



**UNIVERSIDAD TÉCNICA DEL NORTE.**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES.**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO EN**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES.**

**TEMA:**

**“ANÁLISIS DEL FRAMEWORK ASP.NET MVC 5. PROTOTIPO SISTEMA PARA  
EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS  
DE CREATEC S.A.”**

**APLICATIVO:**

**“SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS  
RECURSOS HUMANOS DE CREATEC S.A.”**

**Autor:** Jimmy Fabricio Gómez Cadena.

**Director:** Ing. Diego Javier Trejo España.

**Ibarra – Ecuador.**

**2015.**



## UNIVERSIDAD TÉCNICA DEL NORTE

### CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **JIMMY FABRICIO GÓMEZ CADENA**, con cédula de ciudadanía Nro.1003266952, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: “**Análisis del framework ASP.NET MVC 5. Prototipo sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A.**” con el aplicativo “**Prototipo sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A.**”, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

.....  
**Firma.**

**Nombre:** JIMMY FABRICIO GÓMEZ CADENA.

**Cédula:** 1003266952.

Ibarra a los 10 días del mes de Febrero de 2015.

# UNIVERSIDAD TÉCNICA DEL NORTE



## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto de Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

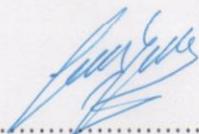
Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

<b>DATOS DE CONTACTO</b>	
<b>CÉDULA DE CIUDADANÍA</b>	1003266952
<b>APELLIDOS Y NOMBRES</b>	Gómez Cadena Jimmy Fabricio.
<b>DIRECCIÓN</b>	Ibarra Luis Eduardo Dávila 1-100
<b>EMAIL</b>	jimmy_31@hotmail.es
<b>TELÉFONO MÓVIL</b>	0997790801

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	ANÁLISIS DEL FRAMEWORK ASP.NET MVC 5. PROTOTIPO SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.
<b>AUTOR:</b>	JIMMY FABRICIO GÓMEZ CADENA.
<b>FECHA:</b>	2015-01-10
<b>PROGRAMA:</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERÍA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR:</b>	ING. DIEGO TREJO ESPAÑA

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, **Jimmy Fabricio Gómez Cadena**, con cédula de ciudadanía Nro. 1003266952, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Firma.....

**Nombre:** Jimmy Fabricio Gómez Cadena.

**Cédula:** 1003266952.

Ibarra a los 10 días del mes de Febrero de 2015.

**CONSTANCIAS**

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 10 días del mes de Febrero de 2015.

Firma.....  


**Nombre:** Jimmy Gómez

**CC:** 1003266952.

Firma.....  


**Nombre:** ING. BETHY CHÁVEZ.

**Cargo:** JEFE DE BIBLIOTECA.

**CERTIFICACIÓN**

Certifico que la tesis “Análisis del framework ASP.NET MVC 5. Prototipo sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A.” ha sido realizada en su totalidad por el señor: Jimmy Fabricio Gómez Cadena portador de la cédula de ciudadanía número: 1003266952.

Presente

De mis consideraciones:

Siendo suscriptores del proyecto de tesis del Egresado JIMMY FABRICIO GÓMEZ CADENA con C.C. 1003266952 quien describió su trabajo con el tema “ANÁLISIS DEL FRAMEWORK

ASP.NET MVC 5. PROTOTIPO SISTEMA PARA EL CONTROL INTERNO DE USO DE TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”, me es grato informar

.....  
Ing. Diego Javier Trejo España.

Director de tesis.

**CERTIFICACIÓN**

Quito, 10 de Febrero del 2015

Señores

UNIVERSIDAD TÉCNICA DEL NORTE

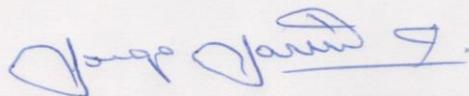
Presente

De mis consideraciones:

Siendo auspiciantes del proyecto de tesis del Egresado JIMMY FABRICIO GÓMEZ CADENA con C.C: 1003266952 quien desarrolló su trabajo con el tema **“ANÁLISIS DEL FRAMEWORK ASP.NET MVC 5. PROTOTIPO SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”**, me es grato informar que se ha superado a satisfacción las pruebas técnicas y la revisión del cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado por parte del egresado: JIMMY FABRICIO GÓMEZ CADENA.

El egresado JIMMY FABRICIO GÓMEZ CADENA puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

ATENTAMENTE:



.....  
Jorge Jarrín Caicedo  
Presidente Ejecutivo  
CREATEC S.A

**CTF CREATEC S.A.**

**DEDICATORIA.****A:**

Dios, por ser quien guía el camino en mi vida, por darme las fuerzas para seguir adelante en los momentos más difíciles y permitirme cumplir todos mis objetivos y metas.

Mis padres Jaime Gómez y Patricia Cadena, por ser un ejemplo para mí, por apoyarme incondicionalmente en cada momento de mi vida, por infundir valores y buenos consejos que son parte importante en mi formación como persona.

Mis hermanos Pablo Gómez y José Luis Gómez, por tantas experiencias vividas en nuestra niñez y juventud.

## **AGRADECIMIENTOS.**

### **A:**

Mis padres, por sus buenos consejos a lo largo de mi vida, y por darme la educación que es la mejor herencia que un hijo puede recibir.

Mi tía Laura Gómez y su esposo Jorge Jarrín, que con sus conocimientos y experiencia me brindan apoyo desde el inicio de mi carrera, por incentivar me a seguir adelante, son pilar importante en mi formación como profesional y personal, por darme la oportunidad y las facilidades para poder desarrollar el presente proyecto de grado en la empresa CREATEC S.A.

Ingeniero Diego Trejo, por su constante ayuda y asesoramiento a lo largo de mi carrera y en la ejecución del proyecto de grado.

Los docentes de la Universidad Técnica del Norte, por impartir sus conocimientos y apoyar a conseguir concluir mis estudios en tan prestigiosa institución.

## ÍNDICE DE CONTENIDO.

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	ii
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	iii
1. IDENTIFICACIÓN DE LA OBRA .....	iii
2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD .....	iv
CONSTANCIAS .....	iv
CERTIFICACIÓN .....	vi
CERTIFICACIÓN .....	vii
DEDICATORIA .....	vii
AGRADECIMIENTOS .....	ix
ÍNDICE DE CONTENIDO .....	x
ÍNDICE DE ILUSTRACIONES .....	xiv
ÍNDICE DE TABLAS .....	xvii
RESUMEN .....	xix
ABSTRACT .....	xx
CAPÍTULO I .....	1
1. Introducción .....	1
1.1. Problema .....	1
1.1.1. Antecedentes .....	1
1.1.2. Situación actual .....	2
1.1.3. Planteamiento del problema .....	3
1.1.4. Prospectiva .....	3
1.2. Objetivos .....	3
1.2.1. Objetivo general .....	3
1.2.2. Objetivos específicos .....	4
1.3. Alcance .....	4
1.4. Justificación .....	6
1.5. Presupuesto .....	7
CAPÍTULO II .....	8
2. Marco teórico .....	8
2.1. Patrón de diseño MVC .....	8

2.1.1. Modelo. ....	8
2.1.2. Vista. ....	8
2.1.3. Controlador. ....	9
2.1.4. Flujo de control. ....	9
2.1.5. MVC en aplicaciones Web. ....	9
2.2. ASP.NET MVC. ....	10
2.2.1. Características del marco ASP.NET MVC. ....	12
2.2.2. Cuando desarrollar con ASP.NET MVC. ....	12
2.2.3. Ventajas de desarrollar con ASP.NET MVC. ....	13
2.2.4. Referencias de ASP.NET MVC. ....	13
2.3. Introducción A ASP.NET MVC 5. ....	25
2.3.1. Lo nuevo en ASP.NET MVC 5. ....	25
2.3.2. Primer proyecto en ASP.NET MVC 5. ....	25
2.3.3. Añadir controladores. ....	30
2.3.4. Añadir vistas. ....	34
2.3.4. Añadir modelado. ....	38
2.4. Introducción a ENTITY FRAMEWORK 6 EN ASP.NET MVC 5. ....	39
2.4.1. Entity Framework 6, Database first. ....	39
2.4.2. Creación de una aplicación web MVC 5 y modelo de datos. ....	41
2.4.3. Generación de vistas. ....	45
2.4.4. Cambios en la Base de datos. ....	50
2.4.5. Personalización de vistas. ....	52
2.4.6. Entity Framework 6, Code first. ....	54
2.5. Vistas Razor en ASP.NET MVC 5. ....	59
2.5.1. Programación básica en Razor. ....	59
2.5.2. Reglas de la sintaxis Razor. ....	61
2.5.3. Pasar información de vista a controlador. ....	63
2.6. Controladores y enrutamiento. ....	67
2.6.1. Acciones en un controlador. ....	68
2.6.2. Action Result. ....	68
2.6.3. Marcar métodos públicos como métodos de no acción. ....	69
2.6.4. Creación de rutas personalizadas. ....	70
2.7. Validaciones. ....	71
2.8. Características para móviles de ASP.NET MVC 5. ....	73

2.8.1. Bootstrap.....	74
2.8.2. JQuery Mobile.....	77
2.9. Publicar en IIS.....	79
CAPÍTULO III.....	86
3. Diagnóstico de situación actual y Marco propositivo del framework ASP.NET MVC. .	86
3.1. Situación actual del FRAMEWORK ASP.NET MVC.....	86
3.2. ASP.NET MVC vs ASP.NET WEB FORMS .....	88
3.2.1. Curva de aprendizaje.....	89
3.2.2. Escalabilidad.....	89
3.2.3. Compatibilidad.....	90
3.2.4. Rendimiento.....	90
3.2.5. Tendencia de uso.....	91
3.2.6. Soporte técnico.....	91
3.2.7. Fácil desarrollo.....	91
3.2.8. Pruebas unitarias.....	92
3.2.9. Control de HTML.....	92
3.2.10. Arquitectura.....	92
3.3. Cuadro comparativo.....	93
3.4. Conclusión.....	93
3.5. Marco propositivo de ASP.NET MVC.....	94
CAPÍTULO IV .....	95
4. Desarrollo de prototipo de sistema de control interno de uso del tiempo de los recursos humano de CREATEC.....	95
4.1. Planeación del proyecto.....	95
4.1.1. Relato del usuario.....	95
4.1.2. Historias de usuario.....	97
4.1.3. Personas y roles del proyecto.....	100
4.1.4. Lista de Objetivos.....	100
4.1.5. Pila de producto.....	102
4.1.6. Planificación de las iteraciones.....	103
4.2. Ejecución de las iteraciones.....	107
4.2.1. Primera iteración (Levantamiento y elaboración de procesos).....	107
4.2.2. Segunda iteración (Construcción de base de datos del sistema).....	110
4.2.3. Tercera iteración (Módulo de clientes y personal).....	111

4.2.4. Cuarta iteración (Módulo de proyectos y Bitácora).....	112
4.2.5. Quinta iteración (Administración de usuarios, parámetros y auditoría).....	113
4.2.6. Sexta iteración (Resultados y reportes).....	114
4.2.7. Séptima iteración (Pruebas y publicación).....	115
4.3. Demostración de los resultados de iteraciones. ....	116
4.3.1. Resultados de primera iteración. ....	116
4.3.2. Resultados de segunda iteración. ....	127
4.3.3. Resultados de tercera iteración. ....	133
4.3.4. Resultados de cuarta iteración.....	133
4.3.5. Resultados de quinta iteración. ....	134
4.3.6. Resultados de sexta iteración. ....	135
4.3.7. Resultados de séptima iteración. ....	135
CAPÍTULO V.....	137
5. Análisis de impactos. ....	137
5.1. Impacto productivo. ....	138
5.2. Impacto organizativo.....	139
5.3. Impacto económico. ....	139
5.4. Impacto innovación.....	140
5.5. Impacto humano.....	141
5.6. CONCLUSIONES. ....	142
5.7. RECOMENDACIONES.....	144
REFERENCIAS BIBLIOGRÁFICAS. ....	145
ANEXOS. ....	147
ANEXO 1: Ejemplos prácticos del uso del framework MVC 5.....	148
ANEXO 2: Manual técnico del aplicativo web. ....	148
ANEXO 3: Manual de usuario del aplicativo web. ....	148
ANEXO 4: Código fuente del aplicativo web. ....	148
ANEXO 5: Backup de la base de datos. ....	148

## ÍNDICE DE ILUSTRACIONES

Ilustración 1. Alcance del proyecto. ....	5
Ilustración 2. Modelo MVC.....	9
Ilustración 3. Asp.Net MVC.....	10
Ilustración 4. ASP. NET MVC.....	11
Ilustración 5. Visual Studio 2013. ....	26
Ilustración 6. Primer proyecto MVC. ....	26
Ilustración 7. Agregar aplicación web. ....	27
Ilustración 8. Poner nombre de proyecto. ....	27
Ilustración 9. Tipos de aplicaciones web. ....	28
Ilustración 10. Explorador de la solución.....	28
Ilustración 11. Resultado de aplicación.....	30
Ilustración 12. Añadir controladores. ....	31
Ilustración 13. Tipos de controlador.....	31
Ilustración 14. Contenido de carpeta Controlles.....	31
Ilustración 15. Mensaje de controlador.....	32
Ilustración 16. Bienvenido a MVC.....	33
Ilustración 17. Resultado de ejecutar controlador. ....	33
Ilustración 18. Añadir vistas.....	35
Ilustración 19. Escoger Layout. ....	35
Ilustración 20. Contenido de carpetas View. ....	36
Ilustración 21. Resultado de ejecutar la vista. ....	36
Ilustración 22. Conexión con SQL Server.....	40
Ilustración 23. Agregar conexión.....	40
Ilustración 24. Explorador de servidores. ....	41
Ilustración 25. Resultado de ejecutar consulta. ....	41
Ilustración 26. Agregar modelo de datos.....	42
Ilustración 27. Asistente de modelo de datos. ....	43
Ilustración 28. Conexión con base de datos.....	43
Ilustración 29. Escoger tabla para el modelo de datos.....	44
Ilustración 30. Modelo de datos.....	44
Ilustración 31. Contenido de carpeta Models. ....	45
Ilustración 32. Vistas con Entity Framework 6. ....	46

Ilustración 33. Crear controlador estudiante.....	46
Ilustración 34. Contenido de carpeta Controllers. ....	47
Ilustración 35. Listar estudiante.....	48
Ilustración 36. Detalle estudiante. ....	48
Ilustración 37. Crear estudiante. ....	49
Ilustración 38. Editar estudiante. ....	49
Ilustración 39. Eliminar estudiante. ....	50
Ilustración 40. Explorador de servidores. ....	51
Ilustración 41. Agregar nuevo elemento en modelo de datos.....	51
Ilustración 42. Escoger tabla.....	52
Ilustración 43. Maestro detalle, estudiante. ....	54
Ilustración 44. Aplicación web MVC. ....	55
Ilustración 45. Consola de administración de paquetes.....	56
Ilustración 46. Agregar una nueva vista. ....	60
Ilustración 47. Resultado al ejecutar vista. ....	62
Ilustración 48. Resultado al ejecutar vista. ....	62
Ilustración 49. Resultado al ejecutar vista. ....	64
Ilustración 50. Resultado al ejecutar una vista. ....	65
Ilustración 51. Resultado al ejecutar vista. ....	66
Ilustración 52. Formulario de validación.....	73
Ilustración 53. Administrador de paquetes Nuget. ....	75
Ilustración 54. Archivos CSS. ....	75
Ilustración 55. Archivos JavaScript.....	76
Ilustración 56. Plantilla Bootstrap. ....	76
Ilustración 57. Administrador de paquetes Nuget. ....	77
Ilustración 58. Archivos CSS. ....	78
Ilustración 59. Archivos JavaScript.....	78
Ilustración 60. Características de Windows.....	79
Ilustración 61. Activar y desactivar características de Windows. ....	80
Ilustración 62. Grupo de aplicaciones IIS.....	80
Ilustración 63. Agregar sitio Web.....	81
Ilustración 64. Configuración de nuevo sitio web. ....	82
Ilustración 65. Publicar sitio web .....	82
Ilustración 66. Crear nuevo perfil. ....	83

Ilustración 67. Configurar publicación. ....	83
Ilustración 68. Tipo de compilación .....	84
Ilustración 69. Archivos publicados. ....	84
Ilustración 70. Examinar sitio web .....	85
Ilustración 71. Sitio web publicado. ....	85
Ilustración 72. Trabajo pendiente, iteración 1. ....	109
Ilustración 73. Horas pendientes, iteración 1.....	109
Ilustración 74. Trabajo pendiente, iteración 2. ....	110
Ilustración 75. Horas pendientes, iteración 2.....	110
Ilustración 76. Trabajo pendiente, iteración 3. ....	111
Ilustración 77. Horas pendientes, iteración 3.....	111
Ilustración 78. Trabajo pendiente, iteración 4 .....	112
Ilustración 79. Horas pendientes, iteración 4.....	112
Ilustración 80. Trabajo pendiente, iteración 5. ....	113
Ilustración 81. Horas pendientes, iteración 5.....	113
Ilustración 82. Trabajo pendiente, iteración 6. ....	114
Ilustración 83. Horas pendientes, iteración 6.....	115
Ilustración 84. Trabajo pendiente, iteración 7. ....	115
Ilustración 85. Horas pendientes, iteración 7.....	116
Ilustración 86. Diagrama de negocio. ....	118
Ilustración 87. Ingreso de clientes. ....	120
Ilustración 88. Ingreso de personal de la empresa. ....	121
Ilustración 89. Catálogo de proyectos, actividades y entregables .....	122
Ilustración 90. Bitácora.....	123
Ilustración 91. Diagrama actividades diarias.....	124
Ilustración 92. Reportes y resultados.....	125
Ilustración 93. Master page.....	125
Ilustración 94. Login del sistema.....	126
Ilustración 95. Modelo de datos del sistema.....	127

## ÍNDICE DE TABLAS

Tabla 1 Presupuesto .....	7
Tabla 2. Referencias de Asp.Net MVC. ....	14
Tabla 3. Referencias de Asp.Net MVC. ....	23
Tabla 4. Referencias de Asp.Net MVC. ....	23
Tabla 5. Referencias de Asp.Net MVC. ....	24
Tabla 6. Referencias de Asp.Net MVC. ....	24
Tabla 7. Controladores y enrutamiento. ....	69
Tabla 8. Anotaciones .....	72
Tabla 9. Comparativa MVC Y Web Forms.....	93
Tabla 10. Historias de usuario 1. ....	97
Tabla 11. Historias de usuario 2. ....	97
Tabla 12. Historias de usuario 3. ....	98
Tabla 13. Historias de usuario 4. ....	98
Tabla 14. Historias de usuario 5. ....	98
Tabla 15. Historias de usuario 6 .....	99
Tabla 16 Historias de usuario 7 .....	99
Tabla 17 Historias de usuario 8. ....	99
Tabla 18 Historias de usuario 8. ....	100
Tabla 19 Roles del proyecto. ....	100
Tabla 20 Pila de producto. ....	102
Tabla 21 Definición de requerimientos. ....	103
Tabla 22 Elaboración de diagrama general de negocio. ....	103
Tabla 23 Elaboración del diagrama de procesos. ....	104
Tabla 24 Maquetación de interfaz de usuario. ....	104
Tabla 25 Herramientas de desarrollo. ....	104
Tabla 26 Diseño de la base de datos de la aplicación. ....	104
Tabla 27 Elaboración de diccionario de datos. ....	105
Tabla 28 Desarrollo de catálogo de clientes. ....	105
Tabla 29 Desarrollo de catálogo de personal de la empresa. ....	105
Tabla 30 Planeación de iteración 10. ....	106
Tabla 31 Desarrollo del módulo de auditoria. ....	106
Tabla 32 Desarrollo de módulo de usuarios. ....	106

Tabla 33 Módulo de resultados y reportes.....	106
Tabla 34 Ejecución de pruebas del sistema. ....	107
Tabla 35 Publicación del sistema. ....	107
Tabla 36 Actores.....	119
Tabla 37 Herramientas de desarrollo .....	126
Tabla 38 Tabla parámetros. ....	128
Tabla 39 Tabla clientes.....	128
Tabla 40 Tabla departamentos.....	128
Tabla 41 Tabla cargos.....	129
Tabla 42 Tabla personal.....	129
Tabla 43 Tabla tipos de proyecto.....	129
Tabla 44 Tabla proyectos.....	130
Tabla 45 Tabla entregable.....	130
Tabla 46 Tabla tipo actividades.....	130
Tabla 47 Tabla actividades.....	131
Tabla 48 Tabla bitácora. ....	131
Tabla 49 Tabla auditoria.....	131
Tabla 50 Tabla actividades no planificadas.....	132
Tabla 51 Tabla de módulos.....	132
Tabla 52 Tabla de evidencias.....	132
Tabla 53 Resultados iteración 3.....	133
Tabla 54 Resultados iteración 4.....	134
Tabla 55 Resultados iteración 5.....	135
Tabla 56 Resultados iteración 6.....	135
Tabla 57 Resultados iteración 7.....	136
Tabla 58 Impactos.....	137
Tabla 59 Impacto productivo.....	138
Tabla 60 Impacto organizativo. ....	139
Tabla 61 Impacto económico.....	139
Tabla 62 Impacto innovación. ....	140
Tabla 63 Impacto humano. ....	141

## **RESUMEN.**

En la actualidad el desarrollo de aplicaciones web ha abierto un extenso campo de investigación y estudio de nuevas herramientas, metodologías y arquitecturas que permiten la creación de aplicaciones web que cumplan con estándares de calidad. El propósito de este proyecto de grado es realizar el análisis de uno de los Frameworks de desarrollo web con más proyección a futuro dentro del ecosistema Microsoft, ASP.NET MVC 5; con el fin de explotar y dar a conocer todas sus bondades, generando la documentación necesaria que sirva de guía de estudio para el desarrollo de nuevas aplicaciones web dentro de CREATEC S.A. y en forma general para estudiantes y profesionales interesados en aprender la herramienta; desarrollando el prototipo de un sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A, que pondrá a prueba todas la características de desarrollo del Framework y poder apreciar las ventajas de construir aplicaciones web con ASP.NET MVC 5. En el capítulo uno; se realiza una introducción general del proyecto de tesis en el que se detalla el problema, los objetivos, el alcance, la justificación y presupuesto del proyecto. En el capítulo dos, se realiza el marco teórico de la herramienta donde se construye una serie de tutoriales que sirven de guía de estudio del Framework. En el capítulo tres, se desarrolla el diagnostico de situación actual y marco propositivo del Framework, en el cual se realiza una comparación entre ASP.NET MVC y Web Forms, Frameworks de desarrollo web de Microsoft. En el capítulo cuatro, se expone la documentación y resultados generados en el desarrollo del prototipo de un sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S. En el capítulo cinco, se desarrolla el análisis de impactos, conclusiones y recomendaciones del proyecto.

## **ABSTRACT.**

Today development of web applications has opened an extensive field of research and study of new tools, methodologies and architectures that they facilitate the creation of web applications which perform quality standards. The purpose of this degree work is to do the analysis of one of the web development frameworks with more projection to the future inside to Microsoft ecosystem, ASP.NET MV5; in order to exploit and to give to know all of their benefits, generating the necessary documentation to serve of a study guide to the development of new web applications within CREATEC S.A. and in general to students and professional people interested in learning that tool; developing the prototype of a system to the internal control of the time of use of the human resources of CREATEC S.A., to test all features of development of the framework to can appreciate the advantages to built web applications with ASP.NET MVC 5. In chapter one; it's done a general introduction from the degree work in which it's detailed the problem, the goals, the scope, the justification and budget. In chapter two, it's done the theoretical framework of the tool in where it's built a range of tutorials that they serve like a guide of the study of the framework. In chapter three, it's developed the diagnosis of the real situation and the purposeful framework of that framework, in which it's done a comparison between ASP.NET MVC and Web Forms, thereof that are the two frameworks of development most important from Microsoft. In chapter four; it's exposed the documentation and the generated results with the development of the prototype of the system for the internal control of the time of use of the human resources from CREATEC S.A. following the format of SCRUM methodology. In chapter five, it's developed the impact analysis, conclusions and recommendations of the project.

# CAPÍTULO I.

## 1. Introducción.

A continuación se presenta una introducción general del proyecto de tesis en el que se detalla el problema, los objetivos, el alcance, la justificación y presupuesto del proyecto.

### 1.1. Problema.

El problema del proyecto de tesis consta del análisis de antecedentes, situación actual, planteamiento del problema y prospectiva.

#### 1.1.1. Antecedentes.

El desarrollo de aplicaciones web y la reciente tendencia de acceso a las mismas a través de dispositivos móviles abren un extenso campo de investigación de nuevas herramientas, metodologías, arquitecturas y buenas prácticas de programación que garanticen aplicaciones que cumplan con estándares de calidad.

Antes de la aparición de las aplicaciones web existía otro tipo de aplicaciones llamadas aplicaciones cliente-servidor las cuales su desarrollo e implementación eran muy costosos puesto que debían ser instaladas por separado en cada computador de los usuarios los cuales realizaban peticiones a un servidor, una actualización o mejora de los sistemas significaba normalmente una mejora de los programas instalados en los clientes, esto originaba un coste elevado en soporte técnico lo cual disminuía la productividad.

Con la aparición de la nueva generación de aplicaciones se solucionaron muchos de los problemas presentados en las anteriores, permitiendo disfrutar de ventajas como:

- Ahorro de tiempo de desarrollo e implementación.

- Solución problemas de compatibilidad.
- Actualizaciones de software inmediatas.
- Portabilidad.
- Multiplataforma.
- Costes bajos en soporte técnico.

Es por eso la importancia de un análisis de estas nuevas herramientas como el caso del <sup>1</sup>Framework ASP.NET MVC 5 en el desarrollo de aplicaciones web para poder llevar un mejor control sobre partes individuales de la aplicación, lo que facilita su desarrollo, modificación y prueba, además que permita explotar todas las ventajas del desarrollo web.

### **1.1.2. Situación actual.**

Actualmente ASP.NET MVC 5 no es muy difundido en el mercado, esto ocasiona que programadores prefieran desarrollar aplicaciones de la forma tradicional con ASP.NET <sup>2</sup>WebForms. Además cabe recalcar que estudiantes y profesionales en informática se inclinan por usar software libre para el desarrollo de sus aplicaciones, todas estas causas ocasionan un desconocimiento de las bondades de este Framework, escasos de programadores y pocas aplicaciones en el mercado.

Hoy en día se puede encontrar información disponible en la página oficial de ASP.NET MVC, esta información es una guía de estudio que permite explotar todas las bondades del Framework.

---

<sup>1</sup> **Framework:** Conjunto estandarizado de metodologías y prácticas utilizadas en el desarrollo de software.

<sup>2</sup> **Asp .Net Web Forms:** Framework desarrollado por Microsoft utilizado para la construcción de aplicaciones web.

### **1.1.3. Planteamiento del problema.**

La falta de conocimiento y difusión en el medio ocasiona un desconocimiento de las características y bondades de ASP.NET MVC 5 en estudiantes y profesionales de la informática lo que reduce la producción de aplicaciones ASP.NET siguiendo el patrón MVC.

### **1.1.4. Prospectiva.**

Con el análisis a realizarse se generará la documentación necesaria para estudiantes y profesionales interesados en estudiar este Framework, lo que disminuirá el desconocimiento del mismo, se evaluará las bondades del Framework con el fin desarrollar nuevas aplicaciones web dentro de CREATEC S.A aplicando los conocimientos obtenidos en el análisis, esto generará un aumento de aplicaciones ASP.NET MVC 5 en el mercado y garantizará sistemas de calidad.

Con el desarrollo del sistema se pretende aplicar los conocimientos obtenidos, demostrar las ventajas de desarrollar aplicaciones con un patrón de diseño MVC, y explotar las bondades Framework ASP.NET MVC 5.

## **1.2. Objetivos.**

A continuación se presentan los objetivos generales y específicos del proyecto de tesis.

### **1.2.1. Objetivo general.**

Analizar el Framework ASP.NET MVC 5 para la implementación del prototipo de un sistema de control interno de uso del tiempo de los recursos humanos de <sup>3</sup>CREATEC S.A. utilizando nuevas técnicas de desarrollo web.

---

<sup>3</sup> CREATEC S.A: Empresa ecuatoriana dedicada al desarrollo de software.

### **1.2.2. Objetivos específicos.**

- Elaborar el marco propositivo del uso de la herramienta.
- Diagnosticar la situación actual del uso de la herramienta.
- Construir la documentación técnica que servirá de guía de estudio para interesados en aprender el Framework, el manual constará de teoría y ejemplos prácticos.
- Desarrollar el prototipo de un sistema para el control interno de uso del tiempo de recursos humanos de CREATEC S.A.
- Realizar un análisis de impactos de uso de la herramienta.

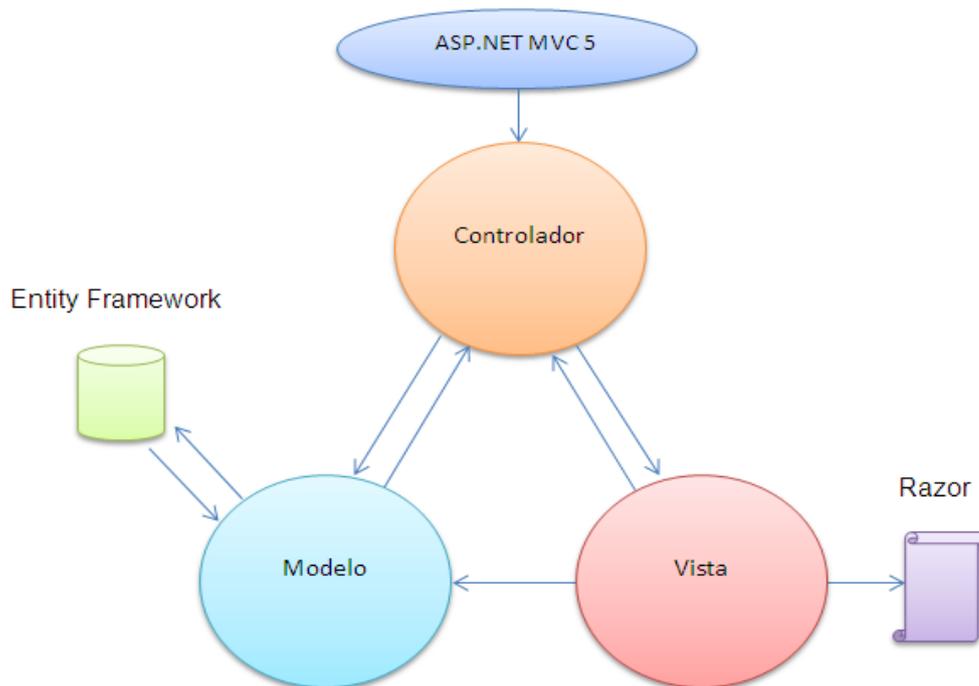
### **1.3. Alcance.**

El análisis cubrirá el estudio del Framework ASP.NET MVC en su quinta versión para la construcción de aplicaciones web que comprenderán de tres funciones: modelo, vista, controlador; Además se estudiará su integración con características de ASP.NET existentes.

Se investigará el motor de vista Razor utilizado por este Framework que básicamente son HTML escrito con pequeñas porciones de código de servidor destinadas a terminar generando la interfaz de usuario de la aplicación web.

Se estudiará la integración de ASP.NET MVC 5 con Entity Framework que es un mapeador objeto relacional que permite trabajar con datos relacionales.

Se investigará la utilización de JQuery y JQuery Mobile en ASP.NET MVC 5 para programación de aplicaciones web-móviles.



**Ilustración 1.** Alcance del proyecto.

Fuente: (Gómez, 2014)

Para aplicar los conocimientos obtenidos en el análisis se desarrollará el prototipo de un sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A. para posteriormente realizar un análisis de impactos del uso del Framework en aplicaciones web.

Es sistema tendrá la capacidad de:

- Planificar las actividades y tiempos de ejecución en proyectos de software realizados por CREATEC S.A.
- Llevar el control del tiempo invertido por la persona en determinada actividad o proyecto de la empresa.
- Al final del mes totalizar tiempos de cada proyecto como también de los empleados.
- Llevar el registro de evidencias al ejecutar cada actividad de un proyecto.
- Llevar el control de entregables de cada actividad en un determinado proyecto.

- Llevar el registro de las actividades diarias realizadas por los empleados de la empresa que no son planificadas en un proyecto

El sistema tendrá los siguientes módulos.

- Administración de usuarios.
- Catálogo de empleados: llevará un registro de los empleados los cuales son encargados de trabajar en proyectos y actividades de la empresa.
- Catálogo de proyectos y actividades: llevará el registro de los proyectos y actividades que se ejecutan en la empresa.
- Bitácora: llevará el registro de tiempos empleados en proyectos y actividades de la empresa.
- Reportes y resultados.

Para la construcción del prototipo se utilizará la metodología de software <sup>4</sup>SCRUM, esta es basada en procesos en el que se aplican un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, para obtener el mejor resultado posible en un proyecto, la misma garantizará el ordenamiento de documentación y desarrollo del prototipo para obtener el resultado esperado.

#### **1.4. Justificación.**

El análisis del Framework ASP.NET MVC 5 como proyecto de tesis permitirá brindar documentación suficiente que servirá de guía de estudio para estudiantes y profesionales en el área de la informática que ampliarán sus conocimientos de este Framework, explotaran todas sus bondades y garantizarán la construcción de aplicaciones de calidad.

---

<sup>4</sup> **Scrum:** modelo de desarrollo de software ágil.

Motivará a desarrolladores de software a utilizar este Framework y nuevas herramientas en la construcción de sus sistemas lo que ocasionará una producción mayor de aplicaciones con patrones de diseño modelo, vista, controlador.

Incentivará a estudiantes a seguir realizando investigaciones de nuevas metodologías y Framework de desarrollo de aplicaciones, en especial en el área de programación web ya que es la tendencia de los últimos tiempos.

### 1.5. Presupuesto.

**Tabla 1** Presupuesto.

DESCRIPCIÓN	COSTO ESTIMADO	COSTO REAL
<b>HARDWARE</b>		
Equipo de computación	800	0
Servidor	5000	0
Equipo móvil	500	500
<b>SOFTWARE</b>		
Sql Server Express Edition	0	0
Visual Estudio 2013 Express para Web	0	0
<b>RECURSO HUMANO</b>	2544	2544
<b>MATERIALES</b>		
Útiles de oficina	100	100
Impresión de documentos	100	100
Otros	50	50
<b>BIBLIOGRAFIA</b>		
Libros y revistas	200	200
Documentos del internet	100	100
Movilización	50	50
<b>SUBTOTAL</b>	9444	3644
10% de imprevistos	944.40	364.40
<b>TOTAL</b>	10388,40	4008.40

Fuente: (Gómez, 2014)

## CAPÍTULO II

### 2. Marco teórico.

Se realiza el análisis de la herramienta donde se construye una serie de ejemplos y prácticas que sirven de guía de estudio del Framework ASP.NET MVC.

#### 2.1. Patrón de diseño MVC.

*“MVC (Modelo-Vista-Controlador) es un patrón de diseño de arquitectura de software que separa la lógica de negocio de la interfaz de usuario, facilita el desarrollo por separado de ambas capas e incrementa reutilización y flexibilidad.”* (Mestras, 2012)

MVC se encuentra definido por el Modelo (lógica de negocio), la vista (interfaz con el usuario) y el controlador (controlador de flujo de trabajo de la aplicación), esto permite identificar las siguientes 3 capas:

##### 2.1.1. Modelo.

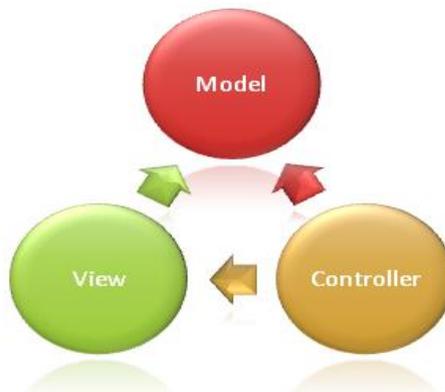
Contiene la lógica de negocio de la aplicación la misma que puede ser prevista en cualquier almacén de datos, encapsula el estado de la aplicación, es independiente del controlador y la vista.

##### 2.1.2. Vista.

Contiene la presentación de la capa de modelado, encargada de la visualización de la interfaz de usuario de la aplicación, puede acceder al modelo pero nunca cambiar su estado, puede ser notificado cuando hay un cambio de estado en el modelo.

### 2.1.3. Controlador.

Es la capa intermedia que interactúa entre la capa de modelo y vista.



**Ilustración 2.** Modelo MVC.

Fuente: (Asp.Net, 2013)

*“Las capas modelo, la vista y el controlador deben comunicarse de una manera estable entre sí, para que sea coherente con las iteraciones que el usuario realizará. Como es lógico la comunicación entre la vista y el controlador es bastante básica pues están diseñados para operar juntos.” (MVC TIW, 2012)*

### 2.1.4. Flujo de control.

- El usuario realiza una petición desde un navegador.
- El controlador recibe la petición.
- El controlador notifica al modelo la petición del usuario.
- Se genera una nueva vista.
- La vista toma los datos del modelo.
- La interfaz de usuario espera otra interacción con el usuario, que comenzara un nuevo ciclo.

### 2.1.5. MVC en aplicaciones Web.

*“Aunque originalmente MVC fue desarrollado para aplicaciones cliente-servidor, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los*

principales lenguajes de programación. Se han desarrollado multitud de Frameworks, comerciales y no comerciales, que implementan este patrón, estos Frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.” (Asp.Net, 2013)

“Los primeros Frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero en el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier formulario al controlador y después recibe de la vista una página completa y actualizada, tanto el modelo como el controlador están completamente alojados en el servidor.” (Asp.Net, 2013)

- Vista: representa la página HTML de la aplicación.
- Controlador: código que obtiene datos dinámicamente y genera el contenido HTML.
- Modelo: representa la información almacenada en cualquier almacén de datos.

## 2.2. ASP.NET MVC.



**Ilustración 3.** Asp.Net MVC

**Fuente:** (Seekdotnet, 2012)

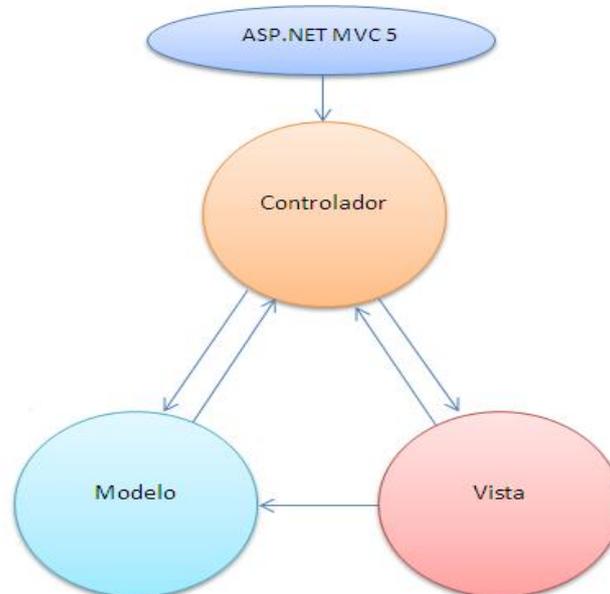
“Es un Framework de desarrollo web que implementa el patrón modelo-vista-controlador el mismo está basado en <sup>5</sup>ASP.NET, permite a los desarrolladores de software construir aplicaciones web utilizando el patrón de diseño MVC (modelo, vista, controlador)” (Asp.Net, 2013)

---

<sup>5</sup> **ASP.NET:** Framework de aplicaciones web creado por Microsoft.

*“El Framework de ASP.NET MVC proporciona una alternativa al modelo de formularios Web Forms de ASP.NET para crear aplicaciones web basadas en MVC”.* (Microsoft, Developer Network, 2014)

*“El marco de ASP.NET MVC es un marco de presentación de peso ligero, altamente comprobable que al igual que con las aplicaciones basadas en formularios web se integra con las características de ASP.NET existentes, como páginas maestras y la autenticación basada en membresía.”* (Asp.Net, 2013)



**Ilustración 4.** ASP. NET MVC.

**Fuente:** (Gómez, 2014)

*“Uno de los principios fundamentales del Framework es “no repetir código dos veces”. ASP.NET MVC incentiva a escribir funcionalidades y comportamiento una sola vez, para reflejarse en todas las capas de la aplicación, esto reduce la cantidad de código repetido, la aplicación se vuelve menos propensa a errores y más fácil de mantener”.* (Asp.Net, 2013)

### 2.2.1. Características del marco ASP.NET MVC.

El marco de ASP.NET MVC proporciona las siguientes características:

- La separación de tareas de la aplicación (lógica de control, lógica de negocio y la lógica de interfaz de usuario)
- La capacidad de prueba y desarrollo basado en (<sup>6</sup>TDD) de forma predeterminada.
- Los componentes del marco de ASP.NET MVC están diseñados de modo que puedan ser reemplazados o personalizados fácilmente.
- Un poderoso componente <sup>7</sup>URL de mapeo que le permite crear aplicaciones que tienen URLs comprensibles y realizar búsquedas.
- El marco MVC admite el uso de la ASP.NET existente: <sup>8</sup>aspx, <sup>9</sup>ascx y archivos de marcado Maestro como plantillas de vista.

### 2.2.2. Cuando desarrollar con ASP.NET MVC.

*“El Framework de MVC no pretende sustituir el modelo de formularios Web Forms, se puede utilizar cualquiera de los Framework para las aplicaciones Web. En el caso de que se cuente con aplicaciones basadas en formularios web existentes no se recomienda migrar a ASP.NET MVC ya que estas siguen trabajando exactamente como siempre lo han hecho.” (Microsoft, Developer Network, 2014)*

---

<sup>6</sup> **TDD:** Práctica de programación para realizar pruebas unitarias.

<sup>7</sup> **URL:** localizador de recursos uniforme.

<sup>8</sup> **Aspx:** extensión de archivo que contiene una página web.

<sup>9</sup> **Ascx:** control de usuario creado dentro de una página web con extensión aspx.

Antes de decidirse a usar el Framework MVC o el modelo de formularios Web Forms, tener en cuenta las ventajas que presentan estos dos Frameworks lo cual es analizado en el capítulo 3.

### **2.2.3. Ventajas de desarrollar con ASP.NET MVC.**

El marco de ASP.NET MVC ofrece las siguientes ventajas:

- Divide con facilidad una aplicación en el modelo, la vista y el controlador.
- Brinda un control completo sobre el comportamiento de una aplicación.
- Procesa solicitudes de aplicaciones Web a través de un solo controlador.
- Permite el diseño de una aplicación compatible con una rica infraestructura de enrutamiento.
- Proporciona un mejor soporte para el desarrollo basado en pruebas (TDD).
- Funciona bien para las aplicaciones web que son apoyados por grandes equipos de desarrolladores y diseñadores web que necesitan un alto grado de control sobre el comportamiento de la aplicación.

### **2.2.4. Referencias de ASP.NET MVC.**

Información de la tabla obtenida de la página oficial MSDN, para más información visitar la página:

([http://msdn.microsoft.com/es-es/library/system.web.mvc\(v=vs.108\).aspx](http://msdn.microsoft.com/es-es/library/system.web.mvc(v=vs.108).aspx))

### a. System.Web.Mvc

System.Web.Mvc es el principal <sup>10</sup> namespace del Framework MVC de ASP.NET, contiene clases que representan controladores, generadores de controladores, resultados de acciones, vistas y modelados.

**Tabla 2.** Referencias de Asp.Net MVC.

Clases	Descripción
AcceptVerbsAttribute	Representa los verbos <sup>11</sup> HTTP que responderá un método de acción.
ActionDescriptor	Proporciona información de un método, tal como: nombre, controlador, parámetros, atributos y filtros.
ActionExecutedContext	Proporciona el contexto para el método ActionExecuted.
ActionExecutingContext	Proporciona el contexto para el método ActionExecuting.
ActionFilterAttribute	Representa la clase de los atributos de filtro
ActionMethodSelectorAttribute	Representa un atributo que se usa para influir en la selección de un método de acción.
ActionNameAttribute	Representa un atributo que retorna el nombre de una acción.
ActionNameSelectorAttribute	Representa un atributo que selecciona un método de acción.
ActionResult	Encapsula el resultado de un método de acción.
AdditionalMetadataAttribute	Proporciona una clase que admite metadatos adicionales.
AjaxHelper	Representa la compatibilidad de HTML con AJAX
AjaxHelper<TModel>	Representa la compatibilidad de HTML con <sup>12</sup> AJAX dentro de una vista.
AjaxRequestExtensions	Clase que extiende la clase HttpRequestBase .
AllowAnonymousAttribute	Atributo que marca controladores y acciones

<sup>10</sup> **Namespace:** Ámbito que contiene un conjunto de clases y objetos relacionados.

<sup>11</sup> **HTTP:** Protocolo de transferencia de texto.

<sup>12</sup> **AJAX:** técnica de desarrollo para desarrollar aplicaciones web interactivas.

Clases	Descripción
AllowHtmlAttribute	Incluye formato HTML en el enlace del modelo.
AreaRegistration	Registra una o más áreas en una aplicación ASP.NET MVC.
AreaRegistrationContext	Encapsula la información para registrar un área en una aplicación de ASP.NET MVC.
AssociatedMetadataProvider	Proporciona una clase para un proveedor de metadatos
AssociatedValidatorProvider	Clases que implementan un proveedor de validación.
AsyncController	Compatibilidad con versiones anteriores a ASP.NET MVC 3.
AsyncTimeoutAttribute	Atributo que establece el valor de tiempo de espera, para un método asincrónico.
AuthorizationContext	Encapsula la información para usar un atributo AuthorizeAttribute.
AuthorizeAttribute	Atributo que se usa para restringir el acceso a un método de acción.
BindAttribute	Atributo que proporciona información de cómo debe producirse un enlace del modelo a un parámetro.
BuildManagerCompiledView	Clase base recopila la clase BuildManagerantes.
BuildManagerViewEngine	Clase base para los motores de vista.
ByteArrayModelBinder	Explorador a una matriz de bytes.
CachedAssociatedMetadata- Provider	Proporciona una clase abstracta para implementar un proveedor de metadatos almacenado en caché.
CachedDataAnnotationsMetadataAttributes	Contenedor para almacenar las anotaciones en caché.
CachedDataAnnotationsModelMetadata	Proporciona un contenedor para almacenar anotaciones en caché.
CachedDataAnnotationsModelMetadataProvide r	Proveedor de metadatos del modelo almacenado en caché.
CachedModelMetadata<TPrototypeCache>	Contenedor de metadatos almacenados en caché.
CancellationTokenModelBinder	Propaga notificaciones.
ChildActionOnlyAttribute	Atributo que indica la llamada a un método de acción solo como acción secundaria.
ChildActionValueProvider	Proveedor de valores para las acciones secundarias.
ChildActionValueProviderFactory	Representa un generador para crear objetos de proveedor para acciones secundarias.
ClientDataTypeModelValidatorProvider	Devuelve los validadores de modelo de tipo de datos.
CompareAttribute	Compara dos propiedades de un modelo.
ContentResult	Contenido definido por el usuario que es el resultado de un método de acción.

