



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD INFORMÁTICA Y ELECTRÓNICA

ESCUELA INGENIERÍA EN SISTEMAS

**“ANÁLISIS DE LA PLATAFORMA WINDOWS AZURE PARA TRABAJAR
EN ENTORNOS CLOUD Y SU APLICACIÓN EN LA ESPOCH”.**

TESIS DE GRADO

Previa a la obtención del título de:

INGENIEROS EN SISTEMAS INFORMÁTICOS

Presentado por:

RENÉ OMAR VILLA LÓPEZ

LILIANA GRIMANEZA GUZMÁN ACÁN

RIOBAMBA – ECUADOR

2011

AGRADECIMIENTO

Agradecemos a Dios quien nos dio la vida.

A nuestros padres y hermanos por todo su apoyo incondicional, sus bendiciones y sus regaños a lo largo de nuestras vidas estudiantiles, que han hecho posible alcanzar una meta más.

A la Escuela Superior Politécnica de Chimborazo y a nuestra querida Escuela de Ingeniería en Sistemas, a todos los maestros que nos han brindado sus conocimientos para llegar a culminar nuestra carrera.

Al Ing. Wladimir Castro director de este trabajo de tesis, quien desde el inicio ha estado oportuno y presto para brindarnos su colaboración.

Al Ing. Jaime Zapata, por sus aportes y sugerencias, que han sido de vital importancia para la culminación del trabajo investigativo.

DEDICATORIA

La presente tesis se la dedico a mi familia que gracias a sus consejos y palabras de aliento crecí como persona. A mis padres y hermanas por su apoyo, confianza y amor. Gracias por ayudarme a cumplir mis objetivos tanto como persona y estudiante. A mi madre por brindarme los recursos necesarios y por hacer de mí una mejor persona a través de sus consejos, enseñanzas y amor. A mi padre que mientras estuvo a mi lado siempre fue un gran amigo. A mis hermanas por estar siempre presente, cuidándome brindándome atención y aliento.

René

El presente trabajo va dedicado a Dios, a mi familia principalmente a mis padres quienes lucharon para que el gran sueño de ser profesional se cumpla, a mis maestros quienes con su guía, ayuda y confianza colaboraron de forma directa en el cumplimiento de esta meta, y a mis amigos quienes fueron un pilar muy importante en esta etapa de mi vida.

Liliana

FIRMAS DE RESPONSABLES Y NOTA

NOMBRES

FIRMAS

FECHA

Ing. Iván Menes

DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

Ing. Raúl Rosero

DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS

Ing. Wladimir Castro

DIRECTOR DE TESIS

Ing. Jaime Zapata

MIEMBRO DE TESIS

Tlgo. Carlos Rodríguez

Dir. Dpto. CENTRO DOCUMENTACIÓN

NOTA DE LA TESIS -----

Nosotros, “René Omar Villa López y Liliana Grimaneza Guzmán Acán, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

René Omar Villa López.

Liliana Grimaneza Guzmán Acán

PORTADA

AGRADECIMIENTO

DEDICATORIA

ÍNDICE GENERAL

CAPÍTULO I	21
1. Marco Referencial.....	21
1.1. Antecedentes.....	21
1.2. Justificación.....	23
1.2.1. Justificación Teórica.....	23
1.2.2. Justificación Metodológica.....	23
1.2.3. Justificación Aplicativa.....	24
1.3. Objetivos.....	24
1.3.1. Objetivo General.....	24
1.3.2. Objetivos Específicos.....	25
1.4. Hipótesis.....	25
CAPÍTULO II	26
2. Marco Teórico	26
2.1. Introducción.....	26
2.1.1. Nube o Cloud Computing.....	26
2.1.1.1. Software como servicio (SaaS).....	27
2.1.1.2. Infraestructura como Servicio (IaaS).....	28
2.1.1.3. Plataforma como Servicio (PaaS).....	28
2.1.2. Aplicaciones que pueden crearse en Windows Azure.....	29
2.1.3. Windows Azure Platform.....	30
2.1.4. Windows Azure Platform AppFabric.....	34
2.1.5. SQL Azure.....	38
2.1.6. Requisitos para desarrollar en Azure.....	39
2.1.6.1. Requisitos.....	39
2.1.6.1.1. Sistemas Operativos Compatibles.....	40
2.1.6.1.2. Software necesario para SDK de Windows Azure.....	40
2.1.6.1.3. Software necesario para Visual Studio.....	40
2.1.6.1.4. Software recomendado.....	41

2.1.6.2.	Instrucciones.	41
2.2.	Windows Azure.	42
2.2.1.	Entorno de ejecución.	42
2.2.2.	Arquitectura de una aplicación.	44
2.2.3.	Creación de un nuevo proyecto.	46
2.2.4.	El development Fabric.....	47
2.2.5.	Partial trust y full trust.	49
2.2.5.1.	Políticas de acceso a código.	50
2.2.5.1.1.	Código de seguridad de acceso.....	50
2.2.5.1.2.	Elementos de seguridad de acceso a código.....	50
2.2.5.1.3.	Evidencia.	50
2.2.5.2.	El nivel de confianza Partial Trust.	51
2.2.5.3.	El nivel de confianza Full Trust.....	52
2.2.6.	Diagnóstico y trazas.....	52
2.2.7.	Configuración en Windows Azure.....	55
2.2.7.1.	Archivo de definición de servicio.	55
2.2.7.2.	Archivo de configuración de servicio.	56
2.2.8.	Despliegue y versionado de aplicaciones.....	57
2.3.	Almacenamiento en Windows Azure.....	58
2.3.1.	Windows Azure Table.....	58
2.3.1.1.	Entidades y tablas.	59
2.3.1.2.	Contexto de acceso a datos.	61
2.3.1.3.	Orígenes de datos.	62
2.3.1.4.	La cadena de conexión.....	62
2.3.2.	Windows Azure Blob.	63
2.3.2.1.	Trabajo con BLOBS	64
2.3.3.	Windows Azure Queue.....	66
2.3.3.1.	Utilización de Colas.	67
2.3.4.	Del development Storage al almacenamiento en la Nube	68
2.3.4.1.	Claves para proteger el almacenamiento.	69
2.3.4.2.	Pasos para cambiar la clave sin parar el servicio.	70
CAPÍTULO III		71
3.	Análisis.....	71
3.1.	Costos del despliegue de una aplicación web en los servidores convencionales.....	71

3.1.1.	Generalidades	71
3.1.2.	Consideraciones Generales	72
3.1.3.	Ubicación de los Servidores	72
3.1.4.	Consideraciones con respecto Departamento de Sistemas y Telemática.	72
3.1.4.1.	Consideraciones con respecto a la Escuela de Ingeniería en Sistemas.....	73
3.1.4.2.	Costos necesarios para desplegar la aplicación en Servidores Locales.	74
3.1.4.2.1.	Costos para comprar nuevo servidor.	74
3.1.5.	Proyección Económica.	76
3.1.5.1.	Resumen y gráfico de la Proyección.....	77
3.2.	Costos de despliegue de una aplicación web en la plataforma de Windows Azure... 78	
3.2.1.	Generalidades	78
3.2.2.	Ofertas de Consumo.....	78
3.2.3.	Ofertas de Suscripción	79
3.2.3.1.	Resumen de los costos.	80
3.2.3.2.	Costos en Windows Azure.....	81
3.2.3.2.1.	Características de la aplicación de la Asociación de Escuela.	81
3.2.3.3.	Cálculo del Costo.....	81
3.2.3.3.1.	Detalle del cálculo mensual.....	82
3.2.4.	Proyección Económica.	82
3.2.4.1.	Resumen y gráfico de la Proyección.....	82
3.3.	Estudio comparativo.	83
3.3.1.	Conclusiones.....	84
3.4.	Tiempo de despliegue de una aplicación web en los servidores de la ESPOCH.	85
3.4.1.	Introducción.	85
3.4.2.	Conceptos.....	86
3.4.3.	Herramientas utilizadas	86
3.4.4.	Consideraciones para el análisis.....	87
3.4.5.	Tiempo necesario en caso de servidores locales.	87
3.5.	Tiempo de despliegue de una aplicación web en la plataforma de Windows Azure .	90
3.5.1.	Introducción	90
3.5.2.	Tiempo necesario en caso de servidores de la Plataforma de Windows Azure	90
3.6.	Estudio comparativo.	92
3.6.1.	Conclusiones.....	93
	CAPÍTULO IV	94

4.	Desarrollo de la aplicación usando Windows Azure	94
4.1.	Instalación y configuración de Visual Studio 2010 como IDE.....	94
4.1.1.	Introducción	94
4.1.2.	Proceso de Instalación	95
4.2.	Instalación y configuración de SQL Server 2008 R2	98
4.2.1.	Introducción	98
4.2.2.	Proceso de Instalación	99
4.3.	Instalación y configuración de las herramientas propias de Windows Azure	110
4.3.1.	Introducción	110
4.3.2.	Proceso de instalación.....	111
4.4.	Implementación del sistema	114
4.4.1.	Introducción	114
4.4.2.	Recopilar la información de la Empresa o Institución.....	115
4.4.2.1.	Breve Reseña Histórica.....	115
4.4.2.2.	Orgánico Funcional.....	116
4.4.2.3.	Autoridades	116
4.4.2.4.	Objetivos Generales.	117
4.4.2.5.	Objetivos Específicos.....	117
4.4.2.6.	Investigar los servicios web que brinda la Asociación	118
4.4.2.7.	Servicios que necesita la Asociación de Escuela	118
4.4.3.	Definición del problema	118
4.4.4.	Propuesta de solución	119
4.4.5.	Desarrollo de la aplicación.	120
4.5.	Desplegar la aplicación en la Nube de Windows Azure.	125
4.5.1.	Creación de una cuenta en la Plataforma de Windows Azure.....	125
4.5.1.1.	Introducción.	125
4.5.1.2.	Proceso de obtención de la cuenta.....	125
4.5.2.	Administración de la cuenta en la Plataforma Windows Azure.....	129
4.5.3.	Creación de la cuenta de Almacenamiento (Storages Accounts.)	130
4.5.4.	Creación de un nuevo Servidor de Base de Datos de SQL Azure.....	132
4.5.5.	Creación de un nuevo Servidor de Hosted.....	146
4.6.	Pruebas del Sistema	150
	CAPÍTULO V	151
5.	Informe Final	151

5.1.	Hipótesis de Investigación.....	151
5.2.	Tipo de hipótesis.	151
5.3.	Determinación de las variables.	152
5.4.	Operacionalización conceptual.	152
5.5.	Operacionalización metodológica.....	152
5.6.	Comprobación de la hipótesis de la investigación	153
5.7.	Comprobación en cuanto a Costos de Windows Azure vs Servidores Locales	153
5.8.	Comprobación del Tiempo de Windows Azure vs Servidores Locales.....	154
5.9.	Comprobación de la Hipótesis	154
	CONCLUSIONES	155
	RECOMENDACIONES	156
	RESUMEN	157
	SUMMARY	158
	BIBLIOGRAFÍA.....	159

ÍNDICE DE ABREVIATURAS

SaaS: Software como Servicio

API: Application Programming Interface

SLA: Acuerdo de nivel de servicio

PaaS: Plataforma como servicio

ISV: Proveedor de software independiente

SDK: Software Development Kit

IP: Internet Protocol

SOAP: Simple Object Access Protocol

REST: Representational State Transfer

XML: Extensible Markup Language

PHP: PHP Hypertext Pre-processor

IIS: Internet Information Services

SP1: Service Pack 1

HTTP: Hypertext Transfer Protocol

IT: Tecnología de la Información

URI: Uniform Resource Identifier

OLEDB: Object Linking and Embedding for Databases. Enlace e incrustación de objetos para bases de datos"

ADO.NET: ActiveX Data Object for .NET. Acceso a datos con .NET

ODBC: Open DataBaseConnectivity

IDE: Entorno de Desarrollo Integrado

URL: UniformResourceLocator.

CGI: Common Gateway Interface

HAL:Hardware abstraction layer

RAM: Random Access Memory

HTTPS: Hypertext Transfer Protocol Secure.

CAS: Políticas de código de acceso.

TCP/IP: Transmission Control Protocol/InternetProtocol.

ACL: Lista de control de acceso.

LINQ: Language Integrated Query.

BLOB:Binary Large Objects.

DESITEL: Departamento de Sistemas y Telemática.

CSS: Cascading Style Sheets.

ÍNDICE DE TABLAS

Tabla II.I: Características de los sistemas operativos tradicional y de Windows Azure.....	43
Tabla III.II: Costos en la compra de un nuevo servidor.....	74
Tabla III.III: Consumo aproximado de sus electrodomésticos.....	75
Tabla III.IV: Cálculo con una regla de tres compuesta.....	75
Tabla III.V: Cálculos del costo de energía eléctrica.....	75
Tabla III.VI: Cálculos del costo al segundo año.....	76
Tabla III.VII: Cálculos del costo al tercer año.....	76
Tabla III.VIII: Cálculos del costo al cuarto año.....	76
Tabla III.IX: Cálculos del costo al quinto año.....	77
Tabla III.X: Cálculos del costo al sexto año.....	77
Tabla III.XI: Costos Proyectados.....	77
Tabla III.XII: Costos según el tamaño de la Instancia de Cómputo.....	80
Tabla III.XIII: Costos del SQL Azure.....	80
Tabla III.XIV: Costos del Ancho de banda y Storage.....	80
Tabla III.XV: Costos con Windows Azure.....	81
Tabla III.XVI: Costos por mes.....	82
Tabla III.XVII: Costos Proyectados.....	82
Tabla III.XVIII: Análisis Comparativo de cada año.....	83
Tabla III.XIX: Análisis Comparativo de la proyección.....	84
Tabla III.XX: Tiempo de despliegue en servidores locales.....	89
Tabla III.XXI: Tiempo de despliegue en servidores de Azure.....	92
Tabla III.XXII: Análisis Comparativo.....	92
Tabla IV.XXIII: Descripción de los formularios principales.....	120
Tabla IV.XXIV: Descripción de ingresos al sistema de los usuarios.....	121
Tabla IV.XXV: Detalle de la Galería Fotográfica.....	122
Tabla IV.XXVI: Detalle de la opción modelo de oficios.....	122
Tabla IV.XXVII: Detalle de la opción Misión, visión, historia y logros.....	123
Tabla IV.XXVIII: Detalle de la opción Déjenos tus comentarios.....	123
Tabla IV.XXIX: Detalle de la opción Eventos.....	124
Tabla IV.XXX: Detalle de la opción Noticias.....	125
Tabla V.XXXI: Operacionalización Conceptual.....	152
Tabla V.XXXII: Operacionalización Metodológica.....	153

ÍNDICE DE FIGURAS

Figura II.1: Las aplicaciones de Windows Azure se ejecutan en centros de datos de Microsoft y pueden accederse a través de Internet.	27
Figura II.2: Windows Azure cuenta con 3 partes principales: el servicio de informática, el servicio de almacenamiento y la estructura.	30
Figura II.3: Servicios de Windows Azure Platform.	32
Figura II.4: Infraestructura de Windows Azure	34
Figura II.5:Funcionamiento del bus de servicios de la plataforma Azure.	35
Figura II.6: La plataforma AppFabric de Windows Azure.....	36
Figura II.7: Arquitectura de una aplicación Windows Azure.....	44
Figura II.8: Creación de un nuevo proyecto.	46
Figura II.9: Asignación de roles al nuevo proyecto.	47
Figura II.10: Ícono de confirmación de ejecución del Development Fabric.....	48
Figura II.11: Alojamiento de servicios en el Development Fabric.....	48
Figura II.12: Código para configurar un rol con Full Trust.....	52
Figura II.13: Código del archivo .config.	53
Figura II.14: Trazas ejecutándose.....	53
Figura II.15: Trazas ejecutándose en el Development Fabric.	54
Figura II.16: Pasos para cambiar el trazo en el Development Fabric.....	54
Figura II.17: Configurar la cadena de conexión para el almacenamiento de Azure.	55
Figura II.18: Archivo de definición de servicio	56
Figura II.19: Archivo de configuración de servicio	56
Figura II.20: Parte del archivo de configuración de servicio	56
Figura II.21: Cuadro de diálogo del editor de propiedades del rol	57
Figura II.22: Entidad Visita.....	60
Figura II.23: Creación de la tabla en caso de que no exista	61
Figura II.24: Clase de contexto de tabla.....	61
Figura II.25: Clase de Origenes de datos.....	62
Figura II.26: Cadena de conexión del archivo de configuración	63
Figura II.27: Cadena de conexión obtenido desde el código	63
Figura II.28: Cadena de conexión con el asistente.....	63
Figura II.29: Cadena de conexión.....	64
Figura II.30: Conexión con windows azure.....	65
Figura II.31: Creación del contenedor.....	65

Figura II.32: Establecimiento de las propiedades para el blob	65
Figura II.33: Propiedades y características del blob.....	66
Figura II.34: Obtener la referencia al blob y eliminación	66
Figura II.35: Utilización de colas	67
Figura II.36: Añadir mensajes a la cola.....	67
Figura II.37: Consumir el mensaje de la cola.....	68
Figura II.38: Cadena de conexión simple	69
Figura II.39: Cadena de conexión con la cuenta de azure.....	69
Figura II.40: Propiedades de la cadena de conexión con el asistente.....	69
Figura II.41: Proyección de Costos en servidores convencionales.....	78
Figura II.42: Proyección en la Plataforma Windows Azure	83
Figura III.43: Comparación de Costos.....	84
Figura III.44: Comparación de Costos.....	84
Figura III.45: Conclusiones en costos	85
Figura III.46: Serena OpenProj	86
Figura III.47: Diagrama Gantt (Parte I Servidor local)	87
Figura III.48: Diagramas Gantt (Parte II Servidor Local)	88
Figura III.49: Diagrama de Red Servidor local	88
Figura III.50: Informe General del análisis Servidor Local.....	89
Figura III.51: Diagrama Gantt (Parte I Azure).....	90
Figura II.52: Diagrama Gantt (Parte II Azure).....	90
Figura III.53: Diagrama de Red Azure.....	91
Figura III.54: Informe General del análisis Azure	91
Figura III.55: Días laborables	92
Figura IV.56: Inicio de la instalacion de Visual Studio 2010.....	95
Figura IV.57: Aceptar los terminos de la licencia para continuar	96
Figura IV.58: Seleccionar la dirección donde se va a instalar	96
Figura IV.59: Instalación de cada uno de los componentes seleccionados	97
Figura IV.60: Instalación terminada exitosamente	97
Figura IV.61: Inicio de Visual Studio 2010.....	98
Figura IV.62: Icono para instalar SQL Server 2008 R2	99
Figura IV.63: Inicio del proceso de instalación.....	99
Figura IV.64: Seleccionar la opción nueva instalacion	100
Figura IV.65: Verificación de las reglas de soporte	100

Figura IV.66: Ingresar la clave del producto.....	101
Figura IV.67: Aceptar los terminos de la licencia	102
Figura IV.68: Aceptar los soportes de las reglas	102
Figura IV.69: Advertencia de Firewall	103
Figura IV.70: Instalación del Rol	103
Figura IV.71: Selección de Servicios y Características	104
Figura IV.72: Verificar las reglas de instalacion.....	105
Figura IV.73: Llenar los campos requeridos	105
Figura IV.74:Verifica que las instancias no se repitan.....	106
Figura IV.75: Configuración del servidor	107
Figura IV.76: Configuracion de la Base de Datos.....	107
Figura IV.77: Confirmación de envío de errores	108
Figura IV.78: Verificación de las reglas.....	108
Figura IV.79: Lectura de la Instalación	109
Figura IV.80: Progreso de Instalación.....	109
Figura IV.81: Instalación Finalizada	110
Figura IV.82: Icono del VSCloudService.....	111
Figura IV.83: Productos a ser instalados	112
Figura IV.84: Aceptar los términos de la licencia	112
Figura IV.85: Requerimientos del Sistema	113
Figura IV.86: Progreso de la instalación del VSCloudService	113
Figura IV.87: Fin de la instalación del VSCloudService	114
Figura IV.88: Orgánico Funcional.	116
Figura IV.89: Página principal de Emprendedores con Tecnología.....	126
Figura IV.90: Opciones de la página de Emprendedores con Tecnología	126
Figura IV.91: Pais y clave de promoción.....	127
Figura IV.92: Advertencia para que el usuario que no se ha logueado	127
Figura IV.93: Logueo e ingreso al Sistema de Windows Azure	127
Figura IV.94: Datos completos para crear la cuenta	128
Figura IV.95: Acuerdo de Servicio del Pase de 30 días para Windows Azure	128
Figura IV.96: Confirmación y registro en la lista de espera.....	129
Figura IV.97: Portal de Administración de Windows Azure	129
Figura IV.98: Portal de Administración de Windows Azure	130
Figura IV.99: Nombre y región para la creación de la cuenta de Storage.....	131

Figura IV.100: Cuenta creada y ventana de propiedades	131
Figura IV.101: Claves primaria y secundaria del Storage Account.....	132
Figura IV.102: Nueva base de datos.....	132
Figura IV.103: Nombre del servidor de base de datos.....	133
Figura IV.104: Región del Servidor de base de datos.....	133
Figura IV.105: Usuario y clave del Servidor de base de datos	134
Figura IV.106: Creación de la regla de firewall del Servidor de base de datos	134
Figura IV.107: Creación de la nueva base de datos	135
Figura IV.108: Datos necesarios de la nueva base de datos	135
Figura IV.109: Icono de administración de base de datos	136
Figura IV.110: Datos necesarios para ingresar a la administración de base de datos	136
Figura IV.111: Página inicial de la Administración de base de datos	137
Figura IV.112: Página inicial de la Administración de base de datos	137
Figura IV.113: Microsoft SQL Server Management Studio.	138
Figura IV.114: Generar scripts para objetos de base de datos	138
Figura IV.115: Seleccionar los objetos de base de datos en el script.....	139
Figura IV.116: Opciones avanzadas del script.....	139
Figura IV.117: Habilitando el script para el tipo de motor de base de datos Azure.	140
Figura IV.118: Elección de la ubicación del archivo del script.....	140
Figura IV.119: Estado de la creación de scripts.....	141
Figura IV.120: Abriendo el archivo *.SQL.....	141
Figura IV.121: Archivo *.SQL	142
Figura IV.122: Cadena de conexión.....	142
Figura IV.123: Añadiendo la conexión al dataset en .NET 2010	143
Figura IV.124: Test de conexión satisfactoria	143
Figura IV.125: Creación de la nueva conexión.	144
Figura IV.126: Nombre de la nueva conexión.	144
Figura IV.127: Opción para seleccionar el procedimiento almacenado.	145
Figura IV.128: Seleccionando el procedimiento almacenado.....	145
Figura IV.129: Clic en finish para terminar la configuración del DataSource.....	146
Figura IV.130: Creando el paquete para la publicación	146
Figura IV.131: Administrando la creación del nuevo hosted service	147
Figura IV.132: Datos para crear el hosted service.....	147
Figura IV.133: Precaución al momento de la creación del hosted.....	148

Figura IV.134: Subiendo la aplicación al Internet.....	148
Figura IV.135: Terminando la subida al Internet de la aplicación.....	149
Figura IV.136: Aplicación de la Asociación de Escuela.....	149
Figura V.137: Conclusiones en costos.....	154

INTRODUCCIÓN

Con la aparición de la tendencia del Internet toda la tecnología se ha ido desarrollando en torno a ésta, ya que desde sus inicios ha sido de gran ayuda para estudiantes y profesionales en todas las áreas de la información.

El uso del Internet hoy en día es muy común razón por lo cual empresas como Microsoft ofertan servicios para aprovechar al máximo esta conexión.

Cuando se habla de un Sistema se entiende que se va a necesitar de un servidor físico, programadores, mantenimiento constante, contratar personal, gastos en energía eléctrica entre otros, por lo tanto las personas que desarrollan no se concentran en la programación y al contrario tienen tantas otras preocupaciones que casi nunca una página web o sistema cumple con los requerimientos para los que fue creado.

Por tal motivo nosotros planteamos el estudio de la Plataforma de Windows Azure que ofrece Microsoft para el despliegue de aplicaciones manteniendo el código y los datos en el Internet o Nube.

La estructura de este documento se encuentra conformada por cuatro capítulos los cuales se detallan a continuación:

El capítulo I Marco Referencial, contempla en principio la descripción de las razones fundamentales para desarrollar la aplicación “Asociación Sistemas” y que objetivos nos permitirá alcanzar.

En segunda instancia el capítulo II Marco teórico, da una referencia teórica sobre las herramientas propias de Windows Azure, sus componentes, funcionamiento, además las herramientas de desarrollo, los métodos de utilización y creación, este capítulo también presenta información en general de lo que es Windows 7.

El capítulo III contiene el análisis de costo y tiempo del despliegue de la aplicación web en los servidores institucionales vs. La plataforma de Windows Azure finalizando con la comprobación de la hipótesis de Investigación

El capítulo IV contiene en detalle la creación de la aplicación y el despliegue en la Plataforma Windows Azure con todos los pasos necesarios para realizar dicho trabajo y por último en el capítulo V se presenta un informe final de la presente Tesis de Investigación.

CAPÍTULO I

1. Marco Referencial.

1.1. Antecedentes.

Entre las tendencias que se encuentran a nivel mundial, cada vez exigen mejores niveles de servicios y uso de mejores prácticas de escalabilidad, confianza, seguridad, disponibilidad. El actual mundo de negocios y prestaciones de servicios está relacionado con la tecnología y la velocidad de cambio de la misma, la cual muchas veces causa serios problemas de productividad en los usuarios de las ya tan conocidas tecnologías de información.

Uno de los mayores problemas a los que se enfrentan muchos proyectos de software en Internet es su capacidad para escalar. Asegurar que un sistema puede asumir cargas crecientes o incluso manejar picos de carga con naturalidad.

Estas necesidades de escalabilidad pueden aparecer paulatinamente, de forma que se tiene tiempo para digerir esta nueva situación; o es posible que de manera eventual

aparezcan cargas enormes, en respuesta a algo que ocurre o incluso a menudo sin previo aviso. Provisionar de manera ágil el hardware necesario para hacer frente a este tipo de situaciones no es sencillo.

Un sistema soporta picos temporales de carga durante determinada época o ante eventos concretos, sin embargo debe estar dimensionado para esos picos de demanda, lo que tiene como consecuencia que durante el tiempo restante del año la carísima infraestructura esta ociosa y desaprovechada.

En definitiva, son innumerables las situaciones en las que es vital contar con capacidades de proceso, almacenamiento y ancho de banda que se puedan utilizar según las necesidades de cada momento y que sean a efectos prácticos, ilimitados.

Desde el punto de vista de los desarrolladores, el mantenimiento de un sistema es un problema latente que se debe tener en cuenta para no cometer errores en la puesta en marcha, pero con la nueva plataforma no se debe tener mucho cuidado con el mantenimiento ya que el despliegue sería igual de fácil para un proyecto pequeño como para uno masivo.

Dentro de la infraestructura las empresas deben estar pendientes de mantener sus propios servidores y centro de datos y esto multiplicando por el número de servidores o máquinas virtuales que tengan disponibles toda esta situación impacta directamente sobre otro importante factor que son los costes de operación.

Finalmente, otra cuestión que preocupa a los administradores de sistemas informáticos es la capacidad para recuperarse ante desastres.

Es así que las plataformas de servicios Cloud permiten aumentar la productividad, mejorando el rendimiento, tiempo de respuesta y sobre todo escalabilidad, despreocuparse de todas las complejidades anteriormente expuestas, centrándose en las necesidades funcionales de las aplicaciones y no en los detalles necesarios para hacerlas funcionar.

En definitiva, escalabilidad, alta disponibilidad y reducción de costes de operación son las principales ventajas.

La plataforma que se va a analizar es Windows Azure la misma que utiliza herramientas propias de Microsoft para su implementación dentro de las cuales están .NETServices,

SQLAzure (Base de datos relacional en la Nube), por lo tanto lo que se va a realizar en el presente trabajo de investigación es su análisis a fondo y levantar una aplicación web informativa para la Asociación de Escuela de Ingeniería en Sistemas para la demostración.

1.2. Justificación.

1.2.1. Justificación Teórica.

Actualmente la gran mayoría de empresas tanto públicas como privadas vienen realizando un estudio de las plataformas, sistemas operativos y servidores locales pero no las que son bajo demanda y que brinden servicios Cloud, razón por la cual se ha visto la necesidad de realizar dicho estudio en esta nueva plataforma Windows Azure para extraer los aspectos más destacables para desarrollar una aplicación en un entorno Cloud.

Hoy en día la escalabilidad, seguridad y mantenimiento a bajo costo es lo más importante para una Institución, es por eso que al no existir el desarrollo Cloud se ha visto la necesidad de realizarlo.

1.2.2. Justificación Metodológica.

Mediante el análisis de toda la plataforma de Windows Azure se extraerán los aspectos más importantes relacionados a la escalabilidad, seguridad, disponibilidad y mantenimiento en el desarrollo de las aplicaciones, las mismas que posteriormente serán utilizadas en la implementación de una aplicación en un entorno Cloud.

Realizar un sistema con servidores locales es una buena forma de implementar una aplicación web, pero con esta plataforma se pretende no tener mucho cuidado tanto en seguridad, escalabilidad y mantenimiento. Mediante el desarrollo con .NET de Microsoft utilizando el Framework 4.0 se desea poner en práctica toda la metodología y

la forma de desarrollo ya conocida con la diferencia que todo el desarrollo es enfocado al entorno Cloud.

Por lo tanto este análisis dará como resultado las ventajas y desventajas de la plataforma Windows Azure.

1.2.3. Justificación Aplicativa.

Actualmente en la Escuela Superior Politécnica de Chimborazo no se ha realizado ningún tipo de sistema basado en entorno Cloud ya sea para producción o de prueba, razón por la cual se ha decidido implementar una aplicación para dar a conocer los beneficios que posee.

La aplicación utilizará todas las herramientas indispensables para la implementación. Entre estas se tiene: .NET como plataforma de desarrollo, SQLAzure que es la base de datos relacional en la Nube, AppFabric que es el encargado de interactuar con los servicios, y también se demostrará el uso de las diferentes formas de almacenamiento de los datos.

Por lo tanto se pretende realizar una aplicación web para la Asociación de Escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo la misma que contendrá información referente a las necesidades de los estudiantes como son: Historia de la Escuela, Tesis realizadas, Noticias y eventos importantes entre otras actividades.

1.3. Objetivos.

1.3.1. Objetivo General.

Analizar la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH”.

1.3.2. Objetivos Específicos.

- ✓ Investigar la plataforma Windows Azure como base.
- ✓ Analizar la plataforma para el desarrollo de las aplicaciones en entornos Cloud dando un enfoque de costos y tiempo de despliegue.
- ✓ Implementar una aplicación web informativa para la Asociación de Escuela de Ingeniería en Sistemas bajo la Plataforma Windows Azure.

1.4. Hipótesis.

Mediante el análisis la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH permitirá desplegar aplicaciones de consumo masivo en un corto tiempo y a bajo costo en comparación con las aplicaciones tradicionales.

CAPÍTULO II

2. Marco Teórico

2.1. Introducción.

2.1.1. Nube o CloudComputing.

La Nube es la respuesta a las necesidades de los departamentos de tecnología de la información de las empresas en la búsqueda de mayor escalabilidad, alta disponibilidad y reducción de los costes operativos.

La idea fundamental es simple: dejar en manos de empresas especializadas la gestión de la infraestructura en la que corren las aplicaciones. Los desarrolladores controlan las aplicaciones y los datos, pero la infraestructura sobre la que corren éstas no es preocupación de los mismos. Es una segunda empresa la que se encarga de proporcionar esta infraestructura y no solo a una, sino a cientos de miles.

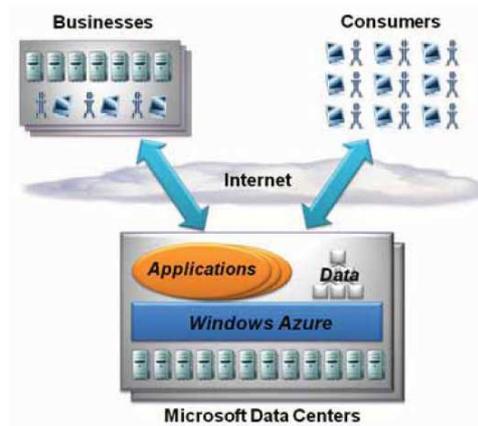


Figura II.1: Las aplicaciones de Windows Azure se ejecutan en centros de datos de Microsoft y pueden accederse a través de Internet.

La ventaja es clara, se trata de economía de escala: considerando los costos por unidad de tiempo de ejecución, es muchísimo más económico establecer para miles de aplicaciones una infraestructura altamente escalable, con una elevada disponibilidad y con ancho de banda casi ilimitado, que hacerlo para una decena de ellas.

Las características fundamentales que debe proporcionar un servicio para poder ser considerado “Cloud Computing”, en el sentido que se ha analizado antes, son:

- Auto-servicio por parte de los clientes.
- Capacidad de medir el servicio.
- Pago por uso.
- Rápida elasticidad.
- Distribución geográfica.

Dentro de los servicios en la Nube se puede distinguir tres categorías fundamentales:

2.1.1.1. Software como servicio (SaaS).

Se trata del cambio a versiones en la Nube de aplicaciones tradicionalmente instaladas en la infraestructura. Por ejemplo Exchange Online es una versión SaaS del servidor de correo Exchange Server de Microsoft, que se ofrece hospedada en los Data Centers de Microsoft o sus partners. Por lo tanto una empresa puede contratarlo y empezar a usarlo

de inmediato sin tener que adquirir licencias ni hardware específico, sin tener que mantenerlo y pagando solo lo que utiliza. Actualmente Microsoft ofrece una amplia selección de aplicaciones a modo SaaS a través de los Online Services¹.

2.1.1.2. Infraestructura como Servicio (IaaS).

Es el hospedaje de aplicaciones existentes dentro de máquinas virtuales en la Nube. Siguen siendo aplicaciones normales, ejecutadas en un sistema operativo común, sólo que se ejecutan dentro de máquinas virtuales en la Nube. Permite despreocuparse de la gestión de los servidores físicos, de modo que se evita problemas de fallos de hardware o desastres locales, y suelen ofrecer capacidad para crecer en el almacenamiento. No obstante sigue siendo necesario ocuparse de muchas otras cuestiones de infraestructura como la alta disponibilidad, el balanceo de carga, gestión del Sistema Operativo, el software base, etc. Microsoft permite gestionar esto de manera privada a empresas a través de System Center o sus productos de virtualización. Los ejemplos de carácter público más conocidos de este tipo de servicio son Amazon EC2² y RackSpace.

2.1.1.3. Plataforma como Servicio (PaaS).

Se trata de los servicios en la Nube que permiten crear aplicaciones específicamente desarrolladas para trabajar en entornos “Cloud” con las ventajas señaladas. Toda la infraestructura de base, tanto el software como el hardware, son transparentes. Se crea la aplicación y se despliega obteniendo alta disponibilidad, alta escalabilidad y nulos costes de operación. La desventaja es que cada plataforma PaaS tiene sus propios API de desarrollo por lo que es complicado mover una aplicación de una a otra Nube. Es en

¹**Online Services:** Servicios Online de Microsoft.

²**Amazon EC2:** Servicio web que proporciona capacidad informática con tamaño modificable en la Nube. Se ha diseñado con el fin de que la informática web resulte más sencilla a los desarrolladores

esta categoría donde se encaja Windows AzurePlatform. Algunos otros ejemplos se tienen App Engine³ de Google o Force.com⁴ de Salesforce.

Además de los centros de datos actuales, Microsoft está construyendo enormes *Data Centers* nuevos en todo el mundo con el propósito de proporcionar servicio adicional en el futuro a los clientes Windows AzurePlatform.

Otra cuestión importantísima Windows AzurePlatform frente a servicios de otros fabricantes es que incluye en el contrato un acuerdo de nivel de servicio (SLA). Es decir, al contrario que en otros servicios PaaS, Microsoft se compromete a garantizar elevados niveles de disponibilidad, capacidad, conectividad así como monitorización y supervisión, lo que es una garantía adicional para aplicaciones críticas empresariales.

2.1.2. Aplicaciones que pueden crearse en Windows Azure.

- Un proveedor de software independiente (ISV) podría crear una aplicación para usuarios comerciales, un método que por lo general se refiere a SaaS. Los ISV pueden utilizar Windows Azure como base para varias aplicaciones SaaS orientadas a los negocios.
- Es posible que un ISV cree una aplicación SaaS para consumidores. Windows Azure se encuentra diseñado para brindar soporte a software muy ampliable y, por lo tanto, es posible que una empresa que planifica dirigirse al gran mercado de consumidores lo elija como plataforma para una nueva aplicación.
- Las empresas pueden usar Windows Azure para crear y ejecutar aplicaciones que utilicen sus propios empleados. Mientras que esta situación pueda no necesitar la enorme escala de una aplicación para consumidores, la confiabilidad y la capacidad de administración que ofrece Windows Azure pueden convertirlo en una opción atractiva.

³ **App Engine:** Permite crear aplicaciones web en los mismos sistemas de Google, no se necesita otro servidor, solo se debe subir la aplicación y los usuarios pueden empezar a utilizarla.

⁴ **Force.com:** Plataforma líder de Cloud Computing para aplicaciones comerciales.

Sin importar lo que realice una aplicación de Windows Azure, la plataforma ofrece los mismos componentes fundamentales.

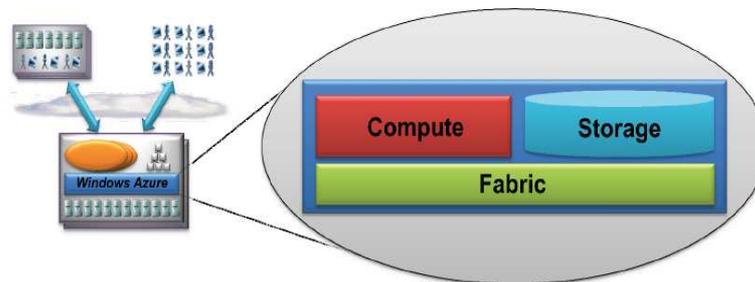


Figura II.2: Windows Azure cuenta con 3 partes principales: el servicio de informática, el servicio de almacenamiento y la estructura.

2.1.3. Windows Azure Platform.

Es un servicio PaaS que forma parte de la oferta de servicios online de Microsoft. Proporciona un entorno familiar y flexible para desarrollar aplicaciones y servicios en la Nube con todas las ventajas que se ha analizado. Con Windows AzurePlatform la empresa puede reducir el tiempo de lanzamiento de los productos y adaptarse fácilmente a medida que la demanda de éstos crezca.

Windows Azurees una plataforma interoperable, que permite desarrollar en diversos lenguajes así como la comunicación con cualquier entorno externo. Además, si bien Visual Studio es la herramienta más productiva a la hora de trabajar con la plataforma, se dispone de herramientas y SDKs⁵ para otros sistemas y entornos.

Windows AzurePlatform proporciona un conjunto de servicios base en la Nube que pude usarse conjuntamente o de manera independiente para permitir a los desarrolladores crear de una manera sencilla aplicaciones que responden de forma idónea a las necesidades de crecimiento de usuarios y clientes.

Windows AzurePlatform está compuesta por los siguientes servicios:

⁵SDK: Software Development Kit- Kit de desarrollo de software. Es un conjunto de herramientas y programas de desarrollo que permite al programador crear nuevas aplicaciones para determinada plataforma o similar.

Windows Azure.- Ofrece a los desarrolladores servicios de ejecución y almacenamiento bajo demanda. Dada la abstracción de la que provee a las aplicaciones se suele definir como el Sistema Operativo de la Nube, lo cual es una forma sencilla de entenderlo. Utilizando este servicio, los desarrolladores podrán desplegar y gestionar sus aplicaciones en los centros de datos de Microsoft. Además ofrece mecanismos simples de almacenamiento (tablas y BLOBS) y de comunicaciones basadas en colas, esta es la parte de la plataforma que toda aplicación Azure utilizará. Aunque es programable en diversos lenguajes en este presente trabajo se utilizará la plataforma .NET.

Microsoft SQLAzure.- Es una base de datos relacional en la Nube que permite tener acceso a nuestros datos relaciones desde cualquier lugar en cualquier momento. Se puede considerar como un servidor de datos SQL Server convencional pero adaptado para funcionar en la Nube, en donde la alta disponibilidad es una de las propiedades. Hay que destacar que SQLAzure es el primer gestor para la Nube realmente relacional, que puede ejecutar y entender consultas SQL.

Windows AzurePlatformAppFabric.- Es la parte de la parte de Windows AzurePlatform que proporciona autorización, autenticación y mensajería de manera que permite la comunicación segura entre las aplicaciones desplegadas en la organización y las aplicaciones desplegadas en Windows Azure.

AppFabricService Bus.- Ofrece a los desarrolladores la flexibilidad para escoger como se comunican sus aplicaciones, solucionando retos impuestos por cortafuegos, NAT, IP dinámicas, etc.

AppFabric Access Control.- Posibilita una autorización simple y segura para servicios web REST, que además puede federarse⁶ con un gran abanico de proveedores de identidades

Windows AzureMarketplace.- Es un mercado online para los desarrolladores para compartir, encontrar, comprar y vender componentes de la construcción de bloques,

⁶**Federación:** Una colección de dominios que han establecido confianza. El nivel de confianza puede variar, pero normalmente incluye la autenticación y casi siempre incluye la autorización. Una federación típica podría incluir varias organizaciones que han establecido la confianza para el acceso compartido a un conjunto de recursos.

plantillas, datos Premium y aplicaciones necesarias para construir aplicaciones de Windows AzurePlatform.

Windows AzureMarketplaceDataMarket.- Antes conocido como Codename⁷ Dallas, incluye datos, imágenes, y los servicios web en tiempo real de los principales proveedores y de los datos de fuentes públicas. Los clientes tendrán acceso a bases de datos tales como demográficos, ambientales, financieros, etc. Datamarket también incluye visualizaciones y análisis de los datos.

Windows AzureMarketplaceApplications.- La sección de aplicaciones de Windows AzureMarketplace ofrece una lista de elementos de componentes de bloques, servicios y servicios/aplicaciones. Estos componentes están diseñados para ser incorporados por otros desarrolladores en sus aplicaciones de Windows AzurePlatform. También incluye herramientas de desarrollo, herramientas administrativas, componentes, plugins y plantillas.

En el siguiente gráfico se muestra los servicios que esta plataforma proporciona y como se relacionan entre sí.

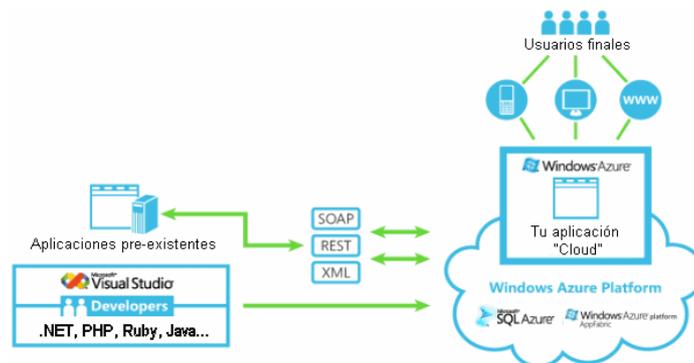


Figura II.3: Servicios de Windows Azure Platform.

Se puede ver en el gráfico como Windows AzurePlatform proporciona servicios para utilizar desde cualquier plataforma gracias al uso de estándares universalmente aceptados como SOAP, REST y XML. Está soportado el desarrollo directo sobre la plataforma con .NET pero también existe para diferentes entornos.

⁷**Codename:** Nombre que se le da a un proyecto antes de lanzarse al mercado

¿Qué es el Windows Azure?

Es el sistema operativo en la Nube de Microsoft. Proporciona un entorno gestionado para la ejecución y el despliegue de aplicaciones y servicios en la Nube. Windows Azure proporciona, como desarrolladores, un entorno de computación bajo demanda y almacenamiento alojado en los centros de datos de Microsoft para las aplicaciones en la web.

Aunque Windows Azure permite reutilizar todos los conocimientos de .NET es también, como ya se ha especificado antes, una plataforma abierta a otros lenguajes y plataformas mediante el uso de estándares y el soporte para lenguajes ajenos a la plataforma .NET como, por ejemplo, PHP, lenguajes no manejados como C/C++ nativo, así como soporte para el protocolo FastCGI⁸.

Desde el punto de vista del desarrollo con .NET, Windows Azure permite ejecutar aplicaciones ASP.NET y código .NET en la Nube, proporcionando una plataforma de ejecución basada, en el Framework⁹ de .NET 3,5 SP1 e IIS 7, complementado con un API de almacenamiento y de “tracing” propios de la plataforma Azure. Ofrece un portal que permite gestionar las aplicaciones Azure de una manera cómoda y natural.

El entorno de ejecución de Windows Azure es **full trust** lo que permite ejecutar servicios de Windows Communication Foundation¹⁰ e incluso llamar a código nativo desde las aplicaciones Azure basadas en .NET.

Además Windows Azure permite proporcionar servicios de almacenamiento no relacional y colas con acceso autenticado, alta redundancia (triple) y accesible mediante una interfaz REST¹¹ que se puede utilizar desde cualquier lenguaje que tenga la capacidad de realizar peticiones HTTP, que es tanto como decir cualquier lenguaje moderno.

⁸**FastCGI:** Es un protocolo para interconectar programas interactivos con un servidor web. Es una variación de Common Gateway Interface (CGI), su principal objetivo es reducir la carga y permitir al servidor web atender más peticiones a la vez.

⁹**Framework:** Es una estructura conceptual y tecnológica de soporte definida. Es diseñado para facilitar el desarrollo de software y dejar de tratar los detalles de bajo nivel para proveer un sistema funcional

¹⁰**WCF:** Es un marco de trabajo para la creación de aplicaciones orientadas a servicios, es posible enviar datos como mensajes asíncronos de un extremo de servicio a otro.

¹¹**REST:** RepresentationalState Transfer es un estilo de arquitectura de software centrada en recursos, los cuales son accedidos mediante el uso de URI's y acciones Http (get, post, put y delete).

