



INSTITUTO TECNOLÓGICO  
“CORDILLERA”

CARRERA DE ANÁLISIS DE SISTEMAS

CONTROL DE PROCESOS DE SOPORTE TÉCNICO, PROYECTOS DE  
DESARROLLO E INFRAESTRUCTURA MEDIANTE UNA APLICACIÓN WEB  
EN LA EMPRESA CITYTECH.

Proyecto de investigación previo a la obtención del título de tecnólogo en análisis de  
sistemas

Autor: Mosquera Vélez Rubén Augusto

Tutor: Ing. Johnny Coronel

Quito – Octubre 2014

## DECLARATORIA

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

---

Mosquera Vélez Rubén Augusto

CC: 1723536015

## CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante Mosquera Vélez Rubén Augusto, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

**PRIMERA: ANTECEDENTE.- a)** El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “Control de procesos de soporte técnico, proyectos de desarrollo e infraestructura mediante una aplicación web en la empresa Citytech.”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. **b)** Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la

obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

**SEGUNDA: CESIÓN Y TRANSFERENCIA.-** Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

**TERCERA: OBLIGACIÓN DEL CEDENTE.-** El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

**CUARTA: CUANTIA.-** La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

**QUINTA: PLAZO.-** La vigencia del presente contrato es indefinida.

**SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.-** Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvencción,

caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

**SÉPTIMA: ACEPTACIÓN.-** Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los 28 días del mes de Octubre del dos mil catorce.

f) \_\_\_\_\_

C.C. N° 1723536015

**CEDENTE**

f) \_\_\_\_\_

Instituto Superior Tecnológico Cordillera

**CESIONARIO**

## AGRADECIMIENTO

Primero me gustaría agradecer a Dios por permitirme gozar un día más de vida, a toda mi familia por ser mi apoyo y guía en este largo camino, en especial a mi madre por ser la fuerza que me impulsa a seguir aunque en el trayecto encuentre obstáculos, por sus palabras de aliento que siempre supieron llegar en el momento adecuado, por confiar en mí y nunca perder la fe, a mi padre por la gran inspiración de mi vida, por su entereza su valentía, por cada consejo que ha sabido brindarme, a mis hermanas por el aliento y palabras de apoyo, este también es el fruto de su trabajo: Me gustaría agradecer todos los docentes que han sido parte de mi formación académica, en especial quisiera agradecer a la Ing. Patricia Garzón, quien ha sabido brindarme su guía y conocimiento, en especial porque ha sabido tener paciencia y pación por enseñar. Me gustaría agradecer a mis compañeros con quienes he compartido momentos de alegría y tristeza, esos momentos quedarán inmortalizado en mi memoria.

## **DEDICATORIA**

El presente trabajo lo dedico a toda mi familia, en especial a mi padre y a mi madre quienes han sido los pilares fundamentales a lo largo de toda mi vida, por no perder la confianza en mí a pesar de las grandes decepciones y sobre todo por enseñarme que la perseverancia es la clave del éxito.

## ÍNDICE GENERAL

Capítulo I: Antecedentes.....	1
1.01 Contexto.....	1
1.02 Justificación .....	2
1.03.01 Conclusión tabla matriz T .....	3
Capítulo II: Análisis de involucrados.....	4
2.01 Requerimientos .....	4
2.01.1 Descripción del Sistema Actual .....	4
2.01.2 Visión y Alcance.....	5
2.01.2 Entrevista .....	6
2.01.4 Matriz de Requerimientos .....	6
2.01.5 Descripción Detallada .....	7
2.02 Mapeo de involucrados .....	13
2.03 Matriz de involucrados .....	13
Capítulo III: Problemas y objetivos .....	14
3.01 Árbol de problemas.....	14
3.01.1 Análisis del árbol de problemas .....	15
3.02 Árbol de objetivos.....	16
3.02.1 Análisis del árbol de objetivos .....	17
3.03 Diagramas de casos de uso.....	17
3.04 Casos de uso de realización .....	19
3.05 Diagramas de secuencia del sistema .....	22
3.05 Especificación de casos de uso .....	23
Capítulo IV: Análisis de alternativas. ....	26
4.01 Matriz de análisis de alternativas.....	26
4.02 Matriz de impactos de objetivos .....	27
4.03. Estándares para el diseño de clases.....	27
4.04. Diagrama de clases .....	29
4.05. Modelo lógico – físico .....	29
4.06. Diagrama de componentes .....	29
4.07. Diagrama de estrategias .....	30
4.08. Matriz de marco lógico .....	31
4.09. Vistas arquitectónicas .....	31
4.09.1 Vista Lógica .....	31
4.09.2 Vista Física.....	32

4.09.3 Vista de desarrollo .....	32
4.09.4 Vista de procesos .....	33
Capítulo V: Propuesta .....	34
5.01 Especificación de estándares de programación.....	34
5.02 Diseño de interfaz de usuario.....	37
5.03 Especificación de pruebas de unidad.....	58
5.04 Especificación de pruebas de aceptación.....	73
5.05 Especificación de pruebas de carga.....	75
5.06 Configuración del ambiente mínimo/ideal.....	75
Capítulo VI: Aspectos Administrativos .....	98
6.01 Recursos.....	98
6.01.01 Recursos humanos.....	98
6.01.02 Recursos materiales.....	98
6.01.03 Recursos tecnológicos.....	99
6.01 Presupuesto.....	99
6.01 Cronograma.....	99
Capítulo VII: Conclusiones y recomendaciones .....	100
7.01 Conclusiones.....	100
7.02 Recomendaciones.....	100
A. Anexos .....	101
A.01 Matriz T.....	101
A.02 Matriz de requerimientos.....	102
A.03 Matriz de involucrados.....	103
A.04 Diagrama de caso de uso. Creación y administración de ticket.....	105
A.05 Diagrama de secuencia. Creación de ticket.....	106
A.06 Diagrama de secuencia. Administración de ticket.....	107
A.07 Diagrama de secuencia. Administración de activo.....	108
A.08 Diagrama de secuencia. Administración del sistema.....	109
A.09 Diagrama de clases.....	110
A.09 Diagrama lógico.....	111
A.10 Modelo Físico .....	112
A.11 Matriz de marco lógico .....	113
A.12 Cronograma.....	115
A.13 Manual de instalación .....	117
A.14 Manual de usuario de la aplicación informática.....	121

A.15 Manual técnico de la aplicación informática.....	143
A.16 Herramientas .....	183
1. Descripción de C#.....	183
2. Descripción de ASP.NET. ....	184
3. Descripción de JavaScript.....	184
4. Descripción de JQuery.....	185
5. Descripción de SQL Server.....	186
A.17 Enlaces .....	188
A.18 Bibliografía .....	187

## ÍNDICE DE TABLAS

1. Matriz T.....	101
2. Diseño de entrevista .....	6
3. Matriz de requerimientos .....	102
4. Descripción del requerimiento RF001 .....	7
5. Descripción del requerimiento RF002 .....	8
6. Descripción del requerimiento RF003 .....	9
7. Descripción del requerimiento RNF001 .....	10
8. Descripción del requerimiento RNF002 .....	11
9. Descripción del requerimiento RNF003 .....	12
10. Matriz de involucrados.....	103
11. Caso de uso, creación de ticket .....	19
12. Caso de uso, administración.....	20
13. Caso de uso, parametros.....	20
14. Caso de uso, activos .....	21
15. Caso de uso, seguridad.....	21
16. Caso de uso, creación de ticket .....	23
17. Caso de uso, administración.....	23
18. Caso de uso, parametros.....	24
19. Caso de uso, activos .....	24
20. Caso de uso, seguridad.....	25
21. Matriz de análisis de alternativas .....	26
22. Matriz de impacto de objetivos .....	27
23. Tipos de datos .....	28
24. Matriz de marco lógico .....	113
25. Prueba de unidad ManejadorArchivoAdjunto .....	58
26. Prueba de unidad ManejadorArchivoAdjunto 2 .....	58
27. Prueba de unidad ManejadorBitacora .....	58
28. Prueba de unidad ManejadorBitacora 2 .....	59
29. Prueba de unidad ManejadorBitacora 3 .....	59
30. Prueba de unidad ManejadorClave .....	59
31. Prueba de unidad ManejadorDepartamento .....	60
32. Prueba de unidad ManejadorDepartamento 2 .....	60

33. Prueba de unidad ManejadorDetalleTicket .....	60
34. Prueba de unidad ManejadorDetalleTicket 2 .....	61
35. Prueba de unidad ManejadorDetalleTicket 3 .....	61
36. Prueba de unidad ManejadorDetalleTicket 4 .....	61
37. Prueba de unidad ManejadorDispositivo .....	62
38. Prueba de unidad ManejadorDispositivo 2 .....	62
39. Prueba de unidad ManejadorEmpresa .....	62
40. Prueba de unidad ManejadorEmpresaServidor .....	63
41. Prueba de unidad ManejadorEmpresaServidor 2 .....	63
42. Prueba de unidad ManejadorEquipo .....	63
43. Prueba de unidad ManejadorEquipo 2 .....	64
44. Prueba de unidad ManejadorEstado .....	64
45. Prueba de unidad ManejadorEstado 2 .....	64
46. Prueba de unidad ManejadorImpresora .....	65
47. Prueba de unidad ManejadorImpresora 2 .....	65
48. Prueba de unidad ManejadorIp .....	65
49. Prueba de unidad ManejadorIp 2 .....	66
50. Prueba de unidad ManejadorModulo .....	66
51. Prueba de unidad ManejadorModulo 2 .....	66
52. Prueba de unidad ManejadorNumeroTelefono .....	67
53. Prueba de unidad ManejadorNumeroTelefono 2 .....	67
54. Prueba de unidad ManejadorObjeto .....	67
55. Prueba de unidad ManejadorObjeto 2 .....	68
56. Prueba de unidad ManejadorObjetoRequerimiento .....	68
57. Prueba de unidad ManejadorObjetoRequerimiento 2 .....	68
58. Prueba de unidad ManejadorObservacion .....	69
59. Prueba de unidad ManejadorObservacion 2 .....	69
60. Prueba de unidad ManejadorParametro .....	69
61. Prueba de unidad ManejadorParametro 2 .....	70
62. Prueba de unidad ManejadorPersonal .....	70
63. Prueba de unidad ManejadorPersonal 2 .....	70
64. Prueba de unidad ManejadorPersonal 3 .....	71
65. Prueba de unidad ManejadorPersonal 4 .....	71
66. Prueba de unidad ManejadorPersonalSubarea .....	71

67. Prueba de unidad ManejadorPersonalSubarea 2 .....	72
68. Prueba de unidad ManejadorPersonalTicket.....	72
69. Prueba de unidad ManejadorPersonalTicket 2.....	72
70. Prueba de aceptación EPA001 .....	73
71. Prueba de aceptación EPA002 .....	73
72. Prueba de aceptación EPA003 .....	74
73. Prueba de aceptación EPA004 .....	74
74. Prueba de aceptación EPA005 .....	75
75. Prueba de carga .....	75
76. Matriz de recursos humanos.....	98
77. Matriz de recursos materiales.....	98
78. Matriz de recursos materiales.....	99
79. Presupuesto .....	99

## ÍNDICE DE FIGURAS

1. Mapeo de involucrados .....	13
2. Arbol de problemas .....	14
3. Arbol de objetivos .....	16
4. Diagrama de caso de uso, administración .....	17
5. Diagrama de caso de uso, creación ticket .....	105
6. Diagrama de caso de uso, activos .....	18
7. Diagrama de caso de uso, seguridades .....	19
8. Diagrama de secuencia, administración sistema .....	22
9. Diagrama de secuencia, creación ticket .....	106
10. Diagrama de secuencia, administrar ticket .....	107
11. Diagrama de secuencia, activos .....	108
12. Diagrama de caso de uso, seguridades .....	109
13. Diagrama de clases .....	110
14. Modelo lógico .....	111
15. Modelo físico .....	112
16. Diagrama componentes .....	29
17. Diagrama de estrategia .....	30
18. Vista lógica .....	31
19. Vista física .....	32
20. Vista de desarrollo .....	32
21. Vista de procesos .....	33
22. Diseño de registro de usuarios .....	38
23. Diseño de página de muestra .....	39
24. Diseño de control ucMultiFileUpload .....	40
25. Diseño de registro de usuarios con errores .....	41
26. Diseño de creación de ticket, cliente .....	42
27. Diseño de creación de ticket, cliente .....	42
28. Diseño de creación de ticket, cliente .....	44
29. Diseño verificación de ticket .....	45
30. Diseño de verificación de datos específicos .....	46
31. Diseño de verificación de datos específicos .....	47

32. Diseño de verificación de datos específicos.....	48
33. Diseño de creación de ticket, direccionador .....	49
34. Diseño de creación de ticket, direccionador.....	50
35. Diseño de creación de ticket, direccionador .....	51
36. Diseño verificación de ticket.....	52
37. Diseño de verificación de datos específicos.....	53
38. Diseño de verificación de datos específicos.....	54
39. Diseño de administración de ticket .....	55
40. Diseño de administración de ticket .....	55
41. Diseño de administración de ticket .....	56
42. Diseño de mantenimiento empresa .....	57
43. Diseño de mantenimiento sucursal.....	57
44. Instalacion visual studio .....	76
45. Instalacion visual studio.....	77
46. Instalacion visual studio.....	78
47. Instalacion visual studio.....	78
48. Instalacion visual studio.....	79
49. Instalacion visual studio .....	79
50. Instalacion visual studio.....	80
51. Instalacion visual studio registro del producto 1.....	81
52. Instalacion visual studio registro del producto 2.....	81
53. Instalacion visual studio registro del producto 3.....	82
54. Instalacion visual studio registro del producto 4.....	83
55. Instalacion visual studio.....	84
56. Instalacion visual studio.....	85
57. Instalacion visual studio.....	85
58. Instalacion visual studio.....	85
59. Instalacion visual studio .....	86
60. Instalacion visual studio.....	86
61. Instalacion visual studio registro del producto 1.....	87
62. Instalacion visual studio registro del producto 2.....	87
63. Instalacion visual studio registro del producto 3.....	88
64. Instalacion visual studio registro del producto 4.....	88
65. Instalacion sql server.....	89

66. Instalacion sql server.....	89
67. Instalacion sql server.....	90
68. Instalacion sql server.....	91
69. Instalacion sql server.....	91
70. Instalacion sql server.....	92
71. Instalacion sql server.....	92
72. Instalacion sql server.....	93
73. Instalacion sql server.....	94
74. Instalacion sql server.....	95
75. Instalacion sql server.....	95
76. Instalacion sql server.....	96
77. Instalacion sql server.....	96
78. Instalacion sql server.....	97
79. Instalacion sql server.....	97
80. Cronograma.....	115

## RESUMEN EJECUTIVO

El control y administración de requerimientos se está realizando mediante una hoja de Excel con macros, la cual sirve como una herramienta de gestión. El usuario (cliente) tiene 2 medios para generar un ticket (vía correo electrónico y vía telefónica). La cuenta de correo y la extensión de mesa de ayuda son controladas por una persona a quien llamaremos direccionador, la misma que se encarga de llenar un pequeño formulario en la hoja de Excel, para después asignar mediante la misma a cualquiera de los diferentes funcionarios de soporte.

El desarrollo del presente proyecto quiere automatizar el registro de todos los procesos que se realiza en la mesa de ayuda, desde la creación del ticket hasta el cierre del mismo, controlando cada uno de los detalles del ticket.

El desarrollo de la aplicación web se realizara en el lenguaje de ASP.NET, y la información de la misma se guardara en una base de datos SQL SERVER, para lo cual nos ayudaremos de Entity Framework para el manejo de datos.

Se espera lograr un sistema robusto y completo que sirva como solución a varias empresas que se muevan en el mismo ámbito.

## Capítulo I: Antecedentes

### 1.01 Contexto

Es una empresa privada e innovadora cuyas actividades se centran en brindar soluciones tecnológicas a nivel nacional, tiene la infraestructura y los recursos para satisfacer la demanda en esta era global.

La empresa CITYTECH está ubicada en la provincia de Pichincha en la parte norte de la ciudad de Quito en el sector de la Carolina, su actividad principal gira alrededor de la tecnología, brindando soluciones de infraestructura y software a varias empresas, por lo cual al momento cuenta con 300 usuarios finales, brindando soporte y mantenimiento al hardware y software que usan para cumplir sus funciones.

La empresa cuenta con 3 áreas principales, mesa de ayuda, gestión funcional y desarrollo. Las cuales se complementan para cumplir con las incidencias que puedan producirse en las empresas clientes.

Para lograr medir el nivel de productividad de cada área se ha implementado una hoja de Excel, la misma que sirve para el registro, seguimiento y creación de estadísticas referente a los tickets. Esta herramienta esta compartida, lo que quiere decir que no pueden acceder dos personas al mismo tiempo sin correr el riesgo de perder información, por lo tanto ha existido perdida de tickets e incomodidad del usuario al tener la necesidad de llamar nuevamente para que su requerimiento sea atendido.

La mesa de ayuda cumple con las funciones de soporte técnico e implementación del hardware y software de las empresas. Cuenta con un direccionador, quien sería la persona encargada en primera instancia, de analizar y dirigir el ticket a la persona y área adecuada para su correcto seguimiento. En el área de gestión funcional se analiza los proyectos a largo plazo, se verifica su viabilidad y posibles soluciones para después

enviar los mismos al área correspondiente. El área de desarrollo se alimenta del área gestión funcional, ya que esta área aprueba y entrega un diseño de solución para que el personal de desarrollo pueda dar la solución ya aprobada por el usuario.

La gran demanda de soluciones innovadoras, modernas y de primer ha hecho que esta empresa evolucione para cumplir con estas necesidades, contando con una estructura empresarial muy sólida para mantener los niveles de atención.

### **1.02 Justificación**

La necesidad es tener muy en cuenta el impacto que tiene la informática en la empresa nos lleva a implementar un sistema de control de requerimientos de todo tipo, para lograr solventar la gran demanda que sufre la misma.

Por los motivos expuestos anteriormente, esta información debe tratarse con prudencia ya que los factores que afectan alguno de sus componentes pueden afectar, también, directa y proporcionalmente a la atención brindada al cliente.

Este sistema es muy importante, de esto depende el continuo funcionamiento de las diferentes áreas, por lo tanto la administración y organización del mismo debería ser uno de los puntos más importantes a tratarse mediante la aplicación a desarrollarse.

ITIL como metodología propone el establecimiento de estándares que nos ayuden en el control, operación y administración de los recursos (ya sean propios o de los clientes). Plantea hacer una revisión y reestructuración de los procesos existentes en caso de que estos lo necesiten (si el nivel de eficiencia es bajo o que haya una forma más eficiente de hacer las cosas), lo que nos lleva a una mejora continua, y para lo cual es necesario contar con parámetros de atención al cliente, con los cuales se podrá definir las deficiencias en el servicio y la manera en la que se podría mejorar.

Para lograr que ITIL sea una herramienta eficiente para la cantidad de usuarios finales que atenderá la empresa, es necesario su sistematización y automatización, la herramienta permitirá obtener datos estadísticos muy importantes para la toma de decisiones ya que se podrán reflejar datos como los tiempos de respuesta, niveles de satisfacción, etc. Esta información permitirá a la empresa conocer más a fondo su realidad y la de las empresas clientes en área tecnológica, lo cual es crucial para brindar soluciones innovadoras y personalizadas.

### **1.03 DEFINICIÓN DEL PROBLEMA (MATRIZ T)**

Véase Anexo A.01

#### **1.03.01 Conclusión tabla matriz T**

Al realizar la tabla anterior se puede evidenciar que un buen control y seguimiento de los tickets puede mejorar el tiempo de respuesta, lo cual significaría un beneficio y crecimiento económico para los dueños, y la seguridad de siempre contar con las herramientas necesarias para sus funciones por parte de los empleados.

## Capítulo II: Análisis de involucrados

### 2.01 Requerimientos

#### 2.01.1 Descripción del Sistema Actual

El control y administración de requerimientos se está realizando mediante una hoja de Excel con macros, la cual sirve como una herramienta de gestión. El usuario (cliente) tiene 2 medios para generar un ticket (vía correo electrónico y vía telefónica). La cuenta de correo y la extensión de mesa de ayuda son controladas por una persona a quien llamaremos direccionador, la misma que se encarga de llenar un pequeño formulario en la hoja de Excel, para después asignar mediante la misma a cualquiera de los diferentes funcionarios de soporte. Cabe recalcar que esta hoja de Excel envía un mail automáticamente al funcionario de soporte indicando la creación del ticket y toda la información que este contiene, pero no es una herramienta 100% confiable ya que han existido pérdida de correos en el mismo. Otra de las funciones del direccionador es identificar el tipo de requerimiento, ya que cada funcionario de soporte se especializa en solucionar cierto tipo de requerimientos, por lo tanto el direccionador debe saber qué tipo de incidencias resuelve cada funcionario de soporte para así poder asignar el ticket a la persona adecuada.

Cuando el usuario (cliente) desea obtener información de su requerimiento usa los medios disponibles, es decir el correo electrónico y el teléfono, el funcionario de soporte está en la obligación de informar al usuario final sobre el progreso y estado de su ticket, si no se cuenta con el número de ticket se hace una búsqueda manual en la hoja de Excel. Si el funcionario de soporte desea cerrar un ticket deberá mandar un mensaje de confirmación a la mesa de ayuda para que la misma mediante la hoja de Excel cierre el ticket en el registro.

Se hace seguimiento de los tickets pendientes al final de todos los días laborales, es decir que se envía una lista de los tickets pendientes a los funcionarios para que los mismos puedan verificar el estado de cada uno, los tickets que no sean cerrados se debe

### **2.01.2 Visión y Alcance**

La aplicación deberá administrar y llevar un registro ordenado y bien clasificado de tickets, alrededor de esta información gira el proceso principal del negocio por lo tanto el correcto uso y referencia del mismo es esencial en el proceso que vamos a automatizar. Se deberá generar reportes de forma fácil y ágil, utilizando métodos de filtrado para poder obtener solo la información requerida.

Se espera que el software mantenga un canal de comunicación constante entre el usuario final y el personal de soporte o desarrollo, así como entre las personas involucradas en el soporte y en el proyecto establecido. Cuando el usuario ingrese un requerimiento, el software deberá informar al personal de soporte automáticamente. Se deberá poder parametrizar los estados y etapas en las que podrá encontrarse un ticket, siempre llevando un registro temporal de cada movimiento desde el momento que se genera el incidente hasta el momento que se soluciona, guardando información detallada del mismo, para más adelante poder generar gráficos estadísticos con la posibilidad de filtrar por tipo de requerimiento, usuario, personal de soporte, personal de desarrollo, tipo de solución, empresa, activo al que se dará soporte y proyectos de desarrollo.

Se gestionara los activos de la empresa mediante un módulo especializado para esto, en el cual se manejara los inventarios de hardware y software disponible. Se podrá definir la infraestructura de cada empresa, desde los equipos de oficina usados,

tomando en cuenta la estructura de la red, hasta el software que se instalara en las computadoras usadas por las empresas, se deberá asignar cada equipo a uno o varios empleados, así el usuario tendrá esa sensación de personalización al momento de generar un ticket. El software tendrá tener un módulo de seguridad, donde definir las políticas para la creación de usuarios, el manejo de las claves y los roles y permisos que tendrá cada usuario. Se mantendrá un registro de actividades de cada usuario, al mismo lo llamaremos módulo de auditoria, se espera que la seguridad de la información haya aumentado considerablemente.

### 2.01.2 Entrevista

Tabla 2.

*Diseño entrevista.*

Preguntas	Objetivos	Análisis posterior
¿Cuáles son las problemáticas principales que desea solventar mediante el sistema?	Definir los puntos débiles en el proceso y así poder mejorarlos.	Se logró identificar las mayores falencias del proceso.
¿Cuál es la información que debería tener un ticket?	Conocer cuales datos son indispensables para un ticket.	Logramos definir los datos que son o no indispensables.
¿Cuál sería el aspecto más importante a tomar en cuenta en el sistema?	Identificar el core del negocio, alrededor del cual girara toda la información y el software.	Identificamos al ticket como la clase o entidad más importante del sistema.
¿Qué usuarios deberían tener acceso al sistema?	Identificar los roles y sus actividades dentro de la empresa.	Los roles en la empresa están bien definidos, lo cual será esencial en el sistema.

### 2.01.4 Matriz de Requerimientos

Véase anexo A.02

## 2.01.5 Descripción Detallada

Tabla 4.

### Descripción del requerimiento RF001

Descripción del requerimiento		Estado:	Revisión.
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014
<b>Identificador</b>	RF001		
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	Nombre del emperrado. Nombre del Requerimiento. Objeto u objetos relacionados al ticket.		
<b>Descripción</b>	La aplicación deberá administrar tickets y cada una de las actividades o etapas que se debe realizar para llegar a la solución del mismo, tomando en cuenta los tiempos de atención, el personal que lo realiza y cualquier observación que pueda realizar el mismo, además de controlar parámetros fundamentales como (estado, tipo de requerimiento, ticket padre, autorización, fecha de inicio , fecha final, usuarios que intervienen)		
<b>Datos de Salida</b>	Reporte de creación de ticket. Correo Electrónico de confirmación.		
<b>Resultados Esperados</b>	Un reporte ordenado y flexible, que permita filtrar por uno o varios parámetros.		
<b>Origen</b>	Ana Játiva Miguel Muñoz Jonathan Ortiz		
<b>Dirigido a</b>	Funcionarios de soporte Usuarios (cliente) Personal administrativo		
<b>Prioridad</b>	5		
<b>Requerimientos asociados</b>	Ninguno		
ESPECIFICACIÓN			
<b>Precondiciones</b>	El usuario deberá estar registrado en el sistema. Se deberá generar el cuestionario antes de crear un ticket.		
<b>Poscondiciones</b>	El usuario podrá recuperar la información del ticket de ser necesario.		
<b>Criterios de aceptación</b>	El ticket tendrá que crearse de forma rápida y fácil. Solo se deberá preguntar al usuario lo indispensable.		

Tabla 5.

*Descripción del requerimiento RF002*

Descripción del requerimiento		Estado:	Revisión.
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014
<b>Identificador</b>	RF002		
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	Requerimiento. Tipo de ciclo. Día en el que se va a generar el ticket.		
<b>Descripción</b>	El software deberá tener la capacidad de generar tareas programadas, las mismas que se podrá definir cada cuanto tiempo esta tarea generara un ticket (diario, semanal, mensual) y bajo qué parámetros se lo hará.		
<b>Datos de Salida</b>	Generación automática de tickets.		
<b>Resultados Esperados</b>	Automatizar ciertos procesos que se deben cumplir cada cierto tiempo.		
<b>Origen</b>	Ana Játiva Roberto Huacanes		
<b>Dirigido a</b>	Funcionarios de soporte Personal administrativo		
<b>Prioridad</b>	3		
<b>Requerimientos asociados</b>	Ninguno		
ESPECIFICACIÓN			
<b>Precondiciones</b>	El usuario deberá estar registrado en el sistema como administrador.		
<b>Poscondiciones</b>	Las tareas programadas generadas solo podrán ser cambiadas por un administrador del sistema.		
<b>Criterios de Aceptación</b>	Las tareas deberán generarse de forma automática y correcta, sin necesidad de dar seguimiento.		

Tabla 6.

*Descripción del requerimiento RF003*

Descripción del requerimiento		Estado:	Revisión.
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014
<b>Identificador</b>	RF003		
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de requerimiento:</b>	Funcional
<b>Datos de Entrada</b>	Opción de direccionamiento. Parametrización de direccionamiento automático. Área y etapa de ser necesario.		
<b>Descripción</b>	La aplicación tendrá dos formas de direccionar los requerimientos, la primera es manual, donde el direccionador deberá fijarse a que área deberá asignar y hacer este proceso. La segunda opción será direccionamiento automático, en este caso el software identificara a que subárea pertenece e identificara a los usuarios con menos tickets pendientes encargados de esta subárea, cuando se allá identificado dicho usuario se le asignara el ticket.		
<b>Datos de Salida</b>	Correo electrónico a los usuarios implicados para confirmar la creación del ticket.		
<b>Resultados Esperados</b>	El software deberá reconocer que tipo de requerimiento está asignado al ticket, con esto se direccionara automáticamente al usuario de soporte ya predefinidos.		
<b>Origen</b>	Ana Játiva Carlos Lagos		
<b>Dirigido a</b>	Direccionador. Usuario de Soporte.		
<b>Prioridad</b>	4		
<b>Requerimientos asociados</b>	Ninguno		
ESPECIFICACIÓN			
<b>Precondiciones</b>	Se deberá crear un ticket. El usuario deberá estar registrado con cualquier rol. Todos los requerimientos deben tener un perfil en el que se definirán los usuarios que podrán ser asignados a dicho ticket.		
<b>Poscondiciones</b>	El perfil de direccionamiento automático solamente podrá ser modificado por un administrador del sistema.		
<b>Criterios de Aceptación</b>	Se deberá direccionar correctamente, reconociendo el tipo de ticket para saber si es necesario autorización o no. Los correos deberán tener confirmación de envío.		

Tabla 7.

*Descripción del requerimiento RNF001*

Descripción del requerimiento		Estado:		Revisión.	
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera		
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014		
<b>Identificador</b>	RNF001				
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de Requerimiento</b>	Funcional		
<b>Datos de Entrada</b>	Parámetro de filtrado. Tipo de Reporte.				
<b>Descripción</b>	El software tendrá la capacidad de generar un reporte consolidado, donde se reflejara todos los tickets generados por área, es decir gestión funcional y mesa de ayuda. Dicho reporte podrá ser filtrado por los siguientes parámetros: Fecha de inicio, fecha final, usuario que genera el ticket, usuario que autoriza, tipo de requerimiento, objeto al que se dará soporte, estado, etapa o área, prioridad, ticket padre, usuario asignado para dar soporte, tiempo de respuesta,(en el caso que el ticket este cerrado).				
<b>Datos de Salida</b>	Reporte impreso. Reporte lógico.				
<b>Resultados Esperados</b>	El software deberá				
<b>Origen</b>	Miguel Muñoz Ana Játiva				
<b>Dirigido a</b>	Administrador del sistema. Directores corporativos.				
<b>Prioridad</b>	4				
<b>Requerimientos asociados</b>	Ninguno				
ESPECIFICACIÓN					
<b>Precondiciones</b>	El usuario deberá estar registrado en el sistema como administrador.				
<b>Poscondiciones</b>	Se podrá utilizar este registro como respaldo físico de los tickets generados.				
<b>Criterios de Aceptación</b>	Se deberá poder filtrar con cualquiera de sus parámetros. El reporte deberá mantener la información ordenada y fiable. Con un diseño agradable y practico				

Tabla 8.

*Descripción del requerimiento RNF002*

Descripción del requerimiento		Estado:	Revisión.
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014
<b>Identificador</b>	RNF002		
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	Parámetros de filtrado.		
<b>Descripción</b>	La capacidad de generar datos estadísticos de los tickets, esto nos servirá para saber cuánto se demora cada usuario en atender un ticket en promedio, el tiempo que los tickets permanecen en cada etapa y estado y la satisfacción del cliente en promedio, este reporte podrá ser filtrado por el tipo de requerimiento, por el tipo de solución, usuario que lo genera, usuario que realiza el soporte, objeto al que se asigna el ticket.		
<b>Datos de Salida</b>	Reporte estadístico físico. Reporte estadístico lógico.		
<b>Resultados Esperados</b>	Se espera un reporte estadístico muy fácil de entender para los administradores del sistema.		
<b>Origen</b>	Carlos Lagos Miguel Muñoz		
<b>Dirigido a</b>	Administrador del sistema. Administrador de proceso.		
<b>Prioridad</b>	4		
<b>Requerimientos asociados</b>	RNF001		
ESPECIFICACIÓN			
<b>Precondiciones</b>	El usuario deberá estar registrado en el sistema como administrador.		
<b>Poscondiciones</b>	Se podrá obtener los resultados estadísticos en cualquier momento.		
<b>Criterios de Aceptación</b>	El reporte debe ser entendible y exacto, con un diseño amigable para el usuario.		

Tabla 9.

*Descripción del requerimiento RNF003*

Descripción del requerimiento		Estado:	Revisión.
<b>Creado por:</b>	Rubén Mosquera	<b>Actualizado por:</b>	Rubén Mosquera
<b>Fecha de creación:</b>	09-06-2014	<b>Fecha de actualización:</b>	18-06-2014
<b>Identificador</b>	RNF003		
<b>Tipo de Requerimiento</b>	Critico	<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	Opción de integración.		
<b>Descripción</b>	El software deberá gestionar los activos físicos y lógicos de la empresa en forma de objetos, tomando en cuenta los siguientes: equipo (Pc, Laptop), impresora (tónor), teléfono, servidor, dispositivo (teléfono móvil, tablet, biométrico, etc.), software.		
<b>Datos de Salida</b>	Se obtendrá la tabla de equipo, impresora, teléfono, servidor, dispositivo, software.		
<b>Resultados Esperados</b>	Deberá integrarse mediante web service a una aplicación interna que administra los activos fijos de la empresa.		
<b>Origen</b>	Jonathan Ortiz Ana Játiva		
<b>Dirigido a</b>	Usuario (cliente) Personal de soporte. Administradores.		
<b>Prioridad</b>	4		
<b>Requerimientos asociados</b>	Ninguno		
ESPECIFICACIÓN			
<b>Precondiciones</b>	El usuario deberá estar registrado en el sistema como administrador del sistema.		
<b>Poscondiciones</b>	Se podrá acceder a todos los activos de la empresa actualizados.		
<b>Criterios de Aceptación</b>	El software deberá poder actualizar los activos en cualquier momento, esto se deberá hacer de forma rápida y fácil para evitar pérdidas de información.		

## 2.02 Mapeo de involucrados



Figura 1. Gráfico del mapeo de involucrados. En este grafico se muestra los principales actores en el proceso.

## 2.03 Matriz de involucrados

Véase Anexo A.03

### Capítulo III: Problemas y objetivos

#### 3.01 Árbol de problemas

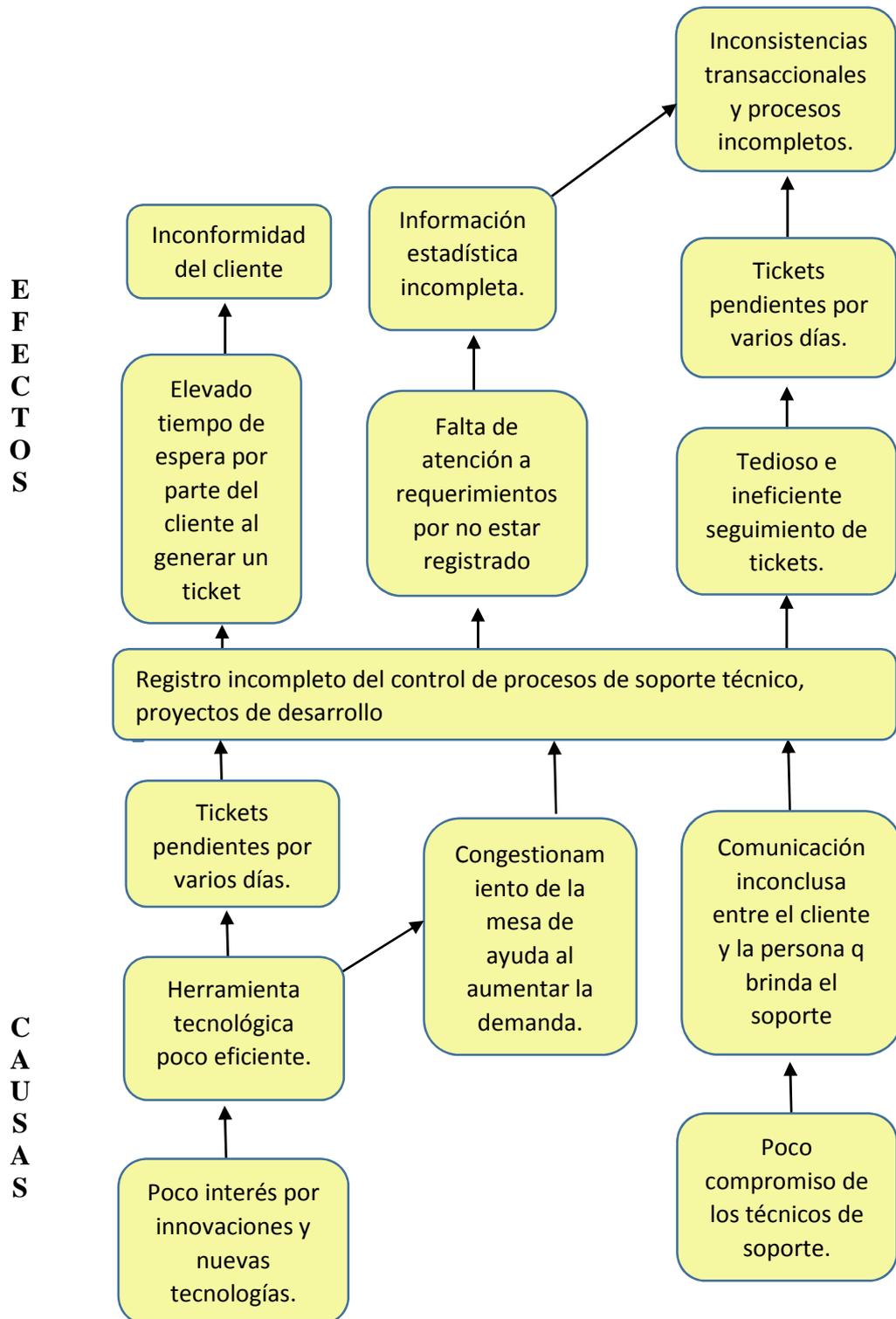


Figura 2. Gráfico del árbol de problemas. Esta figura muestra el principal problema sus causas y efectos en la mesa de ayuda de la empresa Citytech.

### **3.01.1 Análisis del árbol de problemas**

Para el análisis del árbol de problemas debemos tomar en cuenta que la insatisfacción del cliente es el principal punto a tomar en cuenta en nuestro proyecto. Una de las principales causas de este problema es la herramienta tecnológica poco eficiente con la que cuenta la empresa, esto ha ocasionado pérdida de información y por ende atrasos en el servicio. Otra causa es el congestionamiento de la mesa de ayuda como principal canal para la creación de tickets, lo que ocasiona tiempos de espera innecesarios para el cliente, además el poco compromiso de los técnicos de soporte para mantener un registro ordenado y actualizado en todo momento.

Todo esto ocasiona un elevado tiempo de espera por parte del cliente al generar un ticket, registros de tickets incompletos o inexistentes, información estadística incompleta.

Este análisis nos permitirá tener una idea más clara de la situación real de la empresa, además de generar una visión compartida de la problemática principal sus causas y sus efectos.

### 3.02 Árbol de objetivos

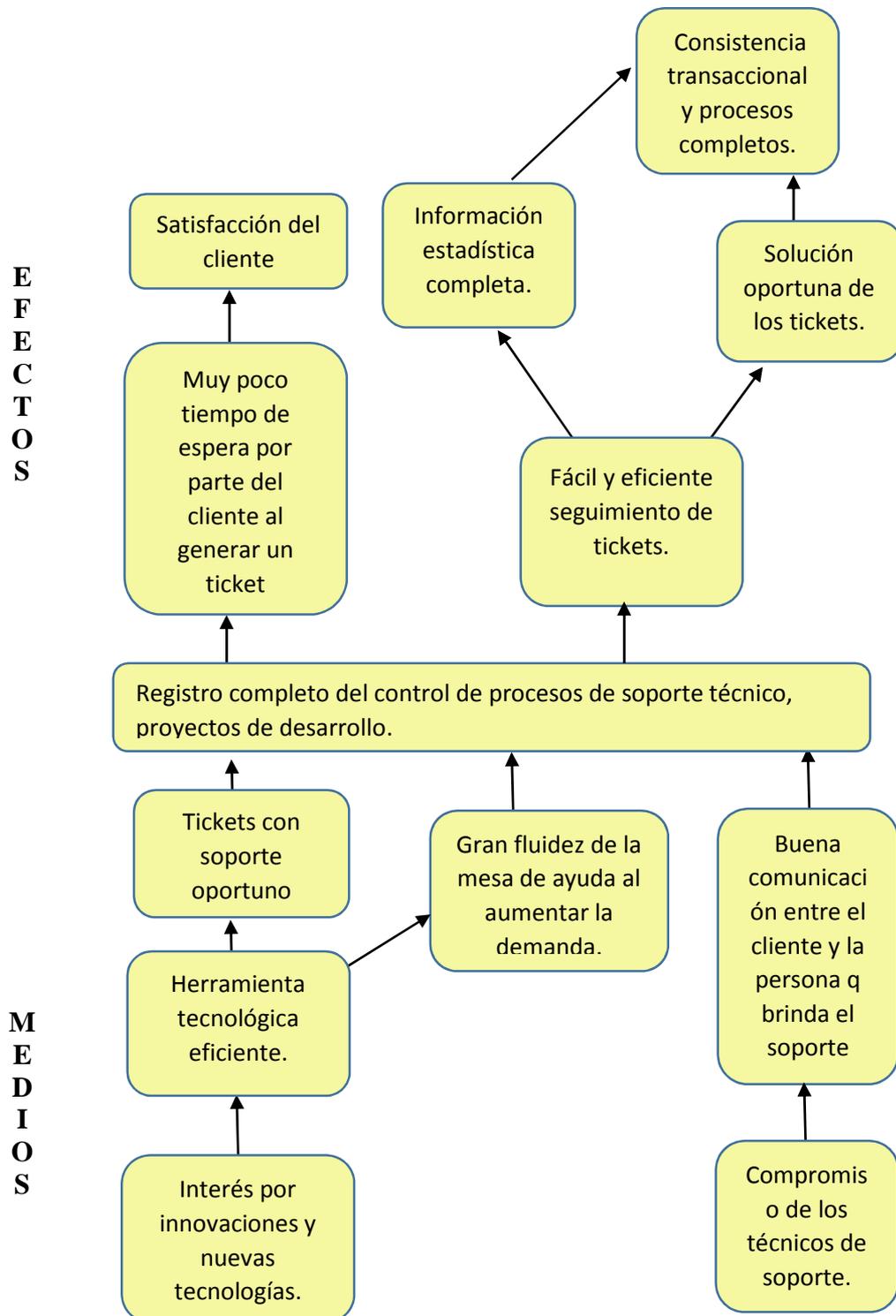


Figura 3. Gráfico del árbol de objetivos. Esta figura muestra el principal objetivo sus causas y efectos en la mesa de ayuda de la empresa Citytech.



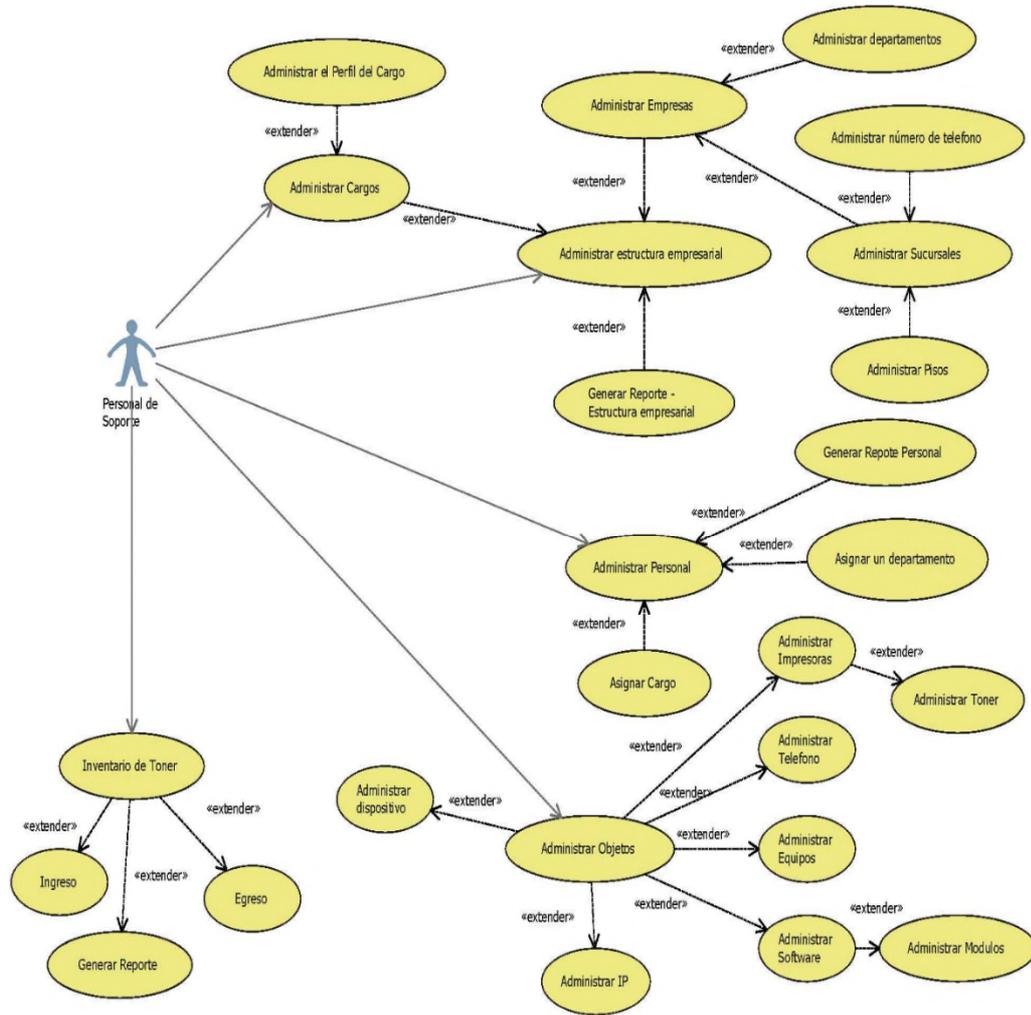


Figura 6. Diagrama de caso de uso. Administración de activos.

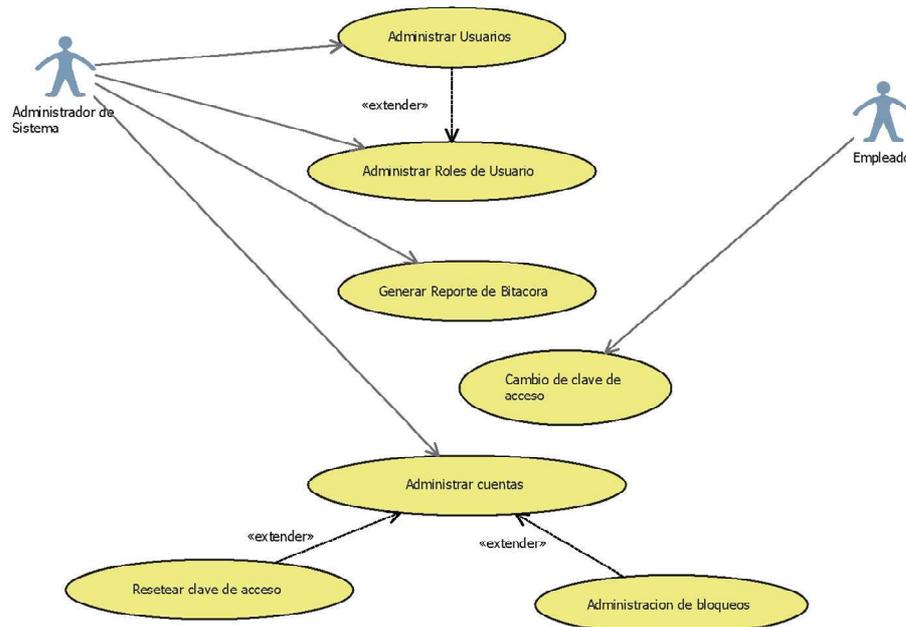


Figura 7. Diagrama de caso de uso. Administración de seguridades

### 3.04 Casos de uso de realización

Tabla 11.

Caso de uso de realización. Creación de tickets.

CASO DE USO DE REALIZACIÓN	
<b>Nombre</b>	Creación de tickets.
<b>Identificador</b>	C.U. 1
<b>Responsabilidades</b>	Creación de tickets.
<b>Tipo</b>	Sistema
<b>Referencias Caso de uso</b>	Caso de uso de creación de tickets.
<b>Referencias Requisitos</b>	Usuario Registrado
<b>PRECONDICIONES</b>	
De Instancia	
<ol style="list-style-type: none"> <li>1. El cliente o usuario deberá acceder al sistema como cliente.</li> <li>2. Seleccionar el tipo de incidencia o requerimiento.</li> <li>3. Tener tickets creados para poder administrarlos.</li> </ol>	
De relación	
Ninguno.	
<b>POSCONDICIONES</b>	
De instancia	
<ol style="list-style-type: none"> <li>1. El ticket generado podrá tener características personalizadas dependiendo de su tipo.</li> </ol>	
De relación	
<ol style="list-style-type: none"> <li>1. Después del registro se podrá realizar el seguimiento y administración del mismo</li> </ol>	
<b>SALIDAS DE PANTALLA</b>	
<ol style="list-style-type: none"> <li>1. El usuario no llena todos los campos requeridos.</li> <li>2. Para proceder acceder al detalle del ticket se deberá ingresar el código del mismo.</li> </ol>	

Tabla 12.