Clases	Descripción
Controller	Responden a las solicitudes HTTP.
ControllerActionInvoker	Invoca los métodos de acción de un controlador.
ControllerBase	Clase base para todos los controladores MVC.
ControllerBuilder	Compila un controlador dinámicamente.
ControllerContext	Encapsula información sobre una solicitud HTTP.
ControllerDescriptor	Encapsula la descripción de un controlador.
ControllerInstanceFilterProvider	Agrega un controlador a la instancia de FilterProviderCollection.
CustomModelBinderAttribute	Invoca un enlazador de modelos personalizado.
DataAnnotationsModelMetadata	Proporciona un contenedor para los metadatos comunes, de un modelo de datos.
DataAnnotationsModelMetadataProvider	Proveedor de metadatos del modelo predeterminado para ASP.NET MVC.
DataAnnotationsModelValidator	Validador de modelo.
DataAnnotationsModelValidator<TAttribute>	Validador de modelo para un tipo de validación especificado.
DataAnnotationsModelValidatorProvider	Validación predeterminada para ASP.NET MVC.
DataErrorInfoModelValidatorProvider	Contenedor de validadores de modelo de información de error.
DefaultControllerFactory	Generador de controladores.
DefaultModelBinder	Implementación concreta de un enlazador de modelos.
DefaultViewLocationCache	Memoria caché para ubicaciones de vista.
DependencyResolver	Soluciones de dependencia que implementan IDependencyResolver.
DependencyResolverExtensions	Proporciona una implementación con seguridad de tipos de GetService y GetServices.
DictionaryValueProvider<TValue>	Proveedores de valores que proceden de una colección que implementa la interfaz IDictionary.
EmptyModelMetadataProvider	Proveedor de metadatos vacío.
EmptyModelValidatorProvider	Proveedor de validación vacío.
EmptyResult	Resultado de controlador que no devuelve nada.
ExceptionContext	Proporciona el contexto para usar la clase HandleErrorAttribute.
ExpressionHelper	Clase auxiliar para obtener el nombre del modelo de una expresión.
FieldValidationMetadata	Metadatos de validación de campos del lado cliente.
FileContentResult	Envía el contenido de un archivo binario a la respuesta.
FilePathResult	Envía el contenido de un archivo a la respuesta.

Clases	Descripción
FileResult	Envía contenido de archivo binario a la respuesta.
FileStreamResult	Envía el contenido binario a la respuesta mediante un Stream.
Filter	Referencia a la implementación de una o más de las interfaces de filtro.
FilterAttribute	Clase de los atributos de acción y filtro de resultados.
FilterAttributeFilterProvider	Proveedor de filtros para los atributos de filtro.
FilterInfo	Encapsula filtros de acción disponibles.
FilterProviderCollection	Colección de proveedores de filtros para la aplicación.
FilterProviders	Punto de registro para los filtros.
FormCollection	Proveedores de valor de formulario para la aplicación.
FormContext	Valida y procesa datos de entrada de un formulario HTML.
FormValueProvider	Proveedor para los valores de formulario.
FormValueProviderFactory	Crea una nueva instancia de un objeto de proveedor de formulario.
GlobalFilterCollection	Contiene todos los filtros globales.
GlobalFilters	Colección de filtros globales.
HandleErrorAttribute	Controla una excepción producida por un método de acción.
HandleErrorInfo	Administra un error producido por un método de acción.
HiddenInputAttribute	Indica si una propiedad de campo se presenta como un elemento input oculto.
HtmlHelper	Compatibilidad para representar los controles HTML en una vista.
HtmlHelper<TModel>	Compatibilidad de controles HTML en una vista fuertemente tipada.
HttpDeleteAttribute	Método de administración de solicitudes DELETE de HTTP.
HttpFileCollectionValueProvider	Valores que proceden de una colección de archivos HTTP.
HttpFileCollectionValueProviderFactory	Proveedor de valores de colección de archivos HTTP.
HttpGetAttribute	Restringe un método de acción de forma que el método administre solamente las solicitudes <sup>13</sup> HTTP GET.

---

<sup>13</sup> **HTTP GET:** Método que pide datos de un recurso especificado.

Clases	Descripción
HttpHeadAttribute	Solicitud HTTP debe ser el método <sup>14</sup> HTTP HEAD.
HttpNotFoundResult	Indica que no se encontró el recurso solicitado.
HttpOptionsAttribute	Restringe un método de forma que el método administre solamente las solicitudes HTTP OPTIONS.
HttpPatchAttribute	Restringe la acción de forma que el método administre solamente las solicitudes HTTP PATH.
HttpPostAttribute	Restringe la acción de forma que el método administre solamente las solicitudes HTTP POST.
HttpPostedFileBaseModelBinder	Enlaza un modelo a un archivo expuesto.
HttpPutAttribute	Restringe la acción de forma que el método administre solamente las solicitudes <sup>15</sup> HTTP PUT.
HttpRequestExtensions	Contiene los valores HTTP enviados mediante una solicitud web.
HttpStatusCodeResult	Devuelve un resultado de la acción con un código de estado de respuesta HTTP.
HttpUnauthorizedResult	Resultado de una solicitud HTTP no autorizada.
JavaScriptResult	Envía el código JavaScript a la respuesta.
JsonResult	Envía contenido con formato JSON a la respuesta.
JsonValueProviderFactory	Métodos de acción para enviar y recibir el texto con formato <sup>16</sup> JSON.
LinqBinaryModelBinder	Asigna una solicitud de explorador a un objeto LINQ Binary.
ModelBinderAttribute	Asocia un tipo de modelo a un tipo de generador de modelos.
ModelBinderDictionary	Contiene enlazadores de modelos para la aplicación.
ModelBinderProviderCollection	Proveedores de enlazadores de modelos.
ModelBinders	Enlazadores de modelos de la aplicación.
ModelBindingContext	Enlazador de modelos.
ModelError	Error durante el enlace de modelos.
ModelErrorCollection	Colección de instancias de ModelError.
ModelMetadata	Contenedor para los metadatos comunes.

---

<sup>14</sup> **HTTP HEAD:** Método que pide datos generalmente de los encabezados de respuesta.

<sup>15</sup> **HTTP PUT:** Método que crea o actualiza información.

<sup>16</sup> **JSON:** Framework de intercambio de datos entre el cliente y el servidor.

Clases	Descripción
ModelMetadataProvider	Proveedor de metadatos personalizado.
ModelMetadataProviders	Contenedor de instancia actual de ModelMetadataProvider.
ModelState	Encapsula el estado del enlace del modelo a una propiedad de un argumento de acción-método o al propio argumento.
ModelStateDictionary	Representa el estado de un intento de enlazar un formulario expuesto a un método de acción.
ModelValidationResult	Contenedor para un resultado de la validación.
ModelValidator	Implementa lógica de validación.
ModelValidatorProvider	Lista de validadores para un modelo.
ModelValidatorProviderCollection	Lista de proveedores de validación.
ModelValidatorProviders	Proveedor de validación actual.
MultiSelectList	Lista de elementos de los cuales los usuarios pueden seleccionar más de un elemento.
MvcFilter	Proporciona una clase de metadatos que contiene referencias de interfaces de filtro.
MvcHandler	Selecciona el controlador que controlará una solicitud HTTP.
MvcHtmlString	Cadena codificada en HTML que no debe volver a codificarse.
MvcHttpHandler	Comprueba y procesa una solicitud HTTP.
MvcRouteHandler	Implementa la interfaz IHttpHandler y le pasa el contexto de solicitud.
MvcWebRazorHostFactory	Crea instancias de archivos MvcWebPageRazorHost.
NameValueCollectionExtensions	Extiende un objeto para que la colección se pueda copiar en un diccionario especificado.
NameValueCollectionValueProvider	Proveedores de valores que proceden de un objeto NameValueCollection.
NoAsyncTimeoutAttribute	Contenedor de conveniencia para el atributo AsyncTimeoutAttribute.
NonActionAttribute	Indica que un método de controlador no es un método de acción.
OutputCacheAttribute	Marcar una acción cuyo resultado estará almacenado en memoria caché.
ParameterBindingInfo	Enlazar parámetros de acciones a un modelo de datos.
ParameterDescriptor	Contiene información que describe un parámetro.
PartialViewResult	Envía una vista parcial a la respuesta.

Clases	Descripción
PreApplicationStartCode	Punto de registro para el código de inicio anterior a la aplicación de ASP.NET Razor.
QueryStringValueProvider	Proveedor de valores para las cadenas de consulta de los objetos NameValueCollection.
QueryStringValueProviderFactory	Crear una nueva instancia de un objeto de proveedor de valores de cadena de consulta.
RangeAttributeAdapter	Adaptador para el atributo RangeAttribute.
RazorView	Crear vistas que tienen sintaxis Razor.
RazorViewEngine	Motor de vista que se emplea para representar una página web que usa la sintaxis Razor.
RedirectResult	Procesamiento de acciones de aplicación al redirigir a un identificador <sup>17</sup> URI.
RedirectToRouteResult	Realiza una redirección mediante el diccionario de valores de ruta especificado.
ReflectedActionDescriptor	Información que describe un método de acción reflejado.
ReflectedControllerDescriptor	Información que describe un controlador reflejado.
ReflectedParameterDescriptor	Información que describe un parámetro de método de acción reflejado.
RegularExpressionAttributeAdapter	Selecciona un adaptador para el atributo RegularExpressionAttribute.
RemoteAttribute	Validador remoto del complemento de validación jQuery.
RequiredAttributeAdapter	Adaptador para el atributo RequiredAttributeAttribute.
RequireHttpsAttribute	Obliga a reenviar una solicitud HTTP no segura sobre HTTPS.
ResultExecutedContext	Contexto para el método OnResultExecuted de la clase ActionFilterAttribute.
ResultExecutingContext	Contexto para el método OnResultExecuting.
RouteCollectionExtensions	Objeto RouteCollection para el enrutamiento MVC.
RouteDataValueProvider	Proveedor de valores para los datos de ruta contenidos en un objeto que implementa la interfaz IDictionary.
RouteDataValueProviderFactory	Crear objetos de proveedor de valores de datos de ruta.
SelectList	Lista que permite a los usuarios seleccionar un elemento.

---

<sup>17</sup> **URI:** conjunto de caracteres que identifica los recursos de una red.

Clases	Descripción
SelectListItem	Elemento seleccionado en una instancia de la clase SelectList.
SessionStateAttribute	Estado de sesión del controlador.
SessionStateTempDataProvider	Datos del estado de sesión al objeto TempDataDictionary actual.
StringLengthAttributeAdapter	Adaptador para el atributo StringLengthAttribute.
TempDataDictionary	Conjunto de datos que persiste de una solicitud a la siguiente.
TemplateInfo	Encapsula la información sobre el contexto de la plantilla actual.
UrlHelper	Crea direcciones URL para ASP.NET MVC dentro de una aplicación.
UrlParameter	Representa un parámetro opcional que usa la clase MvcHandler durante el enrutamiento.
ValidatableObjectAdapter	Adaptador de objetos que se puede validar.
ValidateAntiForgeryTokenAttribute	Impide la falsificación de una solicitud.
ValidateInputAttribute	Marca métodos de acción cuya entrada se debe validar.
ValueProviderCollection	Colección de objetos de proveedor de valores para la aplicación.
ValueProviderDictionary	Diccionario de proveedores de valores para la aplicación.
ValueProviderFactories	Contenedor para objetos de generador de proveedores de valores.
ValueProviderFactory	Crea objetos de proveedor de valores.
ValueProviderFactoryCollection	Colección de generadores de proveedores de valores para la aplicación.
ValueProviderResult	Representa el resultado de enlazar un valor.
ViewContext	Encapsula información relacionada con la representación de una vista.
ViewDataDictionary	Pasa datos entre un controlador y una vista.
ViewDataDictionary<TModel>	Pasa datos fuertemente tipados entre un controlador y una vista.
ViewDataInfo	Encapsula la información sobre el contenido de la plantilla actual que se usa para desarrollar las plantillas y sobre las aplicaciones auxiliares HTML que interactúan con las plantillas.
ViewEngineCollection	Colección de motores de vista disponibles para la aplicación.

Clases	Descripción
ViewEngineResult	Resultado de buscar un motor de vista.
ViewEngines	Colección de motores de vista disponibles para la aplicación.
ViewMasterPage	Información necesaria para compilar una página de vista maestra.
ViewMasterPage<TModel>	Crear una página de vista fuertemente tipada.
ViewPage	Representa una vista como una página de formularios Web Forms.
ViewPage<TModel>	Presenta una vista fuertemente tipada como página de formularios Web Forms.
ViewResult	Representa una vista utilizando una instancia devuelta por un objeto IViewEngine.
ViewResultBase	Proporcionar el modelo a la vista y, luego, representar la vista para la respuesta.
ViewStartPage	Implementa una página de inicio de vista.
ViewTemplateUserControl	Contenedor para los objetos TemplateInfo.
ViewTemplateUserControl<TModel>	Contenedor para los objetos TemplateInfo.
ViewType	Tipo de una vista.
ViewUserControl	Información que se necesita para compilar un control de usuario.
ViewUserControl<TModel>	Información necesaria para crear un control de usuario fuertemente tipado.
VirtualPathProviderViewEngine	Clase base abstracta de la interfaz IViewEngine.
WebFormView	Información que se necesita para compilar una página de formularios Web Forms en ASP.NET MVC.
WebFormViewEngine	Representa un motor de vista que se usa para representar una página de formularios Web Forms para la respuesta.
WebViewPage	Vista con sintaxis ASP.NET Razor.

Fuente: (Developer\_Network, Referencias de ASP.NET MVC, 2014)

## b. System.Web.Mvc.Ajax

“El *System.Web.Mvc.Ajax* contiene clases que permiten escribir script Ajax en una aplicación ASP.NET MVC”. (Microsoft, Developer Network, 2014)

**Tabla 3.** Referencias de Asp.Net MVC.

Clase	Descripción
AjaxExtensions	Soporte para AJAX de ASP.NET.
AjaxOptions	Opciones para ejecutar los scripts Ajax.

Fuente: (Developer\_Network, Referencias de ASP.NET MVC, 2014)

### c. System.Web.Mvc.Async

“El *System.Web.Mvc.Async* contiene clases que permiten acciones asíncronas en una aplicación ASP.NET MVC”. (Microsoft, Developer Network, 2014)

**Tabla 4.** Referencias de Asp.Net MVC.

Clase	Descripción
AsyncActionDescriptor	Proporciona información de una acción asíncrona, tal como su nombre, controlador, parámetros, atributos y filtros.
AsyncControllerActionInvoker	Invoca los métodos de acción de un controlador asíncrono.
AsyncManager	Operaciones asíncronas para la clase AsyncController.
OperationCounter	Contenedor que mantiene un recuento de operaciones asíncronas pendientes.
ReflectedAsyncActionDescriptor	Información sobre una acción asíncrona: nombre, controlador, parámetros, atributos y filtros.
ReflectedAsyncControllerDescriptor	Información que describe un controlador asíncrono: nombre, su tipo y sus acciones.
SynchronousOperationException	Excepción que se produjo durante el procesamiento sincrónico de una solicitud HTTP.

Fuente: (Developer\_Network, Referencias de ASP.NET MVC, 2014)

### d. System.Web.Html.

“El namespace *System.Web.Html* contiene clases que ayudan a representar los controles HTML en una aplicación MVC.” (Microsoft, Developer Network, 2014)

**Tabla 5.** Referencias de Asp.Net MVC.

Clases	Descripción
ChildActionExtensions	Compatibilidad para llamar a los métodos de acción secundaria.
DisplayExtensions	Compatibilidad para presentar los valores de objeto como HTML.
DisplayNameExtensions	Mecanismo que permite obtener nombres para mostrar.
DisplayTextExtensions	Medio para representar los valores de objeto como HTML.
EditorExtensions	Compatibilidad para el elemento input HTML en una aplicación.
FormExtensions	Compatibilidad para HTML en una aplicación.
InputExtensióons	Compatibilidad para los controles de entrada HTML en una aplicación.
LabelExtensions	Compatibilidad para el elemento label de HTML en una vista de ASP.NET MVC.
LinkExtensions	Compatibilidad para los vínculos HTML en una aplicación.
MvcForm	Elemento form HTML en una vista de MVC.
NameExtensions	Identificador HTML y los atributos de nombre de la cadena HtmlHelper.
PartialExtensions	Vista parcial como una cadena codificada en HTML.
RenderPartialExtensions	Compatibilidad para representar una vista parcial.
SelectExtensions	Compatibilidad para realizar selecciones en una lista.
TextAreaExtensions	Compatibilidad para controles textarea HTML.
ValidationExtensions	Compatibilidad para validar la entrada de un formulario HTML.
ValueExtensions	Mecanismo para crear marcado HTML personalizado compatible con enlazadores y plantillas de modelo de ASP.NET MVC.

Fuente: (Developer\_Network, Referencias de ASP.NET MVC, 2014)

#### e. System.Web.Mvc.Razor.

“El namespace *System.Web.Mvc.Razor* contiene clases que permiten trabajar con vistas de *ASP.NET Razor*”. (Microsoft, Developer Network, 2014)

**Tabla 6.** Referencias de Asp.Net MVC.

Clase	Descripción
MvcCSharpRazorCodeParser	Compila las vistas de ASP.NET Razor en clases.
MvcVBRazorCodeParser	Agrega compatibilidad con la palabra clave @model.
MvcWebPageRazorHost	Analizador de ASP.NET Razor y el generador de código para un archivo especificado.

Fuente: (Developer\_Network, Referencias de ASP.NET MVC, 2014)

## **2.3. Introducción A ASP.NET MVC 5.**

*“Asp.NET MVC en su quinta versión es un Framework para la construcción de aplicaciones web escalables basadas en estándares usando patrones de diseño bien establecidos y el poder de ASP.NET Framework.” (Asp.Net, 2013)*

### **2.3.1. Lo nuevo en ASP.NET MVC 5.**

- Se puede configurar sin problemas un proyecto ASP.NET MVC 5 utilizando el asistente de creación de un proyecto de ASP.NET.
- Las plantillas de proyecto MVC se han configurado para utilizar ASP.NET identity para autenticación y identity management.
- La plantilla de proyecto MVC se ha actualizado para utilizar Bootstrap, este framework le dará un aspecto elegante a las aplicaciones desarrolladas con ASP.NET MVC.
- ASP.NET MVC 5 permite especificar la lógica de autenticación: por acción, por controlador, o de forma global para todos los controladores.
- ASP.NET MVC 5 soporta atributos de enrutamiento, con los atributos de enrutamiento puede especificar sus rutas al anotar las acciones y controladores.

### **2.3.2. Primer proyecto en ASP.NET MVC 5.**

A continuación se explica cómo empezar a desarrollar aplicaciones web en ASP.NET MVC 5 Framework.

Este documento enseña los fundamentos de la construcción de una aplicación web ASP.NET MVC 5 utilizando el <sup>18</sup>IDE de desarrollo <sup>19</sup>Visual Studio 2013 con código fuente C#.

Primeramente instalar Visual Studio 2013 o Visual Studio Express 2013 para web, estas dos herramientas de desarrollo soportan ASP.NET MVC 5, en este proyecto se utilizará Visual Studio 2013 Express.



**Ilustración 5.** Visual Studio 2013.

**Fuente:** (MSDN, Visual Studio, 2014)

Abrir Visual Studio 2013, en el menú escoger: archivo, nuevo y proyecto.



**Ilustración 6.** Primer proyecto MVC.

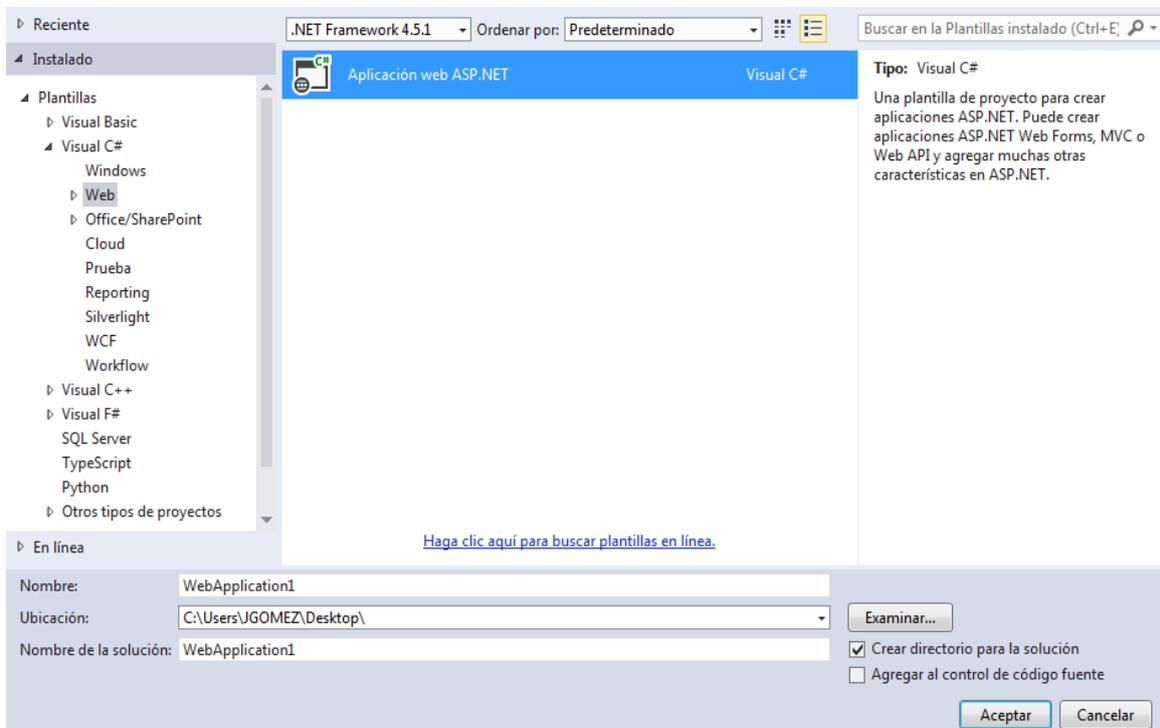
**Fuente:** (Gómez, 2014)

---

<sup>18</sup> **IDE:** Entorno de desarrollo integrado.

<sup>19</sup> **Visual Studio 2013:** Entorno de desarrollo comercializado por Microsoft.

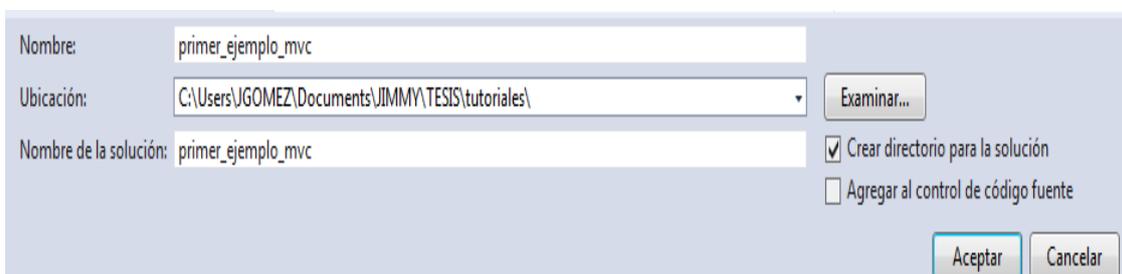
En la lista de opciones del lado izquierdo seleccionar: visual c#, web, seleccionar ASP.NET Web Application.



**Ilustración 7.** Agregar aplicación web.

Fuente: (Gómez, 2014)

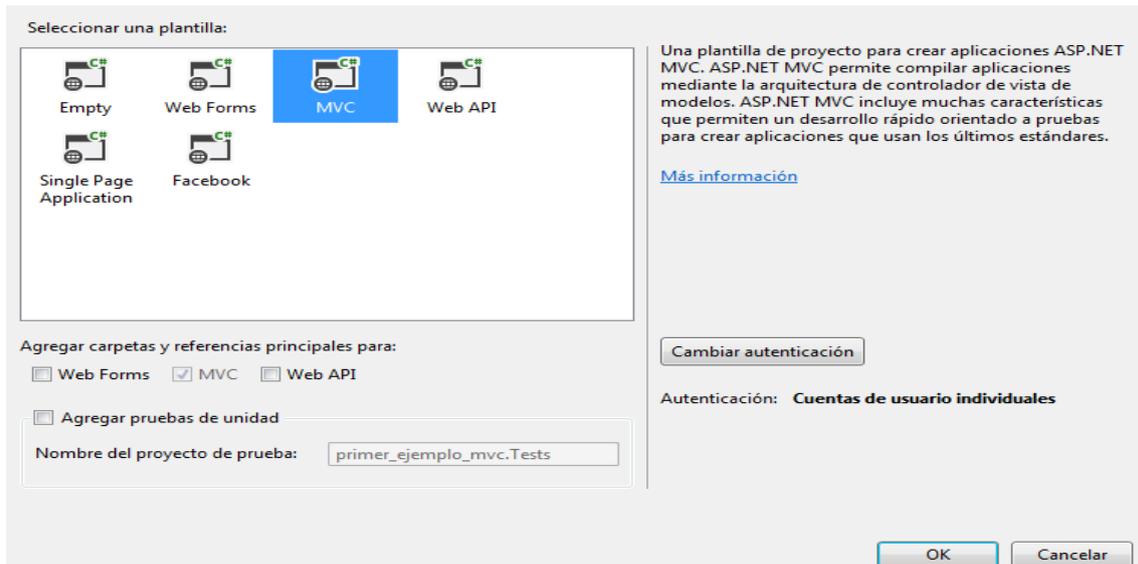
En la parte inferior de la pantalla, configurar los parámetros: nombre, ubicación, nombre de solución y seleccionar aceptar.



**Ilustración 8.** Poner nombre de proyecto.

Fuente: (Gómez, 2014)

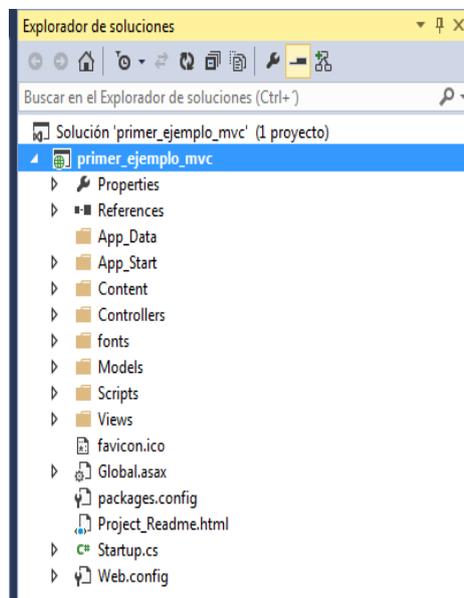
Seleccionar MVC y luego clic en ok.



**Ilustración 9.** Tipos de aplicaciones web.

Fuente: (Gómez, 2014)

Se crea una plantilla predeterminada de un proyecto ASP.NET MVC5.



**Ilustración 10.** Explorador de la solución.

Fuente: (Gómez, 2014)

La solución contiene los siguientes elementos.

- **Referencias:** se añade las librerías a ser utilizadas en la aplicación por lo general librerías de <sup>20</sup>.NET Framework.
- **App\_Data:** carpeta que contiene archivos de datos de la aplicación, pueden ser archivos de base de datos, archivos <sup>21</sup>XML o cualquier otro tipo de almacén de datos.
- **App\_Start:** carpeta que contiene archivos de configuración inicial del proyecto.
- **Content:** carpeta que contiene los archivos de hojas de estilos en cascada (css).
- **Controllers:** carpeta que contiene clases <sup>22</sup>c#, son los controladores Asp.NET MVC de la aplicación, distribuyen entradas de usuario y respuestas, todos los nombres de las clases controladores deben terminar con “Controller”.
- **Fonts:** carpeta que contiene imágenes y fondos utilizados en la aplicación.
- **Models:** carpeta contiene clases c#, son los modelos ASP.NET MVC, almacena lógica de negocio de la aplicación.
- **Scripts:** carpeta que contiene archivos JavaScript que se ejecutan de lado del cliente.
- **View:** carpeta que contiene las vistas ASP.NET MVC de la aplicación, archivos HTML que permiten escribir código c#.
- **Global.asax:** archivo de configuración de eventos y objetos de sesión usados en la aplicación.
- **Packages.config:** lleva un registro de paquetes usados en la aplicación.
- **Startup.cs:** clase c#, primer archivo en ejecutarse dentro de la aplicación, se ejecuta para configurar los componentes que se va a utilizar.

---

<sup>20</sup> **.Net Framework:** Framework de desarrollo de aplicaciones para Windows.

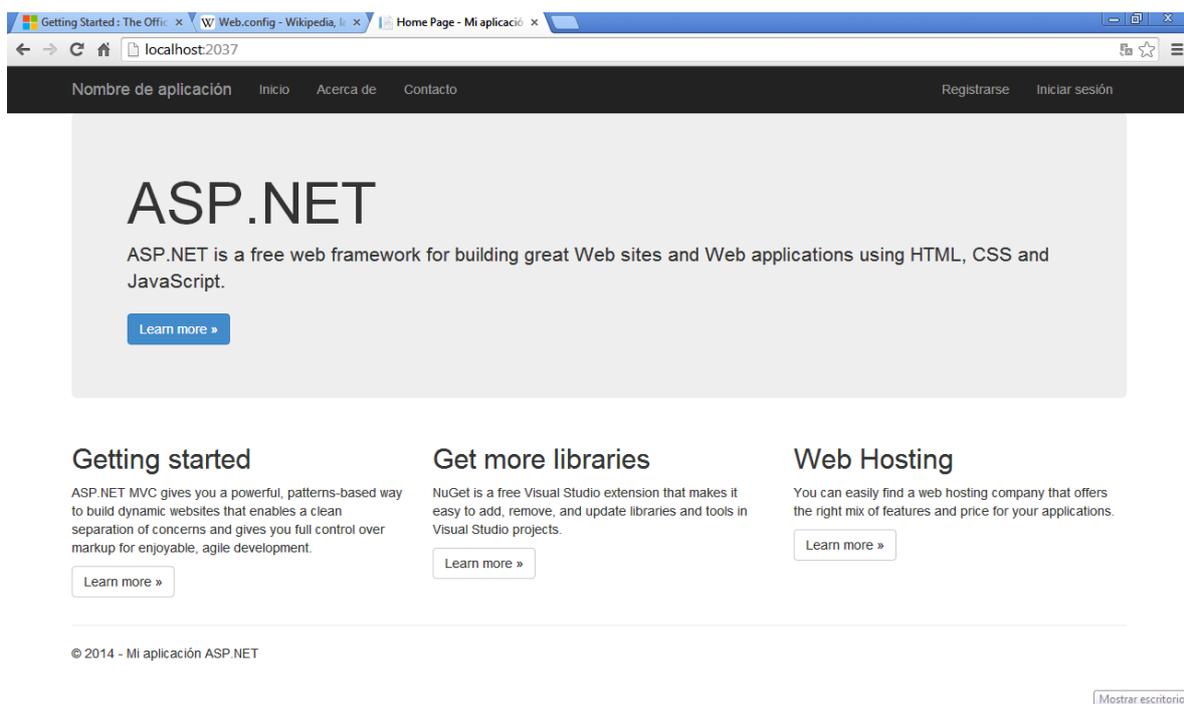
<sup>21</sup> **XML:** Lenguaje utilizado para almacenar datos en forma ordenada.

<sup>22</sup> **C#:** Lenguaje de programación orientado a objetos.

- **Web.config:** registra configuración concerniente a la aplicación web: carga de módulos, configuraciones de seguridad, configuraciones de estado de sesión.

Todos archivos y carpetas pueden ser nombrados, movidos o configurados dependiendo de las necesidades del desarrollador.

Resultado al ejecutar la plantilla por defecto de ASP.NET MVC 5.



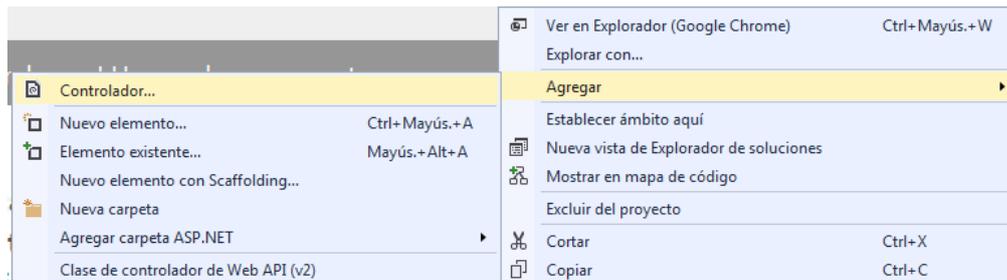
**Ilustración 11.** Resultado de aplicación.

**Fuente:** (Gómez, 2014)

### 2.3.3. Añadir controladores.

En el siguiente ejemplo se explica cómo crear un controlador de ASP.NET MVC en una aplicación web.

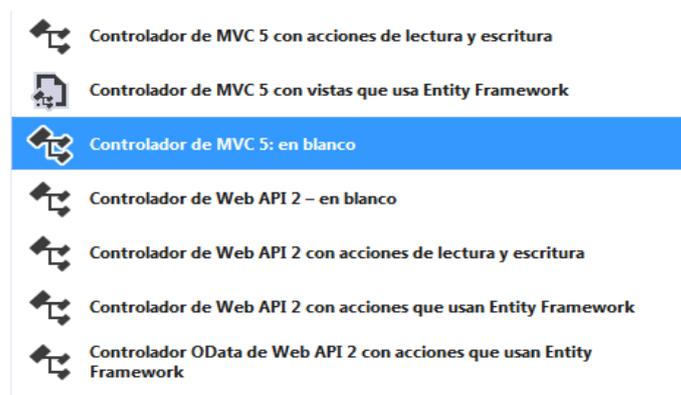
En el explorador de soluciones localizar la carpeta Controllers, clic derecho, Agregar, controlador.



**Ilustración 12.** Añadir controladores.

Fuente: (Gómez, 2014)

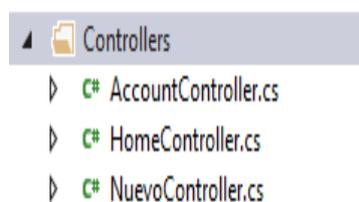
Escoger la opción controlador de MVC 5 en blanco y darle un nombre.



**Ilustración 13.** Tipos de controlador.

Fuente: (Gómez, 2014)

Verificar la creación del controlador llamado NuevoController.cs en el explorador de soluciones.



**Ilustración 14.** Contenido de carpeta Controlles.

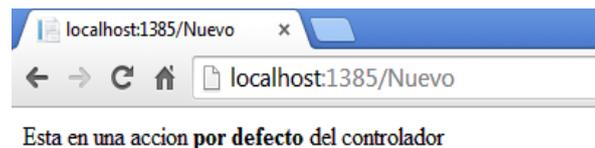
Fuente: (Gómez, 2014)

Añadir el siguiente código en el controlador NuevoController.cs, estos métodos devuelven mensajes al momento de ejecutarlos.

```
Using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace añadir_controlador.Controllers
{
    public class NuevoController : Controller
    {
        public string Index()
        {
            return "Esta en una Acción <b>por defecto</b> del controlador";
        }
    }
}
```

Ejecutar la aplicación y a continuación añadir el nombre del controlador en la barra de direcciones del navegador:

http://localhost:1385/Nuevo

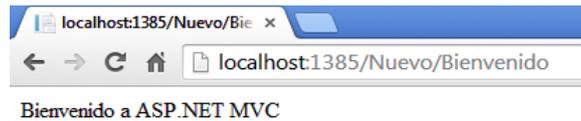


**Ilustración 15.** Mensaje de controlador.

Fuente: (Gómez, 2014)

Se puede determinar en una aplicación ASP.NET MVC el método que se quiere ejecutar en un controlador simplemente especificando en la barra de direcciones del navegador el método a ser ejecutado.

http://localhost:1385/Nuevo/Bienvenido



### Ilustración 16. Bienvenido a MVC.

Fuente: (Gómez, 2014)

Algo novedoso en las aplicaciones ASP.NET MVC es pasar información mediante parámetros de URLs, a continuación se explica cómo realizarlo.

Modificar el método Bienvenido de tal manera que se pueda pasar información mediante parámetros de URLs, el método quedaría de la siguiente manera.

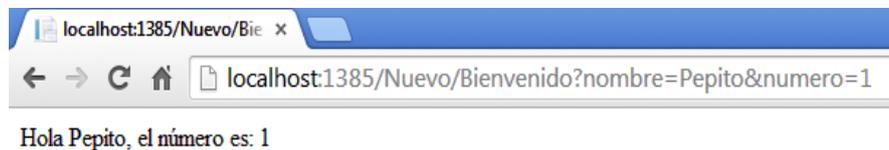
```
public string Bienvenido(string nombre, int numero) {
    nombre = "Pepito";
    numero = 1;
    return HttpUtility.HtmlEncode("Hola " + nombre + ", el número es: " + numero);}

```

Escribir la url de la siguiente manera:

`http://localhost:1385/Nuevo/Bienvenido?nombre=Pepito&numero=1`

Ejecutar el método para observar el resultado.



### Ilustración 17. Resultado de ejecutar controlador.

Fuente: (Gómez, 2014)

Como conclusión la url en una aplicación ASP.NET MVC está constituida por:

`http://servidor:puerto/controlador/acción/parametros`

Existe otra forma de ejecutar un controlador y es mediante el archivo App\_Start \ RouteConfig.cs, lo único que hace es determinar que controlador y los parámetros que se ejecutará por defecto al iniciar la aplicación como se muestra a continuación.

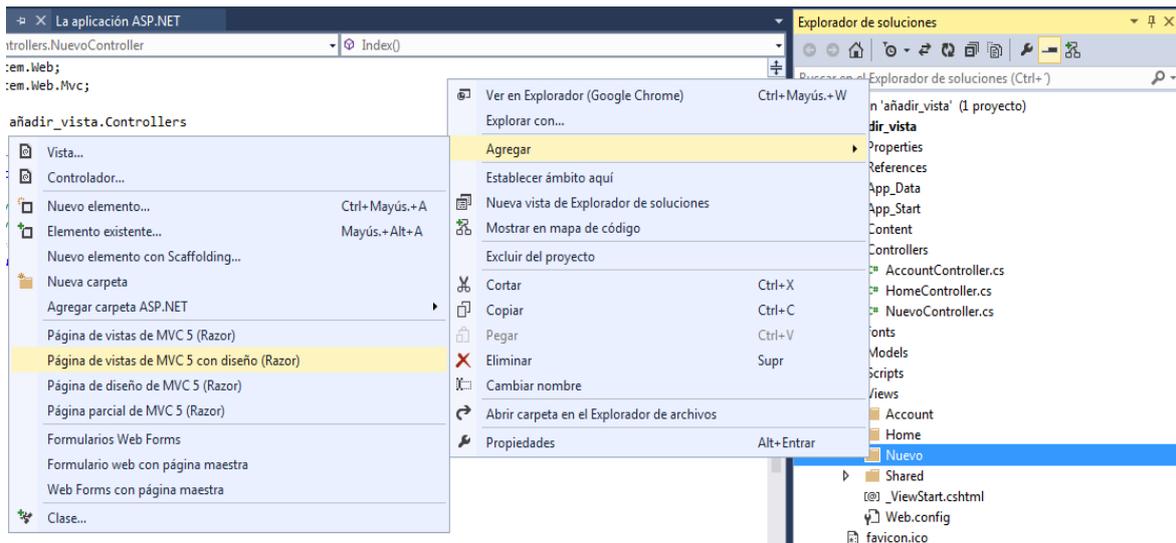
```
Public class RouteConfig {  
  
    public static void RegisterRoutes(RouteCollection routes) {  
  
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");  
  
        routes.MapRoute(  
  
            name: "Default",  
  
            url: "{controller}/{action}/{id}",  
  
            defaults: new { controller = "Nuevo", action = "Bienvenido", id = UrlParameter.Optional }  
  
        );  
    }  
}
```

#### 2.3.4. Añadir vistas.

Las plantillas de vistas basadas en Razor tienen extensiones de archivos cshtml, proporcionan una manera elegante de crear presentaciones en HTML utilizando c#.

En esta parte se explica cómo añadir una vista en un proyecto ASP.NET MVC 5.

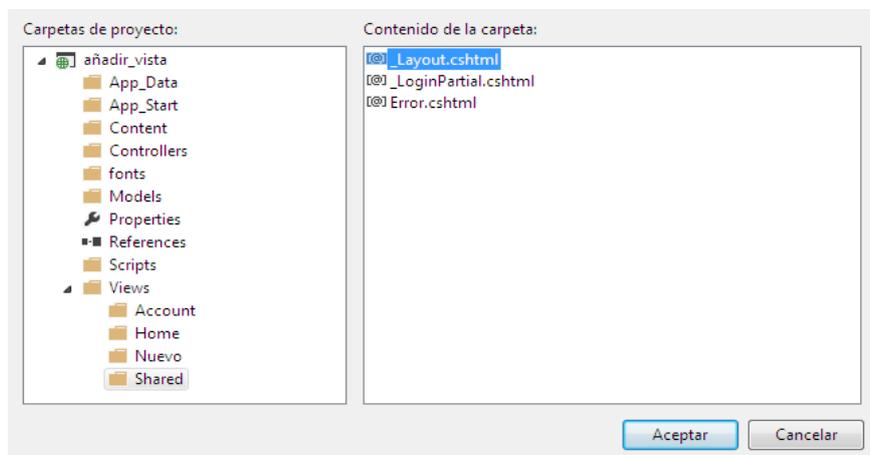
En el explorador de soluciones clic derecho en la carpeta con el nombre del controlador en este caso Nuevo seguido de página de vistas con diseño Razor y darle un nombre.



**Ilustración 18.** Añadir vistas.

Fuente: (Gómez, 2014)

Escoger el <sup>23</sup>Layout o más conocido como página maestra al que va estar asociada la vista.

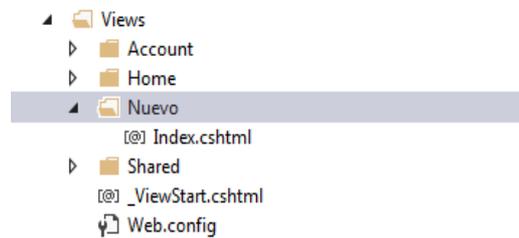


**Ilustración 19.** Escoger Layout.

Fuente: (Gómez, 2014)

<sup>23</sup> **Layout:** Es la parte de diseño gráfico o interfaz gráfica de una aplicación web.

En el explorador de soluciones verificar la creación de la vista `index.cshtml` en la carpeta del controlador Nuevo.



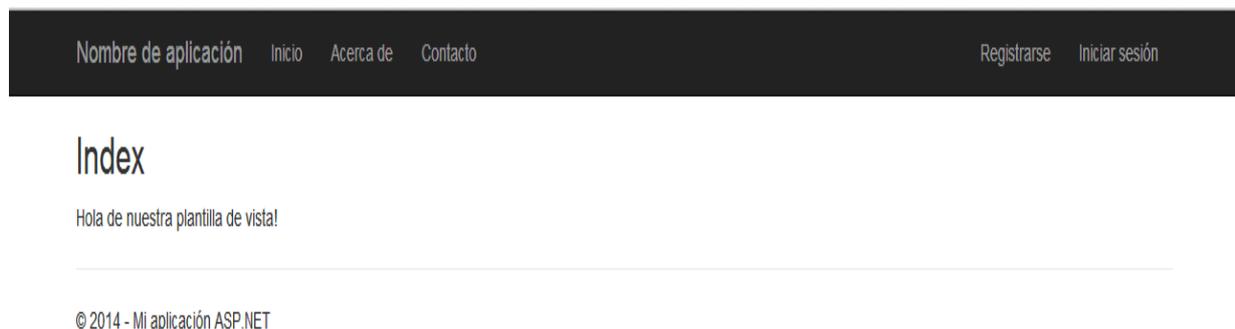
**Ilustración 20.** Contenido de carpetas View.

Fuente: (Gómez, 2014)

Agregar el siguiente código en la vista `index.cshtml`, básicamente lo que hace es determinar cuál es el Layout al que está asociada la vista, darle un título a la página y mostrar un diálogo.

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@{
    ViewBag.Title = "Bienvenido";
}
<h2> Hola de nuestra plantilla vista</h2>
```

Observar el resultado de ejecutar la aplicación.



**Ilustración 21.** Resultado de ejecutar la vista.

Fuente: (Gómez, 2014)

A continuación se analiza como pasar datos desde el controlador a una vista; en la clase controlador, se escribe el código que controla las peticiones entrantes, recupera información de la capa modelo y decide qué tipo de respuesta va enviar de vuelta al navegador.

Regresar a la clase controlador Nuevo.cs y cambiar el método para agregar un ViewBag que retorne un mensaje, ViewBag es un objeto dinámico en el cual se puede agregar cualquier valor; el método modificado queda de la siguiente manera.

```
public class NuevoController : Controller{

    public ActionResult Index(){

        return View();}

    public ActionResult Bienvenido(string nombre, int numero = 1){

        ViewBag.Message = "Hola " + nombre;

        ViewBag.numero = numero;

        return View(); }

}
```

Ahora modificar la vista Index.cshtml perteneciente al controlador llamado Nuevo de tal manera que se pueda visualizar el mensaje almacenado en el ViewBag, el código es el siguiente:

Ejecutar la aplicación escribiendo la siguiente url:

<http://localhost:1458/Nuevo/Index/Bienvenido?nombre=Pepito&numero=4>

### 2.3.4. Añadir modelado.

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@{
    ViewBag.Title = "Bienvenido";
}
<h2>
    Bienvenido </h2>
<ul>
    @for (int i = 0; i < ViewBag.numero; i++) {
        <li> @ViewBag.Message </li>
    }</ul>
```

En esta parte se explica cómo crear clases de la capa modelo en una aplicación ASP.NET MVC, estas contendrán toda la lógica de negocio de la aplicación, para este ejemplo se utilizará una clase persona.