A continuación se muestra la Arquitectura de la Infraestructura de Windows Azure

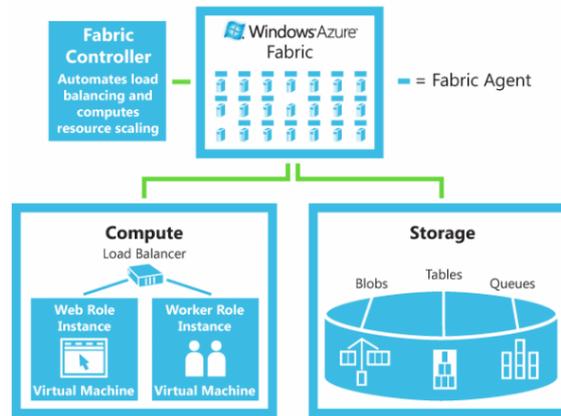


Figura II.4: Infraestructura de Windows Azure

En el gráfico anterior se ve como elemento a destacar el Windows AzureFabric, los cimientos sobre los que se levanta la plataforma Azure. El **Fabric**, como se conoce familiarmente, es el componente de la arquitectura que se encarga de proporcionar los servicios base de Windows Azure de manera transparente respecto a la infraestructura IT subyacente. Por lo tanto nunca se sabe en qué máquina concreta del centro de datos de Microsoft se está ejecutando la aplicación Azure. El Fabric se encarga de asegurar que nuestra aplicación recibe tiempo de ejecución, ancho de banda y recursos en general para su ejecución, balanceando la carga a las máquinas virtuales que considere necesario de manera transparente para nuestra aplicación.

2.1.4. Windows Azure Platform AppFabric.

Proporciona un bus de servicios empresarial y un servicio de control de acceso que permite integrar servicios y aplicaciones que se ejecutan en la Nube, en proveedores de alojamiento tradicionales y en nuestra empresa basándose en estándares de interoperabilidad.

Un bus de servicios empresarial permite orquestar la conectividad segura entre diferentes servicios y aplicaciones a través de cortafuegos y redes utilizando numerosos patrones de comunicación. Los diferentes servicios se registran en el bus de servicios de manera que pueden ser fácilmente accedidos a través de las más variadas tipologías de

red. Si la aplicación tiene que consumir e interactuar con una gran cantidad de servicios, algunos de ellos controlados por terceros, utilizar un bus de servicios permite olvidarse de detalles como la autenticación y autorización, los protocolos de comunicación, los cortafuegos y otras cuestiones técnicas, delegándolos en el bus de servicios.

El bus de servicios de la plataforma Azure facilita la labor de conectar aplicaciones que se ejecutan sobre Windows Azure o contra SQLAzure con aplicaciones que corren en la propia infraestructura y contra servidores de bases de datos convencionales. Otro escenario en el que el bus de servicios ayuda enormemente es cuando se tiene que crear aplicaciones compuestas mediante la integración de diferentes servicios ya existentes y nuevos servicios que se ejecutan en la plataforma Azure.

A continuación se puede ver el esquema de funcionamiento del bus de servicios de la plataforma Azure.

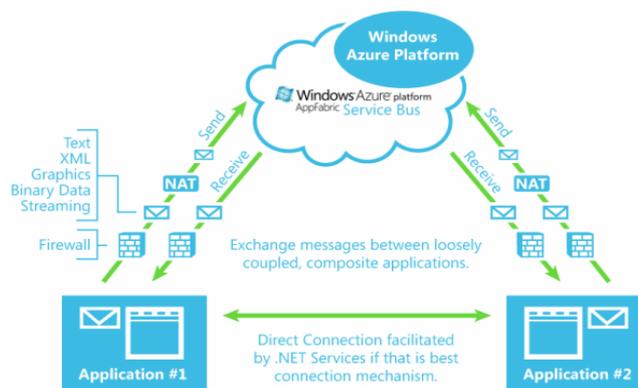


Figura II.5:Funcionamiento del bus de servicios de la plataforma Azure.

Del gráfico anterior los puntos más destacables son:

- El mecanismo preferido de comunicación entre aplicaciones en la Nube es utilizar la infraestructura que el Framework de .NET proporciona para tal fin: Windows CommunicationFoundation.
- Cuando la comunicación directa no es posible por cuestiones relacionadas con la topología de la red, o con el hecho de que no se ha controlado todos los servicios, sino que se debe integrar servicios de terceros, el bus de servicios de .NET puede ahorrar mucho trabajo proporcionando: mecanismos de comunicación amigables con los cortafuegos, comunicaciones basadas en mensajes entre servicios, direccionamiento entre redes heterogéneas (NAT) y servicios de orquestación.

No se puede decir que este componente de la plataforma Azure sea de uso común en la gran mayoría de las aplicaciones Azure, pero sí que es de gran utilidad en las ocasiones en las que orquestar servicios y comunicarlos entre sí es la principal labor que tiene que realizar la aplicación.

Ejecutar aplicaciones y almacenar datos en el Cloud son aspectos importantes de la informática Cloud. No obstante, no son los únicos aspectos. Otra opción es suministrar servicios de infraestructura basados en Cloud. Llenar este vacío es el objetivo de la plataforma AppFabric de Windows Azure.

Las funciones que hoy ofrece AppFabric abordan desafíos de infraestructura comunes para conectar aplicaciones distribuidas. A continuación se muestra estos componentes.

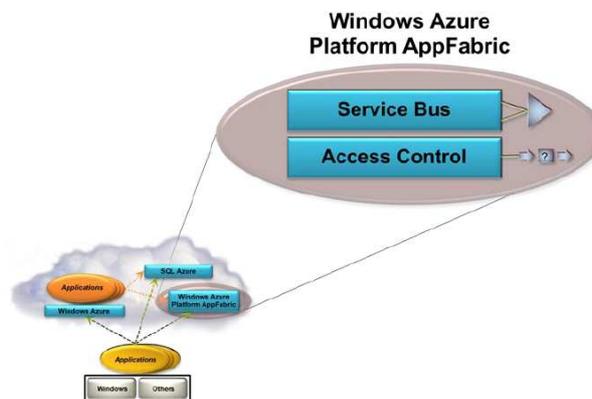


Figura II.6: La plataforma AppFabric de Windows Azure.

Los componentes de la plataforma AppFabric de Windows Azure son:

- **Service Bus:** Exhibir los servicios de una aplicación en Internet puede resultar más difícil de lo que parece. El objetivo de Service Bus es simplificar esto al permitir que una aplicación exhiba terminales que puedan ser accesibles desde otras aplicaciones, ya sean internas o del Cloud. A cada terminal expuesta se le asigna una URI¹², la cual pueden utilizar los clientes para ubicar y acceder al servicio. Service Bus también maneja los desafíos que implica encargarse de la traducción de direcciones de red e ingresar en firewalls sin abrir nuevos puertos para buscar aplicaciones expuestas.

¹²**URI:** Es una cadena corta de caracteres que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema.

- **Control de acceso:** Este servicio permite que una aplicación de cliente REST se autentique y ofrezca una aplicación de servidor con información de identidad. Entonces, el servidor puede usar esta información para decidir lo que puede realizar esta aplicación.

Al igual que con Windows Azure y SQLAzure, se ofrece un portal con acceso desde Internet para que los clientes inicien sesión en AppFabric con una ID de Windows Live. Una vez realizado, estos servicios se pueden utilizar de varias formas. A continuación algunos ejemplos:

- Una empresa desea que el software y sus socios de negocios accedan a una de sus aplicaciones. Esto podría exponer las funciones de esta aplicación a través de los servicios web REST o SOAP, luego registrar sus terminales con Service Bus. De esa forma, los socios de negocios podrán utilizar Service Bus para encontrar estas terminales y acceder a los servicios.
- Una aplicación que se ejecuta en Windows Azure puede necesitar el acceso a los datos almacenados en una base de datos interna. Para realizar esto, se requiere que la base de datos se encuentre disponible vía Internet, un problema que puede solucionarse al crear un servicio que acceda a los datos, y que luego exhiba este servicio vía Service Bus.
- Una empresa que expone servicios de aplicaciones a los socios de negocios. Si se exponen esos servicios utilizando REST, la aplicación puede acudir al Control de Acceso para autenticar y proporcionar información de identidad para cada aplicación cliente. En lugar de mantener la información de forma interna sobre cada aplicación del socio de negocios, podría almacenarse en el servicio de Control de Acceso.

Como ya se ha descrito, la plataforma AppFabric de Windows Azure ofrece servicios de infraestructura basadas en Cloud. Microsoft también está creando una tecnología análoga conocida como Windows Server AppFabric. Como su nombre lo sugiere, los servicios que ofrece se ejecutan en Windows Server (admiten aplicaciones internas) en lugar de funcionar en el Cloud. A su vez, los cuales se enfocan hoy en día en servicios WCF de hosting y en caché distribuida.

2.1.5. SQL Azure.

Es una base de datos relacional en la Nube construida sobre la tecnología de SQL Server. Proporciona servicios de bases de datos altamente escalables y con altísima disponibilidad alojados por Microsoft en la Nube. Estos servicios facilitan enormemente el despliegue de bases de datos.

Una ventaja añadida es que los desarrolladores y el personal de IT no necesitan instalar, actualizar y gestionar la infraestructura de base de datos. La alta disponibilidad, aspecto siempre complejo, es gestionado de manera permanente. La gran ventaja de utilizar SQLAzure frente a otros sistemas de almacenamiento en la Nube es que todos los conocimientos sobre base de datos relacional y el lenguaje de consulta SQL siguen siendo válidos.

No se tiene que adaptar el conocimiento a nuevos paradigmas de almacenamiento, como pasa con otros sistemas de almacenamiento en la Nube no basados en bases de datos relaciones ni SQL. Si se sabe utilizar SQL Server, todos sus conocimientos son válidos para SQLAzure.

SQLAzure, construido sobre SQL Server, permite incluso migrar los backend de datos a la Nube sin tener que tocar ni una sola línea de código de las aplicaciones en un gran número de escenarios. Es cierto que hay algunas características de SQL Server que SQLAzure no soporta, pero si soporta las más usadas:

- Tablas, tablas temporales, vistas, índices, roles, procedimientos almacenados y funciones.
- Consultas complejas y joins entre múltiples tablas.
- Sentencias Insert, Update y Delete.
- Restricciones.
- Transacciones.

Entre las características no soportadas cabe destacar:

- Transacciones distribuidas.
- El broker de mensajes de SQL Server.

- Consultas a servidores remotos.
- Acceso desde tecnologías antiguas, ya obsoletas, en concreto OLEDB.

A la hora de conectar desde las aplicaciones clientes, se puede elegir varios tipos de conexión.

- ADO.NET¹³.
- AccesoODBC¹⁴ native.
- Soporte para PHP.

SQL Azure, es parte de la plataforma .NET, pero tiene un coste adicional al coste de Windows Azure. Windows Azure proporciona servicios de almacenamiento no relacionales.

2.1.6. Requisitos para desarrollar en Azure

Aunque ya se ha comentado que es posible programar para Windows Azure en lenguajes nativos o en PHP, en el presente trabajo se va a centrar en el desarrollo para Windows Azure utilizando la plataforma .NET y C# como lenguaje de desarrollo. Utilizando Visual Studio como IDE¹⁵ y SQL Server Express.

Los pasos a dar para desarrollar en Windows Azure son:

2.1.6.1. Requisitos.

Los requisitos necesarios para una correcta instalación son los siguientes:

¹³**ADO.NET:** Es un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft .NET Framework

¹⁴**ODBC:** OpenDataBaseConnectivity es un estándar de acceso a Bases de datos el objetivo es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (*DBMS* por sus siglas en inglés) almacene los datos

¹⁵**IDE:** Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

2.1.6.1.1. Sistemas Operativos Compatibles.

- ✓ Windows 7,
- ✓ Windows 7 Enterprise,
- ✓ Windows 7 Home Premium,
- ✓ Windows 7 Home Premium 64-bit,
- ✓ Windows 7 Professional,
- ✓ Windows 7 Professional 64-bit,
- ✓ Windows 7 Ultimate,
- ✓ Windows 7 Ultimate 64-bit,
- ✓ Windows Server 2008 R2,
- ✓ Windows Server 2008 Service Pack 2,
- ✓ Windows Vista Service Pack 2.

2.1.6.1.2. Software necesario para SDK de Windows Azure.

- ✓ .NET Framework 3.5 SP1 o superior.
- ✓ IIS 7.0 (con ASP.NET, Activación HTTP de WCF, Contenido Estático, Consola de administración de IIS y de forma opcional CGI).
- ✓ Microsoft SQL Server Express 2008, Microsoft SQL Server Express 2005, Microsoft SQL Server 2005 o Microsoft SQL Server 2008.
- ✓ Revisión: Corregir metadatos WCF al hospedar detrás de un equilibrador de carga.
- ✓ Revisión: Compatibilidad con FastCGI (no es necesario para Windows 7).

2.1.6.1.3. Software necesario para Visual Studio.

- ✓ Revisión: Habilitar IntelliTrace para Windows Azure en SO de 32 bits (no es necesario para SO x64).
- ✓ Revisión: Mejorar la estabilidad de Visual Studio (no es necesario para Windows 7)

- ✓ Windows PowerShell.
- ✓ Módulo URL Rewrite para IIS 7.0.

2.1.6.1.4. Software recomendado.

- ✓ Módulo URL Rewrite para IIS 7.0.
- ✓ Windows PowerShell.

2.1.6.2. Instrucciones.

1. Instalar Microsoft Visual Studio 2010 o Instalar Microsoft Web Developer 2010 Express.
2. Instalar SQL Server 2008 Express Edition.
3. Habilitar IIS 7 con ASP.NET y Activar HTTP de WCF, Contenido Estático, Consola de administración de IIS, y de forma opcional CGI.
4. Instalar Windows Azure Tools para Microsoft Visual Studio 1.3–VsCloudService.exe (incluye el SDK). O bien, instalar simplemente el SDK de Windows Azure: WindowsAzureSDK-x86.exe o WindowsAzureSDK-x64.exe.
5. Instalar la revisión: Habilitar IntelliTrace para Windows Azure en SO de 32 bits (no es necesario para SO x64).
6. Instalar la revisión: Mejoras en la depuración nativa (no es necesario para Visual Studio 2010).
7. Instalar la revisión: Compatibilidad con FastCGI en el entramado de desarrollo (no es necesario para Windows 7 ni para Windows Server 2008 SP2 o posterior).
8. Instalar la revisión: Mejorar la estabilidad de Visual Studio (no es necesario para Windows 7).
9. Instalar la revisión: Corregir metadatos WCF al hospedar detrás de un equilibrador de carga (roles de .NET 3.5) (Windows 7 Hotfix).

Para desplegar las aplicaciones que se desarrolle en la Nube de Windows Azurese necesita:

- Crear una nueva cuenta de Windows Azure.
- Una vez que se tenga acceso al portal se podrá crear un nuevo proyecto y activar los servicios de almacenamiento si se necesita.

2.2. Windows Azure.

En un sentido estricto se debe diferenciar Windows Azure de Windows AzurePlatform. Aunque muchas veces ambas denominaciones se usen de manera casi intercambiable, se debe recordar que Windows Azure es una parte de Windows AzurePlatform, en concreto, la que proporciona la infraestructura de ejecución y almacenamiento a las aplicaciones Azure.

Windows Azure tiene dos partes diferenciadas:

- El entorno de ejecución de Windows Azure.
- Los servicios de almacenamiento y colas de Windows Azure.

2.2.1. Entorno de ejecución.

Windows Azureproporciona un entorno de ejecución, un sistema operativo, en la Nube. Este entorno de ejecución está basado en Microsoft Windows, InternetInformation Server y tecnologías de virtualización.

La infraestructura subyacente es transparente para las aplicaciones. Nunca se puede saber si la aplicación está corriendo en una determinada máquina u otra, o incluso en varias.

Características y Diferencias de los servicios del Sistema Operativo tradicional y de Windows Azure.

SISTEMA OPERATIVO TRADICIONAL	WINDOWS AZURE
<ul style="list-style-type: none"> • Un entorno de ejecución de aplicaciones que actúa como una capa de abstracción sobre el hardware subyacente. 	<ul style="list-style-type: none"> • Un entorno de ejecución de aplicaciones que abstrae de la infraestructura subyacente. Cuando se necesita una escalabilidad y una disponibilidad casi ilimitadas, se necesita una capa de abstracción que abstraiga la arquitectura distribuida en infinidad de máquinas físicas. Si una máquina en concreto de uno de los centros de datos de Microsoft necesita ser sustituida, gracias a Windows Azure es totalmente transparente para las aplicaciones.
<ul style="list-style-type: none"> • Un sistema de archivos compartido con control de acceso. 	<ul style="list-style-type: none"> • Windows Azure se encarga de abstraer los detalles de almacenamiento y así asegurar la escalabilidad, la redundancia, y la disponibilidad del sistema de archivos en la Nube.
<ul style="list-style-type: none"> • Un orquestador de recursos compartidos de computación. 	<ul style="list-style-type: none"> • El entorno de ejecución de Windows Azure debe orquestar los recursos que esas aplicaciones usan, dotándolas del tiempo de ejecución necesario, memoria y ancho de banda y repartiéndolo adecuadamente. Lógicamente para una aplicación que se ejecuta en Windows Azure debe ser totalmente transparente el origen de ese tiempo de ejecución, de esa memoria o del ancho de banda de red.

Tabla II.I: Características de los sistemas operativos tradicional y de Windows Azure.

De la tabla anterior se encuentran detalles interesantes como:

Se ve que Windows Azure resuelve el mismo problema que un sistema operativo tradicional a la hora de abstraer ciertos detalles.

Un sistema operativo tradicional abstrae del hardware y un sistema operativo en la Nube abstrae del centro de datos subyacente.

El ya mencionado FabricController es el componente de la arquitectura de Windows Azure que orquesta y abstrae a las aplicaciones de la infraestructura, jugando el papel

que en un sistema operativo tradicional jugaría la capa de abstracción de hardware o HAL¹⁶.

2.2.2. Arquitectura de una aplicación¹⁷.

La arquitectura de un servicio alojado en Windows Azure se basa en componentes auto-contenidos desarrollados típicamente con código .NET. Estos componentes son conocidos en Windows Azure como roles.

A continuación se puede ver el esquema de la arquitectura de una aplicación Windows Azure:

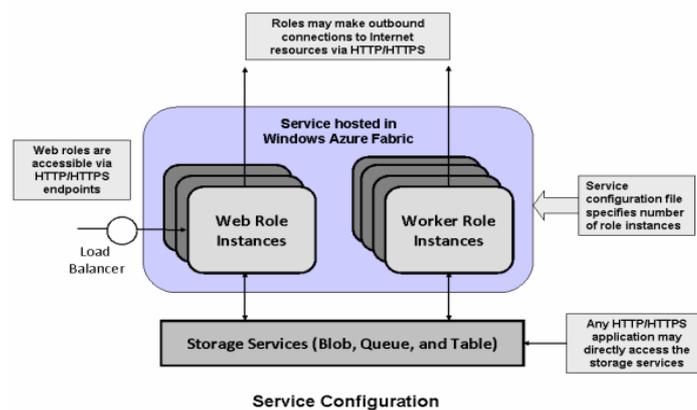


Figura II.7: Arquitectura de una aplicación Windows Azure.

Una aplicación alojada en Windows Azure se implementa como la composición de uno o más roles. Estas aplicaciones pueden ejecutar una o más instancias de cada uno de estos roles. Este detalle se define mediante simples archivos de configuración.

En función del número de instancias de un determinado rol, este rol recibirá más o menos capacidad de proceso.

¹⁶ **HAL:** Hardware AbstractionLayer, es un elemento del Sistema Operativo que funciona como una interfaz entre el software y el hardware del sistema, proveyendo una plataforma de hardware consistente sobre la cual correr las aplicaciones.

¹⁷ <http://www.estoyenlanube.com/recursos/windows-azure/arquitectura-de-una-aplicacion-windows-azure>.

Cada una de las instancias es replicada en múltiples máquinas físicas. Cada instancia recibe aproximadamente el equivalente a una máquina con las siguientes características: un núcleo de aproximadamente 1.7 MHz y 2 Gb de RAM.

Existen dos tipos de roles en Windows Azure:

Web rol: Un 'web rol' es una aplicación basada en web accesible mediante HTTP o HTTPS. Un web rol es alojado en un entorno de ejecución que soporta un subconjunto bastante amplio de ASP.NET y Windows Communication Foundation.

Worker rol: Un 'worker role' es un proceso que corre en segundo plano. Sería el equivalente a un servicio de Windows en la plataforma Windows Azure. Un worker rol se puede comunicar con los servicios de almacenamiento y de colas de Windows Azure y trabajar en conjunción con uno o varios web roles a través de ellos.

Windows Azure impone ciertas restricciones en tiempo de ejecución a lo que un rol puede hacer. Para ello utiliza la combinación de políticas de acceso a código (CAS) de .NET y políticas de seguridad de Windows.

Todos los tipos de roles puede establecer conexiones de salida hacia recursos en Internet usando HTTP o HTTPS y usando TCP/IP sockets, y atender peticiones entrantes y solamente sobre HTTP o HTTPS.

Todos los tipos de roles tienen acceso a ciertos servicios que la plataforma de ejecución de Windows Azure expone mediante las librerías del SDK de Windows Azure:

- Acceso al almacenamiento privado del rol. No se los debe confundir con los servicios de almacenamiento de Windows Azure. Se trata de almacenamiento local que se utiliza típicamente como cache. No se puede confiar en que este tipo de almacenamiento sea persistente en el tiempo y está bastante limitado en capacidad.
- Los servicios para trace (tracing) y diagnóstico de Windows Azure¹⁸.
- Servicios que permiten informar al FabricController del estado de la aplicación.

¹⁸**Diagnóstico de Windows Azure:** Es el registro que se lleva del control y seguimiento de las acciones de los eventos, para la detección de errores.

2.2.3. Creación de un nuevo proyecto.

Las Visual Studio Tools para Windows Azure y el SDK de Windows Azure son elementos imprescindibles para desarrollar aplicaciones con Visual Studio para Windows Azure.

Una vez instalados estos componentes en la máquina de desarrollo se podrá comenzar a crear proyectos Windows Azure, tal y como se puede observar en la siguiente imagen:

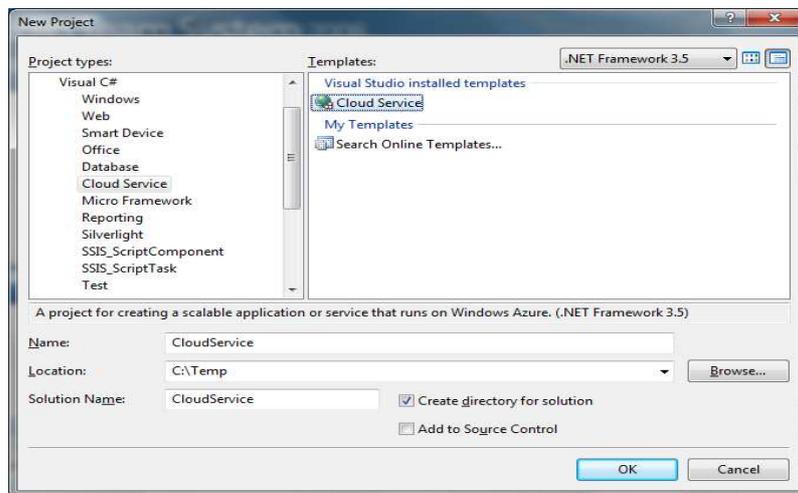


Figura II.8: Creación de un nuevo proyecto.

En la imagen se ha seleccionado un nuevo proyecto de tipo CloudService de Visual C# pero nada impediría crear un proyecto de este tipo en Visual Basic. NET. Es posible crear soluciones de Windows Azure que combinen roles desarrollados en diferentes lenguajes.

Se debe utilizar el.NET Framework 3.5 o superior, sino no se verá la opción de crear un proyecto de tipo CloudService. Una vez confirmada la creación del proyecto, aparecerá un asistente que permite añadir roles al proyecto de Windows Azure. Se podrá añadir tantos roles de tipo ASP.NET Web Rol o Worker Rol, tal y como se puede ver en la siguiente imagen.

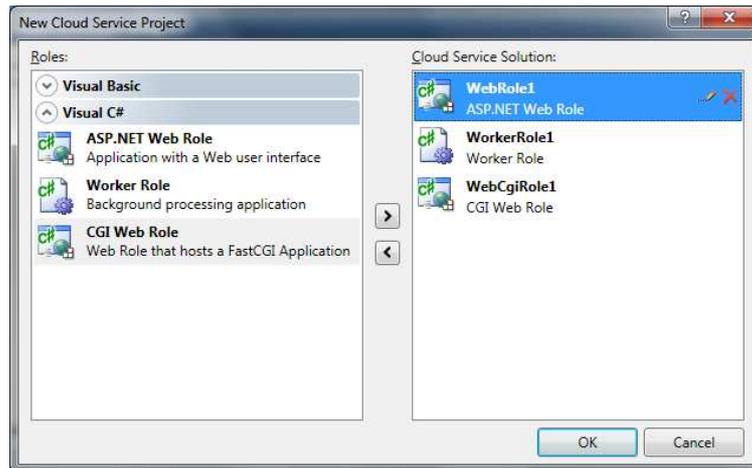


Figura II.9: Asignación de roles al nuevo proyecto.

Se puede editar el nombre de cada uno de los roles.

Se puede añadir más roles tras haber creado el proyecto es posible hacerlo sin problemas posteriormente.

Una vez aceptados los pasos dados en el asistente, se creará el proyecto.

2.2.4. El development Fabric.

Resulta evidente que por una simple cuestión de productividad no es viable tener que desplegar nuestras aplicaciones en la Nube de Windows Azure cada vez que se quiere depurarlas.

Además Windows Azure no permite depurar con Visual Studio las aplicaciones desplegadas.

Otro escenario es que no siempre se tiene que estar conectado a la Nube, puede que la máquina de desarrollo no tenga acceso a Internet en un momento dado.

Es por esto necesario simular de algún modo el entorno de ejecución de Windows Azure, el Fabric, en la máquina de desarrollo. Esta es la misión que lleva a cabo el DevelopmentFabric.

Como parte del SDK de Windows Azure se instala en la máquina de desarrollo este componente, DevelopmentFabric. Se puede ejecutarlo manualmente desde el menú de inicio pero lo habitual será que sean las propias Visual Studio Tools para Windows

Azure las que se encarguen de levantarlo de manera automática cuando se inicie la depuración o la ejecución de un proyecto de Windows Azure.

Cuando el DevelopmentFabric se encuentra en ejecución, se verá un icono similar a una ventana en el área de notificación del operativo Windows, tal y como se puede ver en la siguiente imagen:



Figura II.10: Ícono de confirmación de ejecución del Development Fabric.

Por medio de ese icono desde su menú contextual se podrá acceder a la interfaz del DevelopmentFabric.

El DevelopmentFabric permite alojar e interactuar con los servicios de Windows Azure que se está desarrollando sin tener que desplegarlos en el Fabric real de Windows Azure.

Este es el aspecto que tiene el DevelopmentFabric alojando servicios en la máquina de desarrollo:

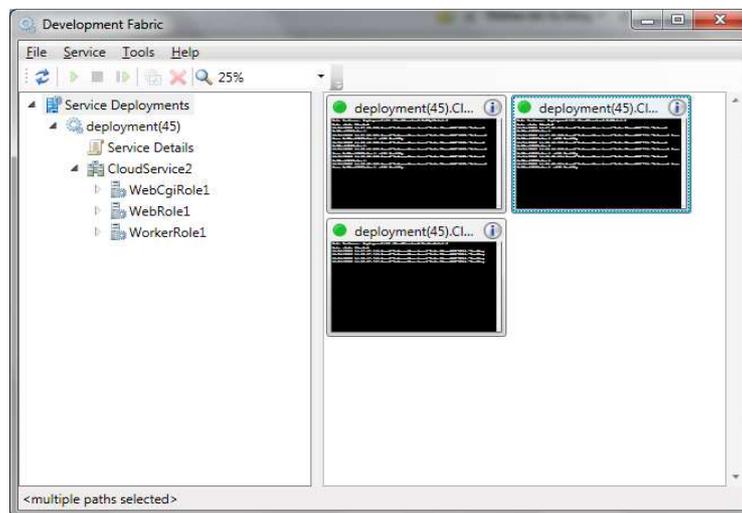


Figura II.11: Alojamiento de servicios en el Development Fabric.

El DevelopmentFabric permite llevar a cabo, entre otras, las siguientes labores:

- Monitorizar el estado de los servicios.

- Establecer el nivel de trazas¹⁹ y verlas.
- Limpiar el log²⁰ de cada instancia o servicios.
- Adjuntar el depurador de Visual Studio a los servicios.
- Inspeccionar el almacenamiento local de los servicios.

2.2.5. Partial trust y full trust.

En esta parte del documento se va a ver qué puede y qué no puede hacer una aplicación Windows Azure.

El Desarrollo de una aplicación de Windows Azure es muy similar a desarrollar una aplicación web en la plataforma .NET con ASP.NET, la naturaleza compartida de los recursos en la Nube exige establecer ciertas limitaciones al comportamiento de las aplicaciones desplegadas en Azure.

Dichas limitaciones están orientadas especialmente a evitar que, intencionadamente o por error, las aplicaciones de Windows Azure tengan comportamientos anómalos, traten de interferir las operaciones de otras aplicaciones, realicen ataques a máquinas en Internet, etc.

La plataforma .NET cuenta con un mecanismo llamado Code Access Security (CAS) que permite delimitar perfectamente que acciones puede llevar a cabo una aplicación .NET.

Este mecanismo es aprovechado por Windows Azure para establecer límites a lo que pueden hacer las aplicaciones.

¹⁹**Traza:**En ASP.NET le permite ver información de diagnóstico acerca de una única solicitud para una página ASP.NET, permite seguir la ruta de acceso de ejecución de una página, mostrar información de diagnóstico en tiempo de ejecución y depurar la aplicación.

²⁰**Log:**Es un registro de actividad de un sistema, instancia o servicio, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones.

2.2.5.1. Políticas de acceso a código.²¹

2.2.5.1.1. Código de seguridad de acceso.

La seguridad de acceso a código (CAS) es un sistema de seguridad que permite a los administradores y desarrolladores de aplicaciones controlar y autorizar a los usuarios. Con CAS, se puede permitir que una aplicación lea y escriba en el registro. Se puede controlar la autorización usando seguridad basada en roles, de:

- El sistema de archivos.
- El registro.
- Impresoras.
- Los registros de sucesos.

2.2.5.1.2. Elementos de seguridad de acceso a código.

Cada sistema de seguridad necesita una forma de identificar a los usuarios y determinar lo que un usuario puede o no puede hacer. Como CAS identifica y asigna los permisos a las aplicaciones en lugar de a la gente, que no puede utilizar los nombres de usuario, contraseñas y listas de control de acceso (ACL). En cambio, CAS identifica assemblies usando evidencias. Una evidencia es una manera de identificar la ubicación donde se almacena, un hash del código del assembly²², o su firma. Un conjunto de pruebas determina el código de grupo al que pertenece y le otorga permisos.

2.2.5.1.3. Evidencia.

²¹<http://vicrodg.blogspot.com/2010/11/seguridad-de-aplicacionesparte-1.html>.

²²**Assembly:** Colección de archivos o ficheros, agrupados juntos para formar una unidad lógica o biblioteca de código parcialmente compilado

La evidencia es la información que se obtiene en tiempo de ejecución acerca de un assembly para determinar a qué grupos de código pertenece. Las formas comunes de pruebas incluyen la carpeta o el sitio Web donde el assembly se está ejecutando y sus firmas digitales.

2.2.5.2. El nivel de confianza Partial Trust.

Este es el nivel de confianza que por defecto tendrá todo rol que se crea. Este es el nivel de confianza recomendado para la mayoría de aplicaciones. En este nivel de confianza se tiene las siguientes limitaciones:

- Se puede realizar casi todas las acciones en ASP.NET con 'medium trust'. Este es el nivel habitual en todos los hosting compartidos de ASP.NET. La mayor dificultad es que algunos controles de terceros mal diseñados exigen 'full trust' para funcionar correctamente.
- Se puede realizar consultas DNS.
- Solo se puede leer y escribir las variables de entorno TMP y TEMP, pero ninguna otra.
- No se tendrá acceso al log de eventos, pero como se verá Windows Azure tiene su propio sistema de trazas y monitorización.
- Solo se puede escribir y leer el archivo en el almacenamiento local de Windows Azure, pero no en ninguna otra ubicación del disco. Típicamente se utilizará el almacenamiento local de Windows Azure como cache.
- No se tiene acceso de ningún tipo a la característica de almacenamiento aislado de .NET.
- No se puede realizar peticiones por OleDb.
- Solo algunas de las características de reflexión están disponibles.
- Solo se puede abrir sockets²³ de salida, sobre TCP y hacia sitios externos.
- Solo se puede realizar conexiones a fuentes SQL externas.

²³**Sockets:** El protocolo TCP/IP gestiona el envío y la recepción de la información los denominados paquetes, que resultan de su división en trozos más pequeños. La biblioteca que controla el envío/recepción de estos paquetes se denomina socket

2.2.5.3. El nivel de confianza Full Trust.

El nivel de confianza 'full trust' confiere a la aplicación de Windows Azure privilegios adicionales. De estos privilegios adicionales los más notables son:

- La capacidad para poder llamar a código nativo mediante P/Invoke²⁴.
- La capacidad para poder usar ensamblados .NET que requieran 'full trust', esto es, todos aquellos que no estén marcados con el atributo *AllowPartiallyTrustedCallers*.
- La capacidad para leer, que no escribir, en el registro de Windows de la máquina sobre la que se está ejecutando el Fabric de Windows Azure.

Para configurar un rol de Azure para que se ejecute con 'full trust' en el archivo de definición del servicio (.csdef) se debe establecer el atributo *enableNativeCodeExecution* del rol correspondiente a true, tal y como se puede ver en el siguiente código:

```
<WorkerRole name="Role" enableNativeCodeExecution="true">  
  <ConfigurationSettings />  
</WorkerRole>
```

Figura II.12: Código para configurar un rol con Full Trust.

2.2.6. Diagnóstico y trazas.

Se ha dicho que en Windows Azure no se tiene acceso a recursos locales del sistema de archivos ni al log de eventos de Windows. Esto plantea la situación de necesitar un mecanismo diferente a los tradicionales para escribir las trazas de nuestra aplicación. Algo que permita detectar y diagnosticar comportamientos anómalos de la aplicación cuando ésta se encuentra desplegada en la plataforma Windows Azure.

El SDK de Windows Azure proporciona un 'trace²⁵listener²⁶', especialmente diseñado para ser utilizado en aplicaciones en la Nube.

²⁴**P/Invoke:** PlatformInvocationServices o Servicios de Invocación de Plataforma, es una de las formas para utilizar código no administrado en .NET.

²⁵**Trace:** Conjunto de métodos y propiedades que ayudan a trazar el camino de la ejecución del código.

²⁶**Listener:** Encargado de escuchar el o los eventos que ocurra en el emisor y responder rápidamente.

Este listener está implementado en la clase *DiagnosticMonitorTraceListener* del espacio de nombres *Microsoft.WindowsAzure.Diagnostics* y deriva de la conocida clase **TraceListener** de la plataforma .NET. Trabajar con este listener es, por tanto, idéntico a trabajar con cualquier otro trace listener de .NET.

Lo primero que se debe hacer es asegurar de que en el archivo de configuración (.config) de la aplicación se configuró correctamente el trace listener de Azure:

```
<system.diagnostics>
  <trace>
    <listeners>
      <add type="Microsoft.WindowsAzure.Diagnostics.DiagnosticMonitorTraceListener, Microsoft.WindowsAzure.Diagnostics, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" name="AzureDiagnostics" >
        <filter type="" />
      </add>
    </listeners>
  </trace>
</system.diagnostics>
```

Figura II.13: Código del archivo .config.

Una vez configurado correctamente el trace listener de Azure, se puede lanzar trazas desde la aplicación usando las clases *Debug* y *Trace* del Framework.

```
Trace.TraceInformation("La aplicación se está ejecutando");
...
Trace.TraceWarning(";Se produjo un warning!");
...
Trace.TraceError(";Se produjo un error critico!");
...
```

Figura II.14: Trazas ejecutándose.

Como parámetro del método Trace correspondiente al nivel del error, se debe pasar el mensaje que se desea que aparezca en el log. Después se podrá establecer el nivel de detalles del log en un momento dado, tanto en el DevelopmentFabric como en el Fabric real de Windows Azure.

Nota: Hay que destacar que todas aquellos mensajes de traza que se lance con el nivel Critical, serán enviados también como alertas y solo deberían ser utilizados en situaciones que exijan una intervención urgente por parte de un operador de la aplicación.

El resultado de las trazas se puede ver en el DevelopmentFabric:

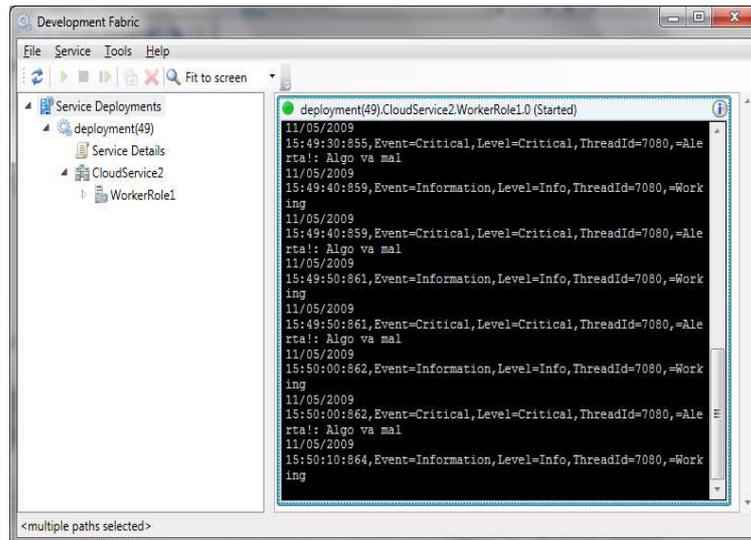


Figura II.15: Trazas ejecutándose en el Development Fabric.

Se puede cambiar el nivel de tracing en el DevelopmentFabric desde el menú contextual del rol correspondiente:

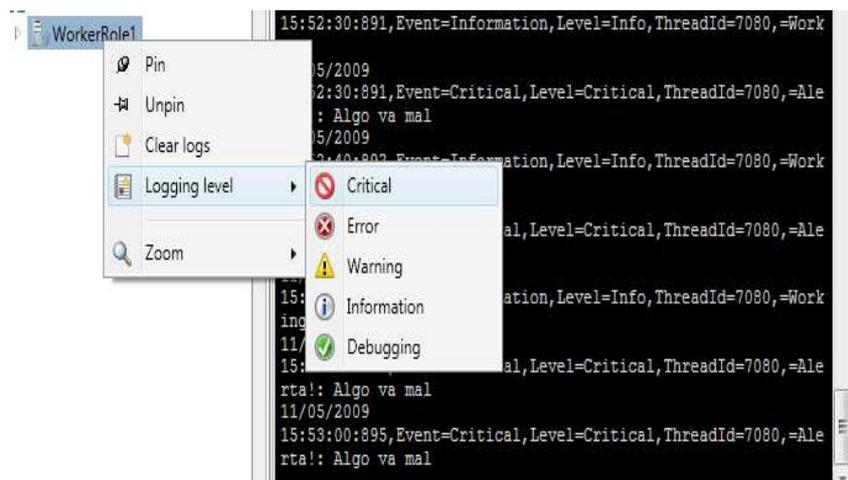


Figura II.16: Pasos para cambiar el traceo en el Development Fabric.

Una vez establecido el nivel de log a Critical, solo las trazas de este nivel aparecen.

Evidentemente ver las trazas de la aplicación en el DevelopmentFabric es muy útil durante el desarrollo y depuración de la misma, pero también se necesitará ver como se está comportando la aplicación una vez desplegada en la Nube. Para ello se cuenta con la posibilidad de almacenar las trazas de la aplicación en el almacenamiento de Windows Azure. Para ello basta con utilizar el método Start de la clase

DiagnosticsMonitor para configurar una cadena de conexión al almacenamiento de Azure.

```
DiagnosticMonitor.Start("DiagnosticsConnectionString");
```

Figura II.17: Configurar la cadena de conexión para el almacenamiento de Azure.

2.2.7. Configuración en Windows Azure.

Todo desarrollador de .NET sabe que el lugar en que se debe poner la configuración de las aplicaciones es en un archivo *app.config* o *web.config*. Este mecanismo funciona perfectamente en Windows Azure siempre y cuando no se necesite cambiar los valores de los parámetros de configuración.

El motivo es que Windows Azure no permite cambiar la configuración almacenada en un archivo .config una vez que se ha desplegado a la Nube. La única opción para reconfigurar la aplicación sería volverla a desplegarla por completo. Evidentemente esto no es operativo y hace que este mecanismo de configuración tradicional no sea todo lo útil que debiera en este tipo de aplicaciones para la Nube.

Así, Windows Azure define un mecanismo propio de configuración basado en dos tipos de archivos:

2.2.7.1. Archivo de definición de servicio.

El archivo de **definición** del servicio (.csdef), especifica los endpoints²⁷ del servicio, y establece la declaración de los parámetros de configuración del mismo. Este archivo no puede ser cambiado una vez que el servicio ha sido desplegado. Este es el aspecto típico de este archivo:

²⁷**Endpoint:** Terminal o punto de acceso a un servicio.

```

<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="CloudService" xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WorkerRole name="WorkerRole" enableNativeCodeExecution="false">
    <ConfigurationSettings>
      <Setting name="message"/>
    </ConfigurationSettings>
  </WorkerRole>
</ServiceDefinition>

```

Figura II.18: Archivo de definición de servicio

2.2.7.2. Archivo de configuración de servicio.

Se observa que el archivo de definición anterior declara un ajuste o parámetro cuyo nombre es *message*. Se puede establecer el valor de este parámetro en el archivo de **configuración** del servicio (.cscfg).

El archivo de configuración del servicio, como se puede deducir de su nombre, permite especificar las opciones de configuración para uno o más roles. Este archivo puede ser dinámicamente modificado una vez que la aplicación está desplegada en Windows Azure. A continuación se puede ver el aspecto de un archivo .cscfg:

```

<?xml version="1.0"?>
<ServiceConfiguration serviceName="CloudService" xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration">
  <Role name="WorkerRole1">
    <Instances count="1" />
    <ConfigurationSettings>
      <Setting name="message" value="Hola, Windows Azure" />
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>

```

Figura II.19: Archivo de configuración de servicio

En este archivo .cscfg, se puede ver el valor de configuración que se ha dado para el parámetro del servicio *message* declarado en el archivo de definición del servicio.

Una vez establecida la configuración del servicio, queda el siguiente paso, leerla. Para ello se cuenta una vez más con la clase **RoleEnvironment**. Esta clase proporciona un método, **GetConfigurationSettingValue** que permite pasarle el nombre del parámetro de configuración a leer (*message*) y devuelve su valor como una cadena.

```

if (RoleEnvironment.IsAvailable)
{
    string message = RoleEnvironment.GetConfigurationSettingValue("message");
}

```

Figura II.20: Parte del archivo de configuración de servicio

Nota: Se debe asegurar de que la aplicación está ejecutándose dentro del Fabric para poder invocar a los métodos de RoleEnvironment. Si no se recibirá una excepción.

Para más comodidad también se puede, desde las propiedades del rol, usar el editor de propiedades que proporcionan las extensiones de Azure para Visual Studio.

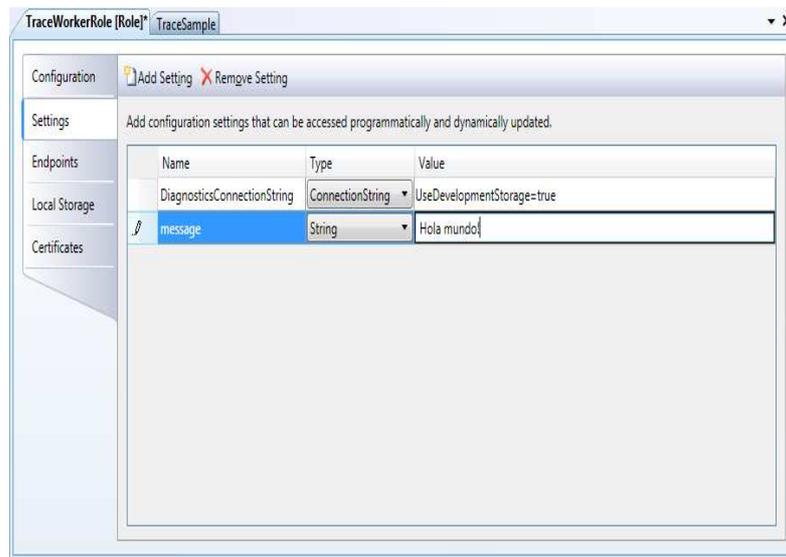


Figura II.21: Cuadro de diálogo del editor de propiedades del rol

2.2.8. Despliegue y versionado de aplicaciones.

Desplegar una aplicación de Windows Azure desde el entorno de desarrollo hasta la Nube es muy sencillo.

Pasos:

- Compilar la aplicación y comprobar que lo hace sin errores.
- Una vez compilada la aplicación, hay que publicarla usando el botón correspondiente. A pesar de lo que pueda parecer debido a su nombre, esto no publicará la aplicación en la Nube, sino que **generará el paquete** que contiene la aplicación.
- Acceder al portal de Windows Azure.
- Subir el paquete de la aplicación.
- Comprobar el correcto comportamiento de la misma en el entorno de prueba.
- Pasar la aplicación a producción.

El proceso es simple, pero exige pasar por un número considerable de pasos.

2.3. Almacenamiento en Windows Azure

La plataforma Windows Azure, además de los servicios para ejecución de aplicaciones en la Nube que se ha visto hasta ahora, también **proporciona servicios de almacenamiento** en la Nube. Estos servicios de almacenamiento están diseñados para ser consumido mediante REST de manera que están totalmente accesibles desde cualquier lenguaje de programación con capacidad para hacer peticiones usando HTTP como protocolo, lo que es tanto como decir cualquier lenguaje.