*Caso de uso de realización. Administración de tickets.*

<b>CASO DE USO DE REALIZACIÓN</b>	
<b>Nombre</b>	Administración de tickets.
<b>Identificador</b>	C.U. 2
<b>Responsabilidades</b>	Administración de tickets.
<b>Tipo</b>	Sistema
<b>Referencias Caso de uso</b>	Caso de uso de administración de tickets.
<b>Referencias Requisitos</b>	Usuario Registrado como personal de soporte.
<b>PRECONDICIONES</b>	
De Instancia	
	<ol style="list-style-type: none"> <li>4. El cliente o usuario deberá acceder al sistema como personal de soporte.</li> <li>5. Tener asignado cualquier ticket.</li> <li>6. Tener autorizado el ticket.</li> </ol>
De relación	
	Ninguno.
<b>POSCONDICIONES</b>	
De instancia	
	<ol style="list-style-type: none"> <li>2. Se podrá re direccionar el ticket a cualquier área.</li> <li>3. Toda acción realizada será registrado con hora y fecha.</li> </ol>
De relación	
	<ol style="list-style-type: none"> <li>2. Después de la administración se podrá generar cuadros estadísticos.</li> </ol>
<b>SALIDAS DE PANTALLA</b>	
	<ol style="list-style-type: none"> <li>3. El usuario no llena todos los campos requeridos.</li> <li>4. Tratar de ingresar a un ticket sin autorización.</li> </ol>

Tabla 13.

*Caso de uso de realización. Administración de parámetros.*

<b>CASO DE USO DE REALIZACIÓN</b>	
<b>Nombre</b>	Administración de parámetros.
<b>Identificador</b>	C.U. 3
<b>Responsabilidades</b>	Administración de parámetros.
<b>Tipo</b>	Sistema
<b>Referencias Caso de uso</b>	Caso de uso de administración de parámetros.
<b>Referencias Requisitos</b>	Usuario registrado como Administrador.
<b>PRECONDICIONES</b>	
De Instancia	
	<ol style="list-style-type: none"> <li>7. El usuario deberá ingresar como Administrador.</li> </ol>
De relación	
	Ninguno.
<b>POSCONDICIONES</b>	
De instancia	
	<ol style="list-style-type: none"> <li>4. Se guardara registro de todos los cambios realizados en los parámetros.</li> </ol>
De relación	
	Ninguno.
<b>SALIDAS DE PANTALLA</b>	
	<ol style="list-style-type: none"> <li>5. El usuario no llena todos los campos requeridos.</li> <li>6. Cuando se quiera cambiar parámetros protegidos.</li> </ol>

Tabla 14.

*Caso de uso de realización. Administración de activos.*

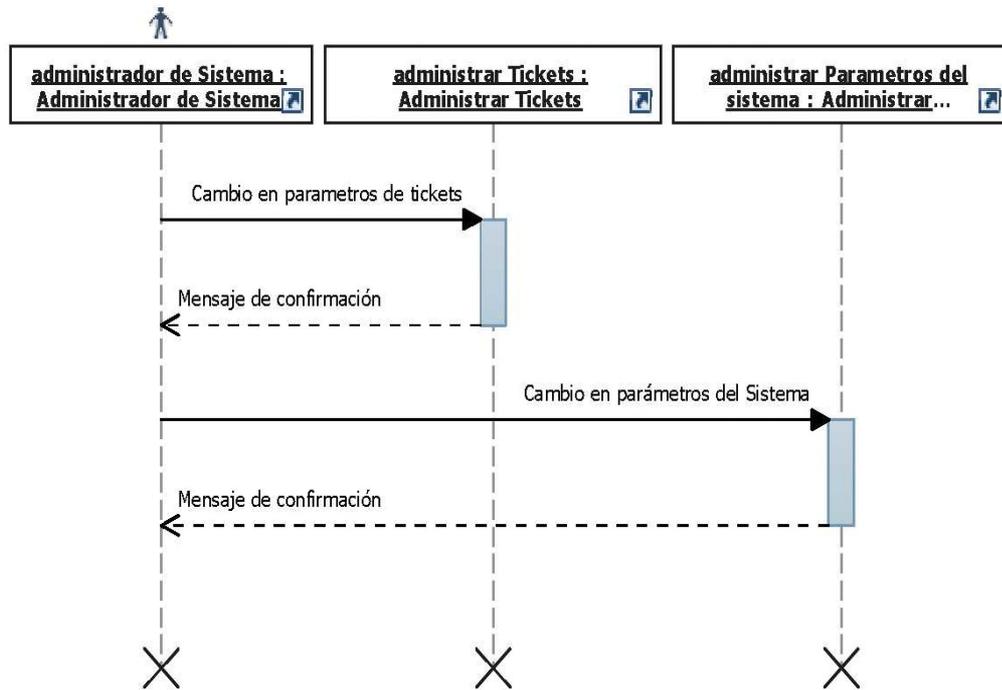
<b>CASO DE USO DE REALIZACIÓN</b>	
<b>Nombre</b>	Administración de activos.
<b>Identificador</b>	C.U. 4
<b>Responsabilidades</b>	Administración de activos.
<b>Tipo</b>	Sistema
<b>Referencias Caso de uso</b>	Caso de uso de administración de activos.
<b>Referencias Requisitos</b>	Usuario registrado como personal de soporte.
<b>PRECONDICIONES</b>	
De Instancia	8. El usuario deberá estar registrado como personal de soporte.
De relación	Ninguno.
<b>POSCONDICIONES</b>	
De instancia	5. Se guardara registro de todos los cambios realizados en los activos.
De relación	Ninguno.
<b>SALIDAS DE PANTALLA</b>	
	7. El usuario no llena todos los campos requeridos.

Tabla 15.

*Caso de uso de realización. Administración de seguridades.*

<b>CASO DE USO DE REALIZACIÓN</b>	
<b>Nombre</b>	Administración de seguridades.
<b>Identificador</b>	C.U. 5
<b>Responsabilidades</b>	Administración de seguridades.
<b>Tipo</b>	Sistema
<b>Referencias Caso de uso</b>	Caso de uso de administración de seguridades.
<b>Referencias Requisitos</b>	Usuario registrado.
<b>PRECONDICIONES</b>	
De Instancia	9. Cualquier usuario registrado podrá cambiar su clave de acceso. 10. Registrarse como administrador para resetear claves de acceso y administrar usuarios.
De relación	Ninguno.
<b>POSCONDICIONES</b>	
De instancia	6. Se guardara registro de todos los cambios realizados en los usuarios del sistema.
De relación	Ninguno.
<b>SALIDAS DE PANTALLA</b>	
	8. El usuario no llena todos los campos requeridos.

### 3.05 Diagramas de secuencia del sistema



**Figura 8.** Diagrama de secuencia. Administración de parámetros.

Véase anexo A.05

Véase anexo A.06

Véase anexo A.07

Véase anexo A.08

### 3.05 Especificación de casos de uso

Tabla 16.

*Especificaciones de Caso de Uso. Creación de Ticket.*

ESPECIFICACIONES DE CASO DE USO	
Casos de Uso	Creación de Ticket.
Identificador	C.U. 1
CURSO TÍPICO DE EVENTOS	
USUARIO	SISTEMA
Usuario registrado en el sistema	<p>El caso de uso empieza cuando el usuario ingresa al sistema.</p> <p>Se debe ingresar a la opción de Crear Ticket.</p> <p>El usuario deberá seleccionar el tipo de requerimiento que se va a crear.</p> <p>Se generara el cuestionario asignado al tipo de requerimiento que se escogió.</p> <p>El usuario deberá llenar las preguntas requeridas.</p> <p>El caso de uso termina.</p>
CURSOS ALTERNATIVOS	
<ol style="list-style-type: none"> <li>1. El usuario en cualquier momento podrá salir del sistema.</li> <li>2. El caso de uso termina.</li> </ol>	

Tabla 17.

*Especificaciones de Caso de Uso. Administración de Ticket.*

ESPECIFICACIONES DE CASO DE USO	
Casos de Uso	Administración de Ticket.
Identificador	C.U. 2
CURSO TÍPICO DE EVENTOS	
USUARIO	SISTEMA
Usuario registrado en el sistema como Administrador	<p>El caso de uso empieza cuando el usuario ingresa al sistema como personal de soporte o desarrollo.</p> <p>Se debe ingresar a la opción de Administración de Ticket.</p> <p>El usuario deberá seleccionar el ticket que se va a administrar.</p> <p>Se podrá cambiar las opciones del ticket, además de adjuntar archivos y observaciones.</p> <p>El usuario deberá guardar el ticket después de cambiar sus propiedades.</p> <p>El caso de uso termina.</p>
CURSOS ALTERNATIVOS	
<ol style="list-style-type: none"> <li>1. El usuario en cualquier momento podrá salir del sistema.</li> <li>2. El caso de uso termina.</li> </ol>	

Tabla 18.

*Especificaciones de Caso de Uso. Administración de parámetros.*

<b>ESPECIFICACIONES DE CASO DE USO</b>	
Casos de Uso	Administración de parámetros.
Identificador	C.U. 3
<b>CURSO TÍPICO DE EVENTOS</b>	
USUARIO	SISTEMA
Usuario registrado en el sistema como Administrador	<p>El caso de uso empieza cuando el usuario ingresa al sistema como administrador.</p> <p>Se debe ingresar a la opción de Administración de parámetros.</p> <p>Seleccionar el parámetro que se va a cambiar para conocer sus opciones.</p> <p>Guardar después de cambiar alguna opción.</p> <p>El caso de uso termina</p>
<b>CURSOS ALTERNATIVOS</b>	
<ol style="list-style-type: none"> <li>1. El usuario en cualquier momento podrá salir del sistema.</li> <li>2. El caso de uso termina.</li> </ol>	

Tabla 19.

*Especificaciones de Caso de Uso. Administración de activos.*

<b>ESPECIFICACIONES DE CASO DE USO</b>	
Casos de Uso	Administración de activos.
Identificador	C.U. 4
<b>CURSO TÍPICO DE EVENTOS</b>	
USUARIO	SISTEMA
Usuario registrado en el sistema como Administrador	<p>El caso de uso empieza cuando el usuario ingresa al sistema como administrador o personal de soporte.</p> <p>Se debe ingresar a la opción de Administración de activos.</p> <p>Seleccionar el activo que se va a cambiar para conocer sus opciones.</p> <p>Guardar después de cambiar alguna opción.</p> <p>El caso de uso termina.</p>
<b>CURSOS ALTERNATIVOS</b>	
<ol style="list-style-type: none"> <li>1. El usuario en cualquier momento podrá salir del sistema.</li> <li>2. El caso de uso termina.</li> </ol>	

Tabla 20.

*Especificaciones de Caso de Uso. Administración de seguridades.*

ESPECIFICACIONES DE CASO DE USO	
Casos de Uso	Administración de seguridades.
Identificador	C.U. 5
CURSO TÍPICO DE EVENTOS	
USUARIO	SISTEMA
Usuario registrado en el sistema como Administrador	<p>El caso de uso empieza cuando el usuario ingresa al sistema como Administrador.</p> <p>Se debe ingresar a la opción de Administración de seguridad.</p> <p>Seleccionar el usuario que se va a cambiar para conocer sus opciones.</p> <p>Guardar después de cambiar alguna opción como contraseña, información personal.</p> <p>El caso de uso termina.</p>
Usuario registrado	<p>El caso de uso empieza cuando el usuario ingresa al sistema.</p> <p>Se ingresa a las opciones de usuario.</p> <p>Se deberá volver a ingresar la clave por seguridad.</p> <p>Cambiar la clave y guardar-</p> <p>El caso de uso termina.</p>
CURSOS ALTERNATIVOS	
<ol style="list-style-type: none"> <li>1. El usuario en cualquier momento podrá salir del sistema.</li> <li>2. El caso de uso termina.</li> </ol>	

## Capítulo IV: Análisis de alternativas.

### 4.01 Matriz de análisis de alternativas

**Tabla 21.**

Matriz de análisis de alternativas.

Objetivos	Impacto sobre el propósito	Factibilidad técnica	Factibilidad financiera	Factibilidad social	Factibilidad política	Total	Categorías
1. Tickets con soporte oportuno al tener una herramienta tecnológica eficiente por innovaciones y nuevas tecnologías.	4	4	3	3	3	17	Alta
2. Gran fluidez de la mesa de ayuda al aumentar la demanda al tener una herramienta tecnológica eficiente.	3	3	2	2	3	13	Media Alta
3. Buena comunicación entre el cliente y la persona que brinda el soporte gracias a compromiso de los técnicos de soporte.	4	3	3	1	1	12	Media
4. Satisfacción del cliente por el poco tiempo de espera al generar un ticket.	3	3	4	3	4	17	Alta
5. Consistencia transaccional y procesos completos por solución oportuna de los tickets al tener una fácil y eficiente seguimiento de tickets.	3	3	4	1	2	13	Media Alta
<b>Total</b>	<b>17</b>	<b>16</b>	<b>16</b>	<b>10</b>	<b>13</b>	<b>72</b>	

#### 4.02 Matriz de impactos de objetivos

**Tabla 22.**

Matriz de impactos de objetivos.

	Factibilidad de lograrse	Impacto en Genero	Impacto Ambiental	Relevancia	Sostenibilidad	Total
<b>O B J E T I V O S</b>	* El tiempo de respuesta permite aumentar la productividad a los mismos costos * Cuenta con financiamiento * Existe tecnología adecuada para su realización	<b>1.</b> Promueve el trato igualitario sin importar el género. * Fortalece la inclusión de la mujer en el ambiente laboral	* Mejora el entorno laboral. * Mejora el entorno social. * Protege el uso de los recursos.	Responde a las expectativas de los beneficiarios Es una prioridad sentida por los beneficiarios	* Aumenta la participación del cliente en el seguimiento de sus tickets. * La empresa estará en capacidad de afrontar mayor número de requerimientos * Se puede conseguir financiamiento a futuro	

#### 4.03. Estándares para el diseño de clases

Para el análisis de la base de datos se ha usado el modelamiento mediante (Visual Studio) tomando en cuenta las necesidades de la empresa.

Las clases se nombraron con un estándar, serán sustantivos en singular con mayúsculas, los tres primeros caracteres serán (TBL) seguido de guion bajo y el nombre de la tabla. En el caso de ser una tabla para romper una relación de muchos a muchos el nombre tendrá el mismo estándar, seguido del nombre de las dos tablas relacionadas sin ninguna separación.

Los atributos de una clase se nombraran con el siguiente estándar, los tres primeros caracteres harán referencia al nombre de la tabla, seguido de un guion bajo y el nombre del campo, los campos se especificaran en orden natural, siendo el primer campo el

campo de llave primaria, la misma que llevara el acrónimo (ID), este campo tendrá la propiedad de ser auto numérico, seguirán los campos naturales de la tabla finalizando con tres campos de control, el primero se llamara (MODFECHA) con el estándar ya mencionado, este campo guardara la fecha que se modificó por última vez esta tabla, el segundo campo se llamara (MODUSUARIO) el mismo que guardara el código del ultimo usuario que modifiko la tabla y el tercer campo será (ESTADO), el mismo que servirá para realizar el borrado lógico del registro. Los nombres de los campos harán referencia a su uso, o a la información que van a almacenar, teniendo nombres cortos y fáciles de entender.

Los tipos de datos que vamos a usar en este modelo serán los siguientes.

**Tabla 23.**

Matriz de tipo de datos.

Tipo SQL	Uso
Int	Para llaves y contadores.
Varchar	Para guardar los nombres, descripciones y para campos de referencia.
Datetime	Para guardar todas las fechas del sistema.
Bit	Para guardar estados de ciertas propiedades de las tablas.

Las asociaciones se nombraran con las dos primeros caracteres (FK) seguido de guion bajo, el nombre de la tabla padre nuevamente guion bajo y el nombre de la tabla hija, todo será en mayúsculas.

El índice se nombrara con los tres primeros caracteres (IDE) seguido de guion bajo y el nombre de la tabla a la que hace referencia, todo esto en mayúsculas. El uso del valor (NULL) se trata de evitar en lo posible, tomando las necesidades y realidad de la empresa.

Para el respaldo respectivo de la base de datos se usará la herramienta nativa de (SQL Server), la cual nos permitirá generar un archivo (.bak) el cual nombraremos como la base de datos. Este archivo se guardará en una carpeta encriptada.

#### 4.04. Diagrama de clases

Véase anexo A.09

#### 4.05. Modelo lógico – físico

Véase anexo A.10

Véase anexo A.11

#### 4.06. Diagrama de componentes

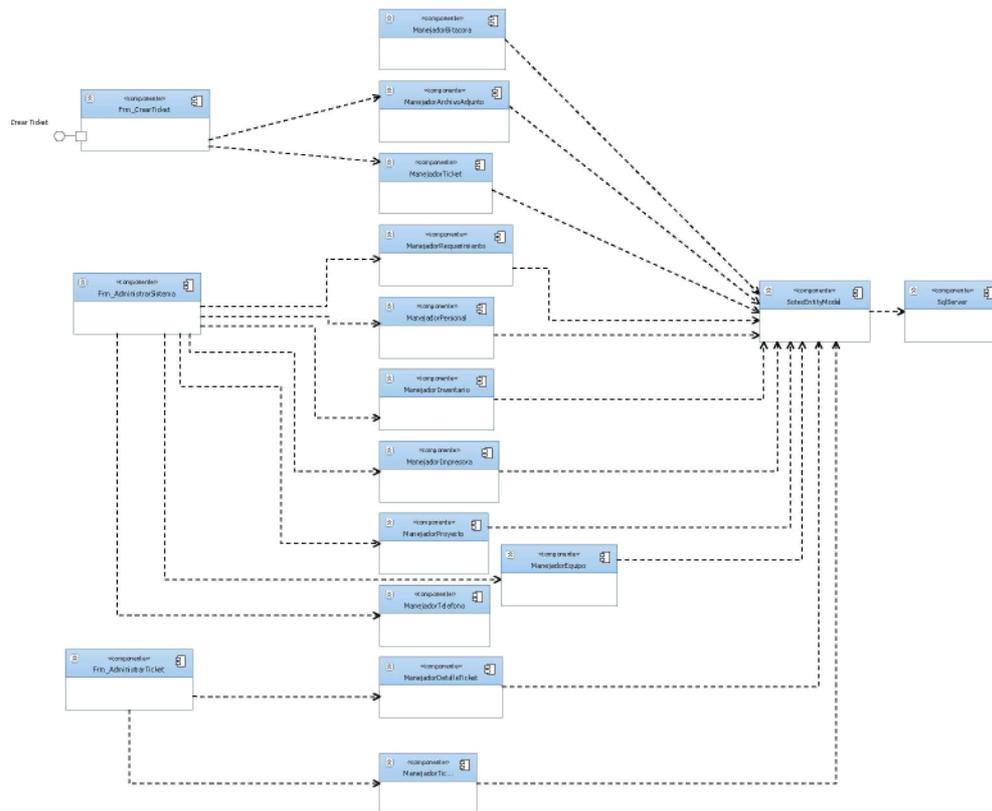


Figura 16. Diagrama de componentes

#### 4.07. Diagrama de estrategias

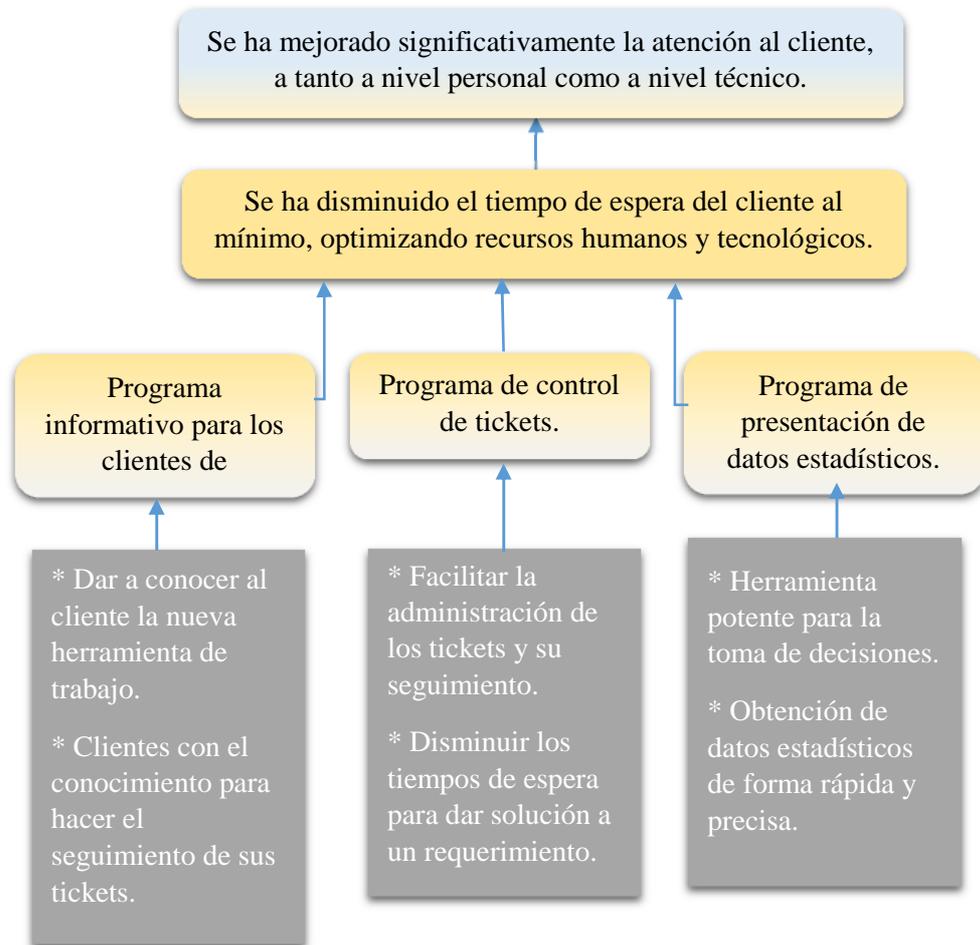


Figura 17. Diagrama de componentes

#### 4.08. Matriz de marco lógico

Véase anexo L.

#### 4.09. Vistas arquitectónicas

##### 4.09.1 Vista Lógica

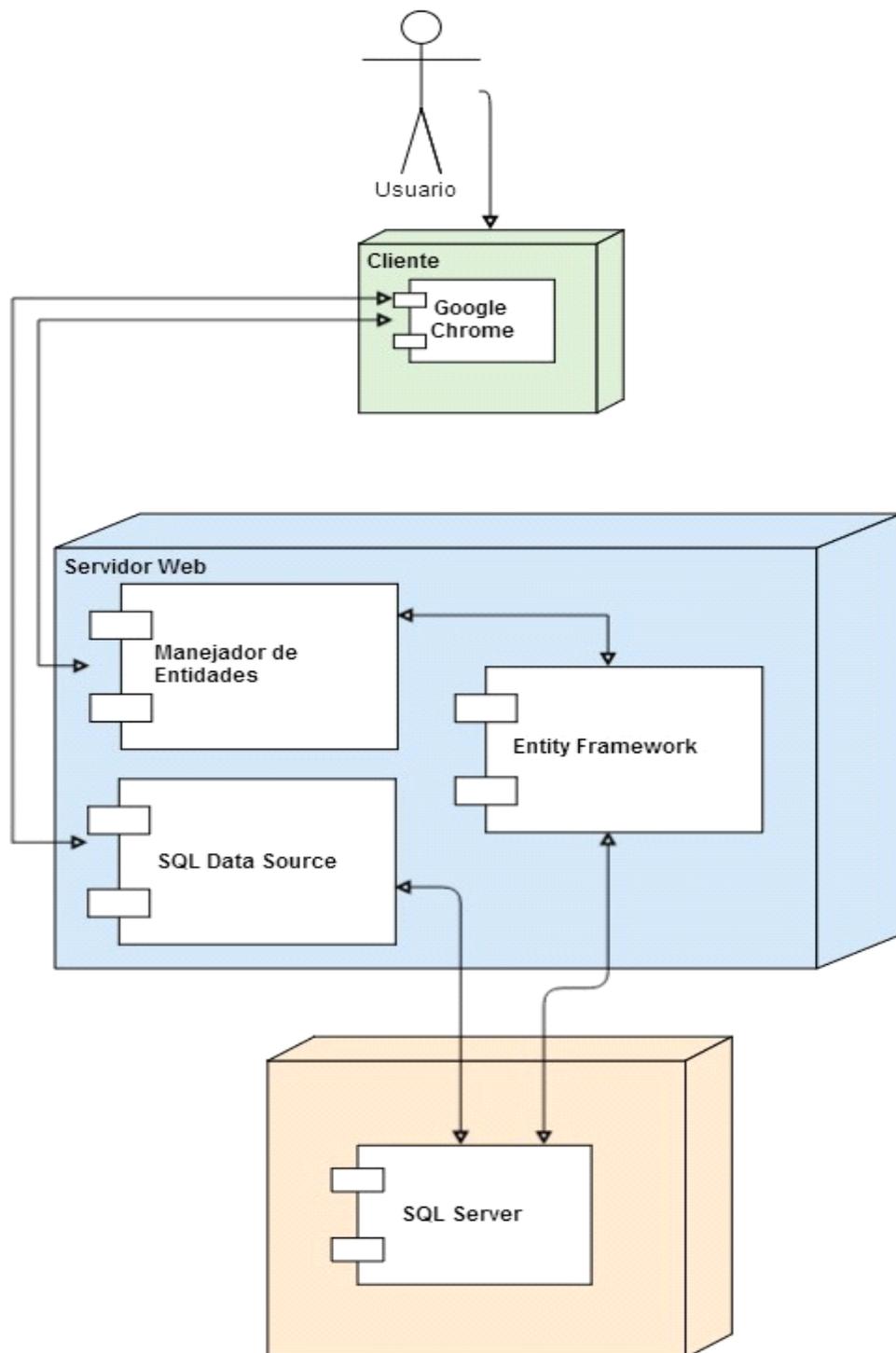


Figura 18. Vista lógica de la solución

### 4.09.2 Vista Física

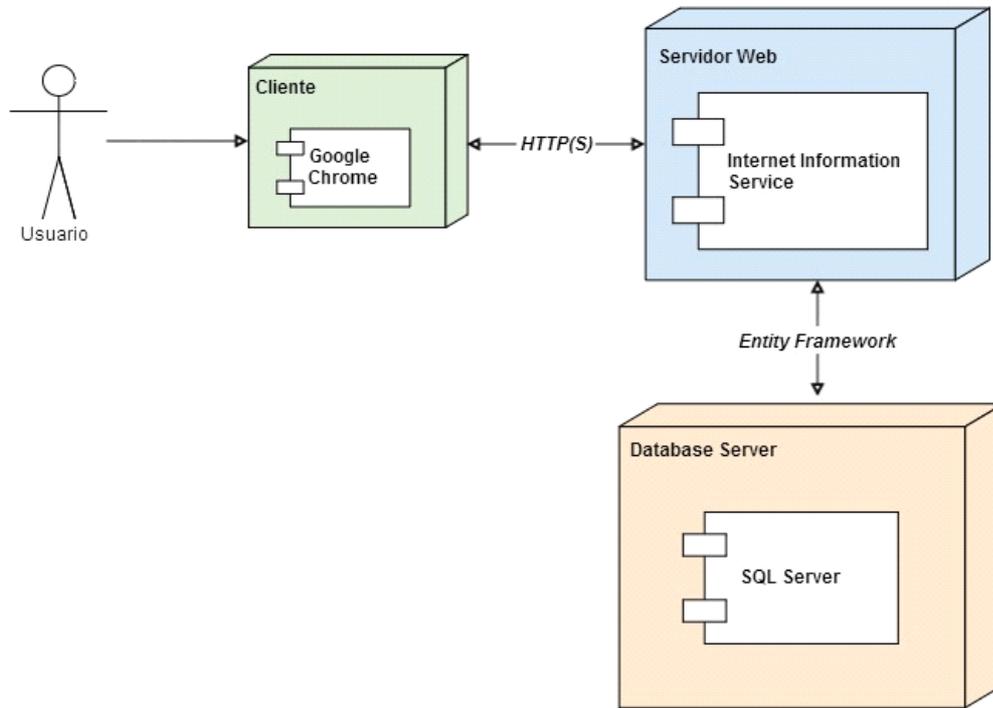


Figura 19. Vista física de la solución

### 4.09.3 Vista de desarrollo

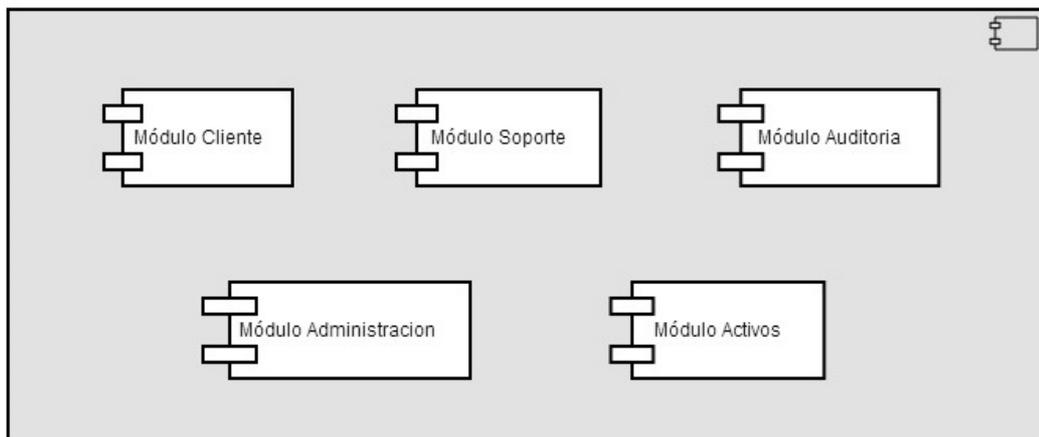


Figura 20. Vista de desarrollo de la solución

#### 4.09.4 Vista de procesos

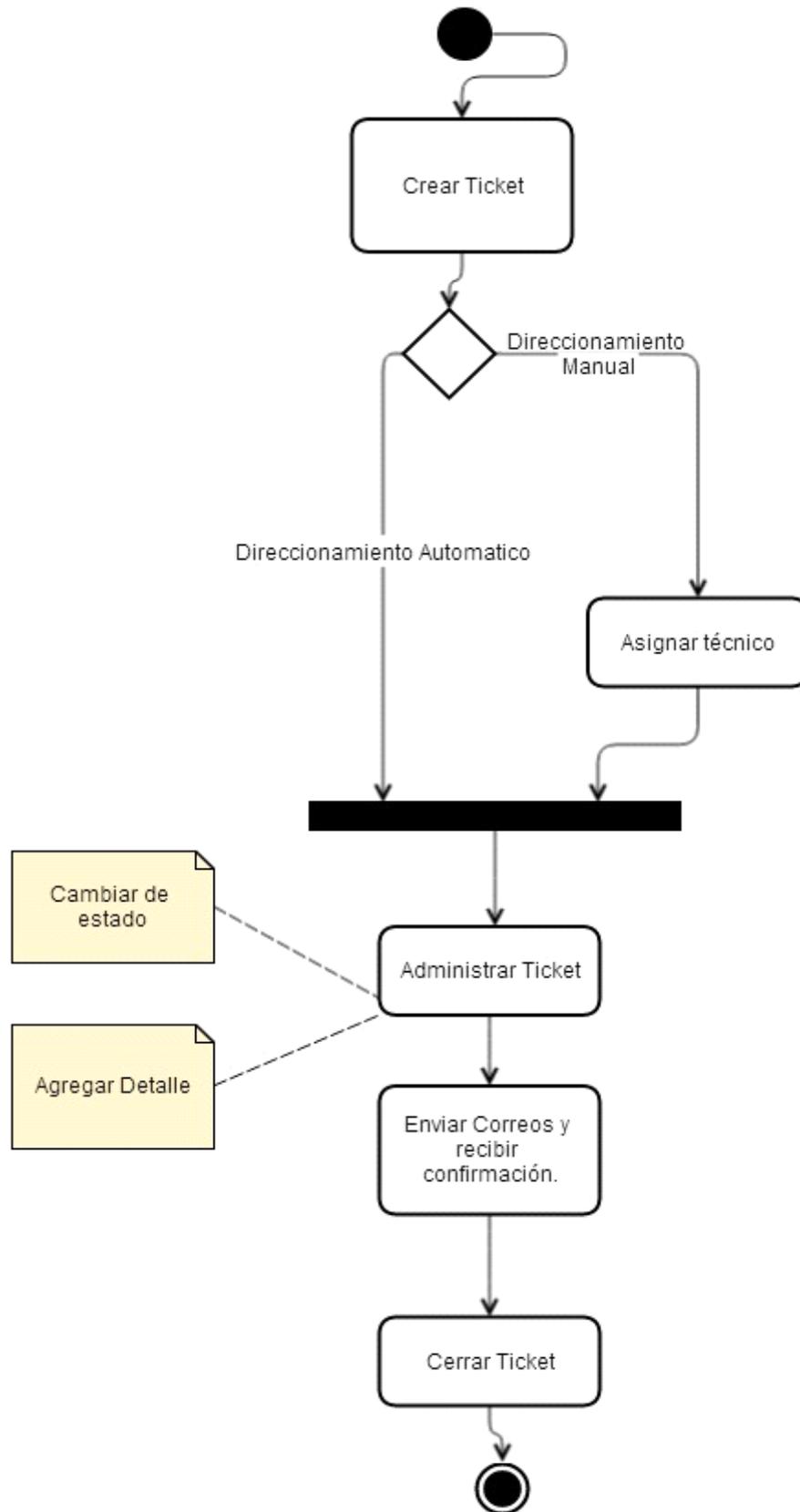


Figura 21. Vista de procesos de la solución

## Capítulo V: Propuesta

### 5.01 Especificación de estándares de programación.

El esquema de nombres es una de las ayudas más importantes para entender el flujo lógico de una aplicación. Un nombre debe más bien expresar el "qué" que el "cómo". Si evita nombres que se refieran a la implementación subyacente (sujeta a cambios), estará conservando un grado de abstracción que lo simplificará todo. Por ejemplo, se usa `cerrarAsignaciones()` en vez de `cerrarElementosdeArreglos()`.

El interés de poner un nombre es que la dificultad para escoger uno adecuado puede indicar que se necesita analizar o definir con mayor precisión el propósito de un elemento. Ponga nombres lo suficientemente largos para que sean elocuentes, pero lo bastante cortos como para que no pequen de palabrería. Desde el punto de vista de la programación, un nombre único sirve solamente para diferenciar un elemento de otro. Los nombres expresivos funcionan como ayuda para el lector, por eso, es lógico dar nombres que sean fáciles de comprender. No obstante, asegúrese de que los nombres escogidos sean compatibles con las reglas de cada lenguaje y con los estándares.

#### Rutinas

Evitamos nombres imprecisos que permitan interpretaciones subjetivas, como por ejemplo `AnalizarEsto()` para una rutina, o bien `xxK8` para una variable. Tales nombres contribuyen más a la ambigüedad que a la abstracción.

En lenguajes orientados a objetos es redundante incluir nombres de clases en el nombre de las propiedades de clases, como por ejemplo `Empresa.EmpresaTitle`. En su lugar, utilice `Empresa.Title`.

Usamos el método verbo-sustantivo para dar nombre a las rutinas que ejecuten alguna operación en un determinado objeto, como por ejemplo `CalculateInvoiceTotal()`.

Todas las sobrecargas deberían llevar a cabo una función similar.

## Variables

Añadimos calificadores de computación (Avg, Sum, Min, Max, Index) después de un nombre de variable donde le resulte apropiado.

Utilizamos pares complementarios en nombres de variables, como min/max, cont, inicio/final, y abierto/cerrado.

Dado que la mayoría de nombres se construyen concatenando varias palabras, emplee una mezcla de mayúsculas y minúsculas para simplificar la lectura. Además, para ayudar a distinguir entre variables y rutinas, utilice el método Pascal de mayúsculas y minúsculas (manejadorPersonalTicket) para los nombres de rutinas, en el que la primera letra de cada palabra está en mayúscula. Para las variables, ponga la primera letra de cada palabra en mayúscula, exceptuando la primera (manejadorPersonalTicket).

Los nombres de variables booleanas deberían establecerse como est. Es decir (inicioEst).

Evitamos usar términos del tipo Flag cuando ponga nombre a variables de estado, que difieren de las variables booleanas en que aquéllas deben tener más de dos valores posibles. En vez de documentFlag, utilice un nombre más descriptivo, del tipo documentFormatType.

Incluso para el caso de una variable de poco uso, que deba aparecer sólo en unas cuantas líneas de código, emplee un nombre descriptivo. Utilice nombres de variables de una sola letra, como i o j sólo para índices cortos.

No utilizamos números o cadenas literales, como por ejemplo `For i = 1 To 7`. En su lugar, emplee constantes con nombre, del tipo `For i = 1 To NUM_DAYS_IN_WEEK` para que resulten fáciles de mantener y comprender.

## Tablas

Cuando ponga nombres a tablas, hágalo en singular. Por ejemplo, use `Employee` en lugar de `Employees`.

Cuando pongamos nombre a las columnas de las tablas, no repita el nombre de la tabla; por ejemplo, evite un campo llamado `EmployeeLastName` de una tabla llamada `Employee`.

No incorpore el tipo de datos en el nombre de una columna. Así reducirá el esfuerzo que podría ser necesario posteriormente para cambiar el tipo de datos.

## Microsoft SQL Server

No ponemos prefijos `sp` a los procedimientos almacenados, ya que se trata de un prefijo reservado para la identificación de procedimientos almacenados de sistemas.

No ponemos prefijos `fn_` a las funciones definidas por el usuario, ya que se trata de un prefijo reservado para funciones integradas.

No ponemos prefijos `xp_` a los procedimientos almacenados extendidos, ya que se trata de un prefijo reservado para la identificación de procedimientos almacenados extendidos.

Minimizamos el uso de abreviaturas; pero si las emplea, use coherentemente las que haya creado. Una abreviatura sólo debe tener un significado y, del mismo modo, a cada palabra abreviada sólo debe corresponder una abreviatura. Por ejemplo, si utiliza

"min." para abreviar "mínimo", hágalo siempre así, y no use también "min." para abreviar "minuto".

Cuando pongamos nombre a las funciones, incluya una descripción del valor que vaya a ser devuelto, como por ejemplo `GetCurrentWindowName()`.

Los archivos y los nombres de carpetas, al igual que los nombres de procedimientos, deben describir claramente su finalidad.

Evitamos reutilizar nombres para elementos diferentes, como por ejemplo una rutina llamada `ProcessSales()` y una variable `iProcessSales`.

Al dar nombre a los elementos, evitamos las palabras mal escritas comúnmente.

## **5.02 Diseño de interfaz de usuario.**

El diseño de la aplicación está basado en tres colores principales, rojo, blanco, negro. Los mismos que son los colores comerciales de la empresa.

Se mantendrá separada la lógica del diseño de las objetos y la lógica de negocio mediante hojas de estilo CCS, para lo cual se incluirá en el proyecto una carpeta con el nombre "style", en la cual guardaremos todas las hojas de estilo utilizadas en el desarrollo de la aplicación.

Para el fondo de la aplicación se usa degradado mediante CSS, con los colores rojo y blanco, poniendo énfasis en el rojo, que es el color más representativo de la empresa, tal como se podrá ver en la siguiente captura de pantalla.

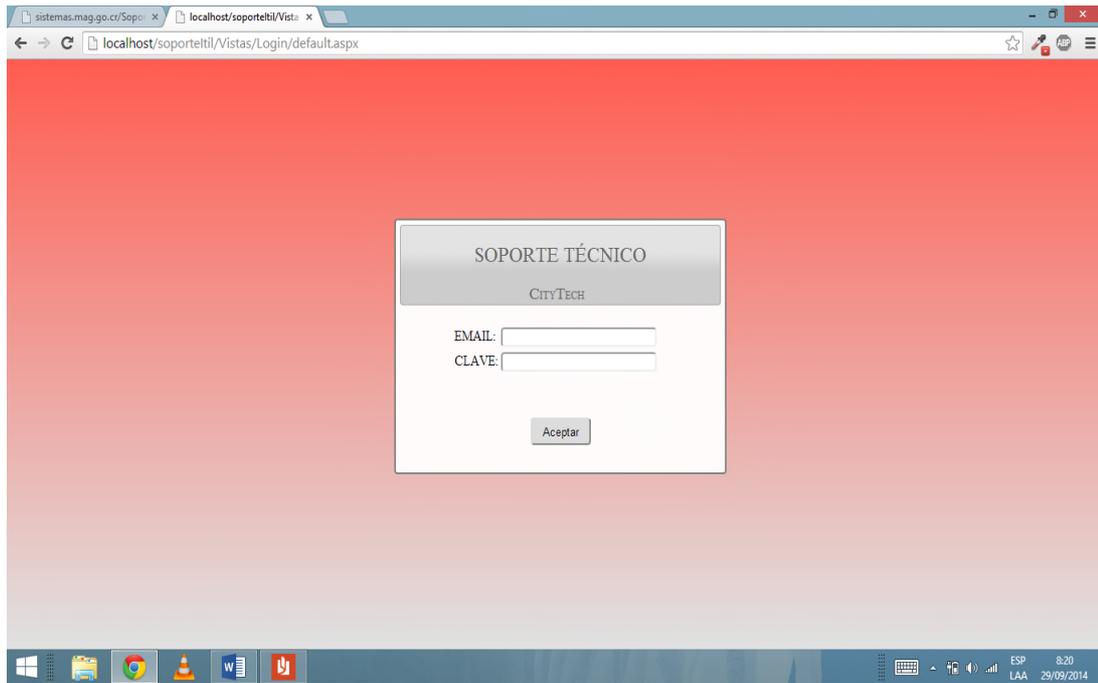


Figura 22. Diseño de registro de usuario.

Todos los formularios excepto el de registro están enlazados a una página de muestra llamada site.Master, la misma que contiene los elementos principales que se mostraran en todos los formularios. Tomar en cuenta que el menú principal de la aplicación se genera dinámicamente, es decir que primero que pone una condición al tipo de usuario que se conecta y dependiendo de su nivel se generara el menú.

El site.Master contiene los siguientes elementos:

- \* 1 Título con etiqueta H1.
- \* 1 Label para mostrar el nombre del usuario conectado.
- \* 1 Menú sin elementos.
- \* 1 Div con estilo de footer o pie de página.

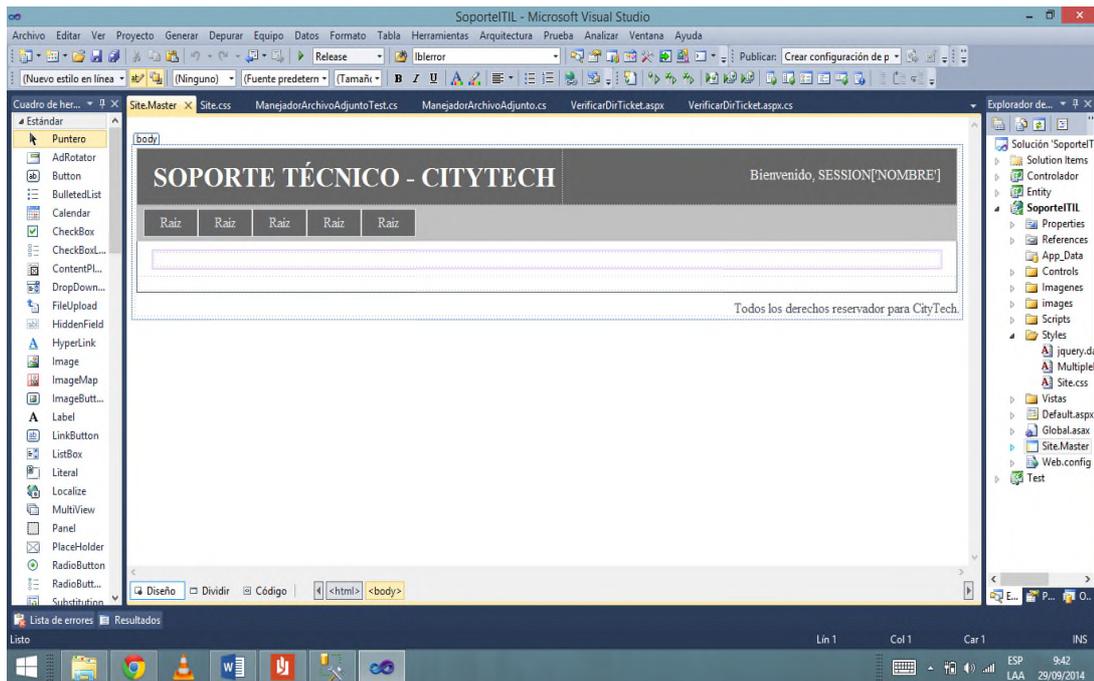


Figura 23. Diseño de la página de muestra

Usaremos un control para la subida de archivos adjuntos al que llamaremos ucMultiFileUpload, el mismo que estará en una carpeta llamada Controls, este control tendrá los siguientes elementos:

- \* 2 Label de información.
- \* 1 File Upload.
- \* 1 HyperLink para aumentar archivos.
- \* 1 Button para controlar el evento de subida.

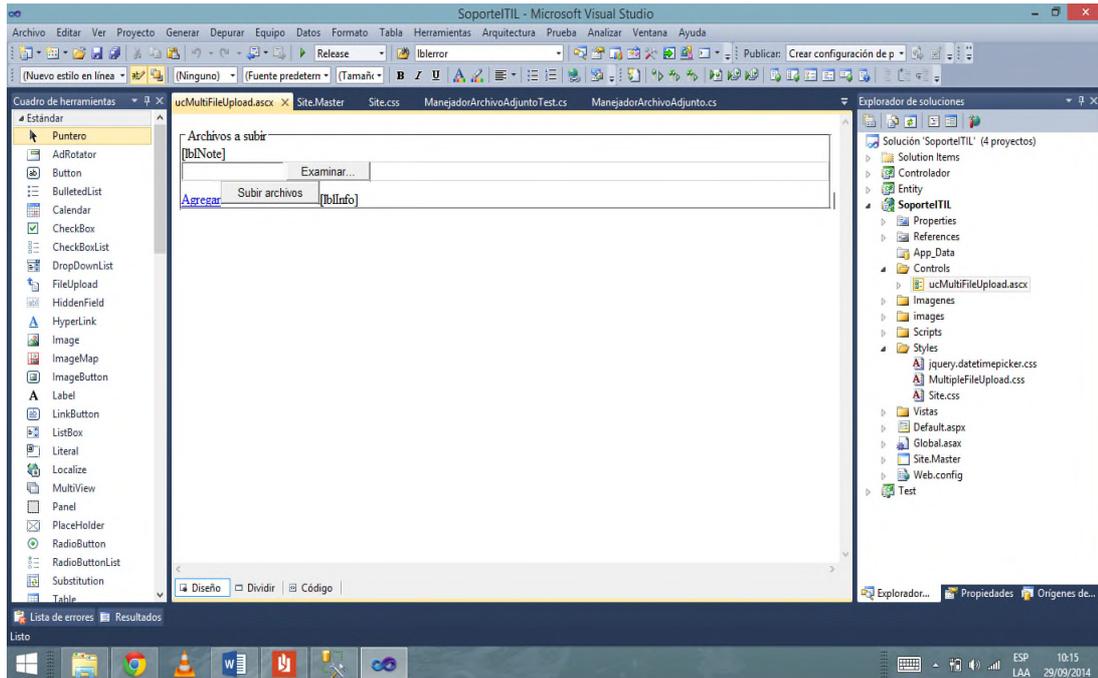


Figura 24. Diseño del control ucMultiFileUpload.

La pantalla de registro contiene los siguientes elementos.

- \* Div Html para dar estilo de POP UP al panel de registro.
- \* Titulo tipo H2.
- \* 3 Labels. 2 para mostrar información y el tercero para mostrar si existe algún problema en el registro.
- \* 2 Text Box para el ingreso del e-mail y la clave.
- \* 2 Elementos de Required Validation para cada text box.
- \* 1 Button para empezar el registro.