En el explorador de soluciones, localizar la carpeta Models, clic derecho, agregar una nueva clase y escribir el siguiente código:

```
public class Persona {

    public int id { get; set; }
    public string nombres { get; set;}
    public string apellidos { get; set;}
    public string direccion { get; set;}
    public int telefono { get; set; }

}
```

Agregar la siguiente clase que servirá de puente entre el modelo y la base de datos, esta clase devuelve un tipo de dato DbSet el cual representa un almacén de datos.

```
public class Datos : DbContext
{
    public DbSet<Persona> personas { get; set; }
}
```

Es importante agregar también la librería de ensamblado que permite trabajar con datos.

## **2.4. Introducción a ENTITY FRAMEWORK 6 EN ASP.NET MVC 5.**

*“Entity Framework es un mapeador objeto-relacional que facilita a los programadores trabajar con datos relacionales usando objetos específicos de dominio, este Framework permite que los datos sean tratados como entidades independientemente de sus representaciones del almacén de datos subyacente”. (MSDN, Entity Framework, 2014)*

Entity Framework permite crear rápidamente una aplicación web con interacción a datos. La característica de ASP.NET Scaffolding genera automáticamente el código <sup>24</sup>CRUD de la aplicación web.

A continuación se muestra como integrar Entity Framework con MVC 5 creando primero una base de datos (Database first) y escribiendo primero código (Code first).

### **2.4.1. Entity Framework 6, Database first.**

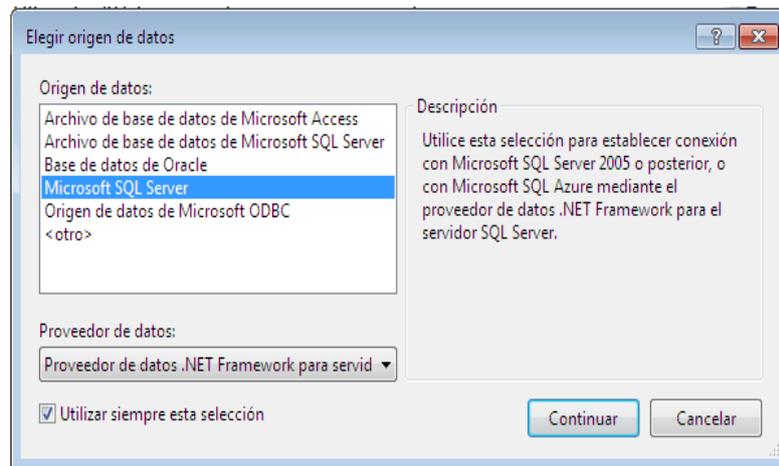
Para este tutorial se utilizará una base de datos que representa estudiantes y sus notas.

Primero, en el explorador de servidores de Visual Studio 2013 agregar una nueva conexión de datos y escoger una fuente de datos en este caso <sup>25</sup>Microsoft SQL Server.

---

<sup>24</sup> **CRUD:** Siglas que representan las funciones básicas en base de datos (Crear, Leer, Actualizar, Eliminar).

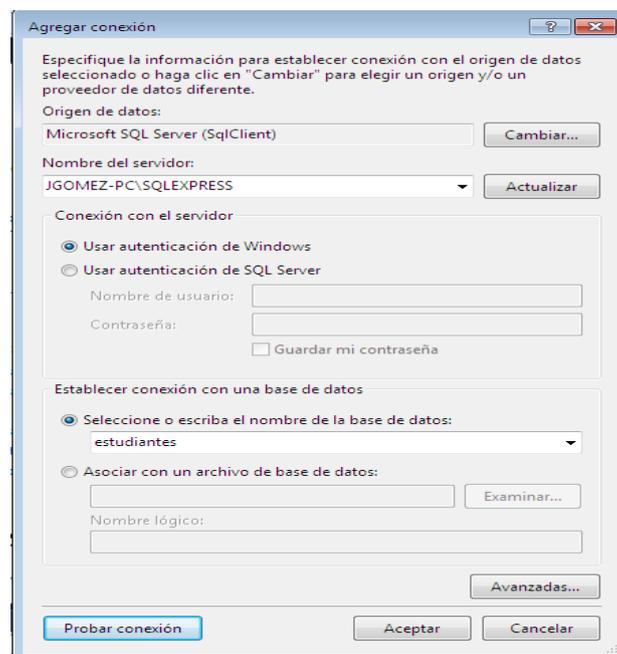
<sup>25</sup> **Microsoft SQL Server:** Sistema de gestión de base de datos.



**Ilustración 22.** Conexión con SQL Server.

Fuente: (Gómez, 2014)

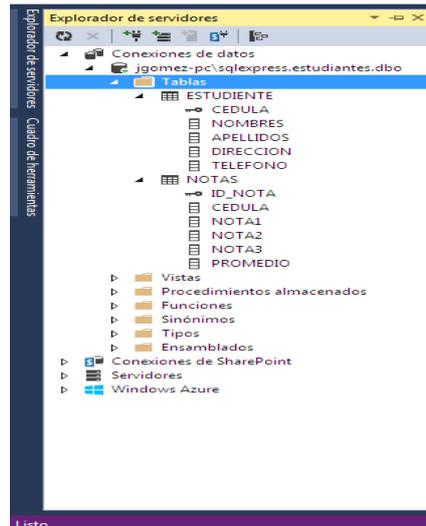
Configurar los parámetros de conexión: nombre del servidor, tipo de autenticación y nombre de la base de datos; probar conexión y aceptar.



**Ilustración 23.** Agregar conexión.

Fuente: (Gómez, 2014)

En el explorador de los servidores verificar si la conexión se realizó con éxito, revisando las tablas y campos de la de la base de datos.



**Ilustración 24.** Explorador de servidores.

Fuente: (Gómez, 2014)

Para poblar de datos la base de datos, seleccionar una tabla, clic derecho y escoger la opción mostrar datos de tabla, se visualizará los datos existentes en la tabla y permitirá agregar nuevos.

	CEDULA	NOMBRES	APELLIDOS	DIRECCION	TELEFONO
	1003266952	Pepito	Perez	Ibarra	999999999
▶*	NULL	NULL	NULL	NULL	NULL

**Ilustración 25.** Resultado de ejecutar consulta.

Fuente: (Gómez, 2014)

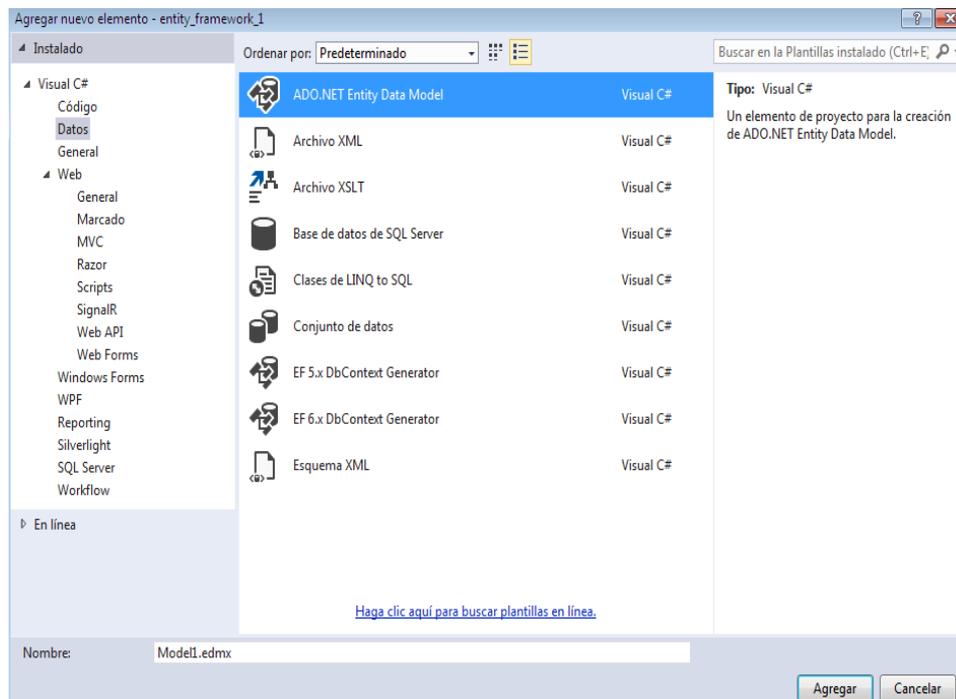
#### 2.4.2. Creación de una aplicación web MVC 5 y modelo de datos.

Este ejemplo se centra en crear una aplicación web MVC 5, y la generación de modelos de datos basados en las tablas de la base de datos estudiante y notas.

Entity Framework crea modelos de datos representados mediante clases, cada clase representa una tabla de la base de datos, cada propiedad representa una columna de la tabla.

Crear una nueva aplicación web ASP.NET MVC, verificar la creación del aplicativo revisando en el explorador de la solución los archivos pertinentes.

En la carpeta Models del explorador de soluciones; clic derecho y seleccionar nuevo elemento; en la ventana, nuevo elemento seleccionar datos en el panel de la izquierda y <sup>26</sup>ADO .NET Entity Data Model en el panel central, darle un nombre al archivo “estudiante”; clic en Agregar.

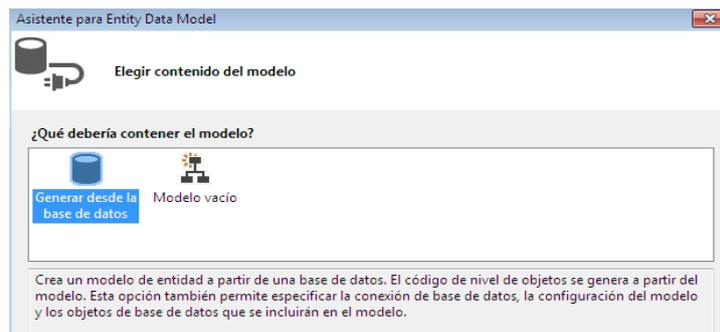


**Ilustración 26.** Agregar modelo de datos.

**Fuente:** (Gómez, 2014)

En el asistente de Entity Framework Data Model, escoger la opción Generar desde base de datos; clic en siguiente.

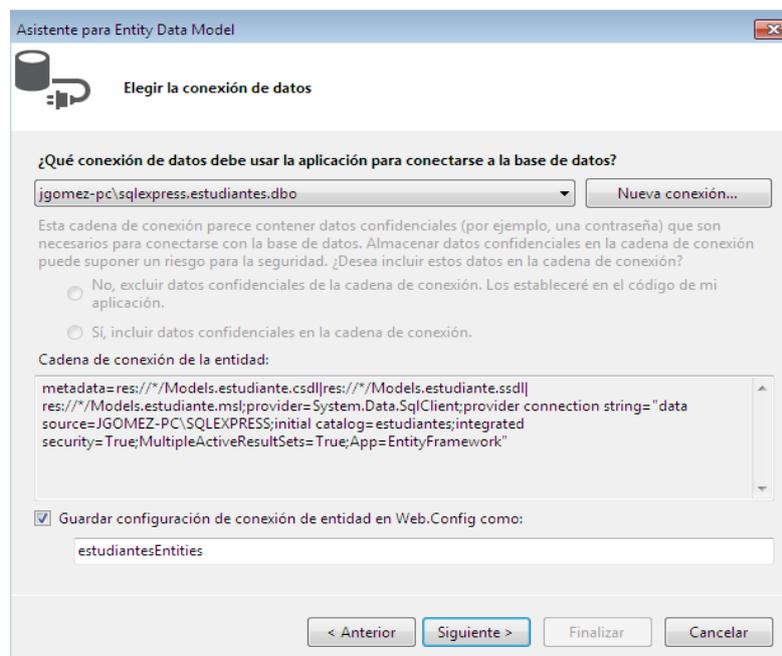
<sup>26</sup> **ADO .NET:** Componente de software que sirve para acceder a datos de una base de datos.



**Ilustración 27.** Asistente de modelo de datos.

Fuente: (Gómez, 2014)

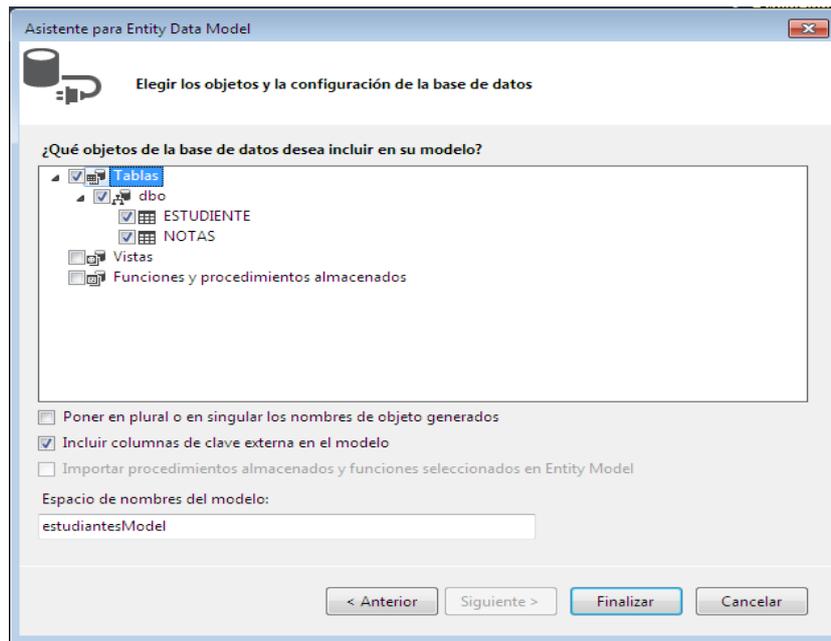
En este paso seleccionar la conexión agregada anteriormente en este ejemplo; clic en siguiente.



**Ilustración 28.** Conexión con base de datos.

Fuente: (Gómez, 2014)

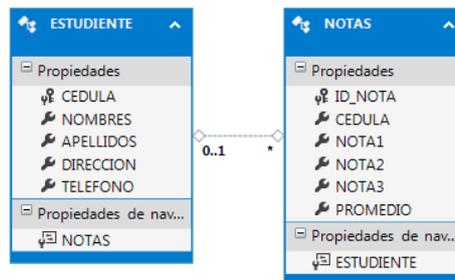
Seleccionar las tablas de la base de datos; clic en finalizar.



**Ilustración 29.** Escoger tabla para el modelo de datos.

Fuente: (Gómez, 2014)

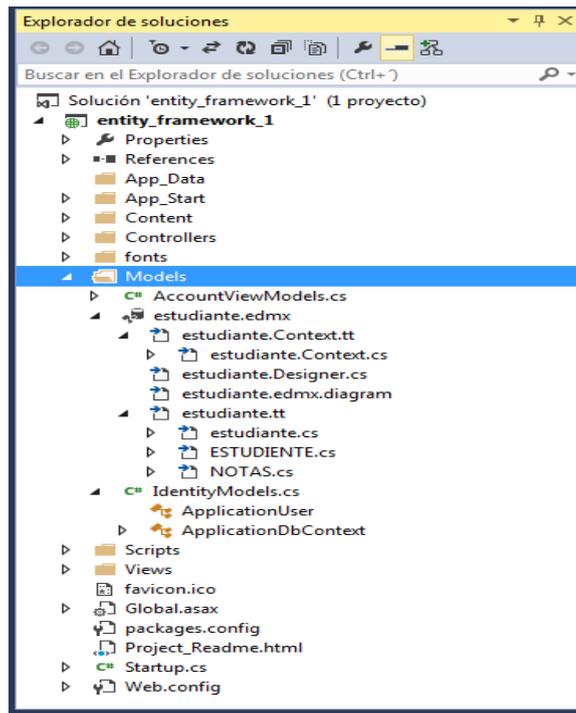
Revisar el diagrama del modelo de datos creado por Entity Framework.



**Ilustración 30.** Modelo de datos.

Fuente: (Gómez, 2014)

En la carpeta Models se puede apreciar los nuevos archivos creados por Entity Framework que representa el modelo de datos.



**Ilustración 31.** Contenido de carpeta Models.

Fuente: (Gómez, 2014)

La clase `Estudiante.Context.cs` contiene las propiedades para cada clase del modelo correspondiente a las tablas de la base de datos.

Las clases `Estudiante.cs` y `NOTAS.cs` representan el modelo de las tablas de la base de datos.

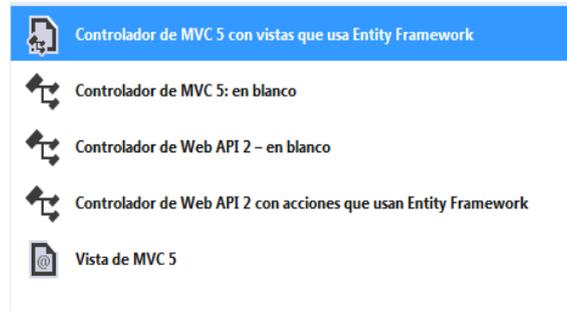
Compilar el proyecto para prevenir algún error.

### 2.4.3. Generación de vistas.

En esta parte se genera el código que proporciona las operaciones básicas para el modelo de datos utilizando el Scaffold de ASP.NET.

El Scaffold incluye el controlador y las vistas que corresponden al modelo de la base de datos estudiantes y sus notas utilizados anteriormente.

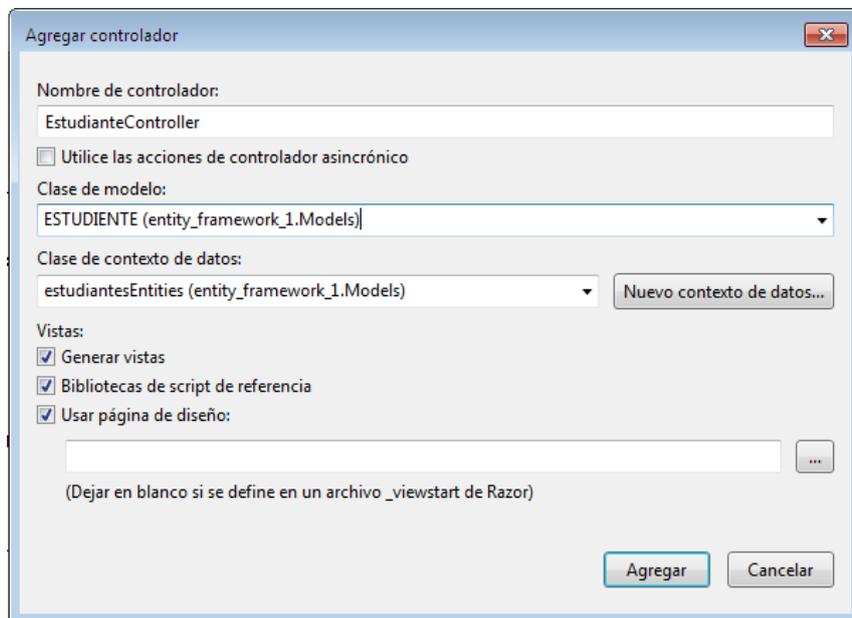
En la carpeta Controllers del explorador de la solución, clic derecho, seleccionar la opción Agregar y Nuevo elemento con Scaffolding. En la ventana Añadir scaffold, seleccionar la opción Controlador MVC 5 con vistas que usa Entity Framework, clic en agregar.



**Ilustración 32.** Vistas con Entity Framework 6.

Fuente: (Gómez, 2014)

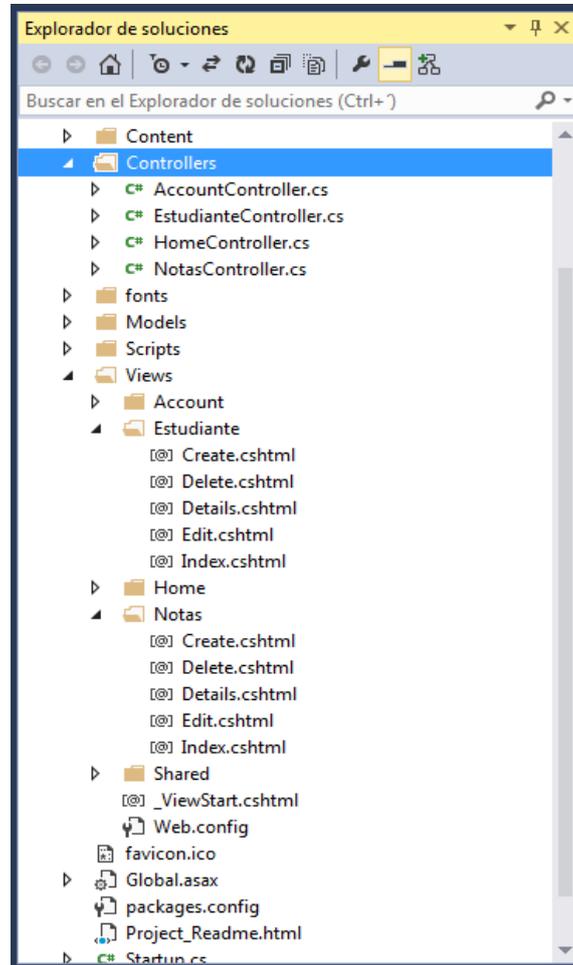
Establecer un nombre al controlador en este caso EstudianteController, Seleccionar la clase de modelo Estudiante y la clase de contexto de datos EstudiantesEntities; clic en Agregar.



**Ilustración 33.** Crear controlador Estudiante.

Fuente: (Gómez, 2014)

Realizar los mismos pasos para añadir el scaffold de la clase de modelo NOTAS. Al final se puede evidenciar los controladores tanto de estudiantes como de notas con sus respectivas vistas con las acciones de creación, eliminación, detalles, vistas, Editar e Índice.



**Ilustración 34.** Contenido de carpeta Controllers.

Fuente: (Gómez, 2014)

En la carpeta Views/Estudiante/Index.cshtml, seleccionar el archivo, clic derecho, Ver en el explorador.

Se mostrara el resultado del código generado con las acciones de listar, crear, editar y eliminar.

Listar datos de Estudiantes.

## Index

[Create New](#)

NOMBRES	APELLIDOS	DIRECCION	TELEFONO	
Pepito	Perez	Ibarra	999999999	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Juan	Lopez	Ibarra	000000000	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Jimmy	Gómez	Ibarra	097766787	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

© 2014 - Mi aplicación ASP.NET

### Ilustración 35. Listar Estudiante.

Fuente: (Gómez, 2014)

Detalles del Estudiante.

## Details

### ESTUDIENTE

<b>NOMBRES</b>	Pepito
<b>APELLIDOS</b>	Perez
<b>DIRECCION</b>	Ibarra
<b>TELEFONO</b>	999999999

[Edit](#) | [Back to List](#)

© 2014 - Mi aplicación ASP.NET

### Ilustración 36. Detalle Estudiante.

Fuente: (Gómez, 2014)

Crear nuevo estudiante.

## Create

### ESTUDIENTE

---

CEDULA	<input type="text"/>
NOMBRES	<input type="text"/>
APELLIDOS	<input type="text"/>
DIRECCION	<input type="text"/>
TELEFONO	<input type="text"/>
	<input type="button" value="Create"/>

[Back to List](#)

---

© 2014 - Mi aplicación ASP.NET

### **Ilustración 37.** Crear Estudiante.

**Fuente:** (Gómez, 2014)

Editar Estudiante.

## Edit

### ESTUDIENTE

---

NOMBRES	<input type="text" value="Pepito"/>
APELLIDOS	<input type="text" value="Perez"/>
DIRECCION	<input type="text" value="Ibarra"/>
TELEFONO	<input type="text" value="999999999"/>
	<input type="button" value="Save"/>

[Back to List](#)

---

© 2014 - Mi aplicación ASP.NET

### **Ilustración 38.** Editar Estudiante.

**Fuente:** (Gómez, 2014)

Eliminar Estudiante.

## Delete

Are you sure you want to delete this?

ESTUDIENTE

<b>NOMBRES</b>	Pepito
<b>APELLIDOS</b>	Perez
<b>DIRECCION</b>	Ibarra
<b>TELEFONO</b>	999999999

Delete | [Back to List](#)

© 2014 - Mi aplicación ASP.NET

### Ilustración 39. Eliminar Estudiante.

Fuente: (Gómez, 2014)

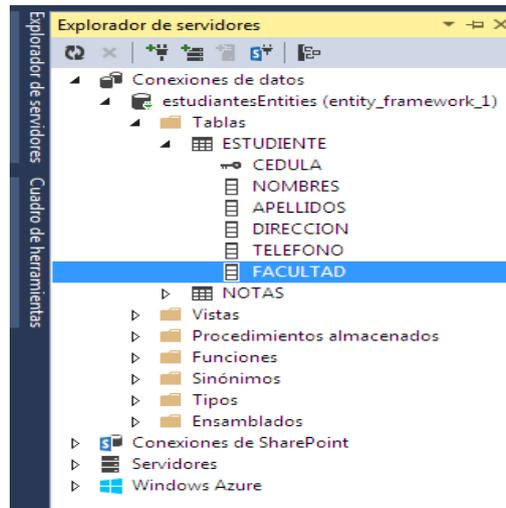
#### 2.4.4. Cambios en la Base de Datos.

El objetivo de este ejemplo es demostrar cómo actualizar el modelo de datos cuando se realiza un cambio en la base de datos, para lo cual agregar una nueva columna en la tabla estudiantes llamada facultad.

En el Explorador de servidores Conexión de datos/EstudiantesEntities/tablas clic derecho en la tabla Estudiante y escoger la opción Nueva consulta; ejecutar la siguiente consulta.

```
ALTER TABLE [ dbo ]. [ ESTUDIENTE ]  
  
ADD [ FACULTAD ] VARCHAR ( 100 ) NULL ;
```

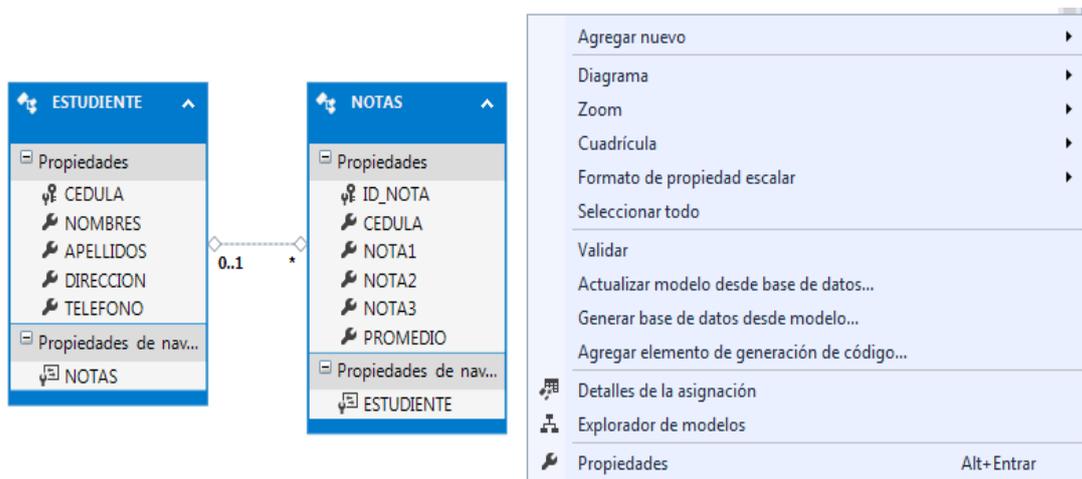
Observar el nuevo campo creado en el Explorador de servidores.



**Ilustración 40.** Explorador de servidores.

Fuente: (Gómez, 2014)

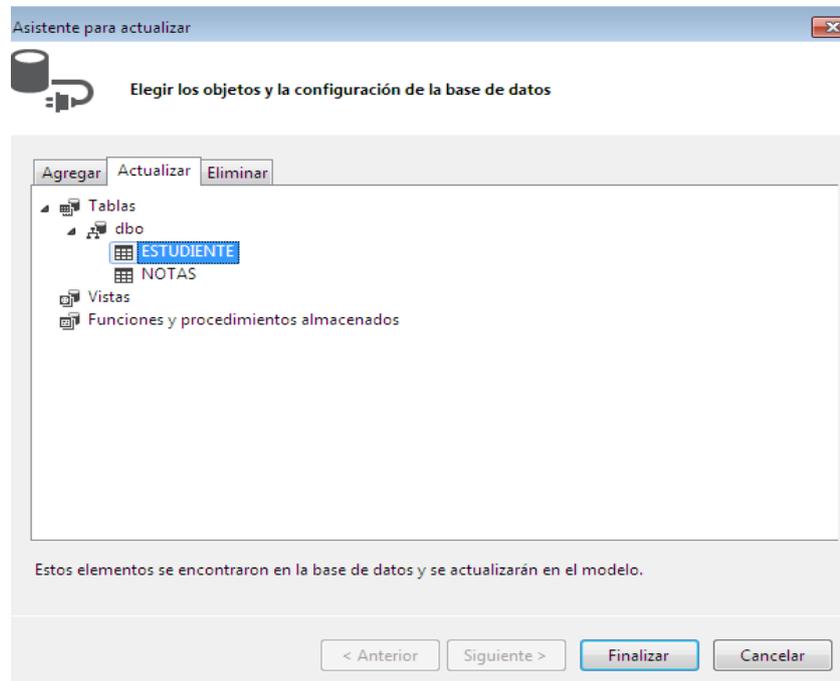
Ahora la base de datos está actualizada pero no el modelo de datos, para lo cual abrir el archivo Estudiante.edmx, clic derecho en cualquier parte de la superficie del diagrama de datos y escoger la opción Actualizar modelo desde base de datos.



**Ilustración 41.** Agregar nuevo elemento en modelo de datos.

Fuente: (Gómez, 2014)

En el asistente de actualización, seleccionar la tabla que se desea actualizar y clic en finalizar.



**Ilustración 42.** Escoger tabla.

Fuente: (Gómez, 2014)

Después del proceso de actualización en el modelo de datos se ha agregado el nuevo campo Facultad; guardar el archivo Estudiante.edmx y compilar el proyecto para que surta efecto los cambios.

Para que el nuevo campo se visualice en la vista es necesario regenerar todas las vistas o agregar el campo manualmente.

#### 2.4.5. Personalización de vistas.

El código generado automáticamente es un buen punto de partida, pero no proporciona toda la funcionalidad a la aplicación, Por lo que es necesario personalizar el código para satisfacer las necesidades particulares del sistema. Actualmente la aplicación muestra la lista de

estudiantes pero no los estudiantes con sus respectivas notas, para lo cual es necesario modificar las vistas.

Abrir la vista Details.cshtml del estudiante, agregar el siguiente código después de la etiqueta de cierre `</dl>` y antes de la etiqueta de cierre `</div>`

```

<table class="table">
  <tr>
    <th>
      NOTA 1
    </th>
    <th>
      NOTA 2
    </th>
    <th>
      NOTA 3
    </th>
    <th>
      PROMEDIO
    </th>
  </tr>
  <tr>
    <td>
      @Html.DisplayFor(modelItem => item.NOTA1)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.NOTA2)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.NOTA3)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.PROMEDIO)
    </td>
    <td>
      @Html.ActionLink("Edit", "Edit", new { id = item.ID_NOTA }) |
      @Html.ActionLink("Details", "Details", new { id = item.ID_NOTA }) |
      @Html.ActionLink("Delete", "Delete", new { id = item.ID_NOTA })
    </td>
  </tr>
}
</table>

```

Como resultado se puede visualizar el estudiante con sus respectivas notas.

## Details

### ESTUDIENTE

<b>NOMBRES</b>	Pepito		
<b>APELLIDOS</b>	Perez		
<b>DIRECCION</b>	Ibarra		
<b>TELEFONO</b>	999999999		

---

NOTA 1	NOTA 2	NOTA 3	PROMEDIO	
1	1	1	2.00	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

[Edit](#) | [Back to List](#)

© 2014 - Mi aplicación ASP.NET

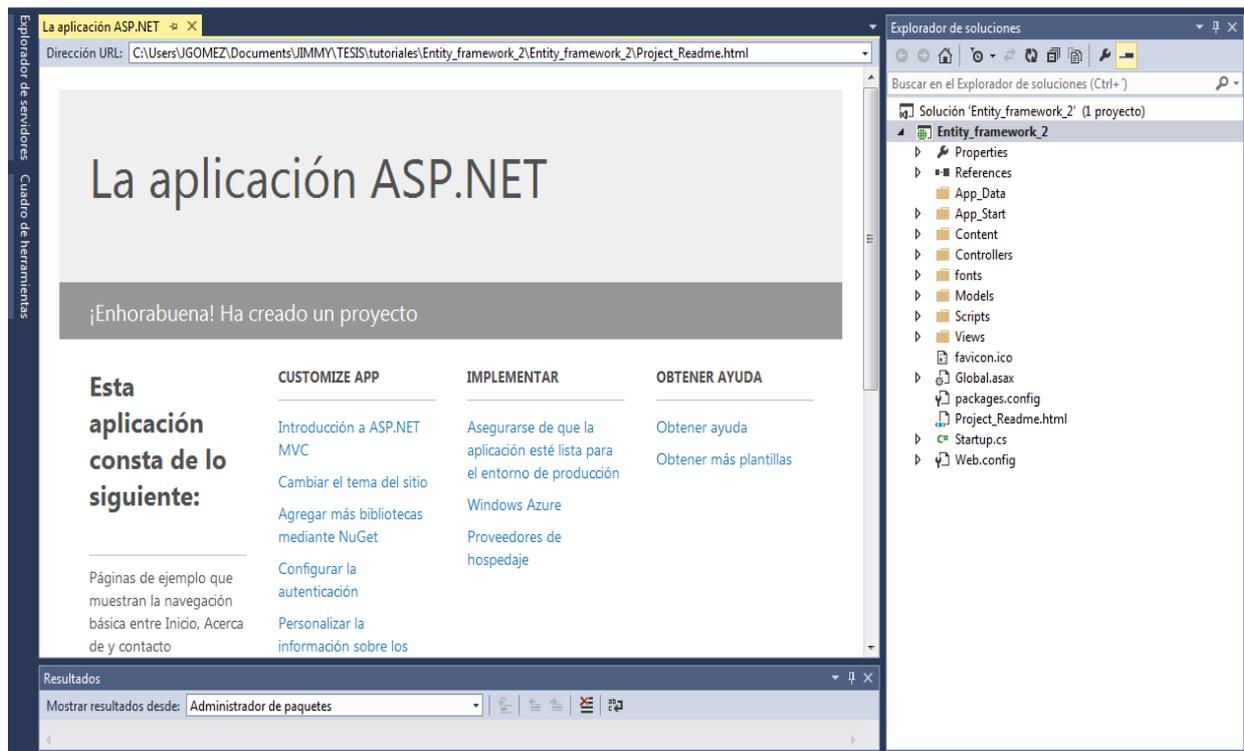
### Ilustración 43. Maestro detalle, estudiante.

Fuente: (Gómez, 2014)

#### 2.4.6. Entity Framework 6, Code first.

En este ejemplo se trabaja con Entity Framework 6, pero esta vez escribiendo primeramente el código del modelo de datos y luego generando la base de datos. Se utilizará la misma lógica de la base de datos trabajada anteriormente, estudiantes y sus notas.

Primeramente crear un nuevo proyecto ASP.NET MVC 5, tal como se lo ha venido creando anteriormente.



**Ilustración 44.** Aplicación web MVC.

Fuente: (Gómez, 2014)

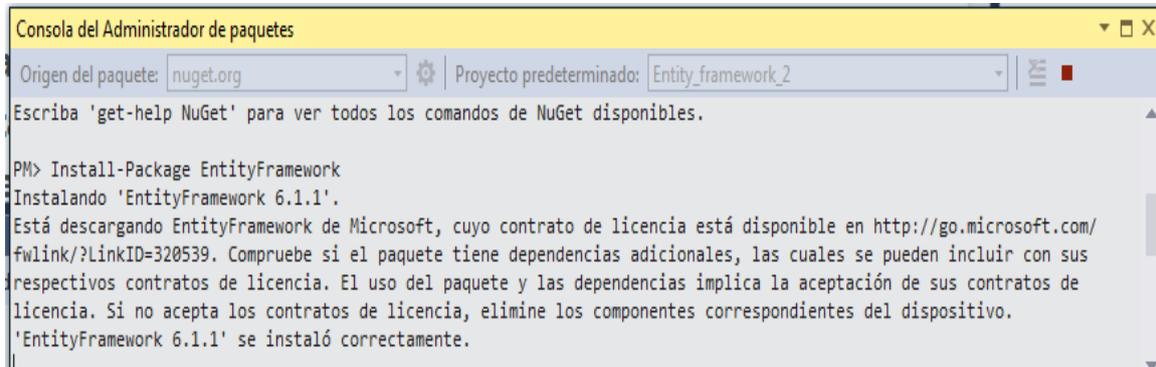
Esta vez se mostrará otra forma de trabajar con Entity Framework en caso de que este no esté instalado.

En la consola de administración de paquetes <sup>27</sup>Nuget escribir el siguiente comando, el mismo que instalará Entity Framework 6 en la aplicación.

---

<sup>27</sup> **Nuget:** Administrador de paquetes de Visual Studio.

### Install-Package EntityFramework



```

Consola del Administrador de paquetes
Origen del paquete: nuget.org | Proyecto predeterminado: Entity_framework_2
Escriba 'get-help NuGet' para ver todos los comandos de NuGet disponibles.

PM> Install-Package EntityFramework
Instalando 'EntityFramework 6.1.1'.
Está descargando EntityFramework de Microsoft, cuyo contrato de licencia está disponible en http://go.microsoft.com/fwlink/?LinkID=320539. Compruebe si el paquete tiene dependencias adicionales, las cuales se pueden incluir con sus respectivos contratos de licencia. El uso del paquete y las dependencias implica la aceptación de sus contratos de licencia. Si no acepta los contratos de licencia, elimine los componentes correspondientes del dispositivo.
'EntityFramework 6.1.1' se instaló correctamente.
  
```

**Ilustración 45.** Consola de administración de paquetes.

Fuente: (Gómez, 2014)

Ahora crear las clases del modelo de datos. En la carpeta Models agregar una nueva clase denominada Estudiante.cs y reemplazar el código predeterminado por este.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace Entity_framework_2.Models{
    public class Estudiante {
        public int cedula { get; set; }
        public string nombres { get; set; }
        public string apellidos { get; set; }
        public string direccion { get; set; }
        public string telefono { get; set; }
        public string facultad { get; set; }
    }
}
  
```

Crear la clase de acceso a los datos para lo cual en el explorador de soluciones, crear una nueva carpeta llamada Acceso\_datos que contendrá la clase llamada EstudianteContext con el siguiente código.

```

Using Entity_framework_2.Models;
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;
namespace Entity_framework_2.Acceso_datos{
    public class EstudianteContext : DbContext
    {
        public EstudianteContext() : base("EstudianteContext")
        {
        }
        public DbSet<Estudiante> Estudiante { get; set; }
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
        }
    }
}

```

Esta clase contiene el método de especificación del nombre de la cadena de conexión, la misma que se especifica en el Web.Config para establecer una conexión con la base de datos.

```

public EstudianteContext() : base("EstudianteContext")
{
}

```

También contiene el código de creación de un DbSet que representa a cada una de las tablas de la base de datos.

```

public DbSet<Estudiante> Estudiante { get; set; }

```

En la carpeta de Acceso a datos, crear una nueva clase llamada EstudianteInitializer.cs, esta clase funciona de inicialización de la base de datos que crea la base de datos cuando lo necesite y carga los datos de prueba; agregar el siguiente código.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;
using Entity_framework_2.Models;
namespace Entity_framework_2.Acceso_datos
{
    Publicclass SchoolInitializer : System.Data.Entity.DropCreateDatabaseIfModelChanges<EstudianteContext>
    {
        protected override void Seed(EstudianteContext context)
        {
            var estudiantes = new List<Estudiante>
            {
                new Estudiante { cedula = "1003266954" , nombres = "Pepito" , apellidos = "Perez", direccion = "Ibarra",
                telefono="0998877655", facultad="FICA" },
                new Estudiante { cedula = "1003266999" , nombres = "Juan" , apellidos = "Perez", direccion = "Ibarra",
                telefono="0998999655", facultad="FECYT" },
                new Estudiante { cedula = "1003269999" , nombres = "Diego" , apellidos = "Lopez", direccion = "Quito",
                telefono="009876877655", facultad="FICAYA" }
            };
            estudiantes.ForEach(s => context.Estudiante.Add(s));
            context.SaveChanges();
        }
    }
}

```

En el Web.config, archivo que se encuentra en la raíz del proyecto; debajo del nodo `<entityFramework>` agregar el siguiente código, que funciona para que Entity Framework tome la clase de inicialización de la base de datos y la ejecute.

```

<contexts>
  <context type="ContosoUniversity.DAL.SchoolContext, ContosoUniversity">
    <databaseInitializer type="ContosoUniversity.DAL.SchoolInitializer, ContosoUniversity" />
  </context>
</contexts>

```

Este código especifica la ruta y el nombre de la clase de inicialización de la base de datos que será ejecutado cada vez que el proyecto sea compilado, si se desea restringir la inicialización de la base de datos puede establecer el atributo

`disableDatabaseInitialization="true"` o `disableDatabaseInitialization="false"` dependiendo de la necesidad del proyecto.

Para este ejemplo se utilizara localDB como base de datos; a continuación se explica cómo crear una localDB utilizando Entity Framework.

En el Web.config agregar la cadena de conexión a la base de datos, en el nodo `<connectionStrings>` escribir el siguiente código.

```
<add name="EstudianteContext" connectionString="Data Source=(LocalDb)\v11.0;Initial Catalog=Estudiante;Integrated Security=SSPI;" providerName="System.Data.SqlClient"/>
```

Generar los controladores y vistas tal como se realizó anteriormente en este ejemplo, compilar y ejecutar el proyecto, para presenciar los resultados.

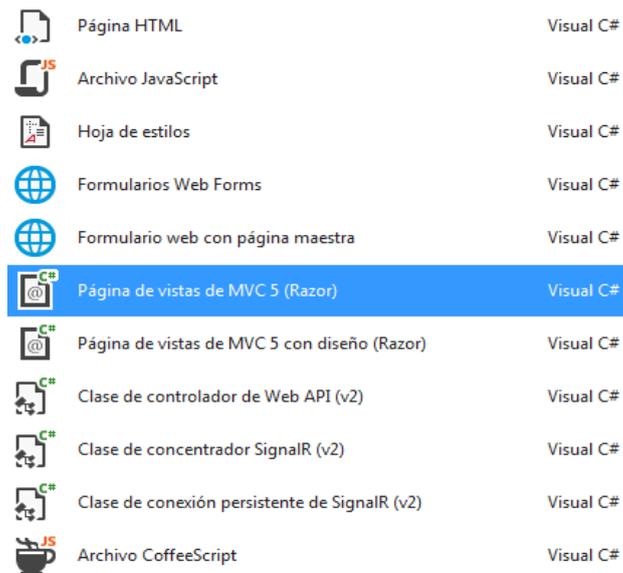
## 2.5. Vistas Razor en ASP.NET MVC 5.

Razor es un motor de vistas dedicado a tareas de presentación, son bloques de código que realizan peticiones al servidor para terminar mostrando información en formato HTML.

### 2.5.1. Programación básica en Razor.

A continuación se presenta una visión general de como programar con sintaxis Razor en aplicaciones ASP.NET MVC 5.

El primer paso es agregar una vista en el proyecto ASP.NET MVC, clic en el explorador de la solución, agregar nuevo elemento y escoger Páginas de vistas MVC 5 (Razor).



**Ilustración 46.** Agregar una nueva vista.

Fuente: (Gómez, 2014)

Abrir la vista.cshtml y examinar el código.

```
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title></title>
</head>
<body>
    <div>
    </div>
</body></html>
```

La vista está constituida por dos bloques, el bloque superior de código c# y el código inferior de código HTML.

### 2.5.2. Reglas de la sintaxis Razor.

El caracter “@” significa que lo que sigue es código no Html, código razor.

Las llaves “{y}” se utilizar para escribir bloques de código razor, si este código tiene más de una línea.

Cada declaración obligatoriamente termina con “;”.

Para declarar variables se utiliza la palabra reservada “var” independientemente del tipo de dato.

Las cadenas de caracteres deben ser escritas entre de comillas “y”, los números no necesitan ser escritas entre comillas.

Se puede combinar código razor con código Html.

Este ejemplo muestra como presentar un mensaje en pantalla con código razor.

```
@{  
    var mensaje = "Hola Razor";  
}  
<!DOCTYPE html>  
<html>  
<head>  
    <meta name="viewport" content="width=device-width" />  
    <title></title>  
</head>  
<body>  
    <div>  
        <p>@mensaje</p>  
    </div>  
</body>  
</html>
```

Resultado al ejecutar el código.



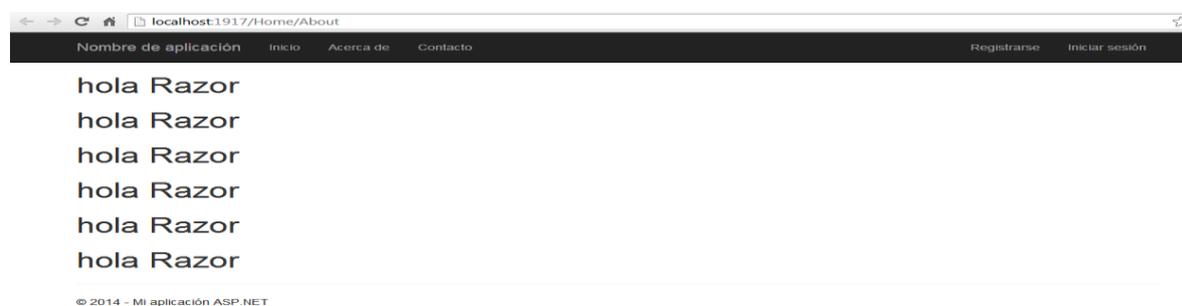
**Ilustración 47.** Resultado al ejecutar vista.

**Fuente:** (Gómez, 2014)

Agregando una sentencia “for” con Razor.

```
@{
    var mensaje = "hola Razor";
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
</head>
<body>
    <div>
        @for (int i = 0; i <= 5; i++)
        {
            <h1>@mensaje</h1>
        }
    </div></body></html>
```

Resultado al ejecutar el código.



**Ilustración 48.** Resultado al ejecutar vista.

**Fuente:** (Gómez, 2014)

Añadir una sentencia “if” y “foreach” con Razor.

```
@{
    var mensaje = "hola Razor";
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
</head>
<body>
    <div>
        @for (int i = 0; i <= 5; i++)
        {
            if(i == 10)
            {
                <h1>@mensaje</h1>
            }
        }
        @foreach (var a in Request.ServerVariables)
        {
            <h1>@mensaje</h1>
        }
    </div>
</body>
</html>
```

### 2.5.3. Pasar información de vista a controlador.

*“Existen tres formas de pasar información de un controlador a una vista, como objeto de modelo inflexible de tipos, como un tipo dinámico (usando @ modelo dinámico), uso de la ViewBag.” (Asp.Net, 2013)*

Para comprender, como pasar información entre vista y controlador, crear tres vistas, la primera tiene un link que realiza una petición al controlador para mostrar la segunda vista que contiene un formulario con datos de nombre y apellido, estos datos serán enviados al

controlador y devueltos a la vista mediante ViewBag a una tercera vista, con el fin de apreciar cómo se interactúa entre vista y controlador.

