p>4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.</p> <p>5.A. El sistema valida que el código de registro a ingresar en esa tabla no sea repetido, si lo es, envía un mensaje de alerta al actor u no guarda los datos.</p>	

Cuadro 5
Detalle Caso de Uso Editar Registros

CASO DE USO	
Identificación	1.5
Nombre	Editar Registros
Descripción	<p>Permite que el actor pueda buscar una tabla específica en el maestro de tablas y editar registros de acuerdo a su necesidad.</p>
Actores	<p>Webmaster</p>
Frecuencia	Ocasional
Precondiciones	<p>El actor debe haber realizado Buscar Tablas (Buscar Tablas 1.3).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de tablas (Buscar Tablas 1.3). 2. Seleccionar en el registro a editar en la lista desplegada. 3. El sistema carga el formulario con los datos del registro seleccionado en los siguientes campos: <ul style="list-style-type: none"> • Código (Requerido) (Bloqueado) • Dato 1 • Dato 2 • Dato 3 • Dato 4 • Valor 1 • Valor 2 • Valor 3

<ul style="list-style-type: none"> • Valor 4 • Texto 1 <p>4. El actor edita los campos requeridos.</p> <p>5. El sistema valida datos de registro a editar (Valida Campos Registro 1.6).</p> <p>6. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	
<p>5.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.</p>	

Cuadro 6

Detalle Caso de Uso Valida Campos Registro

CASO DE USO	
Identificación	1.6
Nombre	Valida Campos Registro
Descripción	<p>Permite validar los campos requeridos de los registros de una tabla específica en el maestro de tablas.</p>
Actores	<p>Webmaster</p>
Frecuencia	<p>Ocasional</p>
Precondiciones	<p>(Crear Registros 1.5)</p>

Continúa →

(Editar Registros 1.4)	
Flujo Normal	
<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • Código (Requerido) (Numérico) 	
Flujo Alternativo	
<p>1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.</p>	

3.3.1.4 Caso de Uso Gestionar Clientes

A continuación se detalla el caso de uso Gestionar Tablas con la Figura 36 y la interacción con el sistema en el Cuadro 7, 8, 9, 10.

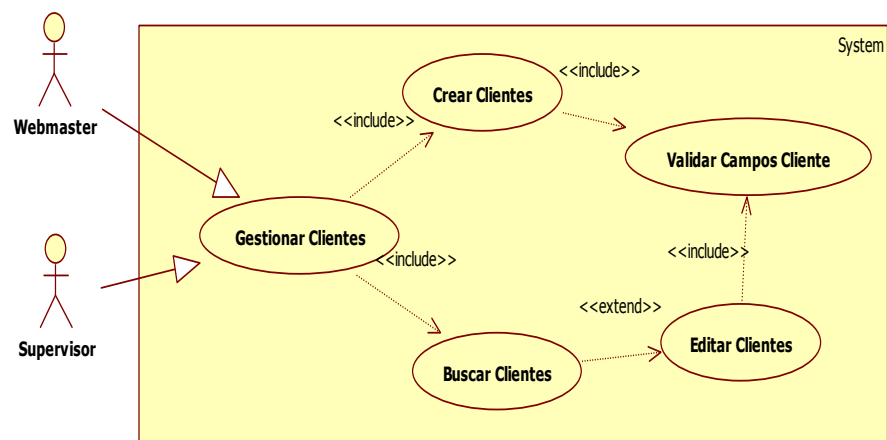


Figura 36. Caso de Uso Gestionar Clientes

Cuadro 7**Detalle Caso de Uso Crear Clientes**

CASO DE USO	
Identificación	1.7
Nombre	Crear Clientes
Descripción	Permite que el actor pueda crear nuevos clientes en el sistema de acuerdo a su necesidad.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado login en el sistema (Login 1.1).
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Clientes 3. El sistema despliega formulario de ingreso de clientes con los siguientes campos: <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) • Perfil (Requerido, Bloqueado, Seleccionado Cliente) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo

Continúa →

<ul style="list-style-type: none"> • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) <p>4. El sistema valida datos de Cliente (Valida Campos Cliente 1.10).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	<p>4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.</p> <p>5.A. El sistema valida que el usuario no exista en el sistema, si existe, envía un mensaje de alerta al actor y no guarda los datos hasta su rectificación.</p>

Cuadro 8

Detalle Caso de Uso Buscar Clientes

CASO DE USO	
Identificación	1.8
Nombre	Buscar Clientes
Descripción	Permite que el actor pueda buscar clientes en el sistema.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	Cliente debe estar registrado en el sistema (Crear Clientes 1.7).
Flujo Normal	

<p>1. El actor debe iniciar sesión (Login 1.1).</p> <p>2. Selecciona el menú Administración > Clientes.</p> <p>3. El sistema despliega formulario de búsqueda.</p> <p>4. El actor puede utilizar los siguientes filtros de búsqueda:</p> <ul style="list-style-type: none"> • Id • Empresa • Usuario • Nombre • Apellido • Email • Estado • Departamento • Teléfono Móvil • Teléfono Fijo • Acceso • Fecha de Registro • Fecha que caduca • Todos los Registros <p>5. El sistema muestra una lista con los registros disponibles, se puede seleccionar el registro deseado para desplegar toda la información.</p>	
Flujo Alternativo	
<p>5.A. Si no existen registros dentro de la tabla el sistema muestra un mensaje que no se encuentran registros.</p>	

Cuadro 9

Detalle Caso de Uso Editar Clientes

CASO DE USO	
Identificación	1.9
Nombre	Editar Clientes

Continúa →

Descripción	
	Permite que el actor pueda editar clientes.
Actores	
	Webmaster, Supervisor
Frecuencia	
	Ocasional
Precondiciones	
	El actor debe haber realizado Buscar Clientes (Buscar Clientes 1.8).
Flujo Normal	
	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de clientes (Buscar Clientes 1.8). 2. Seleccionar el cliente a editar en la lista desplegada, esto muestra toda la información del cliente. 3. Seleccionar modificar en la pantalla de información. 4. El sistema carga el formulario con los datos del cliente seleccionado en los siguientes campos: <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Perfil (Requerido, Bloqueado, Seleccionado Cliente) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) 5. El actor edita los campos requeridos. 6. El sistema valida datos del cliente a editar (Valida Campos Cliente 1.10). 7. El sistema guarda los datos ingresados.

Flujo Alternativo	
6.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.	

Cuadro 10

Detalle Caso de Uso Valida Campos Cliente

CASO DE USO	
Identificación	1.10
Nombre	Valida Campos Cliente
Descripción	Permite validar los campos requeridos del cliente.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	(Crear Clientes 1.7) (Editar Clientes 1.9)
Flujo Normal	<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) (Crear Clientes 1.7) • Perfil (Requerido, Bloqueado, Seleccionado Cliente) (Crear Clientes 1.7) • Fecha de Ingreso (Requerido) • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido)

<ul style="list-style-type: none"> • Área / Departamento (Requerido) • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) 	
Flujo Alternativo	
1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.	

3.3.1.5 Caso de Uso Gestionar Usuarios

A continuación se detalla el caso de uso Gestionar Usuarios con la Figura 37 y la interacción con el sistema en el cuadro 11, 12, 13, 14.

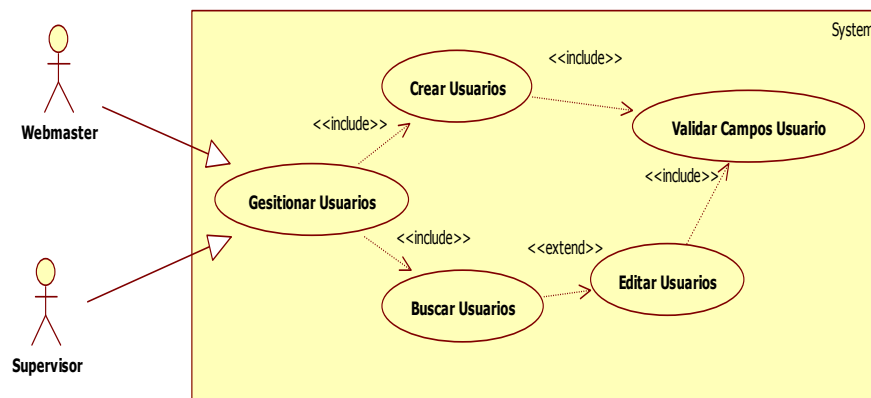


Figura 37. Caso de Uso Gestionar Usuarios

Cuadro 11

Detalle Caso de Uso Crear Usuarios

CASO DE USO	
Identificación	1.11
Nombre	Crear usuarios

Continúa →

Descripción	
	Permite que el actor pueda crear nuevos Usuarios en el sistema de acuerdo a su necesidad.
Actores	
	Webmaster, Supervisor
Frecuencia	
	Ocasional
Precondiciones	
	El actor debe haber realizado login en el sistema (Login 1.1).
Flujo Normal	
	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Usuarios 3. El sistema despliega formulario de ingreso de usuarios con los siguientes campos: <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) • Perfil (Requerido) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) 4. El sistema valida datos de usuario (Valida Campos Usuario 1.14). 5. El sistema guarda los datos ingresados.
Flujo Alternativo	

Continúa →

4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.

5.A. El sistema valida que el usuario no exista en el sistema, si existe, envía un mensaje de alerta al actor y no guarda los datos hasta su rectificación.

Cuadro 12

Detalle Caso de Uso Buscar Usuarios

CASO DE USO	
Identificación	1.12
Nombre	Buscar Usuarios
Descripción	Permite que el actor pueda buscar usuarios en el sistema.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	Usuario debe estar registrado en el sistema (Crear Usuarios 1.11).
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Usuarios. 3. El sistema despliega formulario de búsqueda. 4. El actor puede utilizar los siguientes filtros de búsqueda: <ul style="list-style-type: none"> • Id

- Empresa
- Usuario
- Nombre
- Apellido
- Perfil
- Email
- Estado
- Departamento
- Teléfono Móvil
- Teléfono Fijo
- Acceso
- Fecha de Registro
- Fecha que caduca
- Todos los Registros

5. El sistema muestra una lista con los registros disponibles, se puede seleccionar el registro deseado para desplegar toda la información.

Flujo Alternativo

5.A. Si no existen registros dentro de la tabla el sistema muestra un mensaje que no se encuentran registros.

Cuadro 13

Detalle Caso de Uso Editar Usuarios

CASO DE USO	
Identificación	1.13
Nombre	Editar Usuarios
Descripción	Permite que el actor pueda editar usuarios.
Actores	Webmaster, Supervisor

Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado Buscar Usuarios (Buscar Usuarios 1.12).	
Flujo Normal	
<ol style="list-style-type: none"> 1. Realizada la Búsqueda de usuarios (Buscar Usuario 1.12). 2. Seleccionar el usuario a editar en la lista desplegada, esto muestra la información del usuario. 3. Seleccionar modificar en la pantalla de información. 4. El sistema carga el formulario con los datos del usuario seleccionado en los siguientes campos: <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Perfil (Requerido) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) 5. El actor edita los campos requeridos. 6. El sistema valida datos del usuario a editar (Valida Campos Usuario 1.14). 7. El sistema guarda los datos ingresados. 	
Flujo Alternativo	
6.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.	

Cuadro 14***Detalle Caso de Uso Valida Campos Usuario***

CASO DE USO	
Identificación	1.14
Nombre	Valida Campos Usuario
Descripción	Permite validar los campos requeridos del usuario.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	(Crear Usuarios 1.11) (Editar Usuarios 1.13)
Flujo Normal	<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) (Crear Usuarios 1.11) • Perfil (Requerido) (Crear Usuario 1.11) • Fecha de Ingreso (Requerido) • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido) • Área / Departamento (Requerido) • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido)
Flujo Alternativo	

Continúa →

1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.

3.3.1.6 Caso de Uso Gestionar Empresas

A continuación se detalla el caso de uso Gestionar Empresas con la Figura 38 y la interacción con el sistema en el Cuadro 15, 16, 17, 18.

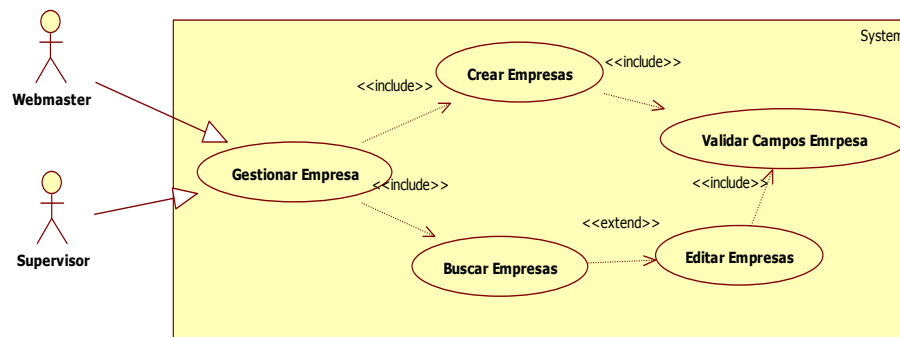


Figura 38. Caso de Uso Gestionar Empresas

Cuadro 15

Detalle Caso de Uso Crear Empresas

CASO DE USO	
Identificación	1.15
Nombre	Crear Empresas
Descripción	Permite que el actor pueda crear nuevas Empresas en el sistema de acuerdo a su necesidad.
Actores	Webmaster, Supervisor

Continúa →

Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado login en el sistema (Login 1.1).	
Flujo Normal	
<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Empresas 3. El sistema despliega formulario de ingreso de empresas con los siguientes campos: <ul style="list-style-type: none"> • RUC (Requerido) • Empresa Nombre (Requerido) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Empresa Activa (Requerido, SI/NO) 4. El sistema valida datos de empresa (Valida Campos Empresa 1.18). 5. El sistema guarda los datos ingresados. 	
Flujo Alternativo	
<p>4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.</p> <p>5.A. El sistema valida que el RUC no exista en el sistema, si existe, envía un mensaje de alerta al actor y no guarda los datos hasta su rectificación.</p> <p>5.B El RUC es validado de acuerdo a los estándares de Longitud y dígito verificador en el Ecuador, si no cumple envía una alerta para que sea ingresado correctamente.</p>	

Cuadro 16***Detalle Caso de Uso Buscar Empresas***

CASO DE USO	
Identificación	1.16
Nombre	Buscar Empresas
Descripción	Permite que el actor pueda buscar empresas en el sistema.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	Empresa debe estar registrada en el sistema (Crear Empresa 1.15).
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Empresas. 3. El sistema despliega formulario de búsqueda. 4. El actor puede utilizar los siguientes filtros de búsqueda: <ul style="list-style-type: none"> • Id • RUC • Empresa • Fecha de Registro • Fecha que caduca • Todos los Registros 5. El sistema muestra una lista con los registros disponibles, se puede seleccionar el registro deseado para desplegar toda la información.

Continúa →

Flujo Alternativo	
	5.A. Si no existen registros dentro de la tabla el sistema muestra un mensaje que no se encuentran registros.

Cuadro 17

Detalle Caso de Uso Editar Empresas

CASO DE USO	
Identificación	1.17
Nombre	Editar Empresas
Descripción	Permite que el actor pueda editar empresas.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado Buscar Empresas (Buscar Empresas 1.16).
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de empresas (Buscar Empresas 1.16). 2. Seleccionar la empresa a editar en la lista desplegada, esto muestra la información de la empresa. 3. Seleccionar modificar en la pantalla de información. 4. El sistema carga el formulario con los datos de la empresa seleccionada en los siguientes campos: <ul style="list-style-type: none"> • RUC (Requerido, Bloqueado)

Continúa →

<ul style="list-style-type: none"> • Empresa Nombre (Requerido) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Empresa Activa (Requerido, SI/NO) <p>5. El actor edita los campos requeridos.</p> <p>6. El sistema valida datos de la empresa a editar (Valida Campos Empresa 1.18).</p> <p>7. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	
6.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.	

Cuadro 18

Detalle Caso de Uso Valida Campos Empresa

CASO DE USO	
Identificación	1.18
Nombre	Valida Campos Empresa
Descripción	Permite validar los campos requeridos de la empresa.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	(Crear Empresas 1.15) (Editar Empresas 1.17)

Continúa →

Flujo Normal	
<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • RUC (Requerido)(Crear Empresas 1.15) • Empresa Nombre (Requerido) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Empresa Activa (Requerido, SI/NO) 	
Flujo Alternativo	
<p>1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.</p>	

3.3.1.7 Caso de Uso Gestionar Solicitudes

A continuación se detalla el caso de uso Gestionar Empresas con la Figura 39 y la interacción con el sistema en el Cuadro 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29.

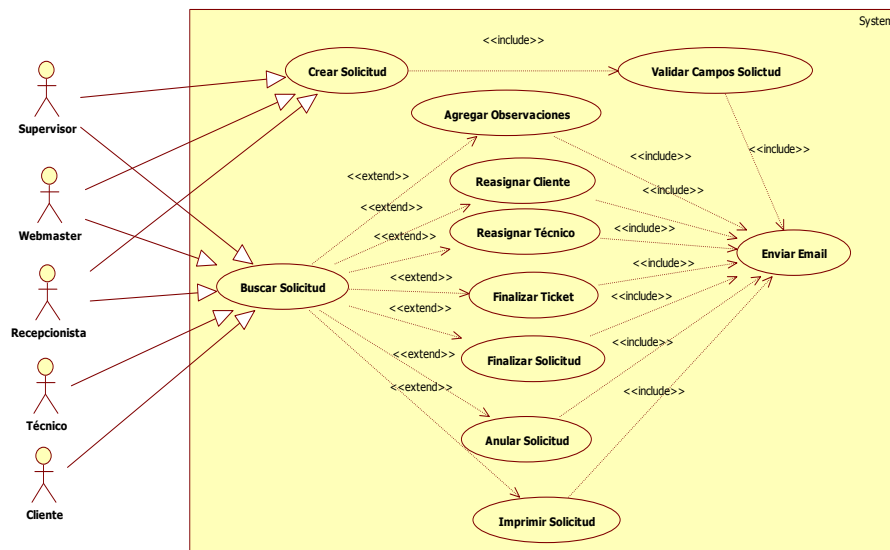


Figura 39. Caso de Uso Gestionar Solicitud

Cuadro 19***Detalle Caso de Uso Crear Solicitud***

CASO DE USO	
Identificación	1.19
Nombre	Crear Solicitud
Descripción	Permite que el actor pueda crear Solicitudes en el sistema de acuerdo a su necesidad.
Actores	Webmaster, Supervisor, Recepcionista
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado login en el sistema (Login 1.1).
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Solicitudes > Nueva Solicitud 3. El sistema despliega formulario de ingreso de solicitudes con los siguientes campos: <ul style="list-style-type: none"> • Cliente (Requerido) • Título Solicitud (Requerido) • Solicitud (Requerido) • Técnico Asignado • Prioridad (Requerido) • Tipo de Solicitud (Requerido) • Archivo Adjunto 4. El sistema valida datos de la solicitud (Valida Campos Solicitud 1.20). 5. El sistema guarda los datos ingresados.

Continúa →

6. El sistema envía un email de notificación al cliente, supervisor, gerente, recepción, técnico	
Flujo Alternativo	
4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.	
5.A El sistema valida si se asignó un técnico en el ingreso de la solicitud para establecer la solicitud en estado Asignado, caso contrario se establece en estado Pendiente.	
6.A. El sistema valida si se asignó un técnico en el ingreso de la solicitud para enviar un mail.	

Cuadro 20

Detalle Caso de Uso Validar Campos Solicitud

CASO DE USO	
Identificación	1.20
Nombre	Valida Campos Solicitud
Descripción	Permite validar los campos requeridos de la Solicitud.
Actores	Webmaster, Supervisor, Recepcionista
Frecuencia	Ocasional
Precondiciones	(Crear Solicitud 1.19)
Flujo Normal	<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • Cliente (Requerido) • Título Solicitud (Requerido) • Solicitud (Requerido)

Continúa →

<ul style="list-style-type: none"> • Prioridad (Requerido) • Tipo de Solicitud (Requerido) 	
Flujo Alternativo	
<p>1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.</p>	

Cuadro 21

Detalle Caso de Uso Envío Email

CASO DE USO	
Identificación	1.21
Nombre	Envío Email
Descripción	Permite enviar mails de las acciones específicas que se realiza sobre la solicitud.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	Ocasional
Precondiciones	<p>(Valida Campos Solicitud 1.20)</p> <p>(Agrega Observación 1.23)</p> <p>(Reasignar Cliente 1.24)</p> <p>(Reasignar Técnico 1.25)</p> <p>(Cerrar Solicitud 1.26)</p> <p>(Finalizar Solicitud 1.27)</p>

Continúa →

(Anular Solicitud 1.28)	
(Imprimir Solicitud 1.29)	
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe haber realizado alguna de las acciones mencionadas en las precondiciones. 2. El sistema recibirá la petición de envío de email. 3. Carga la plantilla de acuerdo a la acción realizada. 4. El sistema consulta los emails a los que debe enviar la notificación. 5. El sistema envía el email mediante el protocolo SMTP con una cuenta de correo pre configurado.
Flujo Alternativo	<ol style="list-style-type: none"> 4.A. Valida en la base de datos que haya direcciones de emails a donde tiene que enviar las notificaciones, de no haberlas el sistema no envía ningún mail y genera un log de envío. 5.A. Si el sistema valida que la dirección de email para envío no está configurada no envía ningún mail y escribe un log de envío.

Cuadro 22

Detalle Caso de Uso Buscar Solicitudes

CASO DE USO	
Identificación	1.22
Nombre	Buscar Solicitudes
Descripción	Permite que el actor pueda buscar solicitudes en el sistema.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	

Ocasional	
Precondiciones	
La Solicitud debe estar registrado en el sistema (Crear Solicitud 1.19).	
Flujo Normal	
<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Solicitudes > Mis Solicitudes o en Reportes > Solicitudes. 3. El sistema despliega formulario de búsqueda. 4. El actor puede utilizar los siguientes filtros de búsqueda: <ul style="list-style-type: none"> • Ticket • Cliente • Técnico • Empresa • Área • Prioridad • Estado • Tipo • Fecha de Registro • Fecha Solicitada • Fecha Asignada • Fecha Finalizada • Título • Todos los Registros 5. El sistema muestra una lista con los registros disponibles, se puede seleccionar el registro deseado para desplegar toda la información. 	
Flujo Alternativo	
5.A. Si no existen registros dentro de la tabla el sistema muestra un mensaje que no se encuentran registros.	

Cuadro 23***Detalle Caso de Uso Agregar Observaciones***

CASO DE USO	
Identificación	1.23
Nombre	Agregar Observaciones
Descripción	<p>Permite que el actor pueda agregar n observaciones a la solicitud.</p>
Actores	<p>Webmaster, Supervisor, Recepcionista, Técnico, Cliente</p>
Frecuencia	Ocasional
Precondiciones	<p>El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Observaciones] en la ventana de la solicitud. 4. El sistema carga el formulario para agregar la observación de la solicitud seleccionada con los siguientes campos: <ul style="list-style-type: none"> • Observación (Requerido) 5. El actor digita la observación requerida y presiona guardar. 6. El sistema guarda los datos ingresados.
Flujo Alternativo	

5.A. El sistema valida que el campo no se encuentra vacío, si lo está, envía una alerta indicando que el campo no puede ser vacío.

6.A. El sistema guarda la observación con la fecha, hora y usuario que la realizo, después despliega toda la ventana de información de la solicitud y muestra la información guardada.