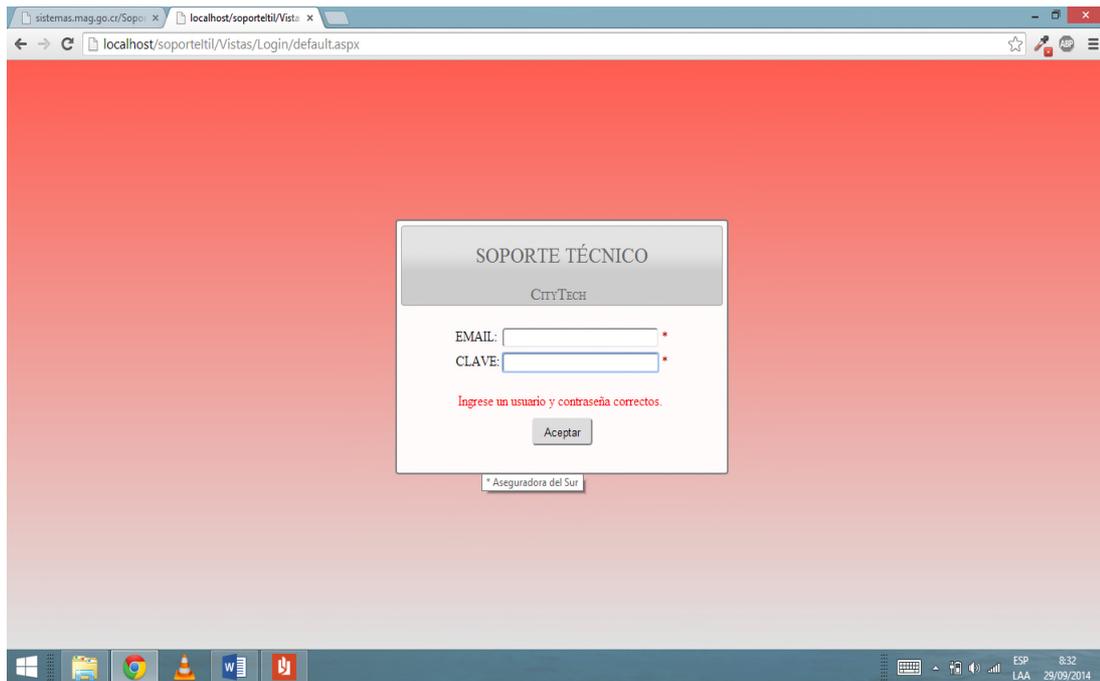


Figura 25. Diseño de registro de usuario con errores.

Al ingresar con perfil de cliente se mostrara la pantalla para la creación de tickets, debemos tener muy claro que de cierta forma el diseño de los objetos de esta pantalla se generan dinámicamente.

Los elementos de la pantalla de creación de ticket son los siguientes.

- \* Un título con etiqueta H2.
- \* 1 Div para contener las opciones principales.
- \* 2 Label de información.
- \* 2 DropDownList, el primero para el tipo de detalle, el segundo para el detalle en si.
- \* 1 Div con datos informativos importantes para el cliente.
- \* 1 Control ucMultiFielUpload.
- \* 1 Text Box con estilo multiLine para la observación del ticket.
- \* 1 Button para aceptar la creación del ticket.

\* 1 Label para mostrar mensajes al usuario.

Existen algunos elementos que se generan solo cuando cumplen ciertas condiciones.

Para ser más específicos, todos los activos que se puedan definir o asignar a un ticket se mostraran dependiendo del detalle de requerimiento que se especifique. En el siguiente caso mostramos como se vería sin aumentar ningún objeto.

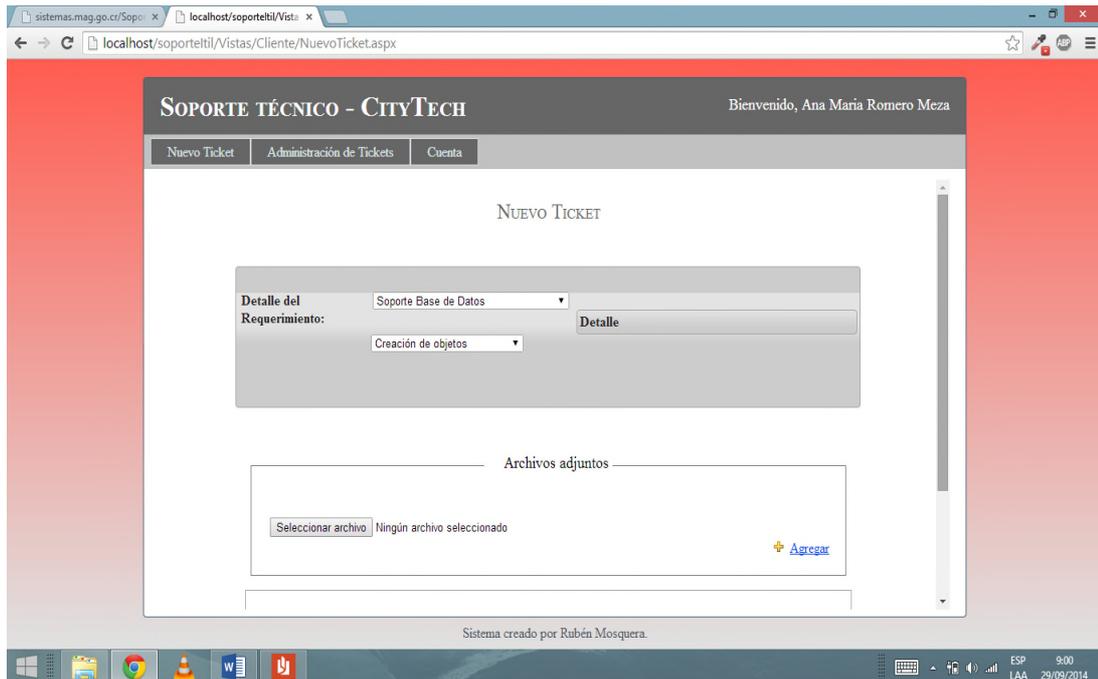


Figura 26. Diseño de creación de Ticket. Nivel de cliente.

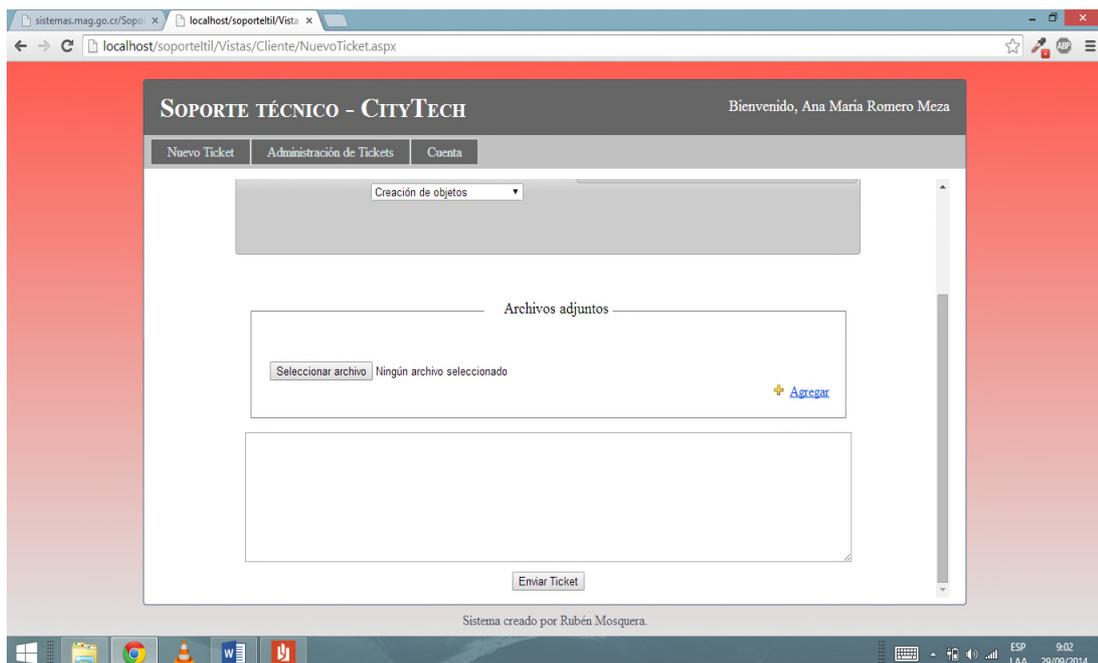


Figura 27. Diseño de creación de Ticket. Nivel de cliente.

Los objetos y sus elementos que se pueden asignar a los requerimientos son los siguientes:

\* Pc o Laptop: contiene 1 label para información, 1 dropDownList si tiene alguna pc asignada y un text box en el caso que no tenga asignada ninguna pc o laptop.

\* Servidor: contiene 1 label para información, 1 dropDownList si existe algún servidor y un text box en el caso que no exista ningún servidor.

\* Impresora: contiene 1 label para información, 1 dropDownList si tiene alguna impresora asignada a su sucursal y un text box en el caso que no tenga asignada ninguna impresora.

\* Teléfono: contiene 1 label para información, 1 dropDownList si tiene algún teléfono asignado y un text box en el caso que no tenga asignada ningún teléfono.

\* Proyecto: contiene 1 label para información, 1 dropDownList si existe algún proyecto y un text box en el caso que no exista.

\* Software Interno: contiene 1 label para información, 1 dropDownList si existe algún software interno y un text box en el caso que no exista.

\* Software externo: contiene 1 label para información, 1 dropDownList si existe algún software externo y un text box en el caso que no exista.

\* Dispositivo: contiene 1 label para información, 1 dropDownList si existe algún dispositivo y un text box en el caso que no exista.

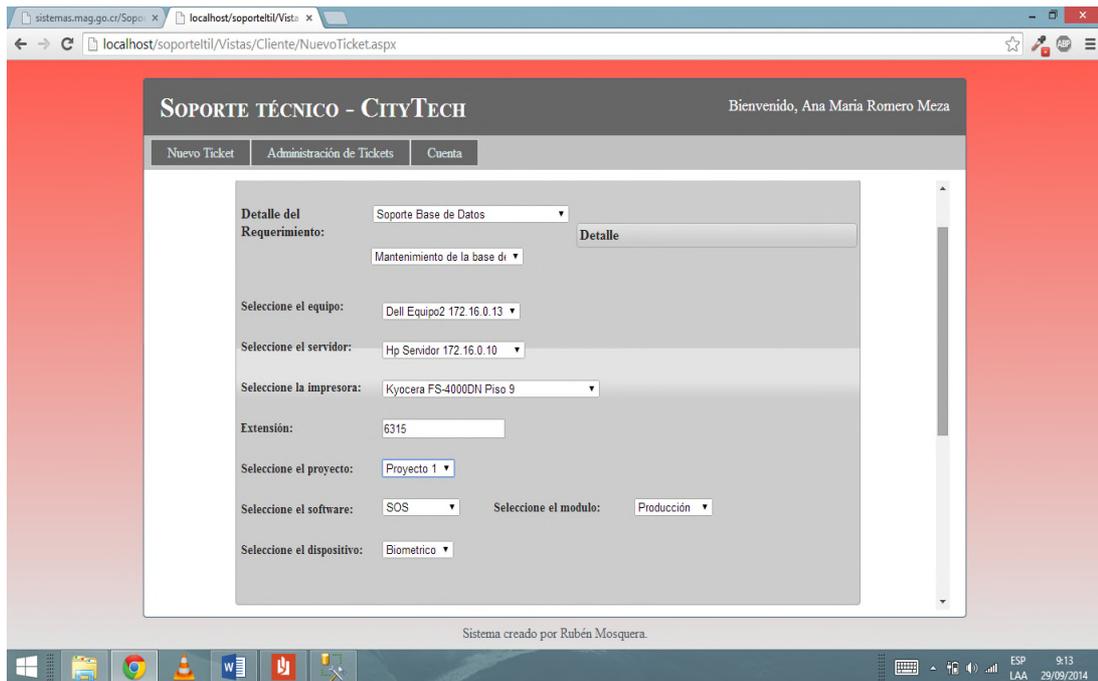


Figura 28. Diseño de objetos dinámicos en creación de Ticket. Nivel de cliente.

Para que el cliente pueda administrar verificar sus tickets se creó el formulario de verificación de tickets. El mismo será responsivo, y la forma que tendrá será diferente dependiendo de si se está verificando todos los tickets o uno en específico.

Cuando se está administrando todos los tickets se tendrá los siguientes elementos:

- \* Título con etiqueta H2
- \* 1 DropDownList para los posibles estados de tickets.
- \* 1 ImageButton para desplegar las opciones de filtrado.
- \* 4 Labels informativos.
- \* 4 TextBox para filtrar datos.
- \* 1 Button para proceder a filtrar.
- \* 1 DataGridView con todos los tickets asignados

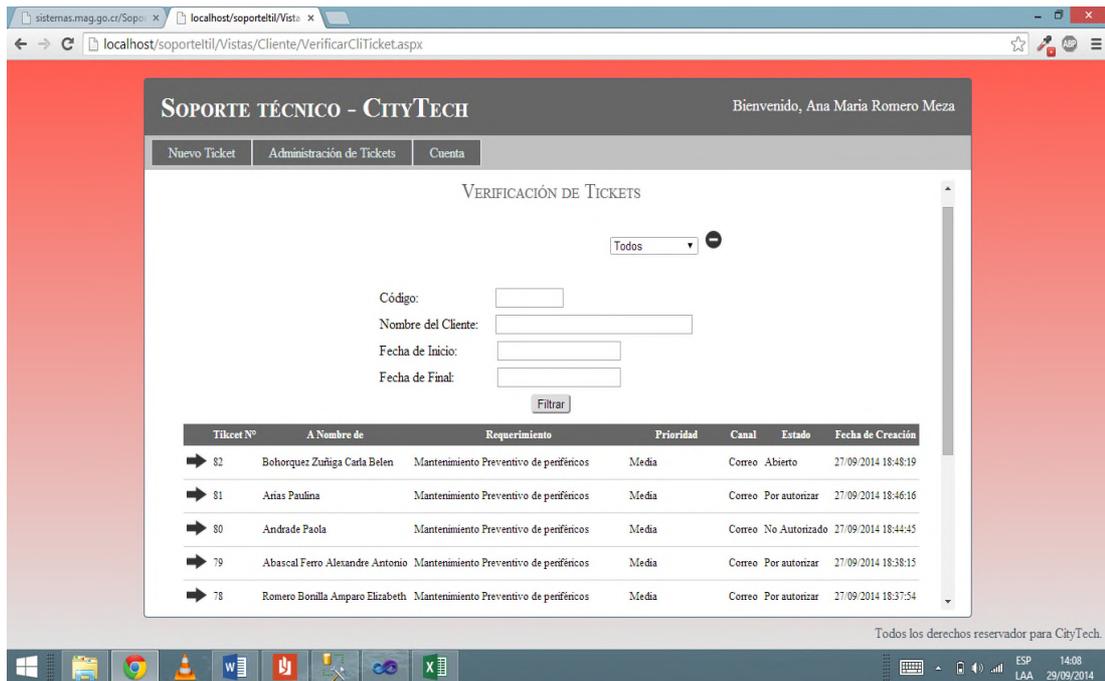


Figura 29. Diseño de verificación de tickets en general.

Cuando se administra un ticket específico se tendrá los siguientes elementos:

- \* 1 Título con etiqueta H2.
- \* 1 Contenedor de Tabs, el mismo que contendrá 3 Tabs.
  - Datos Generales.
  - Detalle.
  - Administración.
- \* Datos Generales, contendrá los siguientes elementos.
  - 29 Labels informativos con los datos del cliente y del ticket.
  - 3 Títulos con etiqueta H4.

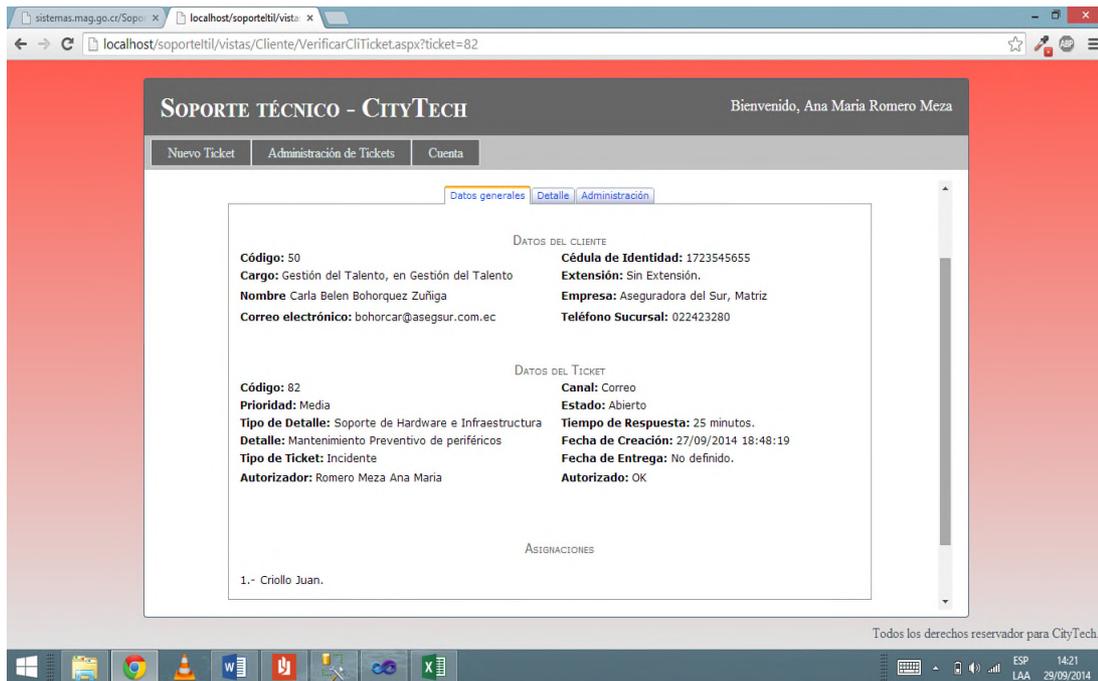


Figura 30. Diseño de verificación de datos específicos de un Ticket. Tab Datos Generales.

\* Detalles. Hay que tomar en cuenta que en este Tab se generaran los detalles necesarios, es decir que puede contener una cantidad indefinida de detalles, los mismos que los generan los propios usuarios, cada detalle contendrá los siguientes elementos:

- 1 Título con etiqueta H5.
- 6 Labels informativos. El ultimo se usara para mostrar la observación.
- Hyperlink, para detallar los archivos adjuntos. Esto se genera dinámicamente, es decir que no sabremos la cantidad exacta.

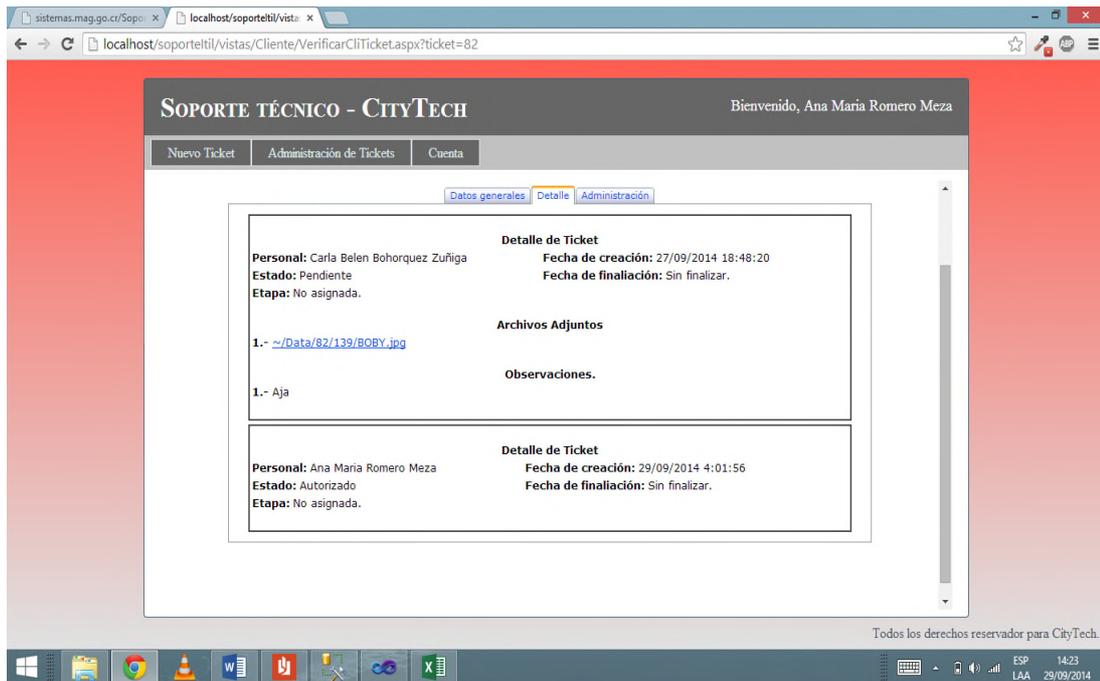


Figura 31. Diseño de verificación de datos específicos de un ticket. Tab Detalles.

\* Administración. En este Tab vamos a poder cerrar o autorizar el ticket según sea el caso.

- 1 Título con etiqueta H2.

- 1 Button para cumplir con la función requerida si es aceptada.

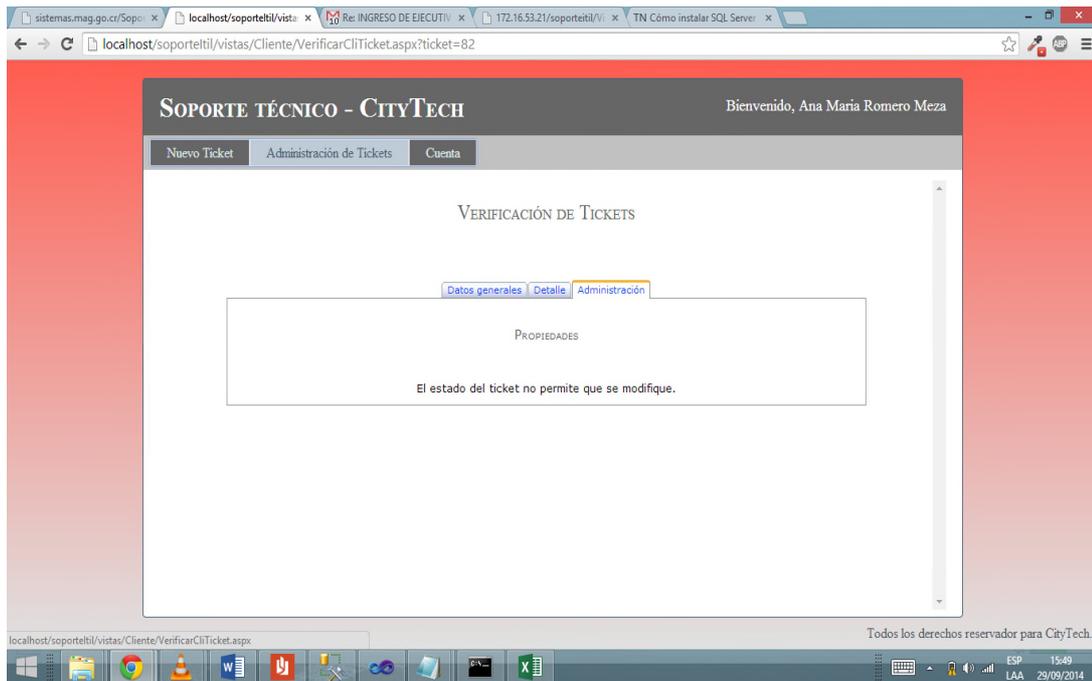


Figura 32. Diseño de verificación de datos específicos de un ticket. Tab Detalles.

Cuando ingresemos al sistema como direccionador la primera pantalla que se nos mostrara será la de creación de ticket, la misma que contara con un diseño parecido al de creación de ticket por parte del cliente, pero con más opciones para un correcto direccionamiento.

Los elementos con los que contara este formulario son los siguientes:

- \* Un título con etiqueta H2.
- \* 1 Div para contener las opciones principales.
- \* 2 Label de información.
- \* 2 CheckBox para direccionamiento y para autorización.
- \* 2 DropDownList para direccionamiento y autorización.
- \* 1 Label y DropDownList para cliente.
- \* 2 TextBox que usaremos para filtrar los clientes y autorizadores.

- \* 2 DropDownList, el primero para el tipo de detalle, el segundo para el detalle en si.
- \* 1 Div con datos informativos importantes para el cliente.
- \* 1 Control ucMultiFielUpload.
- \* 1 Text Box con estilo multiLine para la observación del ticket.
- \* 1 Button para aceptar la creación del ticket.
- \* 1 Label para mostrar mensajes al usuario.

Existen algunos elementos que se generan solo cuando cumplen ciertas condiciones. Para ser más específicos, todos los activos que se puedan definir o asignar a un ticket se mostraran dependiendo del detalle de requerimiento que se especifique. En el siguiente caso mostramos como se vería sin aumentar ningún objeto.

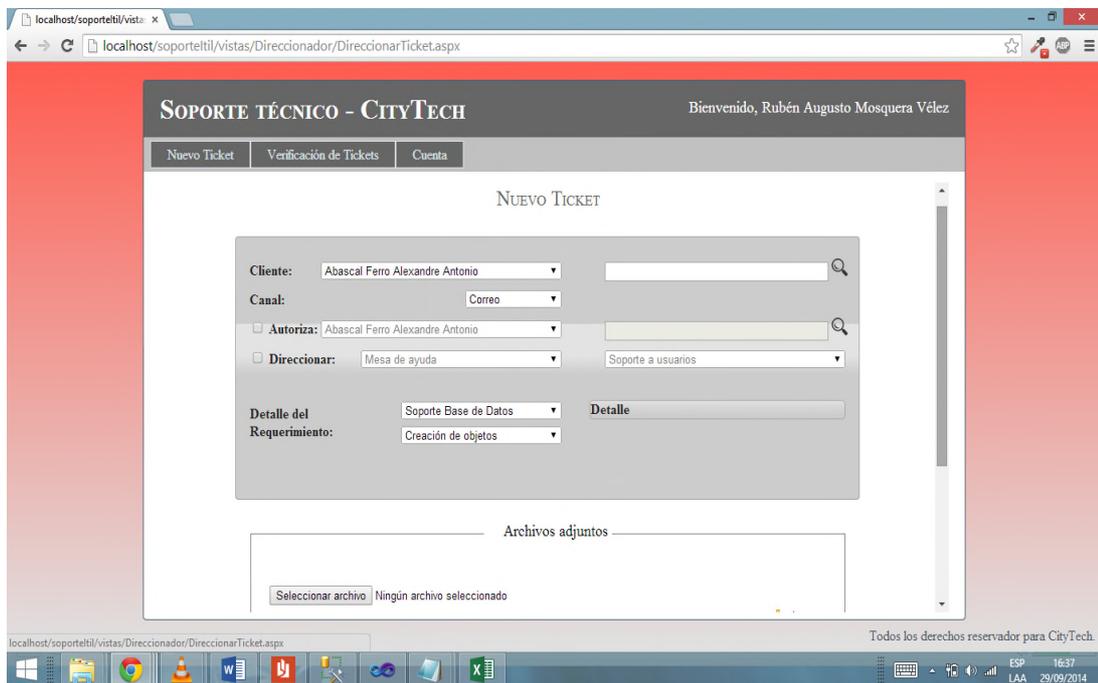


Figura 33. Diseño de creación de Ticket. Nivel de direccionador.

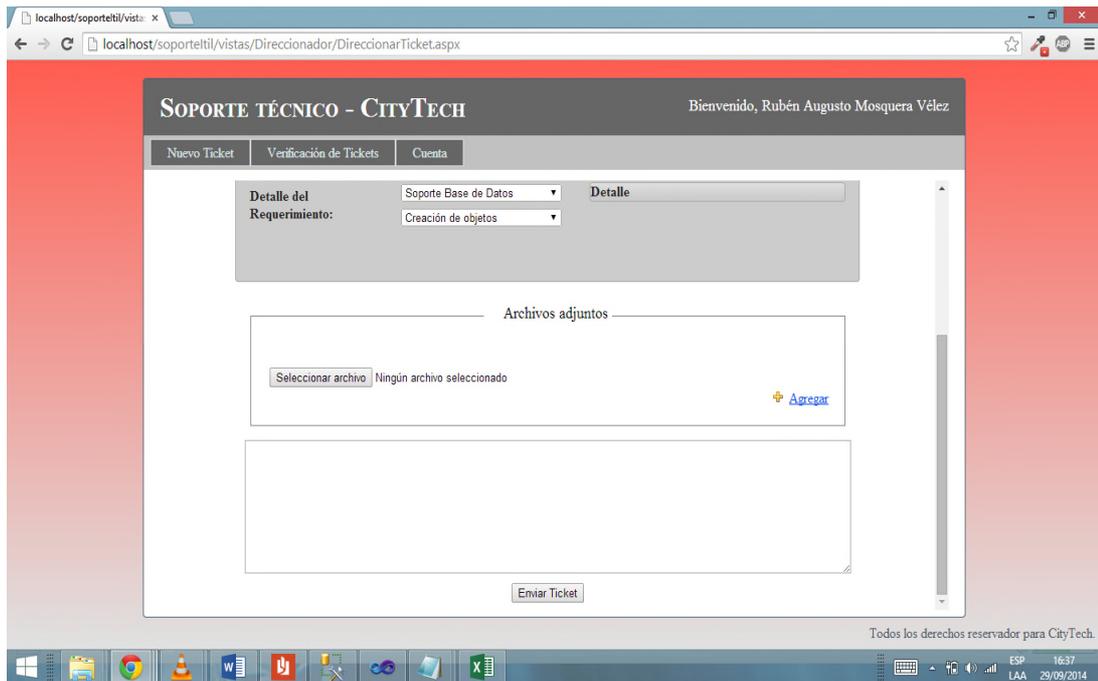


Figura 34. Diseño de creación de Ticket. Nivel de direccionador.

Los objetos y sus elementos que se pueden asignar a los requerimientos son los siguientes:

- \* Pc o Laptop: contiene 1 label para información, 1 dropDownList si tiene alguna pc asignada y un text box en el caso que no tenga asignada ninguna pc o laptop.
- \* Servidor: contiene 1 label para información, 1 dropDownList si existe algún servidor y un text box en el caso que no exista ningún servidor.
- \* Impresora: contiene 1 label para información, 1 dropDownList si tiene alguna impresora asignada a su sucursal y un text box en el caso que no tenga asignada ninguna impresora.
- \* Teléfono: contiene 1 label para información, 1 dropDownList si tiene algún teléfono asignado y un text box en el caso que no tenga asignada ningún teléfono.
- \* Proyecto: contiene 1 label para información, 1 dropDownList si existe algún proyecto y un text box en el caso que no exista.

- \* Software Interno: contiene 1 label para información, 1 dropDownList si existe algún software interno y un text box en el caso que no exista.
- \* Software externo: contiene 1 label para información, 1 dropDownList si existe algún software externo y un text box en el caso que no exista.
- \* Dispositivo: contiene 1 label para información, 1 dropDownList si existe algún dispositivo y un text box en el caso que no exista.

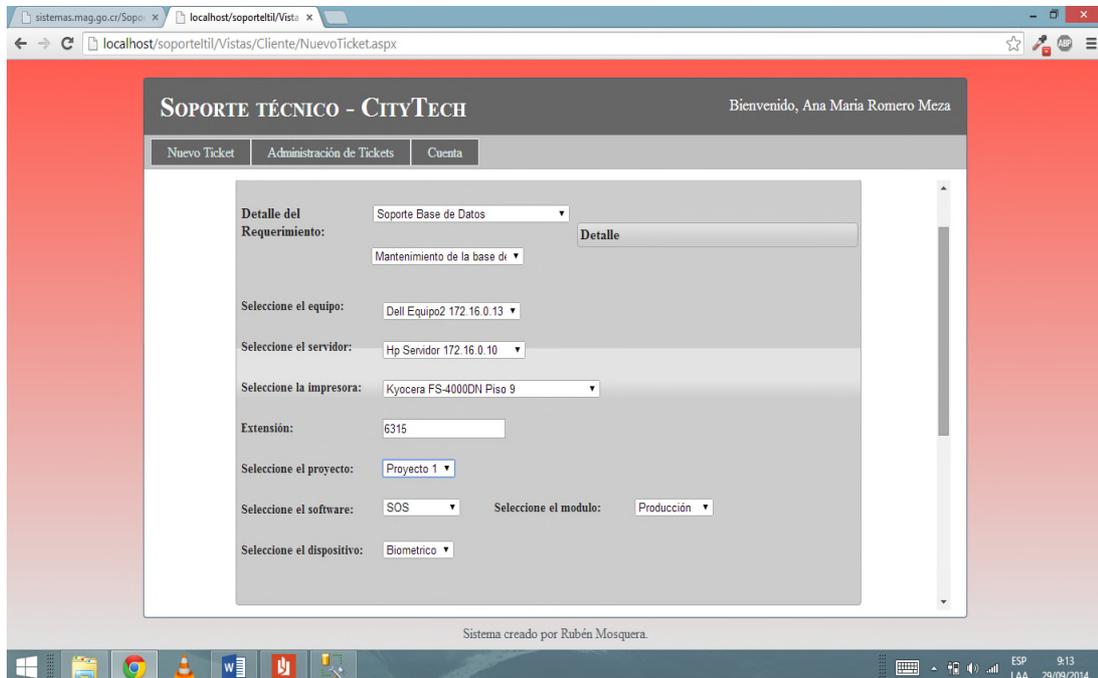


Figura 35. Diseño de objetos dinámicos en creación de Ticket. Nivel de direccionador.

Para que el direccionador puede administrar verificar todos los tickets se creó el formulario de verificación de tickets. El mismo será responsivo, y la forma que tendrá será diferente dependiendo de si se está verificando todos los tickets o uno en específico.

Cuando se está administrando todos los tickets se tendrá los siguientes elementos:

- \* Título con etiqueta H2

- \* 1 DropDownList para los posibles estados de tickets.
- \* 1 ImageButton para desplegar las opciones de filtrado.
- \* 4 Labels informativos.
- \* 4 TextBox para filtrar datos.
- \* 1 Button para proceder a filtrar.
- \* 1 DataGrid con todos los tickets asignados

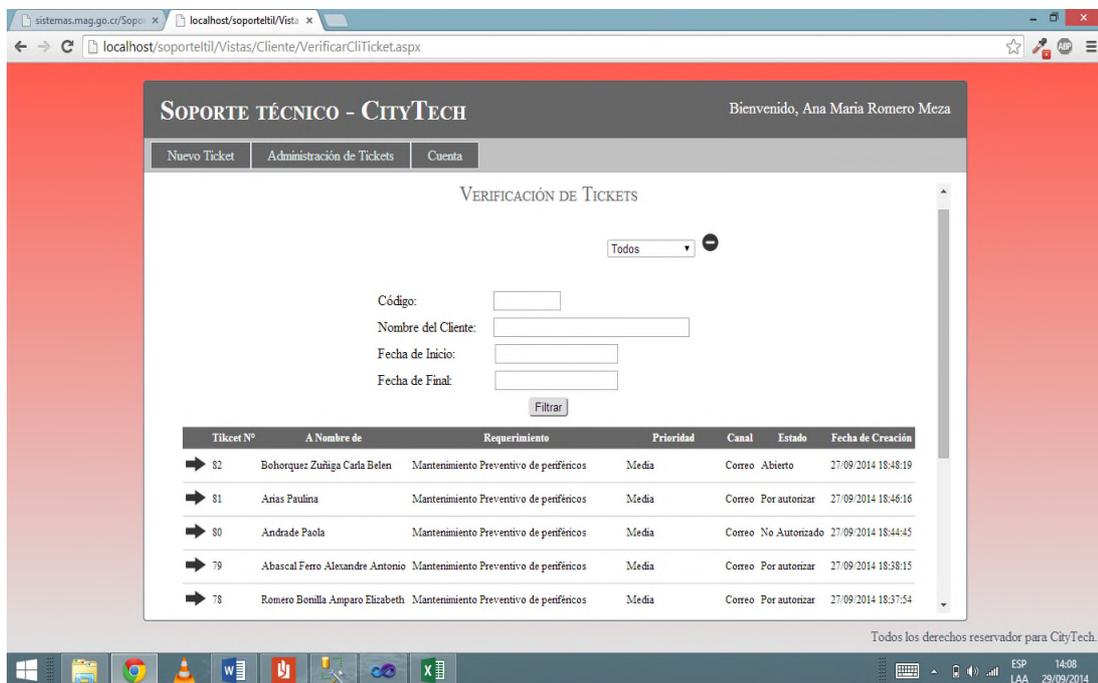


Figura 36. Diseño de verificación de tickets en general.

Cuando se administra un ticket específico se tendrá los siguientes elementos:

- \* 1 Titulo con etiqueta H2.
- \* 1 Contenedor de Tabs, el mismo que contendrá 3 Tabs.
  - Datos Generales.
  - Detalle.

- Administración.

\* Datos Generales, contendrá los siguientes elementos.

- 29 Labels informativos con los datos del cliente y del ticket.

- 3 Titulos con etiqueta H4.

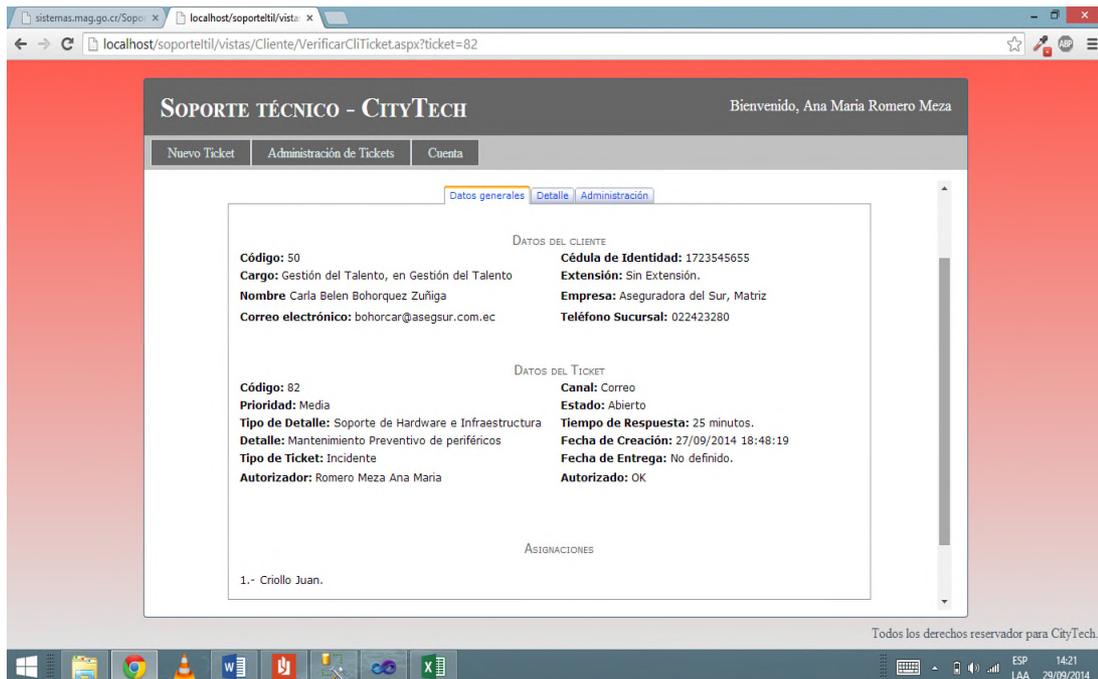


Figura 37. Diseño de verificación de datos específicos de un Ticket. Tab Datos Generales.

\* Detalles. Hay que tomar en cuenta que en este Tab se generaran los detalles necesarios, es decir que puede contener una cantidad indefinida de detalles, los mismos que los generan los propios usuarios, cada detalle contendrá los siguientes elementos:

- 1 Titulo con etiqueta H5.

- 6 Labels informativos. El ultimo se usara para mostrar la observación.

- Hyperlink, para detallar los archivos adjuntos. Esto se genera dinámicamente, es decir que no sabremos la cantidad exacta.

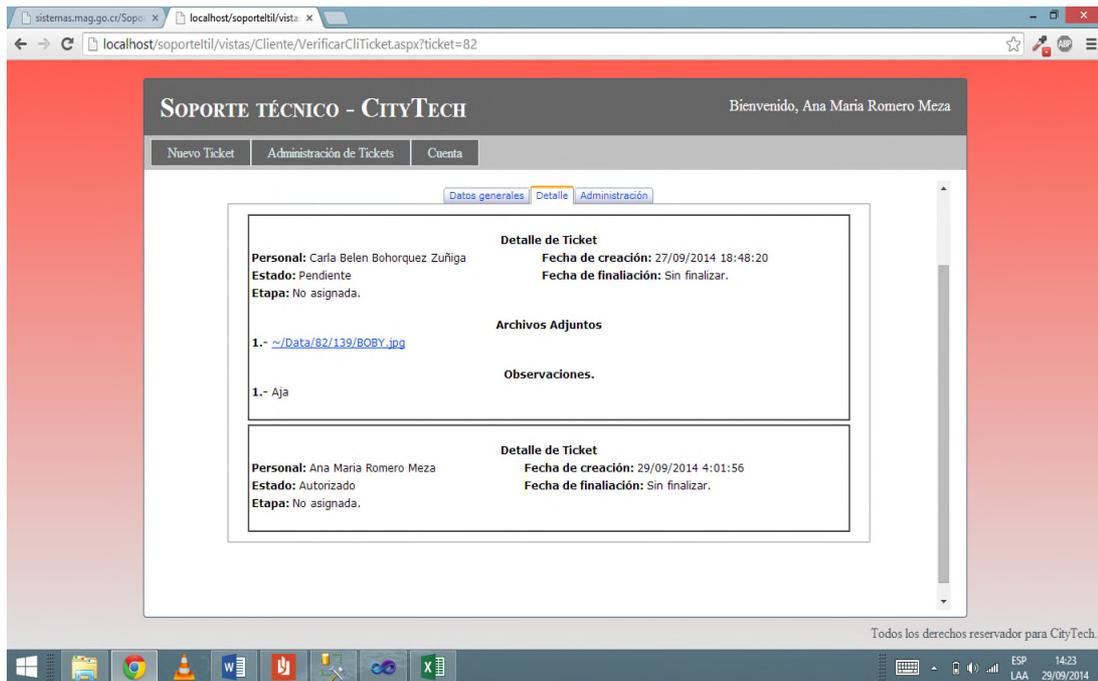


Figura 38. Diseño de verificación de datos específicos de un ticket. Tab Detalles.

\* **Administración.** En este Tab vamos a poder autorizar, pedir autorización, redireccionar o no autorizar el ticket según sea el caso.

- 1 Título con etiqueta H2.
- 19 CheckBox para activar las diferentes opciones.
- 23 DropDownList para diferentes opciones.
- 1 Button para cumplir con la función requerida si es aceptada.
- 8 TextBox para filtrar algunas opciones y uno para escoger fecha.
- 1 TextBox MultiLine para la observación.
- 1Control ucMultiFileUpload.
- 1 Button guardar ticket.

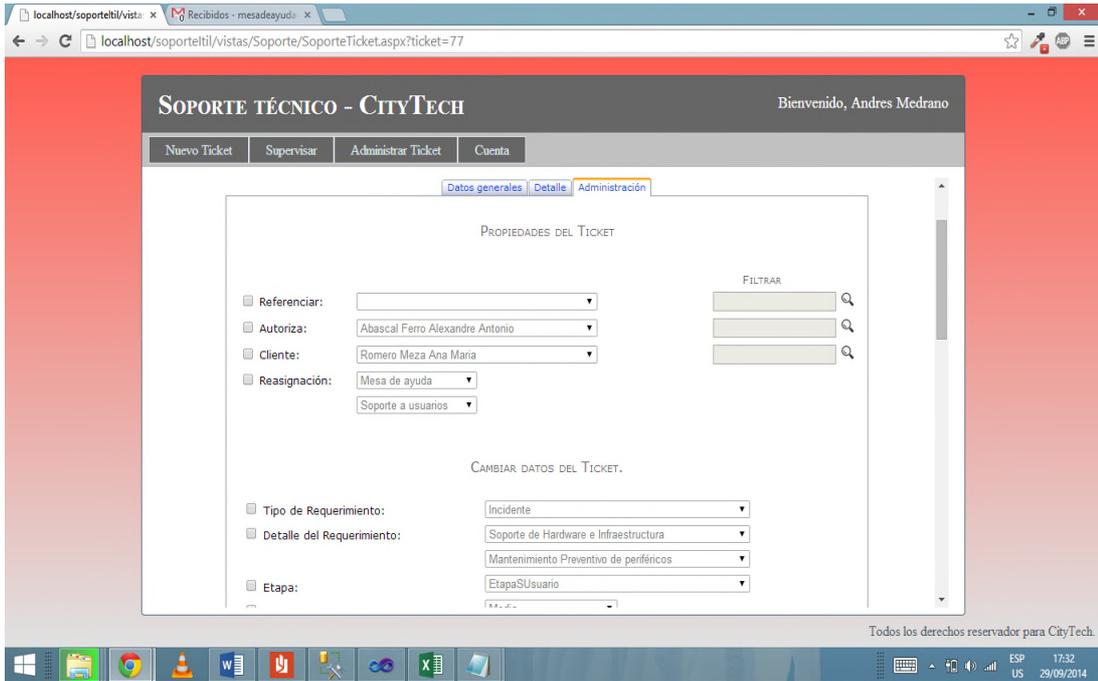


Figura 39. Diseño de administración de ticket.

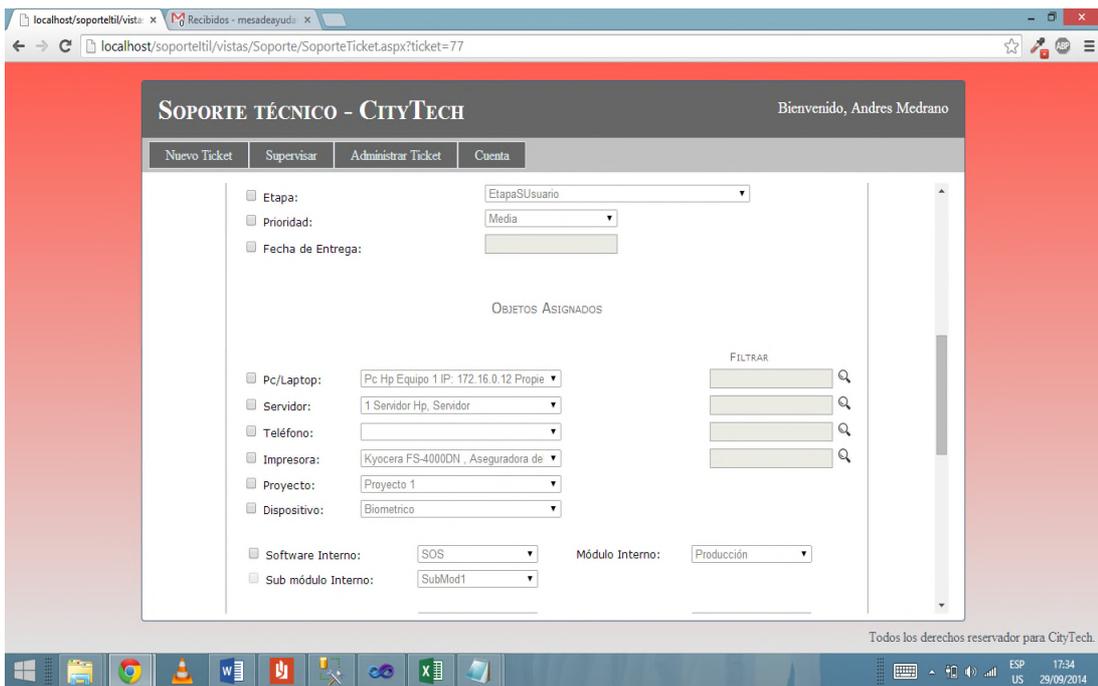


Figura 40. Diseño de administración de ticket.

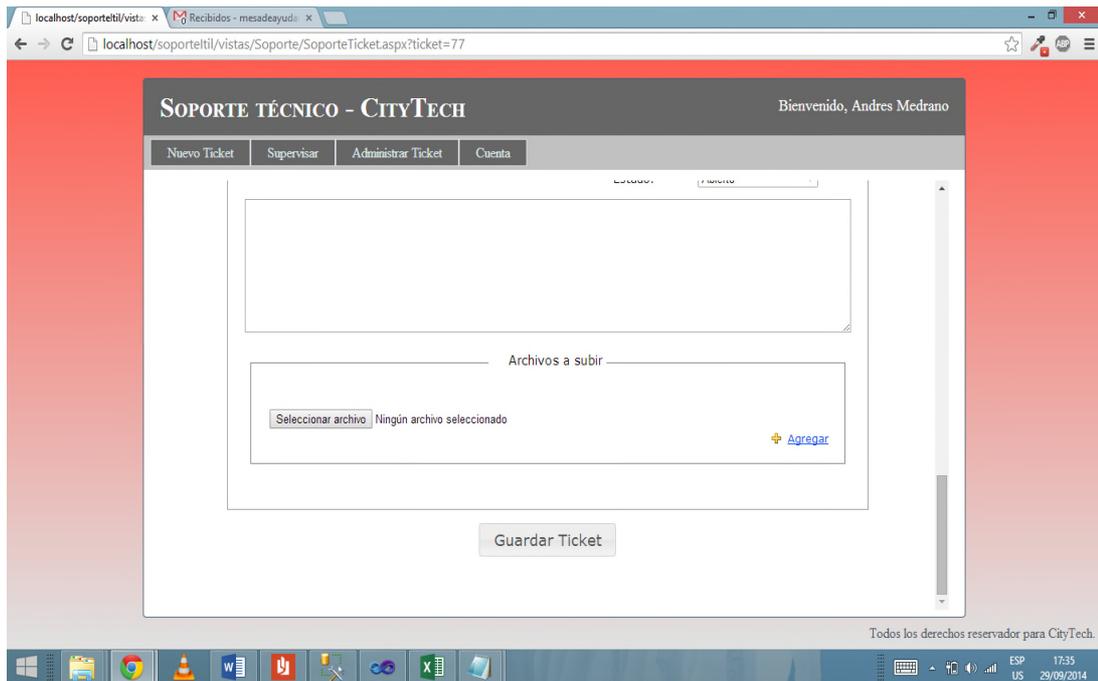


Figura 41. Diseño de administración de ticket.