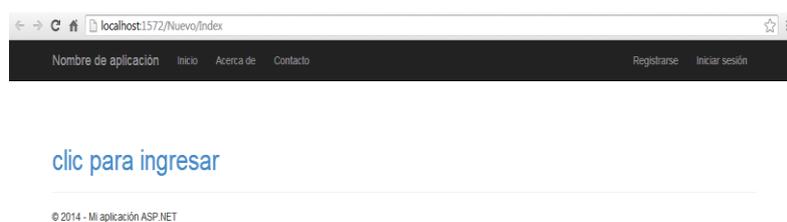
Crear las tres vistas la primera llamada Index.cshtml la segunda formulario.cshtml y guardar.cshtml, además crear un controlador llamado NuevoController.cs.

En el NuevoController agregar una nueva acción llamada Index, esta acción retorna la vista Index.cshtml.

```
public ActionResult Index(){  
  
    return View();  
  
}
```

En la vista Index.cshtml agregar el link que lleva a la vista formulario.cshtml, ejecutar para mostrar el resultado.

```
<h1><a href=" ../Nuevo/formulario">clíc para ingresar</a></h1>
```



### **Ilustración 49.** Resultado al ejecutar vista.

**Fuente:** (Gómez, 2014)

En el NuevoController.cs, agregar una nueva acción llamada formulario, retorna la vista formulario.

```
public ActionResult formulario()
{
    return View();
}
```

En la vista formulario.cshtml, agregar el siguiente código Razor y Html que representa un formulario de ingreso; ejecutar para mostrar el resultado.

```
<form action=" ../Nuevo/Guardar" method="POST">
    Apellido: @Html.TextBox("txtApellido")
    Nombre: @Html.TextBox("txtNombre")
    <input type="submit" value="Guardar" /></form>
```



### Ilustración 50. Resultado al ejecutar una vista.

Fuente: (Gómez, 2014)

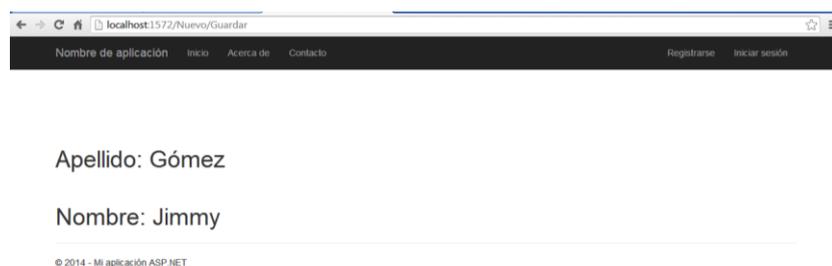
Ahora en el NuevoController se agrega una nueva acción llamada Guardar, esta acción envía los datos a la vista Guardar.cshtml al momento de hacer clic en el botón guardar.

```
public ActionResult Guardar(FormCollection form)
{
    var apellido = form["txtApellido"];
    var nombre = form["txtNombre"];
    ViewBag.Apellido = apellido;
    ViewBag.Nombre = nombre;
    return View();
}
```

Como se puede observar, el método recibe como parámetro un FormCollection para recuperar el valor del “txtApellido” y “txtNombre”, estos valores son almacenados en variables y luego colocados en ViewBag.

En la vista Guardar se agrega el código para obtener los valores de ViewBag, visualizar el resultado.

```
<h1>Apellido: @ViewBag.Apellido</h1>
<br />
<h1>Nombre: @ViewBag.Nombre</h1>
```



### Ilustración 51. Resultado al ejecutar vista.

Fuente: (Gómez, 2014)

Hay que tomar en cuenta que el formulario contiene una llamada a la acción guardar del controlador NuevoContoller.cshtml.

```
<form action=" ../Nuevo/Guardar" method="POST">
```

También puede ser presentado de la siguiente forma que especifica el método y el controlador a ser ejecutado.

```
<form action=@Url.Action("Guardar", "Nuevo")>
```

## 2.6. Controladores y enrutamiento.

Los controladores MVC son encargados de transmitir peticiones en una aplicación ASP.NET MVC. Por ejemplo si se ejecuta en la barra del navegador la url <http://localhost:1572/Nuevo/>, el controlador `NuevoController.cs` es invocado y retorna la respuesta a la petición del navegador; el controlador puede devolver una vista o redirigir a otro controlador.

La clase principal de los controladores es la clase `ControllerBase` que proporciona el control general de MVC, esta contiene la clase `Controller` que es encargada de las siguientes funciones:

- Encontrar el método de acción adecuado en el controlador.
- Obtener los argumentos del método de acción.
- Corregir los errores durante la ejecución de los métodos de acción.
- Representar los tipos de página ASP.NET (vistas).

Algo importante por recordar es que todas las clases de controlador deben llevar el sufijo “Controller”.

```
[HandleError]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        ViewData["Message"] = "Bienvenido a ASP.NET MVC!";
    }
}
```

```
return View();  
}  
public ActionResult About()  
{  
    return View();  
}
```

### 2.6.1. Acciones en un controlador.

Los controladores contienen acciones que son ejecutados al momento de invocar un controlador. Por ejemplo si se realiza una solicitud al navegador con la siguiente url <http://localhost:1572/Nuevo/Index>, el método Index() del controlador Nuevo es ejecutado.

En el siguiente ejemplo se muestra el método de acción de un controlador que devuelve un mensaje.

```
public class MyController : Controller  
{  
    public ActionResult HelloWorld()  
    {  
        ViewData["Message"] = "Hola mundo.";  
        return View();  
    }  
}
```

### 2.6.2. Action Result.

Una acción de un controlador puede devolver una solicitud del explorador, esta respuesta es llamada Action Result.

ASP.NET MVC soporta varios tipos de Action Result, incluyendo:

**Tabla 7.** Controladores y enrutamiento.

Action Result	Método auxiliar	Descripción
ViewResult	View	Retorna una vista como una página web.
PartialViewResult	PartialView	Retorna una sección de una vista que se puede representar dentro de otra vista.
RedirectResult	Redirect	Re-direcciona a otro método de acción utilizando su dirección URL.
RedirectToRouteResult	RedirectToAction RedirectToRoute	Re-direcciona a otro método de acción.
ContentResult	Content	Devuelve un tipo de contenido definido por el usuario.
JsonResult	Json	Devuelve un objeto JSON serializado.
JavaScriptResult	JavaScript	Devuelve un script que se puede ejecutar en el cliente.
FileResult	File	Devuelve la salida binaria para escribir en la respuesta.
EmptyResult	(Ninguno)	Representa un valor devuelto que se utiliza si el método de acción debe devolver un resultado <b>null</b> (vacío).

Fuente: (Developer\_Network, Controladores y metodos de acción en apliccaiones MVC., 2014)

### 2.6.3. Marcar métodos públicos como métodos de no acción.

En un controlador automáticamente todos los métodos son tratados con acciones, es posible marcar métodos públicos como métodos de no acción en caso de que se lo requiera, especificando el atributo NonAction de la siguiente manera.

```
[NonAction]
private void Accion()
{
}
```

#### 2.6.4. Creación de rutas personalizadas.

La tabla de rutas funciona bien en las aplicaciones ASP.NET MVC, sin embargo la tabla puede ser personalizada dependiendo de las necesidades de enrutamiento.

Por ejemplo, si en una aplicación se desea visualizar las peticiones entrantes en el siguiente formato / Archivo/18-07-2014; en el archivo Global.asax crear una ruta personalizada de la siguiente manera.

```
routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.MapRoute(
    "Ruta",
    "Archivo/{entryDate}",
    new { controller = "Archivo", action = "Accion" }
);
```

En la ruta personalizada se especifica: el nombre de la ruta, formato de presentación, controlador y acción que invoca la ruta.

El orden de las rutas es importante, la ruta personalizada debe ir antes de la ruta por defecto, ya que será la primera en ser ejecutada.

El archivo Global.asax queda de la siguiente manera.

```

using System.Web.Mvc;
using System.Web.Routing;
namespace MvcApplication1
{
    public class MvcApplication : System.Web.HttpApplication
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
            routes.MapRoute(
                "Ruta",
                "Archivo/{entryDate}",
                new { controller = "Archivo", action = "Accion" }
            );
            routes.MapRoute(
                "Default",
                "{controller}/{action}/{id}",
                new { controller = "Home", action = "Index", id = "" }
            );
        }
        protected void Application_Start()
        {
            RegisterRoutes(RouteTable.Routes);
        }
    }
}

```

## 2.7. Validaciones.

En esta parte del documento se explica cómo añadir lógica de validación en la capa de modelo, estas reglas de validación se aplican cada vez que un usuario intente crear o editar campos en una vista.

Para escribir validaciones en la capa de modelo es indispensable utilizar la librería de ensamblado (System.ComponentModel.DataAnnotations) namespace que contiene System.Web. DataAnnotations proporciona un conjunto integrado de validación de atributos que se pueden declarar en una clase o propiedad.

Algunas de las anotaciones de validación más importantes:

**Tabla 8.** Anotaciones

Anotación	Descripción
CreditCardAttribute	Especifica que el valor del campo de datos sea un número de tarjeta de crédito válido
DisplayFormatAttribute	Especifica el modo en que los datos dinámicos de ASP.NET muestran y dan formato a los campos de datos.
EditableAttribute	Indica si un campo de datos es modificable o no modificable.
EmailAddressAttribute	Valida una dirección de correo electrónico.
FileExtensionsAttribute	Valida extensiones
KeyAttribute	Valida una o varias propiedades que identifican exclusivamente una entidad.
MaxLengthAttribute	Especifica la longitud máxima de un atributo del modelo
MetadataTypeAttribute	Especifica la clase de metadatos que debe asociarse a una clase de modelo de datos.
MinLengthAttribute	Especifica la longitud mínima del campo de modelo.
PhoneAttribute	Especifica que el campo de modelo sea un número telefónico válido
RangeAttribute	Especifica que el atributo esté dentro de un rango establecido.
RegularExpressionAttribute	Especifica que un valor de campo de datos sea válido dependiendo de la expresión regular ingresada.
RequiredAttribute	Especifica que el campo de datos sea requerido.
StringLengthAttribute	Especifica la longitud mínima y máxima en el campo de datos.
UrlAttribute	Proporciona validación de la dirección URL.

Fuente: (Developer\_Network, DataAnnotations, 2014)

Formato para utilizar anotaciones.

```
[anotación]
Declaración de atributo
```

Ejemplo:

```
[StringLength(60)]
public string nombres { get; set; }
```

La clase “Persona.cs”, modificada con las anotaciones de validación.

```
public class Persona
{
    [Required]
```

```

public int id { get; set; }
[RegularExpression(@"^[A-Z]+[a-zA-Z"]*\s*$")]
public string nombres { get; set; }
[RegularExpression(@"^[A-Z]+[a-zA-Z"]*\s*$")]
public string apellidos { get; set; }
[RegularExpression(@"^[A-Z]+[a-zA-Z"]*\s*$")]
[StringLength(10)]
public string direccion { get; set; }
[PhoneAttribute()]
public int telefono { get; set; }}

```

Ejecutar la aplicación y probar las validaciones.

## Create

Persona

nombres	<input type="text" value="9999"/>	El campo nombres debe coincidir con la expresión regular "[A-Z]+[a-zA-Z"]*\s*\$".
apellidos	<input type="text" value="999999"/>	El campo apellidos debe coincidir con la expresión regular "[A-Z]+[a-zA-Z"]*\s*\$".
direccion	<input type="text" value="988777"/>	El campo direccion debe coincidir con la expresión regular "[A-Z]+[a-zA-Z"]*\s*\$".
telefono	<input type="text" value="uiuiuiui"/>	

[Back to List](#)

© 2014 - Mi aplicación ASP.NET

### Ilustración 52. Formulario de validación.

Fuente: (Gómez, 2014)

## 2.8. Características para móviles de ASP.NET MVC 5.

*“Debido a la creciente popularidad y variedad de Smartphone y tabletas móviles, resulta imprescindible que las aplicaciones basadas en ASP.NET MVC cuenten con interfaces móviles agradables y útiles”.* (Microsoft, MSDN Magazine, 2014)

HTML5 y CSS3 se han convertido rápidamente en estándares para el desarrollo web gracias a la compatibilidad con la mayoría de exploradores web y por su capacidad de presentar interfaces amigables para dispositivos móviles.

Muchos Frameworks como Bootstrap y JQueryMobile trabajan con HTML5 y CSS3 facilitando el diseño de interfaces amigables para dispositivos móviles.

### **2.8.1. Bootstrap.**

Es un Framework de diseño web que mejora la presentación de las aplicaciones dándoles modelos elegantes y responsivos.

Bootstrap 3 es la última versión del Framework donde se nota la madurez que ha adquirido, para obtener más información visitar el sitio web oficial de Bootstrap <http://getbootstrap.com/>.

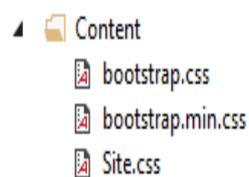
ASP.NET MVC en su quinta versión, incluye por defecto las librerías de Bootstrap, lo que permite trabajar automáticamente con este Framework, en caso de no tener Bootstrap descargar en el administrador de paquetes Nuget.



**Ilustración 53.** Administrador de paquetes Nuget.

Fuente: (Gómez, 2014)

Una vez instalada la librería se puede observar que los archivos <sup>28</sup>JavaScript y <sup>29</sup>Css son agregados en la solución.

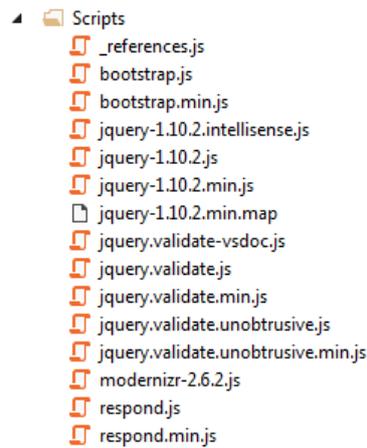


**Ilustración 54.** Archivos CSS.

Fuente: (Gómez, 2014)

<sup>28</sup> **JavaScript:** Lenguaje de programación interpretado utilizado del lado del cliente

<sup>29</sup> **Css:** hoja de estilos en cascada, es un lenguaje para definir los estilos en una aplicación web.



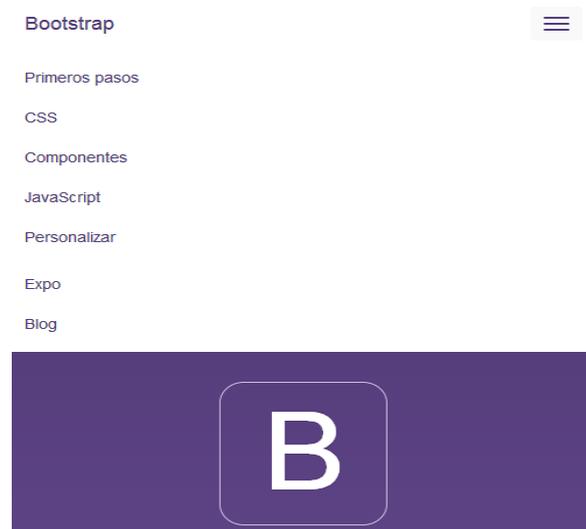
### Ilustración 55. Archivos JavaScript.

Fuente: (Gómez, 2014)

Ahora solo queda referenciar las librerías de Bootstrap en el Layout del proyecto MVC 5.

```
@Styles.Render("~/Content/css/bootstrap.css")  
@Scripts.Render("~/Scripts/bootstrap.js")
```

Página responsiva diseñada con Bootstrap.



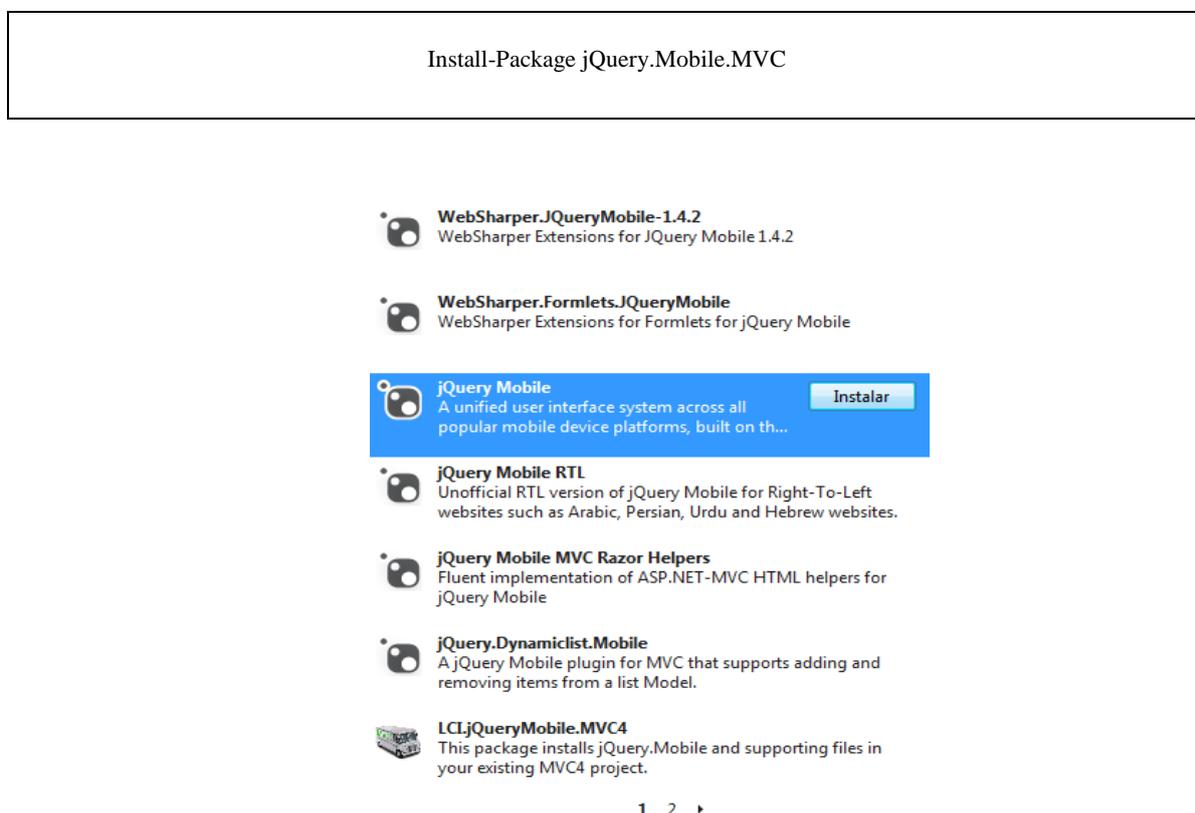
### Ilustración 56. Plantilla Bootstrap.

Fuente: (Gómez, 2014)

## 2.8.2. JQuery Mobile.

JQuery Mobile es una biblioteca de JavaScript y CSS que proporciona el diseño de interfaces que funcionan en todos los principales navegadores móviles, la última versión estable de JQuery Mobile es la 1.4.2, para obtener más información visitar su página oficial [jquerymobile.com](http://jquerymobile.com)

Descargar las librerías de `jQuery.Mobile.MVC` en el administrador de paquetes Nuget e instalarlo, o desde la consola de administración de paquetes introducir el siguiente comando.



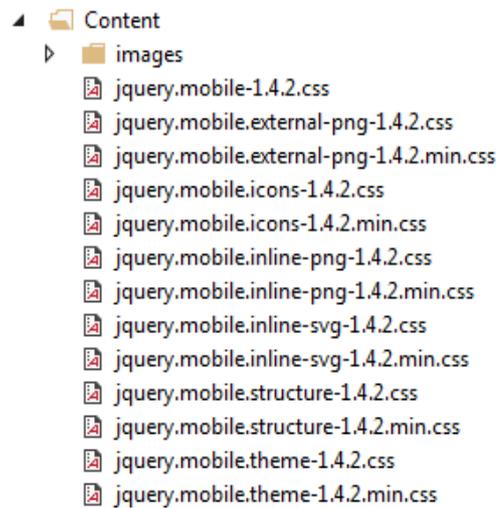
**Ilustración 57.** Administrador de paquetes Nuget.

Fuente: (Gómez, 2014)

El paquete `jQuery.Mobile.MVC` instala lo siguiente:

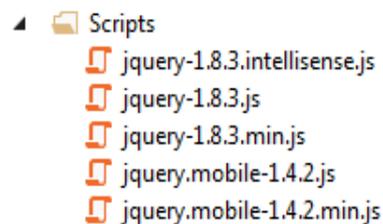
- `BundleMobileConfig.cs` archivo de configuración de las referencias CSS Y JavaScript.

- Archivos de JQuery Mobile CSS.
- Archivos de JavaScript JQuery Mobile.
- Un archivo de diseño de estilo Mobile JQuery (Layout.Mobile.cshtml).



**Ilustración 58.** Archivos CSS.

Fuente: (Gómez, 2014)



**Ilustración 59.** Archivos JavaScript.

Fuente: (Gómez, 2014)

Abrir el archivo Global.asax y agregar el siguiente código, servirá para configurar JQueryMobile en la aplicación MVC 5.

```
BundleMobileConfig.registerBundles(BundleTable.Bundles);
```

Referencias a los archivos CSS y JavaScript.

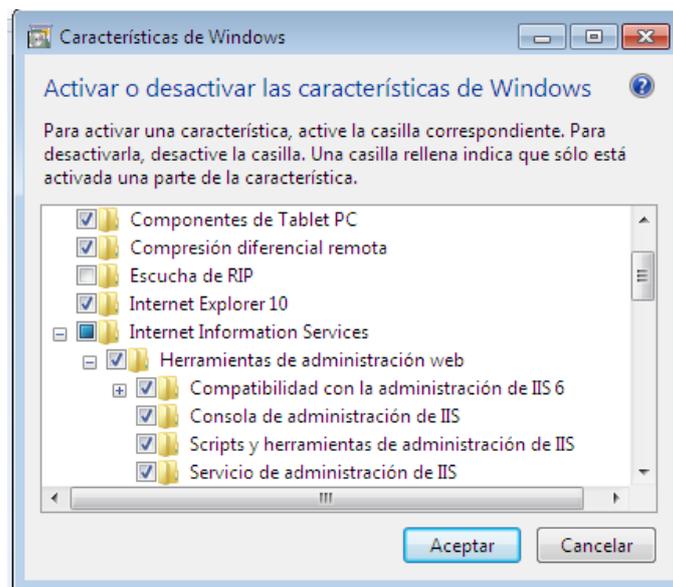
```
@Styles.Render("~/Content/css/jquery.mobile-1.4.2.css")
@Scripts.Render("~/Scripts/jquery.mobile-1.4.2.js")
```

## 2.9. Publicar en IIS.

Para publicar aplicaciones web hechas con ASP.NET MVC, previamente instalar <sup>30</sup>IIS y <sup>31</sup>Web Deploy. Web Deploy se incluye por defecto con la instalación de Visual Studio, en caso de querer instalarlo manualmente descargarlo de la página oficial de IIS.

<http://www.iis.net/downloads/microsoft/web-deploy>

Para trabajar con IIS es necesario activarlo; en el Panel de control, programas y características, activar o desactivar características de Windows, activar la casilla correspondiente a IIS tal como se muestra en la figura.



**Ilustración 60.** Características de Windows.

**Fuente:** (Gómez, 2014)

<sup>30</sup> **IIS:** Internet information server, es un sistema administrador de servicios web.

<sup>31</sup> **Web Deploy:** Administrador de despliegues de páginas web en internet information server.

Posteriormente activar las extensiones de ASP.NET para poder publicar páginas web ASP.NET en IIS. Expandir los nodos “Servicios World Wide Web” y “Características de desarrollo de aplicaciones”, activar la casilla “ASP.NET 4.5”. Tal como se muestra en la figura.



**Ilustración 61.** Activar y desactivar características de Windows.

Fuente: (Gómez, 2014)

Abrir IIS y verificar si en el grupo de aplicaciones la versión del R Framework .Net está en v4.0.

 Grupos de aplicaciones

Esta página permite ver y administrar la lista de grupos de aplicaciones del servidor. Los grupos de aplicaciones están asociados a procesos de trabajo, contienen una o más aplicaciones, y proporcionan aislamiento entre aplicaciones.

Nombre	Estado	Versión de .NET Framework	Modo de canal...	Identidad	Aplicaciones
ASP.NET v4.0	Iniciado	v4.0	Integrada	ApplicationPoolId...	0
ASP.NET v4.0 Cl...	Iniciado	v4.0	Clásica	ApplicationPoolId...	0
Classic .NET Ap...	Iniciado	v4.0	Clásica	ApplicationPoolId...	0
cotizar	Iniciado	v4.0	Integrada	ApplicationPoolId...	0
DefaultAppPool	Iniciado	v4.0	Integrada	ApplicationPoolId...	2
documentos_el...	Iniciado	v4.0	Integrada	ApplicationPoolId...	0
mvc	Iniciado	v4.0	Integrada	ApplicationPoolId...	0
vh_cotizaciones	Iniciado	v4.0	Integrada	ApplicationPoolId...	0

**Ilustración 62.** Grupo de aplicaciones IIS.

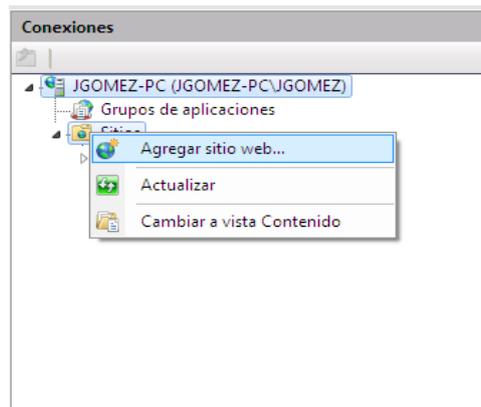
Fuente: (Gómez, 2014)

En caso de que la versión del Framework .Net no esté en v4.0, instalar el Framework .Net; Abrir el símbolo de sistema en modo administrador, a continuación ejecutar aspnet\_regiis.exe instalar ASP.NET en IIS 4, escribiendo los siguientes comandos.

```
cd Windows\Microsoft.Net\Framework\v4.0.30319 aspnet_regiis.exe -i
```

Una vez que se configuró el entorno de IIS, se procede a publicar la aplicación web, para lo cual crear una carpeta llamada “mvc” donde se almacena los archivos de la aplicación web en la siguiente dirección C:\mvc.

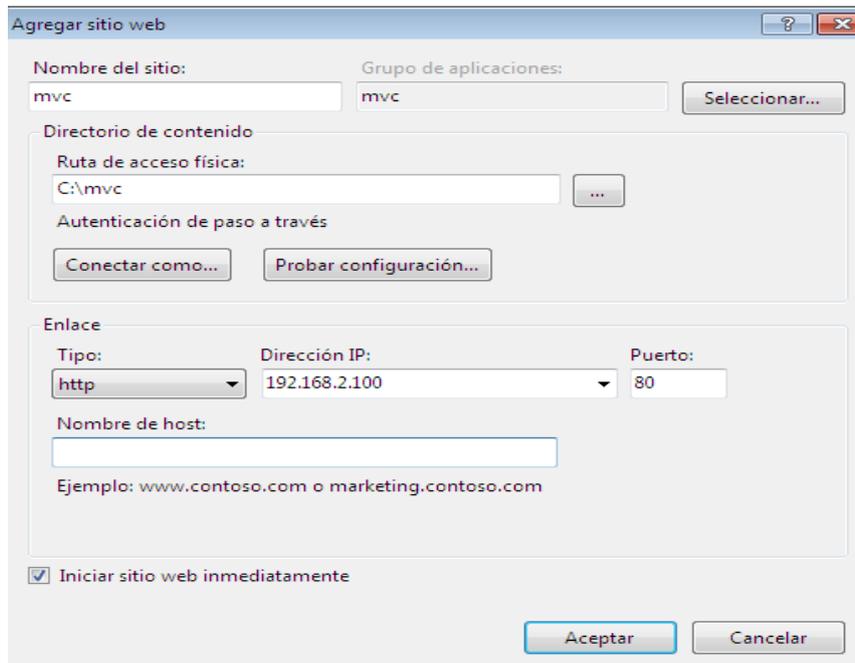
En el administrador de IIS, seleccionar el nodo Sitios, agregar nuevo sitio web.



**Ilustración 63.** Agregar sitio Web.

Fuente: (Gómez, 2014)

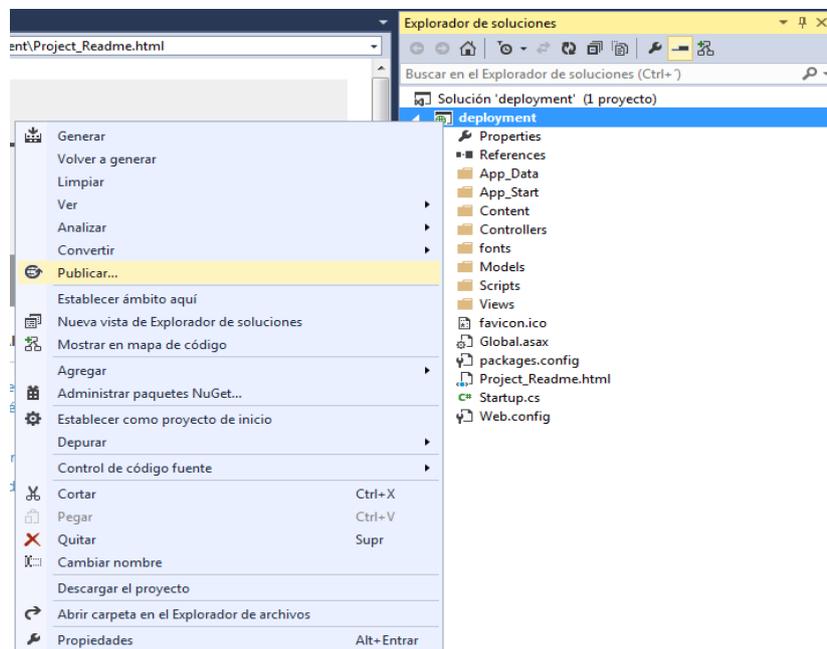
Configurar los parámetros de publicación: nombre del sitio web, ruta de acceso física, protocolo, dirección IP de servidor y el puerto de publicación, tal como se muestra en la figura.



**Ilustración 64.** Configuración de nuevo sitio web.

Fuente: (Gómez, 2014)

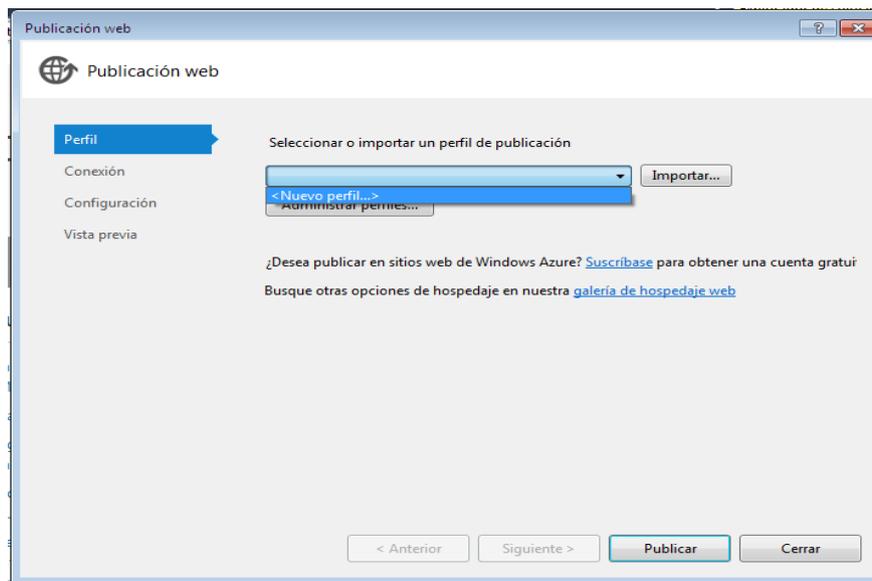
En Visual Studio 2013 seleccionar el proyecto que se desea deployar, clic en publicar.



**Ilustración 65.** Publicar sitio web.

Fuente: (Gómez, 2014)

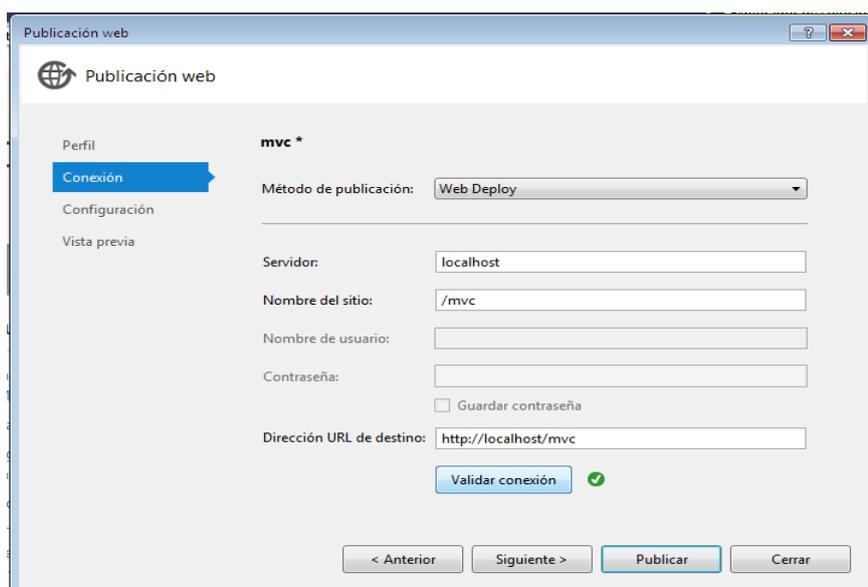
Agregar un nuevo perfil de publicación.



**Ilustración 66.** Crear nuevo perfil.

Fuente: (Gómez, 2014)

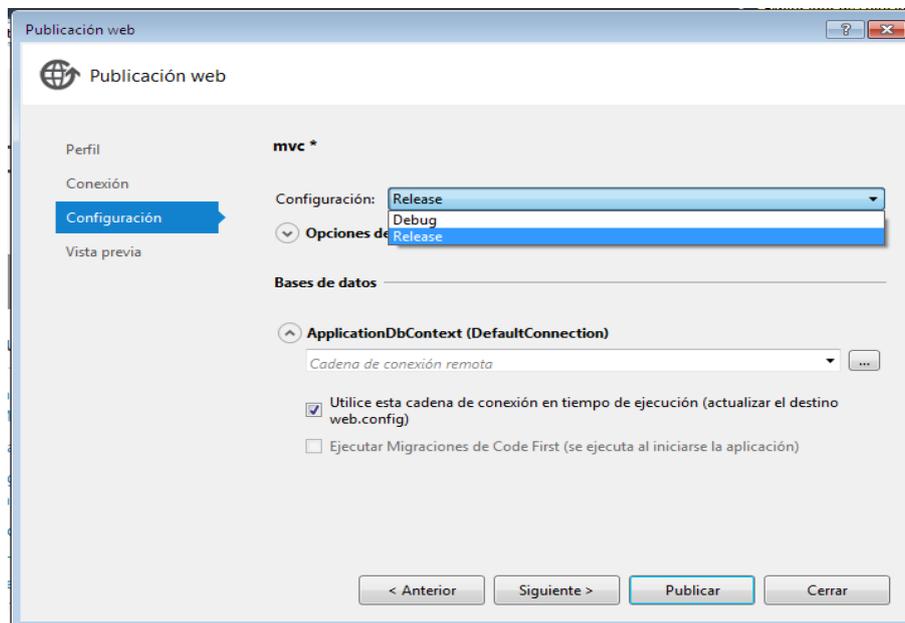
Configurar los parámetros de publicación: método de publicación se selecciona Web Deploy, servidor de alojamiento de la aplicación, nombre del sitio y dirección url de destino, tal como se muestra en la figura.



**Ilustración 67.** Configurar publicación.

Fuente: (Gómez, 2014)

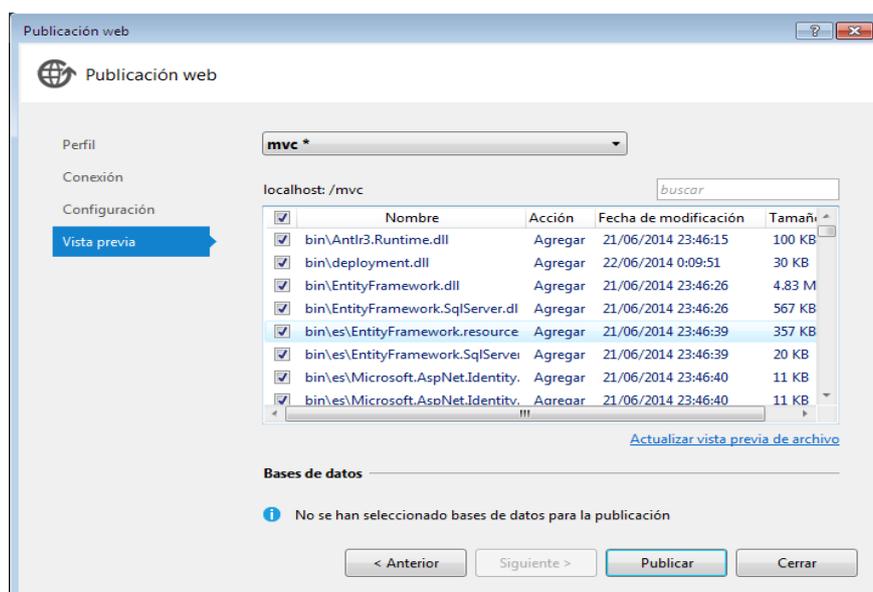
Escoger el método de publicación en este caso “Release” y si existe una conexión con base de datos configurarla tal como se realiza en visual Studio 2013.



**Ilustración 68.** Tipo de compilación.

Fuente: (Gómez, 2014)

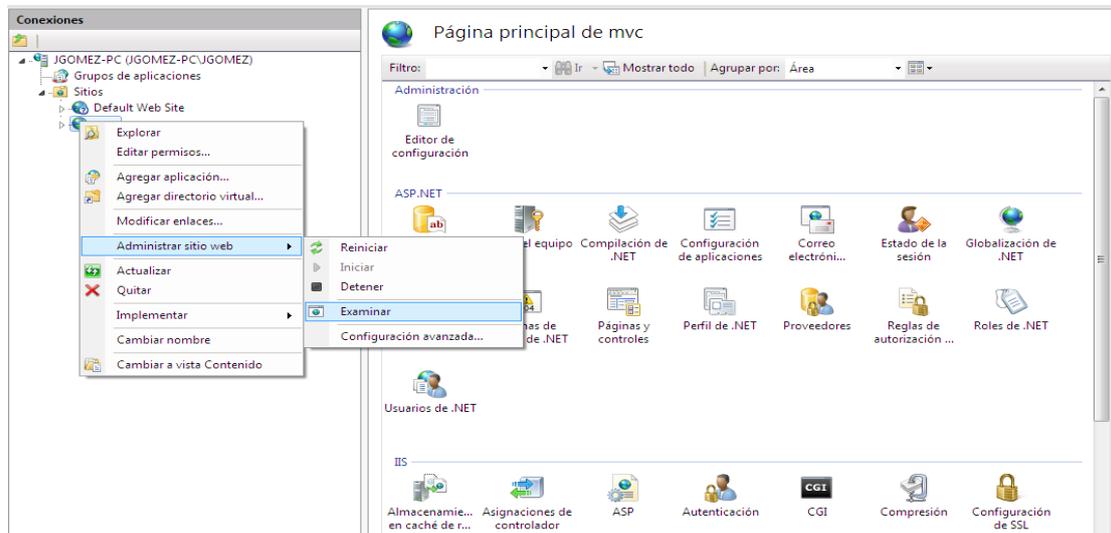
Examinar los archivos que se van a publicar y posteriormente clic en publicar.



**Ilustración 69.** Archivos publicados.

Fuente: (Gómez, 2014)

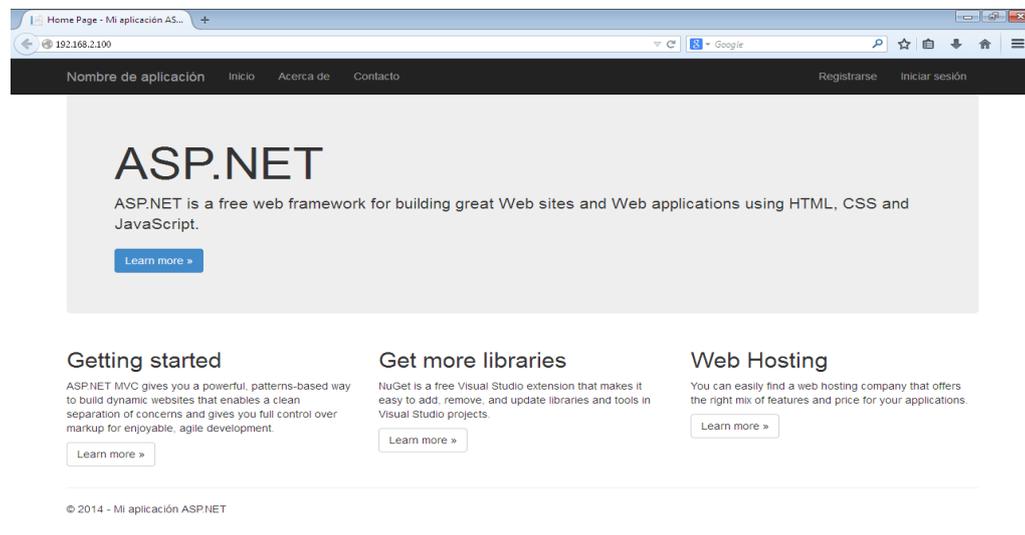
En el administrador de IIS, clic derecho en la aplicación a publicar, seleccionar administrador de sitio web y posteriormente examinar.



**Ilustración 70.** Examinar sitio web.

Fuente: (Gómez, 2014)

Revisar el resultado de publicar la aplicación.



**Ilustración 71.** Sitio web publicado.

Fuente: (Gómez, 2014)

## CAPÍTULO III

### 3. Diagnóstico de situación actual y Marco propositivo del Framework ASP.NET MVC.

#### 3.1. Situación actual del FRAMEWORK ASP.NET MVC.

*En marzo del 2009 se hizo pública la primera versión de ASP.NET MVC, presentada como una alternativa más ligera que ASP.NET tradicional (Web Forms).*

*“El uso de ASP.NET MVC ha crecido a un ritmo de alrededor de 5% por año, y ahora se sitúa en 30%. Esencialmente por enfocar tanto esfuerzo de mercadotecnia en MVC, Microsoft realmente creó una demanda de mercado más grande para él a diferencia de los otros productos ofrecidos por Microsoft.” (Walker, 2013)*

ASP.NET MVC ha llegado a tener gran aceptación en el mercado y ha alcanzado una etapa de madurez, por lo que es tratado como un importante producto con mantenimiento a futuro en el que Microsoft está gastando la mayoría de recursos de investigación y desarrollo.

Microsoft brinda soporte completo del Framework que se distribuye independientemente y como parte de Visual Studio, además es atendido por los mismos desarrolladores del equipo de ASP.NET por lo que el Framework sigue desempeñando un papel vital en la entrega de soluciones a clientes.

Actualmente ASP.NET MVC es distribuido bajo licencia de código abierto Apache 2.0, esto permite que el modelo de desarrollo sea más abierto en el que todos los programadores pueden proporcionar retroalimentación y chequeo de código, corregir errores, desarrollar y probar nuevas funciones usando la información más actualizada de la versión de código fuente y pruebas; esto con el objetivo de construir un circuito de retroalimentación con más fuerza en

los desarrolladores para entregar productos con mejores resultados y de mejor calidad. Se ha integrado con un gran número de tecnologías de código abierto, entre las más importantes destacan: JQuery, JQuery UI, JQuery Mobile, JQuery Validation, Modernizr.js, NuGet, Knockout.js y JSON.NET, todos estos de alta compatibilidad con el Framework.

En la actualidad ASP.NET MVC goza una amplia compatibilidad con proveedores de controles web como: Infragistics, DevExpress y ComponentOne; dedicados al desarrollo de interfaces web enriquecidas, ofreciendo mayor productividad y mejor experiencia de usuario a través de AJAX por lo que la demanda sigue siendo fuerte y el mercado de controles web de terceros continua prosperando, trabajando conjuntamente con el Framework.

Microsoft ha optado por promocionar al Framework mediante conferencias y congresos organizados por lo general cada año donde importantes programadores de Microsoft exponen las bondades y las nuevas sorpresas que trae con cada nueva versión el Framework.

ASP.NET MVC 5 es la versión más reciente del Framework lanzada oficialmente en el mes de noviembre del 2013, está disponible exclusivamente en Visual Studio 2013, esta versión presenta nuevas novedades importantes donde se logra apreciar el nivel de madurez del Framework.

“Las principales novedades presentadas por ASP.NET MVC 5 son:

- *“Soportado por ONE ASP.NET.*
- *Alta compatibilidad con ASP.NET Identity.*
- *Plantillas de página basadas en Bootstrap 3.*
- *Nuevos filtros de autenticación (Authentication filters).*
- *Routing basado en atributos.” (Asp.Net, 2013)*

### 3.2. ASP.NET MVC vs ASP.NET WEB FORMS

ASP.NET MVC y Web Forms son dos Frameworks web desarrollados por Microsoft, simplemente son dos filosofías diferentes para conseguir lo mismo páginas web, el problema es que Microsoft ha posicionado a los dos Framework como incompatibles entre sí, esto ocasiona que programadores web se sientan obligados a decidir entre uno u otro.

*“Microsoft continúa invirtiendo tanto en Web Forms y MVC pero están enfocando tanta publicidad alrededor de MVC que parecía que Web Forms estaba obsoleto. Como resultado, MVC comenzó a ganar cuota en el mercado.”* (Walker, 2013) Pero no quiere decir que Web Forms desaparezca, Microsoft sigue invirtiendo recursos para Web Forms.

ASP.NET MVC propone una forma distinta de desarrollar aplicaciones Web, en este no existe conceptos como: mantenimiento de estado en el viewstate, postback ni los controles de servidor basados en estas características; sin embargo, dado que el Framework está construido bajo ASP.NET, es posible utilizar características como: páginas maestras, vistas .aspx, manejar los lenguajes C# y VB.NET.