En Azure se cuenta con tres servicios de almacenamiento diferentes:

- **El servicio de BLOBS** (Blob Service).
- **El servicio de Tablas** (TableService).
- **El servicio de Colas** (QueueService).

Cada uno sirve a propósitos y necesidades diferentes, pero todos tienen algunas características en común:

- Son servicios diseñados para la alta disponibilidad y escalabilidad que las aplicaciones en la Nube exigen.
- Son servicios diseñados para ser accesibles mediante peticiones HTTP o HTTPS y un API basada en REST.
- Los servicios de almacenamiento siempre colgaran de una cuenta de almacenamiento de Windows Azure que define el namespace²⁸ de los servicios mediante una raíz URL única y común a los servicios que comparten cuenta.

2.3.1. Windows AzureTable.

²⁸**Namespace:**También llamado Espacio de Nombres es un medio para organizar clases dentro de un entorno, agrupándolas de un modo más lógico y jerárquico.

El servicio de tablas de Windows Azure proporciona **almacenamiento estructurado no relacional** basado en tablas. Es preciso recalcar de la frase anterior el hecho de que no se pueden establecer relaciones en estas tablas.

Como todos los servicios de almacenamiento de Windows Azure, proporciona un API REST para acceder a sus funcionalidades. En este caso el API REST de las tablas de Windows Azure sigue el contrato definido por ADO.NET Data Services, lo que proporciona una comodidad extra: se puede usar (y normalmente se usará) la *.NETClient Library for ADO.NET Data Services* para acceder a las tablas e incluso realizar consultas LINQ²⁹ (con ciertas limitaciones) sobre las tablas de Windows Azure, sin tener que trabajar directamente con REST. Es la *.NETClient Library for ADO.NET Data Service* quien se encarga de convertir las peticiones en las peticiones HTTP correspondientes.

2.3.1.1. Entidades y tablas.

Dentro de una cuenta de almacenamiento de Windows Azure se puede crear múltiples tablas. Las tablas se diferencian por su nombre. Dentro de las tablas se almacenarán entidades. Las entidades se representan en C# o en VB.NET como clases que derivan de la clase **TableServiceEntity** y que tiene una serie de propiedades públicas que serán almacenadas en la tabla cuando los objetos sean persistidos.

A continuación se observa cómo sería una entidad *Visita* que dispone de las propiedades *Nombre*, *Lugar* y *Descripción*.

²⁹ **LINQ**: Language Integrated Query es un conjunto de características que agrega capacidades de consulta eficaces a la sintaxis de los lenguajes C# y Visual Basic, incluye patrones estándar para consultar y actualizar datos.

```

public class VisitaEntity : TableServiceEntity
{
    public VisitaEntity()
        : base(Guid.NewGuid().ToString(), String.Empty)
    {
    }

    public VisitaEntity(string nombre, string lugar, string descripcion)
        : base(lugar, nombre)
    {
        Nombre = nombre;
        Lugar = lugar;
        Descripcion = descripcion;
    }

    public string Nombre { get; set; }
    public string Lugar { get; set; }
    public string Descripcion { get; set; }
}

```

Figura II.22: Entidad Visita

Las entidades son, desde el punto de vista de Azure Storage, **colecciones de pares propiedad-valor**, similares a las filas de una tabla en una base de datos relacional. Todas las tablas, y por tanto todas las entidades, tienen dos propiedades que siempre deben aparecer y que identifican de manera unívoca al objeto y el lugar físico donde se almacena (partición).

La primera propiedad es la clave de la partición, **PartitionKey**, que identifica a que partición pertenece una entidad. La única garantía que se tiene sobre las particiones es que acceder a entidades almacenadas en la misma partición va a tener, típicamente, un menor coste que acceder a entidades en particiones diferentes. Es por ello muy importante poner cuidado a la hora de elegir la clave de partición de las entidades para asegurar que entidades que se acceden normalmente al mismo tiempo comparten la misma clave de partición. También es importante que las particiones sean homogéneas en tamaño.

Por ejemplo, si se almacena datos de clientes, una posible clave de partición sería el código postal, si típicamente la aplicación trabaja con los clientes de una determinada zona para, por ejemplo, realizar estadísticas.

La segunda propiedad relevante es la clave de la entidad, **EntityKey**, que identifica de manera unívoca una entidad dentro de una partición.

La combinación de **PartitionKey** y **EntityKey** identifica de manera única una entidad de manera global. Sería el equivalente a una clave única compuesta o primaria compuesta en el mundo relacional.

A diferencia de las tablas de las bases de datos relacionales, una tabla puede contener entidades que tengan diferente número de propiedades. Además no existen conceptos relacionados con la integridad referencial, será la aplicación la encargada de mantener esta integridad de los datos.

Lógicamente antes de poder almacenar datos en una tabla se tendrá que crearla a ésta si es que no existe, algo muy fácil gracias al API de Azure.

```
var storageAccount = CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
...
storageAccount.CreateCloudTableClient().CreateTableIfNotExist(DondeHasEstadoDataContext.VisitasTableName);
```

Figura II.23: Creación de la tabla en caso de que no exista

2.3.1.2. Contexto de acceso a datos.

Otro concepto importante a la hora de trabajar con las tablas de Windows Azure es el **contexto**. El contexto es una clase que deriva de **TableContextService** y es el objeto que permite, usando LINQ, acceder a las tablas almacenadas en la cuenta de almacenamiento de Azure.

Al contexto se le pasará una URL que proporciona la plataforma Azure cuando se crea una cuenta de almacenamiento y que direcciona las tablas a la cuenta de manera única.

Esta es una típica clase de contexto de tabla:

```
public class DondeHasEstadoDataContext
    : TableServiceContext
{
    public const string VisitasTableName = "Visitas";

    public DondeHasEstadoDataContext(string baseAddress, StorageCredentials credentials) :
        base(baseAddress, credentials)
    {
    }

    public IQueryable<VisitaEntity> VisitasTable
    {
        get
        {
            return this.CreateQuery<VisitaEntity>(VisitasTableName);
        }
    }
}
```

Figura II.24: Clase de contexto de tabla

Las clases de contexto proporcionan acceso a las tablas mediante LINQ, devolviendo una implementación de *IQueryable* para la clase concreta de la entidad. Devolver un *IQueryable* va a permitir realizar consultas LINQ sobre la tabla.

2.3.1.3. Orígenes de datos.

Siempre que se accede a datos, se necesita algún tipo de objeto que permita conectar a la fuente de datos. Estos objetos en Windows Azure Storage se llaman por convención **'data sources'** y se utilizan para implementar en un solo objeto todas las operaciones de acceso a datos que se realice sobre las entidades almacenadas en una tabla.

Al contrario que las clases anteriores esta clase no tiene por qué derivar de ninguna otra, tal y como se puede ver en el siguiente fragmento de código:

```
public class VisitasDataSource
{
    private DondeHasEstadoDataContext _dataContext = null;

    public VisitasDataSource()
    {
        var storageAccount = CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
        _dataContext = new DondeHasEstadoDataContext(storageAccount.TableEndpoint.ToString(), storageAccount.Credentials);
        storageAccount.CreateCloudTableClient().CreateTableIfNotExist(DondeHasEstadoDataContext.VisitasTableName);
    }

    public IEnumerable<VisitaEntity> Select()
    {
        var results = from v in _dataContext.VisitasTable
                     select v;

        return results.AsTableServiceQuery<VisitaEntity>().Execute();
    }

    public void Delete(VisitaEntity itemToDelete)
    {
        _dataContext.AttachTo(DondeHasEstadoDataContext.VisitasTableName, itemToDelete, "");
        _dataContext.DeleteObject(itemToDelete);
        _dataContext.SaveChanges();
    }

    public void Insert(VisitaEntity newItem)
    {
        _dataContext.AddObject(DondeHasEstadoDataContext.VisitasTableName, newItem);
        _dataContext.SaveChanges();
    }
}
```

Figura II.25: Clase de Orígenes de datos

Se ve como el **'data source'** agrupa las operaciones que, haciendo uso del contexto que se creó en el constructor de la clase, se realiza sobre las entidades de la tabla. Nótese como el método *Select* utiliza LINQ para devolver una colección enumerable de entidades mediante el método de extensión *AsTableServiceQuery*.

2.3.1.4. La cadena de conexión.

Otro aspecto fundamental siempre que se habla de acceso a datos es establecer la cadena de conexión. Lo mismo ocurre en el almacenamiento de Windows Azure. Ya se ha comentado antes que el SDK de Windows Azure proporciona un entorno simulado del almacenamiento de Azure, el Windows AzureDevelopment Storage que corre en la máquina de desarrollo sobre SQL Server Express.

Para conectar al AzureDevelopment Storage se puede establecer la siguiente entrada en la sección de cadenas de conexión del archivo de configuración:

```
<connectionStrings>
  <add name="DataConnectionString" connectionString="UseDevelopmentStorage=true"/>
</connectionStrings>
```

Figura II.26: Cadena de conexión del archivo de configuración

Para obtener la cadena de conexión desde el código será tan simple como hacer:

```
var storageAccount = CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
_dataContext = new DondeHasEstadoDataContext(storageAccount.TableEndpoint.ToString(), storageAccount.Credentials);
```

Figura II.27: Cadena de conexión obtenido desde el código

Si se quiere posteriormente utilizar una cadena de conexión apuntando a un almacenamiento real en Azure, y no al de desarrollo, el editor de configuración de Visual Studio ayudará con un cuadro de diálogo como este:

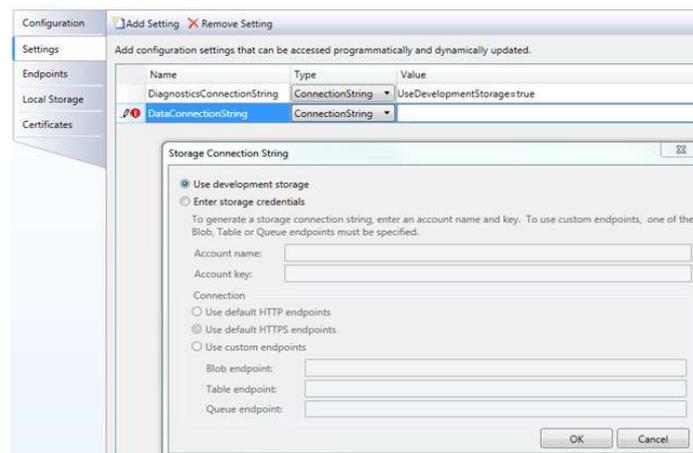


Figura II.28: Cadena de conexión con el asistente

2.3.2. Windows Azure Blob.

El servicio de Blob (*BinaryLargeObjects*, objetos binarios grandes) de Windows Azure Storage proporciona almacenamiento para entidades binarias o archivos. Este tipo de almacenamiento se elige cuando se tenga que almacenar archivos de imágenes, sonidos, videos, o archivos binarios del tipo que sea. El API de Blob es también un API basada en REST.

El servicio de Blob expone dos entidades fundamentales, **contenedores** y **BLOBS**. Todo blob debe alojarse dentro de un contenedor.

Existen **dos tipos** de BLOBS:

- **BLOBS de bloque**, que están optimizados para *streaming*³⁰.
- **BLOBS de página**, que están optimizados para el acceso aleatorio de escritura y lectura en lugares arbitrarios del blob.

Evidentemente se tiene que ser capaz de distinguir unos BLOBS de otros una vez almacenados, por ello los BLOBS soportan que se asocie metadatos³¹ con ellos.

Usando el API del servicio de Blob, los desarrolladores pueden crear una jerarquía de espacios de nombres similar a un sistema de archivos. Los nombres de los Blob pueden seguir una codificación que use un separador de 'directorios' configurable (típicamente el carácter "/") de manera que llamar a un Blob "Imágenes/MiImagen1" y a otro "Sonidos/MiSonido1" implique una organización a nivel lógico de los BLOBS almacenados.

El API de Azure Storage permite operaciones de enumeración de tal manera que se pueda enumerar todos los BLOBS que se encuentran dentro de una jerarquía determinada (por ejemplo "Imágenes" o "Sonidos").

2.3.2.1. Trabajo con BLOBS

Para trabajar con BLOBS el primer paso es establecer una conexión con el servicio de almacenamiento, de manera similar a como se hace en el caso del almacenamiento basado en tablas, basta con obtener la cadena de conexión, en este caso apuntando al Development Storage que se tiene en el archivo de configuración:

```
<connectionStrings>  
<add name="DataConnectionString" connectionString="UseDevelopmentStorage=true"/>  
</connectionStrings>
```

Figura II.29: Cadena de conexión

³⁰**Streaming:** Se trata de una corriente continua (sin interrupción), consiste en la distribución de audio o video por Internet, esta tecnología almacena todo en un búfer.

³¹**Metadatos:** Datos de los datos.

Y abrir una conexión con el siguiente código que usa la clase **CloudStorageAccount** para, a partir de la cadena de conexión, obtener un objeto de la clase **BlogStorageClient** que permitirá luego manipular los BLOBS construidos:

```
// Establecer la conexión con Windows Azure
var storageAccount = CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
_BlobClient = storageAccount.CreateCloudBlobClient();
```

Figura II.30: Conexión con windows azure

Una vez que se tiene el *BLOBStorageClient*, se tendrá que añadir un nuevo contenedor de BLOBS, ya que como se ha dicho todo Blob debe estar en un contenedor. Esta es una tarea simple y directa gracias a la clase **CloudBlobContainer** y su método **CreateIfNotExists**, tal y como se muestra en el siguiente código:

```
// Obtener el contenedor y cleararlo si no existe
_BlobContainer = _BlobClient.GetContainerReference("fotos");
_BlobContainer.CreateIfNotExists();
```

Figura II.31: Creación del contenedor

Los contenedores de BLOBS **pueden ser públicos o privados**. En este caso se observa la creación de un contenedor público, que es el valor por defecto. Es importante tener en cuenta que cualquiera puede acceder a un contenedor público. Si el contenedor fuese privado, solo conociendo la clave de almacenamiento y la cuenta de Azure Storage se podría acceder.

Subir el contenido de un Blob al almacenamiento es una tarea directa también. Basta con obtener una referencia al Blob, invocar al método **UploadFromStream** y establecer las propiedades y metadatos del blob, tal y como se puede ver en el siguiente código:

```
// Creamos un nombre único para el blob y creamos una referencia a ese blob
var blob = _BlobContainer.GetBlobReference(Guid.NewGuid().ToString() + Path.GetExtension(FUPhoto.FileName));
//Subimos el archivo al almacenamiento
blob.UploadFromStream(FUPhoto.FileContent);

// Establecemos los metadatos del blog
blob.Metadata["Name"] = TBPhotoName.Text;
blob.Metadata["Author"] = TBAuthor.Text;
blob.SetMetadata();

// Establecemos el tipo de contenido del blob
blob.Properties.ContentType = FUPhoto.PostedFile.ContentType;
blob.SetProperties();
```

Figura II.32: Establecimiento de las propiedades para el blob

Para enumerar el contenido de un contenedor de BLOBS se debe llamar al método **ListBlobs**, tras obtener una referencia al mismo. Después se puede ir accediendo a las

propiedades y metadatos de cada uno de ellos, pero para ello primero hay que llamar a **FetchAttributes**, que obliga la recogida de los metadatos asociados a cada Blob. El siguiente fragmento de código se observan estas operaciones:

```
// Obtenemos un lista de todos los blobs
var blobs = _BlobContainer.ListBlobs();
var photos = new List<PhotoMetadata>();

//Para cada Blob que almacena una foto, obtenemos los metadatos de la foto
//para mostrarlos en un grid.
foreach (var blobItem in blobs)
{
    var cloudBlob = _BlobContainer.GetBlobReference(blobItem.Uri.ToString());
    cloudBlob.FetchAttributes();
    photos.Add(new PhotoMetadata(blobItem.Uri, cloudBlob.Metadata["Name"], cloudBlob.Metadata["Author"]));
}
```

Figura II.33: Propiedades y características del blob

Nota: La clase PhotoMetadata es una simple clase que se ha creado en el código para contener los metadatos de las fotografías.

Por último eliminar un Blob es tan simple como llamar al método **DeleteIfExists** sobre la referencia a un Blob, tal y como se observa a continuación:

```
// Obtener la referencia al blob y eliminarlo
var blob = _BlobContainer.GetBlobReference(blobUri);
blob.DeleteIfExists();
```

Figura II.34: Obtener la referencia al blob y eliminación

2.3.3. Windows AzureQueue.

El servicio de colas de Windows Azure proporciona un mecanismo fiable y persistente para la comunicación entre aplicaciones de Windows Azure. El API de colas de Windows Azure, construido también sobre REST, está basado en dos tipos de abstracciones: **colas y mensajes**.

Las colas soportan atributos que se especifican como pares clave-valor y que permiten asociar metadatos a éstas, de manera que se las pueda identificar o mantener datos asociados a las mismas.

Cada cuenta de almacenamiento puede contener un número ilimitado de colas de mensajes y cada cola puede contener un número de mensajes ilimitado. La principal limitación en las colas de Windows Azure viene marcada por el hecho de que **el tamaño máximo de mensaje es 8KB**

Cuando un mensaje se lee desde una cola, el consumidor del mensaje es responsable de eliminarlo tras haberlo procesado. Una vez el mensaje se ha leído, durante un periodo de tiempo no estará disponible para otros consumidores. Si no se borra el mensaje en ese intervalo de tiempo, su visibilidad es restablecida de manera que otros consumidores pueden procesarlo.

2.3.3.1. Utilización de Colas.

Para utilizar el servicio de colas de Windows Azure el primer paso que se debe dar es conectarse a una cola. Lógicamente se crea si no existe. El proceso es similar al visto para otros tipos de almacenamiento y ya resulta familiar.

Existe un método **FromConfigurationSetting** en la clase **CloudStorageAccount** que permite configurar la conexión al almacenamiento de Azure. A continuación se puede crear un cliente para el almacenamiento de colas llamando al método **CreateCloudQueueClient**. El método **GetQueueReference** del recién creado **CloudQueueClient** permite obtener una referencia a la cola a través de la que llamará al método **CreateIfNotExists**, que crea la cola si es necesario.

Se puede ver el proceso en el siguiente ejemplo de código:

```
// Establecer la conexión con Windows Azure Storage
var storageAccount = CloudStorageAccount.FromConfigurationSetting("DataConnectionString");
//Obtener un cliente para el servicio de colas
_queueClient = storageAccount.CreateCloudQueueClient();
//Obtener una referencia a la cola y crearla si no existe
_queue = _queueClient.GetQueueReference("cola");
_queue.CreateIfNotExists();
```

Figura II.35: Utilización de colas

El siguiente paso es enviar un mensaje a la cola. Para hacerlo sólo se necesita llamar al método **AddMessage** a través de la referencia a la cola que se haya obtenido:

```
//Añadimos el mensaje a la cola
_queue.AddMessage(new CloudQueueMessage(GetNumberOfPhotos()));
```

Figura II.36: Añadir mensajes a la cola

Este método permite enviar un mensaje como texto o como un array³² de bytes, usando dos sobrecargas.

El proceso de consumir un mensaje es igualmente directo y simple. Basta con llamar al método **GetMessage** de la referencia a la cola. En caso de haber un mensaje en la cola el método devolverá el contenido del mismo, y se obtendrá un nulo (*null*) si está vacía. Para acceder al contenido del mensaje se puede usar la propiedad **AsString** (si el mensaje se envió como un string) o **As Bytes** (si el mensaje se envió como un array de bytes).

```
//Vemos si hay un mensaje en la cola
//Si lo hay leemos el número de fotos
CloudQueueMessage message = _Queue.GetMessage();

if (message != null)
{
    int numberOfPhotos = int.Parse(message.AsString);

    //Procesamos el mensaje
    ...

    //Borramos el mensaje
    _Queue.DeleteMessage(message);
}
```

Figura II.37: Consumir el mensaje de la cola

Nunca olvidar que el código que consume y procesa el mensaje es el encargado de eliminar el mensaje de la cola.

2.3.4. Del development Storage al almacenamiento en la Nube

Se ha explicado el almacenamiento de Azure local proporcionado por el Development Storage y construido sobre SQL Server.

El siguiente paso lógico es subir el almacenamiento a la Nube. Este proceso es muy sencillo y consiste en dos pasos:

El primero crear una nueva cuenta de almacenamiento en la Nube a través del portal de Azure, lo que proporcionará el URI de la cuenta de almacenamiento y la clave de la misma.

³²**Array:** Arreglo lógico de datos de un determinado tipo

El segundo cambiar la cadena de conexión al almacenamiento de Azure para que en lugar de usar el almacenamiento local utilice la cuenta en la Nube.

Hasta ahora se ha estado usando la siguiente cadena de conexión:

```
<Setting name="DataConnectionString" value="UseDevelopmentStorage=true" />
```

Figura II.38: Cadena de conexión simple

Pues bien ahora se debe usar una cadena de conexión similar a la siguiente:

```
<Setting name="DiagnosticsConnectionString" value="DefaultEndpointsProtocol=https;AccountName=ElNombreDeTuCuenta;AccountKey=" />
```

Figura II.39: Cadena de conexión con la cuenta de azure

DefaultEndpointsProtocol puede tomar **https** o **http** como valor y permite especificar qué protocolo se quiere usar para el acceso a la cuenta de almacenamiento. Típicamente será **Https** por motivos de seguridad, ya que así el tráfico irá cifrado y se podrá autenticar al servidor para que no sea suplantado.

AccountName, será el nombre con el que se ha creado la cuenta de almacenamiento en el portal de Azure.

AccountKey, será la clave que el portal de Windows Azure proporciona al crear la cuenta de almacenamiento, tal y como se observa en la siguiente figura:

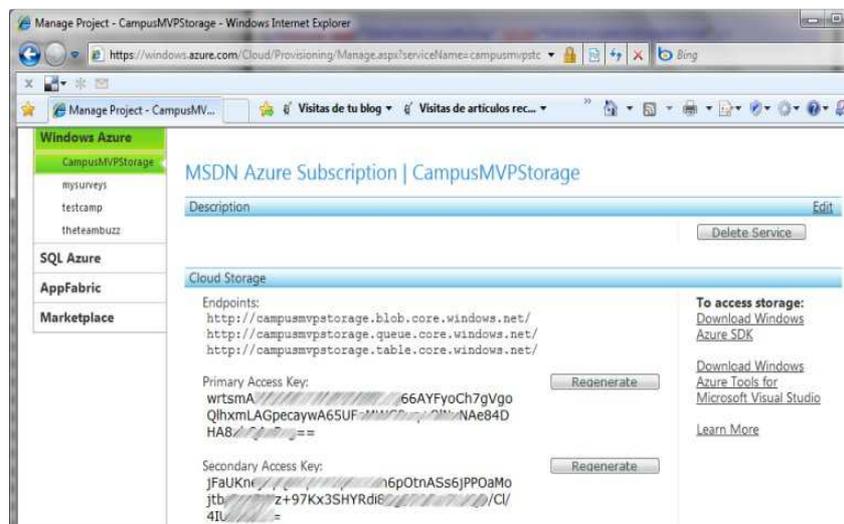


Figura II.40: Propiedades de la cadena de conexión con el asistente

2.3.4.1. Claves para proteger el almacenamiento.

Ambas claves son equivalentes y ambas sirven para acceder a la cuenta de almacenamiento, así que se puede utilizar una u otra indistintamente. Generalmente se usará la primera y se desplegará en la aplicación Azure.

Al tener dos claves lo que se consigue es que no exista ni un segundo de parada de la aplicación si se necesita cambiarla.

2.3.4.2. Pasos para cambiar la clave sin parar el servicio.

El procedimiento es el siguiente:

1.- Cambiar el archivo de configuración de la aplicación Azure para que pase a utilizar la segunda de las claves. Este cambio es instantáneo y no es necesario desplegar de nuevo la aplicación. A partir de este momento se está usando la segunda clave y la primera no hace falta.

2.- En la pantalla anterior se tiene q regenerar la primera clave. A partir de ese mismo instante la clave anterior queda invalidada.

3.- Opcionalmente, se vuelve a cambiar la configuración para usar la nueva clave primera, aunque no es necesario y se puede quedar únicamente con la segunda durante el tiempo que se desee, haciendo el procedimiento en sentido inverso si se viese comprometida.

Como se ve realizar esta actividad es realmente útil para no interrumpir en ningún momento el funcionamiento de la aplicación Azure, cosa que sí se tendría que hacer en caso de que sólo existiera una clave.

CAPÍTULO III

3. Análisis

3.1. Costos del despliegue de una aplicación web en los servidores convencionales

3.1.1. Generalidades

La aplicación de la Asociación de Escuela de Ingeniería en Sistemas Informáticos está desarrollada en la plataforma ASP.NET 4.0 por lo tanto se necesita ponerla en producción para el acceso de los usuarios que en este caso son los estudiantes de la Escuela de Ingeniería en Sistemas.

La Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica cuenta con servidores propios ubicados en el DESITEL (Departamento de Sistemas y

Telemática), y la Escuela de Ingeniería en Sistemas tiene una infraestructura ubicada en el laboratorio de Interoperabilidad y Desarrollo.

3.1.2. Consideraciones Generales

Para realizar el análisis se ha obtenido los datos por medio de una encuesta al Director de DESITEL Dr. Carlos Buenaño (Ver Anexo 1 Parte 1) y al Técnico de los Laboratorios de la Escuela de Ingeniería en Sistemas Ing. Carlos Jara (Ver Anexo 1 Parte 2).

3.1.3. Ubicación de los Servidores

Los servidores se encuentran ubicados en la Provincia de Chimborazo, Cantón Riobamba, y específicamente en la Escuela Superior Politécnica de Chimborazo, en el Departamento de Sistemas y Telemática (DESITEL), la misma que reside en la Panamericana Sur Km 1 ½ y también en el Laboratorio de Interoperabilidad y Desarrollo de la Escuela de Ingeniería en Sistemas.

3.1.4. Consideraciones con respecto Departamento de Sistemas y Telemática.

De acuerdo con las respuestas obtenidas se tiene:

En los servidores institucionales administrados por DESITEL es imposible desplegar la aplicación realizada para la Asociación por los siguientes motivos:

1. No cuenta con el suficiente número de servidores, ya que en la actualidad todos están funcionando a su capacidad máxima.
2. Una aplicación desarrollada en el Sistema Operativo Windows por sugerencias del Gobierno no puede ser puesta en producción, ya que todo se está migrando a servidores Linux.

3. La Escuela de Ingeniería en Sistemas cuenta con infraestructura propia razón por la cual se debe desplegar la aplicación en dichos servidores.

3.1.4.1. Consideraciones con respecto a la Escuela de Ingeniería en Sistemas.

Por lo antes expuesto se ha realizado la encuesta también al Técnico de los laboratorios de la Escuela de Ingeniería en Sistemas obteniendo la siguiente información:

1. La aplicación puede ser desplegada sin mayor problema pero con la condición de que vaya a ser solo para cuestiones de prueba por lo que no cuentan con amplia infraestructura para mantenerla siempre en producción por las características de la misma.
2. En el caso de que exista la predisposición y los recursos necesarios se tiene que realizar lo siguiente:
 - a. Oficio al Director de Escuela de Ingeniería en Sistemas Ing. Raúl Rosero solicitando permiso.
 - b. Oficio al Director de DESITEL Dr. Carlos Buenaño solicitando dominio y además una IP pública.
 - c. Instalar y configurar el Servidor de la Escuela con el Técnico Informático.
3. Para que se mantenga en producción se debe invertir en una nueva máquina con las características necesarias para poder desplegar.
4. En la Escuela Superior Politécnica de Chimborazo para realizar una nueva compra se tiene que realizar una serie de trámites los mismos que se detallan a continuación:
 - a. Solicitar al Sr. Decano de la Facultad de Informática y Electrónica que incluya en el Plan Anual la adquisición de un nuevo servidor con las características necesarias.
 - b. Proceso de Compra en el Portal de Compras Públicas.
 - c. Esperar a que el equipo llegue a la dependencia encargada.

3.1.4.2. Costos necesarios para desplegar la aplicación en Servidores Locales.

3.1.4.2.1. Costos para comprar nuevo servidor.

Tomando en cuenta las consideraciones anteriores el costo anual es:

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
Papeles Politécnicos	10	2.80	28.00
DVD's de Respaldo	5	0,50	2,50
Sueldo al Responsable. (Incluye el mantenimiento).	12	264,00	3160,00
Servidor: ³³ HP ML150 G6 E5504 HP SAS/SATA US Svr Intel® Xeon® Processor (2.00 GHz, 4MB L3 Cache) / Memory 2 GB (1 x 2 GB) PC3-10600E/Network Controller Embedded HP NC107i PCI Express Gigabit /HP Smart Array P410 controller Raid Controller (RAID 0, 1, 0+1)/ Hard Drive None ships standard / Maximum Power Supply 460W Non-Hot Plug / Non-Redundant Power Supply / HP Half-Height SATA DVD-ROM Optical Drive.	1	1312,00	1312,00
Consumo de energía eléctrica.	12	27,49	329,88
UPS que soporte el servidor.	1	50,00	50,00
Licencias	0	0,00	0,00
Total			<u>4882,38</u>

Tabla III.II:Costos en la compra de un nuevo servidor.

Análisis para el despliegue de la aplicación y el primer año en producción.

En la tabla anterior no se toma en cuenta el costo de las licencias ya que en la Encuesta y Entrevista realizada al Director del [Departamento de Sistemas y Telemática ESITEL](#) de la Escuela Superior Politécnica de Chimborazo, [Sr. Ing. Juan Perez](#), el día 25 de

³³ El precio puede variar sin previo aviso,
<http://h18000.www1.hp.com/products/quickspecs/Division/Division.html#10455>, Margarita Pico R.
Email mpico@siglo21.NET

Octubre del 2011³⁴ manifiesta que la ESPOCH cuenta con las necesarias para el uso de las aplicaciones.

El cálculo del consumo de energía eléctrica es:

Según los valores de la Empresa Eléctrica Riobamba S.A.

CARGA	POTENCIA (WATIOS)	TIEMPO/USO (HORAS/DÍA)	CONSUMO MENSUAL(KWH)
Refrigerador	300	8	72
Ducha	3000	0,5	45
Plancha	1000	0,5	15
Foco	100	4	12
Foco ahorrador	20	4	2,4
Fluorescente	40	4	4,8
Televisor	100	4	12
Computador	300	6	54

Tabla III.III:Consumo aproximado de sus electrodomésticos.

Para este cálculo se utilizaron los valores de consumo de energía eléctrica para un computador, pero como es para el uso de un servidor, se consume un poco más de lo habitual:

WATIOS	HORAS	CONSUMO AL MES
300	6	54 kwh
460	24	¿Costo?

Tabla III.IV: Cálculo con una regla de tres compuesta.

CÁLCULOS
costo= $460 * 24 * 54 / 300 * 6$
costo= $596160 / 1800$
costo= 331,2 kwh
costo= $331,2 \text{ kwh} * 0,083 \text{ dólares}$ ³⁵
costo= 27,49 dólares

Tabla III.V: Cálculos del costo de energía eléctrica.

³⁴ Anexo 1.

³⁵ Información por parte del Ing. Rafael Pino funcionario de la Empresa Eléctrica Riobamba S.A.

3.1.5. Proyección Económica.

Una vez encontrado el costo para un año, se procede a hacer una proyección económica utilizando los servidores institucionales.

Para éstas proyecciones en lo que respecta a los sueldos de los técnicos, consumo de energía eléctrica, gastos de suministros se ha utilizado la tabla del Anexo 2.

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
DVD`s de Respaldo	5	0,50	2,50
Sueldo de empleados	12	287,50	3450,00
Consumo de energía eléctrica.	12	27,49	329,88
Suministros para Mantenimientos	1	100,00	100,00
Licencias	0	0,00	0,00
Total			<u>3882,38</u>

Tabla III.VI: Cálculos del costo al segundo año.

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
DVD`s de Respaldo	5	0,50	2,50
Sueldo de empleados	12	311,00	3732,00
Consumo de energía eléctrica.	12	27,49	329,88
Suministros para Mantenimientos	1	100,00	100,00
Licencias	0	0,00	0,00
Total			<u>4164,38</u>

Tabla III.VII: Cálculos del costo al tercer año

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
DVD`s de Respaldo	5	0,50	2,50
Sueldo de empleados	12	334,50	4012,00
Consumo de energía eléctrica.	12	27,49	329,88
Suministros para Mantenimientos	1	100,00	100,00
Licencias	0	0,00	0,00
Total			<u>4444,38</u>

Tabla III.VIII: Cálculos del costo al cuarto año

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
DVD`s de Respaldo	5	0,50	2,50
Sueldo de empleados	12	358,00	4296,00
Consumo de energía eléctrica.	12	27,49	329,88
Suministros para Mantenimientos	1	100,00	100,00
Licencias	0	0,00	0,00
Total			<u>4728,38</u>

Tabla III.IX: Cálculos del costo al quinto año

RECURSOS	CANTIDAD	P. UNITARIO	P. TOTAL
DVD`s de Respaldo	5	0,50	2,50
Sueldo de empleados	12	381,50	4578,00
Consumo de energía eléctrica.	12	27,49	329,88
Suministros para Mantenimientos	1	100,00	100,00
Licencias	0	0,00	0,00
Total			<u>5010,38</u>

Tabla III.X: Cálculos del costo al sexto año

3.1.5.1. Resumen y gráfico de la Proyección.

SERVIDORES CONVENCIONALES					
2011	2012	2013	2014	2015	2016
\$4882,38	\$3882,38	\$4164,38	\$4444,38	\$4728,38	\$5010,38

Tabla III.XI: Costos Proyectados

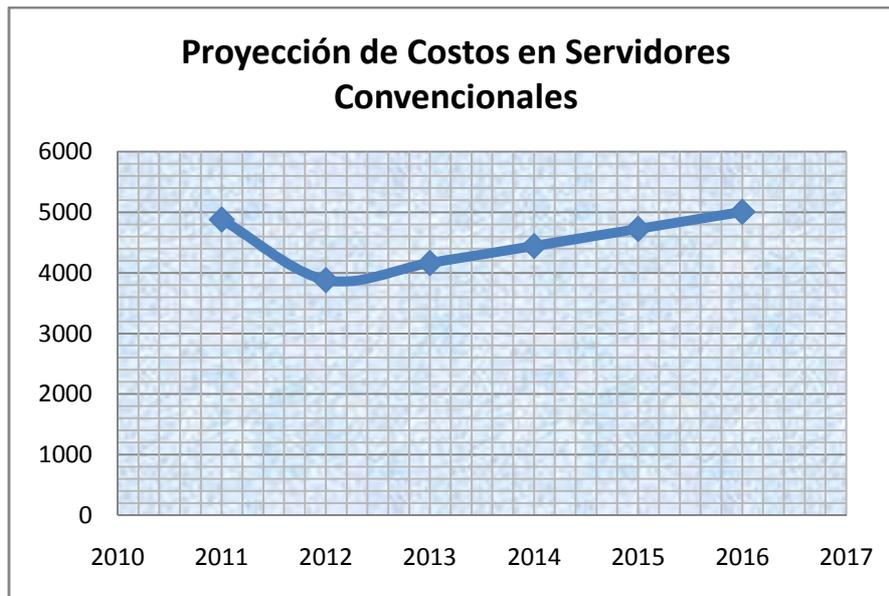


Figura II.41: Proyección de Costos en servidores convencionales

3.2. Costos de despliegue de una aplicación web en la plataforma de Windows Azure

3.2.1. Generalidades

Se tienen dos tipos básicos de ofertas para elegir al comprar una suscripción a la plataforma de Windows Azure. El primer tipo es ofertas de consumo. Este tipo no requiere compromiso - se paga sólo por lo que utiliza. El segundo tipo de oferta es una oferta de compromiso que proporciona un nivel significativo de descuento en el servicio a cambio de un compromiso de seis meses para pagar una cuota base mensual. Cualquier uso por encima de este importe se cobra en nuestras tarifas estándar de consumo. A continuación se presenta un resumen de los diferentes planes:

3.2.2. Ofertas de Consumo

- **Introducción Especial** – Oferta promocional con una cantidad de servicios de la plataforma Windows Azure proporcionados cada mes sin cargo. No requiere ningún

compromiso mensual y es el primer plan para elegir si se tiene alguna duda sobre cuánto va a utilizar cada mes. Para SQL Azure esta oferta sólo es válida durante 3 meses.

- **Consumo** - Flexible "Pague a medida que Avance" plan para todos los servicios de la plataforma de Windows Azure. Si necesita una o más suscripciones adicionales que no requieren ningún compromiso mensual, este es el plan para una empresa PYME.
- **MSDN Premium** - Oferta promocional con una cantidad mensual de servicios de la plataforma de Windows Azure proporcionada como un beneficio adicional a los suscriptores de MSDN Premium. Requiere que el usuario sea un suscriptor de MSDN Premium.

3.2.3. Ofertas de Suscripción

- **Núcleo Acelerador de Desarrollo** - Oferta promocional que ofrece horas de cómputo, almacenamiento de información, transferencias de datos, transacciones de Control de Acceso y conexiones de Bus de Servicios a un precio mensual con descuento para un término de seis meses. Esta oferta está diseñada para ofrecer a los desarrolladores Windows Azure y los recursos de bus de servicios de AppFabric y Control de Acceso que necesitan para desarrollar una solución particular u ofrecerla. Si se piensa en el ciclo de vida del proyecto del desarrollador como diseño-desarrollo-implementación-administración, esta oferta, esencialmente, alimenta la fase de desarrollo de su proyecto.
- **Acelerador Extendido de Desarrollo** - Oferta promocional incluye Núcleo Acelerador de Desarrollo más SQL Azure a un precio mensual con descuento para un término de seis meses. Esto es para aquellos desarrolladores que están incluyendo SQLAzure en sus soluciones.

Para todas las ofertas excepto la oferta de MSDN Premium, se proporciona a miembros de la red de socios de Microsoft un descuento adicional del 5% en todos los cargos excepto las transferencias de datos y almacenamiento de información.

A partir del 1er de febrero de 2010, se aplicaron las tasas de consumo estándar de la plataforma de Windows Azure.

3.2.3.1. Resumen de los costos.

Para entender en los costos en los que se incurren al momento de desplegar la aplicación en la plataforma de Windows Azure.

En la página de Microsoft se presentan las siguientes tablas.

COMPUTE INSTANCE SIZE	CPU	MEMORY	INSTANCE STORAGE	I/O PERFORMANCE	COST PER HOUR
Extra Small	1.0 GHz	768 MB	20 GB	Low	\$0.05
Small	1.6 GHz	1.75 GB	225 GB	Moderate	\$0.12
Medium	2 x 1.6 GHz	3.5 GB	490 GB	High	\$0.24
Large	4 x 1.6 GHz	7 GB	1,000 GB	High	\$0.48
Extra Large	8 x 1.6 GHz	14 GB	2,040 GB	High	\$0.96

Tabla III.XII: Costos según el tamaño de la Instancia de Cómputo.

SQL AZURE	
• Web edition	
Up to 1 GB relational database	\$9.99 / month
Up to 5 GB relational database	\$49.95 / month
• Business Edition	
Up to 10 GB relational database	\$99.99 / month
Up to 20 GB relational database	\$199.98 / month
Up to 30 GB relational database	\$299.97 / month
Up to 40 GB relational database	\$399.96 / month
Up to 50 GB relational database	\$499.95 / month
Data transfers	\$0.10 in / \$0.15 out / GB

Tabla III.XIII: Costos del SQL Azure.

ANCHO DE BANDA	
Data transfers input	\$0.10 GB
Data transfers output	\$0.15 GB
STORAGE	
Azure Storage	\$0.15 GB
Storage transactions	\$0.01 / 10K

Tabla III.XIV: Costos del Ancho de banda y Storage.

3.2.3.2. Costos en Windows Azure.

3.2.3.2.1. Características de la aplicación de la Asociación de Escuela.

La aplicación de la Asociación de Escuela de Ingeniería en Sistemas consta de todos los elementos que cuenta la plataforma Windows Azure, a continuación se detalla los mismos:

- Instancia Pequeña.
- SQL Azure, Web Edition, hasta 1 GB de almacenamiento,
- Almacenamiento en BLOBS aproximadamente 3GB y 50000 transacciones,
- Almacenamiento en Tablas aproximadamente 3GB y 50000 transacciones,
- Comunicación con Queues.
- Se promedia el ancho de Banda de 4GB de entrada y 4GB de salida.

Y un requerimiento especial de los estudiantes de la Asociación de Ingeniería en Sistemas es que la aplicación se ejecute 24 x 7 x 365.

3.2.3.3. Cálculo del Costo.

	CANTIDAD	P. UNITARIO	P. TOTAL
Instancia Pequeña.	12	86,4	1036,80
SQL Azure.	12	9,99	119,88
Ancho de Banda de Entrada	12	0,40	4,80
Ancho de Banda de Salida	12	0,60	7,20
AzureStorageBLOBS	12	1,40	16,80
Azure Storage Tables	12	1,40	16,80
Total			<u>1202,28</u>

Tabla III.XV: Costos con Windows Azure.

3.2.3.3.1. Detalle del cálculo mensual.

CARACTERÍSTICAS	COSTOS
Instancia Pequeña	$\$0,12 \times 24 \times 30 = \$86,40$
SQL Azure	$\$9,99 \times 1 = \$9,99$
Ancho de Banda de Entrada	$\$0,10 \times 4 = \$0,40$
Ancho de Banda de Salida	$\$0,15 \times 4 = \$0,60$
Azure Storage BLOBS	$3\text{GB} \times 2\text{GB} = 6\text{GB}$ $6 \times \$0,15 = 0,90$ $500000/1000 = 50$ $50 \times \$0,01 = 0,5$ $\$0,90 + \$0,50 = \$1,40$
Azure Storage TABLES	$3\text{GB} \times 2\text{GB} = 6\text{GB}$ $6 \times \$0,15 = 0,90$ $500000/1000 = 50$ $50 \times \$0,01 = 0,5$ $\$0,90 + \$0,50 = \$1,40$

Tabla III.XVI: Costos por mes.

3.2.4. Proyección Económica.

Una vez encontrado el costo para un año, se procede a hacer una proyección económica utilizando la Plataforma Windows Azure.

En este caso el costo en los posteriores años se va a mantener constante, ya que desde la fecha de lanzamiento hasta el día de hoy se han mantenido constantes, por lo tanto la siguiente tabla muestra los detalles.

3.2.4.1. Resumen y gráfico de la Proyección

PLATAFORMA WINDOWS AZURE					
2011	2012	2013	2014	2015	2016
\$1202,28	\$1202,28	\$1202,28	\$1202,28	\$1202,28	\$1202,28

Tabla III.XVII: Costos Proyectados

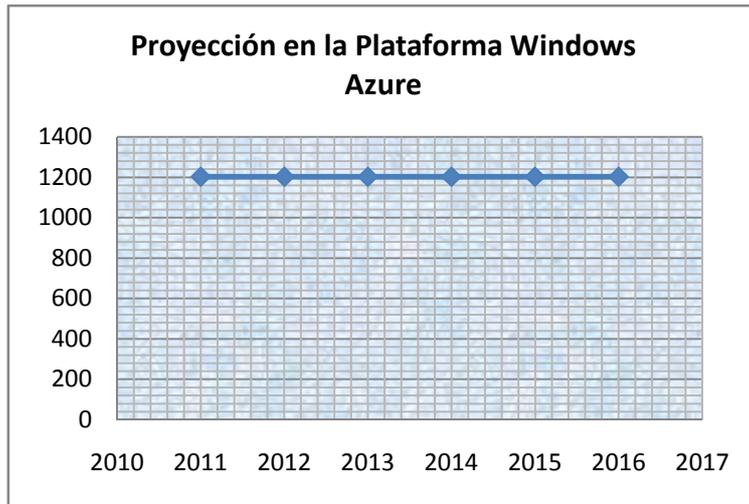


Figura II.42: Proyección en la Plataforma Windows Azure

3.3. Estudio comparativo.

A continuación se muestra las tablas que demuestran la hipótesis planteada

ANÁLISIS COMPARATIVO						
	2011	2012	2013	2014	2015	2016
Plataforma Windows Azure	\$1202,28	\$1202,28	\$1202,28	\$1202,28	\$1202,28	\$1202,28
Servidores Convencionales Institucionales	\$4882,38	\$3882,38	\$4164,38	\$4444,38	\$4728,38	\$5010,38

Tabla III.XVIII: Análisis Comparativo de cada año

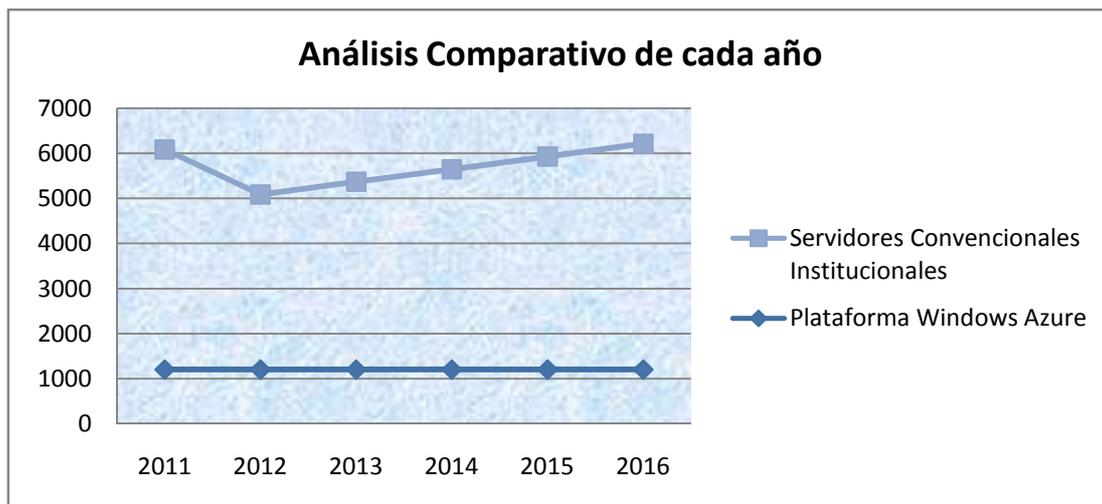


Figura III.43: Comparación de Costos

A continuación se presenta el análisis del Costo total en los cinco años de proyección.