6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)

Cuadro 24

Detalle Caso de Uso reasignar Cliente

CASO DE USO	
Identificación	1.24
Nombre	Reasignar Cliente
Descripción	<p>Permite que el actor en caso de error de ingreso pueda ser asignada la solicitud a otro cliente como solicitante.</p>
Actores	<p>Webmaster, Supervisor, Recepcionista, Técnico, Cliente</p>
Frecuencia	<p>Ocasional</p>
Precondiciones	<p>El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Asignar Cliente] en la ventana de la solicitud. 4. El sistema carga el formulario para asignar a un nuevo cliente la solicitud seleccionada con los

Continúa →

siguientes campos:

- Cliente Solicita (Requerido)

5. El actor selecciona el cliente y presiona Asignar Solicitante.

6. El sistema guarda los datos ingresados.

Flujo Alternativo

5.A. El sistema valida que el campo no se encuentre sin opción, si lo está, envía una alerta indicando que debe seleccionar una opción.

6.A. El sistema guarda la acción con la fecha, hora y usuario que la realizo, después despliega toda la ventana de información de la solicitud y muestra la información guardada.

6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)

Cuadro 25

Detalle Caso de Uso

CASO DE USO	
Identificación	1.25
Nombre	Reasignar Técnico
Descripción	Permite que el actor pueda asignar a otro técnico la solicitud.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).

Continúa →

Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Asignar Técnico] en la ventana de la solicitud. 4. El sistema carga el formulario para asignar a un nuevo Técnico la solicitud seleccionada con los siguientes campos: <ul style="list-style-type: none"> • Técnico Asignado (Requerido) 5. El actor selecciona el Técnico y presiona Asignar Técnico. 6. El sistema guarda los datos ingresados.
Flujo Alternativo	<ol style="list-style-type: none"> 5.A. El sistema valida que el campo no se encuentre sin opción, si lo está, envía una alerta indicando que debe seleccionar una opción. 6.A. El sistema guarda la acción con la fecha, hora y usuario que la realizo, después despliega toda la ventana de información de la solicitud y muestra la información guardada. 6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)

Cuadro 26

Detalle Caso de Uso Finalizar Ticket

CASO DE USO	
Identificación	1.26
Nombre	Finalizar Ticket
Descripción	Permite que el actor pueda finalizar un ticket de atención.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente

Continúa →

Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).	
Flujo Normal	
1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22).	
2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante.	
3. El actor selecciona la opción [Finalizar Ticket] en la ventana de la solicitud.	
4. El sistema carga el formulario para finalizar el ticket de atención en la solicitud seleccionada con los siguientes campos:	
<ul style="list-style-type: none"> Observación (Requerido) 	
5. El actor selecciona ingresa la observación y presiona Finalizar Ticket.	
6. El sistema guarda los datos ingresados.	
Flujo Alternativo	
5.A. El sistema valida que el campo no se encuentre vacío, si lo está, envía una alerta indicando que debe contener información.	
6.A. El sistema guarda la acción con la fecha, hora y usuario que la realizo, después despliega toda la ventana de información de la solicitud y muestra la información guardada.	
6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)	

Cuadro 27

Detalle de Caso de Uso Finalizar Solicitud

CASO DE USO	
Identificación	1.27
Nombre	Finalizar Solicitud

Continúa →

Descripción	
	Permite que el actor pueda finalizar una solicitud completamente.
Actores	
	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	
	Ocasional
Precondiciones	
	El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).
Flujo Normal	
	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Finalizar Solicitud] en la ventana de la solicitud. 4. El sistema carga el formulario para finalizar el ticket de atención en la solicitud seleccionada con los siguientes campos: <ul style="list-style-type: none"> • Observación (Requerido) 5. El actor selecciona ingresa la observación y presiona Finalizar Solicitud. 6. El sistema guarda los datos ingresados.
Flujo Alternativo	
	<ol style="list-style-type: none"> 5.A. El sistema valida que el campo no se encuentre vacío, si lo está, envía una alerta indicando que debe contener información. 6.A. El sistema guarda la acción con la fecha, hora y usuario que la realizo, después despliega toda la ventana de información de la solicitud y muestra la información guardada. 6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)

Cuadro 28***Detalle Caso de Uso Anular Solicitud***

CASO DE USO	
Identificación	1.28
Nombre	Anular Solicitud
Descripción	Permite que el actor pueda anular una solicitud por algún error motivo en especial.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado Buscar Solicitudes (Buscar Solicitudes 1.22).
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Anular Solicitud] en la ventana de la solicitud. 4. El sistema carga el formulario para finalizar el ticket de atención en la solicitud seleccionada con los siguientes campos: <ul style="list-style-type: none"> • Observación (Requerido) 5. El actor selecciona ingresa la observación y presiona Anular Solicitud. 6. El sistema guarda los datos ingresados.
Flujo Alternativo	

Continúa →

5.A. El sistema valida que el campo no se encuentre vacío, si lo está, envía una alerta indicando que debe contener información.

6.A. El sistema guarda la acción con la fecha, hora y usuario que la realizó, después despliega toda la ventana de información de la solicitud y muestra la información guardada.

6.B. El sistema envía la acción generada a los involucrados en la solicitud (Envío Email 1.21)

Cuadro 29

Detalle de Caso de Uso Imprimir Solicitud

CASO DE USO	
Identificación	1.29
Nombre	Imprimir Solicitud
Descripción	Permite que el actor pueda imprimir una solicitud en cualquier etapa del proceso de solución.
Actores	Webmaster, Supervisor, Recepcionista, Técnico, Cliente
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado Imprimir Solicitudes (Buscar Solicitudes 1.22).
Flujo Normal	<ol style="list-style-type: none"> 1. Realizada la Búsqueda de solicitudes (Buscar Solicitudes 1.22). 2. Seleccionar la solicitud con la que se va a trabajar en la lista desplegada, esto muestra la información de la solicitud en una ventana flotante. 3. El actor selecciona la opción [Imprimir Solicitud] en la ventana de la solicitud. 4. El sistema automáticamente envía al administrador de impresiones del navegador.

Continúa →

5. El sistema cierra automáticamente la ventana de información.	
Flujo Alternativo	
4.A. la impresión podrá realizarse siempre que haya al menos una impresora instalada en el computador.	

3.3.1.8 Caso de Uso Gestionar Técnicos

A continuación se detalla el caso de uso Gestionar Tablas con la Figura 40 y la interacción con el sistema en el Cuadro 30, 31, 32, 33.

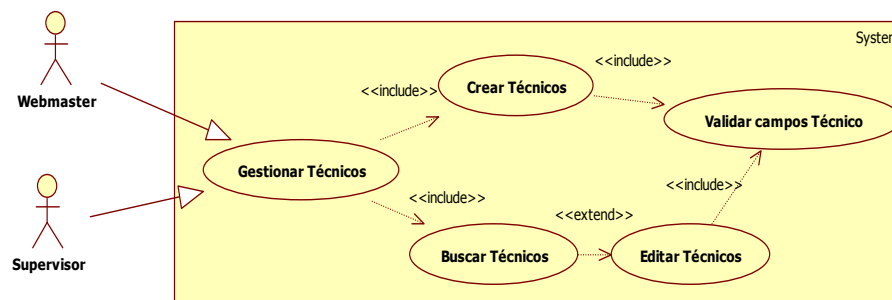


Figura 40. Caso de Uso Gestionar Técnicos

Cuadro 30

Detalle Caso de Uso Crear Técnicos

CASO DE USO	
Identificación	1.30
Nombre	Crear Técnicos
Descripción	Permite que el actor pueda crear nuevos Técnicos en el sistema de acuerdo a su necesidad.
Actores	

Continúa →

Webmaster, Supervisor	
Frecuencia	
Ocasional	
Precondiciones	
El actor debe haber realizado login en el sistema (Login 1.1).	
Flujo Normal	
<p>1. El actor debe iniciar sesión (Login 1.1).</p> <p>2. Selecciona el menú Administración > Técnicos</p> <p>3. El sistema despliega formulario de ingreso de clientes con los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) • Perfil (Requerido, Bloqueado, Seleccionado Técnico) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido, Bloqueado) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) <p>4. El sistema valida datos de Técnico (Valida Campos Técnico 1.10).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	
<p>4.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.</p> <p>5.A. El sistema valida que el usuario no exista en el sistema, si existe, envía un mensaje de alerta al actor y no guarda los datos hasta su rectificación.</p>	

Cuadro 31***Detalle Caso de Uso Buscar Técnicos***

CASO DE USO	
Identificación	1.31
Nombre	Buscar Técnicos
Descripción	Permite que el actor pueda buscar un Técnico en el sistema.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	Técnico debe estar registrado en el sistema (Crear Técnicos 1.7).
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. Selecciona el menú Administración > Técnicos. 3. El sistema despliega formulario de búsqueda. 4. El actor puede utilizar los siguientes filtros de búsqueda: <ul style="list-style-type: none"> • Id • Empresa • Usuario • Nombre • Apellido • Email • Estado • Departamento • Teléfono Móvil • Teléfono Fijo

Continúa →

<ul style="list-style-type: none"> • Acceso • Fecha de Registro • Fecha que caduca • Todos los Registros <p>5. El sistema muestra una lista con los registros disponibles, se puede seleccionar el registro deseado para desplegar toda la información.</p>	
Flujo Alternativo	
5.A. Si no existen registros dentro de la tabla el sistema muestra un mensaje que no se encuentran registros.	

Cuadro 32

Detalle Caso de Uso Editar Técnicos

CASO DE USO	
Identificación	1.32
Nombre	Editar Técnicos
Descripción	Permite que el actor pueda editar Técnicos.
Actores	Webmaster, Supervisor
Frecuencia	Ocasional
Precondiciones	El actor debe haber realizado Buscar Técnicos (Buscar Técnicos 1.31).
Flujo Normal	

Continúa →

<p>1. Realizada la Búsqueda de Técnicos (Buscar Técnicos 1.31).</p> <p>2. Seleccionar el Técnico a editar en la lista desplegada, esto muestra toda la información del Técnico.</p> <p>3. Seleccionar modificar en la pantalla de información.</p> <p>4. El sistema carga el formulario con los datos del Técnico seleccionado en los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Perfil (Requerido, Bloqueado, Seleccionado Técnico) • Fecha de Ingreso (Requerido) • Fecha de Caducidad • Email • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido Bloqueado) • Área / Departamento (Requerido) • Teléfono Móvil • Teléfono Fijo • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido) <p>5. El actor edita los campos requeridos.</p> <p>6. El sistema valida datos del Técnico a editar (Valida Campos Técnico 1.10).</p> <p>7. El sistema guarda los datos ingresados.</p>	
Flujo Alternativo	
	6.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos envía una alerta indicando los campos que tienen que ser llenados.

Cuadro 33

Detalle Caso de Uso Valida Campos Técnico

CASO DE USO	
Identificación	1.33

Nombre	Valida Campos Técnico
Descripción	<p>Permite validar los campos requeridos del Técnico.</p>
Actores	<p>Webmaster, Supervisor</p>
Frecuencia	<p>Ocasional</p>
Precondiciones	<p>(Crear Técnicos 1.30)</p> <p>(Editar Técnicos 1.32)</p>
Flujo Normal	<p>1. El sistema Valida que los siguientes campos estén ingresados en formulario:</p> <ul style="list-style-type: none"> • Nombre (Requerido) • Apellido (Requerido) • Usuario (Requerido) (Crear Técnicos 1.30) • Perfil (Requerido, Bloqueado, Seleccionado Cliente) (Edtar Técnicos 1.32) • Fecha de Ingreso (Requerido) • Usuario Activo (Requerido, SI/NO) • Empresa (Requerido, Bloqueado) • Área / Departamento (Requerido) • Acceso Web (Requerido, SI/NO) • Tiempo de Sesión (Requerido)
Flujo Alternativo	<p>1.A. Valida que todos los campos requeridos sean ingresados en el formulario, si no están todos los campos o si no corresponde al tipo de datos envía una alerta indicando los campos que tienen que ser llenados o no corresponden al tipo de datos permitido.</p>

3.3.1.9 Caso de Uso Calificar Atención

A continuación se detalla el caso de uso Gestionar Tablas con la Figura 41 y la interacción con el sistema en el Cuadro 34.

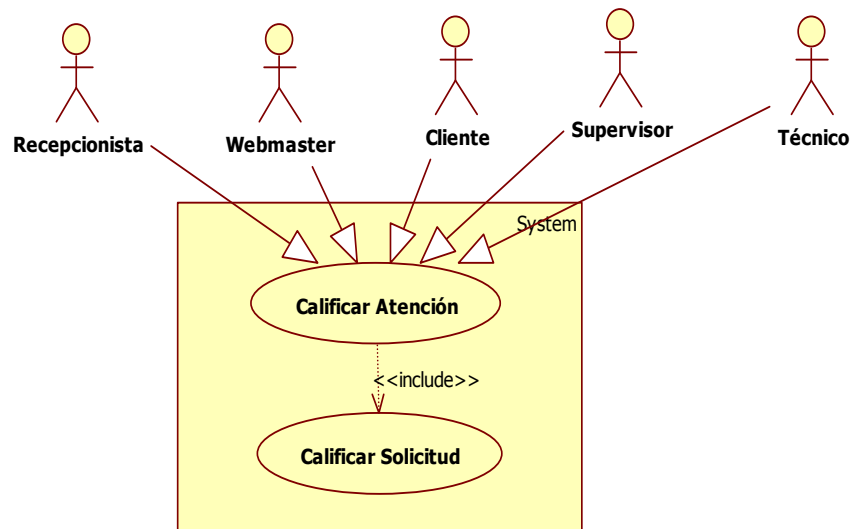


Figura 41. Caso de Uso Calificar Atención

Cuadro 34

Detalle Caso de Uso Calificar Solicitud

CASO DE USO	
Identificación	1.34
Nombre	Califica Solicitud
Descripción	Permite validar si la solicitud está correctamente finalizada.
Actores	Webmaster, Supervisor, Recepcionista, Cliente, Técnico

Frecuencia	
Ocasional	
Precondiciones	
(Login 1.1)	
Flujo Normal	
1. El actor debe iniciar sesión (Login 1.1).	
2. El sistema valida que solicitudes están con el estado finalizado y las despliega en pantalla con la opción de calificación.	
3. El actor seleccionar la calificación deseada por solicitud antes de continuar en el sistema.	
Flujo Alternativo	
3.A. El sistema no permite ingresar al sistema mientras haya solicitudes pendientes de calificación.	

3.3.1.10 Caso de Uso Generar Reportes

A continuación se detalla el caso de uso Gestionar Tablas con la Figura 42 y la interacción con el sistema en el Cuadro 35.

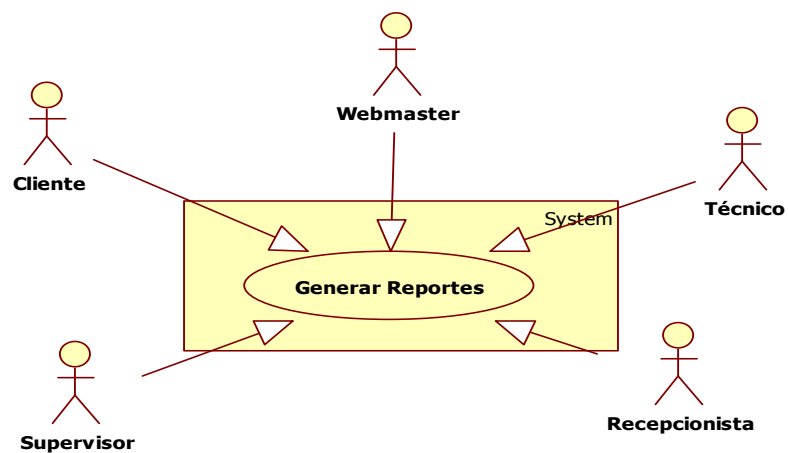


Figura 42. Caso de Uso Generar Reportes**Cuadro 35*****Detalle Caso de Uso Generar Reportes***

CASO DE USO	
Identificación	1.35
Nombre	Generar Reportes
Descripción	Permite al actor ingresar a los distintos reportes de acuerdo a los accesos definidos en su perfil y desplegarlos.
Actores	Webmaster, Supervisor, Recepcionista, Cliente, Técnico
Frecuencia	Ocasional
Precondiciones	(Login 1.1)
Flujo Normal	<ol style="list-style-type: none"> 1. El actor debe iniciar sesión (Login 1.1). 2. El actor debe seleccionar en el menú Reportes 3. El actor selecciona el reporte deseado de acuerdo a los accesos en su perfil. 3. El sistema despliega la información deseada por el usuario.
Flujo Alternativo	3.A. Si no se encuentra la información deseada el sistema muestra un mensaje en pantalla que no existe información.

3.3.2 Diagramas de Secuencia

A continuación se presenta los diagramas de secuencia correspondiente a cada caso de uso.