*“La mayor parte de los beneficios se puede lograr en Web Forms con una arquitectura sólida y prácticas de codificación responsables. Separación limpia en capas, pruebas unitarias y el control directo sobre los resultados de la página son posibles en el modelo de Web Forms”*  
(Walker, 2013)

ASP.NET MVC requiere un conocimiento más profundo del entorno web, puesto que requiere un control más riguroso sobre la información que se envía y recibe del cliente web, mantener el estado entre peticiones, maquetar páginas HTML, crear hojas de estilo, trabajar con scripts.

Al momento de decidir que Framework utilizar para el desarrollo de una aplicación web, es necesario determinar ciertas métricas donde cada uno de los Frameworks sean puestos a prueba, a continuación se plantea las métricas que son factores de análisis para los dos Frameworks.

### **3.2.1. Curva de aprendizaje.**

En ASP.NET MVC la curva de aprendizaje para un desarrollador es muy alta especialmente si nunca se ha desarrollado para web, puesto que requiere conocimientos de la arquitectura MVC como tal, además un dominio perfecto de HTTP, HTML y JavaScript.

En Web Forms la curva de aprendizaje es pequeña, se puede crear aplicaciones web con muy poco conocimiento de HTML y JavaScript, una de las ventajas de Web Forms es que se permite programar para web utilizando el mismo paradigma de aplicaciones de escritorio, basándose en formularios y eventos, esto ayuda a un rápido aprendizaje del Framework principalmente para desarrolladores que anteriormente han venido desarrollando aplicaciones de escritorio.

### **3.2.2. Escalabilidad.**

Ya que tanto ASP.NET MVC y Web Forms son Frameworks de la plataforma .Net proporciona ventajas como productividad, potencia, seguridad y escalabilidad. Sin embargo, la arquitectura, la reutilización de código, el rendimiento, control total del HTML y características de URL Routing hacen que las aplicaciones desarrolladas bajo ASP.NET MVC sean mucho más escalables que las desarrolladas con Web Forms.

### **3.2.3. Compatibilidad.**

Web Forms y MVC son tecnologías muy distintas. Sin embargo, estos dos Frameworks están contruidos bajo el marco ASP.NET, esto permite que todas las funciones de ASP.NET estén disponibles para los dos Frameworks, esto incluye características como la pertenencia, autenticación, roles, clases e interfaces.

Además Web Forms y MVC están disponibles para los lenguajes C# y VB.NET y gozan de una alta compatibilidad con otros Frameworks de desarrollo web como JQuery, JQuery Mobile, AJAX, Bootstrap, JQuery UI, JQuery Validation, Modernizr.js, NuGet, Knockout.js, Entity Framework y JSON.NET.

MVC permite trabajar con motores de vista como Razor y Aspx, mientras Web Forms se encuentra disponible solo para Aspx, dándole un punto a favor a MVC por mayor compatibilidad con otros Frameworks.

### **3.2.4. Rendimiento.**

Las aplicaciones desarrolladas con ASP.NET MVC son mucho más livianas que las desarrolladas con Web Forms ya que la arquitectura con la que está diseñada ASP.NET MVC permite controlar cada byte enviado y recibido del cliente permitiendo un rendimiento óptimo y velocidad de ejecución.

Web Forms para que pueda reconstruir una página web debe enviar una serie de datos llamados viewstate entre el cliente y el servidor, este viewstate tiene a crecer mucho afectando el rendimiento y velocidad de ejecución de la aplicación.

### **3.2.5. Tendencia de uso.**

Microsoft ha destinado muchos recursos en publicidad de ASP.NET MVC, según Microsoft el uso de MVC ha crecido de alrededor de 5% por año, gracias al todo el esfuerzo en mercadotecnia alrededor de MVC, creando una demanda en el mercado mucho más grande para él. Esto llevó a que los desarrolladores crean que Web Forms es una tecnología obsoleta. Como resultado de esta publicidad, abrió una disputa entre que Framework es mejor.

Es verdad que por toda la publicidad que tiene ASP.NET MVC, muchos programadores han optado por usar MVC para sus desarrollos, pero todavía existe una gran comunidad bien consolidada de programadores que utilizan Web Forms por su madurez, su facilidad y rapidez de desarrollo.

### **3.2.6. Soporte técnico.**

Microsoft brinda soporte técnico completo tanto para Web Forms y MVC mediante la difusión de información en sitios oficiales: [www.msdn.com](http://www.msdn.com) y [www.asp.net.com](http://www.asp.net.com). La ventaja que tiene ASP.NET MVC es que en actualidad es distribuido bajo licencia de código abierto en el que todos los desarrolladores pueden proporcionar retroalimentación, chequeo de código, corrección de errores y soporte técnico mediante el repositorio de alojamiento de proyectos de software de Microsoft [www.codeplex.com](http://www.codeplex.com).

### **3.2.7. Fácil desarrollo.**

Web Forms cuenta con controles de servidor ricos, modelo de eventos y ViewState que aumenta la velocidad y la facilidad de desarrollo, además permite desarrollar aplicaciones sin una sola línea de código bajo el paradigma de desarrollo rápido de aplicaciones RAD, mientras que ASP.NET MVC para su utilización necesita un conocimiento profundo de la arquitectura MVC y de programación web lo que hace tardío la construcción de sistemas web.

### **3.2.8. Pruebas unitarias.**

En ASP.NET MVC los controles son clases separadas, por lo que hace posible hacer pruebas unitarias fácilmente de manera automática, en Web Forms el archivo que contiene código al ejecutarse el servidor termina siendo muy pesado conteniendo muchos manejadores de eventos, lo cual hace imposible las pruebas unitarias automáticas.

### **3.2.9. Control de HTML.**

En ASP.NET MVC no existen los controles del lado del servidor, la única opción que tiene el programador es utilizar los controles HTML, permitiendo todo el control sobre el HTML y una alta integración con librerías JavaScript, lo mismo que no pasa con Web Forms que tiene un limitado control sobre el HTML, por la utilización de controles del lado del servidor que genera código ineficiente, pesado y no cumple con los estándares de desarrollo web.

### **3.2.10. Arquitectura.**

ASP.NET MVC tiene embebido la arquitectura MVC en la cual está fuertemente implementada la separación de responsabilidades lo cual hace fácil su mantenimiento y reutilización del código, permite que el rendimiento y escalabilidad sea elevado.

Web Forms tiene una arquitectura basada en páginas, esta arquitectura combina una página de código HTML con un archivo de código que agrega la funcionalidad, esta estructura facilita la creación de páginas que implementan tareas de interacción entre el cliente y el servidor.

### 3.3. Cuadro comparativo.

Para realizar la comparación, se ha realizado un cuadro comparativo donde se evalúan los dos Frameworks en cada una de las métricas, valorados entre 0 y 4 donde 4 es la valoración más alta.

**Tabla 9.** Comparativa MVC Y Web Forms.

Métrica	ASP.NET MVC	ASP.NET WEB FORMS
Curva de aprendizaje	1	4
Escalabilidad	4	2
Compatibilidad	4	3
Rendimiento	4	2
Tendencia de uso	3	3
Soporte técnico	4	4
Fácil desarrollo	1	4
Pruebas unitarias	3	1
Control de HTML	4	3
Arquitectura	4	2
<b>Total</b>	<b>32</b>	<b>28</b>

Fuente: (Gómez, 2014)

### 3.4. Conclusión.

Según la evaluación realizada en el cuadro comparativo posiciona al Framework ASP.NET MVC con 32 puntos como mejor opción de desarrollo por encima de los 28 puntos de Web Forms.

La ventaja de ASP.NET MVC radica básicamente en la escalabilidad, compatibilidad, rendimiento, soporte técnico, control sobre el HTML y arquitectura; pero esto no quiere decir que esta tecnología reemplaza a la otra, ambas tecnologías son muy buenas, todo depende de los

requerimientos, el tipo de proyecto que se quiere desarrollar y las habilidades que presente el equipo de desarrollo, ya que Web Forms posee una alta productividad, flexibilidad y madurez en el mercado, ideal para desarrollar aplicaciones web de manera rápida con una curva de aprendizaje pequeña.

### **3.5. Marco propositivo de ASP.NET MVC**

ASP.NET MVC en un futuro seguirá teniendo soporte completo y mejoras en el producto por parte del equipo de ASP.NET, lo que generará nuevas versiones de la herramienta. La versión más cercana a lanzarse MVC 6 que contará con nuevas funcionalidades como:

La unificación de MVC, <sup>32</sup>Web API y Web Pages, con el fin de utilizar ciertos aspectos de cada Framework para resolver distintos problemas de negocio, lo que significa la consolidación de una estrategia de desarrollo unificada de todas las líneas de productos de ASP.NET.

Otra de las novedades importantes que traerá MVC 6 es el desacoplamiento de System.Web, que permitirá un arranque más rápido del servicio, menor consumo de memoria y que no necesite de ISS para ser ejecutado; esto permitirá ejecutar un proyecto MVC 6 en plataformas como MAC y Linux.

En resumen ASP.NET MVC es un Framework con un horizonte lejano y seguirá teniendo un ritmo acelerado de evolución, aumentando el nivel de productividad y muy rentable para desarrollos web profesionales.

---

<sup>32</sup> **Web API:** Interfaz de programación de aplicaciones web.

## CAPÍTULO IV

### **4. Desarrollo de prototipo de sistema de control interno de uso del tiempo de los recursos humanos de CREATEC.**

#### **4.1. Planeación del proyecto.**

##### **4.1.1. Relato del usuario.**

Para aplicar los conocimientos obtenidos en el análisis del Framework ASP.NET MVC, explotar en la práctica las bondades y ventajas del Framework y poder palpar la experiencia de desarrollar aplicaciones con el patrón de modelo MVC, se desarrolla el prototipo de una aplicación web.

El aplicativo consiste de realizar el seguimiento de proyectos y actividades ejecutadas en CREATEC S.A con fin de llevar un control sobre el tiempo invertido por los recursos humanos de la empresa en el desarrollo de los mismos, la aplicación web tendrá la capacidad de:

- Llevar un catálogo de clientes los cuales solicitan el desarrollo de uno o varios proyectos a la empresa.
- Llevar un catálogo del personal de la empresa. El personal es el encargado de ejecutar los proyectos y actividades, debe ser clasificado por su cargo y el departamento al que pertenece, el personal debe tener dos estados: activo en caso de encontrarse trabajando con normalidad en la empresa o inactivo en caso de encontrarse temporalmente fuera de la empresa, especificar la persona o jefe encargado del empleado, el personal debe manejar un usuario y una contraseña los mismos que serán válidos para ingresar a la aplicación.

- Llevar un catálogo de proyectos. Los proyectos deben ser clasificados por el tipo de proyecto, además los proyectos están constituidos por una o varias actividades. Los proyectos pueden estar en estados: pendiente, planificado, iniciado y terminado.
- Llevar un catálogo de actividades que se necesitan para ser ejecutado el proyecto. Las actividades serán clasificadas por tipo de actividad, cada actividad tiene un empleado encargado de ejecutarla; una actividad puede tener una o varias sub actividades, las actividades tienen estados: pendiente, trabajando, terminado.
- Trabajar con una bitácora. En la bitácora se llevará los registros de los tiempos empleados por el personal en la ejecución de las actividades de cada proyecto.
- Subir evidencias. El sistema debe llevar un registro de las evidencias del trabajo realizado en cada actividad.
- Subir entregables. Los entregables son los resultados o el producto final que se entrega al ser desarrollado cierto proyecto o actividad, los cuales previamente deben ser revisados y garantizado su funcionamiento.
- Registro de actividades diarias realizadas por el personal de la empresa que no están relacionadas con un proyecto planificado.
- Reportes y resultados que deben reflejar el tiempo invertido en cada proyecto y actividad.
- Administración de usuarios y auditoria de la aplicación.

Para el desarrollo del aplicativo se seguirá la metodología de construcción de proyectos SCRUM, ya que la misma permite la planificación y el desarrollo de proyectos de software de manera rápida con excelentes resultados, facilita la organización del trabajo en equipo y se integra perfectamente con la construcción de aplicaciones web siguiendo MVC.

#### 4.1.2. Historias de usuario.

**Tabla 10.** Historias de usuario 1.

Historia de usuario	
Número: 1	Usuario: Administrador de sistema.
Nombre de la historia: catálogo de clientes.	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> El sistema debe llevar un registro de clientes de la empresa, los cuales solicitan el desarrollo de uno o varios proyectos.	

Fuente: (Gómez, 2014)

**Tabla 11.** Historias de usuario 2.

Historia de usuario	
Número: 2	Usuario: Administrador del sistema.
Nombre de la historia: Catálogo del personal.	
Prioridad de negocio: Alta.	
Programador responsable: Jimmy Gómez.	
<b>Descripción:</b> El sistema debe llevar un registro del personal de la empresa. El personal es el encargado de ejecutar los proyectos y actividades de la empresa, debe ser clasificado por su cargo y el departamento al que pertenece, el personal debe tener dos estados: activo en caso de encontrarse trabajando con normalidad en la empresa o inactivo en caso de encontrarse temporalmente fuera de la empresa, especificar la persona o jefe encargado del empleado, el personal debe manejar un usuario y una contraseña los mismos que serán válidos para ingresar a la aplicación.	

Fuente: (Gómez, 2014)

**Tabla 12.** Historias de usuario 3.

Historia de usuario	
Número: 3	Usuario: Jefe de proyecto
Nombre de la historia: Catálogo de proyecto.	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> El sistema debe llevar un registro de proyectos. Los proyectos deben ser clasificados por el tipo de proyecto, además los proyectos están constituidos por una o varias actividades. El proyecto debe tener los estados de pendiente, planificado, iniciado y terminado.	

Fuente: (Gómez, 2014)

**Tabla 13.** Historias de usuario 4.

Historia de usuario	
Número: 4	Usuario: Jefe de proyecto
Nombre de la historia: Catálogo de entregables y evidencias	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> El sistema tendrá la capacidad de llevar el registro de entregables y evidencias de las actividades ejecutadas en cada proyecto.	

Fuente: (Gómez, 2014)

**Tabla 14.** Historias de usuario 5.

Historia de usuario	
Número: 5	Usuario: Jefe de proyecto
Nombre de la historia: Catálogo de actividades.	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> El sistema debe llevar un catálogo de actividades que se necesita para ser ejecutado un proyecto. Las actividades serán clasificadas por tipo de actividad, cada actividad tiene un empleado encargado de ejecutarla; una actividad puede tener una o varias sub actividades, las actividades tienen estados: pendiente, trabajando, terminado.	

Fuente: (Gómez, 2014)

**Tabla 15.** Historias de usuario 6

Historia de usuario	
Número: 6	Usuario: Programador
Nombre de la historia: Bitácora	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> En la bitácora se llevará los registros de los tiempos empleados por el personal en la ejecución de las actividades de cada proyecto.	

Fuente: (Gómez, 2014)

**Tabla 16** Historias de usuario 7

Historia de usuario	
Número: 7	Usuario: Programador
Nombre de la historia: Actividades diarias	
Prioridad de negocio: Alta	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> El sistema llevará el registro de las actividades diarias que no están relacionadas con los proyectos planificados.	

Fuente: (Gómez, 2014)

**Tabla 17** Historias de usuario 8.

Historia de usuario	
Número: 8	Usuario: Gerente
Nombre de la historia: Reportes y resultados.	
Prioridad de negocio: Alta.	
Programador responsable: Jimmy Gómez	
<b>Descripción:</b> Los reportes deben reflejar los resultados y el tiempo invertido en la ejecución de cada proyecto.	

Fuente: (Gómez, 2014)

**Tabla 18** Historias de usuario 8.

<b>Historia de usuario</b>	
<b>Número:</b> 9	<b>Usuario:</b> Administrador del sistema.
<b>Nombre de la historia:</b> Administración de usuarios y auditoria.	
<b>Prioridad de negocio:</b> Media.	
<b>Programador responsable:</b> Jimmy Gómez.	
<b>Descripción:</b> El sistema debe llevar la administración de usuarios y pistas de auditoria.	

Fuente: (Gómez, 2014)

#### 4.1.3. Personas y roles del proyecto.

**Tabla 19** Roles del proyecto.

<b>Persona</b>	<b>Contacto</b>	<b>Rol</b>
Diego Trejo	djtrejo@utn.edu.ec	Scrum Manager
Jorge Jarrín	jjarrin@createc.com.ec	Gestor de producto
Jimmy Gómez	jfgomez@utn.edu.ec	Equipo técnico

Fuente: (Gómez, 2014)

#### 4.1.4. Lista de Objetivos.

- Definición de requerimientos.
- Elaboración de diagrama general de negocio.
- Elaboración del diagrama de procesos.
- Diseño de la base de datos de la aplicación.
- Elaboración de diccionario de datos.
- Maquetación de interfaz de usuario.
- Establecer herramientas de desarrollo.
- Desarrollo de catálogo de clientes.

- Desarrollo de catálogo de personal de la empresa.
- Desarrollo de administración de usuarios.
- Desarrollo de planificación de proyectos y actividades,
- Desarrollo de la Bitácora de registro de los tiempos de proyectos y actividades.
- Desarrollo de evidencias y entregables de los proyectos.
- Desarrollo de actividades diarias de la empresa.
- Desarrollo de resultados y reportes de los proyectos ejecutados.
- Desarrollo auditoria del sistema.
- Ejecución de pruebas del sistema.
- Publicación del sistema.

#### 4.1.5. Pila de producto.

**Tabla 20** Pila de producto.

<b>Id</b>	<b>Requerimiento</b>	<b>Prioridad</b>	<b>Estimación inicial</b>	<b>Factor ajuste</b>	<b>Estimación ajustada</b>
1	Definición de requerimientos.	alta	5	3	8
2	Elaboración de diagrama general de negocio.	media	4	0	4
3	Elaboración del diagrama de procesos.	media	14	2	16
4	Maquetación de interfaz de usuario	baja	15	1	16
5	Herramientas de desarrollo	media	2	2	4
<b>1</b>	<b>Primera Iteración(primer entrega)</b>	<b>alta</b>	<b>41</b>	<b>8</b>	<b>48</b>
5	Diseño de la base de datos de la aplicación.	alta	24	0	24
6	Elaboración de diccionario de datos.	media	5	5	10
<b>2</b>	<b>Segunda Iteración(segunda entrega)</b>	<b>alta</b>	<b>29</b>	<b>5</b>	<b>34</b>
7	Desarrollo de catálogo de clientes.	alta	24	2	26
8	Desarrollo de catálogo de personal de la empresa.	alta	24	0	24
<b>3</b>	<b>Tercera Iteración(tercera entrega)</b>	<b>alta</b>	<b>48</b>	<b>2</b>	<b>50</b>
9	Desarrollo de módulo de proyectos, actividades, entregables, evidencias, Bitácora y actividades diarias.	alta	32	32	64
<b>4</b>	<b>Cuarta Iteración(cuarta entrega)</b>	<b>alta</b>	<b>32</b>	<b>32</b>	<b>64</b>
10	Desarrollo de módulo de auditoria.	media	16	2	18
11	Desarrollo de módulo de usuarios	media	8	0	8
<b>5</b>	<b>Quinta Iteración(quinta entrega)</b>	<b>media</b>	<b>24</b>	<b>2</b>	<b>26</b>
12	Módulo de resultados y reportes.	media	32	0	32
<b>6</b>	<b>Sexta Iteración(sexta entrega)</b>	<b>media</b>	<b>32</b>	<b>0</b>	<b>32</b>
13	Ejecución de pruebas del sistema.	baja	5	2	7
14	Publicación del sistema.	baja	5	-2	3
<b>7</b>	<b>Séptima Iteración(séptima entrega)</b>	<b>baja</b>	<b>16</b>	<b>0</b>	<b>10</b>
	<b>Total desarrollo del prototipo</b>	<b>alta</b>	<b>222</b>	<b>49</b>	<b>264</b>

Fuente: (Gómez, 2014)

#### 4.1.6. Planificación de las iteraciones.

- a. Primera iteración (Levantamiento y elaboración de procesos).

**Tabla 21** Definición de requerimientos.

Definición de requerimientos.			
<b>Tarea</b>	<b>Encargado</b>	<b>Estado</b>	<b>Horas</b>
Reunión, levantamiento de información.	Jimmy Gómez	Completo	4
Definición de requerimientos funcionales y no funcionales del sistema.	Jimmy Gómez	Completo	4
<b>Total</b>			<b>8</b>

Fuente: (Gómez, 2014)

**Tabla 22** Elaboración de diagrama general de negocio.

Elaboración de diagrama general de negocio.			
<b>Tarea</b>	<b>Encargado</b>	<b>Estado</b>	<b>Horas</b>
Identificar las entidades participantes del sistema.	Jimmy Gómez	Completo	2
Elaborar el gráfico de flujo del sistema.	Jimmy Gómez	Completo	2
<b>Total</b>			<b>4</b>

Fuente: (Gómez, 2014)

**Tabla 23** Elaboración del diagrama de procesos.

Elaboración del diagrama de procesos.			
Tarea	Encargado	Estado	Horas
Estudiar los requerimientos para identificar los procesos.	Jimmy Gómez	Completo	4
Elaborar el diagrama de cada uno de los procesos identificados.	Jimmy Gómez	Completo	12
<b>Total</b>			<b>16</b>

Fuente: (Gómez, 2014)

**Tabla 24** Maquetación de interfaz de usuario.

Maquetación de interfaz de usuario			
Tarea	Encargado	Estado	Horas
Elaborar el diseño para la Master Page del sistema.	Jimmy Gómez	Completo	8
Elaborar el diseño para las páginas secundarias.	Jimmy Gómez	Completo	8
<b>Total</b>			<b>16</b>

Fuente: (Gómez, 2014)

**Tabla 25** Herramientas de desarrollo.

Herramientas de desarrollo			
Tarea	Encargado	Estado	Horas
Escoger las herramientas a utilizar en el desarrollo del sistema.	Jimmy Gómez	Completo	4
<b>Total</b>			<b>4</b>

Fuente: (Gómez, 2014)

b. Segunda iteración (Construcción de base de datos del sistema).

**Tabla 26** Diseño de la base de datos de la aplicación.

Diseño de la base de datos de la aplicación.			
Tarea	Encargado	Estado	Horas
Definición de tablas de base de datos.	Jimmy Gómez	Completo	10
Elaboración de modelo objeto relacional.	Jimmy Gómez	Completo	16
<b>Total</b>			<b>26</b>

Fuente: (Gómez, 2014)

**Tabla 27** Elaboración de diccionario de datos.

Elaboración de diccionario de datos.			
Tarea	Encargado	Estado	Horas
Definición de campos y tipos de datos de cada tabla.	Jimmy Gómez	Completo	5
Elaboración de script de la base de datos.	Jimmy Gómez	Completo	3
Configurar el entorno de base de datos.	Jimmy Gómez	Completo	1
Cargar script de base de datos.	Jimmy Gómez	Completo	1
<b>Total</b>			<b>10</b>

Fuente: (Gómez, 2014)

c. Tercera iteración (Módulo de clientes y personal).

**Tabla 28** Desarrollo de catálogo de clientes.

Desarrollo de catálogo de clientes.			
Tarea	Encargado	Estado	Horas
Desarrollar el CRUD de clientes con sus respectivas validaciones.	Jimmy Gómez	Completo	24
<b>Total</b>			<b>24</b>

Fuente: (Gómez, 2014)

**Tabla 29** Desarrollo de catálogo de personal de la empresa.

Desarrollo de catálogo de personal de la empresa.			
Tarea	Encargado	Estado	Horas
Desarrollar el CRUD de empleados con sus respectivas validaciones.	Jimmy Gómez	Completo	24
<b>Total</b>			<b>24</b>

Fuente: (Gómez, 2014)

d. Cuarta iteración (Módulo de proyectos y Bitácora).

**Tabla 30** Planeación de iteración 10.

Desarrollo de módulo de proyectos, actividades, entregables, evidencias Bitácora y actividades diarias			
Tarea	Encargado	Estado	Horas
Desarrollo de catálogo de proyectos.	Jimmy Gómez	Completo	8
Desarrollo de catálogo de actividades.	Jimmy Gómez	Completo	8
Desarrollo de catálogo de entregables y evidencias	Jimmy Gómez	Completo	16
Desarrollo de Bitácora y actividades diarias.	Jimmy Gómez	Completo	32
<b>Total</b>			<b>64</b>

Fuente: (Gómez, 2014)

e. Quinta iteración (Administración de usuarios, parámetros y auditoria).

**Tabla 31** Desarrollo del módulo de auditoria.

Desarrollo del módulo de auditoria.			
Tarea	Encargado	Estado	Horas
Desarrollo de auditoria del sistema.	Jimmy Gómez	Completo	18
<b>Total</b>			<b>18</b>

Fuente: (Gómez, 2014)

**Tabla 32** Desarrollo de módulo de usuarios.

Desarrollo de módulo de usuarios.			
Tarea	Encargado	Estado	Horas
Desarrollo de Login y administración de usuarios	Jimmy Gómez	Completo	8
<b>Total</b>			<b>8</b>

Fuente: (Gómez, 2014)

f. Sexta iteración (Resultados y reportes).

**Tabla 33** Módulo de resultados y reportes.

Módulo de resultados y reportes.			
Tarea	Encargado	Estado	Horas
Elaboración de resultados y reportes	Jimmy Gómez	Completo	32
<b>Total</b>			<b>32</b>

Fuente: (Gómez, 2014)

g. Séptima iteración (Pruebas y publicación).

**Tabla 34** Ejecución de pruebas del sistema.

Ejecución de pruebas del sistema.			
Tarea	Encargado	Estado	Horas
Ejecución de pruebas.	Jimmy Gómez	Completo	7
<b>Total</b>			<b>7</b>

Fuente: (Gómez, 2014)

**Tabla 35** Publicación del sistema.

Publicación del sistema.			
Tarea	Encargado	Estado	Horas
Publicación del aplicativo en IIS	Jimmy Gómez	Completo	3
<b>Total</b>			<b>3</b>

Fuente: (Gómez, 2014)

## 4.2. Ejecución de las iteraciones.

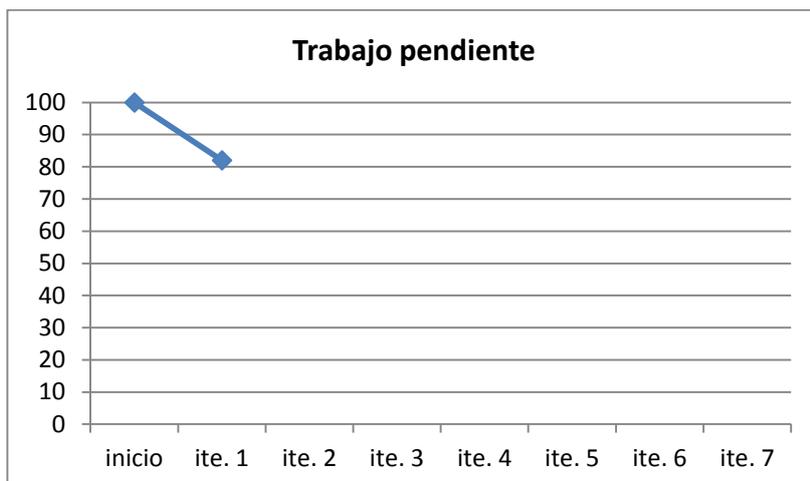
### 4.2.1. Primera iteración (Levantamiento y elaboración de procesos).

En la primera iteración se trabajó colaborativamente, manteniendo reuniones con el gestor del producto, para poder definir correctamente los requerimientos del cliente; esto permitió construir importantes artefactos como: el diagrama general de negocio, definición de roles, casos de uso del sistema, prototipo de interfaz de usuario y definición de herramientas; importantes en el avance del proyecto.

Todas las tareas planificadas en la primera iteración se completaron con éxito, en la ejecución de la iteración se encontraron inconvenientes que al final de la iteración se resolvieron eficazmente, los inconvenientes encontrados son los siguientes:

- Comunicación con el gestor de producto. En la primera reunión, existió un desentendimiento de los requerimientos expuestos por el gestor del producto, por ende se solicitó una segunda reunión para aclarar las dudas, fortaleciendo la comunicación entre el gestor del producto y el equipo técnico.
- Elaboración del diagrama de procesos del sistema. Como en todo desarrollo de software la fase de elaborar los procesos requiere de mucho esfuerzo y tiempo, ya que de esta dependen que las siguientes fases sean ejecutadas con éxito y el producto final sea de calidad.
- Bosquejar una interfaz del sistema amigable con el usuario. Para llegar a determinar una interfaz amigable se llevó a cabo una reunión donde usuarios finales aportaron con ideas que sirvieron para el bosquejo de la interfaz, todo con el fin de familiarizarlos con el sistema, además se estudió técnicas de diseño de páginas web.
- Definición de herramientas de desarrollo. Para la construcción del prototipo se debe seleccionar herramientas que sean compatibles con ASP.NET MVC 5 puesto que el objetivo es desarrollar el prototipo con este Framework; esto ocasionó ciertos inconvenientes a la hora de elegir las correctas herramientas; como solución se seleccionó: el gestor de base de datos SQL SERVER EXPRESS, IDE de desarrollo Visual Studio y como herramienta para hacer reportes Razor PDF; siendo cada uno de estos compatibles y de óptimo rendimiento con el Framework.

**a. Trabajo pendiente del proyecto.**



**Ilustración 72.** Trabajo pendiente, iteración 1.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la primera iteración es de 21%, quedando un 79% de trabajo pendiente por desarrollar en el proyecto.

**b. Horas pendientes en la iteración.**



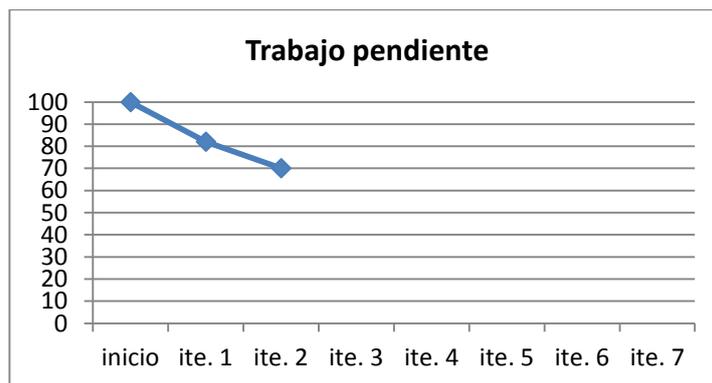
**Ilustración 73.** Horas pendientes, iteración 1.

Fuente: (Gómez, 2014)

#### 4.2.2. Segunda iteración (Construcción de base de datos del sistema).

En la ejecución de la segunda iteración se llevó a cabo trabajos de diseño de la base de datos de la aplicación, elaboración del diccionario de datos y cargado en un gestor de base de datos; trabajos realizados colaborativamente con el equipo técnico del proyecto.

##### a. Trabajo pendiente del proyecto.

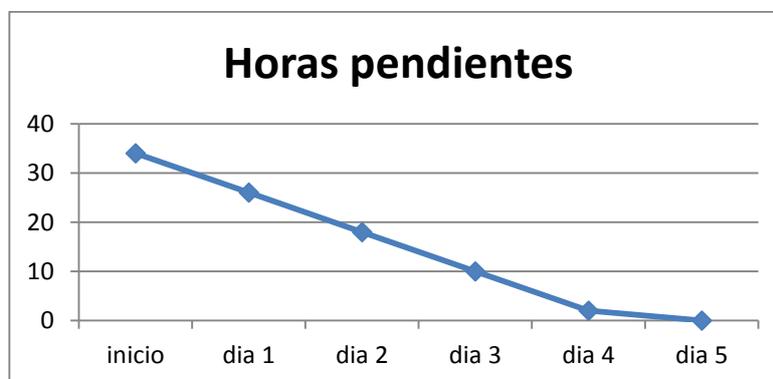


**Ilustración 74.** Trabajo pendiente, iteración 2.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la segunda iteración es de 15%, quedando un 64% de trabajo pendiente por desarrollar en el proyecto.

##### b. Horas pendientes en la iteración.

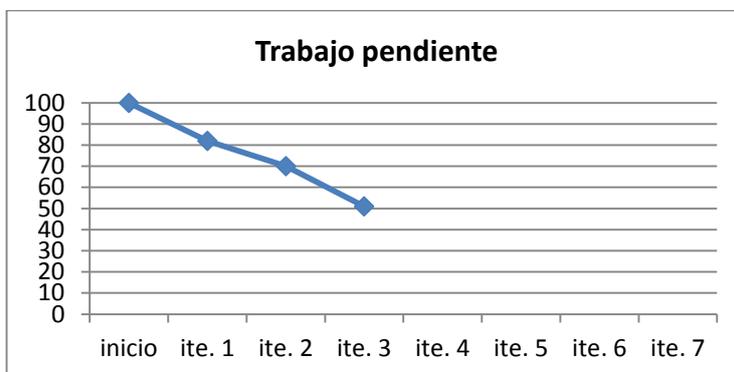


**Ilustración 75.** Horas pendientes, iteración 2.

Fuente: (Gómez, 2014)

### 4.2.3. Tercera iteración (Módulo de clientes y personal).

#### a. Trabajo pendiente del proyecto.

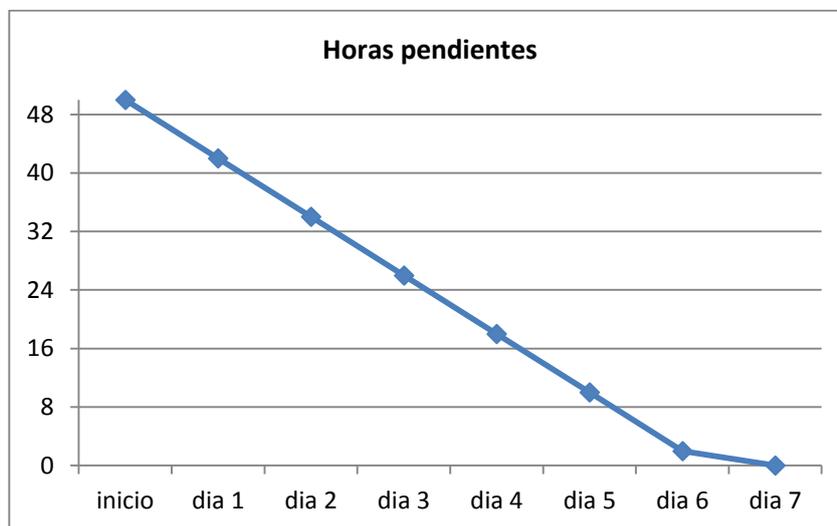


**Ilustración 76.** Trabajo pendiente, iteración 3.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la tercera iteración es de 22%, quedando un 42% de trabajo pendiente por desarrollar en el proyecto.

#### b. Horas pendientes en la iteración.

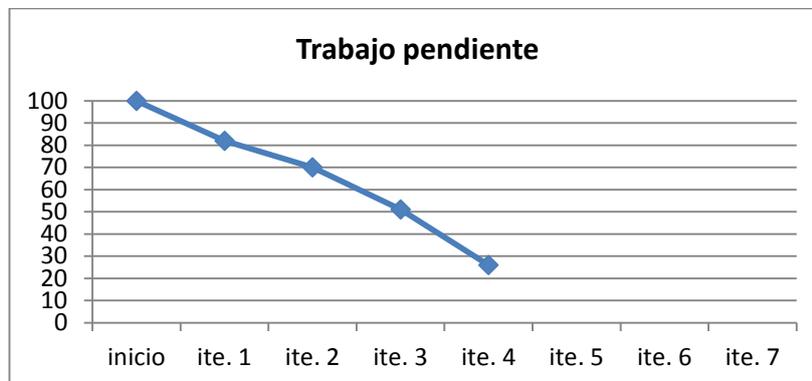


**Ilustración 77.** Horas pendientes, iteración 3.

Fuente: (Gómez, 2014)

#### 4.2.4. Cuarta iteración (Módulo de proyectos y Bitácora).

##### a. Trabajo pendiente del proyecto.

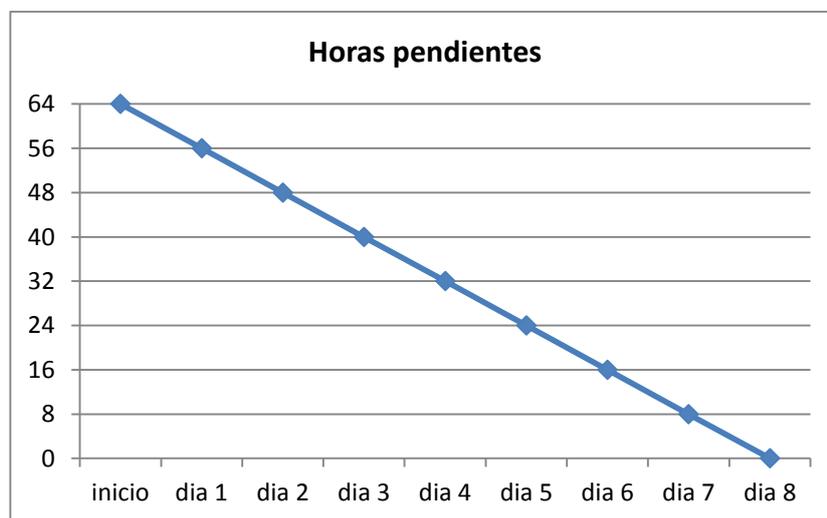


**Ilustración 78.** Trabajo pendiente, iteración 4.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la cuarta iteración es de 13%, quedando un 29% de trabajo pendiente por desarrollar en el proyecto.

##### b. Horas pendientes en la iteración.

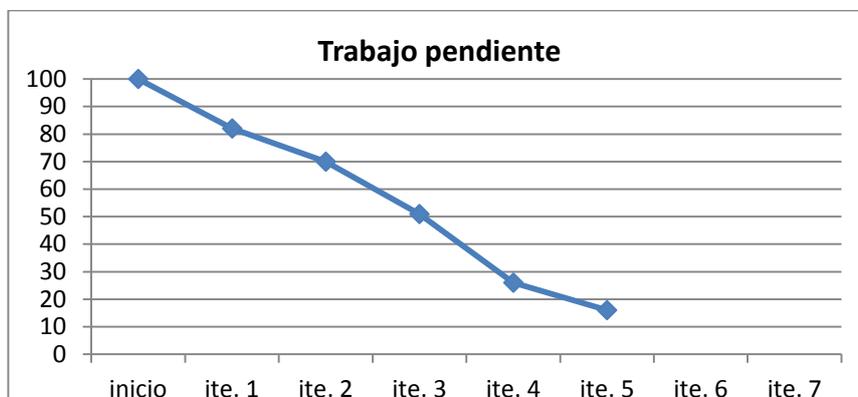


**Ilustración 79.** Horas pendientes, iteración 4.

Fuente: (Gómez, 2014)

#### 4.2.5. Quinta iteración (Administración de usuarios, parámetros y auditoría).

##### a. Trabajo pendiente del proyecto.

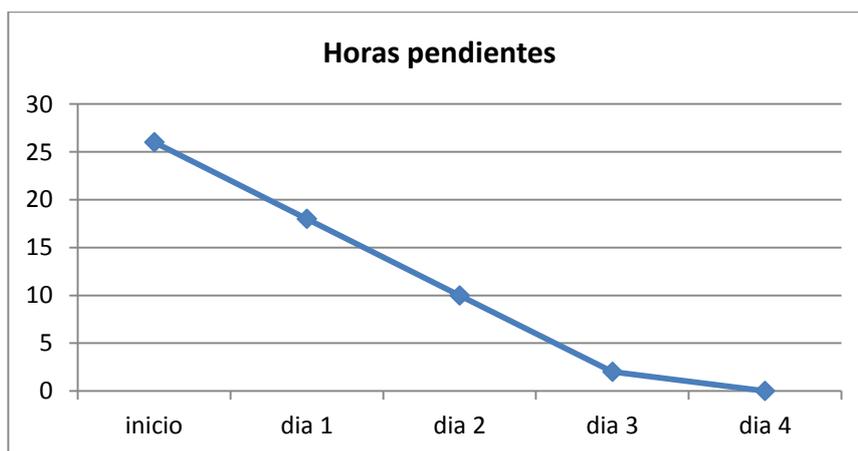


**Ilustración 80.** Trabajo pendiente, iteración 5.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la quinta iteración es de 12%, quedando un 17% de trabajo pendiente por desarrollar en el proyecto.

##### b. Horas pendientes en la iteración.



**Ilustración 81.** Horas pendientes, iteración 5.

Fuente: (Gómez, 2014)

En la ejecución de la iteración 3, 4, 5; el equipo técnico se concentró netamente en el desarrollo del aplicativo, donde se pudo evidenciar las bondades que representa desarrollar

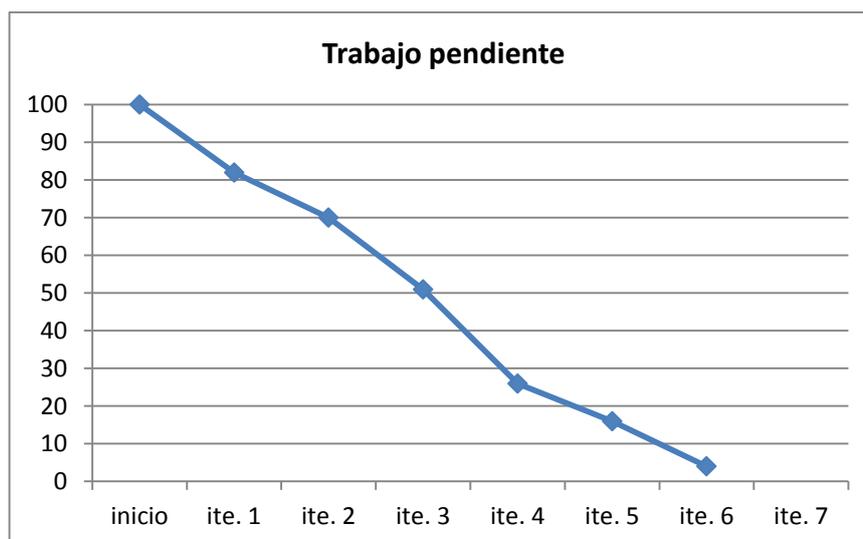
aplicaciones web siguiendo un patrón de diseño arquitectónico, utilizando el Framework ASP.NET MVC. En el transcurso de desarrollo del aplicativo se tuvo constates reuniones de presentación de avances, en el cual se pudo comprobar la funcionalidad del aplicativo.

Las tareas de las iteraciones 3, 4, 5 se culminaron exitosamente y sirvieron como experiencia para resolver inconvenientes como la adaptación del equipo técnico con el manejo de las herramientas de desarrollo, principalmente con el Framework MVC, puesto que es el primer desarrollo con este Framework, lo cual requirió realizar ajustes en el tiempo de entrega de las iteraciones; al final todos los inconvenientes fueron resueltos y sirvieron de experiencia para próximos desarrollos de la empresa.

#### 4.2.6. Sexta iteración (Resultados y reportes).

En la sexta iteración el equipo técnico del proyecto planificó y desarrolló los reportes del aplicativo, los mismos que evidenciaron los resultados de los proyectos y actividades ejecutadas y registradas en el sistema.

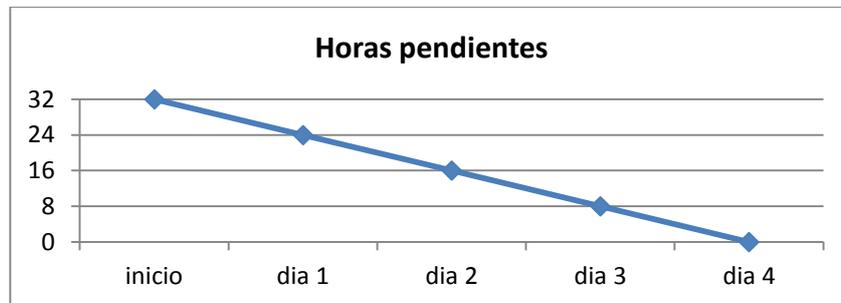
##### a. Trabajo pendiente del proyecto.



**Ilustración 82.** Trabajo pendiente, iteración 6.

El porcentaje de trabajo desarrollado en la sexta iteración es de 13%, quedando un 4% de trabajo pendiente por desarrollar en el proyecto.

#### b. Horas pendientes en la iteración.



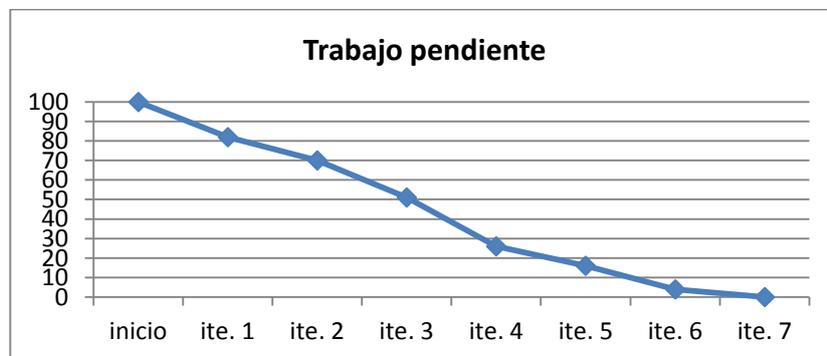
**Ilustración 83.** Horas pendientes, iteración 6.

Fuente: (Gómez, 2014)

#### 4.2.7. Séptima iteración (Pruebas y publicación).

La séptima iteración representa la fase final del proyecto; se realizaron: pruebas para probar el funcionamiento del sistema; con un resultado exitoso. Se pudo comprobar el óptimo funcionamiento del Framework ASP.NET MVC en su quinta versión en aspectos como: rendimiento, escalabilidad y compatibilidad.

#### a. Trabajo pendiente del proyecto.



**Ilustración 84.** Trabajo pendiente, iteración 7.

Fuente: (Gómez, 2014)

El porcentaje de trabajo desarrollado en la séptima iteración es de 4%, quedando un 0% de trabajo pendiente completando en su totalidad el proyecto.