ANÁLISIS COMPARATIVO DE ACUERDO A LA PROYECCIÓN	
Plataforma Windows Azure	\$6133,68
Servidores Convencionales Institucionales	\$27112,28

Tabla III.XIX: Análisis Comparativo de la proyección

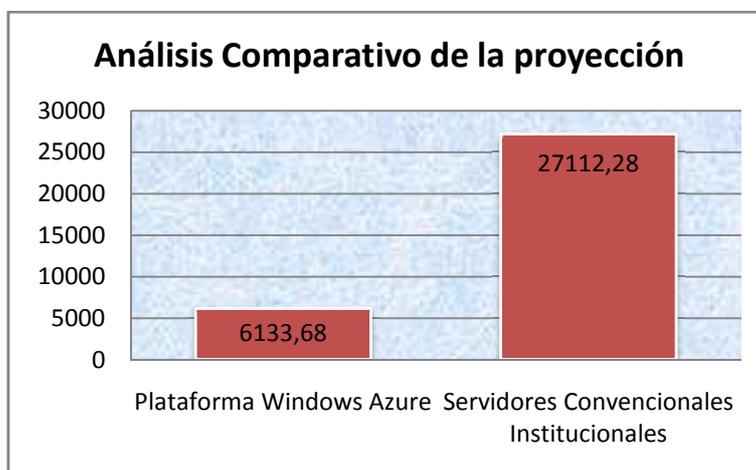


Figura III.44: Comparación de Costos

3.3.1. Conclusiones

Una vez realizado el análisis en cuanto a costo se determina claramente que se comprueba de forma satisfactoria la hipótesis ya que como se observa en la siguiente

figura, desplegar la aplicación en la Plataforma Windows Azure es más económico que desplegar en los servidores institucionales.

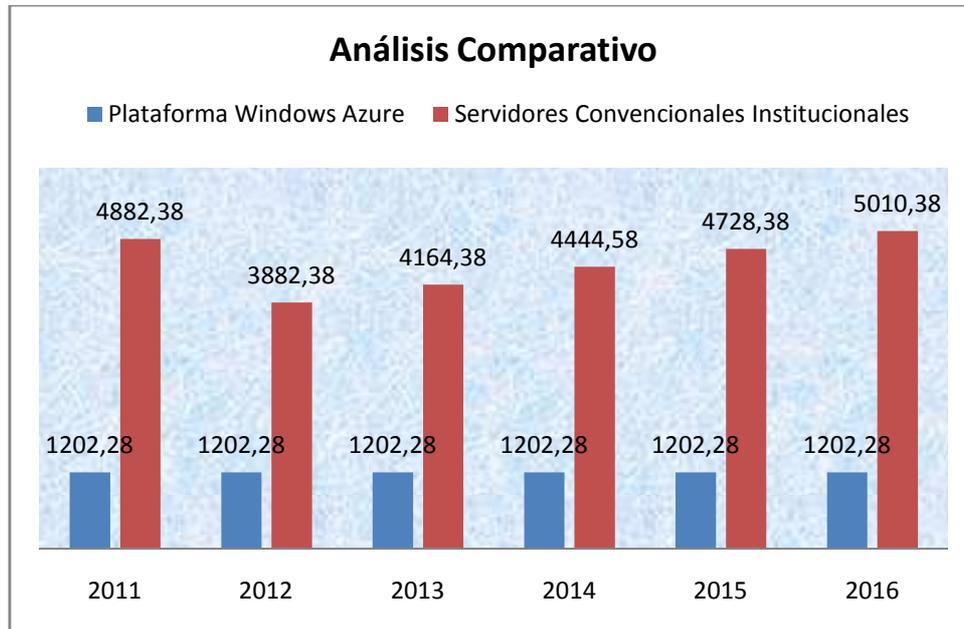


Figura III.45: Conclusiones en costos

3.4. Tiempo de despliegue de una aplicación web en los servidores de la ESPOCH.

3.4.1. Introducción.

Hablar del tiempo es algo muy curioso, pero aprovechar cada minuto es una característica de los tiempos que corren. Disponer de la tecnología adecuada permite realizar varias tareas al mismo tiempo, sobre todo para aquellas personas que lo valoran.

Los datos que se utilizan a continuación fueron proporcionados por el Director del Departamento de Sistemas y Telemática DESITEL Dr. Carlos Buenaño y por el Técnico de los Laboratorios de Ingeniería en Sistemas Ing. Carlos Jara.

3.4.2. Conceptos

Según varios autores dicen: *“El Tiempo es el fuego en el cual nos quemamos”*³⁶, también se puede decir la famosa frase del gran compositor Hector Berlioz. *“El Tiempo es un gran profesor, pero desafortunadamente mata a todos sus pupilos”*.

El tiempo es el recurso más codiciado en todas las actividades cotidianas de los seres humanos, por tal motivo se vuelve muy importante su análisis.

3.4.3. Herramientas utilizadas

Una vez realizadas las encuestas (Ver Anexo 1), con los datos obtenidos se utiliza una de las herramientas muy útiles para contabilizar el tiempo.

La herramienta “Serena OpenProj”, es Open Source. Tiene un tamaño de 6 Mb, se utiliza la versión libre del 2008.



Figura III.46: Serena OpenProj

Serena es una herramienta similar al MS Project que es utilizado para la Gestión de Proyectos, por lo que es fácil el análisis de las actividades y tiempos necesarios para el despliegue de la aplicación.

Serena proporciona diagramas GANTT, Red, y reportes en general.

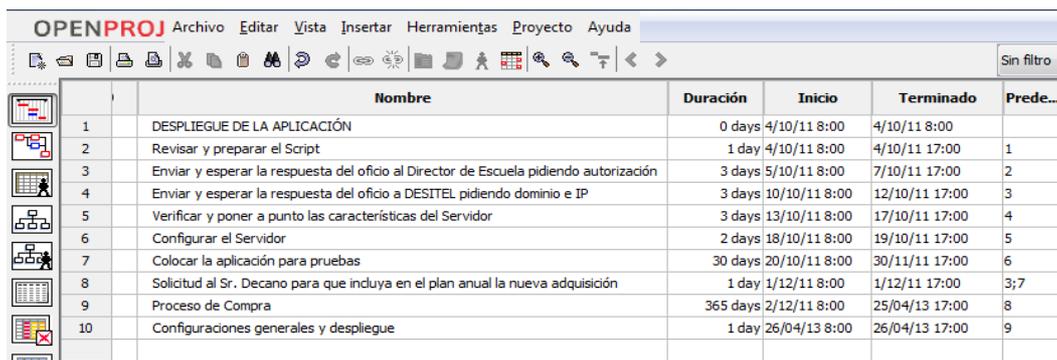
³⁶ Según la película de ciencia ficción StarTrek: Generations

3.4.4. Consideraciones para el análisis.

Como se ve en el Capítulo III, apartado 3.1.4 desplegar una aplicación diseñada en el Sistema Operativo Windows no es posible en los servidores institucionales administrados por DESITEL, razón por lo cual para el análisis se ha tomado la respuesta del Técnico encargado de los laboratorios de Ingeniería en Sistemas.

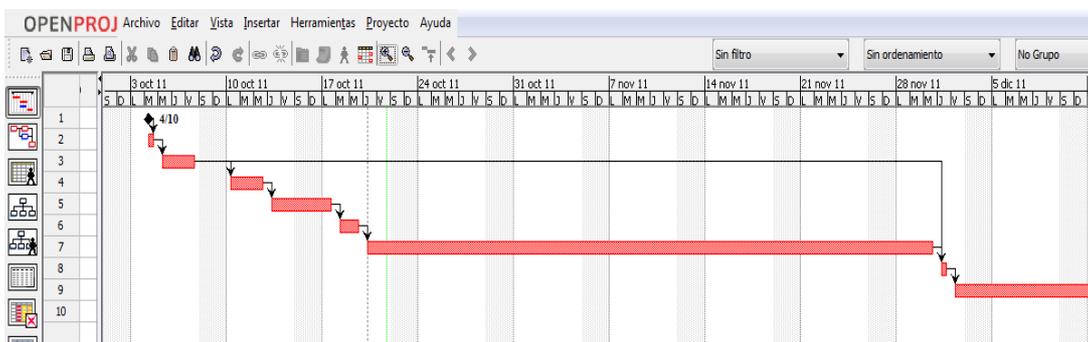
3.4.5. Tiempo necesario en caso de servidores locales.

A continuación se detalla un cronograma de actividades necesarias vs el tiempo a utilizar en las mismas.



	Nombre	Duración	Inicio	Terminado	Prede...
1	DESPLIEGUE DE LA APLICACIÓN	0 days	4/10/11 8:00	4/10/11 8:00	
2	Revisar y preparar el Script	1 day	4/10/11 8:00	4/10/11 17:00	1
3	Enviar y esperar la respuesta del oficio al Director de Escuela pidiendo autorización	3 days	5/10/11 8:00	7/10/11 17:00	2
4	Enviar y esperar la respuesta del oficio a DESITEL pidiendo dominio e IP	3 days	10/10/11 8:00	12/10/11 17:00	3
5	Verificar y poner a punto las características del Servidor	3 days	13/10/11 8:00	17/10/11 17:00	4
6	Configurar el Servidor	2 days	18/10/11 8:00	19/10/11 17:00	5
7	Colocar la aplicación para pruebas	30 days	20/10/11 8:00	30/11/11 17:00	6
8	Solicitud al Sr. Decano para que incluya en el plan anual la nueva adquisición	1 day	1/12/11 8:00	1/12/11 17:00	3;7
9	Proceso de Compra	365 days	2/12/11 8:00	25/04/13 17:00	8
10	Configuraciones generales y despliegue	1 day	26/04/13 8:00	26/04/13 17:00	9

Figura III.47: Diagrama Gantt (Parte I Servidor local)



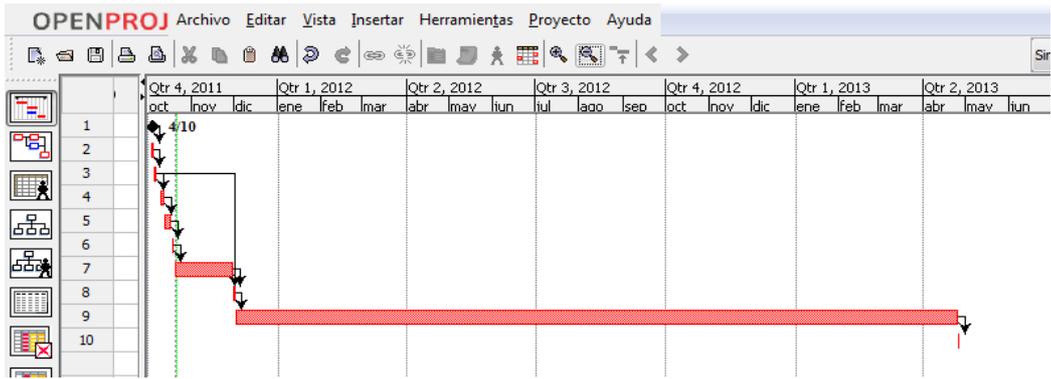


Figura III.48: Diagramas Gantt (Parte II Servidor Local)

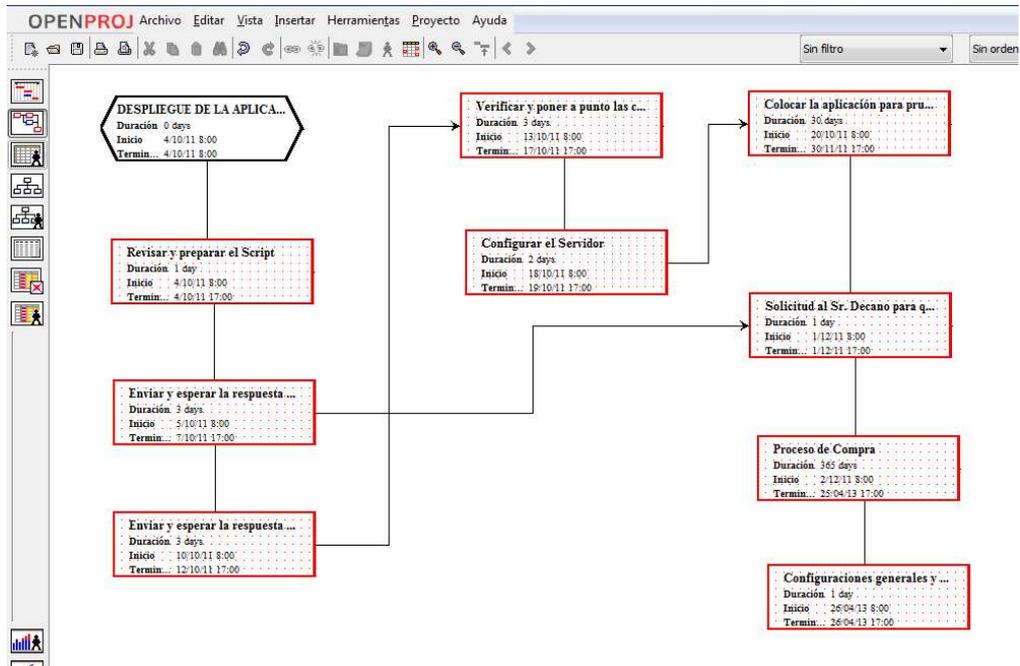


Figura III.49: Diagrama de Red Servidor local

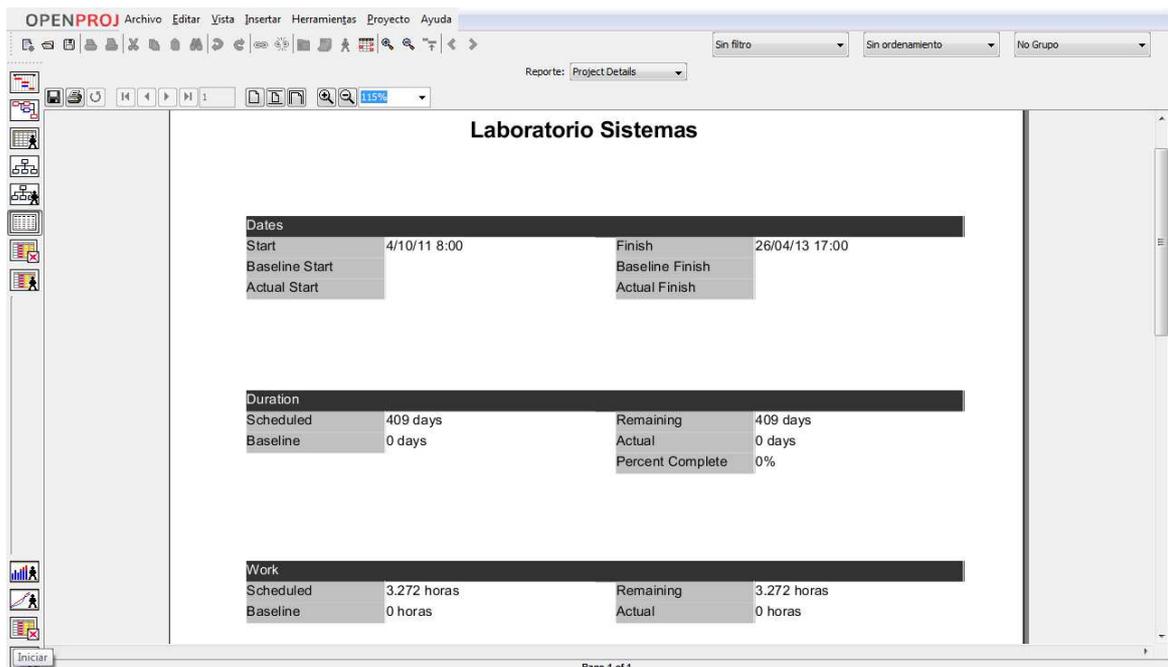


Figura III.50: Informe General del análisis Servidor Local

Como se observa en las imágenes anteriores da como resultado.

Tiempo utilizado para el despliegue en servidores locales:

TIEMPO	VALOR
Fecha de Inicio:	04-10-2011
Fecha de Finalización:	26-04-2013
Cantidad de días empleados en el Despliegue:	409 días
Horas trabajadas o de espera:	3272 horas.

Tabla III.XX: Tiempo de despliegue en servidores locales

Como se observa en la tabla anterior la cantidad de horas trabajadas o de espera es de 3272 se obtiene este valor porque la mayoría de este tiempo empleado en el despliegue se debe a los trámites burocráticos que se tiene que realizar.

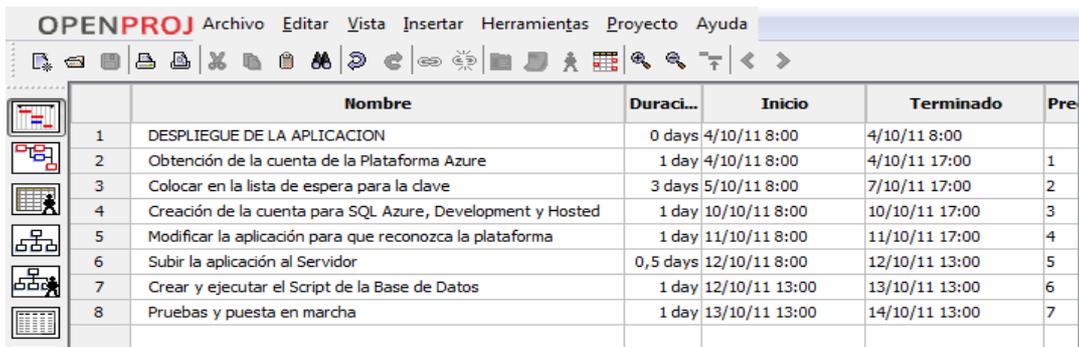
3.5. Tiempo de despliegue de una aplicación web en la plataforma de Windows Azure

3.5.1. Introducción

Como se mencionó anteriormente el tiempo es lo más importante y por lo tanto se necesita analizarlo de una forma correcta.

3.5.2. Tiempo necesario en caso de servidores de la Plataforma de Windows Azure

Cronograma de actividades necesarias vs al tiempo a utilizar.



	Nombre	Duraci...	Inicio	Terminado	Pre
1	DESPLIEGUE DE LA APLICACION	0 days	4/10/11 8:00	4/10/11 8:00	
2	Obtención de la cuenta de la Plataforma Azure	1 day	4/10/11 8:00	4/10/11 17:00	1
3	Colocar en la lista de espera para la clave	3 days	5/10/11 8:00	7/10/11 17:00	2
4	Creación de la cuenta para SQL Azure, Development y Hosted	1 day	10/10/11 8:00	10/10/11 17:00	3
5	Modificar la aplicación para que reconozca la plataforma	1 day	11/10/11 8:00	11/10/11 17:00	4
6	Subir la aplicación al Servidor	0,5 days	12/10/11 8:00	12/10/11 13:00	5
7	Crear y ejecutar el Script de la Base de Datos	1 day	12/10/11 13:00	13/10/11 13:00	6
8	Pruebas y puesta en marcha	1 day	13/10/11 13:00	14/10/11 13:00	7

Figura III.51: Diagrama Gantt (Parte I Azure)

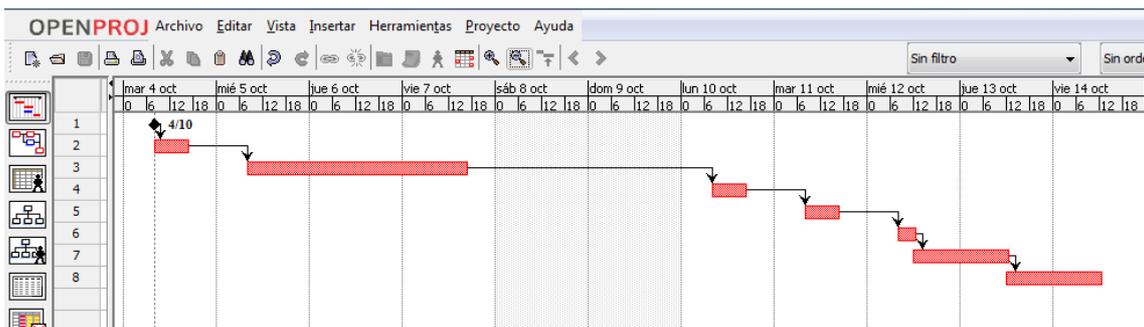


Figura II.52: Diagrama Gantt (Parte II Azure)

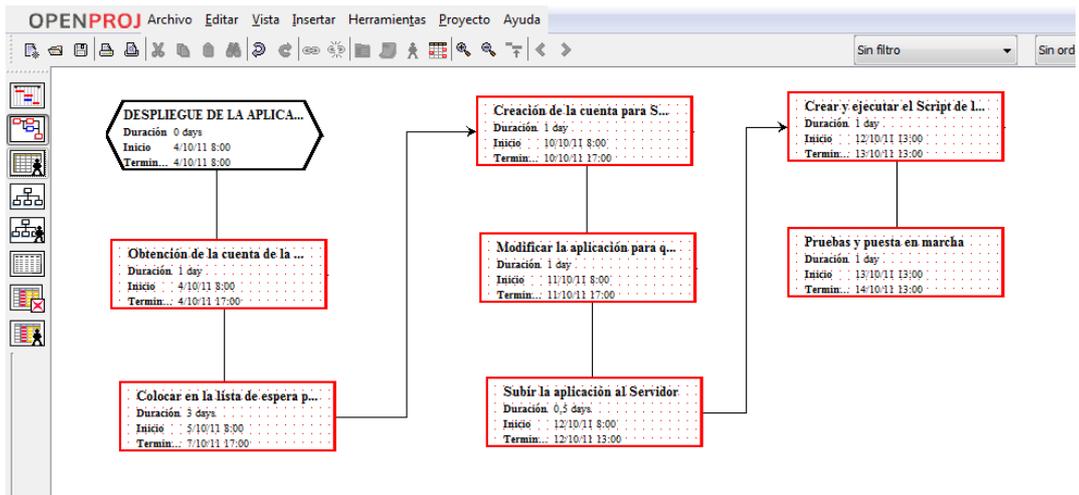


Figura III.53: Diagrama de Red Azure

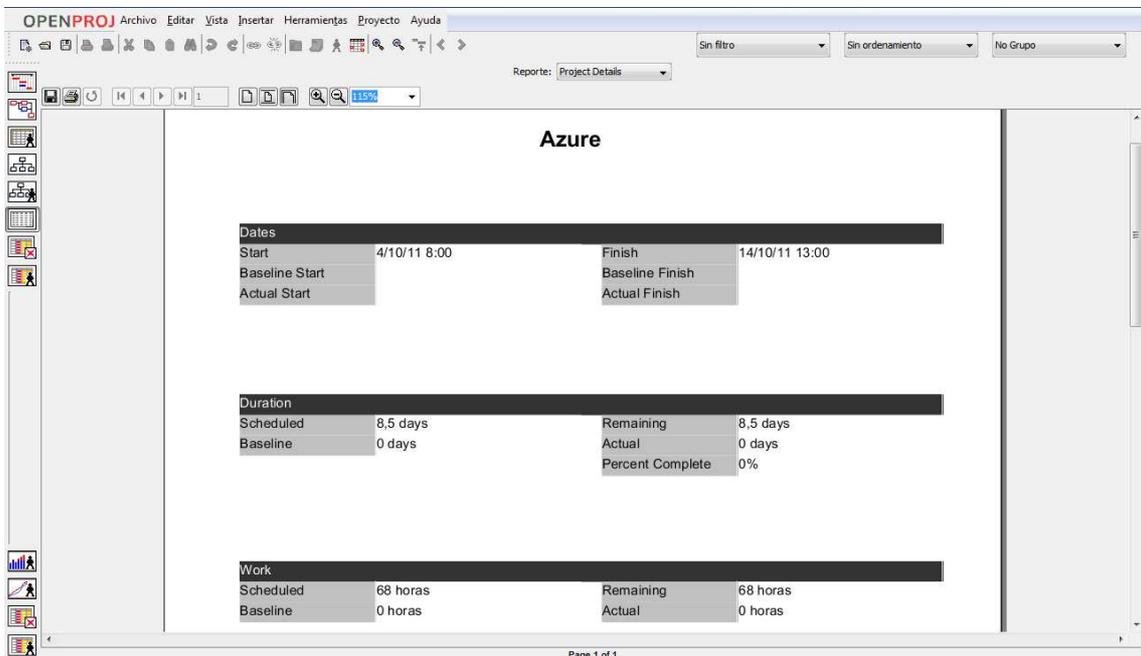


Figura III.54: Informe General del análisis Azure

Como se observa en las imágenes anteriores da como resultado

Tiempo utilizado para el despliegue en servidores de la plataforma de Windows Azure:

TIEMPO	VALOR
Fecha de Inicio:	04-10-2011
Fecha de Finalización:	14-10-2011
Cantidad de días empleados en el	8,5 días

Despliegue:	
Horas trabajadas o de espera:	68 horas.

Tabla III.XXI: Tiempo de despliegue en servidores de Azure

3.6. Estudio comparativo.

TIEMPO	SERVIDORES CONVENCIONALES INSTITUCIONALES	PLATAFORMA WINDOWS AZURE.
Fecha de Inicio:	04-10-2011	04-10-2011
Fecha de Finalización:	26-04-2013	14-10-2011
Cantidad de días empleadas en el Despliegue:	409 días	8,5 días
Horas trabajadas o de espera:	3272 horas.	68 horas.

Tabla III.XXII: Análisis Comparativo

A continuación descripción de los parámetros analizados:

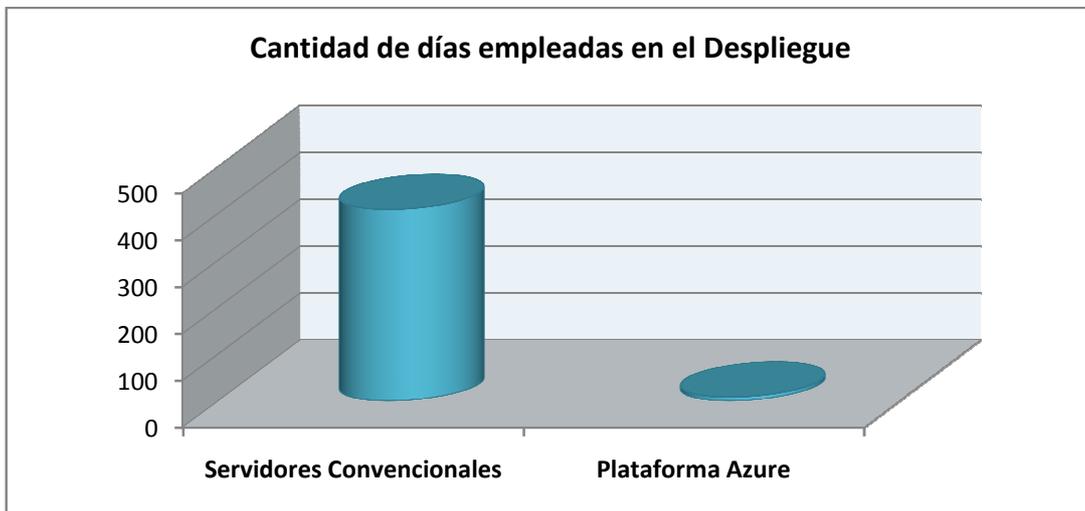


Figura III.55: Días empleados en el despliegue

3.6.1. Conclusiones

Una vez realizado el análisis en cuanto al tiempo en el despliegue de la aplicación se determina claramente que se comprueba de forma satisfactoria la hipótesis ya que como se observó en las figuras anteriores, desplegar la aplicación en la Plataforma Windows Azure se realiza en un tiempo mucho menor que desplegar en los servidores institucionales.

CAPÍTULO IV

4. Desarrollo de la aplicación usando Windows Azure

4.1. Instalación y configuración de Visual Studio 2010 como IDE.

4.1.1. Introducción

La instalación de Microsoft Visual Studio 2010 Professional es algo en lo que se ha puesto un marcado énfasis para producir un sistema claro y sencillo que no lleve al éxito con un mínimo esfuerzo. Por otro lado, durante el proceso, quizás sea bueno conocer algunos aspectos concretos del mismo para producir a la primera una instalación limpia, correcta y alineada con la arquitectura del entorno.

Esta guía de instalación está centrada en el proceso de instalación del IDE de desarrollo de Microsoft de una forma gráfica y rápida.

4.1.2. Proceso de Instalación

Los pasos siguientes muestran claramente la instalación del IDE sobre el Sistema Operativo Windows 7:

El primer paso es buscar en línea el link de instalación, cabe recalcar que Visual Studio 2010 Professional se instala directamente, ya que como es un nuevo servicio, y por lo tanto se necesita estar conectado directamente al Internet y tener un enlace activo.



Figura IV.56: Inicio de la instalación de Visual Studio 2010

A continuación se debe salir de las otras aplicaciones para que no cause problemas al momento de la instalación y aceptar los términos de la licencia.

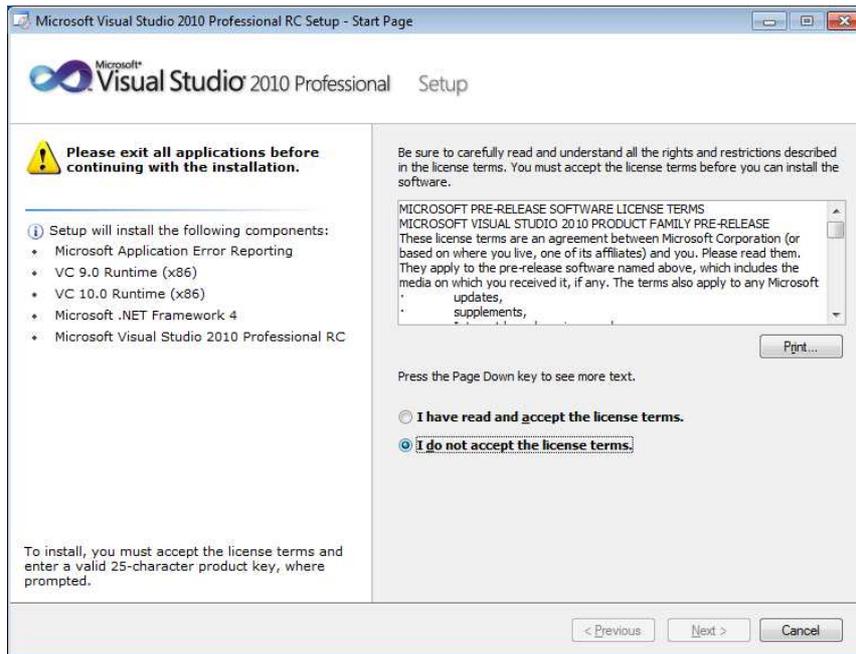


Figura IV.57: Aceptar los terminos de la licencia para continuar

Como todas las aplicaciones de la familia de Microsoft son relativamente fáciles su instalación no se detalla con mayor descripción.

La instalación por motivos didácticos se realizó el tipo **full**, pero al momento de poner en producción se debería realizarlo de manera **custom** para seleccionar los paquetes necesarios de acuerdo a las necesidades.

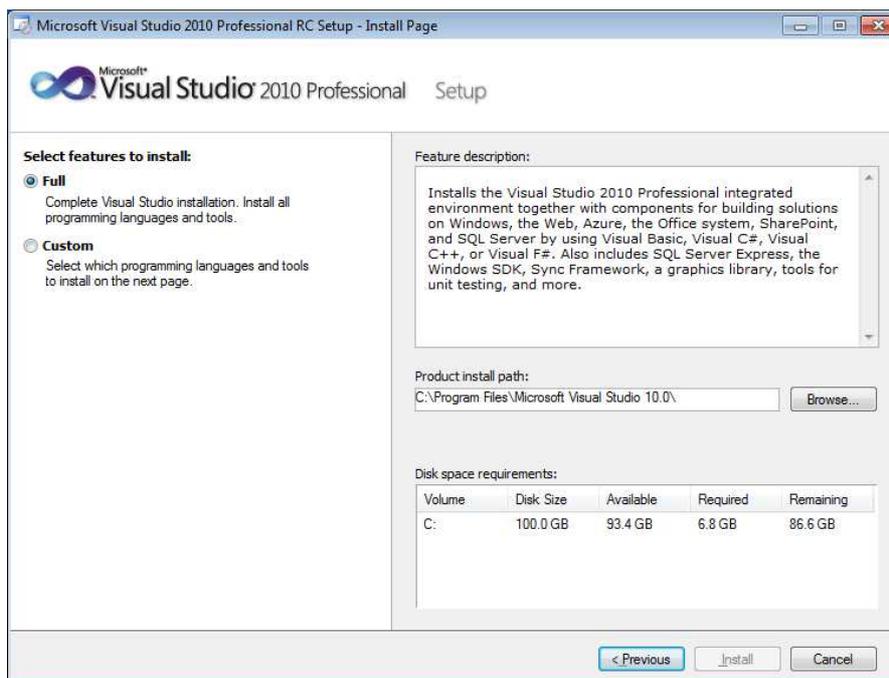


Figura IV.58: Seleccionar la dirección donde se va a instalar

Ahora simplemente se tiene que esperar que se termine la instalación de todos los paquetes seleccionados.

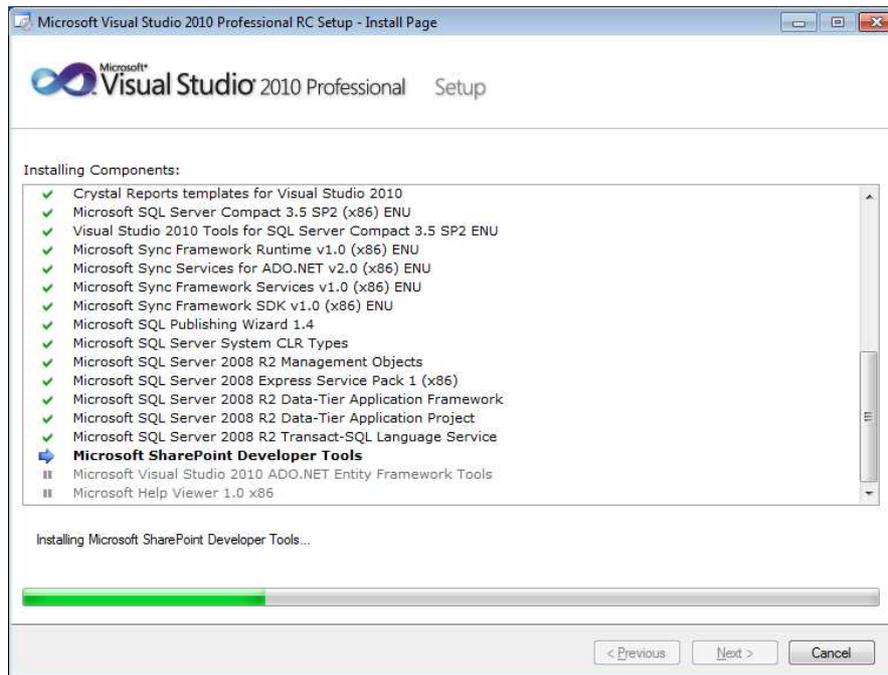


Figura IV.59: Instalación de cada uno de los componentes seleccionados

Terminada la instalación, dar clic en finish y reiniciar el sistema. Por lo tanto el IDE de desarrollo ya se encuentra listo en el Sistema Operativo.

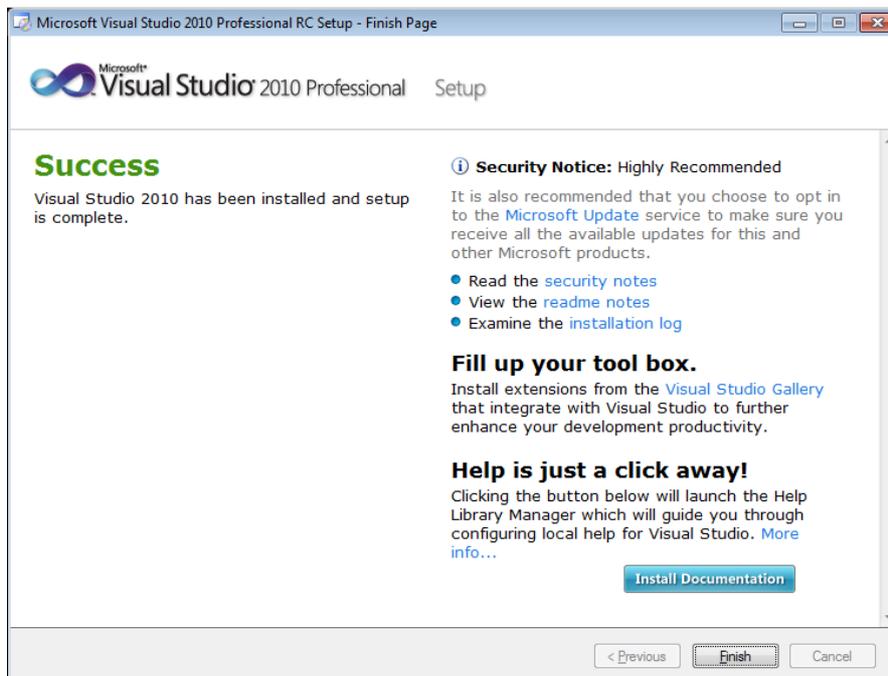


Figura IV.60: Instalación terminada exitosamente



Figura IV.61: Inicio de Visual Studio 2010

Una vez terminado el proceso de instalación, ya se puede comenzar a trabajar con Visual Studio 2010 Professional inmediatamente, tomando en cuenta que **será necesario reiniciar** el Sistema Operativo.

4.2. Instalación y configuración de SQL Server 2008 R2

4.2.1. Introducción

La instalación de SQL Server 2008 R2 es algo en lo que se ha puesto un marcado énfasis para producir un sistema claro y sencillo que no lleve al éxito con un mínimo esfuerzo. Por otro lado, durante el proceso, quizás sea bueno conocer algunos aspectos concretos del mismo para producir a la primera una instalación limpia, correcta y alineada con la arquitectura del entorno.

Esta guía de instalación está centrada en el proceso de instalación de nuevas instancias del motor relacional de SQL Server 2008 R2 de una forma gráfica y rápida.

4.2.2. Proceso de Instalación

Los pasos son para instalar una nueva instancia de SQL Server 2008 R2 sobre el Sistema Operativo:

El primer paso es introducir el DVD o montar la imagen .iso de SQL Server 2008 R2.

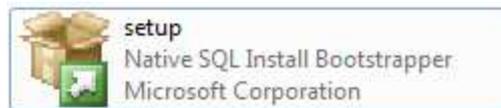


Figura IV.62: Icono para instalar SQL Server 2008 R2

Se tendrá que esperar unos cuantos segundos y aparecerá la siguiente ventana.

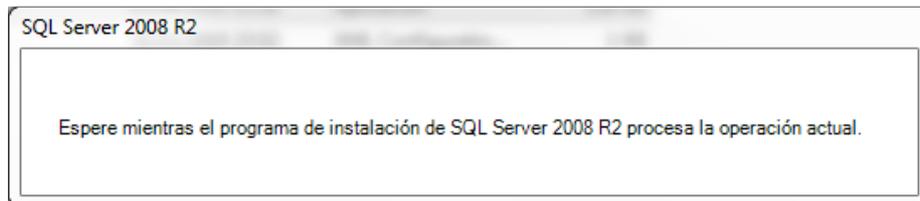


Figura IV.63: Inicio del proceso de instalación

En este punto, y directamente dar clic en el apartado **“Instalación”**.

En cualquier caso, en la primera pantalla denominada **“Planeamiento”**, se podrá encontrar la documentación sobre instalación de SQL Server 2008 R2 así como la aplicación de chequeo de sistema para validar instalaciones futuras de SQL Server 2008 R2 o el SQL Server UpgradeAdvisor con la que se puede analizar migraciones de versiones inferiores hacia SQL Server 2008 R2.



Figura IV.64: Seleccionar la opción nueva instalacion

Una vez en la sección “Installation” proceder al apartado “**New installation or add features to an existing installation**” con la que se instalará una nueva instancia de SQL Server 2008 R2 a nuestro equipo.

Una vez hecho clic en dicha opción proceder con la instalación siguiendo los pasos de las imágenes que se pueden ver a continuación.

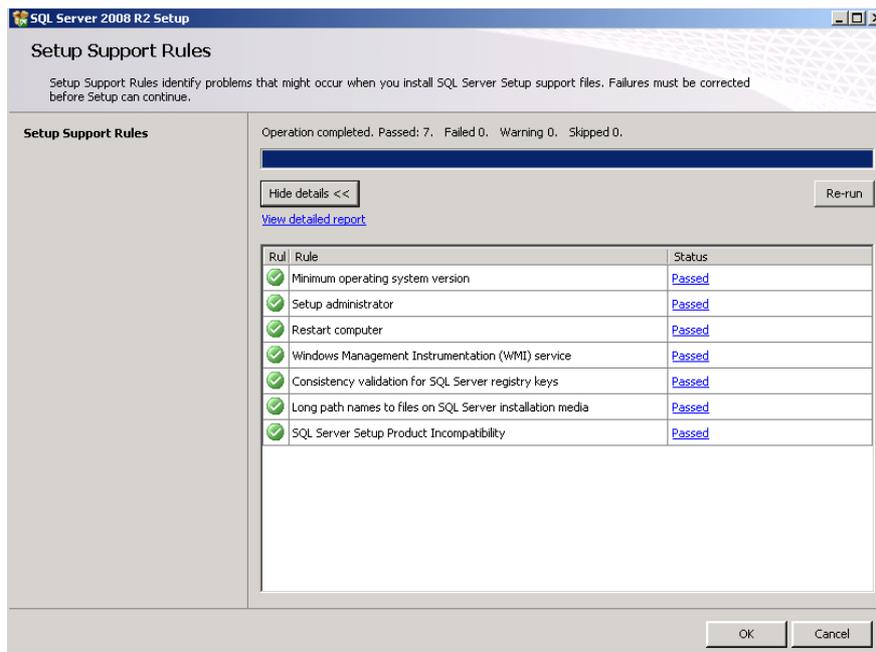


Figura IV.65: Verificación de las reglas de soporte

Antes de empezar el instalador realiza unos chequeos de integridad del sistema para comprobar que está todo listo para la instalación. Si por ejemplo estuviera el sistema esperando para ser reiniciado por cualquier otra instalación previa anterior, no dejaría continuar.

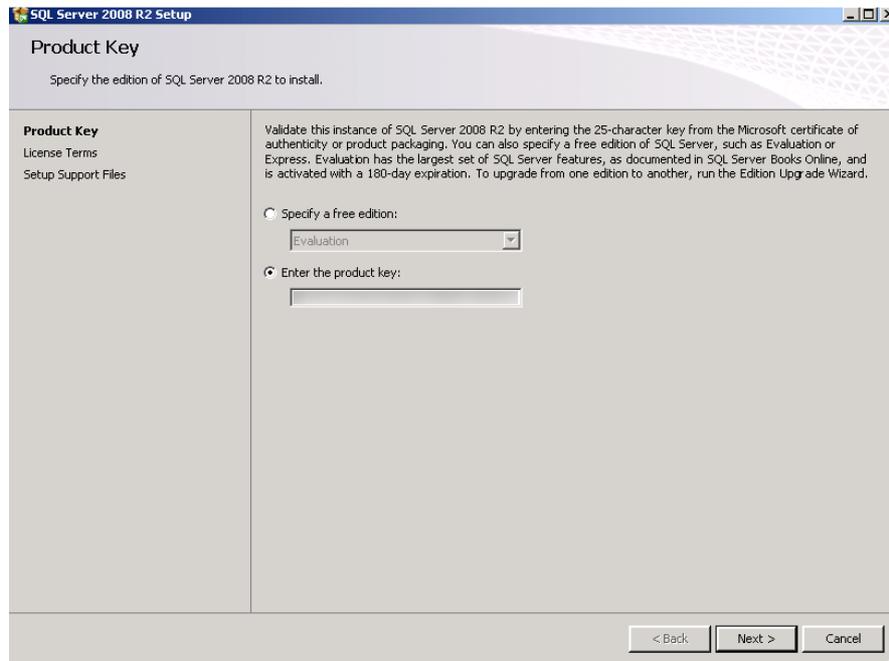


Figura IV.66: Ingresar la clave del producto

Antes de comenzar, es mejor tener el código de licencia preparado (en el caso para la tesis se instaló en modo evaluación).

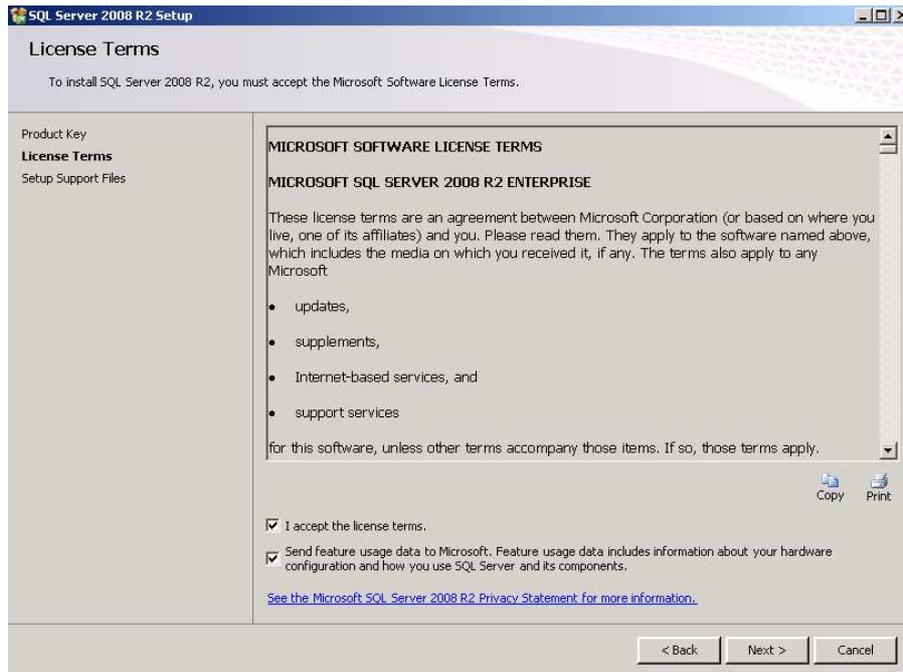


Figura IV.67: Aceptar los terminos de la licencia

Una vez validado el código de licencia de SQL Server 2008 R2 se presenta los términos de la licencia que se tiene que aceptar.