Para el mantenimiento de las tablas usaremos un diseño tipo árbol. Se usara un div para contener todos los elementos del mantenimiento. La base de todos los mantenimientos es:

- \* 1 GridView.
- \* 2 ImageButton.
- \* 3 Label informativos.
- \* 2 TextBox, el primero para mostrar el código, el segundo la descripción.
- \* 1 DropDownList que muestra el estado del registro.

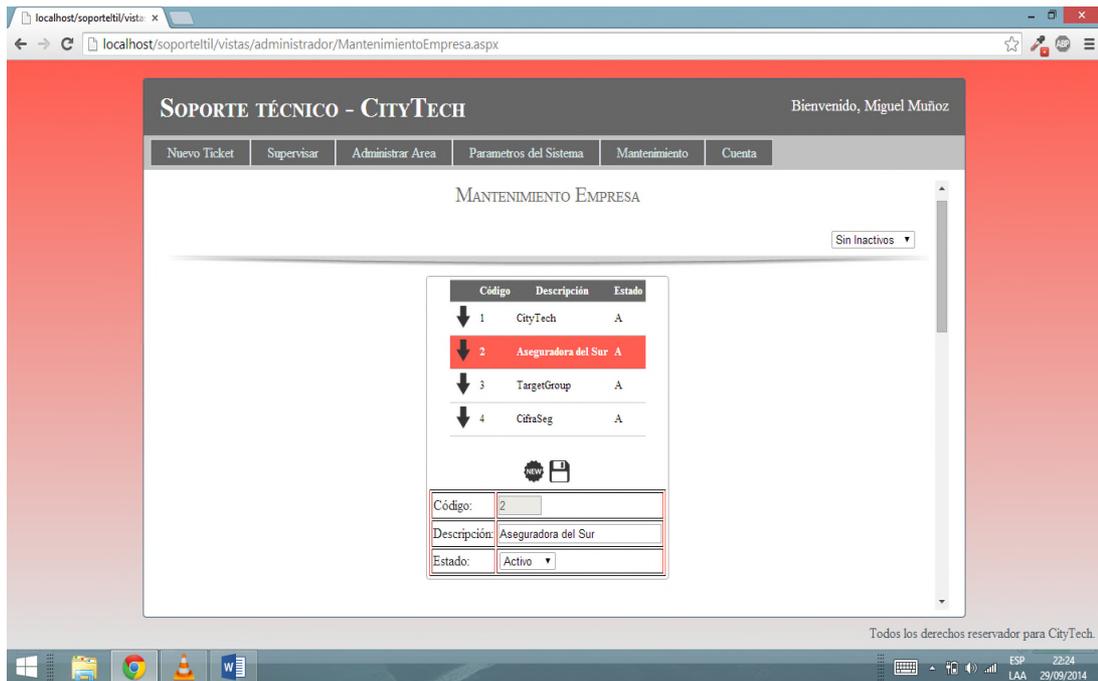


Figura 42. Diseño mantenimiento de empresa.

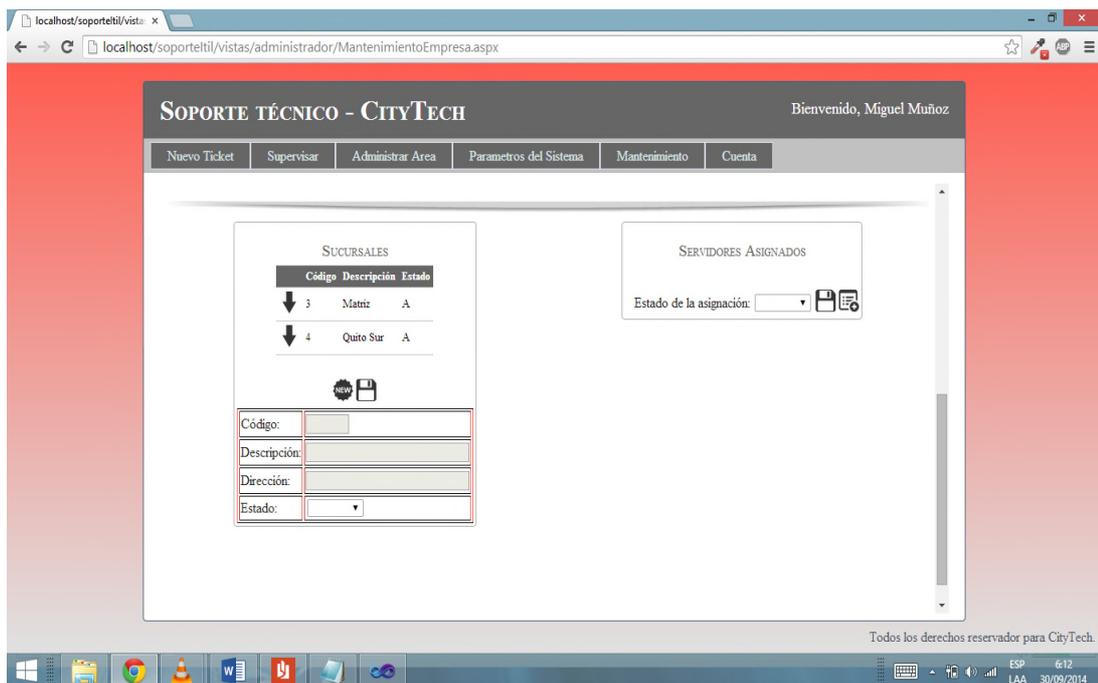


Figura 43. Diseño mantenimiento de Sucursales.

### 5.03 Especificación de pruebas de unidad.

Tabla 25.

*Prueba de unidad ManejadorArchivoAdjuntoTest*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorArchivoAdjuntoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_ARCHIVOADJUNTO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 26.

*Prueba de unidad ManejadorArchivoAdjuntoTest 2.*

2Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorArchivoAdjuntoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_ARCHIVOADJUNTO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 27.

*Prueba de unidad ManejadorBitacoraTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorBitacoraTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_BITACORA.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor y la creación correcta del registro.
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 28.

*Prueba de unidad ManejadorBitacoraTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorBitacoraTest
<b>Método a probar:</b>	ObtenerIP()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método, además de verificar la veracidad de la IP obtenida,
<b>Datos de Entrada:</b>	Ninguno
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 29.

*Prueba de unidad ManejadorBitacoraTest 3.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorBitacoraTest
<b>Método a probar:</b>	ObtenerMAC ()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método, además de verificar la veracidad de la MAC obtenida,
<b>Datos de Entrada:</b>	Ninguno
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 30.

*Prueba de unidad ManejadorClaveTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorClaveTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_CLAVE.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor y la creación correcta del registro.
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 31.

*Prueba de unidad ManejadorDepartamentoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDepartamentoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DEPARTAMENTO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 32.

*Prueba de unidad ManejadorDepartamentoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadoDepartamentoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DEPARTAMENTO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra poner a prueba el método sin ninguna novedad.

Tabla 33.

*Prueba de unidad ManejadorDetalleTicketTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDetalleTicketTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al generar el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DETALLETICKET.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra guardar el registro y desactivar el anterior al mismo sin problemas.

Tabla 34.

*Prueba de unidad ManejadorDetalleTicketTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDetalleTicketTest
<b>Método a probar:</b>	desactivarDetalle()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al modificar el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DETALLETICKET.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor
<b>Comentarios:</b>	Se logra desactivar el anterior al mismo sin problemas.

Tabla 35.

*Prueba de unidad ManejadorDetalleTicketTest 3.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDetalleTicketTest
<b>Método a probar:</b>	cerrarAsignaciones()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al generar el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DETALLETICKET.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor para identificar las asignaciones y cerrarlas.
<b>Comentarios:</b>	Se logra cerrar todas las asignaciones sin problemas.

Tabla 36.

*Prueba de unidad ManejadorDetalleTicketTest 4.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDetalleTicketTest
<b>Método a probar:</b>	obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al generar el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DETALLETICKET.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor.
<b>Comentarios:</b>	Se logra el registro de los detalles sin problemas.

Tabla 37.

*Prueba de unidad ManejadorDispositivoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorDispositivoTest
<b>Método a probar:</b>	obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al generar el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_DISPOSITIVO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor.
<b>Comentarios:</b>	Se logra cerrar todas las asignaciones sin problemas.

Tabla 38.

*Prueba de unidad ManejadorDispositivoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEmpresaTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EMPRESA. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 39.

*Prueba de unidad ManejadorEmpresaTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEmpresaTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EMPRESA.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 40.

*Prueba de unidad ManejadorEmpresaServidorTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEmpresaServidorTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EMPRESASERVIDOR. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 41.

*Prueba de unidad ManejadorEmpresaServidorTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEmpresaServidorTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EMPRESASERVIDOR.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 42.

*Prueba de unidad ManejadorEquipoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEquipoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EQUIPO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 43.

*Prueba de unidad ManejadorEquipoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEquipoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_EQUIPO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 44.

*Prueba de unidad ManejadorEstadoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEstadoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_ESTADO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 45.

*Prueba de unidad ManejadorEstadoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorEstadoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_ESTADO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 46.

*Prueba de unidad ManejadorImpresoraTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorImpresoraTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_IMPRESORA. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 47.

*Prueba de unidad ManejadorImpresoraTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorImpresoraTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_IMPRESORA.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 48.

*Prueba de unidad ManejadorIpTest 3.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorIpTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_IMPRESORA.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 49.

*Prueba de unidad ManejadorIpTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorIpTest
<b>Método a probar:</b>	cambiarIp()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_IP.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al cambiar el registro correctamente.
<b>Comentarios:</b>	Se logra modificar el registro sin ningún problema.

Tabla 50.

*Prueba de unidad ManejadorModuloTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorModuloTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_MODULO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 51.

*Prueba de unidad ManejadorModuloTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorModuloTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_MODULO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 52.

*Prueba de unidad ManejadorNumeroTelefonoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorNumeroTelefonoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_NUMEROTELEFONO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 53.

*Prueba de unidad ManejadorNumeroTelefonoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorNumeroTelefonoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_NUMEROTELEFONO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 54.

*Prueba de unidad ManejadorObjetoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObjetoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OBJETO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 55.

*Prueba de unidad ManejadorObjetoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObjetoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OBJETO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 56.

*Prueba de unidad ManejadorObjetoRequerimientoTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObjetoRequerimientoTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OBJETOREQUERIMIENTO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 57.

*Prueba de unidad ManejadorObjetoRequerimientoTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObjetoRequerimientoTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OBJETOREQUERIMIENTO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 58.

*Prueba de unidad ManejadorObservacionTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObservacionTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OSERVACION. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 59.

*Prueba de unidad ManejadorObservacionTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorObservacionTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_OBSERVACION.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 60.

*Prueba de unidad ManejadorParametroTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorParametroTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PARAMETRO. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 61.

*Prueba de unidad ManejadorParametroTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorParametroTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PARAMETRO.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 62.

*Prueba de unidad ManejadorPersonalTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONAL. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 63.

*Prueba de unidad ManejadorPersonalTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONAL.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 64.

*Prueba de unidad ManejadorPersonalTest 3.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTest
<b>Método a probar:</b>	verificarLogin()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 2 string, Mail, Clave.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia al verificar el usuario.
<b>Comentarios:</b>	Se logra procesar sin problemas.

Tabla 65.

*Prueba de unidad ManejadorPersonalTest 4.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTest
<b>Método a probar:</b>	enviarMail ()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 2 Objeto Entity TBL_PERSONAL, TBL_TICKET. - 1 string OP.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 66.

*Prueba de unidad ManejadorPersonalSubAreaTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalSubAreaTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONALSUBAREA. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 67.

*Prueba de unidad ManejadorPersonalSubAreaTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalSubAreaTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONALSUBAREA.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

Tabla 68.

*Prueba de unidad ManejadorPersonalTicketTest.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTicketTest
<b>Método a probar:</b>	Obtener()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta del método y su eficiencia.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONALTICKET. - 1 Variable String. OP
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor con eficacia.
<b>Comentarios:</b>	Se logra obtener el registro solicitado sin problemas.

Tabla 69.

*Prueba de unidad ManejadorPersonalTicketTest 2.*

Prueba de Unidad	
<b>Identificador de la prueba</b>	ManejadorPersonalTicketTest
<b>Método a probar:</b>	Guardar()
<b>Objetivo de la prueba:</b>	Lograr identificar el tiempo de respuesta y estabilidad del método al crear el registro.
<b>Datos de Entrada:</b>	- 1 Objeto Entity TBL_PERSONALTICKET.
<b>Resultados esperados:</b>	Se espera una respuesta adecuada del servidor al crear un nuevo registro correctamente.
<b>Comentarios:</b>	Se logra guardar el registro sin ningún problema.

## 5.04 Especificación de pruebas de aceptación.

Tabla 70

### *Especificación de la prueba de aceptación EPA001*

<b>Identificador de la prueba:</b>	PRU001
<b>Caso de uso:</b>	C.U 1
<b>Tipo de usuario:</b>	Cliente, direccionador
<b>Objetivo de la prueba:</b>	Creación de usuario.
<b>Sugerencia de eventos:</b>	<ul style="list-style-type: none"> <li>* Ingreso al módulo de direccionador.</li> <li>* Escogemos el cliente final.</li> <li>* Seleccionamos el tipo de direccionamiento que vamos a usar.</li> <li>* Agregamos objetos, archivos adjuntos y observación al ticket.</li> <li>* Enviar ticket a personal de soporte asignado.</li> </ul>
<b>Resultados esperados:</b>	<ul style="list-style-type: none"> <li>* El ticket se crea con la información completa.</li> <li>* Al ser direccionamiento automático se debe asignar al personal de soporte con menos carga laboral.</li> <li>* Se deberá mandar correo al cliente final y al cliente que autoriza el ticket de ser necesario.</li> </ul>
<b>Comentario:</b>	Al crear el ticket se debe especificar un estado y una etapa automáticamente.
<b>Estado:</b>	Aceptado.

Tabla 71

### *Especificación de la prueba de aceptación EPA002*

<b>Identificador de la prueba:</b>	PRU002
<b>Caso de uso:</b>	C.U 2
<b>Tipo de usuario:</b>	Cliente, direccionador, soporte
<b>Objetivo de la prueba:</b>	Administración de tickets.
<b>Sugerencia de eventos:</b>	<ul style="list-style-type: none"> <li>* Ingreso al módulo de soporte</li> <li>* Escogemos el ticket a administrar.</li> <li>* Se carga los datos del ticket.</li> <li>* Cambiamos los parámetros disponibles en cada ticket.</li> <li>* Enviar ticket a personal de soporte asignado.</li> </ul>
<b>Resultados esperados:</b>	<ul style="list-style-type: none"> <li>* El ticket se crea con la información completa.</li> <li>* Al ser direccionamiento automático se debe asignar al personal de soporte con menos carga laboral.</li> <li>* Se deberá mandar correo al cliente final y al cliente que autoriza el ticket de ser necesario.</li> </ul>
<b>Comentario:</b>	Al crear el ticket se debe especificar un estado y una etapa automáticamente.
<b>Estado:</b>	Aceptado.

Tabla 72

*Especificación de la prueba de aceptación EPA003*

<b>Identificador de la prueba:</b>	PRU003
<b>Caso de uso:</b>	C.U 1
<b>Tipo de usuario:</b>	Direccionador
<b>Objetivo de la prueba:</b>	Creación de usuario.
<b>Sugerencia de eventos:</b>	<ul style="list-style-type: none"> <li>* Ingreso al módulo de direccionador.</li> <li>* Escogemos el cliente final.</li> <li>* Seleccionamos el tipo de direccionamiento que vamos a usar.</li> <li>* Agregamos objetos, archivos adjuntos y observación al ticket.</li> <li>* Enviar ticket a personal de soporte asignado.</li> </ul>
<b>Resultados esperados:</b>	<ul style="list-style-type: none"> <li>* El ticket se crea con la información completa.</li> <li>* Al ser direccionamiento automático se debe asignar al personal de soporte con menos carga laboral.</li> <li>* Se deberá mandar correo al cliente final y al cliente que autoriza el ticket de ser necesario.</li> </ul>
<b>Comentario:</b>	Al crear el ticket se debe especificar un estado y una etapa automáticamente.
<b>Estado:</b>	Aceptado.

Tabla 73

*Especificación de la prueba de aceptación EPA004*

<b>Identificador de la prueba:</b>	PRU004
<b>Caso de uso:</b>	C.U 1
<b>Tipo de usuario:</b>	Direccionador
<b>Objetivo de la prueba:</b>	Creación de usuario.
<b>Sugerencia de eventos:</b>	<ul style="list-style-type: none"> <li>* Ingreso al módulo de direccionador.</li> <li>* Escogemos el cliente final.</li> <li>* Seleccionamos el tipo de direccionamiento que vamos a usar.</li> <li>* Agregamos objetos, archivos adjuntos y observación al ticket.</li> <li>* Enviar ticket a personal de soporte asignado.</li> </ul>
<b>Resultados esperados:</b>	<ul style="list-style-type: none"> <li>* El ticket se crea con la información completa.</li> <li>* Al ser direccionamiento automático se debe asignar al personal de soporte con menos carga laboral.</li> <li>* Se deberá mandar correo al cliente final y al cliente que autoriza el ticket de ser necesario.</li> </ul>
<b>Comentario:</b>	Al crear el ticket se debe especificar un estado y una etapa automáticamente.
<b>Estado:</b>	Aceptado.

Tabla 74

*Especificación de la prueba de aceptación EPA005*

<b>Identificador de la prueba:</b>	PRU005
<b>Caso de uso:</b>	C.U 1
<b>Tipo de usuario:</b>	Direccionador
<b>Objetivo de la prueba:</b>	Creación de usuario.
<b>Sugerencia de eventos:</b>	<ul style="list-style-type: none"> <li>* Ingreso al módulo de direccionador.</li> <li>* Escogemos el cliente final.</li> <li>* Seleccionamos el tipo de direccionamiento que vamos a usar.</li> <li>* Agregamos objetos, archivos adjuntos y observación al ticket.</li> <li>* Enviar ticket a personal de soporte asignado.</li> </ul>
<b>Resultados esperados:</b>	<ul style="list-style-type: none"> <li>* El ticket se crea con la información completa.</li> <li>* Al ser direccionamiento automático se debe asignar al personal de soporte con menos carga laboral.</li> <li>* Se deberá mandar correo al cliente final y al cliente que autoriza el ticket de ser necesario.</li> </ul>
<b>Comentario:</b>	Al crear el ticket se debe especificar un estado y una etapa automáticamente.
<b>Estado:</b>	Aceptado.

**5.05 Especificación de pruebas de carga.**

Tabla 75

*Especificación de la prueba de carga*

<b>Identificador de Prueba:</b>	EPC001
<b>Tipo de prueba:</b>	Simulación de toma de pruebas de admisión
<b>Objetivo de la prueba:</b>	Probar si al dar la prueba de admisión se cuelga o se daña la página.
<b>Descripción:</b>	Se realizó la prueba de admisión a los estudiantes de octavos de básica para validar si el sistema no se cuelga al rendir la prueba de admisión con varios estudiantes.
<b>Resultados Esperados:</b>	Página web sin problemas.
<b>Comentarios:</b>	La página web no se cuelga al rendir 50 estudiantes de octavos de básica la prueba de admisión, realizándola en la intranet.

**5.06 Configuración del ambiente mínimo/ideal.**

Para el desarrollo de la aplicación usaremos las siguientes herramientas

En primer lugar nos dirigiremos a la siguiente dirección desde donde nos descargaremos la versión gratuita del visual studio 2010 express para el lenguaje c# sharp

<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-csharp-express>

1) Una vez en la web nos aparecerá la siguiente pagina donde escogeremos el idioma y pulsaremos en install now

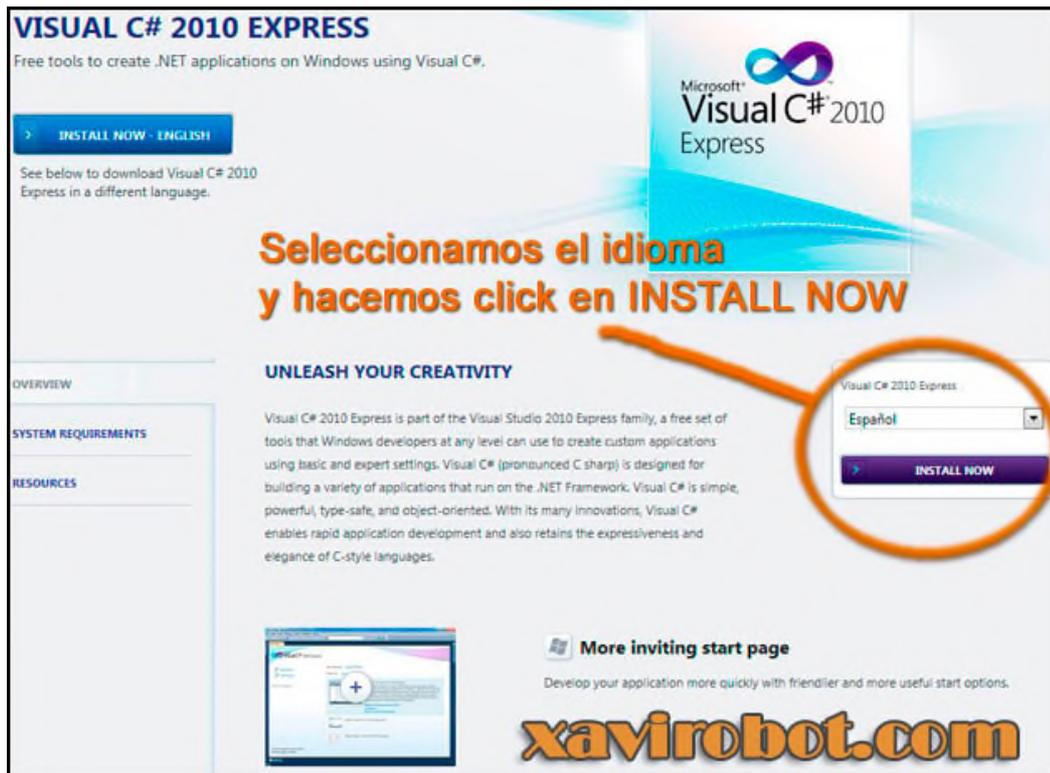


Figura 44. Instalación Visual Studio

2) Una vez pulsado install now nos aparecerá la siguiente pantalla y pulsaremos en la segunda opción Install Visual Studio 2010 Express C# para comenzar la descarga.



Figura 45. Instalación Visual Studio

3) Una vez descargada la aplicación la ejecutaremos para su instalación y nos aparecerá la siguiente pantalla donde antes de pulsar en siguiente deseleccionaremos el checkbox en el caso que no deseemos reportar información sobre la instalación a Microsoft.



Figura 46. Instalación Visual Studio

4) En esta pantalla aceptaremos los términos y condiciones y pulsamos en siguiente.

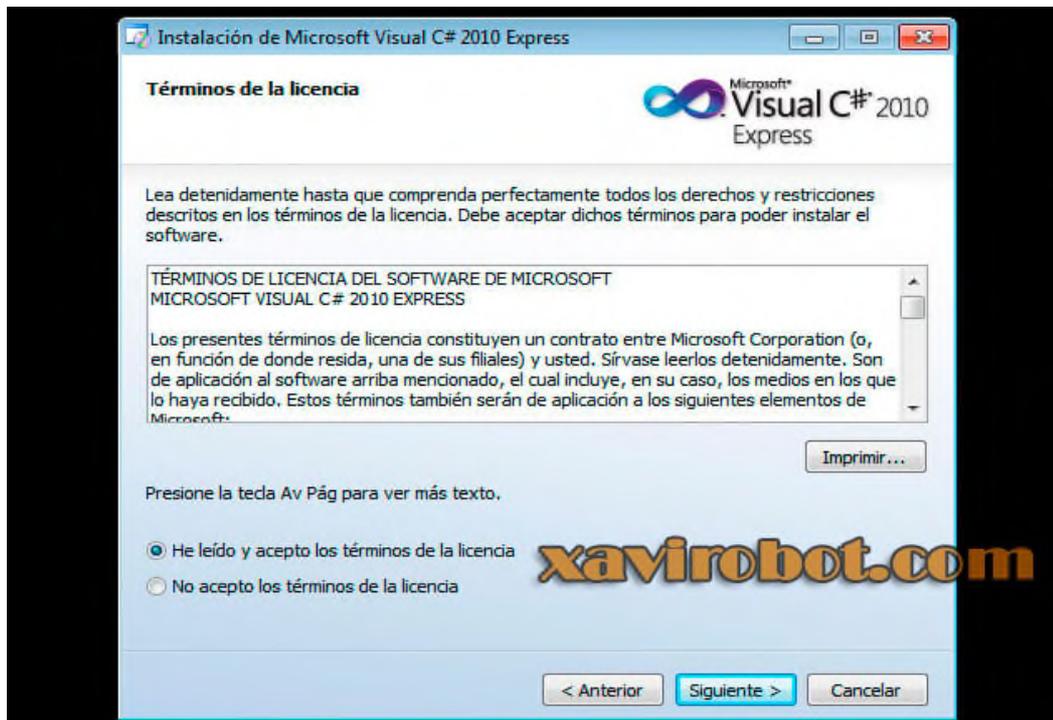


Figura 47. Instalación Visual Studio

5) En esta pantalla nos indica la ruta de instalación y los productos necesarios para la instalación que se descargarán, pulsamos en instalar.

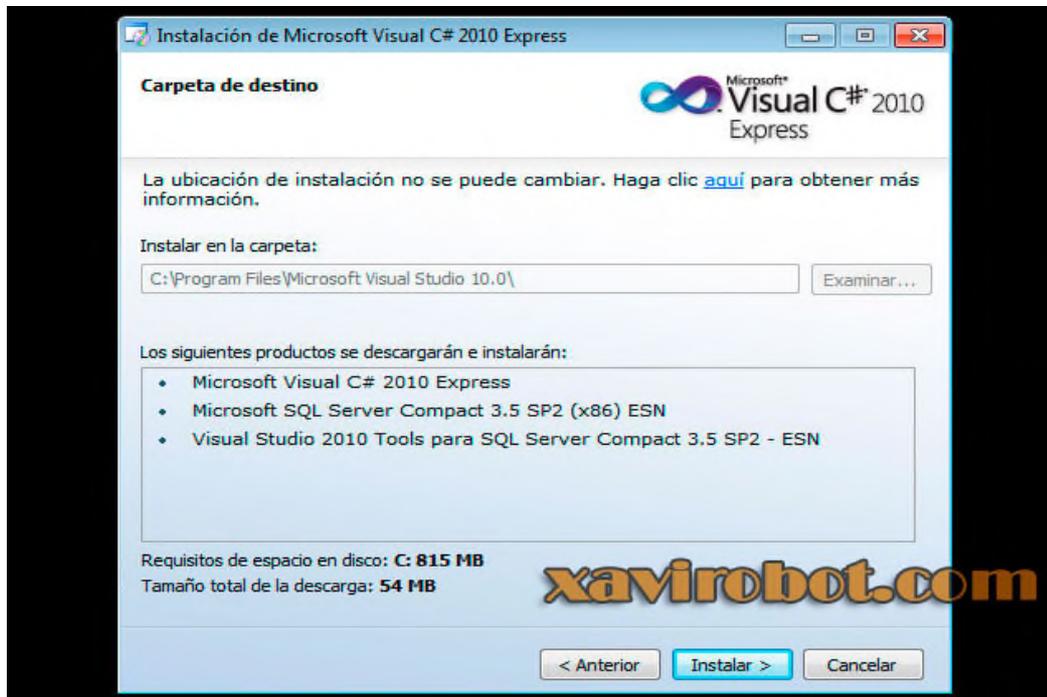


Figura 48. Instalación Visual Studio

6) En esta pantalla empezara el progreso de descarga de los productos

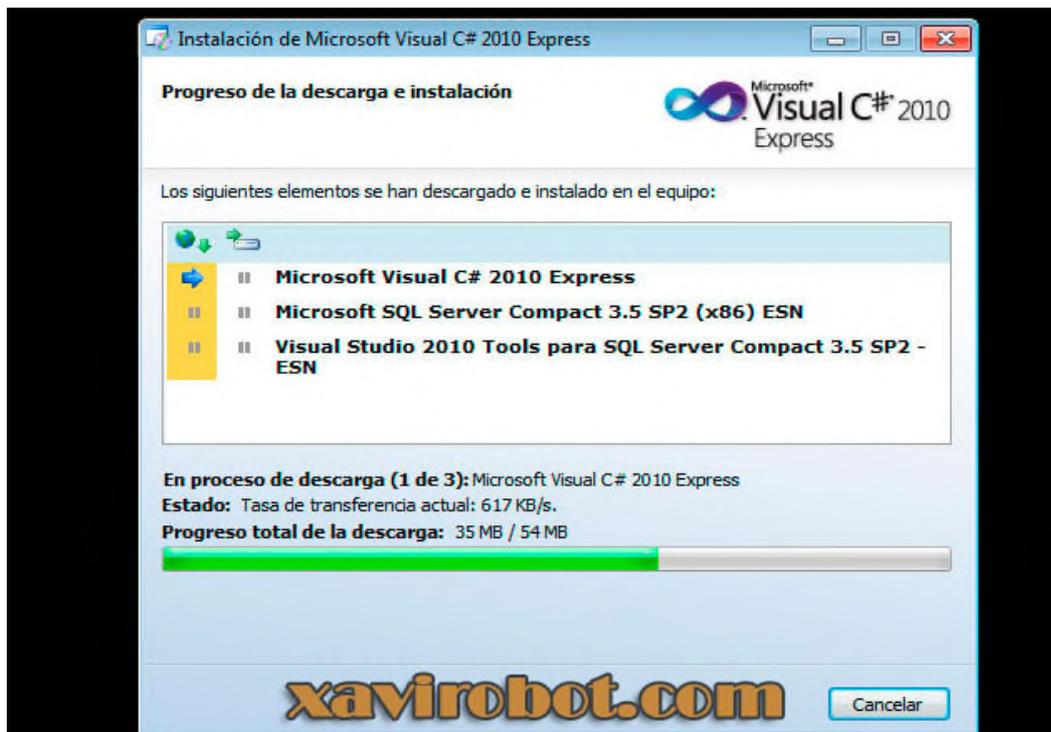


Figura 49. Instalación Visual Studio

7) Una vez descargado e instalado nos aparecerá la siguiente pantalla indicándonos que la instalación ha sido completada y pulsamos en salir.



Figura 50. Instalación Visual Studio

8) Una vez instalado ejecutamos la aplicación y nos dirigiremos al menú de herramientas de la parte superior de visual studio y pulsaremos en el botón de ayuda y nuevamente en el botón de registrar producto. Aunque la aplicación es gratuita es necesario registrarse para que el producto no expire.



Figura 51. Instalación Visual Studio, registro del producto 1.

9) Una vez pulsado el botón de registrar producto nos aparecerá una nueva pantalla donde se nos pedirá la clave del producto, como seguramente no dispondremos de la clave pulsaremos en obtener una clave de registro en línea.

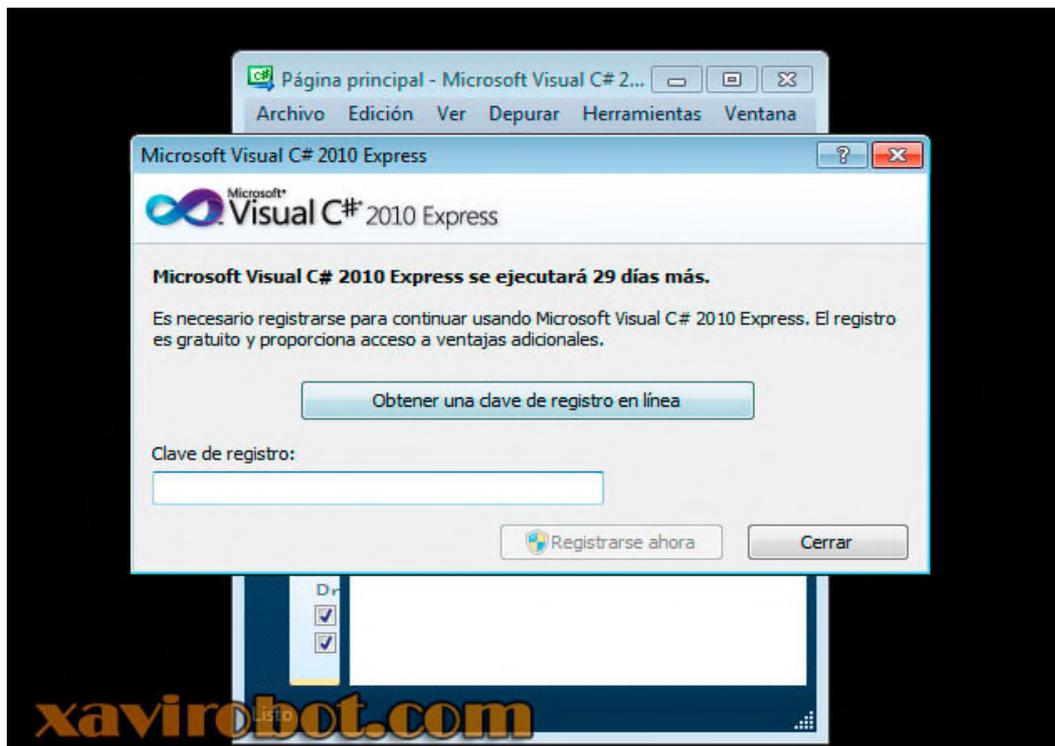


Figura 52. Instalación Visual Studio, registro del producto 2.

10) Una vez pulsado el botón de obtener clave nos enviara a una dirección web de Microsoft donde deberemos introducir una cuenta de correo de Hotmail para identificarnos.

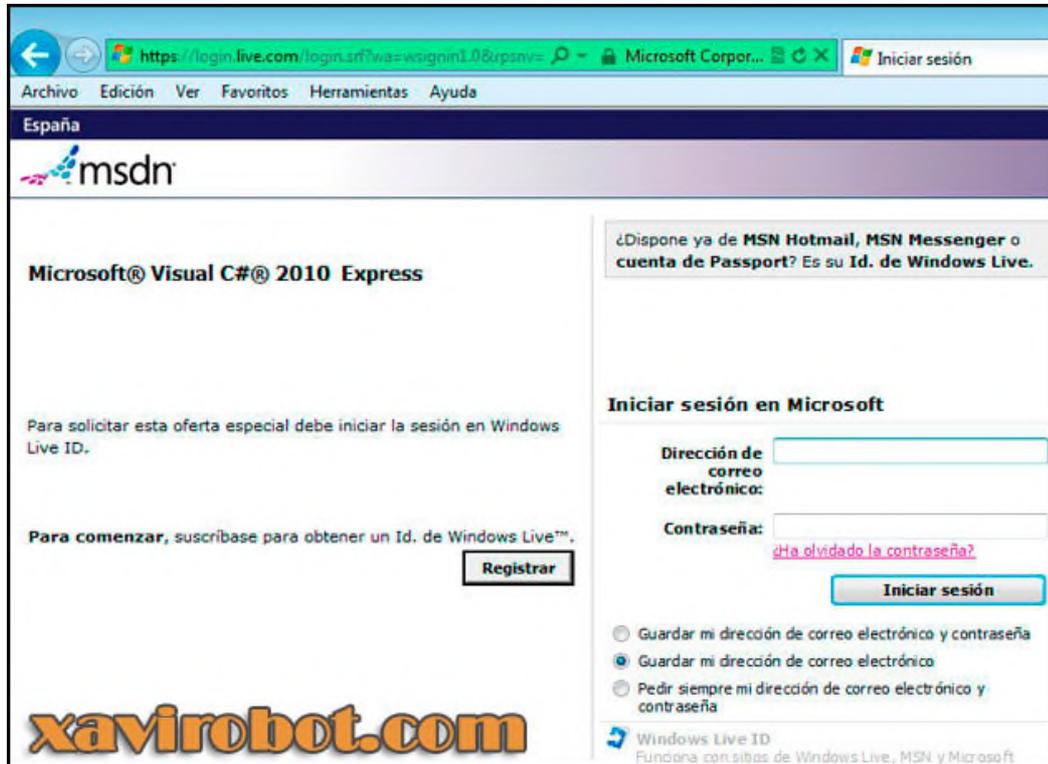


Figura 53. Instalación Visual Studio, registro del producto 3.

11) Una vez identificados nos aparecerá una página donde deberemos rellenar nuestros datos y contestar algunas preguntas.

Una vez cumplimentado el formulario pulsaremos aceptar y la clave nos será enviada al correo electrónico, ya solo deberemos introducirla en la pantalla de registrar producto para poder utilizar nuestro visual studio 2010 express sin miedo a que expire.

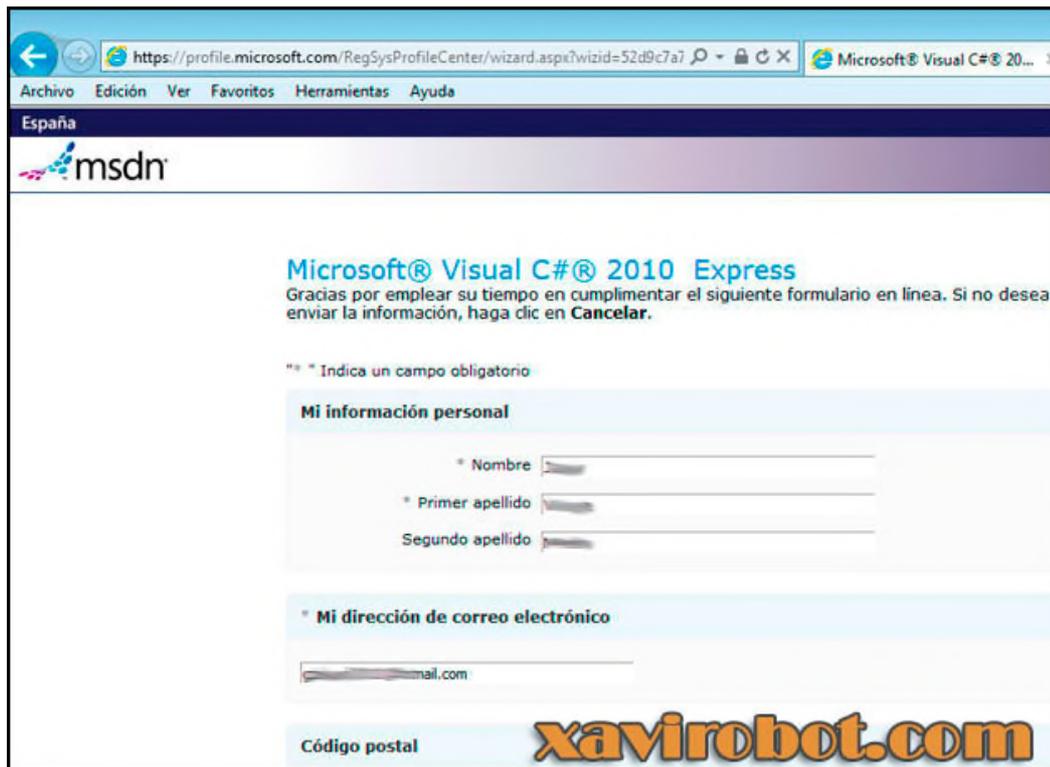


Figura 54. Instalación Visual Studio, registro del producto 4.

12) Una vez tengamos la aplicación instalada y registrada necesitaremos instalar la última versión del framework para que nuestros proyectos puedan correr sin problemas. Para instalar la última versión del framework 4 no dirigiremos a la página de Microsoft.

Una vez estemos en la web pulsaremos en el botón descargar y cuando dispongamos de la descarga simplemente ejecutamos la instalación del framework y listos una vez instalado ya podréis empezar a programar vuestro primer Hola Mundo en C# Sharp con Visual Studio 2010 express.

Para la administración de la base de datos creada en sql server vamos a usar sql server management, a continuación, los pasos a seguir para instalar el producto.

Dirigirse a la ubicación del SQL Server 2012.

Ejecutar SETUP.EXE. Como estoy como usuario, voy a correr el archivo con privilegios elevados. Para ello, doy clic derecho sobre el programa SETUP.EXE y en las opciones que aparecen, selecciono "Run as administrator".

Aparece la siguiente ventana:



Figura 55. Instalación sql server primer paso.

En el panel izquierdo, ir a "Installation" y dar clic en la opción "New SQL Server stand-alone installation or add features to an existing installation":

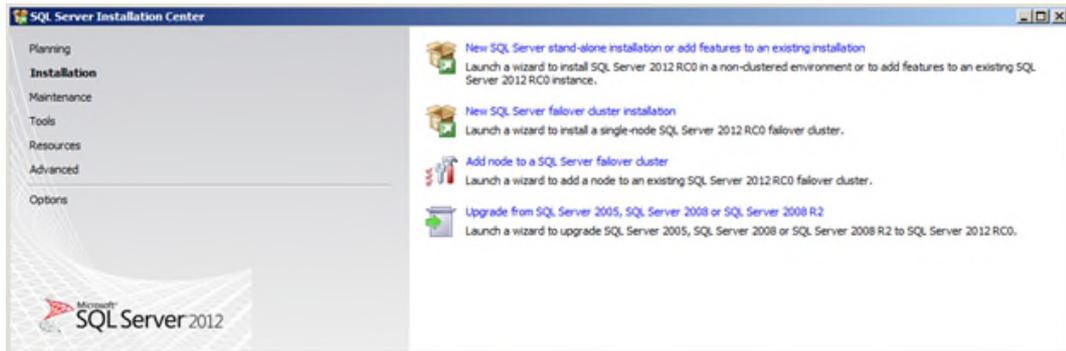


Figura 56. Instalación sql server segundo paso.

Aparece la siguiente ventana:

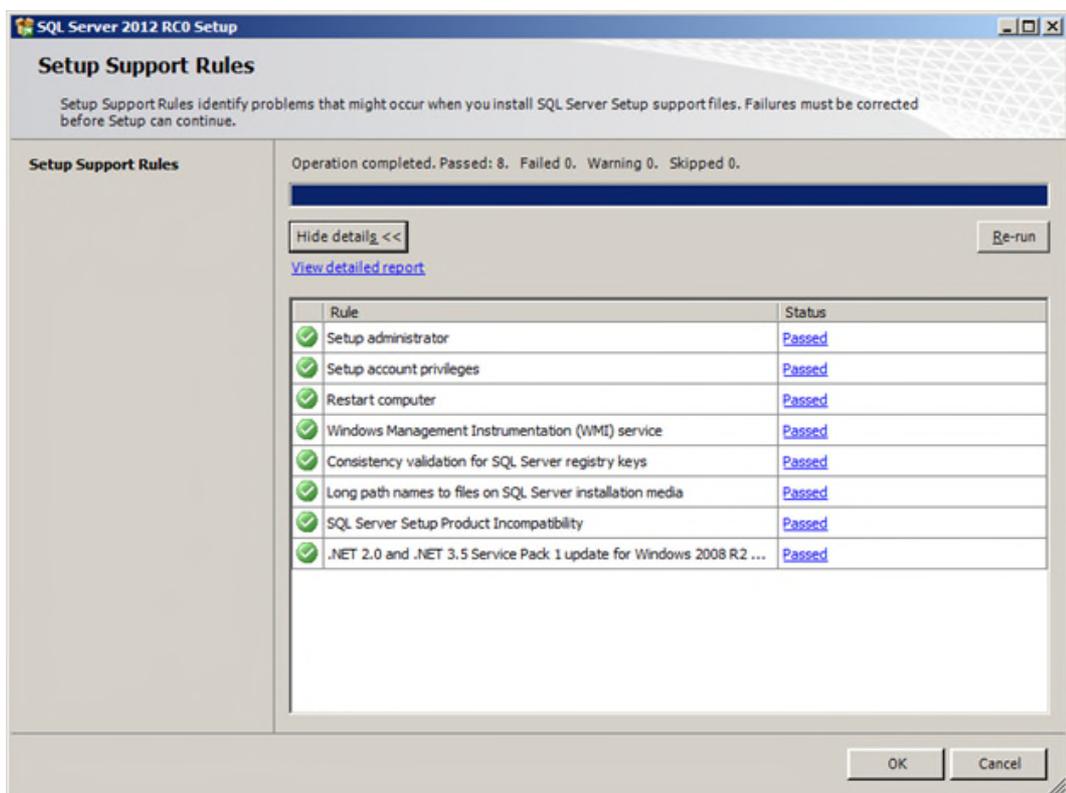


Figura 57. Instalación sql server tercer paso.

Presionar OK. Esperar que se procese la operación:



Figura 58. Instalación sql server cuarto paso.

Para este RC, no hay actualizaciones...

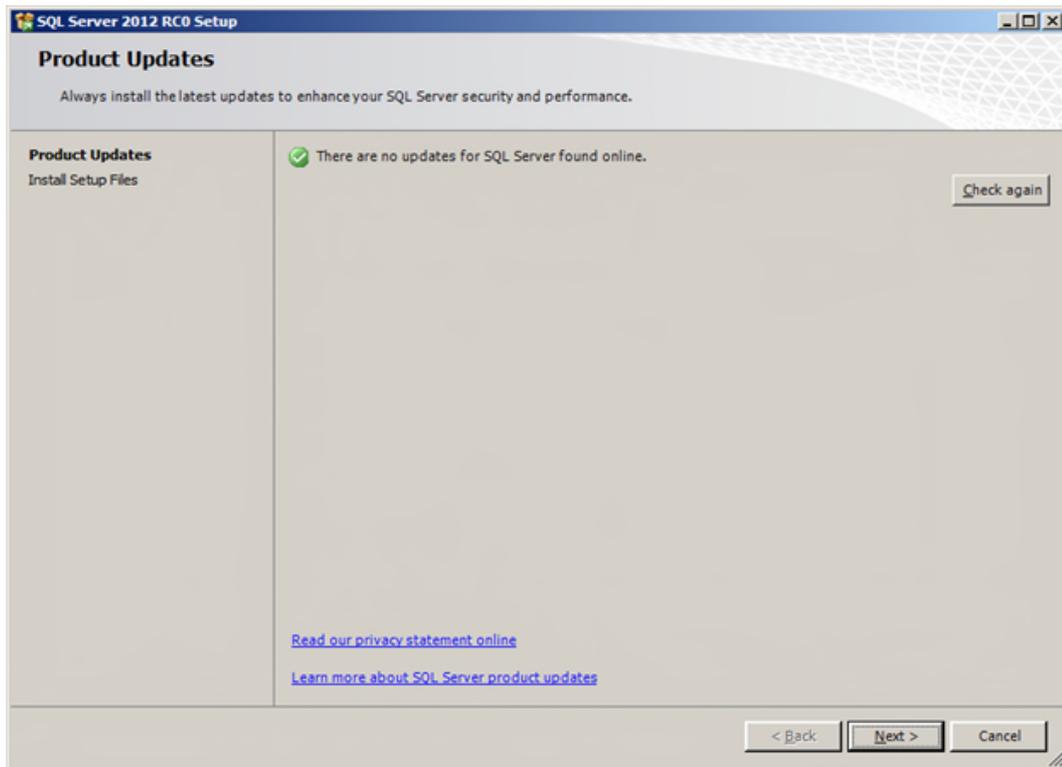


Figura 59. Instalación sql server quinto paso.