3.3.2.1 Diagramas de Secuencia Login

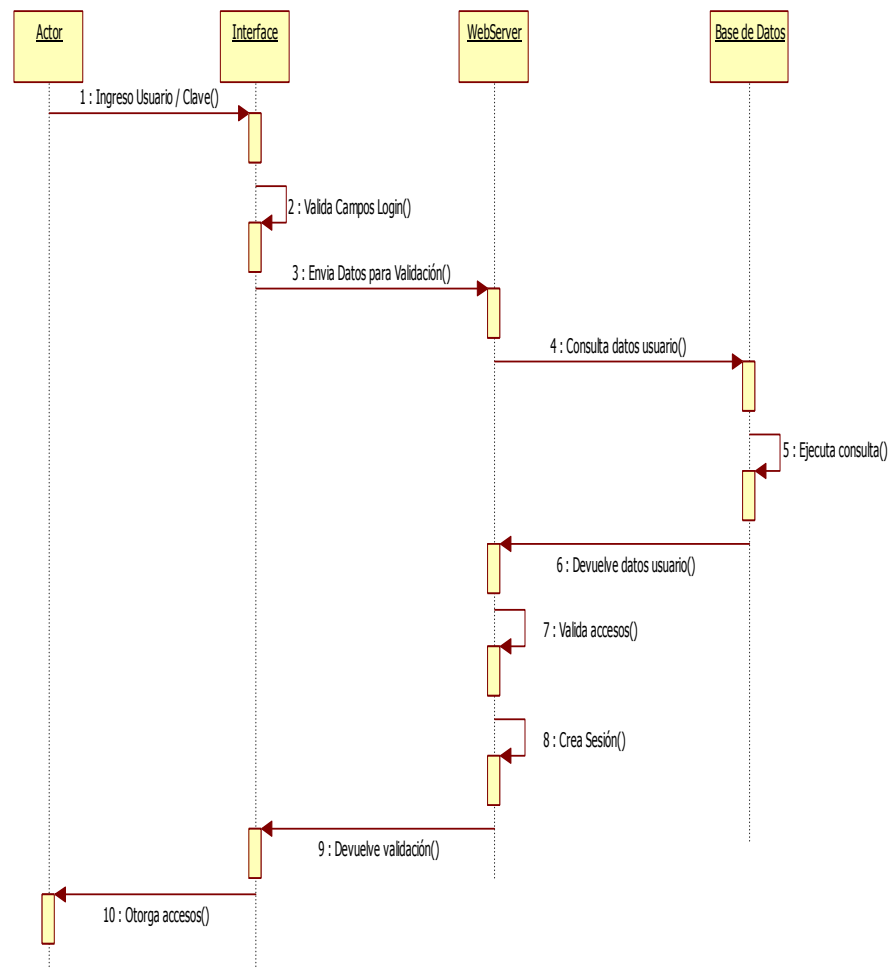


Figura 43. Diagrama Secuencia Login

3.3.2.2 Diagramas de Secuencia Cambiar Clave

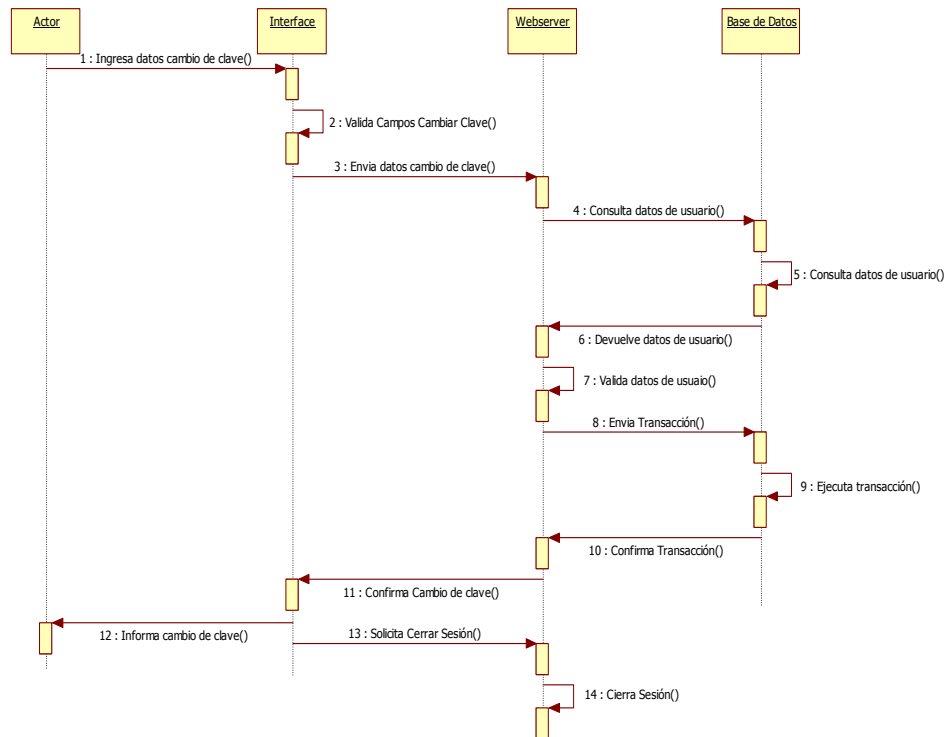


Figura 44. Diagrama de Secuencia Cambio de Clave

3.3.2.3 Diagramas de Secuencia Gestionar Tablas

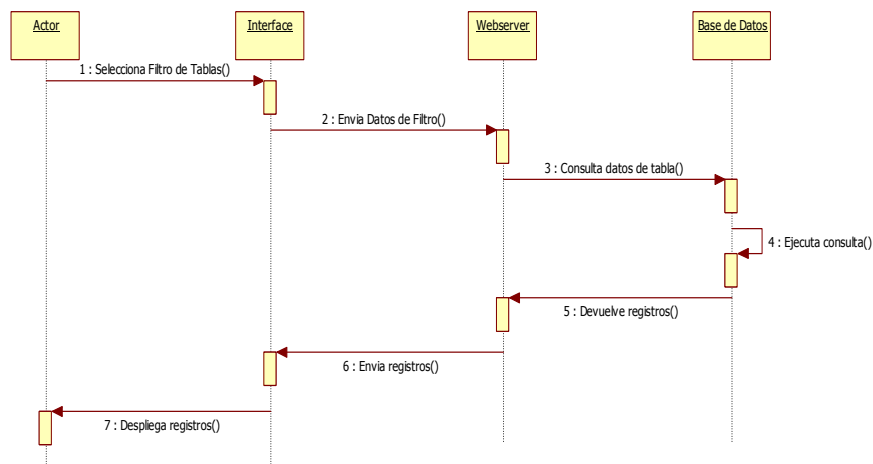


Figura 45. Diagrama de Secuencia Buscar Tablas

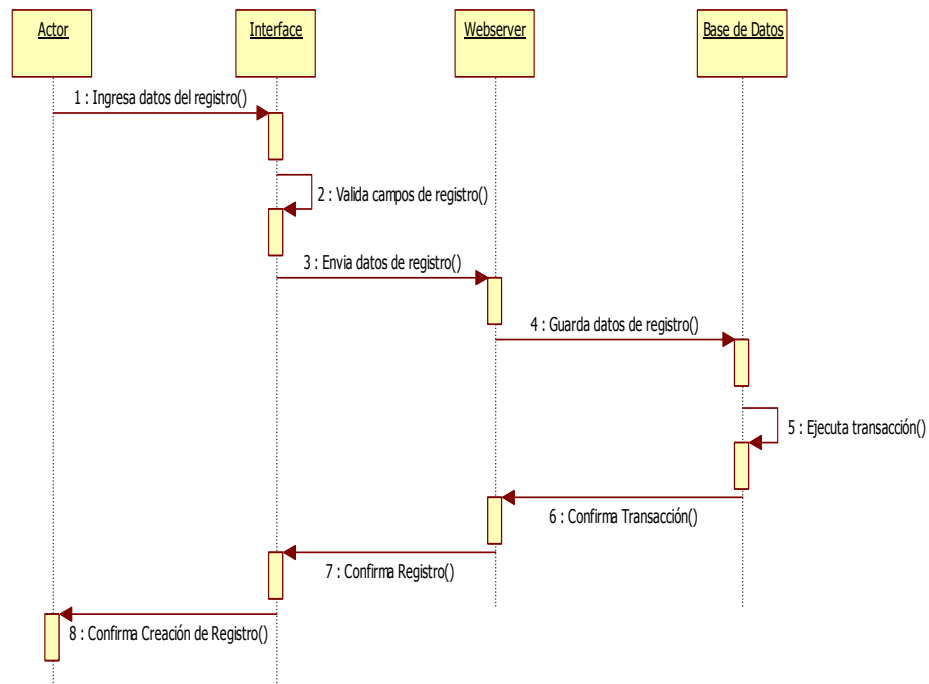


Figura 46. Diagrama de Secuencia Crear Registros

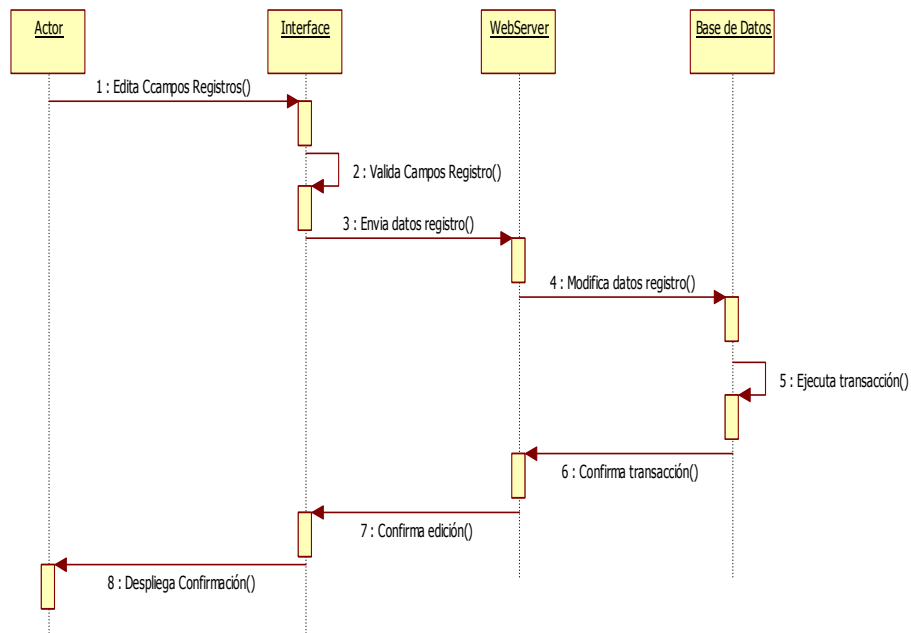


Figura 47. Diagrama de Secuencia Editar Registros

3.3.2.4 Diagramas de Secuencia Gestionar Clientes

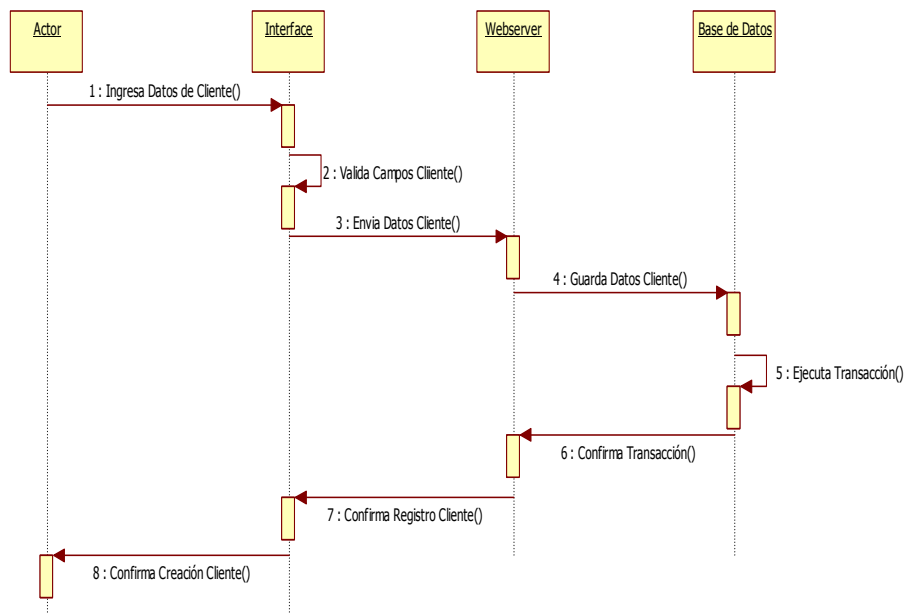


Figura 48. Diagrama de Secuencia Crear Clientes

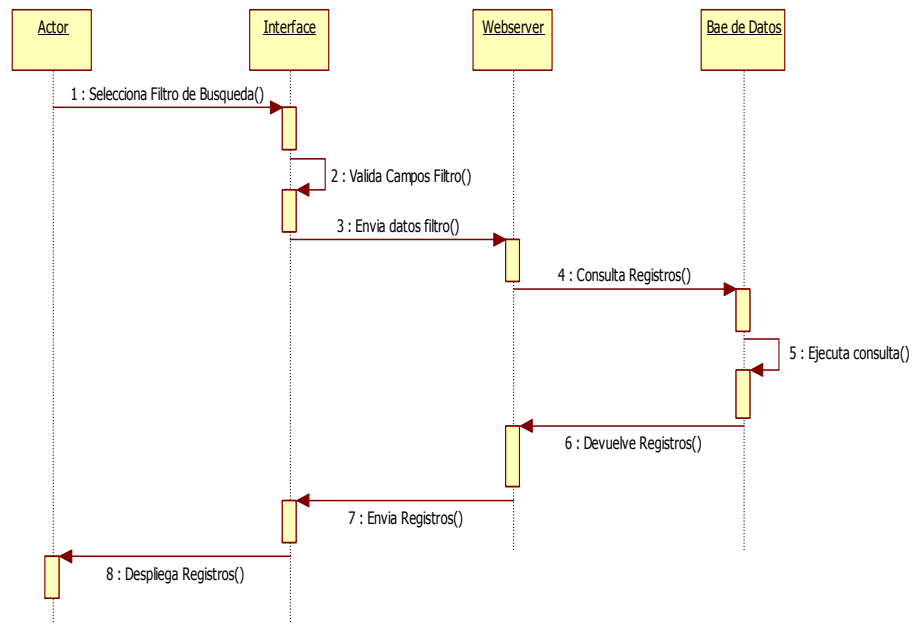


Figura 49. Diagrama de Secuencia Buscar Clientes

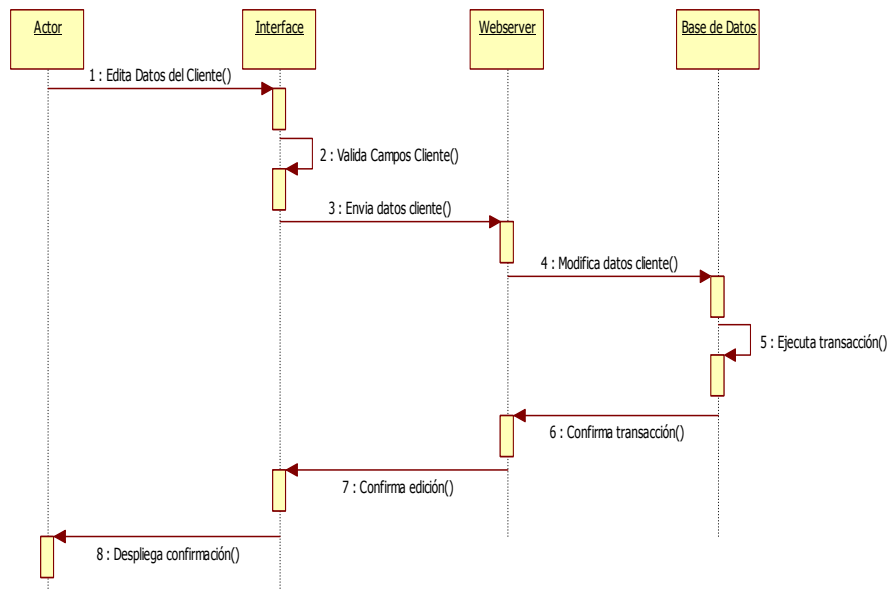


Figura 50. Diagrama de Secuencia Editar Clientes

3.3.2.5 Diagramas de Secuencia Gestionar Usuario

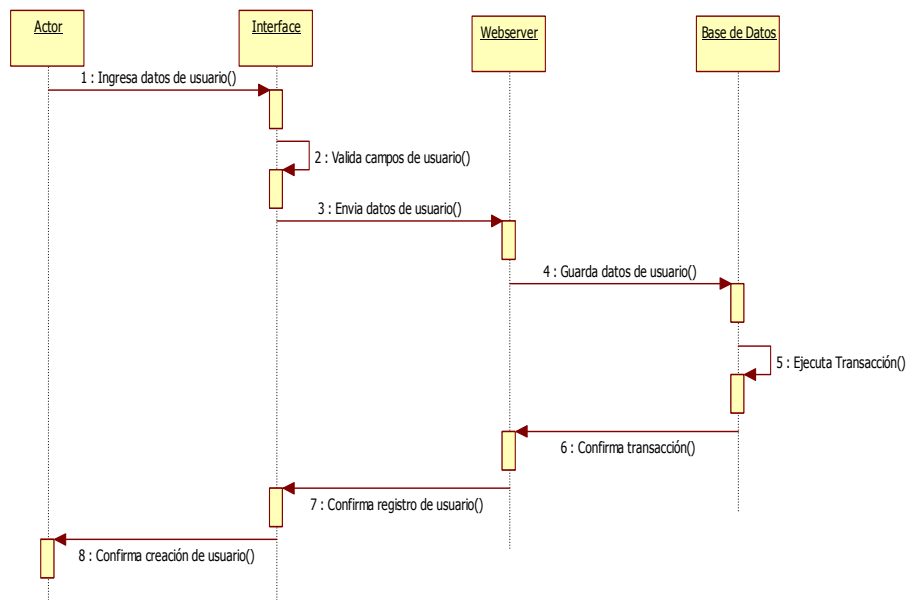


Figura 51. Diagrama de Secuencia Crear Usuarios

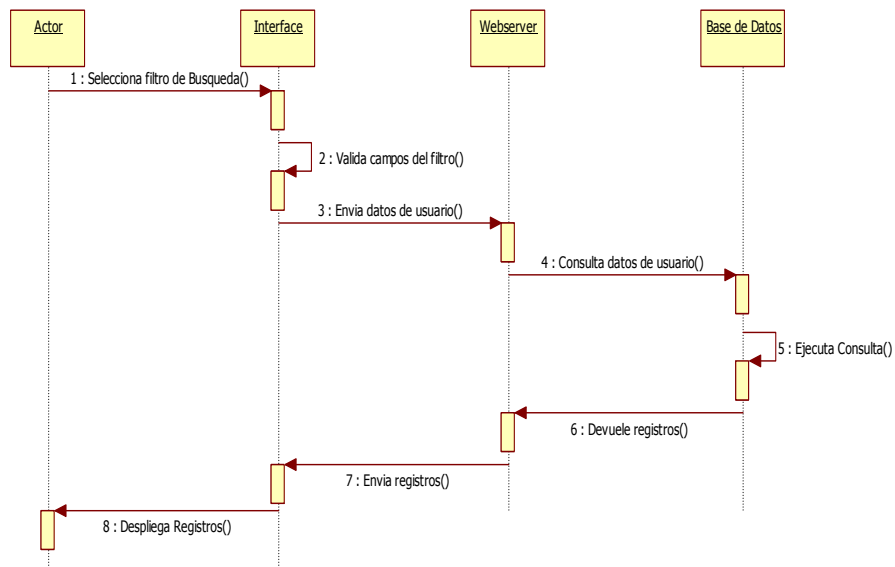


Figura 52. Diagrama de Secuencia Buscar Usuario

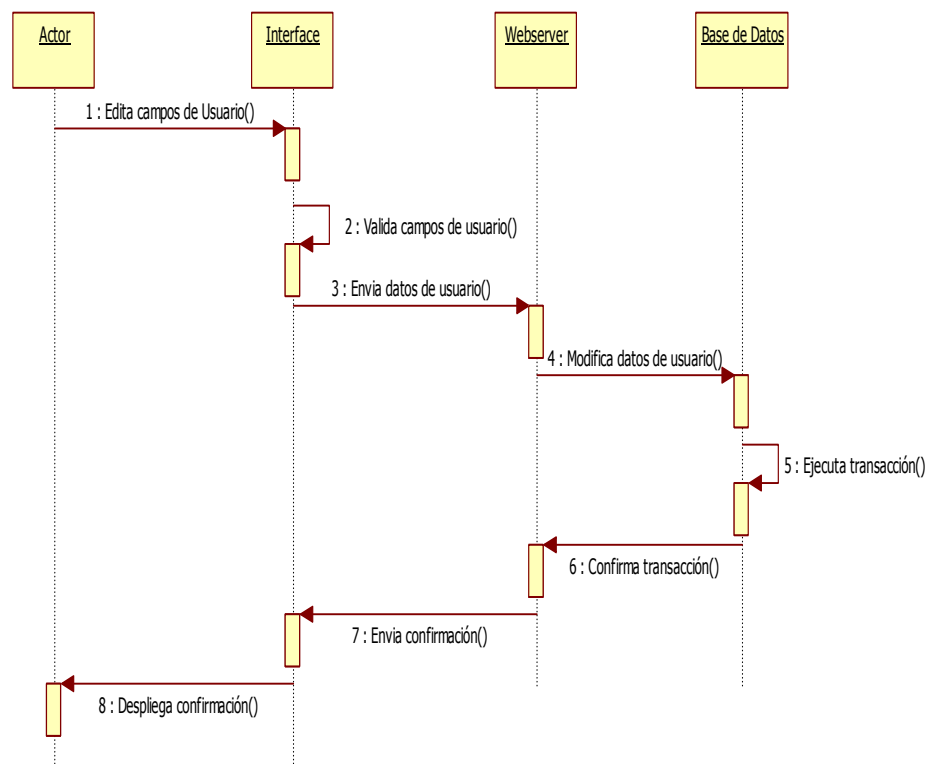


Figura 53. Diagrama de Secuencia Editar Usuario

3.3.2.6 Diagramas de Secuencia Gestionar Empresas

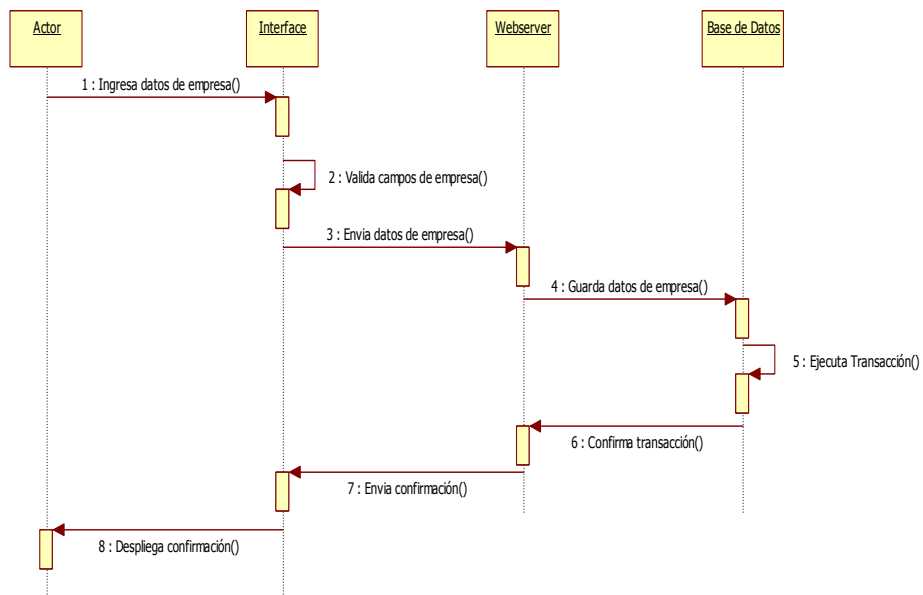


Figura 54. Diagrama de Secuencia Crear Empresa

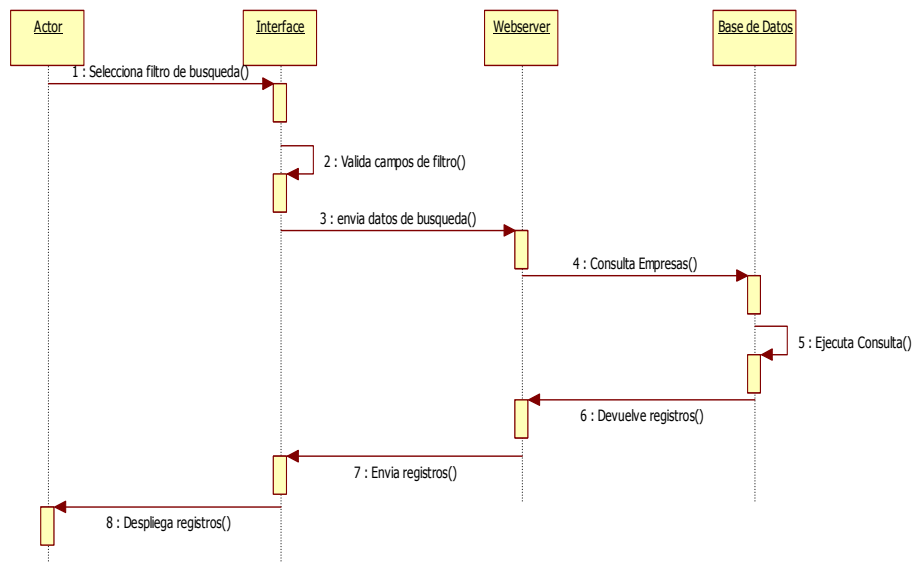


Figura 55. Diagrama de Secuencia Buscar Empresa

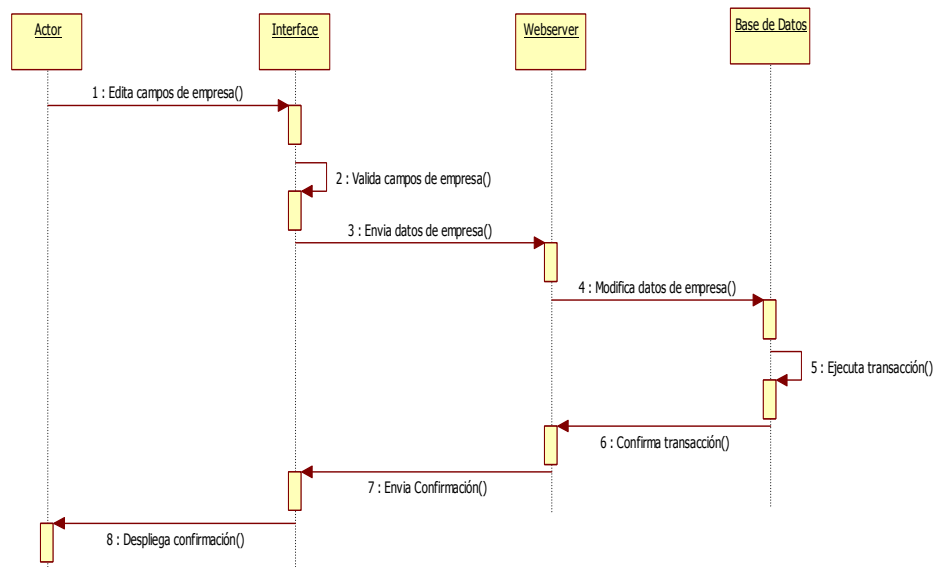


Figura 56. Diagrama de Secuencia Editar Empresa

3.3.2.7 Diagramas de Secuencia Gestionar Solicitudes

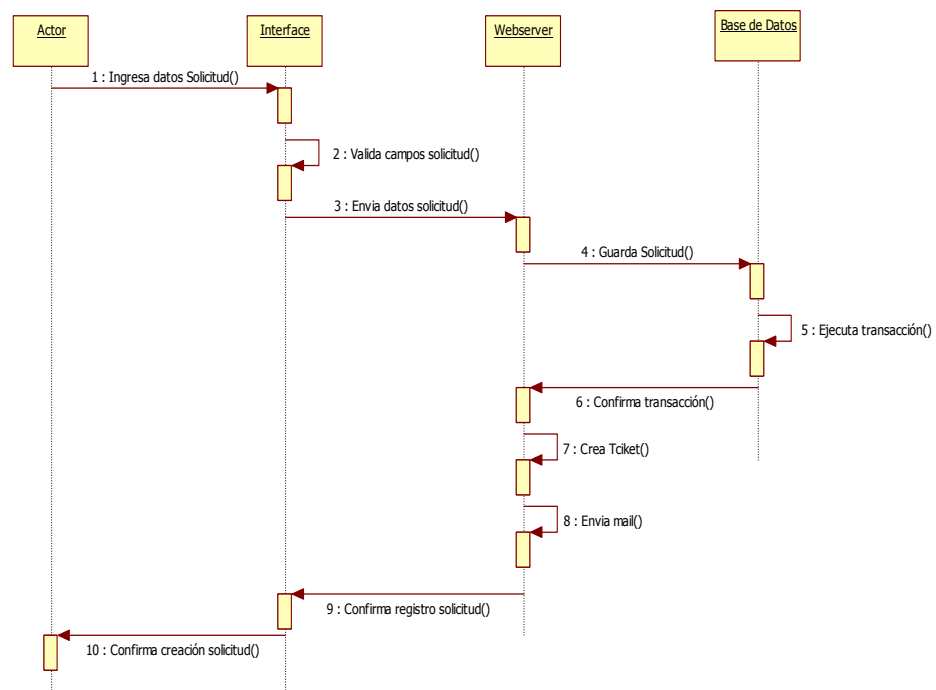


Figura 57. Diagrama de Secuencia Crear Solicitudes

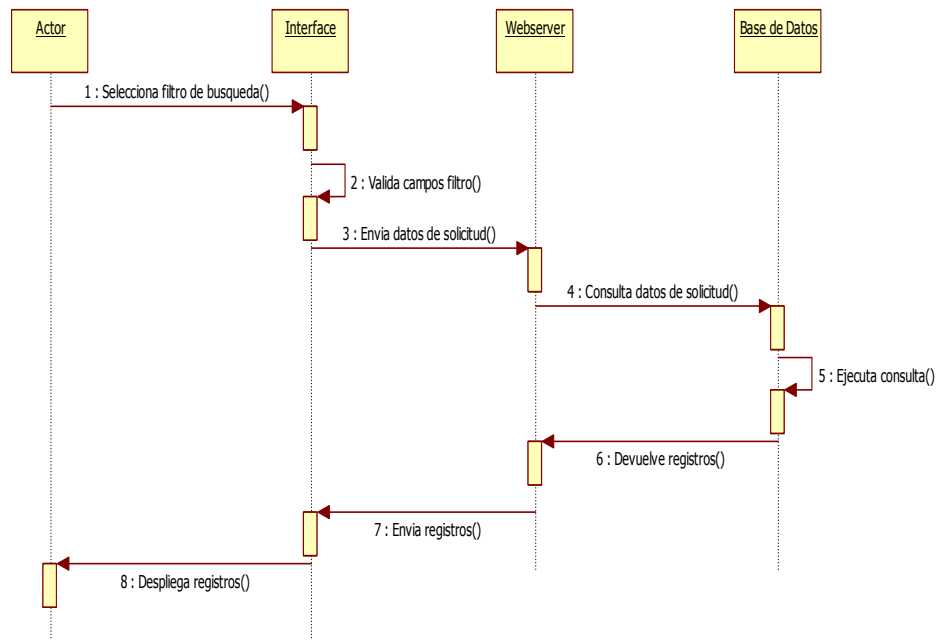


Figura 58. Diagrama de Secuencia Buscar Solicitudes

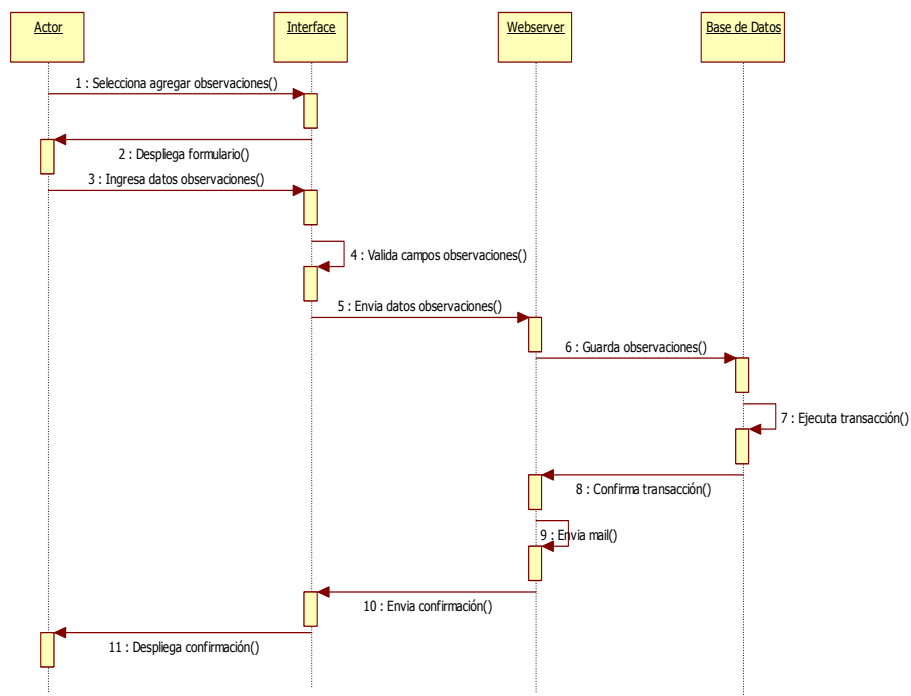


Figura 59. Diagrama de Secuencia Agregar Observaciones

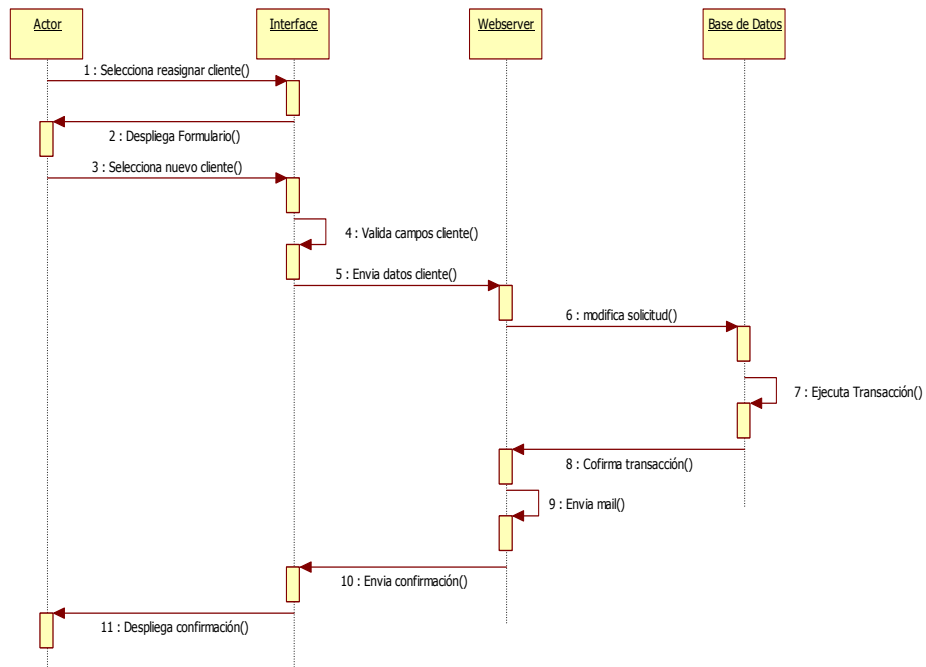


Figura 60. Diagrama de Secuencia Reasignar Cliente

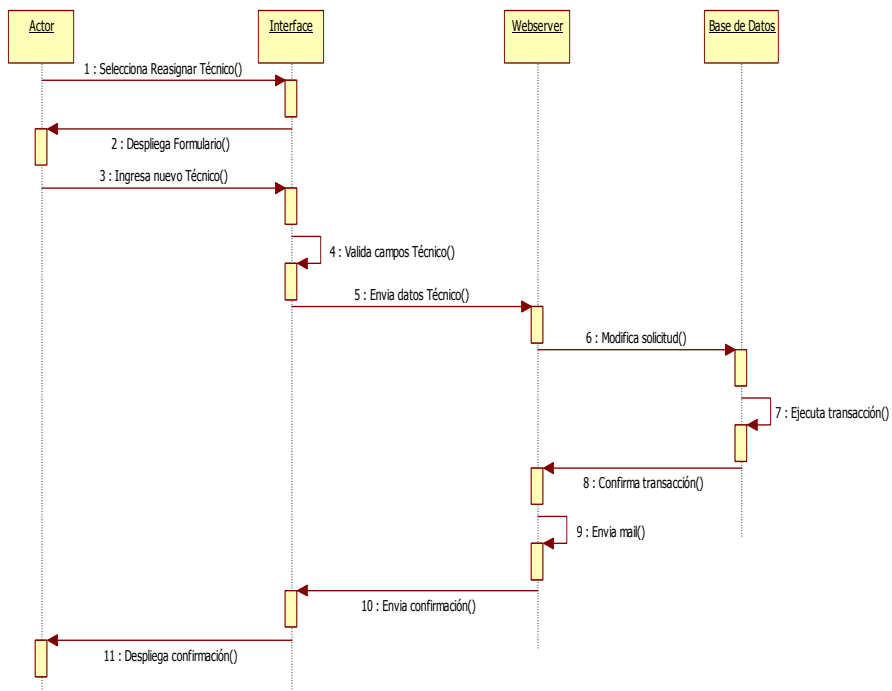


Figura 61. Diagrama de Secuencia Reasignar Técnico

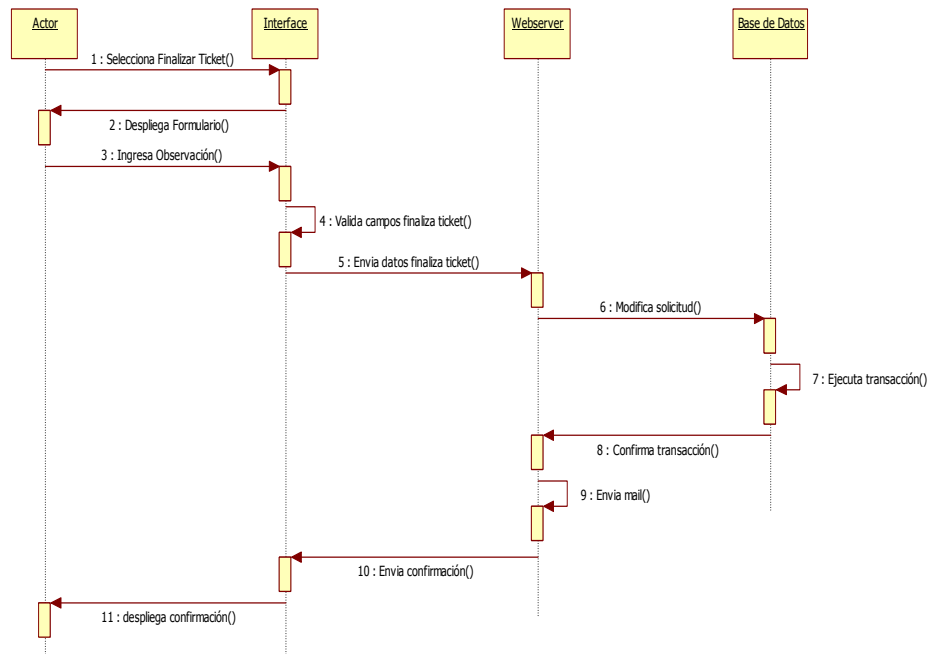


Figura 62. Diagrama de Secuencia Finalizar Ticket

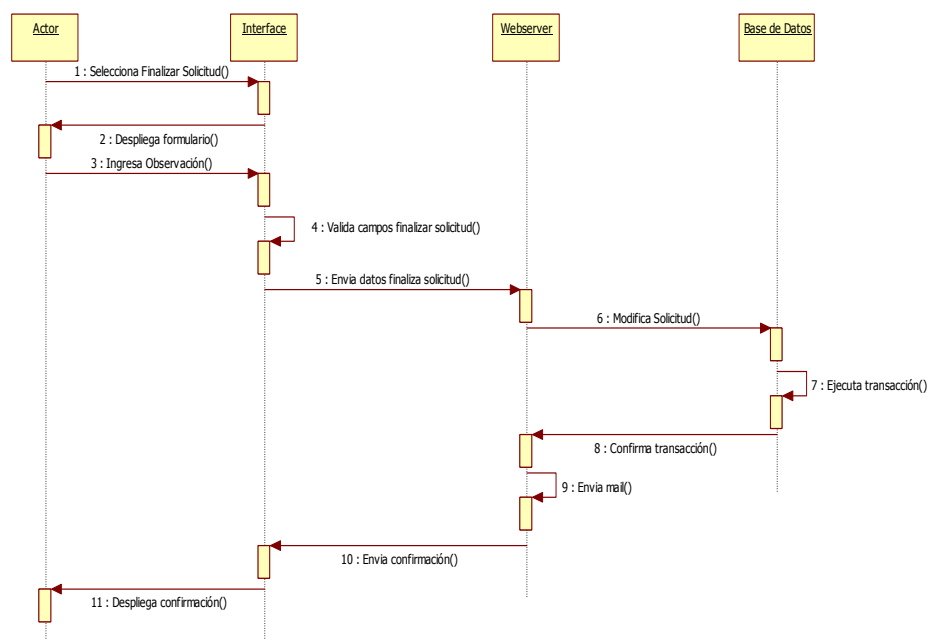


Figura 63. Diagrama de Secuencia Finalizar Solicitud

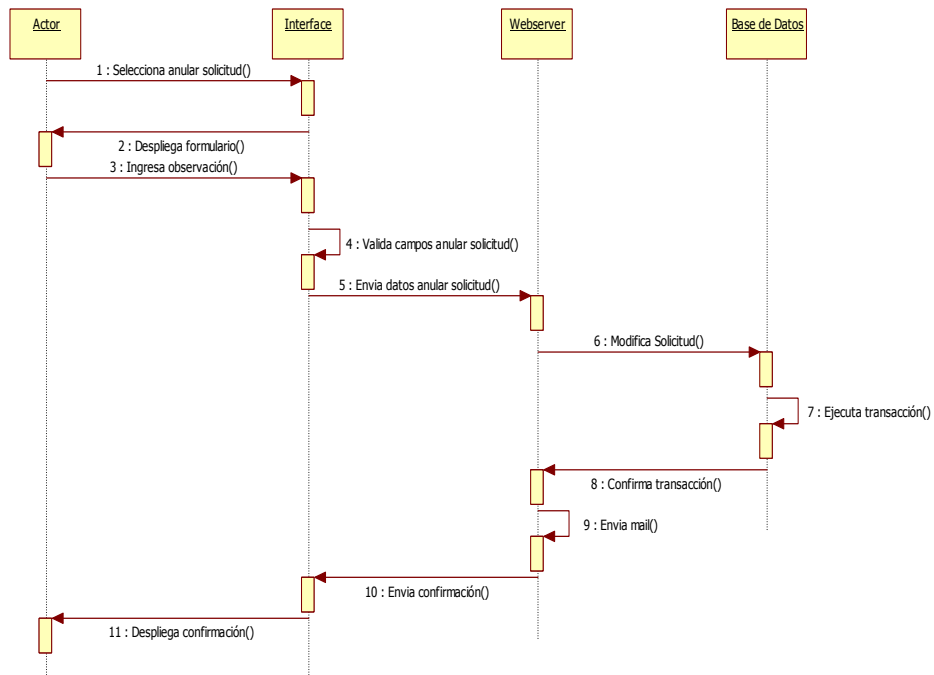