#### b. Horas pendientes en la iteración.



**Ilustración 85.** Horas pendientes, iteración 7.

Fuente: (Gómez, 2014)

### 4.3. Demostración de los resultados de iteraciones.

#### 4.3.1. Resultados de primera iteración.

##### a. Requerimientos del sistema.

El sistema tendrá la capacidad de cumplir los siguientes requerimientos:

- Llevar un catálogo de clientes de la empresa de los cuales solicitan el desarrollo de uno o varios proyectos a la empresa. De los clientes deben incluir la siguiente información: ruc del cliente, nombre, teléfono, dirección, encargado de proyectos de la empresa.
- Llevar un catálogo del personal de la empresa. El personal es el encargado de ejecutar los proyectos y actividades, debe ser clasificado por su cargo y el departamento al que pertenece, el personal debe tener dos estados: activo en caso de encontrarse trabajando con

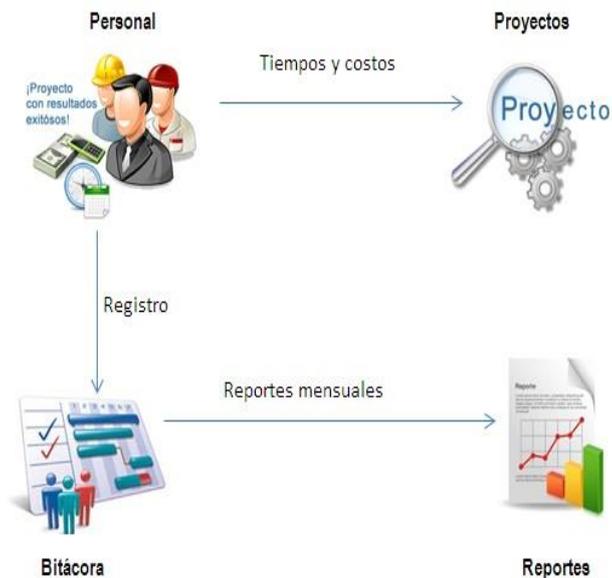
normalidad en la empresa o inactivo en caso de encontrarse temporalmente fuera de la empresa, especificar la persona o jefe encargado del empleado, el personal debe manejar un usuario y una contraseña los mismos que serán válidos para ingresar a la aplicación. De los empleados de la empresa se debe incluir la siguiente información: cédula, nombres, teléfono, email, dirección, estado, usuario, password. Permisos.

- Llevar un catálogo de proyectos. Los proyectos deben ser clasificados por el tipo de proyecto, además los proyectos están constituidos por una o varias actividades. Los proyectos pueden estar en estados: pendiente, planificado, iniciado y terminado. De los proyectos se registró la siguiente información: nombre del proyecto, fecha inicio, fecha fin, descripción, estado, porcentaje, tiempo usado y tiempo estimado.
- Llevar un catálogo de actividades que se necesitan para ser ejecutado el proyecto. Las actividades serán clasificadas por tipo de actividad, cada actividad tiene un empleado encargado de ejecutarla; una actividad puede tener una o varias sub actividades, las actividades tienen estados: pendiente, trabajando, terminado. De las actividades se registró: descripción de la actividad, estado, tiempo de ejecución, tiempo estimado, fecha de planificación, fecha de inicio de la actividad, fecha fin de la actividad.
- Trabajar con una bitácora. En la bitácora se llevará los registros de los tiempos empleados por el personal en la ejecución de las actividades de cada proyecto. De la bitácora se registró: fecha de inicio, fecha de fin, resultado y tiempo empleado.
- Subir evidencias. El sistema debe llevar un registro de las evidencias del trabajo realizado en cada actividad, de las evidencias de llevo el registro de la descripción de la evidencia, fecha de entrega, ubicación de archivo de evidencia.
- Subir entregables. Los entregables son los resultados o el producto final que se entrega al ser desarrollado cierto proyecto o actividad, los cuales previamente deben ser revisados y garantizado su funcionamiento. De los entregables se registró: la descripción del entregable,

fecha estimada de entrega, fecha de entrega, si fue revisado o no el entregable y ubicación del archivo entregable.

- Registro de actividades diarias realizadas por el personal de la empresa que no están relacionadas con un proyecto planificado. De las actividades se llevó la siguiente información: descripción, tiempo usado, fecha inicio, observaciones, si es facturado o no y motivo de facturación.
- Reportes y resultados que deben reflejar el tiempo invertido en cada proyecto y actividad.
- Administración de usuarios y auditoría de la aplicación.

#### b. Diagrama general de negocio.



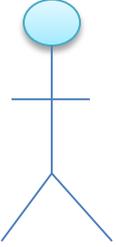
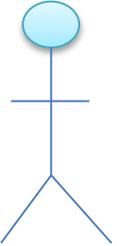
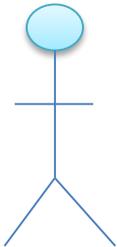
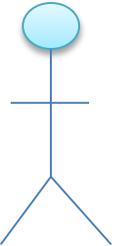
**Ilustración 86.** Diagrama de negocio.

Fuente: (Gómez, 2014)

### c. Diagrama de procesos.

Actores.

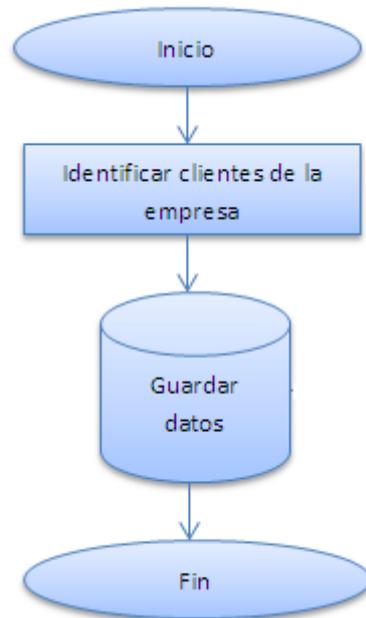
**Tabla 36** Actores.

Actor	Descripción
 Administrador	<p>El administrador del sistema es el encargado de administrar el ingreso y mantenimiento de clientes y personal de la empresa, parámetros y usuarios de la aplicación.</p>
 Gerente	<p>El gerente es la persona que puede visualizar los resultados obtenidos en el desarrollo de proyectos de la empresa.</p>
 Jefe de proyecto	<p>El jefe de proyectos es el encargado de planificar los proyectos y actividades a ejecutarse.</p>
 Empleado	<p>El empleado es la persona encargada de registrar los tiempos de duración en cada proyecto o actividad desarrollada.</p>

Fuente: (Gómez, 2014)

- Catálogo de clientes.

Proceso desarrollado por el administrador del sistema.

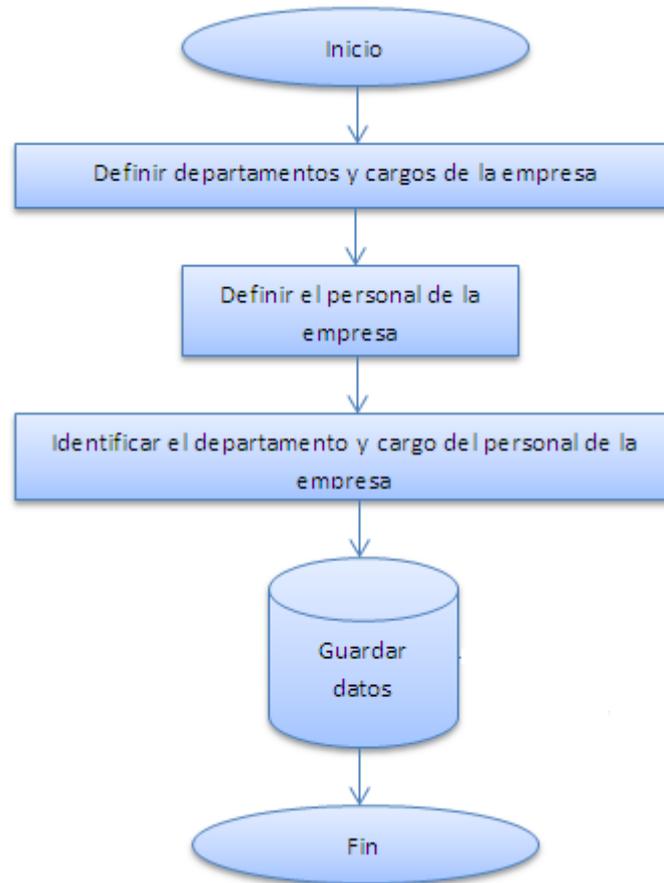


**Ilustración 87.** Ingreso de clientes.

**Fuente:** (Gómez, 2014)

- Catálogo de personal de la empresa.

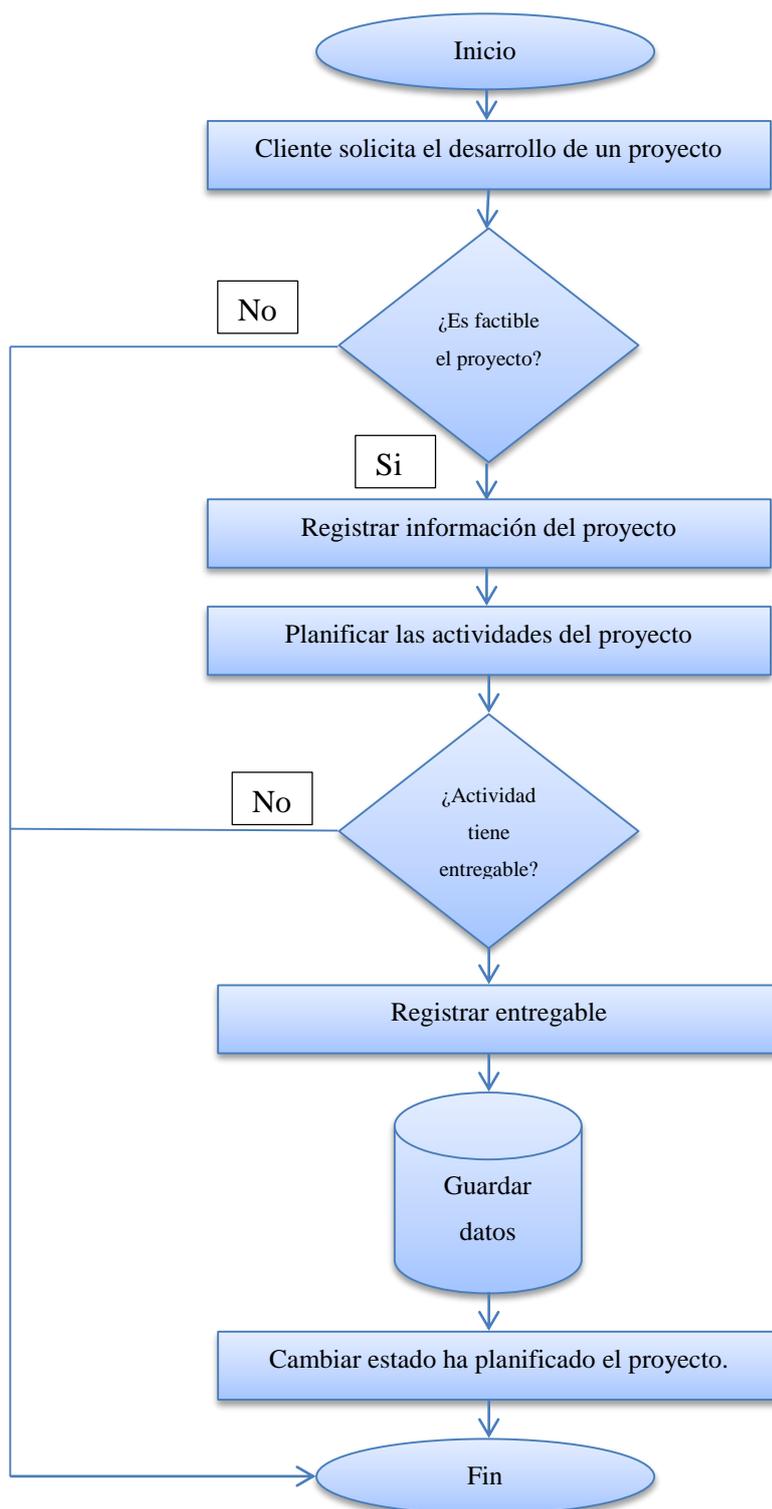
Proceso ejecutado por el administrador de la empresa.



**Ilustración 88.** Ingreso de personal de la empresa.

Fuente: (Gómez, 2014)

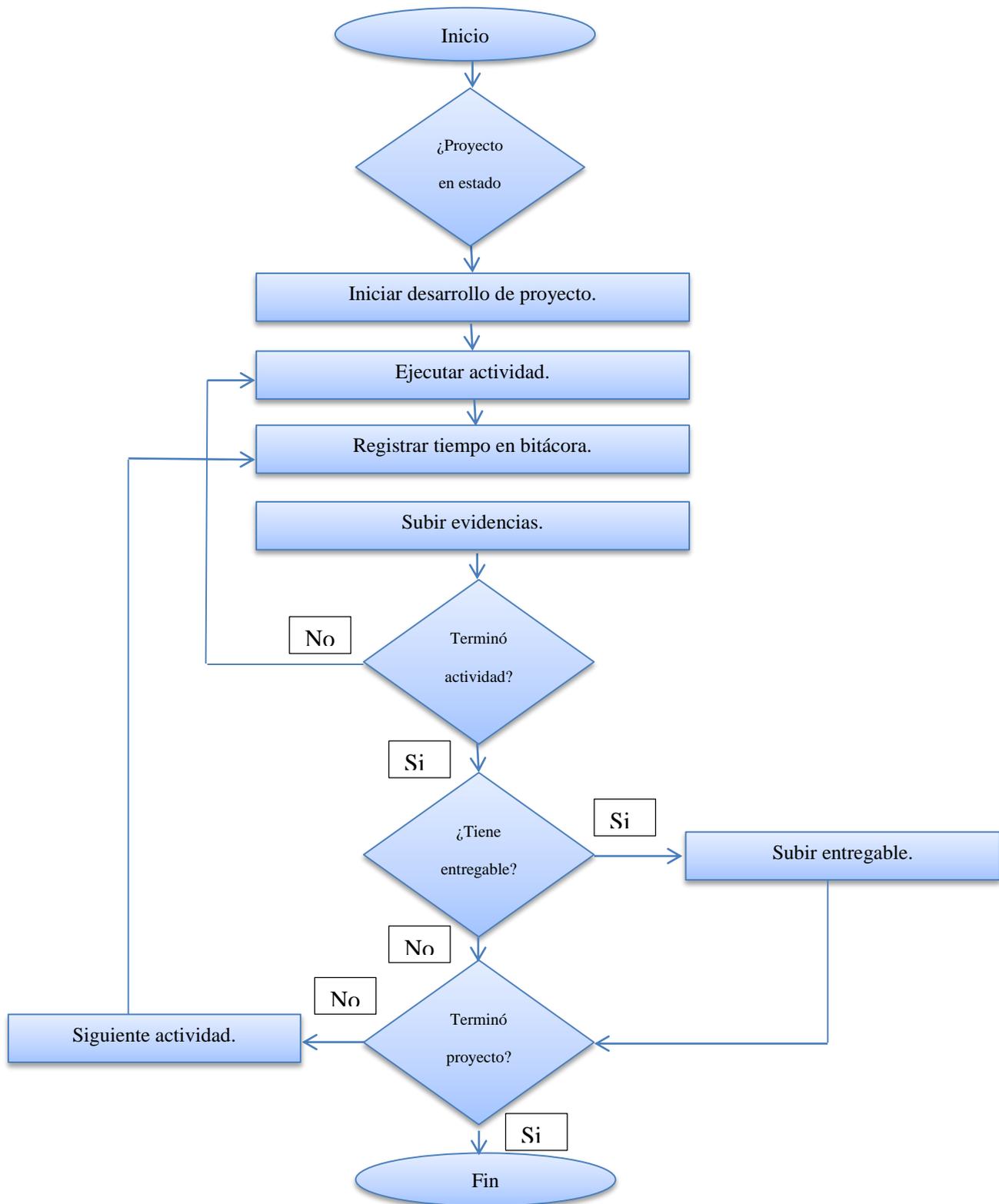
- Catálogo de proyectos, actividades y entregables.



**Ilustración 89.** Catálogo de proyectos, actividades y entregables.

Fuente: (Gómez, 2014)

- Bitácora: Proceso desarrollado por el empleado de la empresa.



**Ilustración 90.** Bitácora.

Fuente: (Gómez, 2014)

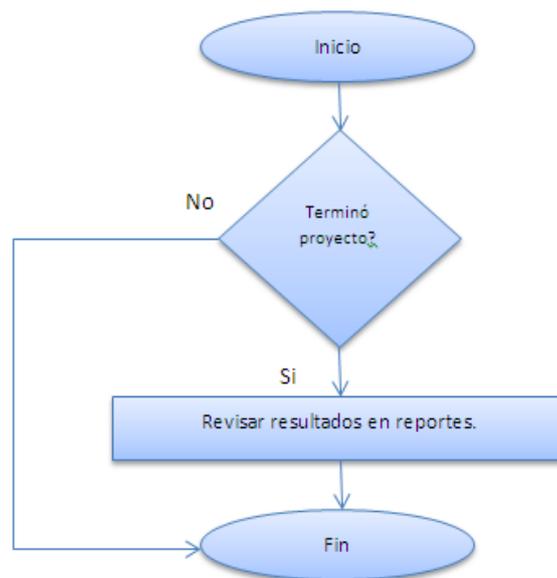
Registro de actividades diarias. Desarrollado por los empleados de la empresa.



**Ilustración 91.** Diagrama actividades diarias.

**Fuente:** (Gómez, 2014)

- Reportes y resultados. Proceso desarrollado por el gerente.



**Ilustración 92.** Reportes y resultados.

Fuente: (Gómez, 2014)

#### a. Maquetación de interfaz de usuario.

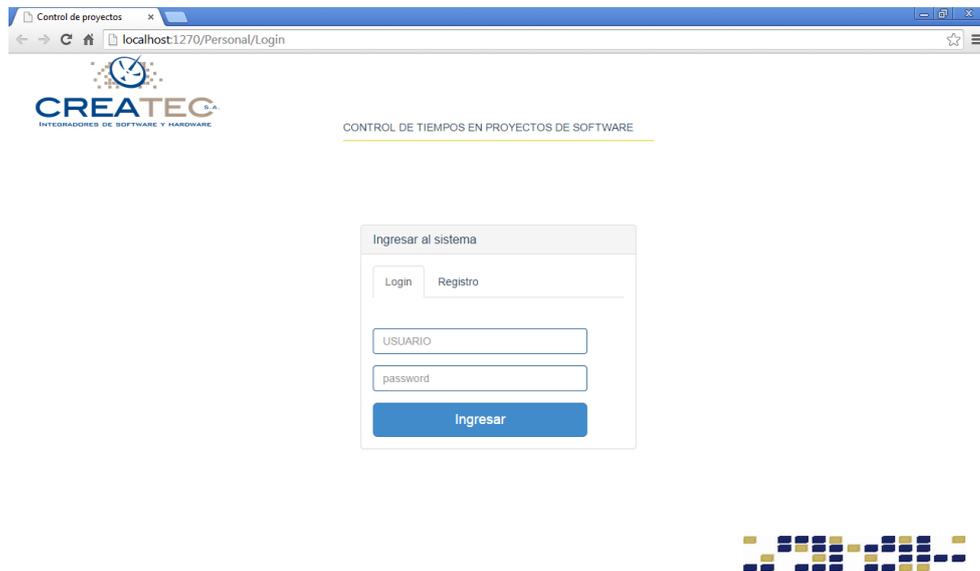
- Diseño de la Master Page que contendrá las páginas hijas del sistema.



**Ilustración 93.** Master page.

Fuente: (Gómez, 2014)

- Diseño del Login del sistema.



**Ilustración 94.** Login del sistema.

Fuente: (Gómez, 2014)

#### **b. Herramientas de desarrollo.**

La selección de herramientas y Frameworks para el desarrollo del sistema es una tarea indispensable ya que las herramientas elegidas deben ser compatibles entre sí y su desempeño el más óptimo, para agilizar el desarrollo del sistema.

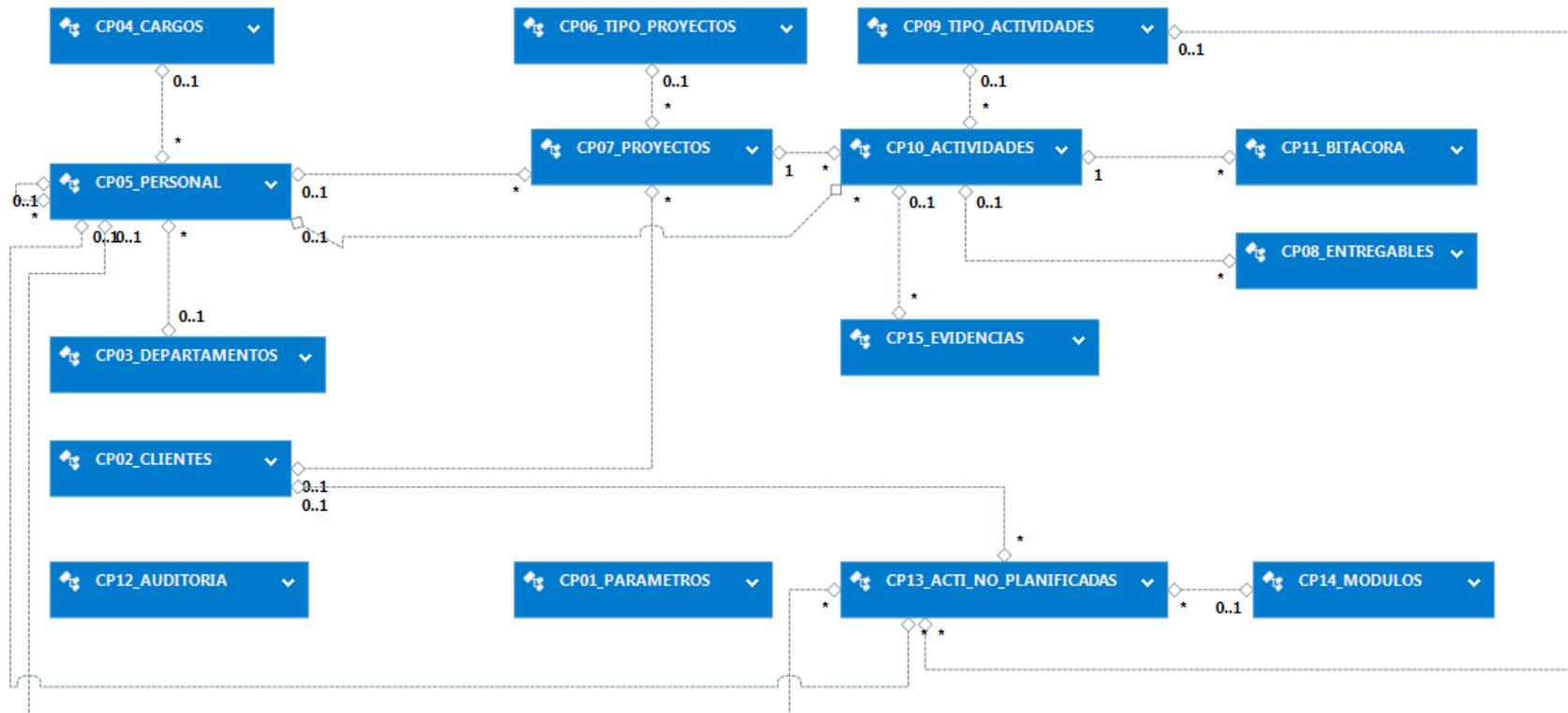
**Tabla 37** Herramientas de desarrollo

Herramienta	Descripción.	Versión.	Edición.
ASP.NET MVC	Framework de desarrollo Web.	5	Libre.
Bootstrap	Framework de diseño web.	3	Libre.
Visual Studio	IDE de desarrollo.	2013	Express.
SQL SERVER	Manejador de base de datos.	9.3	Express.
Razor PDF	Framework para desarrollar reportes	8	Libre.

Fuente: (Gómez, 2014)

### 4.3.2. Resultados de segunda iteración.

#### a. Diseño de base de datos.



**Ilustración 95.** Modelo de datos del sistema.

Fuente: (Gómez, 2014)

## b. Diccionario de datos.

**Tabla 38** Tabla parámetros.

CP01_parametros		
Nombre	Tipo de dato	Restricción
grupo	integer	(pk)
descripción	varchar(100)	
valor	varchar(100)	
id auditoria	Timestamp	(fk)

Fuente: (Gómez, 2014)

**Tabla 39** Tabla clientes

CP02_clientes		
Nombre	Tipo de dato	Restricción
ruc	varchar(13)	(pk)
nombre	varchar(100)	
Teléfono	varchar(20)	
Dirección	varchar(Max)	
Encargado	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 40** Tabla departamentos.

CP03_departamento		
Nombre	Tipo de dato	Restricción
id_departamento	integer	(pk)
descripción	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 41** Tabla cargos.

CP04_cargo		
Nombre	Tipo de dato	Restricción
id_cargo	integer	(pk)
id_departamento	char(3)	(pk) (fk)
descripción	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 42** Tabla personal.

CP05_personal		
Nombre	Tipo de dato	Restricción
cedula	varchar(10)	(pk)
cédula_jefe	varchar(10)	(pk)
id_departamento	integer	(fk)
id_cargo	integer	(fk)
apellido1	varchar(100)	
apellido2	varchar(100)	
nombre1	varchar(100)	
nombre2	varchar(100)	
teléfono	varchar(20)	
email	varchar(100)	
dirección	varchar(Max)	
estado	varchar(100)	
password	varchar(100)	
programas	varchar(100)	
nombre completo	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 43** Tabla tipos de proyecto.

CP06_tipo_proyectos		
Nombre	Tipo de dato	Restricción
id_tipo_proyecto	integer	(pk)
descripción	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 44** Tabla proyectos.

CP07_proyectos		
Nombre	Tipo de dato	Restricción
id_proyecto	integer	(pk)
Id_tipo_proyecto	Integer	(fk)
cédula	varchar(10)	(fk)
ruc	varchar(13)	(fk)
nombre	varchar(100)	
fecha_inicio	timestamp	
fecha_fin	timestamp	
descripción	varchar(1024)	
estado	varchar(100)	
porcentaje	integer	
tiempo_usado	timestamp	
tiempo_estimado	timestamp	

Fuente: (Gómez, 2014)

**Tabla 45** Tabla entregable.

CP08_entregables		
Nombre	Tipo de dato	Restricción
id_entregable	integer	(pk)
id_proyecto	integer	(fk)
id_actividad	integer	(fk)
descripcion	varchar(100)	
fecha_estimada	datetime	
fecha_entrega	datetime	
rebiso	datetime	
path	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 46** Tabla tipo actividades.

CP09_tipo_actividades		
Nombre	Tipo de dato	Restricción
id_tipo_actividad	integer	(pk)
descripcion	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 47** Tabla actividades.

CP10_actividades		
Nombre	Tipo de dato	Restricción
id_proyecto	integer	(pk)(fk)
id_actividad	integer	(pk)
id_actividad_pertenece	integer	(fk)
id_tipo_actividad	integer	(fk)
cedula	varchar(10)	(fk)
descripcion	varchar(Max)	
estado	varchar(100)	
tiempo_usado	timestamp	
tiempo_estimado	timestamp	
fecha_planificacion	datetime	
fecha_inicio	datetime	
fecha_fin	datetime	

Fuente: (Gómez, 2014)

**Tabla 48** Tabla bitácora.

CP11_bitacora		
Nombre	Tipo de dato	Restricción
id_bitacora	integer	(pk)
id_proyecto	integer	(fk)
id_actividad	integer	(fk)
fecha_inicio	datetime	
fecha_fin	datetime	
tiempo_usado	integer	

Fuente: (Gómez, 2014)

**Tabla 49** Tabla auditoria.

CP12_auditoria		
Nombre	Tipo de dato	Restricción
Id_auditoria	integer	(pk)
fecha	datetime	
usuario	varchar(50)	
mensaje	varchar(Max)	
error	varchar(Max)	
servidor	varchar(250)	
modelo	varchar(250)	
controlador	varchar(250)	
vista	varchar(250)	
acción	varchar(250)	
parametros	varchar(250)	
path_aplicación	varchar(250)	
datos	varchar(Max)	
aplicación_cliente	varchar(500)	
cliente	varchar(250)	

Fuente: (Gómez, 2014)

**Tabla 50** Tabla actividades no planificadas.

CP13_ACTI_NO_PLANIFICADAS		
Nombre	Tipo de dato	Restricción
Id_actividad	integer	(pk)
Ruc	Varchar(13)	(fk)
cédula	varchar(10)	(fk)
Id_tipo_actividad	integer	(fk)
descripción	varchar(Max)	
tiempo_usado	timestamp	
fecha_inicio	datetime	
observaciones	varchar(Max)	
facturado	varchar(100)	
motivo_facturacion	varchar(100)	
id_módulo	int	(fk)
solicitante	varchar(10)	(fk)

Fuente: (Gómez, 2014)

**Tabla 51** Tabla de módulos.

CP14_MODULOS		
Nombre	Tipo de dato	Restricción
Id_módulo	integer	(pk)
código	varchar(100)	(fk)
descripción	varchar(100)	

Fuente: (Gómez, 2014)

**Tabla 52** Tabla de evidencias.

CP15_EVIDENCIAS		
Nombre	Tipo de dato	Restricción
Id_evidencia	integer	(pk)
Id_proyecto	integer	(fk)
Id_actividad	integer	(fk)
descripción	varchar(100)	
fecha_entrega	datetime	
path_evidencia	varchar(100)	

Fuente: (Gómez, 2014)

### 4.3.3. Resultados de tercera iteración.

En la demostración y entrega del módulo de clientes y personal se realizó el siguiente cuadro de validación de ítems que debe cumplir el módulo para ser aprobado por el gestor del producto.

**Tabla 53** Resultados iteración 3.

Ítem	Cumple	No cumple
CRUD de clientes de la empresa.	X	
CRUD de cargos de la empresa	X	
CRUD de departamentos de la empresa	X	
CRUD de personal de la empresa.	X	
Clasificación de personal por cargo y departamento.	X	
Manejo de estado del empleado.	X	
Validación de campos.	X	
Facilidad de uso.	X	
Interfaz amigable para el usuario.	X	

Fuente: (Gómez, 2014)

### 4.3.4. Resultados de cuarta iteración.

En la demostración y entrega del módulo de proyectos y Bitácora se realizó el siguiente cuadro de validación de ítems que debe cumplir el módulo para ser aprobado por el gestor del producto.

**Tabla 54** Resultados iteración 4.

Ítem	Cumple	No cumple
CRUD de tipos de proyecto.	X	
CRUD de proyectos.	X	
Clasificación de proyectos por tipo de proyectos.	X	
Creación de proyectos por clientes.	X	
Designación de proyectos al personal de la empresa.	X	
CRUD de tipo de actividades.	X	
CRUD de actividades.	X	
Clasificación de actividades por tipo de actividad.	X	
Agregar actividades a proyectos.	X	
Secuencia de actividades.	X	
Clasificación de actividades padres e hijas.	X	
CRUD de entregables.	X	
Designación de entregables a actividades y proyectos.	X	
Registro de tiempos en Bitácora.	X	
Validación de campos.	X	
Interfaz amigable.	X	
Facilidad de uso.	X	

Fuente: (Gómez, 2014)

#### 4.3.5. Resultados de quinta iteración.

En la demostración y entrega del módulo de Usuarios y auditoria se realizó el siguiente cuadro de validación de ítems que debe cumplir el módulo para ser aprobado por el gestor del producto.

**Tabla 55** Resultados iteración 5.

Ítem	Cumple	No cumple
Agregar usuarios al sistema.	X	
Dar de baja usuarios.	X	
Manejo de Roles de usuario.	X	
Auditoria de campos.	X	
Login del sistema.	X	
Validación de campos.	X	
Interfaz amigable.	X	
Facilidad de uso.	X	

Fuente: (Gómez, 2014)

**4.3.6. Resultados de sexta iteración.**

En la demostración y entrega de los reportes que refleja los resultados del desarrollo de proyectos de la empresa, se validó los siguientes ítems.

**Tabla 56** Resultados iteración 6.

Ítem	Cumple	No cumple
Resultados de tiempos de proyectos planificados	X	
Reporte de tiempos de proyectos planificados	X	
Resultados de actividades	X	
Reporte de actividades	X	
Resultados de actividades diarias	X	
Reporte de actividades diarias.	X	
Facilidad de uso.	X	

Fuente: (Gómez, 2014)

**4.3.7. Resultados de séptima iteración.**

En la última fase de desarrollo del proyecto se realizaron las respectivas pruebas unitarias y de estrés de la aplicación que concluyo con la publicación en IIS, para la entrega se realizó la validación de los siguientes ítems.

**Tabla 57** Resultados iteración 7.

<b>Ítem</b>	<b>Cumple</b>	<b>No cumple</b>
Velocidad de ejecución.	X	
Desempeño.	X	
Disponibilidad.	X	
Escalabilidad.	X	
Flexibilidad.	X	
Mantenimiento.	X	
Seguridad.	X	
Operatividad.	X	
Publicación en IIS.	X	

**Fuente:**(Gómez, 2014)

## CAPÍTULO V

### 5. Análisis de impactos.

El estudio realizado se complementa con un análisis de los impactos en las áreas generales en las que el proyecto influye positiva o negativamente tanto en el desarrollo del sistema con ASP.NET MVC como también el uso del sistema por los empleados de la empresa.

Se logró determinar las siguientes áreas: productivo, organizativo, económico, innovación, humano.

En la siguiente tabla se determina los niveles de rango positivo y negativo.

**Tabla 58** Impactos.

Rango	Impacto
-3	Impacto alto negativo.
-2	Impacto medio negativo.
-1	Impacto bajo negativo.
0	No hay impacto.
1	Impacto bajo positivo.
2	Impacto medio positivo.
3	Impacto alto positivo.

Fuente:(Gómez, 2014)

A continuación se evalúa los impactos en cada una de las áreas.

## 5.1. Impacto productivo.

**Tabla 59** Impacto productivo.

Indicador	-3	-2	-1	0	1	2	3	Total
<b>Desarrollo del sistema</b>								
Tiempo de desarrollo					x			1
Uso de metodología							x	3
Producción de nuevas aplicaciones						x		2
Uso de arquitectura de desarrollo							x	3
<b>Utilización del sistema dentro de la empresa.</b>								
Tiempo de trabajo							x	3
Presentación de resultados							x	3
Facilidad de manejo del sistema						x		2
<b>TOTAL</b>					1	4	12	<b>17</b>

Fuente:(Gómez, 2014)

**Total de impacto productivo: 17/7**

**Total de impacto productivo: 2.4**

**Nivel de impacto productivo: medio positivo.**

## 5.2. Impacto organizativo.

**Tabla 60** Impacto organizativo.

Indicador	-3	-2	-1	0	1	2	3	Total
<b>Desarrollo del sistema.</b>								
Organización de los procesos de desarrollo.						x		2
Organización de grupos de desarrollo.						x		2
Organización de información.							x	3
<b>Utilización del sistema dentro de la empresa.</b>								
Manejo de la información.							x	3
Organización de proyectos y actividades de la empresa.							x	3
Organización de grupos de trabajo.						x		2
Organización del trabajo de los empleados.							x	3
<b>TOTAL</b>						6	12	<b>18</b>

Fuente:(Gómez, 2014)

**Total de impacto productivo: 18/7**

**Total de impacto productivo: 2.57**

**Nivel de impacto productivo: alto positivo.**

## 5.3. Impacto económico.

**Tabla 61** Impacto Económico.

Indicador	-3	-2	-1	0	1	2	3	Total
<b>Desarrollo del sistema.</b>								
Reducción de costes en desarrollo.						x		2
Comercialización de aplicaciones web.				x				0
Reducción de costes en herramientas de desarrollo.					x			1
Reducción de costes de equipos.					x			1
<b>Utilización del sistema dentro de la empresa.</b>								
Reducción de costes en papel.						x		2
Reducción de costes en soporte técnico.				x				0
<b>TOTAL</b>				0	2	4		<b>6</b>

Fuente:(Gómez, 2014)

**Total de impacto productivo: 6/6**

**Total de impacto productivo: 1**

**Nivel de impacto productivo: bajo positivo.**

#### **5.4. Impacto innovación.**

**Tabla 62** Impacto innovación.

<b>Indicador</b>	-3	-2	-1	0	1	2	3	Total
<b>Desarrollo del sistema</b>								
Desarrollo de nuevas aplicaciones web							x	3
Uso de nuevas técnicas y prácticas de programación.							x	3
Uso de nuevos Framework de desarrollo web.							x	3
Investigación de nuevas herramientas.							x	3
<b>Utilización del sistema dentro de la empresa.</b>								
Uso de la aplicación en dispositivos móviles.					x			1
Uso de nuevas aplicaciones en la empresa						x		2
<b>TOTAL</b>					1	2	12	<b>15</b>

**Fuente:**(Gómez, 2014)

**Total de impacto productivo: 15/6**

**Total de impacto productivo: 2.5**

**Nivel de impacto productivo: alto positivo.**

## 5.5. Impacto humano.

**Tabla 63** Impacto humano.

Indicador	-3	-2	-1	0	1	2	3	Total
<b>Desarrollo del sistema</b>								
Aceptación de la herramienta							x	3
Trabajo en equipo						x		2
<b>Utilización del sistema dentro de la empresa.</b>								
Aceptación del sistema							x	3
Trabajo en equipo.						x		2
Colaboración de los empleados.							x	3
<b>TOTAL</b>						4	9	<b>13</b>

Fuente:(Gómez, 2014)

**Total de impacto productivo: 13/5**

**Total de impacto productivo: 2.6**

**Nivel de impacto productivo: alto positivo.**

## 5.6. CONCLUSIONES.

- Después de haber finalizado el presente proyecto de tesis, se puede concluir que el análisis de la herramienta y el desarrollo del aplicativo web resuelven el problema planteado, cumpliendo con todos los objetivos trazados al iniciar el proyecto. Llegando a la conclusión que ASP.NET MVC es un Framework muy maduro, recomendado para la construcción de aplicaciones web robustas, organizadas y que cumplen con estándares de calidad.
- Con el análisis realizado se logró construir la documentación necesaria para estudiar el Framework ASP.NET MVC 5 e incentivar a un análisis más profundo de la herramienta.
- El presente trabajo tuvo como logro difundir las bondades al desarrollar aplicaciones web con ASP.NET MVC 5, con el fin de aumentar su producción en el mercado.
- En la comparativa realizada entre Web Forms y MVC se puede apreciar la ventaja de MVC debido al análisis realizado en aspectos de escalabilidad, compatibilidad, rendimiento, soporte técnico, control sobre el HTML y arquitectura. Pero no quiere decir que la una reemplace a la otra, Su uso depende del tipo de proyecto que se desea desarrollar.
- El uso de SCRUM como metodología de desarrollo en la implementación del aplicativo, promovió la organización y trabajo en equipo, dando prioridad a los requerimientos del cliente, permitiendo obtener muy buenos resultados en poco tiempo.
- En la realización del análisis del Framework se puede apreciar que Microsoft apuesta finalmente por las actuales tendencias de la web construyendo nuevas tecnologías que facilitan el desarrollo de aplicaciones.
- En la construcción del aplicativo se pudo explotar todas sus características y evidenciar las bondades que presenta este Framework, aumentando la productividad de desarrollo en

temas como: escalabilidad, rendimiento, compatibilidad, manejo de una arquitectura y facilidad de pruebas unitarias, convirtiéndolo en una herramienta robusta y perfecta para la construcción de aplicativos a nivel empresarial.

- En el desarrollo de prototipo web se logró llevar un mejor control del tiempo invertido por los recursos humanos de CREATEC S.A. en proyectos de software.
- Con el desarrollo del aplicativo se promovió la planeación y organización de proyectos de software en los empleados de la empresa.
- En el desarrollo del sistema se utilizó ASP.NET MVC 5, esto incentivó a innovar en la construcción de nuevas aplicaciones web dentro de la empresa para en un futuro ser comercializadas.

## 5.7. RECOMENDACIONES.

- Para el desarrollo de aplicaciones web robustas que solucionan problemas de gran complejidad a nivel empresarial se recomienda utilizar ASP.NET MVC, puesto que es una herramienta muy madura, cuenta con todo el respaldo y soporte oficial del equipo técnico de ASP.NET.
- Para obtener información y soporte técnico de la herramienta visitar las páginas oficiales de ASP.NET.
- En caso de desarrollar una aplicación ASP.NET MVC se recomienda tener conocimientos previos del patrón de diseño arquitectónico MVC y conocimientos por lo menos básicos en tecnologías .NET.
- Para contribuir con el desarrollo y corrección de errores de ASP.NET MVC se recomienda visitar el sitio de alojamiento de proyectos de software libre CodePlex de Microsoft.
- En un desarrollo, antes de decidir que herramienta utilizar, analizar aspectos como: requerimientos del cliente, tipo de proyecto, tiempo de desarrollo y conocimientos de equipo técnico; ya que estos garantizan el éxito del proyecto.
- Al decidir las herramientas de programación que se ocupará en una aplicación desarrollada con ASP.NET MVC, se recomienda revisar la compatibilidad entre ellas.
- Para crear interfaces web amigables y vistosas en un proyecto con ASP.NET MVC se recomienda utilizar Bootstrap y JQuery Mobile, ya que estos además de ser muy ricos en interfaces web vistosas cumplen con los estándares de HTML 5.

## REFERENCIAS BIBLIOGRÁFICAS.

- Albaladejo, X. (2013). *Proyectos ágiles*. Obtenido de <http://www.proyectosagiles.org/>
- Albaladejo, X. (2013). *Proyectos ágiles*. Obtenido de <http://www.proyectosagiles.org/fundamentos-de-scrum>
- Asp.Net. (27 de Enero de 2013). *ASP.NET MVC Overview*. Obtenido de <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>
- Castillo, A. (2013). Manual de HTML5. En A. Castillo, *Manual de HTML5*. Theproc.
- Developer\_Network. (2014). *Controladores y metodos de acción en apliccaiones MVC*. Obtenido de [http://msdn.microsoft.com/es-es/library/dd410269\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/dd410269(v=vs.100).aspx)
- Developer\_Network. (2014). *DataAnnotations*. Obtenido de <http://msdn.microsoft.com/en-us/library/system.componentmodel.dataannotations.aspx>
- Developer\_Network. (2014). *Referencias de ASP.NET MVC*. Obtenido de [http://msdn.microsoft.com/es-es/library/gg416515\(v=vs.108\).aspx](http://msdn.microsoft.com/es-es/library/gg416515(v=vs.108).aspx)
- Freeman, A. (2013). Pro ASP.NET MVC4. En A. Freeman, *Pro ASP.NET MVC4*. Apress.
- Freman, A. (2013). Pro ASP.NET 4.5 in c#. En A. Freman, *Pro ASP.NET 4.5 in c#*. Apress.
- Gómez, J. (2014). *Analisis del framework ASP.NET MVC*. Ibarra.
- Haack, J. G. (2013). Professional ASP.NET MVC3. En J. G. Haack, *Professional ASP.NET MVC3*. Programmer to Programmer.
- Madrid, U. C. (2012). *MVC TIW*. Obtenido de <http://www.lab.inf.uc3m.es/~a0080802/RAI/index.html>
- Mestras, J. P. (2012). *Programación orientada a objetos*.
- Microsoft. (2014). *Developer Network*. Obtenido de [http://msdn.microsoft.com/es-es/library/dd381412\(v=vs.108\).aspx](http://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx)
- Microsoft. (2014). *MSDN Magazine*. Obtenido de <http://msdn.microsoft.com/es-es/magazine/>

MSDN. (2014). *Entity Framework*. Obtenido de <http://msdn.microsoft.com/es-es/data/ef.aspx>

MSDN. (2014). *Visual Studio*. Obtenido de <http://www.visualstudio.com/>

Palermo, J. (2012). ASP.NET MVC 4 in Action. En J. Palermo, *ASP.NET MVC 4 in Action*. MANNING.

Seekdotnet. (2012). *Seekdotnet.com Blog*. Obtenido de <http://blog.seekdotnet.com/asp-net/>

Vega, J. (2012). Guia HTML5. En J. Vega, *Guia HTML5*. Maestros del web.

Walker, S. (2013). *MUGPERU Blog*. Obtenido de <http://blog.mugperu.com/>

**ANEXOS.**

Los siguientes anexos de este proyecto de tesis se encuentran adjuntos en formato digital:

**ANEXO 1:** Ejemplos prácticos del uso del Framework MVC 5.

**ANEXO 2:** Manual técnico del aplicativo web.

**ANEXO 3:** Manual de usuario del aplicativo web.

**ANEXO 4:** Código fuente del aplicativo web.

**ANEXO 5:** Backup de la base de datos.



**UNIVERSIDAD TÉCNICA DEL NORTE.**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES.**  
**TRABAJO DE GRADO PREVIO LA OBTENCIÓN DEL TÍTULO EN**  
**INGENIERÍA EN SISTEMAS COMPUTACIONALES.**

**TEMA:**

“ANÁLISIS DEL FRAMEWORK ASP.NET MVC 5. PROTOTIPO SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”

**APLICATIVO:**

“SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”

**Autor:** Jimmy Fabricio Gómez Cadena.

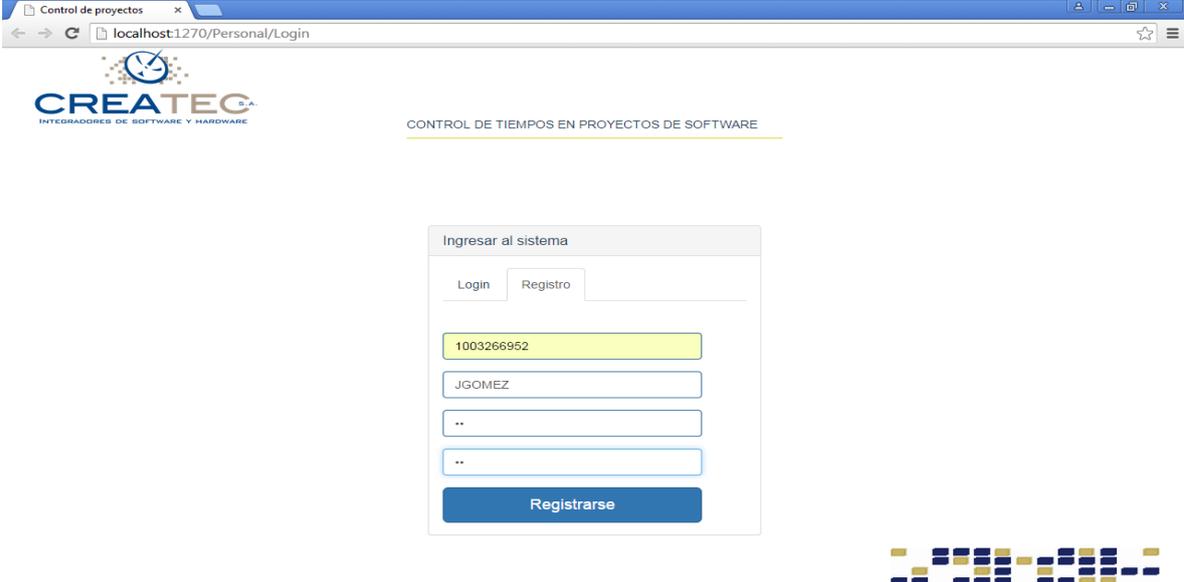
**Director:** Ing. Diego Javier Trejo España.