Presionar Next para que aparezca la siguiente pantalla:

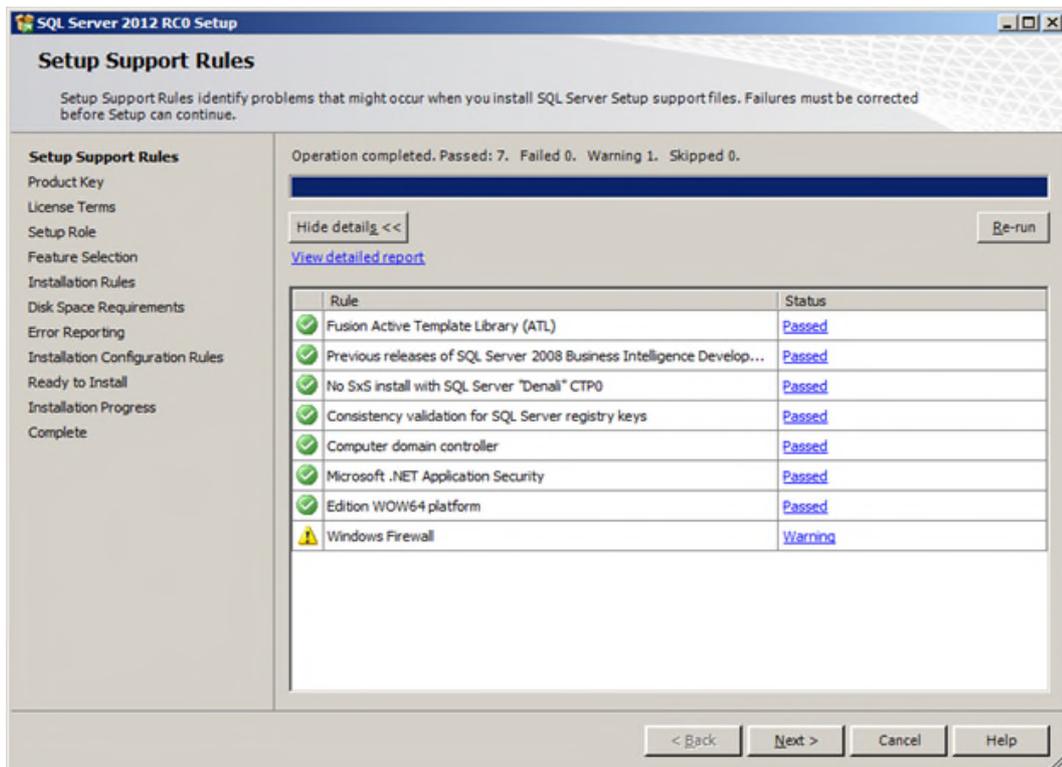


Figura 60. Instalación sql server sexto paso.

Como el Firewall está habilitado, sale ese aviso de precaución... Luego se indicará que hacer para que el Firewall esté debidamente configurado, sobre todo porque esta instalación contiene una instancia. Presionar Next para que aparezca esta pantalla:

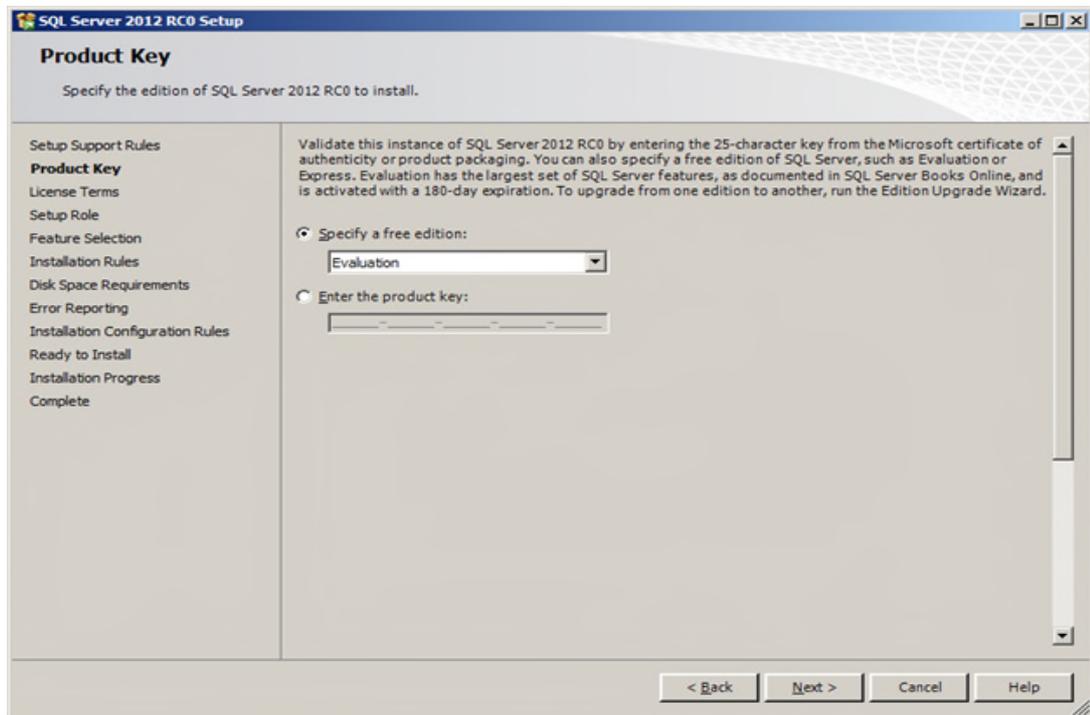


Figura 61. Instalación sql server séptimo paso.

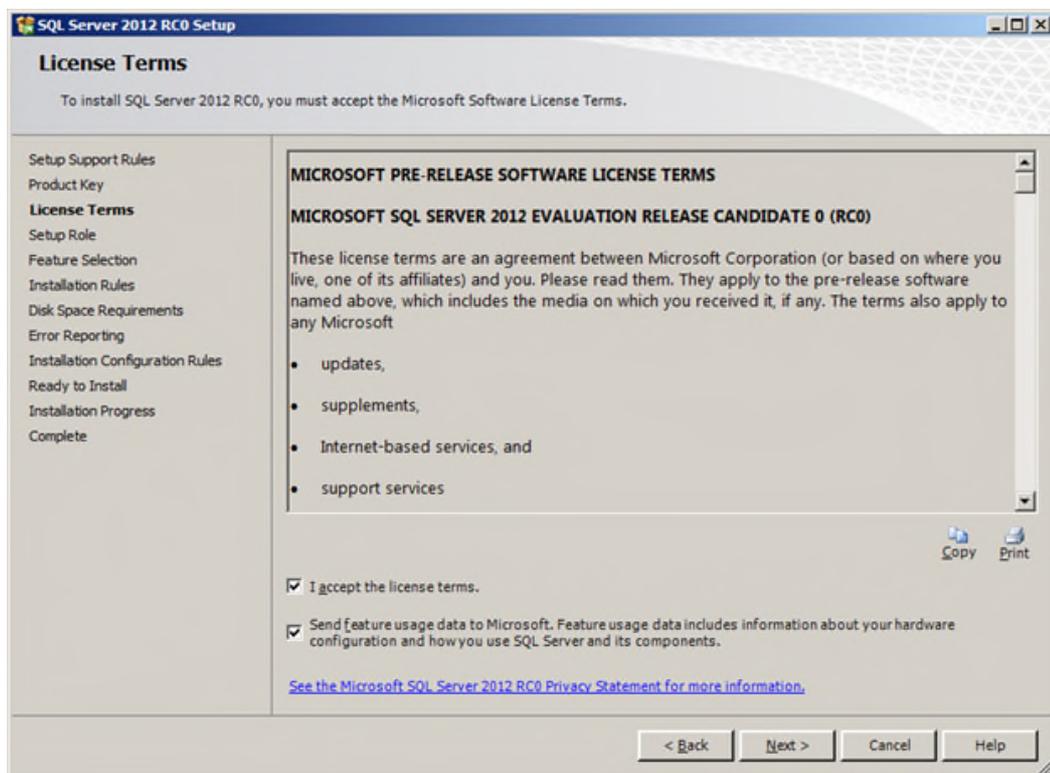


Figura 62. Instalación sql server octavo paso.

En esta pantalla se deben aceptar los términos de licenciamiento. Es opcional lo de enviar utilización del producto hacia Microsoft. Presionar Next

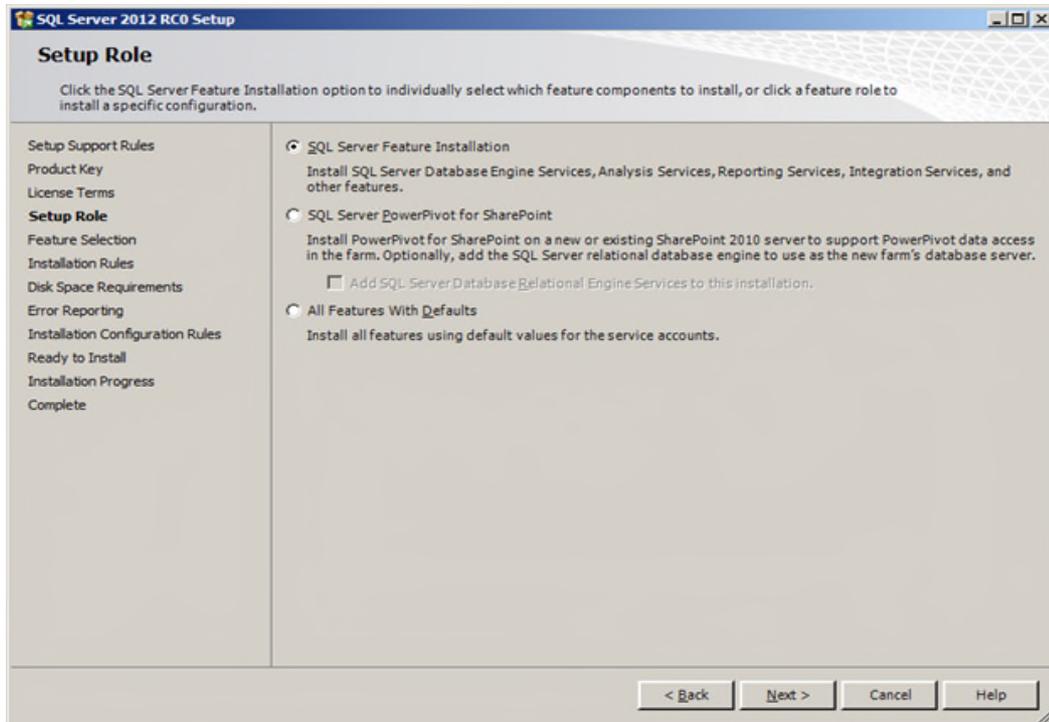


Figura 63. Instalación sql server noveno paso.

Seleccionar la primera opción, "SQL Server Feature Installation". Presionar Next

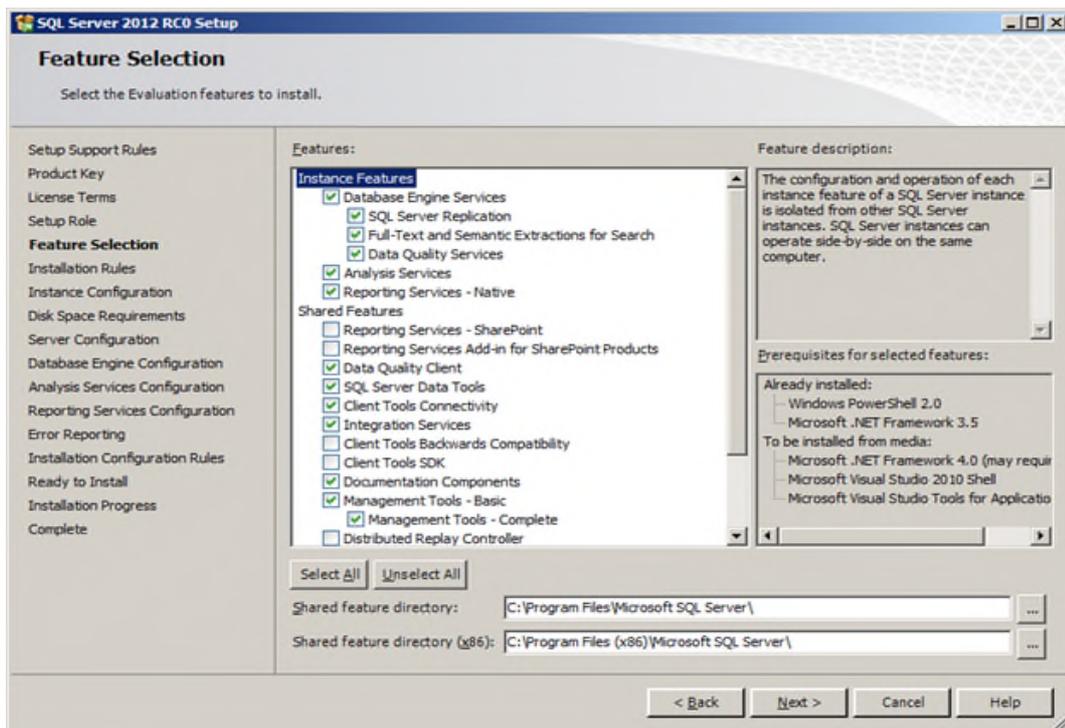


Figura 64. Instalación sql server decimo paso.

Las características a seleccionar son:

- Instance Features
  - Database Engine Services
    - SQL Server Replication
    - Full-Text and Semantic Extractions for Search
    - Data Quality Services
  - Analysis Services
  - Reporting Services - Native
- Shared Features
  - Reporting Services - SharePoint
  - Reporting Services Add-in for SharePoint Products
  - Data Quality Client
  - SQL Server Data Tools
  - Client Tools Connectivity
  - Integration Services
  - Client Tools Backwards Compatibility
  - Client Tools SDK
  - Documentation Components
  - Management Tools - Basic
    - Management Tools - Complete
  - Distributed Replay Controller
  - Distributed Replay Client
  - SQL Client Connectivity SDK
  - Master Data Services
- Redistributable Features

Figura 65. Instalación sql server onceavo paso.

Si al presionar Next, les aparece esta pantalla:

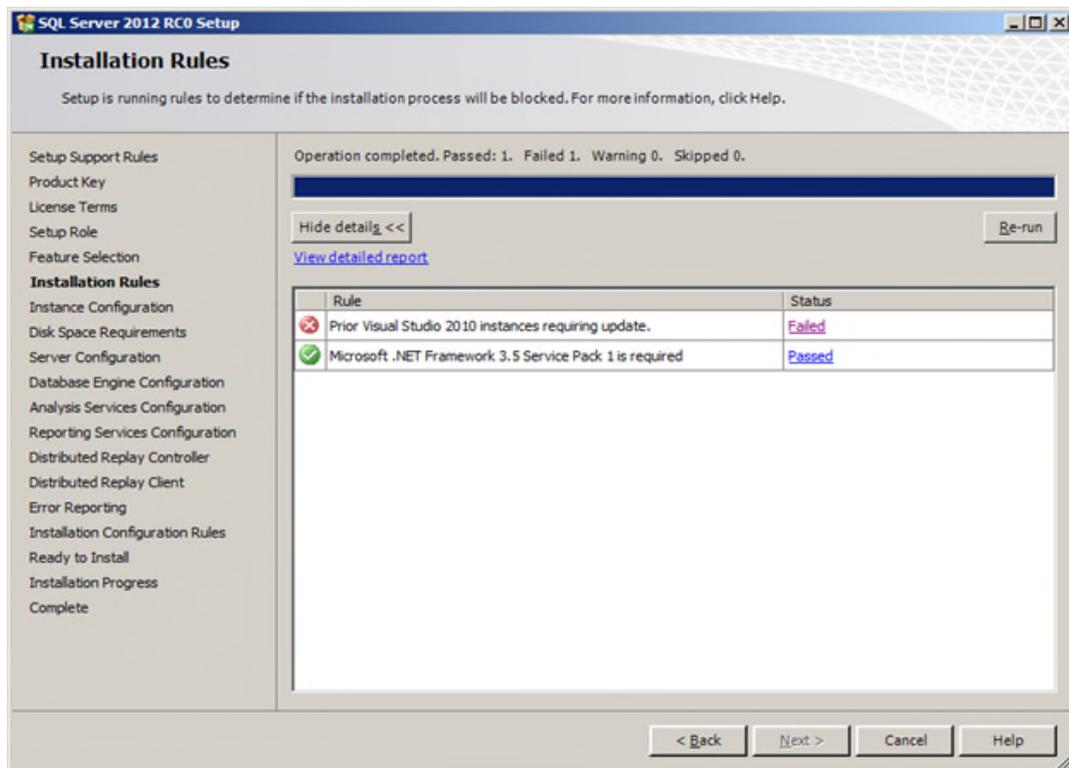


Figura 66. Instalación sql server doceavo paso.

Es porque esta versión de SQL Server requiere el SP1 de Visual Studio 2010; esto porque el encontró rastros de la instalación anterior, es decir la de CTP3.

Así que instalen el SP1 de Visual Studio 2010 para continuar con la instalación...

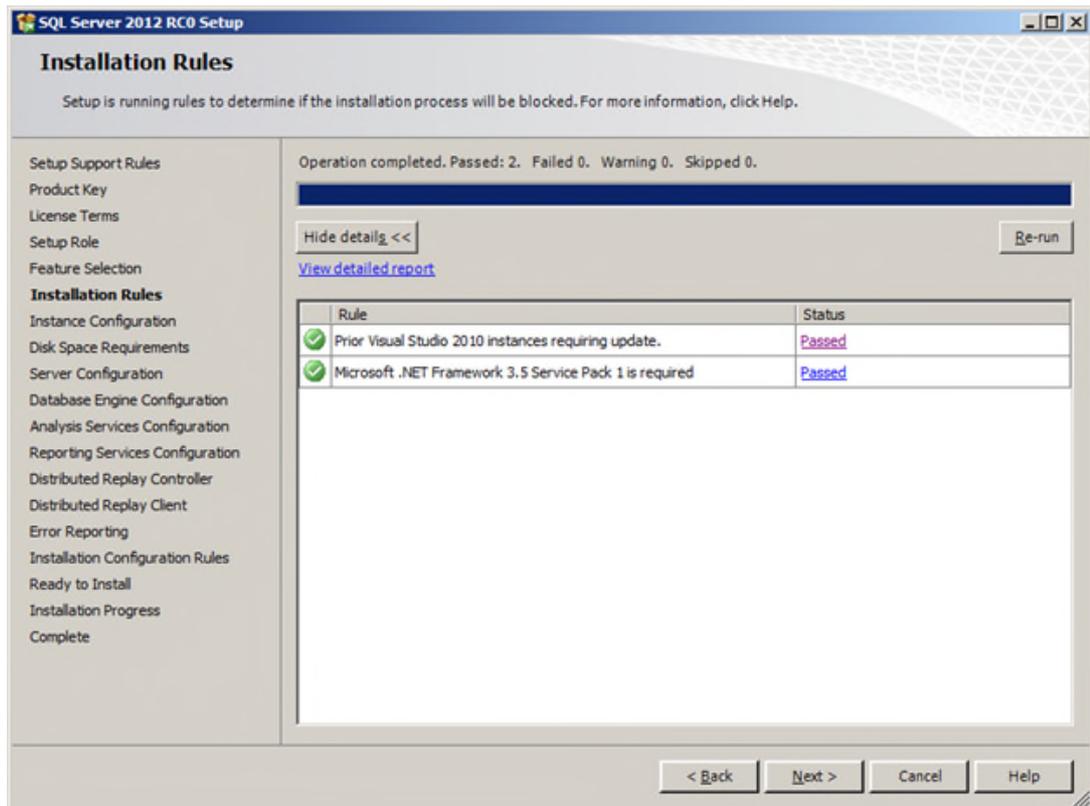


Figura 67. Instalación sql server treceavo paso.

Presionar Next

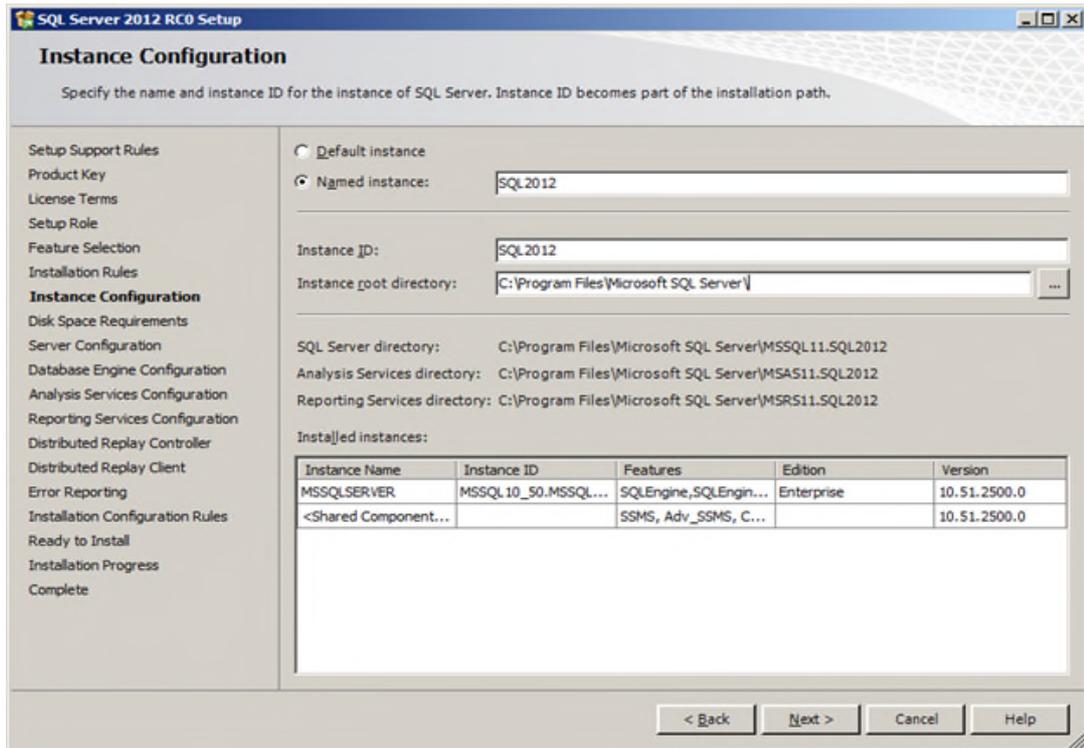


Figura 68. Instalación sql server catorceavo paso.

El nombre de la instancia es sugerido. Y particularmente generé una instancia porque tengo una instancia por omisión de SQL Server 2008 R2 SP1. Presionar Next

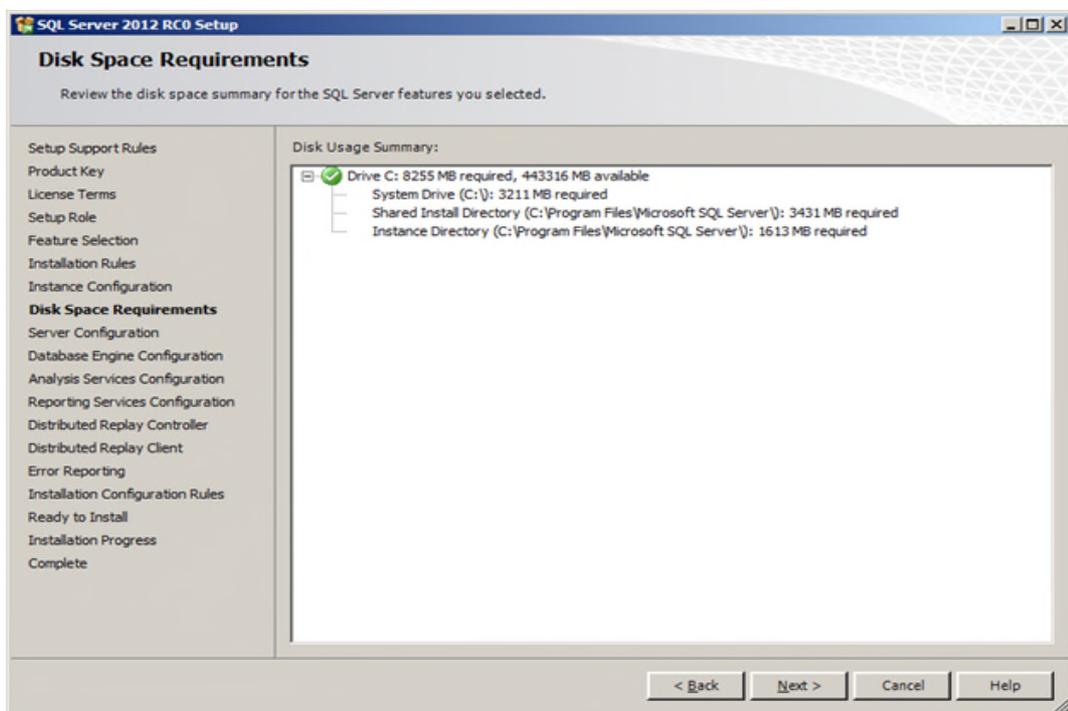


Figura 69. Instalación sql server quinceavo paso.

Presionar Next para que aparezca la pantalla de configuración del servidor

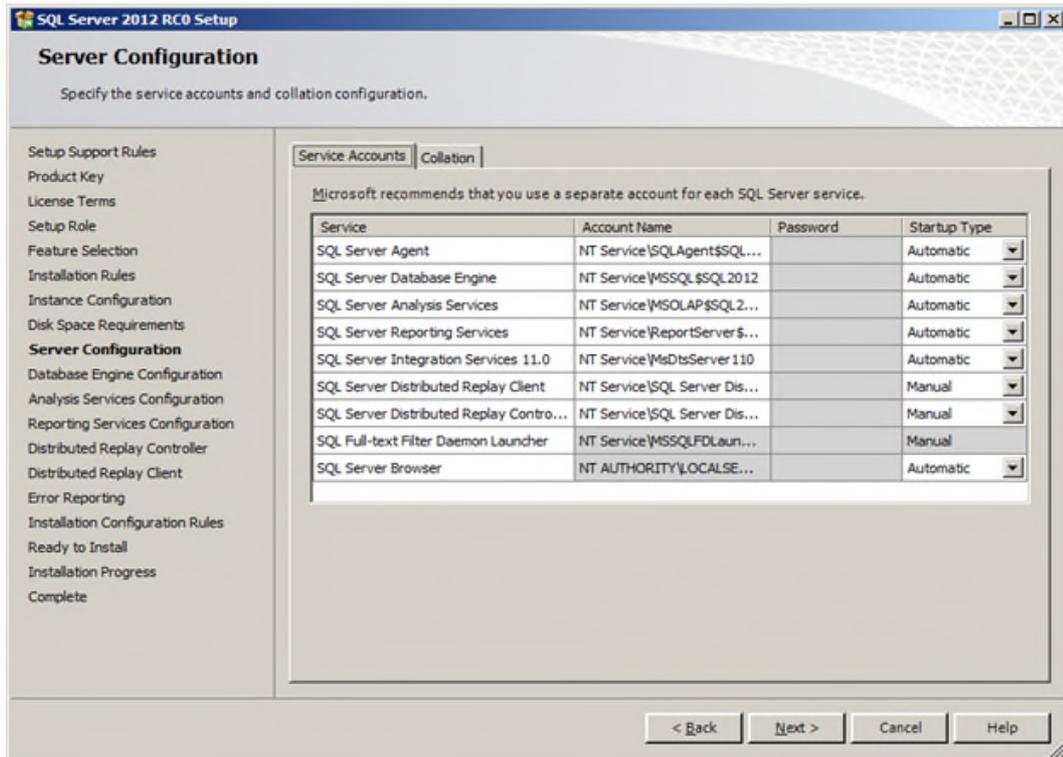


Figura 70. Instalación sql server dieciseisavo paso.

Presionar Next

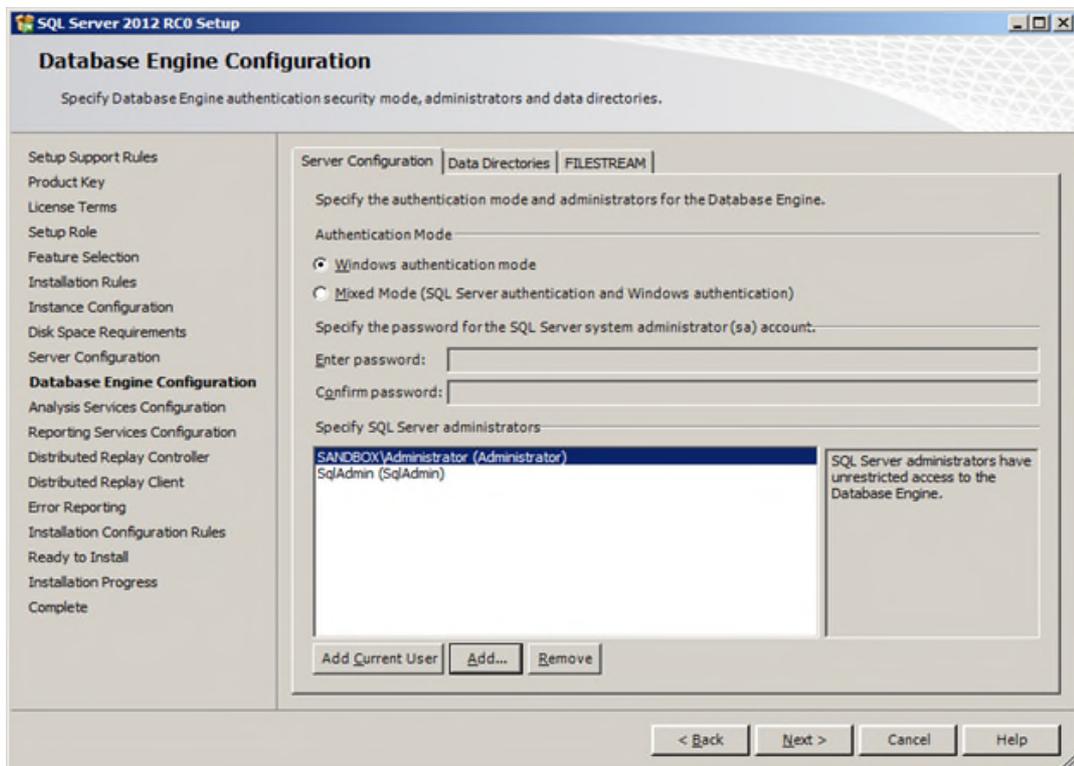


Figura 71. Instalación sql server diecisieteavo paso.

Además de agregar la cuenta de instalación, agrego una de administración; esto porque como mejor práctica la cuenta de instalación no debe quedar registrada como administradora... De hecho, es muy probable que sea incluso una cuenta creada específicamente para dicho fin, por lo cual después se elimina...

Se dejan los directorios de datos por omisión y se habilita FILESTREAM...

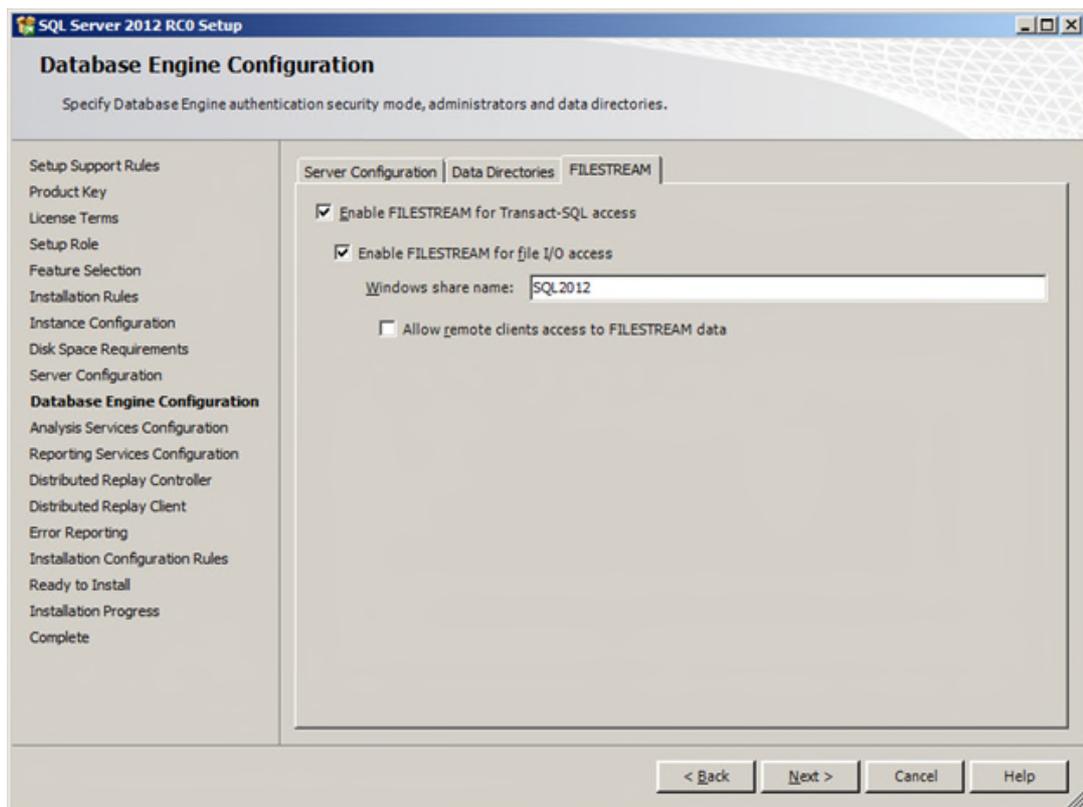


Figura 72. Instalación sql server dieciochoavo paso.

Presionar Next

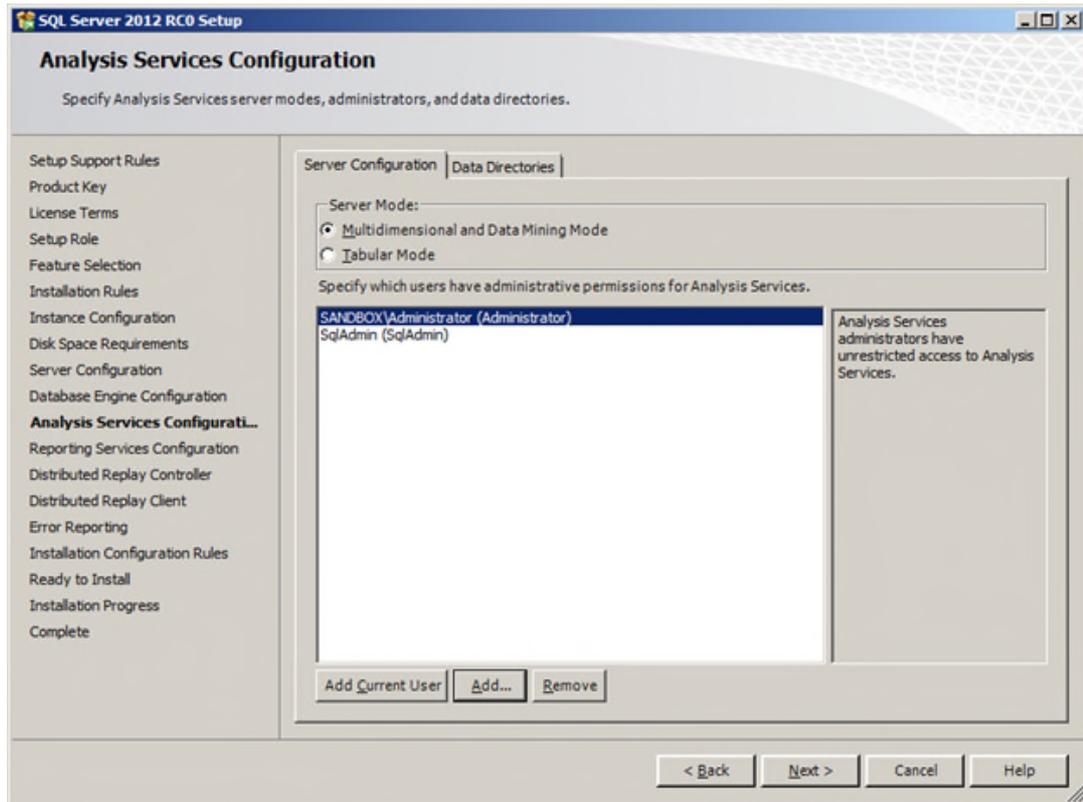


Figura 73. Instalación sql server diecinueveavo paso.

Seleccionar Multidimensional and Data Mining Mode... Y agregar las mismas cuentas de administración...

Luego de presionar Next, aparece la pantalla de configuración de SSRS. Seleccionar la opción de instalar y configurar.

Presionar Next

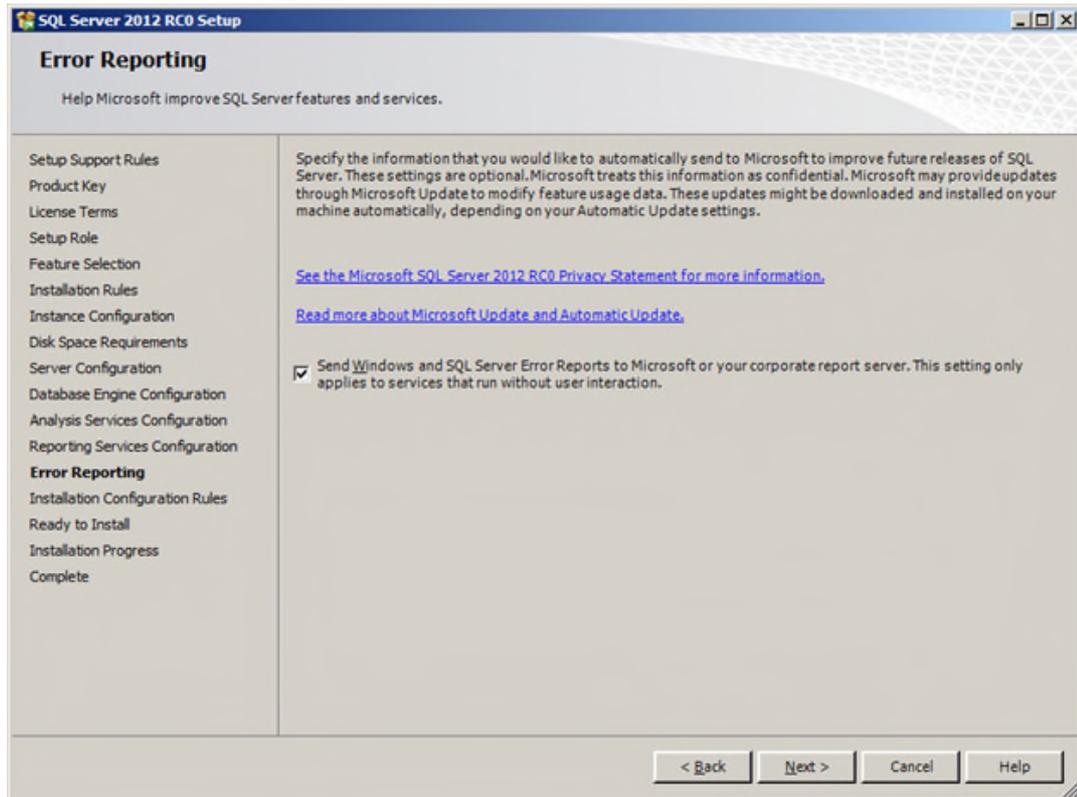


Figura 74. Instalación sql server veinteavo paso.

Presionar Next

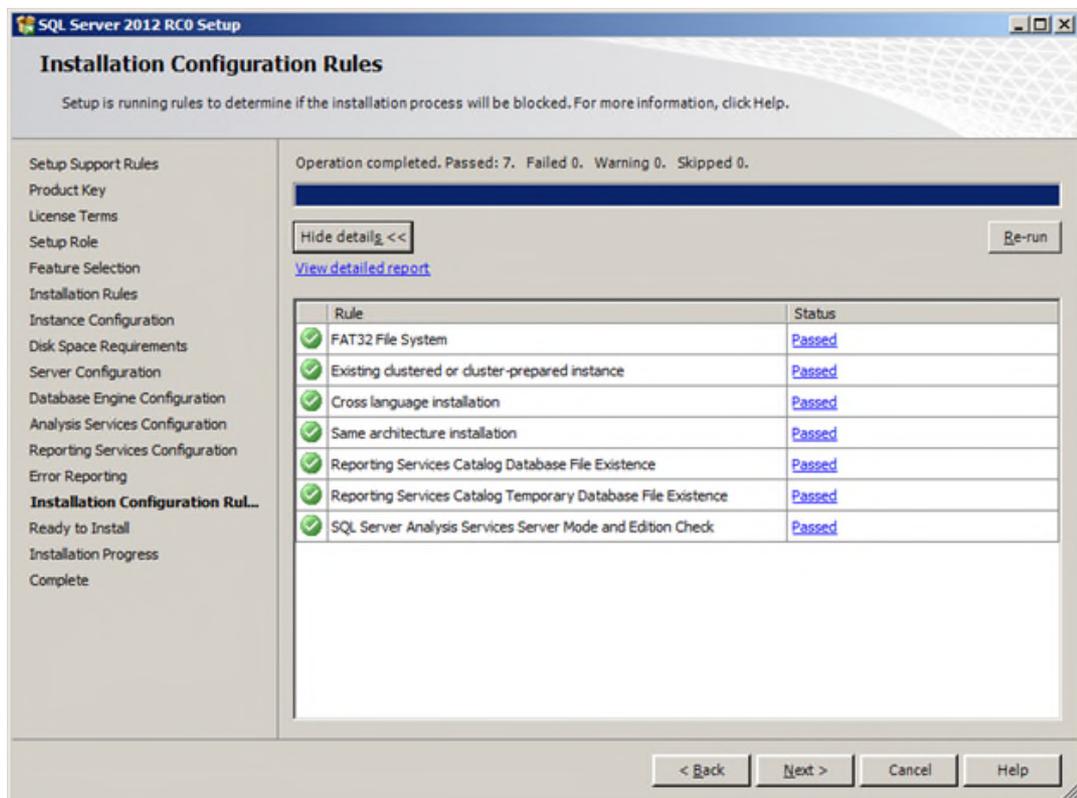


Figura 75. Instalación sql server veintiunavo paso.

Presionar Next para que aparezca la pantalla de "Ready to Install":

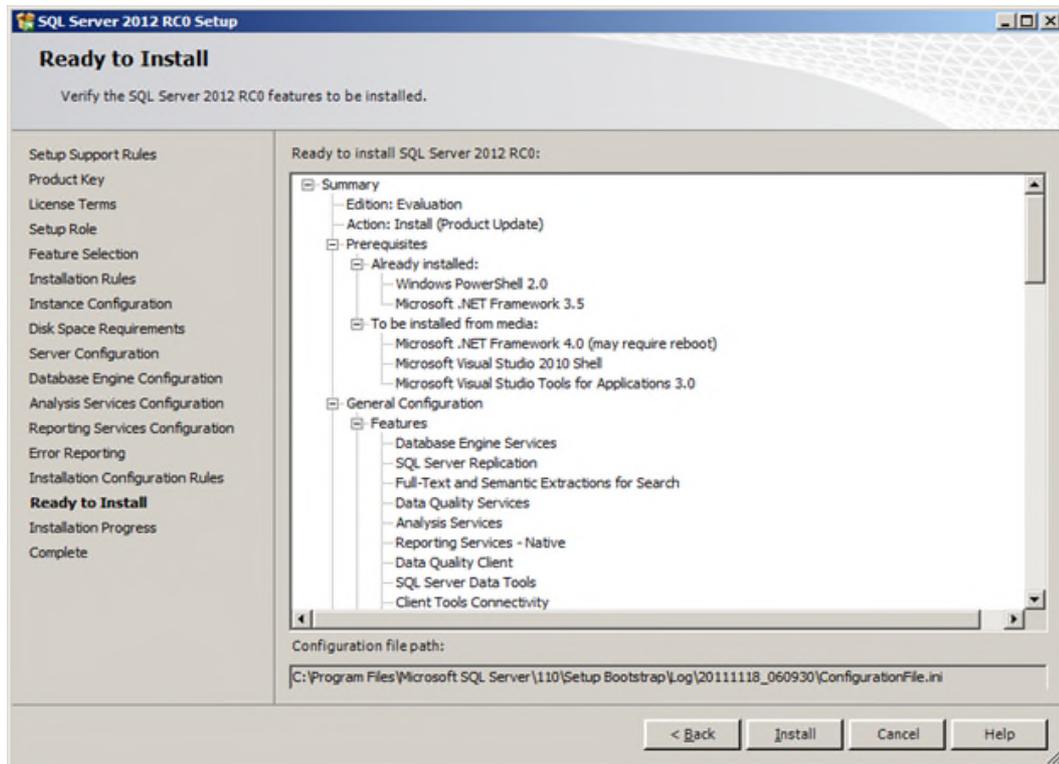


Figura 76. Instalación sql server veintidosavo paso.

Presionar Install y esperar a que termine el proceso...

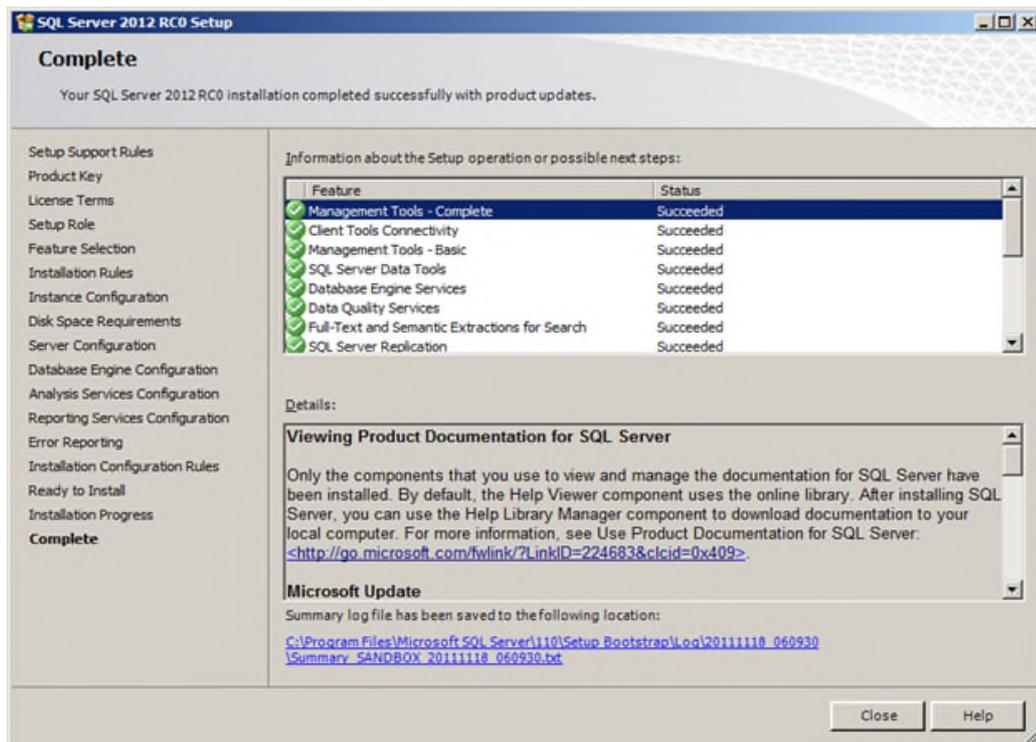


Figura 77. Instalación sql server veintitresavo paso.

Presionar Close.

Ya en el menú principal del servidor, aparecen las opciones para ejecutar SQL Server 2012 RC0:



Figura 78. Instalación sql server veinticuatroavo paso.



Figura 79. Instalación sql server veintisincoavo paso.

## Capítulo VI: Aspectos Administrativos

### 6.01 Recursos.

#### 6.01.01 Recursos humanos.

Se establecen las personas que influyen directa e indirectamente en la planificación y ejecución del proyecto.

Tabla 76.

*Especificación matriz de recursos humanos.*

N°	Descripción	Nombre	Función
1	Tutor	Ing. Johnny Coronel	Dirigir y corregir el proyecto de titulación.
2	Lector	Ing. Jaime Padilla	Revisión y aprobación del proyecto de titulación.
3	Auspiciante	CityTech.	Revisión y autorización de requerimientos.
4	Estudiante	Rubén Mosquera	Desarrollo de la aplicación.
5	Director de proyectos	Ing. Cristian Prado	Aprobar y registrar el tema del proyecto.
6	Gerente de infraestructura	Ing. Miguel Muñoz	Persona encargada de brindar la información necesaria para la parametrización básica de la aplicación.
7	Gerente de proyectos	Ing. Carlos Lagos	Persona encargada de realizar el seguimiento del proyecto.

#### 6.01.02 Recursos materiales.

Se especifica los recursos materiales a usar en el proyecto.

Tabla 77.

*Especificación matriz de recursos materiales.*

N°	Descripción
1	Corriente eléctrica.
2	Transporte.
3	Impresiones a color.
4	Anillados.
5	Empastado
6	Programa de profesionalización.

### 6.01.03 Recursos tecnológicos.

Son un medio que se vale de la tecnología para cumplir con su propósito. Los recursos tecnológicos pueden ser tangibles o intangibles.