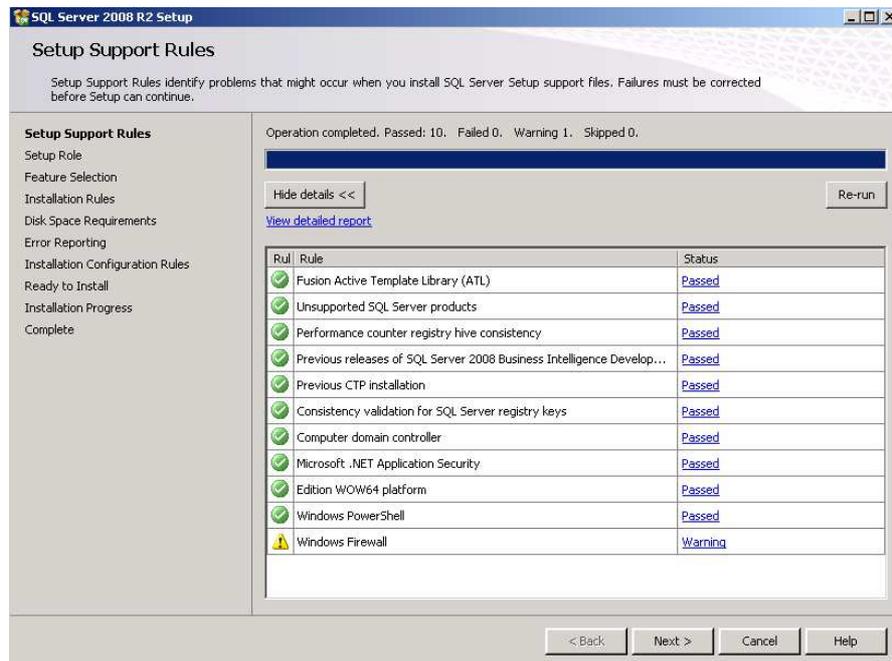


Figura IV.68: Aceptar los soportes de las reglas

Una vez aceptados los términos, en la Figura 7 se observa cómo se realizan de nuevo una serie de chequeos previos a la instalación, que hay que cumplir antes de proseguir.

En este caso, comentar únicamente que siempre que se tenga activo el firewall de Windows, el asistente dará un Warning el apartado “*Windows Firewall*”.

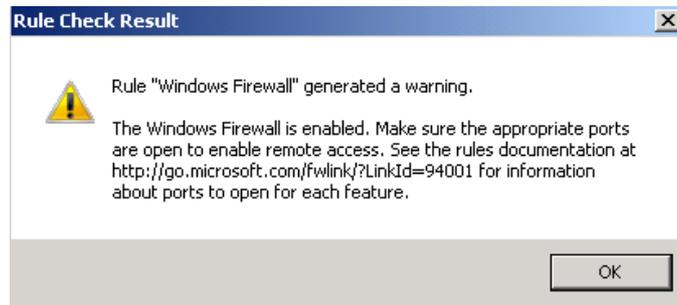


Figura IV.69: Advertencia de Firewall

Esto no impedirá la instalación, simplemente se está informando que tengamos cuidado de habilitar las reglas en el firewall de Windows necesarias para permitir la conexión de clientes remotos hacia el servicio de SQL Server 2008 R2.

Entre otras cosas, conviene recordar que a menos que sea modificado (cosa que es recomendable como buena práctica) el puerto por defecto que escucha SQL Server 2008 R2 es el 1433.

Una vez hecho clic sobre el botón siguiente, pedirá qué Rol vamos a instalar de SQL Server 2008 R2.

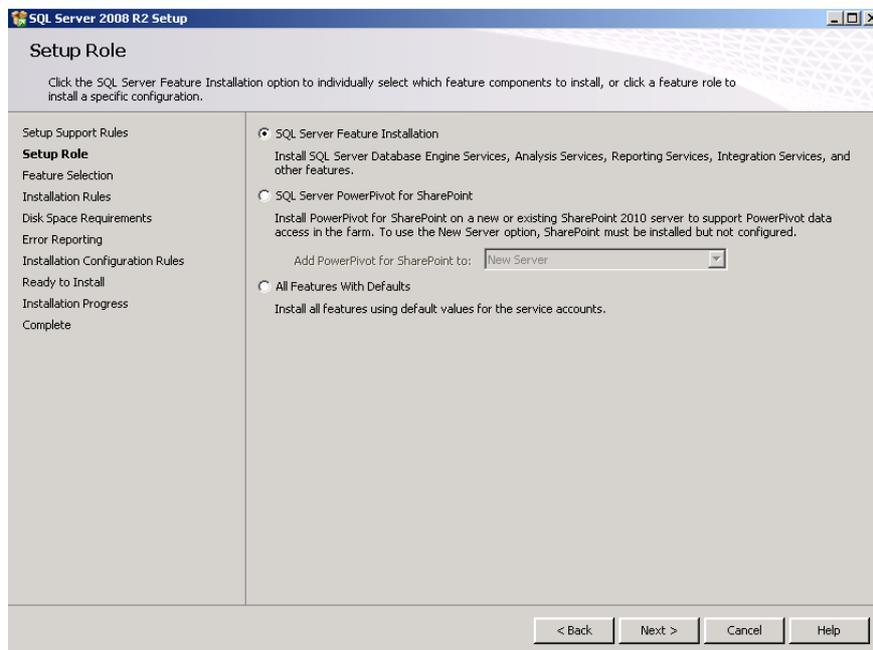


Figura IV.70: Instalación del Rol

Como es una instalación de instancia única de SQL Server 2008 R2, seleccionar la primera de las opciones que salen (SQL Server FeatureInstallation).

Hecho esto, la siguiente pantalla, va a requerir qué servicios y características de SQL Server 2008 R2 se va a instalar.

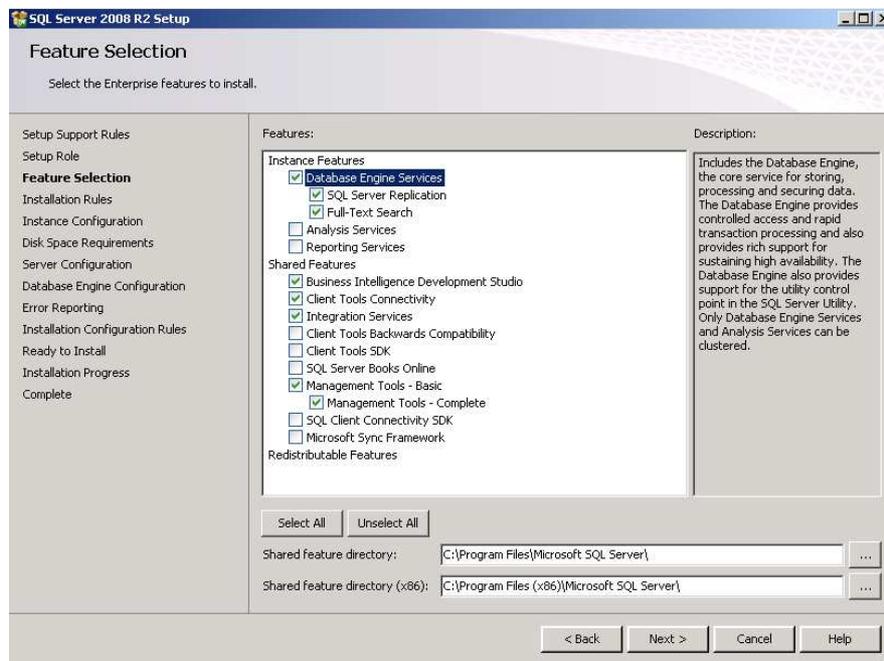


Figura IV.71: Selección de Servicios y Características

Esta es una instalación de motor relacional de SQL Server 2008 R2, en la imagen se pueden observar qué características y servicios se recomienda instalar, pero por motivos didácticos para la tesis se ha instalado todas las características.

Una vez seleccionados los componentes deseados, se realizará un nuevo chequeo enfocado en este caso a detectar posibles incompatibilidades o inconsistencias a la hora de instalar los componentes y características que se han seleccionado especialmente.

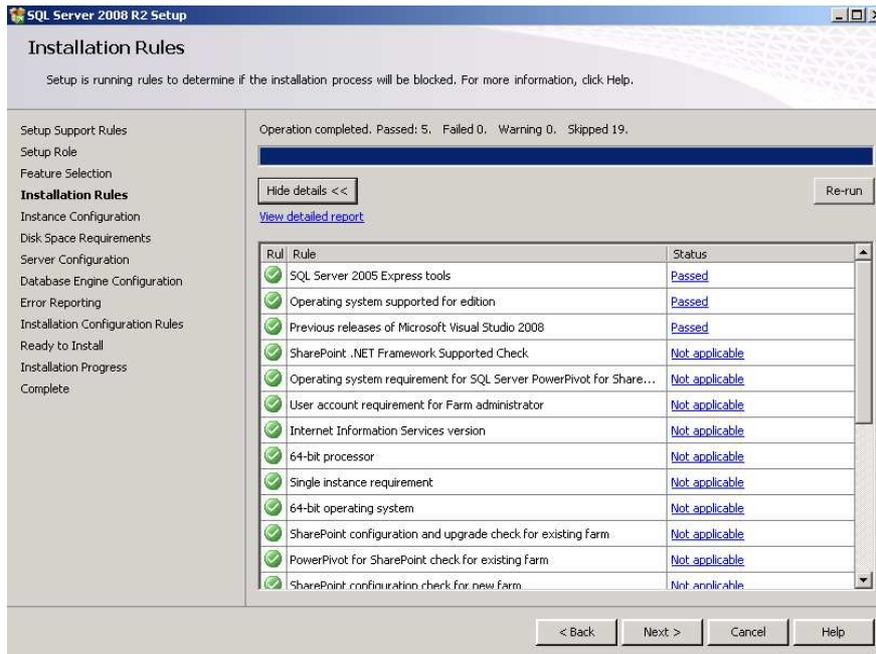


Figura IV.72: Verificar las reglas de instalacion

En la siguiente imagen, se indica tanto la ruta donde desplegará los binarios de SQL Server, como el nombre de la instancia (en el caso de que queramos).

El dar un nombre a la instancia o no, depende de criterio personal. Por tanto, si ya se tiene una instancia con el nombre por defecto, la única opción que se debe escoger es la de NamedInstance y asignar un nombre.

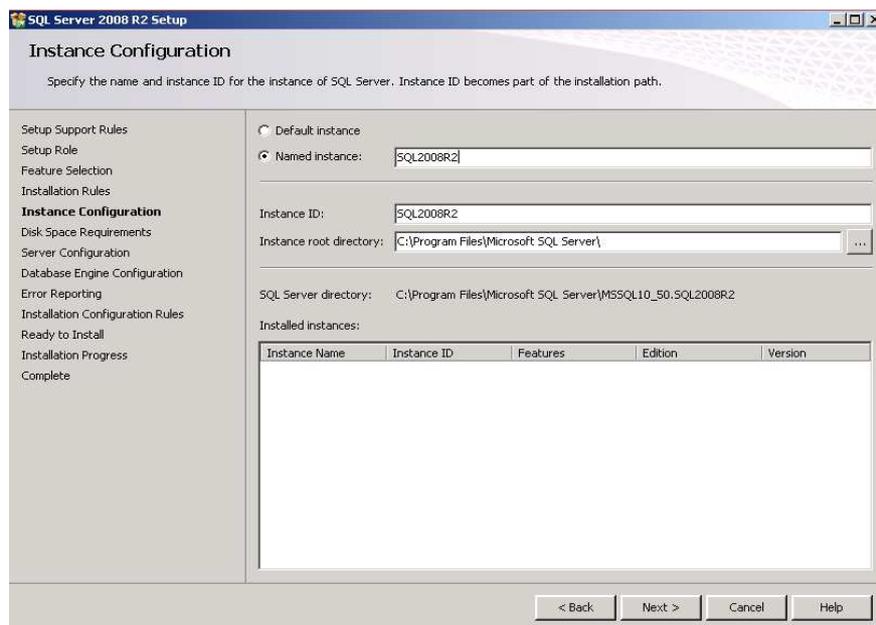


Figura IV.73: Llenar los campos requeridos

En este caso, el nombre de la instancia es “Sistemas”. Puesto que es posible instalar más de una única instancia de SQL Server en una misma máquina, si existieran otras instancias en la máquina, en el DataGrid que aparece en el centro de la figura lo podría detectar para evitar dar un nombre que ya estuviera asignado.

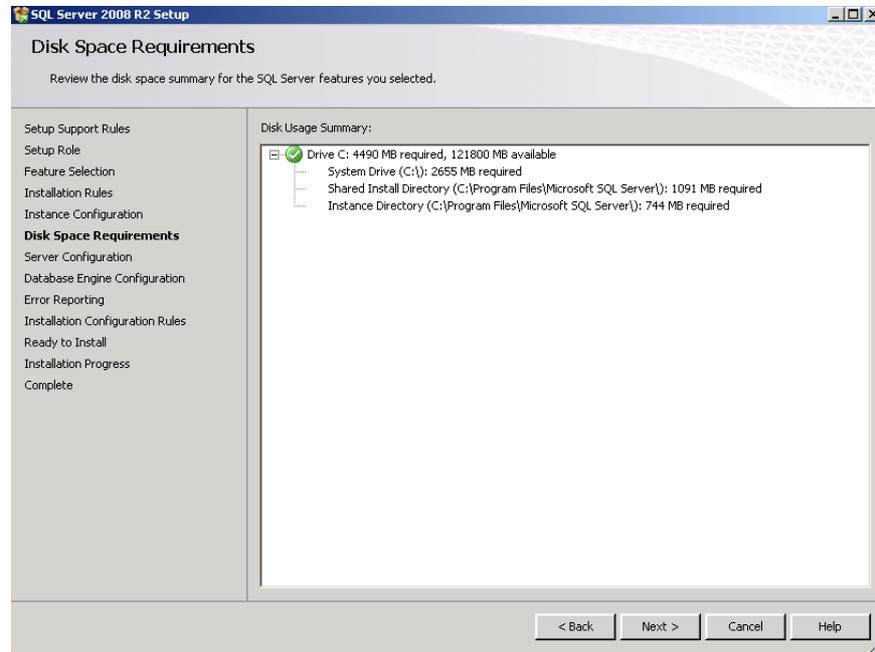


Figura IV.74: Verifica que las instancias no se repitan

Se aprecia un breve resumen relativo al espacio en disco requerido y la ubicación raíz de instalación, pero esto no quiere decir que ya se haya terminado de configurar la instalación.

Si no cumple con el espacio de disco mínimo requerido no se puede continuar con la instalación.

Como se ve en la figura, ahora toca el turno de configurar las cuentas del servicio y el collation.

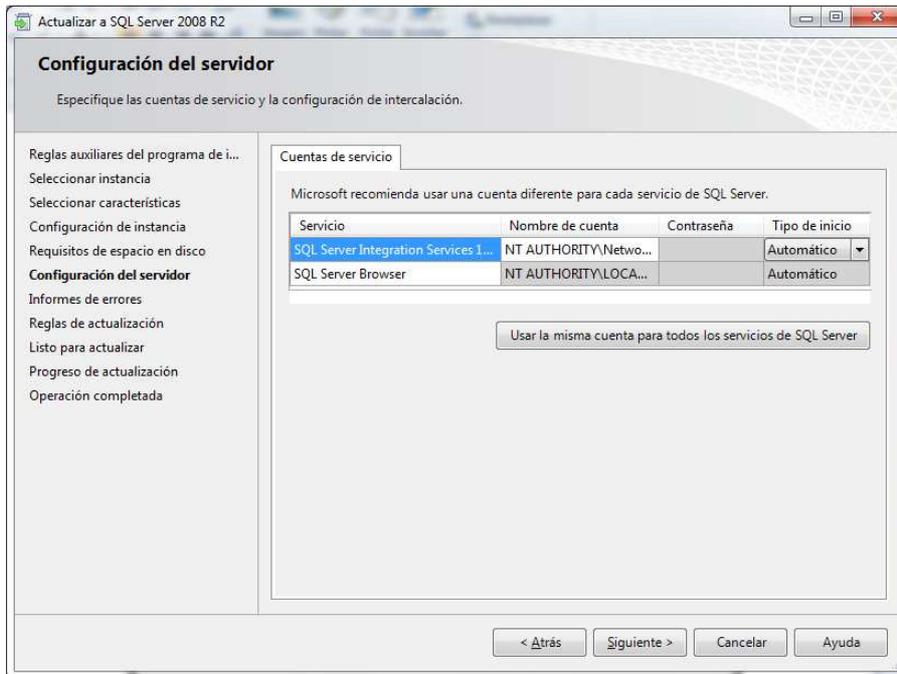


Figura IV.75: Configuración del servidor

Se aprecia que pide introducir una cuenta de usuario para arrancar todos y cada uno de los servicios que hayamos requerido instalar.

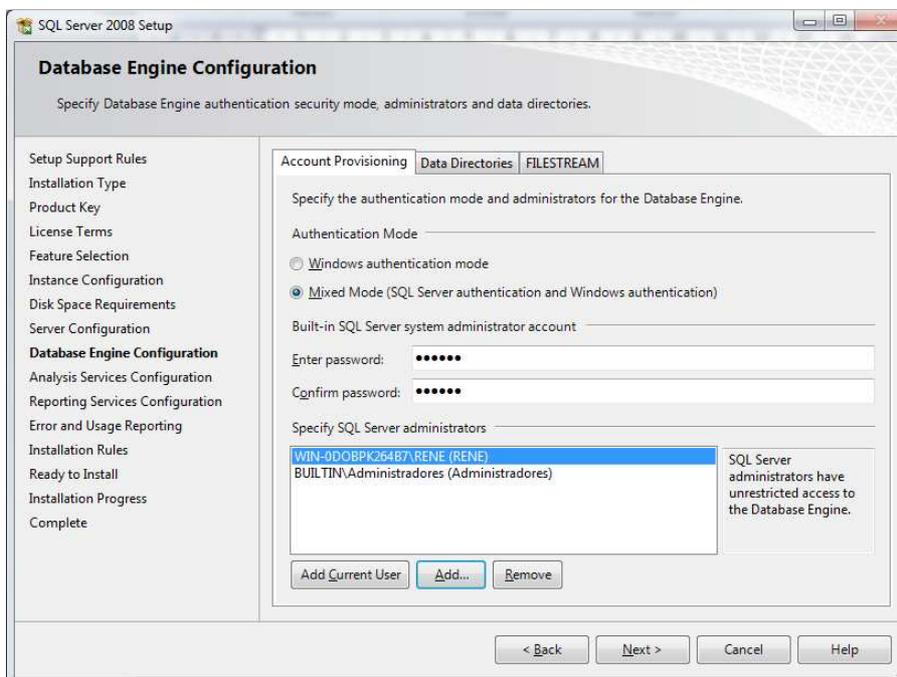


Figura IV.76: Configuración de la Base de Datos

En la imagen se ve el modo de autenticación seleccionar “MixedMode (SQL Server authentication and Windows authentication)”

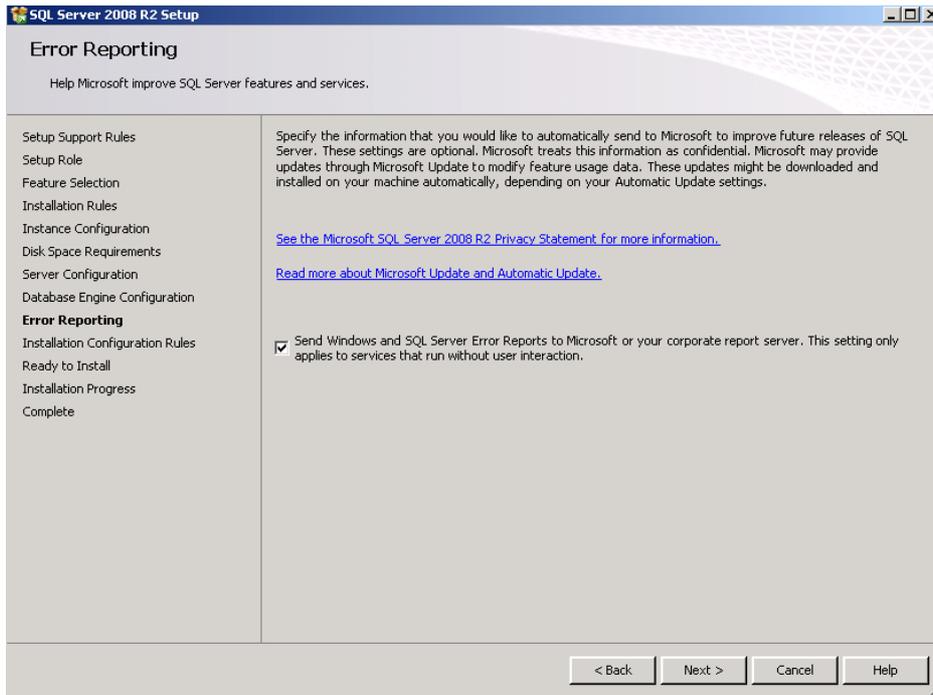


Figura IV.77: Confirmación de envío de errores

Microsoft pregunta si se desea que se envíen errores de reporte para los servicios que no poseen interacción humana. Esto que es obviamente de libre elección, recordar que ayuda a mejorar el soporte que brinda Microsoft y a corregir posibles bugs de producto.

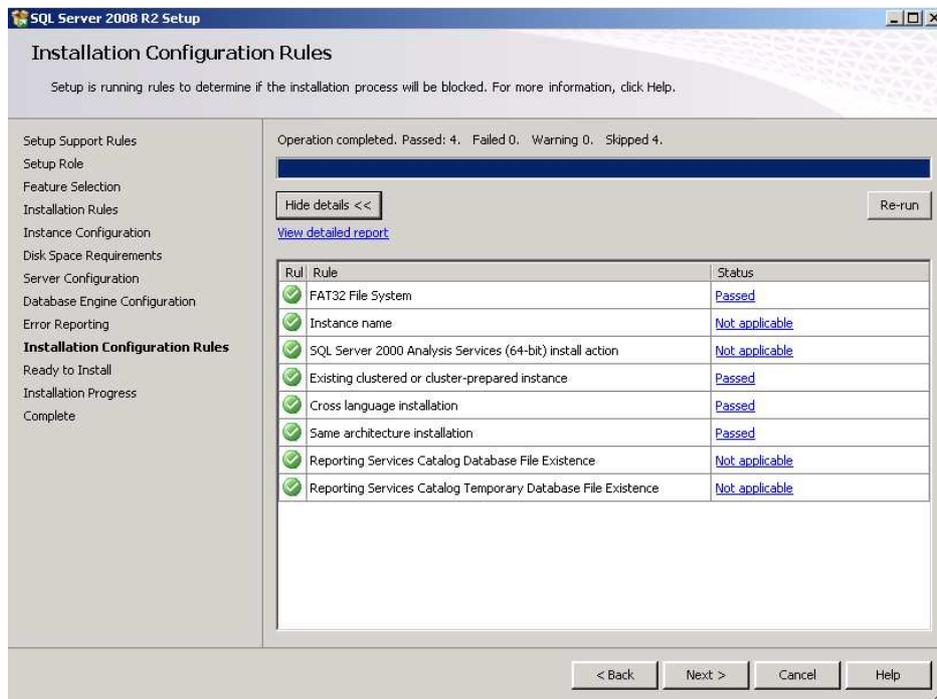


Figura IV.78: Verificación de las reglas

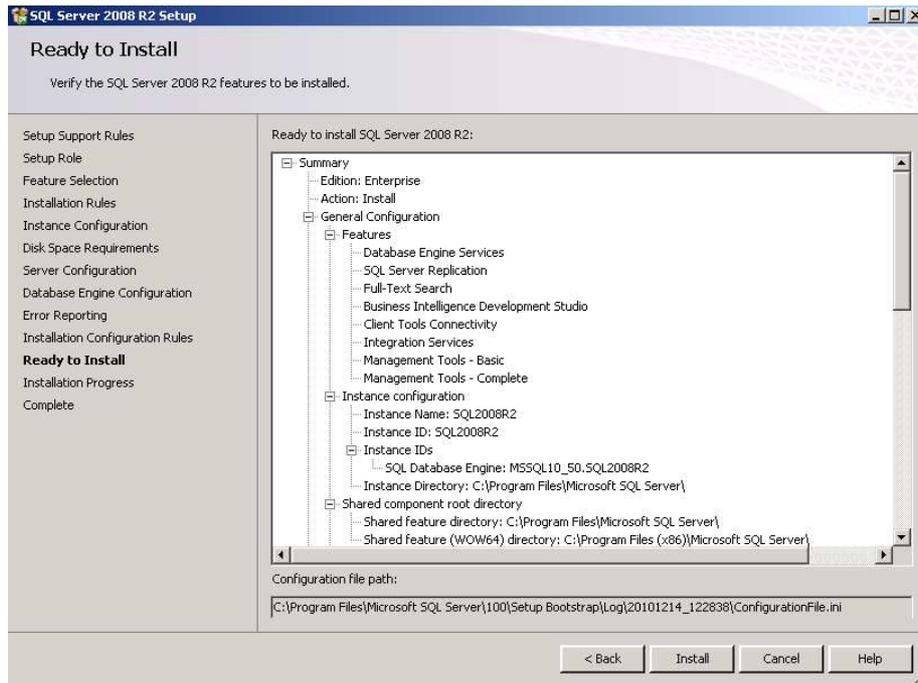


Figura IV.79: Lectura de la Instalación

Resumen de las características y servicios que se van a instalar y donde, y será el último paso a realizar hasta la instalación de SQL Server 2008 R2

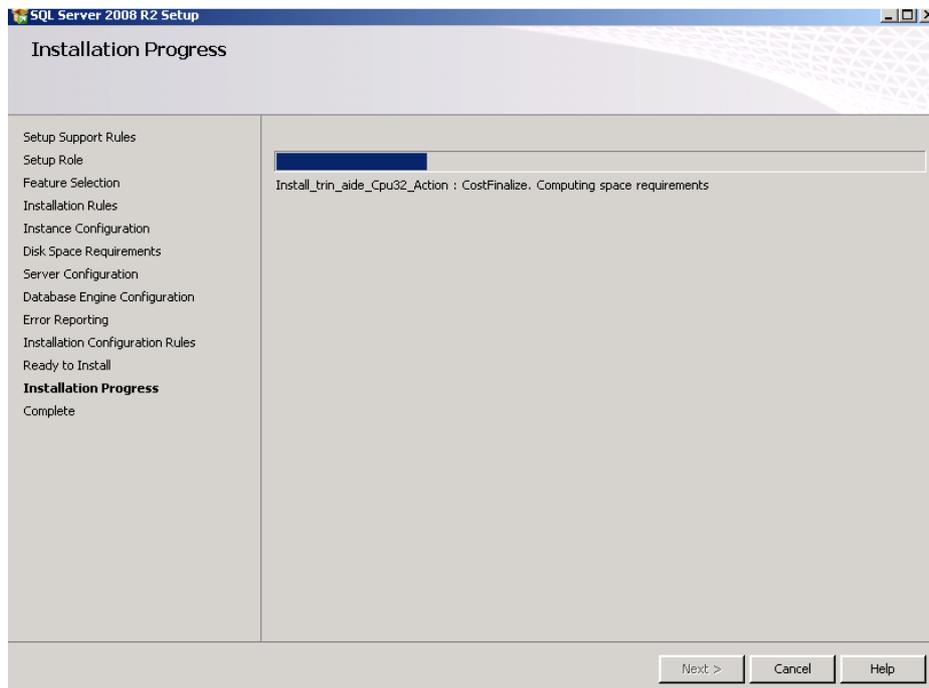


Figura IV.80: Progreso de Instalación

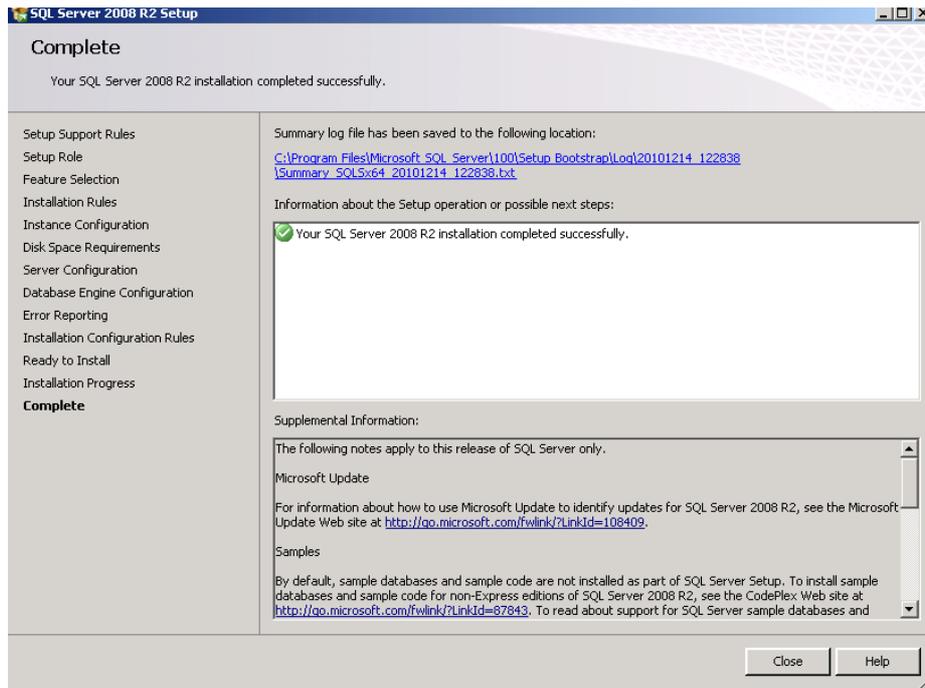


Figura IV.81: Instalación Finalizada

Una vez terminado el proceso de instalación, ya se puede comenzar a trabajar con SQL Server 2008 R2 inmediatamente, puesto que **no será necesario reinicio** alguno.

4.3. Instalación y configuración de las herramientas propias de Windows Azure

4.3.1. Introducción

La instalación de las Windows Azure Tools para Microsoft Visual Studio 1.3 – VsCloudService.exe O el SDK de Windows Azure: WindowsAzureSDK-x86.exe o WindowsAzureSDK-x64.exe en el Sistema Operativo Windows 7 es algo en lo que se ha puesto un marcado énfasis para producir un sistema claro y sencillo que nos lleve al éxito con un mínimo esfuerzo.

Por otro lado, durante el proceso quizás sea bueno conocer algunos aspectos concretos del mismo para producir a la primera una instalación limpia, correcta y alineada a la arquitectura del entorno.

4.3.2. Proceso de instalación

Los pasos para instalar VsCloudService.exe:

Se va a instalar VsCloudService.exe sobre el Sistema Operativo Windows 7 que se encuentra sobre VMWare.

Para la presente investigación se ha instalado VsCloudService.exe por lo que dicho instalador incluye el SDK de Windows Azure para Visual Studio 2010, es muy importante recalcar que para obtener un óptimo rendimiento se recomienda descargar de la página oficial la versión más reciente, ya que es un software que está en constante cambio por su etapa de pruebas.

Todos los productos que se deben instalar para que la máquina se comporte y sea compatible con la Plataforma Windows Azure son relativamente fáciles, basta con ingresar a la página oficial de Microsoft y descargarlos para su correcto funcionamiento

En primer lugar se debe descargar el VsCloudService.exe, una vez terminado se mostrará el siguiente icono.



Figura IV.82: Icono del VSCloudService

A continuación iniciar con el Wizard que será el encargado de guiar toda la instalación.

La siguiente imagen muestra el inicio de la instalación e informa los productos que van a ser actualizados o instalados dependiendo el caso.

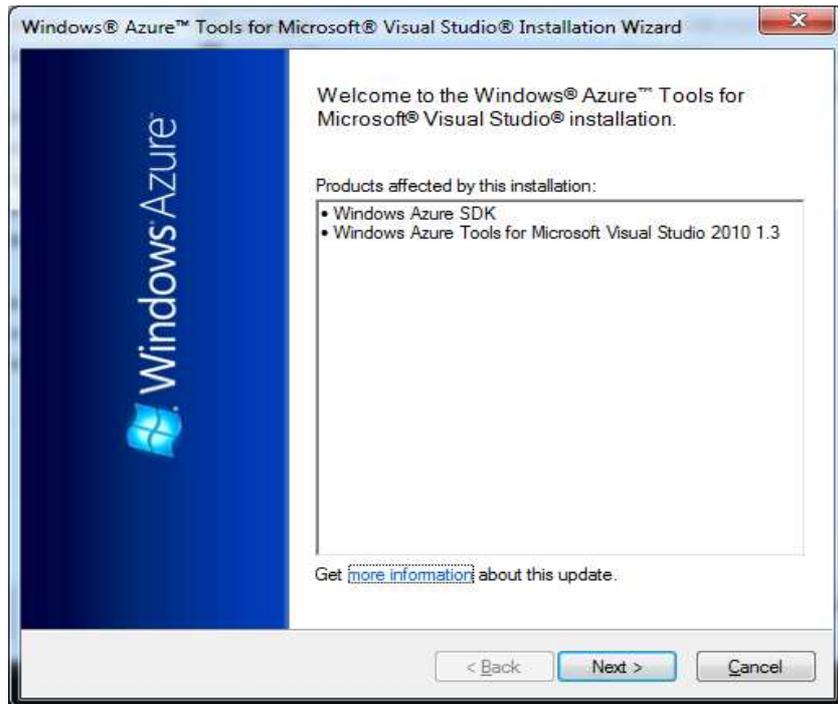


Figura IV.83: Productos a ser instalados

Todos los productos de Microsoft se caracterizan porque vienen incluidos con una licencia, por lo tanto es necesario aceptar los términos de la misma, tomando en cuenta que una buena costumbre es leerla por completo para no tener sorpresas a futuro.

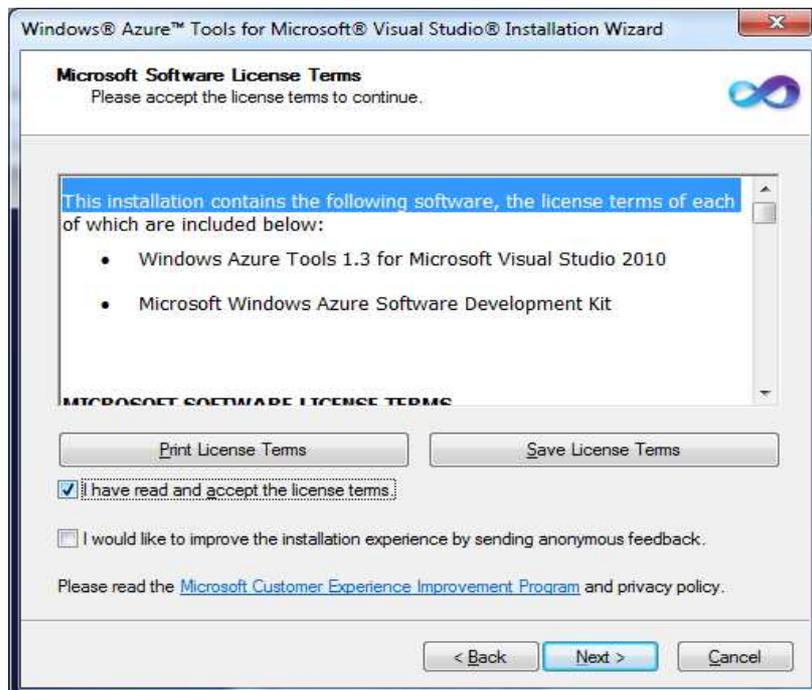


Figura IV.84: Aceptar los términos de la licencia

Se presenta los requerimientos del Sistema y si todo es correcto presionar siguiente.

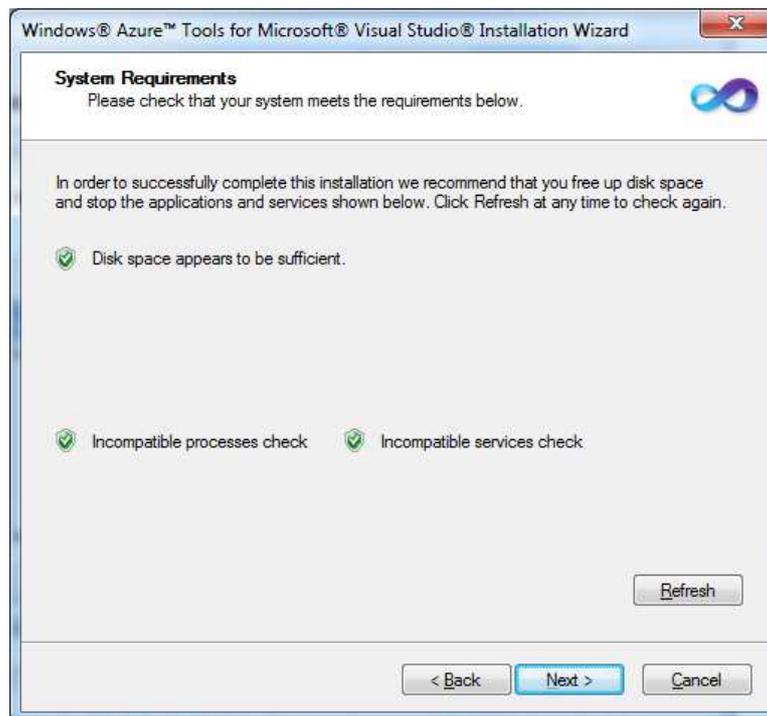


Figura IV.85: Requerimientos del Sistema

Una vez presionado siguiente se mostrará el avance de la instalación y posteriormente presionar finish para terminar con la instalación.

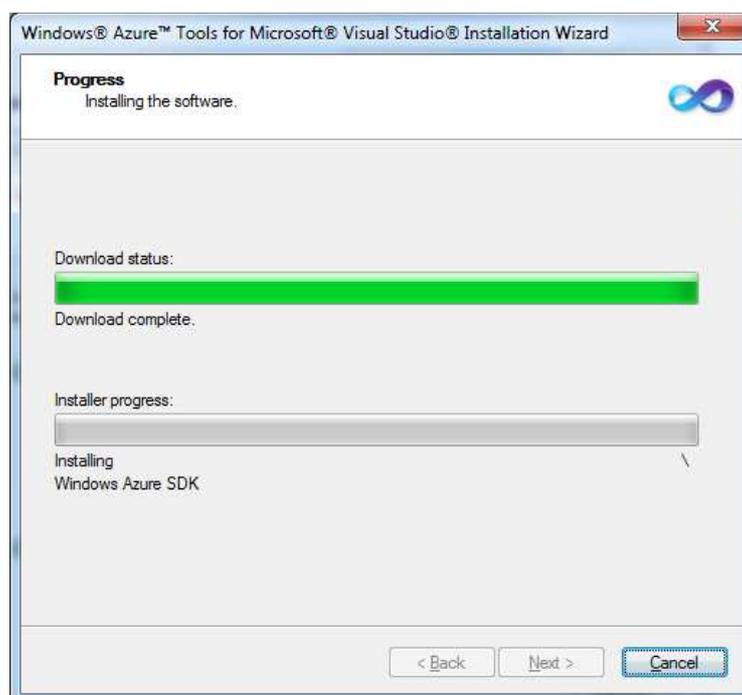


Figura IV.86: Progreso de la instalación del VSCloudService

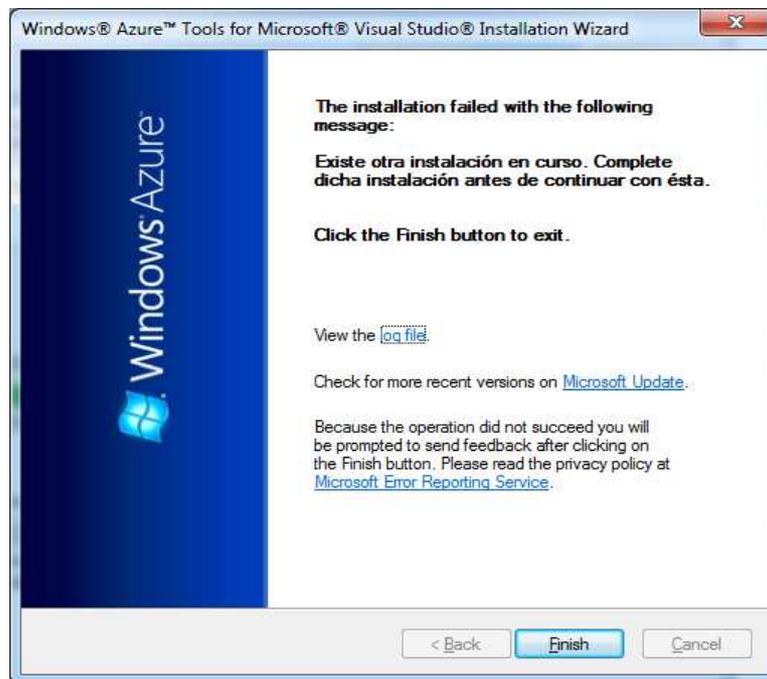


Figura IV.87: Fin de la instalación del VSCloudService

Una vez terminado el proceso de instalación, ya se puede comenzar a trabajar con Visual Studio 2010 Professional, ya que la plataforma de desarrollo acaba de ser actualizada para entender proyectos en la Nube, tomando en cuenta que **será necesario reiniciar** el Sistema Operativo.

4.4. Implementación del sistema

4.4.1. Introducción

Actualmente en la Escuela Superior Politécnica de Chimborazo no se ha realizado ningún tipo de sistema basado en entornos Cloud ya sea para producción o de experimentación, razón por la cual se ha decidido implementar una aplicación para dar a conocer los beneficios que posee.

La aplicación utiliza todas las herramientas indispensables para la implementación. IDE de Desarrollo Visual Studio .NET en la versión 2010 con el Framework 4.0 como plataforma de desarrollo, SQL Azure que es la base de datos relacional en la Nube, AppFabric que es el encargado de interactuar con los servicios, y también se demuestra el uso de las diferentes formas de almacenamiento de los datos.

La Aplicación Web de la Asociación de Escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo contiene información referente a las necesidades de los estudiantes como son: Historia de la Escuela, Tesis realizadas, Noticias y Eventos importantes, Galería de Fotos, Infraestructura entre otras actividades.

4.4.2. Recopilar la información de la Empresa o Institución

4.4.2.1. Breve Reseña Histórica

El 19 de Agosto de 1996, mediante Resolución. No. 236, la Facultad de Ciencias crea adjunto a la Escuela de Computación, la carrera de Ingeniería Electrónica.

Posteriormente se crea la Facultad de Informática y Electrónica, el 29 de enero de 1999, entrando a funcionar como tal el 5 de abril del 2000.

Por lo tanto se crea adjunto la Asociación de Escuela de Ingeniería en Sistemas, la misma desde su creación se encuentra dirigida por los mismos estudiantes de los semestres superiores.

4.4.2.2. Orgánico Funcional



Figura IV.88: Orgánico Funcional.

4.4.2.3. Autoridades.

Actualmente está dirigida por las siguientes autoridades:

Decano.- Ing. Iván Menes

Vicedecano.- Ing. Geovanny Vallejo.

Escuela de Ingeniería en Sistemas – EIS

Director de escuela.- Ing. Raúl Rosero.

Carrera: INGENIERÍA EN SISTEMAS INFORMÁTICOS.

Título: Ingeniero en Sistemas Informáticos.

Duración: 10 Semestres.

4.4.2.4. Objetivos Generales.

El objetivo fundamental de la Escuela de Ingeniería en Sistemas es formar profesionales académicos a nivel superior capacitados para:

- Planificar, diseñar, analizar, evaluar, administrar y desarrollar sistemas de información computacionales.
- Investigar en las áreas del software y hardware para validar la tecnología existente y generar los nuevos proyectos en el ámbito del conocimiento científico – tecnológico que permita optimizar recursos.
- Participar activamente en los programas de desarrollo regional y nacional relacionados con el campo de la informática.

La Asociación de Escuela de Ingeniería en Sistemas, tiene como objetivo principal garantizar la seguridad y dar servicios a toda la colectividad de estudiantes de la escuela, creando un ambiente social, divertido en áreas de capacitación, juegos deportivos, informando las novedades existentes.

4.4.2.5. Objetivos Específicos

Organizar y administrar proyectos de desarrollo de software y sistemas de computación e informática.

- Investigar los temas relacionados en el área de la informática.
- Usar y fomentar el uso de la tecnología relacionada a la computación e informática
- Seleccionar, adaptar e integrar sistemas de computación.
- Aplicar metodologías de análisis y diseño de sistemas, así como la utilización de herramientas innovadoras.

Los objetivos específicos de la Asociación de Escuela de Ingeniería en Sistemas son:

- Brindar capacitación en materias que resultan complicadas a los estudiantes.
- Organizar campeonatos deportivos.

- Brindar seguridad y apoyo a los estudiantes.
- Fomentar el compañerismo entre cada uno de los diferentes semestres y estudiantes.
- Dar a conocer las noticias y eventos relevantes de la Escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo.

4.4.2.6. Investigar los servicios web que brinda la Asociación

Al momento la Asociación de Escuela de Ingeniería en Sistemas no cuenta con un portal web para brindar los servicios, toda la información que reciben es porque se han acercado a las instalaciones de la misma.

4.4.2.7. Servicios que necesita la Asociación de Escuela

En la Escuela Superior Politécnica de Chimborazo se han implementado varios portales de servicios académicos. Para que los usuarios tengan acceso a los servicios informativos deben recurrir físicamente a las instalaciones de la Politécnica. La Asociación de la Escuela de Ingeniería en Sistemas no posee un medio de difusión masivo de sus servicios informativos ni de ningún otro tipo.

La Aplicación web constará de: Información de la misión, visión, logros alcanzados, historia, la nueva infraestructura, galería de fotos, modelo de los oficios entre otros que se utilizan dentro de la Escuela para los diferentes trámites.

4.4.3. Definición del problema

La problemática surge por la inexistencia de un medio que recoja información académica y social de la escuela y brinde servicios accesibles para todos sus usuarios.