**Ibarra – Ecuador.**

**2015.**

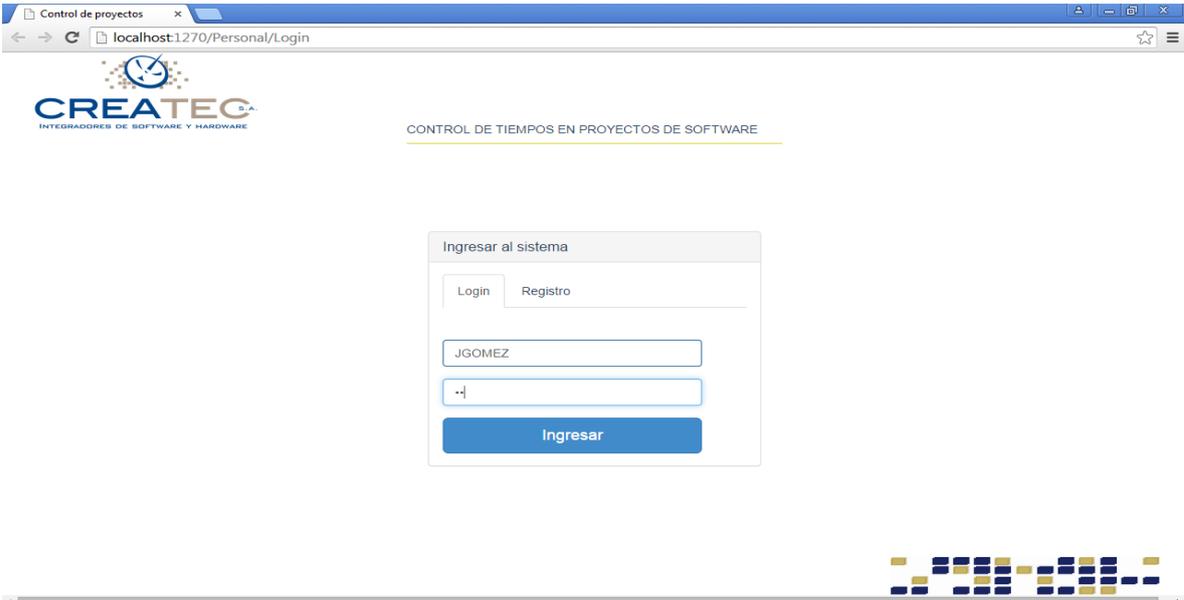
## MANUAL DE USUARIO.

1. Página del inicio del sistema, donde el usuario, si es la primera vez que ingresa a la aplicación podrá registrarse como usuario del sistema, ingresando los datos que se especifica en la figura.



The screenshot shows a web browser window with the address bar displaying "localhost:1270/Personal/Login". The page header includes the logo for "CREATEC S.A. INTEGRADORES DE SOFTWARE Y HARDWARE" and the text "CONTROL DE TIEMPOS EN PROYECTOS DE SOFTWARE". The main content area features a form titled "Ingresar al sistema" with two tabs: "Login" and "Registro". The "Registro" tab is active, showing four input fields: a highlighted field containing "1003266952", a field containing "JGOMEZ", and two empty fields containing "..". A blue "Registrarse" button is positioned below the input fields.

2. Una vez que el usuario ha sido registrado en el sistema, está en la capacidad de hacer Login para ingresar aplicativo ingresando el nombre de usuario y la contraseña.



The screenshot shows the same web browser window as the previous image. The "Ingresar al sistema" form now has the "Login" tab active. The input fields contain "JGOMEZ" and "·|". A blue "Ingresar" button is located below the input fields.

3. En la página de bienvenida del sistema se puede visualizar el menú de los programas que el usuario tiene permisos de utilizar, y en la parte superior derecha de la pantalla se visualiza el nombre del usuario más un menú de los proyectos y actividades en los que está trabajando el usuario.



4. En el menú de administración de la aplicación se puede acceder a las opciones de perfil de usuario, configuraciones de cuenta y cerrar sesión del sistema.



5. En el perfil de usuario se tiene la información a detalle de la persona ingresada en el sistema.

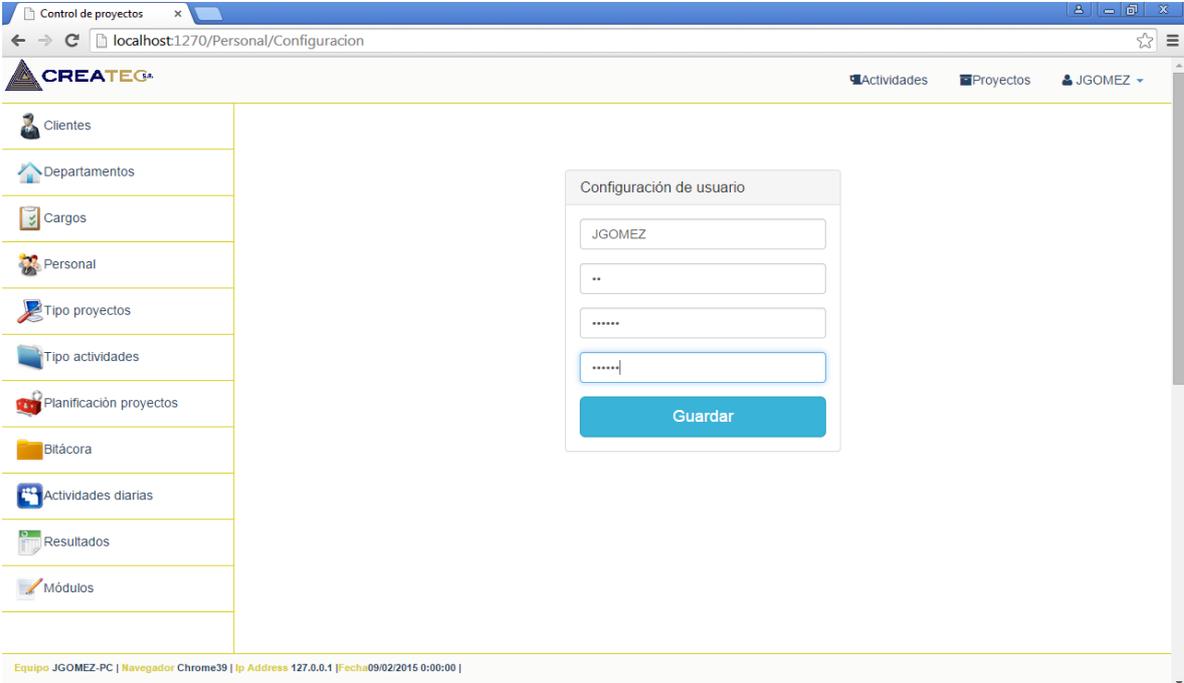


The screenshot shows a web browser window with the URL `localhost:1270/Personal/Perfil_usuario`. The page title is "Perfil de usuario". On the left, there is a sidebar menu with options: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area displays "Información de usuario" with the following details:

<b>Cédula</b>	1003266952
<b>Jefe</b>	Jimmy Gómez
<b>Descripción</b>	Desarrollo
<b>Descripción</b>	Administrador
<b>NOMBRES</b>	Gómez Cadena Jimmy Fabricio
<b>Teléfono</b>	0997790801
<b>Email</b>	jgomez@createc.com.ec
<b>Ciudad/Dirección</b>	El Batan
<b>Usuario</b>	JGOMEZ
<b>Estado</b>	activo

At the bottom of the page, it shows: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |"

6. En la configuración del usuario la persona puede cambiar su nombre y password de la cuenta.



The screenshot shows a web browser window with the URL `localhost:1270/Personal/Configuracion`. The page title is "Configuración de usuario". The sidebar menu is the same as in the previous screenshot. The main content area displays a form for updating user information:

Configuración de usuario

JGOMEZ

..

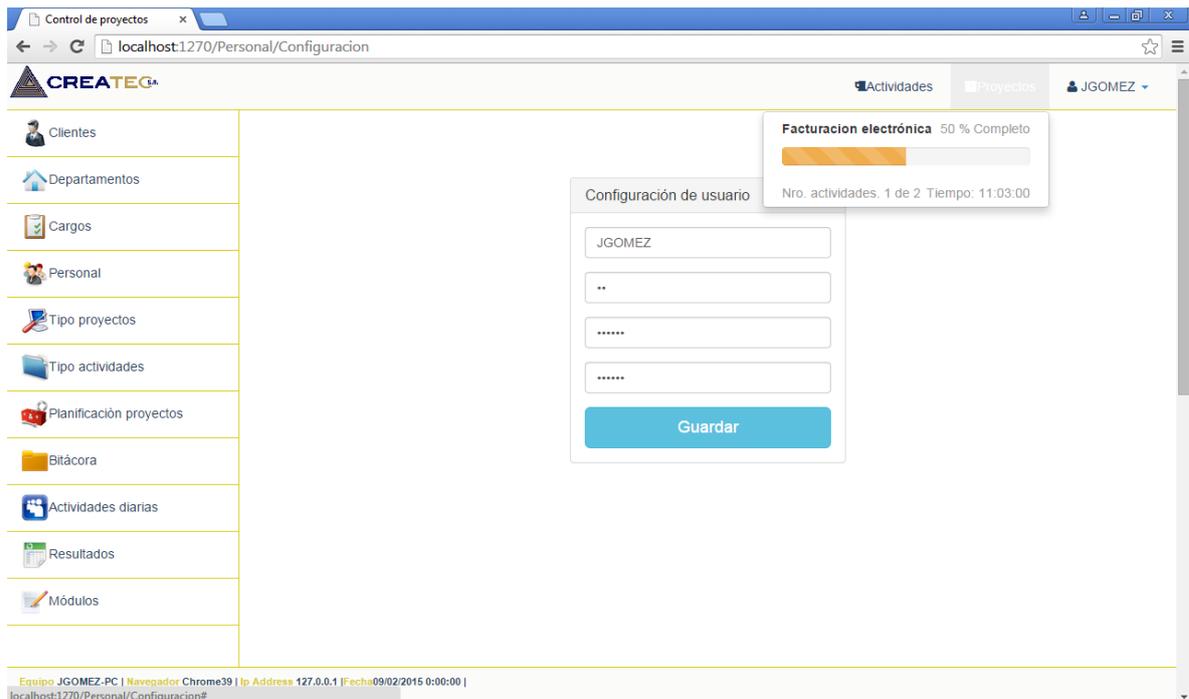
.....

.....|

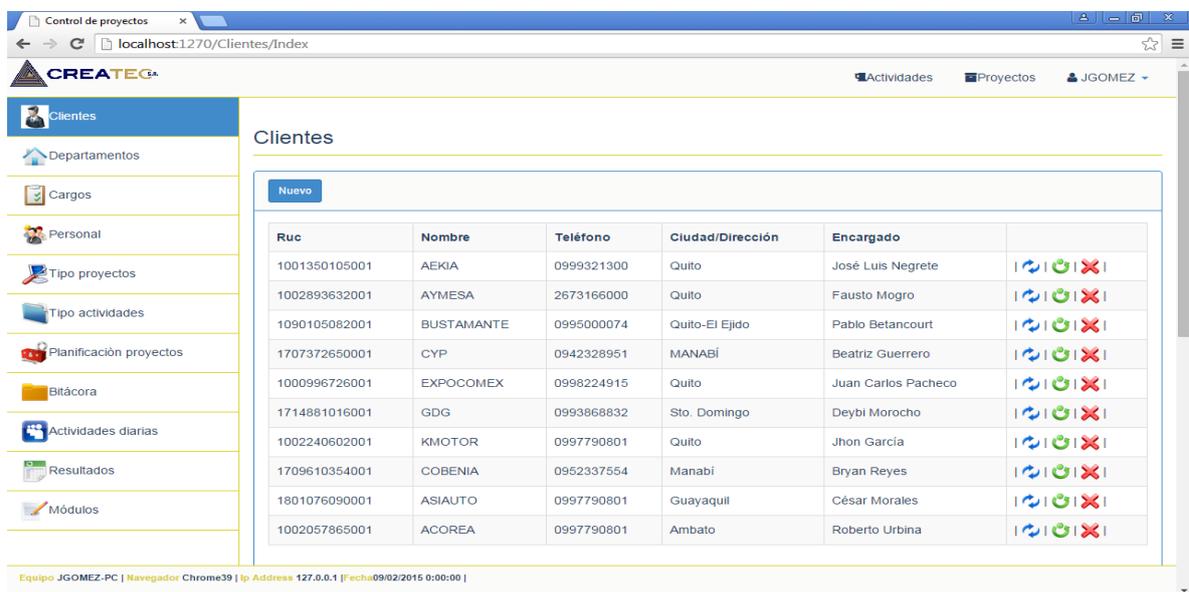
Guardar

At the bottom of the page, it shows: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |"

7. En el menú de proyectos, se visualiza los proyectos y actividades en las que el usuario está trabajando.



8. En el programa Clientes ingresamos y todos los cliente que van a solicitar un desarrollo a la empresa, en la ilustración 8 se muestra la lista de clientes registrados, además de permitir agregar, editar y eliminar clientes.



9. Nuevo cliente.

Control de proyectos x

localhost:1270/Cientes/Create

CREATEG<sup>SA</sup>

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Nuevo Cliente

Atras Guardar

**Ruc**  
1003266952001

**Nombre**  
UTN

**Teléfono**  
0997790801

**Ciudad/Dirección**  
Ibarra, El Olivo

**Encargado**  
Ji

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 10. Editar cliente.

Control de proyectos x

localhost:1270/Cientes/Edit/1001350105001

CREATEG<sup>SA</sup>

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Editar Cliente

Atras Guardar

**Ruc**  
1001350105001

**Nombre**  
AEKIA

**Teléfono**  
0999321300

**Ciudad/Dirección**  
Quito

**Encargado**  
José Luis Negrete

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |  
localhost:1270/Cientes/Edit/1001350105001

## 11. Detalle del cliente.

Control de proyectos x

localhost:1270/Cientes/Details/1001350105001

CREATEG<sup>SA</sup>

Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Detalle Cliente

Atras Editar

<b>Ruc</b>	1001350105001
<b>Nombre</b>	AEKIA
<b>Teléfono</b>	0999321300
<b>Ciudad/Dirección</b>	Quito
<b>Encargado</b>	José Luis Negrete

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 12. Eliminar cliente.

Control de proyectos x

localhost:1270/Cientes/Delete/1001350105001

CREATEG<sup>SA</sup>

Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### ¿Desea eliminar cliente?

Atras Eliminar

<b>Ruc</b>	1001350105001
<b>Nombre</b>	AEKIA
<b>Teléfono</b>	0999321300
<b>Ciudad/Dirección</b>	Quito
<b>Encargado</b>	José Luis Negrete

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

13. El sistema permite trabajar con los departamentos que existen en CREATEG, este programa puede agregar, editar y eliminar departamentos.

Control de proyectos x  
localhost:1270/Departamentos/Index

CREATÉG<sup>SA</sup> Actividades Proyectos JGOMEZ

Cientes  
Departamentos  
Cargos  
Personal  
Tipo proyectos  
Tipo actividades  
Planificación proyectos  
Bitácora  
Actividades diarias  
Resultados  
Módulos

### Departamentos

Nuevo

Descripción	Acciones
Gerencia	
Desarrollo	
Contabilidad	
Soporte técnico	
Recursos humanos	
Investigación	

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 14. Nuevo departamento.

Control de proyectos x  
localhost:1270/Departamentos/Create

CREATÉG<sup>SA</sup> Actividades Proyectos JGOMEZ

Cientes  
Departamentos  
Cargos  
Personal  
Tipo proyectos  
Tipo actividades  
Planificación proyectos  
Bitácora  
Actividades diarias  
Resultados  
Módulos

### Nuevo Departamento

Atras Guardar

Descripción  
GERENCIA

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 15. Editar departamento

Control de proyectos x

localhost:1270/Departamentos/Edit/6

CREATEG®

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Editar Departamento

Atras Guardar

**Descripción**

Gerencia

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 16. Detalle departamento.

Control de proyectos x

localhost:1270/Departamentos/Details/6

CREATEG®

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

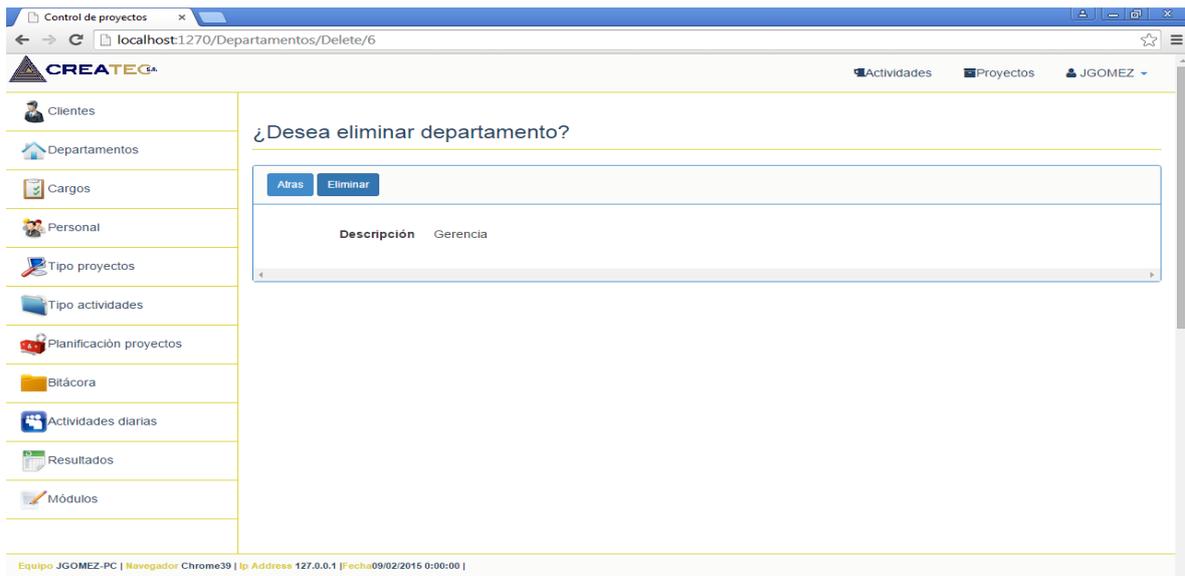
### Detalle Departamento

Atras Editar

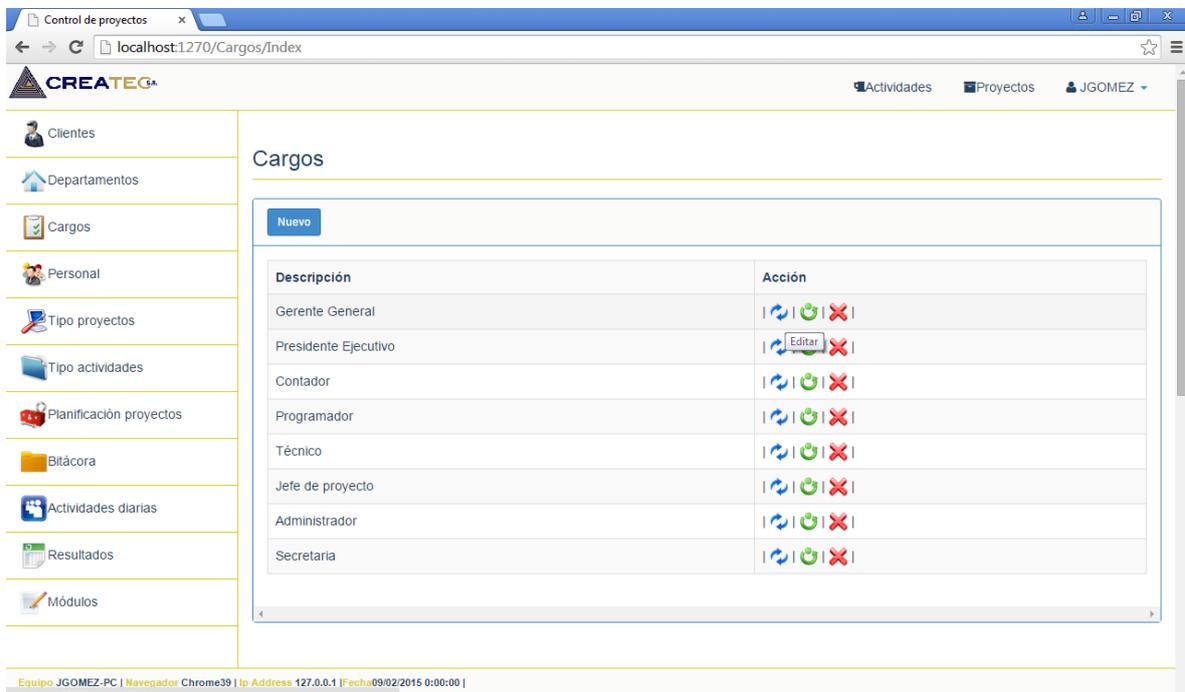
**Descripción** Gerencia

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 17. Eliminar departamento.



18. El sistema permite además trabajar con cargos del personal de CREATEC, permite agregar, editar y eliminar cargos.



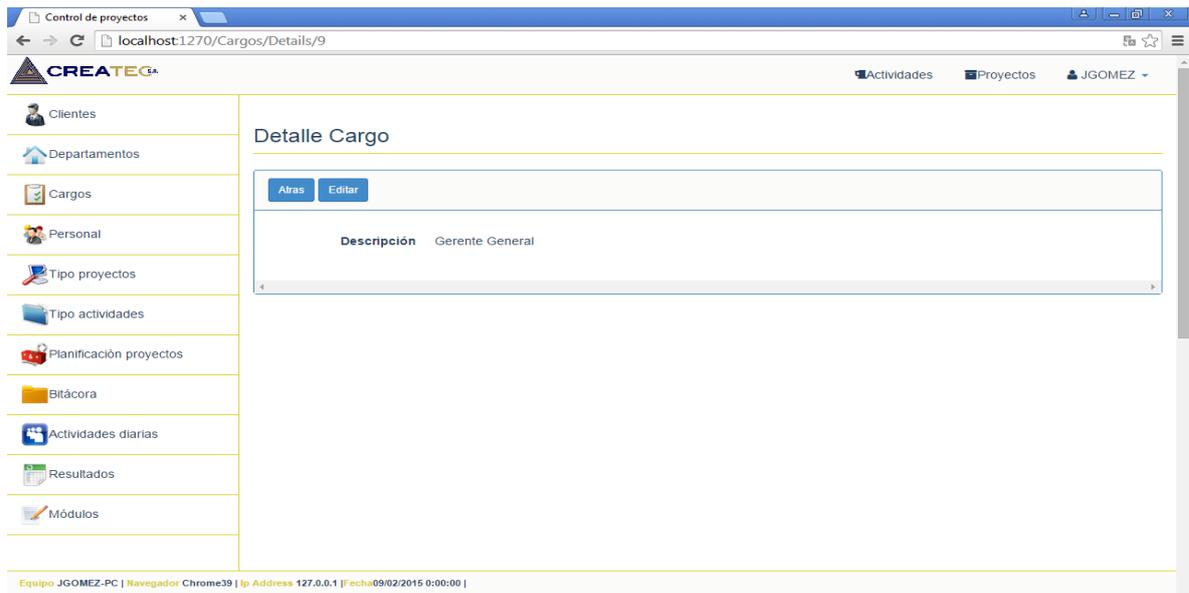
19. Nuevo cargo

The screenshot shows a web browser window with the URL `localhost:1270/Cargos/Create`. The page title is "Nuevo Cargo". On the left, there is a sidebar menu with the following items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area contains a form with two buttons at the top: "Atras" and "Guardar". Below the buttons is a text input field labeled "Descripción" with the value "Gerente". At the bottom of the browser window, the status bar reads: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

## 20. Editar cargo.

The screenshot shows a web browser window with the URL `localhost:1270/Cargos/Edit/9`. The page title is "Editar Cargo". The sidebar menu is identical to the previous screenshot. The main content area contains a form with two buttons at the top: "Atras" and "Guardar". Below the buttons is a text input field labeled "Descripción" with the value "Gerente General". At the bottom of the browser window, the status bar reads: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

## 21. Detalle de cargo.

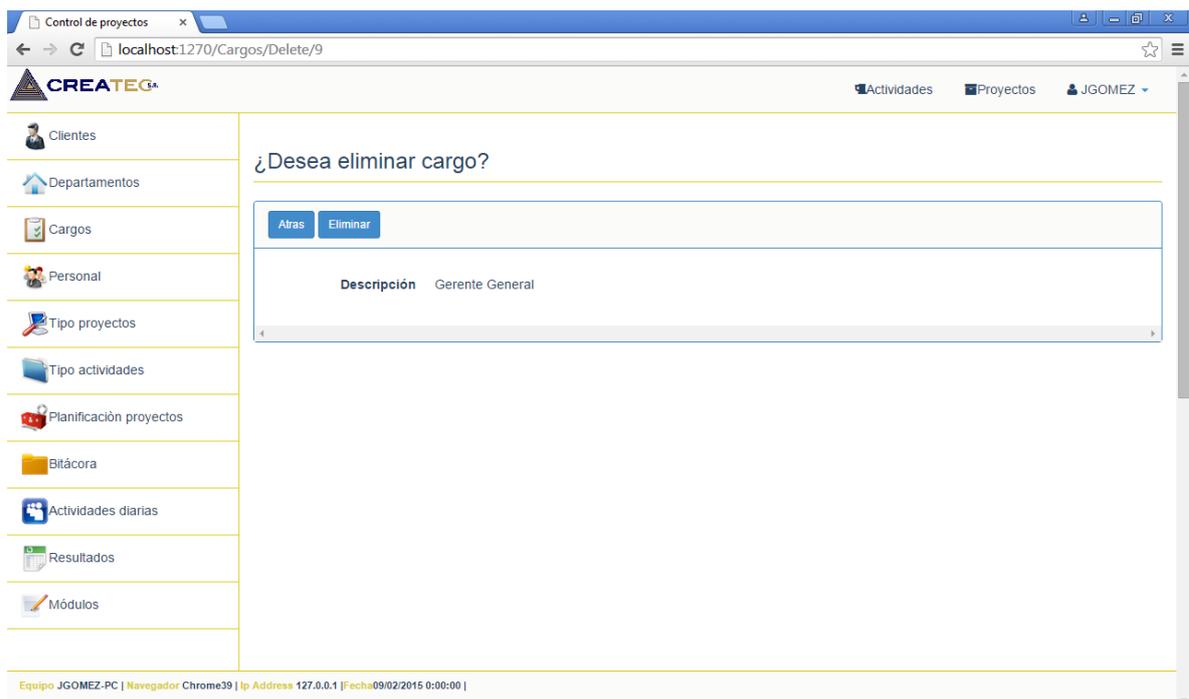


The screenshot shows a web browser window with the URL `localhost:1270/Cargos/Details/9`. The page title is "Detalle Cargo". On the left, there is a sidebar menu with the following items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area contains a table with the following structure:

Descripción
Gerente General

At the top of the table, there are two buttons: "Atras" and "Editar". The status bar at the bottom indicates: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

## 22. Eliminar cargo.



The screenshot shows a web browser window with the URL `localhost:1270/Cargos/Delete/9`. The page title is "¿Desea eliminar cargo?". On the left, there is a sidebar menu with the following items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area contains a table with the following structure:

Descripción
Gerente General

At the top of the table, there are two buttons: "Atras" and "Eliminar". The status bar at the bottom indicates: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

23. El sistema lleva el registro del personal que trabaja en la empresa, permite agregar, editar y eliminar cargos.

Personal de la empresa

Nuevo

<input type="checkbox"/>	Gerencia							
	1001430816	Laura Gomez	Gerente General	activo				
<input type="checkbox"/>	Desarrollo							
	1003266952	Jimmy Gómez	Administrador	activo				
	0401015227	Mercedes Gutierrez	Programador	activo				
	1001883857	Ely Franco	Programador	activo				
	0400545901	Esteban Ortega	Programador	activo				
<input type="checkbox"/>	Contabilidad							
	1000967933	Verónica Naranjo	Contador	activo				
	1000149482	Alejandra Ruiz	Contador	activo				
<input type="checkbox"/>	Soporte técnico							
	1003134234	Juan Pasquel	Técnico	activo				
<input type="checkbox"/>	Recursos humanos							
	1002994539	Ruth Ramirez	Secretaria	activo				

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 | localhost:1270/Personal/Index

24. Nuevo empleado.

Nuevo Empleado

Atras Guardar

**Cédula**  
1003266952

**Jefe**  
Juan Pasquel

**Departamento**  
Gerencia

**Cargo**  
Gerente General

**Primer apellido**  
Gómez

**Segundo apellido**  
Cadena

**Permisos**

- Clientes
- Departamentos
- Cargos
- Personal
- Tipos proyecto
- Tipos actividad
- Módulos
- Planificación proyectos
- Bitácora
- Actividades diarias
- Resultados

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 | localhost:1270/Personal/Create

25. Editar empleado.

Control de proyectos x

localhost:1270/Personal/Edit/1001430816

CREATEG<sup>+</sup> Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Editar Empleado

Atras Guardar

**Cédula**  
1001430816

**Jefe**  
Alejandra Ruiz

**Departamento**  
Gerencia

**Cargo**  
Gerente General

**Primer apellido**  
Gomez

**Segundo apellido**  
Posso

**Permisos**

- Actividades diarias
- Resultados
- Cientes
- Departamentos
- Cargos
- Personal
- Tipos proyecto
- Tipos actividad
- Módulos
- Planificación proyectos
- Bitácora

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 26. Detalle del empleado.

Control de proyectos x

localhost:1270/Personal/Details/1001430816

CREATEG<sup>+</sup> Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

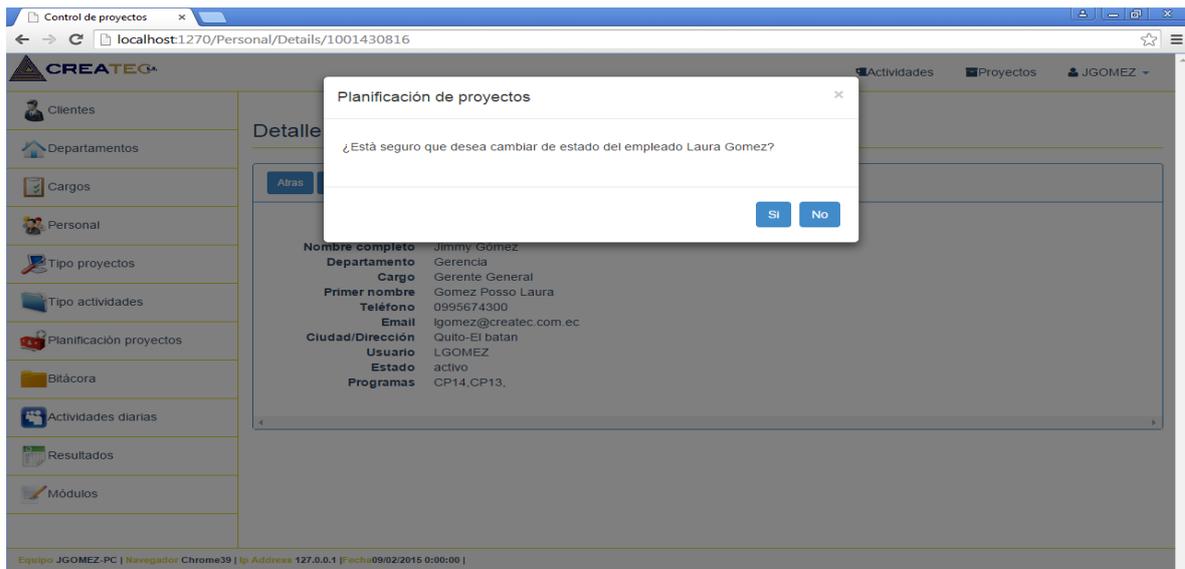
### Detalle Empleado

Atras Editar activo

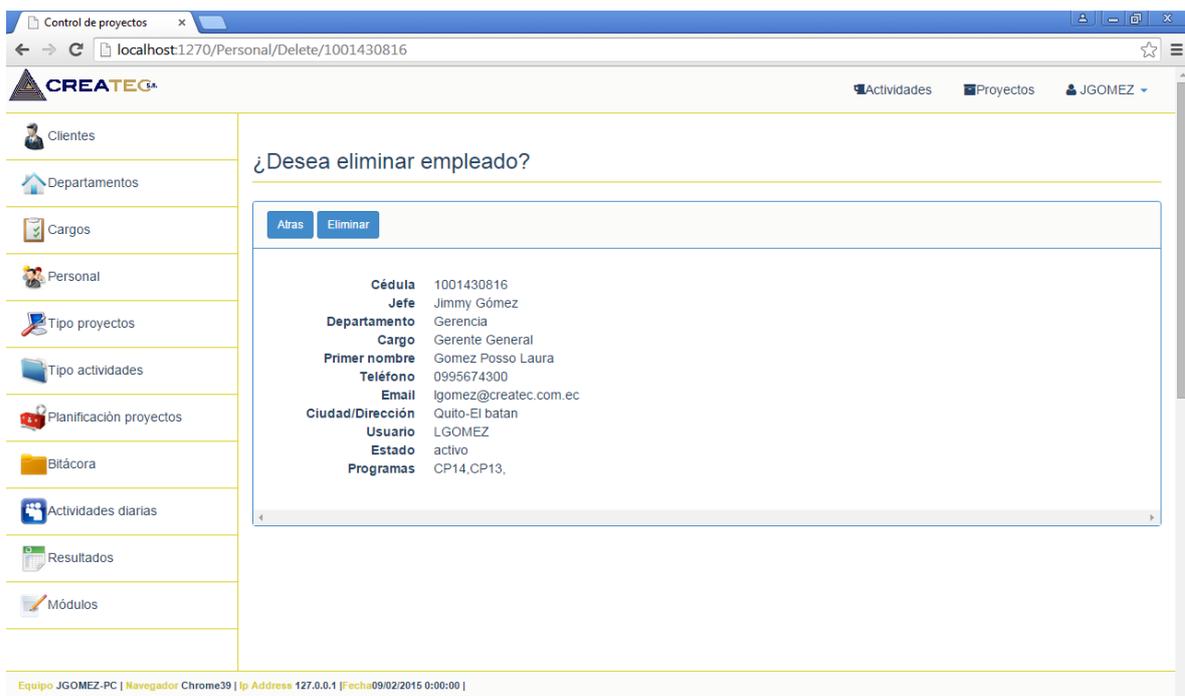
**Cédula** 1001430816  
**Nombre completo** Jimmy Gómez  
**Departamento** Gerencia  
**Cargo** Gerente General  
**Primer nombre** Gomez Posso Laura  
**Teléfono** 0995674300  
**Email** lgomez@createc.com.ec  
**Ciudad/Dirección** Quito-EI batan  
**Usuario** LGOMEZ  
**Estado** activo  
**Programas** CP14,CP13,

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

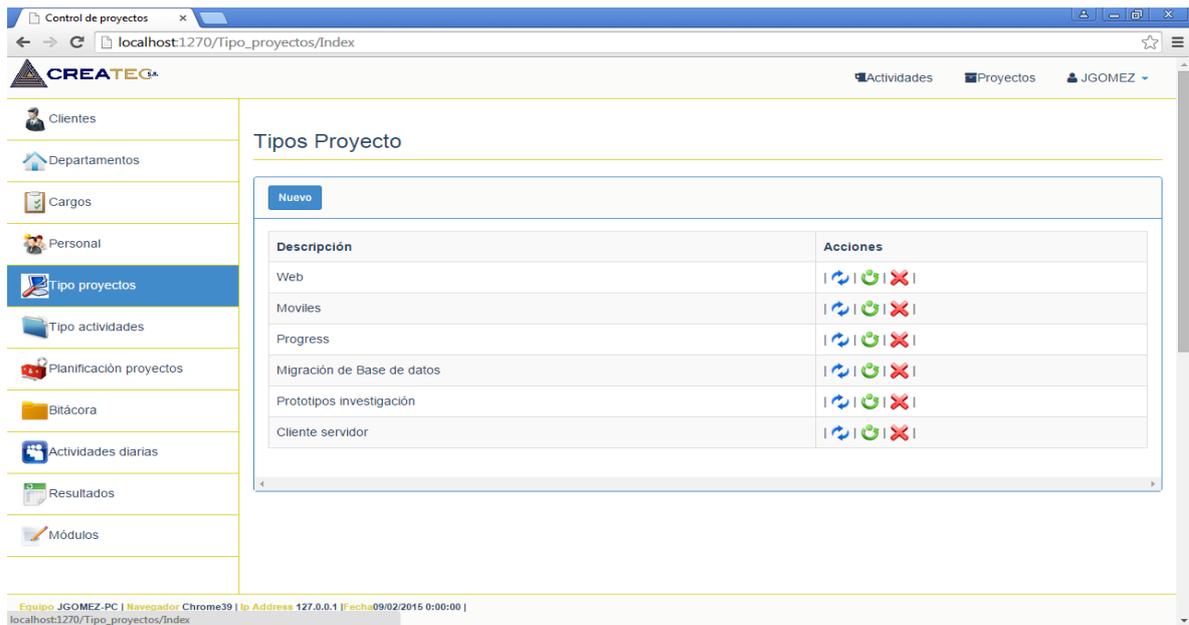
## 27. Además permite dar de baja a empleados de la empresa.



## 28. Eliminar empleado.



29. El sistema permite trabajar on tipos de proyectos ejecutados en CTREATEC, permite agregar, editar y eliminar tipos de proyectos.



Control de proyectos x localhost:1270/Tipo\_proyectos/Index

CREATEG®

Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

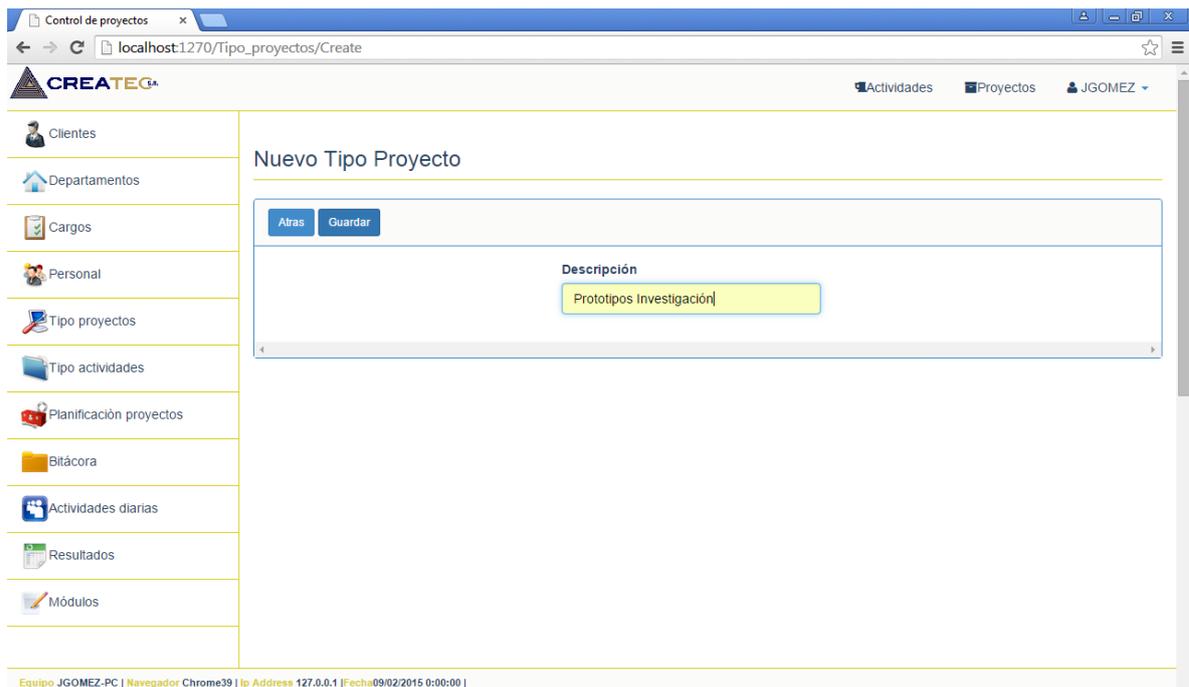
### Tipos Proyecto

Nuevo

Descripción	Acciones
Web	↺   ↻   ✖
Moviles	↺   ↻   ✖
Progress	↺   ↻   ✖
Migración de Base de datos	↺   ↻   ✖
Prototipos investigación	↺   ↻   ✖
Cliente servidor	↺   ↻   ✖

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 | localhost:1270/Tipo\_proyectos/Index

### 30. Nuevo tipo de proyecto.



Control de proyectos x localhost:1270/Tipo\_proyectos/Create

CREATEG®

Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

### Nuevo Tipo Proyecto

Atras Guardar

Descripción

Prototipos Investigación

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

### 31. Editar tipo de proyecto.

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_proyectos/Edit/1`. The page title is "Editar Tipo Proyecto". On the left, there is a sidebar menu with items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area has a header with "Atras" and "Guardar" buttons. Below this is a form with a label "Descripción" and a text input field containing the word "Web". At the bottom of the browser window, the status bar shows: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

### 32. Detalle de tipo de proyecto.

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_proyectos/Details/1`. The page title is "Detalle Tipo Proyecto". The sidebar menu is identical to the previous screenshot. The main content area has a header with "Atras" and "Editar" buttons. Below this is a table with one row: 

Descripción
Web

. At the bottom of the browser window, the status bar shows: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

### 33. Eliminar tipo de proyecto.

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_proyectos/Delete/1`. The application header includes the logo 'CREATEG S.A.' and navigation links for 'Actividades', 'Proyectos', and the user 'JGOMEZ'. A sidebar on the left contains menu items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area displays the question '¿Desea eliminar Tipo proyecto?' with two buttons: 'Atras' and 'Eliminar'. Below this is a table with one row containing the text 'Descripción Web'.

34. El sistema permite definir los tipos de actividades existentes en la empresa, este programa contiene las acciones Crear, editar y eliminar tipos de actividades

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_actividades/Index`. The application header and sidebar are the same as in the previous screenshot. The main content area is titled 'Tipos Actividad' and features a 'Nuevo' button. Below the button is a table listing various activity types and their corresponding actions.

Descripción	Acciones
Planificación	
Documnetación	
De campo	
Desarrollo	
Investigación	
Pruebas	
Soporte técnico	
Mantenimiento sistema	
Mantenimiento BDD	
OTROS	

35. Nuevo tipo de actividad

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_actividades/Create`. The page title is "Nuevo Tipo Actividad". On the left, there is a sidebar menu with items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades (highlighted), Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area has a header with "Actividades", "Proyectos", and "JGOMEZ" dropdowns. Below the header, there are two buttons: "Atras" and "Guardar". A text input field labeled "Descripción" contains the text "Documentacion". At the bottom of the browser window, a status bar reads: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

### 36. Editar tipo de actividad

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_actividades/Edit/1`. The page title is "Editar Tipo Actividad". The sidebar menu is identical to the previous screenshot, with "Tipo actividades" highlighted. The main content area has the same header and buttons ("Atras", "Guardar"). The "Descripción" input field now contains the text "Planificación". The status bar at the bottom remains the same: "Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |".

### 37. Detalle de tipo de actividad.

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_actividades/Details/1`. The application header includes the logo for 'CREATEG' and navigation links for 'Actividades', 'Proyectos', and the user 'JGOMEZ'. A sidebar on the left contains menu items: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades (highlighted), Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area is titled 'Detalle Tipo actividad' and features a table with two columns: 'Descripción' and 'Planificación'. Above the table are buttons for 'Atras' and 'Editar'. The status bar at the bottom indicates the user is on 'Equipo JGOMEZ-PC' using 'Navegador Chrome39' at IP address '127.0.0.1' on 'Fecha 09/02/2015 0:00:00'.

### 38. Eliminar actividad

The screenshot shows a web browser window with the URL `localhost:1270/Tipo_actividades/Delete/1`. The application header and sidebar are identical to the previous screenshot. The main content area is titled '¿Desea eliminar tipo actividad?' and features a table with two columns: 'Descripción' and 'Planificación'. Above the table are buttons for 'Atras' and 'Eliminar'. The status bar at the bottom is the same as in the previous screenshot.

39. El sistema está en l capacidad de planificar proyectos de los cuales se va a llevar el control de los tiempos de los empleados de la empresa.