Tabla 78

*Especificación matriz de recursos materiales.*

N°	Descripción
1	Computadora
2	Internet
3	Flash memory
4	Programa Visual Studio 2010
5	Programa Sql server management
6	Internet information service
7	Programa Microsoft Office
8	Programa Project

### 6.02 Presupuesto.

Tabla 79

*Presupuesto.*

N°	Descripción	Cantidad	Valor unitario	Valor total
1	Luz eléctrica			
2	Transporte	60	0,50	30,00
3	Impresiones blanco y negro	200	0,05	10,00
4	Impresiones a color	400	0,10	40,00
5	Anillado	3	3,50	10,50
6	Empastado	1	18,00	18,00
7	Programa de profesionalización	1	200,00	200,00
8	Computadora personal	1	500,00	500,00
9	Flash memory	1	10,00	10,00
<b>TOTAL</b>				<b>818.50</b>

### 6.03 Cronograma.

Véase anexo A.12

## Capítulo VII: Conclusiones y recomendaciones

### 7.01 Conclusiones.

Para optimizar el proceso de soporte técnico mediante una mesa de ayuda en la empresa Citytech, se implementó una aplicación informática que permita agilizar el proceso de administración y verificación de tickets por incidentes o requerimiento.

Se logró cumplir con cada uno de los requerimientos propuestos por los usuarios de la aplicación ya que se obtienen reportes y datos estadísticos.

El presupuesto que se necesitó para la implementación de la aplicación de help desk es de \$818,50 el cual fue cubierto por la empresa auspiciante y el estudiante.

### 7.02 Recomendaciones.

Se recomienda mantener actualizada la base de datos, en especial los activos fijos para evitar molestias en el cliente al no encontrar el activo necesario.

Mantener un seguimiento constante de los tickets para evitar que se mantenga abiertos o pendientes durante demasiado tiempo.

## A. Anexos

### A.01 Matriz T

Tabla 1, Matriz T

I = Intensidad PC = Potencial de cambio

Situación empeorada	Situación actual				Situación mejorada
Termino del contrato con los clientes, al existir demoras considerables al solucionar incidencias, por perdida de información o descoordinación.	Se maneja los tickets en una hoja de Excel que no da garantías de seguridad requeridas.				Crecimiento exponencial de la empresa al aumentar el número de clientes, gracias al buen servicio y la rapidez de atención.
Fuerzas impulsadoras	I	PC	I	PC	Fuerzas bloqueadoras
El personal de las diferentes áreas tiene el compromiso de entregar todo su esfuerzo para lograr mejorar sustancialmente la calidad de servicio.	4	4	4	4	El seguimiento inadecuado de los tickets, lo cual causa el aumento de tiempo de respuesta y la insatisfacción del usuario final.
La empresa cuenta con la infraestructura necesaria para brindar el servicio óptimo a los usuarios.	3	4	4	4	Aumento de la demanda de incidentes, lo que ocasiona la sensación de desatención.
Campañas de capacitación continuas para el usuario en temas tecnológicos.	3	2	3	2	Utilización errónea por parte del usuario final del hardware y software proporcionado por la empresa.

## A.02 Matriz de requerimientos.

Tabla 3.

*Matriz de requerimientos.*

MATRIZ DE REQUERIMIENTOS						
Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios involucrados
<b>REQUERIMIENTOS FUNCIONALES</b>						
RF001	La aplicación deberá administrar tickets y cada una de las actividades o etapas que se debe realizar para llegar a la solución del mismo, tomando en cuenta los tiempos de atención, el personal que lo realiza y cualquier observación que pueda realizar el mismo, además de controlar parámetros fundamentales como (estado, tipo de requerimiento, ticket padre, autorización, fecha de inicio, fecha final, usuarios que intervienen)	Ana Játiva Miguel Muñoz Jonathan Ortiz	Alta	Funcional	Revisión	Funcionarios de soporte Usuarios (cliente) Personal administrativo
RF002	El software deberá tener la capacidad de generar tareas programadas, las mismas que se podrá definir cada cuanto tiempo esta tarea generara un ticket (diario, semanal, mensual) y bajo qué parámetros se lo hará.	Ana Játiva Roberto Huacanes	Media	Funcional	Revisión	Funcionarios de soporte Personal administrativo
RF003	La aplicación tendrá dos	Ana Játiva	Alta	Funcional	Revisión	Personal Administrativo

### A.03 Matriz de involucrados

Tabla 10.

*Matriz de involucrados*

Actores / involucrados	Interés sobre el problema central	Problemas percibidos	Recursos, mandatos y capacidades	Interés del proyecto	Conflictos potenciales
<b>Gerente General</b>	Sería necesario dar mayor disponibilidad de reportes	Datos inadecuados para la toma de decisiones.	Análisis de reportes.	Contar con reportes actualizados.	Falta de información del proceso en general.
<b>Gerente Infraestructura</b>	Sería necesario que el sistema genere reportes de activos físicos, lógicos y tickets relacionados.	Tedioso seguimiento de los tickets de infraestructura .	Generación de reportes referente a tickets de activos.	Reportes actualizados.	Desinterés por parte del personal de soporte.
<b>Gerente Gestión Funcional</b>	Sería necesario que el sistema genere reportes de tickets relacionado a proyectos.	Tediosa e ineficiente seguimiento de los tickets de gestión funcional.	Generación de reportes referente a proyectos.	Reportes actualizados.	Desorganización con el departamento de infraestructura
<b>Gerente Desarrollo</b>	Sería necesario que el sistema genere reportes de tickets relacionados a desarrollo.	Falta* de datos el para el seguimiento de los tickets de desarrollo.	Generación de reportes del área de desarrollo.	Reportes actualizados.	Incumplimiento de los tiempos establecidos por falta de comunicación.
<b>Director de Área</b>	Sería necesario poder verificar los tickets pendientes para realizar seguimiento.	Mala administración del proceso de mesa de ayuda.	Administración y organización de mesa de ayuda	Seguimiento y administración de cada ticket.	Desconocimiento de procesos atrasados, no comunicados por soporte.
<b>Técnico de Soporte</b>	Sería necesario poder verificar detalladamente cada uno de los tickets asignados.	Falta de información para atender una incidencia, ocasionando retrasos.	Brindar soporte técnico al usuario de forma rápida.	Análisis y administración de sus tickets	Desconocimiento de requerimientos pendientes del cliente.
<b>Direccionalador</b>	Se deberá poder verificar todos los tickets pendientes de direccionar y los devueltos.	Demora al asignar un ticket, lo cual dilata el tiempo de espera del usuario final.	Analizar y asignar los tickets al personal adecuado.	Asignación y seguimiento de tickets.	Desconocimiento de los requerimientos de los clientes.
<b>Analista</b>	Sería necesario poder verificar detalladamente cada uno de los tickets asignados.	Falta de información para atender un requerimiento,	Analiza y valora la posible realización de un proyecto.	Análisis y administración de sus tickets	Desconocimiento de tickets asignados a su nombre.

		ocasionando retrasos.			
<b>Desarrollador</b>	Sería necesario poder verificar detalladamente cada uno de los tickets asignados.	Falta de información para atender un requerimiento.	Desarrolla e implementa nuevas mejoras del software.	Análisis y administración de sus tickets	Retrasos en el proceso de desarrollo por falta de información
<b>Usuario (cliente)</b>	Se deberá poder verificar todos los tickets generados	Retrasos importantes en el proceso diario de trabajo, por incidencias.	Generar ticket el momento que surja una incidencia.	Detalle y consulta de tickets.	Descontento y reclamos por atención deficiente.



A.05 Diagrama de secuencia. Creación de ticket.

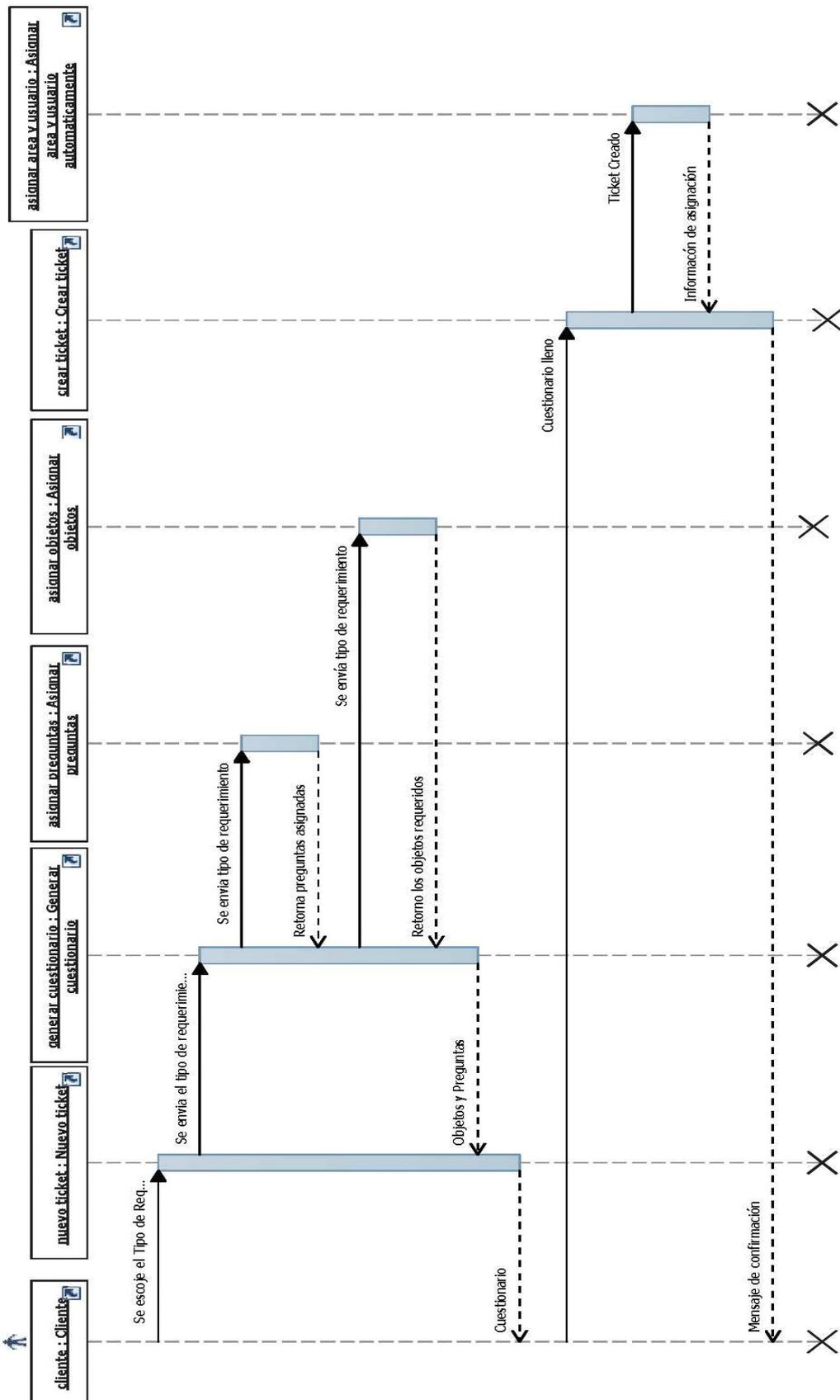


Figura 9. Diagrama de secuencia. Creación de Ticket.

**A.06 Diagrama de secuencia. Administración de ticket.**

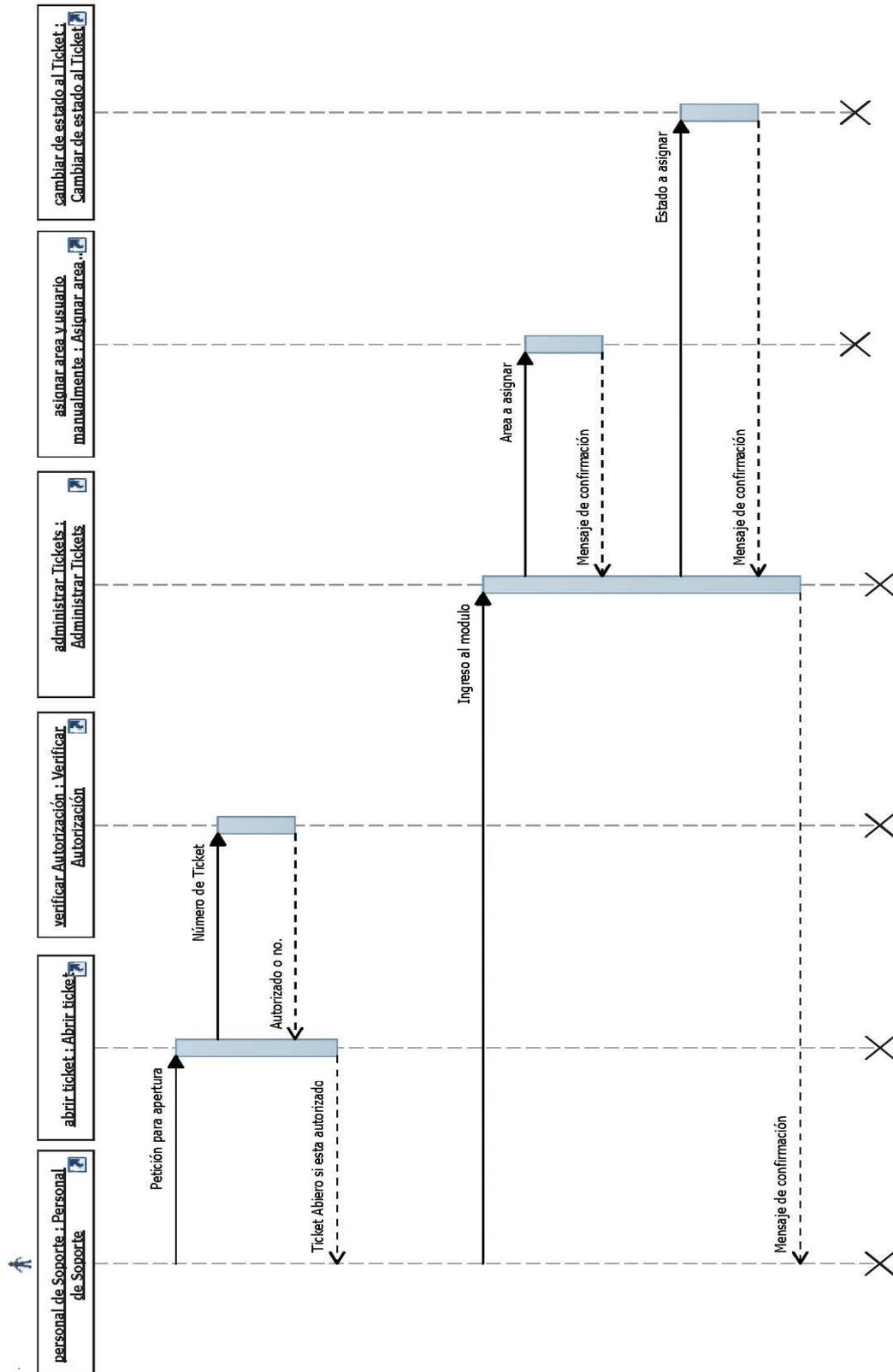


Figura 10 Diagrama de secuencia. Administración de Ticket.

A.07 Diagrama de secuencia. Administración de activo.

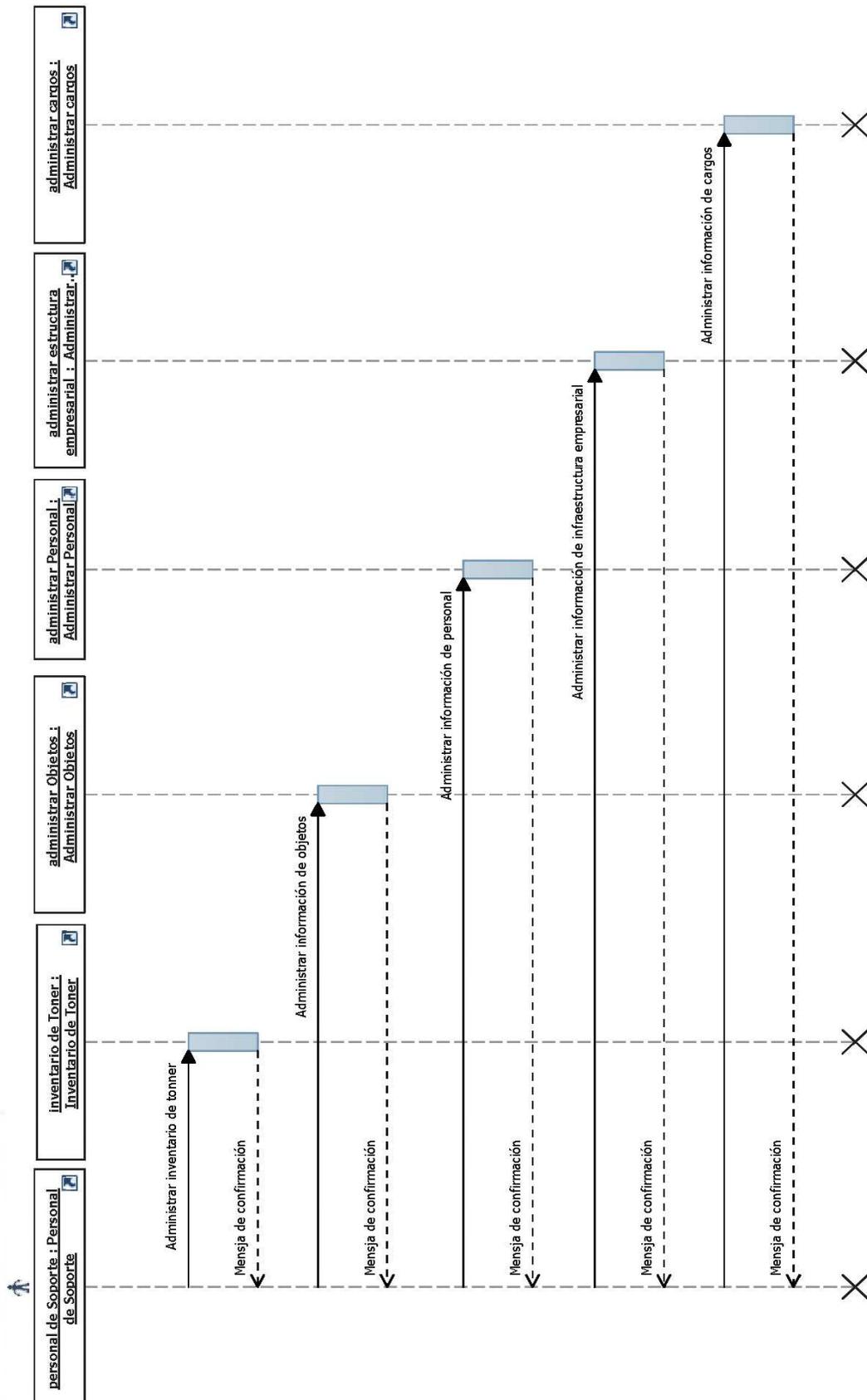


Figura 11 Diagrama de secuencia. Administrar activos.





## A.09 Diagrama lógico.

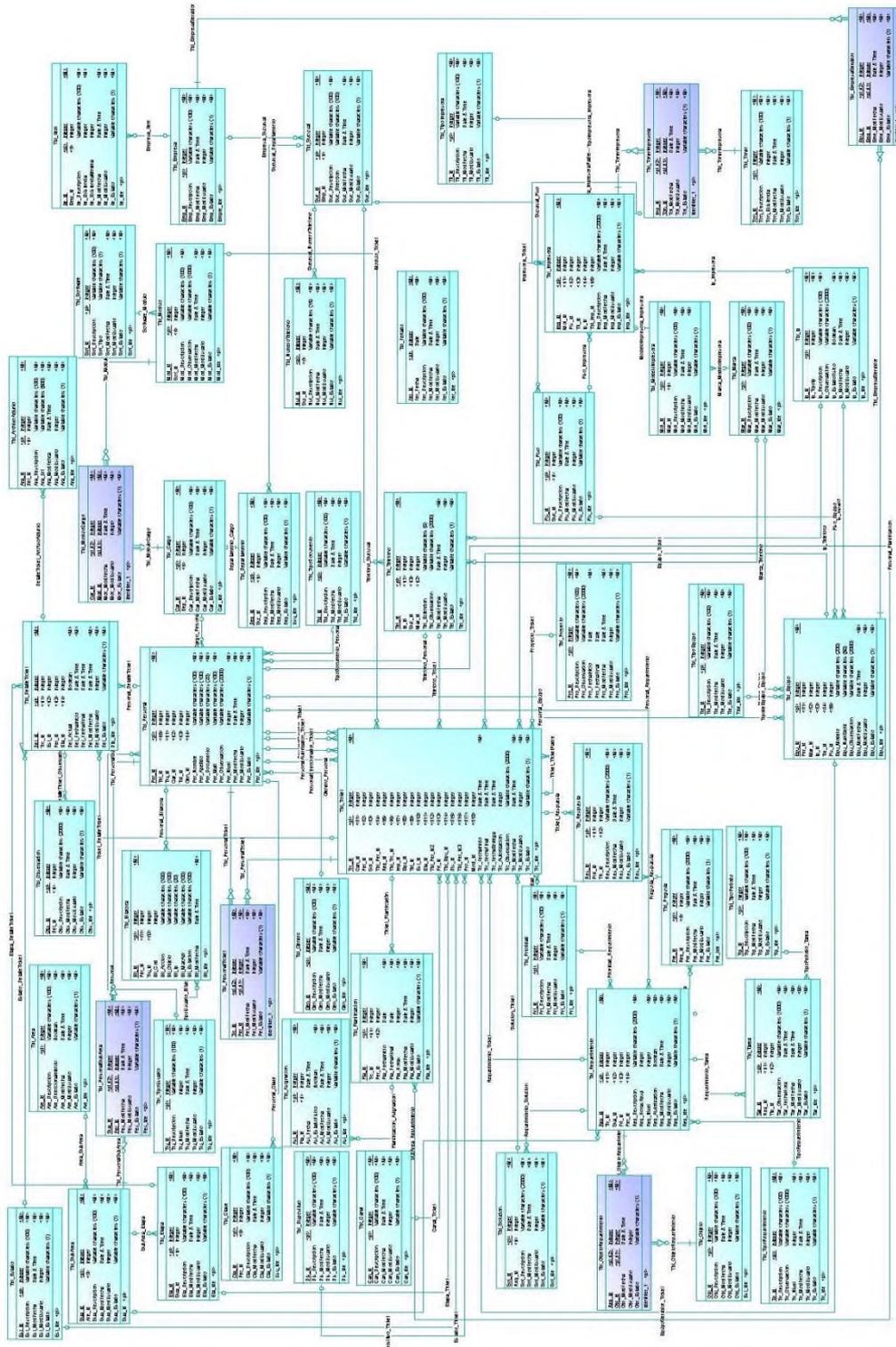


Figura 14. Modelo Lógico



## A.11 Matriz de marco lógico

Tabla 24.

*Matriz de marco lógico.*

Resumen Narrativo	Indicadores	Medios de verificación	Supuestos
<p><b>Fin del proyecto:</b> Aumenta el uso de sistemas de help desk.</p>	<p>Aumente considerablemente el número de empresas que maneja este tipo de sistemas para automatizar su atención de los requerimientos.</p>	<p>Resultados de estadísticas de los diferentes clientes de sistemas de hep desk.</p>	
<p><b>Propósito del proyecto:</b> Se mantiene un servicio de soporte rápido y confiable.</p>	<p>Las quejas por el servicio se eliminarán casi por completo.</p>	<p>Reporte detallado de todos los tickets y las novedades que se han presentado, logrando definir el nivel de satisfacción del cliente.</p>	<p>- Los medios de comunicación se mantienen activos y funcionales.</p>
<p><b>Componentes del proyecto:</b> 1.- Programa informativo para los clientes de CityTech. 2. Programa de control de tickets. 3. Presentación de datos estadísticos.</p>	<p>Aumente exponencialmente la cantidad de clientes que usa el sistema al conocer más a fondo su funcionamiento y confiabilidad. En un año el 75% de tickets deberán ser creados mediante la aplicación web.</p>	<p>Cuadro estadístico que muestra la media de tickets creados vía web o a su vez por otros medios de comunicación habilitados.</p>	<p>- Los clientes aprueban el uso del nuevo sistema para el seguimiento de sus requerimientos. - El personal administrativo da las facilidades para obtener los datos necesarios para personalizar el software a las necesidades de todos los empleados o clientes de la empresa.</p>
<p><b>Actividades del proyecto:</b> 1.1 Capacitar personal de soporte. 1.2 Mejorar la herramienta de trabajo. 1.3 Mejorar el ambiente social laboral. 2.1 Registrar todos los tickets.</p>	<p>El tiempo de respuesta de la persona de soporte disminuye considerablemente, lo que generara en el cliente la sensación de urgencia.</p>		<p>- El departamento de recursos humanos entregara las facilidades para coordinar cursos de capacitación del nuevo sistema de help desk.</p>

2.2 Mejorar la administración de requerimientos.

2.3 Optimizar las vías de comunicación entre el cliente y el personal de soporte.

3.1 Implementar reportes estadísticos.

3.2 Facilitar herramientas para la toma de decisiones.

## A.12 Cronograma

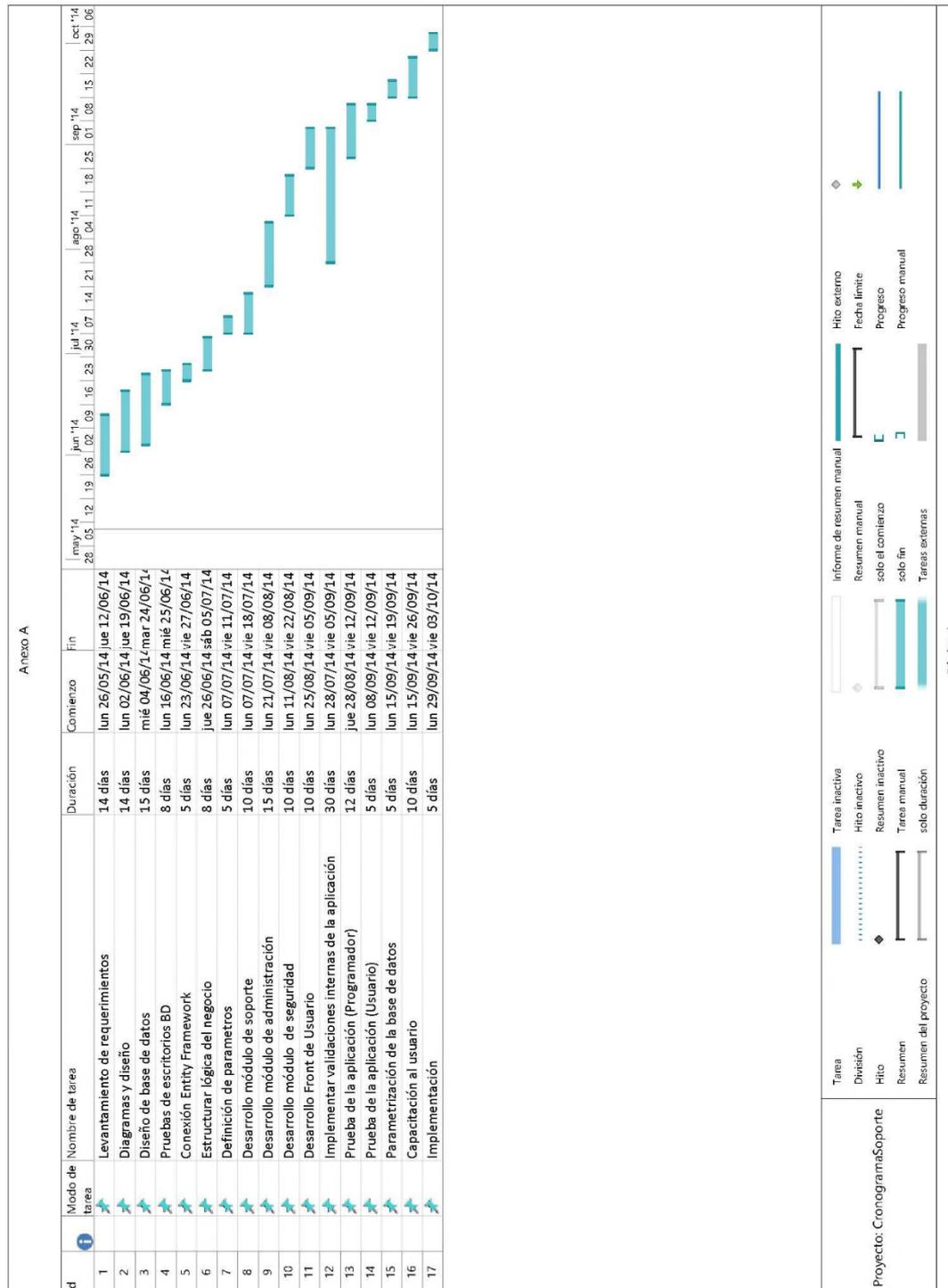


Figura 80. Cronograma de actividades para proyecto.



## MANUAL DE INSTALACIÓN

### SISTEMA DE CONTROL DE SOPORTE TÉCNICO

## **A. 13 Manual de instalación**

### **Introducción**

El diseño y desarrollo de la siguiente aplicación está orientado a proporcionar un servicio de mesa de ayuda, mediante la creación de tickets con diferentes propiedades para un correcto control y seguimiento de los mismos.

El principal objetivo del software es facilitar el manejo y análisis de requerimientos técnicos de cualquier tipo, manteniendo un canal de comunicación entre el personal de soporte y el cliente en la empresa Citytech.

El presente manual pretende servir de base para el entendimiento y la correcta instalación y acondicionamiento de la aplicación.

## ÍNDICE DE CONTENIDO

Manual de instalación.....	119
1. Requerimientos antes de la instalación.....	119
2. Instalación.....	119

## **1. Requerimientos antes de la instalación.**

Para que la aplicación funcione correctamente en nuestro equipo debemos tener instalado cualquier sistema operativo que permita instalar un navegador web de última generación, como google Chrome, IE 9 en adelante, Firefox.

Lo primero que debemos hacer es verificar que nuestro navegador esté funcionando de forma correcta. Si no tenemos instalado ninguno de estos navegadores debemos realizar la instalación de cualquiera de ellos para su correcto funcionamiento.

## **2. Instalación.**

El software está orientado a la web, por lo tanto para tener acceso al mismo únicamente debemos ingresar a la siguiente dirección: “[www.aseguradoradelsur.com.ec/sotec/](http://www.aseguradoradelsur.com.ec/sotec/)”, si nos muestra algún tipo de error por favor comunicarse con el administrador del sistema.

Cuando ya tengamos el acceso a la página debemos verificar que contamos con un usuario para hacer uso del sistema, en el caso de no tener creado el usuario o este deshabilitado debemos comunicarnos con el administrador del sistema para pedirle la creación del usuario correspondiente.

Adicional debemos verificar que el navegador nos permita hacer uso de cookies, esto para evitar pérdida de información.



## MANUAL DE USUARIO DE LA APLICACIÓN INFORMÁTICA

### SISTEMA DE CONTROL DE SOPORTE TÉCNICO

## **A.14 Manual de usuario de la aplicación informática**

### **INTRODUCCIÓN**

El diseño y desarrollo de la siguiente aplicación está orientado a proporcionar un servicio de mesa de ayuda, mediante la creación de tickets con diferentes propiedades para un correcto control y seguimiento de los mismos.

El principal objetivo del software es facilitar el manejo y análisis de requerimientos técnicos de cualquier tipo, manteniendo un canal de comunicación entre el personal de soporte y el cliente en la empresa Citytech.

El presente manual pretende servir de base para el entendimiento y la correcta utilización de la aplicación en cada una de las sesiones que se propusieron en el levantamiento de la información.

## ÍNDICE DE CONTENIDO

Manual de usuario de la aplicación informática .....	125
1. Implementación del sistema.....	125
1.01. Requerimientos del hardware.....	125
1.02. Requerimientos del software.....	125
2. Funcionalidad.....	125
2.01. Tipos de usuario. ....	125
2.02. Ingreso al sistema. ....	126
2.02.01. Ingreso al sistema como cliente.....	127
2.02.02. Ingreso al sistema como direccionador. ....	133
2.02.03. Ingreso al sistema como técnico.....	137

## ÍNDICE DE TABLAS

Manual de usuario de la aplicación informática .....	125
1. Matriz de requerimientos de hardware.....	125
2. Matriz de requerimientos de software.....	125



## ÍNDICE DE FIGURAS

Manual de usuario de la aplicación informática .....	125
1. Pantalla de login.....	126
2. Pantalla de login con datos. ....	127
3. Creación de tickets, clientes.....	128
4. Confirmar creación de tickets. ....	128
5. Confirmar creación de ticket por detalle.....	129
6. Creación de ticket erróneo. ....	129
7. Creación de ticket correcto.. ....	130
8. Datos generales de ticket.....	130
9. Detalles de ticket.....	131
10. Administración de ticket, cliente. ....	131
11. Cambio de clave.....	132
12. Cerrar sesión.....	132
13. Creación de ticket, direccionador.....	133
14. Filtrado de ticket, direccionador. ....	134
15. Verificar ticket sin asignar.....	134
16. Administración de ticket, direccionador. ....	135
17. Autorización de ticket.....	136
18. Ticket no administrable.....	136
19. Administración de ticket, soporte.. ....	137
20. Creación de ticket, soporte.....	138
21. Mantenimiento de usuarios.. ....	138
22. Administración de ticket, soporte. ....	139
23. Cambiar estado de ticket.....	140
24. Administración de empresas. ....	140
25. Administración detalle de ticket.. ....	141

## 1. Implementación del sistema.

### 1.01 Requerimientos del hardware.

Los requerimientos de hardware mínimos son los siguientes:

Tabla 1.

*Matriz de requerimientos de hardware.*

Tipo	Mínimo
Procesador	Intel Pentium 4 y posterior
Espacio libre en disco	350 MB
RAM	512 MB

### 1.02 Requerimientos del software

Los requerimientos de software mínimos son los siguientes:

Tabla 2.

*Matriz de requerimientos de software.*

Requisitos de Windows	Requisitos de Mac	Requisitos de Linux	
Sistemas operativos	Windows XP Service Pack 2 y versiones posteriores Windows Vista Windows 7 Windows 8	Mac OS X 10.6 y versiones posteriores	Ubuntu 12.04 y versiones posteriores Debian 7 y versiones posteriores OpenSuSE 12.2 y versiones posteriores Fedora Linux 17
Navegador	Google Chrome. Internet Explorer 9. Firefox		

## 2. Funcionalidad

### 2.01 Tipos de usuario

El sistema cuenta con 5 tipos de usuario principales, los cuales describiremos a continuación.

- \* Cliente.
- \* Direccionador.
- \* Personal de soporte.
- \* Personal funcional.
- \* Administrador.

## 2.02. Ingreso al sistema

Para ingresar a la aplicación web debemos ingresar a cualquier navegador instalado en nuestro equipo. Vamos a ingresar la siguiente dirección en la barra de búsqueda del navegador. [www.aseguradoradelsur.com.ec/sotec/](http://www.aseguradoradelsur.com.ec/sotec/).

Se mostrara la siguiente pantalla:

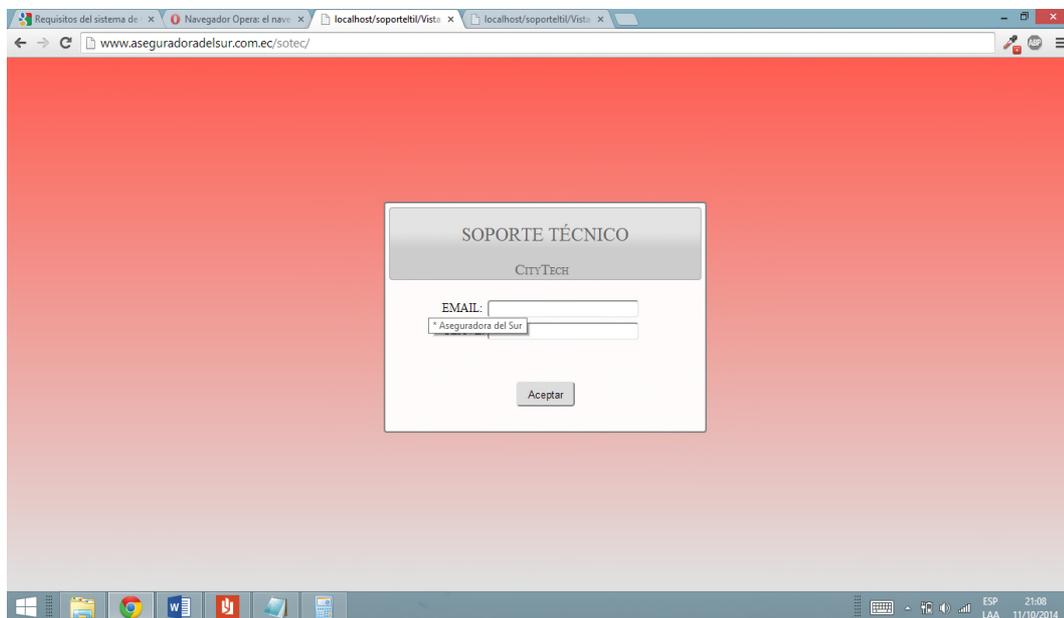


Figura 1. Pantalla de login.

En esta pantalla vemos dos campos para ingresar, en el primero debemos ingresar la dirección de correo electrónico del cliente, y en el siguiente campo la contraseña del mismo.

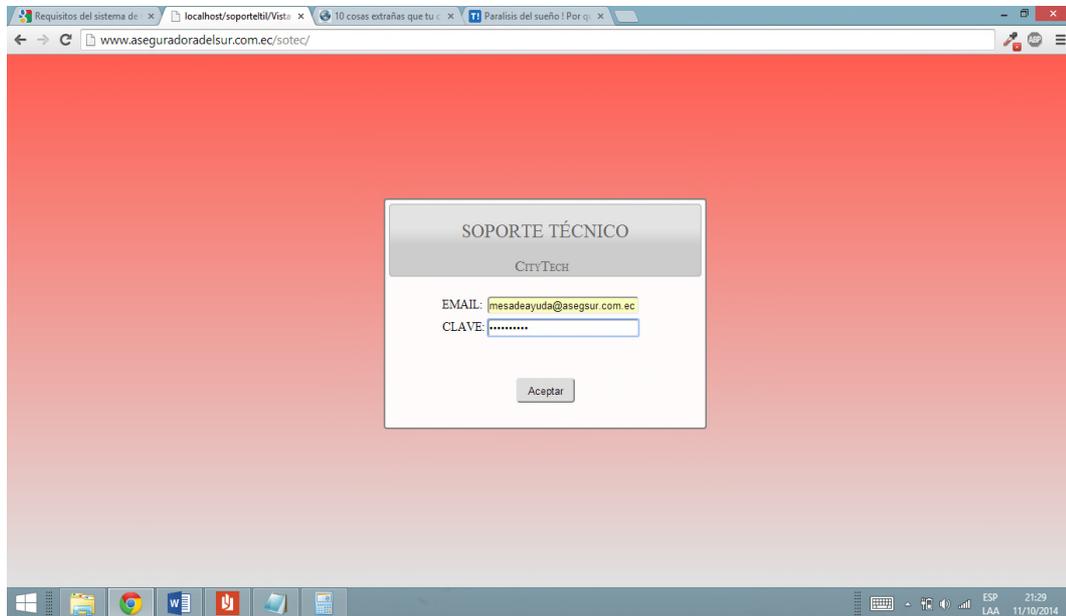


Figura 2. Pantalla de login con datos.

Ingresaremos los datos requeridos e ingresara a la pantalla principal dependiendo del rol de usuario que usted tenga.

### 2.02.01. Ingreso al sistema como cliente

Cuando ingresemos a la aplicación con un usuario de tipo cliente por defecto nos ingresara a la pantalla de creación de ticket.

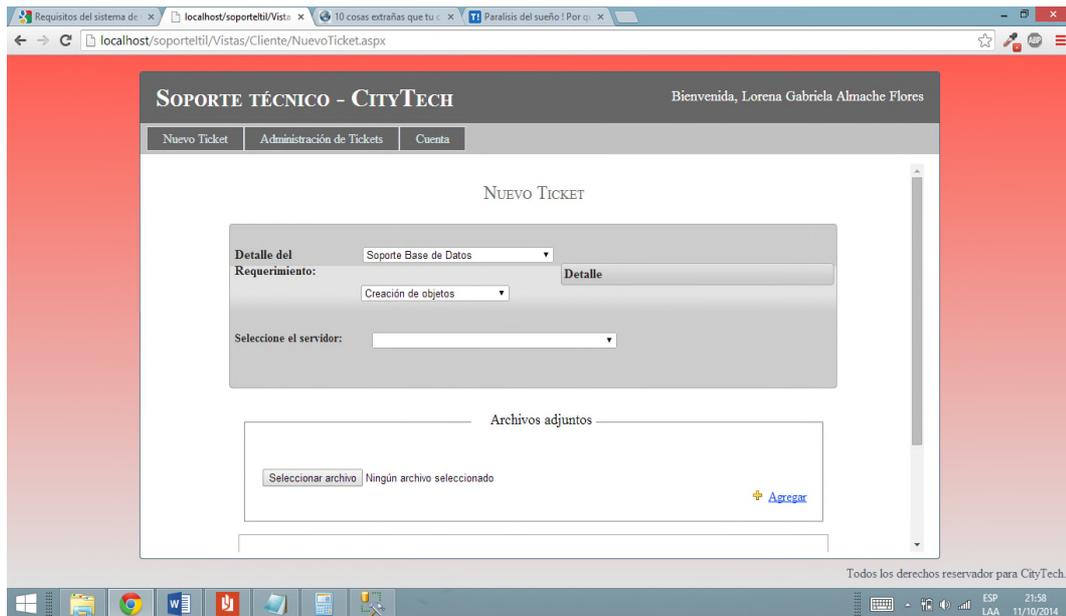


Figura 3. Creación de ticket cliente.

Para poder crear un ticket debemos escoger el detalle del ticket y si deseamos también debemos escoger los objetos que se despliegan después del detalle.

Además tendremos la opción de adjuntar algún archivo y también poner alguna observación del ticket, cuando estemos seguros de la información ingresada daremos click en el botón “enviar ticket”.

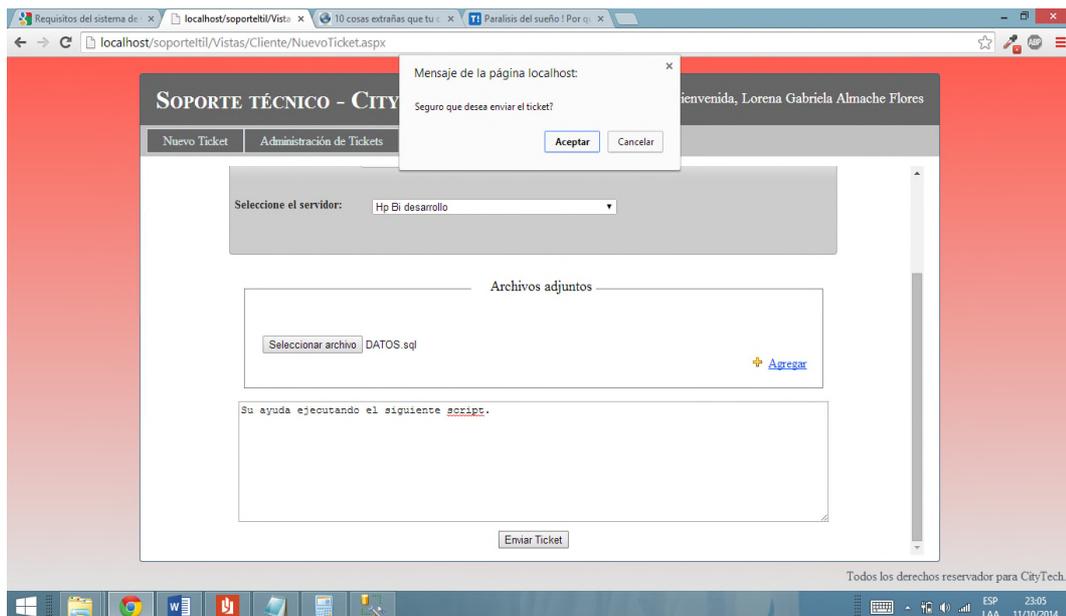


Figura 4. Confirmar creación de ticket.

Cuando ya existe un ticket abierto o pendiente con el mismo detalle nos saldrá un mensaje indicando esto, si se desea crear el ticket aunque vallamos a duplicar dos tickets con el mismo detalle, únicamente debemos activar la opción que se desplegara después del mensaje y a continuación podremos enviar de nuevo el ticket sin ningún problema.

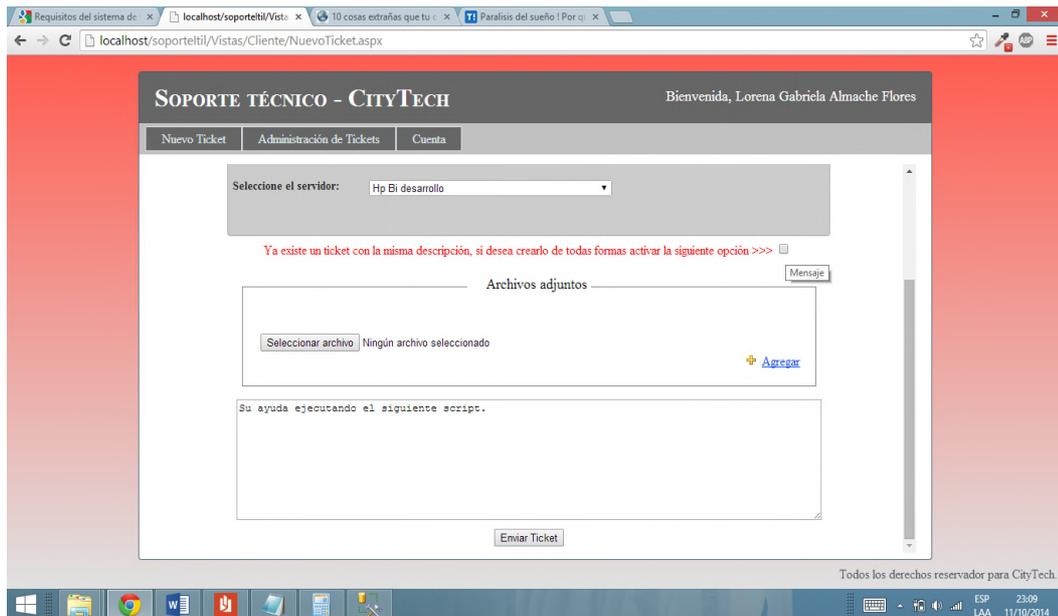


Figura 5. Confirmar creación de ticket, por detalle.

En caso de mostrarse el siguiente mensaje llamar al administrador del sistema.

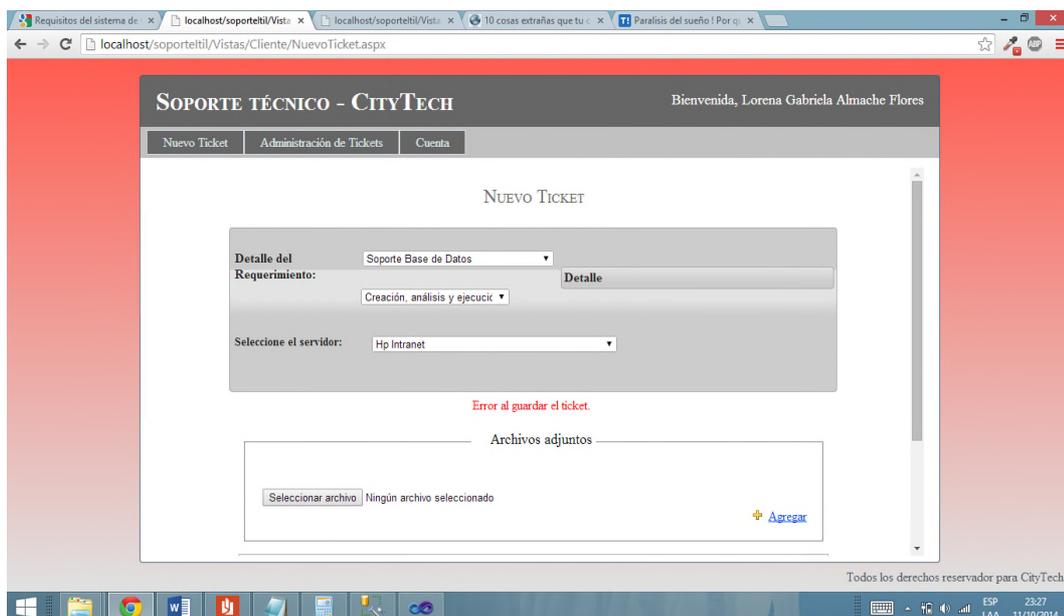


Figura 6. Creación de ticket erróneo.

Si se crea el ticket sin ningún problema se mostrara el siguiente mensaje.