Con lo dicho anteriormente se aclara que se busca mejorar los servicios informativos de la EIS así como reducir tiempos de servicio.

4.4.4. Propuesta de solución

La EIS cuenta con los medios tecnológicos para la implantación de un portal web que solvente el problema planteado.

Con el precedente sentado en las anteriores fases, concluimos que nuestra propuesta de dar solución se basará en la creación de un portal web de la Asociación de Escuela de ingeniería en Sistemas, el mismo que estará desarrollado completamente en la plataforma Windows Azure con todos los beneficios que ésta contiene.

Cabe recalcar que la aplicación se la utilizará para demostrar cada uno de los componentes de la Plataforma como son:

- Windows Azure.
- SQLAzure.
- Servicios de almacenamiento (BLOBS, colas y tablas).
- Web Role.
- Worker Role, entre otros.

La aplicación contendrá los siguientes servicios:

- **Menú Principal**
- Misión, Visión, Historia, Logros Alcanzados.
- Infraestructura.
- Noticias.
- Eventos.
- Galería fotográfica.
- Modelo de Oficios.

- **Links a los diferentes lugares sociales**
- Acceso a Hotmail.
- Acceso a Gmail.
- Acceso a Yahoo.
- Acceso a Facebook.
- Acceso a Twitter.

- **Links a los diferentes servicios que brinda la ESPOCH**

- Acceso al E-VIRTUAL.
- Acceso al OASIS.
- Acceso a la Academia CISCO.
- Acceso a la Academia LINUX.
- Acceso al WEBMAIL.
- Acceso a la página nacional de TESIS.

- **Otros enlaces**

- Inicio.
- Acerca de.
- Salir.
- Déjanos tus comentarios.
- Contador de Visitas.

4.4.5. Desarrollo de la aplicación.

La aplicación está desarrollada en su totalidad en la plataforma Windows Azure con todos los beneficios que presta la misma.

A continuación se detalla el código y herramienta utilizado para cada uno de los componentes descritos anteriormente.

Existen 2 formularios principales los mismos que son: “Default” e “Index_Administrador”.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
Default	Es la página de bienvenida, en la cual se ingresa directamente sin clave de acceso, y lo utiliza un usuario invitado.
Index Administrador	Ingresa mediante clave de acceso, la misma que se valida totalmente hacia SQL Azure.

Tabla IV.XXIII: Descripción de los formularios principales.

INGRESO AL SISTEMA POR PARTE DEL ADMINISTRADOR Y USUARIO

En cualquier momento puede ingresar como administrador, ya que en la parte superior existe un formulario para escribir el Usuario y Clave respectiva, a continuación se muestra los componentes que se utilizan para el Inicio de Sesión.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
LogProcesa.aspx	Esta página utiliza código ASP para su desarrollo, se ejecuta solo cuando accede el usuario Administrador.
clsConexion.cs	Clase que permite la conexión a la base de datos del SQL Azure
clsUsuario.cs	Clases de la lógica de negocio que trabaja con la tabla Usuarios del SQL Azure.

Tabla IV.XXIV: Descripción de ingresos al sistema de los usuarios.

INGRESO AL SISTEMA POR PARTE DE UN USUARIO NORMAL O INVITADO

Un usuario normal o invitado es aquel que solo va a poder utilizar el sistema, pero no podrá modificarlo, tiene el privilegio de descargar archivos.

TRABAJO CON BLOBS DE WINDOWS AZURE

Los BLOBS se han utilizado en las secciones de:

- Galería Fotográfica
- Modelo de Oficios

Galería Fotográfica

Forma parte del Menú Principal de la aplicación, se utiliza BLOBS porque ésta estructura permite ingresar archivos de diferentes tamaños en su almacenamiento. La estructura de los BLOBS se utiliza tanto para administrar la Galería de Fotos como para simplemente visualizarla es decir utiliza el usuario Administrador y el usuario Normal. A continuación se muestra los componentes utilizados.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
subirCal.aspx	Esta página utiliza código ASP para su desarrollo, se ejecuta solo cuando accede el usuario Administrador.

subirCal.aspx.cs	Clase del formulario subirCal.aspx, la misma es la que se une con los BLOBS.
BLOBService.cs	Clase que administra a los BLOBS
VisualizarImagenesBlob.aspx	Esta página permite visualizar las imágenes del Blob
VisualizarImagenesBlob.aspx.cs	Clase del formulario VisualizarImagenesBlob.aspx, la misma es la que se une con los BLOBS.

Tabla IV.XXV: Detalle de la Galería Fotográfica.

Modelo de oficios

La opción Modelo de Oficios está desarrollada mediante BLOBS, pero a la vez implementa QUEUES que utiliza para ir contabilizando el número de imágenes de oficios se ha subido al servidor, toda esta programación de las COLAS o QUEUES se realiza en segundo plano, porque como se explicó en el Capítulo 2, sección 2.2, un worker role no se comunica con el usuario final.

A continuación se muestra los componentes utilizados.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
subirCertificados.aspx	Formulario disponible para los usuarios que son administradores, se utiliza para subir las imágenes de los oficios.
subirCertificados.aspx.cs	Clase del formulario subirCertificados.aspx, la misma es la que se une con los BLOBS.
WorkerRole.cs	La clase del Worker Role de los certificados, se ejecuta en segundo plano.
CertificadosMetadata.cs	Clase de los metadatos o atributos de los certificados.
VisualizarCertificados.aspx	Esta página permite visualizar las imágenes del Blob
VisualizarCertificados.aspx.cs	Clase del formulario VisualizarCertificados.aspx, la misma es la que se une con los BLOBS.

Tabla IV.XXVI:Detalle de la opción modelo de oficios.

TRABAJO CON TABLAS DE WINDOWS AZURE

Las TABLAS se han utilizado en las secciones de:

- Misión, Visión, Historia y Logros.
- Comentarios.

Misión, Visión, Historia y Logros.

Forma parte del Menú Principal de la aplicación se utiliza tablas del propio Windows Azure, porque esta estructura permite ingresar información como si fuera una tabla del SQL con la única diferencia que no se pueden relacionar.

Esta estructura se utiliza tanto para administrar y visualizar es decir utiliza el usuario Administrador y el usuario Normal.

A continuación se muestra los componentes utilizados.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
subirMisionVisionEtc.aspx	Formulario disponible para los usuarios que son administradores, se utiliza para ingresar los datos de misión, visión, etc.
subirMisionVisionEtc.aspx.cs	Clase del formulario subirMisiónVisiónEtc.aspx, la misma es la que interactúa con el resto de clases.
MisionVisionEntity.cs	Clase de los metadatos o atributos de la tabla, aquí se describe los campos y tipo de datos.
MisionVisionDataSource.cs	Clase que interactúa con el dataset que se encuentra en SubirMisionVisionEtc.aspx.
VisualizarMisionVisionEtc.cs	Clase que accede a visualizar la visión y misión

Tabla IV.XXVII: Detalle de la opción Misión, visión, historia y logros

Comentarios.

Forma parte del Menú Principal, ésta estructura se utiliza tanto para administrar y visualizar es decir utiliza el usuario Administrador y el usuario Normal.

A continuación se muestra los componentes utilizados.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
ComentariosAdministrador.aspx	Formulario disponible para los usuarios que son administradores, se utiliza para ingresar los datos de misión, visión, etc.
ComentariosAdministrador.aspx.cs	Clase del formulario ComentariosAdministrador.aspx, la misma es la que interactúa con el resto de clases.
clsComentarios.cs	Clase de los metadatos o atributos de la tabla, aquí se describe los campos y tipo de datos.
ComentariosDataSource.cs	Clase que interactúa con el dataset que se encuentra en ComentariosAdministrador.aspx.

Tabla IV.XXVIII: Detalle de la opción Déjenos tus comentarios.

TRABAJO CON TABLAS DE SQL AZURE

Las tablas del SQL Azure se han utilizado en las secciones:

- Eventos
- Noticias

Eventos.

Forma parte del Menú Principal de la aplicación se utiliza tablas del SQL Azure, porque esta estructura es la misma del SQL Server 2008 R2, por lo que su administración se la puede hacer desde el portal o desde el Management del SQL Server 2008 R2.

Esta estructura se utiliza tanto para administrar y visualizarla es decir utiliza el usuario Administrador y el usuario Normal.

A continuación se muestra los componentes utilizados.

CLASE / MÓDULO / FORMULARIO	ACTIVIDADES
SubirEventos.aspx	Formulario disponible para los usuarios que son administradores, se utiliza para ingresar los datos a la Base de Datos del SQL Azure.
SubirEventosProcesa.aspx	Formulario intermedio que procesa los datos y los envía a la Base de Datos.
clsEventos.aspx	Clases de la lógica de negocio que trabaja con la tabla Eventos del SQL Azure.
VisualizarEventos.aspx	Visualiza los eventos ejecutando mediante un procedimiento almacenado

Tabla IV.XXIX: Detalle de la opción Eventos.

Noticias.

Forma parte del Menú Principal, ésta estructura se utiliza tanto para administrar y visualizar es decir utiliza el usuario Administrador y el usuario Normal.

A continuación se muestra los componentes utilizados.

Clase / Módulo / Formulario	Actividades
SubirNoticias.aspx	Formulario disponible para los usuarios que son administradores, se utiliza para ingresar los datos a la Base de Datos del SQL Azure.
SubirNoticiasProcesa.aspx	Formulario intermedio que procesa los datos y los envía a la

	Base de Datos.
clsNoticias.aspx	Clases de la lógica de negocio que trabaja con la tabla Noticias del SQL Azure.
VisualizarNoticias.aspx	Visualiza las noticias ejecutando mediante un procedimiento almacenado

Tabla IV.XXX: Detalle de la opción Noticias.

4.5. Desplegar la aplicación en la Nube de Windows Azure.

Una vez terminada la explicación del desarrollo se procede a desplegar en la Nube para lo cual se siguen los siguientes pasos:

4.5.1. Creación de una cuenta en la Plataforma de Windows Azure

4.5.1.1. Introducción.

Para este trabajo de investigación se utilizó una Cuenta de Prueba de 30 días, la misma que tiene similares características que una versión comercial y para la creación se realizó lo siguiente:

4.5.1.2. Proceso de obtención de la cuenta

Primero ingresar a la página web oficial de Emprendedores con Tecnología que son los encargados de entregar claves de acceso.



Figura IV.89: Página principal de Emprendedores con Tecnología

Registrarse con la inscripción “BizMXAzure”, y seguir con los pasos que se presentan.

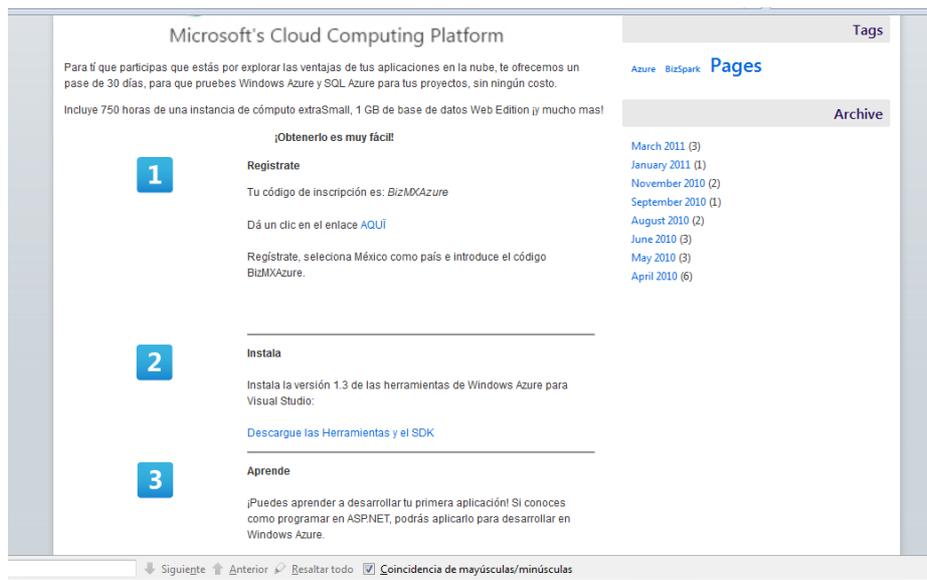


Figura IV.90: Opciones de la página de Emprendedores con Tecnología

Seleccionar presionando clic en la palabra “AQUÍ” para poder acceder a la promoción.



Figura IV.91: Pais y clave de promoción.

Como la cuenta es de promoción se debe seleccionar el país **México**, y la clave de promoción es **BizMXAzure**, dar clic en Enviar

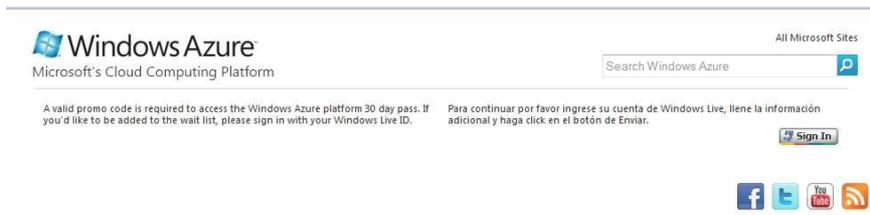


Figura IV.92: Advertencia para que el usuario que no se ha logueado

Enviar una advertencia que para continuar primero se debe autenticar con una cuenta de correo electrónico de Hotmail.

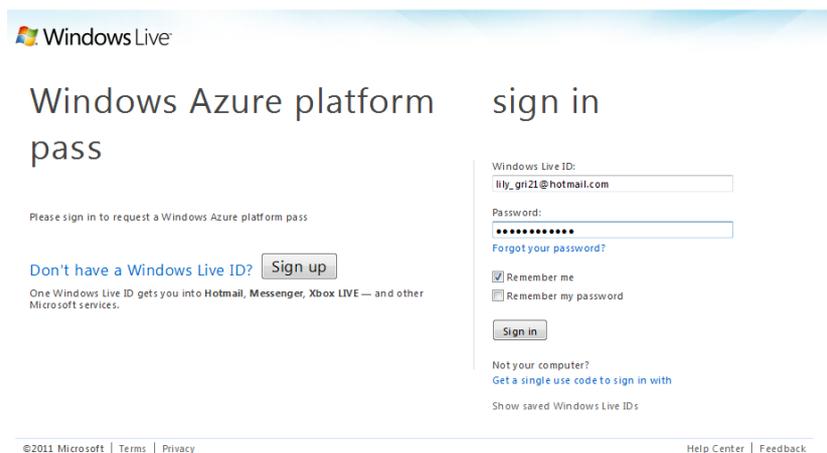
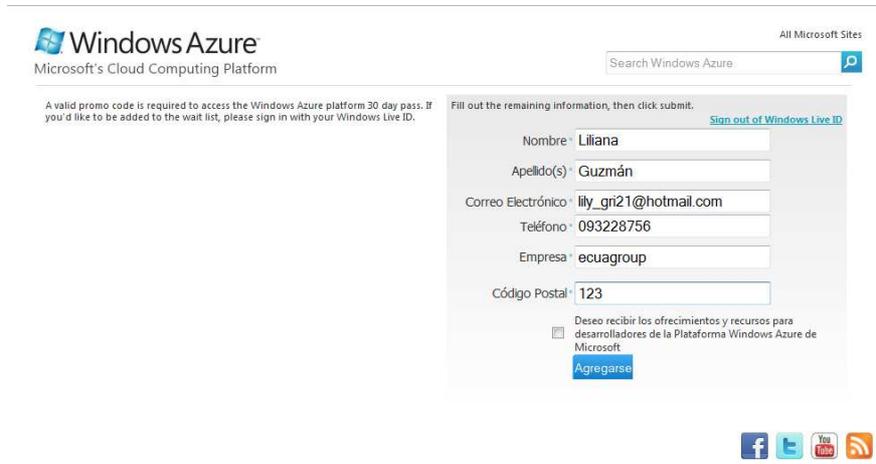


Figura IV.93: Logueo e ingreso al Sistema de Windows Azure

Una vez iniciado la sesión en Hotmail para la plataforma de Windows Azure aparecerá la siguiente ventana en la cual se confirman los datos del usuario.



The screenshot shows the Windows Azure registration page. At the top left is the Windows Azure logo and 'Microsoft's Cloud Computing Platform'. A search bar is at the top right. The main content area contains a message: 'A valid promo code is required to access the Windows Azure platform 30 day pass. If you'd like to be added to the wait list, please sign in with your Windows Live ID.' Below this is a form titled 'Fill out the remaining information, then click submit.' with a 'Sign out of Windows Live ID' link. The form fields are: Nombre: Liliانا, Apellido(s): Guzmán, Correo Electrónico: lily_gri21@hotmail.com, Teléfono: 093228756, Empresa: ecuagroup, and Código Postal: 123. There is a checkbox for 'Deseo recibir los ofrecimientos y recursos para desarrolladores de la Plataforma Windows Azure de Microsoft' which is checked. A blue 'Agregar' button is at the bottom of the form. Social media icons for Facebook, Twitter, YouTube, and RSS are at the bottom right.

Figura IV.94: Datos completos para crear la cuenta

Después de dar clic en “Agregar”, aparecerá el acuerdo de Servicio del Pase de 30 días para Windows Azure, por lo que es necesario aceptar los términos de la licencia y llenar los datos que solicitan, por último presionar Aceptar.



The screenshot shows the 'Acuerdo de Servicio del Pase de 30 días para Windows Azure' page. The title is 'Acuerdo de Servicio del Pase de 30 días para Windows Azure'. Below the title is a scrollable window containing the 'MICROSOFT WINDOWS AZURE 30 DAY PASS TRIAL AGREEMENT'. The agreement text includes: 'Last Updated November, 2010', '1. WHAT THE AGREEMENT COVERS', 'This agreement is between the entity agreeing to these terms ("Customer," "you" or "your") and Microsoft Corporation, ("Microsoft," "we," "us," or "our"). If you are accepting this agreement on behalf of your employer or another legal entity, you represent and warrant that you have the authority to bind that entity and that you are agreeing to these terms on behalf of that entity. If you cannot truthfully make these representations, do not click "accept" below. The terms and conditions of this agreement apply to Customer's access and use of certain trial versions of Microsoft Windows Azure 30 day pass, including any technology, information, software, materials, and updates that Microsoft makes available to Customer as part of such services (the "Services"). Unless otherwise expressly noted, the following terms apply to all Services provided to Customer under this agreement.', '2. DATA', '2.1 Customer's Data. The Services may allow Customer to store, process, access, and query Data from Customer's devices without having to replicate or move Data. "Data" means all data, including all text, sound, or image files that Customer uploads to or that is processed using the Services.', and '2.2 Customer's Responsibility.'. Below the scrollable window is a checkbox with the text 'Al seleccionar esta casilla y escribir su nombre y apellido, usted está aceptando el Acuerdo del Pase de Prueba de 30 días de Windows Azure.' and a blue 'Aceptar' button. Below the checkbox are two input fields: 'First Name: Liliانا' and 'Last Name: Guzman'. Social media icons for Facebook, Twitter, YouTube, and RSS are at the bottom right.

Figura IV.95: Acuerdo de Servicio del Pase de 30 días para Windows Azure

Una vez presionado en Aceptar y si todo fue correcto se ingresa a una lista de espera, que por motivos logísticos se activan en 3 días laborables llegando un mensaje al correo dándonos la bienvenida a la PLATAFORMA WINDOWS AZURE.

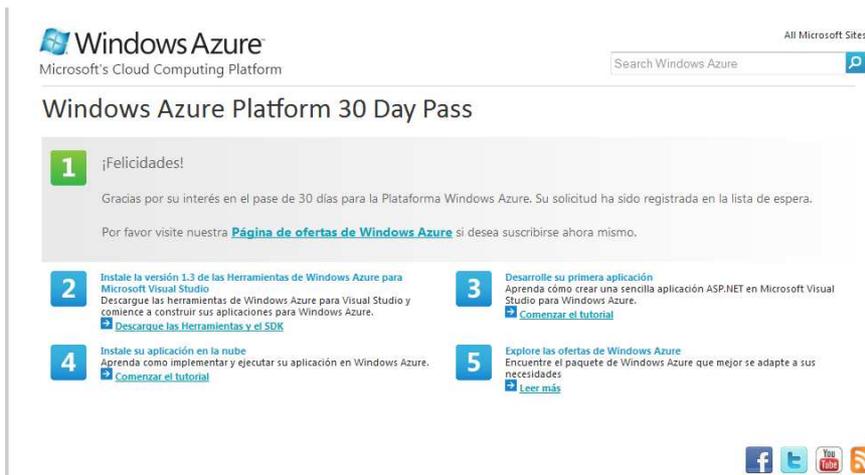


Figura IV.96: Confirmación y registro en la lista de espera.

Terminada la creación de la cuenta en la plataforma se tendrá acceso mediante la URL: <https://windows.azure.com/default.aspx>

4.5.2. Administración de la cuenta en la Plataforma Windows Azure

La pantalla principal del portal de Administración o Management Portal es la siguiente:

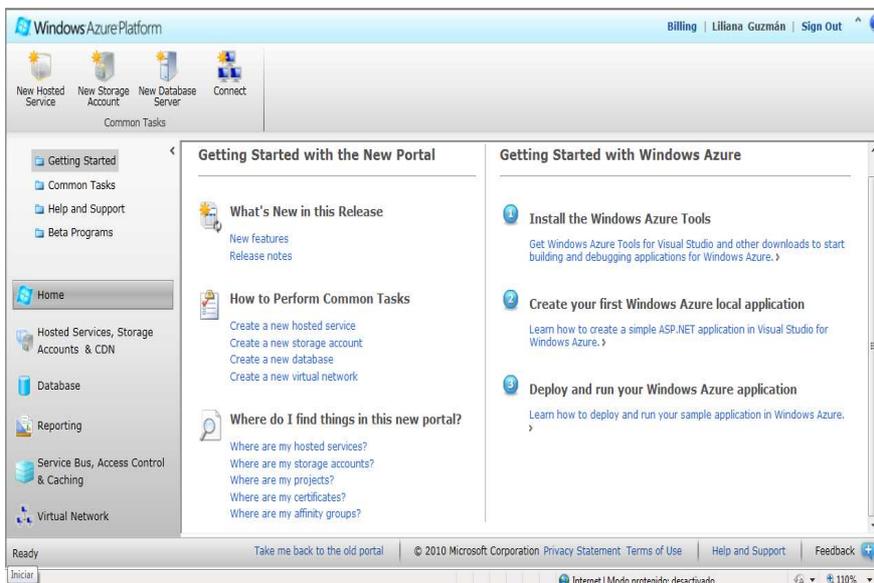


Figura IV.97: Portal de Administración de Windows Azure

En esta ventana se puede administrar

- HostedServices, StoragesAccounts
- Database
- Reporting
- Service Bus, Acces Control
- Virtual Network

4.5.3. Creación de la cuenta de Almacenamiento (StoragesAccounts.)

Las cuentas de almacenamiento son utilizadas tanto para la creación de BLOBS, tablas y colas.

Los pasos son los siguientes:

Ingresar a la página <https://windowsazure.com/default.aspx> con el usuario y clave que tenga acceso a la plataforma.

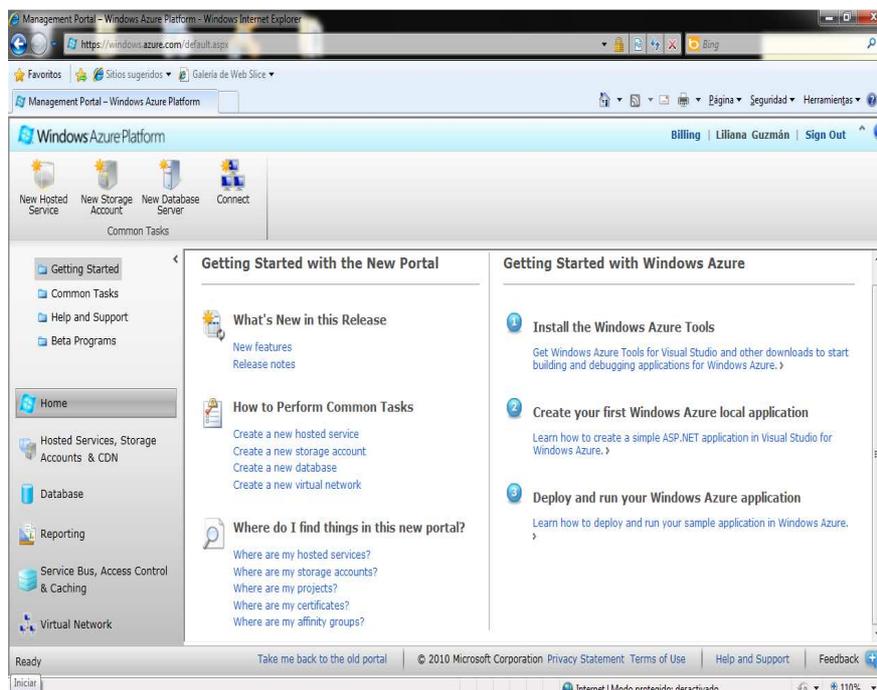


Figura IV.98: Portal de Administración de Windows Azure

Clic en el icono "New Storage Account", para crear una nueva cuenta, en la imagen siguiente ingresar el nombre para la cuenta y la región

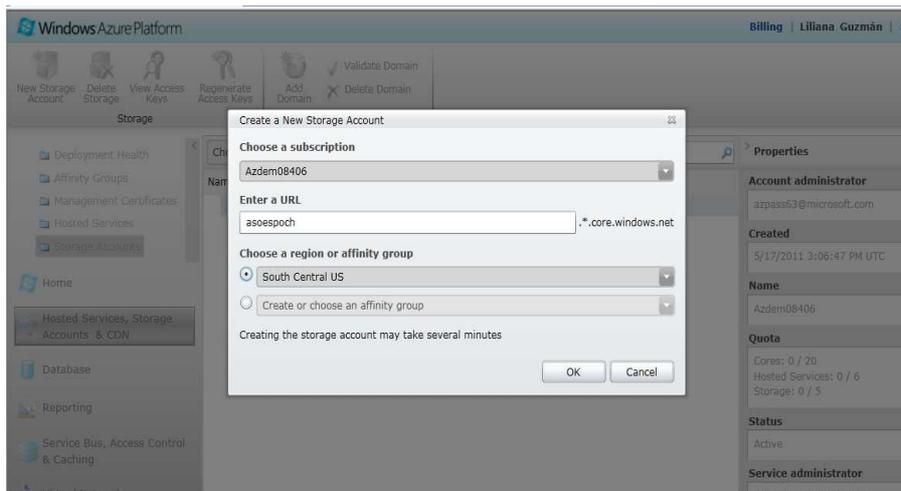


Figura IV.99: Nombre y región para la creación de la cuenta de Storage

Una vez creada la cuenta, en la parte derecha de la imagen se encuentran las propiedades de las cuentas

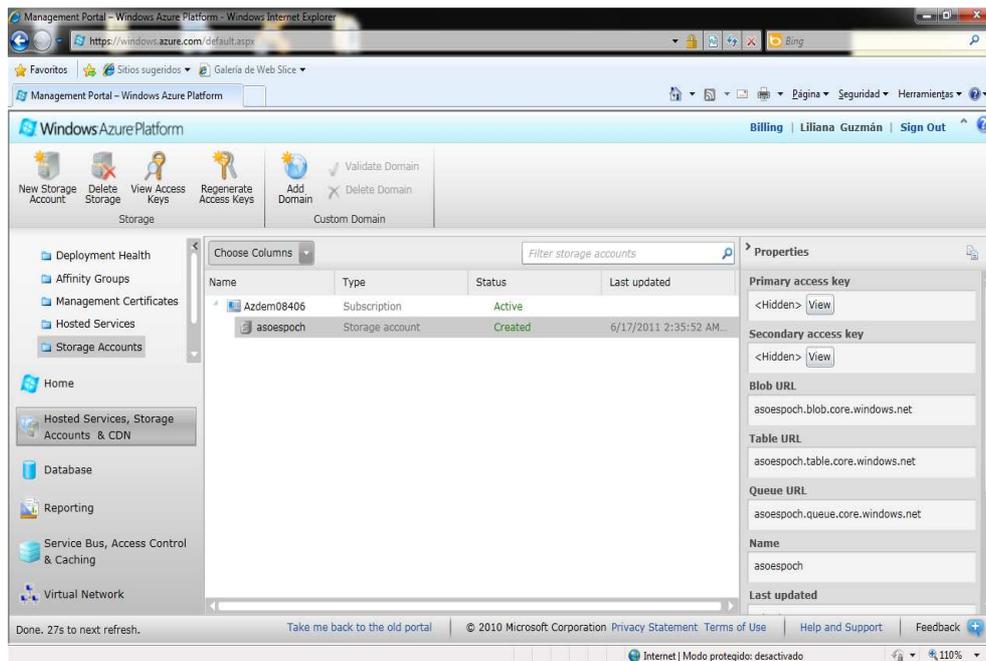


Figura IV.100: Cuenta creada y ventana de propiedades

A continuación se observa las claves tanto primaria como secundaria, las mismas que se deben ingresar en el proyecto de Visual Studio .NET 2010 en el momento de la programación.

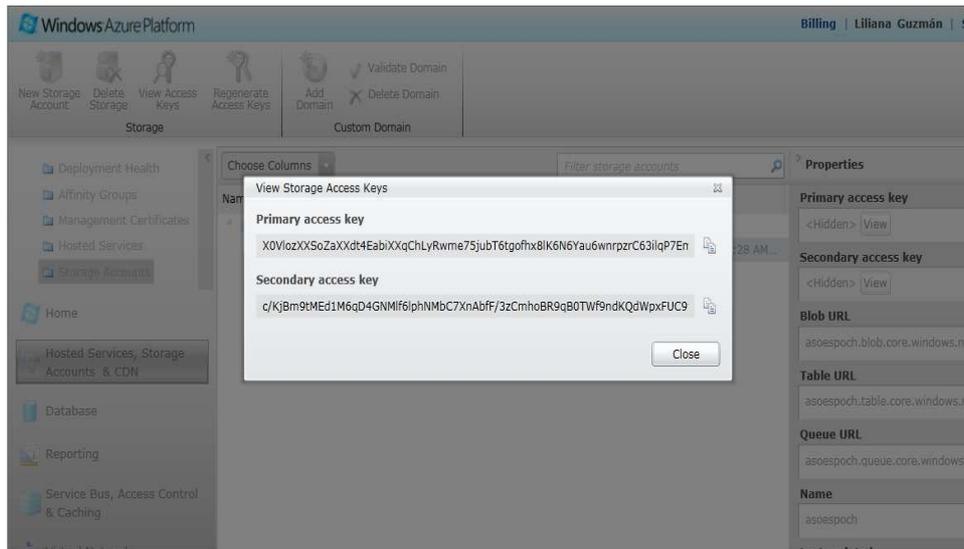


Figura IV.101: Claves primaria y secundaria del Storage Account

4.5.4. Creación de un nuevo Servidor de Base de Datos de SQL Azure.

Pulsamos en el icono Database para crear una nueva base de datos en SQL Azure



Figura IV.102: Nueva base de datos

Aparece la ventana para crear un nuevo servidor de Base de Datos

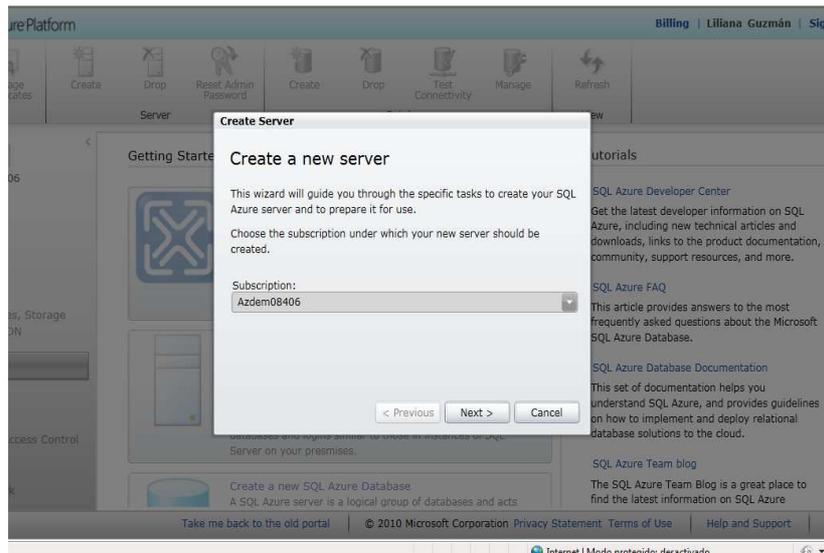


Figura IV.103: Nombre del servidor de base de datos

Seleccionar la región para el nuevo servidor de base de datos de SQL Azure.

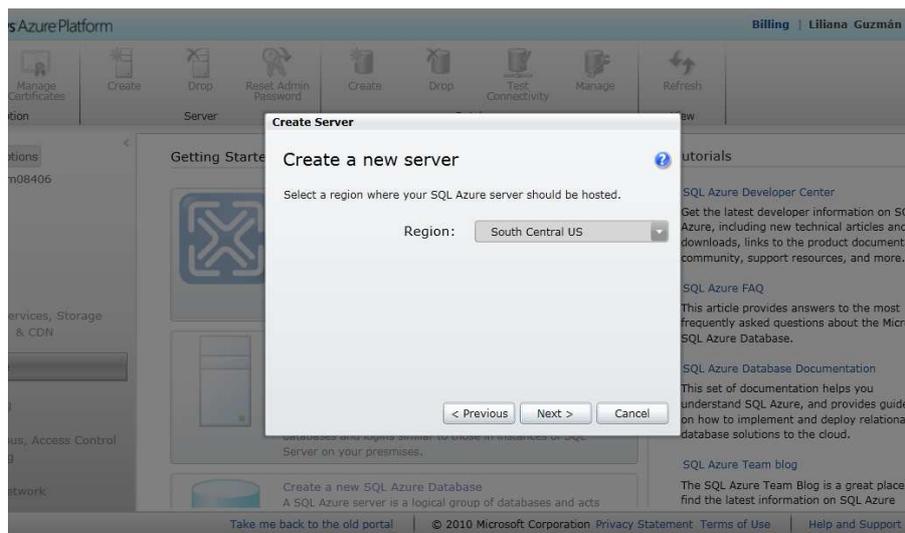


Figura IV.104: Región del Servidor de base de datos

Ingresar el nombre de usuario, y la contraseña, mencionaremos que se utilizó los siguientes datos:

Usuario: Asosistemas

Clave: Ecuagroup123

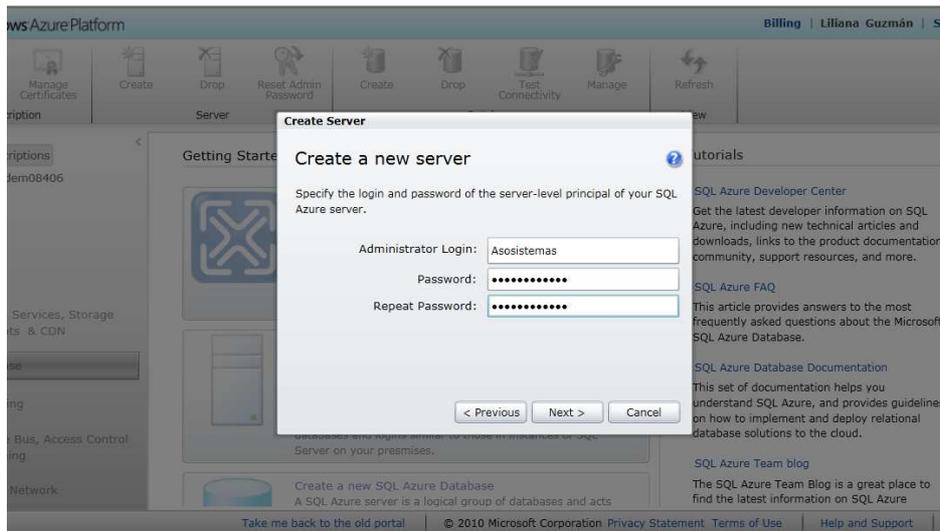


Figura IV.105: Usuario y clave del Servidor de base de datos

Para poder administrar la base de datos desde el SQL Server 2008 R2 es necesario crear las reglas de firewall, es decir permitir que la IP del equipo local pueda acceder a la base de datos del SQL Azure en el Internet.

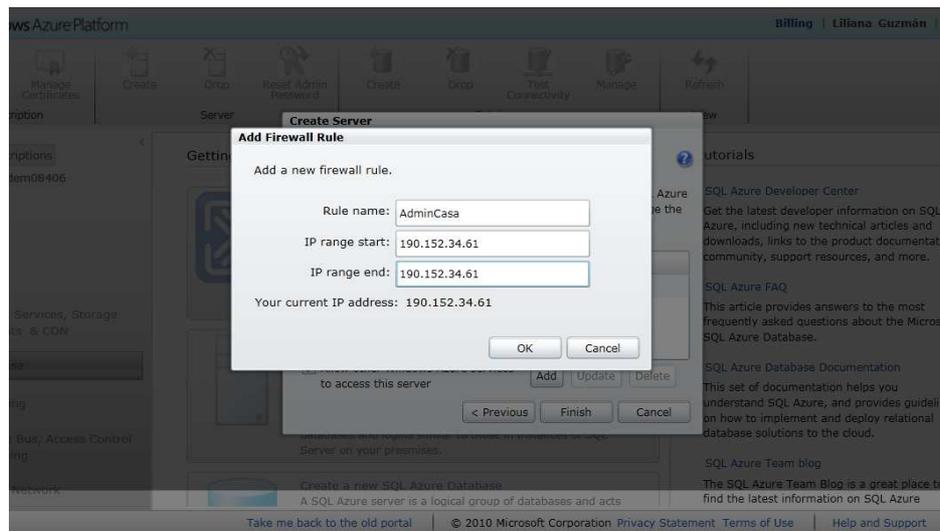


Figura IV.106: Creación de la regla de firewall del Servidor de base de datos

A continuación se observa la creación de la nueva Base de Datos.

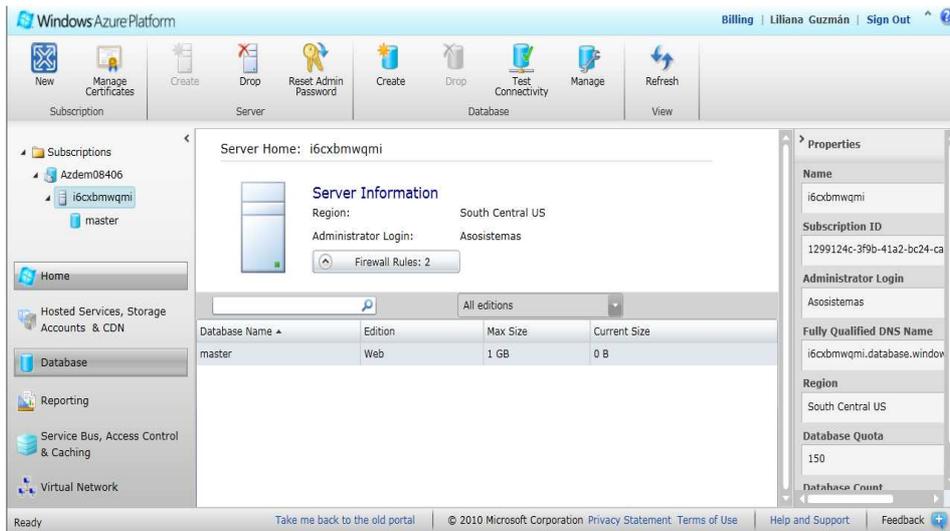


Figura IV.107: Creación de la nueva base de datos

Escribir el nombre de la Base de Datos, la edición y el tamaño máximo, el servidor se llama AsoSistemas.

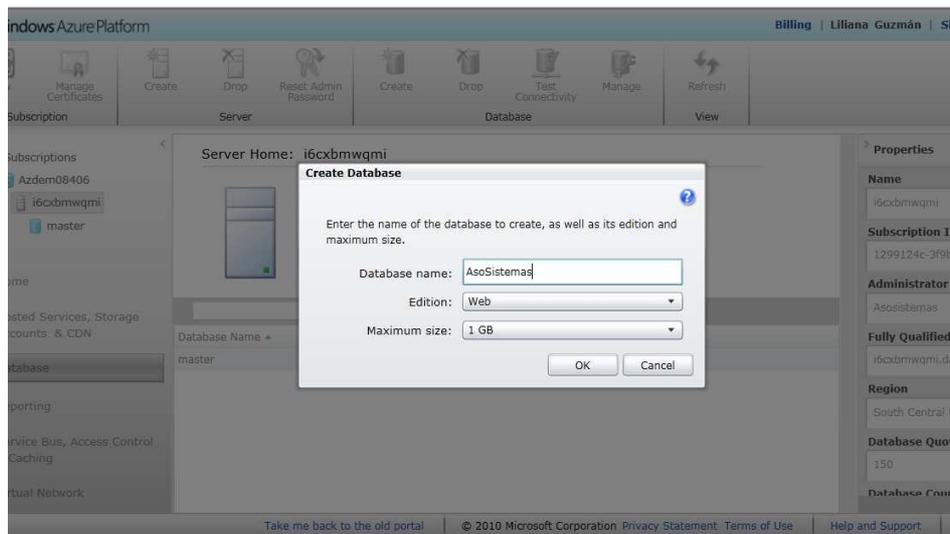


Figura IV.108: Datos necesarios de la nueva base de datos

Para poder administrar la Base de Datos se tiene que dar clic en el ícono Manage, del portal de Administración de Windows Azure y de base de datos.

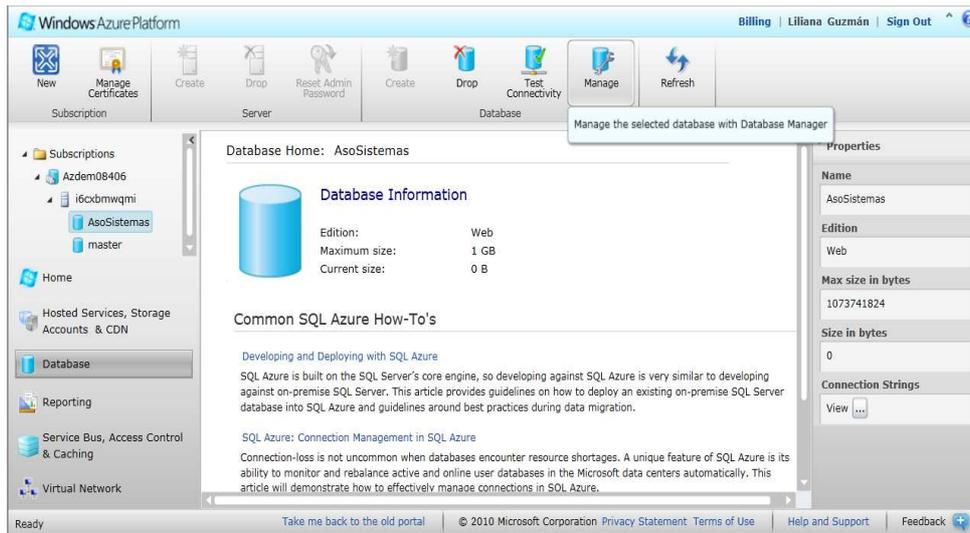


Figura IV.109: Icono de administración de base de datos

Una vez dado clic en el ícono ingresar los datos para la correcta autenticación e ingreso a la administración

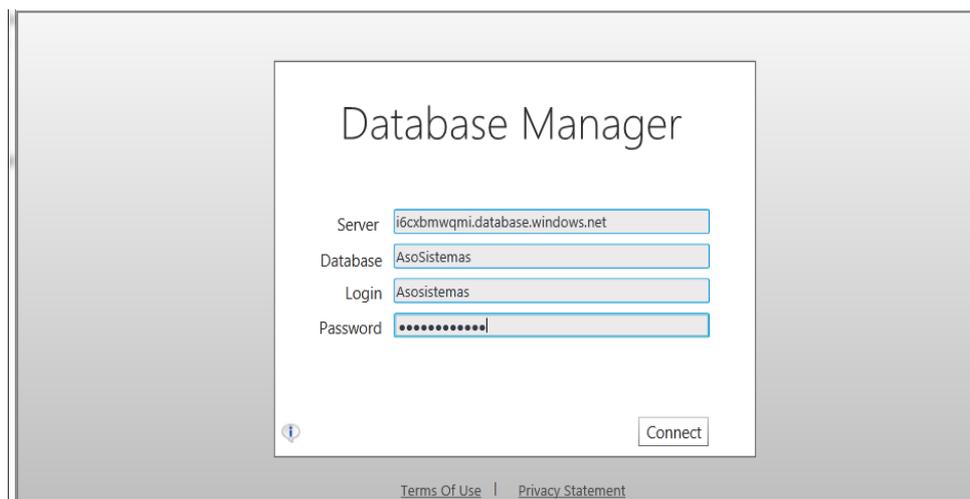


Figura IV.110: Datos necesarios para ingresar a la administración de base de datos

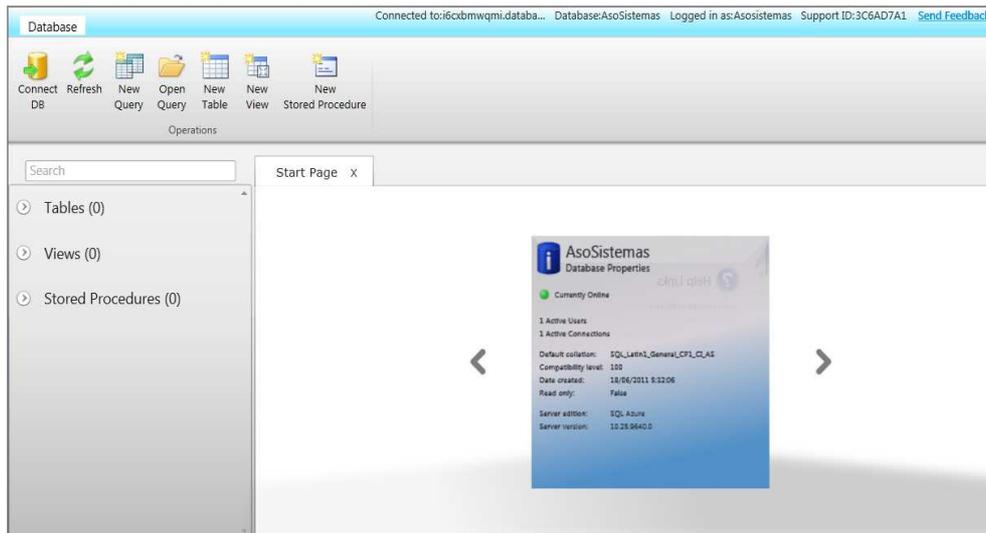


Figura IV.111: Página inicial de la Administración de base de datos

La aplicación en primera instancia se desarrolló en el servidor que tiene las características de la plataforma Windows Azure, con SQL Server 2008 R2, por lo que para levantar la base de datos en la Nube es necesario generar un script en el motor de la base de datos y de ahí ejecutar dicho script

Pasos para generar el script SQL Server 2008 R2

El script tiene que ser compatible con Azure

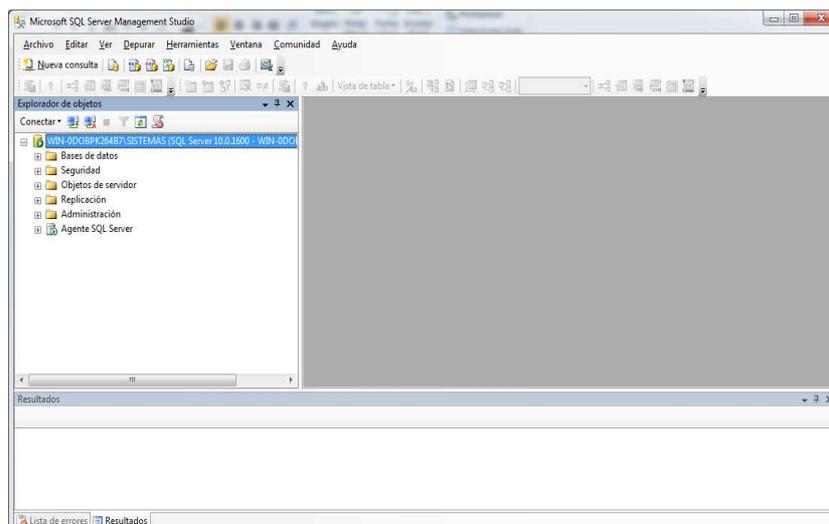


Figura IV.112: Página inicial de la Administración de base de datos

En el Microsoft SQL Server Management Studio conectarse con el servidor de Base de Datos

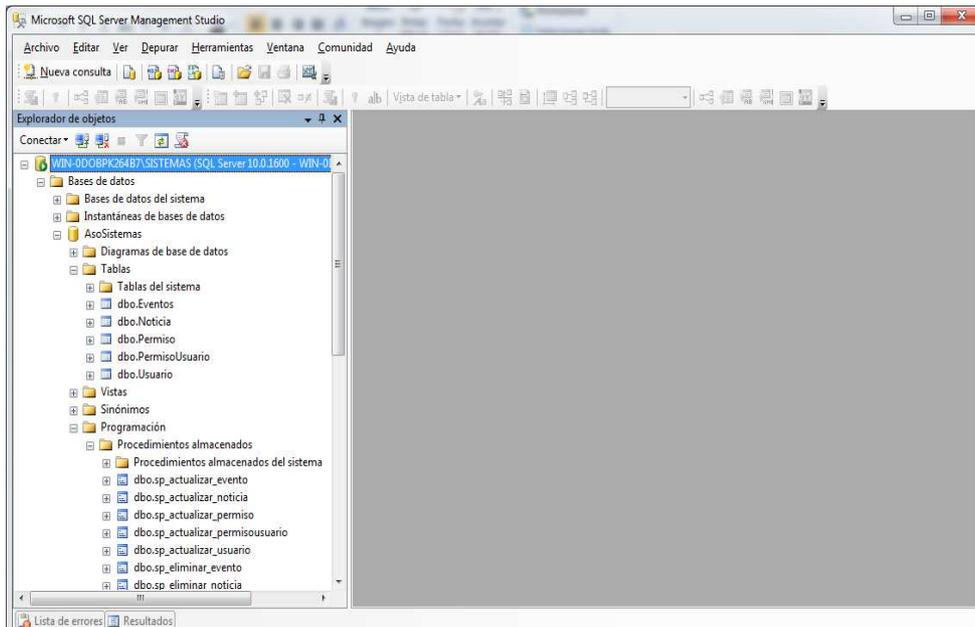


Figura IV.113: Microsoft SQL Server Management Studio.