Control de proyectos x  
localhost:1270/Proyectos/Index

CREATEG<sup>SA</sup> Actividades Proyectos JGOMEZ

Clientes  
Departamentos  
Cargos  
Personal  
Tipo proyectos  
Tipo actividades  
Planificación proyectos  
Bitácora  
Actividades diarias  
Resultados  
Módulos

### Proyectos

Nuevo

Proyecto	Nombre del proyecto	Descripción	Estado	
Cliente servidor	Facturacion electrónica	Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI	Iniciado	🔄 🟢 🛑 📄

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 | localhost:1270/Proyectos/Index

## 40. Nuevo proyecto

Control de proyectos x  
localhost:1270/Proyectos/Create

CREATEG<sup>SA</sup> Actividades Proyectos JGOMEZ

Clientes  
Departamentos  
Cargos  
Personal  
Tipo proyectos  
Tipo actividades  
Planificación proyectos  
Bitácora  
Actividades diarias  
Resultados  
Módulos

### Nuevo Proyecto

Atras Guardar

**Proyecto**  
Web

**Empleado**  
Jimmy Gómez

**Cliente**  
AEKIA

**Nombre del proyecto**  
CONTROL DE PROYECTOS

**Horas estimadas**  
15

**Descripción**  
Proyecto interno de la empresa

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 41. Editar proyecto

Control de proyectos x

localhost:1270/Proyectos/Edit/2

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

### Editar Proyecto

Atras Guardar

**Proyecto**  
Web

**Empleado**  
Jimmy Gómez

**Cliente**  
AEKIA

**Nombre del proyecto**  
FACTURACION ELECTRÓNICA

**Horas estimadas**  
200

**Descripción**  
Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI

**Estado**  
Iniciado

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 42. Detalle del proyecto

Control de proyectos x

localhost:1270/Proyectos/Details/2

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

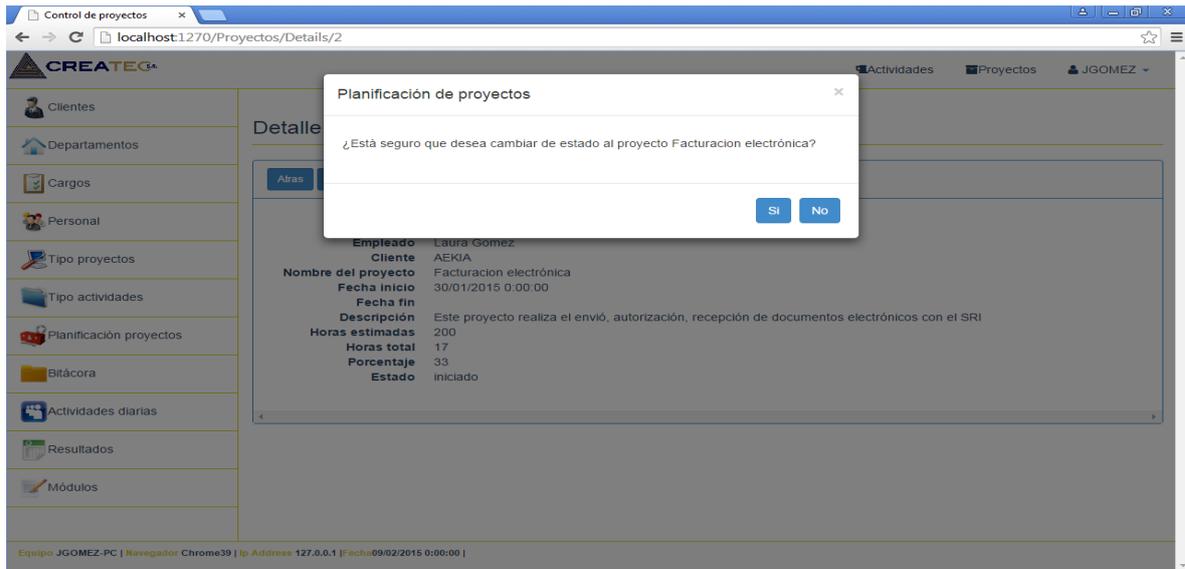
### Detalle Proyecto

Atras Editar Iniciado

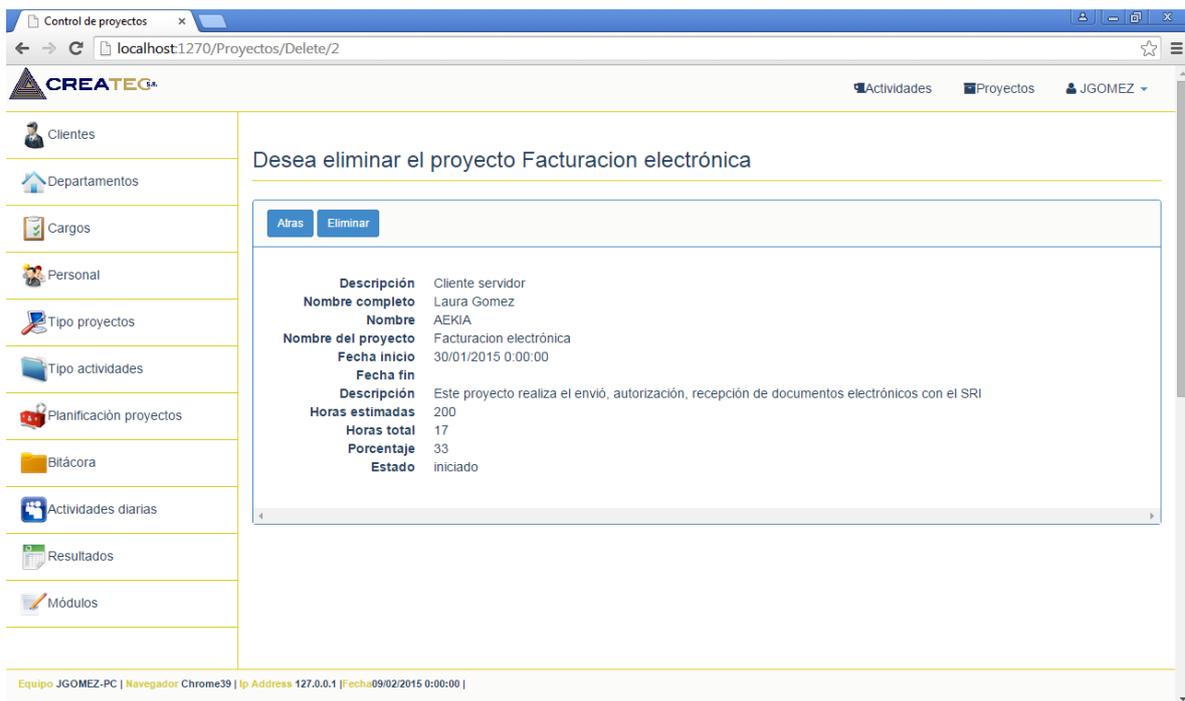
<b>Proyecto</b>	Cliente servidor
<b>Empleado</b>	Laura Gomez
<b>Cliente</b>	AEKIA
<b>Nombre del proyecto</b>	Facturación electrónica
<b>Fecha inicio</b>	30/01/2015 0:00:00
<b>Fecha fin</b>	
<b>Descripción</b>	Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI
<b>Horas estimadas</b>	200
<b>Horas total</b>	17
<b>Porcentaje</b>	33
<b>Estado</b>	Iniciado

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 43. Cambiar estado del proyecto (pendiente, planificado, iniciado, terminado).



#### 44. Eliminar proyectos.



#### 45. El sistema permite agregar actividades en las que el personal va a trabajar dentro del proyecto, además pueden crearse sub actividades.

Equipos: JGOMEZ-PC | Navegador: Chrome39 | Ip Address: 127.0.0.1 | Fecha: 09/02/2015 0:00:00 | localhost:1270/Proyectos/Enviar\_id\_proyecto/2

Nombre	Fecha planificada	Tiempo	Estado
<input type="checkbox"/> Levantamiento de requerimientos	30/01/2015 0:00:00	03:00:00	terminada
Reunión en el SRI	30/01/2015 0:00:00	01:10:00	terminada
<input type="checkbox"/> Doc. Ficha técnica	31/01/2015 0:00:00	05:02:00	terminada
Leer ficha técnica	31/01/2015 0:00:00	03:11:00	terminada
<input type="checkbox"/> Doc. XML	01/02/2015 0:00:00	01:02:00	terminada
Estudiar XML	01/02/2015 0:00:00	03:11:00	terminada
Definición de procesos	02/02/2015 0:00:00	03:02:00	pendiente
<input type="checkbox"/> Desarrollo BDD	08/02/2015 0:00:00	03:01:00	pendiente
TABLAS	04/02/2015 0:00:00	02:01:00	pendiente
DICCIONARIO DE DATOS	06/02/2015 0:00:00	03:02:00	pendiente
PROGRAMACIÓN	08/02/2015 0:00:00	02:01:00	pendiente
PRUEBAS	15/02/2015 0:00:00	01:01:00	pendiente
Soporte	21/03/2015 0:00:00	01:01:00	pendiente

#### 46. Nueva actividad.

Equipos: JGOMEZ-PC | Navegador: Chrome39 | Ip Address: 127.0.0.1 | Fecha: 09/02/2015 0:00:00 |

**Tipo de actividad**  
Planificación

**Empleado**  
Jimmy Gómez

**Descripción**  
Documentación

**Fecha planificada**  
18/02/2015

**Tiempo estimado**  
01:00

#### 47. Editar actividad.

The screenshot shows the 'Editar Actividad' (Edit Activity) form in the CREATÉG system. The browser address bar shows 'localhost:1270/Actividades/Edit/1'. The left sidebar contains a navigation menu with items like 'Clientes', 'Departamentos', 'Cargos', 'Personal', 'Tipo proyectos', 'Tipo actividades', 'Planificación proyectos', 'Bitácora', 'Actividades diarias', 'Resultados', and 'Módulos'. The main content area has a title 'Editar Actividad' and two buttons: 'Atras' and 'Guardar'. Below the buttons are several form fields:

- Tipo de actividad:** A dropdown menu with 'Planificación' selected.
- Empleado:** A dropdown menu with 'Jimmy Gómez' selected.
- Descripción:** A text input field containing 'Levantamiento de requerimientos'.
- Fecha planificación:** A date and time input field showing '30/01/2015 0:00:00'.
- Tiempo estimado:** A text input field showing '03:00'.

At the bottom of the page, a status bar displays: 'Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |'.

#### 48. Detalle de la actividad

The screenshot shows the 'Detalle Actividad' (Activity Details) view in the CREATÉG system. The browser address bar shows 'localhost:1270/Actividades/Details/1'. The left sidebar is identical to the previous screenshot. The main content area has a title 'Detalle Actividad' and two buttons: 'Atras' and 'Editar'. Below the buttons is a table displaying the details of the activity:

<b>Actividad</b>	1
<b>Cédula</b>	0401015227
<b>Proyecto</b>	Facturación electrónica
<b>Tipo de actividad</b>	Planificación
<b>Descripción</b>	Levantamiento de requerimientos
<b>Fecha planificación</b>	30/01/2015 0:00:00
<b>Fecha inicio</b>	
<b>Fecha fin</b>	28/01/2015 0:00:00
<b>Tiempo estimado</b>	03:00:00
<b>Tiempo total</b>	06:02:00
<b>Estado</b>	terminada

At the bottom of the page, a status bar displays: 'Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |'.

#### 49. Eliminar actividad.

Control de proyectos x

localhost:1270/Actividades/Delete/1

CREATEGA

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

¿Desea eliminar la actividad Levantamiento de requerimientos?

Atras Eliminar

Actividad 1  
 Jefe 1003266952  
 Proyecto 7  
 Descripción Planificación  
 Descripción Levantamiento de requerimientos  
 Fecha planificación 30/01/2015 0:00:00  
 Fecha inicio 28/01/2015 0:00:00  
 Tiempo estimado 03:00:00  
 Tiempo total 06:02:00  
 Estado terminada

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 50. Agregar entregables a las actividades.

Control de proyectos x

localhost:1270/Actividades/Index

CREATEGA

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

Nombre	Fecha planificada	Tiempo	Estado	
<input type="checkbox"/> Levantamiento de requerimientos	30/01/2015 0:00:00	03:00:00	terminada	
Reunión en el SRI	30/01/2015 0:00:00	01:10:00	terminada	
<input type="checkbox"/> Doc. Ficha técnica	31/01/2015 0:00:00	05:02:00	terminada	Entregables
Leer ficha técnica	31/01/2015 0:00:00	03:11:00	terminada	
<input type="checkbox"/> Doc. XML	01/02/2015 0:00:00	01:02:00	terminada	
Estudiar XML	01/02/2015 0:00:00	03:11:00	terminada	
Definición de procesos	02/02/2015 0:00:00	03:02:00	pendiente	
<input type="checkbox"/> Desarrollo BDD	08/02/2015 0:00:00	03:01:00	pendiente	
TABLAS	04/02/2015 0:00:00	02:01:00	pendiente	
DICCIONARIO DE DATOS	06/02/2015 0:00:00	03:02:00	pendiente	
PROGRAMACIÓN	08/02/2015 0:00:00	02:01:00	pendiente	
PRUEBAS	15/02/2015 0:00:00	01:01:00	pendiente	
Soporte	21/03/2015 0:00:00	01:01:00	pendiente	

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |  
 localhost:1270/Actividades/Enviar\_id\_actividad/2

## 51. Lista de entregables de la actividad.

Control de proyectos x

localhost:1270/Entregables/Index

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Entregables de la actividad Reunión en el SRI proyecto Facturacion electrónica

Atras Nuevo

Descripción	Fecha estimada	Fecha entrega	Rebiso	
doc	01/02/2015 0:00:00	30/01/2015 5:21:36	si	↺   ↻   ✖
resultados reunión	22/02/2015 0:00:00		no	↺   ↻   ✖

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 52. Nuevo entregable.

Control de proyectos x

localhost:1270/Entregables/Create

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Nuevo Entregable

Atras Guardar

Descripción

Documentacion

Fecha estimada

26/02/2015 x ↕ ▼

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 53. Editar entregable.

Control de proyectos x

localhost:1270/Entregables/Edit/1

CREATEGA

Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Editar Entregable

Atras Guardar

**Descripción**

Doc

**Fecha estimada**

01/02/2015 0:00:00

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

#### 54. Detalle del entregable.

Control de proyectos x

localhost:1270/Entregables/Details/1

CREATEGA

Actividades Proyectos JGOMEZ

Clientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Detalle Entregable

Atras Editar Revisado

**ID** 1

**Id proyecto** Facturacion electrónica

**Id actividad** Reunión en el SRI

**Descripción** doc

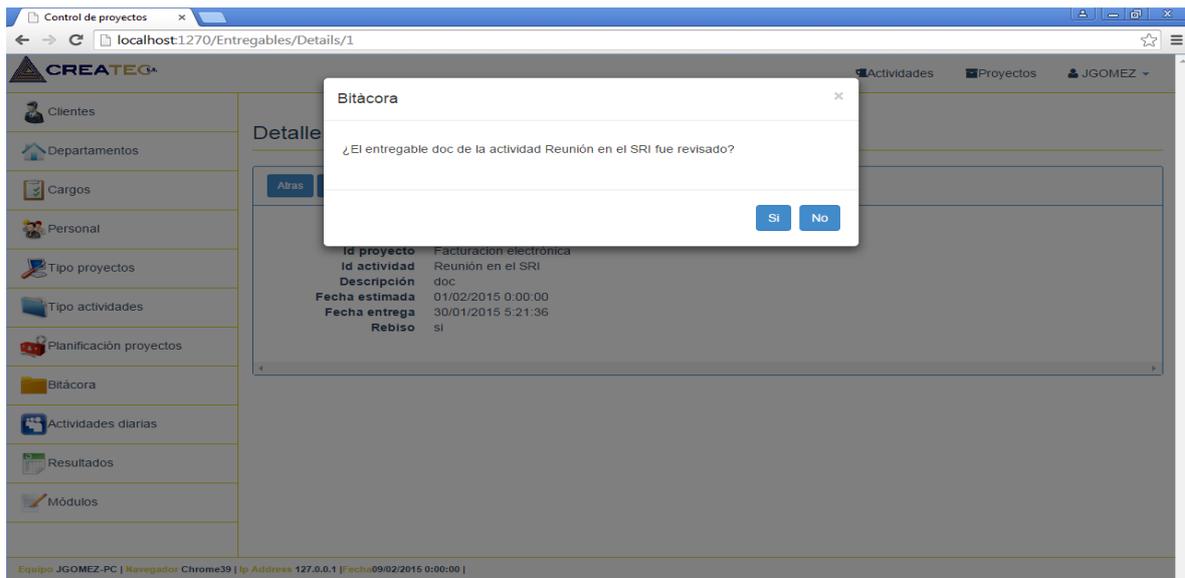
**Fecha estimada** 01/02/2015 0:00:00

**Fecha entrega** 30/01/2015 5:21:36

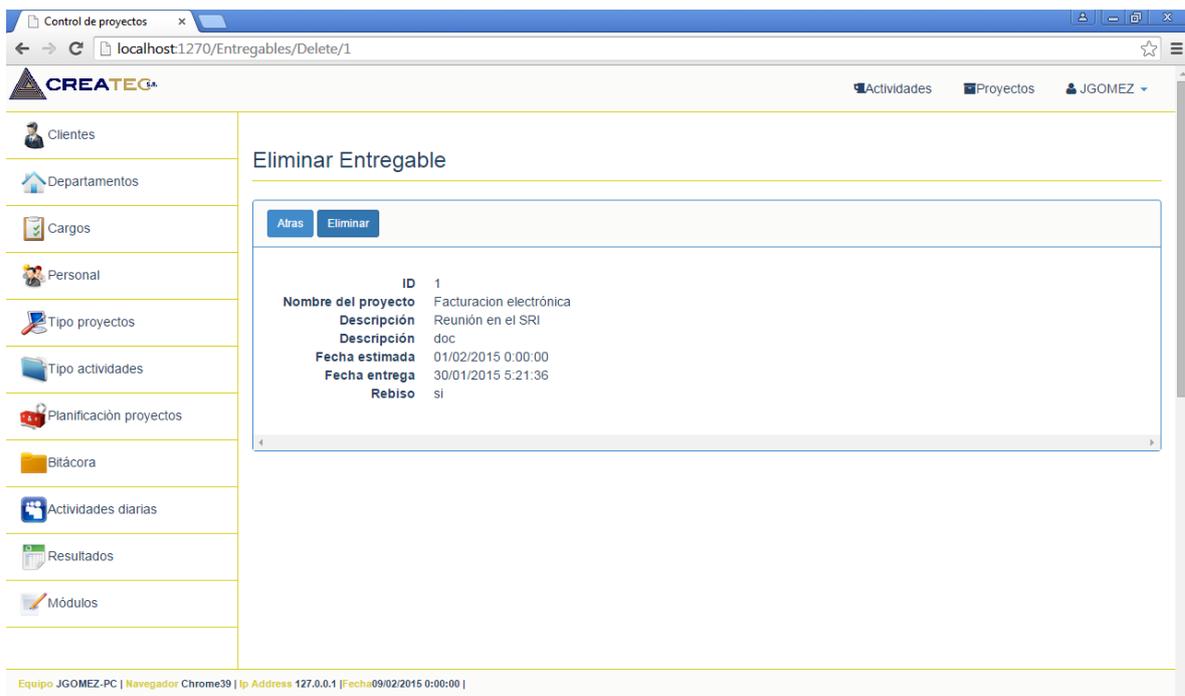
**Revisio** si

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

#### 55. Revisar el entregable.



## 56. Eliminar entregable.



## 57. Bitácora de registro de tiempos empleados en las actividades de determinado proyecto.

Control de proyectos x

localhost:1270/Bitacora/Buscar\_actividades/

CREATEG+

Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

### Bitácora

Proyecto: Facturacion electrónica

Facturacion electrónica

**Descripcion:** Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI

**Cliente:** AEKIA **Fecha fin:**

**Encargado:** Laura Gomez **Horas estimadas:** 200

**Fecha inicio:** 30/01/2015 **Horas trabajadas:** 17

0:00:00 **Porcentaje:** 33

**Estado:** planificado

Detalle Actividad

Facturacion electrónica Bitácora

Equipo: JGOMEZ-PC | Navegador: Chrome39 | Ip Address: 127.0.0.1 | Fecha: 09/02/2015 0:00:00 |

### 58. Árbol de actividades de determinado proyecto.

Control de proyectos x

localhost:1270/Bitacora/Buscar\_actividades/

CREATEG+

Actividades Proyectos JGOMEZ

Cientes Departamentos Cargos Personal Tipo proyectos Tipo actividades Planificación proyectos Bitácora Actividades diarias Resultados Módulos

### Facturacion electrónica

Levantamiento de requerimientos

- Reunión en el SRI

Doc. Ficha técnica

- Leer ficha técnica
- Doc. XML
  - Estudiar XML

Definición de procesos

Desarrollo BDD

- TABLAS
- DICCIONARIO DE DATOS

PROGRAMACIÓN

PRUEBAS

Soporte

Bitácora

Equipo: JGOMEZ-PC | Navegador: Chrome39 | Ip Address: 127.0.0.1 | Fecha: 09/02/2015 0:00:00 |

### 59. Detalle del proyecto y la actividad en la que se va a realizar el registro de los tiempos.

The screenshot shows the 'Bitácora' (Log) section of the CREATeG software. The interface includes a sidebar with navigation options like 'Clientes', 'Departamentos', 'Cargos', 'Personal', 'Tipo proyectos', 'Tipo actividades', 'Planificación proyectos', 'Bitácora', 'Actividades diarias', 'Resultados', and 'Módulos'. The main content area displays details for the project 'Facturación electrónica'.

**Project Search:** Proyecto: Facturación electrónica [buscar]

**Facturación electrónica details:**

- Descripción:** Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI
- Cliente:** AEKIA
- Encargado:** Laura Gomez
- Fecha inicio:** 30/01/2015 0:00:00
- Fecha fin:**
- Horas estimadas:** 200
- Horas trabajadas:** 17
- Porcentaje:** 33
- Estado:** planificado

**Soporte details:**

- Tipo actividad:** Soporte técnico
- Tiempo estimado:** 01:01:00 min
- Fecha planificada:** 21/03/2015 0:00:00
- Tiempo trabajado:** 00:00:00 min
- Fecha inicio:**
- Estado:** pendiente
- Fecha fin:**
- Entregables:** 0
- Evidencias:** 0

Buttons: Iniciar, Terminar

Footer: Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 60. Registro del tiempo en la bitácora de actividades.

The screenshot shows a modal dialog titled 'Soporte' for recording time in the activity log. The dialog contains the following fields:

- Fecha inicio:** 24/02/2015
- Fecha fin:** 24/02/2015
- Tiempo trabajado:** 22:01
- Resultado:** OK

A 'Guardar' (Save) button is located at the bottom right of the dialog.

Background details (partially visible):

- Cliente:** AEKIA
- Encargado:** Laura Gomez
- Fecha inicio:** 30/01/2015 0:00:00
- Tiempo trabajado:** 00:00:00 min
- Estado:** pendiente
- Entregables:** 0

Footer: Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

## 61. Información del tiempo empleado en la actividad.

The screenshot shows the 'Control de proyectos' interface. The left sidebar contains navigation options: Clientes, Departamentos, Cargos, Personal, Tipo proyectos, Tipo actividades, Planificación proyectos, Bitácora, Actividades diarias, Resultados, and Módulos. The main content area is divided into several panels:

- Facturación electrónica (top left):**
  - Descripción:** Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI
  - Cliente:** AEKIA
  - Encargado:** Laura Gomez
  - Fecha inicio:** 30/01/2015 0:00:00
  - Fecha fin:**
  - Horas estimadas:** 200
  - Horas trabajadas:** 18
  - Porcentaje:** 33
  - Estado:** Iniciado
- Soporte (top right):**
  - Tipo actividad:** Soporte técnico
  - Tiempo estimado:** 01:01:00 min
  - Fecha planicada:** 21/03/2015 0:00:00
  - Tiempo trabajado:** 01:01:00 min
  - Fecha inicio:** 02/02/2015 0:00:00
  - Estado:** trabajando
  - Entregables:** 0
  - Fecha fin:**
  - Evidencias:** 0
- Facturación electrónica (bottom left):** A flowchart showing steps: Levantamiento de requerimientos, Reunión en el SRI, Doc. Ficha técnica, and Leer ficha técnica.
- Soporte (bottom right):**
  - Fecha inicio:** 02/02/2015 0:00:00
  - Fecha fin:** 03/02/2015 0:00:00
  - Tiempo trabajado:** 01:01:00
  - Resultado:** OK

At the bottom, a status bar reads: 'Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |'

62. Dar por finalizada una actividad en la que se está trabajando.

This screenshot shows the same project overview as above, but with a modal dialog box titled 'Bitácora' in the center. The dialog contains the text: '¿Está seguro que desea dar por terminada la actividad (Soporte)?' and two buttons: 'Si' and 'No'. The background interface is dimmed.

The status bar at the bottom remains the same: 'Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |'

63. Revisión de los entregables de la actividad.

Control de proyectos x

localhost:1270/Entregables/Subir\_entregables

CREATEG+

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Entregables

Atras

Descripción	Fecha estimada	Fecha entrega	Rebiso
-------------	----------------	---------------	--------

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

64. Revisión de las evidencias registradas en las actividades.

Control de proyectos x

localhost:1270/Evidencias/Index

CREATEG+

Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

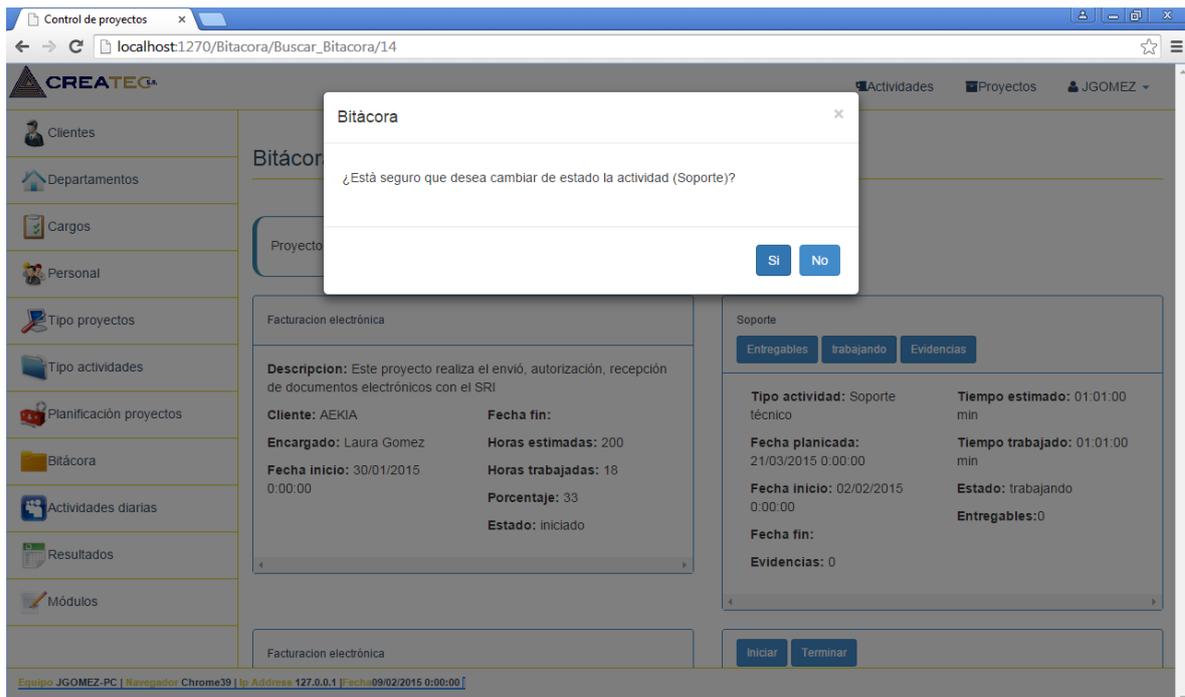
### Evidencias

Atras Nuevo

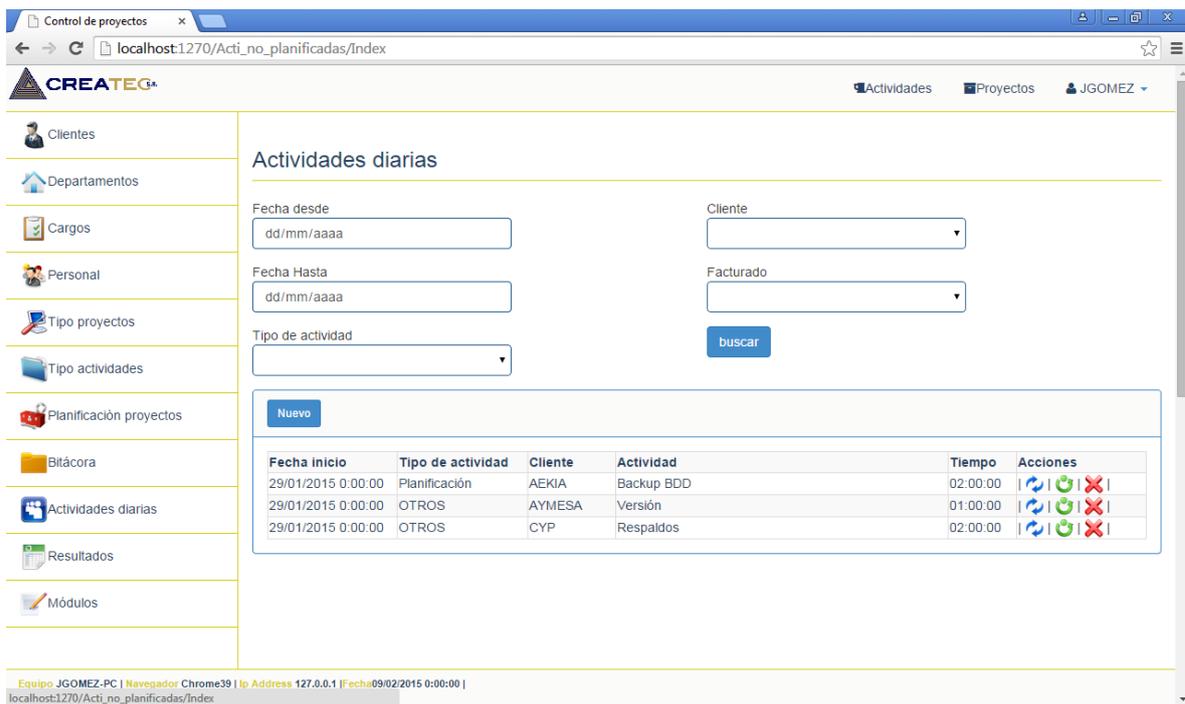
Descripción	FECHA_ENTREGA	PATH_EVIDENCIA	Empleado
-------------	---------------	----------------	----------

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

65. Cambiar estado de las actividades (pendiente, iniciada, terminada)



## 66. Registro de tiempos de actividades diarias no planificadas.



## 67. Agregar nueva actividad.

Control de proyectos x

localhost:1270/Acti\_no\_planificadas/Create

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Nueva actividad

Atras Guardar

**Cliente**  
AEKIA

**Tipo actividad**  
Planificación

**Descripción**  
Backup BDD

**Tiempo total**  
01:00

**Solicitante**  
Jimmy Gómez

**Módulo**  
Activos fijos

**Fecha inicio**  
02/02/2015

**Facturado**  
facturable

**Motivo de facturación**  
interno

**Observaciones**  
BACKUP BDD

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

68. Revisión de resultados de los tiempos registrados en cada proyecto.

Control de proyectos x

localhost:1270/Resultados/Proyectos\_x\_estado/

CREATEG<sup>®</sup> Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

Proyectos planificados Actividades diarias

Revise el resultado de proyectos agrupados por estado

planificado

Buscar

Proyectos

**Facturación electrónica** 33 % Completo

AEKIA Tiempo trabajado: 18

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

69. Detalle de los resultados empleados en cada proyecto.

**Facturación electrónica (Ver reporte)**

**Descripción:** Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI

**Cliente:** AEKIA      **Fecha fin:**

**Encargado:** Laura Gomez      **Horas estimadas:** 200

**Fecha inicio:** 30/01/2015 0:00:00      **Horas trabajadas:** 18

**Porcentaje:** 33

**Estado:** Iniciado

**Reunión en el SRI (Ver reporte)**

**Tipo actividad:** Planificación      **Tiempo estimado:** 01:10:00 min

**Fecha planificada:** 30/01/2015 0:00:00      **Tiempo trabajado:** 06:02:00 min

**Fecha inicio:** 28/01/2015 0:00:00      **Estado:** terminada

**Fecha fin:** 28/01/2015 0:00:00      **Entregables:** 2

**Evidencias:** 1

**Actividades**

```

    graph TD
      A[Levantamiento de requerimientos] --> B[Reunión en el SRI]
      B --> C[Doc. Ficha técnica]
      C --> D[Leer ficha técnica]
      D --> E[Doc. XML]
    
```

**Entregables** | **Evidencias** Mercedes Gutierrez

**Fecha inicio:** 28/01/2015 0:00:00  
**Fecha fin:** 28/01/2015 0:00:00  
**Tiempo trabajado:** 01:01:00  
**Resultado:** Reunion OK

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

70. Reporte detallado de los tiempos empleados en determinado actividad.

**Nombre:** Facturación electrónica      **Fecha fin:**

**Descripción:** Este proyecto realiza el envío, autorización, recepción de documentos electrónicos con el SRI      **Porcentaje:** 33 %

**Tipo proyecto:** Cliente servidor      **Tiempo estimado:** 200

**Cliente:** AEKIA      **Factor ajuste:** -182

**Encargado:** Laura Gomez      **Tiempo total:** 18

**Fecha inicio:** 30/01/2015 0:00:00

**Actividades**

DESCRIPCIÓN	ENCARGADO	FECHA PLANIFICACIÓN	FECHA INICIO	FECHA FIN	ESTIMADO	FACTOR AJUSTE	TOTAL	ESTADO
Levantamiento de requerimientos	0401015227	30/01/2015 0:00:00	28/01/2015 0:00:00	28/01/2015 0:00:00	03:00:00	06:02:00	terminada	Reunión en el SRI
0401015227	30/01/2015 0:00:00	28/01/2015 0:00:00	28/01/2015 0:00:00	01:10:00	06:02:00	terminada	Doc. Ficha técnica	1003266952
31/01/2015 0:00:00	29/01/2015 0:00:00	05:02:00	11:03:00	terminada	Leer ficha técnica	1001883857	31/01/2015 0:00:00	
28/01/2015 0:00:00	28/01/2015 0:00:00	03:11:00	02:02:00	terminada	Doc. XML	1001883857	01/02/2015 0:00:00	
28/01/2015 0:00:00	01:02:00	09:01:00	terminada	Estudiar XML	1001883857	01/02/2015 0:00:00	28/01/2015 0:00:00	
03:11:00	09:01:00	terminada	Definición de procesos	0400545901	02/02/2015 0:00:00			
00:00:00	pendiente	Desarrollo BDD	0400545901	09/02/2015 0:00:00				
pendiente	TABLAS	0400545901	04/02/2015 0:00:00		02:01:00	00:00:00	pendiente	

CP\_control\_proyectos.bak | sesiones (4).xlsx | Mostrar todas las descargas...

71. Reporte de los tiempos empleados en determinado proyecto.

CONTROL DE TIEMPOS EN PROYECTOS DE SOFTWARE

**Descripción:** Reunión en el SRI **Fecha fin:** 28/01/2015 0:00:00

**Proyecto:** Facturacion electrónica **Tiempo estimado:** 01:10:00

**Tipo de actividad:** Planificación **Factor ajuste:** 04:52:00

**Encargado:** Mercedes Gutierrez **Tiempo total:** 06:02:00

**Fecha planificación:** 30/01/2015 0:00:00 **Estado:** terminada

**Fecha inicio:** 28/01/2015 0:00:00

RESULTADO	FECHA INICIO	FECHA FIN	TIEMPO TOTAL
Reunion OK	28/01/2015 0:00:00	28/01/2015 0:00:00	01:01:00
Reunion 2 OK	28/01/2015 0:00:00	28/01/2015 0:00:00	05:01:00

Host: JGOMEZ-PC | Usuario: JGOMEZ | Fecha: 09/02/2015 22:28:07

72. Información del proceso de avance de trabajo de los empleados en determinado proyecto.

Actividades Proyectos JGOMEZ

- Clientes
- Departamentos
- Cargos
- Personal
- Tipo proyectos
- Tipo actividades
- Planificación proyectos
- Bitácora
- Actividades diarias
- Resultados
- Módulos

**Jimmy Fabricio** 0 % Completo  
Nro. actividades finalizadas 0 de 1 Tiempo trabajado: 01:01:00

**Mercedes** 100 % Completo  
Nro. actividades finalizadas 1 de 1 Tiempo trabajado: 06:02:00

**Ely** 100 % Completo  
Nro. actividades finalizadas 2 de 2 Tiempo trabajado: 11:03:00

**Esteban** 0 % Completo

Firefox JGOMEZ-PC | Navegador: Chrome39 | In Address: 127.0.0.1 | Fecha: 09/02/2015 0:00:00 | localhost:1270/Resultados/Buscar\_Bitacora/2#personal

73. Registro de los módulos del sistema Pirámide.

Control de proyectos x localhost:1270/Resultado x localhost:1270/Resultado x localhost:1270/Resultado x

localhost:1270/Modulos/Index

**CREATEG** Actividades Proyectos JGOMEZ

Cientes

Departamentos

Cargos

Personal

Tipo proyectos

Tipo actividades

Planificación proyectos

Bitácora

Actividades diarias

Resultados

Módulos

### Módulos

Nuevo

Código del módulo	Descripción	
AF	Activos fijos	      
FA	Facturación	      
CG	Contabilidad General	      
CA	Contabilidad automática	      
CC	Cuentas por cobrar	      
CP	Cuentas por pagar	      
DE	Documento electrónico	      
TG	Admin General	      
VH	Vehículos	      
IM	Importaciones	      

Equipo JGOMEZ-PC | Navegador Chrome39 | Ip Address 127.0.0.1 | Fecha 09/02/2015 0:00:00 |

CP\_control\_proyectos.bak | sesiones (4).xlsx | [Mostrar todas las descargas...](#)



**UNIVERSIDAD TÉCNICA DEL NORTE.**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES.**  
**TRABAJO DE GRADO PREVIO LA OBTENCIÓN DEL TÍTULO EN**  
**INGENIERÍA EN SISTEMAS COMPUTACIONALES.**

**TEMA:**

SISTEMA PARA

“ANÁLISIS DEL FRAMEWORK ASP.NET MVC 5. PROTOTIPO SISTEMA PARA EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”

**APLICATIVO:**

“EL CONTROL INTERNO DE USO DEL TIEMPO DE LOS RECURSOS HUMANOS DE CREATEC S.A.”

**Autor:** Jimmy Fabricio Gómez Cadena.

**Director:** Ing. Diego Javier Trejo España.

**Ibarra – Ecuador.**

**2015.**

## MANUAL TÉCNICO

A continuación se explica el procedimiento para poner en funcionamiento del sistema para el control interno de uso del tiempo de los recursos humanos de CREATEC S.A.

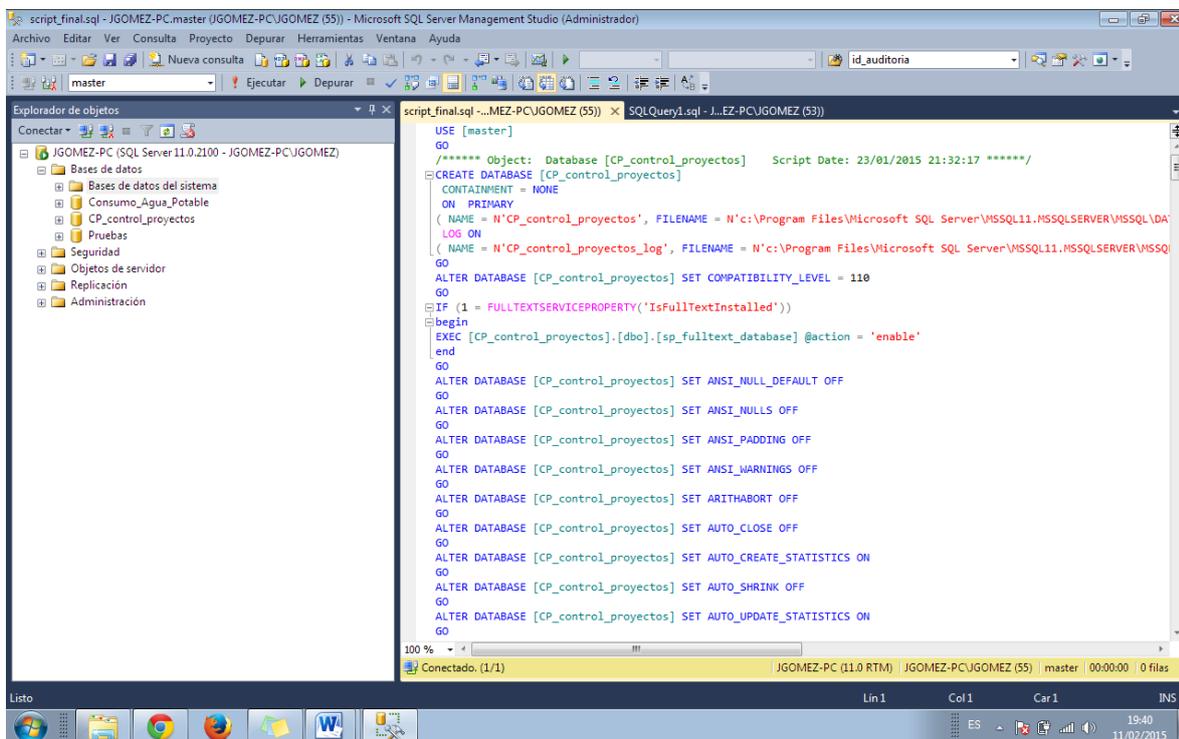
1. En primer lugar descargar he instalar SqlServer 2012 Express que es el manejador de base de datos utilizado en este proyecto, el link de descarga es el siguiente:

<http://www.microsoft.com/es-es/download/details.aspx?id=29062>

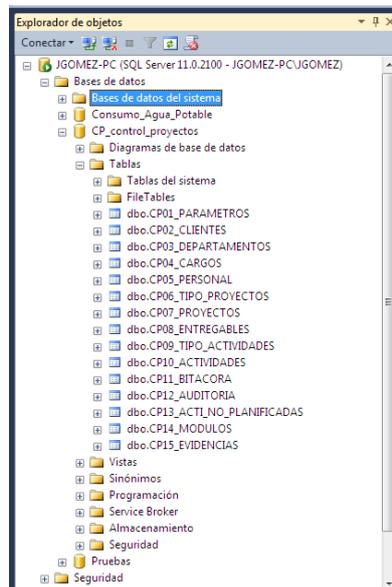
2. Descargar he instalar el IDE de desarrollo Visual Studio 2003, herramienta de desarrollo y compilación del proyecto, el lenguaje de programación utilizado es C#, se lo puede descargar en el siguiente link:

<http://www.visualstudio.com/es-es/downloads/download-visual-studio-vs.aspx>

3. Una vez instaladas las dos herramientas abrir SqlServer 2012 y restaurar el script de la base de datos (CP\_control\_proyectos), adjunto en los anexos del proyecto.



- Una vez ejecutado el script puede visualizar en el explorador de objetos la creación de la base de datos.



- El siguiente paso es crear el usuario administrador del sistema ejecutando el siguiente script.

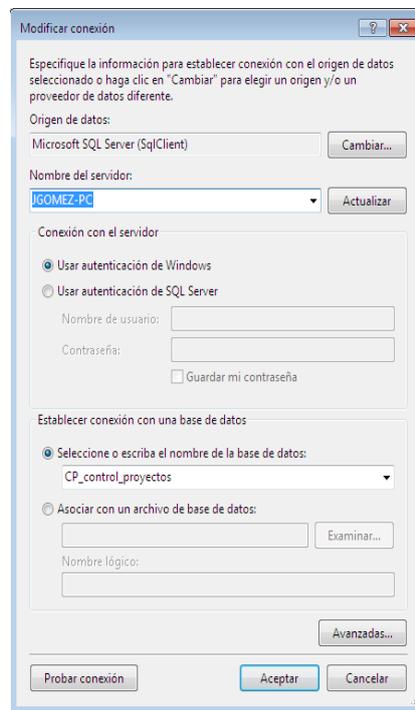
```
insert into CP03_DEPARTAMENTOS values (1,'ADMINISTRACION')
```

```
insert into CP04_CARGOS values (1,'ADMIN')
```

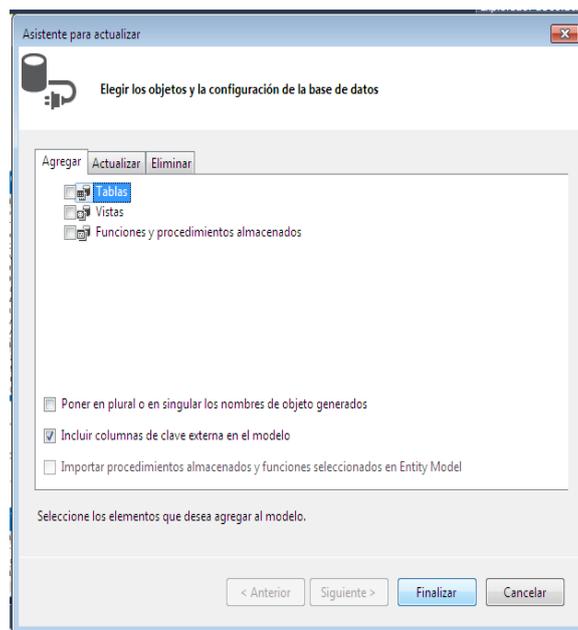
```
insert          into          CP05_PERSONAL          values
(000000000001,000000000001,1,1,'ADMINISTRADOR',' ',' ',' ',' ',' '
,'activo','Admin','E8','CP02,CP03,CP04,CP05,CP06,CP09,CP07,CP11,CP14,C
P13,CP15','Administrador')
```



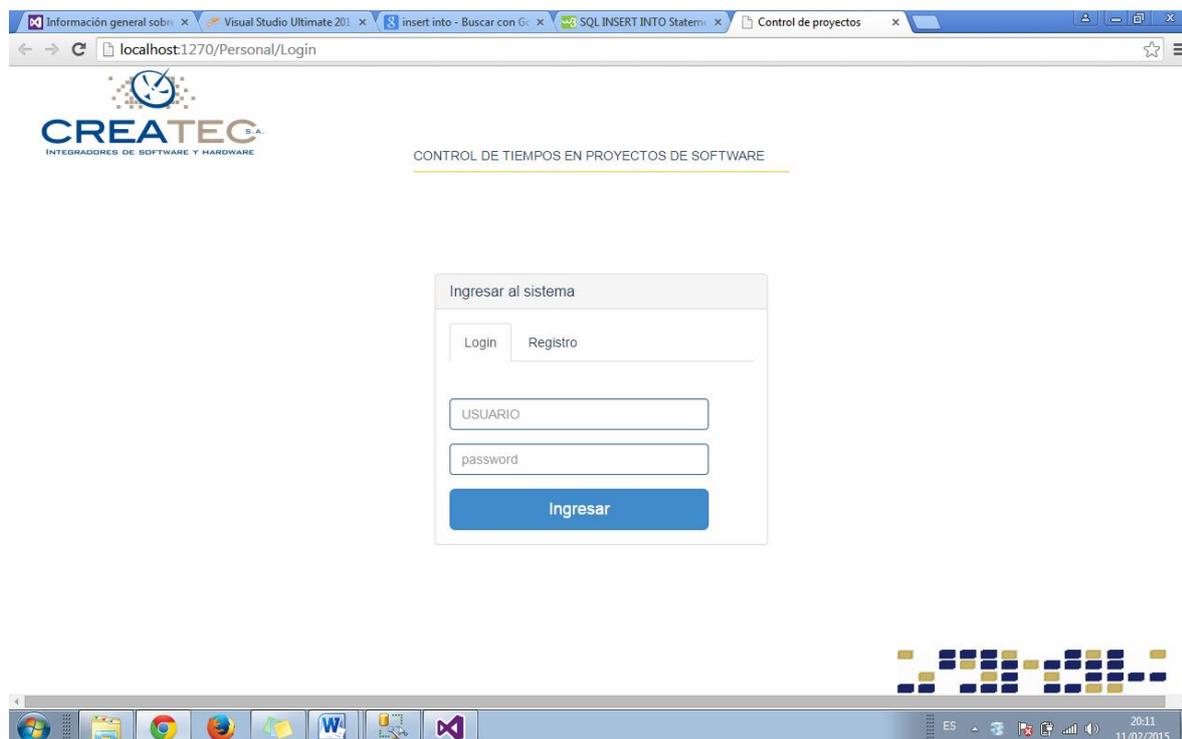
## 8. Ingresar la configuración para la conexión con la base de datos



## 9. Agregar el modelo de la base de datos.



10. Ejecutar el proyecto para comprobar su funcionamiento.



11. Ingresar los datos que ingresamos al ejecutar el script en la BDD; usuario: Admin, password: jj.