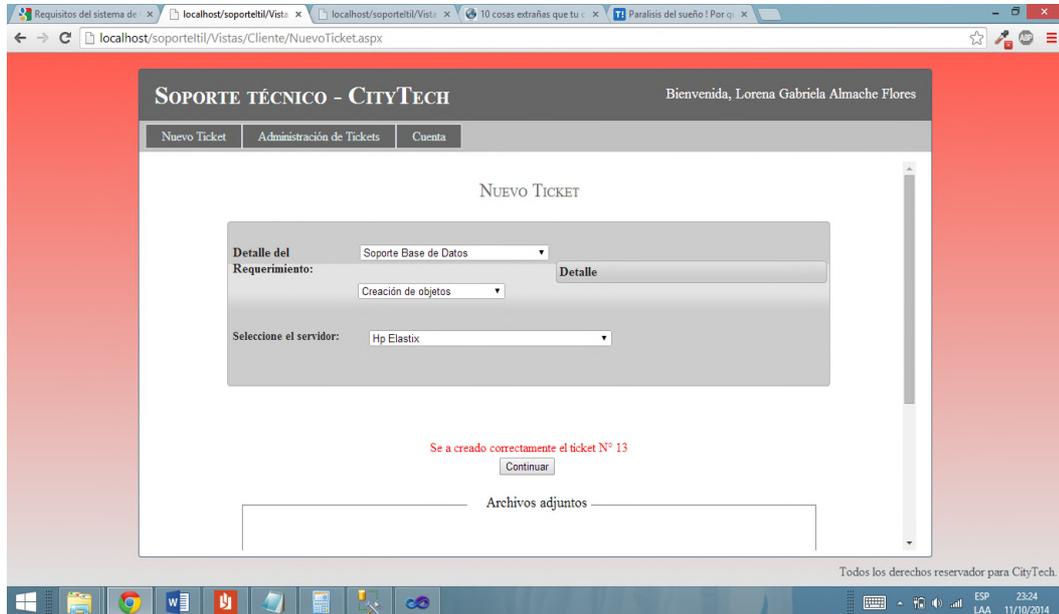


Figura 7. Creación de ticket correcto.

Al dar click en el botón continuar se mostrara el detalle del ticket.

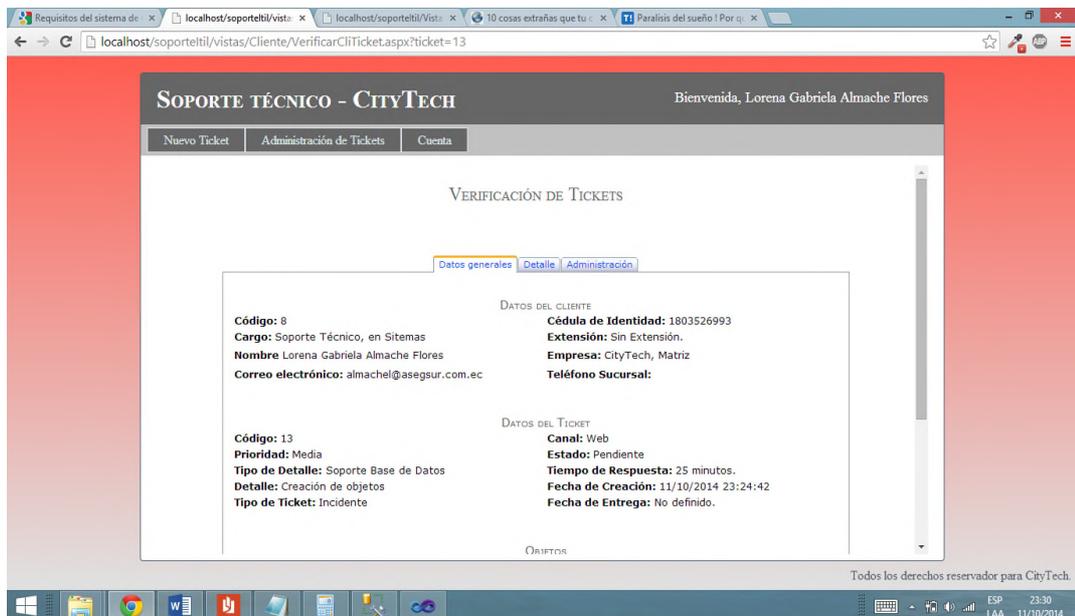


Figura 8. Datos generales del ticket.



Figura 9. Detalles de ticket de ticket.

En esta página podremos verificar todos los detalles de nuestro ticket creado, así como el estado la etapa en la que se encuentra y varios detalles generales del mismo.

Cuando se desee ver todos los tickets que se encuentran a nuestro nombre se debe ingresar en la opción del menú "Administrar ticket", se mostrara la siguiente pantalla:

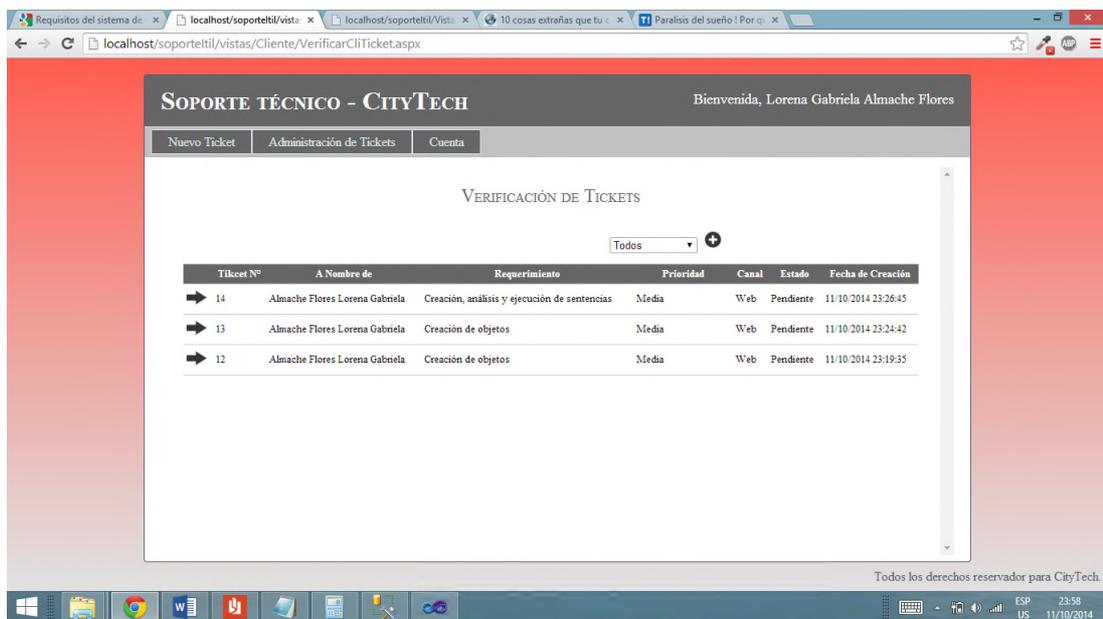


Figura 10. Administración de ticket, cliente.

Para el cambio de clave se debe ingresar a la opción “Cuenta”, “Cambiar Clave”, para luego llenar las opciones requeridas para realizar el cambio de clave.

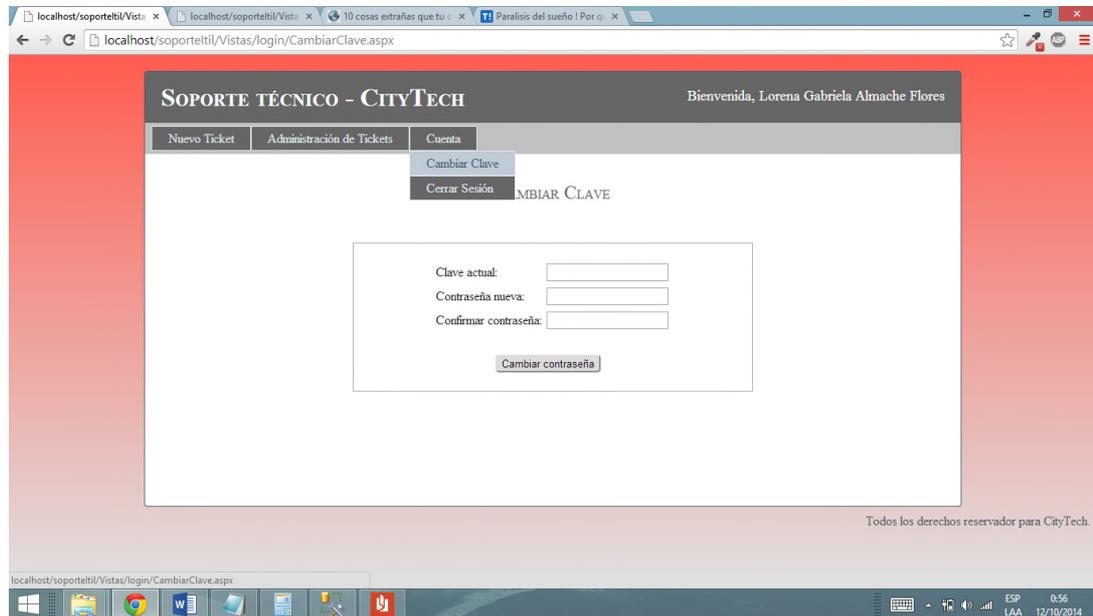


Figura 11. Cambio de clave.

Si lo que deseamos es cerrar la sesión, es necesario ingresar a la opción, “Cuenta”, “Cerrar Sesión”.

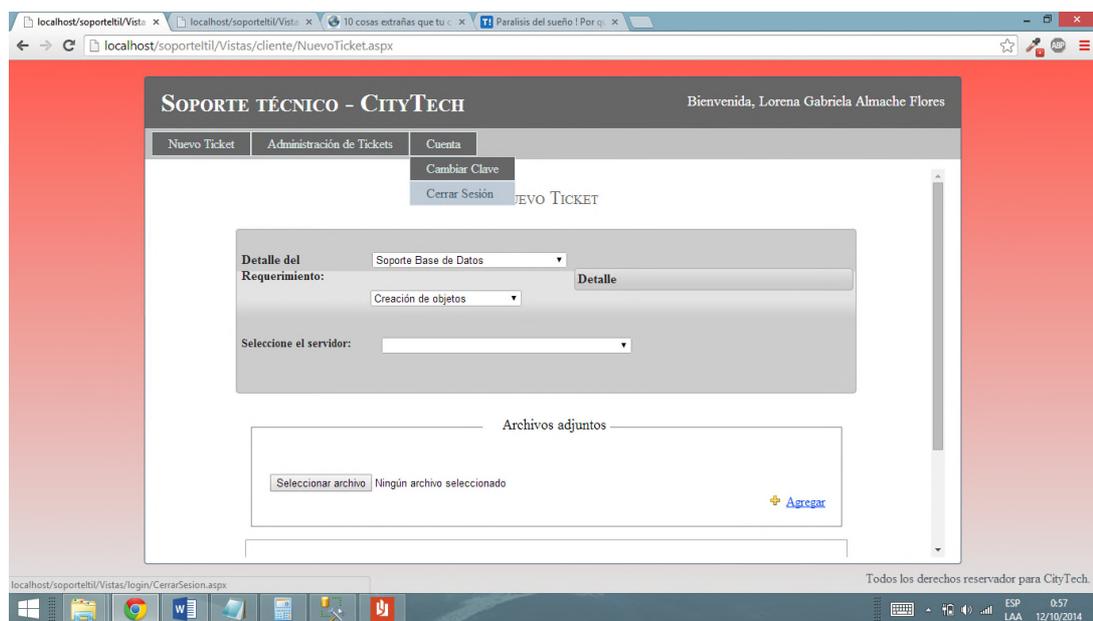


Figura 12. Cerrar Sesión.

## 2.02.02. Ingreso al sistema como direccionador.

Al ingresar a la aplicación como direccionador como defecto nos direccionara a la pantalla para crear un ticket. En esta pantalla tendremos las mismas opciones que el cliente pero adicional podremos escoger el cliente que pide la creación del ticket, el canal por el cual realiza esta solicitud, dos cuadros de selección para indicar si es necesaria una autorización y para saber si se desea direccionar manualmente a alguna área en específico.

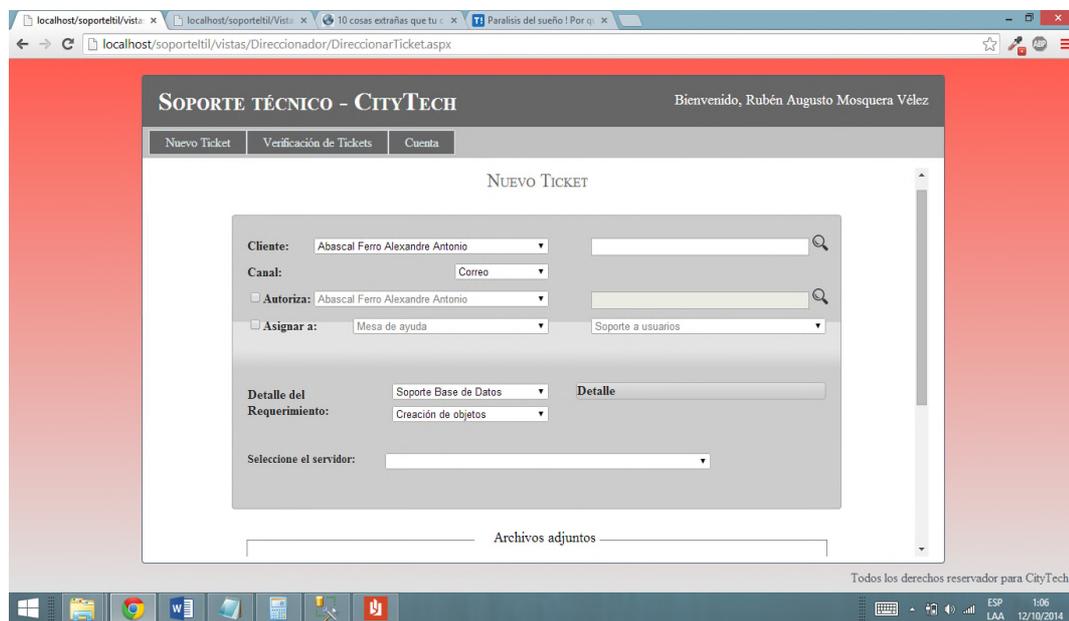


Figura 13. Creación de ticket, direccionador.

La dinámica para crear el ticket es exactamente igual que para el cliente.

Para verificar el estado de todos los tickets, además de saber si existe algún ticket sin direccionar debemos ingresar en la opción de “Verificación de tickets”, en donde encontraremos todos los tickets creados, tendremos la opción de filtrar los tickets por nombre de cliente, numero de ticket, fecha de inicio y fecha final.

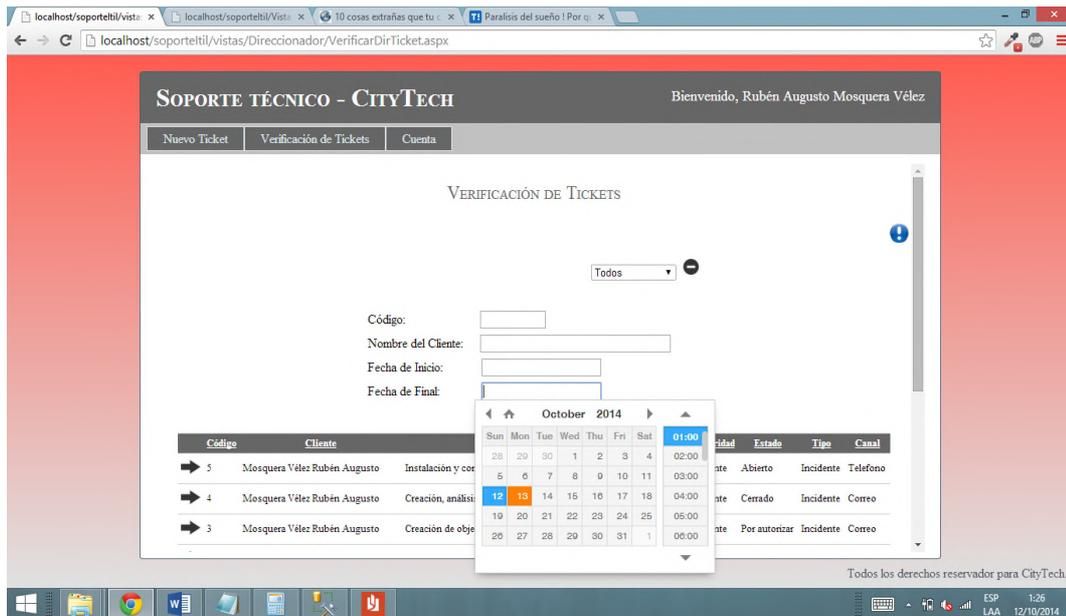


Figura 14. Filtrado de ticket, direccionador.

Como se puede ver en la figura 93, en la parte derecha superior tenemos un aviso de color azul, este aviso aparecerá solo si tenemos algún ticket sin direccionar, esto quiere decir que no se encuentra asignado a ningún técnico de soporte.

Si damos click en este aviso nos mostrara los tickets con este inconveniente de esta manera.

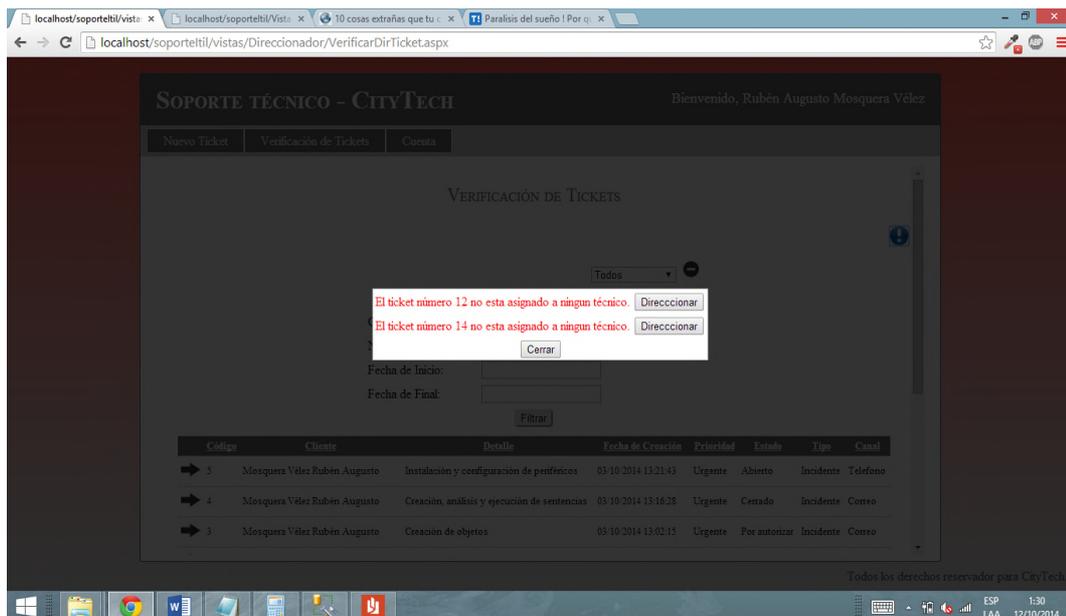


Figura 15. Verificar tickets sin asignación.

En esta pantalla podremos seleccionar el ticket que se desea administrar solamente dado click en el botón que se encuentra a lado de la descripción de dicho ticket, o en su defecto buscar el ticket en la pantalla principal e ingresar dando click en el icono de selección.

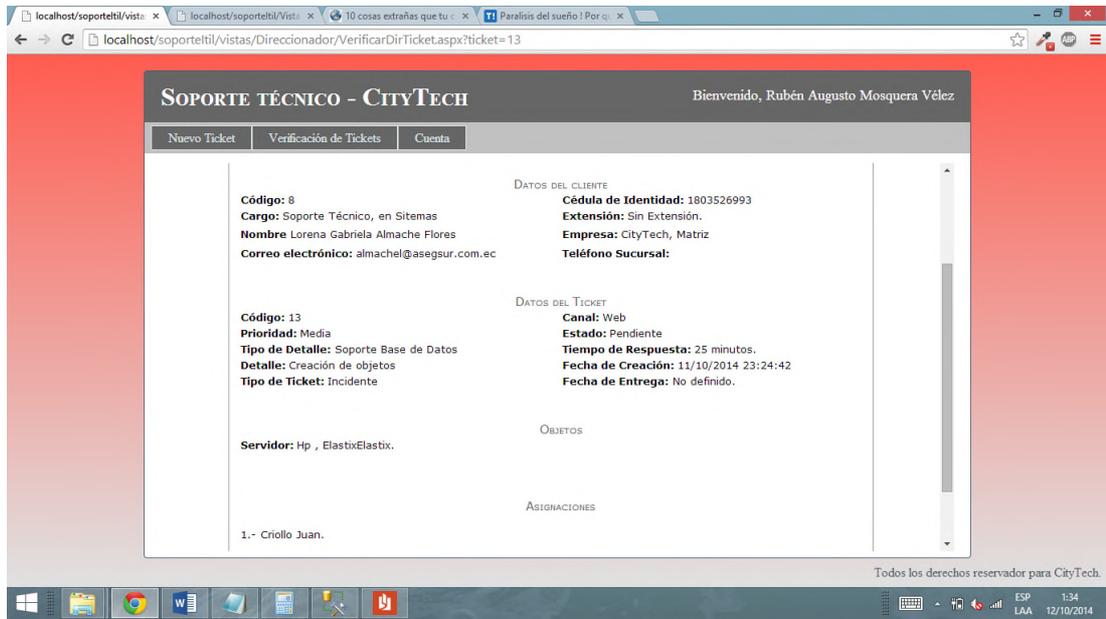


Figura 16. Administración de ticket, direccionador.

Podremos verificar cada uno de los detalles del ticket. Al administrar tendremos tres escenarios.

El primero, cuando el ticket está abierto, en este caso se podrá ver tres opciones, , cuando queremos referenciar a otro ticket, esto nos servirá para definir un ticket padre, esto quiere decir que cuando el ticket padre se cierra entonces los tickets hijos también lo harán, el pedir autorización, lo cual pondrá el ticket en estado de “por autorizar”, y se enviara un correo a la persona que debe autorizar, también podremos asignar a otra área, esto se realiza sin anular ninguna asignación anterior.

En el segundo escenario cuando esta por autorizar, en este estado nos mostrara dos opciones, la de autorizar y la de no autorizar, lo cual nos permitirá controlar el estado de la autorización así como escoger la persona encargada.

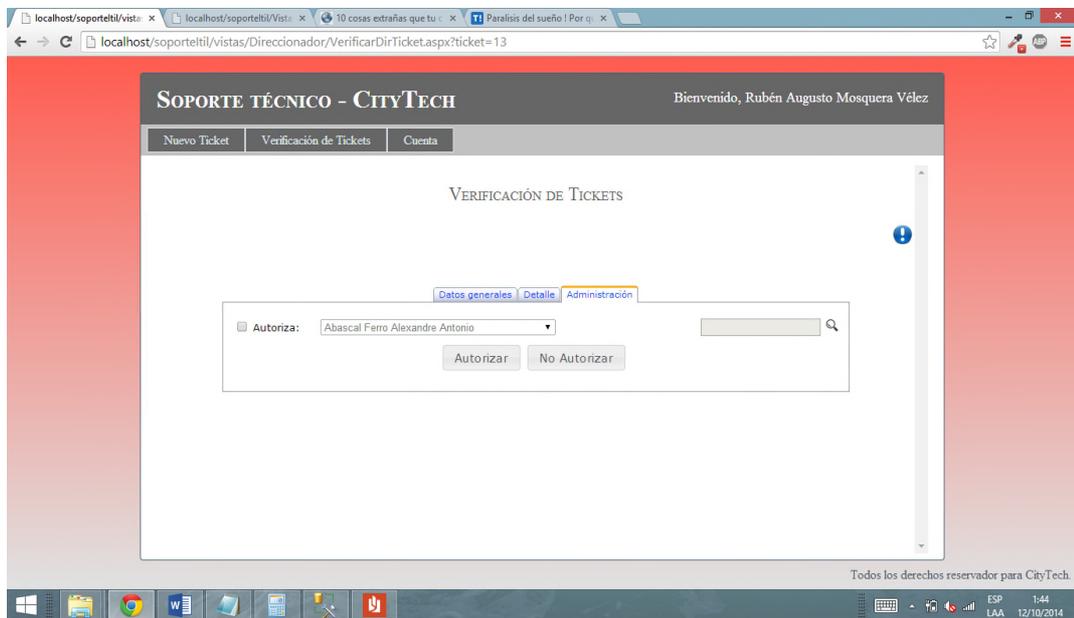


Figura 17. Autorización de ticket.

El tercer escenario se muestra cuando el ticket está cerrado o no autorizado, lo cual no nos permitirá realizar ninguna acción en el ticket.

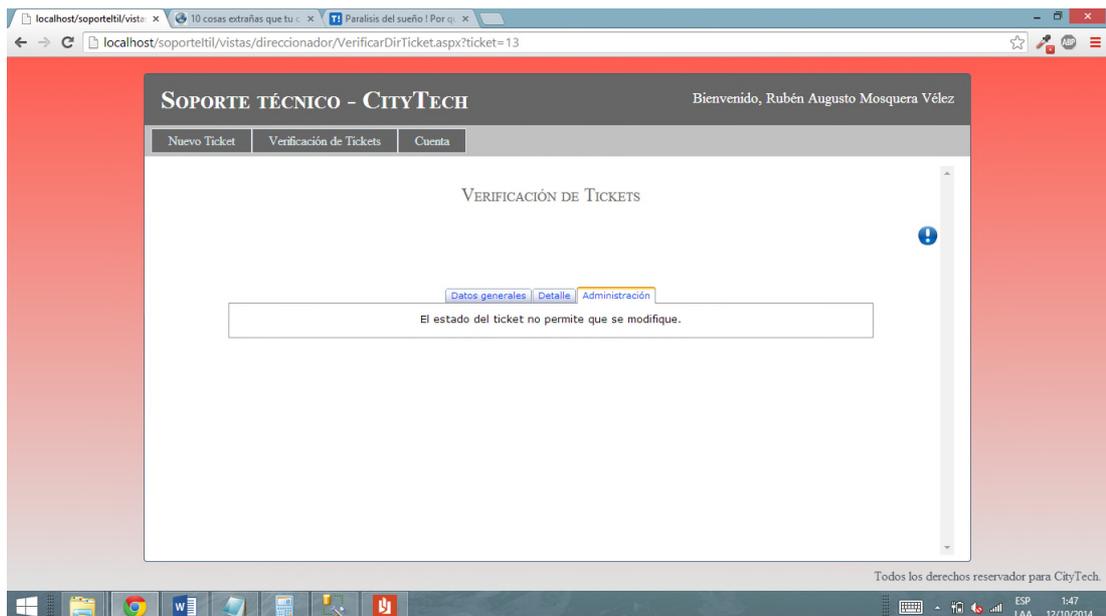
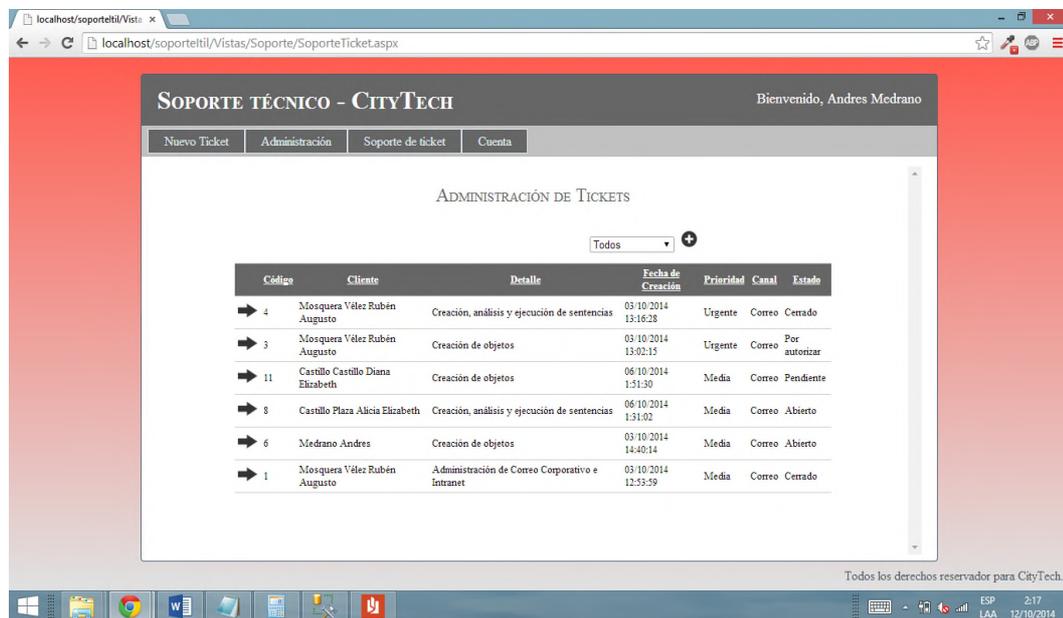


Figura 18. Ticket no administrable.

El resto de opciones son las mismas que para el cliente, se podrá cambiar la contraseña y cerrar la sesión.

### 2.02.03 Ingreso al sistema como técnico.

Cuando ingresemos al sistema como técnico de soporte nos aparecerá la pantalla de administración de ticket. En la misma podremos ver si tenemos algún ticket pendiente a nuestro nombre y también tendremos la opción de filtrar los tickets por diferentes parámetros.



The screenshot shows a web browser window with the URL `localhost/soportetil/Vistas/Soporte/SoporteTicket.aspx`. The page title is "SOPORTE TÉCNICO - CITYTECH" and the user is logged in as "Bienvenido, Andres Medrano". The navigation menu includes "Nuevo Ticket", "Administración", "Soporte de ticket", and "Cuenta". The main content area is titled "ADMINISTRACIÓN DE TICKETS" and features a dropdown menu set to "Todos". Below this is a table of tickets:

Código	Cliente	Detalle	Fecha de Creación	Prioridad	Canal	Estado
4	Mosquera Vélez Rubén Augusto	Creación, análisis y ejecución de sentencias	03/10/2014 13:16:28	Urgente	Correo	Cerrado
3	Mosquera Vélez Rubén Augusto	Creación de objetos	03/10/2014 13:02:15	Urgente	Correo	Por autorizar
11	Castillo Castillo Diana Elizabeth	Creación de objetos	06/10/2014 1:51:30	Media	Correo	Pendiente
8	Castillo Plaza Alicia Elizabeth	Creación, análisis y ejecución de sentencias	06/10/2014 1:31:02	Media	Correo	Abierto
6	Medrano Andres	Creación de objetos	03/10/2014 14:40:14	Media	Correo	Abierto
1	Mosquera Vélez Rubén Augusto	Administración de Correo Corporativo e Intranet	03/10/2014 12:53:59	Media	Correo	Cerrado

At the bottom right of the page, it says "Todos los derechos reservados para CityTech." The Windows taskbar at the bottom shows the date as 12/10/2014 and the time as 2:17.

Figura 19. Administrar tickets, soporte.

Podremos crear tickets de la misma forma que lo haría un direccionador, con la diferencia que tendremos la opción de auto asignar el ticket creado, así podremos controlar los tickets que ya realizamos el soporte pero debemos registrar.

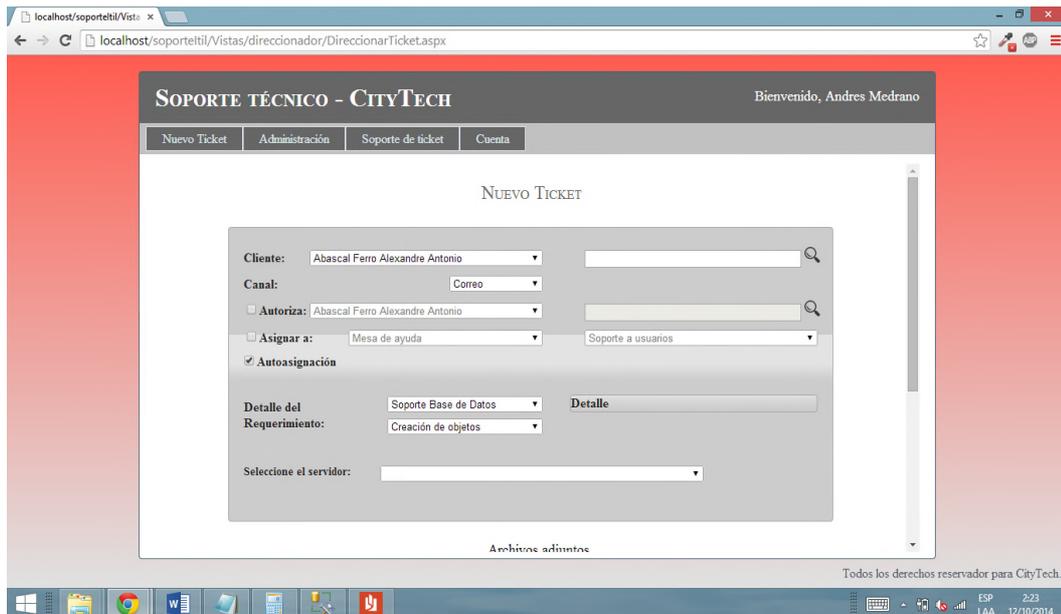


Figura 20. Creación de tickets, soporte.

Además tendremos la opción de crear y modificar cliente. Para ingresar a esta opción debemos ingresar a “Administración”, “Clientes”. Cuando ingresamos como técnico de soporte solo tendremos permiso para crear usuarios de tipo cliente, cuando deseamos crear usuarios con otro perfil debemos crear el usuario como administrador del sistema

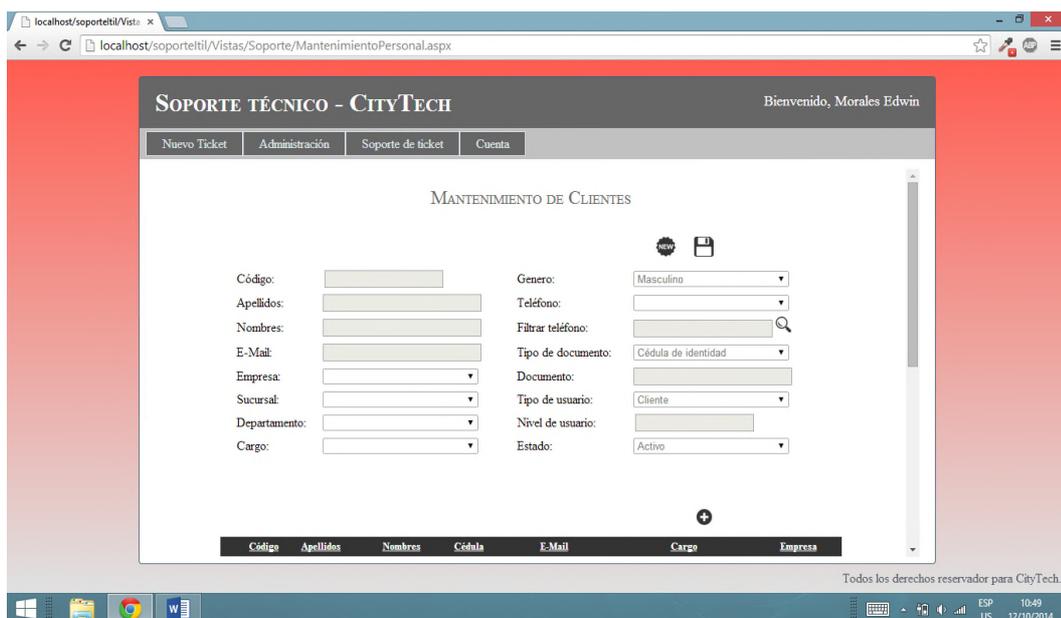


Figura 21. Mantenimiento usuarios.

El personal técnico podrá administrar tickets, tendrá las mismas opciones que el direccionador pero además podrá cambiar ciertos parámetros del ticket, por ejemplo el cliente final, la fecha de entrega, la etapa, prioridad, el detalle del requerimiento, además de poder asignar nuevos objetos al ticket.

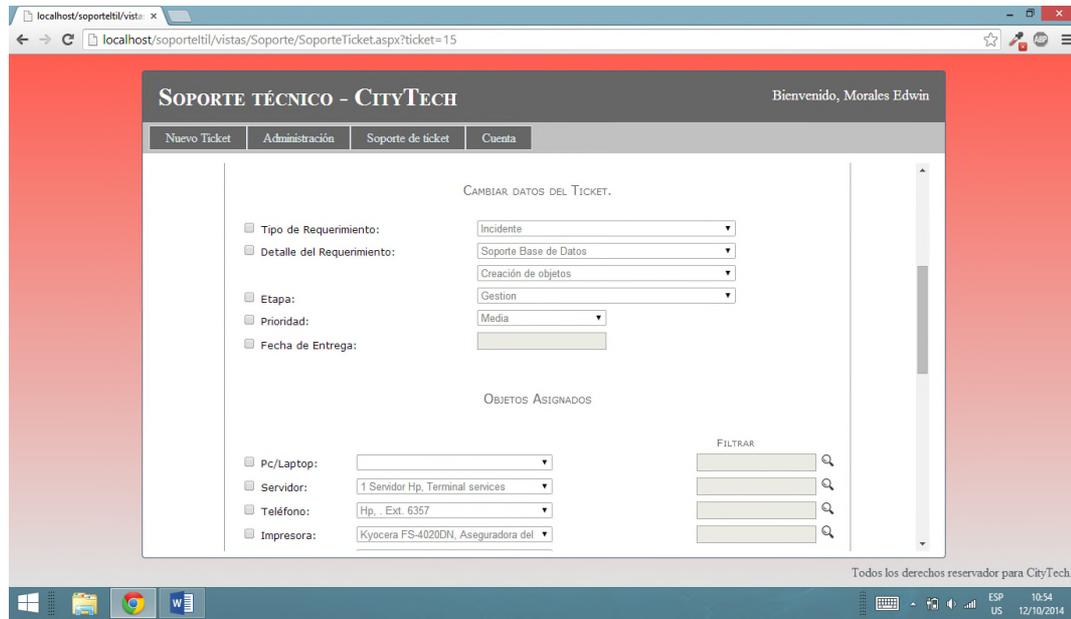


Figura 22. Administración de ticket, soporte.

Tomar en cuenta que se podrá cambiar el estado del ticket, abierto, suspendido, devuelto, cerrado.

\* Abierto.- Quiere decir que vamos a continuar trabajando con el ticket a pesar de que cerramos la sesión del sistema.

\* Suspendido.- Quiere decir que vamos a dejar de trabajar en el ticket por el momento. Tomar en cuenta que el momento que ingresemos nuevamente al ticket este se va a volver a abrir.

\* Devuelto.- Se desactivara la asignación y se mandara al personal adecuado para ser atendido.

\* Cerrado.- El ticket ha sido solventado, en ese momento seleccionamos cerrado para cerrar el mismo.

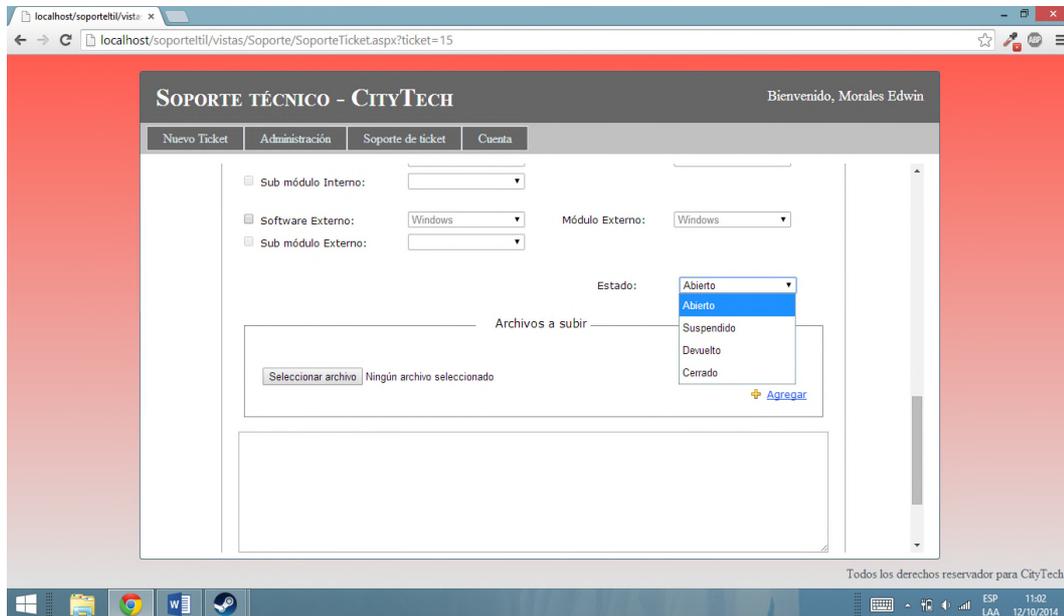


Figura 23. Cambiar estado de ticket.

Para administrar los parámetros, como los detalles se tendrá una pantalla en la que se mostrara una lista con los registros disponibles, a un lado a en la parte inferior se podrá ver todas las opciones disponibles del registro.

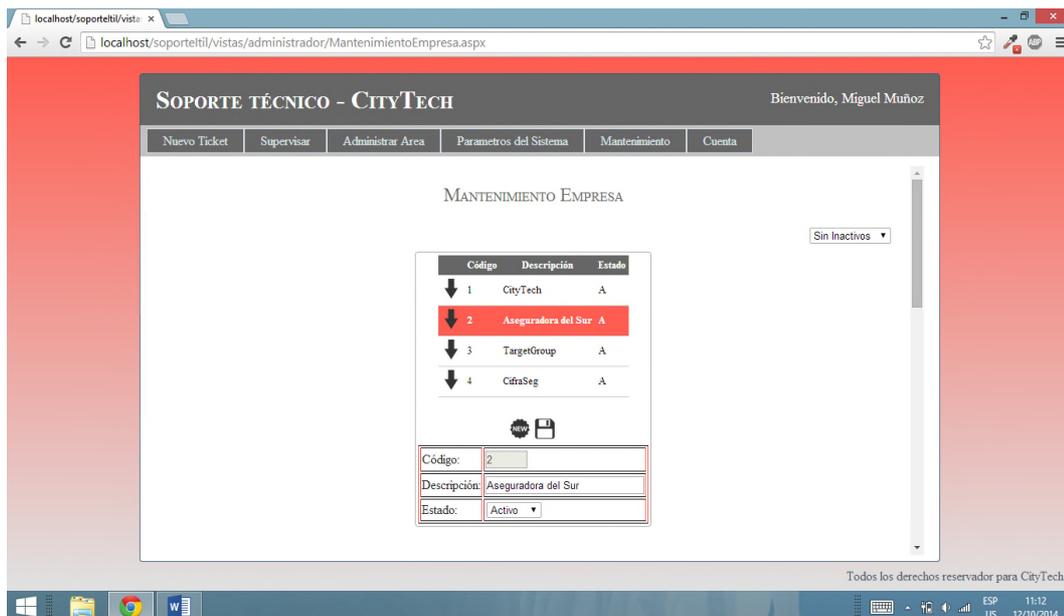


Figura 24. Administración de empresa.

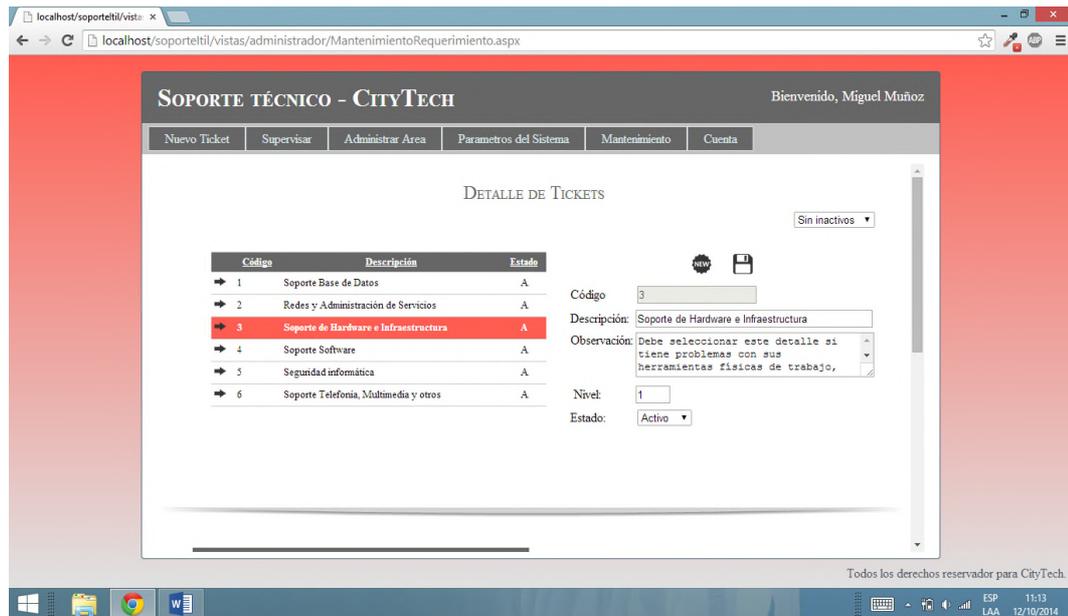


Figura 25. Administración de detalle de ticket.



## MANUAL TÉCNICO DE LA APLICACIÓN INFORMÁTICA

### SISTEMA DE CONTROL DE SOPORTE TÉCNICO

## **A.15 Manual técnico de la aplicación informática**

### **Introducción**

El diseño y desarrollo de la siguiente aplicación está orientado a proporcionar un servicio de mesa de ayuda, mediante la creación de tickets con diferentes propiedades para un correcto control y seguimiento de los mismos.

El principal objetivo del software es facilitar el manejo y análisis de requerimientos técnicos de cualquier tipo, manteniendo un canal de comunicación entre el personal de soporte y el cliente en la empresa Citytech.

El presente manual pretende servir de base para el entendimiento y la correcta utilización de la aplicación en cada una de las sesiones que se propusieron en el levantamiento de la información.

## ÍNDICE

1. Objetivos técnicos.....	146
2. Diccionario de datos. ....	146
2.01. Tabla archivos adjuntos. ....	146
2.02. Tabla área. ....	146
2.03. Tabla bitácora. ....	146
2.04. Tabla canal. ....	146
2.05. Tabla cargo.....	147
2.06. Tabla clave. ....	147
2.07. Tabla departamento.....	147
2.08. Tabla detalle de ticket. ....	147
2.09. Tabla dispositivo. ....	147
2.10. Tabla empresa. ....	147
2.11. Tabla empresa - servidor. ....	147
2.12. Tabla equipo.....	148
2.13. Tabla estado. ....	148
2.14. Tabla etapa. ....	148
2.15. Tabla genero.....	148
2.16. Tabla impresora. ....	148
2.17. Tabla marca. ....	148
2.18. Tabla modelo de impresora.....	148
2.19. Tabla módulo. ....	149
2.20. Tabla número de teléfono.....	149
2.21. Tabla objeto.....	149
2.22. Tabla objeto - requerimiento.....	149
2.23. Tabla observación. ....	149
2.24. Tabla parámetro. ....	149
2.25. Tabla personal.....	149
2.26. Tabla personal - subárea.....	150
2.27. Tabla personal - ticket. ....	150
2.28. Tabla piso. ....	150
2.29. Tabla prioridad.....	150



2.30. Tabla proyecto. ....	150
2.31. Tabla requerimiento. ....	150
2.32. Tabla respuesta. ....	150
2.33. Tabla software. ....	151
2.34. Tabla solución. ....	151
2.35. Tabla subárea. ....	151
2.36. Tabla submódulo. ....	151
2.37. Tabla sucursal. ....	151
2.38. Tabla teléfono. ....	151
2.39. Tabla ticket. ....	151
2.40. Tabla tipo de documento. ....	152
2.41. Tabla tipo de equipo. ....	152
2.42. Tabla tipo de impresora. ....	152
2.43. Tabla tipo de requerimiento. ....	152
2.44. Tabla tipo de ticket. ....	152
2.45. Tabla tipo de usuario. ....	152
3. Base de datos, script. ....	152

## **1. Objetivos técnicos.**

- El acceso a la aplicación web solo sea disponible para las personas que están vinculadas directamente con la empresa, esto es posible mediante la asignación de usuario y contraseña.
- Asignación de rol de usuario a nivel de aplicación.
- Nuestra aplicación tiene la capacidad de soportar el flujo de usuarios activos realizando transacciones simultáneas.

## **2. Diccionario de datos.**

### **2.01. Tabla archivos adjuntos.**

En el modelo la nombraremos como TBL\_ARCHIVOSADJUNTOS, en la cual guardaremos la dirección física del archivo, y a que detalle de ticket está relacionado.

### **2.02. Tabla área.**

En el modelo la nombraremos como TBL\_AREA, en la cual guardaremos las áreas principales de la mesa de ayuda.