Figura 64. Diagrama de Secuencia Anular Solicitud

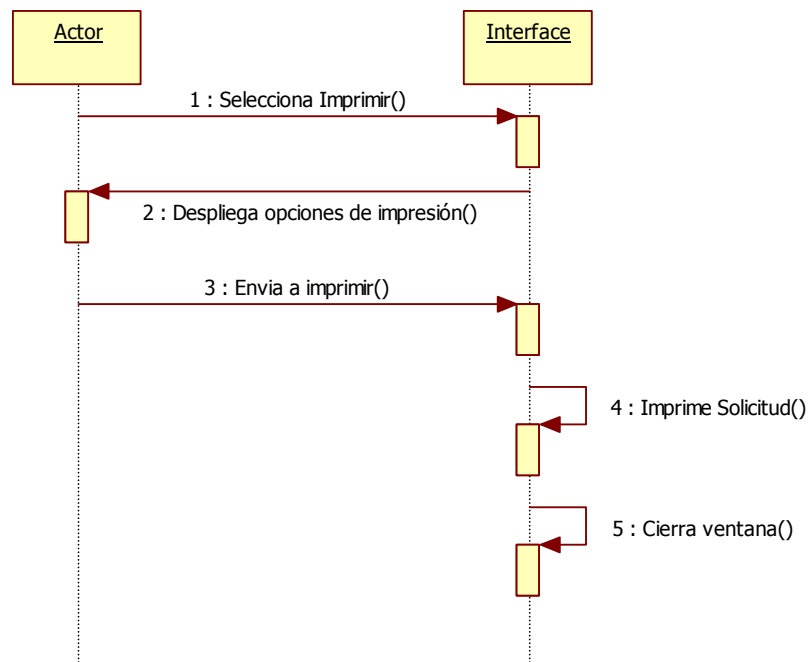


Figura 65. Diagrama de Secuencia Imprimir Solicitud

3.3.2.8 Diagramas de Secuencia Gestionar Técnicos

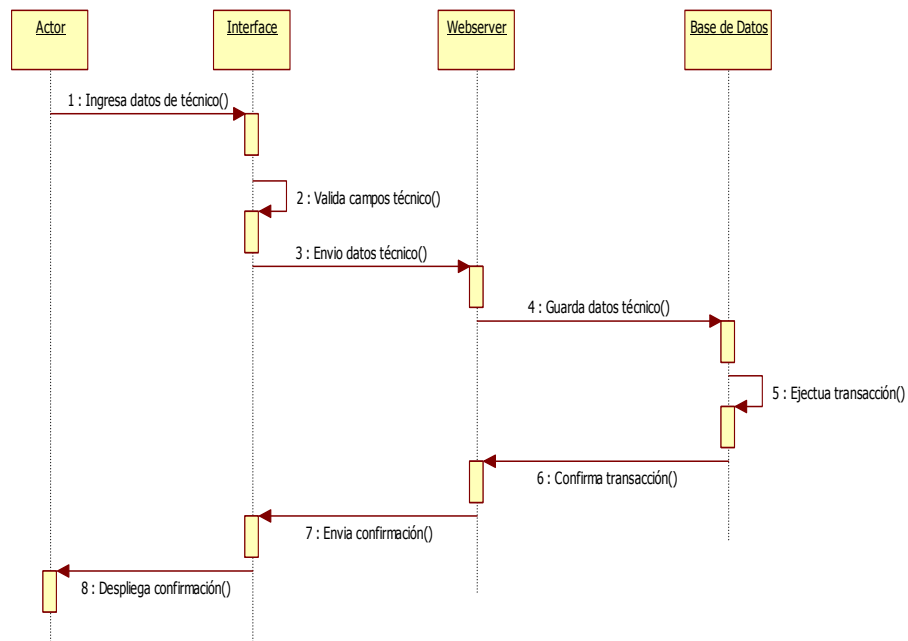


Figura 66. Diagrama de Secuencia Crear Técnicos

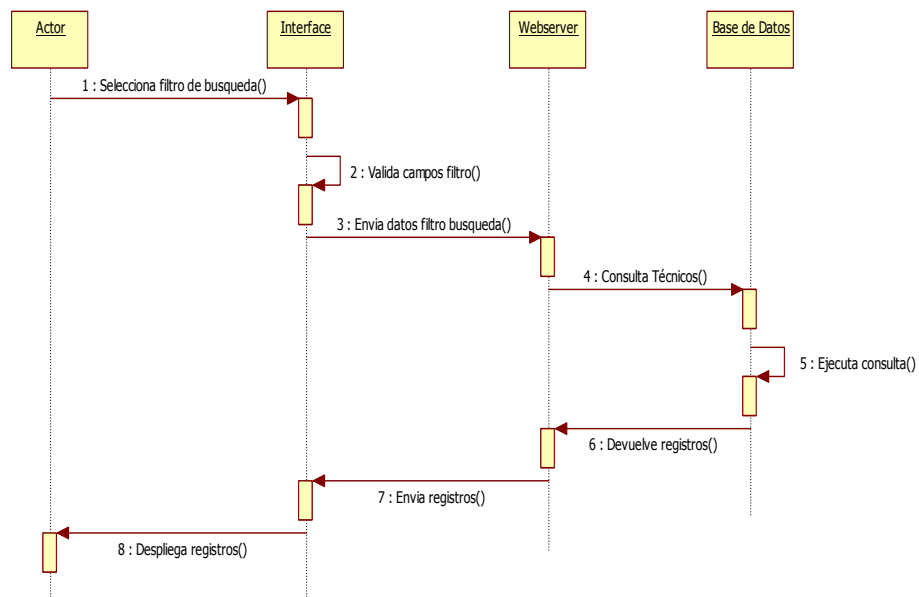


Figura 67. Diagrama de Secuencia Buscar Técnicos

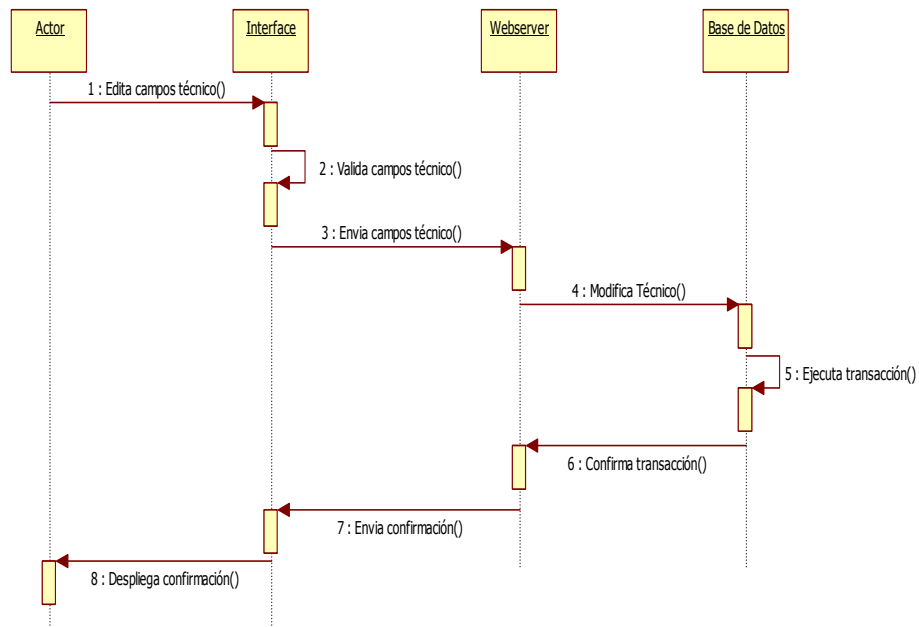


Figura 68. Diagrama de Secuencia Editar Técnicos

3.3.2.9 Diagramas de Secuencia Calificar Atención

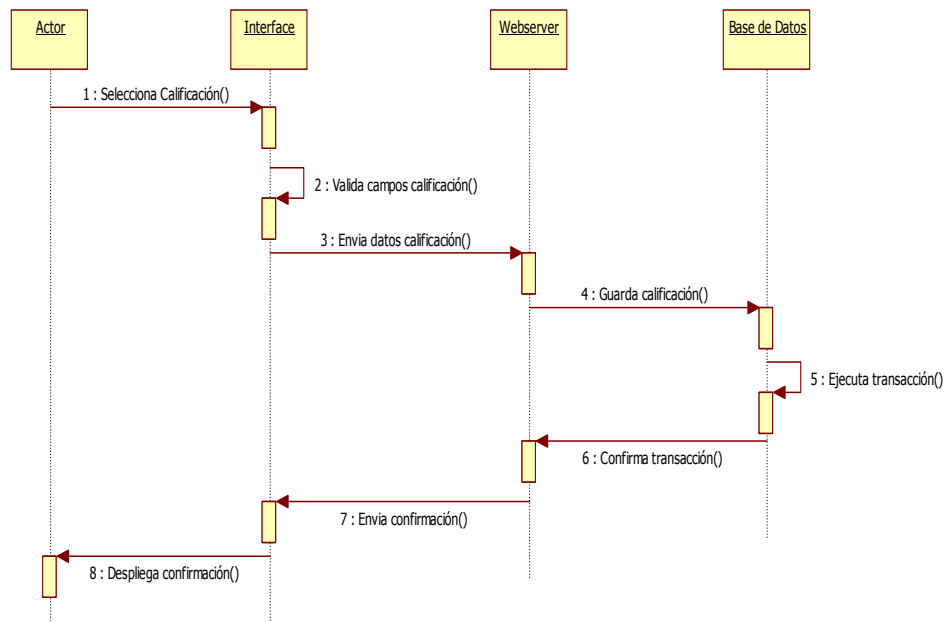


Figura 69. Diagrama de Secuencia Calificar Solicitud

3.3.2.10 Diagramas de Secuencia Generar Reportes

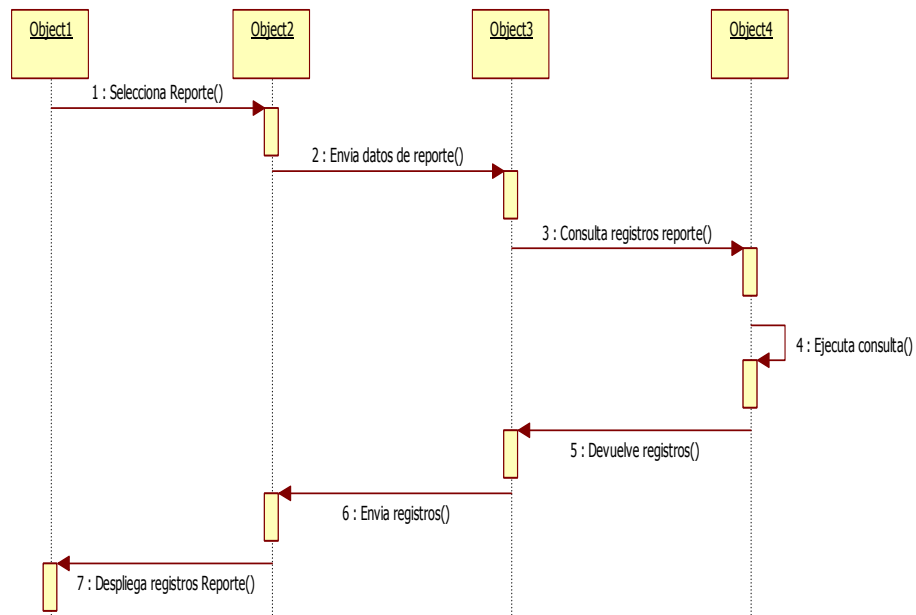


Figura 70. Diagrama de Secuencia Generar Reportes

3.3.3 Diagramas de Colaboración

3.3.3.1 Diagrama de Colaboración Login

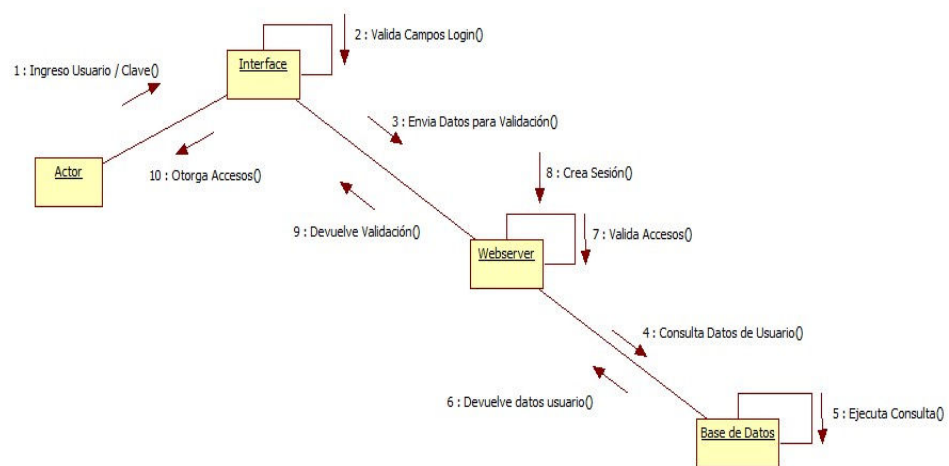


Figura 71. Diagrama de Colaboración Login

3.3.3.2 Diagrama de Colaboración Cambiar Clave

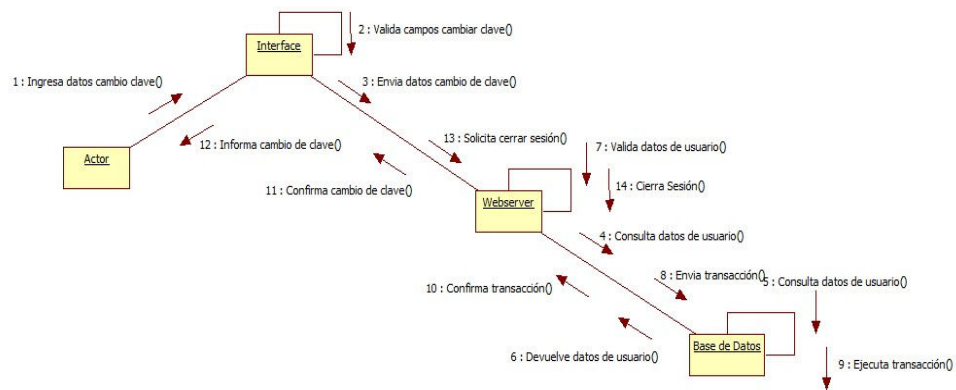


Figura 72. Diagrama de Colaboración Cambiar Clave

3.3.3.3 Diagrama de Colaboración Gestionar Tablas

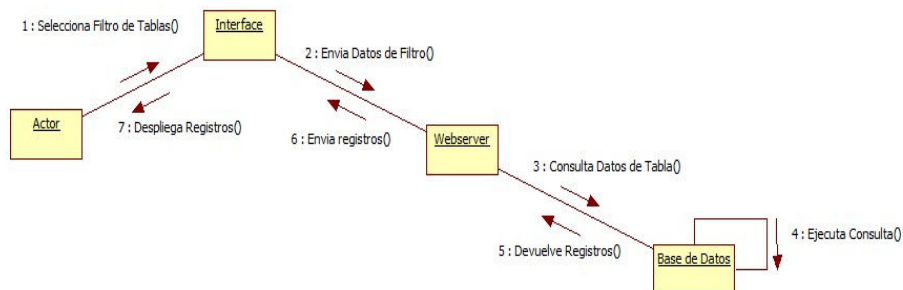


Figura 73. Diagrama de Colaboración Buscar Tablas

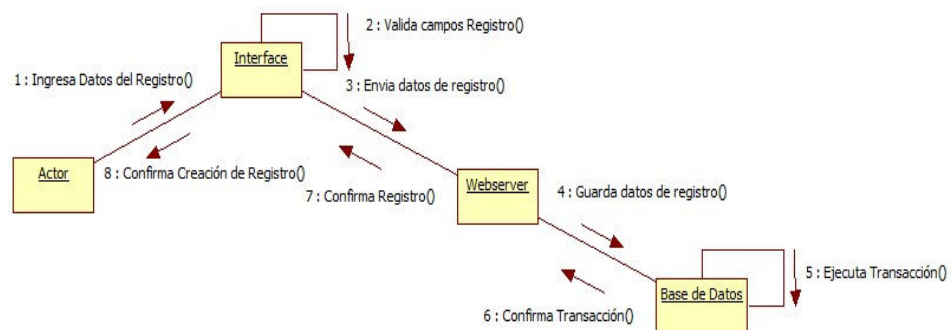


Figura 74. Diagrama de Colaboración Crear Registros

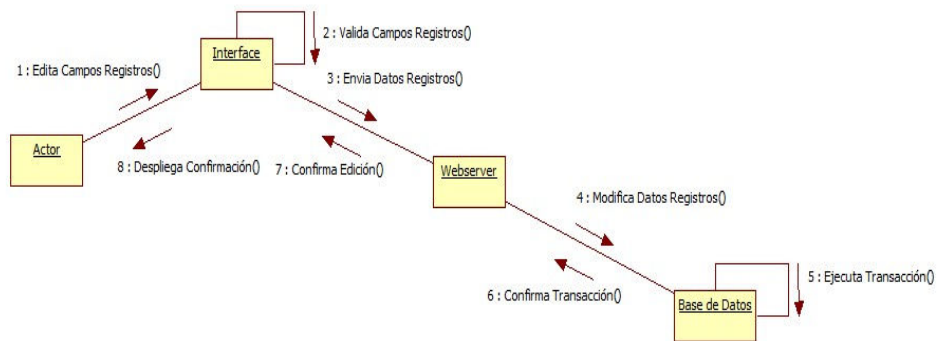


Figura 75. Diagrama de Colaboración Editar Registros

3.3.3.4 Diagrama de Colaboración Gestionar Clientes

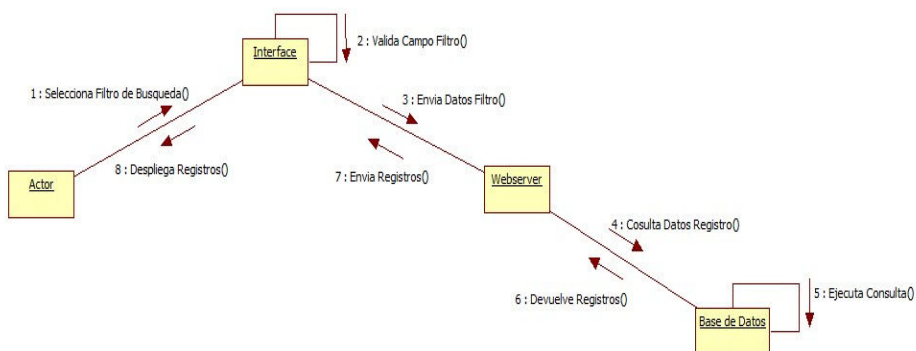


Figura 76. Diagrama de Colaboración Buscar Clientes

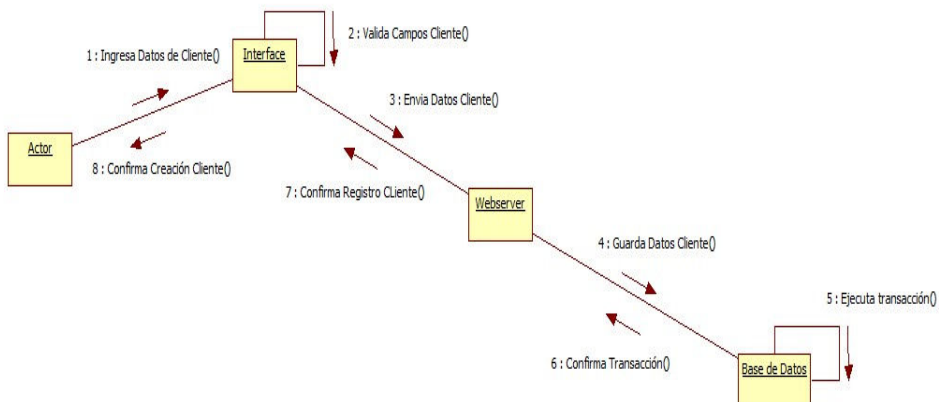


Figura 77. Diagrama de Colaboración Crear Clientes

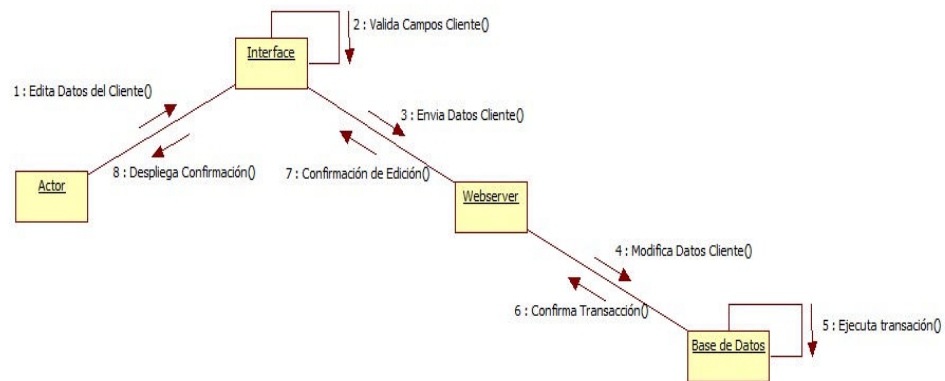


Figura 78. Diagrama de Colaboración Editar Clientes

3.3.3.5 Diagrama de Colaboración Gestionar Usuario

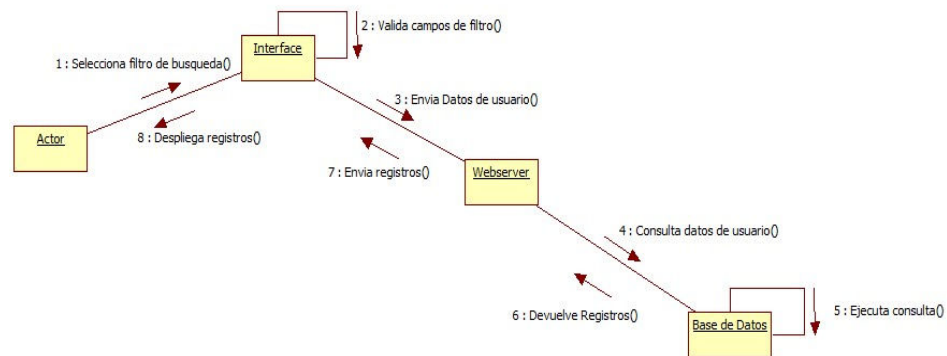


Figura 79. Diagrama de Colaboración Buscar Usuarios

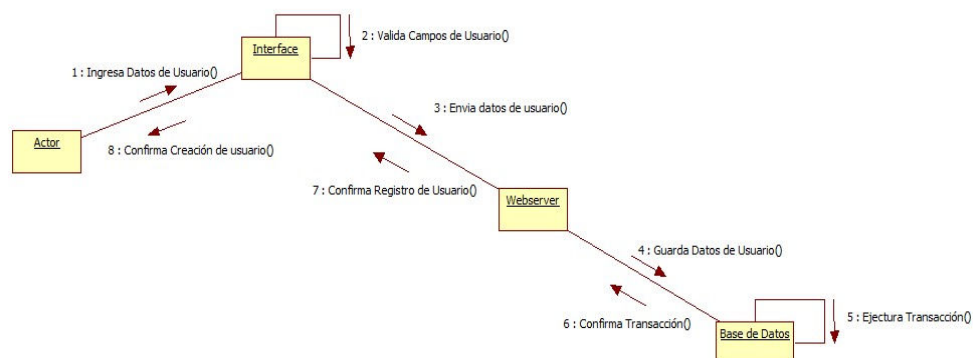


Figura 80. Diagrama de Colaboración Crear Usuarios

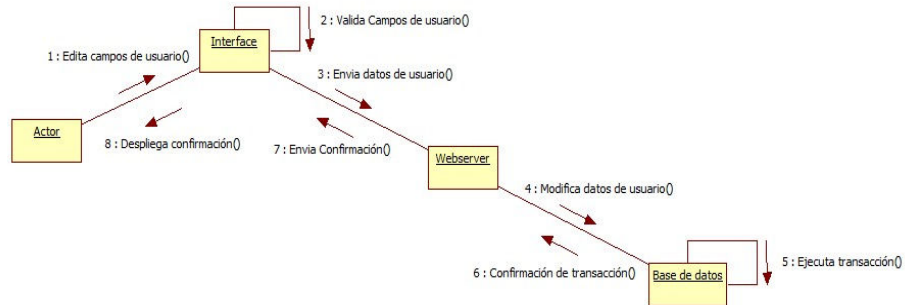


Figura 81. Diagrama de Colaboración Editar Usuarios

3.3.3.6 Diagrama de Colaboración Gestionar Empresas

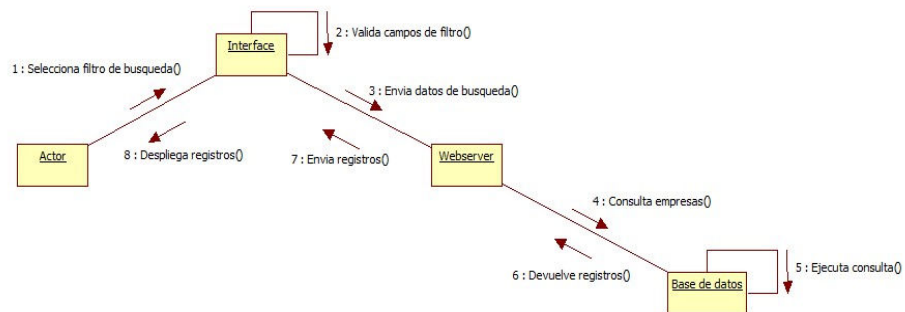


Figura 82. Diagrama de Colaboración Buscar Empresas

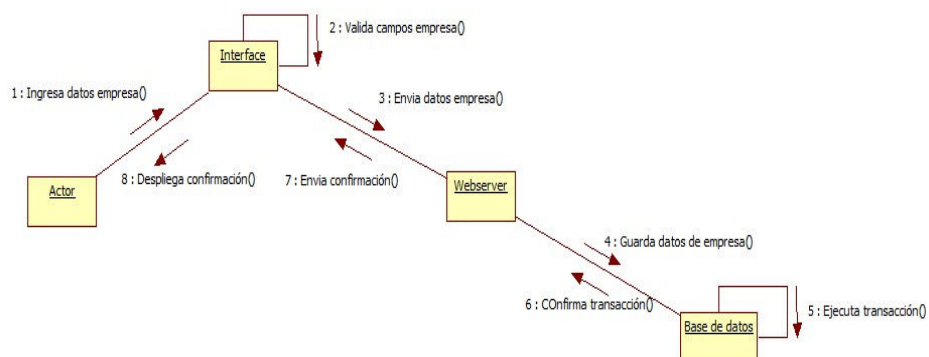


Figura 83. Diagrama de Colaboración Crear Empresas

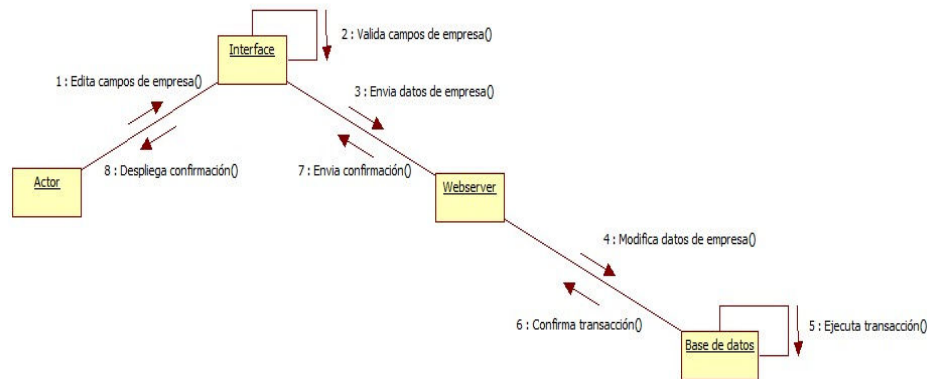