En el nombre de la base de datos, clic secundario tareas, generar scripts

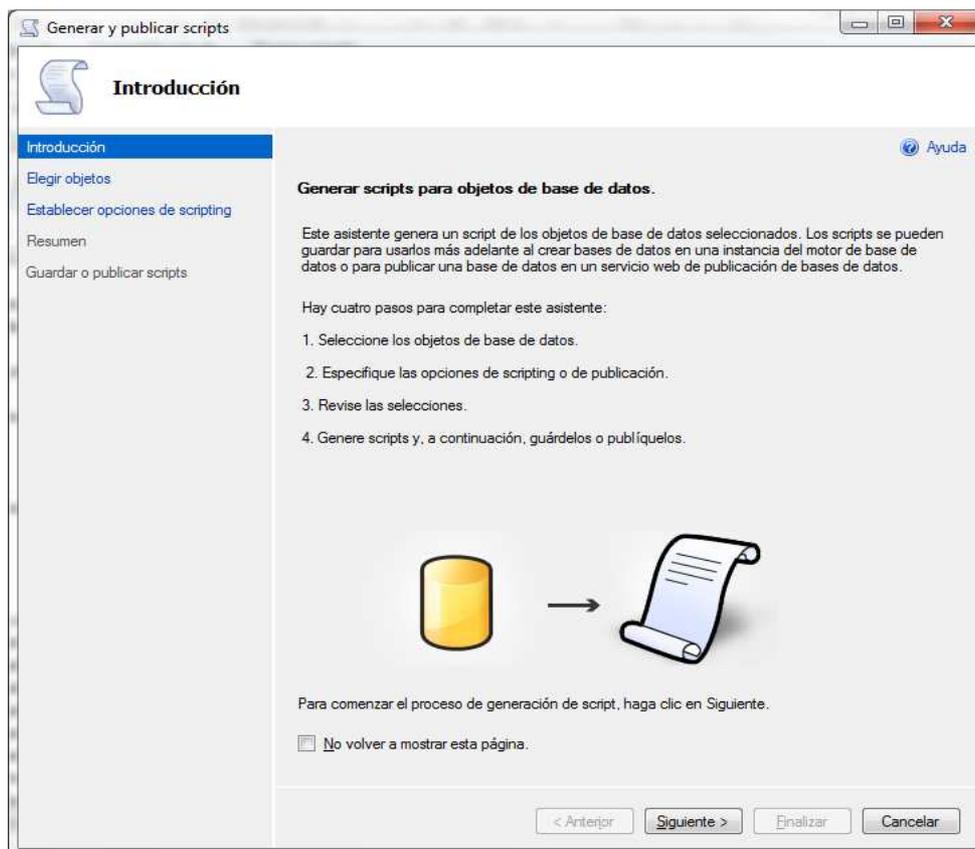


Figura IV.114: Generar scripts para objetos de base de datos

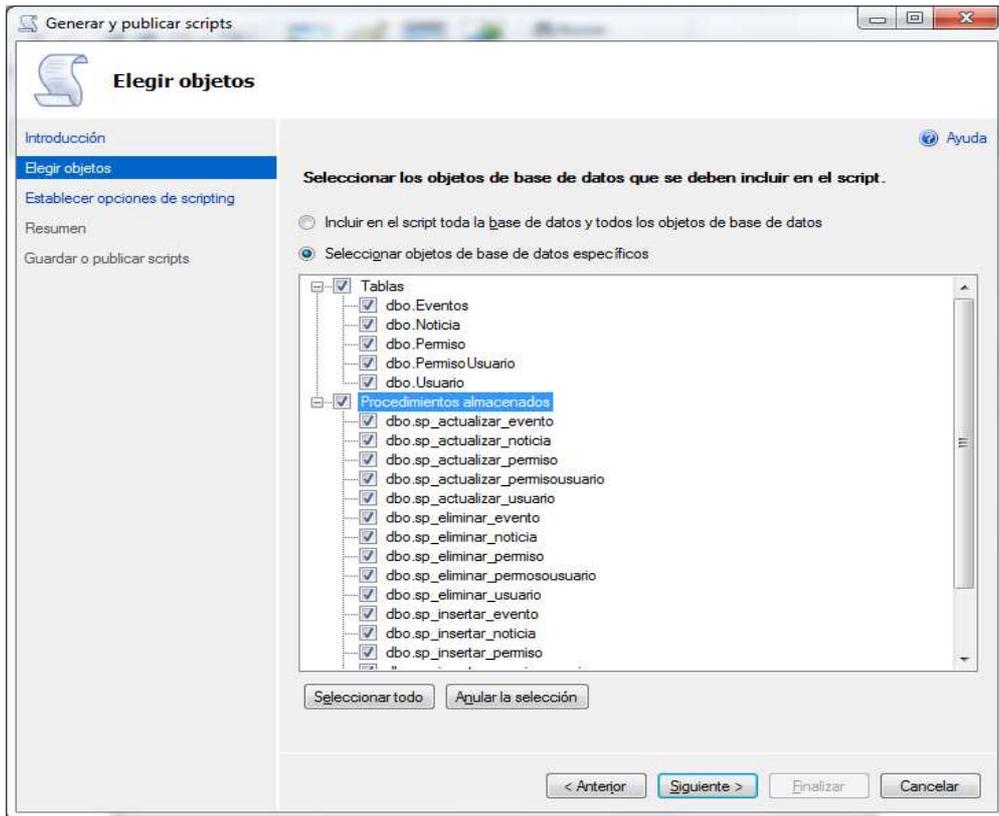


Figura IV.115: Seleccionar los objetos de base de datos en el script.

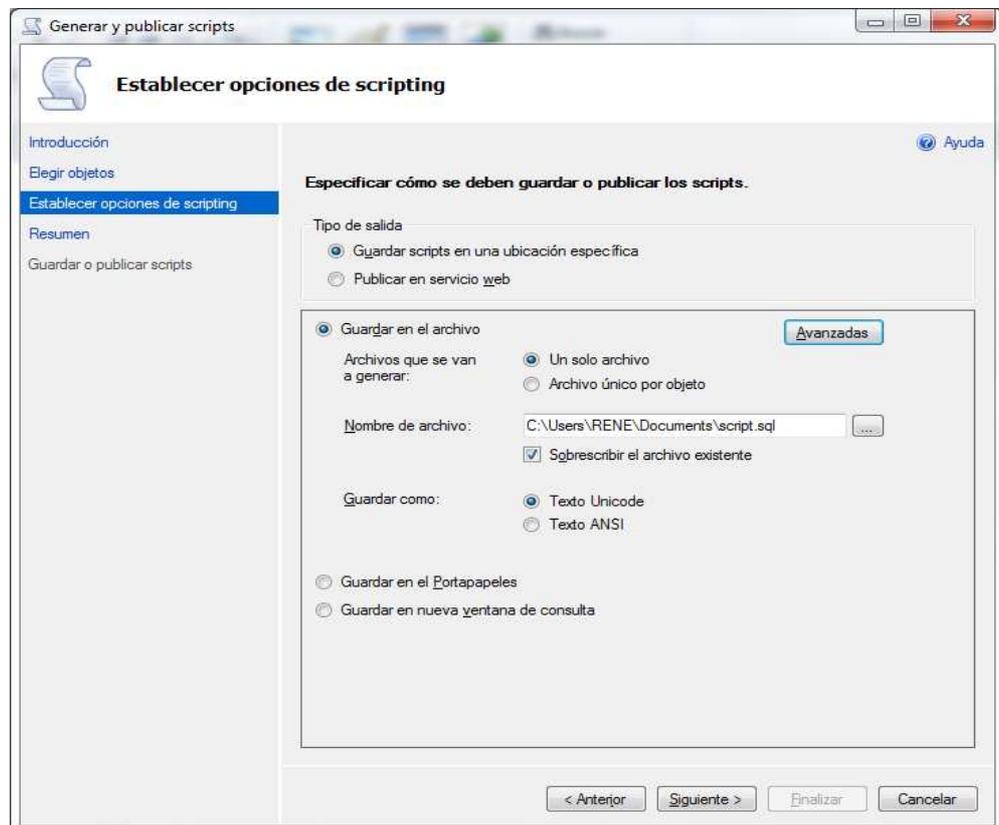


Figura IV.116: Opciones avanzadas del script.

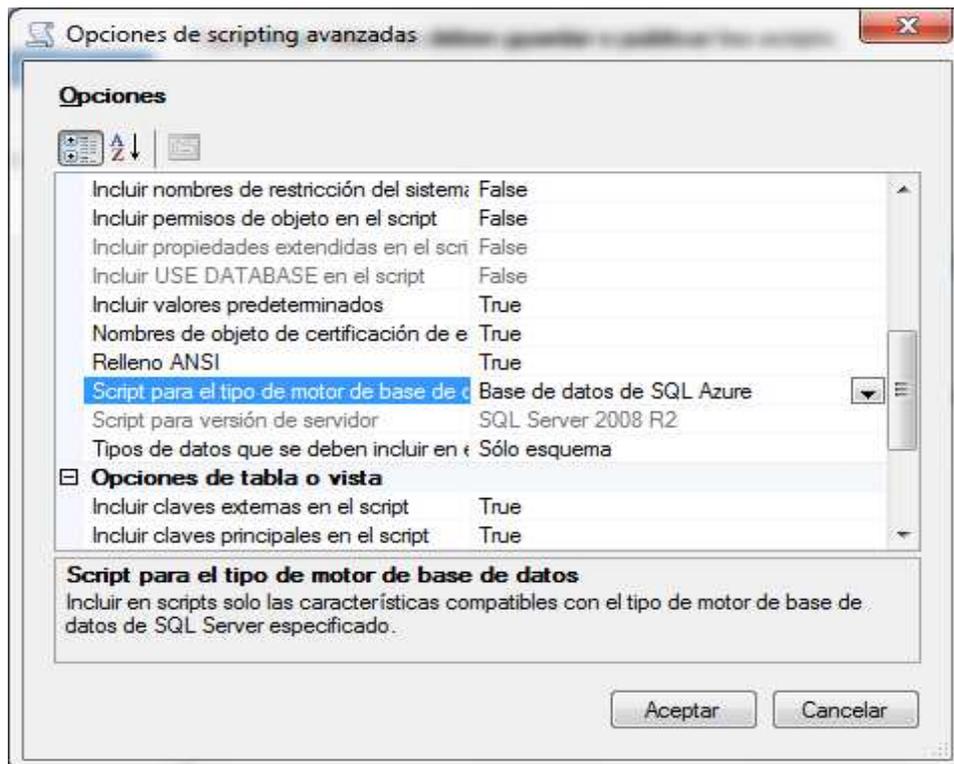


Figura IV.117: Habilitando el script para el tipo de motor de base de datos Azure.

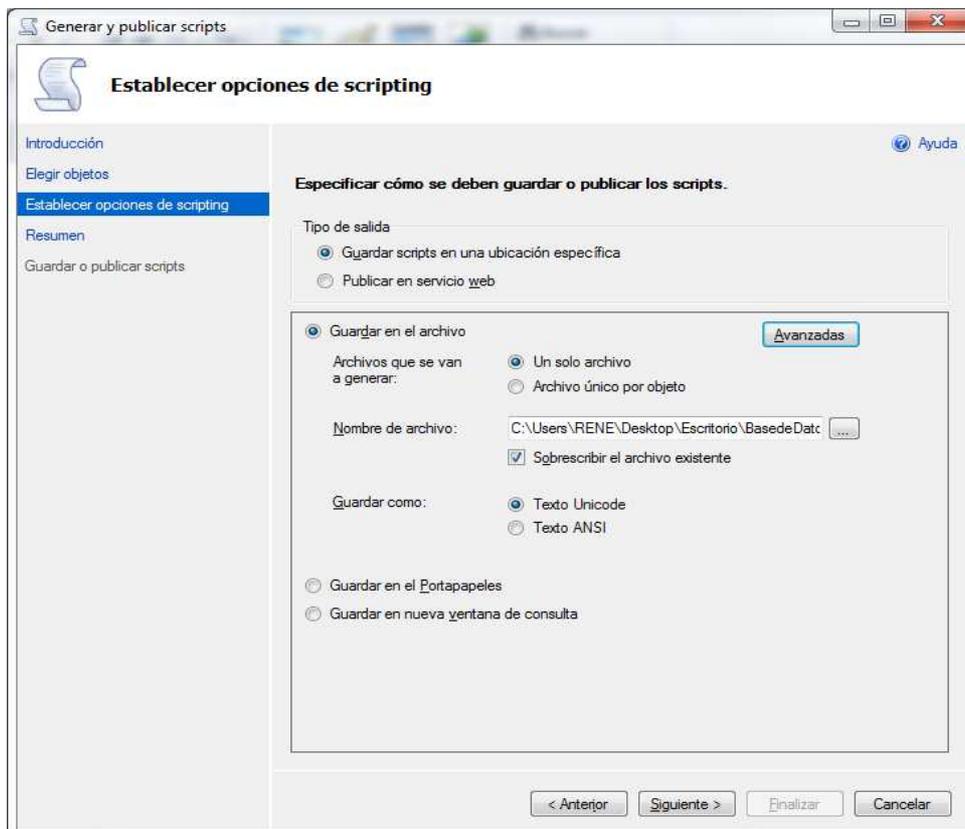


Figura IV.118: Elección de la ubicación del archivo del script.

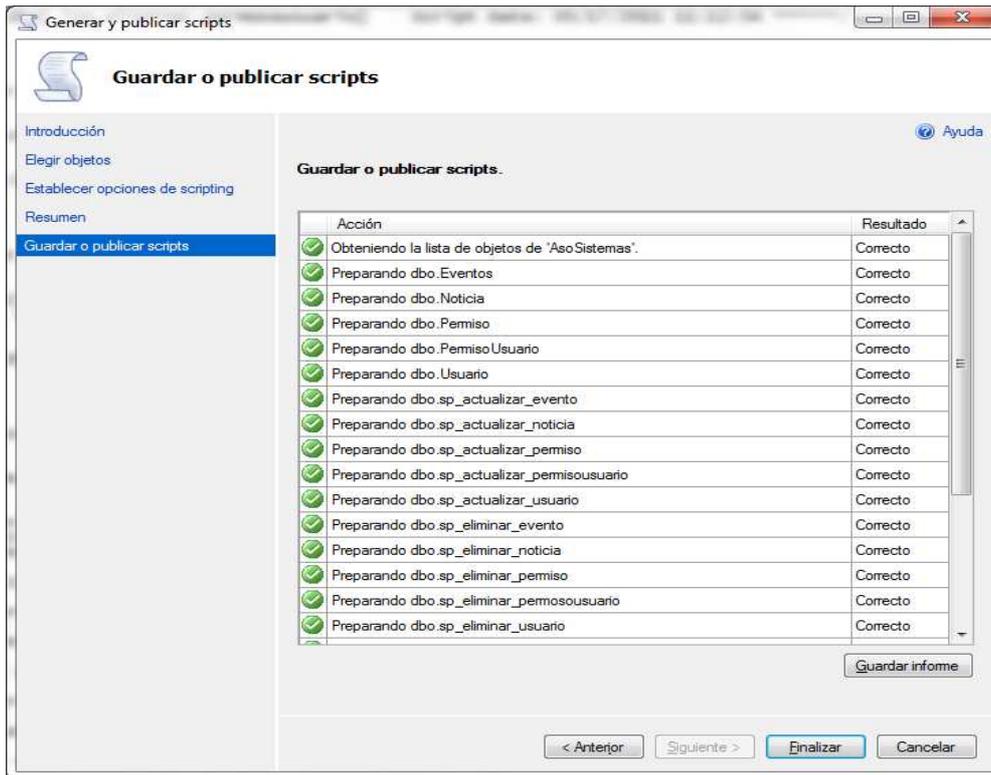


Figura IV.119: Estado de la creación de scripts.

Una vez generado el script, ahora si ejecutamos en SQL Azure, abriendo el archivo *.SQL

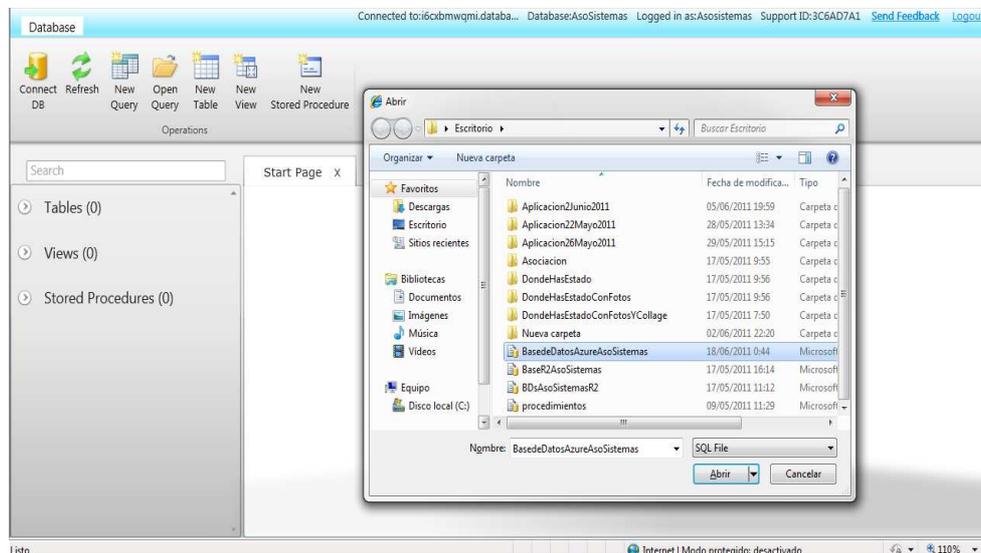


Figura IV.120: Abriendo el archivo *.SQL

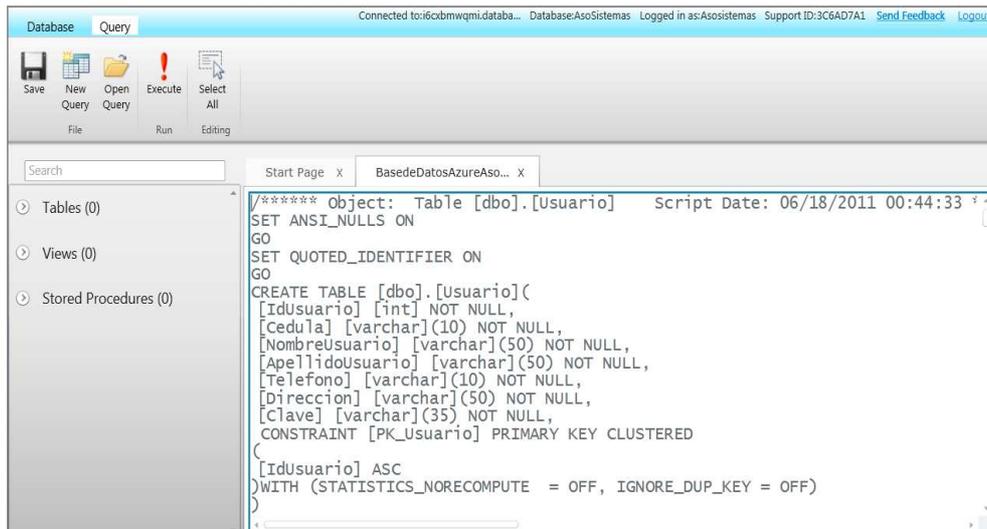


Figura IV.121: Archivo *.SQL

Clic en ejecutar y actualizar

Como ya se tiene creada la base de datos con todos sus componentes, la aplicación en .NET 2010 se necesita la cadena de conexión, para ello visualizamos la cadena de conexión

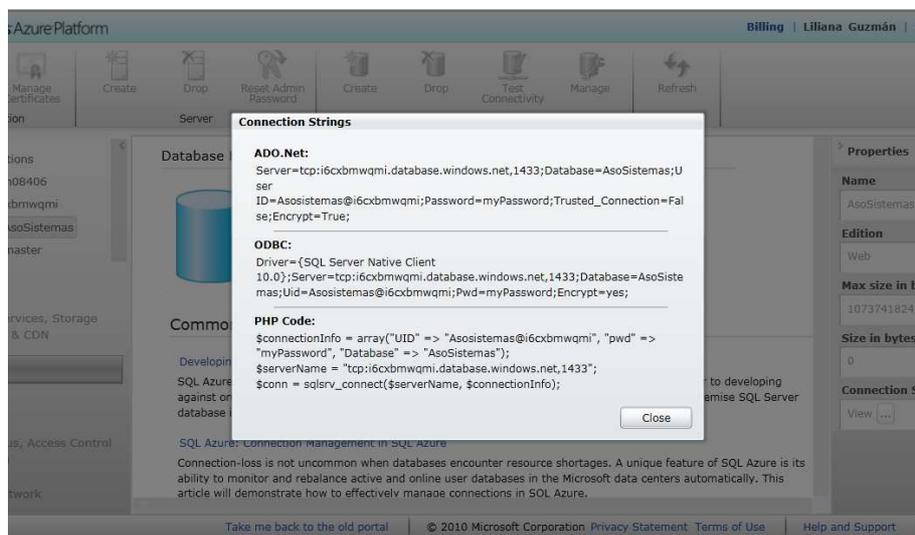


Figura IV.122: Cadena de conexión

Copiar esa clave y pegar en el proyecto de .NET 2010 en la clase clsConexion.cs

Al momento de actualizar el Dataset en Visual Studio .NET 2010

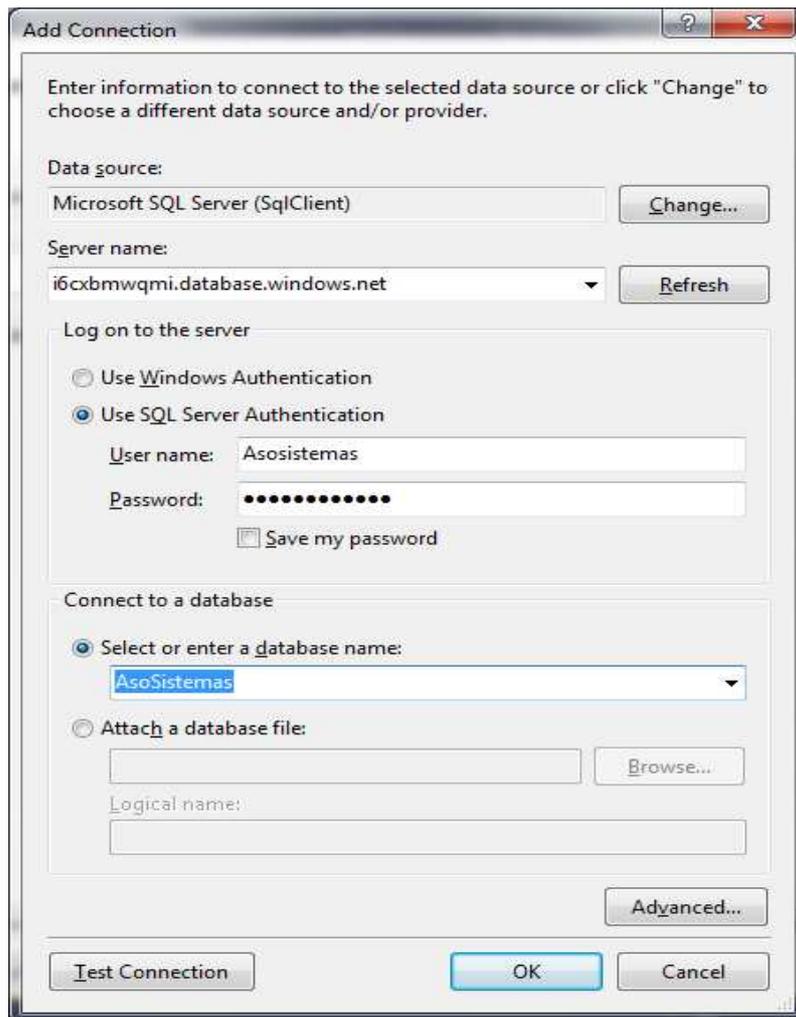


Figura IV.123: Añadiendo la conexión al dataset en .NET 2010

Una vez hecha la conexión se realiza el test para verificar que todo se encuentra satisfactoriamente y tenemos conexión a la Base de Datos del SQLAzure

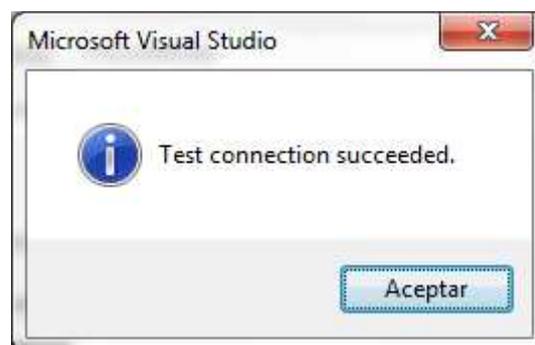


Figura IV.124: Test de conexión satisfactoria

Las pantallas siguientes muestran paso a paso para la nueva conexión con el dataset.

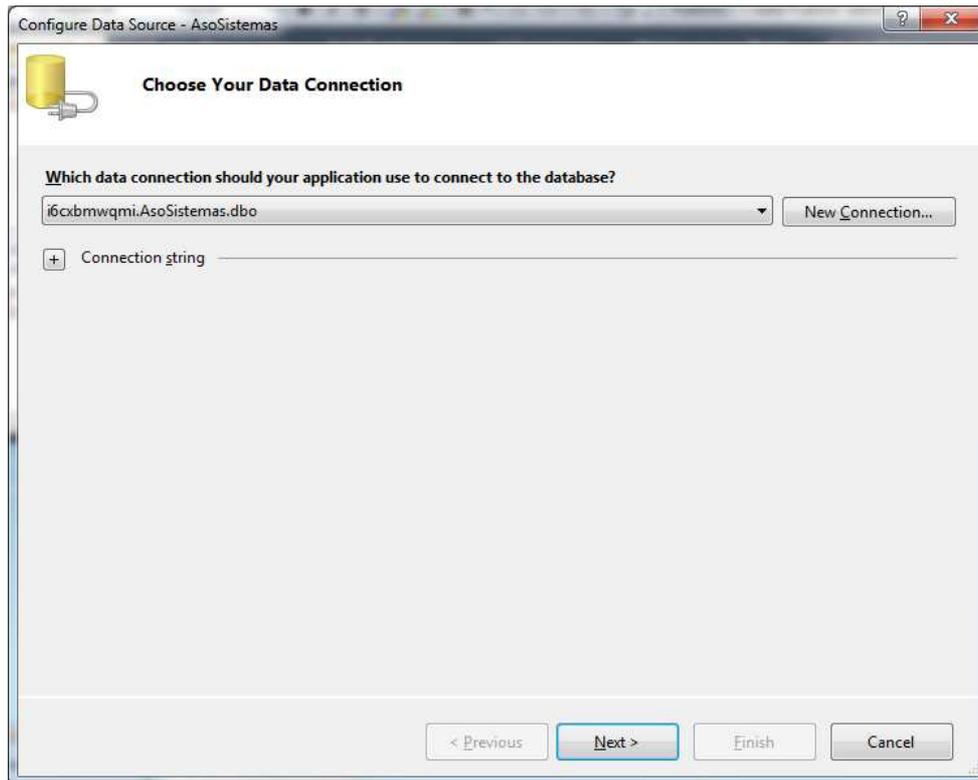


Figura IV.125: Creación de la nueva conexión.

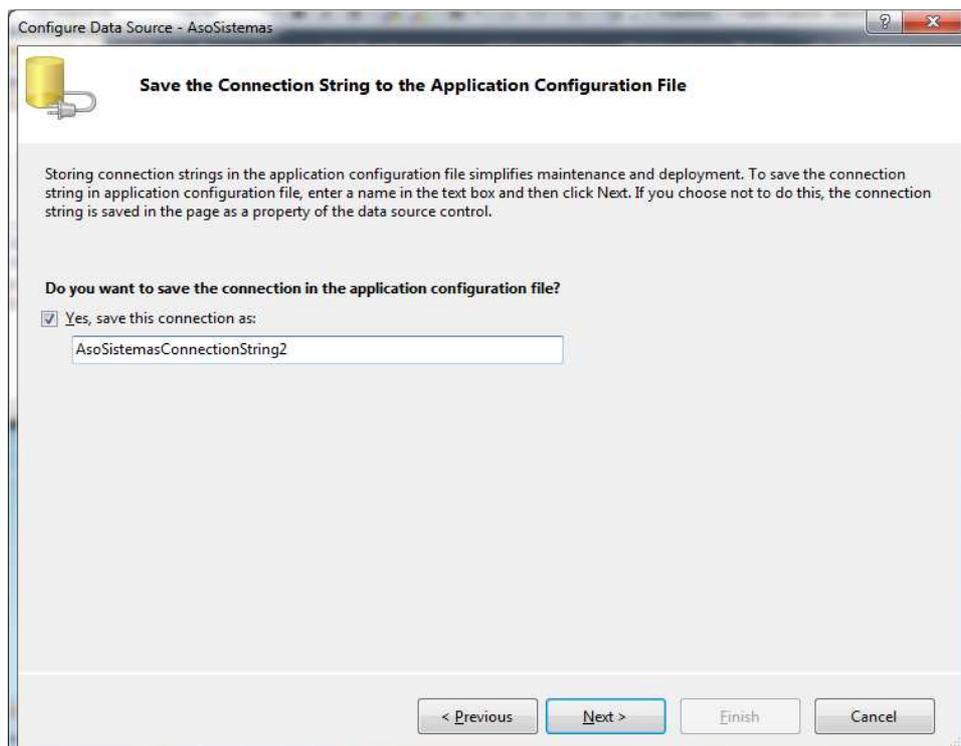


Figura IV.126: Nombre de la nueva conexión.

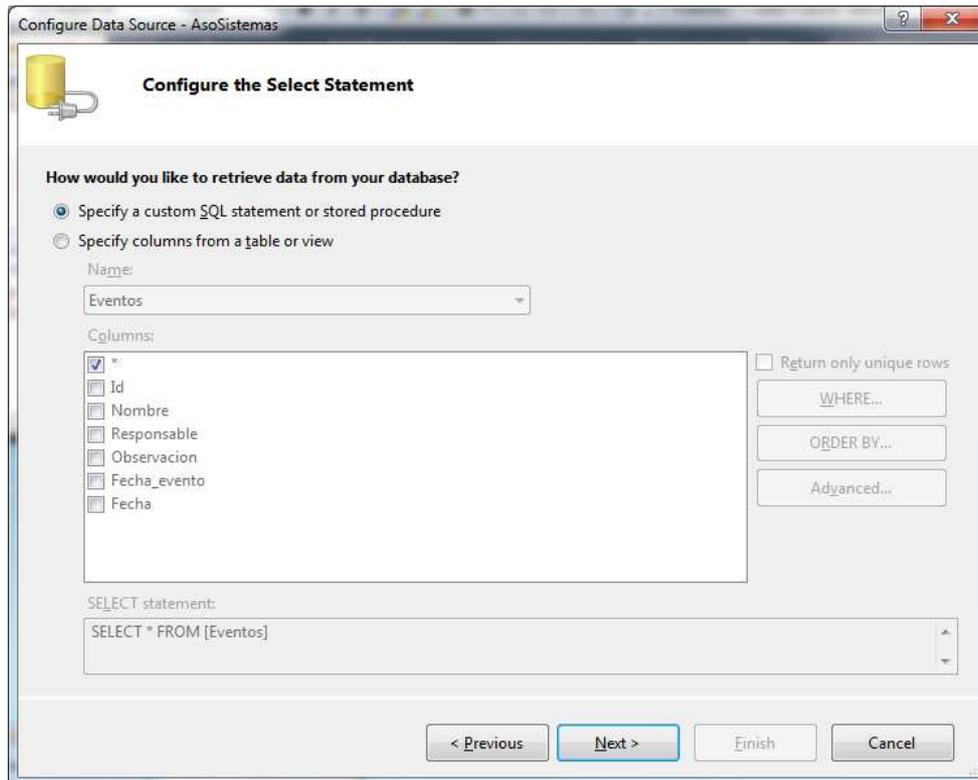


Figura IV.127: Opción para seleccionar el procedimiento almacenado.

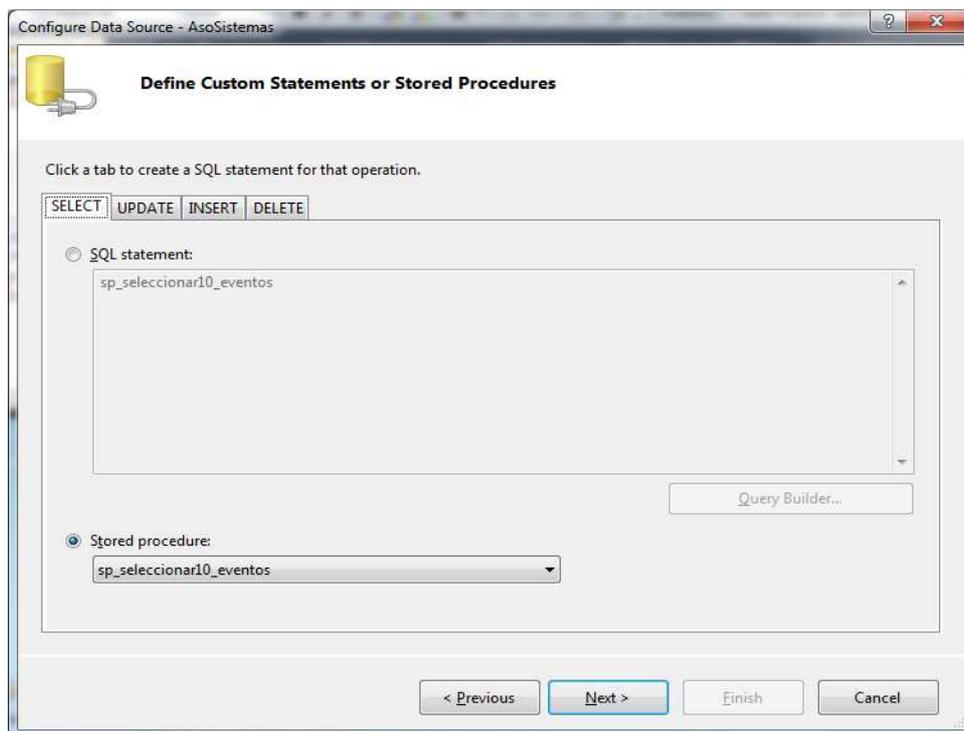


Figura IV.128: Seleccionando el procedimiento almacenado.

Clic en finish para terminar con la configuración del DataSource.

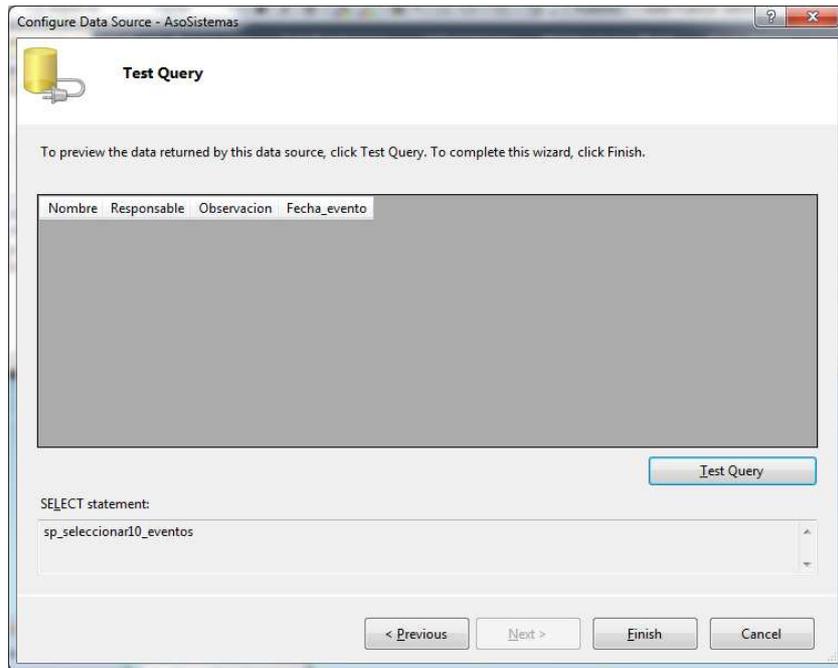


Figura IV.129: Clic en finish para terminar la configuración del DataSource

4.5.5. Creación de un nuevo Servidor de Hosted.

Para poder desplegar una aplicación de Windows Azure lo primero que se debe crear el paquete para la publicación en el .NET 2010

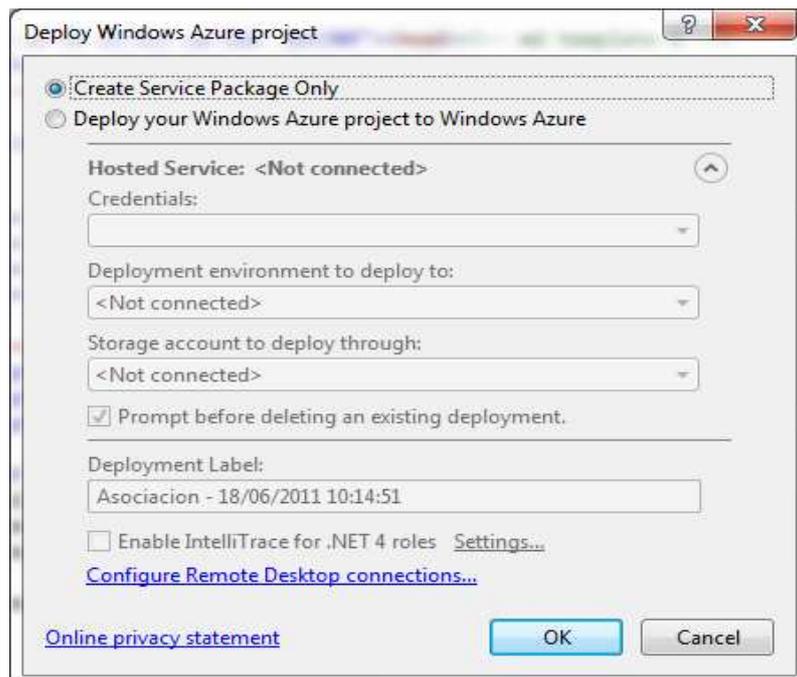


Figura IV.130: Creando el paquete para la publicación

Ahora en la pantalla de administración creamos un nuevo hostedservice.

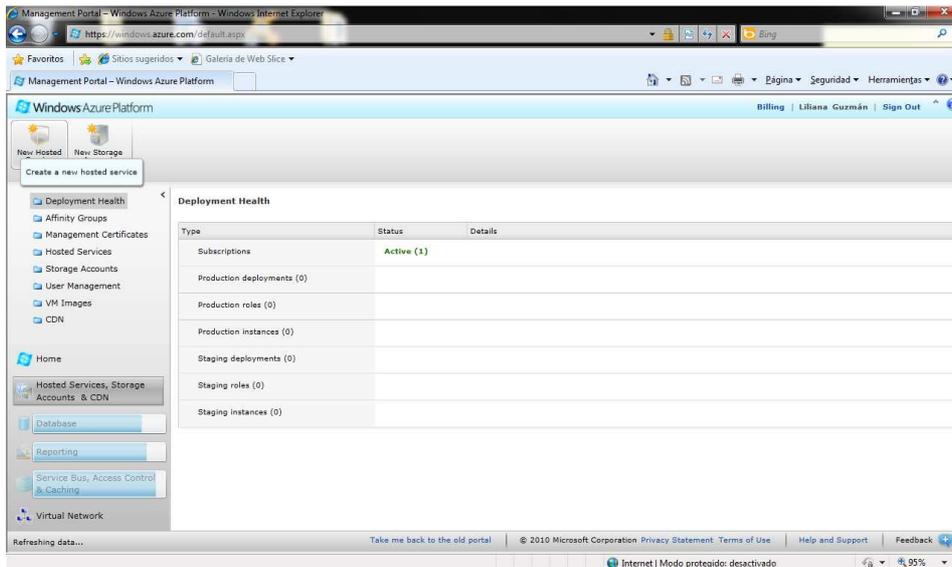


Figura IV.131: Administrando la creación del nuevo hosted service

Ingresar todos los datos para la creación, en este instante es cuando se sube los archivos que se generaron en el .NET 2010 es decir:

- El paquete y
- El archivo de configuración.