### **2.03. Tabla bitácora.**

En el modelo la nombraremos como TBL\_BITACORA, en la cual guardaremos un respaldo de todas las acciones realizadas desde el sistema.

### **2.04. Tabla canal.**

En el modelo la nombraremos como TBL\_CANAL, en la cual guardaremos los canales de comunicación disponibles.

### **2.05. Tabla cargo.**

En el modelo la nombraremos como TBL\_CARGO, en la cual guardaremos los cargos disponibles en las empresas clientes.

### **2.06. Tabla clave.**

En el modelo la nombraremos como TBL\_CLAVE, en esta tabla guardaremos las claves del sistema, esta tabla estará completamente encriptada.

### **2.07. Tabla departamento.**

En el modelo la nombraremos como TBL\_DEPARTAMENTO, en la cual guardaremos los departamentos disponibles en las empresas cliente.

### **2.08. Tabla detalle de ticket.**

En el modelo la nombraremos como TBL\_DETALLETICKET, en la cual guardaremos las acciones que se han realizado a un ticket, es decir creación, suspensión, cierre, etc. Se guardará que persona lo hizo.

### **2.09. Tabla dispositivo.**

En el modelo la nombraremos como TBL\_DISPOSITIVO, en la cual guardaremos los dispositivos disponibles para dar mantenimiento.

### **2.10. Tabla empresa.**

En el modelo la nombraremos como TBL\_EMPRESA, en la cual guardaremos las empresas a las que vamos a dar soporte mediante la mesa de ayuda.

### **2.11. Tabla empresas - servidor.**

En el modelo la nombraremos como TBL\_EMPRESASERVIDOR, en la cual guardaremos las asignaciones de los servidores a las empresas.

### **2.12. Tabla equipo.**

En el modelo la nombraremos como TBL\_EQUIPO, en la cual guardaremos los equipos disponibles, es decir laptops, pcs, servidores.

### **2.13. Tabla estado.**

En el modelo la nombraremos como TBL\_ESTADO, en la cual guardaremos los estados disponibles para asignar a los tickets y sus detalles.

### **2.14. Tabla etapa.**

En el modelo la nombraremos como TBL\_ETAPA, en la cual guardaremos las etapas disponibles, organizadas por area.

### **2.15. Tabla genero**

En el modelo la nombraremos como TBL\_GENERO, en la cual guardaremos los géneros disponibles, por defecto serán: masculino y femenino.

### **2.16. Tabla impresora.**

En el modelo la nombraremos como TBL\_IMPRESORA, en la cual guardaremos las impresoras disponibles para las empresas cliente.

### **2.17. Tabla marca.**

En el modelo la nombraremos como TBL\_MARCA, en la cual guardaremos las marcas disponibles para definir en los activos de las empresas cliente.

### **2.18. Tabla modelo de impresora.**

En el modelo la nombraremos como TBL\_MODELOIMPRESORA, en la cual guardaremos los modelos específicos de las impresoras disponibles, esto para controlar cuando un modelo existe en diferentes departamentos.

### **2.19. Tabla modulo.**

En el modelo la nombraremos como TBL\_MODULO, en la cual guardaremos los módulos de software disponibles para soporte, se relaciona con software.

### **2.20. Tabla número de teléfono.**

En el modelo la nombraremos como TBL\_NUMEROTELEFONO, en la cual guardaremos los números convencionales de las sucursales de las empresas.

### **2.21. Tabla objeto.**

En el modelo la nombraremos como TBL\_OBJETO, en la cual guardaremos los objetos disponibles para dar soporte técnico.

### **2.22. Tabla objeto - requerimiento.**

En el modelo la nombraremos como TBL\_OBJETOREQUERIMIENTO, en la cual definiremos cuando a un requerimiento se le debe pedir un tipo de objeto.

### **2.23. Tabla observación.**

En el modelo la nombraremos como TBL\_OBSERVACION, en la cual guardaremos las observaciones de un ticket, se relaciona con un detalle de ticket en específico.

### **2.24. Tabla parámetro.**

En el modelo la nombraremos como TBL\_PARAMETRO, en la cual guardaremos los parámetros principales del sistema.

### **2.25. Tabla personal.**

En el modelo la nombraremos como TBL\_PERSONAL, en la cual guardaremos los empleados de las empresas clientes, además de nuestros propios empleados, los mismos que relacionaremos con el cargo y tipo de usuario.

### **2.26. Tabla personal - subárea.**

En el modelo la nombraremos como TBL\_PERSONALSUBAREA, en la cual definiremos cuando un empleado pertenece a una subárea en específico, esto para el direccionamiento automático.

### **2.27. Tabla personal - ticket.**

En el modelo la nombraremos como TBL\_PERSONALTICKET, en la cual definiremos cuando a un empleado se le ha asignado un ticket en específico.

### **2.28. Tabla piso.**

En el modelo la nombraremos como TBL\_PISO, en la cual guardaremos los pisos que encontraremos físicamente en las sucursales.

### **2.29. Tabla prioridad.**

En el modelo la nombraremos como TBL\_PRIORIDAD, en la cual guardaremos los tipos de prioridades disponibles para un ticket.

### **2.30. Tabla proyecto .**

En el modelo la nombraremos como TBL\_PROYECTO, en la cual guardaremos todos los proyectos que valla a realizar la empresa.

### **2.31. Tabla requerimiento.**

En el modelo la nombraremos como TBL\_REQUERIMIENTO, en la cual guardaremos el detalle general del requerimiento o incidente.

### **2.32. Tabla respuesta.**

En el modelo la nombraremos como TBL\_RESPUESTA, en la cual guardaremos las respuesta del cliente a una pregunta en específico, esto cuando un objeto no está definido.

### **2.33. Tabla software.**

En el modelo la nombraremos como TBL\_SOFTWARE, en la cual guardaremos las aplicaciones internas y externas que usen las empresas cliente.

### **2.34. Tabla solución.**

En el modelo la nombraremos como TBL\_SOLUCION, en la cual definiremos las soluciones disponibles para un ticket en específico.

### **2.35. Tabla subárea.**

En el modelo la nombraremos como TBL\_SUBAREA, en la cual guardaremos la clasificación o subdivisión de un área en específico.

### **2.36. Tabla submódulo.**

En el modelo la nombraremos como TBL\_SUBMODULO, en la cual guardaremos la clasificación o subdivisión de un módulo de software.

### **2.37. Tabla sucursal.**

En el modelo la nombraremos como TBL\_SUCURSAL, en la cual guardaremos las sucursales disponibles por empresa.

### **2.38. Tabla teléfono.**

En el modelo la nombraremos como TBL\_TELEFONO, en la cual guardaremos los teléfonos internos de las empresas, es decir teléfonos ip.

### **2.39. Tabla ticket.**

En el modelo la nombraremos como TBL\_TICKET, en la cual guardaremos el detalle general del ticket, indicando valores necesarios para el cliente.

### **2.40. Tabla tipo de documento.**

En el modelo la nombraremos como TBL\_TIPODOCUMENTO, en la cual guardaremos el tipo del documento personal, es decir cédula, pasaporte. etc.

#### **2.41. Tabla tipo de equipo.**

En el modelo la nombraremos como TBL\_TIPOEQUIPO, en la cual guardaremos el tipo del equipo, es decir pc, servidor. etc.

#### **2.42. Tabla tipo de impresora.**

En el modelo la nombraremos como TBL\_TIPOIMPRESORA, en la cual guardaremos el tipo impresora, es decir láser, matricial, etc.

#### **2.43. Tabla tipo de requerimiento.**

En el modelo la nombraremos como TBL\_TIPOREQUERIMIENTO, en la cual guardaremos detalle del ticket en forma más generalizada, es decir soporte software, soporte hardware. etc.

#### **2.44. Tabla tipo de ticket.**

En el modelo la nombraremos como TBL\_TIPOTICKET, en la cual guardaremos el tipo del ticket, es decir incidencia, requerimiento. etc.

#### **2.45. Tabla tipo de usuario.**

En el modelo la nombraremos como TBL\_TIPODOUSUARIO, en la cual guardaremos el tipo del usuario con el cual nos vamos a conectar al sistema, es decir cliente, direccionador. etc.

### **3. Base de datos “script”.**

Use Sotec

go

```
create table TBL_ACTIVO (  
    ACT_ID          int          identity,  
    PIS_ID          int          null,
```

```
MDL_ID          int          null,
EMP_ID          int          null,
OBJ_ID          int          null,
PER_ID          int          null,
IP_ID           int          null,
ACT_SERIE       varchar(20)  null,
ACT_MODFECHA    datetime    not null,
ACT_MODUSUARIO  int          not null,
ACT_ESTADO      varchar(1)   not null,

constraint PK_TBL_ACTIVO primary key nonclustered (ACT_ID)
)
go

create index RELATIONSHIP_74_FK on TBL_ACTIVO (
OBJ_ID ASC
)
go

create index MODELO_ACTIVO_FK on TBL_ACTIVO (
MDL_ID ASC
)
go

create index EMPRESA_ACTIVO_FK on TBL_ACTIVO (
EMP_ID ASC
)
go

create index PERSONAL_ACTIVO_FK on TBL_ACTIVO (
PER_ID ASC
)
go

create index IP_ACTIVO_FK on TBL_ACTIVO (
IP_ID ASC
)
go

create index PISO_ACTIVO_FK on TBL_ACTIVO (
PIS_ID ASC
)
go
```

```
create table TBL_ARCHIVOADJUNTO (
  ARA_ID          int          identity,
  DET_ID          int          null,
  ARA_DESCRIPCION varchar(100) not null,
  ARA_URL         varchar(500) not null,
  ARA_MODFEHCA   datetime     not null,
  ARA_MODUSUARIO int          not null,
  ARA_ESTADO     varchar(1)   not null,
  constraint PK_TBL_ARCHIVOADJUNTO primary key nonclustered (ARA_ID)
)
go
create index DETALLETICKET_ARCHIVOADJUNTO_FK on TBL_ARCHIVOADJUNTO (
  DET_ID ASC
)
go
create table TBL_AREA (
  ARE_ID          int          identity,
  ARE_DESCRIPCION varchar(100) not null,
  ARE_DIRECCIONAMIENTO bit      not null,
  ARE_MODFECHA   datetime     not null,
  ARE_MODUSUARIO int          not null,
  ARE_ESTADO     varchar(1)   not null,
  constraint PK_TBL_AREA primary key nonclustered (ARE_ID)
)
go
create table TBL_ASIGNACION (
  ASI_ID          int          identity,
  PLA_ID          int          null,
  ASI_FECHA       datetime     null,
  ASI_ESTADOFISICO bit        not null,
  ASI_MODFECHA   datetime     not null,
  ASI_MODUSUARIO int          not null,
  ASI_ESTADO     varchar(1)   not null,
  constraint PK_TBL_ASIGNACION primary key nonclustered (ASI_ID)
)
go
```

```
create index PLANIFICACION_ASIGNACION_FK on TBL_ASIGNACION (
  PLA_ID ASC
)
```

```
go
```

```
create table TBL_BITACORA (
  BIT_ID          int          identity,
  PER_ID          int          null,
  TIU_ID          int          null,
  BIT_COD         int          null,
  BIT_COD2        int          null,
  BIT_ACCION      varchar(100) null,
  BIT_OBJETO      varchar(100) null,
  BIT_IP          varchar(20)  null,
  BIT_MACADR      varchar(100) null,
  BIT_ESTACION   varchar(100) null,
  BIT_MODFECHA   datetime     not null,
  constraint PK_TBL_BITACORA primary key nonclustered (BIT_ID)
)
```

```
go
```

```
create index PERSONAL_BITACORA_FK on TBL_BITACORA (
  PER_ID ASC
)
```

```
go
```

```
create index TIPOUSUARIO_BITACORA_FK on TBL_BITACORA (
  TIU_ID ASC
)
```

```
go
```

```
create table TBL_CANAL (
  CAN_ID          int          identity,
  CAN_DESCRIPCION varchar(100)  not null,
  CAN_MODFECHA   datetime     not null,
  CAN_MODUSUARIO int          not null,
  CAN_ESTADO     varchar(1)   not null,
  constraint PK_TBL_CANAL primary key nonclustered (CAN_ID)
)
```

```
go
```

```
create table TBL_CARGO (
    CAR_ID          int          identity,
    DEP_ID          int          null,
    CAR_DESCRIPCION varchar(100) not null,
    CAR_MODFECHA   datetime     not null,
    CAR_MODUSUARIO int          not null,
    CAR_ESTADO     varchar(1)    not null,
    constraint PK_TBL_CARGO primary key nonclustered (CAR_ID)
)
go
create index DEPARTAMENTO_CARGO_FK on TBL_CARGO (
DEP_ID ASC
)
go
create table TBL_CLAVE (
    CLA_ID          int          identity,
    PER_ID          int          null,
    CLA_DESCRIPCION varchar(100) not null,
    CLA_MODFECHA   datetime     not null,
    CLA_MODUSUARIO int          not null,
    CLA_ESTADO     varchar(1)    not null,
    constraint PK_TBL_CLAVE primary key nonclustered (CLA_ID)
)
go
create index PERSONAL_CLAVE_FK on TBL_CLAVE (
PER_ID ASC
)
go
create table TBL_DEPARTAMENTO (
    DEP_ID          int          identity,
    EMP_ID          int          null,
    DEP_DESCRIPCION varchar(100) not null,
    DEP_MODFECHA   datetime     not null,
    DEP_MODUSUARIO int          not null,
    DEP_ESTADO     varchar(1)    not null,
    constraint PK_TBL_DEPARTAMENTO primary key nonclustered (DEP_ID)
```

```
)  
go  
create index EMPRESA_DEPARTAMENTO_FK on TBL_DEPARTAMENTO (  
EMP_ID ASC  
)  
go  
create table TBL_DETALLETICKET (  
    DET_ID          int          identity,  
    PER_ID          int          null,  
    EST_ID          int          null,  
    ETA_ID          int          null,  
    TIC_ID          int          null,  
    DET_ACTUAL      bit          not null,  
    DET_FECHAINICIO datetime     not null,  
    DET_FECHAFINAL  datetime     null,  
    DET_MODFECHA    datetime     not null,  
    DET_MODUSUARIO  int          not null,  
    DET_ESTADO      varchar(1)   not null,  
    constraint PK_TBL_DETALLETICKET primary key nonclustered (DET_ID)  
)  
go  
create index TICKET_DETALLETICKET_FK on TBL_DETALLETICKET (  
TIC_ID ASC  
)  
go  
create index ESTADO_DETALLETICKET_FK on TBL_DETALLETICKET (  
EST_ID ASC  
)  
go  
create index PERSONAL_DETALLETICKET_FK on TBL_DETALLETICKET (  
PER_ID ASC  
)  
go  
create index ETAPA_DETALLETICKET_FK on TBL_DETALLETICKET (  
ETA_ID ASC  
)
```

```
go
create table TBL_EMPRESA (
    EMP_ID          int          identity,
    EMP_DESCRIPCION varchar(100)  not null,
    EMP_MODFECHA   datetime      not null,
    EMP_MODUSUARIO int          not null,
    EMP_ESTADO     varchar(1)    not null,
    constraint PK_TBL_EMPRESA primary key nonclustered (EMP_ID)
)
go
create table TBL_EQUIPO (
    EQU_ID          int          identity,
    MDL_ID          int          null,
    PER_ID          int          null,
    TIE_ID          int          null,
    EMP_ID          int          null,
    IP_ID           int          null,
    PIS_ID          int          null,
    EQU_NUMSERIE   varchar(50)   null,
    EQU_OBSERVACION varchar(2000) null,
    EQU_MODFECHA   datetime      not null,
    EQU_MODUSUARIO int          not null,
    EQU_ESTADO     varchar(1)    not null,
    constraint PK_TBL_EQUIPO primary key nonclustered (EQU_ID)
)
go
create index PERSONAL_EQUIPO_FK on TBL_EQUIPO (
    PER_ID ASC
)
go
create index IP_EQUIPO_FK on TBL_EQUIPO (
    IP_ID ASC
)
go
create index TIPODEEQUIPO_EQUIPO_FK on TBL_EQUIPO (
    TIE_ID ASC
)
```

```
)  
go  
create index PISO_EQUIPO_FK on TBL_EQUIPO (  
PIS_ID ASC  
)  
go  
create index EMPRESA_EQUIPO_FK on TBL_EQUIPO (  
EMP_ID ASC  
)  
go  
create index MODELO_EQUIPO_FK on TBL_EQUIPO (  
MDL_ID ASC  
)  
go  
create table TBL_ESTADO (  
EST_ID int identity,  
EST_DESCRIPCION varchar(100) not null,  
EST_MODFECHA datetime not null,  
EST_MODUSUARIO int not null,  
EST_ESTADO varchar(1) not null,  
constraint PK_TBL_ESTADO primary key nonclustered (EST_ID)  
)  
go  
create table TBL_ETAPA (  
ETA_ID int identity,  
SUA_ID int null,  
ETA_DESCRIPCION varchar(100) not null,  
ETA_MODFECHA datetime not null,  
ETA_MODUSUARIO int not null,  
ETA_ESTADO varchar(1) not null,  
constraint PK_TBL_ETAPA primary key nonclustered (ETA_ID)  
)  
go  
create index SUBAREA_ETAPA_FK on TBL_ETAPA (  
SUA_ID ASC  
)
```

```
go
create table TBL_FERIADO (
    FER_ID          int          identity,
    FER_FECHA       datetime     not null,
    FER_DESCRIPCION varchar(100)  null,
    FER_MODFECHA    datetime     not null,
    FER_MODUSUARIO  int          not null,
    FER_ESTADO      varchar(1)   not null,
    constraint PK_TBL_FERIADO primary key nonclustered (FER_ID)
)
go
create table TBL_GENERO (
    GEN_ID          int          identity,
    GEN_DESCRIPCION varchar(100)  not null,
    GEN_MODFECHA    datetime     not null,
    GEN_MODUSUARIO  int          not null,
    GEN_ESTADO      varchar(1)   not null,
    constraint PK_TBL_GENERO primary key nonclustered (GEN_ID)
)
go
create table TBL_IMPRESORA (
    IMP_ID          int          identity,
    EMP_ID          int          null,
    IMP_PADRE       int          null,
    IP_ID           int          null,
    PER_ID          int          null,
    TII_ID          int          null,
    MDL_ID          int          null,
    PIS_ID          int          null,
    IMP_DESCRIPCION varchar(2000) not null,
    IMP_MODFECHA    datetime     not null,
    IMP_MODUSUARIO  int          not null,
    IMP_ESTADO      varchar(1)   not null,
    constraint PK_TBL_IMPRESORA primary key nonclustered (IMP_ID)
)
go
```

```
create index MODELO_IMPRESORA_FK on TBL_IMPRESORA (
MDL_ID ASC
)
go
create index PISO_IMPRESORA_FK on TBL_IMPRESORA (
PIS_ID ASC
)
go
create index TIPOIMPRESORA_IMPRESORA_FK on TBL_IMPRESORA (
TII_ID ASC
)
go
create index IP_IMPRESORA_FK on TBL_IMPRESORA (
IP_ID ASC
)
go
create index IMPRESORA_IMPRESORAPADRE_FK on TBL_IMPRESORA (
IMP_PADRE ASC
)
go
create index PERSONAL_IMPRESORA_FK on TBL_IMPRESORA (
PER_ID ASC
)
go
create index EMPRESA_IMPRESORA_FK on TBL_IMPRESORA (
EMP_ID ASC
)
go
create table TBL_IP (
    IP_ID          int          identity,
    IP_TIPOIP      int          not null,
    IP_DESCRIPCION varchar(100)  not null,
    IP_OBSERVACION varchar(2000) null,
    IP_ESTADOFISICO bit        not null,
    IP_MODFECHA    datetime     not null,
    IP_MODUSUARIO  int          not null,
```

```
IP_ESTADO      varchar(1)      not null,
constraint PK_TBL_IP primary key nonclustered (IP_ID)
)
go
create table TBL_ITEM (
ITE_ID          int          identity,
EMP_ID          int          null,
ITE_DESCRIPCION varchar(100)  not null,
ITE_EXISTENCIA int          not null,
ITE_EXISTENCIAMINIMA int      null,
ITE_MODFECHA   datetime     not null,
ITE_MODUSUARIO int          not null,
ITE_ESTADO     varchar(1)    not null,
constraint PK_TBL_ITEM primary key nonclustered (ITE_ID)
)
go
create index EMPRESA_ITEM_FK on TBL_ITEM (
EMP_ID ASC
)
go
create table TBL_MARCA (
MAR_ID          int          identity,
MAR_DESCRIPCION varchar(100)  not null,
MAR_MODFECHA   datetime     not null,
MAR_MODUSUARIO int          not null,
MAR_ESTADO     varchar(1)    not null,
constraint PK_TBL_MARCA primary key nonclustered (MAR_ID)
)
go
create table TBL_MODELO (
MDL_ID          int          identity,
MAR_ID          int          null,
OBJ_ID          int          null,
MDL_DESCRIPCION varchar(100)  not null,
MDL_MODFECHA   datetime     not null,
MDL_MODUSUARIO int          not null,
```

```
MDL_ESTADO      varchar(1)      not null,
constraint PK_TBL_MODELO primary key nonclustered (MDL_ID)
)
go
create index OBJETO_MODELO_FK on TBL_MODELO (
OBJ_ID ASC
)
go
create index MARCA_MODELO_FK on TBL_MODELO (
MAR_ID ASC
)
go
create table TBL_MODULO (
MOD_ID          int          identity,
SOF_ID          int          null,
MOD_DESCRIPCION varchar(100)  not null,
MOD_OBSERVACION varchar(1000) null,
MOD_MODFECHA    datetime     not null,
MOD_MODUSUARIO  int          not null,
MOD_ESTADO      varchar(1)    not null,
constraint PK_TBL_MODULO primary key nonclustered (MOD_ID)
)
go
create index SOFTWARE_MODULO_FK on TBL_MODULO (
SOF_ID ASC
)
go
create table TBL_MODULOCARGO (
MOD_ID          int          not null,
CAR_ID          int          not null,
MOC_MODFECHA    datetime     not null,
MOC_MODUSUARIO  int          not null,
MOC_ESTADO      varchar(1)    not null,
constraint PK_TBL_MODULOCARGO primary key nonclustered (MOD_ID, CAR_ID)
)
go
```

```
create index TBL_MODULOCARGO_FK on TBL_MODULOCARGO (
MOD_ID ASC
)
go
create index TBL_MODULOCARGO2_FK on TBL_MODULOCARGO (
CAR_ID ASC
)
go
create table TBL_NUMEROTELEFONO (
    NUT_ID          int          identity,
    SUC_ID          int          null,
    NUT_DESCRIPCION varchar(15)  not null,
    NUT_MODFECHA   datetime     not null,
    NUT_MODUSUARIO int          not null,
    NUT_ESTADO     varchar(1)   not null,
    constraint PK_TBL_NUMEROTELEFONO primary key nonclustered (NUT_ID)
)
go
create index SUCURSAL_NUMEROTELEFONO_FK on TBL_NUMEROTELEFONO (
SUC_ID ASC
)
go
create table TBL_OBJETO (
    OBJ_ID          int          identity,
    OBJ_DESCRIPCION varchar(100)  not null,
    OBJ_PREGUNTA   varchar(200)  not null,
    OBJ_MODFECHA   datetime     not null,
    OBJ_MODUSUARIO int          not null,
    OBJ_ESTADO     varchar(1)   not null,
    OBJ_TIPOASIGNACION varchar(3) not null,
    constraint PK_TBL_OBJETO primary key nonclustered (OBJ_ID)
)
go
create table TBL_OBJETOREQUERIMIENTO (
    OBR_ID          int          identity,
    REQ_ID          int          null,
```

```

OBJ_ID          int          null,
OBR_CANTIDAD    int          not null,
OBR_MODFECHA    datetime     not null,
OBR_MODUSUARIO  int          not null,
OBR_ESTADO      varchar(1)   not null,
constraint PK_TBL_OBJETOREQUERIMIENTO primary key nonclustered (OBR_ID)
)
go
create index OBJETO_OBJETOREQUERIMIENTO_FK on TBL_OBJETOREQUERIMIENTO (
OBJ_ID ASC
)
go
create index REQUERIMIENTO_OBJETOREQUERIMIENTO_FK on
TBL_OBJETOREQUERIMIENTO (
REQ_ID ASC
)
go
create table TBL_OBJETOTICKET (
OBT_ID          int          identity,
SUM_ID          int          null,
ACT_ID          int          null,
SOF_ID          int          null,
TIC_ID          int          null,
EQU_ID          int          null,
TEL_ID          int          null,
OBJ_ID          int          null,
IMP_ID          int          null,
MOD_ID          int          null,
PRO_ID          int          null,
OBT_MODFEHCA    datetime     not null,
OBT_MODUSUARIO  int          not null,
OBT_ESTADO      varchar(1)   not null,
constraint PK_TBL_OBJETOTICKET primary key nonclustered (OBT_ID)
)
go
create index TICKET_OBJETOTICKET_FK on TBL_OBJETOTICKET (

```

```
TIC_ID ASC
)
go
create index OBJETO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
OBJ_ID ASC
)
go
create index ACTIVO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
ACT_ID ASC
)
go
create index EQUIPO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
EQU_ID ASC
)
go
create index TELEFONO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
TEL_ID ASC
)
go
create index IMPRESORA_OBJETOTICKET_FK on TBL_OBJETOTICKET (
IMP_ID ASC
)
go
create index SOFTWARE_OBJETOTICKET_FK on TBL_OBJETOTICKET (
SOF_ID ASC
)
go
create index MODULO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
MOD_ID ASC
)
go
create index SUBMODULO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
SUM_ID ASC
)
go
create index PROYECTO_OBJETOTICKET_FK on TBL_OBJETOTICKET (
```

PRO\_ID ASC

)

go

create table TBL\_OBSERVACION (

  OBS\_ID          int          identity,

  DET\_ID          int          null,

  OBS\_DESCRIPCION  varchar(2000)  not null,

  OBS\_MODFECHA    datetime      not null,

  OBS\_MOSUSUARIO  int          not null,

  OBS\_ESTADO      varchar(1)    not null,

  constraint PK\_TBL\_OBSERVACION primary key nonclustered (OBS\_ID)

)

go

create index DETALLETICKET\_OBSERVACION\_FK on TBL\_OBSERVACION (

  DET\_ID ASC

)

go

create table TBL\_PARAMETRO (

  PAR\_ID          int          identity,

  PAR\_MAIL        varchar(200)  not null,

  PAR\_CLAVE       varchar(200)  not null,

  PAR\_MODFECHA    datetime      not null,

  PAR\_MODUSUARIO  int          not null,

  PAR\_ESTADO      varchar(1)    not null,

  constraint PK\_TBL\_PARAMETRO primary key nonclustered (PAR\_ID)

)

go

create table TBL\_PERSONAL (

  PER\_ID          int          identity,

  GEN\_ID          int          null,

  CAR\_ID          int          null,

  SUC\_ID          int          null,

  TIU\_ID          int          null,

  TID\_ID          int          null,

  PER\_NOMBRE      varchar(100)  not null,

```

PER_APELLIDO      varchar(100)      not null,
PER_DOCUMENTO     varchar(20)       not null,
PER_MAIL          varchar(100)       not null,
PER_OBSERVACION   varchar(2000)     null,
PER_NIVEL         int           not null,
PER_MODFECHA      datetime      not null,
PER_MODUSUARIO    int           not null,
PER_ESTADO        varchar(1)       not null,
constraint PK_TBL_PERSONAL primary key nonclustered (PER_ID)
)
go
create index TIPOUSUARIO_PERSONAL_FK on TBL_PERSONAL (
TIU_ID ASC
)
go
create index CARGO_PERSONAL_FK on TBL_PERSONAL (
CAR_ID ASC
)
go
create index TIPODOCUMENTO_PERSONAL_FK on TBL_PERSONAL (
TID_ID ASC
)
go
create index GENERO_PERSONAL_FK on TBL_PERSONAL (
GEN_ID ASC
)
go
create index SUCURSAL_PERSONAL_FK on TBL_PERSONAL (
SUC_ID ASC
)
go
create table TBL_PERSONALSUBAREA (
PER_ID          int           not null,
SUA_ID          int           not null,
PES_MODFECHA    datetime      not null,

```

```
PES_MODUSUARIO    int          not null,
PES_ESTADO        varchar(1)   not null,
constraint PK_TBL_PERSONALSUBAREA primary key nonclustered (PER_ID, SUA_ID)
)
go
create index TBL_PERSONALSUBAREA_FK on TBL_PERSONALSUBAREA (
PER_ID ASC
)
go
create index TBL_PERSONALSUBAREA2_FK on TBL_PERSONALSUBAREA (
SUA_ID ASC
)
go
create table TBL_PERSONALTICKET (
PER_ID            int          not null,
TIC_ID            int          not null,
PET_MODFECHA      datetime     not null,
PET_MODUSUARIO    int          not null,
PET_ESTADO        varchar(1)   not null,
constraint PK_TBL_PERSONALTICKET primary key nonclustered (PER_ID, TIC_ID)
)
go
create index TBL_PERSONALTICKET_FK on TBL_PERSONALTICKET (
PER_ID ASC
)
go
create index TBL_PERSONALTICKET2_FK on TBL_PERSONALTICKET (
TIC_ID ASC
)
go
create table TBL_PISO (
PIS_ID            int          identity,
SUC_ID            int          null,
PIS_DESCRIPCION   varchar(100) not null,
PIS_MODFECHA      datetime     not null,
PIS_MODUSUARIO    int          not null,
```

```
PIS_ESTADO      varchar(1)      not null,
constraint PK_TBL_PISO primary key nonclustered (PIS_ID)
)
go
create index SUCURSAL_PISO_FK on TBL_PISO (
SUC_ID ASC
)
go
create table TBL_PLANIFICACION (
    PLA_ID        int          identity,
    TIC_ID        int          null,
    PER_ID        int          null,
    PLA_FECHAINICIO  datetime    not null,
    PLA_FECHAFINAL  datetime    not null,
    PLA_HORAS      int          not null,
    PLA_MODFECHA    datetime    not null,
    PLA_MODUSUARIO  int          not null,
    PLA_ESTADO     varchar(1)    not null,
    constraint PK_TBL_PLANIFICACION primary key nonclustered (PLA_ID)
)
go
create index TICKET_PLANIFICACION_FK on TBL_PLANIFICACION (
TIC_ID ASC
)
go
create index PERSONAL_PLANIFICACION_FK on TBL_PLANIFICACION (
PER_ID ASC
)
go
create table TBL_PRIORIDAD (
    PRI_ID        int          identity,
    PRI_DESCRIPCION  varchar(100)  not null,
    PRI_MODFECHA    datetime    not null,
    PRI_MODUSUARIO  int          not null,
    PRI_ESTADO     varchar(1)    not null,
    constraint PK_TBL_PRIORIDAD primary key nonclustered (PRI_ID)
```

```
)  
go  
create table TBL_PROYECTO (  
    PRO_ID          int          identity,  
    PRO_DESCRIPCION varchar(100)  not null,  
    PRO_OBSERVACION varchar(2000) null,  
    PRO_FECHAINICIO datetime     not null,  
    PRO_FECHAFINAL  datetime     null,  
    PRO_HORASSOLUCION int         null,  
    PRO_ASIGNACION  int          null,  
    PRO_MODFECHA    datetime     not null,  
    PRO_MODUSUARIO  int          not null,  
    PRO_ESTADO      varchar(1)   not null,  
  
    constraint PK_TBL_PROYECTO primary key nonclustered (PRO_ID)  
  
)  
go  
create table TBL_REPORTE (  
    REP_ID          int          not null,  
    PER_ID          int          null,  
    PER_NOMBRE      varchar(100) null,  
    EST_ID         int          null,  
    EST_DESCRIPCION varchar(100) null,  
    REQ_ID         int          null,  
    REQ_DESCRIPCION varchar(100) null,  
    TIR_ID         int          null,  
    TIR_DESCRIPCION varchar(100) null,  
    SOF_ID         int          null,  
    SOF_DESCRIPCION varchar(100) null,  
    MOD_ID         int          null,  
    MOD_DESCRIPCION varchar(100) null,  
    SUM_ID         int          null,  
    SUM_DESCRIPCION varchar(100) null,  
    EMP_ID         int          null,  
    EMP_DESCRIPCION varchar(100) null,  
    SUC_ID         int          null,  
    SUC_DESCRIPCION varchar(100) null,
```

```

CAN_ID          int          null,
CAN_DESCRIPCION varchar(100)  null,
REP_IDENTIFICADOR int        null,
REP_TICPENDIENTES int        null,
REP_TICCERRADOS int          null,
REP_TICTOTAL    int          null,
REP_TIEMPOTOTAL int          null,
REP_PROMEDIOATENCION varchar(200) null,
REP_TOTALATENCION varchar(200) null,
REP_INCIDENTEDIA decimal(10,2) null,
REP_PORCENTAJETOTAL int        null,
REP_MODFECHA    datetime     null,
REP_MODUSUARIO  int          null,
REP_ESTADO      varchar(1)    null,
constraint PK_TBL_REPORTE primary key nonclustered (REP_ID)
)
go
create table TBL_REQUERIMIENTO (
    REQ_ID          int          identity,
    SUA_ID          int          null,
    PER_ID          int          null,
    TIT_ID          int          null,
    PRI_ID          int          null,
    TIR_ID          int          null,
    REQ_DESCRIPCION varchar(2000) not null,
    REQ_HORASRESP  int          null,
    REQ_NIVEL      int          not null,
    REQ_AUTORIZACION bit        not null,
    REQ_MODFECHA   datetime     not null,
    REQ_MODUSUARIO int          not null,
    REQ_ESTADO     varchar(1)    not null,
    constraint PK_TBL_REQUERIMIENTO primary key nonclustered (REQ_ID)
)
go
create index TIPOREQ_REQUERIMIENTO_FK on TBL_REQUERIMIENTO (
    TIR_ID ASC

```

```
)  
go  
create index SUBAREA_REQUERIMIENTO_FK on TBL_REQUERIMIENTO (  
SUA_ID ASC  
)  
go  
create index PRIORIDAD_REQUERIMIENTO_FK on TBL_REQUERIMIENTO (  
PRI_ID ASC  
)  
go  
create index PERSONAL_REQUERIMIENTO_FK on TBL_REQUERIMIENTO (  
PER_ID ASC  
)  
go  
create index TIPOTICKET_REQUERIMIENTO_FK on TBL_REQUERIMIENTO (  
TIT_ID ASC  
)  
go  
create table TBL_RESPUESTA (  
RES_ID int identity,  
TIC_ID int null,  
OBJ_ID int null,  
RES_DESCRIPCION varchar(2000) not null,  
RES_MODFECHA datetime not null,  
RES_MODUSUARIO int not null,  
RES_ESTADO varchar(1) not null,  
constraint PK_TBL_RESPUESTA primary key nonclustered (RES_ID)  
)  
go  
create index TICKET_RESPUESTA_FK on TBL_RESPUESTA (  
TIC_ID ASC  
)  
go  
create index OBJETO_RESPUESTA_FK on TBL_RESPUESTA (  
OBJ_ID ASC  
)
```

go

```
create table TBL_SOFTWARE (
  SOF_ID          int          identity,
  SOF_DESCRIPCION varchar(100)  not null,
  SOF_TIPO        varchar(1)   null,
  SOF_MODFECHA    datetime     not null,
  SOF_MODUSUARIO  int          not null,
  SOF_ESTADO      varchar(1)   not null,
  constraint PK_TBL_SOFTWARE primary key nonclustered (SOF_ID)
)
```

go

```
create table TBL_SOLUCION (
  SOL_ID          int          identity,
  REQ_ID          int          null,
  SOL_DESCRIPCION varchar(2000) not null,
  SOL_MODFECHA    datetime     not null,
  SOL_MODUSUARIO  int          not null,
  SOL_ESTADO      varchar(1)   not null,
  constraint PK_TBL_SOLUCION primary key nonclustered (SOL_ID)
)
```

go

```
create index REQUERIMIENTO_SOLUCION_FK on TBL_SOLUCION (
  REQ_ID ASC
)
```

go

```
create table TBL_SUBAREA (
  SUA_ID          int          identity,
  ARE_ID          int          null,
  SUA_DESCRIPCION varchar(100)  not null,
  SUA_MODFECHA    datetime     not null,
  SUA_MODUSUARIO  int          not null,
  SUA_ESTADO      varchar(1)   not null,
  constraint PK_TBL_SUBAREA primary key nonclustered (SUA_ID)
)
```

go

```
create index AREA_SUBAREA_FK on TBL_SUBAREA (
ARE_ID ASC
)
go
create table TBL_SUBMODULO (
SUM_ID int identity,
MOD_ID int null,
SUM_DESCRIPCION varchar(100) not null,
SUM_MODFECHA datetime not null,
SUM_MODUSUARIO int not null,
SUM_ESTADO varchar(1) not null,
constraint PK_TBL_SUBMODULO primary key nonclustered (SUM_ID)
)
go
create index MODULO_SUBMODULO_FK on TBL_SUBMODULO (
MOD_ID ASC
)
go
create table TBL_SUCURSAL (
SUC_ID int identity,
EMP_ID int null,
SUC_DESCRIPCION varchar(100) not null,
SUC_DIRECCION varchar(100) null,
SUC_MODFECHA datetime not null,
SUC_MODUSUARIO int not null,
SUC_ESTADO varchar(1) not null,
constraint PK_TBL_SUCURSAL primary key nonclustered (SUC_ID)
)
go
create index EMPRESA_SUCURSAL_FK on TBL_SUCURSAL (
EMP_ID ASC
)
go
create table TBL_TAREA (
TAR_ID int identity,
TIP_ID int null,
```

```
REQ_ID          int          null,
TAR_OBSERVACION varchar(1000)  not null,
TAR_FECHAHORA   datetime     not null,
TAR_MODFECHA    datetime     not null,
TAR_MODUSUARIO  int          not null,
TAR_ESTADO      varchar(1)    not null,
constraint PK_TBL_TAREA primary key nonclustered (TAR_ID)
)
go
create index REQUERIMIENTO_TAREA_FK on TBL_TAREA (
REQ_ID ASC
)
go
create index TIPOPERIODO_TAREA_FK on TBL_TAREA (
TIP_ID ASC
)
go
create table TBL_TELEFONO (
TEL_ID          int          identity,
IP_ID           int          null,
EMP_ID          int          null,
SUC_ID          int          null,
PIS_ID          int          null,
PER_ID          int          null,
MDL_ID          int          null,
TEL_EXTENCION   varchar(6)    null,
TEL_OBSERVACION varchar(2000)  null,
TEL_MODFECHA    datetime     not null,
TEL_MODUSUARIO  int          not null,
TEL_ESTADO      varchar(1)    not null,
constraint PK_TBL_TELEFONO primary key nonclustered (TEL_ID)
)
go
create index IP_TELEFONO_FK on TBL_TELEFONO (
IP_ID ASC
)
```

```
go
create index SUCURSAL_TELEFONO_FK on TBL_TELEFONO (
SUC_ID ASC
)
go
create index MODELO_TELEFONO_FK on TBL_TELEFONO (
MDL_ID ASC
)
go
create index PISO_TELEFONO_FK on TBL_TELEFONO (
PIS_ID ASC
)
go
create index PERSONAL_TELEFONO_FK on TBL_TELEFONO (
PER_ID ASC
)
go
create index EMPRESA_TELEFONO_FK on TBL_TELEFONO (
EMP_ID ASC
)
go
create table TBL_TICKET (
    TIC_ID          int          identity,
    SOL_ID          int          null,
    PRI_ID          int          null,
    REQ_ID          int          null,
    ETA_ID          int          null,
    CAN_ID          int          null,
    TIT_ID          int          null,
    EST_ID          int          null,
    PER_ID          int          null,
    PER_CREADOR    int          null,
    PER_DIRECCIONADOR int      null,
    PER_AUTORIZADOR int      null,
    TIC_PADRE      int          null,
    TIC_FECHAINICIO datetime    not null,
```

```
TIC_FECHAFINAL    datetime    null,
TIC_FECHAENTREGA  datetime    null,
TIC_AUTORIZACION  int         null,
TIC_CALIFICACION  int         null,
TIC_DIRECCIONADO  bit         null,
TIC_OBSERVACION   varchar(2000) null,
TIC_MODFECHA      datetime    not null,
TIC_MODUSUARIO    int         not null,
TIC_ESTADO        varchar(1)  not null,
constraint PK_TBL_TICKET primary key nonclustered (TIC_ID)
)
go
create index CANAL_TICKET_FK on TBL_TICKET (
CAN_ID ASC
)
go
create index PERSONAL_TICKET_FK on TBL_TICKET (
PER_ID ASC
)
go
create index SOLUCION_TICKET_FK on TBL_TICKET (
SOL_ID ASC
)
go
create index USUDIREC_TICKET_FK on TBL_TICKET (
PER_DIRECCIONADOR ASC
)
go
create index REQUERIMIENTO_TICKET_FK on TBL_TICKET (
REQ_ID ASC
)
go
create index TICKET_TICKETPADRE_FK on TBL_TICKET (
TIC_PADRE ASC
)
go
```

```
create index ESTADO_TICKET_FK on TBL_TICKET (
EST_ID ASC
)
go
create index ETAPA_TICKET_FK on TBL_TICKET (
ETA_ID ASC
)
go
create index PERSONALCREADOR_TICKET_FK on TBL_TICKET (
PER_CREADOR ASC
)
go
create index PRIORIDAD_TICKET_FK on TBL_TICKET (
PRI_ID ASC
)
go
create index PERSONALAUTORIZADOR_TICKET_FK on TBL_TICKET (
PER_AUTORIZADOR ASC
)
go
create index TIPOTICKET_TICKET_FK on TBL_TICKET (
TIT_ID ASC
)
go
create table TBL_TIPODOCUMENTO (
    TID_ID          int          identity,
    TID_DESCRIPCION varchar(100)  not null,
    TID_MODFECHA   datetime     not null,
    TID_MODUSUARIO int          not null,
    TID_ESTADO     varchar(1)   not null,
    constraint PK_TBL_TIPODOCUMENTO primary key nonclustered (TID_ID)
)
go
create table TBL_TIPOEQUIPO (
    TIE_ID          int          identity,
    TIE_DESCRIPCION varchar(100)  not null,
```

```
TIE_MODFECHA    datetime    not null,
TIE_MODUSUARIO  int        not null,
TIE_ESTADO      varchar(1)  not null,
constraint PK_TBL_TIPOEQUIPO primary key nonclustered (TIE_ID)
)
go
create table TBL_TIPOIMPRESORA (
    TII_ID        int            identity,
    TII_DESCRIPCION  varchar(100)  not null,
    TII_MODFECHA    datetime      not null,
    TII_MODUSUARIO  int          not null,
    TII_ESTADO      varchar(1)    not null,
    constraint PK_TBL_TIPOIMPRESORA primary key nonclustered (TII_ID)
)
go
create table TBL_TIPOPERIODO (
    TIP_ID        int            identity,
    TIP_DESCRIPCION  varchar(100)  not null,
    TIP_MODFECHA    datetime      not null,
    TIP_MODUSUARIO  int          not null,
    TIP_ESTADO      varchar(1)    not null,
    constraint PK_TBL_TIPOPERIODO primary key nonclustered (TIP_ID)
)
go
create table TBL_TIPOREQUERIMIENTO (
    TIR_ID        int            identity,
    TIR_DESCRIPCION  varchar(100)  not null,
    TIR_OBSERVACION  varchar(1000)  null,
    TIR_NIVEL       int          not null,
    TIR_MODFECHA    datetime      not null,
    TIR_MODUSUARIO  int          not null,
    TIR_ESTADO      varchar(1)    not null,
    constraint PK_TBL_TIPOREQUERIMIENTO primary key nonclustered (TIR_ID)
)
go
```

```
create table TBL_TIPOTICKET (  
    TIT_ID          int          identity,  
    TIT_DESCRIPCION  varchar(100)  not null,  
    TIT_MODFECHA    datetime    not null,  
    TIT_MODUSUARIO  int          not null,  
    TIT_ESTADO      varchar(1)   not null,  
    constraint PK_TBL_TIPOTICKET primary key nonclustered (TIT_ID)  
)  
go  
create table TBL_TIPOUSUARIO (  
    TIU_ID          int          identity,  
    TIU_DESCRIPCION  varchar(100)  not null,  
    TIU_NIVEL       int          not null,  
    TIU_MODFECHA    datetime    not null,  
    TIU_MODUSUARIO  int          not null,  
    TIU_ESTADO      varchar(1)   not null,  
    constraint PK_TBL_TIPOUSUARIO primary key nonclustered (TIU_ID)  
)  
go  
create table TBL_TONER (  
    TON_ID          int          identity,  
    TON_DESCRIPCION  varchar(100)  not null,  
    TON_EXISTENCIA  int          not null,  
    TON_MODFECHA    datetime    not null,  
    TON_MODUSUARIO  int          not null,  
    TON_ESTADO      varchar(1)   not null,  
    constraint PK_TBL_TONER primary key nonclustered (TON_ID)  
)  
go  
create table TBL_TONERIMPRESORA (  
    TON_ID          int          not null,  
    IMP_ID          int          not null,  
    TOI_MODFECHA    datetime    not null,  
    TOI_MODUSUARIO  int          not null,  
    TOI_ESTADO      varchar(1)   not null,  
    constraint PK_TBL_TONERIMPRESORA primary key nonclustered (TON_ID, IMP_ID)
```

```
)  
go  
create index TBL_TONERIMPRESORA_FK on TBL_TONERIMPRESORA (  
TON_ID ASC  
)  
go  
create index TBL_TONERIMPRESORA2_FK on TBL_TONERIMPRESORA (  
IMP_ID ASC  
)  
go
```

## A.16 Herramientas

Para cubrir la necesidad de un sistema integrado bajo web, que facilite sus actividades empresariales, al menor costo posible, y con tecnología de punta.

En base a los objetivos descritos, trabajaremos con las siguientes herramientas.

\* C#.

\* ASP.NET

\* Javascript

\* JQuery

\* Sql Server

### 1. Descripción de C#.

Microsoft C# es un nuevo lenguaje de programación diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Supone una evolución de Microsoft C y Microsoft C++; es sencillo, moderno, proporciona seguridad de tipos y está orientado a objetos. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de Common Language Runtime. Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones.

C# se presenta como Visual C# en el conjunto de programas Visual Studio .NET.

Visual C# utiliza plantillas de proyecto, diseñadores, páginas de propiedades,

asistentes de código, un modelo de objetos y otras características del entorno de desarrollo. La biblioteca para programar en Visual C# es .NET Framework.

## **2. Descripción de ASP.NET.**

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

## **3. Descripción de JavaScript.**

JavaScript (abreviado comúnmente "JS") 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 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 provee al lenguaje JavaScript de 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.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases

#### **4. Descripción de JQuery.**

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas. Microsoft la añadirá en su IDE Visual Studio y la usará junto con los

frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time.

### **5. Descripción de SQL Server.**

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

## A.17 Bibliografía

- Monografias.com S.A. (2013), Metodología ITIL,  
<http://www.monografias.com/trabajos31/metodologia-itol/metodologia-itol.shtml>
- sfe, Estandarizacion de bases de datos orientadas a objetos.  
[https://www.sfe.go.cr/documentos/procedimientos\\_sustantivos/TI\\_I\\_02\\_Estandares\\_de\\_bases\\_de\\_datos.pdf](https://www.sfe.go.cr/documentos/procedimientos_sustantivos/TI_I_02_Estandares_de_bases_de_datos.pdf)
- Microsoft, Convenciones de código de C# (Guía de programación de C#).  
<http://msdn.microsoft.com/es-es/library/ff926074.aspx>
- Microsoft, Guia de instalación para sql server 2012.  
<http://blogs.msdn.com/b/warnov/archive/2011/12/05/paso-a-paso-para-instalar-sql-server-2012-evaluation-en-windows-7-sp1-32-bits.aspx>
- Microsoft, Lenguaje Visual C#. [http://msdn.microsoft.com/es-es/library/aa287558\(v=vs.71\).aspx](http://msdn.microsoft.com/es-es/library/aa287558(v=vs.71).aspx)
- Wikipedia, Asp.Net teoría general. <http://es.wikipedia.org/wiki/ASP.NET>
- Wikipedia, JavaScript teoría general. <http://es.wikipedia.org/wiki/JavaScript>
- Wikipedia, JQuery teoría general. <http://es.wikipedia.org/wiki/JQuery>
- Wikipedia, Sql Server teoría general.  
[http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

## A.18 Enlaces

- <http://www.desarrolloweb.com/manuales/manual-jquery.html>
- <http://librosweb.es/javascript/>
- <https://www.youtube.com/playlist?list=PLE1E885386F318F65>
- [http://msdn.microsoft.com/es-es/library/vstudio/bb399182\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/vstudio/bb399182(v=vs.100).aspx)
- <http://msdn.microsoft.com/es-es/library/bb934498.aspx>
- <http://jqueryui.com/>