Figura 84. Diagrama de Colaboración Editar Empresas

3.3.3.7 Diagrama de Colaboración Gestionar Solicitudes

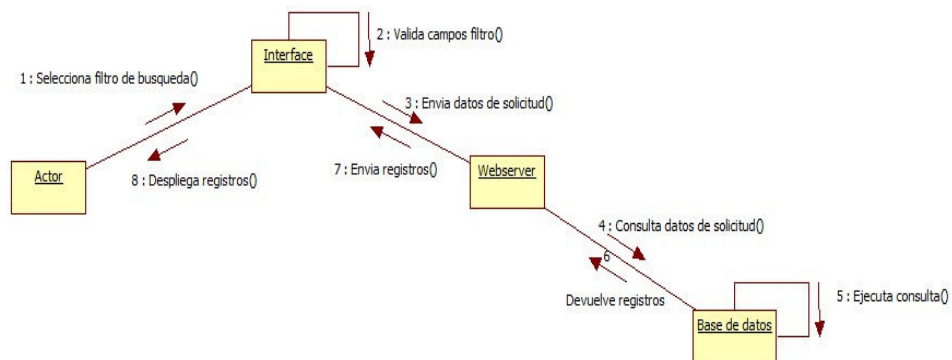


Figura 85. Diagrama de Colaboración Buscar Solicitudes

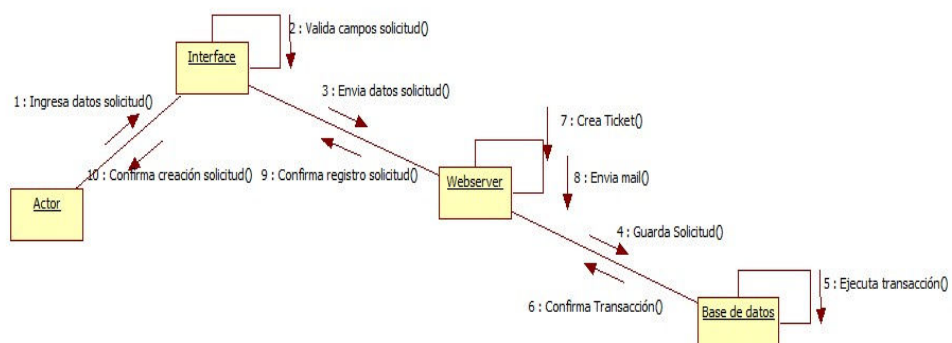


Figura 86. Diagrama de Colaboración Crear Solicitudes

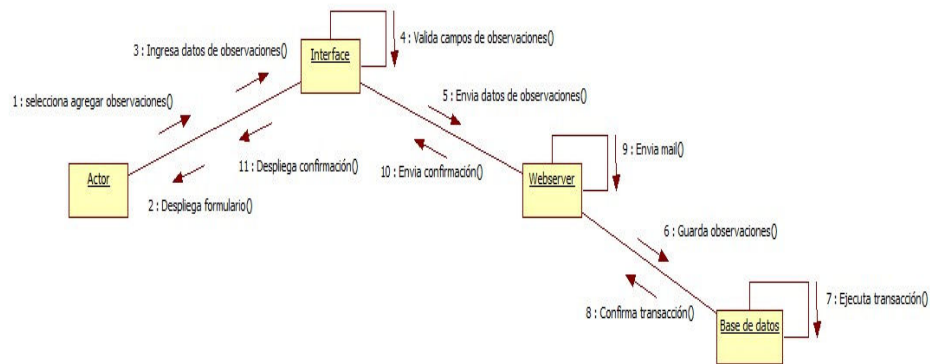


Figura 87. Diagrama de Colaboración Agregar Observaciones

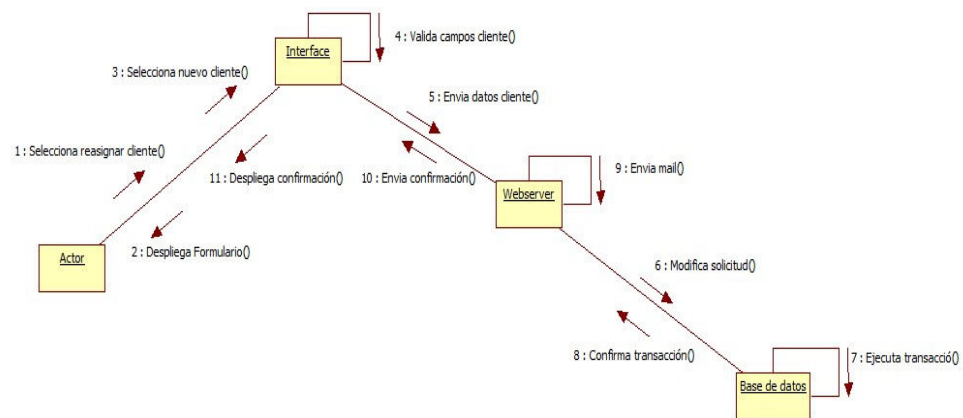


Figura 88. Diagrama de Colaboración Reasignar Cliente

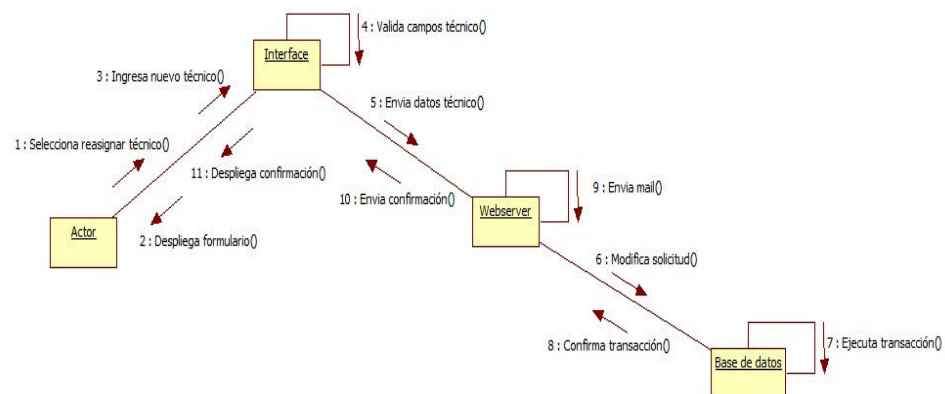


Figura 89. Diagrama de Colaboración Reasignar Técnico

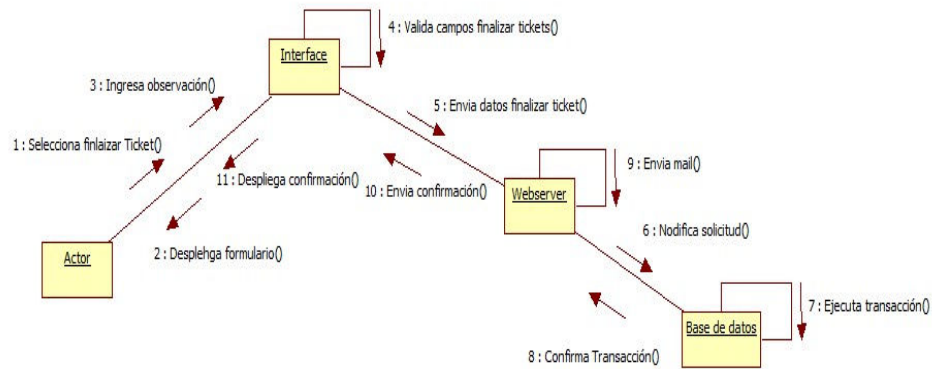


Figura 90. Diagrama de Colaboración Finalizar Ticket

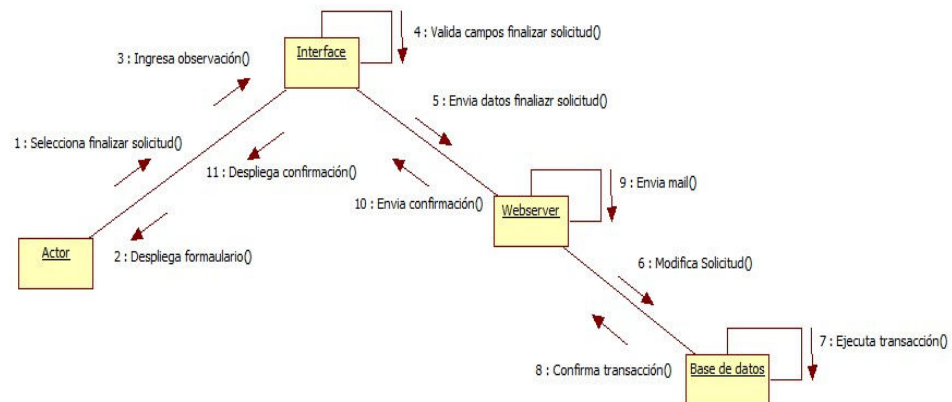


Figura 91. Diagrama de Colaboración Finalizar Solicitud

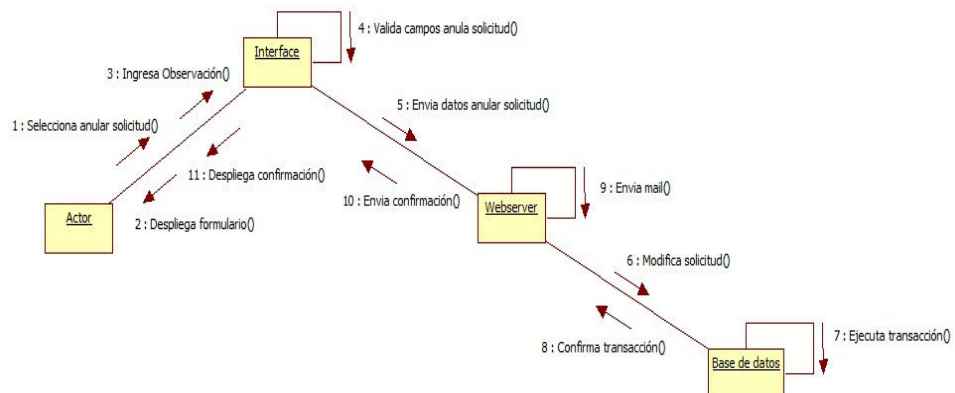


Figura 92. Diagrama de Colaboración Anular Solicitud

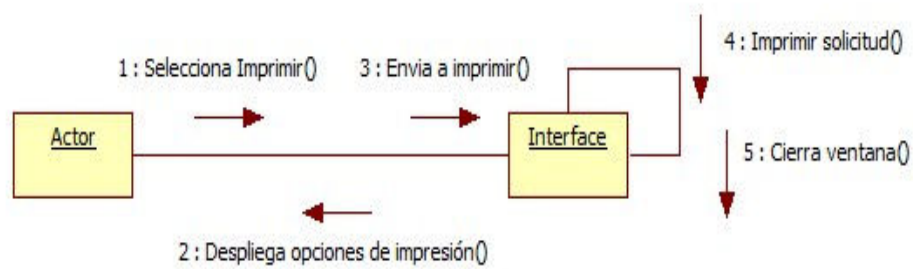


Figura 93. Diagrama de Colaboración Imprimir Solicitud

3.3.3.8 Diagrama de Colaboración Gestionar Técnicos

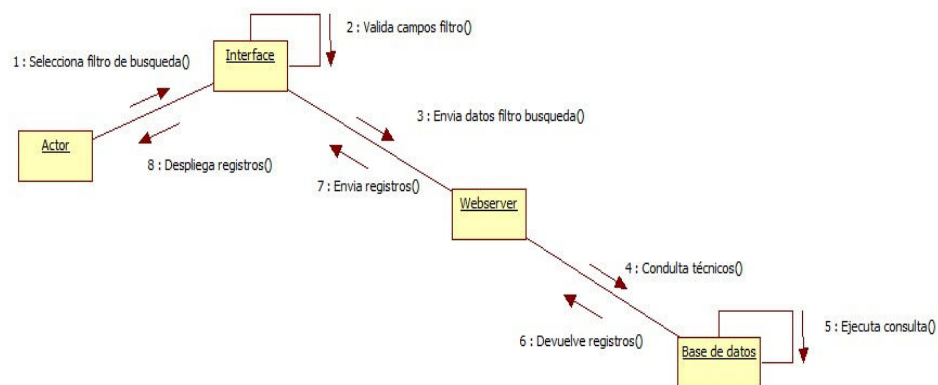


Figura 94. Diagrama de Colaboración Buscar Técnicos

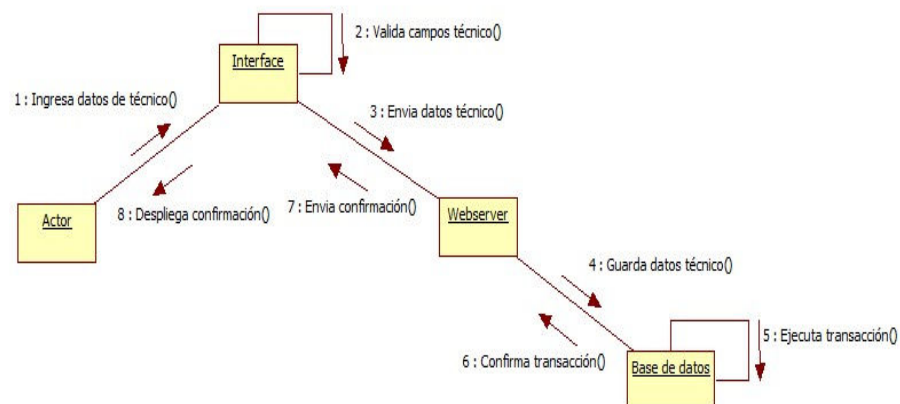


Figura 95. Diagrama de Colaboración Crear Técnicos

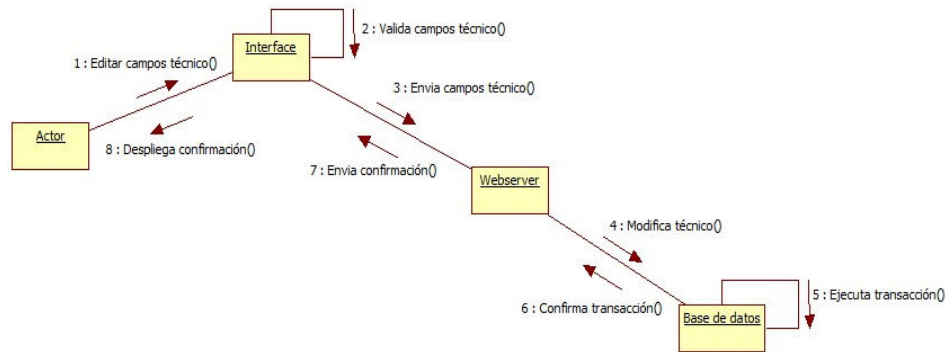


Figura 96. Diagrama de Colaboración Editar Técnicos

3.3.3.9 Diagrama de Colaboración Calificar Atención

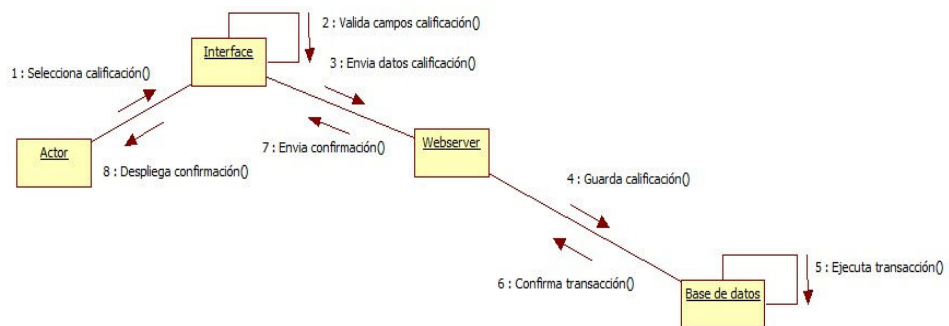


Figura 97. Diagrama de Colaboración Calificar Solicitud

3.3.3.10 Diagrama de Colaboración Generar Reportes

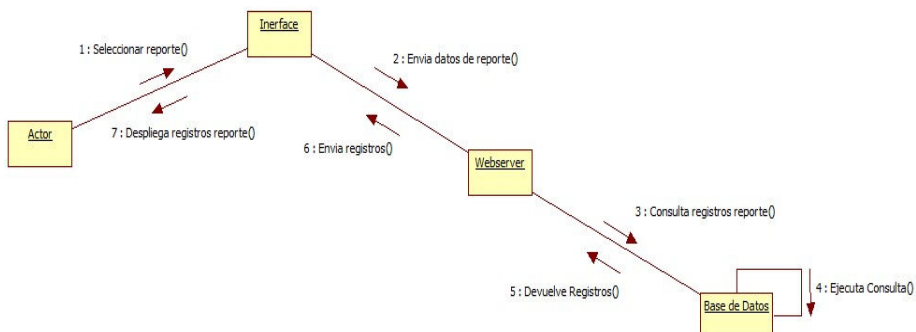


Figura 98. Diagrama de Colaboración Generar Reportes

3.3.4 Diagramas de Estado

Los diagramas de estado muestran todos los estados por los cuales pasan y pueden tomar los objetos en su vida útil dentro de la ejecución de un sistema informático.

3.3.4.1 Diagrama de Estado Clientes

Los clientes son creados en el sistema GSO inicialmente con estado Activo de manera predeterminada, también se puede definir el estado Con Acceso y Sin Acceso, esto depende si se desea que el cliente pueda acceder a crear solicitudes remotamente o solo forma local como parte del portafolio de clientes de la empresa, cuando el cliente ya no forme parte de la empresa cliente el estado puede ser modificado a Inactivo.

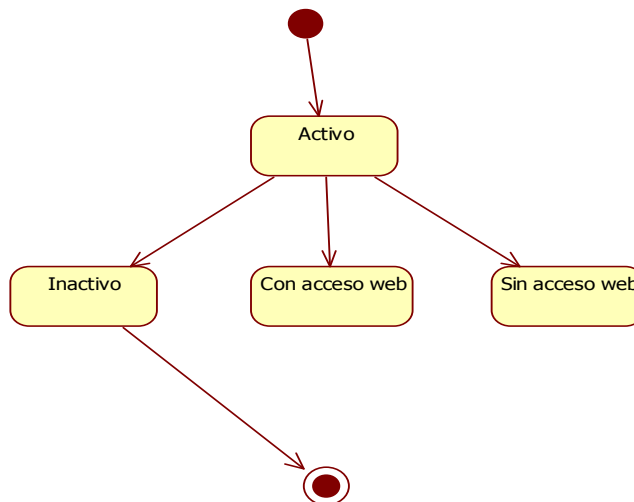