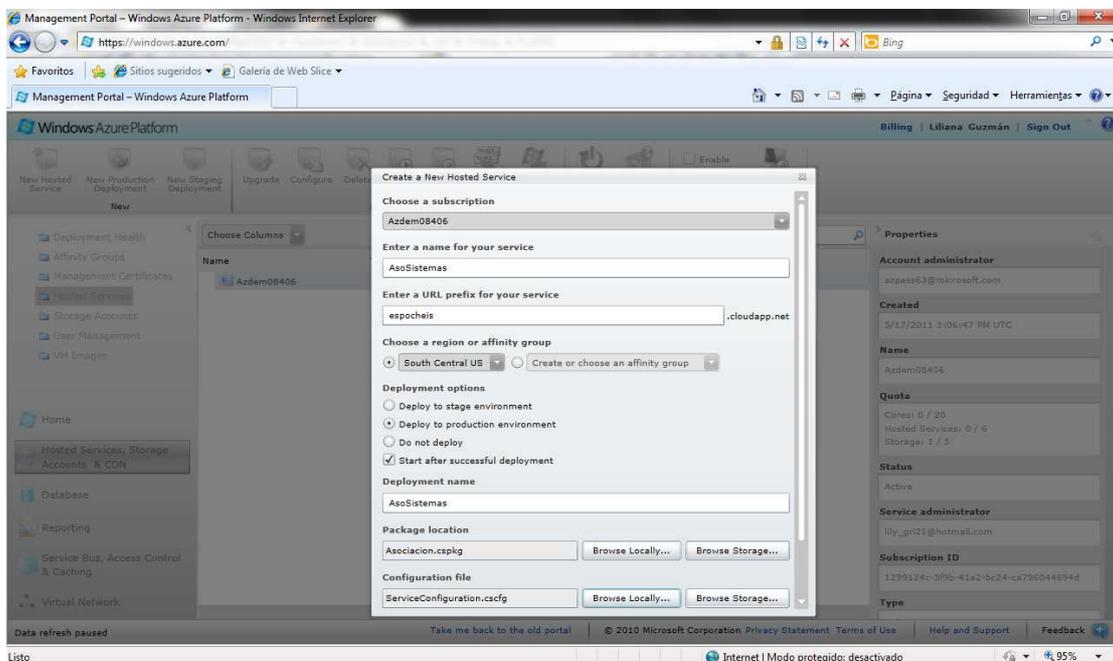


Figura IV.132: Datos para crear el hosted service

Clic en aceptar, pero tomar en cuenta que se generará una advertencia, por lo que se

recomienda dar clic en yes y continuar con el despliegue de la aplicación

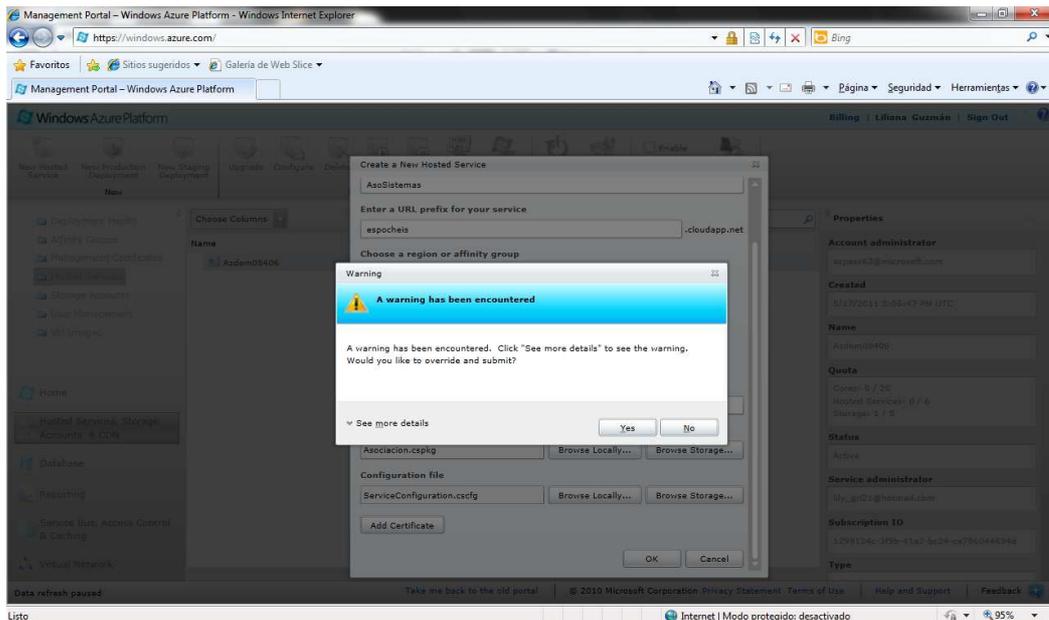


Figura IV.133: Precaución al momento de la creación del hosted

Una vez concluido con el ingreso de todos los datos y archivos esperar a que se despliegue en su totalidad al servidor de la plataforma de Windows Azure.

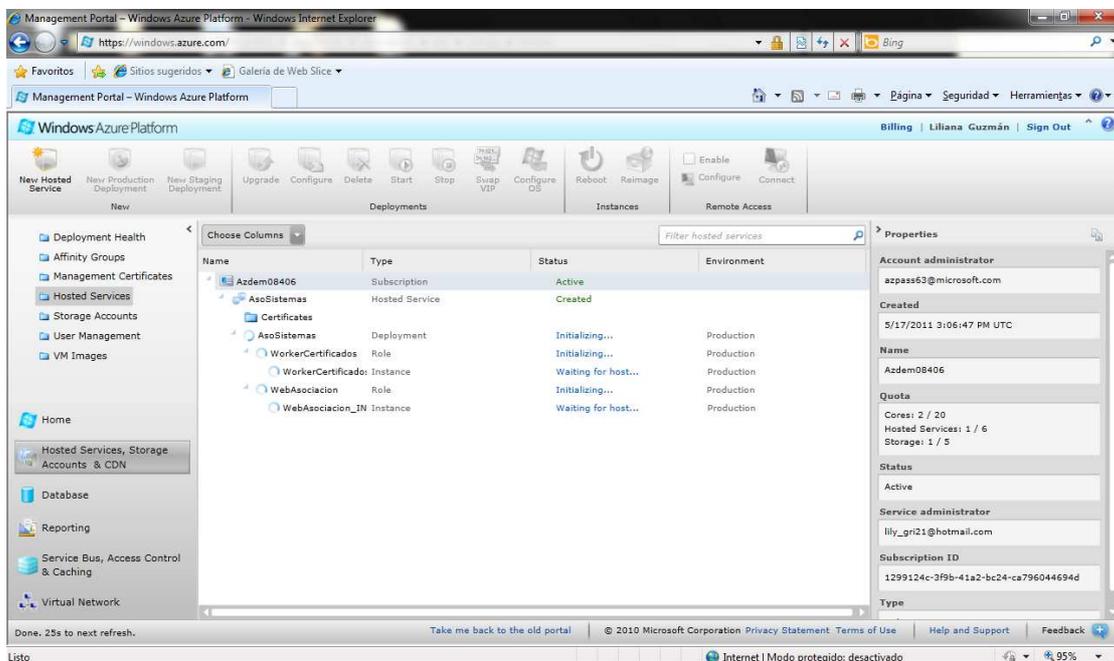


Figura IV.134: Subiendo la aplicación al Internet.

Una vez concluido el despliegue se observará en la pantalla que ya se tiene creado todos los servicios desplegados, tanto el web role como el worker role

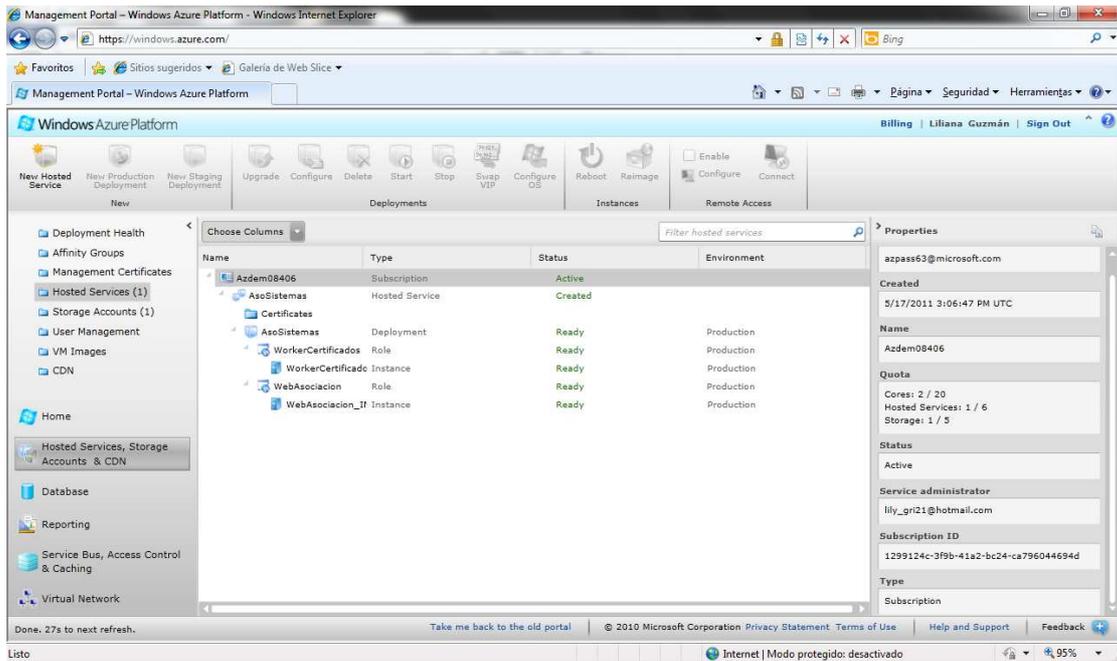


Figura IV.135: Terminando la subida al Internet de la aplicación

Una vez concluida se puede acceder a través de la siguiente URL:

<http://espocheis.cloudapp.net/>



Figura IV.136: Aplicación de la Asociación de Escuela

4.6. Pruebas del Sistema

La aplicación Web para la Asociación de Escuela de Ingeniería en Sistemas Informáticos se la ha publicado el 29 de Abril del 2011, cabe recalcar que se seguía mejorando día a día, por lo tanto todo ese tiempo se pasó realizando las pruebas del sistema y mejorando los errores encontrados.

La aplicación en su totalidad se encuentra terminada y publicada en la URL:

<http://espocheis.cloudapp.net/>

CAPÍTULO V

5. Informe Final

5.1. Hipótesis de Investigación.

Mediante el análisis la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH permitirá desplegar aplicaciones de consumo masivo en un corto tiempo y a bajo costo

5.2. Tipo de hipótesis.

El tipo de hipótesis es: Causa - Efecto

5.3. Determinación de las variables.

De acuerdo a la hipótesis se ha determinado dos variables:

Variable Independiente: Plataforma Windows Azure

Variable Dependiente: Despliegue de aplicaciones de consumo masivo

5.4. Operacionalización conceptual.

VARIABLE	TIPO	CONCEPTO
Plataforma Windows Azure	Complejo Cualitativo	La Plataforma Windows Azure es un servicio PaaS que forma parte de la oferta de servicios online de Microsoft. Proporciona un entorno familiar y flexible para desarrollar aplicaciones y servicios en la Nube. Con la Plataforma Windows Azure la empresa puede reducir el tiempo de lanzamiento de los productos y adaptarse fácilmente a medida que la demanda de éstos crezca.
Despliegue de aplicaciones de consumo masivo	Complejo Cuantitativa	-----

Tabla V.XXXI: Operacionalización Conceptual.

5.5. Operacionalización metodológica.

VARIABLE	CATEGORÍA	INDICADORES	TÉCNICAS	FUENTES DE VERIFICACIÓN
Plataforma	Requerimientos	<ul style="list-style-type: none"> Plataforma 	<ul style="list-style-type: none"> Utilización 	<ul style="list-style-type: none"> Internet

Windows Azure	de utilización	<ul style="list-style-type: none"> • Base de Datos • Herramienta de desarrollo 	directa <ul style="list-style-type: none"> • Revisión de documentos 	<ul style="list-style-type: none"> • Manuales • Cursos Interactivos
Despliegue de aplicaciones de consumo masivo		<ul style="list-style-type: none"> • Costo • Tiempo 	<ul style="list-style-type: none"> • Observación • Comparaciones del costo. 	<ul style="list-style-type: none"> • Reportes • Proyecciones en costos. • Comparación de tiempos utilizados.

Tabla V.XXXII: Operacionalización Metodológica.

5.6. Comprobación de la hipótesis de la investigación

Ahora mediante la valoración de los parámetros se definirán las diferencias de trabajar en entornos Cloud y en servidores locales institucionales, para aceptar o negar la hipótesis

Por lo tanto antes definiremos las hipótesis:

Hi: Mediante el análisis la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH permitirá desplegar aplicaciones de consumo masivo en un corto tiempo y a bajo costo

Ho: Mediante el análisis la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH no permitirá desplegar aplicaciones de consumo masivo en un corto tiempo y a bajo costo

5.7. Comprobación en cuanto a Costos de Windows Azure vs Servidores Locales

Una vez realizado el análisis en el Capítulo III, sección 3.3 en cuanto a costo se comprueba de forma satisfactoria la hipótesis ya que como se observa en la siguiente figura, desplegar la aplicación en la Plataforma Windows Azure es más económico que desplegar en los servidores institucionales.

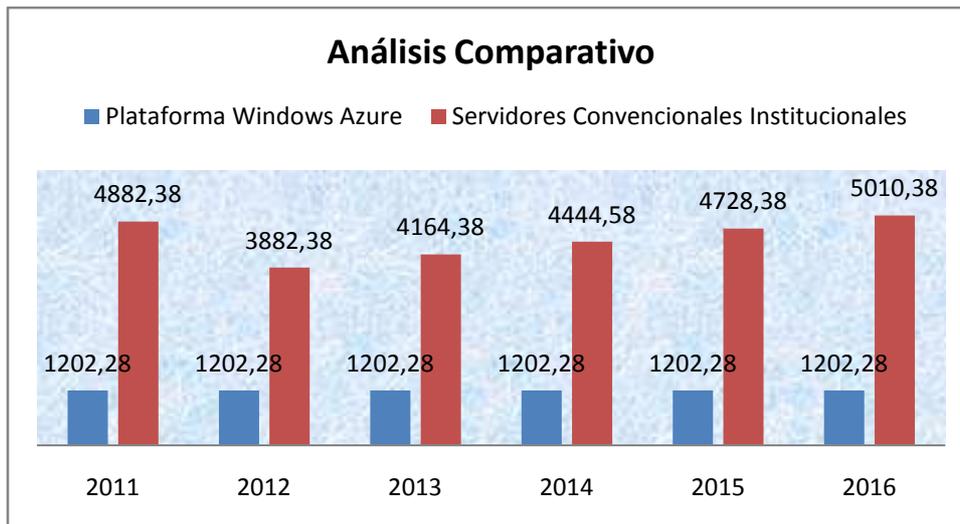


Figura V.137: Conclusiones en costos

5.8. Comprobación del Tiempo de Windows Azure vs Servidores Locales

Una vez realizado el análisis en el Capítulo III, sección 3.6 en cuanto al tiempo en el despliegue de la aplicación se determina claramente que se comprueba de forma satisfactoria la hipótesis ya que como se observó, desplegar la aplicación en la Plataforma Windows Azure se realiza un tiempo mucho menor que desplegar en los servidores institucionales.

5.9. Comprobación de la Hipótesis

Después de realizar el estudio y análisis comparativo se puede definir que con la utilización de la Plataforma Windows Azure en el despliegue de las aplicaciones en entornos Cloud el costo y tiempo es menor que en servidores locales institucionales, por lo tanto se acepta y se comprueba la hipótesis de investigación que dice: H_1 : Mediante el análisis la plataforma Windows Azure para trabajar en entornos Cloud y su aplicación en la ESPOCH permitirá desplegar aplicaciones de consumo masivo en un corto tiempo y a bajo costo.

CONCLUSIONES

- Al realizar el Análisis de la Plataforma Windows Azure para trabajar en entornos Cloud y la correcta instalación de sus herramientas, permitió el desarrollo de la presente aplicación “Asociación Sistemas”.
- La instalación de las herramientas propias de Windows Azure se realizaron con toda normalidad, obteniendo como resultados la facilidad para poner en marcha un servidor similar a los propios de la plataforma.
- El desarrollo de las aplicaciones en la Plataforma son en particular muy sencillos, recalcando que se necesita lógica de programación y conocimientos básicos del Framework 4.0 y SQL Server 2008 R2.
- La Plataforma Windows Azure tiene mucha documentación y ayuda por lo que no es complicado encontrar respuestas en caso de alguna duda durante el desarrollo.
- La tendencia del Internet cada día se está plasmando más en todo el mundo, razón por la cual desarrollar aplicaciones destinadas a la Nube es una muy buena opción.
- Se implementó una aplicación Web de información para la Asociación de Escuela de Ingeniería en Sistemas bajo la plataforma Windows Azure.
- El análisis del costo y el tiempo en el despliegue de la aplicación de la Asociación de Escuela de Ingeniería en Sistemas demostró que la utilización de la Plataforma Windows Azure es la mejor opción por los resultados que se produjo.

RECOMENDACIONES

- Se recomienda el aprendizaje y utilización de esta Plataforma y el despliegue de sus aplicaciones ya sea tanto a nivel empresarial como a nivel estudiantil, por lo que los recursos necesarios son mínimos y de gran utilidad.
- Se recomienda crear una nueva materia optativa para el uso de la Plataforma Windows Azurey así los estudiantes de la Escuela de Ingeniería en Sistemas estarían actualizados en cuanto a esta nueva tendencia.~~al tanto en lo que a tecnología se trata.~~
- Se recomienda que esta tesis puede mejorarse desplegando más aplicaciones tanto de prueba como de producción y así conocer más de sus ventajas, como puede ser a la hora del rendimiento.
- Se recomienda que el Estudiante de Ingeniería en Sistemas debe adquirir conocimientos para desplegar aplicaciones en la Nube.
- Como Ingenieros en Sistemas es muy recomendable crear y programar aplicaciones web alojadas en la Nube con esto se podría prescindir de gran parte de los costos asociados a la operación y despliegue de la aplicación.

RESUMEN

El objetivo de ésta investigación fue el Análisis de la plataforma Windows Azure en entornos Cloud y su aplicación en la Escuela Superior Politécnica de Chimborazo.

El método utilizado para la presente investigación fue el método analítico e investigativo, el sistema informático “Asociación Sistemas” fue creado para comparar los costos y tiempo del despliegue entre un entorno Cloud y una aplicación tradicional, además para el desarrollo se utilizó una computadora portátil HP Pavilion DV4 1280 US, VMware Workstation 7, Windows 7, Visual Studio .NET C# 2010, SQL Server 2008 R2 y SDK propios de la plataforma.

Mediante el estudio, la creación de la aplicación y su posterior análisis comparativo del despliegue se logró establecer que el desarrollo para entornos Cloud disminuye los costos en un 60% y el tiempo con un valor de 80%.

Como conclusión se puede decir que el despliegue en los servidores de Windows Azure es mejor y más productivo en lugar de realizarlo en los servidores locales, ya que se garantiza el correcto uso de los recursos en especial del más importante que es el tiempo.

Se recomienda el aprendizaje y utilización de esta Plataforma y el despliegue de sus aplicaciones ya sea tanto a nivel empresarial como a nivel estudiantil, por lo que los recursos necesarios son mínimos y de gran utilidad.

SUMMARY

The objective of this investigation was the analysis of the Windows Azure Platform in Cloud surroundings and its Application at the Chimborazo Higher Education Polytechnic School.

The method used for the present investigation was the analytical and investigative method. The informatics system "Systems Association" was created to compare costs and display time between the cloud surrounding and a traditional application; moreover, for the development a portable computer HP Pavilion DV4 1280 US, VMware Workstation 7, Windows 7, Visual Studio .NET c# 2010, SQL Server 2008 R2 and SDK of the platform were used.

Through the study and creation of the application and its further comparative display analysis it was possible to establish that the development for the cloud surroundings diminishes costs by 60% and time with a value of 80%.

As a conclusion it can be said that the display in the Windows Azure display servers is better and more productive, instead of carrying it out in the local servers, as it guarantees the correct use of resources and specially of the most important one which is, time.

ANEXO 1

Parte 1

Encuesta dirigida al Director de DESITEL Dr. Carlos Buenaño.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA INGENIERÍA EN SISTEMAS

TEMA: Análisis de la Plataforma Windows Azure para trabajar en entornos cloud y su aplicación en la ESPOCH

OBJETIVO: Conocer la manera en la que se despliega una aplicación tradicional en los servidores institucionales de la Escuela Superior Politécnica de Chimborazo.

JUSTIFICACIÓN: El motivo de la encuesta es conocer cómo se manejan la opción de permisos para el despliegue de una aplicación en los servidores institucionales de la Escuela Superior Politécnica de Chimborazo, los recursos de personal, disponibilidad de hardware y servicios necesarios, además que los costos en los que incurriría dicho pedido y el tiempo que se tarda poder desplegar para su producción la aplicación.

INSTRUCCIONES: Con un \surd responda las preguntas que se detallan a continuación.

CUESTIONARIO

1. ¿Utilizando los servidores institucionales administrador por la Escuela de Ingeniería en Sistemas se puede desplegar una aplicación web diseñada para la Asociación de Escuela, teniendo en cuenta que cuenta con un servidor Proxy?

Si

No

2. ¿Una aplicación web que utiliza el Framework .NET 4.0, base de datos SQL Server 2008 R2 puede ser desplegada sin ningún problema?

Si

No

3. Por favor detalle los trámites que se debe realizar por parte de las personas interesadas para desplegar la aplicación.

.....
.....
.....
.....
.....
.....
.....

4. ¿Cada persona de la Escuela de Ingeniería en Sistemas encargada de los servidores de la misma debe tener ya sus responsabilidades establecidas razón por lo cual, si se desea desplegar una aplicación web es necesario contratar más personal?

Si

No

5. ¿Cuál sería el costo en el que se incurriría por desplegar una nueva aplicación y porque?

.....
.....
.....

6. ¿Cuánto tiempo estima Usted que se llevaría el despliegue de la aplicación en el caso de que no exista ningún inconveniente?

1 a 2 días laborables 3 a 5 días laborables

2 a 3 semanas 1 mes

Otro tiempo

7. ¿Cuánto tiempo estima Usted que se llevaría el despliegue de la aplicación en el caso de que exista inconvenientes (tome en cuenta autorizaciones, compra de nuevos servidores, incompatibilidad de tecnología, licencias, sueldos y salarios etc.)?

1 a 2 semanas 3 a 4 semanas

2 a 3 meses 4 a 5 meses

Otro tiempo

8. ¿Aumentaría el hecho de publicar una aplicación nueva en el valor presupuestado para la Escuela de Ingeniería en Sistemas?

Si No

9. ¿Una vez desplegada o publicada la aplicación cuales son los siguientes pasos a realizar para que siga manteniéndose en producción?

.....
.....
.....
.....

10. ¿En caso de no contar con la disponibilidad de los servidores institucionales administrador por la Escuela de Ingeniería en Sistemas, cuales son los pasos necesarios para poder desplegar la aplicación, por favor escriba lo más detallado posible?

.....
.....
.....
.....
.....

11. ¿Existen aplicaciones ejecutándose actualmente en los servidores de la Escuela de Ingeniería en Sistemas?

Si No

12. ¿Cuántas y cuáles son las aplicaciones que se encuentran actualmente desplegadas en los servidores institucionales?

.....
.....
.....
.....
.....
.....

13. ¿En qué valor oscila el mantenimiento anual de cada aplicación que se encuentra publicada (Por favor detalle lo más que se pueda)?

.....
.....
.....
.....
.....

14. Las siguientes preguntas solo son para información:

a. Conoce Usted el costo de las licencias para SQL Server 2008.

Si No Valor:

b. Conoce Usted el costo de las licencias para utilizar el IDE Visual Studio.

Si No Valor:

c. ¿Cuál es el número de personal encargado de los Servidores de la Escuela de Ingeniería en Sistemas?

.....

d. ¿Conoce Usted cuál es el rango que oscila el sueldo del personal de los encargados de los Servidores de la Escuela de Ingeniería en Sistemas?

.....

e. ¿Por favor detalle la infraestructura con la que cuenta la institución el cual se encuentra administrada por la Escuela de Ingeniería en Sistemas?

.....
.....
.....
.....
.....

f. ¿Ante algún percance de carácter físico como se resuelven los problemas?

.....
.....
.....
.....

GRACIAS POR SU COLABORACIÓN

Parte 2

Encuesta dirigida al Técnico de los Laboratorios de la Escuela de Ingeniería en Sistemas Ing. Carlos Jara.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA INGENIERÍA EN SISTEMAS

TEMA: Análisis de la Plataforma Windows Azure para trabajar en entornos cloud y su aplicación en la ESPOCH

OBJETIVO: Conocer la manera en la que se despliega una aplicación tradicional en los servidores institucionales de la Escuela Superior Politécnica de Chimborazo.

JUSTIFICACIÓN: El motivo de la encuesta es conocer cómo se manejan la opción de permisos para el despliegue de una aplicación en los servidores institucionales de la Escuela Superior Politécnica de Chimborazo, los recursos de personal, disponibilidad de hardware y servicios necesarios, además que los costos en los que incurriría dicho pedido y el tiempo que se tarda poder desplegar para su producción la aplicación.

INSTRUCCIONES: Con un \surd responda las preguntas que se detallan a continuación.

CUESTIONARIO

1. ¿Utilizando los servidores institucionales administrador por la Escuela de Ingeniería en Sistemas se puede desplegar una aplicación web diseñada para la Asociación de Escuela, teniendo en cuenta que cuenta con un servidor Proxy?

Si

No

2. ¿Una aplicación web que utiliza el Framework .NET 4.0, base de datos SQL Server 2008 R2 puede ser desplegada sin ningún problema?

Si

No

3. Por favor detalle los trámites que se debe realizar por parte de las personas interesadas para desplegar la aplicación.

.....
.....
.....
.....

4. ¿Cada persona de la Escuela de Ingeniería en Sistemas encargada de los servidores de la misma debe tener ya sus responsabilidades establecidas razón por lo cual, si se desea desplegar una aplicación web es necesario contratar más personal?

Si

No

5. ¿Cuál sería el costo en el que se incurriría por desplegar una nueva aplicación y porque?

.....
.....
.....

6. ¿Cuánto tiempo estima Usted que se llevaría el despliegue de la aplicación en el caso de que no exista ningún inconveniente?

1 a 2 días laborables 3 a 5 días laborables

2 a 3 semanas 1 mes

Otro tiempo

7. ¿Cuánto tiempo estima Usted que se llevaría el despliegue de la aplicación en el caso de que exista inconvenientes (tome en cuenta autorizaciones, compra de nuevos servidores, incompatibilidad de tecnología, licencias, sueldos y salarios etc.)?

1 a 2 semanas 3 a 4 semanas

2 a 3 meses 4 a 5 meses

Otro tiempo

8. ¿Aumentaría el hecho de publicar una aplicación nueva en el valor presupuestado para la Escuela de Ingeniería en Sistemas?

Si

No

9. ¿Una vez desplegada o publicada la aplicación cuales son los siguientes pasos a realizar para que siga manteniéndose en producción?

.....
.....
.....
.....

10. ¿En caso de no contar con la disponibilidad de los servidores institucionales administrador por la Escuela de Ingeniería en Sistemas, cuales son los pasos necesarios para poder desplegar la aplicación, por favor escriba lo más detallado posible?

.....
.....
.....
.....
.....

11. ¿Existen aplicaciones ejecutándose actualmente en los servidores de la Escuela de Ingeniería en Sistemas?

Si No

12. ¿Cuántas y cuáles son las aplicaciones que se encuentran actualmente desplegadas en los servidores institucionales?

.....
.....
.....

13. ¿En qué valor oscila el mantenimiento anual de cada aplicación que se encuentra publicada (Por favor detalle lo más que se pueda)?

.....
.....
.....

14. Las siguientes preguntas solo son para información:

a. Conoce Usted el costo de las licencias para SQL Server 2008.

Si No Valor:

b. Conoce Usted el costo de las licencias para utilizar el IDE Visual Studio.

Si No Valor:

c. ¿Cuál es el número de personal encargado de los Servidores de la Escuela de Ingeniería en Sistemas?

.....

d. ¿Conoce Usted cuál es el rango que oscila el sueldo del personal de los encargados de los Servidores de la Escuela de Ingeniería en Sistemas?

.....

e. ¿Por favor detalle la infraestructura con la que cuenta la institución el cual se encuentra administrada por la Escuela de Ingeniería en Sistemas?

.....
.....
.....
.....
.....
.....

f. ¿Ante algún percance de carácter físico como se resuelven los problemas?

.....
.....
.....

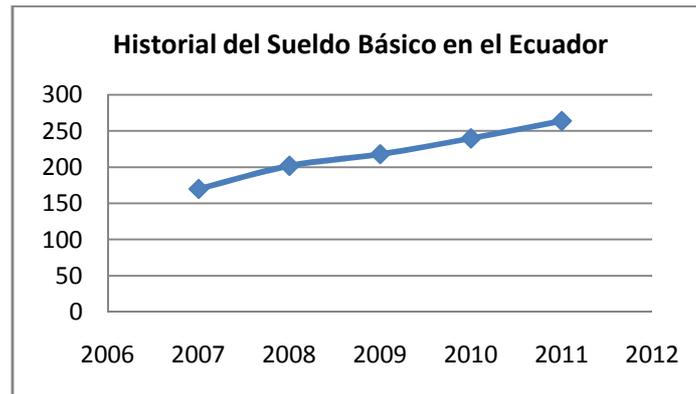
GRACIAS POR SU COLABORACIÓN

ANEXO 2

Parte 1

Sueldos y Salarios en el Ecuador

2007	2008	2009	2010	2011
170	202	218	240	264
US\$/mes	US\$/mes	US\$/mes	US\$/mes	US\$/mes



Ecuación de la recta:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

$$y - 170 = \frac{264 - 170}{2011 - 2007}(x - 2007)$$

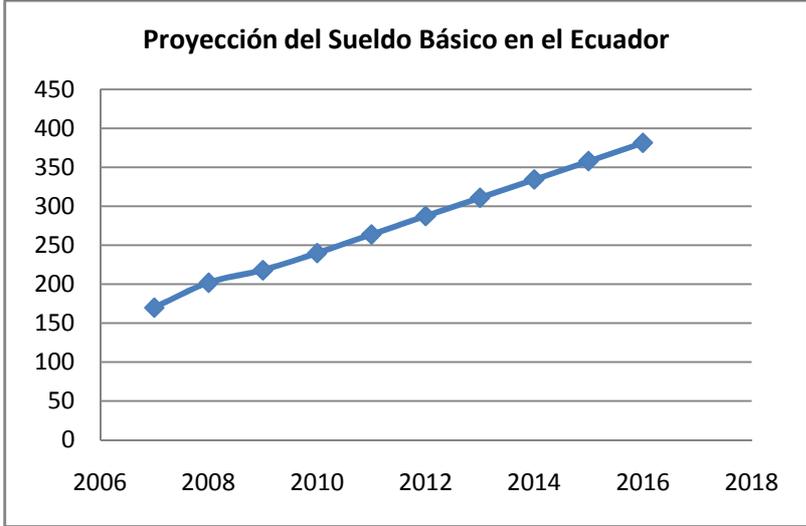
$$y - 170 = 23,5(x - 2007)$$

$$\underline{\underline{y = 23,5x - 46994,5}}$$

Con la ecuación encontrada se procede a proyectar el Salario Básico a los 5 años.

X	Y
2007	170,00
2008	202,00
2009	218,00
2010	240,00
2011	264,00
2012	287,50
2013	311,00
2014	334,50

2015	358,00
2016	381,50



ANEXO 3

MANUAL DEL USUARIO

AUTORES:

René Villa

Liliana Guzmán

La información en este documento es propiedad de los Autores (producción) Asociación Sistemas sujeto a normas legales, no se admite su reproducción total o parcial, pudiendo ser utilizado con el fin de guiarse en el uso del sitio Web de Asociación Sistemas y darle el mejor de los usos a la información que este sistema provee a sus usuarios

CAPÍTULO 1 Presentación

1.1 Presentación

EL sitio creado para la Asociación de Sistemas es una aplicación WEB, la misma que fue diseñada con el fin de proveer información a cerca de la ASOCIACIÓN DE LA ESCUELA DE INGENIERÍA EN SISTEMAS DE LA ESPOCH como es: Infraestructura de la escuela, Información de Eventos, Galería de fotos, Modelo de oficios, Noticias importantes.

Los datos presentados en este sitio Web permite difundir las actividades y fines para los cuales fue creado la página “Asociación Sistemas”, además que permite mantener mejor informado a los estudiantes de la escuela de las actividades, noticias y novedades de la Asociación de Sistemas,

CAPÍTULO 2: Instrucciones

2.1 NAVEGACIÓN EN WEB “Asociación Sistemas”

1. Para que usted pueda acceder al portal de “Asociación Sistemas” digite en la barra de direcciones de su navegador la siguiente URL:

<http://www.espocheis.cloudapp.net>

2. Una vez que haya realizado el paso anterior se le presentará la siguiente página Web en la cual para que usted pueda acceder a más información de “Asociación Sistemas” por defecto usted ingresa al módulo del usuario y si desea administrar la página ingrese sus credenciales en la parte superior.



2.2 Modulo de Usuario

Como se dijo anteriormente se inicia por defecto como usuario normal, y si desea administrar el sitio debe ingresar sus credenciales.

2.2.1 Pagina de Misión Visión Historia y Logros

Si desea ver cuál es la misión visión y logros de la Asociación de Sistemas presione



una vez que presione este botón, la información de misión visión y logros se desplegara en el centro del portal como se muestra a continuación.



2.2.2 Páginas de Infraestructura

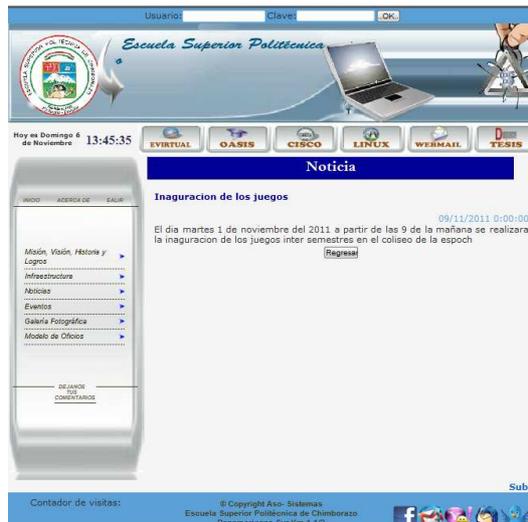
Para acceder a las imágenes de la infraestructura de la escuela de ingeniería en sistemas presione  una vez realizada esta acción aparecerá una galería de fotos de la infraestructura podemos seleccionar una y se puede navegar presionando Next y Previous.



2.2.3 Noticias

Para ver las últimas noticias que ocurre alrededor de la facultad presione  a continuación se presentará en el centro del portal una página que contiene información sobre las últimas noticias relevantes.





2.2.4 Eventos

Para ver los eventos a realizarse en la ESPOCH presione  a continuación se presentara en el centro del portal una página que contiene información sobre los eventos que se realizaran en la Politécnica.





2.2.5 Galería Fotográfica

Para que usted pueda ver o desplegar las fotografías presione  y se desplegara la página de galería fotográfica donde podemos subir fotos.



2.2.6 Modelo de Oficinas

Si desea ver algún modelo de oficinas que desee realizar presione el botón  y continuación se le desplegara un listado de oficinas el cual se puede dar clic en el hipervínculo para que maximice la fotografía y se pueda ver el modelo de oficina.



2.2.7 Déjanos tus comentarios:

Si desea dejar algún comentario presione el botón  y continuación se le desplegara un formulario en donde puede ingresar su comentario.



2.3 Modulo de Administración:

Si usted es un usuario Administrador debe ingresar sus credenciales en el formulario que se encuentra en la parte superior de la pantalla, y presionar el botón OK.





Para poder ingresar los datos de misión visión y logros damos clic en el link



y se desplegara la siguiente información.



2.3.1 Noticias

Si desea ingresar una nueva noticia en "Asociación sistemas" se debe dar click en el



hipervínculo llenar los campos y pulsar en ingresar noticia.

Escuela Superior Politécnica De Chimborazo
Escuela De Ingeniería En Sistemas

Hoy es Domingo 6 de Noviembre 15:37:12

EVIRTUAL OASIS CISCO LINUX WEBMAIL TESIS

Administrando las Noticias

[Click Para ingresar una nueva NOTICIA](#)

Titulo	Fecha	Actualizar	Eliminar
Denuncias en la FEPOCH	30/10/2011 0:00:00	Actualizar	Eliminar
Inauguración de los juegos	09/11/2011 0:00:00	Actualizar	Eliminar

Subir

Contador de visitas:

© Copyright Aso- Sistemas
Escuela Superior Politécnica de Chimborazo
Panamericana Sur Km 1 1/2
All rights reserved 2011

Escuela Superior Politécnica De Chimborazo
Escuela De Ingeniería En Sistemas

Hoy es Domingo 6 de Noviembre 15:38:26

EVIRTUAL OASIS CISCO LINUX WEBMAIL TESIS

Nueva Noticia

Titulo:

Fecha:

Contenido:

[Ingresar Noticia](#)

Subir

Contador de visitas:

© Copyright Aso- Sistemas
Escuela Superior Politécnica de Chimborazo
Panamericana Sur Km 1 1/2
All rights reserved 2011

Escuela Superior Politécnica De Chimborazo
Escuela De Ingeniería En Sistemas

Hoy es Domingo 6 de Noviembre 15:50:10

EVIRTUAL OASIS CISCO LINUX WEBMAIL TESIS

Actualizar Noticia

Id:

Titulo:

Fecha:

Contenido:

[Actualizar](#)

Subir

Contador de visitas:

© Copyright Aso- Sistemas
Escuela Superior Politécnica de Chimborazo
Panamericana Sur Km 1 1/2
All rights reserved 2011

2.3.2 Eventos

Si desea ingresar un nuevo evento en "Asociación sistemas" se debe dar click en el hipervínculo  llenar los campos y pulsar en ingresar eventos.



Nombre	Responsable	Contenido	Fecha del Evento	Fecha Actual	Actualizar	Eliminar
Bienvenida estudiantes de primer semestre	Asociacion de Escuela	En las aulas nuevas del Edificio de Sistemas se va a dar la Bienvenida a todos los estudiantes nuevos de la Escuela de Ingeniería en Sistemas, augurando éxitos en su vida estudiantil	04/10/2011 0:00:00	04/10/2011 0:00:00	Actualizar	Eliminar
Recorrido por las Instalaciones	Ing. Raul Rosero	Por parte del director de Escuela se va a realizar el recorrido a toda la Politécnica con la finalidad de dar la	04/10/2011 0:00:00	04/10/2011 0:00:00	Actualizar	Eliminar

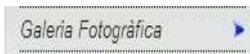


Nombre:
Responsable:
Fecha del Evento:
Fecha Actual:
Contenido:



2.3.3 Galería Fotográfica

Para que usted pueda acceder a la modificación de las fotografías presione

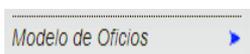


en donde se podrá agregar o eliminar fotos y el resultado se desplegara inmediatamente.



2.3.4 Modelo de Oficios

Si desea ver o ingresar o eliminar un nuevo modelo de oficio en el botón



y continuación se le desplegara una pantalla en donde se muestra los modelos ya ingresados y la opción para ingresar otro modelo de oficio.

Para ingresar a las páginas de la ESPOCH a continuación presione en uno de los botones que se encuentran en la parte inferior de la aplicación.



Soporte al Cliente

Telefónico

Para cualquier inquietud sobre el funcionamiento de esta página puede llamar a los siguientes números: 093228756, 069166249

E- mail

Usted puede enviar un e- mail a: lily_gri@hotmail.com, memo_re@yahoo.es. Bajo las circunstancias normales usted recibirá una contestación automatizada dentro de 15 minutos, detallando las soluciones más comunes de los problemas. Usted recibirá una solución más detallada a su problema particular o pregunta.

GLOSARIO

APLICACIÓN: Programa diseñado para ayudar en la ejecución de una tarea específica tal como en el procedimiento de textos

ARCHIVOS: Recolección de información relacionada, puede ser datos, texto, gráficos, audio, programas, compactados, etc.

ADMINISTRACIÓN: Proceso por el cual se mantiene un sistema a punto y operativo. Es una tarea de la que se encarga el administrador y sus posibles colaboradores. Abarca acciones tales como: configuración de cuentas, dispositivos, seguridad del sistema.

BASE DE DATOS: Conjunto de información útil organizada de una forma específica y almacenada en una computadora que permite el rápido acceso, ordenamiento, análisis y salida de los datos.

COMANDO: Instrucción determinada que indica en un programa la ejecución de una acción específica como guardar, salir, conectar, etc.

CONTRASEÑA: Palabra que en su browser se visualiza con el carácter * y no puede ser leída por otra persona.

COMPILAR: Proceso por el cual se traduce un programa escrito en un lenguaje de programación a lo que realmente entiende el ordenador.

DESCARGAR: Acción mediante la cual se graba información existente en una red (texto, imágenes, audio, videos, programas, etc) en el propio duro.

HTTP: Hypertext transfer Protocol. Protocolo de la Red para la transferencia de páginas de hipertexto, o lo que es lo mismo páginas web.

LOGIN: Programa encargado de la validación de un usuario a la entrada de un sistema. Primero pide el nombre del usuario y después comprueba que el password sea el asignado a este.

MULTIUSUARIO: Capacidad de algunos sistemas para ofrecer sus recursos a diversos usuarios conectados a través de terminales

BIBLIOGRAFÍA

1. 3DEVNET, Windows Azure

http://www.3devnet.com/index.php?option=com_content&view=article&id=19&Itemid=31&lang=es

2011-02-12

2. ALFONSO, Información Salesforce

<http://alfonsogu.com/2008/05/15/%C2%BFque-es-salesforce/>

2011-01-28

3. ARCHIVE, WCF Azure

<http://archive.msdn.microsoft.com/wcfazure>

2011-03-09

4. BLOGS, Creación de Servicios

<http://blogs.renacimiento.com/rgon/Lists/Posts/Post.aspx?ID=16>

2011-02-15

5. CODE.GOOGLE, App engine crea Aplicaciones

<http://code.google.com/intl/es-ES/appengine/whyappengine.html>

2011-02-01

6. DAVIDJRH, Nuevo portal de Windows Azure y SDK 1.3

http://davidjrh.blogspot.com/2010_11_01_archive.html

2011-02-12

7. DESARROLLO WEB, Hosting

<http://www.desarrolloweb.com/faq/408.php>

2011-03-21

8. DESARROLLOWEB, Subir la aplicación a una Nube real

<http://www.desarrolloweb.com/articulos/subir-aplicacion-Nube-dotnet.html>

2011-03-23

9. ESTOY EN LA NUBE, Arquitectura de una Aplicación

<http://www.estoyenlaNube.com/recursos/windows-azure/arquitectura-de-una-aplicacion-windows-azure/>

2011-09-01

10. FORCE, La Nube de la Plataforma

<http://www.salesforce.com/platform/>

2011-01-31 **FORCE, La plataforma de Cloud Computing**

<http://www.salesforce.com/es/platform/>

2011-02-03

11. GEEKS, Configuración de la información de diagnóstico

<http://geeks.ms/blogs/ilanda/archive/2010/12/27/configuraci-243-n-de-la-informaci-243-n-de-diagn-243-stico.aspx>

2011-02-10

12. GEEKS, Conectando a SQLAzure

<http://geeks.ms/blogs/ilanda/archive/2010/06/29/migrando-a-SQL-azure-i.aspx>

2011-03-25

13. GEEKS, Migrando a SQLAzure

<http://geeks.ms/blogs/ilanda/archive/2010/06/25/conect-225-ndonos-a-SQL-azure.aspx>

2011-03-25

14. HOSTING, Metadatos

<http://hosting.udlap.mx/estudiantes/jose.ferrercz/Introduccion%20a%20los%20metadatos.pdf>

2011-03-09

15. Kit de desarrollo de software o devkit (SDK)

<http://www.alegsa.com.ar/Dic/sdk.php>

2011-02-19

16. MSDN, Archivos de configuración

<http://msdn.microsoft.com/es-es/library/ms733932.aspx>

2011-03-011

17. MSDN, Diagnostico de Nube

<http://msdn.microsoft.com/es-es/magazine/ff714589.aspx>

2011-02-12

18. MSDN, Language –Integrate Query(LINQ)

<http://msdn.microsoft.com/es-es/library/bb397926.aspx>

2011-02-17

19. MICROSOFT, Servicios Windows Azure sla

<http://www.microsoft.com/windowsazure/sla/>

2011-01-28

20. MICROSOFT, servicios en línea

<http://www.microsoft.com/online/es-es/default.aspx>

2011-02-05

21. MICROSOFT SYSTEM CENTER, permitela auto-gestión, los sistemas dinámicos

<http://www.microsoft.com/systemcenter/en/us/default.aspx>

2011-02-03

22. MICROSOFT, Windows Azuremarketplace

<http://www.microsoft.com/windowsazure/marketplace/>

2011-03-07

23. MICROSOFT, Windows Communication Foundation (WCF)

<http://msdn.microsoft.com/es-es/library/ms731082.aspx>

2011-03-14

24. MICROSOFT, traza en asp.NET

<http://msdn.microsoft.com/es-es/library/bb386420.aspx#Introduction>

2011-03-21

25. MUNDOGEEK, Compilador C# y P/Invoke

<http://mundogeek.NET/archivos/2009/07/30/c-y-pinvoke/>

2011-02-08

26. PINVOKE, complementoparaVisualStudio

<http://www.pinvoke.NET/>

2011-02-08

27. SUPPORT, AllowPartiallyTrustedCallers para una página Web utilizando Visual C#. NET.

<http://support.microsoft.com/kb/839300/es>

2011-02-10

28. SCRIBD, Sueldos en el Ecuador.

<http://es.scribd.com/doc/46234537/Ecuador>

2011-09-10

29. TELECABLE, web es searchable

http://www.telecable.es/personales/carlosmg1/glosario_s.htm

2011-03-22

30. VICRODg, Código de acceso a seguridad(CAS)

http://vicrodg.blogspot.com/2010_11_01_archive.html

2011-03-16

31. **WIKIPEDIA, Hal Software**
http://es.wikipedia.org/wiki/HAL_%28software%29
2011-02-05
32. **WIKIPEDIA, Streaming**
<http://es.wikipedia.org/wiki/Streaming>
2011-02-18
33. **WIKIPEDIA, FastCGI protocolo para interconectar programas con un servidor web**
<http://es.wikipedia.org/wiki/FastCGI>
2011-03-07
34. **WIKIPEDIA,Framework**
<http://es.wikipedia.org/wiki/Framework>
2011-03-11
35. **WIKIPEDIA, Un UniformResourceIdentifier (URI)**
http://es.wikipedia.org/wiki/Uniform_Resource_Identifier
2011-03-14
36. **WIKIPEDIA, Open DatabaseConnectivity**
<http://es.wikipedia.org/wiki/ODBC>
2011-03-18
37. **WIKIPEDIA, Entorno de desarrollo integrado**
http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado
2011-03-18
38. **WIKIPEDIA, lenguaje ensamblador**
<http://es.wikipedia.org/wiki/Assembly>
2011-03-21
39. **WINDOWS AE45 , Endpoints Configuration of Azure project**
<http://es.windowsazure4e.org/learn/doc/Tasks/Project%20Configuration/Endpoints/index.html>
2011-03-15