Figura 99. Diagrama de Estado Clientes

3.3.4.2 Diagrama de Estado Usuarios

Los Usuarios son creados en el sistema GSO inicialmente con estado Activo de manera predeterminada, también se puede definir el estado Con Acceso y Sin Acceso, cuando un usuario ya no forme parte de la empresa el estado puede ser modificado a Inactivo.

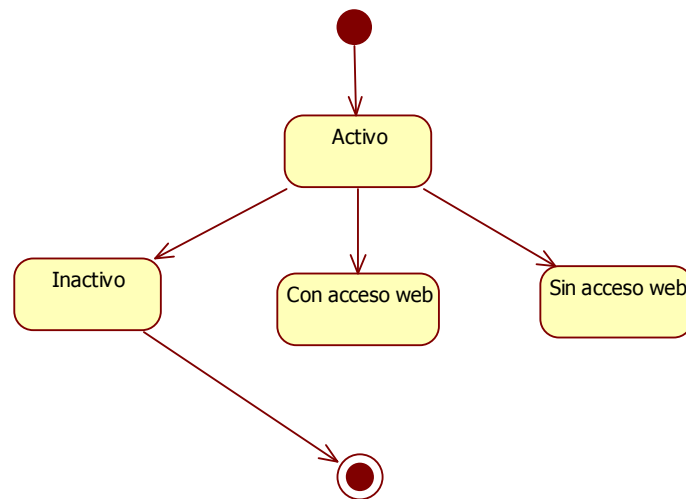


Figura 100. Diagrama de Estado Usuarios

3.3.4.3 Diagrama de Estado Empresas

Las empresas son creadas en el sistema GSO inicialmente con estado Activo de manera predeterminada, cuando una empresa deja de formar parte del portafolio de clientes el estado puede ser modificado a Inactivo.

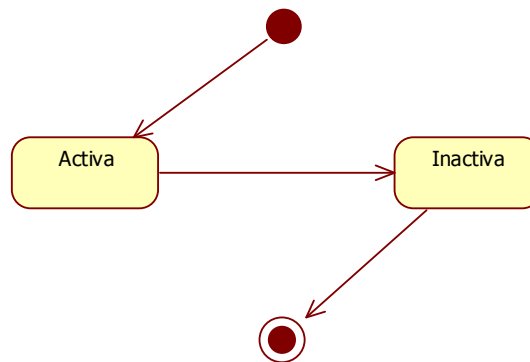


Figura 101. Diagrama de Estado Empresas

3.3.4.4 Diagrama de Estado Solicitudes

Las solicitudes pueden ser creadas de dos formas, pueden ser creadas sin asignar técnico o asignando un técnico de manera inmediata. Si se lo hace de la primera manera la solicitud inicialmente se crea con estado Pendiente, en el segundo caso se crea inicialmente con estado Asignada.

En el primer caso después de ser creada en estado Pendiente al momento de asignar técnico pasa a estado Asignada, cuando la atención a esa solicitud haya terminado pasa a estado Cerrada, y cuando se facturee al cliente o descontemos las horas de soporte brindada al cliente se cambia al estado Finalizado.

Las solicitudes podrán ser anuladas solo si están en estado Pendiente o Asignada en otros estados no podrá ya ser Anulada.

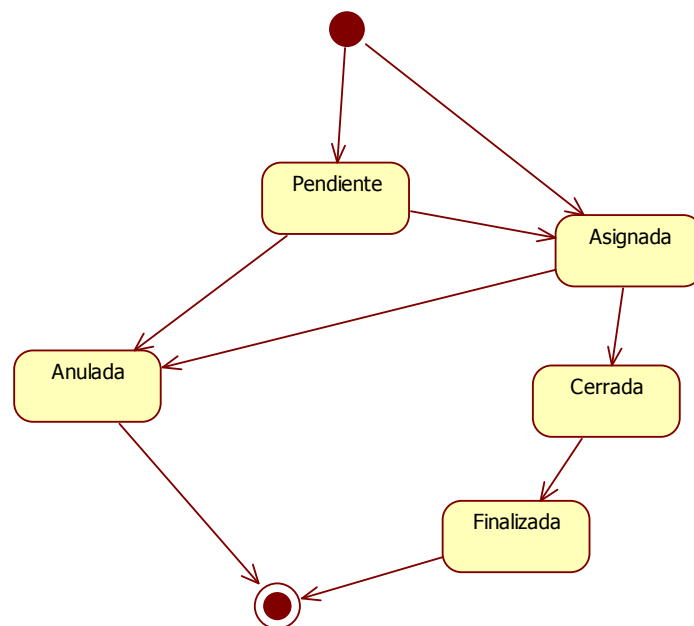


Figura 102. Diagrama de Estado Solicitudes

3.3.4.5 Diagrama de Estado Técnicos

Los Técnicos son creados en el sistema GSO inicialmente con estado Activo de manera predeterminada, también se puede definir el estado Con Acceso y Sin Acceso, cuando un técnico ya no forme parte de la empresa el estado puede ser modificado a Inactivo.

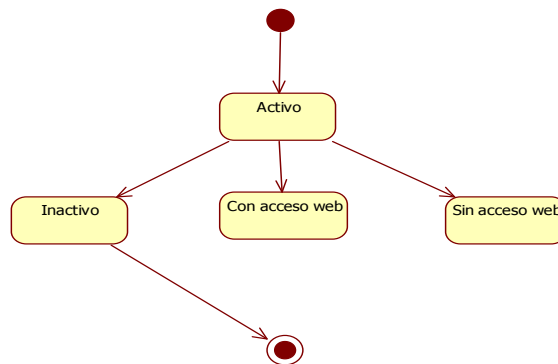


Figura 103. Diagrama de Estado Técnicos

3.3.5 Diagrama de Despliegue

En la Figura 104 se puede observar el diagrama de despliegue, en el cual se define toda la topología de hardware sobre la que va a ejecutarse el sistema G.S.O.

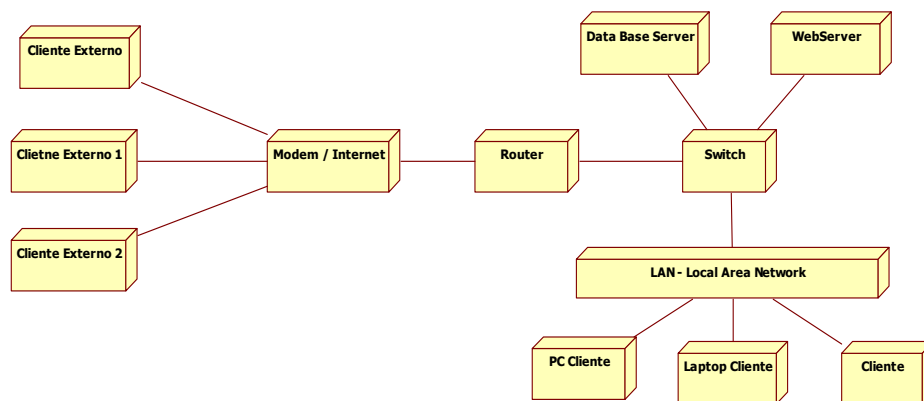


Figura 104. Diagrama de Despliegue Sistema G.S.O

3.3.6 Diagrama de Clases

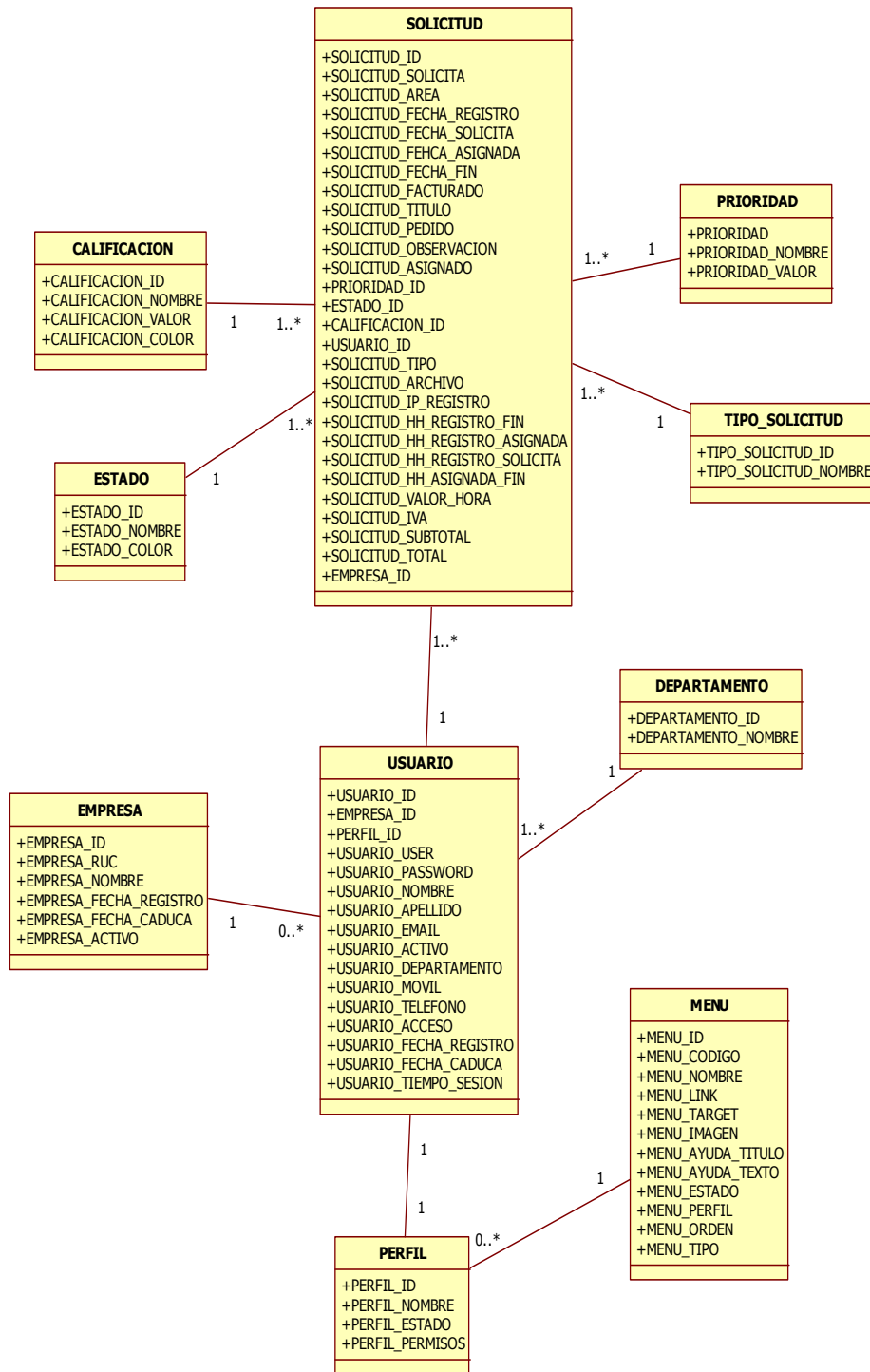


Figura 105. Diagrama de Clases G.S.O.

3.3.7 Diagrama Lógico Base de Datos

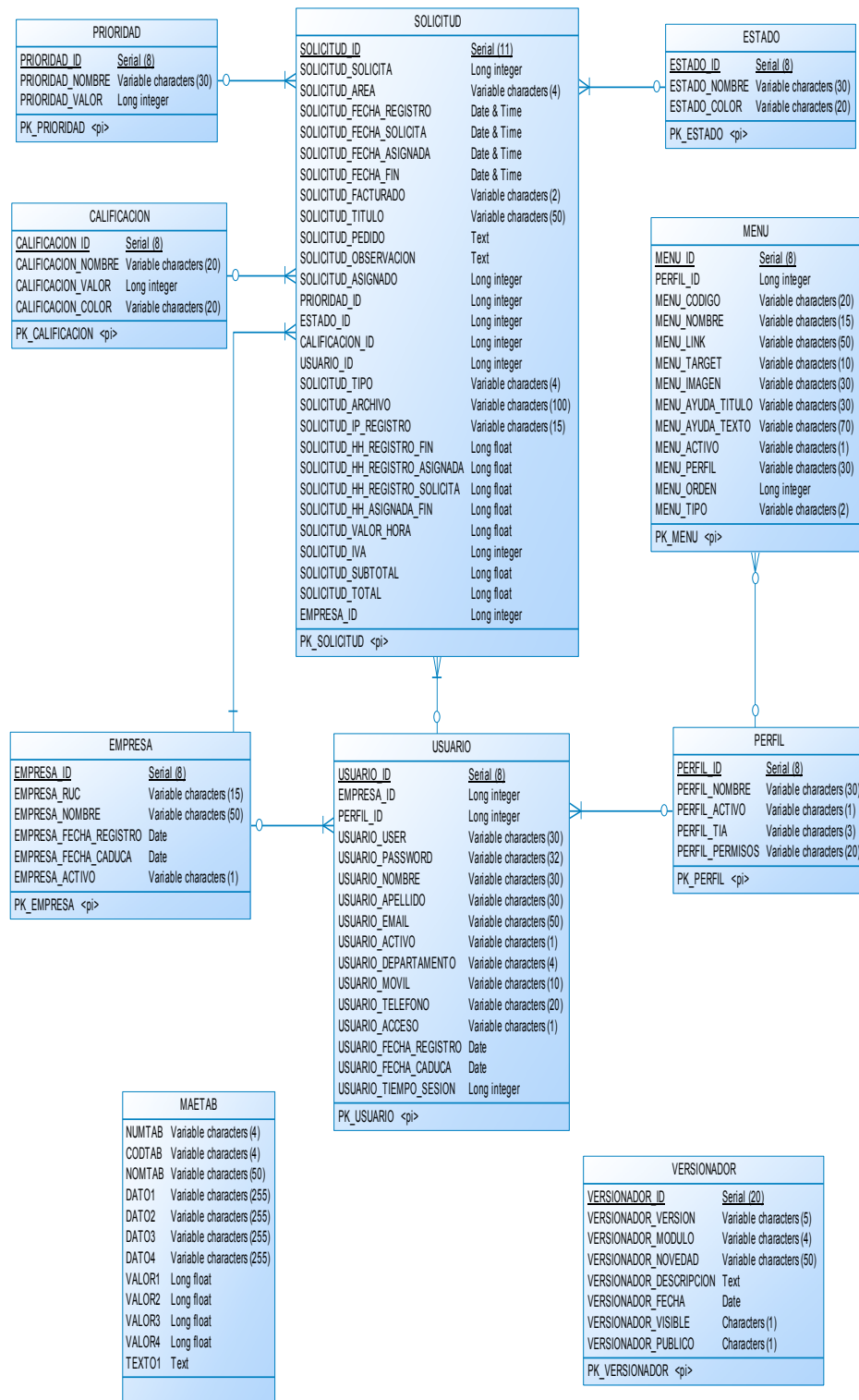


Figura 106. Diagrama Lógico Base de Datos G.S.O.

3.3.8 Diagrama Físico Base de Datos

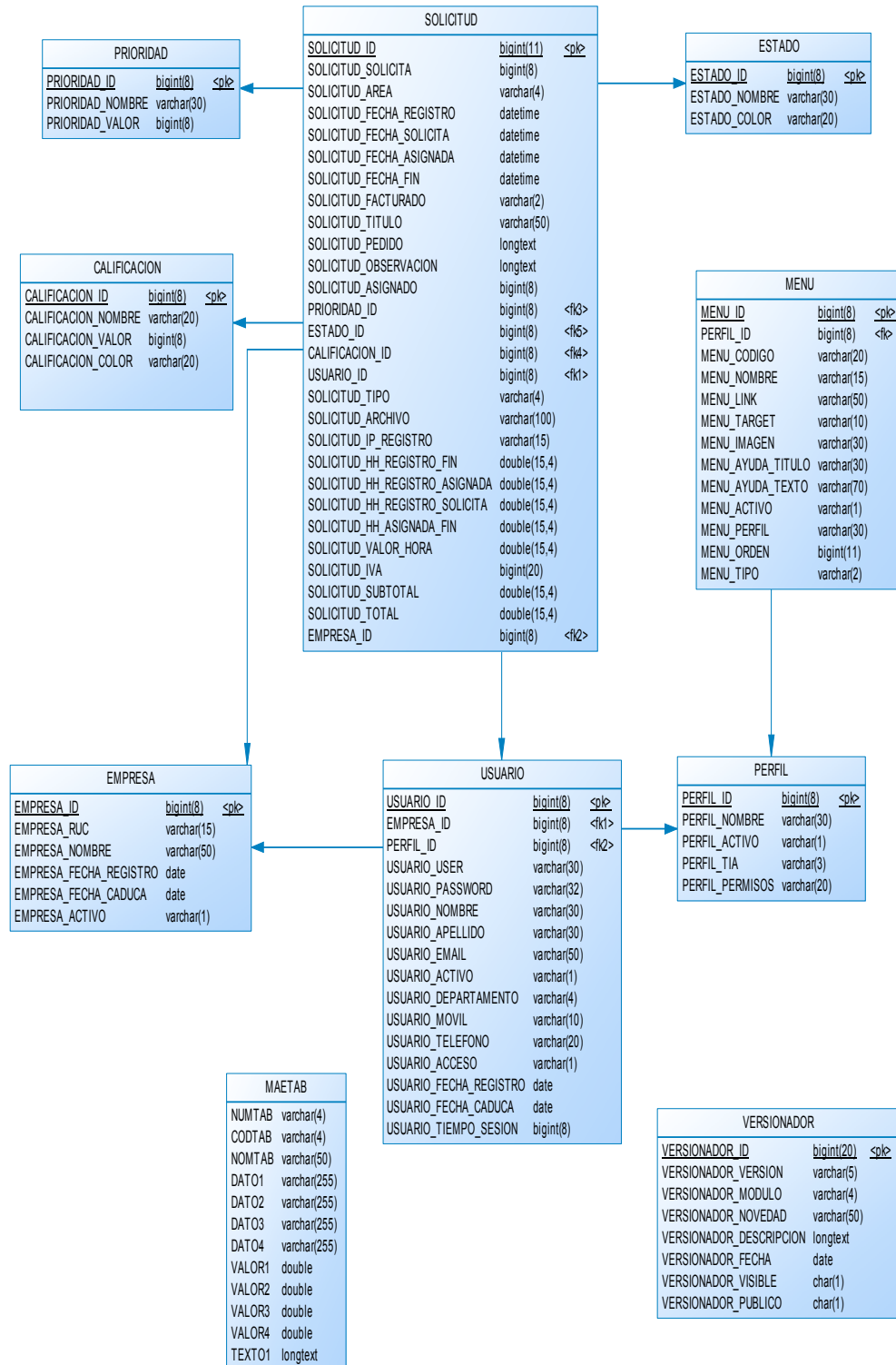


Figura 107. Diagrama Físico Base de Datos G.S.O

3.3.9 Diagrama Navegacional G.S.O.

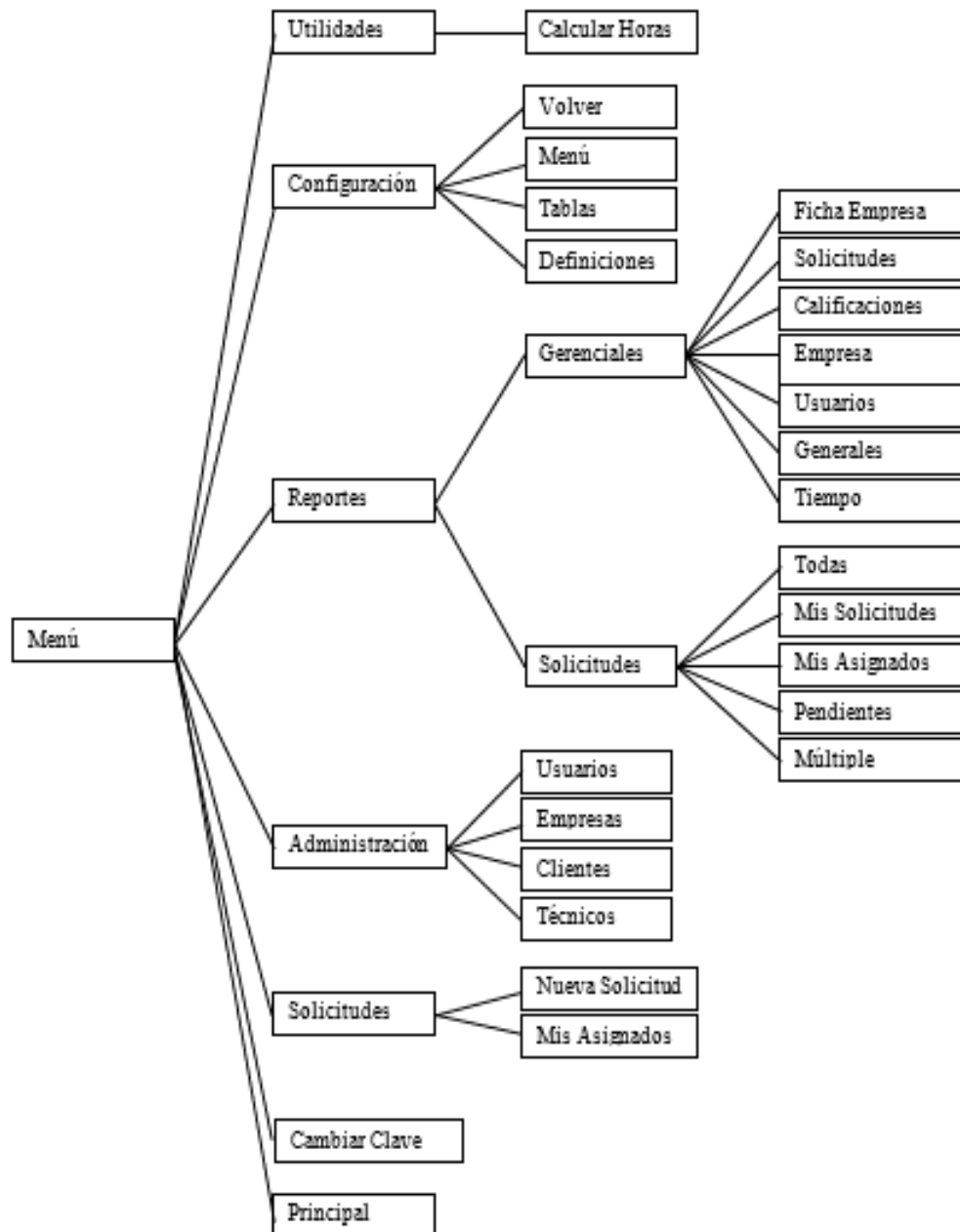


Figura 108. Diagrama Navegacional G.S.O.

CAPÍTULO 4

IMPLEMENTACIÓN

4.1 Arquitectura del sistema

Terminada la fase de análisis y diseño, se utilizaron las siguientes herramientas para el desarrollo de la aplicación y su posterior implementación:

1. Servidor XAMPP
 - a. Apache 2.0
 - b. MySQL 5.x
 - c. PHP 5.x
2. Netbeans 7.x

El tiempo estimado de desarrollo es de 4 meses con un programador senior en lenguaje PHP 5.x y un costo operativo de 7,000.00 USD aproximadamente.

Posterior al desarrollo para la implementación, se realiza de una manera ordenada y estructurada basada en el modelo de recopilación de información.

G.S.O. ha sido diseñado bajo los casos de uso que se utilizaron para el levantamiento de requerimientos funcionales de la empresa WICOM del Ecuador y su rol de negocio.

El aplicativo fue implementado en un servidor Linux con salida a internet mediante una IP Publica, para lo cual el servidor contiene las siguientes características:

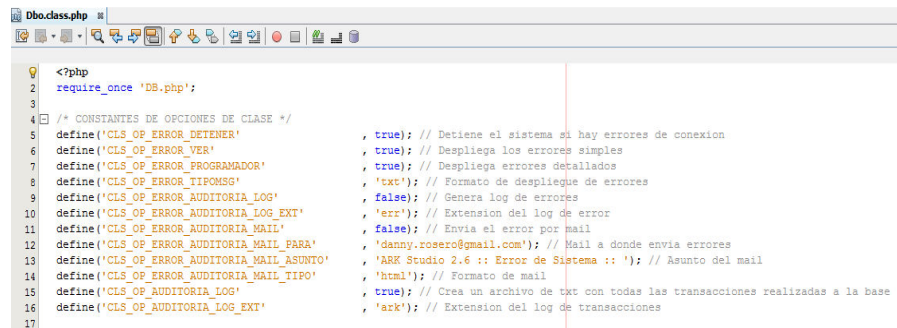
- Linux Centos 6.6 x64
- 2 tarjetas de red una para la LAN y una para la WAN
- Servidor Xampp 1.6.8

La estabilidad de los sistemas operativos Linux fue el factor preponderante para la selección e implementación del sistema, garantizando una alta disponibilidad de la aplicación.

4.2 Codificación

Dentro de la extensa codificación de G.S.O. “Gestión de Solicitudes Online” se puede encontrar secciones de código que le agregan un valor alto al sistema, como la clase de conexión a la base de datos, la que se muestra en las Figuras 109, 110, 111, 112, 113 y 114:

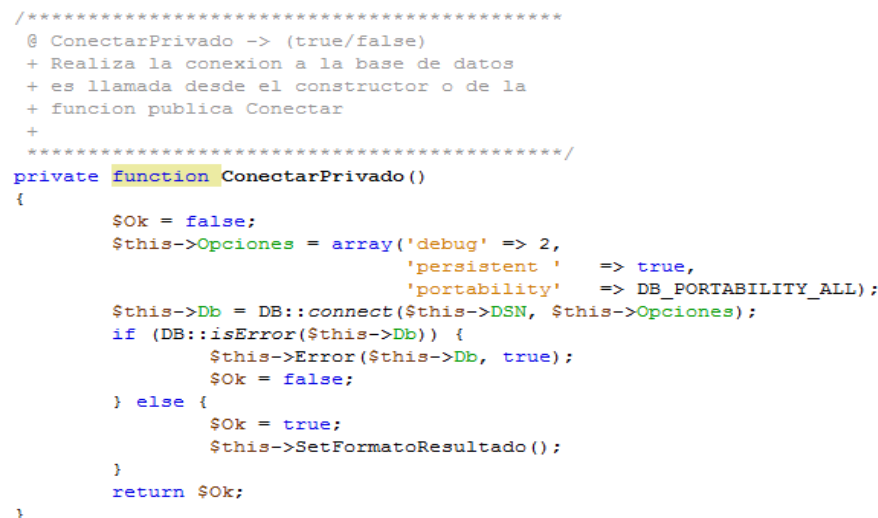
a. Clase de conexión a la base de datos



```

Dbo.class.php
1 <?php
2 require_once 'DB.php';
3
4 /* CONSTANTES DE OPCIONES DE CLASE */
5 define('CLS_OP_ERROR_DETENER' , true); // Detiene el sistema si hay errores de conexion
6 define('CLS_OP_ERROR_VER' , true); // Despliega los errores simples
7 define('CLS_OP_ERROR_PROGRAMADOR' , true); // Despliega errores detallados
8 define('CLS_OP_ERROR_TIPOMSG' , 'txt'); // Formato de despliegue de errores
9 define('CLS_OP_ERROR_AUDITORIA_LOG' , false); // Genera log de errores
10 define('CLS_OP_ERROR_AUDITORIA_LOG_EXT' , 'err'); // Extension del log de error
11 define('CLS_OP_ERROR_AUDITORIA_MAIL' , false); // Envia el error por mail
12 define('CLS_OP_ERROR_AUDITORIA_MAIL_PARA' , 'danny.rosero@gmail.com'); // Mail a donde envia errores
13 define('CLS_OP_ERROR_AUDITORIA_MAIL_ASUNTO' , 'ARK Studio 2.6 :: Error de Sistema :: '); // Asunto del mail
14 define('CLS_OP_ERROR_AUDITORIA_MAIL_TIPO' , 'html'); // Formato de mail
15 define('CLS_OP_AUDITORIA_LOG' , true); // Crea un archivo de txt con todas las transacciones realizadas a la base
16 define('CLS_OP_AUDITORIA_LOG_EXT' , 'ark'); // Extension del log de transacciones
17
  
```

Figura 109. Configuración Básica de la Clase de Base de Datos



```

/*****
@ ConectarPrivado -> (true/false)
+ Realiza la conexion a la base de datos
+ es llamada desde el constructor o de la
+ funcion publica Conectar
+
*****/
private function ConectarPrivado()
{
    $Ok = false;
    $this->Opciones = array('debug' => 2,
                          'persistent' => true,
                          'portability' => DB_PORTABILITY_ALL);
    $this->Db = DB::connect($this->DSN, $this->Opciones);
    if (DB::isError($this->Db)) {
        $this->Error($this->Db, true);
        $Ok = false;
    } else {
        $Ok = true;
        $this->SetFormatoResultado();
    }
    return $Ok;
}
/*****
  
```

Figura 110. Función de conexión privada a la Base de Datos

```

/*****
@ CambiarConexion -> (true/false)
+ Nos permite cambiar la conexion a una
+ diferente base de datos con solo enviarle
+ el DSN correspondiente.
+
*****/
public function CambiarConexion($db_informacion = NULL) {
    if (is_null($db_informacion)) {
        $this->Desconectar();
        if($this->GeneraDsn($this->DSN))
            return $this->ConectarPrivado();
    }else{
        $this->Desconectar();
        if($this->GeneraDsn($db_informacion))
            return $this->ConectarPrivado();
    }
}

```

Figura 111. Función Cambio de conexión

```

/*****
@ Conectar -> (true/false)
+ Realiza la conexion a la base de datos
+ es llamada desde fuera es de acceso publico
+ se le pasa la informacion de la Base
+
*****/
public function Conectar($db_informacion = NULL) {
    if (is_null($db_informacion))
        if($this->GeneraDsn($this->DSN))
            return $this->ConectarPrivado();
    else
        if($this->GeneraDsn($db_informacion))
            return $this->ConectarPrivado();
}

```

Figura 112. Función conexión publica a la base de datos

```

/*****
@ SetFormatoResultado
+ Setea el formato en el qu queremos que nos
+ devuelva el resultado del realizar un Query
+ se le puede pasar 3 opciones
+ ORDENADO, ASOCIADO (Default), OBJETO
+
*****/
public function SetFormatoResultado($formato='ASOCIADO') {
    switch ($formato) {
        case 'ORDENADO': $formato = DB_FETCHMODE_ORDERED; break;
        case 'ASOCIADO': $formato = DB_FETCHMODE_ASSOC; break;
        case 'OBJETO': $formato = DB_FETCHMODE_OBJECT; break;
        default: $formato = DB_FETCHMODE_ASSOC;
    }

    $this->Db->setFetchMode($formato);
}

```

Figura 113. Función que define el tipo de resultado


```
*****  
8 Query -> (true/false)  
+ Recibe el Sql lo prepara y lo ejecuta, se  
+ le puede ocupar de dos formas como  
+ se muestra a continuacion:  
+  
+ $DbCon => Objeto de conexion valido  
+ (1)  
+ $Sql = 'SELECT * FROM USUARIO';  
+ $DbCon->Query($Sql);  
+ (2)  
+ $Sql = 'SELECT *  
+      FROM USUARIO  
+      WHERE USERNAME = ? AND PASSWORD = ?';  
+ $Data = array('danny','danny2007')  
+ $DbCon->Query($Sql, $Data);  
+  
+  
+-----/  
  
public function Query($sql='', $datos=NULL){  
    $Ok = false;  
    if(is_null($datos)){  
        if($sql==''){  
            $this->Error('[Sql] La cadena esta vacia no existe operacion para ejecutar en el Query');  
        }else{  
            $this->$ResTmp = $this->Db->query($sql);  
            if (DB::isError($this->$ResTmp)) {  
                $this->Error($this->$ResTmp, true);  
                $Ok = false;  
            } else {  
                $Ok = true;  
                $this->SiguienteRegistro();  
            }  
        }  
    }else{  
        if($sql=='')  
            $this->Error('[Sql] La cadena esta vacia no existe operacion para ejecutar en el Query');  
        else{  
            $this->$ResTmp = $this->Db->query($sql, $datos);  
            if (DB::isError($this->$ResTmp)) {  
                $this->Error($this->$ResTmp, true);  
                $Ok = false;  
            } else {  
                $Ok = true;  
                $this->SiguienteRegistro();  
            }  
        }  
    }  
    return $Ok;  
}
```

Figura 114. Función Pública Query

4.3 Pruebas

4.3.1 Pruebas de Caja Blanca “Secuencia”

a. Prueba uno

Cuadro 36

Proceso de interacción creación de técnico

PROCESO DE INTERACCIÓN - CREACIÓN DE TÉCNICO		
	PASO	ACCIÓN
Flujo Normal	1	Inicio de Sesión
	2	Autenticación de usuario y clave proporcionados
	3	Al autenticarse correctamente visualizará el menú general, debe seleccionar <i>Administración->Técnicos</i>

Continúa

	4	Al ingresar al menú indicado visualizará el formulario de registro de técnico
	5	Debe llenar los campos indicados en el formulario
		Nombre (Requerido)
		Apellido (Requerido)
		Usuario (Requerido)
		Perfil (Requerido, Bloqueado, Seleccionado Técnico)
		Fecha de Ingreso (Requerido)
		Fecha de Caducidad
		Email
		Usuario Activo (Requerido SI/NO)
		Empresa (Requerido)
		Área / Departamento (Requerido)
		Teléfono Móvil
		Teléfono Fijo
		Acceso Web (Requerido Si/NO)
		Tiempo de Sesión (Requerido)
	6	Al ingresar todos los campos el sistema valida campos registrados
	7	Si los campos requeridos están completos, el sistema valida usuario, si el usuario no está ya registrado, se guarda la información proporcionada
Flujo Alternativo	PASO	ACCIÓN
	7	Si no están llenos todos los campos requeridos, el sistema envía una alerta indicando los campos faltantes
	8	Si el usuario ya está registrado en el sistema, este envía un mensaje de alerta y no guarda la información

Cuadro 37

Prueba Webmaster creación de técnico

PRUEBA WEBMASTER - CREACIÓN DE TÉCNICO				
No	Actor	Escenario	Resultado	Observación
001	WEBMASTER	Inicio de Sesión Usuario WEBMASTER Clave S3gur0	Satisfactorio	
002	WEBMASTER	Autenticación de Usuario y Clave	Satisfactorio	
003	WEBMASTER	Visualización de Menú General	Satisfactorio	
004	WEBMASTER	Acceso a Menú: Administración->Técnicos	Satisfactorio	
005	WEBMASTER	Visualización de formulario de registro	Satisfactorio	
006	WEBMASTER	Llenado de Campos	Satisfactorio	
Nombre: Guillermo				

Continúa →

Apellido: Obando			
Usuario: GOBANDO			
Perfil: Técnico			
Fecha de Ingreso: 2011/06/20			
Fecha de Caducidad			
Email: gobando@wicom.com.ec			
Usuario Activo: SI			
Empresa: Wicom del Ecuador			
Área / Departamento: Sistemas			
Teléfono Móvil			
Teléfono Fijo			
Acceso Web: SI			
Tiempo de Sesión: 30			
007	WEBMASTER	Validación de Campos requeridos	Satisfactorio
008	WEBMASTER	Validación de Usuario existente	Satisfactorio
009	WEBMASTER	Guarda información proporcionada	Satisfactorio

b. Prueba dos

Cuadro 38

Proceso de interacción creación de Solicitud

PROCESO DE INTERACCIÓN - CREACIÓN DE SOLICITUD		
	PASO	ACCIÓN
Flujo Normal	1	Inicio de Sesión
	2	Autenticación de usuario y clave proporcionados
	3	Al autenticarse correctamente visualizará el menú general, debe seleccionar <i>Solicitudes -> Nueva Solicitud</i>
	4	Al ingresar al menú indicado visualizará el formulario de registro de Solicitud
	5	Debe llenar los campos indicados en el formulario
		Cliente (Requerido)
		Título de Solicitud (Requerido)
		Solicitud (Requerido)
		Técnico Asignado
		Prioridad (Requerido)
		Tipo de Solicitud (Requerido)
		Archivo Adjunto
	6	Al ingresar todos los campos el sistema valida campos requeridos

Continúa →

	7	Si los campos requeridos están completos, se guarda la información proporcionada
	8	El sistema envía un mail de notificación al cliente, supervisor, gerente, recepción, técnico
Flujo Alternativo	PASO	ACCIÓN
	7	Si no están llenos todos los campos requeridos, el sistema envía una alerta indicando los campos faltantes
	8	El sistema valida asignación de técnico en ingreso de la solicitud para establecer estado Asignado, caso contrario estado Pendiente
	9	El sistema valida asignación de técnico en el ingreso de la solicitud para el envío de mail

Cuadro 39

Prueba Drosero creación de Solicitud

PRUEBA DROSERO CREACIÓN DE SOLICITUD				
No	Actor	Escenario	Resultado	Observación
001	DROSERO	Inicio de Sesión Usuario DROSERO Clave Daniel	Satisfactorio	
002	DROSERO	Autenticación de Usuario y Clave	Satisfactorio	
003	DROSERO	Visualización de Menú General	Satisfactorio	
004	DROSERO	Acceso a Menú: Solicitudes->Nueva Solicitud	Satisfactorio	
005	DROSERO	Visualización de formulario de registro	Satisfactorio	
006	DROSERO	Llenado de Campos	Satisfactorio	
		Cliente: Adriana Plaza		
		Título de Solicitud: Soporte help desk		
		Solicitud: Soporte impresora		
		Técnico Asignado:		
		Prioridad: Media		
		Tipo de Solicitud: Help Desk En Sitio		
		Archivo Adjunto:		
007	DROSERO	Validación de Campos requeridos	Satisfactorio	
008	DROSERO	Guarda información proporcionada	Satisfactorio	
009	DROSERO	Envío de mail cliente, supervisor, gerente, recepción, técnico	Satisfactorio	

4.3.2 Prueba de Caja Negra “Proceso”

a. Prueba uno

Cuadro 40

Proceso interacción finalizar solicitud

PROCESO DE INTERACCIÓN FINALIZAR SOLICITUD		
	PASO	ACCIÓN
Flujo Normal	1	Ingreso al Sistema G.S.O.
	2	Revisión Solicitudes Pendientes
	3	Buscar Solicitud a finalizar
	4	Seleccionar la solicitud, ésta se cargará en el formulario del sistema
	5	El actor ingresa la información correspondiente a la gestión realizada, y solución del problema de la siguiente manera:
		En atención a la solicitud # 0001, se realizó el siguiente procedimiento
		explicación del procedimiento (soporte) realizado, y solución del mismo
	6	El actor selecciona Finalizar Solicitud
	7	Fin de la gestión de Finalizar Solicitud

Cuadro 41

Prueba Drosero finalizar solicitud

PRUEBA DROSERO FINALIZAR SOLICITUD				
No	Actor	Escenario	Resultado	Observación
001	DROSERO	Inicio de Sesión Usuario DROSERO Clave Daniel	Satisfactorio	
002	DROSERO	Visualización de Solicitudes Pendientes	Satisfactorio	
003	DROSERO	Búsqueda de Solicitud Deseada	Satisfactorio	
004	DROSERO	FINALIZAR SOLICITUD/ Selección de Solicitud y carga de información	Satisfactorio	
005	DROSERO	FINALIZAR SOLICITUD/ Llenado de Observación, con gestión y solución de la	Satisfactorio	

Continúa →

		solicitud	
006	DROSERO	FINALIZAR SOLICITUD/ Selección de botón Finalizar Solicitud	Satisfactorio
007	DROSERO	FINALIZAR SOLICITUD/ Registrar solución del problema	Satisfactorio

b. Prueba dos

Cuadro 42

Proceso interacción Calificar Solicitud

PROCESO DE INTERACCIÓN PROCESO CALIFICAR SOLICITUD		
	PASO	ACCIÓN
Flujo Normal	1	Ingreso al Sistema G.S.O.
	2	Visualización de solicitudes Finalizadas
	3	Seleccionar Solicitud a calificar
	4	Revisión Solicitud Seleccionada
	5	Calificar Solicitud
	6	Continuar con la siguiente solicitud
	7	No hay solicitudes pendientes de calificación
	8	Fin proceso Calificar Solicitud

Cuadro 43

Prueba Webmaster calificación de solicitud

PRUEBA WEBMASTER CALIFICACIÓN DE SOLICITUD				
No	Actor	Escenario	Resultado	Observación
001	WEBMASTER	Inicio de Sesión Usuario WEBMASTER Clave S3gur0	Satisfactorio	
002	WEBMASTER	Visualización de Solicitudes Finalizadas	Satisfactorio	
003	WEBMASTER	Selección de Solicitud Deseada	Satisfactorio	

Continúa →

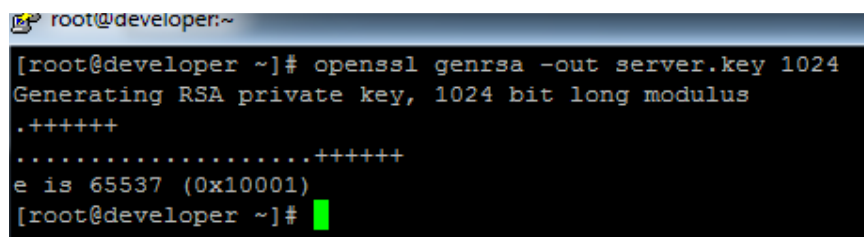
004	WEBMASTER	CALIFICAR SOLICITUD/ Revisión de gestión y solución realizada a la solicitud	Satisfactorio
005	WEBMASTER	CALIFICAR SOLICITUD/ Selección de la calificación obtenida por la gestión (Excelente, Muy Buena, Buena, Regular, Mala)	Satisfactorio
006	WEBMASTER	CALIFICAR SOLICITUD/ Selección solicitud siguiente por calificar	Satisfactorio
007	WEBMASTER	CALIFICAR SOLICITUD/ No existen solicitudes pendientes por calificar Fin Proceso Calificación	Satisfactorio

4.4 Seguridades

Para garantizar la seguridad de la información de la empresa WICOM del Ecuador, el sistema G.S.O. será implementado en un servidor web seguro utilizando el protocolo HTTPS “Hypertext Transfer Protocol Secure” para el acceso al sistema.

El servidor seguro cuenta con un certificado SSL con nivel de encriptación de 1024 bits, para su implementación se sigue los pasos a continuación:

1. Crear una llave de 1024 bits, como muestra la Figura 115.



```

root@developer:~
[root@developer ~]# openssl genrsa -out server.key 1024
Generating RSA private key, 1024 bit long modulus
..+++++
.....+++++
e is 65537 (0x10001)
[root@developer ~]#

```

Figura 115. Creación de llave SSL

2. Crear un CSR "Certificate Signing Request" con la información de WICOM del Ecuador, como se muestra en la Figura 116.

```
[root@developer ~]# openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:EC
State or Province Name (full name) [Berkshire]:Pichincha
Locality Name (eg, city) [Newbury]:Quito
Organization Name (eg, company) [My Company Ltd]:WICOM del Ecuador
Organizational Unit Name (eg, section) []:Soporte
Common Name (eg, your name or your server's hostname) []:gsowicom.dnet.ec
Email Address []:danny.rosero@gmail.com
problems making Certificate Request
29227:error:0D07A07C:asn1 encoding routines:ASN1_mbstring_ncopy:illegal character:s:a_mbstr.c:162:
[root@developer ~]#
```

Figura 116. Creación de CSR

3. Crear el certificado SSL como lo muestra la Figura 117, para lo cual se necesita la llave server.key y el archivo server.csr.

```
[root@developer ~]# openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
Signature ok
subject=/C=EC/ST=Pichincha/L=Quito/O=WICOM del Ecuador/OU=Soporte/CN=gsowicom.dnet.ec/emailAddress=danny.rosero@gmail.com
Getting Private key
[root@developer ~]#
```

Figura 117. Creación de Certificado.

4. Copiar los archivos de certificado y llave a los directorios del servidor, como se muestra en las Figuras 118, 119.

```
[root@developer ~]# cp server.crt /opt/lampp/etc/ssl.crt/
```

Figura 118. Copia certificado.

```
[root@developer ssl.key]# cp /root/server.key ../ssl.key/
```

Figura 119. Copia llave.

5. Iniciar el servicio SSL del servidor, como muestra la Figura 120.

```
[root@developer ssl.key]# /opt/lampp/lampp startssl
```

Figura 120. Inicio de servidor SSL.

6. Validación de certificado en el navegador, como muestra las figuras 121 y 122.

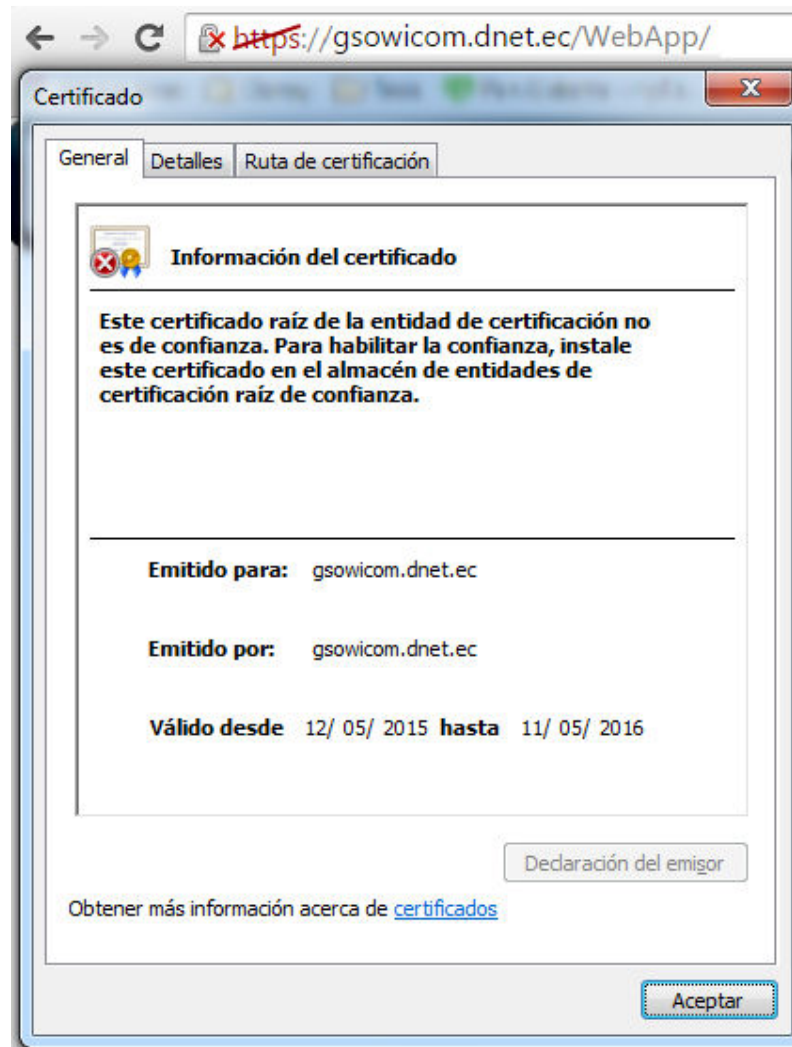


Figura 121. Información de certificado.

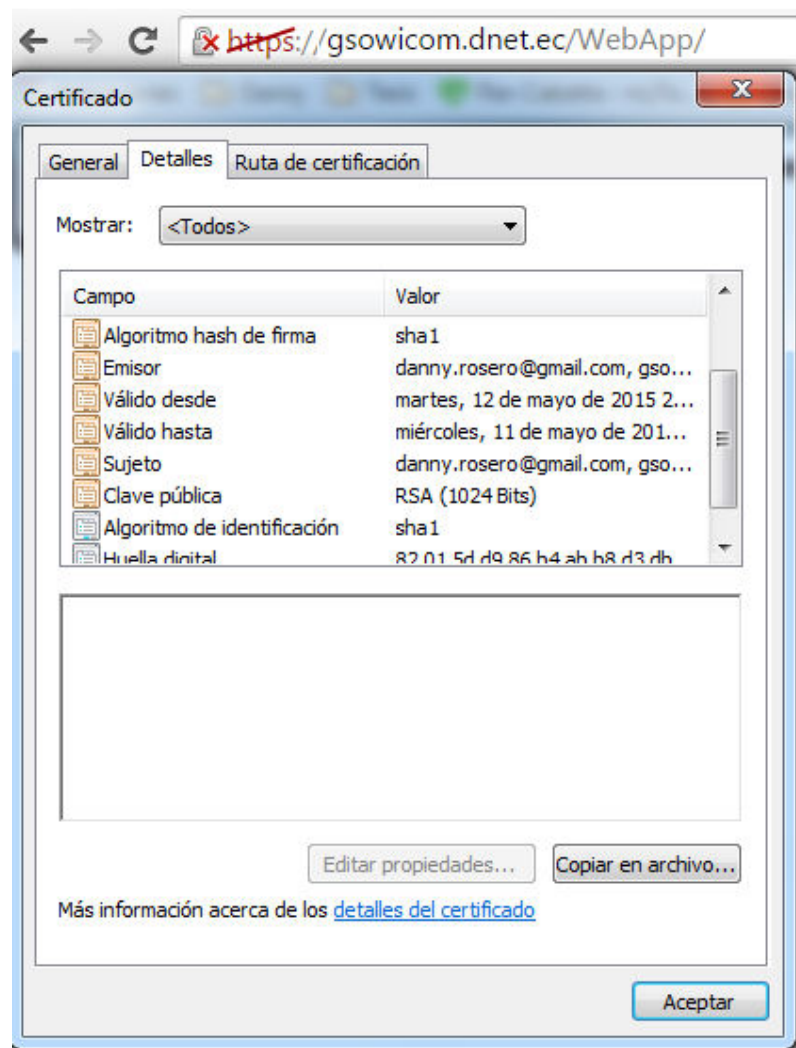


Figura 122. Detalle de certificado.

4.5 Manuales G.S.O.

4.5.1 Manual de Referencia Técnica

Para revisar el manual de usuario final por favor dirigirse al (ANEXO B).

4.5.2 Manual de Usuario Final

Para revisar el manual de usuario final por favor dirigirse al (ANEXO C).

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Durante el proceso de desarrollo e implementación del software para la gestión de solicitudes de helpdesk online G.S.O. en la empresa WICOM del Ecuador se pudo comprobar que la sistematización de procesos de una área o de una empresa por completo, es el camino más eficiente para llegar a una optimización de procesos, tiempos y recursos dentro de un área, además de brindarnos una manera eficaz de control sobre los mismos.
- Se pudo comprobar el rendimiento y funcionalidad de las herramientas de libre distribución para generar proyectos de gama amplia, la incursión de bajos costos y resultados de calidad orienta a las empresas a utilizar más de este tipo de herramientas, G.S.O. es un ejemplo palpable de la calidad de resultados que se puede obtener al utilizar este tipo de herramientas y tecnologías llevando a la empresa a un ahorro considerable y beneficios económicos a futuro.
- El buen análisis y diseño previo al desarrollo es base fundamental para obtener un software de calidad, una vez más queda demostrado que la norma IEEE830 para el levantamiento de requerimientos en base a RUP y UML es la forma más eficiente para definir un sistema en el ámbito de requerimientos que podrán ser implementados de manera clara en el desarrollo consiguiente.
- WICOM del Ecuador demostró conformidad total con el sistema implementado, G.S.O. se acoplo perfectamente a los procesos del departamento de Soporte Técnico, cumpliendo con las expectativas propuestas inicialmente al solicitar el auspicio para el desarrollo del presente proyecto.
- En la actualidad los sistemas orientados a la nube toman fuerza dentro de las empresas, la facilidad de mantenimiento, distribución y acceso los

han convertido en los grandes ganadores de los últimos años, PHP, JavaScript, HTML y CSS han demostrado ser lenguajes robustos con los que se puede crear aplicaciones altamente transaccionales y robustas.

5.2 Recomendaciones

- Para mantener una saludable vida útil de la base de datos, se recomienda realizar mantenimientos periódicos a las tablas de e índices en MySQL, la forma más óptima de hacerlo es utilizando la herramienta PHPMyAdmin, con esto se mantienen tiempos óptimos en las transacciones a la base de datos.
- El IDE de desarrollo adecuado para el mantenimiento o expansión del sistema es NetBeans, permite mantener una alta organización sobre el código y la estructura del proyecto, su integración y autoayuda permite realizar un desarrollo más rápido y eficiente.
- Si se presentan dudas sobre algún ítem del sistema por parte de los usuarios podrán consultar el manual correspondiente entregado en la empresa donde podrán guiarse paso a paso como realizar un proceso.
- En caso del programador de la empresa, responsable de velar por el funcionamiento del sistema G.S.O. y encargado de realizar los mantenimientos respectivos, podrá de igual forma consultar en el manual técnico entregado a la empresa.
- Para mantener un nivel de servicio excelente se recomienda a la empresa WICOM del Ecuador revisar los indicadores de calidad en el sistema G.S.O. alimentado constantemente por la calificación de los clientes a los diversos requerimientos solicitados e ingresados al sistema.

BIBLIOGRAFÍA

Ley de la propiedad intelectual. (19 de Mayo de 1998). Obtenido de Ley de la propiedad intelectual: <http://www.cetid.abogados.ec/archivos/56.pdf>

Plan Nacional del Buen Vivir. (24 de Junio de 2013). Obtenido de Plan Nacional del Buen Vivir: <http://www.buenvivir.gob.ec/documents/10157/26effa35-aaa8-4aec-a11c-be69abd6e40a>

Aguilar Uribe, S. (29 de Noviembre de 2014). *Metodología RUP.* Obtenido de Metodología RUP: <https://prezi.com/ngkncfupqx0h/metodologia-rup/>

Casasola Romero, O. (27 de Agosto de 2010). *Programación en Castellano.* Obtenido de Introducci-on a UML: http://programacion.net/articulo/introduccion_a_uml_181

Cortés Álvarez, J. (17 de Febrero de 2013). *Metodología RUP.* Obtenido de Metodología RUP: <http://es.slideshare.net/cortesalvarez/metodologia-rup>

Eguiluz, J. (s.f.). *Libros Web.* Obtenido de Introducción a Ajax: <http://librosweb.es/libro/ajax/>

Garcia Castellano, F. (2 de Marzo de 2003). *Tutorial de PHP.* Obtenido de Tutorial de PHP: <http://flanagan.ugr.es/php/index2.htm>

Guevara, D., Blanco, M., Alejos, O., Pinto, A., Valera, O., & Vasquez, L. (25 de Junio de 2012). *Metodología RUP.* Obtenido de Metodología RUP: <http://mtdologiarup.blogspot.com/>

Gutierrez, D. (1 de Abril de 2011). *Casos de Uso.* Obtenido de Diagramas de Caos de Uso: http://www.codecompiling.net/files/slides/UML_clase_02_UML_casos_de_uso.pdf

Juan Fuente, A. A., & Cueva Lovelle, J. M. (1 de Enero de 2006). *Justificación del uso del proceso unificado de software como modelo*. Obtenido de <https://forja.rediris.es/docman/view.php/227/369/Proceso%20Unificado.pdf>

Lamarca Lapuente, M. (08 de Diciembre de 2013). *Hipertexto*. Obtenido de Modelo OOHDM: <http://www.hipertexto.info/documentos/oohdm.htm>

López Rodríguez, C. (28 de Julio de 2003). *Ejemplo Desarrollo metodología RUP*. Obtenido de Ejemplo Desarrollo metodología RUP: <http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/>

Minkyu, L., Hyunsoo, K., Jeongil, K., Jangwoo, L., & Deukkyu, G. (1 de Enero de 2005). *StarUML 5.0 User Guide*. Obtenido de [http://staruml.sourceforge.net/docs/user-guide\(en\)/toc.html](http://staruml.sourceforge.net/docs/user-guide(en)/toc.html)

Sanchez, V. (31 de Julio de 2012). *Todo Sobre UML*. Obtenido de Todo Sobre UML: <http://es.slideshare.net/jonathanmatias2012/todo-sobre-uml>

Silva, F., Fajardo, J., & Amado, C. (1 de Abril de 2007). *Diseño de Sistemas de Información*. Obtenido de <http://sismeca12009.blogspot.com/2009/04/estudio-de-la-factibilidad.html>

Villalobos, J. (16 de Octubre de 2010). *Código Programación*. Obtenido de Introducción a Javascript: http://codigoprogramacion.com/cursos/javascript/88-introduccion-javascript.html#.VUfY6PI_Oko

Zavaleta, R. (08 de Agosto de 2010). *Roussez's Blog*. Obtenido de Metodología FDD – Feature Driven Development / Desarrollo Basado en Funciones:

<https://rousselz.wordpress.com/2010/08/08/metodologia-fdd-feature-driven-development-desarrollo-basado-en-funciones/>